



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Домашнее задание № 1

Реализация однослойного персептрона

ПО КУРСУ:

«Теория искусственных нейронных сетей»

Студент *Караник А. А.*

Преподаватель *Каганов Ю. Т.*

Москва, 2024 г.

ОГЛАВЛЕНИЕ

1. Цель.....	3
2. Постановка задачи.....	3
3. Практическая реализация	4
3. Вывод.....	7

1. Цель

Целью данной работы является изучение принципов работы однослойного персептрона, его реализации на языке программирования высокого уровня, а также исследование влияния различных функций активации (линейная, сигмоида, гиперболический тангенс и ReLU) на процесс обучения модели.

2. Постановка задачи

1. Реализовать однослойный персептрон с возможностью использования различных функций активации: линейной, сигмоиды, гиперболического тангенса и ReLU.
2. Сгенерировать искусственные данные для цифр и букв, каждая из которых представлена в виде матрицы 5x4.
3. Обучить персептрон на этих данных, используя различные функции активации.
4. Зафиксировать и визуализировать зависимость функции потерь от количества эпох обучения для каждой функции активации в виде графиков.
5. Провести сравнительный анализ работы персептрона при различных функциях активации, сделав выводы на основе полученных результатов.

3. Практическая реализация

Практическая реализация направлена на обучение однослойного персептрона распознавать буквы русского алфавита и цифры, представленные в виде двоичных массивов, где каждая цифра и буква имеет размер поля 5x4 (то есть 20 элементов). Эти массивы состоят из 0 и 1, где 1 обозначает закрашенную ячейку, а 0 — пустую. Каждому символу (цифра или буква) соответствует уникальная двоичная матрица.

Этапы реализации:

1. Генерация данных для символов:

- Используется функция `generate_data()`, которая создает набор данных для определённых букв русского алфавита, а также цифр от 0 до 9. Каждый символ представлен в виде одномерного массива длиной, который описывает его форму в виде матрицы 5x4.

2. Определение функций активации:

- В коде реализуются различные функции активации: линейная, сигмоида, гиперболический тангенс и ReLU. Эти функции позволяют контролировать выходное значение нейрона в зависимости от входного сигнала. Каждая функция активации имеет свою производную, которая используется для расчёта градиента во время обучения.

```

def linear(self, x):
    return x
def sigmoid(self, x):
    return 1 / (1 + np.exp(-x))
def tanh(self, x):
    return np.tanh(x)
def relu(self, x):
    return np.maximum(0, x)

def linear_derivative(self, x):
    return np.ones_like(x)
def sigmoid_derivative(self, x):
    return x * (1 - x)
def tanh_derivative(self, x):
    return 1 - np.tanh(x) ** 2
def relu_derivative(self, x):
    return np.where(x > 0, 1, 0)

```

3. Обучение персептрона:

- Для каждой функции активации создаётся экземпляр однослойного персептрона. Персептрон обучается на наборе данных (буквах и цифрах) в течение 1000 эпох.

```

perceptron = Perceptron(input_size, output_size, activation=activation)
perceptron.train(data, labels, n_epochs=1000)

```

- В процессе обучения происходит расчет ошибки для каждого символа, которая затем уменьшается с помощью алгоритма градиентного спуска. Ошибка накапливается и сохраняется для каждой эпохи, чтобы потом можно было проанализировать прогресс обучения.

```

def mse_derivative(self, output, target):
    return 2 * (output - target)
def train(self, X, y, n_epochs):
    for epoch in range(n_epochs):
        total_loss = 0
        correct_predictions = 0
        for i in range(len(X)):
            x = X[i].flatten()
            target = np.zeros(self.output_size)
            target[y[i]] = 1

            output = self.activate(np.dot(x, self.weights) + self.bias)

            error = (target - output) ** 2
            total_loss += np.sum(error)
            mse_grad = self.mse_derivative(output, target)
            adjustment = self.learning_rate * mse_grad * self.activate_derivative(output)

            self.weights -= np.outer(x, adjustment)
            self.bias -= adjustment

            if np.argmax(output) == y[i]:
                correct_predictions += 1

        self.losses.append(total_loss / len(X))
        self accuracies.append(correct_predictions / len(X))

