

# RCGNN:一种基于实数编码的遗传神经网络预测方法

吴红<sup>1</sup> 吴值民<sup>2</sup>

(解放军理工大学理学院 南京 211101)<sup>1</sup> (解放军理工大学工程兵工程学院 南京 210007)<sup>2</sup>

**摘要** 将遗传算法与神经网络相结合,提出一种实数编码、自适应选择、算术交叉、高斯变异、爬山操作的改进遗传BP神经网络RCGNN,利用遗传算法对神经网络权值和阈值进行优化。以时间序列预测的实例进行编程计算表明,用遗传算法进行网络训练,其收敛速度快,最终总误差最小,预测准确率高。对算法中参数进行的相应研究表明,增加爬山操作次数能很好地提高网络训练的速度,同时使误差下降快;爬山操作越多,收敛速度越快,最终误差越小,但计算运行时间也会增加。

**关键词** 预测, BP神经网络, 遗传算法, 实数编码

## RCGNN: A Real Number Coding Genetic Neural Network Prediction Model

WU Hong<sup>1</sup> WU Zhi-min<sup>2</sup>

(Institute of Sciences, PLA Univ. of Sci. & Tech., Nanjing 210001, China)<sup>1</sup>

(Engineering Institute of Engineering Corps, PLA Univ. of Sci. & Tech., Nanjing 210007, China)<sup>2</sup>

**Abstract** In order to overcome the shortcomings of traditional error back propagation algorithm for updating the weights of the multi-layer forward neural networks, such as the low precision of the solutions, the slow search speed and easy convergence to the local minimum points, we propose an improved genetic neural network RCGNN, which combines genetic algorithm and neural network, contains real number coding, arithmetic crossover, Gauss mutation and climbing operation. The result of a illustrative sequence prediction example shows that, the method has a faster training speed, less total error and can predict more accurate. When adjusting the parameters in the algorithm, we discover that adding climbing operation times can effectively quicken the training speed of network and lessen the error diminished faster, more climbing operation can get quicker convergence speed and smaller final total error, but need more time to run this process.

**Keywords** Sequence prediction, Genetic algorithm, Neural network, Real number coding

人工神经网络已广泛应用在语言识别、自动控制、预测、评价等方面<sup>[1-3]</sup>,其准确性明显优于回归模型。但它也存在很多缺点,如学习收敛速度慢,通常需要经过几千次迭代甚至更多,容易陷入局部极小点而无法得到全局最优解。将神经网络与其它优化算法相结合是提高神经网络全局收敛能力和收敛速率的有效途径<sup>[4,5]</sup>。遗传算法(Genetic Algorithm, GA)<sup>[6,7]</sup>最早由Holland于1975年提出。该方法以一个随机染色体种群开始进化,适应度高的染色体被选择进行交叉和变异操作,产生的子代与父代不同,但从父代继承了某些遗传因素,当某一特定代产生或进化收敛时过程停止。它是一种高效和计算简单的优化算法,其内在并行性和较强的局部搜索可以快速有效在全局范围寻求最优点,完全可以应用于神经网络权值的训练和学习中,提高神经网络的学习效率和学习速度。本文提出一种改进的遗传神经网络模型RCGNN,通过遗传算法对BP神经网络的权值进行训练,并用一个时间序列预测实例来检验该预测模型的性能。

网络即可模拟任意复杂的非线性问题<sup>[7]</sup>。因而,本文采用三层的网络模式。由于预测模型通常是由多个输入值来预测多个输出值,其结构为:一个输入层,节点数为 $m$ ,一个隐层,节点数为 $l$ ,一个输出层,节点数为 $n$ ,图1给出了神经网络结构图。输入为 $X=(x_1, x_2, \dots, x_m)'$ ,输入层到隐层权值为 $W_1=(w_{1ij})_{m \times l}$ ,隐层阈值为 $B=(b_1, b_2, \dots, b_l)'$ ,隐层到输出层权值为 $W_2=(w_{2ij})_{l \times n}$ ,输出层阈值为 $S=(s_1, s_2, \dots, s_n)'$ ,输入为 $Y=(y_1, y_2, \dots, y_n)'$ 。

