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 №3

Тема:

**«Разработка сверточной нейронной сети»**

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](#_Toc34578578)

[Постановка задачи 4](#_Toc34578579)

[Описание набора данных 5](#_Toc34578580)

[Метрика качества решения задачи 6](#_Toc34578581)

[Подготовка данных для нейронной сети 7](#_Toc34578582)

[Тестовые конфигурации сетей 8](#_Toc34578583)

[Результаты экспериментов 14](#_Toc34578584)

[Анализ результатов 16](#_Toc34578585)

# 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.

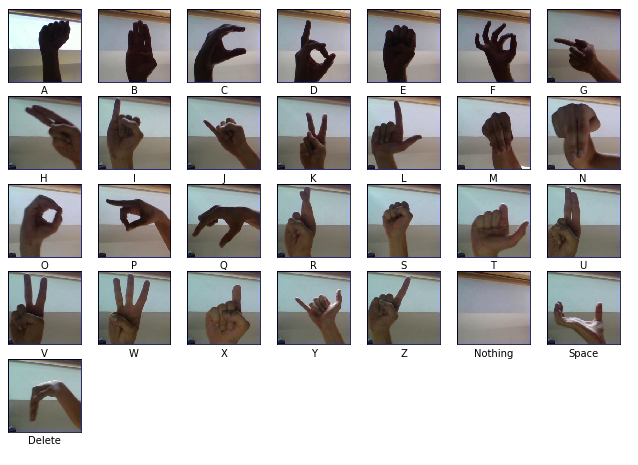
8. Тестирование обученных глубоких моделей.

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.

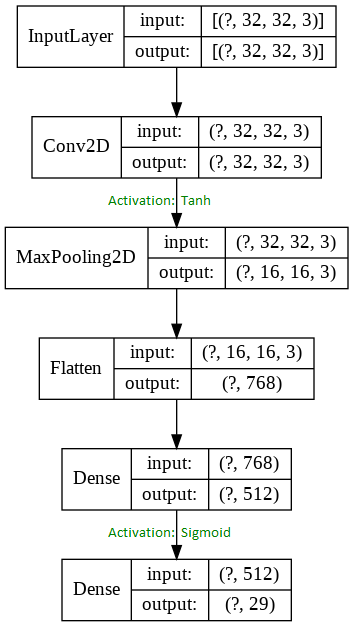
# Метрика качества решения задачи

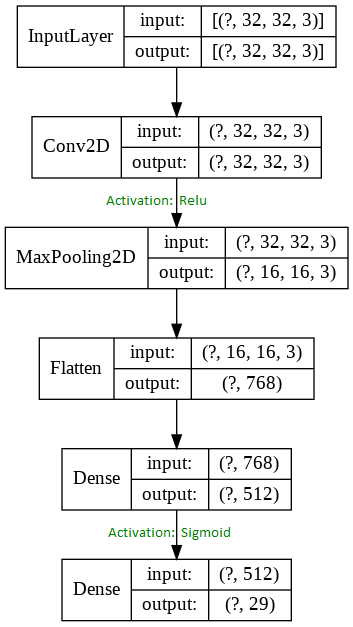
Для оценки качества задачи классификации выбрана метрика "" (точность), показывающая отношение числа правильных предсказаний к числу предсказаний класса для всех изображений:  
где – количество изображений, – значения, выданные сетью при классификации изображений, – истинные значения классов элементов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

