

⑮
89-95

基于神经网络的二阶非线性系统的自适应控制

A NEURAL NETWORK FOR THE ADAPTIVE CONTROL
OF A 2 ORDER NONLINEAR SYSTEM

汪庭凯 秦安松 黄赛男
Wang Tingkai Qin Ansong Huang Saninan
(重庆大学自动化系)

TP273.2

摘要 本文采用神经网络对二阶非线性系统进行特征辨识,并将其用于自适应控制,从而得到了一种新型的自适应控制方法.仿真实验表明该方法具有很强的学习能力和自适应性。

关键词 神经网络;反向传播法;自适应控制;特征辨识

非线性系统

中国图书资料分类法分类号 TP18;TP27

ABSTRACT This paper presents a new method of self tuning adaptive control which adopts a neural network to identify the characteristics of a second order nonlinear system, which is in turn applied to adaptive control. The simulations show that the method has good capability of study and adaptation.

KEY WORDS neural networks, back-propagation algorithm, adaptive control, characteristic identification

0 引 言

自适应控制技术为解决由于参数变化而引起的模型不确定性问题提供了有力的工具。自适应控制器的设计,无论是自校正控制还是模型参考自适应控制,都是在对系统参数的辨识,或者在对系统输出估计的基础上进行控制的。对于传统的辨识或估计技术,主要存在着如下问题:(1)要求较多的先验知识,而这些先验知识在实际系统中往往很难获取;(2)系统的非线性问题一般是对非线性元素进行线性化处理来解决的,但这只能处理一些特殊的非线性元素;(3)结构复杂系统的辨识计算相当费时,使得实现快速系统的实时控制存在一定的困难。近年来,人工智能,特别是神经网络的研究,为解决这些问题提供了较有效的方法^[1]。

* 收文日期 1992-03-19

本文的主要内容曾在电气自动化控制理论及应用会议(1991)上宣读
国家自然科学基金资助课题。

神经网络是由神经元相互连接而成的非线性动态系统^[2]。这种并行分布式处理系统能够通过动态学习不断地改善其性能^[1]。Nielsen 的研究已经证明^[3]，至少具有一个隐含层的神经网络能够模拟任意函数。此外，神经网络具有很强的实时性，在学习完成之后，网络的输入输出呈直觉匹配状态。由于神经网络的这些优点，近年来各国学者对神经网络用于非线性系统的控制进行了探讨，提出了一些较为有效的方法^[4~6]。

本文在此基础上对二阶非线性控制系统进行了研究，提出一种将反向传播神经网络与自适应控制技术相结合的控制方法。该方法采用神经网络来辨识系统的非线性特性，并以此制定自适应控制策略，从而获得了较好的控制效果。

1 反向传播神经网络及其学习算法

一个反向传播神经网络的基本结构如图1所示。它是由一个输入层、一个输出层和至少一个隐含层组成的网络。同一层神经元之间无相互连接。神经元是网络中的基本处理单元，其基本定义如下：

$$Y = f\left(\sum_{i=1}^n W_i X_i - Q\right) \quad (1)$$

式中 Y 为神经元的输出， X_i 为该神经元的第 i 个输入， W_i 为相应的连接权系数， Q 为该神经元的阈值， f 为一单调递增的非线性函数。

没有隐含层的简单神经网络，虽有多种有效的学习方法，容易训练，但功能不强，不能求解高阶谓词问题。1986年 Rumelbart, Hinton 和 Williams 提出的“反向传播算法”为多层神经网络的训练提供了有效的手段。这一算法的主要思想是：^[7]当有信息输入时，输入信息经输入层、隐含层及输出层作用后得到

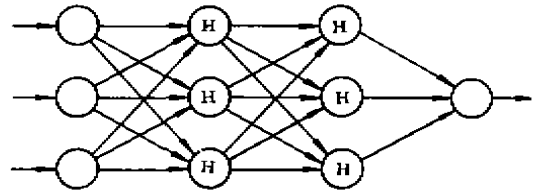


图1 神经网络的基本结构

神经网络的输出值，将该值与期望的输出进行比较，若有误差 E 存在，则利用阶层型网络的特点，从输出层开始反向地一层一层地确定 E 对每个节点输出的偏导数，并利用梯度下降法修改各个连接权系数。

