

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ
компьютерной безопасности и
криптографии

Реферат

Постановка и возможные пути решения задачи обучения нейронных сетей

студента 5 курса 531 группы

специальности 10.05.01 «Компьютерная безопасность»

факультета компьютерных наук и информационных технологий

Минуситова Амиля Куанышкалиевича

Научный руководитель

д.ф.-м.н., доцент

подпись, дата

И. И. Слеповичев

Саратов 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Нейронные сети. Основные понятия	4
1.1 Искусственный нейрон	5
1.2 Функции активации	6
1.3 Типы архитектур нейросетей	9
2 Обучение искусственных нейронных сетей.....	12
2.1 Задачи, решаемые на основе нейронных сетей	14
2.2 Обучение с учителем.....	19
2.3 Обучение без учителя.....	24
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	29

ВВЕДЕНИЕ

Термин «нейронная сеть» появился в середине XX века. В 1943 году У. Мак-Каллок и У. Питтс в своей работе «Логическое исчисление идей, относящихся к нервной активности» описали вычислительную модель нейронной сети на основе математических алгоритмов и теоретических данных о связях внутри головного мозга. Эта модель получила название «пороговая логика» и проложила путь к созданию первых алгоритмов машинного обучения.

В современном мире нейронные сети развиваются с очень большой скоростью и в самых разных направлениях. Они применяются для распознавания образов или голоса, предсказания следующего шага, анализа и классификации входной информации по параметрам. Так как в основе идеи создания искусственных нейронных сетей (ИНС) лежат связи биологических нейронов, то в теории ИНС может выполнить любые задачи, которые может выполнить человек и даже те, которые ему не по силам, но на практике до этого ещё далеко.

Цель данной работы – разобрать основные понятия, связанные с нейронными сетями, определить задачи и способы обучения ИНС.

1 Нейронные сети. Основные понятия

Нейронные сети — это математические алгоритмы, которые являются основой современного машинного обучения. Их главная задача — находить закономерности в большом объеме данных, обрабатывать их и делать выводы.

Прообразом первой модели искусственной нейронной сети послужили данные о работе мозга человека. У. Мак-Каллок и У. Питтс сформулировали основные положения теории деятельности головного мозга. Ими были получены следующие результаты:

- разработана модель нейрона как простейшего процессорного элемента, выполняющего вычисление переходной функции от скалярного произведения вектора входных сигналов и вектора весовых коэффициентов;
- предложена конструкция сети таких элементов для выполнения логических и арифметических операций;
- сделано основополагающее предположение о том, что такая сеть способна обучаться, распознавать образы, обобщать полученную информацию.

Данная модель показала неплохие результаты для своего времени, но уже было ясно, что она не ухватывает некоторые основные аспекты человеческого интеллекта. Чтобы создать искусственный интеллект, необходимо построить систему с похожей архитектурой.

Мозг человека состоит примерно из 10^{10} нейронов, соединённых многочисленными связями друг с другом. Нейроны – это специальные клетки мозга, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут, в свою очередь, активироваться (рисунок 1). Нейрон активируется тогда, когда суммарный уровень сигналов,

пришедших в его ядро из дендритов, превысит определенный уровень (порог активации).[1]

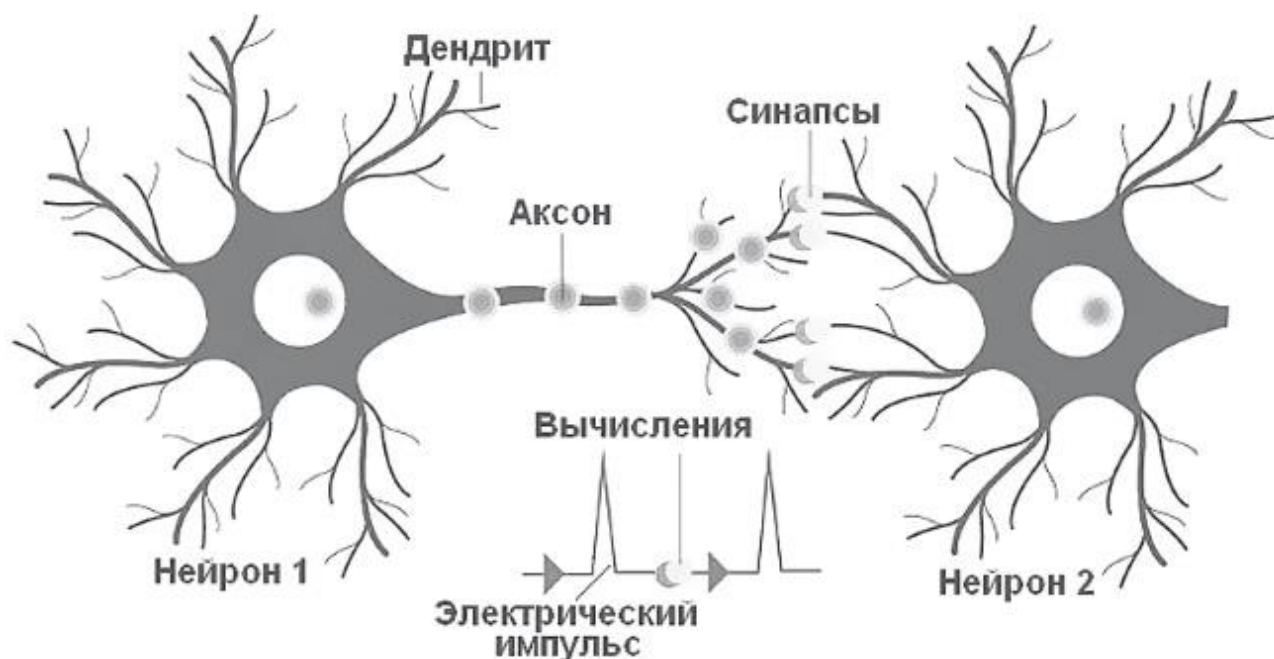


Рисунок 1 – связь нейронов[1]

Так как нейрон соединён с тысячами других нейронов, информация, поступающая в мозг в виде сигнала, обрабатывается одновременно на тысячах «микропроцессорах» и преобразуется в выходной сигнал. Цель развития этого направления – создать искусственную сеть нейронов, способную параллельно вычислять задачи разного уровня сложности с предсказуемо верным результатом.

