文章编号：1000－8152(2001)02－0241－04

基于增强型算法并能自动生成规则的模糊神经网络控制器

吴耿锋　　傅忠谦
（上海大学计算机学院·上海，200072）（中国科技大学电子科学技术系·合肥，230026）

摘要：给出了一种基于增强型算法并能自动生成控制规则的模糊神经网络控制器 RBFNNC(reinforcements based fuzzy neural network controller)．该控制器能根据被控对象的状态通过增强型学习自动生成模糊控制规则．RBFNNC 用于倒立摆小车平衡系统控制的仿真实验表明了该系统的结构及增强型学习算法是有效和成功的．

关键词：增强型学习；规则生成；神经网络；模糊控制器
文献标识码：A

Reinforcement Based Fuzzy Neural Network Control with Automatic Rule Generation

WU Gengfeng
(School of Computer Engineering and Science, University of Shanghai · Shanghai, 200072, P. R. China)
FU Zhongqian
(Department of Electronic Science and Technolgy, University of Science and Technology of China · Hefei, 230026, P. R. China)

Abstract: A reinforcement based fuzzy neural network controller (RBFNNC) is proposed. A set of optimised fuzzy control rules can be automatically generated through reinforcement learning based on the state variables of object system. RBFNNC was applied to a cart-pole balancing system and shows significant improvements on the rule generation.

Key words: reinforcement learning; rule generation; neural network; fuzzy controller

1　引言(Introduction)

神经网络和模糊控制器的结合构成了模糊神经网络控制器．这样的控制器不仅具有神经网络的学习和记忆功能，又具有模糊控制器易于描述结构化知识、推理清晰的特点[1,2]．

设计一个性能优良的模糊神经网络控制器的关键是设计一组优化的推理规则．对于输入输出样本知识较多的控制系统，可以有多种方法得到规则，例如用基于竞争 AVQ(adaptive vector quantization)的积空间聚类方法就能有效地获得控制规则[2]．但是，对于输入输出样本知识缺少或不完善的控制系统，作为机器学习方法之一的增强型算法却有独到的用武之地．增强型算法仅仅依赖于系统的状态，不断评估系统模糊误差和所谓的增强因子，通过删选和优化获得一组有效的控制规则．本文构造了一个基于增强型算法并能自动生成控制规则的模糊神经网络控制器 RBFNNC(reinforcement based fuzzy neural network controller)，仿真试验表明，该控制器用于具有非线性动力学特征的倒立摆小车平衡系统取得良好的控制效果．文中较为详细地介绍了系统的结构、增强型算法的具体实现及仿真试验的结果．

2　RBFNNC 的系统结构(System architecture of RBFNNC)

RBFNNC 系统组成如图 1 所示．其中，模糊神经网络控制模块、信号统计处理模块和被控对象组成了基本模糊控制器．增强因子评估模块和规则自动生成模块赋予系统学习和自适应调整的功能．下面分述各部分的功能：

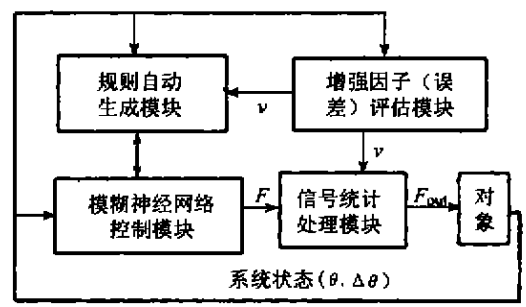


图 1　RBFNNC 系统结构
Fig. 1　Architecture of RBFNNC system

收稿日期：1999－01－19；收修改稿日期：2000－04－13．

模糊神经网络控制模块：该模块由一个五层网络组成(如图2所示)，其主要功能是存储模糊控制规则，并以系统状态为精确量输入，完成对输入的模糊化、模糊规则推理和输出的去模糊处理，最后输出精确控制量.网络的模糊规则可由专家给出，也可由规则自动生成模块自动生成.

