Pi-sigma 神经网络混合学习算法及收敛性分析

聂 永,邓 伟
NIE Yong,DENG Wei

苏州大学 计算机科学与技术学院,江苏 苏州 215006
College of Computer Science,Suzhou University of Science and Technology,Suzhou,Jiangsu 215006,China
E-mail:210513053@suda.edu.cn

NIE Yong,DENG Wei.Hybrid learning algorithm for Pi-sigma neural network and analysis of its convergence.Computer Engineering and Applications,2008,44(35):56-58.

Abstract: This paper uses a hybrid genetic algorithm to training Pi-sigma neural network and this algorithm is once applied to resolve a function optimizing problem.The hybrid genetic algorithm incorporates the stronger global search of genetic algorithm into the stronger local search of simplex method,and can search out the global optimum faster than genetic algorithm.The experiments show that the hybrid genetic algorithm can achieve better performance.At last,the hybrid genetic algorithm is proved converge to the global optimum with the probability of 1.

Key words: hybrid genetic algorithm;Pi-sigma neural network;algorithm convergence

摘 要:将一种解决函数优化问题的混合遗传算法用于 Pi-sigma 神经网络的训练。这种混合算法充分利用遗传算法算法的全局搜索能力,又利用了单纯型法的局部搜索能力,因此该混合遗传算法可以使 Pi-sigma 神经网络更快的收敛到全局最优解,而且收敛速度比遗传算法更快。实验证明了这种算法的优越性。最后还证明了该算法可以以概率 1 收敛到全局最优解。

关键词:混合遗传算法;Pi-sigma 神经网络;算法收敛性
DOI:10.3778/j.issn.1002-8331.2008.35.017 **文章编号**:1002-8331(2008)35-0056-03 **文献标识码**:A **中图分类号**:TP18

1 前言

Pi-sigma 神经网络[1]是一种高阶前馈型神经网络,另外还有与 Pi-sigma 神经网络类似的高阶神经网络,比如 sigma-Pi 神经网络,Pi-sigma-pi 神经网络,Sigma-pi-sigma 神经网络等。Pi-sigma 神经网络是由 Ghosh J 和 Shin Y 在 1992 年提出来的,Pi-sigma 神经网络比起多层感知器来,结构简单,训练参数少,收敛速度快。因此,经过 20 年的发展,Pi-sigma 神经网络得到广泛的应用。

Pi-sigma 神经网络的训练方法有很多,最常用的是梯度下降法,牛顿法,共轭梯度法等,这些算法虽然可以使 Pi-sigma 神经网络快速的收敛,但是其缺点也比较明显,其中最突出的是这些算法易收敛到局部最优解而不是全局最优解。因此使用一种解决函数优化问题的混合遗传算法(Hybrid Genetic Algorithm,HGA)[2]来训练 Pi-sigma 神经网络,实验表明这种算法可以使 pi-sigma 神经网络收敛到全局最优解,而且速度比单独使用遗传算法要快。文献[2]所用混合遗传算法是把遗传算法和可变多面题法(也叫单纯型法)相结合,在原有算法的基础之上做了修改,把原有算法中的的遗传算法改为带有精英保留策略的遗传算法。最后还证明了该混合遗传算法以概率 1 收敛到全局最优解。

2 Pi-sigma 神经网络及混合遗传算法

2.1 Pi-sigma 神经网络

Pi-sigma 神经网络是一种高阶前馈型神经网络,与 BP 神经网络不同的是,在 Pi-sigma 神经网络模型的输出层中采用乘法神经元代替了加法神经元。Pi-sigma 神经网络的一个特点就是隐层和输出层的连接权值固定为 1,在学习训练中不需要改变,因此 Pi-sigma 神经网络的训练参数较少,收敛速度也较快。

Pi-sigma 神经网络的拓扑结构如图 1。

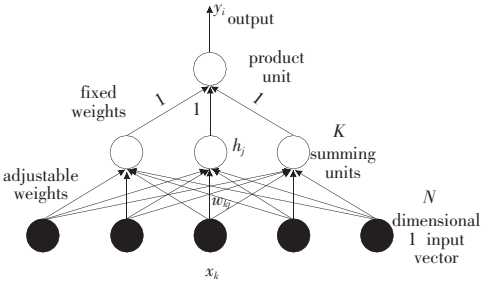


图 1 Pi-sigma 神经网络拓扑结构图

Pi-sigma 神经网络的输入 $\boldsymbol{x}$ 是一个 N 维矢量,x_k 是 $\boldsymbol{x}$ 的第 k 个分量,神经网络的隐层包含有 K 个加法神经元,设 h_j 为第 j

