

模糊系统与前向神经网络的结合

On the Combination of Fuzzy Systems with Feed-forward Neural Networks

潘维民

沈理

(北京邮电大学计算机学院 100876) (中国科学院计算技术研究所 100080)

Abstract Fuzzy system and neural networks have a lot of similar properties in many prospects. And in some other prospects, fuzzy system and neural networks supplement each other. In this paper, the combination of fuzzy system and neural networks is surveyed in detail.

Keywords Fuzzy system, Feed-forward neural networks, Neuro-fuzzy system

1 引言

模糊系统方法和神经网络技术是近年来计算智能领域研究热点,被广泛地应用于复杂系统、非确定性等难于建立比较准确的数学模型的问题,并在自动控制、计算机图像处理、语音识别、手写体识别等领域有重要应用。模糊系统与神经网络的结合也越来越受到人们的重视。模糊系统和神经网络的结合可以分为模糊系统与前向网络的结合和与反馈网络的结合两类。模糊系统与反馈网络的结合主要有模糊联想记忆^[1]、模糊共振网络(Fuzzy ART)^[2]和模糊自适应特征映射(Fuzzy SOM)等。本章主要综述模糊系统和前向网络的结合。

模糊系统和神经网络的结合主要有两类:模糊神经网络(Fuzzy neural networks)和神经模糊系统(neuro-fuzzy systems)。模糊神经网络又可以分为模糊算子神经网络和模糊信息神经网络。

神经模糊系统将神经网络作为实现模糊系统模型的一种工具,即在神经网络的框架下实现模糊系统或模糊系统的一部分功能。神经模糊系统虽具有一些自己所具有的而神经网络不具备的特性,但它没有跳出神经网络的框架。神经模糊系统从结构上来看,一般是四层或五层的前向神经网络。

模糊神经网络是神经网络的模糊化。模糊神经网络对神经网络的模糊化主要体现在两个方面:神经算子和神经网络的数据流。因而模糊神经网络可以分为两大类:一类将神经算子改为模糊逻辑的T范和T余范算子或其他的模糊逻辑算子,其输入和权重仍为实数,在此我们称之为模糊算子神经网络;另一类模糊神经网络的输入和权重为模糊集,其运算是以 α -截集为基础,在此我们称之为模糊信息网络。模糊神经网络的

结构多种多样,可以是典型的神经网络结构,也可以是为实现特殊功能而采取特殊的结构。

神经模糊系统和模糊神经网络的初始化与神经网络不同,它不采用随机初始化的方法,而是由数据或专家的知识来获得规则,并把其作为神经模糊系统初始化的基础,从而将先验知识引入神经网络。

神经模糊系统的计算一般为简单的算术算子,以实现模糊系统或组合规则推理为目的;而模糊神经网络以模糊信息的处理为目的,其计算是以T范、T余范或在 α -截集、扩张原理的基础上的模糊算术为主。

2 模糊系统与神经网络的等价

2.1 函数逼近

模糊系统与神经网络除了都是无模型系统外,它们都是函数的全局逼近器。模糊系统以其插值机理来逼近任意的连续函数^[3]。不但传统的模糊系统模型是任意连续函数的全局逼近器,而且神经网络与模糊系统的不同结合能逼近不同的函数,如模糊神经网络可以逼近模糊函数^[4]。神经网络也是任意连续函数的全局逼近器^[6,7,14]。设任意连续函数 $h(x)$,对于紧空间 X 和任意小的正数,总能找到一个三层的前向神经网络 $N(x)$ 满足:

$$\int_x \|N(x) - h(x)\| dx < \epsilon$$

在前向神经网络家族中,RBF神经网络是最优的函数逼近器^[5],即对于任意的神经网络 $N(x)$ 总存在一个RBF神经网络 $N'(x)$,满足:

$$\int_x \|N'(x) - h(x)\| dx < \int_x \|N(x) - h(x)\| dx$$

2.2 神经网络与模糊系统的等价性

模糊系统和神经网络的等价性主要有两个方面:

TS 模型的等价性和 Madani 模型的等价性。对于 TS 模型,首先 Jang^[15]给出了标准的 Gaussian RBF 神经网络等价于限制的 TS-型模糊系统。Hunt^[16]指出推广的 Gaussian RBF 神经网络等价于 TS-型模糊系统。Benitez^[3]证明了若一个三层的神经网络,隐含单元的激发函数为对数函数(logistic),输出层的激发函数为单元函数,设 $N(x)$,则存在一个模糊系统的输出也为 $N(x)$ 。