1.1 Искусственный нейрон

Искусственный нейрон так же как его биологический аналог получает на вход некоторые сигналы (исходные данные либо выходные сигналы других нейронов нейронной сети) через входной канал. Каждый входной сигнал x_i проходит через соединение, имеющее определенную интенсивность w_i (или вес); этот вес соответствует синаптической активности биологического нейрона. С каждым нейроном связано определенное пороговое значение.

Вычисляется взвешенная сумма входов $\sum_{i=1}^n w_i x_i$, из нее вычитается пороговое значение P и в результате получается величина активации нейрона $\sum_{i=1}^n w_i x_i - P$, которая называется пост-синаптическим потенциалом нейрона (PSP). Сигнал активации преобразуется с помощью функции активации и в результате получается выходной сигнал нейрона $y = \varphi(\sum_{i=1}^n w_i x_i - P)$ (рисунок 2).

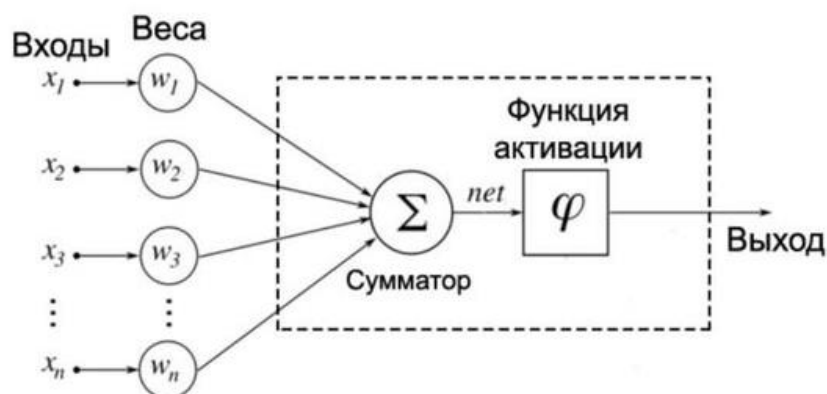


Рисунок 2 – искусственный нейрон[1]

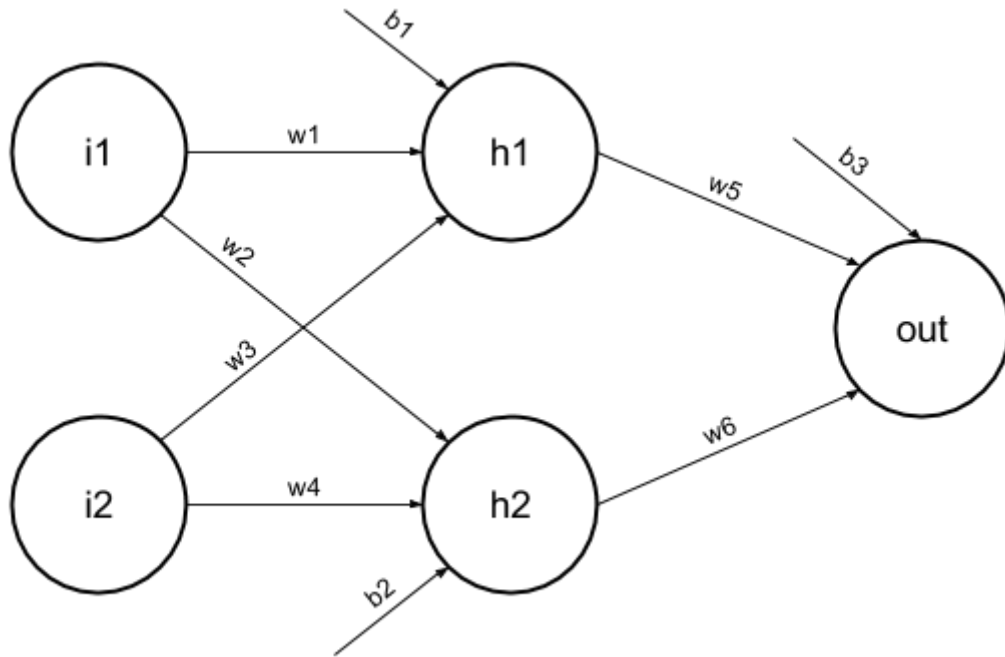
1.2 Функции активации

Просто так подавать взвешенную сумму на выход достаточно бессмысленно. Нейрон должен как-то обработать ее и сформировать адекватный выходной сигнал. Именно для этих целей и используют функцию активации. Она преобразует взвешенную сумму в какое-то число, которое и является выходом нейрона.

Для разных типов искусственных нейронов используют самые разные функции активации. В общем случае их обозначают символом $\varphi(\text{net})$. Указание взвешенного сигнала в скобках означает, что функция активации принимает взвешенную сумму как параметр.

Зачем нужна нелинейная функция активации? Нейронная сеть без функции активации – это, по сути, просто модель линейной регрессии. Функция активации выполняет нелинейное преобразование входных данных, что делает ИНС способной обучаться и выполнять более сложные задачи.

Предположим у нас есть модель нейронной сети



Скрытый слой (слой 1): $\bar{z1} = \bar{W1} * \bar{X} + \bar{b1} * \bar{a1}$, где

- $\bar{z1}$ - векторный выход первого уровня;
- $\bar{W1}$ - векторные веса, присвоенные нейронам скрытого слоя, т.е. $w1$, $w2$, $w3$ и $w4$;
- \bar{X} - векторные входные функции, т.е. $i1$ и $i2$;
- $\bar{b1}$ - векторное смещение, присвоенное нейронам в скрытом слое, т.е. $b1$ и $b2$;
- $\bar{a1}$ - векторная форма любой линейной функции.

Выходной слой (слой 2): $\bar{z2} = \bar{W2} * \bar{a1} + \bar{b2}$. Произведя несложные вычисления получим: $\bar{z2} = [\bar{W2} * \bar{W1}] * \bar{X} + [\bar{W2} * \bar{b1} + \bar{b2}]$.

Заменим $[\bar{W2} * \bar{W1}] = \bar{W}$ и $[\bar{W2} * \bar{b1} + \bar{b2}] = \bar{b}$.

Получим конечный результат: $\bar{z2} = \bar{W} * \bar{X} + \bar{b}$, который также является линейной функцией.

