

(12)

第17卷第4期  
1998年8月

红外与毫米波学报  
J. Infrared Millim. Waves

Vol. 17, No. 4  
August, 1998

303-307

# 函数逼近神经网络的一种快速学习算法\*

朱炬波 马士林

(国防科技大学七系, 湖南, 长沙, 410073)

TP18

**摘要** 通过分析待学习参数的不同特性, 提出一种快速收敛方法, 使网络的收敛速度大大提高. 通过对多种函数的实验, 与不同的网络结果对比, 表明此方法具有很强的普遍性.

**关键词** 神经网络, 函数逼近, 子波分析, 学习算法.

## 引言

人工神经网络(ANN)是新近发展起来的一门交叉学科, 它涉及生物、电子、计算机、数学和物理等学科. 长期以来, 人们设法用物理实现系统去模仿人脑, 完成类似人脑的工作. 人工神经网络就是采用物理可实现的系统来模仿人脑神经细胞的结构和功能的系统. 人工神经网络技术在模式识别、信号处理、系统辨识、自动控制等方面得到了广泛的应用. 在系统辨识及控制中可以对一般非线性函数进行一致最优逼近.

文献[1]首先提出了用子波分析的方法来构造新的神经网络, 其基本思想是改造 BP 网络的激活函数, 能同时调整伸缩因子. 子波分析是用一族函数表示或逼近一信号或函数, 这一族函数称为子波函数系, 它是通过一基本子波函数的不同尺度的平移和伸缩构成的. 记基本子波函数为  $h(x)$ , 伸缩和平移因子分别为  $a, b$ , 则子波变换函数基底为  $h_{a,b}(x) = h(\frac{x-b}{a})$ . 文献[2]采用的子波基函数为  $h(x) = \cos(1.75x)e^{-x^2/2}$ , 事实上它不是子波基函数, 因为它不满足允许性条件, 正是由于这一点导致它的算法与文献[1]稍有不同(常数参数), 文献[3]提出了任何非多项式函数都可以作为网络的激活函数, 可以任意逼近任何连续函数, 所以文献[2]中的基函数确实可以作为激活函数(为了便于比较, 本文计算中也以它为激活函数).

本文通过分析网络中待学习参数的特性, 认为待学习的参数分为线性参数与非线性参数两类. 根据两类参数不同的特性, 提出改进的学习算法来调整参数.

## 1 改进的学习算法

确定神经网络结构以后, 需要训练网络. 设网络的输入是  $x_i$ , 输出是  $\hat{f}(x_i)$ , 实际函数值  $f(x_i)$  指导网络的学习. 文献[1]的输出为:

\* 国家 863 基金(编号 863-306-ZD-03-3)资助项目  
稿件收到日期 1997-08-13, 修改稿收到日期 1997-12-24

$$\hat{f}(x) = \sum_{i=1}^N \omega_i h\left(\frac{x-b_i}{a_i}\right) + \bar{f}_0, \quad (1)$$

式(1)中,  $a_i, b_i$  是伸缩因子,  $\omega_i$  是权参数,  $\bar{f}_0$  是为了逼近非零均值函数而引进的常数因子参数(因为子波基函数是零均值的, 无法逼近常数函数).

文献[2]的输出为:

$$\hat{f}(x) = \sum_{i=1}^N \omega_i h\left(\frac{x-b_i}{a_i} + f_0\right), \quad (2)$$

式(2)中参数意义同式(1). 因为文献[2]中的基函数是非零均值的, 故不需常数因子  $\bar{f}_0$ . 事实上, 参数  $f_0$  不是独立参数, 可以并入参数  $a_i, b_i$  中. 因此我们确定的输出为:

$$\hat{f}(x) = \sum_{i=1}^N \omega_i h\left(\frac{x-b_i}{a_i}\right). \quad (3)$$

文献[1]、[2]的学习算法为:

- (1) 对  $\omega_i, a_i, b_i$  给定初值 ( $i=1, \dots, N$ );
- (2) 通过共轭梯度法求出新的  $\omega_i, a_i, b_i$  ( $i=1, \dots, N$ );
- (3) 检验误差大小, 达到精度要求则停止, 否则返回(2).

我们观察参数  $\omega_i, a_i, b_i$  在式(3)中所表现的特性. 参数  $a_i, b_i$  是非线性的, 其非线性体现在基函数上, 而  $\omega_i$  则不同, 输出  $\hat{f}(x)$  关于它是线性的. 具体地说, 当  $a_i, b_i$  取定后,  $\hat{f}(x)$  为  $\omega_i$  的线性组合. 充分利用这一点, 使我们在用共轭梯度法求解参数时, 只让  $a_i, b_i$  参与迭代,  $\omega_i$  在每次迭代中最小二乘法计算, 有可能加快收敛速度, 大大减少计算量.

