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: Фундаментальная информатика и информационные технологии

Магистерская программа: Когнитивные системы

**ОТЧЕТ**

по лабораторной работе №5

Тема:

**«Применение переноса обучения для решения задачи, поставленной во второй лабораторной работе»**

Выполнили:

Краснов Александр (гр. 381806м4)

Разин Вячеслав (гр. 381806м4)

Ахмедов Саркар (гр. 381806м4)

Проверила:

доцент, к.т.н. Кустикова В.Д.

Нижний Новгород

2019

Оглавление

[Введение 3](#_Toc34578514)

[Постановка задачи 4](#_Toc34578515)

[Описание набора данных 5](#_Toc34578516)

[Метрика качества решения задачи 6](#_Toc34578517)

[Подготовка данных для нейронной сети 7](#_Toc34578518)

[Выбор натренированной модели 8](#_Toc34578519)

[Тестовые конфигурации сетей 9](#_Toc34578520)

[Результаты экспериментов 12](#_Toc34578524)

[Анализ результатов 13](#_Toc34578525)

# Введение

Пусть - множество изображений, – множество номеров (или наименований) классов. Существует неизвестная *целевая зависимость* – отображение , значения которой известны только на объектах конечной обучающей выборки . Требуется построить алгоритм , способный классифицировать произвольный объект .

В рамках данной лабораторной работы ставится задача классификации изображений из [американского языка жестов (ASL Alphabet).](https://www.kaggle.com/grassknoted/asl-alphabet)

# 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.

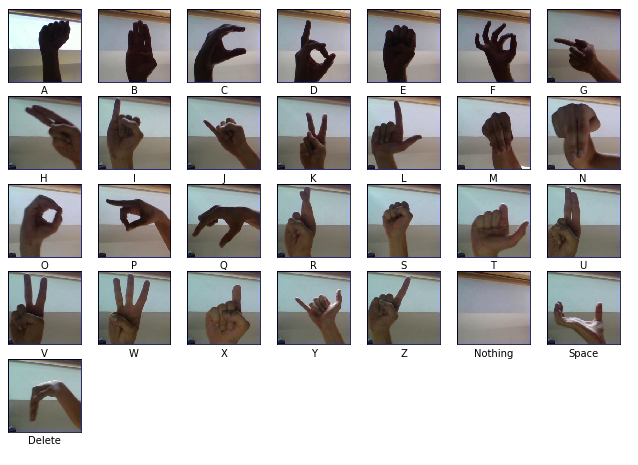
8. Тестирование обученных глубоких моделей.

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 лабораторных работ был выбран один из представленных на Kaggle набор данных <https://www.kaggle.com/grassknoted/asl-alphabet> (ASL Alphabet).



*Рис.1. Представители всех классов набора данных ASL alphabet*

Набор данных ASL Alphabet представляет собой набор изображений из американского языка жестов. Набор включает в себя 87 000 изображений размером 200х200 пикселей. Есть 29 классов, из которых 26-для букв A-Z и 3 класса SPACE, DELETE и NOTHING. В каждом классе 3000 изображений. Для тестирования использовано 10% изображений.

Для реализации различных архитектур нейронных сетей была выбрана открытая библиотека глубокого обучения Keras, написанная на языке программирования Python.

# Метрика качества решения задачи

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

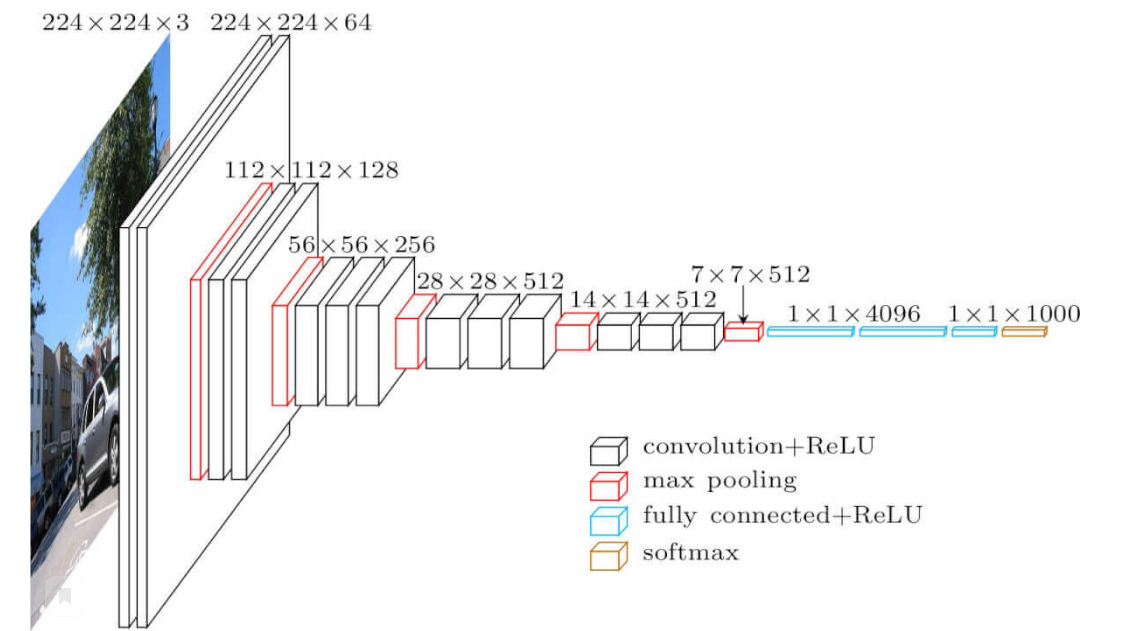
# Подготовка данных для нейронной сети

В силу большого количества входных изображений было решено сжать каждое из них до размера 64 x 64 пикселей. Сжатие происходит с сохранением соотношения сторон, так как все исходные изображения имеют разрешение 200 x 200. После этого изображения добавляются в массив, нормализуются делением на 255 и разделяются на тренировочную и тестовую выборку в соотношении 9 к 1. Также создаётся отдельный массив для меток, отвечающих за принадлежность изображения к одному из 29 классов. Используется метод one-hot encoding.

