Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра информатики

Реферат

на тему

**ПЕрсептрон**

|  |  |  |
| --- | --- | --- |
| Студент |  | А. В. Немкович |
| Преподаватель |  | М. В. Стержанов |

Минск 2019

СОДЕРЖАНИЕ

[Введение 3](#_Toc24571748)

[1. Определение персептрона 5](#_Toc24571749)

[2. Классификация персептронов 8](#_Toc24571750)

[2.1 Персептрон с одним скрытым слоем 8](#_Toc24571751)

[2.2 Однослойный перспетрон 8](#_Toc24571752)

[2.3 Многослойный персептрон 9](#_Toc24571753)

[2.4 Общая схема классификации перспетронов 10](#_Toc24571754)

[3. Задачи, решаемые персептронами. теоремы Розенблатта 11](#_Toc24571755)

[4. Персептронная представляемость и разделимость 12](#_Toc24571756)

[4.1 Персептронная представляемость 12](#_Toc24571757)

[4.1.1 Проблема функции ИСКЛЮЧАЮЩЕГО ИЛИ 12](#_Toc24571758)

[4.2 Линейная разделимость 14](#_Toc24571759)

[4.2.1 Преодоление ограничения линейной разделимости 15](#_Toc24571760)

[5. Алгоритм обучения однослойного персептрона 17](#_Toc24571761)

[Заключение 20](#_Toc24571762)

[Список использованных источников 21](#_Toc24571763)

# **ВВЕДЕНИЕ**

Понятия искусственного нейрона и искусственной нейронной сети появились достаточно давно, еще в 1943 году. Эта была чуть ли не первая статья, в которой предпринимались попытки смоделировать работу мозга. Ее автором был Уоррен Мак-Каллок.

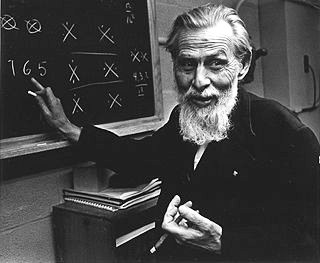


Рисунок 1 – Уоррен Мак-Каллок

Эти идеи продолжил нейрофизиолог Фрэнк Розенблатт. Он предложил схему устройства, моделирующего процесс человеческого восприятия, и назвал его «персептроном» (от латинского perceptio – восприятие). В 1960 году Розенблатт представил первый нейрокомпьютер – «Марк-1», который был способен распознавать некоторые буквы английского алфавита.



Рисунок 2 – Фрэнк Розенблатт

Таким образом персептрон является одной из первых моделей нейросетей, а «Марк-1» – первым в мире нейрокомпьютером.

Персептроны стали очень активно исследовать. На них возлагали большие надежды. Однако, как оказалось, они имели серьезные ограничения. Был такой ученый Минский, который был сокурсником Розенблатта. Видимо, ему не очень понравилось, как все вокруг боготворили персептроны, и он написал целую книгу (1971 год), в которой провел детальнейший их анализ, попутно показав, что они не так уж много и умеют, да и вообще сильно ограничены.

Розенблатт не успел написать ответ Минскому, так как погиб в свой 43 день рождения при крушении лодки.

С тех пор энтузиазм ученых в исследовании персептронов и искусственных сетей поутих. Хотя Минский позднее говорил, что сожалеет, что его книга нанесла такой удар по концепции персептронов. Стали перспективными другие области. Про нейросети забыли. Но потом были открыты новые виды сетей, а также алгоритмы их обучения, что вновь возродило интерес к этой области.

# **1. ОПРЕДЕЛЕНИЕ ПЕРСЕПТРОНА**

В основе персептрона лежит математическая модель восприятия информации мозгом. Разные исследователи по-разному его определяют. В самом общем своем виде (как его описывал Розенблатт) он представляет систему из элементов трех разных типов: сенсоров, ассоциативных элементов и реагирующих элементов.

