

基于改进 Elman 网络模型的软件可靠性预测

程绪超^{1,2}, 陈新宇^{1,2}, 郭平¹

(1. 北京师范大学 图形图像与模式识别实验室, 北京 100875; 2. 中国科学院软件所 计算机科学国家重点实验室, 北京 100190)

摘 要: 为提高神经网络模型对软件可靠性预测结果的准确性和可信性, 提出了一种基于多目标优化算法改进 Elman 网络模型 Mop-Elman (multi-objective optimization-based improved Elman neural network) 的方法: 1) 在 Elman 网络基础上, 设计输出层的延迟反馈层, 作为另一个状态层; 2) 以网络的结构和 2 个状态层的初始输出值为网络配置的变量, 以网络的预测精度和顽健性为目标, 采用 NSGA-II (non-dominated sorting genetic algorithm II) 进行多目标优化得到帕累托解, 最大化网络预测精度与顽健性之和从而确定网络配置。通过两组实际软件失效数据对 Mop-Elman 进行实验验证, 并与前馈网络、Elman 网络、单目标优化 Elman 网络以及多目标优化 Elman 网络进行比较研究, 结果表明 Mop-Elman 的预测结果具有较高的准确性和可信性。

关键词: 软件可靠性预测; 可信性; 反馈神经网络; NSGA-II; 多目标优化

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2011)04-0086-08

Software reliability prediction with an improved Elman network model

CHENG Xu-chao^{1,2}, CHEN Xin-yu^{1,2}, GUO Ping¹

(1. Image Processing and Pattern Recognition Laboratory, Beijing Normal University, Beijing 100875, China;

2. State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In order to improve accuracy and dependability of using neural network for software reliability prediction, a multi-objective optimization-based improved Elman recurrent network method (Mop-Elman) was proposed. First, on the basis of the Elman network, a self-delay feedback of the output layer as another context layer was designed. Second, the network architecture and the initial outputs of these two context layers were taken as variables of network configuration setting, and NSGA-II was employed to simultaneously optimize prediction performance and robustness, then the Pareto solution was obtained. After that, by maximizing the sum of prediction performance and robustness, the final network configuration setting was determined. Finally, the proposed method was compared with the feed-forward neural network, the Elman network, both the single-objective and the multi-objective optimization Elman networks with respect to two real software failure data. It demonstrated that the proposed Mop-Elman achieves higher prediction accuracy and dependability.

Key words: software reliability prediction; dependability; recurrent network; NSGA-II; multi-objective optimization

收稿日期: 2010-04-21; 修回日期: 2011-01-04

基金项目: 国家高技术研究发展计划 (“863”计划) 基金资助项目 (2009AA010314); 国家自然科学基金资助项目 (90820010, 60911130513); 中国科学院计算机科学重点实验室开放课题基金资助项目 (SYSKF0906)

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2009AA010314); The National Natural Science Foundation of China (90820010, 60911130513); The Open Project of State Key Laboratory of Computer Science of Chinese Academy of Sciences (SYSKF0906)

1 引言

可信软件的重要属性之一是可靠性,对软件可靠性进行预测一直是人们重视的研究方向。软件可靠性预测主要是推断软件失效行为,从而量化软件的可靠性状态和行为,帮助开发可信软件。围绕软件可靠性预测,国内外已建立的软件可靠性模型不下上百种^[1-5],其中基于神经网络的软件可靠性模型的研究越来越受到重视^[6-8]。研究表明,神经网络具备非常强的非线性运算能力,且可根据失效过程的动态变化进行自适应调整。然而目前基于神经网络建立的软件可靠性模型大多采用静态前馈神经网络^[6,7],而且建立的模型一般只考虑预测结果的准确性,对模型本身的顽健性和预测结果的可信性考虑较少。因而研究如何利用神经网络构建软件可靠性模型获得更高的预测准确性和可信性,继而增强模型的通用性是具有重要的理论意义和实用价值的。

能够更直接反映过程动态特性的是反馈神经网络(recurrent neural network)^[9],目前大多数应用是利用反馈网络的动态特性进行过程预测^[10-14]。虽然软件的失效过程也是一个随时间变化的动态过程,但是利用反馈网络的动态特性进行软件可靠性的预测研究却很少^[8]。Elman 网络是一种典型的动态反馈神经网络,它具备映射动态特征的功能,具有适应时变特性的能力^[12]。神经网络的结构对动态系统的预测具有重要的影响。此外,为了提高预测结果的可信性,一般采取神经网络训练多次取平均值的方法,但始终没有一种模型既能保证准确性又能直接观察结果的可信性。因此,本文在改进神经网络结构的基础上,考虑采用优化算法同时优化软件可靠性预测结果的准确性和可信性,提出一种基于多目标优化算法改进 Elman 网络预测模型 Mop-IElman 方法。在此基础上,采用了两组失效数据验证和比较 Mop-IElman 方法与前馈网络、Elman 网络、单目标优化的 Elman 网络以及多目标优化的 Elman 网络在软件可靠性预测上的整体性能。

