使用三层神经元网络的先验信息新学习方法*

吕柏权[1**] 村田纯一[2] 平泽宏太郎[2]

(1. 上海大学自动化系控制与决策教研室, 上海 200027; 2.九州大学信息科学研究院, 福冈, 日本)

摘要 提出一种使用三层神经元网络的先验信息的新学习方法. 通常, 当神经元网络被用于函数逼近时, 没有考虑权之间的关系而独立地学习, 这样, 学习的结果往往不理想. 其原因是在学习中权是相互影响的. 为了克服这一问题, 首先, 给出了一些有关权的先验信息, 然后基于此提出部分权学习和其余权由精确数学方程计算的新学习方法. 这方法在权的学习中几乎保持精确数学结构. 另外, 使用不等式先验信息的学习方法也被提出了. 无论使用不等式还是等式先验信息的学习, 因网络权的自由度被限制而加快了学习速度并保证误差较小. 数值仿真的结果支持提出方法.

关键词 先验信息 神经元网络学习 部分参数学习 精确数学结构

近年来, 人工神经元网络已在许多方面广泛地被应用. 大部分应用使用前馈网络和误差反向传播学习方法. 但在前馈网络的反向传播学习算法中, 存在四种彼此相互关联的问题: 收敛速度、误差的局部极小、学习后网络的泛化能力和目标函数与神经元网络的结构的关系.

先验信息在设定网络结构和约束神经元网络权方面是非常有效的, 它可帮助网络权快速地收敛全局最小点和改善网络的泛化能力, 对于解决上述四种问题是非常重要的. 我们知道神经元网络结构在克服局部最小点以及改善网络的泛化能力和减少计算量方面起着重要作用, 不用先验信息来设计神经元网络结构的一些方法如各种遗传算法等计算时间太长. 因为通常的学习算法在某种意义上是并行计算, 每个权是独立修正的或不考虑权之间的作用修正的, 实际上,

2003-04-15 收稿, 2003-07-11 收修改稿
*上海市教委自然科学基金(03AK14)资助项目
** E-mail: bqlu@mail.shu.edu.cn

在权学习中, 神经元网络权之间是相互影响的, 这样, 若不考虑它们的作用, 将导致网络的泛化能力下降, 学习收敛速度变慢和局部极小. 例如, 速降法, 每个变量不是按照协调修正, 而是按照自己的速降方向修正. 因已有多数学习法都是基于局部信息, 它们敏感于由均匀随机分布产生的权初值, 即使学习速度比较快也易于陷入局部极小. 另外, 已有一些不用任何先验信息的解决局部极小方法, 如遗传算法和随机探索等方法, 但这些方法求全局最小的计算时间太长. 正如上述所说, 考虑先验信息也许是解决上述四种问题的一个好方法, 因为局部先验信息可以帮助加快学习; 全局先验信息可以帮助设计网络的结构, 克服局部极小和改善网络的泛化能力. 这样, 如果知道教师函数一些信息的话, 我们可以使用这些信息来设计网络结构和改善其学习效果, 这要求教师函数的一些信息, 它告诉我们比训练数据更多的教师函数特性. 所以, 先验信息的利用将给网络更好的特性.

已有许多使用先验信息解决网络泛化能力[1~10]和网络结构与目标函数的关系[11~18]的有益结果. 这些大致地分为两类: 一类是利用先验信息构造网络结构, 如基于 Bayes 推理的 Bayes 网络[11~14, 19], 基于 IF-THEN 表达先验信息的模糊网络[15~18]. 模糊和 Bayes 先验信息被用于选择网络结构, 但这方法不能使用精确的数学方程和不等式表示的先验信息. 另一类是把先验信息作为约束加入目标函数中来改善网络的泛化能力, 如 regularization 法[9]. 但这类方法不清楚先验信息与网络结构的关系. 实际上, 第二类是第一类的一种特例, 如 regularization 法在某程度上网络的权服从 Gauss 分布的 Bayes 方法.

综上可知: 不同的先验信息可应用于解决不同的种类问题. 本文试图寻找一些其他先验信息来解决两种问题: 三层网络学习的收敛速度和局部极小. 当然, 在学习中, 泛化能力也因利用全局先验信息随着收敛速度的改善和避免局部极小而得到改善.

在本文, 我们将研究如何利用四种先验信息来解决上述问题. 第一种是教师函数的偏导数作为已知的先验信息, 第二种是教师函数的偏导数为未知, 但在特殊点的值作为先验信息. 在某些应用中, 偏导数是可以得到的且是有效的, 如 CAD/CAM 精密加工中, 工件的形状比较特殊, 在每一点的精确形状并不需要, 但在特定部分点, 其长度、角度和曲率是重要的, 其他点的有点误差是可以接受的; 此时, 工件用神经元网络表示, 角度和曲率(导数)是可以得到的, 且是网络学习的重要数据. 在这两种情况, 我们发现网络的某些权在一定的条件下可以用其他的权以精确的方程来表示, 并提出一个基于这些精确的方程的新学习方法, 此方法是部分参数从训练数据中学习, 其余的参数则是由这些精确的方程计算, 这是不同于利用先验信息的其他方法, 如 regularization 法和 Bayes 方法. 因这些方程是全局成立, 故它们从全局角度引导学习. 同时, 因使用 BP 算法而使用局部先验信息, 这样, 在权学习过程中, 因权之间相互影响比较小从而加快学习速

度. 第三种是在两点之间的教师函数凹凸性作为不等式先验信息. 第四种是教师函数的值和其值的变化为先验信息, 这给出了可用于权初始化的不等式关系. 因这些方法从全局角度引导局部学习而加快学习和保证较小的误差.

1 问题的说明

考虑如下 n 输入-单输出的三层神经元网络

$$y(x)=\sum_{k=1}^{m}\alpha_k\varphi(W_k^{\mathrm{T}}X+\theta_k), \tag{1}$$

这里α_k和 $W_k=[w_{1,k}, w_{2,k}, \cdots, w_{n,k}]^{\mathrm{T}}$ 分别是连接隐层与输出层的权和输入层与隐层的权, $X=[x_1, x_2, \cdots, x_n]^{\mathrm{T}}\in\mathbb{R}^n$ 为神经元网络的输入, $\varphi(x)=(1-e^{-x})/(1-e^{-x})$, θ_k是阈值, 训练数据集为$\{X(j), f(X(j))\}(j=1,2,\cdots,P)$, P 是训练数据总数. 对于输出层是单调函数ϕ 的任意网络, 训练数据可以转换成$\phi^{(-1)}(f(X(i)))$, 这样, 网络可简化为(1)式, 为了清楚起见, 在本文, 设网络的结构为(1)式. 神经元网络训练由 BP 法进行, 其误差函数为

$$E(W,\alpha,\theta)=\sum_{j=1}^{P}(y(X(j)-f(X(j))))^2. \tag{2}$$

在训练数据中, 有时我们有许多教师函数$f(X)$的信息. 这些信息包括教师函数$f(X)$的导数, 神经元网络输出值与教师函数 $f(X)$相等的特定点以及教师函数 $f(X)$的凹凸性. 这些先验信息在设定权的初始值时和权学习中, 约束神经元网络权.

