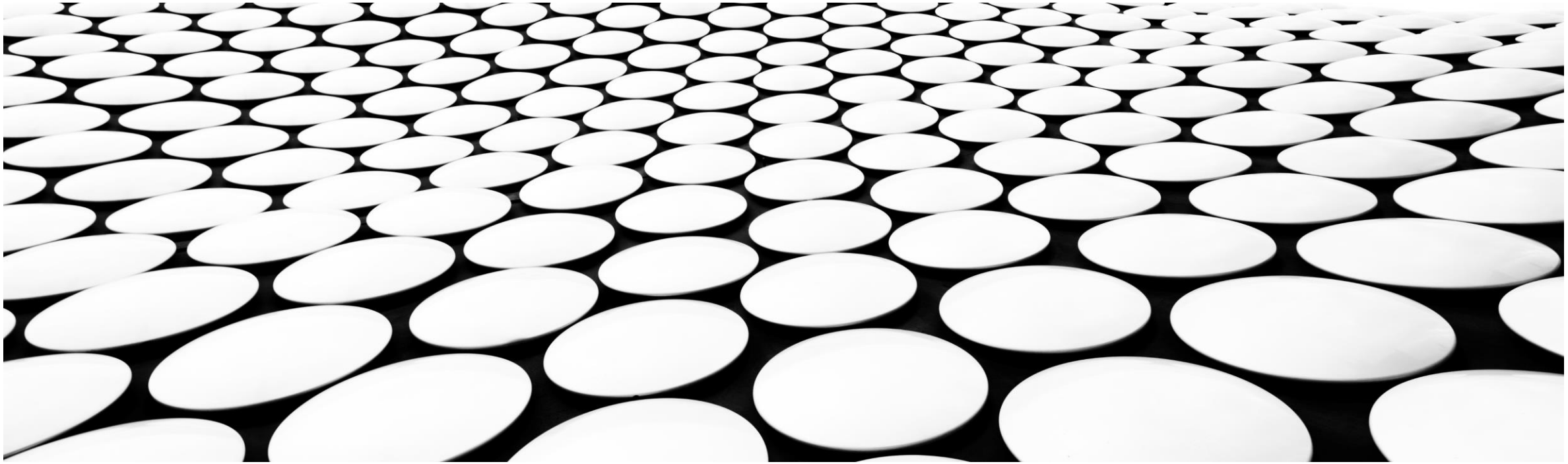
深度学习

邱怡轩



今天的主题

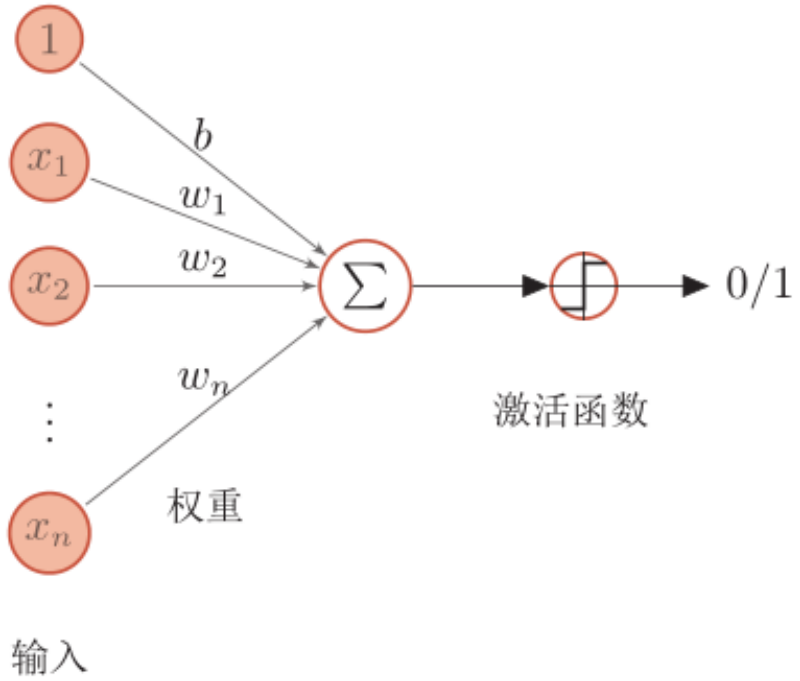
- 前馈神经网络

前馈神经网络

前馈神经网络 ⊂ 人工神经网络

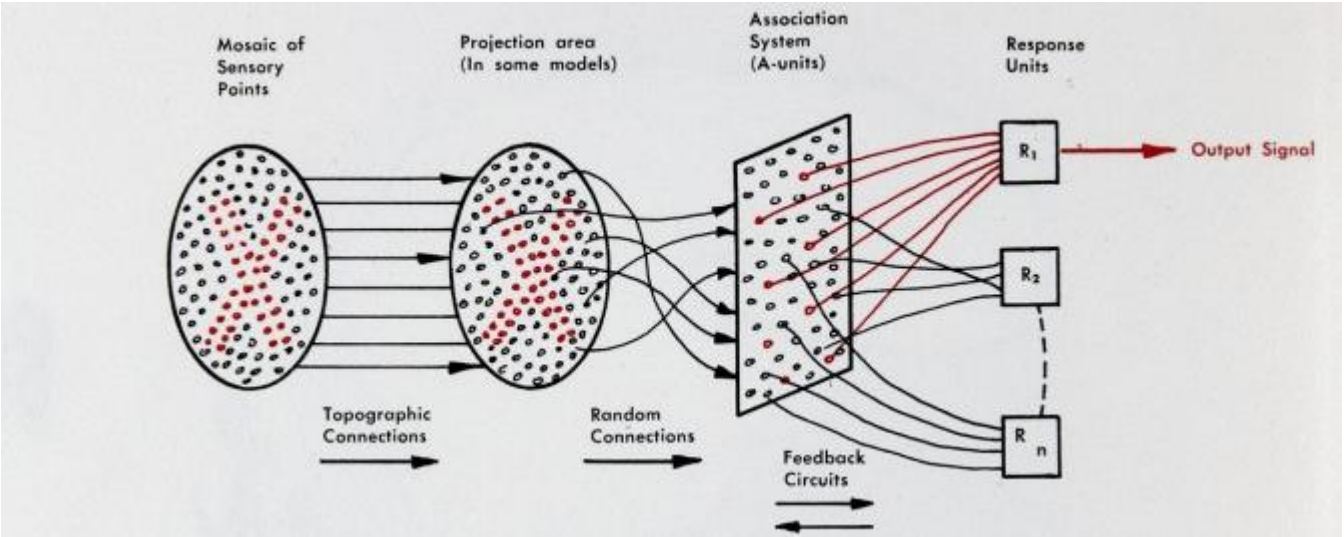
人工神经元

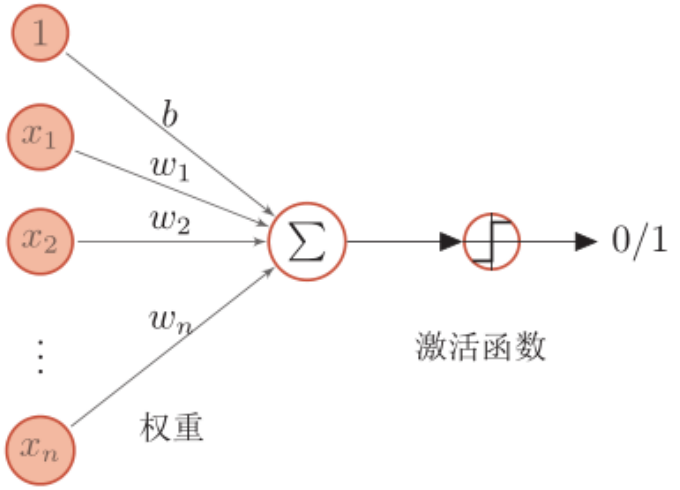
- 神经网络是连接主义模型的典型代表
- 基本构成单元为人工神经元
- “知识”存储在神经元之间的连接上
- 神经元负责信号/数据的输入和输出



