

文章编号:1001-9014(2005)01-0015-04

基于模糊化输入和反转提高神经网络分类性能的方法

武妍^{1,2}, 王守觉^{2,3}

(1. 同济大学 计算机科学与工程系, 上海 200092; 2. 同济大学 半导体与信息技术研究所, 上海 200092;
3. 中国科学院半导体研究所 神经网络实验室, 北京 100083)

摘要:为有效提高神经网络的分类性能,首先提出了一个可处理模糊输入的模糊神经网络结构,然后由模糊输出和非模糊目标输出定义了代价函数,推导出相应的学习算法,并对该模糊神经网络进行反转,提出了模糊化的反转算法.最后,通过计算机仿真实验的模式分类问题,验证了所提出的方法的有效性.实验结果表明,所提出的方法具有学习效率高、分类准确率高、泛化能力高的优点.

关键词:模糊神经网络;学习算法;模糊输入;反转;分类
中图分类号:TP183 **文献标识码:**A

METHOD FOR IMPROVING CLASSIFICATION PERFORMANCE OF NEURAL NETWORK BASED ON FUZZY INPUT AND NETWORK INVERSION

WU Yan^{1,2}, WANG Shou-Jue^{2,3}

(1. Department of Computer Science and Engineering, Tongji University, Shanghai 200092, China;
2. Institute of Semiconductors and Information Technology, Tongji University, Shanghai 200092, China;
3. Lab of Artificial Neural Networks, Institute of Semiconductors, CAS, Beijing 100083, China)

Abstract: In order to effectively improve the classification performance of neural network, first architecture of fuzzy neural network with fuzzy input was proposed. Next a cost function of fuzzy outputs and non-fuzzy targets was defined. Then a learning algorithm from the cost function for adjusting weights was derived. And then the fuzzy neural network was inverted and fuzzified inversion algorithm was proposed. Finally, computer simulations on real-world pattern classification problems examine the effectiveness of the proposed approach. The experiment results show that the proposed approach has the merits of high learning efficiency, high classification accuracy and high generalization capability.

Key words: fuzzy neural network; learning algorithm; fuzzy input; inversion; classification

引言

模糊神经网络是一种可以用来处理模糊数据的网络结构.多层前向神经网络可以通过扩展它的输入、连接权和(或)目标值到模糊数值来实现模糊化^[1-5].模糊化神经网络的一种很重要的应用是求解分类问题.本文主要是从神经网络的输入模糊化的角度更有效的解决模式分类问题.这主要基于两种情况的考虑:一是实际中,很多特性值是不能精确测量的,所以需要用一个区间的模糊数来进行描述;

二是对精确的值模糊化后,可以避免过训练的问题,提高泛化性能^[3].

训练好的神经网络可以被看作是从输入空间到输出空间的非线性映射,这一映射可以记作: $O = F(W; X)$.可以很清楚地看出,除了调整权值 W 这种通常的做法,还可以适当的调整输入 X ,使 O 接近理想的输出(记作 t).这又叫做神经网络的反转(inversion).反转的思想在通常的BP算法中有所研究^[6].本文将反转的思想推广到模糊化的神经网络中,这时, X 的调整是调整每个训练本的模糊数,即

收稿日期:2004-03-20,修回日期:2004-09-13

基金项目:国家自然科学基金资助项目(60135010)

作者简介:武妍(1967-),女,山西晋中人,同济大学计算机系副教授,博士,主要从事神经网络、模式识别研究.

Received date: 2004-03-20, revised date: 2004-09-13

模糊化的程度,并据此来确定一个合适的模糊宽度,从而进行更有效的学习。

本文提出了一个可以处理模糊输入的多输入、多输出的神经网络结构,并提出了模糊化的反转算法.最后,通过计算机仿真实际的模式分类问题,讨论和验证了所提出的方法的有效性。

1 模糊神经网络结构

在定义神经网络结构之前,首先简单的描述一下有关模糊数和区间的一些基本操作:

$$X + Y = [x^L, x^U] + [y^L, y^U] = [x^L + y^L, x^U + y^U]$$

$$k \cdot X = k \cdot [x^L, x^U] = \begin{cases} [kx^L, kx^U] & \text{如果 } k \geq 0 \\ [kx^U, kx^L] & \text{如果 } k < 0 \end{cases}$$

其中, $X = [x^L, x^U]$ 和 $Y = [y^L, y^U]$ 是区间值, k 是一个实数值. 上标 L 和 U 分别表示上限和下限. 在下文中,用大写字母表示模糊数和区间值,用小写字母表示实数值.

