第49卷第2期
2009年3月

大连理工大学学报
Journal of Dalian University of Technology

Vol. 49, No. 2
Mar. 2009

电子信息工程、管理工程

文章编号：1000-8608(2009)02-0276-06

基于 Kalman 滤波的基本 Elman 网络训练新算法

李小兵， 田玉松， 邱天爽*

（大连理工大学 电子与信息工程学院，辽宁 大连 116024）

摘要： Kalman 滤波算法应用于基本 Elman 网络学习时，收敛速度较快，但收敛精度往往不高；而基于梯度下降的 BP 算法可以以很高的精度实现输入输出的非线性映射，但在极值点处收敛速度缓慢. 针对上述问题，提出一种将 Kalman 滤波算法应用于基本 Elman 网络的新学习训练算法. 该算法结合 Kalman 滤波算法和基于梯度下降的 BP 算法的优点来训练网络，以基本 Elman 网络隐层单元输出作为非线性系统的状态变量，通过 Kalman 滤波算法实现状态变量的快速准确跟踪，然后通过梯度下降法修正权值以保证精度. 另外，在训练过程中，通过增加训练样本的信息内容来提高网络收敛的精度. 仿真结果表明了该算法的有效性.

关键词： Kalman 滤波；梯度下降算法；基本 Elman 网络

中图分类号： TN911.6　　**文献标志码：** A

0 引 言

基本 Elman 网络[1,2]是动态神经网络[3]的一种，它将前馈网络隐节点的值反馈到前一层节点上或者在本层节点上进行自反馈，实现了非线性动力学系统的映射. 因为其结构相对简单，运算量小，而且网络训练仅仅需要输入输出信息，无需网络内部状态信息，所以，在实时系统辨识方面应用广泛.

目前，基本 Elman 网络训练多采用基于梯度下降的 BP 算法和扩展 Kalman 滤波算法[4,5]. Kalman 滤波算法应用于基本 Elman 网络的学习时尽管收敛速度较快，但收敛精度不高[1]. 其原因在于这种扩展 Kalman 滤波算法仅能实现线性部分的快速跟踪和逼近，但对于非线性部分则很难实现. 另一方面，基于梯度下降的 BP 算法应用于基本 Elman 网络的学习时可以以很高的精度实现输入输出的非线性映射，但其不足之处在于在极值点处收敛速度缓慢，而且将其应用于非线性系统时跟踪性能不佳. 考虑到 Kalman 滤波算法和梯度下降法算法的长处和不足，本文提出一种基于 Kalman 滤波的新学习算法. 该算法将基本 Elman 网络的隐层单元输出作为非线性系统的状态变量，通过 Kalman 滤波算法实现状态变量的快速准确跟踪，然后通过梯度下降法修正权值. 同时，在训练过程中，不断更新训练样本的内容，从而提高训练的收敛精度.

1 基本 Elman 网络及用于网络训练的扩展 Kalman 滤波算法

1.1 基本 Elman 网络

基本 Elman 网络是一种动态递归神经网络，它主要由输入层、隐层、结构单元层和输出层组成，其中每个隐层节点都有一个与之对应的结构单元节点相连接，输入层节点与隐层节点、隐层节点与输出层节点以及结构层节点与隐层节点之间都有可调节权相连接. 基本 Elman 网络输入-输出关系为

$$\hat{\mathbf{X}}(n) = \varphi_{\mathrm{h}}(\mathbf{S}^{\mathrm{h}}(n)) \tag{1}$$

$$\mathbf{S}^{\mathrm{h}}(n) = \mathbf{W}^{\mathrm{b}}(n)\,\hat{\mathbf{X}}(n-1) + \mathbf{W}^{\mathrm{a}}(n)\mathbf{U}(n) \tag{2}$$

$$\mathbf{Y}(n) = \phi_{\mathrm{o}}(\mathbf{S}^{\mathrm{o}}(n)) \tag{3}$$

收稿日期： 2007-02-11； **修回日期：** 2009-01-16.
基金项目： 国家自然科学基金资助项目(60372081,30570475,60872122)；辽宁省科学技术基金资助项目(2001101057)；教育部博士学科点专项科研基金资助项目(20050141025).
作者简介： 李小兵(1967-)，男，博士生，讲师；邱天爽*(1954-)，男，教授，博士生导师，E-mail：qiutsh@dlut.edu.cn.

