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 №4

Тема:

**«Начальная настройка весов полносвязных и сверточных нейронных сетей»**

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](#_Toc34578550)

[Постановка задачи 4](#_Toc34578551)

[Описание набора данных 5](#_Toc34578552)

[Метрика качества решения задачи 6](#_Toc34578553)

[Подготовка данных для нейронной сети 7](#_Toc34578554)

[Автокодировщик 8](#_Toc34578555)

[Тестовые конфигурации сетей 9](#_Toc34578556)

[Результаты экспериментов 10](#_Toc34578557)

[Анализ результатов 11](#_Toc34578558)

# 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.

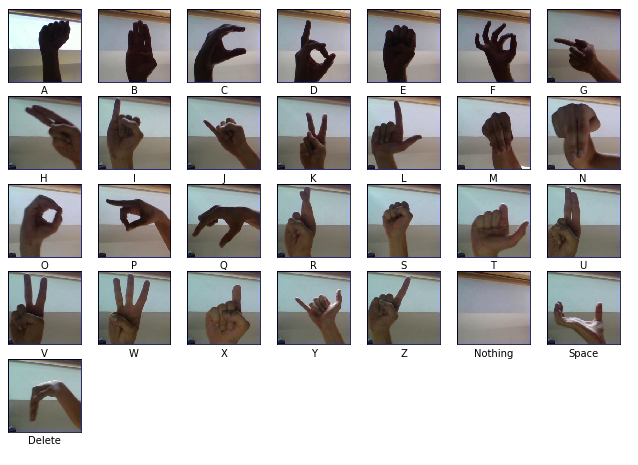
8. Тестирование обученных глубоких моделей.

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

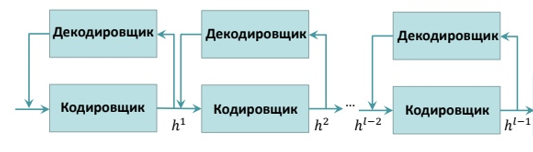
В силу большого количества входных изображений было решено сжать каждое из них до размера 64 x 64 пикселей. Сжатие происходит с сохранением соотношения сторон, так как все исходные изображения имеют разрешение 200 x 200. После этого изображения добавляются в массив, нормализуются делением на 255 и разделяются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

# Автокодировщик

В качестве алгоритма без учителя для инициализации начальных весов сети будем использовать автокодировщик.

Автокодировщик – нейронная сеть, которая пытается максимально приблизить значения выходного сигнала к входному, т.е. наилучшим образом аппроксимировать тождественное преобразование. Сеть разделяется на две принципиальные части: кодировщик h = f(x), обеспечивающий кодирование входных данных, и декодировщик y = g(h), восстанавливающий по коду вход. Cкрытый слой h описывает код, используемый для представления переданных на вход данных. Идеальный автокодировщик позволяет добиться равенства входа и выхода x = g(f(x)) и по существу обеспечивает копирование входного сигнала.

Чаще всего автокодировщики применяют каскадно для обучения глубоких (многослойных) сетей. Стекавтокодировщиков (рис. 2), может быть применен при работе с глубокими сетями. Каждый автокодировщик обучается как сеть прямого распространения и работает с одним слоем. Это позволяет постепенно снижать размерность и настраивать параметры.



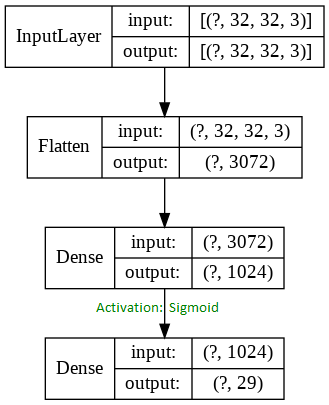
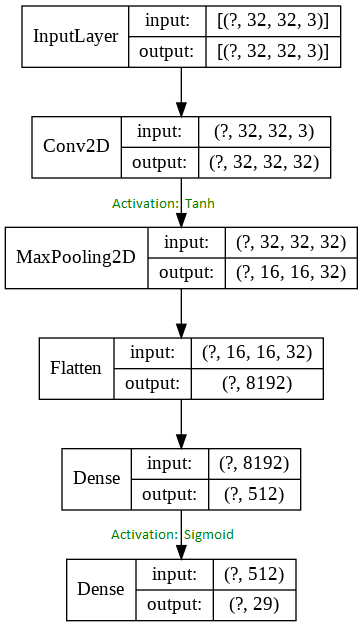
*Рис.2. Стек автокодировщиков*

# Тестовые конфигурации сетей

В ходе работы были рассмотрены одна полносвязная и одна сверточная модели нейронной сети из предыдущих работ. На всех скрытых и сверточных слоях применён автокодировщик. Схемы данных конфигураций представлены ниже:

*Рис.3.* FCNN модель

*Рис.4.* CNN модель



# Результаты экспериментов

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory режиме GPU.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | 1  FCNN | 1  FCNN and AE | 2  CNN | 2  CNN and AE |
| Количество скрытых нейронов | 1024  29 | 1024  29 | 512  29 | 512  29 |
| Количество фильтров | - | - | 32 | 32 |
| Количество свёрточных слоёв | - | - | 1 | 1 |
| Батч | 128 | 128 | 128 | 128 |
| Количество эпох | 10 | 10 | 5 | 5 |
| Время обучения автокодировщика(-ов) (сек) | - | 31.92 | - | 36.67 |
| Время обучения сети (сек) | 52.69 | 26.45 | 31.96 | 14.59 |
| Точность (Accuracy) на тренировочном наборе, % | 89.24 | 91.01 | 98.05 | 98.24 |
| Ошибка на тренировочном наборе | 0.4 | 0.33 | 0.11 | 0.11 |
| Точность (Accuracy) на тестовом наборе, % | 89.31 | **91.49** | 98.19 | **98.27** |
| Ошибка на тестовом наборе | 0.4 | 0.31 | 0.1 | 0.1 |

# Анализ результатов

В данной работе использованы полносвязная сеть с одним скрытым слоем и сверточная сеть с одним сверточным и одним скрытым слоем из предыдущих работ. Для эксперимента выбраны не самые эффективные по точности модели, чтобы лучше оценить перспективы использования автокодировщиков.

Для полносвязной модели применение автокодировщика дало 2%, для свёрточной – 0.1%. С учётом временных затрат на обучение автокодировщика, применение данного подхода малоэффективно. Предпочтительнее использовать более другие модели сверточных сетей с большим количеством фильтров или сверточных слоёв.