

自适应神经模糊推理系统建模研究^{*}

鲁斌 何华灿

(西北工业大学计算机科学与工程系 西安 710072)

Research on Modeling with Adaptive Neuro-Fuzzy Inference System

LU Bin HE Hua-Can

(Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072)

Abstract With rapid development of the fuzzy control application field, the existing system for fuzzy inferring modeling cannot more and more suit the requirements of fuzzy control. About how to apply the theories of fuzzy control to practice rapidly and conveniently, this paper presents a reasonable and practical method, which supports all sorts of fuzzy inferring system of MAMDANI and SUGENO to be modeled not only by tuning references of membership functions, but also by tuning fuzzy inferring structure. The modeling instance shows that it's practical and effective.

Keywords Universal logics, Fuzzy control, Modeling

1 引言

目前,模糊逻辑理论与应用研究已取得了丰富的成果,尤其在自动控制领域的成功应用,引起了学术界和工业界的广泛关注。与传统控制方法相比,模糊控制利用人类专家控制经验,对于非线性、复杂对象的控制显示了鲁棒性好、控制性能高的优点。模糊控制器主要采用 Mamdani 型或者 Sugeno 型模糊推理方法。Mamdani 型模糊推理,由于其规则的形式符合人们思维和语言表达的习惯,因而能够方便的表达人类的知识,而 Sugeno 型模糊推理具有计算简单,利于数学分析的优点。

针对模糊逻辑尤其是模糊控制的迅速推广应用,MathWorks 公司开发了 Fuzzy Logic 工具箱^[1]。该工具箱以其功能强大和方便易用的特点得到了用户的广泛欢迎,用户可以方便地建立模糊推理系统(FIS: fuzzy inference system)。然而,随着控制对象的改变建立一个合适的 FIS,即 FIS 的调整却比较困难。鉴于此,Fuzzy Logic 工具箱同时提供了自适应神经模糊推理系统(ANFIS: adaptive neuro-fuzzy inference system),利用反向传播(BP: Back Propagation)算法通过调整隶属函数参数来完成对输入-输出数据对的建模。

模糊逻辑与神经网络的结合形成的模糊神经网络同时具有模糊逻辑易于表达人类知识和神经网络的分布式信息存储以及学习能力的优点,对于复杂系统的建模和控制提供了有效的工具。然而,由于 ANFIS 比 FIS 复杂得多,并且 Sugeno 型模糊推理易于和优化、自适应等方法结合,因此,目前 ANFIS 只支持单输出 Sugeno 型 FIS 的建模,解模糊方法只能采用加权平均法,并且不同的规则不能共享同一个隶属函数,这大大地限制了其应用范围。由于 Mamdani 型 FIS 应用的广泛性,建立一个完善的 ANFIS 势在必行。

本文在 Sugeno 型 ANFIS 的基础上,建立了一个功能完善、方便易用的扩展型 ANFIS (EANFIS: Extended ANFIS),它支持各种形式的 Mamdani 型和 Sugeno 型 FIS 建模,

不仅可以调整隶属函数参数,更可以调整模糊推理结构。

2 扩展型 ANFIS 的结构

EANFIS 不仅具备 Sugeno 型 ANFIS 的全部功能,而且支持各种形式的 Mamdani 型和 Sugeno 型 FIS 建模,采用 BP 算法,既可以单独调整隶属函数参数,又可以单独调整模糊推理结构,而且还可以一起进行调整。由于隶属函数参数的调整采用了与 Sugeno 型 ANFIS 类似的方法,因此本文只着重介绍模糊推理结构的调整方法。

为了实现变结构模糊推理功能^[2,3],EANFIS 采用了文[4]提供的可包容一切逻辑形态和推理模式的、灵活的、开放的、自适应的泛逻辑算子簇,对模糊推理操作的基本算子进行参数化,实现了规则前提的集成、规则的激活以及规则输出合成的柔性化,从而高度模型化了系统。

在具体介绍模糊推理结构的调整方法之前,先总结模糊推理的一般过程。由于 Sugeno 型 FIS 可以看作是 Mamdani 型 FIS 的特例,下面我们就以 Mamdani 型 FIS 为主进行研究。

2.1 模糊推理的一般过程

考虑一个多输入单输出的 Mamdani 型 FIS,给定的模糊规则集具有以下形式:

$$\begin{aligned} \text{IF } x_1 = A_{i1} \text{ AND } \dots \text{ AND } x_n = A_{in} \\ \text{THEN } y = B_i, i = 1, 2, \dots, M \end{aligned} \quad (1)$$

