УДК 004.89

Применение свёрточных нейронных сетей для задач классификации фруктов на изображении

И. В. Гущин, А. Е. Споров, А. С. Тапузов
Харьковский национальный университет имени В.Н. Каразина, Украина

В статье предложен способ решения специализированной задачи распознавания образов - задачи классификации фруктов на изображениях с использованием многослойной свёрточной нейронной сети (CNN). Рассмотрены общие сведения о механизме работы CNN. Приводится описание выбранной для решения поставленной задачи нейронной сети ResNet. Описан способ создания программной системы на языке Python, способной выполнять задачу классификации по 30 классам изображений с фруктами и позволяющей выполнять задачу последующей маркировки изображений, содержащих несколько объектов. Приводятся выводы о применимости нейросети ResNet для классификации искомого набора данных, показаны метрики точности выбранной архитектуры.

Ключевые слова: *компьютерное зрение, машинное обучение, свёрточные нейронные сети, задача классификации.*

У статті запропоновано спосіб вирішення спеціалізованого завдання розпізнавання образів - завдання класифікації фруктів на зображеннях з використанням багатошарової згорткової нейронної мережі (CNN). Розглянуто загальні відомості про механізм роботи CNN. Наводиться опис обраної для вирішення поставленого завдання нейронної мережі ResNet. Описано спосіб створення програмної системи на мові Python, здатної виконувати завдання класифікації по 30 класам зображень з фруктами і дозволяє виконувати завдання подальшого маркування зображень, що містять кілька об'єктів. Наводяться висновки про можливість застосування нейромережі ResNet для класифікації шуканого набору даних, показані метрики точності обраної архітектури.

Ключові слова: *комп'ютерний зір, машинне навчання, згорткові нейронні мережі, задача класифікації.*

The method for solving a specialized problem of pattern recognition - the problem of classifying fruits on images by using a multilayered convolutional neural network (CNN) has been proposed in the article. General information about the mechanism of CNN work has been considered. The description of the neural network ResNet chosen for the task solution is given. The presented method for creating a software system on Python that can perform the task for classifying 30 classes of images with fruits allows performing the task of subsequent marking of images containing several objects. The conclusions about the applicability of the ResNet neural network for the classification of the required data set are presented. The accuracy metrics of the selected architecture are shown.

Key words: *computer vision, machine learning, convolutional neural networks, classification task.*

1. Введение

В последние годы в мире информационных технологий большое количество задач решается различными методами машинного обучения, в частности, с помощью нейронных сетей. Одним из самых актуальных на сегодняшний день направлением является компьютерное зрение, которое позволяет решать задачи, связанные с выделением и классификацией объектов на изображениях и видео Например, на данный момент технологии компьютерного зрения уже помогают

медикам распознавать раковые клетки на медицинских снимках. Серьезным стимулом к развитию исследований в данном направлении является стремительное повышение интеллектуальности портативной техники и современных интеллектуальных автомобилей. Именно алгоритмы компьютерного зрения позволяет мобильным устройствам выполнять свои многочисленные функции (напр., фотокамеры могут распознавать лица, выполнять размытие фона и другие функции, необходимые пользователям устройства). Современные автомобили, в свою очередь, используют данную технологию для ориентирования в пространстве, распознавания знаков, пешеходов и других препятствий.

Для решения таких задач используются нейронные сети специальной архитектуры - свёрточная нейронная сеть (англ. Convolutional Neural Network). Данный тип нейронных сетей специально предназначен для решения задач классификации изображений. На протяжении нескольких последних лет нейронные сети этого типа показывает наилучшие результаты на практике. В частности, на ежегодно проводящемся конкурсе по компьютерному зрению ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [1] лучшие результаты показывали такие архитектуры свёрточных сетей, как AlexNet, VGG, GoogleNet, ResNet и др. Данные типы свёрточных нейронных сетей были натренированы на наборе данных (датасете) ImageNet [2] с изображениями, принадлежащими к 1000 классам разного характера. Однако, для решения специализированных задач, например, такой как классификация фруктов на изображении, готовые натренированные нейронные сети не могут реализовать классификацию на достаточном уровне точности по двум основным причинам:

1. Архитектура нейронной сети не содержит последнего слоя с необходимым набором классов;
2. Нейронная сеть была натренирована на сильно обобщенном датасете. Например, из 1000 классов датасета ImageNet, лишь несколько являются фруктами, поэтому для решения специализированной задачи имеющихся весов будет недостаточно.

