

(19)



**Евразийское
патентное
ведомство**

(21) **201600637** (13) **A1**

(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ**

(43) Дата публикации заявки
2016.12.30

(51) Int. Cl. **G06E 1/00** (2006.01)

(22) Дата подачи заявки
2015.03.06

(54) **НЕЙРОННАЯ СЕТЬ И СПОСОБ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ**

(31) **61/949,210; 62/106,389**

(32) **2014.03.06; 2015.01.22**

(33) **US**

(86) **PCT/US2015/019236**

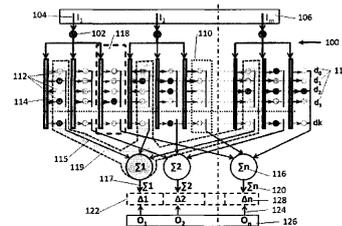
(87) **WO 2015/134900 2015.09.11**

(71) Заявитель:
ПРОГРЕСС, ИНК. (US)

(72) Изобретатель:
Песчанский Дмитрий Виленович (DE)

(74) Представитель:
Андреева М.Ю. (RU)

(57) Изобретение относится к искусственной нейронной сети и способу её обучения. Нейронная сеть содержит множество входов для получения входных сигналов, и синапсы, соединённые с входами и имеющие корректировочные веса. Сеть дополнительно содержит дистрибьюторы. Каждый дистрибьютор подключен к одному из входов для получения соответствующего входного сигнала и выбирает один или более корректировочных весов в соответствии со значением входного сигнала. Сеть также содержит нейроны. Каждый нейрон имеет выход и соединён по меньшей мере с одним из входов через один синапс и создаёт нейронную сумму путём суммирования корректировочных весов, выбранных из каждого синапса, подключенного к соответствующему нейрону. Кроме того, сеть содержит калькулятор коррекции веса, который получает желаемый выходной сигнал, определяет отклонение нейронной суммы от желаемого значения выходного сигнала и изменяет соответствующие корректировочные веса, используя это определённое отклонение. Суммирование этих изменённых корректировочных весов при определении нейронной суммы минимизирует рассматриваемое отклонение, таким образом обучая нейронную сеть. Технический результат выражается в сокращении времени обучения и уменьшении вычислительных ресурсов, необходимых для обучения нейронных сетей.



201600637
A1

201600637
A1

НЕЙРОННАЯ СЕТЬ И СПОСОБ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

ОТСЫЛКИ К РОДСТВЕННЫМ ЗАЯВКАМ

Данная заявка испрашивает приоритет согласно предварительной заявке США № 61/949,210, поданной 06 марта 2014, и предварительной заявке США № 62/106,389, поданной 22 января 2015, содержание которых включено в настоящий документ посредством ссылки.

ОБЛАСТЬ ТЕХНИКИ

Нижеследующее изобретение, в целом, относится к искусственной нейронной сети и способу её обучения.

ПРЕДШЕСТВУЮЩИЙ УРОВЕНЬ ТЕХНИКИ

В машинном обучении искусственные нейронные сети являются семейством статистических алгоритмов обучения, они построены по принципу биологических нейронных сетей, которые представляют собой центральные нервные системы животных, в частности головной мозг. Искусственные нейронные сети позволяют производить оценку и аппроксимацию функций многих переменных. Такие нейронные сети используются для решения широкого спектра задач, которые трудно решить с помощью обычного алгоритмического программирования, в том числе задач компьютерного зрения и распознавания речи.

Искусственные нейронные сети обычно представляют собой системы из специальных устройств - "нейронов", которые могут производить вычисления на основании величин сигналов, приходящих на их входы, и, в результате их адаптивной природы, способны к машинному обучению, а также к

распознаванию образов. Каждый нейрон обычно соединяется с несколькими входами сети через синапсы, которые имеют синаптические веса.

Нейронные сети не программируются как обычное программное обеспечение, а обучаются. Такое обучение обычно осуществляется посредством анализа необходимого количества репрезентативных примеров путем статистического или алгоритмического подбора синаптических весов, таким образом, чтобы заданный набор входных образов соответствует заданному набору выходных образов. Классические нейронные сети, как правило, критикуют за длительное время обучения и необходимость больших вычислительных ресурсов для их обучения.

Различные искусственные нейронные сети описаны в следующих патентах США: 4,979,124; 5,479,575; 5,493,688; 5,566,273; 5,682,503; 5,870,729; 7,577,631; и 7,814,038.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Нейронная сеть включает в себя множество входов нейронной сети, каждый вход сконфигурирован для приёма входного сигнала, имеющего определенное входное значение. Нейронная сеть также включает в себя множество синапсов, где каждый синапс соединён с одним из множества входов и включает в себя множество корректировочных весов, где каждый корректировочный вес определяется своим весовым значением (весом). Нейронная сеть также включает в себя набор дистрибьюторов (распределителей сигналов). Каждый дистрибьютор операционно соединён с одним из множества входов для приёма соответствующего входного сигнала и сконфигурирован для выбора одного или более корректировочных весов из множества корректировочных весов в соответствии с величиной входного сигнала. Нейронная сеть также включает в себя набор нейронов. Каждый нейрон имеет, по меньшей мере, один выход и соединён с, по меньшей мере,

одним из множества входов посредством одного из множества синапсов, при этом каждый нейрон сконфигурирован для того, чтобы суммировать весовые значения корректировочных весов, выбранных от каждого синапса, соединённого с соответствующим нейроном, и тем самым создавать нейронную сумму. Кроме того, нейронная сеть включает в себя калькулятор коррекции веса, сконфигурированный для получения желаемого выходного сигнала, имеющего определенное значение, для определения отклонения нейронной суммы от значения желаемого выходного сигнала и для изменения соответствующих значений корректировочных весов, используя определённое таким образом отклонение нейронной суммы. Суммирование изменённых значений корректировочных весов для определения нейронной суммы минимизирует отклонение нейронной суммы от желаемого значения выходного сигнала, что обеспечивает обучение нейронной сети.

Определение отклонения нейронной суммы от желаемого выходного сигнала может включать деление желаемого значения выходного сигнала на нейронную сумму, чтобы таким образом получить коэффициент отклонения. В таком случае, изменение значений соответствующих корректировочных весов может включать умножение каждого корректировочного веса, используемого для получения нейронной суммы, на коэффициент отклонения.

Отклонение нейронной суммы от желаемого выходного сигнала может в другом варианте представлять собой математическую разность между ними. В таком случае, получение соответствующих изменённых корректировочных весов может включать распределение математической разности между всеми корректировочными весами, используемыми для получения этой нейронной суммы. Такое распределение математической разности между всеми корректировочными весами служит для того, чтобы приблизить каждую нейронную сумму к желаемому значению сигнала.

Распределение математической разности может также включать деление определённой разности поровну между всеми корректировочными весами, использованными для получения нейронной суммы.

Дистрибьютор может быть дополнительно сконфигурирован для назначения множества коэффициентов воздействия соответствующему множеству корректировочных весов таким образом, что каждый коэффициент воздействия назначается одному из множества корректировочных весов в заранее заданной пропорции, чтобы получить соответствующую нейронную сумму.

Каждое соответствующее множество коэффициентов воздействия может определяться функцией распределения воздействия. Множество входных сигналов формируют диапазон значений сигналов, разделённый на интервалы в соответствии с функцией распределения интервалов таким образом, что каждое входное значение попадает в соответствующий его величине интервал, и каждый корректировочный вес соответствует одному из интервалов. Также каждый дистрибьютор может использовать соответствующее полученное входное значение сигнала для выбора соответствующего интервала. Кроме того, каждый дистрибьютор может назначать соответствующее множество коэффициентов воздействия корректировочному весу, соответствующему выбранному интервалу и, по меньшей мере, еще одному корректировочному весу, соответствующему интервалу, примыкающему к выбранному соответствующему интервалу.

Каждый нейрон может быть сконфигурирован так, чтобы суммировать произведения корректировочных весов с учетом назначенных коэффициентов воздействия для всех синапсов, соединённых с ним.

Заранее определённая пропорция коэффициентов воздействия может быть задана согласно статистическому распределению, например, с использованием распределения Гаусса.

Калькулятор коррекции веса может быть сконфигурирован так, чтобы назначать часть определённой ранее разности каждому корректировочному весу, используемому для получения нейронной суммы, согласно пропорции, установленной соответствующим коэффициентом воздействия.

Каждый корректировочный вес может дополнительно определяться набором индексов. Эти индексы могут включать входной индекс, сконфигурированный, чтобы идентифицировать корректировочный вес, соответствующий входу, индекс интервала, сконфигурированный, чтобы определять выбранный интервал для соответствующего корректировочного веса, а также нейронный индекс, сконфигурированный, чтобы определять корректировочный вес соответствующему нейрону.

Каждый корректировочный вес может дополнительно определяться с помощью индекса доступа, сконфигурированного, чтобы подсчитывать количество раз, когда входной сигнал обращается к соответствующему корректировочному весу во время обучения нейронной сети.

В изобретении раскрывается также способ обучения данной нейронной сети.

Вышеуказанные отличительные признаки и преимущества, а также другие отличительные признаки и преимущества настоящего краткого описания будут очевидно выражены в последующем подробном описании предпочтительного варианта(ов) и способа(ов) осуществления описанного изобретения, а также в прилагаемых чертежах и прилагаемой формуле изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

РИСУНОК 1 является иллюстрацией предшествующего уровня техники, классической искусственной нейронной сети.

РИСУНОК 2 является иллюстрацией “прогрессивной нейронной сети” (п-сети) имеющей множество синапсов, набор дистрибьюторов и множество корректировочных весов, связанных с каждым синапсом.

РИСУНОК 3А является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей множество синапсов и один синаптический вес, расположенный выше каждого дистрибьютора.

РИСУНОК 3В является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей множество синапсов и набор синаптических весов, расположенных ниже соответствующего множества корректировочных весов.

РИСУНОК 3С является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей множество синапсов и один синаптический вес, расположенный выше каждого дистрибьютора, и набор синаптических весов, расположенных ниже соответствующего множества корректировочных весов.

РИСУНОК 4А является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей один дистрибьютор для всех синапсов данного входа и один синаптический вес, расположенный выше каждого дистрибьютора.

РИСУНОК 4В является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей один дистрибьютор для всех синапсов данного входа и набор синаптических весов, расположенных ниже соответствующего множества корректировочных весов.

РИСУНОК 4С является иллюстрацией части п-сети, показанной на Рисунке 2, имеющей один дистрибьютор для всех синапсов данного входа и имеющей один синаптический вес, расположенный выше каждого дистрибьютора и набор синаптических весов, расположенных ниже соответствующего множества корректировочных весов.

РИСУНОК 5 является иллюстрацией разделения диапазона значений входного сигнала на индивидуальные интервалы в п-сети, показанной на Рисунке 2.

РИСУНОК 6А является иллюстрацией одного варианта распределения для значений коэффициента воздействия корректировочных весов в p -сети, показанной на Рисунке 2.

РИСУНОК 6В является иллюстрацией другого варианта распределения для значений коэффициента воздействия корректировочных весов в p -сети, показанной на Рисунке 2.

РИСУНОК 6С является иллюстрацией ещё одного варианта распределения для значений коэффициента воздействия корректировочных весов в p -сети, показанной на Рисунке 2.

РИСУНОК 7 является иллюстрацией входного образа для p -сети, показанной на Рисунке 2, а также одной соответствующей таблицы, представляющей образ в форме цифрового кода и другой соответствующей таблицы, представляющей тот же образ как набор соответствующих интервалов.

РИСУНОК 8 является иллюстрацией варианта воплощения p -сети, показанной на Рисунке 2, которая обучена распознавать два разных образа, при этом p -сеть сконфигурирована так, чтобы распознавать изображение, который включает некоторые черты каждого образа;

РИСУНОК 9 является иллюстрацией варианта воплощения p -сети, показанной на Рисунке 2, с примером распределения синаптических весов вокруг «центрального» нейрона.

РИСУНОК 10 является иллюстрацией варианта воплощения p -сети, показанной на Рисунке 2, отображающей равномерное распределение обучающих отклонений между корректировочными весами.

РИСУНОК 11 является иллюстрацией варианта воплощения p -сети, показанной на Рисунке 2, использующей изменение корректировочных весов во время обучения p -сети.

РИСУНОК 12 является иллюстрацией варианта воплощения п-сети, показанной на Рисунке 2, где основной алгоритм создаёт первичный набор выходных нейронных сумм и где созданный набор используется для создания нескольких «побеждающих» «сумм – победителей» с либо сохранившимися, либо увеличившимися значениями, а вклад остальных сумм уменьшается и сводится к нулю.

РИСУНОК 13 является иллюстрацией варианта воплощения п-сети, показанной на Рисунке 2, распознающей сложный образ с элементами нескольких разных образов.

РИСУНОК 14 является иллюстрацией модели объектно-ориентированного программирования для п-сети, показанной на Рисунке 2, при использовании Унифицированного Языка Моделирования (язык UML).

РИСУНОК 15 является иллюстрацией последовательности общего формирования п-сети, показанной на Рисунке 2.

РИСУНОК 16 является иллюстрацией типового анализа и подготовки данных для формирования п-сети, показанной на Рисунке 2.

РИСУНОК 17 является иллюстрацией типового процесса создания входа, делающего возможным взаимодействие п-сети, показанной на Рисунке 2, с входными данными во время обучения и применения п-сети.

РИСУНОК 18 является иллюстрацией типового процесса создания нейронных блоков для п-сети, показанной на Рисунке 2.

РИСУНОК 19 является иллюстрацией типового процесса создания каждого синапса, соединённого с нейронными блоками.

РИСУНОК 20 является иллюстрацией процесса обучения п-сети, показанной на Рисунке 2.

РИСУНОК 21 является иллюстрацией процесса обучения нейронного блока в п-сети, показанной на Рисунке 2.

РИСУНОК 22 является иллюстрацией наращивания нейронных сумм во время обучения п-сети, показанной на Рисунке 2.

РИСУНОК 23 является схемой последовательности операций способа, использованного для обучения нейронной сети, показанной на рисунках 2-22.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

Классическая искусственная нейронная сеть 10, как показано на Рисунке 1, обычно включает устройства ввода данных 12, синапсы 14 с синаптическими весами 16, нейроны 18, включающие сумматор 20 и устройство для активации функции 22, нейронные выходы 24 и калькулятор коррекции веса 26. Каждый нейрон 18 соединён через синапсы 14 с двумя или более устройствами ввода данных 12. Значения синаптических весов 16 обычно выражаются числовыми значениями записанными в ячейках памяти при цифровой (программной или аппаратной) реализации п-сети или с помощью электрического сопротивления, проводимости, напряжения, электрического заряда, магнитных свойств или других параметров при аналоговой реализации п-сети.

Обучение классической нейронной сети 10, как правило, основано на применении способа «обучение с учителем» то есть использовании набора обучающих пар 28. Каждая обучающая пара 28 обычно состоит из входного изображения 28-1 и желаемого выходного изображения 28-2, то есть набора «обучающих сигналов». Обучение классической нейронной сети 10 обычно осуществляется следующим образом. Входное изображение в форме набора входных сигналов (I_1-I_m) поступает на устройства ввода данных 12 и перемещается в синаптические веса 16 с изначальными весами (W_1). Значение входного сигнала изменяется весами, обычно умножением или делением каждого значения сигнала (I_1-I_m) на соответствующий вес. Из синаптических весов 16 изменённые входные сигналы передаются на соответствующие нейроны 18. Каждый нейрон 18 получает набор сигналов от группы синапсов

14, относящихся к этому нейрону 18. Сумматор 20, входящий в нейрон 18, суммирует все входные сигналы, изменённые весами и полученные этим нейроном и формирует нейронную сумму. Устройства активации 22 получают эти нейронные суммы и изменяют их в соответствии с заданной математической функцией активации, таким образом формируя соответствующие выходные сигналы всех нейронов в виде наборов выходных сигналов нейронов ($\sum F_1 \dots \sum F_n$).

Полученный выходной образ, определённый выходными сигналами нейронов ($\sum F_1 \dots \sum F_n$), сравнивается с помощью калькулятора коррекции веса 26 с заранее определёнными желаемыми выходными образами ($O_1 - O_n$). На основе определённой разности между полученным выходным образом нейрона $\sum F_n$ и желаемым выходным образом O_n , формируются, используя предварительно запрограммированный алгоритм, корректирующие сигналы для изменения синаптических весов 16. После корректировки всех синаптических весов 16 набор входных сигналов ($I_1 - I_m$) вновь вводится в нейронную сеть 10, и выполняются новые корректировки. Вышеуказанный цикл повторяется до тех пор, пока разность между полученным выходным образом нейрона $\sum F_n$ и желаемым выходным образом O_n не будет меньше некоторой заранее определённой величины ошибки. Один цикл обучения сети со всеми заданными образами обычно называется "учебной эпохой". Как правило, с каждой учебной эпохой величина ошибки уменьшается. В зависимости от числа входов ($I_1 - I_m$), выходов и нейронов, обучение классической нейронной сети 10 может потребовать значительного количества учебных эпох, которое, в некоторых случаях, может достигать сотен тысяч.

