|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

***Метод повышения производительности рабочей***

***станции за счет вычислительного кластера***

Студентка ИУ7-41м **Новоженов В.А.**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКР  **Ковтушенко А.П.**

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер **Строганов Ю.В.**

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

# **АННОТАЦИЯ**

**выпускной квалификационной работы**

Новоженова Владимира Алексеевича

на тему: «Метод повышения производительности рабочей станции за счет вычислительного кластера»

Объем ВКР: количество страниц 80, иллюстраций 25, таблиц 6, использованных источников 30, приложений 1.

Ключевые слова: сиамские нейронные сети, распознавание кодовых слов, аутентификация, идентификация.

Объект исследования: вычислительные кластеры и методы повышения производительности при их использовании.

Цель работы: создание метода повышения производительности в задаче обучения нейронных сетей с использованием вычислительного кластера.

В ходе работы был создан метод повышения производительности вычислений, которые необходимо провести на рабочей станции. На рабочей станции развернута нейронная сеть, распознающая пользователей по клавиатурному почерку. В следствии, так называемого «concept drifting», появляется необходимость в дообучении нейронной сети. Дообучение нейронной сети является вычислительно сложной задачей, требующей много ресурсов и выполняющей большое число математических операций. Вследствие этого, не следует нагружать ресурсы рабочей станции задачей дообучения нейронной сети, а отправить задачу на вычислительный кластер. Само дообучение нейронной сети также следует реализовать в соответствии со спецификой кластера, а для повышения скорости получения результата и исключения дублей заданий, отправляемых на кластер следует использовать особую систему контроля заданий.

Результатом работы является сам метод повышения производительности при помощи высокопроизводительного кластера. Сам метод применим для обучения любого класса нейронных сетей, созданных на основе фреймворков TensorFlow и Pytorch.

Новизна работы состоит в использовании системы отслеживания заданий обучения нейронных сетей, отправляемых на вычислительный кластер для вычисления параллельно и распределенно.

Прогнозом развития метода является аппроксимация метода и системы контроля заданий под любой класс вычислительных математически-сложных задач, которые следует отправлять на выполнение на кластер и получать результат вычисления в другой среде, отличной от кластерной (например на некотором сервере или рабочей станции).

Автор ВКР «\_\_\_\_\_\_\_\_\_\_\_\_\_» «Новоженов Владимир Алексеевич»

**СОДЕРЖАНИЕ**

[**АННОТАЦИЯ** 2](#_Toc11867411)

[**ВВЕДЕНИЕ** 4](#_Toc11867412)

[**1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ** 6](#_Toc11867413)

[**1.1 Известные методы верификации подписи с использованием нейронных сетей** 8](#_Toc11867414)

[**1.2 Метод разблокировки смартфона Face ID** 16](#_Toc11867415)

[**1.3 Выводы** 19](#_Toc11867416)

[**2 КОНСТРУКТОРСКИЙ РАЗДЕЛ** 20](#_Toc11867417)

[**2.1 Архитектура алгоритма распознавания рукописного текста** 20](#_Toc11867418)

[**2.2 Математическая модель метода распознавания рукописного текста** 23](#_Toc11867419)

[**2.3 Состав ансамбля нейронных сетей** 39](#_Toc11867420)

[**2.4 Описание обучающей и тестовой выборок** 40](#_Toc11867421)

[**3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ** 43](#_Toc11867422)

[**3.1 Детали подготовки данных для обучения и тестирования системы** 43](#_Toc11867423)

[**3.2 Оценка качества работы метода** 48](#_Toc11867424)

[**3.3 Пути развития предложенного метода** 54](#_Toc11867425)

[**3.4 Выводы** 56](#_Toc11867426)

[**ЗАКЛЮЧЕНИЕ** 57](#_Toc11867427)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 58](#_Toc11867428)

[**ПРИЛОЖЕНИЕ A** 61](#_Toc11867429)

# 

# **ВВЕДЕНИЕ**

**Цель работы:** создание метода повышения производительности в задаче обучения нейронных сетей с использованием вычислительного кластера.

**Задачи:**

* Проанализировать методы повышения производительности в решении задачи обучения нейронных сетей. Проанализировать методы взаимодействия с вычислительным кластером по внешнему программному интерфейсу;
* Создать метод взаимодействия с кластером в рамках задачи распределенного обучения нейронных сетей с использованием среды контроля заданий;
* Протестировать разработанную систему в решении практических задач обучения нейронных сетей;

**Объект** – вычислительные кластеры и методы повышения производительности при их использовании.

**Проблема** – ресурсо-затратность и время-затратность в задаче обучения нейронных сетей без использования кластера или с использованием кластера без использования системы обработки заданий.

В настоящее время особую популярность в решении самых разных задач получили нейронные сети. С их помощью можно и определять наличие объектов на изображениях и выявлять паттерны и закономерности во временных рядах и обрабатывать и распознавать любой звук. Но перед использованием нейронных сетей их необходимо обучить на, зачастую, огромных массивах данных. Это могут быть тысячи изображений или миллионы строк данных. А для обучения нейронных сетей на них может потребоваться несколько десятков (а иногда даже тысячи) эпох (одна эпоха – полное прохождение всего набора данных в прямом и обратном направлении).

Основной проблемой является тот факт, что суммарное время обучения нейронной сети может достигать месяцев даже с использованием GPU-ускорителей, что в ряде задач может превышать время, по которому наступает «concept drifting» и не является приемлемым.

По этой причине целесообразно проводить вычисления не на машине, на которой уже развернута нейронная сеть и происходит обработка в реальном времени, а отправлять вычислительное задание на высокопроизводительный кластер. Также для контроля статуса выполнения заданий, устранения дублей заданий, подбора вычислительных ресурсов кластера, которые можно использовать для решения задания и соответствующей корректировки задания-скрипта обучения нейронной сети (который также необходимо реализовать и запускать соответствующим образом для использования всех доступных заданию мощностей и использования распределенных параллельных вычислений) целесообразно использовать систему планирования и управления заданиями, отправляемыми на кластер.

В работе проиллюстрировано взаимодействие приложения на рабочей машине пользователя, удостоверяющее его личность по клавиатурному почерку посредством отправки запросов на сервер, на котором развернута нейронная сеть. Сервер с нейронной сетью должен инициировать отправку задачи дообучения нейронной сети в систему управления заданиями на кластере. Из системы управления заданиями вычислительная задача отправляется на кластер для ускорения вычислений. Затем обученная нейронная сеть заменяет активную (по какому-либо методу разворачивания) на сервере.

# 

# **1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ**

Аналитический раздел посвящается анализу объекта технологической проработки и исследования.

Наиболее эффективным методом анализа является системный, основанный на последовательно-параллельном принципе: от «объекта» к «задачам». Независимо от природы и характера объекта анализ строится на определении целей на уровне проблем и задач и возможных путях реализации этих целей на уровне, т.е. выполнении каких-либо конкретных действий, которые выражаются вопросами, требующими решения.

Последовательность процедур анализа следующая: краткая характеристика объекта и определение вида задач разработки (исследования); сравнение различных признаков состояния объекта с признаками его развития, ретроспективы или окружающей его среды, т.е. рассмотрение альтернативы, выбранной на основе синтеза возможных вариантов, решений.

Существует множество вычислительно сложных задач. К ним можно отнести и компьютерное моделирование, в ходе которого решается ряд научно-технических или инженерных задач. Также к ним относятся и обучение нейронных сетей и классических алгоритмов машинного обучения. В ходе этих процессов производится огромное количество математических операций, для вычисления которых нужно большое количество ресурсов.

Обычные серверные или рабочие станции далеко не всегда обладают объемом вычислительных ресурсов, которые позволили бы быстро и эффективно решить какую-либо вычислительно сложную задачу.

Именно для решения таких задач конструируются вычислительные кластеры, состоящие из большого количества узлов (базовых инженерных объектов), связанных между собой на базе шинной архитектуры или коммутатора. С помощью использования кластера возможно производить вычисления не только параллельно, но и распределенно  
[ГОСТ Р ИСО/МЭК 10746-3-2001].

Также кластерные системы позволяют обеспечить повышенную надежность в решении ряда задач (например, производя вычисления одновременно на двух и более узлах, чтобы, в случае отказа одного из них, вычисления не были прерваны).

Кластерные системы обладают также большим потенциалом к масштабированию, которое реализуется путем увеличения количества вычислительных узлов.

Разделяют два вида кластеров:

* Гомогенные кластерные системы – системы, состоящие из одинаковых узлов;
* Гетерогенные кластерные системы – системы, состоящие из разнотипных узлов;

Отличительной особенностью кластерных вычислительных систем является высокий уровень доступности. Так как каждый отдельный вычислительный узел функционирует самостоятельно, на основе своей операционной системы (отсутствует единая операционная система), а также отсутствует совместно используемая память, это приводит к отсутствию проблемы когерентности кэшей. У каждого узла вычислительного кластера своя память. А результат вычислений на узле по некоторому интерфейсу отправляется на управляющую машину кластера, где, как правило, агрегируется с результатами, полученными на других узлах.

Для обеспечения высокого уровня надежности, доступности и бесперебойности на кластерных системах необходимо наличие специализированного программного обеспечения, которое организовывало бы бесперебойную работу при отказе одного или более узлов. Такое программное обеспечение осуществляет механизм передачи сообщений поверх стандартных сетевых протоколов.

### [Источник: <http://www.nsc.ru/win/elbib/data/show_page.dhtml?77+858> Кластерные вычислительные системы]

### [Цилькер Б.Я. Организация ЭВМ и систем : Учебник для вузов / Б.Я. Цилькер, С.А. Орлов. - 2-е изд. - СПб.: Питер, 2011. - 688 с. - ISBN 978-5-49807-862-5.]

Идентификация и аутентификация тесно связаны между с собой. Во многих схемах доступа, для прохождения авторизации первым этапом является идентификация, а вторым аутентификация.

Идентификация - это присвоение субъектам или объектам идентификатора или сравнение текущего идентификатора с перечнем уже существующих идентификаторов. Аутентификация - это проверка подлинности пользователя, например, с помощью пароля, электронной подписи или биометрических данных. [ГОСТ Р ИСО/МЭК 9594-8-98]

Данный процесс, в рамках распознавания рукописного текста, часто разделяют на несколько этапов:

1) Регистрация подписи. Человек оставляет подпись несколько раз, для сбора статистики;

2) Для пользователя определяются уникальные характеристики подписи;

3) Введение образца, тем же методом, которым получали эталон;

4) Сравнение эталона с образцом;

5) Качественная и количественная оценки несовпадений.

Среди систем аутентификации большими перспективами в настоящее время обладают биометрические системы, основанные на поведенческой характеристике человека и учитывающие особенности, характерные для подсознательных движений человека в процессе воспроизведения какого-либо действия. К таким методам относится аутентификация по рукописному почерку, голосу и другие.

Существует несколько статических подходов к решению указанной задачи. Их можно разделить на два вида: структурные и эталонные (Рисунок 1).

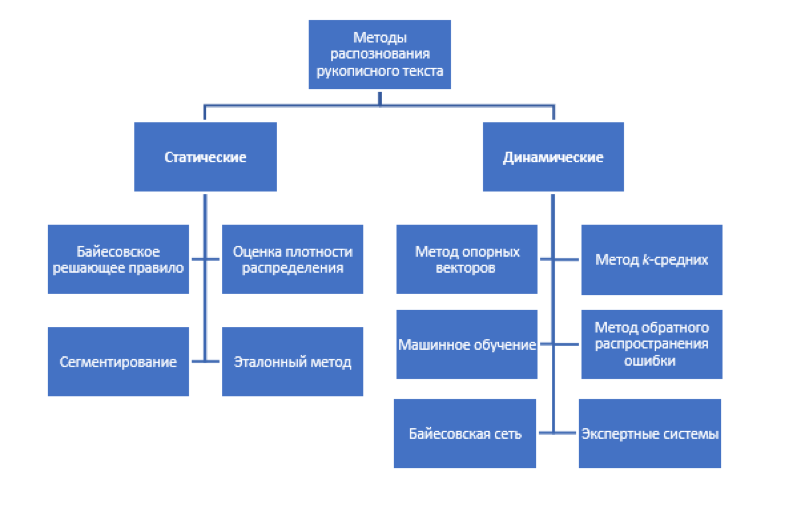


Рисунок 1 Методы распознавания рукописного текста

Первый метод основан на выделении и анализе различных структурных элементов символа и их признаков, обычно они используются в примитивных алгоритмах, так как каждый из них малоэффективен в задаче идентификации. Второй предполагает сравнение распознаваемого символа с набором заданных эталонов, хранящихся на сервере.

К статическим методам идентификации человека по его подписи можно отнести:

1. Поиск центра тяжести подписи (разбиение образца на области и последующее нахождение блока с максимальным скоплением черных пикселей);
2. Отношение числа белых пикселей к черным;
3. Определение угла наклона (путем нахождения центра тяжести в 1 и 4 четверти образца и вычисления угла наклона, относительно оси x);
4. Проекция на оси (число пикселей подписи, вошедших в точку проекции на ось с последующим размытием методом Гаусса).

Расчет разницы между полученными векторами ответов в рамках задачи идентификации часто дает некорректный результат. Перечисленные методы не позволяют решить задачу в общем виде, когда необходимо различать пользователя, независимо от внешних условий.

## **1.1 Известные методы верификации подписи с использованием нейронных сетей**

Верификация подписи - это биометрическая технология, используемая для идентификации личности. Чаще всего документы подтверждают именно подписью и не всегда есть возможность отличить опытные подделки от оригинала простым осмотром.

