

(19)



**Евразийское
патентное
ведомство**

(11) **044182**

(13) **B1**

(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ**

(45) Дата публикации и выдачи патента
2023.07.28

(51) Int. Cl. **G06F 16/22 (2019.01)**
G06N 3/08 (2006.01)

(21) Номер заявки
202191759

(22) Дата подачи заявки
2019.04.04

(54) **ИЕРАРХИЧЕСКАЯ ПАМЯТЬ ПОСЛЕДОВАТЕЛЬНОСТЕЙ**

(43) **2021.12.15**

(56) US-A1-20140214852
US-A1-20100287124
US-A1-20150242743

(86) **РСТ/RU2019/000211**

(87) **WO 2020/204744 2020.10.08**

(71)(72)(73) Заявитель, изобретатель и
патентовладелец:

**СЕРЕБРЕННИКОВ ОЛЕГ
АЛЕКСАНДРОВИЧ (RU)**

(74) Представитель:
Ковальчук Н.С. (RU)

(57) Изобретение относится к области информационных и поисковых технологий, анализа и обработки информации и прогнозирования, хранения и обработки данных, к области искусственных нейронных сетей. Память Последовательностей (ПП) предназначена для ввода последовательностей и создания статистической карты весов совместной встречаемости объектов последовательностей и анализа карты для решения задач: 1) прогнозирования появления следующего/следующих объектов последовательности в прошлом или будущем; 2) определения контекста и точки смены контекста последовательности с присвоением отдельным участкам последовательности уникальных идентификаторов контекста; 3) ввода последовательностей идентификаторов контекста в ПП следующего уровня иерархии с целью создания Иерархической ПП; 4) представления причинно-следственных связей как связей взаимной встречаемости объектов разного уровня иерархии для анализа; 5) выявления причинно-следственных связей соответствующего уровня иерархии для производства выводов и суждений.

B1

044182

044182

B1

Изобретение относится к области информационных и поисковых технологий, в области анализа и обработки информации и прогнозирования, к области хранения и обработки данных, к области искусственных нейронных сетей.

Известны поисковые системы Интернет. Однако поисковые системы хранят данные в индексе и индексе является машиной нумерации объектов последовательностей. Поэтому, поисковые машины не приспособлены для хранения и поиска нумерованных последовательностей. Алгоритмы индексирования последовательностей (текстовой или другой информации) поисковыми машинами устроены так, что не хранят вес связей взаимной встречаемости объектов в последовательностях и потому не строят множества связей "будущего" и "прошлого" для каждого уникального объекта множества последовательностей. Причинами названных недостатков является то, что поисковые системы предназначены для индексирования и поиска информации, а не для создания памяти последовательностей. Индекс поисковых машин не предназначен для анализа взаимной встречаемости отдельных объектов последовательности и потому реализация такого анализа с помощью индекса поисковой машины имеет очень высокую трудоемкость.

Известны патенты "Рекурсивный индекс (РИ) для поисковых машин" RU2459242 и US9679002 (далее "патенты Серебренникова"). РИ является прототипом Памяти Последовательностей и позволяет хранить множества связей "будущего" и "прошлого" для каждого уникального объекта множества последовательностей. РИ значительно снижает трудоемкость изучения взаимной встречаемости объектов последовательностей по сравнению с индексом поисковых машин. Однако РИ является индексом и предназначен для анализа нумерованных последовательностей, что увеличивает размер хранения и не позволяет сделать на его основе Устройство Памяти Нумерованных Последовательностей. Названные патенты также не предлагают методы анализа и прогнозирования, основанные на использовании ранговых Кластеров (множеств частотных объектов будущего или прошлого).

Память последовательностей (ПП).

Прототипами ПП являются полносвязные искусственные нейронные сети (НС) с известной архитектурой. В отличие от нейронов коры мозга, отдельные нейроны НС не кодируют объектов, а потому хранимые в нейросети данные не являются памятью последовательностей таких объектов. Прогностические возможности нейросетей не детерминированы - в процессе обучения НС порождают структуру связей, которая прямо не основана на статистике встречаемости объектов в последовательностях и потому результат работы НС не вполне предсказуем. Другим существенным недостатком НС является отсутствие в НС устройства измерения и синхронизации времени, пространства и других измеримых величин, а также устройства для принятия решений с учетом эмоций и этических норм.

Другим прототипом ПП является матрица связей "каждый с каждым" (cross bar). Однако матрица связей не имеет искусственных нейронов встречаемости (ИНВ) и, кроме того, в отличие от "косынки" ПП названная матрица имеет избыточное число соединений "каждый с каждым".

Иерархическая память последовательностей (ИПП).

Прототипом ИПП являются патенты Серебренникова. Однако недостатком патентов Серебренникова является то, что РИ не предлагает способов анализа паттернов последовательностей и не предусматривает создания синтетических объектов. Поэтому РИ не может быть основой создания Иерархической Памяти Последовательностей.

Еще одним прототипом ИПП являются Искусственные Нейронные Сети (НС). Недостатком НС является то, что искусственные нейроны полносвязного слоя не кодируют отдельных объектов и потому НС не может кодировать последовательностей объектов разного уровня иерархии, таким образом делая невозможным создание ИПП в принципе. Кроме того, НС не имеют устройства синхронизации последовательностей объектов разной природы и потому не соответствуют требованиям многопоточности, что не позволяет сопоставлять последовательности объектов разной природы и делает невозможным создание сильного Искусственного Интеллекта на основе НС известной архитектуры. Названные недостатки НС являются принципиальным препятствием на пути создания сильного ИИ на основе НС.

Другим прототипом ИПП является так называемая Временная Иерархическая Память, описанная в работе "Временная Иерархическая Память" [Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp] и других работах названного автора и его соавторов. Однако в работе "Иерархия НТМ соответствует пространственной и временной иерархии реального мира" ["Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp] авторы отмечают, что предложенная ими модель Временной Иерархической Памяти (ВИП - русский аналог английской аббревиатуры НТМ (Hierarchical Temporal Memory)) имеет некоторые ограничения: "Как бы мы организовали словарный вход в сенсорном массиве, где каждая входная линия представляет различное слово, так чтоб могли быть найдены локальные пространственные корреляции? Мы еще не знаем ответа на этот вопрос, но мы подозреваем, что НТМ могут работать с такой информацией". НТМ также не предлагает механизмов создания синтетических объектов и потому не предлагает способа создания ИПП.

Задачей, на решение которой направлено изобретение, является создание технологии так называемого сильного Искусственного Интеллекта, а именно создание устройства - аналога коры мозга, а также создание способов обработки информации, обеспечивающих выявление причинно-следственных связей и производство выводов и суждений. Отличием настоящего изобретения от прототипов и аналогов явля-

ется то, что изобретение основано на представлении сознания как статистической модели мира (действия законов мира), построенной путем ввода и запоминания связей между объектами множества последовательностей, которые отражают действие законов названного мира и потому содержат статистически допустимые причинно-следственные связи, удовлетворяющие названным законам, какими бы ни были сами законы. Это позволяет ПП и ИПП в процессе обучения создавать и наполнять статистическую модель внешнего мира, способную предсказать статистически достоверные следствия известных причин или наоборот обнаружить статистически достоверные причины известных следствий.

Единый технический результат (ЕТР), который может быть получен в результате осуществления заявленного изобретения (группы изобретений), состоит в создании Иерархической Многопоточной Синхронной Памяти Ненумерованных Последовательностей.

Для пояснения сущности заявленного изобретения представлены следующие графические материалы.

- Фиг. 1 - Временная Иерархическая Память согласно Джеффу Хокинсу.
- Фиг. 2 - Диаграмма Кластера K_N ключевого объекта.
- Фиг. 3 - Окно Внимания из пяти объектов последовательности.
- Фиг. 4 - Фрагмент последовательности. Веса w Кластеров уменьшаются от $K1$ к $K4$.
- Фиг. 5 - Последовательность, представленная Кластерами 1-го ранга.
- Фиг. 6 - Обучающая обратная связь. Сообщает предыдущему хиту (объекту) каким оказался новый объект последовательности.
- Фиг. 7 - Упреждающая связь построения гипотез.
- Фиг. 8 - Обучение и прогнозирование в РИНП.
- Фиг. 9 - Упреждающие связи известных объектов указывают на один и тот же участок последовательности в будущем, которое мы хотим спрогнозировать.
- Фиг. 10 - Рост числа гипотез по мере увеличения глубины прогнозирования.
- Фиг. 11 - Ключевой Объект (КО), а также по три частотных объекта (-3, -2 и -1) и (1, 2 и 3), расположенных, соответственно, до и после Ключевого Объекта в конкретной последовательности.
- Фиг. 12 - Сформированные ранговые Кластеры - три для прошлого K^{-3} , K^{-2} , K^{-1} и три для будущего K^1 , K^2 , K^3 .
- Фиг. 13 - Введенный фрагмент последовательности, состоящий из трех объектов.
- Фиг. 14 - Когерентные кластеры объектов C_2 , C_3 и C_4 ...
- Фиг. 15 - Пример трех последовательностей.
- Фиг. 16 - Частотные объекты Кластера ключевого объекта А.
- Фиг. 17 - Множество последовательностей с объектом В.
- Фиг. 18 - Множество последовательностей с объектом С.
- Фиг. 19 - Множество последовательностей с объектом D.
- Фиг. 20 - Кластеры прошлого элементов В, С и D.
- Фиг. 21 - Обратная проекция Кластера.
- Фиг. 22 - Ранговая Обратная Проекция.
- Фиг. 23 - Появление одного объекта для обозначения нескольких эквивалентных смыслов.
- Фиг. 24 - Сжатие Кластеров четырех объектов до одного Кластера $Cont(4)$.
- Фиг. 25 - Замена последовательности объектов последовательностью Труб.
- Фиг. 26 - Диаграмма Эйлера-Венна для логического отрицания.
- Фиг. 27 - Формирование гипотез показано красными пунктирными стрелками.
- Фиг. 28 - Обратно-упреждающие связи между Трубами.
- Фиг. 29 - Образование обратно-упреждающих связей между Трубами.
- Фиг. 30 - Использование иерархии связей для производства выводов.
- Фиг. 31 - Вход S шины С.
- Фиг. 32 - Функциональная схема Памяти Последовательностей.
- Фиг. 33 - Градация связей между объектами последовательностей. Объект 1 является самым поздним из введенных объектом последовательностей, а объект N самым ранним.
- Фиг. 34 - Соединение "каждый с каждым" в форме матрицы $N \times N$ шин (cross bar).
- Фиг. 35 - Половина матрицы - косынка (Half Cross bar), здесь А - входы шин и В - выходы шин.
- Фиг. 36 - Запись и считывание связей в объединенном узле косынки матрицы.
- Фиг. 37 - Пример коммутации соединения "сам с собой".
- Фиг. 38 - Обратная рекуррентная связь показана красной стрелкой.
- Фиг. 39 - Регулярная одноранговая матрица из двух секций.
- Фиг. 40 - Последовательное соединение двух секций одноранговых косынок.
- Фиг. 41 - Коммутация двуранговой матрицы как пример коммутации многоранговой матрицы.
- Фиг. 42 - Генератор матрицы.
- Фиг. 43 - Топология матрицы из шести одноранговых секций со связями 1-го, 2-го, 3-го, 4-го, 5-го, 6-го ранга. Зеленым показан двухранговый генератор матрицы со связями 1-го и 2-го рангов.
- Фиг. 44 - Направления Вход-выход:

1) \Rightarrow - запись (записываются обратные связи);

2) \Rightarrow - считывание прошлого;

3) \Leftarrow - считывание будущего.

Фиг. 45 - Шины объектов показаны синим, а шины Труб показаны зеленым.

Фиг. 46 - Два счетчика встречаемости для прямого и обратного порядка объектов $C_n \rightarrow C_k$ и $C_k \rightarrow C_n$.

Фиг. 47 - Считывание значения и направления инверсии.

Фиг. 48 - Нейрон Встречаемости.

Фиг. 49 - Нейрон - 1; шины объектов C_1 и C_2 - 2 и 3; вентили шин объектов C_1 и C_2 в положении "открыто" - 4 и 5; связь между шинами объектов - 6; вентиль связи шины между объектами в положении "закрыто" - 7.

Фиг. 50 - Нейрон - 1; шины объектов C_1 и C_2 - 2 и 3; вентили шин объектов C_1 и C_2 в положении "закрыто" - 4 и 5; связь между шинами объектов - 6; вентиль связи шины между объектами в положении "открыто" - 7.

Фиг. 51 - Счетчики - 1 для направлений $C_k \rightarrow C_n$ и $C_n \rightarrow C_k$, запись в память счетчика на пересечении шин объектов C_1 и C_k в косынке ранга k производится только при одновременной подаче сигналов $S_1=1$ и $S_k=(1-\Delta S_{1,k})$ на шины объектов C_1 и C_k косынки ранга k в направлении обратных связей; связь взаимной встречаемости -2; вентиль связи переведен в положение "открыто" - 3.

Фиг. 52 - Сила сигналов S_i на шинах объектов C_i .

Фиг. 53 - Счетчик - 1, вентиль в положении "открыто" - 2.

Фиг. 54 - Считывание обратной связи.

Фиг. 55 - Считывание рангового Кластера многогранговой матрицы.

Фиг. 56 - Считывание ранговых связей многогранговой матрицы.

Фиг. 57 - Считывание весов трех последовательных ранговых связей матрицы.

Фиг. 58 - Считывание полного Кластера многогранговой матрицы.

Фиг. 59 - Считывание весов трех последовательных ранговых связей матрицы.

Фиг. 60 Схема Искусственного нейрона иерархии памяти последовательностей (ИНИ). ИНИ обеспечивает связь соседних слоев иерархии Памяти Последовательностей М1 и М2: А - сенсоры с функцией активации (ϕ) сенсора для получения Кластер и Калибра Трубы с весами частотных объектов на выходе матрицы М1 объектов нижнего уровня иерархии, В - сумматор (Σ) весов частотных объектов Кластера Трубы с функцией активации (ϕ) нейрона, С - связи шины нейрона с шинами объектов матрицы М2 Памяти Последовательностей верхнего уровня иерархии, D - сенсоры для запоминания Окна Внимания - объектов Генератора Трубы на входе матрицы М1 объектов нижнего уровня иерархии, Е - обратная связь выхода нейрона с объектами Генератора Трубы на входе матрицы М1 нижнего уровня иерархии.

Фиг. 61 - Схема искусственного нейрона традиционной нейросети (персептрон).

Фиг. 62 - Схема коммутации нейрона искусственного нейрона иерархии памяти последовательностей (ИНИ) с матрицами нижнего уровня М1 (слой Объектов) и матрицы М2 (слой Труб 1-го рода) с помощью ИНИ. А - Кластер весов частотных объектов, В-сумматор нейрона с функцией активации, С - пересечения выходной шины нейрона с шинами матрицы верхнего уровня М2 (Труб 1-го рода).

Фиг. 63 - Архитектура матриц различных уровней иерархии.

Фиг. 64 - Расположение групп сенсоров и сумматоров ИНИ в косынках. 1 - сенсоры группы D, 6 - сенсоры группы А, группа сумматоров В размещена на выходах (желтый блок В).

Фиг. 65 - Окно внимания показано зелеными стрелками в группе D сенсоров.

Фиг. 66 - Обучение Нейрона Сочетаний.

Фиг. 67 - Работа Нейрона Сочетаний.

Фиг. 68 - Схема коммутации искусственного нейрона сочетаний памяти последовательностей (ИНС) с матрицами нижнего уровня М1 (слой Объектов) и матрицы М2 (слой сочетаний) с помощью ИНС. А - Кластер весов частотных объектов, В - сумматор нейрона с функцией активации, С - пересечения выходной шины нейрона с шинами матрицы верхнего уровня М2 (Слоя сочетаний), D - объекты сочетания, а также нейрон.

Фиг. 69 - Слой устойчивых сочетаний.

Фиг. 70 - Функциональная схема рекуррентной Памяти Последовательностей.

Фиг. 71 - Воспоминание Трубы по ненормализованному Кластеру.

Фиг. 72 - Активные частотные шины показаны красным, а пассивные зеленым.

Фиг. 73 - Калибр Труб слоя измерений.

Фиг. 74 - Слой частотных шин без пересечений показан цветными линиями, а слой меток серыми с пересечениями "каждый с каждым" как в слое частотных шин, так и в слое меток.

Фиг. 75 - Слой частотных шин синхронизации измерений. В данном случае показана троичная система исчисления, имеющая по три шины в каждом разряде.

Фиг. 76 - Слои синхронизации измерений в архитектуре косынки.

Фиг. 77 - Архитектура матрицы со слоями измерений.

Фиг. 78 - Архитектура матрицы с слоем измерений и группами сенсоров А и А1, D и D1, генерато-

ром измерений G, Сумматорами В и В1, а также сенсором С.

Фиг. 79 - Строение крупной пирамидной клетки коры большого мозга V слоя (по Г.И. Полякову).

Фиг. 80 - Модель пирамидального нейрона.

1. Вступление.

Хотя первые нейронные сети (нейросети) являлись полностью связными сетями, состоящими из перцептронов, наибольшее распространение в настоящее время получили архитектуры Сверточных Нейросетей (СНС или Convolutional Neural Network - CNN). СНС используют каскад блоков свертки и линеаризации (ReLU - rectified linear unit) карт признаков и лишь в качестве последнего блока обработки все также полностью связную сеть перцептронов.

Число исследователей нейросетей довольно велико, а инвестиции в эту область стремительно растут, однако это до сих пор не привело к появлению универсального ИИ (искусственного интеллекта), а состоятельность подхода к созданию универсального ИИ на основе нейросетей подвергается сомнению.

В 2003 году Джефф Хокинс издал труд "Об интеллекте" ["On Intelligence", Jeff Hawkins & Sandra Blakeslee, ISBN 0-8050-7456-2], в котором он отметил как недостаток подхода коннективистов (энтузиасты нейросетей) отсутствие у них достаточных знаний о работе мозга и ключевых качествах человеческого интеллекта. Свой подход Хокинс называет "Biological and Machine Intelligence (BAMI)" - Биологический и Машинный Интеллект (БИМИ). В рамках предлагаемого БИМИ подхода Хокинс впервые переформулировал содержание известного "поведенческого" Теста Тьюринга на наличие интеллекта: предсказание, а не поведение является доказательством интеллекта. В своих работах Хокинс приходит к выводу, что физическим носителем человеческого интеллекта является неокортекс, ключевыми функциями которого являются:

Неокортекс хранит последовательности паттернов.

Неокортекс вспоминает паттерны автоассоциативно.

Неокортекс хранит паттерны в инвариантной форме.

Неокортекс хранит паттерны иерархически.

Позже в своей работе 2006 года, названной "Временная Иерархическая Память" ["Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp], авторы предлагают техническую концепцию памяти (фиг. 1), которая реализует хранение пространственных паттернов и временных последовательностей паттернов.

Тем не менее, в главе 3.2. "Иерархия НТМ соответствует пространственной и временной иерархии реального мира" ["Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp] авторы отмечают, что предложенная ими модель Временной Иерархической Памяти (ВИП - русский аналог английской аббревиатуры НТМ (Hierarchical Temporal Memory)) имеет некоторые ограничения: "Как бы мы организовали словарный вход в сенсорном массиве, где каждая входная линия представляет различное слово, так чтоб могли быть найдены локальные пространственные корреляции? Мы еще не знаем ответа на этот вопрос, но мы подозреваем, что НТМ могут работать с такой информацией".

Теория и практика нейросетей до сих пор игнорировали тот факт, что все явления природы представлены наблюдаемыми последовательностями, в которых законы мироздания проявляются как статистическая повторяемость, а Джефф Хокинс не смог уйти от концепции "карты признаков", которая сама по себе не является последовательностью. Настоящее описание исходит из представлений о мозге, как о памяти последовательностей, где картина мира представлена связями, вес которых зависит от повторяемости связей в последовательностях в природе.

2. Память Последовательностей (ПП).

2.1. Общий подход.

2.1.1. Концепция взаимной встречаемости.

Рекурсивный индекс (РИ) для поисковых машин (RU2459242, US9679002) является памятью последовательностей. Развитие Рекурсивного индекса позволяет хранить как сами последовательности, так и последовательности паттернов (в патентах названы "сферой", "будущим" и "прошлым") каждого из объектов последовательностей. РИ значительно снижает трудоемкость изучения взаимной встречаемости объектов последовательностей, что имеет ключевое значение для развития ИИ.

РИ реализует следующие алгоритмы:

1. индексирования последовательностей объектов (Ключевые объекты),

2. поиска в последовательностях Ключевого Объекта,

3. извлечения из индекса последовательностей длиной R объектов (R-последовательности), расположенных до и/или после Ключевого Объекта,

4. построения (+R)-полусферы будущего, состоящей из всех найденных в индексе R-последовательностей, начинающихся Ключевым объектом и (-R)-полусферы прошлого, состоящей из всех найденных в индексе R-последовательностей, заканчивающихся Ключевым Объектом,

5. построения R-сферы объектов (Частотные объекты), объединяющей Ключевые объекты последовательностей (+R)-полусферы будущего и (-R)-полусферы прошлого.

Все Частотные объекты, попадающие в (+R)-полусферу будущего и (-R)-полусферу прошлого конкретного Ключевого объекта формируют соответственно Кластер будущего и Кластер прошлого этого

Ключевого объекта. Для любого Ключевого объекта можно построить два типа Кластеров - Кластер прошлого и Кластер будущего. Для того чтобы учесть, что Кластер содержит частотные объекты из "прошлого" и "будущего", объектам из будущего будем присваивать знак плюс, а объектам из прошлого - знак минус. Поскольку РИ реализует память последовательностей, то далее о Рекурсивном индексе все чаще станем говорить, как о памяти последовательностей.

ПП позволяет обнаруживать и исследовать пространственные и временные корреляции внутри и между последовательностями и в своей основе опирается на концепцию анализа взаимной встречаемости объектов последовательностей.

2.1.2. От текстовых документов к последовательностям объектов любой природы.

Договоримся считать, что последовательности состоят из конечного множества уникальных объектов и эти объекты могут сочетаться в последовательностях по неизвестным нам правилам.

Под поиском в современных поисковых системах понимается ввод уникального объекта (ключевое слово или фраза), вхождение которого надо найти в хранимых последовательностях. Поисковые машины Интернет создавались для работы с документами, поэтому поисковики оперируют понятием "номер документа", а порядок следования слов в документе определяется порядковыми номерами слов документа ("позиция слова в документе"). В более общем случае от понятия "документ" следует перейти к понятию "цепи событий/объектов" или "последовательности событий/объектов", а от понятия "номера документа" к понятию "номер цепи" или "номер последовательности". Поскольку события происходят, и объекты появляются в пространстве и времени, то в общем случае в качестве "номера цепи" следует использовать штамп времени/места поступления в память объекта цепи данных. А когда мы не можем установить абсолютное время наступления события, то мы можем определить время наступления событий как сдвиг относительно времени начала последовательности. Так, если мы не можем знать, когда именно произошло событие, снятое на видео, то мы определенно можем сказать на какой минуте/секунде или даже на каком по порядку кадре видео такое событие произошло. Тем не менее для простоты мы часто будем прибегать к примерам последовательностей текстовой информации.

2.1.3. Устойчивые сочетания, понятия.

Возьмем в качестве примера последовательностей человеческую речь или последовательность слов в текстах и исследуем совместную встречаемость слов речи или текстов. Например, известно, что НЛП является аббревиатурой словосочетания "нейролингвистическое программирование". Поэтому сочетание двух слов "нейролингвистическое" встречается в речи и текстах часто, а словосочетание с обратным порядком слов "лингвистическое нейро" практически никогда.

Если бы мы захотели создать машину, которая использует различие частоты совместной встречаемости прямого и обратного словосочетаний в качестве критерия устойчивости словосочетания или критерия нового смысла порождаемого сочетанием, то мы могли бы взять множество примеров устойчивых словосочетаний и словосочетаний, которые порождают новые понятия (новые понятия, часто обозначаются аббревиатурами), статистически замерить соотношение M весов прямой и обратной встречаемости слов таких сочетаний и использовать это соотношение для автоматического определения устойчивости таких словосочетаний и порождение ими новых понятий.

Простейшим решением для поиска величины M взаимной встречаемости любых двух слов языка было бы создание таблицы $N \times N$, где в качестве наименования столбцов и строк выступали бы N слов языка, например, перечисленных в алфавитном порядке. Если в ячейку на пересечении строки i и столбца j заносить число случаев Q , когда слова i и j встретились в порядке $i \rightarrow j$, а в ячейку на пересечении строки j и столбца i заносить число случаев W , когда слова i и j встретились в обратном порядке $j \rightarrow i$, то в ячейках, симметричных относительно диагонали, окажутся числа Q и W для каждой пары слов i и j . Собственно $M=Q/W$. При переходе к исследованию взаимной встречаемости трех объектов нам пришлось бы рассматривать не таблицу, а куб размером $N \times N \times N$ и для исследования взаимной встречаемости R объектов объем куба увеличился бы до размера N^R .

Логика исследования взаимной встречаемости может быть использована для выявления устойчивых словосочетаний, не образующих понятия, таких как, например "тигр", "полосатый" и "лед", ведь очевидно, что пара слов "тигр" и "полосатый" вместе встречается чаще, чем пара слов "тигр" и "лед".

Следуя описанной логике, можно исследовать взаимную встречаемость также объектов, не образующих сочетания, а разделенных некоторым числом других объектов. Замечание 1 (Снижение трудоемкости определения веса взаимной встречаемости): Механизм выявления устойчивых словосочетаний и понятий путем построения куба размера N^R прост, но трудоемкость способа достаточно высока. Использование Рекурсивного Индекса позволяет существенно снизить трудоемкость задачи исследования взаимной встречаемости путем построения вокруг объекта i шара из K фрагментов последовательностей радиусом R объектов до и после объекта i , и исследования взаимной встречаемости объекта i с другими объектами шара, что позволяет решать задачу на множестве объектов $2 \times R \times K$, а не на множестве объектов N^R , за счет чего трудоемкость снижается и это позволяет решать задачу взаимной встречаемости с использованием слабых процессоров и "на лету".

2.1.4. Направленные связи между объектами последовательностей.

На какую связь между словами может указывать тот факт, что два или более слов не образуют словосочетаний, однако чаще других слов вместе встречаются в одном тексте, даже будучи разделенными другими словами? О чем может говорить тот факт, что такие слова могут появляться в тексте или речи вместе, но чаще в "прямом", а не в "обратном" порядке?

Обратимся теперь к более общему примеру - к событиям. Если бы мы изучали события, снятые на камеру, то могли обнаружить, что в цепи событий возникновения пожара появление дыма чаще всего предшествует появлению огня. Исследуя тексты или видео, содержащие описание или кадры возникновения пожара, мы могли бы обнаружить то же самое - сначала появляется дым, а затем огонь или наоборот. Вместе с тем, слова "дым" и "огонь" не обязательно образуют словосочетание, а могут быть разделены множеством других слов. То же самое можно сказать о словах "полосатый" и "тигр", они могут не образовывать словосочетания, но часто находятся в одном и том же описании тигра или событий с участием тигра. В более общем случае, видимо, разумно ожидать, что в последовательности событий событие-причина всегда предшествует событию-следствию, таким образом, пара этих событий образует устойчивую направленную последовательность событий «причина»→«следствие», разделенных другими промежуточными событиями, причем видимо частота встречаемости прямой последовательности «причина»→«следствие» будет выше, чем частота встречаемости последовательности «следствие»→«причина».

Вместе с тем, выявление причинно-следственных является выводом, а значит машина, позволяющая делать выводы о совместной встречаемости объектов в последовательности, является машиной, позволяющей делать выводы или "думающей" машиной.

Таким образом, если мы хотим построить машину, делающую выводы, то нам следует построить машину анализирующую взаимную встречаемость каждого объекта с каждым в множестве последовательностей событий. Такая машина будет машиной выявления причинно-следственных связей между объектами, разделенными большим объемом промежуточной информации.

Теперь наша машина умеет делать выводы. Однако если воспользоваться техникой построения таблиц, которая была приведена выше, то трудоемкость алгоритма такой машины будет пропорциональна N в степени N . Поэтому нам следует построить аппарат представления, записи и анализа последовательностей, который позволяет вести их анализ и делать выводы "на лету".

2.1.5. Одновременные и параллельные последовательности.

В работах Джеффа Хокинса "On Intelligence", Jeff Hawkins & Sandra Blakeslee, ISBN 0-8050-7456-2], ["Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp] словосочетание "временная иерархическая память" содержит слово "временная", которое следует понимать, как последовательность пространственных паттернов, следующих один за другим во времени. Однако для выявления временных корреляций между временными последовательностями паттернов, поступающими в память и уже хранимыми в ней, необходимо определить понятие "одновременности". В быту мы называем одновременными события, происходящие в одно и то же время, однако переходя к сохраненным последовательностям, мы часто не знаем, когда они были записаны, и потому понять, являются ли последовательности одновременными по времени их проявления, можно только сравнивая и анализируя сходство событий и объектов таких последовательностей. В общем случае, одновременными мы станем называть последовательности, общим началом или концом которых является один и тот же уникальный объект или время, или место. Проявления такого объекта в разных каналах поступления информации (зрение, слух ...) можно отнести к разным проявлениям одного и того же объекта именно благодаря одновременности поступления информации об объекте по разным каналам информации.

Однако не все одновременные последовательности могут коррелировать между собой. Так видеозапись кошки и аудиозапись ее мяукания могут коррелировать, если названные записи являются записями одного и того же события, а падение метеорита в одной части Земли и рождение ребенка в другой могут не иметь корреляций.

Параллельными станем называть коррелирующие между собой одновременные последовательности. Разные словоформы слова являются примером вырожденной формы параллельных последовательностей, каждая из которых состоит из самого объекта в разных словоформах. Другой формой параллельных последовательностей могут быть объекты-синонимы и сочетания-синонимы. Еще одним примером параллельных последовательностей паттернов может служить множество текстов на разных языках, каждый из которых является переводом одного и того же исходного текста или два текста с описанием одного и того же события, но написанные разными людьми. Использование параллельных текстов легло в основу обучения системы машинного перевода Google, в рамках которой нейросеть создала свой абстрактный язык, карта смыслов которого представлена в работе Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation [Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation (<https://arxiv.org/abs/1611.04558>) и русский вариант <https://m.geektimes.ru/post/282976/>]. Работа Google в режиме обучения использует тексты, заведомо удовлетворяющие условию наличия смысловой корреляции между ними, и это напоминает процесс познания реальности физического мира мозгом - он обучается на примерах предъявления через разные органы

чувств заведомо параллельных последовательностей: мы видим кошку и слышим ее мяукание. Мозг, благодаря механизмам обнаружения временных корреляций между заведомо параллельными чувственными временными последовательностями аудио- и видеопаттернов, делает заключение о принадлежности звука "мяукание" наблюдаемому объекту - кошке. Таким образом, предъявление параллельных временных последовательностей является ключевым для обучения, а аппарат выявления корреляций является базовым механизмом производства прогнозов (выводов), а "многопоточность" (Многопоточная память последовательностей) памяти последовательностей, возможно, является абсолютно необходимой для ИИ.

Мы способны обдумывать что-то лежа, без движения, в тишине и с закрытыми глазами, в этот момент процесс "обдумывания" работает с объектами памяти, а не с объектами реальности, и потому для абстрактного мышления критически важной является возможность выявлять корреляции между последовательностями памяти, часть из которых не является одновременными, а часть одновременных не является параллельными. Выявление факта параллельности двух или более последовательностей, размещенных в памяти, по существу, и является одним из базовых механизмов абстрактных выводов ИИ. Прогнозирование развития реальных событий также связано со способностью обнаруживать корреляции: если найдена корреляция вводимой последовательности с последовательностью, размещенной в памяти, то последовательность из памяти можно использовать в качестве прогноза появления объектов вводимой последовательности. Поэтому нам нужен аппарат "временной памяти последовательностей", архитектура которого позволяет:

- извлекать "одновременные" последовательности;
- выявлять временные корреляции между одновременными последовательностями и определять, какие из последовательностей являются параллельными.

Современный человек научился фиксировать и измерять значительно больше, чем позволяют ему его собственные органы чувств. Анализ параллельных последовательностей мира (изменения геомагнитного, гравитационного поля, силы солнечного ветра и так далее, последовательности природных явлений) и сравнение их с событиями в жизни и поведением людей могут привести к неочевидным и потому неожиданным результатам и открытиям.

2.1.6. Многопоточная память последовательностей.

Говоря о последовательностях и памяти последовательностей, мы абстрагируемся от природы рассматриваемых последовательностей: это могут быть последовательности образов зрительных (зрение) или звуковых (слух) или осязательных или других человеческих чувств и ощущений. Последовательностями могут быть также данные приборов измеряющих изменение полей, скоростей, местоположений и других измеряемых показателей. Причем все эти последовательности могут быть одновременными во времени и/или пространстве и параллельными по смысловой значимости, а, значит, такие последовательности могут отражать одно и то же проявление реальности, являться частями одного процесса или явления, и потому между последовательностями должна наблюдаться корреляция. Такая корреляция дает интеллекту информацию, необходимую для того, чтобы делать выводы, которые были бы без нее неочевидны. Это означает, что память последовательностей должна уметь одновременно обрабатывать множество последовательностей разной природы, объекты которых подаются в память последовательностей по разным каналам связи памяти с реальностью и уметь устанавливать между такими последовательностями связь. Обработку памяти нескольких последовательностей далее будем называть многопоточной или многоканальной памятью последовательностей.

2.1.7. Память последовательностей и логика.

Логику описывают как "науку о законах мышления и его формах", а также как "ход рассуждений, умозаключений", а логические элементы являются основой современных компьютеров. При переходе от вычислительной модели к модели нейронной сети возникает вопрос: нужно ли реализовывать аппарат логики отдельно от памяти последовательностей или память последовательностей и является аппаратом логики, аппаратом "рассуждений и умозаключений"?

Если бы люди не могли представить описание логики в текстовом виде - в виде последовательности, то знание о такой логике невозможно было бы передать через рукописи потомкам. Поэтому человечество оперирует логикой, описанной последовательностями: любая известная логика (классическая и другие) имеет текстовое и формализованное описание, а каждый текст или формула является последовательностью. Поэтому можно утверждать: любой логический аппарат, который можно представить последовательностью, также можно запомнить и воспроизвести с помощью памяти последовательностей. И наоборот: только логический аппарат, который невозможно описать последовательностью, также невозможно запомнить или воспроизвести с помощью памяти последовательностей. С этой точки зрения аппарат логики является формальным описанием работы памяти последовательностей, а не наоборот.

2.2. Некоторые нейроанalogии процессов памяти последовательностей
Переходя к Памяти Последовательностей, мы фактически переходим к имитации работы нейронов мозга и потому будет уместно договориться о некоторых аналогиях.

2.2.1. Первичные, вторичные и т.д. нейроны.

Известно, что внешние образы способны возбуждать конкретные нейроны коры, это было проде-

монстрировано на примере "нейрона Била Клинтона" или "нейрона Дженнифер Анистон". Другими словами, есть нейроны, закрепленные за объектами реального мира и для удобства, мы станем их называть Первичными нейронами. Кора мозга по сути является моделью внешнего мира, и последовательностям объектов внешнего мира должны соответствовать последовательности возбуждений отдельных первичных нейронов коры. Первичные нейроны мозга возбуждаясь, передают свое возбуждение множеству нейронов (вторичные, третичные и так далее нейроны), с которыми они связаны прямыми или обратными связями.

2.2.2. Связность нейронов.

Нейроны могут иметь прямые и обратные связи, а также боковые связи. Боковыми связями будем считать связи, допускающие сравнение между собой нейронов, связанных боковой связью.

2.2.3. Возбуждение: порождение Кластера.

В нашей аналогии первичным нейронам соответствуют ключевые объекты последовательности, а вторичным нейронам частотные объекты Кластера ключевого объекта. Силе синапса между первичным и вторичным нейронами в нашей модели соответствует вес частотного объекта в Кластере ключевого объекта последовательности.

2.2.4. Возбуждение обратных связей: обратная проекция Кластера.

Поскольку нейроны имеют не только прямые, но и обратные связи, то вторичные нейроны, возбуждаемые по обратной связи, будут находиться в условном "прошлом" первичного нейрона. Ниже мы покажем, что обратная проекция возбуждения имеет значение для обнаружения параллельных смыслов (синонимия в широком смысле). В модели Рекурсивного Индекса (или Памяти Последовательностей) обратное возбуждение нейронов будет представлено обратной проекцией Кластера, а именно - построением Кластеров прошлого для каждого из частотных объектов исходного Кластера и определение их суперпозиции.

2.2.5. Последовательность возбуждений: суперпозиция Кластеров.

Если возбуждать в мозге первичные нейроны в некоторой последовательности, то часть вторичных нейронов будет возбуждаться повторно чаще, чем другие вторичные нейроны. В результате часть вторичных нейронов будет оставаться возбужденными, в части нейронов возбуждение будет затухать. Благодаря интерференция возбуждений в коре мозга могут возникать волны возбуждения и затухания ["Compression and Reflection of Visually Evoked Cortical Waves" (https://www.researchgate.net/publication/6226590_Compression_and_Reflection_of_Visually_Evoked_Cortical_Waves)]. Для моделирования такого волнового возбуждения нейронов традиционно используются модели осциллирующих нейронов. Однако, первичные нейроны связаны не со всеми нейронами коры, а лишь с некоторыми - "вторичными нейронами", поэтому волна возбуждения может передаваться не всем окружающим нейронам, а лишь тем, с которыми возбужденный нейрон имеет направленную связь, сформированную в процессе обучения вводом последовательностей. В модели Рекурсивного Индекса (или Памяти Последовательностей) последовательность возбуждений будет представлена суперпозицией Кластеров.

2.2.6. Ослабление: снижение веса объекта.

Поскольку возбуждение нейронов со временем ослабевает, то возбуждение первичных нейронов (аналоги объектов вводимой последовательности) будет уменьшаться пропорционально "расстоянию" до последнего возбужденного нейрона в последовательности. Чем дальше находится ранее возбужденный первичный нейрон от последнего возбужденного первичного нейрона последовательности, тем меньше он возбужден - его возбуждение ослабевает больше чем у нейронов, которые были возбуждены позднее него.

2.2.7. Окно внимания.

Под Окном Внимания мы будем понимать очередь объектов последовательности определенного размера. Во время общения мы способны в точности воспроизвести только несколько последних слов, которые слышали, а остальное помним "по смыслу". Те слова, которые мы помним и можно назвать Окном Внимания. В рамках нейроанalogии Окно Внимания можно представить как очередь последовательно возбужденных нейронов, уровень возбуждения которых снижается из конца в начало очереди. Таким образом, Окном Внимания является очередь N первичных нейронов, в которой "последний вошедший" будет самым возбужденным, а возбуждение "первого вышедшего" будет самым ослабленным. То есть возбуждение всех первичных нейронов начиная с $(N+1)$ -го первичного нейрона в прошлом считаем полностью затухшими. Строго говоря, из-за наличия прямых и обратных, а также боковых связей, первичные нейроны из начала очереди могут возбуждаться первичными нейронами из конца очереди, поэтому Окно Внимания - не имеет строгой нейроанalogии и скорее является последним известным фрагментом последовательности, порядок следования объектов в котором определен.

2.2.8. Паузы: Прерывания ввода.

Прерывания последовательностей имеют больше значение, поскольку прерывание последовательности может означать изменение контекста. Примером прерывания в текстах могут служить знаки препинания. Значение прерываний хорошо видно на известном примере: "Казнить нельзя помиловать", в котором расстановка знаков препинания полностью меняет смысл сказанного. Прочтите фразы: "Казнить

нельзя помиловать" "Казнить нельзя, помиловать" и "Казнить, нельзя помиловать" и обратите внимание, что знаку препинания в речи соответствует пауза. Поскольку мозг сначала постигает речь, а затем письменность, то постижение смысла речи связано не со знаками препинания, а с паузами в речи: "Казнить < _____ > нельзя помиловать" или "Казнить нельзя < _____ > помиловать".

Не только речь, но и зрение использует паузы - прерывания ввода. Прерыванием зрения может служить пауза между саккадами глаз при переводе взгляда с одного места на другое, ведь каждая саккада, по сути, порождает отдельный контекст - саккада с фокусом на нос, саккада с фокусом на глаза, саккада с фокусом на губы... Таким образом, благодаря саккадам распознавание изображений также связано с распознавания последовательности изображений. Известно, что при распознавании лиц глаза человека исследуют несколько разных элементов лица (глаза, нос и так далее) и благодаря саккадам процесс узнавания превращается в процесс узнавания последовательности изображений разных элементов лица. Нейронные сети распознавания лиц вместо этого работают со статическими изображениями, а вместо работы с последовательностями используют технологии свертки, пуллинга и другие техники нейросетей для работы с картами признаков. Поэтому при обработке изображений с помощью нейросети, подача на вход нейросети последовательности элементов лица (нос, глаза и так далее) может оказаться более оптимальным с точки зрения скорости и качества распознавания лица или другого изображения. Порядок подачи элементов лица на вход нейронной сети для распознавания также может иметь значение, поэтому подавать последовательность элементов возможно следует в одном и том же порядке. Это соответствовало бы привычке конкретного человека рассматривать лицо в определенной последовательности саккад.

Таким образом паузы являются одним из признаков изменения контекста последовательности, и это следует использовать.

Прерываниями также могут служить как определенный уровень эмоций, так и нарушения этических норм.

2.3. Кластеры.

2.3.1. Пространственные паттерны объектов и их последовательностей.

Зрительный образ любого объекта представляет собой пространственный паттерн пикселей, который научились успешно распознавать сверточные нейронные сети.

В случае последовательностей пространственным паттерном каждого уникального объекта последовательности является множество его связей с другими уникальными объектами с учетом частоты их совместной встречаемости в последовательностях. Поэтому мы станем называть такие связи "частотными связями", которые станем обозначать идентификаторами самих уникальных объектов, с которыми такая связь установлена. Поэтому иногда вместо словосочетания "частотные связи" мы будем использовать словосочетание "частотные объекты". Вес связи будет определяться частотой совместной встречаемости.

Любой пространственный паттерн далее будем называть Кластером. Кластеры будем формировать путем анализа взаимной встречаемости ключевого объекта Кластера с другими уникальными объектами последовательностей. Мерой сходства объектов и последовательностей между собой будем считать меру сходства их Кластеров (пространственных паттернов). Биологическим аналогом уникального объекта является нейрон коры мозга, а Кластер в этой аналогии играет роль набора синапсов, связывающих этот нейрон с другими нейронами мозга.

Каждый Кластер является множеством объектов и потому над Кластерами можно проводить операции как над множествами. Вместе с тем каждому частотному объекту

Кластера присвоен весовой коэффициент и потому Кластер также является массивом или матрицей, или тензором. Кластер также можно представить вектором.

2.3.2. Инвариантность паттернов.

Поскольку Кластеры являются отражением взаимной встречаемости объектов в последовательностях, то Кластер представляет собой контекст появления объекта в последовательностях и потому является инвариантным представлением объекта. В частности, Кластер может быть инвариантным по отношению к словоформам одного слова и к его синонимам. Словоформы и синонимы являются смысловыми копиями друг друга и потому их Кластеры должны быть похожи. В случае текстовых последовательностей не важно в какой словоформе появляется само ключевое слово в тексте, но важно какие частотные слова попадут в Кластер ключевого слова. Инвариантность Кластера позволяет Рекурсивному Индексу искать параллельные фрагменты и синонимы.

2.3.3. Каков размер Кластера?

Насколько различимы частоты слов в Кластере? Ответ на этот вопрос дают известные эмпирические закономерности, называемые законом Хипса и законом Ципфа. Закон Ципфа гласит: "Если все слова языка (или просто достаточно длинного текста) упорядочить по убыванию частоты их использования, то частота n-го слова в таком списке окажется приблизительно обратно пропорциональной его порядковому номеру n (так называемому рангу этого слова). Например, второе по используемости слово встречается примерно в два раза реже, чем первое, третье - в три раза реже, чем первое, и так далее". Таким образом частоты слов в Кластере будут различаться обратно пропорционально их рангам в Кластере.

Согласно закону Хипса, число уникальных слов в тексте пропорционально квадратному корню из числа всех слов текста. Таким образом, Кластер, построенный на корпусе последовательностей из 10 ты-

сяч слов, будет содержать всего 100 уникальных слов, частотность которых будет убывать обратно пропорционально рангу слов в списке Кластера, согласно закону Ципфа. Последнее - сотое частотное слово будет встречаться в исходном тексте в 100 раз реже чем первое частотное слово в списке инвариантов Кластера.

Если частоту использования первого частотного слова принять равной единице, то частоту использования k слов в соответствии с законом Ципфа можно представить "гармоническим рядом" (расчеты 1. Сумма гармонического ряда):

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{k} + \dots$$

Сумма первых n членов гармонического ряда будет:

$$s_n = \sum_{k=1}^n \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Таким образом:

$$\begin{array}{ll} s_1 = 1 & s_6 = \frac{49}{20} = 2,45 \\ s_2 = \frac{3}{2} = 1,5 & s_7 = \frac{363}{140} \approx 2,593 \\ s_3 = \frac{11}{6} \approx 1,833 & s_8 = \frac{761}{280} \approx 2,718 \\ s_4 = \frac{25}{12} \approx 2,083 & s_{10^3} \approx 7,484 \\ s_5 = \frac{137}{60} \approx 2,283 & s_{10^6} \approx 14,393 \end{array}$$

Как отмечалось выше, из закона Хипса следует, что текст из 10 тысяч слов будет содержать всего 100 уникальных слов (они же частотные слова Кластера), и соответственно для текста длиной один миллион слов Кластер будет содержать всего одну тысячу уникальных частотных слов. При этом для Кластера из одной тысячи слов полная частота встречаемости в единицах частотности первого слова составит 7,484 единиц (см. расчеты 1), из которых частота встречаемости первого частотного слова составляет 1 единицу или 13,36%, частота второго слова - около 7%, третьего - 3,6% ... от общей частотности слов в Кластере в единицах частотности первого слова. Как видим, первая сотня частотных слов Кластера в %-ном отношении будет определяющей для Кластеров из 1 тысячи (4 страницы текста) или даже из 1 миллиона слов (4 тысячи страниц текста).

Утверждение 1.

Таким образом, текст объемом от 10 до 250 тысяч слов может быть описан Кластером, размер которого не более, скажем, 100-500 частотных слов.

2.3.4. Порождение и векторное представление Кластеров объекта в РИ.

Векторное представление слов было предложено довольно давно, но, как видно из публикации Кристофер Олах [Christopher Olah, 2014, (<http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>), исследования взаимной встречаемости слов проводились с использованием нейронных сетей, а не Рекурсивного Индекса. "Использование векторных представлений слов в последнее время стало главным "секретом фирмы" во многих системах обработки естественного языка, решающих в том числе задачи выделения именованных сущностей (named entity recognition), частеречной разметки (part-of-speech tagging), синтаксического анализа и определения семантических ролей (semantic role labeling)." - отмечается в другой публикации [Luong et al. (2013) https://nlp.stanford.edu/~lmthang/data/papers/conll13_morpho.pdf].

Приведенные в названных публикациях методы исследования взаимной встречаемости слов с построением векторов взаимной встречаемости, кажутся трудоемкими и не очевидными, в то время как Рекурсивный Индекс предлагает существенно менее трудоемкий и интуитивно понятный способ, чем избран авторами названных публикаций.

При использовании цифровых методов обработки Кластеры и объекты может быть удобно представлять в виде векторов. Рассмотрим процесс создания Кластеров для слов языка по корпусу документов, хранимых в РИ. Формулировка задачи поиска:

1) на вход РИ на каждом цикле подается ключевое слово - образец, по которому машина должна найти документы (последовательности) содержащие такой образец - ключевое слово. Здесь использована аналогия с поисковыми запросами к поисковым машинам, где образец называется "ключевым словом" или "набором ключевых слов". В каждом документе слово имеет порядковый номер N , присвоенный РИ при индексации соответствующего документа.

2) Для введенного ключевого слова надо построить сферу радиуса R слов из ранее проиндексированных последовательностей, проходящих через ключевое слово при каждом его появлении. Для этого следует извлечь из памяти РИ массив частотных слов (связей) - слова, характеризующиеся совместной

встречаемостью с ключевым словом внутри сферы радиуса R слов.

Приступим к построению массива связей ключевого слова во всех ранее проиндексированных документах. Для этого мы станем:

- 1) найти в документах все фрагменты, содержащие такое ключевое слово;
- 2) изменять нумерацию слов в фрагментах так, чтобы ключевое слово в каждом из фрагментов имело номер N, слова фрагмента слева от ключевого слова (прошлое) соответственно имели номера (N-1), (N-2), (N-3) и так далее (N-R), а слова справа (будущее) от ключевого слова имели бы соответственно номера (N+1), (N+2), (N+3) и так далее (N+R).

Таким образом, цепочки слов каждого из фрагментов будут содержать в центре ключевое слово и потому документы будут как бы "проходить" через ключевое слово, образуя шар радиусом R с центром в ключевом слове.

Пересчитав число появлений каждого уникального частотного слова, попавшего в шар со знаком "плюс" (полусфера будущего), а также попавшего в шар со знаком "минус" (полусфера прошлого) мы получим два множества уникальных частотных объектов, умноженный на вес совместной встречаемости уникальных частотных слов шара с ключевым словом поискового запроса (формула 1):

$$K_P = - \sum_{j=1}^R (w_j * C_j) - \text{Кластер прошлого объекта N последовательности.}$$

$$K_F = \sum_{i=1}^R (w_i * C_i) - \text{Кластер будущего объекта N последовательности.}$$

$$K_N = \sum_{i=1}^R (w_i * C_i) - \sum_{j=1}^R (w_j * C_j) - \text{полная сфера (Кластер будущего и Кластер прошлого) объекта N последовательности.}$$

где C_i - частотные слова в Кластере будущего (+) и C_j - в Кластере прошлого (-) соответственно, а w_i и w_j - весовые коэффициенты совместной встречаемости ключевого объекта C_N , породившего Кластер K_N , с соответствующим частотным объектом C_j или C_i Кластера. Коэффициенты w_i могут, например, быть равны суммарной частоте встречаемости объекта C_i с объектом C_N в корпусе последовательностей РИ.

Кластер K_N является массивом частотных объектов C_i , каждый из которых умножен на число w_i появлений объекта C_i в Кластере K_N . Если предположить, что частотные объекты C_i являются единичными векторами, образующими оси декартовой системы координат, то весовые коэффициенты w_i являются значением проекции вектора K_P или K_F на оси координат. Например, если в Кластере K_F объекта C_N ="тигр" дважды встретился объект C_1 ="джунгли" и один раз объект C_2 ="рыжий", то осями в ней будут служить единичные вектора слов C_1 и C_2 , а проекциями вектора объекта C_N на ось C_1 будет $w_1=2$ и на ось C_2 будет $w_2=1$.

Определение 1.

Кластер K_N ключевого объекта C_N является разложением вектора ключевого объекта по координатным осям множества частотных объектов C_i Кластера K_N , а весовые коэффициенты w_i являются проекциями вектора C_N на оси C_i .

2.3.5. Сравнение Кластеров.

2.3.5.1. Коллинеарность.

Утверждение 2.

Коллинеарность векторов слов C_N и C_K , представленных координатами K_N и K_K указывает на то, что слова обладают тождественным смыслом - являются параллельными объектами: или словоформами одного слова, или синонимами, или переводами текста на разные языки или описанием одного и того же явления разными словами [<http://colah.github.io/posts/2014-07-NLP-RXNs-Representations/>].

Длину каждого из коллинеарных векторов можно выразить через длину меньшего из векторов умноженную на величину " λ ".

$$\overrightarrow{C_N} = \lambda * \overrightarrow{C_0}$$

Или в терминах Кластеров (формула 2 - коллинеарные вектора объектов одинакового смысла):

$$\overrightarrow{K_N} = \lambda * \overrightarrow{K_0}$$

2.3.5.2. Сравнение длины нормализованных векторов.

Кластер является вектором в пространстве частотных объектов. Нормализуем веса частотных объектов Кластера так, чтобы их сумма была равна единице. Если вес каждого объекта обозначить как w_i , то сумма всех весов частотных объектов Кластера будет (формула 3 - : суммарный вес объектов Кластера)

$$W_\Sigma = \sum_{i=1}^n w_i$$

И тогда для нормализованных весов частотных объектов ω_i получим (формула 4 - нормализация веса частотных объектов Кластера):

$$\omega_i = \frac{w_i}{W_\Sigma} = \frac{w_i}{\sum_{i=1}^n w_i}$$

и очевидно, что

$$\omega_{\Sigma} = \sum_{i=1}^n \omega_i = 1$$

Учитывая, что число частотных объектов в сравниваемых Кластерах может отличаться, необходимо договориться о мере смыслового тождества таких Кластеров разной размерности. Для этого

1. Проекция ω_{1i} и ω_{2i} коллинеарных векторов Кластеров K_1 и K_2 на ось частотного объекта C_i не должна отличаться более чем на $\Delta\omega_i$.

Сумма разницы весов нормализованных коллинеарных векторов $\sum_{i=1}^N |\omega_{\Sigma 1} - \omega_{\Sigma 2}|$ по всему множеству N объектов Памяти Последовательностей не должна превышать некоторой погрешности "коллинеарности" $\Delta\omega_{\Sigma}$ (формула 5 - Погрешность сравнения нормализованных векторов):

2.

$$|\omega_{1i} - \omega_{2i}| \leq \Delta\omega$$

$$\sum_{i=1}^N |\omega_{1i} - \omega_{2i}| \leq \Delta\omega_{\Sigma}$$

Можно сказать и так, что разница нормализованных профилей двух тождественных по смыслу Кластеров не должна превышать некоторой погрешности (формула 6 - Максимальная погрешность совпадения нормализованных профилей весов Кластеров): $\Delta\bar{K}_{\max} \geq \bar{K}_j - \bar{K}_i$.