Следовательно, можно сделать вывод что, нелинейная функция активации позволит ИНС учиться в соответствии с разницей между ошибками.

В общем случае, выбор функции активации может быть совершенно любым. Однако при решении конкретных задач выбор может быть ограничен исходными условиями, алгоритмом обучения ИНС, вычислительными возможностями ЭВМ и некоторыми другими фактами.[2]

Таблица 1 – часто используемые функции активации.

Название	Формула	Применение	Преимущества	Недостатки
Линейная	$y = k * z$	Линейная ассоциативная память, авторегрессионные модели.	Быстрое вычисление, непрерывность, монотонность.	Применима только в некоторых типах НС.
Пороговая	$y = \text{sign}(z)$	Персептроны, автоассоциативные НС, самоорганизующиеся НС.	Быстрое вычисление, нелинейность.	Разрыв в нулевой точке не позволяет дифференцировать, из-за чего неприменимы градиентные методы обучения.
Сигмоидальная	$y = \frac{1}{1 + e^{-cz}}$	Широкое.	Нелинейность, монотонность, дифференцируемость в области определения, быстрое вычисление производной.	Сравнительно невысокая скорость вычисления.
Гиперболический тангенс	$y = \text{th}(cz) = \frac{\sin cz}{\cos cz}$	Широкое.	Нелинейность, монотонность, дифференцируемость в области определения, быстрое вычисление производной.	Медленное вычисление
Рациональная сигмоида	$y = \frac{z}{ z + c}$	Широкое.	Нелинейность, монотонность, дифференцируемость в области определения, быстрое вычисление производной.	Существенных недостатков нет.
Гауссиан	$y = \exp(-\frac{z^2}{2\sigma^2})$	Сети с радиально-базисной функцией активации.	Нелинейность, дифференцируемость, быстрое вычисление производной.	Сравнительно невысокая скорость вычисления

1.3 Типы архитектур нейросетей

Нейросети представляют собой совокупность искусственных нейронов, определенным образом соединенных между собой и с внешней средой с помощью связей, определяемых весовыми коэффициентами. Множество параллельно работающих процессоров обеспечивают высокое быстродействие ИНС.

В целом вид выполняемого сетью преобразования обусловлен не только характеристиками составляющих ее нейронов, но и особенностями архитектуры сети, а именно топологией межнейронных связей, способами обучения этой сети, наличием или отсутствием конкуренции между нейронами, направлением и способом управления и синхронизации, способом передачи информации между нейронами.

С точки зрения топологии можно выделить 3 основные типа ИНС:

- 1) Сети с прямыми связями, показаны на рисунке 3;

Простейшими из них являются однослойные перцептроны. Более сложными и мощными по своим функциональным возможностям являются многослойные НС прямого распространения, наиболее широко в настоящее время используемые на практике. В многослойных нейронных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях. Внешние входные сигналы подаются по входам нейронов входного слоя (0-й слой). Далее сигналы распространяются от входа к выходу по слоям. Выходами сети являются выходные сигналы последнего слоя. Кроме входных и выходных слоев может быть один или несколько скрытых слоев. Как правило, нейроны одного слоя имеют один тип активационной функции.

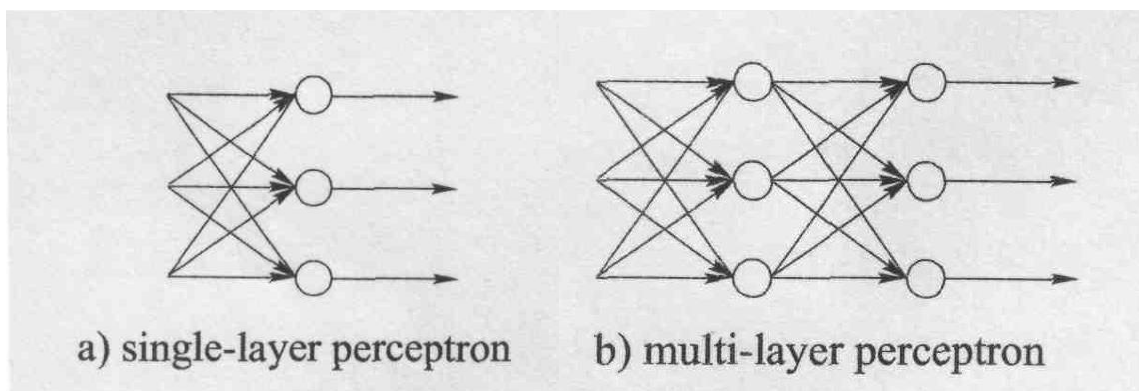


Рисунок 3 – сети с прямыми связями[3]

2) Сети с обратными связями, показаны на рисунке 4;

В сетях такой архитектуры информация с последующих слоев передается на предыдущий. Введение обратных связей в НС, с одной стороны, расширяет ее функциональные возможности, с другой стороны, может создать дополнительные проблемы (например, с устойчивостью ее функционирования). Обратными связями могут охватываться как отдельные слои, так и несколько слоев и даже вся сеть. Наиболее часто используются сети Хопфилда или Элмана.

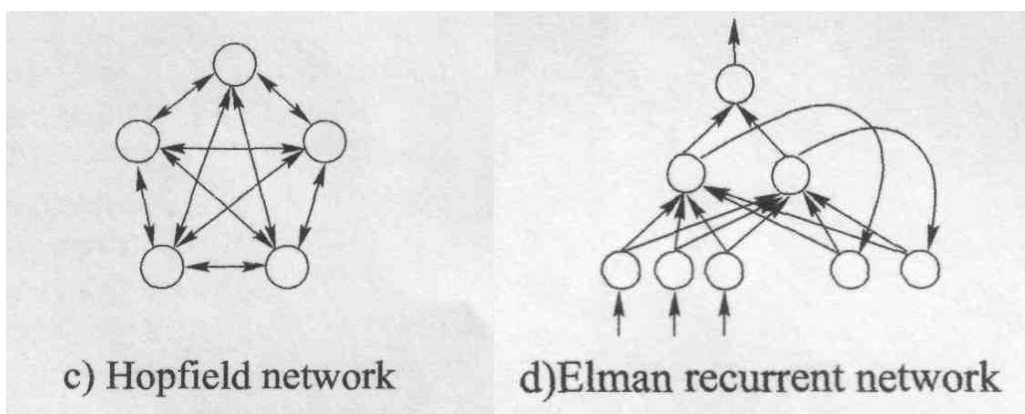


Рисунок 4 – сети с обратными связями[3]

3) Гибридные (регулярные) сети, показаны на рисунке 5.