Технологии сравнения требуются для оптимизации документооборота в таких сферах, как страховое дело, банковское дело, электронная коммерция, офисная деятельность, контроль физического доступа и для учета рабочего времени.

В большинстве случаев, для распознавания рукописного текста используются нейронные сети. К основным преимуществам в решении этой задачи с использованием нейронных сетей можно отнести способность обучаться самостоятельно и автоматически на основе выборок, быть продуктивными на “зашумленных” данных, с возможностью параллельной реализации и эффективными инструментами для обработки больших баз данных.[9]

Нейронная сеть - это соединенные и взаимодействующие между собой простые процессоры (искусственные нейроны). Искусственный нейроны устроены проще, чем процессоры, используемые в персональных компьютерах.

Каждый нейрон принимает и передает сигналы, получаемые от других нейронов. Когда эти нейроны соединены в достаточно большую сеть с управляемым взаимодействием, они способны вычислять сложные задачи, обучаясь в процессе работы.

Нейроны во входном слое формируют интерфейс нейронной сети. В изображенной нейросети изображен входной, внутренние и выходной слои (*Рисунок 2*).[7]

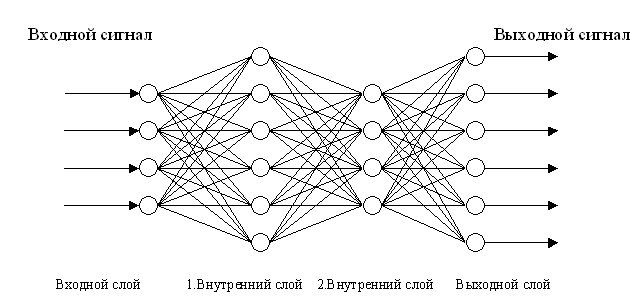


Рисунок 2 Базовая структура нейронной сети

Вся информация вводится в нейронную сеть с помощью входного слоя. Все слои обрабатывают сигналы до тех пор, пока они не достигнут слоя на выходе.[10]

Нейронная сеть решает задачу преобразования информации необходимым образом. При обучении используется учитель, который оценивает поведение нейронной сети, сравнивая эталонные значения пар на входе и на выходе. Кроме этого, сети обучаются с помощью различных алгоритмов. Обучающий алгоритм изменяет веса так, чтобы поведение нейронной сети приводило к желаемым результатам. Не обученная сеть не будет давать желаемые ответы.

Наиболее общая схема математической модели нейрона представлена на Рисунок 3. Суммирующая функция собирает все сигналы от отправителя. Значением данной функции является взвешенная сумма, где веса представляют собой синаптические мощности.[8]

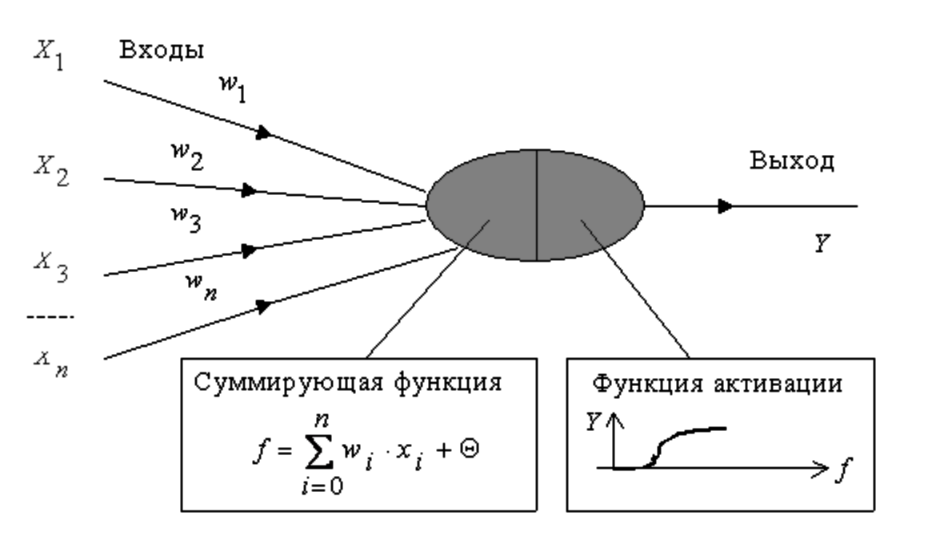


Рисунок 3 Простая математическая модель нейрона

Существует большое количество разновидностей нейронных сетей [22]. Постоянно появляются новые виды, некоторые из них похожи по архитектуре на предыдущие, но различаются методами обучения, или же некоторые схожи методами обучения, но имеют разительные отличия в архитектуре. Есть и уникальные новые сети.

Примером нейронной сети может послужить простая сеть прямого распространения (Рисунок 4).

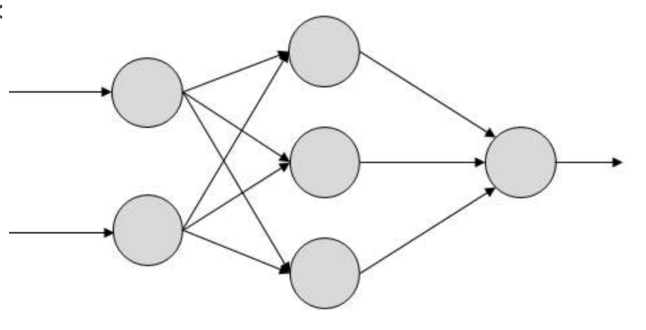


Рисунок 4 - Схематичное изображение нейросети прямого распространения

Такие сети передают информацию от входа к выходу. Поток информации является однонаправленным. Блок передает информацию на другие единицы, от которых он не получает никакой информации, также в реализации нет петли обратной связи (Рисунок 4). Сеть имеет фиксированные входы и выходы.

Обычно нейронные сети обучаются методом обратного распространения ошибки, в котором сеть получает входные и выходные данные, такое обучение называется обучением “с учителем”, и ее отличием от обучения “без учителя” заключается в отсутствии получения выходных данных.

Такие сети в чистом виде используют редко, обычно комбинируют с другими типами сетей.[11]

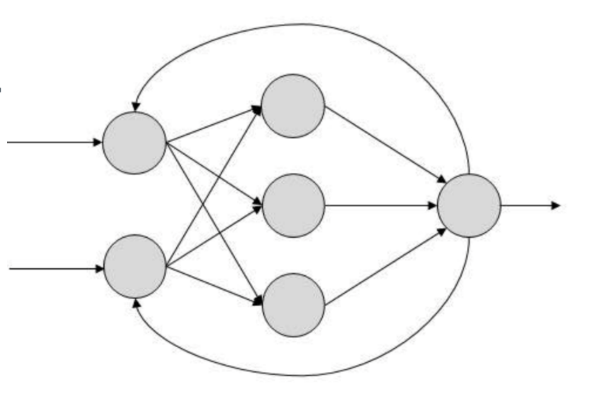


Рисунок 5 Схематичное изображение нейросети с обратной связью

Обучение с учителем обычно используется в задачах регрессии и классификации. Вторая задача рассмотрена в данной работе. Разработчик сети является для нее учителем, так как подает входные данные и желаемые данные на выходе, тем самым обучая сеть выдавать нужные ответы. Обучение без учителя встречается существенно реже, решает задачу группировки данных по определенным параметрам.

Существует еще один метод, называемый обучением с подкреплением. Такой метод применим, если есть возможность дать оценку сети. Каждый раз, когда сеть дает правильный ответ, она поощряется, так нейронной сети предоставляется возможность найти любой способ достижения цели.

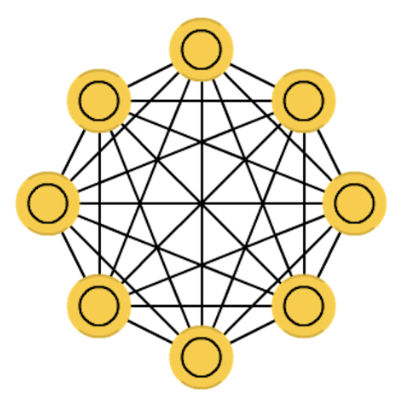


Рисунок 6 - Схематичное изображение нейронной сети Хопфилда

Сеть Хопфилда (*Рисунок 6*) это полносвязная нейронная сеть с симметричной матрицей связей. Данные получает каждый узел, являясь входом, далее при обучении узлы становятся скрытыми слоями, после они становятся выходом. Обучение сети происходит следующим образом - значения нейронов устанавливаются в соответствии с желаемым шаблоном.

Далее вычисляются веса, которые не меняются в дальнейшем. После обучения сети на одном или нескольких шаблонах, она всегда будет сводится к одному из этих шаблонов. Эту сеть часто называют сетью с ассоциативной памятью. Например, получая наполовину зашумленную таблицу, она может восстановить ее до полной.[12]

Нейронная сеть вида машины Больцмана представлена ниже (Рисунок 7). Процесс ее обучения похож на обучение сети Хопфилда. Используется метод обратного распространения ошибки или по алгоритму сравнительной сходимости. Главное ее отличие от сети Хопфилда в нейронах, некоторые из них помечены как входные, а некоторые, как скрытые.

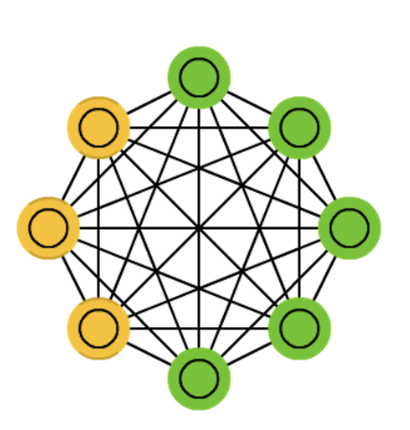


Рисунок 7 - Схематичное изображение нейросети цепи Маркова

Также широко применим вид нейросети, называемый ограниченной машиной Больцмана (*Рисунок 8*).

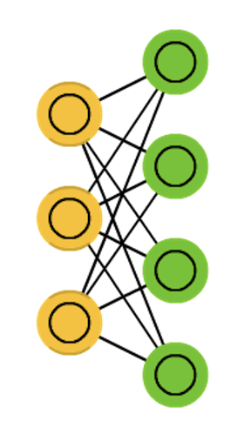


Рисунок 8 - Схематичное изображение сети ограниченной машины Больцмана

Нейроны одного типа в ограниченной машине Больцмана между собой не связаны. Сеть можно обучать так же, как нейронную сеть прямого распространения, однако, вместо прямой передачи и обратного распространения ошибки нужно передавать данные сначала в прямом направлении, затем в обратном. Далее проходит обучение по методу прямого и обратного распространения ошибки.

Нейронная сеть типа автокодировщик представлена на Рисунок 9.

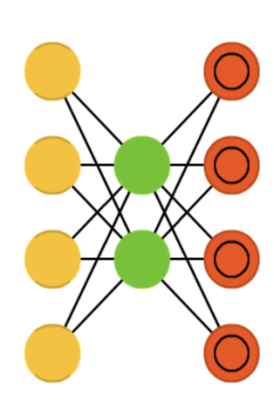


Рисунок 9 Схема сети “Автокодировщик”

“Автокодировщик” использует прямое распространение. Основной идеей является автоматическое сжатие информации. В сети скрытые слои меньше входного и выходного, и она симметрична. Сеть можно обучить методом обратного распространения ошибки, задавая ошибку равной разнице между входным и выходным слоями.[17]

Помимо приведенных выше сетей, еще различают сверточные и глубинные сверточные нейронные сети (*Рисунок 10*).

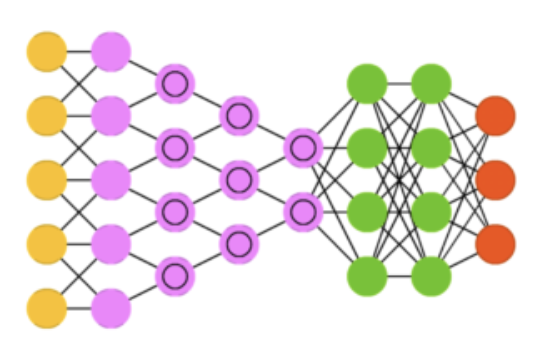


Рисунок 10 Схема сверточной нейросети

Данная архитектура имеет сильные отличия от других приведенных выше. Обычно такие сети используются для обработки изображений, иногда для аудио. Типичными задачи для сетей данного типа является классификация изображений. Эти сети часто используют так называемый “сканер”, который берет небольшую область, обычно из левого верхнего угла и сдвигается вправо на один пиксель, для расчета нового квадрата.

После чего эти данные попадают на сверточные слои, в которых не все узлы соединены между собой, такие слои сжимаются глубиной. На практике к таким сетям обычно присоединяют сеть прямого распространения для дальнейшей обработки. Такие сети называют глубинными.

В статье [1] применение нейронных сетей в распознавании рукописного текста предлагается использовать нейронную сеть, которую можно будет обучать с учителем и самостоятельно определять уровень ее надежности. В статье рассматривается задача распознавания слитного рукописного текста, обсуждаются различные подходы к решению данной задачи. Рассмотрен подход к распознаванию вместе написанного слова, включающий в себя этап разбиения, основанный на анализе структуры слова, и процедуру распознавания изображения, основанную на нейронной сети.