其中, $x_i \in X_i, y \in Y, A_{i1}, A_{i2}, \dots, A_{in}$ 为语言变量 x_i 的语言值(模糊集合), B_1, B_2, \dots, B_M 为语言变量 y 的语言值。

在 Mamdani 模型中,推理过程如下:

(A1) 逻辑联接词 AND 用来连接规则中的各个前提,通常被定义为 $\text{MIN}(\wedge)$ 算子;当然,逻辑联接词 OR ($\text{MAX}(\vee)$) 也常常用来连接规则中的各个前提。

第 i 条规则的激活度为:

$$\tau_i = A_{i1}(x_1) \wedge A_{i2}(x_2) \wedge \dots \wedge A_{in}(x_n).$$

(A2) 前提和结论的蕴涵操作采用 $\text{MIN}(\wedge)$ 算子。

第 i 条规则的输出为:

^{*} 基金项目: 国家教委博士学科点专项科学基金(98069923)。鲁斌 博士生,主要从事人工智能理论,不精确推理的研究。

$$F_i(y) = \tau_i \wedge B_i(y).$$

(A3)逻辑联接词 OR 用来合成所有规则的输出,通常被定义为 MAX(V)算子。

控制器的合成输出 F 为:

$$F(y) = \bigvee_{i=1}^M F_i(y) = \bigvee_{i=1}^M \tau_i \wedge B_i(y).$$

(A4)解模糊输出值通常由重心法(COG)得到。

现在,我们将对 Mamdani 推理的(A1)~(A3)过程进行柔性化,为了实现这一目的,我们使用泛模糊逻辑算子簇。

2.2 泛模糊逻辑算子簇

不考虑外界干扰等因素,在 EANFIS 中,使用文[4]提供的零级泛与运算、零级泛或运算和零级泛蕴含运算来参数化模糊推理操作。如果系统输入含有测量误差或外界干扰,那么可先进行卡尔曼滤波,然后再进行后续处理。

下面分别对各个算子簇予以介绍:

(1) 零级泛与运算:

$$T(x_1, x_2, \dots, x_n, h) = (\max(0^m, x_1^m + x_2^m + \dots + x_n^m - (l-1)))^{\frac{1}{m}} \quad (2)$$

(2) 零级泛或运算:

$$S(x_1, x_2, \dots, x_n, h) = 1 - (\max(0^m, (1-x_1)^m + (1-x_2)^m + \dots + (1-x_n)^m - (l-1)))^{\frac{1}{m}}$$

(3) 零级泛蕴含运算:

$$I(x, y, h) = (\min(1+0^m, 1-x^m+y^m))^{\frac{1}{m}} \quad (3)$$

在上述三个算子簇中,广义相关系数 h 与 m 具有以下关系:

$$m = (3-4h)/4h(1-h), h \in [0, 1], m \in \mathbb{R}.$$

2.3 模糊推理过程的柔性化

使用零级泛与运算、零级泛或运算和零级泛蕴含运算替换 Mamdani 推理的(A1)~(A3)过程中的基本算子,从而使模糊推理操作柔性化,具体构造过程如下:

(1) 规则前提合成的柔性化

将规则中连接各个前提的 AND 用零级泛与运算替换,得到:

$$\tau_i = T(x_1, x_2, \dots, x_n, h_{T_i}) = (\max(0^m \tau_{i1}, x_{11}^m \tau_{i1} + x_{21}^m \tau_{i1} + \dots + x_{n1}^m \tau_{i1} - (n-1)))^{\frac{1}{m}}.$$

当 $h=1$ 时, $\tau_i = \min(x_1, x_2, \dots, x_n)$, 这是常见的 MIN(\wedge)算子。

当 $h=0.75$ 时, $\tau_i = x_1 x_2 \dots x_n$, 这是算术积(product)算子。

当 $h=0.5$ 时, $\tau_i = \max(0, x_1 + x_2 + \dots + x_n - (n-1))$ 这是有界积(bounded product)算子。

如果规则中的前提是用逻辑联接词 OR 连接的,那么将 OR 用零级泛或运算替换,得到:

$$\tau_i = S(x_1, x_2, \dots, x_n, h_s) = 1 - (\max(0^m s, (1-x_1)^m s + (1-x_2)^m s + \dots + (1-x_n)^m s - (n-1)))^{\frac{1}{m}}$$