Целью данной работы является создание классификатора изображений с применением свёрточной нейронной сети подходящей архитектуры для специализированной задачи классификации изображений, а также выбор метрик точности.

2. Архитектура LeNet и общая структура свёрточной нейронной сети

В 1998 году ведущим исследователем свёрточных нейронных сетей стал Yann LeCun. После многочисленных успешных исследований им была опубликована работа [3], в которой описана архитектура одной из первых свёрточных нейронных сетей LeNet [4]. Затем на основе архитектуры сети LeNet было разработано большое количество других архитектур, но все они имеют одну и ту же базовую структуру, аналогичную структуре сети LeNet. Базовая структура простейшей свёрточной нейронной сети приведена на рис.1.

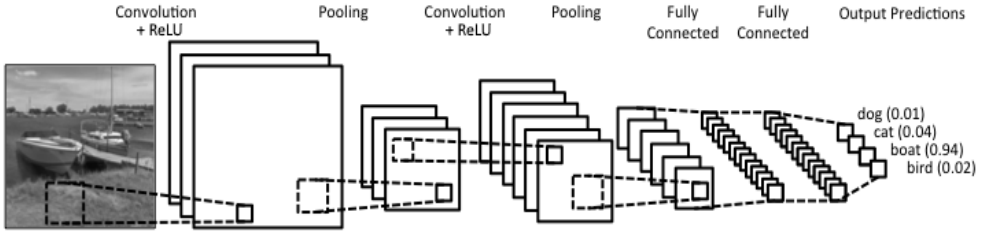

Рис. 1. Базовая архитектура простейшей свёрточной нейронной сети LeNet.

Рассмотрим основные этапы распознавания изображений. На первом, подготовительном этапе, изображение оцифровывается и подготавливается к обработке. Для этого поданное на вход цифровое изображение преобразуется в матрицу W[X, Y, 3], где X и Y - разрешение изображения, а 3 –размерность RGB канала. Полученная таким образом матрица используется в качестве входных данных для свёрточной нейронной сети.

Базовая архитектура свёрточной нейронной сети (см. рис. 1) состоит из четырех основных типов слоев:

1. свёрточный слой (англ., convolution layer);
2. слой линейной ректификации (англ., Rectified Linear Unit, ReLU);
3. слой пулинга или слой субдискретизации (англ., pooling layer);
4. полносвязный слой (англ., fully connected layer).

Рассмотрим подробно структуру и назначение каждого слоя.

2.1. Свёрточный слой

Свёрточный слой (convolution layer) – основа нейронной сети, представлен набором трёхмерных матриц, называемыми фильтрами [5]. С помощью каждого фильтра проводится операция свертки матрицы исходных данных. Модель содержит еще один гиперпараметр P - толщину заполнения нулями (англ., zero-padding). Заполнение нулями применятся к входной матрице для того, чтобы можно было управлять размером матрицы - результата после применения фильтров. В результате свертки получится матрица размером $W^* \times H^* \times D^*$, где

$$W^* = \frac{W-F+2P}{S} + 1$$
$$H^* = \frac{H-F+2P}{S} + 1$$
$$D^* = K.$$

Матрицу, полученную после применения сверточного фильтра, еще называют картой активации (англ., activation map, feature map)) [6]. Данная матрица соответствует найденной особенности на изображении (напр., кривые, углы).

2.2. Слой линейной ректификации

Слой линейной ректификации (слой активации) является дополнительным слоем. Он применяется после каждого свёрточного слоя, содержит функцию активации (Rectified Linear Unit, ReLU), которая для ускорения вычислений обычно определяется таким образом:

$$f(x) = \max(0, x),$$

где x – входное значение для нейрона.

Операция ReLU – поэлементная операция, применяется к каждому пикселю. Она заменяет все отрицательные значения на карте активации нулевыми. Данная операция используется для введения нелинейности в свёрточную нейронную сеть. Это необходимо, поскольку операция свёртки - линейная операция (сложение и перемножение матриц), а большинство данных реального мира, которые используются для обучения свёрточной нейронной сети, являются нелинейными.

Вместо ReLU могут использоваться и другие функции активации, такие как tanh или sigmoid [8]. При больших значениях ошибки сети, для избегания так называемого «умирания» функции ReLU, можно применять Leaky ReLU [7].

