用于神经网络容错的动态冗余 BP 算法 [1]

许荔秦 胡东成

(清华大学自动化系 北京 100084)

摘 要 多层感知器(MLP)的容错性传统上采用改进算法和部件冗余方法. 该文提出了一种动态冗余 BP 算法, 这种方法在传统的带冲量项的自适应 BP 算法的学习过程中, 根据各权值"重要度"的不同选取重要的权值进行冗余处理. 该算法能有效地提高网络的容错能力, 与学习中注入故障这一典型的容错改进算法相比, 尽管容错能力并不突出, 但相对可节省大量的学习时间.

关键词 多层前向神经网络, 容错, 冗余, 动态冗余

中图号 TN-052

1 引 言

多层感知器 (MLP) 在许多问题中得到了广泛的应用. 但是, 在一些重要应用中要求 MLP 有较高的可靠性, 也就是说, 要求 MLP 有较强的容错性, 随着 MLP 硬件实现的进展, 这一点将显得更为重要.

MLP 的容错性在近几年来已得到了很大的重视, 已有不少文献做过这方面的研究. 从研究方法上来看, 提高 MLP 的容错性主要有两种方法: 改进 MLP 的学习算法 [1−3] , 在网络学习后对 MLP 进行结构冗余 [4−6] . 但两种方法各有缺点: 冗余方法要耗费大量硬件资源, 这样冗余后的网络更为庞大, 规模无法很好地控制; 改进学习算法需要牺牲大量的学习时间, 而在故障模型复杂及网络规模加大之后有时不能保证收敛.

解决这一矛盾可以用一种动态冗余容错性设计方法, 在用传统的 BP 算法进行学习的同时, 动态地对隐层神经元进行冗余处理, 最终得到的网络比原网络的容错性有较大的提高, 由于在学习中进行冗余处理, 因此学习时间比改进算法小, 而网络规模在学习中进行改进, 因此也易于控制.

2 研究对象与模型

本文研究网络为模式识别中应用最为广泛的单隐层前向网络, 用于识别器中, 其输出层神经元作用函数为硬限幅函数, 隐层神经元作用函数为 sigmoid . 用 $W_{ij}^{(n)}$ 表示连接第 $n-1$ 层第 j 个神经元与第 n 层第 i 个神经元的权值, 用 $\mathrm{bias}_i^{(n)}$ 表示第 n 层第 i 个神经元的偏置, 用 $O_i^{(n)}$ 表示第 n 层第 i 个神经元的输出, 用 I_i 表示第 i 个输出层神经元的净输入, 则有

$$I_i = \sum_{j=1}^{N_1} (W_{ij}^{(2)} O_j^{(1)} - \mathrm{bias}_i^{(2)}) \tag{1}$$

$$O_i^{(2)} = \mathrm{sgn}(I_i) \tag{2}$$

其中 N_1 为隐层神经元数目, $\mathrm{sgn}(\,)$ 为硬限幅函数, $\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$.

[1] 1999-09-29 收到, 2000-02-03 定稿

国家自然科学基金和清华大学博士学位论文基金的资助项目

传统的学习算法为 BP 算法，为了提高收敛速度以及收敛到全局最优点，可以采用带冲量项的自适应学习算法，算法如下：

$$\Delta W_{ij}^{(l)}(t+l) = mc \bullet \Delta W_{ij}^{(l)}(t) + \mathrm{lr}(t) \bullet mc \bullet \frac{\partial J}{\partial W_{ij}^{(l)}(t)} \tag{3}$$

其中 mc 为冲量项，为一常量，可以取 0.9， J 为期望输出与实际输出之间的方差， $\mathrm{lr}(t)$ 为学习速率，为一自适应量，当 J 减少时， $\mathrm{lr}(t+1) = \mathrm{lr}(t) \times 1.05$，当 J 增加到上一步的 1.05 倍时， $\mathrm{lr}(t+1) = \mathrm{lr}(t) \times 0.7$。由于采用 BP 算法，要求各层作用函数可微，因此在学习中令输出层作用函数为 sigmoid 函数。而在后面的仿真实验中令输出层作用函数为硬限幅函数。

MLP 中最常见的故障为权值连接断路与神经元损坏，本文仅针对这两种故障进行讨论。对于权值连接断路故障，可以用权值 $W_{ij}^{(n)}$stuck-at-0s ($W_{ij}^{(n)}$s-a-0) 来表示；对于神经元损坏，主要表现为神经元输出为正电源或负电源电位，因此可以用 $O_i^{(n)}$ stuck-at-1($O_i^{(n)}$s-a-1) 或 stuck-at-(−1)(s-a-(−1)) 来表示。同时，由于输入层与输出层单元故障为非常严重的故障，会严重影响网络输出，因此不在本文研究范围内。本文所设定的故障模型为单故障模型，即整个网络在运行中最多只发生一个故障，包括权值故障与隐层神经元 bias 故障。

