

# 基于参数动态调整的动态模糊神经网络的 软件可靠性增长模型

刘 遼<sup>1,2</sup> 郭立红<sup>1</sup> 肖 辉<sup>1</sup> 王建军<sup>1</sup> 王改革<sup>1,2</sup>

(中国科学院长春光学精密机械与物理研究所 长春 130033)<sup>1</sup> (中国科学院研究生院 北京 100039)<sup>2</sup>

**摘要** 利用遗传算法对动态模糊神经网络的自身参数进行动态调整(GA-DFNN),并将其应用于软件可靠性增长模型(SRGM)的研究。在对动态模糊神经网络进行训练的过程中,用遗传算法求得动态模糊神经网络自身参数的优化解,根据得到的参数建立基于动态模糊神经网络的软件失效数据预测模型。利用3组软件缺陷数据,对用GA-DFNN建立的SRGM和模糊神经网络(FNN)以及BP神经网络(BPN)建立的SRGM的预测能力进行了比较,仿真结果证实,根据GA-DFNN建立的SRGM的短期预测能力稳定,短期预测误差小,且具有一定的通用性。

**关键词** 软件可靠性增长模型,动态模糊神经网络,遗传算法,短期预测

中图分类号 TP183 文献标识码 A

## Software Reliability Growth Model Based on Dynamic Fuzzy Neural Network with Parameters Dynamic Adjustment

LIU Luo<sup>1,2</sup> GUO Li-hong<sup>1</sup> XIAO Hui<sup>1</sup> WANG Jian-jun<sup>1</sup> WANG Gai-ge<sup>1,2</sup>

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun 130033, China)<sup>1</sup>

(Graduate University of the Chinese Academy of Sciences, Beijing 100039, China)<sup>2</sup>

**Abstract** The parameters of dynamic fuzzy neural network were dynamically adjusted by genetic algorithm (GA-DFNN), and GA-DFNN was used to study software reliability growth model (SRGM). The optimal solution of DFNN's parameters was resolved by genetic algorithm in the DFNN's training process, and according to the DFNN which has the optimal parameters, software failure data prediction model was established. According to 3 groups of software defects data, we compared the SGRM's predictive ability established by GA-DFNN with SGRM's predictive ability established by fuzzy neural network (FNN) and BP neural network (BPN). The simulation results confirm that the SRGM established by GA-DFNN has steady short period prediction, and its short period prediction error is small and it has some versatility.

**Keywords** Software reliability growth model, Dynamic fuzzy neural network, Genetic algorithm, Short period prediction

### 1 前言

软件可靠性技术中的一个重要分支为软件可靠性的评估与预测。软件可靠性增长模型(SRGM)是软件可靠性评估和测试的关键与核心。时至今日,软件可靠性模型已有上百种。早期出现的均为参数型软件可靠性增长模型,比如J-M模型<sup>[1]</sup>、G-O模型<sup>[2,3]</sup>等,但这类模型大多建立在非现实前提下,致使预测精确度不高<sup>[4]</sup>。Katherine<sup>[5]</sup>提出考虑具有故障相关性的可靠性增长模型,之后参数型软件可靠性模型得到发展。谢景燕<sup>[6]</sup>将排错过程中的不完美排错考虑在内,建立了基于NHPP的新型软件可靠性增长模型。但Kuei-Chen<sup>[7]</sup>等人认为NHPP、S型或者指数型参数模型可能只适合某些特殊失效数据,因此应用范围有限。之后Kai-yuan Cai<sup>[8]</sup>通过2组实验验证了基于NHPP建立的软件可靠性模型并不完全

