## Лабораторная работа №2. Простые нейронные сети. Персептрон

В качестве научного предмета искусственные нейронные сети впервые заявили о себе в 40-е годы. Стремясь воспроизвести функции человеческого мозга, исследователи создали простые аппаратные (а позже программные) модели биологического нейрона и системы его соединений. Когда нейрофизиологи достигли более глубокого понимания нервной системы человека, эти ранние попытки стали восприниматься как весьма грубые аппроксимации. Тем не менее, на этом пути были достигнуты впечатляющие результаты, стимулировавшие дальнейшие исследования, приведшие к созданию более изощренных сетей.

Σ

x1, w1

xn, wn

Порог

Выход

Рис. 1. Персептронный нейрон

Первое систематическое изучение искусственных нейронных сетей было предпринято Маккалокком и Питтсом в 1943 г. [I]. Позднее в работе [3] они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 1, использовалась в большей части их работы. Элемент **Σ** умножает каждый вход *х* на вес *w* и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае – нулю. Эти системы (и множество им подобных) получили название *персептронов.* Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов (см. рис. 2), хотя в принципе описываются и более сложные системы**.**

В 60-е годы персептроны вызвали большой интерес и оптимизм. Розенблатт [4] доказал замечательную теорему об обучении персептронов, объясняемую ниже. Уидроу [5-8] дал ряд убедительных демонстраций систем персептронного типа, и исследователи во всем мире стремились изучить возможности этих систем. Первоначальная эйфория сменилась разочарованием, когда оказалось, что персептроны не способны обучиться решению ряда простых задач. Минский [2] строго проанализировал эту проблему и показал, что имеются жесткие ограничения на то, что могут выполнять однослойные персептроны, и, следовательно, на то, чему они могут обучаться. Так как в то время методы обучения многослойных сетей не были известны, исследователи перешли в более многообещающие области, и исследования в области нейронных сетей пришли в упадок. Недавнее открытие методов обучения многослойных сетей в большей степени, чем какой-либо иной фактор, повлияло на возрождение интереса и исследовательских усилий.

Σ

Σ

Σ

x1

x2

xn

Выход1

Выход2

Выходn

Рис. 2. Персептрон со многими выходами

Работа Минского, возможно, и охладила пыл энтузиастов персептрона, но обеспечила время для необходимой консолидации и развития лежащей в основе теории. Важно отметить, что анализ Минского не был опровергнут. Он остается важным исследованием и должен изучаться, чтобы ошибки 60-х годов не повторились.

Несмотря на свои ограничения, персептроны широко изучались (хотя не слишком широко использовались). Теория персептронов является основой для многих других типов искусственных нейронных сетей, и персептроны иллюстрируют важные принципы. В силу этих причин они являются логической исходной точкой для изучения искусственных нейронных сетей.

**ПЕРСЕПТРОННАЯ ПРЕДСТАВЛЯЕМОСТЬ**

Доказательство теоремы обучения персептрона [4] 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.

1

2

3

9

Четный

**Рис. 3. Система распознавания изображений**

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 (см. рис. 3). Представима ли такая машина персептроном? То есть, может ли быть сконструирован персептрон и настроены его веса (неважно каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это так, 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.

**Проблема функции ИСКЛЮЧАЮЩЕЕ ИЛИ**

Один из самых пессимистических результатов Минского показывает, 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 рис. 4. Обозначим один вход через *х,* а другой через *у,* тогда все их возможные комбинации будут состоять из четырех точек на плоскости *х*-у, как показано на рис. 5. Например, точка *х* = 0 и *у* = 0 обозначена на рисунке как точка А Табл. 2.1 показывает требуемую связь между входами и выходом, где входные комбинации, которые должны давать нулевой выход, помечены А0 и А1, единичный выход – В0 и В1.

