Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГООБРАЗОВАНИЯ

«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙУНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Кафедра компьютерной безопасности и математического

обеспечения информационных систем

**КУРСОВАЯ РАБОТА**

по дисциплине «Языки программирования»

**Проектирование и разработка программы для распознавания рукописных цифр с помощью нейронной сети**

Руководитель

\_\_\_\_\_\_\_ К.Р. Джукашев

подпись инициалы фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2022г.

Студент группы

21КБ(с)РЗПО

\_\_\_\_\_\_\_ Н.С. Фот

подпись инициалы фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2022г.

Оренбург 2022

**Содержание**

Введение1

Теоретическая основа курсовой работы5

Теория о всех вариантов алгоритмов: нейросети5

Выбор методов и алгоритмов. Варианты активационных функций 6

Активационная функция 8

# **Введение**

На данный момент можно точно сказать одно - количество информации, генерируемое человечеством за определённый промежуток времени, невозможно обработать традиционными алгоритмическими конструкциями. В связи с этим, смотря на невероятно эффективную работу мозга, программистами был разработан специальный алгоритм, который смог с эмулировать работу отдельных нейронов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения - одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Актуальность работы обусловлена тем, что в современном мире всё больше людей взаимодействуют с компьютером, заведомо не имея даже начальных навыков компьютерной грамотности, из-за чего уменьшение порога для входа за счёт понимания компьютером более высокоуровневых языков является одной из главных задач перед программистами.

Таким образом мы пришли к выводу о том, что использование выше описанных нейронных сетей является большой необходимостью из-за чего целью данной курсовой работы стала разработка программного продукта, реализующего работу и демонстрацию работы нейронной сети на примере распознавании рукописных цифр и символов.

Задачами же можно назвать:

* реализацию работы самой нейронной сети
* реализацию механизма её обучения
* реализацию пользовательского интерфейса

# **Теоретическая основа курсовой работы**

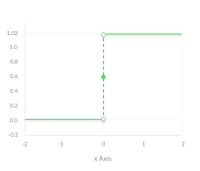
**1.1 Теория о всех вариантов алгоритмов: нейросети**

Нейронная сеть (также искусственная нейронная сеть, ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

В качестве образов могут выступать различные по своей природе объекты: символы текста, изображения, образцы звуков и т. д. При обучении сети предлагаются различные образцы образов с указанием того, к какому классу они относятся. Образец, как правило, представляется как вектор значений признаков. При этом совокупность всех признаков должна *однозначно определять класс*, к которому относится образец. В случае, если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно. По окончании обучения сети ей можно предъявлять неизвестные ранее образы и получать ответ о принадлежности к определённому классу.

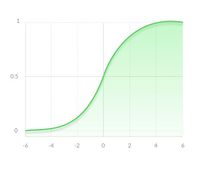
Топология такой сети характеризуется тем, что количество нейронов в выходном слое, как правило, равно количеству определяемых классов. При этом устанавливается соответствие между выходом нейронной сети и классом, который он представляет. Когда сети предъявляется некий образ, на одном из её выходов должен появиться признак того, что образ принадлежит этому классу. В то же время на других выходах должен быть признак того, что образ данному классу не принадлежит. Если на двух или более выходах есть признак принадлежности к классу, считается, что сеть «не уверена» в своём ответе.

* 1. **Выбор методов и алгоритмов. Варианты активационных функций.**

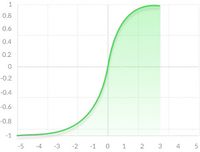
1) Ступенчатая функция (англ. *binary step function*) является пороговой функцией активации. То есть если zz больше или меньше некоторого значения, то нейрон становится активированным. Такая функция отлично работает для бинарной классификации. Но она не работает, когда для классификации требуется большее число нейронов и количество возможных классов больше двух.

2) Линейная функция (англ. *linear function*) представляет собой прямую линию, а это значит, что результат этой функции активации пропорционален переданному аргументу. В отличии от предыдущей функции, она позволяет получить диапазон значений на выходе, а не только бинарные 0 и 1, что решает проблему классификации с большим количеством классов.