2.3.5.3. Представление рядом Фурье.

Любому цифровому объекту множества объектов корпуса последовательностей соответствует уникальный цифровой идентификатор C_i , а в Кластере K_N ключевого объекта C_N каждому частотному объекту C_i соответствует вес w_i совместной встречаемости с ключевым объектом C_N , породившим Кластер K_N . Упорядочим идентификаторы объектов C_i в возрастающей последовательности C_0, C_1, C_2, \dots по оси ординат, а по оси абсцисс станем откладывать вес объекта w_0, w_1, w_2, \dots . Тогда Кластер K_N ключевого объекта можно изобразить диаграммой (фиг. 2). Если договориться, что значения C_0, C_1, C_2, \dots , являются гармониками Кластера K_N с амплитудами w_0, w_1, w_2, \dots , то математически такой Кластер K_N частотных слов можно представить рядом Фурье.

Такое представление Кластеров K_N позволяет применить известные числовые методы для анализа смысла последовательностей, объекты которых представлены своими Кластерами:

1. построить вектор смысла и осуществить поиск коллинеарного ему вектора в векторах смысла различных слов, определяя таким образом наиболее близкое по смыслу слово или понятие доступное памяти последовательностей;

2. применить обратное преобразование Фурье для квантования в составляющие его гармоники - коды частотных слов;

3. выявлять синонимы;

4. выявлять новые понятия.

Все приведенные выше рассуждения применимы как для Кластера объектов, так и для суммы Кластера нескольких объектов, что позволяет представлять вектором и сумму Кластеров.

2.3.6. Единство процессов воспоминания и обучения в РИ.

Кластер K_N объекта C_N является "Кластером объекта C_N по всем последовательностям" Рекурсивного Индекса. Для уменьшения времени формирования Кластера K_N , его значение разумно хранить в РИ и обновлять на каждом цикле ввода объекта C_N в РИ. Пополнение Кластера K_N объектами новых последовательностей (обучение РИ) и внесение изменений в весовые коэффициенты w_i частотных объектов C_i Кластера K_N является процессом обучения Рекурсивного Индекса по объекту C_N . Поскольку для обучения Кластера K_N его приходится извлекать из Рекурсивного Индекса, то можно говорить, что каждый ввод объекта C_N новой последовательности приводит сначала к "воспоминанию" Кластера K_N , а затем к обучению Кластера K_N на примере использования объекта C_N во вводимой последовательности. Более конкретно ввод объекта C_N сопровождается воспроизведением Кластера K_P и Кластера K_F объекта C_N , что соответствует режиму "воспоминаний" паттерна использования объекта C_N в прошлом K_P и "прогнозированию" возможного поведения последовательности в "будущем" K_F .

2.3.7. Функция ослабления весов объектов в Окне Внимания.

ИНИ содержит Сумматор с функцией активации Сумматора, множество Сенсоров группы А, каждый из которых оснащен функцией активации и ячейкой памяти для размещения Соответствующего значения веса А и размещен на выходе одной из шин Устройства ПП уровня иерархии N, а также множество Сенсоров D, каждый из которых оснащен ячейкой памяти и устройством измерения и изменения по меньшей мере одной из характеристик сигнала и размещен на входах одной из шин ПП уровня иерархии N; причем каждый из Сенсоров группы D связан с выходом Сумматора, а каждый из Сенсоров группы А связан с входом Сумматора, кроме того выход Сумматора оснащен связью со входом одной из шин Устройства ПП верхнего уровня иерархии (N+1); режим обучения ИНИ осуществляют циклами, причем на каждом цикле упорядоченное множество из одного или более сигналов обучения (далее "Окно Внимания") подаются на входы одной или более шин ПП уровня иерархии N, причем сигналы в Окне Внима-

ния упорядочены с использованием функции ослабления. Каждый из сигналов проходит через один или более ИНВ размещенных в уровне иерархии N ПП и названные один или более ИНВ изменяет одну из характеристик сигнала кодирующую вес совместной встречаемости и на выходе каждой из множества шин ПП уровня иерархии N получают сигнал, кодирующий вес совместной встречаемости, из которого извлекается значение веса совместной встречаемости соответствующей шины, и вес передается в Сумматор, где веса, полученные с выходов разных шин, складываются и запоминается значение суммы цикла, после чего Окно Внимания изменяется и цикл обучения повторяется, причем на каждом очередном цикле обучения значение суммы очередного цикла сравнивается со значением суммы предыдущего цикла, и, если значение суммы очередного цикла обучения равно или меньше значения суммы предыдущего цикла обучения, обучение ИНИ останавливается и каждым сенсором группы A названное Соответствующее значение веса A (далее "вес активации"), полученное для цикла обучения с максимальной суммой весов присваивается в качестве значения активации функции активации сенсора A, Сумматор присваивает функции активации Сумматора значение максимальной суммы весов или присваивает значение числа сенсоров группы A с ненулевыми значениями Соответствующих весов A или присваивает оба названных значения, а каждый сенсор ИНИ группы D на входе каждой из шин ПП уровня иерархии N, на которые были поданы сигналы во время цикла обучения с максимальной суммой весов, измеряет и размещает в ячейке памяти Сенсора D Соответствующее значение D по меньшей мере одной из характеристик сигнала обучения, кодирующей названное значение функции ослабления D сигнала шины в Окне Внимания; в режиме воспроизведения ИНИ сигнал воспроизведения подают на одну или множество шин ПП уровня иерархии N и получают на выходе множества шин ПП уровня иерархии N веса совместной встречаемости и, если вес совместной встречаемости, полученный на выходе шины, равен или превышает значение функции активации сенсора A такой шины, сенсор A посылает в Сумматор или значение веса активации или единичное значение или оба значения, а Сумматор суммирует полученные значения функций активации сенсоров A и сравнивает полученную сумму со значением суммы активации Сумматора и, если суммарное значение равно или превышает значение функции активации Сумматора, то сигнал активации ИНИ подается на выход Сумматора, который затем одновременно поступает на вход одной из шин уровня иерархии (N+1) ПП и на входы сенсоров группы D уровня иерархии N ПП, ячейка памяти каждого из которых содержит названное Соответствующее значение функции ослабления D, причем каждый из названных сенсоров группы D изменяет сигнал Сумматора в соответствии с Соответствующим значением функции ослабления D и подает измененный сигнал на вход соответствующей шины ПП уровня иерархии N или не изменяет сигнала и подает неизмененный сигнал на вход соответствующей шины ПП уровня иерархии N.

Один из одновременно поданных на каждую из шин сигналов изменяется с тем, чтобы разница в сигналах указывала направление от шины с большим номером к шине с меньшим номером или наоборот от шины с меньшим номером к шине с большим номером, а каждый ИНВ снабжен не одним, а двумя Счетчиками, один для изменения последнего значения встречаемости в направлении от шины с большим номером к шине с меньшим номером и второй для изменения последнего значения встречаемости в направлении от шины с меньшим номером к шине с большим номером.

Как говорилось выше весовой коэффициент w_i является весом совместной встречаемости ключевого объекта C_N с частотным объектом C_i Кластера на всем корпусе последовательностей. Однако в каждой конкретной последовательности эти два объекта могут быть разделены разным числом других объектов g и очевидно, что каждому случаю взаимной встречаемости должен соответствовать разный вес связи, который будет тем меньше, чем большим числом объектов g разделены объекты C_N и $C_{N \pm r}$ в конкретной последовательности. Если ослабление связи изобразить как снижение насыщенности цвета объекта, то последовательный ввод последовательности объектов, последним из которых был введен объект C_0 будет выглядеть так, как показано на фиг. 3.

Снижение веса связи (синапса) вычисляется для каждого случая совместной встречаемости отдельно. Разумно ослаблять вес конкретного случая взаимной встречаемости с ростом расстояния r между объектами C_1 и C_i : чем больше расстояние между объектами, тем слабее между ними связь, которую мы станем выражать показателем веса связи w^r , где r - ранг связи. В общем случае функция ослабления веса объекта C_i в последовательности, показанной на фиг. 3, может быть любой (формула 7 - функция ослабления Окна Внимания): $w^r = f(r)$, или $w = f(r)$

А для появления произвольного частотного объекта C_i во всех последовательностях полусферы (Кластера) ключевого объекта C_N функцию следует записать как (формула 8 - суммарный вес объекта в Кластере):

$$w_i = \sum_{s=1}^S f_i^s(r)$$

где S - число последовательностей, вошедших в полусферу (Кластер) ключевого объекта C_i . Кластер ключевого объекта C_N , построенный на последовательностях числом S (формула 9):

$$K_N = \sum_{i=1}^I w_i * C_i = \sum_{i=1}^I C_i * \left(\sum_{s=1}^S f_i^s(r) \right)$$

2.3.8. Функция нумерации объектов Окна Внимания.

Как показано выше, функция ослабления веса $f(r)$ позволяет присвоить весовые коэффициенты w объектам каждой последовательности сферы R и суммировать их для определения совокупного веса такого частотного объекта Кластера будущего или прошлого. Поскольку в разных решениях и разными исследователями могут применяться разные функции, то далее будем говорить просто о функции ослабления веса $f(r)$ или о весе w объекта (формула 10 - функция ослабления).

$$w = f(r)$$

Функция ослабления может применяться к ослаблению любой физической характеристики - частоты, силы, напряженности или напряжения и так далее. Например, если функция $f(r)$ применена к частоте колебаний сигнала, то мы получим частотное разделение и сможем определять ранг частотного объекта относительно ключевого по частоте сигнала частотного объекта.

Очевидно, что значение r может быть определено функцией нумерации (формула 11 - функция нумерации объектов Окна Внимания)

$$r = g(w)$$

Замечание 2 (функция нумерации и весовая функция Окна Внимания): Функция расстановки или нумерации частотных объектов в Окне Внимания $r=g(w)$, является функцией, обратной весовой функции $w=f(r)$.

Функция ослабления может быть линейной, как например $w = f(r) = \left(1 - \frac{r}{R}\right)$, а для ускоренного снижения веса может использоваться распределение Паретто $w=f(r)=1-\left(\frac{r}{R}\right)^k$ или закон Ципфа $w = f(r) = \frac{1}{r}$, или квадратичную функцию $w = f(r) = \frac{1}{r^2}$ или $w = f(r) = \frac{1}{2^r}$ или экспоненциальную функцию $w = f(r) = \frac{1}{e^r}$ и так далее. С практической точки зрения имеет смысл выбрать функцию, для которой для каждого $0 < r \leq R$ выполняется условие (формула 12):

$$f(r) > \sum_{j=r+1}^R f(j)$$

Это позволяет выделить объекты разного ранга по их весу.

Тем не менее, формула 12 не учитывает тот факт, что в Кластере каждый из частотных объектов C_i может встречаться на расстоянии r от ключевого объекта C_N число раз меньше или равное S , где S - число последовательностей, попавших в сферу ключевого слова C_N (см. формула 8). Учитывая это обстоятельство неравенство (формула 12) можно переписать в виде (формула 13):

$$f_i(r) > \sum_{j=r+1}^R \sum_{s=1}^S f_i^s(r)$$

Формула 13 позволяет ранжировать частотные объекты в Кластере, считая суммарный вес каждого частотного объекта вероятностью его появления на расстоянии $r = g(f_i(r))$.

2.3.9. Кластеризация последовательности.

Кластер объекта является инвариантным представлением объекта (вообще говоря, обратное утверждение не корректно - одному Кластеру может соответствовать несколько объектов с одинаковым смыслом, в частности для языка, из-за наличия словоформ и синонимии. В общем случае из-за наличия параллельных объектов и последовательностей) и для анализа гипотез появления следующих или предыдущих объектов последовательностей, вместо последовательности объектов удобно оперировать последовательностью порожденных ими Кластеров, поэтому функция ослабления веса $f(r)$ применяется не только к объектам, но и к их Кластерам. Поэтому также будем говорить о функции ослабления веса $f(r)$ или о весе w_i Кластера K_i для объекта C_i .

На фиг. 4 приведен пример последовательности объектов и Кластеров, порожденных этими объектами. По мере удаления от последнего введенного объекта последовательности, насыщенность цвета Кластеров меняется вместе с изменением насыщенности цвета объектов, породивших эти Кластеры. Как и в случае с последовательностью объектов (см. фиг. 3) градиентом цвета Кластеров мы показали, что прочность связи частотных объектов одного Кластера с частотными объектами других Кластеров меняется с увеличением расстояния между ними в соответствии с функцией ослабления $f(r)$.

Из Множества Трубы извлекают и складывают весовые коэффициенты встречаемости всех частотных Объектов, получая Суммарный Вес Трубы.

Суммируя Кластеры (формула 9) с учетом их ослабления в последовательности получим (формула 14 - сумма Кластеров объектов Окна Внимания)

$$K^{\Sigma} = \sum_{r=1}^R f(r) * K_r = \sum_{r=1}^R f(r) * \left\{ \sum_{i=1}^1 w_i * C_i \right\} = \sum_{r=1}^R f(r) * \left\{ \sum_{i=1}^1 C_i * \left(\sum_{s=1}^S f_i^s(r) \right) \right\}$$

Очевидно, что:

$$K^{\Sigma} \propto (f(r))^2$$

Утверждение 3.

Учитывая условие (формула 13) можно утверждать, что веса частотных объектов, чей ранг больше единицы $r > 1$, окажутся такого порядка малости, что ими можно пренебречь и потому при расчете K^{Σ} вместо полных Кластеров K_r можно складывать только Кластеры первого ранга. K_r^1 (формула 15 - сумма Кластеров последовательности):

$$K^{\Sigma} = \sum_{r=1}^R f(r) * K_r^1$$

При построении Полного Кластера объекта для сферы радиуса R , мы помещали этот объект в центр сферы радиуса R и вырезали из всех последовательностей куски длиной $(+R)$ до объекта и $(-R)$ после него затем размещая эти куски в сфере. Тем не менее любую последовательность объектов можно представить последовательностью связей 1-го ранга между соседними объектами, что соответствует Кластерам 1-го ранга (фиг. 5).

Массив "будущего" или массив "прошлого", или ранговое множество ранга, отличного от множества базового ранга, представляют множеством, производным от множества МСП.

Названное ранговое множество является множеством первого ранга и содержит весовые коэффициенты частотных Объектов, непосредственно соседствующих с названным ключевым Объектом в названных последовательностях.

Утверждение 4.

Любой Полный Кластер для сферы радиуса R является линейной композицией Кластеров 1-го ранга множества всех уникальных объектов и (формула 15) является корректной.

Утверждение 5.

Однако и Кластер произвольного ранга N также является производной Кластера первого ранга и потому любой полный Кластер для сферы радиуса R может быть также представлен как линейная композиция Кластеров ранга N множества всех уникальных объектов.

Еще одним важным свойством нумерованной памяти последовательностей является симметричность весов, а именно то, что вес связи будущего $w_{N \rightarrow i}$ равен весу связи прошлого $w_{i \rightarrow N}$. Поэтому для каждого уникального объекта C_N память последовательностей может хранить только один ранговый Кластер будущего (или прошлого), а Кластер прошлого (или будущего) можно синтезировать как линейную композицию всех связей объекта C_N в Кластерах будущего K_i (или прошлого) всех других уникальных объектов C_i памяти последовательностей. Продемонстрируем как это сделать и для этого предположим, что все объекты памяти последовательности пронумерованы от 1 до Max и для каждого объекта память хранит Кластер будущего. Задача: построить Кластер прошлого K_N для некоторого объекта C_N памяти последовательностей:

Шаг 1) Принимаем $i=1$.

Шаг 2) Извлекаем из Кластера K_i вес $w_{i \rightarrow N}$ связи $C_i \rightarrow C_N$ ключевого объекта C_i с частотным объектом Кластера C_N .

Шаг 3) Принимаем, что извлеченный вес $w_{i \rightarrow N}$ связи будущего $C_i \rightarrow C_N$ соответствует весу $w_{N \rightarrow i}$ связи прошлого $C_N \rightarrow C_i$, между объектом C_N и объектом C_i .

Шаг 4) Добавляем вес $w_{N \rightarrow i}$ связи объектов $C_N \rightarrow C_i$ в Кластер прошлого K_N объекта C_N .

Шаг 5) Принимаем $i=i+1$ и если $i \leq Max$, то переходим на шаг 1, а если $i > Max$, то Кластер прошлого K_N объекта C_N сформирован.

Утверждение 6.

В памяти нумерованных последовательностей для каждого уникального объекта C_N достаточно хранить один Кластер будущего (или прошлого) K_N , а Кластер прошлого (или будущего) K_N можно построить как линейную композицию всех Кластеров будущего (или прошлого) K_i памяти последовательностей.

2.3.10. Порождение вопросов.

В памяти сохраняют некоторое множество всех ранговых множеств базового ранга в качестве эталонного (далее "Эталонное Состояние Памяти" или "ЭСП"), а любое "Мгновенное состояние памяти" (далее "МСП") или его часть сравнивают с ЭСП или его частью для выявления отклонений МСП от ЭСП.

Память Последовательностей оперирует конечным числом уникальных объектов и полное множество весов совместной встречаемости каждого уникального объекта C_N с каждым другим уникальным объектом C_K в каждый момент времени характеризует состояние Памяти Последовательностей (далее "Состояние памяти" или "Состояние создания"). Если каждый из объектов представить Кластером 1-го ранга "будущего", то линейная композиция Кластеров 1-го рода будущего всех уникальных объектов

морковку и мне нужно вспомнить, что вчера я просил вас привезти мне морковку для моего домашнего кролика. Однако, если я не вспомню о своей просьбе принести морковку, то могу счесть ваше поведение неадекватным - не соответствующим нормальному устойчивому статистическому состоянию сознания. Проводя параллель с сознанием, устойчивое статистическое состояние можно назвать "нормальным состоянием сознания", а отклонение от такого устойчивого статистического состояния можно охарактеризовать как "состояние измененного сознания".

Утверждение 7.

Из формулы "Состояния сознания" (формула 16) следует, что каждый Кластер 1-го ранга является подмножеством матрицы "состояния сознания" K_{state} и значит любое состояние Памяти Последовательностей можно описать подмножеством элементов матрицы K_{state} .

Поэтому любой массив "будущего" или массив "прошлого", или ранговое множество ранга, отличного от множества базового ранга, представляют множеством, производным от множества МСП.

2.3.11. Анализ "Состояния памяти".

Анализ Состояния памяти можно проводить методами, которые описаны в настоящей работе, но для анализа можно применить также и технологию нейросетей, в частности сверточных нейросетей. Для этого нейросеть следует обучать на различных "Состояниях памяти", вводя веса взаимной встречаемости размещенные в матрице Состояния памяти (формула 17) в качестве исходных данных или "карты признаков" во время обучения и использования такой нейросети.

3. Память нумерованных последовательностей.

Ранее мы рассмотрели реализацию памяти последовательностей в виде Рекурсивного Индекса, где все последовательности пронумерованы, и каждая имеет свой уникальный идентификатор, и все объекты внутри последовательности также пронумерованы. Нумерация снимает ограничения на процесс "воспоминания" всей последовательности и любого ее фрагмента, так как каждую последовательность можно восстановить по идентификатору последовательности, а любой объект последовательности по его номеру. Однако память человека не нумерует последовательности и, тем не менее, умеет их хранить и извлекать. Как она это делает?

3.1. Особенности нумерованных последовательностей.

Поясним, как работает память нумерованных последовательностей на примере с сетью дорог со светофорами. Представим, что машины движутся по пересекающейся сети дорог с перекрестками оборудованными светофорами, а маршруты движения машин задаются последовательностью светофоров, которые должна проехать машина, однако полные маршруты никому не известны, а светофор конкретной дороги на перекрестке загорается зеленым светом только в случае, если три перекрестка, которые машина проехала последними подтвердят, что машина их проехала в определенном порядке. Поэтому загореться зеленым для машины может, в большинстве случаев, только один светофор каждого перекрестка, в зависимости от того, какие три перекрестка перед этим проехала машина. Таким образом, система светофоров не зная маршрутов, знает как управлять движением по этим маршрутам, основываясь на том, что каждому светофору известны три последовательных предыдущих перекрестка, которые должна проехать машина, чтобы иметь право на выезд с перекрестка под этот светофор.

3.1.1. Связность хитов Рекурсивного Индекса.

Несмотря на то что Рекурсивный Индекс является нумерованным, он обладает потенциалом, который мы станем использовать для отказа от нумерации и перехода к Памяти Нумерованных Последовательностей.

Рассматривая хиты РИ можно сказать, что это частично пересекающиеся фрагменты последовательности, причем уникальным объектом хита является ключевой объект, а объектами пересечения хита со предыдущим и следующим хитами являются "предыдущий" и "следующий" объекты, размещенные в хите ключевого объекта, которые представляют собой соответственно обратную и упреждающую связи ключевого объекта с предыдущим и последующим хитами последовательности. Именно это свойство Рекурсивного Индекса - частичное пересечение хитов, мы и станем использовать для отказа от нумерации, заменив нумерацию механизмом сравнения множество хитов на предмет их частичного совпадения, что должно указывать на связь между ними. Таким образом, от детерминированного механизма поиска в РИ мы перейдем к вероятностному поиску в Нумерованной Памяти Последовательностей.

На момент создания любого хита запись в него "следующего" объекта последовательности невозможна, пока не введен такой следующий объект последовательности. Таким образом, запись в хит "следующего" объекта возможна лишь на шаге ввода такого "следующего" объекта и записи в РИ следующего хита. То есть Рекурсивному Индексу необходима обратная связь с предыдущим хитом, по которой на очередном шаге индексации предыдущему хиту будет сообщено, каким оказался новый объект последовательности (фиг. 6).

Как видно на фиг. 6, связь, которая в процессе запоминания (индексирования) последовательности является обратной, при воспоминании (извлечении последовательности из индекса) используется как упреждающая связь, что позволяет вспомнить "следующий" объект последовательности.

3.1.2. Отказ от нумерации объектов последовательности.

Указанный технический результат для объекта "Способ создания и функционирования ПП" дости-

гается за счет того, что в указанном способе, где цифровая информация представлена множеством машиночитаемых массивов данных, каждый из которых является последовательностью множества уникальных Объектов, а каждый из названных Объектов представлен уникальным машиночитаемым значением Объекта, причем каждый Объект (далее "ключевой Объект") появляется, по меньшей мере, в некоторых последовательностях, Память Последовательностей обучают, подавая последовательности Объектов на вход памяти, а память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, предшествующие названному ключевому Объекту в названной последовательности (далее "частотные Объекты прошлого"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Прошлого", а также память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, следующие за названным ключевым Объектом в названной последовательности (далее "частотные Объекты будущего"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Будущего"; множество объектов каждого из производных массивов данных "Прошлое" и "Будущее" разбивают на подмножества (далее "ранговые множества"), каждое из которых содержит только частотные Объекты, равноотстоящие от названного ключевого Объекта или в "Прошлом", или в "Будущем", а каждому уникальному ключевому Объекту ставят во взаимное соответствие и сохраняют в ПП сам названный ключевой Объект и по меньшей мере одно из названных ранговых множеств названного уникального ключевого Объекта, содержащее, по меньшей мере, значение счетчика взаимной встречаемости названного уникального ключевого Объекта с каждым уникальным частотным Объектом, а также делают доступным поиск названного рангового множества весовых коэффициентов по введенному названному уникальному ключевому Объекту или поиск названного уникального ключевого Объекта по названному ранговому множеству или его части.

Используя обратные и упреждающие связи можно отказаться от нумерации объектов последовательности и создать память последовательностей в базе данных, в которой хит будет содержать фрагмент - очередь из N последовательных объектов последовательности. Ранее, пользуясь нейроаналогией, мы назвали такую очередь Окном Внимания [2.2.7.], однако для нумерованной памяти последовательностей Окно Внимания является нумерованным отрезком, в котором порядок объектов очереди задан обратнo-упреждающими связями. Сдвигая очередь на по меньшей мере один объект $\Delta=1$ вперед мы получаем новых хит - Окно Внимания, в котором отсутствует самый ранний объект предыдущего хита и добавлен новый самый поздний объект. Таким образом каждый следующий хит $(k+1)$ будет содержать фрагмент предыдущего хита k , а хиты k и $(k+2)$ разделенные хитом $(k+1)$ будут отличаться на четыре объекта - пару самых ранних объектов и пару самых поздних. В общем случае два хита k и $(k+n)$, разделенные другими хитами будут содержать общий фрагмент длиной h :

$$h = N - 2 * (n + \Delta)$$

Если мы хотим, чтобы n последовательных хитов содержали общий фрагмент заданной длины h , то длина хранимого в хите фрагмента (Окна Внимания) должна быть равна:

$$N = h + 2 * (n + \Delta)$$

Для $\Delta=1$, $n=3$ и $h=3$ получим $N=3+2 \times (3+1)=11$. Полученный результат демонстрирует минимальную длину фрагмента, который следует хранить в хите, чтобы три последовательных хита содержали общий фрагмент длиной 3 объекта. Повышая число совпадающих объектов (при условии, что $N > (2 \times n)$), мы снижаем вероятность ошибки при поиске связанных хитов обратно пропорционально числу сочетаний из

N по n : $\frac{1}{C_n^N} = \frac{n!(N-n)!}{N!}$, где N является числом всех уникальных объектов на множестве которых строится память последовательностей.

Нейроны образуют между собой физические обратнo-упреждающие связи. В индексе нумерованных последовательностей функцию физической связи нейронов выполняют процессы поиска и сравнения хитов, что и создает обратнo-упреждающую связь между хитами, хранящими одинаковых фрагмент последовательности. Понятно, что такой же фрагмент может оказаться частью хита, не относящегося к нужной последовательности и потому, как уже отмечалось выше, процесс вспоминания в памяти нумерованных последовательностей, будет носить не детерминированный, а вероятностный характер.

Хит Ненумерованной Памяти Последовательностей помимо названных фрагментов "предыдущих" и "следующих" объектов также может содержать другие данные:

Хит={объекты прошлого, объект хита, объекты будущего, другие данные}

Проиллюстрируем это на примере последовательности букв латинского алфавита (A, B, C, D, E, F, G,...) хиты объектов B и C будут содержать одинаковые фрагменты {B, C, D} (формула 18 - формула хи-

та нумерованной последовательности):

$$\text{хит}_B = \{ _ , A, B, C, D \}$$

$$\text{хит}_C = \{ A, B, C, D, E \}$$

где символом пробела " _ " обозначена пустая обратная связь начала последовательности.

Как видим в приведенных выше хитах совпадает фрагмент последовательности {B, C, D}, что позволяет Рекурсивному Индексу принять решение о принадлежности обоих хитов к одной последовательности (см. фиг. 7) и позволяет предсказать появление следующего объекта последовательности - буквы E, основываясь на ее появлении в "хите_C". Очевидно также, что построение гипотезы сводится к поиску представленных хитами последовательных фрагментов последовательности, пересечение которых является непустым множеством (формула 19):

$$\text{хит}_B \cap \text{хит}_C = \{ B, C, D \}$$

и последующему поиску гипотезы E, как дополнения $\Delta(\text{хит}_B)$ множества объектов "хита_B" до множества объектов "хита_C" (формула 20):

$$E = \Delta(\text{хит}_B) = (\text{хит}_C) \setminus (\text{хит}_B)$$

Как видим (см. фиг. 7), работа памяти нумерованных последовательностей дает не точный, а вероятностный результат при извлечении последовательностей, по сути, Рекурсивный Индекс Нумерованных Последовательностей (РИНП) является ассоциативной памятью.

Нетрудно заметить, что если извлечь все объекты прошлого или объекты будущего из всех хитов конкретного уникального объекта во всех последовательностях и объединить извлеченные объекты прошлого или будущего в одно множество, то мы получим Кластер прошлого или Кластер будущего этого уникального объекта, о которых мы говорили выше. Кластер объекта строится по всем последовательностям, содержащим объект, для которого строится Кластер, а потому при построении Кластера номер последовательности знать не обязательно.

Разные последовательности могут иметь одинаковые фрагменты, поэтому в нашем примере со светофорами, на перекрестке могут загораться зеленым не один, а несколько светофоров одновременно, но гореть они могут с разной яркостью: все зеленые светофоры ведут на дороги, соответствующие нашему маршруту, но чем ярче зеленый светофор, тем чаще использовалась эта дорога для маршрута, которым вы следуете. Вы видите и красные светофоры, означающие дороги, которые раньше никогда не использовались для маршрута, которым вы следуете и, если вы выберете дорогу с красным светофором, то можете не добраться до цели своего маршрута, а может дорога, обозначенная красным светофором, окажется короче дорог даже с самыми яркими зелеными светофорами - просто эту дорогу с красным светофором никто еще не использовал для маршрута, которым вы следуете.

Несмотря на различие в названиях и упреждающая и обратная связи являются одной и той же связью объектов. Память последовательностей (см. фиг. 8) в процессе индексирования (запоминания или обучения) создает обратные связи, которые используются как связи с упреждением в процессе извлечения последовательностей (воспоминания или прогнозирования).

Как видно, для нумерованных последовательностей механизм построения гипотез является единственным процессом, обеспечивающим извлечение последовательностей из памяти.

3.2. Прогнозирование.

Далее станем различать два типа предсказаний и прогнозов: предсказание/прогнозирование будущего и восстановление/реконструкция прошлого, а также коррекцию ошибок ввода.

Определение 2.

Прогнозирование будущего станем называть "задачей ученого", который используя последовательность известных объектов/событий пытается предсказать следующее объект/событие и последовательность будущих событий, а задачу восстановления прошлого станем называть "задачей следопыта", который пытается определить объекты/события и последовательность событий предшествующие цепи известных объектов/событий. Можно также сказать, что задача ученого сводится к предсказанию следствий по известным причинам, а задача следопыта сводится к определению причин по известным следствиям.

3.2.1. Общий подход к прогнозированию.

Из предыдущих рассуждений следует, что каждый объект последовательности, записанный в память нумерованных последовательностей, имеет обратные и упреждающие связи со всеми объектами, записанными в память последовательностей в прошлом и будущем этих последовательностей на глубину R объектов, где R представляет радиус сферы прошлого/будущего. Это позволяет "увидеть" гипотезы неизвестных объектов 5, 6 и 7 в Кластерах будущего, построенных для известных объектов последовательности 2, 3 и 4 (см. фиг. 9).

Вместе с тем, число гипотез экспоненциально растет с ростом глубины прогнозирования (см. фиг. 10) тем самым снижая вероятность реализации более глубоких гипотез.

Несмотря на то что рассмотрено прогнозирование для "задачи ученого", понятно, что при решении "задачи следопыта" число гипотез также будет расти с ростом глубины прогнозирования в прошлое.

3.2.2. Ранги частотных объектов и ранги Кластеров.

В главах, посвященных нумерованным последовательностям, мы описывали Кластер ключевого объекта, в который мы записывали все частотные объекты, входящие в шар радиуса R . Объяснялось это тем, что внутри шара находились объекты, чьи номера в последовательностях были известны. Однако для анализа нумерованных последовательностей удобнее рассматривать множество Кластеров, в каждый из которых войдут только частотные объекты с одинаковым рангом, лежащие на поверхности сферы радиуса r с центром в ключевом объекте, где $1 \leq r \leq R$ для полусферы будущего и $-R \leq r \leq -1$ для полусферы прошлого. Таким образом, мы получим множество Кластеров прошлого $K_{-R}, \dots, K_{-r}, \dots, K_{-1}$ и множество Кластеров будущего $K_1, \dots, K_r, \dots, K_R$, где нижний индекс r - это ранг соответствующего Кластера. Ранг Кластера определяется рангом частотных объектов соответствующего ранга, которые в него входят. На фиг. 11 показан Ключевой Объект (КО), а также по три частотных объекта (-3, -2, и -1), предшествующих Ключевому в конкретной последовательности и три (1, 2 и 3), расположенные после Ключевого Объекта. Частотные объекты первого ранга (-1 и 1) - это объекты, непосредственно связанные с ключевым объектом в последовательностях, записанных в РИ. Частотные объекты второго ранга (-2 и 2) - это объекты, отделенные от ключевого объекта объектом первого ранга, и объекты третьего ранга (-3 и 3) - это объекты, отделенные от Ключевого Объекта объектами первого и второго рангов (-2, -1, 1 и 2) и так далее. Понятно, что в Кластер первого ранга входят частотные объекты первого ранга, в Кластер второго ранга входят частотные объекты второго ранга и так далее.

3.2.3. Техника извлечения из памяти нумерованных последовательностей.

Проиллюстрируем технику ранговых Кластеров, которая позволяет извлекать нумерованные последовательности из памяти последовательностей. Для того чтобы техника работала, необходимо в процессе запоминания последовательностей формировать для каждого уникального объекта памяти ранговые Кластеры его встречаемости с другими уникальными объектами последовательностей, размещенных в памяти нумерованных последовательностей. Примем, что Окно Внимания равно 7, так что для каждого уникального объекта "N" в процессе обучения памяти последовательностей сформированы шесть ранговых Кластеров - три для прошлого K^{-3}, K^{-2}, K^{-1} и три для будущего K^1, K^2, K^3 (фиг. 12).

Предположим введенный фрагмент последовательности состоит из трех объектов $\{1, 2, 3\}$ (фиг. 13), последний введенный из которых обозначен цифрой "3", а найти нам надо объекты "4", "5", "6", являющиеся возможным продолжением предъявленного нам фрагмента.

Из постановки задачи следует, что для последнего введенного объекта нам известны три ранговых Кластера K^1_3, K^2_3, K^3_3 . Нижний индекс в обозначении рангового Кластера K^1_3 означает объект "3", для которого построен ранговый Кластер, а ранг Кластера "1", "2" и "3" указан в верхнем индексе. Понятно, что объект "4" является одним из объектов ранговых Кластеров будущего K^1_3, K^2_2, K^3_1 , а в каждом из Ранговых Кластеров прошлого $K^{-3}_4, K^{-2}_4, K^{-1}_4$ самого объекта, должны находиться соответствующие рангу объекты прошлого "3", "2" и "1":

$$\langle 4 \rangle \in K^1_3$$

$$\langle 4 \rangle \in K^2_2$$

$$\langle 4 \rangle \in K^3_1$$

А для поиска в памяти копий вводимой последовательности также должны выполняться условия (формула 21):

$$\langle 3 \rangle \in K^{-1}_4$$

$$\langle 2 \rangle \in K^{-2}_4$$

$$\langle 1 \rangle \in K^{-3}_4$$

Следует заметить, что поскольку последовательности нумерованные, то названным условиям (формула 21) может соответствовать не один, а несколько уникальных объектов памяти и этот случай мы рассмотрим ниже [3.2.5]. Предположим, мы нашли удовлетворяющие названным условиям один или более элементов "4" (фиг. 13). Если найденный объект "4" является продолжением заданного фрагмента, то в его первом ранговом Кластере будущего K^4_1 находится объект "5", являющийся продолжением последовательности, для которого одновременно должны выполняться следующие условия (формула 22 - подтверждение гипотезы):

$$\langle 5 \rangle \in K^1_4$$

$$\langle 5 \rangle \in K^2_3$$

$$\langle 5 \rangle \in K^3_2$$

А также для копий (формула 23 - гипотеза подтверждает наличие копий):

$$\langle 4 \rangle \in K^{-1}_5$$

$$\langle 3 \rangle \in K^{-2}_5$$

$$\langle 2 \rangle \in K^{-3}_5$$

Если ранее в качестве объекта "4" были найдены более одного претендента, то на этапе поиска объ-

екта "5" часть из кандидатов объекта "4" не смогут удовлетворить условиям (формула 22), что позволит сузить число кандидатов на каждой следующей итерации поиска продолжения фрагмента. Для следующего объекта "6", который должен содержаться в ранговом Кластере K^5_1 , должны также выполняться уже известные нам условия:

$$\langle 6 \rangle \in K^1_5$$

$$\langle 6 \rangle \in K^2_4$$

$$\langle 6 \rangle \in K^3_3$$

А также для копий (формула 24):

$$\langle 5 \rangle \in K^{-1}_6$$

$$\langle 4 \rangle \in K^{-2}_6$$

$$\langle 3 \rangle \in K^{-3}_6$$

И снова на этой итерации можно избавиться от претендентов на объект "4" и "5", если они не удовлетворяют условиям (формула 24).

Таким образом, использование обратных и упреждающих связей, ранговых Кластеров и их обратных проекций, позволяет реализовать извлечение последовательностей, размещенных в памяти нумерованных последовательностей.

Техника применения ранговых Кластеров для извлечения последовательностей приведена в качестве примера, чтобы продемонстрировать возможность извлечения последовательностей из памяти нумерованных последовательностей. В то же время, профессионалы могут предложить и другую технику извлечения, основанную на использовании памяти нумерованных последовательностей и использовании Кластеров, в духе подхода, изложенного в настоящей работе.

3.2.4. Весовое условие для копий

Если вводимая последовательность является копией последовательности, ранее размещенной в памяти последовательностей, то вес известных (R-1) объектов ($C_1, C_2, \dots, C_{(R-1)}$) вводимой последовательности в Ранговых Кластерах прошлого, порожденных последним введенным или спрогнозированным объектом C_R , должны удовлетворять условию (формула 25 - Весовое условие наличия копий вводимой последовательности):

$$(w_{(R-1)} * C_{(R-1)}) \in K^{-1}_R \text{ причем } w_{(R-1)} \geq f(1)$$

$$(w_{(R-2)} * C_{(R-2)}) \in K^{-2}_R \text{ причем } w_{(R-2)} \geq f(2)$$

$$(w_1 * C_1) \in K^{R-1}_R \text{ причем } w_1 \geq f(R-1)$$

Если условия (формула 25) выполняются, то объект C_R может быть продолжением последовательности, ранее размещенной в памяти последовательностей.

3.2.5. Полный и Ранговый Кластеры ключевого объекта.

Труба Указанный технический результат для объекта "Способ создания и функционирования ПП" достигается за счет того, что в указанном способе, где цифровая информация представлена множеством машиночитаемых массивов данных, каждый из которых является последовательностью множества уникальных Объектов, а каждый из названных Объектов представлен уникальным машиночитаемым значением Объекта, причем каждый Объект (далее "ключевой Объект") появляется, по меньшей мере, в некоторых последовательностях, Память Последовательностей обучают, подавая последовательности Объектов на вход памяти, а память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, предшествующие названному ключевому Объекту в названной последовательности (далее "частотные Объекты прошлого"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Прошлого", а также память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, следующие за названным ключевым Объектом в названной последовательности (далее "частотные Объекты будущего"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Будущего"; множество объектов каждого из производных массивов данных "Прошлого" и "Будущее" разбивают на подмножества (далее "ранговые множества"), каждое из которых содержит только частотные Объекты, равноотстоящие от названного ключевого Объекта или в "Прошлом", или в "Будущем", а каждому уникальному ключевому Объекту ставят во взаимное соответствие и сохраняют в ПП сам названный ключевой Объект и по меньшей мере одно из названных ранговых множеств названного уникального ключевого Объекта, содержащее, по меньшей мере, значение счетчика взаимной встречаемости названного уникального клю-

чего Объекта с каждым уникальным частотным Объектом, а также делают доступным поиск названного рангового множества весовых коэффициентов по введенному названному уникальному ключевому Объекту или поиск названного уникального ключевого Объекта по названному ранговому множеству или его части.

Продолжая рассуждения предыдущего раздела, рассмотрим численные методы прогнозирования появления объектов.

В общем случае Полный Кластер ключевого объекта C_N будет определяться так (формула 26):

$$K_N = [w_1 * C_1; w_2 * C_2; \dots; w_n * C_n;]$$

Весовые коэффициенты w_i будут суммой всех весов объекта C_i во всех последовательностях корпуса (формулы 27):

$$w_i = \sum_{l=1}^I \sum_{r=1}^R f(r),$$

где I - число появлений объекта C_N в корпусе последовательностей, R - радиус сферы, а функция $f(r)$ определена только для $1 \leq r \leq R$ где появился объект C_i .

Формула 26 описывает встречаемость объекта со всеми другими объектами на корпусе последовательностей, однако для некоторых задач анализа будет важным разделить Полный Кластер объекта на Ранговые Кластеры в каждый из которых войдут только частотные объекты одного ранга $1 \leq r \leq R$ (формула 28 - Кластер ранга z):

$$K_N^r = \left[\sum_{i=1}^k C_1; \sum_{i=1}^l C_2; \dots; \sum_{i=1}^m C_n; \right]$$

где k, l, m - число появления соответственно частотных объектов $C_1; C_2; \dots; C_n$ на расстоянии r от ключевого объекта C_N .

Ранговый кластер показывает вероятность появления конкретных частотных объектов на определенном расстоянии r от ключевого объекта на всем корпусе последовательностей.

Теперь формула 26 Полного Кластера ключевого объекта может быть переписана с использованием Ранговых Кластеров так (формула 29 - Полный Кластер объекта):

$$K_N = \pm \sum_{r=1}^R f(r) * K_N^r$$

Разумеется, Полный и Ранговые Кластеры могут быть построены как для сферы будущего (со знаком плюс), так и для сферы прошлого (со знаком минус).

Так же как Кластер объекта является инвариантным представлением объекта, инвариантным представлением последовательности может служить Кластер последовательности.

Поскольку каждый объект последовательности порождает Кластер, то связь между порождаемыми Кластерами будет ослабевать ("затухать") с увеличением расстояния между ними по закону $f(r)$. Если функция ослабления $f(r)$ представлена, например законом Ципфа, то сумма Кластеров с учетом ослабления связей может быть представлена как сумма:

$$K_{\Sigma} = \sum_{r=1}^R K_r * \frac{1}{r} = K_1 + 0,50 * K_2 + 0,33 * K_3 + 0,25 * K_4$$

В общем случае Полным Кластером последовательности будет (формула 30 - Полный Кластер объекта):

$$K_{\Sigma} = \sum_{r=1}^R K_r * f(r)$$

где $f(r)$ - функция ослабления веса связи между Кластерами, а r - расстояние между кластерами или ранг Кластера частотного объекта относительно Кластера ключевого объекта.

Учитывая, что сами Кластеры содержат множество частотных объектов совместной встречаемости с объектом, породившим такой Кластер, а каждому частотному объекту в Кластере присвоен вес, то вес каждого частотного объекта при сложении Кластеров может быть умножен на вес Кластера в последовательности Кластеров.

Определение 3.

Полный Кластер последовательности K_{Σ} (формула 30) далее будем называть Трубой и обозначать T . Операция суммирования Кластеров порождает Кластер, поэтому можно говорить о Свертке Кластеров объектов последовательности в один Кластер - в Трубу.

3.2.6. Когерентные Ранговые Кластеры.

В заявленном способе сравнивают ранговые множества разных рангов (далее "Когерентные множества") для известных ключевых Объектов последовательности, причем ранг рангового множества для каждого ключевого Объекта выбирают соответствующим числом Объектов последовательности, разделяющих названный ключевой Объект и Объект-Гипотезу (далее "Фокальный Объект когерентных множеств"), возможность появления которого проверяют.

Назовем Когерентными Кластерами такие Ранговые Кластеры разных объектов последовательно-

сти, ранг которых определен в отношении местоположения одного и того же объекта этой же последовательности. На фиг. 14 Ранговые Кластеры объектов показаны в виде окружностей. Видно, что объект C_1 одновременно находится на пересечении Ранговых Кластеров ($K_4^3 \cap K_3^2 \cap K_2^1$) соответственно объектов C_4 , C_3 и C_2 , поэтому нарисованные в виде окружностей Ранговые Кластеры K_4^3 , K_3^2 , K_2^1 и называются Когерентными. Объект C_1 должен быть частотным объектом соответствующих Когерентных Кластеров объектов C_2 , C_3 и C_4 и это обстоятельство можно использовать для построения и анализа гипотез появления объектов, а также для коррекции ошибок ввода.

Объект C_1 для Когерентных Кластеров K_4^3 , K_3^2 , K_2^1 будем называть "фокальным объектом" Когерентных Кластеров или "фокусом когерентности" (см. фиг. 14).

Очевидно, что фокальный объект является результатом пересечения Когерентных Кластеров (формула 31 - гипотеза как пересечение Когерентных Кластеров):

$$C_1 = (K_4^3 \cap K_3^2 \cap K_2^1)$$

$$C_2 = (K_4^2 \cap K_3^1 \cap K_1^{-1})$$

$$C_3 = (K_4^1 \cap K_2^{-1} \cap K_1^{-2})$$

$$C_4 = (K_3^{-1} \cap K_2^{-2} \cap K_1^{-3})$$

Несмотря на то что в формуле выше (формула 31) использовались знаки равенства, пересечению Когерентных Кластеров может соответствовать не один фокальный объект, а множество. Наличие более одного фокального объекта может быть обусловлено наличием его синонимов или другими причинами. Поэтому корректнее будет записать так (формула 32 - гипотеза как одно из пересечений Когерентных Кластеров):

$$C_1 \in (K_4^3 \cap K_3^2 \cap K_2^1)$$

$$C_2 \in (K_4^2 \cap K_3^1 \cap K_1^{-1})$$

$$C_3 \in (K_4^1 \cap K_2^{-1} \cap K_1^{-2})$$

$$C_4 \in (K_3^{-1} \cap K_2^{-2} \cap K_1^{-3})$$

Сравнивая вес фокального объекта или фокальных объектов суммы Когерентных Кластеров (КК) с весами других частотных объектов суммы КК, можно сделать заключение о вероятности появления одного или другого фокального объекта в качестве соответствующего объекта последовательности.

Как уже говорилось выше, объект последовательности C_1 одновременно является частотным объектом Когерентных Кластеров: K_4^3 (Кластер ранга $r=3$ для объекта C_4), K_3^2 (Кластер ранга $r=2$ для объекта C_3) и K_2^1 (Кластер ранга $r=1$ для объекта C_2). То есть:

$$C_1 \in \left\{ \begin{array}{l} K_4^3 \\ K_3^2 \\ K_2^1 \end{array} \right.$$

Аналогично:

$$C_2 \in \left\{ \begin{array}{l} K_4^2 \\ K_3^1 \\ K_1^{-1} \end{array} \right.,$$

а также

$$C_3 \in \left\{ \begin{array}{l} K_4^1 \\ K_2^{-1} \\ K_1^{-2} \end{array} \right.$$

и наконец:

$$C_4 \in \left\{ \begin{array}{l} K_3^{-1} \\ K_2^{-2} \\ K_1^{-3} \end{array} \right.$$

Очевидно, что сумма значений весовой функции $f_1(r)$ для объекта последовательности C_1 (он же фокальный объект Когерентных Кластеров) в сумме Когерентных Кластеров K_4^3 , K_3^2 и K_2^1 :

$$f_1^\Sigma(r) = \sum_{i=1}^n f_1^i(r)$$

должна стремиться к максимуму среди суммарных весов $f_1^\Sigma(r)$ всех частотных объектов C_i суммы Когерентных Кластеров:

$$f_1^\Sigma(r) \rightarrow \max(f_1^\Sigma(r)) \text{ для всех } C_1 \in (K_4^3 + K_3^2 + K_2^1)$$

Аналогично для других объектов последовательности в примере (фиг. 4) получим:

$$f_2^\Sigma(r) \rightarrow \max(f_2^\Sigma(r)) \text{ для всех } C_2 \in (K_4^2 + K_3^1 + K_1^{-1})$$

а также

$$f_3^\Sigma(r) \rightarrow \max(f_3^\Sigma(r)) \text{ для всех } C_3 \in (K_4^1 + K_2^{-1} + K_1^{-2})$$

и

$$f_4^\Sigma(r) \rightarrow \max(f_1^\Sigma(r)) \text{ для всех } C_4 \in (K_3^{-1} + K_2^{-2} + K_1^{-3})$$

Описанное свойство гипотезы $f_x^\Sigma(r) \rightarrow \max(f_1^\Sigma(r))$ стремиться к максимуму веса среди суммарных весов частотных объектов в сумме Когерентных Кластеров можно использовать для решения задач "ученого" и "следопыта" - поиска гипотез продолжения последовательности в будущее или прошлое, а также для восстановления объектов, записанных с ошибкой или отсутствующих объектов последовательности.

3.2.7. Прогнозирование будущего

Если нам известны R объектов Окна внимания последовательности и надо решить задачу ученого спрогнозировав появление в будущем объекта $C_{(R+n)}$ с номером (R+n), то сумма Когерентных Кластеров $KK(C_{(R+n)})$ в общем случае может быть рассчитана так формула 33 - поиск гипотез будущего):

$$C_{R+n}: \left\{ \begin{array}{l} \in KK(C_{(R+n)}) = \sum_{r=1}^{R+n-1} K_r^{((R+n)+(1-r))} \\ f_{R+n}^\Sigma(r) \rightarrow \max(f_1^\Sigma(r)) \text{ для всех } C_i \in KK(C_{(R+n)}) \\ \in \bigcap_{r=1}^{R+n-1} K_r^{((R+n)+(1-r))} \end{array} \right.$$

Как было показано выше, при поиске ранее записанных копий последовательности (воспоминание) для объекта C_{R+n} из ранее записанной в память копии одновременно с выполнением условия (формула 33) должны быть также выполнены следующие условия (формула 34 - дополнительные условия для поиска копий):

$$C_{(R+n)-1} \in K_{R+n}^{-1}$$

$$C_{(R+n)-2} \in K_{R+n}^{-2}$$

.....

$$C_R \in K_{R+n}^{-n}$$

.....

$$C_{R-1} \in K_{R+n}^{-(n+1)}$$

.....

$$C_0 \in K_{R+n}^{-(R+n)}$$

а также должны быть выполнены весовые условия копии (формула 25).

3.2.8. Прогнозирование прошлого

А задачи следопыта (предсказание появления объекта прошлого $C_{(-n)}$) сумма Когерентных Кластеров может быть найдена так (формула 35 - поиск гипотез прошлого):

$$C_{-n}: \left\{ \begin{array}{l} \in KK(C_{(-n)}) = \sum_{r=1}^{R+n-1} K_r^{(n+(1-r))} \\ f_{-n}^\Sigma(r) \rightarrow \max(f_1^\Sigma(r)) \text{ для всех } C_i \in KK(C_{(-n)}) \\ \in \bigcap_{r=1}^{R+n-1} K_r^{(n+(1-r))} \end{array} \right.$$

А также для поиска копий (формула 36 - Дополнительные условия для поиска копий):

$$C_{-n+1} \in K_{-n}^1$$

$$C_{-n+2} \in K_{-n}^2$$

.....

$$C_0 \in K_{-n}^n$$

.....

$$C_R \in K_{-n}^{(n+R)}$$

а также должны быть выполнены весовые условия объектов копии (формула 25).

3.2.9. Последовательность прогнозов.

Предположим, нам известны R объектов Окна внимания, и мы хотим решить задачу ученого (формула 33), построив гипотезу появления следующих n объектов будущего.

Очевидно, что прогнозы следует делать, последовательно увеличивая глубину прогнозирования, начав с $n=R+1$, затем перейдя к $n=R+2$ и так далее до $n=R+N$.

Поэтому начинаем с гипотез для объекта $C_{(R+1)}$:

$$C_{(R+1)}: \begin{cases} \in KK(C_{(R+1)}) = \sum_{r=1}^R K_r^{((R+1)+(1-r))} \\ f_{R+1}^{\Sigma}(r) \rightarrow \max(f_i^{\Sigma}(r)) \text{ для всех } C_i \in KK(C_{(R+1)}) \\ \in \bigcap_{r=1}^R K_r^{((R+1)+(1-r))} \end{cases}$$

Если мы хотим убедиться, что вводимая последовательность является копией ранее записанной последовательности, то для каждой из гипотез $C_{(R+1)}$ проверяем выполнение условий (формула 25 и формула 34).

Для каждой из гипотез $C_{(R+1)}$ ищем продолжение $C_{(R+2)}$:

$$C_{R+2}: \begin{cases} \in KK(C_{(R+2)}) = \sum_{r=1}^{R+1} K_r^{((R+2)+(1-r))} \\ f_{R+2}^{\Sigma}(r) \rightarrow \max(f_i^{\Sigma}(r)) \text{ для всех } C_i \in KK(C_{(R+2)}) \\ \in \bigcap_{r=1}^{R+1} K_r^{((R+2)+(1-r))} \end{cases}$$

Для каждой из гипотез $C_{(R+2)}$ проверяем выполнение условий (формула 25 и формула 34).

И так далее до $n=N$:

$$C_{R+N}: \begin{cases} \in KK(C_{(R+N)}) = \sum_{r=1}^{R+N-1} K_r^{((R+N)+(1-r))} \\ f_{R+N}^{\Sigma}(r) \rightarrow \max(f_i^{\Sigma}(r)) \text{ для всех } C_i \in KK(C_{(R+N)}) \\ \in \bigcap_{r=1}^{R+N-1} K_r^{((R+N)+(1-r))} \end{cases}$$

для каждой из гипотез $C_{(R+N)}$ проверяем наличие копий с помощью выполнения условий (формула 25 и формула 34).

Аналогичным образом решается и задача следопыта.

3.2.10. Обратная проекция Кластера.

Память последовательностей позволяет отображать объект в соответствующий ему Кластер и можно ожидать, что существует обратное отображение Кластера в соответствующий ему объект или объекты - Родители, которые могли породить такой Кластер. Если операцию порождения Кластера для уникального объекта можно назвать декомпозицией объекта в Кластер, то обратную операцию проекцией Кластера на объект. Поэтому обратное отображение Кластера на объект или объекты станем называть Обратной Проекцией Кластера.