人工神经元

- 现代的人工神经元结构基本源自Rosenblatt





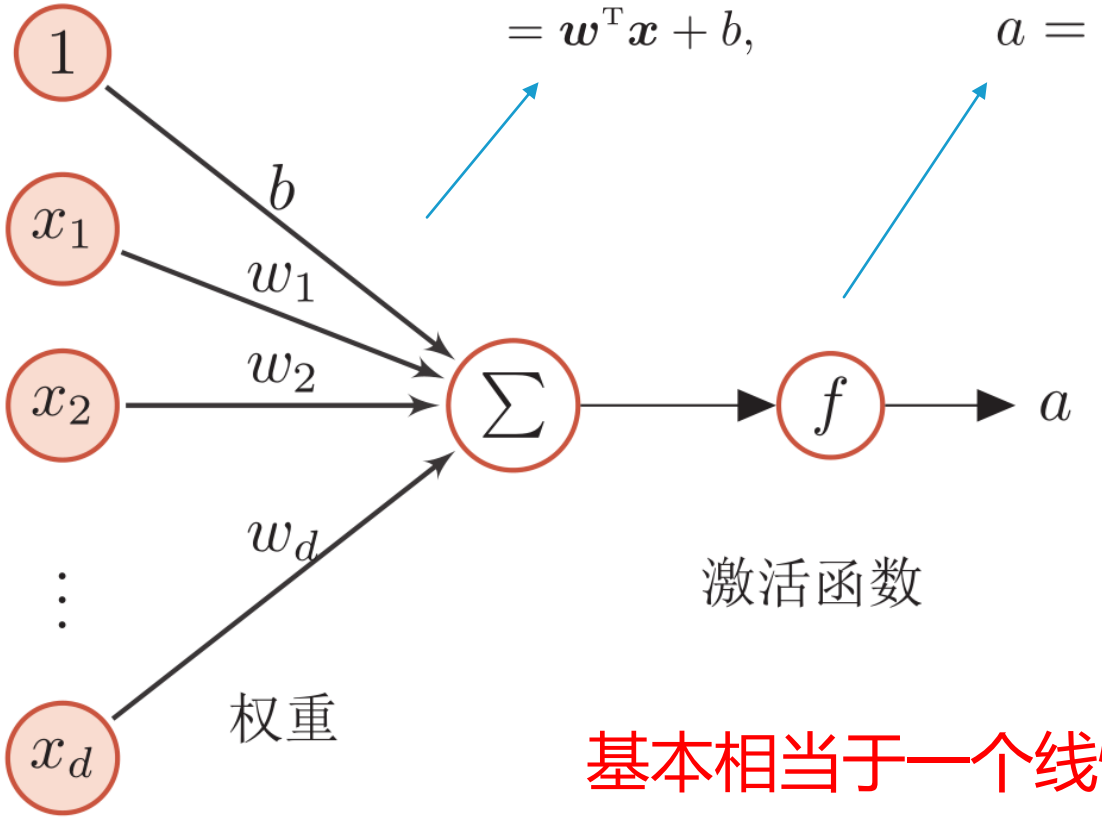
人工神经元

- 输入信号 → 加权求和 → 激活输出

$$z = \sum_{i=1}^{d} w_i x_i + b$$

$$= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b,$$

$$a = f(z)$$



基本相当于一个线性模型

激活函数

- Rosenblatt当时的激活函数是一个阶梯函数，

  例如 $a = f(z) = \begin{cases} 1, & z \geq 0.5 \\ 0, & z < 0.5 \end{cases}$

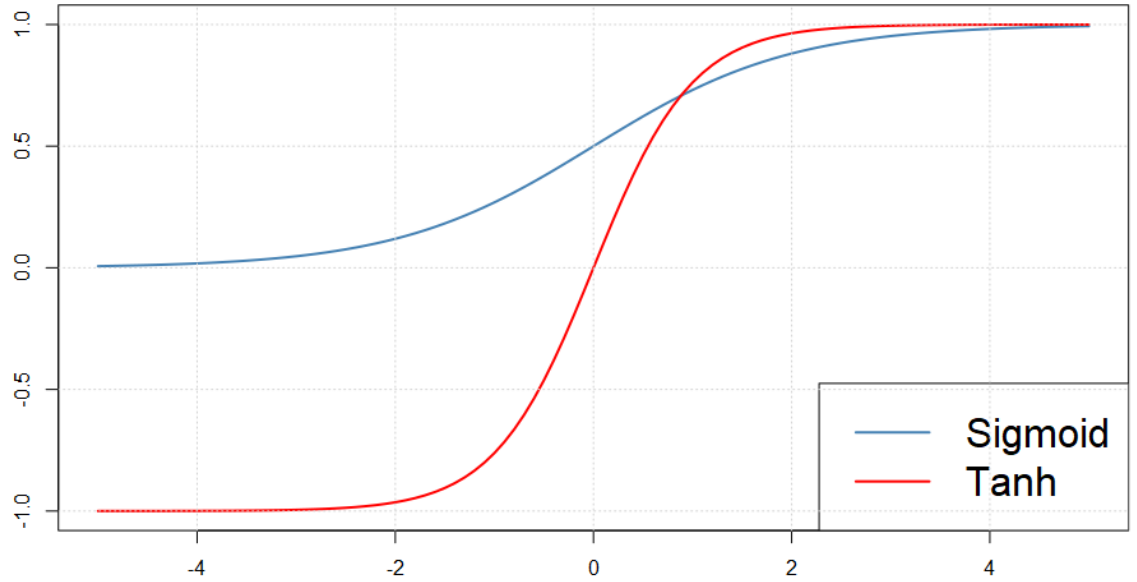
- 当今广泛使用的激活函数一般满足如下性质：
  - 非线性
  - 连续，除了少数点外处处可导
  - 计算简单、数值稳定

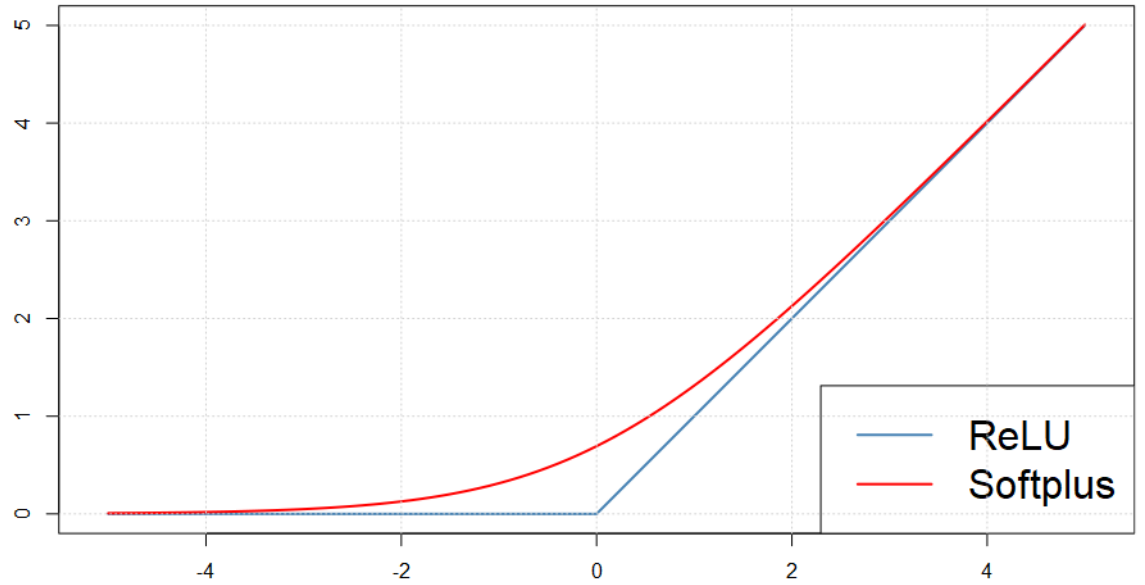
常见激活函数

- Sigmoid $$\sigma(x) = \frac{1}{1+\exp(-x)}$$
- Tanh $$\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$$
- ReLU $$\begin{aligned}\mathrm{ReLU}(x) &= \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \\ &= \max(0, x).\end{aligned}$$
- Softplus $$\mathrm{softplus}(x) = \log(1+\exp(x))$$