$$\boldsymbol{S}^{o}(n)=\boldsymbol{W}^{o}(n)\ \hat{\boldsymbol{X}}(n) \tag{4}$$

式中：$\boldsymbol{W}^{b}(n)$ 为状态矢量 $\hat{\boldsymbol{X}}(n-1)$ 的前馈权值矩阵；$\boldsymbol{W}^{a}(n)$ 为输入矢量的前馈权值矩阵，含阈值；$\boldsymbol{W}^{o}(n)$ 为隐层到输出层的权矩阵，含阈值；$\boldsymbol{S}^{h}(n)$ 和 $\boldsymbol{S}^{o}(n)$ 分别为隐节点和输出节点的线性输出；$\varphi_{h}(\cdot)$ 和 $\phi_{o}(\cdot)$ 分别为隐节点和输出节点的线性激活函数. 通过网络结构图可以看出，如果将隐层的输出看做是系统的一种内在状态，Elman 具有状态记忆功能，它的输出是上下文相关的，因此具有记忆功能.

1.2　扩展 Kalman 滤波算法应用于基本 Elman 网络学习训练

扩展 Kalman 滤波算法应用于基本 Elman 网络时，通常将基本 Elman 网络中待修正的权值作为非线性系统的状态变量 $\boldsymbol{X}$，以 $\boldsymbol{X}$ 的已知滤波估计值 $\hat{\boldsymbol{X}}$ 为参考点，将该非线性系统在 $\hat{\boldsymbol{X}}$ 附近进行泰勒展开，取其一次项得到 $\boldsymbol{X}$ 的线性化方程，导出有限范围的近似滤波估值算法. 设一基本 Elman 网络有 M 个连接权 $\boldsymbol{w}(n)$ 和 p 个输出节点，其状态空间方程模型可描述为

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n) \tag{5}$$

$$\boldsymbol{d}_{0}(n)=\boldsymbol{c}(\boldsymbol{w}(n)\quad \boldsymbol{u}(n)\quad \boldsymbol{r}(n))+\boldsymbol{v}(n) \tag{6}$$

式中：权矢量 $\boldsymbol{w}(n)$ 起状态矢量的作用；$\boldsymbol{u}(n)$ 和 $\boldsymbol{r}(n)$ 分别表示输入和递归节点的活动；$\boldsymbol{d}_{0}(n)$ 表示模型的期望响应；测量噪声 $\boldsymbol{v}(n)$ 为零均值白噪声过程，其协方差阵 $\boldsymbol{R}(n)=E\{\boldsymbol{v}(n)\boldsymbol{v}^{H}(n)\}$，为对角阵. 将该模型与线性动态系统模型比较可知，两者的差别在于观测矩阵 $\boldsymbol{c}(\boldsymbol{w}(n)\quad \boldsymbol{u}(n)\quad \boldsymbol{r}(n))$ 是非线性的，因此需要对其线性化以便应用 Kalman 滤波算法. 为此，对式(6) 采用求偏导数的方法线性化为

$$\boldsymbol{d}_{0}(n)=\boldsymbol{C}(n)\boldsymbol{w}(n)+\boldsymbol{v}(n) \tag{7}$$

其中

$$\boldsymbol{C}(n)=\frac{\partial \boldsymbol{d}_{0}(n)}{\partial \boldsymbol{w}(n)}=\frac{\partial \boldsymbol{c}(\boldsymbol{w}(n)\quad \boldsymbol{u}(n)\quad \boldsymbol{r}(n))}{\partial \boldsymbol{w}(n)} \tag{8}$$

经过线性化后，状态空间方程模型即可采用 Kalman滤波算法进行估计了.

