

Автор:

Дмитриев Егор Андреевич

студент

ФГАОУ ВО «Самарский национальный исследовательский
университет им. академика С.П. Королева»
г. Самара, Самарская область

РЕАЛИЗАЦИЯ МНОГОСЛОЙНОЙ СЕТИ ПЕРСЕПТРОНА

Аннотация: в статье рассматривается способ реализации многослойной сети персептрана.

Ключевые слова: градиент, градиентный спуск, машинное обучение.

Основные определения

Определение 1. Градиент – вектор, указывающий направление наибольшего возрастания некоторой величины от одной точки пространства к другой (скалярного поля).

Определение 2. Градиентный спуск – метод нахождения локально максимума (минимума), который заключается в движении по направлению градиента (антиградиента).

Определение 3. Машинное обучение – нахождение отображения, в частности алгоритма классификации, который строится по множеству, называемому обучаемой выборкой, а качество обучения проверяется по множеству, называемому тестовой выборкой.

Введение

На сегодняшний момент нейронные сети широко применяют в машинном обучении. Существует большое разнообразие нейронных сетей, которые хорошо работают на определенных данных и для конкретной задачи. В данной работе рассматривается наиболее общая модель нейронной сети. Будет приведена реализация на языке Python.

Постановка проблемы

Необходимо реализовать нейронную сеть многослойного персептрана

Описание нейронной сети

Нейронная сеть представляет собой отображение из пространства одной размерности, в пространство другой размерности (возможно, той же самой размерности). В данной работе рассматривается нейронная сеть, которая использовалась для задачи классификации. Была использована база данных mnist, в которой содержаться 60000 тысяч рукописных изображений цифр от 0 до 9 для обучения, и 10000 тысяч изображения для тестирования.

Модель нейронной сети изображена на рисунке 1.

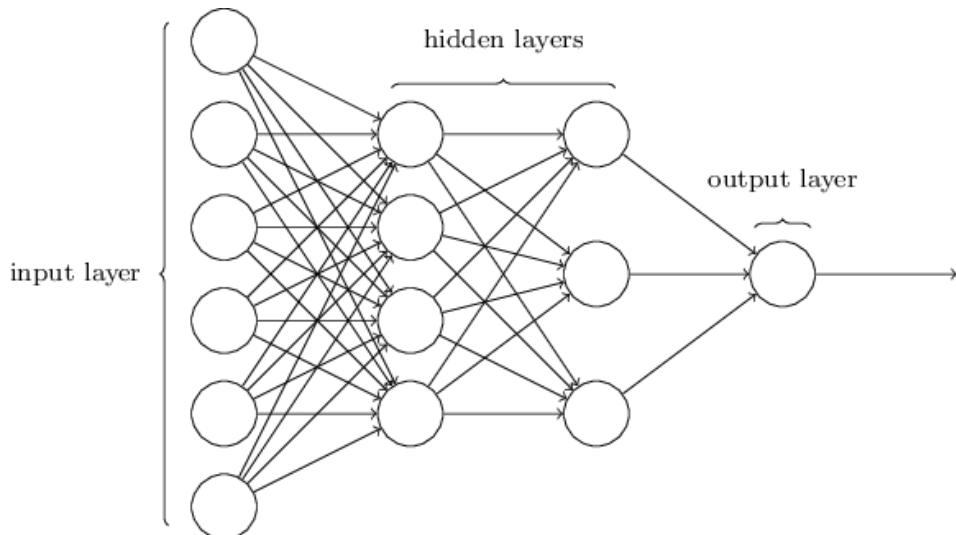


Рис. 1. Сеть многослойного персептрана

Использовался метод обучения обратного распространения ошибки, в основе которого лежит метод градиентного спуска, но рассчитывался не полный градиент, а стохастический. В качестве функции активации использовалась функция сигмоида.

Программная реализация

```

class NeuralNetwork:
    def __init__(self, sizes):
        self.num_layers = len(sizes)
        self.sizes = sizes
        self.biases = [np.random.randn(y, 1) for y in sizes[1:]]
        self.weights = [np.random.randn(y, x)/np.sqrt(x)
                        for x, y in zip(sizes[:-1], sizes[1:])]
        self.learnrate = 0.01
        self.number_of_epoch = 20
        self.mini_batch_length = 10
        self.reg = 5

```

Рис. 2

В конструкторе использованы поля для количества слоев и нейронов в каждом слое, параметров обучения, коэффициента обучения, количество эпох, то есть полных прохождений по выборке, коэффициент регуляризации L1, количество элементов для подсчета стохастического градиента.

```

def feedforward(self, init):
    outs = [np.zeros((y, 1)) for y in self.sizes]
    args = [np.zeros((y, 1)) for y in self.sizes]
    outs[0] = init
    i = 1
    for w, b in zip(self.weights, self.biases):
        outs[i] = sigmoid(np.dot(w, outs[i - 1]) + b)
        args[i] = (np.dot(w, outs[i - 1]) + b)
        i += 1
    return outs, args

def backprop(self, x, y):
    outs, args = self.feedforward(x)
    errors = [np.zeros(b.shape) for b in self.biases]
    errors[self.num_layers - 2] = outs[self.num_layers - 1] - y
    i = self.num_layers - 2
    for w, z in zip(reversed(self.weights[1:]), reversed(args[1:-1])):
        errors[i - 1] = np.multiply(np.dot(w.T, errors[i]), sigmoid_prime(z))
        i -= 1
    nabla_w = [np.dot(error, out.T) for error, out in zip(reversed(errors), reversed(outs[:-1]))]
    nabla_b = errors
    return reversed(nabla_w), nabla_b

```

Рис. 3

Реализация прямого распространения сигнала и обратного распространения ошибки.

```
Epoch 13:9107/10000
Epoch 14:9108/10000
Epoch 15:9113/10000
Epoch 16:9100/10000
Epoch 17:9116/10000
Epoch 18:9124/10000
Epoch 19:9114/10000
```

Рис. 4

Достигнутый результат за 20 эпох.

Заключение

В данной работе была приведена реализация многослойной сети персептрана.

Список литературы

1. Минский М. Perceptrons / М. Минский, С. Пейперт. – М.: Мир, 1971.
2. Хайкин С. Нейронные сети. – М.: Вильям, 2008.