当 $h=1$ 时, $\tau_i = \max(x_1, x_2, \dots, x_n)$, 这是常见的 MAX(V)算子。

当 $h=0.75$ 时, $\tau_i = 1 - (1-x_1)(1-x_2)\dots(1-x_n)$, 这是算术和(sum)算子。

当 $h=0.5$ 时, $\tau_i = \min(n-1, x_1 + x_2 + \dots + x_n)$, 这是有界和(bounded sum)算子。

(2) 模糊蕴含操作的柔性化

使用零级泛蕴含运算,可以得到第 i 个规则输出的模糊子集 $F_i(y)$:

$$F_i(y) = I(\tau_i, B_i(y), h_I) = (\min(1+0^m I, 1-\tau_i^m I + B_i^m I))$$

$$(y))^{\frac{1}{m I}}.$$

当 $h=0.75$ 时, $F_i(y) = \min(1, B_i(y)/\tau_i)$, 这是 Goguen 蕴含算子。

当 $h=0.5$ 时, $F_i(y) = \min(1, 1-\tau_i + B_i(y))$, 这是 Lukasiewicz 蕴含算子。

(3) 规则输出合成的柔性化

由泛蕴含运算的性质知道 $B_i \subset F_i$, 因此,根据文[4],在合成所有规则的输出时,使用零级泛与运算代替 OR, 得到:

$$F(y) = T(F_1(y), F_2(y), \dots, F_M(y), h_{T_2}) = (\max(0^m T_2, F_1^m T_2(y) + F_2^m T_2(y) + \dots + F_M^m T_2(y) - (M-1)))^{\frac{1}{m T_2}}$$

当 $h=1$ 时, $F(y) = \min(F_1(y), F_2(y), \dots, F_M(y))$, 这是常见的 MIN(\wedge)算子。

当 $h=0.75$ 时, $F(y) = F_1(y)F_2(y)\dots F_M(y)$, 这是算术积(product)算子。

当 $h=0.5$ 时, $F(y) = \max(0, F_1(y) + F_2(y) + \dots + F_M(y) - (n-1))$, 这是有界积(bounded product)算子。

通过上述三步柔性化过程,即可构造出变结构的模糊推理系统,可用图 1 形象地说明。

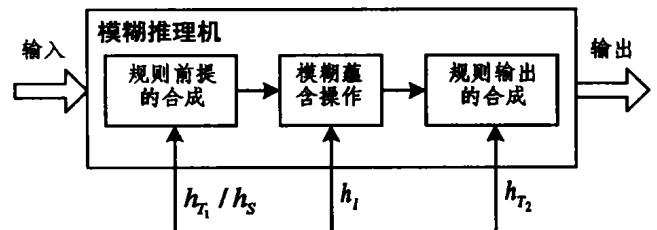


图 1 模糊推理过程的柔性化

3 模糊推理结构的学习方法

讨论了 EANFIS 的结构,再来研究其学习方法。本文采用了具有自适应学习率的 BP 算法来进行模糊推理结构参数的学习。

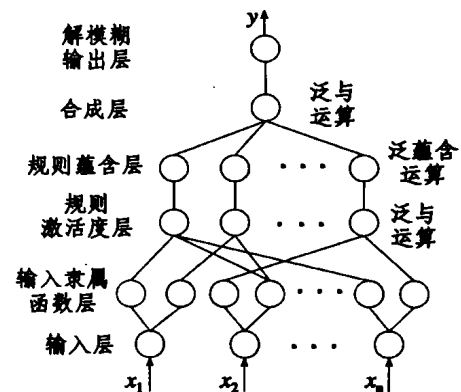


图 2 模糊神经网络结构图

3.1 模糊神经网络结构

我们来研究一个多输入单输出的模糊推理系统的学习算法,模糊规则与式(1)具有相同的形式,可以用一个六层的模糊神经网络(FNN: Fuzzy Neural Networks)来实现,其结构如图 2 所示。

假设 x_i^k 表示第 k 层的第 i 个输入, net_j^k 表示第 k 层的第 j 个结点的净输入, y_j^k 表示第 k 层的第 j 个结点的输出,即 $y_j^k = x_j^{k+1}$, n 表示输入个数, M 表示规则数,则 FNN 的各层的处理

过程可表示如下:

① 输入层:

$$net_j^1 = x_j^1, j = i, y_j^1 = net_j^1$$

其中, x_j^1 就是 FNN 的第 i 个输入 $x_i, i = 1, 2, \dots, n$. 此层共有 n 个结点.