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60572074)。
作者简介:聂永(1983-),硕士,主要研究方向:神经网络;邓伟(1967-),副教授,主要研究方向:神经网络、人工智能、模式识别等。
收稿日期:2007-12-25 **修回日期**:2008-03-03

个隐层神经元的输出，于是：

$$h_j=\sum_{k=1}^{N}w_{kj}x_k+\theta_{kj},j=1,2,\cdots,K \tag{1}$$

$$y_i=\sigma(\prod_{j=1}^{K}h_j) \tag{2}$$

其中 w_{kj} 为输入 x_k 和第 j 个加法神经元之间的连接权值，θ_{kj} 为第 j 个加法神经元的阈值。

$\sigma(x)$ 代表的是非线性激活函数，这里使用的是 sigmoidal 函数，即：

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

2.2　混合遗传算法

2.2.1　遗传算法

文献[2]中的混合遗传算法中的遗传算法部分采用的是实数编码方式，正交交叉算子和高斯变异算子。在原有算法的基础之上修改了一下遗传算法部分，为后面证明这种混合遗传算法的收敛性做准备。

设两个父体分别为 P 和 Q，则基于实数编码方式的正交交叉操作步骤如下：

(1)混合交叉

$X_1[i]=P[i];X_2[i]=Q[i];X_3[i]=r*P[i]+(1-r)*Q[i],i=1,2,\cdots,n$

r 为一参数，$0<r<1$，这里取 $r=0.5$。

(2)用 X_1、X_2、X_3 按正交表 L_9 产生 9 个新个体并分别计算他们的适应度值。

(3)按照正交实验方法计算最佳水平组合并产生对应的第 10 个个体，计算其适应度值。

(4)从 X_1、X_2、X_3 和新产生的个体中选择最好的两个个体取代 P 和 Q。

变异操作如下：

采用高斯变异算子，在实数编码方式下，变异操作对个体 X 的每一个分量 $X[i]$ 作用一个随机偏差量，即：$X'[i]=X[i]+\delta,i=1,2,\cdots,n$，其中，$\delta$ 为标准正态随机变量。

修改后的遗传算法描述如下：

(1)初始化，随机产生一个分布均匀的初始群体(包含 N 个初始解)；

(2)从上一代中找出适应度函数值最大的个体，保留起来，不参与下面的交叉和变异运算；

(3)交叉运算，按两两配对原则将群体中的个体配对，并且执行正交交叉运算；

(4)变异运算，将群体中的每个个体按照一定的变异率进行变异操作；

(5)若满足中止条件，则算法中止，否则，转步骤(2)。

2.2.2　局部搜索

局部搜索采用的可变多面体法也称为单纯型法。单纯型法用 $n+1$ 个 n 维欧氏空间中的顶点来构造搜索过程中的多面体，选取 $n+1$ 个相邻的个体作为初始顶点。单纯型法包含下列几种操作：

(1)找出 $n+1$ 个顶点函数值最大及最小的点 X_h 和 X_h，然后找出去掉 X_h 后的由 n 个顶点构成的多边形的形心 X_c；

(2)通过反射操作得到反射点 X_r：$X_r[k]=X_c[k]+a*(X_c[k]-X_h[k])$，其中 $X[k]$ 为 X 的第 k 个分量，a 为反射系数；

(3)若 $f(X_c)<f(X_l)$，则执行扩大操作 $X_e[k]=X_c[k]+r*(X_r[k]-X_c[k])$，其中 $r>1$ 为扩展系数；

(4)若对多边形中除去 X_h 外的任一顶点 X_i，均有 $f(X_r)>f(X_i)$，则执行收缩操作，得到 X_s：$X_s[k]=X_c[k]+b*(X_h[k]-X_c[k])$，其中 $0<b<1$ 为收缩系数；

(5)若 $f(X_r)>f(X_h)$，则使所有点向最小点靠近，即令 $X_i[k]=X_l[k]+0.5*(X_i[k]-X_l[k])$，其中 $X_i[k]$ 为第 i 个顶点的第 k 个分量；

(6)令 X_r、X_e 和 X_s 中的最好的点代替 X_h。

2.2.3　混合遗传算法描述

使用遗传算法重要的一步就是确定适应度函数，这里使用的适应度函数和神经网络输入输出的均方误差有关。因此定义适应度函数如下：

$$fit(p)=\frac{1}{1+MES(p)} \tag{4}$$

其中 $MES(p)$ 为神经网络输入输出的均方误差，p 代表样本，$MES(p)$ 定义为：

$$MES(p)=\sum_{i=1}^{n}(y_i-c_i)^2 \tag{5}$$

其中 y_i 为 pi-sigma 神经网络的实际输出，c_i 为神经网络的期望输出。

混合遗传算法具体步骤描述如下：

(1)初始化种群，假设种群大小为 M，染色体采用实数编码方式，编码包括 pi-sigma 神经网络的连接权值，阈值等；确定交叉概率 p_c，变异概率 p_m，最大代数，停止准则等；

