МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**ИСПОЛЬЗОВАНИЕ ИНС ДЛЯ РАБОТЫ С ЦИФРАМИ**

Работу выполнила\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.А. Гаврилова

(подпись)

Направление 01.03.02 «Прикладная математика и информатика»

Направленность (профиль) «Системное программирование и компьютерные технологии» (Математическое и программное обеспечение вычислительных машин)

Научный руководитель

канд. техн. наук, доц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А.Полупанов

(подпись)

Нормоконтролер

ст.преп.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В.Харченко

(подпись)

Краснодар

2018

**РЕФЕРАТ**

Курсовая работа 25ст., 13 рис., 10 источников.

ИНС, РАСПОЗНАВАНИЕ ЦИФР, НЕЙРОННАЯ СЕТЬ, ПРОЦЕСС ОБУЧЕНИЯ, НЕЙРОН.

Объектом изучения является искусственная нейронная сеть, алгоритмы распознания образов.

Цель работы: разобрать задачу распознания рукописных цифр для ИНС

В курсовой работе были изложены основные проблемы, связанные с данной темой и рассмотрены различные методы для решения данной проблемы. Были изучены основные аспекты ИНС, а также составлен план будущей программы.

СОДЕРЖАНИЕ

Введение……………………………………………………………………..……4

1 Постановка задачи……………………………………………………………...5

2 Нейронные сети…………………………………………………………………6

2.1 Краткая история ИНС………...……………………………...…...…....6

2.1.1 Недостатки ИНС……………………..……………………..…7

2.2 Перцептроны………….………………………………………………..8

2.3 Процесс обучения ИНС………………………..……………...……...10

2.3.1 Сбор данных для ИНС…………………………………....….11

2.4 Архитектура нейронных сетей…………………………….……..…..13

3 Структура сети………………………………………………...……………….17

3.1 Распознавание образов………………….…….………………………17

3.2 Реализация ИНС……………………………..……………………......17

3.3 Обратное распространение……………………….……….………….19

3.4 Функция потерь…………………………………………..…………...19

4 Распознавание цифр с помощью ИНС…………………………………..…...21

4.1 Разбор поставленной задачи………………………………….……....21

Заключение……………………………………………………………………….24

Список использованных источников…………………………………………...25

**ВВЕДЕНИЕ**

Искусственные нейронные сети появились уже давно, но с каждым годом они набирают все большую и большую популярность. Их используют практически во всех современных сферах, будь то медицина, аналитика или бизнес. Конечно, во многом это связанно с развитием технологий, но и также несколькими причинами:

ИНС легки для понимания и просты в использовании. Пользователю не нужно иметь глубокие знания, например, в сфере аналитики, как при написании программ для статистики. Разумеется, человеку нужно знать как работать с данными для нейронной сети, как правильно подбирать их и иметь представление о самой структуре искусственной сети. Однако, несмотря на это ИНС все же находится в более выигрышном положении.

У нейронных сетей большой потенциал благодаря своей нелинейности. Это свойство вновь делает нейронные сети более выгодными в использовании, нежели обычные программы. Во много это связано с тем, что у ИНС нет такой проблемы как проблема размерности.

Как не странно, но даже для восприятия человеком нейронные сети кажутся более удобными, ведь за их основы взяты нейроны головного мозга человека, что и делает их более привлекательными для использования.

Распознание образов ИНС несомненно актуально сейчас, так как с помощью таких сетей возможно развитие искусственного интеллекта, создание более надежных программ выявления ботов и т.д.

**1 Постановка задачи**

В рамках выполнения курсовой работы были поставлены следующие задачи:

- рассмотрение истории ИНС и параллели из биологии;

- разбор структуры нейронных сетей;

- рассмотрение существующих архитектур ИНС;

- разработка алгоритма построения нейронной сети для распознания цифр.

**2 Нейронные сети**

**2.1 Краткая история ИНС**

Идеи о искусственного интеллекта появились уже очень давно. Впервые задатки чего-то приблизительно похожего на ИТ было разработано в середине прошлого века, когда возникла идея создать подобие головного мозга человека. По сути, люди предложили рассмотреть нейроны мозга, множество маленьких компьютеров, каждый из которых выполнял бы определенные задачи [5].

