

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики — процессов управления

# Отчет

По индивидуальному проекту по теме  
«Сравнение эффективности распознавания изображений  
рукописных цифр полносвязными и сверточными  
нейросетями, обученными на датасете MNIST»

Выполнил Илямаков А.Ю.,  
студент 3 курса, группа 21.Б02

Санкт-Петербург  
2024

## 1. Описание и постановка задачи:

Цель данного проекта заключается в создании моделей нейронной сети: полносвязной и сверточной для распознавания рукописных цифр, а также сравнение эффективности их распознавания.

Для решения данной задачи использовался датасет MNIST (Modified National Institute of Standards and Technology) — это широко используемый набор данных в области машинного обучения, который содержит изображения рукописных цифр от 0 до 9.

Каждое изображение в наборе данных MNIST имеет размер 28x28 пикселей, представлено в оттенках серого (значения пикселей от 0 до 255), и каждое изображение ассоциировано с меткой (цифрой от 0 до 9), которую оно представляет.

Данные разделены на тренировочное и тестовое множества, в тренировочном множестве 60 тысяч изображений, в тестовом — 10 тысяч. В каждом классе около 6000 изображений в тренировочном множестве и 1000 в тестовом, данные распределены равномерно. На валидационные данные приходится 20% от тренировочных.

Поддержка датасета MNIST есть во многих крупных фреймворках для работы с нейронными сетями. Я же использовал его из библиотеки языка Python Keras.



*пример изображения*

Предобработка данных заключалась в масштабировании значения пикселей изображений в диапазон от 0 до 1 и преобразования меток классов (целочисленных значений) в формат one-hot encoding.

## 2. Описание используемой нейронной сети.

### 1. Полносвязная нейронная сеть.

Модель представляет собой полносвязную нейронную сеть со слоями:

- Flatten(input\_shape=(28, 28, 1)): Этот слой Flatten преобразует входные данные в одномерный массив. Входной input\_shape=(28, 28, 1) означает, что входные данные представляют собой изображения размером 28x28 пикселей в оттенках серого (1 канал). После применения слоя Flatten, каждое изображение размером 28x28 пикселей будет преобразовано в одномерный массив длиной 784 (28\*28).
- Dense(512, activation='relu'): Это полносвязный слой с 512 нейронами и функцией активации ReLU ('relu'). После слоя Flatten мы имеем одномерный массив длиной 784, который подается на этот слой. Каждый нейрон в этом слое соединен с каждым элементом входного массива, и активация ReLU применяется к выходу каждого нейрона.
- Dense(256, activation='relu').
- Dense(256, activation='relu').
- Dense(10, activation='softmax'): Это последний полносвязный слой с 10 нейронами, которые соответствуют 10 классам цифр от 0 до 9. Функция активации softmax используется здесь для преобразования выхода нейронов в вероятности принадлежности к каждому из 10 классов. Каждый нейрон в этом слое представляет вероятность того, что входное изображение принадлежит определенному классу цифры.

Размер пакета данных, передаваемых на каждой итерации обучения равен 16 (batch\_size=16). Используется оптимизатор Adam, который обновляет веса модели на основе градиентов функции потерь, в качестве функции потерь - категориальная кросс-энтропия, определяет разницу между предсказанными и истинными метками классов.

