

Нижегородский государственный университет им. Н. И. Лобачевского
Институт информационных технологий, математики и механики

Направление подготовки Прикладная математика и информатика
Магистерская программа Вычислительные методы и суперкомпьютерные
технологии

Образовательный курс «Методы глубокого обучения для решения задач
компьютерного зрения»

Отчёт

по лабораторной работе № 5

«Применение переноса обучения глубоких нейронных сетей»

задача

«Классификация персонажей Симпсонов»

Выполнили:

студенты гр. 381603м4

Вершинина О.

Розанов А.

Рой В.

Нижний Новгород
2017

Оглавление

Постановка задачи	3
Формат данных для предоставления нейронной сети	4
Тестовые конфигурации нейронных сетей	5
Результаты экспериментов	6

Постановка задачи

Необходимо исследовать возможность переноса обучения для решения целевой задачи классификации персонажей из мультфильма «Симпсоны».

В ходе работы необходимо решить следующие задачи:

1. Найти исходную задачу, которая близка по смыслу к целевой задаче.
2. Найти натренированную модель для решения исходной задачи.
3. Выполнить три типа экспериментов по переносу знаний:
 - Использование структуры глубокой модели, построенной для решения исходной задачи, с целью обучения аналогичной модели для решения целевой задачи;
 - Использование модели, построенной для решения исходной задачи, в качестве фиксированного метода извлечения признаков при построении модели, решающей целевую задачу;
 - Тонкая настройка параметров модели, построенной для решения исходной задачи, с целью решения целевой задачи.

Формат данных для предоставления нейронной сети

Исходные данные представляют собой набор jpg изображений, различного разрешения. Несколько примеров представлены на рис. 1.

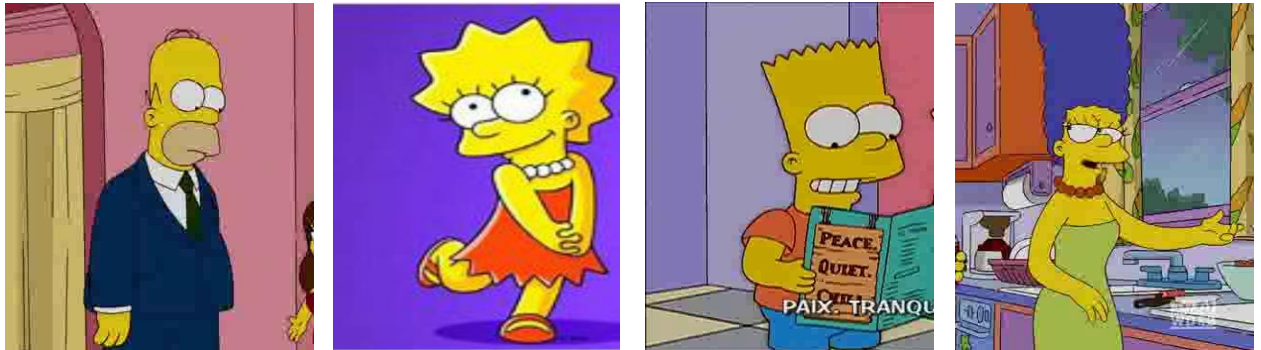


Рис.1. Персонажи из Симпсонов. Слева-направо: Гомер Симпсон, Лиза Симпсон, Барт Симпсон, Мардж Симпсон.

Для предварительной обработки данных использован скриптовый язык Python. Интерпретатор языка входит в дистрибутив Anaconda, который содержит ряд пакетов для анализа данных и машинного обучения.

Библиотека глубокого обучения MXNet может работать с различными типами входных данных, в том числе с однородными многомерными массивами ndarray из пакета NumPy. Для того, чтобы привести входные данные к такому формату, необходимо использовать библиотеку OpenCV для Python (opencv-python).

Был разработан скрипт, в котором с помощью функции cv2.imread считываются трёхканальные .jpg изображения в формате BGR (стандартное цветовое пространство OpenCV) и конвертируются в формат RGB; нормализуются (значение каждого пикселя делится на 255) и масштабируются до размера 28x28. Формируется массив меток от 0 до 17, соответствующих восемнадцати персонажам. Затем данные случайным образом делятся на обучающую и тестовую выборки в отношении 85% к 15%. Полученные ndarray-массивы X_train, X_test, y_train, y_test сохраняются в файлы для последующего использования нейронной сетью.

Тестовые конфигурации нейронных сетей

В качестве исходной задачи была выбрана задача классификации изображений на основе базы данных ImageNet, которая содержит 1000 классов различных изображений. В качестве натренированной модели была выбрана нейронная сеть resnet-18, которая содержит 20 сверточных слоев, 18 слоев с функцией активации relu, 2 слоя с пространственным объединением и 1 полносвязный слой на 1000 нейронов с функцией активации softmax. Подробная архитектура сети представлена в файле lab5/doc/resnet18.jpg. Данная модель на базе данных ImageNet показывает следующую точность:

1. Top-1: 69.57%
2. Top-5: 89.24%

В данной лабораторной работе были произведены следующие 3 эксперимента:

1. Использование модели без изменений, но с полным ее переобучением. Веса инициализируются случайным образом.
2. Замена классификатора в исходной модели. Веса в нем инициализируются случайным образом. Оставшаяся часть модели используется как метод выделения признаков и данная часть модели не переобучается. В качестве нового классификатора был выбран классификатор с одним полносвязным скрытым слоем на 300 нейронов и функцией активации tanh и еще одним полносвязным слоем с функцией активации softmax.
3. Тонкая настройка параметров модели. В данном эксперименте обучается вся нейронная сеть. При этом классификатор заменяется на новый со случайными весами. А оставшаяся часть модели инициализируется весами из натренированной модели. В качестве нового классификатора был выбран классификатор с одним полносвязным слоем на 18 нейронов с функцией активации softmax.

Результаты экспериментов

Эксперименты проводились при следующих параметрах обучения: batch size = 10, optimizer = 'sgd', learning rate = 0.01. Обучение сети проводилось до тех пор, пока точность на тренировочной выборке не становилась равной 1.

Таблица 1. Результаты экспериментов

Эксперимент №	Время обучения модели, с	Точность классификации на тестовой выборке
1	10990	0.8673
2	311	0.4726
3	17584	0.8764

Наилучшую точность показал третий эксперимент, что означает, что обучающих данных хватило для тонкой настройки модели. При этом набор данных целевой задачи сильно отличается от набора данных исходной задачи, поэтому точность при втором эксперименте хуже, чем при первом и третьем, в которых проводилось полное обучение сети.