第 50 卷 第 4 期 2001 年 4 月
1000-3290/2001/50(04)/0674-08

物 理 学 报
ACTA PHYSICA SINICA

Vol. 50, No. 4, April, 2001


# 神经网络的自适应删剪学习算法及其应用*

陈 成 常胜江 袁景和 张延炘

(南开大学现代光学研究所,教育部光学信息技术科学开放实验室,天津 300071)

K. W. Wong

(香港城市大学电子工程系,香港,中国)

(2000 年 5 月 28 日收到;2000 年 7 月 17 日收到修改稿)

在局域卡尔曼滤波算法的基础上,提出了一种自适应删剪学习算法,这一算法的核心是用网络训练结束后得到的局域的误差协方差矩阵测量权重的重要性,通过删除不重要的权重,得到一个紧凑的网络结构. 广义异或逻辑函数和手写体数字识别的计算机模拟结果显示该方法是一种有效的网络规模优化算法.

关键词:神经网络,模式识别,广义卡尔曼滤波,删剪

PACC:4230,8730

## 1 引 言

近年来,神经网络的研究大都致力于解决复杂的现实问题[1,2]. 反向传播算法(BP 算法)是一种常用的训练前馈型神经网络的算法. 这种算法的缺点是收敛速度慢,学习过程中,如果参数选择不当,会导致无法收敛或者使训练陷入局域最小[3]. 一般地,解决这两个缺陷的方法是使用二阶算法.

计算前馈型网络的权重以实现一个理想的输入/输出映射可以被视为一个高阶非线性系统的辨识问题,这一问题可以由广义卡尔曼滤波(extended Kalman filter,缩写为 EKF)算法解决[4,5]. 这种算法属于二阶梯度算法,与传统的 BP 算法相比,卡尔曼滤波算法中需要调整的参数较少,使得它的学习更为容易,而且,由于卡尔曼滤波算法自动估算出权重调整的最佳步长值(即卡尔曼增益),所以网络的收敛速度及效率都会有较大提高. 但是,由于误差协方差矩阵的规模是由权重的总数目决定且在每一次迭代过程中都必须调整,因而其计算复杂度即使对中等规模的网络也是过高的. 为了降低计算的复杂度和提高学习速度,人们提出了局域的卡尔曼滤波算法[6]. 然而,如何把局域卡尔曼滤波算法与权重删剪算法结合起来,在网络训练的同时也能优化网络的规模大小仍是一个没有解决的问题.

在神经网络研究中,除了寻找适当的网络权重,另一个重要的问题是如何确定适当的网络规模以避免网络规模过大引起的过度训练(overtraining)现象出现[7],特别当训练样本含有噪声或数目较少时. 解决此问题的有效方法是使用权重删剪方法,人们已提出了几种权重删剪学习算法,如:optimal brain damage(缩写为 OBD)[8],optimal brain surgeon(缩写为 OBS)[9]和基于全局卡尔曼滤波的删剪算法等方法[10]. 前两种方法需要计算 Hessian 矩阵,而计算这个矩阵需要额外的输入、输出样本对才能获得,这两种方法不适用于在线学习情况. 与上述两种方法相比,全局的卡尔曼滤波算法没有这方面的缺陷,但是,由于协方差矩阵必须在学习及删剪过程中计算,而此矩阵的规模是由权重的总数目决定的,所以计算的复杂度即使对中等规模的网络而言也是非常高的. 当我们考虑文献[11]中的应用例子时就会清楚这一点. 此例中,用于手写体数字识别的神经网络有 6860 个权重. 全局卡尔曼滤波算法训练和删剪的计算复杂度分别为 $O(6860^2)$ 和 $O(6860^3)$. 这样大的计算复杂度使得卡尔曼滤波算法很难应用. 因此,对于卡尔曼滤波算法来说,减少训练和删剪的计算复杂度是其研究的一个重点.

本文在一个局域卡尔曼滤波算法基础上,提出了一种自适应删剪学习算法,这一算法的核心是用训练结束后的局域误差协方差矩阵测量权重的重要

* 国家自然科学基金(批准号:69877005)资助的课题.

性,通过删除不重要的权重,得到一个紧凑的网络结构.我们把此算法应用于二个识别问题:广义异或逻辑函数和手写体数字识别问题.计算机模拟的结果表明该算法是一种有效的优化网络规模的算法.同时,复杂度的计算表明该方法能够有效减少解决大规模实际问题时对内存和存贮空间的要求.

## 2　基于局域卡尔曼滤波的学习和删剪算法

### 2.1　符号说明

不失一般性,我们考虑一个 $M$ 层前馈网络,它由一个线性输入层、几个非线性隐藏层和一个输出层组成.为方便起见,将本文所用符号列在下面:

$N_n$ 为第 $n$ 层的神经元总数(包括阈值单元);

$x_i^1(t)$ 为输入模式 $t$ 的第 $i$ 个元素;

$x_i^n(t)$ 为对于模式 $t$ 的第$(n,i)$(第 $n$ 层第 $i$ 个神经元)个神经元的输出;

$d_i(t)$ 为对于模式 $t$ 的输出层第 $i$ 个神经元的理想输出;

$w_{i,j}^n(t)$ 为神经元$(n,j)$与$(n+1,i)$间的互连权重;

$net_i^n(t)$ 为神经元$(n,i)$的净输入;

$N_w$ 为权重的总数目;

$\theta_i^n$ 为神经元$(n,i)$的偏置.

