

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 №2

Тема:

«Разработка 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..... | 3 |
| 2. Тренировочные и тестовые наборы данных | 4 |
| 3. Метрика качества решения..... | 5 |
| 4. Разработанные программы | 5 |
| 5. Тестовые конфигурации сетей | 5 |
| 6. Результаты экспериментов | 9 |
| 7. Анализ результатов | 11 |

Постановка задачи

Цели

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.

Тренировочные и тестовые наборы данных

Выбранная задача - Intel Image Classification:

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

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



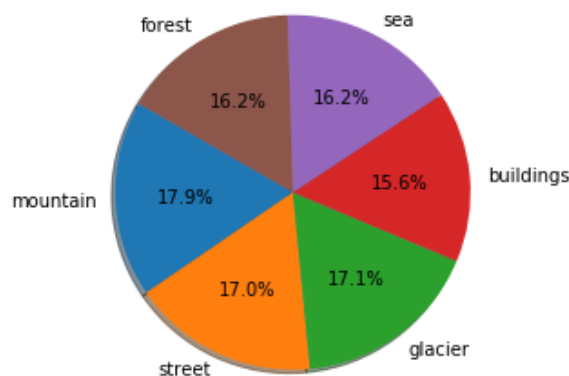
Тренировочная выборка содержит 14034 изображений.

Тестовая выборка содержит 3000 изображений.

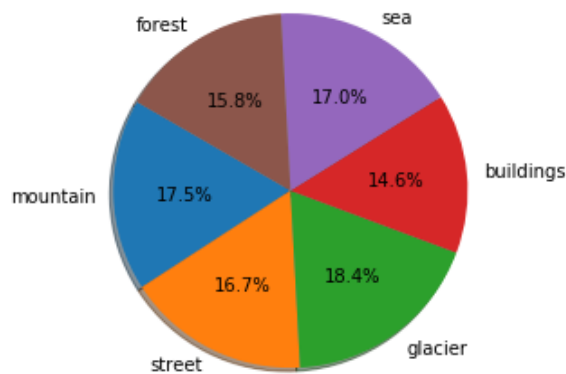
Размер каждого изображения 150x150.

| № | 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

Lab2.ipynb – скрипт для обучения полносвязных нейронных сетей.