符合实际缺陷数据。随着神经网络的出现,大量的基于缺陷数据作为驱动的软件可靠性增长模型出现, Yu-shen Su<sup>[9]</sup>, Q. P. Hu<sup>[10]</sup>等人使用基于神经网络的方法建立软件可靠性预计模型,并证实了其预测能力的相对准确型。尹乾<sup>[11]</sup>对基于神经网络的软件可靠性模型进行了深入研究,提出了级联软件可靠性模型、软件可靠性混合模型,最后引出了分层混合软件可靠性模型,通过实验证明了基于神经网络的软件可靠性模型比参数性软件可靠性模型在预测能力方面优势明显。Bo Yang<sup>[12]</sup>认为以数据作为驱动的软件可靠性模型能够缓解参数型软件可靠性模型不真实假设的问题,适合不同的失效数据建立软件可靠性模型。Pai Ping-feng<sup>[13]</sup>提出了使用支持向量基(SVM)加以模拟退火算法的方法建立软件可靠性增长模型,通过模拟退火算法对SVM所建立的模型的3个自身参数进行筛选,从而提高模型的预测精度。但软件缺陷数

到稿日期:2012-04-25 返修日期:2012-07-11 本文受国家重点实验室研究基金项目(SKLLIM0902-01),中科院知识创新项目(KGCX2-YW-911-2)资助。

刘 遼(1983-),男,博士生,主要研究方向为软件可靠性,E-mail:lllzx@sina.com。

据彼此存在联系,比如失效关联,错误修改过程中引入的新的错误等均是动态的且难以进行分类和界定。楼俊钢<sup>[14]</sup>对利用神经网络建立的软件可靠性增长模型做了统计,也确定了使用神经网络进行预测可以取得较好的效果。针对此类情况,本文选择动态模糊神经网络(DFNN)作为建立 SRGM 的基础。在原理上,模糊逻辑推理可以用一种有效的方式去获取失效关联度等不精确的特征。但 DFNN 的预测效果受其自身参数的影响,因此需要优化算法对参数进行动态调整,以使 DFNN 建立的 SRGM 的缺陷数据预测效果更加理想。

## 2 动态模糊神经网络<sup>[15]</sup>

### 2.1 动态模糊神经网络的特点

“动态”是指 DFNN 的网络结构并非预先设置好的,而具有动态变化特性。即网络在开始学习前没有已经设定好的模糊规则,网络的模糊规则是在学习过程中逐一生成的。与模糊神经网络(FNN)相比,动态模糊神经网络(DFNN)具有如下特点:

- (1) 连续学习的方法。
- (2) 参数估计与结构辨识同时进行。
- (3) 分级学习方法的使用大大提升了网络学习的效率。
- (4) 在网络训练过程中使用了商空间来划分整个输入空间,因此模糊规则数不随输入的增加而呈指数增长。

(5) 修剪技术的应用,使得网络的结构不会连续增长,避免了网络出现过度拟合和过度训练现象,从而确保了网络的泛化能力。

对于建立 SRGM 来说,DFNN 还具有以下特点:

- (1) 采用了模糊化的运行机制,因此加强了系统的容错能力。
- (2) DFNN 可同时处理确定性和不确定性信息。
- (3) DFNN 的自身参数确定下来以后,DFNN 预测效果非常稳定。

因此,鉴于软件错误引入率、失效数据的关联性和测试排错人员对软件的逐渐熟悉而促进的技能提升等很难进行界定和分类的难点,适合使用 DFNN 来建立 SRGM。

### 2.2 动态模糊神经网络的结构

动态模糊神经网络的结构如图 1 所示。图中  $x_1, x_2, \dots, x_r$  作为网络的输入变量,  $Y$  作为网络的输出,  $MF_{ij}$  是第  $i$  个输入变量的第  $j$  个隶属函数,  $R_j$  是第  $j$  条模糊规则,  $N_j$  表示第  $j$  个归一化节点,  $W_j$  是第  $j$  个规则的连接权,  $u$  指网络总的规则数。第 1 层是输入层;第 2 层是隶属函数层;第 3 层是 T-范数层;第 4 层是归一化层;第 5 层是输出层。本文不对 DFNN 的结构做详细介绍,具体内容参见文献[5]。

