

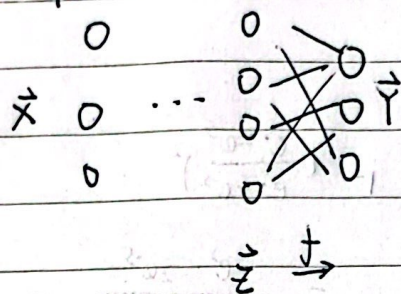
No.

## 模式识别第三周作业

Date.

何东阳

4.1 证明: 假设对于一个网络F

映射关系是  $\hat{y} = F(x)$ 其中  $\hat{y}$  是  $m$  维向量  $(y_1, y_2, \dots, y_m)$ ,  $x$  是  $n$  维向量  $(x_1, x_2, \dots, x_n)$ 假设网络最后一层的映射关系是  $\hat{y} = f(z)$ 其中  $z$  是倒数第二层的输出向量, 示意图如下:

假如激活函数是线性函数, 相当于每层都是线性函数

则  $\hat{y} = \bar{w}_n \cdot z_n$ ,  $\bar{w}$  是最后一层权重, 同理

$$z_n = w_{n-1} \cdot z_{n-1}, \dots \Rightarrow \hat{y} = w_n \cdot w_{n-1} \dots w_1 \cdot x$$

其中  $w_1, \dots, w_n$  是每层权重, 因此  $\hat{y}$  和  $x$  是线性关系

如果多层感知机节点的激活函数采用线性函数, 网络无法实现非线性映射

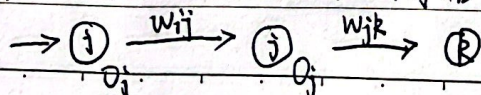
4.2 (1): 证明: (i):  $\theta(s) = \frac{1}{1+e^{-s}} = \frac{e^s}{(1+e^{-s})e^s} = \frac{e^s}{e^s+1}$

(ii):  $\theta(1-s) = \frac{1}{1+e^{-(1-s)}} = 1 - \frac{e^s}{1+e^s} = 1 - \theta(s)$  (由(i)可知)

(iii):  $\theta'(s) = \frac{(1-e^s)}{(1+e^{-s})^2} = \frac{e^{-s}}{(1+e^{-s})^2} = \left(1 - \frac{1}{1+e^{-s}}\right) \left(\frac{1}{1+e^{-s}}\right)$   
 $= \theta(s)(1-\theta(s))$

(2):  $f(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} = \frac{e^s(e^s - e^{-s})}{e^s(e^s + e^{-s})} = \frac{e^{2s} - 1}{e^{2s} + 1} = \left(\frac{e^{2s}}{e^{2s} + 1} - \frac{1}{e^{2s} + 1}\right) = \theta(2s) - 1$

$f'(s) = 2\theta'(2s) = 4 \frac{\partial \theta(s)}{\partial s} \bigg|_{2s} = 4 \frac{e^s}{(1+e^{-s})^2} = 1 - \tanh(s)^2$

(3): 使用  $\tanh()$  后: 使用以下神经网络模型

扫描全能王 创建



其中对于节点  $j$ ,  $O_j = f(\text{net}_j) = f(\sum_i w_{ij} O_i)$

对于输出  $k$ ,  $O_k = y_j$ , 该期望输出为  $y_j$

使用 MSE 评估有:  $E = \frac{1}{2} \sum_j (y_j - y_j)^2$

梯度为  $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ij}} = \delta_j O_j$ , 其中  $\delta_j = \frac{\partial E}{\partial \text{net}_j}$

又  $\delta_j = \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial \text{net}_j} = -(y_j - y_j) f'(\text{net}_j)$ , 对于输出节点,

$\delta_j = \sum_k \frac{\partial E}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial O_j} \cdot \frac{\partial O_j}{\partial \text{net}_j} = \sum_k \delta_k w_{jk} f'(\text{net}_j)$ , 对于隐层节点,

对于  $f(x) = \tanh(x)$  其中  $f'(\text{net}_j) = 1 - \tanh^2(\text{net}_j)$ , 代入得

$\delta_j = \begin{cases} -(y_j - y_j)(1 - \tanh^2(\text{net}_j)) & \text{输出节点} \\ \sum_k \delta_k w_{jk} f'(1 - \tanh^2(\text{net}_j)) & \text{隐层节点} \end{cases} \Rightarrow \text{更新公式: } w_{ij}(t+1) = w_{ij}(t) - \eta \delta_j(t) O_i(t)$

- 为何多层感知不用 tanh: 因为 tanh 是饱和激活函数,  $x$  不断增大或减小时,  $y$  会逐渐趋于固定, 接近 0 或 1, 导致梯度很小, 学习效果不好。

