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**

Направления подготовки: «Фундаментальная информатика и информационные технологии», «Прикладная математика и информатика»

Магистерские программы: «Компьютерная графика», «Вычислительные методы и суперкомпьютерные технологии»

Образовательный курс «Глубокое обучение»

**ОТЧЕТ**

по лабораторной работе №5

**Применение переноса обучения для решения задачи классификации изображений игры Quick, Draw! с помощью библиотеки TensorFlow**

**Выполнили:**

студенты группы 381706-3м

Бебнев Виктор

Голякова Елена

студенты группы 381703-3м

Митрохина Юлия

Береснева Юлия

Нижний Новгород

2018

Содержание

[Цели 3](#_Toc532936446)

[Задачи 3](#_Toc532936447)

[Постановка задачи 4](#_Toc532936448)

[Описание задачи 4](#_Toc532936449)

[Выбор библиотеки 4](#_Toc532936450)

[Тренировочные и тестовые наборы данных 5](#_Toc532936451)

[Метрики качества решения задачи 5](#_Toc532936452)

[Формат хранения данных 7](#_Toc532936453)

[Разработанная программа 8](#_Toc532936454)

[Тестовые конфигурации сетей 8](#_Toc532936455)

[Конфигурация №1 Прямое использование обученной модели 8](#_Toc532936456)

[Конфигурация №2 Использование структуры обученной модели 9](#_Toc532936457)

[Конфигурация №3 Тонкая настройка параметров модели 10](#_Toc532936458)

[Результаты и анализ экспериментов 11](#_Toc532936459)

[Эксперимент №1 11](#_Toc532936460)

[Эксперимент №2 13](#_Toc532936461)

[Эксперимент №3 15](#_Toc532936462)

[Выводы 17](#_Toc532936463)

# Цели

Цель: исследовать возможности переноса обучения для решения целевой задачи, выбранной изначально для выполнения работ.

# Задачи

Выполнение практической работы предполагает решение следующих задач:

1. Поиск исходной задачи, близкой по смыслу к целевой задаче.
2. Поиск натренированной модели для решения исходной задачи.
3. Выполнение трех типов экспериментов по переносу знаний, описанных в лекциях.
4. Сбор результатов экспериментов.

# Постановка задачи

## Описание задачи

В качестве задачи компьютерного зрения была выбрана задача классификации рукописных изображений, представленная набором данных The Quick Draw dataset – коллекция рисунков пользователей игры [Quick, Draw!](https://quickdraw.withgoogle.com/), содержащая 345 категорий изображений, из которых для лабораторной работы были взяты следующие 5 категорий по 5 000 примеров:

* *spoon - ложка,*
* *paintbrush - кисть,*
* *smiley face - улыбающееся лицо,*
* *wheel - колесо,*
* *bush - куст.*



Рис.1 Примеры рисунков из набора данных Quick, Draw!

Рисунки представлены набором точек – *(x, y, t)* - вектор координат пикселей на плоскости со значением времени первой точки и разметкой, включающей в себя информацию о стране игрока и о том, что требовалось нарисовать.

## Выбор библиотеки

Для выполнения данной практической работы была выбрана библиотека глубокого обучения TensorFlow, использующая в качестве интерфейса язык программирования Python.

Для проверки корректности установки библиотеки была выполнена разработка и запуск тестового примера сети для задачи классификации рукописных цифр из набора данных MNIST. На данном примере достигнутая точность равна 0,931.

## Тренировочные и тестовые наборы данных

В используемом наборе данных 25 000 (5 категорий по 5 000 примеров) изображений, из которых использовали как:

* тренировочных – 17 500 (70%);
* валидационных – 2 500 (10%);
* тестовых – 5 000 (20%)

В ходе обучения наборы тщательно перемешиваются перед выборками.