Регулярность сети состоит в том, что каждый нейрон связан определенным образом со своими соседями. Как правило, нейроны располагаются в узлах некоторой регулярной решетки (сетки), и кроме внешних входных сигналов на каждый нейрон поступают выходные сигналы в некотором смысле соседних нейронов. Понятие соседства зависит от типа используемой решетки.[3]

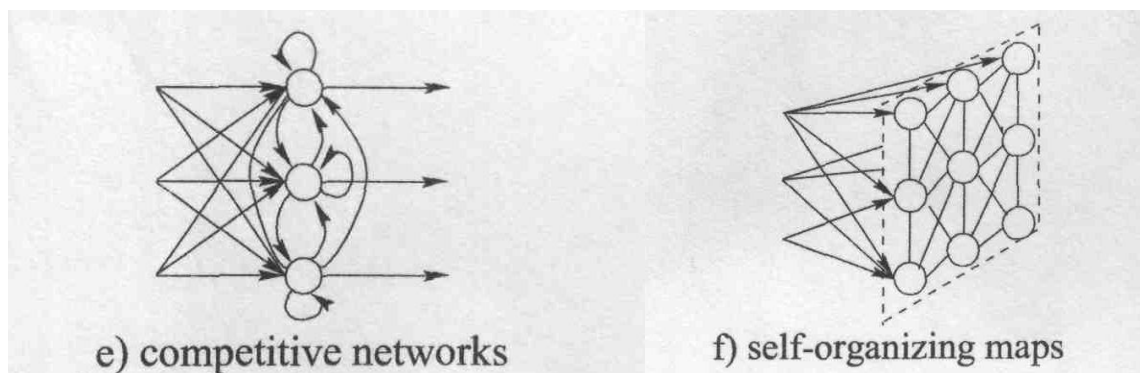


Рисунок 5 – гибридные (регулярные) сети[3]

2 Обучение искусственных нейронных сетей

Процесс функционирования нейронной сети – это набор действий, которые она способна выполнять, зависящий от величин синаптических связей. Поэтому, задавшись определенной структурой сети, соответствующей какой-либо задаче, необходимо найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением нейронной сети, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время функционирования.

Существуют три парадигмы обучения: "с учителем", "без учителя" и смешанная.

1) В первом случае нейронная сеть располагает правильными ответами (выходами) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода.

2) Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям.

3) При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, сложность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут

быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению.

Известны 4 основных типа правил обучения:

1) Правило коррекции по ошибке.

При обучении с учителем для каждого входного примера задан желаемый выход Y . Реальный выход сети y может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала $(P-Y)$ для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда персептрон ошибается.

2) Обучение Больцмана.

Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

3) Правило Хебба.

Самым старым обучающим правилом является постулат обучения Хебба. Хебб опирался на следующие нейрофизиологические наблюдения: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом.

4) Обучение методом соревнования.

В отличие от обучения Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно, как правило "победитель берет всё". Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования

позволяет кластеризовать входные данные: подобные примеры группируются в сеть в соответствии с корреляциями и представляются одним элементом. При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счет такого изменения сохраненного в сети вектора весов связей победившего нейрона, при котором он становится чуть ближе к входному примеру.[4]

2.1 Задачи, решаемые на основе нейронных сетей

Нейронная сеть, прежде всего, является универсальным аппроксиматором нелинейной векторной функции нескольких переменных $Y = \varphi(X)$, где X - входной и Y - выходной вектор. Постановка многих практических задач может быть сведена к именно к аппроксимационному представлению. Необходимо построить такое отображение $\varphi: X \rightarrow Y$, чтобы на каждый возможный входной сигнал формировался правильный выходной сигнал в соответствии:

- а) со всеми примерами обучающей выборки;
- б) со всеми возможными входными сигналами, которые не вошли в обучающую выборку (что значительно сложнее).

Именно поэтому, нейросети находят широкое применение при решении многих проблем, к которым относятся: аппроксимация и интерполяция, распознавание и классификация образов и изображений, сжатие данных, прогнозирование, идентификация, управление, ассоциативная память и многие другие. Рассмотрим самые распространённые из них.

Задача *классификации* – это крупнейшая задача машинного обучения в целом, которая ставит своей целью назначить метку класса наблюдениям из предметной области.

В этом случае X – некоторое представление образа, а Y – номер класса, к которому принадлежит этот образ. Пусть имеется набор данных, каждый из которых представляет вектор X . На входы сети подаются поочередно наборы данных. То есть каждый вход соответствует компоненте входного вектора. Каждый выход нейросети соответствует своему классу.

Алгоритмы классификации оцениваются на основе их результатов. Точность измерений (Accuracy) – это популярный показатель, используемый для оценки производительности модели на основе предсказанных меток классов. Точность классификации не идеальна, но это хорошая отправная точка для многих задач классификации.[5]

Существует 4 основных типа классификации:

1. Бинарная классификация;

Этот тип классификации предполагает разделение набора данных на две категории. Это означает, что выходная переменная может принимать только два значения, как показано на рисунке 6.

Популярные алгоритмы, которые можно использовать для двоичной классификации:

- Логистическая регрессия;
- Метод К-ближайших соседей;
- Дерево решений;
- Метод опорных векторов;
- Наивный байесовский классификатор.

