****

**Московский авиационный институт**

**(Национальный исследовательский университет)**

**Институт № 3**

**Кафедра 311**

**Интеллектуальные системы**

**Лабораторная работа № 1**

**«Перцептроны»**

**Выполнил студент  
Плотников Антон Сергеевич**

**Группа М3З-301-БК**

**Дата 25.04.2021 г.**

**Принял преподаватель  
Кос Оксана Игоревна**

Оглавление

[Цель лабораторной работы 3](#_Toc105774989)

[Глава 1. Перцептрон 4](#_Toc105774990)

[Глава 2. Классификация перцептронов 5](#_Toc105774991)

[Глава 3. Принцип работы перцептрона 6](#_Toc105774992)

[Глава 4. Реализация программы 8](#_Toc105774993)

[Итоги лабораторной работы 10](#_Toc105774994)

[Список литературы 11](#_Toc105774995)

# Цель лабораторной работы

Изучить:

1. Принцип работы простейших перцептронов;
2. Реализовать один из них.

# Глава 1. Перцептрон

Перцептрон, или персептрон — математическая или компьютерная модель восприятия информации мозгом (кибернетическая модель мозга), предложенная Фрэнком Розенблаттом в 1958 году и впервые реализованная в виде электронной машины «Марк-1» в 1960 году.