2　扩展 Kalman 滤波算法应用于基本 Elman 网络训练的新算法

不同于扩展 Kalman 滤波算法以基本 Elman 网络的权值作为非线性系统的状态矢量 $\boldsymbol{X}$，本文提出的新算法以基本 Elman 网络的隐层神经元输出状态作为非线性系统的状态矢量，在处理时简单地视权值和状态变量为线性关系，这样有利于 Kalman 滤波器准确快速地估计状态. 同样，为了应用 Kalman 滤波算法对基本 Elman 网络进行训练，首先是对基本 Elman 网络的输入-输出关系线性化. 由式(1)、(2)可得基本 Elman 网络的状态方程为

$$\hat{\boldsymbol{X}}(n)=\varphi_{h}[\boldsymbol{W}^{b}(n)\ \hat{\boldsymbol{X}}(n-1)+\boldsymbol{W}^{a}(n)\boldsymbol{U}(n)] \tag{9}$$

另外，由式(3)、(4) 得到观测方程为

$$\boldsymbol{Y}(n)=\phi_{o}(\boldsymbol{W}^{o}(n)\ \hat{\boldsymbol{X}}(n)) \tag{10}$$

考虑到基本 Elman 网络应用时经常取 $\phi_{o}(\cdot)$ 为线性函数，则式(10) 可简化为

$$\boldsymbol{Y}(n)=\boldsymbol{W}^{o}(n)\ \hat{\boldsymbol{X}}(n) \tag{11}$$

如果隐层激活函数 $\varphi_{h}(\cdot)$ 为线性的，则由式(9) 可得到状态方程的线性化方程

$$\hat{\boldsymbol{X}}(n)=\boldsymbol{W}^{b}(n)\ \hat{\boldsymbol{X}}(n-1)+\boldsymbol{W}^{a}(n)\boldsymbol{U}(n) \tag{12}$$

在式(12) 中，令 $\boldsymbol{A}(n)=\boldsymbol{W}^{b}(n)$，$\boldsymbol{B}(n)=\boldsymbol{W}^{a}(n)$，可以得到线性化方程

$$\hat{\boldsymbol{X}}(n)=\boldsymbol{A}(n)\ \hat{\boldsymbol{X}}(n-1)+\boldsymbol{B}(n)\boldsymbol{U}(n) \tag{13}$$

但如果 $\varphi_{h}(\cdot)$ 为非线性的，则可对式(9) 在零状态处进行泰勒展开，略去高阶项可得到线性化方程为

$$\hat{\boldsymbol{X}}(n)=\varphi_{h}(0)\boldsymbol{I}+\varphi_{h}'(0)[\boldsymbol{W}^{b}(n)\ \hat{\boldsymbol{X}}(n-1)+\boldsymbol{W}^{a}(n)\boldsymbol{U}(n)] \tag{14}$$

式中，令 $\boldsymbol{A}(n)=\varphi_{h}'(0)\boldsymbol{W}^{b}(n)$，$\boldsymbol{B}(n)=\varphi_{h}'(0)\boldsymbol{W}^{a}(n)$，则可得到如下线性状态方程：

$$\hat{\boldsymbol{X}}(n)=\boldsymbol{A}(n)\ \hat{\boldsymbol{X}}(n-1)+\boldsymbol{B}(n)\boldsymbol{U}(n)+\varphi_{h}(0)\boldsymbol{I} \tag{15}$$

这样，由式(11)、(13) 或式(11)、(15) 组成的是一个线性系统. 采用普通的 BP 算法修正权值 $\boldsymbol{W}^{o}(n)$. 另外，根据输入 $\boldsymbol{U}(n)$、输出 $\boldsymbol{Y}(n)$ 和前一时刻的状态 $\hat{\boldsymbol{X}}(n-1)$，采用 Kalman 滤波算法容易估计出与权值 $\boldsymbol{A}(n)$、$\boldsymbol{B}(n)$ 和 $\boldsymbol{W}^{o}(n)$ 相对应的状态变量，设其为 $\hat{\boldsymbol{X}}^{*}(n)$. 具体 Kalman 滤波算法的数值计算方法见文献[6]. 另一方面，由式(13) 或(15) 可递推得到状态变量的另一个值 $\hat{\boldsymbol{X}}(n)$. 这样，$\hat{\boldsymbol{X}}^{*}(n)$ 和 $\hat{\boldsymbol{X}}(n)$ 的差异可认为是权值 $\boldsymbol{A}(n)$ 和 $\boldsymbol{B}(n)$ 偏离其真值而造成的. 此时，采用梯度下降法修正权系数 $\boldsymbol{A}(n)$ 和 $\boldsymbol{B}(n)$. 具体方法描述如下：

