

# **АННОТАЦИЯ**

В настоящей практике была создана модель для классификации изображений с использованием сверточной нейронной сети (CNN), а также проведено сравнение с более простой полносвязной нейронной сетью. В качестве изображений для классификаций были взяты изображения рукописных цифр из базы данных MNIST.

Данная работа посвящена теме «Классификация изображений с использованием сверточной сети (CNN)». Работа состоит из TODO

Во Введении обоснована актуальность выбранной темы, определены объект и предмет исследования, представлены цель и задачи на разработку.

В Исследовательском разделе проведен анализ предметной области и сравнение используемых нейронных сетей, после чего определены требования для моделей, которые будут обучены в этой работе.

В Аналитическом разделе был проведен выбор инструментов, с помощью которых будет проведено обучение, а также разработаны архитектуры используемых сетей.

В Технологическом разделе было проведено само обучение моделей с заданными архитектурами сетей, после чего проведено тестирование и рассмотрены результаты.

В Заключении подведены итоги данной работы.

# **СОДЕРЖАНИЕ**

АННОТАЦИЯ.....	1
ВВЕДЕНИЕ.....	3
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ.....	5
1.1 Анализ предметной области.....	5
1.2 Сравнение архитектур сетей.....	6
1.3 Требование к моделям.....	6
Вывод по разделу 1.....	7
2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ.....	8
2.1 Выбор инструментов и методов разработки.....	8
2.2 Архитектура сетей.....	9
Вывод по разделу 2.....	10
3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ.....	11
3.1 Описание процесса обучения моделей.....	11
3.2 Описание процесса тестирования.....	11
Вывод по разделу 3.....	17
ЗАКЛЮЧЕНИЕ.....	18

# **ВВЕДЕНИЕ**

Темой настоящей ознакомительной практики является «Классификация изображений с использованием сверточной сети (CNN)». Задача классификации изображений является всё более чаше встречаемой за последний год в связи с большим развитием машинного обучения и возможностью решать задачи не только алгоритмически, когда разработчик вручную пишет четкие инструкции, которые решают поставленную задачу, но и с использованием машинного обучения, когда через предоставление данных модели и автоматического поиска закономерностей, можно строить прогнозы. Главным преимуществом нового подхода стала возможность решать такие задачи, для которых сложно или даже невозможно написать алгоритм решения. К таким задачам относится и задача классификации изображений.

Однако для разных задач машинного обучения существуют разные модели, требующих разное количество ресурсов и решающих поставленную задачу с разной точностью. Для работы с изображениями лучше всего подходит сверточные нейронные сети, однако следует понять на сколько лучше они справляются с задачей чем архитектурно более простые сети.

Цель работы заключается в создании сверточной нейронной сети и сравнение её с более простой нейронной сетью, как в архитектуре, так и на результатах тестирования.

Для достижения поставленной цели требуется решить следующие основные задачи:

- провести анализ предметной области;
- провести сравнение сетей в архитектуре;
- определить требования к моделям;
- провести обучение моделей;
- провести тестирование моделей;

- провести сравнение моделей на результатах тестирования;
- сделать выводы по проделанной работе.

В данной работе объектом исследования будет представлена сверточная нейронная сеть и простая нейронная сеть, способные классифицировать изображения цифр и соответствующим поставленным требованиям.

Предметом исследования является машинное обучение, к которому относятся представленные модели, способные решать поставленную задачу.

# **1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**

## **1.1 Анализ предметной области**

Прежде всего стоит разобраться в необходимости использования машинного обучения для решения задачи классификации изображений и невозможности решить это алгоритмически.

Изображение является матрицей, где каждая ячейка является пикселием, содержащим информацию о цвете. Даже простой объект как цифра может быть представлен на изображении в разных местах, иметь разные масштабы, ориентацию или частично не нарисован. Алгоритмически учесть все эти факты практически невозможно, особенно если задача может видоизменяться. Например в задаче распознавания цифр может измениться размер матрицы, цвет фона или цифр, добавиться тени или шум.

