Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Приложение нейросетевых алгоритмов**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. Э. Айрапетов

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Шиян

**Цель работы:** получение знаний и практических навыков построения однослойных нейронных сетей.

**Задача 1**. Пусть имеется функция XOR, в которой две точки {(0, 0), (1, 1)} принадлежат к одному классу, а две другие точки {(1, 0), (0, 1)} – к другому. Покажите, как разделить два этих класса, используя функцию активации ReLU.

**Описание решения**. Для начала взглянем на наши данные.

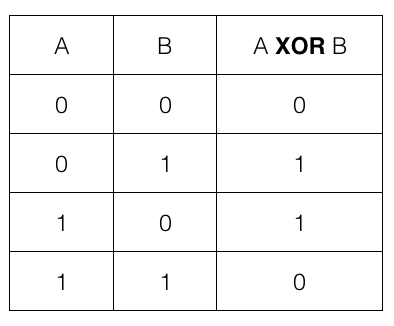


Рисунок 1 – Таблица XOR

Точки можно разделить по их значению в результате выполнения функции XOR (получатся классы 0 и 1). Для ReLU неважно какое именно значение будет у второго класса (главное, что положительное), но для определенности возьмём его равным 1. В функцию ReLU значение будет передаваться в следующем виде:

где w – веса, x – входы, b – порог. Для вычисления весов и порога составим систему уравнений в соответствии с нашими входными данными.

Интерпретировать это можно следующим образом:

Тогда w­1 = w2 = -1, b = 2. Существуют также и другие решения, но главное условие: w1 + w2 <= -b

**Задача 2.** Пусть имеется двухмерный набор данных, в котором все точки с x1 > x2 принадлежат к положительному классу, а все точки с x1 < x2 к отрицательному. Разделителем для этих двух классов является линейная гиперплоскость (прямая линия), определяемая уравнением x1 − x2 = 0. Создайте набор тренировочных данных с 20 точками, сгенерированными случайным образом в положительном квадранте единичного квадрата. Снабдите каждую точку меткой, указывающей на то, превышает или не превышает её первая координата x1 вторую координату x2.

А. Реализуйте алгоритм перцептрона, обучите его на полученных выше 20 точках и протестируйте его точность на 1000 точках, случайно сгенерированных в единичном квадрате. Используйте для генерирования тестовых точек ту же процедуру, что и для тренировочных.

Б. Замените критерий перцептрона на нейрон типа адалайн (рассмотреть дискретный случай) при реализации тренировки и повторите определение точности вычислений на тех же тестовых точках, которые использовали перед этим. Удалось ли вам в каком-то из способов получить лучшую точность? Как вы считаете, в каком случае классификация тех же 1000 тестовых точек не изменится значительно, если использовать другой набор из 20 тренировочных точек?

**Описание решения.** Для начала необходимо сформировать набор тренировочных и тестовых данных.

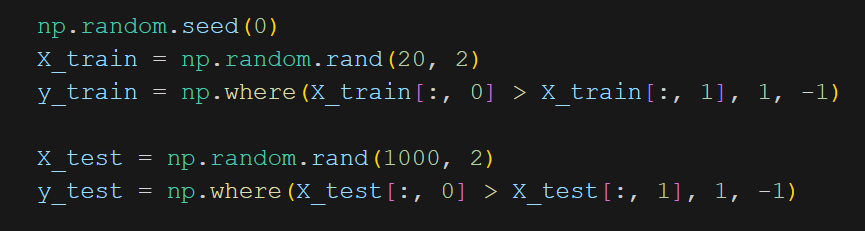


Рисунок 2 – Формирование данных

Далее реализуем классы, отвечающие за обучение наших нейронов. Для перцептрона коррекция весов происходит по формуле:

Реализация класса перцептрон представлена на рисунке 3.

