第三章 前馈神经网络

3.4 误差反传（BP）算法

回顾

- 3.1 单层感知器
  - 模型：单计算节点感知器实际上就是一个 M-P 神经元模型
  - 功能：解决线性可分问题
  - 局限性：不能解决线性不可分问题
  - 学习算法：有导师学习
- 3.2 多层感知器
  - 模型：有隐层的多层前馈网络
  - 功能：能够求解非线性问题
  - 局限性：隐层神经元的学习规则尚无所知

3.3BP 算法及改进 - 主要内容

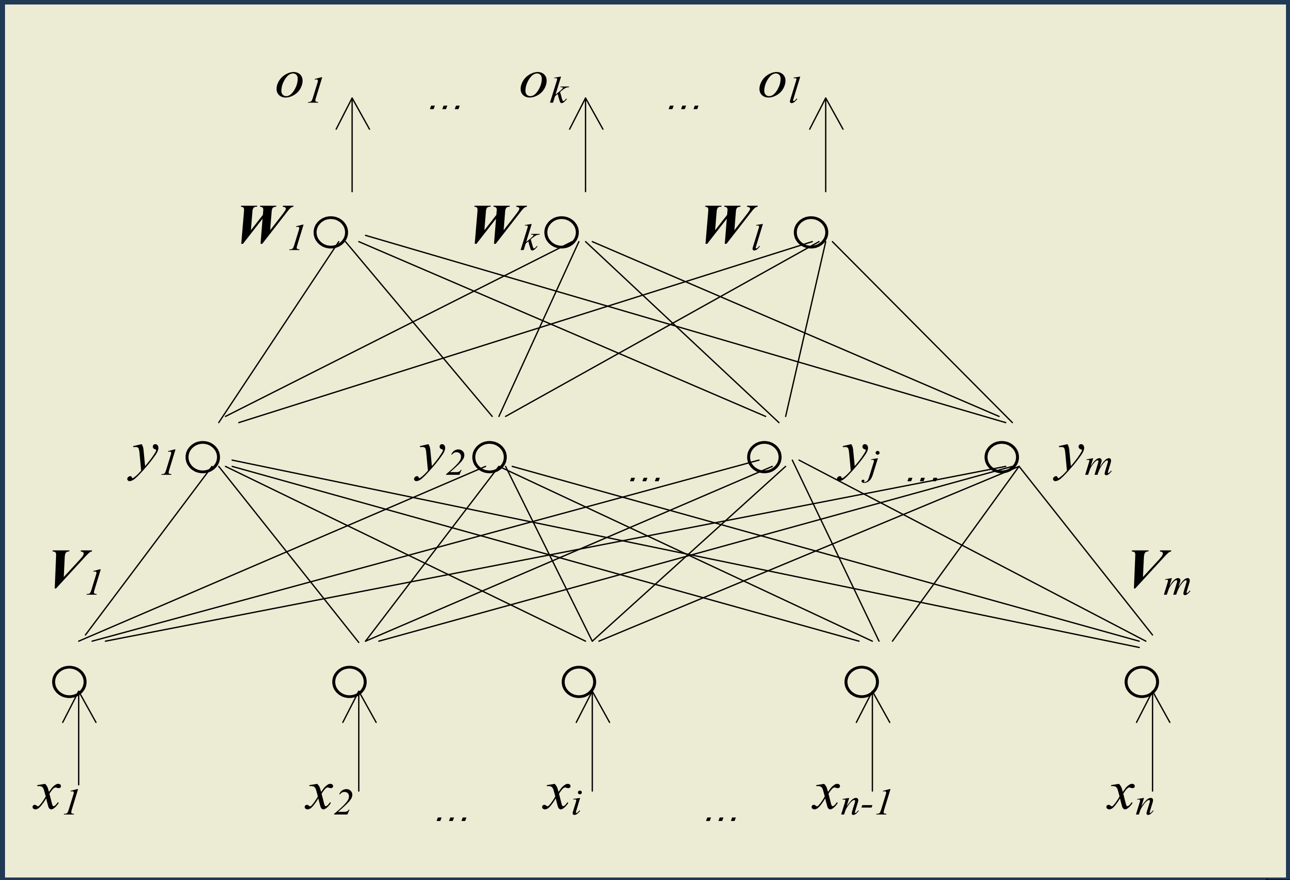
- 引言
- 基于 BP 算法的多层前馈网络模型
- BP 算法的实现
  - 基本思想
  - 推导过程
  - 程序实现
- BP 学习算法的功能
- BP 学习算法的局限性
- BP 学习算法的改进