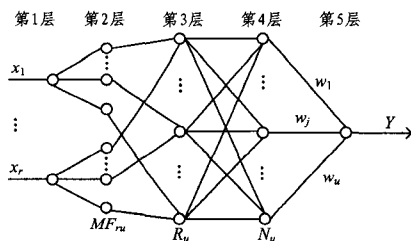


图 1 DFNN 的结构

### 2.3 动态模糊神经网络的学习算法<sup>[16]</sup>

首先,介绍 DFNN 的 10 个参数:  $\epsilon_{\min}$  和  $\epsilon_{\max}$  分别为 DFNN 的最小调试标准和最大调试标准;  $\epsilon_{\max}$  和  $\epsilon_{\min}$  分别为最大输出误差和最小输出误差;  $\gamma$  为衰变常量;  $\beta$  为收敛常数;  $\sigma_0$  为第 1 条模糊规则隶属函数的宽度;  $k$  为径向基函数单元的重迭因子;  $k_w$  为宽度递增因子;  $k_{err}$  为规则门檻。

其次,介绍动态模糊神经网络学习算法的步骤:

(1) 第一个数据进来以后,生成第一个规则,并确定第一个规则中的 DFNN 的自身参数。

(2) 从第二个数据起,每来一个样本  $(X_i, t_i)$  ( $i > 1$ ) ( $X_i$  是输入,  $t_i$  是期望输出), 计算  $d_i(j)$ , 即得:

$$d_i(j) = \| X_i - C_j \|, j=1, 2, \dots, u \quad (1)$$

式中,  $u$  是存在的模糊规则数。然后找寻出  $d_i(j)$  的最小值  $d_{\min}$ :

$$d_{\min} = \arg \min(d_i(j)) \quad (2)$$

(3) 实际的输出与期望的输出之间的误差  $e_i$ :

$$\| e_i \| = \| t_i - Y_i \| \quad (3)$$

式中,  $Y_i$  为实际的输出。

(4) 若满足  $d_{\min} > K_d$  ( $K_d$  为最大边界有效半径), 就转到(5), 否则转到(6)。

(5) 若  $e_i > K_e$  成立 ( $K_e$  由动态模糊神经网络期望精度而预先设定), 则生成新规则, 确定此规则下的 DFNN 自身参数, 并计算误差下降率以及第  $i$  个规则的重要性  $\eta_i$  ( $i=1, 2, \dots, u$ ); 若条件不满足, 则自动调整已存在规则的 DFNN 后件参数。

(6) 若  $e_i > K_e$  成立, 则调整已经存在的规则下的 DFNN 前件参数中的宽度, 随后调整规则的 DFNN 后件参数; 反之则自动调整已经存在的规则的 DFNN 后件参数。

