第二章：深度学习反向传播
——优化论五部曲

Jason 博士

网易微专业 x 稀牛学院

人工智能数学基础微专业

回归与分类

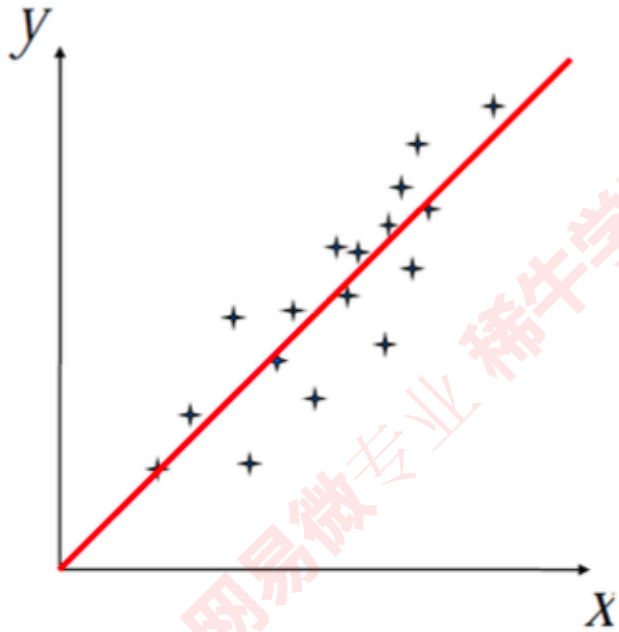
神经网络

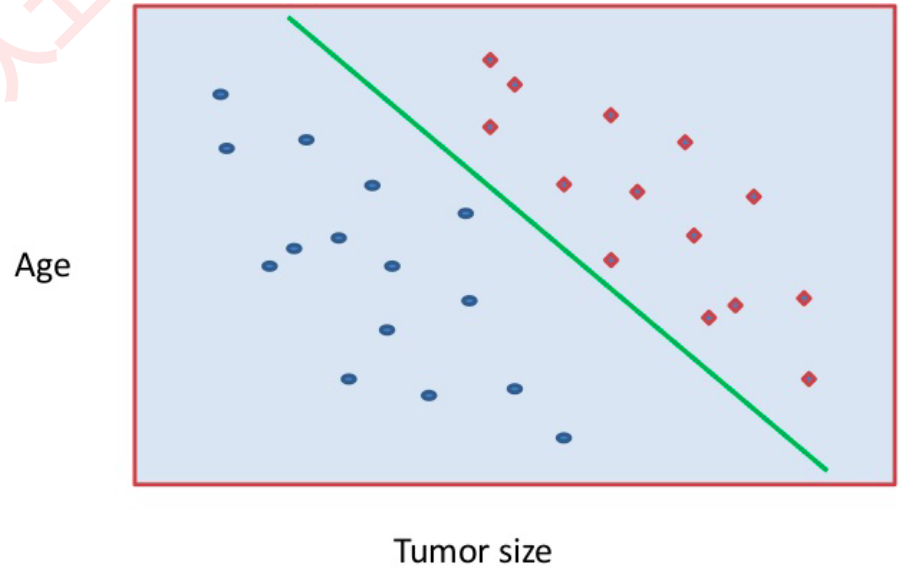
BP 算法

计算图

回归 VS 分类

- 给定训练样本集 $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}, \mathbf{x}_i \in \mathbb{R}^d$
  - 回归：预测的是连续值
  - 分类：预测的是离散值
    - 二分类：$y_i \in \{0, 1\}$





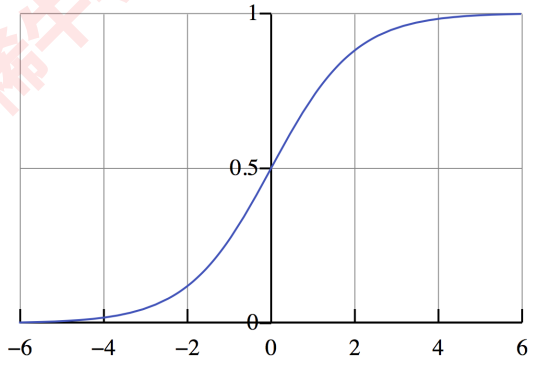
Sigmoid 函数

- $y = \mathbf{w}^T\mathbf{x} + b$
- 非线性映射 1

$$z = \begin{cases} 0 & y < 0 \\ 0.5 & y = 0 \\ 1 & y > 0 \end{cases}$$

- 非线性映射 2

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



S 函数求导 (课堂手推)

- $\sigma(x) = \frac{1}{1+e^{-x}}$
- $\frac{1}{dx}\sigma(x) = \sigma(x) - \sigma(x)^2$