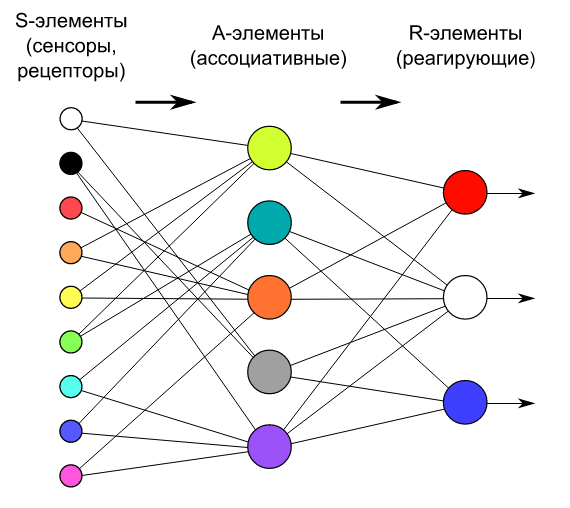
[](https://neuralnet.info/wp-content/uploads/2017/08/4-perceptron.png)

Рисунок 3 – Персептрон по Розенблатту

Рассмотрим принцип работы персептрона.

Первыми в работу включаются S-элементы. Они могут находиться либо в состоянии покоя (сигнал равен 0), либо в состоянии возбуждения (сигнал равен 1).

Далее сигналы от S-элементов передаются A-элементам по так называемым S-A связям.

Затем сигналы от сенсорных элементов, прошедших по S-A связям попадают в A-элементы, которые еще называют ассоциативными элементами. Стоит заметить, что одному A-элементу может соответствовать несколько S-элементов. Если сигналы, поступившие на A-элемент, в совокупности превышают некоторый его порог ​θ, то этот A-элемент возбуждается и выдает сигнал, равный 1. В противном случае (сигнал от S-элементов не превысил порога A-элемента), генерируется нулевой сигнал.

Почему A-элементы назвали ассоциативными? Дело в том, что A-элементы являются агрегаторами сигналов от сенсорных элементов. Например, у нас есть группа сенсоров, каждый из которых распознает кусок буквы «Д» на исследуемой картинке. Однако только их совокупность (то есть когда несколько сенсоров выдали сигнал, равный 1) может возбудить A-элемент целиком. На другие буквы А-элемент не реагирует, только на букву «Д». То есть он **ассоциируется** с буквой «Д». Отсюда и такое название.

Можно привести и другой пример. На самом деле наши глаза состоят из невероятного количества S-элементов (сенсоров), улавливающих падающий свет (около 140 000 000). И у нас какой-то A-элемент, который распознает конкретную часть лица. И вот мы увидели на улице человека. Некоторые A-элементы, которые распознали конкретные части лица, возбуждаются.

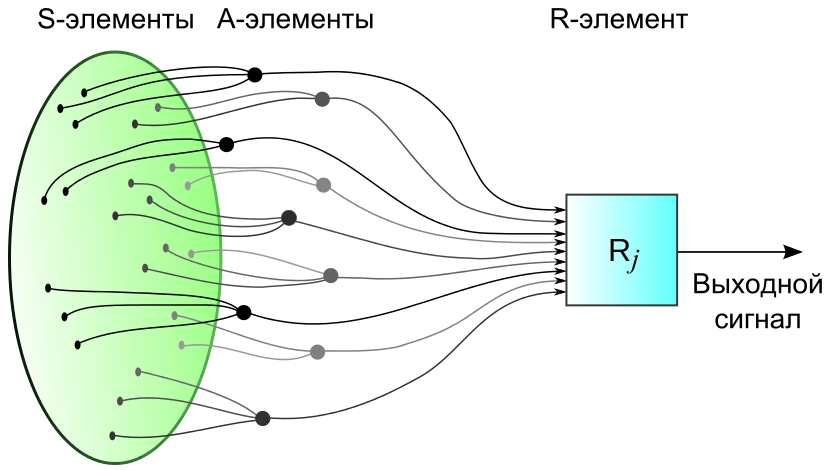
[](https://neuralnet.info/wp-content/uploads/2017/08/4-perceptron-physics.png)

Рисунок 4 –Модель перспетрона на примере глаза

Далее сигналы, которые произвели возбужденные A-элементы, направляются к сумматору (R-элемент), действие которого нам уже известно. Однако, чтобы добраться до R-элемента, они проходят по A-R связям, у которых тоже есть веса. Однако здесь они уже могут принимать любые значения (в отличие от S-A связей).

Финальный аккорд. R-элемент складывает друг с другом взвешенные сигналы от A-элементов и, если превышен определенный порог, генерирует выходной сигнал, равный 1. Это означает, что в общем потоке информации от глаз мы распознали лицо человека.