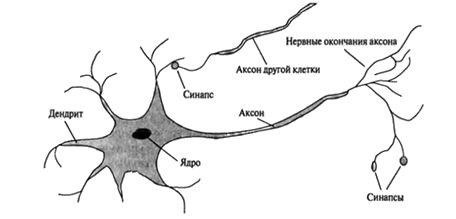


Рисунок 1 – Пример нейрона головного мозга

Теперь немного биологии. Нейрон – это клетка, и у каждого человека огромное количество таких клеток. Благодаря этим клеткам мозг способен перерабатывать полученную информацию так, чтобы в конечном счете человек мог распознавать образы без ошибок. Чтобы немного лучше понять строение нейрона, обратите внимание на изображение выше. Нейронные сети головного мозга представляют собой некоторую совокупность всех нейронов. Каждый нейрон работает по аналогичному с другими нейронами алгоритму. Чтобы биологическая нейронная сеть работала и давала различные результаты, а не всегда один и тот же, существует синапс.

Синапс – это место стыка выхода одного нейрона и входа другого. Каждый синапс индивидуален, из-за чего и получаются разные выходные данные[9].

Если добавить в это математику, то можем получить некую нейронную сеть, и тогда уже нейрон следует рассматривать не как клетку, а скорее, как функцию. У каждого нейрона есть потенциал, который можно рассмотреть, как функцию от потенциалов синапсов его дендритов. Допустим, что нейрон принимает может принимать два состояния: возбужденное и невозбужденное. Эти состояния будем использовать как указатели, то есть, нейрон вычисляет линейную комбинацию потенциалов синапсов, а затем сравнивает полученные результаты с ее пороговым значением, и если этот результат больше порога, то нейрон переходит в состояние возбужденное.

Можно сказать, что мозг каждый раз при получении каких-либо данных занимается решением некоторой векторной функции. И полученные значения могут изменяться со временем, такое явление назвали обучением, которое способствует тому, что в конечном итоге при обучении ИНС, будет получена практически идеальная копия мозга человека.

В следствии развития ИНС появилось такое понятия как машинное обучение. Машинное обучение – это довольно широкая область, изучающая различные способы построения обучающих алгоритмов.

**2.1.1 Недостатки ИНС**

У нейронных сетей существует ряд недостатков, из-за которых они до сих пор полностью не вытеснили привычные алгоритмы решения каких-либо задач.

Один из недостатков – это то, что полученное решение всегда приблизительно. То сеть, если ИНС поставлена задача определить какой-то символ, то она выдаст наиболее подходящий результат в процентом соотношении.

Также у ИНС существуют проблемы, которые возникают в ходе ее обучения. В основном это связано с подбором обучающих примеров. Нередко обучение может привести к тупиковым ситуациям. Из-за процесса обучения пока нельзя применять ИНС для систем реального времени.

**2.2 Перцептроны**

Чтобы лучше понять работу нейронных сетей, следует разобраться с тем, что такое перцептрон. Они появились еще в середине прошлого века и разработал их ученный Фрэнк Розенблатт. Рассмотрим, что представляют собой перцептроны [6].

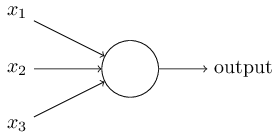


Рисунок 2 – Перцептрон

В примере, показанном на рисунке 1, у перцептрона имеется 3 бинарных вхождения x1, x2, x3. В зависимости от задачи вхождений может быть гораздо больше или же очень мало. Розенблатт ввел простое правило, по которому получался на выходе результат. Он ввел такое понятие как веса, w1, w2, …, действительные числа, которые представляют собой отношение между входящими и выходящими данными. Выходные данные нейрона могут принимать значения 1 или 0, в зависимости от разницы сумы от порога. Порог (threshold), так же как и веса, представляет собой действительное число, которое является параметром нейрона. Для большей ясности, зависимость выходных данных от весов и входных данных рассмотрим формулу (1):

(1)

Так и работают перцептроны. Но это лишь базовая модель, в применении они принимать какие-либо решения, взвешивая имеющиеся данные. Разберем такую возможность на простом примере.

