

Научная статья  
УДК 004.8

## ЭЛЕМЕНТАРНЫЙ ПЕРСЕПТРОН. ПРОЦЕСС НАСТРОЙКИ

Евгения Игоревна Новосельцева<sup>1</sup>, Ольга Николаевна Юркова<sup>2</sup>

<sup>1,2</sup> Брянский государственный инженерно-технологический университет,  
Брянск, Россия

<sup>1</sup> zhenechka.novoselceva@mail.ru

<sup>2</sup> yurkova\_olga@mail.ru

**Аннотация.** Обучение перцептрона проводится на примерах выбранной задачи классификации. Знаком успешного обучения является правильное распознавание всех примеров. Обучение считается успешным со всеми правильно распознанными примерами, только если входные значения могут быть разделены геометрически с помощью прямой.

**Ключевые слова:** перцептрон, обучение, входные данные

Scientific article

## ELEMENTARY PERCEPTRON. THE SETUP PROCESS

Evgeniya I. Novoseltseva<sup>1</sup>, Olga N. Yurkova<sup>2</sup>

<sup>1,2</sup> Bryansk State Engineering and Technology University, Bryansk, Russia

<sup>1</sup> zhenechka.novoselceva@mail.ru

<sup>2</sup> yurkova\_olga@mail.ru

**Abstract.** Perceptron training is carried out using examples of the selected classification task. The indicator of successful training is the correct recognition of all examples. Training is considered successful with all correctly recognized examples only if the input values can be separated geometrically using a straight line.

**Keywords:** perceptron, training, input data

Элементарный перцептрон – это один нейрон с активационной функцией в виде порогового элемента с несколькими входами и одним выходом. Выходной сигнал может принимать только одно из двух значений: минус 1 и +1 (или 0 и 1) (рис. 1–2) [1].

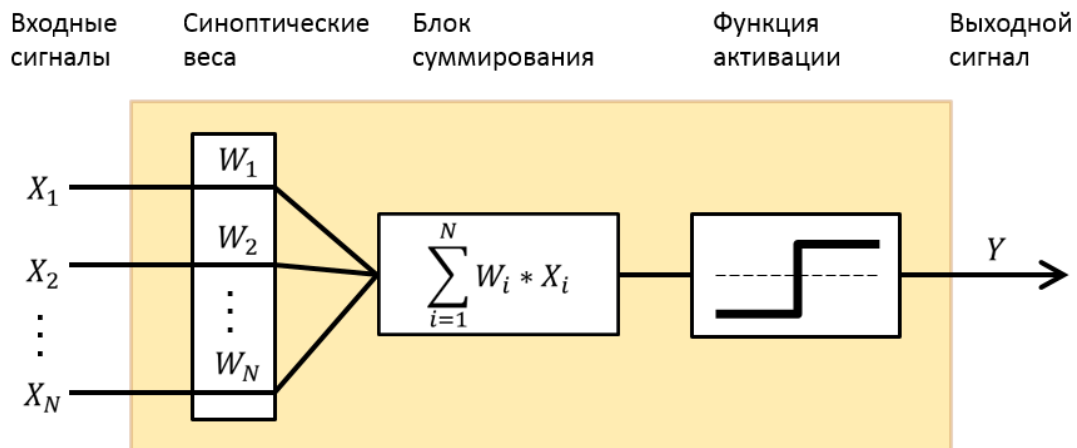


Рис. 1. Схема элементарного персептрона

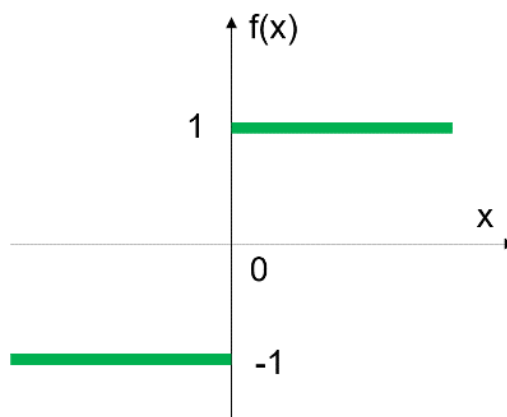


Рис. 2. Активационная функция в виде порогового элемента

Обучение персептрона начинается с подготовки обучающей выборки.

Обучающая выборка – различные варианты значений, которые поступают на вход персептрона, а также соответствующие им желаемые выходные сигналы [2, 3].

Первый шаг – нормализация входных данных.

Все данные, используемые для обучения персептрона, должны быть нормализованы. Чтобы их нормализовать, необходимо найти максимально возможное по модулю значение для каждого из входов  $x_i$ , и все поступающие значения на соответствующий вход разделить на это число.

После этого входные данные будут представлены в диапазоне от  $-1$  до  $1$ . Тем самым обеспечивается корректная работа обучающей модели.