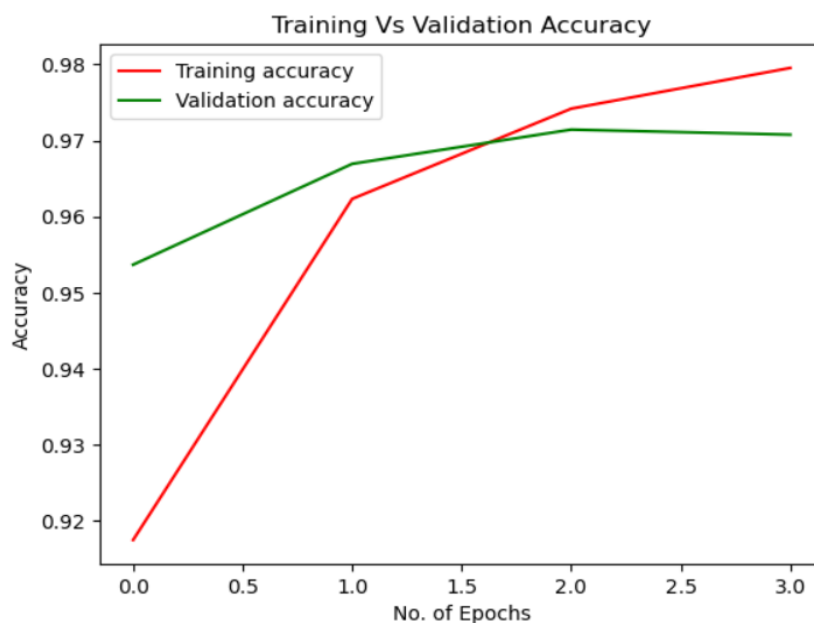
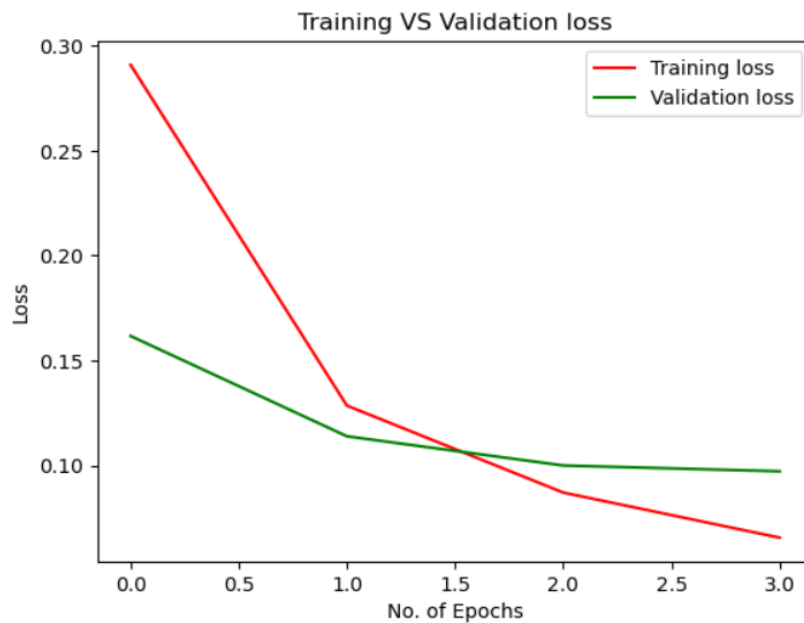
В качестве метрики оценки качества модели выбрана **точность ('accuracy')**, которая измеряет долю правильно предсказанных классов.

Модель была обучена в течение 4 эпох. Каждая эпоха состояла из 3000 шагов с временем обучения 20 секунд на эпоху. Первая эпоха показала точность на уровне 88.7% на обучающем наборе и функцию потерь (0.36). На валидационном наборе точность была выше (96%). Следующие эпохи показали улучшение производительности модели. Точность на обучающем наборе увеличилась до 98% к концу 4-ой эпохи, а функция потерь снизилась до 0.062.

---

Epoch 1/4	3000/3000	21s	7ms/step	- accuracy: 0.8877	- loss: 0.3597	- val_accuracy: 0.9597	- val_loss: 0.1344
Epoch 2/4	3000/3000	19s	6ms/step	- accuracy: 0.9674	- loss: 0.1085	- val_accuracy: 0.9674	- val_loss: 0.1163
Epoch 3/4	3000/3000	20s	7ms/step	- accuracy: 0.9770	- loss: 0.0766	- val_accuracy: 0.9703	- val_loss: 0.1117
Epoch 4/4	3000/3000	20s	7ms/step	- accuracy: 0.9807	- loss: 0.0616	- val_accuracy: 0.9751	- val_loss: 0.0920

*информация о каждой эпохе обучения*



*графики метрики качества и функции ошибки*

## 2. Сверточная нейронная сеть.

Модель представляет собой сверточную нейронную сеть (CNN) с несколькими сверточными слоями, слоями пулинга и полносвязными слоями для классификации изображений:

- Слои свертки ('Conv2D'):
  1. 32 фильтра размером (3,3), функция активации ReLU, входной размер (28, 28, 1)
  2. 64 фильтра размером (3,3), функция активации ReLU.