常见激活函数

- Sigmoid
- Tanh
- ReLU
- Softplus





不知道怎么选择时

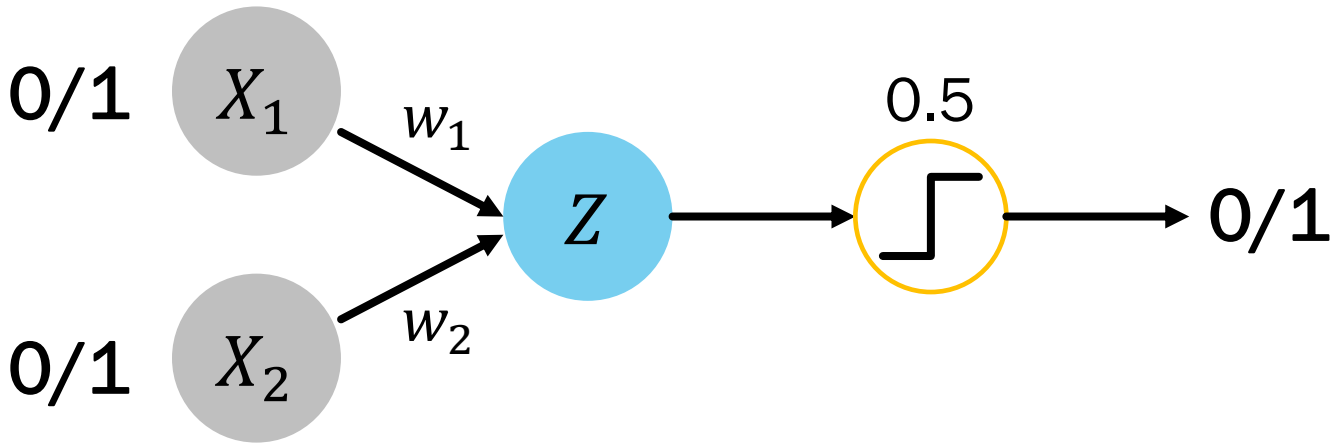
就以ReLU作为默认选项

编程实现

- 在编程实现中需要特别注意数值稳定性
- 特别是牵涉到指数函数 exp(x)

思考题

- 试着用Rosenblatt的感知器模拟逻辑运算
- 两个输入：$X_1 \in \{0,1\},\ X_2 \in \{0,1\}$
- 线性函数：$z = w_1X_1 + w_2X_2$
- 一个输出：若$z > 0.5$，则$Y = 1$，否则$Y = 0$



思考题

- 尝试找出适当的参数w_1和w_2，使得这个人工神经元可以实现

1. AND运算
2. OR运算
3. XOR运算

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

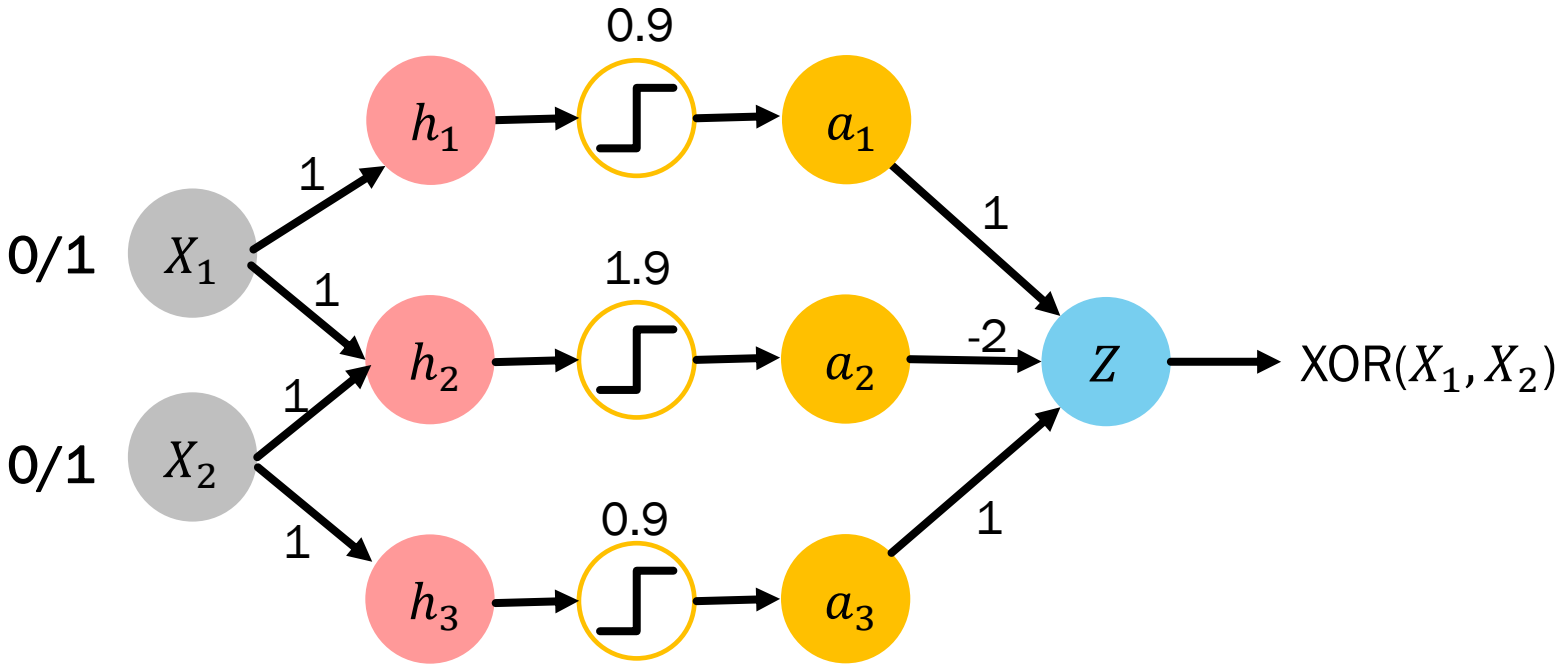
INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

