

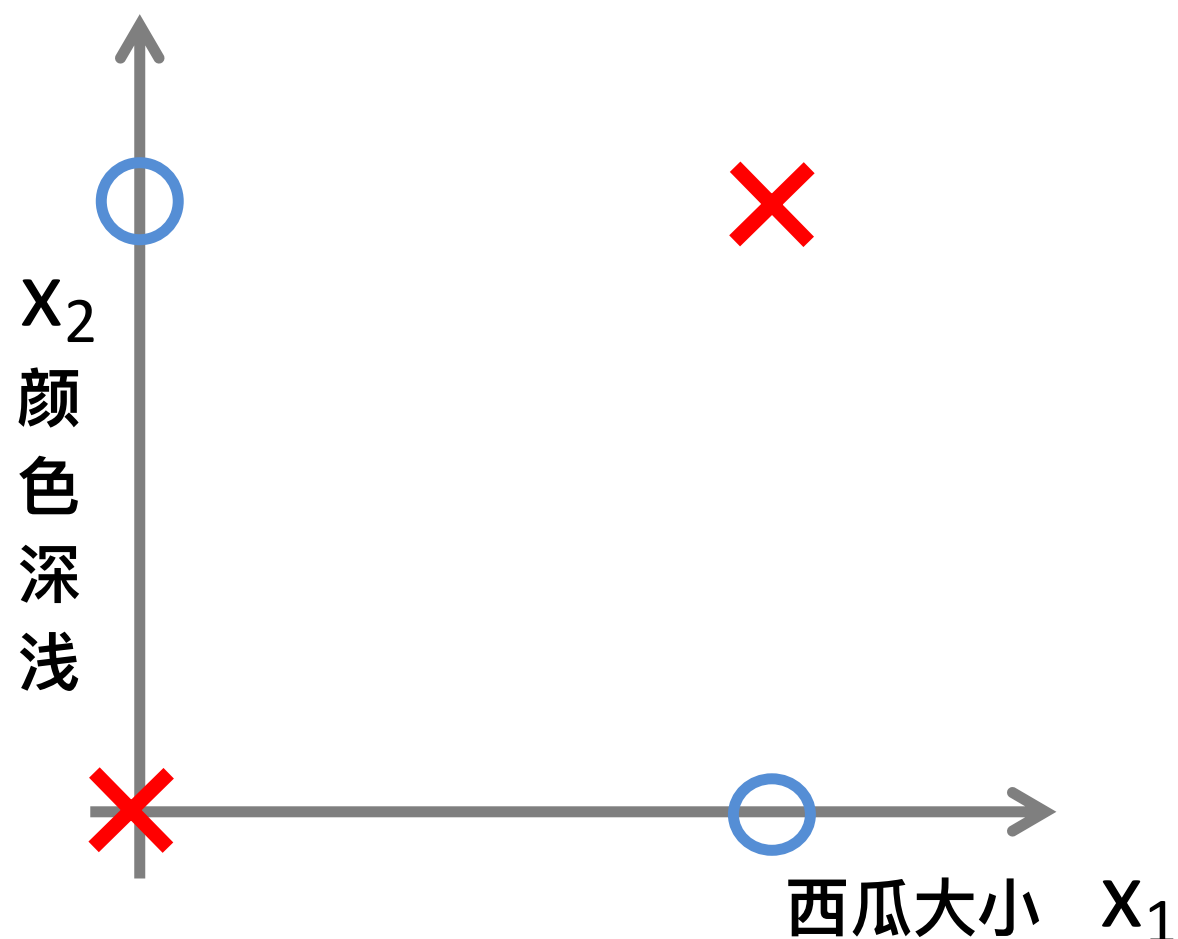
# 深度学习

构造神经网络解决XOR问题

**为什么神经网络首要解决XOR问题？**

# XOR问题

西瓜好坏与西瓜大小、西瓜颜色深浅的关系可由四个样本表示如下：



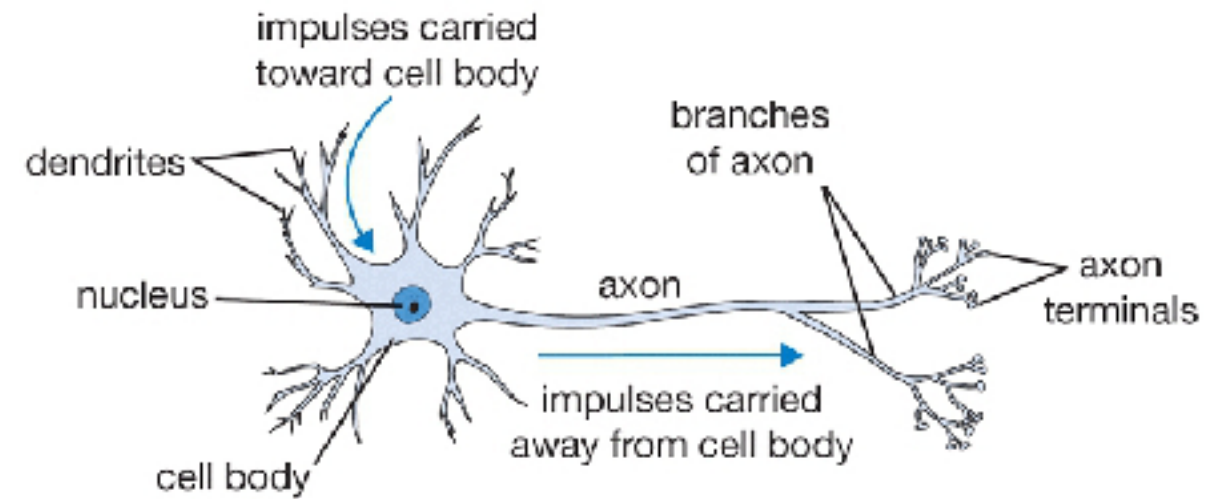
思考：如何使用一条直线区分两种类别？



无法区分

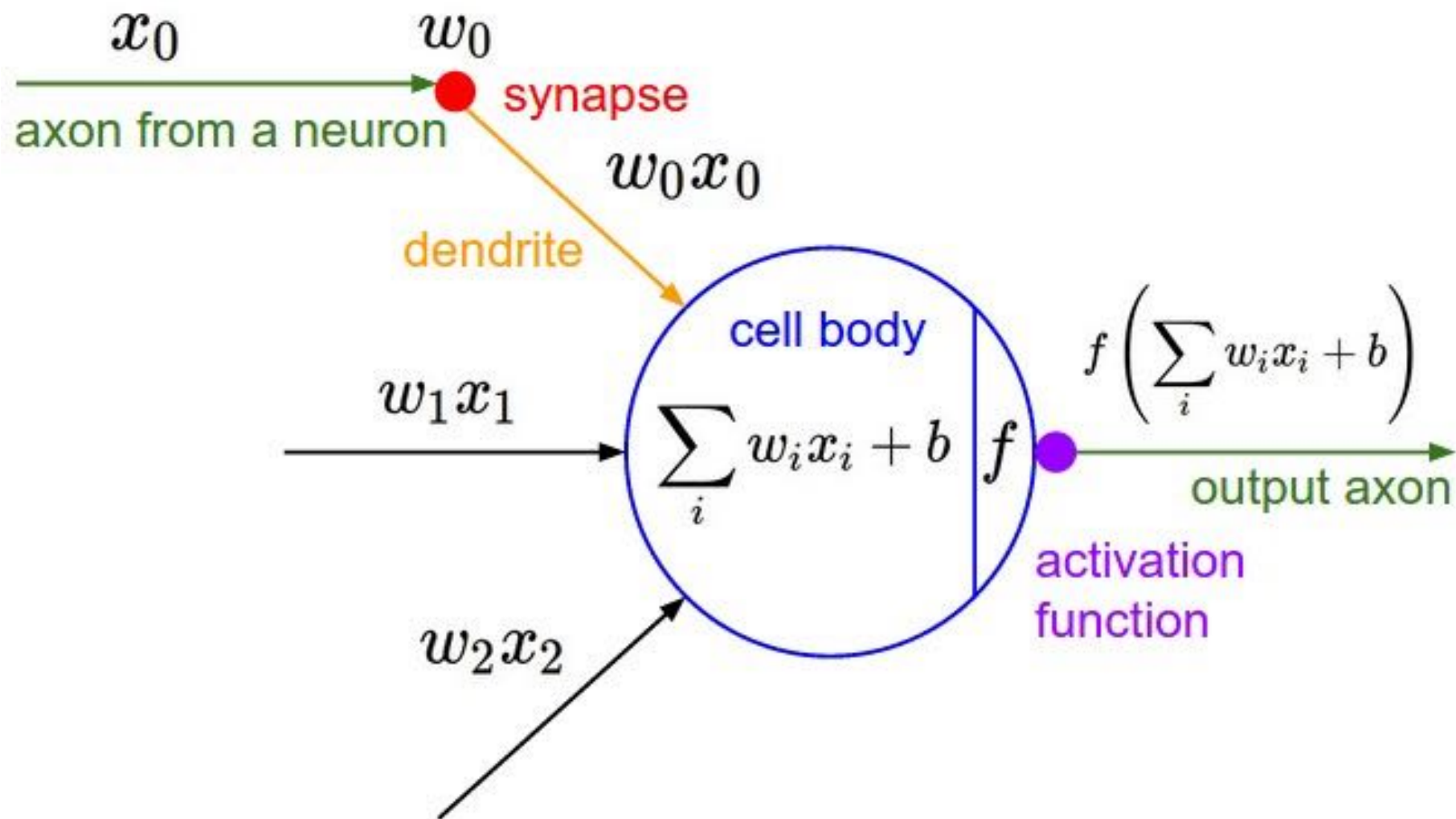
XOR即异或门，也是一个半加器。  
在图像上表现出线性不可分。

# 神经元如何工作



1. 树突存在多个，每个均可以传导电流进入胞体并存储。
2. 树突与上一个信号源的连接强弱程度不同，连接强度大则传导的电流大。
3. 胞体可以将每个树突传导来的电流存储在一起。
4. 每个神经元的胞体可以存储的电流多少不同，既触发传导下去的阈值不同。
5. 胞体传导出的电流强弱不一定与存储的电流相同。
6. 轴突传导出去的电流可以被多个树突检测到。

