Simple Introduction of Back Propagation Neuron Network

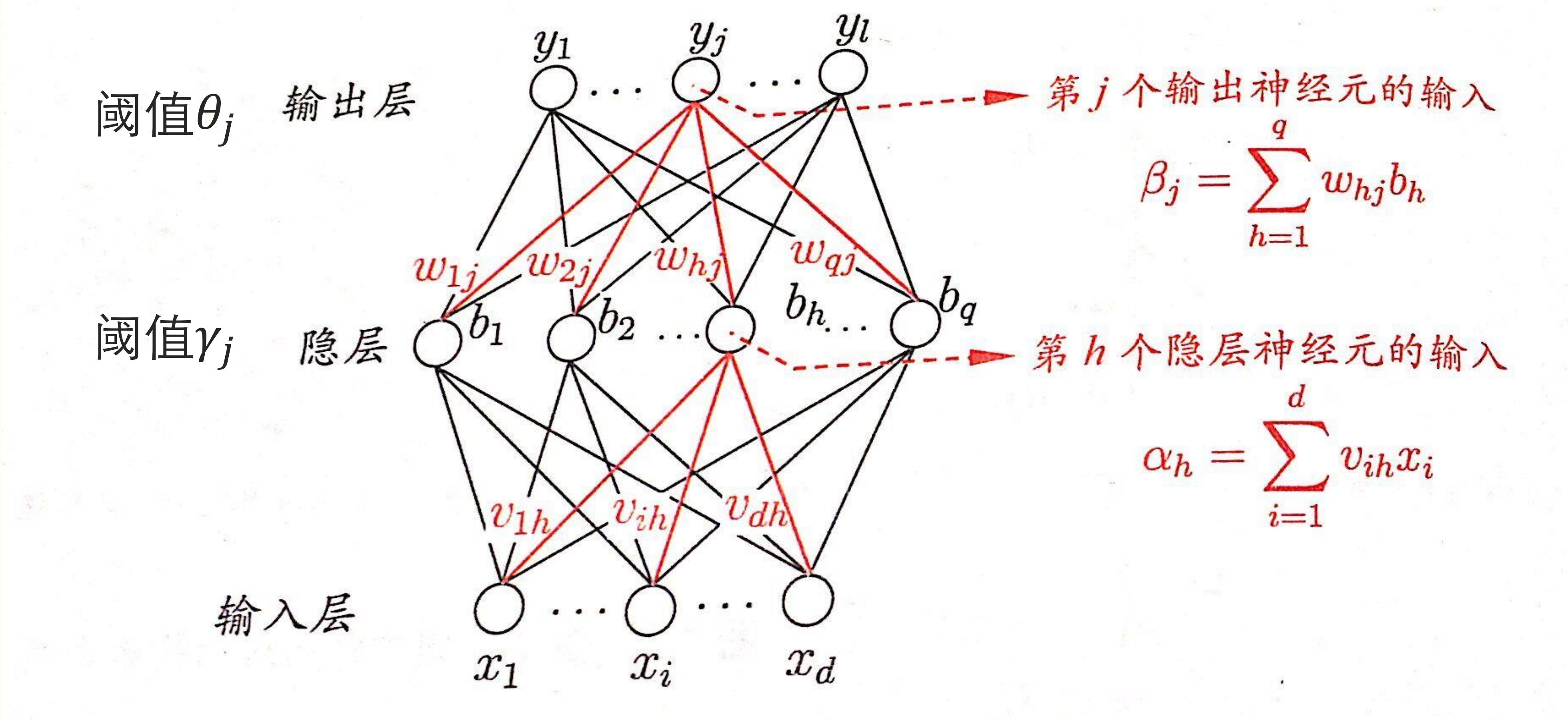
背景介绍

Back propagation(BP)神经网络是1986年由Rumelhart和McClelland为首的科学家提出的，是一种按照误差逆向传播算法训练的多层前馈神经网络。