启示

- 单层的感知器具有很大的局限
- 可以将神经元串联起来
- 构建多层的神经网络

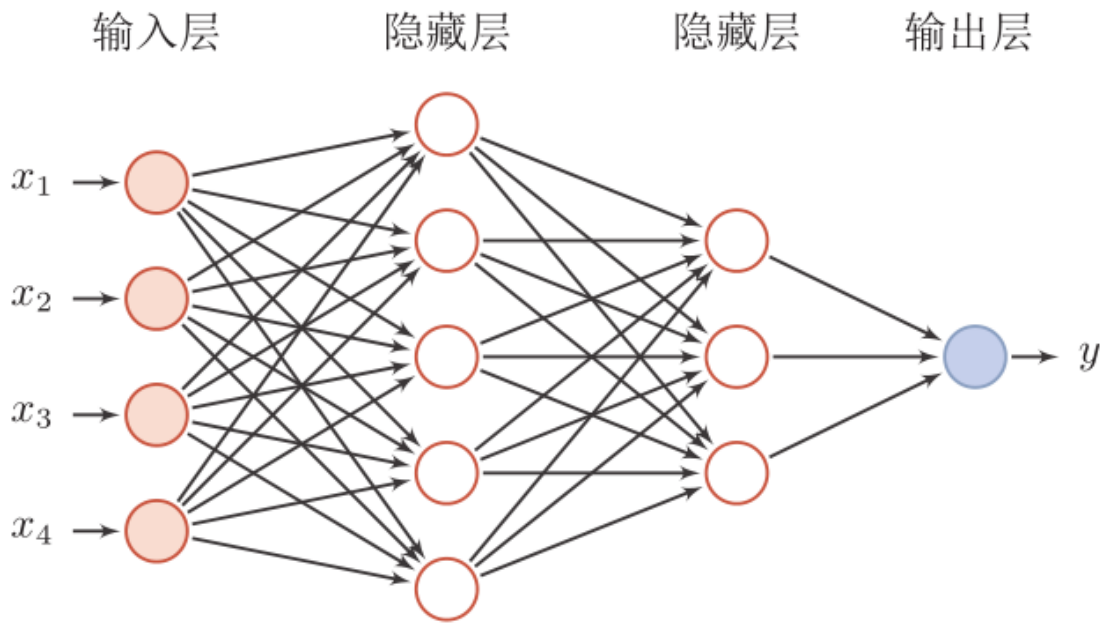
XOR问题

- 只需增加一层神经元，即可模拟XOR运算



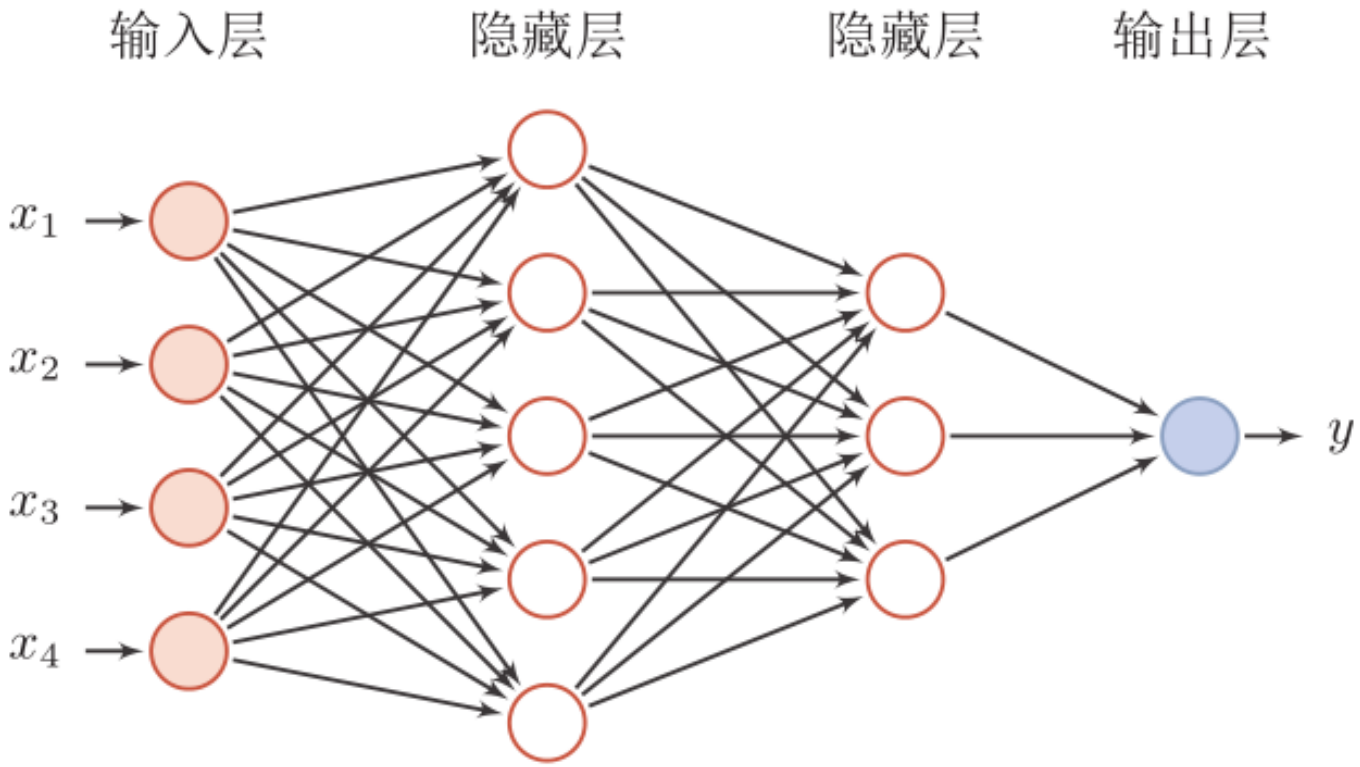
前馈神经网络

- 将神经元按照一定的规则组合，可以得到复杂的神经网络
- 前馈神经网络是一种结构相对简单的神经网络
- 也称为全连接神经网络、多层感知器



前馈神经网络

- 各神经元分别属于不同的层，层内无连接
- 相邻两层之间的神经元全部两两连接
- 信号从输入层向输出层单向传播



正式定义

本页内容取自邱锡鹏
《神经网络与深度学习》

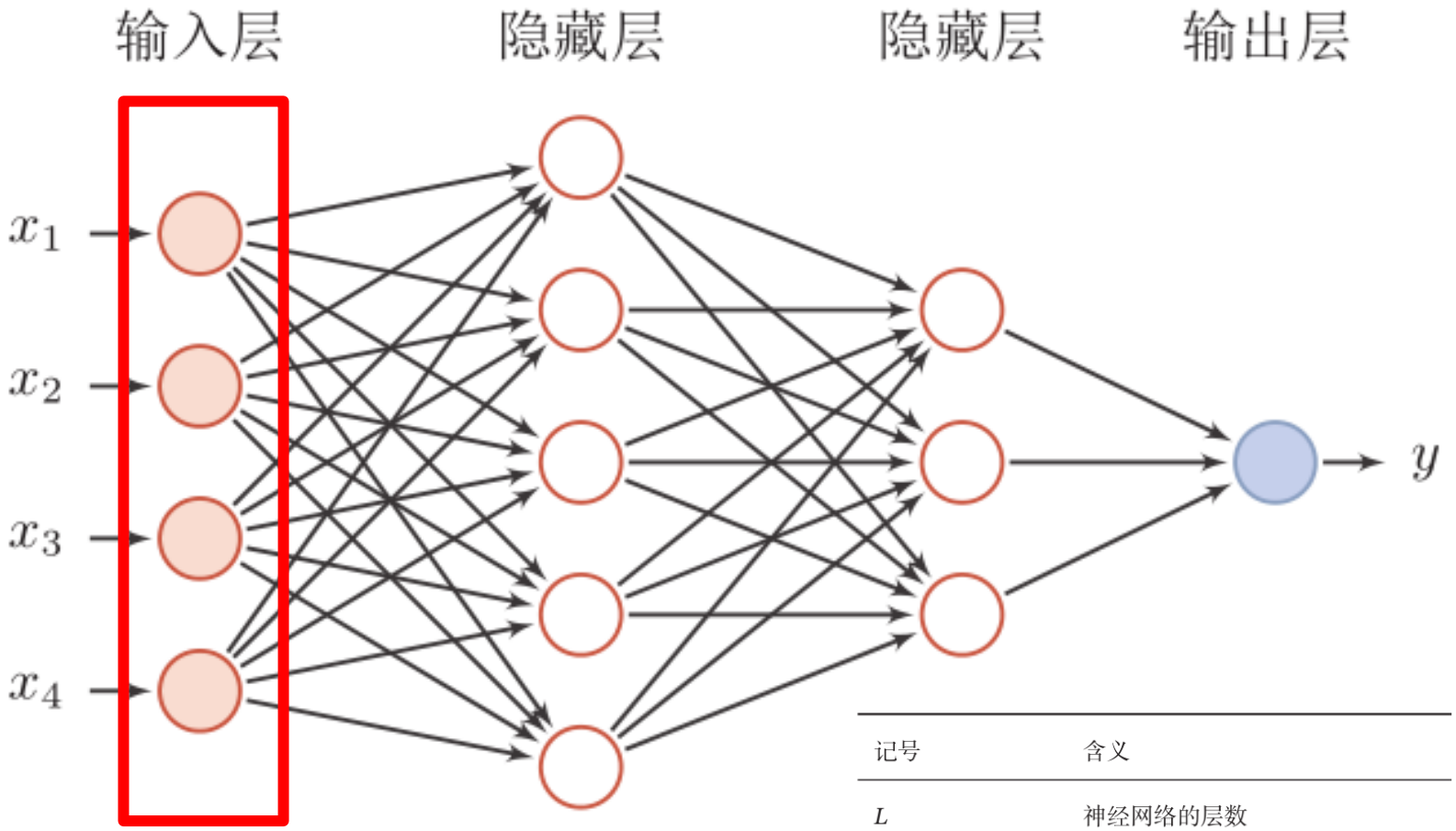
记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\boldsymbol{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\boldsymbol{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入（净活性值）
$\boldsymbol{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出（活性值）