2 Mop-IElman 网络模型

为方便,统一对文中出现的符号做如下说明^[12]:

w^1/w^2 权值向量连接输入层/隐层到隐层/输出层;

w^3/w^4 权值向量连接状态层 1/2 到隐层/输出层

net_h_i, o_h_i	隐层第 i 个节点的输入和输出;
net_c1_i, net_c2_i	状态层 1/2 第 i 个节点的输入值;
o_c1_i, o_c2_i	状态层 1/2 第 i 个节点的输出值;
$g(x)$	隐层节点的激活函数;
k	训练次数;
$slide_w$	滑动窗口大小;
z^{-1}	延迟单元;
m_c	动量因子;
η	学习率;
$\nabla f(w(k))$	误差函数的梯度。

软件可靠性失效数据有 2 种表示形式:时间失效数和失效间隔时间。当使用时间失效数时,软件失效的动态预测过程按神经网络进行滑动预测的映射方法可描述为^[8]

$$P: \{State_t; O_{t-slide_w+1}, \dots, O_{t-1}, O_t\} \rightarrow O'_{t+1} \quad (1)$$

其中, $State_t$ 代表 t 时刻训练好的网络; $O_{t-slide_w+1}, \dots, O_{t-1}, O_t$ 给出了最近的在滑动窗口内 $slide_w$ 个失效情况,作为网络的输入;输出 O'_{t+1} 是对下一阶段失效情况的预测,这称为单步预测。当得到新的失效记录时,则更新训练样本集。使用失效间隔时间形式的神经网络表示原理是相同的。

单步预测在软件开发过程中更具有实时的指导意义,因此本文采用单步预测。神经网络对权值初始值非常敏感,不同的初始化可能产生截然不同的输出结果。为避免预测的随机性,须进行 m 次不同的网络初始化,取其平均值作为预测输出值。但除了关注模型预测精度,还应观察模型的顽健性,保证预测结果的稳定可信,从而在整体上提高模型的性能。此外,神经网络更适用于平滑的数据样本^[6],将失效数据转换成累计失效数或累计失效时间形式有利于提高网络的性能。

Mop-IElman 是在改进 Elman 网络结构基础上提出的模型,采用了多目标优化算法预测软件可靠性。关于 Elman 网络的详细描述可参见文献^[12]。

与以往研究不同之处在于:一是通过增加新的状态层改变网络结构从而得到改进的 IEIelman 网络,并考虑 2 个状态层初始值对网络模型影响;二是结合 NSGA-II^[15,16]进行 IEIelman 网络预测结果准确性和可信性的多目标优化。

2.1 IElman 网络

Elman 网络通过状态层对隐层的本地连接反馈，使它不仅对现有的输入数据敏感，也会综合历史输入的数据。考虑到输出层直接反映网络的输出，为清晰地表示网络的最终输出，仿照隐层的本地反馈，本文设计增加输出层的本地反馈^[12,13]改进 Elman。如此，IElman 网络的结构由图 1 所示，共包含 5 层，除了 Elman 网络的 4 层，新增加了输出层的延迟反馈层。所以 IElman 网络具有 2 个状态层，分别称之为状态层 1 和状态层 2。状态层 1 为隐层的延迟反馈层，状态层 2 则为新增的输出层的延迟反馈层。

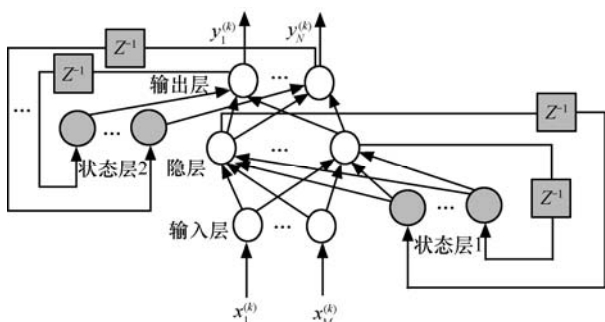


图 1 IElman 网络结构

IElman 网络的表达式为

$$\begin{aligned}
 o_c1_i^{(k)} &= net_c1_i^{(k)} \\
 net_h_j^{(k)} &= \sum_{i=1}^M w_{i,j}^1 x_i^{(k)} + \sum_{i=1}^L w_{i,j}^3 o_c1_i^{(k)} \\
 o_h_j^{(k)} &= g(net_h_j^{(k)}) \\
 o_c2_i^{(k)} &= net_c2_i^{(k)} \\
 y_j^{(k)} &= \sum_{i=1}^L w_{i,j}^2 o_h_i^{(k)} + \sum_{i=1}^N w_{i,j}^4 o_c2_i^{(k)} \quad (2)
 \end{aligned}$$

由于状态层 1 的输入是隐层输出的一步延迟，状态层 2 的输入是输出层输出的一步延迟，因此有：

$$\begin{aligned}
 net_c1_i^{(k)} &= o_h_i^{(k-1)} \\
 net_c2_i^{(k)} &= y_i^{(k-1)} \quad (3)
 \end{aligned}$$