20世纪80年代中期，BP算法被提出，解决了多层神经网络隐层连接权的学习问题，并在数学上给出了完整的推导。

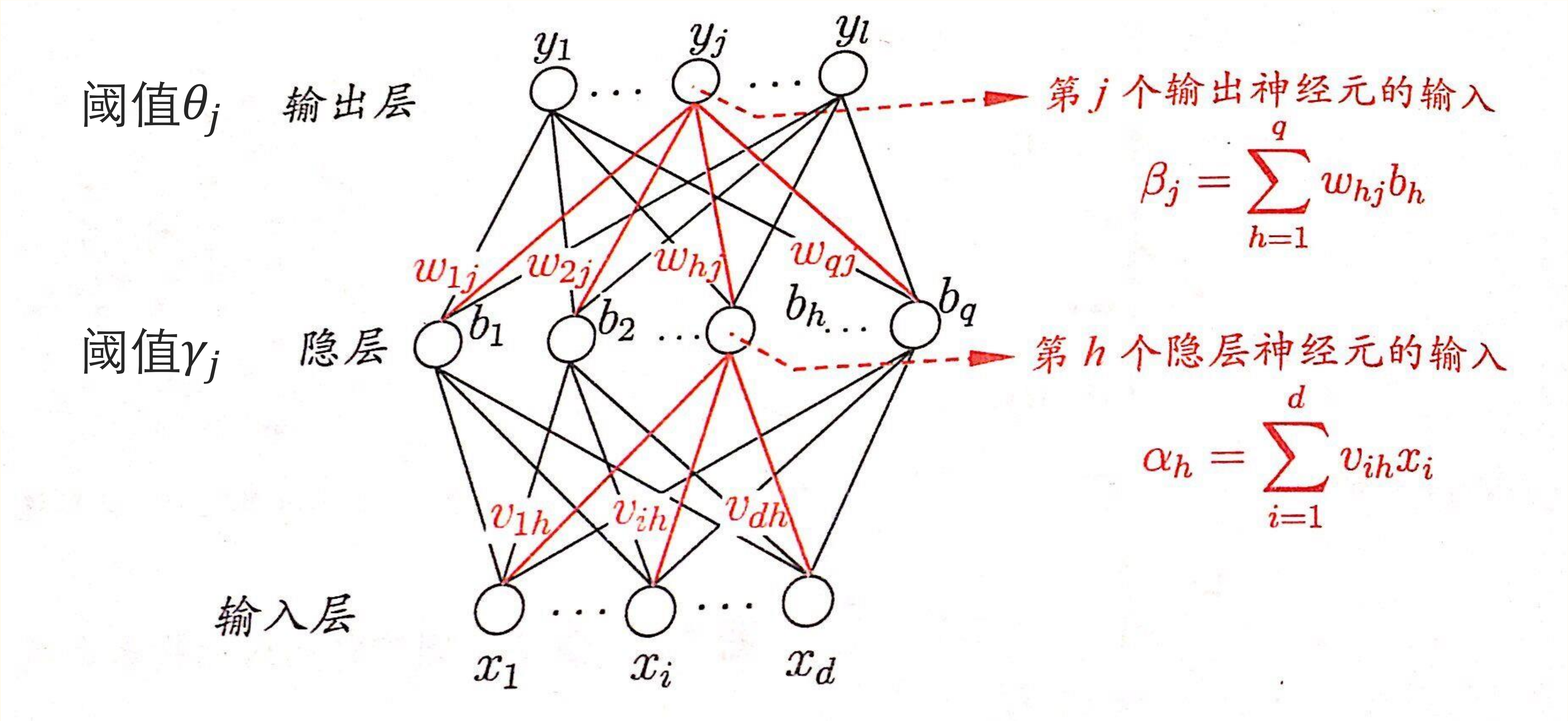
一个包含足够多神经元的隐层的BP神经网络能够以任意精度逼近任意复杂度的连续函数

BP算法



假设激励函数都使用Sigmoid函数，对训练例$(\boldsymbol{x}_k, \boldsymbol{y}_k)$，假定神经网络的输出为$\hat{\boldsymbol{y}}_k = (\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots, \hat{y}_l)$，即$\hat{y}_j = f(\beta_j - \theta_j)$

则均方误差 $\boldsymbol{E}_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j - y_j)^2$