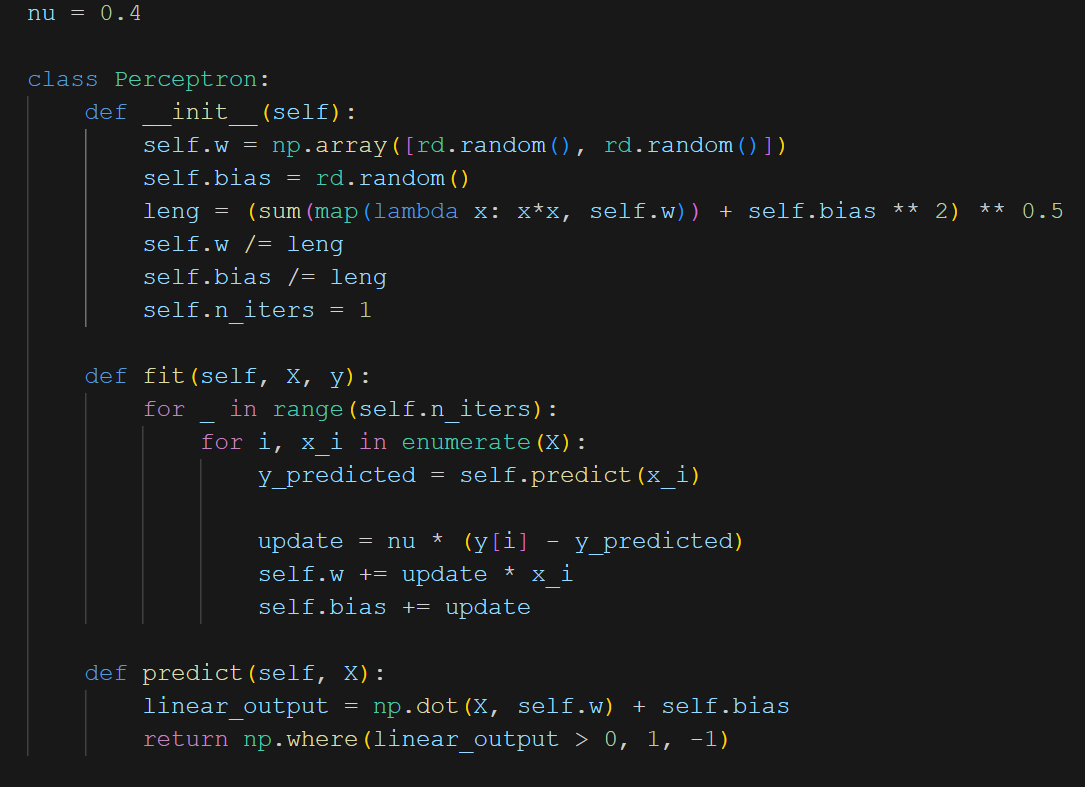


Рисунок 3 – Класс перцептрон

В случае с нейроном типа адалайн при коррекции весов мы считаем не выход нейрона, а сумму взвешенных сигналов. Т. е. для коррекции весов справедлива формула:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Метод обучения для адалайна

При тестировании оба типа показали себя вполне способными решать поставленную задачу и нельзя наверняка сказать кто из них лучше справляется, так как точность варьировалась от 0.7 до 0.99 при разных запусках. При использовании других 20 точек для обучения классификация может остаться измениться в лучшую или худшую сторону в зависимости от репрезентативности выборки и наличия шума в ней. Если другой набор обучающих данных имеет схожую репрезентативность и в нем столько же шума, то на классификацию это никак не повлияет.

**Задача 3**. Требуется разработать и исследовать нейронную сеть обратного распространения, предназначенную для распознавания образов.

Даны в виде матрицы 3 х 3 4 латинские буквы X, Y, L, I.

Требуется:

1. Построить и обучить нейронную сеть, которая могла бы решать задачу распознавания символов.

2. Произвести тестирование нейронной сети при добавлении шума.

**Описание решения.** Реализуем нейросеть с 1 открытым слоем и функцией активации гиперболический тангенс. В цикле по эпохам для каждого обучающего примера прогоним его по всей сети и посчитаем ошибку. Затем используя формулы для коррекции весов исправим веса каждого нейрона.

Было сформировано 20 тренировочных примеров и 8 тестовых. И в тренировочных, и в тестовых данных был привнесен шум для повышения обобщающей способности нейросети.

Экспериментальным образом была выявлена примерно оптимальная архитектура и параметры обучения: архитектура – 9:3:4, количество эпох – 1000, скорость обучения – 0.004. В результате обучения максимальная точность может достигать 0.85 – 1.0.

