文章编号: 1000- 0887(2008)09- 1117- 07

Elman 网络梯度学习法的收敛性*

吴 微, 徐东坡, 李正学

(大连理工大学 应用数学系, 辽宁 大连 116023)

(郭兴明推荐)

摘要: 考虑有限样本集上 Elman 网络梯度学习法的确定性收敛性. 证明了误差函数的单调递减性. 给出了一个弱收敛性结果和一个强收敛结果, 表明误差函数的梯度收敛于 0, 权值序列收敛于固定点. 通过数值例子验证了理论结果的正确性.

关 键 词: Elman 神经网络; 梯度学习算法; 收敛性; 单调性

中图分类号: TP183 文献标识码: A

引 言

Elman 网络是一类带隐层的递归神经网络, 每个隐单元通过反馈连接到全部隐单元[1- 2]. 这个反馈连接使得 Elman 神经网络能够学习, 识别和生成时间模式与空间模式[3]. 用于时间序列分类, Elman 网络是一种强有力的工具[4], 线性与非线性动态系统建模[5]. Elman 网络已经广泛地应用在许多不同的领域, 如语言习得[6]、噪声生成[7]、故障定位[8]和流估计[9]. Elman 网络包含在流行的 Matlab 神经网络工具箱里[10].

类似于前馈神经网络情况, 梯度学习算法普遍用于训练递归神经网络[11- 12], 主要由于它的简单性. Elman 网络梯度法的收敛性理论分析有助于理解方法的行为, 引导这个最简单学习方法的进一步改进.

对于对角型 Elman 网络, 下面(1)式中 $\boldsymbol{V}_2$ 是对角阵时, 文献[13- 14]研究了梯度法的收敛性. 对于一般型 Elman 网络, 已存在的收敛结果[15]是概率性的渐进收敛性, 其假设训练样本的个数趋紧于无穷. 本文, 只考虑有限训练样本集可以利用, 不适合用随机理论方法处理. 相反, 用离线梯度法训练 Elman 网络, 相应地证明了一些确定性收敛结果. 表明在训练过程中误差函数单调下降, 它的梯度收敛于 0. 本文的证明用到了文献[16- 17]中的一些技巧.

1 Elman 神经网络结构

如图 1 所示, Elman 神经网络具有 N 个输入神经元, M 个隐层神经元和一个输出神经元. 输入和递归权值矩阵分别为 $\boldsymbol{V}_1 \in R^{M\times N}$ 和 $\boldsymbol{V}_2 \in R^{M\times M}$. 为表述的简单, 我们合并权值矩阵为

* 收稿日期: 2007- 12- 05; 修订日期: 2008- 07- 19
基金项目: 国家自然科学基金资助项目(10471017)
作者简介: 吴微(1953—), 男, 黑龙江牡丹江人, 教授, 博士生导师(联系人. Tel: + 86- 411- 84708294; E- mail: wuweiw@dlut.edu.cn).

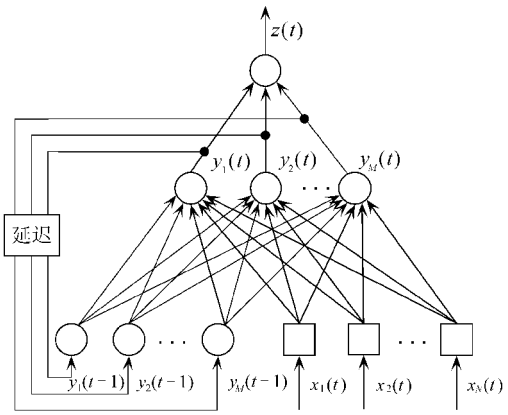

图 1 Elman 神经网络 $N-M-1$ 结构

$$\boldsymbol{W}_1 = (\boldsymbol{V}_1, \boldsymbol{V}_2) \in R^{M\times(N+M)}, \tag{1}$$

$\boldsymbol{w}_0 \in R^M$ 表示隐层神经元和输出神经元连接的权值向量. 对于时间序列$\{\boldsymbol{x}(t), t = 1, 2, \cdots\} \subset R^N$ 输入给网络, 用 $\boldsymbol{y}(t) \in R^M$ 表示隐层 t 时刻的相应输出, 定义 $\boldsymbol{y}(0) = 0$. 假定 $g: \mathbf{R} \to \mathbf{R}$ 为隐层神经元和输出神经元的激活函数, 对任意向量 $\boldsymbol{a} = (a_1, a_2, \cdots, a_M)^{\mathrm{T}} \in R^M$, 引入如下向量函数:

$$\boldsymbol{G}(\boldsymbol{a}) = (g(a_1), g(a_2), \cdots, g(a_M))^{\mathrm{T}}. \tag{2}$$

