УДК 004.048

КЛАССИФИКАЦИЯ ОБЪЕКТОВ ПРИ ПОМОЩИ ОБРАБОТКИ МАЛОЙ ВЫБОРКИ ДАННЫХ СВЁРТОЧНОЙ НЕЙРОСЕТЬЮ

Ильичев В.Ю. [1]

[1] *Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: patrol8@yandex.ru*

Работа посвящена разработке и обучению специального типа нейронной сети для работы с малой выборкой данных. Такая сеть содержит несколько свёрточных слоёв, хорошо работающих при решении задач классификации объектов. Топография нейросети и методики работы с ней реализованы в виде программы на языке Python с использованием модуля глубокого обучения Keras.
Описана последовательность процедуры разделения исходного набора данных по исследуемым объектам (файлам изображений) на обучающую и валидационную выборки, а также процесса подбора типов слоёв сети и их параметров. Выбран метод компиляции модели, позволяющий достичь максимальной эффективности её работы.
В статье описан также процесс обучения нейросетевой модели на достаточно небольшом массиве изображений, получена зависимость качества классификации от количества циклов (эпох) обучения. Определено оптимальное количество эпох обучения, которое позволяет добиться максимального качества классификации объектов.
Так как машинное обучение является непрерывно и стремительно развивающейся областью науки, то перечисленные этапы можно реализовать разными способами, из которых выбран лишь один возможный. Язык Python и его модули позволяют программистам и учёным творчески решать данную задачу различными методами, и затем сравнивать качество результатов, чему способствует свободное распространение модулей и популярность данного языка программирования во всём мире.
Основной библиотекой функций для работы с нейросетями является модуль Keras, но при создании описываемого программного продукта использованы также модуль обработки массивов данных Numpy, модуль машинной графики Matplotlib, модули Graphviz и Pydot для визуализации топографии нейросетей. Программа отличается тем, что чаще всего даже достаточно сложная процедура реализуется в виде очень короткой части программного кода и исследователь имеет широкие возможности экспериментирования с подбором наилучшего набора параметров используемых функций с точки зрения достижения наивысшего качества модели.
По результатам представленной работы сформулированы выводы и даны рекомендации для дальнейшего применения рассмотренной методики.

Ключевые слова: свёрточные сети, нейросетевое программирование, язык Python, модуль Keras, классификация изображений.

CLASSIFICATION OF OBJECTS BY PROCESSING A SMALL SAMPLE OF DATA BY A CONVOLUTION NEURAL NETWORK

Ilichev V.Y. [1]

[1] *Kaluga Branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru*

The work is devoted to the development and training of a special type of neural network for working with small data sampling. Such a network contains several convolutional layers that work well when solving object classification problems. The topography of the neural network and methods of working with it are implemented in the form of a Python program using the Keras deep learning module.
The procedure of dividing the initial data set by the examined objects (image files) into training and validation samples, as well as the process of selecting types of network layers and their parameters, is described. The method of compiling the model was chosen, allowing to achieve maximum efficiency of its work.
The article also describes the process of training the neural network model on a fairly small array of images, the dependence of the classification quality on the number of learning cycles (eras) is obtained. The optimal number

of learning eras has been determined, which allows to achieve the maximum quality of classification of objects.
Since machine learning is a continuously and rapidly developing field of science, the above stages can be implemented in various ways, of which only one possible is chosen. The Python language and its modules allow programmers and scientists to creatively solve this problem by various methods, and then compare the quality of the results, which is facilitated by the free distribution of modules and the popularity of this programming language throughout the world.
The main function library for working with neural networks is the Keras module, but when creating the described software product, the Numpy data array processing module, the Matplotlib machine graphics module, the Graphviz and Pydot modules for visualizing the topography of neural networks are also used. The program differs in that most often even a rather complex procedure is implemented in the form of a very short part of the program code and the researcher has wide possibilities of experimenting with the selection of the best set of parameters for the functions used in terms of achieving the highest quality of the model.
Based on the results of the presented work, conclusions are formulated and recommendations are made for further application of the considered methodology.

Keywords: convolution networks, neural network programming, Python language, Keras module, image classification.

Введение. Одной из проблем, требующих решения с помощью современных методов обработки данных, является задача машинной классификации объектов (то есть сортировки выборки объектов по классам) [1].

Классификация данных используется во многих отраслях, например, таких как:

- защита информации, где необходимо отделить секретную и важную информацию от менее важной, актуальные данные от неактуальных;
- отделение исправных изделий от неисправных;
- создание контекстных выборок в веб-приложениях [2];
- медицинская диагностика;
- распознавание образов на изображениях.

