

5.1 神经网络

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

阈值

$$\text{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

5.2 感知机与多层网络

0 可看作固定输入为 1.0 的哑结点. dummy node.

$$w_i \leftarrow w_i + \Delta w_i$$

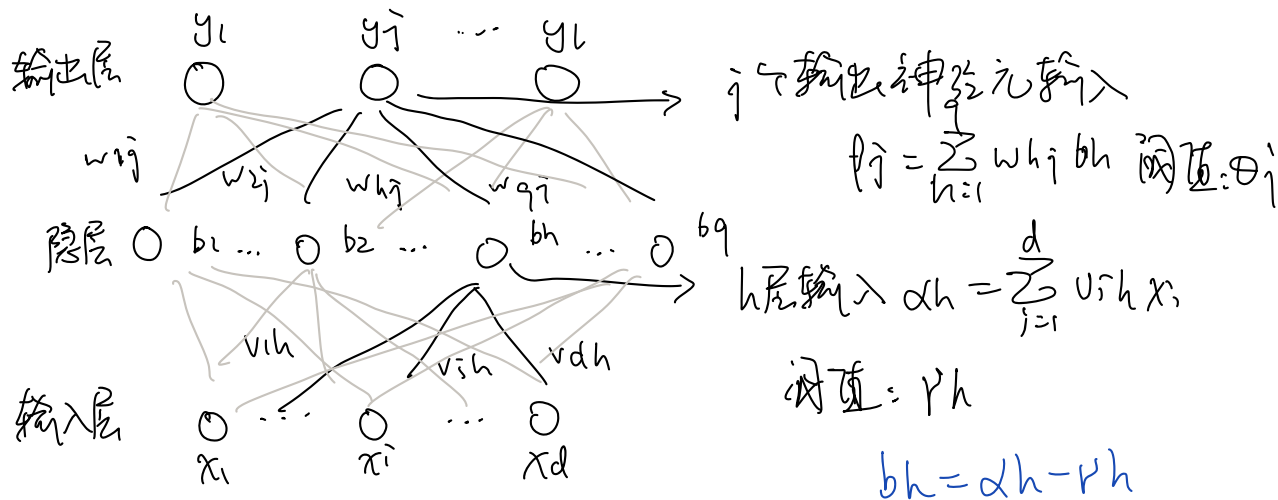
$$\Delta w_i = \eta (y - \hat{y}) x_i$$

↑
学习率

5.3 误差逆传播算法. BP.

$$D = \{(x_1, y_1), \dots, (x_m, y_m)\}. \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}^l$$

d 特征维数 l 输出维数



对训练集 (x_k, y_k) , 假设输入为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$ 即 $\hat{y}_i^k = f(\beta_i - \theta_i)$

均为误差. $E_k = \frac{1}{2} \sum_{i=1}^l (\hat{y}_i^k - y_i^k)^2$

BP 基于梯度下降 $\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}}$

$$\therefore \frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \quad \underbrace{\hspace{1cm}} \rightarrow b$$

$$g_j = - \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}$$

$$= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial [f(\beta_j - \theta_j)]}{\partial \theta_j}$$

$$= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot f'(\beta_j - \theta_j) = \frac{\partial [\frac{1}{2} \sum_{k=1}^L (\hat{y}_j^k - y_j^k)^2]}{\partial \hat{y}_j^k} \cdot f'(\beta_j - \theta_j)$$

$$= -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \quad \text{sigmoid: } f(x) = f(x)(1-f(x))$$

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

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

$$\text{类似地: } \Delta \theta_j = -\eta g_j$$

$$\Delta v_{ih} = \eta e_h x_i$$

$$\Delta \varphi_h = -\eta e_h$$

$$e_h = - \frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h}$$

$$= b_h(1-b_h) \sum_{j=1}^L w_{hj} g_j$$

$$\Delta v_{ih} = -\eta \frac{\partial E_k}{\partial v_{ih}}$$

$$\frac{\partial E_k}{\partial v_{ih}} = \sum_{j=1}^L \underbrace{\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}}_{(-g_j)} \cdot \underbrace{\frac{\partial \beta_j}{\partial b_h}}_{w_{hj}} \cdot \underbrace{\frac{\partial b_h}{\partial \alpha_h}}_{f'(\alpha_h - \varphi_h)} \cdot \underbrace{\frac{\partial \alpha_h}{\partial v_{ih}}}_{x_i}$$