(1) 设状态变量 $\hat{\mathbf{X}}^*(n)$ 和 $\hat{\mathbf{X}}(n)$ 是 N 维列向量,则总误差为

$$e(n)=\frac{1}{2}[\hat{\mathbf{X}}^*(n)-\hat{\mathbf{X}}(n)]^{\mathrm{T}}[\hat{\mathbf{X}}^*(n)-\hat{\mathbf{X}}(n)] \tag{16}$$

(2) 设权值 $\mathbf{A}(n)$ 和 $\mathbf{B}(n)$ 分别为 N×N 和 N×1 矩阵,对总误差 $e(n)$ 求权值的偏导数

$$\frac{\partial e(n)}{\partial \mathbf{A}(n)}=\begin{pmatrix}\frac{\partial e(n)}{\partial a_{11}} & \cdots & \frac{\partial e(n)}{\partial a_{1\mathrm{N}}}\\ \vdots & & \vdots\\ \frac{\partial e(n)}{\partial a_{\mathrm{N}1}} & \cdots & \frac{\partial e(n)}{\partial a_{\mathrm{NN}}}\end{pmatrix} \tag{17}$$

$$\frac{\partial e(n)}{\partial \mathbf{B}(n)}=\begin{pmatrix}\frac{\partial e(n)}{\partial b_1}\\ \vdots\\ \frac{\partial e(n)}{\partial b_{\mathrm{N}}}\end{pmatrix} \tag{18}$$

式中 $a_{11},\cdots,a_{\mathrm{NN}}$ 与 $b_1,\cdots,b_{\mathrm{N}}$ 分别为矩阵 $\mathbf{A}(n)$ 和 $\mathbf{B}(n)$ 中的元素. 由于式(13) 或(15) 表示的是一个关于权值 $\mathbf{A}(n)$ 和 $\mathbf{B}(n)$ 的线性方程,对其求偏导数十分方便. 下面仅以 $\frac{\partial e(n)}{\partial a_{ij}}$ 的求取为例来说明偏导数的求法:

$$\frac{\partial e(n)}{\partial a_{ij}}=[\hat{x}_i^*(n)-\hat{x}_i(n)]\frac{\partial \hat{x}_i(n)}{\partial a_{ij}}=[\hat{x}_i^*(n)-\hat{x}_i(n)]\hat{x}_j(n-1) \tag{19}$$

式中 $\hat{x}_i^*(n)$ 与 $\hat{x}_i(n)$ 分别为 $\hat{\mathbf{X}}^*(n)$ 和 $\hat{\mathbf{X}}(n)$ 的第 i 个元素.

(3) 进行权值的更新:

$$\mathbf{A}(n+1)=\mathbf{A}(n)-r\frac{\partial e(n)}{\partial \mathbf{A}(n)} \tag{20}$$

$$\mathbf{B}(n+1)=\mathbf{B}(n)-r\frac{\partial e(n)}{\partial \mathbf{B}(n)} \tag{21}$$

式中 r 为学习率.

这样,通过不断迭代学习,最终权值收敛到稳定值 $\mathbf{A}$、$\mathbf{B}$ 和 $\mathbf{W}^{\mathrm{o}}$. 此时根据 $\mathbf{A}(n)$ 与 $\mathbf{W}^{\mathrm{b}}(n)$ 及 $\mathbf{B}(n)$ 与 $\mathbf{W}^{\mathrm{a}}(n)$ 的关系得到 $\mathbf{W}^{\mathrm{b}}(n)$ 和 $\mathbf{W}^{\mathrm{a}}(n)$ 的稳定值 $\mathbf{W}^{\mathrm{b}}$ 和 $\mathbf{W}^{\mathrm{a}}$:

$$\mathbf{W}^{\mathrm{b}}=\varphi_{\mathrm{h}}'(0)^{-1}\mathbf{A} \tag{22}$$

$$\mathbf{W}^{\mathrm{a}}=\varphi_{\mathrm{h}}'(0)^{-1}\mathbf{B} \tag{23}$$

这里,从求解的过程容易看出,此时得到的稳定权值是式(9)、(10)所确定的系统的近似最优权值.