В публикации “Идентификация подписи с помощью радикальных функций” [2] описывается исследование идентификации on-line подписи с помощью радиальных функций (определенные в евклидовом пространстве Rn, значение которых в каждой точке зависит только от расстояния между этой точкой и началом координат) и вейвлетов (математические функции, позволяющая анализировать различные частотные компоненты данных). Кроме этого, описывается представление подписи в виде функции, применение для ее описания радиальных функций и вейвлет преобразований, также было приведено сравнение полученных функций подписей одного автора, разных людей с целью идентификации подписей.[18]

В статье [3] рассматривается задача распознавания слитного рукописного текста, а также выявляются различные подходы к решению данной задачи, отмечаются достоинства и недостатки рассматриваемых подходов. Также предлагается комбинированный подход к распознаванию слитно написанного слова, который включает в себя процедуру разбиения, созданную на анализе структуры слова, а также процедуру распознавания, с использованием нейронной сети.

В публикации [4] рассматривается задача распознавания текста на китайском языке с использованием ключей на основе нейронных сетей. В статье рассмотрены этапы процесса распознавания рукописного текста. Разработанный в результате исследования алгоритм производит оффлайн распознавание текста на основе разбиения иероглифа на ключи и их распознование с помощью неокогнитрона.

Описываемое приложение позволит значительно упростить работу переводчиков, а также поможет при изучении китайского языка. Минус предложенного метода заключается в сложности реализации системы и отсутствия возможности накопления знаний.

В работе [3] представлен метод классификации рукописных образов графического файла, основанный на выделении точек, несущих наибольшую информацию об их идентификации с обработкой и инвертированием информации по положению, размеру и толщине линий для выделения рукописного русского текста.

В статье [4] описывается модификация и применение программы для распознавания рукописных букв русского алфавита. В рамках работы была использована программа на базе нейронной сети для распознавания рукописных цифр, написанная AndrewNg Стэнфордского университета. К минусам данной программы можно отнести то, что сеть ограничена лишь русским алфавитом.

Исходя из предложенных выше методов, можно сделать вывод о наличии заметных ограничений практической применимости известных методов для качественного адаптивного распознавания рукописного текста.

## **1.2 Метод разблокировки смартфона Face ID**

Одной из самых обсуждаемых новой функций после выхода Iphone X в 2017 году, стала разблокировка смартфона с помощью Face ID [16]. Компания Apple создала безрамочный телефон, в связи с чем пришлось отказаться от TouchID и разработать новую систему разблокировки смартфона. Новый смартфон оснащен фронтальной глубинной камерой, благодаря которой появилась возможность создавать 3D карту лица. TrueDepth камера — это система камер и датчиков в iPhone X. В нее входят семь фронтальных камер, инфракрасные камеры, инфракрасный излучатель и точечный проектор.

Для электронной системы Face ID и датчика TrueDepth лицо представлено впадинами, выступами, морщинами и шероховатостями. 3D модель не выглядит как лицо, которое привычно видеть на фотографиях, на ней практически невозможно распознать знакомого человека с помощью простого просмотра. При каждом использовании Face ID, модель дополняется данными. В результате накапливается полноценный слепок лица.

Face ID используется при разблокировке, а также при оплате с помощью ApplePay, в покупках AppleStore, а также при входе в различные приложения.

Благодаря глубокому обучению iPhone X смог распознавать мельчайшие детали на лице человека, тем самым “узнавая” владельца. Частота ошибок при использовании Face ID 1:1 000 000, сотрудники Apple, утверждают, что этот результат превосходит ранее представленный компанией Touch ID.

Для регистрации Touch ID требуется от 10 до 15 нажатий на датчик, для Face ID достаточно смотреть на телефон в обычном режиме, далее необходимо повернуть голову по кругу, для того, чтобы зарегистрировать лицо, находящееся в разных позах. Этого достаточного для того, чтобы приступить к разблокировке телефона.

Стандартная задача классификации нейронной сети, которая может выявить является ли лицо пользователя, пытающегося разблокировать телефон, лицом идентичным владельца смартфона, не применима в данном случае. Для этого необходимо использовать некоторые тренировочные данные, содержащие в себе истинные и ложные значения. Для ложных значений сеть должна содержать в себе лица, не принадлежащие пользователю.

В случае такой архитектуры, Apple не сможет тренировать более высокопроизводительную “оффлайн” сеть. Считается, что Face ID основан на сиамской сверточной нейросети, которая обучается “оффлайн”, чтобы отображать лица в низко-размерном скрытом пространстве, созданным для максимизации различия между лицами разных людей.[24]

Так как Apple в своих разработках использовало огромное количество лиц и вычислительных мощностей, модель стала восприимчива к таким случаям, как близнецы, маски и другие подобные способы обхода системы.

Преимущества такого подхода очевидны, Apple разработала модель plug and play (дословно переводится, как установи и играй), которая распознает “хозяина” без долгой подготовки и обработки данных, для этого достаточно зарегистрировать Face ID на устройстве. Технология адаптирована к резким и постепенным изменениям. К резким можно отнести, шапки и шляпы, макияж и очки. К постепенным можно отнести, например, растущую бороду или набор веса. Это делается путем добавления опорных векторов лица, вычисленных на основе нового внешнего вида человека, на его карту.

Специалисты из вьетнамской компании по информационной безопасности Bkav, нашли способ обойти аутентификацию Face ID, с помощью маски, напечатанной на 3D-принтере. Маска не полностью соответствует лицу, была создана объемная модель лица с помощью специального сканера, далее ее распечатали на 3D-принтере. Система детектировала эту модель как владельца, хотя реалистично выглядели только части, созданные из двумерных изображений. [25]

## **1.3 Выводы**

Все описанные методы, кроме Face ID и Touch ID используются в качестве идентификации самого текста, а не человека, его написавшего. В настоящее время методы авторизации пользователя с использованием нейронных сетей используются в основном крупными компаниями, без открытого кода. Поэтому проблема достаточно не изучена, а существующие методы ограничены областью применения и ресурсами.

Хотя часто небольшим и средним компаниям также необходимы наработки качественной авторизации для сохранения данных.

# **2 КОНСТРУКТОРСКИЙ РАЗДЕЛ**

## **2.1 Архитектура алгоритма распознавания рукописного текста**

Под архитектурой понимается базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы. [стандарт IEEE 1471].

На сегодняшний день существует большое количество стандартных архитектур построения нейронных сетей, что существенно облегчает задачу построения нейронной сети с нуля и сводит ее к подбору подходящей конкретной задаче структуре сети. Для реализации описанной системы был выбран ансамбль из десяти сверточных сиамских сетей, ветви которых имеют LeNet-подобную архитектура, также использован расчет проекций по x и y.

Если одна модель недостаточно точно решает задачу классификации, то несколько подобных моделей можно объединить в ансамбль, путем усреднения их голосов. Точность моделей-экспертов по отдельности будет меньше точности набора таких моделей. Подобные техники ансамблирования классифицируются в бэггинг и бустинг [22].

Бэггинг - техника с независимыми друг от друга моделями и их усредненными комбинациями. Обычно для каждой модели используются случайные подвыборки данных, поэтому каждая немного отличается друг от друга. Модели в бэггинге не коррелируют между собой, что приводит к уменьшению разногласий между ними. Для решения задачи рассмотрен этот способ. Его называют bootstrap, bagging или Bootstrapaggregation. Бутстрэп позволяет оценивать статистики сложных распределений, Бэггинг основан именно на этом статистическом методе.

Суть метода заключается в следующем: существует выборка X размера N. Из выборки равномерно берется N объектов с возвращением, вероятность выборки каждого объекта принято считать 1 к N. Каждый новый объект выбирается из N объектов (количество не изменяется). Из-за возвращения в выборке могут оказаться повторы, поэтому новая выборка переименована в X1, после прохождения данной процедуры M раз, получается M подвыборок X1,...,Xm. С новым количеством подвыборок можно рассчитывать статистики исходного распределения [23].

Бустинг используется для построения ансамблей последовательно друг за другом. В данной технике основная идея заключается в обучении последующей модели на ошибках предыдущей. Они имеют неравную вероятность появления в последующих моделях, и чаще появятся те, что дают наибольшую ошибку. Для предсказания используются модели, такие как деревья решений, регрессии, классификаторы и т. д. Так как каждая последующая сеть обучается на ошибках предыдущей модели, получение конечно ответа случается раньше, чем при технике бэггинг. Важным для такого ансамбля считается выбрать верный критерий остановки, чтобы не прийти к переобучению.

Пусть сети будут иметь низкую полноту (отношение верно определенных ко всем определенным), но высокую точность. Тогда за счет ансамбля из нескольких сетей необходимо постараться увеличить полноту, сохраняя при этом качество. Сети имеют от 100 тысяч до 200 тысяч обучаемых коэффициентов, что относительно мало (многие известные архитектуры имеют по 5-10 миллионов коэффициентов). Малое количество коэффициентов дает быструю работу сетей.

Для обучения и тестирования сети использован материал из SigComp2011 (Signature Verification Competitionfor Onlineand Offline ICDAR 2011, используемый в соревнованиях по программированию) [15], который содержит изображения подписей в формате PNG, отсканированные с разрешением 400 точек на дюйм, цвет RGB. Множество подписей было разделено на два подмножества: тестовое и обучающее. Каждая выборка содержала “ложные” и “истинные” подписи.

В качестве основы системы был выбран ансамбль сверточных сиамских нейронных сетей.

Сиамская нейронная сеть - это класс архитектур нейронных сетей, которые содержат две или более идентичных подсетей. идентичные здесь означает, что они имеют одинаковую конфигурацию с одинаковыми параметрами и весами. Обновление параметров отражается в обеих подсетях. Сиамские нейронные сети популярны среди задач, которые включают в себя поиск сходства или взаимосвязи между двумя сопоставимыми вещами.

Эта архитектура может научиться различать расстояния между конкретными данными, такими как изображения. Идея состоит в том, чтобы передавать пары данных через сиамские сети (или просто передавать данные в два разных шага через одну и ту же сеть), сеть отображает их в низкоразмерных характеристиках пространства, как n-мерный массив, а затем сеть обучается, чтобы сделать такое сопоставление, что данные точек из разных классов были как можно дальше, в то время как данные точек из одного и того же класса находились как можно ближе.

В результате сеть научится извлекать наиболее значимые функции из данных и сжимать их в массив, создавая изображение.

Сиамские архитектуры хороши в задачах распознавания почерка, так как:

1. Разделение весов между подсетями означает меньшее количество параметров для обучения, и это, в свою очередь, означает, что требуется меньше данных и меньшая склонность к переобучению;
2. Каждая подсеть производит представление своего входного вектора. Если входные данные имеют одинаковый вид, например, совпадают два предложения или совпадают два изображения, имеет смысл использовать аналогичную модель для обработки аналогичных входных данных. Таким образом, есть векторы представления с одинаковой семантикой, что облегчает их сравнение.

## **2.2 Математическая модель метода распознавания рукописного текста**

В каждой сети из ансамбля происходит инициализация весов модели случайными значениями. Далее необходимо настроить параметры модели. Для этой цели используют метод обратного распространения ошибки (backpropagation).

Существует разновидности градиентного спуска: обычный, стохастический и «mini-batch». Для первого, прежде чем он сможет менять параметры, необходим один проход по обучающим данным.

Для стохастического достаточно вычисленного градиента только на одном элементе обучения, далее параметры изменяются пропорционально приближенному градиенту, после каждого объекта обучения. Данный метод значительно экономит время.

Третий вид градиентного спуска иногда называют пакетный градиентный спуск (mini-batch). Он совмещает в себе плюсы методов, описанных выше. Вычисляет один шаг, используя сразу весь набор данных, тогда как стохастический за шаг использует только один элемент. Объединяя оба метода получается пакетный спуск, который за раз обрабатывает, часть элементов, а не все или один. Веса в свободном порядке распределяются по группам и меняются на сумму разности всех весов в той или иной группе.[5]

В работе используется способ градиентного спуска.

Далее производится проход вперед, т.е. картинка обрабатывается последовательно указанными слоями. Сверточные слои обрабатывают картинку с помощью фильтров (которые содержат веса). У каждой сети есть свой заранее заданный размер фильтра. Благодаря их разным размерам, каждая сеть может детектировать разные признаки.

Сверточный слой из входной картинки получает выходную и, если размер фильтра 3x3, то каждая точка на выходной картинке рассчитана из области 3x3 исходной картинки (*Рисунок 11*).