对于 Madani 模型,从由模糊系统而演变来的模糊基函数(Fuzzy Basis Function, FBF)函数出发, Kim 指出 FBF 为 RBF 函数的组合,并且指出 GRNN(General Regress Neural Networks)为 FBF 神经网络的特例^[12]。

神经网络具有强大的学习功能,然而其通过学习所得的输出结果难于解释,神经网络通常被看作一个黑匣子。为此,人们煞费苦心地在研究神经网络的解释^[6]。

模糊系统是由 if...then 规则构成的,因而易于解释。但是模糊系统的学习能力差,很难适应动态复杂系统的建模。由模糊系统和神经网络技术相结合而产生的模糊神经网络和神经模糊系统等综合了模糊系统和神经网络的优点,成为易于解释的神经网络系统或具有强大学习能力和自适应能力的模糊系统。模糊系统与神经网络的结合把先验知识以模糊规则的形式引入神经网络,从而使先验知识得以利用。

3 模糊算子神经网络

模糊算子神经网络将模糊逻辑算子用于神经网络中,从而使神经网络具有处理语言值变量的能力。首先我们给出模糊算子神经网络的突触算子和膜算子的一般性描述。设模糊算子神经网络的两个节点 i 和 j 的输出分别为 x_i 和 x_j ,其间的权重为 w_{ij} 。

3.1 模糊算子神经网络的计算:

突触算子:

$$u_j = w_{ij} \otimes x_i,$$

其中 $\otimes$ 是模糊算子。

膜算子:

$$x_j = f(\bigoplus u_j - \theta)$$

其中 $\bigoplus$ 为模糊和算子, θ 为阈值, f 为模糊转移函数。

常见的模糊逻辑算子:

$$1. \otimes = \min, \bigoplus = \max^{[9,10,13]};$$

$$2. \otimes = *, \bigoplus = \max^{[11]};$$

$$3. x_i \otimes w_{ij} = x_i * w_{ij}, \bigoplus u_j = \sum_{j=1}^n u_j / n^{[12]}$$

模糊算子神经网络的输入是模糊集,输出为实数集,它的激发函数可以为任意的连续函数。其中模糊集都是以模糊向量来表示的。设模糊集 $\sum_{i=1}^n u_i/x_i$,则与其等价的模糊向量为 $(u_1, u_2, \dots, u_n)$ 。由此可见模糊算子神经网络实现映射:

$$f: [0,1]^n \rightarrow R$$

模糊算子神经网络是全局逼近器^[12]。对于任意的连续函数 $f: [0,1] \rightarrow R$ 存在一个模糊算子神经网络与其等价。

3.2 模糊算子神经网络的学习

模糊算子神经网络的学习一般是梯度下降法^[17,18]。max 和 min 型的模糊算子神经网络的应用很广泛^[19~21]。文^[18]给出了含有 max 和 min 算子的求导方法和反传算法并证明其收敛性质。设模糊算子神经网络有 $N+1$ 层, n 代表第 n 层, w_{nj} 为第 n 层节点 j 与第 $n+1$ 层节点 i 之间的权重, I_{nj} 和 O_{nj} 分别为第 n 层节点 j 的输入和输出, T_k 和 O_k 分别为输出层的第 k 个节点的理想输出和神经网络输出。

$$I_{nj} = \max_j \{ \min \{ w_{nj-1}, O_{j-1} \} \}$$

$$O_{nj} = f(I_{nj})$$

其中 f 为激发函数。误差函数为:

$$E = \frac{1}{2} \sum_{k=1}^s \sum_{j=1}^n (T_k - O_k)^2$$

令: $\delta_{(N+1)k} = (T_k - O_k) f'(I_{(N+1)k})$

$$\delta_{nj} = f'(I_{nj}) \sum_{i=1}^s \{ \delta_{(n+1)i} \log(w_{nj} \wedge O_{nj} - \bigvee_{j=1}^n (w_{nj} \wedge O_{nj})) \log(w_{nj} - O_{nj}) \}$$

则神经网络的权重修改为:

$$\Delta w_{nj} = -\eta \frac{\partial E}{\partial w_{nj}} = \eta \delta_{(n+1)i} \log \{ w_{nj} \wedge O_{nj} - \bigvee_{j=1}^n (w_{nj} \wedge O_{nj}) \} \log(O_{nj} - w_{nj})$$

$$\text{其中 } \log(x) = \begin{cases} 1 & x > 0 \\ \frac{1}{2} & x = 0 \\ 0 & x < 0 \end{cases}$$