为方便起见, 结合 $\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t-1)$ 为 $N+M$ 维向量

$$\boldsymbol{u}(t) = \begin{pmatrix} \boldsymbol{x}(t) \\ \boldsymbol{y}(t-1) \end{pmatrix}, \quad t = 1, 2, \cdots, \tag{3}$$

隐层的输入为

$$\boldsymbol{s}(t) = \boldsymbol{W}_1\boldsymbol{u}(t) = \boldsymbol{V}_1\boldsymbol{x}(t) + \boldsymbol{V}_2\boldsymbol{y}(t-1), \quad t = 1, 2, \cdots, \tag{4}$$

隐层的输出为

$$\boldsymbol{y}(t) = \boldsymbol{G}(\boldsymbol{s}(t)). \tag{5}$$

网络的输出为

$$z(t) = g(\boldsymbol{w}_0 \bullet \boldsymbol{y}(t)), \quad t = 1, 2, \cdots. \tag{6}$$

2 梯度学习算法

对于任意矩阵 $\boldsymbol{A} \in R^{m\times n}$ 与向量 $\boldsymbol{x} \in R^n$, 其范数分别定义为 $\|\boldsymbol{A}\| = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}$ 和 $\|\boldsymbol{x}\| = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$, 显然 $\|\boldsymbol{A}\boldsymbol{x}\| \leqslant \|\boldsymbol{A}\|\|\boldsymbol{x}\|$.

定义 2.1 给定 $m \times n$ 矩阵 $\boldsymbol{A} = (a_{ij})$, $\mathrm{vec}\boldsymbol{A}$ 定义为 mn 维向量如下:

$$\mathrm{vec}\boldsymbol{A} = (a_{11}, a_{21}, \cdots, a_{m1}, a_{12}, a_{22}, \cdots, a_{m2}, \cdots, a_{1n}, a_{2n} \cdots, a_{mn})^{\mathrm{T}}. \tag{7}$$

定义 2.2 $\boldsymbol{A} = (a_{ij})$ 是 $p \times q$ 矩阵, $\boldsymbol{A}^{(n)}$ 定义为 $\boldsymbol{A}^{(n)} = (a_{ij}^n)$.

定义 2.3 $\boldsymbol{A} = (a_{ij})$ 是 $m \times n$ 矩阵, $\boldsymbol{B} = (b_{ij})$ 是 $p \times q$ 矩阵, Kronecker 积 $\boldsymbol{A} \otimes \boldsymbol{B}$ 是 $mp \times nq$ 矩阵, 其定义为

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{pmatrix}. \tag{8}$$

定义 2.4 对于(2)式 $\boldsymbol{G}(\boldsymbol{a})$, 定义 $\boldsymbol{G}'(\boldsymbol{a})$ 为对角矩阵, 其对角元素为 $g'(a_1), \cdots, g'(a_M)$. 同样定义 $\boldsymbol{G}''(\boldsymbol{a})$ 为对角矩阵, 其对角元素为 $g''(a_1), \cdots, g''(a_M)$.

现在描述梯度学习算法. 假设训练样本集为$\{\boldsymbol{x}(t), O(t)\}_{t=1}^{Q}$, 令

$$\boldsymbol{w}_1 = \mathrm{vec}\,\boldsymbol{W}_1, \tag{9}$$

$$\boldsymbol{w} = \mathrm{vec}(\boldsymbol{w}_0, \boldsymbol{W}_1). \tag{10}$$

定义平方误差函数

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{t=1}^{Q}(O(t) - g(\boldsymbol{w}_0 \cdot \boldsymbol{G}(\boldsymbol{W}_1 \boldsymbol{u}(t))))^2. \tag{11}$$

