|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине:

Введение в искусственный интелект

|  |  |  |
| --- | --- | --- |
| Студент |  | Абидоков Рашид Ширамбиевич |
| Группа |  | РК6-11М |
| Вариант |  | 1-2 |
| Тема лабораторной работы |  | Программирование искусственного нейрона |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Абидоков Р. Ш.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Федорук В. Г.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2020 г.*

Оглавление

[Задание на лабораторную работу 3](#_Toc58181827)

[Описание входных данных 3](#_Toc58181828)

[Описание используемой модели 4](#_Toc58181829)

[Описание программной реализации 5](#_Toc58181830)

[Результаты обучения 6](#_Toc58181831)

# Задание на лабораторную работу

**Цель лабораторной работы** – создание программы, реализующей искусственный нейрон; разработка процедуры обучения нейрона; использование полученных результатов для решения тестовых задач классификации и аппроксимации

**Тип нейрона** – персептрон

**Вариант обучающих данных** на Рис.1

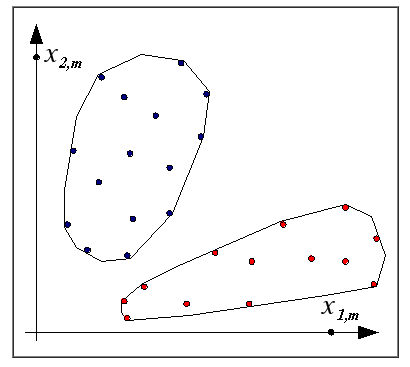


Рис 1.

# Описание входных данных

Были взяты точки, распределенные на плоскости в соответствии с Рис. 1. Координаты точек и их класс приведены в таблице ниже. Из них был сформирован входной файл points.txt

Табл. 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **класс** |
| 1 | 7 | 2 | 0 |
| 2 | 3 | 10 | 1 |
| 3 | 12 | 3 | 0 |
| 4 | 6 | 5 | 1 |
| 5 | 14 | 0 | 0 |
| 6 | 6 | 9 | 1 |
| 7 | 17 | -2 | 0 |
| 8 | 8 | 13 | 1 |
| 9 | 16 | 10 | 1 |

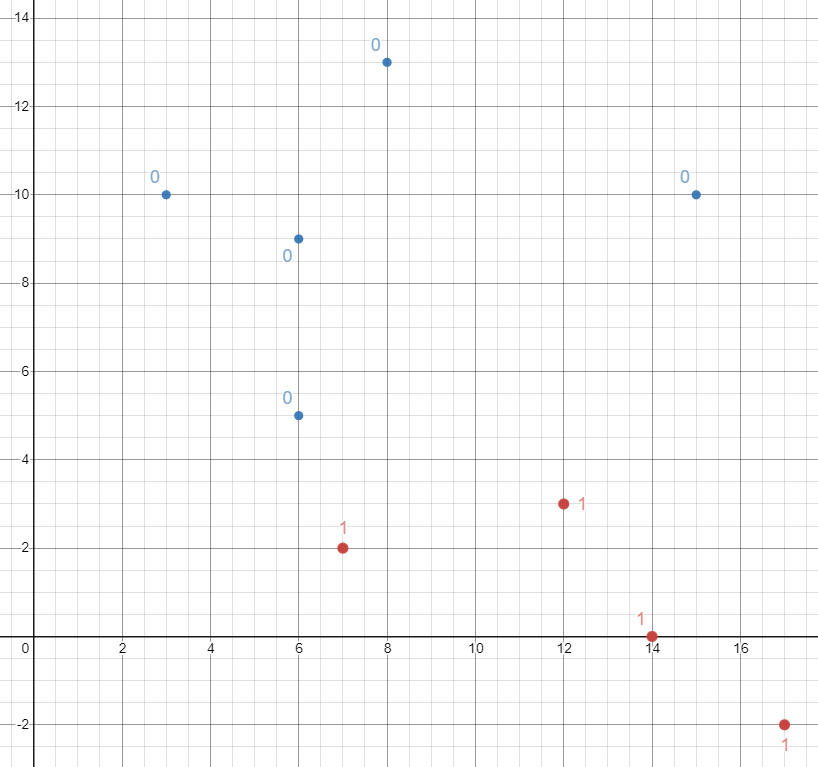


Рис 2. Принятое распределение точек

# Описание используемой модели

Перцептрон – это нейрон со ступенчатой функцией активации, т.е.