Σ

F

x, w1

y, w2

Сигнал

Выход

**Рис. 4. Однонейронная система**

В сети на рис. 4 функция *F* является обычным порогом, так что OUT принимает значение ноль, когда NET меньше 0,5, и единица в случае, когда NET больше или равно 0,5. Нейрон выполняет следующее вычисление:

NET = xw1 + yw2 (1)

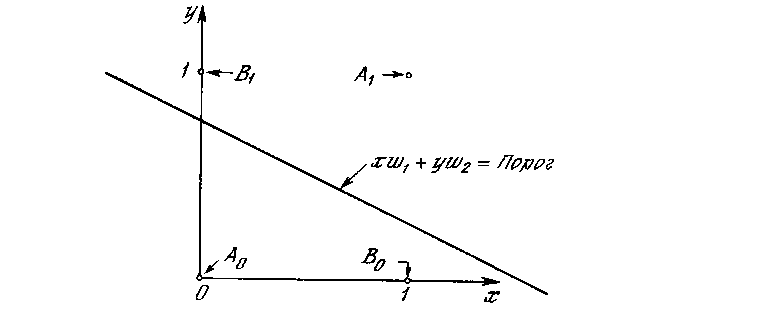
Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого табл. 1. Чтобы понять это ограничение, зафиксируем NET на величине порога 0,5. Сеть в этом случае описывается уравнением (2). Это уравнение линейно по *х* и *у,* т. е. все значения по *х* и *у,* удовлетворяющие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

xw1 + yw2 = 0,5 (2)

***Таблица 1. Таблица истинности для функции ИСКЛЮЧАЮЩЕЕ ИЛИ***

|  |  |  |  |
| --- | --- | --- | --- |
| Точки | Значения *х* | Значения *у* | Требуемый выход |
| A0 | 0 | 0 | 0 |
| B0 | 1 | 0 | 1 |
| B1 | 0 | 1 | 1 |
| A1 | 1 | 1 | 0 |

Любые входные значения для *х* и *у* на этой линии будут давать пороговое значение 0,5 для NET. Входные значения с одной стороны прямой обеспечат значения NET больше порога, следовательно, OUT=1. Входные значения по другую сторону прямой обеспечат значения NET меньше порогового значения, делая OUT равным 0. Изменения значений *w*1, *w*2 и порога будут менять наклон и положение прямой. Для того чтобы сеть реализовала функцию ИСКЛЮЧАЮЩЕЕ ИЛИ, заданную табл. 1, нужно расположить прямую так, чтобы точки А были с одной стороны прямой, а точки В – с другой. Попытавшись нарисовать такую прямую на рис. 5, 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.



**Рис. 5. Проблема ИСКЛЮЧАЮЩЕЕ ИЛИ**

Взглянув на задачу с другой точки зрения, рассмотрим NET как поверхность над плоскостью *х-у.* Каждая точка этой поверхности находится над соответствующей точкой плоскости *х-у* на расстоянии, равном значению NET в этой точке. Можно показать, что наклон этой NET-поверхности одинаков для всей поверхности *х-у.* Все точки, в которых значение NET равно величине порога, проектируются на линию уровня плоскости NET (см. рис. 6).

