

一种改进的 GEP 神经网络演化方法及其应用^①

An Improved Method for Evolving GEP Neural Network

齐 星 秦绪佳 李 曲 (浙江工业大学 软件学院 浙江 杭州 310023)

摘 要: 标准 GEP 方法作为一种新型的演化计算方法在回归和分类方面已经取得了很好的效果。本文在简要介绍采用 GEP 演化神经网络的基础上,指出了该方法在二次及以上建模问题中所存在的缺陷,提出并实现了一种改进的 GEP 神经网络演化方法(IGEPNN),并将其应用到回归和分类问题中。实验证明该方法在回归和分类问题中具有良好的学习效果。

关键词: 基因表达式编程 神经网络 回归 分类

1 GEP神经网络的基本方法

1.1 GEP概述

2001 年 12 月,葡萄牙的 Candida Ferreira 在遗传算法(Genetic Algorithms, GAs)和遗传编程(Genetic Programming, GP)的基础上提出了基因表达式编程(Gene Expression Programming, GEP)的概念^[1], GEP 同时结合了前两者的优点,为解决复杂的建模和优化问题提供了巨大潜力。目前国内外进行 GEP 的研究逐渐升温,唐常杰^[2]和 Chi Zhou^[3]等人对 GEP 进行了一些较为前沿和深入的研究。

GEP 同传统 GP 和 GA 算法比较,最大的特点就是个体的基因型表示形式。GEP 的基因型个体由定长的头部和尾部组成,头部元素来自函数集和终结符集 2 个集合,尾部元素则全部来自终结符集,并且头部长度 h 和尾部长度 t 须满足关系:

$$t = h \times (n - 1) + 1 \tag{1}$$

其中, n 是所需变量数最多的函数的参数个数,这一设计使得基于基因型个体的遗传操作都具有很好的合法性。

GEP 的个体表现型被称为表达式树(Expression Tree)。表达式树是通过顺序扫描基因型个体的字符,按照层次顺序构成。例如,若定义函数集 $F = \{+, -, *, /, Q\}$ (依次为加、减、乘、除和开方函数),终结符集 $T = \{a, b, c, d\}$, 则 $n = 2$, 取头部长度 $h = 8$,

由头尾部关系式(1)可得尾部长度 $t = 9$ 。假定有基因型个体: $Q-+/ab*-bccdabadc$, 它可以转化为如图 1 所示的表达式树结构(未被编码到表达式树中的字符“ $abadc$ ”则成为非编码区域)。

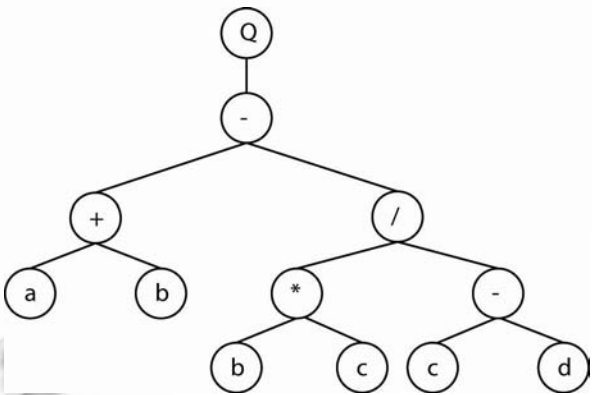


图 1 ET 树结构

1.2 GEP神经网络

从结构上说,一个神经网络具有 3 种不同类型的单元:输入、隐藏和输出单元。输入单元上提供一种激活模式,该模式以向前传播的方式从输入单元经一层或多层的隐藏单元到达输出单元。这个激活模式在从一个单元进入其它单元时需要与其传播方向上的连接权值相乘。然后,所有的输入激活值相加,只有当输入结果大于该单元的阈值时这个单元才被激活。

^① 基金项目:国家自然科学基金项目(60673063);国家 863 计划项目(2007AA12Z141);浙江省自然科学基金项目(Y107551);浙江省教育厅项目(20060815)
收稿时间:2008-09-23

Ferreira 在 GEP 的基础上模拟神经网络结构, 创建了 GEP 神经网络(GEP neural network, GEPNN)^[4]。GEP 神经网络是 GEP 与人工神经网络的结合, 它是利用 GEP 中个体的进化来代替人工神经网络的学习过程。为此, 我们必须修改 GEP 的染色体以使其能够将一个完整的神经网络, 包括其结构、权值和阈值, 完全编码在一个线性染色体中。

在 GEP 神经网络中, 染色体结构除了头部和尾部以外, 还含有两个附加域 D_w 和 D_t , 分别用来对权值和阈值进行编码。在 GEP 神经网络编码中, D_w 出现在染色体结构的尾部之后, 其长度 $d_w = h * n$ 。 D_t 出现在 D_w 域之后, 其长度 $d_t = h$, 这两个域分别由权值和阈值集合中元素的索引构成。

