

### Компьютерная модель персептрона

Одной из важнейших проблем искусственного интеллекта является *проблема распознавания образов*, которая состоит в следующем. Допустим, имеется множество из  $n$  объектов:  $\{O_1, O_2, \dots, O_n\}$ , каждый из которых может быть отнесен к одному из классов  $K_1, K_2, \dots, K_m$ . Это позволило бы создать автоматы, способные распознавать буквы и звуки, читать и воспринимать устную речь, свободно ориентироваться в окружающей обстановке, обучаться и самообучаться.

Природа в результате длительной эволюции создала такое "устройство", способное распознавать образы, классифицировать объекты, обучаться и самообучаться. Им является мозг человека, обезьяны, других высокоорганизованных животных. Согласно нейрофизиологической теории, мозг представляет собой совокупность связанных между собой нервных клеток — *нейронов*, каждая из которых может находиться в возбужденном и невозбужденном состояниях. Нейроны соединены между собой и образуют сложные *нейронно-сетевые структуры*, определяющие мыслительную деятельность человека. В процессе обучения электрические импульсы, многократно проходя через одни и те же *межнейронные (синаптические) связи*, изменяют их свойства. Часто используемые связи усиливаются, редко используемые ослабляются и даже отмирают. В результате мозг запоминает поступающую информацию, последовательность действий, образы различных предметов, возникает распределенная память.

А что, если создать электронную модель реального нейрона и на его базе построить искусственную нейросеть? Можно ли ее научить распознавать образы? Чтобы ответить на этот вопрос, познакомимся с абстрактной моделью нейрона, предложенной Мак-Каллоком и Питсом, которая называется *формальным нейроном*.

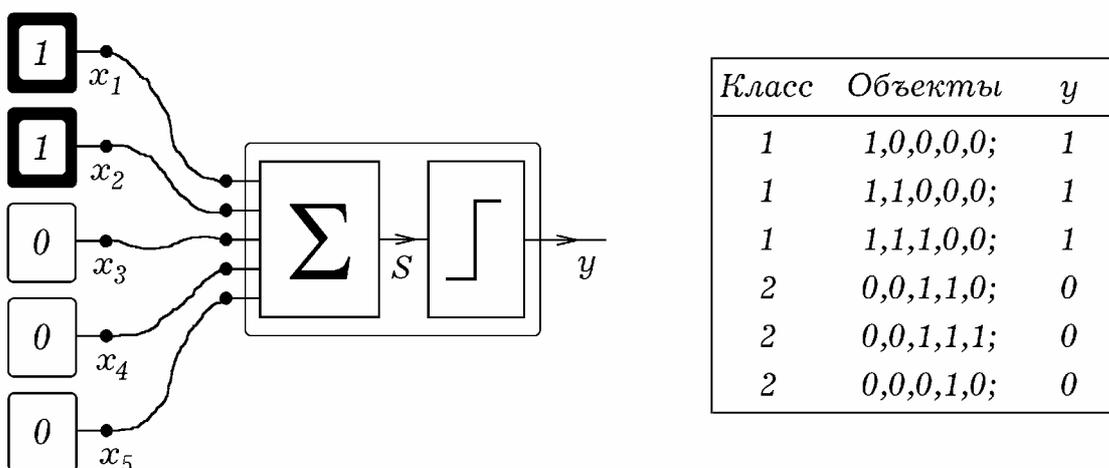


Рис. 1. Формальный нейрон как простейший персептрон.

Под формальным нейроном понимают гипотетический автомат с  $n$  входами  $x_1, x_2, \dots, x_n$ , и одним выходом  $y$ , характеризующийся порогом  $h$  и весами входов  $\omega_1, \omega_2, \dots, \omega_n$ . Он состоит из суммирующего и порогового элементов (рис. 1). Его выход возбужден ( $y=1$ ), когда взвешенная сумма всех входных сигналов превышает порог срабатывания. В противном случае выход не возбужден ( $y=0$ ). Это можно записать так:

$$y = \begin{cases} 1, & \text{если } \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n > h, \\ 0, & \text{если } \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n \leq h. \end{cases}$$