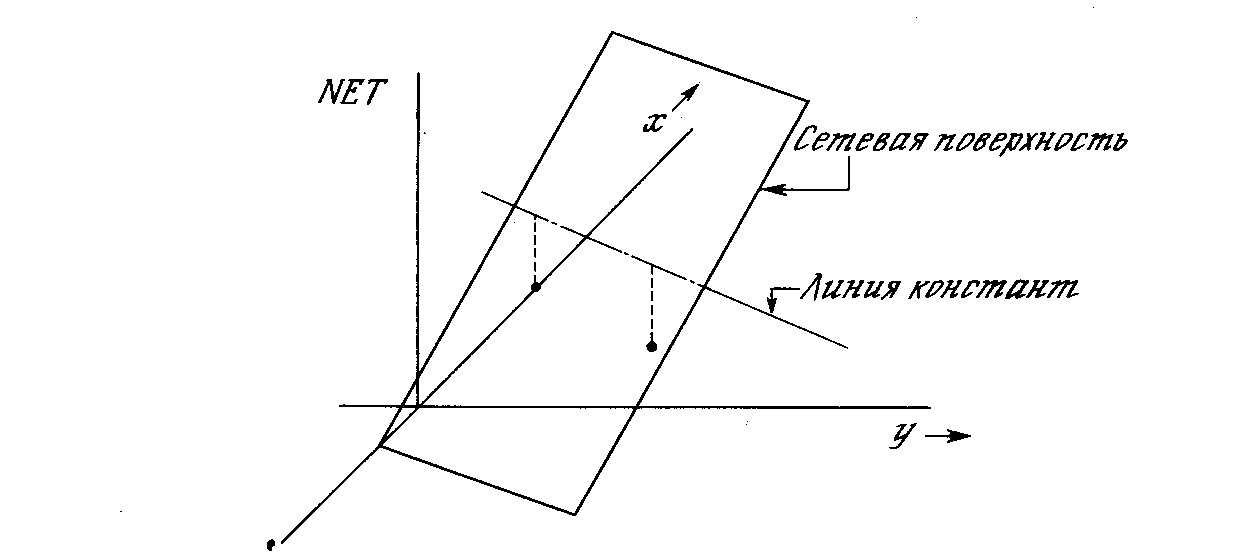


Рис. 6. Персептронная NET-плоскость

Ясно, что все точки по одну сторону пороговой прямой спроецируются в значения NET, большие порога, а точки по другую сторону дадут меньшие значения NET. Таким образом, пороговая линия разбивает плоскость *х-у* на две области. Во всех точках по одну сторону пороговой прямой значение OUT равно единице, по другую сторону – нулю.

#### Линейная разделимость

Как мы видели, невозможно нарисовать прямую линию, разделяющую плоскость *х-у* 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, и они 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 n-мерное пространство, рассекаемое «гиперплоскостью» – геометрическим объектом, который рассекает 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 *п* двоичными входами может иметь 2n 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 22n функций от n переменных.

Таблица 2. Линейно разделимые функции

|  |  |  |
| --- | --- | --- |
| n | 22n | Число линейно разделимых функций |
| 1 | 4 | 4 |
| 2 | 16 | 14 |
| 3 | 256 | 104 |
| 4 | 65536 | 1882 |
| 5 | 4,3х109 | 94572 |

(Взято из R. 0. Winder, Single-stage logic. Paper presented at the AIEE Fall General Meeting, 1960.)

Как видно из табл. 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.

#### Преодоление ограничения линейной разделимости

К концу 60-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 рис. 7.

