第12卷 第1期
1993年2月

红外与毫米波学报
J. Infrared Millim. Waves

Vol. 12, No. 1
February, 1993

多层神经网络逆向传播算法简明解析推导和一种变型MRⅢ计算技术*

陈继述

(宁波大学物理系，浙江，宁波，315211)

摘要：给出了多层神经网络逆向传播法整套计算公式的简明解析推导，提出了一种变型的MRⅢ计算技术.

关键词：多层前馈神经网络，逆向传播算法，MRⅢ计算技术.

引言

Rumelhart 1986年发表的逆向传播法[1]无疑是迄今最受重视的一种神经网络学习算法. 但由于神经网络的变量太多，变量的记号易互相混淆，所以不容易清晰严谨地推导出全套计算公式，影响到这一重要算法的完整阐述和提高普及，甚至会使人产生逆向传播法没有一套完整计算方程式的糊涂观念. 本文第一节对多层神经网络设计了一套变量的简明记号，并用这套记号简捷地推导出逆向传播法的全套计算公式. 这对逆向传播法的改进和推广都是有益的.

Widrow 于 1990 年[2]、Holler 于 1989 年[3]为了克服在数字计算机上实现逆向传播法的障碍，发展出一套线性适应性多元系统第三规则(简称MRⅢ)的计算技术，后来曾据此研制出若干种有实际用途的神经网络商品[4]. 本文第二节提出一种原理上类似MRⅢ、但更为直截了当的规则，即变型MRⅢ，目的也是为了便于用数字计算机实现逆向传播法. 变型MRⅢ与原型MRⅢ的优劣比较需要今后通过计算机实验才能定论.

1 典型逆向传播算法的简明解析推导

考虑多层前馈神经网络模型，共有 l 层，层的编号记为 (m)，$(m=1,2,\cdots,l)$. 其中的 (1) 为输入层的编号，(l) 为输出层的编号，其余 (m) 为中间层的编号. 第 m 层中的神经元总数记为 N_m. 第 m 层中神经元编号记为 $1(m), 2(m), \cdots, i(m), \cdots, N_m(m)$. 第 m 层第 $i(m)$ 个神经元的线性输出记为

本文1992年11月30日收到.
*浙江省自然科学基金资助课题.

$$S_{j(m)},\quad \begin{pmatrix} j(m)=1(m),2(m),\cdots,i(m),\cdots,N_m(m) \\ m=1,2,\cdots,l. \end{pmatrix}. \tag{1}$$

其非线性输出(即实际输出)记为

$$X_{j(m)},\quad \begin{pmatrix} j(m)=1(m),2(m),\cdots,i(m),\cdots,N_m(m) \\ m=1,2,\cdots,l \end{pmatrix}. \tag{2}$$

输入层(1)的第 $j(1)$ 个神经元接收到外来信号 x 的分量记为

$$x_j,\quad (j=1,2,\cdots,N_1). \tag{3}$$

假设我们的多层神经网络模型中只有相邻两层神经元之间才有突触联系. 第 $i(m)$ 神经元与第 $i(m-1)$ 神经元之间的突触联系权重记为

$$W_{i(m),i(m-1)},\quad \begin{pmatrix} i(m)=1(m),2(m),\cdots,N_m(m) \\ i(m-1)=1(m-1),2(m-1),\cdots,N_{m-1}(m-1) \\ m=1,2,\cdots,l \end{pmatrix} \tag{4}$$

在我们的模型中，线性输出 $S_{j(m)}$ 与非线性输出(即实际输出) $X_{j(m)}$ 的表达式为

$$S_{j(m)}=\sum_{j(m-1)=1(m-1)}^{N_{m-1}(m-1)} W_{j(m),j(m-1)}X_{j(m-1)}, \tag{5}$$

$$X_{j(m)}=sgm(S_{j(m)}),\quad \begin{pmatrix} j(m)=1(m),2(m),\cdots,N_m(m) \\ m=1,2,\cdots,l \end{pmatrix}. \tag{6}$$

式中 sgm 表示 S 形曲线非线性函数. 从式(1)到式(6)即是本文讨论的多层前馈神经网络模型和针对它设计的一套简明记号. $x_{j(0)}$ = 外来输入信号 x_j.

