УДК 004.048

ОБРАБОТКА СТАТИСТИЧЕСКИХ ДАННЫХ МЕТОДОМ ГЛУБОКОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДУЛЯ KERAS

Ильичев В.Ю., Юрик Е.А.

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

В статье описана разработанная авторами методика работы с нейронными сетями, предназначенными для классификации результатов статистических исследований. Методика реализована с использованием нейросетевого модуля Keras и ряда других модулей языка Python. Работа со статистическими данными с помощью библиотек Python организована путём последовательного осуществления ряда этапов: подбора и подготовки данных, выбор типа и топографии создаваемой сети, способа её обучения (реализовано так называемое глубокое обучение), оценки точности получаемых результатов обучения. Способы осуществления каждого из описанных этапов в настоящее время не являются достаточно отработанными, т.к. язык Python и методы, реализуемые с его помощью, находятся в постоянном непрерывном развитии. Так как Python, как и его библиотеки, являются свободно распространяемым программным обеспечением, то это упрощает и интенсифицирует развитие по всему миру этого универсального языка и специализированных модулей для него. Но, с другой стороны, каждый разработчик программ на этом языке реализует свои идеи по-разному. Из всех используемых в данной работе модулей следует отметить библиотеки работы с массивами Numpy, с графическим выводом результатов Matplotlib, выводом топографии сети Graphviz и Pydot. Хотя в программе применяется много модулей, но команды, реализуемые с их помощью, являются несложными для освоения даже неспециалистами в программировании. Созданный программный продукт был обучен на выборке экспериментальных клинических исследований показателей здоровья большой группы пациентов, результатом предсказаний должно являться наличие заболеваний сердца у тестируемых пациентов. Расчёты проведены на организованной средствами модуля Keras нейронной сети, произведена оценка точности прогнозирования в зависимости от количества циклов (эпох) обучения. В результате найдено число эпох, при котором достигается максимальное качество обучения созданной сети. В заключение сформулированы выводы по результатам представленной работы и даны рекомендации для её дальнейшего использования.

Ключевые слова: нейросетевое программирование, глубокое обучение, язык Python, модуль Keras, прогнозирование событий

PROCESSING OF STATISTICS BY DEEP LEARNING USING KERAS MODULE

Ilichev V.Y., Yurik E.A.

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

The article describes the methodology developed by the authors to work with neural networks designed to classify the results of statistical studies. The technique is implemented using the Keras neural network module and a number of other Python language modules. Work with statistical data using Python libraries is organized by sequentially implementing a number of stages: selecting and preparing data, choosing the type and topography of the network being created, how to train it (the so-called «deep learning» is implemented), assessing the accuracy of the resulting training results. The methods for carrying out each of the described steps are not currently sufficiently developed, since the Python language and the methods implemented by it are in constant continuous development. Since Python, like its libraries, are freely available software, this simplifies and attenuates the development of this universal language and its specialized modules around the world. But, on the other hand, each developer of programs in this language implements their ideas in different ways. Of all the modules used in this work, note the libraries for working with arrays Numpy, with the graphical output of results Matplotlib, and the topography of the network Graphviz and Pydot. Although the program uses many modules, the commands implemented using them are easy to master even by specialists in programming. The software product created was trained on a sample of experimental clinical studies of health indicators of a large group of patients, the result of the predictions should be the presence of heart diseases in the patients tested. Calculations were carried out on a neural network organized by the Keras module, and prediction accuracy was estimated depending on the number of learning cycles (eras). As a result, we found the number of eras at which the maximum quality of training of the created network is achieved. Conclusions were drawn on the results of the submitted work and recommendations were made for its further use.

Keywords: neural network programming, deep learning, Python language, Keras module, event prediction

Получение новых знаний практически в любой отрасли науки невозможно представить без сбора и обработки статистических данных.

Обработка данных наблюдений или экспериментов осуществляется с целью вывода зависимостей, позволяющих описать исследуемые процессы (эмпирико-теоретический метод исследований). В настоящее время анализ данных превратился в отдельную науку на стыке математики и информатики.