```

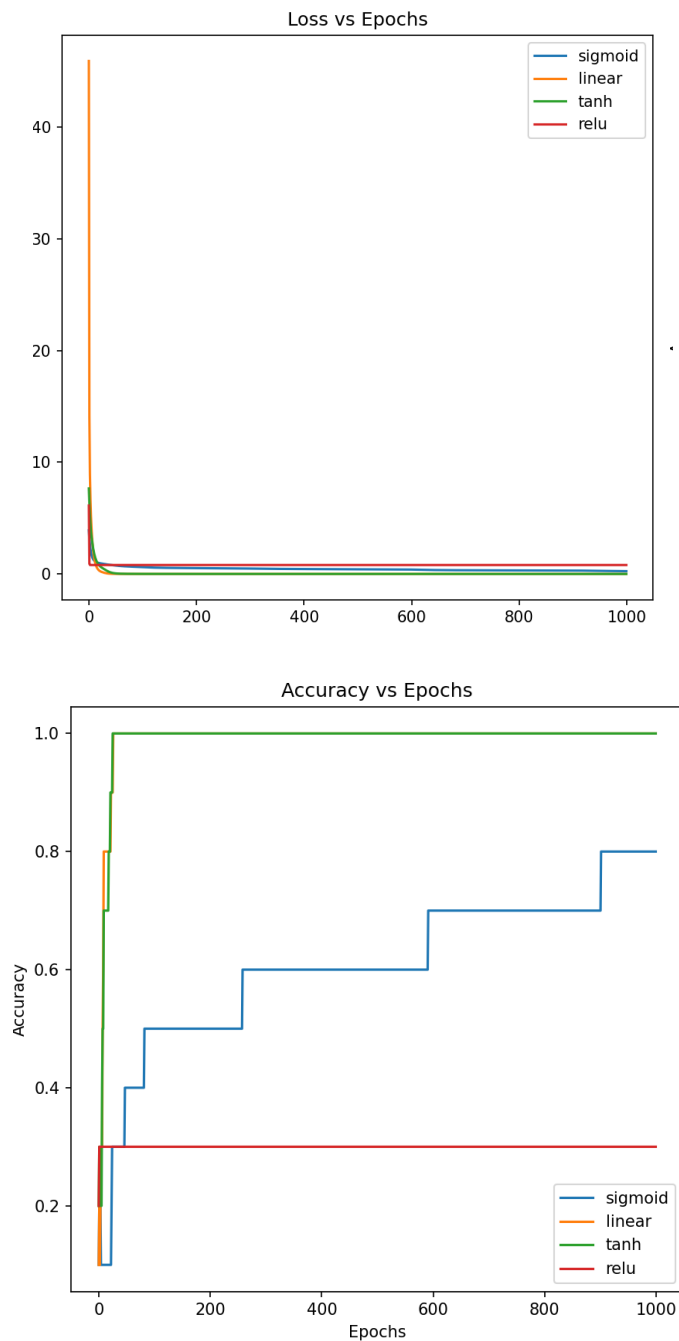
4. Анализ ошибки:

- Ошибка, рассчитанная для каждого символа, сохраняется для каждой эпохи. Итоговая ошибка для каждой функции активации (линейной, сигмоида, гиперболический тангенс и ReLU) собирается в отдельный список.
- После обучения по каждой функции активации, средняя ошибка для всех символов записывается в массив для последующего построения графиков.

5. Построение графиков:

- После завершения обучения для каждой функции активации, строятся графики, отображающие зависимости потерь и точности от числа эпох. Это помогает наглядно увидеть, насколько быстро модель обучается и как различные функции активации влияют на процесс обучения.
- На графике отображаются кривые потерь и точности для всех функций активации (линейной, сигмоиды, гиперболического тангенса и ReLU).

Результаты экспериментов:



3. Вывод

В ходе работы была реализована модель однослойного персептрона для распознавания букв русского алфавита и цифр, используя различные функции активации: линейную, сигмоиду, гиперболический тангенс и ReLU.