

Таким образом, при добавлении контура к символам, объем памяти, занимаемой текстом, увеличивается линейно.

Предложенный и проанализированный метод тайной передачи информации в тексте-контейнере основан на реализации текстовой стеганографии путем изменения такого параметра текста-контейнера, как контур символа.

ЛИТЕРАТУРА

1. Урбанович, П.П. Защита информации методами криптографии, стеганографии и обфускации / П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.

УДК 004.032.26

Н.А. Жилияк, доц., канд. техн. наук;
Джереми Убонг Чарлес, маг.
(БГТУ, г. Минск)

НЕЙРОННЫЕ СЕТИ КАК РЕШЕНИЕ ЗАДАЧИ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

В данном материале рассматривается, как предобученные нейронные сети могут быть использованы для решения задачи классификации изображений.

За последние годы появилось большое количество моделей, созданных и обученных профессионалами с использованием большого количества данных и огромных вычислительных мощностей. Многие из этих моделей находятся в открытом доступе, и любой может использовать их для решения своих задач совершенно бесплатно.

В качестве примера рассмотрим задачу классификации изображений из конкурса LifeCLEF Plant Identification Task. Задача заключается в том, чтобы предсказать таксономический класс растения, основываясь на нескольких его фотографиях [1].

Для обучения доступно 47815 изображений растений, каждое из которых принадлежит к одному из 500 классов. Необходимо построить модель, которая будет возвращать список наиболее вероятных классов растения. Позиция верного класса растения в списке предсказанных классов (ранг) определяет качество системы.

Эта задача моделирует реальный жизненный сценарий, где человек пытается идентифицировать растение, изучая его отдельные части (стебель, лист, цветок и др.). Таким образом, первичный показатель качества определяется как следующая средняя оценка S :

$$S = \frac{1}{U} \sum_{u=1}^U \frac{1}{\rho_u} \sum_{\rho=1}^{\rho_u} s_{u,\rho},$$

где U – количество пользователей, которые имеют хотя бы одно фото в тестовой выборке, ρ_u – число уникальных растений, которые сфотографировал пользователь U , $s_{u,\rho}$ – значение от 0 до 1, рассчитывается как обратное число от ранга верного класса растения в списке наиболее вероятных классов.

Для решения задачи будем использовать нейронные сети, которые обучались на 1,2 миллионах изображений из базы данных ImageNet. Изображения содержат объекты, относящиеся к 1000 разных классов, такие как компьютер, стол, кот, собака и другие объекты, с которыми мы часто встречаемся в повседневной жизни.

В качестве базовых архитектур мы выбрали VGG16, VGG19, ResNet50 и InceptionV3. Эти сети были натренированы на огромном количестве изображений и уже умеют распознавать простейшие объекты, поэтому можно надеяться, что они помогут нам создать достойную модель для классификации растений.

Препроцессинг изображений – это предварительная обработка изображений. Основной целью предобработки, в нашем случае, является выявление наиболее важной части изображения и удалении ненужного шума [2].

Все изображения в обучающей сети можно разделить на категории в зависимости от части растения, изображенного на них: Entire (растение целиком), Branch (ветвь), Flower (цветок), Fruit (фрукт), LeafScan (скан листа), Leaf (лист), Stem (стебель). Для каждой из этих категорий был подобран свой наиболее подходящий метод предварительной обработки.

Мы не будем изменять Entire и Branch изображения, поскольку зачастую большая часть изображения содержит полезную информацию, которую мы не хотим потерять.

Модель будем строить при помощи Keras с TensorFlow в качестве бэк-энда. Keras – мощная библиотека машинного обучения, предназначенная для работы с нейронными сетями, которая позволяет строить всевозможные модели: от простых, таких как перцептрон, до очень сложных сетей, предназначенных для обработки видео. И что очень важно в нашем случае, Keras позволяет использовать предобученные нейронные сети и оптимизировать модели как с помощью CPU, так и GPU [4].

ШАГ 1. Сперва мы загружаем предобученную модель без полносвязных слоев и к ее выходу применяем операцию пулинга

(pooling). В нашем случае лучшие результаты показал “средний” пулинг (GlobalAveragePooling), его и возьмем для построения модели.

Затем прогоняем изображения из тренировочного сета через полученную сеть, а полученные признаки сохраняем в файл. Немного позже вы увидите, зачем это нужно.

ШАГ 2. Мы могли бы заморозить все слои предобученной сети, добавить поверх нее свою полносвязную сеть, а затем обучать полученную модель. Чтобы сэкономить время, используем те признаки, которые мы сохранили на предыдущем шаге, для того, чтобы на них обучить полносвязную сеть.

Рассмотрим архитектуру полносвязной сети, которую будем обучать. После ряда экспериментов было выяснено, что одна из лучших архитектур имеет следующую структуру:

- три плотных слоя по 512 нейронов. За каждым dense-слоем идет Dropout-слой, с параметром 0.5. Это значит, что в каждом слое на каждом проходе сети мы случайным образом выбрасываем сигналы примерно половины нейронов;

- выходной слой представляет собой softmax на 500 классов;
- в качестве функции потерь мы используем categorical cross-entropy, а оптимизируем сеть при помощи Adam;

- также было замечено, что использование функции selu (scaled exponential unit) вместо relu в качестве функции активации помогает сети сходиться быстрее.

ШАГ 3. На этом шаге добавляем обученную полносвязную сеть поверх предобученной модели. Функцию потерь оставляем без изменений, а для обучения сети будем использовать другой оптимизатор.