Существует множество различных классических нейронных сетей, в том числе сеть Хопфилда, ограниченная машина Больцмана, сеть радиально-базисных функций, рекуррентная нейронная сеть и другие. Конкретные задачи

классификации и кластеризации требуют применения определённого типа нейронной сети - Самоорганизующихся Карт Кохонена., которые используют только входные образы в качестве входных обучающих данных, при этом желаемый выходной образ, соответствующий определённому входному образу, формируется непосредственно в процессе обучения, на основе одного » «нейрона – победителя», имеющего выходной сигнал с максимальным значением.

Как было отмечено выше, одной из основных проблем, связанных с существующими классическими нейронными сетями, такими, как нейронная сеть 10, является то, что их успешное обучение может потребовать значительного количества времени. К другим проблемам классических сетей может относиться большое потребление вычислительных ресурсов, что в свою очередь вызывает потребность в мощных компьютерах. А также невозможность увеличить размер сети без полного переобучения этой сети (не масштабируемость сети), неспособность выполнить дополнительное обучение уже обученной сети, предрасположенность к таким явлениям, как "паралич сети" и "зависание на локальном минимуме", что не даёт возможности предсказать, будет ли конкретная нейронная сеть в состоянии обучиться с данным набором образов в данной последовательности. Кроме того, могут существовать ограничения, связанные с конкретной последовательностью образов, которые вводятся в процессе обучения, поскольку изменение порядка ввода обучающих образов может привести к «зависанию» сети,

Относительно остальных чертежей, на которых соответствующие ссылочные номера относятся к соответствующим компонентам, на Рисунке 2 показано схематическое изображение прогрессивной нейронной сети, далее "прогрессивная сеть", или "п-сеть" 100. П-сеть 100 включает в себя множество (набор) входов 102 п-сети. Каждый вход 102 сконфигурирован для приёма входного сигнала 104, при этом входные сигналы представлены как $I_1, I_2 \dots I_m$

на Рисунке 2. Каждый входной сигнал $I_1, I_2 \dots I_m$ представляет значение некоторой(ых) характеристики(ик) входного образа 106, например, величину, частоту, фазу, угол поляризации сигнала или ассоциируется с различными частями входного образа 106. Каждый входной сигнал 104 имеет конкретное входное значение, при этом вместе множество входных сигналов 104 описывают входной образ 106.

Каждое входное значение может находиться в пределах диапазона значений, который лежит между $-\infty$ и $+\infty$ и может быть установлен в цифровой и/ или аналоговой форме. Диапазон входных значений может зависеть от набора обучающих образов. В простейшем случае, диапазон входных значений может быть разностью между наименьшим и наибольшим значениями входных сигналов для всех обучающих образов. По практическим причинам, диапазон входных значений может быть ограничен за счёт устранения входных значений, которые считаются слишком высокими. Например, такое ограничение диапазона входных значений может быть достигнуто с помощью известных статистических методов уменьшения дисперсии, таких как выборка по значимости. Другим примером ограничения диапазона входных значений может быть назначение всем сигналам, которые ниже, чем заранее заданный минимальный уровень, определённого минимального значения и назначение всем сигналам, превышающим заранее заданный максимальный уровень, определённого максимального значения.

П-сеть 100 также включает в себя множество (набор) синапсов 118. Каждый синапс 118 соединён с одним из множества входов 102, включает в себя множество корректировочных весов 112, а также может включать синаптический вес 108, как показано на Рисунке 2. Каждый корректировочный вес 112 определяется соответствующим весовым значением 112. П-сеть 100 также включает в себя набор дистрибьюторов 114. Каждый дистрибьютор 114 операционно соединён с одним из множества входов 102 для приёма

соответствующего входного сигнала 104. Кроме того, каждый дистрибьютор 114 сконфигурирован так, чтобы выбирать один или несколько корректировочных весов из множества корректировочных весов 112 в соотношении с величиной входного сигнала.

П-сеть 100 дополнительно включает в себя набор нейронов 116. Каждый нейрон 116 имеет, по меньшей мере, один выход 117 и соединён с, по меньшей мере, одним из множества входов 102 посредством одного из синапсов 118. Каждый нейрон 116 сконфигурирован для того, чтобы суммировать или добавлять весовые значения корректировочных весов 112, выбранных от каждого синапса 118, соединённого с соответствующим нейроном 116, и тем самым создавать на выходе нейронную сумму 120, иначе обозначаемую как $\sum n$. Отдельный дистрибьютор 114 может быть использован для каждого синапса 118 данного входа 102, как показано на Рисунках 3А, 3В и 3С, или один дистрибьютор может быть использован для всех таких синапсов, как показано на Рисунках 4А, 4В и 4С. Во время формирования или настройки п-сети 100, всем корректировочным весам 112 присваиваются начальные значения, которые могут изменяться в процессе обучения п-сети. Начальное значение корректировочного веса 112 может быть назначено как в классической нейронной сети 10, например, веса могут быть выбраны случайным образом, вычислены с помощью заранее определённой математической функции, выбраны из заранее определённого шаблона и т.д.

П-сеть 100 также включает в себя калькулятор коррекции веса 122. Калькулятор коррекции веса 122 сконфигурирован для приёма желаемого, т.е. заранее определённого выходного сигнала 124, имеющего определенную величину сигнала, отражающего часть желаемого выходного образа 126. Калькулятор коррекции веса 122 также сконфигурирован для определения отклонения 128 нейронной суммы 120 от значения желаемого выходного сигнала 124, то есть ошибки обучения и для изменения соответствующих

значений корректировочных весов, используя определённое отклонение 128, чтобы таким образом суммируя изменённые значения корректировочных весов для определения нейронной суммы 120, минимизировать отклонение этой нейронной суммы от желаемого значения выходного сигнала 124 и таким образом обучать п-сеть 100.

По аналогии с классической сетью 10, рассматриваемой на Рисунке 1, отклонение нейронной суммы 128 также можно охарактеризовать как ошибку обучения между определённой нейронной суммой 120 и значением желаемого выходного сигнала 124. По сравнению с классической нейронной сетью 10, рассматриваемой на Рисунке 1, в п-сети 100 значения входного сигнала 104 изменяются при проходе через веса только в процессе предварительной настройки сети и не изменяются во время обучения п-сети. Вместо того, чтобы изменять проходящие сигналы, обучение п-сети 100 обеспечивается путём изменения значений корректировочных весов 112. Кроме того, хотя каждый нейрон 116 включает в себя суммирующую функцию, согласно которой нейрон суммирует значения корректировочных весов, нейрон 116 не требует, и, по сути, характеризуется отсутствием функции активации, которая, напротив, предусмотрена устройством функции активации 22 в классической нейронной сети 10.

В классической нейронной сети 10 коррекция веса во время обучения осуществляется путём изменения синаптических весов 16, а в п-сети 100 соответствующая коррекция веса обеспечивается за счёт изменения значений корректировочных весов 112, как показано на Рисунке 2. Соответствующие корректировочные веса 112 могут быть включены в блоки коррекции веса 110, расположенные на всех или некоторых синапсах 118. В нейросетевых компьютерных эмуляциях каждый из синаптических и корректировочных весов может быть представлен либо с помощью цифрового устройства, такого как ячейка памяти и/или с помощью аналогового устройства. В нейросетевых

программных эмуляциях значения корректировочных весов 112 могут генерироваться с помощью соответствующего запрограммированного алгоритма, а в аппаратных эмуляциях могут использоваться известные методы управления памятью.

В п-сети 100 отклонение 128 нейронной суммы 120 от желаемого выходного сигнала 124 может быть представлено в виде математически вычисленной разности между ними. Кроме того, генерация соответствующих изменённых корректировочных весов 112 может включать в себя пропорциональное распределение вычисленной разности между всеми корректировочными весами, используемым для получения нейронной суммы 120. В таком варианте осуществления п-сети получение соответствующих изменённых корректировочных весов 112 позволит нейронной сумме 120 быть приведённой к желаемому значению выходного сигнала в пределах небольшого количества эпох, а в некоторых случаях всего лишь за одну эпоху, что позволяет быстро обучить п-сеть 100. В конкретном случае, распределение математической разности между корректировочными весами 112, использованными для получения нейронной суммы 120, может включать деление определённой разности поровну между каждым корректировочным весом, использованным для получения соответствующей нейронной суммы 120.

В другом варианте осуществления п-сети определение отклонения 128 нейронной суммы 120 от желаемого значения выходного сигнала может включать разделение желаемого значения выходного сигнала на нейронную сумму, чтобы таким образом получить коэффициент отклонения. В данном случае изменение соответствующих изменённых корректировочных весов 112 включает умножение каждого корректировочного веса, используемого для получения нейронной суммы 120, на коэффициент отклонения. Каждый дистрибьютор 114 может быть дополнительно сконфигурирован для

назначения множества коэффициентов воздействия 134 множеству корректировочных весов 112. В настоящем варианте осуществления каждый коэффициент воздействия 134 может быть назначен одному из множества корректировочных весов 112 в заранее определённой пропорции, чтобы получить соответствующую нейронную сумму 120. Для соотнесения каждого корректировочного веса 112, соответствующему коэффициенту воздействия 134 может быть использована номенклатура коэффициента воздействия " $C_{i,d,n}$ ", как показано на Рисунках.

Каждый из множества коэффициентов воздействия 134, соответствующих определённому синапсу 118, определяется соответствующей функцией распределения воздействия 136. Функция распределения воздействия 136 может быть одинаковой для всех коэффициентов воздействия 134, либо только для множества коэффициентов воздействия 134, относящихся к определённому синапсу 118. Каждое из множества входных значений может быть получено в диапазоне значений 138, разделённом на интервалы или подразделы "d" в соответствии с функцией интервального распределения 140, таким образом, что каждое входное значение поступает в соответствующий интервал "d", и каждый корректировочный вес соответствует одному из этих интервалов. Каждый дистрибьютор 114 может использовать соответствующее полученное входное значение для выбора соответствующего интервала "d", и назначать соответствующее множество коэффициентов воздействия 134 корректировочному весу 112, соответствующему выбранному интервалу "d" и, по меньшей мере, одному корректировочному весу, соответствующему интервалу, примыкающему к выбранному интервалу, как например $W_{i,d+1,n}$ или $W_{i,d-1,n}$. В другом примере, не имеющем ограничительного характера, заранее заданное соотношение коэффициентов воздействия 134 может быть определено согласно статистическому распределению.

Получение нейронной суммы 120 может включать изначальное назначение соответствующих коэффициентов воздействия 134 каждому корректировочному весу 112 в соответствии со значением входного сигнала 102, затем умножение данных коэффициентов воздействия на значения соответствующих задействованных корректировочных весов 112, затем суммирование посредством каждого нейрона 116 отдельных произведений корректировочного веса 112 и назначенного коэффициента воздействия 134 для всех синапсов 118, соединённых с ним.

Калькулятор коррекции веса 122 может быть сконфигурирован так, чтобы назначать соответствующие коэффициенты воздействия 134, чтобы получать соответствующие изменённые корректировочные веса 112. В частности, калькулятор коррекции веса 122 может назначать часть вычисленной математической разности между нейронной суммой 120 и желаемым выходным сигналом 124 каждому корректировочному весу 112, используемому для получения нейронной суммы 120, согласно пропорции, установленной соответствующими коэффициентами воздействия 134. Кроме того, математическая разность, разделённая между корректировочными весами 112, использованными для получения нейронной суммы 120, может быть далее разделена на соответствующий коэффициент воздействия 134. Впоследствии результат деления нейронной суммы 120 на соответствующий коэффициент воздействия 134 может быть добавлен к корректировочному весу 112 для того, чтобы привести нейронную сумму 120 к желаемому значению выходного сигнала.

Обычно формирование п-сети 100 происходит до начала обучения п-сети. Тем не менее, в отдельном варианте осуществления, если во время обучения п-сеть 100 принимает входной сигнал 104, для которого изначальные корректировочные веса не были ранее сгенерированы, соответствующие величине сигнала корректировочные веса 112 могут быть сгенерированы. В

таком случае, определённый дистрибьютор 114 будет определять подходящий интервал "d" для определённого входного сигнала 104, и группа корректировочных весов 112 с начальными значениями будет генерироваться для данного входа 102, данного интервала "d" и всех соответствующих нейронов 116. Также, каждому сгенерированному новому корректировочному весу 112 может быть назначен соответствующий коэффициент воздействия 134

Каждый корректировочный вес 112 может быть определён с помощью набора индексов, сконфигурированных для идентификации позиции каждого соответствующего корректировочного веса в п-сети 100. Набор индексов может в частности включать входной индекс "i", сконфигурированный для идентификации корректировочного веса 112, соответствующего определённому входу 102, индекс интервала "d", сконфигурированный, чтобы определять обозначенный выше выбранный интервал соответствующему корректировочному весу и нейронный индекс "n", сконфигурированный, чтобы определять корректировочный вес 112 соответствующему нейрону 116 с номенклатурой " $W_{i,d,n}$ ". Таким образом, каждому корректировочному весу 112, соответствующему определённому входу 102, назначается определённый индекс подстрочный "i", чтобы обозначить позицию субъекта. Аналогичным образом каждому корректировочному весу "W", соответствующему определённому нейрону 116 и соответствующему синапсу 118 назначаются определённые подстрочные индексы "n" и "d", чтобы обозначить данную позицию корректировочного веса в п-сети 100. Набор индексов может также включать в себя подстрочный индекс доступа "a", сконфигурированный для подсчета количества раз, когда входной сигнал 104 обращается к соответствующему корректировочному весу 112 во время обучения п-сети 100. Другими словами, каждый раз, когда определённый интервал "d" и соответствующий корректировочный вес 112 выбираются для обучения из

множества корректировочных весов в соответствии с входным значением, индекс доступа "a" увеличивается для подсчёта входного сигнала. Индекс доступа "a" может быть использован для дальнейшего определения текущего состояния или статуса каждого корректировочного веса, путём принятия номенклатуры " $W_{i,d,n,a}$ ". Каждому из индексов "i", "d", "n" и "a" может быть назначено номинальное значение в диапазоне от 0 до $+\infty$.

Различные возможности деления диапазона входных сигналов 104 на интервалы $d_0, d_1 \dots d_m$ показаны на Рисунке 5. Определённое распределение интервалов может быть однородным или линейным, что, например, может быть достигнуто путём определения всех интервалов "d" с одинаковым размером. Все входные сигналы 104, имеют свои конкретные значения, которые ниже заранее определённого минимального уровня могут рассматриваться как имеющие нулевое значение, в то время как всем входным сигналам, имеющим свои значения в превышающие заранее определённый максимальный уровень, может быть назначен заданный определённый максимальный уровень, как также показано на Рисунке 5. Заданное распределение интервалов также может быть неравномерным или нелинейным, как, например, симметричное, асимметричное или неограниченное. Нелинейное распределение интервалов "d" может быть полезно, когда диапазон входных сигналов 104 может рассматриваться как непрактично большой, и определённая часть диапазона может включать входные сигналы, которые считаются наиболее важными, например, в начале, в середине или в конце диапазона. Распределение интервалов также может быть описано с помощью случайной функции. Все предыдущие примеры не имеют ограничительного характера, так что и другие варианты распределения интервалов тоже возможны.

Количество интервалов "d", находящихся в пределах выбранного диапазона входных сигналов 104, может быть увеличено с целью оптимизации

п-сети 100. Такая оптимизация п-сети 100 может потребоваться, например, в связи с увеличением сложности входных образов для обучения 106. Например, большее количество интервалов может потребоваться для многоцветных изображений, чем для моно-цветных изображений, и большее количество интервалов может потребоваться для сложных орнаментов, чем для простой графики. Увеличенное число интервалов может потребоваться для точного распознавания изображений со сложными цветовыми градиентами, в отличие от изображений, описанных контурами, а также при большем общем количестве обучающих образов. Уменьшение числа интервалов "d" также может быть необходимо в случаях с высокой величиной шума и/или помех, высокой разнородность обучающих образов, а также при чрезмерном потреблении вычислительных ресурсов.

В зависимости от задачи или типа информации, обрабатываемой п-сетью 100, например, визуальные или текстовые данные, данные с датчиков различного характера, может быть назначено разное количество интервалов и тип их распределения. Для каждого интервала "d" входного значения сигнала может быть назначен соответствующий корректировочный вес данного синапса с индексом "d". Таким образом, некоторый интервал "d" будет включать все корректировочные веса 112 с индексом "i", относящиеся к данному входу, индексом "d", относящимся к данному интервалу и все значения индекса "n" от 0 до n. В процессе обучения п-сети 100 дистрибьютор 114 определяет каждое значение входного сигнала, и, таким образом, относит данный входной сигнал 104 к соответствующему интервалу "d". Например, если существуют 10 равных интервалов "d" в пределах диапазона входных сигналов от 0 до 100, входной сигнал, имеющий значение между 30 и 40, будет связан с интервалом 3, то есть, "d" = 3.

