一种新的模糊规则动态调整正则项系数的神经网络学习方法*

武 妍[1,2] 张立明[2]

([1]同济大学计算机科学与工程系，上海，200331；
[2]复旦大学电子工程系，上海，200433)

摘要 从偏差-方差模型出发，提出了一种通过模糊规则推理动态调整正则项系数的新方法，并有效地确定了模糊推理规则和隶属度函数．并将该方法与 BP 算法和固定正则项系数的方法进行了比较，该方法具有精度高、收敛快和泛化能力高等优点，通过实例表明了该方法的有效性．

关键词 神经网络，模糊规则推理，泛化能力，正则化．

A NOVEL NEURAL NETWORK LEARNING METHOD OF DYNAMICALLY TUNING REGULARIZATION COEFFICIENT ACCORDINT TO FUZZY RULES*

WU Yan[1),2)] ZHANG Li-Ming[2)]

([1)]Department of Computer Science and Engineering, Tongji University, Shanghai 200331, China;
[2)]Department of Electronic Engineering, Fudan University, Shanghai 200433, China)

Abstract Based on bias-variance model, a novel method of dynamically tuning the regularization coefficient by fuzzy rules inference was proposed. The fuzzy inference rules and membership functions were effectively determined. Furthermore, the method was compared with the traditional BP algorithm and fixed regularization coefficien's method. The result is that the proposed method has the merits of the highest precision, rapid convergence and best generalization capacity. The capacity proposed method is shown to be a very effective method by several examples simulation.

Key words neural network, fuzzy rule inference, generaliation capacity, regularization method..

引言

随着多层前馈神经网络的广泛应用，泛化(generalization)问题日益引起了人们的重视．迄今为止对泛化问题的研究主要集中在两个方面：(1) 网络确定，需要多少学习样本数可保证泛化性能；(2) 学习样本数确定，如何提高泛化性能．对问题(1)的研究通常是在 VC 维理论的框架中进行的，相比之下，问题(2)更具有实际意义．由于最优网络的确定是一个 NP 完全问题，因此现在一般研究的是给定网络结构，如何提高泛化性能的问题．通常采用各种"正则化"法(regularization)[1~2]来解决这一问题．但也存在训练时间的增加、以及如何选择正则项系数等问题．特别是正则项系数选择是十分关键的，采取固定正则项系数的方法[3~5]，如果选择的不恰当，为达到同样的误差往往要花费更长的训练时间，有的效果甚至还不如传统的 BP 算法．Weigend 等[6]提出了一种权值终止法，并给出了几条在训练过程中动态调整正则系数的试探方法，该方法用到了太阳黑子等预测问题中，获得了较好的效果，但其对正则项系数的调整较粗略，而且没有给出增加和减少量的系数选取．另外，在训练过程中对其的选取非常敏感，为了更好地完成训练过程，并获取较好的泛化能力，需要一种能根据训练过程的变化，很好地给出正则项系数的变化量，以适应其变化需要的方法．

本文提出了采用模糊规则推理动态调整正则项

* 国家自然科学基金(批准号 39870194)资助项目
稿件收到日期 2001-09-27，修改稿收到日期 2002-02-21

* The project supported by the National Natural Science Foundation of Chinan(No. 39870194)
Received 2001-09-27, revised 2002-02-21

系数的方法，并将此方法用到非线性函数逼近、实际数据的模式分类等问题中，取得了很好的效果.

1 偏差-方差模型

当数据集的样本数无限时，误差平方和函数可以写成[2]

$$E = \frac{1}{2}\int |y(x) - <t|x>|^2 p(x)dx + \frac{1}{2}\int |<t^2|x> - <t|x>^2| p(x)dx, \quad (1)$$

其中，$p(x)$是输入数据的概率分布，$<t|x>$是目标数据的条件平均.式(1)中的第二项独立于网络函数$y(x)$.要获得优化的网络函数只需最小化误差平方和，即让$y(x)=<t|x>$，使得式(1)中的第一项消失.第二项表示了数据中的内在噪声而且在误差中设置了一个下限.