(2)按照式(4)计算适应度函数值；

(3)按照赌盘方法进行选择，为了保证混合算法的收敛性，选择出适应度函数值最大的染色体并保留下来，不参与下面的交叉和变异操作；

(4)交叉运算，按照交叉概率 p_c 对上步选择出来的染色体进行交叉运算；

(5)变异运算，按照变异概率 p_m 进行变异运算；

(6)解码染色体，并使用局部搜索算法进行局部优化；

(7)检查是否满足停止准则，若满足转步骤(9)，否则转步骤(8)；

(8)对局部优化算法求得的解重新编码，转步骤(2)；

(9)算法结束。

3　混合 GA 的收敛性分析

证明 1　单纯形法的收敛性

证明：单纯形法是一种直接搜索方法，其基本思想和其他直接方法有所不同。通过反射、扩展、压缩等方法，求得一个较好点，用它取代最高点，构成新的单纯型，这样每一次迭代的解距离最优解越来越近，因此经过有限次迭代以后总会收敛到全局最优解[3]。

证明 2　带有精英保留策略的十进制遗传算法以概率 1 收敛到全局最优解

证明：设 λ_i 为搜索空间，F_k 为第 k 代搜索空间 λ_i 中最大的适应度值，F^* 为所求问题的最优解，令 $\lim \boldsymbol{P}(F_k=F_0=F^*)=1$。则遗传算法的传输矩阵为：

$$\boldsymbol{P}=\begin{pmatrix}\boldsymbol{Q} & 0\\ T & S\end{pmatrix}=\begin{pmatrix}1 & 0\\ T & S\end{pmatrix} \tag{6}$$

其中 $\boldsymbol{Q}$ 为闭群，是 A_0 的转换矩阵，并且 Q 只有一个元素就是 1，$\boldsymbol{P}$ 是一个转换矩阵。根据 Markov 理论，可以得到结论：经过有限步迭代，搜索空间 λ_i 可以达到 A_0，即：

$$\lim \boldsymbol{P}(F_k=F_0=F^*)=1 \tag{7}$$

这也就是说，带有精英保留策略的实数编码遗传算法可以以概率 1 收敛到全局最优解[4-5]。

证明 3 本文的混合遗传算法以概率 1 收敛到全局最优解

证明：要证明混合遗传算法以概率 1 收敛，就要证明 $f(x^k)$ 以概率 1 收敛到全局最优解 $f(x^*)$。设 $\{x^1,x^2,\cdots,x^k,\cdots\}$ 是混合遗传算法产生的一个序列，令 $x^*=\arg\min f(x)$。

$N_\varepsilon=\{x|x\in s,f(x)-f(x^*)<\varepsilon\}$，其中 N_ε 代表全局最优解 x^* 的 ε 邻域，ε 为一个充分小的数。E_k 为混合算法产生的序列 x^k 落入邻域 N_ε 的事件。由于遗传算法采用了精英保留策略，使得混合算法产生的序列是非增序列，即：

$$f(x^1)\geqslant f(x^2)\geqslant\cdots\geqslant f(x^k) \tag{8}$$

式(8)每一项同时减去全局最优解 $f(x^*)$ 得到：

$$f(x^1)-f(x^*)\geqslant f(x^2)-f(x^*)\geqslant\cdots\geqslant f(x^k)-f(x^*) \tag{9}$$

这就说明了，当 x^1 落入 N_ε，则 x^2 一定落入 N_ε，即：

$$E_1\subset E_2\subset\cdots\subset E_k\subset\cdots \tag{10}$$

由此可知，$\boldsymbol{P}(E_k)$ 是个非递减序列，对任何 $k\in R$，有：

$$p\{\omega|\omega\notin E_k,k>M\}=1-p\{\omega|\omega\notin E_k,k>M\}<\lambda$$

$$p\{\omega|f(x^k)>f(x^*)+\varepsilon\}\leqslant p\{\omega|x^k\notin N_\varepsilon,k>M\}\leqslant\lambda^{k-M}$$

$$\lim_{k\to\infty}p\{\omega|f(x^k)>f(x^*)+\varepsilon\}=0$$

故，$\boldsymbol{P}(E_k)$ 是一个非递减序列且有上确界的序列，故 $\lim_{k\to\infty}\boldsymbol{P}(E_k)$ 存在。且：