神经元$(n+1,i)$的输出由下式给出:

$$x_i^{n+1}(t) = f\Big(\sum_{j=1}^{N_n-1} w_{i,j}^n x_j^n(t) + \theta_i^{n+1}\Big) = f\Big(\sum_{j=1}^{N_n} w_{i,j}^n x_j^n(t)\Big), \tag{1}$$

其中 $f(\cdot)$ 是神经元的激励函数.本文中,激励函数采用 sigmoid 函数:

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{2}$$

或双曲正切函数:

$$f(x) = \frac{1-\mathrm{e}^{-x}}{1+\mathrm{e}^{-x}}. \tag{3}$$

第 $n$ 层的输出矢量 $\boldsymbol{x}^n(t)$、理想的输出矢量 $\boldsymbol{d}(t)$ 以及互连权重 $\boldsymbol{w}_i^n$ 分别写成下列形式:

$$\boldsymbol{x}^n(t) = [x_1^n(t), x_2^n(t), \cdots, x_{N_n}^n(t)]^T \quad (N_n \times 1), \tag{4}$$

$$\boldsymbol{d}(t) = [d_1(t), d_2(t), \cdots, d_{N_M}(t)]^T \quad (N_M \times 1), \tag{5}$$

$$\boldsymbol{w}_i^n = [w_{i,1}^n, w_{i,2}^n, \cdots, w_{i,N_n}^n]^T \quad (N_n \times 1), \tag{6}$$

$$\boldsymbol{w}^n = [(\boldsymbol{w}_1^n)^T, (\boldsymbol{w}_2^n)^T, \cdots, (\boldsymbol{w}_{N_{n+1}}^n - 1)^T]^T \quad (N_n(N_{n+1}-1) \times 1), \tag{7}$$

$$\boldsymbol{w} = [(\boldsymbol{w}^1)^T, (\boldsymbol{w}^2)^T, \cdots, (\boldsymbol{w}^{M-1})^T]^T \quad (N_w \times 1). \tag{8}$$

### 2.2　局域 EKF 训练算法

互连权重矢量 $\boldsymbol{w}$ 被看作一个稳态非线性动力学系统的状态,假定第 $t$ 个训练模式送入网络,并且假定所有权重分矢量是已知的,只有 $\boldsymbol{w}_i^n$ 为未知的,则系统的状态可以由下述方程描述:

$$\boldsymbol{w}_i^n(t) = \boldsymbol{w}_i^n(t-1) + \boldsymbol{v}_i^n(t) \tag{9}$$

$$\boldsymbol{d}(t) = \boldsymbol{h}(\boldsymbol{w}_i^n(t), \boldsymbol{x}^1(t)) + \boldsymbol{\varepsilon}(t) = \boldsymbol{x}^M(t) + \boldsymbol{\varepsilon}(t), \tag{10}$$

其中 $h(\boldsymbol{w}_i^n(t), \boldsymbol{x}^1(t))$ 是描述系统的时变函数,$\boldsymbol{\varepsilon}(t)$ 和 $\boldsymbol{v}_i^n(t)$ 是平均值为零、协方差为 $\boldsymbol{R}(t)$ 和 $\boldsymbol{Q}_i^n(t)$ 的噪声.

网络参数 $\boldsymbol{w}_i^n$ 的估算可以利用标准 EKF 方法获得

$$\hat{\boldsymbol{w}}_i^n(t) = \hat{\boldsymbol{w}}_i^n(t-1) + \boldsymbol{K}_i^n(t)[\boldsymbol{d}(t) - \hat{\boldsymbol{x}}^M(t)], \tag{11}$$

$$\boldsymbol{K}_i^n(t) = [\boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1)] \cdot \boldsymbol{H}_i^n(t)^T \{\boldsymbol{H}_i^n(t)[\boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1)]\boldsymbol{H}_i^n(t)^T + \boldsymbol{R}(t)\}^{-1}, \tag{12}$$

$$\boldsymbol{P}_i^n(t) = \boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1) - \boldsymbol{K}_i^n(t)\boldsymbol{H}_i^n(t)[\boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1)], \tag{13}$$

其中 $\boldsymbol{K}_i^n$ 为卡尔曼增益(Kalman gain),$\boldsymbol{P}_i^n$ 为误差协方差矩阵,$\hat{\boldsymbol{x}}^M$ 和 $\hat{\boldsymbol{w}}_i^n$ 分别是网络输出和权重的估算值,$\boldsymbol{H}_i^n$ 为梯度矩阵,它由 $\boldsymbol{x}^M(t)$ 在 $\hat{\boldsymbol{w}}_i^n(t-1)$ 处线性化给出.

$$\boldsymbol{H}_i^n(t) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{w}_i^n)}{\partial \boldsymbol{w}_i^n}\right|_{\boldsymbol{w}_i^n = \hat{\boldsymbol{w}}_i^n(t-1)}. \tag{14}$$

