进化神经网络中的变异算子研究*

郑志军, 郑守淇

(西安交通大学 计算机科学与技术系,陕西 西安 710049)
E-mail: zjzheng@263.net; sqzheng@xjtu.edu.cn
http://www.xjtu.edu.cn

摘要: 针对进化神经网络中遗传算法收敛速度慢和容易早熟这两个难题,提出了一个启发性的变异算子.该算子采用了自适应的变异率和启发式的变异位的选择策略.在多代无进化时,通过提高变异率扩大搜索范围,同时减小变异量进行更细致的搜索.求解 XOR 问题的实验表明,该算法既具有很快的收敛速度又能自动维持群体的多样性.

关 键 词: 遗传算法;进化;神经网络;启发式变异算子;多样性
中图法分类号: TP18 文献标识码: A

由于遗传算法存在着容易早熟和收敛速度慢两个难题,尤其是它的很弱的局部搜索能力,使得神经网络训练精度不能达到很高,许多研究者因此认为它不适合进化神经网络[1].

提高遗传算法的搜索能力,关键是设计合理的遗传算子.而在 3 个遗传算子中,只有变异操作才能使搜索跳出早熟,到达任何范围,实现全局搜索,也只有它具备局部搜索能力,因此受到格外的关注.

对变异的研究集中于变异率上,如采用自适应变异率来加快收敛速度[2],为了跳出早熟,算法中引入大变异算子[3].但对变异点的选择研究得不多,虽然文献[4]给出了一种确定变异点的有指导的变异方法,但由于采用梯度算法,有陷入局部极小点之嫌.

通过反复的实验和归纳研究,文中基于效率较低的二进制编码提出了一套启发性的变异算子,并以进化前向神经网络的权值为例,证明该算子可指导搜索沿着尽可能快的方向找到最优解,效果非常好.

1 传统变异算子的缺陷

设计变异算子主要要解决 3 个问题:变异率、变异位置和变异量的选择.

对于变异率,采用自适应的变异率将使算法的性能得到改善,但其计算复杂,且虽然考虑了个体的适应度、平均适应度、最大适应度等情况[2],但实际上还有其他因素可能会对算法性能产生关联影响,仅仅考虑这些因素未必全面,因此在一些应用中算法的效果并不好.

在遗传算法中,搜索总是沿着选择和交叉操作指示的方向朝着较优区域进行的,它只是通过变异操作实现以这一方向为中心的某一邻域内的局部搜索.若变异量大,则搜索步长大,在最优解附近可能毫无效果,浪费了搜索时间.若变异量小,则搜索步长小,搜索范围窄,搜索速度会很慢.所以搜索步长的大小直接影响了搜索效率,而搜索步长由变异量决定.在进化神经网络中,由于权值是由二进制编码串表示的,因此串中不同的位置表示的权值大小不同,越往高位所表示的权值就越大,低位则相对较小,如图 1 所示.对第 k 位来说,其表示的权值大小为 $2^{k-1}/(2^u-1)$.对高位的变异(1→0 或 0→1)将使权值改变很大,而低位变异则改变较小.因此,变异位置决定了变异量.研究表明,在进化前期应该加大搜索步伐,以扩大搜索范围,选准搜索方向;在进化后期,搜索步伐应该变小,以

* 收稿日期: 2000-09-07; 修改日期: 2000-11-20

作者简介: 郑志军(1967－),男,浙江江山人,博士,讲师,主要研究领域为计算智能,分布式计算,计算机系统结构;郑守淇(1927－),男,上海人,教授,博士生导师,主要研究领域为计算机系统结构,分布式计算,人工智能,计算机网络.

细化搜索尽快找到最优解.如果机械地按照这种理论,在进化前期变异点应该选择在高位,而在变异后期变异点应该选择在低位.但实验表明,这种作法在进化开始不久就会出现早熟,究其原因主要是在进化前期变异点过于集中在高位,群体的多样性太差的缘故.

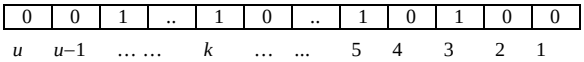

Fig.1 Binary representation of connection weight

图 1 连接权值的二进制编码表示

传统的变异是按一定的概率决定是否对染色体的每一位进行变异,但在进化神经网络中这一方法显然不合适,因为即使变异率很小,但只要变异发生在较高的基因座上,就可能造成权值的极大改变,使得变异量的大小难以控制.在进化神经网络中,变异的单位表示一个连接权的染色体段.因此变异操作应改为:按一定的概率决定是否对染色体中表示连接权的每一段进行变异,若变异,则按一定的策略选择变异点.

针对上述问题,文中设计了一种启发式变异算子,该算子通过引入自适应的变异率和启发式的变异位的选择,实现算法的快速搜索.