本文只对单故障进行考虑的原因在于：对于一个较为可靠的系统（系统中任一部件发生故障的概率 P_f 较小，可认为 < 0.2），单故障在故障发生中占绝大多数。为简单计，不妨设各部件发生故障的随机变量独立同分布，概率都为 P，总部件数为 N，则发生故障的概率 $P_f = 1-(1-P)^N$，而发生单故障的概率为 $P_s = NP(1-P)^{N-1}$，我们对 P_s/P_f 进行计算，即可得单故障在故障总数中所占比例，对 P 取值 0.1~0.00001，对 N 取值 10~10000，所得结果如图 1 所示。由图可知， $P_f < 0.2$ 时，单故障在总故障中所占比例 $> 90\%$，传统上的神经网络是比较可靠的，因此本文对于单故障的分析还是有较大可信度的。

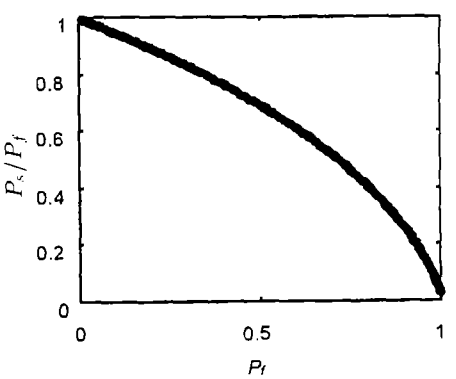

图 1　不同故障发生率下单故障占总故障中的比例

3 动态冗余 BP 算法

传统的学习算法未考虑将各权值的重要度平均分配，因此导致各权值重要度分布不均，使得 MLP 的容错性受到很大限制。对网络冗余最有效的是将最“重要”的权值进行冗余，使其分布均匀，最直接的方法是对与之相连的隐层神经元进行冗余。这其中最重要的是对权值“重要度”的判断。

故障对于网络的影响表现在于对样本的识别错误. 当某个输出层神经元的净输入 I_i 在故障的作用下符号发生改变时，其网络的输出即出现错误. 由 (1) 式可知，当权值故障与隐层神经元故障发生时，I_i 的符号都可能发生变化. 对于某个样本，记无故障发生时 I_i 的值为 I_d，不妨设 $I_d>0$，若发生 $W_{ij}^{(2)}$ s-a-0 故障，那么当 $W_{ij}^{(2)}O_j^{(1)}<-I_d$ 时，I_i 的值将发生翻转，在此故障下对于 I_i 的最大可能改变为 $\left|W_{ij}^{(2)}O_j^{(1)}\right|$. 同样，若发生神经元 s-a-1$(-1)$ 故障，对于 I_i 的最大可能改变为 $2\left|W_{ij}^{(2)}O_j^{(1)}\right|$. 若 $|I_d|$ 小于此最大可能改变，且符号满足要求，则故障将影响到此样本的正确识别. 同时也可看出，隐层神经元故障是比权值故障更为严重的故障，若网络可以对所有类型的隐层神经元故障容错，同样也可以对所有的权值故障容错.

由上面的分析可知，对于一个容错性较好的网络，应当使得 $\left|W_{ij}^{(2)}O_j^{(1)}\right|$ 相对于 $|I_d|$ 的值尽量小. 对于一个经训练得到的网络，由于 W_{ij} 分布的随机性和 $O_j^{(1)}$ 输出的随机性，无法估算 I_d 的大小，但是可以预见的是，若权值分布平均，即 $|W_{ij}^{(2)}|/\Sigma_j|W_{ij}^{(2)}|$ 尽量小，则网络的容错性越强. 对于一个相对于 $\Sigma_j|W_{ij}^{(2)}|$ 很大的权值 $W_{ij}^{(2)}$，可以将与之相连的隐层神经元做冗余处理，这样使得所有与之相连的权值尽量小，这样就可以提高网络的容错性. 以上的讨论未涉及第一层的权值，是因为第一层权值故障只影响 $O_j^{(1)}$，因此可归并为隐层神经元的故障. 基于以上分析，我们提出了以下算法：

动态冗余 BP 算法：

设置初始训练步数 epoch(0)=minEpoch，初始隐层神经元 $N_h(0)=N_h0$, (minEpoch 和 N_h0 为常数) 对神经网络各参数赋予初值，令 $t=0$，