考虑图 2 用传统方法表示的神经网络, 它由两个输入单元(i_1 和 i_2), 两个隐藏单元(h_1 和 h_2)和一个输出单元(o_1)构成(为了简单, 阈值均等于 1, 并省略)。它还可以用一个传统的树形结构来表示, 如图 3 所示, 其中 **a**, **b** 分别表示两个输入 i_1 和 i_2 , “**D**” 表示具有两个连通分量的函数。该函数用参数乘以相应的权值, 并对所有的输入激活值求和减去阈值以得到向前的输出。我们可以将这个神经网络-树线性化成如下形式:

0123456789012
DDDabab123456

其中粗体结构对权值进行编码。该编码方式在解码构成树结构时, 我们约定相应的 D_w 值从染色体中自后向前读取, 按照从下到上, 从右到左的顺序出现在树的分支上。

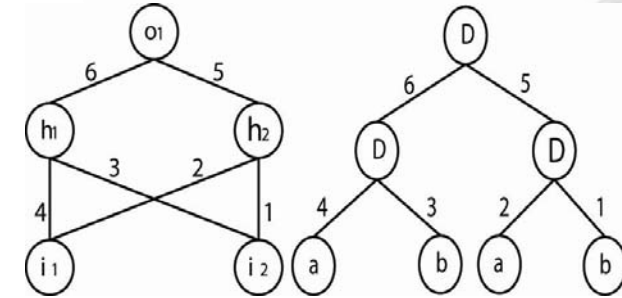


图 2 传统神经网络结构

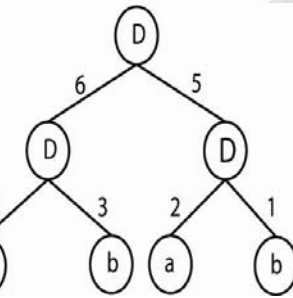


图 3 传统树形结构

2 改进的GEP神经网络演化方法

GEP 神经网络已经能够解决异或问题, 另外还能用来解决 6-位多路由器问题等。我们分析 GEP 神经网络的表达式树结构及其解码方式可知: 对终结符而言, 它的输出值就是终结符本身; 对于非终结符而言,

它的输出值是每个子结点值与对应权值相乘的累加和与阈值的差, 而每个子结点的输出为关于终结点的线性表达式, 这样解码表达式后就不可能产生一个二次或二次以上的函数表达式。因此, 无论染色体如何进化, 它所产生的表达式都只能是一次函数, 因而该 GEP 神经网络结构不能解决二次及以上的建模问题。

传统 GEP 神经网络的上述缺陷, 极大地限制了它的应用。要想更好地使用 GEP 神经网络解决问题, 必须对其进行改进。对比 GEP 与 GEP 神经网络结构不难发现, 导致 GEP 神经网络存在这个缺陷是因为 GEP 神经网络中使用的函数符, 并没有实际的运算意义, 而仅仅是在所带连通分量上的不同, 最后运算方式都是每个连通分量与相应权值相乘的累加和减去阈值。而 GEP 中使用的函数运算符都有本身的运算意义, 比如乘号运算符 “*”, 就是表示它的两个连通分量相乘, 这些函数运算符的存在为高次表达式的产生提供了可能。由此, 如果我们在 GEP 神经网络中使用这些具有实际运算意义的函数运算符, 而同时在 GEP 编码基础上同时加上权值和阈值信息, 这样就组成了一种改进的 GEP 神经网络演化方法 (IGEPNN), 而且这种 IGEPNN 演化方法具备产生高次函数表达式的能力。

与传统 GEP 神经网络结构相同, 我们将这种 IGEPNN 结构编码在一个头/尾域和权、阈值附加域的结构中, 只是函数符有所不同, 这样带来了解码方式的不同。例如对如下一个染色体: *--+aaaa**469818617**(其中黑体部分前 6 位表示权值域, 后 3 位表示阈值域), 对应表达式树为如图 4 所示。如果对应权值数组: $W=\{-1.553, 1.612, -1.606, -0.487, 1.475, -0.253, -1.91, 1.427, -0.103, -1.375\}$ 和阈值数组 $T=\{-1.14, 1.125, -1.173, -0.74, 0.393, 1.135, -0.625, 1.523, -0.029, -1.634\}$, 则表达式树可进一步表示为如图 5 所示, 它所对应的输出表达式经简化后为: $y=5.4a^2-0.7a-5.5$ 。理论上, 在基因长度允许的范围之内, 该表达式可以拟合任意复杂程度的高阶表达式, 从而跨越了文献[4]中所提出的神经网络只能解决线性函数逼近的问题。