Предобученная нейронная сеть уже выучила много абстрактных и общих признаков, и чтобы не сбить найденные веса, мы будем тренировать сеть с очень маленькой скоростью обучения. Такие оптимизаторы как Adam и RMSProp сами подбирают скорость обучения, в нашем случае выбранная скорость может оказаться слишком высокой, поэтому они нам не подходят. Чтобы иметь возможность самим задавать скорость обучения, мы будем использовать классический SGD-оптимизатор.

Чтобы улучшить качество итогового классификатора, нужно помнить следующее:

- уменьшайте скорость обучения на плато, чтобы слишком сильно не уйти в сторону минимума (*ReduceLROnPlateau callback*);

- если на протяжении нескольких эпох ошибка на валидационных данных не уменьшается, то стоит прекратить обучение (*EarlyStopping callback*);

– обычно дообучение моделей занимает много времени и когда мы закрываем *.ipynb*-файлы, весь динамический вывод теряется. Я рекомендую сохранять информацию об обучении в файл (*CSVLogger callback*), чтобы в дальнейшем можно было проанализировать, как проходит обучение модели.

Аугментация данных. Поскольку на заключительном шаге мы обучаем сеть целиком, здесь мы можем использовать аугментацию. Но вместо стандартного *ImageDataGenerator* из *Keras* мы будем использовать *Imgaug* – библиотеку, которая предназначена для аугментирования изображений. Важной особенностью *Imgaug* является то, что можно явно указать, с какой вероятностью преобразование должно быть применено к изображению. Кроме того, в этой библиотеке есть возможность объединять преобразования в группы, и выбирать, какую из групп применить [5].

Для аугментации мы выбираем те преобразования, которые могут происходить в реальной жизни, например, зеркальное отражение фото (по горизонтали), повороты, увеличение, зашумление, изменение яркости и контрастности. Если вы хотите использовать большое количество преобразований, то очень важно не применять их одновременно, поскольку для сети будет очень сложно извлечь полезную информацию из фото.

В метаданных к каждому изображению есть оценка качества (средняя оценка пользователей, показывающая, насколько хорошо изображение подходит для классификации). Мы предположили, что изображения с оценкой 1 и 2 достаточно шумные, и хотя они могут содержать полезную информацию, в итоге они могут негативно повлиять на качество классификатора. Эту гипотезу мы проверили при обучении *InceptionV3*. Изображений с оценкой 1 в тренировочном сете оказалось совсем немного, всего 1966, поэтому мы решили не использовать их при обучении. В результате, сеть обучалась лучше на изображениях с рейтингом выше единицы.

Выводы. Дообучение нейронных сетей, которые были натренированы на более чем 1 миллионе изображений, позволило значительно улучшить решение. Наш подход показал, что предобученные модели могут значительно улучшить качество в задачах классификации изображений, особенно в ситуациях, когда отсутствует достаточное количество данных для обучения. Даже если ваша базовая модель не имеет ничего общего с решаемой задачей, она все равно может оказаться полезной, поскольку уже умеет узнавать самые простые объекты окружающего мира.

ЛИТЕРАТУРА

1. LifeCLEF – Multimedia Retrieval in CLEF. – Режим доступа: <https://www.imageclef.org/>.
2. Предварительная обработка изображений. – Режим доступа: <https://www.machinelearningmastery.ru/image-pre-processing-c1aec0be3edf/>.
3. Active Contour Model – Режим доступа: https://scikit-image.org/docs/dev/auto_examples/edges/plot_active_contours.html.
4. Keras – библиотека машинного обучения. – Режим доступа: <https://keras.io/>.
5. Imgaug – библиотека для аргументирования изображений. – Режим доступа: <https://github.com/aleju/imgaug>.

УДК004. (630.65)

В.В. Побединский^{1,2}, проф., д-р техн. наук;
С.В. Ляхов¹, доц., канд. техн. наук; М.Н. Салихова², асп.;
А.С. Некрасов¹, магистрант
¹(УГЛТУ, Екатеринбург), ²(УрГАУ, Екатеринбург)

МОДЕЛИРОВАНИЕ ПРОЦЕССА ТО И Р ПАРКА ЛЕСОЗАГОТОВИТЕЛЬНОЙ ТЕХНИКИ ПЕРЕДВИЖНЫМИ СРЕДСТВАМИ

Аннотация. В статье рассмотрены вопросы технической эксплуатации лесозаготовительной техники передвижными средствами в производственных условиях лесозаготовок на мастерских участках. Для исследования процессов ТО и Р использовано в системе Matlab приложение SimEvents для визуально-блочного событийного моделирования. Предложена имитационная модель и ее программная реализация процесса распределения трудозатрат на ТО, проводимого на базе машины СРПМ-3.

Затраты на техническую эксплуатацию парка лесозаготовительной техники достигают около 30% от общепроизводственных, поэтому проблема совершенствования системы технического обслуживания техники остается актуальной. Специфика такой техники в том, что она работает на удаленных расстояниях от основной производственной базы, где находятся ПЦТО или РММ, поэтому становится неэффективным транспортирование ЛЗМ с лесосеки на ремонты, например, I, II группы сложности или на ТО-1. Однако, чтобы обосновать экономически целесообразные расстояния транспортирования машин до ПЦТО, следует в каждом конкретном случае делать соответствующие исследовательские работы [1]. Самым достоверным методом исследований подобных процессов можно считать имитационное моде-