В настоящее время наиболее эффективным инструментом для создания классификаторов является использование программ, в основе которых лежат методы искусственного интеллекта (чаще всего, основанные на применении нейронных сетей).

В представленной работе рассмотрена реализация двухклассового распределения изображений с помощью функций библиотек обработки данных и нейросетевого программирования Keras, SkiPy, PIL.

Особенность представленной задачи заключается в том, что нейросеть должна иметь специальную архитектуру для работы с относительно небольшим количеством не очень качественных исходных данных для обучения и валидации – топографию с использованием свёрточных слоёв [3]. Данное направление развития нейросетей в настоящее время является очень востребованным, т.к. обычно сложно получить базу данных по объектам с большим количеством качественных данных.

Цель исследования. Частной задачей данного исследования является создание нейронной сети для классификации изображений – разбиения базы фотографий на два класса – изображений кошек и собак (бинарная классификация). Для обучения и валидации

создаваемой сети используется относительно небольшая база фотографий в формате .jpg. При этом выбираются первые попавшиеся по поиску в системе google, как правило не очень качественные, фотографии кошек и собак, на многих из которых присутствуют также люди и множество посторонних предметов, а изображения животных занимают часто только малую площадь изображения.

Для обучения сети используется по 1000 произвольно выбранных фотографий по каждому классу (всего 2000 изображений), для валидации – по 400 фотографий по каждому классу.

При этом используются специальные возможности модуля Keras [4]:

- функция создания генератора для обучения модели Keras;
- специальное сочетание разных типов слоёв нейросети для эффективной работы с небольшим массивом данных;
- тонкая настройка модели с помощью длительного обучения.

При использовании больших выборок данных и методик глубокого обучения можно достичь точности классификации до 98%. Однако, при работе с малой выборкой некачественных данных достижимая точность классификации гораздо ниже. Более ранние исследования, проведённые с базой данных из 25000 изображений, показали, что при классификации можно добиться точности не более 60% [5]. Однако, в последнее время после детальной разработки архитектуры нейросети, работающей с малой выборкой данных, наблюдается значительный прогресс и в некоторых случаях достигнута точность классификации до 80%.

Данное исследование посвящено именно проблеме использования малых выборок. В данном случае для целей классификации лучше всего зарекомендовали себя достаточно сложные свёрточные (convolution) нейросети с использованием метода глубокого обучения, которые и применены в разрабатываемой программе.

Разработанный подход был опробован на специально созданной нейросети, на основе которой скомпилирована и обучена модель классификации. Следующей задачей, решаемой в работе, является оценка достигаемого качества классификации объектов.

Материал и методы исследования. Рассмотрим кратко принципы, использованные в процессе создания программы на языке Python для классификации объектов с помощью нейронной сети специальной архитектуры с применением функций библиотеки Keras.

Как уже отмечено выше, наиболее подходящим инструментом для осуществления классификации изображений является применение свёрточных сетей. Основным их свойством является способность выделять характерные особенности исследуемых объектов и поэтому в сфере распознавания образов у них на данный момент нет конкурентов.

Принцип организации свёрточной сети основан на том, что пиксели изображения, находящиеся рядом, более сильно влияют на характеристику моделируемого признака, чем

пиксели, находящиеся далеко друг от друга [6].

Свёрточная сеть является намного более «продвинутым» вариантом обычной, полносвязной, нейронной сети, в котором для дополнительной обработки областей изображения применяется так называемая свёртка. В библиотеке Keras свёрточный слой общей нейросети обозначается Conv2D. Этот слой подобен полносвязному слою, и так же содержит веса и смещения, которые подвергаются оптимизации (подбору). Кроме того, слой Conv2D содержит ещё фильтры («ядра»), создающие свёртки, значения параметров которых также должны оптимизироваться.

В нашей задаче использовано мало обучающих образцов, поэтому главная задача – не дать сети переобучиться [7]. Переобучение происходит, когда модель начинает действовать по неправильным шаблонам, обобщая и выводя правила из абсолютно частных образцов данных (подобно образованному человеку, не умеющему применять свои многочисленные знания на практике).

Основным направлением борьбы с переобучением является подбор энтропической мощности (entropic capacity) модели – то есть насколько много информации разрешено хранить внутри модели. Модель, которая может хранить много информации, может быть более точной за счёт использования большего количества функций, но она также в большей степени подвержена риску хранения неверных функций. Между тем, модель, которая может хранить только несколько функций, должна будет сосредоточиться на наиболее значимых особенностях, найденных в данных, и они, скорее всего, будут действительно верны и будут обобщены лучше.