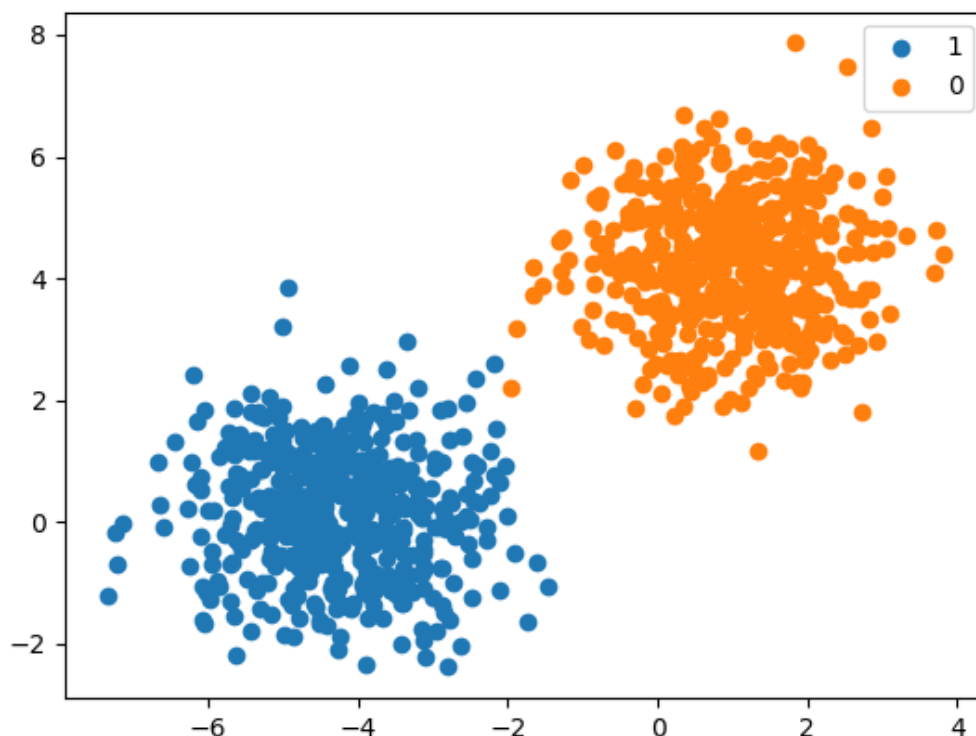


Рисунок 6 – пример бинарной классификации[Листинг А]

2. Многоклассовая классификация;

Данная классификация предполагает, что классов более двух. Например

- Классификация лиц;
- Классификация видов растений;
- Оптическое распознавание символов.

В отличие от бинарной классификации, мультиклассовая классификация (рисунок 7) не имеет понятия нормальных и аномальных исходов. Вместо этого примеры классифицируются как принадлежащие к одному из ряда известных классов.

Многие алгоритмы, используемые для двоичной классификации, могут использоваться и для мультиклассовой:

- Метод k-ближайших соседей;
- Дерево решений;
- Наивный байесовский классификатор;
- Случайный лес;
- Градиентный бустинг.

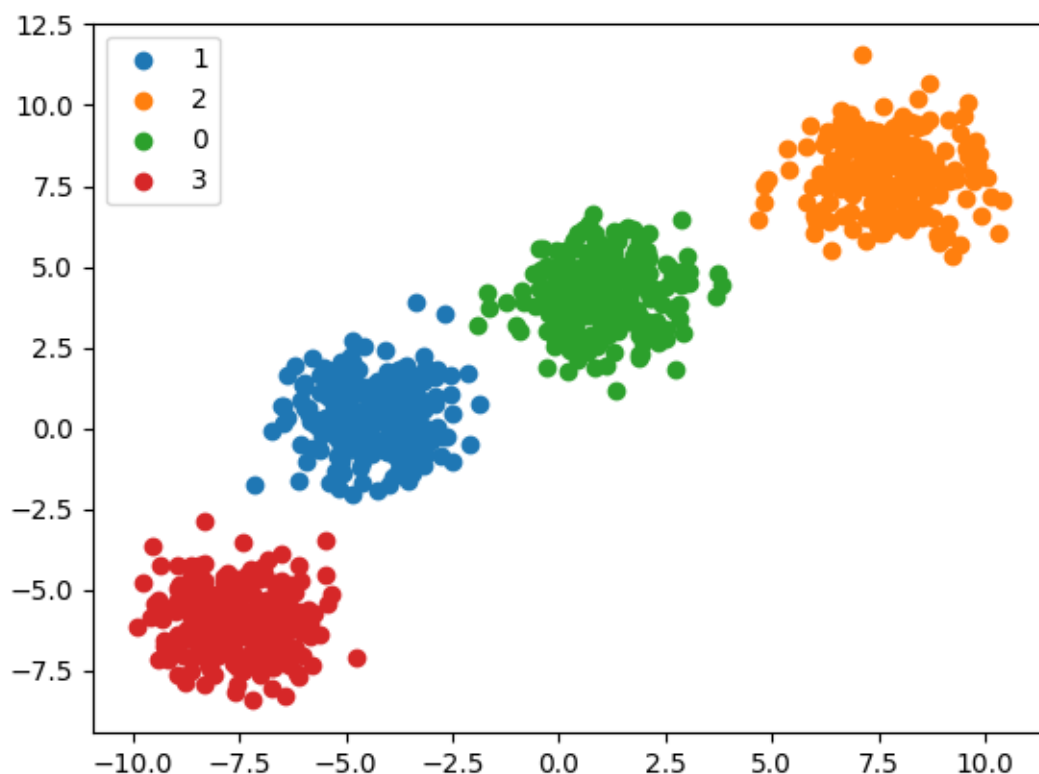


Рисунок 7 – пример многоклассовой классификации[Листинг А]

3. Классификация по нескольким меткам;

Такая классификация полагает, что имеется более двух меток классов.

Рассмотрим пример классификации объектов на фотографии, причем изображение может иметь несколько объектов, таких как «велосипед», «яблоко», «человек» и т.д. Это не похоже на двоичную и мультиклассовую классификации, где для каждого примера прогнозируется одна метка класса.

По сути, это модель, которая делает несколько прогнозов двоичной классификации для каждого примера.

Алгоритмы классификации, используемые для двоичной или мультиклассовой классификации, не могут использоваться напрямую для классификации по нескольким меткам. Могут использоваться специализированные версии стандартных алгоритмов:

- Деревья решений с несколькими ярлыками
- Случайные леса с несколькими ярлыками
- Градиентный бустинг с несколькими ярлыками

4. Несбалансированная классификация.

Несбалансированная классификация относится к задачам классификации, в которых количество примеров в классах распределяется неравномерно.

Как правило, такие задачи представляют собой задачи двоичной классификации, где большинство примеров в наборе обучающих данных относятся к нормальному классу, а меньшая часть примеров относится к ненормальному классу (рисунок 98. Примеры:

- Обнаружение мошеннических операций
- Обнаружение выбросов
- Медицинские диагностические тесты

Эти проблемы моделируются как задачи бинарной классификации, хотя могут потребоваться специальные методы.

