第32卷 第23期 计 算 机 工 程 2006年12月
Vol.32 No.23 Computer Engineering December 2006
·人工智能及识别技术· 文章编号：1000—3428(2006)23—0214—02 文献标识码：A 中图分类号：TP311

软件可靠性多模型综合的预测方法

吴 勤，侯朝桢，原菊梅
（北京理工大学信息科学技术学院自动控制系，北京 100081）

摘 要：针对软件可靠性工程领域中存在的不同可靠性模型预测相同软件结果不一致的问题，提出了一种用神经网络实现的软件可靠性多模型综合预测方法，能够有效地解决软件工程中的实际问题，通过实例分析验证了该方法的有效性。
关键词：软件可靠性预测；神经网络；多模型综合

Multi-model Prediction Method for Software Reliability

WU Qin, HOU Chaozhen, YUAN Jumei
(Department of Automatic Control, School of Information Science and Technology, Beijing Institute of Technology, Beijing 100081)

【Abstract】 In order to solve the problem which different results predicted by different reliability models for the same software in practice of reliability engineering of software, this paper presents a new method using neural network to get a multi-model prediction method for software reliability. This technique solves the problem in practice of reliability engineering of software more availably. It uses some data sets to demonstrate the approach's efficiency.
【Key words】 Software reliability prediction; Neural network; Multi-model

自1973年IEEE软件可靠性年会召开以来，软件可靠性成为IEEE、ACM、AIAA、MRI及其它学术、工业和政府部门的主要研究题目之一。软件及软件密集系统被大量使用，软件对现代军事装备发展起着越来越重要的作用。目前软件可靠性的研究也成为武器系统可靠性研究的热点之一。

作为软件可靠性研究领域中重要的一部分，软件可靠性评估主要是基于各种软件可靠性模型上进行，目前还没有被推荐为广泛应用的模型。由于现有的模型精度存在着局限性，即模型往往只对某一个或几个软件工程项目或其某一段数据能够达到较高的预测水平，使得模型潜藏着不健壮性。单个模型会不适应软件工程项目的细微变化而使预测误差急剧增大，在现今软件工程项目各种因素犬牙交错、复杂多变的情况下，会造成可靠性管理工程人员得出错误结论的后果，所以有必要对软件可靠性评估的方法进行进一步的研究，以使其解决软件可靠性模型的稳健性问题[1]。为解决这一问题，本文提出了一种基于神经网络的软件可靠性多模型综合预测方法。该方法可以综合各种可靠性模型特点，使不同类型和特性的模型互相补偿，能够更准确地预测软件的可靠性。

1 软件可靠性的多模型综合

1.1 可靠性多模型综合的特点[1,2]

目前的各种软件可靠性模性特点各异，其前提假设也各不相同，对于同一个软件工程项目，各模型对可靠性的估计差别比较大，因此有必要综合各种模型的评价结果，进行多模型的综合评价，以获得更准确的预测值。

实际的软件失效过程不能完全用某个模型来描述，很多软件的失效过程是分段满足某种模型的。这从一个侧面说明要想更准确地分析软件可靠性需要综合多模型来实现这一目标。文献[3]在实验的基础上认为，混合模型一般比单个分模型的效果要好，而且混合模型抗数据“噪声”能力强，即混合模型精度更高，稳健性更好。而具有高稳健性的高精度估计才是现实工程急需的，也是本文的目标。并且混合模型不但对短期预测有效，而且对长期预测效果更加明显。

1.2 可靠性多模型综合的原理

将现有的可靠性模型作为分模型，综合模型的形式如下：

$$M_s = W_{JM} \times M_{JM} + W_{GO} \times M_{GO} + W_{LV} \times M_{LV} \cdots W_n \times M_n + b \quad (1)$$

其中，M_S 是综合模型，M_{JM}, M_{GO}, M_{LV} …分别代表JM,GO,JM模型等；W_{JM}, W_{GO}, W_{LV} …分别代表各分模型对综合模型的贡献，可以理解为权值。如某模型特别适合于某一软件项目，W_i 相对较大。b代表修正误差。