Существуют различные способы подбора энтропической мощности. Основной из них - выбор количества параметров в модели, т.е. количества слоев и количество нейронов каждого слоя. Второй способ – подбор модели оптимизации энтропии при компиляции созданной нейросети.

В нашем случае используется свёрточная сеть с большим количеством слоев и с применением фильтров на каждом уровне обработки данных, применяемых для отсева данных. Отсев лишних данных способствует снижению опасности переобучения, не позволяя слою обрабатывать дважды одинаковую картину, таким образом устраняя ложное впечатление увеличения количества данных.

Разработанная нейронная сеть формируется последовательным соединением следующих слоёв [8] (рис. 1):

- InputLayer – входной слой для подачи на него обучающих образцов изображений;
- Три последовательных свёрточных слоя Conv2D, связанных с активационными слоями Activation с передаточной функцией ReLU (линейный выпрямитель Rectified Linear Unit) и

последующими выходными тензорами MaxPooling2D. Целью активационного слоя является достижение бинарности данных на выходе («да-нет»), а выходные тензорные слои уменьшают размерность выходной выборки на основе выбора только максимального значения из подвыборок размером pool_size;

- Flatten - слой для преобразования изображения выходной выборки в одномерный вектор;
- Два полносвязных слоя Dense с последующей активацией слоями Activation, использующих соответственно функцию ReLU и сигмоидную функцию. Между данными слоями внедрён дополнительный слой Dropout - слой прореживания для решения проблемы переобучения сети.

Выход сети является бинарным с двумя вариантами классификации (кошка-собака).

После задания типов и параметров слоёв в программе предусмотрены команды для вывода топографии нейросети на экран и в графический файл (рис. 1).