2.3. Слой пулинга

Слой пулинга (слой субдискретизации, pooling layer) предназначен для уменьшения размерности каждой карты активации [8] при сохранении наиболее важной информации об изображении. Кроме того, пулинг уменьшает количество параметров нейронной сети и, как следствие количество вычислений в нейронной сети. Таким образом, слой пулига служит для предотвращения проблемы переобучения сети (англ., overfitting). Операции, выполняемые в данном слое, реализуются с применением разных функций: max, avg, sum и т. д.

2.4. Полносвязный слой

Полносвязный слой (fully connected layer) – традиционный слой многослойных нейронных сетей, который использует функцию активацию softmax [19] в выходном слое. Однако, в ряде случаев могут использоваться и другие функции активации, например, SVM [19]. Данный слой является полносвязным. Это значит, что каждый нейрон в предыдущем слое соединен с каждым нейроном в следующем. В нейронной сети последний слой этого типа формирует высокоуровневые особенности (англ., features) входного изображения. Эти сформированные особенности и используются для классификации входного изображения на основании тренировочного датасета. Помимо решения задачи классификации, добавление полносвязного слоя уменьшает вычислительные затраты, необходимые для обучения сети. Заметим, что сумма вероятностей, которые получаются в результате вычислений в полносвязном слое, должна составлять 1 (см. рис. 1).

3. Выбор архитектуры сети для решения задачи классификации фруктов

Для решения поставленной специализированной задачи классификации фруктов на изображении была выбрана свёрточная нейронная сеть ResNet (Residual Network) [9]. Архитектура данной сети реализует идею передачи значений выхода и входа двух последовательно расположенных свёрточных слоёв для последующих слоёв (рис. 2) [9].

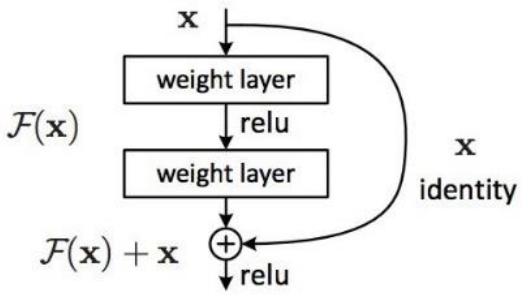

Рис. 2. Строительный блок ResNet.

Это первый вид нейронный сетей, в которых количество слоев может быть очень большим без риска деградации сети. Так, размеры некоторых сетей могут достигать 1000 слоёв, и это не приводит к росту ошибки классификации. Таким образом, данная архитектура позволяет справиться с так называемой деградацией нейронной сети, когда на определённом этапе увеличение слоев больше не дает положительного результата, а лишь увеличивает ошибку (данное ухудшение качества работы никак не связано с переобучением [9]).

Архитектура ResNet имеет большое количество вариантов: ResNet-50, ResNet-101, ResNet-152 и другие. Число в названии обозначает количество слоев данной нейронной сети.

Для решения специализированной задачи классификации фруктов на изображении была выбрана архитектура ResNet-50. Такой выбор был сделан на основании того, что данный вид нейронной сети требует наименьших вычислительных затрат для обучения.

4 Данные и программные средства для обучения нейронной сети

В качестве набора данных для обучения нейронной сети был выбран датасет [12], содержащий 970 изображений, принадлежащих к 30 классам различных фруктов. Каждое изображение сформировано по принципу – один класс фруктов на одном изображении. Фрукты расположены под различными ракурсами и иногда перекатываются листьями, рукой человека или другими объектами, не связанными с анализируемыми классами фруктов. Изображения размещены в 30 папках, каждая из которых названа соответствующим классом фрукта, т. е. изображения класса X располагаются в папке с именем X.

Обычно, для решения подобных задач используют библиотеки для работы с нейронными сетями, такие как TensorFlow, Torch, Theano, Caffe. Одной из реализаций высокоуровневых библиотек является Keras, которая является надстройкой над TensorFlow и Theano. Основной программный интерфейс этих библиотек реализован на языке программирования Python, который и был выбран для программного решения задачи. Указанные библиотеки содержат готовые слои, строительные блоки, функции и модели различных нейронных сетей. Кроме того, важным является то, что многие библиотеки уже содержат заранее натренированные модели на датасетах ImageNet или CIFAR-10 [13] для различных популярных нейронный сетей.

Для реализации классификатора для нашей задачи была выбрана библиотека Keras [14], которая содержит модель сети ResNet-50 с уже натренированными

весами на датасете ImageNet. Таким образом, был использован широко распространенный подход «передача знаний» (англ., transfer learning) – процесс заимствования предобученной нейронной сети.