Если порог не превышен, то выход персептрона равен -1. То есть мы не выделили лицо из общего потока информации.

Так как R-элемент определяет выход персептрона в целом, его назвали реагирующим.

Сформулируем теперь точное определение персептрона:

**Персептрон** (*Perceptron*) — простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.

# **2. КЛАССИФИКАЦИЯ ПЕРСЕПТРОНОВ**

Различают персептроны с одним скрытым слоем и многослойные перспетроны. Рассмотрим подробнее каждый их них.

## 2.1 Персептрон с одним скрытым слоем

Персептрон с одним скрытым слоем — персептрон, у которого имеется только по одному слою S, A и R элементов.

Его частным случаем является однослойный перспетрон.

## 2.2 Однослойный перспетрон

Его ключевая особенность состоит в том, что каждый S-элемент однозначно соответствует одному A-элементу, все S-A связи имеют вес, равный +1, а порог A элементов равен 1.

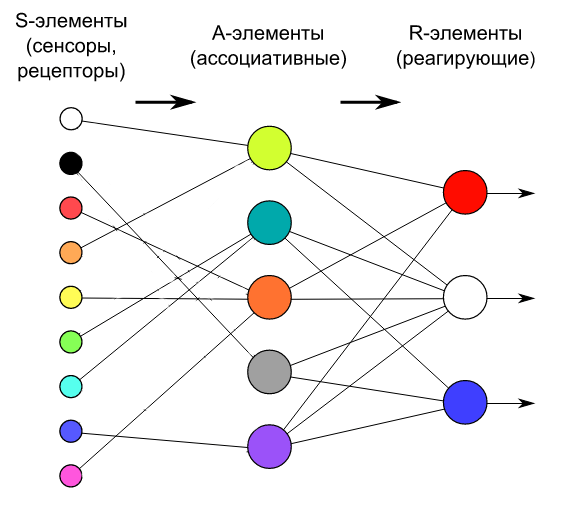


Рисунок 5 – Однослойный персептрон

Рассмотрим первый S-элемент на рисунке 5. Пусть он генерирует сигнал, равный единице. Сигнал проходит по S-A связи и не изменяется, так как любое число, умноженное на 1 равно самому себе. Порог любого А-элемента равен 1. Так как сенсор произвел сигнал, равный 1, то А-элемент однозначно возбудился. Это означает, что он выдал сигнал, равный 1 (так как он тоже может генерировать только 1 или 0 на своем выходе). Далее этот единичный сигнал умножается на произвольный вес A-R связи и попадает в соответствующий R-элемент, который суммирует все поступившие на него взвешенные сигналы, и если они превышают его порог, выдает +1. В противном случае выход данного R-элемента равен -1.

Не считая сенсорных элементов и S-A связей, мы только что описали схему работы искусственного нейрона. И это неслучайно. Однослойный персептрон действительно представляет собой совокупность искусственных нейронов с небольшим отличием. В отличие от искусственного нейрона, у однослойного персептрона входные сигналы могут принимать фиксированные значения: 0 или 1. У искусственного нейрона на вход можно подавать любые значения.

В персептроне R-элементы суммируют взвешенные входные сигналы и, если взвешенная сумма выше некоторого порога, выдают 1. Иначе выходы R-элементов были бы равны -1.

Такое поведение легко задается функцией активации под названием функция единичного скачка. Отличие заключается в том, что функция единичного скачка выдает 0, если порог не превышен, а здесь выдает -1, но это несущественно.

## 2.3 Многослойный персептрон

Под многослойным персептроном понимают два разных вида: многослойный персептрон по Розенблатту и многослойный персептрон по Румельхарту.

Многослойный персептрон по Розенблатту содержит более 1 слоя А-элементов.

Многослойный персептрон по Румельхарту является частным случаем многослойного персептрона по Розенблатту, с двумя особенностями:

– S-A связи могут иметь произвольные веса и обучаться наравне с A-R связями.

– Обучение производится по методу обратного распространения ошибки.