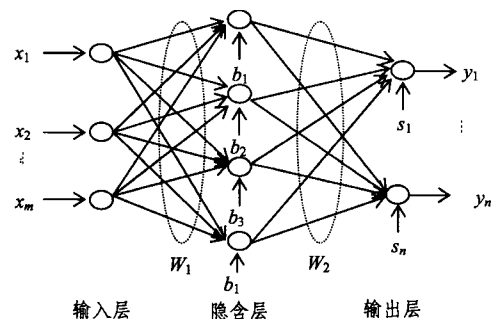


图1 神经网络结构图

神经元的激励函数各层可采用不同的函数,这里隐层和输出层采用相同的对数 Sigmoid 函数:

### 1 基本 BP 神经网络

对于复杂动力系统预测一般采用多层前向BP神经网络<sup>[7-9]</sup>,该网络有一个源节点输入层,一个或多个隐含层,一个输出层。Robert Hecht-Nielsen于1989年证明了用一个三层

吴红 讲师,主要研究方向为应用数学、神经网络、遗传算法。

$$f(v) = \frac{1}{(1 + \exp(-aX))} \quad (1)$$

## 2 RCGNN: 基于实数编码的遗传神经网络

BP神经网络常用的学习算法是梯度下降算法,这种方法收敛速度慢,易振荡,而且易陷入局部极值点。遗传算法是在整个解空间进行搜索,收敛速度快,而且易得到全局最优解<sup>[8,9]</sup>。

以下为我们提出的遗传神经网络模型设计:RCGNN用遗传算法对神经网络进行训练,能够很好地解决神经网络方法存在的问题。它的收敛性取决于染色体表示方式,初始群体,适应度函数,遗传算子的设计。

### 2.1 染色体表示

神经网络学习的目标是找到恰当的网络权值和阈值,使误差最小。解空间中的数据是遗传算法的表现型形式,从表型到基因型的映射称为编码。用一个个体中基因来表示神经网络所有权值和阈值,将神经网络权值矩阵和阈值矩阵变成一维矩阵,按  $W_1, B, W_2, S$  的次序将所有权值阈值连接在一起,形成遗传算法实数编码的染色体表示,如图2所示,编码长度为  $N = m \times l + l + l \times n + n$ 。

$w_{111}, \dots, w_{11l}, w_{211}, \dots, w_{21l}$	$b_1, \dots, b_l$	$w_{211}, \dots, w_{21n}, w_{221}, \dots, w_{22n}$	$s_1, \dots, s_n$
--	-------------------	--	-------------------

图2 染色体表示

### 2.2 产生初始种群

初始权值取为均匀分布在  $(-2.4/F, 2.4/F)$  的随机小数,其中  $F$  为所连单元的输入端个数。设初始种群为  $M$ ,按照公式(2)随机生成  $M$  个在  $(-2.4/F, 2.4/F)$  上均匀分布的一维矩阵。

$$\text{parent} = (\text{rand}(M, m \times l + l + l \times n + n) - 0.5) \times (2.4/F) \quad (2)$$

### 2.3 自适应选择

为了改善轮盘赌选择操作中开始阶段由于个体差异大,使选择概率差异过大和后期阶段由于个体之间差异小,选择概率差异过小的问题,可以按个体优劣次序设计出一种固定选择概率。设种群大小为  $N$ ,为了保持稳定有效的选择压力,对父代中的个体按由优到劣排序<sup>[5,6]</sup>,序号为  $1, 2, \dots, M$ ,并按序号赋予每个个体一个等级,序号为  $i$  的个体被选中的概率  $P_i$  被定义为:

$$P_i = \frac{2(M-i+1)}{M(M+1)} \quad (3)$$

其中  $N$  为种群大小,  $i$  为个体的排名等级。实现自适应选择的算法为:

```
p = 2 * (M - [1:M] + 1) ./ M / (M + 1); %选择概率计算
p = p';
parent = [parent, f];
parent = sortrows(parent, N + 1); %种群排序
f = parent(:, N + 1);
parent = parent(:, 1:N);
p = cumsum(p);
p = [0; p];
for i = 1:M
    r = rand; k = 1;
    while r > p(k)
        k = k + 1;
    end
```

```
son(i, :) = parent(k - 1, :);
```

```
end
```