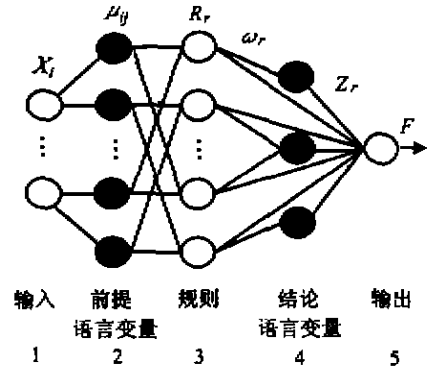


图 2 模糊神经网络

Fig. 2 Fuzzy neural network

网络的第一层为输入层，每个节点表示1个输入变量，它把精确量传送到第二层进行模糊化，该层每个节点表示输入隶属函数的1个语言变量(如正大、负小等)，其输出 $\mu_{ij}(x)$ 将馈至所有用到该前提的规则中去.这里关键是隶属函数的值就是网络的权重.

第三层为规则层，每个节点表示1条规则，它对来自第二层的模糊输入 $\mu_{ij}(x)$ 进行所谓 softmin 运算[3]，得到每条规则的激活强度(或称该规则可被应用的信任度).定义第 r 条规则的激活强度 ω_r 为：

$$\omega_r = \frac{\sum_i \mu_{ij} e^{-k\mu_{ij}}}{\sum_i e^{-k\mu_{ij}}}, \tag{1}$$

参数 k 控制 softmin 运算的硬度，当 $k \to \infty$ 时，该运算相当于一般的最小化运算.当 k 为有限值时，将得到输入的一个可微函数.

第四层的每个节点代表输出隶属函数的1个语言值，它对 softmin 的运算结果进行去模糊处理.增强型算法希望知道每条规则在控制过程中对总输出的贡献，这里采用反函数来计算每条规则的精确输出.考虑到隶属函数多取为高斯函数或三角形函数，这里以三角形函数为例，则其函数表达式如下：

$$\mu_{\Psi(SC,SL,SR)}(x) = \begin{cases} 1 - |x - SC|/SR, & x \in [SC, SC + SR], \\ 1 - |x - SC|/SL, & x \in [SC - SL, SC], \\ 0, & \text{otherwise}. \end{cases}$$

其中 Ψ 表示某个语言变量，如正大、负小等，SC,SL,SR 分别表示语言变量 Ψ 的三角形隶属函数的中心值和左、右边界值.例如，规则 r 在正大(PL, Positive Large)模糊集中的输出表示为：

$$Z_r = \mu^{-1}_{PL(SC,SL,SR)}(\omega_r). \tag{2}$$

所有被激活的规则的输出最后由第五层通过加权平均得精确控制量 F：

$$F = \frac{\sum_r \omega_r \mu^{-1}_{\Psi(SC,SL,SR)}(\omega_r)}{\sum_r \omega_r}. \tag{3}$$

增强因子评估模块：该模块的主要功能是为增强型算法提供评估依据.它对系统的状态进行分析并给出系统模糊误差 E，进而得到增强因子 v 以确定奖惩强度，从而获得最佳控制输出.模糊误差 E 可通过比较实际输出和期望输出直接获得，也可以通过一组描述模糊误差的模糊规则(可因被控制系统性能而异)间接获得[4,5].本模块采用后者的方法，即用一组模糊规则(见图8)进行推理获得模糊误差 E，这样做不仅简化了计算，而且因引入模糊推理，提高了学习速度.获得 E 后可按下式求得增强因子 v：

$$v = 1 - 2 * |E|. \tag{4}$$

信号统计处理模块：该模块实际上是对来自模糊神经网络控制模块的输出 F 和增强因子评估模块的输出 v 进行统计处理.它在 F 的均值和标准偏差 e^{-v} 的基础上用高斯随机变量进行计算[3]获得直接作用于被控对象的输出 F_{out}.增强因子 v 对系统的影响在这里得到充分的体现.