3 仿真结果

为了验证本文算法对基本 Elman 网络学习的有效性,采用基本 Elman 网络对如下的非线性系统进行仿真:

$$y(n)=0.4\sin(0.5(y(n-1)+y(n-2)))\times\cos(0.5(y(n-1)+y(n-2)))+1.2u(n) \tag{24}$$

其中系统输入 $u(n)$ 为均值为 0、方差为 1 的高斯随机白噪声序列. 系统初值满足 $y(-1)=0$, $y(-2)=0$. 基本 Elman 网络结构采用 1 个输入单元、3 个隐单元及结构单元和 1 个输出单元的 3 层网络. 输出层激活函数采用线性函数,隐层激活函数分别采用线性函数和对称 s 函数进行仿真. 网络初始状态选取为零状态,学习率 $r=0.1$. 另外,网络收敛条件为 20 个样本的总误差小于 0.1.

为了比较本文新算法与其他算法的性能,对基本 Elman 网络分别采用梯度下降法(如文献[3])、本文新算法进行学习训练. 其中当隐层单元取线性函数时较为简单,仅采用本文新算法进行仿真,当隐层单元取对称 s 函数时采用两种方法进行仿真.

采用上述方法训练网络时,选取 10 组不同的输入 $u(n)$ 进行 10 次仿真. 对每一次仿真,取前 20 点输入 $u(n)$ 及其对应输出 $y(n)(0\leqslant n\leqslant 19)$ 作为训练样本,选其后 80 点数据为检验样本. 基本 Elman 网络初始权值取为小的随机数,初始状态取为零状态. 网络训练次数选为 500. 本文给出一组输入 $u(n)$ 的仿真结果. 图 1 和 2 为隐层单元取线性函数时得到的误差收敛曲线和系统拟合曲线,图 3 和 4 为隐层单元取对称 s 函数时得到的误差收敛曲线和系统拟合曲线. 在图 1 和 3 中,虚线为梯度下降法对应的误差收敛曲线,实线为本

