9. Quantum-noise randomized data-encryption for WDM fiber-optic networks [Electronic resourse]. – URL: https://www.researchgate.net/publication/232796734_Quantum_cryptography_A_step_towards_global_key_distribution (дата обращения: 08.02.2019).

10. Запущена единственная в СНГ многоузловая квантовая сеть [Электронный ресурс]. – Режим доступа: http://technopark.ifmo.ru/zapushhena-edinstvennaya-v-sng-mnogouzlovaya-kvantovaya-set/ (дата обращения: 08.02. 2019).

11. Квантовый компьютер и квантовая связь [Электронный ресурс]. – Режим доступа: http://www.tadviser.ru/index.php/ (дата обращения: 08.02.2019).

УДК 004.032.26

Соснин А. С., Суслова И. А.

РЕАЛИЗАЦИЯ ЭЛЕМЕНТАРНОЙ НЕЙРОННОЙ СЕТИ

Александр Сергеевич Соснин
Студент магистратуры
salexandr18@gmail.com
Ирина Александровна Суслова
кандидат педагогических наук, доцент
irina.suslova@rsvpu.ru
ФГАОУ ВО «Российский государственный профессионально-педагогический университет», Россия, Екатеринбург

REALIZATION OF ELEMENTARY NEURAL NETWORK

Aleksandr Sergeevich Sosnin
Irina Aleksandrovana Suslova
Russian State Vocation Pedagogical University, Russia, Yekaterinburg

Аннотация*. В статье раскрываются реализация нейронной сети и ее последующее обучение.*

Abstract. *The article describes the implementation of the neural network and its subsequent training.*

Ключевые слова: *нейронная сеть, активационная функция, Python.*

Keywords: *neural network, activation function, Python.*

В данный момент нейронные сети уже прочно вошли в нашу жизнь. Нейронные сети широко применяются для решения широкого диапазона различных задач и активно применяются в тех места где обычные алгоритмические решения не эффективны либо вообще не применимы.

Что такое нейронная сеть?

Анализирую статьи и литературу по данной теме, становится понятно, что при описании нейронных сетей, часто проводятся параллели с человеческим мозгом. Первые попытки создания искусственного интеллекта возникли в 60-х годах. На начальных этапах, исследователи пыталась полностью скопировать структуру мозга человека основ. Именно поэтому и возникла терминология, структура нейронной сети и сами элементы — персептроны.

Нейронную сеть проще описывать сети как математическую функцию, которая отображает заданный вход в желаемый результат, не вникая в подробности.

Нейронные сети состоят из следующих компонентов:

- входной слой, x;
- произвольное количество скрытых слоев;
- выходной слой, ŷ;
- набор весов и смещений между каждым слоем;
- выбор функции активации для каждого скрытого слоя σ; в этой работе используется функция активации Sigmoid.

На диаграмме показана архитектура двухслойной нейронной сети (обратите внимание, что входной уровень обычно исключается при подсчете количества слоев в нейронной сети) (рисунок 1).

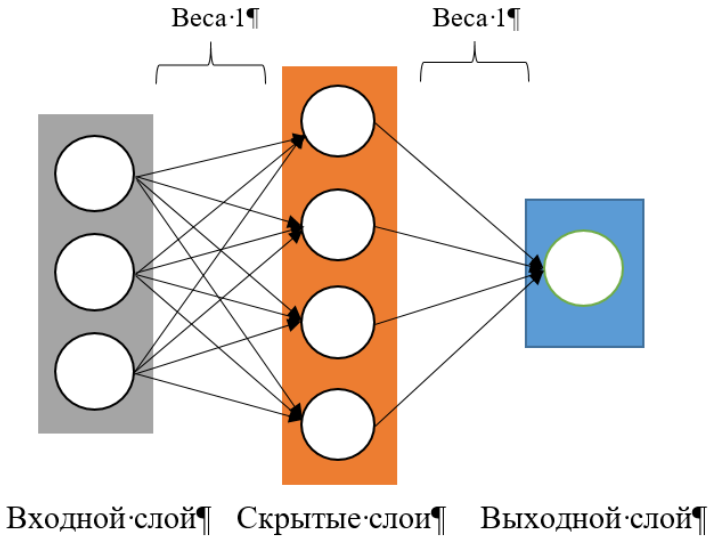

Рисунок 1 — Архитектура двухслойной нейронной

Создание класса Neural Network на Python выглядит следующим образом (рисунок 2).

```
class
NeuralNetwork:
                    def __init__(self, x, y):
                        self.input = x
                        self.weights1 = np.random.rand(self.input.shape[1],4)
                        self.weights2 = np.random.rand(4,1)
                        self.y = y
                        self.output = np.zeros(y.shape)
```

Рисунок 2 — Класс Neural Network

Обучение нейронной сети

Выход $\hat{\mathbf{y}}$ простой двухслойной нейронной сети:

$$\hat{y} = \sigma(W_2\sigma(W_1 x + b_1) + b_2 \quad (1)$$

В приведенном выше уравнении (1), веса W и смещения b являются единственными переменными, которые влияют на выход $\hat{y}$.