Где номер объекта обучающей выборки, номер параметра,

Обучение нейрона сводится к подбору весов путем минимизации среднеквадратичной ошибки на обучающей выборке

Где количество объектов обучающей выборки, класс i-го объекта

В силу дискретности функции активации и, как следствие, невозможности брать производные функции ошибки, невозможно использование методов оптимизации выше нулевого порядка, таких как, например, градиентные методы.

Поэтому обучение производилось с помощью правила персептрона:

1. Выбираются начальные значения весов
2. Для каждой обучающей пары выполняется ряд циклов уточнений входных весов:

Вычисляется

Если , то

Если , то

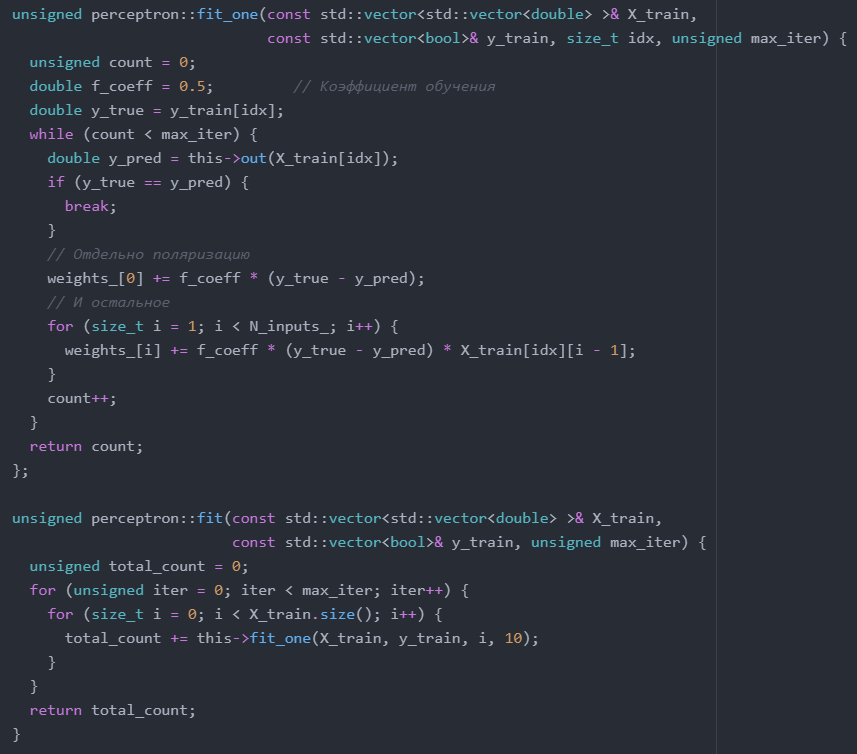
Если или , то цикл прекращается

1. Повторяем циклические проходы по обучающей выборке до тех пор, пока решение не сойдется либо пока не будет превышено максимальное количество итераций

# Описание программной реализации

Программная реализация выполнена на языке C++ с использованием компилятора gcc. Считываются координаты и классы точек из входного txt-файла, задается начальное приближение весов, нейрон обучается по алгоритму, описанному выше, после чего выводятся веса после обучения и метки классов

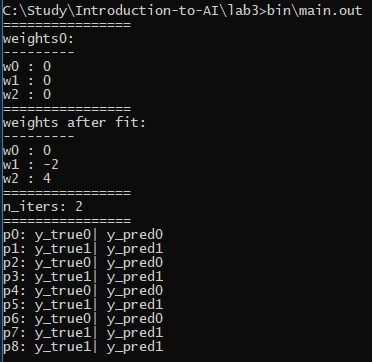
Листинг 1. Функции обучения класса нейрона

**

# Результаты обучения

Обучение производилось с двумя различными начальными значениями весов. В первом случае все веса принимались нулевыми – при таком начальном приближении решение сошлось за два шага

