图像处理与机器学习

Digital Image Processing and Machine Learning

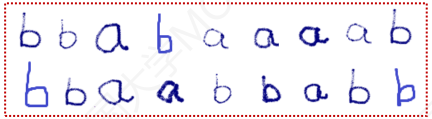
主讲人：黄琳琳

电子信息工程学院

人工神经网络

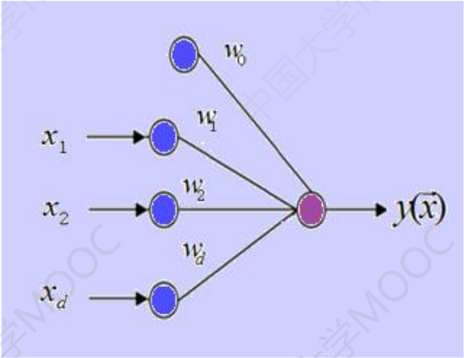
- 人工神经网络训练（Training）

  -- 训练样本集



- 人工神经网络测试（Testing）

  -- 测试样本集



分类精度如何？ $y(\vec{x}) = f(\vec{w}^T \vec{x})$

$$y(\vec{x}) = f(w_0 x_0 + w_1 x_1 + \ldots w_d x_d)$$

线性分类函数

判别函数

- 高斯密度下的判别函数

$$p(\vec{x}\mid\omega_i)=\frac{1}{(2\pi)^{d/2}\,|\Sigma_i|^{1/2}}\exp\left[-\frac{1}{2}(\vec{x}-\mu_i)\Sigma_i^{-1}(\vec{x}-\mu_i)\right]$$

$$\Sigma_k=\sigma^2 I \quad \Sigma_k=\Sigma$$

线性分类函数(LDF)

$$y_k(\vec{x})=\vec{w}_k{}^t\vec{x}+w_{k0}$$

$$\Sigma_k=arbitrary \quad y_k(\vec{x})=\vec{x}^t\vec{W}_k\vec{x}+\vec{w}_k{}^t\vec{x}+w_{k0}$$

非线性分类函数(QDF)

多层神经网络

多层神经网络

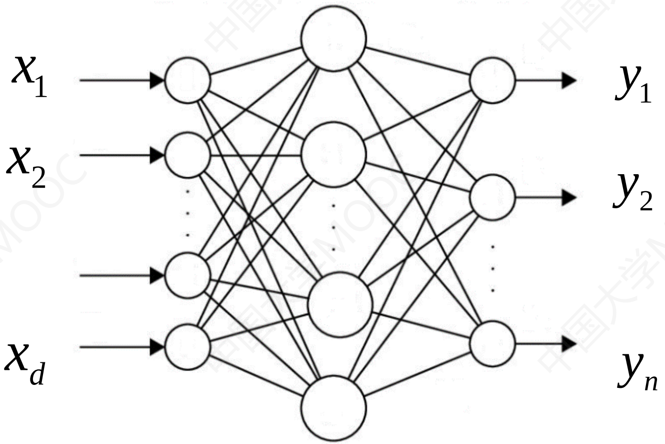
➢ 多层神经网络

名称	层数	网络结构	学习规则
多层感知机 (Back-Propagation， BP)	多层	前馈	误差修正律
径向基函数 (Radial Basis Function， RBF)	多层	前馈	误差修正律
自组织映射 (Selp-Organization Map， SOM)	多层	前馈	竞争律

多层神经网络

➢ 多层感知机

-- 正向计算输出、反向修正权值



反向传播网络

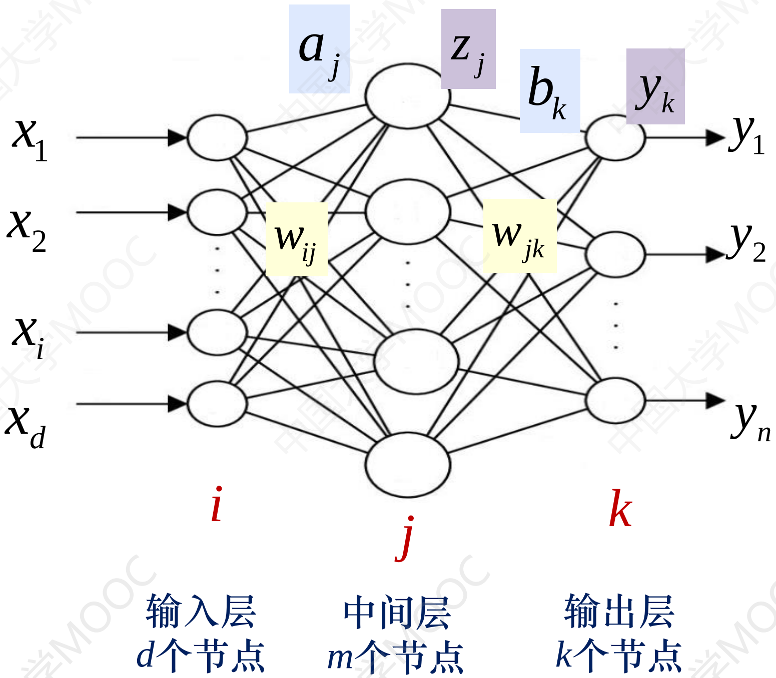
(Back-Propagation， BP)

多层神经网络

- ✓ 正向计算网络输出
  - -- 计算隐层节点的输出
  - -- 计算输出层节点的输出
- ✓ 反向权值修正
  - -- 从输出节点开始
  - -- 误差反向传播
  - -- 由总误差实施修正