在实际应用中，我们必须处理有限数据集的问题.假定我们用包含N个模式的训练数据集D来决定网络模型$y(x)$.现在来考虑所有可能的数据集，而且都取自相同的联合概率分布$p(x,t)$.式(1)中第一项的积分项$|y(x)-<t|x>|^2$的值将取决于它被训练的特定数据集D.我们可以通过考虑在整个数据集上的平均来终止这种依赖性，这种均值(即期望)可以记作

$$\varepsilon_D[|y(x)-<t|x>|^2]. \quad (2)$$

我们将式(2)分解成

$$\varepsilon_D[y(x)-<t|x>|^2] = |\varepsilon_D[y(x)]-<t|x>|^2 + \varepsilon_D[|y(x)-\varepsilon_D[y(x)]|^2]. \quad (3)$$

式(3)中的第一项称为偏差的平方，第二项称为方差.偏差用来测量网络函数$y(x)$在所有数据集上的平均同目标函数$<t|x>$的不同的程度；方差用来测量网络函数$y(x)$对特定数据集选择的敏感性.

从上面的讨论可以看出，神经网络的训练是在偏差和方差之间的一种自然的平衡.

2 模糊规则推理动态调整正则项系数的方法

2.1 正则化方法及存在问题

在"正则化"方法中，算法的最大变化是在误差函数中增加一惩罚项Ω，增加惩罚项后的总体的误差函数为

$$\widetilde{E} = E + \gamma\Omega. \quad (4)$$

其中，E是通常的误差平方和函数，与前文的偏差相对应，Ω为正则函数，γ为正则项系数，$\gamma\Omega$与前文的方差项相对应.

网络函数$f(x)$对训练数据很吻合的话，将会导致E的值很小，函数$f(x)$很平滑的话，将会导致Ω的值很小.网络训练的结果是在最小化E和最小化正则项Ω之间的一种折衷，这种方法的主要缺点是难以得到最优参数，它对正则项系数很敏感.如果系数太小，将不会对复杂性起到惩罚作用；如果其系数太大，导致对复杂性惩罚太大，所有的权值将会趋向零，很难找到优化的权值.我们提出一种在学习过程中采用模糊规则推理动态调整其系数的方法(以下称FRC方法)，从而更好对网络训练，并达到提高神经网络泛化能力的目的.

2.2 FRC方法

FRC方法的主要思想是，用式(4)的整体误差函数代替通常的误差平方和函数，并将正则项函数取如式(5)所示的形式：

$$\Omega = \sum_{i\in C}\frac{w_i^2}{1+w_i^2}. \quad (5)$$

其中C指所有连接权值的集合，而正则项系数则通过模糊规则的推理来调整.

下面对我们的方法做一解释，并进而推导出调整正则项系数的启发式知识.采取式(5)惩罚项以后，相当于给定了一个权值的先验分布，该先验概率为

$$p(w_i)\propto\exp(-\gamma\frac{w_i^2}{1+w_i^2}). \quad (6)$$

从式(6)可以看出，当$|w_i|$大时，Ω大，而$p(w_i)$小.所以选择这一先验分布，就表明我们期望权值是小的.正则项系数γ控制了权值的分布，γ越大，$p(w_i)$越尖陡，大权值的概率越小，也可压缩小的权值，使其变得更小，这样小的权值使网络映射更接近于线性变换，相应的复杂性变小，并进而减小方差，提高其泛化能力；γ越小，$p(w_i)$越平缓，大权值的概率越大，泛化能力变坏.所以通过加大γ可以使大权值的概率变小，从而优化到小的权值.但γ大到一定程度以后，可能会使大多数的权值变为零或接近零，这样又会使模型过于简单，而不能对数据样本进行很好的映射，从而使偏差变大.因此这时应适当的减小γ，以保证偏差和方差的折衷.

上述思路和做法与简单原则是一致的.在偏差满足要求的情况下，应该使模型越简单越好，即方差项越小越好.但这两个项通常是竞争的，所以可以通过调整正则项系数加以权衡折衷，最终达到最小化

偏差和期望的组合.