2 启发式的变异算子

2.1 自适应的变异率

自适应的变异率定义为

$$P_m(g)=0.001+NG*cof, \tag{1}$$

其中 g 表示当前代数;NG 表示自上次出现最优解以来至当前代为止连续未出现更优解的代数;cof 是个变异率提高的系数,通常取值很小,如 0.005,它决定了阈值.

下面对其进行简要分析:

若进化过程顺利(指每代都有更优解出现),NG=0,变异率 $P_m(g)$=0.001≈0,则可认为选择操作和交叉操作效果较好,不需要引入变异操作,让其快速进化.

而未进化时间越长,NG 越大,此时变异率 P_m 升高,可认为依靠选择操作和交叉操作难以在现有的群体中找到最优解,因而需要扩大搜索范围.

当未进化代数达到设定的某一阈值时(若 cof=0.005,则阈值为 200 代),P_m=1,可认为此时群体已陷入早熟状态,进化过程已经停止,必须强制执行变异操作,使算法跳出早熟状态.

可以证明,当进入早熟状态之后,该算法最多经过 $\frac{u}{2v}$ 次变异即可跳出,其中 u 表示每个连接权的编码长度,v 表示每次变异的基因位的个数(一般取值较小).举例来说,若每个权值用 16 位二进制编码,每次变异其中的 2 位,则经过 4 次变异即可跳出早熟区域.

该算法只需考虑进化结果而无须考虑影响进化的因素,简单易用.对于早熟状态,它不是通过一次很大的变异跳出,而是通过连续几次变异量不大的变异逐步跳出早熟区域,而且只要有一次成功就结束强变异操作,因而不致过多破坏当前的进化成果.

2.2 启发式变异位的选择

进化神经网络权值的过程,实质上是一个反复调整权值以满足学习误差要求的过程.在相应的遗传算法中,是通过在染色体对应的二进制数据串上加上(或减去)一个权值的变化量实现的.权值的变化量可以用一个其中一位为 1 而其他位为 0 的二进制数据串来表示,如图 2 所示.在 k 位置为 1 的二进制数据串可表示大小为 $2^{k-1}/(2^u-1)$的权值的变化量.因此遗传算法的变异运算可以通过将原染色体对应的二进制数据串和一个用二进制数据串表示的变异量相加(或相减)运算来实现.

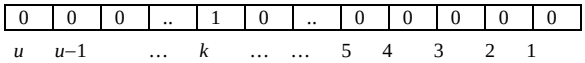

Fig.2 Binary representation of connection weight variable

图 2 连接权变化量的二进制编码表示

在搜索初期,k 应尽量取在高位,以扩大搜索范围;如果经过一定代数的搜索还不能发现更优解,则认为目前搜索步长太大,应减小变异量,进行更细致的搜索,此时 k 应该向低位移动,移动的步长 dynstep 可用下式计算:

```
if (counter>=n){
    counter=0;
    dynstep=dynstep+1;
}
```

其中n为设定的连续未出现更优解的代数的阈值,一般取为几十,*counter*为计数器,初时置*dynstep*=0,*counter*=0.

通过大量的实验发现,权值的变化量以不超过染色体所表示的最大权值的一半为宜,即随机产生的 k 的取值范围为[1,u−1].

考虑了大小随进化过程动态变化这一因素,权值的变化量可用在第 m 位置 1 其他位置 0 的二进制串表示,如图 3 所示.

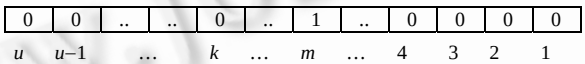

Fig.3 Final binary representation of connection weight variable

图 3 连接权变化量的最终二进制表示

其中 m 的计算公式如下:

$$m=\begin{cases}1, & k-dynstep\le 1,\\ k-dynstep, & k-dynstep>1.\end{cases} \tag{2}$$

尽管 k 是随机产生的,值的大小可能大也可能小,但变量 *dynstep* 使得 k 总是向低位移动.

权的调整既可能沿增加的方向也可能沿减小的方向变化,因此进化神经网络中的变异操作可通过下列的二进制运算实现:

$$A'=\begin{cases}A+B, & r=0,\\ A-B, & r=1.\end{cases} \tag{3}$$

其中A为待变异的染色体对应的二进制数据串,B为权值变化量对应的二进制数据串,r为随机产生的二进制数.

如:某染色体对应的二进制数据串为:0 1 0 0 1 1 1 1 1 0 0 0 1 1,

权值的变化量对应的二进制串为:0 0 0 0 0 0 0 0 1 0 0 0 0 0,

则当 r=0 时变异后的染色体变为:0 1 0 1 0 0 0 0 0 0 0 0 1 1.