网络学习的目的是找到 $\boldsymbol{w}^*$，使得

$$E(\boldsymbol{w}^*) = \min E(\boldsymbol{w}). \tag{12}$$

对 $E(\boldsymbol{w})$ 关于 $\boldsymbol{w}_0$ 和 $\boldsymbol{w}_1$ 分别求偏导，定义相应的梯度函数

$$E_{\boldsymbol{w}_0}(\boldsymbol{w}) = -\sum_{t=1}^{Q}(O(t) - z(t))g'(\boldsymbol{w}_0 \cdot \boldsymbol{y}(t))\boldsymbol{y}(t), \tag{13}$$

$$E_{\boldsymbol{w}_1}(\boldsymbol{w}) = -\sum_{t=1}^{Q}(O(t) - z(t))g'(\boldsymbol{w}_0 \cdot \boldsymbol{y}(t))(\boldsymbol{P}(t))^{\mathrm{T}}\boldsymbol{w}_0, \tag{14}$$

这里

$$\boldsymbol{P}(t) = \boldsymbol{G}'(\boldsymbol{s}(t))((\boldsymbol{u}(t))^{\mathrm{T}} \otimes \boldsymbol{I}_M + \boldsymbol{V}_2\boldsymbol{P}(t-1)), \tag{15}$$

初始条件为 $\boldsymbol{P}(0) = 0$，$\boldsymbol{I}_M$ 是 $M \times M$ 单位阵.

任意给定初始权值 $\boldsymbol{w}^0$，生成权值序列 $\boldsymbol{w}^k$

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \eta E_{\boldsymbol{w}}(\boldsymbol{w}^k), \qquad k = 0, 1, 2, \cdots, \tag{16}$$

这里

$$E_{\boldsymbol{w}}(\boldsymbol{w}^k) = \begin{pmatrix} E_{\boldsymbol{w}_0}(\boldsymbol{w}^k) \\ E_{\boldsymbol{w}_1}(\boldsymbol{w}^k) \end{pmatrix}, \tag{17}$$

其中，$\eta \in (0, 1)$ 是常数学习率.

令

$$\Delta\boldsymbol{w}^k = \boldsymbol{w}^{k+1} - \boldsymbol{w}^k, \tag{18}$$

那么

$$\Delta\boldsymbol{w}_0^k = \eta\sum_{t=1}^{Q}(O(t) - z^k(t))g'(\boldsymbol{w}_0^k \cdot \boldsymbol{y}^k(t))\boldsymbol{y}^k(t), \tag{19}$$

$$\Delta\boldsymbol{w}_1^k = \eta\sum_{t=1}^{Q}(O(t) - z^k(t))g'(\boldsymbol{w}_0^k \cdot \boldsymbol{y}^k(t))(\boldsymbol{P}(\boldsymbol{w}_1^k, t))^{\mathrm{T}}\boldsymbol{w}_0^k, \tag{20}$$

这里

$$z^k(t) = z(\boldsymbol{w}^k, t),\ \boldsymbol{y}^k(t) = \boldsymbol{y}(\boldsymbol{w}_1^k, t), \tag{21}$$

$$\boldsymbol{s}^k(t) = \boldsymbol{s}(\boldsymbol{w}_1^k, t),\ \boldsymbol{u}^k(t) = \boldsymbol{u}(\boldsymbol{w}_1^k, t). \tag{22}$$

本文中将会用到如下 3 个假设:

(A1) 对于任意的 $r \in \mathbf{R}$，$|g(r)|$，$|g'(r)|$，$|g''(r)|$ 有界.

(A2) 在学习过程(16)中，$\|\boldsymbol{w}_0^k\|$，$\|\boldsymbol{V}_2^k\|$ $(k = 0, 1, 2, \cdots)$ (参照(1)、(9)~(10)式)有界.

(A3) 存在一个有界闭集 $\Phi \subset R^{M(N+M+1)}$，使得$\{\boldsymbol{w}^k\} \subset \Phi$，且 $\Phi_0 = \{\boldsymbol{w} \in \Phi: E_{\boldsymbol{w}}(\boldsymbol{w}) = 0\}$ 是有限点集.

注 2.1 激活函数 g 通常取为Sigmoid 型函数，其满足于假设(A1). 为了保证迭代过程的收敛性，经常用到文献[18]中的假设(A2). 假设(A3)将用于获得强收敛结果.