Одним из примеров Обратной Проекции Кластера может служить техника проецирования Когерентных Ранговых Кластеров на один или более фокальных объектов. Рассмотрим ее подробнее.

Предположим в памяти хранится три последовательности содержащих элемент А в середине (фиг. 15).

Очевидно (фиг. 16), что для элемента А Кластером будущего будет множество $K_A=(B,C,D)$.

Предположим, что в памяти также содержатся еще по две последовательности с каждым из элементов В, С и D (фиг. 17- 19).

Построим теперь Ранговые Кластеры прошлого первого ранга - K_B , - K_C и K_D соответственно для элементов В, С и D (фиг. 20).

Как видно (фиг. 20), в каждом из Кластеров прошлого элементов В, С и D присутствует элемент А, что позволяет его обнаружить двумя способами:

1. Объект А можно обнаружить путем поиска пересечения множеств Кластеров $\{-K_B^{-1} \cap -K_C^{-1} \cap -K_D^{-1}\}$, однако, как и в случае фокальных объектов, пересечение может содержать более одного объекта.

2. При наличии в пересечении нескольких объектов, вероятность появления каждого объекта можно определить как его суммарный вес W_B , W_C или W_D в сумме Кластеров $\{(K_B^{-1})+(-K_C^{-1})+(-K_D^{-1})\}$.

3. Очевидно, объекты, найденные на пересечении Кластеров, будут иметь вес выше, чем другие частотные объекты в сумме Кластеров.

Таким образом техника построения Обратной проекции позволяет выделить гипотезы.

Для приведенного выше примера (фиг. 16) Кластера будущего K_A^1 объекта А, содержащего частотные объекты (В, С, D), Обратная Проекция Первого Ранга показана на фиг. 20. Пунктирными окружностями показаны Кластеры прошлого K_B^{-1} , K_C^{-1} и K_D^{-1} частотных объектов В, С и D, точкой пересечения которых является объект А, который является Родителем Кластера K_A^1 . В данном случае Кластерами являются Ранговые Кластеры первого ранга ($r=1$) и они Когерентны. В общем случае в фокусе Обратной Проекции могут появиться не один, а несколько объектов, потенциальных Родителей Кластера для кото-

рого делалась Обратная Проекция.

Если рассмотренную ранее технику определения фокального объекта Когерентных Кластеров можно назвать техникой "продольного" проецирования, поскольку объекты-источники пересекающихся Кластеров (Когерентных Ранговых Кластеров) расположены на самой последовательности (в ее плоскости), то Обратную Проекцию следовало бы назвать "поперечной" проекцией Когерентных Кластеров, потому что объекты-источники пересекающихся Когерентных Кластеров лежат в плоскости, перпендикулярной линии последовательности.

Как и в случае с продольной проекцией Когерентных Кластеров, поперечная проекция ((обратная проекция) может определить несколько фокальных объектов. В случае текста это могут быть, например, "словоформы" одного слов или синонимы.

3.2.11. Ранговая Обратная Проекция.

Для каждого из частотных Объектов конкретного рангового или полного множества в способе из памяти извлекают ранговое множество, для которого названный частотный Объект является ключевым Объектом, извлеченные ранговые множества одинакового ранга сравнивают с целью определения по меньшей мере одного Объекта-Гипотезы.

Хотя фиг. 21 демонстрирует Обратную проекцию первого ранга, понятно, что с помощью техники Обратной проекции второго и более высокого ранга можно строить гипотезы появления объектов второго и более высокого ранга в обратной проекции на последовательность. Понятно также, что для гипотезы, которая является частью копии последовательности, хранимой в памяти последовательностей, Обратная проекция каждого ранга должна содержать известный объект последовательности соответствующего ранга g относительно гипотезы. Так для последовательности (фиг. 22) должно выполняться условие (формула 37 - Ранговая Обратная Проекция)

$$G \in \{-K_B^{-4} \cap -K_C^{-4} \cap K_D^{-4}\}$$

$(W_G * G)$ является членом суммы $\{(K_B^{-4}) + (-K_C^{-4}) + (-K_D^{-4})\}$

$$F \in \{-K_B^{-3} \cap -K_C^{-3} \cap K_D^{-3}\}$$

$(W_F * F)$ является членом суммы $\{(K_B^{-3}) + (-K_C^{-3}) + (-K_D^{-3})\}$

$$G \in \{-K_B^{-2} \cap -K_C^{-2} \cap K_D^{-2}\}$$

$(W_E * E)$ является членом суммы $\{(K_B^{-2}) + (-K_C^{-2}) + (-K_D^{-2})\}$

3.2.12. Структура данных нумерованной памяти последовательностей.

Как отмечалось выше, [2.3.9] полный Кластер уникального объекта может быть представлен линейной композицией одного из ранговых Кластеров, поэтому в памяти достаточно хранить любой один ранговый Кластер, предпочтительно Кластер первого ранга. Тем не менее хранение в памяти последовательностей полного Кластера снижает время доступа к нему, так как избавляет от необходимости расчета полного Кластера как линейной композиции ранговых кластеров. В памяти можно также хранить несколько Кластеров последовательных рангов от первого ранга до ранга N , что позволяет снизить трудоемкость операций для построения обратных проекций и проведения операций над когерентными ранговыми кластерами. Поэтому для каждого уникального объекта, память последовательностей должна хранить, по меньшей мере:

1. Уникальных цифровой код объекта.
2. Один ранговый Кластер объекта, предпочтительно Кластер первого ранга.

Преимуществом предложенной структуры данных нумерованной памяти последовательностей перед индексом поисковых машин является существенно более высокая производительность прогнозирования, за счет хранения по меньшей мере одного рангового Кластера.

Память нумерованных последовательностей может также хранить содержащие Ключевой Объект фрагменты последовательности, представляющие собой очередь из нескольких объектов соответствующей последовательности в которой Ключевой объект занимает определенное ранее постоянное местоположение (например в середине, конце, начале или в другой определенной позиции очереди), причем при вводе последовательностей в память на каждом цикле ввода на вход памяти подаются названную очередь объектов Окна Внимания, а на следующем цикле ввода очередь сдвигают по меньшей мере на один объект в будущее или прошлое.

Как и в случае с индексом поисковых систем, все названные данные памяти последовательностей могут храниться в хитах.

3.2.13. Применение предложенной техники прогнозирования.

3.2.13.1. Техника Когерентных Ранговых Кластеров.

Для всех известных объектов последовательности строятся Ранговые Когерентные Кластеры с фокусом на объект прогнозирования. Если техника Когерентных Ранговых Кластеров дала более одной гипотезы, то встает задача выбора из них наиболее подходящей, которая в своей обратной проекции должна содержать максимальное число известных предшествующих (задача ученого) или последующих (задача следопыта) объектов последовательности.

3.2.13.2. Техника Обратной проекции.

Поскольку множество гипотез представляет собой Кластер, порожденный известными объектами последовательности, то к нему можно применить проиллюстрированную выше технику Обратной проекции Кластера. То есть построить для каждой гипотезы полный Кластер прошлого и Ранговые Кластеры прошлого с целью найти в этих Кластерах известные предшествующие объекты последовательности. В случае задачи ученого следует использовать обратную проекцию множества гипотез будущего на предшествующие объекты последовательности, а для задачи следопыта следует использовать обратную проекцию гипотез прошлого на последующие объекты последовательности.

3.2.13.3. Техника Ранговых Кластеров.

Для каждой из гипотез строятся Ранговые Кластеры с фокусом на известные объекты последовательности - предшествующие объекты последовательности для задачи ученого или последующие объекты для задачи следопыта. Известные объекты последовательности должны содержаться в соответствующих Ранговых Кластерах гипотезы и наиболее подходящей можно считать ту, в Ранговых Кластерах которой содержится больше таких объектов или их вес максимален.

3.2.13.4. Алгоритм поиска гипотез.

Приведем описанный выше алгоритм поиска гипотез в кратком изложении:

1. построение ранговых кластеров для каждого из известных объектов последовательности (формула 28).
2. расчет когерентных кластеров для каждого из объектов последовательности, появление которых прогнозируется - гипотезы (формула 33, формула 35).
3. выбор из множества частотных объектов каждого Когерентного Кластера (КК) в качестве гипотезы таких частотных объектов КК, которые обладают максимальными весами среди всех объектов КК и одновременно являются фокальными объектами пересечения этих Ранговых Кластеров (формула 33, формула 35).
4. построение Ранговых Кластеров обратной проекции для каждой гипотезы для оценки правильности сделанного прогноза путем поиска известных соответствующих объектов последовательности в каждом из Ранговых Кластеров с высоким весом. (формула 28).
5. Выявление наличия в памяти копий вводимой последовательности, путем проверки условия (формула 25).

3.2.14. Коррекция ошибок ввода.

В заявленном способе при вводе Объекта, уникальный цифровой код которого мог быть введен с ошибкой, сравнение ранговых множеств осуществляют с целью выявления возможной ошибки.

Считается, что ухо человека распознает около 60% слов, произнесенных другими людьми, а 40% сказанного человек домысливает, то есть строит гипотезы о том, что могло быть сказано, на основе того, что он слышал и понял ранее. При этом ошибочно распознанным может быть как последнее услышанное слово, так и ранее распознанное слово может оказаться распознанным неверно. Ошибки ввода также случаются при работе программного обеспечения распознавания. Например, при распознавании текста могут быть распознаны неверно отдельные буквы или слова в середине слова или фразы.

Поскольку гипотезы разумно строить, опираясь на известные объекты последовательности, то решение задачи ученого или задачи следопыта является экстраполяцией смысла известной части последовательности соответственно на будущее или на прошлое [3.2.13.4]. Обнаружение же ошибки ввода внутри известного участка последовательности является задачей интерполяции. В случае интерполяции анализ возможного "ошибочного" объекта можно проводить путем одновременного решения и задачи ученого с опорой на известные объекты последовательности, предшествующие "ошибочному", и задачи следопыта с опорой на известные объекты последовательности, следующие за "ошибочным" с использованием алгоритма поиска гипотез [3.2.13.4].

3.2.15. Кластеры как исходные данные или карты признаков для нейронных сетей.

Как было показано выше, Рекурсивный Индекс реализует два противоположных процесса:

декомпозиция объекта в карту его признаков;

синтез объекта на основе карты признаков.

Любой созданный Рекурсивным Индексом Кластер объекта является декомпозицией объекта в карту его признаков. В свою очередь Когерентные Ранговые Кластеры и Обратная проекция Кластера позволяет решить обратную задачу - идентифицировать объект, которому могла соответствовать заданная карта признаков. Это существенно расширяет круг задач искусственного интеллекта, которые может решать система, состоящая из Рекурсивного Индекса и нейронной сети.

В систему вводят объекты последовательности. Для каждого объекта вводимой последовательности с помощью Рекурсивного Индекса порождают Кластер и подают Кластер на вход нейронной сети в качестве карты признаков. Последовательность объектов с помощью Рекурсивного Индекса представляют последовательностью их Кластеров, которую подают в нейронную сеть для обучения нейронной сети, или для решения задач и принятия решений. На вход нейронной сети могут подавать не только исходные Кластеры объектов последовательности, но также и другие типы Кластеров, описанные в настоящей работе.

Как отмечают специалисты, используемые алгоритмы и технологии обучения нейронных сетей не позволяют людям понять механизм принятия решений нейронными сетями. Это ограничивает использование нейронных сетей, особенно в тех областях, где принятие решений может быть связано с риском для жизни человека. Непредсказуемость работы нейронных сетей, в частности, связана с использованием метода обратного распространения ошибки, присваивающего связям сети веса, которые предсказать невозможно. Поэтому одним из преимуществ формирования карт признаков с помощью Рекурсивного Индекса (Памяти Последовательностей) является то, что Рекурсивный Индекс (Память Последовательностей) позволяет определить вес каждого из частотных объектов в Кластере для любого ключевого объекта последовательности, что может позволить определять и веса связей нейронной сети.

3.2.16. Обобщения.

Известно, что когда сила множества входных сигналов в нейрон превышает некоторый потенциал действия нейрона, то нейрон генерирует выходной импульс - спайк. Важным для нас в этом представлении является то, что замечая постоянное возбуждение одной и той же группы нейронов, мозг может "назначать" ранее свободный нейрон "ответственным" за эту группу нейронов и всякий раз, когда такая группа нейронов возбуждается, "ответственный" за нее нейрон отслеживает уровень возбуждения группы и, если уровень возбуждения превышает некоторый критический уровень, то "ответственный" нейрон испускает спайк.

Далее мы рассмотрим механизмы синтеза новых объектов, ответственных за одновременное возбуждение группы объектов, условие возбуждения профиля которых описано ранее (формула 6).

4. Трубы. Уплотнение смысла и времени.

4.1. Синтез обобщений.

Как было показано выше, предложенная техника обратной проекции Кластера исследуемого объекта позволяет отобразить Кластер в множество возможных объектов-Родителей Кластера. Такое множество имеет меньшую размерность, чем Кластер и состоит из объектов, объединенных смысловой общностью. Это может быть синонимия в широком смысле - словоформы одного слова, разные слова с одинаковым смыслом (синонимы), части обобщенного понятия и так далее. Можно говорить о синтезе инвариантного представления для исследуемого объекта и объектов множества обратной проекции Кластера исследуемого объекта.

Поскольку каждый из уникальных объектов памяти последовательностей может быть представлен Кластером, то и множеству обратной проекции Кластера может соответствовать отдельный объект (фиг. 23), в частности такой, который мы искусственно для этого создадим - синтезируем, поэтому такой объект является синтетическим. В языке аналогом такого подхода является создание аббревиатур, а также назначение одной из словоформ в качестве "исходной" или "нейтральной", так например все словоформы "шел", "иду", "шел" считаются словоформами слова "идти" хотя любая из словоформ могла претендовать на то, чтобы быть исходной формообразующей.

Синтез множества обратной проекции Кластера будем далее называть "поперечный синтезом", имея в виду возможную замену исследуемого объекта последовательности другим объектом из множества обратной проекции, что приводит к синтезу альтернативных вариантов последовательности. Далее рассмотрим "продольный синтез", имея в виду сжатие исходных последовательностей до более короткой последовательности синтетических объектов.

Обратная Проекция Кластера порождает множество тождественных объектов, одним из которых может быть объект, которому принадлежит Кластер, подвергнутый Обратной Проекции, и в таком случае необходимость в синтезе нового объекта отпадает. Поэтому нужен механизм позволяющий принять решение о синтезе нового объекта или об отказе от синтеза в пользу уже существующего уникального объекта последовательностей.

Решение о синтезе нового объекта принимается, если погрешность тождественности профиля (или нормализованного профиля) исходного Кластера при его сравнении с аналогичными профилями Кластеров объектов Обратной Проекции превышает величину допустимой погрешности ΔK_{\max} (формула 6).

4.2. Смысловое сжатие последовательностей.

4.2.1. Труба. Калибр Трубы.

В заявленном способе из Множества Трубы извлекают и складывают весовые коэффициенты встречаемости всех частотных Объектов, получая Суммарный Вес Трубы.

Представляется очевидным, что вероятность совместной встречаемости в тексте слов "молоко" и "сыр" выше, чем слов "молоко" и "нефть", поэтому Кластеры слов "молоко" и "сыр" должны содержать больше одинаковых частотных слов (например, слово "корова", "брожение", "животноводство" и другие), чем Кластеры слов "молоко" и "нефть". Другими словами пересечение Кластеров слов "молоко" и "сыр" будет содержать больше объектов, чем пересечение Кластеров слов "молоко" и "нефть".

Число объектов $(K_{\text{МОЛОКО}} \cap K_{\text{СЫР}}) > \text{Число объектов } (K_{\text{МОЛОКО}} \cap K_{\text{НЕФТЬ}})$.

Это означает, что, складывая Кластеры родственных слов, таких как "молоко" и "сыр" $(K_{\text{МОЛОКО}} + K_{\text{СЫР}})$, мы обнаружим увеличение веса слов входящих в пересечение $(K_{\text{МОЛОКО}} \cap K_{\text{СЫР}})$, соответствующих контексту словосочетания и входящих в каждый из Кластеров, в то время как веса слов, не

попавших в множество пересечения, изменяться не будут. Математически множество объектов с растущими весами назовем контекстом $Cont$ и определим как сумму Кластеров объектов ($K_{МОЛОКО} + K_{СЫР}$) без их симметрической разницы ($K_{МОЛОКО} \Delta K_{СЫР}$):

$$Cont = (K_{МОЛОКО} + K_{СЫР}) - \Delta Cont_{СЫР}^{МОЛОКО}$$

где $\Delta Cont_{СЫР}^{МОЛОКО}$ является представлением линейной алгебры для операции поиска симметрической разницы множеств ($K_{МОЛОКО} \Delta K_{СЫР}$).

В общем случае контекстом последовательности из R объектов или Трубой последовательности будет сумма Кластеров всех объектов без их симметрической разницы (формула 38 - Труба - контекст последовательности):

$$T = Cont(R) = f(1) * K_1 + \sum_{i=2}^R \{f(i) * K_i - \Delta Cont_{i-1}^i\} = \sum_{i=1}^R f(i) * K_i - \Delta Cont(R)$$

где $\Delta Cont_n^m$ - представление симметрической разницы множеств ($K_m \Delta K_n$) объектов m и n как операции линейной алгебры $\Delta Cont_n^m \leftrightarrow (K_m \Delta K_n)$, а $f(i)$ - функция ослабления, которая в некоторых случаях может быть принята равной единице и тогда:

$$Cont(R) = K_1 + \sum_{i=2}^R \{K_i - \Delta Cont_{i-1}^i\} = \sum_{i=1}^R K_i - \Delta Cont(R)$$

Понятно, что при изменении контекста последовательности должно меняться и содержание множества $Cont(R)$. По мере ввода объектов с неизменным контекстом скорость изменения $Cont(R)$ будет снижаться по закону Хипса, согласно которому число уникальных объектов последовательности прямо пропорционально квадратному корню из числа всех объектов последовательности и значит скорость увеличения числа уникальных объектов будет ниже скорости увеличения числа введенных объектов пропорционально корню из всех объектов последовательности. Так в последовательности из 250 тысяч объектов уникальными будут (строго говоря уникальным, будет число слов $= c \times 0,2\%$, где c - некоторая константа.) $0,2\%$, а для последовательности из 360 тысяч объектов уникальными окажется уже лишь $0,16\%$, то есть во второй последовательности удельный вес уникальных объектов будет на 25% меньше, при том, что сама вторая последовательность будет длиннее первой на 44% . Кроме того, объекты в последовательности, например, слова в тексте, не являются случайным набором, они связаны контекстом и порядок их следования подчинен законам языка. Следовательно, сохранение тематики текста должно замедлять рост суммарного веса объектов множества $Cont(R)$, а изменение тематики напротив приводит к быстрому снижению числа объектов множества $Cont(R)$ с одновременным уменьшением максимальных весов входящих в $Cont(R)$ объектов. Снижение числа объектов и их веса при смене контекста происходит из-за замены прежней группы частотных объектов множества $Cont(R)$ соответствующих прежнему контексту новыми, в результате чего общий вес $Cont(R)$ должен сначала упасть, а затем начать расти по мере формирования множества $Cont(R)$ частотных объектов новой тематики.

В заявленном способе из Суммарного Веса Трубы очередного Множества Трубы вычитают Суммарный Вес Трубы предыдущего Множества Трубы и, если разница не превышает установленной погрешности, то результат сохраняют в качестве множества Калибра Трубы, создают идентификатор синтетического Объекта и ставят друг другу в соответствие названный идентификатор, множество Калибра Трубы и множество объектов Окна Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

Запоминание содержания множества $Cont(R)$ на пике суммарного веса его объектов позволяет синтезировать Кластер $Cont(R)$ соответствующий контексту участка последовательности, расположенной между пиками $Cont(R)$ - предыдущим пиком и тем, который мы только что запомнили. Вычисляя суммарный вес объектов $Cont(R)$ с вводом каждого нового объекта последовательности мы можем определить момент, когда рост суммарного веса сменится снижением и множество контекста с пиковым значением суммарного веса объектов $Cont_{max}(R)$ перед началом снижения суммарного веса будет соответствовать множеству контекста Трубы.

Множеству $Cont_{max}(R)$ присваивается идентификатор несуществующего ранее - "синтетического" объекта и такой вновь синтезированный объект добавляется в множество уникальных объектов Памяти Последовательностей. Одновременно с этим создаются упреждающие и обратные связи такого синтетического объекта со всеми объектами последовательности, ввод которых привел к возникновению синтетического объекта Трубы (формула 39 - максимальное значение контекста и Трубы):

$$T_{max} = Cont_{max}(R)$$

Для того чтобы избежать ошибок определения контекста с пиковым суммарным значением $Cont_{max}(R)$ из-за случайного снижения суммарного веса объектов, следует использовать известные методы усреднения или сглаживания кривой изменения суммарного веса.

Операция определения $Cont_{max}(R)$ позволяет "сжать" (фиг. 24) исходную последовательность объектов до Кластера $Cont_{max}(R)$.

Разделение последовательности на отрезки или участки между пиковыми значениями $\text{Cont}_{\max}(\mathbf{R})$, позволяет заменить исходную последовательность объектов или их Кластеров более короткой последовательностью синтетических объектов - Труб (фиг. 25), которым соответствуют Кластеры $\text{Cont}_{\max}(\mathbf{R})$, позволяя произвести смысловое "сжатие" исходной последовательности объектов до последовательности синтетических объектов $\text{Cont}_{\max}(\mathbf{R})$.

Определение 4.

Генератор Трубы - это последовательность объектов, породившая Кластер Трубы.

Соответствие Кластера $\text{Cont}_{\max}(4)$ сразу четырем объектам C_1, C_2, C_3 и C_4 и их Кластерам K_1, K_2, K_3 и K_4 , не только "сжимает" последовательность до одного Кластера $\text{Cont}_{\max}(\mathbf{R})$, но также создает обратные и упреждающие связи синтетического объекта $\text{Cont}_{\max}(\mathbf{R})$ между объектами C_1, C_2, C_3 и C_4 и их Кластерами K_1, K_2, K_3 и K_4 , создавая основу для осуществления логических выводов.

Операцию удаления симметрической разности Кластеров объектов из множества Трубы назовем Калибровкой Трубы, а результат назовем Калибром и станем обозначать как K_T . Очевидно, что калибр трубы и является множеством $\text{Cont}(\mathbf{R})$ (формула 40):

$$K_T = \text{Cont}(\mathbf{R})$$

Теперь выражение для контекста последовательности (формула 38) можно переписать так (формула 41 - калибр трубы):

$$K_T = T - \Delta K_T = \sum_{i=1}^R (f(r) * K_i) - \Delta K_T$$

где

$$\Delta K_T = \Delta \text{Cont}(\mathbf{R})$$

Анализ изменения Трубы и ее Калибра можно проводить с применением известных методов математического анализа, линейной алгебры, статистического анализа и других известных математических методик, поэтому здесь останавливаться на них не станем.

Ранее мы выделили два типа Кластеров - Кластер будущего и Кластер прошлого, поэтому и Труба, построенная с использованием Кластеров только одного типа, соответственно будет Трубой прошлого или Трубой будущего. На последовательности Труб и их Калибров можно строить как Полный Кластер, так и Ранговые Кластеры, что позволяет строить гипотезы на разных уровнях абстракции и смысла, а также, по сути, создает обратные и упреждающие связи при переходе с более высокого порядка на низкие, порождая возможность построения выводов и суждений.

Из комбинаторики известно, что "число размещений с повторениями" равняется N^k , где N - число всех уникальных объектов множества уникальных объектов, а k - число объектов в фрагменте, на котором построена Труба. Например для множества из 100 тысяч уникальных объектов число размещений с повторениями будет равна 100 и соответственно вероятность повторения фрагмента из 10 объектов в разных хитах будет равна $1/(100)$. На самом деле вероятность повторения будет значительно ниже, потому что не все сочетания уникальных объектов допустимы, как и повторения не часты. Тем не менее приведенная величина вероятности позволяет понять, что Труба, построенная на фрагменте из десяти объектов, с очень высокой вероятностью будет содержать "воспоминание" памяти последовательностей - объекты для продолжения такого фрагмента.

4.2.2. Вычисление Калибра Трубы.

Каждый уникальный частотный Объект, который не встречается по меньшей мере в одном из массивов или ранговых множеств объектов Окна Внимания, или удаляют из Множества Трубы или приравнивают нулю его вес, а полученное множество считают множеством Калибра Трубы, затем названному множеству Калибра Трубы ставят в соответствие существующий или вновь созданный Объект Памяти Последовательностей (далее "Синтетический Объект"), а также ставят в соответствие Окно Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

Нетрудно заметить, что Труба будущего содержит связи с объектами будущего в каждой из последовательностей, на которых была обучена память последовательностей, то есть Труба содержит все связи с возможными будущими объектами текущей последовательности "записанными" в память последовательностей. Труба будущего содержит ветвления (возможные продолжения или гипотезы), исходящие из каждого из объектов текущей последовательности и для всех объектов последовательности в совокупности не все продолжения каждого из объектов могут быть реализованы. Это означает, что для выделения в памяти только реализуемых гипотез продолжения заданной последовательности, нужна еще одна операция - Калибровка Трубы.

Существует эффект, называемый латеральным торможением. Нервная система использует латеральное торможение для фокусировки или "обострения" первичного сигнала, чтобы обеспечить проведение первичного сигнала в желаемом направлении без ослабления, одновременно подавляя тенденцию сигналов к латеральному распространению. Калибровка Трубы, по сути, и является операцией такой

"фокусировки" Трубы.

Калибровка позволяет убрать такие ветки развития последовательности, которые не являются продолжением одновременно для всех известных объектов последовательности. Если считать известными объектами последовательности и те, чье появление было ранее спрогнозировано, то это позволит экстраполировать прогноз дальше в будущее или прошлое. В терминах поиска гипотез мы можем определить Калибровку так:

Определение 5.

Калибровка Трубы будущего - это операция порождения массива будущего, содержащего объекты будущего с их весовыми коэффициентами, который мы назовем Калибром Трубы и который, в частности, содержит все статистически допустимые продолжения текущей последовательности в будущем для решения "задачи ученого". Соответственно массив Калибра Трубы прошлого содержит все статистически допустимые продолжения текущей последовательности в прошлое для решения "задачи следопыта".

В процессе ввода в ПП последовательности объектов "Окна внимания" для каждого из объектов Окна Внимания как для ключевого Объекта из памяти извлекают по меньшей мере один названный массив или ранговое множество, содержащее весовые коэффициенты встречаемости частотных Объектов, из всех названных массивов или множеств извлекают весовые коэффициенты встречаемости каждого уникального частотного Объекта, общего одновременно для всех названных массивов или множеств, и складывают их, таким образом формируя Множество Трубы, содержащее суммарные весовые коэффициенты встречаемости каждого уникального частотного Объекта со всеми объектами Окна Внимания.

Согласно определению Калибр Трубы K_T равен сумме Гипотез H_r (формула 42):

$$K_T = \sum_{r=1}^R (H_r)$$

Как отмечалось выше, множество Калибра Трубы является подмножеством Трубы (формула 43):

$$K_T = T - \Delta K_T = \sum_{i=1}^R (K_i) - \Delta K_T$$

где ΔK_T представляет симметрическую разность полных Кластеров K_i ключевых объектов последовательности, а каждый полный Кластер K_i является множеством частотных объектов:

$$K_i = (w_1 * C_1, w_2 * C_2, w_3 * C_3, \dots, w_n * C_n),$$

причем число частотных объектов n в каждом Кластере может быть разным.

Для вычисления Калибра Трубы K_T путем удаления из Трубы T объектов названной симметрической разности ΔK_T в заявленном способе каждый уникальный частотный Объект, который не встречается по меньшей мере в одном из массивов или ранговых множеств объектов Окна Внимания, или удаляют из Множества Трубы или приравнивают нулю его вес, а полученное множество считают множеством Калибра Трубы, затем названному множеству Калибра Трубы ставят в соответствие существующий или вновь созданный Объект Памяти Последовательностей (далее "Синтетический Объект"), а также ставят в соответствие Окно Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

Содержанием разности ΔK_T являются все "тупиковые объекты" C_i , которые не являются гипотезами одновременно для всех объектов фрагмента последовательности, на которой построена Труба. Множество тупиковых объектов входящих в ΔK_T можно определить из алгебры множеств как "дополнение" множества K_T до множества T (формула 44)

$$\Delta K_T = T \setminus K_T$$

В теории множеств операция дополнения соответствует логическому отрицанию, поэтому поправка ΔK_T является логическим отрицанием Калибра Трубы K_T (фиг. 26):

$$\Delta K_T \leftrightarrow \neg K_T \leftrightarrow \overline{K_T}$$

Значение ΔK_T можно определить также как симметрическую разность кластеров всех объектов последовательности:

$$\Delta K_T = K_1 \Delta K_2 \Delta K_3 \Delta \dots \Delta K_n$$

учитывая, что симметрическая разность $A \Delta B = (A \setminus B) \cup (B \setminus A)$ (формула 45):

$$\Delta K_T = \left(\left(\left((K_1 \setminus K_2) \cup (K_2 \setminus K_1) \right) \setminus K_3 \right) \cup \left(K_3 \setminus \left((K_1 \setminus K_2) \cup (K_2 \setminus K_1) \right) \right) \right) \setminus K_4 \cup \dots$$

Для удаления из трубы T всех объектов C_i , принадлежащих множеству ΔK_T , введем в рассмотрение квантор-массив Z кванторов $(z_1, z_2, z_3, \dots, z_i, \dots, z_N)$, равномошный массиву T (в котором $0 < i \leq N$, а N - число частотных объектов в массиве Трубы T), причем $z_i = 1$ для каждого из объектов $C_i \in \Delta K_T$, а также $z_i = 0$ для каждого из объектов $C_i \notin \Delta K_T$. Тогда для ΔK_T как для массива будет справедливо равенство (формула 46):

$$\Delta K_T = \bar{Z} * T$$

и окончательно формула вычисления массивов частотных объектов Калибра Трубы примет вид

(формула 47):

$$K_T = T - \bar{Z} * T = \sum_{i=1}^r (f(r) * K_i) - \bar{Z} * \left(\sum_{i=1}^r (f(r) * K_i) \right) = (\bar{Z}_1 - \bar{Z}) * \sum_{i=1}^r (f(r) * K_i)$$

где \bar{Z}_1 - единичное множество равномощное множеству \bar{Z} , а \bar{Z}_1 - содержит в качестве объектов только единицы.

Веса некоторых объектов Калибра Трубы, посчитанные по приведенной выше формуле, окажутся избыточными. При расчете поправки Калибра трубы ΔK_T (см. формула 45) мы не учитывали тот факт, что веса частотных объектов из Кластера каждого объекта $C_i \in \Delta K_T$, при вычислении Калибра Трубы (см. формула 47) были сложены с весами частотных объектов из Кластера каждого объекта $C_i \notin \Delta K_T$. Пользуясь нейроанalogией, можно сказать, что мы "затормозили" первичные нейроны ($C_i \in \Delta K_T$) тупиковых последовательностей, но "торможение" не коснулось вторичных, третичных и так далее нейронов таких тупиковых последовательностей и эти, оставленные возбужденными вторичные и так далее нейроны, способны расфокусировать Калибр Трубы и способствовать зашумлению искомой гипотезы. Поэтому веса частотных объектов, оставшихся в массиве Калибра Трубы (см. формула 47) должны быть дополнительно уменьшены на значение их весов в Кластерах тупиковых объектов ($C_i \in \Delta K_T$), а в значение ΔK_T внесена "весовая поправка" ΔW (формула 48 - весовая поправка Калибра Трубы):

$$\Delta W = \sum_{i=1}^R \sum_{j=1}^n (w_{ji} * C_j)$$

так что

$$\Delta K_T = \bar{Z} * T + \Delta W$$

и тогда формула расчета Калибра Трубы примет вид (формула 49 - Калибр Трубы с учетом весовой поправки):

$$K_T = T - (\bar{Z} * T + \Delta W) = (\bar{Z}_1 - \bar{Z}) * \sum_{i=1}^R (K_i) - \sum_{i=1}^R \sum_{j=1}^n (w_{ji} * C_j)$$

Замечание: 3.

Калибр Трубы (см. формула 49) не содержит объектов последовательности, на которой он был построен, потому что Кластер крайнего левого или крайнего правого объекта последовательности для Трубы будущего или Трубы прошлого не будет содержать объектов последовательности и потому все объекты последовательности как тупиковые окажутся в множестве ΔK_T . Таким образом, восстановит последовательность из Калибра Трубы посчитанному по приведенной формуле (формула 49) нельзя.

Отсутствие объектов последовательности в Калибре Трубы понятно, ведь Калибр Трубы, по сути, содержит только продолжения последовательности в будущее (задача ученого) или прошлое (задача следопыта) из последнего известного объекта последовательности. Таким образом запоминание Генератора Трубы (определение 4) представляется необходимым шагом, дополняющим вычисление Калибра Трубы.

Отсутствие в Калибре Трубы объектов последовательности, на которых был построен Калибр Трубы противоречит известным фактам о возбуждении нейронов - первичные нейроны, которым в нашей модели соответствуют названные объекты последовательности, остаются возбужденными с учетом затухания, при этом передавая возбуждение вторичным и так далее нейронам, которым в нашей модели соответствуют объекты Калибра Трубы.

Из Суммарного Веса Трубы очередного Множества Трубы вычитают Суммарный Вес Трубы предыдущего Множества Трубы и, если разница не превышает установленной погрешности, то результат сохраняют в качестве множества Калибра Трубы, создают идентификатор синтетического Объекта и ставят друг другу в соответствие названный идентификатор, множество Калибра Трубы и множество объектов Окона Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

Во время названного цикла Множество Трубы сравнивают с ранее сохраненным по меньшей мере одним множеством Калибра Трубы, и если отличие Множества Трубы от множества Калибра Трубы сравнимо с некоторой погрешностью, то из Памяти Последовательностей извлекают Генератор Трубы, соответствующий названному множеству Калибра Трубы и используют названный Генератор Трубы в качестве результата поиска (далее "воспоминаний") Памяти Последовательностей в ответ на ввод Окона Внимания в качестве поискового запроса.

Для эмуляции работы нейронов Калибр Трубы следовало бы дополнить объектами последовательности Генератора Трубы $S=(C_1, C_2, C_3, \dots, C_R)$, где R - число объектов последовательности для которой строится Труба. Однако добавление объектов без их весов может сделать добавление скаляра S к Калибру ΔK_T незаметным, поэтому было бы полезно присвоить каждому из частотных объектов C_i , множества S веса, соответствующие суммарному весу W_i , частотных объектов Кластера K_i , такого объекта C_i , и тогда:

$$S = (W_1 * C_1; W_2 * C_2; W_3 * C_3; \dots; W_R * C_R)$$

где R - размер окна внимания.

Поэтому в формулу Калибра Трубы K_T может быть полезно учесть поправку, добавляющую к Трубе T множество известных объектов последовательности S. Это оправдано с точки зрения нейроанalogии, ведь объекты последовательности по сути являются аналогами возбужденных первичных нейронов. Объекты (C_2, C_3, \dots, C_R) следовало добавить с отрицательным знаком, имея в виду, что они уже "в прошлом", а объект C_1 с положительным, поскольку он в "настоящем". Однако используя нейроанalogии можно сказать, что все объекты $(C_1, C_2, C_3, \dots, C_R)$ остаются возбужденными и потому находятся в "настоящем", хотя возбуждение тех из них, которые были введены на r циклов ранее последнего, должно затухать по закону ослабления:

$$f_r = f(r)$$

и тогда с учетом ослабления значением S будет:

$$S = (f_1 * W_1 * C_1; f_2 * W_2 * C_2; f_3 * W_3 * C_3; \dots; f_R * W_R * C_R)$$

где объект C_1 является последним введенным в очередь, а объект C_R самым "старым" из введенных объектов очереди.

Значение K_T с указанной поправкой будет таким (формула 50 - Калибр Трубы с учетом весовой поправки и объектов последовательности):

$$K_T = S + K_T = S + (\bar{Z}_1 - \bar{Z}) * \sum_{i=1}^R (K_i) - \sum_{i=1}^R \sum_{j=1}^n (w_{ji} * C_j)$$

Независимо от учета ослабления, использование знака плюс или минус для S в формуле (см. формула 50) увеличит или уменьшит вес аналогичных частотных объектов Калибра Трубы, что следует учитывать в дальнейших рассуждениях.

Как и Труба, Калибр может быть построен для Трубы будущего и для Трубы прошлого, соответственно будем различать Калибр Трубы прошлого и Калибр Трубы будущего или просто Калибр будущего (K_T) и Калибр прошлого (K_{-T}). Как и в отношении последовательностей Кластеров, в отношении последовательностей Калибров также может быть определена операция свертки с образованием синтетических объектов.

4.2.3. Смысл поправки ΔW . Остаток Калибра.

Исследуем значение "весовой поправки" ΔW неограниченно добавляя новые объекты C_1 и не удаляя старые объекты последовательности C_R . Для этого вычислим значение Калибра Трубы без учета "весовой поправки" (см. формула 47) и посмотрим, как будет меняться Калибр Трубы.

Представим себе, что мы начали вводить последовательность, которая возможно уже содержалась в памяти последовательностей. Мы вводим первый объект последовательности и построив для него Калибр Трубы находим в памяти тысячи последовательностей, которые могут быть продолжением введенного объекта. Затем мы вводим второй объект и построив Калибр Трубы для двух введенных объектов обнаруживаем, что множество объектов Калибра Трубы уменьшилось, как и число последовательностей, которые содержатся в памяти последовательностей и могут быть продолжением введенных двух объектов. То есть увеличение числа введенных объектов приведет к уменьшению в Калибре Трубы числа последовательностей, которые могли быть продолжением введенного фрагмента. Продолжая вводить новые объекты фрагмента в какой-то момент мы получим Калибр Трубы, содержащий только копию вводимой последовательности, а далее получим массив ΔW частотных объектов Калибра Трубы, не объединенных принадлежностью к хотя бы одной хранимой в памяти последовательности, но состоящий из частотных объектов, содержащихся в Кластере каждого из объектов последовательности. Можно предположить, что такое множество объектов, не являясь множеством гипотез, тем не менее характеризует контекст введенной последовательности. Станем называть этот массив Остатком Калибра. Дальнейший ввод объектов вводимой последовательности при сохранении ее контекста, должен приводить к увеличению суммарного веса частотных объектов Остатка Калибра. Рост суммарного веса Остатка Калибра должен продолжаться до тех пор, пока не изменится контекст последовательности - например пока не будет изменена тема изложения, скажем с животноводства на добычу нефти.

Сделаем оценку возможной длины последовательности, на которой возникает Остаток Калибра. В повседневном общении люди используют словарь, содержащий от 2 до 10 тысяч слов, поэтому "достаточно длинной" можно считать последовательность из 5-10 слов. Вероятность повторного ввода такой последовательности обратно пропорционально числу размещений с повторениями (строго говоря вероятность будет существенно выше, поскольку последовательности не являются случайным набором объектов, следование объектов друг за другом и их сочетаемость подчинены определенным правилам) из 10000 по 5 слов или из 10000 по 10 слов. Длина в 5-10 слов примерно соответствует средней длине предложения в русском языке - около 10 слов. Значит Остаток Калибра может возникать на предложении средней длины.

Замечание 4.

Благодаря вычитанию "весовой поправки" ΔW (формула 49), в случае бесконечного увеличения

размера R Окна Внимания (то есть с добавлением нового объекта C_1 , "старый" объект C_R из Окна Внимания не удаляется), можно ожидать, что Калибр Трубы сойдется к копии вводимой последовательности или к пустому множеству, если память не содержит копий вводимой последовательности. При постоянном значении размера R Окна Внимания (очередь: ввод нового объекта C_1 сопровождается удалением самого "старого" объекта C_R), в каждом цикле Калибр Трубы будет строиться на обновленной очереди объектов последовательности, что должно приводить к циклическому изменению суммарного веса всех объектов Калибра, за счет изменения контекста по мере смещения Окна Внимания вдоль вводимой последовательности.

Следует заметить, поиск Контекста (пикового суммарного значения веса объектов Калибра Трубы) можно заменить поиском пауз и прерываний, которые должны останавливать рост суммарного значения веса объектов Трубы. Это может упростить программную и аппаратную реализацию синтеза новых объектов (формула 51 - Труба последовательности):

$$T = \sum_{i=1}^R (f(r) * K_i)$$

4.2.4. Обратная проекция Трубы.

В заявленном способе для каждого из частотных Объектов конкретного рангового или полного множества из памяти извлекают ранговое множество, для которого названный частотный Объект является ключевым Объектом, извлеченные ранговые множества одинакового ранга сравнивают с целью определения по меньшей мере одного Объекта - Гипотезы.

Как отмечалось выше [3.2.10], с помощью Обратной Проекции любой Кластер можно спроецировать на один или более объектов-Родителей такого Кластера. Поэтому Обратная Проекция Трубы должна породить множество объектов потенциальных Родителей Трубы. В последнем случае, надо убедиться, что профиль весов в Кластере каждого существующего объекта-Родителя с некоторой допустимой точностью совпадает с профилем весов частотных объектов Трубы. Если точность совпадения профилей Кластера и Трубы достаточная, то Труба соответствует уже существующему уникальному объекту, который можно считать Родителем Трубы.

Можно ожидать, что при неизменности тематики (контекста) последовательности не будет меняться и значение Обратной Проекции Трубы. Поэтому каждый раз при вводе нового объекта последовательности можно рассчитывать новое значение Обратной Проекции Трубы сравнивая его со значением Обратной Проекции Трубы при вводе предыдущего объекта последовательности. Быстрое изменение Обратной Проекции Трубы должно совпасть с прохождением пика значений суммарных весов Трубы.

4.2.5. Выбор объекта для обозначения Трубы.

В качестве "лучшего" Родителя Трубы должен быть выбран тот из объектов-Родителей, порожденных Обратной Проекцией Трубы, величина допустимой погрешности для которого ΔK_{\max} удовлетворяет условию (формула 6) и минимальна среди всех потенциальных объектов - Родителей Обратной Проекции Трубы.

Если среди потенциальных объектов-Родителей не нашлось объекта, удовлетворяющего условию (формула 6), то принимается решение синтезировать новый уникальный объект, поставив ему в соответствие Кластер Трубы. В результате получаем множество синтетических объектов, отображением которых является соответствующий Кластер Трубы.

4.2.6. Последовательность Труб и их идентификация.

В заявленном способе очередность создания последовательных Калибров Трубы, содержащих частотные объекты Памяти Последовательностей текущего уровня иерархии (далее "уровень иерархии M1") сохраняют в Памяти Последовательности как последовательность Синтетических Объектов Памяти Последовательностей более высокого уровня иерархии (далее "уровень иерархии M2").

Каждое текущее множество Калибра Трубы при этом связывают с Синтетическим Объектом (далее "Частотный Синтетический Объект"), которому поставлено в соответствие множество Калибра Трубы, предшествующее текущему в последовательности Калибров Трубы, путем размещения в названном текущем множестве Калибра Трубы весового коэффициента взаимной встречаемости Синтетического Объекта (далее "Ключевой Синтетический Объект"), поставленного в соответствие текущему множеству Калибра Трубы, с названным Частотным Синтетическим Объектом.

Последовательность Синтетических Объектов вводят в Память Последовательностей в качестве одного из машиночитаемых массивов данных Памяти Последовательностей уровня иерархии M2, являющихся последовательностью множества уникальных Объектов.

Приведем примеры алгоритмов создания последовательности Труб для вводимой последовательности уровня иерархии M1 ПП и их идентификации существующими или синтезированными объектами на следующем уровне иерархии M2 ПП. Тем не менее, в рамках предложенного подхода создания Труб и их идентификации, использования различных техник и методов сравнения Кластеров, алгоритм может быть и иным.

1. Вводим в Окно Внимания очередной объект последовательности.
2. Строим Трубу для Окна Внимания, строится Труба и сохраняем ее значение.

3. Сравниваем суммарный вес частотных объектов построенной Трубы (текущий вес) с суммарным весом частотных объектов Трубы, полученной на предыдущем шаге (предыдущий вес). Если текущий вес больше предыдущего, то запоминаем текущий вес в качестве предыдущего и продолжаем ввод последовательности. Если текущий вес меньше предыдущего, то

- a. Нормализуем веса частотных объектов полученной Трубы [2.3.5.2].
- b. Определяем множество объектов Родителей полученной Трубы путем расчета Обратной Проекции полученной Трубы.
- c. Нормализуем веса частотных объектов в Кластерах объектов-Родителей.
- d. Сравниваем нормализованный профиль весов Трубы с нормализованными профилями Кластеров объектов-Родителей.
- e. Выбираем из объектов-Родителей объект с минимальным значением $\Delta\bar{K}$ удовлетворяющим условию $\Delta\bar{K}_{\max} \geq \Delta\bar{K}$ (формула 6).
- f. Если в Обратной Проекции отсутствует объект, удовлетворяющий описанным условиям, то синтезируем новый объект и сопоставляем ему Кластер Трубы в Памяти Последовательностей.

Или так:

1. Вводим в Окно Внимания очередной объект последовательности.
2. Строим Трубу для Окна Внимания и запоминаем ее значение.
3. Строим Обратную Проекцию Трубы и запоминаем ее значение.
4. Сравниваем значение Обратной Проекции Трубы с ее предыдущим значением и если изменения не превышают допустимых, то продолжаем ввод последовательности, а если превышают, то сравниваем нормализованный профиль весов частотных объектов Трубы с нормализованными профилями частотных объектов Кластеров объектов Обратной Проекции и выбираем объект с минимальным значением $\Delta\bar{K}$ удовлетворяющим условию $\Delta\bar{K}_{\max} \geq \Delta\bar{K}$ (формула 6).

5. Если в Обратной Проекции отсутствует объект, удовлетворяющий описанным условиям, то синтезируем новый объект и сопоставляем ему Кластер Трубы в Памяти Последовательностей.

Понятно, что приведенные алгоритмы лишь служат примерами, в то время как человек, обладающий нужными знаниями, может предложить и другие алгоритмы, в рамках подхода изложенного в настоящей работе.

4.2.7. Роль Окна Внимания для Памяти Последовательностей.

В заявленном способе ввод последовательностей в память осуществляют, как правило, циклами, причем на каждом цикле в память вводят очередь объектов последовательности (далее "Окно Внимания"), а при переходе к следующему циклу очередь объектов увеличивают или сдвигают по меньшей мере на один объект в будущее или прошлое.

Во вводной части мы давали довольно простое определение Окну Внимания и теперь детализируем и углубим это определение.

В Памяти Последовательностей каждому объекту соответствует Кластер, а каждому Кластеру может соответствовать объект. Представим это несколько по-другому.

Предположим, у нас есть два устройства Измельчитель яблок и Восстановитель яблок. Предположим также, что если на вход Измельчителя подать яблоко, то на выходе из Измельчителя получим яблочное пюре и если подать яблочное пюре на вход Восстановителя, то на выходе из Восстановителя получим исходное яблоко, но на 10% меньшего размера.

Соединим теперь Выход Измельчителя с входом Восстановителя (прямая связь), а выход Восстановителя с входом Измельчителя (обратная связь), возьмем большой мешок с яблоками одинакового размера и начнем их подавать на вход Измельчителя в тот момент, когда предыдущие уменьшенные на 10% яблоки вернулись из Восстановителя в Измельчитель.

Что мы заметим? Мы положили в Измельчитель первое яблоко исходного размера, и оно вернулось уменьшенным на 10%. Теперь на вход Измельчителя мы подадим уже два яблока - новое стандартного размера и старое на 10% меньше. На втором цикле работы системы из Восстановителя выйдут уже два яблока - одно уменьшенное на 10% и другое уменьшенное на 19% и добавив к ним яблоко стандартного размера мы запустим уже эти три яблока в цикл снова и так далее каждый раз на вход Измельчителя мы будем подавать все более длинную очередь яблок разного размера. Мы можем восстановить порядок подачи яблок в систему, ранжируя яблоки по размеру. Это и есть физический смысл Окна Внимания: создавать рекуррентную связь между объектами и наследовать порядок объектов исходной последовательности. Несмотря на то что "Восстановитель" мы ранее не описывали, мы это сделаем позже.

Ранее мы говорили об Окне Внимания постоянной длины, однако приведенный в примере рекуррентный механизм подачи яблок позволяет всегда подавать на вход в систему всего одно новое яблоко и в тоже время сохранять в системе очередь всех яблок "Окна Внимания". Такой подход позволяет Памяти Последовательностей преодолеть ограничение на длину Окна Внимания и использовать динамическое Окно Внимания изменяемой длины.

Размер Окна Внимания не может увеличиваться бесконечно и нам следует определить условия, при

которых размер Окна Внимания перестанет увеличиваться или текущее Окно Внимания будет аннулировано. Ранее в качестве условий ограничивающих использование текущего Окна Внимания были: 1) достижение максимального значения суммарного веса частотных объектов Калибра Трубы и 2) ввод прерывания последовательности. В первом случае мы сравниваем два последовательных измерения суммарного веса Калибра Трубы и, если последняя сумма меньше предыдущей, то считаем что при предыдущем измерении был достигнут максимум суммарного веса Калибра Трубы и контекст последовательности поменялся. Во втором случае прерыванием, в частности, является ввод пустого объекта, что приводит к равенству двух последовательных измерений суммарного веса Трубы, то есть суммарный вес Трубы в двух последовательных измерениях не менялся. Поэтому и первое, и второе условия связаны с изменением суммарного веса Трубы или Калибра Трубы. Хотя приведенные условия кажутся справедливыми для текстовой информации и языка в целом, последовательности объектов другой природы могут потребовать исполнения условий, которые заранее неизвестны и могут отличаться от перечисленных, но можно заметить, что эти условия вероятно должны быть связаны с измерением суммарного контекста, потому что именно контекст является инвариантным паттерном вводимой последовательности любой природы. Таким образом использование Сумматора для искусственных нейронов является единственным решением, но условия функции активации нейрона для последовательностей разной природы и возможно для разных случаев последовательностей объектов одинаковой природы могут, тем не менее, отличаться.

Кроме того, следует договориться и о том, какой именно смысл несет выражение "Окно Внимания перестанет увеличиваться", например: 1) при наступлении условий смены контекста текущее "динамическое Окно Внимания" аннулируется и заменяется новым "динамическим Окном Внимания", в котором следующий объект станет единственным объектом, после чего "динамическое Окно Внимания" до следующего наступления условий смены контекста или 2) при некоторых условиях мы заканчиваем "фазу роста" Окна Внимания и фиксируем текущую длину "динамического Окна Внимания", а далее изменяем его по принципу очереди (FIFO) как "статическое Окно Внимания", добавляя в Окно новый (самый поздний) объект и выбрасывая самый ранний объект Окна или 3) если условия смены контекста не наступают, то при достижении некоторой максимальной длины "динамического Окна Внимания" мы фиксируем его длину и далее изменяем Окно Внимания по принципу очереди (FIFO) как "статическое Окно Внимания", добавляя в Окно новый (самый поздний) объект и выбрасывая самый ранний объект Окна.

Третий вариант изменения Окна Внимания представляется наиболее разумным. Таким образом, "фаза роста" "динамического Окна Внимания" прекращается, если или 1) наступили условия смены контекста или 2) введено прерывание или 3) достигнута максимальная длина динамического Окна Внимания, причем в последнем случае динамическое Окно Внимания становится "стационарным" и меняется как очередь до наступления условий смены контекста или до ввода прерывания.

5. Иерархическая память.

5.1. Иерархия устойчивых сочетаний.

5.1.1. Память устойчивых сочетаний.

Как отмечалось (замечание 1) поиск устойчивых сочетаний на множестве из N объектов имеет трудоемкость N^R . Покажем, как использование Памяти Последовательностей снижает трудоемкость определения устойчивых сочетаний и делает процесс простым.

Станем каждой паре последовательных объектов присваивать идентификатор нового объекта. В случае со словосочетанием "Организация Объединенных Наций" это приведет к образованию двух новых искусственных объектов C_1 ="Организация Объединенных" и C_2 ="Объединенных Наций". Всякий раз, когда объекты C_1 и C_2 будут встречаться вместе вес их связи первого ранга в Памяти Последовательностей будет увеличиваться, то есть в Кластере 1-го ранга объекта C_1 объект C_2 будет иметь вероятно или максимальный вес или один из максимальных весов, что является "подсказкой" для его преимущественного использования после объекта C_1 .

Таким образом, кодируя каждую пару объектов последовательности новым синтетическим объектом мы решили задачу с трудоемкостью N^2 , поэтому назовем множество вновь созданных объектов "слоем n_2 " или "объектами n_2 ".

Аналогичным образом мы перейдем к поиску устойчивых сочетаний объектов слоя n_2 создав множество новых объектов слоя n_3 , затем слоя n_4 и так далее.

5.1.2. Сочетания 1-го ранга.

Устойчивые сочетания в каждом слое n_2 , n_3 , n_4 и так далее могут быть только первого ранга и это упрощает работу с ними. Каждому искусственному идентификатору слоя n_2 ставим в соответствие направленную связь пары объектов последовательностей n_1 , Каждому искусственному идентификатору слоя n_3 ставим в соответствие направленную связь пары объектов последовательностей n_2 , и так далее (фиг. 27) (формула 52 - искусственные объекты для сочетаний объектов нижнего слоя):

$$C_1^{n_2} = \{C_1^{n_1} \rightarrow C_2^{n_1}\}$$

$$C_1^{n_3} = \{C_1^{n_2} \rightarrow C_2^{n_2}\}$$

Теперь Кластер каждого объекта слоя n_1 можно представить множеством объектов слоя n_2 (формула 53 - Кластер объекта слоя n_1 , содержащий частотные объекты слоя n_2):

$$K(C_1^{n2}) = \{w_a * C_a^{n2}; w_b * C_b^{n2}; w_c * C_c^{n2}; \dots, w_x * C_x^{n2};\}$$

Выбирая из множества Кластера (формула 53) самые весомые w_i связи $n2$, а затем строя гипотезу появления следующего объекта такой связи, зная, какой паре объектов такая связь соответствует (формула 52).