Именно машинное обучение позволяет найти закономерности в автоматическом режиме, однако для этого требуется большой набор данных, на котором будет происходить обучение.

Однако в машинном обучении существует множество разных моделей, которые могут решать одну и ту же задачу с разной точностью и с разными требованиями к ресурсами. Современные глубокие нейронные сети, насчитывающие десятки и сотни слоев, способны решать достаточно сложные задачи, однако требуют не только большим вычислительных мощностей, но и большого объема данных. В связи с ограниченным объемом вычислительных ресурсов и достаточно малым набором данных для глубоких сетей (60000 изображений для обучения и 10000 изображений для тестирования), то архитектура созданных нейронных сетей будет небольшой для возможности проведения обучения, но достаточно сложной для возможности классифицировать изображения цифр с достаточной сложностью.

## **1.2 Сравнение архитектур сетей**

Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение по принципу биологических нейронных сетей — сетей нервных клеток живого организма.

Сверточная нейронная сеть — специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов.

Сверточная нейронная сеть использует некоторые особенности зрительной коры, в которой существуют простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определенного набора простых клеток. Таким образом, идея сверточной нейронных сетей заключается в чередовании сверточных слоев и субдискретизирующих слоев (слоев подвыборки). Структура сети — односторонняя, многослойная.

## **1.3 Требование к моделям**

В результате исследования данных нейронных сетей было определено, что для задачи классификации изображений цифр существует множество результатов, благодаря которым можно оценить требуемую точность для таких сетей.

Для 2-уровневой полносвязной нейронной сети удалось достичь ошибки 1.6%, а для простой сверточной нейронной сети — 0.31%.

Так как это наилучшие результаты, а в данной работе потребуется лишь сравнить модели, то попытаемся обучим полносвязную нейронную до ошибки 2-3%, чтобы убедиться в её возможности решать исходную задачу, а после обучим сверточную нейронную сеть, которая должна иметь ошибку меньшую чем у полносвязной нейронной сети.

Учитывая существующие результаты для сверточной нейронной сети, то попытаемся обучить эту сеть до ошибки 0.5-1%.

## **Вывод по разделу 1**

Был проведен анализ предметной области, сбор и анализ представленных моделей, способных решить исходную задачу, а также определены основные требования к моделям.

Анализ предметной области показал, что машинное обучение позволяет решить некоторые алгоритмически сложные задачи, а также помог определить подходящие по возможностям для данной работы модели, способных решить задачу классификации изображений.

На основе поставленной ранее задачи были предъявлены требования по точности и сложности для обучаемых нейронных сетей.

## **2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ**

### **2.1 Выбор инструментов и методов разработки**

Проведить обучения моделей можно разными способами: использование облачных сервисов, предоставляющих вычислительные мощности для обучения достаточно сложных сетей; использование своего оборудования, таких как компьютер или ноутбук, обладающих хорошими вычислительными мощностями для ежедневных задач, но зачастую недостаточных для обучение даже простых глубоких сетей; использование готовой базовой модели и проведение её дообучения под конкретную задачу.

Так как в данной работе используются небольшие нейронные сети, чтобы провести простое сравнение сверточной нейронной сети с обычной, а также задача классификации изображений цифр является достаточно простой, чтобы её могли решить даже относительно простые модели, то нет необходимости в выделении больших вычислительных мощностей. К тому же облачные сервисы являются платными и требуют хорошей подготовки для их использования, что также делает их непригодными для использования в данной задачи.

Дообучение базовых моделей не годится для данной работы, потому что все они достаточно сложные для такой простой задачи как классификация изображений цифр. К тому же такие модели могут использовать методы, способных улучшить результаты работы, но затрудняющих использование модели для сравнения с простыми аналогами.

Использование своего же оборудования позволяет полностью контролировать процесс обучения, а также обладает достаточными вычислительными ресурсами для обучения данных моделей.

После анализа преимуществ и недостатков каждого из подхода было решено использовать свое оборудование.

