

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Прикладные интеллектуальные системы и экспертные
системы»

Методы искусственного интеллекта. Искусственные нейронные сети.

Студент

Лобов М.Ю.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Доцент, к.п.н.

Липецк 2022 г.

Краткая теоретическая справка

Нейронная сеть — это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса машинного обучения, называемый глубоким обучением, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг. Он создает адаптивную систему, с помощью которой компьютеры учатся на своих ошибках и постоянно совершенствуются. Таким образом, искусственные нейронные сети пытаются решать сложные задачи, такие как резюмирование документов или распознавание лиц, с более высокой точностью.

Нейронные сети помогают компьютерам принимать разумные решения с ограниченным участием человека. Они могут изучать и моделировать отношения между нелинейными и сложными входными и выходными данными. Например, нейронные сети могут выполнять следующие задачи.

Архитектура нейронных сетей повторяет структуру человеческого мозга. Клетки человеческого мозга, называемые нейронами, образуют сложную сеть с высокой степенью взаимосвязи и посылают друг другу электрические сигналы, помогая людям обрабатывать информацию. Точно так же искусственная нейронная сеть состоит из искусственных нейронов, которые взаимодействуют для решения проблем. Искусственные нейроны — это программные модули, называемые узлами, а искусственные нейронные сети — это программы или алгоритмы, которые используют вычислительные системы для выполнения математических вычислений.

Базовая нейронная сеть содержит три слоя взаимосвязанных искусственных нейронов:

1) Входной слой

Информация из внешнего мира поступает в искусственную нейронную сеть из входного слоя. Входные узлы обрабатывают данные, анализируют или классифицируют их и передают на следующий слой.

2) Скрытый слой

Скрытые слои получают входные данные от входного слоя или других скрытых слоев. Искусственные нейронные сети могут иметь большое количество скрытых слоев. Каждый скрытый слой анализирует выходные данные предыдущего слоя, обрабатывает их и передает на следующий слой.

3) Выходной слой

Выходной слой дает окончательный результат обработки всех данных искусственной нейронной сетью. Он может иметь один или несколько узлов. Например, при решении задачи двоичной классификации (да/нет) выходной слой будет иметь один выходной узел, который даст результат «1» или «0». Однако в случае множественной классификации выходной слой может состоять из более чем одного выходного узла.

Искусственные нейронные сети можно классифицировать по тому, как данные передаются от входного узла к выходному узлу. Ниже приведены несколько примеров.

1) Нейронные сети прямого распространения

Нейронные сети прямого распространения обрабатывают данные в одном направлении, от входного узла к выходному узлу. Каждый узел одного слоя связан с каждым узлом следующего слоя. Нейронные сети прямого распространения используют процесс обратной связи для улучшения прогнозов с течением времени.

2) Алгоритм обратного распространения

Искусственные нейронные сети постоянно обучаются, используя корректирующие циклы обратной связи для улучшения своей прогностической аналитики. Проще говоря, речь идет о данных, протекающих от входного узла к выходному узлу по множеству различных путей в нейронной сети. Правильным является только один путь, который сопоставляет входной узел с правильным выходным узлом.

3) Свёрточные нейронные сети

Скрытые слои в свёрточных нейронных сетях выполняют определенные математические функции (например, суммирование или фильтрацию), называемые свертками. Они очень полезны для классификации изображений, поскольку могут извлекать из них соответствующие признаки, полезные для распознавания и классификации. Новую форму легче обрабатывать без потери функций, которые имеют решающее значение для правильного предположения. Каждый скрытый слой извлекает и обрабатывает различные характеристики изображения: границы, цвет и глубину.

Обучение нейронной сети — это процесс обучения нейронной сети выполнению задачи. Нейронные сети обучаются путем первичной обработки нескольких больших наборов размеченных или размеченных данных. На основе этих примеров сети могут более точно обрабатывать неизвестные входные данные.

Контролируемое обучение

При контролируемом обучении специалисты по работе с данными предлагают искусственным нейронным сетям помеченные наборы данных, которые заранее дают правильный ответ. Например, сеть глубокого обучения, обучающаяся распознаванию лиц, обрабатывает сотни тысяч изображений человеческих лиц с различными терминами, связанными с этническим происхождением, страной или эмоциями, описывающими каждое изображение.

Нейронная сеть медленно накапливает знания из этих наборов данных, которые заранее дают правильный ответ. После обучения сеть начинает делать предположения об этническом происхождении или эмоциях нового изображения человеческого лица, которое она никогда раньше не обрабатывала.