5 Настройка нейронной сети ResNet-50

Процесс настройки заранее натренированной сети ResNet-50 для решения задачи классификации изображений с фруктами будем осуществлять поэтапно.

Этап 1. Сначала необходимо внести изменение в последние слои нейронной сети ResNet-50. Так, необходимо заменить полносвязный классифицирующий слой для 1000 классов на свои слои, необходимые для классификации на 30 классах. Для этого был создан новый двумерный слой пулинга с функцией avg и новый выходной полносвязный слой, имеющий 30 нейронов с функцией активации softmax в соответствии с количеством классов в тренировочном датасете. После этих изменений была пересоздана модель сети.

Этап 2. Для того, чтобы уменьшить риск переобучения нейронной сети на имеющемся небольшом наборе обучающих данных (910 изображений + 60 тестовых) была применена аугментация как тренировочных, так и тестовых изображений (искажение изображений и за счет этого увеличение их количества). Для этого были применены такие типы искажений: масштабирование RGB канала в пределах от 0 до 1 (вместо [0, 255] по умолчанию), случайный поворот изображения в пределах 90 градусов, вертикальное и горизонтальное зеркальные отображения, случайное масштабирование изображения в пределах 20%, сдвиг изображения по горизонтали и вертикали в пределах 20%, а также сдвиг RGB канала в пределах 20%.

Этап 3. На этом этапе необходимо обучить добавленные слои. Фактически, в библиотеке Keras в реализации ResNet-50 содержится 174 слоя (50 из них свёрточные) с учетом активации ReLU, сложения, и батч нормализации. Для того, чтобы убрать случайную инициализацию параметров добавленных слоёв, необходимо их обучить. Чтобы не проводить заново обучение всей сети, нужно заморозить все веса, кроме вновь добавленных (с первого по 151 слой включительно).

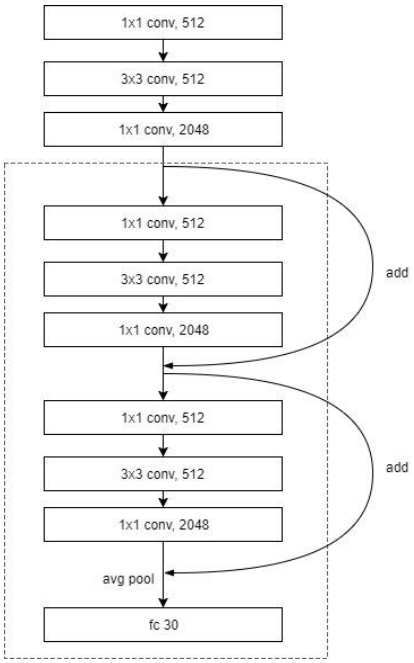

Рис. 3. Последние обучаемые блоки сети ResNet-50, добавленные при transfer learning подходе

Этап 4. Наконец, необходимо выполнить обучение последних двух блоков сети ResNet-50 (см. рис. 3). В качестве функции оптимизации использовался стохастический градиентный спуск [17] с маленьким шагом обучения 0.0001 и моментом 0.9. В качестве функции ошибки была применена категориальная кросс-энтропия, так как используется вероятностное интерпретирование результата на конечном слое нейронной сети [18].

Для обучения были выбраны такие параметры: размер блока изображений (batch size), который будет проходить через сеть за один шаг равен 10 (такой размер шага выбран так, чтобы число тренировочных изображений делилось на это число нацело); число шагов, которое нужно для прохода одной эпохи равно 91. Для нашей задачи это число вычислено так: количество всех изображений (910) делится на размер блока (10), т. е. каждая эпоха будет состоять из 91 шагов. Для обучения было выбрано 15 эпох, данное число было выбрано эмпирическим путём.

В процессе обучения, после первой эпохи, ошибка составляла 0.657 и точность 0.814. В конце обучения мы получили ошибку 0.394 (loss) и точность 0.906 (accuracy). Таким образом, после обучения нам удалось понизить ошибку на 0.262 и увеличить точность на 0.092.

6. Результаты тестирования обученной нейронной сети

Для тестирования нейронной сети было выделено 60 изображений с фруктами (по 2 изображения на класс). Применение аугментации с указанными выше параметрами увеличивает их фактическое количество до 142. Загрузив нашу обученную модель с диска, мы пропустили через сеть тестовые

изображения 10 раз, затем усреднили результаты и получили следующие данные:

1. Ошибка классификации составила 0.4272
2. Точность: 0.865
3. Среднее время предсказания класса одного изображения: 0.68 сек.