根据上面的解释和讨论,我们获取了以下启发式知识:(1) 当误差 E 继续减小时,说明偏差在减小,学习过程进行得很好,为了不导致大的方差量,可以通过增加正则项系数 γ,来减小大权值的概率,复杂性变小,使网络所逼近函数的曲面更光滑,并减小方差量,使整体误差变小.但增加量不能太大,否则又会使偏差变大,具体大小与误差变化的大小有关.(2) 当误差 E 小于期望的值时,说明偏差已经很小,但整体的误差 $\widetilde{E}$ 可能还较大,即方差量较大.为了减小方差量,应增加正则项系数 γ,原因及解释同(1).(3) 当以上两种情况同时成立时,也应增加正则项系数 γ,原因及解释同(1)、(2).(4) 梯度下降法使整体误差 $\widetilde{E}$ 减小,但偏差 E 可能会增大,为了进一步减小偏差,需考虑减小正则项系数 γ,使其权值不会继续被减小,进而使复杂性变大,能使网络映射与样本数据变得更吻合,从而减小偏差 E.这时,若 E 增幅较小,而且大于期望的值 σ 时,γ 减小量较小,否则会使方差变化太大;若 E 增幅较大,而且大于期望的值 σ 时,同样应减小正则项系数 γ,但减小量略大,具体大小与误差变化的大小有关.(5) 特殊情况下,所有权值都变为零,应去掉正则项.

表 1 是根据上述启发式知识转换得到的模糊规则表.表中 CE 表示误差的变化量,并且 $CE = E(n) - E(n-1)$,$E(n)$表示第 n 次的误差平方和,σ 是期望的值,"负小"、"零"、"正小"、"正大"是定义在论域 CE、$E(n) - \sigma$ 上的模糊集合,"负大"、"负小"、"零"、"正大"则是定义在正则项系数变化量($\Delta\gamma$)上的模糊集合.

对于每个模糊集合的隶属度函数的选取,为了在精度和计算量之间能较好的兼顾,我们将每个输入、输出变量分别取了 4 个模糊集合,并采用了三角形分布和半梯形分布的隶属度函数,表 2 是我们所得到的各隶属度函数的参数,其中MF1、MF2、MF3、MF4 表示每个变量的 4 个模糊集合所对应的隶属度函数.另外,MF1 取图 1(a)的形式,MF4 取图 1(b)的形式,其余的取图 1(c)的形式.得到了模糊规则和隶属度函数以后,我们就可以将模糊推理功能融入带动量项的 BP 算法中,即每训练一次后,根据误差等的变化量,通过模糊规则推理动态的调整正则项系数.以下具体说明神经网络的结构和算法.

表 1　正则项系数变化量 $\Delta\gamma$ 模糊规则表

Table 1　The fuzzy rules for the change of regularization coefficient, $\Delta\gamma$

$E(n)-\sigma$ \ CE	负小	零	正小	正大
负小	正大	正大	正大	正大
零	正大	零	负小	负大
正小	零	负小	负小	负大
正大	零	负小	负小	负大

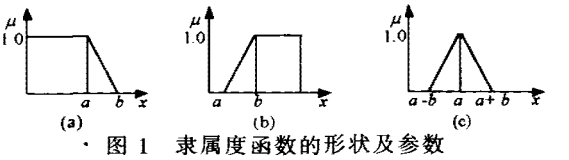

图 1　隶属度函数的形状及参数

Fig. 1　Shapes and parameters of the membership functions

我们采用的神经网络是带一个隐含层的三层前向神经网络,并且采取在线学习的方式对神经网络进行训练.完整的算法描述为:

(1) 初始化参数:期望的误差值 σ、正则项系数 $\gamma = 0.0$、学习率 $\alpha = 0.85$、动量项因子 $\beta = 0.9$;

(2) 给所有的连接权值(W)和阈值(θ)赋初值,一般取为小的随机数;

(3) 给定一样本,输入 x 和期望的输出 t;

(4) 计算实际输出 y,有

$$y_k = f\left(\sum_{j=0}^{M-1} w_{jk} f\left(\sum_{i=0}^{L-1} w_{ij} x_i - \theta_j\right) - \theta_k\right)$$

其中 $k = 0, 2, \ldots, N-1$,f 为 Sigmoid 函数;