### 2.3　基于局域 EKF 的删剪算法

利用矩阵变换定理,

$$(\boldsymbol{I} + \boldsymbol{PHR}^{-1}\boldsymbol{H}^T)^{-1}\boldsymbol{P} = \boldsymbol{P}^{-1} + \boldsymbol{HR}^{-1}\boldsymbol{H}^T)^{-1}$$

$$= \boldsymbol{P} - \boldsymbol{PH}(\boldsymbol{H}^T\boldsymbol{PH} + \boldsymbol{R})^{-1}\boldsymbol{H}^T\boldsymbol{P},$$

以及 $\boldsymbol{P} = [\boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1)]$和 $\boldsymbol{H} = \boldsymbol{H}_i^n(t)^T$，(13)式可以重写为

$$\boldsymbol{P}_i^n(t) = \{[\boldsymbol{P}_i^n(t-1) + \boldsymbol{Q}_i^n(t-1)]^{-1} + \boldsymbol{H}_i^n(t)^T\boldsymbol{R}(t)^{-1}\boldsymbol{H}_i^n(t)\}^{-1}. \quad (15)$$

假设训练在 $t_0$ 次迭代后收敛，相应的协方差矩阵 $\boldsymbol{P}_i^n(t)$也收敛到一个稳定值 $\boldsymbol{P}_i^n(\infty)$，则(15)式可表示为

$$\boldsymbol{P}_i^n(\infty)^{-1}\boldsymbol{Q}_i^n(t-1)[\boldsymbol{P}_i^n(\infty) + \boldsymbol{Q}_i^n(t-1)]^{-1} = \boldsymbol{H}_i^n(t)^T\boldsymbol{R}(t)^{-1}\boldsymbol{H}_i^n(t), \quad (16)$$

其中

$$\boldsymbol{P}_i^n(\infty) = \lim_{t\to\infty}\boldsymbol{P}_i^n(t). \quad (17)$$

如果训练在 $T$ 时刻停止，且 $T > t_0$，则可以推断

$$\frac{1}{T - t_0}\sum_{t_0=t_0+1}^{T}\boldsymbol{P}_i^n(\infty)^{-1}\boldsymbol{Q}_i^n(t-1)[\boldsymbol{P}_i^n(\infty) + \boldsymbol{Q}_i^n(t-1)]^{-1} = \frac{1}{T - t_0}\sum_{t_0=t_0+1}^{T}\boldsymbol{H}_i^n(t)^T\boldsymbol{R}(t)^{-1}\boldsymbol{H}_i^n(t). \quad (18)$$

矩阵 $\boldsymbol{Q}_i^n(t)$和 $\boldsymbol{R}(t)$是先前未知的，为简单起见，把它们都设为对角形式，如：

$$\boldsymbol{Q}_i^n(t) = \delta\boldsymbol{I}_{N_n\times N_n} \quad (19)$$

和

$$\boldsymbol{R}(t) = \boldsymbol{I}_{N_M\times N_M}, \quad (20)$$

其中 $\boldsymbol{I}$ 为单位矩阵，$\delta$ 为一常数. 当 $T$ 很大时，(18)式等号右边的项将趋近 $\boldsymbol{H}(t)^T\boldsymbol{H}(t)$的期望值，即

$$\delta\boldsymbol{P}_i^n(\infty)^{-1}[\boldsymbol{P}_i^n(\infty) + \delta\boldsymbol{I}]^{-1} = \boldsymbol{E}[\boldsymbol{H}_i^n(t)^T\boldsymbol{H}_i^n(t)]. \quad (21)$$

由于协方差矩阵是准对称正定的，因而它可以分解为以下形式：

$$\boldsymbol{P}_i^n(\infty) = \boldsymbol{FGF}^{-1}, \quad (22)$$

其中 $\boldsymbol{G}$ 为对角矩阵，它的对角矩阵元是 $\boldsymbol{P}_i^n(\infty)$的本征值，而 $\boldsymbol{F}$ 是由相应的本征矢量组成，由此，(21)式可以被表示为

$$\begin{aligned}\boldsymbol{E}[\boldsymbol{H}_i^n(t)^T\boldsymbol{H}_i^n(t)] &= \delta(\boldsymbol{FGF}^{-1})^{-1}(\boldsymbol{FGF}^{-1} + \boldsymbol{F}\delta\boldsymbol{IF}^{-1})^{-1},\\ &= \delta[(\boldsymbol{FGF}^{-1} + \boldsymbol{F}\delta\boldsymbol{IF}^{-1})(\boldsymbol{FGF}^{-1})]^{-1},\\ &= \delta[(\boldsymbol{FG}(\boldsymbol{G} + \delta\boldsymbol{I})\boldsymbol{F}^{-1}]^{-1},\\ &= \boldsymbol{F}\left[\frac{\delta}{\boldsymbol{G}(\boldsymbol{G} + \delta\boldsymbol{I})}\right]\boldsymbol{F}^{-1}. \quad (23)\end{aligned}$$

对于网络能收敛的 $\delta$ 值一般是很小的，$\boldsymbol{P}_i^n(\infty)$的最大本征值要远大于 $\delta$，因而(23)式可近似为

$$\begin{aligned}\boldsymbol{E}[\boldsymbol{H}_i^n(t)^T\boldsymbol{H}_i^n(t)] &\approx \boldsymbol{F}[\delta\boldsymbol{G}^{-2}]\boldsymbol{F}^{-1},\\ &= \delta[\boldsymbol{FG}^2\boldsymbol{F}^{-1}]^{-1},\\ &= \delta\boldsymbol{P}_i^n(\infty)^{-2}. \quad (24)\end{aligned}$$

将 $\boldsymbol{H}_i^n(t)$的定义式(14式)代入(24)式，得到方程

$$E\left[\left(\frac{\partial \boldsymbol{h}}{\partial w_{ik}^n}\right)^2\right] \approx \delta[\boldsymbol{P}_i^n(\infty)^{-2}]_{kk}, \quad (25)$$

其中$[\boldsymbol{P}_i^n(\infty)^{-2}]_{kk}$表示它的第 $k$ 个对角元素.