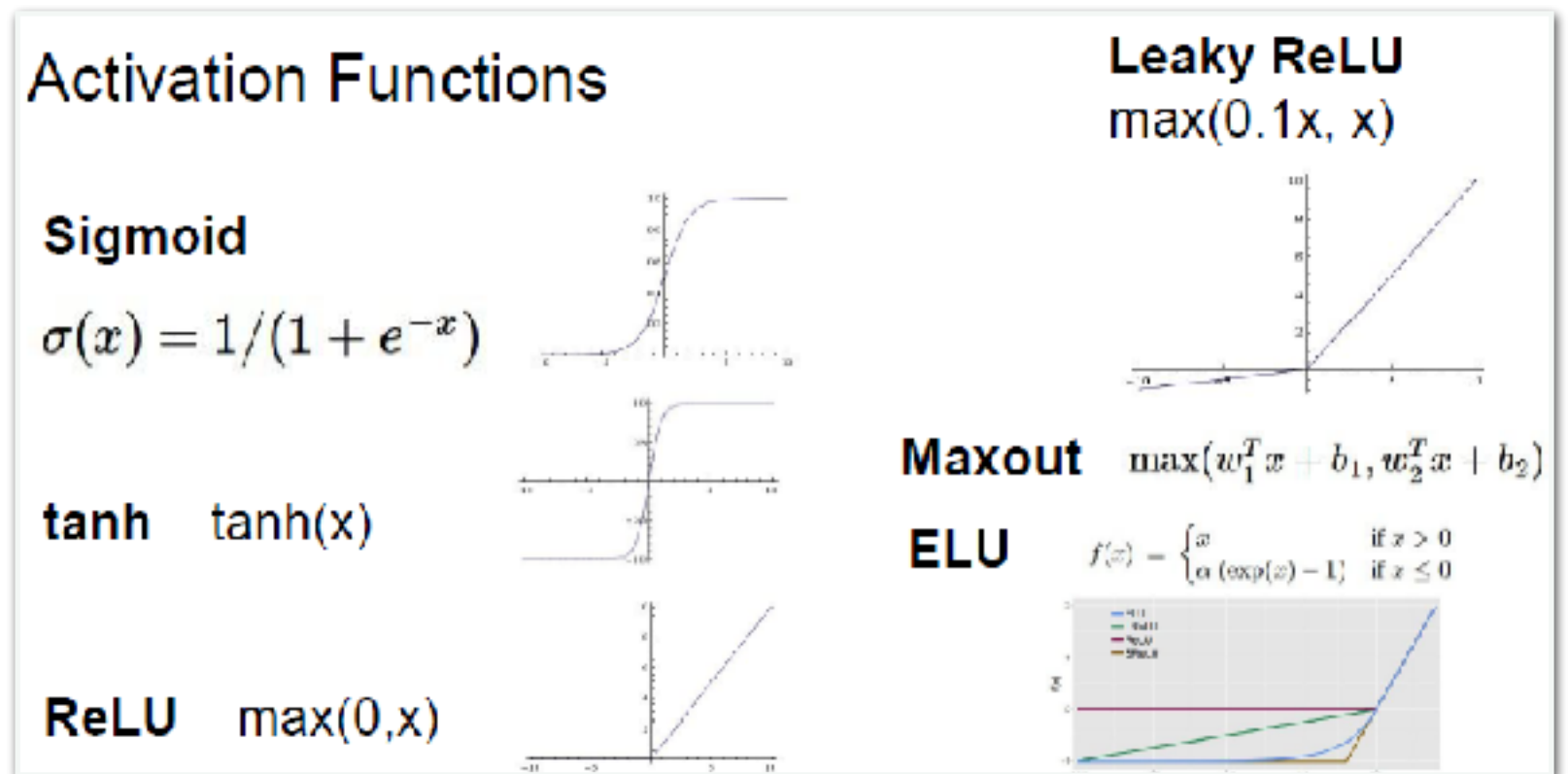
# 构造一个神经元



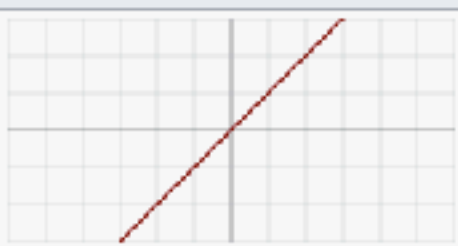

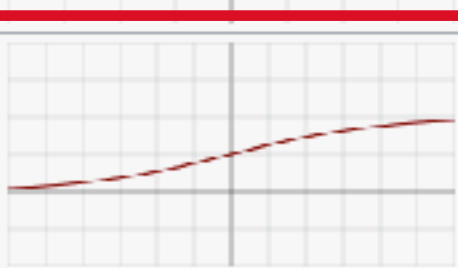
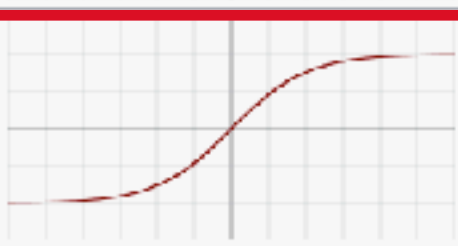
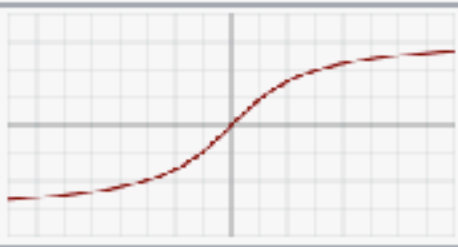
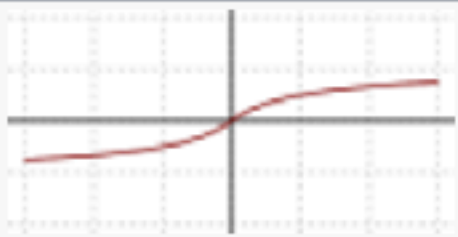
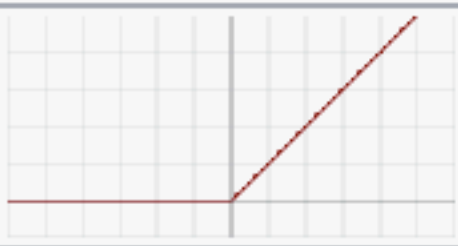
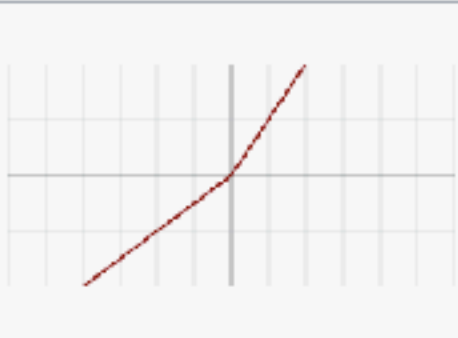
我们使用 $x$ 表示树突接收到的电流，使用 $w$ 表示树突的电阻，用sigma表示对输入的电流汇总，用 $f$ 表示对输入电流的转换（或者叫做激活函数），即轴突的输出。