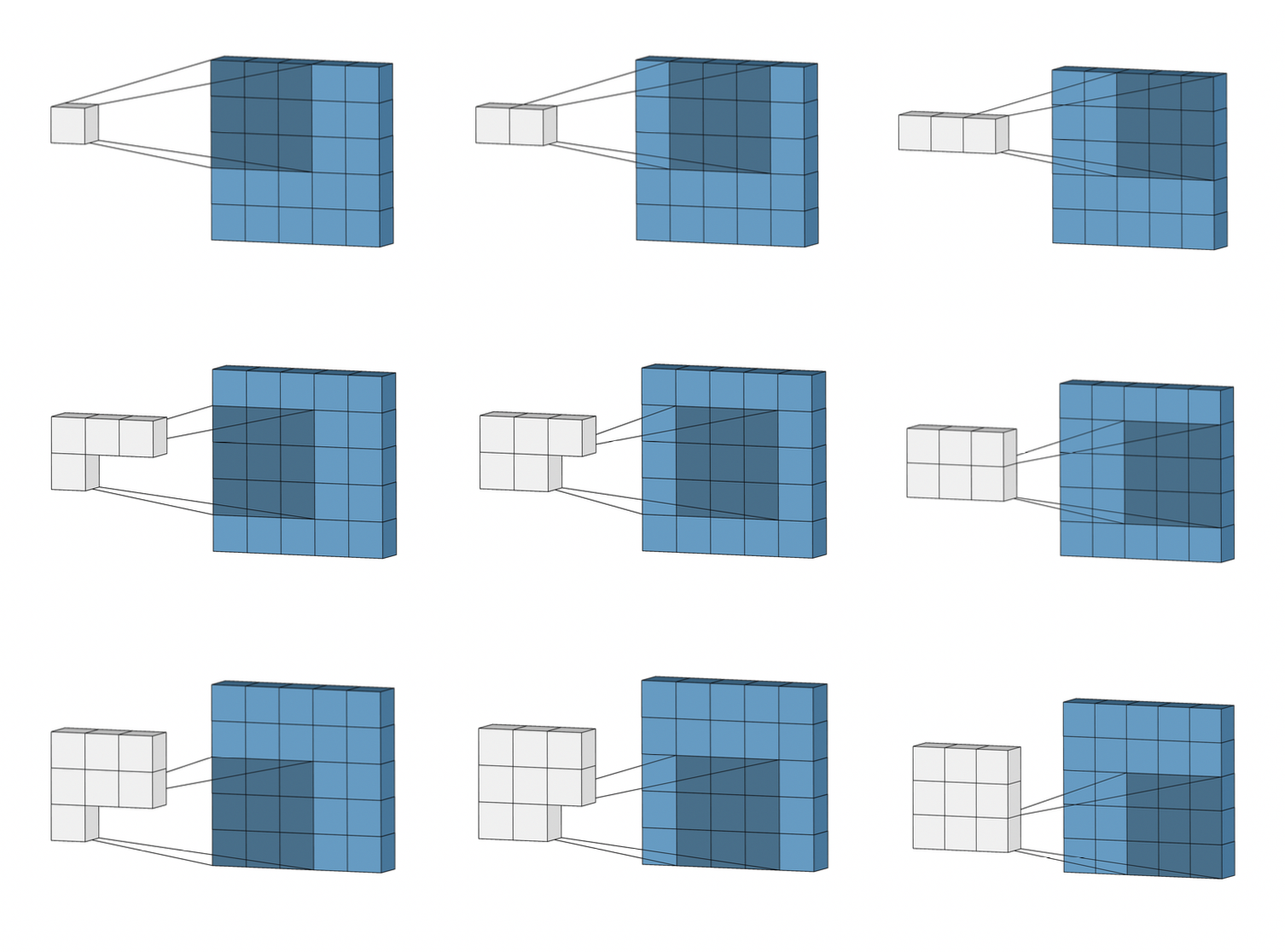


Рисунок 11 Двумерная свертка

В двумерном случае область проходит над двумерным изображением на входе, поэлементно выполняя операцию умножения с той частью входных данных, над которой оно сейчас находится, и затем суммирует все полученные значения в один выходной пиксель. На *Рисунок 12* изображен проход фильтра размер 2х2, с весами равными 0,5.

|  |  |
| --- | --- |
|  | (2.1) |

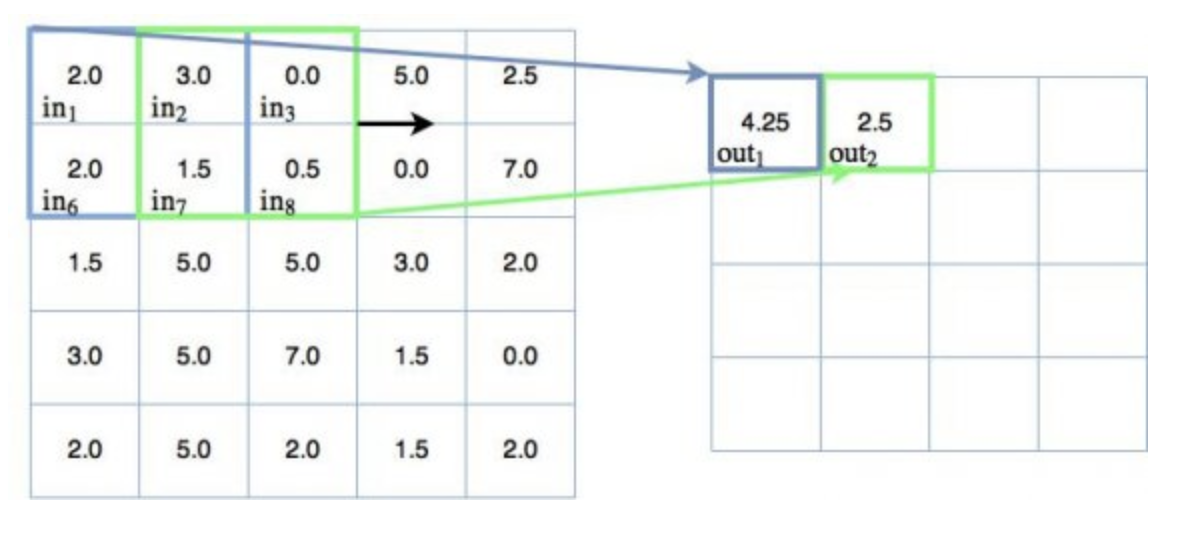


Рисунок 12 Результат прохождения фильтра 2х2

После прохождения матрицы фильтров по входной матрицы, получаются признаки, которые являются взвешенными суммами (веса - это значения матрицы фильтра). Фильтры, применяемые на входных узлах, не меняют свои веса, поэтому могут обучиться выявлять определенные признаки, например, различные формы исходного объекта.

Размер ядра сверточной нейронной сети определяет количество признаков, которые будут объединены для получения нового признака на выходе. Следовательно, для любого сверточного слоя необходимо множество фильтров. Дополним *Рисунок 12* несколькими сложенными (stacked) выходами операции свертки. Их несколько, потому что существует несколько обучаемых фильтров, каждый из которых производит собственный 2D выход. Множество фильтров называют каналами.[6]

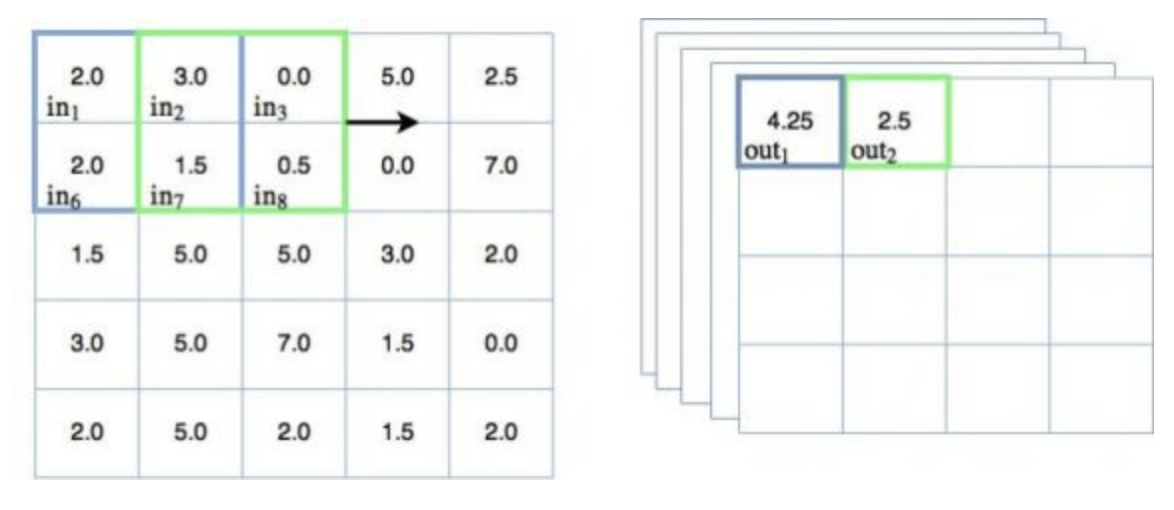


Рисунок 13 Сложенные выходы операции свертки

Так как крайние пиксели не могут попасть в центр ядра, а информацию на них нужно сохранить, техника Padding добавляет к краям поддельные нулевые пиксели, которые не влияют на результат, но при этом сохраняют центр у исходных пикселей (*Рисунок 14*)

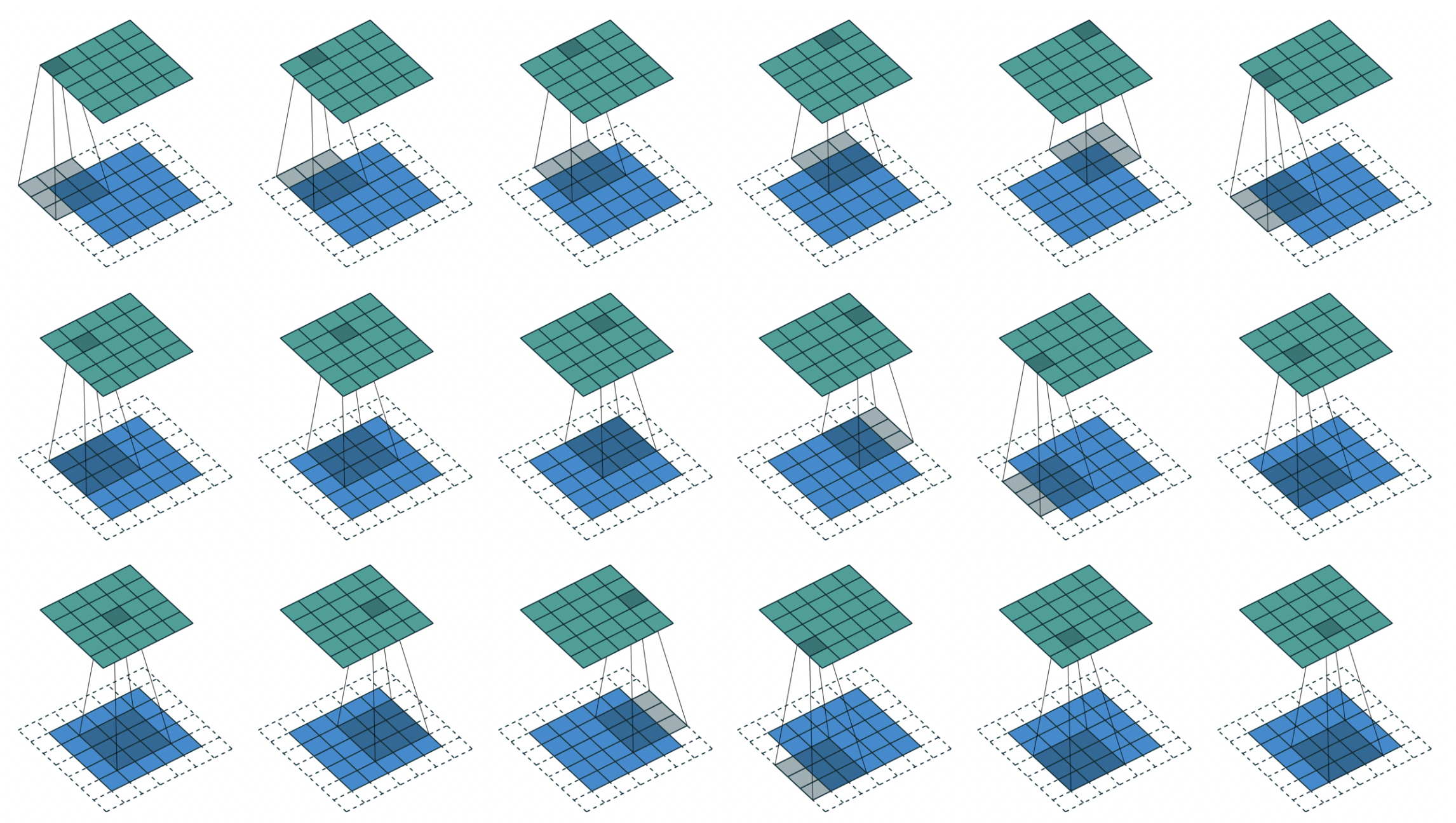


Рисунок 14 Техника Padding

После прохождения данного алгоритма находится проекция на ось x и y, которая считает количество пикселей по осям. Для этого были подобраны размеры фильтров так, чтобы итоговая картинка после сверток по ширине была равна 1, и по высоте также равнялась 1.

Для случаев, когда сверточных слоев было несколько подряд, существует понятие рецептивное поле - это размер окошка на входе, которое нужно подать, чтобы получить определенный пиксель на выходе. Калькулятор рецептивных полей [13].

После каждого сверточного слоя идет функция активации.

Функция активации (ФА) применяется для нормализации входных данных. Если на входе задано большое число, или диапазон значений слишком велик, есть возможность скорректировать ФА.

Выбор вида функции активации влияет на функциональные возможности и метод обучения нейронной сети. Например, метод обратного распространения ошибки может хорошо работать на двухслойных и трехслойных сетях, при дальнейшем увеличении глубины алгоритм испытывает трудности в связи с затуханием градиента.

Sigmoid, линейная, ReLu и гиперболический тангенс самые популярные функции активации, отличающиеся диапазоном значений и областью применения.

Производная у сигмоидальной функции меньше единицы на всей области определения, поэтому после нескольких итераций ошибка будет стремиться к нулю. А так как ошибка распространяется от входного слоя к выходному путем домножения текущего результата на производную функции активации, то уже после второго или третьего слоя результаты перестанут быть точными.

Для функции с неограниченной производной, например, как, гиперболический тангенс, значение функции ошибки может начать расти на каждом шаге, что приведет к неустойчивому обучению.

