用连续回归神经网络求解泛函极值问题 [1]

刘贺平　　张兰玲　　孙一康

(北京科技大学自动化系 北京 100083)

摘　要　针对信息科学和控制理论中经常涉及的一类泛函极值问题，提出基于连续回归神经网络的求解方法．推导了求解泛函的连续 BPTT 算法，进而对该算法进行改进，得出一种在线学习算法，为并行实现打下了基础．

关键词　泛函极值，连续回归神经网络，BPTT 算法，在线学习算法

中图号　TN-052, O177

1 引　言

在信息科学和控制理论中，经常要遇到求解泛函 $J=\varphi(t_f,y(t_f))+\int_{t_0}^{t_f}F(\dot{y},y,t)\mathrm{d}t$ 在给定初值 $y(t_0)$ 条件下的极值问题．已有的求解方法中，一种方法是将该问题转化为微分方程的边值问题去解决，不仅需要较强的条件，而且求解比较困难，另一种方法是将待求轨迹 $y(t)$ 在有穷维空间离散化，然后用有穷维空间的优化方法直接求解泛函极值问题 [1]，这是一种离线的数值求解方法，因而难以满足实际应用中的实时性要求．

K.Funahashi 和 Y.Nakamura 从理论上证明了任何一个动态系统在有限时间内的轨迹都可由一个连续回归网络输出单元的内部状态来任意逼近 [2]．基于这一理论，构造一个连续回归神经网络，把待求泛函作为网络的训练函数，就可利用神经网络的学习能力，通过优化网络参数直接求解泛函极值问题．

连续型回归网络的轨迹学习算法主要有两类：一种是 BPTT(Back Propagation Through Time) 算法 [3-6]，另一种是实时回归学习 RTRL(Real Time Recurrent Learning) 算法 [7]．BPTT 算法的缺点在于学习过程的非因果性，即需求解一个给定终值的微分方程，因而训练过程不能在线进行．RTRL 算法虽然没有学习过程非因果性的缺点，但由于其计算量太大，若不并行实现学习算法，也难以满足实时性的要求．此外，这两种算法在推导过程中，期望轨迹均已知，网络准则函数的形式是 $J_1=\int_{t_0}^{t_f}F_1(y,t)\mathrm{d}t$，未涉及 $\dot{y}$ 项．本文研究的求解问题比 J_1 形式更具有一般性，在 J 中引入了 $\dot{y}$ 项，而且由于无期望轨迹可循，因此 J 的求解过程具有一定的难度和复杂性．针对泛函 J 的极值求解问题，我们推导了相应的 BPTT 算法，并对 BPTT 算法进行改进，提出用连续回归神经网络求解泛函极值问题的一种可并行实现的在线算法．

2 泛函极值问题的连续型 BPTT 算法

考虑泛函：

$$J=\varphi(t_f,y(t_f))+\int_{t_0}^{t_f}F(\dot{y},y,t)\mathrm{d}t \tag{1}$$

的极值问题．设所用网络的网络方程为

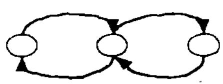
[1] 1998-11-30 收到， 1999-06-21 定稿

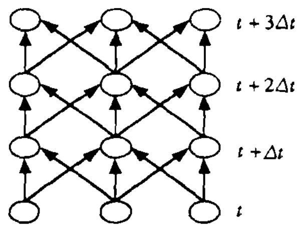
$$\dot{y} = f(y(t), w, I(t)), \tag{2}$$

其中，$y = [y_1, y_2, \cdots, y_n]^T$ 为神经元状态向量 (n 为神经元数)，$w = [w_{11}, \cdots, w_{1n}, w_{21}, \cdots, w_{mn}]^T$ 为要调节的参数矢量，$I = [I_1, \cdots, I_n]^T$ 为外部输入向量。给定训练函数 J，通过修正 w，调整 $f[y(t), w, I(t)]$，达到使 J 最小，从而确定极值曲线 $y(t)$。

