

Постановка задачи

Цель работы – изучение применимости метода переноса обучения к выбранной (целевой) задаче, где целевая задача – распознавание 101-ого класса на данных **101_ObjectCategories**.

В качестве исходной задачи рассматривалась задача распознавания 1000-и классов с помощью VGG16.

Описание программной реализации

Разработан скрипт реализующий:

1. Загрузку данных с помощью модуля Pillow. Входные изображения хранятся в формате .jpeg и имеют 3 цветовых канала R,G,B. Преобразование входных данных: нормализация ($[0..255] \rightarrow [0..1]$), приведение размера к 128x128, разбиение на тренировочную и тестовую выборки в соотношении 70% к 30%.
2. Загрузку основных модулей Keras и TensorFlow для дальнейшей работы, установка начальных параметров. Описание тестируемых моделей глубоких сверточных сетей.
3. Загрузку стандартной реализации VGG16 (keras.applications.VGG16) с заранее обученными весами.

Тестовые конфигурации сетей

В качестве исходной задачи использовалась сеть – VGG16

(<http://www.cs.toronto.edu/~frossard/post/vgg16/>) с обученными весами для сверточных слоев

(https://github.com/fchollet/deep-learning-models/releases/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5).

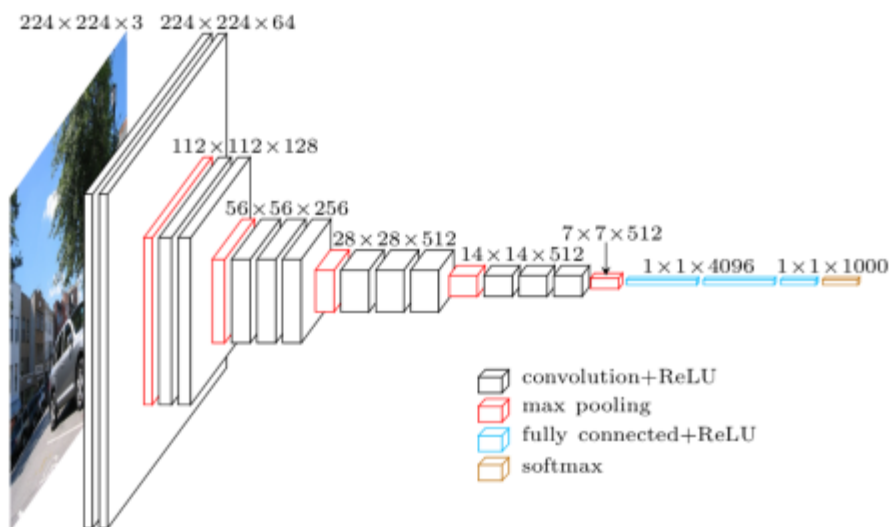
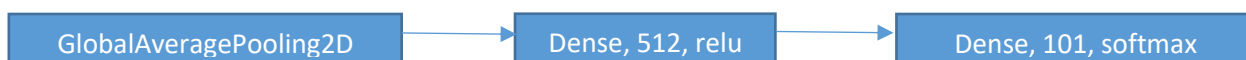


рис. 1. Архитектура VGG16

Для решения целевой задачи исходный классификатор VGG16 заменили на классификатор распознающий 101 класс.



Описание экспериментов:

1. Замена классификатора исходной задачи для распознавания 101-ого класса, обучение нового классификатора на 101_ObjectCategories (веса в сверточных слоях заморожены), тестирование на 101_ObjectCategories
2. Замена классификатора исходной задачи для распознавания 101-ого класса, обучение всей сети на 101_ObjectCategories (веса в сверточных слоях проинициализированы обученными весами из исходной VGG16), тестирование на 101_ObjectCategories
3. Замена классификатора исходной задачи для распознавания 101-ого класса, обучение всей сети на 101_ObjectCategories, при этом веса сверточных слоев инициализируются произвольными значениями, используется только структура исходной VGG16. Тестирование на 101_ObjectCategories

Эксперименты

Тип	Параметры	Общее время выполнения (с)	Качество решения целевой задачи (точность на тестовом наборе %)
Использование модели, построенной для решения исходной задачи, в качестве фиксированного метода извлечения признаков при построении модели, решающей целевую задачу	Learning Rate = 0.01 Batch Size = 32	570	0.771
	Learning Rate = 0.005 Batch Size = 16	636	0.770
	Learning Rate = 0.005 Batch Size = 64	535	0.727
Тонкая настройка параметров модели, построенной для решения исходной задачи, с целью решения целевой	Learning Rate = 0.01 Batch Size = 32	1380	0.8
	Learning Rate = 0.005 Batch Size = 16	1588	0.8
	Learning Rate = 0.005 Batch Size = 64	1285	0.9
Использование структуры глубокой модели, построенной для решения исходной задачи, с целью обучения аналогичной модели для решения целевой задачи	Learning Rate = 0.01 Batch Size = 32	1374	0.559
	Learning Rate = 0.005 Batch Size = 16	1588	0.486
	Learning Rate = 0.005 Batch Size = 64	1277	0.515

Вводы:

Сеть исходной задачи с обученными весами (эксперименты 1, 2) показала лучшие результаты чем та же сеть, обученная на 101_ObjectCategories или более простая сеть обученная на 101_ObjectCategories (см. Лабораторная 3). Это объясняется тем, что загруженные веса обучались на выборке большого размера (значительно больше, чем 101_ObjectCategories).

Наибольшая точность достигается при тонкой настройке параметров.