Обучение данных моделей будет проводиться на языке программирования Python, с использованием библиотек, позволяющих облегчить обучение сетей, проведение тестирования и анализ результатов. В качестве среды разработки был выбран Jupyter lab, который обладает достаточным функционалом для реализации поставленной задачи.

Для обучения и анализа моделей на Python будут использоваться такие библиотеки, как:

- torch — работа с моделями;
- numpy — работа с вычислениями;
- matplotlib — построение графиков.

## 2.2 Архитектура сетей

Исходя из требований к моделям стоит разобраться с архитектурой простой нейронной сети и сверточной нейронной сети и их методами обучения.

Нейронная сеть:

Входной слой размером 784 — количество пикселей в изображении.

1 скрытый слой размером 800 — количество нейронов в лучшей найденной реализации.

Выходной слой размером 10 — количество цифр.

Функция потерь — CrossEntropyLoss (для классификации).

Оптимизатор — Adam.

Функция активации — ReLU.

Сверточная нейронная сеть:

2 сверточных слоя - размеры ядер 3, число выходных каналов 32 и 64 соответственно.

2 пулинга — MaxPool2d с параметром 2.

1 скрытый слой размером 3136 — число каналов умноженное на размер матрицы после понижения размерности ( $64 * 7 * 7$ ).

Выходной слой размером 10 — количество цифр.

Оптимизатор — Adam.

Функция активации — ReLU.

Особенности: Dropout ( $p = 0.3$ ).

## **Вывод по разделу 2**

По результатам из возможных инструментов для обучения были выбраны наиболее подходящие: было выбрано оборудование, способ обучения, язык программирования, библиотеки и среда разработки. Обе модели должны быть обучены на языке Python с использованием torch, основной библиотекой для обучения.

С учетом требований и представленных ресурсов были спроектированы архитектур для простой нейронной сети и для сверточной нейронной сети.

# **3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ**

## **3.1 Описание процесса обучения моделей**

Прежде чем начинать обучать любую модель, надо подготовить входные данные. Цифры в датасете MNIST представлены матрицей 28x28 с целыми числами от 0 до 255, где чем больше число, тем выше освещенность пикселя. Для обеих моделей каждую такую матрицу надо нормализовать, чтобы значения были вещественными от 0 до 1. Для обычной нейронной сети каждую матрицу надо представить также в виде вектора размером  $28 \times 28 = 784$ .

Для ускорения обучения использовались батчи с размерами 64.

Наилучшие результаты обучения для обычной нейронной сети происходили при learning rate равном 0.01 на первых 20 эпохах, после чего это значение понижалось до 0.001 и модель дообучался еще на 20 эпохах. Дальнейшее обучение либо не приводило к росту точности, либо даже ухудшало результат.

Наилучшие результаты обучения для сверточной нейронной сети происходили при learning rate равном 0.001 с последующим его понижением в 10 раз каждые 3 эпохи.

## **3.2 Описание процесса тестирования**

Процесс тестирования модели является важным этапом, ведь именно он показывает результаты модели, что помогает отразить её способность обобщать и работать с данными, с которыми она не встречалась во время обучения. Именно для этого было отложено 1000 экземпляров цифр, на которых проводилось тестирование обеих нейронных сетей.

Датасет для тестирования также был подготовлен отдельно для каждой модели, как это описано выше.

При тестировании обнаружилось, что сверточная нейронная сеть достигает высоких результатов уже после первой эпохи обучения (больше 98%), тогда как обычная нейронная сеть достигает этих результатов через большее количество эпох обучения (около 10-20). Однако стоит заметить, что каждая эпоха обучения для сверточной нейронной сети занимает около 15 секунд, тогда как за то же время обычная нейронная сеть проходит 3-4 эпохи.

Итоговые результаты тестирования сетей представлены в таблице ниже.