② 输入隶属函数层:

$$net_j^2 = f_{ij}(x_i), y_j^2 = net_j^2$$

其中, f_{ij} 是第 i 个输入变量在第 j 个结点中的模糊集合的隶属函数, $j = 1, 2, \dots, N, N$ 表示输入隶属函数的总个数. 此层共有 N 个结点.

③ 规则激活度层:

$$net_j^3 = T(x_1^2, x_2^2, \dots, x_n^2, h_{Tj}), y_j^3 = net_j^3$$

其中, y_j^3 即第 j 条规则的激活度 a_j, h_{Tj} 是第 j 条规则前提的广义相关系数, $j = 1, 2, \dots, M$. 此层共有 M 个结点.

④ 规则蕴含层:

$$net_j^4 = I(x_1^3, g_j(y), h_{Ij}), y_j^4 = net_j^4$$

其中, g_j 是输出变量在第 j 条规则中的语言值的隶属函数, y_j^4 是第 j 个规则的输出, h_{Ij} 是第 j 条规则蕴含的广义相关系数, $j = 1, 2, \dots, M$. 此层共有 M 个结点.

⑤ 合成层:

$$net_1^5 = T(x_1^4, x_2^4, \dots, x_M^4, h_{T_2}), y_1^5 = net_1^5$$

其中, y_1^5 是所有规则输出的合成, h_{T_2} 是规则输出合成的广义相关系数. 此层只有一个结点.

⑥ 解模糊输出层:

$$net_1^6 = F(x_1^5), y_1^6 = net_1^6$$

其中, F 是解模糊函数, y_1^6 是 FNN 的输出. 此层只有一个结点.

3.2 具有自适应学习率的 BP 算法

在给出模糊推理结构的学习方法之前,先证明下面几个公式,由于零级泛或运算具有与零级泛与运算类似的结构,故只给出零级泛与运算和零级泛蕴含运算的相关公式.

$$\text{公式 1: } \frac{dm}{dh} = \frac{d\left(\frac{3-4h}{4h(1-h)}\right)}{dh} = -\frac{4h^2-6h+3}{4h^2(1-h)^2},$$

$$h \in (0, 0.75) \text{ 或 } h \in (0.75, 1).$$

证明: 由于 $m = \frac{3-4h}{4h(1-h)}$, 两边取对数得: $\ln m = \ln(3-4h) - \ln 4 - \ln h - \ln(1-h)$, 两边对 h 求导得: $\frac{1}{m} \left(\frac{dm}{dh}\right) = \frac{-4}{3-4h} - \frac{1}{h} - \frac{-1}{1-h} = \frac{1}{1-h} - \frac{1}{h} - \frac{4}{3-4h}$, 于是,

$$\frac{dm}{dh} = \frac{3-4h}{4h(1-h)} \left(\frac{1}{1-h} - \frac{1}{h} - \frac{4}{3-4h}\right) = -\frac{4h^2-6h+3}{4h^2(1-h)^2}.$$

证毕.

$$\text{公式 2: } \frac{-1}{m^2} \frac{dm}{dh} = 1 + \frac{3}{(4h-3)^2} = C,$$

$$h \in (0, 0.75) \text{ 或 } h \in (0.75, 1).$$

证明: $\frac{-1}{m^2} \frac{dm}{dh} = -\frac{16h^2(1-h)^2}{(3-4h)^2} \left(-\frac{4h^2-6h+3}{4h^2(1-h)^2}\right) = \frac{16h^2-24h+12}{16h^2-24h+9} = 1 + \frac{3}{(4h-3)^2}$, 用 C 表示原式即得.

证毕.

$$\text{公式 3: } \frac{\partial T(x_1, x_2, \dots, x_n, h)}{\partial x_j} = A_T^{\frac{1}{m}-1} x_j^{m-1}, \text{ 其中, } A_T = x_1^m + x_2^m + \dots + x_n^m - (n-1).$$

证明: 令 $A_T = x_1^m + x_2^m + \dots + x_n^m - (n-1)$, 由(2)式得

$$\frac{\partial T(x_1, x_2, \dots, x_n, h)}{\partial x_j} = \frac{\partial(x_1^m + x_2^m + \dots + x_n^m - (n-1))^{\frac{1}{m}}}{\partial x_j}$$

$$= \frac{1}{m} (x_1^m + x_2^m + \dots + x_n^m - (n-1))^{\frac{1}{m}-1} m x_j^{m-1} =$$

$$A_T^{\frac{1}{m}-1} x_j^{m-1}$$

证毕.