规则自动生成模块：该模块在系统完全没有输入、输出样本知识的情况下，提供一个基于增强型算法的规则自动生成程序，其唯一要求的输入是系统的状态和增强因子 v.具体算法见第3节.

倒立摆小车平衡系统：倒立摆问题是一个典型的非线性动力学系统控制问题.用传统的方法控制必须先建立复杂的数学模型，然后求解微分方程.用模糊控制的方法解决这一问题就变得很容易.在RBFNNC中，我们选择偏转角 θ 和偏转角变化速度(角速度)$\Delta\theta$ 为 RBFNNC 的输入.摆的位置在垂线右方时，θ 取正值，反之，θ 取负值.当摆从垂线右方偏落时，$\Delta\theta$ 为正值，反之，$\Delta\theta$ 取负值.RBFNNC 的输出为作用力 F_{out}，F_{out} 的正方向向右(如图3所示).

整个系统限制在竖直平面内.

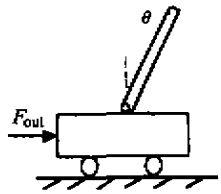


图 3　倒立摆小车平衡系统

Fig. 3　Cart-pole balancing system

3　RBFNNC 中的规则自动生成算法(Algorithm of automatic rule generation in RBFNNC)

以倒立摆小车平衡系统为实例,系统的状态变量偏转角 θ 和角速度 $\Delta\theta$ 为网络的输入,网络的输出为力 F.规则自动生成分两步进行:

第一步:删除结论与预期结论互相矛盾的规则.

1) 构造一个如图 2 所示的完全模糊神经网络,所谓"完全"网络是指包含所有可能的规则在内的网络.为了便于表达规则自动生成的过程,我们略去图 2 中的输入层和输出层,仅保留前提语言变量层、规则层和结论语言变量层,改画成如图 7(a) 所示的网络.

2) 读入系统状态变量 θ 和 $\Delta\theta$,模糊神经网络控制模块通过模糊推理得到每条规则 R_r 对总输出 F 的贡献 $Z_r = \mu^{-1}_{\Psi(SC,SL,SR)}(\omega_r)$ 和该时刻的总输出 F;

3) 增强因子评估模块通过一组预先存储的模糊规则,以读入的系统状态变量 θ 和 $\Delta\theta$ 为输入,推理得到该时刻的系统模糊误差 E,并根据式(4)计算出增强因子 v;

4) 确定网络在现状态下输出值 F_{opt} 应具有的最佳值的符号 $\mathrm{sgn}(F_{opt})$(最佳值的确切值不可能知道,但其在现状态下应具有的最佳符号值是可以预测的);

5) 对所有规则 R_r 逐一判断,删除激活强度小于某一阈值(可根据实际情况确定)的规则,删除 $\mathrm{sgn}(Z_r) \neq \mathrm{sgn}(F_{opt})$ 的规则;

6) 基于 F 和 v,信号统计处理模块计算出最后的作用力 F_{out} 并馈至倒立摆系统,从而获得新的系统状态 θ 和 $\Delta\theta$;

7) 返回 2) 直至满足迭代次数为止.

经过第一步筛选后,规则将能输出正确的符号,但不少规则还具有相同的前提(可参阅图 7(b)),需要通过第二步进行优化处理.

第二步:优化前提相同的规则组.

假设经第一步筛选后剩下 R 条规则,将其分成 m 组 $R^1, R^2, R^3, \cdots, R^m$,同一组规则 $R^i (1 \leqslant i \leqslant m)$ 都具有相同的前提.R^i_j 表示 $R^i (1 \leqslant j \leqslant l_i)$ 中的某一条规则,l_i 是每组的总规则数,每组中规则的总数不一定相等.具体步骤如下:

1) 获得系统状态 θ 和 $\Delta\theta$;

2) 在前提相同的每个规则组 R^i 中,随机选择 1 条规则 R^i_j,m 组共选择 m 条规则;

3) 模糊神经网络控制模块计算每条规则的输出 $Z^i_j = \mu^{-1}_{\Psi(SC,SL,SR)}(\omega^i_j)$ 和 m 条规则的总输出 F:

4) 增强因子评估模块计算系统模糊误差 E 和增强因子 v;

5) 按下式计算由所选各条规则产生的误差 $ERR_{R^i_j}$:

$$ERR_{R^i_j} = \frac{\omega_{R^i_j}}{\sum_{k=1}^{m} \omega_{R^k_j}} \cdot E \cdot \left(1 - \frac{|F - Z^i_j|}{F_{max} - F_{min}}\right). \quad (5)$$

$\omega_{R^k_j}$ 表示各条规则 $R^k_j (k = 1, \cdots, m)$ 的激活强度,Z^i_j 为第 R^i_j 条规则的输出,E 为系统模糊误差,F_{max} 和 F_{min} 分别为输出 F 可能的最大和最小值.显然,误差 $ERR_{R^i_j}$ 主要由归一化的激活强度、整个系统的误差和一个比例因子决定,该因子与整个网络的输出及各条规则输出之差的归一化值有关;

6) 累计每次迭代后的各条规则 R^i_j 的 $ERR_{R^i_j}$;

7) 基于 F 和 v,信号统计处理模块计算出最后的作用力 F_{out} 并馈至倒立摆系统,从而获得新的系统状态 θ 和 $\Delta\theta$;

8) 返回 1) 直至满足迭代次数为止.

最后在前提相同的规则组中仅保留累计误差 $ERR_{R^i_j}$ 最小的规则,由此得到一组优化的控制规则(可参阅图 7(c)).

4　RBFNNC 的仿真结果(Simulation result of RBFNNC)

定义偏转角 θ、角速度 $\Delta\theta$ 和力 F 的隶属函数如图 4 和图 5,定义系统模糊误差 E 的隶属函数如图 6.图 7 为规则自动生成中模糊神经网络拓扑结构的变化情况.图 7(a)是学习前包含所有规则的"完全"网络;图 7(b)是经过第一步学习删除结论互相矛盾的规则后的模糊神经网络;图 7(c)是学习结束后包含有效规则的网络.最后生成的规则可立即用于对倒立摆小车平衡系统的仿真控制以验证规则的有效性.

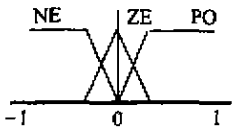


图 4 θ和$\Delta\theta$的隶属函数

Fig. 4 Membership function of θ and $\Delta\theta$

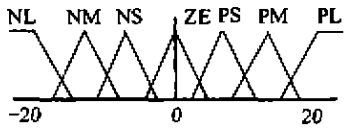


图 5 作用力 F的隶属函数

Fig. 5 Membership function of action F

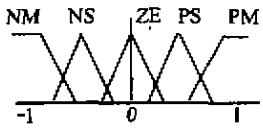
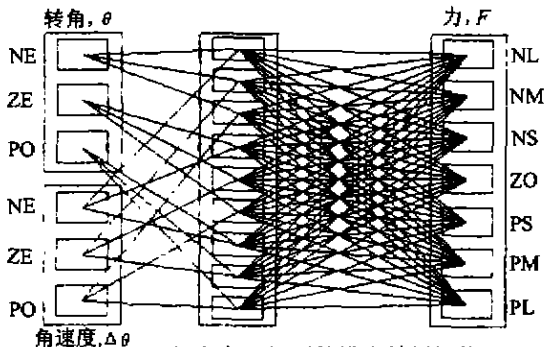
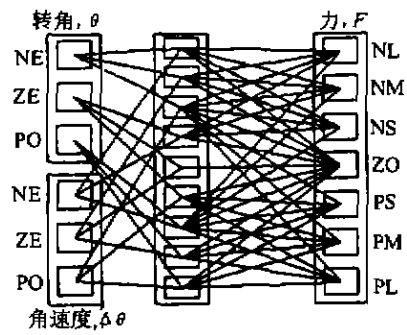