4 模糊信息神经网络

模糊信息神经网络的主要目的是用神经网络结构来处理模糊信息。模糊信息神经网络的结构大多数与神经网络的结构相同,但不同的是^[23],增加了一些模糊逻辑算子层。模糊信息神经网络的输入、输出和权重都为模糊数,其中主要为 L-R 型模糊数。常见的包括两种:三角形模糊数和梯形模糊数。由于模糊集可以分解为 α -水平截集的并,所以模糊信息神经网络的运算为 α -水平截集的运算。模糊信息神经网络主要有以下特点:

1) 神经网络上的数据流为用 α -水平截集表示的

模糊集;

2) 神经算子为 α -水平截集的运算, 有时有模糊逻辑算子;

3) 权重为三角形模糊数或梯形模糊数;

4) 激发函数是以扩张原理为基础的任意连续函数。

4.1 模糊信息网络的计算

设模糊信息神经网络的结构如图1所示。

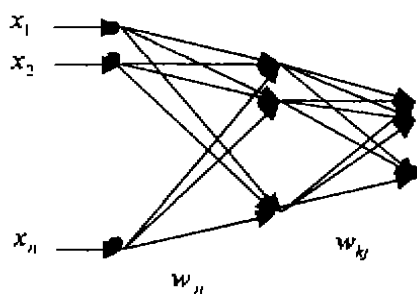


图1 模糊信息神经网络的结构

对于 α -截集, 模糊信息神经网络的输入分别为 $[x_1^{\alpha 1}, x_1^{\alpha 2}], [x_2^{\alpha 1}, x_2^{\alpha 2}], \dots, [x_n^{\alpha 1}, x_n^{\alpha 2}]$ 。隐含层节点 j 和输入层节点 i 之间的权重和节点 j 的阈值分别为 $w_{ji}^* = [w_{ji}^{\alpha 1}, w_{ji}^{\alpha 2}]$ 和 $\theta_j^* = [\theta_j^{\alpha 1}, \theta_j^{\alpha 2}]$, 节点 j 的激发函数为 f_j 。则隐含层节点 j 的输出为:

$$O_j^* = [O_j^{\alpha 1}, O_j^{\alpha 2}] = f_j \left(\sum_{i=1}^n w_{ji}^* * x_i^* - \theta_j^* \right) = f_j \left(\sum_{i=1}^n \min \{ x_i^{\alpha 1} * w_{ji}^{\alpha 1}, x_i^{\alpha 2} * w_{ji}^{\alpha 2} \} - \theta_j^{\alpha 1} \right), f_j \left(\sum_{i=1}^n \max \{ x_i^{\alpha 2} * w_{ji}^{\alpha 1}, x_i^{\alpha 1} * w_{ji}^{\alpha 2} \} - \theta_j^{\alpha 2} \right)$$

设输出层节点 k 和隐含层节点 j 之间的权重和节点 k 的阈值分别为 $w_{kj}^* = [w_{kj}^{\alpha 1}, w_{kj}^{\alpha 2}]$ 和 $\theta_k^* = [\theta_k^{\alpha 1}, \theta_k^{\alpha 2}]$, 则节点 k 的输出为:

$$y_k^* = [y_k^{\alpha 1}, y_k^{\alpha 2}] = \left(\sum_{j=1}^m w_{kj}^* * y_j^* - \theta_k^* \right) = \left[\sum_{j=1}^m \min \{ y_j^{\alpha 1} * w_{kj}^{\alpha 1}, y_j^{\alpha 2} * w_{kj}^{\alpha 2} \} - \theta_k^{\alpha 1}, \sum_{j=1}^m \max \{ y_j^{\alpha 2} * w_{kj}^{\alpha 1}, y_j^{\alpha 1} * w_{kj}^{\alpha 2} \} - \theta_k^{\alpha 2} \right]$$

其中 n 为输入变量的个数, m 为隐含单元的个数。设 l 为输出空间的维数。由上可见当模糊集被分解为 b 个 α -截集时, 模糊信息神经网络对于一个样本需要 $2b$ 次运算才能完成。

4.2 模糊信息网络的学习

模糊信息神经网络的学习算法主要有反传算法、遗传算法和快速反传算法等。其中反传算法和快速反传算法^[22~26]最为常用。模糊信息神经网络经过反传算法学习后, 有时权重的下界超过其上界, 因而不是模糊