假设所取神经网络隐层数目为  $N$ , 在函数逼近区间取的样点数为  $M$ . 我们把  $a_i$  的初值取成同一值, 大小为每个样点段的长度.  $b_i$  的初值对应于  $N$  个样点段中每个样点段的中间值<sup>[2]</sup>. 网络对于  $x_j$  的输出为:

$$f(x_j) = \sum_{i=1}^N \omega_i h\left(\frac{x_j - b_i}{a_i}\right), \quad (j=1, M) \quad (4)$$

定义 LMS 能量函数:

$$E = \frac{1}{2} \sum_{j=1}^M [f(x_j) - \hat{f}(x_j)]^2,$$

则

$$E = \frac{1}{2} \sum_{j=1}^M [f(x_j) - \sum_{i=1}^N \omega_i h\left(\frac{x_j - b_i}{a_i}\right)]^2; \quad (5)$$

文献[1]、[2]的迭代格式是给定  $a_i, b_i, \omega_i$  的初值, 求出式(5)关于所有参数在初值处的梯度向量, 按能量最优下降方式, 调整参数求  $E$  的最小值点. 由于  $\omega_i$  是线性参数, 我们调整迭代格式: 给定  $a_i, b_i$  的初值, 通过最小二乘法可以求得第一次的  $\omega_i$  ( $i=1, 2, \dots, N$ ) 值, 再利用共轭梯度法:

$$a_i^{k+1} = a_i^k - \alpha \partial E / \partial a_i, \quad (k=1, 2, \dots) \quad (6)$$

$$b_i^{k+1} = b_i^k - \beta \partial E / \partial b_i, \quad (k=1, 2, \dots) \quad (7)$$

调整参数  $a_i, b_i$ . 在式(6)、(7)中  $\alpha, \beta$  可以取为 0.0104.

引入误差函数

$$ERR = \sqrt{\frac{\sum_{j=1}^M [\hat{f}(x_j) - f(x_j)]^2}{\sum_{j=1}^M [f(x_j) - \bar{f}]^2}} \quad (8)$$

来控制迭代次数, 其中  $\bar{f} = \sum_{j=1}^M f(x_j) / M$ .

算法步骤为

- (1) 给定初值  $a_i, b_i, (i=1, \dots, N)$ , 由式(5)用最小二乘法得  $\omega_i, (i=1, \dots, N)$ ;
- (2) 通过共轭梯度法式(6)、(7)迭代新的  $a_i, b_i, (i=1, \dots, N)$ ;
- (3) 检验误差大小(式 8), 达到精度要求为止, 否则返回(1).

## 2 实验结果及分析

### 2.1 对无噪声采样函数的逼近

对多种函数进行实验, 均得到了满意的效果.

例 1<sup>[2]</sup>

$$f(x) = \begin{cases} -2.18x - 12.864, & -10 \leq x < -2; \\ 4.246x, & -2 \leq x < 0; \\ 10e^{-0.05x-0.5} \sin[(0.03x + 0.7)x], & 0 \leq x \leq 10; \end{cases} \quad (9)$$

在  $[-10, 10]$  上均匀取样点 100 个, 隐层单元数为 10, 用本文方法经 363 次迭代,  $ERR < 0.04$ , 而在文献[1]中, 经 10000 次迭代后,  $ERR = 0.05057$ , 文献[2]算法需要 1500 次迭代才使  $ERR < 0.04$ . 在第 363 次迭代后,  $a_i, b_i$  的值见表 1, 函数(9)的逼近结果见图 1.

表 1 函数(9)的逼近结果( $a_i, b_i$  值)  
Table 1 Approximation results of function (9):  $a_i, b_i$

$a_i$ 初值	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
$a_i$ 终值	2.08	2.73	1.93	2.81	1.72	2.04	2.56	2.12	1.75	1.58
$b_i$ 初值	-9.0	-7.0	-5.0	-3.0	-1.0	1.0	3.0	5.0	7.0	9.0
$b_i$ 终值	-10.7	-8.3	-5.1	-2.8	-1.5	0.59	3.19	5.09	6.55	8.54

对上面的函数, 取不同的样点数( $N=10, ERR=0.04$ ), 其结果见表 2.