2.2 网络学习算法

学习算法采用自适应学习率附加动量梯度下降算法调整权值，表示为

$$\begin{aligned}
 W(k+1) &= W(k) + (1 - m_c) \eta \nabla f(W(k)) + \\
 & m_c (W(k) - W(k-1)) \quad (4)
 \end{aligned}$$

其中，学习率随着网络性能变化而自适应地调整。误差函数采用

$$E = \frac{1}{2} \sum_{k=1}^T \sum_{i=1}^N (Y_i^{(k)} - y_i^{(k)})^2 \quad (5)$$

$Y_i^{(k)}$ 是网络输出层第 i 个节点的真实输出， T 为训练总次数。 W^1 和 W^2 的变换梯度公式同前馈网络^[17]， W^3 和 W^4 的变换梯度如式(6)和式(7)：

$$\begin{aligned}
 \nabla f(w_{i,j}^3(k)) &= \frac{\partial E}{\partial w_{i,j}^3(k)} = \sum_{i=1}^N \frac{\partial E}{\partial y_i^{(k)}} \frac{\partial y_i^{(k)}}{\partial o_h_j^{(k)}} \frac{\partial o_h_j^{(k)}}{\partial w_{i,j}^3(k)} \\
 &= - \sum_{i=1}^N (Y_i^{(k)} - y_i^{(k)}) w_{i,j}^2(k) f'(net_h_j^{(k)}) o_c1_j^{(k)} \quad (6)
 \end{aligned}$$

$$\begin{aligned}
 \nabla f(w_{i,j}^4(k)) &= \frac{\partial E}{\partial w_{i,j}^4(k)} = \frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial w_{i,j}^4(k)} \\
 &= -(Y_j^{(k)} - y_j^{(k)}) o_c2_i^{(k)} \quad (7)
 \end{aligned}$$

2.3 Mop 多目标优化

一方面，IElman 网络的结构对网络性能有非常重要的影响；另一方面，IElman 网络的性能也受状态层初始值的影响^[14]。从式(2)可知，状态层 1 和状态层 2 的初始输出 $o_c1^{(1)}$ 和 $o_c2^{(1)}$ 对于 IElman 网络的输出具有一定影响。从式(3)可见 $net_c1^{(1)} = o_h^{(0)}$ ， $net_c2^{(1)} = y^{(0)}$ ，默认情况下 $o_h^{(0)} = 0$ ， $y^{(0)} = 0$ ，将导致 $o_c1^{(1)} = 0$ ， $o_c2^{(1)} = 0$ 。然而根据式(6)和式(7)，它们对于网络的学习算法具有不可忽略的作用。因此须通过配置 IElman 网络的这 2 个方面以某种优化方法提高它的整体性能。

2.3.1 NSGA-II

NSGA-II^[15,16] 是基于 NSGA 改进的非支配排序遗传算法，是目前公认的一种并行全局寻优算法，特别适于解决复杂的多目标优化问题。它与标准遗传算法的区别仅在于选择算子的操作过程，交叉与变异算子仍保持不变。本文借助 NSGA-II 寻找 IElman 网络的优化配置，在选择过程中采用了拥挤联赛选择算子。

首先，建立一个随机初始种群 P_n 并初始化进化代数计数器 $n=0$ ；然后对种群 P_n 的所有个体进行非劣分层，并计算每一非劣层所有个体的拥挤距离；接着使用拥挤联赛选择算子、交叉算子和变异算子建立规模为 N 的子代种群 Q_n ；最后，对联合的亲代种群和子代种群进行非劣分层并填充新一代种群 P_{n+1} ；如此循环，直到进化到指定的最大进化代数 $MaxGen$ 。

当将 NSGA-II 和模型的优化结合时，只需确定目标函数表达式、目标个数、决策变量个数、决策

变量范围、种群的大小以及最大进化代数便可利用文献[15]提供的伪代码实现。

2.3.2 目标函数

假设经过归一化后的失效数据集 $\{O_1, O_2, \dots, O_t\}$ 用于网络配置的寻优, 如果 t_0 是第一个预测点, 那么就需要利用数据点 $\{O_1, O_2, \dots, O_{t_0-1}\}$ 训练网络, 获得预测值 O'_{t_0} 。由于共需预测 m 次, 把第 j 次的预测结果记录为 $O'_{t_0,j}$, $j=1, \dots, m$ 。重复上述预测步骤直到满足预定的停止条件, 记录所有的 $O'_{i,j}$, $i=t_0, \dots, t$; $j=1, 2, \dots, m$ 。记

$$\begin{aligned} O'_i &= \frac{1}{m} \sum_{j=1}^m O'_{i,j} \\ SE_i &= |O'_i - O_i|^2 \\ SD_i &= \frac{1}{m} \sum_{j=1}^m (O'_{i,j} - O'_i)^2, \\ i &= t_0, \dots, t \end{aligned} \quad (8)$$