另一方面，训练后网络的方差期望值可以表示为

$$E[(d - \hat{x}^M)^2] = E[(\boldsymbol{h}(w_i^n) + \boldsymbol{\varepsilon} - \boldsymbol{h}(\hat{w}_i^n))^2]. \quad (26)$$

假定 $\hat{w}_i^{n*}$ 是当 $\hat{w}_i^n$ 的第 $k$ 个元素(即 $\hat{w}_{ik}^n$)被置为 0 时相应的权重向量，$\hat{x}^{M*}$ 为相应的网络输出，假定网络输出的误差与噪声项是相互独立的，则方差期望值可以写成

$$\begin{aligned}E[(d - \hat{x}^{M*})^2] &= E[(\boldsymbol{h}(w_i^n) + \boldsymbol{\varepsilon} - \boldsymbol{h}(\hat{w}_i^{n*}))^2]\\ &= E[(\boldsymbol{h}(w_i^n) + \boldsymbol{\varepsilon} - \boldsymbol{h}(\hat{w}_i^n) + \boldsymbol{h}(\hat{w}_i^n) - \boldsymbol{h}(\hat{w}_i^{n*}))^2]\\ &\approx E[(\boldsymbol{h}(w_i^n) + \boldsymbol{\varepsilon} - \boldsymbol{h}(\hat{w}_i^n))^2]\\ &\quad + E[(\boldsymbol{h}(\hat{w}_i^n) - \boldsymbol{h}(\hat{w}_i^{n*}))^2]. \quad (27)\end{aligned}$$

由于 $\hat{w}_i^n$ 的第 $k$ 个元素被删除引起的误差增量值为

$$\begin{aligned}\Delta E_k^n &= E[(d - \hat{x}^{M*})^2] - E[(d - \hat{x}^M)^2]\\ &\approx E[(\boldsymbol{h}(\hat{w}_i^n) - \boldsymbol{h}(\hat{w}_i^{n*}))^2]. \quad (28)\end{aligned}$$

对(28)式等号右边项进行 Taylor 展开，忽略掉高次项后可表示为

$$\begin{aligned}\Delta E_k^n &= E[(\boldsymbol{h}(\hat{w}_i^n) - \boldsymbol{h}(\hat{w}_i^{n*}))^2]\\ &\approx E\left[\left(\frac{\partial \boldsymbol{h}(\hat{w}_i^n)}{\partial w_{ik}^n}\right)^2\right](\hat{w}_{ik}^n)^2\\ &\approx \delta[\boldsymbol{P}_i^n(\infty)^{-2}]_{kk}(\hat{w}_{ik}^n)^2. \quad (29)\end{aligned}$$

(29)式描述了删掉权重 $\hat{w}_{ik}^n$所引起网络输出误差变化的大小，根据此变化的大小，可以判断权重 $\hat{w}_{ik}^n$的重要性. 基于以上的分析，删剪过程如下进行：

1) 利用 EKF 方程训练网络，$i$ 以递增的顺序，$n$ 以递减的顺序；

2) 用(29)式对所有权重的重要性进行估算；

3) 根据删剪权重对 $\Delta E_k{}^n$ 值变化的大小，对所有权重的重要性进行排列，排列的方法为 $\{s_1, s_2, \cdots, s_{N_w}\}$，其中 $\Delta E_{s_i} \leqslant \Delta E_{s_j}$ $(i < j)$；

4) 令 $\Delta\hat{w}_{s_k} = \hat{w}_{s_k}$ (当 $k = 1, \cdots, k'$)，且 $\Delta\hat{w}_{s_k} = 0$ (当 $k > k'$)，用 $\Delta\hat{w}_{s_k}$ 按方程(6)—(8)重新构成 $\Delta\hat{w}_i^n$；

5) 估算由于权重(从 $w_{s_1}$到 $w_{s_{k'}}$)被删除引起的平均期望误差的增量变化

$$\Delta E_{[s_1, s_{k'}]} = \sum_{n=1}^{M-1}\sum_{i=1}^{N_n}\delta(\Delta\hat{w}_i^n)^T\boldsymbol{P}_i^n(\infty)^{-2}(\Delta\hat{w}_i^n). \quad (30)$$

6) 如果这个估算的误差的变化量不超过阈值(如 $\Delta E < \text{threshold}$),则定义 $k' = k' + 1$,转到步骤 4) 否则停止,删掉从 $w_{s_1}$ 到 $w_{s_{k'-1}}$ 的 $k'-1$ 个权重.