下面作几点说明:

当 $P_m(g)$达到 1 时,说明无论是粗化搜索还是细化搜索都不可能找到更优解,必须强制进行变异操作,跳出现有区域,在新的区域内从头开始下一轮搜索.此时需将 *counter*=0,*dynstep*=0,以便搜索开始时步长恢复最大,扩大范围,加快速度.

同样的理由,当出现更优解时,也需将*counter*,*dynstep*置0.

需要进一步澄清的是,在上述置0操作中提到的搜索初期并不是指整个进化过程的初期,而是指搜索进入一个新的更好的区域后(早熟区域可以被认为是最差的区域)从头开始下一轮搜索的初期.

2.3 启发式变异算子对群体多样性的影响

由于变异的单位对应于一个连接权的染色体段,所以只要考察一段即可,而不必考察整个染色体.

考察如图 4 所示的染色体,若在第 m 位发生了变异(可用第 m 位加 1 表示),则在染色体的基因串 $a_{m+1}a_{m+2}a_{m+3}..a_{u-1}a_u$ 的每一位都有可能发生连锁变异.

a_u	a_{u-1}	..	a_{m+2}	a_{m+1}	a_m	..	..	a_3	a_2	a_1
u	$u-1$	...	$m+2$	$m+1$	m	…	…	…	3	2	1

Fig.4 Sketch map of mutation zones of chromosome with length u

图 4 长度为 u 的染色体变异区域示意图

对第 $m+1$ 位而言,只有当 $a_m=1$ 时才有可能变异,由于染色体的每一位是随机产生的,第 m 位为 0 或 1 的概率各为 $\frac{1}{2}$,由此可以认为,该位变异的条件概率为

$$P(a_{m+1}/a_m)=\frac{1}{2}. \tag{4}$$

由于第 $m+1$ 位变异的前提是第 m 位的变异,因此 $P(a_{m+1})$等于 $P(a_{m+1}a_m)$,有

$$P(a_{m+1})=P(a_{m+1}a_m)=P(a_{m+1}/a_m)*P(a_m)=\frac{1}{2}*1=\frac{1}{2}=(\frac{1}{2})^1. \tag{5}$$

对第 $m+2$ 位而言,它的变异与否取决于第 $m+1$ 位的情况,同理可得,$P(a_{m+2}/a_{m+1})=\frac{1}{2}$.

$$P(a_{m+2})=P(a_{m+2}a_{m+1}a_m)=P(a_{m+2}/a_{m+1})*P(a_{m+1})=\frac{1}{2}*\frac{1}{2}=(\frac{1}{2})^2. \tag{6}$$

依此类推,对第 u 位而言,变异概率为

$$P(a_u)=P(a_ua_{u-1\ldots}a_{m+2}a_{m+1}a_m)=(\frac{1}{2})^{u-m}. \tag{7}$$

这说明,若在染色体的某一位(第 m 位)发生了变异,则将造成第[$m+1$,u]位共 $u-m$ 位发生连锁变异的可能性为$(\frac{1}{2})^{u-m}$.

举例来说,若权值用 16 位二进制编码表示,u=16,当变异发生在第 14 位时,则除了第 14 位变异外,还有$(\frac{1}{2})^{16-14}$ 即 25%的概率使 15,16 位发生变异;当变异发生在第 4 位时,则除了第 4 位变异外,还有$(\frac{1}{2})^{16-4}$ 即 0.0244%的概率使其他 12 位发生变异.

显然,与传统的变异相比,同样只进行一次变异,启发式变异算子将使群体保持更好的多样性.尤其是当变异发生在低位时,此时是搜索步长较小的时候,本应特别容易早熟,而采用本算法之后却使群体的多样性达到高峰.换句话说,当陷入早熟后,经一次变异,有非常大的概率即可使算法脱离早熟状态.

3 实例验证

以遗传算法训练一个 3 层 BP 网的权值为例,训练样本来源于一个 3 输入的异或求解问题.

实验参数为输入层单元数为 3,1 个隐层,3 个隐层单元,输出层单元数为 1,共有 16 个权值(含 4 个阈值),初始权值的范围为[−10.0,10.0],每个权值采用 16 位二进制编码以提高误差精度,因而染色体总长度为 16*16=256 位,群体规模为 20,交叉率为 0.95,结束条件是所有样本的累加误差平方和小于 10^{-6},cof=0.005.

实验表明,当学习误差精度要求较高时,用标准遗传算法没有一次能求出满意解,总是进化到一定代数时即发生早熟现象.因此实验中不采用标准遗传算法作参照系,而采用性能更优的在选择前保留当前最优值的改良算法(文中称其为改良标准遗传算法)作参照系.

实验**1**. 随机考察一次进化过程,表1是运行结果.