3 主要结果

介绍一些在证明过程中要用到的符号:

$$\Delta \boldsymbol{s}^k(t) = \boldsymbol{s}^{k+1}(t) - \boldsymbol{s}^k(t), \tag{23}$$

$$\Delta \boldsymbol{y}^k(t) = \boldsymbol{y}^{k+1}(t) - \boldsymbol{y}^k(t), \tag{24}$$

$$\prod_{i=j}^{k} \boldsymbol{A}_i = \begin{cases} \boldsymbol{A}_k \boldsymbol{A}_{k-1} \cdots \boldsymbol{A}_j, & j \leqslant k, \\ 1, & j > k. \end{cases} \tag{25}$$

如下 3 个引理类似于文献[14] 的对应结果, 因此省略其证明过程.

引理 3.1 对于迭代公式(15) 及零初始条件, 推得

$$\boldsymbol{P}(t) = \sum_{j=0}^{t-1} \prod_{l=t-j+1}^{t} (\boldsymbol{G}'(\boldsymbol{s}(l))\boldsymbol{V}_2)\boldsymbol{G}'(\boldsymbol{s}(t-j))(\boldsymbol{u}(t-j))^{\mathrm{T}} \otimes \boldsymbol{I}_M. \tag{26}$$

引理 3.2 对于(23) 式中的 $\Delta \boldsymbol{s}^k(t)$, 推得

$$\Delta \boldsymbol{s}^k(t) = ((\boldsymbol{u}^k(t))^{\mathrm{T}} \otimes \boldsymbol{I}_M + \boldsymbol{V}_2^k \boldsymbol{P}(\boldsymbol{w}_1^k, t))\Delta \boldsymbol{w}_1^k + \delta^k(t),$$

这里

$$\delta^k(t) = \sum_{j=1}^{t-1} \prod_{l=t-j+1}^{t-1} (\boldsymbol{V}_2^k \boldsymbol{G}'(\boldsymbol{s}^k(l)))\Delta \boldsymbol{V}_2^k \boldsymbol{G}'(\boldsymbol{s}^k(t-j))\Delta \boldsymbol{s}^k(t-j) + \frac{1}{2}\sum_{j=1}^{t-1} \prod_{l=t-j+1}^{t-1} (\boldsymbol{V}_2^k \boldsymbol{G}'(\boldsymbol{s}^k(l)))\boldsymbol{V}_2^{k+1}\boldsymbol{G}''(\sigma^k(t-j))(\Delta \boldsymbol{s}^k(t-j))^{(2)},$$

向量 $\sigma^k(t)$ 的每个分量落在向量 $\boldsymbol{s}^k(t)$ 和向量 $\boldsymbol{s}^{k+1}(t)$ 的相应分量之间.

引理 3.3 假设(A1) 和(A2) 成立, 那么

$$\| \boldsymbol{y}^k(t) \| \leqslant C_0,$$

$$\| \Delta \boldsymbol{s}^k(t) \| \leqslant C_1 \| \Delta \boldsymbol{w}_1^k \|,$$

$$\| \Delta \boldsymbol{y}^k(t) \| \leqslant C_2 \| \Delta \boldsymbol{w}_1^k \|.$$

下一个引理因其证明基本上与文献[19] 中定理 14.1.5 相同[20], 故省略.

引理 3.4 设 $\boldsymbol{F}: \Phi \subset R^m \to R^m (m \geqslant 1)$ 在有界闭集 Φ 上连续, 集合 $\Phi_0 = \{\boldsymbol{w} \in \Phi: \boldsymbol{F}(\boldsymbol{w}) = 0\}$ 是有限点集. 序列 $\{\boldsymbol{w}^k\} \subset \Phi$ 满足 $\lim_{k\to\infty} \boldsymbol{F}(\boldsymbol{w}^k) = 0$ 和 $\lim_{k\to\infty} \| \boldsymbol{w}^{k+1} - \boldsymbol{w}^k \| = 0$. 那么, 存在 $\boldsymbol{w}^* \in \Phi_0$, 使得 $\lim_{k\to\infty} \boldsymbol{w}^k = \boldsymbol{w}^*$.

下面的定理是我们的主要结果.