## 3　计算机模拟

### 3.1　广义 XOR 问题

首先用广义 XOR 问题来检验上述算法的有效性.所用的网络有两个输入单元,20 个隐藏层神经元和一个输出神经元.隐藏层和输出层的神经元的激活函数均为双曲正切函数(见(3)式).权重的总数目为 81.设初始的协方差矩阵 $\boldsymbol{P}_i^n(0) = I$,$\delta = 0.0001$.初始权重在区间 $[-0.2, 0.2]$ 之间随机选取.在模拟中,随机数产生器产生 6100 对随机数,其中 6000 对用于网络训练,另 100 对用于测试.当训练收敛之后,送入 100 对测试数据,平均测试误差可由下式得到

$$E_{\text{test}} = \frac{1}{100}\sum_{t=1}^{100}[\boldsymbol{d}(t) - \hat{\boldsymbol{x}}^M(t)]^2. \qquad (31)$$

当 $k$ 个权重 $w_{s_1}$ 到 $w_{s_k}$ 被删除后,实际的平均测试误差由下述公式计算而得:

$$E_{[s_1,s_k]} = \frac{1}{100}\sum_{t=1}^{100}[\boldsymbol{d}(t) - \hat{\boldsymbol{x}}^M(t)]^2. \qquad (32)$$

由删除权重引起的误差实际变化量由下式计算:

$$\Delta E_{\text{actual}} = E_{[s_1,s_k]} - E_{\text{test}}. \qquad (33)$$

将由删除权重引起的误差实际变化量和根据(30)式估算的误差变化量 $\Delta E_{[s_1,s_k]}$ 作比较,其结果如图 1 所示.它表明估算值和实际值非常接近.图 2 显示了估算的误差变化量 $\Delta E_{[s_1,s_k]}$、实际变化量 $\Delta E_{\text{actual}}$ 与裁减的权重数目的关系,它同样显示了估算值与实际值在 $k$ 小于 53 时非常接近.此图表明用基于局域 EKF 的裁减方法删除 67% 的权重不会使误差明显增加.

### 3.2　手写体数字识别

在这里,我们把上述删剪学习算法和视觉学习规则结合起来,以解决手写体数字不变性特征提取及识别问题.所用的网络结构为一个 3 层前馈型网络.基于 EKF 的删剪学习算法和迹(trace)学习规则结合用来训练和删剪输入层与隐藏层之间的网络权重,在训练中自组织地提取不变特征信息.隐藏层与输出层之间权重的训练由标准的 EKF 删剪学习算

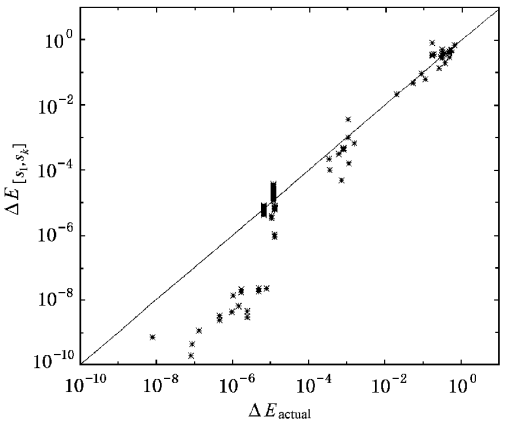



图 1　广义 XOR 问题中的估算误差变化值 $\Delta E_{[s_1,s_k]}$(30)式与实际值 $\Delta E_{\text{actual}}$ 关系图

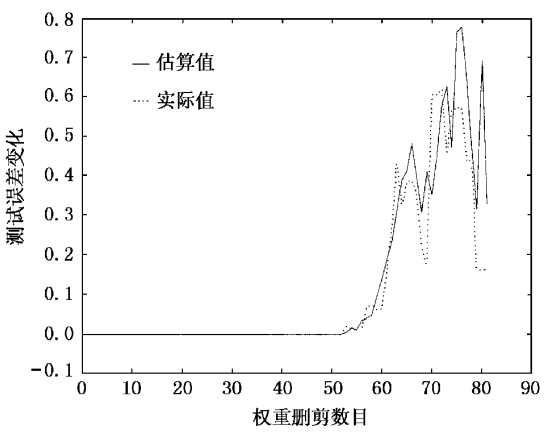



图 2　测试误差变化与权重删剪数目关系图

法完成.迹学习规则[12,13]首先是用于自组织的不变性替换,又推广应用于神经网络的学习中.由于输入模式的不变性特征提取是由隐藏层完成的,我们定义第 $i$ 个隐藏层神经元对第 $m$ 类输入模式的迹为:$\mathrm{T}_{mi}(t) = \eta \mathrm{T}_{mi}(t-1) + (1-\eta)x_i^h(t)$[12],其中 $\eta$ 是一个控制参数,用以控制迹和新的输入之间的相互影响,$x_i^h(t)$ 是第 $i$ 个隐藏神经元的输出.当隐藏层的训练收敛后(迹稳定),输出层利用提取的不变性特征对输入模式进行归类(识别).在这个网络中,第 $i$ 个隐藏层神经元只与 $L_i$ 个输入层神经元连接,是局域互连,而隐藏层到输出层是全互连的.