另外,通常的神经网络的激活函数可以被扩展到区间的输入-输出关系:

$$f(\text{Net}) = f([net^L, net^U]) = [f(net^L), f(net^U)]$$

其中, f 是一个 S 型函数:

$$f(x) = 1 / \{1 + \exp(-x)\}.$$

另外,模糊数上的操作是通过扩张原理来定义的:

$$\mu_{A+B} = \max \{ \mu_A(x) \wedge \mu_B(y) : z = x + y \}$$

$$\mu_{kA}(z) = \max \{ \mu_A(x) : z = kx \}$$

$$\mu_{Net}(z) = \max \{ \mu_{Net}(x) : z = f(x) \}$$

下面,使用以上定义在模糊数上的操作来构建能处理模糊输入的具有 n_i 个输入单元、 n_h 个隐单元、 n_o 个输出单元的三层前向神经网络结构. 其中,各层模糊神经元的操作如下:

$$\text{输入单元: } O_{pi} = X_{pi}, i = 1, 2, \dots, n_i, \quad (1)$$

$$\text{隐单元: } O_{pj} = f(\text{Net}_{pj}), j = 1, 2, \dots, n_h, \quad (2)$$

$$\text{Net}_{pj} = \sum_{i=1}^{n_i} O_{pi} \cdot w_{ji} + \theta_j, \quad (3)$$

$$\text{输出单元: } O_{pk} = f(\text{Net}_{pk}), k = 1, 2, \dots, n_o, \quad (4)$$

$$\text{Net}_{pk} = \sum_{j=1}^{n_h} O_{pj} \cdot w_{kj} + \theta_k, \quad (5)$$

可以看出,这种神经网络是传统 3 层前向神经网络的扩展. 在这里, X_{pi} , Net_{pj} , Net_{pk} , O_{pi} , O_{pj} , O_{pk} 是模糊数, $p = 1, 2, \dots, m$ (m 是样本数).

在式(1)~式(5)中,计算每个单元的输入-输出关系涉及到模糊数上的模糊算术运算,例如,图 1

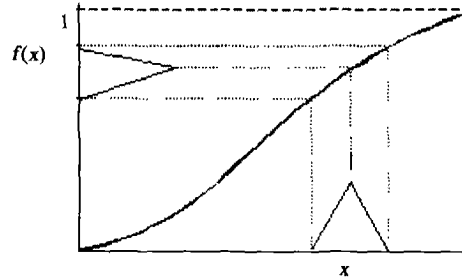


图 1 模糊化神经网络的激活函数

Fig. 1 Fuzzification of the activation function of neural network

所示的是模糊数通过激活函数 $f(\cdot)$ 的非线性映射.

在本文中,式(1)~式(5)的计算是通过输入值 X_{pi} 进行区间算术运算而完成的. 其中, $O_{pi} = [o_{pi}^L, o_{pi}^U] = X_{pi} = [x_{pi}^L, x_{pi}^U]$

$$O_{pj} = [o_{pj}^L, o_{pj}^U] = [f(\text{net}_{pj}^L), f(\text{net}_{pj}^U)]$$

$$O_{pk} = [o_{pk}^L, o_{pk}^U] = [f(\text{net}_{pk}^L), f(\text{net}_{pk}^U)]$$

$$\text{net}_{pj}^L = \sum_{i=1}^{n_i} w_{ji} o_{pi}^L + \sum_{i=1}^{n_i} w_{ji} o_{pi}^U + \theta_j$$

$$\text{net}_{pj}^U = \sum_{i=1}^{n_i} w_{ji} o_{pi}^L + \sum_{i=1}^{n_i} w_{ji} o_{pi}^L + \theta_j$$

$$\text{net}_{pk}^L = \sum_{j=1}^{n_h} w_{kj} o_{pj}^L + \sum_{j=1}^{n_h} w_{kj} o_{pj}^L + \theta_k$$

$$\text{net}_{pk}^U = \sum_{j=1}^{n_h} w_{kj} o_{pj}^U + \sum_{j=1}^{n_h} w_{kj} o_{pj}^L + \theta_k$$

2 模糊神经网络学习

学习的目的是通过调整权值 w_{ji} 和 w_{kj} , 阈值 θ_j 和 θ_k 以及输入模糊化的程度: x_{pi}^L 和 x_{pi}^U , 最小化目标输出和神经网络实际输出的误差.