采用平均平方误差 (MSE, mean square error) 和平均方差 (MSD, mean square deviation) 来分别评测软件可靠性模型的预测精度和顽健性, 即预测结果的准确性和非随机性 (稳定性)。采用 MSE 和 MSD 作为评测预测精度和顽健性可行性分析可参见文献[8], 它们的表达式分别如下:

$$\begin{aligned} MSE &= \frac{1}{t - t_0 + 1} \sum_{i=t_0}^t SE_i \\ MSD &= \frac{1}{t - t_0 + 1} \sum_{i=t_0}^t SD_i \end{aligned} \quad (9)$$

由此, 为得到最大的适应度值, 可得到 2 个目标函数:

$$L1(\theta) = 10 - MSE \quad (10)$$

$$L2(\theta) = 10 - MSD \quad (11)$$

其中, θ 表示网络配置, 根据影响网络整体性能的因素分析, $\theta=(slide_w, number_hidden, candidate1, candidate2)$, 包括了输入层节点数 $slide_w$ 、隐层节点数 $number_hidden$, 状态层 1 的初始输出值 $o_c1^{(1)}$ 记为 $candidate1$, 以及状态层 2 的初始输出值 $o_c2^{(2)}$ 记为 $candidate2$ 。根据目标函数式(10)和式(11)可利用 NSGA-II 寻找到既保证预测精度又具有顽健性的网络配置 θ 。

3 实验分析

实验中利用两组软件失效数据对 Mop-IElman

进行检验。

3.1 评价标准

为有效地表示模型预测结果的准确性和可信性, 本文采用 MSE、MSD, 以及 MSE+MSD 3 种评价标准, 其计算方式如式(9)所示。MSE 越小表示预测准确性越高。MSD 越小表示模型预测可信性越高, 即模型每次预测结果的随机性越小, 预测结果更稳定。稳定性好的网络其预测结果的可信性也要相对高, 所以 MSD 在某种程度上反映了预测结果的可信性。在保证一定 MSE 和 MSD 基础上, MSE+MSD 越小表明预测的整体性能越好。

3.2 实验步骤

首先讨论实验所需的参数设置。网络训练次数 m 若取值太小, 则无法反映不同权值初始化导致网络输出值的波动情况; 但若 m 取值太大, 优化和学习过程会耗时更多; 所以本文首先对 $m=5$ 进行了研究。

t_0 表示从第 t_0 个数据点开始预测。若 t_0 过小, 则初始预测过程就没有足够的点; 而神经网络是数据驱动模型, 如果没有足够可用的数据点无法完成自适应调整。因此, 预测点不宜过早。相反, 若 t_0 过大, 预测点过晚, 则失去神经网络用于预测的真正目的。再者本文所采用的两组数据点不多, 因此折中取 $t_0=6$ 。

根据影响网络整体性能的因素分析, NSGA-II 寻优需要确定的网络配置 $\theta=(slide_w, number_hidden, candidate1, candidate2)$ 。虽然 NSGA-II 采用全局搜索的方法可自动配置网络, 但不能期望在较短时间内所有的参数都通过它来自动配置, 因此必须预先设定一些参数的值。滑动窗口 $slide_w$ 表示使用前 $slide_w$ 个历史数据来预测后继数据点。若 $slide_w$ 过小, 则预测依据的历史信息太少; 而 $slide_w$ 过大, 则导致权值连接增多, 学习过程的时间又会成倍增多。此外, BP 和 Elman 网络模型必须先确定网络的结构才可以和 Mop-IElman 方法进行比较, 所以, 有必要先确定滑动窗口 $slide_w$ 的大小即输入层神经网络个数。参照文献[8]以及经验, 本文设定滑动窗口 $slide_w=3$ 。因此所需配置的网络参数精简为 $\theta=(number_hidden, candidate1, candidate2)$ 。

图 2 给出了寻找最优网络配置 θ 的具体算法流程图 (图 2 中的样本数据均是经过归一化的失效数据), 而具体的实验步骤描述如下:

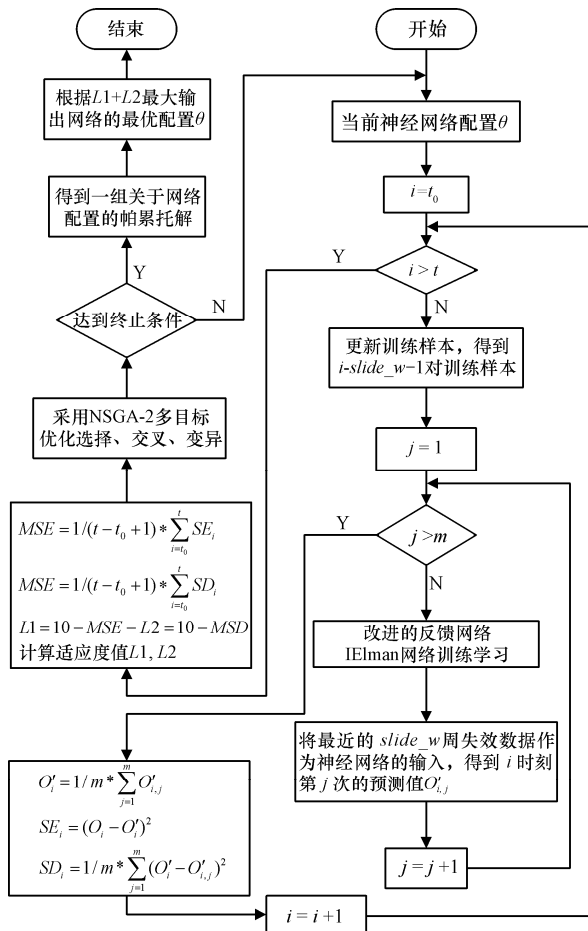


图 2 Mop-IElman 网络模型算法流程图

- 1) 采用 $O_i/maxd$ 将失效数据集归一化到 $[0,1]$, 其中, $maxd$ 是经验得到的最大累计失效数;
- 2) 对 θ 进行实数编码, 其中, $number_hidden$ 是介于 $[\min_hidden, \max_hidden]$ 的整数, 而 $candidate1$ 和 $candidate2$ 则介于 $[-1,1]$;
- 3) 生成初始种群大小 $P=30$, 设置遗传的代数 $MaxGen=50$;
- 4) 利用 NSGA-II 计算 2 个目标函数值, 进行多目标优化, 寻找帕累托解;
- 5) 虽然帕累托解集内的所有解没有优劣之分, 但以预测精度与顽健性之和达到最大为标准, 从帕累托解集里选出最后的解, 返回给网络, 作为网络的优化配置;
- 6) 将配置好的网络进行 m 次训练, 取其平均值作为每个数据点的预测值;
- 7) 最后, 将预测结果进行反归一化变换, 得到最终预测值。

3.3 实验研究

本节将两组实际工程的软件失效数据应用于 BP

前馈网络 (back propagation)、Elman 网络、以及以隐层节点个数为单目标或多目标优化的 Elman 网络, 包括单目标优化 (Sop, single-object optimization) 预测准确性函数 $L1$ (Sop-L1) 以及单目标优化网络的顽健性函数 $L2$ (Sop-L2), 多目标优化经典 Elman (Mop-Elman) 模型和 Mop-IElman 网络模型比较。

根据 Kolmogorov 定理, 传统网络的隐层节点个数为 $2 \times slide_w + 1$ ^[18]。BP 网络隐层和输出层激活函数均采用 sigmoid 型函数 tansig。Elman 和 IElman 反馈网络隐层也采用 tansig 函数, 但输出层采用线性激活函数 purelin。单目标或多目标优化 Elman 模型时, 隐层节点的个数介于 $[\min_hidden, \max_hidden]$ 之中。

取 $\min_hidden_node=3$ 是由于隐层节点个数一般不小于输入层 ($slide_w=3$)。若隐层节点个数的取值上限过大, 则会延长学习的过程。取 $\max_hidden_node=10$ 时, 算法的时间尚能接受; 若取更大则会导致网络在训练时产生过拟合, 并不一定能提高预测性能。

3.3.1 实验数据

采用软件可靠性工程中经典的 NTDS (naval tactical data system, 海军战术数据系统) 数据^[19] (见表 1) 和一组实际工程失效数据集 LCCS (large-scale command and control system, 大型指挥与控制系统)^[20] (见表 2) 作为实验数据。NTDS 数据是美国海军战术数据系统在开发、测试和用户 3 个阶段发现的错误数和故障间隔时间, 在这 3 个阶段分别发现 26、5 和 3 共 34 个错误, 以失效间隔时间形式给出。LCCS 数据是美国空军一个大型指挥与控制软件系统在 15 周测试时间中所观察到的每周失效数据, 以时间失效数的形式给出。

表 1 NTDS 失效间隔时间数据/日

失效	间隔	失效	间隔	失效	间隔	失效	间隔
1	9	10	7	19	6	27	87
2	12	11	1	20	1	28	47
3	11	12	6	21	11	29	12
4	4	13	1	22	33	30	9
5	7	14	9	19	6	31	135
6	2	15	4	23	7	32	258
7	5	16	1	24	91	33	16
8	8	17	3	25	2	34	35
9	5	18	3	26	1		

表2 LCCS 软件失效数据

周	失效数	周	失效数
1	203	9	80
2	136	10	64
3	183	11	27
4	47	12	42
5	46	13	55
6	71	14	62
7	54	15	11
8	57		

3.3.2 实验结果

表3是将Mop-IElman模型用于NTDS和LCCS两组数据得到的最终网络配置 θ 。实验结果如表4和表5所示,图3和图4对应的是NTDS的预测结果图以及误差图,而图5和图6则对应LCCS的预测结果图和误差图。