定理 3.1 误差函数由(11) 式给出, 假设(A1)、(A2) 均成立, 对任意初始权值 $\boldsymbol{w}^0$, 权值序列 $\{\boldsymbol{w}^k\}$ 由算法(16) 式生成, 学习率 η 满足下面(33) 式, 则有

(a) $E(\boldsymbol{w}^{k+1}) \leqslant E(\boldsymbol{w}^k), \quad k = 0, 1, 2, \cdots;$

(b) 存在 $E^* \geqslant 0$ 使得 $\lim_{k\to\infty} E(\boldsymbol{w}^k) = E^*$;

(c) $\lim_{k\to\infty} \| \Delta \boldsymbol{w}^k \| = 0, \ \lim_{k\to\infty} \| E_{\boldsymbol{w}}(\boldsymbol{w}^k) \| = 0.$

而且, 如果假设(A3) 也成立, 则有强收敛性:

(d) 存在 $\boldsymbol{w}^* \in \Phi_0$, 使得 $\lim_{k\to\infty} \boldsymbol{w}^k = \boldsymbol{w}^*$.

证明 由(22) ~ (24) 式, Taylor 展开公式, 引理 3.2 和学习规则(20) 式, 我们有

$$-\sum_{t=1}^{Q} O(t) - z^k(t))g'(\boldsymbol{w}_0^k \cdot \boldsymbol{y}^k(t))\boldsymbol{w}_0^k \cdot \Delta \boldsymbol{y}^k(t) = -\frac{1}{\eta}\| \Delta \boldsymbol{w}_1^k \|^2 + \beta_1^k, \tag{27}$$

这里

$$\rho_1^k = -\sum_{t=1}^{Q}(O(t)-z^k(t))g'(\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))(\boldsymbol{w}_0^k)^{\mathrm{T}}\boldsymbol{G}'(\boldsymbol{s}^k(t))\,\delta^k(t)-$$

$$\frac{1}{2}\sum_{t=1}^{Q}(O(t)-z^k(t))g'(\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))(\boldsymbol{w}_0^k)^{\mathrm{T}}\boldsymbol{G}''(\tau^k(t))(\Delta\boldsymbol{s}^k(t))^{(2)}, \tag{28}$$

向量 $\tau^k(t)\in R^M$, 其分量落在 $\boldsymbol{s}^{k+1}(t)$ 和 $\boldsymbol{s}^k(t)$ 的相应分量之间.

再次利用 Taylor 展开公式, 且注意到(10)~(11)、(19)、(27)~(28)式, 我们有

$$E(\boldsymbol{w}^{k+1})-E(\boldsymbol{w}^k)=$$

$$-\sum_{t=1}^{Q}(O(t)-z^k(t))g'(\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))\Delta\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t)-$$

$$\sum_{t=1}^{Q}(O(t)-z^k(t))g'(\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))\boldsymbol{w}_0^k\cdot\Delta\boldsymbol{y}^k(t)+\rho_2^k=$$

$$-\frac{1}{\eta}\|\Delta\boldsymbol{w}^k\|^2+\rho_1^k+\rho_2^k, \tag{29}$$

这里

$$\rho_2^k=-\frac{1}{2}\sum_{t=1}^{Q}(O(t)-z^k(t))g''(\theta^k(t))(\boldsymbol{w}_0^{k+1}\cdot\boldsymbol{y}^{k+1}(t)-\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))^2-$$

$$\sum_{t=1}^{Q}(O(t)-z^k(t))g'(\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t))\Delta\boldsymbol{w}_0^k\cdot\Delta\boldsymbol{y}^k(t), \tag{30}$$

$\theta^k(t)$ 是 1 个实数, 落在 $\boldsymbol{w}_0^k\cdot\boldsymbol{y}^k(t)$ 和 $\boldsymbol{w}_0^{k+1}\cdot\boldsymbol{y}^{k+1}(t)$ 之间.

由假设(A1)、(A2), 引理 3.2 和引理 3.3, 存在 1 个正常数 C^* 使得

$$|\rho_i^k|\leqslant C^*\|\Delta\boldsymbol{w}^k\|^2,\qquad i=1,2. \tag{31}$$

结合(29)和(31)式导出

$$E(\boldsymbol{w}^{k+1})-E(\boldsymbol{w}^k)\leqslant-\left(\frac{1}{\eta}-2C^*\right)\|\Delta\boldsymbol{w}^k\|^2. \tag{32}$$

因此, 结论(a)是正确的, 如果学习率足够小使得

$$0<\eta<\frac{1}{2C^*}, \tag{33}$$

这里, C^* 是(31)式中的常数.

因为非负序列$\{E(\boldsymbol{w}^k)\}$单调下降且有下界, 那么一定存在 1 个极限值 $E^*\geqslant 0$, 使得 $\lim_{k\to\infty}E(\boldsymbol{w}^k)=E^*$. 因此结论(b)证得.