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)},$$
$$\boldsymbol{a}^{(l)} = f_l(\boldsymbol{z}^{(l)}).$$

- 信息前馈传播方式

$$\boldsymbol{x} = \boldsymbol{a}^{(0)} \to \boldsymbol{z}^{(1)} \to \boldsymbol{a}^{(1)} \to \boldsymbol{z}^{(2)} \to \cdots \to \boldsymbol{a}^{(L-1)} \to \boldsymbol{z}^{(L)} \to \boldsymbol{a}^{(L)} = \phi(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}))$$

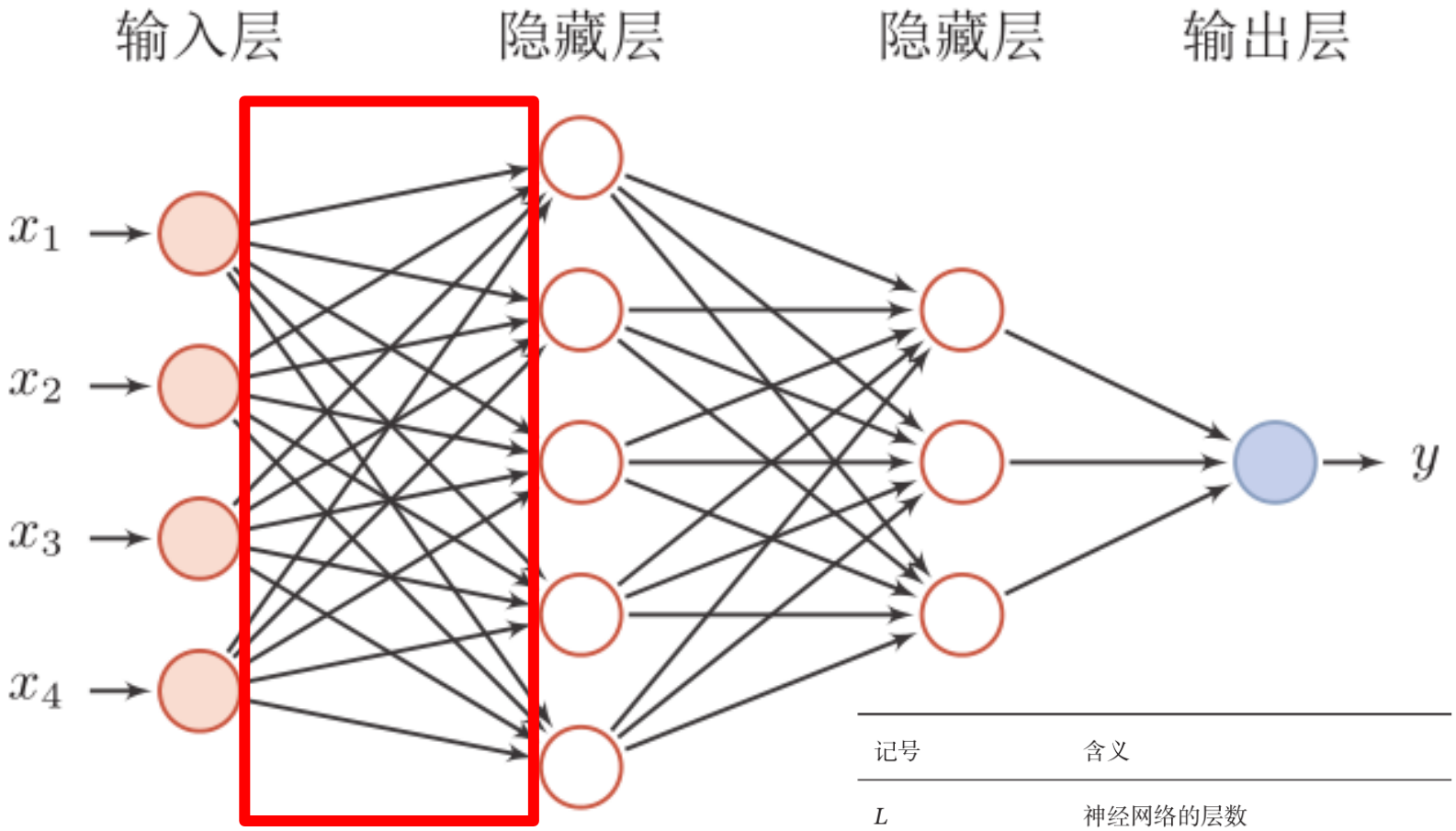
前馈神经网络



$$a^{(0)} = x \in \mathrm{R}^4$$

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\boldsymbol{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\boldsymbol{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入（净活性值）
$\boldsymbol{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出（活性值）