表3 NTDS 和 LCCS 数据对应 Mop-IElman 网络配置

Mop-IElman 网络配置	NTDS	LCCS
<i>number_hidden</i>	6	7
<i>candidate1</i>	[-0.737 327 55 0.906 557 76 -0.800 913 37 -0.710 415 15 0.882 400 84 -0.722 992 42]	[-0.499 208 83 0.241 988 84 0.586 916 52 -0.994 614 6 0.590 822 89 0.482 731 53 0.418 446 63]
<i>candidate2</i>	-0.292 128 65	-0.881 550 66

表4 NTDS: 各种模型与 Mop-IElman 对比实验结果

模型	<i>num- ber_hidden</i>	MSE	MSD	MSE+MSD
BP	7	4 248.087	162.307 9	4 410.395
Elman	7	4 005.097	306.552 2	4 311.649
Sop-L1	8	3 732.043	351.293 5	4 083.337
Sop-L2 (Mop-Elman)	6	4 059.219	113.452	4 172.671
Mop-IElman	6	3 719.806	108.902 1	3 828.708

表5 LCCS: 各种模型与 Mop-IElman 对比实验结果

模型	<i>num- ber_hidden</i>	MSE	MSD	MSE+MSD
BP	7	1 512.16	260.25	1 772.41
Elman (Sop-L1)	7	631.22	511.67	1 142.89
Sop-L2	6	963.79	269.31	1 233.10
Mop-Elman	8	781.03	386.80	1 167.83
Mop-IElman	7	827.12	230.99	1 058.11

3.3.3 结果分析

分析对NTDS系统的实验结果(如图3和图4所示),自第6个失效到第21个失效,各类模型在预测结果的准确度上和原始数据都非常相近。这是由于NTDS系统在这段时间内发生的失效情况和前5个失效间隔的时间相当,数据没有明显的异常,所以不论是BP网络,还是Elman,都能进行很好的预测;但自第22个失效开始,数据波动比较大,时而间隔很久才发生失效(258天),时而间隔很短发生失效(1天)。反馈网络因能动态自适应这种数据波动而进行调整,从而在后期表现出更好的预测性能。所以Mop-IElman方法不仅在前期,且在后期的预测点都非常接近原始失效数据。通常预测性能并不能仅从预测结果准确度一个角度评价,须结合模型的顽健性。从表4可知,虽然各类算法的MSE值比较接近,但Mop-IElman不仅获得了很好的预测准确性,其MSD值也是最小的,表示它每次预测的值是最稳定的。从整体性能考虑,Mop-IElman方法对NTDS数据表现出最优的预测性能。

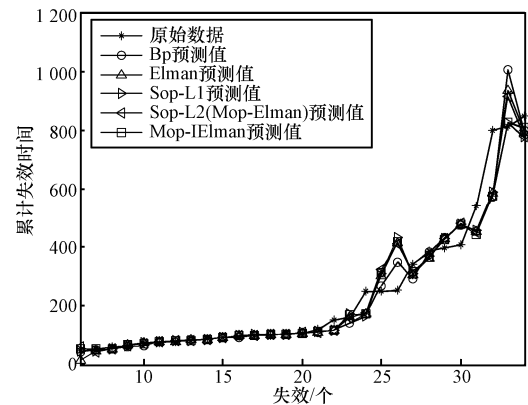


图3 各种模型预测 NTDS 数据的累计失效时间

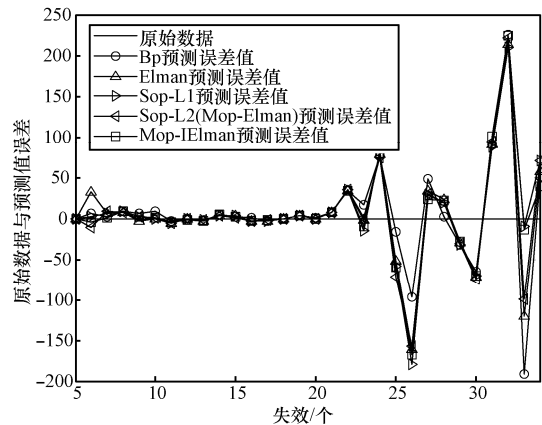


图4 各种模型预测 NTDS 数据的误差

分析图 5 和图 6, Mop-IElman 方法在 LCCS 数据上也显示出非常好的预测性能趋势。仅在第 6 周和第 11 周没有比其他模型好,但随着失效数据收集过程的不断增加,整体趋势表现最优。观察表 5, BP 网络的 MSE 值最大,说明在进行实际工程的软件可靠性预测时, Elman 网络要优于前馈网络。在顽健性 MSD 上, Elman 网络顽健性不高,结果的波动性大,每次预测结果随机性较强,因此可信性不足。BP 网络的顽健性最好而预测准确度最低也进一步说明了前馈网络较反馈网络不适于解决动态问题。Mop-IElman 则获得了较高的预测准确性和可信性。