图 6 误差 E的隶属函数

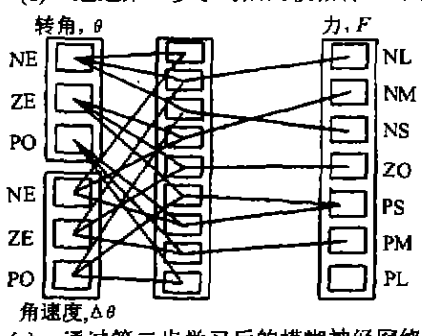
Fig. 6 Membership function of error E



(a) 包含全部规则的模糊神经网络



(b) 通过第一步学习后的模糊神经网络



(c) 通过第二步学习后的模糊神经网络

图 7 规则自动生成中模糊神经网络拓扑结构的变化情况

Fig. 7 Change process of topology during automatic rule generation

E	$\dot{\theta}$ \ θ	NE	ZE	PO
	NE	NM	NS	ZE
	ZE	NS	ZE	PS
	PO	ZE	PS	PM

图 8 一组用于推理获得误差 E的模糊规则

Fig. 8 A set of rules used to infer fuzzy error E

F	$\dot{\theta}$ \ θ	NE	ZE	PO
	NE		NM	PS
	ZE	NL	ZE	PM
	PO	NS	PS	

(a)

F	$\dot{\theta}$ \ θ	NE	ZE	PO
	NE		NL	ZE
	ZE	NL	ZE	PL
	PO	NL	PM	

(b)

F	$\dot{\theta}$ \ θ	NE	ZE	PO
	NE			PM
	ZE	NL	ZE	PL
	PO	NS	PL	

(c)

F	$\dot{\theta}$ \ θ	NE	ZE	PO
	NE		NL	PL
	ZE	NL	ZE	PL
	PO	NM		

(d)

图 9 由 RBFNNC规则自动生成程序产生的几组模糊控制规则

Fig. 9 Several groups of rules generated by RBFNNC

在进行规则自动生成仿真实验时，我们把图 8所示的一组模糊规则用于增强因子评估模块，用其推理获得系统模糊误差 E，进而计算出增强因子 v. 选择不同的初始角和倒立摆小车的其他参数进行仿真实验. 实验表明，规则自动生成的成功率约在80%以上，这主要与倒立摆小车的参数选择有关. 在成功的实验中，RBFNNC 能很好地使小车上的倒立摆按所要求的精度平衡在垂线附近，而且小车也非常接近中心位置. 对那些控制效果不怎么好的规则，如果辅以隶属函数调整功能，则系统同样可运行得很好. 图 9 为一组通过学习并经过检验获得的有效规则，倒立摆的初始角分别设为 $\pm 30°$，在给定的目标误差范围内(如取 $\pm 10°$)，规则生成的第一步一般只要迭代 200 次左右即可达到目标要求，第二步则要迭代 500 次左右. 图 9 中(a)规则组对应的是图 7中(c)的结果. 可以看出，用模糊控制有时仅用几条规则就能获得很好的控制效果.

参考文献(References)

[1] Patrikar A and Provence J. Control of dynamic system using fuzzy

(下转第 248 页)