2 先验信息以及神经元网络权之间关系

本节将研究由四类先验信息给出的神经元网络权之间关系. 第 1 类先验信息是教师函数$f(x_1, x_2, \cdots, x_n)$的某固定点导数, 这给出了描述连接输入层-隐层权和连接隐层-输出层权关系的方程. 第 2 类先验信息是神经元网络输出值与教师函数 $f(X)$值相等的特定点, 这也给出了网络权关系的方程. 第 3 类先验信息是教师函数 $f(X)$的凹凸性, 这给出了关于网络权关系的不等式. 第 4 类先验信息是由教师函数 $f(X)$值和其变化值得到的不等式, 这可用来设定权的初始值. 第 1 类方程和第 2 类方程比第 3 类不等式和第 4 类不等式在约束权的方面更严格和更有效. 对于第 1 类方程约束和第 2 类方程约束, 网络部分权从数据中修正, 其他权则由这些约束方程的计算而得到. 第 3 类不等式约束和第 4 类不等式约束可以帮助权的初始化, 当然. 这些不等式也可用于权的学习, 如 regularization 法一样.

2.1 教师函数的导数和参数之间关系

在本节, 为了不失一般性, 设教师函数$f(X)$在原点的导数为已知的. 因任意 n 维到 1 维的映射都可以用神经元网络来近似[20], 对于给定的教师函数$f(x_1, x_2, \cdots,$

x_n)和任意小正数ε，存在整数 m 使

$$\left| f(x_1, x_2, \cdots, x_n) - \sum_{k=1}^{m} \alpha_k \varphi(W_k^{\mathrm{T}} X + \theta_k) \right| < \varepsilon \tag{3}$$

成立. 设 $f(X) \in C^{\infty}$ 和 $\Delta F = f(X) - \sum_{k=1}^{m} \alpha_k \varphi(W_k^{\mathrm{T}} + \theta_k)$，并将$\Delta F(X)$在 X=0 处 Taylor 展开，得

$$\Delta F = f(0,0,\cdots,0) - \sum_{k=1}^{m} \alpha_k \varphi(\theta_k) + \sum_{i=1}^{n} \left(f_{x_i}^{(1)}(0,0,\cdots,0) - \sum_{k=1}^{m} \alpha_k w_{i,k} \varphi^{(1)}(\theta_k) \right) x_i + \cdots.$$

若让上式每项为零，则 $f(0) = \sum_{k=1}^{m} \alpha_k \varphi(\theta_k)$, $f_{x_{i1}}^{(j)}(0) = \sum_{k=1}^{m} \alpha_k w_{i1,k}^{(j)} \varphi^{(j)}(\theta_k)$,

$$f_{x_{i1}, x_{i2}^j}^{(j+1)}(0) = \sum_{k=1}^{m} \alpha_k w_{i2,k}^{j} w_{i1,k} \varphi^{(j+1)}(\theta_k), \tag{4}$$

这里，$f_{x_{i2}, x_{i1}^j}^{(j+1)}(0)$ 是关于 x_{i2} 的一阶导数和 x_{i1} 的 j 阶导数. j=1,2,⋯, $i1$=1,2,⋯, 为清楚起见，设 θ_k=1，整理方程(4)，并得

$$A_1 B_1 = C_1, \tag{5}$$

$$A_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ w_{i1,1} & w_{i1,2} & w_{i1,3} & w_{i1,3} \\ \vdots & \vdots & \vdots & \vdots \\ w_{i1,1}^{m1-1} & w_{i1,1}^{m1-1} & w_{i1,1}^{m1-1} & w_{i1,1}^{m1-1} \end{bmatrix}, \quad B_1 = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{m1} \end{bmatrix}, \quad C_1 = \begin{bmatrix} \dfrac{f(0)}{\varphi(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k \\ \dfrac{f_{x_{i1}}^{(1)}(0)}{\varphi^{(1)}(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k w_{i1,k} \\ \vdots \\ \dfrac{f_{x_{i1}}^{(m1-1)}(0)}{\varphi^{(m1-1)}(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k w_{i1,k}^{m1} \end{bmatrix};$$

或

$$A_2 B_2 = C_2, \tag{6}$$

$$A_2 = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{m1} \\ \alpha_1 w_{i2,1} & \alpha_2 w_{i2,2} & \cdots & \alpha_{m1} w_{i2,m1} \\ \vdots & \vdots & & \vdots \\ \alpha_1 w_{i2,1}^{m1-1} & \alpha_2 w_{i2,2}^{m1-1} & \cdots & \alpha_{m1} w_{i2,m1}^{m1-1} \end{bmatrix}, \quad C_2 = \begin{bmatrix} \dfrac{f_{x_{i1}}^{(1)}(0)}{\varphi^{(1)}(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k w_{i1,k} \\ \dfrac{f_{x_{i1}}^{(2)}(0)}{\varphi^{(2)}(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k w_{i1,k} w_{i2,k} \\ \vdots \\ \dfrac{f_{x_{i1}, x_{i2}^{m1}}^{(m1+1)}(0)}{\varphi^{(m1)}(1)} - \sum\limits_{k=m1+1}^{m} \alpha_k w_{i1,k} w_{i2,k}^{m1} \end{bmatrix},$$

$$B_2=[w_{i1,1},w_{i1,2},\cdots,w_{i1,m1}]^{\mathrm{T}},\ m1\leqslant m,\ i1=1,2,\cdots,n,\ i2=1,2,\cdots,n.$$

因

$$\begin{vmatrix} 1 & 1 & \cdots & 1 \\ w_{i1,1} & w_{i1,2} & \cdots & w_{i1,m1} \\ \vdots & \vdots & & \vdots \\ w_{i1,1}^{m1-1} & w_{i1,2}^{m1-1} & \cdots & w_{i1,m1}^{m1-1} \end{vmatrix}=(-1)^{\frac{m1(m1-1)}{2}}\prod_{1\leqslant k1<k2\leqslant m1}(w_{i1,k1}-w_{i1,k2}),\tag{7}$$

矩阵 A_1 的逆存在, 当且仅当存在 $m1$ 使得

$$w_{i1,k1}\neq w_{i1,k2},\ k1\neq k2,\ k1,k2=1,2,\cdots,m1.\tag{8}$$

否则可以减少 $m1$ 直到矩阵 A_1 的逆存在. 若当 $m1=m$ 时, (8)式成立, 则

$$B_1=A_1^{-1}C_1.\tag{9}$$

由上式可以看出: 在 B_1 中的α_k (k=1,2,$\cdots$,m)可用 $w_{i1,k1}$, $k1$=1,2,$\cdots$,m 表示, 换句话说, 在(8)式成立条件下, 输出层-隐层的权可用输入层-隐层的权表示, 这对网络权的学习是非常重要的. 但(8)式不是必要条件, 如果(8)式不成立, 仍可用 $w_{i1,k1}$ 表示α_k(k=1,2,$\cdots$,m).

定理 1 对于给定的 $m1$, 若 $m1=m$, 则 A_1 的逆存在, 否则隐层节点可删除直到 $m1=m$(见附录 A).

定理 2 若某神经元网络和教师函数的任意阶导数都满足(4)式, 则除了某些节点可以删除外, 在输出层和输入层的神经元网络权是惟一的(见附录 A).

定理 3 若节点数为 $N1m$ 的神经元网络 $N1$, 节点数为 $N2m$ 的神经元网络 $N2$($N1m>N2m$)和教师函数的任意阶导数都满足(4)式, 则除了 $N1$ 和 $N2$ 的某些节点可以删除外, $N2$ 的输入层权与 $N1$ 的输入层权相等(证明与定理 2 相同).

我们已知道输入层权和输出层权之间的关系, 下面我们将研究输入层权之间的关系. 若对于满足(8)和(10)式的 $i2$ 输入,

$$\alpha_k\neq 0,\ k=1,2,\cdots,m,\tag{10}$$

则由(6)式得

$$B_2=A_2^{-1}C_2.\tag{11}$$

这表明: 在(8)和(10)式成立的条件下, 当 $i1\neq i2$ 时, $w_{i1,k1}$ 可由 $w_{i2,k2}$ 表示, 实际上, $\alpha_k\neq 0$ 否则第 K 个隐层节点可被删去.

