МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

Направление подготовки: «Прикладная математика и информатика»

Магистерская программа: «Вычислительные методы и суперкомпьютерные технологии»

Образовательный курс «Глубокое обучение»

**ОТЧЕТ**

по лабораторной работе №3

**Разработка сверточных нейронных сетей**

**Выполнил:**

студент группы 381703-3м

Жариков Алексей

Нижний Новгород

2018

Содержание

[Цели 3](#_Toc530326965)

[Задачи 4](#_Toc530326966)

[Решаемая задача 5](#_Toc530326967)

[Выбор библиотеки 6](#_Toc530326968)

[Метрика качества решения задачи 6](#_Toc530326969)

[Тренировочные и тестовые наборы данных 6](#_Toc530326970)

[Конфигурации нейронных сетей 7](#_Toc530326971)

[Описание содержимого директории src 8](#_Toc530326972)

[Результаты экспериментов 8](#_Toc530326973)

[Анализ результатов 9](#_Toc530326974)

[Выводы 10](#_Toc530326975)

[Литература 11](#_Toc530326976)

# Цели

1. ***Цель*** настоящей работы состоит в том, чтобы построить архитектуру сверточной нейронной сети, которая позволяет решать практическую задачу с высокими показателями качества.
2. Задачи
3. Выполнение практической работы предполагает решение ***следующих задач***:
4. 1. Разработка нескольких архитектур сверточных нейронных сетей (варьируются количество слоев и виды функций активации на каждом слое) в формате, который принимается выбранной библиотекой глубокого обучения.
5. 2. Обучение разработанных глубоких моделей.
6. 3. Тестирование обученных глубоких моделей.
7. 4. Публикация разработанных программ/скриптов в репозитории на GitHub.
8. 5. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

# Решаемая задача

В данной работе рассматривается набор данных Fashion MNIST, который содержит 70 000 изображений в оттенках серого в 10 категориях. На изображениях показаны отдельные предметы одежды с низким разрешением (28 на 28 пикселей):

Получить можно данный набор по ссылке

<https://www.kaggle.com/zalando-research/fashionmnist> ,

но в ходе работы использовался функционал используемой библиотеки глубокого обучения, непосредственно из нее, просто импортировав и загрузив данные

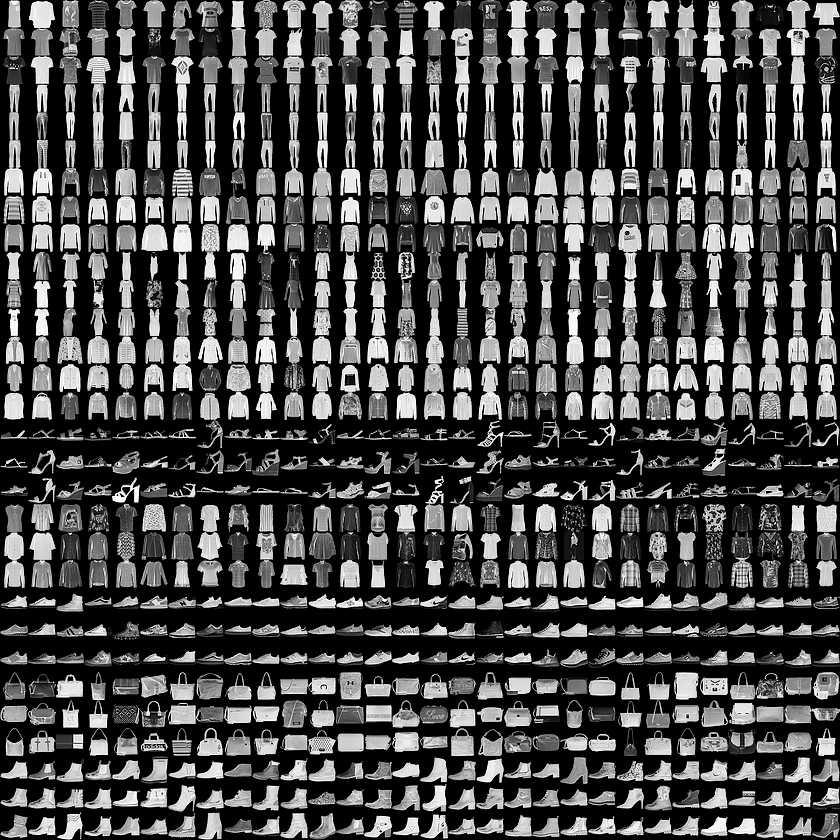


Рисунок 1 Примеры набора данных

# Выбор библиотеки

Для выполнения лабораторных работ выбрана библиотека TensorFlow для языка программирования Python c надстройкой Keras .

На этапе проверки корректности установки библиотеки выполнена разработка и запуск тестового примера сети для решения задачи классификации рукописных цифр набора данных MNIST. Достигнута точность 0.9225.

# Метрика качества решения задачи

В качестве метрики точности решения используется отношение угаданных животных ко всем в тестовой выборке:

# Тренировочные и тестовые наборы данных

# Обучающее множество 60.000 примеров и тестовый набор 10000 примеров. Всего имеется 10 классов различной одежды и обуви.