(1) 令最大训练步数为 epoch(t)，对网络进行训练，算法如 (3) 式；

(2) 对各输出神经元 i，计算所有与之相连的权值绝对值之和 $\mathrm{ABSW}_i=\Sigma_j|W_{ij}^{(2)}|$，对每一个隐层神经元 j，若对任何一个 i，有 $\frac{|W_{ij}^{(2)}|}{\mathrm{ABSW}_i}>C(C$ 为常数)，则 $\mathrm{Re}(j)=1$，若对所有输出神经元 i 计算的结果都有 $\frac{|W_{ij}^{(2)}|}{\mathrm{ABSW}_i}\le C$，则 $\mathrm{Re}(j)=0$；

(3) 若对所有 j，有 $\mathrm{Re}(j)=0$，则令 $\mathrm{epoch}(t+1)=\min\{\mathrm{epoch}(t)\times \mathrm{ie}$，$\max\mathrm{Epoch}\}$, (ie 和 maxEpoch 为常数)，转入 (4)；否则对隐层神经元 j 进行如下处理：与输出层相连的各权值连接变为原值 $W_{ij}^{(2)}$ 的 1/2，与输入层相连的各权值连接不变，隐层单元 $\mathrm{bias}_i^{(1)}$ 不变，输出层单元 $\mathrm{bias}_i^{(2)}$ 不变，同时添加一个对隐层神经元 j 的冗余隐层神经元，该隐层神经元的参数与变化后的隐层神经元 j 完全一致；对所有隐层神经元做完处理后，令 $\mathrm{epoch}(t+1)=\max\{\mathrm{epoch}(t)/\mathrm{ie},\min\mathrm{Epoch}\}$，由于网络拓扑结构发生了变化，因此学习算法 (3) 式中各惯性量 $\Delta W_{ij}^{(t)}(t+1)=0$. 对某个隐层神经元的冗余结构如图 (2) 所示.

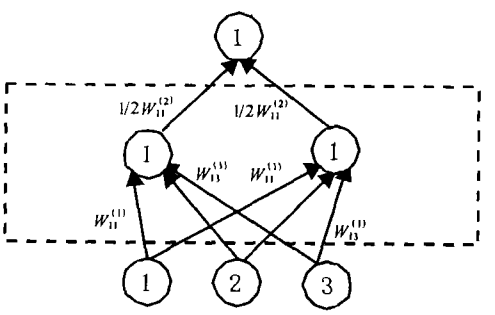

图 2 对隐层神经元的冗余结构

(4) 若对所有 j，有 $\mathrm{Re}(j)=0$，且步骤 (1) 中在训练步数内收敛，则算法结束，结果收敛；若步骤 (1) 中在训练步数内不收敛且总训练步数 AllEpoch=$\Sigma_t$epoch$(t)>\max$ AllEpoch (max AllEpoch 为常数)，则算法结束，结果不收敛；否则 $t=t+1$，转至 (1).

算法中 N_h0 为初始隐层神经元数，可取 4；minEpoch 与 maxEpoch 可分别取 25，100；ie 可取 1.05~1.1，ie 为一个自适应学习系数，可以加快算法的收敛速度；maxAllEpoch 针对不同规模的网络取不同的数；最重要的一个参数为 C，C 的倒数表示了冗余单个权值占所有权值总和的大小，因此其大小决定了所得网络的规模和容错性，其值越大，则所得到的网络权值越平均，容错能力越强，但同时网络规模越大。根据网络规模的大小不同，C 一般选取 3~6。在下面的例子中取值 4。

4 几种算法的比较

Behnam S. Arad 提出了一种提高网络容错性的 SC 算法 [3]。这种算法采用训练中对网络注入故障，具体是选取一个故障集 K(隐层神经元 s-a-1 及 s-a-(−1) 故障)，在 BP 学习中的每一步中，改变优化函数为 $E = \Sigma_{k\in K} E_k$，其中 E_k 为注入其中一个故障后的优化函数，再采用 BP 或其改进算法进行学习。采用这种算法可以大幅度提高网络的容错性，但是由于在每一步中都须注入所有类型的故障，因此要耗用大量的学习时间，对于上面的故障集，一个隐层神经元数为 N_h 的网络所需的时间大致为 BP 算法的 $2 \times N_h$ 倍。