3 IGEPNN求解问题

下面通过实验来验证 IGEPNN 在回归和分类问题中的效果。实验采用的平台是新西兰怀卡托大学开发的开源数据挖掘和知识分析平台 WEKA^[5]。我们在原

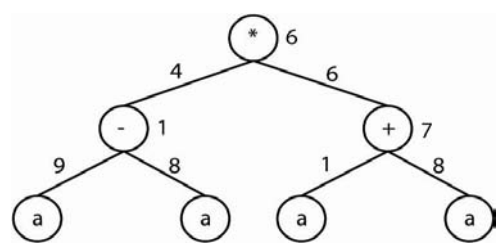


图 4 表达式树结构

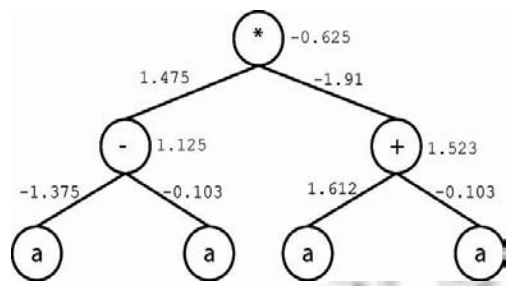


图 5 标注权、阈值的表达式树结构

来平台之上进行扩展，实现了标准 GEP 算法和 IGEPPN 算法，下面的实验结果均是在此扩展的 WEKA 平台上运行得出。实验中 GEP 各基本参数设置如表 1 所示：

表 1 实验中的 GEP 参数

运行次数	5	IS 转座概率	0.1
进化代数	3000	RIS 转座概率	0.1
种群大小	50	基因转座概率	0.1
函数集	+-* /	单点重组概率	0.3
头部长度	8	两点重组概率	0.3
基因个数	1	基因重组概率	0.1
变异概率	0.044	附加域转座概率	0.1

3.1 回归问题

在回归问题中，我们采用的目标函数是浮点常数二次函数 $y = 2.718a^2 + 3.1416a$ 。很明显，我们可以选取终结符集 $T = \{a\}$ 。实验采用的适应度函数为：

$$f_i = \sum_{j=1}^G (M - |C_{(i,j)} - T_j|)$$
 (2)

其中 M 是选择范围， $C_{(i,j)}$ 是染色体个体 i 对于适应度样本(来自集合中)的返回值，而 T_j 是适应度样本的目标值。因此，对于一个完美适应的情况， $C_{(i,j)} = T_j$ 且

$f_i = f_{\max} = G \cdot M$ 。实验从区间[-10, 10]选取了 10 个数据，并采用 100 作为选择范围，因此，训练最大适应度 $f_{\max} = 1000$ 。为了使结果更加准确，我们采用了目前较流行的 k 折分层交叉确认法进行测试(测试中均取 k = 5)，则测试最大适应度 $f_{\max} = 800$ ，运行结果见表 2：

表 2 回归问题运行结果

运行次数		1	2	3	4	5
训练最大适应度		994.72	998.07	987.71	998.92	999.10
5-折交叉确认	1	796.62	774. 50	792.69	789.61	795.69
	2	783.50	795.91	793.98	771.59	786.52
	3	798.85	795.05	761.94	770.83	781.87
	4	788.96	794.00	787.35	792.52	789.86
	5	795.62	795.94	797.36	78520	769.52

从实验结果中我们可以看出，我们这种 IGEPPN 演化方法对函数表达式进行了一个比较好的拟合。该函数无论在训练还是在测试集上均取得了较好的逼近效果，证明这种神经网络演化方法具有很好的函数发现和学习能力。

3.2 分类问题

我们引用美国加利福尼亚大学机器学习知识库[6]中的标准数据对这种 IGEPPN 演化方法用于分类规则挖掘进行了实例测试，数据集属性的主要描述见表 3：

表 3 数据集属性

数据集	样本数	属性数	类别数
Balance Scale	625	4	3
Breast Cancer W	699	9	2
Glass	214	9	6
Iris	150	4	3
Labor	57	15	2
Pima Indian	768	8	2
Wine	178	13	3
Zoo	101	17	7

它给出了每个数据集的样本数、属性数和类别数。在 IGEPPN 中，分类规则仍然是用数学表达式的方式来表达，它所返回的值为浮点数，所以需要得到的浮点数结果转换成类别号。对于二分类问题，只需要一条 GEP 神经网络规则就可以实现分类，即当 GEP 神经

网络的输出值大于或等于 0 时，则属于其中一类，否则属于另一类。精确表示如下(其中 X 表示实例的属性向量):