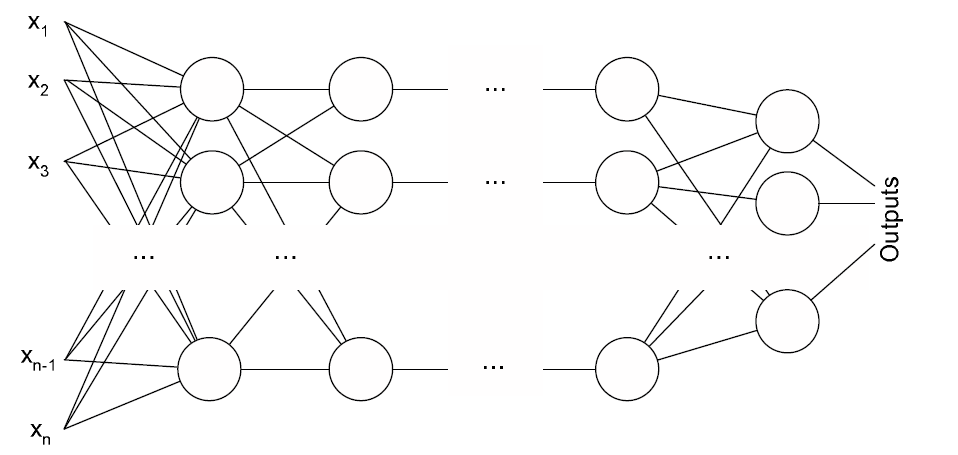


Рисунок 6 – Многослойный перспетрон

## 2.4 Общая схема классификации перспетронов

Для облегчения запоминания классификации приведём опорную схему.



Рисунок 7 – Общая схема классификации персептронов

# **3. ЗАДАЧИ, РЕШАЕМЫЕ ПЕРСЕПТРОНАМИ. ТЕОРЕМЫ РОЗЕНБЛАТТА**

**Теорема 1**

«Если имеется поле сенсоров (матрица) и какая-то классификация, зависящая от него, то множество элементарных персептронов, проводящих успешную классификацию не является пустым.»

Под полем сенсоров понимается множество всех S-элементов. Под классификацией – отнесение объектов к некоторым выделенным классам (для примера, кошки и собаки). Под «непустым множеством элементарных персептронов, проводящих успешную классификацию» понимается, что найдется хотя бы один перспетрон, справившийся с классификацией объектов.

**Теорема 2**

«Если имеется поле сенсоров (матрица) и какая-то классификация, зависящая от него, то процесс обучения с коррекцией ошибок, начинающийся с произвольного исходного состояния, всегда приведёт к достижению решения в течение конечного промежутка времени.»

Под произвольным исходным состоянием тут понимается персептрон с произвольными S-A и A-R весами связей. Под решением в теореме понимается персептрон с определенными весами, успешно решающий нашу задачу на классификацию.

Таким образом, из двух вышеизложенных теорем можно сделать вывод, что с помощью персептронов можно решать задачи классификации, но при этом нужно принимать ко вниманию следующие оговорки:

– Речь идет об элементарных персептронах.

– Объекты классификации должны обладать свойством линейной разделимости (подробнее о ней ниже).

**4. ПЕРСЕПТРОННАЯ ПРЕДСТАВЛЯЕМОСТЬ И РАЗДЕЛИМОСТЬ**

## 4.1 Персептронная представляемость

*Доказательство* теоремы обучения *персептрона* показало, что *персептрон* способен научиться всему, что он способен представлять. Важно при этом уметь различать представляемость и обучаемость. Понятие представляемости относится к способности *персептрона* (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

Для иллюстрации проблемы представляемости допустим, что у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим также, что мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным номером и зажигающей *индикатор* на своей панели при предъявлении карты с нечетным номером. Представима ли такая машина *персептроном*? То есть возможно ли сконструировать *персептрон* и настроить его веса (неважно, каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это достижимо, то говорят, что *персептрон* способен представлять желаемую машину. Мы увидим, что возможности представления однослойными *персептронами* весьма ограниченны. Имеется много простых машин, которые не могут быть представлены *персептроном*, независимо от того, как настраиваются его веса.

### 4.1.1 Проблема функции ИСКЛЮЧАЮЩЕГО ИЛИ

Один из самых пессимистических результатов М.Л. Минского гласит, что однослойный *персептрон* не может воспроизвести такую простую функцию, как *ИСКЛЮЧАЮЩЕЕ ИЛИ*. Это *функция* от двух аргументов, каждый из которых может быть нулем или единицей. Она принимает *значение* единицы, когда один из аргументов равен единице (но не оба). Проблему можно проиллюстрировать с помощью однослойной однонейронной системы с двумя входами, показанной на рисунке 8.