在隐藏层对输入模式进行不变性提取中,我们使用下述系统方程来描述这个非线性系统:

$$\boldsymbol{w}_i^n(t) = \boldsymbol{w}_i^n(t-1) + \boldsymbol{v}_i^n(t), \qquad (34)$$

$$\begin{aligned}\boldsymbol{e}(t) &= \frac{1}{\eta}[\boldsymbol{T}(t) - \boldsymbol{x}^h(t)] \\ &= \boldsymbol{T}(t-1) - \boldsymbol{x}^h(t).\end{aligned} \qquad (35)$$

与第 2 节的推导类似,局域 EKF 和迹的迭代方程可表示为

$$\hat{\boldsymbol{w}}_i^h(t) = \hat{\boldsymbol{w}}_i^h(t-1) + \boldsymbol{K}_i^h(t) \cdot [\boldsymbol{T}(t-1) - \hat{\boldsymbol{x}}^h(t)], \tag{36}$$

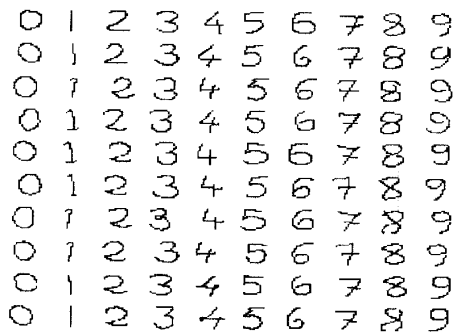
$$\boldsymbol{K}_i^h(t) = [\boldsymbol{P}_i^h(t-1) + \boldsymbol{Q}(t)]\boldsymbol{H}_i^h(t)^T \cdot \{\boldsymbol{H}_i^h(t)[\boldsymbol{P}_i^h(t-1) + \boldsymbol{Q}(t)] \cdot \boldsymbol{H}_i^h(t)^T + \boldsymbol{R}^h(t)\}^{-1}, \tag{37}$$

$$\boldsymbol{P}_i^h(t) = [\boldsymbol{P}_i^h](t-1) + \boldsymbol{Q}(t)] - \boldsymbol{K}_i^h(t)\boldsymbol{H}_i^h(t) \cdot [\boldsymbol{P}_i^h(t-1) + \boldsymbol{Q}(t)], \tag{38}$$

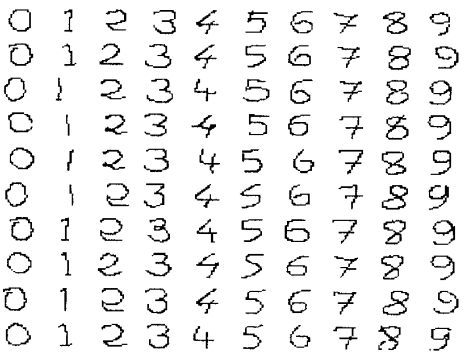
$$T_i(t) = \eta T_i(t-1) + (1-\eta)\hat{x}_i^h(t). \tag{39}$$

隐藏层根据(36)—(39)式自组织地进行不变性特征提取.最后的分类由隐藏层和输出层完成,用标准的 EKF 删剪学习算法进行训练.

本文采用一个小规模的手写体数字数据库[13]作为来考察网络性能的样本.这些数字是从一个更大的数据库[14]中提取出来的.图 3 给出了所有的 200 个样本,其中 100 个样本作为训练样本,另 100 个样本作为测试样本,每一数字由 $20\times20$ 个像素构成.在模拟中,网络规模为 $N_1 = 20\times20$, $N_2 = 14\times14$, $N_3 = 1\times10$.每个隐藏层神经元与 $L_i = 5\times5$ 个输入神经元有互连,即接收场为 $5\times5$,网络权重的总数目为 6860.令 $P_i^2(0) = P_i^3(0) = I$, $\eta = 0.6$—0.7.权重的初始值在区间[-0.1,0.1]中随机选取的.迹的初始值 $T(0)$定为零.图 4(a),(b)显示了实际的训练误差、估算的误差值和测试样本产生的误差与隐藏层和输出层中权重删剪数目的关系.从图 4 中可以看出,估算的方差非常接近实际的方差值.图 5(a),(b)给出了训练样本和测试样本的识别率与隐藏层和输出层中权重删剪数目的关系.