# 神经元的激活函数

- 阶梯函数
- 线性函数
- 饱和线性函数
- 对数S形函数
- 强制非负校正函数
- .....



优秀的激活函数可以使神经网络更好的工作。设置线性激活函数或者不设置激活函数，会导致输出永远是输入的线性组合。为了能够解决上述提到的XOR问题，则必须选择非线性激活函数。

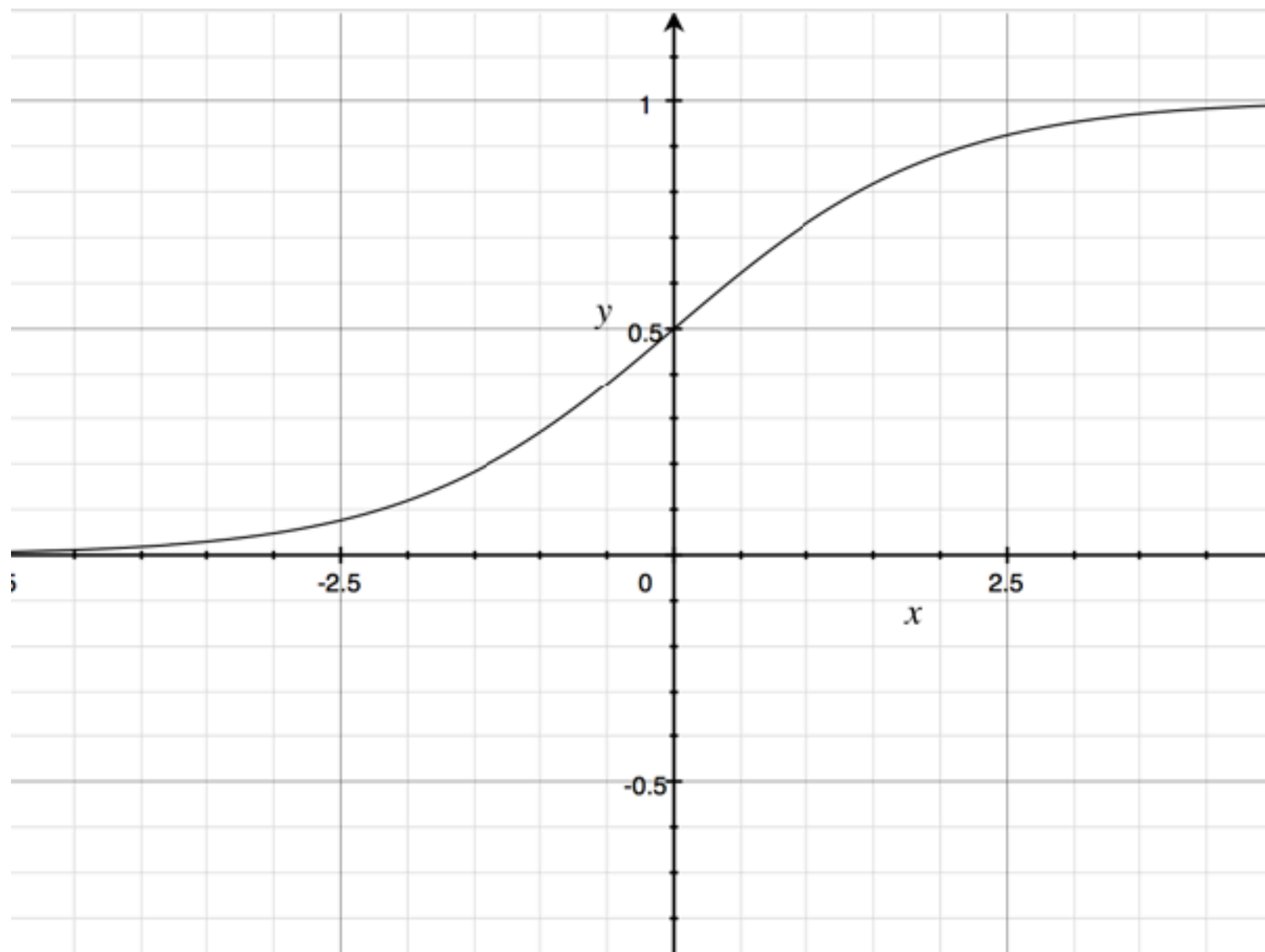
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Softsign [7][8]		$f(x) = \frac{x}{1 +  x }$
Rectified linear unit (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Leaky rectified linear unit (Leaky ReLU)[10]		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