Модель	Точность, %	Время обучения, сек
CNN	99.27	144
NN	98.60	129

Где CNN — сверточная нейронная сеть, NN — обычная нейронная сеть

Из представленных результатов видно, что сверточная нейронная сеть справляется с изображениями намного лучше, уменьшив ошибку почти в 2 раза.

Обученные модели были сохранены, для возможного дальнейшего использования. Обе модели способны к предсказанию цифры, изображенной в матрице 28x28 при небольшой подготовке данных.

Результаты на тестовой выборке для сверточной нейронной сети изображены на Рисунках 1 — 3.

```
Test set: Average loss: 0.0000, Accuracy: 9844/10000 (98.44%)
Time: 15 seconds
Test set: Average loss: 0.0000, Accuracy: 9866/10000 (98.66%)
Time: 30 seconds
Test set: Average loss: 0.0000, Accuracy: 9898/10000 (98.98%)
Time: 45 seconds
Test set: Average loss: 0.0000, Accuracy: 9918/10000 (99.18%)
Time: 60 seconds
Test set: Average loss: 0.0000, Accuracy: 9922/10000 (99.22%)
Time: 77 seconds
Test set: Average loss: 0.0000, Accuracy: 9923/10000 (99.23%)
Time: 94 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 110 seconds
Test set: Average loss: 0.0000, Accuracy: 9925/10000 (99.25%)
Time: 127 seconds
Test set: Average loss: 0.0000, Accuracy: 9927/10000 (99.27%)
Time: 144 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 160 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 177 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 194 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 210 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 227 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 244 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 261 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 277 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 294 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 311 seconds
Test set: Average loss: 0.0000, Accuracy: 9926/10000 (99.26%)
Time: 328 seconds
```

Рисунок 1 — первая часть вывода тестирования (CNN)

Результаты на тестовом датасете для нейронной сети:

```
Test set: Average loss: 0.0001, Accuracy: 9669/10000 (96.69%)
Time: 4 seconds
Test set: Average loss: 0.0001, Accuracy: 9701/10000 (97.01%)
Time: 7 seconds
Test set: Average loss: 0.0001, Accuracy: 9783/10000 (97.83%)
Time: 11 seconds
Test set: Average loss: 0.0001, Accuracy: 9799/10000 (97.99%)
Time: 15 seconds
Test set: Average loss: 0.0001, Accuracy: 9798/10000 (97.98%)
Time: 19 seconds
Test set: Average loss: 0.0001, Accuracy: 9799/10000 (97.99%)
Time: 23 seconds
Test set: Average loss: 0.0001, Accuracy: 9823/10000 (98.23%)
Time: 27 seconds
Test set: Average loss: 0.0001, Accuracy: 9821/10000 (98.21%)
Time: 31 seconds
Test set: Average loss: 0.0001, Accuracy: 9807/10000 (98.07%)
Time: 35 seconds
Test set: Average loss: 0.0001, Accuracy: 9813/10000 (98.13%)
Time: 39 seconds
Test set: Average loss: 0.0001, Accuracy: 9815/10000 (98.15%)
Time: 43 seconds
Test set: Average loss: 0.0001, Accuracy: 9835/10000 (98.35%)
Time: 47 seconds
Test set: Average loss: 0.0001, Accuracy: 9829/10000 (98.29%)
Time: 51 seconds
Test set: Average loss: 0.0001, Accuracy: 9831/10000 (98.31%)
Time: 55 seconds
Test set: Average loss: 0.0001, Accuracy: 9781/10000 (97.81%)
Time: 58 seconds
Test set: Average loss: 0.0001, Accuracy: 9822/10000 (98.22%)
Time: 62 seconds
Test set: Average loss: 0.0001, Accuracy: 9832/10000 (98.32%)
Time: 67 seconds
Test set: Average loss: 0.0001, Accuracy: 9836/10000 (98.36%)
Time: 71 seconds
Test set: Average loss: 0.0001, Accuracy: 9850/10000 (98.50%)
Time: 75 seconds
Test set: Average loss: 0.0001, Accuracy: 9826/10000 (98.26%)
Time: 79 seconds
Test set: Average loss: 0.0001, Accuracy: 9842/10000 (98.42%)
Time: 82 seconds
Test set: Average loss: 0.0001, Accuracy: 9848/10000 (98.48%)
Time: 87 seconds
Test set: Average loss: 0.0001, Accuracy: 9850/10000 (98.50%)
Time: 91 seconds
```