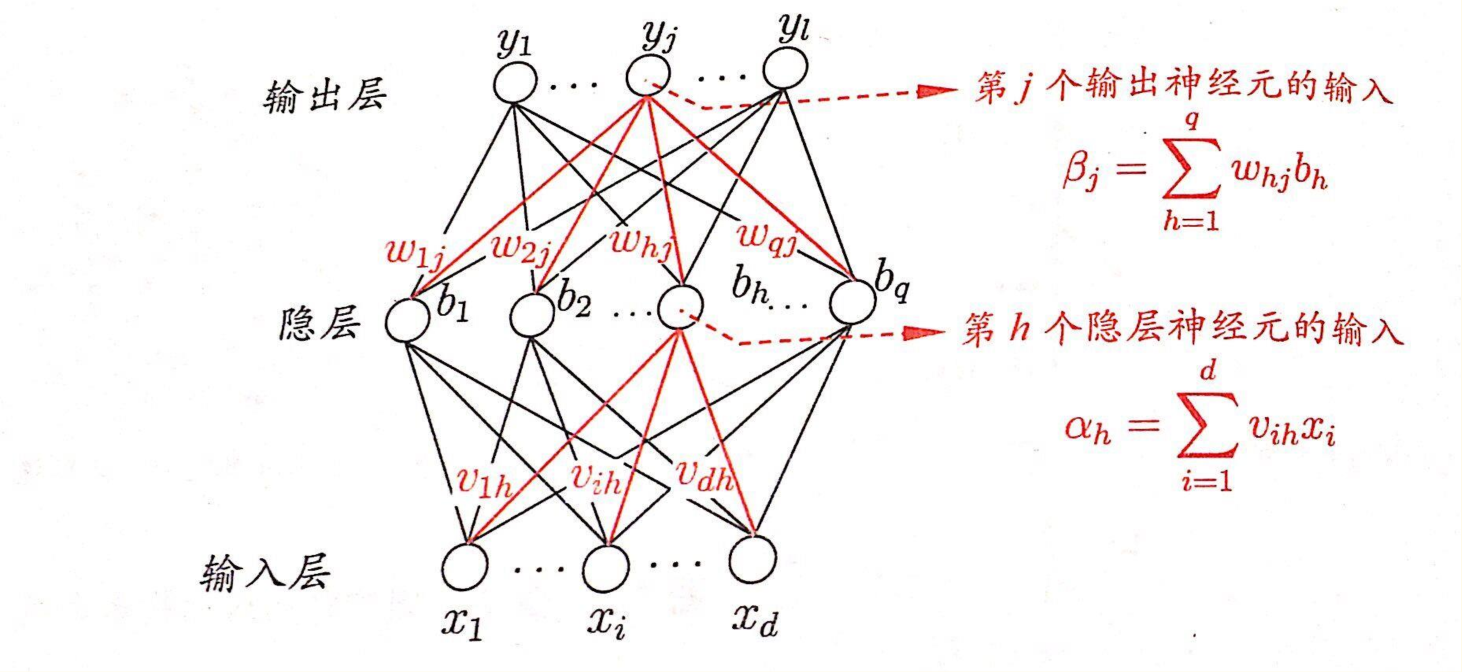


BP算法基于梯度下降策略，以目标的负梯度方向对参数进行调整，设η为学习率，则对于参数w的更新公式为

$$w_{hj}=w_{hj}+\Delta w_{hj}$$

$$\Delta w_{hj}=-\eta\frac{\partial E_k}{\partial w_{hj}}$$

BP算法



$$\boldsymbol{E}_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j - y_j)^2$$

$$\hat{y}_j = f(\beta_j - \theta_j)$$

$$f'(x) = f(x)(1 - f(x))$$

$$\frac{\partial \boldsymbol{E}_k}{\partial \boldsymbol{w}_{hj}} = \frac{\partial \boldsymbol{E}_k}{\partial \hat{\boldsymbol{y}}_j^k} \cdot \frac{\partial \hat{\boldsymbol{y}}_j^k}{\partial \boldsymbol{\beta}_j} \cdot \frac{\partial \boldsymbol{\beta}_j}{\partial \boldsymbol{w}_{hj}}$$

BP算法

$$\boldsymbol{E}_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j - y_j)^2$$

$$\hat{y}_j = f(\beta_j - \theta_j)$$

$$f'(x) = f(x)(1 - f(x))$$

$$\frac{\partial \boldsymbol{E}_k}{\partial \boldsymbol{w}_{hj}} = \frac{\partial \boldsymbol{E}_k}{\partial \hat{\boldsymbol{y}}_j^k} \cdot \frac{\partial \hat{\boldsymbol{y}}_j^k}{\partial \boldsymbol{\beta}_j} \cdot \frac{\partial \boldsymbol{\beta}_j}{\partial \boldsymbol{w}_{hj}}$$

其中

$$\frac{\partial \boldsymbol{E}_k}{\partial \hat{\boldsymbol{y}}_j^k} \cdot \frac{\partial \hat{\boldsymbol{y}}_j^k}{\partial \boldsymbol{\beta}_j} = \left(\hat{\boldsymbol{y}}_j^k - \boldsymbol{y}_j^k\right) f'\left(\beta_j - \theta_j\right)$$

$$= \hat{\boldsymbol{y}}_j^k(\hat{\boldsymbol{y}}_j^k - \mathbf{1})(\boldsymbol{y}_j^k - \hat{\boldsymbol{y}}_j^k)$$