Пусть у нас имеется 3 фактора, на которые будет опираться нейронная сеть:

1. Хорошая ли погода?
2. Есть ли в компании знакомый?
3. Насколько далеко находит мероприятие?

Эти факторы будут представлены как три бинарных входных

значения: x1, x2 и x3. Допустим, x1 = 1 в случае хорошей погоды, иначе x1 = 0. Тоже самое для x2 = 1, если присутствует знакомый и x2 = 0 если нет. Аналогично для x3.

Итак, предположим, что факторы имеет разную значимость, то есть разные веса. Пусть w1 = 6 для погоды, w2 = 2 и w3 = 2 для остальных.

С помощью порога и весов, можно получить разные модели для прогнозирования результата. Например, если порог будет равняться 3, то тогда перцептрон будет решать, поехать на мероприятие или нет, уже учитываю и погоду, и остальные факторы.

Разумеется, перцептрон не является точной копией модели поведения человека, но представленный ниже рисунок 10 как раз иллюстрирует, как перцептроны взвешивают различные данные для получения результата.

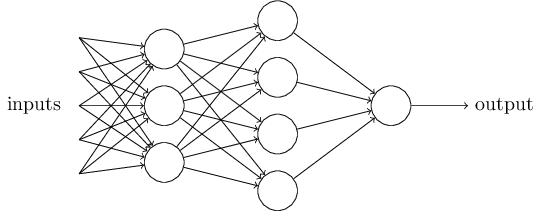
****

Рисунок 3 – Трехслойная сеть

На рисунке 3 видно, что сеть уже состоит из нескольких связанных слоев. В данной сети, первая часть – это то, что называется первым слоем. Далее следуют слои, которые получают какой-либо результат, основываясь на результате, полученном в предыдущем слое.

Для работы нейронной сети существует так называемый процесс обучения, который может настроить веса и смещения сети. С помощью обучения ИНС может научиться находить проблемы и решать их.

**2.3 Процесс обучения ИНС**

Будем рассматривать процесс обучения на примере перцептронов [7]. Для того, чтобы проверить обучилась ли сеть, необходимо после пробега сетью алгоритма обучения и выдачи результат, поменять значения весов. Если эти значения немного изменить, то результат приблизиться к нужному. Проблема заключается в том, что если в ИНС, состоящей из перцептронов, сделать даже небольшие изменения, может случиться так, что одно значение будет верным, в то время как остальные могу начать вычисляться абсолютно неверно. В этом случае вводится сигмоидная функция:

(2)

Сигмоида – это логическая функция, служащая часто для сглаживания значений. В случае с нейронной сетью, применение такой функции позволить производить небольшие изменения в весах и смещения и не вызывать при этом сильные расхождения в результатах.

**2.3.1 Сбор данных для нейронной сети**

Как же решить поставленную задачу с помощью ИНС? Прежде всего необходимо собрать данные, которые сеть будет анализировать, причем подобрать удачные данные не всегда удается сразу, так как зачастую не понятно какие именно материалы лучше подойдут для восприятия и анализа сети. Именно решение такого рода проблемы часто становится одной из сложнейших. Также проблемой может быть решить какое количество таких данных необходимо для удачного обучения сети, ведь если их будет недостаточно, то в конечном счете сеть не будет работать корректно, так как обучение не удачно, то же самое может случиться если задать слишком много данных. Решить эту проблему можно путем подбора данных, то есть, запускать алгоритм для разного количества переменных. Сами же данные разработчик может выбирать, ссылаясь на личный опыт [5].

Прежде чем отправить данные для анализа в ИНС их необходимо привести к виду, который сеть сможет прочитать. Для обучения сети существует такое понятие как обучающий вектор, который состоит из одного значения на вход сети и, обычно, одно значение на выход сети. При первых попытках обучения сети редко получается добиться идеального результата, приходится либо несколько раз запускать алгоритм обучения, либо пытаться улучшить сеть.

**2.4 Архитектура нейронных сетей**

Сеть, как правило, состоит из входного и выходного слоев, между которыми расположены скрытые слои, количество которых зависит от самого разработчика. Наглядный пример сети, состоящий из более чем одного скрытого слоя представлен на рисунке 4 [2].