Примеры алгоритмов:

- Случайное недосэмплирование;

- Алгоритм SMOTE;
- Логистическая регрессия с учетом затрат;
- Деревья принятия решений с учетом затрат;
- Чувствительные к стоимости машины опорных векторов.

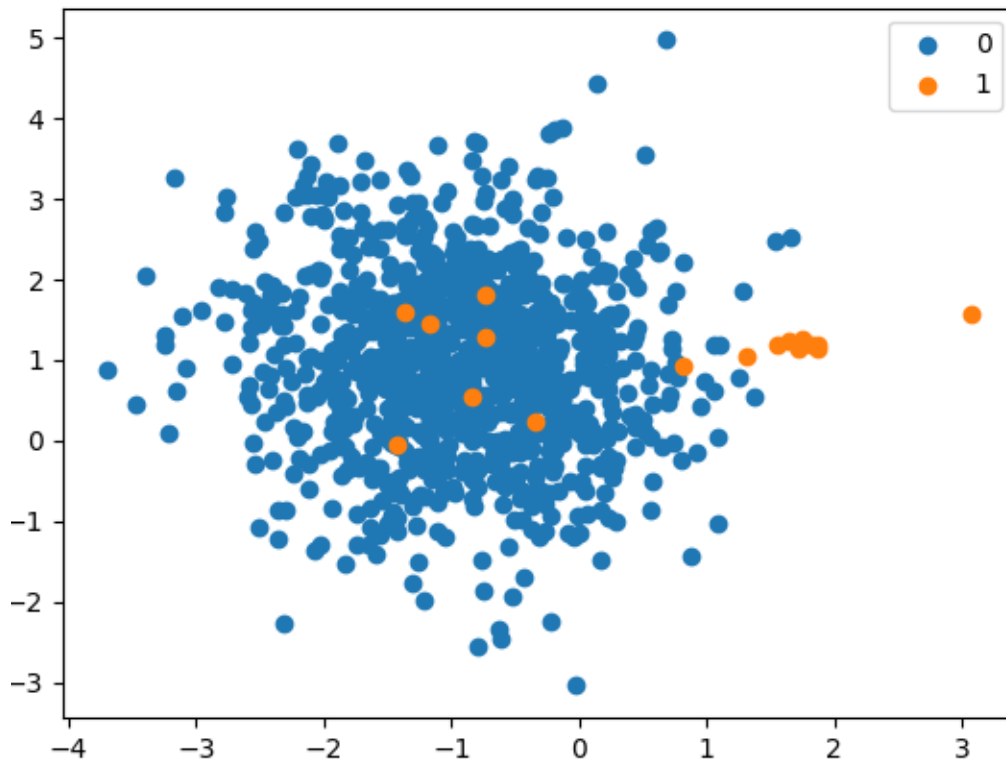


Рисунок 8 – пример несбалансированной классификации[Листинг А]

Мы можем видеть один основной кластер для примеров, принадлежащих классу 0, и несколько разрозненных примеров, принадлежащих классу 1.[6]

Реализация программы представлена в листинге А.

В задачах *прогнозирования* в качестве входных сигналов X используются временные ряды, представляющие значения контролируемых параметров на некотором интервале времени. В качестве выходных сигналов Y обычно выступает некоторое множество переменных, являющееся подмножеством входных сигналов. Прогноз будущих значений переменной осуществляется на основе её предыдущих значений. Прогнозирование временных рядов часто сводится к задаче регрессии.

Применяются два основных вида прогнозирования:

1) Многошаговый прогноз, используемый при необходимости долгосрочного прогноза, цель которого состоит в идентификации общих тенденций и наиболее важных поворотных точек во временном ряду. При этом для предсказания временного ряда в фиксированный период используется конфигурация текущих значений. Затем прогноз снова вводится в сеть для прогнозирования следующего периода.

2) Одношаговый прогноз, где в отличие от предыдущего случая повторно вводить спрогнозированные ранее данные не требуется. НС предсказывает значение ряда на какой-либо момент времени и для каждого последующего прогнозного значения использует фактическую величину, а не значение, которое было предсказано ранее.

Фильтрация используется для извлечения полезной информации из набора зашумленных данных. Шумы возникают по разным причинам, например, при передаче информационных сигналов через зашумленный канал или в результате измерений с помехами. Фильтры чаще всего применяются при решении следующих задач:

- Фильтрация – извлечение полезной информации в дискретные моменты времени t из данных, полученных из данных до момента t включительно;
- Сглаживание – отличается от предыдущей обработки тем, что есть возможность использовать данные, полученные не только до момента t , но и после него.[5]

2.2 Обучение с учителем

При обучении с учителем нейронная сеть располагает правильными ответами (выходами сети) y^M на каждый входной пример X^M . Вместе они называются обучающей парой $\{(X^M, y^M)\}$. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам, тем самым минимизировалась ошибка. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса

подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемого уровня.

Общий алгоритм обучения ИНС с учителем выглядит так.

На вход подается множество входных нейронов X .

Определяется функция ошибки E . Обычно это средняя квадратичная ошибка(MSE), но есть и другие варианты такие как: средняя абсолютная ошибка(MAE), средняя абсолютная процентная ошибка(MAPE) и т.д.

$$E = \frac{1}{p} \sum_{i=1}^p (d_i - y_i)^2$$

Где, P — количество обработанных нейронной сетью примеров; y_i — выход; d_i — желаемый (идеальный) выход нейронной сети.

Процедура обучения нейронной сети сводится к процедуре коррекции весов связей. Целью процедуры коррекции весов есть минимизация функции ошибки E .

1) Перед началом обучения весовые коэффициенты устанавливаются случайным образом

2) На первом этапе на вход в определенном порядке подаются учебные примеры. На каждой итерации вычисляется ошибка для учебного примера E_L (ошибка обучения) и по определенному алгоритму производится коррекция весов. Целью процедуры коррекции весов есть минимизация ошибки E_L .

3) На втором этапе обучения производится проверка правильности работы. На вход в определенном порядке подаются контрольные примеры. На каждой итерации вычисляется ошибка для контрольного примера E_G (ошибка обобщения). Если результат неудовлетворительный то, производится модификация множества учебных примеров и повторение цикла обучения.