*linear function*

 3) Сигмоидная функция (англ. *sigmoid function*), которую также называет логистической (англ. *logistic function*), является гладкой монотонно возрастающей нелинейной функцией. И так как эта функция нелинейна, то ее можно использовать в нейронных сетях с множеством слоев, а также обучать эти сети методом обратного распространения ошибки. Сигмоида ограничена двумя горизонтальными асимптотами y=1y=1 и y=0y=0, что дает нормализацию выходного значения каждого нейрона. Кроме того, для сигмоидной функции характерен гладкий градиент, который предотвращает "прыжки" при подсчете выходного значения. Помимо всего этого, у этой функции есть еще одно преимущество, для значений x>2x>2 и x<−2x<−2, yy "прижимается" к одной из асимптот, что позволяет делать четкие предсказания классов.

*sigmoid function*

4) Функция гиперболического тангенса (англ. *hyperbolic tangent*). Эта функция является скорректированной сигмоидной функцей

tanh(z)=2⋅sigma(2z)−1tanh(z)=2⋅sigma(2z)−1,

то есть она сохраняет те же преимущества и недостатки, но уже для диапазона значений (−1;1)(−1;1).

*hyperbolic tangent*

* 1. **Типы нейронных сетей**

Есть десятки видов нейросетей, которые отличаются архитектурой, особенностями функционирования и сферами применения. При этом чаще других встречаются сети трех видов.

Нейронные сети прямого распространения (Feed forward neural networks, FFNN). Прямолинейный вид нейросетей, при котором соседние узлы слоя не связаны, а передача информации осуществляется напрямую от входного слоя к выходному. FFNN имеют малую функциональность, поэтому часто используются в комбинации с сетями других видов.

Сверточные нейронные сети (Convolutional neural network, CNN). Состоят из слоев пяти типов:

* входного;
* свертывающего;
* объединяющего;
* подключенного;
* выходного.

Каждый слой выполняет определенную задачу: например, обобщает или соединяет данные.

Сверточные нейросети применяются для классификации изображений, распознавания объектов, прогнозирования, обработки естественного языка и других задач.

Рекуррентные нейронные сети (Recurrent neural network, RNN). Используют направленную последовательность связи между узлами. В RNN результат вычислений на каждом этапе используется в качестве исходных данных для следующего. Благодаря этому, рекуррентные нейронные сети могут обрабатывать серии событий во времени или последовательности для получения результата вычислений.

RNN применяют для языкового моделирования и генерации текстов, машинного перевода, распознавания речи и других задач.

* 1. **Типы задач, которые решают нейронные сети**

Выделяют несколько базовых типов задач, для решения которых могут использоваться нейросети.

* Классификация. Для распознавания лиц, эмоций, типов объектов: например, квадратов, кругов, треугольников. Также для распознавания образов, то есть выбора конкретного объекта из предложенного множества: например, выбор квадрата среди треугольников.
* Регрессия. Для определения возраста по фотографии, составления прогноза биржевых курсов, оценки стоимости имущества и других задач, требующих получения в результате обработки конкретного числа.
* Прогнозирования временных рядов. Для составления долгосрочных прогнозов на основе динамического временного ряда значений. Например, нейросети применяются для предсказания цен, физических явлений, объема потребления и других показателей. По сути, даже работу автопилота Tesla можно отнести к процессу прогнозирования временных рядов.
* Кластеризация. Для изучения и сортировки большого объема неразмеченных данных в условиях, когда неизвестно количество классов на выходе, то есть для объединения данных по признакам. Например, кластеризация применяется для выявления классов картинок и сегментации клиентов.
* Генерация. Для автоматизированного создания контента или его трансформации. Генерация с помощью нейросетей применяется для создания уникальных текстов, аудиофайлов, видео, раскрашивания черно-белых фильмов и даже изменения окружающей среды на фото.
  1. **Обучение нейронной сети**

Обучение нейронной сети — это процесс обучения нейронной сети выполнению задачи. Нейронные сети обучаются путем первичной обработки нескольких больших наборов размеченных или неразмеченных данных. На основе этих примеров сети могут более точно обрабатывать неизвестные входные данные.