早期最常用激活函数

注意：一个神经网络中不一定只有一种激活函数。

# Sigmoid激活函数

$$y = \frac{1}{1+e^{-x}}$$



优点：

定义域为R，值域为(0,1)。  
可以用来做二分类。  
类似于生物神经元。

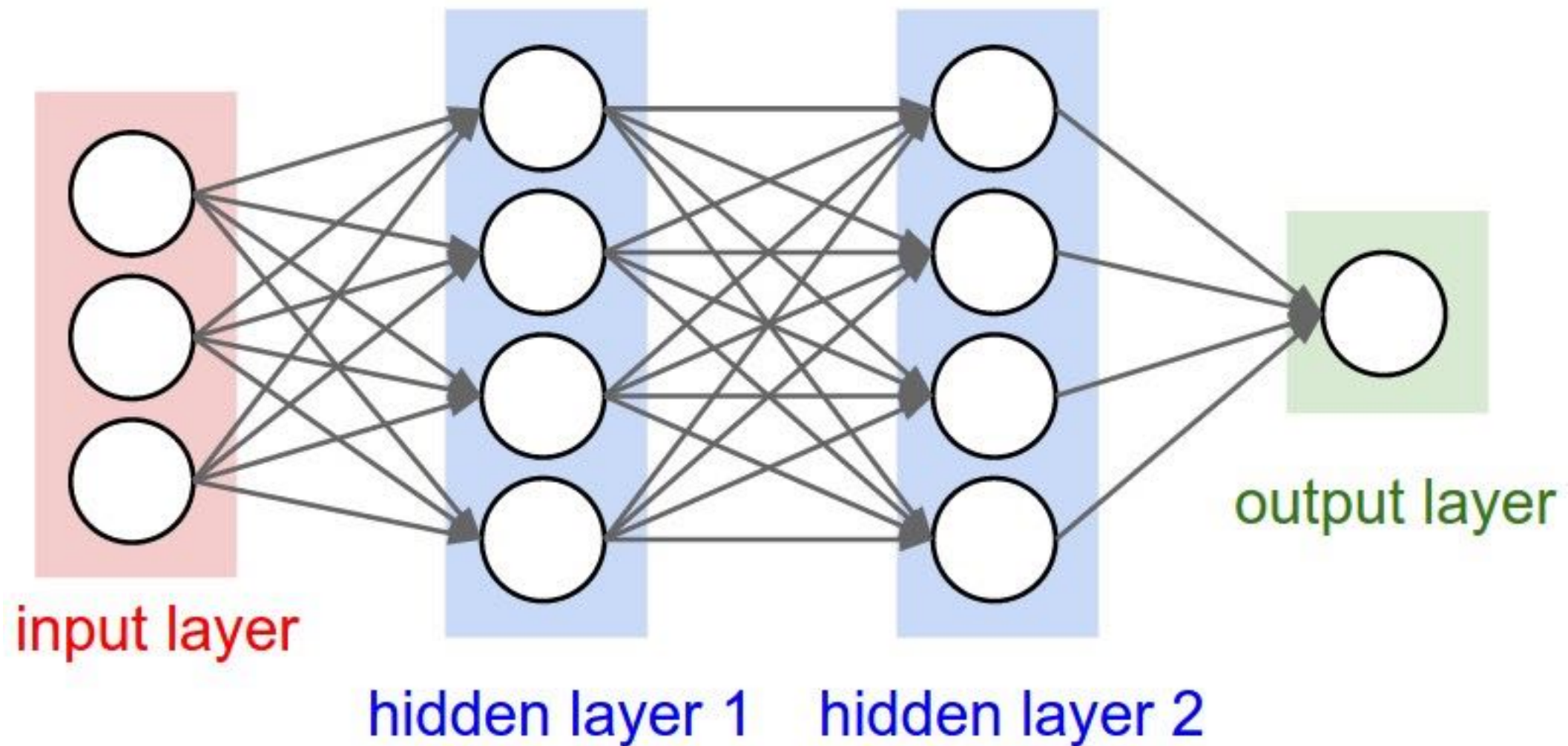
缺点：

计算量比较大。  
求导涉及除法。  
会导致梯度消失。

**sigmoid神经元可以模拟简单的逻辑回归。**



# 构造神经网络



图中每一个圆圈代表一个神经元(unit)。多个神经（一竖行）元组成一个层（layer）。层与层的神经元之间全连接。第一层我们称之为输入层，最后一层为输出层，其余为隐藏层。神经网络是一个有向无环图。