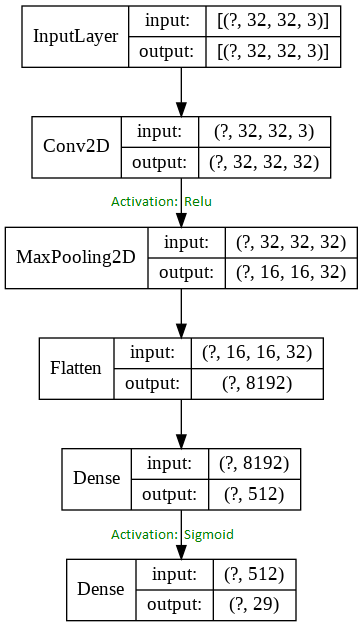
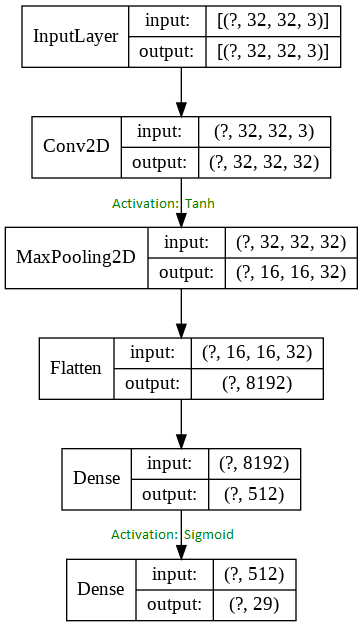
# Подготовка данных для нейронной сети

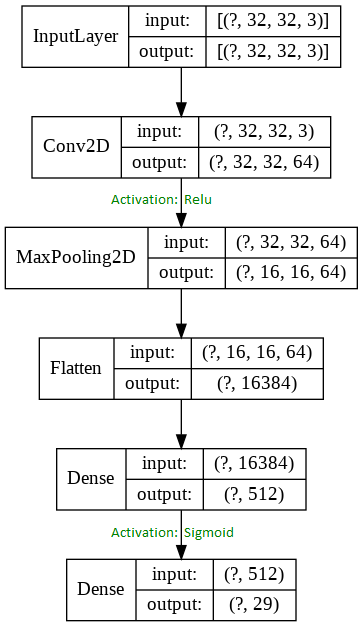
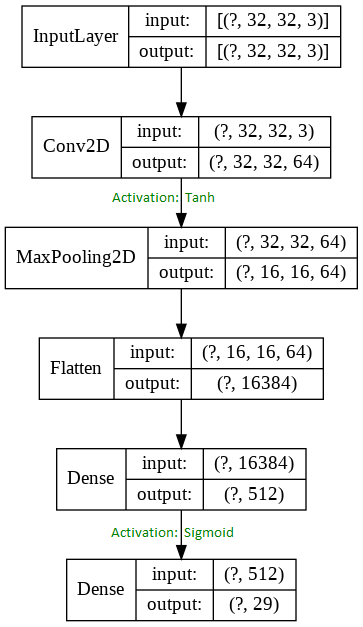
В силу большого количества входных изображений было решено сжать каждое из них до размера 64 x 64 пикселей. Сжатие происходит с сохранением соотношения сторон, так как все исходные изображения имеют разрешение 200 x 200. После этого изображения добавляются в массив, нормализуются делением на 255 и разделяются 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.