# Выбор натренированной модели

В качестве натренированной модели для решения целевой задачи была выбрана нейронная сеть VGG-16.

VGG16 — модель сверточной нейронной сети, предложенная K. Simonyan и A. Zisserman из Оксфордского университета в статье “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Модель достигает точности 92.7% — топ-5, при тестировании на ImageNet в задаче распознавания объектов на изображении. Этот датасет состоит из более чем 15 миллионов изображений, принадлежащих к 1000 классам.

VGG16 — одна из самых знаменитых моделей, отправленных на соревнование ILSVRC-2014. Она является улучшенной версией AlexNet, в которой заменены большие фильтры (размера 11 и 5 в первом и втором сверточном слое, соответственно) на несколько фильтров размера 3х3, следующих один за другим. Сеть VGG16 обучалась на протяжении нескольких недель при использовании видеокарт NVIDIA TITAN BLACK.

Архитектура сети VGG-16 представлена ниже:

*Рис.2. Архитектура VGG-16*

# Тестовые конфигурации сетей

Общая конфигурация сети:

|  |
| --- |
| Input (50 \* 50 RGB image) |
| Conv (3,3), 64, relu |
| Conv (3,3), 64, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 128, relu |
| Conv (3,3), 128, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 256, relu |
| Conv (3,3), 256, relu |
| Conv (1,1), 256, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 512, relu |
| Conv (3,3), 512, relu |
| Conv (1,1), 512, relu |
| **Maxpooling(2,2), 2** |
| Conv (3,3), 512, relu |
| Conv (3,3), 512, relu |
| Conv (1,1), 512, relu |
| **Maxpooling(2,2), 2** |
|  |
| FC – 512, sigmoid |
| FC – 29, softmax |

**Эксперимент 1**

Использование структуры глубокой модели, построенной для решения исходной «Задачи А», с целью обучения аналогичной модели для решения «Задачи В»

* Предполагается, что модель, построенная для решения исходной задачи, обучается на данных, подготовленных для решения целевой задачи
* При этом веса модели инициализируются случайным образом
* Эксперимент реализует перенос знаний для родственных доменов

Т. е. в этом эксперименте используем только структуру модели. Чтобы применить данную модель VGG16 к нашей задаче, заменим полносвязный классификатор и заменим на последнем слое количество выходов с 1000 на 29.

**Эксперимент 2**

Использование модели, построенной для решения исходной «Задачи А», в качестве фиксированного метода извлечения признаков при построении модели, решающей «Задачу В»

* Идея данного подхода состоит в том, чтобы удалить из глубокой модели классификатор (последние полностью связанные слои) и рассматривать начальную часть сети как метод выделения признаков
* Взамен старого классификатора можно поместить новый классификатор (например, другой набор полностью связанных слоев или машину опорных векторов) и обучить его на признаках, построенных с использованием начальной части сети
* Эксперимент реализует перенос признакового описания

Во втором эксперименте возьмем сверточную основу VGG16, обученную на наборе ImageNet.

Пропустим наш набор данных через предварительно обученную сверточную основу VGG16, таким образом выделив признаки.

Далее возьмем полносвязный классификатор и обучим его на полученных признаках.

**Эксперимент 3**

Тонкая настройка параметров модели, построенной для решения исходной «Задачи А», с целью решения «Задачи В»

* Последние слои глубокой модели, соответствующие классификатору, который решает «Задачу А», заменяются новым классификатором (например, набором полностью связанных слоев с другим количеством выходов)
* Полученная модель обучается как единая система
* Эксперимент реализует перенос обучения на основе экземпляров

Последние слои глубокой модели VGG16, соответствующие классификатору, который решает ImageNet, заменим новым полносвязным классификатором с количеством выходов 29. И обучим модель как единую систему.

# Результаты экспериментов

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory режиме GPU.

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | 1  Использование только структуры глубокой модели | 2  Извлечение признаков и полносвязный классификатор | 3  Тонкая настройка |
| Батч | 128 | 128 | 128 |
| Количество эпох | 3 | 10 | 3 |
| Оптимизатор | adam | adam | adam |
| Скорость обучения | 0.0001 | 0.001 | 0.0001 |
| Время обучения (сек) | 61.08 | 80.38 | 60.78 |
| Точность (Accuracy) на тренировочном наборе, % | 99.13 | 95.41 | 99.04 |
| Ошибка на тренировочном наборе | 0.02 | 0.17 | 0.03 |
| Точность (Accuracy) на тестовом наборе, % | 99.24 | 94.43 | **99.43** |
| Ошибка на тестовом наборе | 0.02 | 0.19 | 0.01 |

# Анализ результатов

Полученные модели показали высокую точность (больше 90%), но они уступают сверточным нейронным сетям из предыдущих работ по точности и по времени работы. Поэтому использование переноса обучения в данной задаче нецелесообразно.