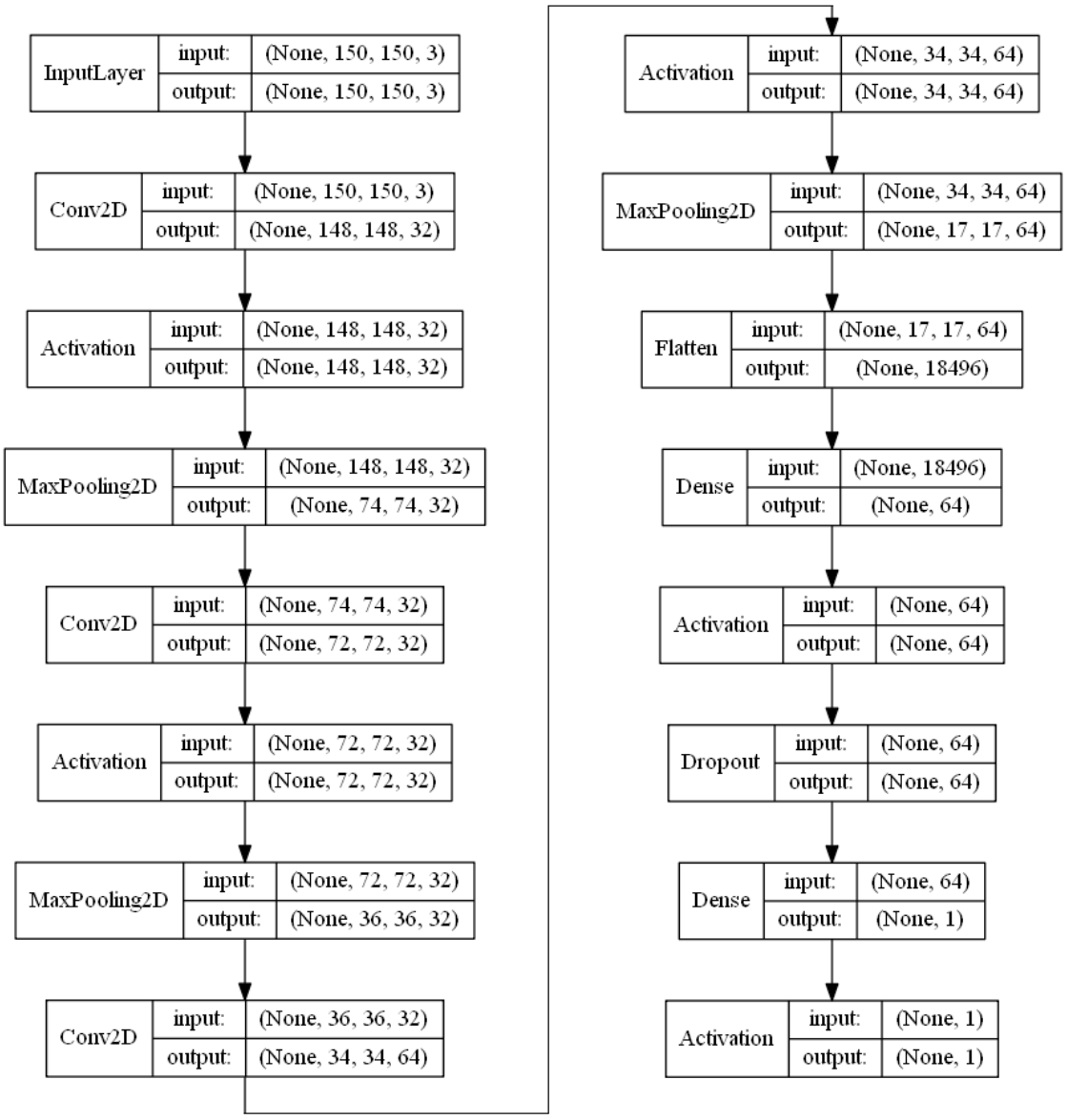

Рис. 1. Топография созданной нейронной сети.

Затем производится компиляция созданной нейросети, для чего использован метод бинарной кроссэнтропии в качестве функции потерь (её назначение описано выше), метрика точности и один из самых современных оптимизаторов адаптивного обучения Rmsprop [9].

После этого осуществляется собственно процесс обучения сети и создания модели классификации. Для исключения переобучения применяется функция перемешивания обучающей выборки и генерации некоторого дополнительного объёма данных с помощью класса предобработки изображений ImageDataGenerator модуля Keras. Этот класс позволяет, в частности, поворачивать изображения на произвольный угол, перемещать их по вертикали или горизонтали, масштабировать и изменять палитру цветов изображения.

После каждого цикла (эпохи) обучения производится валидация модели на соответствующем наборе изображений и определяется точность осуществления ею классификации. Значения достигнутой точности записываются в массив данных.

В начале программы в качестве исходных данных заданы переменные с параметрами выборки данных и настройками обучения сети. Например, в рассматриваемом случае задаётся размер изображений (150×150 пикселей), пути к директориям на жёстком диске, в которых размещены обучающие и валидационные выборки изображений, количество образцов изображений для обучения сети и её валидации, число эпох обучения.

Также задаётся размер подвыборки из 16 изображений. Такое деление изображений на подвыборки необходимо для ускорения работы компьютера и исключения переполнения оперативной памяти.

В конце программы заданы команды для вывода на экран и в файл графика по результатам обучения модели, а также для сохранения самой модели в файл для возможности её дальнейшего использования для целей классификации объектов (для этого написана отдельная небольшая программа на языке Python).

Результаты. Выполнение расчётов с помощью созданной программы осуществлялось на компьютере с двухъядерным процессором (CPU) с тактовой частотой 2 ГГц и 4 Гб оперативной памяти. Обучение модели по 100 эпохам заняло около 15 часов. Это время может быть значительно сокращено при использовании более мощной конфигурации компьютера и процессора мощной графической карты (GPU вместо CPU).

Созданный массив результатов оценки точности классификации в процентах в зависимости от количества эпох обучения выведен в графическом виде с помощью библиотеки Matplotlib на рис. 2.

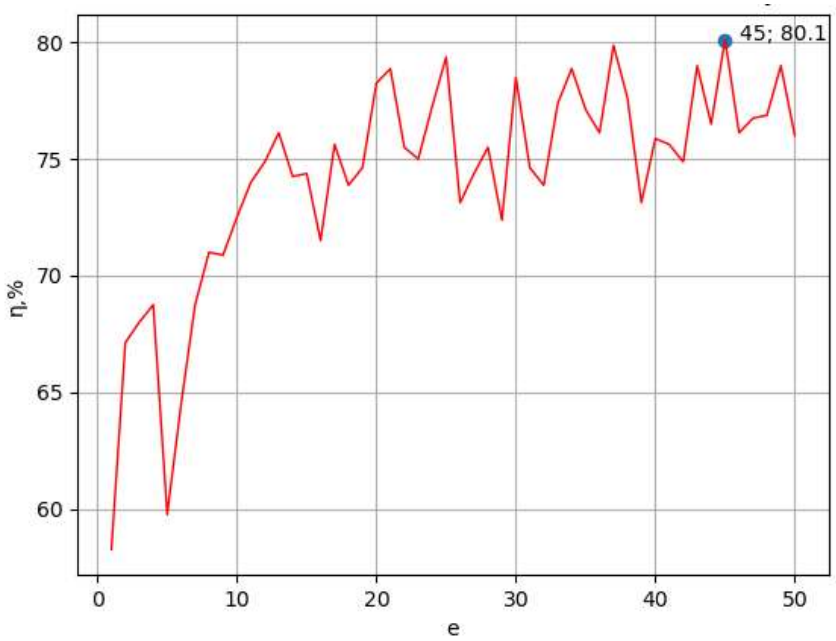

Рис. 2. Зависимость качества классификации нейросетью от количества эпох обучения.

