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

Нижний Новгород

2019

Оглавление

[1. Постановка задачи](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187372) 3

[2. Тренировочные и тестовые наборы данных](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187373) 4

[3. Метрика качества решения](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187374) 5

[4. Разработанные программы](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187374) 5

[5. Формат данных, предоставляющийся на вход сети](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375) 6

[6. Описание экспериментов и конфигурации](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375) 7

[7. Результаты экспериментов](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375) 10

[8. Выводы](file:///C:\Users\user\Downloads\Kursovaya.docx#_Toc420187375) 11

**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.

[](https://github.com/a-litsov/deep-learning/blob/master/lab2/img/accuracy.png)

**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.

|  |  |
| --- | --- |
| *Параметры* | *Версия* |
| 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 (Accuracy) на тренировочном наборе, % | 67.27 | 60.49 | 60.16 | 61.22 | **97.86** | **97.56** |
| Ошибка на тренировочном наборе | 0.87 | 1.04 | 1.05 | 1.03 | 0.09 | 0.11 |
| Точность (Accuracy) на тестовом наборе, % | 50.33 | 56.37 | 48.13 | 51.73 | **72.83** | **74.97** |
| 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.