Формирование гипотез показано пунктирными стрелками (фиг. 27). Для того чтобы предсказать появление четвертого объекта последовательности (объект C_4^{n1}), находим в слое $n2$ объект с наибольшим весом C_3^{n2} и строим гипотезу о появлении объекта C_4^{n1} последовательности. Аналогичным образом строим гипотезу появления объекта C_4^{n2} и с его помощью гипотезу появления объекта C_5^{n1} . Наличие нескольких уровней $n2$, $n3$, $n4$ и так далее позволяет прогнозировать появление объектов, расположенных в далеком будущем.

Те устойчивые сочетания объектов, которые не были "забыты" в процесс "чистки" будут возвращать "подсказку" каким должен быть следующий объект последовательности.

Поскольку объекты слоя $n2$ по сути являются связью существующих объектов $C_{n1}^{n2} = \{C_{n1}^{n1} \rightarrow C_{n2}^{n1}\}$ слоя $n1$, то для идентификации объектов слоя $n2$ можно использовать идентификатор объекта слоя $n1$ на который направлена связь в паре этих объектов: $C_{n1}^{n2} = C_{n2}^{n1}$.

Для слоя $n3$ аналогично имеем

$$\begin{cases} C_1^{n3} = \{C_1^{n2} \rightarrow C_2^{n2}\} \\ C_1^{n2} = C_2^{n1} \\ C_2^{n2} = C_3^{n1} \end{cases}$$

И тогда

$$C_1^{n3} = \{C_2^{n1} \rightarrow C_3^{n1}\}$$

Поэтому можно считать, что

$$C_1^{n3} = C_3^{n1}$$

И так далее.

Замечание 5.

Как видим объект $n2$ это связь 1-го ранга между объектами $C_{n1}^{n1} \rightarrow C_{n2}^{n1}$, объекты $n3$ это связь 1-го ранга между объектами $C_{n2}^{n1} \rightarrow C_{n3}^{n1}$ и так далее по цепи будущих объектов последовательности. Поэтому создание идентификаторов для сочетаний можно избежать, а вместо этого использовать существующие идентификаторы объектов и их Кластеры 1-го ранга.

Алгоритм прогнозирования сводится к выполнению следующих шагов:

1. Строим Кластер 1-го ранга для известного ключевого объекта,
2. Ищем в построенном Кластере частотные объекты с самым высоким весом совместной встречаемости и считаем один или более найденных объектов гипотезами продолжения последовательности;

Для найденных гипотез повторяем шаги 1 и 2 пока не достигнем нужной глубины прогнозирования.

Каким смыслом будет обладать приведенный алгоритм, если вместо Кластера 1-го ранга мы станем использовать Кластер 2-го или более высокого ранга?

5.1.3. Устойчивые сочетания высокого ранга.

В заявленном способе сравнивают ранговые множества разных рангов (далее "Когерентные множества") для известных ключевых Объектов последовательности, причем ранг рангового множества для каждого ключевого Объекта выбирают соответствующим числом Объектов последовательности, разделяющих названный ключевой Объект и Объект-Гипотезу (далее "Фокальный Объект когерентных множеств"), возможность появления которого проверяют.

Если использование Кластера 1-го ранга позволяло нам прогнозировать появление следующего объекта, то использование Кластера 2-го ранга позволит прогнозировать будущие объекты последовательности, разделенные одним неизвестным объектом. А использование Кластеров 3-го ранга позволит прогнозировать появление каждого третьего объекта последовательности, разделенные двумя неизвестными объектами. И так далее...

Таким образом, исполнение трех шагов алгоритма с использованием Кластера 1-го ранга дает последовательность-прогноз длиной 3 объекта; исполнение трех шагов алгоритма с использованием Кластера 2-го ранга дает последовательность-прогноз длиной 3 объекта и с глубиной прогнозирования 6 объектов так далее ... исполнение k шагов алгоритма с использованием Кластера n -го ранга даст последовательность-прогноз из k объектов с глубиной прогнозирования ($n \times k$) объектов последовательности.

Каждая из полученных последовательностей-прогнозов является Окном Внимания одной длины, но с разной глубиной прогнозирования и это позволяет сравнивать прогнозы разной глубины, сравнивая полные Кластеры каждого из Окон Внимания с Кластером Трубы последовательности. Если погрешность $\Delta \bar{K}$ (формула 6) между контекстом прогноза (Кластером соответствующей глубины прогнозирования) и текущим контекстом последовательности (Кластером Трубы последовательности) не превышает максимального $\Delta \bar{K}_{\max} > \Delta \bar{K}$, то можно сделать вывод о соответствии контекста прогноза текущему контексту последовательности, а если $\Delta \bar{K} > \Delta \bar{K}_{\max}$, то контекст прогноза отличается от текущего контекста последовательности, что может говорить об ошибке прогнозирования.

5.1.4. "Чистка" памяти устойчивых сочетаний.

В повседневной жизни люди пользуются словарем из 2000-10000 слов. Для множества из 10000 слов максимально возможным является сто миллионов сочетаний из двух слов (10000×10000) и, хотя не любые сочетания возможны, наличие механизма "забывания" редких сочетаний кажется необходимым для того, чтобы избежать переполнения памяти ненужными сочетаниями.

Механизм "забывания" можно реализовать многими способами. Предпочтительным способом может служить способ, при котором "забывается" та часть сочетаний, которые входят в X% (например, 5, 10, 20, 30 и так далее процентов) с наименьшим весом совместной встречаемости среди сочетаний слоя. Таким образом, "чистка" памяти будет предотвращать ее переполнение, при этом из памяти будут удаляться "слабые", а "сильные" сочетания будут оставаться.

5.2. Иерархия Труб.

5.2.1. Связность Труб.

Являясь следующим слоем иерархии Памяти Последовательностей, Трубы устанавливают связь между конкретным отрезком последовательности (Генератор Трубы) который представлен суммой Кластеров, порождаемых объектами этого отрезка последовательности, с одной стороны, и следующим слоем иерархии последовательностей Памяти Последовательностей, в которой членами последовательности являются Трубы.

Связывать объекты в следующем слое иерархии следует также, как они были связаны в предыдущем - в слое вводимых последовательностей, а именно объекты, следующие за ключевым, должны присутствовать в Кластере будущего этого ключевого объекта. Каждому Кластеру Трубы можно сопоставить идентификатор Трубы поскольку в Памяти Последовательностей каждому объекту соответствует Кластер, а каждому Кластеру должен соответствовать объект.

Обозначим Кластер Трубы T2 объектом T2 который выглядит как маленький темный объект в Кластере Трубы T1 (фиг. 28).

Поскольку Кластеры Труб появляются при операции суммирования Кластеров объектов, то речь идет о добавлении идентификатора Трубы T1 в Кластеры объектов последовательности в качестве обратной связи с предыдущей Трубой. Сказанное можно проиллюстрировать более аккуратным рисунком (фиг. 29).

Кластеры объектов 5 и 4 породили Трубу T2 которой был присвоен идентификатор T2 показанный кружком. Этот идентификатор объекта T2 затем был унаследован Кластерами объектов 3, 2 и 1 (был в них добавлен) и в результате этого объект T2 оказался в Кластере Трубы T1, а сама Труба T1 была представлена в виде объекта T1. Таким образом между объектами T1 и T2 была создана связь. Продолжая процесс порождения новых Труб последовательности и добавляя в Кластер новой Трубы идентификатор предыдущей Трубы мы получим последовательность идентификаторов Труб, которые порождают Кластеры Труб, содержащие идентификатор предыдущей Трубы.

Несмотря на то что добавление идентификатора Трубы в Кластер следующей Трубы кажется искусственным шагом, он вполне соответствует логике построения Кластеров будущего и прошлого, в которые попадают объекты, следующие за текущим или предшествующие ему в последовательности объектов. В рамках такой логики можно сказать, что мы создали обратную связь между двумя Трубами следующими одна за другой в последовательности Труб, хотя можно было вместо этого создать упреждающую связь путем размещения идентификатора T1 в Кластере предыдущей Трубы T2, что сути, конечно, не меняет: мы создали механизм представления отдельных Труб в виде связной последовательности идентификаторов этих Труб.

Имея в виду то, что каждой из Труб, как и другим объектам Памяти Последовательностей мы присваиваем идентификатор, то как и для других объектов нумерованной памяти последовательностей хит Трубы должен содержать фрагмент (формула 18) последовательности - Генератор Трубы, над которым построена Труба.

Замечание б.

Идентификатору Трубы следует ставить в соответствие не только Кластер Трубы, но и Генератор Трубы, иначе модель Трубы окажется неполной.

Создание связности в слое Труб позволяет:

восстановить параллельные последовательности, то есть отрезки последовательностей, которые могли служить Генераторами Кластера Трубы, обладающими одинаковым контекстом (смыслом);

восстановить полную последовательность объектов из ее отрезков используя последовательность идентификаторов Труб и их Генераторы;

связанная последовательность идентификаторов Труб с соответствующими каждой Трубе Кластерами, сама является последовательностью полносвязной Памяти Последовательностей, причем отрезки этой последовательности Труб также можно свернуть в Трубы, создав тем самым следующий более компактный слой иерархии Памяти Последовательностей.

5.2.2. Слои Труб.

Как уже отмечалось, исходную последовательность объектов можно представить в виде последовательности синтетических объектов - Труб. Строить Трубы можно как используя принцип максимизации

суммарного веса частотных объектов Кластера Трубы, или разбивая последовательность на смежные участки, например разбивая текст на предложения, или разбивая последовательность на равные по длине отрезки или иным способом. А последовательность Труб можно представить последовательностью их идентификаторов и свернуть их в Трубу следующего уровня.

Повторяя этот процесс на разных уровнях иерархии, получим множество слоев Труб над слоем объектов исходной последовательности.

Определение 6.

Трубы, построенные над последовательностью объектов, станем называть Трубами 1-го рода, а Трубы, построенные над последовательностью Труб 1-го рода, станем называть Трубами 2-го рода и так далее до Труб k-го рода.

Это создает мощный механизм смыслового и временного уплотнения исходных последовательностей объектов в более компактные синтетические образования.

Сворачивание разных слоев иерархии в Трубы позволяет производить многократное смысловое сжатие последовательности на разных уровнях иерархии, создает обратную-упреждающую связь с накопленным опытом и служит основой механизма воспоминаний, а также производства умозаключений и выводов (фиг. 30).

5.2.3. Смежные Трубы.

Следующие друг за другом Трубы назовем смежными. Смежные Трубы строятся для смежных участков последовательности или Смежных Окон Внимания.

Разбивать последовательность на участки можно разными способами. Например:

1. Произвольно разбить последовательность на участки, длина которых продиктована техническими ограничениями, например числом ножек микросхемы, на которые вводятся сигналы объектов Окон Внимания или другими особенностями технической реализации, ограничивающими ввод Окон Внимания.
2. Разбить последовательность на отрезки, разделенные паузами ввода или прерываниями [2.2.8].
3. Разбить исходную последовательность на отрезки, определенные условием максимального суммарного веса частотных объектов Трубы (формула 39) и так далее.

Следует отметить, что при неизменности контекста последовательности, произвольное разбиение не может уменьшить рост суммарного веса частотных объектов Трубы (формула 39), а ввод пауз или прерываний способен лишь приостановить этот рост. Таким образом, разбивая исходную последовательность объектов на отрезки произвольной длины или разделенные прерываниями, а затем строя Трубы для каждого из участков, мы получаем последовательность Труб 1-го рода, для которой можно построить последовательность Труб 2-го рода и уже ее исследовать на выполнение условия достижения максимального суммарного веса частотных объектов Трубы (формула 39). Однако для сворачивания последовательности Труб 2-го рода в Трубы 3-го рода и последовательность Труб 2-го рода также как и исходная последовательность объектов может быть разбита на смежные участки описанными выше способами, а уже последовательность Труб 3-го рода может исследоваться на выполнение условия достижения максимального суммарного веса частотных объектов Трубы 4-го рода (формула 39). И так далее.

Поэтому самым простым способом разбиения исходной последовательности на смежные участки является разбиение на участки, разделенные паузами, а если паузы долго нет, то на отрезки некоторой максимальной длины, определенной техническими ограничениями системы ввода.

6. Память эмоций.

Природа эмоций хорошо описана в книге Алексея Редозубова "Логика эмоций" [<http://www.aboutbrain.ru/wp-content/plugins/download-monitor/download.php?id=6>]. Алексей Редозубов отмечает, что существенную роль в планировании и выборе решений имеет эмоциональная память, а именно оценка по шкале "плохо-хорошо", которую человек присваивает всему что с ним происходит. Эмоции и рефлексы имеют одну и ту же природу:

оценка ощущениями - состояние "хорошо-плохо" связанное непосредственно с ощущениями, с тем, как мы рефлекторно реагируем на сигналы рецепторов;

оценка эмоциями - состояние "хорошо-плохо", следующее из того, как оценил смысл происходящего наш мозг.

Таким образом, можно сказать, что эмоции являются виртуальной суммарной оценкой физических ощущений человека.

В своих дальнейших рассуждениях Алексей Редозубов показывает, что в ситуации выбора, планирования или принятия решений человек строит гипотезы на тему различных сценариев своего поведения (возможных последовательностей своих действий и последствий таких действий) и получает из памяти рефлекторную и эмоциональную оценку таких сценариев, а полученные оценки для разных сценариев сравнивает между собой и делает выбор в пользу эмоционально более предпочтительного сценария. Так ситуация выбора, известная как "парадокс буриданова осла" вынуждает мозг осла фантазировать на тему того, как он поедает копну сена А или копну сена Б и сравнивает эмоциональный отклик этих фантазий - и победит та, которая окажется эмоционально более привлекательной по шкале "плохо-хорошо". Одна из копен сена может оказаться предпочтительной, например, если в одной из копен осел заметил пучок лю-

бимой травы или, например, если в силу каких-то обстоятельств более ранний опыт выбора правой или левой копны сена был эмоционально более приятным или наоборот неприятным.

Память эмоций является одним из каналов многоканальной памяти последовательностей (см. раздел "Многopotочная память последовательностей"). Эмоции являются последовательностями сигналов чувств и ощущений из спектра "хорошо-плохо", а источником сигналов являются сигналы нервной системы и опыт, накопленный интеллектом и связанный с памятью, полученной по каналам других чувств и ощущений. Так любая последовательность действий, которая привела человека к ожогу руки от прикосновения к раскаленной сковородке, получит в его памяти эмоциональную оценку "плохо", а сценарий с возможным прикосновением к сковородке будет воспроизводить виртуальное ощущение ожога руки, напоминая о негативном опыте, полученном ранее. В результате, планируя прикосновение к сковородке, человек построит в памяти виртуальную последовательность своих действий в каждом из сценариев, построит гипотезы возможных результатов таких действий, извлечет из памяти эмоциональную оценку таких возможных результатов и станет выбирать между разными сценариями прикосновения к сковородке, основываясь на суммарной эмоциональной оценке каждого из сценариев. В результате в памяти человека победит тот сценарий, в котором перед прикосновением к сковородке человек пытается определить, не является ли сковородка горячей, при этом, не касаясь самой сковороды или человек просто откажется от мысли потрогать сковороду.

Память эмоций может быть представлена множеством объектов, представляющих дисперсные значения эмоциональной и этической оценки "плохо-хорошо". Во время обучения эти объекты эмоциональной оценки следует добавлять в Кластер Трубы наряду с объектами вводимой последовательности во время обучения Памяти Последовательностей. При воспроизведении (считывании) последовательности из Памяти объекты последовательности будут воспроизводиться вместе с объектами эмоциональной оценки, и мы станем использовать объекты эмоциональной оценки для ранжирования решений по шкале "плохо-хорошо" и принятия эмоционально приемлемых решений.

7. Аппаратная реализация Памяти Последовательностей (ПП).

Указанный технический результат для объекта "ПП" достигается за счет того, что в ПП, содержащей два соединенных между собой множества N параллельных пронумерованных шин, из которых первое множество расположено над вторым множеством так, что шины первого и второго множества образуют в плане множество пересечений вида "матрица", причем концы каждого множества N параллельных шин, расположенные с одной из сторон "матрицы", используются в качестве входов, а противоположные им концы - в качестве выходов так, что сигналы, поданные на входы первого множества N параллельных пронумерованных шин, считываются как с выходов первого множества N параллельных пронумерованных шин, так и с выходов второго множества N параллельных пронумерованных шин при наличии в пересечении шин первого и второго множества Элементов Коммутации шин первого и второго множеств, угол пересечения β° между шинами первого и второго множеств выбирается, исходя из функциональных и геометрических требований к устройству памяти, причем шины первого и второго множеств с одинаковыми номерами соединены друг с другом в месте их пересечения так, что множество таких соединений образует диагональ матрицы, разделяя матрицу на две симметричные треугольные полуматрицы (далее "Косынки"), по меньшей мере одна из которых (далее "Первая Косынка") используется путем соединения по меньшей мере одним Искусственным Нейроном Встречаемости (ИНВ) каждой двух шин, по меньшей мере, с несопадающими номерами по одной из первого и второго множества в местах их пересечения так, что концы шин первого множества являются входами и концы второго множества являются выходами Косынки, а ИНВ используется в качестве названного Элемента Коммутации для накопления, хранения и считывания веса совместной встречаемости объектов, которым соответствуют шины, соединенные названным ИНВ; каждый из названных ИНВ функционирует, по меньшей мере, как Счетчик с функцией активации и ячейкой памяти для хранения последнего значения и значения порога активации ИНВ; перед началом эксплуатации устройства последнему значению присваивается некоторое исходное значение, которое сохраняется в ячейке памяти Счетчика, в ячейке памяти также сохраняется значение порога активации ИНВ; в режиме обучения каждый раз, когда сигналы поданы одновременно на каждую из шин соединенных посредством ИНВ, названный ИНВ измеряет одну из характеристик сигнала на каждой их шин, затем сравнивает измеренные значения характеристик и, если результат сравнения соответствует значению порога активации ИНВ, ИНВ считывает последнее значение из ячейки памяти, увеличивает названное последнее значение на величину изменения встречаемости и сохраняет новое последнее значение в ячейке памяти, а в режиме воспроизведения сигнал подается по меньшей мере на одну из названных шин, соединенных посредством ИНВ, сигнал проводится через ИНВ, где из ячейки памяти извлекается последнее значение, одна из характеристик сигнала изменяется соответственно извлеченному последнему значению и названный измененный сигнал передается во вторую из названных шин, соединенных посредством ИНВ, для извлечения названного последнего значения из названной одной из характеристик сигнала и использования названного последнего значения в качестве веса совместной встречаемости объектов, которым соответствуют шины.

7.1. Устройство Памяти Последовательностей (ПП).

В работе над архитектурой будем исходить из того, что имеем множество M уникальных объектов

S последовательностей, каждый из которых может иметь связь с любым другим объектом множества M. Каждый объект в архитектуре представлен входом S шины C (фиг. 31). Ввод объекта последовательности в ПП осуществляется подачей сигнала S на вход шины C.

Таким образом, физически объект C представлен сигналом S, который подается на вход шины C Памяти Последовательностей (фиг. 32), что должно приводить к порождению множества весовых коэффициентов $W=\{w_1, w_2, \dots, w_n\}$, соответствующих частотных объектов Кластера $K=\{w_1 \times C_1, w_2 \times C_2, \dots, w_n \times C_n\}$.

7.1.1. Косынка.

Как и прежде будем называть связями первого ранга связи между двумя соседними объектами последовательности; связями второго ранга - связь между объектами последовательности разделенными одним объектом; ...; связью N-го ранга назовем связь между объектами последовательности разделенными (N-1) объектами (см. фиг. 33).

Рассмотрим матричную архитектуру Памяти Последовательностей вида crossbar (см. фиг. 34)) с множеством шин $C=\{C_1, C_2, C_3, \dots, C_n\}$, которая обеспечивает связь шин между собой между собой "каждый с каждым" размером $N \times N$, то есть матрица шин является полносвязной.

Показанные на рисунке (фиг. 34) в вертикальном и горизонтальном рядах шины C_1, C_2, C_3, C_4 - это шины одних и тех же объектов S множества уникальных объектов Памяти Последовательностей $M=(S1, S2, S3, S4)$. Недостатком геометрии crossbar (фиг. 34) является то, что сигнал объекта приходится одновременно подавать на две его шины - вертикальную и горизонтальную. Чтобы подавать сигнал объекта только на одну шину, необходимо соединить вертикальную и горизонтальную шины каждого из объектов, однако геометрия crossbar препятствует этому, так как расстояние между шинами определяется размером матрицы и чем больше шин в матрице, тем большим числом шин разделены вертикальная и горизонтальная шины объекта и тем сложнее их соединить. Геометрию crossbar можно улучшить, если использовать только половину матрицы crossbar (фиг. 35).

В местах пересечения шин с совпадающими номерами размещены Искусственные Нейроны Встречаемости (ИНВ), каждый из которых содержит по меньшей мере один Счетчик веса взаимной встречаемости, а шины с совпадающими номерами соединены между собой параллельной связью, которая параллельна Нейрону Встречаемости.

Параллельная связь снабжена элементом, который изменяет сигнал при переходе из одной шины с совпадающими номерами в другую, чтобы задать направление считывания для Счетчика названного ИНВ.

На диагонали матрицы находятся N пересечений "сам с собой", а число пересечений "каждый с другими" продублировано за счет симметричных относительно диагонали узлов матрицы, отличающихся лишь очередностью следования объектов $a \rightarrow b$ и $b \rightarrow a$ сверху и снизу от диагонали матрицы. Если "сложить" матрицу по диагонали, то узлы связей $a \rightarrow b$ и $b \rightarrow a$ окажутся друг над другом. Если названные связи $a \rightarrow b$ и $b \rightarrow a$ расположить параллельно друг другу в пересечении шин a и b, а каждую из связей задействовать в зависимости от порядка следования объектов a и b в последовательности $a \rightarrow b$ или $b \rightarrow a$, то это позволит использовать для каждого объекта не две, а одну шину, избавляя матрицу от избыточности и устраняя необходимость коммутации вертикальной и горизонтальной шин одного и того же объекта (фиг. 36). Если на шины объектов a и b одновременно подавать сигналы она из характеристик которого отличается (например, напряжение), то направление $a \rightarrow b$ или $b \rightarrow a$ будет определяться названной разницей (например, разницей потенциалов). Таким образом, для записи или считывания веса связи объектов a и b, на их шины необходимо, например, чтобы в пересечении шин образовалась разница потенциалов, которая соответствует считыванию или записи связи $a \rightarrow b$ или $b \rightarrow a$. Количество одновременного включения обеих шин в режиме обучения запоминается Счетчиком, значение которого каждый раз увеличивается на известную величину совместной встречаемости, предпочтительно на единицу. Поскольку нас может интересовать считывание веса связи объектов в направлении "будущего" и в направлении "прошлого", то этому может соответствовать, например, смена полярности (смена входа и выхода) на шине каждого из объектов.

Поэтому в заявленном ПП в отдельных случаях входы Косынки используются в качестве выходов, а выходы в качестве входов.

Очевидно, что предложенная архитектура позволяет реализовать матрицу "Состояния памяти" (формула 17), а значит "Утверждение 7" справедливо также в отношении предложенной архитектуры и любое состояние памяти последовательностей можно получить с помощью линейных преобразований над весами связей расположенных в пересечениях шин предложенной архитектуры.

В соответствии с заявленным способом ранговое множество уникального объекта является множеством первого ранга и содержит весовые коэффициенты частотных Объектов, непосредственно соседствующих с названным ключевым Объектом в названных последовательностях. Кроме того, в памяти сохраняют ограниченное число ранговых множеств, а массив данных "Будущее" и массив данных "Прошлое" формируют как линейную композицию весовых коэффициентов или ранговых множеств массива данных МСП.

Возможно, для некоторых случаев будет полезно избавиться от пересечений "сам с собой", но в случае наличия связей "сам с собой", они могут быть реализованы с помощью "петли" из шины как показано ниже (фиг. 37), причем "петля" по сути является параллельным соединением шин с одинаковыми, которые уже соединены посредством Счетчика. Таким образом, сигнал из одной из шин проходит в другую шину через такое параллельное соединение и поскольку встречаемость "сам с собой" не имеет направления $a \rightarrow a$, то считать значение встречаемости можно только в одном направлении. Тем не менее, чтобы задать направление прохождения сигнала через Счетчик - сигнал на входе Счетчика отличался от сигнала на выходе Счетчика, в параллельное соединение шин следует встроить элемент, должным образом изменяющий характеристики сигнала при переходе из одной шины в другую.

Станем записывать в матрицу только связи первого ранга для каждого объекта C_n с каждым другим объектом C_k , принадлежащим полному множеству объектов Памяти Последовательностей. Тогда полный Кластер объекта C_n можно считать с косынки рекуррентной связью (фиг. 38):

Шаг 1. Вводим в косынку объект C_n , а на выходе получаем связи первого ранга объекта C_n - множество частотных объектов C_k и веса w_k их связи совместной встречаемости с объектом C_n первого рода.

Шаг 2. Вводим в косынку полученные на предыдущем шаге объекты C_k , а на выходе получаем связи первого ранга каждого из введенных объектов C_k - множество частотных объектов C_j и веса w_j их связи совместной встречаемости с объектом C_k первого рода.

Шаг 3. Повторяем шаг 2, пока не выполним необходимое число циклов ввода и вывода.

Шаг 4. Суммируем веса w_{uz} , полученные на каждом из шагов z для уникального объекта C_u из полного множества объектов Памяти Последовательностей $w_{u\Sigma} = \sum_{z=1}^Z w_{uz}$.

Шаг 5. Представляем полный Кластер объекта C_n как множество $K_n = \{w_{1\Sigma} \times C_1, w_{2\Sigma} \times C_2, w_{3\Sigma} \times C_3, \dots, w_{N\Sigma} \times C_N\}$.

Для учета функции ослабления (формула 10)) перед складыванием весов (шаг 4), вес w_{uz} связи ранга z объекта C_n с объектом C_u , следует умножить на значение функции ослабления $f(r)$ связи соответствующего ранга.

Замечание 7.

В процессе обучения матрицы создаются обратные связи n -го ранга, которые могут считываться как упреждающие при смене направления считывания.

Замечание 8.

Складывая веса, полученные на каждом из циклов ввода для каждого из уникальных объектов C_i полного множества объектов Памяти Последовательности, получим суммарный вес соответствующего объекта C_i в полном Кластере объекта C_n для Окна Внимания, размер которого равен числу циклов ввода. Таким образом, для получения полного Кластера объекта можно использовать только одну косынку Памяти Последовательностей.

Искусственные Нейроны Встречаемости такой единственной косынки накапливают и хранят веса встречаемости для каждого уникального объекта (ключевой объект) со всеми другими уникальными объектами последовательностей (частотные объекты).

Таким образом, для каждого уникального ключевого Объекта в памяти сохраняют по меньшей мере одно ранговое множество или "будущего", или "прошлого" ранг которого одинаков для всех уникальных ключевых Объектов (далее "базовый ранг" множества), причем каждый весовой коэффициент взаимной встречаемости ключевого и частотного Объектов названного рангового множества относится к частотному Объекту, который непосредственно соседствует в последовательностях с названным ключевым Объектом или разделен с названным ключевым Объектом числом частотных объектов, соответствующим базовому рангу.

В памяти сохраняют некоторое множество всех ранговых множеств базового ранга в качестве эталонного (далее "Эталонное Состояние Памяти" или "ЭСП"), а любое "Мгновенное состояние памяти" (далее "МСП") или его часть сравнивают с ЭСП или его частью для выявления отклонений МСП от ЭСП.

Массив "будущего" или массив "прошлого", или ранговое множество ранга, отличного от множества базового ранга, представляют множеством, производным от множества МСП.

Массив данных "Будущее" и массив данных "Прошлое" формируют как линейную композицию весовых коэффициентов или ранговых множеств массива данных МСП.

7.1.2. Односекционная матрица.

Искусственные Нейроны Встречаемости односекционной матрицы кодируют ранговое множество весов одного и того же ранга, предпочтительно первого ранга.

Для каждого уникального ключевого Объекта в памяти сохраняют по меньшей мере одно ранговое множество или "будущего", или "прошлого" ранг которого одинаков для всех уникальных ключевых Объектов (далее "базовый ранг" множества), причем каждый весовой коэффициент взаимной встречаемости ключевого и частотного Объектов названного рангового множества относится к частотному Объекту, который непосредственно соседствует в последовательностях с названным ключевым Объектом или разделен с названным ключевым Объектом числом частотных объектов, соответствующим базовому рангу.

Названное ранговое множество, являясь множеством первого ранга, содержит весовые коэффициенты частотных Объектов, непосредственно соседствующих с названным ключевым Объектом в названных последовательностях.

В памяти сохраняют ограниченное число ранговых множеств.

Косынку (фиг. 38) можно использовать для получения как полного, так и рангового Кластера сферы будущего или прошлого с некоторым радиусом R .

Для получения Полного Кластера сферы радиуса R необходимо организовать цикл, при котором сигналы объектов на выходе передаются на вход косынки, а веса каждого из уникальных объектов Полного Кластера суммируются с весами этого объекта, полученными на предыдущих циклах. Полное количество циклов должно соответствовать радиусу сферы R .

Для получения Рангового Кластера необходимо организовать цикл, при котором сигналы объектов на выходе передаются на вход косынки, а веса каждого из уникальных объектов Рангового Кластера фиксируются на выходе из матрицы лишь в конце последнего цикла. Полное количество циклов должно соответствовать рангу Кластера R .

Недостатком использования матрицы с одной косынкой является ее медленная работа, а именно то, что для получения как Полного, так и Рангового Кластеров, необходимо число циклов работы косынки, равное радиусу сферы (он же и ранг) для которой рассчитывается Кластер. Еще более длительным процессом окажется извлечение Ранговых Кластеров для построения гипотез продолжения последовательности и для построения Обратной Проекции Когерентных Кластеров.

Память Последовательностей, представленная Односекционной матрицей, является матрицей массива "Состояния памяти" (формула 16) и [2.3.11]. Если подать сигналы на входы шин всех объектов Односекционной матрицы, то на выходе мы получим множество весов, представляющих проекцию вектора "Состояния сознания" на координатные оси, объектов Памяти Последовательностей. Любой Кластер Памяти Последовательностей может быть воспроизведен с помощью Односекционной матрицы как проекция вектора \vec{K}_{state} на названные оси. Это позволяет реализовать Память Последовательностей только с использованием Односекционной матрицы, например односекционной матрицы первого ранга. Тем не менее мы рассмотрим другие архитектуры, обладающие более высокой производительностью.

Нетрудно заметить, что связи косынки "Состояния сознания" графически (фиг. 38) можно представить точками, размещенными в пересечениях косынки, а вес встречаемости передавать или разным размером точек или цветом или другим образом. Это позволяет представить косынку в виде картинку и использовать картинку в качестве исходных данных для перцептронных или сверточных или других нейросетей с известной архитектурой. Целью такого обучения нейросети может быть установление соответствия между графическим представлением весов "Состояния сознания" косынки и множеством объектов, которые сформировали такое графическое представление. В робототехнике обученная таким образом нейросеть известной архитектуры может контролировать отклонения "Состояния сознания" Памяти Последовательностей робота (Мгновенное Состояние Памяти - МСП) от заданного эталонного состояния (Эталонное Состояние Памяти - ЭСП) с целью коррекции состояния памяти и реагировать на другие определенные человеком состояния ПП. Например, применением нейросетей для работы с Памятью Последовательностей может быть обучение нейросети по распознаванию Кластеров Труб или Кластеров Калибров Труб, что может позволить активировать шины Труб и Генераторы Трубы в Памяти Последовательностей, используя механизмы распознавания нейросетей.

Различия между ЭСП и МСП могут также использоваться для прогнозирования появления новых объектов, а значит для поиска (обнаружения) и коррекции ошибок ввода последовательностей.

При отклонении МСП от ЭСП выполняют алгоритм поиска или алгоритм прогнозирования, или алгоритм коррекции ошибки, или их комбинацию.

При вводе Объекта, уникальный цифровой код которого мог быть введен с ошибкой, сравнение ранговых множеств осуществляют с целью выявления возможной ошибки.

7.1.3. Одноранговая матрица.

В ПП одна или более Косынок (далее "Секции") соединены последовательно, причем в названных ИНВ каждой из Секций накапливаются, хранятся и предоставляются для считывания только веса связей Первого Ранга, а шины с одинаковыми номерами каждых двух последовательных Секций N и $(N+1)$ соединены так, что выходы шин Секции N служат входами шин смежной Секции $(N+1)$, причем в режиме обучения или все Секции обучаются одновременно или обучается только одна Секция X , а последнее значение встречаемости в памяти Счетчика, расположенного в пересечении двух конкретных шин любой из секций, приравнивается последнему значению Счетчика, расположенного в пересечении тех же двух конкретных шин Секции X , а в режиме воспроизведения Секция $(N+1)$ используются для повторного изменения сигналов, поступивших из Секции N .

Производительность "Односекционной матрицы" [7.1.2] со связями первого ранга можно улучшить, увеличив число косынок в матрице. Как отмечалось ранее [2.3.9], связь между любыми двумя соседними объектами последовательности является связью первого ранга и последовательно соединив косынку содержащую связи первого ранга с другой косынкой, содержащей те же самые связи первого ранга, затем

соединив две первые с третьей косынкой и так далее, можно создать матрицу, содержащую только косынки со связями первого ранга, последовательно соединенные друг с другом. Ниже показана матрица, имеющая две косынки со связями первого ранга (фиг. 39).

Матрицу, содержащую R-секций тождественных косынок со связями первого ранга, далее станем называть "Одноранговая матрица". Последовательное соединение косынок можно условно изобразить как показано ниже (фиг. 40).

Под "последовательностью" соединения косынок имеется в виду, что выходы шин первой секции являются входами второй секции и так далее. В случае с матрицей из двух секций за один цикл можно извлечь Кластер для сферы с радиусом $R=2$. При этом веса объектов после считывания передаются на выход из матрицы и там складываются с весами этого объекта, полученными из других секций матрицы.

Последовательное соединение односекционных матриц позволяет за один цикл работы Одноранговой матрицы получить Кластеры для сфер разного радиуса на выходе каждой односекционной матрицы.

7.1.4. Многограновая матрица.

В ПП одна или более Косынок (далее "Ранговые Секции") соединены последовательно, причем в названных ИНВ каждой из Секций накапливаются, хранятся и предоставляются для считывания веса связей одинакового Ранга (далее "Ранг Секции") и Ранги смежных Секций отличаются на единицу или другую величину, а шины с одинаковыми номерами каждых двух смежных Секций Ранга N и Ранга $(N+1)$ соединены так, что выходы шин Секции Ранга N служат входами шин смежной Секции Ранга $(N+1)$, причем в режиме обучения Нейроны Встречаемости каждой Секции обучаются на связях Ранга соответствующего Рангу Секции, а в режиме воспроизведения каждая Секция определенного Ранга используется для считывания сигналов измененных названными ИНВ Секции соответствующего Ранга.

Недостатком архитектуры "косынки" (фиг. 35) является то, что она имеет высокую трудоемкость исследования связей высокого ранга - взаимной встречаемости объектов, разделенных в последовательности множеством других объектов.

Тем не менее архитектура косынки [7.1.1] может быть использована в качестве "Ранговой косынки" для прошивки связей более высокого ранга. Для связей соответствующего ранга создается своя косынка и в ней прошиваются связи объекты C_n с объектами C_k соответствующего ранга, а считывание связей объекта C_n с объектами C_k соответствующего ранга производится или однократно, или в соответствии с пошаговым алгоритмом, изложенным в разделе [7.1.1]. Используя отдельную Ранговую косынку для прошивки связей соответствующего ранга, и соединив Ранговые косынки разного ранга в единую архитектуру, можно одновременно работать со связями разного ранга - записывать их и считывать как описано в разделе [7.1.1]. Разные секции такой матрицы могут независимо друг от друга воспроизводить Кластер соответствующего ранга, поэтому связи между секциями являются параллельно-последовательными (см. фиг. 41).

Следует заметить, что, как и Односекционная матрица [7.1.2], каждая из ранговых косынок характеризуют устойчивое статистическое "состояние памяти" последовательностей или "состояние сознания" [2.3.11] с различной глубиной связей (ранги) и именно это позволяет увеличить глубокого прогнозирования с использованием архитектуры Многограновой матрицы и как следствие увеличить производительность архитектуры.

7.1.5. Генератор геометрии матрицы.

Архитектуру матрицы (фиг. 35) можно для этого улучшить, используя "косынку" в качестве генератора более сложной геометрии со связями "каждого с каждым". Фиг. 42 демонстрирует генератор, состоящий из двух косынок и имеющий соответственно связи первого и второго рангов "каждый с каждым". Далее любую архитектуру с генератором "косынка" или более сложным генератором из нескольких косынок будем называть просто "матрица".

Увеличивая число косынок или генераторов (фиг. 42) матрицы мы сможем линейно увеличивать и число пересечений "каждый с каждым" (см. фиг. 43).

Предложенная матрица связей (фиг. 43) позволяет реализовать связи 1-го ранга в первой "косынке" матрицы, связи 2-го ранга во второй "косынке" и так далее до N-ной "косынки", реализующей связи N-го ранга.

Замечание 9.

В процессе обучения матрицы создаются обратные связи, которые могут считываться как упреждающие при смене направления считывания. Поэтому несмотря на то, что на матрице (фиг. 43) входы A показаны справа, а выходы B слева, ранговые блоки могут следовать слева направо или справа налево в зависимости от того, какой режим матрицы включен. Для записи последовательностей в матрицу и для считывания связей "прошлого" нумерация ранговых блоков будет прямая 1, 2, 3, ..., N. Для считывания связей "будущего" нумерация ранговых блоков была бы справа налево. Поэтому матрицы удобно рисовать, нумеруя ранговые блоки, не указывая направления вход \rightarrow выход (фиг. 44), так как при неизменности нумерации входы и выходы могут меняться местами.

Таким образом, входы Косынки могут использоваться в качестве выходов, а выходы в качестве входов.

Понятно, что форма матрицы (фиг. 43) зависит от использованного генератора матрицы (показан

светлыми линиями шин) и в зависимости от формы генератора, числа косынок входящих в генератор и их соединения внутри генератора, а также соединения генераторов между собой, матрица может принимать разные формы. Матрица может по форме не только плоской, но и объемной, поднимаясь по спирали, как молекула ДНК, создавая необходимое количество пересечений "каждый с каждым" очередного ранга с добавлением к матрице каждого нового генератора на новом витке/слое. Изменяя угол между входами А и выходами В (см. фиг. 19) можно физически менять и геометрию генератора, а значит и топологию архитектуры в зависимости от условий производства или требований дизайна. Соединяя друг с другом генераторы расположенные один над другим, можно получать слоистую (вафельную) архитектуру чипа, каждый слой которого представлен отдельным генератором или архитектурой, порожденной генератором.

7.1.6. Слои матрицы.

В матрице размещаются два слоя: слой объектов - слой М1, и слой Труб - слой М2 (фиг. 45).

7.1.7. Некоторые замечания.

Для разделения режимов обучения, вспоминания и возможно других, матрица должна иметь канал управления режимами и переключаться в соответствующий режим после получения управляющего сигнала.

Поскольку из-за большого числа уникальных объектов все шины матрицы не могут быть выведены на отдельные ножки разъема микросхемы, то необходим блок коммутации, обеспечивающий коммутацию шин матрицы с ограниченным числом ножек микросхемы, позволяющий передавать полученные на ножки входящие сигналы объектов на соответствующую сигналу шину матрицы, а также позволяющий считывать с шин матрицы выходящие сигналы объектов и передавать их на ножки разъема микросхемы.

Другим решением для коммутации матрицы с внешними устройствами может служить беспроводный ввод и вывод сигналов объектов. Для этого матрицу снабжают радио приемо-передающим радиокоммутатором (РК). Входящий сигнал для каждой из шин матрицы может быть получен РК и передан на соответствующую шину. Сигналы, считанные на выходе с шин, поступают в РК, который передает сигналы по радиоканалу внешним потребителям.

В процессе работы ПП появляются синтетические объекты. Поэтому матрица должна обладать дополнительными шинами, которые не принадлежат ни одному известному объекту последовательностей ПП.

7.2. Слой объектов.

7.2.1. Искусственный нейрон взаимной встречаемости объектов.

Названный ИНВ накапливает, хранит и предоставляет для считывания значение веса совместной встречаемости для двух объектов последовательности, которые или не разделены другими объектами, а непосредственно следуют друг за другом в последовательности, образуя связь Первого Ранга, или разделены одним или более объектами в последовательности, образуя соответственно связь Второго или более высокого Ранга.

7.2.1.1. Устройство Искусственного Нейрона Встречаемости (ИНВ).

Искусственным Нейроном Совместной Встречаемости (ИНВ) или "Счетчиком" назовем элемент, расположенный в месте пересечения шин объектов C_n и C_k и предназначенный для накапливания значения случаев взаимной встречаемости объектов $C_n \rightarrow C_k$ в процессе обучения матрицы, а также воспроизведения накопленного значения в процессе считывания. Как отмечалось выше [7.1.1], в пересечении шин объектов C_n и C_k должны быть расположены не один, а два Счетчика (А и В) из которых один запоминает прямую $C_n \rightarrow C_k$, а второй обратную $C_k \rightarrow C_n$ встречаемость объектов C_n и C_k , а также сенсор С который измеряет соотношение весов прямой и обратной встречаемости $i = a/b$ (далее i "показатель инверсии"), а также направление инверсии \vec{I} , которым мы станем называть направление от большего веса к меньшему, значение инверсии для которого должно быть больше единицы $i > 1$. (фиг. 46).

Сенсор инверсии может быть выполнен, например, в виде конденсатора С (фиг. 47). А именно в режиме обучения на шину двух объектов Окна Внимания подаются сигналы разной силы, что позволяет запитать оба Счетчика каждый из которых соединен с одной из обкладок конденсатора С, а заряд конденсатора и его полярность зависят от значения показателя инверсии i и от направления инверсии \vec{I} .

Таким образом, в режиме обучения конденсатор заряжается тем сильнее, чем выше показатель инверсии, а полярность заряда совпадает с направлением инверсии. В режиме "воспроизведения" который следует за режимом обучения конденсатор разряжается и на входах/выходах шин А и В появляется разность потенциалов, соответствующая значению инверсии, а полярность указывает направление инверсии. Это позволяет "предсказывать" появление следующего объекта последовательности. Несмотря на то что здесь приведен пример с использованием конденсатора, специалист знакомый с уровнем техники может предложить и другие способы реализации сенсора инверсии, не выходя за рамки уровня, раскрытого в настоящей работе.

Однако далее процесс обучения и считывания связей мы будем иллюстрировать только одним из направлений (фиг. 48), например на направлении $C_n \rightarrow C_k$. Напомним также, что смена полярности шин объектов соответствует смене направления с "будущего" на "прошлое" или наоборот.

Поскольку Нейроны Встречаемости размещаются в "косынках" матрицы, то Счетчики имеют Ранг соответствующей "косынке". Таким образом, каждый Счетчик может быть уникально идентифицирован идентификаторами двух объектов пик, а также рангом g . Поэтому можно использовать следующую идентификацию счетчиков: $\Sigma_{n,k,g}$.

Счетчик - название условное, он может быть, например, мемристором или элементом, использующим другие физические принципы, однако в любом случае задачей Счетчика является накопление значения совместной встречаемости объектов на пересечении шин которых находится такой Счетчик.

Счетчик также может обладать возможностью частично или полностью "забывать" число появлений объекта в последовательностях, в зависимости от частоты появления объекта в последовательностях. Последнее свойство позволяет "забывать" о встречаемости объектов, которые редко встречаются в последовательностях.

Счетчик должен обрабатывать входящий сигнал противоположных направлений и генерировать выходной сигнал, соответствующий по направлению входящему.

Каждое из сочетаний объектов $C_k \rightarrow C_n$ и $C_n \rightarrow C_k$ должно быть представлено отдельным счетчиком (фиг. 51).

Поскольку в режиме обучения матрица записывает обратные связи, то если сигнал считывания памяти совпадает с направлением сигналов обучения, то матрица будет воспроизводить Кластер прошлого (прямое воспроизведение), а если входящий сигнал противоположен сигналам обучения, то матрица будет воспроизводить Кластер будущего (обратное воспроизведение).

7.2.1.2. Состояние нейрона до обучения.

До обучения нейрон можно условно представить, как показано ниже (фиг. 49).

До обучения память нейрона 1 пуста, поэтому нейрон 1 показан штриховой линией. Пока вес связи $\Sigma_{n,k,k}=0$, связь 6 в пересечении шин объектов C_n и C_k не существует, чему соответствует положение "закрыто" условного вентиля 7. В состоянии с закрытым вентилям 7, связь 6 не может пропустить сигнал между шинами нейрона 1 и потому в режиме считывания условные вентили 4 и 5 шин объектов находятся в положении "открыто" проводя сигналы далее по шинам объектов.

7.2.1.3. Состояние нейрона после обучения.

Память обученного нейрона не пуста и поэтому нейрон 1 изображен сплошной линией. Если в нейрон записано любое значение взаимной встречаемости $\Sigma_{n,k,k}>0$, связь в пересечении шин объектов C_n и C_k открывается, чему соответствует положение "открыто" условного вентиля 7. В состоянии с открытым вентилям 7, связь 6 может пропустить сигнал между шинами нейрона 1 и потому в режиме считывания условные вентили 4 и 5 шин объектов находятся в положении "закрыто" не проводя сигналы далее по шинам объектов, а вместо этого вентили 4 и 5 коммутируют шины объектов со связью нейрона, что позволяет пропустить сигнал из одной шины объекта C_n через связь 6 в другую шину C_k , считывая значение веса $\Sigma_{n,k,k}$ и передавая его во вторую шину C_k .

Замечание 10.

Обучение нейронов возможно только в направлении обратной связи, однако считывать значение взаимной встречаемости нейрона можно как в направлении обратной, так и в направлении упреждающей связи, причем считывание упреждающей и обратной связей должно быть возможно как в направлении $C_n \rightarrow C_k$ так и в обратном направлении $C_k \rightarrow C_n$.

7.2.1.4. Запись в память нейронов многогранговой матрицы.

Запись происходит в режиме обучения нейронов (запоминание). Чтобы обучить нейроны на пересечении шин вводимых объектов, в матрицу одновременно вводят не менее двух объектов последовательности. Для ввода объектов в матрицу их кодируют соответственно сигналами S_1, \dots, S_n , которые подаются на шины объектов. Сигналы объектов должны кодировать идентификатор объекта, а место объекта в Окне Внимания должно кодироваться другой измеримой характеристикой сигнала объекта, которую условно назовем "силой" сигнала. Самым сильным является сигнал S_1 самого позднего объекта Окна Внимания, а сила сигнала любого объекта C_k Окна Внимания вычисляется как функция силы сигнала S_1 и ранга объекта k в Окне Внимания (формула 54):

$$S_k = f(k, S_1)$$

Формула 54 позволяет вычислить разницу $\Delta S_{1,k}=(S_1-S_k)$ между силой сигнала S_1 и S_k объектов C_1 и C_k . Если разница между силами сигналов шин объектов C_1 и C_k равна $\Delta S_{1,k}$, то это позволяет обучать нейроны "косынки" ранга k только на разнице $\Delta S_{1,k}$. Поэтому предпочтительным является решение, при котором обучаются только нейроны на пересечениях шины самого позднего объекта C_1 Окна Внимания с шинами других объектов C_k Окна Внимания. Запоминать значения взаимной встречаемости будут только Счетчики с идентификаторами $\Sigma_{1,k,k}$, расположенные на шине самого позднего объекта C_1 Окна Внимания в пересечениях с другими объектами C_k Окна Внимания, а ранг k Счетчика будет соответствовать рангу объекта C_k в Окне Внимания. То есть значение взаимной встречаемости объектов C_1 и C_2 будет записано в Счетчике косынки 2-го ранга. По сути это косынка 1-го ранга, так как объекты C_1 и C_2 не разделены другими объектами, но тогда сумму пришлось бы обозначать не $\Sigma_{1,k,k}$, а $\Sigma_{1,k,k+1}$. Значение C_1 и C_3

будет записано в Счетчике косынки 3-го ранга и так далее значение C_1 и C_n будет записано в Счетчике косынки n -го ранга.

Таким образом, процесс обучения нейрона можно описать так:

1. Перед началом обучения нейроны содержат нулевое значение $\Sigma_{n,k}=0$. Значение веса взаимной встречаемости объектов C_n и C_k разделенных $(r-1)$ объектами последовательности перед обучением равно нулю $\Sigma_{n,k}=0$, а после начала обучения нейроны накапливают значение веса встречаемости $\Sigma_{n,k}=\Sigma_{n,k}+f(k)$.

2. Во время обучения Счетчик одновременно получает два входящих сигнала S_1 и S_k соответственно на шинах C_1 и C_k косынки ранга k .

3. Вес связи взаимной встречаемости объектов C_1 и C_k увеличивается только в нейроне косынке ранга k , где разница силы сигналов соответствует величине $(S_1-S_k)=\Delta S_{1,k}$, то Счетчик запоминает новое значение $\Sigma_{1,k,k}=\Sigma_{1,k}+f(k)$, совместной встречаемости объектов C_1 и C_k , где k -ранг объекта C_k в Окне Внимания и ранг косынки матрицы, причем значение $f(k)$ может изменяться или быть постоянной величиной, в частности может быть $f(k)=1$.

Пример.

Рассмотрим пример линейной функции ослабления сигналов $f(n)=(1-(n-1)/N)$ и тогда сила сигналов на шинах объектов Окна Внимания в режиме запоминания матрицы будет (формула 55):

$$S_n = S_1 * (1 - (n - 1)/N)$$

где N - полное число ранговых блоков в ПП, а номер конкретного рангового блока n - целочисленная величина, удовлетворяющая условию $0 < n \leq N$.

Поскольку функция ослабления линейная, то разница $\Delta S=(S_n-S_{n+1})$ в измеряемой характеристике сигнала между каждыми двумя последовательными входами C_n и C_{n+1} будет постоянной величиной. Тем не менее функция ослабления может быть выбрана и нелинейной.

Сигнал силой S_1 всегда подается на шину самого позднего объекта Окна Внимания; сигнал силой S_2 - на шину предпоследнего объекта и так далее до S_n , который подается на шину n -го объекта Окна Внимания введенного на $(n-1)$ объект ранее самого позднего. Тогда в каждый момент времени распределение силы сигналов объектов Окна Внимания будет таким как показано на фиг. 52.

Под "силой" сигнала понимается любая измеряемая характеристика - сопротивление, емкость, напряжение, сила тока, частот и так далее, или другая характеристика, зависящая от перечисленных.

В блоке матрицы 1- ранга значение Счетчика изменяется между шиной с силой сигнала S_1 и шиной с силой сигнала S_2 , разница между которыми равна $\Delta S_1=S_1/N$.

В блоке матрицы 2-го ранга значение Счетчика изменяется между шиной с силой сигнала S_1 и шиной с силой сигнала S_3 , разница между которыми равна $\Delta S_2=2 \times \Delta S_1$.

В блоке матрицы 3-го ранга значение Счетчика изменяется между шиной с силой сигнала S_1 и шиной с силой сигнала S_4 , то есть между шиной с силой сигнала S_1 и шиной, чей сигнал слабее S_1 на величину $\Delta S_3=2 \times S_1$.

И так далее в блоке матрицы $(k-1)$ -го ранга значение Счетчика изменяется между шиной с силой сигнала S_1 и шиной с силой сигнала S_n , разница между которыми равна $\Delta S_{(n-1)}=(n-1) \times \Delta S_1$.

7.2.1.5. Считывание памяти нейрона матрицы.

Несмотря на то что для нумерации объектов Окна Внимания при записи в ПП мы использовали функцию ослабления, при считывании ПП такой необходимости нет сигналы всех шин на входе в матрицу могут быть одинаковой силы. Это, в частности, дает преимущество, так как позволяет избежать необходимости нормализации профиля Кластера Трубы на выходе из матрицы и позволяет сравнивать не нормализованные, а истинные Кластеры Труб и Субтруб.

В то время как запись в память нейрона происходит только в одном направлении - в направлении обратных связей, считывать память нейрона можно в обоих направлениях, как в направлении обратных, так и в направлении упреждающих связей, как в направлении $C_k \rightarrow C_n$, так и в обратном направлении $C_n \rightarrow C_k$ (фиг. 53).

Для считывания значений взаимной встречаемости объекта C_n с другими объектами C_k в режиме считывания памяти нейронов на шину объекта C_n подают сигнал названного объекта S_n . Считывание значения памяти нейрона (считывание значения Счетчика) на пересечении шин объектов C_n и C_k в косынке ранга r возможно только в случае, если значение взаимной встречаемости названных объектов больше нуля $\Sigma_{n,k,r} > 0$.