2 基于神经网络的多模型综合预测

2.1 神经网络模型

根据式(1)的表达形式，关键在于求解 W_i 和b，而这是比较困难的，本文提出用前馈神经网络模型来解决这一问题。根据上述形式，构造前馈神经网络。该神经网络分为输入和输出层，输入层有n个神经元，分别代表用各种分模型计算的软件故障数。输出层有1个神经元，代表实际软件测试过程中测试得到的软件故障数目。其结构见图1。

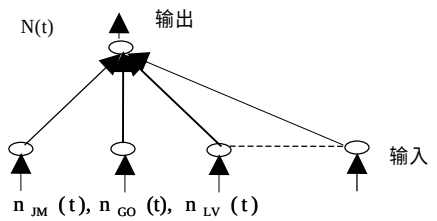

图1 多模型预测软件可靠性神经网络模型

基金项目：国防科工委技术基础基金资助项目
作者简介：吴 勤(1981－)，男，硕士生，主研方向：复杂武器系统可靠性分析；侯朝桢，教授、博导；原菊梅，讲师、博士生
收稿日期：2005-12-23 **E-mail：**wuqin-430@sohu.com

神经网络输出：

$$N(t)=n_{JM}(t)\times W_{JM}+n_{GO}(t)\times W_{GO}+n_{LV}(t)\times W_{LV}+\cdots n_n(t)\times W_n+b$$

其中 W_{JM}, W_{GO}, W_{LV} …,b 与 1.2 定义相同。N(t)是在 t 时刻软件测试得到的故障数目，$n_{JM}(t), n_{GO}(t), n_{LV}(t)$ 是在 t 时刻用各分模型估计得到的故障数的期望值，作为神经网络的输入。对照神经网络结构与式(1)可知，该神经网络与综合模型在结构上是等价的。由此可知，W_{JM}, W_{GO}, W_{LV} …就是训练好的神经网络的权值，b 就是神经网络的阈值。先通过测试得到的失效数据对神经网络进行训练，训练好的网络就可用于分析某时刻软件出现的故障数。

2.2 神经网络算法描述[4]

本文的神经网络采用 LMS 算法，基于梯度下降规则来减小网络的训练误差。具体算法步骤为：

(1)计算误差 $e(k) = t(k)-W_i \times p_i(k) - b$ ；

(2)权值修正公式 $\Delta Wi = 2\alpha \times e(k) \times p_i(k)$ ；

(3)阈值修正公式 $\Delta b = 2\alpha \times e(k)$ ；

(4)重复上述过程直到均方误差：

$$MSE = \frac{1}{N}\sum_{k=1}^{N} e(k)^2$$

达到所需精度时停止训练。

3 实例分析

本文选取 4 个软件可靠性模型作为分模型。分别为JM, GO,LV,YO这 4 种模型。它们是目前在可靠性分析中应用比较广泛的 4 种模型。其中JM模型是最具代表性的早期软件可靠性马尔可夫过程的数学模型；而GO和YO模型是最著名的NHPP过程模型；LV模型是典型的Baysian模型[5]。

选取丹麦信息与数学建模技术大学提供的教学数据包[6]中的某一软件失效数据作为实例进行分析。以其前 18 组数据作为训练数据，分别用各可靠性分模型计算此软件在 t_1 到 t_{18} 时故障数的期望值。分别记为 $n_{JM}(t_i), n_{Go}(t_i), n_{LV}(t_i), n_{YO}(t_i)$ (i=1,2,…,18)，以此作为神经网络的输入，实际测试的软件故障数记为 $N(t_1), N(t_2), ..., N(t_{18})$ ，作为神经网络的输出，训练该神经网络。训练好的神经网络可用于预测该软件在以后时刻的故障数。

剩余两组数据检验神经网络的预测效果，以 $n_{JM}(t_i), n_{Go}(t_i), n_{LV}(t_i), n_{YO}(t_i)$ (i=19,20)作为神经网络的输入，通过神经网络运算可得到在 t_{19}, t_{20} 时刻该软件的故障数。表 1 和表 2 分别给出该软件的训练数据与检验数据。从表 2 可以看出，网络预测得到的故障数与实际测试得到的故障数十分接近，其准确度要明显优于采用各分模型计算得到的结果。