(7) 若  $\eta_i > k_{err}$  成立, 则删除第  $i$  条规则; 反之则自动调整已经存在规则的 DFNN 后件参数。

(8) 判断训练是否结束, 若没有则返回(2); 反之则结束学习过程。图 2 为 DFNN 学习算法的流程图。

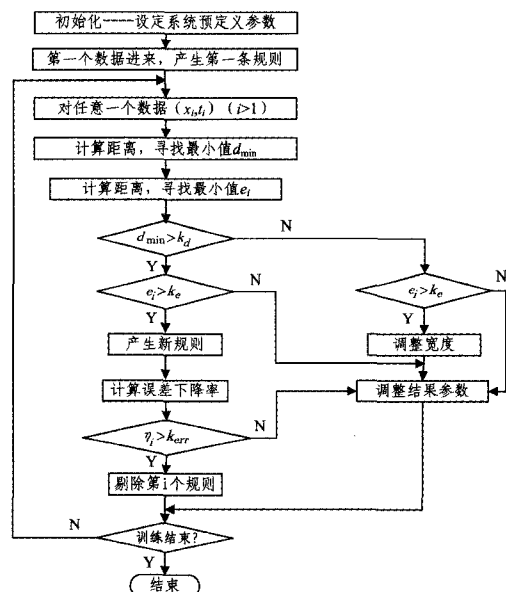


图 2 动态模糊神经网络的学习算法

### 3 参数动态调整的 DFNN

#### 3.1 选择遗传算法作为参数动态调整方法的原因

用 DFNN 建立的 SRGM 预测能力稳定的原因在于,一旦 DFNN 自身参数确定,根据同一组训练数据进行多次预测所得到的预测值就一致,预测结果不会出现波动。在 DFNN 自身参数集合变化不大并且参数没有超过各自参数临界值的情况下,DFNN 的预测结果波动很小。虽然遗传算法容易陷入局部最优并且不宜得到全局最优解,但次优解甚至更次优解均可以满足 DFNN 的预测能力精确的要求。将我所某国防项目的软件失效数据共分为 60 个周期,对后 12 个周期的缺陷数据分别短期预测,分别选择了 3 组不同的参数集合(其中参数集合 1 和参数集合 2,3 相差较大,参数集合 2,3 相近)来证明参数在小范围内变化对预测所带来的微小影响。如图 3 所示,参数集合 2 和参数集合 3 在短期预测结果上基本重合,而参数集合 1 较参数集合 2,3 相差较多,从第 55 个周期的缺陷数据开始才出现微小的变化。参数集合 2 和参数集合 3 之间仍然可以取得很多种参数集合,单步预测结果也基本上和参数集合 2,3 的预测输出重合。

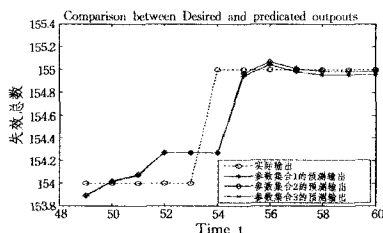


图 3 基于 3 组相近参数集合的 DFNN 预测结果

#### 3.2 基于遗传算法<sup>[17]</sup>的动态模糊神经网络(GA-DFNN)

遗传算法来源于对生物进化过程的模拟,它将问题的求解表示成染色体的适者生存的过程,通过染色体的一代代进化,最终收敛到最适合环境的个体(即问题的最优解或者称为满意解)。

##### 3.2.1 初始化染色体

为了方便查看 DFNN 的 10 个自身参数的取值,将对染色体进行实数编码。设参数集合为  $P = \{P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10}\}$ , 参数集合  $P$  则为一串实数编码的染色体。其中  $P_i = \{P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{in}\} (i=1, 2, \dots, 10; j=1, 2, \dots, n)$ ,  $P_{ij}$  是参数集合  $P$  中第  $i$  个参数的第  $j$  种可能取值。其中,每一个参数均有属于自己的取值范围,取值范围由动态模糊神经网络性质决定,它们分别为:  $0 < \epsilon_{\min} < \epsilon_{\max}; 0 < e_{\min} < e_{\max}; \sigma_0 > 0; k_{\sigma} > 0; k_w > 1; k > 1; 0 < \gamma < 1; 0 < \beta < 1$ 。染色体群体规模过大会造成计算爆炸,染色体群体规模过小会过早收敛于局部最优解,所以选择适中的群体规模可避免计算爆炸和解过早收敛的问题。因此,本文将初始染色体数设置为 20 个。

##### 3.2.2 适应度函数

基于 DFNN 建立的 SRGM 模型在训练阶段过程中拟合度很好,但有时在预测过程中会持续出现预测输出和实际输出重合度很差的情况,这种情况称为预测效果畸变,如图 4、

图 5 所示。为了防止预测效果畸变,本文提出了一种仿预测的方法。即将训练数据分成两部分,第一部分用于训练,用由第一部分数据训练好的 DFNN 网络对第二部分的数据进行预测,用于仿预测的数据量不超过训练数据的 15% 即可。第二部分数据仅需要判断网络是否会在仿预测阶段出现畸变,若出现畸变,则淘汰这组参数集合。在仿真实验中通过仿预测手段,可以有效地屏蔽掉预测过程中出现的预测误差偏大问题。