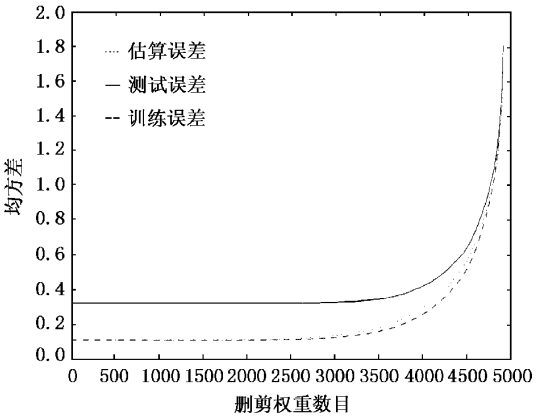


(a) 训练集

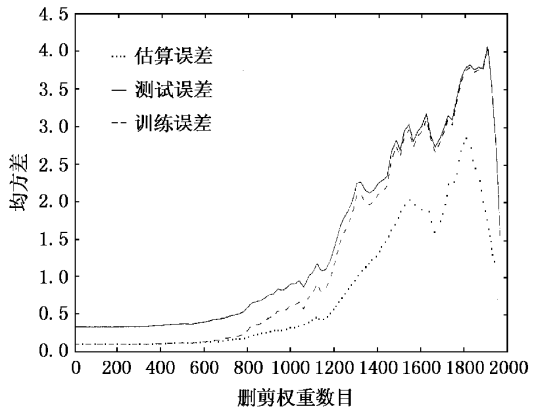


(b) 测试集

图 3 手写体数字




(a)为隐藏层



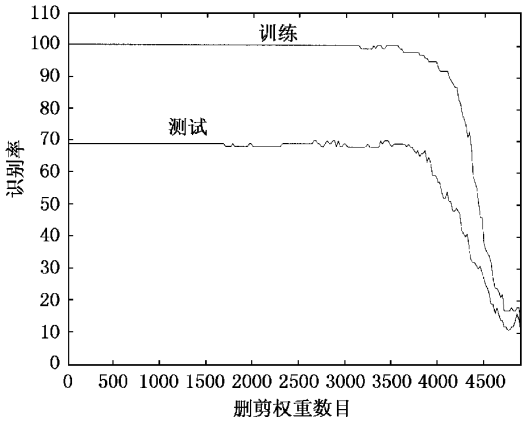

(b)为输出层

图 4 估算误差、测试误差、训练误差与删剪权重数目的关系图

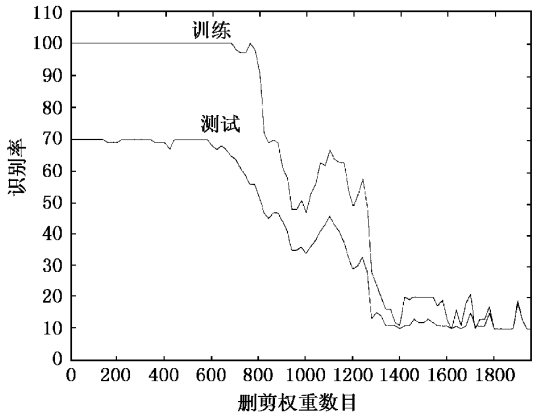
表 1 给出了网络性能的一些数据,包括学习过程的迭代次数、当4040个权重(隐藏层3460个,输

表 1 四种网络的性能(第一、二种网络使用一个相同的数据库,第三、第四种网络采用另一个数据库)

| | 网络模型 | 迭代次数 | 训练集的识别率/% | 测试集的识别率/% | 所需权重数目 |
|---|---|---|---|---|---|
| 1 | 局域 EKF 训练和删剪算法 | ~200 | 100 | 70 | 2820 |
| 2 | Wallis[12] | ~70 | 95 | 55 | 10400 |
| 3 | Peng[10] | ~500 | 95 | 64.83 | 6860 |
| 4 | 背传(BP)网络[10] | ~500 | 95 | 57.92 | 6860 |



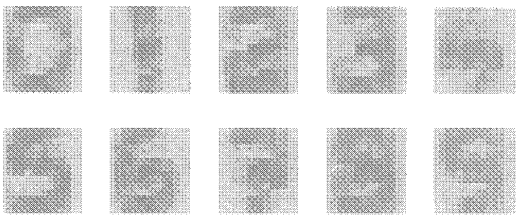

(a)为隐藏层



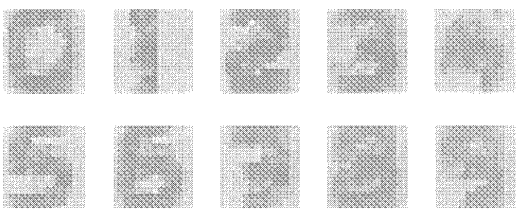

(b)为输出层

图 5　识别率与删剪权重数目的关系图

出层 580 个)被删除后训练和测试样本的识别率. 为了作比较,表 1 中还给出了其他算法的结果,如 Wallis 在文献[13]报道的同一数据库的结果,Peng 等在文献[11]中使用的稀疏迹神经网络的结果以及 BP 网络的结果. 其中 Peng 的模拟是基于另一个不同的数据库,但训练集和测试集的大小与本文是相同的. 用我们的算法所得到的结果要优于其他三种算法. 图 6(a),(b)给出了稳定的迹和测试样本被送入训练后的网络时隐藏层神经元输出的模式. 它们高度的相似性表明我们提出的算法能够有效地提取输入模式的不变特征.



(a)为网络收敛后的迹



(b)为手写体数字输入网络时的输出

图 6

## 4　计算复杂度和存贮要求

为了论证本文提出的训练和删剪方法比全局 EKF 算法在解决实际问题时更具有实用性,我们比较了两种方法的计算复杂度和存储要求. 在训练过程中,局域 EKF 方法每一次迭代的计算复杂度为 $O(\sum_{n=1}^{M-1}(N_n)^2N_{n+1})$,在删减过程中,局域 EKF 方法的计算复杂度为 $O(\sum_{n=1}^{M-1}(N_n)^3N_{n+1})$. 相应的训练和删减的存储要求都是

$$O\left(\sum_{n=1}^{M-1}(N_n)^2N_{n+1}\right).$$

全域 EKF 训练和删剪的计算复杂度分别为 $O((N_w)^2)$和 $O((N_w)^3)$,而相应的训练和删剪的存储要求都是 $O((N_w)^2)$. 表 2 按顺序给出了对上述二个问题每次迭代中训练、删剪网络所要求的计算复杂度. 从表 1 可以看出,全域 EKF 算法与局域算法相比,有高的多的计算复杂度和存储要求,对大尺度问题更是这样. 局域 EKF 算法较低的计算复杂度和存储要求使得它更适合于解决大尺度问题.