2.2 主要学习点与参数之间的关系

我们已经讨论过在教师函数导数为已知的情况下网络权之间的关系. 现在, 讨论如何利用教师函数的初始值, 终值和最大值与最小值等主要学习点来确定教师函数的大致形状, 如通过不同的两点可惟一确定一条直线, 通过任意不同的三点可惟一确定一条二阶曲线等等. 设 L 个 X 主要学习点的神经网络的输出值精

确等于相应教师函数的值, 其主要学习点为 $X(i)=(x_1(i),x_2(i),\cdots,x_n(i))$, $i=1,2,\cdots,L$, 则

$$f(X(i))=\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}} X(i)+\theta_k),\ i=1,2,\cdots,L. \tag{12}$$

这是另一种先验信息, 它隐含网络权的关系.

2.3 教师函数的凸凹性与网络参数之间的关系

我们知道: 在某区间内, 教师函数的凸凹性可作为先验信息. 其凸凹性定义如下:

定义 1 若 D 是点 $X(i)=(x_1(i),x_2(i),\cdots,x_n(i))$, $i=1,2,\cdots$ 的凸集, 在 D 上, 如果对于任不同点 $X(i)$和 $X(j)$, $i\neq j$, 下式都成立, 实函数 $H(X)$被定义是凸的[21].

$$H[(1-\theta)X(i)+\theta X(j)]\leqslant(1-\theta)H(X(i))+\theta H(X(j)),\ 0\leqslant\theta\leqslant1,\ i\neq j. \tag{13}$$

若对于 $0<\theta<1$, 上式取不等号也成立, 则被称为严格凸的. 由这定义, 若教师函数在点 $X(i)$和 $X(j)$之间是凸的话, 则先验信息可以表示为

$$\begin{aligned}&\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}}[(1-\theta)X(i)+\theta X(j)]+\theta_k)\\&\leqslant(1-\theta)\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}} X(i)+\theta_k)+\theta\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}} X(j)+\theta_k),\ 0\leqslant\theta\leqslant1,\ i\neq j,\end{aligned} \tag{14}$$

上式表示网络权的关系. 同理, 若教师函数在点 $X(i)$和点 $X(j)$之间是凹的话, 则先验信息只把上式的≤变成≥即可.

2.4 神经网络权的上下界

本节将讨论神经网络权的上下界, 它将给出神经网络完全逼近教师函数的必要条件. 若神经网络完全逼近教师函数 $f(X)$, 则在点 $X(i)$和 $X(j)$点, 其变化一定等于 $f(X)$的变化, 即

$$\begin{aligned}f(X(i))-f(X(j))&=\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}} X(i)+\theta_k)-\sum_k^m \alpha_k \varphi(W_k^{\mathrm{T}} X(j)+\theta_k)\\&=\sum_k^m \alpha_k \varphi^{(1)}(W_k^{\mathrm{T}} X(\zeta)+\theta_k)(W_k^{\mathrm{T}}(X(i)-X(j))).\end{aligned} \tag{15}$$

这里, $X(\zeta)\in[\min(X(i),X(j)),\max(X(i),X(j))]$. 若对上式两边取绝对值, 则

$$|f(X(i))-f(X(j))|\leqslant\sum_k^m|\alpha_k|\,\|\varphi^{(1)}(W_k^{\mathrm{T}} X(\zeta)+\theta_i)\|\,\|(W_k^{\mathrm{T}}(X(i)-X(j)))\|. \tag{16}$$

因 $\varphi(x)=(1-e^{-x})/(1+e^{-x})$, 则 $\varphi^{(1)}(x)=0.5(1-\varphi^2(x))\leqslant0.5$, 这样,

$$|f(X(i))-f(X(j))|\leqslant 0.5\sum_{k}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\|X(i)-X(j)\|,$$
$$\sum_{k}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\geqslant \max\frac{2|f(X(i))-f(X(j))|}{\|X(i)-X(j)\|}\geqslant\frac{2|f(X(i))-f(X(j))|}{\|X(i)-X(j)\|}, \tag{17}$$

这里，$\|W_k^{\mathrm{T}}\|=\sqrt{\sum_{i1}^{n}w_{i1,k}^2}$，$n$ 为神经元网络的输入数. 令

$$g_1(W,\alpha)=\max\frac{2|f(X(i))-f(X(j))|}{\|X(i)-X(j)\|}-\sum_{k}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\leqslant 0. \tag{18}$$

这也是一种用不等式表示的先验信息. 另外，由(16)式，对于任一 L 有

$$\begin{aligned}
&|f(X(i))-f(X(j))|\\
&=\left|\sum_{k\neq L}^{m}\alpha_k\varphi^{(1)}(W_k^{\mathrm{T}}X(\zeta)+\theta_k)W_k^{\mathrm{T}}(X(i)-X(j))+\alpha_L\varphi^{(1)}(W_L^{\mathrm{T}}X(\zeta)+\theta_L)W_L^{\mathrm{T}}(X(i)-X(j))\right|\\
&\geqslant|\alpha_L\varphi^{(1)}(W_L^{\mathrm{T}}X(\zeta)+\theta_L)W_L^{\mathrm{T}}(X(i)-X(j))|-\\
&\left|\sum_{k\neq L}^{m}\alpha_k\varphi^{(1)}(W_k^{\mathrm{T}}X(\zeta)+\theta_k)W_k^{\mathrm{T}}(X(i)-X(j))\right|\\
&\geqslant\left|\alpha_L\varphi^{(1)}(W_L^{\mathrm{T}}X(\zeta)+\theta_L)W_L^{\mathrm{T}}(X(i)-X(j))\right|-0.5\sum_{k\neq L}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\|(X(i)-X(j))\|,
\end{aligned}$$

即

$$\begin{aligned}
&|\alpha_L\varphi^{(1)}(W_L^{\mathrm{T}}X(\zeta)+\theta_L)W_L^{\mathrm{T}}(X(i)-X(j))|\\
&\leqslant|f(X(i))-f(X(j))|+0.5\sum_{k\neq L}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\|(X(i)-X(j))\|,
\end{aligned}$$

这样，

$$|\alpha_L M_c|\|W_L^{\mathrm{T}}(X(i)-X(j))\|\leqslant|f(X(i))-f(X(j))|+0.5\sum_{k\neq L}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\|(X(i)-X(j))\|,$$

这里 $M_c=\min[\varphi^{(1)}(W_k^{\mathrm{T}}X(i)+\theta_k),\ \varphi^{(1)}(W_k^{\mathrm{T}}X(j)+\theta_k)]$. 令

$$\begin{aligned}
g_2(W,\alpha)=&|\alpha_L M_c|\|W_L^{\mathrm{T}}(X(i)-X(j))\|-|f(X(i))-f(X(j))|\\
&-0.5\sum_{k\neq L}^{m}|\alpha_k|\|W_k^{\mathrm{T}}\|\|(X(i)-X(j))\|,
\end{aligned}$$

则

$$g_2(W,\alpha)\leqslant 0, \tag{19}$$

$|W_L^{\mathrm{T}}|$，α_L 越小和其他权越大，(19)式越易满足. 故联合(18)式和(19)式，便得其可能解域. 当然，由(12)式，有

$$|f(X(i))| \leqslant \sum_k^m \left|\alpha_k \varphi(W_k^{\mathrm{T}} X(i)+\theta_k)\right| \leqslant \sum_k^m |\alpha_k|, \sum_k^m |\alpha_k|$$
$$\geqslant \max_i |f(X(i))|, \ i=1,2,\cdots,P,$$

$$g_3(W,\alpha)=\max_i |f(X(i))|-\sum_k^m |\alpha_k| \leqslant 0, \tag{20}$$

令

$$g_4(W,\alpha)=|\alpha_L|-|f(X(i))|-\sum_{k\neq L}^m |\alpha_k| \leqslant 0. \tag{21}$$

与(19)式相似, 有不等式(18)~(21), 给出了神经元网络的权上下界, 它可用于设定权初始值和权的学习. 在设定权初始值时, 选择使(19)和(21)式成立比较小的权. 但当权的个数比较多时, 即使较小的权, 也能使(18)和(20)式成立. 另外, 为了使(19)和(21)式成立, 权的最大绝对值也不要大.