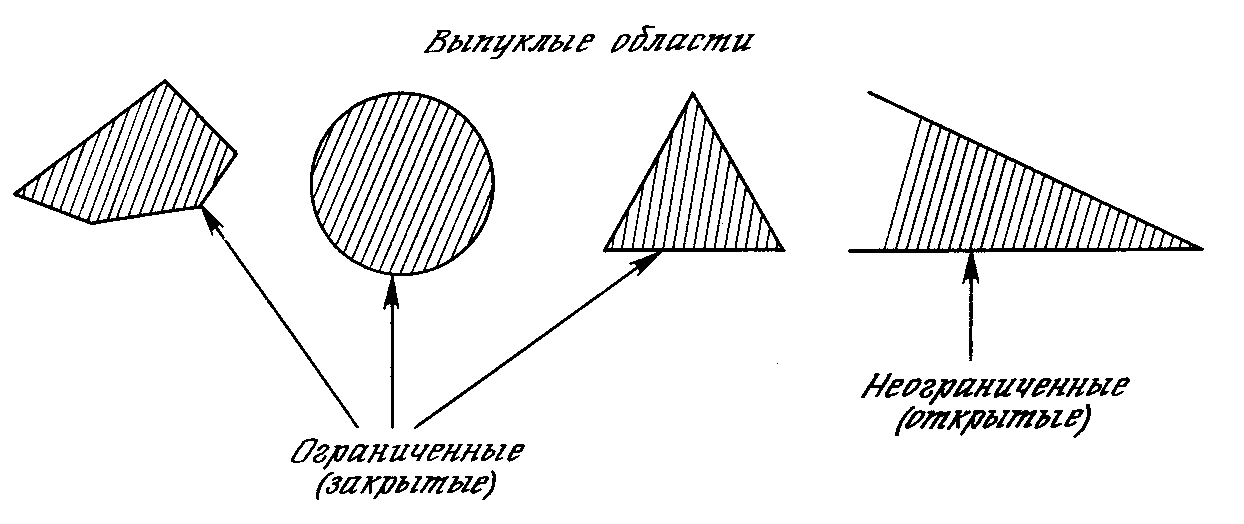


Рис. 7. Выпуклые ограниченные и неограниченные области

Чтобы уточнить требование выпуклости, рассмотрим простую двухслойную сеть с двумя входами, подведенными к двум нейронам первого слоя, соединенными с единственным нейроном в слое 2 (см. рис. 8). 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 рис. 8 каждый нейрон слоя 1 разбивает плоскость *х-у* на две полуплоскости, один обеспечивает единичный выход для входов ниже верхней линии, другой – для входов выше нижней линии. На рис. 8 показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя равен единице только внутри V-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. Так как они образованы с помощью операции И над областями, задаваемыми линиями, то все такие многогранники выпуклы, следовательно, только выпуклые области и возникают. Точки, не составляющие выпуклой области, не могут быть отделены от других точек плоскости двухслойной сетью.

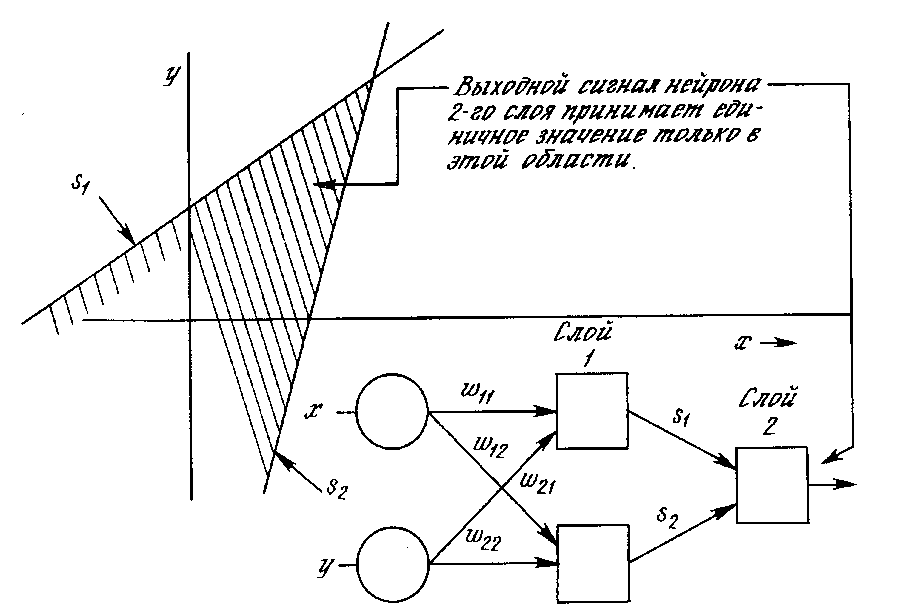


Рис. 8. Выпуклая область решений, задаваемая двухслойной сетью

Нейрон второго слоя не ограничен функцией И. Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое ИЛИ. Имеется 16 двоичных функций от двух переменных. Если выбирать подходящим образом веса и порог, то можно воспроизвести 14 из них (все, кроме ИСКЛЮЧАЮЩЕЕ ИЛИ и ИСКЛЮЧАЮЩЕЕ НЕТ).