表 2 改变  $M$  值后函数(9)的逼近结果  
Table 2 Approximation results of function (9) (with  $M$  changed)

样点数 $M$	100	50	20
迭代次数 $K$	363	290	49

由表 2 可见,  $M$  取值不大, 但迭代效果较好, 这验证了小波网络多分辨率的优越性, 它只与函数的平缓(频率)有关, 与样点数关系不大, 另外两个例子(见表 3)也说明了这一点:

表3 其它函数的逼近结果(改变  $M$  或  $N$  值)  
Table 3 Approximation results of other functions (with  $M$  or  $N$  changed)

$f(x)=\sin(2\pi x)$ $x \in [0, 1]$	样点数 $M$	50	40	30
$N=5, ERR=0.04$	迭代次数 $K$	14	4	3
$f(x)=\sin(4\pi x)$ $x \in [0, \frac{1}{2}]$	隐层单元数 $N$	8	4	2
$M=8, ERR=0.04$	迭代次数 $K$	1	2	49

例2 与文献[1]其它结果的对比见表4。

表4 与其它算法逼近结果的比较  
Table 4 Approximation results compared with other methods

	$f(x)$	$e^x$	$\sin x$	$x+1$	$x^2+1$
(1)	ERR	0.0438	0.04	0.04	0.0285
BP网络逼近	迭代次数	1500	3500	5000	6500
(2)	ERR	0.0438	0.0573	0.0460	0.0285
文献[2]结果	迭代次数	1300	500	1300	1000
(3)	样点区间	$[-2.5, 2.5]$	$[0, 2\pi]$	$[-2.5, 2.5]$	$[-2.5, 2.5]$
本文的结果	ERR	0.0438	0.04	0.0460	0.0285
( $N=5, M=10$ )	迭代次数	8	26	10	42

## 2.2 对带噪声采样函数的逼近

对函数加入随机噪声再进行逼近.  $\tilde{f}(x_i) = f(x_i) + a\varepsilon_i$ ,  $\varepsilon_i$  服从  $N(0, 1)$  分布,  $\tilde{f}(x_i)$  为采样值,  $f(x_i)$  为真实值.

例3  $\tilde{f}(x_i) = f(x_i) + \varepsilon_i$ ,  $f(x_i)$  是函数(9), 逼近结果见图2, 其中  $N=5, M=100; ERR=0.07, k=495$ .



图1 函数(9)的网络逼近结果  
Fig. 1 Approximation results of function (9)



图2 带噪声函数(9)的逼近结果  
Fig. 2 Approximation results of function (9) with noise

例4 令  $\tilde{f}(x_i) = f(x_i) + \frac{\varepsilon_i}{5}$ ,  $f(x) = \sin(x)$ ,  $x \in [0, 2\pi]$ ; (10)

函数(10)逼近结果见图3(粗线是比较曲线, 点划线是数据线), 其中  $N=10, M=100; ERR=0.05, K=1.1$ .

从上述例子可见,这种神经网络的抗噪声性较好,这种学习算法受噪声的影响也较小. 本文提出的学习算法,大大加快了网络的收敛速度. 通过对各种函数的实验以及与不同网络算法的对比,显示了此方法的普遍性. 这是由于,我们针对待学习的参数的不同特性,在迭代格式中加以不同的处理,使之改善了寻优的结构,提高了能量函数的下降速度. 但其理论还需进一步研究;另外,这种方法是否能改善子波神经网络的参数求解避开局部极小,也有待进一步讨论. 作者认为,这种方法本身也是一种梯度算法,不大可能改善局部极小的问题. 对于局部极小问题,只能用其它算法如遗传算法,模拟退火算法等等.

我们认为,对于目前神经网络的各种学习算法,仔细分析待学习参数的不同特性,加以合适的处理,有助于加快网络的收敛速度. 本文介绍的方法,可以展望它将在实时数据处理方面得到较好的应用.

**致谢** 本文的计算是在国防科技大学数学技术实验室完成的,在此表示感谢!

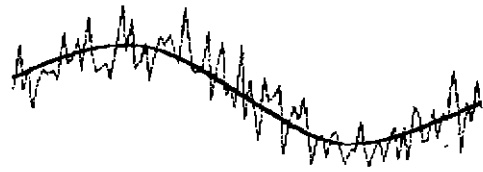


图3 带噪声函数(10)的逼近结果  
Fig. 3 Approximation results of function (10) with noise

#### REFERENCES

- 1 ZHANG Q. Benveniste A. *IEEE Transactions on Neural Networks*, 1992, 3(6): 889~898
- 2 SONG Xiang-Yu, et al. *Journal of Infrared and Millimeter Waves* (宋湘豫等. 红外与毫米波学报), 1997, 16(2): 33~37
- 3 ZHU Wen-Ge. *ACTA Mathematica Applicata Sinica* (朱文革. 应用数学学报), 1997, 20(2): 175~180

### A FAST LEARNING ALGORITHM OF NEURAL NETWORKS FOR APPROXIMATING FUNCTION\*

ZHU Ju-Bo MA Shi-Ling

(Seventh Department, University of National Defense Technology, Changsha, Hunan 410073, China)

**Abstract** A fast learning algorithm of neural network for approximation function was presented by analyzing different characters of parameters in network. This method can speed up the rate of network convergence. By approximating various functions, and comparing with other methods, this algorithm is shown to have satisfactory performances.

**Key words** neural network, approximating function, wavelet analysis, learning algorithm.

\* The project supported by the National High Technology Foundation  
Received 1997-08-13, revised 1997-12-24