令上式等于$\boldsymbol{g}_j$

BP算法

$$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$$

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

其中

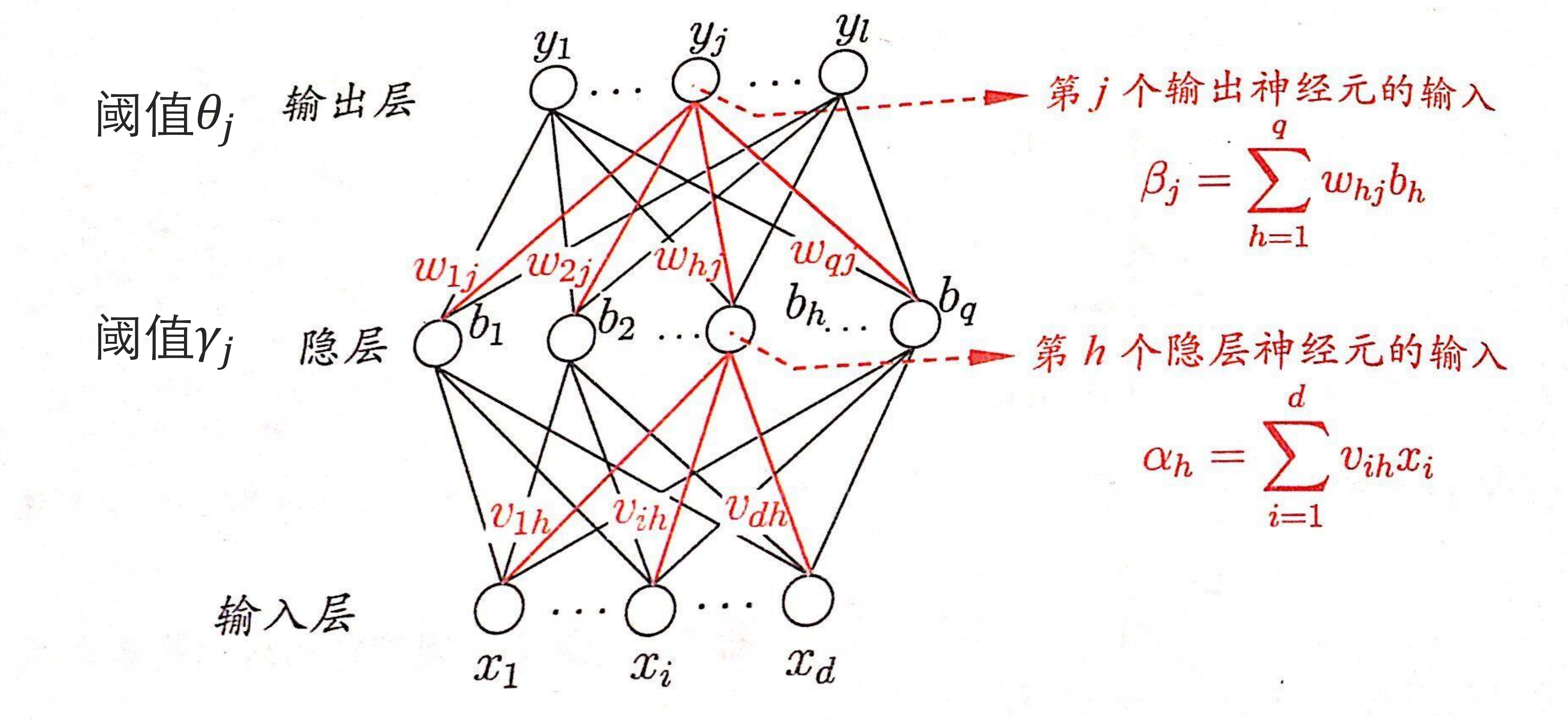
$$\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} = \left(\hat{y}_j^k - y_j^k\right) f'\left(\beta_j - \theta_j\right)$$
$$= \hat{y}_j^k(\hat{y}_j^k - 1)(y_j^k - \hat{y}_j^k)$$

令上式等于g_j

可得

$$\Delta w_{hj} = -\eta g_j b_h$$

BP算法



$$\hat{y}_j = f(\beta_j - \theta_j)$$

$$b_h = f(a_h - \gamma_j)$$

Thank you