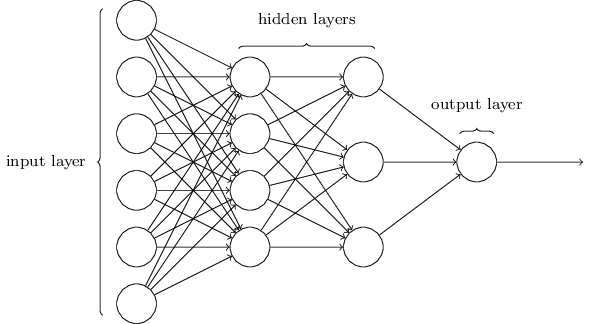


Рисунок 4 – Пример сети

За все время существования ИНС было разработано не мало различных архитектур для лучшей работы сетей с определенными задачами. Перечислять их все нет смыла, поэтому рассмотрим наиболее популярные и часто используемы типы нейронных сетей, опираясь на научную статью «The Neural Network Zoo» [4].



Рисунок 5 – Схема нейронной сети прямого распространения

1. Нейронные сети прямого распространения (FFNN)

FFNN – это одни из самых обычных нейронных сетей, их часто используют как дополнение к сети другого типа, являются линейными и передают данные от входа к выходу. Они представляются в виде ориентированного графа. Метод обучения FFNN называется методом обучения без учителя, он отличается от других тем, что в других методах входные данные вводятся самостоятельно. Примерная архитектура такого рода сетей представлена на рисунке 5.

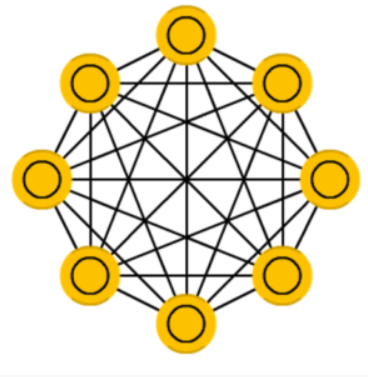


Рисунок 6 – Схема нейронной сети Хопфилда

1. Нейронные сети Хопфилда (НСХ)

Такие ИНС, как видно по схеме на рисунке 6, значительно отличаются от FFNN, они являются полносвязными и однослойными. При запуске обучения каждый из узлов сети становится на некоторое время скрытым, а после завершения они становятся выходом. Так же, как и FFNN, НСХ могут обучаться по методу без учителя. Сеть всегда будет сводиться к шаблону или шаблонам, по которым она обучилась. Такая сеть стабилизируется в зависимости от общей “энергии” и “температуры” сети. Для каждого нейрона есть свой порог активации, который зависит от температуры, при прохождении которого нейрон получается одно из двух возможных значений.

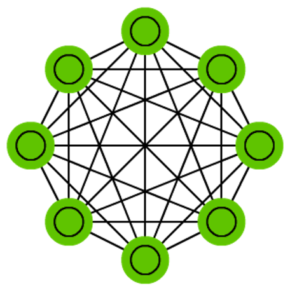


Рисунок 7 – Схема цепи Маркова

1. Цепи Маркова

Являются предшественниками машин Больцмана и НСХ и имеют схожую архитектуру с НСХ. Основной смысл таких сетей — это ответить на вопрос: смогу ли я попасть в какой-либо узел, находясь в другом? В таких сетях каждое следующее состояние напрямую зависит от предыдущего состояния.

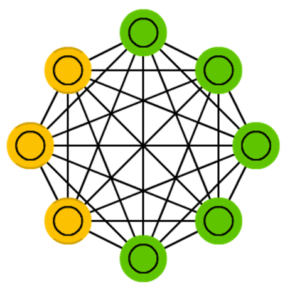


Рисунок 8 – Схема машины Больцмана

1. Машина Больцмана

Сети такого рода имеют большую схожесть с НСХ, но в отличие от нее, здесь уже изначально имеются нейроны, которые отмечены как скрытые и входные. Обучается такая сеть с помощью метода обратного распространения ошибки.

