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 №2

Тема:

**«Разработка полносвязной 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](#_Toc32823563)

[Постановка задачи 4](#_Toc32823564)

[Описание набора данных 5](#_Toc32823565)

[Метрика качества решения задачи 6](#_Toc32823566)

[Подготовка данных для нейронной сети 7](#_Toc32823567)

[Тестовые конфигурации сетей 8](#_Toc32823568)

[Результаты экспериментов 8](#_Toc32823569)

[Анализ результатов 10](#_Toc32823570)

# 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 немецких дорожных знаков](https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign) (The German Traffic Sign Benchmark).

# 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.

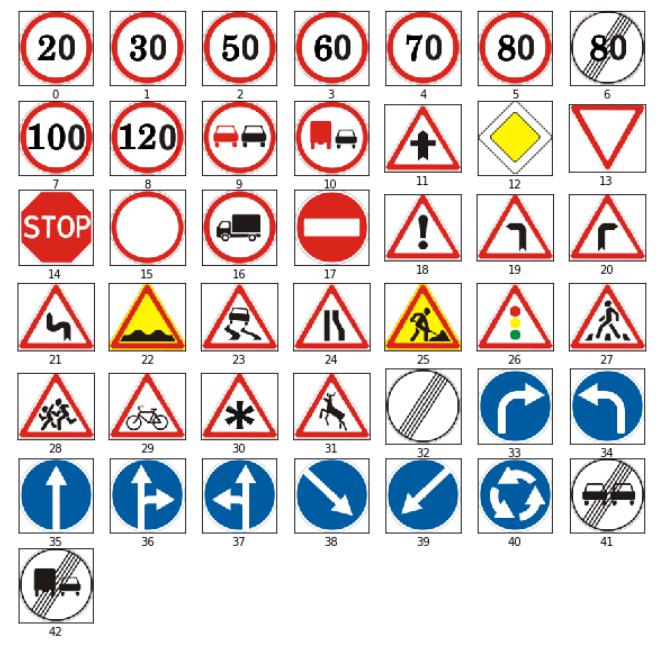
7. Обучение разработанных глубоких моделей.

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/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign> (The German Traffic Sign Benchmark).

*Рис.1. Представители всех классов набора данных The German Traffic Sign Benchmark*

Данный набор содержит в общей сложности более 51840 изображений разных разрешений, 39209 из которых используются для тренировки нейронной сети, а 12631 - для тестирования. На каждом изображении содержится один дорожный знак. Набор данных содержит 43 класса 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 50 x 50 пикселей. Сжатие происходит с сохранением соотношения сторон, уменьшенное изображение находится по центру, а недостающие пиксели имеют черный цвет. 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 обучающей выборки находится в пронумерованной папке, номер которой соответствует конкретному классу изображений от 0 до 42. Используя метод one-hot encoding, представляем каждый класс как вектор состоящий из 43 чисел. В итоге мы имеем два двумерных массива: массив изображений, готовых к классификации, и массив меток, отвечающий за то, какому классу данные изображения принадлежат.

# Тестовые конфигурации сетей

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

|  |  |  |
| --- | --- | --- |
| Input: Input layer | ReLu: Dense | Softmax: Dense |
| (50,50) -> (None, 7500) | (None, 7500)->(None, 512) | (None, 512)->(None, 43) |

*Рис 2.* Модель 1.

|  |  |  |
| --- | --- | --- |
| Input: Input layer | Sigmoid: Dense | Softmax: Dense |
| (50,50) -> (None, 7500) | (None, 7500)->(None, 512) | (None, 512)->(None, 43) |

*Рис 3.* Модель 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Input: Input layer | ReLu: Dense | ReLu: Dense | Softmax: Dense |
| (50,50)->(None, 7500) | (None, 7500)->(None, 1024) | (None, 1024)->(None, 512) | (None, 512)->(None, 43) |

*Рис 4.* Модель 3.

|  |  |  |  |
| --- | --- | --- | --- |
| Input: Input layer | Sigmoid: Dense | Sigmoid: Dense | Softmax: Dense |
| (50,50)->(None, 7500) | (None, 7500)->(None, 1024) | (None, 1024)->(None, 512) | (None, 512)->(None, 43) |

*Рис 5.* Модель 4.

# Результаты экспериментов

Все вычислительные эксперименты выполнялись с помощью Google Colaboratory режиме GPU.

Входные данные:

* Train\_images = (39209, 7500)
* Train\_labels = (39209, 43)
* Test\_images = (12631, 7500)
* Test\_labels = (12631, 43)
* loss = categorical\_crossentropy
* optimizer = Adam
* learning\_rate = 0.001
* batch\_size = 256
* num\_epochs = 20

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Train accuracy | Train loss | Train time (sec) | Test accuracy | Test loss | Test time (sec) |
| 1 | 0.9573 | 0.1988 | 25.4944 | 0.8347 | 1.1182 | 1.1658 |
| 2 | 0.9856 | 0.0828 | 24.6878 | **0.8549** | 0.6419 | 1.123 |
| 3 | 0.9692 | 0.1225 | 26.1582 | 0.8388 | 1.3031 | 1.1561 |
| 4 | 0.98 | 0.0847 | 26.8309 | 0.8539 | 0.6414 | 1.0997 |

# Анализ результатов

Лучший результат показала первая модель нейросети с одним слоем и функцией активации sigmoid. Использование функции активации relu не оправдало себя. Использование дополнительного скрытого слоя в 3 и 4 моделях дало очень близкие результаты с 1 и 2 моделью соответственно.