取式(6)中的 $j(m)$ 为 $j(m-1)$，代入式(5)，得到线性输出 S 由第 $(m-1)$ 层向第 m 层逐层传递的前馈关系式

$$S_{j(m)}=\sum_{j(m-1)=1(m-1)}^{N_{m-1}(m-1)} W_{j(m),j(m-1)}sgm(S_{j(m-1)}),\quad \begin{pmatrix} j(m)=1(m),2(m),\cdots,N_m(m) \\ m=2,3,\cdots,l \end{pmatrix} \tag{7}$$

由式(7)及式(6)可得到

$$\frac{\partial S_{j(m)}}{\partial S_{j(m-1)}}=sgm'(S_{j(m-1)})W_{j(m),j(m-1)},\quad \begin{pmatrix} j(n)=1(n),2(n),\cdots,N_n(n), \\ j(n-1)=1(n-1),2(n-1),\cdots,n_{n-1}(n-1), \\ n=2,3,\cdots,l \end{pmatrix} \tag{8}$$

$$\frac{\partial S_{j(n)}}{\partial W_{i(m),i(m-1)}}=\begin{cases} sgm(S_{i(m-1)})=X_{i(m-1)}, & \text{如 } j(n)=i(m) \\ 0, & \text{如 } j(n)\neq i(m) \end{cases}$$

$$\begin{pmatrix} i(m)=1(m),2(m),\cdots,N_m(m), \\ i(m-1)=1(m-1),2(m-1),\cdots,N_{(m-1)}(m-1), \\ m=2,3,\cdots,l \end{pmatrix} \tag{9}$$

其中

$$sgm'\triangleq(S_{j(n-1)})=\frac{\mathrm{d}}{\mathrm{d}S_{j(n-1)}}sgm(S_{j(n-1)}).$$

在单输入 x 的作用下，整个多层神经网络输出端的输出为 $x_{(l)}=(x_{1(l)}, x_{2(l)}, \cdots, x_{N_l(l)})$. 现在我们要求这个输出接近预定的正确输出值 $d_{(l)}=(d_{1(l)}, d_{2(l)}, \cdots, d_{N_l(l)})$，并把下式定义的量 ε^2 称为多层神经网络在处理单输入 x 时的平方误差或瞬时平方误差：

$$\varepsilon^2 \triangleq \|d_{(l)}-x_{(l)}\| = \sum_{j(l)=1(l)}^{N_l(l)} (d_{j(l)}-x_{j(l)})^2 = \sum_{j(l)=1(l)}^{N_l(l)} (d_{j(l)}-sgm(S_{j(l)}))^2,$$

故有

$$\frac{\partial \varepsilon^2}{\partial S_{j(l)}} = -2(d_{j(l)}-sgm(S_{j(l)}))sgm'(S_{j(l)}). \tag{10}$$

训练多层神经网络的目标是修正网络中所有权重 $W_{i(m),j(m-1)}$ 的值，使得 ε^2 尽可能并尽快地单调减小到极小. 为此，可采用最陡降落法. 在最陡降落法中，修正 $W_{i(m),j(m-1)}$ 的办法为

$$\Delta W_{i(m),j(m-1)} = -\mu \frac{\partial \varepsilon^2}{\partial W_{i(m),j(m-1)}}, \tag{11}$$

式中的 μ 为描述训练速度 (即网络学习速度) 的系数. 应用最陡降落法式 (11)，剩下的问题是计算下列诸量：

$$\frac{\partial \varepsilon^2}{\partial W_{i(m),j(m-1)}}, \quad \begin{pmatrix} i(m)=1(m), 2(m), \cdots, N_m(m) \\ i(m-1)=1(m-1), 2(m-1), \cdots, N_{m-1}(m-1) \\ m=2, 3, \cdots, l \end{pmatrix} \tag{12}$$

计算应从输出层 (l) 开始，先计算式中 $m=l$ 的量，其次计算 $m=l-1$ 的量，再逐层向输入层方向计算. 这和信号传递的前馈关系式 (7) 的计算方向正好相反，因此称为“逆向传播算法”.