2.1 权值的调节算法

首先,将 m 个训练模式, $\mathbf{x}_p = (x_{p1}, x_{p2}, \dots, x_{pn_i})$, $p = 1, 2, \dots, m$ 中的每一个训练模式进行初始模糊化,一种简单的方法是给每个特性值增加一个扩展的宽度: a_{train} , 使其变成一个对称三角形模糊数(如图 2 所示).

将模糊化训练模式记作:

$$X_p = (X_{p1}, X_{p2}, \dots, X_{pn_i}).$$

根据模糊输入 X_p , 定义目标输出

$$t_p = (t_{p1}, t_{p2}, \dots, t_{pn_o}):$$

$$t_{pk} = \begin{cases} 1 & \text{如果 } X_p \text{ 是第 } k \text{ 类} \\ 0 & \text{否则} \end{cases} \quad k = 1, 2, \dots, n_o.$$

然后,定义要被最小化的代价函数:

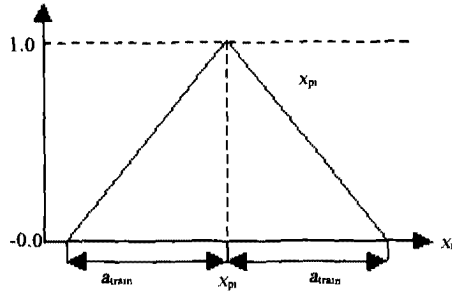


图2 模糊化输入特性值

Fig. 2 Fuzzification of each attribute value

$$e_p = \sum_{k=1}^{n_o} \left\{ \frac{(t_{pk} - o_{pk}^L)^2}{2} + \frac{(t_{pk} - o_{pk}^U)^2}{2} \right\}, \quad (6)$$

根据式(6)定义的代价函数,通过梯度下降方法推导出调节权值、阈值的学习算法.因为篇幅所限,在此省略了相应的算式.

2.2 反转算法

除了对权值进行学习,还可以对网络进行反转,根据神经网络输出矢量的误差修改输入矢量,通过调整权值和输入矢量值的双重作用,来最小化代价函数.

x_{pi}^L 和 x_{pi}^U 的调节算式如下:

$$x_{pi}^U(t+1) = x_{pi}^U(t) - \eta \frac{\partial e_p}{\partial x_{pi}^U} + \alpha \Delta x_{pi}^U(t)$$

$$x_{pi}^L(t+1) = x_{pi}^L(t) - \eta \frac{\partial e_p}{\partial x_{pi}^L} + \alpha \Delta x_{pi}^L(t)$$

其中 η 称为步长, α 称为动量项因子.

下面分四种情况给出 $\frac{\partial e_p}{\partial x_{pi}^U}$ 和 $\frac{\partial e_p}{\partial x_{pi}^L}$ 的值.

(1) 如果 $w_{kj} \geq 0$ 而且 $w_{ji} \geq 0$

$$\frac{\partial e_p}{\partial x_{pi}^U} = - \sum_{k=1}^{n_o} \delta_{pk}^U \sum_{j=1}^{n_h} w_{kj} o_{pj}^U (1 - o_{pj}^U) w_{ji}$$

$$\frac{\partial e_p}{\partial x_{pi}^L} = - \sum_{k=1}^{n_o} \delta_{pk}^L \sum_{j=1}^{n_h} w_{kj} o_{pj}^L (1 - o_{pj}^L) w_{ji}$$

(2) 如果 $w_{kj} \geq 0$ 而且 $w_{ji} < 0$

$$\frac{\partial e_p}{\partial x_{pi}^U} = - \sum_{k=1}^{n_o} \delta_{pk}^U \sum_{j=1}^{n_h} w_{kj} o_{pj}^L (1 - o_{pj}^L) w_{ji}$$

$$\frac{\partial e_p}{\partial x_{pi}^L} = - \sum_{k=1}^{n_o} \delta_{pk}^U \sum_{j=1}^{n_h} w_{kj} o_{pj}^U (1 - o_{pj}^U) w_{ji}$$