If GEPNN_RULE(X)>0 Then X ∈ class A
Else X ∉ class A (3)

当遇到 n(n>2)分类问题时，须将其转换成 n 个二分类问题来解决，因此需要一个如下的规则群来解决 n 分类问题。

If GEPNN_RULEj(X)>0 Then X ∈ class j
Else X ∉ class j (j=1,2,...n) (4)

同样，由于演化计算方法本身存在随机性，为了更加客观地评价分类器的分类精度，实验采用了 k 折分层交叉确认法(测试中均取 k =5)。为了便于比较，同时测试了标准 GEP 算法在相同测试集上的分类结果，如表 4 所示。

表 4 GEP 和 IGEPNN 算法分类结果比较(%)

数据集	GEP						IGEPNN					
	1	2	3	4	5	平均	1	2	3	4	5	平均
Balance Scale	91.84	91.84	90.72	92	91.52	91.58	88	88.16	88.48	88	89.28	88.38
Breast Cancer W	94.86	95.99	95.57	96.14	95.71	95.65	95.71	95.71	96.42	95.57	96.14	95.91
Glass	60.76	59.81	61.22	58.89	61.68	60.47	62.15	57.01	57.01	56.07	57.48	57.94
Iris	96	94.67	94.67	97.33	95.33	95.59	94	94	93.33	93.33	94.67	93.87
Lablror	66.67	61.40	66.67	64.91	64.91	64.91	73.68	77.19	78.95	73.68	75.44	75.77
Pima Indian	70.57	69.92	70.18	69.27	68.36	69.66	71.35	74.22	72.27	71.74	72.01	72.32
Wine	89.89	89.33	89.89	91.01	93.82	90.79	90.45	90.45	87.64	89.33	88.76	89.33
Zoo	86.14	87.13	88.12	87.13	89.11	87.53	89.11	89.11	90.10	89.11	89.11	89.31

从实验结果可以看出，在 8 个测试数据集中，IGEPNN 方法在 Breast Cancer W, Labor, Pima Indian, Zoo 4 个数据集中比标准 GEP 方法精度高，这说明 IGEPNN 方法在分类规则挖掘方面的能力与 GEP 相当。值得一提的是 Labor 数据集上，IGEPNN 方法比标准 GEP 方法分类精度提高超过 10%，这说明 GEP 神经网络在求解某些问题时比标准 GEP 方法具有更好的效果。

然而，从进化效率的角度来说，根据上面我们的证明可以看到，由于 GEP 神经网络的权值和阈值在表达式树中出现在表达式树的分支上，更进一步来说，

权值和阈值在基因内部出现在附加域中，从而更少地占用头部和尾部的基因位，让出更多的空间给函数符和属性变量，使得 GEP 基因本身的表现和搜索能力更强。

4 结论与展望

GEP 是一种全新的自动程序设计方法，它同时继承了 GA 和 GP 的优点，具有很强的进化能力，但是传统 GEP 神经网络在解决二次及以上回归和分类问题上却存在缺陷。本文对传统 GEP 神经网络进行了改进并提出了一种改进的 GEP 神经网络演化方法，通过实验对其在回归和分类问题上进行了验证。

通过实验，我们可以看出，这种 IGEPNN 演化方法能够很好地解决传统 GEP 神经网络所无法解决的二次及二次以上回归和分类问题，而且在分类问题上其精度与标准 GEP 方法相比，在不同数据集上的结果相当。这证明这种 IGEPNN 演化方法在回归和分类问题是可行的，今后的工作主要是如何进一步提高分类器的分类精度。另外，由于我们这种网络结构在进化过程中涉及大量浮点运算，所以如何有效地降低算法的运算时间也是需要进一步研究的问题。

参考文献

1 Ferreira C. Gene expression programming: a new adaptive algorithm for solving problems. Complex Systems, 2001,13(2):87 – 129.

2 彭京,唐常杰,李川,等.M-GEP:基于多层染色体基因表达式编程的遗传进化算法.计算机学报, 2005, 28(9):1459 – 1466.

3 Chi Z, Xiao WM, Tirpak TM, Nelson PC. Evolving accurate and compact classification rules with gene expression programming. IEEE Transactions on Evolutionary Computation, 2003,2(6):519 – 531.

4 Ferreira C. Designing Neural Networks Using Gene Expression Programming. The 9th Online World Conference on Soft Computing in Industrial Applications, 2004.

5 Witten IH, Frank E. Data Mining: Practical machine learning tools and techniques. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

6 Murphy PM, Aha DW. UCI Repository of Machine Learning Database. <http://archive.ics.uci.edu/ml/dataset-ts.html>.