## Метрики качества решения задачи

Качество решения выбранной задачи оценивается с использованием различных метрик:

* accuracy – точность – это отношение числа верно классифицированных изображений к общему числу изображений в выборке

Введем некоторые обозначения для определения следующих величин:

* TP — истино-положительное решение;
* TN — истино-отрицательное решение;
* FP — ложно-положительное решение;
* FN — ложно-отрицательное решение;

тогда можем определить вычисление следующих метрик по формулам:

* precision – это доля изображений действительно принадлежащих данному классу относительно количества всех изображений, которые сеть отнесла к этому классу
* recall – полнота – это доля найденных сетью изображений, принадлежащих классу относительно количества всех изображений этого класса в тестовой выборке
* f1-score – F-мера – это гармоническое среднее между точностью и полнотой

Формула этой метрики позволяет одновременно учитывать значения точности и полноты, поскольку придает им одинаковый вес, а значит будет одинаково падать при изменении любой из них. Отметим, что чем выше показатели точности и полноты, тем лучше классификатор.

* support – это количество примеров, классифицированных к определенному классу в выборке;

Таким образом, по классам в сумме значение этой метрики равняется числу изображений в выборке, конкретно в работе 5000 изображений.

При наличии нескольких наборов данных и высчитанных для них значений метрик precision и recall, можно высчитать также метрики:

* micro avg – это micro средние значения, рассчитываемые для precision и recall:

где *c* – отметка класса.

Из формул следует, что при нескольких классах значение значения micro-метрик всегда совпадают, поскольку .

* macro avg – это macro средние значения, рассчитываемые для precision и recall:

где *c* – отметка класса, *#c* – число классов.

* weighed avg – это взвешенные средние значения, рассчитываемые для precision и recall с весами:

где - количество примеров в классе.

## Формат хранения данных

Исходный формат хранения данных - используемый набор данных хранится на сервере-источнике в виде бинарных файлов:

* <https://console.cloud.google.com/storage/browser/quickdraw_dataset/full/binary>
* <https://drive.google.com/drive/folders/1ku5w3lUnQnr8cmA4J9Ra9yV5Xd7y7awk>

в формате, указанном в Таблице 1 ниже:

**Таблица 1**

**Исходный формат хранения данных**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| key\_id | 64-bit unsigned integer | Уникальный идентификатор |
| word | string | Категория изображения |
| recognized | boolean | Указано, что слово было опознано в игре |
| timestamp | datetime | Дата создания рисунка |
| countrycode | string | Код страны участника |
| drawing | string | JSON массив, содержащий вектор рисования |

Формат данных на входе сети – в ходе проведения экспериментов данные подавались в следующих видах:

* 3-мерный массив нормализованных координат данных, длины равной максимальной длине набора точек рисунка (недостающие элементы заполнялись нулями);
* нормализованные бинарные изображения, отмасштабированные по размеру 28х28, полученные с помощью библиотеки OpenCV из данных источника;
* в ходе подготовки данных все данные разметки были векторизованы.

# Разработанная программа

Разработанная программа содержит следующие файлы в директории src:

* parse\_data.py - подготовка данных;
* run\_create\_dataset.py - выбор данных, которые распознала сеть google, распределение на run, train и validate выборки;
* run\_train\_conv.py - непосредственно обучение сети, использует класс NetworkBase;
* NetworkBase.py - описание основных методов для работы с нейронной сетью;
* ConvNetwork.py – описание архитектуры полносвязной нейронной сети;
* run\_statisctic\_conv.py - получение статистики по итогам обучения.

# Тестовые конфигурации сетей

Для конфигурации сетей в качестве исходной задачи была выбрана задача …, которая содержит N классов изображений, в качестве объекта переноса знаний была выбрана нейронная сеть …, решающая эту задачу. Натренированная модель содержит M <adj> слоев, … .