# Каждый пример обучения и теста назначается одному из следующих классов:

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Описание** | **Кол.**  **трен** | **Кол.**  **тест** |
| 0 | Футболки / сверху | 6000 | 1000 |
| 1 | Брюки | 6000 | 1000 |
| 2 | Пуловер | 6000 | 1000 |
| 3 | Платье | 6000 | 1000 |
| 4 | Пальто | 6000 | 1000 |
| 5 | Сандали | 6000 | 1000 |
| 6 | Рубашка | 6000 | 1000 |
| 7 | Кроссовки | 6000 | 1000 |
| 8 | Мешок | 6000 | 1000 |
| 9 | Ботильоны | 6000 | 1000 |

Таблица 1 Классы и количество примеров в наборе данных

# Конфигурации нейронных сетей

В данной работе были рассмотрены четыре конфигурации сверточных нейронных сетей с одним и двумя сверточными слоями. Ядра сверточных слоев выбраны следующих размеров:

• 2х2

• 3х3

В каждом сверточном слое к картинке может применяться по несколько фильтров. Значение фильтров выбирались из следующих:

* 128
* 64
* 32

Активационная функция на слоях выбирается из следующих:

На выходном слое:

Разработанные программы/скрипты

В директории расположен один файл Lab3.ipynb, в котором и находятся рассмотренные конфигурации, описанные

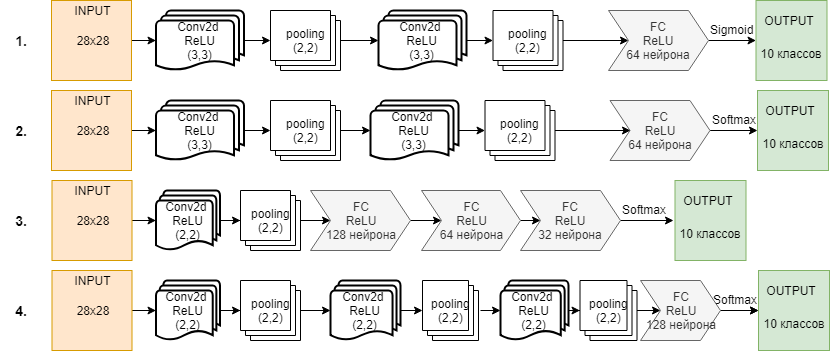
ниже

Рисунок 2. Схемы конфигураций сетей

# Результаты экспериментов

Тестовая инфраструктура:

ОП: Python Jupiter cloud colab.research.google.com

Параметры обучения:

* Кол-во эпох -3
* Скорость обучения – 0.01

В работе рассмотрены несколько конфигураций.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Количество сверточных слоев** | **Количество полно-связных**  **слоев** | **Функции активации** | **Кол.**  **эпох** | **Скор.**  **обуч.** | **Результат** | | | |
| Точность на тренировочном множестве | Точность на тестовом множестве | Время обучения, с | Время теста  одного изобр., 10-5с |
| 1 | 2 | 1 | Relu- relu- relu | 20 | 0,01 | 0.8694 | 0.8831 | 70.355 | 11.8707 |
| 2 | 2 | 1 | Relu- relu- relu | 20 | 0,01 | 0.8446 | 0.8655 | 75.602 | 12.0097 |
| 3 | 1 | 3 | Relu- relu- relu- relu | 20 | 0,01 | 0.9027 | **0.8984** | 72.956 | 11.5165 |
| 4 | 3 | 1 | Relu- relu- relu- relu | 20 | 0,01 | 0.6878 | 0.7626 | 83.862 | 12.6115 |

Таблица 2 Конфигурации и результаты

**Графики точности классификации после обучения на тестовых наборах:**

|  |  |
| --- | --- |
| **№ эксперимента** | **График** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

Таблица 3 Графики точностей на тестовых и обучающем наборах

**Матрица расхождения:**

Для каждой конфигурации строилась матрица расхождений, где мы получаем картину, на какие классы сеть дает верные\неверные ответы.

По итогу можно увидеть, что классификация футболок, пальто и рубашки являются основной причиной ошибок сети, что в общем-то и логично, так как образы их похожи.

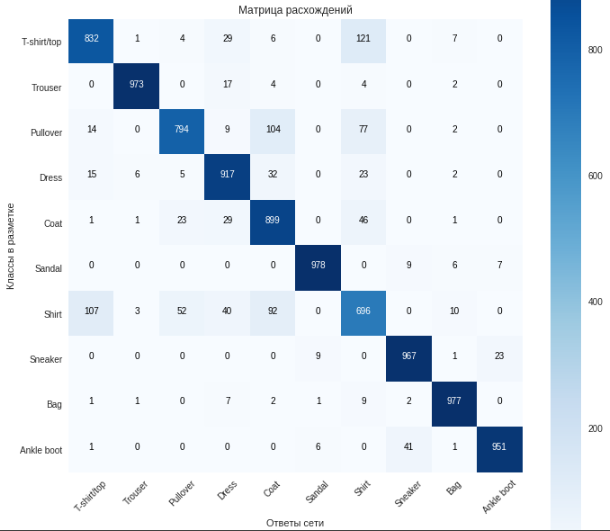


Рисунок 3 Матрица расхождения для 3-ей конфигурации

# Выводы

1. Наилучший результат был получен на нейронной сети с конфигурацией №4. Сверточные нейронные сети дали лучший результат по сравнению с полносвязными нейронными сетями. Максимальная точность в сверточных нейронных сетях – 0.8984, а в полностью связанных – 0.8880.