(3) 如果 $w_{kj} < 0$ 而且 $w_{ji} \geq 0$

$$\frac{\partial e_p}{\partial x_{pi}^U} = - \sum_{k=1}^{n_o} \delta_{pk}^L \sum_{j=1}^{n_h} w_{kj} o_{pj}^U (1 - o_{pj}^U) w_{ji}$$

$$\frac{\partial e_p}{\partial x_{pi}^L} = - \sum_{k=1}^{n_o} \delta_{pk}^U \sum_{j=1}^{n_h} w_{kj} o_{pj}^L (1 - o_{pj}^L) w_{ji}$$

(4) 如果 $w_{kj} < 0$ 而且 $w_{ji} < 0$

$$\frac{\partial e_p}{\partial x_{pi}^U} = - \sum_{k=1}^{n_o} \delta_{pk}^U \sum_{j=1}^{n_h} w_{kj} o_{pj}^L (1 - o_{pj}^L) w_{ji}$$

$$\frac{\partial e_p}{\partial x_{pi}^L} = - \sum_{k=1}^{n_o} \delta_{pk}^L \sum_{j=1}^{n_h} w_{kj} o_{pj}^U (1 - o_{pj}^U) w_{ji}$$

其中,

$$\delta_{pk}^L = (t_{pk} - o_{pk}^L) o_{pk}^L (1 - o_{pk}^L)$$

$$\delta_{pk}^U = (t_{pk} - o_{pk}^U) o_{pk}^U (1 - o_{pk}^U)$$

3 模糊分类规则

通常对一个新模式的分类是采用赢者胜出的规则.令 $\mathbf{o}_p = (o_{p1}, \dots, o_{pc})$ 是对于 n 维非模糊输入矢量 $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$ 的 c 维输出矢量,当下列关系满足时,输入模式 \mathbf{x}_p 被分为 l 类:

$$o_{pk} < o_{pl} \quad k=1, 2, \dots, c \quad (k \neq l), \quad (7)$$

在本文中,为了与通常的 BP 网络的结果进行比较,将测试模式的模糊化的宽度设定为 0,这样式(7)所描述的规则也可以适用于模糊输入的神经网络中.另外,专门针对模糊输入矢量 \mathbf{X}_p ,以上分类规则不能直接应用到对其的分类中,必须重新定义模糊数的不等式关系 $O_{pk} < O_{pl}$:

$$O_{pk} < O_{pl} \Leftrightarrow O_{pk}^U < O_{pl}^L$$

这样,可以得到对于模糊输入 \mathbf{X}_p 的分类规则:

$$O_{pk}^U < O_{pl}^L \quad k=1, 2, \dots, c \quad (k \neq l), \quad (8)$$

如果没有一类满足式(8),则该输入模式被拒识.

4 实验结果及分析

为了验证提出的模糊输入加反转的方法(记作 I-FIBP)及学习算法的性能,选取一些实际的多分类的模式分类问题进行了实验,并与一种改进的加动量项 BP 算法(记作 MBP)、单纯的模糊化输入的方法(记作 FIBP)做一些比较.在此就几个有代表性的例子的实验结果做一说明.

4.1 葡萄酒数据分类问题

UCI 数据库的葡萄酒数据分类问题共包含 178 个样本,分为 3 类.每个样本由 13 个实数特征值描述.该问题是根据这 13 个特性值分类 3 种酒,这是一个典型的 3 分类问题.使用一个具有 13 个输入单元、10 个隐单元和 3 个输出单元的 3 层前向神经网络来求解这一问题.另外,根据经验,初始的模糊化的宽度(a_{train})设定为 0.04.

为了估计每种方法的分类性能和泛化能力,采用了 10-交叉验证的技术.也就是说训练集由 160 个样本构成,测试集由 18 个样本构成.仿真结果如表

1 所示.

表 1 葡萄酒数据的仿真结果

Table 1 Simulation results on wine data

	错误分类率(%)		
	50 次	100 次	150 次
I-FIBP	2.1	2.0	1.8
FIBP	4.3	3.2	2.6
MBP	4.1	4.4	4.7

表 1 所示的是不同训练次数时,3 种不同方法各取一个最优的学习率、动量因子等参数,对测试集的平均错误分类率.从中可以看出,I-FIBP 方法具有最好的测试效果,错误分类率最低,即泛化能力(在此是指对测试样本的正确分类性能)最高.当训练次数为 50 次时,I-FIBP 的泛化能力基本上比 FIBP 提高了一倍.并且 I-FIBP 仅花很少的训练次数就可以达到较高的泛化能力,明显的优于另外两种方法.另外,I-FIBP 的泛化能力也好于文[3]中所报道的使用多个 α -截集的结果,而且比它简单.