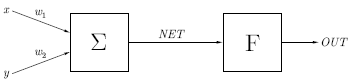


Рисунок 8 – Однослойная однонейронная система с двумя входами

Обозначим один вход через , а другой через , тогда все их возможные комбинации будут состоять из четырех точек на плоскости , как показано на рисунке 9. Например, точка  и  обозначена на рисунке как точка . Таблица 1 показывает требуемую *связь* между входами и выходом, где входные комбинации, которые должны давать нулевой *выход*, помечены  и , единичный *выход* -  и .

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | Значение | Значение | Требуемый выход |
|  | 0 | 0 | 0 |
|  | 1 | 0 | 1 |
|  | 0 | 1 | 1 |
|  | 1 | 1 | 0 |

Таблица 1 – Таблица значений функции *исклюающего или*

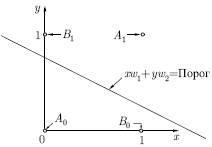


Рисунок 9 – Отображение значений функции *иключающего или* на плоскости

В сети на рисунке 8 *функция*  является обычным порогом, так что  принимает *значение* 0, когда  меньше 0,5, и 1 в случае, когда  больше или равно 0,5. *Нейрон* выполняет следующее *вычисление*:

Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого таблицей 1. Чтобы понять это ограничение, зафиксируем  на величине порога 0,5. *Сеть* в этом случае описывается уравнением (2). Это уравнение линейно *по*  и , т. е. все значения *по*  и , удовлетворяющие этому уравнению, будут лежать на некоторой *прямой* в плоскости .

Любые входные значения для  и  на этой линии будут давать пороговое *значение* 0,5 для . Входные значения с одной стороны *прямой* обеспечат значения  больше порога, следовательно, . Входные значения *по* другую сторону *прямой* обеспечат значения  меньше порога, делая . Изменения значений  и порога будут менять наклон и положение *прямой*. Для того чтобы *сеть* реализовала функцию *ИСКЛЮЧАЮЩЕЕ ИЛИ*, заданную таблицей 1, нужно расположить прямую так, чтобы точки ,  были с одной стороны *прямой*, а точки ,  - с другой. Попытавшись нарисовать такую прямую на рисунке 9, убеждаемся, что это невозможно. Это означает, что какие бы значения ни приписывались весам и порогу, *сеть* неспособна воспроизвести соотношение между входом и выходом, требуемое для представления функции *ИСКЛЮЧАЮЩЕЕ ИЛИ*.

## 4.2 Линейная разделимость

Как мы убедились, невозможно нарисовать прямую линию, разделяющую *плоскость*  так, чтобы реализовывалась *функция* *ИСКЛЮЧАЮЩЕЕ ИЛИ*. К сожалению, этот пример не единственный. Имеется обширный *класс* функций, не реализуемых однослойной сетью. Об этих функциях говорят, что они являются линейно неразделимыми: они-то и накладывают определенные ограничения на возможности однослойных сетей.

*Линейная разделимость* ограничивает *однослойные сети* задачами классификации, в которых *множества* точек (соответствующих входным значениям) могут быть разделены геометрически. Для нашего случая с двумя входами разделитель является *прямой* линией. В случае трех входов разделение осуществляется плоскостью, рассекающей трехмерное *пространство*. Для четырех или более входов *визуализация* невозможна, и необходимо мысленно представить  -мерное *пространство*, рассекаемое "*гиперплоскостью*" — геометрическим объектом, который делит *пространство* четырех или большего числа измерений.

Так как *линейная разделимость* ограничивает возможности *персептронного представления*, то важно знать, является ли данная *функция* разделимой. К сожалению, не существует простого способа определить это, если число переменных велико.

*Нейрон* с  двоичными входами может иметь  различных входных образов, состоящих из нулей и единиц. Так как каждый *входной* образ может соответствовать двум различным бинарным выходам (*единица* и ноль), то всего имеется  функций от  переменных.