При контролируемом обучении специалисты по работе с данными предлагают искусственным нейронным сетям помеченные наборы данных, которые заранее дают правильный ответ. Например, сеть глубокого обучения, обучающаяся распознаванию лиц, обрабатывает сотни тысяч изображений человеческих лиц с различными терминами, связанными с этническим происхождением, страной или эмоциями, описывающими каждое изображение.

Нейронная сеть медленно накапливает знания из этих наборов данных, которые заранее дают правильный ответ. После обучения сеть начинает делать предположения об этническом происхождении или эмоциях нового изображения человеческого лица, которое она никогда раньше не обрабатывала.

Искусственный интеллект — это область компьютерных наук, которая исследует методы предоставления машинам возможности выполнять задачи, требующие человеческого интеллекта. Машинное обучение — это метод искусственного интеллекта, который дает компьютерам доступ к очень большим наборам данных для дальнейшего обучения. Программное обеспечение для машинного обучения находит шаблоны в существующих данных и применяет эти шаблоны к новым данным для принятия разумных решений. Глубокое обучение — это разновидность машинного обучения, в котором для обработки данных используются сети глубокого обучения.

Традиционные методы машинного обучения требуют участия человека, чтобы программное обеспечение работало должным образом. Специалист по работе с данными вручную определяет набор соответствующих функций, которые должно анализировать программное обеспечение. Это ограничение делает создание и управление программным обеспечением утомительным и трудозатратным процессом.

С другой стороны, при глубоком обучении специалист по работе с данными предоставляет программному обеспечению только необработанные данные. Сеть глубокого обучения извлекает функции самостоятельно и обучается более независимо. Она может анализировать неструктурированные наборы данных (например, текстовые документы), определять приоритеты атрибутов, данных и решать более сложные задачи.

Например, при обучении программного обеспечения с алгоритмами машинного обучения правильно идентифицировать изображение домашнего животного вам потребуется выполнить следующие шаги:

Найти и вручную отметить тысячи изображений домашних животных: кошек, собак, лошадей, хомяков, попугаев и т. д.

Сообщить программному обеспечению с алгоритмами машинного обучения, какие функции необходимо найти, чтобы оно могло идентифицировать изображение методом исключения. Например, оно может подсчитать количество ног, а затем проверить форму глаз, ушей, хвоста, цвет меха и так далее.

Вручную оценить и изменить помеченные наборы данных, чтобы повысить точность программного обеспечения. Например, если в вашем тренировочном наборе слишком много изображений черных кошек, программное обеспечение правильно определит черную кошку, но не белую.

При глубоком обучении нейронные сети будут обрабатывать все изображения и автоматически определять, что сначала им требуется проанализировать количество ног и форму морды, а уже после посмотреть на хвосты, чтобы правильно идентифицировать животное на изображении.

**2 Выбор активационной функции**

Для реализации работы нейронной сети изначально было необходимо выбрать одну из выше представленных активационных функций нейрона. Основным претендентом стала функция, выраженная на основе «Rectified Linear Unit», или в сокращении «ReLU». Говоря про «родительскую» функцию, можно отметить то, что именно она наиболее часто используется для активации при глубоком обучении.

Рис. 1. - Функция ReLU

ReLU

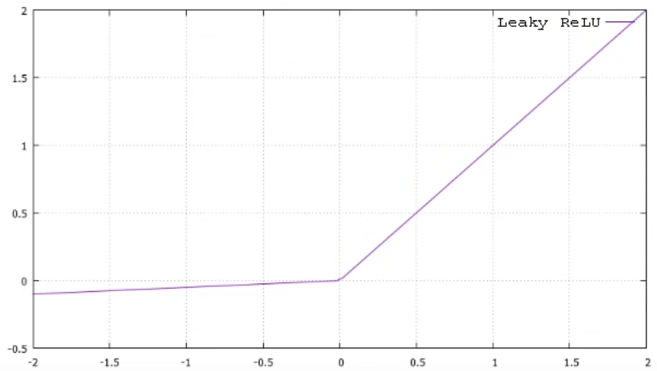
Данная функция возвращает 0, если принимает отрицательный аргумент, в случае же положительного аргумента, функция возвращает само число.