Если вес  $i$ -ого входа положительный ( $\omega_i > 0$ ), то вход возбуждающий, если отрицательный ( $\omega_i < 0$ ), — вход тормозящий.

```
uses crt;                                     {Программа ПР-1}
const N=6; h=0.2;                             {Модель нейрона}
x: array[1..N,1..5] of integer=((1,0,0,0,0), (1,1,0,0,0),
(1,1,1,0,0), (0,0,1,1,0), (0,0,1,1,1), (0,0,0,1,0));
w: array[1..5] of real=(1,1,1,-1,-1);
var m, i, j, k, y: integer; S: real;
BEGIN Clrscr;
  For m:=1 to N do begin
    S:=0; For i:=1 to 5 do S:=S+w[i]*x[m,i];
    If S>h then y:=1 else y:=0;
    Writeln('Объект ',m,': ', 'S= ',S:2:1, '; y= ',y);
  end; Readkey;
END.
```

Так как выход искусственного нейрона может находиться в двух состояниях, то он позволит разделить объекты только на два класса. А если создать компьютерную модель простейшего распознающего устройства на базе одного нейрона с 5 входами (рис. 1)? Это совсем несложно, — чтобы проимитировать работу нейрона достаточно написать программу, которая вычисляла бы алгебраическую сумму его возбужденных входов и использовала бы оператор условного перехода. Так работает программа ПР-1, написанная на языке Borland Pascal 7.0. В ней формальному нейрону последовательно предъявляются цифровые образы 6 объектов, каждый длиной 5 бит. Объекты образуют два класса:

$$K_1 = \{10000, 11000, 11100\} \text{ и } K_2 = \{00110, 00111, 00010\}.$$

Для того, чтобы нейрон правильно классифицировал объекты, зададим матрицу весов входов и порог срабатывания следующим образом:  $\omega_i = (1,1,1,-1,-1)$ ,  $h = 0,2$ . После запуска программы получаем:

|                        |                         |
|------------------------|-------------------------|
| Объект 1: S= 1.0; y= 1 | Объект 4: S= 0.0; y= 0  |
| Объект 2: S= 2.0; y= 1 | Объект 5: S= -1.0; y= 0 |
| Объект 3: S= 3.0; y= 1 | Объект 6: S= -1.0; y= 0 |

В 1958 г. Фрэнк Розенблат создал простейшую *кибернетическую модель мозга*, фактически являющуюся первым *нейрокомпьютером*, которую назвал *персептроном* (от англ. *percept* — воспринимать). Персептрон Розенבלата пред-

ставлял собой электронное устройство, распознающее оптические изображения, проецируемые на двумерную матрицу светочувствительных элементов. Сейчас под персептроном понимают обучаемую нейросеть, состоящую из датчиков (сенсоров), ассоциативных и реагирующих элементов с заданной матрицей весовых коэффициентов. Рассмотренная выше модель нейрона является однослойным персептроном. В многослойных персептронах присутствуют дополнительные слои ассоциативных элементов. В общем случае персептрон оперирует с цифровыми образами объектов, каждый из которых представим в виде массива нулей и единиц, ассоциируя различные входные стимулы с соответствующими им выходными реакциями.