BPTT 算法的基本思想是将网络随时间变化的状态沿时间展开为空间形式，从而使误差反传成为可能，如图 1 所示。

图 1(a) 为一个回归网络，图 1(b) 为网络沿时间展开为多层网络的形式。由图可见，回归网络可沿时间演化为一个多层前向网络，网络神经元状态的变化即沿“层”前传。





(a) (b)

图 1 回归网络结构及传播形式图

首先，将连续网络以步长 Δt 离散化，对该离散网络引入 BP 算法，再使 Δt 趋于零，求其极限值，即可获得连续型学习算法。

(2) 式中，$\dot{y}_i$ 可由下式近似求得

$$\dot{y}_i \approx [y_i(t+\Delta t) - y_i(t)]/\Delta t. \tag{3}$$

(2) 式可离散化为

$$y(t+\Delta t) = y(t) + \Delta t \cdot f(y(t), w, I). \tag{4}$$

(1) 式可离散化为

$$J = \varphi(t_f, y(t_f)) + \sum_{t=t_0}^{t_f-\Delta t} F\{\frac{1}{\Delta t}\cdot[y(t+\Delta t) - y(t)], y(t), t\}\cdot\Delta t. \tag{5}$$

在图 1 中，$t + i\Delta t$ 层单元输出设定为 $y(t+i\Delta t)$。当 t 和 $t+\Delta t$ 层之间的权值 w 有一微小变化 dw 时，其影响将通过 $y(t+\Delta t)$ 最终传播给 J，其值为 $dw^T\cdot[(\partial y(t+\Delta t)/\partial w]^T\cdot(\partial^+ J)/[\partial y(t+\Delta t)]$，$\partial^+$ 为有序偏导数 [4]。由于 J 是在 $[t_0, t_f]$ 上定义的，故有

$$\frac{\mathrm{d}J}{\mathrm{d}w} = \sum_{t=t_0}^{t_f-\Delta t}(\frac{\partial y(t+\Delta t)}{\partial w})^T\cdot\frac{\partial^+ J}{\partial y(t+\Delta t)}. \tag{6}$$

由 (4) 式，有

$$\partial y(t+\Delta t)/\partial w = \Delta t\partial f/\partial w. \tag{7}$$

令 $\lambda(t)=\partial^+J/\partial y(t)$ ，由链式规则得

$$\lambda(t)=\frac{\partial^+J}{\partial y(t)}=\frac{\partial J}{\partial y(t)}+(\frac{\partial y(t+\Delta t)}{\partial y(t)})^T\cdot\frac{\partial^+J}{\partial y(t+\Delta t)}=\frac{\partial J}{\partial y(t)}+(\frac{\partial y(t+\Delta t)}{\partial y(t)})^T\cdot\lambda(t+\Delta t). \quad (8)$$

由 (4) 式求出 $\partial y(t+\Delta t)/\partial y(t)$ ，由 (5) 式求出 $\partial J/\partial y(t)$ ，代入 (8) 式整理后可得

$$(\lambda(t)-\lambda(t+\Delta t))\cdot 1/\Delta t=(\partial f/\partial y)^T\lambda(t+\Delta t)+(\partial F/\partial y)|_t-(1/\Delta t)(\partial F/\partial\dot{y})|_t-(\partial F/\partial\dot{y})|_{t-\Delta t})$$

令 $\Delta t\to 0$ 得

$$-\dot{\lambda}(t)=(\frac{\partial f}{\partial y})^T\lambda(t)+\frac{\partial F}{\partial y}-\frac{d}{dt}(\frac{\partial F}{\partial\dot{y}}). \quad (9)$$

当 $t=t_f$ 时，由 (3) 式、 (5) 式可知

$$\lambda(t_f)=\partial\varphi/\partial y(t_f)+\partial F/\partial\dot{y}(t_f). \quad (10)$$