Функция активации гиперболический тангенс используется только в случае, когда на вход подаются как отрицательные, так и положительные значения, ее использование только при положительных значениях ухудшает результаты сети. Диапазон значений функции: [-1,1]

|  |  |
| --- | --- |
|  | (2.2) |

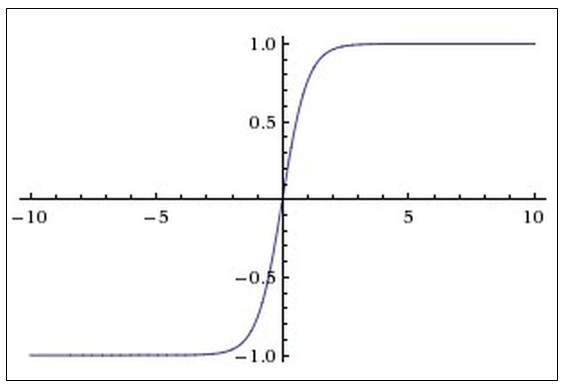


Рисунок 15 - График функции активации (гиперболический тангенс)

Из достоинств данной ФА обычно выделяют простое вычисление производной, область значений [-1,1]. К недостаткам относятся затухание или увеличение градиентов, а также значительную ресурсоемкость, по сравнению с ReLu.

Функция активации ReLu, обычно ее называют выпрямитель. Она реализует простой пороговый переход в нуле. И имеет функцию:

|  |  |
| --- | --- |
|  | (2.3) |

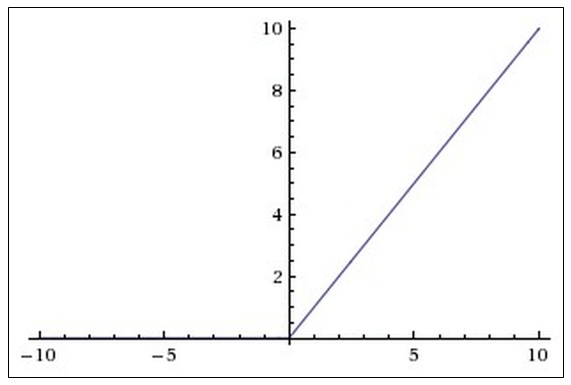


Рисунок 16 График функции активации (ReLu)

Из достоинств часто выделяют отсутствие ресурсоемких операций, простое отсечение “ненужных” деталей [21], отсутствие быстрого разрастания/затухания градиента (так как ее производная равна или нулю, или единице) и быстрое обучение. К недостаткам относят случаи недостаточной надежности и сильную зависимость от инициализации весов.

В данной работе в качестве функции активации в сверточных слоях применяется ReLU, а на выходе в последней итерации применяется sigmoid. Диапазон значений функции принадлежит отрезку [0,1]. Большие по модулю отрицательные числа функция превращает в 0, большие положительные в 1. Функция имеет следующий вид:

|  |  |
| --- | --- |
|  | (2.4) |

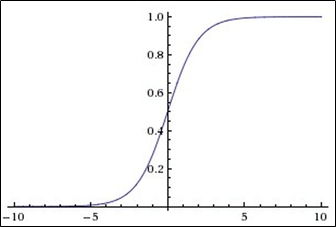


Рисунок 17 График ФА Sigmoid

Сигмоида непрерывная, монотонно возрастающая, дифференцируемая функция.

Из неблагоприятных свойств сигмоиды выделяют насыщение функции с той или иной стороны (0 или 1), градиент на этих участках становится близок к нулю.

После функции активации на начальных этапах применяются слои объединения (pool). Слои объединения необходимы для уменьшения числа вычислений. Например, картинка 100x100 большая, у нее значения ближайших пикселей примерно одинаковое, поэтому к ней можно применить слой объединения. А с уменьшением размера картинки необходимо увеличивать количество фильтров, чтобы находить больше признаков.

Pooling это разновидность техники скользящего фильтра по картинке, где не применяются веса, а используется статистическая функция. Например, есть maxpooling и meanpooling, в первом применяется функция нахождения максимума, для второго находится среднее. В работе использовался maxpooling, принцип действия изображен на Рисунок 18.

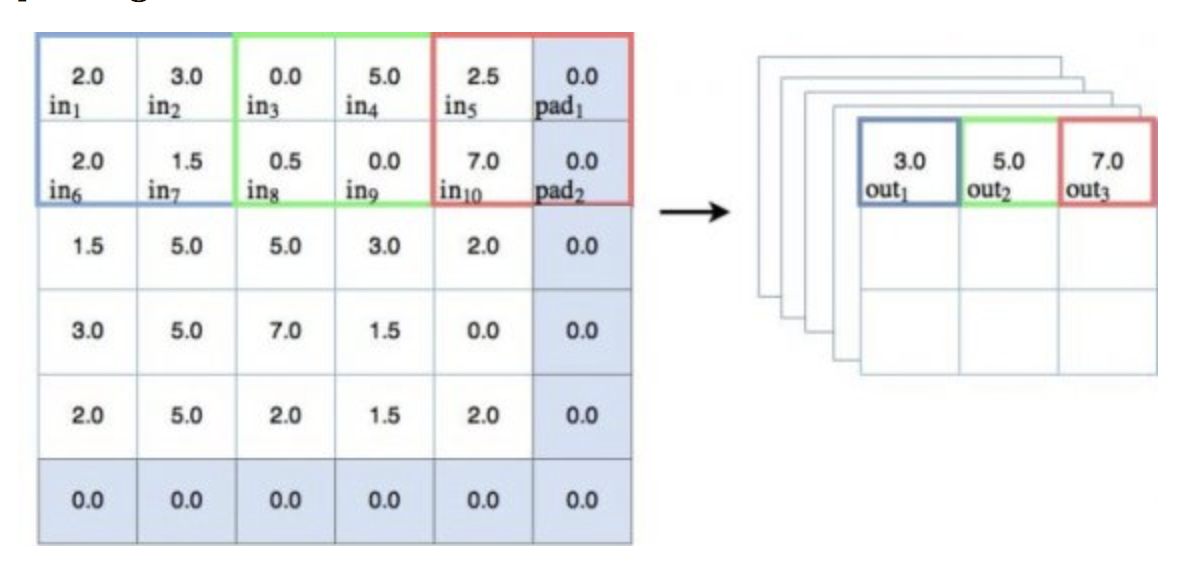


Рисунок 18 MaxPooling

На Рисунок 18 изображен проход фильтра, используя статистическую функцию max.

|  |  |
| --- | --- |
|  | (2.5) |

Шаг окна для объединения слоев равен 2.

На Рисунок 19 изображены шаги вдоль оси x, для предотвращения перекрытия окна, по оси y должен быть такой же шаг, равный двум. Размер выхода станет меньше, так как шаг больше 1. На рисунке 6 размер окна на выходе стал равен 3х3. Уменьшение количества обучаемых параметров в модели облегчает ее, это называется даунсемплингом.

Pooling обобщает низкоуровневые данные и дает возможность нейросети переходить от данных с высоким разрешением до информации с более низким разрешением. Благодаря pooling и сверточным фильтрам можно детектировать объект на изображении [14].

После завершения прохода вперед вычисляется функция ошибки. Для нее есть ответ сети (вероятность от 0 до 1 того что картинки одинаковые) и правильный ответ (1 или 0). На основе этих значений считается функция ошибки.

В качестве целевой функции определения ошибки была выбрана модифицированная Binary cross entropy (BCE new)

|  |  |
| --- | --- |
| *,* | (2.6) |

где - это ответ сети, полученный в ходе обработки данных слоями;

- это ожидаемый ответ сети, каким он должен быть, он указывается на этапе генерации пар т.к. при обучении известно какие картинки находятся в паре, будет ли пара позитивной или негативной.

Недостатки классической функции ошибки BCE:

1. Не учитывает разницы между негативными и позитивными парами обучающей выборки (позитивных пар меньше в три раза, чем негативных);
2. Не удовлетворяет высокой точности (highprecision), т.к. сеть, по результатам тестирования “запоминала” ответы. Поэтому необходимо использовать коэффициент и для получения модифицированной функцию ошибки.

Учитывая указанные выше недостатки, функция была модифицирована с целью увеличения ответственности нейронной сети за FP (ложные положительные срабатывания)

Пусть

|  |  |
| --- | --- |
|  | (2.7) |
|  | (2.8) |

Коэффициент отвечает за FP - ложно-положительное решение. В ходе работы было принято штрафовать сеть за частое срабатывание FP. Если сеть дает положительный ответ, то пусть это будет максимально уверенно, т.к. в приложениях идентификации цена ошибки достаточно высока. В связи с этим, необходимо было найти коэффициент, который мог бы “штрафовать” модель за ложно-положительные решения, при этом он должен зависеть от качества модели.

Для оценки качества моделей использует ряд метрик, таких как Accuracy (достоверность), Precision (точность), Recall (полнота), F1 (F Мера- среднее гармонической точности и полноты), ошибки первого рода - вероятность ложного сопоставления с объектом в базе (FAR - falseaccessrate) и ошибки второго рода - вероятность отказа распознавания объекта, находящегося в базе (FRR — falserejectrate).

Например, одной из метрик оценки качества классификатора является отношение количества подписей, по которым классификатор принял правильное решение к размеру обучающей выборки, такая метрика называется точность (Accuracy).

|  |  |
| --- | --- |
|  | (2.9) |

где, P – количество подписей по которым классификатор принял правильное решение, а N – размер обучающей выборки.

К особенностям этой метрики можно отнести то, что она присваивает одинаковые веса всем видам подписей, что не работает для задач, в которых распределение на обучающей выборке смещено в сторону одного или нескольких классов. По ним у классификатора накапливается больше информации, в связи с чем в рамках таких классов, классификатор выдает более взвешенные решения. Опасность использования метрики достоверности в задачах заключается в том, что показатель может быть равен 80% или 90% на выходе, но при этом выдавать показать в 30% в рамках некоторых классов.

Для решения этой проблемы нужно обучить классификатор на полностью сбалансированных данных, последствия такого подхода могу забрать необходимую информацию об относительной частоте подписей, которая может быть при прочих равных условиях важна для принятия правильного решения.

Точность (Precision) - это доля подписей действительно принадлежащих данному классу относительно всех подписей которые система отнесла к этому классу.

Полнота системы (Recall) – это доля найденных классификатором подписей принадлежащих классу относительно всех подписей этого класса в тестовой выборке.

|  |  |
| --- | --- |
|  | (2.10) |
| , | (2.11) |

где TruePositive — истинно-положительное решение;

TrueNegative — истинно-отрицательное решение;

FalsePositive — ложно-положительное решение;

FalseNegative — ложно-отрицательное решение.

Эти значения легко рассчитать на основании таблицы контингентности, которая составляется для каждого класса отдельно.

|  |  |
| --- | --- |
|  | (2.12) |
|  | (2.13) |

Таблица 1 Таблица контингентности

|  |  |  |  |
| --- | --- | --- | --- |
| Категория | | **Оценка эксперта** | |
| положительная | отрицательная |
| **Оценка системы** | положительная | TruePositive | FalsePositive |
| отрицательная | FalseNegative | TrueNegative |

FalsePositive и FalseNegative, в статистике первый вид ошибок называют ошибкой I-го рода, а второй — ошибкой II-го рода.

В таблице содержится информация сколько раз система приняла верное и сколько раз неверное решение по подписям заданного класса.

F-мера - еще одна метрика качества классификатора, которая объединяет внутри себя полноту и точность. Максимальная полнота и точность недостижимы на реальных данных, но F-мера может выявить лучшее решение. F-мера представляет собой среднее гармоническое между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

|  |  |
| --- | --- |
|  | (2.14) |

В формуле выше дан одинаковый вес точности и полноте, в связи с чем, при падении, например точности, F-мера будет падать пропорционально. Если в задаче есть четкое понимание, что более важным фактором является полнота/точность, то вводится дополнительный коэффициент и формула приобретает вид:

|  |  |
| --- | --- |
| , | (2.15) |

где β принимает значения в диапазоне 0<β<1 для приоритезации точности;

при β>1 приоритет отдается полноте;

при β=1 формула сводится к предыдущей и получается сбалансированная F-мера (также ее называют F1).

Для оценки возьмем текущей задачи возьмем полноту системы (recall). И введем коэффициент в формулу бинарной кросс энтропии.

|  |  |
| --- | --- |
|  | (2.16) |

Если нейронная сеть становится более чувствительной к выборке, то и точность должна быть соответственно высокой.

Функция используется в расчетах только с фиксированными коэффициентами, то есть при дифференцировании они представлены константами и формула ошибки представлена весовой бинарной кросс энтропией.

Далее проход назад - имея численное значение функции ошибки, можно вычислить градиенты весов на каждом слое, для их последующего обновления. Значение весов будут меняться в ту сторону, которая даст нам наилучший результат. Данный метод называется методом обратного распространения ошибки и используется для обучения многослойных нейросетей.

Данный алгоритм проходит два раза по всем слоям сети - вперед и в обратную сторону. При прохождении вперед вектор на входе подается на входной слой нейронной сети и распространяется по всем последующим слоям от слоя к слою последовательно. Далее формируется выход, который является ответной реакцией сети на вектор, поданный на вход. Во время прохода вперед все инициализированный веса зафиксированы и менять свое значение не могут. Во время прохода в обратную сторону происходит настройка весов, путем вычисления разницы между полученным и желаемым выходом, тем самым формируя сигнал ошибки, которая в дальнейшем распространяется в обратную сторону и меняет значения весов с целью приблизить результат на выходе к желаемому в последующих проходах по сети вперед.