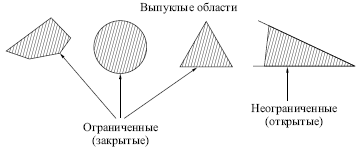
|  |  |  |
| --- | --- | --- |
|  |  | Число линейно разделимых функций |
| 1 | 4 | 4 |
| 2 | 16 | 14 |
| 3 | 256 | 104 |
| 4 | 65536 | 1882 |
| 5 | 4,3x109 | 94572 |
| 6 | 1,8x1019 | 150281134 |

Таблица 2 – Таблица количества значений разделимых функций для входов

Как следует из таблицы 2, *вероятность* того, что случайно выбранная *функция* окажется *линейно разделимой*, весьма мала даже для умеренного числа переменных. *По* этой причине однослойные *персептроны* на практике ограничены простыми задачами.

### 4.2.1 Преодоление ограничения линейной разделимости

К концу 1960-х годов проблема *линейной разделимости* была хорошо понята. К тому же, было известно, что это серьезное ограничение представляемости *однослойными сетями* можно преодолеть, добавив дополнительные слои. Например, двухслойные сети можно получить каскадным соединением двух *однослойных сетей*. Они способны выполнять более общие классификации, отделяя те точки, которые содержатся в выпуклых ограниченных или неограниченных областях. Область называется выпуклой, если для любых двух ее точек соединяющий их *отрезок* целиком лежит в области. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутрь круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей представлены на рисунке 10.

  
Рисунок 10 – Примеры выпуклых областей

Чтобы уточнить требование выпуклости, рассмотрим простую двуслойную *сеть* с двумя входами, которые подведены к двум нейронам первого слоя, соединенными с единственным нейроном в слое 2 (см. Рисунок 11). Пусть порог выходного нейрона равен 0,75, а оба его веса равны 0,5. В этом случае для того, чтобы порог был превышен и на выходе появилась *единица*, требуется, чтобы оба нейрона первого уровня на выходе имели единицу. Таким образом, *выходной нейрон* реализует логическую функцию И. На рисунке 11 каждый *нейрон* слоя 1 разбивает *плоскость*  на две полуплоскости, один обеспечивает единичный *выход* для входов ниже верхней линии, другой — для входов выше нижней линии. На рисунке 11 показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя равен единице только внутри   -образной области. Аналогично, во втором слое может быть использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во *входной слой* может быть образован выпуклый многоугольник любой желаемой формы.

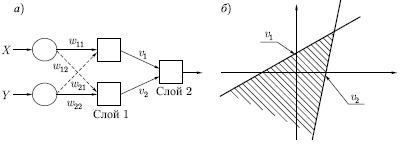


Рисунок 11 – Пример персептрона для реализации выпуклой области

# **5. АЛГОРИТМ ОБУЧЕНИЯ ОДНОСЛОЙНОГО ПЕРСЕПТРОНА**

*Будем считать, что значению "**ложь" соответствует числовое**значение 0, а значению "**Истина" соответствует 1. Персептроном* будем называть устройство, вычисляющее следующую систему функций:

где  - веса *персептрона*,  - порог,  - значения входных сигналов, скобки [] означают переход от булевых (логических) значений к числовым значениям *по* правилам, изложенным выше.

Обучение *персептрона* состоит в подстройке весовых коэффициентов. Пусть имеется набор пар векторов , называемый обучающей выборкой. Будем называть нейронную *сеть* обученной на данной *обучающей выборке*, если при подаче на входы сети каждого вектора  на выходах всякий раз получается соответствующий *вектор* .

Предложенный Ф.Розенблаттом метод обучения состоит в итерационной подстройке *матрицы весов*, последовательно уменьшающей ошибку в выходных векторах. *Алгоритм* включает несколько шагов:

1. Начальные значения весов всех нейронов  полагаются случайными
2. Сети предъявляется входной образ , в результате формируется выходной образ
3. Вычисляется вектор ошибки , делаемой сетью на выходе. Изменение вектора весовых коэффициентов в области малых ошибок должно быть пропорционально ошибке на выходе и равно нулю, если ошибка равна нулю.
4. Вектор весов модифицируется по следующей формуле (дельта -правило): . Здесь  - темп обучения.
5. Шаги 2-4 повторяются для всех обучающих векторов. Один цикл последовательного предъявления всей выборки называется эпохой.

Обучение завершается по истечении нескольких эпох: а) когда итерации сойдутся, т.е. вектор весов перестает изменяться, или б) когда полная, просуммированная по всем векторам абсолютная ошибка станет меньше некоторого малого значения.