Известно множество способов анализа данных: регрессионный, канонический, кластерный, дискриминантный, корреляционный, факторный и множество других [1; 2]. При выполнении каждого типа анализа преследуются разные цели.

В последнее время широкое распространение получил нейросетевой анализ, основанный на имитации работы нервной системы человека. Для осуществления данного метода обработки данных необходимо создать так называемую нейронную сеть, а затем обучить её на исторических данных. Модель, полученная в результате обучения, при правильной организации этого процесса обладает свойствами предсказания событий с определённой высокой долей вероятности.

Целью данной работы является разработка программы, реализующей один из вариантов нейронной сети, позволяющей на раннем этапе прогнозировать наличие либо отсутствие заболеваний сердца у пациентов по известным параметрам их исследования в центре здоровья.

При этом также ставится задача демонстрации порядка работы с модулем нейросетевого программирования Keras, реализующим метод «глубокого обучения» (deep learning). Также используются другие библиотеки языка Python. Для «тренировки» сети набор экспериментальных данных обрабатывается специальным образом.

После обучения нейронной сети необходимо решить задачу оценки качества созданной на её основе модели исследуемого процесса возникновения заболеваний.

Материалы и методы исследования

Часто традиционными методами не удаётся описать отклик системы на воздействие входных факторов, но с этой задачей во многих случаях может справиться обучаемая нейронная сеть. Способ построения нейронной сети напоминает организацию нервной системы человека из нейронов [3].

Рассмотрим, например, использование нейронной сети при проверке предполагаемого диагноза больного на основании анализа показателей клинических исследований. Значения физиологических показателей подаются на нейроны первого слоя сети, обрабатываются, и уже в усиленном или ослабленном виде передаются на следующий слой. Таких слоёв необходимо создать несколько (входной, выходной и промежуточные). При обучении сети по факторам и известным результатам их воздействия на систему формируются коэффициенты усиления или ослабления сигналов межнейронными связями. Если система должна выдать значение одного результата (отсутствия или наличия заболевания), то последний слой такой сети состоит только из одного нейрона, на котором и вырабатывается отклик – ответ системы, необходимый исследователю.

При создании нейронной сети ключевыми моментами, от которых зависит качество дальнейшего прогнозирования, являются её архитектура (количество слоёв и нейронов в каждом слое, их взаимные связи или их отсутствие), набор факторов для обучения и формирования прогноза, а также способ обучения сети.

Особенности любой научной работы в первую очередь определяются составом исходных данных для исследования. В нашем случае были использованы открытые базы данных по болезням сердца в некоторых странах, хорошо подходящие для целей машинного обучения [4]. В них приводятся только данные анкетирования и медицинских замеров, без указания личных данных пациентов.

Эти данные собраны по следующим медицинским учреждениям:

1. Кливлендская клиника.
2. Венгерский институт кардиологии, Будапешт.
3. Ветеранский медицинский центр, Лонг-Бич, Калифорния.
4. Университетская больница, Цюрих, Швейцария.

По каждому учреждению для каждого пациента собрано по 74 параметра, из которых для диагностики болезней сердца имеют смысл только 14 (причём многие из них выражены условными обозначениями, отображающими показатель не количественно, а качественно), а именно: возраст, пол, тип грудной боли, кровяное давление в покое, холестерин в крови, сахар в крови, результаты кардиографии, частота сердечного ритма, стенокардия при физической нагрузке, наклон пика ST-сегмента на кардиограмме при нагрузке, форма пика ST-сегмента, количество крупных сосудов (0–3), окрашенных при флюорографии, тип сердечного ритма. Результат (14-й параметр) выражен бинарным образом: 0 – риск сердечных заболеваний незначительный, 1 – риск сердечных заболеваний высокий (сужение диаметров крупных сосудов превышает 50 %).