将 (7) 式及 $\lambda(t)=\partial^+J/\partial y(t)$ 代入 (6) 式，并令 $\Delta t\to 0$ 得

$$\mathrm{d}J/\mathrm{d}w=\int_{t_0}^{t_f}(\partial f/\partial w)^T\lambda(t)\mathrm{d}t. \quad (11)$$

同理

$$\mathrm{d}J/\mathrm{d}I=\int_{t_0}^{t_f}(\partial f/\partial I)^T\cdot\lambda(t)\mathrm{d}t. \quad (12)$$

对于有限时间区间的轨迹学习，通常认为神经网络可调参数的变化过程要比神经元状态的变化过程缓慢的多 [8] ，因此可以假设在 t_f-t_0 时间长度内， w 和 I 为常值，所以

$$w(k+1)=w(k)-\eta\mathrm{d}J/[\mathrm{d}w(k)], \quad (13)$$

$$I(k+1)=I(k)-\eta\mathrm{d}J/[\mathrm{d}I(k)], \quad (14)$$

其中， $w(k)$ 、 $I(k)$ 表示第 k 个学习周期内的权值与输入.

3 泛函极值问题的在线学习算法

上节给出算法的缺陷在于 (9) 、 (10) 式描述的协态方程是在已知终值条件 $\lambda(t_f)$ 的情况下，求解 $\lambda(t)$ 在 $[t_0,t_f]$ 的轨迹，是一个非因果过程。为此，我们对上节的算法进行改进，在 BPTT 算法的基础上提出泛函极值问题的在线学习算法.

引理 1 设 $\phi_Z(t,t_0)$ 和 $\phi_\lambda(t,t_0)$ 分别是方程 $\dot{Z}=A(t)Z$ 和 $\dot{\lambda}=-A^T(t)\lambda$ 的状态转移矩阵，则有 $\phi_Z(t,t_0)=\phi_\lambda^{-T}(t,t_0)$ 。证明略.

为简化 (9) 式，定义

$$L=\frac{\partial f}{\partial y},\qquad e(t)=\frac{\mathrm{d}}{\mathrm{d}t}(\frac{\partial F}{\partial\dot{y}})-\frac{\partial F}{\partial y},$$

则 (9) 式成为

$$\dot{\lambda}(t) = -L^T\lambda(t) + e(t), \tag{15}$$

其解为

$$\lambda(t) = \phi_\lambda(t,t_0)\lambda(t_0) + \int_{t_0}^{t}\phi_\lambda(t,\tau)e(\tau)\mathrm{d}\tau,$$

其中 ϕ_λ 为 (15) 式的状态转移矩阵. 假设 $\overline{\lambda}(t)$ 为 $\lambda(t_0)=0$ 时 (15) 式的状态轨迹，则对于任意的 $\lambda(t_0)$ ，其状态轨迹 $\lambda(t)$ 与 $\overline{\lambda}(t)$ 之间满足

$$\lambda(t) - \overline{\lambda}(t) = \phi_\lambda(t,t_0)\lambda(t_0). \tag{16}$$

当 $t=t_f$ 时， $\lambda(t_f) - \overline{\lambda}(t_f) = \phi_\lambda(t_f,t_0)\lambda(t_0)$ ，即

$$\lambda(t_0) = \phi_\lambda^{-1}(t_f,t_0)(\lambda(t_f) - \overline{\lambda}(t_f)). \tag{17}$$

下面推导权值的修正公式. 由 (11) 式和 (16) 式得

$$\frac{\mathrm{d}J}{\mathrm{d}w_{ij}} = \int_{t_0}^{t_f}(\frac{\partial f}{\partial w_{ij}})^T(\overline{\lambda}(t) + \phi_\lambda(t,t_0)\lambda(t_0)\mathrm{d}t.$$