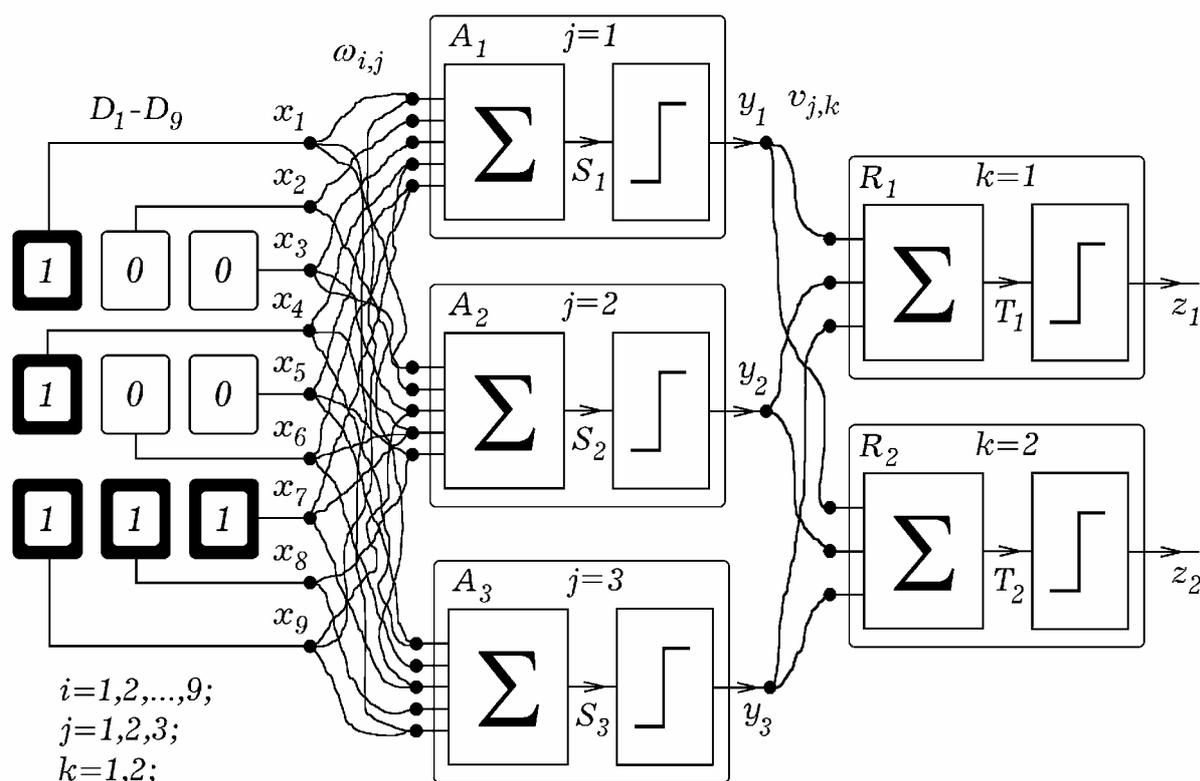


Рис. 2. Схема трехслойного персептрона.

Рассмотрим трехслойный персептрон, состоящий из слоя сенсоров или датчиков  $D_1, D_2, \dots, D_9$ , слоя ассоциативных элементов  $A_1, A_2, A_3$ , и слоя реагирующих элементов  $R_1, R_2$ . Если уровень воздействия на датчик превышает некоторое пороговое значение, то на его выходе появляется 1, а иначе — 0. Ассоциативный элемент работает как формальный нейрон: на выходе появляется 1, когда сумма всех весов возбужденных входов превышает порог срабатывания. Веса входов  $\omega_{i,j}$  принимают значения -1, 0 или 1. Реагирующий  $R$ -элемент работает так: когда сумма всех весов возбужденных входов положительна, на выходе 1, а когда отрицательна — на выходе — 0. Веса входов  $v_{j,k}$  реагирующего элемента могут принимать произвольные значения. Так как выход реагирующего элемента имеет два состояния, то персептрон с двумя  $R$ -

элементами может классифицировать объекты на четыре класса ( $2^2 = 4$ ), соответствующие выходным сигналам 00, 01, 10, 11.

| Объект 1 | Объект 2 | Объект 3 | Объект 4 | Датчики           |
|----------|----------|----------|----------|-------------------|
| 0 1 0    | 0 0 0    | 1 0 0    | 1 1 1    | $x_1$ $x_2$ $x_3$ |
| 0 1 0    | 1 1 1    | 1 0 0    | 0 0 1    | $x_4$ $x_5$ $x_6$ |
| 0 1 0    | 0 0 0    | 1 1 1    | 0 0 1    | $x_7$ $x_8$ $x_9$ |

Рис. 3. Цифровые образы объектов, предъявляемых перцептрону.

Создать перцептрон не просто, зато его можно сравнительно легко промоделировать на компьютере. Рассмотрим программу ПР-2, моделирующую работу трехслойного перцептрона, изображенного на рис. 2. Допустим, перцептрон должен различать четыре объекта, представленные на рис. 3. Учитывая расположение датчиков  $D_1, D_2, \dots, D_9$ , предъявляемые перцептрону объекты кодируются так:

$$O_1 = (0,1,0,0,1,0,0,1,0), \quad O_2 = (0,0,0,1,1,1,0,0,0),$$

$$O_3 = (1,0,0,1,0,0,1,1,1), \quad O_4 = (1,1,1,0,0,1,0,0,1).$$

Для того, чтобы перцептрон правильно классифицировал эти объекты, зададим веса связей, соединяющих датчики с ассоциативными элементами, и ассоциативные элементы с реагирующими элементами, следующим образом:

$$\omega_{i,j} = \begin{pmatrix} 0 & -1 & 1 & 1 & 0 & 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 1 & -1 & -1 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 \end{pmatrix}, \quad \nu_{j,k} = \begin{pmatrix} 2 & 2 & -2 \\ -2 & 2 & 2 \end{pmatrix}.$$