Для всех корректировочных весов 112 каждого синапса 118, соединённого с данным входом 102, дистрибьютор 114 может назначать

значения коэффициента воздействия 134 в соответствии с интервалом "d", связанным с определённым входным сигналом. Дистрибьютор 114 может также назначать значения коэффициента воздействия 134 в соответствии с заранее определённым распределением значений коэффициента воздействия 134 (как показано на Рисунке 6), таким как кривая синусоидального, нормального, логарифмического распределения или функция случайного распределения. Во большинстве случаев сумма или интеграл коэффициента воздействия 134 или $C_{i,d,n}$ для конкретного входного сигнала 102, относящегося к каждому синапсу 118, будет иметь значение 1 (единица).

$$\sum_{Synapse} C_{i,d,n} = 1 \quad \text{or} \quad \int_{Synapse} C_{i,d,n} = 1 [1]$$

В простейшем случае корректировочному весу 112, который наиболее близко соответствует значению входного сигнала, может быть назначено значение коэффициента воздействия 134 $C_{i,d,n} = 1$, в то время как корректировочные веса для других интервалов могут получить значение коэффициента воздействия $C_{i,d,n} = 0$.

П-сеть 100 ориентирована на сокращение продолжительности обучения и использования других ресурсов во время обучения п-сети по сравнению с классической нейронной сетью 10. Хотя некоторые из элементов, описанных здесь как часть п-сети 100, обозначены определёнными названиями или терминами, которые известны тем, кто знаком с классическими нейронными сетями, данные названия используются для простоты и могут быть применены в значении, отличающемся от их аналогов в классических нейронных сетях. Например, синаптические веса 16, управляющие величинами входных сигналов (I_1 - I_m), задаются в процессе общей настройки классической нейронной сети 10 и изменяются в процессе обучения классической сети. С

другой стороны, обучение п-сети 100 выполняется за счёт изменения корректировочных весов 112, в то время как синаптические веса 108 не изменяются в процессе обучения. Кроме того, как описано выше, каждый из нейронов 116 включает сумматор, но не включает в себя устройство активации 22, что характерно для классической нейронной сети 10.

Обычно п-сеть 100 обучается путём обучения каждого нейронного блока 119, который содержит соответствующий нейрон 116 и все соединённые с ним синапсы 118, и корректировочные веса 112, соединённые с данным нейроном. Соответственно, обучение п-сети 100 включает изменение корректировочных весов 112, вносящих вклад в соответствующий нейрон 116. Изменения в корректировочных весах 112 происходят на основе алгоритма группового обучения, включённого в способ 200, подробно описанный ниже. В описанном алгоритме ошибка обучения, то есть отклонение 128, определяется для каждого нейрона, на основании чего каждому из весов 112, использованных при определении суммы, полученной на каждом соответствующем нейроне 116 определяются и назначаются корректирующие значения. Введение подобных корректирующих значений во время обучения предназначено для сокращения отклонений 128 данного нейрона 116 до нуля. Во время обучения дополнительным образом могут появиться новые ошибки, связанные с образами, использованными ранее. Для устранения таких вторичных ошибок, после завершения одной обучающей эпохи могут быть вычислены ошибки для всех обучающих образов всей п-сети 100 и, если подобные ошибки будут больше, чем заранее заданные значения, могут быть проведены одна или несколько дополнительных обучающих эпох, пока ошибки не станут меньше, чем заранее намеченное определённое значение.

На рисунке 23 изображён способ 200 обучения п-сети 100, который был описан выше в отношении Рисунков 2-22. Способ 200 начинается в фрейме 202, в котором способ включает получение через вход 102 входного сигнала

104, имеющего определенное значение. После фрейма 202 способ переходит к фрейму 204. В фрейме 204 способ включает передачу входного сигнала 104 дистрибьютору 114, операционно соединённому со входом 102. Либо в фрейме 202, либо в фрейме 204 способ 200 может включать определение каждого корректировочного веса 112 с помощью набора индексов. Как было описано выше относительно структуры п-сети 100, набор индексов может включать в себя входной индекс "i", сконфигурированный, чтобы идентифицировать корректировочный вес 112, соответствующий входу 102. Набор индексов может также включать в себя индекс интервала "d", сконфигурированный, чтобы определять выбранный интервал для соответствующего корректировочного веса 112 и нейронный индекс "n", сконфигурированный, чтобы определять корректировочный вес 112 соответствующему нейрону 116 как " $W_{i,d,n}$ ". Набор индексов может также включать в себя индекс доступа "a", сконфигурированный, чтобы подсчитывать количество раз, когда входной сигнал 104 обращается к соответствующему корректировочному весу 112 во время обучения п-сети 100. Более того, текущее состояние/статус каждого корректировочного веса может принять номенклатуру " $W_{i,d,n,a}$ ".

После фрейма 204 способ переходит к фрейму 206, в котором способ включает выбор, посредством дистрибьютора 114, в соответствии с значением входного сигнала, одного или нескольких корректировочных весов 112 из множества корректировочных весов, расположенных на синапсе 118, соединённом с данным входом 102. Как было описано выше, каждый корректировочный вес 112 определяется его соответствующим весовым значением. В фрейме 206 способ может дополнительно включать в себя назначение, посредством дистрибьютора 114, множества коэффициентов воздействия 134 множеству корректировочных весов 112. В фрейме 206 способ может также включать в себя назначение каждого коэффициента воздействия 134 одному из множества корректировочных весов 112 в заранее определённой

пропорции, чтобы получить нейронную сумму 120. Помимо этого, в фрейме 206 способ может включать суммирование, посредством нейрона 116, произведения корректировочного веса 112 и назначенного коэффициента воздействия 134 для всех синапсов 118, соединённых с данным нейроном. Кроме того, в фрейме 206 метод может включать назначение, посредством калькулятора коррекции веса 122, части определённой разности каждому корректировочному весу 112, использованному для получения нейронной суммы 120, согласно пропорции, установленной соответствующим коэффициентом воздействия 134.

Как было описано выше относительно структуры п-сети 100, множество коэффициентов воздействия 134 может быть определено с помощью функции распределения воздействия 136. В таком случае способ может дополнительно включать получение входного значения в диапазоне значений 138, разделённом на интервалы "d" в соответствии с функцией распределения интервалов 140, таким образом, что каждое входное значение поступает в соответствующий интервал, и каждый корректировочный вес 112 соответствует одному из интервалов. Кроме того, данный способ может включать использование, посредством дистрибьютора 114, полученного входного значения для выбора соответствующего интервала "d" и для назначения соответствующего множества коэффициентов воздействия 134 корректировочному весу 112, соответствующему выбранному интервалу "d" и, по меньшей мере, одному корректировочному весу, соответствующему интервалу, примыкающему к выбранному интервалу "d". Как описано выше относительно структуры п-сети 100, корректировочные веса 112, соответствующие интервалу, примыкающему к выбранному интервалу "d", могут быть идентифицированы, например, как $W_{i,d+1,n}$ или $W_{i,d-1,n}$.

После фрейма 206 способ переходит к фрейму 208. В фрейме 208, способ включает суммирование значений выбранных корректировочных весов 112

определённым нейроном 116, соединённым с входом 102 посредством синапса 118 чтобы создать нейронную сумму. Как описано выше относительно структуры п-сети 100, каждый нейрон 116 имеет, по меньшей мере, один выход 117. После фрейма 208 способ переходит к фрейму 210, в котором способ включает получение калькулятором коррекции веса 122 желаемого выходного сигнала 124, имеющего определенное значение. После фрейма 210 способ переходит к фрейму 212, в котором способ включает определение калькулятором коррекции веса 122 отклонения 128 нейронной суммы 120 от желаемого значения выходного сигнала 124.

Как сказано выше в описании п-сети 100, определение отклонения 128 нейронной суммы 120 от желаемого выходного сигнала может включать определение математической разности между ними. Кроме того, изменение соответствующих корректировочных весов 112 может включать распределение математической разности между всеми корректировочными весами, используемыми для получения нейронной суммы 120. В качестве альтернативы распределению математической разности может быть деление определённой разности поровну между каждым корректировочным весом 112, используемым для получения нейронной суммы 120. В ещё одном варианте осуществления, определение отклонения 128 может также включать разделение значения желаемого выходного сигнала 124 на нейронную сумму 120, чтобы таким образом получить коэффициент отклонения. Кроме того, в таком случае изменение соответствующих корректировочных весов 112 может включать умножение каждого корректировочного веса 112, используемого для получения нейронной суммы 120, на полученный коэффициент отклонения.

После фрейма 212 способ переходит к фрейму 214. В фрейме 214 способ включает изменение, посредством калькулятора коррекции веса 122, соответствующих значений корректировочных весов, с использованием определённого отклонения 128. Изменённые значения корректировочных весов

впоследствии могут быть просуммированы, а затем использованы для определения новой нейронной суммы 120. Суммированные изменённые значения корректировочных весов могут затем служить для того, чтобы минимизировать отклонение нейронной суммы 120 от значения желаемого выходного сигнала 124, тем самым обучая п-сеть 100. После фрейма 214 способ 200 может включать возвращение к фрейму 202, чтобы выполнять дополнительные обучающие эпохи до тех пор, пока отклонение нейронной суммы 120 от значения желаемого выходного сигнала 124 не будет сведено к достаточному минимуму. Другими словами, дополнительные обучающие эпохи могут быть выполнены, чтобы привести нейронную сумму 120 к желаемому выходному сигналу 124 в пределах заранее заданного отклонения или величины ошибки, таким образом, что п-сеть 100 можно будет считать обученной и готовой к работе с новыми образами.

Как правило, входные образы 106 должны быть подготовлены для обучения п-сети 100. Подготовка п-сети 100 к обучению обычно начинается с формирования набора обучающих образов, включая входные образы 106 и, во многих случаях, желаемые выходные образы 126, соответствующие данным входным образам. Входные образы 106 (показанные на Рисунке 2), которые определяются входными сигналами $I_1, I_2 \dots I_m$, для обучения п-сети 100 выбираются в соответствии с задачами, для решения которых формируется п-сеть, например, для распознавания изображений людей или иных объектов, распознавания определённой деятельности, кластеризации или классификации данных, анализа статистических данных, распознавания паттернов, прогнозирования или контроля определённых процессов. Соответственно, входные образы 106 могут быть представлены в любом формате, подходящем для введения в компьютер, например, форматы JPEG, GIF или PPTX, в виде таблиц, схем, диаграмм и графиков, также различные форматы документов или набор символов.

Подготовка к обучению п-сети 100 может также включать преобразование выбранных входных образов 106 для их унификации, что удобно для обработки данных образов с помощью п-сети 100, например, переводя все образы в формат, имеющий одинаковое количество сигналов, или, в случае изображений, одинаковое количество пикселей. Цветные изображения могут быть, например, представлены в виде комбинации трёх основных цветов. Преобразование образов может также включать изменение характеристик, например, передвижение изображения в пространстве, изменение визуальных характеристик образа, таких как разрешение, яркость, контрастность, цвета, точка просмотра, перспектива, фокусное расстояние и центр фокуса, а также добавление символов, цифр или записей.

После выбора количества интервалов, определённый входной образ может быть преобразован во входной образ в формате интервалов, то есть значения реального сигнала могут быть записаны в виде номеров или обозначений интервалов, к которым принадлежат данные соответствующие сигналы. Эта процедура может проводиться в каждой обучающей эпохе для данного образа. Однако образ также может быть сформирован в виде набора интервальных чисел единой длины. Например, на Рисунке 7 исходный образ представлено в виде изображения, в то время как в таблице "Изображение в цифровом формате" то же самое изображение представлено в форме цифровых кодов, а затем в таблице "Изображение в формате интервалов" изображение представлено в виде набора интервальных чисел, в котором диапазон сигналов разделен на 10 интервалов.

Описанная структура р-сети 100 и обучающего алгоритма или способа 200, как указано, допускает непрерывное или итеративное обучение п-сети, таким образом нет обязательной необходимости формировать полный набор обучающих входных образов 106 в начале процесса обучения. Можно сформировать сравнительно небольшой стартовый набор обучающих образов,

и данный стартовый набор может быть расширен по мере необходимости. Входные образы 106 могут быть разделены на различные категории, например, набор фотографий одного человека, набор фотографий кошек или набор фотографий автомобилей, таким образом, что каждая категория соответствует одному выходному образу, как например имя человека или конкретный знак. Желаемые выходные образы 126 представляют собой цифровое поле или таблицу, где каждая точка соответствует определённому числовому значению от $-\infty$ до $+\infty$ или аналоговому значению. Каждая точка желаемого выходного образа 126 может соответствовать выходу одного из нейронов п-сети 100. Желаемые выходные образы 126 могут быть закодированы с помощью цифровых или аналоговых кодов изображений, таблиц, текста, формул, наборов символов, таких как штрих-коды или звуки.

В простейшем случае каждый входной образ 106 может соответствовать выходному образу, кодирующему данный входной образ. Одной из точек такого выходного образа может быть назначено максимально возможное значение, например, 100%, в то время как всем остальным точкам могут быть назначены минимально допустимые значения, например, ноль. В таком случае, после обучения, будет возможно вероятностное распознавание различных образов в форме процентного сходства с обучающими образами. На Рисунке 8 показан пример того, как п-сеть 100, обученная распознаванию двух образов, квадрата и круга, может распознавать изображение, содержащее некоторые черты каждой фигуры, выраженные в процентах, причём сумма не обязательно равна 100%. Такой процесс распознавания образов, определяющий процент сходства между различными изображениями, используемыми для обучения, может быть использован для классификации конкретных изображений.

Чтобы повысить точность и исключить ошибки, кодирование может быть выполнено с использованием набора нескольких нейронных выходов, а не одного выхода (см. ниже). В простейшем случае, выходные образы могут

быть подготовлены до обучения. Тем не менее, также возможно, чтобы п-сеть 100 сформировала выходные образы во время обучения.

В п-сети 100 есть также возможность инвертировать входные и выходные образы. Другими словами, входные образы 106 могут быть заданы в виде поля или таблицы цифровых или аналоговых значений, где каждая точка соответствует одному входу п-сети, в то время как выходные образы могут быть представлены в любом формате, подходящем для введения в компьютер, например, с использованием форматов JPEG, GIF, PPTX, в виде таблиц, схем, диаграмм и графиков, документов различных форматов или набора символов. Полученная п-сеть 100 может быть вполне пригодна для систем архивирования, а также для ассоциативного поиска изображений, музыкальных выражений, уравнений или наборов данных.

После подготовки входных образов 106, как правило, необходимо сформировать п-сеть 100 или установить параметры существующей п-сети для работы с данной задачей (задачами). Формирование п-сети 100 может включать определение следующих параметров:

- размеры п-сети 100, определяемые числом входов и выходов;
- величины синаптическиз весов 108 для всех входов;
- количество корректировочных весов 112;
- распределение коэффициентов воздействия корректировочных весов ($C_{i,d,n}$) для разных значений входных сигналов 104; и
- желаемая точность обучения.

Количество входов определяется на основании размеров входных образов 106. Например, для изображений количество входов может быть равно количеству пикселей в изображении в то время как выбранное количество выходов может зависеть от размера желаемых выходных образов

126. В некоторых случаях выбранное количество выходов может зависеть от количества категорий обучающих изображений.

Значения отдельных синаптических весов w_{ij} могут находиться в диапазоне от $-\infty$ до $+\infty$. Значения синаптических весов w_{ij} , которые меньше, чем 0 (ноль), могут означать усиление сигнала, которое может быть использовано для усиления воздействия сигналов от определённых входов или от определённых образов, например, для более эффективного распознавания человеческих лиц на фотографиях, содержащих большое количество разных людей или объектов. С другой стороны, значения синаптических весов w_{ij} , которые больше, чем 0 (ноль), могут быть использованы для обозначения ослабления сигнала, которое может быть использовано для уменьшения количества необходимых расчётов и повышения рабочей скорости п-сети 100. Обычно, чем больше значение синаптического веса, тем более ослабленным является сигнал, передаваемый на соответствующий нейрон. Если все синаптические веса w_{ij} , соответствующие всем входам равны, и все нейроны одинаково связаны со всеми входами, нейронная сеть становится универсальной и наиболее эффективной для выполнения задач общего характера, например, когда очень мало известно о природе образов заранее. Тем не менее, такая структура, как правило, увеличивает количество необходимых расчётов во время обучения и работы.

На рисунке 9 показан вариант осуществления п-сети 100, в котором взаимодействие между входом и соответствующими нейронами уменьшается в соответствии со статистически нормальным (гауссовым) распределением. Неравномерное распределение синаптических весов w_{ij} может привести к тому, что весь входной сигнал направляется на намеченный или "центральный" нейрон для данного входа, таким образом присваивая значение нуля данному синаптическому весу. Кроме того, неравномерное распределение синаптических весов может привести к тому, что другие нейроны получат

сниженные значения входного сигнала, например, с использованием нормального, логонормального, синусоидального или другого распределения. Значения синаптических весов 10^8 для нейронов 116 , получающих сниженные значения входных сигналов, могут увеличиваться вместе с увеличением их расстояния от "центрального" нейрона. В таком случае количество вычислений может быть уменьшено и функционирование п-сети может ускориться. Такие сети, которые представляют собой сочетание известных полносвязных и неполносвязных нейронных сетей, могут быть чрезвычайно эффективными для анализа образов с сильными внутренними паттернами, например, таких как человеческие лица или последовательные кадры кинофильма.