Если на пересечении шин объектов C_n и C_k значение счетчика $\Sigma_{n,k,r} > 0$, то сигнал с шины объекта C_n поступает через нейрон на шину объекта C_k , при этом считывая значение $\Sigma_{n,k,r}$ и вычисляя значение веса совместной встречаемости $w_{k(n,r)}=f(S_n, \Sigma_{n,k,r})$, сигнал C_k и значение веса $w_{k(n,r)}$ выводятся на шину объекта C_k (фиг. 54). Таким образом на входе шины объекта C_n имеем исходный сигнал S_n , а на выходе нейрона имеем сигнал веса совместной встречаемости $w_{k(n,r)}=f(S_n, \Sigma_{n,k,r})$ который выводится в шину объекта C_k .

7.2.1.6. Считывание рангового Кластера объекта.

Многоранговая матрица.

Многоранговая матрица показана на фиг. 55.

При считывании рангового Кластера K_n объекта C_n (фиг. 55) на шину объекта C_n подается сигнал считывания, а веса совместной встречаемости объекта C_n с другими объектами C_k считываются с нейронов косынки ранга r , расположенных в пересечении шин объектов C_n и C_k . На выходе шины объекта C_k получим вес совместной встречаемости $w_{k(n,r)}$.

Таким образом, процесс чтения памяти нейрона можно описать так:

1. Для считывания весов совместной встречаемости объекта C_n с другими объектами Памяти Последовательностей, на шину объекта C_n подается сигнал S_n
2. Если значение Счетчика расположенного на пересечении шин C_n и C_k не равно нулю, то используя значение Счетчика $\Sigma_{n,k,r}$ Счетчик преобразует входящий сигнал S_n в сигнал веса совместной встречаемости $w_{k(n,r)}$: $w_{k(n,r)} = f(S_n, \Sigma_{n,k,r})$
3. Сигнал совместной встречаемости $w_{k(n,r)}$ передается на шину объекта C_k , где считывают вес совместной встречаемости w_k
4. Ранговый Кластер K_n^i объекта C_n формируют как вес w_k , полученный с шины объекта C_k , считанный с косынки ранга i матрицы связей.

Продемонстрируем это на примере. Предположим, что нам надо считать все связи третьего ранга для объекта C_n , причем объект C_n в косынке связей ранга 1 имеет связь только с объектом C_k , а в косынке связей ранга 2 имеет связь только с объектом C_l , и в косынке связей ранга 3 имеет связь только с объектом C_m (фиг. 56).

С матрицы будут считаны веса только с ранговой косынки ранга три, поэтому результатом считывания весов третьего ранга будет единственный вес w_m объекта C_m , а Кластер будет таким:

$$K_n^3 = \{w_{l(m,3)} * C_m\}$$

Одноранговая матрица.

При считывании Кластера объекта C_n на его шину подается сигнал S_n . В матрице при этом считываются только веса 1-го ранга между косынками матрицы последовательных секций [1 и 2], ..., [г и (г+1)], [(г+1) и (г+2)], ... и так далее. Это проиллюстрировано ниже (фиг. 57).

Рассмотрим матрицу, которая имеет три косынки (секции) со связями первого ранга (фиг. 57). Перед нами стоит задача считать ранговый Кластер третьего ранга K_n^3 объекта C_n , поэтому сумматоры на выходе шин матрицы настраиваются на запоминание только весов, полученных с третьей косынки. Предположим, что объект C_n в косынке 1 имеет связь только с объектом C_k , а объект C_k в косынке 2 имеет связь только с объектом C_l , и объект C_l в косынке 3 имеет связь только с объектом C_m . Для считывания рангового Кластера третьего ранга K_n^3 объекта C_n сигнал S_n подается на шину объекта C_n и после считывания веса совместной встречаемости $w_{n(k,1)} = f(S_n, \Sigma_{n,k,1})$ между объектами C_n и C_k вес взаимной встречаемости выводится в сумматор весов частотного объекта C_k , а сигнал S_k , с шины объекта C_k поступает в сумматор между шинами объектов C_k и C_l , где считывает значение веса взаимной встречаемости $w_{k(l,2)} = f(S_k, \Sigma_{k,l,2})$ объектов C_k и C_l и передает значение веса в сумматор весов шины объекта C_l , а сигнал S_l с шины объекта C_l поступает в сумматор между шинами объектов C_l и C_m , где считывает значение веса взаимной встречаемости $w_{l(m,3)} = f(S_l, \Sigma_{l,m,3})$ объектов C_l и C_m и передает значение веса в сумматор весов шины объекта C_m .

Сумматоры шин выбирают только веса, полученные со счетчиков третьей секции, и ранговый Кластер третьего ранга для объекта C_n будет содержать только вес объекта C_m :

$$K_n^3 = \{w_{l(m,3)} * C_m\}$$

7.2.1.7. Считывание полного Кластера объекта.

Многоранговая матрица.

Для считывания полного Кластера объекта C_n считывают сумму всех ранговых Кластеров объекта C_n . Для этого на шину объекта C_n подается сигнал считывания, а значения Счетчиков считывают на пересечении шины объекта C_n с каждым объектом C_k каждой из ранговых косынок. Полный вес w_k совместной встречаемости объекта C_n с объектом C_k рассчитывают, как сумму всех весов встречаемости названных объектов в косынках разных рангов $w_{k(n,r)}$ (формула 56):

$$w_k = \sum_{i=1}^N \sum_{j=1}^R w_{k(i,j)}$$

где j - ранг косынки, а R - полное число ранговых косынок, i - номер/идентификатор объекта, а N - полное число уникальных объектов C_i .

Таким образом, процесс чтения памяти нейрона можно описать так:

1. Для считывания весов совместной встречаемости объекта C_n с другими объектами Памяти Последовательностей, на шину объекта C_n подается сигнал S_n .
2. Если значение Счетчика расположенного на пересечении шин C_n и C_k не равно нулю, то используя значение Счетчика $\Sigma_{n,k,r}$ Счетчик преобразует входящий сигнал S_n в сигнал веса совместной встречаемости $w_{k(n,r)}$: $w_{k(n,r)} = f(S_n, \Sigma_{n,k,r})$.
3. Сигнал совместной встречаемости $w_{k(n,r)}$ передается на шину объекта C_k где складывается с веса-

ми совместной встречаемости объекта C_k полученными со Счетчиков в пересечениях шины C_k с объектом C_n в косынках других рангов Памяти Последовательностей:

$$w_k = \sum_{i=1}^N \sum_{j=1}^R w_{k(i,j)}$$

где R - ранг косынки, а N - полное число уникальных объектов C_i .

4. На выходе матрицы с шины каждого объекта C_k Памяти Последовательностей снимают сигнал суммарного веса совместной встречаемости w_k объекта C_k с объектом C_n формируя полный Кластер объекта C_n .

Продемонстрируем это на примере. Предположим, что нам надо считать все связи всех рангов для объекта C_n , причем объект C_n в косынке связей ранга 1 имеет связь только с объектом C_k , а в косынке связей ранга 2 имеет связь только с объектом C_l , и в косынке связей ранга 3 имеет связь только с объектом C_m (фиг. 58).

Для считывания полного Кластера объекта C_n сигнал S_n подается на шину объекта C_n и параллельно поступает в нейроны, расположенные на пересечении с шинами объектов C_k , C_l и C_m . Значение веса совместной встречаемости $w_{n(k,1)}=f(S_n, \Sigma_{n,k,1})$ между объектами C_n и C_k выводится в сумматор весов частотного объекта C_k , значение веса взаимной встречаемости $w_{k(l,2)}=f(S_k, \Sigma_{k,l,2})$ объектов C_k и C_l выводится в сумматор весов шины объекта C_l , а значение веса взаимной встречаемости $w_{l(m,3)}=f(S_l, \Sigma_{l,m,3})$ объектов C_l и C_m выводится в сумматор весов шины объекта C_m .

Таким образом, получим полный Кластер объекта C_n :

$$K_n = \{w_{n(k,1)} * C_k; w_{k(l,2)} * C_l; w_{l(m,3)} * C_m\}$$

Одноранговая матрица.

При считывании Кластера объекта C_n на его шину подается сигнал S_n . В матрице при этом считываются только веса 1-го ранга между косынками матрицы последовательных секций, 1 и 2, ..., r , и $r+1$, ..., $r+1$, и $r+2$, и так далее. Это проиллюстрировано ниже (фиг. 59).

Рассмотрим матрицу, которая имеет три косынки со связями первого ранга (фиг. 59). Перед нами стоит задача считать полный Кластер объекта C_n . Предположим, что объект C_n в косынке 1 имеет связь только с объектом C_k , а объект C_k в косынке 2 имеет связь только с объектом C_l , и объект C_l в косынке 3 имеет связь только с объектом C_m . Для считывания полного Кластера объекта C_n сигнал S_n подается на шину объекта C_n и после считывания веса совместной встречаемости $w_{n(k,1)}=f(S_n, \Sigma_{n,k,1})$ между объектами C_n и C_k вес взаимной встречаемости выводится в сумматор весов частотного объекта C_k , а сигнал S_k , с шины объекта C_k поступает в сумматор между шинами объектов C_k и C_l , где считывает значение веса взаимной встречаемости $w_{k(l,2)}=f(S_k, \Sigma_{k,l,2})$ объектов C_k и C_l и передает значение веса в сумматор весов шины объекта C_l , а сигнал S_l , с шины объекта C_l поступает в сумматор между шинами объектов C_l и C_m , где считывает значение веса взаимной встречаемости $w_{l(m,3)}=f(S_l, \Sigma_{l,m,3})$ объектов C_l и C_m и передает значение веса в сумматор весов шины объекта C_m .

Таким образом, получим полный Кластер объекта C_n :

$$K_n = \{w_{n(k,1)} * C_k; w_{k(l,2)} * C_l; w_{l(m,3)} * C_m\}$$

7.2.1.8. Считывание Когерентных Ранговых Кластеров.

Когерентные Ранговые Кластеры были подробно рассмотрены в разделе [3.2.6], а этом разделе мы лишь рассмотрим способы считывания Когерентных Ранговых Кластеров с матрицы с учетом особенностей работы с матрицей [замечание 9].

Названные Кластеры $K_i^{(k-i)}$ можно считывать, подавая сигналы на шины объектов Окна Внимания на входе матрицы и считывая их ранговые Кластеры в соответствующем ранговом блоке матрицы:

Каждый объект C_i Окна Внимания, содержащего всего $(k-1)$ объект, подают на вход соответствующей шины матрицы и считывают с матрицы ранговый Кластер $K_i^{(k-i)}$, где верхний индекс означает ранг Кластера, а нижний индекс означает номер объекта в Окне Внимания. Для самого позднего объекта ОВ будет считан Кластер ранга 1, для предыдущего объекта ОВ - Кластер ранга 2 и так далее для самого раннего объекта ОВ - Кластер ранга $(k-1)$.

После считывания ранговых Кластеров $K_i^{(k-i)}$ для всех объектов Окна Внимания над полученными Кластерами проводят операции как с когерентными ранговыми Кластерами Окна Внимания.

7.3. Интеграция ПП с Нейронной Сетью известной архитектуры.

Поскольку в процессе работы ПП на выходах шин матрицы ПП можно считать множество весов совместной встречаемости объектов последовательностей, то такое множество можно использовать в качестве исходных данных (например, представить картой признаков) для подачи на вход Нейронной Сети с известной архитектурой, содержащей или только полносвязный слой перцептронов, или множество слоев, включая сверточные Convolution, ReLU, Pooling, (Subsampling) и полносвязные слои перцептронов, соединенные с использованием любой известной архитектуры, включая архитектуру GoogleNet Inception, ResNet, Inception-ResNet и любые другие известные архитектуры.

Другим способом интеграции с традиционной нейросетью может быть соединение выходов матри-

цы ПП с входами традиционной нейросети для передачи весов Кластера Трубы или Калибра Трубы на вход традиционной нейросети в качестве исходных данных.

По меньшей мере один из названных массивов будущего или прошлого, или названных множеств Трубы или Калибра Трубы, или Эталонное Состояние Памяти (ЭСП), или Мгновенное Состояние Памяти (МСП), или совокупность названных массивов и множеств, или любое множество, которое является производным от названных массивов и множеств, вводят в качестве исходных данных в искусственную нейронную сеть перцептронов или свёрточную нейронную сеть или другую искусственную нейронную сеть с известной архитектурой.

Выходы шин ПП уровня иерархии М1 используют в качестве входов для искусственной нейронной сети перцептронов или свёрточной нейронной сети или другой искусственной нейронной сети с известной архитектурой, которую используют в качестве названного множества ИНИ.

7.4. Недостатки традиционных нейросетей.

Известно, что нейросеть с известной архитектурой (традиционная нейросеть) способна создавать или создает синтетические объекты. Например, сверточные слои сверточных сетей создают новые объекты более высокого уровня иерархии путем свертки исходных изображений и создания карт признаков, причем процесс свертки изображений хорошо описан и понятен. Каждый из перцептронов полносвязного слоя традиционных нейросетей по сути является объектом более высокого уровня иерархии чем объекты на который был обучен перцептрон и с которыми связаны множество входов перцептрона. Однако известные способы обучения слоев перцептронов и устройство перцептрона не позволяют конкретным перцептронам образовать рекуррентные связи с конкретными последовательностями или фрагментами последовательностей, на которых был обучен полносвязный слой перцептронов. Причем, все перцептроны полносвязного слоя обучаются одновременно, что не позволяет определить порядок обучения конкретных перцептронов слоя и построить из обученных перцептронов последовательности объектов следующего уровня иерархии. Как следствие, традиционные нейросети не позволяют создать Память Последовательностей разных уровней иерархии.

Названные ограничения традиционных нейросетей, не позволяют анализировать смысловую встречаемость на разных уровнях смысловой иерархии, то есть не позволяют исследовать причинно-следственные связи разного уровня иерархии.

Перечисленные недостатки ограничивают применение традиционных нейросетей отдельными специализированными задачами, не позволяя создать на их основе универсальный Искусственный Интеллект, так называемый Сильный ИИ.

Для устранения названных недостатков традиционных нейронных сетей необходимо использовать несколько различных искусственных нейронов нового типа, описание которых приведено ниже, а также Иерархическую ПП.

Указанный технический результат для объекта "Иерархическая Память Последовательностей (ИПП)" достигается за счет того, что ИПП состоит из множества соединенных между собой устройств Памяти Последовательностей (ПП) по п.22 так, что каждая пара смежных ПП уровней иерархии N и (N+1) (далее "уровни иерархии М1 и М2") связана множеством искусственных нейронов (далее - искусственные нейроны иерархии - ИНИ).

7.5. Искусственный Нейрон Иерархии (ИНИ) Памяти Последовательностей.

ИНИ содержит Сумматор с функцией активации Сумматора, множество Сенсоров группы А, каждый из которых оснащен функцией активации и ячейкой памяти для размещения Соответствующего значения веса А и размещен на выходе одной из шин Устройства ПП уровня иерархии N, а также множество Сенсоров D, каждый из которых оснащен ячейкой памяти и устройством измерения и изменения по меньшей мере одной из характеристик сигнала и размещен на входах одной из шин ПП уровня иерархии N; причем каждый из Сенсоров группы D связан с выходом Сумматора, а каждый из Сенсоров группы А связан с входом Сумматора, кроме того выход Сумматора оснащен связью со входом одной из шин Устройства ПП верхнего уровня иерархии (N+1); режим обучения ИНИ осуществляют циклами, причем на каждом цикле упорядоченное множество из одного или более сигналов обучения (далее "Окно Внимания") подаются на входы одной или более шин ПП уровня иерархии N, причем сигналы в Окне Внимания упорядочены с использованием функции ослабления. Каждый из сигналов проходит через один или более ИНВ размещенных в уровне иерархии N ПП и названные один или более ИНВ изменяет одну из характеристик сигнала кодирующую вес совместной встречаемости и на выходе каждой из множества шин ПП уровня иерархии N получают сигнал, кодирующий вес совместной встречаемости, из которого извлекается значение веса совместной встречаемости соответствующей шины, и вес передается в Сумматор, где веса, полученные с выходов разных шин, складываются и запоминается значение суммы цикла, после чего Окно Внимания изменяется и цикл обучения повторяется, причем на каждом очередном цикле обучения значение суммы очередного цикла сравнивается со значением суммы предыдущего цикла, и, если значение суммы очередного цикла обучения равно или меньше значения суммы предыдущего цикла обучения, обучение ИНИ останавливается и каждым сенсором группы А названное Соответствующее значение веса А (далее "вес активации"), полученное для цикла обучения с максимальной суммой весов присваивается в качестве значения активации функции активации сенсора А, Сумматор присваивает

функции активации Сумматора значение максимальной суммы весов или присваивает значение числа сенсоров группы А с ненулевыми значениями Соответствующих весов А или присваивает оба названных значения, а каждый сенсор ИНИ группы D на входе каждой из шин ПП уровня иерархии N, на которые были поданы сигналы во время цикла обучения с максимальной суммой весов, измеряет и размещает в ячейке памяти Сенсора D Соответствующее значение D по меньшей мере одной из характеристик сигнала обучения, кодирующей названное значение функции ослабления D сигнала шины в Окне Внимания; в режиме воспроизведения ИНИ сигнал воспроизведения подают на одну или множество шин ПП уровня иерархии N и получают на выходе множества шин ПП уровня иерархии N веса совместной встречаемости и, если вес совместной встречаемости, полученный на выходе шины, равен или превышает значение функции активации сенсора А такой шины, сенсор А посылает в Сумматор или значение веса активации или единичное значение или оба значения, а Сумматор суммирует полученные значения функций активации сенсоров А и сравнивает полученную сумму со значением суммы активации Сумматора и, если суммарное значение равно или превышает значение функции активации Сумматора, то сигнал активации ИНИ подается на выход Сумматора, который затем одновременно поступает на вход одной из шин уровня иерархии (N+1) ПП и на входы сенсоров группы D уровня иерархии N ПП, ячейка памяти каждого из которых содержит названное Соответствующее значение функции ослабления D, причем каждый из названных сенсоров группы D изменяет сигнал Сумматора в соответствии с Соответствующим значением функции ослабления D и подает измененный сигнал на вход соответствующей шины ПП уровня иерархии N или не изменяет сигнала и подает неизмененный сигнал на вход соответствующей шины ПП уровня иерархии N.

В соответствии с заявленным способом, во время цикла ввода последовательности объектов Окна Внимания, Множество Трубы (на выходе матрицы) сравнивают с ранее сохраненным по меньшей мере одним множеством Калибра Трубы, и если отличие Множества Трубы от множества Калибра Трубы сравнимо с некоторой погрешностью, то из Памяти Последовательностей извлекают Генератор Трубы, соответствующий названному множеству Калибра Трубы и используют названный Генератор Трубы в качестве результата поиска (далее "воспоминаний") Памяти Последовательностей в ответ на ввод Окна Внимания в качестве поискового запроса.

7.5.1. Устройство искусственного нейрона иерархии (ИНИ).

Схема Искусственного Нейрона Иерархии Памяти Последовательностей - ИНИ (фиг. 60) отличается от схемы искусственного нейрона традиционных нейросетей - персептрона (фиг. 61) тем, что ИНИ связывает две матрицы Памяти Последовательностей M1 и M2, каждая из которых представляет собой соседние уровни иерархии Памяти Последовательностей, причем входами сумматора ИНИ являются выходы всех шин матрицы нижнего уровня иерархии M1, а единственный выход ИНИ имеет обратную связь со входами всех шин матрицы нижнего уровня иерархии M1 и используется для запоминания Генератора Трубы, в то же время выход ИНИ также является входом одной из шин матрицы M2 и потому связан со всеми шинами матрицы верхнего уровня иерархии M2 по схеме "каждый с каждым".

Коммутация ИНИ с матрицами M1 и M2 показана на фиг. 62.

Хотя полно связанные матрицы разного уровня иерархии M1, M2, ..., MX коммутируются между собой с использованием ИНИ (фиг. 62), благодаря тому что каждый ИНИ является входом одной из шин матрицы следующего уровня, матрицы всех уровней могут выполнены в виде одной матрицы или одной "косынки" (фиг. 63), при этом необходимо, чтобы связи между шинами матриц разного уровня были не "каждый с каждым", а соответствовали схеме коммутации ИНИ (фиг. 62) с помощью сенсоров группы А и D со входами и выходами матрицы уровня M1. Причем соединения "каждый с каждым" выход ИНИ имеет со всеми шинами лишь внутри слоя матрицы M2 (фиг. 63)

Для того чтобы разделить слои, группу сенсоров D размещают в одной из косынок, например в первой косынке - прямоугольная область 1, группе сенсоров А соответствует например последняя шестая косынка - прямоугольная область 6, а группы счетчиков С матрицы каждого уровня расположены в косынках 2, 3, 4 и 5, причем в пересечении шин относящихся к разным уровням иерархии счетчики предпочтительно отсутствуют, а связь между шинами разных уровней иерархии осуществляется только посредством ИНИ и их сенсоров групп А и D. Сумматоры В должны располагаться в любом месте между областями 1 и 6, например, сумматор ИНИ может быть размещен на диагонали в области 7 (обведена кругом) где шина ИНИ пересекается "сама с собой" (фиг. 64).

7.5.2. Сенсоры группы А с функцией активации.

7.5.2.1. Обучение.

В процессе обучения сенсоры группы А имеют функцию активации и работают только, если шина Трубы активна, то есть если на шину Трубы подали сигнал обучения. Будем называть шину такой Трубы или "Активная" или "Горячая" Шина Трубы или Труба. Сенсор на пересечении шины Трубы с шиной частотного объекта Кластера Трубы фиксирует суммарный вес частотного объекта в Кластере Трубы на каждом цикле ввода Окна Внимания (фиг. 65).

На каждом цикле значение суммарного веса частотного объекта $w_{i\Sigma}$ Кластера Трубы в каждом сенсоре А сравнивается с новым значением $w_{(i+1)\Sigma}$ и, если $w_{(i+1)\Sigma} \leq w_{i\Sigma}$, то значение сенсора обнуляется

$w_{(i+1)\Sigma}=0$, а сам сенсор блокируется. Делается это для того, чтобы удалить из Кластера Трубы симметрическую разницу Кластеров ключевых объектов Окна Внимания. Новое ненулевое значение $w_{i\Sigma}$ каждым сенсором А передается в сумматор В где полученное значение суммируется с аналогичными значениями полученными с других сенсоров А с целью расчета значения Калибра Трубы:

$$K_T = \sum_{i=1}^n w_{i\Sigma}$$

Результат суммирования сравнивается Сумматором с результатом предыдущего цикла и, если полученный результат равен или меньше предыдущего, то сумматор В отключает режим обучения Трубы, а максимальное значение $K_{T_{max}}$ суммы весов Калибра Трубы Сумматор сохраняет в качестве значения функции активации Сумматора. После отключения режима обучения Трубы значение $w_{i\Sigma}$ сенсора блока А соответствующее значению $K_{T_{max}}$ сохраняется в сенсоре в качестве порогового значения $w_{i\phi}$ функции активации ϕ сенсора А.

7.5.2.2. Работа.

После обучения все сенсоры А Трубы продолжают фиксировать текущее значение суммарного веса частотного объекта $w_{i\Sigma}$ в текущем Кластере на выходе матрицы М1 и сравнивают его со значением функции активации сенсора $w_{i\phi}$. Если суммарный вес частотного объекта в Текущем Кластере равен или превышает значение функции активации (формула 57 - Условия активации Трубы):

$$w_{i\Sigma} \geq w_{i\phi}$$

то сенсор А посылает единичный сигнал или значение веса активации или оба значения в сумматор В. Сенсоры с нулевыми значениями активации (пассивные сенсоры) в процессе работы не участвуют и сигналов в Сумматор не посылают или их сигналы не учитываются Сумматором.

7.5.3. Сумматор В с функцией активации.

7.5.3.1. Обучение.

На первом цикле обучения Трубы сумматор получает значения суммарного веса частотного объекта со всех шин частотных объектов, а сенсоры всех других шин сумматором блокируются до конца обучения. На каждом цикле сумматор также блокирует сенсоры, значение суммарного веса которых по меньшей мере в двух последовательных циклах не увеличилось $w_{(i+1)\Sigma}=w_{i\Sigma}$. Это позволяет удалить из Кластера Трубы симметрическую разницу Кластеров объектов Окна Внимания.

Полученные с сенсоров значения суммарного веса частотного объекта C_k Кластера Трубы $w_{i\Sigma}^k$ складываются, а значение суммы суммарных весов сравнивается с аналогичным значением, полученным на предыдущем цикле ввода Окна Внимания $w_{(i+1)\Sigma}^k$ и если сумма перестала увеличиваться:

$$\sum_{k=1}^K w_{i\Sigma}^k \geq \sum_{k=1}^N w_{(i+1)\Sigma}^k$$

то сумматор посылает всем сенсорам А сигнал отключения режима обучения, а каждый сенсор с ненулевыми суммарными весом присылает сумматору сигнал готовности. Сигналом готовности может служить значение "единица" и тогда сумма единиц будет равна числу сенсоров или сигналом готовности может служить значение веса и тогда суммой будет суммарный вес встречаемости всех частотных объектов Кластера Трубы. Сумматор складывает сигналы готовности и запоминает сумму в качестве значения функции ϕ активации нейрона ИПП.

7.5.3.2. Работа.

Во время работы сумматор получает сигналы активации от сенсоров группы А и, если сумма сигналов равна значению функции активации ϕ , то Труба активируется и на выход сумматора подается сигнал Трубы, который одновременно поступает на вход шины Трубы в матрице М2 где порождает Кластер для которого шина Трубы является ключевым объектом, а также подается на входы матрицы М1, активируя Генератор Трубы.

7.5.4. Сенсоры группы С (счетчики взаимной встречаемости матрицы М2).

7.5.4.1. Обучение.

Во время обучения Трубы на шине Трубы в группе С сила сигнала обучения Трубы подается без использования функции ослабления, так как во время обучения Труба является самым поздним объектом Окна Внимания матрицы М2 и значит ее сигнал в Окне Внимания М2 не должен ослабляться. Сила сигналов предыдущих Труб в Окне Внимания матрицы М2 ослабляется аналогично тому, как ослабляются сигналы объектов в Окне Внимания матрицы М1. При этом увеличиваются значения счетчиков в пересечениях шин Труб Окна Внимания.

7.5.4.2. Работа.

При активации сумматора В сигнал Трубы подается на вход шины Трубы матрицы С, что порождает Кластер Трубы на выходе матрицы М2.

7.5.5. Сенсоры группы D.

7.5.5.1. Обучение.

В процессе обучения Трубы на вход Матрицы M1 нижнего уровня иерархии подаются сигналы объектов Окна Внимания, пронумерованные/ранжированные в Окне Внимания с использованием функции ослабления. Поэтому для запоминания Генератора Трубы сенсор D каждой из шин объектов Окна Внимания запоминает значения силы сигнала или значение функции ослабления на активных шинах объектов Окна Внимания.

7.5.5.2. Работа.

При активации сумматора В сигнал с него поступает на все сенсоры группы D, где к сигналу применяется сохраненное значение функции ослабления или сигнал приводится в соответствие с сохраненным значением силы сигнала на шине объекта при обучении. Это позволяет подать на вход матрицы M1 сигналы объектов, упорядоченных как в Окне Внимания, которое соответствовало Трубе при ее запоминании.

7.5.6. Последовательность работы ИНИ.

При активации сумматора В сигнал на выходе ИНИ через обратную связь активирует Генератор Трубы на входе матрицы нижнего уровня, одновременно проходя по шине ИНИ через полносвязную матрицу объектов верхнего уровня, где порождает Кластер связей совместной встречаемости с объектами верхнего уровня иерархии, соответствующий направлению обратной связи. Если использовать только связи первого ранга, то множество связей ИНИ прошлого не будет отличаться от множества связей ИНИ будущего и поэтому считывание связей встречаемости ИНИ и других объектов матрицы верхнего уровня не будет зависеть от направления считывания связей ИНИ в матрице верхнего уровня.

Таким образом при активации сумматора В архитектура порождает Генератор ИНИ на входе и соответствующий Генератору Кластер ИНИ на выходе матрицы нижнего уровня, а в качестве ключевого объекта матрицы верхнего уровня ИНИ порождает Кластер частотных объектов на выходе матрицы верхнего уровня.

Очевидно, что Кластер частотных объектов на выходе матрицы верхнего уровня иерархии может активировать сумматор ИНИ более высокого уровня иерархии и так далее, что позволяет рекуррентно возвращать на входы матриц Генераторы Труб все более и более высокого уровня, возвращая тем самым все более и более глубокие воспоминания.

7.5.7. Преимущества ИНИ перед персептроном

Прототипом Искусственного Нейрона Иерархии Памяти Последовательностей (ИНИ) является персептрон традиционных нейросетей, в том числе сверточных. Преимуществами ИНИ перед прототипом являются то, что:

1. ИНИ соединяет две полносвязных матрицы M1 и M2 разного уровня иерархии памяти последовательностей,
2. Входы ИНИ соединены с выходами матрицы M1 что позволяет ИНИ использовать веса, определяемые статистикой совместной встречаемости объектов матрицы M1, а не алгоритмом обратного распространения ошибки, который делает значения веса конкретных входов нейрона непредсказуемыми.
3. Выходной канал сумматора ИНИ соединен упреждающей связью с одним входом матрицы M2 (шина Трубы) и обратной связью с множеством входов матрицы M1 (множество объектов Генератора Трубы),
4. Наличие обратной связи, обеспечивает ИНИ
 - a. автоассоциативность памяти последовательностей, позволяя извлекать из памяти введенные в нее последовательности и их части (Генератор Трубы), которые соответствуют Кластеру весов частотных объектов Трубы, а также
 - b. реализует связь между разными уровнями иерархии полносвязной памяти последовательностей.

Внешне ИНИ отличается от персептрона главным образом наличием обратной связи с Генератором Трубы, однако работа ИНИ существенно отличается от работы персептрона, что позволяет достигнуть следующего технического результата:

1. подавать на входы в ИНИ множество весов, каждый из которых определен статистикой взаимной встречаемости объектов матрицы M1, в то время как веса входов персептрона определяются с помощью абстрактного математического аппарата.
2. ассоциировать множество весов на входе в ИНИ с объектом (шина Трубы) следующего уровня иерархии (в матрице M2), в то время как сигнал с выхода персептрона может быть использован в качестве исходных данных того же абстрактного математического аппарата, что не позволяет идентифицировать объект следующего уровня иерархии;
3. наличие у ИНИ названной обратной связи в процессе обучения ИНИ позволяет ассоциировать названное множество весов взаимной встречаемости объектов Трубы с последовательностью объектов Генератора Трубы
4. наличие у ИНИ названной обратной связи при активации ИНИ позволяет активировать также названную последовательность Генератора Трубы на входе матрицы M1 в качестве воспоминания, что "обостряет" контекст вводимой последовательности (извлекаются воспоминания, соответствующие кон-

тексту вводимой последовательности) и такое действие аналогично эффекту от латерального торможения, имеющему место в коре мозга.

7.6. Слои устойчивых сочетаний.

Устойчивые сочетания, вероятно, являются уровнем иерархии, которые в случае текстовых последовательностей могут быть представлены как аббревиатурами, так и устойчивыми короткими конструкциями (последовательностями) объектов. Например, в примерах "я пошел в кино", "ты пошел в кино" и "он пошел в кино" можно выделить две конструкции: "...пошел в кино", которая не меняется, и более сложная "кто-то пошел в/на..." с изменением местоимения. Тем не менее, устойчивые сочетания могут быть представлены в слое объектов как один из объектов, что позволяет памяти последовательности слоя объектов накапливать статистику совместной встречаемости такого устойчивого сочетания с другими объектами. В общем случае под устойчивым сочетанием следует понимать последовательность объектов, длина которой короче некоторой характерной длины Окна Внимания и значит сочетания не будут обнаружены с использованием Окон Внимания.

Знание устойчивых сочетаний упрощает прогнозирование появления следующего объекта последовательности, если текущий контекст последовательности предполагает использование устойчивого сочетания, начало которого уже в последовательности появилось, а окончание еще только может последовать. В этой ситуации система должна генерировать подсказку и подсказкой должен быть следующий объект устойчивого сочетания или все объекты устойчивого сочетания, упорядоченные в порядке их следования в сочетании. Таким образом, задача прогнозирования появления следующего объекта последовательности с использованием устойчивых сочетаний распадается на две задачи:

1. Выборка множества объектов-гипотез, показатель инверсии которых в паре с последним известным объектом последовательности превышает некоторое пороговое значение показателя инверсии (см. далее 7.4.1).

2. Проверка каждого объекта - гипотезы на соответствие текущему контексту последовательности.

Как отмечалось ранее [3.2.1], идентификация устойчивых сочетаний может решаться в рамках общего подхода к прогнозированию. Мы также говорили (замечание 5), что создания синтетических идентификаторов устойчивых сочетаний и шин для них можно избежать, если использовать прогнозирование появления нового объекта с помощью когерентных ранговых кластеров, в фокусе которых находится неизвестный объект последовательности. Тем не менее, присваивать синтетические идентификаторы устойчивым сочетаниям объектов представляется полезным, так как это помогает строить последовательности из устойчивых сочетаний.

7.6.1. Показатель инверсии.

Одним из способов обнаружения устойчивых сочетаний может быть техника сравнения веса совместной встречаемости объектов в прямом и обратном направлениях, как это было описано выше [2.1.3] и [7.2.1.1]. Для этого напомним, что расположенный в пересечениях каждой пары шин Нейрон Встречаемости содержит не один, а два счетчика по одному для суммирования весов встречаемости двух объектов в каждом из направлений $C_n \rightarrow C_k$ и $C_k \rightarrow C_n$ и именно эти значения встречаемости следует сравнивать с тем, чтобы выделить устойчивые сочетания, для которых отношение веса прямой встречаемости к весу обратной встречаемости (далее "показатель инверсии") должно быть выше некоторой критической величины, характерной для устойчивых сочетаний $I > I_{\max}$ или попадать в небольшой процент $X\%$ пар с самым высоким весом совместной встречаемости:

$$\left(1 - \frac{w}{w_{\max}}\right) * 100\% \leq X\%$$

или одновременно обладать и высокой инверсией и попадать в малый процент пар (формула 58 - Критические значения показателя инверсии и/или веса как условие устойчивости сочетания):

$$\left(\begin{array}{l} I > I_{\max} \\ \left(1 - \frac{w}{w_{\max}}\right) * 100\% \leq X\% \end{array} \right)$$

Тем не менее, измерения только показателя инверсии может оказаться недостаточно, если частота встречаемости конкретных двух объектов, например, существенно ниже средней частоты встречаемости объектов в Памяти Последовательностей. Поэтому может быть полезно дополнительно к инверсии измерять наибольший из двух вес совместной встречаемости (прямой или обратной) и сравнивать его со средним показателем встречаемости для всей матрицы или для устойчивых сочетаний объектов.

Таким образом для выделения устойчивого сочетания объектов и присваивания ему идентификатора объекта следует, по меньшей мере, сравнивать значения показателя инверсии конкретной пары объектов (отношение весов прямой и обратной встречаемости объектов) со средним значением показателя инверсии для других пар объектов в матрице. Если показатель инверсии конкретной пары объектов попадает в $X\%$ с самым высоким показателем инверсии, то принимается решение о том, что пара этих конкретных объектов является устойчивым сочетанием в направлении, которое определено наибольшим весом совместной встречаемости этих объектов. Пара объектов запоминается в качестве Генератора Трубы, их последовательность в устойчивом сочетании задается значением функции ослабления или соответст-

вующей ей функции нумерации, а значения функции запоминаются в качестве весов Счетчиков в пересечении Генератора Трубы с шиной соответствующего объекта. Объединение пар объектов в устойчивые сочетания должно уменьшать число объектов, удовлетворяющих условиям (формула 58).

В частном случае исполнения ПП каждый из названных Искусственных Нейронов Встречаемости (ИНВ) дополнительно оснащен сенсором Инверсии с функцией активации и памятью для считывания веса встречаемости объектов со Счетчиков противоположных направлений встречаемости конкретного ИНВ, а также для определения соотношения весов встречаемости противоположных направлений, причем перед обучением сенсора Инверсии функции активации названного сенсора присваивают пороговое значение, которое сохраняют в памяти, а при обучении на входы по меньшей мере двух шин названного Устройства, соединенные одним из названных ИНВ с Сенсором Инверсии, подают сигналы обучения упорядоченные с использованием функции ослабления так, что измеренная разница названной одной из характеристик сигналов соответствует значению порога активации такого ИНВ, ИНВ активируется и понуждает сенсор Инверсии к считыванию значения Счетчиков в каждом из противоположных направлений встречаемости и сравнению одного значения с другим и, если отношение названных значений превышает пороговое значение, то сенсор Инверсии понуждает послать сигнал на названные по меньшей мере две шины, а полученный сигнал используют как сигнал обучения двух Искусственных Нейронов Встречаемости (далее "Сенсоры D"), установленных на пересечении каждой из названных по меньшей мере двух шин и шины устойчивого сочетания, которую таким образом обучают, причем названные Сенсоры D в качестве веса совместной встречаемости запоминают значения названной функции ослабления, которые отражают порядок следования объектов сочетания.

7.6.2. Текущий контекст устойчивого сочетания.

Понятно, что если последним введенным словом является слово "организация", то это может быть началом устойчивого сочетания "Организация Объединенных Наций", если общий контекст является международным политическим, а не криминальным, например. Если же контекст "криминальный", то продолжением могло быть "организация преступных сообществ", а не "организация объединенных наций". Поэтому множество объектов-гипотез следует проверять на соответствие текущему контексту последовательности.

Можно также предположить, что конкретное устойчивое сочетание может соответствовать нескольким контекстам и значит в случае с устойчивыми сочетаниями запоминание множества контекстов должно быть иницировано устойчивым сочетанием - Генератором Трубы. Для того чтобы избежать порождения бесконечного числа Труб для одной и той же пары объектов напомним о том, что во время обучения матрицы Кластер на выходе постоянно активует Трубы, чьи Кластеры являются подмножествами текущего Кластера. Это должно приводить к активации шины Трубы устойчивого сочетания. Поэтому новая Труба для устойчивого сочетания должна создаваться только в случае, если текущий Кластер не привел к активации Трубы этого конкретного устойчивого сочетания.

Еще одной особенностью образования устойчивых сочетаний является то, что заведомо не известно, сколько именно объектов содержит устойчивое сочетание. Поэтому техника создания Трубы устойчивого сочетания позволяет удлинять устойчивое сочетание "новым объектом" записывая для "удлиненного" сочетания новую Трубу. Число объектов в каждой новой Трубе может увеличиваться до тех пор, пока показатель инверсии встречаемости Трубы очередного устойчивого сочетания с "новым объектом" превышает пороговое значение.

Такое требование приведет к тому, что, для сочетания из трех объектов будет создано две Трубы - первая Труба для сочетания первого и второго объекта, и вторая Труба для сочетания первой Трубы с третьим объектом. Это кажется расточительным, однако создает механизм увеличения длины устойчивых сочетаний и добавления к ним варибельности. Примером варибельности может быть встречаемость сочетания "я иду" с сочетаниями "в кино" или "на стадион", при этом первая Труба может быть создана для сочетания "я иду", вторая труба для "я иду"+"в кино" и Третья труба для "я иду"+"на стадион".

Рассмотрим устройство такого нейрона подробнее.

7.6.3. Нейрон Сочетаний.

На каждом уровне иерархии ПП вес взаимной встречаемости объектов последовательностей отражает меру причину следственной связи между объектами, а Нейрон Сочетаний позволяет выделить устойчивые причинно-следственные связи в каждом из уровней Иерархической ПП. Таким образом поиск причинно-следственных связей сводится к анализу встречаемости объектов разных уровней иерархии. При вводе в ПП любой последовательности она создает текущий Кластер Трубы на выходе матрицы и этот Кластер активует Кластеры Труб, являющиеся подмножествами текущего Кластера Трубы, а те в свою очередь активируют через нейроны ИПП активируют Генераторы сходных по смыслу последовательностей, а также шины следующего уровня иерархии. Все это в итоге должно приводить к активации всех уровней иерархии ПП и к активации Генераторов Труб в каждом уровне иерархии. Выявление устойчивых активированных связей должен обеспечить Нейрон Сочетаний (ИНС).

ПП может быть дополнительно оснащена Искусственными Нейронами Сочетаний (ИНС), а обучение ИНС, состоящего из Сумматора с функцией активации и памятью порогового значения функции ак-

тивации Сумматора, который связан с выходами группы Сенсоров D, а также с группой сенсоров C, вход каждого из которых связан с выходом одной из шин названного множества шин уровня M1 и выход каждого из сенсоров группы C связан с Сумматором, Искусственного Нейрона Сочетаний (ИНС), производится, если сигнал обучения получен с двух названных Сенсоров D, причем полученный сигнал обучения передается в Сумматор, связанный с Сенсорами D, а Сумматор понуждается к активации группы сенсоров C, каждый из которых измеряет и запоминает значение веса взаимной встречаемости на выходе названной одной из шин из множества шин уровня M1, а также возвращает или единицу или названное измеренное значение встречаемости или оба названных значения Сумматору, который суммирует единицы и запоминает число сенсоров C с ненулевыми значениями встречаемости или суммирует веса и запоминает сумму весов всех сенсоров C или суммирует отдельно единицы и веса и запоминает оба названных значения в качестве порогового значения функции активации Сумматора; а в режиме "воспроизведения" каждый сенсор из группы сенсоров C измеряет значение веса и передает или единицу или значение веса или оба эти значения в Сумматор, а сумматор суммирует названные значения и сравнивает их с пороговым значением функции активации Сумматора и, если сумма превышает пороговое значение, то Сумматор подает сигнал воспроизведения на входы названной пары сенсоров группы D, с помощью которых извлекается из памяти сохраненные значения функции ослабления и сигнал "воспроизведения" передается на входы одной из пары шин или на обе шины названного множества шин названного Устройства.

7.6.3.1. Обучение нейрона сочетаний.

Режим обучения является режимом обучения матрицы. Обучение шины устойчивого сочетания происходит при вводе Окна Внимания вводимой последовательности (фиг. 66), причем на наличие устойчивых сочетаний проверяется два самых поздних объекта Окна Внимания. Шина для обучения новому устойчивому сочетанию выбирается или случайно или последовательно берется следующая свободная шина. На выбранную шину слоя шин устойчивых сочетаний в слое объектов подается сигнал обучения, и шина устойчивого сочетания переходит в режим ожидания сигнала обучения от Сенсора F на пересечении шин объектов устойчивого сочетания. На рисунке веса прямой и обратной встречаемости названных объектов соединены сенсором F и показаны кругами разного размера.

При вводе Окна Внимания сигналы поступают в том числе на шины двух самых поздних объектов Окна Внимания и сенсор инверсии F на пересечении названных шин измеряет значение и направление инверсии в пересечении шины первого и второго объектов сочетания и, если выполняются условия устойчивости сочетания (формула 58), то Сенсор F передает сигнал обучения на шины каждого из объектов, а те в свою очередь передают сигнал обучения на активную шину устойчивого сочетания через Нейроны Встречаемости, расположенные на пересечении шины устойчивого сочетания с шинами объектов устойчивого сочетания и названные нейроны запоминают вес совместной встречаемости для шины устойчивого сочетания и каждого из объектов сочетания.

После этого остается связать шину устойчивого словосочетания с контекстом, а именно связать шину устойчивого сочетания с множеством сенсоров группы A (фиг. 66 и фиг. 68), которые связывают Сумматор В шины устойчивого сочетания с шинами Кластера частотных объектов контекста.

Таким образом:

1. Память Последовательностей активирует для обучения шину Трубы слоя сочетаний.
2. при выполнении условий (формула 58) Сенсор F Нейрона Встречаемости на пересечении шин объектов сочетания активирует шины объектов для обучения и посылает сигнал активации шины устойчивого сочетания в нейроны ИНВ на пересечении активной шины устойчивого сочетания и каждой из шин объектов устойчивого сочетания, а каждый ИНВ запоминает вес встречаемости шин объектов с шиной сочетания. В данном случае значением встречаемости может быть или ноль или единица, поскольку шина устойчивого сочетания является выделенной для этих объектов сочетания.
3. Через нейроны ИНВ сигнал Сенсора F поступает в сумматор В и в группу сенсоров А на выходы матрицы Памяти Последовательностей уровня M1 и на один из входов M2 слоя сочетаний.
4. Сумматор В запоминает или число всех сенсоров А, которое равно числу частотных объектов в текущем Кластере на выходе матрицы M1 или полный суммарный вес всех частотных объектов названного Кластера или и то и другое
5. каждый из сенсоров А запоминает текущее значение веса частотного объекта в Кластере Окна Внимания на выходе матрицы M1
6. Сенсоры группы D запоминают ИНИ самый поздний объект сочетания ИНИ пару объектов устойчивого сочетания в качестве Генератора Трубы размещая в сенсорах группы D на пересечении шины трубы и объектов сочетания значения функции ослабления для каждого из объектов, причем для более позднего объекта значение функции ослабления больше(меньше снижает силу сигнала объекта на входе матрицы).
7. Шина устойчивого сочетания в слое M2 создает связи "каждый с каждым" с другими сочетаниями, а именно с предыдущим сочетанием и следующим сочетанием в последовательности сочетаний. Слой M2 может являться:

- а. или слоем объектов и тогда связь с уровнем M2 отсутствует, а шина сочетаний имеет связи "каж-

дый с каждым" в слое объектов;

в. или слоем Труб и тогда шина сочетаний может иметь связи "каждый с каждым" с шинами контекстных Труб и с шинами сочетаний.

Шина устойчивого сочетания может иметь пересечения как со всеми шинами слоя М1 (слой объектов), потому что сочетание часто является новым объектом, так и со всеми шинами слоя М2 (слой сочетаний или Труб), который является слоем устойчивых конструкций. Наличие пересечения со всеми шинами слоя М1 также позволяет наращивать длину устойчивого сочетания за счет добавления к нему новых объектов, с которыми шина устойчивого сочетания образует устойчивые сочетания.

Сигнал, поступивший на вход шины устойчивого сочетания матрицы М2, создает память последовательностей с предыдущим устойчивым сочетанием или Трубой, которое появлялось во вводимой последовательности, причем шины последовательности устойчивых сочетаний образуют Окна Внимания матрицы М2.

7.6.3.2. Работа нейрона сочетаний.

В режиме воспроизведения каждый из сенсоров А сумматора В нейрона сочетаний следит за весом соответствующего частотного объекта в Кластере на выходе матрицы М1 и, если вес частотного объекта соответствует весу активации, то как и в случае с ИНИ сенсор А нейрона сочетаний:

1. или посылает в сумматор единичный сигнал

2. или посылает в сумматор вес частотного объекта в Кластере Сумматор В:

1. или суммирует единичные сигналы от сенсоров группы А и активирует нейрон, если сумма единичных сигналов равна числу частотных объектов Кластера, на котором был научен Нейрон Сочетаний

2. или суммирует веса частотных объектов полученные от сенсоров А и функция активации активирует нейрон сочетаний если сумма весов частотных объектов превышает сумму, на которой был обучен нейрон сочетаний.

3. Или одновременно считает число объектов по п.1 и сумму весов по п.2. И активирует нейрон если выполнены оба условия

При активации сумматор В посылает сигнал по обратной связи в сенсор группы D более позднего объекта устойчивого сочетания или на сенсоры D обоих объектов сочетания и активирует шину более позднего или пары объектов сочетания (Генератора Трубы) что приводит к подаче сигнала в шину более позднего или пары объектов на вход матрицы М1 в качестве "подсказки". "Подсказка" должна использоваться для замены пары шин объектов одной шиной сочетания при вводе следующего Окна Внимания, так как если такой замены не делать, то ввод Окна Внимания состоящего из исходных объектов сочетания не позволит увеличивать число объектов в Трубе сочетаний и как следствие не позволит наращивать длину устойчивых конструкций. Обратная связь может также подавать сигнал на сенсоры D пары объектов устойчивого сочетания с учетом значений функции затухания, на которых был обучен Нейрон Сочетаний; это приводит к подаче сигналов на шины обоих объектов устойчивого сочетания. В первом случае Генератором Трубы устойчивого сочетания будет более поздний объект сочетания, а во втором случае Генератором Трубы будут оба объекта, а порядок их следования будет определен функцией затухания (фиг. 67).

Таким образом, при возникновении на выходе матрицы М1 Кластера на котором был обучен Нейрон Сочетания, такой нейрон активируется и подает на вход матрицы М1 "подсказку" - Генератор Трубы в качестве "воспоминаний" устойчивого сочетания из двух объектов, а также активирует память последовательности устойчивых сочетаний, соответствующих исходной последовательности в матрице М2 устойчивых сочетаний.

7.7. Слой Труб.

Слой Труб располагаются над слоем устойчивых сочетаний и также используют архитектуру косынок и нейроны ИНИ, которые обеспечивают межуровневую коммутацию матрицы разной иерархии памяти последовательностей (фиг. 69).

7.7.1. Субтрубы - подмножества Трубы.

По мере ввода объектов очередной последовательности и суммирования Кластеров объектов на выходе для получения Трубы вводимой последовательности, в матрице будут возбуждаться существующие Трубы (далее "Субтрубы"), в Кластерах которых конкретные набор частотных объектов имеет вес меньший, чем текущий вес этих частотных объектов на выходе матрицы (формула 57). Таким образом Субтрубы являются "подмножествами" Кластера Трубы вводимой последовательности. Это будет приводить к активации шин слоя Труб, идентификаторы которых являются подмножествам текущего Кластера матрицы. Соответственно Кластер Трубы вводимой последовательности в слое Труб будет иметь и сигналы шин Субтруб, ранее созданных матрицей. Так будут появляться Трубы в Трубе, а также активироваться Счетчики в пересечениях шин слоя Труб памяти последовательностей в пересечениях шины Трубы с шинами Субтруб. Это обеспечивает контекстную связь между одинаковыми смыслами (контекстами) различных последовательностей, а также создает иерархию смыслов по мере наполнения ПП от слоя объектов к слою Труб, затем слою Труб 1-го рода и так далее по иерархии ПП.

Физически контекстная связь между Трубами будет возникать в узлах матрицы в пересечении входа вновь порожденной Трубы вводимой последовательности и порожденных ранее Труб. Механизм по-

рождения связей в узлах уже был описан ранее и потому мы не будем останавливаться на этом здесь. Следует, тем не менее, заметить, что подача сигнала на любой из входов "Слой Труб" будет порождать на выходе слоя Труб матрицы Кластер, содержащий в качестве частотных объектов идентификаторы Труб. Такой Кластер является набором смыслов, представленных объектами частотных Труб - подмножеств Трубы такого Кластера.

7.7.2. Слой Труб k-го рода.

Фиг. 69 иллюстрирует добавление в матрицу входов матрицы одного слоя Труб (слой Труб 1-го рода). Последовательные слои Труб мы будем называть Трубами 1-го, 2-го, 3-го и так далее k-го рода. Соседние слои Труб являются матрицами последовательных уровней и связаны нейронами ИНИ. Слой Труб k-го рода, как и все другие слои матрицы, являются полностью связанными и работают одинаково в режиме обучения и в режиме воспроизведения: в режиме обучения счетчики в узлах матрицы запоминают и инкрементально увеличивают вес взаимной встречаемости Труб на пересечении шин которых находится счетчик; в режиме воспроизведения при активации любого входа слоя Труб матрицы порождает Кластер частотных объектов этого слоя Труб и возбуждает Кластеры Субтруб следующего более высокого уровня иерархии на выходе этого слоя матрицы. Сенсоры, расположенные в текущем слое Труб и принадлежащие нейронам (ИНИ) следующего уровня иерархии будут обучаться и активироваться как описано ранее [7.3].

Как было показано Слой объектов, а за ним Слой Труб k-го рода работают аналогичным образом, обеспечивая свертку последовательности объектов, порождаемых предыдущим слоем. Поэтому представленная архитектура матрицы имеет возможность масштабирования как вертикального с увеличением Слоев, так и горизонтального с увеличением числа ранговых блоков. Это означает, что необходимость расширения матрицы может сталкиваться только с техническими ограничениями. Вместе с тем, представляется, что, как и коре мозга, представленной архитектуре матрицы может быть достаточно и нескольких слоев.

7.7.3. Память Последовательностей Труб.

Трубы в качестве объектов Памяти Последовательностей представлены только в соответствующем слое Труб. Поэтому для Трубы 1-го рода, Кластер Трубы (Далее Кластер М1 - Кластер объектов нижнего уровня иерархии), вообще говоря, не содержит частотных объектов, с которыми Труба как ключевой объект слоя М2 встречается в Памяти Последовательностей Слоя Труб (слой верхнего уровня иерархии М2). Такой Кластер М1 это всего лишь контекст, по которому Трубу можно узнать на выходе Слоя Объектов (слой нижней иерархии М1). Тем не менее, Труба является ключевым объектом Памяти Последовательностей Слоя М2 и там ее Кластером является Кластер М2 - Кластер верхнего уровня иерархии, частотными объектами которого являются другие Трубы Слоя Труб (М2). Поскольку каждому частотному объекту Кластера М2 (каждой частотной Трубе, входящей в Кластер М2) соответствует Кластер М1, то множество частотных объектов Кластера М2 можно переписать как множество частотных Кластеров М1 с их частотными объектами М1 и линейная композиция названных частотных Кластеров М1, должна быть тождественна Кластеру М1, который мы присвоили ключевой Трубе в качестве контекста в Слое М1. То есть, построение Обратной Проекции Кластеров М1 каждой из Труб, являющихся частотными объектами Кластера М2 (будущего или прошлого), в качестве результата должно дать нам Кластер М1 ключевой Трубы, для которой был построен Кластер М2 частотных Труб.

Таким образом, иерархическая Память Последовательностей имеет рекуррентные циклические связи между уровнями иерархии и представляет единое целое.

7.7.4. Генераторы Трубы.

7.7.4.1. Окно Внимания постоянной длины.