- Пулингговые слои ('MaxPooling2D'):
  1. размер пула (2,2) с шагом (strides) 2.
  2. размер пула (2,2) с шагом (strides) 2.
- Flatten(): слой, преобразующий выходы последнего сверточного слоя в одномерный вектор перед подачей на полносвязные слои
- Dense(128, activation='relu'): Это полносвязный слой с 128 нейронами и функцией активации ReLU ('relu'). После слоя Flatten мы имеем одномерный массив длиной 784, который подается на этот слой. Каждый нейрон в этом слое соединен с каждым элементом входного массива, и активация ReLU применяется к выходу каждого нейрона.
- Dense(10, activation='softmax'): Это последний полносвязный слой с 10 нейронами, которые соответствуют 10 классам цифр от 0 до 9. Функция активации softmax используется здесь для преобразования выхода нейронов в вероятности принадлежности к каждому из 10 классов. Каждый нейрон в этом слое представляет вероятность того, что входное изображение принадлежит определенному классу цифры.

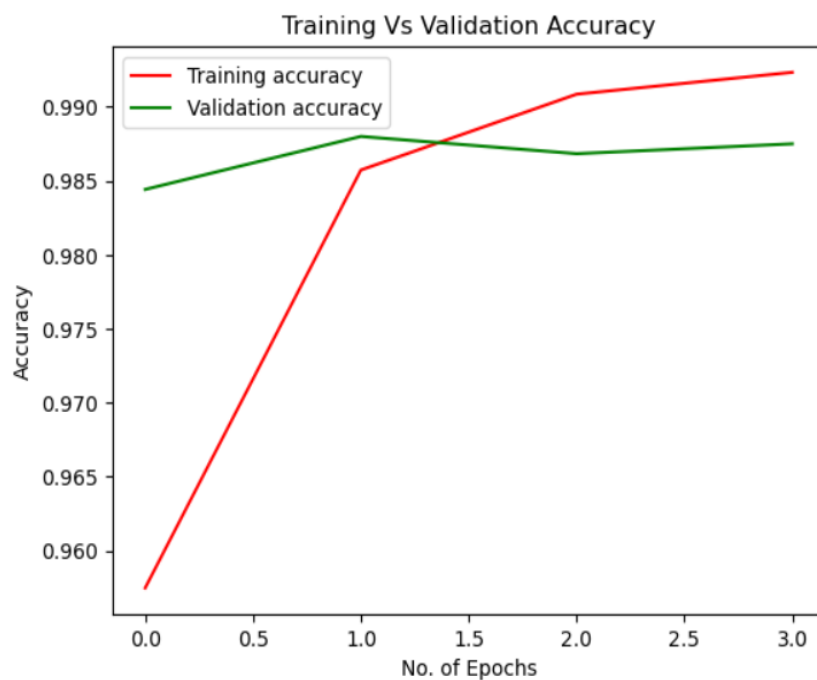
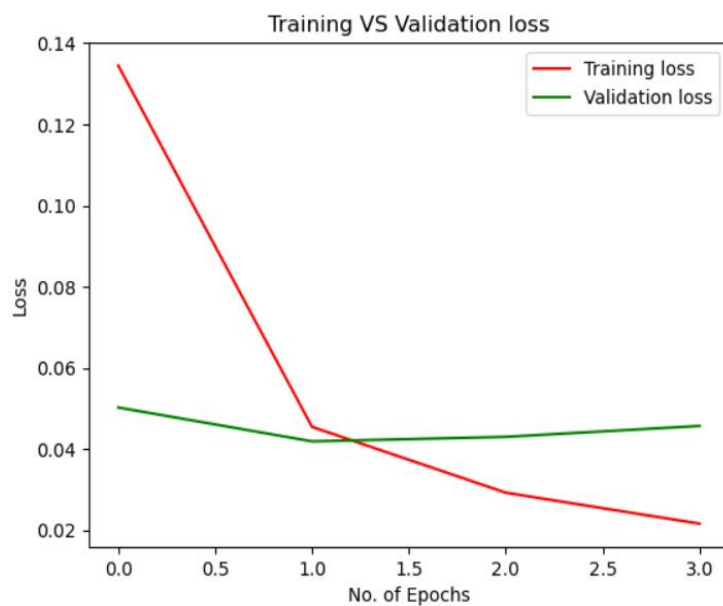
Размер пакета данных, передаваемых на каждой итерации обучения равен 32 (batch\_size=32). Используется оптимизатор Adam, который обновляет веса модели на основе градиентов функции потерь, в качестве функции потерь - категориальная кросс-энтропия, определяет разницу между предсказанными и истинными метками классов.