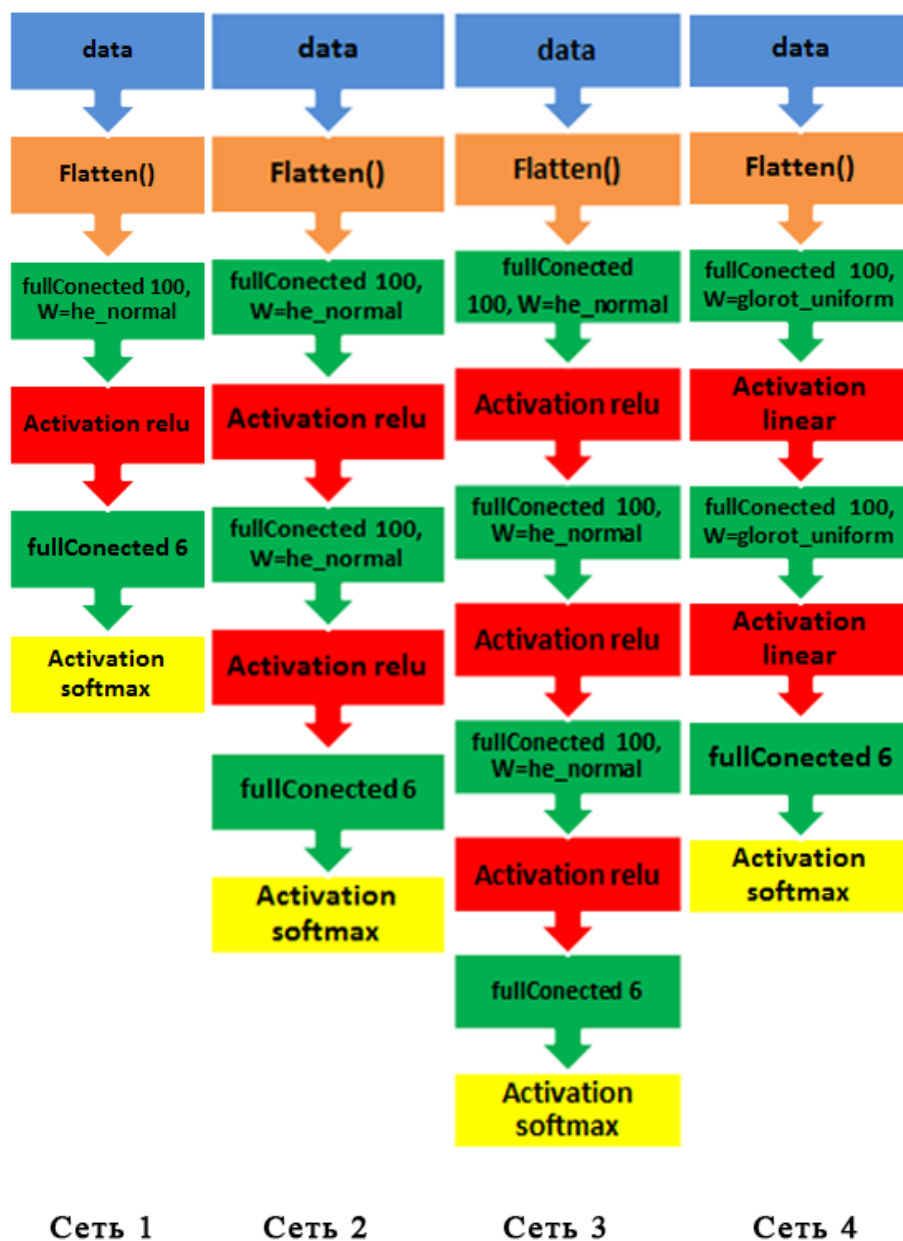
Тестовые конфигурации сетей

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()`.



- Сеть 1: 1 скрытый слой ReLU
- Сеть 2: 2 скрытых слоя ReLU
- Сеть 3: 3 скрытых слоя ReLU
- Сеть 4: 2 скрытых слоя Linear
- Сеть 5: 2 скрытых слоя tanh
- Сеть 6: 3 скрытых слоя tanh
- Сеть 7: 6 скрытых слоя tanh
- Сеть 8: 6 скрытых слоя tanh



Сеть 5

Сеть 6

Сеть 7

Сеть 8

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> |
|----------------------|---|
| Операционная система | Windows 10 |
| GPU | NVIDIA GeForce GTX 750 Ti; Intel Core i5-6400 CPU @ 2.70 GHz |
| Python | 3.7.5 |
| TensorFlow | 2.0.0 |

Параметры обучения:

| | |
|-------------------|-------|
| Скорость обучения | 0.001 |
| Количество эпох | 10/15 |
| Размер пачки | 128 |

Результаты экспериментов:

| Номер сети | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--|-----------|-----------|-----------|----------------|--------------|----------------|-----------|-------------------------------------|
| Батч | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| Количество эпох | 10 | 10 | 10 | 15 | 15 | 15 | 15 | 15 |
| Количество скрытых нейронов | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 512 256 128 64 32 16 |
| Количество скрытых слоев | 1 | 2 | 3 | 2 | 2 | 3 | 6 | 6 |
| Скорость обучения | 0.001 | 0.001 | 0.001 | 0.1 | 0.001 | 0.001 | 0.001 | 0.001 |
| Функция активации | relu | relu | relu | linear | tanh | tanh | tanh | tanh |
| Инициализация весов | he_normal | he_normal | he_normal | glorot_uniform | he_normal | glorot_uniform | he_normal | he_normal |
| Общее время | 06:48 | 06:45 | 06:43 | 10:47 | 10:10 | 11:35 | 10:44 | 38:12 |
| Точность (Ассигасу) на тренировочном наборе, % | 16.97 | 17.90 | 16.97 | 16.97 | 42.67 | 41.90 | 37.12 | 36.09 |
| Ошибка на тренировочном наборе | 13.38 | 13.23 | 13.38 | 13.38 | 1.530 | 1.506 | 1.567 | 1.570 |
| Точность (Ассигасу) на тестовом наборе, % | 16.7 | 17.50 | 16.70 | 16.70 | 43.73 | 43.83 | 40.97 | 36.80 |

| | | | | | | | | |
|---------------------------------|-------|--------|--------|--------|-------|-------|-------|-------|
| Ошибка на тестовом наборе | 13.45 | 13.289 | 13.415 | 13.449 | 1.556 | 1.515 | 1.578 | 1.627 |
|---------------------------------|-------|--------|--------|--------|-------|-------|-------|-------|

Анализ результатов

1. Для текущей задачи не является оптимальным использование полностью связанных нейронных сетей. Сверточные или другие нейронные сети обеспечат лучшие результаты
2. Небольшое количество изображений на каждую категорию