深度学习

第 1 讲　感知机

Tian Lv

YouYing AI

2022 年 10 月 26 日

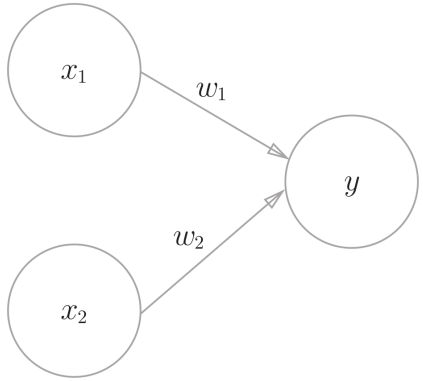
感知机是什么

感知机是由美国学者 Frank Rosenblatt 在 1957 年提出来的。

感知机是作为神经网络（深度学习）的起源的算法。学习感知机的构造是学习通向神经网络和深度学习的一种重要思想。

感知机是什么

感知机接收多个输入信号，输出一个信号。



图：有两个输入的感知机

x_1、x_2 是输入信号，y 是输出信号，w_1、w_2 是权重。图中的 ◯ 称为“神经元”或者“节点”。输入信号被送往神经元时，会被分别乘以固定的权重 $(w_1x_1$、$w_2x_2)$。神经元会计算传送过来的信号的总和，只有当这个总和超过了某个界限值时，才会输出 1。这也称为“神经元被激活”。这里将这个界限值称为阈值，用符号 θ 表示。

感知机是什么

感知机接收多个输入信号，输出一个信号。

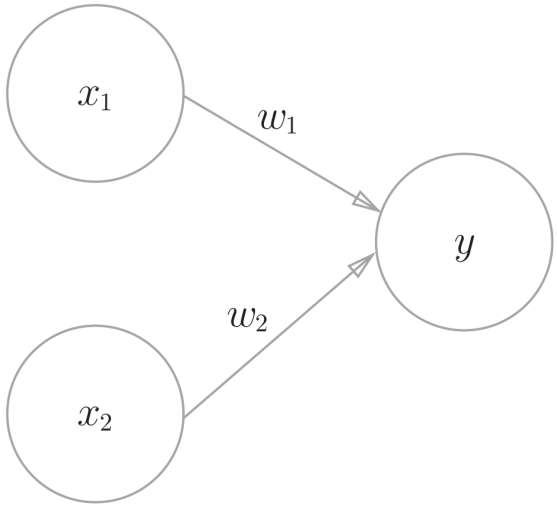


图: 有两个输入的感知机

$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

与门

简单逻辑电路

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

表: 与门的真值表

$$
\begin{aligned}
(w_1, w_2, \theta) &= (0.5, 0.5, 0.7) \\
&= (0.5, 0.5, 0.8) \\
&= (1.0, 1.0, 1.0)
\end{aligned}
$$

与非门 (NAND gate) 和或门

简单逻辑电路

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

表: 与非门的真值表

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

表: 或的真值表

$(w_1, w_2, \theta) = (-0.5, -0.5, -0.7)$

$(w_1, w_2, \theta) = (0.5, 0.5, 0.2)$

感知机的实现

```
def AND(x1, x2):
    w1, w2, theta = 0.5, 0.5, 0.7
    tmp = x1*w1 + x2*w2
    if tmp <= theta:
        return 0
    elif tmp > theta:
        return 1
```

感知机的实现

$$y = \begin{cases} 0 & (b + w_1 x_1 + w_2 x_2 \leq 0) \\ 1 & (b + w_1 x_1 + w_2 x_2 > 0) \end{cases}$$

```
def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

感知机的实现

```
def NAND(x1, x2):
  x = np.array([x1, x2])
  w = np.array([-0.5, -0.5])
  b = 0.7
  tmp = np.sum(w*x) + b
  if tmp <= 0:
    return 0
  else:
    return 1
```