公式 4:

$$\frac{\partial T(x_1, x_2, \dots, x_n, h)}{\partial h} =$$

$$\begin{cases} A_T^{\frac{1}{m}-1} (A_T \ln A_T - m B_T) C, & h \in (0.75, 1) \text{ 或 } h \in (0, 0.75) \text{ 且 } A_T > 0 \\ 0, & h \in (0, 0.75) \text{ 且 } A_T \leq 0 \\ \text{不存在} & h = 0, 0.75, 1 \end{cases}$$

其中, $B_T = x_1^m \ln x_1 + x_2^m \ln x_2 + \dots + x_n^m \ln x_n$.

证明: 根据文[4], 当 $h \in (0, 0.75)$ 且 $A_T \leq 0$, 或 $h = 0, 0.75, 1$ 时, 公式显然成立, 现在只需证明 $h \in (0.75, 1)$ 或 $h \in (0, 0.75)$ 且 $A_T > 0$ 时的情形.

C 同公式 2, A_T 同公式 3, 令 $B_T = x_1^m \ln x_1 + x_2^m \ln x_2 + \dots + x_n^m \ln x_n$, 由(2)式得

$$\frac{\partial T(x_1, x_2, \dots, x_n, h)}{\partial h} = \frac{\partial(x_1^m + x_2^m + \dots + x_n^m - (n-1))^{\frac{1}{m}}}{\partial h}$$

$$= \frac{\partial(x_1^m + x_2^m + \dots + x_n^m - (n-1))^{\frac{1}{m}} \frac{\partial m}{\partial h}}{\partial m} = (x_1^m + x_2^m + \dots + x_n^m - (n-1))^{\frac{1}{m}-1} (m(x_1^m \ln x_1 + x_2^m \ln x_2 + \dots + x_n^m \ln x_n) - (x_1^m + x_2^m + \dots + x_n^m - (n-1)) \ln(x_1^m + x_2^m + \dots + x_n^m - (n-1))) \frac{1}{m^2} \frac{\partial m}{\partial h}$$

$$= A_T^{\frac{1}{m}-1} (m B_T - A_T \ln A_T) (-C) = A_T^{\frac{1}{m}-1} (A_T \ln A_T - m B_T) C$$

证毕.

$$\text{公式 5: } \frac{\partial I(x_1, x_2, h)}{\partial x_j} = (-1)^j A_I^{\frac{1}{m}-1} x_j^{m-1}, \text{ 其中, } A_I = 1 - x_1^m + x_2^m, j = 1, 2.$$

证明: 令 $A_I = 1 - x_1^m + x_2^m$, 由(3)式得

$$\frac{\partial I(x_1, x_2, h)}{\partial x_j} = \frac{\partial(1 - x_1^m + x_2^m)^{\frac{1}{m}}}{\partial x_j}$$

$$= \frac{1}{m} (1 - x_1^m + x_2^m)^{\frac{1}{m}-1} (-1)^j m x_j^{m-1} = (-1)^j A_I^{\frac{1}{m}-1} x_j^{m-1}$$

证毕.

公式 6:

$$\frac{\partial I(x_1, x_2, h)}{\partial h} =$$

$$\begin{cases} A_I^{\frac{1}{m}-1} (A_I \ln A_I - m B_I) C, & h \in (0.75, 1) \text{ 或 } h \in (0, 0.75) \text{ 且 } A_I < 1 \\ 0, & h \in (0, 0.75) \text{ 且 } A_I \geq 1 \\ \text{不存在} & h = 0, 0.75, 1 \end{cases}$$

其中, $B_I = -x_1^m \ln x_1 + x_2^m \ln x_2$.

证明: 根据文[4], 当 $h \in (0, 0.75)$ 且 $A_I \geq 1$, 或 $h = 0, 0.75, 1$ 时, 公式显然成立, 现在只需证明 $h \in (0.75, 1)$ 或 $h \in (0, 0.75)$ 且 $A_I < 1$ 时的情形.

C 同公式 2, A_I 同公式 5, 令 $B_I = -x_1^m \ln x_1 + x_2^m \ln x_2$, 由(3)式得

$$\frac{\partial I(x_1, x_2, h)}{\partial h} = \frac{\partial(1 - x_1^m + x_2^m)^{\frac{1}{m}}}{\partial h} = \frac{\partial(1 - x_1^m + x_2^m)^{\frac{1}{m}} \frac{\partial m}{\partial h}}{\partial m}$$

$$= (1 - x_1^m + x_2^m)^{\frac{1}{m}-1} (-(1 - x_1^m + x_2^m) \ln(1 - x_1^m + x_2^m) + m(x_1^m \ln x_1 + x_2^m \ln x_2)) \frac{1}{m^2} \frac{\partial m}{\partial h}$$

$$= A_I^{\frac{1}{m}-1} (-A_I \ln A_I + m B_I) (-C) = A_I^{\frac{1}{m}-1} (A_I \ln A_I - m B_I) C$$

证毕.