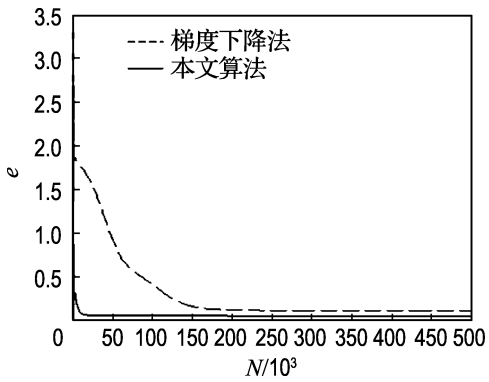

图 1 隐层单元为线性时的误差收敛曲线

Fig. 1 Error convergence curve of linear hidden-layer nodes network

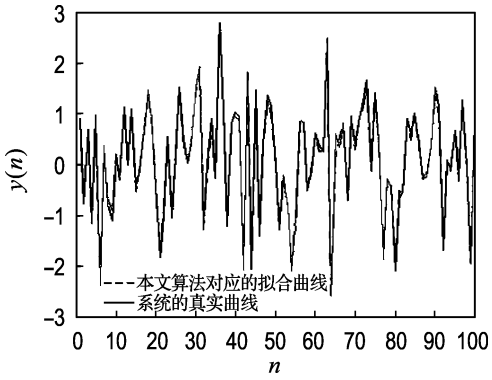

图 2　隐层单元为线性时的拟合曲线

Fig. 2　Fitted curve of linear hidden-layer nodes network

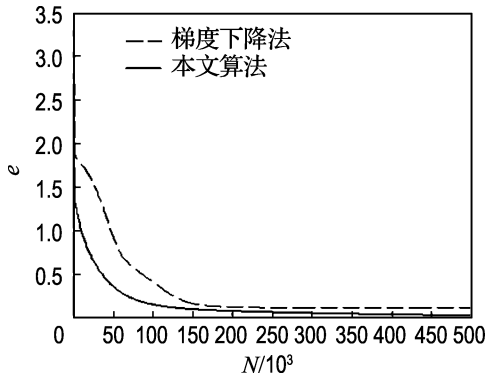

图 3　隐层单元为 s 函数时的误差收敛曲线

Fig. 3　Error convergence curve of s function hidden-layer nodes network

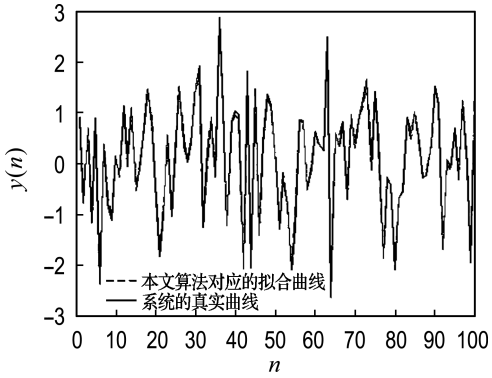

图 4　隐层单元为 s 函数时的拟合曲线

Fig. 4　Fitted curve of s function hidden-layer nodes network

文新算法对应的误差收敛曲线. 图 5 为采用梯度下降法训练网络时得到的拟合曲线.

从图 1、3 可以看出，采用本文新算法训练网络时，收敛速度较梯度下降法训练网络时快，最终的收敛误差也相对要小. 同时比较图 2、4 和 5 也可以看出本文新算法的拟合效果要比梯度下降法好.

表 1 给出了对 10 组不同的输入 $u(n)$，分别采用梯度下降法和本文新算法训练且基本 Elman 网络隐层分别取线性和对称 s 函数情况下训练的最终收敛误差.

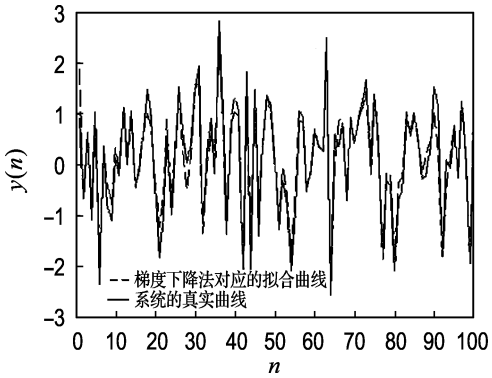

图 5　梯度下降法对应的拟合曲线

Fig. 5　Fitted curve of gradient descend algorithm

表 1　采用两种训练方法训练网络得到的最终收敛误差

Tab. 1　Comparison of convergence error by two different training network algorithms

序号	梯度下降法	本文算法	
		隐层为线性函数	隐层为 s 函数
1	3.686 3	0.056 2	0.252 1
2	0.106 6	0.046 2	0.025 2
3	0.341 6	0.028 4	0.072 4
4	0.176 9	0.029 9	0.089 0
5	0.053 3	0.038 7	0.068 9
6	0.163 4	0.047 1	0.082 2
7	0.163 4	0.068 3	0.039 4
8	0.113 4	0.071 6	0.253 9
9	0.114 2	0.038 5	0.052 9
10	7.353 1	0.025 8	0.051 4

注：3 种情况对应的平均值分别为 1.217 3、0.045 1、0.098 7

从表中可以看出，采用梯度下降法训练网络时，除第 5 组和第 7 组数据最终得到收敛的结果外，另有 6 组数据（第 2 组、第 3 组、第 4 组、第 6 组、第 8 组和第 9 组）最终陷入局部极小，其余两组数据（第 1 组和第 10 组）开始有收敛趋势，但随着网络训练的进行最终得到发散的结果. 而采用本文新算法训练网络时，无论隐层单元为线性函数还是对称 s 函数，仿真结果大多比梯度下降法训练网络得到的仿真结果好. 尤其是当隐层单元取线性函数，采用本文新算法训练网络得到的训练结果相当好，但当隐层单元取对称 s 函数时，最终平均收敛误差变大，而且有两组数据（第 1 组和第 8 组）在训练过程中陷入局部极小. 可能的原因

是隐层单元取对称 s 函数时，基本 Elman 网络跟踪性能变差.