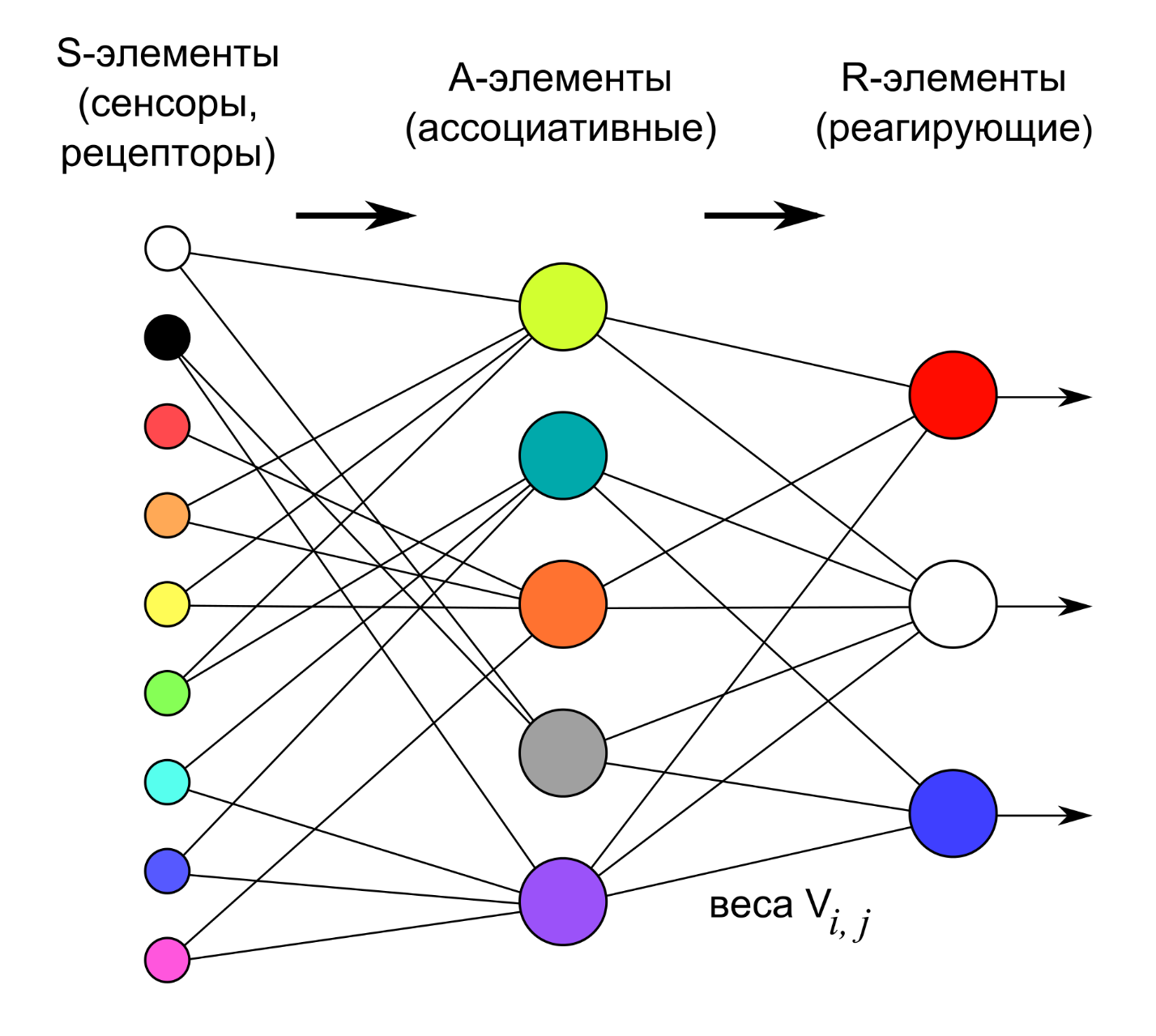


Рисунок 1. Логическая схема перцептрона с тремя выходами

Перцептрон состоит из трёх типов элементов:

1. Сенсоры – принимают сигналы и передают ассоциативным элементам;
2. Ассоциативные элементы – обрабатывают сигналы от Сенсоров и передают реагирующим элементам;
3. Реагирующие элементы – обрабатывают сигналы от ассоциативных элементов.

Перцептроны создают набор ассоциаций между входными стимулами и необходимым реакциями.

# Глава 2. Классификация перцептронов

1. Перцептрон с одним скрытым слоем (элементарный перцептрон) — перцептрон, у которого имеется только по одному слою S, A и R элементов.
2. Однослойный персептрон — перцептрон, каждый S-элемент которого однозначно соответствует одному А-элементу, S-A связи всегда имеют вес 1, а порог любого А-элемента равен 1. Часть однослойного персептрона соответствует модели искусственного нейрона. Его ключевая особенность состоит в том, что каждый S-элемент однозначно соответствует одному A-элементу, все S-A связи имеют вес, равный +1, а порог A элементов равен 1. Часть однослойного перцептрона, не содержащая входы, соответствует искусственному нейрону, как показано на Рисунке 1. Таким образом, однослойный перцептрон — это искусственный нейрон, который на вход принимает только 0 и 1. Однослойный персептрон также может быть и элементарным персептроном, у которого только по одному слою S,A,R-элементов.
3. Многослойный перцептрон по Розенблатту — перцептрон, который содержит более 1 слоя А-элементов.
4. Многослойный перцептрон по Румельхарту — частный случай многослойного персептрона по Розенблатту, с двумя особенностями:
   1. S-A связи могут иметь произвольные веса и обучаться наравне с A-R связями;
   2. Обучение производится по специальному алгоритму, который называется обучением по методу обратного распространения ошибки.

# Глава 3. Принцип работы перцептрона

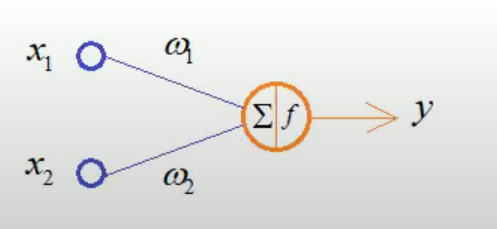


Рисунок 2. – Простейший перцептрон

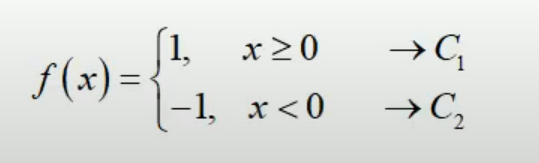
На рисунке 2 изображен простейший прецептрон, который решает задачу классификации двух образов, представленных характеристиками x1 и x2. Из этого следует, что если наша активационная функция на рисунке 3 будет давать значение 1, то мы будем делать вывод, что активационных сигнал принадлежит классу C1, в ином случае сигнал будет принадлежать классу C2.

Рисунок 3 – Активационная функция

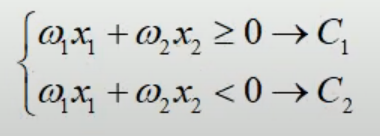


Рисунок 4 – Граница разделения

Рисунок 4 показывает, что граница разделения двух классов C1 и C2 находится на значении ноль. То есть, когда наша сумма больше нуля, это класс C1. Если же значение меньше нуля, то это класс C2. Из этого следует, что, когда выполняется равенство на рисунке 5, мы находимся на границе раздела двух классов.

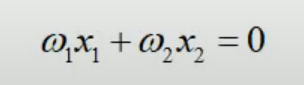


Рисунок 5 – Пороговое равенство

Выражение на рисунке 5 можно записать и в другом виде (рисунок 6). Это есть ничто иное, как прямая с угловым коэффициентом. То есть мы получаем уравнение прямой, которая проходит через начало системы координат (рисунок 6).