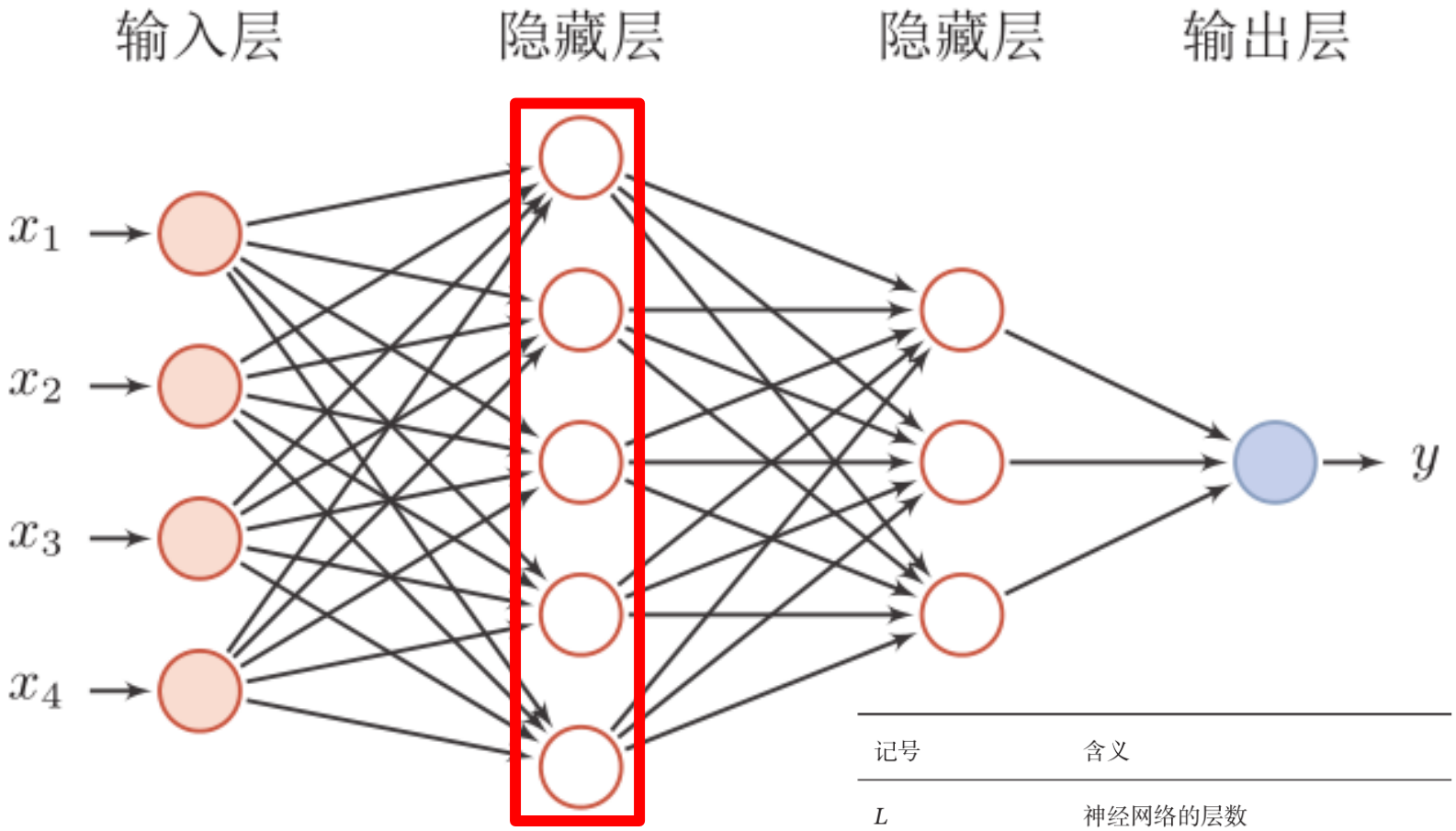
前馈神经网络



$$W^{(1)} \in \mathrm{R}^{5\times4}$$

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\boldsymbol{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\boldsymbol{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入（净活性值）
$\boldsymbol{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出（活性值）

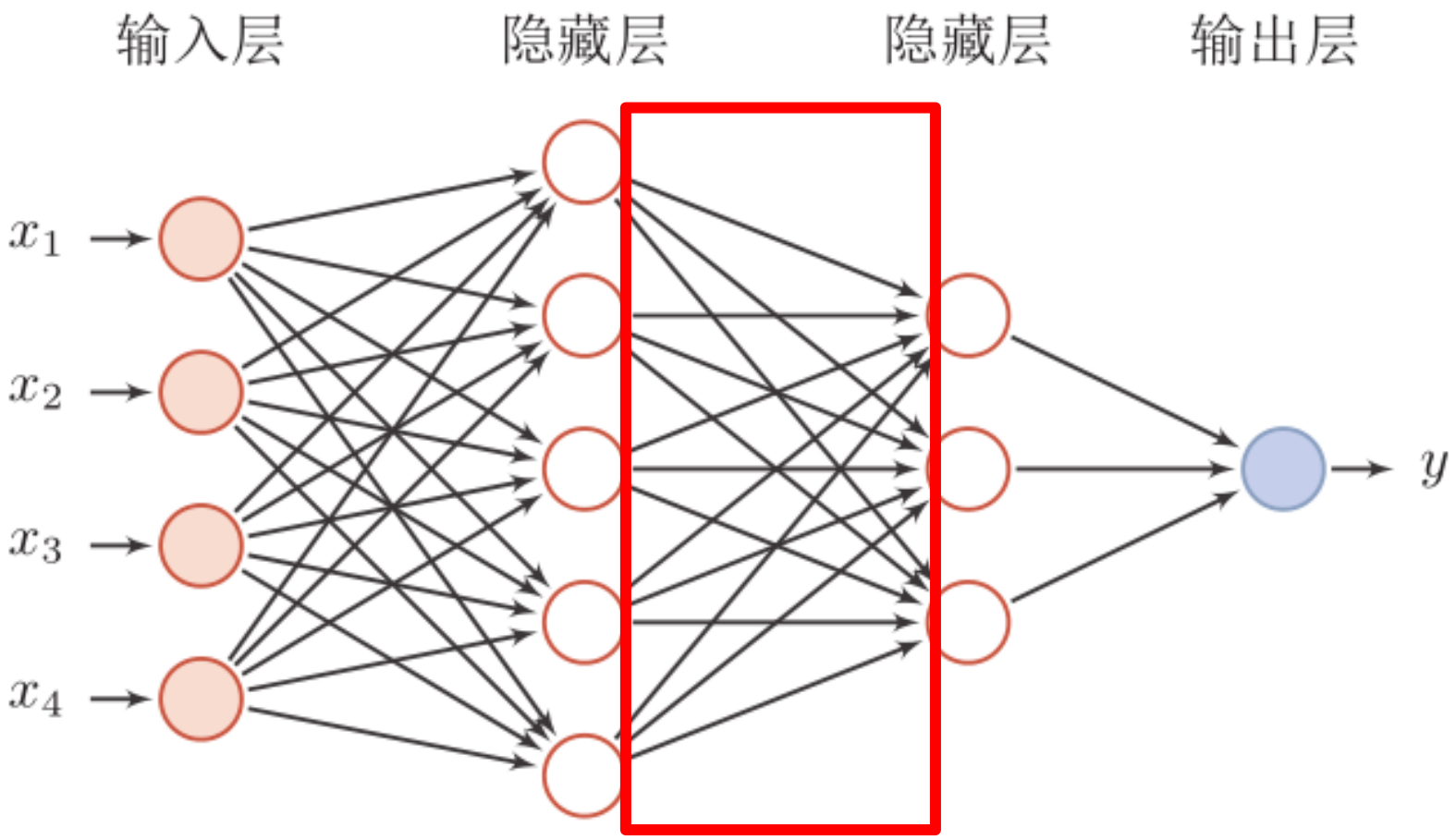
前馈神经网络



$$a^{(1)} \in \mathrm{R}^5$$

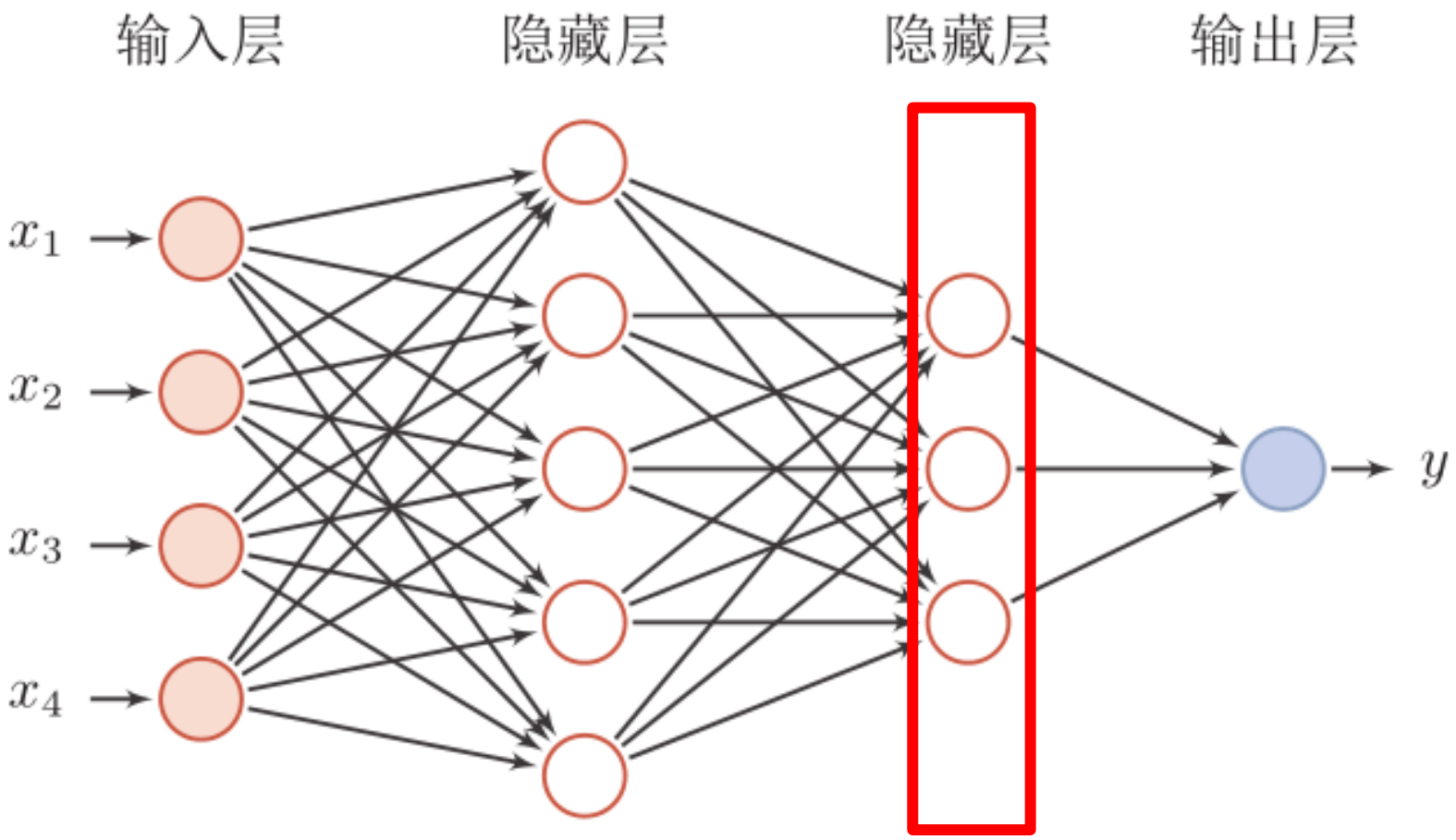
记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\boldsymbol{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\boldsymbol{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入（净活性值）
$\boldsymbol{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出（活性值）

