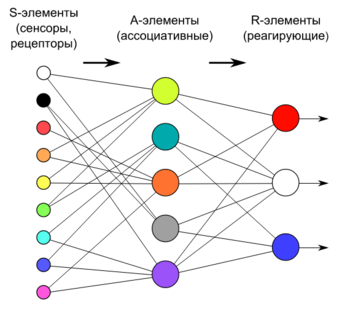
Отчёт по реализации перцептрона для бинарной классификации



**Перцептрон** (англ. *Perceptron*) — простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.



### Обучение перцептрона

Задача обучения перцептрона — подобрать такие w0,w1,w2,…,wnw0,w1,w2,…,wn, чтобы sign(σ(w0+w1⋅x1+w2⋅x2+…+wn⋅xn))sign(σ(w0+w1⋅x1+w2⋅x2+…+wn⋅xn)) как можно чаще совпадал с y(x)y(x) — значением в обучающей выборке (здесь σσ — функция активации).

**Класс**Perceptron содержит:

1. **Параметры модели**:
   * weights[] - массив весов
   * bias - смещение
   * learningRate - скорость обучения (α)
   * epochs - количество эпох обучения
2. **Методы**:
   * Sigmoid(z) - реализация сигмоидной функции

**Сигмоида**:

return 1.0 / (1.0 + Math.Exp(-z));

* + Predict(inputs) - предсказание для входных данных

**Предсказание**:

z = bias;

for (int j = 0; j < weights.Length; j++)

{

z += inputs[i][j] \* weights[j];

}

outputs[i] = Sigmoid(z);

* + Train() - обучение модели
  + EvaluateAccuracy() - оценка точности
  + Генерация синтетических данных (GenerateData())

1. **Основные принципы:**
   * Веса инициализируются малыми случайными значениями (±0.01)
   * Смещение инициализируется нулём
   * Для каждой эпохи:
   * Методом градиентного спуска ищется минимум функции потерь
     + Обновляются параметры
   * каждые 10 эпох выводятся данные

**4.Генерация данных**:

* + Создаётся линейно разделимая выборка с шумом
  + Используется сигмоидная модель для генерации меток
  + Добавляется 5-10% шума для реалистичности

**Остальные методы:**

**1.Градиенты**:

double error = predictions[i] - trainLabels[i];

dw[j] += error \* trainInputs[i][j];

db += error;

**2.Обновление весов**:

weights[j] -= learningRate \* dw[j] / trainInputs.Length;

bias -= learningRate \* db / trainInputs.Length;

По итогу модель достигла точности 86% при шуме 5% и точности 84% при шуме 10%

https://github.com/Svifif/Polimorpheus/tree/main/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F%20%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0/perceptron