集。由于这个原因许多模糊信息神经网络对其学习算法增加了一些限制, 遗传算法可以用于模糊信息神经网络的学习过程^[27], 但其收敛过程比较慢。

设 $d_{ks}^* = [d_{ks}^{\alpha 1}, d_{ks}^{\alpha 2}]$ 和 $y_{ks}^* = [y_{ks}^{\alpha 1}, y_{ks}^{\alpha 2}]$ 分别为样本 s 的理想输出和神经网络输出, 则误差函数为:

$$E^* = \sum_{s=1}^S [(d_{ks}^{\alpha 1} - y_{ks}^{\alpha 1})^2 + (d_{ks}^{\alpha 2} - y_{ks}^{\alpha 2})^2]$$

$$E = \sum_{s=1}^S \alpha * E^*$$

设权重和阈值都为三角形模糊数, $w_{ki} = \sum_{\alpha} \alpha * [w_{ki}^{\alpha 1}, w_{ki}^{\alpha 2}] = (w_{ki}^1, w_{ki}^*, w_{ki}^2)$, $\theta_k = (\theta_k^1, \theta_k^*, \theta_k^2)$, 其中 $w_{ki}^* = \frac{w_{ki}^1 + w_{ki}^2}{2}$, $\theta_k^* = \frac{\theta_k^1 + \theta_k^2}{2}$, 则:

$$\frac{\partial E^*}{\partial w_{ki}^1} = \frac{\partial E^*}{\partial w_{ki}^*} \left(1 - \frac{\alpha}{2} \right) + \frac{\partial E^*}{\partial w_{ki}^2} \frac{\alpha}{2}$$

$$\frac{\partial E^*}{\partial w_{ki}^2} = \frac{\partial E^*}{\partial w_{ki}^*} \left(1 - \frac{\alpha}{2} \right) + \frac{\partial E^*}{\partial w_{ki}^1} \frac{\alpha}{2}$$

$$\Delta w_{ki}^1 = -\eta \sum_s \frac{\partial E^*}{\partial w_{ki}^1}$$

$$\Delta w_{ki}^2 = -\eta \sum_s \frac{\partial E^*}{\partial w_{ki}^2}$$

在 $t+1$ 时刻令:

$$u_{ki}^1(t+1) = w_{ki}^1(t) + \Delta w_{ki}^1(t)$$

$$u_{ki}^2(t+1) = w_{ki}^2(t) + \Delta w_{ki}^2(t)$$

则 $t+1$ 时刻权重为:

$$w_{ki}^1(t+1) = \min \{ u_{ki}^1(t+1), u_{ki}^2(t+1) \}$$

$$w_{ki}^2(t+1) = \max \{ u_{ki}^1(t+1), u_{ki}^2(t+1) \}$$

对于阈值和其它节点权重改变可依此类推。当神经网络上的数据为梯形模糊数时其计算相对复杂一些^[28], 但其本质基本相同。

模糊信息神经网络实现一个模糊空间到另一个模糊空间的映射:

$$f: F^n \rightarrow F^m$$

其中 F^n 是 n 维的模糊空间, F^m 是 m 维的模糊空间, f 为模糊信息神经网络所表示的映射。模糊信息神经网络与模糊算子神经网络和神经模糊系统相比, 其缺点是计算量大。另外, 模糊信息神经网络学习时的限制使学习算法的收敛速度减慢。

5 神经模糊系统

神经模糊系统的应用较为广泛, 被应用于自动控制和模式识别等领域。神经模糊系统主要用来实现模糊系统的整体或模糊系统的推理机。当其实现推理机时其功能相当于映射:

$$f: [0, 1]^n \rightarrow [0, 1]^m$$

当实现模糊系统时其功能相当于映射:

$$f: R^n \rightarrow R^m$$

并且为输入空间 R^n 到输出空间 R^m 的任意连续函数

的全局逼近器。

5.1 实现模糊推理机的神经模糊系统

这种神经模糊系统的结构如图2所示。

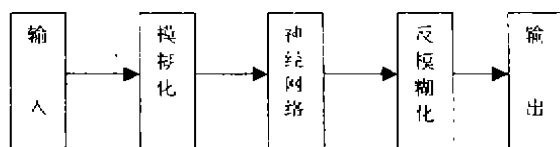


图2 模糊系统模型

这种神经模糊系统的模糊化和反模糊化是用模糊系统理论中的方法来实现的。神经网络主要承担规则存储和推理的任务。其中神经网络通常采用传统的神经网络结构和适当的学习算法。权重为任意的实数。