注意到(16)、(18)和(32)式, 得到

$$\eta\sum_{k=0}^{\infty}E_w(\boldsymbol{w}^k)=\sum_{k=0}^{\infty}\|\Delta\boldsymbol{w}^k\|^2<\infty, \tag{34}$$

因此

$$\lim_{k\to\infty}\|E_w(\boldsymbol{w}^k)\|=\lim_{k\to\infty}\|\Delta\boldsymbol{w}^k\|=0. \tag{35}$$

最后, 证明强收敛性(d). 利用引理 3.4, 取 $\boldsymbol{F}(\boldsymbol{w})=E_w(\boldsymbol{w})$. 那么, 由结论(c), 假设(A3)和(35)式, 推得结论(d). 定理证明完毕.

4 数值例子

Elman 神经网络用于学习 2 阶延迟的 XOR 问题[12]. 网络包含 2 个输入单元和 1 个偏置单元($N=3$, 如图 1), 4 个隐层单元($M=4$)和 1 个输出单元. 隐层和输出层神经元激活函数

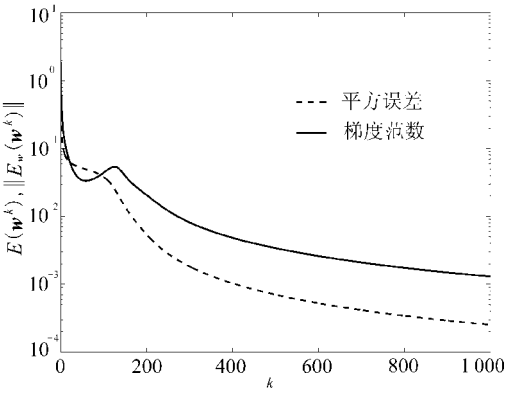

图 2 误差函数与梯度的范数变化曲线

均取为 $g(x) = (e^x - e^{-x})/(e^x + e^{-x})$. 学习率 η 设定为 0.2, 在 $[-1, 1]$ 闭区间内随机选取初始权值 $\boldsymbol{w}^0$, 当迭代步数达到 1 000 或者误差函数小于 $E < 0.000\,1$, 则停止训练.

试验结果如图 2 和表 1 所示. 从图 2, 我们看到随着迭代步数的增加, 误差单调递减, 对应梯度趋近于 0, 与收敛性结果吻合. 表 1 给出了网络训练停止后, 实际输出 $z(t)$ 与目标输出 $O(t)$ 的对比. 表 1 中行标签 no, x_i, O 和 z 分别代表样本序号, 外部输入向量的第 i 个分量, 目标输出和网络实际输出. 我们观察到, 例如, 当 no = 4 和 no = 7 时, 有相同的外部输入 $(x_1, x_2) = (0, 1)$. 由于递归性, 网络的响应是完全不同的.

表 1 训练后网络的性能

no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x_1	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1
x_2	1	0	0	1	1	0	1	1	1	0	0	1	1	0	1	0
O	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
z	0.00	0.99	0.00	0.97	0.99	0.99	0.00	0.99	0.97	0.99	0.00	0.98	0.97	0.99	0.00	0.98

5 结 论

本文证明了有限样本集上 Elman 网络梯度学习法的一些确定性收敛结果. 作为比较, 已存在的收敛结果[15]是概率性的渐进收敛性, 其假设训练样本的个数趋紧于无穷. 这个方法使用常学习率. 收敛性分析的关键是证明学习过程中误差函数的单调性, 假设权值有界和学习率足够小(见(33)式). 我们建立了一些弱收敛和强收敛结果. 弱收敛结果意味着 $\lim_{k\to\infty} \|E_{\boldsymbol{w}}(\boldsymbol{w}^k)\| = 0$. 在 $E_{\boldsymbol{w}}(\boldsymbol{w})$ 包含有限零点条件下, 我们证明了强收敛结果 $\lim_{k\to\infty} \boldsymbol{w}^k = \boldsymbol{w}^*$. 这里, $\boldsymbol{w}^*$ 是 $E(\boldsymbol{w})$ 的局部极小点. 通过数值试验演示了定理的合理性.

致谢 作者感谢编辑及审稿人富有洞察力的评论和有价值的建议, 提高了本文的质量和可读性.