引言－－BP算法的提出

- 提高网络性能（如分类能力）的有效途径
  - 包含隐层的多层前馈网络
    - **长期以来没有提出*解决权值调整问题的有效算法*。**
  - 非线性连续转移函数
- BP (Error Back Propagation ，BP) 算法
  - 1986年，Rumelhart 和 McCelland 领导的科学家小组《Parallel Distributed Processing》一书
  - 应用对象：多层前馈网络
  - 具有非线性连续转移函数

3.4 误差反传（ BP ）算法

3.4.1 基于 BP 算法的多层前馈网络模型



□模型的数学表达

输入向量： $X=(x_1,x_2,\ldots,x_i,\ldots,x_n)^T$

隐层输出向量： $Y=(y_1,y_2,\ldots,y_j,\ldots,y_m)^T$

输出层输出向量： $O=(o_1,o_2,\ldots,o_k,\ldots,o_l)^T$

期望输出向量： $d=(d_1, d_2,\ldots,d_k,\ldots,d_l)^T$

输入层到隐层之间的权值矩阵： $V=(V_1,V_2,\ldots,V_j,\ldots,V_m)$

隐层到输出层之间的权值矩阵： $W=(W_1,W_2,\ldots,W_k,\ldots,W_l)$

各个变量之间如何建立联系，来描述整个网络？

3.4.1 基于 BP 算法的多层前馈网络模型

对于输出层：

$$o_k = f(net_k) \quad k=1,2,\ldots,l \qquad (3.4.1)$$

$$net_k = \sum_{j=0}^{m} w_{jk} y_j \quad k=1,2,\ldots,l \qquad (3.4.2)$$

对于隐层：

$$y_j = f(net_j) \quad j=1,2,\ldots,m \qquad (3.4.3)$$

$$net_j = \sum_{i=0}^{n} v_{ij} x_i \quad j=1,2,\ldots,m \qquad (3.4.4)$$

3.4.1 基于 BP 算法的多层前馈网络模型

单极性 Sigmoid 函数：

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.4.5)$$

双极性 Sigmoid 函数：

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

3.4.2 BP 学习算法

误差反传（BP）算法

一、网络误差与权值调整

输出误差 E 定义：$E=\frac{1}{2}(\boldsymbol{d}-\boldsymbol{O})^2=\frac{1}{2}\sum_{k=1}^{l}(d_k-o_k)^2$ **(3.4.6)**

将以上误差定义式展开至隐层：

$$E=\frac{1}{2}\sum_{k=1}^{l}[d_k-f(net_k)]^2=\frac{1}{2}\sum_{k=1}^{l}[d_k-f(\sum_{j=0}^{m}w_{jk}y_j)]^2$$

(3.4.7)

3.4.2 BP 学习算法

误差反传（BP）算法

一、网络误差与权值调整

进一步展开至输入层：

$$E=\frac{1}{2}\sum_{k=1}^{l}\{d_k-f[\sum_{j=0}^{m}w_{jk}f(net_j)]\}^2$$

$$=\frac{1}{2}\sum_{k=1}^{l}\{d_k-f[\sum_{j=0}^{m}w_{jk}f(\sum_{i=0}^{n}v_{ij}x_i)]\}^2 \quad (3.4.8)$$

3.4.2 BP 学习算法

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$$

$j=0,1,2,\ldots,m;\ \ k=1,2,\ldots,l$ (3.4.9a)

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}}$$

$i=0,1,2,\ldots,n;\ \ j=1,2,\ldots,m$ (3.4.9b)

式中负号表示梯度下降，常数 $\eta\in(0,1)$ 表示比例系数。

在全部推导过程中，对输出层有 $j=0,1,2,\ldots,m;\ \ k=1,2,\ldots,l$

对隐层有 $i=0,1,2,\ldots,n;\ \ j=1,2,\ldots,m$

3.4.2 BP 学习算法

二、 BP 算法推导

对于输出层，式 (3.15a) 可写为

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} \quad (3.4.10a)$$

对隐层，式 (3.15b) 可写为

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}} = -\eta \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial v_{ij}} \quad (3.4.10b)$$