Как отмечалось выше [5.2.3] следует выбрать максимальную длину Окна Внимания, определенную техническими ограничениями матрицы, например числом ножек микросхемы или другими ограничениями. Желательно, чтобы такое ограничение превышало среднюю длину отрезка ("непрерывный" отрезок) между соседними прерываниями [2.2.8]. Например, средняя длина предложения в русском языке равна 10 слов, так что и длина отрезков между прерываниями (знаками препинания) будет в среднем равна 10 объектам русского языка, а техническим ограничением может служить "максимальная" длина Окна Внимания в несколько десятков объектов. В некоторых случаях длина "непрерывных" отрезков и отрезков с постоянным контекстом может превышать максимальную длину Окна Внимания. Для ввода Генератора Трубы такой "длинной" последовательности в память придется вводить множество последовательных Окон Внимания, максимальной длины. Поэтому далее рассмотрим именно этот случай, как самый общий, а все остальные будут частными случаями этого общего. Рассмотрим последовательный ввод Окон Внимания постоянной длины.

Целью порождения Труб является смысловое сжатие исходной последовательности объектов [4.2.1] и извлечение из Памяти Генератора Трубы. Формула Трубы (формула 39) не учитывает длину Генератора Трубы и, если Генератор Трубы длиннее Окна Внимания, то функцию нумерации объектов в Окне Внимания, которая является обратной для функции ослабления (формула 10 и формула 11), следует расширить на всю длину Генератора Трубы.

На каждом шаге ввода последовательности объектов Генератора Трубы очередь объектов Окна

Внимания сдвигается на один объект в Будущее внутри Генератора Трубы, при этом из очереди Окна внимания выбывает самый ранний объект и добавляется самый поздний объект очереди. Это приводит к тому, что веса выбывших из очереди "самых ранних" объектов будут иметь равный вес - наименьший для Окна Внимания и к этим одинаковым весам необходимо применить функцию нумерации. Для этого может быть использована любая функция нумерации, например та, что была использована для нумерации объектов Окна Внимания. Другим решением может быть отказ от нумерации объектов, выбывшим из Окна Внимания, однако предпочтительным является применение функции нумерации (обратная весовой функции) ко всем объектам Генератора Трубы. Будем считать далее, что веса всех объектов Генератора Трубы определены, а ранжирование весов является функцией нумерации объектов Генератора.

7.7.4.2. Динамическое Окно Внимания.

Работа динамического Окна Внимания была проиллюстрирована на примере с яблоками [4.2.7]. Важным преимуществом динамического Окна Внимания является то, что на каждом шаге в систему можно подавать только один объект, что позволяет устранить ограничения ввода Окна Внимания большого размера. Однако реализация динамического Окна Внимания требует наличия блока, который в примере [4.2.7] был назван "Восстановителем", который бы мог восстанавливать объекты по их Кластерам, обеспечивая рекуррентную связь с предыдущим вводом. Для реализации рекуррентного механизма шины объектов необходимо продублировать новым слоем шин рекуррентной - обратной связи. С учетом сказанного Функциональная схема Памяти Последовательностей (фиг. 32) должна быть дополнена рекуррентной связью, которая и является частью концепции Труб (фиг. 70).

Нетрудно заметить, что архитектура Трубы в качестве рекуррентной связи аналогична архитектуре "Косынка" (фиг. 38), рассмотренной ранее [7.1.1].

Свойствами "Восстановителя" обладают сенсоры группы А искусственных нейронов иерархии, описанные выше. Описанную технику обучения и считывания Труб, можно использовать для обучения и считывания отдельных объектов с той лишь разницей, что Генератором будет служить не множество объектов Окна Внимания, а один объект, которым был порожден Кластер. Аналогично могут запоминаться устойчивые сочетания объектов. Использование искусственных нейронов Труб для создания рекуррентной связи для шин объектов, позволяет создать механизм поиска синонимов, Кластеры которых тождественны с некоторой погрешностью. В этом случае Генератором Кластера может быть множество синонимов, а обратная проекция такого Кластера позволяет определить такое множество синонимов.

7.7.5. Использование ненормализованного Кластера Трубы.

Поскольку мы ожидаем обнаружить на выходе матрицы Кластеры Субтруб, то нормализация всего выходного Кластера Трубы его сравнение с отдельными нормализованными Кластерами Субтруб, являющихся подмножеством Кластера Трубы, может не дать желаемого результата - обнаружить таким способом Субтрубы может оказаться затруднительно. Это способно затруднить использование нормализованного представления Трубы для целей использования Трубы в качестве обратной смысловой связи.

Еще одним ограничением при считывании Кластера может оказаться использование функции ослабления сигналов объектов Окна Внимания на входе матрицы. Использование функции ослабления при запоминании взаимной встречаемости объектов вводимой последовательности было продиктовано использованием ранговой матрицы, вместо матрицы, состоящей из одной косынки [7.1.1]. Однако сила входного сигнала не используется при генерации Кластера, так как в Кластере учитываются только веса попавших в него объектов, а не сила сигнала на их шинах. Кроме того, если бы сила сигналов учитывалась при чтении Кластера на выходе матрицы, то применение весовой функции к входным сигналам объектов Окна Внимания могло бы приводить к включению в Кластер преимущественно частотных объектов самого позднего объекта Окна Внимания, так как значение функции ослабления для него равно единице и его сигнал не ослаблен вообще и чем сильнее функция ослабления, тем сильнее влияние Кластера самого позднего объекта Окна Внимания в выходном Кластере матрицы. При использовании достаточно сильной функции ослабления это могло бы приводить к практической тождественности Кластера самого позднего объекта ОВ и выходного Кластера ОВ на выходе матрицы.

По изложенным выше причинам, в режиме считывания Кластера на входные шины матрицы можно подавать сигналы объектов с учетом функции ослабления силы (для целей нумерации объектов Генератора) или одинаковой силы, а также отказаться от нормализации Кластера на выходе матрицы. Это может упростить архитектуру и конструкцию матрицы, а также снизить сложность ее работы за счет отказа от нормализации векторов для проверки условия коллинеарности и переходу равенству векторов. Под равенством $\bar{\Omega} = \bar{W}$ можно, например, понимать равенство координат или (формула 59)

$$\omega_i - \Delta\omega_i \leq w_i$$

где ω_i - веса частотных объектов запомненной ранее Трубы, а w_i - текущие веса частотных объектов Кластера на выходе из матрицы.

Таким образом, в предпочтительном исполнении матрицы в режиме считывания Кластера на вход матрицы подаются сигналы объектов Окна Внимания с использованием функции ослабления или без ее использования, а на выходе матрицы при измерении весовых коэффициентов частотных объектов Кластера влияния функции ослабления не учитывается и веса не нормализуются, а вместо сравнения норма-

лизованных значений весов полученного Кластера с нормализованными весами ранее созданных в матрице Труб, сравниваются значения ненормализованных весов частотных объектов Кластера с весами частотных объектов Кластеров ранее созданных Труб или Субтруб. Равенство этих весов (формула 59) означает равенство (коллинеарность) векторов полученного Кластера (его части) и ранее записанной Трубы (Субтрубы) тождественного смысла (фиг. 71).

7.7.6. Считывание Трубы для Окна Внимания.

Для считывания Трубы на шины матрицы одновременно или последовательно, или последовательно-параллельно подаются сигналы всех объектов Окна Внимания $\{C_1, C_2, C_3, \dots, C_R\}$ для которой строится Труба с учетом ослабления:

$$T = \sum_{i=1}^R f(i) * K_i$$

где R - это размер Окна Внимания, для которого построена Труба.

Без учета ослабления $f(i)=1$.

7.7.7. Считывание контекста последовательности (Трубы).

Как отмечалось в разделе [4.2] контекстом последовательности является Труба с максимальным значением суммарного веса частотных объектов Кластера Трубы (формула 39):

$$T_{\max} = \text{Cont}_{\max}(R)$$

7.7.7.1. Окно Внимания фиксированной длины.

Можно сравнивать Трубы, построенные на Оках Внимания с постоянным размером R. С началом ввода последовательности сначала размер Окна Внимания увеличивается с одного объекта до R объектов, а по достижении размера R Окно Внимания с вводом каждого нового объекта последовательности из очереди Окна Внимания удаляется самый ранний объект последовательности, в результате чего ОВ как бы сдвигается в будущее вдоль последовательности на один объект. Для каждого n-го Окна Внимания строится Труба T_n :

$$T_n = \sum_{i=1}^R f(i) * K_i$$

И суммарные веса последовательных Труб сравниваются с целью обнаружить точку перегиба кривой суммарного веса $W_{\Sigma,n}$ частотных объектов Кластера Трубы T_n (формула 3), в которой кривая меняет тренд с восходящего на нисходящий и значение T_{\max} принимается равным значению Трубы T_n в найденной точке перегиба, где $W_{\Sigma,n} > W_{\Sigma,(n+1)}$

Если неравенство выполнено, то принимаем $T_{\max}=T_n$.

Последовательность объектов Окна Внимания, породившего Трубу T_{\max} , запоминаем в качестве Генератора Трубы.

7.7.7.2. Окно Внимания переменной длины.

Размер Окна Внимания неограниченно увеличивается начиная с одного объекта последовательности. Размер Окна Внимания может быть ограничен или появлением паузы (рост суммы весов частотных объектов Трубы остановился) или достижением максимального веса частотных объектов Трубы. Для каждого n-го Окна Внимания строится Труба T_n :

$$T_n = \sum_{i=1}^R f(i) * K_i$$

а суммарные веса последовательных Труб сравниваются с целью обнаружить точку перегиба кривой суммарного веса $W_{\Sigma,n}$ частотных объектов Кластера Трубы T_n (формула 3), в которой кривая меняет тренд с восходящего на нисходящий и значение T_{\max} принимается равным значению Трубы T_n в найденной точке перегиба, где

$W_{\Sigma,n} > W_{\Sigma,(n+1)}$

Если неравенство выполнено, то принимаем $T_{\max}=T_n$.

Очевидно, что сравнение более длинных Труб будет давать более гладкую кривую изменения суммарного веса частотных объектов Кластера Трубы.

Трубу T_{\max} запоминаем в группе сенсоров А нейрона ИНИ, а последовательность объектов Окна Внимания, породившего Трубу T_{\max} запоминаем в группе сенсоров D нейрона ИНИ в качестве Генератора Трубы.

7.8. Многопоточная Память Последовательностей.

Последовательности событий/объектов, полученные разными органами чувств, являются событиями/объектами разной природы и должны быть представлены памятью разных последовательностей. Это означает, что в матрице существуют слои объектов разной природы, которые не имеют пересечений друг с другом в слое Памяти Последовательностей объектов, но связаны Искусственными Нейронами Метки (ИНМ). ИНМ позволяют синхронизировать во времени и пространстве последовательности объектов разной природы, полученные по разным каналам получения информации. Синхронизация во времени и

пространстве позволяет обнаружить совместную встречаемость событий разной природы, например, события которые мы слышим и которые видим. ИМН должен быть похож на ИНИ, но должен активироваться любой из параллельных (одновременных) последовательностей, то есть, последовательность мяукания кошки (Генератор "звуковой" Трубы) полученная нами по каналу слуха, должна быть воспроизведена одновременно со вводом последовательности ее зрительных образов полученных по каналу зрения (Генератор "зрительной" Трубы).

Для этого каждый из слоев Памяти Последовательностей объектов разной природы М1 должен, например, иметь собственную группу сенсоров А и собственный сумматор В, а также собственную группу D сенсоров Генератора Трубы, а выходы всех сумматоров должны быть соединены со входом одной и той же Трубы слоя М2. Таким образом активация сумматора любого из слоев объектов разной природы уровня М1 приведет к подаче сигнала на шину Трубы и на все группы D сенсоров Генератора Трубы. Причем Генератор Трубы, состоящий из нескольких Генераторов объектов разной природы, при активации Трубы в режиме "воспроизведения" только одной из групп А сенсоров разной природы, должен активировать все Генераторы разной природы, и они должны вводиться в косынку одновременно, так же как они вводились бы в реальности, воспроизводя одновременно и звук мяукания и вид кошки, например.

Другим решением по синхронизации может служить слой Труб первого рода, в котором Трубы из объектов разной природы смешаны и имеют связи "каждый с каждым" в слое Труб. Это позволяет Трубе зрительных образов кошки оказаться в последовательности Труб рядом с Трубой звуковых образов мяукания кошки и взаимная встречаемость таких Труб будет иметь высокий вес совместной встречаемости, что в представленной концепции Памяти Последовательностей означает высокую степень связи звуковых и зрительных образов кошки друг с другом. В таком случае достаточно нейронов ИНИ и нейроны ИНИМ могут для этого не понадобиться. Вместе с тем, в таком представлении мы приходим к устойчивым сочетаниям Труб и необходимости использования нейронов устойчивых сочетаний ИНС в слое Труб, что кажется логичным, с точки зрения необходимости как можно более одинаковой организации всех слоев иерархии Памяти Последовательности.

Слои измерений служат слоями объектов отличной от других природы и потому в каждом слое измерений должна быть своя группа сенсоров А, D и свой сумматор В, а шина Трубы М2 может быть общая с объектами другой природы или отдельной, но имеющей связи "каждый с каждым" в слое Труб. Для появления Трубы для слоя измерений для конкретных моментов времени слоя измерения времени или для отдельных местоположений слоя измерений пространства необходимо создавать Трубы и значит нужна причина, которая служит триггером создания Трубы времени или пространства, такой же триггер каким служит максимум суммы весов Калибра Трубы для Трубы контекста последовательности в слое объектов, который мы рассматривали на примере текстовой информации. Поскольку само по себе время и пространство не несут контекстной нагрузки, то видимо в качестве триггера создания Трубы в слое измерений следует считать создание Трубы контекста в любом из слоев объектов разной природы. В таком случае Трубы измерений будет служить "меткой" измерений для Трубы объектов конкретной природы: Труба связанная с появлением зрительного образа кошки будет иметь метку времени и Труба, связанная с появлением звукового образа кошки будет также иметь свою метку времени, возможно метки будут совпадать, но в общем случае могут и несколько отличаться и необходим способ "округления" меток для их сопоставления как одновременных. Учитывая, что Труба измерений относится к конкретному объекту слоя Труб - к Трубе объектов конкретной природы, то в этом случае лучше использовать Искусственные Нейроны Метки, который будет привязывать метку слоя измерений к Трубе контекста объектов конкретной природы, а активироваться Труб будет или при введении метки (или метки с округлением) в слой измерений или при появлении Калибра Трубы (контекста) на выходе косынки объектов конкретной природы.

7.9. Слой синхронизации.

7.9.1. Архитектура систем измерения и исчисления.

ИПП, как правило, дополнительно оснащена с образованием устройства измерения длины метки (далее - УИДМ) одной или более следующими друг за другом группами измерительных шин (далее "слой измерений"), причем для каждой группы выбирается система исчисления и оснащается числом шин, соответствующим избранной системе исчисления - две шины для двоичной, три для троичной системы исчисления и так далее, и каждая из шин соединяется с источником сигнала; на шину подается сигнал, если значение равно единице и сигнал на шину не подается, если значение равно нулю; затем определяется направление увеличения разрядности групп от групп меньшей разрядности к группам большей разрядности измерений и каждой шине одной группы разрядности присваивается одинаковая мера длины так, что в смежных группах разрядности мера длины одной шины группы большей разрядности равна сумме мер длины всех шин меньшей группы разрядности; а для измерения длины выход каждой шины связывается со входом Сенсора А1, а выход каждого Сенсора А1 связывается со входом вычислителя длины метки, на первом шаге исполнения вычислитель рассчитывает "групповую длину", для этого суммируют шины с включенным сигналом в каждой группе разрядности и умножают сумму на произведение чисел, каждое из которых представляет собой число всех шин в каждой из групп меньшей разряд-

ности или единицу в случае, если группа меньшей разрядности отсутствует, а на втором шаге вычислитель длины метки суммирует все групповые длины, а полученная сумма используется в качестве измеренной длины метки.

В УИДМ с целью генерации сигналов названным источником сигнала длина представлена последовательностью одной или более следующих друг за другом групп значений, причем каждое из значений может быть или нулем, или единицей, для каждой группы выбрана система исчисления и в группе размещено число названных значений, соответствующее системе исчисления - два разряда для двоичной, три для троичной системы исчисления и так далее, для последовательности групп определено направление увеличения разрядности групп от групп меньшей разрядности к группам большей разрядности измерений, а при увеличении измеряемой величины на единицу соответствующего разряда выбирают значение такого разряда равное нулю и устанавливают значение равным единице, а если значения такого разряда равного нулю нет, то все значения такого разряда кроме одного приравнивают нулю, а также находят в группе большей разрядности значение равное нулю и приравнивают это значение единице; при уменьшении измеряемой величины на единицу соответствующего разряда выбирают значение такого разряда равное единице и устанавливают значение равным нулю, а если значения такого разряда равного единице нет, то все значения такого разряда кроме одного приравнивают единице, а также находят в группе большей разрядности значение равное единице и приравнивают это значение нулю; определяют групповую длину, для чего суммируют значения соответствующей группы и умножают сумму на число всех значений в смежной группе меньшей разрядности или на единицу, если группа меньшей разрядности отсутствует, а затем складывают все групповые длины и сумму групповых длин используют в качестве измеренной длины метки.

7.9.1.1. Способ измерения и округления.

Абсолютные измерения.

Рассмотрим пример архитектуры слоя синхронизации Памяти Последовательностей.

Предположим слой состоит из девяти "частотных шин" - первые три первого разряда (1,2,3), затем три второго разряда (4,5,6) и последние три третьего разряда (7,8,9), а также из генератора частоты 1 Гц. Рассмотрим полный цикл работы частотных шин:

1. Шины первого разряда. Шины 1,2,3 включаются и выключаются последовательно с частотой 1 Гц (каждую секунду) в порядке нумерации и выключаются с концом 3-х секундного цикла. Соответственно шина 1 остается включенной 3 секунды, шина 2 остается включенной 2 секунды и шина 3 остается включенной 1 секунду. Затем цикл включения шин повторяется снова. Таким образом, включение шин 1,2,3 можно представить бесконечным рядом 1,2,3, 1,2,3, 1,2,3, 1,2,3, 1,2,3, 1,2,3, 1,2,3, 1,2,3..... в котором каждая цифра означает включение шины с соответствующим номером.

2. Шины второго разряда. Каждая из шин 4,5,6 включается после очередного полного цикла включения шин 1,2,3, и остается включенной до конца цикла 9-го цикла. Соответственно шина 4 остается включенной 9 с, шина 5 остается включенной 6 секунд и шина 6 остается включенной 3 секунды. Затем цикл включения шин возобновляется снова. То есть шины 4,5,6 включаются последовательно с частотой 0,33 Гц (каждые три секунды) в порядке нумерации и через каждые 9 с цикл включения шин повторяется. Таким образом, включение шин 4,5,6 можно представить бесконечным рядом 4,5,6, 4,5,6, 4,5,6, 4,5,6, 4,5,6, 4,5,6, 4,5,6..... в котором каждая цифра означает включение шины с соответствующим номером.

3. Шины третьего разряда. Каждая из шин 7,8,9 включается после полного цикла включения шин 4,5,6. То есть шины 7,8,9 включаются последовательно с частотой 0,111 Гц (каждые девять секунд) в порядке нумерации. И если в матрице нет шин большего разряда, то после 27-секундного цикла включения шин процесс включения шин матрицы или прекращается, или начинается снова с п. 1.

Продемонстрируем соответствие между включением шин и временем в виде таблицы (табл. 1).

Таблица 1

| | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Разряд 1 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | |
| Разряд 2 | | | | 4 | | | | | | 6 | | | 4 | | | 5 | | | | 6 | |
| Разряд 3 | | | | | | | | | | 7 | | | | | | | | | | 8 | |
| Секунды | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| | | | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | |

Как видно из табл., 2 секундам соответствует сигналы двух шин 1 и 2 (далее будем просто указывать старшую из шин разряда, в данном случае шину 2), а например 17 секундам соответствует набор сигналов (далее "метка" измерения) шин: 2,5,7. В примере мы использовали троичную систему исчисления, но могли использовать двоичную или десятичную, или любую другую.

Важно отметить, что каждая из частотных шин 4,5,6 является, как бы, Трубой, продолжительность которой можно выразить одним и тем же набором шин 1,2,3. А каждая из шин 7,8,9 является Трубой для одного и того же набора шин 4,5,6. И так далее. Таким образом, окончание цикла нижнего разряда должен запускать новую шину верхнего разряда. Механизм создания Труб позволяет создать слой "измеряемых величин", причем весь этот слой и его части могут быть настроены на любую систему исчисления - двоичную, троичную, четверичную и так далее..... Для двоичной системы каждая Труба следующего слоя должна быть построена над двумя шинами предыдущего уровня, для троичной над тремя, для четверичной над четырьмя и так далее...

Как видно, Трубой в данном примере мы по сути назвали полный размер нижнего слоя - продолжительность (если это время) или длину (если это расстояние) или угловой размер (если это угловые градусы) и т.д.

Очевидно, что полное число включенных частотных шин, умноженное на периодичность их включения, соответствует полному времени в секундах, которое потребовалось для включения этих шин. Например, одновременное включение шин определенное формулой 2,5,7 соответствует включению шин (1,2),(4,5),(7) и времени (формула 60 Пример расчета разницы меток измерения):

$$(2 * 1\text{sec}) + (2 * 3\text{sec}) + (1 * 9\text{sec}) = 17\text{sec}$$

Включение шин (1,2), (4,5), (7) можно изобразить, как показано ниже (фиг. 72). Измерение длины Трубы.

Будем считать, что для слоя измерений Калибром слоя измерений является разница двух последовательных измерений.

Предположим (табл. 1) для двух последовательных измерений мы имеем две метки измерения {3,5,7,11} что соответствует времени $(3*1+2*3+1*9+3*27)=99\text{sec}$ и {3,4,8,11} что соответствует времени $(3*1+1*3+2*9+3*27)=105\text{sec}$.

Определим разницу меток как разницу в числе шин соответствующего разряда и тогда разница между метками $\{3,4,8,11\} - \{3,5,7,11\} = \{0, -1, +1, 0\}$, что соответствует времени $(0*1-1*3+1*9+0*27) = 6\text{sec}$.

Таким образом, вычитая из последней метки предыдущую, мы можем рассчитать длительность между последовательными событиями, например, между событиями начала и окончания обучения Трубы контекста, которую мы также будем называть "Длиной Трубы" контекста.

В приведенном примере мы использовали запись, в которой номера шин были записаны в возрастающем порядке, что соответствует записи, в которой большие разряды записаны справа, а меньшие слева. Это не совпадает с записью чисел, где старшие разряды находятся слева. Если следовать традиционной записи разрядности чисел слева направо, то формулу (формула 60) можно переписать так:

$$\{11,8,4,3\} - \{11,7,5,3\} = \{0, +1, -1, 0\},$$

что все также соответствует промежутку времени $(0*27+1*9-1*3+0*1)=6\text{sec}$.

Округление измерений.

Расчет Длины трубы можно провести имея Метку начала и Метку окончания, однако конкретный Сенсор А1 может оперировать только значением одной шины и, если поставить задачу определения длины на уровне Сенсоров группы А1, то можно воспользоваться методикой округления длины.

Результат, который мы получили выше (формула 60) имеет тот недостаток, что в нем приходится одновременно оперировать и положительными и отрицательными величинами, а представленная архитектура оперирует только положительными (есть сигнал или его нет). Для вычисления Длины можно использовать цифровую обработку, а можно расширить архитектуру, добавив отрицательные, но тогда шин будет вдвое больше. Если мы не хотим увеличивать число шин и цифровую обработку, то можно использовать технику округления. Для этого, например, можно исключить отрицательные значения и считать равными нулю (сигнала нет).

В предыдущем примере (формула 60) мы получили $\{0,-1,1,0\}$ и после округления получим $\{0,0,1,0\}$, что соответствует округленной разнице в 9sec вместо точной разницы в 6sec. Предложенная техника основана на том факте, что первая отрицательная разница (в направлении от больших разрядов к меньшим) возникает как раз в разряде, который определяет старший разряд погрешности округления и потому его следует округлять. Округление имеет тот же порядок величины, что и точное измерение. Это вполне напоминает свойство человеческой памяти. Вместе с тем, мы округлили размер (длину или продолжительность) Трубы, но ниже опишем, как найти Трубу по точной метке времени.

Округление в матрице можно проиллюстрировать следующим рисунком (фиг. 73).

Логическая операция округления записана ниже в табл. 2.

Таблица 2

Логическая операция округления двух измерений на одной шине

| Тип | Второе значение (N+1) | Первое значение N | Результат округления |
|-----|-----------------------|-------------------|----------------------|
| 1 | 1 | 1 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 |

Тем не менее, использованная для режима обучения логика округления (табл. 2) исходила из того, что метка начала N всегда меньше метки окончания N+1. Такое предположение может быть не справедливым для систем измерения отличных от времени, которое движется только вперед. Например, при измерении расстояния от точки до точки оно может как увеличиваться, так и уменьшаться, хотя пройденный путь может только увеличиваться. То же самое можно сказать о поворотах, при том, что повороты в противоположные стороны соответствуют угловым величинам с обратными знаками, сумма всех углов поворотов всегда растет. Если использовать шкалу пройденного пути, то логикой (табл. 2) все также можно пользоваться. В других случаях следует изменить логику округления сенсора С. Кроме того,

предположение, что метка начала N всегда меньше метки окончания $N+1$, не справедливо при любом поиске по метке, потому что метка поиска может быть как меньше, так и больше метки конкретного события (Трубы).

В то время, как всю метку видит Сумматор, каждый из сенсоров $A1$ оперирует лишь значением одной из частотных шин и не может знать какая из меток больше. Поэтому можно использовать логику округления (табл. 3), имея в виду, что такая логика потенциально имеет вдвое большую ошибку округления, что при поиске должно приводить к зашумлению - выдаче значительно большего числа воспоминаний, соответствующих метке поиска, а значит может понадобиться дополнительный фильтр.

Таблица 3

Логическая операция округления двух измерений на одной шине с двойной ошибкой

| Тип | Одно значение | Другой значение | Результат округления |
|-----|---------------|-----------------|----------------------|
| 1 | 1 | 1 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 1 |

Предложенная логика (табл. 3) нейтральна к порядку ввода значений и нейтральна к тому какое из значений больше и какое меньше.

Знакомый с уровнем техники может предложить другую логику округления, не выходя за уровень техники, определенный настоящей работой.

Сравнение измерений.

Длина одинаковых событий может совпадать лишь приблизительно. В частности, при измерении длины одного и того же события разными сенсорами (например ушами и глазами) длина Трубы (события) может отличаться. Поэтому при поиске необходимо иметь возможность сравнения округленных значений длины Трубы и иметь возможность сравнивать Трубы с началом и окончанием в некоторой окрестности. Нужен нечеткий поиск (fuzzy search), который позволит найти Трубы метка начала которых лежит в некоторых пределах измерений. Для этого удобно сравнивать точную или округленную длину Трубы с погрешностью заданного измерения.

Еще одной задачей поиска является то, что, например, если в качестве метки поиска (МП) введен январь месяц 2000 года, то все Трубы с меткой начала (МН) в январе, длина которых измеряется в секундах, минутах, часах, днях и неделях с полной длительностью не более месяца могут соответствовать такому запросу. В приведенном примере погрешность была задана младшим разрядом метки поиска - один месяц.

Если в качестве метки запроса введено длина в один километр, то любую Трубу, которая имеет длину менее километра можно считать соответствующим запросу.

Если оперировать длиной Трубы округленной с помощью логики (табл. 2 или 3), то поскольку такая длина приблизительно, то ее сравнение будет происходить в более широких пределах, чем при сравнении точной длины просто потому, что точность с которой было проведено округление неизвестна.

В связи с вышеизложенным, представляется необходимым ввести правила сравнения, которые бы позволяли решать описанные задачи поиска.

Определение 7.

Сравнимыми можно считать два результата округления длины, значения которых:

1. или, например, расположены в группе шин одного и того же разряда $МП \approx МН$.
2. Или, например, имеют ненулевые значения смежных разрядов $МП \sim МН$, а именно $МП \leq МН$ или $МП \geq МН$.
3. Или, если сравниваемые результаты округления отличаются числом значений младшего разряда, то есть значения длин $МП > МН$.

Примеры сравнения для каждого из случаев (определение 7):

1. В первом случае сравнимыми с значением $\{0,0,1,0\}$ можно считать или само значение $\{0,0,1,0\}$ или $\{0,0,2,0\}$ или $\{0,0,3,0\}$.

2. Во втором случае сравнимыми с значением $\{0,0,1,0\}$ можно считать или $\{0,0,1,1\}$ или $\{0,0,0,1\}$ или $\{0,1,1,0\}$ или $\{0,1,0,0\}$, а также, например, $\{0,2,0,0\}$ или $\{0,3,0,0\}$ и так далее.

3. В третьем случае, сравнимыми со значением $\{0,0,1,0\}$ можно считать или $\{1,1,0,0\}$ или $\{1,2,0,0\}$ или $\{3,3,0,0\}$ и так далее.

7.9.1.2. Слои измерений.

Слой измерений состоит из слоя частотных шин и слоя меток (фиг. 74).

Частотные шины.

Частотные шины слоя измерений не имеют соединений "каждый с каждым".

Вместо этого шины самой малой разрядности имеют генератор, который поочередно включает шины разряда с первой по последнюю (далее "цикл разряда") так что к ранее включенным шинам ("активные" шины) добавляются новые активные шины того же разряда. Шины каждого разряда включаются через равные промежутки времени (далее "шаг разряда"), причем выключение предыдущей вызывает включение следующей, пока не достигнута последняя шина разряда. Последняя шина разряда N :

1. выключает все шины разряда N кроме первой и цикл разряда N повторяется сначала.
2. включает очередной шаг разряда N+1.

Таким образом цикл разряда N равен одному шагу разряда N+1, а сумма шагов разряда N равна циклу разряда N.

Шины меток измерений.

Поскольку каждая метка измерений содержит набор активных частотных шин, то для запоминания меток разумно использовать дополнительный слой Меток косынки, который является слоем Труб измерений над слоем частотных шин. Шина метки измерений в слое Труб измерений является одной из шин слоя Труб 1-го рода (Трубы над слоем объектов) и поэтому имеет связи "каждый с каждым" со всеми Трубами контекста, что позволяет связать метку измерений с любой из Труб контекста или с несколькими Трубами контекста, если метки Труб совпадают, например, Трубы были образованы в одном и том же месте пространства или в одно и то же время. Труба контекста тоже может быть связан не с одной, а несколькими метками, если, например, одна и та же Труба контекста возникала в разное время или в разных местах. Слой меток представлен полносвязным с тем, чтобы можно было хранить последовательность меток, что позволяет "отмотать назад" время или пройденный путь или другое измерение.

7.9.2. Устройство и работа Искусственного Нейрона Метки измерений (ИНМ).

В ИПШ в качестве названного вычислителя длины метки используется Искусственный Нейрон Метки (ИНМ), который кроме названных сенсоров А1 оснащен множеством Сенсоров С, причем каждый Сенсор С оснащен по меньшей мере тремя соединениями, ИНМ также оснащен Сумматором В1 с функцией активации, памятью и вычислителем, а также Сумматор В1 оснащен входом и выходом; Первым соединением Сенсор С соединен с выходом Сумматора В1, а Вторым соединением Сенсор С соединен с выходом Сумматора одного из множества ИНИ названного Устройства, и Третьим соединением Сенсор С соединен с входами множества Сенсоров D названного ИНИ; вход Сумматора В1 соединен с выходами множества Сенсоров А1, вход каждого из которых соединен с одной из измерительных шин названного УИДМ; используется ИНМ в режиме обучения и в режиме воспроизведения; в режиме обучения на вход Сумматора В1 подается сигнал активации; который затем передается на выход Сумматора В1 и далее на Первые соединения множества Сенсоров С, каждый из которых переходит в режим ожидания сигнала обучения ИНИ на своем Втором соединении, и, когда на Втором соединении появляется сигнал обучения ИНИ, то Сенсор С передает сигнал обучения через Первое соединение Сенсора С в Сумматор В1 ИНМ, а сам Сенсор С понуждается к установке соединения между Первым соединением и Третьим соединением Сенсора С для использования в режиме "воспроизведения", что позволяет в режиме воспроизведения передавать сигнал с выхода Сумматора В1 на входы множества Сенсоров D; при подаче сигнала активации на вход Сумматора В1 все Сенсоры А1, соединенные со входом Сумматора В1, понуждаются к измерению наличия сигнала в измерительной шине, а затем к передаче Сумматору В1 в качестве Первого значения "ноль", если на шине сигнала нет, или значение "единица", если сигнал на шине есть, а полученные от всех Сенсоров А1 названные Первые значения используются Сумматором В1 для вычисления Первой длины метки и размещения Первой длины метки в памяти; после поступления сигнала обучения ИНИ через Сенсор С в Сумматор В1, Сумматор В1 понуждает Сенсоры А1 к повторному измерению наличия сигнала в измерительной шине, а затем к передаче Сумматору В1 в качестве Второго значения "ноль", если на шине сигнала нет, или значения "единица", если сигнал на шине есть, а полученные от всех Сенсоров А1 названные Вторые значения, Сумматор В1 использует для вычисления Второй длины метки и размещает Вторую длину метки в памяти; Сумматор В1 извлекает из памяти Первую и Вторую длину метки и вычисляет разницу Первой и Второй длин метки и сохраняют ее в памяти в качестве значения функции активации Сумматора В1; в режиме воспроизведения Сенсор А1 измеряет наличие сигнала в измерительной шине, а затем передает Сумматору В1 в качестве Третьего значения "ноль", если на шине сигнала нет, или значение "единица", если сигнал на шине есть, а после получения от всех Сенсоров А1 Третьего значения Сумматор В1 вычисляет Третью длину метки, а затем вычисляет разницу между Первой длиной метки и Третьей длиной метки и сравнивает полученный результат с названным значением функции активации Сумматора В1 с использованием алгоритма сравнения, результатом работы которого является вывод "сравнимы" или "не сравнимы", и, если Третья и Четвертая длины метки "сравнимы", то Сумматор В1 подает сигнал активации на выход и сигнал активации через Сенсор С передается в названную группу Сенсоров D нейрона ИНИ.

ИНМ предназначен для активации Генератора ИНИ и Генератора ИНМ при вводе метки измерений или длины Трубы в качестве поискового запроса к ПШ.

Архитектура слоев матрицы, содержащей Слой частотных шин М1, Слой объектов М1, а также Слой меток измерений М2 и "Слой М2 Труб контекста" соответствует архитектуре других слоев матрицы (фиг. 76). Однако не все узлы матрицы используются для коммутации ИНМ и потому мы перерисуем матрицу чтобы показать те что используются (фиг. 77).

"Слой меток измерений М2" сам по себе является полносвязным и может запоминать последовательность меток измерения. Он также образует следующие группы соединений с другими слоями:

1. По меньшей мере одну группу соединений "каждый с каждым" в "Слое М1 объектов". Группа соединений обозначена D1, (фиг. 77). Тем не менее предпочтительной является архитектура с двумя груп-

пами соединений E1 и D1 в "слое M1 объектов".

2. Предпочтительно по меньшей мере одну группу соединений "каждый с каждым" в "Слое M2 Труб контекста". Группа соединений обозначена C (фиг. 77).

3. По меньшей мере одну косынку или матрицу crossbar соединений в Слое частотных шин M1. Группа соединений обозначена A1, (фиг. 77).

4. Очередная шина метки из "Слоя меток измерений M2" (фиг. 77) активируется одновременно с шиной Трубы "Слоя M2 Труб контекста", связанный нейронами ИНИ [7.3] со "Слоем M1 объектов". Проллюстрируем работу слоя меток измерений на примере всего двух нейронов ИНИ и ИНМ (фиг. 78).

7.9.3. Функционирование ИНМ.

7.9.3.1. Обучение.

Задачей обучения ИНМ является запоминание метки - момента начала и продолжительности обучения ИНИ, а также создание связи между созданной меткой и Генератором ИНИ.

На каждую из Слоя частотных шин M1 Генератор G циклически подает сигналы активации шин как описано выше [7.7.1.1].

В режиме обучения всегда активны по меньшей мере одна шина ИНМ и одна шина ИНИ. Поскольку шина метки служит фиксации времени и продолжительности обучения ИНИ, то активность этих двух шин синхронна - при включении новой шины ИНИ для обучения ("начало обучения"), включается и очередная свободная шина ИНМ для запоминания метки для ИНИ, а с окончанием обучения ИНИ ("конец обучения") его шина выключается и выключает шину ИНМ, который должен запомнить метку для ИНИ и продолжительность создания Трубы для ИНИ.

7.9.3.2. Процесс воспроизведения (поиска).

Задачей воспроизведения является воспроизведение Генератора трубы контекста и Генератора метки ИНМ в ответ на ввод метки измерения в слой измерений. Введенная метка представляет собой поисковый запрос и по сути является меткой в окрестности округления Генератора ИНМ. В режиме воспроизведения в ПП вводят метку измерения, которая активирует ИНМ, а он в свою очередь активирует Генератор ИНИ на входе в матрицу. Генератор ИНМ может быть активирован также через сенсор C при активации ИНИ.

7.9.3.3. Сенсоры группы A1.

Каждый из сенсоров группы A1 установлен в пересечении одной из шин Слоя частотных шин ("частотная шина") и одной из шин Слоя меток измерений M2 ("шина Метки"). Сенсор группы A1 является нейроном типа ИВК и при подаче сигнала на одну из шин слоя Труб (активная Труба) все нейроны на пересечении активной Трубы с активными шинами слоя измерений изменяют значение веса встречаемости с исходного значения "не встречались" на значение "встречались", например, с нуля на единицу. Таким образом после активации шины Трубы все ИВК на пересечении шины Трубы с шинами метки измерения получают значение веса равное 1, а все остальные будут или закрыты, или будут иметь значение ноль.

В режиме обучения с точным замером Сенсор A1 может запоминать и хранить или не запоминать и не хранить значений состояний начала и окончания. В первом случае сохраненные данные отправляются в Сумматор B1 по его запросу или используются Сенсором A1 для вычисления точной или округленной длины Трубы, а в последнем случае сенсор A1 сразу отправляет в Сумматор B1 измеренное состояние частотной шины для "Метки начала" и состояние частотной шины на момент "Метки окончания".

В режиме "воспоминания" сенсор A1 активируется при вводе в слой частотных шин "метки поиска" в качестве поискового запроса. А именно при вводе некоторой метки в качестве поискового запроса сенсор A1 измеряет состояние частотной шины и отправляет значение в Сумматор B1.

7.9.3.4. Сенсоры группы D1 и E1.

Сенсоры группы D1 полностью аналогичны сенсорам группы D нейрона ИНИ и запоминают тот же Генератор Трубы, что и ИНИ. Поэтому ИНМ может использовать группу собственных сенсоров D1 для запоминания Генератора Трубы или использовать группу сенсоров D Генератора Трубы ИНИ. Предпочтительным представляется использовать одну группу сенсоров D, это упростит конструкцию и снизит стоимость.

Сенсоры группы E1 нейрона ИНМ аналогичны сенсорам D нейрона ИНИ и предназначены для запоминания и активации Генератора ИНМ, однако в отличие от сенсоров D в случае с сенсорами E1 нет необходимости запоминать порядок следования измерительных шин, поскольку в слое измерений функция ослабления не используется. Поэтому каждый из сенсоров E1 может иметь только два состояния - открыт или закрыт. Все сенсоры E1 Генератора ИНМ переводятся в состояние "открыт".

7.9.3.5. Сенсор C.

В режиме обучения сенсор C ведет себя как нейрон ИНВ, который установлен на пересечении шин ИНМ и ИНИ и фиксирует вес совместной встречаемости объектов ИНМ и ИНИ. Однако конкретная метка начала может быть присвоена Трубе контекста лишь один раз и поэтому повторных обучений ИНИ на той же метке ИНМ быть не должно. Это позволяет сенсору C запоминать лишь два состояния - связь есть или связи нет.

В режиме воспроизведения Сенсор C должен обеспечивать активацию Генератора ИНИ и Генера-

тора ИНМ или при активации ИНИ или при активации ИНМ. Таким образом, оба названных Генератора активируются или при появлении на выходе матрицы Кластера контекста, способного активировать ИНИ или если была введена такая метка измерений, способная активировать ИНМ.

В режиме обучения выходы Сумматора В1 (ИНМ) и Сумматора В (ИНИ) могут активироваться одновременно или по очереди, сначала один, а затем другой. В последнем случае сенсор С может служить мостом для передачи сигнала активации с первой шины на вторую. При одновременной активации шин ИНИ и ИНМ в момент начала обучения, сенсор С должен запомнить связь между шинами ИНИ и ИНМ. Запоминание Сенсором С связи между ИНИ и ИНМ может происходить в ответ на появление разницы в характеристиках сигналов активации ИНИ и ИНМ, или наоборот, в ответ на отсутствие разницы в характеристиках названных сигналов.

В режиме воспроизведения работа сенсора С должна определять какая именно группа сенсоров D или D1 будет использоваться. Чтобы избежать повторной активации Генератора Трубы (группа сенсоров D является копией группы D1) предпочтительной представляется работа сенсора С при которой в режиме воспроизведения сенсор С имеет два входа и два выхода, причем входом может служить как выход Сумматора В, так и выход Сумматора В1, а выходами - шина, ведущая в сенсорам группы D нейрона ИНИ и шина, ведущая к сенсорам группы E1 нейрона ИНМ. Это позволяет избежать активации одинаковых Генераторов D и D1, а также позволяет избежать требования к микросхеме в использовании дополнительных сенсоров группы D1, что снизит сложность архитектуры и стоимость процессоров Памяти Последовательностей.

7.9.3.6. Сумматор В1.

Получив от каждого из Сенсоров группы А1 состояние частотных шин на момент Метки начала и Метки окончания, Сумматор В1 рассчитывает точное значение Метки длины (длина Трубы контекста) как абсолютное значение разницы между Меткой начала и Меткой окончания, а также запоминает, по меньшей мере, Метку начала или Метку окончания и Метку длины или все названные метки [7.7.1.1].

В режиме "воспоминания" Сумматор В1 получает от каждого из Сенсоров группы А1 значение состояния частотной шины, соответствующее "Метке поиска", вычисляет "Длину поиска" как абсолютное значение разницы между "Меткой поиска" и "Меткой начала" и сравнивает полученное значение "Длины поиска" с сохраненным значением "Длины трубы". Если "Длина трубы" сравнима (определение 7) с "Длиной поиска", то функция активации Сумматора В1 активирует выход Сумматора В1 и посылает сигнал в сенсор С, который активирует группу сенсоров D Генератора Трубы. Понятно, что вместо Метки начала при вычислении Длины поиска можно использовать и Метку окончания, что приведет к смещению "совпадения" на Длину Трубы в будущее.

Сумматор В1 получает с каждого из сенсоров А1 значение, представленное нулем или единицей, и рассчитывает точное значение метки в единицах шин наименьшего разряда, а для этого Сумматор В1 (формула 61 Алгоритм расчета длины):

1. суммирует значения, полученные от всех Сенсоров С, установленных на частотных шинах одинакового разряда N,
2. каждую полученную сумму умножает на произведение (N-1) множителей, каждый из которых равен числу частотных шин в смежном младшем разряде (N-2), а сумму самого младшего разряда умножает на единицу или использует саму сумму самого младшего разряда в качестве "произведения" для следующего шага,
3. суммирует полученные на предыдущем шаге произведения и полученный результат представляет собой точное значение метки в единицах самого младшего разряда.

Пример.

Предположим, в самом младшем разряде 10 шин, в следующем 20, в следующем 30 и в самом старшем 40. Предположим, что в самом старшем разряде сигналы есть в 4 шинах, в смежном младшем разряде 3, в следующем 2 и в самом младшем разряде 1 шина, что соответствует записи (слева старшие разряды и справа младшие): 1111,111,11,1

Тогда длина в единицах самого младшего разряда будет равна:

$$(1+1+1+1)*30*20*10 + (1+1+1)*20*10 + (1+1)*10 + 1 = 24\ 000 + 600 + 20 + 1 = 24\ 621$$

7.9.3.7. Запоминание меток в Трубе.

Для запоминания множества измерительных шин с ненулевыми значениями сигнала (далее "Метка измерений"), на котором был обучен ИНМ, измерительные шины в ИПП оснащены сенсорами группы E1, а Сенсор С оснащен Четвертым соединением, которое связывают с названными сенсорами группы E1; в режиме обучения сигнал обучения поступает на Первое или Второе соединение Сенсора С, который передает сигнал обучения на Четвертое соединение и группу Сенсоров E1, каждый из которых запоминает вес совместной встречаемости с соответствующей шиной слоя измерения, если в названной шине измерений есть сигнал метки измерений; в режиме воспроизведения после срабатывания функции активации ИНИ или после срабатывания функции активации ИНМ, Сенсор С получает от одного из Сумматоров сигнал активации и передает через Четвертое соединение сигнал активации на группу Сенсоров E1 и, если вес совместной встречаемости соответствующего Сенсора E1 больше нуля, сигнал акти-

вации передается через Сенсор E1 в названную шину в качестве сигнала воспоминания значения метки измерений, на которой обучен ИНМ.

При необходимости метка начала или метка окончания, или обе могут передаваться на выход матрицы измерений и запоминаться в качестве значений частотных объектов в Кластере Калибра Трубы.

7.9.3.8. Заключение.

Благодаря сравнению "Длины трубы" с "Длиной поиска", будут "воспроизведены" (активированы или "вспомнены") все Трубы контекста, длина которых сравнима (определение 7) с Длиной конкретных труб контекста.

Таким образом, последовательное появление Труб объектов разной природы (например, по каналу зрения и по каналу слуха) будет сопровождаться появлением связи через ИНВ расположенный в пересечении шин этих Труб в слое Труб M2, причем каждая из Труб будет иметь последовательные метки измерений, например метки времени следующие одна за другой или метки местоположения, последовательно расположенные по маршруту следования.

Очевидно, что матрицу частотных шин M1 можно приспособить и для ввода длины события, для этого следует или продублировать частотные шины средствами ввода как метки начала так и метки длины или для ввода меток начала и меток длины следует использовать разные сигналы, хотя бы одна из характеристик которых позволяет сенсорам A1 определить какого рода сигнал они получили - сигнал метки начала или сигнал метки длины. Ввод в матрицу частотных шин M1 метки начала и метки длины позволило бы Памяти Последовательности вспоминать события за определенное время, привязанные к некоторой точке начала событий. Например, ответить на вопрос "Что происходило вчера?", так как в вопросе есть продолжительность - "день" или "сутки", а также метка начала - "вчера" или начало вчерашнего дня или начало вчерашних суток.

Синхронизация измерений (сравнение длины) событий позволяет выявить одновременные и параллельные последовательности и события [2.1.5]. Например, образ кошки, полученный по каналам зрения, будет синхронизирован со звуками мяуканья, полученными по каналам слуха, поскольку они синхронны во времени. Аналогичным способом эти образы могут быть синхронизированы и в пространстве, которое представляет из себя еще один слой измерений.

7.9.3.9. Альтернативные решения.

Альтернативной может быть архитектура синхронизации измерений, в которой частотные шины отсутствуют, а в Трубы синхронизации записывается полное число циклов работы генератора частоты. Однако такая модель будет иметь тот недостаток, что в отсутствии уникального набора шин "метки события", поиск Генератора Трубы контекста станет невозможным без использования цифровых фильтров, которые понадобятся для поиска Трубы, содержащей запись нужного числа циклов генератора частоты.

7.9.4. Другие особенности измерений.

Перечислим другие особенности реализации Слоя синхронизации:

1. Выбор точки отсчета.
2. Выбор направления относительно точки отсчета + или -.
3. Синхронизация измерений с разными точками отсчета.

7.9.4.1. Точка отсчета.

Отсчет времени жизни человека начинается в момент его рождения, однако это вовсе не значит, что раньше мир не существовал. Поэтому выбор точки отсчета является важным и для разных реализаций может отличаться. Более того, если для шкалы температур мы можем придерживаться мнения о существовании абсолютного нуля -273 градуса Кельвина, то в отношении времени с уверенностью выбрать точку отсчета трудно, хотя бы потому, что достоверно неизвестен момент возникновения вселенной, а также неизвестно существовало ли время до ее возникновения.

7.9.4.2. Положительная и отрицательная шкала.

Возвращаясь к температуре, следует учесть, что в системах измерения Си и Фаренгейт точка отсчета отличается от абсолютного нуля и потому возникает необходимость наличия как положительной, так и отрицательной шкалы измерения, что может приводить к увеличению числа частотных шин, необходимых для отсчета и синхронизации в обоих направлениях.

7.9.4.3. Синхронизация измерений.

Когда мы пытаемся дотронуться рукой до движущегося предмета, то для оценки места встречи руки и предмета (для оценки точки пересечения траекторий), наш мозг использует модель с двумя точками отсчета в пространстве - местоположение предмета и местоположение руки в момент начала сближения. Таким образом, в большинстве прикладных задач робототехники может возникать необходимость использования множества точек отсчета, а значит Слой синхронизации должен содержать достаточное число частотных шин и быть расширяемым.

Однако выбора точек отсчета недостаточно, для синхронизации необходимо, чтобы Память Последовательностей могла делать оценки и сравнивать их значения.

7.9.5. Синхронизация времени

Представленная архитектура измерений позволяет представить любой момент времени в виде метки события - набора шин синхронизации времени (далее также "метка времени").

Время является примером смешанной системы исчисления, так как в минуте 60 с, в часе 60 мин, в сутках 24 ч, в году до 365 дней и число лет может быть ограничено. Архитектура слоя синхронизации измерений позволяет реализовать такую сложную модель, разместив следующие слои:

60 шин с частотой переключения 1 Гц

60 шин с частотой переключения $\frac{1}{60}$ Гц

24 шины с частотой переключения $\frac{1}{(60*24)}$ Гц

365 шин с частотой переключения $\frac{1}{(60*24*365)}$ Гц

Любое количество шин с частотой переключения 1 раз в год

В качестве примера применения Алгоритма расчета длины (формула 61) посчитаем полную длину времени трех лет, что соответствует следующему числу частотных шин 111,0,0,0,0 и длина года в единицах самого младшего разряда (секунда) будет равна:

$$(1+1+1)*365*24*60*60 + 0*365 + 0*24 + 0*60 + 0*60 = 94\ 608\ 000 \text{ секунд}$$

Понятно, что для учета более коротких промежутков времени перед шинами с частотой 10 Гц можно расположить другие с более высокой частотой, а для учета десятилетий, столетий, тысячелетий и так далее, можно разместить шины с частотой переключения раз в 10, 100, 1000 лет и так далее вверх и вниз по архитектуре шин синхронизации.

Как было многократно показано ранее, Труба ("Труба контекста") соответствует контексту последовательности и, если в Генератор Трубы записать идентификаторы всех активных шин Слоя синхронизации (далее набор активных шин измерения времени, станем звать "метка времени"), которые были включены в момент записи Генератора Трубы, то в Кластере Трубы контекста можно будет найти идентификаторы шин "метки времени" и посчитать по ним "абсолютное время" записи Трубы, отсчитанное с начала включения Слоя синхронизации, если начало включения Слоя синхронизации было выбрано в качестве точки начала отсчета времени.

Вместе с тем, желательно сравнивать фрагменты последовательностей сходной продолжительности, а для этого необходимо рассчитать длительность ввода сравниваемых фрагментов, привязанных ко времени начала их ввода. Таким образом, необходимо привязываться не к "абсолютному времени" начала работы Слоя синхронизации, а к "относительному времени" начала ввода последовательностей или их фрагментов в Память Последовательности. Для этого надо знать время записи Генератора предыдущей Трубы, которую можно найти в памяти последовательностей используя обратно-упреждающую связь между последовательными Трубами [5.2.1].

Таким образом, если в матрице разместить описанный блок синхронизации Памяти Последовательностей, то запоминание в Генераторе Трубы идентификаторов шин "метки времени" Слоя синхронизации позволяет привязать Трубу контекста к абсолютному и относительному времени Слоя синхронизации Памяти Последовательностей, а наличие в Трубе контекста идентификатора Трубы синхронизации, позволяет найти Генератор Трубы контекста не только по Кластеру Трубы, но и по "метке времени", которой соответствует время создания этого Кластера.

В рассмотренном примере переключение шин Слоя синхронизации позволяет измерять время с необходимой точностью. Архитектура позволяет начать отчет времени с момента создания вселенной или отсчитывать время в прошлое от исходного времени системы. Можно учесть также недели, месяцы и любые другие периоды.

Современное производство радиоэлектронных компонентов с нормой 10 нанометров позволяет разместить 1000 шин на кремниевой подложке шириной всего 10 микрометров.

7.9.6. Синхронизация пространства.

При синхронизации пространства выбор точки отсчета является особенно важным и поэтому нужны частотные шины как для шкалы с положительными, так и для шкалы с отрицательными значениями. Люди воспринимают себя самих как точку отсчета расстояния до предметов и видимо для роботов точкой отсчета также будут являться они сами или местоположение их датчиков, например видеокамер или других устройств, наблюдающих окружающий мир в видимых или невидимых глазу/уху/... излучениях или проявлениях. В качестве абсолютной точки отсчета для робота можно также выбрать точку его первого включения или точку, где был произведен робот, или точку, где робот работает и так далее. Для людей такой точкой может являться, например, малая родина.