假定有一个 M 层反向传播神经网络，把第 n 层的第 i 个节点记为 O_i^n ， O_i^n 的输入是 X_i^n ，输出是 Z_i^n ，输入输出函数记为 f ，即 $Z_i^n = f(X_i^n)$ ，从 O_i^n 到 O_j^{n+1} 的连接权为 W_{ij}^{n+1} ， Q_i^n 为 O_i^n 的阈值。

对于输入模式 P ，如果输出层 M 的第 j 个节点的实际输出为 Z_j^M ，理想的输出为 d_j ，则误差为：

$$E = \frac{1}{2} \sum (d_j - Z_j^M)^2 \quad (2)$$

为了减少误差 E ，根据梯度下降法， W_{ij}^{n+1} 的修正量为：

$$\Delta W_{ij}^{n+1} = -\epsilon \frac{\partial E}{\partial W_{ij}^{n+1}} \quad (3)$$

式中 ϵ 为修正系数。将上式的偏导数展开，有

$$\frac{\partial E}{\partial W_{ij}^{n+1}} = \frac{\partial E}{\partial X_j^{n+1}} \frac{\partial X_j^{n+1}}{\partial W_{ij}^{n+1}} = \frac{\partial E}{\partial X_j^{n+1}} Z_i^n \quad (4)$$

当 $n = M$ 时

$$\frac{\partial E}{\partial X_j^M} = \frac{\partial E}{\partial X_j^M} = \frac{\partial E}{\partial Z_j^M} \frac{\partial Z_j^M}{\partial X_j^M} = - (d_j - Z_j^M) \cdot f'(X_j^M) \quad (5)$$

当 $n \neq M$ 时

$$\frac{\partial E}{\partial X_j^n} = \sum_i \frac{\partial E}{\partial X_i^{n+1}} \frac{\partial X_i^{n+1}}{\partial Z_j^n} \frac{\partial Z_j^n}{\partial X_j^n} = \sum_i \frac{\partial E}{\partial X_i^{n+1}} \cdot W_{ij}^{n+1} \cdot f'(x_j^n) \quad (6)$$

将以上公式加以整理,即可得到连接权 W_{ij}^{n+1} 的修正量公式:

$$\Delta W_{ij}^{n+1} = - e_i^n Z_j^{n+1} \quad (7)$$

式中 $e_i^n = \frac{\partial E}{\partial X_i^n}$, 当 $n = M$ 时

$$e_j^M = (Z_j^M - d_j) \cdot f'(X_j^M) \quad (8)$$

当 $n \neq M$ 时,

$$e_i^n = f'(X_i^n) \sum_j W_{ij}^{n+1} e_j^{n+1}, \quad n = M - 1, \dots, 2 \quad (9)$$

式中 e_i^n 可以看作是一般化的误差信号,它的计算是从输出层 ($n = M$) 开始,到 $n = 2$ 递推进行的,就好象是把输出层所产生的误差信号 e_j^M ,以相反的方向一层一层地传到输入层,故这种方法称为误差反向传播算法。

2 基于神经网络的二阶非线性自适应控制系统的结构及学习算法

许多实际的生产过程都可以用一个二阶非线性系统来描述。一类二阶非线性对象可以用下式表示:

$$Y_{k+1} = f_1(Y_k) + f_2(Y_{k-1}) + b_0 u_k, \quad b_0 \neq 0 \quad (10)$$

$f_1(\cdot)$ 和 $f_2(\cdot)$ 均为未知的非线性函数, b_0 为未知的非零常数或函数。本文采用如图(2)所示的神经网络来进行特征识别。

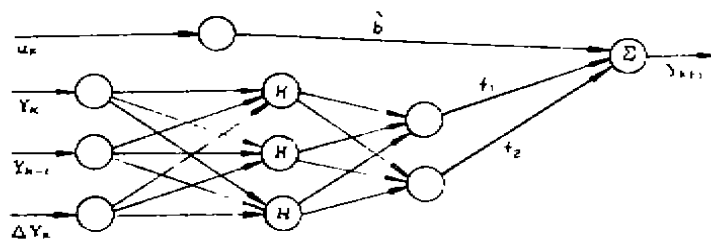


图2 神经网络结构图

该神经网络的估计模型为:

$$\hat{Y}_{k+1} = \hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k)) + \hat{b}(k) \cdot u_k, \hat{b}(k) \neq 0 \quad (11)$$