Рисунок 2 — вторая часть вывода тестирования (NN)

```
Time: 94 seconds
Test set: Average loss: 0.0001, Accuracy: 9851/10000 (98.51%)
Time: 98 seconds
Test set: Average loss: 0.0001, Accuracy: 9852/10000 (98.52%)
Time: 103 seconds
Test set: Average loss: 0.0001, Accuracy: 9851/10000 (98.51%)
Time: 107 seconds
Test set: Average loss: 0.0001, Accuracy: 9856/10000 (98.56%)
Time: 112 seconds
Test set: Average loss: 0.0001, Accuracy: 9857/10000 (98.57%)
Time: 116 seconds
Test set: Average loss: 0.0001, Accuracy: 9857/10000 (98.57%)
Time: 120 seconds
Test set: Average loss: 0.0001, Accuracy: 9855/10000 (98.55%)
Time: 125 seconds
Test set: Average loss: 0.0001, Accuracy: 9860/10000 (98.60%)
Time: 129 seconds
Test set: Average loss: 0.0001, Accuracy: 9855/10000 (98.55%)
Time: 133 seconds
Test set: Average loss: 0.0001, Accuracy: 9856/10000 (98.56%)
Time: 137 seconds
Test set: Average loss: 0.0001, Accuracy: 9857/10000 (98.57%)
Time: 141 seconds
Test set: Average loss: 0.0001, Accuracy: 9851/10000 (98.51%)
Time: 145 seconds
Test set: Average loss: 0.0001, Accuracy: 9852/10000 (98.52%)
Time: 149 seconds
Test set: Average loss: 0.0001, Accuracy: 9853/10000 (98.53%)
Time: 153 seconds
Test set: Average loss: 0.0001, Accuracy: 9856/10000 (98.56%)
Time: 157 seconds
Test set: Average loss: 0.0001, Accuracy: 9856/10000 (98.56%)
Time: 161 seconds
Test set: Average loss: 0.0001, Accuracy: 9855/10000 (98.55%)
Time: 165 seconds
Test set: Average loss: 0.0001, Accuracy: 9856/10000 (98.56%)
Time: 169 seconds
Test set: Average loss: 0.0001, Accuracy: 9855/10000 (98.55%)
Time: 174 seconds
Test set: Average loss: 0.0001, Accuracy: 9857/10000 (98.57%)
Time: 178 seconds
Test set: Average loss: 0.0001, Accuracy: 9857/10000 (98.57%)
Time: 182 seconds
Test set: Average loss: 0.0001, Accuracy: 9858/10000 (98.58%)
Time: 187 seconds
```

Рисунок 3 — третья часть вывода тестирования (NN)

## **Вывод по разделу 3**

В технологическом разделе был описан процесс обучения сверточной и обычной нейронных сетей. Также был описан процесс тестирования моделей и проанализированы его результаты.

Из полученных результатов можно сделать вывод, что сверточная нейронная сеть справляется лучше с работой с изображениями, что дает возможность получить лучшие результаты предсказания по сравнению с обычными нейронными сетями.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения настоящей работы были выполнены следующие задачи:

- проведен анализ предметной области;
- проведено сравнение сетей в архитектуре;
- определены требования к моделям;
- проведено обучение моделей;
- проведено тестирование моделей;
- проведено сравнение моделей на результатах тестирования;
- сделаны выводы по проделанной работе.

По окончанию обучения были получены модели, способные использоваться на реальных задачах и предсказывать изображенную цифру с достаточно большой точностью.

Цели и задачи, поставленные для выполнения работы, были выполнены.