Однако для полноты модели нужен учет направлений. В качестве модели можно принять геодезическую модель с двумя координатами - широтой и долготой или с двумя положениями - слева или справа (однако люди фактически используют угловые величины - "сзади" имея в виду угол 180 градусов или "сбоку" имея в виду угол 90 градусов и так далее), а также с возвышением/снижением в отношении точки отсчета - "горизонта". Таким образом, три слоя синхронизации пространства могут оказаться достаточно точными для роботов-андроидов.

7.10. Слой эмоций и этических норм.

Слой эмоций и этических норм создается в виде ограниченного числа шин "Слоя эмоций" Памяти

Последовательностей. Каждая шина представляет собой дискретное значение определенной эмоции по шкале "плохо - хорошо". Этические нормы могут быть также представлены объектами Памяти Последовательностей, а объекты шинами косынки и также соответствующими шкале "плохо-хорошо". Соответствующие значения эмоций и этических норм, а также последовательности эмоций и норм присваиваются событиям в процессе обучения Памяти Последовательностей путем активации шины соответствующего дискретного значения эмоции/этической нормы в момент ввода соответствующего события. Если бы объекты эмоций были объектами Памяти Последовательности объектов, то имели бы веса встречаемости с другими уникальными объектами Памяти Последовательностей объектов и могли быть найдены в последовательностях в качестве объектов последовательностей. Однако природа эмоций не может одновременно совпадать с природой объектов разной природы, в частности, информация которую человек получает по каналам слуха, осязания и зрения является информацией разной природы, а если бы мы имели чувство магнитного поля? Вероятно поэтому следует сделать вывод о том, что, по меньшей мере, эмоции не являются объектами Памяти Последовательностей (слоя) объектов.

По меньшей мере одна из эмоций или этических норм в ИПП кодируется лишь одной из названных групп измерительных шин, причем каждая шина названной группы измерений (далее "шина эмоций") кодирует определенное дискретное значение названной одной из эмоций или этических норм.

Шины эмоций в ИПП соединены "каждый с каждым" и в пересечениях каждой пары шин эмоций установлен (ИНВ).

В некоторых исполнениях ИПП более одной из названных групп измерительных шин используются в качестве группы измерительных шин определенной эмоции или этической нормы, причем дискретные значения различной разрядности для названной эмоции или этической нормы кодируются шинами группы соответствующей разрядности.

7.10.1. Рефлекторные эмоции.

Рефлекторные эмоции служат защитой нашему организму. Обжегшись о горячую сковороду, мы отдергиваем руку в сторону противоположную сковороде. Ударившись головой о трубу, мы отклоняем голову в сторону противоположную трубе. Если в нас летит камень, то мы прогнозируем (представим последовательность полета до попадания в нас) куда он может попасть и отпрянем в безопасную (противоположную) сторону. Таким образом, можно предположить, что рефлексы имеют в общем реверсивное действие, выводящее пострадавшую часть тела (или которая может пострадать) в сторону противоположную исходящей опасности. В первом приближении это можно представить как отматывание назад последовательности, которая привела к опасной ситуации.

Позитивные рефлекторные эмоции приводят скорее к повтору последовательности сначала. Например, если мы голодны, то едим, но едим мы порциями, каждая из которых сопоставима с вместимостью рта. Так что за первой порцией следует вторая, за ней третья и так далее, пока чувство сытости не остановит этот процесс поедания следующей порции.

Общим в представлении негативных и позитивных эмоций является то, что система стремится вернуться в состояние с максимальным положительным или минимальным отрицательным значением эмоции, то есть система пытается максимизировать значение эмоционального состояния. Таким образом, если негативные эмоции представить негативной шкалой значений - чем сильнее негативная эмоция, тем ниже ее отрицательное значение и выше абсолютное, а позитивные положительной - чем сильнее позитивная эмоция, тем выше ее положительное и абсолютное значение, то в обоих случаях система стремится увеличить значение эмоции путем возврата к более высокому значению эмоционального состояния системы.

Точка эмоционального равновесия системы находится между отрицательными и положительными значениями и значит ее можно считать точкой начала отсчета - нулем и называть "точкой комфорта". Для устойчивости состояния комфорта каждое позитивное эмоциональное состояние должно уравновешиваться противным ему негативным состоянием. Это означает, что возникновение позитивной эмоции с последующим отклонением от точки комфорта в конце концов должно приводить к нарастанию негативной эмоции, которая противоположна названной позитивной. Например, если мы голодны, то негативная эмоция - голод вынуждает искать пищу и когда мы начинаем есть, сначала мы компенсируем негативную эмоцию голода, а когда чувство голода компенсировано, то по мере еды возникает и начинает расти отрицательная эмоция перенасыщения, что в конечном итоге приводит к отказу от еды и запуску позитивной эмоции переваривания пищи. И так далее цикл смены эмоций повторяется. Поэтому при работе над системой важно составить уравновешенную карту эмоций и циклов их смены, чтобы не допустить саморазрушения системы.

7.10.1.1. Негативные рефлекторные эмоции.

Итак, в качестве рабочей модели рефлекторных реакций системы на острую негативную эмоцию можно предложить следующую: последовательность событий/объектов приводит к усилению негативной эмоции и в зависимости от уровня негативности эмоции и скорости ее усиления, достигается некая "точка возврата", которая служит триггером для реверса последовательности, и последовательность событий отматывается назад, то есть воспроизводится в обратном порядке до "точки комфорта".

Таким образом, нам нужно определить момент, когда система вышла из "точки комфорта" и мо-

мент, когда система достигла "точки возврата". В терминах Слоя синхронизации [7.7] нам следует определить длину между названными точками. Следует заметить, что "Точка возврата" также служит триггером паузы [2.2.8] и прерывает текущую последовательность. Видимо шкала эмоций может содержать синтетический объект или заканчиваться синтетическим объектом с условным названием "Точка возврата", при появлении которого ввод последовательности прерывается, создается Труба и последовательность отматывается назад на длину такой Трубы до "точки комфорта".

7.10.1.2. Позитивные эмоции.

В качестве рабочей модели рефлекторных реакций системы на позитивную эмоцию можно предложить следующую: последовательность событий/объектов приводит к позитивной эмоции, которую система пытается продлить или усилить. В частности, если эмоция пропадает с окончанием последовательности событий (Трубы), то система повторяет такую последовательность событий сначала, а если по мере ввода последовательности растет противоположная отрицательная эмоция, то достижение паритета положительной и негативной эмоции (достижение "точки комфорта") порождает паузу [2.2.8], ввод последовательности прекращается и образуется Труба. Если по мере ввода последовательности положительная эмоция растет, то система продолжает ввод последовательности, пока не возникнет компенсирующая отрицательная эмоция и не будет достигнута "точка комфорта".

Таким образом, необходимо определить "точку комфорта", с которой последовательность следует начать снова, а также "точку возврата", которая служит триггером для повтора последовательности с "точки комфорта". "Точка возврата" также может служить триггером паузы [2.2.8]. Видимо шкала эмоций может содержать синтетический объект или заканчиваться синтетическим объектом с условным названием "Повтор Последовательности", при появлении которого последовательность будет отматываться назад на полную длину Трубы такой последовательности до "точки комфорта" или на X% длины Трубы и с этой точки воспроизводиться снова. Повтор воспроизведения будет продолжаться до тех пор, пока не будет достигнута "точка насыщения". Точкой насыщения может служить и достижение негативной эмоции с ее "точкой Реверса" или точка физической усталости или истощения. Хорошим примером недостижимости "точки насыщения" может служить нерест горбуши, когда рыба гибнет по пути к месту нереста или сразу после нереста. То есть "точка возврата" находится за рубежом возможности возврата и по сути является "точкой невозврата".

7.10.2. Приобретенные эмоции.

Если при вводе текстов или просмотре фильмов, определение их эмоциональной окраски можно автоматизировать, то при наблюдении явлений природы такая оценка должна вводиться в Память Последовательностей наряду с информацией о явлении, поскольку природа такой оценки является субъективной и социальной и даже детей приходится этому обучать. Во многих случаях абстрактные правила поведения продиктованы неочевидными последствиями для социума или отдельных людей, которые и человек может недооценивать, а другие абстрактные правила являются вариацией на тему "не делай другим того, что самому бы не понравилось" и таким эмоциям роботы сами тоже научиться не могут, если только природа их "эмоций" не будет человеческой. Поэтому обучать машины правильно оценивать события кажется необходимым. Вероятно, можно создать учебные примеры ситуаций для роботов, для их быстрого обучения в области эмоциональной и этической оценки различных событий, с которыми роботы могут столкнуться в жизни и автоматически настраивать "психику" робота в момент его производства.

Однако необходимо, чтобы наличие эмоциональной оценки приводило к блокированию или стимулированию определенных гипотез с целью формирования множества допустимых и множества предпочтительных гипотез. Например, если один из прогнозов действий робота связан с ухудшением эмоциональной окраски событий или даже к недопустимой эмоциональной окраске (например заканчивается гибелью человека), то робот должен исключить такой прогноз из числа допустимых, а если прогноз заканчивается улучшением эмоциональной окраски событий, то такой прогноз следует относить к множеству допустимых или даже к множеству предпочтительных гипотез, в зависимости от степени изменения эмоциональной окраски.

Результатом активаций шин в Слое эмоций во время ввода последовательностей (обучения ИП) будет то, что идентификаторы эмоциональных шин появятся в Трубах контекста и при построении прогнозов робот станет извлекать Генераторы Труб и выбирать из них те, которые стимулируют, а не блокируют действия робота. Таким образом, матрица Памяти Последовательностей впервые позволяет реализовать искусственный интеллект способный к этической и эмоциональной оценке событий.

7.10.3. Шкала эмоций как шкала синхронизации измерений.

Если каждое чувство представить в виде шкалы или одного из разрядов шкалы измерений, а градацию чувства "хорошо-плохо" в виде значений такого разряда или шкалы, то архитектура слоя эмоций может быть аналогичной архитектуре слоя синхронизации измерений [7.7.1]. Это позволит извлекать из памяти последовательности, соответствующие определенному набору эмоций с округлением или последовательности с определенной "амплитудой" изменения эмоций.

7.11. Архитектура Многопоточной Синхронной Иерархической Памяти Последовательностей (МСИПП).

ИПП может быть оснащена с образованием Многопоточной Иерархической Памяти Последова-

тельность (МИПП) двумя или более различными УИДМ, множеством нейронов ИНИ и множеством нейронов ИНМ, которые соединены названным множеством Сенсоров С; причем множество названных Сенсоров С соответствующего нейрона ИНИ представлено подмножествами, каждое из которых связывает соответствующий ИНИ посредством названных ИНМ с названными различными Устройствами измерения длины; а множество названных Сенсоров С соответствующего нейрона ИНМ связывает соответствующий ИНМ с группами Сенсоров D различных нейронов ИНИ; в режиме обучения соответствующего ИНИ также обучаются один или более названных ИНМ, каждый из которых связывает названный ИНИ с различными УИДМ; в режиме воспроизведения после срабатывания функции активации Сумматора В одного из нейронов ИНИ или после срабатывания функции активации Сумматора В1 одного из нейронов ИНМ такой нейрон передает сигнал активации соответственно на Второе или Первое соединение Сенсора С, а Сенсор С передает сигнал активации через свое Третье и Четвертое соединение на группу Сенсоров группы D соответствующего нейрона ИНИ и на группу сенсоров E1 соответствующего нейрона ИНМ.

МИПП оснащена с образованием Многопоточной Синхронной Иерархической Памяти Последовательностей (МСИПП) множеством ИПП объектов разной природы (ИППРП), каждая из которых оснащена множеством слоев измерений разной природы, причем, по меньшей мере, пара ИППРП используют по меньшей мере один слой измерений одинаковой природы, а для синхронизации меток измерений такого по меньшей мере одного слоя измерений одинаковой природы в таких по меньшей мере двух ИППРП, названные слои измерений одинаковой природы оснащают одним генератором сигналов измерений или оснащают разными генераторами сигналов измерений одинаковой природы, в каждый из которых вводят одинаковые координаты точки начала отсчета измерений.

МСИПП оснащена множеством ИПП объектов разной природы (ИППРП), причем все названные ИППРП используют общий слой синтетических объектов иерархии M2 так, что последовательности синтетических объектов названного слоя состоят из синтетических объектов, каждый из которых порожден слоем иерархии M1 одной из названных ИППРП.

Для обеспечения многопоточности измерений в Иерархической Памяти Последовательностей, ИПП снабжают множеством слоев измерений разной природы, например, слоем измерения времени, местоположения, эмоций, этических норм и любых других измерений. Множество сенсоров С связанные с нейронами ИНИ разбивают на подмножества и каждое подмножество Сенсоров С связывают с ИНМ различных слоев измерений. Таким образом, каждый нейрон ИНИ оказывается связанным с разными слоями измерений. В режиме обучения ИНИ одновременно с ИНИ в каждом слое измерений обучается нейрон ИНМ, шина обучения которого активна в момент обучения ИНИ. Это позволяет связать каждый нейрон ИНИ через Сенсоры С с нейронами ИНМ слоев измерений разной природы. Например, один и тот же ИНИ может быть связан с нейроном ИНМ слоя измерения времени, с нейроном ИНМ слоя измерения местоположения и с нейроном ИНМ слоя измерения эмоций и этических норм.

Для целей синхронизации МСИПП предпочтительной архитектурой является объединение между собой множества Иерархических Памятей Последовательностей (ИПП), каждая из которых является ИПП для объектов разной природы. Например, одна ИПП используется для визуальных образов, другая для звуковых, третья для текстовой информации и так далее. По меньшей мере, самый нижний слой каждой из ИПП - слой памяти последовательностей объектов, должен хранить только последовательности объектов названной уникальной природы, а поскольку слой измерений размещается на уровне слоя объектов, то каждая из ИПП должна иметь собственные слои измерений для каждого из измерений.

Принципами объединения названных ИПП в единую Многопоточную ИПП являются следующие:

1. Слои измерений одинаковой природы для ИПП объектов разной природы должны быть синхронизированы путем использования единой точки отсчета генераторов G (фиг. 78) слоев измерения одинаковой природы. ИПП объектов разной природы также могут быть синхронизированы путем использования одного и того же генератора G для слоев измерений одинаковой природы. Например, для всех слоев измерения времени генератор времени должен иметь или одинаковую точку отсчета или это должен быть один и тот же генератор времени для всех слоев измерения времени для каждой из ИПП объектов разной природы. Так же и слои геодезических измерений должны иметь или единую точку отсчета координат или единый Генератор координат.

2. Все ИПП объектов разной природы должны быть объединены на уровне одного из слоев Труб, например на уровне Труб 1-го рода или на уровне Труб 2-го рода и так далее. Это становится возможным потому, что Трубы являются синтетическими объектами смысла и как объекты теряют свою исходную природу, что позволяет объединить синтетические объекты на любом из уровней Труб.

Использование выделенных Генераторов для слоев измерения одной природы в ИПП объектов разной природы разумно использовать в ситуации, например, когда местоположение различных ИПП отличается. Примером такой ситуации может являться человечество как множество людей, имеющих единую общность как человечество - общую историю, общую планету и так далее. Так каждый из людей обладает собственной ИПП, однако знания и достижения всего человечества, представленная архивами и базами данных, является примером МИПП общей для всего человечества в целом и такая МИПП синхронизирована во времени и в пространстве путем привязки к единой шкале времени (в наших терминах это

Генератор времени), а также такая МИПП синхронизирована в пространстве с помощью геодезической привязки или привязки к географии Земли, сначала путем присваивания и использования географических названий, а затем путем присваивания и использования геодезических координат.

Использование отдельных слоев измерений для каждой ИПП позволяет создавать собственные метки измерений для каждой Памяти Последовательностей объектов уникальной природы, причем такие метки каждой из Памятей Последовательностей объектов любой другой природы оказываются синхронизированными между собой из-за использования или единой точки отсчета или единого Генератора. Так, например, звуковой ряд мяуканья кошки и визуальный ряд появления кошки будут синхронизированы во времени и в пространстве, что позволяет одновременно извлечь обе последовательности из ИПП визуальных образов и из ИПП звуковых образов, введя в обе ИПП в качестве поискового запроса метку времени появления кошки или метку местоположения появления кошки.

7.12. Сравнение ИПП с корой мозга.

7.12.1. Внутренний зернистый слой.

Сначала отметим следующие факты:

1. Из экспериментов известен нейрон Билла Клинтона, что говорит о "назначении" мозгом нейронов, отвечающих за распознавание уникальных объектов. Таким образом можно утверждать, что каждый из нейронов может кодировать определенный уникальный объект.

2. Известно также, что внутренний зернистый слой коры мозга состоит из звездчатых нейронов, у которых нет единственного ярко выраженного "выхода" - аксона и это делает звездчатые нейроны похожими на объекты Памяти Последовательностей, входы которых формируются статистически как результат "сочетаемости" с другими объектами - нейронами.

Перечисленные выше факты позволяют сравнить рассмотренную нами модель косынок полносвязной Памяти Последовательностей со внутренним зернистым слоем коры мозга. Как видно, ввод ключевых объектов последовательности в память последовательностей порождает Кластеры, состоящие из множества частотных объектов. Аналогично возбуждение первичных нейронов внутреннего зернистого слоя (вроде нейрона "Билла Клинтона") коры порождает волны возбуждения во вторичных нейронах внутреннего зернистого слоя коры. Можно предположить, что, так называемые, "карты признаков", подаваемые на вход нейросетей, являются отдаленными аналогами Кластеров Памяти Последовательностей или аналогами множества вторичных нейронов коры мозга, возбужденных первичным нейроном (например нейроном "Билла Клинтона"). Это означает, что Память Последовательностей может служить источником "карты признаков" для различных нейросетей, в том числе для сверточных нейросетей. При этом в качестве карты признаков в нейросеть можно вводить или множество весов частотных объектов Кластера, который получен на выходе косынки, или множество весов прямой и/или обратной встречаемости части или всех объектов косынки. В последнем случае нейросеть можно научить на опасных отклонениях от нормального "Состояния сознания" Памяти Последовательностей с тем, чтобы нейросеть отсеживала опасные отклонения от нормального "Состояния сознания" Памяти Последовательностей.

7.12.2. Пирамидальные нейроны - синтетические объекты.

Сравнивая то, что известно о функционировании пирамидальных нейронов (см. фиг. 79 и 80) с функционированием синтетических объектов памяти последовательностей, легко заметить сходство функционирования.

Это позволяет предположить, что пирамидальные нейроны могут быть предназначены для кодирования синтетических объектов, то есть объектов, созданных разумом для соединения смежных слоев иерархии Памяти Последовательностей, как это было показано для синтетических объектов памяти последовательностей. Исходя из такого представления о пирамидальных нейронах несложно предположить, что слои 3a и 3b коры мозга представляют собой пирамидальные нейроны, которые соединяют разные уровни иерархии Памяти Последовательностей, а именно соединяют внутренний и внешний зернистый слои коры мозга, причем внешний зернистый слой представляет собой Память Последовательностей Труб или слой M2 верхнего уровня иерархии, в то время как внутренний зернистый слой представляет собой слой объектов или нижний слой иерархии M1 коры мозга. Обращает на себя внимание то, что исследователи выделяют не один, а два слоя пирамидальных нейронов, расположенных между двумя зернистыми слоями. Это также совпадает с предложенной моделью Памяти Последовательностей, в которой есть два типа "пирамидальных" нейронов - ИНИ и ИНМ. Джефф Хокинг [["Hierarchical Temporal Memory", Jeff Hawkins & Dileep George, Numenta Corp]] предполагал, что связанные между собой разные участки коры мозга являются разными уровнями иерархии и Хокинг назвал это Hierarchical Temporal Memory - Временной Иерархической Памятью. Вероятно, второй слой пирамидальных нейронов (в нашей модели это нейроны ИНМ) связывает один участок коры мозга иерархии M1 с другим участком коры мозга иерархии M2. Как следует из настоящей работы, любой участок коры мозга является иерархической памятью, содержащей по меньшей мере два слоя ИПП - внутренний и внешний зернистые слои, связанные между собой пирамидальными нейронами. Поэтому, учитывая гипотезу Джеффа Хокинга, можно предположить, что сендвич из двух слоев M1 и M2 внутренней иерархии каждого из участков мозга, связан вторым слоем пирамидальных нейронов с иерархией разных участков коры мозга, образуя многослойную иерархию, состоящую из слоев каждого из участков коры и также из иерархии участков коры

между собой. В такой памяти исходящие сигналы одного участка коры являются входящими сигналами другого участка коры мозга. Таким же образом можно связывать между собой слои обработки памяти последовательностей и именно поэтому число слоев Труб может быть достаточно большим.

7.12.1. Недостатки нейросетей.

Несмотря на то что функционально перцептрон моделирует работу пирамидальных нейронов - много входом и один выход, при определении весов входящих связей перцептрона современные нейросети полагаются на математические методы, в частности метод обратного распространения ошибки и другие абстрактные методики, которые напрямую не связаны с механизмами памяти последовательностей и не основаны на ее работе. В то время как Память Последовательностей обучают, вводя последовательности, то есть строя в Памяти Последовательностей статистическую модель внешнего мира, методики обучения нейросетей способны обучить нейросеть лишь решению узко специализированных задач.

7.13. Преимущества аппаратной реализации Памяти Последовательностей.

Как было показано ранее, программная реализация Памяти Последовательностей с использованием рекурсивного индекса поисковых машин, требует хранения в индексе множества хитов для каждого из уникальных объектов последовательностей, из-за чего задача порождения Кластера уникального объекта имеет высокую трудоемкость. Задача порождения Кластера уникального объекта с помощью нейронных сетей решается методами обучения нейронной сети (метод обратного распространения ошибки), которые не позволяют прямо связать значения весовых коэффициентов искусственных нейронов, со статистикой появления уникальных объектов, а это не позволяет достоверно утверждать, что поведение нейронной сети будет полностью определяться картиной мира, полученной нейронной сетью в процессе обучения, а ее решения будут предсказуемы. Последнее обстоятельство ограничивает использование нейронных сетей в задачах, где решения, принимаемые нейронной сетью, связаны с безопасностью людей. Еще одним недостатком нейронных сетей является тот факт, что искусственные нейроны по сути не повторяют функции нейронов кортекса в части кодирования конкретных уникальных объектов реальности, в то время как нейроны кортекса кодируют, что доказано на так называемых "нейронах Билла Клинтона". Как известно, моделью работы искусственных нейронов является модель пирамидальных нейронов - много входов и один выход с порогом срабатывания. В настоящей работе показано, что пирамидальные нейроны кодируют синтетические объекты. Таким образом, искусственные нейронные сети реализуют функционал слоя пирамидальных нейронов, пренебрегая функциями зернистых слоев нейронов.

Переход к аппаратной реализации Памяти Последовательностей в виде матрицы, позволяет порождать Кластер уникального объекта за один цикл работы матрицы: подавая на вход матрицы сигнал соответствующего уникального объекта, на выходе матрицы получаем множество сигналов Кластера названного уникального объекта.

Использование аппаратной реализации ПП в виде матрицы также позволяет автоматизировать задачу производства выводов через порождение синтетических объектов с их упреждающими и обратными связями к существующим уникальным объектам. Синтетические объекты и названные связи порождаются в процессе обучения и использования аппаратной реализации ПП.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ создания и функционирования Памяти Последовательностей, при котором цифровая информация представлена множеством машиночитаемых массивов данных, каждый из которых является последовательностью множества уникальных Объектов, а каждый из названных Объектов представлен уникальным машиночитаемым значением Объекта, причем каждый Объект (далее "ключевой Объект") появляется, по меньшей мере, в некоторых последовательностях, характеризующийся тем, что Память Последовательностей (далее "ПП") обучают, подавая последовательности Объектов на вход памяти, а память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, предшествующие названному ключевому Объекту в названной последовательности (далее "частотные Объекты прошлого"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Прошлого", а также память при каждом появлении ключевого Объекта извлекает из названной последовательности объекты, следующие за названным ключевым Объектом в названной последовательности (далее "частотные Объекты будущего"), увеличивает на единицу значение счетчика совместной встречаемости ключевого Объекта с каждым уникальным частотным Объектом и обновляет значение счетчика новым значением, а множество значений счетчика для разных уникальных частотных Объектов объединяет в массив весовых коэффициентов взаимной встречаемости ключевого Объекта с уникальными частотными Объектами массива данных "Будущего"; множество объектов каждого из производных массивов данных "Прошлого" и "Будущее" разбивают на подмножества (далее "ранговые множества"), каждое из которых содержит только частотные Объекты, равноотстоящие от названного ключевого Объекта или в "Прошлом", или в "Будущем", а каждому уникальному ключевому Объекту ставят

во взаимное соответствие и сохраняют в ПП сам названный ключевой Объект и по меньшей мере одно из названных ранговых множеств названного уникального ключевого Объекта, содержащее, по меньшей мере, значение счетчика взаимной встречаемости названного уникального ключевого Объекта с каждым уникальным частотным Объектом, а также делают доступным поиск названного рангового множества весовых коэффициентов по введенному названному уникальному ключевому Объекту или поиск названного уникального ключевого Объекта по названному ранговому множеству или его части.

2. Способ по п.1, отличающийся тем, что для каждого уникального ключевого Объекта в памяти сохраняют по меньшей мере одно ранговое множество или "будущего", или "прошлого" ранг которого одинаков для всех уникальных ключевых Объектов (далее "базовый ранг" множества), причем каждый весовой коэффициент взаимной встречаемости ключевого и частотного Объектов названного рангового множества относится к частотному Объекту, который непосредственно соседствует в последовательностях с названным ключевым Объектом или разделен с названным ключевым Объектом числом частотных объектов, соответствующим базовому рангу.

3. Способ по п.2, отличающийся тем, что в памяти сохраняют некоторое множество всех ранговых множеств базового ранга в качестве эталонного (далее "Эталонное Состояние Памяти" или "ЭСП"), а любое "Мгновенное состояние памяти" (далее "МСП") или его часть сравнивают с ЭСП или его частью для выявления отклонений МСП от ЭСП.

4. Способ по п.3, отличающийся тем, что массив "будущего" или массив "прошлого", или ранговое множество ранга, отличное от множества базового ранга, представляют множеством, производным от множества МСП.

5. Способ по п.3, отличающийся тем, что при отклонении МСП от ЭСП выполняют алгоритм поиска или алгоритм прогнозирования, или алгоритм коррекции ошибки, или их комбинацию.

6. Способ по п.2, отличающийся тем, что названное ранговое множество является множеством первого ранга и содержит весовые коэффициенты частотных Объектов, непосредственно соседствующих с названным ключевым Объектом в названных последовательностях.

7. Способ по п.2, отличающийся тем, что в памяти сохраняют ограниченное число ранговых множеств.

8. Способ по п.7, отличающийся тем, что массив данных "Будущее" и массив данных "Прошлое" формируют как линейную композицию весовых коэффициентов или ранговых множеств массива данных МСП.

9. Способ по п.8, отличающийся тем, что при вводе Объекта, уникальный цифровой код которого мог быть введен с ошибкой, сравнение ранговых множеств осуществляют с целью выявления возможной ошибки.

10. Способ по п.1, отличающийся тем, что сравнивают ранговые множества разных рангов (далее "Когерентные множества") для известных ключевых Объектов последовательности, причем ранг рангового множества для каждого ключевого Объекта выбирают соответствующим числу Объектов последовательности, разделяющих названный ключевой Объект и Объект-Гипотезу (далее "Фокальный Объект когерентных множеств"), возможность появления которого проверяют.

11. Способ по п.1, отличающийся тем, что для каждого из частотных Объектов конкретного рангового или полного множества из памяти извлекают ранговое множество, для которого названный частотный Объект является ключевым Объектом, извлеченные ранговые множества одинакового ранга сравнивают с целью определения по меньшей мере одного Объекта-Гипотезы.

12. Способ по п.1, отличающийся тем, что ввод последовательностей в память осуществляют циклами, причем на каждом цикле в память вводят очередь объектов последовательности (далее "Окно Внимания"), а при переходе к следующему циклу очередь объектов увеличивают или сдвигают по меньшей мере на один объект в будущее или прошлое.

13. Способ по п.12, отличающийся тем, что во время названного цикла для каждого из объектов Окна Внимания как для ключевого Объекта из памяти извлекают по меньшей мере один названный массив или ранговое множество, содержащее весовые коэффициенты встречаемости частотных Объектов, из всех названных массивов или множеств извлекают весовые коэффициенты встречаемости каждого уникального частотного Объекта, общего одновременно для всех названных массивов или множеств, и складывают их, таким образом формируя Множество Трубы, содержащее суммарные весовые коэффициенты встречаемости каждого уникального частотного Объекта со всеми объектами Окна Внимания.

14. Способ по п.13, отличающийся тем, что из Множества Трубы извлекают и складывают весовые коэффициенты встречаемости всех частотных Объектов, получая Суммарный Вес Трубы.

15. Способ по п.14, отличающийся тем, что из Суммарного Веса Трубы очередного Множества Трубы вычитают Суммарный Вес Трубы предыдущего Множества Трубы и, если разница не превышает установленной погрешности, то результат сохраняют в качестве множества Калибра Трубы, создают идентификатор синтетического Объекта и ставят друг другу в соответствие названный идентификатор, множество Калибра Трубы и множество объектов Окна Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

16. Способ по п.15, отличающийся тем, что каждый уникальный частотных Объект, который не встречается по меньшей мере в одном из массивов или ранговых множеств объектов Окна Внимания, или удаляют из Множества Трубы или приравнивают нулю его вес, а полученное множество считают множеством Калибра Трубы, затем названному множеству Калибра Трубы ставят в соответствие существующий или вновь созданный Объект Памяти Последовательностей (далее "Синтетический Объект"), а также ставят в соответствие Окно Внимания, далее именуемое Генератором Трубы, причем в Памяти Последовательностей сохраняют поставленные друг другу в соответствие названные Синтетический Объект, множество Калибра Трубы, а также Генератор Трубы.

17. Способ по п.16, отличающийся тем, что во время названного цикла Множество Трубы сравнивают с ранее сохраненным по меньшей мере одним множеством Калибра Трубы, и если отличие Множества Трубы от множества Калибра Трубы сравнимо с некоторой погрешностью, то из Памяти Последовательностей извлекают Генератор Трубы, соответствующий названному множеству Калибра Трубы и используют названный Генератор Трубы в качестве результата поиска (далее "воспоминаний") Памяти Последовательностей в ответ на ввод Окна Внимания в качестве поискового запроса.

18. Способ по п.16, отличающийся тем, что очередность создания последовательных Калибров Трубы, содержащих частотные объекты Памяти Последовательностей текущего уровня иерархии (далее "уровень иерархии M1"), сохраняют в Памяти Последовательности как последовательность Синтетических Объектов Памяти Последовательностей более высокого уровня иерархии (далее "уровень иерархии M2").

19. Способ по п.18, отличающийся тем, что каждое текущее множество Калибра Трубы связывают с Синтетическим Объектом (далее "Частотный Синтетический Объект"), которому поставлено в соответствие множество Калибра Трубы, предшествующее текущему в последовательности Калибров Трубы, путем размещения в названном текущем множестве Калибра Трубы весового коэффициента взаимной встречаемости Синтетического Объекта (далее "Ключевой Синтетический Объект"), поставленного в соответствие текущему множеству Калибра Трубы, с названным Частотным Синтетическим Объектом.

20. Способ по п.18, отличающийся тем, что последовательность Синтетических Объектов вводят в Память Последовательностей в качестве одного из машиночитаемых массивов данных Памяти Последовательностей уровня иерархии M2, являющихся последовательностью множества уникальных Объектов.

21. Способ по п.16, отличающийся тем, что по меньшей мере один из названных массивов будущего или прошлого, или названных множеств Трубы или Калибра Трубы, или ЭСП, или МСП, или совокупность названных массивов и множеств, или любое множество, которое является производным от названных массивов и множеств, вводят в качестве исходных данных в искусственную нейронную сеть перцептронов или свёрточную нейронную сеть или другую искусственную нейронную сеть с известной архитектурой.

22. Память Последовательностей (далее "ПП"), содержащая два соединенных между собой множества N параллельных пронумерованных шин, из которых первое множество расположено над вторым множеством так, что шины первого и второго множества образуют в плане множество пересечений вида "матрица", причем концы каждого множества N параллельных шин, расположенные с одной из сторон "матрицы", используются в качестве входов, а противоположные им концы - в качестве выходов так, что сигналы, поданные на входы первого множества N параллельных пронумерованных шин, считываются как с выходов первого множества N параллельных пронумерованных шин, так и с выходов второго множества N параллельных пронумерованных шин при наличии в пересечении шин первого и второго множества Элементов Коммутации шин первого и второго множеств, характеризующаяся тем, что угол пересечения β° между шинами первого и второго множеств выбирается, исходя из функциональных и геометрических требований к устройству памяти, причем шины первого и второго множеств с одинаковыми номерами соединены друг с другом в месте их пересечения так, что множество таких соединений образует диагональ матрицы, разделяя матрицу на две симметричные треугольные полуматрицы (далее "Косынки"), по меньшей мере одна из которых (далее "Первая Косынка") используется путем соединения по меньшей мере одним Искусственным Нейроном Встречаемости (ИНВ) каждой двух шин, по меньшей мере, с несовпадающими номерами по одной из первого и второго множества в местах их пересечения так, что концы шин первого множества являются входами и концы второго множества являются выходами Косынки, а ИНВ используется в качестве названного Элемента Коммутации для накапливания, хранения и считывания веса совместной встречаемости объектов, которым соответствуют шины, соединенные названным ИНВ; каждый из названных ИНВ функционирует, по меньшей мере, как Счетчик с функцией активации и ячейкой памяти для хранения последнего значения и значения порога активации ИНВ; перед началом эксплуатации устройства последнему значению присваивается некоторое исходное значение, которое сохраняется в ячейке памяти Счетчика, в ячейке памяти также сохраняется значение порога активации ИНВ; в режиме обучения каждый раз, когда сигналы поданы одновременно на каждую из шин соединенные посредством ИНВ, названный ИНВ измеряет одну из характеристик сигнала на каждой их шин, затем сравнивает измеренные значения характеристик и, если результат сравнения соответствует значению порога активации ИНВ, ИНВ считывает последнее значение из ячейки памяти, увеличивает названное последнее значение на величину изменения встречаемости и сохраняет новое последнее зна-

чение в ячейке памяти, а в режиме воспроизведения сигнал подается по меньшей мере на одну из названных шин, соединенных посредством ИНВ, сигнал проводится через ИНВ, где из ячейки памяти извлекается последнее значение, одна из характеристик сигнала изменяется соответственно извлеченному последнему значению и названный измененный сигнал передается во вторую из названных шин, соединенных посредством ИНВ, для извлечения названного последнего значения из названной одной из характеристик сигнала и использования названного последнего значения в качестве веса совместной встречаемости объектов, которым соответствуют шины.

23. Устройство по п.22, отличающееся тем, что названный ИНВ накапливает, хранит и предоставляет для считывания значение веса совместной встречаемости для двух объектов последовательности, которые или не разделены другими объектами, а непосредственно следуют друг за другом в последовательности, образуя связь Первого Ранга, или разделены одним или более объектами в последовательности, образуя соответственно связь Второго или более высокого Ранга.

24. Устройство по п.22, отличающееся тем, что входы Косынки используются в качестве выходов, а выходы - в качестве входов.

25. Устройство по п.22, отличающееся тем, что один из одновременно поданных на каждую из шин сигналов изменяется с тем, чтобы разница в сигналах указывала направление от шины с большим номером к шине с меньшим номером или наоборот от шины с меньшим номером к шине с большим номером, а каждый ИНВ снабжен не одним, а двумя Счетчиками, один для изменения последнего значения встречаемости в направлении от шины с большим номером к шине с меньшим номером и второй для изменения последнего значения встречаемости в направлении от шины с меньшим номером к шине с большим номером.

26. Устройство по п.22, отличающееся тем, что в местах пересечения шин с совпадающими номерами также размещены названные ИНВ, каждый из которых содержит по меньшей мере один Счетчик, а шины с совпадающими номерами соединены между собой параллельной связью, которая параллельна Нейрону Встречаемости.

27. Устройство по п.26, отличающееся тем, что параллельная связь снабжена элементом, который изменяет сигнал при переходе из одной шины с совпадающими номерами в другую, чтобы задать направление считывания для Счетчика названного ИНВ.

28. Устройство по п.22, отличающееся тем, что дополнительно содержит память, в которой размещено по меньшей мере одно эталонное значение счетчика по меньшей мере для одного конкретного ИНВ, а последнее значение счетчика названного ИНВ извлекается и сравнивается с названным эталонным значением.

29. Устройство по п.28, отличающееся тем, что дополнительно содержит вычислитель для расчета названного эталонного значения, а расчет эталонного значения производится с использованием последних значений счетчиков по меньшей мере двух различных ИНВ косынки.

30. Устройство по п.28, отличающееся тем, что ИНВ оснащен средствами замены последнего значения счетчика названным эталонным значением, а замена последнего значения эталонным значением производится при поступлении в устройство соответствующей инструкции.

31. Устройство по п.23, отличающееся тем, что одна или более Косынок (далее "Секции") соединены последовательно, причем в названных ИНВ каждой из Секций накапливаются, хранятся и предоставляются для считывания только веса связей Первого Ранга, а шины с одинаковыми номерами каждых двух последовательных Секций N и (N+1) соединены так, что выходы шин Секции N служат входами шин смежной Секции (N+1), причем в режиме обучения или все Секции обучаются одновременно или обучается только одна Секция X, а последнее значение встречаемости в памяти Счетчика, расположенного в пересечении двух конкретных шин любой из секций, приравнивается последнему значению Счетчика, расположенного в пересечении тех же двух конкретных шин Секции X, а в режиме воспроизведения Секция (N+1) используются для повторного изменения сигналов, поступивших из Секции N.

32. Устройство по п.23, отличающееся тем, что одна или более Косынок (далее "Ранговые Секции") соединены последовательно, причем в названных ИНВ каждой из Секций накапливаются, хранятся и предоставляются для считывания веса связей одинакового Ранга (далее "Ранг Секции") и Ранга смежных Секций отличаются на единицу или другую величину, а шины с одинаковыми номерами каждых двух смежных Секций Ранга N и Ранга (N+1) соединены так, что выходы шин Секции Ранга N служат входами шин смежной Секции Ранга (N+1), причем в режиме обучения Нейроны Встречаемости каждой Секции обучаются на связях Ранга соответствующего Рангу Секции, а в режиме воспроизведения каждая Секция определенного Ранга используется для считывания сигналов измененных названными ИНВ Секции соответствующего Ранга.

33. Устройство по п.25, отличающееся тем, что каждый из названных ИНВ дополнительно оснащен сенсором Инверсии с функцией активации и памятью для считывания веса встречаемости объектов со Счетчиков противоположных направлений встречаемости конкретного ИНВ, а также для определения соотношения весов встречаемости противоположных направлений, причем перед обучением сенсора Инверсии функции активации названного сенсора присваивают пороговое значение, которое сохраняют в памяти, а при обучении на входы по меньшей мере двух шин названного Устройства, соединенные од-

ним из названных ИНВ с Сенсором Инверсии, подают сигналы обучения упорядоченные с использованием функции ослабления так, что измеренная разница названной одной из характеристик сигналов соответствует значению порога активации такого ИНВ, ИНВ активируется и понуждает сенсор Инверсии к считыванию значения Счетчиков в каждом из противоположных направлений встречаемости и сравнению одного значения с другим и, если отношение названных значений превышает пороговое значение, то сенсор Инверсии понуждают послать сигнал на названные по меньшей мере две шины, а полученный сигнал используют как сигнал обучения двух Искусственных Нейронов Встречаемости (далее "Сенсоры D"), установленных на пересечении каждой из названных по меньшей мере двух шин и шины устойчивого сочетания, которую таким образом обучают, причем названные Сенсоры D в качестве веса совместной встречаемости запоминают значения названной функции ослабления, которые отражают порядок следования объектов сочетания.

34. Устройство по п.33, отличающееся тем, что дополнительно оснащено Искусственными Нейронами Сочетаний (ИНС), а обучение ИНС, состоящего из Сумматора с функцией активации и памятью порогового значения функции активации Сумматора, который связан с выходами группы Сенсоров D, а также с группой сенсоров С, вход каждого из которых связан с выходом одной из шин названного множества шин уровня M1 и выход каждого из сенсоров группы С связан с Сумматором, Искусственного Нейрона Сочетаний (ИНС), производится, если сигнал обучения получен с двух названных Сенсоров D, причем полученный сигнал обучения передается в Сумматор, связанный с Сенсорами D, а Сумматор понуждается к активации группы сенсоров С, каждый из которых измеряет и запоминает значение веса взаимной встречаемости на выходе названной одной из шин из множества шин уровня M1, а также возвращает или единицу, или названное измеренное значение встречаемости, или оба названных значения Сумматору, который суммирует единицы и запоминает число сенсоров С с ненулевыми значениями встречаемости или суммирует веса и запоминает сумму весов всех сенсоров С или суммирует отдельно единицы и веса и запоминает оба названных значения в качестве порогового значения функции активации Сумматора; а в режиме "воспроизведения" каждый сенсор из группы сенсоров С измеряет значение веса и передает или единицу, или значение веса, или оба эти значения в Сумматор, а сумматор суммирует названные значения и сравнивает их с пороговым значением функции активации Сумматора и, если сумма превышает пороговое значение, то Сумматор подает сигнал воспроизведения на входы названной пары сенсоров группы D, с помощью которых извлекается из памяти сохраненные значения функции ослабления и сигнал "воспроизведения" передается на входы одной из пары шин или на обе шины названного множества шин названного Устройства.

35. Иерархическая Память Последовательностей (ИПП), характеризующаяся тем, что состоит из множества соединенных между собой устройств Памяти Последовательностей (ПП) по п.1 так, что каждая пара смежных ПП уровней иерархии N и (N+1) (далее "уровни иерархии M1 и M2") связана множеством искусственных нейронов (далее - искусственные нейроны иерархии - ИНИ).

36. Устройство по п.35, отличающееся тем, что выходы шин ПП уровня иерархии M1 используют в качестве входов для искусственной нейронной сети персептронов или свёрточной нейронной сети или другой искусственной нейронной сети с известной архитектурой, которую используют в качестве названного множества ИНИ.

37. Устройство по п.35, отличающееся тем, что ИНИ содержит Сумматор с функцией активации Сумматора, множество Сенсоров группы A, каждый из которых оснащен функцией активации и ячейкой памяти для размещения Соответствующего значения веса A и размещен на выходе одной из шин Устройства ПП уровня иерархии N, а также множество Сенсоров D, каждый из которых оснащен ячейкой памяти и устройством измерения и изменения по меньшей мере одной из характеристик сигнала и размещен на входах одной из шин ПП уровня иерархии N; причем каждый из Сенсоров группы D связан с выходом Сумматора, а каждый из Сенсоров группы A связан с входом Сумматора, кроме того выход Сумматора оснащен связью со входом одной из шин Устройства ПП верхнего уровня иерархии (N+1); режим обучения ИНИ осуществляют циклами, причем на каждом цикле упорядоченное множество из одного или более сигналов обучения (далее "Окно Внимания") подаются на входы одной или более шин ПП уровня иерархии N, причем сигналы в Окне Внимания упорядочены с использованием функции ослабления; каждый из сигналов проходит через один или более ИНВ, размещенных в уровне иерархии N ПП, и названные один или более ИНВ изменяет одну из характеристик сигнала, кодирующую вес совместной встречаемости, и на выходе каждой из множества шин ПП уровня иерархии N получают сигнал, кодирующий вес совместной встречаемости, из которого извлекается значение веса совместной встречаемости соответствующей шины, и вес передается в Сумматор, где веса, полученные с выходов разных шин, складываются и запоминается значение суммы цикла, после чего Окно Внимания изменяется и цикл обучения повторяется, причем на каждом очередном цикле обучения значение суммы очередного цикла сравнивается со значением суммы предыдущего цикла, и, если значение суммы очередного цикла обучения равно или меньше значения суммы предыдущего цикла обучения, обучение ИНИ останавливается и каждым сенсором группы A названное Соответствующее значение веса A (далее "вес активации"), полученное для цикла обучения с максимальной суммой весов, присваивается в качестве значения активации функции активации сенсора A, Сумматор присваивает функции активации Сумматора значение

максимальной суммы весов или присваивает значение числа сенсоров группы А с ненулевыми значениями Соответствующих весов А или присваивает оба названных значения, а каждый сенсор ИНИ группы D на входе каждой из шин ПП уровня иерархии N, на которые были поданы сигналы во время цикла обучения с максимальной суммой весов, измеряет и размещает в ячейке памяти Сенсора D Соответствующее значение D по меньшей мере одной из характеристик сигнала обучения, кодирующей названное значение функции ослабления D сигнала шины в Окне Внимания; в режиме воспроизведения ИНИ сигнал воспроизведения подают на одну или множество шин ПП уровня иерархии N и получают на выходе множества шин ПП уровня иерархии N веса совместной встречаемости и, если вес совместной встречаемости, полученный на выходе шины, равен или превышает значение функции активации сенсора А такой шины, сенсор А посылает в Сумматор или значение веса активации или единичное значение или оба значения, а Сумматор суммирует полученные значения функций активации сенсоров А и сравнивает полученную сумму со значением суммы активации Сумматора и, если суммарное значение равно или превышает значение функции активации Сумматора, то сигнал активации ИНИ подается на выход Сумматора, который затем одновременно поступает на вход одной из шин уровня иерархии (N+1) ПП и на входы сенсоров группы D уровня иерархии N ПП, ячейка памяти каждого из которых содержит названное Соответствующее значение функции ослабления D, причем каждый из названных сенсоров группы D изменяет сигнал Сумматора в соответствии с Соответствующим значением функции ослабления D и подает измененный сигнал на вход соответствующей шины ПП уровня иерархии N или не изменяет сигнала и подает неизменный сигнал на вход соответствующей шины ПП уровня иерархии N.

38. Устройство по п.35, отличающееся тем, что дополнительно оснащено с образованием устройства измерения длины метки (далее - УИДМ) одной или более следующими друг за другом группами измерительных шин (далее "слой измерений"), причем для каждой группы выбирается система исчисления и оснащается числом шин, соответствующим избранной системе исчисления - две шины для двоичной, три для троичной системы исчисления и так далее, и каждая из шин соединяется с источником сигнала; на шину подается сигнал, если значение равно единице и сигнал на шину не подается, если значение равно нулю; затем определяется направление увеличения разрядности групп от групп меньшей разрядности к группам большей разрядности измерений и каждой шине одной группы разрядности присваивается одинаковая мера длины так, что в смежных группах разрядности мера длины одной шины группы большей разрядности равна сумме мер длины всех шин меньшей группы разрядности; а для измерения длины выход каждой шины связывается со входом Сенсора А1, а выход каждого Сенсора А1 связывается со входом вычислителя длины метки, на первом шаге исполнения вычислитель рассчитывает "групповую длину", для этого суммируют шины с включенным сигналом в каждой группе разрядности и умножают сумму на произведение чисел, каждое из которых представляет собой число всех шин в каждой из групп меньшей разрядности или единицу в случае, если группа меньшей разрядности отсутствует, а на втором шаге вычислитель длины метки суммирует все групповые длины, а полученная сумма используется в качестве измеренной длины метки.

39. Устройство по п.38, отличающееся тем, что в УИДМ с целью генерации сигналов названным источником сигнала длина представлена последовательностью одной или более следующих друг за другом групп значений, причем каждое из значений может быть или нулем, или единицей, для каждой группы выбрана система исчисления и в группе размещено число названных значений, соответствующее системе исчисления - два разряда для двоичной, три для троичной системы исчисления и так далее, для последовательности групп определено направление увеличения разрядности групп от групп меньшей разрядности к группам большей разрядности измерений, а при увеличении измеряемой величины на единицу соответствующего разряда выбирают значение такого разряда равное нулю и устанавливают значение равным единице, а если значения такого разряда равного нулю нет, то все значения такого разряда кроме одного приравнивают нулю, а также находят в группе большей разрядности значение равное нулю и приравнивают это значение единице; при уменьшении измеряемой величины на единицу соответствующего разряда выбирают значение такого разряда равное единице и устанавливают значение равным нулю, а если значения такого разряда равного единице нет, то все значения такого разряда кроме одного приравнивают единице, а также находят в группе большей разрядности значение равное единице и приравнивают это значение нулю; определяют групповую длину, для чего суммируют значения соответствующей группы и умножают сумму на число всех значений в смежной группе меньшей разрядности или на единицу, если группа меньшей разрядности отсутствует, а затем складывают все групповые длины и сумму групповых длин используют в качестве измеренной длины метки.

40. Устройство по п.38, отличающееся тем, что по меньшей мере одна из эмоций или этических норм кодируется лишь одной из названных групп измерительных шин, причем каждая шина названной группы измерений (далее "шина эмоций") кодирует определенное дискретное значение названной одной из эмоций или этических норм.

41. Устройство по п.38, отличающееся тем, что более одной из названных групп измерительных шин используются в качестве группы измерительных шин определенной эмоции или этической нормы, причем дискретные значения различной разрядности для названной эмоции или этической нормы кодируются шинами группы соответствующей разрядности.

42. Устройство по п.40, отличающееся тем, что шины эмоций соединены "каждый с каждым" и в пересечениях каждой пары шин эмоций установлен ИНВ.

43. Устройство по п.38, отличающееся тем, что в качестве названного вычислителя длины метки используется Искусственный Нейрон Метки (ИНМ), который кроме названных сенсоров А1 оснащен множеством Сенсоров С, причем каждый Сенсор С оснащен по меньшей мере тремя соединениями, ИНМ также оснащен Сумматором В1 с функцией активации, памятью и вычислителем, а также Сумматор В1 оснащен входом и выходом; Первым соединением Сенсор С соединен с выходом Сумматора В1, а Вторым соединением Сенсор С соединен с выходом Сумматора одного из множества ИНИ названного Устройства, и Третьим соединением Сенсор С соединен с входами множества Сенсоров D названного ИНИ; вход Сумматора В1 соединен с выходами множества Сенсоров А1, вход каждого из которых соединен с одной из измерительных шин названного УИДМ; используется ИНМ в режиме обучения и в режиме воспроизведения; в режиме обучения на вход Сумматора В1 подается сигнал активации; который затем передается на выход Сумматора В1 и далее на Первые соединения множества Сенсоров С, каждый из которых переходит в режим ожидания сигнала обучения ИНИ на своем Втором соединении, и, когда на Втором соединении появляется сигнал обучения ИНИ, то Сенсор С передает сигнал обучения через Первое соединение Сенсора С в Сумматор В1 ИНМ, а сам Сенсор С понуждается к установке соединения между Первым соединением и Третьим соединением Сенсора С для использования в режиме "воспроизведения", что позволяет в режиме воспроизведения передавать сигнал с выхода Сумматора В1 на входы множества Сенсоров D; при подаче сигнала активации на вход Сумматора В1 все Сенсоры А1, соединенные со входом Сумматора В1, понуждаются к измерению наличия сигнала в измерительной шине, а затем к передаче Сумматору В1 в качестве Первого значения "ноль", если на шине сигнала нет, или значение "единица", если сигнал на шине есть, а полученные от всех Сенсоров А1 названные Первые значения используются Сумматором В1 для вычисления Первой длины метки и размещения Первой длины метки в памяти; после поступления сигнала обучения ИНИ через Сенсор С в Сумматор В1, Сумматор В1 понуждает Сенсоры А1 к повторному измерению наличия сигнала в измерительной шине, а затем к передаче Сумматору В1 в качестве Второго значения "ноль", если на шине сигнала нет, или значения "единица", если сигнал на шине есть, а полученные от всех Сенсоров А1 названные Вторые значения, Сумматор В1 использует для вычисления Второй длины метки и размещает Вторую длину метки в памяти; Сумматор В1 извлекает из памяти Первую и Вторую длину метки и вычисляет разницу Первой и Второй длин метки и сохраняют ее в памяти в качестве значения функции активации Сумматора В1; в режиме воспроизведения Сенсор А1 измеряет наличие сигнала в измерительной шине, а затем передает Сумматору В1 в качестве Третьего значения "ноль", если на шине сигнала нет, или значение "единица", если сигнал на шине есть, а после получения от всех Сенсоров А1 Третьего значения Сумматор В1 вычисляет Третью длину метки, а затем вычисляет разницу между Первой длиной метки и Третьей длиной метки и сравнивает полученный результат с названным значением функции активации Сумматора В1 с использованием алгоритма сравнения, результатом работы которого является вывод "сравнимы" или "не сравнимы", и, если Третья и Четвертая длины метки "сравнимы", то Сумматор В1 подает сигнал активации на выход и сигнал активации через Сенсор С передается в названную группу Сенсоров D нейрона ИНИ.

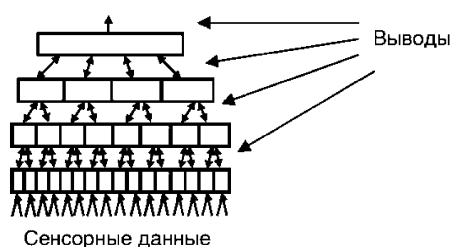
44. Устройство по п.43, отличающееся тем, что для запоминания множества измерительных шин с ненулевыми значениями сигнала (далее "Метка измерений"), на котором был обучен ИНМ, измерительные шины оснащены сенсорами группы Е1, а Сенсор С оснащен Четвертым соединением, которое связывают с названными сенсорами группы Е1; в режиме обучения сигнал обучения поступает на Первое или Второе соединение Сенсора С, который передает сигнал обучения на Четвертое соединение и группу Сенсоров Е1, каждый из которых запоминает вес совместной встречаемости с соответствующей шиной слоя измерения, если в названной шине измерений есть сигнал метки измерений; в режиме воспроизведения после срабатывания функции активации ИНИ или после срабатывания функции активации ИНМ, Сенсор С получает от одного из Сумматоров сигнал активации и передает через Четвертое соединение сигнал активации на группу Сенсоров Е1 и, если вес совместной встречаемости соответствующего Сенсора Е1 больше нуля, сигнал активации передается через Сенсор Е1 в названную шину в качестве сигнала воспоминания значения метки измерений, на которой обучен ИНМ.