### 2.4 适应度函数

在遗传算法中,以个体适应度的大小来确定该个体被遗传到下一代群体中的概率。对应适应值为该可行解与最优解距离,一般把目标函数值进行适当的变形,转换成适应度的值。对于神经网络,训练误差越小则个体越优,则先计算每个个体对应所有样本的绝对值误差和,其倒数作为适应度。由于激励函数 Sigmoid 函数值域为  $(0, 1)$ ,因此要先对样本数据进行相应映射处理,尽量使输入输出值均匀分布在  $0$  到  $1$  之间,考虑到激励函数不能取到值  $0$  或  $1$ ,所以这里将数据线性映射处理到区间  $[0.1, 0.9]$ 。设样本总体中某列元素为  $X$ ,某样本原始数据为  $x_k$ ,处理后数据为  $x_k'$ 。

$$x_k' = (x_k - \min(X)) / (\max(X) - \min(X)) \times 0.8 + 0.1 \quad (4)$$

计算神经网络输出值。对每个个体进行解码,计算出个体对应神经网络的各权值和阈值,按公式(4)计算出每个样本对应的输出值。

$$Y = f(f(X' \cdot W_1 + B') \cdot W_2 + S')' \quad (5)$$

计算绝对误差和  $E$ ,其中  $N_x$  为样本的个数。

$$E = \sum_{i=1}^{N_x} \sum_{j=1}^n |y_j - d_j| \quad (6)$$

个体适应度值为误差的倒数。算法为:

```
for i = 1:M
    f(i) = 0; %适应度值初始化为 0
    W1 = reshape(parent(i, 1:m * l), m, l);
    B = parent(i, m * l + 1:m * l + l);
    W2 = reshape(parent(i, m * l + l + 1:m * l + l + l * n), l, n);
    S = parent(i, m * l + l + l * n + 1:m * l + l + l * n + n);
    for j = 1:Nx
        Ix = x(j, :);
        f(i) = f(i) + abs(y(j) - f(Ix * W1 + B) * W2 + S);
    end
    f(i) = 1/f(i);
end
```

### 2.5 算术交叉算子

设选中的两父代个体  $x_1, x_2$ ,交叉后新生成个体为  $x_1', x_2'$ ,则

$$\begin{cases} x_1' = \alpha x_1 + (1-\alpha)x_2 \\ x_2' = (1-\alpha)x_1 + \alpha x_2 \end{cases} \quad (7)$$

其中  $\alpha$  为随机生成的数,满足  $0 \leq \alpha \leq 1$ ,设遗传算法中间代存入放入到矩阵  $\text{son}$  中,进行交叉的两个个体分别为  $X, Y$ ,算法如下:

```
marry = randperm(M); %两两个体随机配对
for i = 1:M/2
    r1 = rand;
    if r1 < pc
        r = rand;
        X = son(marry(i), :);
        Y = son(marry(M - i + 1), :);
        son(marry(i), :) = r * X + (1 - r) * Y;
        son(marry(M - i + 1), :) = (1 - r) * X + r * Y;
    end
end
```

### 2.6 高斯变异算子

对于个体  $x = (x_1, x_2, \dots, x_n)$ ,加入一个小的扰动,作为个

体的变异,设个体变异后变为  $x'=(x_1',x_2',\dots,x_n')$ 。随机生成变异基因位  $r$ 。则

$$x_i' = \begin{cases} x_i & i \neq r \\ x_i + \alpha & i = r \end{cases} \quad (8)$$

其中  $\alpha$  为与个体有相同维数的随机数矩阵,值较小。算法为:

```

for i=1:M
    r=rand;
    if r<pm
        son(i,:)=son(i,:)+randn(1,N)*0.2;
    end
end
end

```

## 2.7 爬山操作

当当代交叉、变异完成后,找出群体中的最优个体,通过领域搜索实施爬山操作,具体操作如下:

- (1)给最优个体一个小的扰动,做法同变异。
  - (2)判断新的个体是否更优,更优则替换原个体。
- 重复(1)(2),直到达到一次操作次数为止。

## 2.8 RCGNN 遗传神经网络算法流程图

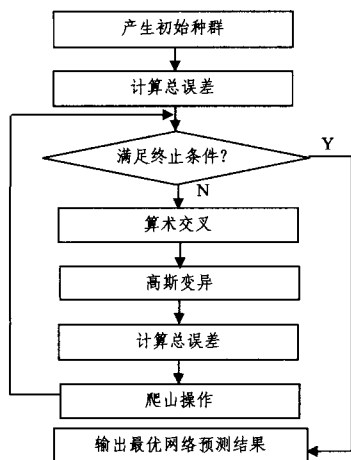


图3 RCGNN 遗传神经网络算法流程图

## 3 时间序列预测模型

时间序列<sup>[9]</sup>是指某种统计指标的数值,按照时间先后顺序排列起来的数列。时间序列预测是用历史的数据对未来进行预测,设某时间序列为  $\{x_n\}$ ,  $n=1,2,\dots,n_0$ 。用  $x_n$  和其前  $m$  个值预测  $x_n$  后第  $k$  个值,即用数学式描述为:

$$x_{n+k} = g(x_n, x_{n-1}, \dots, x_{n-m+1}) \quad (9)$$

用前  $m$  个数据  $x_n, x_{n-1}, \dots, x_{n-m+1}$  预测  $x_{n+1}$  的方法进行预测。

## 4 实例

引用文献<sup>[10]</sup>中的数据,某省 1980—1997 年的能源年消耗总量经数据标准化处理,得到时间序列  $\{x_n\} = \{0.7311, 0.7484, 0.7664, 0.7815, 0.7975, 0.8233, 0.8371, 0.8564, 0.8727, 0.8817, 0.8798, 0.8880, 0.9024, 0.8967, 0.9144, 0.9269, 0.9364, 0.9425\}$ , 以此对 1999 年的消耗总量进行预测。

这里使用基于 Windows 的 MATLAB 编程,采用 RCGNN 模型对时间序列进行预测,具体做法为用前四年情况预测后一年,取出 1998 年作为最终模型的检验不参加网络训练,则训练样品中共有 13 个样本,网络参数为:  $m=4, l=m \times 2 + 1, n=1$ 。遗传算法参数取为群体大小  $M=50$ ,取交叉概率  $P_c=0.9$ ,变异概率  $P_m=0.1$ ;爬山次数  $E=20$ ;终止进化代数  $G=200$  代。

仿真结果如表 1 所示。

表 1 仿真结果

样本	样	本	输	入	输出	期望
1	0.7311	0.7484	0.7664	0.7815	0.7975	0.7975
2	0.7484	0.7664	0.7815	0.7975	0.8199	0.8233
3	0.7664	0.7815	0.7975	0.8233	0.8440	0.8371
4	0.7815	0.7975	0.8233	0.8371	0.8564	0.8564
5	0.7975	0.8233	0.8371	0.8564	0.8692	0.8727
6	0.8233	0.8371	0.8564	0.8727	0.8792	0.8817
7	0.8371	0.8564	0.8727	0.8817	0.8868	0.8798
8	0.8564	0.8727	0.8817	0.8798	0.8909	0.8880
9	0.8727	0.8817	0.8798	0.8880	0.9012	0.9024
10	0.8817	0.8798	0.8880	0.9024	0.9121	0.8967
11	0.8798	0.8880	0.9024	0.8967	0.9108	0.9144
12	0.8880	0.9024	0.8967	0.9144	0.9270	0.9269
13	0.9024	0.8967	0.9144	0.9269	0.9363	0.9364
检验	0.8967	0.9144	0.9269	0.9364	0.9434	0.9425
预测	0.9144	0.9269	0.9364	0.9425	0.9502	

从表 1 看出,1998 年检验得到预测误差率为 0.001,网络输出值与期望值误差较小,用神经网络进行时间序列模型的预测准确率较高。为便于分析,以数据序号为横坐标,以网络输出数据和期望数据为纵坐标,绘图如图 4,线以及线上点为期望值,小圆圈为神经网络输出值,可以看到数据拟合得较好。