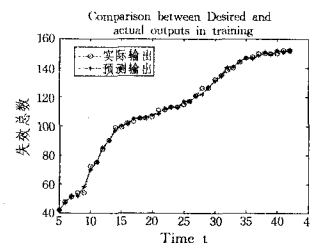


图 4 DFNN 训练过程

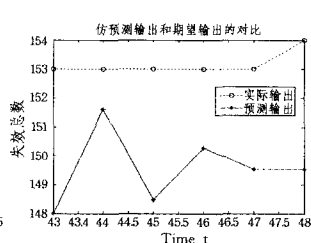


图 5 仿预测阶段的预测畸变

设仿预测阶段的误差平方和为 GA 的目标函数  $f(x)$ , 即

$$f(x) = SSE = \sum_{i=1}^n (y_i - m(t_i))^2 \quad (4)$$

式中,  $y_i$  为实际输出,  $m(t_i)$  为预测输出。适应度函数为标准适应度函数:

$$F(x) = \begin{cases} C_{\max} - f(x), & f(x) < C_{\max} \\ 0, & f(x) \geq C_{\max} \end{cases} \quad (5)$$

式中,  $C_{\max}$  为一充分大的正数。

##### 3.2.3 选择算子

遗传算法使用选择来对染色体进行优胜劣汰,适应度较高的个体遗传到下一代的概率较大,而适应度较低的个体遗传到下一代的概率较小。为减轻计算机在选择过程中的运算量,本文选择排序选择方法。在生成 20 组参数集合染色体之后,将这 20 组染色体按照适应度函数从高到底的排列,将事先设计好的概率表按顺序分配给这 20 组染色体作为它们的选择概率,从中选取 7 组直接复制到下一子代中。

##### 3.2.4 交叉算子

交叉是指把两个父代个体的部分基因彼此交换,重新组成新个体的过程。因为染色体参数集合  $P$  并不是一个排列,所以仅选用简单的交叉方式即可。单点交叉符合此要求。本文选择 10 组染色体参数集合进行单点交叉。对这 10 组染色体参数集合随机配对,并对它们进行单点交叉。

##### 3.2.5 变异算子

变异算子的基本作用是对某染色体上个别基因值做变动,这有利于改善遗传算法的局部搜索能力并维持群体的多样性,防止出现早熟现象。在 20 组染色体参数集合中还剩下 3 组染色体参数集合用于变异,一般情况下,变异的概率不应过高,但对于初始个体为 20 的参数集合群体来说,它能为参数集合增加更多新的参数元素,因此变异过程占有极其重要的作用。变异过程为  $P_{ij} \rightarrow P_{im} (j \neq m)$ 。

##### 3.2.6 终止条件

当最优个体的适应度达到给定的阈值,或者最优个体的

适应度不再上升,或者迭代次数达到预测的迭代次数时,算法终止。算法终止预定的迭代次数一般设置为 100~500 次,本文选择迭代次数 300 作为遗传算法的终止条件。