$$\lim_{k\to\infty}p(E_k)=p(\bigcup_{k=1}^{\infty}E_k)=1 \tag{11}$$

任给一个正数 λ，$0<\lambda<1$，则存在一个正数 M，当 $k>M$ 时，有：$p(E_k)\geqslant 1-\lambda$，但 $p\{\omega|\omega\notin E_k,k>M\}=1-p\{\omega|\omega\notin E_k,k>M\}<\lambda$。任给一个正整数 ε，$p\{\omega|f(x^k)>f(x^*)+\varepsilon\}\leqslant p\{\omega|x^k\notin N_\varepsilon,k>M\}\leqslant\lambda^{k-M}$，因此：

$$\lim_{k\to\infty}p\{\omega|f(x^k)>f(x^*)+\varepsilon\}=0 \tag{12}$$

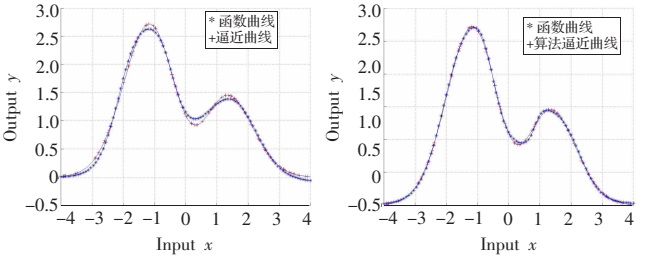
结论：带有精英保留策略的混合遗传算法以概率 1 收敛到全局最优解。

4 实验及仿真

为了验证算法的有效性，做了函数逼近实验，并和遗传算法做了对比。测试函数使用的 hermite 多项式，定义如下：

$$h(x)=1.1(1-x+2x^2)\exp(-\frac{x^2}{2}) \tag{13}$$

实验是在 Matlab6.5 环境下完成的，遗传算法采用的精英保留策略的实数编码方式，局部搜索算法采用的是单纯形法，训练过程中 GA 和 HGA 采用相同的训练参数，函数逼近的效果如图 2。



(a)GA 算法的函数逼近图 (b)HGA 算法的函数逼近图

图 2 GA 和 HGA 算法的函数逼近图

算法分析如表 1 所示。

表 1 算法分析表

算法	训练时间/s	训练误差	遗传代数
HGA 算法	6.750	0.056 787	15
GA 算法	9.054	0.297 409	24

由图 2 和表 1 可知，采用混合遗传算法性能、收敛速度都要优于遗传算法，由此可见 HGA 的优越性。

5 结论

使用一种函数优化问题的混合遗传算法训练 pi-sigma 神经网络，克服了 GA 算法收敛速度慢的问题。实验结果表明，该方法可以得到比较好的结果。

参考文献：

[1] Ghosh J,Efficient S Y.Higher-order neural networks for classification and function approximation[J].Int J Neural Syst,1992,4(3):323-350.

[2] 彭伟，卢锡城.一种函数优化问题的混合遗传算法[J].软件学报，1999,10(8):819-823.

[3] Avriel M.Nonlinear programming:analysis and methods[M].[S.l.]:Prentice-Hall Inc.,1976.

[4] YAO Wen-jun.Study on back propagation network optimization with genetic algorithms[J].Wuhan Inst Chem Tech,2004,16(1):124-129.

[5] Rudolph G.Convergence analysis of canonical genetic algorithm[J].IEEE Trans on Neural Networks,1994,5(2):96-101.

(上接 42 页)

[6] 夏桂梅，曾建潮.一种基于单纯形法的随机微粒群算法[J].计算机工程与科学，2007,29(1):90-93.

[7] 李爱国.多粒子群协同优化算法[J].复旦学报：自然科学版，2004,43(5):923-925.

[8] 张丽平，俞欢军，陈德钊，等.粒子群优化算法的分析与改进[J].信息与控制，2004,33(5):513-517.

[9] Shi Y,Eberhart R C.Fuzzy adaptive particle swarm optimization[C]//Proceedings of the IEEE Conference on Evolutionary Computation.Soul:IEEE,2001:101-106.

[10] Xie X F,Zhang W J,Yang Z L.Dissipative particle swarm optimization[C]//Proceedings of the 2002 Congress on Evolutionary Computation.Hawaii:IEEE,2002:1456-1461.

[11] Yin P Y,YU S S,WANG P P,et al.A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems[J].Computer Standards & Interfaces,2005,28(4):441-445.