2.5　利用教师函数偏导数的学习

我们知道: 若对于(9)和(11)式, $m1=m$, 则(9)和(11)式的右边与神经元网络的参数无关. 因此, 权的每次迭代可以利用这些式子. 在(8)和(10)式成立条件下, 若对(9)和(11)式的两边求偏导, 则

$$\frac{\partial \alpha_k}{\partial w_{i1,j1}}=T1^{(k)}_{w_{i1,j1}}, \ \frac{\partial w_{i1,j1}}{\partial w_{i2,j2}}=T2^{(i1,j1)}_{w_{i2,j2}}, \ \frac{\partial \alpha_k}{\partial w_{i2,j2}}=\sum_{i1,j1} T1^{(k)}_{w_{i1,j1}} T2^{(i1,j1)}_{w_{i2,j2}}, \ k=1,2,\cdots,m, \tag{22}$$

这里, $T1^{(k)}_{w_{i1,j1}}$, $T2^{(k)}_{w_{i2,j2}}$ 分别是 $A_1^{-1}C_1$ 的第 k 个元素对 $w_{i1,\,j1}$ 的偏导数和 $A_2^{-1}C_2$ 的 $j1$ 个元素对 $w_{i2,\,j2}$ 的偏导. 误差函数 E, 在 W 和 α 处, Taylor 展开:

$$\Delta E(W,\alpha)=\frac{\partial E}{\partial W}\Delta W+\frac{\partial E}{\partial \alpha}\Delta\alpha+\cdots=\sum_i^n\sum_k^m \frac{\partial E}{\partial w_{i,k}}\Delta w_{i,k}+\sum_k \frac{\partial E}{\partial \alpha_k}\Delta\alpha_k+\cdots. \tag{23}$$

若对输入节点 $i1$ 和 $i2$, (8)式成立, 则

$$\Delta E(W,\alpha)=\sum_{i\neq i1,i2}^n\sum_k^m \frac{\partial E}{\partial w_{i,k}}\Delta w_{i,k}+\sum_{j1}\frac{\partial E}{\partial w_{i1,j1}}\left(\sum_{j2} T2^{(i1,j1)}\Delta w_{i2,j2}\right)+$$
$$\sum_k \frac{\partial E}{\partial \alpha_k}\left(\sum T1^{(k)}_{w_{i1,j1}} T2^{(i1,j1)}_{w_{(i2,j2)}}\Delta w_{i2,j2}\right)+\sum_{k1}\frac{\partial E}{\partial w_{i2,k1}}\Delta w_{i2,k1}+\cdots=$$
$$\sum_{j2}\left(\sum_{j1}\frac{\partial E}{\partial w_{i1,j1}}T2^{(i1,j1)}_{w_{(i2,j2)}}+\sum_k\frac{\partial E}{\partial \alpha_k}\sum_{j1}T1^{(k)}_{w_{(i1,j1)}}T2^{(i1,j1)}_{w_{(i2,j2)}}+\frac{\partial E}{\partial w_{(i2,j2)}}\right)\Delta w_{i2,j2}$$
$$+\sum_{i\neq i1,i2}\sum_k\frac{\partial E}{\partial w_{i,k}}\Delta w_{i,k}+\cdots.$$

按照速降法, 有如下学习算法:

$$\Delta w_{i2,j2}=-\eta\left(\sum_{j1}\frac{\partial E}{\partial w_{i1,j1}}T2^{(i1,j1)}_{w_{(i2,j2)}}+\sum_{k}\frac{\partial E}{\partial \alpha_k}\sum_{j1}T1^{(k)}_{w_{(i1,j1)}}T2^{(i1,j1)}_{w_{(i2,j2)}}+\frac{\partial E}{\partial w_{(i2,j2)}}\right),$$
$$\Delta w_{i,j}=-\eta\frac{\partial E}{\partial w_{i,k}},\ i\neq i1,i2. \tag{24}$$

这里, η是学习系数. 权可由(9)、(11)和(24)式修正. 其中 $w_{i2,k}$, $w_{i,k}(i\neq i1,\ i2)$由(24)式修正, $w_{i1,j1}$ 由(11)式计算得到, α由(9)式计算得到. 其学习过程将在 2.9 小节介绍.

2.6 利用教师函数主要学习点的学习

现在讨论如何使用 L 个主要学习点(不失一般性, 设 $L\leqslant m$)来修正权. 首先, 由(12)式给出关于网络权的一些方程, 然后, 联合这些方程和 BP 算法来修正权. 由(12)式有

$$f(x_1(1),x_2(1),\cdots,x_n(1))=\sum_{k=1}\alpha_k\varphi_k(1),$$
$$\cdots \tag{25}$$
$$f(x_1(L),x_2(L),\cdots,x_n(L))=\sum_{k=1}\alpha_k\varphi_k(L),\ \varphi_k(i)=\varphi(W_k^{\mathrm{T}}X(i)+1),\ \theta_k=1.$$

为了获得微小变化的线性约束, 对(25)式两边的权求导, 得

$$\sum_{k=1}\Delta\alpha_k\varphi_k(1)+\sum_{k=1}\sum_{j}\alpha_k\varphi_k^{(1)}(1)x_j(1)\Delta w_{j,k}=0,$$
$$\cdots \tag{26}$$
$$\sum_{k=1}\Delta\alpha_k\varphi_k(1)+\sum_{k=1}\sum_{j}\alpha_k\varphi_k^{(1)}(1)x_j(L)\Delta w_{j,k}=0,$$

这里, 为了保证(25)式成立, $\Delta w_{j,k}$和$\Delta\alpha_k$是非常小的, 同时, 定义误差函数为

$$E(W,\alpha)=\sum_{k\notin Z}[y_k-f(X(k))]^2+\sum_{k\in Z}B_k[y_k-f(X(k))]^2,\ Z=1,2,\cdots,L, \tag{27}$$

这里, B_k是比较大的惩罚系数. 在学习中, 选择足够大的 B_k使(26)式成立.

