

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: Прикладная математика и информатика**

**Магистерская программа: Компьютерные науки и приложения**

## **ОТЧЕТ**

по лабораторной работе №4

Тема:

**«Начальная настройка весов полностью связанных нейронных  
сетей»**

**Выполнили:** студенты группы 381803-4м  
Котова О.А.

---

Подпись  
Лицов А.

---

Подпись  
Синицкая О.

---

Подпись

**Преподаватель:** доцент, к.т.н. Кустикова  
В.Д.

---

Подпись

Нижегород  
2019

# Оглавление

1. Постановка задачи .....	<a href="#"><u>3</u></a>
2. Тренировочные и тестовые наборы данных .....	<a href="#"><u>4</u></a>
3. Метрика качества решения.....	<a href="#"><u>5</u></a>
4. Разработанные программы .....	<a href="#"><u>5</u></a>
5. Формат данных, предоставляющийся на вход сети .....	<a href="#"><u>6</u></a>
6. Описание экспериментов и конфигурации.....	<a href="#"><u>7</u></a>
7. Результаты экспериментов .....	<a href="#"><u>10</u></a>
8. Выводы .....	<a href="#"><u>11</u></a>

# **Постановка задачи**

## **Цели**

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 практической 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.

## Тренировочные и тестовые наборы данных

Выбранная задача - Intel Image Classification:

<https://www.kaggle.com/puneet6060/intel-image-classification>.

Исходные данные хранятся в директориях seg\_pred, seg\_test, seg\_train в формате jpg и размера 150x150.

- seg\_pred содержит 7301 изображений
- seg\_test - 3000 изображений, которые распределены по папкам
  - buildings
  - forest
  - glacier
  - mountain
  - sea
  - street
- seg\_train - 14034 изображений, которые распределены по папкам
  - buildings
  - forest
  - glacier
  - mountain
  - sea
  - street

Данные содержат около 25 тыс. цветных изображений размером 150x150, распределенных по 6 категориям: здания, лес, ледник, гора, море, улица. Изображения хранятся в формате jpg.



Тренировочная выборка содержит 14034 изображений.

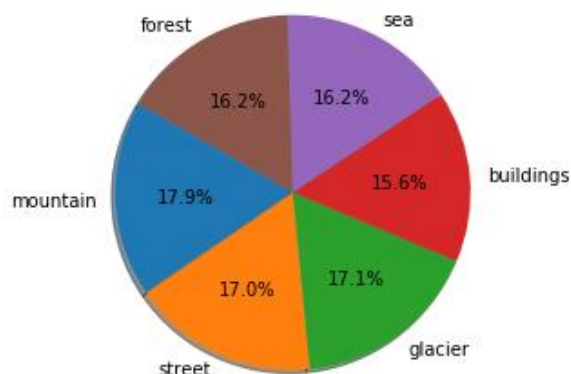
Тестовая выборка содержит 3000 изображений.

Размер каждого изображения 150x150.

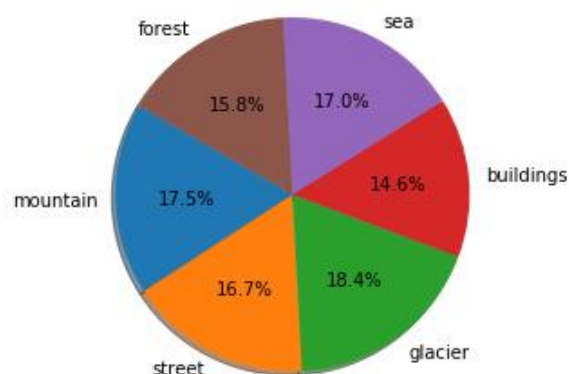
№	Категории	Размер тренировочной выборки	Размер тестовой выборки
1	mountain	2512	525

2	street	2382	501
3	glasier	2404	553
4	buildings	2191	437
5	sea	2274	510
6	forest	2271	474

Процентное соотношение категорий. Тренировочная выборка:



Процентное соотношение категорий. Тестовая выборка:



## Метрика качества решения

Для оценки качества решения задачи выбрана метрика "Точность" ("Accuracy"). Она вычисляет, как часто прогнозы соответствуют меткам. Иными словами, частота с которой  $y_{pred}$  совпадает с  $y_{true}$ .

$$accuracy(y_{pred}, y_{true}) = \frac{1}{N} \sum_{i=1}^N 1(y_{pred_i} == y_{true_i})$$

## Разработанные программы

Lab4.ipynb – скрипт для обучения нейронных сетей.