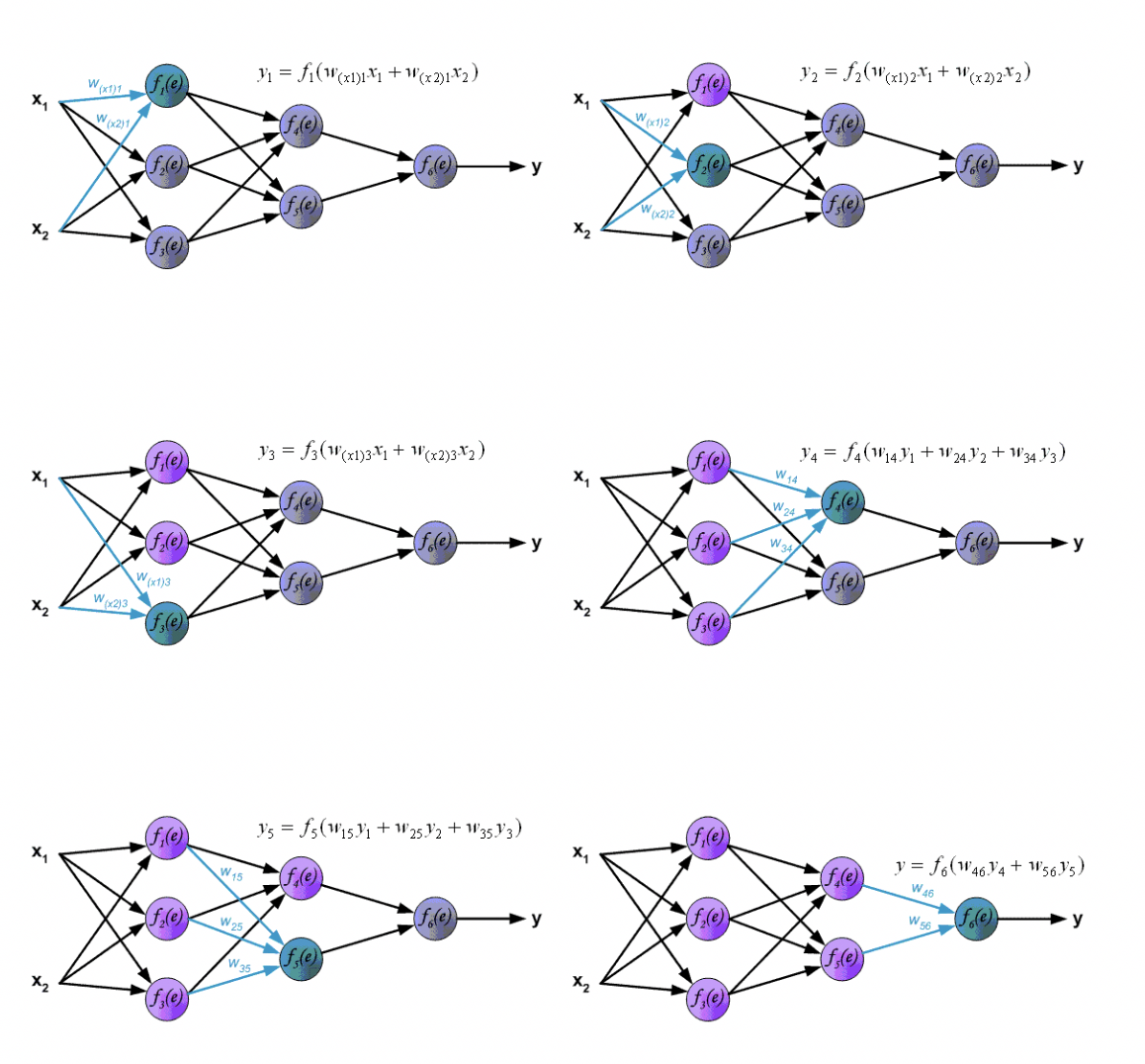


Рисунок 19 Предсказание сети с использованием существующих весов

На Рисунок 19 изображен проход вперед с использованием инициализированных весов. На каждом шаге находится сумма входных значений, умноженная на веса. После этого сумма умножается на функцию активации, в работе используется ReLu. Чтобы найти значение сети на выходе и сравнить его с желаемым результатом.

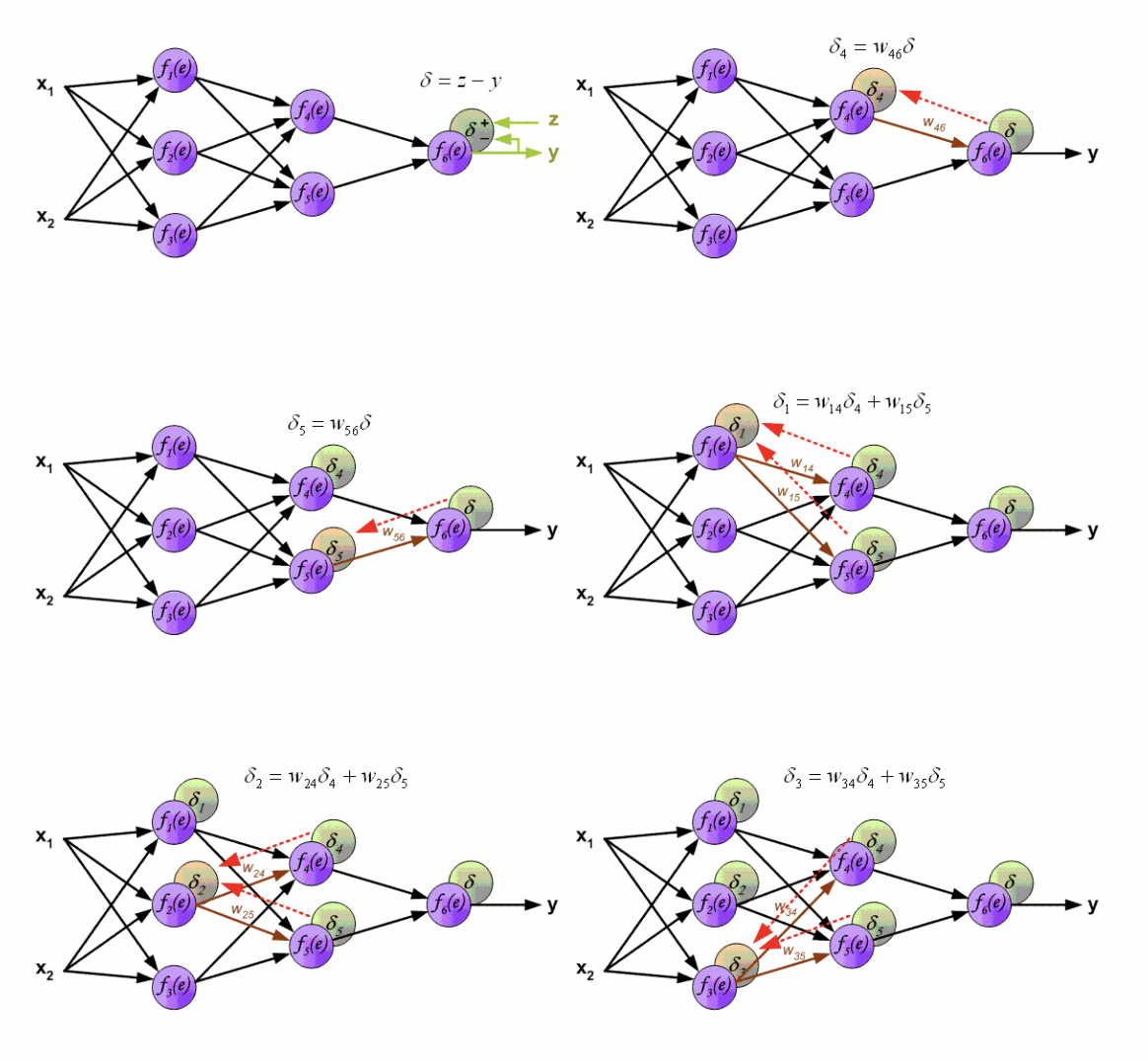


Рисунок 20 Процесс подготовки значений на нейронах перед изменением весов

На Рисунок 20 представлено нахождение разницы между желаемым и полученным результатом и умножение на производную функции активации от входного значения данного нейрона. После подсчета дельты нейрона необходимо обновить веса всех исходящих синапсов этого нейрона (*Рисунок 21*). Умножение производной функции активации от входного значения на сумму произведений всех исходящих весов и дельты нейрона с которой этот синапс связан.

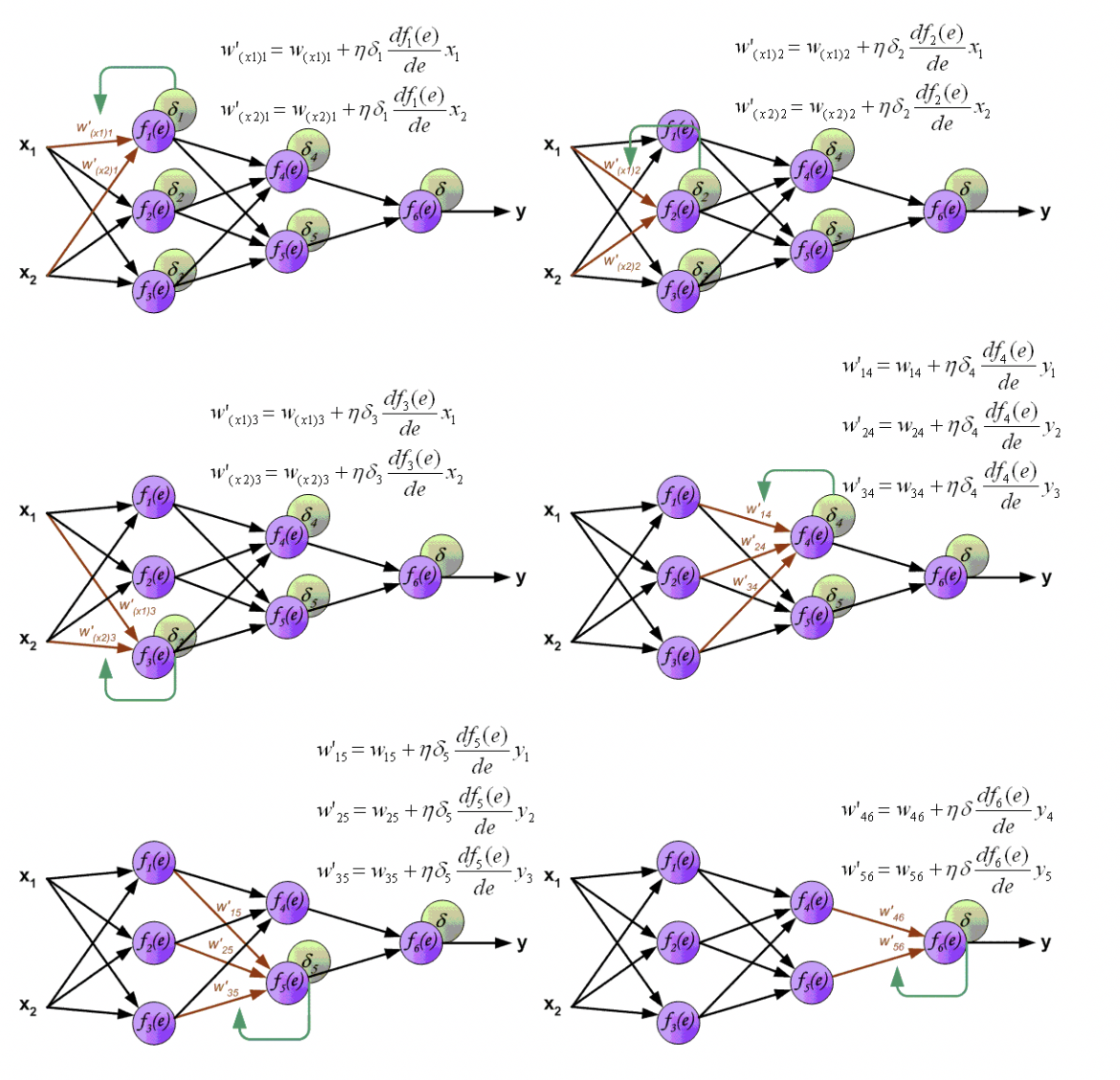


Рисунок 21 Процесс изменения весов

## **2.3 Состав ансамбля нейронных сетей**

Архитектуру для каждой ветви сверточной сиамской нейронной сети из ансамбля можно представить в виде таблицы 2.

Таблица 2 - Архитектура ветви нейронной сети

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Функция активации | Параметры слоя |
| 1 | Сверточный | ReLU | 8 фильтров 3х3, шаг 1х1, без отступов |
| 2 | Сверточный | ReLU | 8 фильтров 3х3, шаг 1х1, без отступов |
| 3 | MaxPooling |  | размер фильтра 3х3 |
| 4 | Сверточный | ReLU | 16 фильтров 5х5, шаг 2х2, отступ 2х2 |
| 5 | Сверточный | ReLU | 16 фильтров 5х5, шаг 2х2, отступ 2х2 |
| 6 | MaxPooling |  | размер фильтра 3х3 |
| 7 | Сверточный | ReLU | 16 фильтров 3х3, шаг 1х1, отступ 1х1 |
| 8 | Сверточный | Sigmoid | 8 фильтров 3х3, шаг 1х1, отступ 1х1 |
| 9 | Fullconnected |  |  |

В каждой из десяти сетей есть две ветви с архитектурой, указанной в таблице 2. Главное отличие между сетями - размер окна фильтров и шаг, благодаря чему сети могут распознавать различные признаки у подписей. После прохода по данной архитектуре по каждой из ветви считается 128-мерный полносвязный вектор.