Из 4 баз данных для исследования выбрана наибольшая по количеству записей (объёму) – база Венгерского института кардиологии. Для обучения нейронной сети не использовались столбцы 10–13, в которых практически отсутствуют данные (они имеются лишь по некоторым пациентам). Кроме того, исключено небольшое количество строк, в которых часть записей в столбцах 1–9 отсутствует. В результате такой предварительной обработки базы в ней осталась 261 строка. Для обучения базы использовались столбцы 1–9 с факторами и столбец 14 с результатами.

Для загрузки файла данных и перемешивания строк базы данных случайным образом (для исключения ложного обучения) использована библиотека Numpy [5]. Затем для реализации алгоритма работы с нейронной сетью использован современный высокоуровневый модуль Keras, обладающий широкими возможностями, написанный на языке Python и являющийся надстройкой для модуля более низкого уровня организации Tensorflow (так называемого фреймворка).

Рассмотрим порядок создания нейронной сети и работы с ним с помощью модуля Keras.

1. Указываем столбцы факторов – входных параметров X (с 1 по 9) и столбец результата – целевой переменной Y (14-й).

2. Выбираем модель нейронной сети Sequential (прямого распространения СПР) – с последовательным соединением нейронных слоев, в которой информация проходит только в прямом направлении. Как показала практика, такая сеть хорошо подходит для задач классификации и идентификации событий, подобных рассматриваемой.

3. Определяем слои нейронной сети. Входной слой имеет размерность 9 согласно количеству действующих факторов. Число нейронов выбрано достаточно условно равным 12 (в результате оптимизации сети для достижения наилучших результатов предсказаний). Далее добавлено ещё 3 слоя, состоящих соответственно из 15, 8 и 10 нейронов.

Созданная топология сети отображена средствами графических библиотек Graphviz и Pydot для Python [5] и показана на рис. 1.

На данном рисунке наглядно видно количество слоёв, нейронов в каждом слое, способ взаимного соединения слоёв, тип каждого слоя.

Тип активизации сетей выбран Rectified linear units (ReLU), считающийся в настоящее время наиболее подходящим для сетей с глубоким обучением. Выходной слой имеет один нейрон, т.к. целью сети является идентификация только одного параметра, для активизации этого слоя использован метод активизации с помощью сигмоидной функции, хорошо фильтрующей шумы.

4. Для возможности использования сеть нужно скомпилировать. Для этого использован популярный метод оптимизации Adam (полное название «метод адаптивной инерции») [6].

5. Выбираем метод обучения сети fit, включающий в себя разбиение данных на тренировочные (в данном случае матрица X из столбцов 1–9) и валидационные (матрица-вектор Y из столбца 14). Также задаётся количество проходов (эпох) обучения и количество строк для одного шага обучения. В дальнейшем будет рассмотрено влияние количество эпох на качество обучения созданной сети.

6. В конце программы задаём функцию evaluate модуля Keras, оценивающую точность прогноза по созданной модели и вывода значения точности на экран.