由于设计好了一套简明的变量记号，应用式 (10) 及式 (9) (取其中 $n=m=l$)，便得到 $m=l$ 的式 (12).

$$\frac{\partial \varepsilon^2}{\partial W_{i(l),j(l-1)}} = \sum_{S_{j(l)}} \frac{\partial \varepsilon^2}{\partial S_{j(l)}} \frac{\partial S_{j(l)}}{\partial W_{i(l),j(l-1)}} = \frac{\partial \varepsilon^2}{\partial S_{i(l)}} \frac{\partial S_{i(l)}}{\partial W_{i(l),j(l-1)}}$$

$$= -2[d_{i(l)}-sgm(S_{i(l)})]sgm'(S_{i(l)})x_{i(l-1)}. \tag{13}$$

应用式 (10)、(8) (取 $j(n)$ 为 $j(l)$，把 $j(n-1)$ 改为 $i(l-1)$) 和 (9) (取 $m=l-1$，把 $j(n)$ 改为 $i(l-1)$)，便得到 $m=l-1$ 的式 (12)：

$$\frac{\partial \varepsilon^2}{\partial W_{i(l-1),j(l-2)}} = \sum_{S_{j(l)}} \sum_{S_{i(l-1)}} \frac{\partial \varepsilon^2}{\partial S_{j(l)}} \frac{\partial S_{j(l)}}{\partial S_{j(l-1)}} \cdot \frac{\partial S_{j(l-1)}}{\partial W_{i(l-1),j(l-2)}}$$

$$= \sum_{S_{j(l)}} \frac{\partial \varepsilon^2}{\partial S_{j(l)}} \frac{\partial S_{j(l)}}{\partial S_{i(l-1)}} \frac{\partial S_{i(l-1)}}{\partial W_{i(l-1),j(l-2)}}$$

$$= \sum_{S_{j(l)}} -2(d_{j(l)}-sgmS_{j(l)})sgm'(S_{j(l)})sgm'(S_{i(l-1)})$$

$$\times W_{j(l),i(l-1)}x_{i(l-2)}, \tag{14}$$

如此继续逆向逐层计算. 不难看出，全部式 (12) 可以综合为下列方程组：

$$\frac{\partial \varepsilon^2}{\partial W_{i(m),i(m-1)}} = \sum_{S_{j(l)}} \sum_{S_{j(l-1)}} \cdots \sum_{S_{j(m)}} \frac{\partial \varepsilon^2}{\partial S_{j(l)}} \frac{\partial S_{j(l)}}{\partial S_{j(l-1)}} \frac{\partial S_{j(l-1)}}{\partial S_{j(l-2)}} \cdots \frac{\partial S_{j(m+1)}}{\partial S_{j(m)}} \frac{\partial S_{j(m)}}{\partial W_{i(m),i(m-1)}}$$

$$\begin{gathered} i(m) = 1(m), 2(m), \cdots, N_m(m) \\ \left(i(m-1) = 1(m-1), 2(m-1), \cdots, N_{m-1}(m-1)\right) \\ m = l, l-1, l-2, \cdots, 3, 2 \end{gathered} \tag{15}$$

式中所有偏微商都可由式 (10)、(8) 和 (9) (选取适当的下标后) 得到.

将式 (11) 和 (15) 与作为网络模型的式 (8)、(9)、(10) 联立，便构成多层前馈神经网络逆向传播法的整套计算公式. 公式很冗长复杂，但推导却严格简明，有助于逆向传播法的深入研究和应用.

2 一种变型 MRⅢ计算技术

MRⅢ计算技术可简述如下：将式 (15) 改写为

$$\frac{\partial \varepsilon^2}{\partial W_{i(m),i(m-1)}} = \frac{\partial \varepsilon^2}{\partial S_{i(m)}} x_{i(m-1)}, \tag{16}$$

其中

$$\frac{\partial \varepsilon^2}{\partial S_{i(m)}} = \sum_{S_{j(l)}} \sum_{S_{j(l-1)}} \cdots \sum_{S_{j(m+1)}} \frac{\partial \varepsilon^2}{\partial S_{j(l)}} \frac{\partial S_{j(l)}}{\partial S_{j(l-1)}} \cdots \frac{\partial S_{j(m+1)}}{\partial S_{i(m)}}. \tag{17}$$