基于遗传算法的动态模糊神经网络的学习过程如图 6 所示。

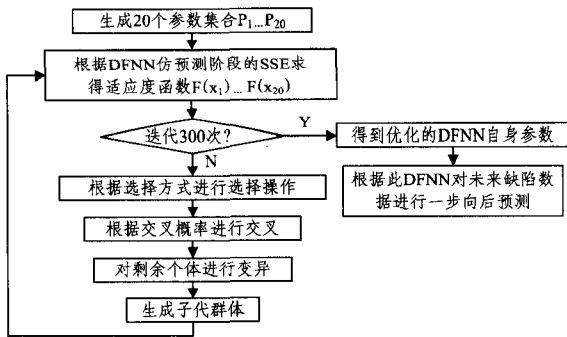


图 6 基于 GA 的 DFNN 学习算法

#### 4 基于 GA-DFNN 建立的 SRGM

基于神经网络建立的 SRGM 均是以数据作为驱动的软件可靠性增长模型,  $N$  输入  $M$  输出作为数据驱动的 SRGM 的框架。为了提高预测的精度以及对短期预测, 本文选择 4 输入 1 输出作为数据驱动的框架, 以  $x_i, x_{i+1}, x_{i+2}, x_{i+3} (i=1, 2, 3, \dots, n)$  作为可靠性增长模型的输入, 以  $x_{i+4}$  作为可靠性增长模型的输出, 以此对 GA-DFNN 进行训练并完成预测。本文选择 2 组公开数据和我所某型号项目软件采集到的失效数据来对 GA-DFNN 建立的 SRGM 进行验证, 并将其与经典 BPN、FNN 所建立的 SRGM 进行比较。BPN 和 FNN 是以前被公认的具有强预测能力的神经网络, 而神经网络结构的类型也能影响预测精度<sup>[18]</sup>, 因此将以 GA-DFNN 建立的 SRGM 与 BPN、FNN 建立的 SRGM 预测模型进行比较。

第一组数据来源于文献[1], 一共有 32 组缺陷数据, 选取前 26 个数据作为训练集合, 其中第 24~26 个缺陷数据作为仿预测数据使用, 后 6 个作为预测数据。经过 SA-DFNN 的学习算法, 得到 DFNN 的自身参数为:  $\epsilon_{\min}=0.02, \epsilon_{\max}=1.4, \epsilon_{\max}=4, \epsilon_{\min}=0.02, \gamma=0.9, \beta=0.95, \sigma_0=1.1, k=1.1, k_w=1.05, k_{err}=0.01$ 。因为 FNN 和 BPN 建立的 SRGM 的预测结果不稳定, 所以分别选择预测效果较好的一组进行比较。基于 FNN 建立的 SRGM 的网络结构为 7 个隐层节点, 网络学习率为 0.08, 迭代次数设置为 100。基于 BPN 建立的 SRGM 的网络结构为 9 个隐层节点, 迭代次数设置为 100。预测结果和预测误差如图 7、图 8 所示。

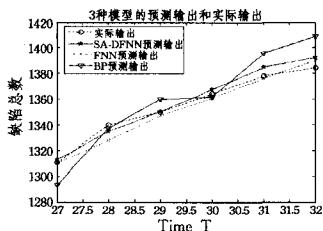


图 7 3 种模型的预测输出和实际输出

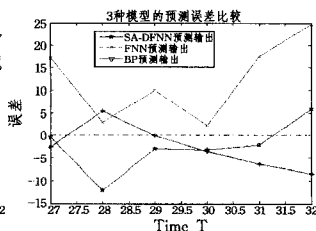


图 8 3 种模型的预测误差

除了用式(4)来判断预测效果, 也可以用 AE 即预测误差均值来判断预测效果。

$$AE = \frac{1}{M} \sum_{i=1}^M \left| \frac{m(t_i) - y_i}{y_i} \right| \quad (6)$$

这 3 种模型建立的 SRGM 的预测 SSE 和 AE 值如表 1 所列。可以看出, 基于 SA-DFNN 建立的 SRGM 的预测 SSE 和 AE 均为最小值, 预测效果比 FNN 和 BPN 所建立的 SRGM 要好。

表 1 3 种模型的 SSE 和 AE

SGRM	SSE	AE
SA-DFNN	159.8488	0.0032
FNN	207.4216	0.0033
BPN	1.3228e+003	0.0091

第二组数据来源于文献[7], 一共有 22 组缺陷数据, 选取前 18 个数据作为训练集合, 其中第 17, 18 个缺陷数据作为仿预测数据使用, 后 5 个作为预测数据。经过 SA-DFNN 的学习算法, 得到 DFNN 的自身参数为:  $\epsilon_{\min}=0.05, \epsilon_{\max}=1.4, \epsilon_{\max}=1.8, \epsilon_{\min}=0.1, \gamma=0.98, \beta=0.98, \sigma_0=1, k=1.2, k_w=1.1, k_{err}=0.005$ 。因为 FNN 和 BPN 建立的 SRGM 的预测结果不稳定, 所以分别选择预测效果较好的一组进行比较。基于 FNN 建立的 SRGM 的网络结构为 5 个隐层节点, 网络学习率为 0.08, 迭代次数设置为 100。基于 BPN 建立的 SRGM 的网络结构为 9 个隐层节点, 迭代次数设置为 100。预测结果和预测误差如图 9、图 10 所示。