45. Устройство по п.43, отличающееся тем, что оснащено с образованием Многопоточной Иерархической Памяти Последовательностей (МИПП) двумя или более различными УИДМ, множеством нейронов ИНИ и множеством нейронов ИНМ, которые соединены названным множеством Сенсоров С; причем множество названных Сенсоров С соответствующего нейрона ИНИ представлено подмножествами, каждое из которых связывает соответствующий ИНИ посредством названных ИНМ с названными различными Устройствами измерения длины; а множество названных Сенсоров С соответствующего нейрона ИНМ связывает соответствующий ИНИ с группами Сенсоров D различных нейронов ИНИ; в режиме обучения соответствующего ИНИ также обучаются один или более названных ИНМ, каждый из которых связывает названный ИНИ с различными УИДМ; в режиме воспроизведения после срабатывания функции активации Сумматора В одного из нейронов ИНИ или после срабатывания функции активации Сумматора В1 одного из нейронов ИНМ такой нейрон передает сигнал активации соответственно

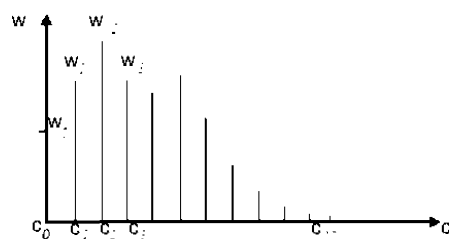
на Второе или Первое соединение Сенсора С, а Сенсор С передает сигнал активации через свое Третье и Четвертое соединение на группу Сенсоров группы D соответствующего нейрона ИНИ и на группу сенсоров E1 соответствующего нейрона ИНМ.

46. Устройство по п.45, отличающееся тем, что МИПП оснащена с образованием Многопоточной Синхронной Иерархической Памяти Последовательностей (МСИПП) множеством ИППРП объектов разной природы (ИППРП), каждая из которых оснащена множеством слоев измерений разной природы, причем, по меньшей мере, в паре ИППРП использован по меньшей мере один слой измерений одинаковой природы, а для синхронизации меток измерений такого по меньшей мере одного слоя измерений одинаковой природы в таких по меньшей мере двух ИППРП, названные слои измерений одинаковой природы оснащены одним генератором сигналов измерений или оснащены разными генераторами сигналов измерений одинаковой природы, в каждый из которых введены одинаковые координаты точки начала отсчета измерений.

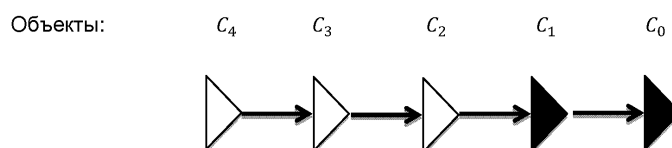
47. Устройство по п.46, отличающееся тем, что МСИПП оснащена множеством ИППРП объектов разной природы (ИППРП), причем во всех названных ИППРП использован общий слой синтетических объектов не ниже второго уровня иерархии (M2) так, что последовательности синтетических объектов названного слоя состоят из синтетических объектов, каждый из которых порожден слоем иерархии M1 одной из названных ИППРП.



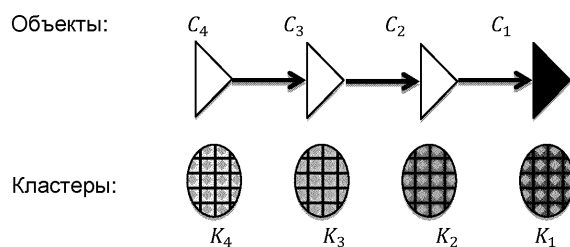
Фиг. 1



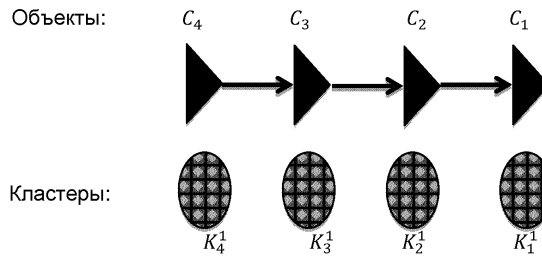
Фиг. 2



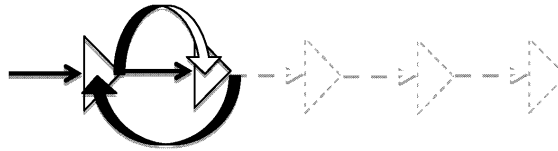
Фиг. 3



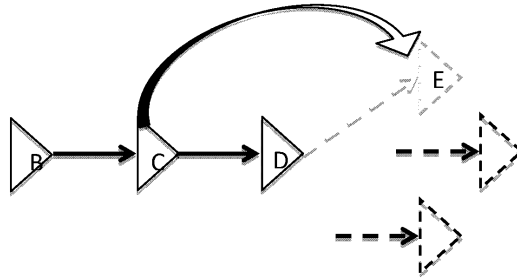
Фиг. 4



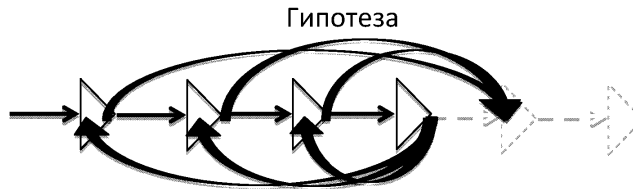
Фиг. 5



Фиг. 6

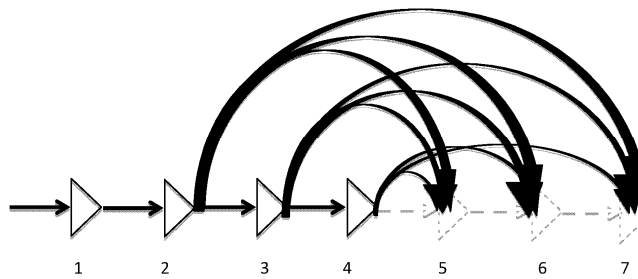


Фиг. 7

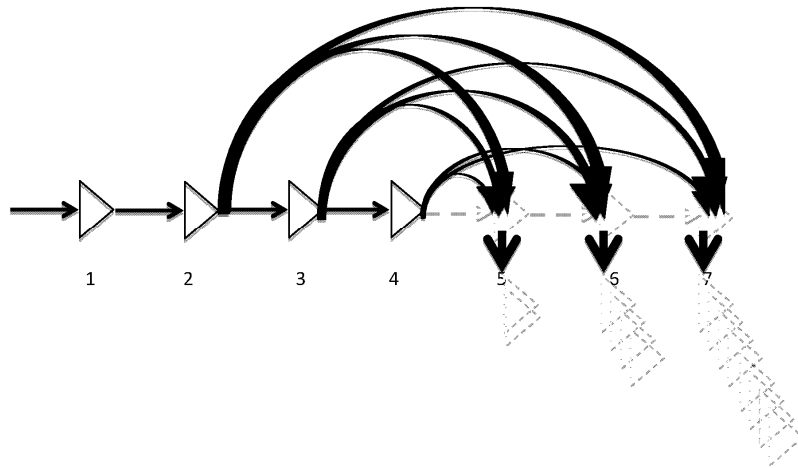


Обучение

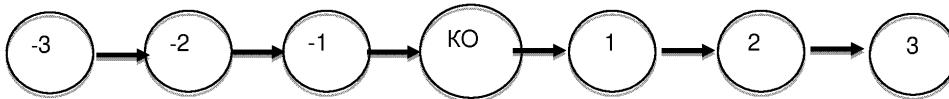
Фиг. 8



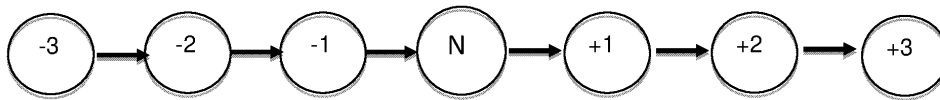
Фиг. 9



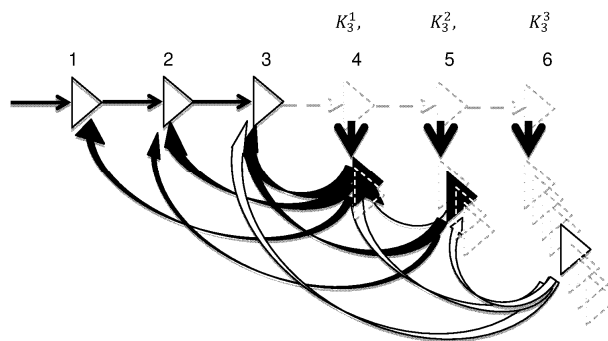
Фиг. 10



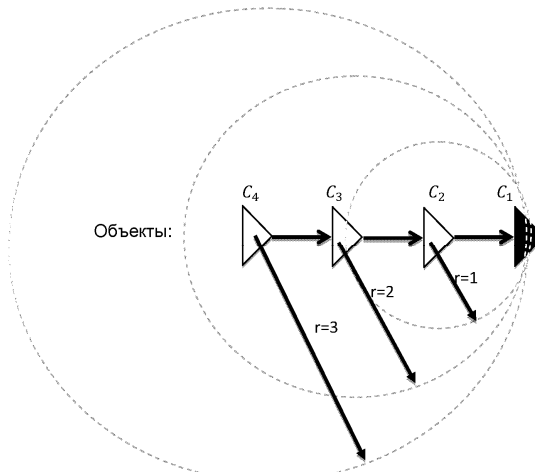
Фиг. 11



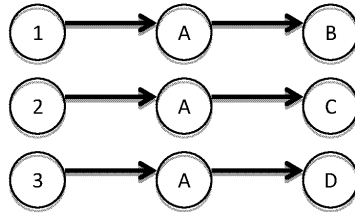
Фиг. 12



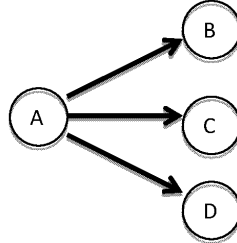
Фиг. 13



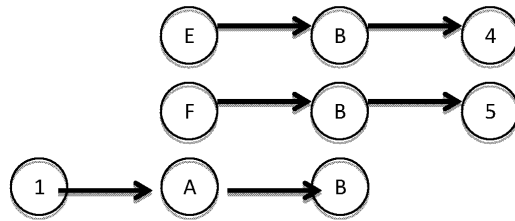
Фиг. 14



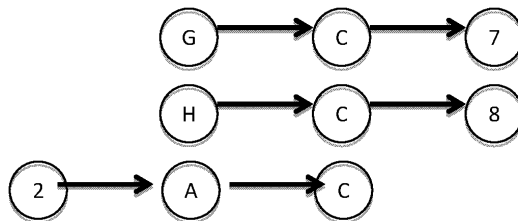
Фиг. 15



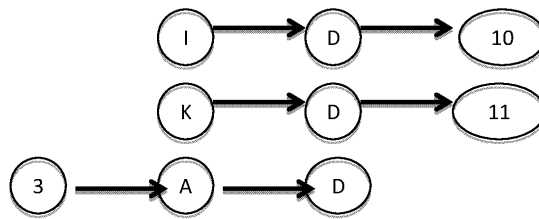
Фиг. 16



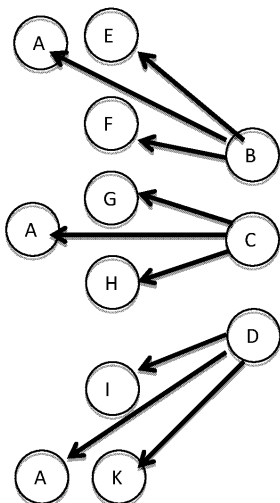
Фиг. 17



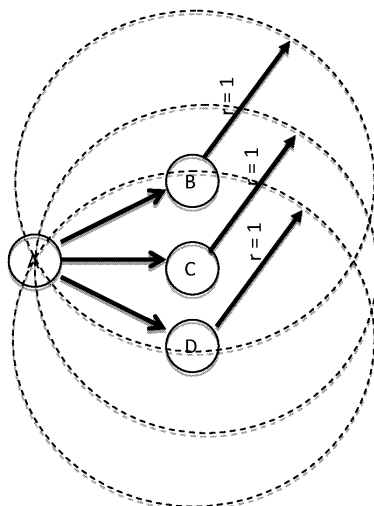
Фиг. 18



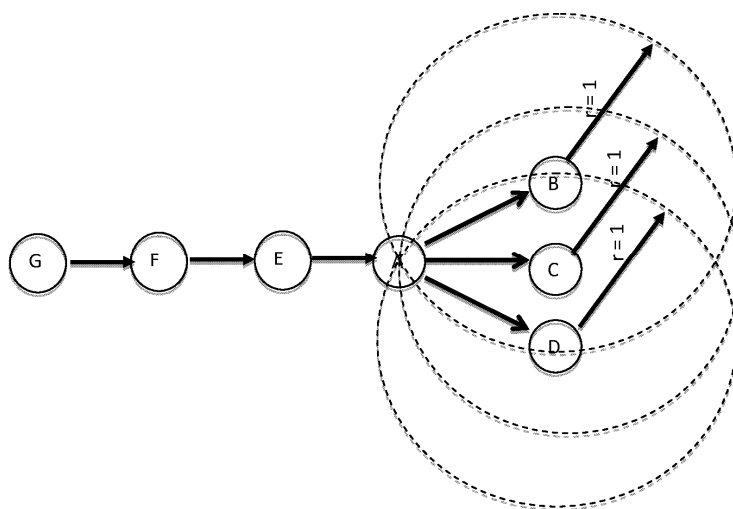
Фиг. 19



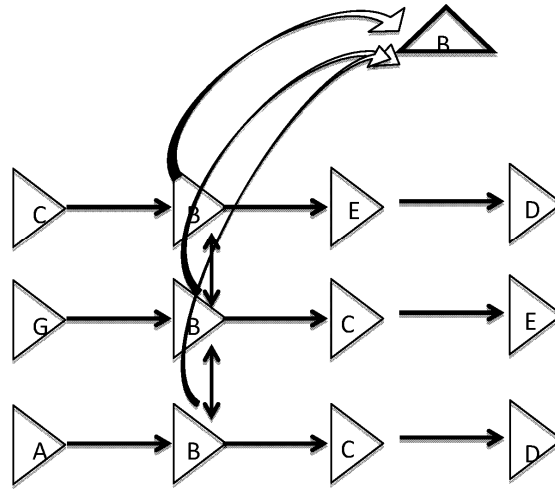
Фиг. 20



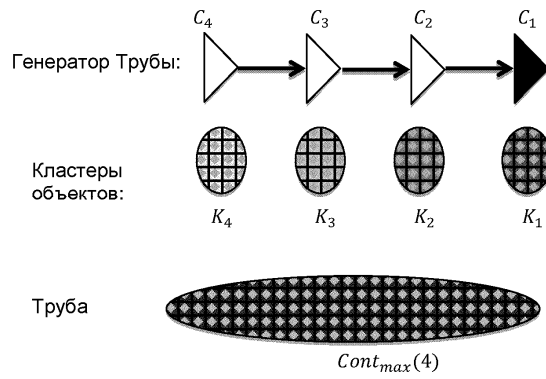
Фиг. 21



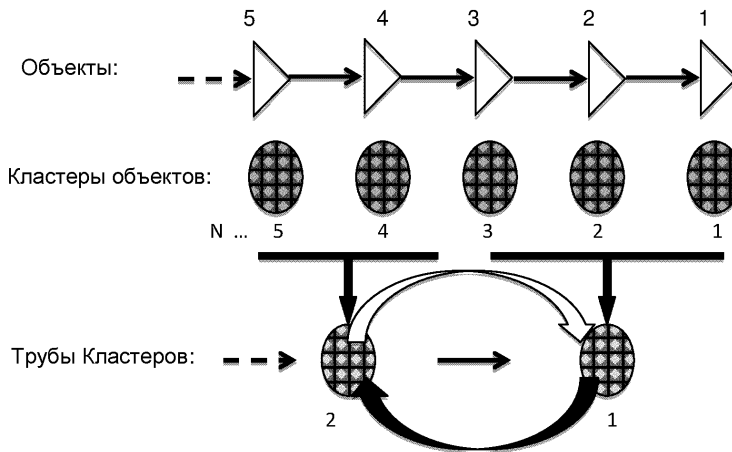
Фиг. 22



Фиг. 23

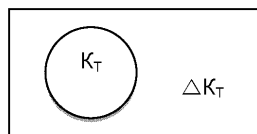


Фиг. 24

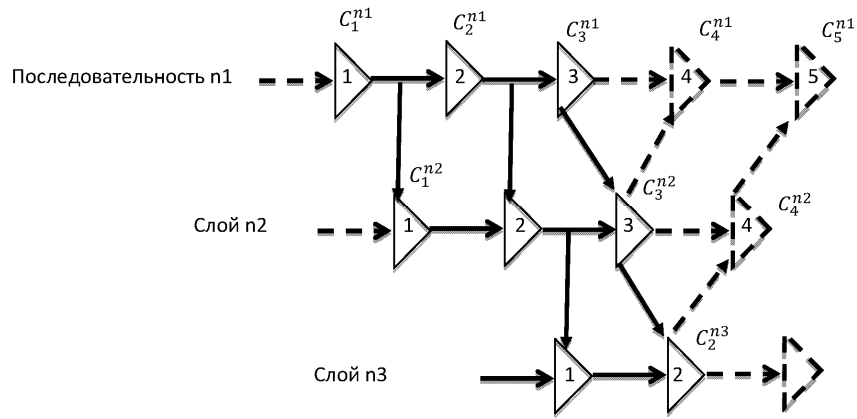


Обратные и упреждающие связи между Трубами

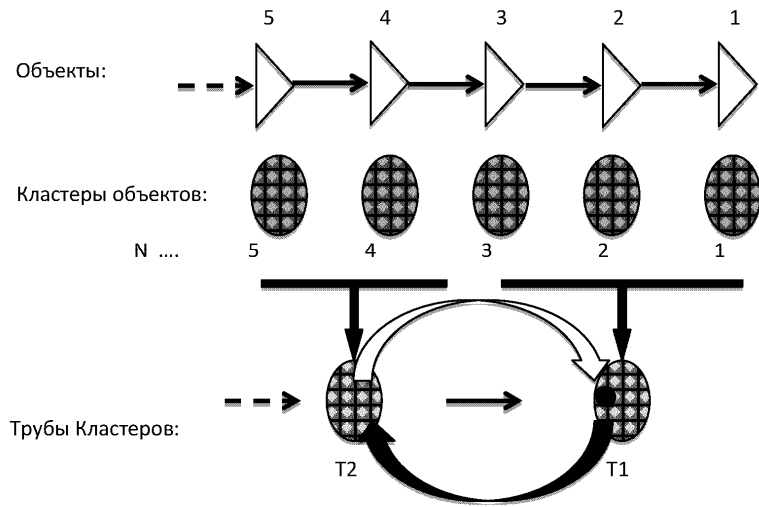
Фиг. 25



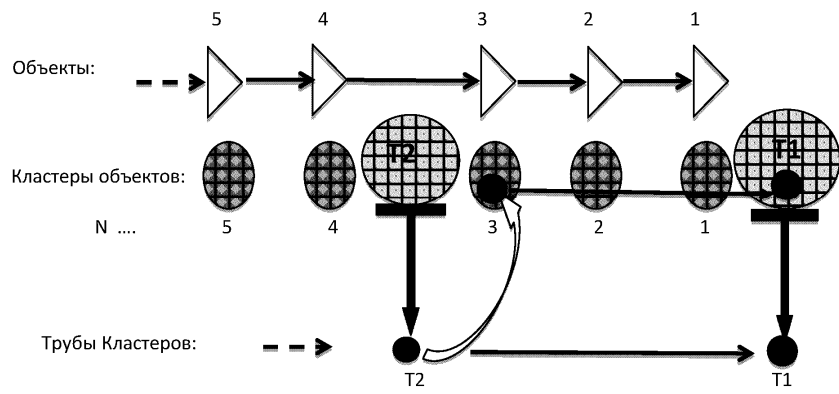
Фиг. 26



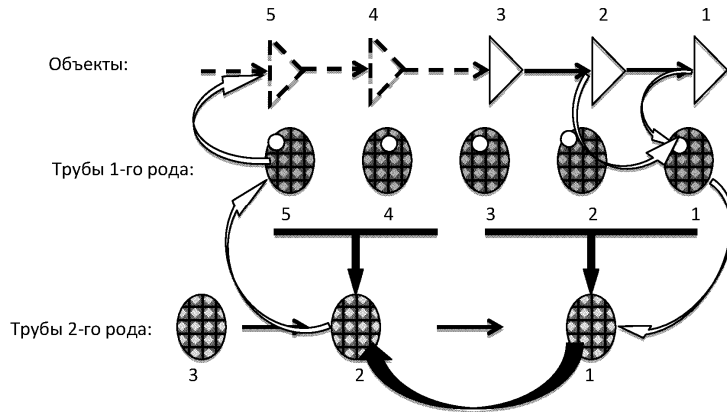
Фиг. 27



Фиг. 28



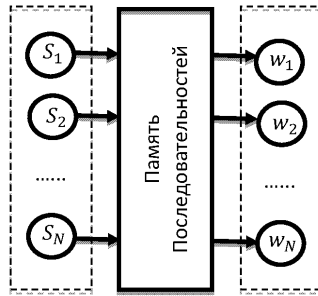
Фиг. 29



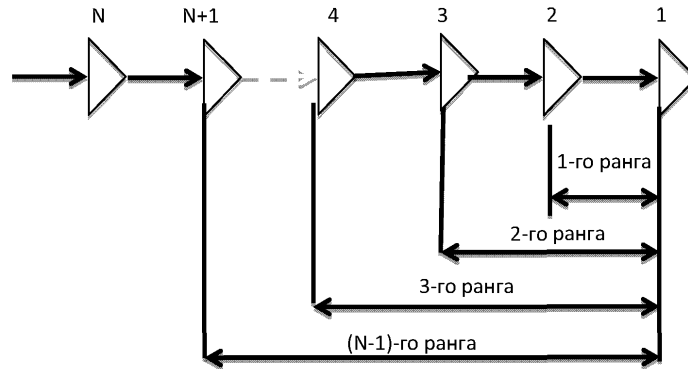
Фиг. 30



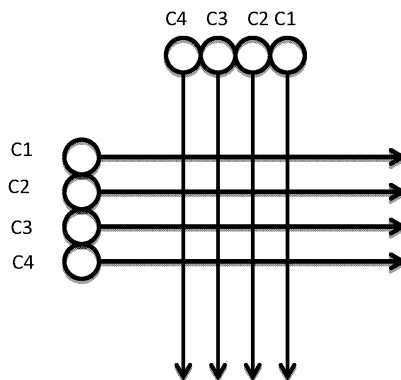
Фиг. 31



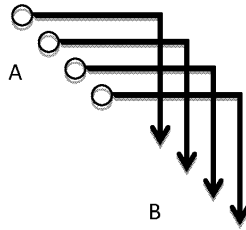
Фиг. 32



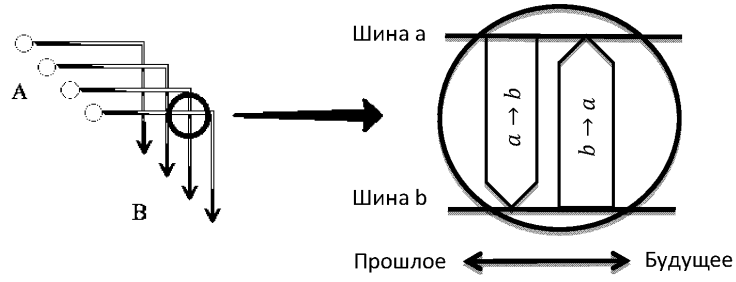
Фиг. 33



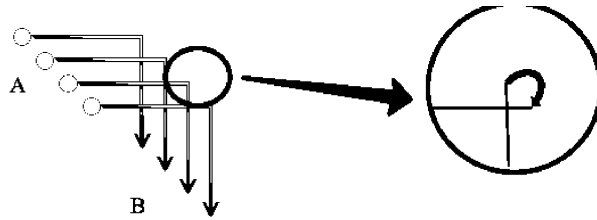
Фиг. 34



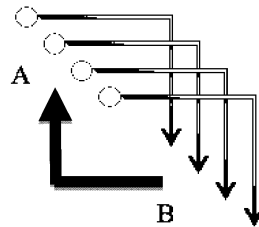
Фиг. 35



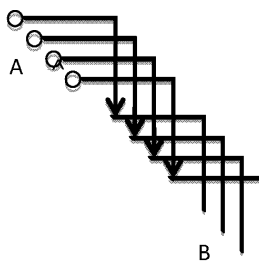
Фиг. 36



Фиг. 37



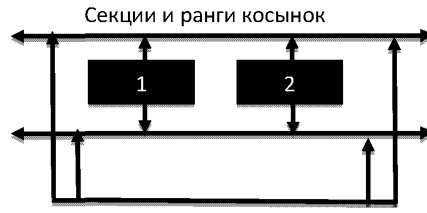
Фиг. 38



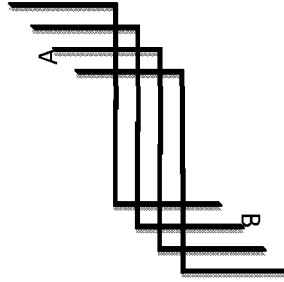
Фиг. 39



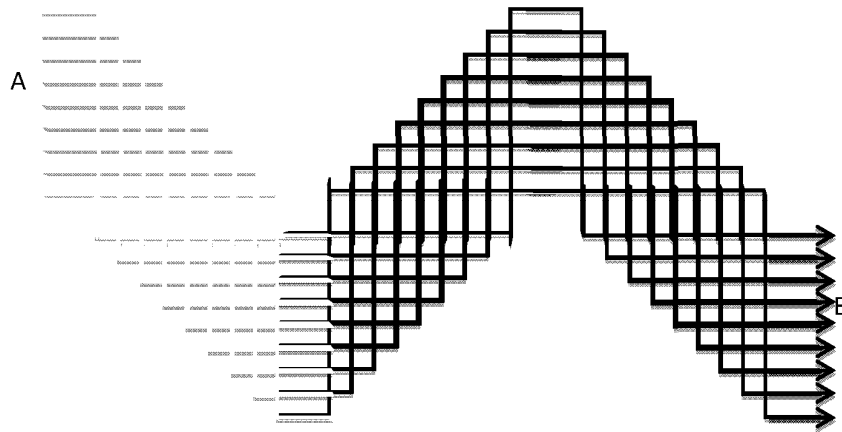
Фиг. 40



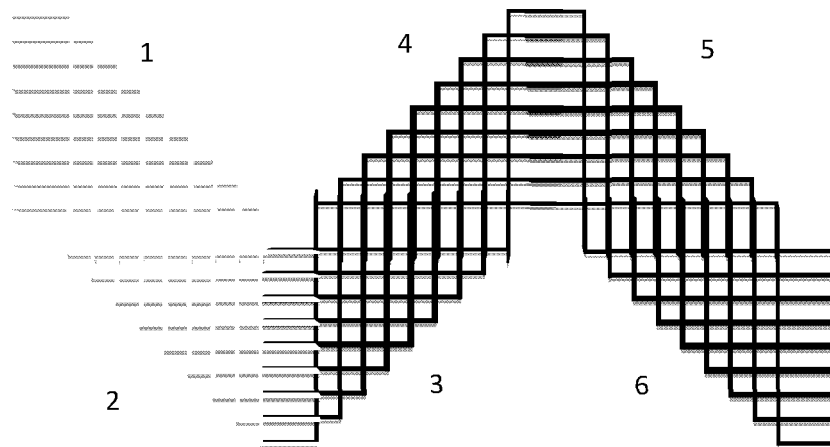
Фиг. 41



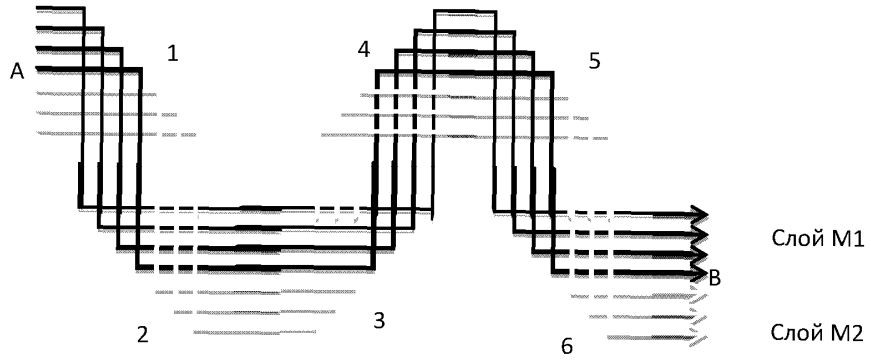
Фиг. 42



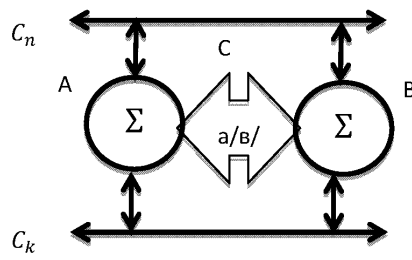
Фиг. 43



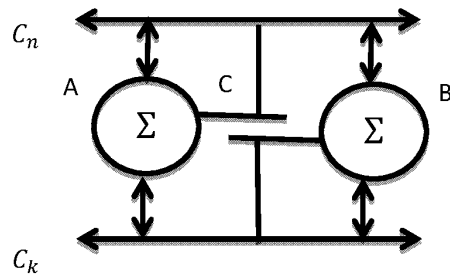
Фиг. 44



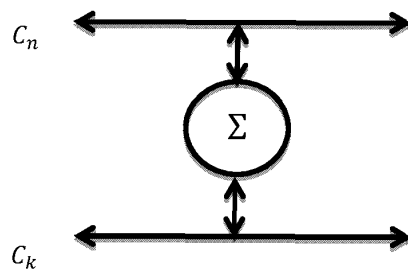
Фиг. 45



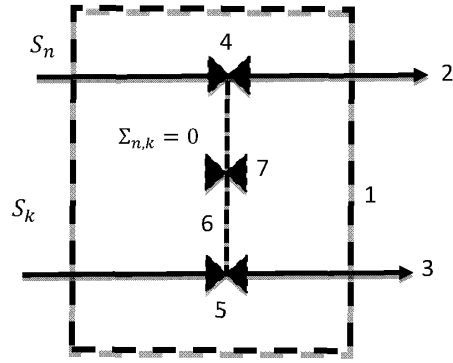
Фиг. 46



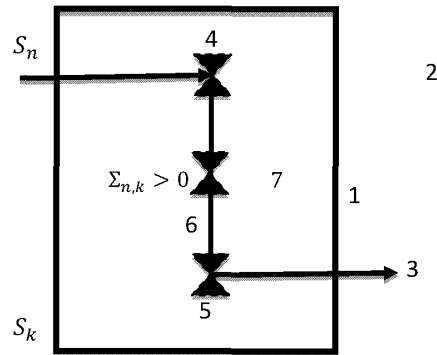
Фиг. 47



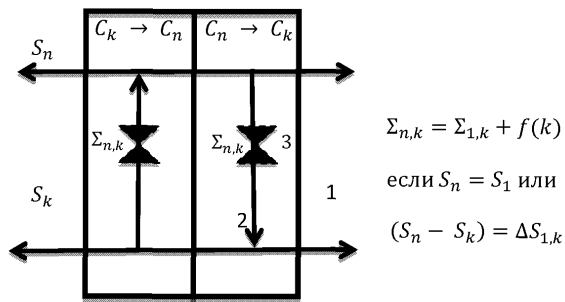
Фиг. 48



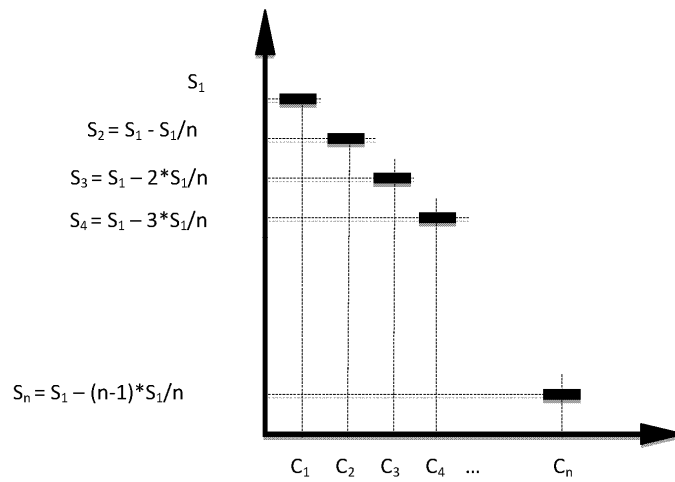
Фиг. 49



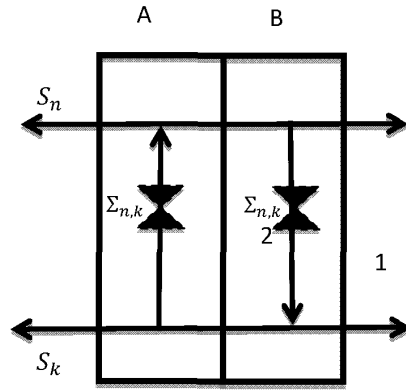
Фиг. 50



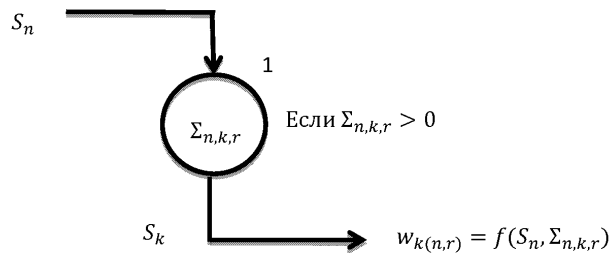
Фиг. 51



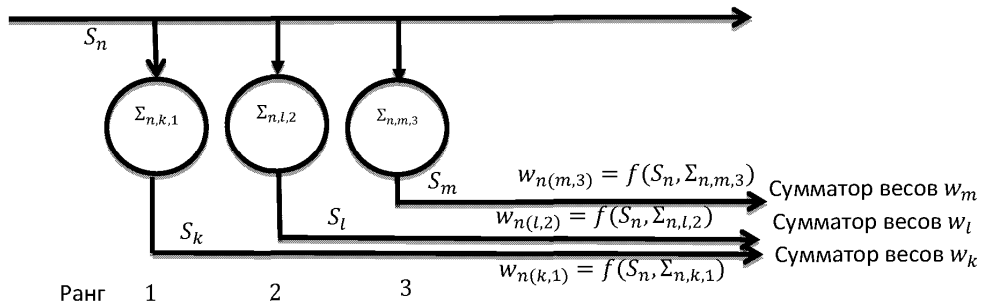
Фиг. 52



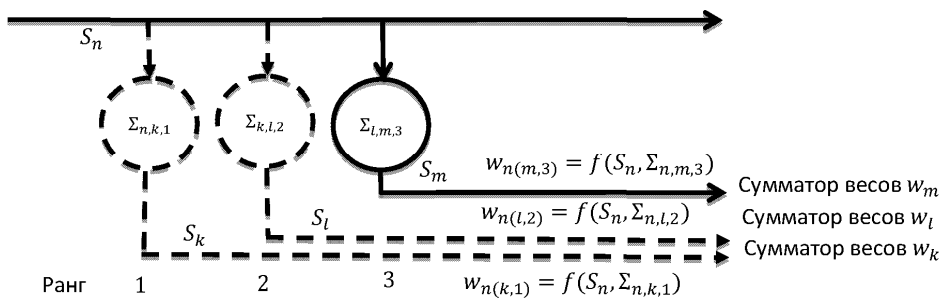
Фиг. 53



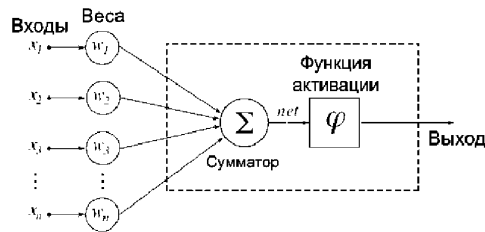
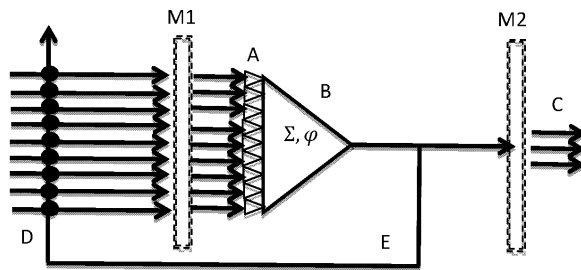
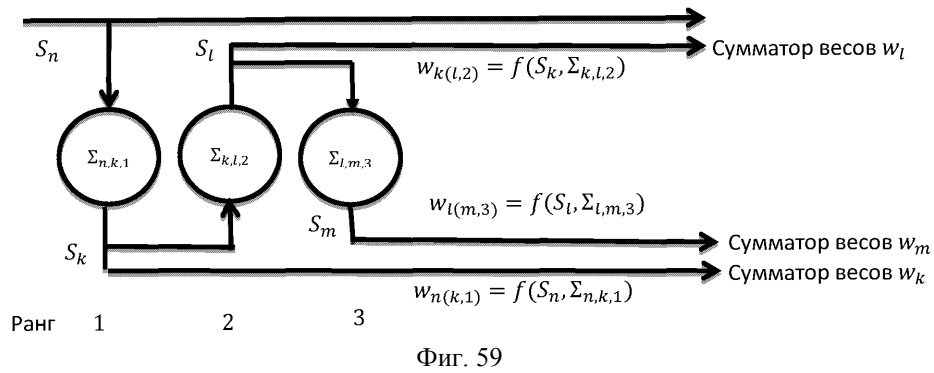
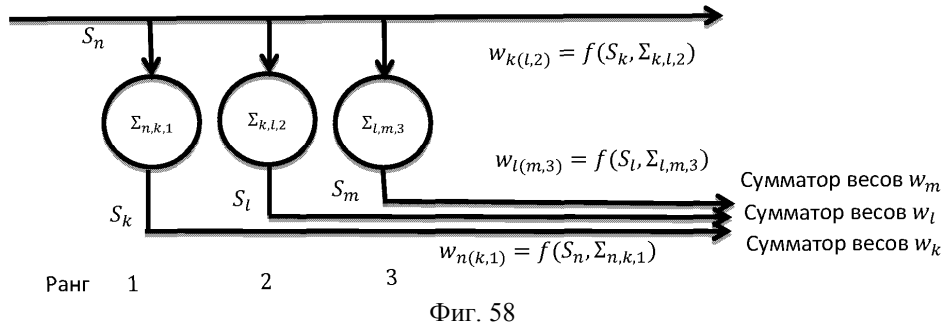
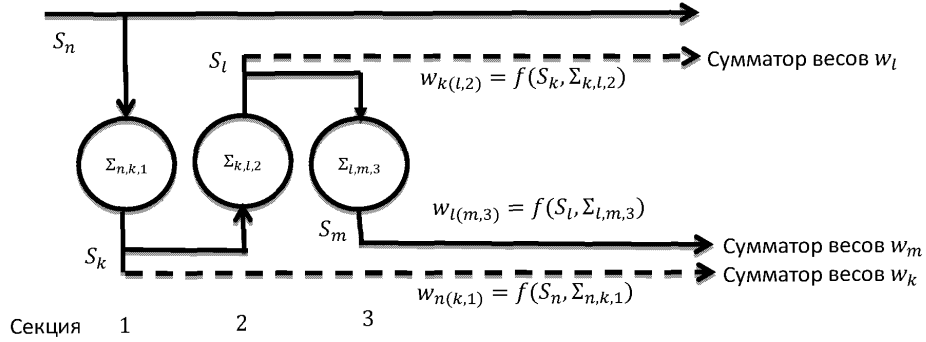
Фиг. 54

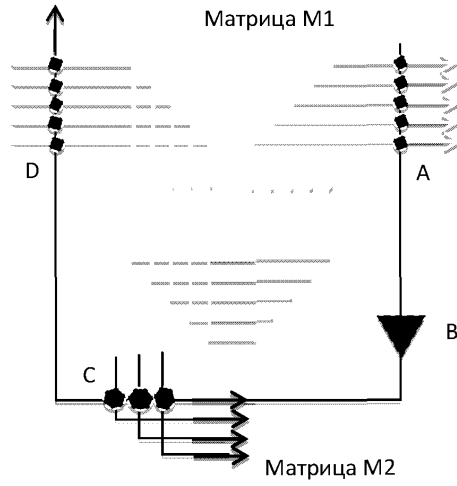


Фиг. 55

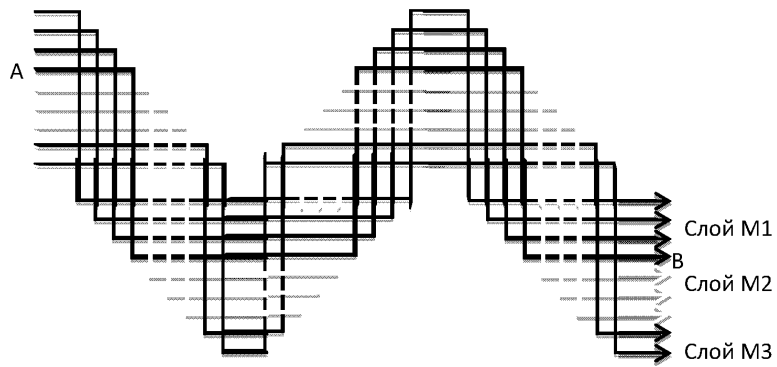


Фиг. 56

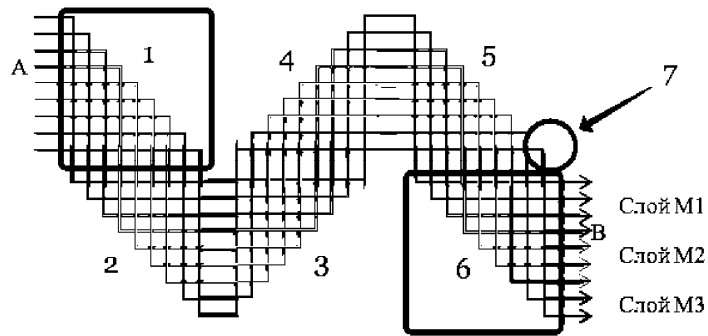




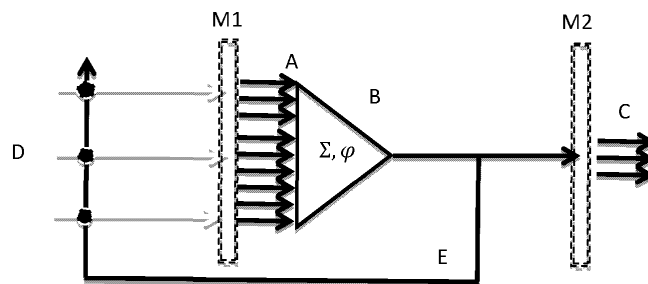
Фиг. 62



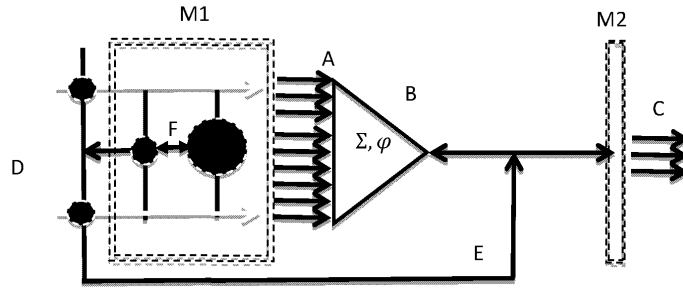
Фиг. 63



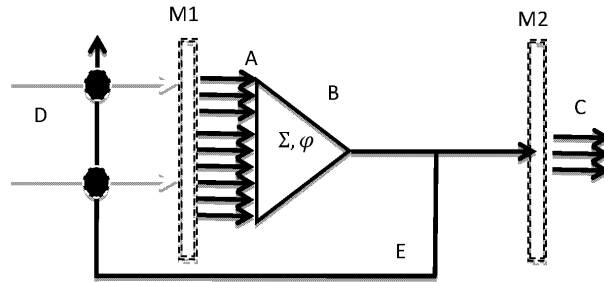
Фиг. 64



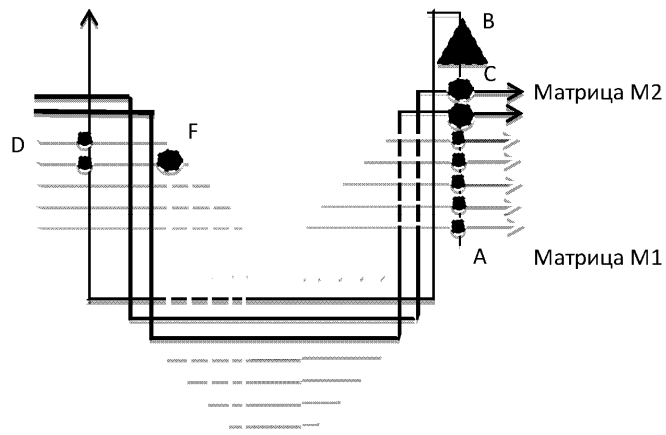
Фиг. 65



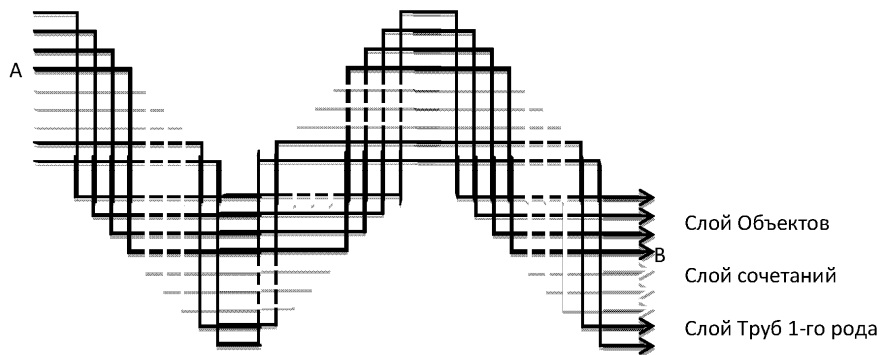
Фиг. 66



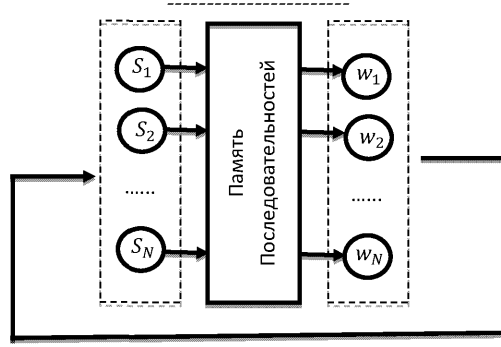
Фиг. 67



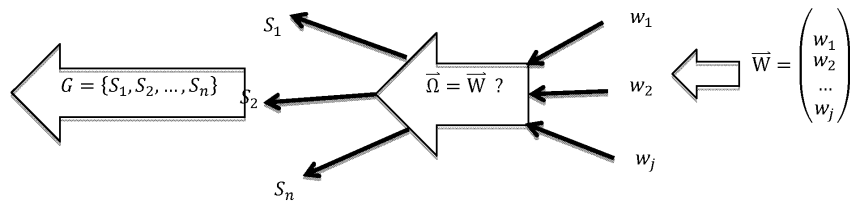
Фиг. 68



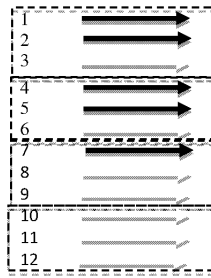
Фиг. 69



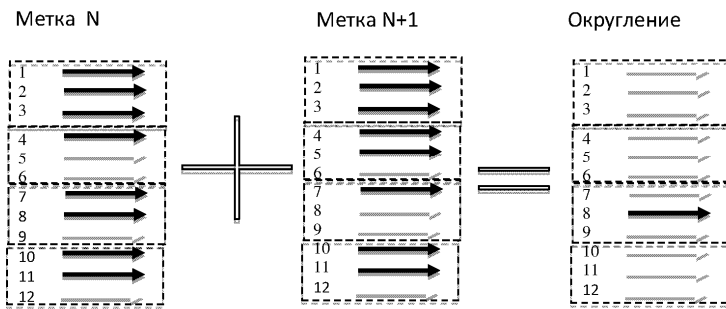
Фиг. 70



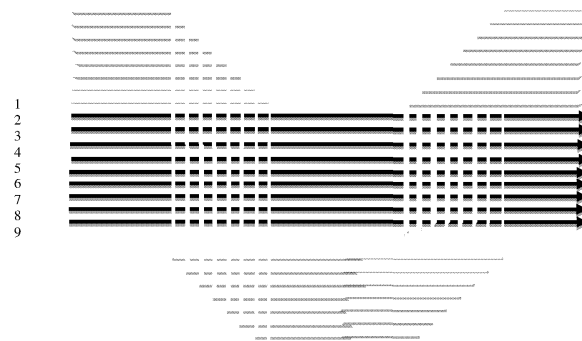
Фиг. 71



Фиг. 72



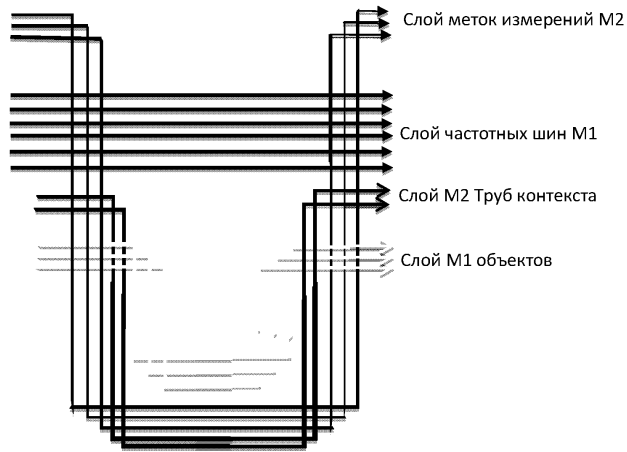
Фиг. 73



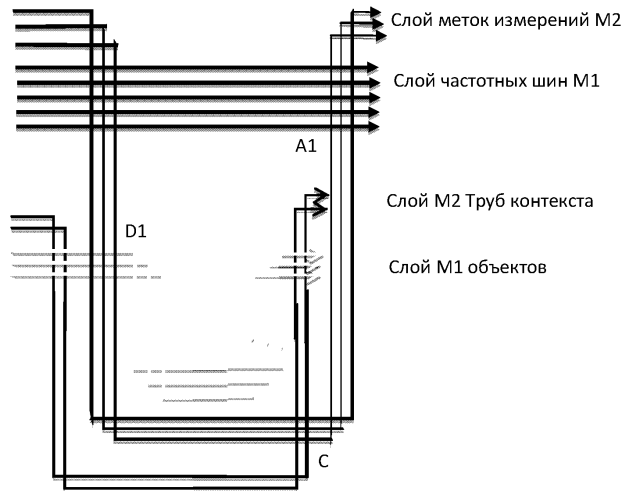
Фиг. 74



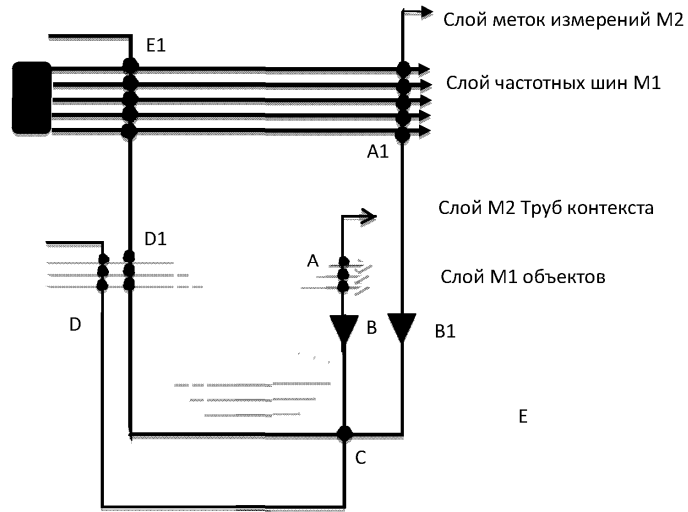
Фиг. 75



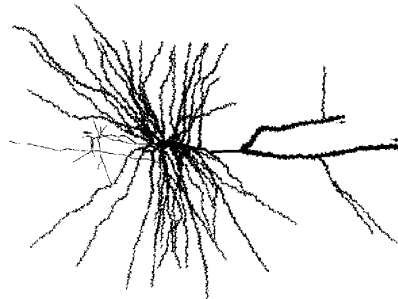
Фиг. 76



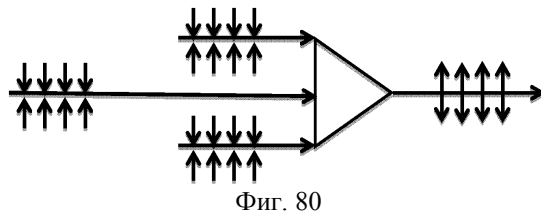
Фиг. 77



Фиг. 78



Фиг. 79



Фиг. 80