Реализация на Python

```
from numpy import exp, array, random, dot

class NeuralNetwork():
    def __init__(self):
        # Устанавливаем воспроизводимость рандома
        random.seed(1)

        # Моделируем нейрон с тремя входами и одним выходом
        # Заполним матрицу 3 x 1 случайными весами от -1 до 1
        # и средним 0.
        self.synaptic_weights = 2 * random.random((3, 1)) - 1

    # Функция, описывающая S-образную кривую
    def __sigmoid(self, x):
        return 1 / (1 + exp(-x))

    # Производная функции
    def __sigmoid_derivative(self, x):
        return x * (1 - x)

    # Обучение нейронной сети
    def train(self, training_set_inputs, training_set_outputs,
number_of_training_iterations):
        for iteration in range(number_of_training_iterations):
            # Пропускаем обучающие данные через нейросеть
            output = self.think(training_set_inputs)

            # Вычисляем ошибку
            error = training_set_outputs - output

            # Умножаем ошибку на вход и на производную функции
            adjustment = dot(training_set_inputs.T, error *
self.__sigmoid_derivative(output))

            # Регулируем нейронную сеть
            self.synaptic_weights += adjustment
```

```

# Функция действия нейрона
def think(self, inputs):
    return self.__sigmoid(dot(inputs, self.synaptic_weights))

if __name__ == "__main__":

    # Инициализируем нейронную сеть
    neural_network = NeuralNetwork()

    print ("Random starting synaptic weights: ")
    print (neural_network.synaptic_weights)

    # Обучающие данные
    training_set_inputs = array([[0, 0, 1], [1, 1, 1], [1, 0, 1], [0, 1, 1]])
    training_set_outputs = array([[0, 1, 1, 0]]).T

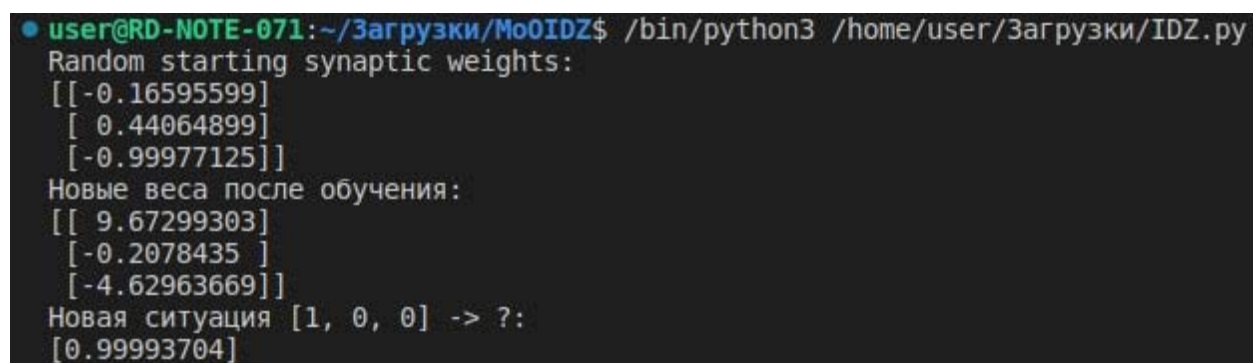
    # Обучение нейронной сети
    neural_network.train(training_set_inputs, training_set_outputs, 10000)

    print ("Новые веса после обучения: ")
    print (neural_network.synaptic_weights)

    # Тестирование работы нейронной сети в новой ситуации
    print ("Новая ситуация [1, 0, 0] -> ?: ")
    print (neural_network.think(array([1, 0, 0])))

```

Результат выполнения программы



```

user@RD-NOTE-071:~/Загрузки/Мо0IDZ$ /bin/python3 /home/user/Загрузки/IDZ.py
Random starting synaptic weights:
[[-0.16595599]
 [ 0.44064899]
 [-0.99977125]]
Новые веса после обучения:
[[ 9.67299303]
 [-0.2078435 ]
 [-4.62963669]]
Новая ситуация [1, 0, 0] -> ?:
[0.99993704]

```

Рисунок 1 – Результат выполнения программы

Вывод

В ходе выполнения индивидуального домашнего задания были изучены теоретические основы нейросетей и реализована модель искусственной нейронной сети на языке Python.

Для этого использовалась библиотека `numpy`, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

Подводя итоги, хочется отметить, что искусственные нейронные сети в настоящее время являются полезным инструментом, позволяющим решать нетривиальные задачи, и продолжают активно использоваться в различных областях деятельности человека.