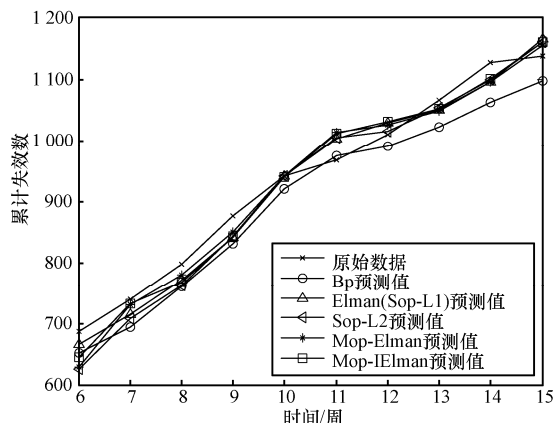


图 5 各种模型预测 LCCS 数据的累计失效数

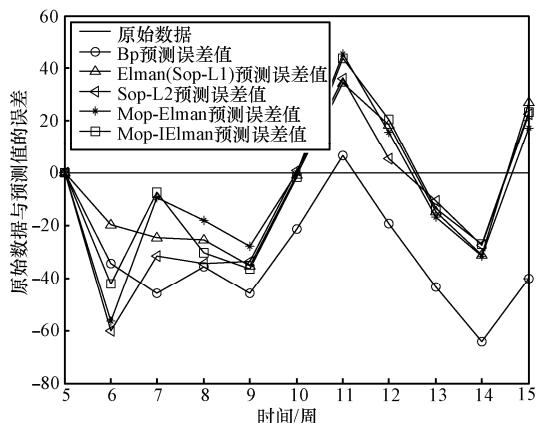


图 6 各种模型预测 LCCS 数据的误差

再者,从表 4 和表 5 中 Sop-L1 和 Sop-L2 的预测结果可知,若 Elman 网络单纯以隐层节点数为变量分别优化预测的准确性和顽健性,其隐层节点个数难以达到一致。此外,观察 Mop-Elman 的实验结果,在 NTDS 数据上,使用 NSGA-II 在 Elman 模型上的并行寻优,并没有明显改善预测性能,虽然获得了和 Sop-L2 同样的效果,预测结果较稳定,但精度不高,说明经典的 Elman 网络预测准确性不

如 IElman; 在 LCCS 数据上,预测结果的顽健性较 Sop-L1 和 Elman 都得到改善,精度也有所提高,但同 Mop-IElman 相比,模型预测结果的稳定性不够,整体性能不如 Mop-IElman。因此,难以实现在一维搜索空间进行 Elman 网络的多目标优化。Mop-IElman 通过增加状态层 2 对反馈神经网络输出层的反馈延迟,以隐层节点个数和 2 个状态层的初始值为变量,扩大了变量的搜索空间。实验显示 Mop-IElman 具有较好的网络顽健性,而且预测准确性也较高,其整体性能 MSE+MSD 也是最优的。这表明, Mop-IElman 网络整体性能优于 BP 网络、Elman 网络、单目标优化的 Elman 网络以及多目标优化的 Elman 网络。

由于采用 NSAG-II 进行搜索时,是通过增加状态层 2,以隐层节点个数和 2 个状态层的初始值为变量,扩大解搜索空间的方式来获取可能的更优解,那么值得对 Mop-IElman 方法的时间复杂度进行评估和分析。因为 NSGA-II 算法的时间复杂度是 $O(MN^2)$ ^[15],其中, M 为目标函数的数量, N 为种群规模。神经网络的时间复杂度是指数级别的。所以, Mop-IElman 方法的时间复杂度等于 NSGA-II 算法的时间复杂度乘于神经网络的时间复杂度。可见 Mop-IElman 方法是牺牲了大量的时间获取预测结果的提高。但是由于本文的两组数据的数据量很小,目标函数的个数仅 2 个,利用 Mop-IElman 方法进行失效预测时的时间相对于实际失效发生的间隔时间还是可以接受的。

4 结束语

通过实际软件工程的两组失效数据验证可看出,使用具有动态特性的 Elman 反馈网络比前馈网络更加适合于预测软件的可靠性。单纯依靠隐层节点数为变量,同时优化网络的预测精度和顽健性难以保证这 2 个性能指标同时获得优化。而通过采用隐层和输出层反馈改进 Elman 网络的结构和考虑 2 个状态层的初始值以扩大变量搜索空间,会更有利于优化网络的性能。因此,在此基础上提出的 Mop-IElman 方法能得到很好的网络预测精度和顽健性,预测结果准确性和可信性都较高。综合考虑预测准确性和可信性这 2 个要素,与前馈网络、Elman 网络、单目标优化的 Elman 网络以及多目标优化的 Elman 模型比较, Mop-IElman 网络具有最优的整体性能。