前馈神经网络



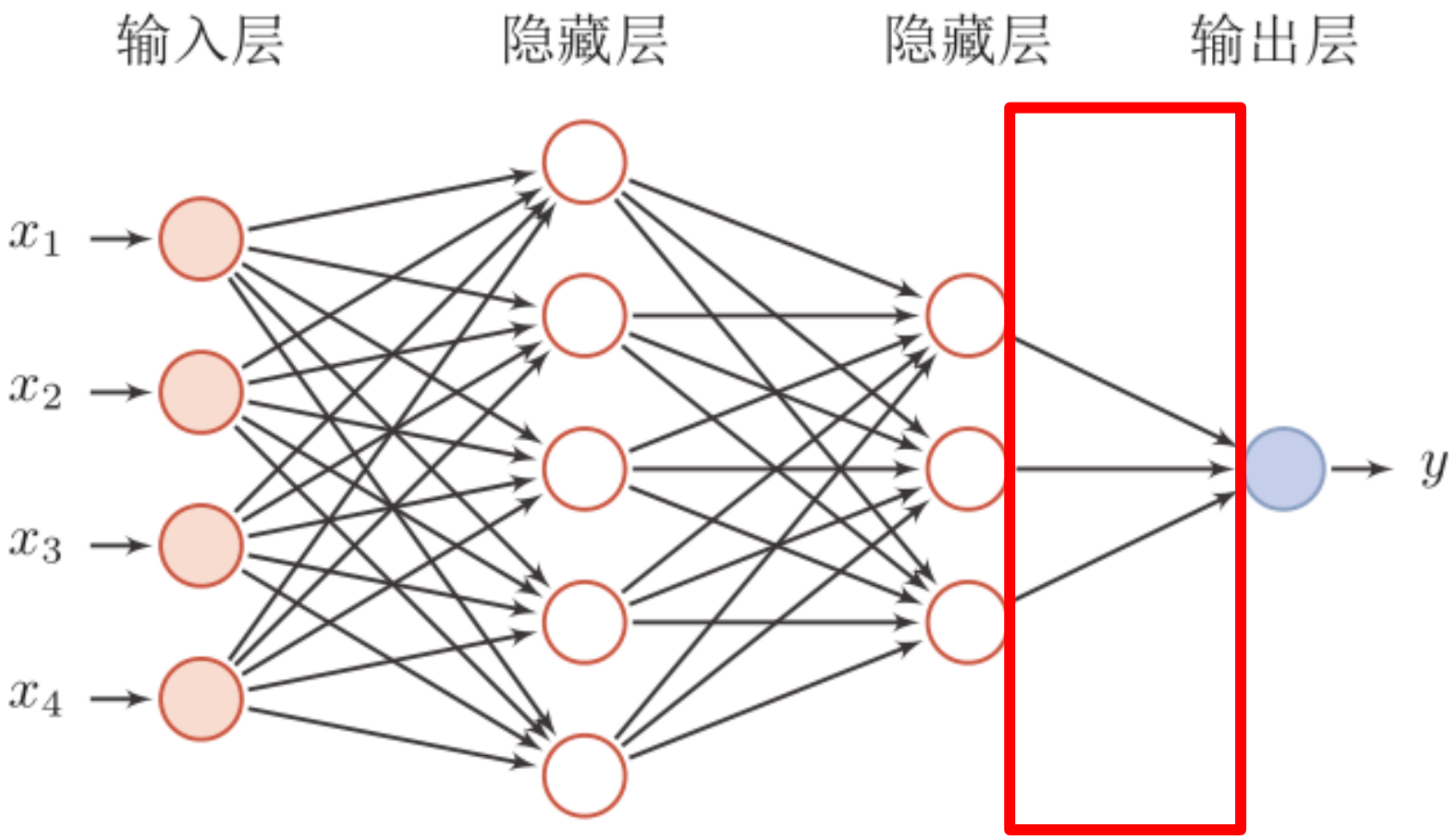
$$W^{(2)} \in \mathrm{R}^{3\times5}$$

前馈神经网络



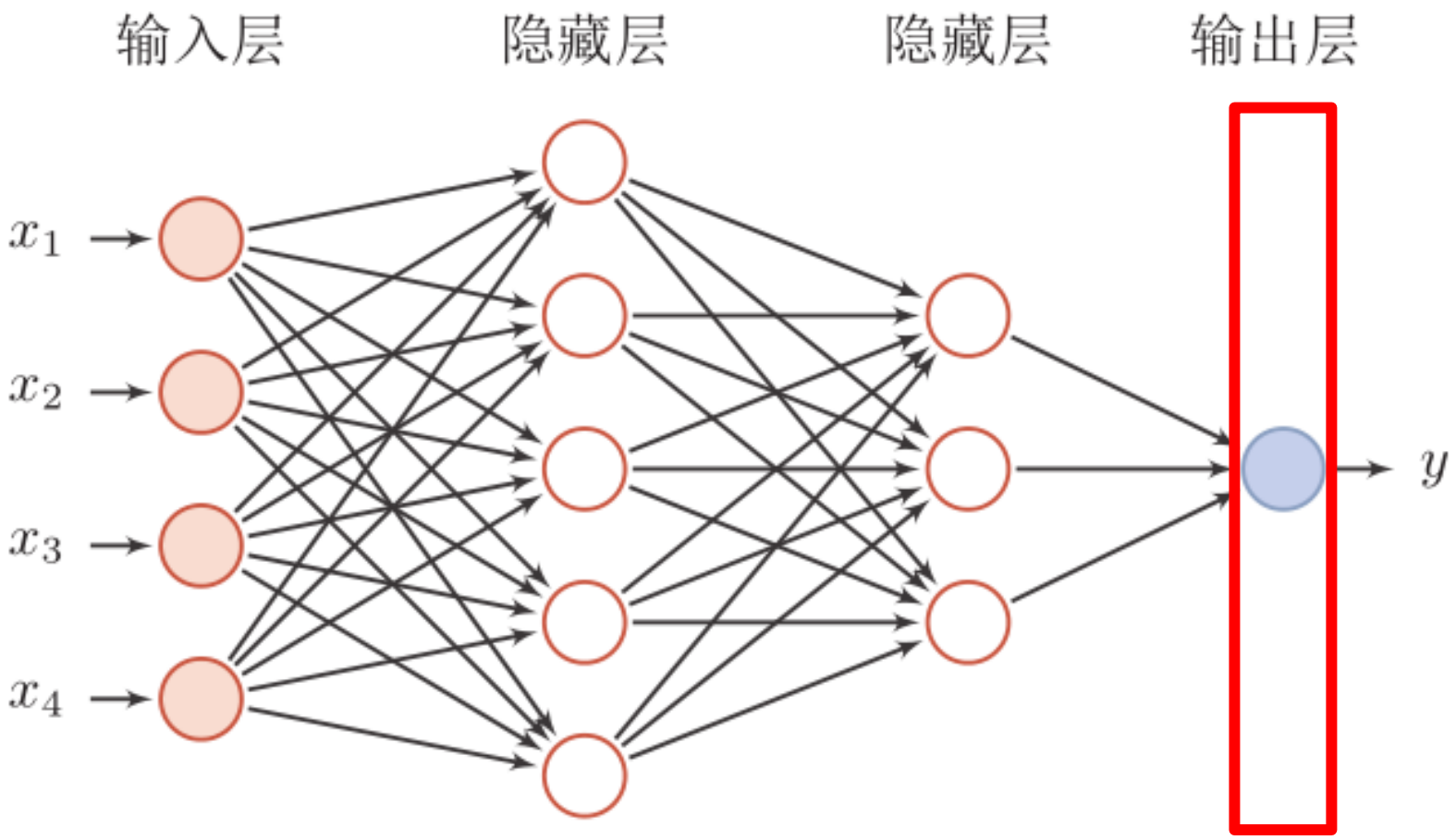
$a^{(2)} \in \mathrm{R}^3$

前馈神经网络



$W^{(3)} \in \mathrm{R}^{1\times 3}$

前馈神经网络



$a^{(3)} \in \mathrm{R}$

前馈神经网络

- 本质上，前馈神经网络是一个复合函数
- $y = f^5(f^4(f^3(f^2(f^1(x)))))$
- 每一个 f^i 可以是固定的（如激活函数），或者是带参数的（如带权重的线性变换）
- 每一个 f^i 都相对简单
- 但复合之后可以变得非常复杂

通用近似定理

本页定理取自邱锡鹏
《神经网络与深度学习》

- 之前我们展示了，一个两层的前馈神经网络可以拟合XOR函数
- 事实上，两层的网络几乎可以拟合任意函数！

定理 4.1 – 通用近似定理（Universal Approximation Theorem）[Cybenko, 1989, Hornik et al., 1989]： 令 $\varphi(\cdot)$ 是一个非常数、有界、单调递增的连续函数，$\mathcal{I}_d$ 是一个 d 维的单位超立方体 $[0,1]^d$，$C(\mathcal{I}_d)$ 是定义在 $\mathcal{I}_d$ 上的连续函数集合。对于任何一个函数 $f \in C(\mathcal{I}_d)$，存在一个整数 m，和一组实数 $v_i, b_i \in \mathbb{R}$ 以及实数向量 $\mathbf{w}_i \in \mathbb{R}^d$，$i = 1, \cdots, m$，以至于我们可以定义函数