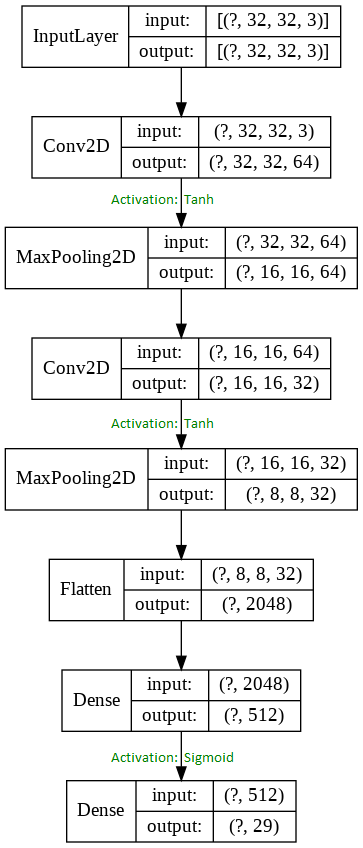
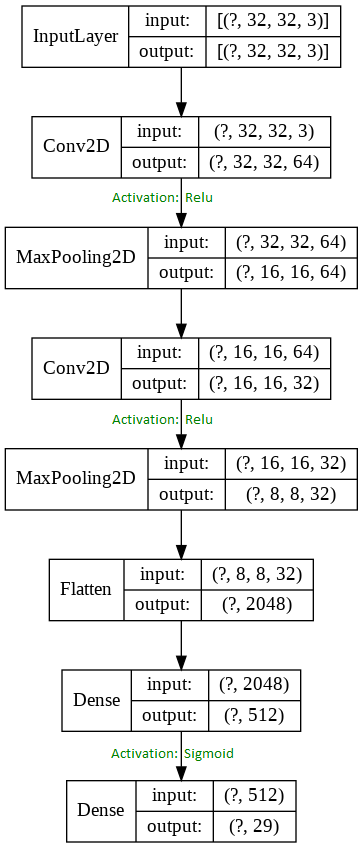
# Тестовые конфигурации сетей

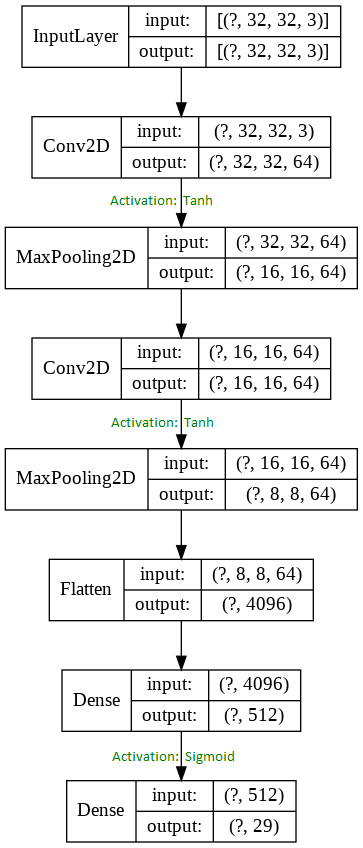
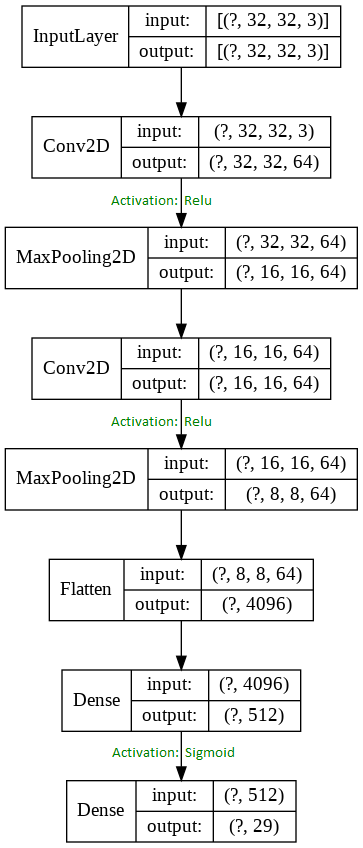
В ходе лабораторной работы были изучены восемь различных конфигураций сетей. Варьировались количество скрытых слоев и функции активации. Схемы конфигураций представлены ниже:

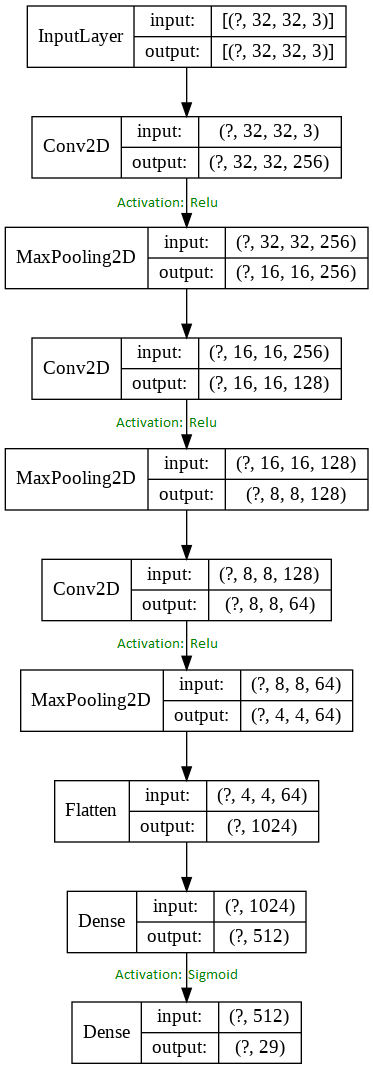
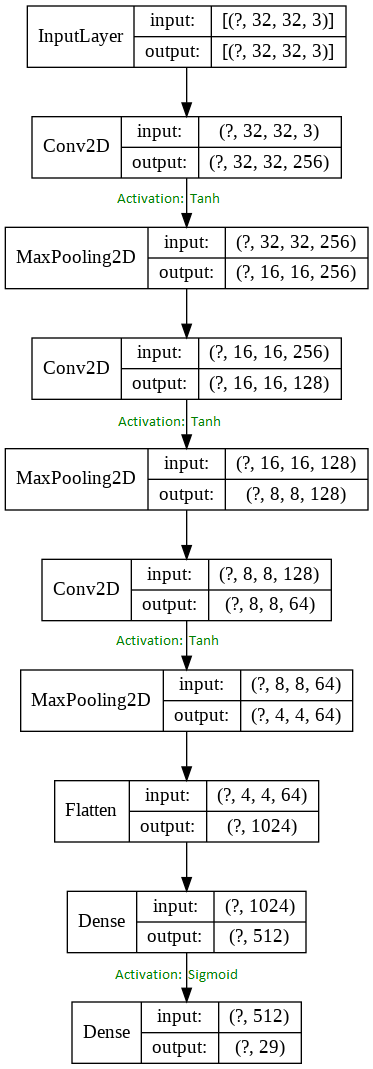












# Результаты экспериментов

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory режиме GPU.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модель сети | 1 | 2 | 3 | 4 | 5 | 6 |
| Количество скрытых нейронов | 512  29 | 512  29 | 512  29 | 512  29 | 512  29 | 512  29 |
| Количество фильтров | 3 | 3 | 32 | 32 | 64 | 64 |
| Количество свёрточных слоёв | 1 | 1 | 1 | 1 | 1 | 1 |
| Функция активации на свёрточных слоях | relu | tanh | relu | tanh | relu | tanh |
| Батч | 128 | 128 | 128 | 128 | 128 | 128 |
| Количество эпох | 5 | 5 | 5 | 5 | 5 | 5 |
| Скорость обучения | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| Оптимизатор | adam | adam | adam | adam | adam | adam |
| Время обучения, сек | 20.53 | 20.16 | 31.37 | 31.96 | 48.66 | 48.86 |
| Точность (Accuracy) на тренировочном наборе, % | 80.80 | 88.69 | 98.02 | 98.05 | 99.09 | 98.79 |
| Ошибка на тренировочном наборе | 0.68 | 0.46 | 0.09 | 0.11 | 0.05 | 0.07 |
| Точность (Accuracy) на тестовом наборе, % | 80.68 | 89.49 | 97.72 | 98.19 | 99.08 | 98.24 |
| Ошибка на тестовом наборе | 0.64 | 0.42 | 0.09 | 0.1 | 0.05 | 0.08 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модель сети | 7 | 8 | 9 | 10 | 11 | 12 |
| Количество скрытых нейронов | 512  29 | 512  29 | 512  29 | 512  29 | 512  29 | 512  29 |
| Количество фильтров | 64  32 | 64  32 | 64  64 | 64  64 | 256  128  64 | 256  128  64 |
| Количество свёрточных слоёв | 2 | 2 | 2 | 2 | 3 | 3 |
| Функция активации на свёрточных слоях | relu | tanh | relu | tanh | relu | tanh |
| Батч | 128 | 128 | 128 | 128 | 128 | 128 |
| Количество эпох | 5 | 5 | 5 | 5 | 5 | 5 |
| Скорость обучения | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| Оптимизатор | adam | adam | adam | adam | adam | adam |
| Время обучения, сек | 39.11 | 39.2 | 47.14 | 46.98 | 115.26 | 116.44 |
| Точность (Accuracy) на тренировочном наборе, % | 99.39 | 99.15 | 99.48 | 99.55 | 99.7 | 99.72 |
| Ошибка на тренировочном наборе | 0.03 | 0.05 | 0.03 | 0.03 | 0.02 | 0.01 |
| Точность (Accuracy) на тестовом наборе, % | 98.79 | 99.24 | 98.83 | 99.56 | **99.9** | 99.22 |
| Ошибка на тестовом наборе | 0.05 | 0.04 | 0.04 | 0.03 | 0.02 | 0.03 |