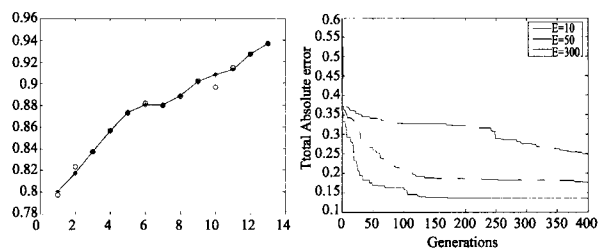


图 4 输出数据与期望值的比较

图 5 绝对误差

进行仿真计算时,对算法中参数进行调整发现,增加爬山操作的次数  $E$ ,可以有效提高预测准确率,如图 5,分别取爬山操作次数  $E=10, E=50, E=300$ ,随着  $E$  的增加,总误差明显减少,在遗传算法中加入爬山操作可以有效加快收敛速度,但达到收敛所花的时间也会大幅增加;所花时间分别为(单位为秒):272.89,1273,8816。

**结束语** 本文针对 BP 神经网络误差反馈训练方法收敛速度慢,易陷入局部极值点的缺陷,提出一种改进遗传神经网络模型 RCGNN,以一个时间序列预测的实例进行仿真计算,表明遗传算法进行网络训练收敛速度快,最终总误差最小,预测准确率高。对算法中参数进行的相应研究表明,增加爬山操作次数能很好地提高网络训练的速度,同时使误差下降快,爬山操作越多,收敛速度越快,最终误差越小,但计算运行时间也会增加。总之,遗传神经网络为解决现实中大量由于时间、技术、资金的局限影响,而无法得知影响因子或影响因子难以求解的预测问题提供了一个非常有效的方法。

## 参考文献

- [1] 高隽. 神经网络原理及仿真实例[M]. 北京:机械工业出版社,2003
- [2] 周开利,康耀红. 神经网络模型及其 MATLAB 仿真程序设计[M]. 北京:清华大学出版社,2005

(下转第 194 页)

### 3.3 算法实现

插入算法:

(1)新建一个 Node\_Info 结点;

(2)Node\_type=0 或 1;

IsModify=True;

ModifyTimestamp=当前时间;

CacheTimestamp= $-\infty$ ;

NodeValue=结点值;

(3)计算结点所在子树的度  $d_i$ ;

(4)把结点插入到结点列表 Node\_list;

删除算法:

(1)把结点加入到 Delete-Queue 队列中;

(2)在 SubTree-Queue<sub>i</sub> 删除对应的结点;

更新算法:

(1)for (Each Node in SubTree-queue)

(2){ 结点数据值更新;

(3)当前时间戳=更新时间;

(4)将其插入到 Node\_list 列表中;

(5) }

传输算法:

(1)传输队列清空

(2)for ( Each Node in Delete-queue)

(3){ 向移动主机发送删除消息;

(4)在 SubTree-queue 中删除已传输的结点;

(5) }

(6)for (Each Node in Node\_list)

(7){ 计算结点的度  $d_i$ ;

(8)计算优先级 Priority;

(9) }

(10)while( 没有收到停止消息 or 数据都已更新)

(11){ 从 Node\_list 中选取 Priority 最大的结点;

(12){ CacheTimeStamp=当前时间;

(13)将该元素加到传输队列尾部;

(14) }

(15) }

此算法建立了一种基于 XML 结构的移动数据库数据更新策略。与用于关系数据库增量视图更新的算法相比,XSIU 算法整体上采用树型结构的思想,并将增量视图分成若干个子树。该算法增加了对增量视图中需更新的各个子树结点度的计算,由于计算结点度的方法简单且易于实现,对算法的时间复杂度和空间复杂度增加很小,因此该算法在移动数据库环境中实用可行。另外,在数据的传输过程,采用按 Priority 级高的优先传输,这样该算法可以很好地动态适用网络带宽。