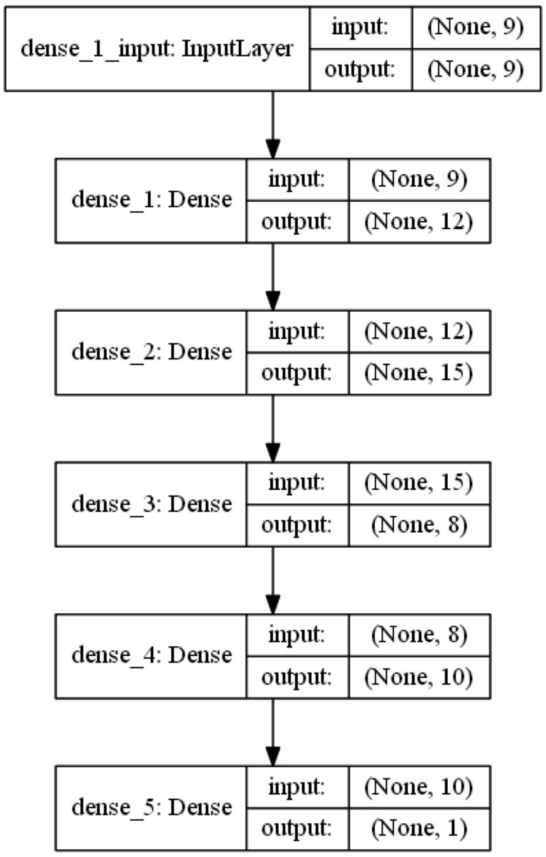

Рис. 1. Топография созданной нейронной сети

Результаты исследования и их обсуждение

Одной из целей исследований была оценка влияния количества эпох обучения на точность достигаемого результата. Нейросеть обучалась последовательно по 100 эпохам 500 раз (итого по 50000 эпохам), после каждого обучения оценивалась точность прогнозирования и заносилась в массив. Созданный массив результатов оценки точности в процентах в зависимости от количества эпох выведен в графическом виде с помощью библиотеки Matplotlib на рис. 2.

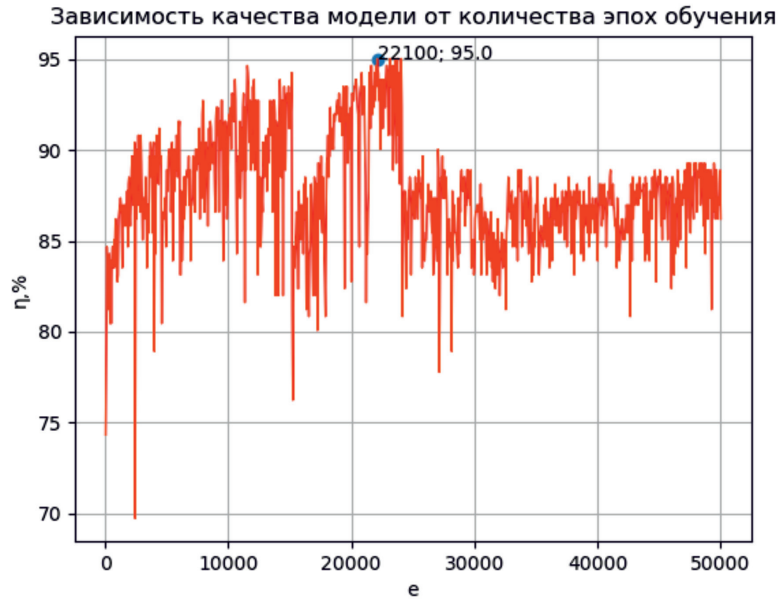

Рис. 2. Зависимость качества прогнозирования нейросетью от количества эпох обучения

Из рис. 2 можно сделать вывод, что качество обучения сети нестабильно в зависимости от количества эпох, хотя явно недообученная система (с количеством эпох до 1000) даёт более низкую точность прогноза. Нестабильность результата обучения можно объяснить снижением и повышением корреляционной связи между количеством нейронов в слоях и типом использованных слоёв и количеством данных для обучения при их варьировании. Максимальная точность прогноза наблюдается вблизи количества эпох 10000 и 20000. С помощью функции max библиотеки Numpy найдено максимально достигнутое качество прогноза 95% при количестве эпох обучения 22100. Поэтому такое количество эпох и рекомендуется использовать для обучения созданной нейросети.

С использованием описанных методов организации, обучения и оценивания нейронных сетей можно создавать сети различной топографии и подбирать из них подходящие наилучшим образом для решения поставленных задач. В случае необходимости кроме описанных библиотек в программе на Python можно использовать и дополнительные, специализированные библиотеки [7].

В данной работе рассмотрена методика работы с нейронными сетями с использованием библиотек языка Python. К настоящему времени эта задача является нетривиальной, так как язык Python и дополнительные модули для работы с ним динамично развиваются, появляются новые команды, изменяется синтаксис. Это создает определённые сложности в освоении этого универсального языка, но одновременно сильно расширяет его возможности.

