Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»

Институт Информационных Технологий, Математики и Механики

Направление: Прикладная математика и информатика

Магистерская программа: Компьютерные науки и приложения

**ОТЧЕТ**

по лабораторной работе №3

Тема:

**«Разработка свёрточных сетей»**

Выполнили:

Крапивин Александр (гр. 381803м4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Краснов Александр (гр. 381806м4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Разин Вячеслав (гр.381806м4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Ягудина Анастасия (гр. 381803м4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Проверил:

доцент, к.т.н. Кустикова В.Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2019

Оглавление

[1. Постановка задачи 3](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187372)

[2. Описание набора данных 4](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187373)

[3. Метрика качества решения 5](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187374)

[4. Подготовка данных для нейронной сети 5](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187374)

[5. Разработанные программы 5](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375)

[6. Вычислительные эксперименты 6](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375)

[7. Анализ результатов 7](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375)

**Постановка задачи**

* **Цель работы**

Цель настоящей работы состоит в том, чтобы получить базовые навыки работы с одной из библиотек глубокого обучения (Caffe, Torch, TensorFlow, MXNet или какая-либо другая библиотека на выбор студента) на примере полностью связанных нейронных сетей.

* **Задачи**

Выполнение практической работы предполагает решение следующих задач:

1. Выбор библиотеки для выполнения практических работ курса.

2. Установка выбранной библиотеки на кластере (параметры аутентификации и инструкция по работе с кластером выложена в отдельной задаче в системе redmine).

3. Проверка корректности установки библиотеки. Разработка и запуск тестового примера сети, соответствующей логистической регрессии, для решения задачи классификации рукописных цифр набора данных MNIST (пример разобран в лекционных материалах).

4. Выбор практической задачи компьютерного зрения для выполнения практических работ.

5. Разработка программ/скриптов для подготовки тренировочных и тестовых данных в формате, который обрабатывается выбранной библиотекой.

6. Разработка нескольких архитектур полностью связанных нейронных сетей (варьируются количество слоев и виды функций активации на каждом слое) в формате, который принимается выбранной библиотекой.

7. Обучение разработанных глубоких моделей.

8. Тестирование обученных глубоких моделей.

9. Публикация разработанных программ/скриптов в репозитории на GitHub.

10. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

**Описание набора данных**

Для текущей лабораторной работы был выбран один из представленных на ресурсе Kaggle наборов данных: <https://www.kaggle.com/moltean/fruits>.



Рисунок 1. Некоторые представители набора данных Fruits 360

Этот набор данных состоит из 82110 изображений различных фруктов. Все изображения разбиты на два множества: тренировочный и тестовый наборы. Тренировочный набор состоит из 61488 изображений, а тестовый содержит 20622. Каждое изображение является трехканальным, разешение каждого из них составляет пикселей. На изображениях представлены различные фрукты, каждый из которых может принадлежать ровно одному классу, всего насчитывается 120 различных классов.

Для реализцации полносвязной нейронной сети была выбрана открытая библиотека глубокого обучения Keras, написаннная на языке программирования Python.

**Метрика качества решения задачи**

Для оценки качества данной задачи классификации выбрана метрика "" (точность). Она показывает отношение числа правильных предсказаний к числу предсказаний класса для всех изображений:  
где – количество изображений, – значения, выданные нашей сетью при классификации изображений, – истинные значения классов фруктов с изображения

**Подготовка данных для нейронной сети**

В силу большого количества входных изображений было решено сжать каждое из них до размера пикселей. После этого преобразуем изображение (трёхмерный массив числе от 0 до 255 включительно) в один вектор и приводим каждый элемент вектора к значению с отрезка [0, 1].

Изначально каждое изображение располагается в папке, имя которой совпадает с названием класса, которому принадлежит изображение. Мы преобразуем имя каждого класса к числу от 0 до 119, а после, используя метод one-hot encoding, представляем каждый класс как вектор состоящий из 120 чисел. В итоге мы имеем два двумерных массива – один отвечает за изображения, а второй за класс, которому это изображение принадлежит.

**Разработанные программы**

Реализация свёрточной нейронной сети полностью выполнена в одном файле – fruits\_simple.py. Для обучения и тестирования моделей необходимо выполнить в командной строке следующую команду:   
python fruits\_simple.py epochs batch close  
где epochs – число эпох обучения, batch\_size – размер batch’a

**Вычислительные эксперименты**

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory, а в качестве оптимизатора использовался keras.optimizers.Adam

Рассмотрим сети со следующей архитектурой:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape

==============================================

conv2d\_1 (Conv2D(65, (3,3))) (None, 50, 50, 64)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2D(2,2)) (None, 25, 25, 64)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 40000)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, **close**)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 120)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activate | Epochs | Batch | Close\_n | Accuraty (train) | Accuraty (test) | Time, sec |
| ReLu | 10 | 256 | 1700 | 0.9926 | 0.971 | 126.5 |
| ReLu | 10 | 256 | 1024 | 0.9999 | 0.978 | 91.3 |
| ReLu | 10 | 256 | 512 | 0.9999 | 0.97 | 66.7 |
| ReLu | 5 | 256 | 1024 | 0.9919 | 0.966 | 45 |
| Than | 10 | 256 | 1024 | 0.9999 | 0.979 | 64.4 |
| Sigmoid | 10 | 256 | 1024 | 0.999 | 0.98 | 64.1 |

**Анализ результатов**

Во время данной работы мы убедились, что даже самые простые сверточные сети способны решать задачу классификации изображений лучше многослойный полносвязных сетей – такой эффект достигается из-за операции свёртки. Для свёрточных сетей нужно меньшее количество эпох, чтобы добиться высоких результатов на тренировочной выборке.