表 2　全局和局域 EKF 训练和删剪算法的计算复杂度、存贮容量要求的比较

| 问题 | 局域 EKF 训练和删剪算法 | | | | 全局 EKF 训练和删剪算法 | | | |
|---|---|---|---|---|---|---|---|---|
| | 计算复杂度 | | 存贮容量要求 | | 计算复杂度 | | 存贮容量要求 | |
| | 训练(每次迭代) | 删剪 | 训练 | 删剪 | 训练(每次迭代) | 删剪 | 训练 | 删剪 |
| GXOR | $O(6.2\times10^2)$ | $O(9.8\times10^3)$ | $O(6.2\times10^2)$ | $O(6.2\times10^2)$ | $O(6.6\times10^3)$ | $O(5.3\times10^5)$ | $O(6.6\times10^3)$ | $O(6.6\times10^3)$ |
| HDR | $O(5.0\times10^5)$ | $O(7.8\times10^7)$ | $O(5.0\times10^5)$ | $O(5.0\times10^5)$ | $O(4.7\times10^7)$ | $O(3.2\times10^{11})$ | $O(4.7\times10^7)$ | $O(4.7\times10^7)$ |

注:GXOR 为广义 XOR 问题;HDR 为手写体数字识别.

## 5 结 论

本文提出了一个用于训练和删剪前馈网络的局域 EKF 算法.此方法的核心是使用协方差矩阵中的块来度量权重的重要性并删除不重要的权重.我们把这一算法直接应用于广义 XOR 问题.同时,这个算法还与迹学习规则结合用于自组织地提取手写体数字的不变特征.全域 EKF 和局域 EKF 方法的比较表明,局域 EKF 训练和删剪方法的计算复杂度和存贮要求远比全域方法要低,特别是大尺度问题,这在解决现实的复杂问题时是非常有益的.而且,计算机模拟表明本方法能有效的删除作用不显著的权重.


[1] S. J. Chang *et al.*, *Acta Phys. Sin.*, **47**(1998), 1102 (in Chinese) [常胜江等,物理学报,47(1998),1102].

[2] J. Y. Shen *et al.*, *Acta Phys. Sin.*, **47**(1998),1966(in Chinese) [申金媛等,物理学报,47(1998),1966].

[3] J. S. Zhang, X. C. Xiao, *Chin. Phys.*, **9** (2000), 408.

[4] Y. Iigunl, H. Sakai, H. Tokumaru, *IEEE Trans. Signal Processing*, **40**(1992), 959.

[5] S. Singhal, L. Wu, Proc. IEEE Int. Conf. Acoust, Speech and Signal Processing, (1989),p.1187.

[6] S. Shah, F. Palmieri, M. Datum, *Neural Networks*, **5** (1992),779.

[7] R. Reed, *IEEE Trans. Neural Networks*, **4**(1993), 740.

[8] Y. LeCun *et al.*, Advances in Neural Information Processing System,Ed. D. S. Touretsky (1990),p. 396.

[9] B. Hassibi, D. G. Stork, Avances in Neural Information Processing System Eds. Hanson *et al.*, (1993), p.164.

[10] J. Sum, C. S. Leung, G. H. Young, W. K. Kan, *IEEE Trans. Neural Networks*, **10**(1999), 161.

[11] H. Peng, L. Sha, Q. Gan, Y. Wei, *Electr. Lett.*, **34** (1998), 292.

[12] P. Foldiak, *Neural Computation*, **3**(1991), 194.

[13] G. Wallis, *Neural Networks*, **9**(1996), 1513.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, *Neural Computation*, **1** (1989), 541.

# ADAPTIVE TRAINING AND PRUNING FOR NEURAL NETWORKS: ALGORITHMS AND APPLICATION*

CHEN SHU　CHANG SHENG-JIANG　YUAN JING-HE　ZHANG YAN-XIN
(*Institute of Modern Optics, Nankai University, Laboratory of Optical Information Science, Chinese Ministry of Education, Tianjin* 300071, *China*)
K. W. WONG
(*Department of Electronics Engineering, City University of Hong Kong, Hong Kong, China*)
(Received 28 May 2000; revised manuscript received 17 July 2000)

ABSTRACT

Finding an optimal network size is one of the major concerns when building a neural network. In using the local extended Kalman filter (EKF) algorithm, we propose an efficient approach that combines EKF training and pruning as a whole. In particular, the covariance matrix obtained along with the local EKF training can be utilized to indicate the importance of the network weights. As a result, the network size can be determined adaptively to keep pace with the changes in input characteristics. The effectiveness of this algorithm is demonstrated on generalized XOR logic function and handwritten digit recognition.

**Keywords**: neural networks, pattern recognition, extended Kalman filtering, pruning
**PACC**: 4230, 8730

* Project supported by the National Natural Science Foundation of China (Grant No. 69877005).