В качестве метрики оценки качества модели выбрана **точность ('accuracy')**, которая измеряет долю правильно предсказанных классов.

Модель была обучена в течение 4 эпох. Каждая эпоха состояла из 1500 шагов с временем обучения 25-40 секунд на эпоху. Первая эпоха показала точность на уровне 90% на обучающем наборе и функцию потерь (0.3). На валидационном наборе точность была выше (98.4%). Следующие эпохи показали улучшение производительности модели. Точность на обучающем наборе увеличилась до 99.2% к концу 4-ой эпохи, а функция потерь снизилась до 0.002.

Epoch 1/4	1500/1500	25s	16ms/step	- accuracy: 0.9034	- loss: 0.2993	- val_accuracy: 0.9844	- val_loss: 0.0502
Epoch 2/4	1500/1500	23s	15ms/step	- accuracy: 0.9861	- loss: 0.0439	- val_accuracy: 0.9880	- val_loss: 0.0419
Epoch 3/4	1500/1500	23s	15ms/step	- accuracy: 0.9913	- loss: 0.0278	- val_accuracy: 0.9868	- val_loss: 0.0430
Epoch 4/4	1500/1500	41s	16ms/step	- accuracy: 0.9927	- loss: 0.0209	- val_accuracy: 0.9875	- val_loss: 0.0457

*информация о каждой эпохе обучения*



*графики метрики качества и функции ошибки*

### **3. Краткое описание программы с указанием языка программирования и используемых библиотек.**

Программа написана на языке Python в программе Jupyter Notebook. Состоит из основных частей:

1. Подключение библиотек
2. Импортирование данных
3. Обработка данных
4. Создание модели
5. Обучение модели
6. Тестирование модели

Использованы библиотеки:

1. Time для определения времени обучения.
2. TensorFlow для создания, обучения и развертывания нейронных сетей.
3. Matplotlib для визуализации данных: построения графиков функций потерь, метрик и других данных в процессе обучения нейронных сетей.
4. NumPy для вычислений, предобработки данных, поддержки многомерных массивов (тензоров)
5. Keras для построения и обучения нейронных сетей, работающий поверх TensorFlow. Позволяет определять модели как последовательные структуры слоев

#### 4. Ссылка на репозиторий.

<https://github.com/boulevardll/neural-networks>

#### 5. Вывод и сравнение нейронных сетей.

	Model name	Время обучения (s)	Accuracy (%)	Loss	Кол-во параметров(вес модели)
0	Полносвязная	80	97.600000	0.080000	669.962 (2.56 MB)
1	Сверточная	80	99.000000	0.030000	421.642 (1.61 MB)

В ходе данного исследования было проведено сравнение эффективности двух типов нейронных сетей, сверточной и полносвязной, в задаче распознавания рукописных цифр на наборе данных MNIST. Сверточная нейронная сеть продемонстрировала высокую точность распознавания рукописных цифр по сравнению с полносвязной сетью. Это объясняется способностью сверточных слоев выделять локальные пространственные особенности изображений, что позволяет эффективно выявлять признаки на различных уровнях абстракции. В целом, на основании проведенного сравнительного анализа, сверточная нейронная сеть является более предпочтительным выбором для задачи распознавания рукописных цифр на наборе данных MNIST. Она демонстрирует высокую точность, эффективность и способность к обобщению, что делает ее идеальным инструментом для решения подобных задач.