5.2 实现模糊系统的神经模糊系统的结构及运算

当神经模糊系统用于实现整个模糊系统时通常采用四层或五层的前向网络结构。主要有基于神经网络的模糊控制系统(NN-FLSC)^[25,26]、基于自适应网络的模糊推理系统(AFIS)^[23,26]、基于近似推理的智能系统(ARIC)^[27]等。其典型结构如图3所示。图中典型的神经模糊系统由四层处理单元组成。

第一层为输入层，输入层并不改变任何量。

第二层为模糊化层，将第 i 个输入变量划分为 n_i 个模糊集 $A_{i1}, A_{i2}, \dots, A_{in_i}$ ，对于每个节点 A_{ij} 的输出为：

$$u_{A_{ij}} = f((x_i - m_{A_{ij}}) / \sigma_{A_{ij}})$$

其中 $m_{A_{ij}}$ 和 $\sigma_{A_{ij}}$ 分别为模糊集 A_{ij} 的中心点和宽度， f 为模糊隶属函数，可以为高斯函数、Sigmoid 函数、三角形等。其中用 Sigmoid 函数时相对比较复杂^[22~24]。用 Sigmoid 函数作为模糊隶属函数时，隶属函数 A_{ij} 由两个节点 A_{ij}^1 和 A_{ij}^2 来实现：

$$u_{A_{ij}} = [f_{A_{ij}^1}(x) + f_{A_{ij}^2}(x)] / 2$$

而其他类型的隶属函数只用一个节点来实现。

若要使模糊集满足：

$$\sum_{j=1}^{n_i} u_{A_{ij}}(x_i) = 1$$

则需要另一个正规化层^[28]。

第三层为规则层，其中每个节点代表一条规则，每条规则至多与第二层中的每组一个节点相连。设节点 r 且与其相连的第二层的节点分别为 $A_{1r_1}, A_{2r_2}, \dots, A_{n_r r_{n_r}}$ 则其输出为：

$$O_r(x) = \prod_{i=1}^n u_{A_{ir}}(x_i)$$

其中 $\prod$ 为乘积运算或者为 T-余范。

第四层为输出层。当采用中心法反模糊化时，其输出为：

$$y(x) = \frac{\sum_i O_i(x) w_i}{\sum_i O_i(x)}$$

其中 w_i 为输出模糊集的峰值点， $\sum$ 为和或者为 T-范算子。对于其它反模糊化的方法其计算可以直接套用。

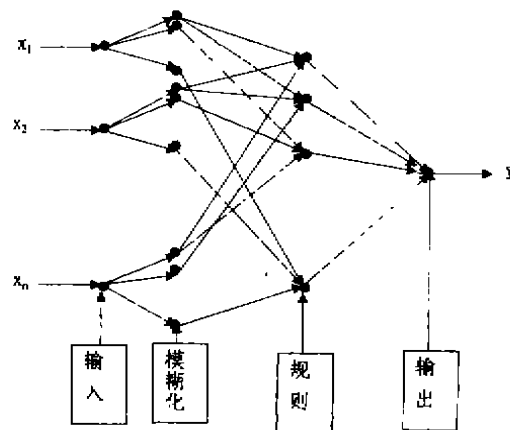


图3 典型的神经模糊系统的结构

5.3 神经模糊系统的学习

神经模糊系统的学习通常为 BP 算法。当神经网络不含有 T-范或 T-余范算子时学习算法与传统的 BP 相类似。当有 T-范或 T-余范时可由模糊算子神经网络的学习算法得到^[30]。另外，再励学习(Reinforcement)等也被用于神经模糊系统的学习。

神经模糊系统的结构可以通过对规则冗余的判断来优化^[22~24]。当模糊规则满足以下三条时，则规则为冗余的：

- 1) 模糊规则的后件相同；
- 2) 前件有一些相同；
- 3) 不同的前件的和为输入变量的支撑集。

结论 本文对前向神经网络和模糊系统的结合作了全面系统的综述。模糊系统和前向神经网络在一些条件下是等价的。这种等价性主要表现在，对于任意的具有对数激发函数的神经网络，存在 TS 型模糊系统与其等价。任意 Madani 型模糊系统是多 RNF 神经网络的组合。

模糊系统与前向神经网络的结合大致可以分为三

(下转第47页)