我们用一个分类器的仿真实验进行算法的比较。采用 MLP 进行数码识别，'0', ···, '9'10 个数字采用 7 段数码管表示，每段用 3 点表示，因此，输入向量为 21 维，输出 10 维，为识别结果。3 种算法分别为自适应冲量 BP 算法 (BP)，动态冗余 BP 算法 (BPR) 和 SC 算法。算法在 PC 机上用 Matlab 实现。PC 配置为 Celeron300A CPU，运行于 Windows98 系统上。对于网络容错性能的评估采用计算所有样本在各类故障下发生分类错误的概率。对输入数据进行学习得到网络，BP 及 SC 算法中隐层神经元数目采用 14，动态冗余 BP 算法中 ie=1.1，$C = 4$，maxAllEpoch=1000。采用不同初值分别进行 50 次学习，对所得网络进行统计分析，可得表 1 中数据：

表 1　三种算法的结果比较

	N_h	P_w	P_b	epochs	time(s)
BP	14	1.03×10^{-3}	0.184	161	8.89
BPR	13.8	1.81×10^{-5}	0.0528	360	30.7
SC	14	1.38×10^{-5}	8.43×10^{-3}	181	231

表中各符号意义：N_h 隐层神经元数，P_w 所有样本在所有权值单故障下发生分类错误的概率，P_b 所有样本在所有隐层神经元单故障下发生分类错误的概率，epochs 学习迭代步数，time 学习时间，单位为秒。

值得注意的是，对于 BPR 算法，在 53 次实验中有 3 次实验不收敛，上表中数据由其中的 50 次收敛实验统计而得。

由数据可以看出，BPR 及 SC 算法都能大幅度提高网络的容错能力，而 SC 算法更胜一筹，尤其是对于隐层神经元的故障。但是通过比较训练时间可知，BPR 所需时间是 BP 算法的 3~4 倍，而 SC 算法则需 20 倍以上。因此从训练时间上讲，BPR 有一定的优势，在权值故障在网络故障中占主要故障时，由于两种算法对于权值容错的概率相差不多，BPR 算法更能体现其优势。

5 结　论

本文提出了在神经网络设计中采用动态设计的方法，在改进传统的算法的同时，动态地对某些主要部件进行冗余处理，即经几步学习之后，分析是否有很重要的节点与权值，如有，则做一定的冗余处理。这种学习算法可以有效地控制网络的规模，同时在对学习时间牺牲不是很大的情况下对网络的容错性有显著的提高。

这种算法同时也存在一定的问题，如在学习中并不是每一次都能收敛，而控制网络规模的常数的选取还只能尝试或凭经验得到。因此研究算法的收敛性及常数 C 的选取是今后的一个研究方向。

参 考 文 献

[1] A. F. Murray, P. J. Edwards, Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training, IEEE Trans. on Neural Networks, 1994, NN-5(5), 792–802.

[2] P. Kerlinzin, P. Refregier, Theoretical investigation of the robustness of multilayer perceptrons, Analysis of the linear case and extension to nonlinear networks, IEEE Trans. on Neural Networks, 1995, NN-6(3), 560–571.

[3] B. S. Arad, A. El-Anway, On fault tolerant training of feedforward neural networks, Neural Networks, 1997, 10(3), 539–557.

[4] D. S. Phatak, I. Koren, Complete and partial fault tolerance of feedforward neural nets, IEEE Trans. on Neural Networks, 1995, NN-6(2), 446–456.

[5] D. S. Phatak, I. Koren, Fault tolerance of feedforward neural nets for classification tasks, IJCNN, Nagoya, Japan, 1992, II-386–391.

[6] Xu Liqin, Hu Dongcheng, A new component redundancy method in MLP's fault tolerance, ICEMI'99, Harbin, China, 1999, I-72–76.

A DYNAMIC REDUNDANCY BP ALGORITHM APPLIED IN THE FAULT TOLERANCE OF NEURAL NETWORKS

Xu Liqin Hu Dongcheng

(*Department of Automation, Tsinghua University, Beijing* 100084, China)

Abstract There are two main types of approaches in the research of fault tolerance of Multilayer Perceptrons(MLP): improvement in the learning algorithm and component redundancy after training. A dynamic redundancy BP algorithm is presented. In the training steps of the conventional adaptive BP algorithm with a momentum term, the most important weights are replicated based on their "significance". Applying the algorithm the fault tolerance of a network can be improved effectively. Compared with fault injection while training — a typical improved learning algorithm, although this dynamic redundancy algorithm gives no prominence in fault tolerance, the training time can be greatly reduced.

Key words Multilayer perceptrons, Fault tolerance, Redundancy, Dynamic redundancy

许荔秦： 男，1973 年生，博士，主要研究方向为人工神经网络的鲁棒性分析与容错性设计.

胡东成： 男，1946 年生，教授，博士生导师，主要研究方向为自动测试、故障诊断与可靠性.