Естественно, правильно рассчитанные значения весов и смещений определяют точность предсказаний. Процесс обучение нейронной сети представляет собой тонкую настройку весов и смещений, основываясь на обработке входных данных.

Каждая итерация обучающего процесса включает в себя следующие шаги:

- вычисление прогнозируемого выхода ŷ, называемого прямым распространением.
- обновление весов и смещений, называемых обратным распространением.

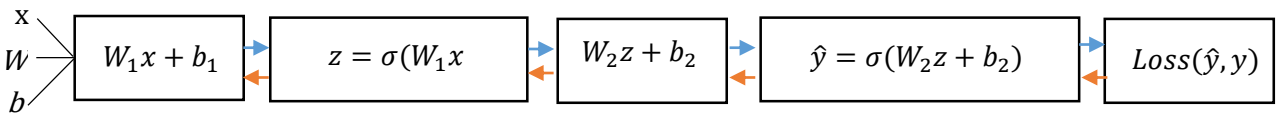

Рисунок 3 — Протекание процесса обучения

Прямое распространение

Как показано на графике (рисунок 3.), прямое распространение — это просто несложное вычисление. Простая 2-слойная нейронная сеть может быть описана следующей формулой (2).

$$\hat{y} = \sigma(W_2\sigma(W_1x + b_1) + b_2 \qquad (2)$$

Функция прямого распространения на Python-е может быть реализована следующим образом (рисунок 4). Стоит заметить, что для упрощения, начальное смещение равно 0.

```
class
NeuralNetwork:
                def __init__(self, x, y):
                    self.input      = x
                    self.weights1   = np.random.rand(self.input.shape[1],4)
                    self.weights2   = np.random.rand(4,1)
                    self.y          = y
                    self.output     = np.zeros(self.y.shape)


                def feedforward(self):
                    self.layer1 = sigmoid(np.dot(self.input, self.weights1))
                    self.output = sigmoid(np.dot(self.layer1, self.weights2))
```

Рисунок 4 — Функция прямого распространения

Оценка «добротности» или точности работы нейронной сети происходит через обратную величину — величину ошибки. Для расчёта оценки величины ошибки применяются функции расчёта ошибки.

Функция ошибки

Существует огромное количество видов функций определения ошибки, выбор конкретной функции происходит исходя из конкретной ситуации. В данной реализации используется функция расчёта ошибки **сумму квадратов ошибок**.

$$Sum - of - Squares\ Error = \sum_{i=1}^{n}(y - \hat{y})^2 \qquad (3)$$

Сумма квадратов ошибокравна среднему значению разницы между каждым полученным и фактическим значением.

Главной целью обучения нейронной сети является найти такой набор весов и смещений, который даст минимальное значение функции ошибки.

Обратное распространение

После получения величины ошибки, требуется выполнить обратное распространение ошибки с целью обновления значений весов и смещения.

Для нахождения значений корректировки весов и смещений, требуется вычислить производную функции ошибки по отношению к весам и смещениям.

Из математического анализа известно, что производная функции — равна тангенсу угла наклона функции.

После вычисления производной, происходит обновление весов и смещения. Это называется градиентным спуском.

Вычислить производную функции потерь по отношению к весам и смещениям невозможно, так как уравнение функции потерь не содержит весов и смещений. Для вычисления производной функции потерь применяется формула (4).

$$Loss(y, \hat{y}) = \sum_{i=1}^{n}(y - \hat{y})^2$$

$$\frac{\partial Loss(y, \hat{y})}{\partial W} = \frac{\partial Loss(y, \hat{y})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z} * \frac{\partial z}{\partial W}, \text{где } z = Wx + b$$

$$= 2(y - \hat{y}) * derivative\ of\ sigmoid\ function * x$$

$$= 2(y - \hat{y}) * z(1 - z) * x \quad (4)$$

Реализация функции обратного распространения (backpropagation) на Python представлена на рисунке 5.

```
def backprop(self):
    d_weights2 = np.dot(self.layer1.T, (2*(self.y - self.output) * sigmoid derivative(self.output)))
    d_weights1 = np.dot(self.input,  (np.dot(2*(self.y - self.output) * sigmoid derivative(self.output), self.weights2.T) * sigmoid derivative(self.layer1)))
    self.weights1 += d_weights1
    self.weights2 += d_weights2
```

Рисунок 5 — Реализация функции обратного распространения

Проверка работы нейросети

После реализации нейронной сети на Python были проведены тесты на примере идеальных весов (таблица 1).