На рисунке 9 показан вариант п-сети 100 , который эффективен для распознавания локальных паттернов. Для того, чтобы улучшить идентификацию общих паттернов, $10-20\%$ от сильных соединений, где значения синаптических весов 10^8 малы или равны нулю, могут быть распределены по всей п-сети 100 , в детерминистическом, например в форме сети, или случайном (вероятностном) порядке. Фактическое формирование п-сети 100 , предназначенной для работы с конкретной задачей, осуществляется с помощью программы, например, написанной на объектно-ориентированном языке программирования, которая генерирует основные элементы п-сети, такие как синапсы, синаптические веса, дистрибьюторы, корректировочные веса, нейроны и т.д., как программные объекты. Такая программа может назначить отношения между отмеченными объектами и алгоритмами, определяющими их действия. В частности, синаптические и корректировочные веса могут быть сформированы в начале формирования п-сети 100 , вместе с установкой их начальных значений. П-сеть 100 может быть полностью сформирована до начала её обучения, и может быть изменена или дополнена на более позднем фрейме, что необходимо, например, когда информационная ёмкость сети

истощается, или в случае фатальной ошибки. Завершение построения п-сети 100 также возможно в процессе обучения.

Если п-сеть 100 формируется заранее, количество выбранных корректировочных весов на определённом синапсе может быть равно количеству интервалов в диапазоне входных сигналов. Кроме того, корректировочные веса могут быть сгенерированы после формирования п-сети 100, в качестве ответа на появление определенных интервалов. По аналогии с классической нейронной сетью 10, выбор параметров и настроек п-сети 100 осуществляется серией направленных экспериментов. Такие эксперименты могут включать (1) формирование п-сети с одинаковыми синаптическими весами 108 на всех входах и (2) оценку значений входных сигналов для выбранных образов и первоначальный выбор количества интервалов. Например, для распознавания бинарных (одноцветных) изображений, может быть достаточно иметь только 2 интервала; для качественного распознавания 8-битных изображений, может быть использовано до 256 интервалов; аппроксимация сложных статистических зависимостей может потребовать десятков или даже сотен интервалов; для больших баз данных количество интервалов может измеряться в тысячах.

В процессе обучения п-сети 100 значения входных сигналов могут округляться по мере того, как они распределяются между определёнными интервалами. Таким образом, точность входных сигналов, превышающая ширину диапазона, делённого на количество интервалов, может не потребоваться. Например, если диапазон входного значения устанавливается для 100 единиц, а число интервалов 10, не потребуется точность более чем ± 5 . Эксперименты могут также включать (3) выбор равномерного распределения интервалов по всему диапазону значений входных сигналов и простейшее распределение для коэффициентов воздействия корректировочных весов $C_{i,d,n}$ может приравняться к 1 для корректировочных весов, соответствующих

интервалу для конкретного входного сигнала, в то время как коэффициенты воздействия для всех остальных корректировочных весов может приравняться к 0 (нулю). Эксперименты могут дополнительно включать (4) обучение п-сети 100 с одним, несколькими или всеми подготовленными обучающими образами с заранее определённой точностью.

Время обучения п-сети 100 для заранее определённой точности может быть установлено экспериментальным путём. Если точность и время обучение п-сети 100 являются удовлетворительными, выбранные настройки могут быть либо сохранены, либо изменены, для продолжения поиска более эффективного варианта. Если требуемая точность не достигнута, в целях оптимизации может быть оценено влияние конкретных изменений, которое может проводиться по одному, либо группами. Такая оценка изменений может включать изменение, либо увеличение либо уменьшение, количества интервалов; изменение типа распределения коэффициентов воздействия корректировочных весов ($C_{i,d,n}$), тестирование вариантов с неравномерным распределением интервалов, таких как использование нормального, степенного, логарифмического или логонормального распределения; и изменение значений синаптических весов 108, например, их переход к неравномерному распределению.

Если требуемое время обучения для точного результата считается чрезмерным, обучение с увеличенным числом интервалов можно оценить с точки зрения его влияния на время обучения. Если в результате время обучения было сокращено, увеличение числа интервалов может повторяться до тех пор, пока желаемое время обучения не будет достигнуто без потери необходимой точности. Если время обучения растёт с увеличением числа интервалов вместо того, чтобы уменьшаться, может быть проведено дополнительное обучение с уменьшенным числом интервалов. Если сокращение числа интервалов приводит к снижению времени обучения,

количество интервалов может далее уменьшаться до получения желаемого времени обучения.

Формирование установок п-сети 100 может осуществляться посредством обучения с заранее определённым временем обучения и экспериментальным определением точности обучения. Параметры могут быть улучшены с помощью экспериментальных изменений, аналогичных тем, которые описаны выше. Реальная практика с различными п-сетями показала, что процедура выбора установок, как правило, не вызывает затруднений и не отнимает много времени.

Фактическое обучение п-сети 100 как часть способа 200, показанного на Рисунке 23, начинается с подачи входного образа сигналов $I_1, I_2 \dots I_m$ на устройства ввода сети 102, откуда они передаются на синапсы 118, проходят через синаптический вес 108 и поступают на дистрибьютор (или группу дистрибьюторов) 114. В зависимости от значения входного сигнала дистрибьютор 114 устанавливает число интервала "d", которому соответствует данный входной сигнал 104, и назначает коэффициенты воздействия корректировочных весов $C_{i,d,n}$ для всех корректировочных весов 112 блоков коррекции веса 110 всех синапсов 118, соединённых с соответствующим входом 102. Например, если интервалу "d" может быть установлено значение 3 для первого входа, для всех весов $W_{1,3,n}$ $C_{1,3,n} = 1$, в то время как для всех других весов $i \neq 1$ и $d \neq 3$, $C_{i,d,n} = 0$

Для каждого нейрона 116, обозначенного как "n" в уравнении [2], приведенном ниже, выходные нейронные суммы $\sum 1, \sum 2 \dots \sum n$ формируются путём умножения каждого корректировочного веса 112, обозначенного как $W_{i,d,n}$ в уравнении [2], на соответствующий коэффициент воздействия корректировочного веса $C_{i,d,n}$ для всех синапсов 118 вносящих вклад в определённый нейрон и путём сложения всех полученных значений:

$$\sum_n = \sum_{i,d,n} W_{i,d,n} \times C_{i,d,n} [2]$$

Умножение $W_{i,d,n} \times C_{i,d,n}$ может быть выполнено различными устройствами, например дистрибьюторами 114, устройствами хранящими веса или напрямую нейронами 116. Суммы передаются через нейронный выход 117 на калькулятор коррекции веса 122. Желаемые выходные сигналы $O_1, O_2 \dots O_n$, описывающие желаемый выходной образ, 126 также направляются в калькулятор 122.

Как сказано выше, калькулятор коррекции веса 122 является вычислительным устройством для подсчёта изменённого значения корректировочных весов путём сравнения выходных нейронных сумм $\sum 1, \sum 2 \dots \sum n$ с желаемыми выходными сигналами $O_1, O_2 \dots O_n$. Рисунок 11 показывает набор корректировочных весов $W_{i,d,1}$, вносящих вклад в нейронную выходную сумму $\sum 1$, которые умножаются на соответствующий коэффициент воздействия корректировочного веса $C_{i,d,1}$, и эти произведения впоследствии суммируются выходной нейронной суммой $\sum 1$:

$$\sum 1 = W_{1,0,1} \times C_{1,0,1} + W_{1,1,1} \times C_{1,1,1} + W_{1,2,1} \times C_{1,2,1} + \dots [3]$$

При начале обучения, например, во время первой эпохи, корректировочные веса $W_{i,d,1}$ не соответствуют входному образу 106, используемому для обучения, таким образом, выходные нейронные суммы $\sum 1$ не равняются соответствующему желаемому выходному образу 126. На основе начальных корректировочных весов $W_{i,d,1}$, система коррекции веса вычисляет корректирующее значение $\Delta 1$, которое используется для изменения всех корректировочных весов, вносящих вклад в выходную нейронную сумму $\sum 1$ ($W_{i,d,1}$). П-сеть 100 допускает различные варианты для её формирования и

использования общих корректирующих сигналов для всех корректировочных весов $W_{i,d,n}$, вносящих вклад в определённый нейрон 116.

Ниже приведены два показательных варианта не имеющие ограничительного характера для формирования и использования общих корректирующих сигналов. Вариант 1 – формирование и использование корректирующих сигналов на основе разности между желаемыми выходными сигналами и полученными выходными суммами следующим образом:

- Вычисление равного корректирующего значения Δ_n для всех корректировочных весов, вносящих вклад в нейрон “n” согласно уравнению:

$$\Delta_n = (O_n - \sum_n) / S [4],$$

Где:

O_n – желаемый выходной сигнал, соответствующий выходной нейронной сумме \sum_n ;

S – количество синапсов, соединённых с нейроном “n”.

- Изменение всех корректировочных весов $W_{i,d,n}$, вносящих вклад в нейрон “n” согласно уравнению:

$$W_{i,d,n} \text{ modified} = W_{i,d,n} + \Delta_n / C_{i,d,n} [5],$$

Вариант 2 – формирование и использование корректирующих сигналов на основе соотношения желаемых выходных сигналов и полученных выходных сумм следующим образом:

- Вычисление равного корректирующего значения Δ_n для всех корректировочных весов, вносящих вклад в нейрон “n” согласно уравнению:

$$\Delta_n = O_n / \sum_n [6],$$

- Изменение всех корректировочных весов $W_{i,d,n}$, вносящих вклад в нейрон “n” согласно уравнению:

$$W_{i,d,n, \text{ modified}} = W_{i,d,n} \times \Delta_n [7],$$

Изменение корректировочных весов $W_{i,d,n}$ любым возможным вариантом направлено на уменьшение ошибки обучения для каждого нейрона 116 путём приближения его выходной суммы \sum_n к значению желаемого выходного сигнала. Таким образом, ошибка обучения для данного образа может уменьшаться до тех пор, пока не станет равной нулю или близкой к нулю.

Пример изменения корректировочных весов $W_{i,d,n}$ во время обучения показан на Рисунке 11. Значения корректировочных весов $W_{i,d,n}$ устанавливаются до начала обучения путём случайного распределения весов с весовыми значениями, установленным на $0 \pm 10\%$ от диапазона корректировочных весов, окончательное распределение веса достигается после обучения. Описанное вычисление общих сигналов проводится для всех нейронов 116 в п-сети 100. Описанная процедура обучения одного обучающего образа может повторяться для всех других обучающих образов. Такая процедура может привести к появлению ошибок обучения для некоторых из образов, которым сеть была обучена ранее, так как некоторые корректировочные веса $W_{i,d,n}$ могут участвовать в нескольких образах. Соответственно, обучение с другим образом может частично нарушить распределение корректировочных весов $W_{i,d,n}$, сформированных для предыдущих образов. Тем не менее, из-за того, что каждый синапс 118 включает в себя набор корректировочных весов $W_{i,d,n}$, обучение с новыми

образами, несмотря на возможное увеличение ошибки обучения, не приводит к удалению образов, которым п-сеть 100 была ранее обучена. Более того, чем больше синапсов 118 вносит вклад в каждый нейрон 116 и чем больше количество корректировочных весов $W_{i,d,n}$ на каждом синапсе, тем меньше обучение конкретному образу влияет на обучение другим образам.

Каждая обучающая эпоха обычно заканчивается с существенным уменьшением общей ошибки обучения и/или локальных ошибок обучения для всех обучающих образов. Ошибки могут быть оценены с помощью известных статистических методов, таких как, например, Средняя Квадратичная Ошибка (СКО), Средняя Абсолютная Ошибка (САО) или Средняя Стандартная Ошибка (ССО). Если общая ошибка или некоторые из локальных ошибок слишком высоки, могут проводиться дополнительные обучающие эпохи до тех пор пока ошибка не будет снижена до значения меньшего, чем заранее заданное значение ошибки. Описанный ранее процесс распознавания образов с определением процента сходства между различными образами, использованными для обучения (как показано на Рисунке 8) сам по себе является процессом классификации образов по ранее определённым категориям.

Для кластеризации, то есть, разделения образов на естественные классы или группы, которые не были заранее указаны, основной алгоритм обучения способа 200 можно изменить с помощью изменённого подхода Самоорганизующихся Карт Кохонена (СК). Желаемый выходной образ 126, соответствующий данному входному образу может быть сформирован непосредственно в процессе обучения п-сети 100 на основе набора «нейронов – победителей» с максимальными значениями выходной нейронной суммы 120. Рисунок 22 показывает, как использование основного алгоритма способа 200 может генерировать первичный набор выходных нейронных сумм, при этом набор далее преобразуется таким образом, что несколько наибольших

сумм сохраняют своё значение или увеличиваются, в то время как все другие суммы сводятся к нулю. Этот модифицированный набор выходных нейронных сумм – «победителей» может быть принят в качестве желаемого выходного образа 126.

Сформированный, как описано выше, набор желаемых выходных образов 126 включает в себя кластеры или группы. Таким образом набор желаемых выходных образов 126 допускает кластеризацию линейно неразделимых образов, в отличие от классической сети 10. На рисунке 13 показано, как описанный подход может помочь с кластеризацией сложных абстрактных образов "кошка-машина", где разные особенности образа присваиваются разным кластерам – кошкам и автомобилям. Набор желаемых выходных образов 126, созданных как описано выше, может быть использован, например, для создания различных классификаций, статистического анализа, выбора образов на основе критериев, сформированных в результате кластеризации. Кроме того, желаемые выходные образы 126, полученные п-сетью 100, могут использоваться в качестве входных образов для другой или дополнительной п-сети, которая также может быть сформирована по типу данной описанной п-сети 100. Сформированные таким образом желаемые выходные образы 126 могут быть использованы для последующего слоя многослойной п-сети.

Обучение классической нейронной сети 10 обычно осуществляется способом обучения с учителем, который основан на предварительно подготовленных парах, состоящих из входного образа и желаемого выходного образа. Тот же общий способ также используется для обучения п-сети 100, однако увеличенная скорость обучения п-сети 100 допускает также возможность обучения с «внешним учителем». Роль внешнего учителя может выполняться, например, человеком или компьютерной программой. Действуя в качестве внешнего учителя, человек может участвовать в выполнении

определенной физической задачи или действовать в игровой среде. П-сеть 100 принимает входные сигналы в виде данных о конкретной ситуации и её изменениях. Сигналы, отражающие действия учителя, могут вводиться в качестве желаемых выходных образов 126 и позволять п-сети 100 обучаться по базовому алгоритму. Таким образом, моделирование различных процессов может быть получено п-сетью 100 в режиме реального времени.

Например, п-сеть 100 можно обучить управлять транспортным средством путём получения информации о дорожных условиях и действиях водителя. С помощью моделирования большого количества разнообразных критических ситуаций, та же п-сеть 100 может быть обучена многими разными водителями и может накопить больше навыков вождения, чем обычно может накопить любой отдельный водитель. П-сеть 100 способна оценить определённое дорожное условие в 0,1 секунды или быстрее и аккумулировать значительный "опыт вождения", который может повысить безопасность дорожного движения в различных ситуациях. П-сеть 100 также может быть обучена взаимодействию с компьютером, например, с компьютером для игры в шахматы. Способность п-сети 100 легко переходить от режима обучения в режим распознавания и наоборот позволяет реализовать режим "обучения на ошибках", в случае обучения п-сети 100 с внешним учителем. В таком случае, частично обученная п-сеть 100 может производить свои собственные действия, например, управлять технологическим процессом. Учитель может контролировать действия п-сети 100 и корректировать эти действия в случае необходимости. Таким образом может обеспечиваться дополнительное обучение п-сети 100.

Информационная ёмкость п-сети 100 очень большая, но не безграничная. С установленными параметрами п-сети 100, такими как количество входов, выходов и интервалов, и с увеличением количества образов, которым обучается п-сеть, после определённого количества образов величина ошибок обучения также может увеличиться. Когда обнаруживается такое увеличение

получаемых ошибок, количество и/или величина ошибки могут быть снижены за счёт увеличения размера п-сети 100, так как п-сеть позволяет увеличивать количество нейронов 116 и/или количество интервалов "d" на п-сети или в её компонентах между обучающими эпохами. Расширение п-сети 100 может обеспечиваться за счёт добавления новых нейронов 116, добавления новых входов 102 и синапсов 118, изменения распределения коэффициентов воздействия корректировочных весов $C_{i,d,n}$ и деления существующих интервалов "d".

В большинстве случаев п-сеть 100 будет обучаться, чтобы обеспечить её способность распознавать образы, паттерны и корреляции, присущие образу или набору образов. Процесс распознавания в простейшем случае повторяет первые этапы процесса обучения в соответствии с основным алгоритмом, описанным как часть способа 200. В частности:

- прямое распознавание начинается с форматирования образа в соответствии с теми же правилами, которые используются для форматирования образов для обучения;
- образ передаётся на входы обученной п-сети 100, дистрибьюторы назначают корректировочные веса $W_{i,d,n}$, соответствующие значениям входных сигналов, которые были установлены в процессе обучения, а нейроны генерируют соответствующие нейронные суммы, как показано на Рисунке 8;
- если полученные выходные суммы, представляющие выходной образ 126, полностью соответствуют одному из образов, которым п-сеть 100 обучалась, происходит точное распознавание объекта; и
- если выходной образ 126 частично соответствует нескольким образам, на которых обучалась п-сеть 100, результат показывает степень сходства с разными образами в процентах. Рисунок 13

показывает, что в процессе распознавания сложного образа, который сделан на основе комбинации образов кошки и автомобиля, выходной образ 126 представляет собой комбинацию данного образа и показывает процент вклада каждого исходного образа в комбинацию.

