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

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.

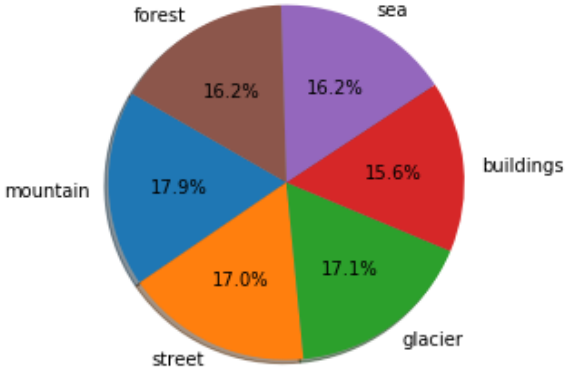
Тестовая выборка содержит 3000 изображений.

Размер каждого изображения 150x150.

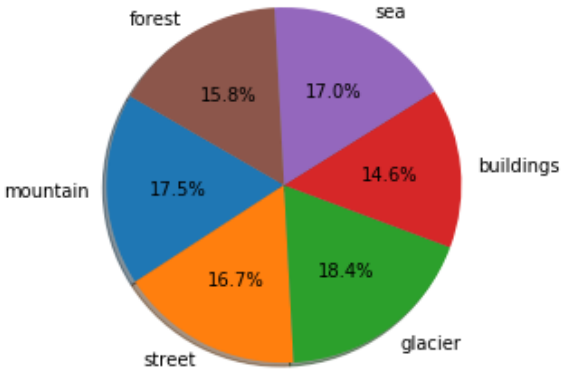
№	*Категории*	*Размер тренировочной выборки*	*Размер тестовой выборки*
1	mountain	2512	525

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.

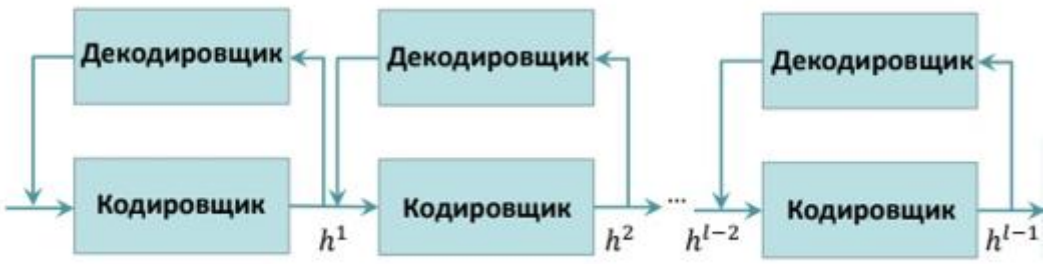
Форма x: (batch_size, 150, 150, 3).
Форма y: (batch_size, 6).

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().

Работа автокодировщика

В качестве алгоритма без учителя для инициализации начальных весов сети будем использовать автокодировщик.

Чаще всего автокодировщики применяют каскадно для обучения глубоких (многослойных) сетей. Автокодировщики применяют для предварительного обучения глубокой сети без учителя. Для этого слои обучаются друг за другом, начиная с первых. К каждому новому необученному слою на время обучения подключается дополнительный выходной слой, дополняющий сеть до архитектуры автокодировщика, после чего на вход сети подается набор данных для обучения. Веса необученного слоя и дополнительного слоя автокодировщика обучаются при помощи метода обратного распространения ошибки. Затем слой автокодировщика отключается и создается новый, соответствующий следующему необученному слою сети. На вход сети снова подается тот же набор данных, обученные первые слои сети остаются без изменений и работают в качестве входных для очередного обучаемого автокодировщика слоя.



Стек автокодировщиков, показанный на рисунке выше, может быть применен при работе с глубокими сетями. Каждый автокодировщик обучается как сеть прямого распространения и работает с одним слоем. Это позволяет постепенно снижать размерность и настраивать параметры.

В задаче семантической сегментации используются **развертоочные нейронные сети.** Состоят из двух блоков: сверточный слой – последовательное применение преобразований свертки, функции активации и пространственного объединения; развертывающий слой – выполнение обратных преобразований расщепления, активации и развертывания.