Также было произведено тестирование классификатора на 30 разных изображений (по 1 изображению на каждый класс фрукта) с выводом результатов предсказания (маркировки изображения предсказанными классами). На рис. 4 представлен пример такого вывода для топ-двух предсказанный классов для первых трёх изображений.

```
1. Искомый класс : "apples" | предсказанные классы :  "apples" : "0.259027"; "blueberries" : "0.221447";

2. Искомый класс : "apricots" | предсказанные классы :  "blueberries" : "0.907302"; "avocados" : "0.028806";

3. Искомый класс : "avocados" | предсказанные классы :  "avocados" : "0.932468"; "apples" : "0.0218184";
```

Рис. 4. Вывод результата маркировки изображений классами.

Проведенное тестирование показало, что количество правильно предсказанных топ-первых классов составляет 21 из 30. То есть точность классификации при данном подходе составила 70%.

Выводы

В данной работе представлен процесс создания классификатора изображений и применения его к специализированной задачи классификации изображений с фруктами. Была кратко рассмотрена архитектура свёрточных нейронных сетей. Был обоснован выбор нейронной сети ResNet-50 для решения задачи. Представлен процесс настройки модели сети для решения задачи, а также набор тренировочных изображений с фруктами для обучения нейронной сети.

В результате нейронная сеть показала ошибку 0.4272, точность 0.865, среднее время предсказания одного изображения 0.68 сек и при втором подходе к тестированию точность – 70%. Это является хорошим результатом с учётом относительно небольших вычислительных мощностей и небольшого времени, потраченного на обучение сети.

Полученная таким образом свёрточная нейронная сеть может быть использована в приложениях для работы с изображениями, например, для поиска изображений с конкретным классом фруктов, группировки изображений по классам фруктов и для других прикладных задач. Также при незначительной правке архитектуры полученная модель может применяться для задачи маркировки изображений (multi-label classification).

ЛИТЕРАТУРА

1. ImageNet Large Scale Visual Recognition Competition [Электронный ресурс]. - Режим доступа: http://image-net.org/challenges/LSVRC/
2. ImageNet [Электронный ресурс]. - Режим доступа: http://www.image-net.org/

3. Yann LeCunn, Leon Bottou, Yoshua Bengio, Patrick Haffner. Gradient-Based Learning Applied to Document Recognition [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
4. Clarifai: What is visual recognition? [Электронный ресурс]. – Режим доступа: https://www.clarifai.com/technology
5. Wikipedia: Kernel (image processing) [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Kernel_(image_processing)
6. The data science blog. An Intuitive Explanation of Convolutional Neural Networks [Электронный ресурс]. – Режим доступа: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
7. Learn OpenCV. Understanding Activation Functions in Deep Learning [Электронный ресурс]. – Режим доступа: https://www.learnopencv.com/understanding-activation-functions-in-deep-learning/
8. CS231n Convolutional Neural Networks for Visual Recognition [Электронный ресурс]. – Режим доступа: http://cs231n.github.io/convolutional-networks/
9. Kaiming He, Xiangyu Zhag, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1512.03385
10. Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariant Shift [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1502.03167
11. Reddit MachineLearning: How does DenseNet compare to ResNet and Inception? [Электронный ресурс]. – Режим доступа: https://www.reddit.com/r/MachineLearning/comments/67fds7/d_how_does_densenet_compare_to_resnet_and/
12. VICOS. Fruit Image Data set [Электронный ресурс]. – Режим доступа: http://www.vicos.si/Downloads/FIDS30
13. The CIFAR-10 dataset [Электронный ресурс]. – Режим доступа: https://www.cs.toronto.edu/~kriz/cifar.html
14. Keras: The Python Deep Learning Library [Электронный ресурс]. – Режим доступа: https://keras.io/
15. TensorFlow: An open-source software library for Machine Intelligence [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/
16. Anaconda: Python Data Science Platform [Электронный ресурс]. – Режим доступа: https://www.anaconda.com/download/
17. UFLDL Tutorial. Optimization: Stochastic Gradient Descent [Электронный ресурс]. – Режим доступа: http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/
18. Dr. Kevin Koidl. Loss Functions in Classification Tasks [Электронный ресурс]. – Режим доступа: https://www.scss.tcd.ie/Kevin.Koidl/cs4062/Loss-Functions.pdf
19. CS231n CNN for Visual Recognition. Linear Classification. [Електроннийресурс]. – Режим доступу: http://cs231n.github.io/linear-classify/

Надійшла – 15.12.2017.