令 $\theta_{ij}(t_f) = \int_{t_0}^{t_f}(\frac{\partial f}{\partial w_{ij}})^T\phi_\lambda(t,t_0)\lambda(t_0)\mathrm{d}t$ ，则

$$\frac{\mathrm{d}J}{\mathrm{d}w_{ij}} = \int_{t_0}^{t_f}(\frac{\partial f}{\partial w_{ij}})^T\overline{\lambda}(t)\mathrm{d}t + \theta_{ij}(t_f). \tag{18}$$

令 $\Delta\lambda(t_f) = \lambda(t_f) - \overline{\lambda}(t_f)$ ，由 (17) 式得

$$\theta_{ij}(t_f) = \left[\int_{t_0}^{t_f}\phi_\lambda^{-T}(t_f,t)\frac{\partial f}{\partial w_{ij}}\mathrm{d}t\right]^T\Delta\lambda(t_f). \tag{19}$$

由引理 1 可知， (19) 式中的 $\int_{t_0}^{t_f}\phi_\lambda^{-T}(t_f,t)\frac{\partial f}{\partial w_{ij}}\mathrm{d}t$ 是下列方程:

$$\dot{Z}_{ij}(t) = L(t)Z_{ij}(t) + \partial f/\partial w_{ij} \tag{20}$$

满足 $Z_{ij}(t_0)=0$ ， $t=t_f$ 的解. 其中 $Z_{ij}(t) = [Z_{ij,1}(t),\cdots,Z_{ij,n}(t)]^T$($n$ 为神经元数) . 因此，(19) 式变为

$$\theta_{ij}(t_f) = Z_{ij}^T(t_f)\cdot\Delta\lambda(t_f). \tag{21}$$

同理可得

$$\frac{\mathrm{d}J}{\mathrm{d}I_i} = \int_{t_0}^{t_f}(\frac{\partial f}{\partial I_i})^T\cdot\overline{\lambda}(t)\mathrm{d}t + \gamma_i(t_f). \tag{22}$$

其中 $\gamma_i(t_f)$ 为

$$\gamma_i(t_f) = P_i^T(t_f)\cdot\Delta\lambda(t_f). \tag{23}$$

$P_i(t_f)$ 为下列方程:

$$\dot{P}_i(t) = L(t)P_i(t) + \partial f/\partial I_i \tag{24}$$

满足 $P_i(t_0) = 0$ ， $t = t_f$ 的解. 其中 $P_i(t) = [P_{i,1}(t), \cdots, P_{i,n}(t)]^T$ (n 为神经元数) .

由 (18)~(24) 式可知，从数值仿真的角度来看，对于每个可调参数，在线学习算法比原先的 BPTT 算法增加了 (20) 、 (24) 式的数值积分，因而增加了计算量. 但由于是因果系统，采用神经网络可以并行实现. 从这个角度看，在线学习算法比 BPTT 算法更为优越.

4 并行实现仿真结果

为了验证本文提出的实时算法的有效性，对下式:

$$J(x(t)) = [x(\pi/2) - 1]^2 + \int_0^{\pi/2} (\dot{x}^2 - x^2)\mathrm{d}t, \quad x(0) = 0$$

给出的泛函极值曲线求解问题进行了基于神经网络的并行实现仿真实验，选用由两个神经元构成的回归网络，网络方程为

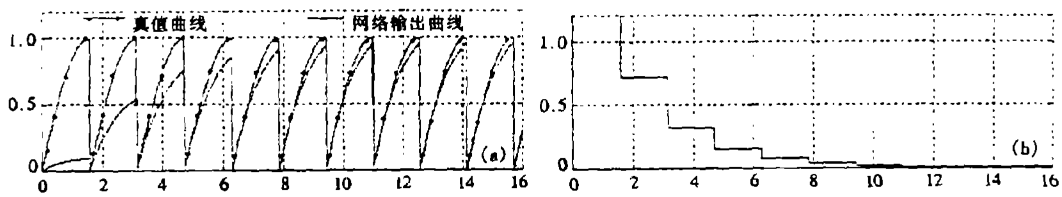
$$\dot{y}_i = -y_i + \sigma\left(\sum_{j \neq i} w_{ij} y_j\right) + I_i, \quad i = 1, 2,$$