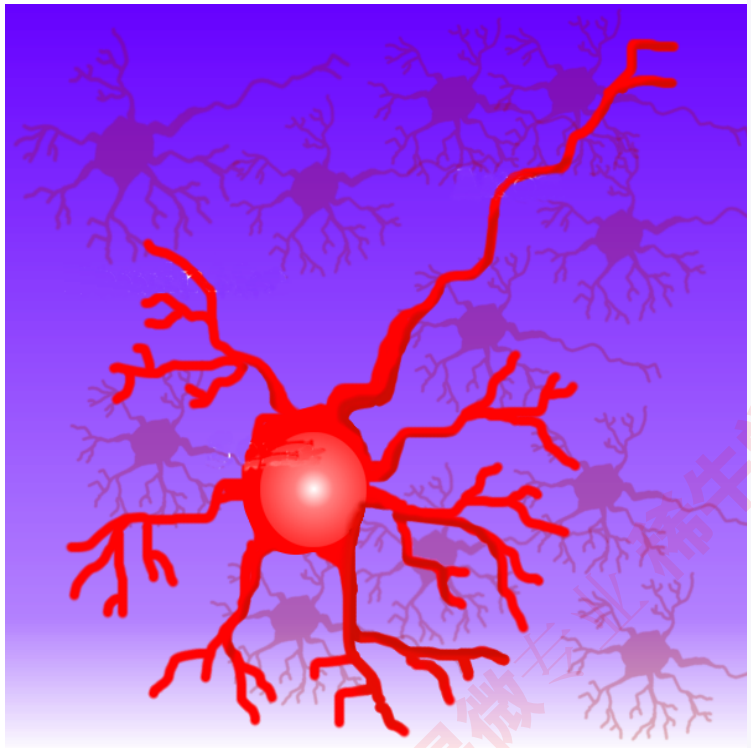
回归与分类

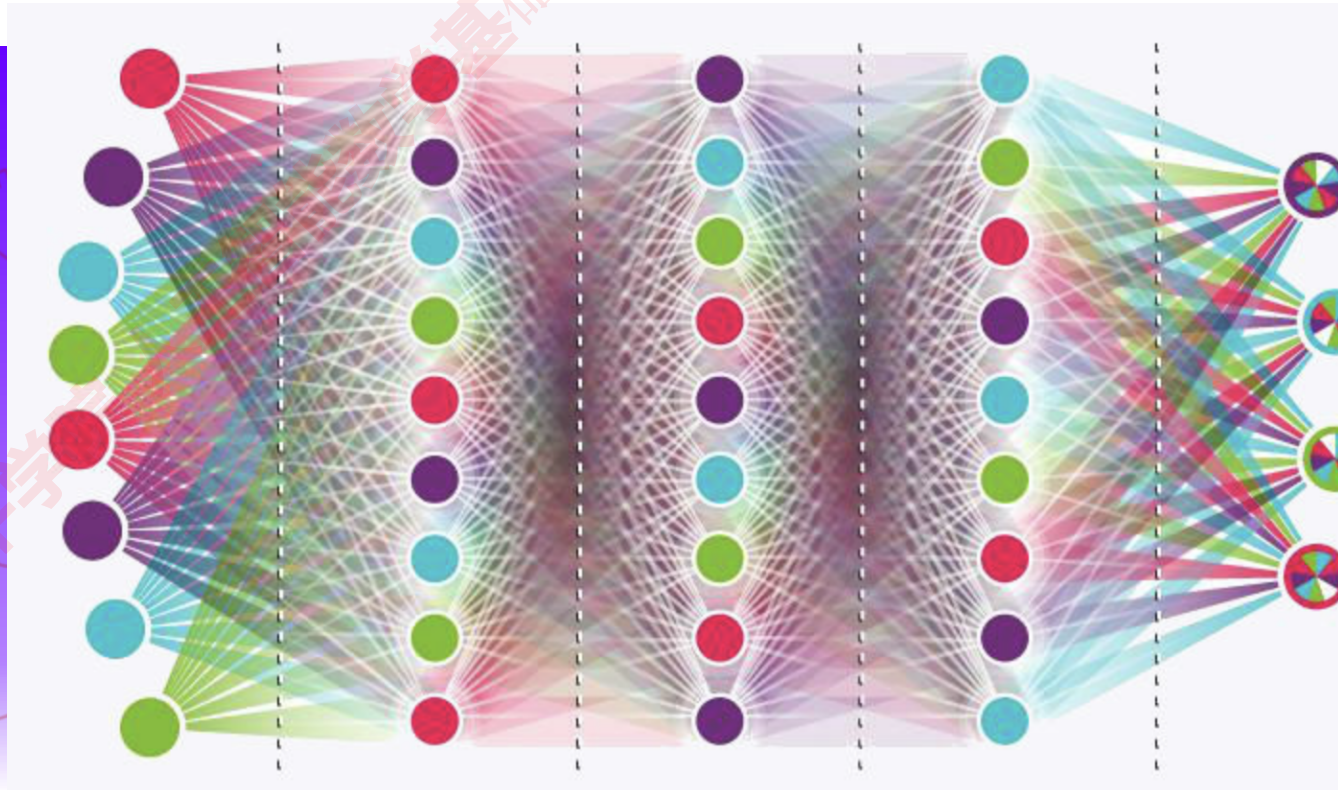
神经网络

BP 算法

