

**Пильцов Михаил Владимирович,**  
к.т.н., доцент, Ангарский государственный технический университет,  
e-mail: mpilcov@yandex.ru

**Позднухов Алексей Александрович,**  
обучающийся группы ЭН-21-1, Ангарский государственный технический университет,  
e-mail: aleksejdragunov910@gmail.com

## **ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА ТЕХНИЧЕСКИХ ДАННЫХ**

**Piltsov M.V., Pozdnukhov A.A.**

### **APPLICATION OF NEURAL NETWORKS FOR TECHNICAL DATA ANALYSIS**

**Аннотация.** Рассматривается применение нейронных сетей с использованием Keras, которая представляет собой открытую нейросетевую библиотеку, написанную на языке Python. Данная библиотека нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой.

**Ключевые слова:** нейрон, нейронные сети, машинное обучение, Keras.

**Abstract.** The article discusses the application of neural networks using Keras, which is an open neural network library written in the Python language. This library is aimed at operational work with deep learning networks, while being designed to be compact, modular and extensible.

**Keywords:** neuron, neural networks, machine learning, Keras.

Нейронные сети неразрывно связаны с термином «машинное обучение». Впервые определение этому термину дал Артур Самуэль – известный исследователь в области искусственного интеллекта, в 1959 году. По мнению Самуэля, машинное обучение – это процесс, в результате которого ЭВМ будут способны показать поведение, которое в них не было явно запрограммировано.

Технологии машинного обучения уже довольно длительное время применяются в различных сферах. Большое число интеллектуальных систем находит широкое применение в повседневной жизни: автоматические торговые системы, медицинская диагностика, голосовые помощники, техническая диагностика, системы, оценивающие кредитоспособность физических и юридических лиц и т. д.

Целью машинного обучения является научить интеллектуальную систему находить зависимость ответов от свойств объектов, которые являются входными данными, чтобы в дальнейшем использовать её для прогнозов, то есть научить систему решать задачи автоматизации принятия решений. Такие интеллектуальные системы строятся на основе нейронных сетей.

Любая нейронная сеть строится на основе нейронов. Под нейроном подразумевается математический объект (функция), имеющий несколько входов и один выход. Нейрон, имеющий два входа изображён на рисунке 1 [1].

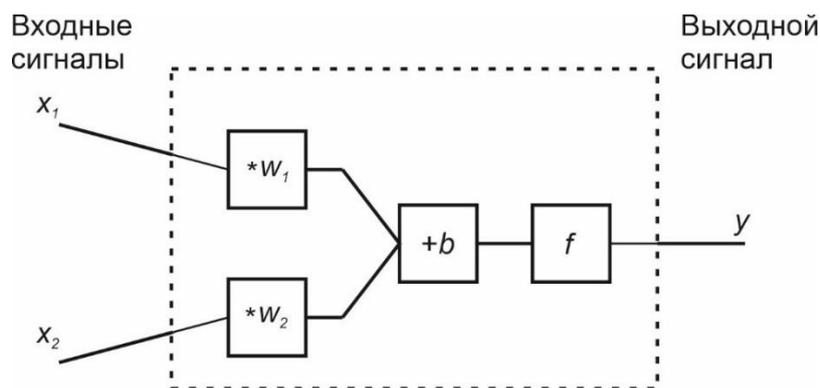


Рисунок 1 – Модель нейрона с двумя входами и одним выходом

Из рисунка видно, что выходной сигнал нейрона  $y$  зависит от входных сигналов  $x_1$  и  $x_2$ , весов  $w_1$  и  $w_2$ , смещения  $b$ , а также от функции активации  $f$ :

$$y = f(x_1 w_1 + x_2 w_2 + b). \quad (1)$$

Функция активации  $f$  нужна для того, чтобы выходной сигнал лежал в диапазоне от 0 до 1, при любых значениях входных сигналов, весов и смещения. Иными словами, данная функция сужает диапазон значений выходного сигнала нейрона до интервала (0, 1). В качестве функции активации, как правило, применяется сигмоида, которая описывается выражением:

$$f(x) = \frac{1}{1 + e^x}.$$

Нейроны, рассмотренные выше, связывают между собой, создавая нейронные сети. На рисунке 2 приведена простая нейронная сеть, состоящая из трёх нейронов [2].