从上面的结果看，采用本文新算法训练网络大多都能得到较好的训练结果，但当基本 Elman 网络的隐层单元激活函数取为对称 s 函数时，会出现数据陷入局部最小的情况. 本文认为这是选取的权系数初始值与对应的 20 个训练样本不当造成的. 为此，如果在训练过程中，不断增大样本集，也就是不断更改训练样本以增加新的训练信息，则有望找到适当的训练样本来训练网络. 方法是在某步迭代训练过程中，当误差能量函数减到很小(认为网络训练已经陷入局部极小)时，本文对训练样本作如下更改：假设当前迭代训练的 20 个训练样本为输入 $u(n)$ 及其对应输出 $y(n)$($n_0 \leqslant n \leqslant n_0+19$，这里 n_0 为样本选取的起始值，初始时 $n_0=0$)，则下一步迭代训练时，增加一个训练样本——$u(n_0+20)$ 及其对应的 $y(n_0+20)$，并减少一个训练样本——$u(n_0)$ 及其对应的 $y(n_0)$. 这样的处理可以在训练过程中不断增加新的训练信息，有望避免训练陷入局部极小. 同时，由于在网络训练过程中仅仅改变了训练样本，不会增加算法的复杂性. 另外，训练样本的更改是在训练陷入局部极小的基础上进行的，此时，基本 Elman 网络的权值已经趋于稳定，而且其隐层的状态值由于采用 Kalman 滤波算法进行跟踪而具有较好的延续性. 这样当增加新的训练样本时，输出误差能量函数不会出现大的波动，算法较稳定.

本文视 20 个样本的误差能量大于 0.1，而训练迭代后误差能量减小量低于 10% 为训练陷入局部极小，对上述两组陷入局部极小的数据重新进行仿真，得到的收敛误差相应为 0.135 2 和 0.104 5. 从结果可以看出，改进后最终收敛误差有较大程度的减小. 本文给出采用第 8 组数据进行仿真的收敛误差和拟合曲线. 图 6 是改进前后收敛误差曲线. 图 7、8 分别表示改进后和改进前采用本文新算法对网络进行训练得到的拟合曲线.

从图 6 中可以看出算法改进后的网络收敛误差较改进前要小. 同时图 7、8 也显示算法改进后得到的拟合曲线精度较改进前得到的拟合曲线精度要高. 另外从得到的结果可以看出对第 1 组和第 8 组数据，改进后最终收敛误差仍然较大，但考虑到本文训练时最多用到了输入 $u(n)$的前 100 点数据，如果增加输入的点数，有望得到更好的结果.

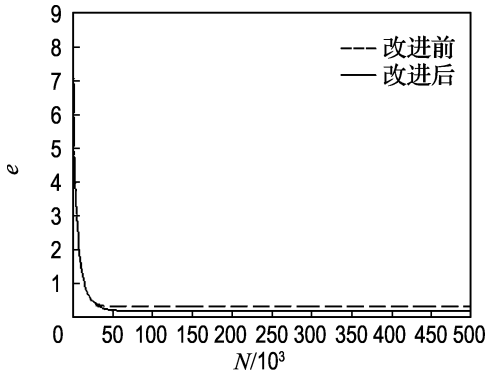

图 6 改进前后误差收敛曲线比较

Fig. 6 Comparison of error convergence curves of improved and unimproved algorithms

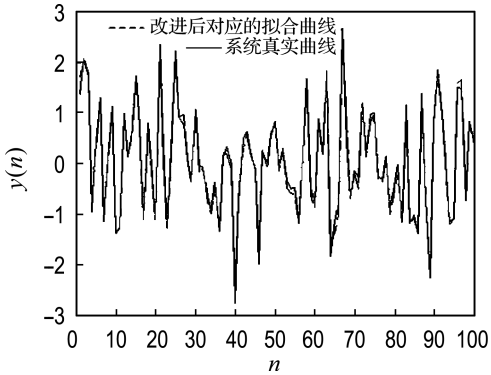

图 7 改进后得到的拟合曲线

Fig. 7 Fitted curve of improved algorithms

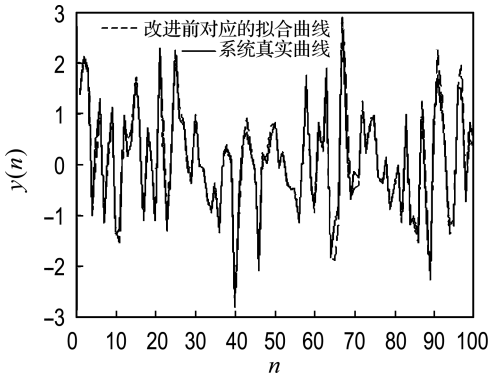

图 8 改进前得到的拟合曲线

Fig. 8 Fitted curve of unimproved algorithms

将基本 Elman 网络中待修正的权值作为非线性系统的状态变量进行网络训练时，结构单元的输出是一个动态递推过程，每一步迭代过程，按 BP 算法求解 $\mathbf{C}(n)$(如式(8)所示)只用到了一阶梯度，因而会导致结构单元连接权的稳定性较差. 而本文新算法首先采用一个线性系统(如式(11)、(13)或式(11)、(15)所示)模拟基本 Elman 网络，然后在此基础上进行网络权值的估计，不必每一

