

改进进化策略及其在神经网络训练中的应用

柯 晶¹ 姜 静² 李歧强¹

¹(山东大学控制科学与工程学院, 济南 250061)

²(海军航空工程学院自动控制系, 烟台 264001)

E-mail: kejing@tom.com

摘 要 进化策略是一类策略参数自适应进化算法。文章提出了一种改进进化策略(MES), MES采用基于个体排序的随机自适应 Gaussian-Cauchy 混合变异策略, 将 Gaussian 和 Cauchy 变异算子结合起来以达到全局探索和局部搜索之间的动态平衡。此外, MES 还使用重组算子以进一步提高算法的性能。将该算法用于多层前向神经网络训练, 数值仿真结果显示了该算法的有效性。

关键词 进化算法 进化策略 神经网络

文章编号 1002-8331-(2006)04-0068-03 文献标识码 A 中图分类号 TP18

Modified Evolution Strategy and its Application to Neural Network Training

Ke Jing¹ Jiang Jing² Li Qiqiang¹

¹(School of Control Science and Engineering, Shandong University, Jinan 250061)

²(Department of Automatic Control, Naval Aeronautical Engineering Institute, Yantai 264001)

Abstract: Evolution strategies are a class of evolutionary algorithms with self-adaptation of strategy parameters. A Modified Evolution Strategy (MES) is proposed. MES employs random adaptive Gaussian-Cauchy hybrid mutation strategy based on the rank order of individuals. Gaussian and Cauchy mutation operators are combined to achieve dynamic balance of global exploration and local search. Recombination operator is also adopted in MES to further enhance the performance. The proposed algorithm is applied to multilayer feedforward neural network training. Numerical simulation results show the effectiveness of the proposed algorithm.

Keywords: evolutionary algorithm, evolution Strategies, Neural Network

1 引言

进化算法是一类基于遗传变异和自然选择等生物进化机制的全局随机搜索算法。进化算法包括4个主要分支:遗传算法、进化策略(Evolution Strategies, ES)、进化规划和遗传规划^[1]。

与其它类型的进化算法不同,进化策略最初主要是为求解参数优化问题而设计的。此外,进化策略也是进化算法中最早引入自适应机制的算法^[2]。

$(\mu+\lambda)$ -ES 和 (μ, λ) -ES 是目前最常用的两种进化策略^[1,2]。在 $(\mu+\lambda)$ -ES 中, μ 个父代个体产生 λ 个子代个体,然后从这 $\mu+\lambda$ 个父代个体和子代个体的并集中选出 μ 个最好的个体进入下一代进化。而在 (μ, λ) -ES 中,同样是 μ 个父代个体产生 λ 个子代个体,但只从 λ 个子代个体构成的集合中选出 μ 个最好的个体替换父代进入下一代进化。由于 $(\mu+\lambda)$ 选择方法不能适应变动的环境,不利于实施策略参数的自适应机制,因而目前的研究者更倾向于采用 (μ, λ) -ES^[1,2]。

该文提出了一种改进进化策略(Modified Evolution Strategy, MES)。MES采用随机自适应 Gaussian-Cauchy 混合变异和重组

算子以提高算法性能。作为具体应用,将该算法用于多层前向神经网络训练,仿真结果显示该算法优于传统的进化策略方法。

2 进化策略

2.1 基本进化策略

进化策略适合于非线性、不可微和多极值复杂函数的全局优化问题,此类全局优化问题可以归结为如下形式的函数极值问题:

已知目标函数 $f(x)$ 为定义在 $S \subseteq R^n$ 上的实函数,欲 $f(x)$ 求的极小值,即求点 $x^* \in S$ 使得:

$$\forall x \in S: f(x^*) \leq f(x)$$

由文献[3],求解上述函数优化问题的基本 (μ, λ) -ES 可以归纳为如下步骤:

(1) $g=0$, 随机生成 μ 个初始个体。每个个体表示为实值向量对 (x_i, η_i) , 其中, $i=1, 2, \dots, \mu$, x_i 称为目标变量, η_i 称为策略参数。