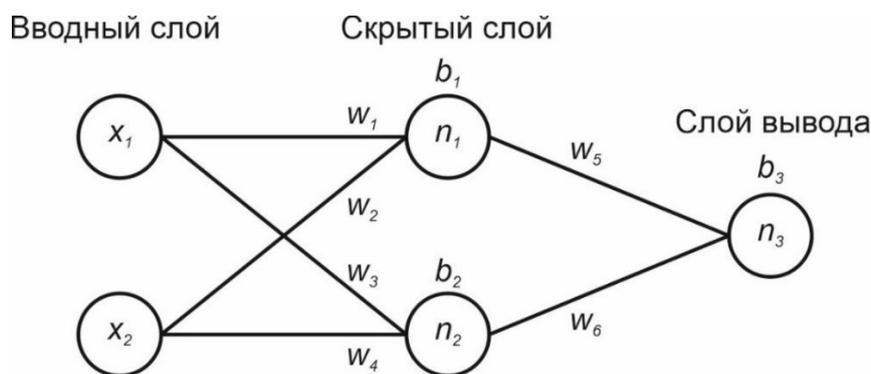


Рисунок 2 – Нейронная сеть

Вводный слой образуют два входа  $x_1$  и  $x_2$ . В скрытом слое расположены два нейрона  $n_1$  и  $n_2$ . Скрытыми называют слои, расположенные между входным и выходными слоями, при этом сеть может иметь несколько скрытых слоёв. В выходном слое находится один нейрон  $n_3$ . Заметим, что каждый из трёх нейронов на рисунке 2 имеет свои веса и смещения.

При описании нейронных сетей часто упоминается термин «обучение». Считается, что нейронная сеть обучена, если все веса и смещения её нейронов подобраны так, чтобы выходной нейрон выдавал сигнал, соответствующий входным данным. Например, входными данными сети на рисунке 2 могут быть рост и вес, а выходным сигналом пол (0 – женский пол, 1 – мужской пол). Если сеть верно обучена, то веса и смещения будут подобраны так, что мужскому полу будет соответствовать больший рост и вес, а женскому меньший.

Для оценки эффективности нейронной сети нужен некий критерий. В качестве такого критерия может использоваться среднеквадратическая ошибка, определяемая по формуле:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{true} - y_{predict})^2, \quad (2)$$

где:  $n$  – число рассматриваемых объектов (число человек, чей рост и вес были даны нейронной сети для обучения);  $y_{true}$  – истинное значение выходной переменной (пол, соответствующий данному росту и весу);  $y_{predict}$  – предполагаемое значение переменной (пол, предсказанный нейронной сетью, при поданных входных данных роста и веса).

Если для обучения сети использовать данные только одного человека, его рост, вес и правильный пол, то выражение (2) примет вид:

$$MSE = (y_{true} - y_{predict})^2. \quad (4)$$

Очевидно,  $MSE$  зависит от  $y_{predict}$ , а последнее от значений всех весов и смещений всех нейронов, входящих в сеть, иными словами, для сети на рисунке 2:

$$MSE(w_1, w_2, w_3, w_4, w_5, w_6, b_1, b_2, b_3).$$

Соответственно, веса и смещения нужно подбирать так, чтобы минимизировать ошибку  $MSE$ . Для этого применяют метод стохастического градиентного спуска, суть которого демонстрируют следующие уравнения:

$$w_i \leftarrow w_i - \eta \frac{\partial MSE}{\partial w_i}, \quad b_i \leftarrow b_i - \eta \frac{\partial MSE}{\partial b_i},$$

где:  $\eta$  – константа, называемая оценкой обучения, с помощью неё можно контролировать скорость изменения весов и смещений.

Из последних двух выражений видно, что если частная производная от ошибки по весу или смещению положительна, т.е. увеличение этого веса или смещения приведёт к увеличению ошибки, то вес или смещение будет уменьшено на величину частной производной, умноженной на оценку обучения и наоборот.

Таким образом, используя всё время новые данные для обучения или те же самые, подавая их на вход нейронной сети снова и снова, можно добиться того, что веса и смещения станут такими, что ошибка  $MSE$  будет минимальной.

Чтобы не тратить время на создание нейронных сетей с нуля можно воспользоваться доступными бесплатными библиотеками, например, Keras, кото-

рая представляет собой открытую нейросетевую библиотеку, написанную на языке Python. Данная библиотека нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой.

Рассмотрим пример использования Keras для создания нейронной сети, предсказывающей риск заболевания диабетом, на основе анализа восьми параметров:

1. число беременностей;
2. концентрация глюкозы в плазме через 2 часа после введения в пероральном глюкозотолерантном тесте;
3. диастолическое артериальное давление;
4. толщина кожной складки в районе трицепса;
5. концентрация инсулина в сыворотке крови;
6. индекс массы тела;
7. функция, описывающая генетическую предрасположенность к диабету;
8. возраст.