[参 考 文 献]

[1] Elman J L. Finding structure in time[J]. Cognitive Science, 1990, **14**(2): 179–211.

[2] Tsoi A C, Back A D. Locally recurrent globally feedforward networks: a critical review of architeetures[J]. IEEE Transactions on Neural Networks, 1994, **5**(2): 229–239.

[3] WANG De-liang, LIU Xiao-mei, Ahalt S C. On temporal generalization of simple recurrent networks[J]. Neural Networks, 1996, **9**(7): 1099–1118.

[4] Kremer S C. On the computational power of Elman-style recurrent net works[J]. IEEE Transactions on Neural Networks, 1995, **6**(4): 1000–1004.

[5] Pham D T, Liu X. Training of elman networks and dynamic system modeling[J]. International Journal of Systems Science, 1996, **27**(2): 221–226.

[6] Cartling B. On the implicit acquisition of a context-free grammar by a simple recurrent neural network[J]. Neurocomputing, 2008, **71**(7/9): 1527–1537.

[7] LI Xiang, CHEN Zeng-qiang, YUAN Zhu-zhi, et al. Generating chaos by an Elman network[J]. IEEE Transactions on Circuits and Systems-Ⅰ, 2001, **48**(9): 1126-1131.

[8] Ekici S, Yildirim S, Poyraz M. A transmission line fault locator based on Elman recurrent networks [J]. Applied Soft Computing, DOI: 10.1016/J.asoc.2008.04.011.

[9] Neto L B, Coelho P H G, Soares de Mello J C C B, et al. Flow estimation using an Elman networks [A]. In: Wunsch D, Ed. Proceedings of 2004 IEEE International Joint Conference on Neural Networks [C]. Budapest, Hungary: IEEE Press, 2004, 831-836.

[10] Demuth H B, Beale M H, Hagan M T. Neural Network Toolbox User's Guide [M]. atick, MA: The Mathworks Inc, 2007.

[11] Jes s O D, Hagan M T. Back propagation algorithms for a broad class of dynamic networks[J]. IEEE Transactions on Neural Networks, 2007, **18**(1): 14-27.

[12] Williams R J, Zisper D. A learning algorithm for continually runningfully recurrent neural networks [J]. Neural Computation, 1989, **1**(2): 270-280.

[13] Ku C C, Lee K Y. Diagonal recurrent neural networks for dynamic systems control[J]. IEEE Transaction on Neural Networks, 1995, **6**(1): 144-156.

[14] XU Dong-po, LI Zheng-xue, WU Wei, et al. Convergence of gradient descent algorithm for diagonal recurrent neural networks[A]. In: CUI Guang-zhao, Ed. International Conference on Bio-Inspired Computing: Theories and Applications [C]. Zhengzhou, China: IEEE Press, 2007.

[15] Kuan C M, Hornik K, White H. A convergence results for learning inrecurrent neural networks[J]. Neural Computation, 1994, **6**(3): 420-440.

[16] WU Wei, FNG Guo-rui, LI Zheng-xue, et al. Convergence of an online gradient method for BP neural networks[J]. IEEE Transactionson Neural Networks, 2005, **16**(3): 533-540.

[17] WU Wei, SHAO Hong-mei, QU Di. Strong convergence for gradient methods for BP networks training[A]. In: ZHAO Ming-sheng, SHI Zhong-zhi, Eds. Proceedings of 2005 International Conference on Neural Networks and Brains [C]. Beijing, China: IEEE Press, 2005, 332-334.

[18] Gori M, Maggini M. Optimal convergence of on-line back propagation[J]. IEEE Transaction on Neural Networks, 1996, **7**(1): 251-254.

[19] Ortega J, Rheinboldt W. Iterative Solution of Nonlinear Equations in Several Variables [M]. New York: Academic Press, 1970.

[20] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 2001, 149.

Convergence of Gradient Method for Elman Networks

WU Wei, XU Dong-po, LI Zheng-xue

(Department of Applied Mathematics, Dalian University of Technology, Dalian, Liaoning 116024, P.R. China)

Abstract: The gradient method for training Elman networks with finite training sample set is considered. The monotonicity of the error function in the iteration is shown. A weak and a strong convergence results are proved, indicating that the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively. A numerical example is given to support the theoretical findings.

Key words: Elman network; gradient learning algorithm, convergence; monotonicity