对输出层和隐层各定义一个误差信号，令

$$\delta_k^o = -\frac{\partial E}{\partial net_k} \quad (3.4.11a) \qquad \delta_j^y = -\frac{\partial E}{\partial net_j} \quad (3.4.11b)$$

综合应用式 (3.4.2) 和 (3.4.11a) ，可将式 (3.4.10a) 的权值调整式改写为

$$\Delta w_{jk} = \eta \delta_k^o y_j \qquad (3.4.12a)$$

综合应用式 (3.13) 和 (3.21b) ，可将式 (3.17b) 的权值调整式改写为

$$\Delta v_{ij} = \eta \delta_j^y x_i \qquad (3.4.12b)$$

可以看出，只要计算出式 (3.4.12) 中的误差信号 δ^o 和 δ^y，权值调整量的计算推导即可完成。下面继续推导如何求误差信号 δ^o 和 δ^y 。

对于输出层，δ^{o} 可展开为

$$\delta_k^o = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial net_k} = -\frac{\partial E}{\partial o_k} f'(net_k) \qquad (3.4.13a)$$

对于隐层，δ^{y} 可展开为

$$\delta_j^y = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial net_j} = -\frac{\partial E}{\partial y_j} f'(net_j) \qquad (3.4.13b)$$

下面求式 (3.23) 中网络误差对各层输出的偏导。

对于输出层，利用式 (3.4.6) ：

$$E=\frac{1}{2}\sum_{k=1}^{l}(d_k-o_k)^2$$

可得：

$$\frac{\partial E}{\partial o_k}=-(d_k-o_k) \qquad (3.4.14a)$$

对于隐层，利用式 (3.4.7) ：

$$E=\frac{1}{2}\sum_{k=1}^{l}[d_k-f(\sum_{j=0}^{m}w_{jk}y_j)]^2$$

可得：

$$\frac{\partial E}{\partial y_j}=-\sum_{k=1}^{l}(d_k-o_k)f'(net_k)w_{jk} \qquad (3.4.14b)$$

将以上结果代入式 (3.4.13) ，并应用式 (3.4.15) ：

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

误差反传（BP）算法

得到：

$$\delta_k^o=(d_k-o_k)o_k(1-o_k) \qquad (3.4.15a)$$

$$\delta_j^y=[\sum_{k=1}^{l}(d_k-o_k)f'(net_k)w_{jk}]f'(net_j)$$

$$=(\sum_{k=1}^{l}\delta_k^o w_{jk})y_j(1-y_j) \qquad (3.4.15b)$$

至此两个误差信号的推导已完成。

将式 (3.4.15) 代回到式 (3.4.12) ，得到三层前馈网的 BP 学习算法权值调整计算公式为：

$$\begin{cases} \Delta w_{jk} = \eta\delta_k^o y_j = \eta(d_k - o_k)o_k(1-o_k)y_j & \textbf{(3.4.16a)} \\ \Delta v_{ij} = \eta\delta_j^y x_i = \eta\left(\sum_{k=1}^{l}\delta_k^o w_{jk}\right)y_j(1-y_j)x_i & \textbf{(3.4.16b)} \end{cases}$$

误差反传（BP）算法

$$o_k = f(net_k)$$

$$y_j = f(net_j)$$

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

$$net_k = \sum_{j=0}^{m} w_{jk} y_j$$

$$net_j = \sum_{i=0}^{n} v_{ij} x_i$$

$$E = \frac{1}{2}\sum_{k=1}^{l}(d_k - o_k)^2 = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(net_k)]^2 = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(\sum_{j=0}^{m} w_{jk} y_j)]^2$$

$$= \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(net_j)]\}^2 = \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(\sum_{i=0}^{n} v_{ij} x_i)]\}^2$$

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}}$$

$$\delta_k^o = -\frac{\partial E}{\partial net_k}$$

$$\Delta w_{jk} = \eta \delta_k^o y_j$$

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}} = -\eta \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial v_{ij}}$$

$$\delta_j^y = -\frac{\partial E}{\partial net_j}$$

$$\Delta v_{ij} = \eta \delta_j^y x_i$$