Входы не обязательно должны быть двоичными. Вектор непрерывных входов может представлять собой произвольную точку на плоскости *х-у.* В этом случае мы имеем дело со способностью сети разбивать плоскость на непрерывные области, а не с разделением дискретных множеств точек. Для всех этих функций, однако, линейная разделимость показывает, что выход нейрона второго слоя равен единице только в части плоскости *х-у,* ограниченной многоугольной областью. Поэтому для разделения плоскостей P и Q необходимо, чтобы все P лежали внутри выпуклой многоугольной области, не содержащей точек Q (или наоборот).

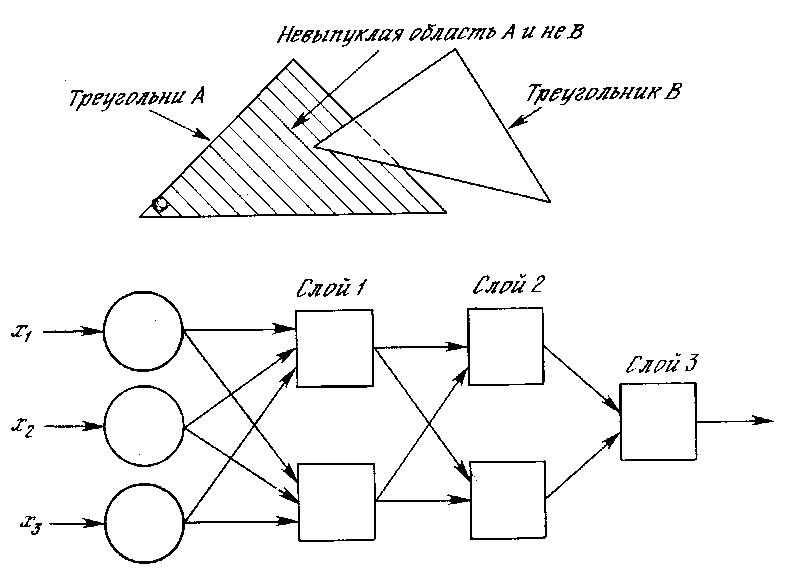


Рис. 9. «Вогнутая» область решений, задаваемая трехслойной сетью

Трехслойная сеть, однако, является более общей. Ее классифицирующие возможности ограничены лишь числом искусственных нейронов и весов. Ограничения на выпуклость отсутствуют. Теперь нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рис. 2.9 иллюстрируется случай, когда два треугольника A и B, скомбинированные с помощью функций «A и не B*»,* задают невыпуклую область. При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок не все выходные области второго слоя должны пересекаться. Возможно, следовательно, объединять различные области, выпуклые и невыпуклые, выдавая на выходе единицу, всякий раз, когда входной вектор принадлежит одной из них.

Несмотря на то, что возможности многослойных сетей были известны давно, в течение многих лет не было теоретически обоснованного алгоритма для настройки их весов. В последующих главах мы детально изучим многослойные обучающие алгоритмы, но сейчас достаточно понимать проблему и знать, что исследования привели к определенным результатом.

#### Эффективность запоминания

Серьезные вопросы имеются относительно эффективности запоминания информации в персептроне (или любых других нейронных сетях) по сравнению с обычной компьютерной памятью и методами поиска информации в ней. Например, в компьютерной памяти можно хранить все входные образы вместе с классифицирующими битами. Компьютер должен найти требуемый образ и дать его классификацию. Различные хорошо известные методы могли бы быть использованы для ускорения поиска. Если точное соответствие не найдено, то для ответа может быть использовано правило ближайшего соседа.

Число битов, необходимое для хранения этой же информации в весах персептрона, может быть значительно меньшим по сравнению с методом обычной компьютерной памяти, если образы допускают экономичную запись. Однако Минский [2] построил патологические примеры, в которых число битов, требуемых для представления весов, растет с размерностью задачи быстрее, чем экспоненциально. В этих случаях требования к памяти с ростом размерности задачи быстро становятся невыполнимыми. Если, как он предположил, эта ситуация не является исключением, то персептроны часто могут быть ограничены только малыми задачами. Насколько общими являются такие неподатливые множества образов? Это остается открытым вопросом, относящимся ко всем нейронным сетям. Поиски ответа чрезвычайно важны для исследований по нейронным сетям.