Если после нескольких итераций алгоритма обучения ошибка обучения E_L падает почти до нуля, в то время как ошибка обобщения E_G в начале падает а затем начинает расти, то это признак эффекта переобучения, который отображён на рисунке 9. В этом случае обучение необходимо прекратить.[7]

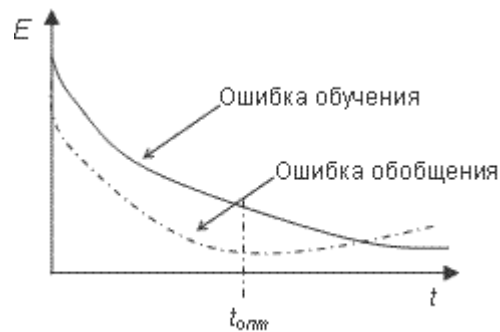


Рисунок 9 – переобучение НС[7]

Метод Розенблатта.

Данный метод был предложен Ф.Розенблаттом в 60-х годах XX века Для нейронной сети, названной персептрон. Персептрон имеет пороговую функцию активации, его схема представлена на рисунке 10.

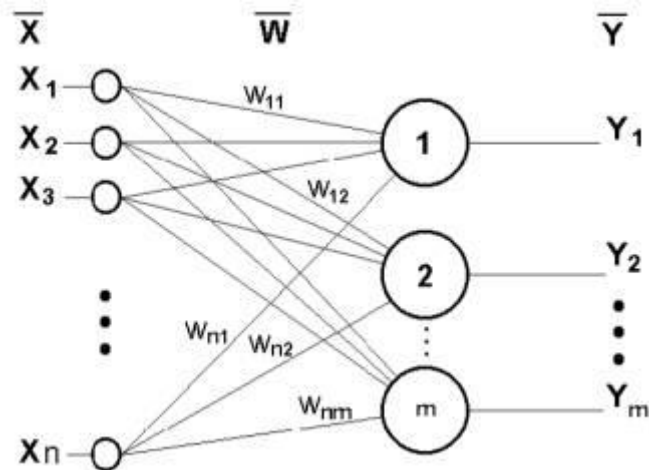


Рисунок 10 – однослойный персептрон[8]

Процедуру обучения Розенблатта для однослойного персептрона можно представить так:

$$w_{ij}(t + 1) = w_{ij}(t) + \alpha x_i d_j$$

Где, x_i – i -й вход НС; d_j – желаемый (идеальный) j -й выход НС; α – скорость обучения.

Алгоритм обучения Розенблатта строится следующим образом:

- 1) На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения;
- 2) На вход ИНС подаётся учебный пример;
- 3) Если выход y_i не совпадает с идеальным выходом d_j , то происходит то происходит модификация весов по исходной формуле;
- 4) Цикл с шага 2 пока выход y_i не совпадает с идеальным выходом d_j или весовые коэффициенты не перестанут меняться

Весовые коэффициенты меняются только в том случае, если реальное выходное значение не совпадает идеальным выходным значением.

Метод обратного распространения ошибки.

Впервые метод был описан в 1974 г. А. И. Галушкиным, а также независимо и одновременно Полом Дж. Вербосом. Вариант решения обучения многослойной сети предложил Д. Румельхарт в 1986 году. Основная идея метода состояла в распространение сигналов ошибки от выходов НС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

В данном методе минимизация функции ошибки E ведётся методом градиентного спуска. Подстройка весовых коэффициентов происходит следующим образом:

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} \quad (1)$$

Где, w_{ij} – весовой коэффициент синаптической связи, соединяющей i -й и j -й узлы НС; $\alpha \in (0,1)$ – коэффициент скорости обучения.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} * \frac{\partial y_j}{\partial s_j} * \frac{\partial s_j}{\partial w_{ij}}$$

Где, y_j – выход нейрона j ; s_j – взвешенная сумма его входных сигналов, то есть аргумент активационной функции.

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{\partial y_k}{\partial s_k} * \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{\partial y_k}{\partial s_k} * w_{jk}^{(n+1)}$$

Где, k – число нейронов слоя $n + 1$.

Введя новую переменную

$$\delta_j^{[n]} = \frac{\partial E}{\partial y_j} * \frac{\partial y_j}{\partial s_j}$$

получим рекурсивную формулу для расчетов величин $\delta_j^{[n]}$ слоя из величин $\delta_j^{[n+1]}$ слоя $n+1$

$$\delta_j^{[n]} = \frac{\partial E}{\partial y_j} * \sum_k \delta_k^{[n+1]} w_{jk}^{[n+1]} \quad (2)$$

Для выходного слоя:

$$\delta_i^{[n]} = \frac{\partial y_i}{\partial s_i} * (y_i^{[N]} - d_i) \quad (3)$$

Теперь мы можем записать (1) в раскрытом виде:

$$\Delta w_{ij} = -h * \delta_j^{[n]} * y_i^{[n-1]} \quad (4)$$

Алгоритм обучения НС с помощью процедуры обратного распространения ошибки строится следующим образом:

- 1) На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения;
- 2) На вход НС подаётся учебный пример. Для каждого примера:

2.1) Фаза прямого распространения сигнала: Вычислить выход каждого нейрона НС.

2.2) Фаза обратного распространения сигнала:

Для выходного слоя вычислить изменения весов $\Delta w^{[N]}$ по (4) и (3).

Для всех остальных слоев вычислить $\Delta w^{[n]}, n = (N - 1), \dots, 1$ по (4) и (2)

2.3) Скорректировать все веса НС:

$$w_{ij}^{[n]}(t) = w_{ij}^{[n]}(t - 1) + \Delta w_{ij}^{[n]}(t)$$

3) Если ошибка НС существенна, перейти на шаг 2, иначе - конец работы.[8]

2.3 Обучение без учителя

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Обучение без учителя является намного более правдоподобной моделью обучения для биологической системы. Развита Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с predetermined идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения сравнивает статистические свойства обучающего множества и группирует схожие векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Следовательно, выходы подобной сети должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является серьезной проблемой.

Обычно не сложно идентифицировать связь между входом и выходом, установленную сетью.

Главная черта, делающая обучение без учителя привлекательным, это его "самостоятельность". Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Настройка синапсов может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Исходя из этого построены методы обучения Хебба.