(5) 根据整体的误差函数调整连接权值和阈值,整体误差函数为

$$\widetilde{E} = \frac{1}{2}\sum_{k=0}^{N-1}(y_k - t_k)^2 + \gamma\sum\frac{w_{ij}^2}{1 + w_{ij}^2} + \gamma\sum\frac{w_{jk}^2}{1 + w_{jk}^2}.$$

表 2　输入/输出变量隶属度函数参数表

Table 2　Parameters of membership functions for the input/output variables

		$E(n)-\sigma$	CE	$\Delta\gamma$
MF1	a	−0.001	−0.001	-4×10^{-6}
	b	0.01	0.00	-2×10^{-6}
MF2	a	0.05	0.00	-2×10^{-6}
	b	0.05	0.001	-2×10^{-6}
MF3	a	0.5	0.5	5×10^{-7}
	b	0.5	0.5	-2×10^{-6}
MF4	a	0.9	1.0	2×10^{-6}
	b	10.0	10.0	3×10^{-6}

权值和阈值的调整根据下式进行：

$$w_{jk}(t+1)=w_{jk}(t)+\alpha\cdot\Delta w_{jk}(t+1)+\beta\cdot\Delta w_{jk}(t)+\gamma\cdot 2\cdot w_{jk}(t)/(1+(w_{jk}(t))^2)^2,$$

$$w_{ij}(t+1)=w_{ij}(t)+\alpha\cdot\Delta w_{ij}(t+1)+\beta\cdot\Delta w_{ij}(t)+\gamma\cdot 2\cdot w_{ij}(t)/(1+(w_{ij}(t))^2)^2,$$

$$\theta_k(t+1)=\theta_k(t)+\alpha\delta_k(t+1),$$

$$\theta_j(t+1)=\theta_j(t)+\alpha\delta_j(t+1).$$

其中，

$$\Delta w_{ij}(t+1)=\delta_j(t+1)x_i,$$

$$\delta_k(t+1)=y_k(t)(1-y_k(t))(t_k-y_k(t)),$$

$$\Delta w_{jk}(t+1)=\delta_k h_j,$$

$$\delta_j(t+1)=h_j(t)(1-h_j(t))\sum_{k=0}^{N-1}\Delta w_{jk}(t+1)w_{jk}(t).$$

这里，$h_j(t)$指 t 时刻时，第 j 个隐节点的输出.

(6) 根据取大—取小重心模糊推理方法[7]求得正则项系数的改变量，调整正则项系数；

(7) 整体误差是否达到要求，若是，则停止，否则转(3)继续训练.

3 方法的实现及实例模拟

我们用 C 语言分别编制了 BP 算法、FRC 算法、以及固定正则项系数的算法，并用 586PC 机进行模拟实现.另外，基于前向神经网络主要应用于模式识别和函数逼近，为了验证我们提出的算法的性能，我们选取一些实例进行了实验.由于篇幅所限，在此仅选取几个有代表性的例子的实验结果作一说明.

3.1 实例 1:3 输入-单输出的非线性函数的逼近

以下 3 变量的非线性函数是研究者常用的，为了能较好的与现有的成果比较所获得的结果，我们

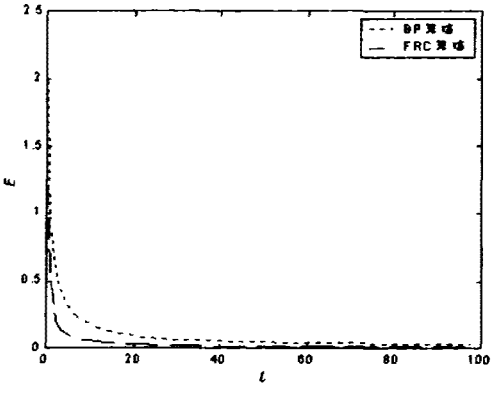

图 2 训练误差随训练次数变化的曲线

Fig.2 The training error as a function of the number of epochs

表 3 不同方法的训练与测试结果

Table 3 Training and testing results for different methods

	ERR1	ERR2
加动量项 BP 算法	0.029866	9.07791
$\gamma=2\times10^{-6}$	0.017929	6.6006
$\gamma=1.2\times10^{-5}$	0.089776	50.951855
FRC	0.0083	0.78409

也采用该函数，即

$$y=(1.0+\sqrt{x_1}+\frac{1}{x_2}+x_3^{-1.5})^2,$$

其中，x_1、x_2、x_3 的取值范围为[1,5].