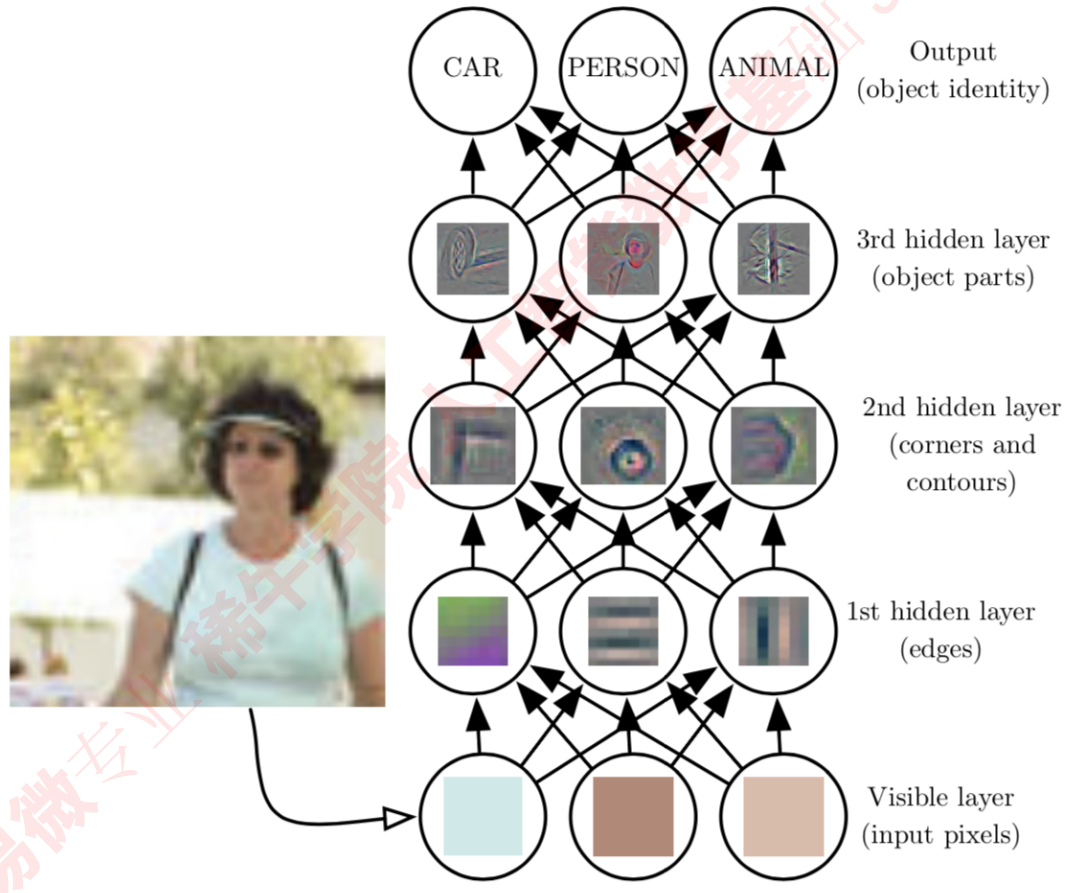
计算图

架构



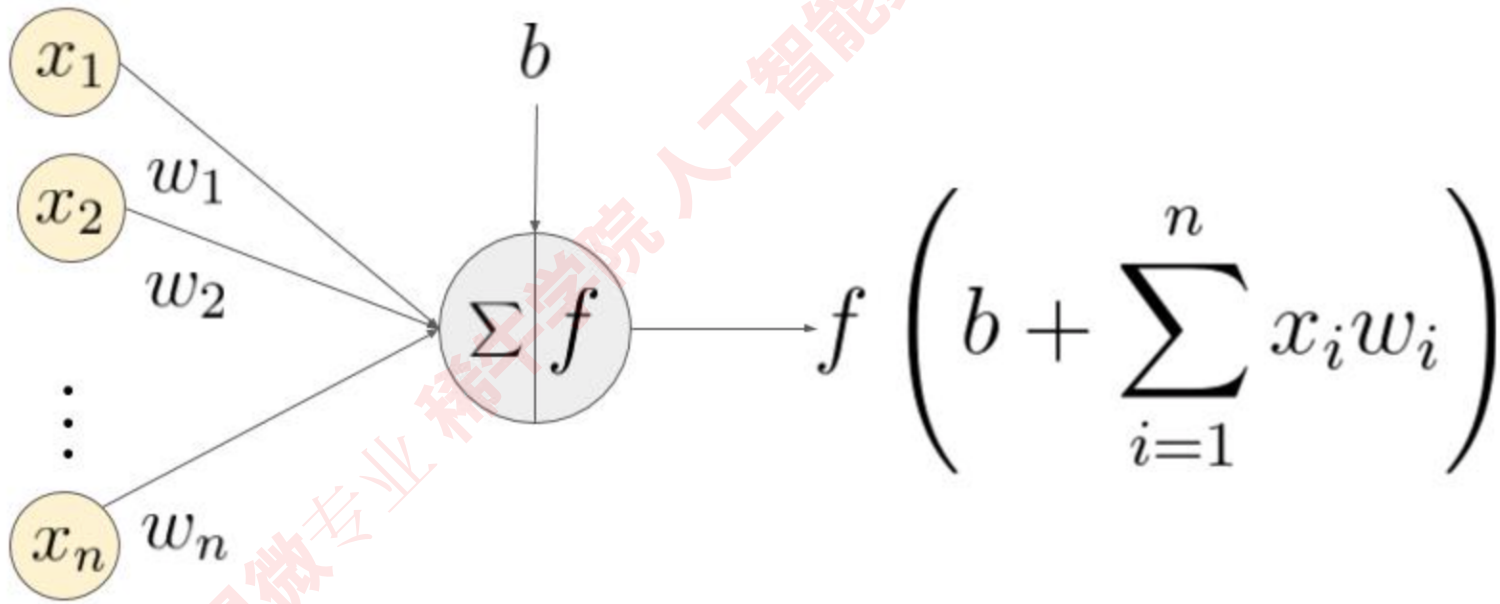


功能 (DL 例子)