(2) 对于 $i=1, 2, \dots, \mu$, 评价每个个体 (x_i, η_i) 的目标函数值

基金项目:山东省自然科学基金项目(编号:Y2003G01)

作者简介:柯晶,男,博士,副教授,主要研究方向为人工智能与进化计算、系统建模与优化、数字信号处理。姜静,男,讲师,硕士,主要研究方向为自动测试技术与智能仪器。李歧强,博士,教授,博士生导师,主要研究方向为复杂系统优化、人工神经网络、进化计算。

$f(x_i)$ 。

(3) 对于 $i=1, 2, \dots, \mu$, 每个父代个体 (x_i, η_i) 平均产生 $\frac{\lambda}{\mu}$ 个子代个体, 即总共产生 λ 个子代个体:

$$\eta'_k(j) = \eta_i(j) \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_j(0, 1)) \quad (1)$$

$$x'_k(j) = x_i(j) + \eta'_k(j) \cdot N_j(0, 1) \quad (2)$$

其中: $j=1, 2, \dots, n, k=1, 2, \dots, \lambda; \eta_i(j), \eta'_k(j), x_i(j)$ 和 $x'_k(j)$ 分别表示向量 η_i, η'_k, x_i 和 x'_k 的第 j 个元素; $N(0, 1)$ 表示一个服从期望为 0、标准差为 1 的一维 Gaussian 分布的随机变量的实现, $N_j(0, 1)$ 表示对于下标 j 的每个值随机变量都要重新抽样; 学习率 τ' 和 τ 分别设为 $(\sqrt{2\sqrt{n}})^{-1}$ 和 $(\sqrt{2n})^{-1}$ 。

(4) 对于 $k=1, 2, \dots, \lambda$, 评价每个子代个体 (x_k, η_k) 的目标函数值 $f(x_k)$ 。

(5) 根据目标函数值在 λ 个子代个体中选择 μ 个最佳个体作为下一代进化的父代。

(6) $g=g+1$, 如已经满足终止准则, 例如整个群体已经发现的最优目标函数值小于某一阈值或迭代次数已达到最大允许进化代数, 则停止迭代; 否则转到步骤(3)。

需要指出的是, 上述步骤(3)这一特别的变异机制使得进化策略能够进化本身的策略参数, 在搜索过程中充分利用了合适的内在模式和好的目标函数值之间的一个隐含关联, 由此导致的根据目标函数值曲面的拓扑特性而使策略参数不断进化的机制, 被称为策略参数自适应机制^[2]。

2.2 快速进化策略

受快速模拟退火算法的启发, Yao 等用满足 Cauchy 分布的 Cauchy 变异算子代替通常采用的 Gaussian 变异算子, 提出了快速进化策略(Fast Evolution Strategies, FES)^[3]和快速进化规划^[4]。

Cauchy 分布的概率密度函数为^[5]:

$$f_\theta(x) = \frac{1}{\pi} \frac{\theta}{x^2 + \theta^2}, -\infty < x < \infty$$

其中 $\theta > 0$ 为比例参数, $\theta=1$ 时称为标准 Cauchy 分布。相应的分布函数为:

$$F_\theta(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\theta}\right)$$

Cauchy 分布的特点是期望不存在, 方差无限大, 比 Gaussian 分布更有可能产生较大的变异^[5]。快速进化策略的基本思想就是修改公式(2)为^[3]:

$$x'_k(j) = x_i(j) + \eta'_k(j) \cdot \xi_j \quad (3)$$

其中 ξ_j 表示服从 Cauchy 分布的随机变量的实现且对于下标 j 的每个值随机变量要重新抽样。

研究表明, 快速进化策略在求解多极值复杂问题时取得了很好的结果, 但对于局部极值较少的问题来说, 快速进化策略的性能与采用 Gaussian 变异算子的传统进化策略差别不大^[3]。

3 改进进化策略

进化策略是一种有效的函数优化方法, 但在实际应用时存在寻优速度较慢和易陷入局部极值点的缺点。为此该文提出一种改进进化策略, 其基本思想可以概括为: (1) 基于个体排序的随机自适应 Gaussian-Cauchy 混合变异; (2) 采用重组算子和策略参数下界。