Например, если для обучения были использованы несколько фотографий определённого человека, распознанное изображение может соответствовать 90% первого изображения, 60% второго изображения и 35% третьего изображения. Может быть такое, что распознанный образ будет соответствовать с определённой вероятностью изображениям других людей или даже животных, это означает, что между изображениями есть некоторое сходство. Тем не менее, вероятность такого сходства, скорее всего будет не велика. На основании таких вероятностей может быть определена достоверность распознавания, например, на основе теоремы Байеса.

С помощью п-сети 100 также возможно реализовать многоступенчатое распознавание, которое сочетает в себе преимущества алгоритмических и нейросетевых способов распознавания. Такое многоступенчатое распознавание может включать:

- первоначальное распознавание образа с помощью предварительно обученной сети посредством использования не всех, а только 1% - 10% входов, которые будут здесь обозначены как "основные входы". Такая часть входов может быть распределена в п-сети 100 либо равномерно, либо случайным образом, либо с помощью какой-либо другой функции распределения. Например, так может быть осуществлено точное распознавание человека на фотографии, которая содержит множество других объектов;
- выбор наиболее информативных объектов или частей объектов для дальнейшего детального распознавания. Такой выбор может быть

проведён в соответствии со структурами определённых объектов, которые предварительно установлены в памяти, как в алгоритмическом способе, или в соответствии с градиентом цвета, яркостью и/или глубиной изображения. Например, при распознавании портретов могут быть выбраны следующие зоны распознавания: глаза, уголки рта, форма носа, а также некоторые специфические особенности, таких как татуировки, номера транспортных средств или номера домов тоже могут быть выбраны и распознаны, используя аналогичный подход; и

- детальное распознавание выбранных образов при необходимости также возможно.

Формирование компьютерной эмуляции п-сети 100 и её обучение может быть проведено на основе изложенного выше описания, при использовании любого языка программирования. Например, может быть использовано объектно-ориентированное программирование, в котором синаптические веса 108, корректировочные веса 112, дистрибуторы 114 и нейроны 116 представляют объекты программирования или классы объектов, отношения между классами объектов устанавливаются посредством ссылок и/или сообщений, а алгоритмы взаимодействия устанавливаются между объектами и между классами объектов.

Формирование и обучение программной эмуляции п-сети 100 может включать в себя следующее:

Подготовку к формированию и обучению п-сети 100, в частности:

- преобразование наборов обучающих входных образов в цифровую форму в соответствии с данной задачей;
- анализ полученных цифровых образов, в том числе выбор параметров входных сигналов, которые будут использоваться для обучения, например, частот, величин, фаз или координат; а также

- установка диапазона для обучающих сигналов, количества интервалов в пределах этого диапазона и распределения коэффициентов воздействия корректировочного веса $C_{i,d,n}$.

2. Формирование программной эмуляции п-сети, включая:

- формирование набора входов для п-сети 100. Например, количество входов может быть равно количеству сигналов во соответствующем обучающем образе;
- формирование набора нейронов, где каждый нейрон представляет собой суммирующее устройство;
- формирование набора синапсов с синаптическими весами, где каждый синапс соединён с одним входом п-сети и одним нейроном;
- формирование блоков коррекции веса в каждом синапсе, где блоки коррекции веса включают дистрибьюторы и корректировочные веса, и где каждый корректировочный вес имеет следующие характеристики:
 - Входной индекс корректировочного веса (i);
 - Нейронный индекс корректировочного веса (n);
 - Индекс интервала корректировочного веса (d); и
 - Начальное значение корректировочного веса ($W_{i,d,n}$).
 - назначение корреляции между интервалами и корректировочными весами.

3. Обучение каждого нейрона одним входным образом, включая:

- назначение коэффициентов воздействия корректировочных весов $C_{i,d,n}$, включая:
 - определение интервала, соответствующего входному сигналу входного обучающего образа, принимаемого каждым входом; и

- назначение величины коэффициентов воздействия корректировочных весов $C_{i,d,n}$ всем корректировочным весам на всех синапсах.
- Вычисление выходной нейронной суммы (\sum_n) для каждого нейрона “n”, путём сложения значения корректировочного веса $W_{i,d,n}$ всех синаптических весов, вносящих вклад в нейрон, помноженный на соответствующие коэффициенты воздействия корректировочных весов $C_{i,d,n}$:

$$\sum_n = \sum_{i,d,n} W_{i,d,n} \times C_{i,d,n}$$

- Вычисление отклонения или ошибки обучения (T_n) посредством вычитания выходной нейронной суммы \sum_n из соответствующего желаемого выходного сигнала O_n :

$$T_n = O_n - \sum_n;$$

- Вычисление равного корректирующего значения (Δ_n) для всех корректировочных весов, вносящих вклад в нейрон “n” посредством деления ошибки обучения на количество синапсов “S”, соединённых с нейроном “n”:

$$\Delta_n = T_n / S$$

- Изменение всех корректировочных весов $W_{i,d,n}$, вносящих вклад в соответствующий нейрон, путём прибавления к каждому корректировочному весу корректирующего значения Δ_n , делённого

на соответствующие коэффициенты воздействия корректировочных весов $C_{i,d,n}$:

$$W_{i,d,n} \text{ изменённые} = W_{i,n,d} + \Delta_n / C_{i,d,n}.$$

Другой способ вычисления равного корректирующего значения (Δ_n) и изменения корректировочных весов $W_{i,d,n}$ для всех корректировочных весов, вносящих вклад в нейрон “n”, может включать следующее:

- Деление сигнала желаемого выходного образа O_n на выходную нейронную сумму \sum_n :

$$\Delta_n = O_n / \sum_n$$

- Изменение корректировочных весов $W_{i,n,d}$, вносящих вклад в нейрон, путём умножения корректировочных весов на корректирующее значение Δ_n :

$$W_{i,d,n} \text{ изменённые} = W_{i,d,n} \times \Delta_n$$

4. Обучение n-сети 100, используя все обучающие образы, включая:

- повторение описанного выше процесса для всех выбранных обучающих образов, которые включены в одну учебную эпоху; и
- определение ошибки или ошибок определённой эпохи обучения, путём сравнения этой ошибки (этих ошибок) с заранее установленным приемлемым уровнем ошибки и повторение обучающих эпох до тех пор, пока ошибки обучения не станут меньше заранее заданного допустимого уровня ошибки.

Реальный пример программной эмуляции п-сети 100 с использованием объектно-ориентированного программирования описывается ниже и показан на Рисунках 14-21.

Формирование класса объектов Нейронного Блока может включать формирование:

- набора объектов класса синапсов;
- нейрона 116, представляющего переменную величину, в котором в процессе обучения выполняется прибавление ; и
- калькулятора 122, представляющего набор переменных величин, в котором хранятся значения желаемых нейронных сумм 120 и в процессе обучения выполняется вычисление корректирующих значений Δ_n .

Класс Нейронного Блока, выполняющего обучение п-сети 100, может включать:

- формирование нейронных сумм 120;
- установку желаемых сумм;
- вычисление корректирующего значения Δ_n ; и
- прибавление вычисленного корректирующего значения Δ_n к корректировочным весам $W_{i,n,d}$.

Формирование объектов класса Синапсов может включать:

- набор корректировочных весов $W_{i,n,d}$; и
- указатель, обозначающий вход, соединённый с синапсом 118.

Класс Синапсов может выполнять следующие функции:

- инициализацию корректировочных весов $W_{i,n,d}$;
- умножение этих весов $W_{i,n,d}$ на коэффициенты $C_{i,d,n}$; и
- коррекцию данных весов $W_{i,n,d}$.

Формирование класса объектов Входного Сигнала может включать:

- набор индексов на синапсах 118, соединённых с данным входом 102;
- переменную величину, которая включает значение входного сигнала 104;
- значения возможного минимума и максимума входного сигнала;
- количество интервалов “d”; и
- длину интервала.

Класс Входного Сигнала может выполнять следующие функции:

- формирование структуры п-сети 100, включая:
 - Добавление и удаление связей между входом 102 и синапсами 118; и
 - Установку количества интервалов “d” для синапсов 118 конкретного входа 102.
- установку параметров минимального и максимального входного сигнала 104;
- вклад в функционирование п-сети 100:
 - установка входного сигнала 104; и
 - установка коэффициентов воздействия корректировочных весов $C_{i,d,n}$.

Формирование класса объектов П-сети включает набор объектных классов:

- Нейронный Блок; и
- Входной Сигнал.

Класс П-сети выполняет следующие функции:

- установка количества объектов класса Входного Сигнала;
- установка количества объектов класса Нейронного Блока; и

- групповой запрос функций объектов Нейронного Блока и Входного Сигнала.

Во время процесса обучения могут быть сформированы циклы, где:

- нейронная выходная сумма, равная нулю, формируется до начала цикла;
- все синапсы, вносящие вклад в данный Нейронный Блок, корректируются. Для каждого синапса 118:
 - На основании входного сигнала 102 дистрибьютор формирует набор коэффициентов воздействия корректировочных весов $C_{i,d,n}$;
 - Все веса $W_{i,n,d}$ указанного синапса 118 корректируются, и для каждого веса:
 - Значение веса $W_{i,n,d}$ умножается на соответствующий коэффициент воздействия корректировочного веса $C_{i,d,n}$;
 - Результат умножения прибавляется к формирующейся выходной нейронной сумме;
 - вычисляется корректирующее значение Δ_n ;
 - корректирующее значение Δ_n делится на коэффициент воздействия корректировочного веса $C_{i,d,n}$, то есть $\Delta_n / C_{i,d,n}$; и
 - все синапсы 118, вносящие вклад в данный Нейронный Блок, корректируются. Для каждого синапса 118 все веса $W_{i,n,d}$ данного синапса корректируются, и для каждого веса его значение изменяется на соответствующее корректирующее значение Δ_n .

Ранее обозначенная возможность дополнительного обучения п-сети 100 позволяет сочетать обучение с распознаванием образов, что способствует ускорению процесса обучения и улучшению его точности. При обучении п-сети 100 на наборе последовательно меняющихся образов, как, например,

обучение на последовательных кадрах фильма, которые немного отличаются друг от друга, дополнительное обучение может включать:

- обучение с первым образом;
- распознавание следующего образа и установление процента сходства между новым образом и образом, на котором сеть обучалась изначально. Дополнительное обучение не требуется, если ошибка распознавания меньше её заранее заданного значения;
- и
- если ошибка обучения превышает заранее заданное значение, проводится дополнительное обучение.

Обучение p -сети 100 посредством приведённого выше основного обучающего алгоритма является эффективным для решения задач распознавания образов, но не исключает потери или искажения данных из-за перекрытия образов. Таким образом, использование p -сети 100 для целей памяти, хотя и возможно, но может быть не совсем надёжным. Настоящий вариант осуществления описывает обучение p -сети 100, которая обеспечивает защиту от потери или искажения информации. В основной обучающий алгоритм сети может быть введено дополнительное ограничение, которое требует, чтобы каждый корректировочный вес $W_{i,n,d}$ мог быть обучен только один раз. После первого обучающего цикла значение веса $W_{i,n,d}$ остается зафиксированным или постоянным. Этого можно добиться путём ввода дополнительного индекса доступа "а" для каждого корректировочного веса, который является описанным выше индексом, представляющим количество доступов к данному корректировочному весу $W_{i,n,d}$ во время процесса обучения.

Как описано выше, каждый корректировочный вес может принять номенклатуру $W_{i,n,d,a}$, где "а" – количество доступов к данному весу во время

процесса обучения. В простейшем случае, для неизменённого, то есть незафиксированного веса, $a = 0$, в то время как для весов, которые были изменены или зафиксированы с помощью описанного основного алгоритма, $a = 1$. Более того, при применении основного алгоритма, корректировочные веса $W_{i,n,d,a}$ с фиксированным значением $a = 1$ могут быть исключены из весов, на которых делаются изменения. В таком случае, уравнения [5], [6] и [7] можно преобразовать следующим образом:

Значение	Основной алгоритм	Обучающий алгоритм с зафиксированными весами
Равное корректирующее значение - Вариант 1	$\Delta_n = (O_n - \sum_n) / S$ [4],	$\Delta_n = (O_n - \sum_n) / S_0$ [8], где S_0 – сумма $C_{i,d,n,a}$ всех корректировочных весов $W_{i,n,d,a}$, вносящих вклад в данный нейрон и имеющих индекс $a = 0$
Изменённый корректировочный вес - Вариант 1	$W_{i,n,d}$ изменённый = $W_{i,n,d} + \Delta_n / C_{i,d,n}$ [5],	$W_{i,n,d,0}$ изменённый = $W_{i,n,d,0} + \Delta_n / C_{i,d,n,0}$ [9], где $W_{i,n,d,0}$ являются весами, вносящими вклад в данный нейрон и имеющими индекс $a = 0$, и $C_{i,d,n,0}$ являются коэффициентами воздействия корректировочных весов для корректировочных весов, вносящих вклад в данный нейрон и имеющих индекс $a = 0$
Изменённый корректировочный вес - Вариант 2	$W_{i,n,d}$ изменённый = $W_{i,n,d} \times \Delta_n$ [7]	$W_{i,n,d,0}$ изменённый = $W_{i,n,d,0} \times \Delta_n$ [10]

Ограничение, указанное выше, может быть частично применено к коррекции ранее обученных корректировочных весов $W_{i,n,d,a}$, но только тех весов, которые образуют наиболее важные образы. Например, в рамках обучения на наборе портретов одного человека, одно конкретное изображение может быть объявлено первичным и ему может быть назначен приоритет. После обучения на таких приоритетных образах все корректировочные веса $W_{i,n,d,a}$, которые изменились в процессе обучения, могут быть зафиксированы, то есть их индекс $a = 1$, таким образом обозначая вес как $W_{i,n,d,1}$, и другие изображения того же человека могут оставаться изменяемыми. Такой приоритет может включать другие образы, например те, которые используются в качестве ключей шифрования и/или содержат важные числовые данные.

Изменения корректировочных весов $W_{i,n,d,a}$ могут также быть не полностью запрещены, а ограничены ростом индекса "a". То есть, каждое последующее использование веса $W_{i,n,d,a}$ может быть использовано для уменьшения его способности изменяться. Чем чаще определённый корректировочный вес $W_{i,n,d,a}$ используется, тем меньше вес изменяется с каждым доступом, и, таким образом, в процессе обучения на последующих образах, предыдущие, сохранённые образы изменяются меньше и подвергаются меньшим искажениям. Например, если $a = 0$, то любое изменение в весе $W_{i,n,d,a}$ возможно; при $a = 1$ возможность изменения веса может быть снижена до $\pm 50\%$ от значения веса; при $a = 2$ возможность изменения может быть снижена до $\pm 25\%$ от значения веса.

После достижения заранее определённого количества доступов, как указано индексом "a", например, при $a = 5$, дальнейшее изменение веса $W_{i,n,d,a}$ может быть запрещено. Такой подход может обеспечить сочетание высокого интеллекта и информационной безопасности в пределах одной п-сети 100. С помощью сетевого механизма расчета ошибки, уровни допустимых ошибок могут быть установлены таким образом, что информация с потерями в

пределах заранее определённого диапазона точности может быть сохранена, при этом диапазон точности может быть назначен в соответствии с конкретной задачей. Другими словами, для п-сети 100, работающей с визуальными образами, ошибка может быть установлена на уровне, который не может быть обнаружен невооруженным глазом, что обеспечит существенный «запас прочности» и увеличение ёмкости памяти. Всё вышперечисленное может сделать возможным обеспечение высокоэффективного хранения визуальной информации, например, фильмов.

- Возможность выборочно очищать память компьютера может быть ценной для дальнейшего функционирования п-сети 100 на высоком уровне. Такая выборочная очистка памяти может быть сделана путём удаления некоторых образов без потери или искажения остальной части хранящейся информации. Такая очистка может быть проведена следующим образом:
- Определение всех корректировочных весов $W_{i,n,d,a}$, которые участвуют в формировании образа, например, введением образа в сеть или составлением списка корректировочных весов, использованных для каждого образа;
- Сокращение индекса “а” для соответствующих корректировочных весов $W_{i,n,d,a}$; и
- Замена корректировочных весов $W_{i,n,d,a}$ нулём или случайным значением, близким к среднему значению диапазона возможных значений для данного веса, когда индекс “а” сокращён до нуля.

Надлежащий порядок и последовательность сокращения индекса "а" могут быть экспериментально выбраны, чтобы определить сильные паттерны, скрытые в последовательности образов. Например, для каждых 100 образов, вводимых в п-сеть 100 в процессе обучения, индекс "а" может снижаться на

единицу, пока он не достигнет нулевого значения. В таком случае значение "а" может расти соответственно с введением новых образов. Состязание между ростом и сокращением "а" может привести к ситуации, когда случайные изменения постепенно удаляются из памяти, в то время как корректировочные веса $W_{i,n,d,a}$, которые использовались и подтвердились много раз, могут быть сохранены. Когда п-сеть 100 обучается на большом количестве образов с одинаковыми атрибутами, например, одинаковыми предметами или аналогичной средой, часто используемые корректировочные веса $W_{i,n,d,a}$ постоянно подтверждают своё значение и информация в этих областях становится очень стабильной. Кроме того, случайные помехи будут постепенно исчезать. Другими словами, п-сеть 100 с постепенным уменьшением индекса "а" может служить в качестве эффективного фильтра помех.

