

Дообучение классификатора

Влад Шахуро, Владимир Гузов



Обзор задания

В данном задании предлагается настроить обученную нейросеть для задачи классификации видов птиц. Для реализации используйте библиотеку [Keras](#).



Описание задания

Дообучение (finetuning) — широко используемый метод обучения нейронной сети на небольших наборах данных. Глубокая сеть, обучаемая с нуля на небольших базах, подвержена переобучению. Чтобы это предотвратить, сеть инициализируется весами, обученными на большом наборе данных (например, ImageNet), а затем донастраивается с помощью градиентного спуска на целевой базе. В случае классификации верхний слой обученной сети заменяется на полносвязный слой с softmax-активацией и количеством выходов, равным количеству классов. Благодаря тому, что обученные на большой базе нижние слои запоминают универсальные, не привязанные к конкретным изображениям или классам, признаки, нейросеть быстрее сходится, а также в меньшей степени подвержена переобучению.

Советы

1. Выставляйте learning rate ниже, чем при обучении с нуля, чтобы избежать сильного искажения признаков на нижних слоях.
2. Для достижения хороших результатов на малом количестве данных и уменьшения времени на обучение можно заморозить несколько нижних слоев, сделав их необучаемыми. В Keras за это отвечает параметр слоя `trainable`.

Интерфейс программы, данные и скрипт для тестирования

Необходимо реализовать две функции: `train_classifier`, обучающую классификатор на основе предобученной нейросети, и `classify`, классифицирующую входные изображения с обученной

моделью. Функция `train_classifier` возвращает готовую модель нейросети, а `classify` — словарь размером N , ключи которого — имена файлов, а значения — числа, означающие метку класса. Здесь N — количество изображений.

Скрипт для тестирования `classification_test` принимает на вход директорию с тренировочной и тестовой выборками, обучает детектор и подсчитывает точность *acc* многоклассовой классификации. Точность *acc* на скрытой выборке конвертируется в итоговый балл:

$acc \geq 0.80$ — 10 баллов

$acc \geq 0.77$ — 9 баллов

$acc \geq 0.75$ — 8 баллов

$acc \geq 0.70$ — 6 баллов

$acc \geq 0.65$ — 5 баллов

$acc \geq 0.60$ — 4 балла

$acc \geq 0.40$ — 2 балла

Для обучения алгоритма выдается публичная выборка размеченных изображений птиц. Программа тестируется на двух тестах. В каждом из тестов нейросеть сначала обучается с флагом `fast_train=True` в функции `train_classifier`. Функция обучения с этим флагом должна работать недолго, не больше 15 минут. Для этого поставьте 1 эпоху обучения и несколько батчей. Обученная модель для тестирования не используется, этот этап необходим только для проверки работоспособности функции обучения. Затем в первом тесте алгоритм тестируется на публичной выборке, во втором тесте — на скрытой выборке. Для тестирования загружается обученная модель `birds_model.hdf5`. Чтобы уложиться в ограничение размера файла с весами модели, дообучайте легковесную предобученную сеть (например, ResNet50). Результаты второго теста и итоговый балл скрыты до окончания срока сдачи задания. Итоговый балл считается по последней посылке с ненулевой точностью. Для уменьшения количества потребляемой памяти в Keras можно при обучении использовать генераторы батчей (функция `model.fit_generator`).

Полезные ресурсы

[Предобученные модели в Keras](#)