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу:

$$w_{ij}(t) = w_{ij}(t - 1) + \alpha y_i^{(n-1)} y_j^{(n)}$$

Где $y_i^{(n-1)}$ – выходное значение нейрона i слоя $(n-1)$, $y_j^{(n)}$ – выходное значение нейрона j слоя n ; $w_{ij}(t)$ и $w_{ij}(t - 1)$ – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях t и $t-1$ соответственно; коэффициент α – скорость обучения. При обучении данным методом усиливаются связи между возбужденными нейронами.

Существует также и *дифференциальный метод обучения Хебба*:

$$w_{ij}(t) = w_{ij}(t - 1) + \alpha [y_i^{(n-1)}(t) - y_i^{(n-1)}(t - 1)] * [y_j^{(n)}(t) - y_j^{(n)}(t - 1)]$$

Здесь $y_i^{(n-1)}(t)$ и $y_i^{(n-1)}(t - 1)$ – выходное значение нейрона i слоя $n-1$ соответственно на итерациях t и $t-1$; $y_j^{(n)}(t)$ и $y_j^{(n)}(t - 1)$ – то же самое для нейрона j слоя n . Как видно из формулы сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть так:

4) На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.

5) На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямых сетей, то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется функция активации нейрона, в результате чего получается его выходное значение $y_i^{(n)}$, $i = 0, \dots, M_i - 1$, где M_i – число нейронов в слое i ; $n=0, \dots, N-1$, а N число слоев в сети.

6) На основании полученных выходных значений нейронов по формулам Хебба производится изменение весовых коэффициентов.

7) Цикл с шага 2, пока выходные значения сети не стабилизируются с заданной точностью.

Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

Другой метод обучения без учителя *алгоритм Кохонена* предусматривает подстройку синапсов на основании их значений от предыдущей итерации.

$$w_{ij}(t) = w_{ij}(t - 1) + \alpha[y_i^{(n-1)} - w_{ij}(t - 1)]$$

Обучение, в таком случае, сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $y_i^{(n-1)}$ и весовыми коэффициентами его синапсов.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ, и подстройка весов проводится только для него. Эта, так называемая, аккредитация может

сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение.[9]

ЗАКЛЮЧЕНИЕ

Работа Мак-Каллока и Питтса положила начало науки о нейросетях, которая на сегодняшний день развивается с очень большой скоростью и задействована во многих сферах деятельности человека.

Существуют три парадигмы обучения ИНС: "с учителем", "без учителя" и смешанная. Обучение «с учителем» основывается на наличие множества учебных примеров, используемых для "настройки" сети, и второе множество контрольных примеров, для оценки качества работы сети. В 60х годах Розенблатт предложил метод обучения итерационной подстройкой матрицы весов, последовательно уменьшающей ошибку на выходе. Метод обратного распространения ошибки, предложенный Румельхартом, заключается в распространении сигналов ошибки от выходов сети к её входам.

Процесс обучения «без учителя», заключается в настройке весов синапсов, на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации. При обучении методом Хебба усиливаются связи между возбужденными нейронами.

В данной работе были рассмотрены основные понятия, связанные с функционированием нейронных сетей, методы и алгоритмы обучения ИНС.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Нейронные сети. Параллели из биологии [Электронный ресурс]
URL: <https://studfile.net/preview/7146498/page:26/> (дата обращения: 23.12.2022).
Загл. с экрана. Яз. рус.
- 2 Activation functions in Neural Networks [Электронный ресурс] URL:
<https://www.geeksforgeeks.org/activation-functions-neural-networks/> (дата
обращения: 23.12.2022). Загл. с экрана. Яз. англ.
- 3 Нейросетевые технологии. Основные типы архитектуры нейронных
сетей. [Электронный ресурс] URL: <https://studfile.net/preview/5683134/page:2/>
(дата обращения: 23.12.2022). Загл. с экрана. Яз. рус.
- 4 Анил К. Дж. Искусственные нейронные сети. [Электронный ресурс]
URL: <https://www.scorcher.ru/neuro/science/neurocomp/mem52.php> (дата
обращения: 23.12.2022). Загл. с экрана. Яз. рус.
- 5 Нейросетевые технологии. Основные задачи, решаемые с
применением нейронных сетей. [Электронный ресурс] URL:
<https://studfile.net/preview/5683134/page:9/> (дата обращения: 23.12.2022). Загл. с
экрана. Яз. рус.
- 6 Классификация (Classification). [Электронный ресурс]
URL: <https://www.helenkapatsa.ru/klassifikatsiia/> (дата обращения: 23.12.2022).
Загл. с экрана. Яз. рус.
- 7 Золотухина И.А. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ. Методы и алгоритмы обучения нейронных сетей.
[Электронный ресурс] URL:
http://www.rusnauka.com/11_NPE_2014/Informatica/2_167276.doc.htm (дата
обращения: 24.12.2022). Загл. с экрана. Яз. рус.
- 8 Нейронные сети : обучение с учителем [Электронный ресурс] URL:
<https://scorcher.ru/neuro/science/perceptron/mem32.php> (дата обращения:
23.12.2022). Загл. с экрана. Яз. рус.

9 Обучение НС с учителем. Обучение искусственных нейронных сетей. [Электронный ресурс] URL: <https://studfile.net/preview/7466650/page:11/> (дата обращения: 24.12.2022). Загл. с экрана. Яз. рус.

Листинг А

```
import numpy

from numpy import where

import collections

from collections import Counter

import sklearn

from sklearn.datasets import make_blobs, make_multilabel_classification, make_classification

import matplotlib

from matplotlib import pyplot


#n, k = int(input('Введите кол-во объектов\t')), int(input('Введите кол-во классов\t'))

#X, y = make_blobs(n_samples=n, centers=k, random_state=3)

#X, y = make_multilabel_classification(n_samples = 1000, n_features = 2, n_classes = 2, n_labels =
2, random_state = 2)

X, y = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_classes=2, n_clusters_per_class=1, weights=[0.99,0.01], random_state=2)


counter = Counter(y)


for label, _ in counter.items():

    row_ix = where(y == label)[0]

    pyplot.scatter(X[row_ix, 0], X[row_ix, 1], label=str(label))

pyplot.legend()

pyplot.show()
```