联合这些线性约束和 BP 算法, 在$\Delta w_{j,k}$和$\Delta\alpha_k$之间, 选取使系数矩阵是可逆的 L 个参数, 这样, 这些参数可由(26)式来修正权. 若其系数矩阵不是可逆的, 则可以减少 L 直到系数矩阵可逆为止. 为了方便, 假设 L 个参数被选为$\Delta\alpha_k$, $k=1,2,\cdots,L$, 则

$$AB=-CD,\ B=FD,\ F=-A^{-1}C,\ C=[H\mid E],$$
$$D=[\Delta\alpha_{L+1},\Delta\alpha_{L+2},\cdots,\Delta\alpha_m,\Delta w_{1,1},\cdots,w_{n,m}]^{\mathrm{T}},$$
$$A=\begin{bmatrix}\varphi_1(1) & \varphi_2(1) & \cdots & \varphi_L(1)\\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_L(2)\\ \vdots & \vdots & & \vdots\\ \varphi_1(L) & \varphi_2(L) & \cdots & \varphi_L(L)\end{bmatrix},$$

$$E=\begin{bmatrix} \varphi_1^{(1)}(1)x_1(1) & \varphi_2^{(1)}(1)x_2(1) & \cdots & \varphi_L^{(1)}(1)x_n(1) \\ \varphi_1^{(1)}(2)x_1(2) & \varphi_2^{(1)}(2)x_2(2) & \cdots & \varphi_L^{(1)}(2)x_n(2) \\ \vdots & \vdots & & \vdots \\ \varphi_1^{(1)}(L)x_1(L) & \varphi_2^{(1)}(L)x_n(L) & \cdots & \varphi_L^{(1)}(L)x_n(L) \end{bmatrix},$$

$$H=\begin{bmatrix} \varphi_{L+1}(1) & \varphi_{L+2}(1) & \cdots & \varphi_m(1) \\ \varphi_{L+1}(2) & \varphi_{L+2}(2) & \cdots & \varphi_m(2) \\ \vdots & \vdots & & \vdots \\ \varphi_{L+1}(L) & \varphi_{L+2}(L) & \cdots & \varphi_m(L) \end{bmatrix},\ B=[\Delta\alpha_1,\Delta\alpha_2,\cdots,\Delta\alpha_L]^{\mathrm{T}}, \tag{28}$$

从上式可得

$$\Delta E(W,\alpha)=\sum_{i1,i2}^{n,m}\left(\frac{\partial E}{\partial w_{i1,i2}}+\sum_k^L\frac{\partial E}{\partial\alpha_k}F_{k,(i1^*m+i2+m-l)}\right)\Delta w_{i1,i2}$$
$$+\sum_{j=L+1}^{m}\left(\frac{\partial E}{\partial\alpha_j}+\sum_k^L\frac{\partial E}{\partial\alpha_k}F_{k,(j-L)}\right)\Delta\alpha_j+\cdots,$$

这里, $F_{k,(j-L)}$是矩阵 F 的$(k, j-k)$元素. 按照速降法, 得如下算法:

$$\Delta w_{i1,i2}=-\eta\left(\frac{\partial E}{\partial w_{i1,i2}}+\sum_k^L\frac{\partial E}{\partial\alpha_k}F_{k,i1m+i2+m-l}\right),$$
$$\Delta\alpha_j=-\eta\left(\frac{\partial E}{\partial\alpha_j}+\sum_k^L\frac{\partial E}{\partial\alpha_k}F_{k,i-L}\right), \tag{29}$$

网络的权除了$\Delta\alpha_k$, k=1,2,$\cdots$,L 由(28)式修正外, 其他权由(29)式修正.

2.7　利用教师函数凹凸性学习

本节讨论利用(14)式的学习方法. 设教师函数在点 $X(1)$和 $X(2)$之间是凸的, 在点 $X(3)$和 $X(4)$之间是凹的, 则惩罚函数为

$$E_1(W,\alpha,\zeta)=E(W,\alpha)+\sum_{i=1}^{2}\zeta_i s_i^+(W,\alpha)^2,\ s_1^+(W,\alpha)=\max\{0,s_1(W,\alpha)\},$$
$$s_1(W,\alpha)=\sum_k\alpha_k\varphi(W_k^{\mathrm{T}}[(1-\theta)X(1)+\theta X(2)]+\theta_k)-(1-\theta)\sum\alpha_k\varphi(W_k^{\mathrm{T}}X(1)+\theta_k)$$
$$-\theta\sum\alpha_k\varphi(W_k^{\mathrm{T}}X(2)+\theta_k)+b_1, \tag{30}$$
$$s_2^+(W,\alpha)=\min\{0,s_2(W,\alpha)\},s_2(W,\alpha)=-\sum_k\alpha_k\varphi(W_k^{\mathrm{T}}[(1-\theta)X(3)+\theta X(4)]+\theta_k)$$
$$+(1-\theta)\sum\alpha_k\varphi(W_k^{\mathrm{T}}X(3)+\theta_k)+\theta\sum\alpha_k\varphi(W_k^{\mathrm{T}}X(4)+\theta_k)+b_2,\ 0<\theta<1,$$

这里, b_i(i=1,2)是据教师函数凸凹度而得到的正实数, 被用来调节教师函数凸凹程度. θ 由[0, 1]均匀分布得到的. 按照速降法, 有如下算法:

$$\Delta W=-\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial W},\ \Delta\alpha=-\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial\alpha},\ \Delta\zeta=\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial\zeta}. \quad (31)$$

正如前面所说，ζ_i $(i=1,2)$可用比较大的正数置换. 现在讨论上式平衡点问题. 若 $\frac{\partial s_i(W,\alpha)}{\partial\alpha}=0$, 则

$$\varphi(W_k^{\mathrm{T}}((1-\theta)X(1)+\theta X(2)+\theta_k))=(1-\theta)\varphi(W_k^{\mathrm{T}}X(1)+\theta_k)+\theta\varphi(W_k^{\mathrm{T}}X(2)+\theta_k),$$
$$\varphi(W_k^{\mathrm{T}}((1-\theta)X(3)+\theta X(4)+\theta_k))=(1-\theta)\varphi(W_k^{\mathrm{T}}X(3)+\theta_k)+\theta\varphi(W_k^{\mathrm{T}}X(4)+\theta_k),$$

因为θ 由[0, 1]均匀分布随机得到, 对于[0, 1]之间的任意θ, 这些方程都成立, 这表明: φ_k在[$X(1)$, $X(2)$]和[$X(3)$, $X(4)$]之间对于所有的k是线性平面, 即 $s_1=b_1$, $s_2=b_2$. 另外, 由优化方法, 可得 $s_i^+=0$, $i=1,2$, $\frac{\partial E_1(W,\alpha)}{\partial W}=0$, $\frac{\partial E_1(W,\alpha)}{\partial\alpha}=0$, 但$\varphi_k$ 不是线性平面, 则 $s_1=-b_1$, $s_2=-b_2$, 为此把 s_i^+ 变成 $s_i^+/\|\alpha\|^2$, 这样,

$$\frac{\partial s_1(W,\alpha)}{\partial\alpha_k}\frac{1}{\|\alpha\|^2}-\frac{2s_1(W,\alpha)\alpha_k}{\|\alpha\|^4}=0,\ \frac{\partial s_1(W,\alpha)}{\partial\alpha_k}\|\alpha\|^2=2s_1(W,\alpha)\alpha_k,$$
$$s_1(W,\alpha)-b_1=2s_1(W,\alpha),\ s_1(W,\alpha)=-b_1,\ k=1,2,\cdots,m. \quad (32)$$

这正满足 $s_1^+=\max\{0,s_1\}$, 即先验知识成立. 当然, 其他先验知识与此类似. 这表明权收敛于误差函数的局部最小点.