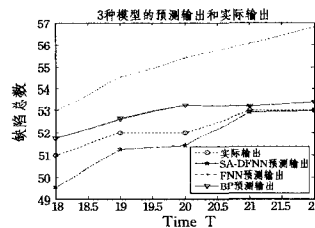


图 9 3 种模型的预测输出和实际输出

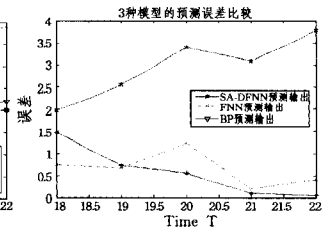


图 10 3 种模型的预测误差

这 3 种模型建立的 SRGM 的预测 SSE 和 AE 值如表 2 所列。可以看出, 在预测阶段基于 SA-DFNN 建立的 SRGM 的 SSE 比基于 BPN 建立的 SRGM 的 SSE 略大, 但 AE 为 3 个模型中的最小值, 并且预测效果非常稳定。

表 2 3 种模型的 SSE 和 AE

模型	SSE	AE
SA-DFNN	2.9869	0.0112
FNN	45.5915	0.0564
BPN	2.7050	0.0124

第三组数据来源于我所某型号项目软件, 一共采集到 60 组缺陷数据, 如表 3 所列。选取前 48 个数据作为训练集合, 其中第 43~47 个缺陷数据作为仿预测数据使用, 后 12 个作为预测数据。经过 SA-DFNN 的学习算法, 得到 DFNN 的自身参数为:  $\epsilon_{\min}=1, \epsilon_{\max}=4, \epsilon_{\max}=1.9, \epsilon_{\min}=0.01, \gamma=0.97, \beta=0.94, \sigma_0=1, k=1.2, k_w=1.1, k_{err}=0.001$ 。因为 FNN 和 BPN 建立的 SRGM 的预测结果不稳定, 所以分别选择预测效果较好的一组进行比较。基于 FNN 建立的 SRGM 的网

络结构为 8 个隐层节点,网络学习率为 0.05,迭代次数设置为 100。基于 BPN 建立的 SRGM 的网络结构为 9 个隐层节点,迭代次数设置为 100。预测结果和预测误差如图 11、图 12 所示。

表 3 某型号项目软件失效数据

periods	失效数据	periods	失效数据	periods	失效数据
1	16	21	111	41	152
2	24	22	111	42	152
3	27	23	113	43	153
4	33	24	113	44	153
5	42	25	115	45	153
6	48	26	117	46	153
7	51	27	121	47	153
8	54	28	126	48	154
9	54	29	126	49	154
10	72	30	132	50	154
11	75	31	135	51	154
12	84	32	138	52	154
13	90	33	140	53	154
14	99	34	144	54	155
15	99	35	147	55	155
16	102	36	148	56	155
17	103	37	150	57	155
18	105	38	150	58	155
19	105	39	150	59	155
20	106	40	150	60	155

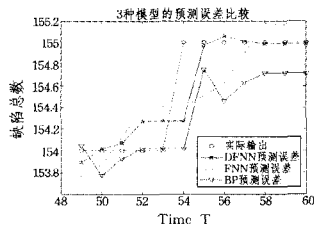


图 10 3 种模型的预测输出和实际输出

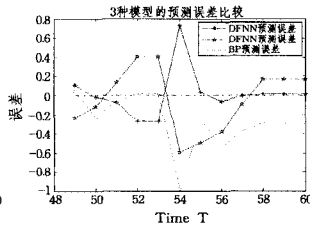


图 11 3 种模型的预测误差

这 3 种模型建立的 SRGM 的预测 SSE 和 AE 值如表 4 所列。可以看出,基于 SA-DFNN 建立的 SRGM 的预测 SSE 和 AE 均为最小值,预测效果比 FNN 和 BPN 所建立的 SRGM 要好。

