前馈神经网络结构自删除算法的研究

赵启林，陈 斌，卓家寿

（河海大学土木工程学院，江苏 南京 210098）

摘要 综述了利用删除法进行前馈神经网络设计的研究现状，并在重点分析根据隐节点输出相关性进行自删除的几种算法的基础上，在一个较高层次上提出了一种新的隐节点自删除算法.算例说明了这种算法不仅可以压缩线性相关隐节点，而且可以删除不重要的隐节点，其重新计算量也大大减小.

关键词 前馈神经网络；BP 算法；自删除算法

中图分类号：O302 文献标识码：A 文章编号：1000-1980(2000)04-0063-04

设计神经网络的方法有两种：一种是从较少的单元出发，然后逐渐地增添，以达到网络性能要求[1]；另一种是删除法，就是从一个有冗余节点的大规模网络开始，通过适当的方法，在训练过程中或训练收敛后逐步删除那些不必要的节点或权值[2].通常有以下各种算法能够在不同程度上保证删除那些冗余的节点.

a. 复杂性调整方法，即在网络误差函数中引入表示网络结构复杂性的惩罚项，通过在优化过程中降低网络结构的复杂性[3].目标函数一般定义为如下形式：

$$J(W) = E(W) + \lambda C(W) \tag{1}$$

式中：$E(W)$——网络误差项；$C(W)$——网络结构复杂性惩罚项；λ——调整参数.这种方法存在主要问题是 $C(W)$的定义不统一，而且对 λ 的取值比较敏感[4].

b. 灵敏度计算方法，即预先估计经过训练的各节点或连接对于误差函数的重要程度，然后删除那些灵敏度值较小的节点或连接[5].但实际应用中，本文作者发现由于出现线性相关项，连接权值往往出现许多相同值，即不同连接的灵敏度完全相同而无法删除的问题.

c. 相互作用的修剪方法[6]、增益方法[7]等，本文不再赘述.

d. 本文将重点研究的相关性自删除算法.该法根据隐节点输出的相关性程度进行网络调整，即将相关性程度较大的节点进行合并，达到简化网络结构的目的.

1 相关性自删除算法

文献[8]中认为在网络训练时，各节点的输出可看作是随时间出现的序列.衡量隐层节点输出相似性的方法就是计算这些时间序列的互相关系数，其计算式为

$$\rho(o_i^{(m)} o_j^{(m)}) = \frac{\mathrm{COV}(O_i^{(m)}, O_j^{(m)})}{\sqrt{D(O_i^{(m)})}\sqrt{D(O_j^{(m)})}} \tag{2}$$

式中：$\rho(o_i^{(m)} o_j^{(m)})$——$O_i^{(m)}$和 $O_j^{(m)}$的互相关系数；$O_i^{(m)}$，$O_j^{(m)}$——第 m 层节点 i 与节点 j 的输出；$D(\cdot)$——方差；$\mathrm{COV}(O_i^{(m)}, O_j^{(m)})$——协方差，

$$\mathrm{COV}(O_i^{(m)}, O_j^{(m)}) = E\{[O_i^{(m)} - E(O_i^{(m)})][O_j^{(m)} - E(O_j^{(m)})]\} \tag{3}$$

式中 $E(\cdot)$表示均值.

当 $\rho(o_i^{(m)} o_j^{(m)}) = 0$ 时，表示 $O_i^{(m)}$与 $O_j^{(m)}$是不相关的；当 $\rho(o_i^{(m)} o_j^{(m)})$越接近 1 时，则说明 $O_i^{(m)}$与 $O_j^{(m)}$的线性相关程度越大.若两个神经元的互相关系数大于给定的互相关系数阈值时，则对这两个节点进行合并.

收稿日期：1999-01-19

作者简介：赵启林(1972—)，男，江苏淮安人，博士研究生，水工结构专业.