По графику видно, что качество обучения колеблется в зависимости от количества эпох и вначале качество классификации в среднем увеличивается. Колебания оценки качества модели можно объяснить наличием неявной связи между архитектурой сети и числом эпох. Наилучшее качество обучения (более 80%) было достигнуто при числе эпох 45. При дальнейшем увеличении количества эпох качество модели не повышается (график изображён только для 50 начальных эпох, т.к. далее оценка качества колеблется около значений 75-80%, не превышая их). Полученное качество обучения близко к предельно допустимым значениям при небольшом наборе данных, поэтому можно сделать вывод, что использованная топография сети с применением свёрточных слоёв хорошо справилась с задачей бинарной классификации.

Обсуждение и заключение. Особенность модели, основанной на описанной нейросети, имеющей в составе несколько свёрточных слоёв, состоит в том, что она способна реализовать высокое качество классификации объектов даже при малом количестве исходных данных для обучения и валидации. Такой анализ данных является крайне востребованным в настоящее время, так как абсолютное большинство баз данных как раз состоят из небольшого объёма не очень качественных образцов данных.

По результатам выполненной работе можно сформулировать следующие рекомендации:

- при разбиении исходной базы данных следует отводить на обучающую выборку в несколько раз большее количество объектов, чем на валидационную выборку;
- важно правильно подобрать топографию нейронной свёрточной сети, которая будет давать высокое качество классификации именно при малом количестве исходных данных;

- надо использовать наиболее современный и совершенный метод компиляции сети;
- избегать переобучения сети, которое в рассматриваемой задаче очень вероятно;
- при разработке модели можно использовать оценку её качества, предложенную в статье.

Описанные принципы создания и обучения свёрточных нейросетей с использованием модуля Keras [10] и прочих библиотек языка Python рекомендуется применять также для классификации любых других объектов по малым выборкам данных.

Список литературы

1. Москвитин А.А., Созиев Т.М. Особенности современных методов интеллектуального анализа данных. В сборнике: Современные методы интеллектуального анализа данных. Российский экономический университет им. Г.В. Плеханова. 2016. С. 11-18.

2. Стоянченко С.С., Демин О.В. Исследование применения алгоритмов классификации для интеллектуальных веб-приложений. В сборнике: Инновационные материалы и технологии. 2018. С. 52-54.

3. Толстых А.А., Голубинский А.Н. Алгоритм выбора архитектуры полносвязной сети в задачах распознавания изображений на основе сверточных нейронных сетей. В сборнике: Радиолокация, навигация, связь. Воронежский гос. университет. 2019. С. 156-163.

4. Machine Learning Repository. Center for Machine Learning and Intelligent Systems. [Электронный ресурс]. URL: https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/ (дата обращения: 30.06.2020).

5. Philippe Golle, Palo Alto. Machine Learning Attacks Against the Asirra CAPTCHA. [Электронный ресурс]. URL: http://xenon.stanford.edu/~pgolle/papers/dogcat.pdf (дата обращения: 30.06.2020).

6. Толстых А.А., Голубинский А.Н. Программа тестирования эффективности распознавания объектов на изображениях сверточными нейронными сетями. Свидетельство о регистрации программы для ЭВМ RU2019616219, 20.05.2019. Заявка № 2019615108 от 26.04.2019.

7. Воронецкий Ю.О., Жданов Н.А. Методы борьбы с переобучением искусственных нейронных сетей. Научный аспект. 2019. Т. 13. № 2. С. 1639-1647.

8. A Beginner's Guide to Neural Networks and Deep Learning. A.I. Wiki. [Электронный ресурс]. URL: https://pathmind.com/wiki/neural-network (дата обращения: 30.06.2020).

9. Vitaly Bushaev. Understanding RMSprop - faster neural network learning. [Электронный ресурс]. URL: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a (дата обращения: 30.06.2020).

10. Keras: The Python Deep Learning library. Keras documentation. [Электронный ресурс]. URL: https://keras.io/#why-thisname-keras. (дата обращения 30.06.2020).