表 4 3 种模型的 SSE 和 AE

模型	SSE	AE
SA-DFNN	0.7017	0.0017
FNN	1.2472	0.0018
BPN	1.7655	0.0018

**结束语** 通过 DFNN 建立软件可靠性增长模型,并在训练数据中选择末尾少数训练数据来做仿预测实验。在仿预测实验中利用 GA 求出 DFNN 自身参数的优化解,实现了 DFNN 自身参数根据软件缺陷数据动态调整的功能。在 DFNN 自身参数确定之后,则可对软件缺陷数据进行短期预测。利用 3 组软件缺陷数据,将 GA-DFNN 建立的 SRGM 和 FNN 以及 BPN 建立的 SRGM 相比较,证明了基于 GA-DFNN 建立的 SRGM 在短期预测方面精度高,预测性能稳定,具有一定通用性。

## 参考文献

- [1] Chang Yen-chang, Liu Ching-ti. A generalized JM model with applications to imperfect debugging in software reliability[J]. Applied Mathematical Modelling, 2009, 33: 3578-3588
- [2] 刘宏伟,杨孝宗,曲峰,等.非齐次泊松过程类软件可靠性增长模型[J].同济大学学报:自然科学版,2004,32(8):1071-1074
- [3] Goel A L, Okumoto K. Time-dependent error-detection rate model for software reliability and other performance measures [J]. IEEE Transaction on Reliability, 1979, 28: 206-211
- [4] 张荣辉,姜楠,勾郎.一种考虑缺陷关联的软件可靠性增长模型[J].计算机工程,2008,34(8):44-46
- [5] Goseva-Popstojanova K, Trivedi K S. Failure correlation in software reliability model[J]. IEEE Trans Reliability, 2000, 49(1): 37-48
- [6] 谢景燕,安金霞,朱纪洪.考虑不完美排错情况的 NHPP 类软件可靠性增长模型[J].软件学报,2010,21(5):942-949
- [7] Chiu K-C, Huang Y-S, Lee T-Z, et al. A study of software reliability growth from the perspective of learning effects[J]. Reliability Engineering and System Safety, 2008, 93: 1410-1421
- [8] Cai Kai-yuan, Hu De-bin, Bai Cheng-gang, et al. Does software reliability growth behavior follow a non-homogeneous Poisson process[J]. Information and Software Technology, 2008, 50: 1232-1247
- [9] Su Yu-shen, Huang Chin-yu. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models[J]. The journal of Systems and Software, 2007, 80: 606-615
- [10] Hu Q P, Xie M, Ng S H, et al. Robust recurrent neural network modeling for software fault detection and correction prediction [J]. Reliability Engineering and System Safety, 2007, 92: 332-340
- [11] 尹乾.基于神经网络的软件可靠性模型[D].北京:北京师范大学,2006
- [12] Yang Bo, Li Xiang, Xie Min. A generic data-driven software reliability model with model mining technique[J]. Reliability Engineering and System safety, 2010, 95: 671-678
- [13] Pai Ping-feng, Hong Wei-chiang. Software reliability forecasting by support vector machines with simulated annealing algorithms [J]. The journal of systems and software, 2006, 79: 747-755
- [14] 楼俊钢,江建慧,帅春燕,等.软件可靠性模型研究进展[J].计算机科学,2010,37(9):13-20
- [15] 伍世虔,徐军.动态模糊神经网络-应用与设计[M].北京:清华大学出版社,2008
- [16] 陈彦,李月明.基于动态模糊神经网络的变桨距系统辨识[J].电气技术,2011(1):18-20
- [17] 高玮,尹志喜.现代智能仿生算法及其应用[M].北京:科学出版社,2011
- [18] 刘喜成.软件可靠性工程手册[M].北京:电子工业出版社,1997