2.8 利用网络权的上下界的学习

本节讨论使用(18)~(21)式的学习法. 其惩罚函数为

$$E_1(W,\alpha,\zeta)=E(W,\alpha)+\zeta_i g_i^+(W,\alpha)^2,\ i=1,2,3,4,\ g_i^+(W,\alpha)=\max\{0,g_i(W,\alpha)\}.$$

按照速降法, 有

$$\Delta W=-\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial W},\ \Delta\alpha=-\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial\alpha},\ \Delta\zeta_i=-\eta\frac{\partial E_1(W,\alpha,\zeta)}{\partial\zeta}, \quad (33)$$

这里, ζ_i $(i=1,2,3,4)$可以用比较大的正数置换. 当$\alpha_k>0$, $k=1,2,\cdots,m$ 时, $\frac{\partial g_1(W,\alpha)}{\partial\alpha_k}=-\|W_k\|<0$, $\frac{\partial g_3(W,\alpha)}{\partial\alpha_k}=-1<0$, 则$\alpha_k$将远离 0; 当$\alpha_k<0$, $k=1,2,\cdots,m$ 时, $\frac{\partial g_1(W,\alpha)}{\partial\alpha_k}=\|W_k\|>0$, $\frac{\partial g_3(W,\alpha)}{\partial\alpha_k}=1>0$, 则$\alpha_k$也远离 0 直到先验知识满足为止, 即 $g_1^+=0$, $g_3^+=0$. 其他权与此类似, 因此, 此情形不同于 regularization 法[1, 2], 但对于 g_2, g_4, 当$\alpha_L>0$ 时, $\frac{\partial g_2(W,\alpha)}{\partial\alpha_L}=M_c\|W_L\|\|X(i)-X(j)\|>0$, $\frac{\partial g_4(W,\alpha)}{\partial\alpha_L}>0$, 则$\alpha_k$ 趋近于 0, 此情形类似于 regularization 法, 但这些约束条件是依据完

全逼近的网络得到的, 它使我们知道在权学习中被使用的理由.

2.9 学习过程

上述已经得到了使用先验知识的学习算法, 当先验知识以方程给出时, 网络的部分参数由使用局部信息的速降法来修正, 其他部分参数则由使用全局信息的精确计算来修正. 为了加快学习, 可采用具有惯性项 BP 算法, 在权的初始化时要满足(18)和(19)式, 因为使用不等式先验知识的学习方法基本上与惩罚函数优化方法相同, 这里, 只给出使用由方程表示的先验知识的参数修正过程.

1) 在网络权中选择 $m1$ 或 L 个参数使它们尽可能地满足(8)或(28)式, 否则减少 $m1$ 或 L, 再一次选择这些参数,其初始值由(5)和(6)式或(25)、(18)和(20)式得到, 其他权初始值由使(18)和(20)式成立的随机数产生.

2) 除了 $m1$ 或 L 个参数外, 部分参数由(24)或(29)式修正, 其他参数由(9)和(10)或(28)式修正.

3) 再一次在网络权中选择 $m1$ 或 L 个参数使它们尽可能地满足(8)或(28)式, 然后, 转到步骤 2)直到误差函数满足要求为止.

3 仿真

本节从两个方面验证上述理论.一个方面是即使从局部最小的坏状态出发, 使用先验知识也能跳出局部最小, 例子 1 和 2 的结果支持此结论; 另一方面是先验知识对加快学习速度也是有效的. 例子 3 的结果支持此结论.

3.1 例子 1—— 具有偏导数信息的函数近似

在这个例子中, 假设教师函数 $f(x)$在原点的一阶和二阶偏导数是先验知识, 函数 $f(x)$[21]为

$$f(x)=0.2+0.8(x+0.7\sin(2\pi x)). \tag{34}$$

在 $0<x<1$ 之间每隔 0.05 取一点, 共取 19 点训练数据用来仿真, 每隔 0.01 取一点, 共取 99 点用来验证网络插值能力. 在学习中使参数强烈相互影响, 网络的结构为 1-60-1, 输入层权和输出层的权初值分别满足(18)和(20)式的均匀分布(–3, 0)和(–0.5, 0.5)随机产生. 仿真在如下 2 种情况下迭代 2000 次:

情况 1: $f(0)$和 $f_x^{(1)}(0)$ 为先验知识,学习系数和惯性系数分别为 0.02 和 0.2.

情况 2: $f(0)$, $f_x^{(1)}(0)$ 和 $f_x^{(2)}(0)$ 为先验知识. 学习系数和惯性系数分别为 0.03 和 0.02.

学习结果与学习系数 0.015 和惯性系数 0.2 的 EBP 算法迭代 2000 次和 5000 次比较,其误差曲线如图 1 所示, 插值输出如图 2 所示. EBP 法 5000 次后误差为 0.02, 迭代 2000 次和 5000 次后的插值误差分别为 2.469 和 0.093. 本文方法对于

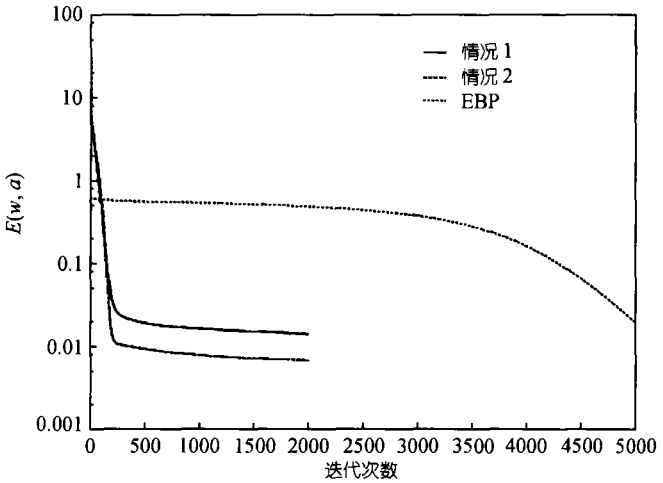

图1 例1的误差曲线

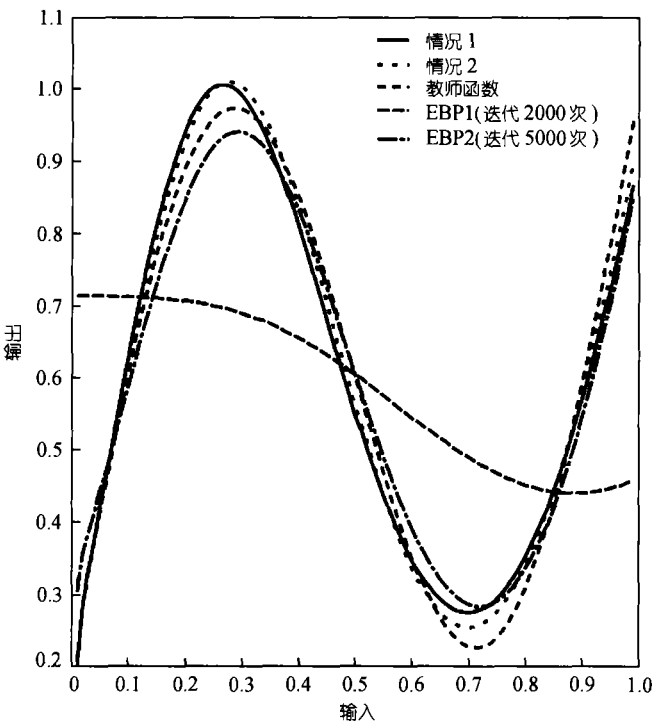

图2 例1的插值网络输出

情况1只用430次就达到，对于情况2只用184次就达到，迭代2000次后，情况1和2的学习误差和插值误差分别为0.014，0.007和0.079，0.037. 在接近原点，EBP法的误差比提案方法大，情况1的比情况2的大. 若使用更高阶偏导数的话，

其效果会更好, 表明这种方法是非常有效的.