Следующим шагом рассчитывается L1 норма (поэлементная разность векторов), получается полносвязный слой, содержащий 1 число - 1 нейрон, далее он активизируется функцией активации Sigmoid для приведения к диапазону [0,1], что будет вероятностью принадлежности подписи к истинной. [26]

## **2.4 Описание обучающей и тестовой выборок**



Рисунок 22 Образец подписей тестовой выборки

Выше представлены шесть образцов подписей из набора данных SigComp2011 [15]. Верхний ряд: подписи из китайского подмножества, нижний ряд: подписи из голландского подмножества. В связи с тем, что китайские подписи имеют сильную специфику и их распознавание желательно делать специально под них, было принято решение оставить для работы подписи только голландского подмножества.

В наборе исходных данных есть как online так и offline подписи. Так как для решения задачи было решено использовать только конечную подпись, без процесса ее написания, то online картинки убраны из исходных данных. Online метод распознавания подразумевает известность, того как именно рисовалась подпись (видео написания).

Структура данных имеет следующий вид: папка data\_signatures содержит полностью исходные данные, внутри 2 папки – train и test. Папка train содержит папки Questioned(1287) и Reference(646). Формат названий картинок – ImageNumber\_PersonID.png. В Reference(646) лежат отсортированные по людям подписи, причем там только истинные подписи – т.е. заходя в папку с индексом определенного человека – можно быть увереным, что там находятся только подписи, сделанные этим человеком лично. В папке Questioned(1287) находятся как истинные подписи, так и подделки – стоит заметить, что класс у них разный. Пример 2 подписей человека под индексом 013:

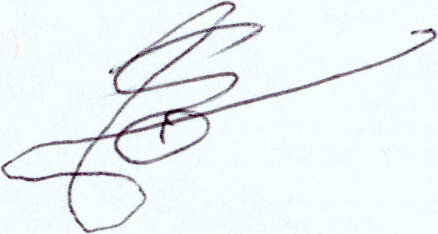


Рисунок 23 Подделка и оригинал подписи человека с индексом 013

Левая имеет название 04\_0203013.PNG, тогда как правая 07\_013.PNG. Т.к. они оба находятся в одной папке с индексом человека 013, но имеют в названии файлов разные индексы людей (0203013 и 013) – то левая картинка является подделкой подписи человека с индексом 013. Правая при этом оригинальная подпись.

Была выбрана архитектура и математическая модель метода распознавания рукописного текста.

Введена модернизированная функция ошибки, которая позволяет “штрафовать” модель, в случае, когда происходит много ложных срабатываний. Протестирован набор окон фильтров, а также сформирована архитектура всех нейронных сетей из ансамбля.

# 

# **3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ**

## **3.1 Детали подготовки данных для обучения и тестирования системы**

Разработка проводилась на языке программирования Python, т.к. он является высокоуровневым языком программирования общего назначения, ориентированным на повышение производительности разработчика и читаемости кода. Синтаксис Python прост и понятен.

Плюсы использования:

* большое количество готовых библиотек, ориентированных на нейронные сети;
* интерпретатор Python реализован практически на всех платформах и операционных системах;
* расширяемость языка, этому придается большое значение и, как пишет сам автор, язык был задуман именно как расширяемый. Интерпретатор написан на С и исходный код доступен для любых манипуляций;

Кроме этого, для построения нейронных сетей использовались библиотеки TensorFlow и Keras. Сети были обучены с использованием GoogleColabLaboratory.

TensorFlow — это открытая программная библиотека машинного обучения для решения задач построения и тренировки нейронной сети с целью автоматического выделения и классификации образов, достигая качества человеческого восприятия.

Keras — открытая библиотека для нейронных сетей, на языке Python. Она нацелена на работу с сетями глубинного обучения, и созданная так, чтобы быть компактной, расширяемой и модульной.

GoogleColabLaboratory - это бесплатная среда, которая не требует настройки и полностью работает в облаке, дает доступ к мощным вычислительным ресурсам.

Для реализации были использованы Python библиотеки cycler 0.10.0, decorator 4.4.0, imageio 2.5.0, kiwisolver 1.1.0, matplotlib 3.1.0, networkx 2.3, NumPy 1.16.3, opencv-python 4.1.0.25, Pillow 6.0.0, pyparsing 2.4.0, python-dateutil 2.8.0, PyWavelets 1.0.3, scikit-image 0.15.0, scipy 1.3.0, six 1.12.0.

Matplotlib - бибилиотека на языке Python, позволяющая визуализировать изображения.

Библиотека NumPy используется для работы с многомерными массивами, поддерживает высокоуровневые математические функции для работы с массивами.

SciPy дает возможность искать минимумы и максимумы функций, обрабатывать изображения.

OpenCV — библиотека машинного зрения, которую часто применяют в роботах для распознавания объектов.

Pillow (PythonImagingLibrary) поддерживает бинарные, полутоновые, индексированные, полноцветные и CMYK изображения. Дает возможность преобразовывать изображения из одного формата в другой, используется для внесения правок в изображение (различные фильтры, масштабирование, рисование, матричные операции и т. д.).

**Предварительная обработка**

Для реализации была необходима предварительная обработка данных:

1) Изменение размеров образцов к размеру 100x100 пикселей представлена на *Рисунок 25*. Размер выбирался таким образом, чтобы при меньшем размере картинок (ускорение работы системы) сохранились все мелкие детали, исходная картинка подписи изображена на *Рисунок 25*.



Рисунок 24 Первоначальная подпись



Рисунок 25 Изменение образца к размеру 100x100 пикселей

2) Бинаризация картинки – преобразование фона в белый цвет, а подписи в черный (2 возможных значения). Т.к. картинки хорошего качества, то пороговая бинаризация была выбрана. В качестве порога взято число 230. (у исходного образца 3 канала (т.е. цветная). Образец конвертируется в серую картинку (становится 1 канал). Значения картинки принимают значения от 0 до 255, где 0 это черный цвет, а 255 белый. Для тестирования и обучения необходимы только значения 0 и 255. Для этого все значения, меньше 230 были преобразованы в 0, а все что больше 230 в 255).



Рисунок 26 Результат бинаризации картинки

3) Инверсия цвета. Замена цвета подписи на белый, а фон на черный (*Рисунок 26*).



Рисунок 27 Результат инверсии цвета

4) Утончение. Толщина линий на подписи была преобразована в 1 пиксель. Необходимо для удаления влияния на написание таких факторов, как качество бумаги, толщина ручки/стилуса/пера, и т.д. (*Рисунок 27*).



Рисунок 28 Образец подписи после предварительной обработки

Далее генерируются пары для обучения и для теста. В папке pairs лежат сгенерированные пары картинок на основе SigComp11\_offline\_prcd.

Обучение сиамских сетей делается на основе определения схожести 2 картинок, поэтому для обучения нужны пары одинаковых и разных классов, склеим предварительно обработанные картинки. В итоге получаются пары двух классов – когда две подписи принадлежат одному человеку (позитивные пары), и когда они принадлежат разным людям (негативные пары). Примеры сгенерированных пар:



Рисунок 29 Типичная позитивная пара



Рисунок 30 Типичная негативная пара



Рисунок 31 Негативная пара

Несмотря на схожесть подписей на Рисунок 31пара является негативной, т.к. были взяты настоящая подпись и подделка.

Описанные сети отличаются размерами рецептивного поля – это то “окно”, которое видит сеть с помощью сверток на исходном изображении. Маленькое рецептивное поле – больше внимания мелким деталям. Большое рецептивное поле – большее внимание общему внешнему виду подписи.

## **3.2 Оценка качества работы метода**

Тестирование нейронной сети производится с использованием примеров тестовой выборки из описанной библиотеки.

На тестовой выборке достаточно использовать классификатор на подписях и соотнести его решение с заведомо известным правильным решением. Но для того, чтобы принимать решение хуже или лучше справляется с работой новая версия алгоритма нам необходима численная метрика его качества.

Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда эти метрики используются по отдельности, иногда в качестве базиса для производных метрик, таких как F-мера или R-Precision.

В таблице 4 показан пример результирования коэффициентов нейросети.

Таблица 4 Пример результирования коэффициентов

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № картинки | NN1 | NN2 | NN3 | NN4 | NN5 | NN6 | NN7 | NN8 | NN9 | NN10 | Prj (x,y) | Итог |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |  |
| 2 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |  |
| 3 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |  |
| Итог | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

Если выполняется условие Count(True)>Count(False), то подписи являются сопоставимыми.

Чтобы выявить важность каждой сети, относительно остальных применялась следующие расчеты, указанные в таблицах 3, 4 и 5. Это было сделано с целью убрать возможность одинакового обучения нескольких сетей одновременно.

Таблица 3 Расчет ответов в % всех НС, исключая первую

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метрика | 1 | 2NN | 3NN | 4NN | 5NN | 6NN | 7NN | 8NN | 9NN | 10NN | Prj (x,y) |
| % F сети, при TN = F/F | 0 | 90,41 | 92,38 | 91,77 | 90,10 | 91,70 | 92,20 | 93,00 | 93,3 | 91,66 | 93,8 |
| % T сети, при TN = F/F | 0 | 3,68 | 1,71 | 2,32 | 4,02 | 2,44 | 1,94 | 1,00 | 0,79 | 2,43 | 0,34 |
| % F сети, при TP = T/T | 0 | 0,34 | 0,46 | 0,63 | 0,16 | 0,20 | 0,49 | 0,80 | 0,82 | 1,18 | 0,91 |
| % T сети, при TP = T/T | 0 | 1,85 | 1,73 | 1,56 | 2,03 | 1,98 | 1,70 | 1,40 | 1,36 | 1,01 | 1,28 |
| % F сети, при FP = T/F | 0 | 0,02 | 0,04 | 0,01 | 0,04 | 0,00 | 0,01 | 0,10 | 0,06 | 0,07 | 0,07 |
| % T сети, при FP = T/F | 0 | 0,08 | 0,06 | 0,09 | 0,07 | 0,10 | 0,10 | 0,10 | 0,05 | 0,04 | 0,04 |
| % F сети, при FN = F/T | 0 | 1,92 | 2,40 | 2,13 | 2,20 | 2,00 | 2,78 | 2,70 | 2,9 | 3,18 | 3,15 |
| % T сети, при FN = F/T | 0 | 1,70 | 1,22 | 1,49 | 1,42 | 1,62 | 0,84 | 0,90 | 0,72 | 0,44 | 0,47 |
|  | 0 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

В таблице 3 колличество F сети когда TN = F/F значит следующее: сколько сеть в процентах указала False, когда правильный ответ –False, и т.д.

Данная оценка выбрана для того, чтобы оценить насколько часто одна взятая сеть была “права” в смысле итогового решения.

Рисунок 32 Процент ответов нейросетей, исключая первую

Таблица 4 Расчет ответов в абсолютных значениях всех НС, исключая первую

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метрика | 1 | 2NN | 3NN | 4NN | 5NN | 6NN | 7NN | 8NN | 9NN | 10NN | Prj (x,y) |
| количество F сети когда TN =F/F | 0 | 14714 | 15035 | 14936 | 14658 | 14916 | 14998 | 15147 | 15185 | 14917 | 15257 |
| количество T сети когда TN = F/F | 0 | 599 | 278 | 377 | 655 | 397 | 315 | 166 | 128 | 396 | 56 |
| количество F сети когда TP = T/T | 0 | 55 | 75 | 102 | 26 | 33 | 80 | 125 | 134 | 192 | 148 |
| количество T сети когда TP = T/T | 0 | 301 | 281 | 254 | 330 | 323 | 276 | 231 | 222 | 164 | 208 |
| количество F сети когда FP = T/F | 0 | 4 | 7 | 2 | 6 | 0 | 1 | 9 | 9 | 11 | 11 |
| количество T сети когда FP = T/F | 0 | 13 | 10 | 15 | 11 | 17 | 16 | 8 | 8 | 6 | 6 |
| количество F сети когда FN = F/T | 0 | 312 | 391 | 346 | 358 | 325 | 452 | 440 | 472 | 517 | 512 |
| количество T сети когда FN = F/T | 0 | 277 | 198 | 243 | 231 | 264 | 137 | 149 | 117 | 72 | 77 |
|  | 0 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 |

Таблица 5 Расчет полученных TP, TN, FP, FN для всех НС, исключая первую

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метрика | 1 | 2NN | 3NN | 4NN | 5NN | 6NN | 7NN | 8NN | 9NN | 10NN | Prj (x,y) |
| TP | 0 | 578 | 479 | 497 | 561 | 587 | 413 | 380 | 339 | 236 | 285 |
| TN | 0 | 14718 | 15042 | 14938 | 14664 | 14916 | 14999 | 15156 | 15194 | 14928 | 15268 |
| FP | 0 | 612 | 288 | 392 | 666 | 414 | 331 | 174 | 136 | 402 | 62 |
| FN | 0 | 367 | 466 | 448 | 384 | 358 | 532 | 565 | 606 | 709 | 660 |
|  | 0 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 | 16275 |
| Precision |  | 0,49 | 0,62 | 0,56 | 0,46 | 0,59 | 0,56 | 0,69 | 0,71 | 0,37 | 0,82 |
| Recall |  | 0,61 | 0,51 | 0,53 | 0,59 | 0,62 | 0,44 | 0,4 | 0,36 | 0,25 | 0,3 |

По результатам тестирования оценивается ошибка работы нейросетевой модели. Оценивались достоверность, точность, Recall полнота, F1 (F Мера- среднее гармонической точности и полноты), FRR (ошибки второго первого рода), FAR (ошибки первого рода).