步迭代都利用 BP 算法求解 $\mathbf{C}(n)$，这样一方面可以使一次 Kalman 滤波算法的计算开销减小；另一方面，可以有效提高基本 Elman 网络的跟踪能力，因此本文新算法有更好的稳定性. 同时，Kalman滤波算法是一种比梯度下降算法更高效地使用高阶信息的算法，相比单纯采用 BP 算法训练网络时，神经元输出状态初始值的不适当对算法收敛性能有一定影响. 本文新算法在每一步迭代过程中首先采用 Kalman 滤波算法实现式(11)、(13)或式(11)、(15)组成的线性系统状态的快速获取，此时估计的系统状态可认为是这一步迭代在当前权值下神经元输出状态的无偏及最小方差估计，在此基础上，对上述线性系统采用 BP 算法修正权值，可以得到更快的收敛速度. 本文对 10 组数据进行的仿真结果也证明了这一结论. 另外，本文新算法在训练过程中可以不断增加新的训练信息，有望提高训练效果，这也是梯度下降法和 Kalman 滤波算法不具备的. 文献[2]也给出了一种结合梯度下降法和滤波算法的分层学习算法，与之相比，本文新算法除了保持有较高精度和较快收敛速度的优点外，复杂度相对较小.

4　结　语

本文针对 Kalman 滤波算法和梯度下降法在基本 Elman 网络学习训练中各自存在的优点和不足，提出一种新的学习算法. 该算法以基本 Elman网络本身隐层单元的输出作为非线性系统的状态矢量，以梯度下降法修正网络权值，在学习训练过程中，可以不断更新训练样本，增加训练信息. 本文新算法不仅具有较高的收敛精度和较快的收敛速度，而且计算复杂度较小. 仿真结果表明了该算法的有效性.

参考文献：

[1] 何玉彬，李新忠. 神经网络控制技术及应用[M]. 北京：科学出版社，2000

[2] 陈　巍，吴　捷. 递归神经网络的卡尔曼滤波及分层学习算法[J]. 华南理工大学学报(自然科学版)，1998，**26**(4)：44-48

[3] HAYKIN S. **Neural Networks：A Comprehensive Foundation** [M]. Beijing：Tsinghua University Press，2001

[4] SINGHAL S，WU L. Training feed-forward networks with the extended Kalman algorithm [J]. **International Conference on Acoustics，Speech and Signal Processing**，1989，**2**：1187-1190

[5] PUSKOVIUS G V，FELDKUMP L A. Decoupled extended Kalman filter training of feed-forward layered networks [C] // **International Joint Conference on Neural Networks — IJCNN-91-Seattle Part 1 (of 2)**. Piscataway：IEEE，1991

[6] 张贤达. 现代信号处理[M]. 北京：清华大学出版社，2002

A new training algorithm based on Kalman filter of essential Elman network

LI　Xiao-bing，　TIAN　Yu-song，　QIU　Tian-shuang*

(School of Electronic and Information Engineering，Dalian University of Technology，Dalian 116024，China)

Abstract：In the application of essential Elman network，the Kalman filter algorithm has high convergence speed but low precision；on the contrary，the BP algorithm with gradient descend technique has high precision to map the output with input nonlinearly，but usually with low convergence speed at extreme point. Therefore，a new algorithm based on Kalman filter is proposed to accomplish essential Elman network training. The advantages of Kalman filter and BP algorithm are combined to optimize the new algorithm. The outputs of the hidden layer of the essential Elman network are set as the state varibles of a nonlinear system. Kalman filter algorithm is used for rapid tracking of the state variables and BP algorithm is used for more precise weights modification. Furthermore，some new learning information is added to improve the precision of the results during the learning process. The validity of the new algorithm is supported by the simulation results.

Key words：Kalman filter；gradient descend algorithm；essential Elman network