从表1可见,整个过程保持了很高的进化速度.以算法的设计标准,即200代未出现更优解即被视为早熟的标准来看,进化过程只出现过一次早熟,并经一次变异操作就跳出了,这说明算法不易陷入局部极值点,即使出现了这种情况,也具有迅速逃离的能力.

Table 1 The result of a random experiment

表1 一次随机实验结果

Generations①	Training error of best individual②
0	1.335 280
100	0.027 047
200	0.000 784
300	0.000 057
400	0.000 038
500	0.000 031
600	0.000 030
700	0.000 029
800	0.000 026
900	0.000 025
1 000	0.000 015
1 100	0.000 012
1 200	0.000 009
1 300	0.000 007
1 400	0.000 006
1 500	0.000 006
1 600	0.000 006
1 700	0.000 005
1 800	0.000 002
1 847	0.000 001

①代数,②最优个体学习误差.

实验 **2**. 分别用改良标准遗传算法、自适应变异遗传算法和启发式的变异算法训练 BP 网的权值.进行 100 次重复实验,每次实验初始群体随机产生,当运行代数达到 40000 代而仍未满足学习误差精度时则认为算法已陷于局部最优解.实验数据见表 2.

Table 2 Comparison of performance of three algorithms

表 2 3 种算法性能比较

	Improved canonical genetic algorithm①	Adaptive mutation algorithm②	Heuristic mutation algorithm③
Average generations for convergence④	24 257	10 993	1 852
Number of local optimum⑤	57	0	0
Fault rate⑥(%)	57	0	0

①标准遗传算法,②自适应变异算法,③启发式变异算法,④平均收敛代数,⑤陷于局部最优解的次数,⑥失败率.

从中可见,算法的改进效果非常明显.

实验 **3**. 分别用改良标准遗传算法、自适应变异遗传算法、启发式的变异算法和δ学习规则训练 BP 网络中的权值.BP 算法采用惯性校正法,惯性系数为 0.9,学习系数为 0.7.机器为 Pentium II /200,32M 内存.四者运行时间见表 3.

Table 3 Comparison of running time for training BP with four algorithms

表 3 4 种算法训练 BP 网运行时间比较

	BP algorithm①	Improved canonical genetic algorithm②	Adaptive mutation algorithm③	Heuristic mutation algorithm④
Running time (min.)⑤	22	19	11	2

①BP 算法,②改良标准遗传算法,③自适应变异算法,④启发式变异算法,⑤运行时间(分).

结果表明,改良标准遗传算法的训练速度和δ学习规则的训练速度差别不大,自适应变异的遗传算法的训练时间则大大缩短,启发式变异算法的运行时间只有 BP 算法的十分之一.

4 结 论

对 XOR 问题的实验表明:

(1) 文中提出的带启发式变异算子的进化算法较之传统算法具有很快的搜索速度和良好的全局搜索能力;

(2) 只要算法设计得当,用遗传算法进化神经网络仍然可以达到很高的学习精度和很快的学习速度,其速

度可以远远高于神经网络本身采用的学习算法的学习速度,用遗传算法进化神经网络是完全可行的.

当然,文中提出的算法还有待于进一步推敲和改进,以更好地发掘遗传算法的潜力,解决更多的应用问题.

References:

[1] Yao, X., Liu, Y. A new evolutionary system for evolving artificial neural networks. IEEE Transactions on Neural Networks, 1997,8(3):694~713.

[2] Srinivas, M., Patnaik, L.M. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man, & Cybernetics, 1994,24(4):656~665.

[3] Ma, Jun-shui, Liu, Gui-zhong, Jia, Yu-lan. The big mutation: an operation to improve the performance of genetic algorithms. Control Theory and Applications, 1998,15(3):404~407 (in Chinese).

[4] Bhandari, D., Pal, N.R., Pal, S.K. Directed mutation in genetic algorithms. Information Sciences, 1994,79(3):251~270.

附中文参考文献:

[3] 马钧水,刘贵忠,贾玉兰.改进遗传算法搜索性能的大变异操作.控制理论与应用,1998,15(3):404~407.

Study on a Mutation Operator in Evolving Neural Networks*

ZHENG Zhi-jun, ZHENG Shou-qi

(*Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an* 710049, *China*)

E-mail: zjzheng@263.net; sqzheng@xjtu.edu.cn

http://www.xjtu.edu.cn

Abstract: In order to solve two difficult problems of premature convergence and slow searching speed of genetic algorithms in evolution neural network, a heuristic mutation operator is presented in this paper. Adaptive probability of mutation and heuristic mutation points selected is applied in it. When no evolution appears after many generations, the range of search will be extended by increasing probability of mutation, and a fine search will be started. The experiments of XOR problem demonstrate that the operator has fine ability of speedy convergence and maintains the diversity of the population automatically.

Key words: genetic algorithm; evolution; neural network; heuristic mutation operator; diversity

* Received September 7, 2000; accepted November 20, 2000