## **Формат данных, предоставляющийся на вход сети**

С помощью класса ImageDataGenerator и его метода `flow_from_directory()` генерируем пакеты. Данные возвращаются в формате (x, y), где x, y - numpy массивы.

Форма x: (batch\_size, 150, 150, 3).

Форма y: (batch\_size, 6).

Методу `fit_generator` подается на вход генератор данных в формате (x, y). Сети подается на вход массив numpy формата (150, 150, 3), который "сглаживается" сетью с помощью метода `Flatten()`.

## Описание экспериментов и конфигурации

### Эксперимент 1

Простой автокодировщик.

#### Конфигурация 1:

Input (150 * 150 RGB image)
Flatten
FC 1000, relu
FC 512, relu
FC 256, relu
FC 128, relu
FC 64, relu
FC 32, relu
FC 16, relu
FC 6, softmax

### Эксперимент 2

Глубокий автокодировщик.

#### Конфигурация 2:

Input (150 * 150 RGB image)
Flatten
FC 1024, relu
FC 512, relu
FC 256, relu
FC 128, relu
FC 64, relu
FC 32, relu

FC 16, relu
FC 6, softmax

### Эксперимент 3

Сверточный автокодировщик.

#### Конфигурация 3:

Input (150 * 150 RGB image)
Conv (3,3), 32, relu
Maxpooling(2,2), 1
Conv (3,3), 16, relu
Maxpooling(2,2), 1
Conv (3,3), 8, relu
Maxpooling(2,2), 1
Flatten
FC 1024, relu
FC 512, relu
FC 256, relu
FC 128, relu
FC 64, relu
FC 32, relu
FC 16, relu
FC 6, softmax



## Результаты экспериментов

В таблице приведены конфигурация системы и программное обеспечение, с помощью которых проводилось обучение и тестирование построенных моделей.

<i>Параметры</i>	<i>Версия</i>
GPU	Tesla P100, having 3584 CUDA cores, 16GB(16.28GB Usable) GDDR6 VRAM Tesla P100 Spec Sheet
Python	3.7.5
TensorFlow	2.0.0

Параметры обучения:

Количество эпох	20
Размер пачки	128

# Результаты экспериментов:

Модель	1 Простой автокодировщик	1 Без автокодировщика	2 Глубокий	2 Без авт	3 Сверточный	3 Без авт
Количество скрытых нейронов	512 256 128 64 32 16 6	512 256 128 64 32 16 6	128 64 32 16 6	128 64 32 16 6	1024 512 256 128 64 32 16 6	1024 512 256 128 64 32 16 6
Батч	128	128	128	128	128	128
Количество эпох	20	20	20	20	20	20
Общее время (сек)	507	165	505	163	946	181
Точность (Ассигасу) на тренировочном наборе, %	67.27	60.49	60.16	61.22	<b>97.86</b>	<b>97.56</b>
Ошибка на тренировочном наборе	0.87	1.04	1.05	1.03	0.09	0.11
Точность (Ассигасу) на тестовом наборе, %	50.33	56.37	48.13	51.73	<b>72.83</b>	<b>74.97</b>
Ошибка на тестовом наборе	1.30	1.24	1.35	1.41	1.53	1.62

## **Выводы**

В ходе выполнения лабораторной работы была получена модель 3, которая позволяет решать выбранную практическую задачу с достаточно высокими показателями качества. Была спроектирована и разработана программная реализация, позволяющая обучать различные конфигурации нейронных сетей. С помощью полученной реализации были произведены эксперименты на выбранном наборе данных. Во время экспериментов была измерена ошибка классификации. Полученные результаты отражены в настоящем отчете.