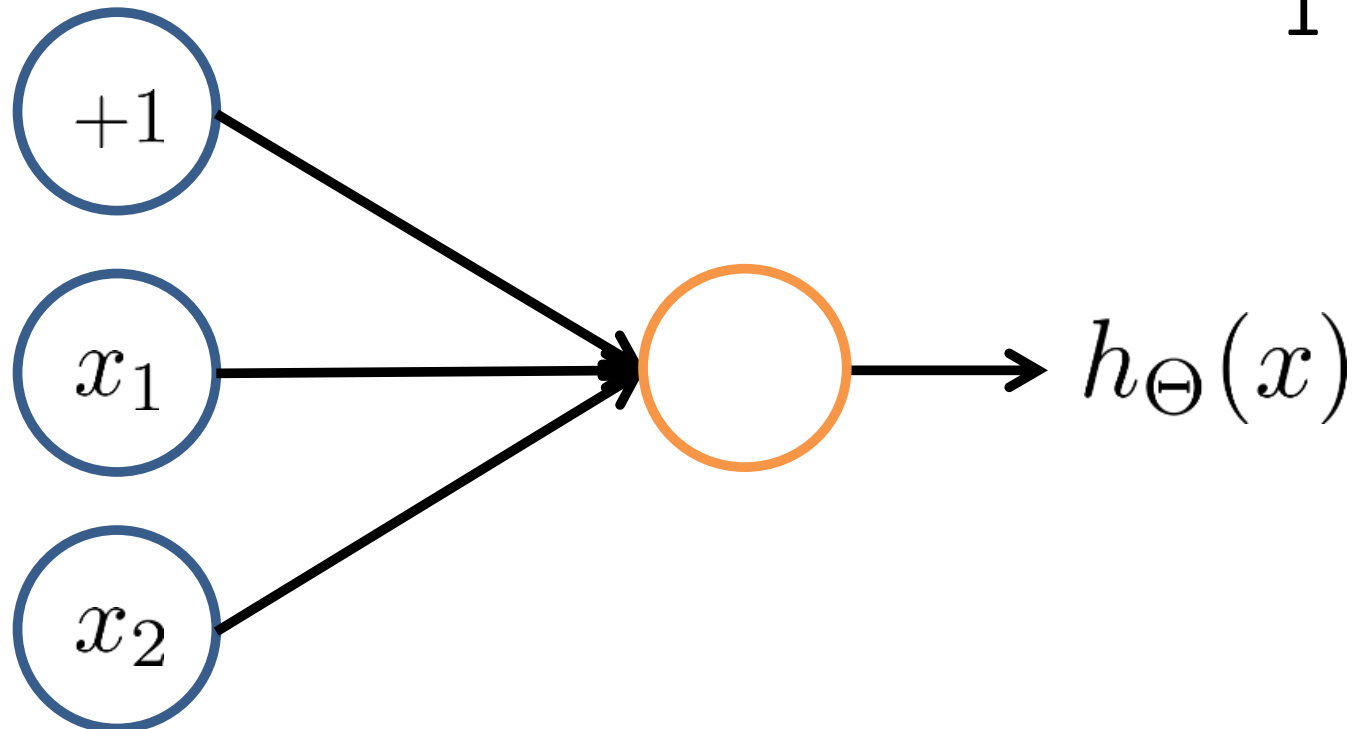
# 利用ANN构造逻辑运算单元

# 构造AND逻辑运算单元

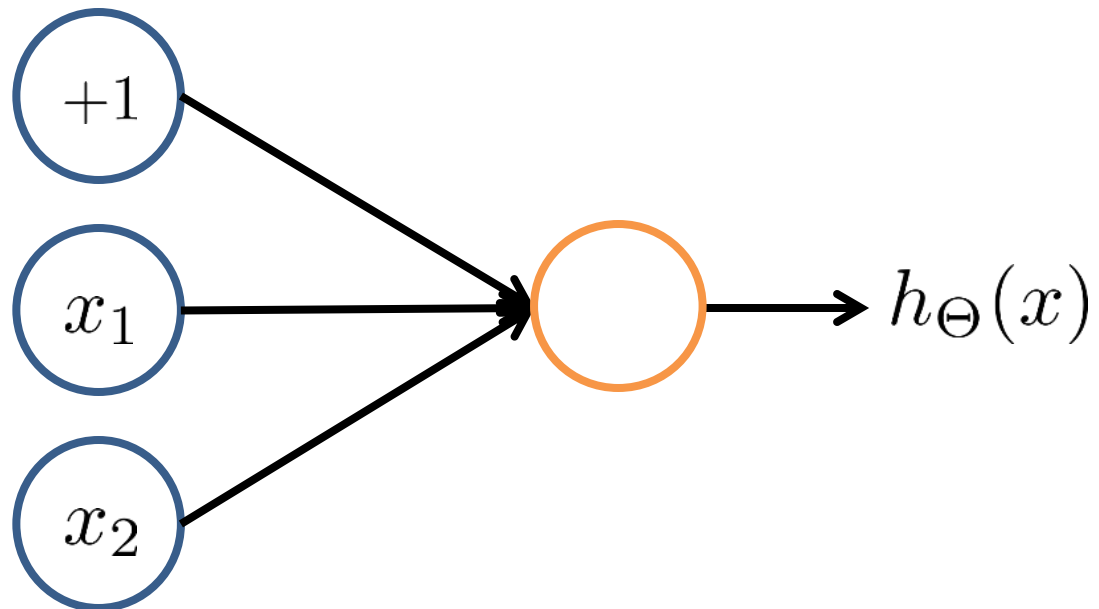
$$x_1, x_2 \in \{0, 1\}$$

$$y = x_1 \text{ AND } x_2$$

$x_1$	$x_2$	$h_{\Theta}(x)$
0	0	
0	1	
1	0	
1	1	

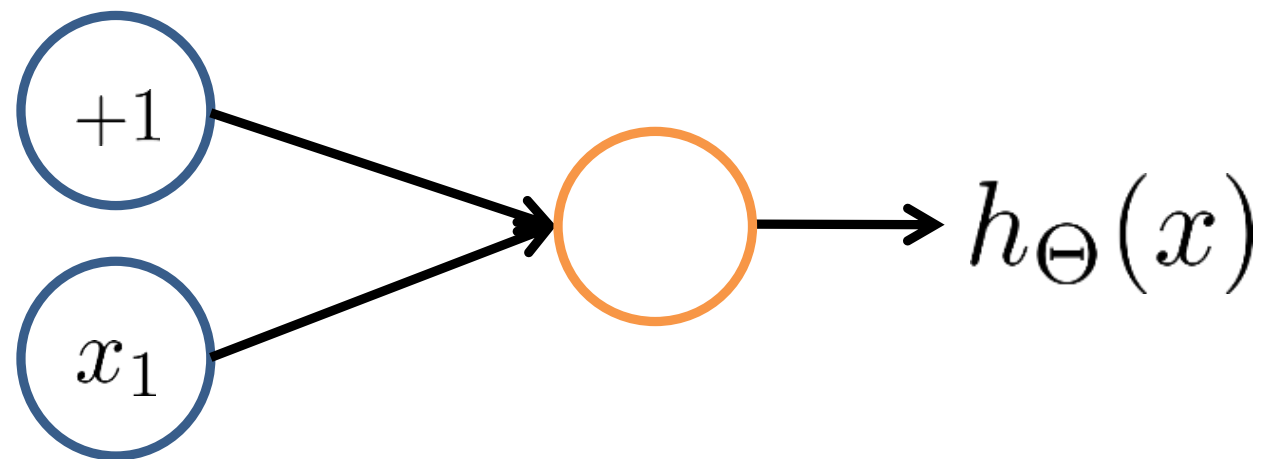