Таблица 1 — Результаты тестовых измерений весов нейронной сети

X1	X2	X3	Y
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

Идеальный набор весов

Тренировка нейронной сети из 1500 итераций. Анализируя график зависимости величины ошибки от количества обучающих интераций (рисунок 6), можно сделать вывод, что что величина ошибки монотонно уменьшается. Это несомненно согласуется с алгоритмом спуска градиента, о котором мы говорили ранее.

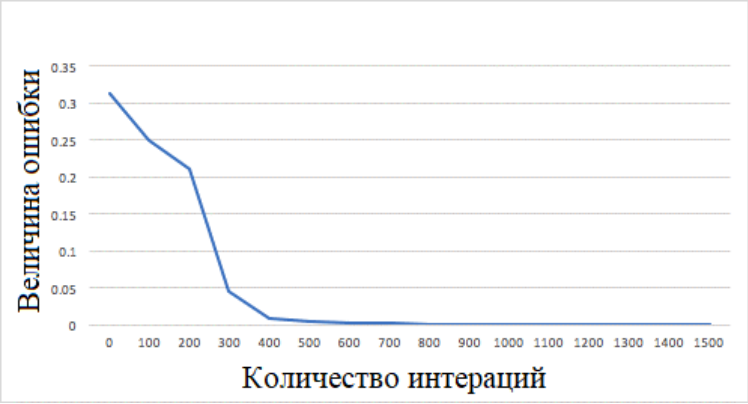

Рисунок 6 — График зависимости величины ошибки от количества обучающих интераций

В таблице 2 представлен результат работы нейронной сети.

Таблица 2 — Результат работы нейронной сети

Точность	Значение Y
0,023	0
0.979	1
0.975	1
0.025	0

Из таблицы 1 можно сделать вывод что алгоритм прямого и обратного распространения показал успешную работу нейронной сети, а результат сходятся с истинными значениями.

Стоит заметить, что имеется небольшая разница между результатами фактическими значениями. Это допустимо, поскольку предотвращает переобучение и позволяет нейронной сети лучше обобщать данные.

Вывод

Мы подробно разобрали процесс реализации элементарной нейронной сети. Применение библиотек глубокого обучения, таких как TensorFlow и Keras, не требуют полного понимания всех процессов и принципов работы нейронной сети и позволяют создавать нейронные сети, несмотря на это, что начинающим разработчикам важно иметь глубокое понимание функционирования нейронных сетей.

Список литературы

1. Функции активации в нейронных сетях [Электронный ресурс]. – URL: http://www.aiportal.ru/articles/neural-networks/activation-function.html (дата обращения: 28.01.2019).

2. Фаустова, К. И. Нейронные сети: применение сегодня и перспективы развития [Электронный ресурс] / К. И. Фаустова // Территория науки. – 2017. – № 4. – URL: https://cyberleninka.ru/article/n/neyronnye-seti-primenenie-segodnya -i-perspektivy-razvitiya (дата обращения: 29.01.2019).

3. Филатова, Т. В. Применение нейронных сетей для аппроксимации данных [Электронный ресурс] / Т. В. Филатова // Вестник Томского государственного университета. – 2004. – № 284. – URL: https://cyberleninka.ru/article/n/primenenie-neyronnyh-setey-dlya-approksimatsii-dannyh (дата обращения: 29.01.2019).

УДК 004.032.26

Соснин А. С., Суслова И. А.

ФУНКЦИИ АКТИВАЦИИ НЕЙРОСЕТИ: СИГМОИДА, ЛИНЕЙНАЯ, СТУПЕНЧАТАЯ, RELU, TAHN

Александр Сергеевич Соснин
Студент магистратуры
salexandr18@gmail.com
Ирина Александровна Суслова
кандидат педагогических наук, доцент
irina.suslova@rsvpu.ru
ФГАОУ ВО «Российский государственный профессионально-педагогический университет», Россия, Екатеринбург

FUNCTIONS OF NEURAL NET ACTIVATION: SIGMOID, LINEAR, STEP, RELU, TAHN

Aleksandr Sergeevich Sosnin
Irina Aleksandrovana Suslova
Russian State Vocation Pedagogical University, Russia, Yekaterinburg

Аннотация*. В статье раскрываются основные активационные функции, рассматриваются их отличительные особенности, присущие им достоинства и недостатки.*

Abstract*. The article reveals the main activation functions, highlights their distinctive features, their inherent strengths and weaknesses.*