Mop-IElman 方法不足之处在于采用 NSGA-II 算

法进行全局搜索, 搜索空间维度较高, 导致算法较为费时, 今后有必要研究算法效率的提高方法, 去除一些冗余的计算; 对参数 m 、 t_0 、 $slide_w$, 出于加速搜索考虑, 需要预先设定; 若能提高算法的运行效率, 可适当增大 m 的取值突出模型的顽健性; 若实验数据较多, 则应适当增加 t_0 的取值; 权值初始值始终是影响网络学习过程的重要因素, 而初始值的选取是一个经典难题, 没有理论指导; NSGA-II 是多目标优化算法中的代表, 也可以采用其他的多目标优化算法; 实验所采用的数据量很小, 不能明显突出算法的优势, 需要考虑将其应用于大型软件工程; 单纯采用 MSD 作为衡量预测可信性的指标还是不够的。总之, 进行多目标优化神经网络模型使其应用于构建可信软件系统值得进一步深入研究。

致谢

本文的部分理论和实验工作得到了阿尔托大学智能电力电子实验室高晓智教授的极大帮助, 在此表示感谢。

参考文献:

- [1] LYU M. Handbook of Software Reliability Engineering[M]. New York: IEEE Computer Society Press, McGraw Hill, 1996.
- [2] LI S, YIN Q, GUO P. A hierarchical mixture of software reliability model for prediction[J]. Applied Mathematics and Computation, 2007, 185(2):1120-1130.
- [3] GUO P, LYU M. A case study on stacked generalization with software reliability growth modeling data[A]. Proceedings of the 8th International Conference on Neural Information Processing[C]. Shanghai, China, 2001.3763-3769.
- [4] XING F, GUO P, LYU M. A novel method for early software quality prediction based on support vector machine[A]. Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering[C]. Chicago, Illinois, USA, 2005.213-222.
- [5] GUO P, LYU M. A new approach to optical multilayer learning neural network[A]. Proceedings of the 2001 International Conference on Artificial Intelligence[C]. Las Vegas, Nevada, SUA, 2001.426-432.
- [6] KHOSHGOFTAAR T, SZABO R. Using neural networks to predict software faults during testing[J]. IEEE Transactions on Reliability, 1996, 45(3):456-462.
- [7] SU Y, HUANG C. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models[J]. Journal of Systems and Software, 2007, 80(4): 606-615
- [8] HU Q, XIE M, NG S. Robust recurrent neural network modeling for software fault detection and correct prediction[J]. Reliability Engineering and System Safety, 2007, 92(3): 332-340.
- [9] MIKAEL B. A guide to recurrent neural networks and back propagation[J]. Electrical Engineering, 2001, 2001(2): 1-10.
- [10] WU Y, WU Y. Stability analysis for recurrent neural networks with time-varying delay[J]. International Journal of Automation and Computing, 2009, 6(3):223-227.
- [11] LIU J, NIU D. A novel recurrent neural network forecasting model for power intelligence center[J]. Journal of Central South University of Technology, 2008, 5(15): 726-732.
- [12] GAO X, GAO X, OVASKA S. Trajectory control based on a modified Elman neural network[A]. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics[C]. Orlando, Florida, USA, 1997.2505-2510.
- [13] GAO X, GAO X, OVASKA S. A modified Elman neural network model with application to dynamical system identification[A]. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics[C]. Beijing, China, 1996.1376-1381.
- [14] GAO X, OVASKA S. Genetic algorithm training of elman neural network in motor fault detection[J]. Neural Computing and Applications, 2002, 11(1):37-44.
- [15] DEB K, PRATAP A, AGARWAL S. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182-197.
- [16] Kanpur genetic algorithms laboratory[EB/OL]. <http://www.iitk.ac.in/kangal/index.shtml>, 2010.
- [17] GOEL A, SAXENA S, BHANOT S. A fast learning algorithm for training feedforward neural networks[J]. International Journal of Systems Science, 2006, 37(10): 709-722.
- [18] KURKOVA V. Kolmogorov's theorem and multilayer neural networks[J]. Neural Networks, 1992, 5(3): 501-506.
- [19] 徐仁佐, 谢曼, 郑人杰. 软件可靠性模型及其应用[M]. 北京: 清华大学出版社, 南宁: 广西科学技术出版社, 1994.
XU R Z, XIE M, ZHENG R J. Software Reliability Model and Its Application[M]. Beijing: Tsinghua University Press, Nanning: Guangxi Science and Technology Press, 1994.
- [20] 黄锡滋. 软件可靠性、安全性与质量保证[M]. 北京: 电子工业出版社, 2002: 126-126.
HUANG X Z. Software Reliability, Safety and Quality Assurance[M]. Beijing: Electronic Industry Press, 2002. 126-126.

作者简介:



程绪超 (1986-), 女, 安徽池州人, 北京师范大学硕士生, 主要研究方向为软件可靠性。

陈新宇 (1976-), 男, 浙江宁海人, 博士, 博士后, 北京师范大学讲师, 主要研究方向为软件可靠性与分布式系统。

郭平 (1957-), 男, 山西洪洞人, 博士, 北京师范大学教授、博士生导师, 主要研究方向为计算智能与可信软件。