```
def OR(x1, x2):
  x = np.array([x1, x2])
  w = np.array([0.5, 0.5])
  b = -0.2
  tmp = np.sum(w*x) + b
  if tmp <= 0:
    return 0
  else:
    return 1
```

异或门

感知机的局限

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

表: 异或门的真值表

线性和非线性

感知机的局限

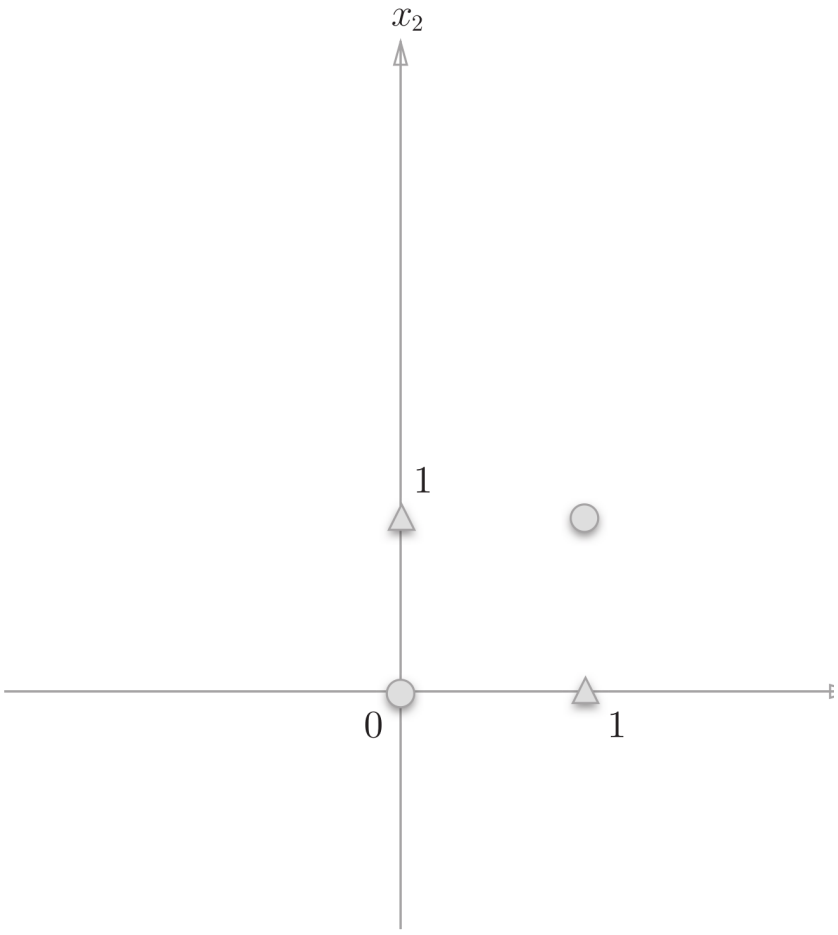


图: ◯ 和 △ 表示异或门的输出

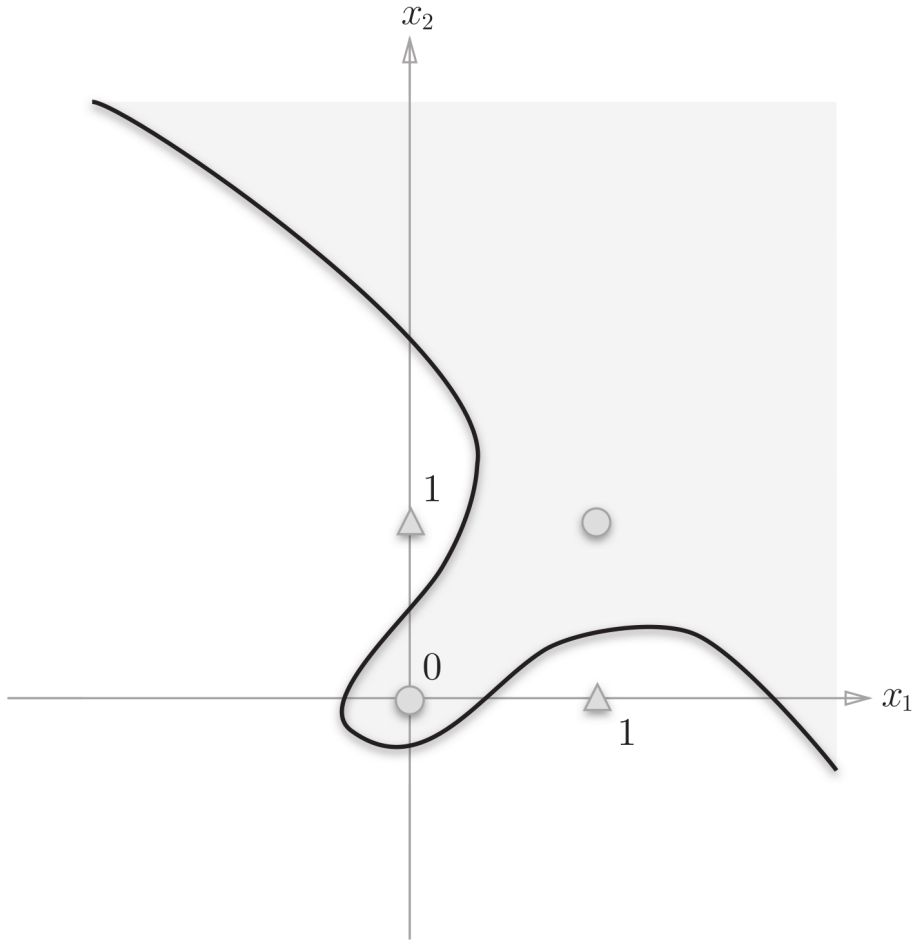


图: 使用曲线可以分开 ◯ 和 △

已有门电路的组合

多层感知机

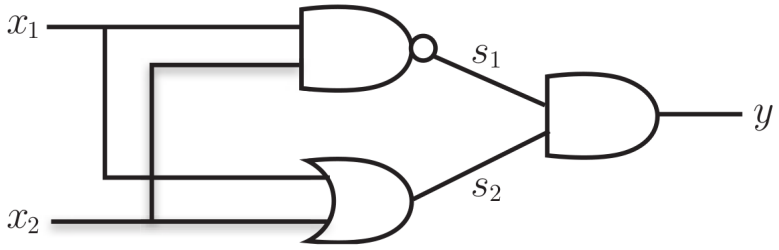


图: 通过组合与门、与非门、或门实现异或门

x_1	x_2	s_1	s_2	y
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	0	1	0

图: 异或门的真值表

异或门的实现

多层感知机