式中 $\hat{f}_1(Y_k, W(k))$ 和 $\hat{f}_2(Y_{k-1}, W(k))$ 是 $f_1(\cdot)$ 和 $f_2(\cdot)$ 的估计值, $\hat{b}(k)$ 为 b_0 的估计值, $W(k)$ 为网络中各神经元之间的连接权系数, $\hat{f}_1(\cdot)$, $\hat{f}_2(\cdot)$ 及 $\hat{b}(k)$ 又称为对象的特征量。与之相应的控制器为如下的变比例调节

$$u_k = \frac{d_{k+1} - \hat{f}_1(Y_k, W(k)) - \hat{f}_2(Y_{k-1}, W(k))}{\hat{b}(k)} \quad (12)$$

式中 d_{k+1} 为控制的目标值。由此可设计出自适应控制系统的总体结构如图3所示。

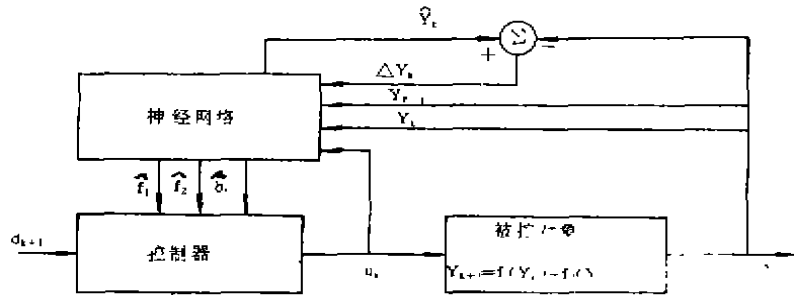


图3 控制系统结构图

由(10)、(11)、(12)式得:

$$Y_{k+1} = f_1(Y_k) + f_2(Y_{k-1}) + b_0 \frac{d_{k+1} - \hat{f}_1(Y_k, W(k)) - \hat{f}_2(Y_{k-1}, W(k))}{\hat{b}(k)} \quad (13)$$

即对象的输出值依赖于神经网络的权值,而神经网络的输出 $\hat{Y}_{k+1} = d_{k+1}$ 与神经网络的权值无关。故误差 $e_{k+1} = d_{k+1} - Y_{k+1}$ 。定义误差能量函数为:

$$E_k = \frac{1}{2} e_{k+1}^2 = \frac{1}{2} (d_{k+1} - Y_{k+1})^2 \quad (14)$$

根据梯度下降法调节 $\hat{W}(k)$ 和 $\hat{b}(k)$ 可使 E_k 降低。 E_k 对 $\hat{b}(k)$ 的偏导数为:

$$\begin{aligned} \frac{\partial E}{\partial \hat{b}(k)} &= \frac{b_0}{\hat{b}(k)} \left[\frac{d_{k+1} - \hat{f}_1(Y_k, W(k)) - \hat{f}_2(Y_{k-1}, W(k))}{\hat{b}(k)} \right] e_{k+1} \\ &= \frac{b_0}{\hat{b}(k)} u_k \cdot e_{k+1} \end{aligned} \quad (15)$$

故 $\hat{b}(k)$ 的修正量公式为:

$$\hat{b}(k+1) = \hat{b}(k) - \eta_k \frac{M}{\hat{b}(k)} u_k e_{k+1} \quad (16)$$

E_k 对 $W(k)$ 的偏导数为:

$$\frac{\partial E_k}{\partial W^i(k)} = \frac{b_0}{\hat{b}(k)} \cdot \frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^i(k)} e_{k+1} \quad \forall W^i(k) \in W(k) \quad (17)$$

故权矢量 $W(k)$ 的修正量公式为:

$$W^i(k+1) = W^i(k) - \epsilon_k \frac{M}{\hat{b}(k)} \frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^i(k)} \cdot e_{k+1} \quad \forall W^i(k) \in W(k) \quad (18)$$

其中 η_k 和 ϵ_k 均为第 k 步距上权系数的学习速率, M 为计算 b_0 的初值,按照第二节所介绍的反向传播算法,即可从 $\hat{f}_1(\cdot)$, $\hat{f}_2(\cdot)$ 所在的层开始反向地确定(16)式对每个节点连接权值的偏导数,从而修正其连接权系数。