3.2　例子 2—— 利用主要学习点的函数近似

以(34)式的主要学习点为先验知识, 在0<x<2之间每隔0.02取一点, 共取100点用来仿真. 网络的结构为 1-60-1, 误差函数由(27)式定义的, 其中 B_k=500, k=1、15, 36, 64, 86 为主要学习, 即 L=5. 除了在 60 个α_i中随机选择使(28)式的矩阵 A 可逆的α_i, i=i_1, i_2, i_3, i_4, i_5, 并由(28)式计算外, 输入层权和输出层权的初始值分别满足(18)和(20)式的均匀分布(–3, 0)和(–0.5, 0.5)随机产生. 学习系数为 0.0000000001. 学习结果与学习系数 0.001 和惯性系数 0.02 的 EBP 算法比较, 学习后的误差曲线和输出曲线分别如图 3 和 4 所示. 误差曲线由(2)式计算. 这些结果表明提案的方法学习速度是比较快的, 再一次表明提案的方法是有效的.

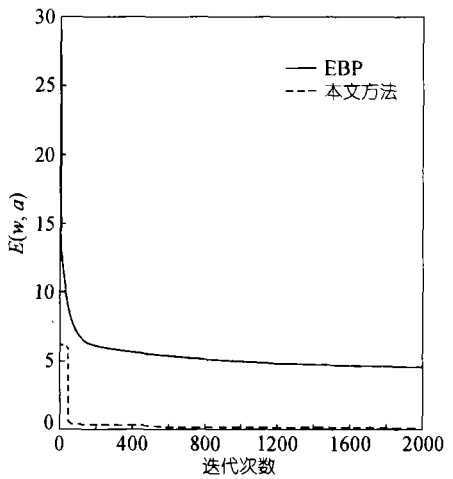

图 3　例 2 的学习曲线

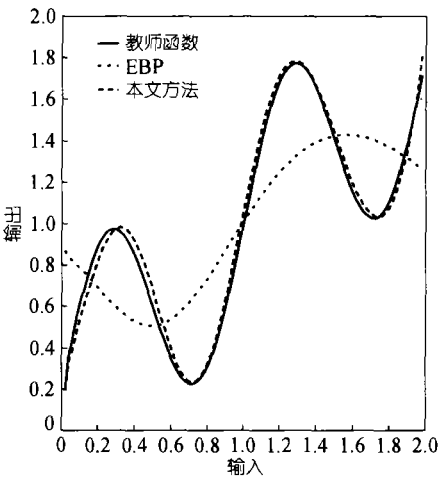

图 4　例 2 的网络输出和学习算法

3.3　例子 3——利用教师函数凹凸性的函数近似

在本仿真中, 教师函数(34)的凹凸性作为先验知识. 网络的结构为 1-60-1, 输入层权和输出层权的初始值分别满足(18)和(20)式的均匀分布(–3, 0)和(–0.5, 0.5)随机产生. 在 0<x<1 之间每隔 0.05 取一点, 共取 19 点用来仿真, 每隔 0.01 取一点, 共取 99 点用来验证网络插值能力. 由图 2 知道: 教师函数在(0.1, 0.4)即(X(2), X(8))是凸的, (0.5,0.9)即(X(10), X(18))是凹的. 误差函数如(30)式所示, 0<b_1, b_2<0.3, θ在每次迭代由均匀分布(0.4, 0.7)随机产生. 学习系数和惯性系数分别为 0.0015 和 0.2. 以 b_1=b_2=0.125, ζ_1=ζ_2=200 进行 10000 次仿真. 学习结果与学习系数 0.0015 和惯性系数 0.2 的 EBP 算法比较, 学习后的误差曲线(由(2)式计算)如图 5 所示, 表明使用凹凸性的方法是以较小误差快速收敛的, 另外, 当积分系数ε充分小, ζ_1, ζ_2尽可能取大一些. 插值网络输出如图 6 所示. 我们也使用了不同的

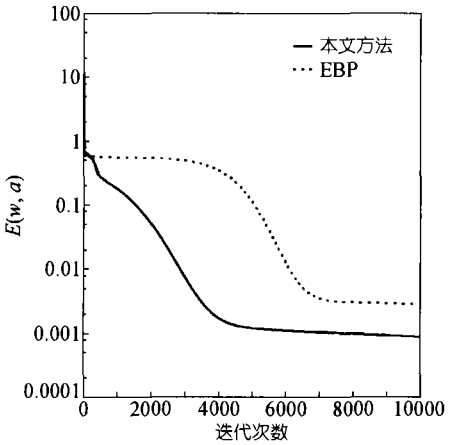

图 5 例 3 的误差曲线图

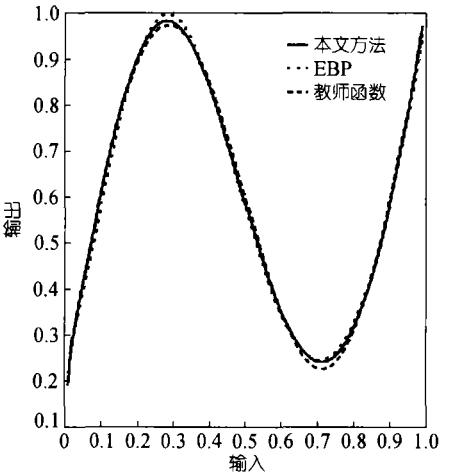

图 6 例 3 的插值网络输出

初值得到相同的结果. 这表明了本文方法的有效性与初始值无关.

4 结论

本文提出了四类描述网络权之间关系的先验知识和相应学习方法, 该方法把教师函数的数学结构或形状作为先验知识. 当先验知识由方程表示时, 在网络权学习时, 该方法使权之间的相互影响尽可能小. 当先验知识用不等式表示时, 该方法可调整网络的形状. 所有的仿真表现出良好的性能, 特别是学习速度比 EBP 法快.

参 考 文 献

1 Williams P M. Bayesian regularization and pruning using Laplace prior. Neural Computation, 1995, 7: 117~143

2 Hasegawa A. Generalization of shift invariant neural networks: Image processing of corneal endothelium. Neural Networks, 1996, 9(2): 345~356

3 Igelnik B. The ensemble approach to neural networks learing and generalization. IEEE Trans on Neural Networks, 1999, 10(1): 19~29

4 Ridella S, Rovetta S, Zunino R. Representation and generalition properties of class entropy networks. IEEE Trans on Neural Networks, 1999, 10(1): 31~47

5 Jack S N. Weight smoothing to improve networks generalization. IEEE Trans on Neural Networks, 1994, 5(5): 753~762

6 wang Lipo. Noise injection into inputs in sparsely connected hopfieled and winner-take-all networks networks. IEEE Trans on System Man and Cybernetics, 1997, 27(5): 868~870

7 Juha K, Jyrk J. Generalization of principal component analysis optimization problem and neural networks. Neural Networks, 1995, 8(4): 549~562

8 Borghese N A, Arbib M A. Generalization of temporal sequences using local dynamic programming.

Neural Networks, 1995, 8(1): 39~54
9 Goutte C. Regularization with a pruning prior. Neural Networks, 1997, 10(6): 1053~1059
10 Reed R. Pruning algorithms —— a survey. Neural Networks, 1993, 4(5): 740~747
11 Daniel T. Solving inverse problem by Bayesian neural network iterative inversion with ground truth incorporation. IEEE Trans on Signal Processing, 1997, 45(11): 553~567
12 Mostaghimi M. Bayesian estimation of a decision using information theory. IEEE Trans on System, Man and Cybernetics, 1997, 27(4): 506~517
13 MacKay D J C. Bayesian interpolation. Neural Computation, 1992, 4(3): 415~447
14 MacKay D J C. A practical Bayesian framework for backpropagation. Neural Computation, 1992, 4(3): 448~478
15 Zhang Y Q. Compensatory neurofuzzy systems with fast learning algorithms. IEEE Trans on Neural Networks, 1998, 9(1): 83~105
16 Carpenter G A, Grossberg S. Fuzzy Artmap: A neural network architecture for incremantal supervised learning of analog multidimensional maps. IEEE Trans on Neural Networks, 1992, 3: 698~713
17 Farag W A, Quintana V H, Torres G L. A genetic based nerual-fuzzy approach for modeling and control of dynamical systems. IEEE Trans on Neural Networks, 1998, 9(5): 756~767
18 Karayiannis N B, Pai P I. Fuzzy algorithms for learning vector quantization. IEEE Trans on Neural Networks, 1996, 7(5): 1196~1211
19 Chen Zhe. On different facets of refularization theory. Neural Computation, 2002, 14(12): 2791~2847
20 Chen T P. Approximation capability to functions of several variables, nonlinear functional and operators by radial basis function neural networks. IEEE Trans on Neural Networks, 1995, 6(4): 904~917
21 Saaty T L, Bram J. Nonlinear Mathematics. New York: McGraw-Hil, 1964
22 Verma B. Fast training of multilayer perceptrons. IEEE Trans on Neural Networks, 1997, 8(6): 1314~1320

附录 A

定理 1 的证明.