Описанные варианты осуществления п-сети 100, обучающейся без потери информации, позволяют создать память п-сети с большой ёмкостью и надёжностью. Такая память может использоваться в качестве высокоскоростной компьютерной памяти большой ёмкости, обеспечивающей скорость бóльшую, даже чем система "кэш-памяти", при этом подобная память не увеличит стоимость и сложность компьютера, как это обычно бывает с системой "кэш-памяти". Согласно опубликованным данным, во время записи фильма с помощью нейронных сетей, память может быть сжата в десятки или сотни раз без существенной потери качества записи. Другими словами, нейронная сеть способна работать как очень эффективная программа архивирования (компрессии данных). Комбинируя эту способность нейронных сетей с высокой скоростью обучения п-сети 100, можно обеспечить создание системы высокоскоростной передачи данных, памяти с высокой ёмкостью, а также обеспечить высокоскоростное дешифрование программных

мультимедийных файлов, т.е. реализовать высокоэффективные кодеки (устройства кодирования – декодирования) .

В п-сети 100 данные хранятся в виде набора корректировочных весов $W_{i,n,d,a}$, что представляет собой кодированную запись. Это делает маловероятным или просто невозможным декодирование записей или несанкционированный доступ к п-сети без использования идентичной сети и ключа, посредством существующих хакерских способов. Таким образом, п-сеть 100 может обеспечить высокую степень защиты данных. Кроме того, в отличие от обычной компьютерной памяти, повреждение отдельных элементов памяти п-сети 100 представляет незначительный негативный эффект, так как другие элементы в значительной степени компенсируют утраченные функции. В процессе распознавания образов, внутренние паттерны использованного образа практически не искажаются в результате повреждения одного или нескольких элементов. Всё вышеперечисленное может значительно повысить надёжность компьютеров и позволить использовать блоки памяти, которые при обычных условиях считались бы неисправными. Кроме того, этот тип памяти менее уязвим для атак хакеров из-за отсутствия постоянного адреса (ов) для критических байтов в п-сети 100, что делает её непроницаемой для атаки с помощью различных компьютерных вирусов.

Ранее обозначенный процесс распознавания образов с определением процентного сходства между разными образами, используемыми при обучении, может также быть использован в качестве процесса классификации образов в соответствии с заранее определёнными категориями, как было отмечено выше. Для кластеризации, то есть разделения образов на заранее не определённые естественные классы или группы, основной процесс обучения может быть изменён. Данный вариант осуществления может включать:

- Подготовку набора входных образов для обучения, без добавления подготовленных выходных образов;

- Формирование и обучение сети с формированием выходных нейронных сумм, как это происходит в соответствии с основным алгоритмом;
- Выбор из полученных выходных образов тех, у которых максимальная выходная сумма, то есть выбор «выхода – победителя», или группы «выходов – победителей», которые могут быть организованы по аналогии с сетью Кохонена;
- создание желаемого выходного образа, в котором «выход – победитель» или группа «выходов – победителей» получают максимальные значения. В то же время:
 - Количество выбранных победивших выходов может быть определено заранее, например, в диапазоне от 1 до 10, или победившие выходы могут быть выбран в соответствии с правилом "не менее чем N% от максимальной нейронной суммы", где "N", может быть, например, в пределах 90 - 100%; и
 - Все остальные выходы могут быть приведены к нулю.
- Обучение в соответствии с основным алгоритмом, используя созданные желаемые выходные образы, Рис 13; и
- повторение всех процедур для других образов с формированием для каждого образа различных победителей или групп-победителей.

Набор выходных желаемых образов, сформированных вышеуказанным способом, может быть использован для описания кластеров или групп, в которых множество входных образов может естественным образом разделиться. Такой набор желаемых выходных образов может быть использован для проведения различных классификаций, например, для выбора образов в соответствии с установленными критериями, и в статистическом

анализе. Вышесказанное также может быть использовано для вышеупомянутой инверсии входных и выходных образов. Другими словами, сформированные выходные образы п-сети могут быть использованы в качестве входных образов для другой, то есть дополнительной сети или слоя многослойной сети, а выходом дополнительной сети могут быть образы, представленные в любой форме, подходящей для ввода в компьютер.

В п-сети 100 после одного цикла обучения с описанным выше алгоритмом желаемые выходные образы могут быть получены с небольшим варьированием (разбросом точек) выходной суммы, что может замедлить процесс обучения, а также может уменьшить его точность. Для улучшения обучения п-сети 100 начальное варьирование точек может быть искусственно увеличено или расширено, так что варьирование величины точек будет охватывать весь диапазон возможных выходных значений, например, от -50 до +50, как показано на Рис 21. Такое расширение исходного варьирования точек может быть как линейным так и нелинейным.

Ситуация может сложиться так, что максимальное значение определённого выхода окажется исключением или ошибкой, например, из-за проявления помех. Такое может проявляться в появлении максимального значения, окружённого множеством малых сигналов. При выборе выходов - победителей, малыми значениями сигнала можно пренебречь, путём выбора в качестве победителей наибольших сигналов, окружённых другими достаточно большими сигналами. Для этой цели могут быть использованы известные статистические способы уменьшения дисперсии, такие как выборка по значимости. Такой подход может обеспечить удаление помех, сохраняя при этом основные ценные паттерны. Создание групп-победителей обеспечивает кластеризацию линейно неразделимых образов, т.е. образов, которые относятся к более чем одному кластеру, как показано на Рисунке 13. Вышесказанное

может обеспечить значительное повышение точности и уменьшить количество ошибок при кластеризации.

В процессе обучения п-сети 100 типичными ошибками, подвергающимися коррекции, являются:

	Типичная ошибка нейронной сети	Способ коррекции п-сети 100
	Ошибки при выборе обучающих образов. Например, набор изображений людей включает изображение кошки	Удаление соответствующего желаемого выходного образа или введение ограничения на его демонстрацию
	Ошибки сети, которые не были исправлены в процессе обучения. Например, определённый образ распознан неверно из-за того, что сеть не может разделить некоторые характеристики объекта (эффект линейной неотделимости).	Дополнительное обучение п-сети 100 после обнаружения ошибки; введение дополнительных желаемых выходных образов
	Снижение точности из-за достижения предела информационной ёмкости сети	Расширение п-сети 100

Исправление ошибок также возможно при помощи вышеописанного алгоритма обучения с внешним учителем.

Подробное описание и рисунки обеспечивают и подкрепляют раскрытие изобретения, но объём раскрытия определяется исключительно формулой изобретения. Хотя некоторые из лучших образов действия и других вариантов осуществления заявленного раскрытия были подробно описаны, существуют и другие различные альтернативные образцы и варианты для практического осуществления изобретения, определённого в прилагаемой

формуле изобретения. Кроме того, варианты осуществления, показанные на чертежах или характеристики различных вариантов осуществления, упомянутые в настоящем описании, не следует обязательно понимать как варианты осуществления, независимые друг от друга. Скорее возможно, что каждая из характеристик, описанных в одном из примеров варианта осуществления может быть объединена с одной или несколькими другими желаемыми характеристиками из других вариантов осуществления, что может привести к другим вариантам осуществления, не описанным в тексте или на чертежах. Соответственно, такие другие варианты осуществления попадают в рамки объема прилагаемой формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Нейронная сеть, содержащая:

множество входов нейронной сети, каждый вход сконфигурирован для приёма входного сигнала, имеющего входное значение;

множество синапсов, в котором каждый синапс соединён с одним из множества входов и включает в себя множество корректировочных весов, где каждый корректировочный вес определяется весовым значением;

набор дистрибьюторов, в котором каждый дистрибьютор операционно соединён с одним из множества входов для приёма соответствующего входного сигнала и сконфигурирован для выбора одного или более корректировочных весов из множества корректировочных весов в соответствии с входным значением;

набор нейронов, в котором каждый нейрон имеет, по меньшей мере, один выход и соединён с, по меньшей мере, одним из множества входов посредством одного из множества синапсов, и в котором каждый нейрон сконфигурирован для того, чтобы суммировать весовые значения корректировочных весов, выбранных от каждого синапса, связанного с соответствующим нейроном, и тем самым создавать нейронную сумму; а также

калькулятор коррекции веса, сконфигурированный для приёма желаемого выходного сигнала, имеющего значение, для определения отклонения нейронной суммы от желаемого значения выходного сигнала и для изменения соответствующих значений корректировочных весов используя определённое отклонение, таким образом, что суммирование изменённых значений корректировочных весов для определения нейронной суммы минимизирует отклонение нейронной суммы от желаемого значения выходного сигнала, чтобы таким образом обучить нейронную сеть.

2. Нейронная сеть по пункту 1, в которой:

определение отклонения нейронной суммы от желаемого выходного сигнала включает деление желаемого значения выходного сигнала на нейронную сумму, чтобы таким образом получить коэффициент отклонения; а также

изменение значений соответствующих корректировочных весов включает умножение каждого корректировочного веса, используемого для получения нейронной суммы, на коэффициент отклонения.

3. Нейронная сеть по пункту 1, в которой отклонение нейронной суммы от желаемого выходного сигнала представляет собой математическую разность между ними, и в которой получение соответствующих изменённых корректировочных весов включает распределение математической разности между всеми корректировочными весами, используемыми для получения этой нейронной суммы.

4. Нейронная сеть по пункту 3, в которой распределение математической разности включает деление определённой разности поровну между всеми корректировочными весами, использованными для получения нейронной суммы.

5. Нейронная сеть по пункту 3, в которой:

каждый дистрибьютор дополнительно сконфигурирован для назначения множества коэффициентов воздействия соответствующему множеству корректировочных весов таким образом, что каждый коэффициент воздействия назначается одному из множества корректировочных весов в заранее определённой пропорции, чтобы получить соответствующую нейронную сумму;

каждый нейрон сконфигурирован так, чтобы суммировать произведения корректировочного веса и назначенного коэффициента воздействия для всех синапсов, связанных с ним; а также

калькулятор коррекции веса сконфигурирован так, чтобы назначать часть определённой разности каждому корректировочному весу,

используемому для получения нейронной суммы, согласно пропорции, установленной соответствующим коэффициентом воздействия.

6. Нейронная сеть по пункту 5, в которой:

каждое соответствующее множество коэффициентов воздействия определяется функцией распределения воздействия;

множество входных значений поступает в диапазон значений, разделённый на интервалы в соответствии с функцией распределения интервалов таким образом, что каждое входное значение поступает в соответствующий интервал, и каждый корректировочный вес соответствует одному из интервалов; а также

каждый дистрибьютор использует соответствующее полученное входное значение для выбора соответствующего интервала, и для назначения соответствующего множества коэффициентов воздействия корректировочному весу, соответствующему выбранному интервалу и, по меньшей мере, одному корректировочному весу, соответствующему интервалу, примыкающему к выбранному соответствующему интервалу.

7. Нейронная сеть по пункту 6, в которой каждый корректировочный вес дополнительно определяется набором индексов включающим:

входной индекс, сконфигурированный, чтобы идентифицировать корректировочный вес, соответствующий входу;

индекс интервала, сконфигурированный, чтобы определять выбранный интервал для соответствующего корректировочного веса; а также

нейронный индекс, сконфигурированный, чтобы определять корректировочный вес соответствующему нейрону.

8. Нейронная сеть по пункту 7, в которой каждый корректировочный вес дополнительно определяется с помощью индекса доступа, сконфигурированного, чтобы подсчитывать количество раз, когда

входной сигнал обращается к соответствующему корректировочному весу во время обучения нейронной сети.

9. Способ обучения нейронной сети, содержащий:

получение, через вход в нейронную сеть, входного сигнала, имеющего входное значение;

передача входного сигнала дистрибьютору, функционально соединённому со входом;

выбор, посредством дистрибьютора, в соответствии с входным значением, одного или нескольких корректировочных весов из множества корректировочных весов, в котором каждый корректировочный вес определяется значением веса, и расположен на синапсе, подключенном к входу;

суммирование значений выбранных корректировочных весов нейроном, соединённым с входом посредством синапса и имеющим, по меньшей мере, один выход, чтобы создать нейронную сумму;

получение калькулятором коррекции веса желаемого выходного сигнала, имеющего значение;

определение калькулятором коррекции веса отклонения нейронной суммы от желаемого значения выходного сигнала; а также

изменение калькулятором коррекции веса значений соответствующих корректировочных весов с помощью определённого отклонения, таким образом, что суммирование изменённых значений корректировочных весов для определения нейронной суммы минимизирует отклонение нейронной суммы от желаемого значения выходного сигнала, таким образом обучая нейронную сеть.

10. Способ по пункту 9, в котором:

вышеуказанное определение отклонения нейронной суммы от желаемого значения выходного сигнала включает деление желаемого значения выходного сигнала на нейронную сумму, чтобы таким образом получить коэффициент отклонения; а также

вышеуказанное изменение соответствующих корректировочных весов включает умножение каждого корректировочного веса, используемого для получения нейронной суммы, на коэффициент отклонения.

11. Способ по пункту 9, в котором вышеупомянутое определение отклонения нейронной суммы от желаемого значения выходного сигнала включает определение математической разности между ними, и в котором вышеуказанное изменение соответствующих корректировочных весов включает распределение математической разности между всеми корректировочными весами, используемыми для получения нейронной суммы.

12. Способ по пункту 11, в котором вышесказанное распределение математической разности включает деление определённой разности поровну между всеми корректировочными весами, использованным для получения нейронной суммы

13. Способ по пункту 9, дополнительно содержащий:
назначение, посредством дистрибьютора, множества коэффициентов воздействия множеству корректировочных весов, и включает назначение каждого коэффициента воздействия одному из множества корректировочных весов в заранее определённой пропорции, чтобы получить нейронную сумму;

суммирование, посредством нейрона, произведений корректировочного веса и назначенного коэффициента воздействия для каждого синапса, связанного с ним; а также

добавление, посредством калькулятора коррекции веса, части определённой разности каждому корректировочному весу, используемому для получения нейронной суммы согласно пропорции, установленной соответствующим коэффициентом воздействия.

14. Способ по пункту 13, в котором множество коэффициентов воздействия определяется функцией распределения воздействия; способ дополнительно содержащий:

получение входных значений в диапазоне значений, разделённом на интервалы в соответствии с функцией распределения интервалов таким образом, что каждое входное значение поступает в соответствующий интервал, и каждый корректировочный вес соответствует одному из интервалов; а также

использование, посредством дистрибьютора, полученного входного значения для выбора соответствующего интервала, и для назначения множества коэффициентов воздействия корректировочным весам, соответствующим выбранному интервалу и, по меньшей мере, одному корректировочному весу, соответствующему интервалу, примыкающему к выбранному соответствующему интервалу.

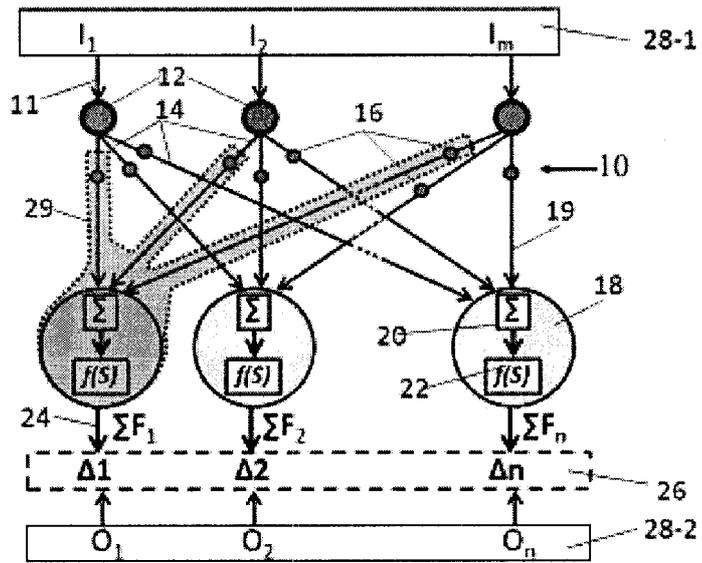
15. Способ по пункту 14, дополнительно содержащий дополнительное определение каждого корректировочного веса набором индексов, в котором набор индексов включает:

входной индекс, сконфигурированный, чтобы идентифицировать корректировочный вес, соответствующий входу;

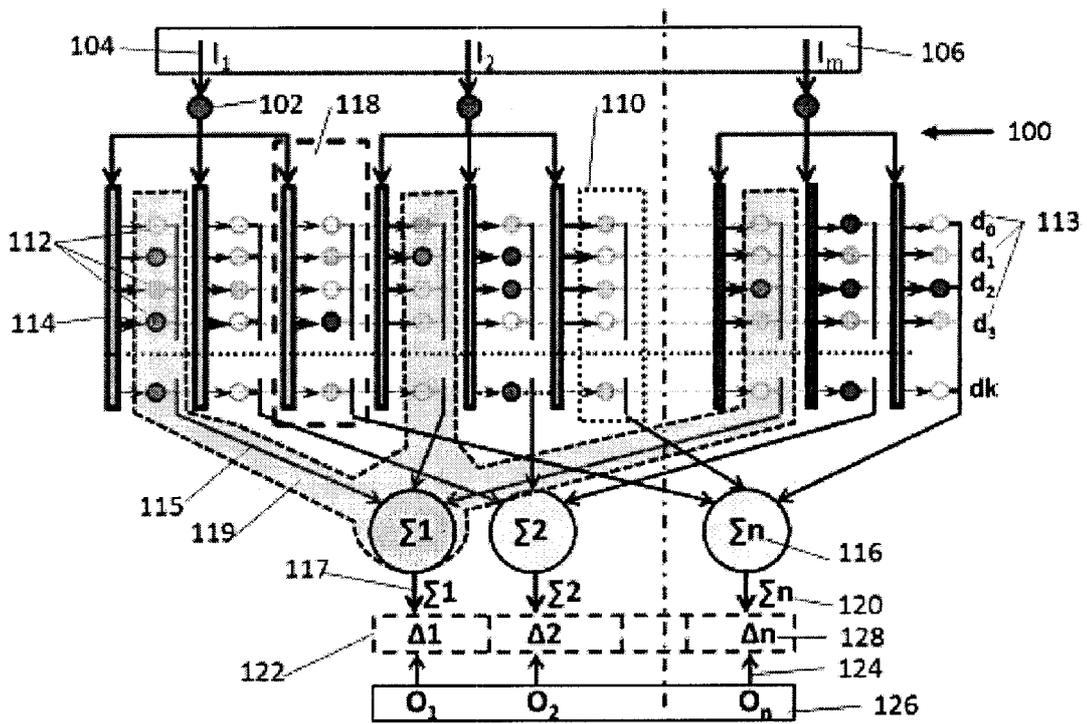
индекс интервала, сконфигурированный, чтобы определять выбранный интервал для соответствующего корректировочного веса; а также

нейронный индекс, сконфигурированный, чтобы определять корректировочный вес соответствующий нейрону.