3 学习算法对系统误差的影响

关于学习算法对系统误差的影响,为简单起见,我们对神经元不带阈值的情况进行讨

论。对能量函数:

$$E_k = \frac{1}{2} e_{k+1}^2 = \frac{1}{2} (d_{k+1} - Y_{k+1})^2$$

设 $\frac{dE_k}{dt}$ 为能量函数在学习系数 $(\hat{b}(k), W(k))$ 上的方向导数, 则:

$$\frac{dE_k}{dt} = e_{k+1} \frac{de_{k+1}}{dt} = e_{k+1} \left(\frac{\partial e_{k+1}}{\partial \hat{b}(k)} \frac{\partial \hat{b}(k)}{\partial t} + \sum_j \frac{\partial e_{k+1}}{\partial W^j(k)} \frac{\partial W^j(k)}{\partial t} \right) \quad (19)$$

由(15)、(16)两式有:

$$\frac{\partial e_{k+1}}{\partial \hat{b}(k)} = \frac{b_0}{b(k)} u_k \quad (20)$$

$$\frac{\partial \hat{b}(k)}{\partial t} = \eta_k \frac{M}{b(k)} u_k \cdot e_{k+1} \quad (21)$$

由(17)、(18)两式有:

$$\sum_j \frac{\partial e_{k+1}}{\partial W^j(k)} = \sum_j \frac{b_0}{b(k)} \frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^j(k)} e_{k+1} \quad (22)$$

$$\frac{\partial W^j(k)}{\partial t} = -\varepsilon \frac{M}{b(k)} \frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^j(k)} \quad (23)$$

所以

$$\begin{aligned} \frac{dE_k}{dt} &= -\frac{b_0}{(b(k))^2} M \eta_k u_k^2 e_{k+1}^2 - \frac{b_0}{(b(k))^2} M \varepsilon_k e_{k+1}^2 \sum_j \left[\frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^j(k)} \right]^2 \\ &= -\frac{b_0 M}{[b(k)]^2} e_{k+1}^2 \left[\eta_k u_k^2 + \varepsilon_k \sum_j \left[\frac{\partial [\hat{f}_1(Y_k, W(k)) + \hat{f}_2(Y_{k-1}, W(k))]}{\partial W^j(k)} \right]^2 \right] \end{aligned} \quad (24)$$

若取 $M, b_0 > 0, \eta_k > 0, \varepsilon_k > 0$, 则有:

$$\frac{dE_k}{dt} < 0 \quad (25)$$

即系统误差的平方将随着学习算法的学习而逐渐减小, 这说明学习自身是收敛的。

若神经网络采用的是具有阈值的神经元, 其阈值学习的基本原理与权值的调整完全相同, 也是采用反向传播学习方法。根据误差能量函数(14)式, 具体的调整公式如下:

$$\Delta H^j(k) = r_k \frac{\partial E_k}{\partial H^j(k)} \quad (26)$$

$$H^j(k+1) = H^j(k) - r_k \frac{\partial E_k}{\partial H^j(k)} \quad \forall H^j(k) \in H(k) \quad (27)$$

$$\frac{\partial E_k}{\partial H^j(k)} = \frac{b_0}{b(k)} \frac{\partial [\hat{f}_1(Y_k, H(k)) + \hat{f}_2(Y_{k-1}, H(k))]}{\partial H^j(k)} \cdot e_{k+1} \quad \forall H^j(k) \in H(k) \quad (28)$$

其中 $H(k)$ 为神经网络第 k 步所有阈值的集合, r_k 是第 k 步神经元阈值的学习率, 可根据下式参考选值:

$$r_k = u_k \frac{1}{\sum_j \left[\frac{\partial [\hat{f}_1(Y_k, H(k)) + \hat{f}_2(Y_{k-1}, H(k))]}{\partial H^j(k)} \right]^2} \quad (29)$$

加入阈值的学习后, 神经网络的学习就不仅仅是连接权系数的学习, 同时还包括了阈值的学习。通过权值, 阈值的不断学习调整, 使网络逐渐达到所期望的指标。

4 仿真实验

笔者对各种非线性对象,采用 Basic 语言,在 IBM PC/XT 微型计算机上进行了大量的仿真实验。限于篇幅,仅举以下二例。例 1 对象为 $Y_{k+1} = f_1(Y_k) + f_2(Y_{k-1}) + b_0 u_k$