Порог срабатывания ассоциативных элементов должен быть равен  $h = 0,5$ .

```

uses crt;
const x: array[1..9] of integer=(1,1,1,
                                0,0,1,
                                0,0,1);
w: array[1..3,1..9] of real=((0,-1,1,1,0,1,-1,-1,0),
(0,1,1,-1,0,1,-1,-1,0), (1,0,1,-1,0,0,1,-1,1));
v: array[1..2,1..3] of real=((2,2,-2), (-2,2,2));
var m, l, j, k : integer;
S, y : array[1..3] of real; T, z : array[1..2] of real;
BEGIN Clrscr;
  For j:=1 to 3 do begin S[j]:=0;
    For i:=1 to 9 do S[j]:=S[j]+w[j,i]*x[i];
    If S[j]>0.5 then y[j]:=1 else y[j]:=0;
    Writeln('j= ',j,' | S= ',S[j]:2:1,' | y= ',y[j]:1:1); end;
  For k:=1 to 2 do begin T[k]:=0;
    For j:=1 to 3 do T[k]:=T[k]+v[k,j]*y[j];
    If T[k]>0 then z[k]:=1 else z[k]:=0;
    Writeln('k=',k,' | T=',T[k]:2:1,' | Выход z=',z[k]:1:0); end;
  Readkey;
END.
```

Результат работы программы представлен ниже.

```
= ОБЪЕКТ 1 =
j= 1 | S= -2.0 | y[1]= 0.0
j= 2 | S= 0.0 | y[2]= 0.0
j= 3 | S= -1.0 | y[3]= 0.0
k= 1 | T= 0.0 | Выход z[1]= 0
k= 2 | T= 0.0 | Выход z[2]= 0
= ОБЪЕКТ 2 =
j= 1 | S= 2.0 | y[1]= 1.0
. . . . .
= ОБЪЕКТ 4 =
j= 1 | S= 1.0 | y[1]= 1.0
j= 2 | S= 3.0 | y[2]= 1.0
j= 3 | S= 3.0 | y[3]= 1.0
k= 1 | T= 2.0 | Выход z[1]= 1
k= 2 | T= 2.0 | Выход z[2]= 1
```

Из распечатки видно, какие значения принимают величины  $S_j$ ,  $T_k$ ,  $j=1,2,3$ ,  $k=1,2$ , а также выходы  $y_j$  и  $z_k$  в случае, когда персептрон предьявляются объекты, изображенные на рис. 3. Объекту  $O_1$  соответствуют состояния  $z_1=0$ ,  $z_2=0$ , объекту  $O_2$  — состояния  $z_1=1$ ,  $z_2=0$ , объекту  $O_3$  — состояния  $z_1=0$ ,  $z_2=1$ , объекту  $O_4$  — состояния  $z_1=1$ ,  $z_2=1$ . Таким образом, наша компьютерная модель персептрона правильно распознает рассмотренные выше объекты. Вы можете дальше поэкспериментировать: изменить веса связей, предьявить персептрон другие объекты, промоделировать персептрон с большим числом ассоциативных и реагирующих элементов.

Чтобы обучить персептрон, учитель (специальное автоматическое устройство) предьявляет ему объекты, составляющие известные классы. В зависимости от реакции персептрона учитель соответствующим образом изменяет веса связей  $\omega_{i,j}$  и  $\nu_{j,k}$ . Если персептрон дает правильный ответ, веса связей, соединяющих возбужденные нейроны, автоматически увеличиваются, а веса связей, соединяющих возбужденный и невозбужденный нейроны, уменьшаются. При правильном обучении независимо от начального состояния весовых коэффициентов и последовательности предьявления объектов, персептрон обучится распознавать образы за конечное число шагов.

Нейрокомпьютеры применяются для управления самолетами, ракетами, технологическими процессами в режиме реального времени; распознавания человеческих лиц, отпечатков пальцев, букв, звуков, электрических сигналов, заболеваний; прогнозирования погоды, политической или финансовой ситуации, исхода лечения, поведения противника; оптимизации технических конструкций, экономических стратегий, лечения больного и т.д.

#### Литература

1. Майер Р.В. Задачи, алгоритмы, программы // [Персональная страница Р.В.Майера] / ГГПИ [Глазов, 2010]. URL: <http://maier-rv.glazov.net> (<http://komp-model.narod.ru>) (дата обращения 20.09.10).