**ОБУЧЕНИЕ ПЕРСЕПТРОНА**

Способность искусственных нейронных сетей обучаться является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами моделируют себя в результате попыток достичь лучшей модели поведения.

Используя критерий линейной разделимости, можно решить, способна ли однослойная нейронная сеть реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Чтобы сеть представляла практическую ценность, нужен систематический метод (алгоритм) для вычисления этих значений. Розенблатт [4] сделал это в своем алгоритме обучения персептрона вместе с доказательством того, что персептрон может быть обучен всему, что он может реализовывать.

Обучение может быть с учителем или без него. Для обучения с учителем нужен «внешний» учитель, который оценивал бы поведение системы и управлял ее последующими модификациями. При обучении без учителя, рассматриваемого в последующих главах, сеть путем самоорганизации делает требуемые изменения. Обучение персептрона является обучением с учителем.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и сеть становится в определенном смысле самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается». Доказательство Розенблатта стало основной вехой и дало мощный импульс исследованиям в этой области. Сегодня в той или иной форме элементы алгоритма обучения персептрона встречаются во многих сетевых парадигмах.

**АЛГОРИТМ ОБУЧЕНИЯ ПЕРСЕПТРОНА**

Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. Допустим, что входные образы нанесены на демонстрационные карты. Каждая карта разбита на квадраты и от каждого квадрата на персептрон подается вход. Если в квадрате имеется линия, то от него подается единица, в противном случае – ноль. Множество квадратов на карте задает, таким образом, множество нулей и единиц, которое и подается на входы персептрона. Цель состоит в том, чтобы научить персептрон включать индикатор при подаче на него множества входов, задающих нечетное число, и не включать в случае четного.