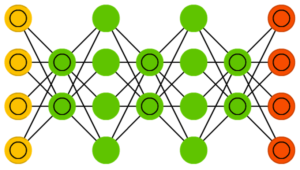


Рисунок 9 – Схема сети типа deep belief

1. Сеть типа deep belief

Сети состоят из нескольких соединенных RBM или VAE. Обучение происходит поблочно, причем каждому блоку нужно лишь уметь закодировать предыдущий. Это техника получила название “жадное обучение”, которое заключается в выборе оптимальных локальных решений, но не гарантируется, что будет выдан оптимальный результат.

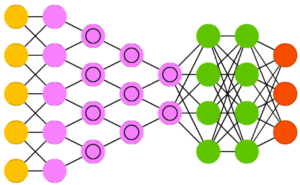


Рисунок 10 – Схема сверточной нейронной сети

1. Сверточные нейронные сети и глубинные нейронные сети (CNN, DCNN)

CNN имеют значительные отличия от рассмотренных раннее сетей. Их чаще всего используют для обработки изображений. Один из способов применения CNN – это классификация изображений. Сети такого рода обычно используют “сканер”, например, если есть изображение 200x200, сеть, вместо того, чтобы обрабатывать 40000 пикселей, будет считать квадрат размера 20x20, а затем сдвинется на 1 пиксель и посчитает новый квадрат. На практике к концу CNN прикрепляют FFNN. Такие сети уже называются глубинными (DCNN – deep convolutional neural networks).

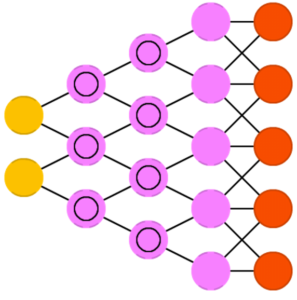


Рисунок 11 – Схема развертывающийся нейронной сети

1. Развертывающие нейронные сети

Их также называют обратными графическими сетями, они являются обратными к сверточным сетям. К ним также можно прикрепить FFNN. В большинстве случаев сети передается не строки, а бинарные вектора.

**3 Структура сети**

**3.1 Распознавание образов**

Как уже было выведено раннее, ИНС обучаются, и именно это делает их более выгодными на фоне программ, построенных по традиционным алгоритмам. Во время обучения ИНС могут составлять обобщения из уже выученных ранее материалов, а также обнаруживать зависимость между входными и выходными данными. Таким образом, можно сделать вывод, что при удачном обучении сети она способна дать верный результат, ссылаясь на данные, которые выучила в ходе обучения [10].

Сама задача по распознаванию образов ставится следующим образом. Существует некоторый способ, по которому происходит кодировка данных, например, рукописных цифр, принадлежащих определенному и известному конечному классу множеств {Ai} (i = 1-n), так же имеется какое-то конечное множество объектов. Про каждый из этих объектов должно быть известно к какому классу он относится. Все что требуется – это создать такой алгоритм, который при получении нового объекта, смог на основе уже имеющихся данных, определить к какому именно классу относится этот полученный объект. Качество, по которому оценивается алгоритм обучения сети, определяется по количеству ошибок (чем меньше ошибок, тем лучше).

Распознавание можно разбить на три части: извлечение признаков, распознавание и вывод решения.

Что такое извлечение признаков? Прежде всего следует понимать, что ИНС не воспринимает данные так, как они подаются. И именно для и существует извлечение признаков, благодаря которому происходит преобразование поданного на вход объекта к такому виду, который ИНС сможет понять и работать с ним. Одним из самых удобных оказалось представление данных в виде точки, принадлежащей евклидовому пространству Rn. Чтобы распознавание объектов оказалось качественным необходимо правильно подобрать размерность n. То есть, какое-либо изображение будет представляться как объект, состоящий из пар координат точки. Такое представление информации в дальнейшем будет удобно кодировать, так как можно отбросить члены, которые не будут влиять на классификацию. Извлечение признаков напрямую зависит от кодировки, которая была применена.

Распознавание последовательностей групп объектов, например, слов, будет отличаться от распознавания единственных объектов, например, цифр. Главное их различие будет заключаться в том, что существует огромное количество последовательностей с разными длинами, и для их обучения понадобится очень много времени. А для распознания одного объекта требуется гораздо меньше времени. Но как же распознавать последовательности? Можно разбивать последовательность на группы более мелких последовательностей, которые успешно распознаются сетью. Из уже полученных результатов складывается необходимый ответ. Для этого существуют алгоритмы склеивания [8].