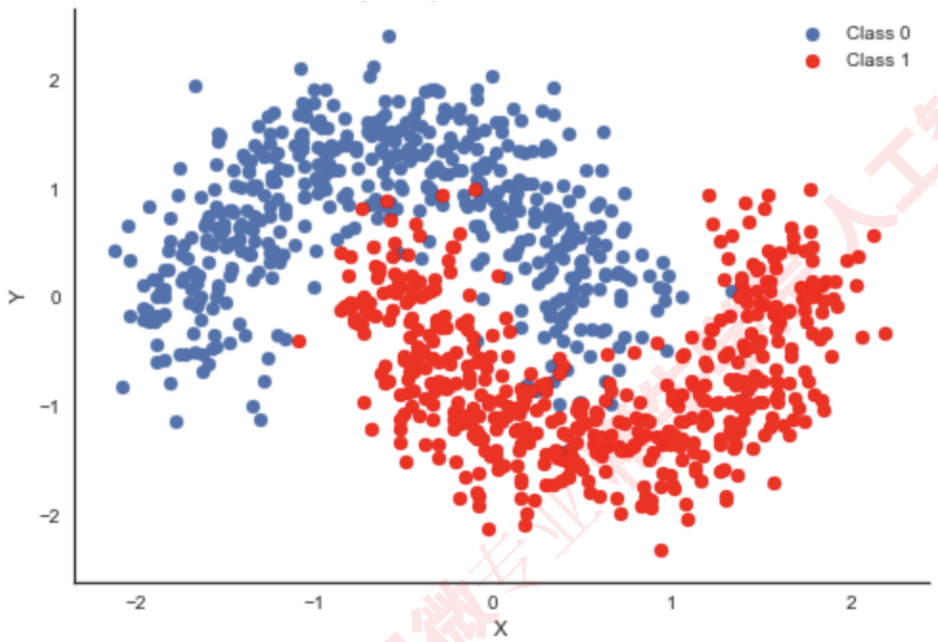


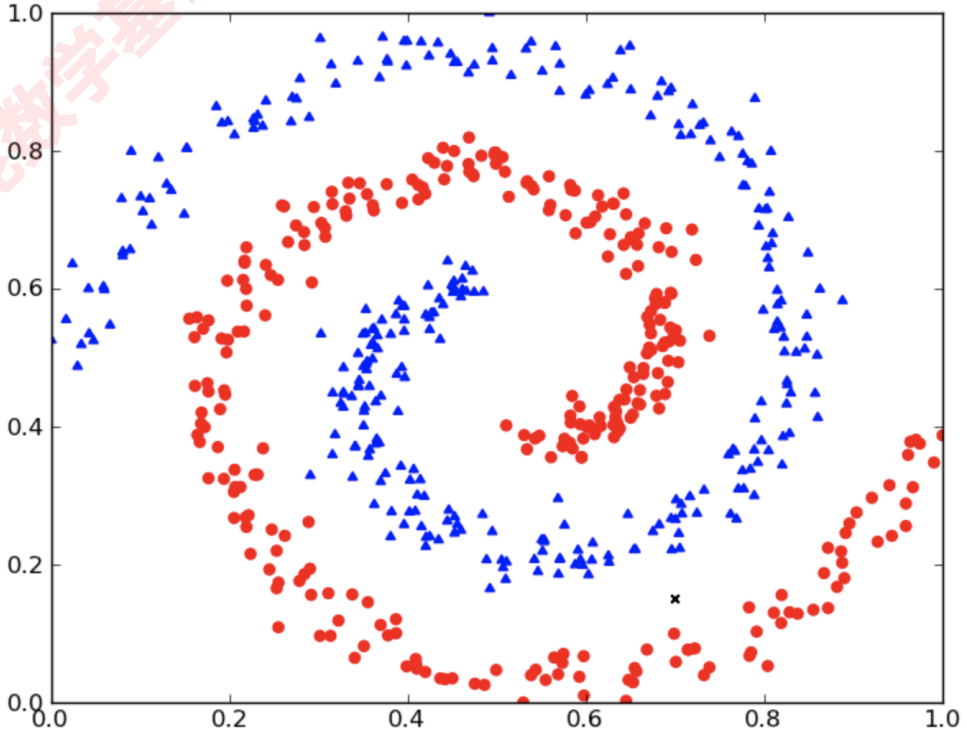
神经元

- 激活函数
- 相互嵌套

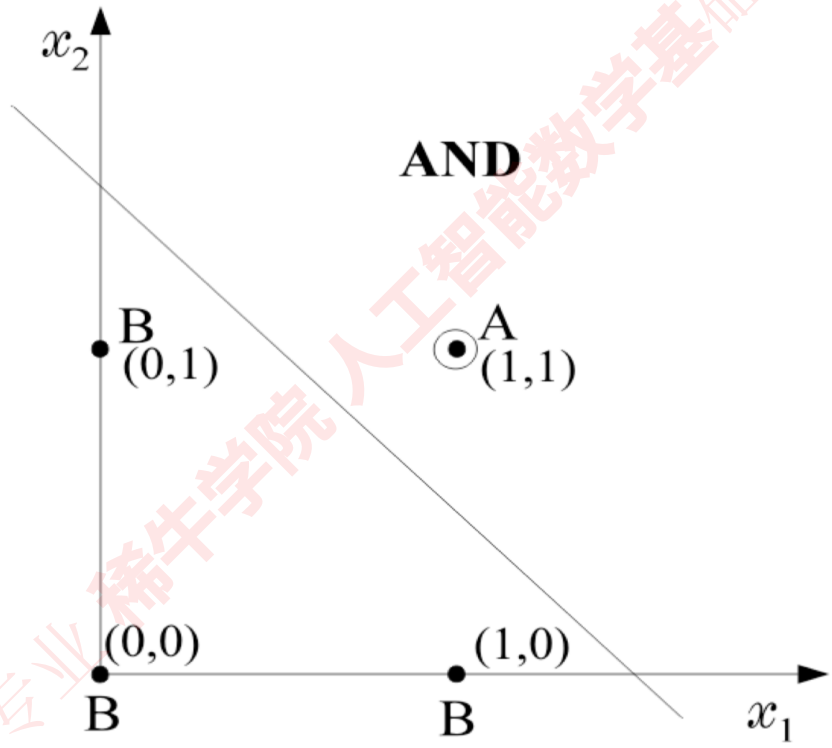


切分原理

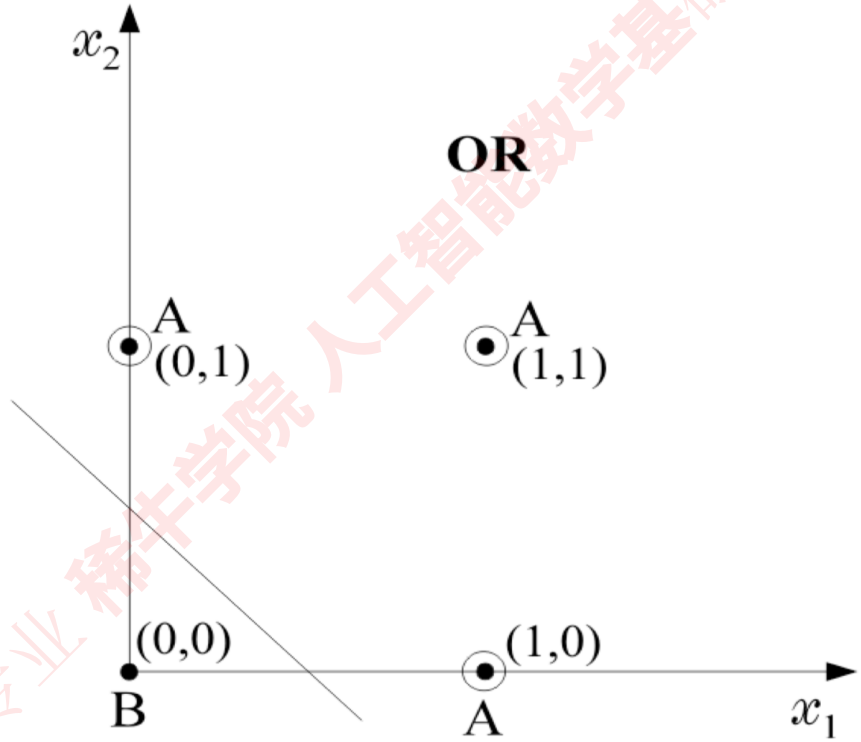




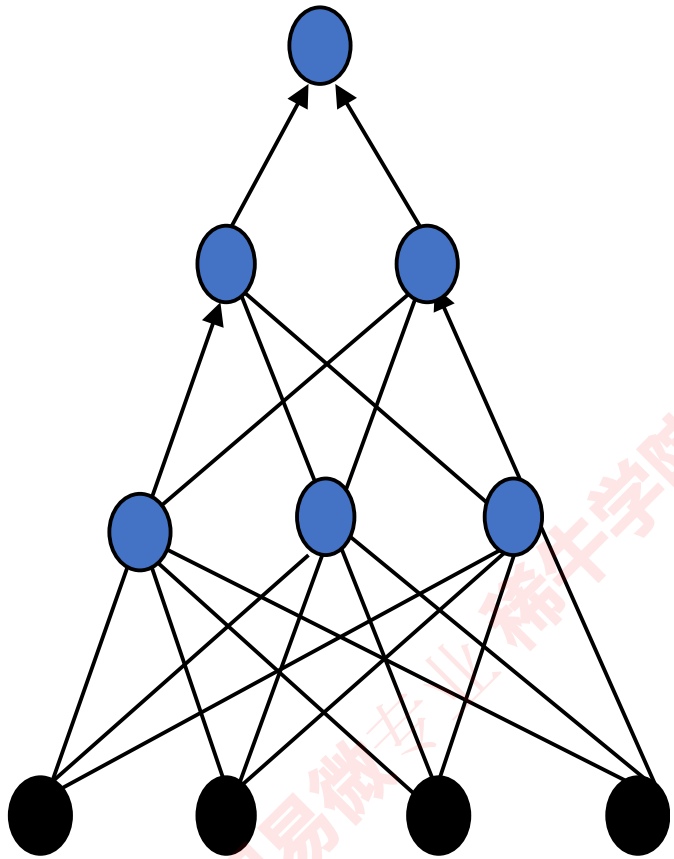
逻辑与

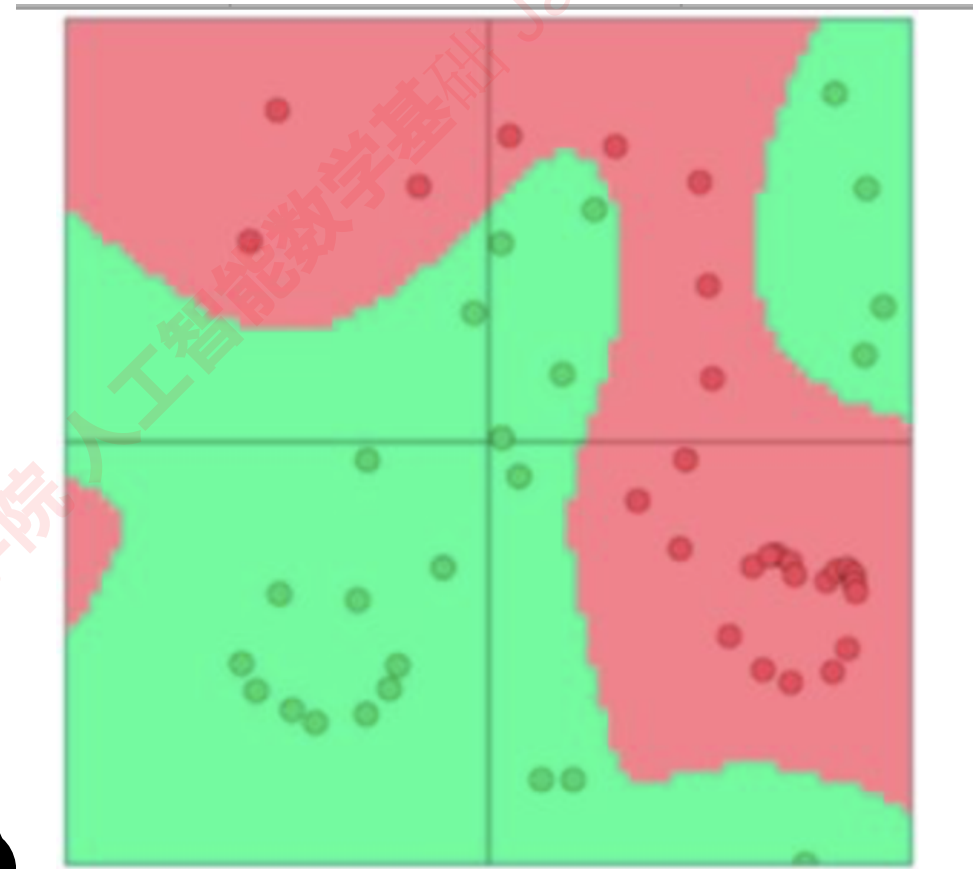


逻辑或

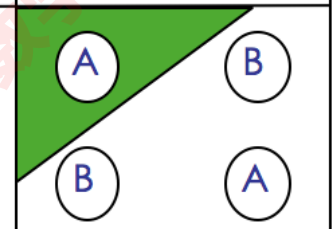
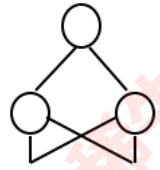
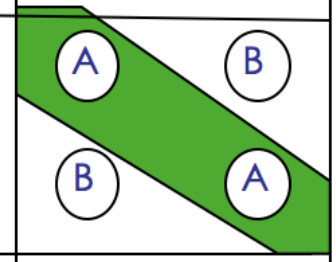
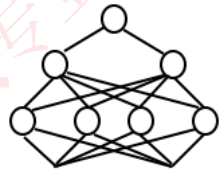
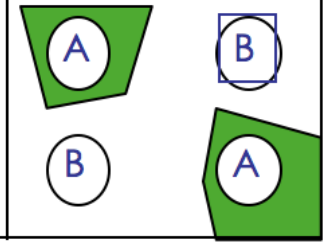


空间切分





切分原理

Structure	Types of Decision Regions	Exclusive-OR Problem