На первый взгляд может показаться, что она линейна и имеет те же проблемы что и линейная функция, но это не так и ее можно использовать в нейронных сетях с множеством слоев.

Преимущества ReLU перед сигмоидой и гиперболическим тангенсом:

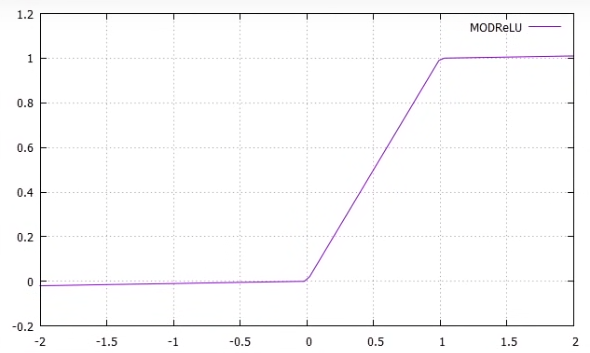
* Очень быстро и просто считается производная. Для отрицательных значений - 0, для положительных - 1.
* Разреженность активации. В сетях с очень большим количеством нейронов использование сигмоидной функции или гиперболического тангенса в качестве активационный функции влечет активацию почти всех нейронов, что может сказаться на производительности обучения модели. Если же использовать ReLU, то количество включаемых нейронов станет меньше, в силу характеристик функции, и сама сеть станет легче.

Используемая в данной курсовой работе функция является модифицированной функцией «Leaky ReLU», которая в свою очередь имеет такой вид:

Рис. 2. - Функция Leaky ReLU

Leaky ReLU

Причиной использования именно модифицированной активационной функции можно назвать специфику поставленной задачи. Алгоритм, при считывании пикселей, будет получать на вход его цвет, и так как это значение имеет верхний и нижние пределы, а именно 255 и 1 соответственно, «Leaky ReLU» нуждается в доработке:

Рис. 3. - Функция MOD ReLU

MOD ReLU

**3 Алгоритм работы нейронной сети**

**3.1 Введение**

Нейронная сеть представляет собой обыкновенную функцию, принимающую на вход одномерный массив данных, и выдающая подобный одномерный массив данных в зависимости от весов отдельных нейронов и их смещения. Но перед разбором весов и нейронов смещения, для начала стоит разобраться, какие именно данные принимает нейронная сеть в случае данной курсовой работы.

**3.2 Входной слой**

В рамках данной курсовой работы, созданная мною нейронная сеть обрабатывает ручной ввод пользователя принимая на вход одномерный массив значений цветов холста размером 28 на 28 пикселей. Представить математически это можно следующим образом:

, 0 1

Получив на вход ровно 786 значений, мы тем самым заполнили первый слой нашей нейронной сети, всего который может насчитываться от 3 штук в числе который входной слой, который выступает приёмником вводимых данных, скрытый слой, количество которых и определяет глубину нейронной сети, и выходной слой, в данных которых находится ответ нейронной сети.

**3.3 Скрытый слой**

Скрытые слои нейронной сети содержат ненаблюдаемые обрабатывающие блоки (нейроны). Значение каждого скрытого нейрона — это некоторая функция предикторов. Точная форма этой функции частично зависит от типа сети и в моём случае является формой MOD ReLU.

Каждый предыдущий слой нейронной сети, вне зависимости от своего типа, исключая разве что выходной слой, так или иначе передаёт свои значения далее посредством перемножения собственных нейронов на их веса прибавляя смещение. Выглядит такая формула примерно таким образом:

**3.3.1 Веса**

Вес представляет силу связи между нейронами. Например, если вес соединения узлов 1 и 3 больше, чем узлов 2 и 3, это значит, что нейрон 1 оказывает на нейрон 3 большее влияние. Нулевой вес означает, что изменения входа не повлияют на выход. Отрицательный вес показывает, что увеличение входа уменьшит выход. Вес определяет влияние ввода на вывод.

В самом начале обучения веса нейронной сети заполняются либо случайными значениями и процессе обучения меняются тем самым реализуя механизм работы сети.

**3.3.2 Смещение**

Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения. Это гарантирует, что даже когда все входы будут равны нулю, нейрон будет активен.