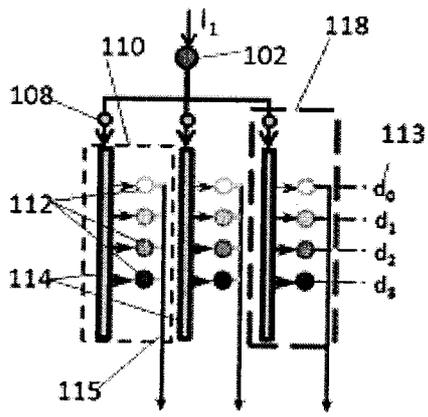
16. Способ по пункту 15, дополнительно содержащий дополнительное определение каждого корректировочного веса с помощью индекса доступа, сконфигурированного, чтобы подсчитывать количество раз, когда входной сигнал обращается к соответствующему корректировочному весу во время обучения нейронной сети.



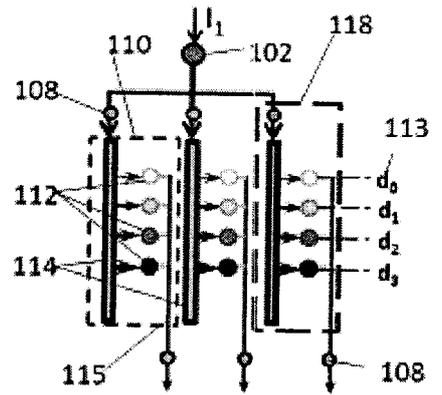
Фиг. 1 (Прототип)



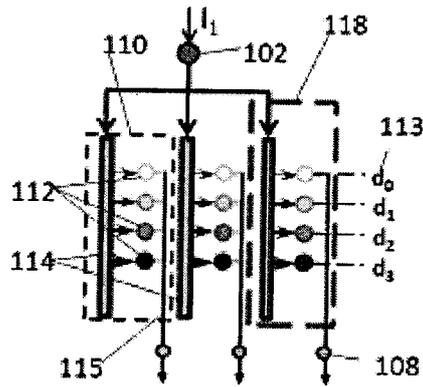
Фиг. 2



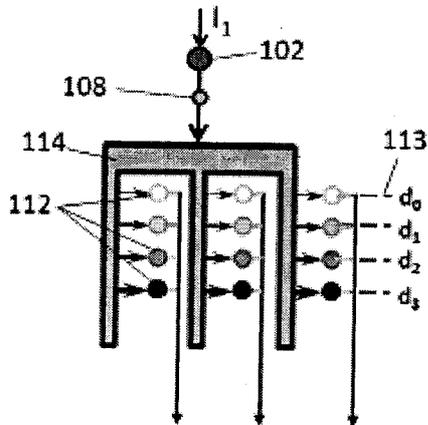
Фиг. 3А



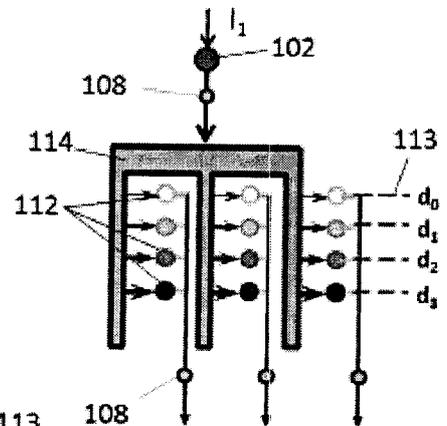
Фиг. 3С



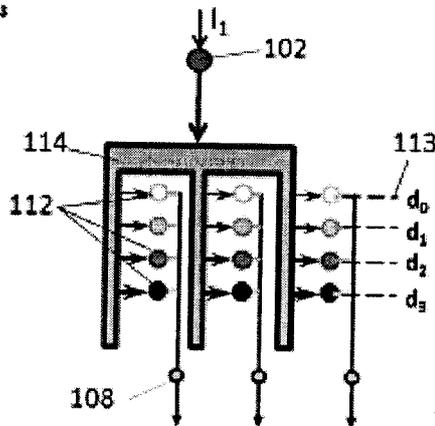
Фиг. 3В



Фиг. 4А



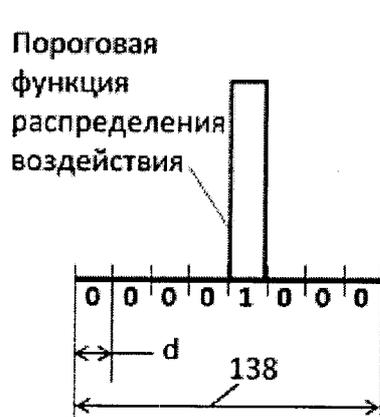
Фиг. 4С



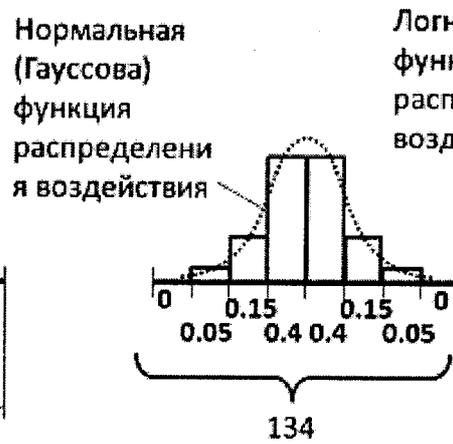
Фиг. 4В



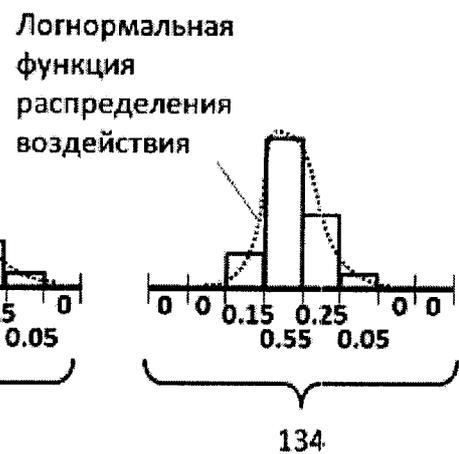
Фиг. 5



Фиг. 6А



Фиг. 6В



Фиг. 6С

Исходный образ

Образ в цифровом коде

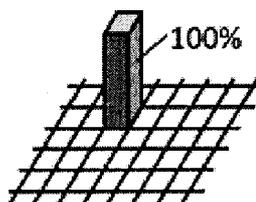
Образ в формате интервалов



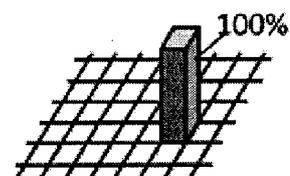
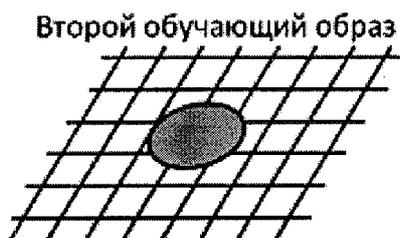
127	109	91	82
142	113	85	77
112	125	71	15
101	112	92	5

12	10	9	8
14	11	8	7
11	12	7	1
10	11	9	0

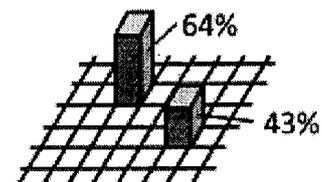
Фиг. 7



Результат после обучения на первом обучающем образе

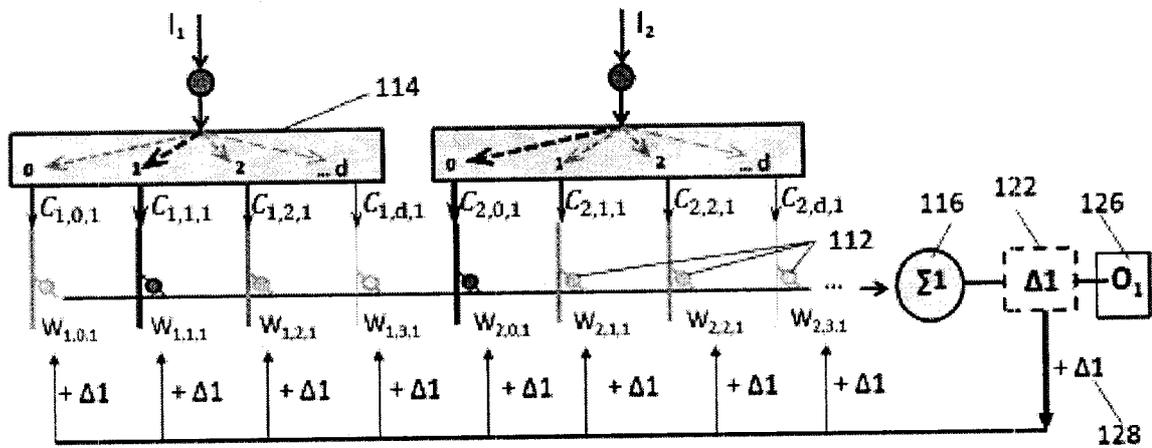
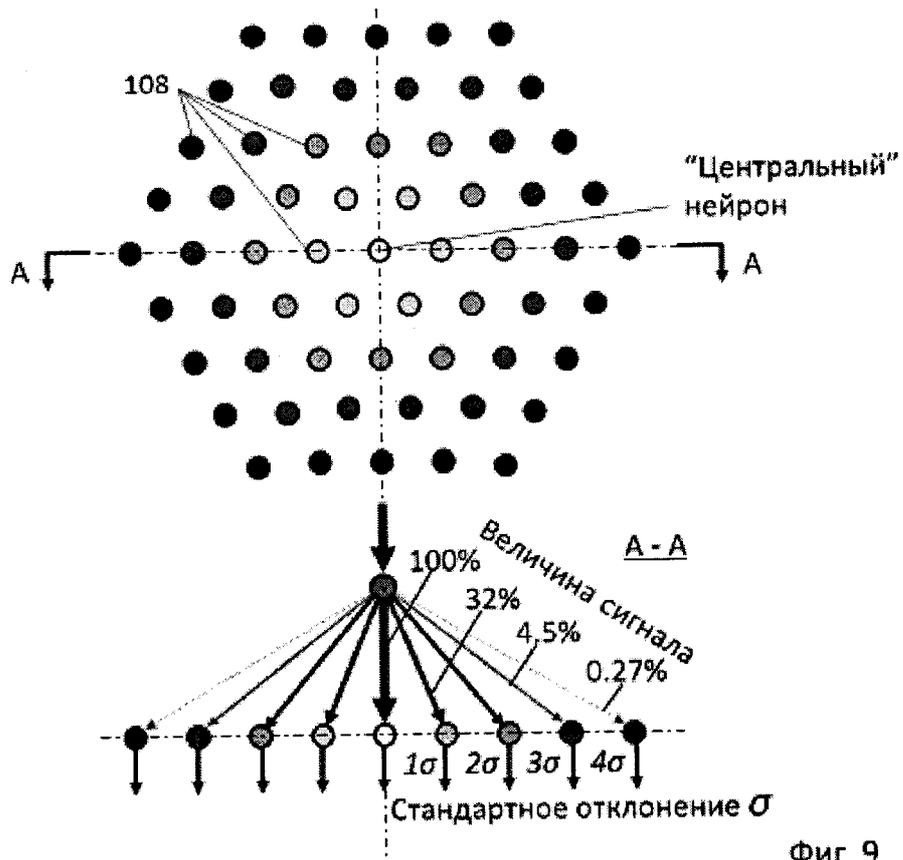


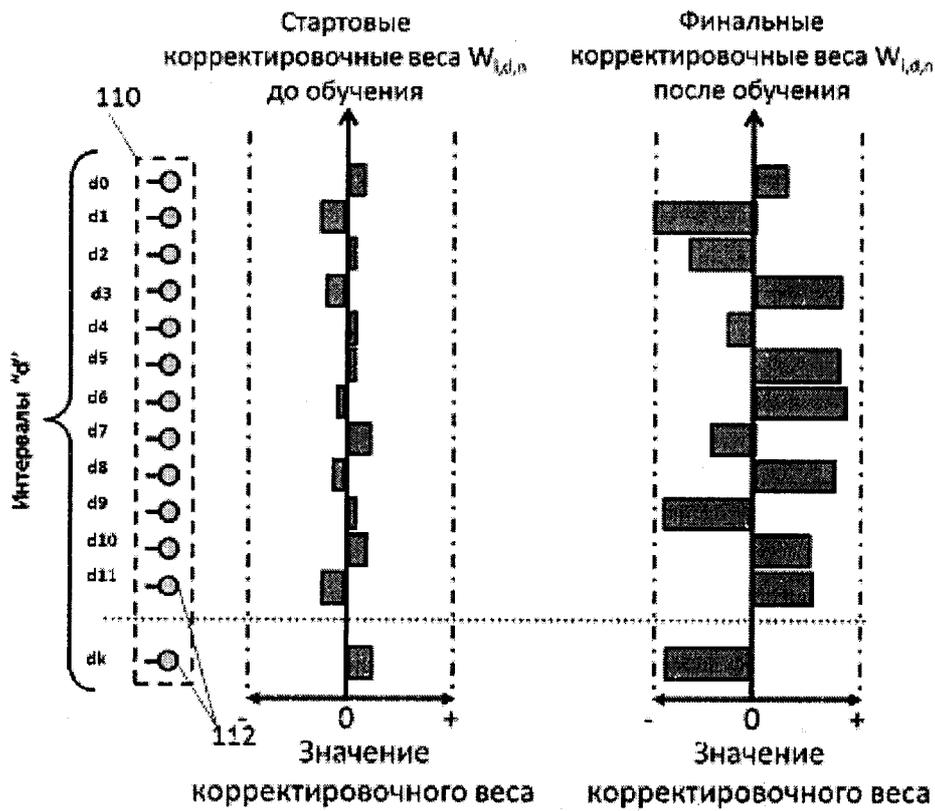
Результат после обучения на втором обучающем образе