表 1 训练数据

数据组	故障发现时间(天)	n_{JM}	n_{GO}	n_{LV}	n_{YO}	实际故障数 N
1	9	1.174	1.726	1.653	1.890	1
2	21	1.759	2.884	2.462	2.673	2
3	24	2.849	3.567	2.973	3.953	3
4	26	5.958	6.976	5.431	7.417	6
5	31	11.591	12.121	9.531	12.442	11
6	43	16.952	18.987	14.263	19.298	16
7	50	20.431	21.323	18.692	22.513	19
8	58	25.843	25.342	23.215	25.891	24
9	63	27.948	27.215	25.723	27.141	25
10	70	32.421	31.862	30.431	32.359	29
11	71	33.727	32.541	32.294	34.131	32
12	77	34.396	33.321	33.119	35.643	33
13	78	37.531	34.231	34.391	36.492	36
14	85	43.953	40.967	39.532	43.6534	42
15	91	49.362	45.543	44.371	48.964	47
16	92	52.634	50.452	46.834	51.326	49
17	103	53.732	54.343	49.792	53.641	52
18	120	57.496	57.653	53.527	56.751	55

表 2 检验数据

数据组	故障发现时间	n_{JM}	n_{GO}	n_{LV}	n_{YO}	神经网络计算结果	实际错误数 N
19	157	60.573	59.634	55.792	58.432	57.932	57
20	167	61.431	61.239	56.352	60.836	59.123	58

4 结论

本文针对软件可靠性工程中存在的实际问题，提出了一种基于神经网络的软件可靠性多模型综合预测方法。有以下几个特点：(1)很好地解决了软件可靠性模型与软件项目实际情况不符合的问题，有效地解决了软件可靠性模型存在的稳健性问题；(2)方法简单，构造的神经网络结构简单，其训练好的权值与阈值在综合模型中都有明确的物理意义；(3)对于软件的预测准确性相当高，特别适合于可靠性分析要求高的软件项目。本文的方法不仅适用于软件可靠性的预测，也可用在软件可靠性的评估上，为同类问题提出了一种新的思路。

参考文献

1 邹丰忠, 徐仁佐. 软件可靠性多模型综合评估[J]. 同济大学学报, 2002, 30(10): 1183-1185.

2 邹丰忠, 刘海青, 王　林. 软件可靠性综合模型[J]. 武汉大学学报, 2003, 36(1): 105-108.

3 Lyu M R, Nikora A. Applying Reliability Models More Effectively[J]. IEEE Software, 1992, 9(4): 45-52.

4 丛　爽. 神经网络、模糊系统及其在运动控制中的应用[M]. 合肥: 中国科学技术大学出版社, 2001.

5 Lyu M R. Handbook of Software Reliability Engineering[M]. McGraw-Hill Publishing, 1995.

6 Vladicescu F P. Performance Evaluation of Computers Courses [EB/OL]. http://www.imm.dtu.dk/popentiu/pec/pec.html.

（上接第 195 页）

参考文献

1 樊　莉. 人工智能中的 A*算法应用及编程[J]. 微机发展, 2003, 13 (5): 33-35.

2 Romein J W, Bal H E. Solving the Game of Awari Using Parallel Retrograde Analyisis[J]. IEEE Computer, 2003, 38(10), 26-33.

3 Bal H E, Casanova H, Dongarra J, et al. Blueprint for a New Computing Infrastructure (2nd Edition)[M]. Springer, 2002: 463-489.

4 王京辉. 基于 PVM 的启发式搜索的并行计算模型设计[J]. 计算机工程, 2005, 31(1): 68-70.

5 Karp A H. Porgramming for Parallism[J]. IEEE Computer, 1987, 20(5). 43-49.

6 Quinn M J. Parallel Computing — — Theory and Practice[M]. McGraw-Hill,1994.

7 鞠九滨, 杨　鲲, 徐高潮. 使用资源利用率作为负载平衡系统的负载指标[J]. 软件学报, 1996, 7(4): 238-243.