Для обучения сети будет использоваться файл «prima-indians-diabetes.csv» в котором строка за строкой идут через запятую 9 параметров, восемь из которых рассмотрены выше. Последний девятый параметр может принимать значения или «1» или «0». Соответственно, единица означает наличие диабета, а ноль его отсутствие.

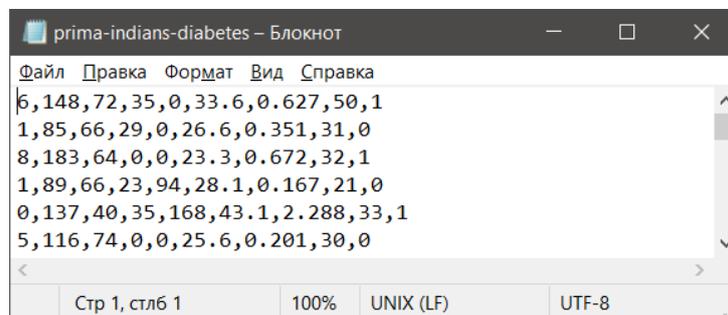


Рисунок 3 – Пример файла с данными для обучения нейронной сети

В листинге 1 приведён код, реализующий нейронную сеть. В строках с 1 по 4 подключаются необходимые библиотеки. В строке 6 происходит чтение файла с обучающими данными для сети и запись их в массив dataset. В строках 7 и 8 из массива dataset выделяются входные сигналы для сети (восемь параметров, описанных выше) и ответы, которые от сети ожидают (девятый параметр, описанный выше). В 9 строке происходит сортировка данных X и Y на те, что будут использоваться для обучения сети и те, что будут использоваться для её тестирования. При этом только 20 % данных для обучения будут использоваться для тестирования сети. В строках с 10 по 15 создаётся непосредственно нейронная сеть, состоящая из 5 слоёв. Вводный слой содержит 12

нейронов и 8 входов, в качестве функции активации у этих нейронов выбрана функция `gelu`. Выходной слой содержит только один нейрон и сигмоиду в качестве функции активации. В 16 строчке происходит компиляция модели сети, после чего её можно обучать. Обучение выполняется в 17 строчке с помощью функции `fit`. При этом число проходов по всем обучаемым данным выбрано 1000. После того, как обучение будет закончено, параметры модели, её веса и смещения будут сохранены в файл «`weights.h5`». Эти данные можно будет впоследствии загрузить и использовать созданную сеть для предсказания заболевания диабетом при других входных данных.

```
1. from keras.models import Sequential
2. from keras.layers import Dense, Dropout
3. from sklearn.model_selection import train_test_split
4. import numpy
5. numpy.random.seed(2)
6. dataset = numpy.loadtxt("prima-indians-diabetes.csv", delimiter=",")
7. X = dataset[:,0:8]
8. Y = dataset[:,8]
9. x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
10. model = Sequential()
11. model.add(Dense(12, input_dim=8, activation='relu'))
12. model.add(Dense(15, activation='relu'))
13. model.add(Dense(8, activation='relu'))
14. model.add(Dense(10, activation='relu'))
15. model.add(Dense(1, activation='sigmoid'))
16. model.compile(loss="binary_crossentropy", optimizer="adam", metrics=['accuracy'])
17. model.fit(x_train, y_train, epochs = 1000, batch_size=20, validation_data=(x_test, y_test))
18. model.save('weights.h5')
```

### Листинг 1 – Код, демонстрирующий создание и обучение нейронной сети с помощью библиотеки Keras

Разобрав приведённый выше код и оценив его сложность, можно сделать вывод о том, что наличие в свободном доступе бесплатных библиотек, например, таких, как Keras, делает практическое применение нейронных сетей доступным для большого числа заинтересованных в этом специалистов, что будет способствовать все большему использованию подобных библиотек для решения актуальных технических задач.

## ЛИТЕРАТУРА

1. **Галушкин, А. И.** Нейронные сети: основы теории / А.И. Галушкин. - Москва : Гор. линия-Телеком, 2012. - 496 с.
2. **Омельяненко, Я.** Эволюционные нейросети на языке Python : практическое руководство / Я. Омельяненко ; пер. с англ. В. С. Яценкова. - Москва : ДМК Пресс, 2020. - 310 с.