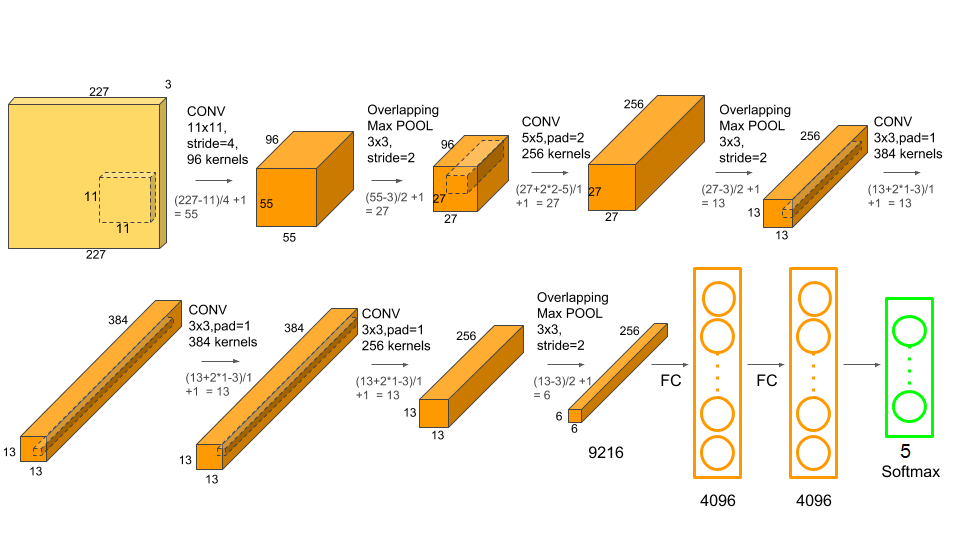


Рис.2 Схема тестовых архитектур сетей

Ниже рассмотрели следующие типы конфигураций:

## Конфигурация №1 Прямое использование обученной модели

Перенос архитектуры

Ниже рассмотрели следующие типы конфигураций:

Рис.3 Конфигурация №1

Рассматривалась сеть из 3-х блоков:

## Конфигурация №2 Использование структуры обученной модели

Перенос весов

Ниже рассмотрели следующие типы конфигураций:

<image>

Рис.4 Конфигурация №2

Рассматривалась сеть из 3-х блоков:

## Конфигурация №3 Тонкая настройка параметров модели

Часть архитектуры, остальное перенастраивается

Ниже рассмотрели следующие типы конфигураций:

<image>

Рис.5 Конфигурация №3

Рассматривалась сеть из 3-х блоков:

# Результаты и анализ экспериментов

В ходе работы были проведены 3 эксперимента на машине со следующими характеристиками:

* операционная система Ubuntu 16.04, CUDA 9.1
* оперативная память RAM 60 GB
* видеокарта Geforce GTX 1060 6 GB x2 Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz.
* Python 3.6, TensorFlow 1.12.

Проведем сравнительный анализ полученных результатов в ходе проведенных экспериментов и опишем его в виде Таблицы 5.

**Таблица 5**

**Сводная таблица результатов обучения по проведенным экспериментам**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Число блоков** |  | **Результаты обучения** | | |
| Точность | Ошибка | Время |
| **1** | 3 |  | 0.95 | 0.2 | 1ч 15 мин |
| **2** | 4 |  | 0.35 | 1.37 | 1ч 05 мин |
| **3** | 3 |  | 0.95 | 0.18 | 1ч 45 мин |

Таблица 5 позывает, что слой нормализации дает пусть и небольшой, но выигрыш в функции потерь и достаточно хорошее значение точности. Возможно, при масштабируемости данная конфигурация покажет еще лучшие показатели.

# Выводы

В ходе лабораторной работы исследовали возможности переноса обучения для решения целевой задачи, выбранной изначально для выполнения практических работ. Искали исходную задачу, близкую по смыслу к целевой задаче, и натренированную модель для решения исходной задачи.

Выполнили три типа экспериментов, описанных выше в отчете, для каждого из которых была сконструирована определенная конфигурация, было произведено обучение модели на выбранном наборе данных задачи [Quick, Draw!](https://quickdraw.withgoogle.com/), провели тестирование программы, в ходе которого собирали метрики качества решения задачи.

Собранные метрики, показатели и результаты были визуализированы с помощью модуля TensorBoard библиотеки TensorFlow, полученные графики представлены выше в данном отчёте.

В ходе экспериментов было установлено, что ….

Таким образом, цель работы - исследовать возможности переноса обучения для решения целевой задачи, выбранной изначально для выполнения работ - была достигнута.