虽然只要利用式 (8)、(9) 和 (10) 就可写出 $\frac{\partial \varepsilon^2}{\partial S_{i(m)}}$ 的冗长表达式，但计算是很繁琐的. MRⅢ计算技术的核心内容是用计算机实验测量出差分比值 $\frac{\Delta \varepsilon^2}{\Delta S_{i(m)}}$ 来代替 $\frac{\partial \varepsilon^2}{\partial S_{i(m)}}$，即认为式 (17) 可近似为

$$\frac{\partial \varepsilon^2}{\partial S_{i(m)}} \backsimeq \text{测量值} \frac{\Delta \varepsilon^2}{\Delta S_{i(m)}}. \tag{18}$$

于是式 (16) 近似为

$$\frac{\partial \varepsilon^2}{\partial W_{i(m),i(m-1)}} = \frac{\Delta \varepsilon^2}{\Delta S_{i(m)}} x_{i(m-1)}, \tag{19}$$

将其代入式 (11)，便得到修正多层神经网络权重 $W_{i(m),i(m-1)}$ 的 MRⅢ计算技术，即

$$\Delta W_{i(m),i(m-1)} = -\mu \frac{\Delta \varepsilon^2}{\Delta S_{i(m)}} x_{i(m-1)}. \tag{20}$$

我们提出的变型 MRⅢ计算技术，其核心内容是计算机测量的差分比值不再是 $\frac{\Delta \varepsilon^2}{\Delta S_{i(m)}}$，而改为 $\frac{\Delta \varepsilon^2}{\Delta W_{i(m),i(m-1)}}$，把它作为 $\frac{\partial \varepsilon^2}{\partial W_{i(m),i(m-1)}}$ 的近似值，即把整个式 (16) 近似为

$$\frac{\partial \varepsilon^2}{\partial W_{i(m),i(m-1)}} \backsimeq \frac{\Delta \varepsilon^2}{\Delta W_{i(m),i(m-1)}}, \tag{21}$$

再代入式 (11)，即为修正多层神经网络权重的变型 MRⅢ计算技术：

$$\Delta W_{i(m),i(m-1)} = -\mu\frac{\Delta\varepsilon^2}{\Delta W_{i(m),i(m-1)}},$$
$$i(m) = 1(m), 2(m), \cdots, N_m(m)$$
$$\left(i(m-1) = 1(m-1), 2(m-1), \cdots, N_{m-1}(m-1)\right) \tag{22}$$
$$m = l, l-1, l-2, \cdots, 3, 2$$

式 (22) 比式 (20) 要直截了当得多，但式 (22) 中待测定的差分比值 $\frac{\Delta\varepsilon^2}{\Delta W_{i(m),i(m-1)}}$ 的个数和未知数 $\triangle W_{i(m),i(m-1)}$ 的个数相当. 而式 (20) 中待测定的差分比值 $\frac{\Delta\varepsilon^2}{\Delta S_{i(m)}}$ 的个数却比未知数 $\triangle W_{i(m),i(m-1)}$ 的个数要少得多. 这可能是变型 MRⅢ式 (22) 的严重缺点，但也可能是因为 MRⅢ式 (20) 没有把计算机实测值代替繁琐计算技术充分发挥，使其反而不如式 (22) 彻底所致. 但式 (20) 与 (22) 孰优孰劣，必须根据今后足够多的计算机实验结果加以论证.

参考文献

1 Rumelhart D E, Hinton G E, Williams R J. in *Parallel Distributed Processing*, **1**. ch. 8, D E Rumelhart and J L McClelland Eds., Cambridge, MA: M.I.T. Press, 1986

2 Andes D, Widrow B, Lehr M et al. *Proc. Intl. Joint Conf. on Neural Networks*, 1990,**1**:533

3 Holler M et al. *Proc. Intl. Joint Conf. on Neural Networks*, 1989,**2**:191

4 Widrow B et al. *Proc. IEEE*, 1990,**78**:1415

A SIMPLE FORMAL DERIVATION OF MULTI-LAYERED NEURAL NETWORK BACKPROPAGATION ALGORITHM AND A VARIANT OF MRⅢ TECHNIQUE*

Chen Jishu

(*Department of Physics, Ningbo University, Ningbo, Zhejiang* 315211, *China*)

Abstract: A simple formal derivation of backpropagation algorithm for multi-layered feedforward neural networks is given. A variant of Madaline Rule Ⅲ (MRⅢ) technique is proposed.

Key words: multi-layered feedforward neural networks, backpropagation algorithm, MRⅢ technique.

* The project supported by the Natural Science Foundation of Zhejiang Province, China.