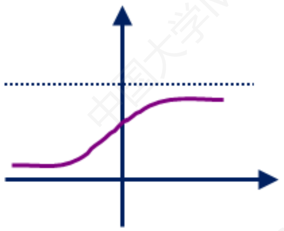
多层神经网络

➢ 正向计算输出



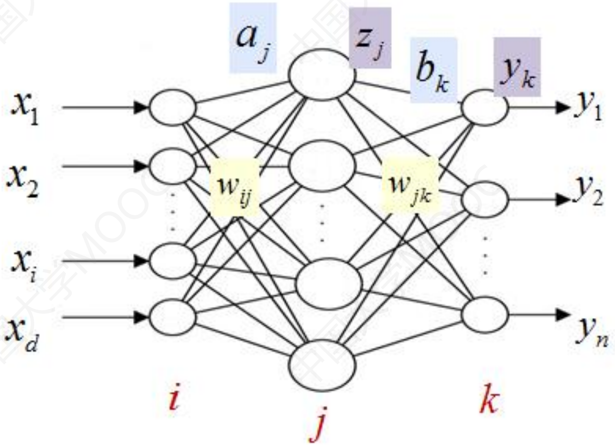
$$a_j = \sum_{i=1}^{d} w_{ij} x_i \qquad z_j = g(a_j)$$

$$b_k = \sum_{j=1}^{m} w_{jk} z_j \qquad y_k = g(b_k)$$



多层神经网络

- 反向调节权值 $a_j = \sum_{i=1}^{d} w_{ij} x_i \quad z_j = g(a_j) \quad b_k = \sum_{j=1}^{m} w_{jk} z_j \quad y_k = g(b_k)$



$$E = \frac{1}{2}\{\sum_{n=1}^{N}[y(x^n) - t^n]^2\} \qquad w(n+1) = w(n) - \eta\frac{\partial E^n}{\partial w}$$

$$\frac{\partial E}{\partial w_{jk}} = \boxed{\frac{\partial E}{\partial y_k}} \bullet \boxed{\frac{\partial y_k}{\partial w_{jk}}} \qquad \frac{\partial E}{\partial y_k} = (y_k - t)$$

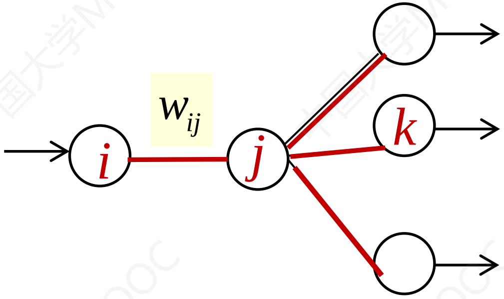
$$\frac{\partial y_k}{\partial w_{jk}} = \frac{\partial y_k}{\partial b_k} \cdot \frac{\partial b_k}{\partial w_{jk}} \qquad \frac{\partial y_k}{\partial b_k} = y_k(1 - y_k) \qquad \frac{\partial b_k}{\partial w_{jk}} = z_j$$

$$\frac{\partial E}{\partial w_{jk}} = (y_k - t) \cdot y_k(1 - y_k) \cdot z_j$$

多层神经网络

➤ 反向调节权值

$$a_j = \sum_{i=1}^{d} w_{ij} x_i \quad z_j = g(a_j) \quad b_k = \sum_{j=1}^{m} w_{jk} z_j \quad y_k = g(b_k)$$



$$E = \frac{1}{2}\{\sum_{n=1}^{N}[y(x^n) - t^n]^2\} \qquad w(n+1) = w(n) - \eta \frac{\partial E^n}{\partial w}$$

$$\frac{\partial y_k}{\partial w_{ij}} = \frac{\partial y_k}{\partial b_k} \cdot \frac{\partial b_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ij}} \qquad \frac{\partial y_k}{\partial b_k} = y_k(1 - y_k)$$

$$\frac{\partial E}{\partial w_{ij}} = \sum_{k=1}^{n} \left(\frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial w_{ij}}\right)$$

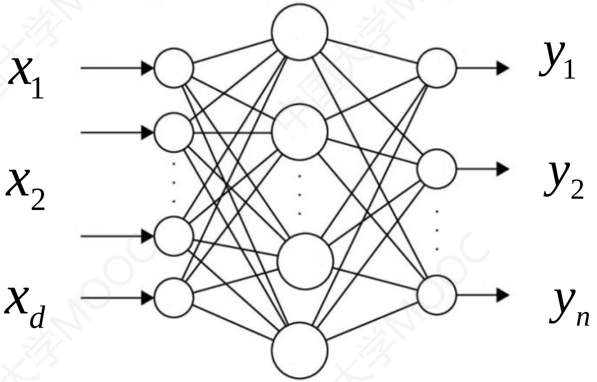
$$\frac{\partial b_k}{\partial z_j} = w_{jk} \qquad \frac{\partial z_j}{\partial a_j} = z_j \cdot (1 - z_j) \qquad \frac{\partial a_j}{\partial w_{ij}} = x_i$$

$$\frac{\partial E}{\partial y_k} = (y_k - t)$$

$$\frac{\partial E}{\partial w_{ij}} = \sum_{k}^{n} \left[(y_k - t) \cdot y_k \cdot (1 - y_k) \cdot z_j \cdot (1 - z_j) \cdot w_{jk} \cdot x_i\right]$$

神经网络训练

- 多层感知机



$y(\vec{x})$

非线性函数

误差反向传播
(Back-Propagation， BP)

1986年，多层感知机
引发了神经网络研究的热潮…

谢　谢

本课程所引用的一些素材为主讲老师多年的教学积累，来源于多种媒体及同事和同行的交流，难以一一注明出处，特此说明并表示感谢！