我们根据其中 20 组训练数据[8~9]，利用加动量项的 BP 算法、固定正则项系数 γ 的方法(分别取 0.000002,0.000012)以及我们所提出的 FRC 方法分别来逼近这一非线性系统.采用不同的隐含层节点(3,4,6 个)进行了实验，并用另外 20 组数据[8~9]进行测试.但因篇幅所限，在此仅给出 6 个隐节点的神经网络训练 50000 次后的结果，归纳后的结果见表 3，其中 ERR1 指所有训练数据的误差平方和，ERR2 指所有测试数据的误差平方和.

图 2 和 3 给出了 FRC 方法与 BP 算法(加动量项)的学习与测试误差(E)随训练次数(t)变化的情况，其中，横坐标表示的是 100 到 50000 次的训练次数.

另外，为了说明我们的方法的正确性，我们还将所得到的结果与其它文献的结果进行了比较，比较结果见表 4.

文献[8]中的结果是通过对 3 个带 2 个隐层的 BP 网络进行学习训练后得到的，文献[9]中的结果是利用一改进的模糊神经网络在 586 微机上学习了

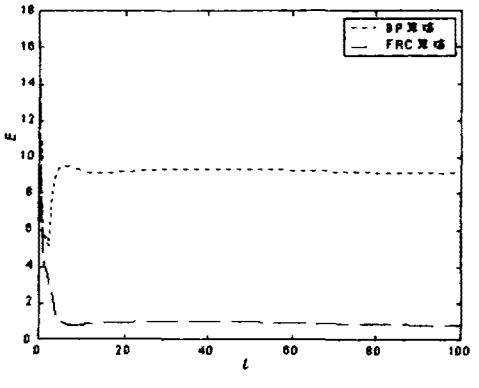

图 3 测试误差随训练次数变化曲线

Fig.3 The testing error as a function of the number of epochs

表 4　本文方法与其它文献中方法的训练与测试结果比较

Table 4　The Comparison of training and testing results between the present method and methods in other references

	ERR1	ERR2
文献[8]的方法	0.04973	28.12991
文献[9]的方法	0.047521	*
文献[10]的方法	*	19.2
FRC 方法	0.0083	0.78409

* 原文献中未给出结果

大约 3h 后得到的,文献[10]中的结果是用一种模糊神经网络的快速学习方法学习后得到的.通过比较可以看出,我们的方法比多个双隐层的 BP 网络、模糊神经网络的结果都好,这说明我们的方法是非常有效的.

总之,利用本文方法所得到的学习误差最小,而且测试样本的误差也最小,达到了很好的学习效果和泛化能力.

虽然本文方法取得了很好的泛化效果,但其所花的时间并不是很多,以下是各种方法是在 586/166PC 计算机上训练 50000 次所需的时间:BP 算法为 3min,56s,FRC 方法为 4min23s,固定正则项系数方法为 4min22s. 由此可以看出,当训练次数达到 50000 次时,我们的方法比 BP 算法仅仅多了 27s,而与正则项固定的方法基本相同.

3.2　实例 2:Iris 植物数据识别问题

著名的 Iris 数据库是人们广泛使用的用于模式分类的实例系统.它含有 150 个例子,分为 setosa、versicolor、virginica3 类,每类包含 50 个例子,并由 4 个实数特征值描述,分别表示萼片(sepal)长度,萼片宽度,花瓣(petal)长度,花瓣宽度.问题是根据这 4 个特性值分类 3 种 Iris 植物.

为了验证我们所提出的方法,并将我们的方法与 BP 算法、固定正则项系数的方法进行比较,我们取其中 4/5 的数据用于学习训练,其余的 1/5 用于测试网络的泛化性能.我们对不同隐节点数的网络分别进行了学习,当网络能够对所有的训练样本正确分类时,我们得到了表 5 和 6 的结果.