文献[9,10]则从矩阵奇异值分解的角度来压缩线性相关部分.设第 i 隐层有 n_i 个节点,对于 m 个输入样本,收敛后的网络第 i 隐层输出可以用 $m \times n_i$ 阶矩阵 $\boldsymbol{A}_i$ 来表示.对 $\boldsymbol{A}_i$ 进行奇异值分解可得如下形式:$\boldsymbol{A}_i = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$,其中 $m \times m$ 阶矩阵 $\boldsymbol{U}$ 和 $n \times n$ 阶矩阵 $\boldsymbol{V}$ 是正交矩阵;$\boldsymbol{\Sigma}$ 是由 $\boldsymbol{A}$ 的奇异值构成的 $m \times n$ 阶对角矩阵,$\boldsymbol{\Sigma} = [\mathrm{diag}\{S_1, \cdots, S_p\} : 0]$,$p = \min(m, n)$,$S_1 \geqslant S_2 \geqslant \cdots \geqslant S_p \geqslant 0$,$S_1, \cdots, S_p$ 称为矩阵 $\boldsymbol{A}$ 的奇异值.知道矩阵 $\boldsymbol{A}$ 的 Frobenius 范数为

$$\| \boldsymbol{A} \|_{\mathrm{F}} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} | a_{ij} |^2\right)^{\frac{1}{2}} \tag{4}$$

通过证明可得

$$\| \boldsymbol{A} \|_{\mathrm{F}}^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 = \sum_{i=1}^{p} S_i^2 \tag{5}$$

如果矩阵 $\boldsymbol{A}_i$ 的 p 个奇异值中有 q 个奇异值起主导作用,就能说明可以从 $\boldsymbol{A}$ 中选择一个 $m \times q$ 阶的 $\boldsymbol{B}$ 矩阵来逼近 $\boldsymbol{A}$.从另一个角度来说,从隐层节点中选取一个子集就可消除多余部分,达到删除多余隐层节点的目的.

但是通过以上分析,我们可以看出这两种设计方法都存在明显不足.利用互相关系数进行删除时,只能删除互相关性强的隐节点,同时由于是一个时序判断法,有可能出现未收敛时过多删除隐节点的情况,以及在整个时序中不断进行互相关系数的计算与判别,增加了每一次迭代的计算时间.虽然利用奇异值分解不仅可以压缩线性相关部分与不重要的部分,但删除后的网络必须重新初始化,重新训练,增加了计算时间.

针对以上问题,本文从无限维空间内积的概念出发,提出了一种新的自删除算法.该算法不仅可以压缩线性相关项与删除不重要的项,而且删除后的网络结构只需要一次矩阵运算或少量的隐层与输出层之间的权值调整就可以重新形成网络的权值连接.

2 网络自删除设计的新方法

前馈神经网络主要应用于模式分类、函数近似与样本回归问题.对于函数近似和样本回归问题输出单元一般用线性单元,并不设置阈值,因为设置阈值造成的不连续性易引起函数近似与样本回归计算上的困难[11].

对于真实函数 $f(X)$ 而言,网络的近似函数可以表达为

$$f_u(\boldsymbol{X}) = \sum_{i=1}^{m} \boldsymbol{\beta}_i g_i(\boldsymbol{X}) \tag{6}$$

其中

$$g_i(\boldsymbol{X}) = 1/(1 + \mathrm{e}^{-(\boldsymbol{a}_i \boldsymbol{X} - T_i)}) \tag{7}$$

式中:$\boldsymbol{X}$——输入向量,$\boldsymbol{X} = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}$;$\boldsymbol{\beta}_i$——隐层与输出层的连接权向量,$\boldsymbol{\beta}_i = (\beta_{i1}, \cdots, \beta_{in})^{\mathrm{T}}$;$n$——输出层单元数;$g_i$——隐层非线性处理函数;$\boldsymbol{a}_i$——输入层与隐层的权值向量,$\boldsymbol{a}_i = (a_{1i}, a_{2i}, \cdots, a_{mi})$;$T_i$——第 i 个隐节点的阈值.

从式(6)可以看出前馈神经网络实际上是变基函数空间上($g_i(X)$ 是随 $\boldsymbol{a}_i$ 与 T_i 变化而变化)的函数逼近与样本回归,这样它就比传统的函数逼近,样本回归方法有更广泛的适应性.