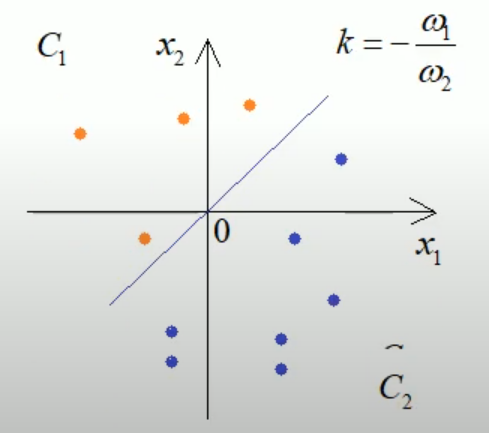


Рисунок 6 – Система координат

Относительно этой прямой можно определить множество точек для класса C1 и множество точек C2. Такая прямая получила название «Разделяющая прямая» (в многомерном случае - «Разделяющая гиперплоскость»).

# Глава 4. Реализация программы

Ниже представлена реализация обучающегося однонейронного перцептрона на языке Java.

import java.util.Arrays;  
  
import static java.util.Arrays.*copyOf*;  
  
// Однонейронный перцептрон с двумя входами  
public class Perceptron {  
 double[] entries; // Входы  
 double result; // Выход  
 double[] weights; // Весовые коэф.  
 double wk = 0.1; // Коэф. обучения  
  
 // Логическое "ИЛИ"  
 double[][] learnPatterns = { // Шаблоны обучения  
 {0,0, 0}, // {Вход, Вход, Ожидаемый Выход}  
 {0,1, 1},  
 {1,0, 1},  
 {1,1, 1}  
 };  
  
 public Perceptron() {  
 this.entries = new double[2];  
 this.weights = new double[entries.length];  
  
 for (int i = 0; i < weights.length; i++) {  
 weights[i] = Math.*random*() \* 0.2 + wk;  
 }  
 }  
  
 */\*\*  
 \* Функция вычисления выхода нейрона  
 \*/* public void calculateResult() {  
 result = 0;  
 for (int i = 0; i < entries.length; i++) {  
 result += entries[i] \* weights[i]; // Передаём уровень активности входных нейронов по весовому коэфу на выходной нейрон  
 }  
  
 result = result > 0.5 ? 1 : 0; // Определяем преодолён ли порог для активации нейрона  
 }  
  
 */\*\*  
 \* Обучение перцептрона.  
 \* Принцип: изменением весов в зависимости от ошибки до тех пор, пока ошибка не станет приемлемой  
 \* (не станет равно нулю в нашем случае)  
 \*/* public int learn() {  
 double globalErr;  
 int iterations = 0;  
 do { // Обучаем до тех пор, пока глобальная ошибка больше 0  
 globalErr = 0;  
 iterations++;  
 for (double[] learnPattern : learnPatterns) {  
 entries = *copyOf*(learnPattern, learnPattern.length - 1); // Входами стали элементы шаблона обучения (без последнего)  
 calculateResult();  
 double localErr = learnPattern[2] - result; // Считаем отклонение полученного результата от идеального из шаблона обучения  
 globalErr += Math.*abs*(localErr);  
 for (int i = 0; i < entries.length; i++) {  
 weights[i] += wk \* localErr \* entries[i];  
 }  
 }  
 } while (globalErr > 0);  
  
 return iterations;  
 }  
  
 public void test() {  
 int iterations = learn();  
 System.*out*.printf("Перцептрон обучен за %d итерации%n", iterations);  
 for (double[] learnPattern : learnPatterns) {  
 entries = *copyOf*(learnPattern, learnPattern.length - 1);  
 System.*out*.printf("Входные нейроны: %s%n", Arrays.*toString*(entries));  
 calculateResult();  
 System.*out*.printf("Результат: %f%n", result);  
 }  
 }  
  
 public static void main(String[] args) {  
 new Perceptron().test();  
 }  
}

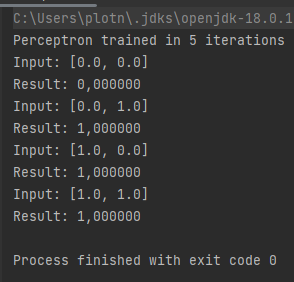


Рисунок 7. Результат выполнения прогрммы

# Итоги лабораторной работы

Мы изучили и научились реализовывать простейшие нейронные сети – перцептроны.

# Список литературы

1. [[wikipedia.org](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD)] – Перцептрон (Год, старницы, эл. Ресурс, ссылка в работе)