FNN 具有多层感知器的结构形式, 采用具有自适应学习率的 BP 算法来修正其中的可调参数 h_{Tj}, h_{Ij} 和 $h_{T_2}, j = 1, 2$.

..., M, 定义指标函数为:

$$E(n) = \frac{1}{2} (d(n) - y_1^s(n))^2, \text{ 其中 } y_1^s(n) \text{ 为第 } n \text{ 次迭代的}$$

输出, d(n) 为参考输出。

那么误差信号将由第六层向第一层依次反传。

① 解模糊输出层:

$$\delta_1^s(n) = \frac{-\partial E(n)}{\partial net_1^s(n)} = d(n) - y_1^s(n)$$

② 合成层:

$$\delta_1^s(n) = \frac{-\partial E(n)}{\partial net_1^s(n)} = \frac{-\partial E(n)}{\partial net_1^s(n)} \frac{\partial net_1^s(n)}{\partial x_1^s(n)} = \delta_1^s(n) F'(x_1^s(n))$$

$$\Delta h_{T_2}(n) = \frac{-\partial E(n)}{\partial h_{T_2}(n)} = \frac{-\partial E(n)}{\partial net_1^s(n)} \frac{\partial net_1^s(n)}{\partial h_{T_2}(n)} = \delta_1^s(n) \frac{\partial T(x_1^s(n), x_2^s(n), \dots, x_5^s(n), h_{T_2}(n))}{\partial h_{T_2}(n)}$$

③ 蕴含层:

$$\delta_1^s(n) = \frac{-\partial E(n)}{\partial net_1^s(n)} = \frac{-\partial E(n)}{\partial net_1^s(n)} \frac{\partial net_1^s(n)}{\partial x_1^s(n)} = \delta_1^s(n) \frac{\partial T(x_1^s(n), x_2^s(n), \dots, x_5^s(n), h_{T_2}(n))}{\partial x_1^s(n)}$$

$$\Delta h_{I_j}(n) = \frac{-\partial E(n)}{\partial h_{I_j}(n)} = \frac{-\partial E(n)}{\partial net_1^s(n)} \frac{\partial net_1^s(n)}{\partial h_{I_j}(n)} = \delta_1^s(n) \frac{\partial I(x_1^s(n), g_j(y), h_{I_j}(n))}{\partial h_{I_j}(n)}$$

④ 规则激活度层:

$$\delta_j^s(n) = \frac{-\partial E(n)}{\partial net_j^s(n)} = \frac{-\partial E(n)}{\partial net_j^s(n)} \frac{\partial net_j^s(n)}{\partial x_1^s(n)} = \delta_1^s(n) \frac{\partial I(x_1^s(n), g_j(y), h_{I_j}(n))}{\partial x_1^s(n)}$$

$$\Delta h_{T_{1,j}}(n) = \frac{-\partial E(n)}{\partial h_{T_{1,j}}(n)} = \frac{-\partial E(n)}{\partial net_j^s(n)} \frac{\partial net_j^s(n)}{\partial h_{T_{1,j}}(n)} = \delta_j^s(n) \frac{\partial T(x_1^s(n), x_2^s(n), \dots, x_5^s(n), h_{T_{1,j}}(n))}{\partial h_{T_{1,j}}(n)}$$

于是, FNN 中各可调参数的学习算法为

$$h_{T_2}(n+1) = h_{T_2}(n) + \eta \Delta h_{T_2}(n)$$

$$h_{I_j}(n+1) = h_{I_j}(n) + \eta \Delta h_{I_j}(n), j=1, 2, \dots, M$$

$$h_{T_{1,j}}(n+1) = h_{T_{1,j}}(n) + \eta \Delta h_{T_{1,j}}(n), j=1, 2, \dots, M$$

其中, η 为可调参数的学习率。

训练过程中学习率的调整采用如下的策略:

- 当误差连续四次减小时, 则增加学习率;
- 当误差变化连续两次出现振荡即一次增加和一次减少交替发生, 则减小学习率。

为了提高 BP 算法的收敛速度, 可调参数的起始值应选在 0.5 左右。至于其它改进收敛速度的措施, 可以参考文[5]。

至此, 模糊控制结构的学习算法构造完成, 下面通过仿真实例来说明 EANFIS 的建模方法及其有效性。

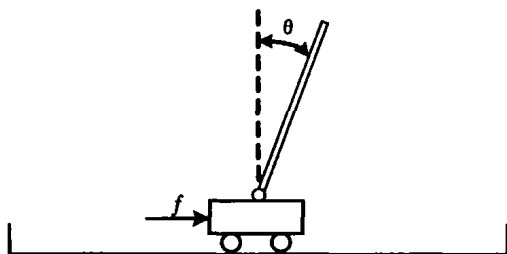


图3 一阶倒立摆系统

4 EANFIS 建模应用

以图3所示的一阶倒立摆为例, 说明使用 EANFIS 建模

的方法及效果。摆杆的底部用铰链固定在一辆可以在有限或无限长的轨道上运动的小车, 通过向小车施力来控制摆杆保持垂直。该系统的数学模型如下:

$$\ddot{\theta} = \frac{g \sin \theta - \cos \theta \left[\frac{f + m_p l \dot{\theta}^2 \sin \theta}{m_c + m_p} \right]}{l \left[\frac{4}{3} - \frac{m_p (\cos \theta)^2}{m_c + m_p} \right]}$$

$$\ddot{x} = \frac{f + m_p l [\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta]}{m_c + m_p}$$

倒立摆动力方程的参数设置如下: $g=9.8 \text{ m/s}^2$; m_c 是小车的质量 (1.0kg); m_p 是倒立摆的质量 (0.1kg); l 是倒立摆的长度 (0.5m)。

4.1 建立初始 FIS

本例以摆杆偏角 θ (rad) 和角速度 $\dot{\theta}$ (rad/s) 为输入量, 外力 f (N) 为输出量, 约束条件是:

$$-0.3 \leq \theta \leq 0.3, -1 \leq \dot{\theta} \leq 1, -30 \leq f \leq 30$$

都采用均匀对称的高斯型隶属函数作为语言值, 如表 1。

表 1 训练参数表

隶属函数		参数	宽度 σ	中心点 c
输入 1	N(负)		0.2548	-0.3
	P(正)		0.2548	0.3
输入 2	N(负)		0.8493	-0.1
	P(正)		0.8493	0.1
输出	GN(负大)		8.493	-30
	N(负)		8.493	-10
	P(正)		8.493	10
	GP(正大)		8.493	30

控制规则 4 条, 如表 2。其中, 各条规则所有前提之间是与关系, 规则权重为 1。

表 2 训练参数表

输入 1		输入 2	N	P
输入 1	N		GN	N
	P		P	GP

FIS 设置如下:

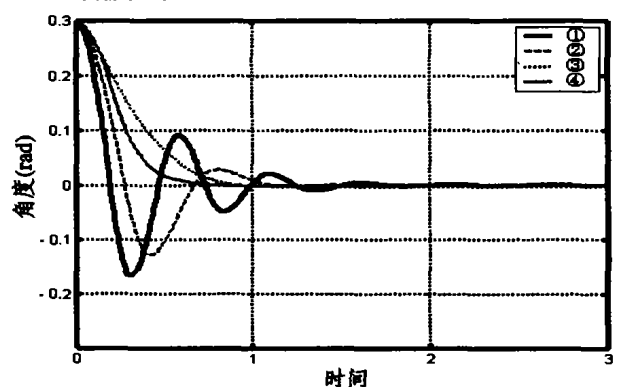


图 4 结果比较

模糊推理与算子取算术积 (product, 也就是 prod 算子), 或算子取算术和 (sum, 也就是 probor 算子), 蕴含算子取算术积 (product), 合成算子取算术和 (sum), 采用重心法 (centroid) 解模糊。

设置一阶倒立摆的初始值为：
 $\theta = 0.3 \text{ rad}, \dot{\theta} = 0 \text{ rad/s}$, 采样时间为 0.01s。

使用初始 FIS 进行仿真得到图 4 中的结果①, 超调大, 到达期望值(倒立摆稳定竖直)的过渡时间较长。

4.2 FIS 的训练过程

下面使用 100 个样本点分别采用三种方法训练一阶倒立摆的 FIS, 并且比较其训练结果, 观察是否有所改善。

方法一: 只训练隶属函数参数。

使用初始 FIS, 训练 47 次后得到图 4 中的结果②。

训练前后隶属函数参数的变化情况请参看表 3。

表 3 隶属函数参数表

隶属函数	参数	宽度 σ		中心点 c	
		训练前	训练后	训练前	训练后
输入 1	N	0.2548	0.315	-0.3	-0.2557
	P	0.2548	0.2721	0.3	0.2103
输入 2	N	0.8493	0.7012	-0.1	-0.9707
	P	0.8493	0.6383	0.1	0.9892
输出	GN	8.493	8.4930	-30	-30
	N	8.493	8.4957	-10	-9.9987
	P	8.493	8.4810	10	9.8335
	GP	8.493	8.4933	30	29.9999