从表中的结果可以看出,当神经网络取不同的隐含层节点数时,我们的方法都得到了最好的结果.而固定正则项系数的方法和 BP 算法则不同,对于前者,不同的正则项系数,当隐含层节点数取不同的值时,得到了不同的效果;对于后者,隐含层节点数较少的情况较好.另外,我们将所得到的结果与文献[11]中采用一进化的模糊系统实现的结果进行了比较,其对所有的样本经过一系列的试凑和学习过程,最后还产生了 3 个错分类.我们所得到的结果明显优于文献[11]中的结果.

表 5　学习和测试结果(隐节点数 = 10)

Table 5　The learning and testing results with 10 hidden units

	学习次数	测试正确率
加动量项 BP 算法	15904	90%
$\gamma = 2 \times 10^{-6}$	16019	93.33%
$\gamma = 1.2 \times 10^{5}$	52153	100%
FRC 算法	10888	100%

表 6　学习和测试结果(隐节点数 = 6)

Table 6　The learning and testing results with 6 hidden units

	学习次数	测试正确率
加动量项 BP 算法	21471	96.67%
$\gamma = 2 \times 10^{6}$	17484	100%
$\gamma = 1.2 \times 10^{-5}$	15389	96.67%
FRC 算法	9493	100%

4　结论

本文提出了一种模糊动态调整正则项系数的神经网络学习方法,通过实验表明该方法明显优于传统的 BP 算法和固定正则项系数的方法.其具有最小的训练误差和最好的泛化能力.而且为了达到相同的误差,其所需的训练次数最少.另外,每一次所花费的时间来看,该方法与固定正则项系数的方法基本相同,而比传统的 BP 算法多出的时间也可以忽略不计.

本文所提出的方法和进行的实验是在隐含层节点确定的情况下进行的,但如本文引言中所述,确定结构的问题同样是泛化问题所应研究的问题.因此,对于神经网络,一种理想的学习方法是需同时考虑学习算法和结构的确定算法.一种可行的方法是将以上正则化方法与结构优化算法结合起来,我们将对这种方法进行进一步的研究,相信会取得更好的效果.

REFERENCES

[1]Moody J E. The effective number of parameters: an analy-

sis of generalization and regularization in nonlinear learning systems. In: *Advances in Neural Information Processing Systems* 4, San Mateo: Morgan Kaufmann, 1992, 847—854

[2] Bishop C M. *Neural Networks for Pattern Recognition*, New York: Oxford University Press: USA, 1995

[3] Saito K, Nakano R. Second-order learning algorithm with squared penalty term. *Neural Computation*, 2000, **12**: 709—729

[4] Aires F, Schmitt M, Chedin A, *et al*. The "weight smoothing" regulation of MLP for Jacobian stabilization. *IEEE Trans. on Neural Networks*, 1999, **10**(6): 1502—1510

[5] Ishikawa M. Structural learning with forgetting. *Neural Networks*, 1996, **9**(3): 509—521

[6] Weigend A S, Rumelhart D E, Huberman B A. Generalization by weight-elimination with application to forecasting. In: *Advances in Neural Information Processing Systems*, San Mateo: Morgan Kaufmann, 1991, 875—882

[7] WU Yan, SHI Hong-Bao. A method of dynamically tuning BP algorithm parameters according to fuzzy rules. *Computer Research and Development* (武妍, 施鸿宝. 一种模糊规则动态调整 BP 算法中参数的方法. **计算机研究与发展**) 1998, **35**(8): 689—693

[8] Takagi H, Hayashi I. NN-driven fuzzy reasoning. *Int. Journal of Approximate Reasoning*, 1991, **5**: 191—213

[9] WANG Shi-Tong. *Neural Fuzzy Systems and Applications*. Beijing: Publication of Beijing University of Aeronautics & Astronautics (王士同. **神经模糊系统及其应用**. 北京: 北京航空航天大学出版社), 1998

[10] LI Dong-Mei, LIU Jun-Qiang, HU Heng-Zhang. A fast algorithm for training a class of fuzzy neural networks. in: *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Hefei, China, 2000, Vol. **2**: 852—856

[11] Shi Y H, Eberhart R, Chen Y B. Implementation of evolutionary fuzzy systems. *IEEE Trans. on Fuzzy Systems*, 1999, **7**(2): 109—119