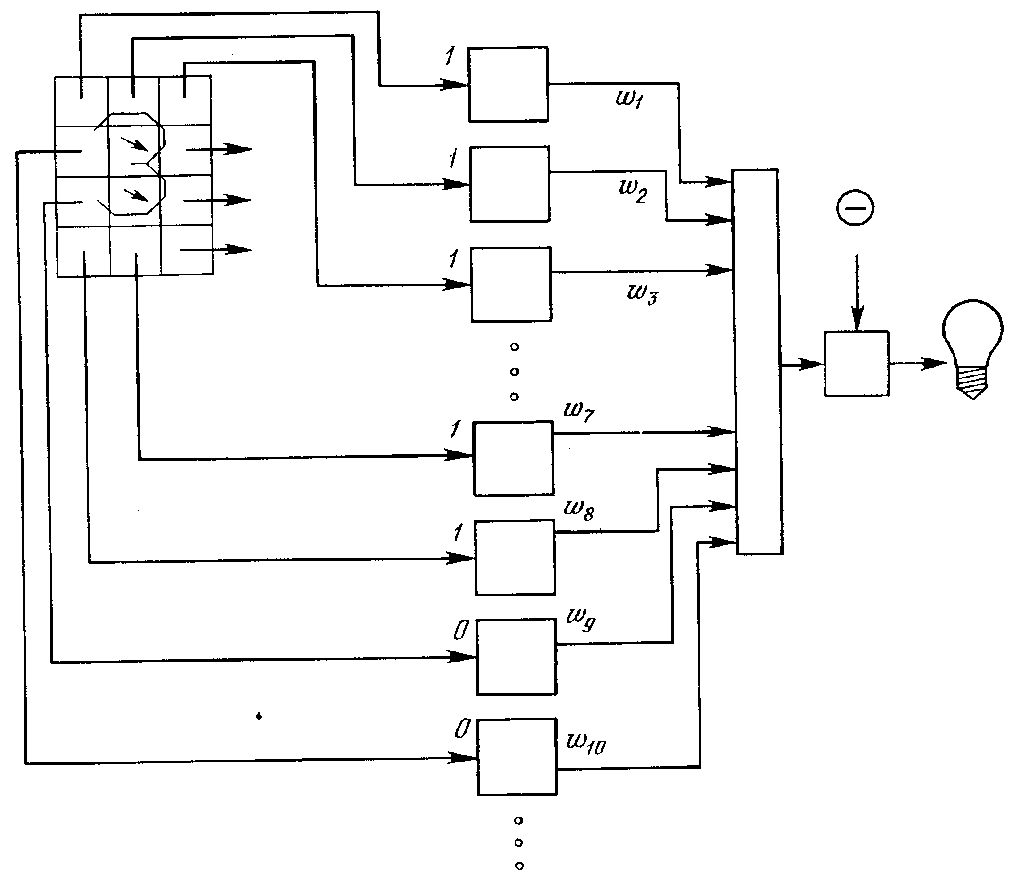


Рис. 10. Персептронная система распознавания изображений

На рис. 10 показана такая персептронная конфигурация. Допустим, что вектор **Х** является образом распознаваемой демонстрационной карты. Каждая компонента (квадрат) **Х** – (*x*1, *x*2, …, *x*n) – умножается на соответствующую компоненту вектора весов **W** – (*w*1, *w*2, ..., *w*n). Эти произведения суммируются. Если сумма превышает порог Θ, то выход нейрона Y равен единице (индикатор зажигается), в противном случае он – ноль. Эта операция компактно записывается в векторной форме как Y = **XW,** а после нее следует пороговая операция.

Для обучения сети образ **Х** подается на вход и вычисляется выход Y. Если выход Y правилен, то ничего не меняется. Однако если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку.

Чтобы увидеть, как это осуществляется, допустим, что демонстрационная карта с цифрой 3 подана на вход и выход Y равен 1 (показывая нечетность). Так как это правильный ответ, то веса не изменяются. Если, однако, на вход подается карта с номером 4 и выход Y равен единице (нечетный), то веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если карта с номером 3 дает нулевой выход, то веса, присоединенные к единичным входам, должны быть увеличены, чтобы скорректировать ошибку.

Этот метод обучения может быть подытожен следующим образом:

1. Подать входной образ и вычислить Y.

2 а. Если выход правильный, то перейти на шаг 1;

б. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам; или

в. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.

3. Перейти на шаг 1.

За конечное число шагов сеть научится разделять карты на четные и нечетные при условии, что множество цифр линейно разделимо. Это значит, что для всех нечетных карт выход будет больше порога, а для всех четных – меньше. Отметим, что это обучение глобально, т. е. сеть обучается на всем множестве карт. Возникает вопрос о том, как это множество должно предъявляться, чтобы минимизировать время обучения. Должны ли элементы множества предъявляться - последовательно друг за другом или карты следует выбирать случайно? Несложная теория служит здесь путеводителем.

#### Дельта-правило

Важное обобщение алгоритма обучения персептрона, называемое дельта-правилом, переносит этот метод на непрерывные входы и выходы. Чтобы понять, как оно было получено, шаг 2 алгоритма обучения персептрона может быть сформулирован в обобщенной форме с помощью введения величины δ, которая равна разности между требуемым или целевым выходом T и реальным выходом Y

δ = (T - Y). (3)

Случай, когда δ=0, соответствует шагу 2а, когда выход правилен и в сети ничего не изменяется. Шаг 2б соответствует случаю δ > 0, а шаг 2в случаю δ < 0.

В любом из этих случаев персептронный алгоритм обучения сохраняется, если δ умножается на величину каждого входа *х*i и это произведение добавляется к соответствующему весу. С целью обобщения вводится коэффициент «скорости обучения» η), который умножается на δ*х*i*,* что позволяет управлять средней величиной изменения весов.