```
def XOR(x1, x2):
  s1 = NAND(x1, x2)
  s2 = OR(x1, x2)
  y = AND(s1, s2)
  return y
```

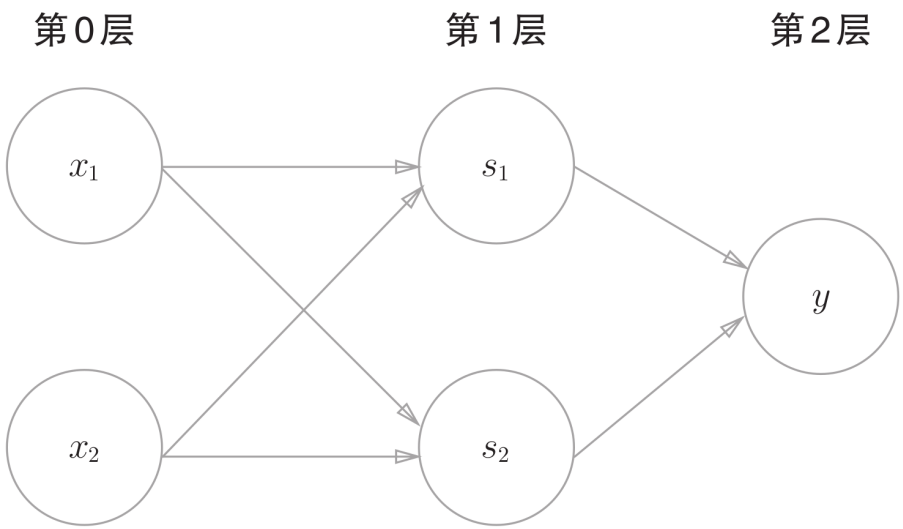


图: 用感知机表示异或门