**3.2 Реализация ИНС**

Итак, нейронные сети состоят из следующих элементов:

- входной слой x,

- произвольное количество скрытых слоев,

- выходной слой y,

- также для реализации данной архитектуры необходим набор весов и смещений w и b.

Переменные w и b часто являются единственными переменными, из-за которых изменяется значение y.

Для более точного результата необходимо правильно подобрать значение весов и смещения, то есть обучить сеть. Каждая попытка обучения состоит также нескольких шагов:

- необходимо вычислить прогноз y, которое называют прямым распространением;

- после каждой попытки обновить значение весов и смещений. Этот процесс называют обратным распространением.

**3.3 Обратное распространение**

Что же представляет собой обратное распространение? Основная идея этого метода – это распространение некоторой ошибки, которая появилась в ходе обучения, от выхода к входу сети. Этот алгоритм относится к методу обучения с учителем. В ходе обучения на вход многократно даются данные, которые ИНС должна распознать. С каждой попыткой образы, которые подаются обычно немного изменяют.

Более подробно это описать так:

Предложенные входные данные разбивают на две последовательности: одна будет обучающей другая контролирующей.

Далее следует провести инициализацию всех имеющихся весов, чтобы определить начальное значение ошибки. Затем производится первый запуск обучения, в ходе которого идет проверка, а именно вычисляется разность полученным выходным результатом каждого нейрона и полученном. Если они не совпадают, то производится процедура обратного распространения, вследствие чего идет изменение значения весов.

После вновь повторяем все предыдущие шаги, пока не будет получен удовлетворяющий результат.

**3.4 Функция потерь**

Для полной оценки обучения этого недостаточно, необходимо добавить функцию потерь, которая будет характеризовать потери в случае ошибки сети. Таким образом, основная задача такого рода обучения – это найти такой набор весов и смещений, чтобы минимизировать функцию потери.

В большинстве случаев, функция потерь может вычисляться с помощью суммы квадратов ошибок. Такая сумма является разницей между желаемым и полученным значением.

Одно из свойств таких функций – это довольно высокая чувствительность к наличию выбросов, то есть к редким значениям, которые могут существенно отличаться от получаемых до этого результатов.

**4 Распознание цифр с помощью ИНС**

**4.1 Разбор поставленной задачи**

При попытке дать ИНС распознать рукописные цифры, может возникнуть ряд проблем, так как у каждого человека есть свой стиль написания. Возьмем, к примеру, цифры 3 и 5. Некоторые варианты их написания могут быть очень схожи с написанием 8 и 6. Что же делать в таком случае? Можно смоделировать построение одной цифры в виде последовательности элементов, но тогда нужно будет учесть ряд таких фактор как то, что одна и та же цифра может быть написана разными способами, также возможно не совсем правильно написание. Невозможно учесть все факторы, поэтому следует взять несколько из них.



Рисунок 12 – Представление рукописных цифр

На рисунке 12 представлены варианты написания цифр. Будет фокусироваться на написании программы, которая будет решать проблему определения отдельных цифр от 0 до 9.

Определим какую сеть хотим построить. Можно создать глубокую нейронную сеть, которая содержит несколько скрытых слоев, или теневую нейронную сеть, содержащую один скрытый слой.

Построим трехслойную нейронную сеть, как представлена на рисунке 13.

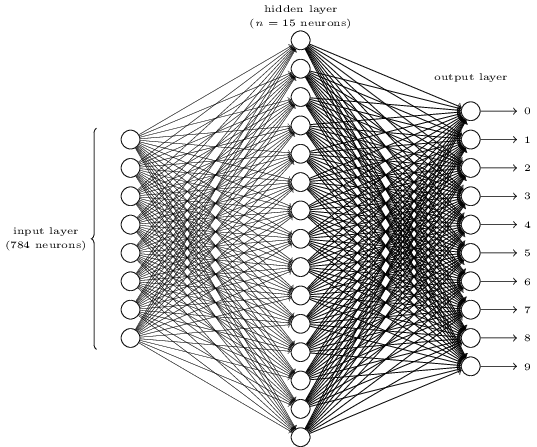


Рисунок 13 – Трехслойная нейронная сеть

На входящий слой подаются нейроны, которую будут представлять собой пиксели изображения. Пусть тренировочная память будет вмещать 28x28 пикселей, тогда входящий слой будет состоять из 784 нейронов [3].