В алгебраической форме записи

Δi = ηδ*x*i, (4)

*w*(n+1) = *w*(n) + Δi, (5)

где Δi – коррекция, связанная с i-м входом *х*i; *w*i(n+1) – значение веса i после коррекции; *w*i{n) -значение веса i до коррекции.

Дельта-правило модифицирует веса в соответствии с требуемым и действительным значениями выхода каждой полярности, как для непрерывных, так и для бинарных входов и выходов. Эти свойства открыли множество новых приложений.

#### Трудности с алгоритмом обучения персептрона

Может оказаться затруднительным определить, выполнено ли условие разделимости для конкретного обучающего множества. Кроме того, во многих встречающихся на практике ситуациях входы часто меняются во времени и могут быть разделимы в один момент времени и неразделимы в другой. В доказательстве алгоритма обучения персептрона ничего не говорится также о том, сколько шагов требуется для обучения сети. Мало утешительного в знании того, что обучение закончится за конечное число шагов, если необходимое для этого время сравнимо с геологической эпохой. Кроме того, не доказано, что персептронный алгоритм обучения более быстр по сравнению с простым перебором всех возможных значений весов, и в некоторых случаях этот примитивный подход может оказаться лучше.

На эти вопросы никогда не находилось удовлетворительного ответа, они относятся к природе обучающего материала. В различной форме они возникают и при рассмотрении других сетевых парадигм. Ответы для современных сетей, как правило, не более удовлетворительны, чем для персептрона. Эти проблемы являются важной областью современных исследований.

#### Задания для работы в аудитории

* 1. Ознакомится с теоретическими сведениями, изложенными в методическом указании и ответить на контрольные вопросы.
  2. По изложенному в методическом указании алгоритму спроектировать алгоритм и создать программу песептронной системы для распознавания четных и нечетных чисел.
  3. Модифицировать алгоритм и программу, введя в неё, дельта-правило.

#### Задания для самостоятельной работы

По заданию преподавателя реализовать персептронную систему для разделения пространств различных объектов. Результаты оформить в виде отчета и защитить на следующем занятии.

#### Контрольные вопросы

1. Изложить принцип работы персептрона.
2. Что такое персептронный нейрон, и каков принцип его действия?
3. В чем заключается проблема «ИСКЛЮЧАЮЩЕЕ ИЛИ»?
4. Изложите принцип линейной разделимости и пути его преодоления.
5. Какова эффективность запоминания информации персептроном?
6. Изложите принципы обучения персептрона.
7. Опишите алгоритм обучения персептрона.
8. Каким образом можно модифицировать алгоритм обучения персептрона?
9. Какие классы задач могут быть решены при помощи персептронных систем?

#### Литература

1. McCulloch W. W., Pitts W. 1943. A logical calculus of the ideas imminent in nervous activiti. Bulletin of Mathematical Biophysics 5:115-33. (Русский перевод: Маккаллок У. С., Питтс У. Логическое исчисление идей, относящихся к нервной деятельности. Автоматы. – М.: ИЛ. – 1956).
2. Minsky M. L, Papert S. 1969. Perseptrons. Cambridge, MA: MIT Press. (Русский перевод: Минский М. Л., Пейперт С. Персептроны. – М: Мир. – 1971).
3. Pitts W. Moculloch W. W. 1947. How we know universals. Bulletin of Mathematical Biophysics 9:127-47.
4. Rosenblatt F. 1962. Principles of Neurodinamics. New York: Spartan Books. (Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М: Мир. – 1965).
5. Widrow В. 1961. The speed of adaptation in adaptive control system, paper \*1933-61. American Rocket Society Guidance Control and Navigation Conference.
6. Widrow B. 1963. A statistical theory of adaptation. Adaptive control systems. New York: Pergamon Press.
7. Widrow В., Angell J. B. 1962. Reliable, trainable networks for computing and control. Aerospace Engineering 21:78-123.

Widrow В., Hoff M. E. 1960. Adaptive switching circuits. 1960 IRE WESCON Convention Record, part 4, pp. 96-104. New York: Institute of Radio Engineers.