显然,训练一次,I-FIBP 比 MBP 算法的计算量大,所以仅从训练次数比较,还不能说明 I-FIBP 方法的效率高.所以,还对训练时间进行了比较.表 2 所示的是当训练误差取 0.01 时, I-FIBP 方法和 MBP 算法的学习效率比较.可见,要达到相同的训练误差,I-FIBP 比 MBP 算法需要少得多的训练时间,所以训练速度最快,效率高.

表 2 学习效率比较

Table 1 Comparison of training efficiency

算法	训练次数	CPU 时间(时间)
MBP	7135	916(约 52 秒)
I-FIBP	116	76(约 4 秒)

4.2 Iris 植物数据识别问题

iris 数据库是著名的、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 3 类.每类包含 50 个例子,并由四个实数特征值描述.该问题是根据这四个特性值分类 3 种 iris 植物,这也是一个典型的 3 分类问题.

本文使用一个具有 4 个输入单元、10 个隐单元和 3 个输出单元的 3 层的前向神经网络来求解这一分类问题.根据经验,初始的模糊化的宽度(a_{train})设定为 0.03.并将 150 个样本都用作训练样本,主要来考察不同方法的学习效果.图 3 所示的是不同训练次数(t)时,对训练样本的错误分类率(ER).从图

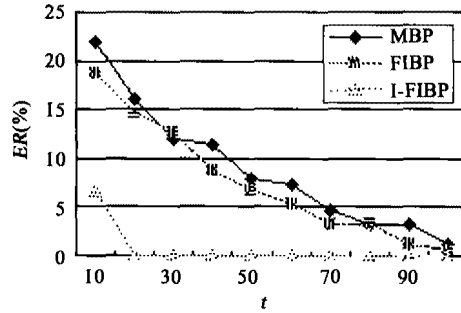


图 3 IRIS 植物分类问题的仿真结果

Fig. 3 Simulation results on iris plant classification problem

3 的结果可以看出,I-FIBP 方法训练速度最快,实际上在训练次数不到 20 次时,分类的正确率就达到了 100%.对于该问题,MBP 和 FIBP 的效果差别不大,只是 FIBP 稍好一些.

5 结语

本文提出了一种通过模糊化输入和反转提高神经网络分类性能的新方法.并通过计算机仿真实验的模式分类问题,将其实验结果与单纯的模糊化输入神经网络、带动量项的 BP 网络的结果进行了比较.我们发现通过模糊化训练模式的输入值可以起到提高泛化能力的作用.另外,本文的方法具有学习效率高,分类准确率高的优点,是一种很有效的学习和分类方法.

REFERENCES

- [1] Ishibuchi H, Kwon K, Tanaka H, A learning algorithm of fuzzy neural networks with triangular fuzzy weights [J], *Fuzzy Sets and Systems*, 1995, **71**: 277—293.
- [2] Hayashi Y, Fuzzy neural network with fuzzy signals and weights [J], *International Journal of Intelligent Systems*, 1993, **8**: 527—537.
- [3] Ishibuchi H, Nii M, Fuzzification of input vectors for improving the generalization ability of neural networks [C], In: *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska, May 4-9, 1998, **2**: 1153—1158.
- [4] LI Zhen-Quan, Kecman V, Ichikawa A, Fuzzified neural network based on fuzzy number operation [J], *Fuzzy Sets and Systems*, 2002, **130**:291—304.
- [5] Ishibuchi H, Nii M, Numerical analysis of the learning of fuzzified neural networks from fuzzy if-then rules [J], *Fuzzy Sets and Systems*, 2001, **120**:281—307.
- [6] WU Yan, WANG Shou-Jue, A new algorithm to improve the learning performance of neural network through result-feedback [J], *Computer Research and Development*, (武妍,王守觉,一种通过反馈提高神经网络学习性能的新算法,计算机研究与发展), 2004, **41**(19):1488—1492.