$$F(\mathbf{x}) = \sum_{i=1}^{m} v_i \varphi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i), \tag{4.33}$$

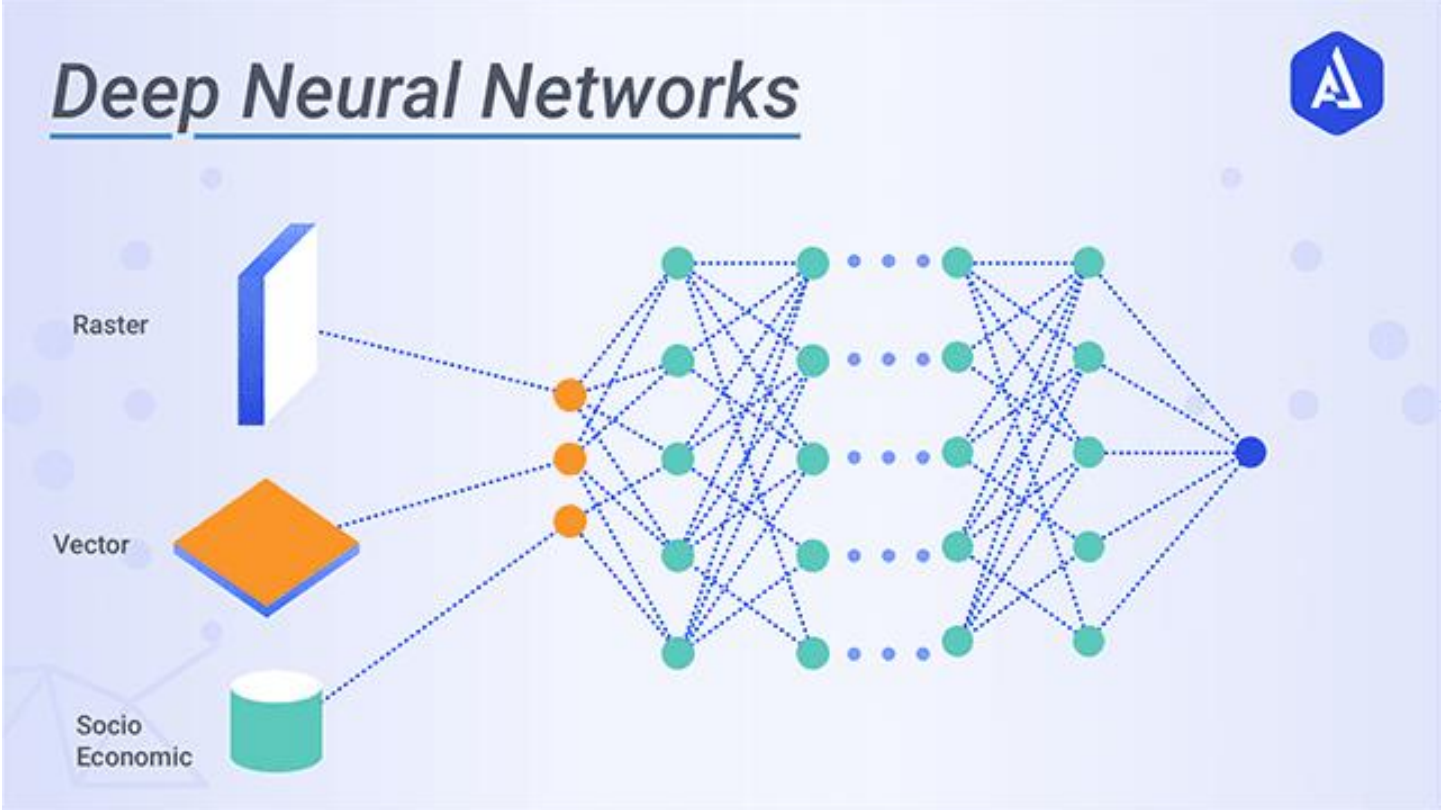
作为函数 f 的近似实现，即

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon, \forall \mathbf{x} \in \mathcal{I}_d. \tag{4.34}$$

其中 $\epsilon > 0$ 是一个很小的正数。

深度神经网络

- 根据通用近似定理，两层的前馈神经网络已经可以近似绝大部分函数
- 但一些新的研究表明，网络的深度可能发挥着重要的作用



深度神经网络

- 对于网络深度的研究是当前的热点课题

模型训练

- 神经网络作为一个函数类具有良好的性质
- “万能函数”
- 接下来的问题在于如何估计其中的参数

- $\hat{y} = \varphi(x;\theta)$
- $L = \ell(y,\hat{y}) = \ell(y,\varphi(x;\theta))$

模型训练

- 神经网络作为一个函数类具有良好的性质
- “万能函数”
- 接下来的问题在于如何估计其中的参数

神经网络

神经网络参数，如权重

- $\hat{y} = \varphi(x;\theta)$
- $L = \ell(y,\hat{y}) = \ell(y,\varphi(x;\theta))$

损失函数

梯度下降法

- $\theta_{k+1} = \theta_k - \alpha_t \cdot \left.\frac{\partial \ell(y,\varphi(x;\theta))}{\partial \theta}\right|_{\theta=\theta_k}$
- 核心在于求损失函数对参数的导数