### 4 实验结果

通过表 1 给出的实验数据对提出的 XSIU 算法进行模拟试验分析,在模拟实验中使用了 IBM 的 XML 生成器<sup>[6]</sup>以及 xmlgen<sup>[7]</sup>生成实验的 XML 文档,并与以下两种情况进行比较:(1)既不引入数据元长度也不引入结点的度  $d_i$ ,即为 PIU 算法;(2)在 PIU 算法的基础上仅引入数据元长度。通过图 2

可以发现,在网络带宽较低时,XSIU 算法具有较高的数据新鲜度,明显优于情况(1)和(2)。但当网络带宽充足时,三者相差不大。该实验证明 XSIU 算法确实能够有效地提高移动环境下视图更新的性能。

表 1 实验数据

参数	说明	值
W	网络平均带宽	100k~800kbmps
T <sub>i</sub>	需要更新的子树	1~50 个
d <sub>i</sub>	结点的度	10~30
w <sub>j</sub>	结点的权值	0.15~0.3
l <sub>j</sub>	结点的长度大小	10~40k

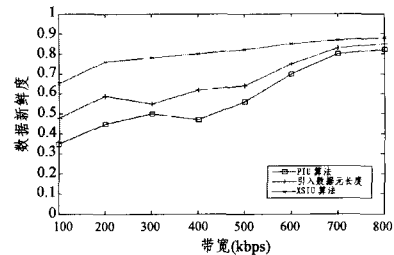


图 2 数据新鲜度比较

**结束语** 本文在基于移动关系数据库的增量视图更新算法 PIU 的基础上,提出了一种改进的基于 XML 结构的移动数据库视图增量更新算法 XSIU。在改进的算法中引用树型结构中结点的度和数据元的长度,对数据新鲜度的值进行修正,从而极大地提高了视图的数据新鲜度,使得用户能更方便地在移动环境中进行访问。

### 参考文献

- [1] Extensible Markup Language. <http://www.w3.org/XML>
- [2] 林怀忠,周波,等. 移动数据库的视图增量更新算法. 计算机集成制造系统-CIMS, 2004,10(3):312-315
- [3] Lauzac S W, Chrysanthis P K. Programming views for mobile database clients [A]// Proceedings of 9th International Workshop on Database and Expert Systems Applications[C]. Vienna, Austria: IEEE Computer Society Press, 1998:408-413
- [4] XIR: Cache Invalidation Strategy for XML Data in Mobile Environment// Proceedings of the 6th ACM International Workshop on Data Engineering for Wireless and Mobile Access MobiDE '07. 2007:79-82
- [5] Pacitti E, Simon E, Melo R. Improving data freshness in lazy master schemes[A]// Proceedings of 18th International Conference on Distributed Computing Systems [C]. Amsterdam, Netherlands: IEEE Computer Society Press, 1998:164-171
- [6] Diaz A I, Lovell D. IBM XML Generator. <http://www.alphaworks.ibm.com/tech/xmlgenerator>
- [7] Schmidt A, Waas F, Kersten M L, et al. Xmark: A benchmark for xml data management// Proceedings of VLDB. 2002:974-985

(上接第 180 页)

- [3] 赵瑞,申金山,郭院成,等. 神经网络在城市用水量预测中的应用研究[J]. 郑州工业大学学报,2000,21(1)
- [4] Sexton R S, Gupta J N D. Comparative evaluation of genetic algorithm and back propagation for training neural networks[J]. Information Sciences, 2000, 129:45-59
- [5] Seaton R S, Dorsey R E, Johnson J D. Optimization of neural networks, A comparative analysis of the genetic algorithm and simulated annealing[J]. Europ J of Operational Research, 1999, 114:589-601

- [6] 王小平,曹立明. 遗传算法实现与软件实现[M]. 西安:西安交通大学出版社,2002
- [7] 李敏强. 遗传算法的基本理论与应用[M]. 北京:科学出版社,2002
- [8] 卢厚清,吴值民,吴凤丽,等. 多随机因素下多品种库存优化的改进遗传算法[J]. 解放军理工大学学报,2006,7(2):166-169
- [9] 钟颖,汪秉文. 基于遗传算法的 BP 网络时间序列预测模型[J]. 系统工程与电子技术,2002,24(4):9-11
- [10] 欧阳洁. 市场预测与决策分析方法[M]. 广东:广东经济出版社,2004