由于 Cauchy 变异算子产生大变异的概率较高, 可以减小

陷入局部极值点的危险, 而 Gaussian 变异算子则具有良好的局部搜索能力, 可以加快局部收敛速度, 因而一些研究人员提出了平均变异、混合变异等改进方法, 尝试将 Cauchy 变异算子和 Gaussian 变异算子的优点结合起来^[2,4,5]。一般来说, 群体中目标函数值比较小的个体可能位于最优点附近, 因而应该以局部搜索为主, 而目标函数值比较大的个体则应加大变异力度, 在更大空间范围内进行探索, 以尽快逃离不利区域。借鉴列队竞争算法^[6]的基本思想, 在改进进化策略中采用一种基于个体排序的随机自适应 Gaussian-Cauchy 混合变异方式, 其基本思想是: 按照目标函数值对父代个体进行非降序排序, 根据此排序为每个父代个体分别指定 Gaussian 变异和 Cauchy 变异选择概率, 使得排名越靠前的个体 Gaussian 变异的概率越大而 Cauchy 变异的概率越小, 排名越靠后的个体 Cauchy 变异的概率越大而 Gaussian 变异的概率越小, 从而将 Gaussian 变异的良好局部搜索能力和 Cauchy 变异的大范围探索能力有效结合起来, 达到全局探索和局部探索之间的一种动态平衡。

尽管进化策略使用变异算子作为主要搜索算子^[3], 但适当使用重组算子可以提高进化策略的性能, 为此在改进进化策略中引入策略参数的中间重组^[2], 即修改公式(1)为:

$$\eta'_k(j) = \text{rec}(\eta_i(j), \eta_{m_j}(j)) \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_j(0, 1)) \quad (4)$$

其中重组算子 $\text{rec}(\cdot)$ 定义为:

$$\text{rec}(\eta_i(j), \eta_{m_j}(j)) = \beta_j \cdot \eta_i(j) + (1 - \beta_j) \cdot \eta_{m_j}(j)$$

其中: 随机整数 $m_j \in \{1, 2, \dots, \mu\}$ 且 $m_j \neq i$, β_j 为 $[0, 1]$ 区间内均匀分布的随机数。对于下标 j 的每个值 m_j 和 β_j 都要重新抽样。

在传统进化策略中, 随着进化代数的增加, 策略参数逐渐趋向零, 最终导致进化基本停止^[7], 为此在公式(4)的基础上对策略参数的最小值进行限制:

$$\eta'_k(j) = \max\{\zeta, \eta'_k(j)\} \quad (5)$$

其中 ζ 为预先设定的策略参数下界^[7]。

综上, 改进进化策略就是将基本 (μ, λ) -ES 中的步骤(3)用如下步骤代替:

按照目标函数值对 μ 个父代个体进行非降序排序, 排序后序号越小的父代个体对应目标函数值也越小。对 $i=1, 2, \dots, \mu$, 每个父代个体 (x_i, η_i) 根据排序结果采用随机自适应混合变异方式平均产生 $\frac{\lambda}{\mu}$ 个子代个体。具体方法是: 计算 $p_i = \frac{i-1}{\mu-1}$, 如果 $\text{rand} < p_i$, 按式(4)、(5)和(3)生成子代个体, 否则按式(4)、(5)和(2)生成子代个体, 其中 rand 为 $[0, 1]$ 区间上均匀分布的随机数。按此方式总共产生 λ 个子代个体。

参考进化策略的一般命名习惯, 将上述改进进化策略标记为 (μ, λ) -MES。

4 改进进化策略用于神经网络训练

作为改进进化策略的应用, 下面考虑多层前向神经网络的训练问题^[8]。

进化策略中目标变量的元素定义为多层前向神经网络的所有待确定参数, 即群体中每个个体就表示神经网络的一组参数, 利用改进进化策略搜索最优网络参数, 使如下准则函数达到最小:

$$J = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^T |d_j^i - y_j^i| \quad (6)$$