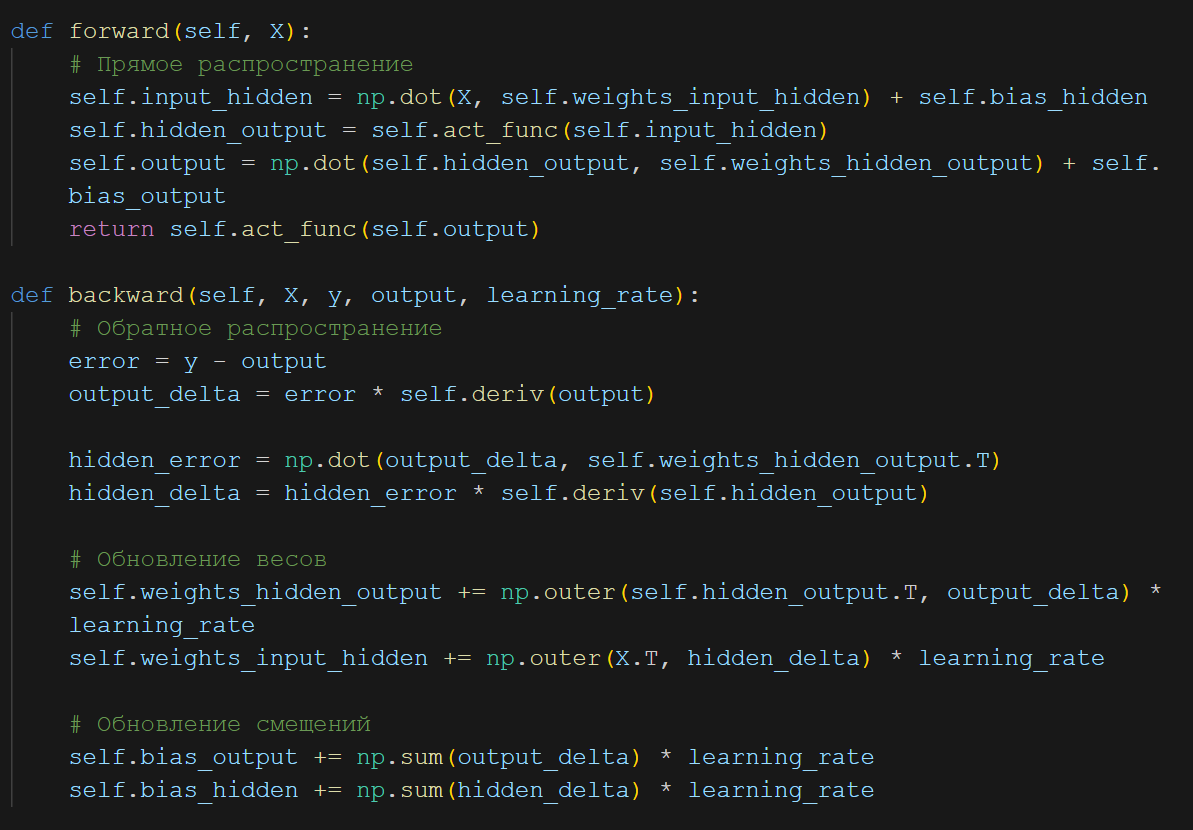


Рисунок 5 – Реализация прямого и обратного прохода

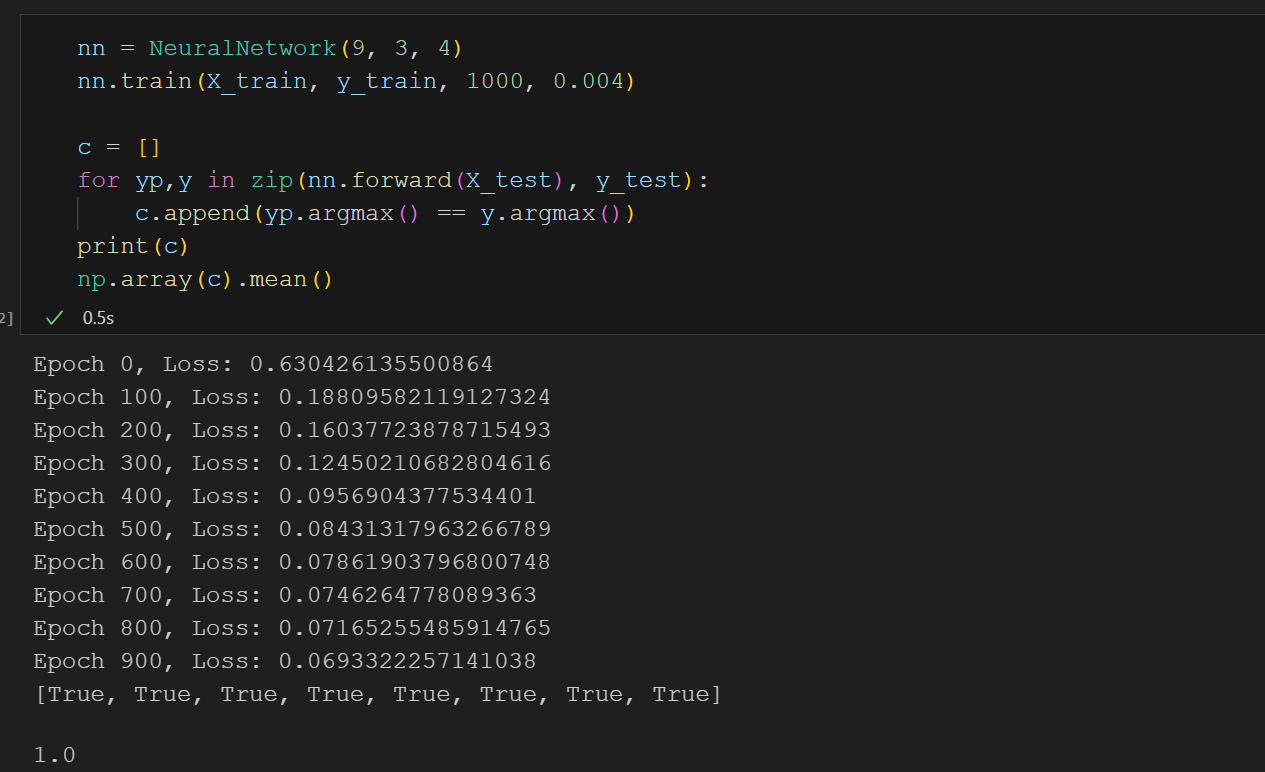


Рисунок 6 – Тестирование результатов обучения

**Вывод:** По результатам выполнения лабораторной работы были рассмотрены нейроны типа адалайн и перцептрон, их плюсы и минусы, а также обучение по правилу обратного распространения ошибки. Были реализованы нейронные сети, предназначенные для классификации двухкомпонентных векторов и классификации букв.