首先,给出几个定义 在可变基函数空间 $\{g\} \in L^2$(L^2 表示定义域 X 中平方可积的函数空间)中:$\forall\, g_i, g_j \in \{g\}$,内积$\langle g_i, g_j \rangle$定义为

$$\langle g_i, g_j \rangle = \int_{\boldsymbol{X}} g_i(\boldsymbol{X}) g_j(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \tag{8}$$

内积导出的范数 $\| \cdot \|$ 则可以定义为

$$\| g_i \| = \left[\int_{\boldsymbol{X}} g_i^2(\boldsymbol{X}) \mathrm{d}\boldsymbol{X}\right]^{\frac{1}{2}} \tag{9}$$

根据文献[12]可知上式符合内积的定义.同时,由内积与范数可以定义函数 g_i 与 g_j 的夹角余弦 $\cos\theta$.

$$\cos\theta = \frac{\langle g_i, g_j \rangle}{\| g_i \| \cdot \| g_j \|} \tag{10}$$

则 $\cos\theta$ 的值就反映了 g_i 与 g_j 的相关程度.当 $\cos\theta = 0$ 时,表示 g_i 与 g_j 正交;$\cos\theta \to 1$ 时,表示 g_i 与 g_j 相

关程度越大.

范数代表了 g_i 的长度,可以用下式反映各隐节点的重要程度:

$$\lambda_i = \frac{|\beta_i| \cdot \|g_i\|}{\|f(X)\|} \quad (i = 1,\cdots,m) \tag{11}$$

对于样本回归问题,其中 $f(X)$一般情况下是未知的,可以用式(6)中的近似函数 $f_\mu(X)$代替,进行近似计算,可以有下式:

$$\lambda_i = \frac{|\beta_i| \cdot \|g_i\|}{\|f_\mu(X)\|} = \frac{|\beta_i| \cdot \|g_i\|}{\|\sum_{j=1}^{\mu}\beta_j g_j\|} \tag{12}$$

式中 $f_\mu(X)$即可取式(6)中的近似函数.

λ_i 的值越小则说明 g_i 的重要程度越小,当小于一个给定的阈值时,则可以自动删除.

这样可以根据式(10)和式(12)删除相关性较大与重要程度较小的隐节点,形成新的网络结构.易知保持基函数不变,只要利用 BP 算法适当调整 β_j 的值就可以得到新的权值连接.这可以通过以下分析得知.

对复杂性网络结构,经过训练后可得 $f_\mu(X) = \sum_{j=1}^{\mu}\beta_j g_j(X)$,若 g_k 与 g_l 线性相关,那么有 $g_k = rg_l$. r 为一个待定常数,那么调整后的近似函数 $f_{\mu-1}(X) = \sum_{j=1}^{\mu}\boldsymbol{\beta}_j g_j(X) + (\boldsymbol{\beta}_k r + \boldsymbol{\beta}_l) g_{\mu-1}$. 于是问题变成了 r 的确定.这样,对调整后的网络结构,保持隐层与输入层的权值及隐层的阈值不变,用一般的 BP 算法,就可以很快得到调整后的 $\boldsymbol{\beta}_j$ 的权值.这样所需的计算量很小.

算法的具体步骤如下:

a. 选择复杂性结构,对权值、阈值初始化,用‘批处理’算法学习[13]至收敛;

b. 利用式(10)判别隐节点基函数的相关性,利用式(12)确定每一个隐节点的重要程度;

c. 将相关性大的隐节点合并,重要性小的隐节点删除;

d. 对于保留的隐节点与输入单元的权值,以及隐节点的阈值保持不变,利用 BP 算法对隐节点与输出单元的权值进行学习至收敛[13].

3 算　例

为了说明本方法的可行性与首效性,我们选择了文献[13]算例 13.3 中的前 10 个样本,分别用矩阵奇异值分解与本算法进行了对比研究.在该算例中,输入向量分别为水温(℃)、透明度(m)、DO 浓度(mg/L)、盐分浓度(‰)、过滤后的COD浓度(mg/L)、输出向量为大阪湾的 COD 浓度.本算例的目的是通过以往的实测数据来预测大阪湾海水的 COD 实际浓度.取复杂性结构 5—10—1,整体误差 0.04.利用奇异值分解得到优化网络结构 5—2—1,利用本算法得到优化结构 5—1—1.对两个新的优化结构用 BP 算法学习至收敛的对比见图 1.由图可知,本算法由于保留基函数的特征性,不仅学习次数少,而且每次迭代所需计算量也大幅度减少.但利用奇异值分解得到优化结构,所有权值与阈值都得初始化,迭代次数成倍增加,而且每次迭代的计算量也是本算法的数倍(因为它得同时调整输入层与隐层的权值和隐层与输出层的权值、阈值).