Распознавание образа

Фиг. 8

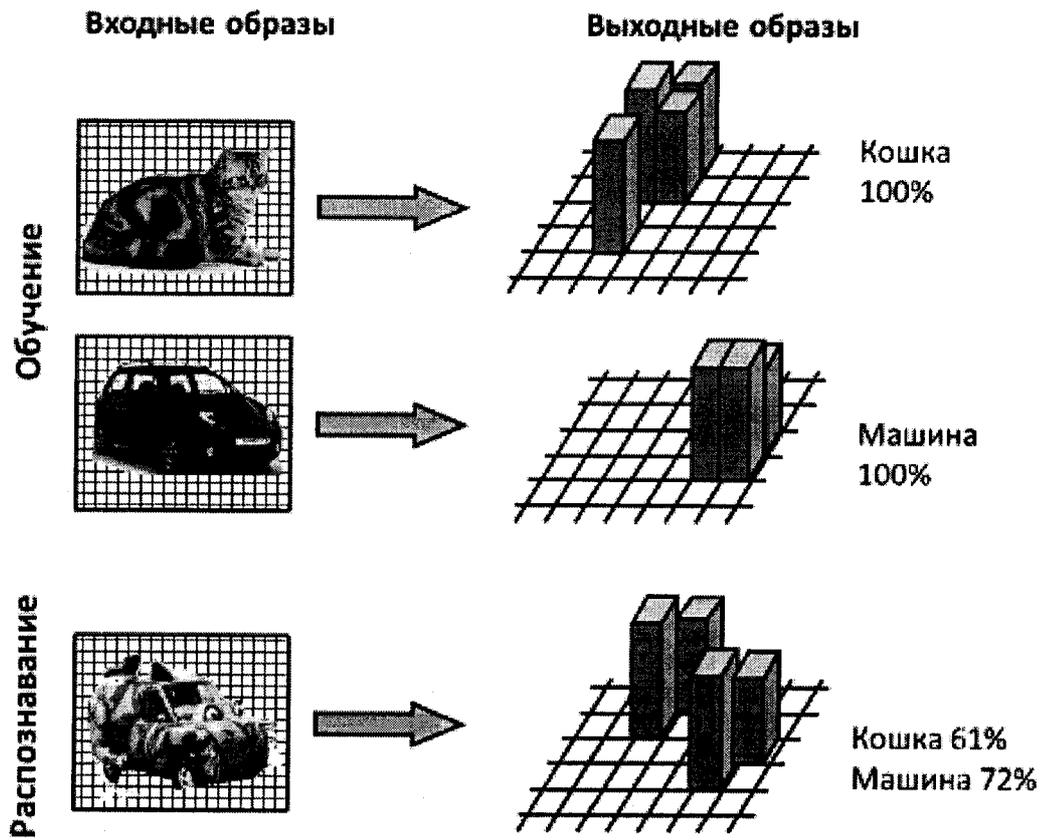




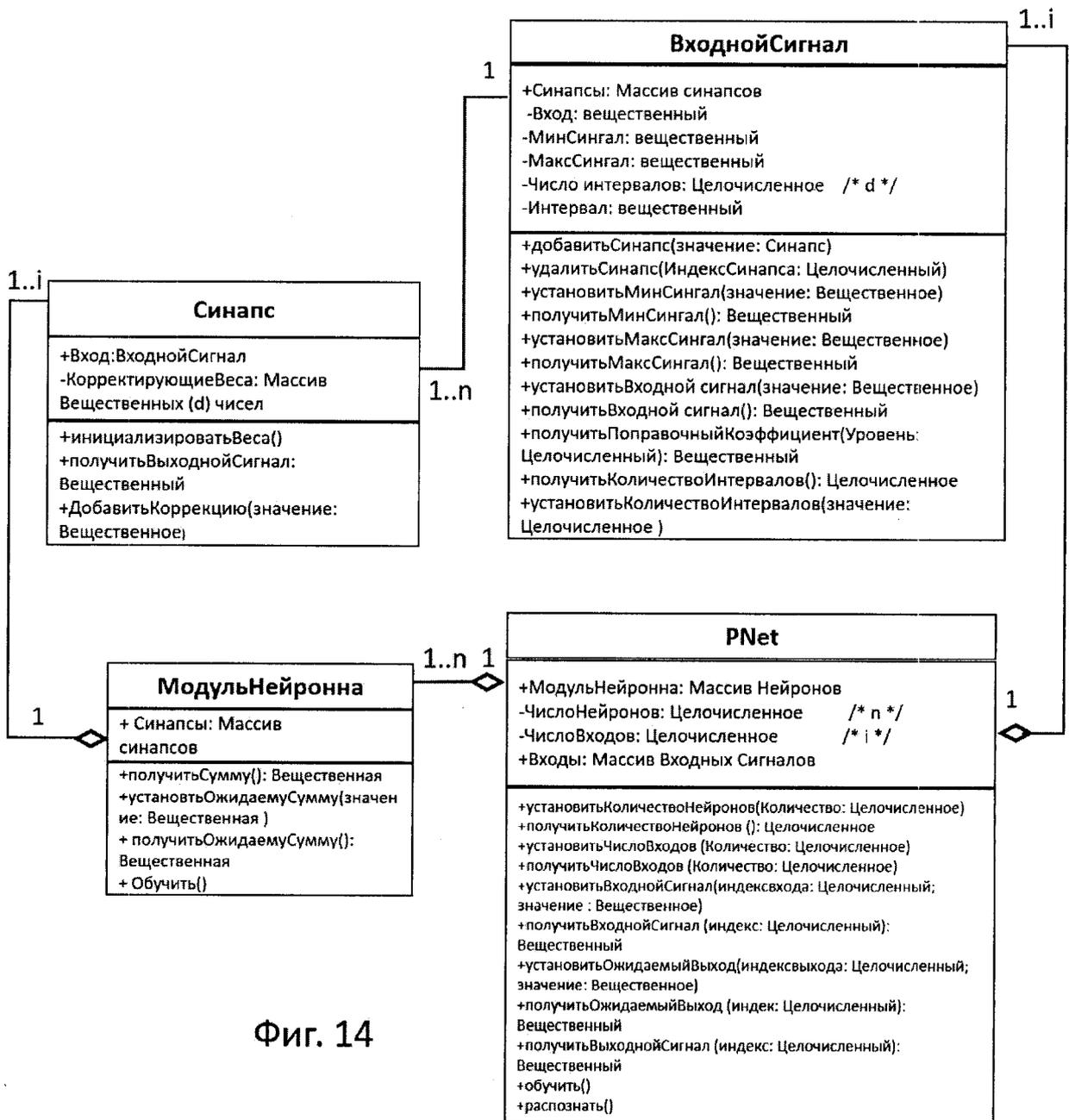
Фиг. 11



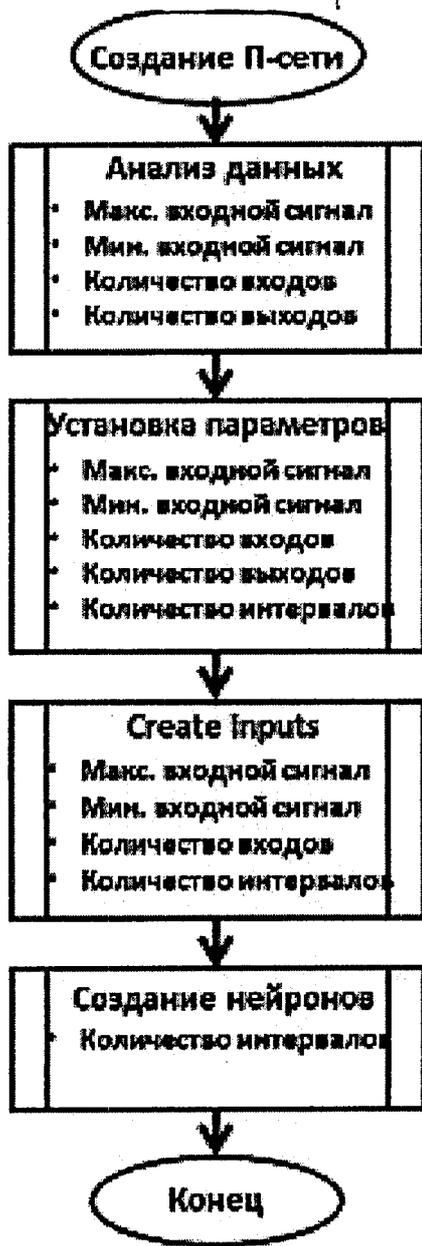
Фиг. 12



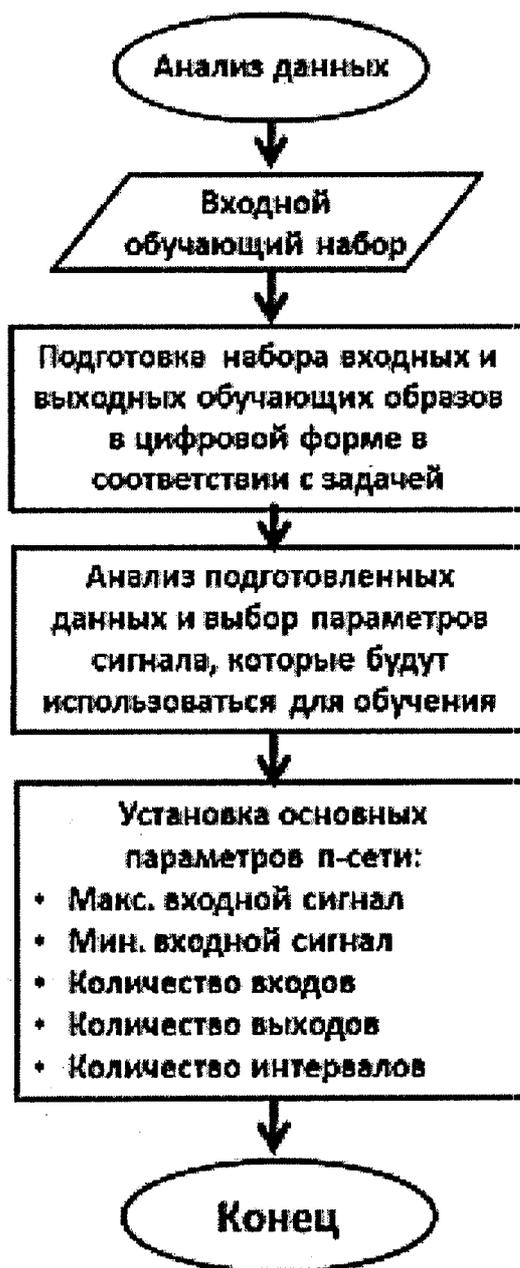
Фиг. 13



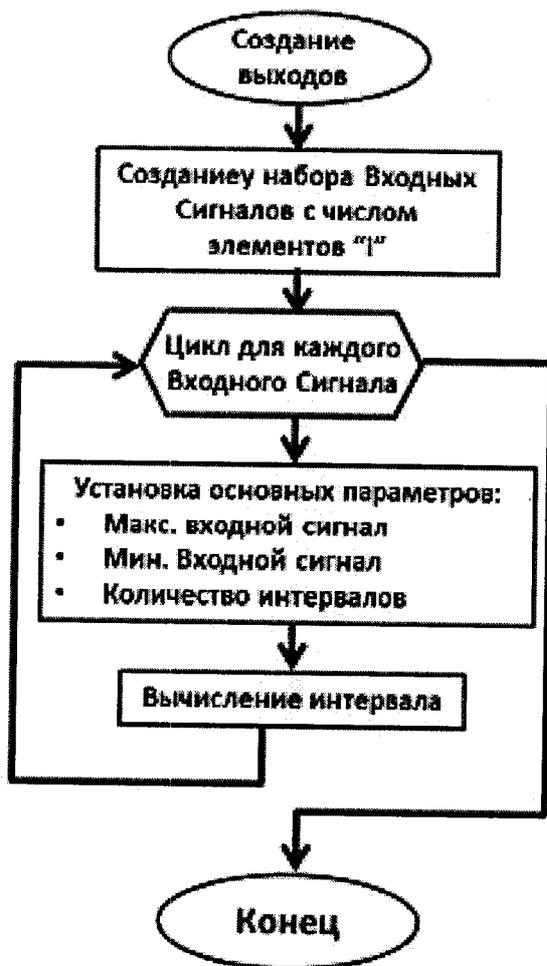
Фиг. 14



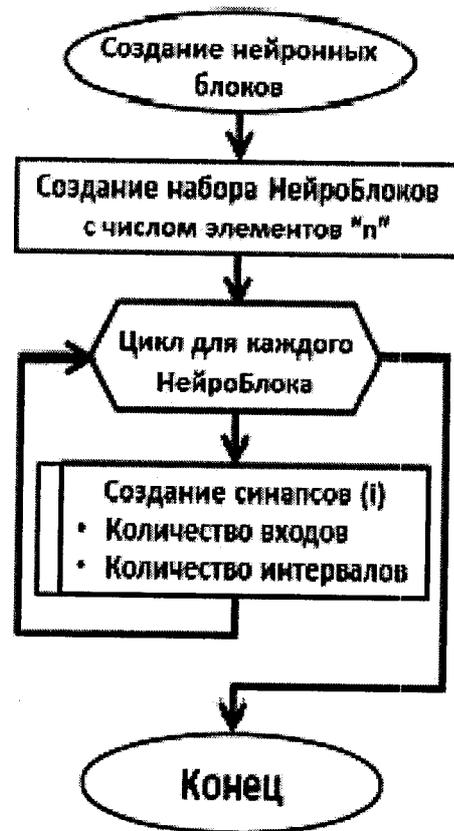
Фиг. 15



Фиг. 16



Фиг. 17



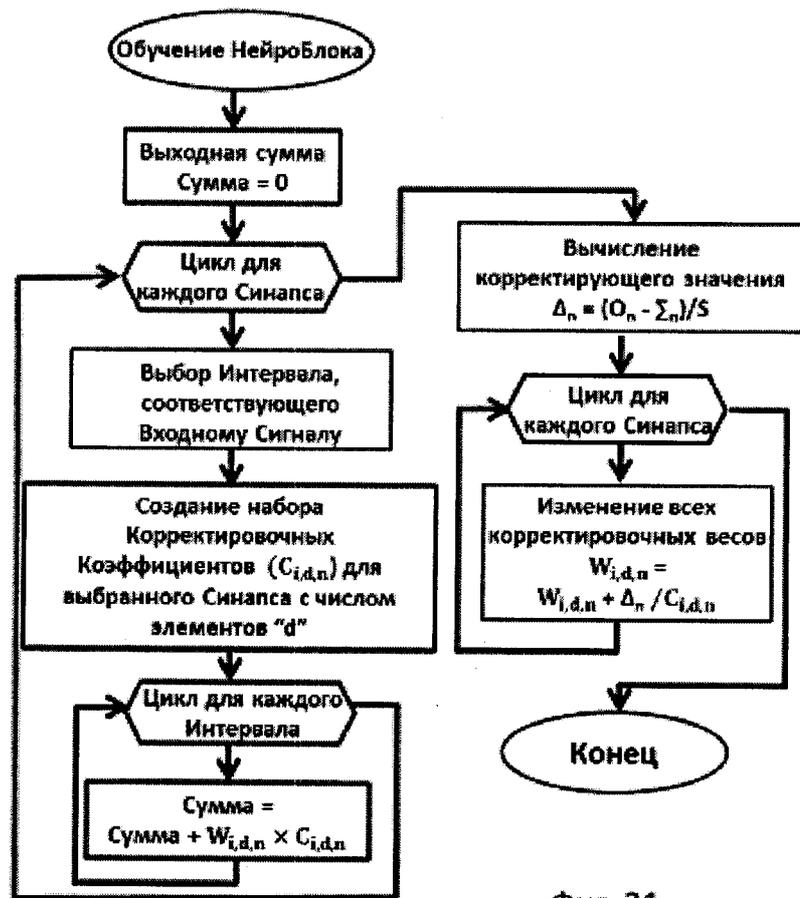
Фиг. 18



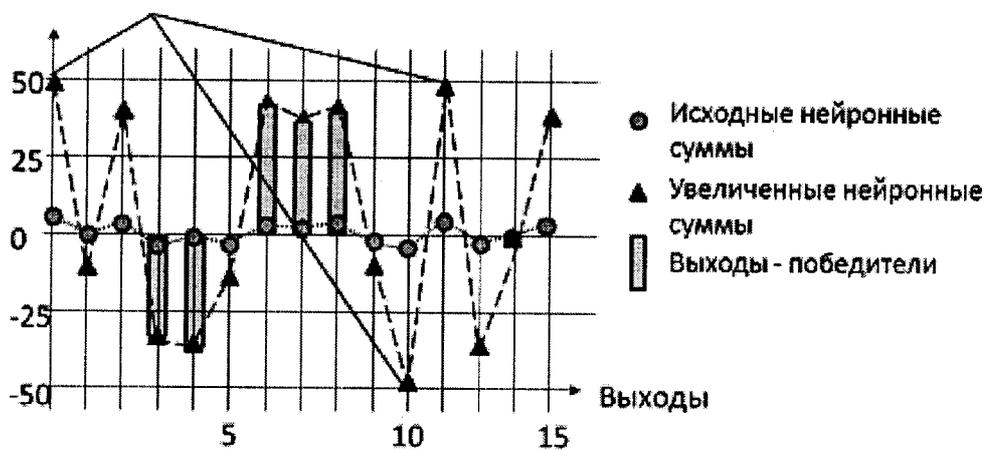
Фиг. 19



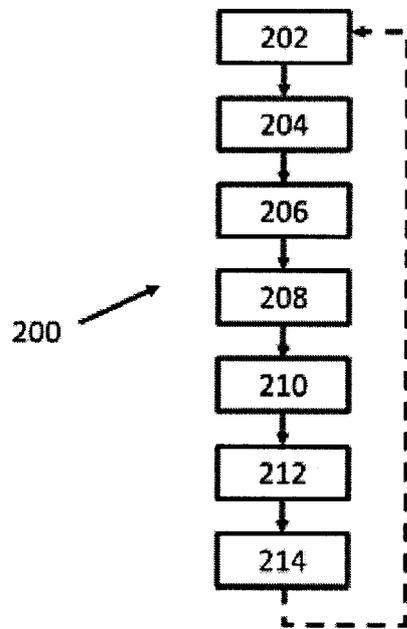
Фиг. 20



Фиг. 21



Фиг. 22



Фиг. 23