# 构造逻辑运算单元OR



$x_1$	$x_2$	$h_{\Theta}(x)$
0	0	
0	1	
1	0	
1	1	

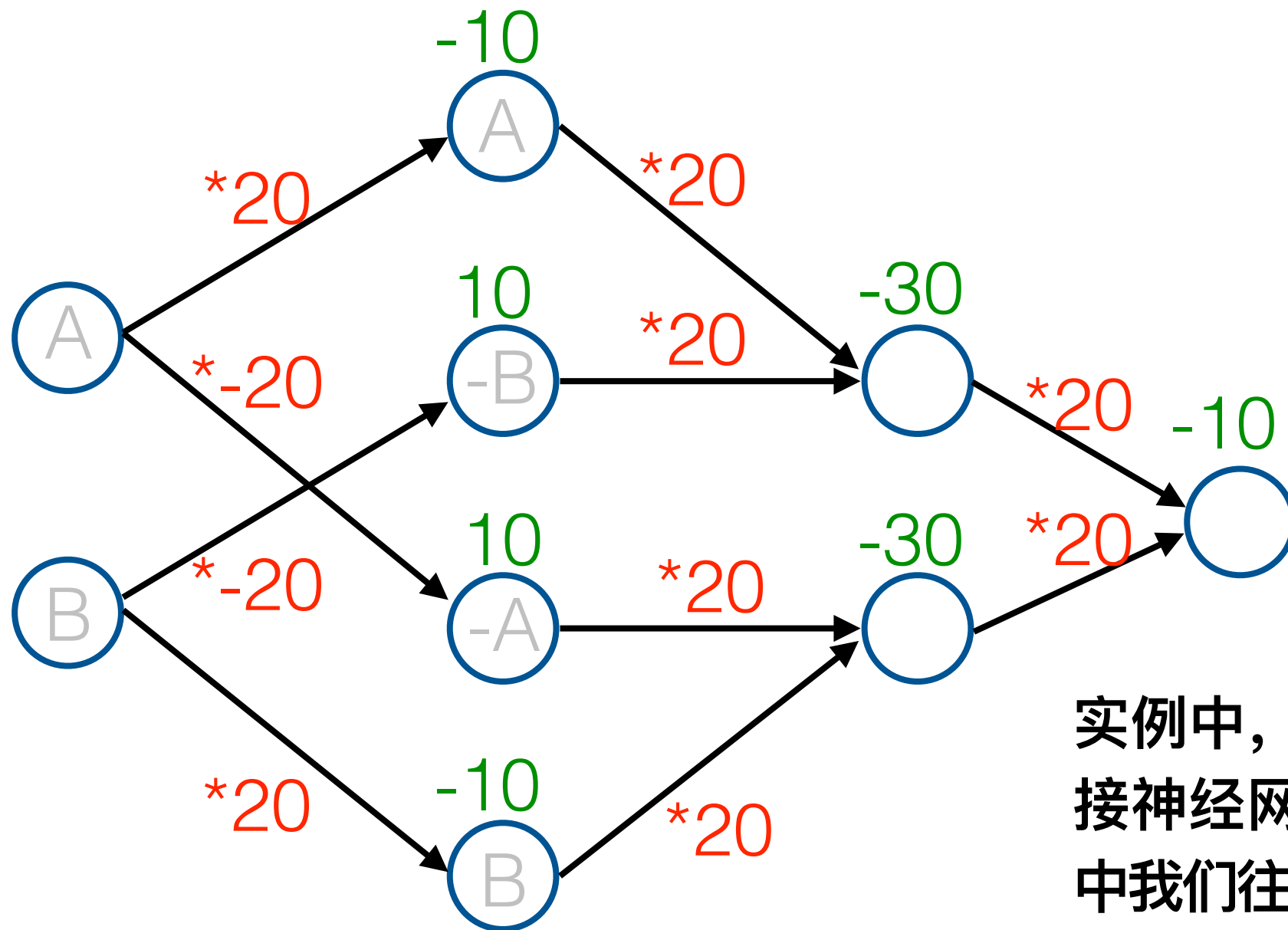
# 构造逻辑运算单元NOT



$x_1$	$h_{\Theta}(x)$
0	
1	

# XOR如何构造?

$$A \text{ xor } B = (A \text{ and } (\text{not } B)) \text{ or } ((\text{not } A) \text{ and } B)$$