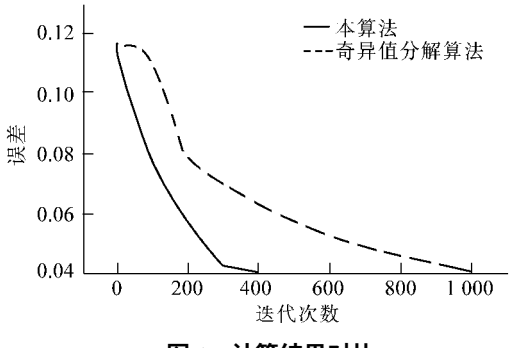

图 1　计算结果对比

Fig.1　Comparison of computational outcome

4 结　论

a. 本文在内积和范数概念的基础上定义隐层节点的相关性与重要性,以确定隐节点的合并与删除,概念清晰,方法易于编程.

b. 本方法由于保留了剩余基函数的特征性,重新形成网络的权值连接时,所需的计算量小.

参考文献：

[1] Yoshio Hirose, Koichi Yamashita, Shimpei Hijyia. Back-propagation algorithm which varies the number of hidden units[J]. Neural Networks, 1991,2(4):61 ~ 66.
[2] 何述东,瞿坦,黄献青,等. 多层前向神经网络结构的研究进展[J]. 控制理论与应用,1998,15(3):313 ~ 319.
[3] Hinton G E,Nowlan S J. How learning can guide evolution[J]. Complex Systems, 1987,1(3):495 ~ 502.
[4] Weigend A S, Rumelhart D E,Hu berman B A. Generalization by weight-elimination applied to currency exchange rate predittion[A]. In: Zornetzwr S F. Proc of 1991 IJCNN[C]. San Diego:Academic Press,1991. 837 ~ 841.
[5] Karnin E D. A Simple procedure for pruning back-progation trained neural networks[J]. IEEE Trans Neural Networks, 1990,1(2):239 ~ 242.
[6] Sietsma J,Dow R J E. Creating artificial neural networks that generalize[J]. Neural Networks, 1991,2(1):67 ~ 69.
[7] Kruschke J K. Benefits of gain: speeded learning and minimal hidden layers in back propagation networks[J]. IEEE Trans Syst Man and Cybern, 1991,1(1):273 ~ 280.
[8] 汪家道,孔宪梅,陈大融,等. 节点自删除神经网络及其在磨粒识别中的应用[J]. 清华大学学报,1998,38(4):42 ~ 46.
[9] 陆系群,余英林. 设计前馈神经网络结构的一种新方法[J]. 华南理工大学学报,1998,26(1):112 ~ 115.
[10] Psichogios D C,Ungar L H. SVD-NET: An algorithm that antomatically selects network structure[J]. Neural Networks, 1994,5(3):244 ~ 248.
[11] Tin-You Kwok,Dit-Yan Yeung. Constructive algorithms for structure learning in feedforword neural networks for regression problems[J]. Neural Networks,1997,8(3):630 ~ 645.
[12] 邓述渝. 应用泛函分析[M]. 南京:河海大学出版社,1995. 146 ~ 151.
[13] 越振宇,徐用懋. 模糊理论和神经网络的基础与应用[M]. 北京:清华大学出版社,1996. 173 ~ 180.

Self-deleting Algorithm of Feedforward Neural Networks

ZHAO Qi-lin, CHEN Bin, ZHUO Jia-shuo

(*College of Civil Engineering, Hohai Univ., Nanjing* 210098, *China*)

Abstract: In this paper, recent development of self-deleting algorithm of feedforward neural networks is surveyed. On the basis of the analysis of the self-deleting algorithm according to the similarity of hidden nodes output, a new self-deleting algorithm of hidden nodes, which synthesized the similarity of hidden nodes and the importance of them, is put forward. The example proves that this algorithm not only reduces linearly related hidden nodes, but also deletes unimportant nodes, reducing computation greatly.

Key words: feedforward neural network; BP algorithm; self-deleting algorithm