Single-Layer 	Half Plane Bounded By Hyper plane	
Two-Layer 	Convex Open Or Closed Regions	
Three-Layer 	Arbitrary (Complexity Limited by No. of Nodes)	

回归与分类

神经网络

BP 算法

计算图

(Back Propagation, BP) 算法 (手推 1)

(Back Propagation, BP) 算法 (手推 2)

(Back Propagation, BP) 算法 (手推 3)

BP 算法数学描述总结

符号定义

- x_j^l: 层 l 下的节点 j 的输入
- $w_{i,j}^l$: 从层 $l-1$ 中的节点 i 到层 l 中的节点 j 这一段的权重
- $\sigma(x) = \frac{1}{1+e^{-x}}$
- θ_j^l : 层 l 下的节点 j 的偏置
- O_j^l: 层 l 下的节点 j 的输出
- t_j: 目标值
- $E = \frac{1}{2}\sum_{k \in K}(O_k - t_k)^2$

BP(输出层)

- 目标是计算：$\frac{\partial E}{\partial w_{j,k}}$
- 具体计算

$$
\begin{aligned}
\frac{\partial E}{\partial w_{j,k}} &= \frac{\partial}{\partial O_k}\frac{1}{2}\sum_{k\in K}\left(O_k - t_k\right)^2 \times \frac{\partial}{\partial x_k}O_k \times \frac{\partial}{\partial w_{j,k}}x_k \\
&= \left(O_k - t_k\right) \times \frac{\partial}{\partial x_k}\sigma\left(x_k\right) \times O_j \\
&= \left(O_k - t_k\right) \times \sigma\left(x_k\right)\left(1 - \sigma\left(x_k\right)\right) \times O_j \\
&= \underbrace{\left(O_k - t_k\right) O_k \left(1 - O_k\right)}_{\delta_k} O_j
\end{aligned}
$$