Результаты сравнения метрик качества моделей без 1 НС и полностью вся модель после тестирования представлены в таблице 6:

Таблица 6 Метрики качества моделей

|  |  |  |
| --- | --- | --- |
| Характеристика | Значение без 1 НС | Значение Все НС |
| Accuracy | 0,956132 | 0,9673086 |
| Precision | 0,954424 | 0,9240331 |
| Recall | 0,376720 | 0,707936 |
| F1 | 0,540212 | 0,801677 |
| FRR | 0,623280 | 0,292063 |
| FAR | 0,001109 | 0,005991 |

Рисунок 33 Метрики качества моделей

По таблице 6 видно, что отклик системы, после удаления одной из сетей снизился, более чем на 30%. Так как на данном этапе происходит чистка сети и в результате нужно убрать действительно ненужные сети, в данном случае нельзя сказать, что 1 НС ни за что не отвечает. Анализ был произведен для каждой сети. Результаты опыта показали, что максимально хорошие метрики качества остаются у модели с десятью НС и проекцией на ось x и y.

Помимо отключения каждой из сетей, были проведены тесты порога, когда сеть возвращает число от 0 до 1 которое означает степень ее уверенности что это одинаковые картинки. Если сеть ответила числом больше порога – будем считать, что сеть сказала 1, если меньше – считаем, что 0. Для тестирования системы были взяты пороги 0.5; 0.4; 0.35. Также тесты проводились как на всех образцах (в том числе и подделки подписей), так и на только оригинальных подписях без наличия подделок. Заключительный тест проводился на 3 и 5 парах входных картинках.

## **3.3 Пути развития предложенного метода**

Один из следующих возможных шагов развития работы может быть оптимизацией текущих процессов. Например, для процесса аутентификации необходимо сравнить поданную подпись с тремя эталонными подписями. Чтобы не хранить все 4 картинки, можно заранее прогонять их через сеть и хранить только 128 - мерные вектора, которые получаются в результате работы алгоритма, это упростит и облегчит хранение, а также потребуется меньше вычислительных мощностей, так как каждую подпись можно будет обрабатывать только один раз и далее уже сравнивать вектора между собой.

Следующий шаг в развитии работы - создание мобильного приложения под Android, это аргументировано тем, что стоимость разработки под данную мобильную операционную систему дешевле. Средний заработок разработчика на языке программирования Objective-C порядка 210 тысяч рублей, при среднем заработке разработчика Kotlin 180 тысяч рублей по данным исследований за первое полугодие 2018 года [19]. Также iOS по некоторым исследованиям более защищенная от взлома операционная система [20].

На Рисунок 34 представлен прототип приложения. Для начала работы с ним необходимо ввести 3 подписи, именно столько достаточно для того, чтобы распознавать в дальнейшем, человека.

Такое приложение может позволить себе любая небольшая или средняя компания.

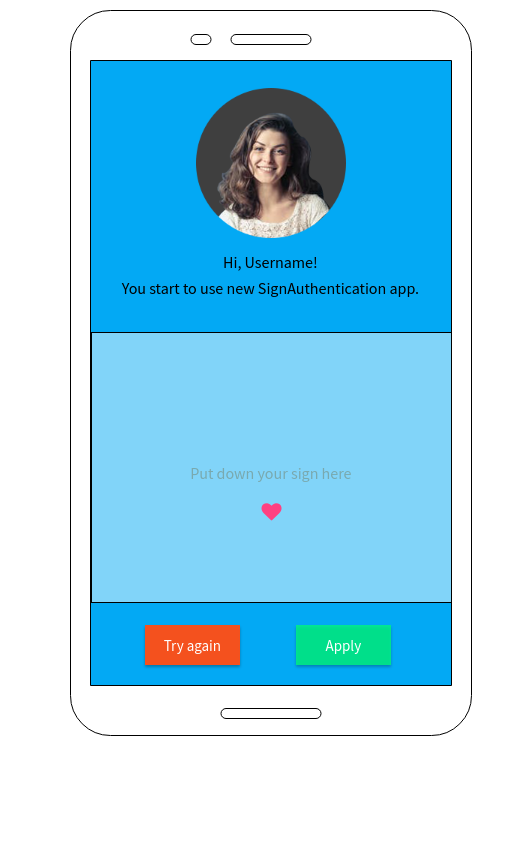


Рисунок 34 Прототип мобильного приложения для Android

Сценарий работы приложения, опираясь на текущую реализацию может быть следующим:

1. Пользователь регистрируется в приложении (оставляет три своих “эталонных” подписи);
2. Сеть сравнивает попытку аутентификации с текущими тремя сохраненными векторами (сравнение классов)

Сравнение классов происходит путем первоначальной подготовки входной картинки, методами, описанными в работе. Далее новая подпись по очереди прогоняется и сравнивается с текущими “эталонными” подписями.

3) “Голосование” 10 нейросетей и вычисление проекций их на x и на y.

Блок-схема реализации алгоритма изображена на *Рисунок 35*.

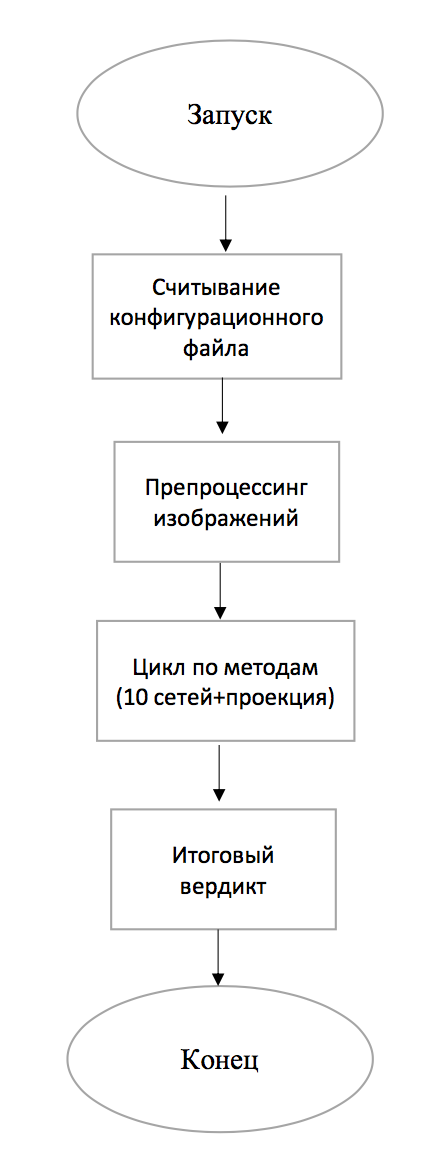


Рисунок 35 Схема реализации алгоритма

## **3.4 Выводы**

Был проведен ряд тестов, который меняет количество нейронных сетей, размер и количество фильтров в свертке. Проводились тесты поочередного отказа от каждой из сетей, для проверки, не выдают ли они одинаковые ответы с одной из других сетей. Протестирована логика оценки качества модели. Протестирован порог степени уверенности сети.

Выявлено, что лучшие результаты показывает ансамбль из десяти нейронных сетей, вместе с проекциями на оси x и y.

# 

# **ЗАКЛЮЧЕНИЕ**

Был проведен анализ предметной области и выбран метод распознавания рукописного текста с использованием ансамбля сиамских нейронных сетей.

В результате анализа обучаемости системы сиамских нейронных сетей было установлено, что для удовлетворительного результата (свыше 90%) необходимо как минимум 400 обучающих пар.

Описанная система подходит для случаев, когда число пользователей невелико, например, для "мобильного девайса", когда число пользователей ограничено (например, в iPhone, максимальное число отпечатков - 10 для touch ID или 2 для Face ID).

Метод применим для распознавания подписи человека и в перспективе возможно его использование в программном обеспечении для авторизации пользователя.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Князев А. В. Распознавание слитного рукописного текста / Вестник МЭИ, 2013. С. 15.
2. Попова В. В. Распознавание рукописного текста на китайском языке на основе ключей при помощи нейронных сетей // Информационные технологии XXI века, 2013. С. 47
3. РАСПОЗНАВАНИЕ РУКОПИСНЫХ СИМВОЛОВ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВОЙ ТЕХНОЛОГИИ, Андреев Всеволод Владимирович, Журавлев Михаил Сергеевич, // Прикладная Информатика № 2, с. 115-117
4. Исрафилов Х. С. Применение нейронных сетей в распознавании рукописного текста // Молодой ученый. - 2016. - №29. - С. 24-27 [Электронный ресурс]. – Ресурс доступа: <https://moluch.ru/archive/133/37372/> (Дата обращения: 2018-11-15)
5. Кулакова О.А., Воронова Л.И. // Московский Технический Университет Связи и Информатики. - 2017. С. 10-11.
6. Comp. nanotechnol., 2016, выпуск 3, страницы 224–241. Защита информации. Биометрические технологии идентификации в системах контроля и управления доступом. С 224-241.
7. Алгоритмы оффлайн-распознавания рукописных цифр. Козлов Владимир Дмитриевич. Выпускная квалификационная работа. Научный руководитель к. ф.–м. н, доцент С. И. Гуров. Московский государственный университет имени М. В. Ломоносова. С. 73.
8. Маркова С.В., Жигалов К.Ю. Применение нейронной сети для создания системы распознавания изображений // Фундаментальные исследования. – 2017. – № 8-1. С 60-64.
9. Geoffrey E. Hinton. Dynamic Routing Between Capsules // Google Brain, 2017. С 15-18.
10. 36 материалов о нейросетях: книги, статьи и последние исследования [Электронный ресурс]. – Ресурс доступа: [https://habrahabr.ru/company/neurodatalab/blog/336218](https://habrahabr.ru/company/neurodatalab/blog/336218/) (дата обращения: 11.02.2019)
11. Применение нейросетей в распознавании изображений [Электронный ресурс]. – Ресурс доступа: https://geektimes.ru/post/74326 (дата обращения: 19.01.2019)
12. Метрики качества [Электронный ресурс]. – Ресурс доступа: <http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html>(дата обращения: 21.01.2019)
13. Калькулятор рецептивных полей [Электронный ресурс]. – Ресурс доступа: https://fomoro.com/projects/project/receptive-field-calculator (дата обращения: 11.12.2018)
14. Pooling // объединение слоев [Электронный ресурс]. – Ресурс доступа: http://datareview.info/article/eto-nuzhno-znat-klyuchevyie-rekomendatsii-po-glubokomu-obucheniyu-chast-2 (дата обращения: 21.01.2019)
15. Набор данных для обучения и тестирования [Электронный ресурс]. – Ресурс доступа: http://www.iapr-tc11.org/mediawiki/index.php/ICDAR\_2011\_Signature\_Verification\_Competition\_(SigComp2011) (дата обращения: 15.02.2019)
16. Сверточная нейронная сеть на PyTorch[Электронный ресурс]. – Ресурс доступа: [https://neurohive.io/ru/tutorial/cnn-na-pytorch](https://neurohive.io/ru/tutorial/cnn-na-pytorch/) (дата обращения: 10.03.2019)
17. Как работает FaceID в iPhone X: алгоритм на языке Python[Электронный ресурс]. – Ресурс доступа: <http://ai-news.ru/2018/03/kak_rabotaet_faceid_v_iphone_x_algoritm_na_yazyke_python.html> (дата обращения: 14.04.2019)
18. Face ID Security Guide. – Ресурсдоступа: <https://www.apple.com/business/site/docs/FaceID_Security_Guide.pdf>
19. Зарплаты ИТ-специалистов на середину 2018 года [Электронный ресурс]. – Ресурс доступа: <https://vc.ru/flood/43849-zarplaty-it-specialistov-na-seredinu-2018-goda> (дата обращения: 01.05.2019)
20. EnterpriseAndroidvs IOS [Электронный ресурс]. – Ресурс доступа: <https://medium.com/predict/enterprise-android-vs-ios-which-is-more-secure-51a95cac53c2> (дата обращения: 11.03.2019)
21. Сверточная нейронная сеть, часть 1: структура, топология, функции активации и обучающее множество [Электронный ресурс]. – Ресурс доступа: <https://habr.com/ru/post/348000/> (дата обращения: 23.02.2019)
22. Градиентый бустинг [Электронный ресурс]. – Ресурс доступа: <https://neurohive.io/ru/osnovy-data-science/gradientyj-busting/> (дата обращения: 26.04.2019)
23. Разновидности нейронных сетей [Электронный ресурс]. – Ресурс доступа: <https://tproger.ru/translations/neural-network-zoo-1/> (дата обращения: 05.05.2019)
24. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес [Электронный ресурс]. – Ресурс доступа: <https://habr.com/ru/company/ods/blog/324402/#1-begging> (дата обращения: 03.05.2019)
25. Логистическая функция ошибки [Электронный ресурс] <https://dyakonov.org/2018/03/12/логистическая-функция-ошибки/> (дата обращения: 03.05.2019);
26. Майков К.А. Козлова Ю.А. Метод идентификации пользователя по рукописному кодовому слову // Новые информационные технологии в автоматизированных системах, М.- 2019, с.19-21;

# **ПРИЛОЖЕНИЕ A**