В частности, для обеспечения гибкости работы с нейросетями язык Python с подключением модуля глубокого обучения Keras и прочих дополнительных модулей обладает следующими неоспоримыми преимуществами, одновременного сочетания которых нет в других программных продуктах [8]:

– модульность, компактность, расширяемость;

– лёгкость освоения неспециалистами;

– множество инструментов работы с нейронными сетями;

– реализация принципов глубокого обучения, получившего известность в последнее время и способного создавать прогнозы для более широкого круга явлений, чем традиционные способы обучения (такие как ненаправленная графическая модель);

– возможность эффективного использования графических процессоров (видеокарт) для вычислений;

– широкие возможности визуализации результатов с использованием модуля Matplotlib.

Данная статья иллюстрирует лишь частный случай – пример работы с модулем глубокого обучения Keras для распознавания заболеваний. Кроме этой цели, модуль Keras всё чаще начинают использовать при

машинном переводе текстов, распознавании условных графических обозначений, образов и даже людей по фотографиям [9].

Заключение

Из выполненной и описанной работы можно сформулировать следующие особенности работы с модулем Keras, которые необходимо учитывать для дальнейшей оптимизации работы с ним:

– важно правильно определить состав исходных факторов для обучения и прогноза, действительно оказывающих влияние на результат;

– необходимо выбрать тип и топографию нейронной сети, наиболее легко и быстро обучаемой и дающей наиболее точный прогноз;

–целесообразно выбрать наиболее эффективный метод компиляции сети из всех существующих, которые со временем совершенствуются;

– обязательно нужно подобрать количество эпох обучения, причём в этом случае важно не получить недообучения или переобучения сети, т.к. в этих случаях точность прогнозирования с её помощью уменьшается. Оптимальное количество эпох можно подобрать методом, описанным в данной статье.

Принципы использования модуля глубокого обучения Keras, изложенные в статье, можно применить для создания нейросетей с целью классификации многих других событий и объектов [10].

Список литературы

1. Москвитин А.А., Созиев Т.М. Особенности современных методов интеллектуального анализа данных // Современные методы интеллектуального анализа данных в экономических, гуманитарных и естественнонаучных исследованиях: материалы международной научно-практической конференции. Пятигорск, 2016. С. 11–18.

2. Низаметдинов Ш.У., Румянцев В.П. Анализ данных: учебное пособие для студентов высших учебных заведений. М., 2012. 285 с.

3. Голоскоков К.П. Применение нейронных сетей в задачах прогнозирования и проблемы идентификации моделей прогнозирования на нейронных сетях // Современные проблемы прикладной информатики: сборник научных трудов. 2006. С. 116–120.

4. Machine Learning Repository. Center for Machine Learning and Intelligent Systems. [Electronic resource]. URL: https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease (date of access: 03.07.2020).

5. Сысоева М.В., Сысоев И.В. Программирование для «нормальных» с нуля на языке Python. Учебник. В двух частях. Часть 1. М.: ООО «МАКС Пресс», 2018. 176 с.

6. Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization."arXiv preprint arXiv:1412.6980 (2014). [Electronic resource]. URL: https://arxiv.org/abs/1412.6980v8 (date of access: 03.07.2020).

7. Зубов М.В., Пустыгин А.Н. Анализ путей исполнения программ по исходному коду с использованием универсальных промежуточных представлений // Известия Юго-Западного государственного университета. 2015. № 4 (61). С. 12–19.

8. Тур А.И., Кокоулин А.Н., Южаков А.А., Лукичев А.Н. Подготовка системы распознавания объектов на базе Tensorflow и Keras // Международная конференция по мягким вычислениям и измерениям. 2018. Т. 1. С. 651–653.

9. Keras: The Python Deep Learning library. Keras documentation. [Electronic resource]. URL: https://keras.io/#whythisname-keras (date of access: 03.07.2020).

10. Хайбрахманов С.А. Основы научных расчётов на языке программирования Python: учебное пособие. Челябинск: Изд. Челябинского государственного университета, 2019. 96 с.