Второй слой – скрытый слой. Обозначим количество нейронов в скрытом слой как n.

Когда построена сеть, для тренировки сети подключим библиотеку MINST, которая содержит больше тысячи изображений рукописных цифр. MNIST состоит их двух частей. Первая часть – это непосредственно сами тренировочные изображения размера 28x28 пикселей. Вторая часть – это изображения для тренировки, которые используются для определения того, насколько хорошо сеть распознает цифры.

Обозначим за x пространственный вектор, каждое вхождение в вектор представляет собой количество серых пикселей в изображении. На выходе будет пространственный вектор y = y(x), состоящий из 10 элементов, так как сеть будет определять цифры от 0 до 9, то есть если на вход будет подано изображение 2, то вектор имеет вид [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]. Каждый элемент в данном векторе y представляет определенный класс цифр.

Необходим алгоритм, благодаря которому будем находить веса и смещения, чтобы выходные данные были приближенными к y(x) для всех тренировочных вхождений.

Также нам понадобится такие переменные как: итерация – счетчик, который увеличивается каждый раз, когда нейронная сеть проходит один сет обучения, эпоха – величина, увеличивающаяся при прохождении всего обучения полностью, чем выше будет эпоха, тем лучше обучена сеть, еще нужно включить в алгоритм обучения функцию потерь. После каждого прохождения обучения будем изменять веса так, чтобы уменьшить размер потерь, если результат расходится с ожидаемым.

Таким образом, процесс обучения будет заключаться в попытке уменьшить функцию потерь после каждого прохождения программой самого обучения, вследствие изменения весов и смещений.

Если при изменении всех необходимых параметров сеть все равно не удовлетворяет желаемому результату, можно изменить количество нейронов в скрытых слоях.

Для написания ИНС воспользуемся Python 3.6, так как он лучше всего подходит для создания сетей. Как было сказано выше, понадобится подключить библиотеку MNIST. Помимо MNIST понадобится также библиотека NumPy для удобной и быстрой работы с алгебраическими функциями и TensorFlow – библиотека, помогающая в построении и решении ИНС.

**ЗАКЛЮЧЕНИЕ**

Конечно, на сегодняшний день ИНС способны не только различать цифры, но и другие символы, например, рукописный текст, а также различного рода изображения, причем такие сети используют подобный алгоритм.

Данный алгоритм может быть также усовершенствован и использоваться даже для разработке различного рода программного обеспечения для персональных компьютеров.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Ту Дж., Гонсалес Р. Принципы распознавания образов – 1978.

2 Neural networks and deep learning // (Engl). – URL:

http://neuralnetworksanddeeplearning.com/chap1.html [26 Ноября 2018]

3 How to build a neural network to recognize handwritten digits with TensorFlow // (Engl). – URL: https://www.digitalocean.com/community/tutorials/how-to-build-a-neural-network-to-recognize-handwritten-digits-with-tensorflow [26 Ноября 2018]

4 The Neural network zoo // (Engl). – URL: http://www.asimovinstitute.org/neural-network-zoo/ [5 Декабря 2018]

5 Практический нерокомпьютинг // Уральский Государственный Технический Университет – 2005. – URL: http://314159.ru/scherbatsky/scherbatsky2.pdf [7 Декабря 2018]

6 History of the Perceptron // (Engl). – URL: https://web.csulb.edu/~cwallis/artificialn/History.htm [5 Декабря 2018]

7 Уоссермен Ф. Нейрокомпьютерная техника – 1992.

8 Digital Recognition using Neural Network // (Engl). – URL: https://pdfs.semanticscholar.org/e248/4f2b2d53f3d53395e5ab79be15adf867cb29.pdf [8 Декабря]

9 A 'Brief' History of Neural Nets and Deep Learning // (Engl). – URL: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/ [8 Декабря]

# 10 Хайкин С. Нейронные сети. Полный курс – 2008.