控制，$t \geq T$ 时为线性控制，$D(z)$ 为线性控制器：

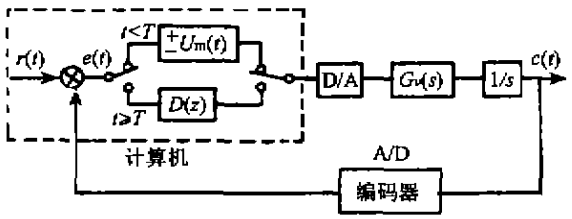


图 5 最速控制框图

Fig. 5 Block diagram of time-optimal control

表 2 计算结果

Table 2 Computational results

$x(0)$ (rad)	$y(0)$ (rad/s)	$z(0)$ (rad/s²)	t_1 (s)	t_2 (s)	t_3 (s)
0.4	0	0	0.67015	0.00854	0.00506
0.39	0	0	0.65348	0.00854	0.00506
0.39	0.03	0	0.68736	0.00807	0.00524
0.4	−0.03	0	0.63869	0.00902	0.00488
−0.2	0.06	0	0.30703	0.00951	0.00470
−0.19	0.09	0	0.27955	0.01000	0.00452
−0.21	0.03	0	0.33710	0.00902	0.00488
−0.21	0.09	0	0.30854	0.01000	0.00452
0.4	0.1	−0.04	0.80210	0.00700	0.00566
0.3	0.03	0.1	0.52947	0.00807	0.00524
0.3	−0.03	−0.5	0.47990	0.00902	0.00488
−0.25	0.06	−0.1	0.38279	0.00951	0.00470
−0.25	−0.06	0.5	0.46567	0.00761	0.00542

设被控对象 $G_v(s)$ 传递函数为：

$$G_v(s) = \frac{K_v\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{0.084 \times 128.805^2}{s^2 + 2 \times 0.65 \times 128.805 s + 128.805^2},$$

其中最大转速

$$\omega_{max} = 0.6 \text{ rad/s},$$

则

$$| U_m | = \omega_{max} = 0.6 \text{ rad/s}.$$

选取初始状态空间中若干点，计算其 t_1，t_2 和 t_3 值，结果如表 2.

文中取 $\mu_k = 0.1$，计算精度 ε 值为 1.0×10^{-5}. 按表 2 数据，经仿真[1]及实验[4]表明，采用此方法计算出的最速控制时间是完全正确的，获得了预期结果.

参考文献(References)

[1] Du Qinjian and Ma Jiaguang. A time-optimal control method switched by time [J]. Opto-Electronic Engineering, 1998, 25(5): 30 – 36 (in Chinese)

[2] Zheng Yong and Li Ruitang. The time-optimal control of servomechanisms [J]. Control Theory and Applications, 1989, 6(4): 57 – 63(in Chinese)

[3] Hu Zuchi and Lin Yuanqu. Numerical Analysis [M]. Beijing: Higher Education Press, 1987 (in Chinese)

[4] Du Qinjian. Experimental application of time-optimal control method switched by time [J]. Opto-Electronic Engineering, 1999, 26(6): 51 – 54 (in Chinese)

本文作者简介

杜勤德 1966年生，副教授，1988年北京航空航天大学自动控制系学士毕业，1991年中国科学院光电技术研究所硕士毕业，现于四川大学电气信息学院任教，主要从事计算机控制理论及应用方面的科研及教学工作.

(上接第 244 页)

logic and neural networks [J]. International Journal of Intelligent Systems, 1993, 8: 727 – 748

[2] Kosko B. Neural Network and Fuzzy Systems [M]. Englewood Cliffs, NJ: Prentice-Hall, 1991

[3] Berenji H R and Khedbar P. Learning and tuning fuzzy logic controllers through reinforcements [J]. IEEE Transactions on Neural Networks, 1992, 3(5): 724 – 740

[4] Nauck D and Kruse R. NEFCON-I: An X-window based simulator for neural fuzzy controllers [A]. Proc. IEEE Int. Conf. Neural Networks 1994 at IEEE World Congress on Computational Intelligence [C], Orlanda, 1994

[5] Nauck D and Kruse R. A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation [A]. Proc. IEEE Int. Conf. on Neural Networks [C], San Francisco, 1993, 1022 – 1027

本文作者简介

吴敬锋 1945年生，教授，博士生导师，1970年毕业于中国科技大学并在该校任教，1998年调任上海大学计算机学院常务副院长，先后主持完成了国家自然科学基金重大项目子课题、国家地震局"八五"重点项目以及第三世界科学院国际合作项目等，获安徽省1998年度和国家地震局1999年度科技进步二等奖，目前主要兴趣是：模糊控制，模糊神经控制，专家系统以及智能实时系统等.

傅忠谦 1959年生，副教授，1982年毕业于中国科技大学，现为中国科技大学电子科学与技术系微机教研室副主任，目前从事模糊控制，模糊神经控制和计算机网络等研究.