若对于给定 $i1$ 输入, (8)式成立, 则 α_k 可以用 $w_{i1,k1}$ 表示. 对于任意的 $i1$, 设 $m1=m$, 若(8)式不成立, 则至少存在两个隐层节点 $k1$ 和 $k2$ 使对于 $i1=1, 2, \cdots, n$, $w_{i1,k1}=w_{i1,k2}$, $k1\neq k2$. 为了简单起见, 假设 $w_{1,k1}=w_{1,k2}$ 和 $w_{2,k3}=w_{2,k4}$ 分别只对 $k1<k2$ 和 $k3<k4$ 成立, 则由(5)和(6)式, 有 $A_3B_3=C_3$, $A_4B_4=C_4$,

$$C_3=\left[\frac{f(0)}{\varphi(1)},\frac{f_{x_1}^{(1)}(0)}{\varphi^{(1)}(1)},\cdots,\frac{f_{x_1}^{(m-1)}(0)}{\varphi^{(m-1)}(1)}\right]^{\mathrm{T}},\ C_4=\left[\frac{f(0)}{\varphi(1)},\frac{f_{x_2}^{(1)}(0)}{\varphi^{(1)}(1)},\cdots,\frac{f_{x_2}^{(m-1)}(0)}{\varphi^{(m-1)}(1)}\right]^{\mathrm{T}};\quad \text{(A1)}$$

$$\begin{aligned}B_3&=[\alpha_1,\cdots,\alpha_{k1-1},\alpha_{k1+1},\cdots,\alpha_{k1}+\alpha_{k2},\cdots,\alpha_m]^{\mathrm{T}},\\ B_4&=[\alpha_1,\cdots,\alpha_{k3-1},\alpha_{k3+1},\cdots,\alpha_{k3}+\alpha_{k4},\cdots,\alpha_m]^{\mathrm{T}};\end{aligned}\quad \text{(A2)}$$

$$A_3=\begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ w_{1,1} & \cdots & w_{1,k1-1} & w_{1,k1+1} & \cdots & w_{1,m} \\ \vdots & & \vdots & \vdots & & \vdots \\ w_{1,1}^{m-2} & \cdots & w_{1,k1-1}^{m-2} & w_{1,k1+1}^{m-2} & \cdots & w_{1,m}^{m-2} \end{bmatrix},$$

$$A_4=\begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ w_{2,1} & \cdots & w_{2,k3-1} & w_{2,k3+1} & \cdots & w_{2,m} \\ \vdots & & \vdots & \vdots & & \vdots \\ w_{2,1}^{m-2} & \cdots & w_{2,k3-1}^{m-2} & w_{2,k3+1}^{m-2} & \cdots & w_{2,m}^{m-2} \end{bmatrix}. \quad \text{(A3)}$$

与(9)式类似有 $B_3=A_3^{-1}C_3$, $B_4=A_4^{-1}C_4$, 从(A1)式, α_1, $\cdots$, α_{k1-1}, α_{k1+1}, $\cdots$, α_{k2-1}, $\alpha_{k1}+\alpha_{k2}$, α_{k2+1}, $\cdots$, α_m 可用 $w_{1,k}$, $k=1,\cdots, m$ 得到, 而(A2)式, α_1, $\cdots$, α_{k3-1}, α_{k3+1}, $\cdots$, α_{k4-1}, $\alpha_{k3}+\alpha_{k4}$, α_{k4+1}, $\cdots$, α_m 可用 $w_{2,k}$, $k=1,\cdots, m$ 得到, 因此, α_1, $\cdots$, α_m 可用 $w_{1,k}$ 或 $w_{2,k}$, $k=1, 2, \cdots, m$ 得到, 除非 $k1=k3$ 和 $k2=k4$ 或 $k1=k4$ 和 $k2=k3$, 在这种任意情况, B_3, B_4 变成完全一样, 此时, 节点 $k1$ 和 $k2$(或 $k3$ 和 $k4$)在网络中起相同作用, 这意味着其中一节点可以被删除.

定理 2 的证明.

设满足(4)式的两个网络: $\sum\alpha_k\varphi(W1_k^{\mathrm{T}}X+1)$ 和 $\sum\beta_k\varphi(W2_k^{\mathrm{T}}X+1)$, 则对于任意 j, 有 $\sum_k\alpha_k=\sum_k\beta_k$, $\sum_k\alpha_k w1_{i,k}^j=\sum_k\beta_k w2_{i,k}^j$, $\sum_k\alpha_k w1_{l,k}w1_{i,k}^j=\sum_k\beta_k w2_{i,k}w2_{i,k}^j$. 设 $w1_{\max}(i,\ k1)=\max_k\{|w1_{i,k}|\}$ 和 $w2_{\max}(i,\ k2)=\max_k\{|w2_{i,k}|\}$ 则当 $j\to\infty$ 时, $w1_{\max}^j\gg w1_{i,k}^j$, $k1\neq k$ 和 $w2_{\max}^j\gg w2_{i,k}^j$, $k2\neq k$, 对于任意 i, 有 $w1_{i,k1}=w2_{i,k2}$, $\alpha_{k1}=\beta_{k2}$, 这样,

$$\sum_{k\neq k1}\alpha_k=\sum_{k\neq k2}\beta_k,\ \sum_{k\neq k1}\alpha_k w1_{i,k}^j=\sum_{k\neq k1}\beta_k w2_{i,k}^j,\ \sum_{k\neq k1}\alpha_k w1_{i1,k}w1_{i,k}^j=\sum_{k\neq k1}\beta_k w2_{i1,k}w2_{i,k}^j$$

与上相似, 设 $w1_{\max}(i3,\ k3)=\max_{k\neq k1}\{|w1_{i,j}|\}$和 $w1_{\max}(i4,\ k5)=\max_{k\neq k1}\{|w2_{i,j}|\}$, 则对于任意 i, 有 $w1_{i,k3}=w2_{i,k4}$, $\alpha_{k3}=\beta_{k4}$, 诸如此类, 直到对任意给定的 $w2(i,\ l)$, 都可以找到 $w1(i,\ j)$, 使对于任意 i, $w1(i,\ j)=w2(i,\ l)$. 反之, 对任意给定的 $w1(i,\ l)$, 都可以找到 $w2(i,\ j)$, 使对于任意 i, $w2(i,\ j)=w1(i,\ l)$, 若将两网络节点的顺序重新定义, 则对于任意节点 k, $w2(i,\ k)=w1(i,\ k)$, $\alpha_k=\beta_k$.