Описание экспериментов и конфигурации

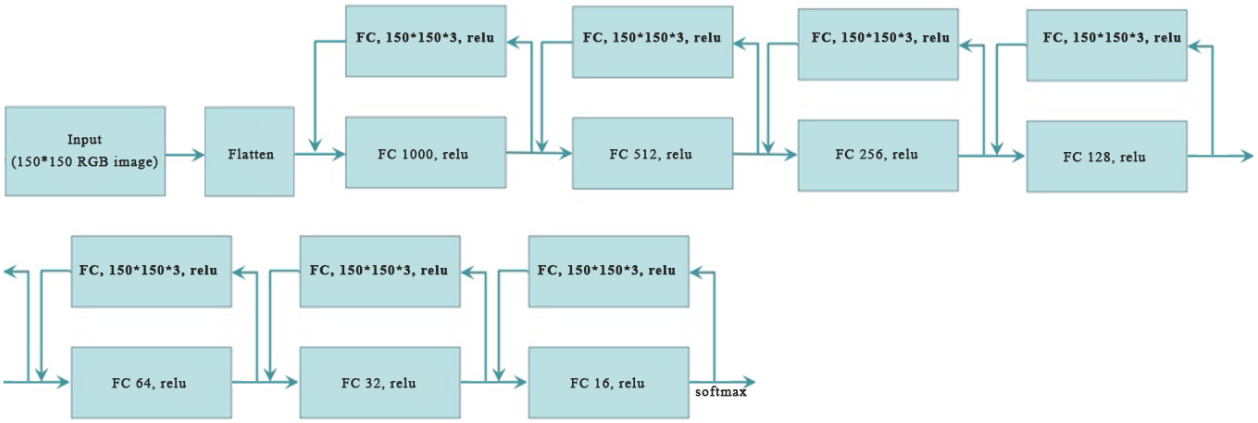
Эксперимент 1

Стек полносвязных автокодировщиков

Исходная конфигурация:

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

С применением автокодировщиков:



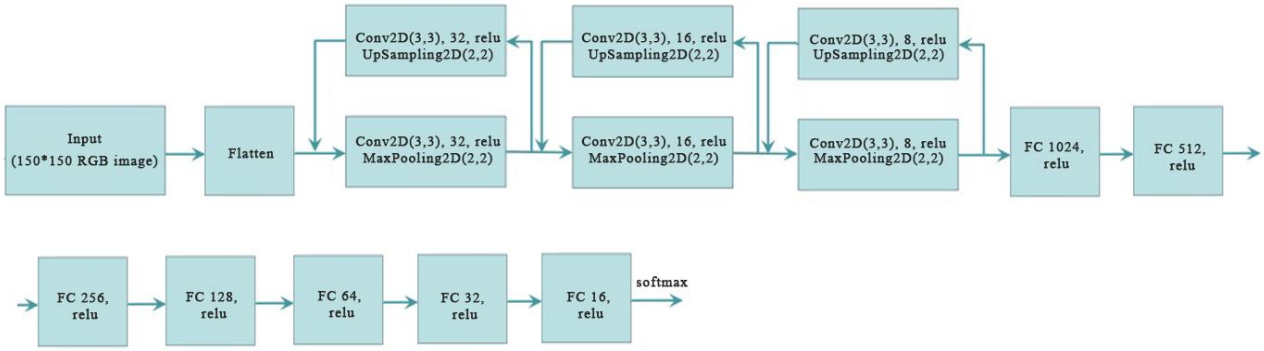
Эксперимент 2

Стек сверточных автокодировщиков

Исходная конфигурация:

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

С применением автокодировщиков:



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 FCNN	1 FCNN без ae	2 CNN	2 CNN без ae
Количество скрытых нейронов	1024 512 256 128 64 32 16 6	1024 512 256 128 64 32 16 6	1024 512 256 128 64 32 16 6	1024 512 256 128 64 32 16 6
Количество фильтров	-	-	32 16 8	32 16 8
Количество свёрточных слоёв	-	-	3	3
Батч	128	128	128	128
Количество эпох	20	20	20	20
Общее время (сек)	505	163	946	181
Время тренировки автокодировщиков (сек)	342	-	765	-
Время тренировки сети (сек)	163	163	181	181
Точность (Accuracy) на тренировочном наборе, %	60.16	61.22	**97.86**	**97.56**
Ошибка на тренировочном наборе	1.05	1.03	0.09	0.11
Точность (Accuracy) на тестовом наборе, %	48.13	51.73	**72.83**	**74.97**
Ошибка на тестовом наборе	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.

В то же время можно сделать вывод, что начальная настройка весов через методы обучения без учителя малоэффективна. С учетом затрат на обучение моделей для инициализации весов эти методы дают слишком маленький прирост в качестве.