# Анализ результатов

Варьируемые параметры:

* количество сверточных слоёв (троек, состоящих из свёртки, функции активации и пространственного объединения) от 1 до 3
* количество фильтров (3, 32, 64, 128, 256)
* функции активации на сверточных слоях (relu, tanh).

В качестве классификатора во всех конфигурациях использовался один полносвязный скрытый слой с 512 нейронами и функцией активации sigmoid, зарекомендовавшей себя в предыдущей работе. В сверточных слоях в качестве пространственного объединения всегда использовался maxpooling фильтром 2x2 шагом 2.

Итоговые результаты:

* Конфигурации 1-2: лучшую точность показывает сеть с tanh – 89.49%. Этот результат меньше, чем у ранее полученной лучшей полносвязной сети. Трёх фильтров явно недостаточно. Попробуем использовать количество фильтров, близкое к числу классов.
* Конфигурации 3-4: лучшую точность показывает сеть с tanh – 98.19%. Использовав 32 фильтра, мы улучшили показатели на 10% и обошли лучшую полносвязную сеть и по времени, и по результату. Попробуем ещё увеличить число фильтров.
* Конфигурации 5-6: лучшую точность показывает сеть с relu – 99.08%. Использовав 64 фильтра, мы улучшили показатели примерно на 1%. Попробуем добавить сверточный слой.
* Конфигурации 7-8: лучшую точность показывает сеть с tanh – 99.24%. Добавив сверточный слой с 32 фильтра, мы улучшили показатели примерно на 0.1%. Попробуем использовать на втором сверточном слое 64 фильтра, как и на первом.
* Конфигурации 9-10: лучшую точность показывает сеть с tanh – 99.56%. Изменив число фильтров на втором сверточном слое, добились улучшения 3%. Попробуем использовать три сверточных слоя.
* Конфигурации 11-12: лучшую точность показывает сеть с relu – 99.9%. Использовав 3 сверточных слоя (256 на перовом слое, 128 на втором слое, 63 на третьем слое) мы достигли максимальной точности.

Итак, максимальный результат показала одиннадцатая модель с тремя свёрточными слоями с relu на каждом. Использование трёх свёрточных слоёв оправдывает себя, когда нужна очень высокая точность. Однако это значительно увеличило время работы сети - 116 сек против 40 сек с одним свёрточным слоем. Функция активации tanh показала себя значительно лучше лишь в начале при использовании одного сверточного слоя с 3 фильтрами. В дальнейшем побеждали модели как с relu, так и с tanh со скромным разрывом. Стоит добавить, что все модели недообучались, и что на большем количестве эпох можно довести результат до 100%.

Таким образом, свёрточные сети отлично себя показали, значительно опередив полносвязные модели и по времени, и по точности.