$$b_h(1-b_h)$$

$$\Delta \varphi_h = -\eta e_h = -\eta \frac{\partial E_k}{\partial \varphi_h}$$

$$\frac{\partial E_k}{\partial \varphi_h} = \sum_{j=1}^L \underbrace{\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}}_{(-g_j)} \cdot \underbrace{\frac{\partial \beta_j}{\partial b_h}}_{w_{hj}} \cdot \underbrace{\frac{\partial b_h}{\partial \varphi_h}}_{f'(\alpha_h - \varphi_h)}$$

$$\text{计算: } \hat{y}_j^k \rightarrow g_j \rightarrow e_h \rightarrow w_{hj}, v_{ih}, \theta_j, \varphi_h$$

$$\text{累积误差: } E = \frac{1}{n} \sum_{k=1}^n E_k$$

缺点: 易过拟合.

解决: ① 早停. 若 $E(\text{in}) \downarrow$, $E(\text{test}) \uparrow$, 则停止训练. 返回 $\arg \min_{(w_{ij}, v_{ij}, \theta_{ij}, \rho_{ij})}$

② 正则化.

$$\bar{E} = \lambda \frac{1}{n} \sum_{k=1}^n \bar{E}_k + \underbrace{(1-\lambda) \sum_i w_i^2}$$

5.4 全局最小与局部极小.

如何接近全局最小: ① 多步搜索.

② 模拟退火. 一定概率接受更差结果.

③ 随机梯度下降.

5.5 其他神经网络.

5.5.1 RBF神经网络. 径向基函数.

单隐层, 隐层函数为径向基. 输出层是隐层线性组合.

输入 d 维向量 x , $\varphi(x) = \sum_{i=1}^q w_i \rho(x, c_i)$.

q : 隐层神经元个数.

c_i : i 个隐层神经元中心.

w_i : i 个隐层神经元权重.

$\rho(x, c_i)$: 径向基函数, 一般为 x 到 c_i 的欧氏距离.

$\rho(x, c_i) = e^{-\beta \|x - c_i\|^2}$ 高斯径向基函数.

步骤: ① 确定 c_i 通过随机采样或聚类.

② BP 确定 w_i, β_i .

5.5.2 ART网络, 竞争型学习.

无监督, 每时仅有一个神经元被激活.

可进行增量学习. 与阈值相关.

(可监督: ARTMAP).

5.5.3 SOM网络. 自组织映射.

竞争学习无监督. 保持高维数据拓扑结构.

(空间相似 \rightarrow 邻近神经元)

5.5.4 级联相关网络. 结构自适应性

逐层增加隐层层次结构.

数据小易过拟合.

5.5.5 Elman网络. 递归神经网络.

可有环形结构. 处理时间等状态变化.

Sigmoid + BP.

5.5.6 Boltzman 递归神经网络

基于能量模型.

bool 神经元. $E(s) = - \sum_{i=1}^n \sum_{j=1}^n w_{ij} s_i s_j - \sum_{i=1}^n \theta_i s_i$.
 $s \in \{0, 1\}^n$. energy \rightarrow 阈值.
连接权. 神经元状态.

标准 Boltzman 网. 全连接图. 复杂度太高.

受限 Boltzman 网 (RBM) 同一层不存在连接

使用“对比散度”训练. (CD 算法).

有 d 个显层神经元与 q 个隐层神经元.

v : 显层状态向量. h : 隐层状态向量.

$$P(v|h) = \prod_{i=1}^d P(v_i|h).$$

$$P(h|v) = \prod_{j=1}^q P(h_j|v).$$

$$\begin{aligned} \text{训练数据 } v &\xrightarrow{i=1} \text{概率分布变换得 } h \xrightarrow{} v' \xrightarrow{} h' \\ &\rightarrow \Delta W = \eta (v h^T - v' h'^T) \end{aligned}$$

5.6. 深度学习

深层神经网络. 多层 (8, 9 <).

无监督逐层训练. 深度神经网络.

① 预训练 + 微调: DBN = n^* RBM, 各层训练完成再对全网络 BP.

② 权共享: CNN: 每个卷积层多个特征映射. 每个特征映射由多个神经元构成平面. 网络中每组神经元平权.

ReLU: 修正线性整流函数. $f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$

前若干层是特征表示, 最后一层分类.