其中 $\sigma(\cdot)$ 为 sigmoid 函数，初值设定为 $y_1(0) = x(0) = 0$, $y_2(0) = \dot{x}(0) =$ 任意， $I(0)$ 、 $w(0)$ 为 $[-0.5,0.5]$ 间的随机数，学习步长 $\eta = 1$. 当

$$(\|\mathrm{d}J/\mathrm{d}w\|_1 + \|\mathrm{d}J/\mathrm{d}I\|_1 < \varepsilon$$

时，学习过程终止. 其中 $\|\cdot\|$ 表示 l_1 范数， ε 为预先设置的小的正数.

仿真结果如图 2 所示，由图 2 可见，经过 8 个周期的学习，系统就输出了所需的极值曲线。



(a) 真值曲线与网络输出曲线

(b) $\left\|\frac{\mathrm{d}J}{\mathrm{d}w}\right\| + \left\|\frac{\mathrm{d}J}{\mathrm{d}I}\right\|_1$ 变化曲线

图 2 解泛函极值问题神经网络学习过程曲线

5 结 论

本文基于连续回归神经网络的动态逼近能力，提出泛函极值问题的神经网络求解方法，推导了相应的 BPTT 算法，并对该算法作进一步的改进，提出一种在线算法，为并行实现奠定了理论基础，使实时应用成为可能。仿真结果表明：该算法具有较理想的快速性和精度。

参 考 文 献

[1] 叶庆凯，郑应平编著. 变分法及其应用. 国防工业出版社，1991.

[2] Funahashi K, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks. Neural Network, 1993, 6: 801-806.

[3] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation. in Parallel Distributed Processing. Rumelhart, D.E., McClelland, J.L. Eds., Cambridge,MA: M.I.T Press, 1986.

[4] Werbos P J. Backpropagation through time: What it does and how to do it, Proc. of IEEE, 1990, 78(10): 1550-1560.

[5] Pearlmutter B A. Learning state space trajectories in recurrent neural network, IEEE Proc. IJCNN, 1989, 2: 365-372.

[6] Sato M. A Learning algorithm to teach spatiotemporal patterns to recurrent neural networks. Biological Cybernetics, 1990, 62: 259-263.

[7] Williams R J, Zipser D. A Learning algorithm for continually running fully recurrent neural networks. Neural Computation. 1989, 1(2): 270-280.

[8] Baldi P. Gradient learning algorithm overview: A general dynamical systems perspective. IEEE Trans. on Neural Networks. 1995. 6(1): 182-195.

A CONTINUOUS TIME RECURRENT NEURAL NETWORK BASED METHOD TO SOLVE FUNCTIONAL MINIMIZATION PROBLEM

Liu Heping Zhang Lanling Sun Yikang

(*Department of Automation, University of Science and Technology Beijing, Beijing* 100083)

Abstract In this paper, the continuous time recurrent neural network is proposed to solve the functional minimization problem, which is often involved in estimation and control. At first, the continuous time BPTT algorithm corresponding to the problem is presented. Then, an on-line algorithm based on the amendments of the BPTT algorithm is discussed. This on-line algorithm paves the way for parallel realization.

Key words Functional minimization, Continuous time recurrent neural network, BPTT algorithm, On-line learning algorithm

刘贺平： 男，1951 年生，博士后，副教授，目前主要研究方向为自适应控制，智能控制，人工神经网络及应用等.

张兰玲： 女，1970 年生，博士，工程师，现从事信息科学理论及计算机应用方面的研究.

孙一康： 男，1932 年生，教授，博士生导师，国家教委科技组自动控制组成员，目前主要从事轧钢自动化理论研究和技术改造工作.