其中 N 为训练集的样本数, d_j^i 和 y_j^i 分别为对应第 i 个样本时网络第 j 个输出节点的期望输出值和实际输出值, T 为网络输出节点数。

需要指出的是, 上述准则函数是典型的不可微函数, 不能直接采用传统的基于梯度信息的优化方法求解, 但对于进化策略等进化算法来说, 非线性不可微函数的优化正是其优势所在。

Iris 数据集是测试分类器性能的基准问题之一^[9]。Iris 数据集包含 Setosa、Versicolor 和 Virginica 三种 Iris 花的 150 组测量数据, 每种 Iris 花有 50 组数据, 每组数据包含 Iris 花的四种属性: 萼片长度、萼片宽度、花瓣长度和花瓣宽度。

用三层前向神经网络对 Iris 数据集进行分类: 输入层有 4 个节点, 每个节点代表 Iris 花的一个属性; 输出层有 3 个节点, 每个节点代表一类 Iris 花, 即对应 Setosa、Versicolor 和 Virginica 这三种 Iris 花, 网络的期望输出分别为 (1, 0, 0)、(0, 1, 0) 和 (0, 0, 1); 隐含层的节点数目选为 10。用改进进化策略确定网络连接权重 (神经元阈值作为一类特殊权重处理), 每个个体表示神经网络的一组连接权重, 所以共有 $4 \times 10 + 10 \times 3 + 3 = 83$ 个参数。隐含层节点和输出层节点的激发函数采用如下 Sigmoid 函数:

$$f(u) = \frac{1}{1 + e^{-\frac{u}{u_0}}} \quad (7)$$

实验采用 (30, 100)-MES, 最大进化代数 300。策略参数随机初始化范围设为 [1, 10], 策略参数下界 $\zeta = 10^{-5}$, 最大限定为 10。网络连接权重随机初始化范围设为。实验中直接使用原始 Iris 数据, 不对数据进行归一化处理。

网络训练结束对样本 i 进行测试时, 对网络输出进行如下变换:

$$z_j^i = \begin{cases} 1, & y_j^i = y_m^i \\ 0, & y_j^i < y_m^i \end{cases} \quad j=1, 2, 3 \quad (8)$$

其中 $y_m^i = \max\{y_1^i, y_2^i, y_3^i\}$ 。在此基础上, 如果

$$\sum_{j=1}^3 |d_j^i - z_j^i| = 0 \quad (9)$$

则认为对此样本分类正确, 否则就认为分类错误。

首先考虑使用所有 150 个样本对网络进行训练, 然后再用这 150 个样本对网络进行测试, 即训练集与测试集相同。表 1 为 MES 方法 40 次随机仿真实验的具体结果。由表中数据可以得出 MES 方法的平均正确分类数为 148.7。

粒子群优化 (Particle Swarm Optimization, PSO)^[10]是近年来广受关注的一种新型进化算法。文献[9]采用 PSO 方法对上述网络进行训练。为便于参考, 表 2 列出了文献[9]的 PSO 方法对所有 150 个样本 (未归一化数据) 的分类结果。对照表 1 和 2 可知, 应用 MES 训练多层前向神经网络是十分有效的。

表 1 MES 方法对全部 150 个样本的分类结果

正确分类数	149	148
达到此正确分类数的运行次数	28	12

下面考虑网络的泛化能力。将 150 组数据分为训练集和测试集, 每个集合 75 组数据。利用训练集训练神经网络, 用测试集对训练好的网络进行测试。MES 的参数设置同上, 表 3 第 1 行数据即为 MES 方法 40 次随机仿真实验的统计结果。由表中

数据可知网络的泛化能力是比较好的。

表 2 PSO 方法对全部 150 个样本的分类结果

正确分类数	149	148	147	146
达到此正确分类数的运行次数	11	16	6	3
正确分类数	145	144	100	99
达到此正确分类数的运行次数	1	1	1	1