输入不同输出1

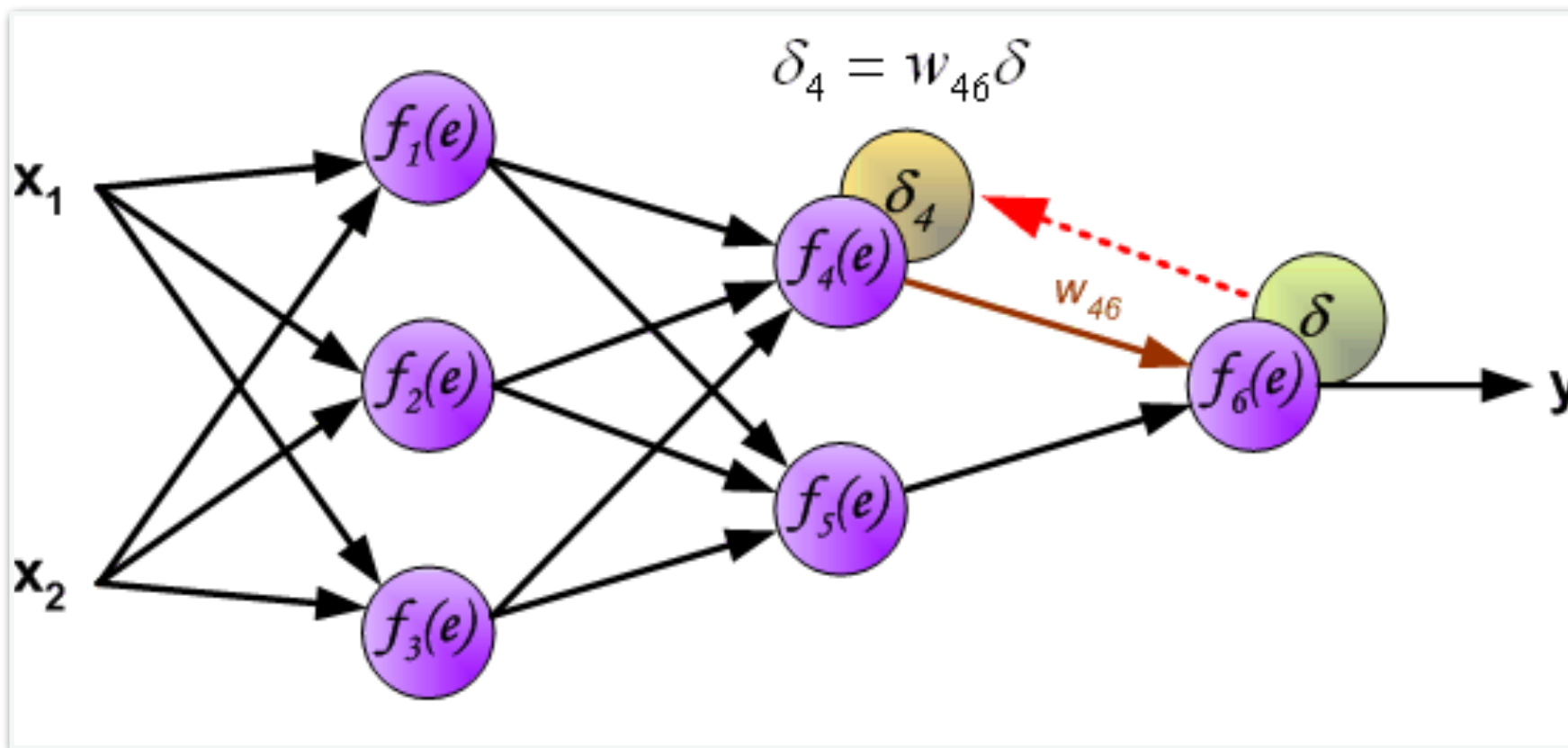
$x_1$	$x_2$	$h_{\Theta}(x)$
0	0	0
0	1	1
1	0	1
1	1	0

实例中，我们并非使用的是全连接神经网络，但是为了方便实际中我们往往使用全连接神经网络。

# 参数设置

手动设置参数?

NO!



实际中，我们并不手动设计神经网络的结构和参数。而是使用**反向传播算法**自动求参数。

# 小节

- 神经元的作用就是接收信号，变换信号，传出信号。
- 人工神经元接收多个输入并汇总，由激活函数处理后输出。
- 常用激活函数S型函数、双曲正切函数、修正线性单元、softmax函数。
- Sigmoid函数的值域决定了其可以做二分类。
- 神经网络是分层结构的。包括输入层、隐藏层、输出层。隐藏层可以是多层。
- 神经网络可以构造出逻辑单元并解决XOR问题。



# Python实现神经网络

# THANKS