Объясним данный *алгоритм* более подробно. Подаем на вход *персептрона* такой *вектор* , для которого уже известен правильный ответ. Если выходной сигнал *персептрона* совпадает с правильным ответом, то никаких действий предпринимать не надо. В случае ошибки, необходимо обучить *персептрон* правильно решать данный пример. Ошибки могут быть двух типов. Рассмотрим каждый из них.

Первый тип ошибки: на выходе *персептрона* — 0, а правильный ответ — 1. Для того чтобы *персептрон* выдавал *правильный ответ*, необходимо, чтобы сумма в правой части (3) стала больше. Поскольку переменные принимают значения 0 или 1, увеличение суммы может быть достигнуто за счет увеличения весов . Однако нет смысла увеличивать веса при переменных , которые равны нулю. Таким образом, следует увеличить веса  при тех переменных , которые равны 1.

**Первое правило.**

Если на выходе *персептрона* получен 0, а *правильный ответ* равен 1, то необходимо увеличить веса связей между одновременно активными нейронами. При этом выходной *персептрон* считается активным.

Второй тип ошибки: на выходе *персептрона* - 1, а правильный ответ равен нулю. Для обучения правильному решению данного примера следует уменьшить сумму в правой части (3). Следовательно, необходимо уменьшить веса связей  при тех переменных, которые равны 1 (поскольку нет смысла уменьшать веса связей при равных нулю переменных ). Необходимо также провести эту процедуру для всех активных нейронов предыдущих слоев (для многослойных персептронов). В результате получаем второе правило.

**Второе правило.**

Если на выходе *персептрона* получена *единица*, а *правильный ответ* равен нулю, то необходимо уменьшить веса связей между одновременно активными нейронами.

Таким образом, процедура обучения сводится к последовательному перебору всех примеров обучающего *множества* с применением правил обучения для ошибочно решенных примеров. Если после очередного *цикла* предъявления всех примеров окажется, что все они решены правильно, то процедура обучения завершается.

Нерассмотренными остались два вопроса. Первый — о *сходимости* процедуры обучения. Второй — на сколько нужно увеличивать (уменьшать) веса связей при применении правил обучения.

Ответ на первый вопрос дают следующие теоремы.

**Теорема о сходимости *персептрона*.**

Если существует *вектор* параметров , при котором *персептрон* правильно решает все примеры обучающей выборки, то при обучении *персептрона* *по* вышеописанному алгоритму решение будет найдено за конечное число шагов.

**Теорема о "зацикливании" *персептрона*.**

Если не существует вектора параметров , при котором *персептрон* правильно решает все примеры обучающей выборки, то при обучении *персептрона* *по* данному правилу через конечное число шагов *вектор* весов начнет повторяться.

Таким образом, данные теоремы утверждают, что, запустив процедуру обучения *персептрона*, через конечное время мы либо получим обучившийся *персептрон*, либо ответ, что данный *персептрон* поставленной задаче обучиться не может.

# **ЗАКЛЮЧЕНИЕ**

Персептрон является видом нейронных сетей, подходящим для решения задачи классификации в силу представляемости им линейно-разделимых функций. В то же время проблема разделимости может быть устранена переходом к многослойным персептронам для решения более сложных задач.

Исследования свойств персептрона, его ограниченностей стали основой в изучении новых видов сетей и алгоритмов их обучения.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Персептроны [Электронный ресурс] – Режим доступа: <https://neuralnet.info/chapter/персептроны/#История-персептрона> – Дата доступа: 13.11.2019

[2] Персептроны. Представимость и разделимость [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/88/88/lecture/20529> – Дата доступа: 11.11.2019

[3] Персептроны. Обучение персептрона [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/88/88/lecture/20531> – Дата доступа

07.11.2019

[4] Обучение персептрона. Дельта-правило [Электронный ресурс] – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/perceptron-learning.html> – Дата доступа 09.11.2019

[5] Теоремы Розенблатта об элементарных перцептронах [Электронный ресурс] – Режим доступа: <https://studopedia.org/5-95544.html> – Дата доступа 09.11.2019

[6] Основы ИНС [Электронный ресурс] – Режим доступа: <https://neuralnet.info/chapter/основы-инс/#Искусственный-нейрон> – Дата доступа 12.11.2019