为了进一步验证 MES 方法的有效性, 分别应用 (30, 100)-ES、(30, 100)-FES、(30, 100)-MMES 和 (30, 100)-HMES 等 4 种进化策略对上述网络进行训练, 其中 MMES 表示采用平均变异算子^[12]的进化策略, HMES 表示混合变异进化策略^[9]。考虑到适当设置策略参数下界可以在很大程度上提高进化策略的寻优能力^[9], 在这 4 种进化策略中引入和 MES 相同的策略参数下界。为了便于比较, 每种方法都进行 40 次随机仿真实验, 实验设置与应用 MES 时的完全相同。表 3 列出了这 4 种进化策略方法的实验统计结果。表 4 为相应的双边 t 检验结果 (显著性水平), 其中 “+” 表示两种方法的均值有显著差异, “-” 表示无显著差异^[10], 空白表示同一方法不比较。由表 3 和 4 可知 MES 方法的性能优于其它 4 种进化策略方法。

表 3 不同方法对 75 个测试样本的分类结果

方法	正确分类数	
	均值	标准差
MES	70.750	1.354 006
ES	57.825	11.851 404
FES	68.800	4.052 223
MMES	68.400	4.180 541
HMES	68.400	4.198 901

表 4 双边 t 检验结果

	MES	ES	FES	MMES	HMES
MES		+	+	+	+
ES	+		+	+	+
FES	+	+		-	-
MMES	+	+	-		-
HMES	+	+	-	-	

5 结论

作为最早引入自适应机制的进化算法之一, 进化策略非常适合于非线性、不可微和多极值复杂函数优化问题。提出了一种改进进化策略并用于多层前向神经网络训练, 数值仿真结果显示了该方法的有效性。(收稿日期: 2005 年 6 月)

参考文献

1. Eiben A E, Smith J E. Introduction to evolutionary computing[M]. Berlin: Springer, 2003
2. 李敏强, 寇纪淦等. 遗传算法的基本理论与应用[M]. 北京: 科学出版社, 2002
3. Yao X, Liu Y. Fast evolution strategies[J]. Control and Cybernetics, 1997; 26(3): 467~496
4. Yao X, Liu Y, Lin G M. Evolutionary programming made faster[J]. IEEE Trans Evolutionary Computation, 1999; 3(2): 82~102
5. 贾立, 俞金寿. 基于自适应混合变异进化策略的神经模糊系统及应用研究[J]. 系统仿真学报, 2001; 13: 122~125
6. 鄢烈祥. 列队竞争算法解组合优化问题[J]. 湖北工学院学报, 2000; 15(2): 1~4

(下转 141 页)

对应了一些具体攻击报警。

在 CBRIDRA 中将案例检索分为两级匹配方法,先是对攻击报警序列的细粒度匹配,然后是对攻击模式的粗粒度匹配。匹配过程概略为:

步骤 1 将新的攻击序列与案例库中各案例的攻击事件进行匹配,并设定一个阈值。如果两者的相同事件个数大于阈值,则认为此案例与新攻击序列相似,输出检索案例,否则进行下一步。

步骤 2 确定新的攻击序列中各个攻击分属的攻击类型,找出案例中包含这些攻击分类的攻击模式,检索出相似程度最高的攻击模式,输出检索案例,否则进行下一步。

步骤 3 等待收集更多新的报警信息,重新进行检索。

步骤 4 等待时间超过一定值时,由专家结合知识库进行审查,人工进行案例检索或确定其为新的案例。

4.5 攻击案例的管理

经过攻击案例检索后,要将新攻击事件的有关信息组织成新的案例添加到案例库各组成表中去,并建立关键字索引字段。录入时各项内容应尽量完整准确,如果信息不完整,应手工添加,完整反映案例各方面特征。在系统运行一段时间后,由于实际报警的增多以及对攻击过程认识的深入,要对案例库进行整理,删除冗余信息,如合并相似攻击实例,提炼特征信息,如攻击源 IP、目的 IP、攻击端口 port 的概化,使各表结构更加精炼概括,既体现出攻击案例明显特征,又方便使用,防止案例库的过度膨胀,影响检索效率。

4.6 专家知识系统

专家知识系统作为系统辅助决策及安全管理的重要部分,充分运用已获得的知识规则和专家经验,利用人机交互技术,完成各功能模块中不能或难于自动完成的决策工作。它主要实现以下三个部分工作:一是案例的人工匹配检索。当遇到新的攻击事件经过两级匹配仍无法确定案例时,就需要管理员进行人工匹配。二是案例库和基础知识库的扩充与管理。随着对黑客攻击的研究,已取得的知识经验可以直接加以利用,录入到