Вектор значений для нормализации входов  $x_i$  необходимо хранить, чтобы использовать его при нормализации входных данных для тестирования персептрона. На рис. 3–4 представлен пример нормализации данных.

```

for j in range(x_count):
# находим максимальный по модулю элемент в каждом столбце
x_norm[j]= x[0][j]
for i in range(ex_count):
    if np.abs(x[i][j]) > x_norm[j] :
        x_norm[j] = np.abs(x[i][j])
# x_norm – вектор для нормализации

```

Рис. 3. Формирование вектора нормализации

### **Нормализация:**

```

# делим элементы столбца на максимальный по модулю элемент
for j in range(x_count):
for i in range(ex_count):
    x[i][j] = x[i][j] / x_norm[j]

```

Рис. 4. Нормализация данных

Второй шаг – инициализация синоптических весов и смещения.

Значения синоптических весов  $w_i(0)$  и смещение нейрона  $b$  устанавливаются равными случайным числам в диапазоне  $(-0.1... 0.1)$ .

Здесь  $w_i(t)$  – вес связи от  $i$ -го элемента входного сигнала к нейрону в момент времени  $t$ . Размерность вектора  $W$  равна количеству элементов во входном сигнале  $X$  (рис. 5–6) [2].

```

# начальные веса нейрона (количество весов равно количеству входов)
w = (np.random.rand(x_count)-0.5)/10
# смещение нейрона
b = np.random.rand()/10

```

Рис. 5. Инициализация весов нейрона  $W$  и смещения нейрона  $b$

```

y = 0
for j in range(x_count):
    y = y + x[j] * w[j]
y = y+b
if y < 0 :
    y=-1
else:
    y=1

```

Рис. 6. Вычисление выходного сигнала нейрона на основе пороговой функции активации

Третий шаг – предъявление перцептрону обучающего входного сигнала.

Текущий выход нейрона вычисляется на основе входного обучающего сигнала по формуле

$$y(t) = f\left(\sum_{i=1}^N w_i(t)x_i(t) + b\right). \quad (1)$$

Четвертый шаг – сравниваем желаемый выходной сигнал  $d(t)$  из обучающего примера с сигналом  $y(t)$ , вычисленным на шаге 3.

Если выход нейрона  $y(t)$  соответствует правильному значению  $d(t)$ , переходим к следующему шагу. Иначе выполняется адаптация (настройка, коррекция) значений весов нейрона  $w_i$  по формуле

$$w_i(t+1) = w_i(t) + r(d(t) - y(t))x_i(t), \quad (2)$$

здесь  $r$  – шаг обучения, который влияет на скорость обучения ( $0 < r < 1$ , например  $r = 0,05$ ; чем меньше значение  $r$ , тем плавнее и медленнее изменяются параметры нейрона).

Если желаемый выход нейрона (массив  $D$ ) не совпадает с вычисленным сигналом нейрона  $Y$ , то корректируем веса нейрона  $W$  (т. е. выполняем обучение нейрона) (рис. 7).

```

flag = True      # чтобы зайти в цикл
while flag:
# очередной полный проход всех примеров обучения (эпоха обучения)
    flag = False
    for i in range(ex_count):
        y = y_porog(x[i])
        if y != d[i]: # если текущий выход нейрона не совпал с желаемым
            flag = True # если хотябы 1 раз меняем веса
            for j in range(x_count):
                w[j] = w[j] + r*(d[i]-y)*x[i][j] # делаем коррекцию весов

```

Рис. 7. Коррекция весов

Если есть еще примеры для обучения, то переходим к шагу 3, иначе обучение завершено.

После успешного обучения модели на графике видно разделение выходных данных прямыми. Например, если у персептрона два входа, разделитель объектов является прямой линией в двухмерной системе координат. В случае трех входов разделение осуществляется плоскостью, пересекающей трехмерное пространство. Для четырех или более входов разделение двух объектов выполняется гиперплоскостью – геометрическим объектом, который пересекает пространство четырех или большего числа измерений [4].

И последний шаг – тестирование персептрона.

Оно проводится на примерах, по которым не проводилось обучение. В результате тестирования должна быть определена надежность распознавания, которая в первом приближении вычисляется как отношение количества ошибочно распознанных примеров к общему количеству примеров тестирования.

В заключение можно сделать выводы, что данный метод обучения элементарного перцептрона подходит для решения таких задач, как, например, разделить 2 или 3 города на карте, используя прямые.

#### *Список источников*

1. Перцептрон // Википедия. Свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/Перцептрон> (дата обращения: 04.10.2022).
2. Хайкин. С. Нейронные сети : полный курс. 2-е изд. М. : Вильямс, 2018. 1069 с.
3. Ясницкий Л. Н. Введение в искусственный интеллект. М. : Академия, 2019. 176 с.
4. Галушкин. А. И. Нейронные сети. Основы теории. М. : Горячая линия. Телеком, 2018. 496 с.