Листинг 2. Вывод программы при нулевом начальном приближении



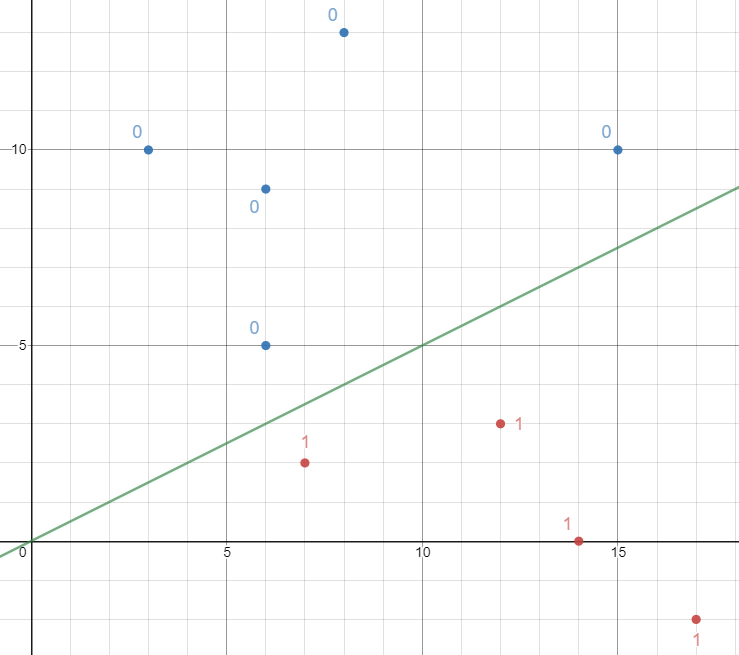


Рис 3. Прямая, соответствующая полученным весам

Табл. 2

|  |  |  |  |
| --- | --- | --- | --- |
| N шага | 0 | 1 | 2 |
|  | 0 | -0.5 | 0 |
|  | 0 | -3.5 | -2 |
|  | 0 | 1 | 4 |

Во втором случае в качестве начальных весов были приняты значения  
(-60, -9.5, -3) – при таких весах разделяющая прямая выглядит так, как показано на Рис. 4. Т.е. приближение можно считать плохим

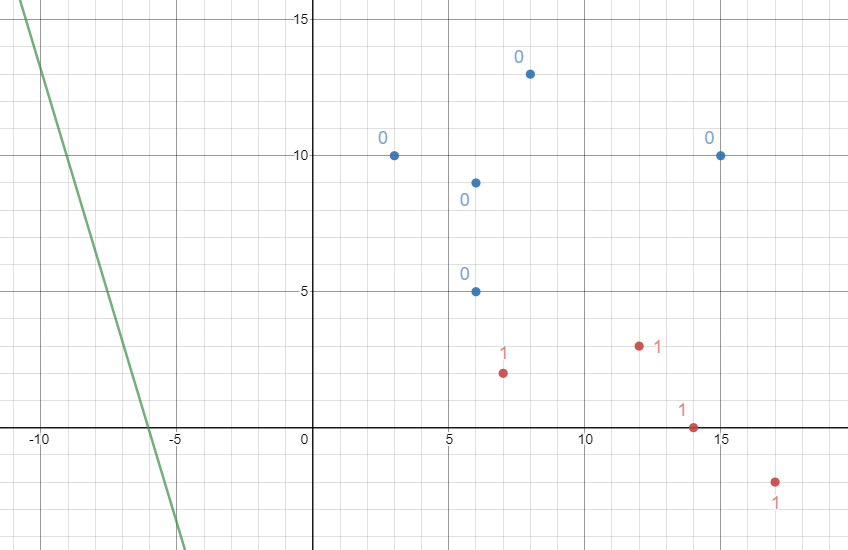


Рис 4. Прямая, соответствующая "плохому" нач. приближению

В этом случае решение сошлось за 4 итерации

Листинг 3. Вывод программы при плохом начальном приближении

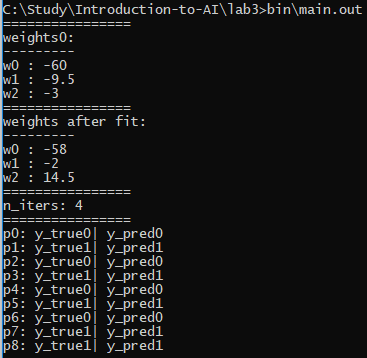


Табл. 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N шага | 0 | 1 | 2 | 3 | 4 |
|  | -60 | -59.5 | -59 | -58.5 | -58 |
|  | -9.5 | -8 | -6.5 | -5 | -2 |
|  | -3 | 2 | 7 | 12 | 14.5 |

Итоговая разделяющая прямая выглядит аналогично Рис. 3