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**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Магистерская программа: «Компьютерная графика»

Образовательный курс «Глубокое обучение»

**ОТЧЕТ**

По лабораторной работе №4

**Начальная настройка весов полностью связанных нейронных сетей**

**Выполнили:**

Студенты группы 381706-2м

Привалов Даниил

Бабаев Иван

Зубарева Екатерина

Фадеев Алексей

Нижний Новгород

2018

Содержание

[Цели 3](#_Toc535969270)

[Задачи 4](#_Toc535969271)

[Решаемая задача 5](#_Toc535969272)

[Метрика качества решения задачи 6](#_Toc535969273)

[Тренировочные и тестовые наборы данных 6](#_Toc535969274)

[Разработанные программы/скрипты 7](#_Toc535969275)

[Тестовые конфигурации сетей 8](#_Toc535969276)

[Результаты экспериментов 9](#_Toc535969277)

[Анализ результатов 10](#_Toc535969278)

# Цели

1. Цель настоящей работы состоит в том, чтобы использовать методы обучения без учителя для настройки начальных значений весов сетей, построенных при выполнении предшествующих практических работ.

# Задачи

Выполнение лабораторной работы предполагает решение следующих задач:

1. Выбор архитектур нейронных сетей, построенных при выполнении предшествующих практических работ.
2. Выбор методов обучения без учителя для выполнения настройки начальных значений весов сетей.
3. Применение методов обучения без учителя к выбранному набору сетей.
4. Сбор результатов экспериментов.

# Решаемая задача

Была выбрана задача классификации дорожных знаков. Количество классов - 43. Датасет: <http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>

**Архив обучающего набора имеет следующую структуру:**

* Один каталог на класс
* Каждый каталог содержит один CSV-файл с аннотациями («GT- <ClassID> .csv») и обучающими изображениями.
* Обучающие изображения сгруппированы по наборам
* Каждый набор содержит 30 изображений одного дорожного знака

**Формат изображения**

* Каждое изображение содержит один дорожный знак
* Изображения содержат границу в 10% вокруг фактического дорожного знака (не менее 5 пикселей), чтобы обеспечить подходы по краям
* Изображения хранятся в формате PPM (Portable Pixmap, P6)
* Размеры изображения варьируются от 15x15 до 250x250 пикселей.
* Изображения не обязательно имеют квадратную форму
* Дорожный знак необязательно находится по центру изображения.
* Ограничительная рамка дорожного знака является частью описания.

Описания предоставляются в файлах CSV. Поля разделены знаком ";" (точка с запятой). Описания содержат следующую информацию:

* Имя файла: Имя файла соответствующего изображения
* Ширина: ширина изображения
* Высота: высота изображения
* ROI.x1: X-координата верхнего левого угла ограничительной рамки дорожного знака
* ROI.y1: Y-координата верхнего левого угла ограничительной рамки дорожного знака
* ROI.x2: X-координата нижнего правого угла ограничительной рамки дорожного знака
* ROI.y2: Y-координата нижнего правого угла ограничительной рамки дорожного знака
* ClassId: номер класса дорожного знака

**Предобработка данных**

* Автоматическое выравнивание яркости (histogram equalization).
* Обрезка по центру.
* Перевод в черно-белое изображение.
* Увеличение до заданного размера (48х48).

# Метрика качества решения задачи

В качестве метрики точности решения используется отношение правильно классифицированных знаков ко всем знакам в тестовой выборке:

# Тренировочные и тестовые наборы данных

39203 изображений различных знаков для тренировочных данных.

12630 изображений используется при финальном тестировании модели.

# Разработанные программы/скрипты

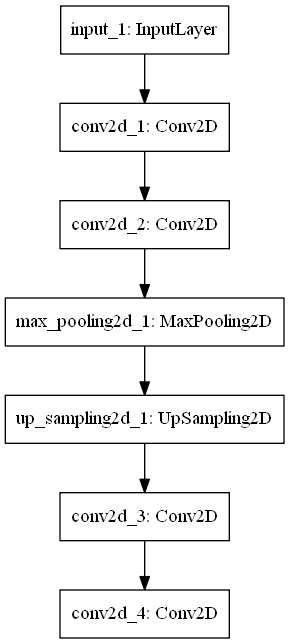
В репозитории в папке lab4 находятся все три сконфигурированные модели:

* autoencoder\_1.py
* autoencoder\_2.py
* autoencoder\_3.py

Скрипты генерируют модели и запускают их обучение и тестирование.

Подготовка файлов датасета к работе подробно описана в README репозитория.

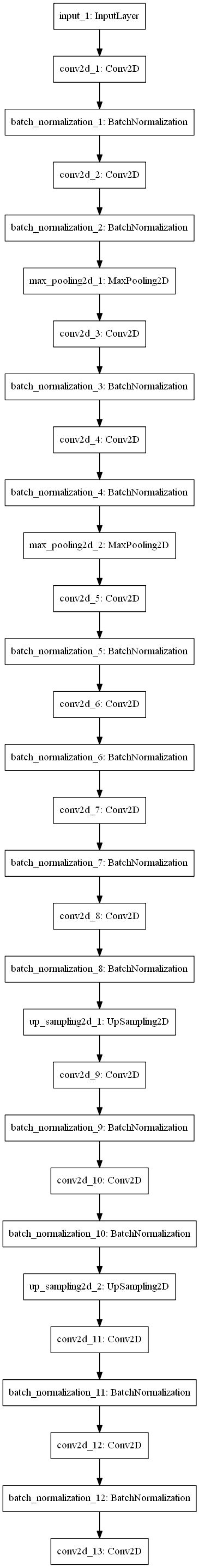
# Тестовые конфигурации сетей



# Авто кодировщик 1.

# C:\Users\Ivan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\model.png

# Авто кодировщик 2.



# Авто кодировщик 3.

# Результаты экспериментов

Параметры обучения автокодировщика:

* Функция ошибки='mean\_squared\_error ',
* Оптимизационный алгоритм =' RMSprop() ',
* batch\_size=64,
* Количество эпох – 30,

Параметры обучения:

* Функция ошибки='categorical\_crossentropy',
* Оптимизационный алгоритм ='adam',
* batch\_size=128,
* Количество эпох – 20,
* Скорость обучения – 0.001.

**Параметры PC:**

OS: Windows 10

CPU: Intel Core i5-3470 3.2Ghz (4CPUs)

RAM: 16384Mb

, DDR3

GPU: Nvidia GeForce GTX 970 1050 MHz, 4058Mb, 256-bit GDDR5

Cuda: 9.0.176

Python: 3.5.4

Keras: 2.2.4

Tensorflow-gpu: 1.12.0

|  |  |  |  |
| --- | --- | --- | --- |
| Автокодировщик | Конфигурация | Точность на тестовом множестве | Время, с |
| Сверточный | Непредобученная | 0.0 | 0,0 |
| Предобученная | 0.0 | 0.0 |
| Полносвязанный | Непредобученная | 0.0 | 0.0 |
| Предобученная | 0.0 | 0.0 |

Результаты обучения с использованием автокодировщика

# Анализ результатов