方法二: 只训练模糊推理结构。

模糊推理与算子取零级泛与算子簇 T_1 , 或算子取零级泛或算子簇 S , 蕴含算子取零级泛蕴含算子簇 I , 合成算子取零级泛与算子簇 T_2 , 输入输出隶属函数参数取初始值, 采用重心法解模糊, 训练 23 次后得到图 4 中的结果③。

训练前后模糊推理算子参数的变化情况请参看表 4。由于本例中各条规则所有前提之间均为“与”关系, 故模糊推理或算子没有参与运算。

表 4 模糊推理算子参数表

参数	规则	规则 1	规则 2	规则 3	规则 4
		h_{T_1}	训练前	0.6	0.6
	训练后	0.6	0.6	0.5737	0.57471
h_I	训练前	0.5	0.5	0.5	0.5
	训练后	0.5	0.5	0.11166	0.57229
h_{T_2}	训练前	0.5			
	训练后	0.5			

方法三: 隶属函数参数和模糊推理结构一起进行训练。

模糊推理与算子取零级泛与算子簇 T_1 , 或算子取零级泛或算子簇 S , 蕴含算子取零级泛蕴含算子簇 I , 合成算子取零级泛与算子簇 T_2 , 输入输出隶属函数参数取初始值, 采用重心法解模糊, 训练 12 次后得到图 4 中的结果④。

训练前后隶属函数参数的变化情况请参看表 5-1。

表 5-1 隶属函数参数表

隶属函数	参数	宽度 σ		中心点 c	
		训练前	训练后	训练前	训练后
输入 1	N	0.2548	0.3201	-0.3	-0.2631
	P	0.2548	0.2712	0.3	0.191
输入 2	N	0.8493	0.8637	-0.1	-0.9703
	P	0.8493	0.8709	0.1	0.9892
输出	GN	8.493	8.4930	-30	-30
	N	8.493	8.4957	-10	-9.9987
	P	8.493	8.4810	10	9.8335
	GP	8.493	8.4933	30	29.9999

模糊推理算子参数的变化情况请参看表 5-2。

表 5-2 模糊推理算子参数表

参数	规则	规则 1	规则 2	规则 3	规则 4
		h_{T_1}	训练前	0.6	0.6
	训练后	0.6	0.6	0.57378	0.57471
h_I	训练前	0.5	0.5	0.5	0.5
	训练后	0.5	0.5	0.205	0.57229
h_{T_2}	训练前	0.5			
	训练后	0.5			

比较图 4 中的结果①~④可以看出, 结果②比结果①超调小; 结果③没有超调, 并且到达期望值的过渡时间很短; 结果④不但无超调, 而且达到期望值的过渡时间比③更短。就本例而言, 方法三优于方法二, 方法二优于方法一, 这在一定程度上说明合适的模糊推理结构对改善模糊推理效果起着重要的作用。当然, 建模时究竟采用哪种方法效果最好, 应具体问题具体分析, 最终目标就是建立一个适合控制对象的 FIS 模型。EANFIS 为此提供了一个高效便捷的途径。

结论 模糊控制理论研究促进了模糊控制应用领域的迅速发展, 如何将自适应、变结构模糊控制等先进的理论成果迅速、方便地用于实践, 本文提供了一种切实可行的途径。建模实例表明, EANFIS 是实用的和有效的。

参考文献

- 1 User's Guide version 2 of Fuzzy Logic Toolbox for use with MATLAB [M]. Mathworks, 2001
- 2 Yager R R. Analysis of flexible structured fuzzy logic controllers [J]. IEEE Trans. System, Man, and Cybernetics, 1994, 24(7): 1035 ~ 1043
- 3 陈丹, 何华灿, 王晖. 基于连续可控 T 范数的模糊控制方法研究 [J]. 控制理论与应用, 2001, 18(5): 717 ~ 721
- 4 何华灿, 等. 泛逻辑学原理[M]. 北京: 科学出版社, 2001
- 5 张乃尧, 阎平凡. 神经网络与模糊控制[M]. 北京: 清华大学出版社, 1998