攻击案例库中。同时研究中也发现一些攻击的确定规则,这可以作为规则推理使用,增强推理判断的确定性。三是安全防卫管理决策的制定。除了利用攻击案例中的解答信息外,很大程度还要依靠系统管理员和安全分析专家依据攻击事件和系统状况利用经验公式和知识进行分析,做出例如关闭某项服务、关闭端口等防卫响应。

5 结论

针对基于规则和模型的入侵检测专家系统中提取知识和总结规则存在的困难,新颖地将基于案例推理这种方法引入入侵检测领域。设计了 CBRIDRA 模型组成框架及各功能模块,并对其中主要关键技术提出了解决思路。该模型利用攻击知识经验直接推理入侵事件,确定攻击事件特征,产生防卫响应,实现了入侵检测的安全防卫目的。

原型系统主要基于 windows 平台,用 Delphi7.0 和 SQL Server2000 开发,已经在国家“863”课题“集成化网络安全防卫系统 Net-Keeper”中得以初步实现。实践表明,设计的模型是切实可行的。(收稿日期:2005 年 7 月)

参考文献

- 1.Rebecca Gureley Bace 著,陈明奇,吴秋新等译.入侵检测[M].北京:人民邮电出版社,2001-06
- 2.M Esmaili,B Balachandran,R Safavi-Naini et al.Case-Based Reasoning for Intrusion Detection[C].In:Proceeding of the 12th Annual Computer Security Applications Conference,1996:214-222
- 3.刘大有,赵宇霆,艾景军.基于事例的推理系统[C].见:陆汝钤主编.世纪之交的知识工程与知识科学,北京:清华大学出版社,2001:313-338
- 4.郭艳红,邓贵仕.基于事例的推理(CBR)研究综述[J].计算机工程与应用,2004;40(21):1-5
- 5.A Aamodi, E Plaza.Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches[C].In: AICom- Artificial Intelligence Communications, IOS Press, 1994; 7:39-59
- 9.Kennedy J, Eberhart R C, Shi Y. Swarm intelligence[M]. San Francisco: Morgan Kaufmann Publishers, 2001
- 10.盛骤, 谢式千, 潘承毅. 概率论与数理统计[M]. 第 2 版. 北京: 高等教育出版社, 1989
1. M Ward, K H Bennett. Recursion Removal/Introduction by Formal Transformation: an Aid to Program Development and Program Comprehension[J]. Computer Journal, 1999; 42(8): 650-673
2. R S Bird. Notes on recursion elimination[J]. Communications of the ACM, 1977; 20(6): 434-439
3. Yanhong A Liu, Scott D Stoller. From recursion to iteration: what are the optimizations[C]. In: Proceedings of the ACM SIGPLAN 2000 Workshop on Partial Evaluation and Semantics-Based Program Manipulation. New York: ACM Press, 2000: 73-82
4. Robert Kruse, C L Tondo, Bruce Leung. Data Structures & Program Design in C[M]. Prentice Hall, 1997
5. 朱振元, 朱承. 递归算法的非递归实现[J]. 小型微型计算机系统, 2003; 24(3): 567-570
6. 孟林, 李忠. 递归算法的非递归化研究[J]. 计算机科学, 2001; 28(8): 96-98

(上接 70 页)

7. Ohkura K, Matsumura Y, Ueda K. Robust evolution strategies[J]. Applied Intelligence, 2001; 15(3): 153-169
8. 杨建刚. 人工神经网络实用教程[M]. 杭州: 浙江大学出版社, 2001

(上接 75 页)

点付出的代价是转化后的程序执行效率往往不能达到最佳。本文下一步的工作是研究利用新方法实现转换自动化的可能性及提高性能的可能性。(收稿日期:2005 年 7 月)

参考文献

1. M Ward, K H Bennett. Recursion Removal/Introduction by Formal Transformation: an Aid to Program Development and Program Comprehension[J]. Computer Journal, 1999; 42(8): 650-673
2. R S Bird. Notes on recursion elimination[J]. Communications of the