实际测试时,除了在正常的上下文和条件下测试对象的正确性以外,检查对象的健壮性也是非常重要的。为了测试对象的健壮性,需要有意生成不合法的方法调用序列及传给对象非法的参数。采用上文给出的策略生成的测试序列只能检验对象工作的正确性。这里我们集中讨论如何从 MtSS 生成用于检验对象健壮性的测试序列。文中我们把检验健壮性的测试用例称为负面测试用例,把不符合 MtSS 的方法序列称为非法方法序列。

一个类的 MtSS 规定了类中方法被调用的所有可能的合法顺序。照此推理,如果一个序列不符合 MtSS,则该序列就是一个非法序列。生成非法序列的方法有多种,下面我们简要介绍几种较简单的方法。

4.1 遗漏法

遗漏法是指从 MtSS 生成的方法序列中有意删掉一个或多个固定方法。所谓固定方法是指在任何生成的方法序列中必然出现的方法。例如,从 Account 类的 MtSS 可知,方法 open(), setupAccnt(), close(), 第一个 deposit 以及最后一个 withdraw 在每个生成的方法序列中都是必然出现。如果去掉其中一个或几个方法就得到一个非法的方法序列,例如,若去掉 setupAccnt() 方法,将得到一个非法方法序列。

4.2 颠倒次序法

该方法是指对调方法序列中固定方法的位置。显然,被互换位置的方法必须不同,才能使新产生的方法序列与原有序列不同。从 Account 类的 MtSS 可以看出,在所有可能的方法序列里,方法 deposit() 必然紧跟在方法 setupAccnt() 之后,如果颠倒两者的顺序,我们可得到下面的负面测试序列:

TestSeq $\Rightarrow$ open * deposit * setupAccnt * deposit * withdraw * acctSummary * withdraw * close

4.3 冗余法

冗余法指在序列里连续重复一次或多次原本只允许调用一次的方法。例如,从 Account 类的 MtSS 可知,方法 setupAccnt() 只允许出现一次,因此如果连续调用该方法两次,我们就得到一个非法方法序列,该非法方法序列能作为一个负面测试用例。

小结 对 OO 软件测试技术的研究是近几年才受到关注的领域,前些年的研究工作主要集中于 OO 建模方法和 OO 语言。学术界普遍认为测试 OO 软件要比测试传统软件更困难,所以,能否找到有效的适用 OO 软件的测试技术很大程度上影响着 OO 的前途。

测试一个类中的方法能否正确交互是 OO 软件测试的重要方面,一个类的 MtSS 明确定义调用该类中方法的合法顺序,因此从 MtSS 产生的一个方法序列能作为一个测试用例。本文提出了基于 MtSS 的测试用例生成方法。为了从数量极大的可用测试用例中选取有效的测试用例,本文提出了基于划分的测试用例生成策略,并给出了3种有效的划分准则,使用这3种划分准则,能构造出比较有效的方法序列作为测试用例。对象的健壮性是整个系统可靠性的前提和重要保证。为此,本文还讨论了如何从 MtSS 生成用于测试对象健壮性的测试用例,并给出了几种简单易用的方法。

参考文献

- 1 Kirani S, Tsai W T. Method sequence specification and verification of classes. Journal of Object-Oriented Programming, 1994, 7(6): 28~38
- 2 郑人杰主编. 计算机软件测试技术. 清华大学出版社, 1992

(上接第57页)

种类型:模糊算子神经网络、模糊信息神经网络和神经网络模糊系统。前两种是神经网络的模糊化,后一种以神经网络为依托,实现模糊系统。模糊算子神经网络采用 T-范和 T-余范算子作为神经算子。模糊算子神经网络具有计算和学习算法简单的优点,然而,其能力有限,只能实现由 $[0, 1]^n$ 到实数空间的映射。模糊信息神经网络主要用来处理模糊信息,其计算和学习都比较复杂,是三者中最为复杂的,而功能最为强大。实现模糊空间到模糊空间的映射(包含实数空间的映射)。

神经模糊系统以实数运算为主,为实现模糊系统的工具。神经模糊系统实现了模糊系统和神经网络之间的优势互补。模糊系统为神经网络提供了利用先验知识的方法,为神经网络的初始化提供基础。同时模糊系统为神经网络输出结果的解释提供了依据。神经网络为模糊系统提供了强大的学习功能,从而,为解决模糊系统中的难点——隶属函数的确定,模糊推理规则的获取和模糊规则和隶属函数的调整提供了解决手段。神经模糊系统被广泛地应用于自动控制等领域。(参考文献共37篇,略)