其中 $b_0 = 1$

$$f_1(Y_k) = \begin{cases} 1 & Y_k > 2 \\ Y_k / 2 & 1 \leq Y_k \leq 2 \\ 0 & Y_k < 1 \end{cases}$$

$$f_2 = (Y_{k-1}) = \frac{e^{Y_{k-1}} - e^{-Y_{k-1}}}{e^{Y_{k-1}} + e^{-Y_{k-1}}}$$

期望跟踪的目标 d_k 为一阶跃函数,阶跃值的大小为 2.5。采用图 2 所示的神经网络结构来识别该对象的特征量。神经元的输入输出函数采用 S 型函数。各神经元之间的连接权系数的初值在 $-25 \sim 25$ 之间任意选择,学习率的初值在 $0.1 \sim 1$ 之间选择。该控制系统的第一次阶跃响应曲线与网络经过多次学习后的系统阶跃响应曲线如图 4 所示。不难看出,网络经过多次学习后,控制品质得到了很大的改善,几乎没有超调。



图 4 网络第一次学习时系统响应曲线与多次学习后的响应曲线

例 2 对象的形式仍为 $Y_{k+1} = f_1(Y_k) + f_2(Y_{k-1}) + b_0 u_k$ 其中 $b_0 = 1$

$$f_1(Y_k) = \sin(Y_k)$$

$$f_2(Y_{k-1}) = \frac{e^{Y_{k-1}} - e^{-Y_{k-1}}}{e^{Y_{k-1}} + e^{-Y_{k-1}}}$$

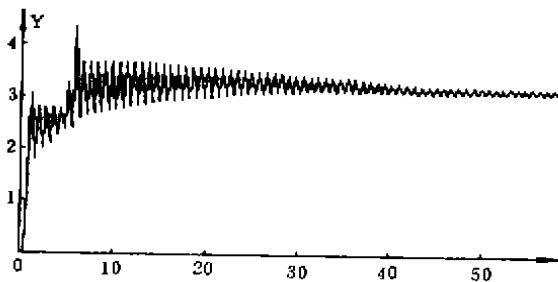


图 5(a) 网络第一次学习时系统响应曲线

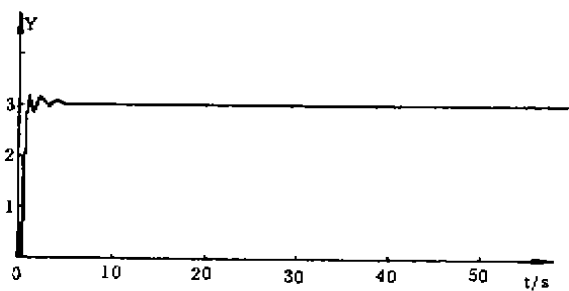


图 5(b) 网络经多次学习后的系统响应曲线

期望跟踪的目标 d_k 仍为阶跃函数,阶跃值 $d_k=3$,与之相应的神经网络结构及初始值的设置与例一相同。图5(a)、(b)分别给出了该系统的第一次阶跃响应曲线和经过多次学习后的阶跃响应曲线。不难看出,该系统具有很的学习能力和适应能力。

5 结束语

运用神经网络对系统的非线性特征进行辨识为自适应控制技术解决非线性系统问题提供了一条新的途径。本文对神经网络在控制领域的应用进行了初步的探索,取得了较为满意的结果。这说明神经网络在自动控制领域的应用具有很大的潜力。当然,也存在一些问题尚待解决,如网络结构的设计准则。这些问题的解决必将更好地推动神经网络在控制领域的应用。

参 考 文 献

- 1 Behnam Bavarian. Introduction of Neural Network for Intelligent control. IEEE Control Systems Magazine, 1988, 8(2)
- 2 Minsky M, Papert S. Perceptron. Cambridge, MA, MIT Press
- 3 R. Hecht-Nielsen. Theory of the Back Propagation Neural Network Proc. IEEE Int. Conf. on Neural Networks, 1989 593~605
- 4 D. Psaltis, A. Sideris, A. A. Yamanura, "A Multilayered Neural Network Controller", IEEE Control Systems Magazine, 1988, 8(2)
- 5 W. Li & J. E. Slotine, "Neural Network Control of Unknown Nonlinear Systems", Proc. American Control Conference, 1989, 1136~1141
- 6 Fu-chuang Chen. Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control, IEEE Control Systems Magazine, 1990, 10(3)
- 7 张鸿宾. 人工神经网络的学习算法(1). 计算机科学, 1990, (2)