BP(隐层)

- 目标是计算：$\frac{\partial E}{\partial w_{i,j}}$
- 具体计算

$$
\begin{aligned}
\frac{\partial E}{\partial w_{i,j}} &= O_j\left(1-O_j\right) O_i \sum_{k \in K}\left(O_k-t_k\right) O_k\left(1-O_k\right) w_{j,k} \\
&= O_i O_j\left(1-O_j\right) \sum_{k \in K} \delta_k w_{j,k} \\
&= O_i \delta_j
\end{aligned}
$$

BP 总结

- 对于输出层节点 $k \in K$

$$\frac{\partial E}{\partial w_{j,k}} = O_j \delta_k$$

其中 $\delta_k = (O_k - t_k)\, O_k\, (1 - O_k)$

- 对于隐层节点 $j \in J$

$$\frac{\partial E}{\partial w_{i,j}} = O_i \delta_j$$

其中 $\delta_j = O_j\, (1 - O_j) \sum_{k \in K} \delta_k w_{j,k}$

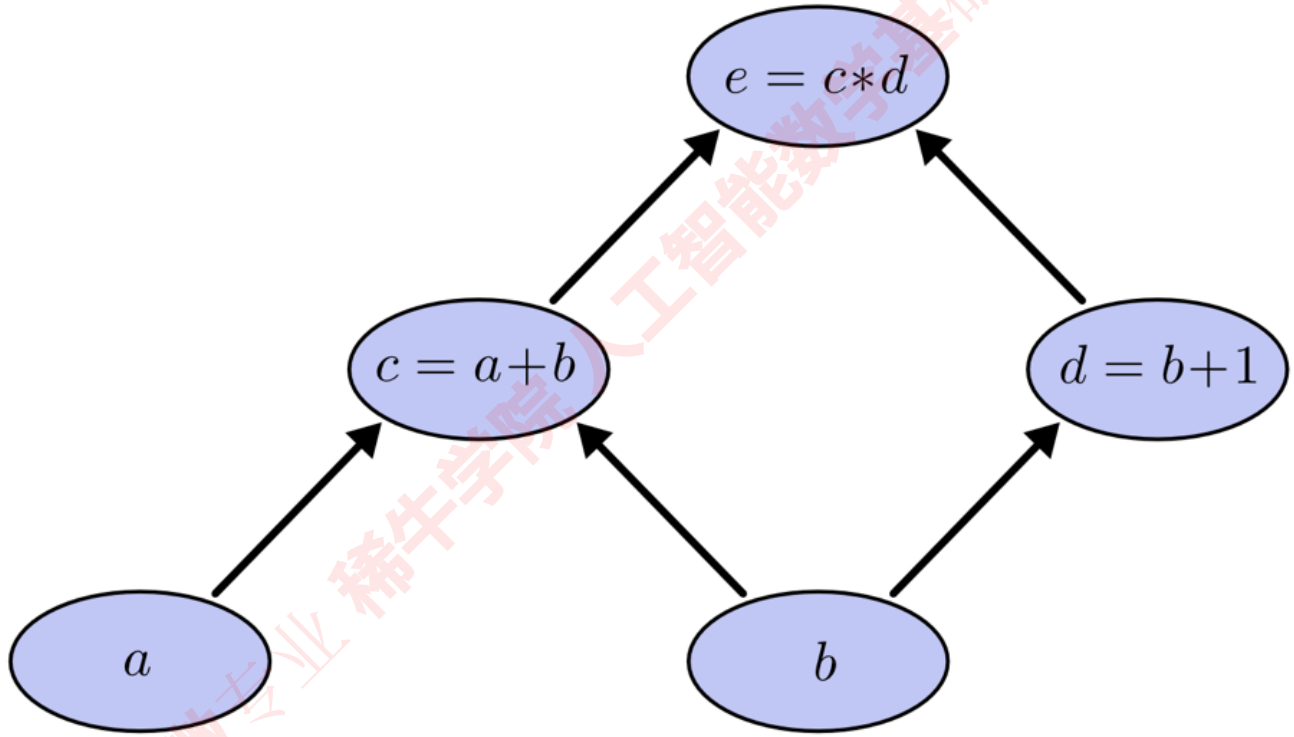
回归与分类

神经网络

BP 算法

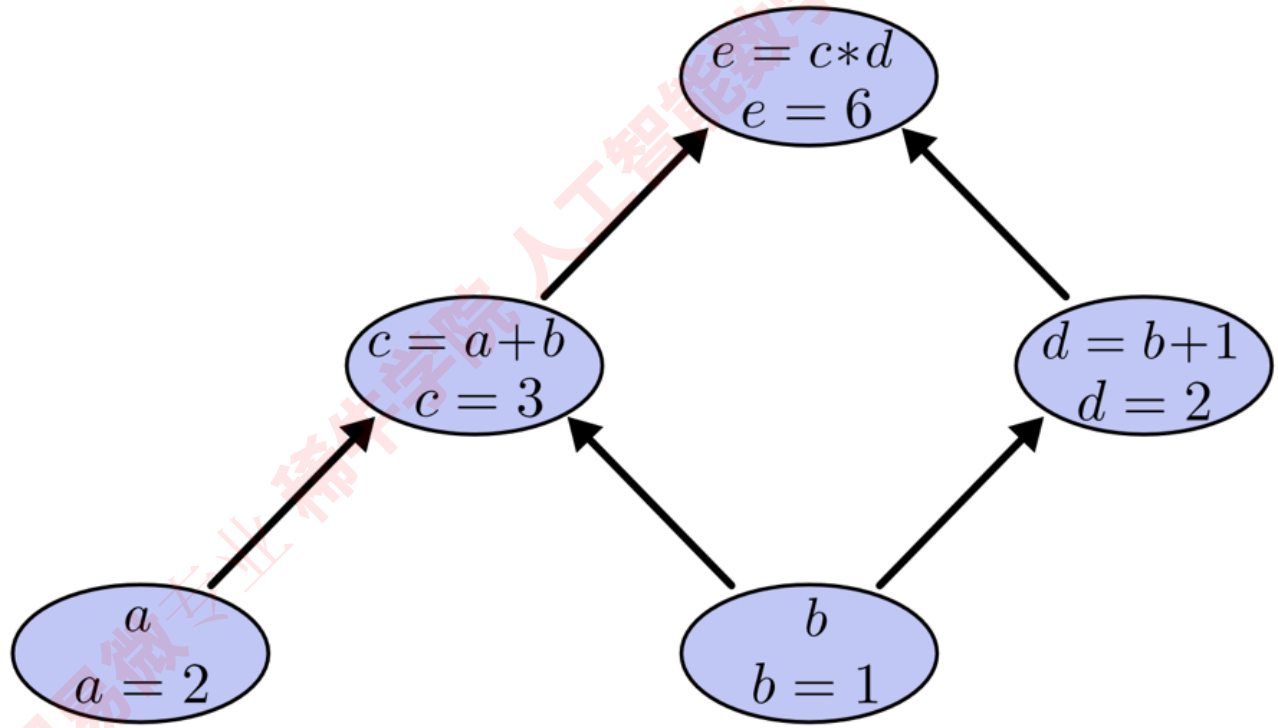
计算图

计算图

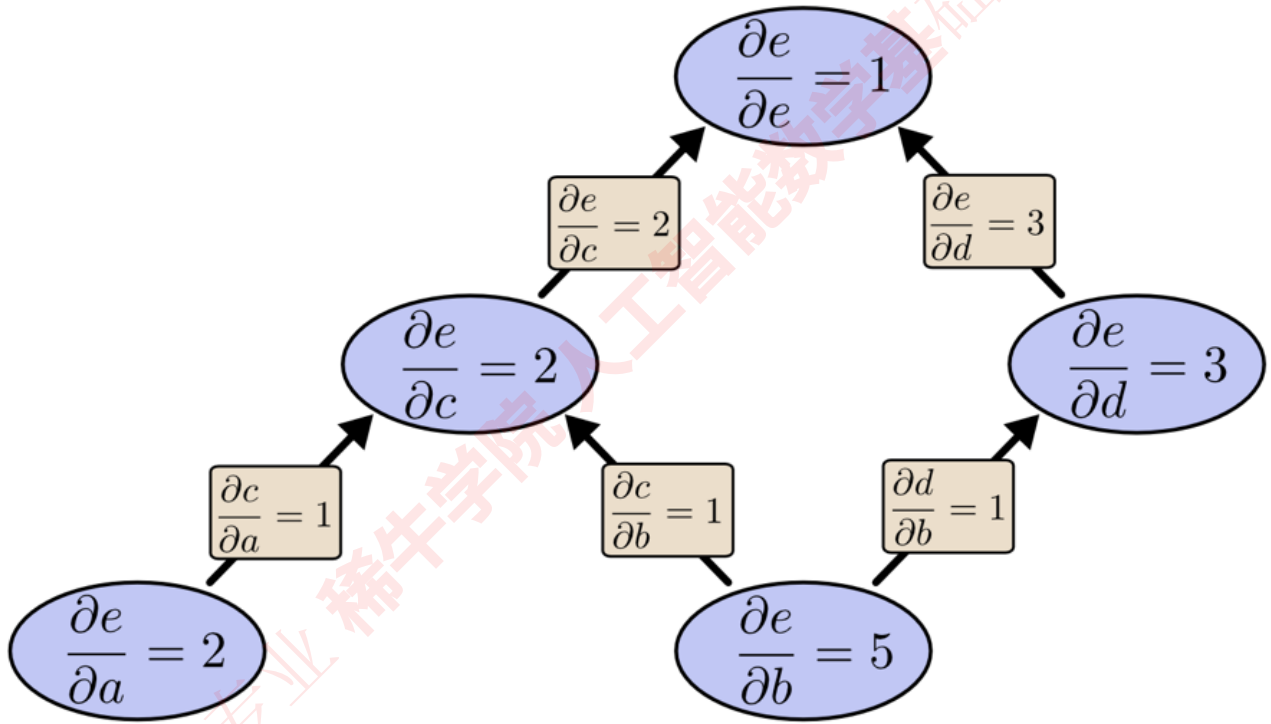


计算图

$$\frac{\partial e}{\partial a} = \frac{\partial e}{\partial c} \times \frac{\partial c}{\partial a}$$
$$\frac{\partial e}{\partial b} = \frac{\partial e}{\partial c} \times \frac{\partial c}{\partial b} + \frac{\partial e}{\partial d} \times \frac{\partial d}{\partial b}$$



BP



BP 案例演示

本章总结

本章参考资料

- Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

- Calculus on Computational Graphs: Backpropagation