

ニューラルネットワークを用いたソフトウェアのテスト進捗度予測
に関する考察

篠原康秀*

今西学†

土肥正‡

尾崎俊治†

1. はじめに

近年、日常生活において使用されるシステムの多くはソフトウェアによって制御されるようになってきている。このように情報化された社会ではソフトウェアの故障が重大な事故の原因となる場合が少なくない。ソフトウェアの故障による事故を事前に予防するために、ソフトウェアの品質は信頼性、操作性、安全性などの様々な要因から評価される。特に、ソフトウェアの信頼性は代表的な品質評価要因として重要視されている。現在までに、ソフトウェアのテスト工程において発見されるフォールト数から信頼性を定量的に評価する方法として、ソフトウェア信頼度成長モデルが数多く提案されている [1]。

本研究では、時間計測型ソフトウェア信頼度成長モデルのひとつである Jelinski-Moranda モデル [2] (以下 JM モデル) に着目し、ニューラルネットワークを用いたテスト進捗度予測ツールを作成することを目指す。まず最初に JM モデルについて簡単な説明を行い、続いて、3層ニューラルネットワークを用いてフォールトの発見事象間隔を推定するためのノンパラメトリックモデルを提案する。最後に実際のテストデータを用いて、ソフトウェアテストの進捗度予測に関するパフォーマンスを比較し、本研究で示したニューラルネットワークモデルの有効性を示す。

2. 時間計測型ソフトウェア信頼度成長モデル

代表的な時間計測型ソフトウェア信頼度成長モデルである JM モデルについて説明する。このモデルは、ソフトウェア信頼度成長モデルの中でも古典的なものであるが、その単純な構造ゆえに拡張性が高く、実務上頻繁に用いられている。JM モデルでは、テスト工程におけるフォールト発見事象に関して次の仮定をおく。

- 1) 発生する各ソフトウェア故障に対する故障率は、その時点でのソフトウェア内に潜在するフォールト数に比例し、一定である。
- 2) 発見されたフォールトの修正時間は無視できるものとし、残存フォールト数はソフトウェア故障が発生するごとに1個だけ減少する。
- 3) ソフトウェア内に残存する全てのフォールトは、任意の時刻において同一のソフトウェア故障の発生可能性を有する。

これらの仮定から、故障時間間隔を表す確率変数 $X_i (i = 1, 2, \dots, N)$ に対する故障率は

$$z(x_i) = \phi[N - (i - 1)] \quad (1)$$

となる。ここで、

$N (> 0)$ = テスト開始前にソフトウェア内に潜在する総フォールト数、

$\phi (> 0)$ = 残存フォールト1個当たりの故障率

である。(1)式からソフトウェア信頼度及び MTBF (平均故障時間間隔) は次式で与えられる。

$$R(x_i) = \exp[-\phi(N - i + 1)x_i], \quad (2)$$

$$E[X_i] = 1/[\phi(N - i + 1)]. \quad (3)$$

3. ニューラルネットワークモデル

ソフトウェアのテスト工程において、ソフトウェアに内在しているフォールトは時系列データとして観測される。そこで本研究では、ソフトウェア内に存在していたフォールトが発見されてから次のフォールトが発見されるまでの時間をニューラルネットワークを用いて推定する。

ニューラルネットワークは、Fig. 1に示すように入力層、中間層、出力層の3層から構成される。各層は任意の数のユニットから構成されており、各ユニットの入出力応答はシグモイド関数

$$f(x) = 1/[1 + \exp\{-(x + \theta)\}] \quad (4)$$

により与えられる。ここで x は各ユニットの入力値であり、オフセット値 θ は生理学的根拠から導入された変数である。また、中間層の各ユニットと前後層の各ユニット間には結合係数が与えられている。

一般に、階層型ニューラルネットワークでは、誤差逆伝搬法 (バックプロパゲーション) を用いてネットワークの内部パラメータを変更し、入力データと出力データ間の関係を学習する。ネットワークの内部パラメータはシグモイド関数のパラメータであるしきい値 θ と結合係数である。

誤差逆伝搬法には、入力データと教師信号の2種類のデータが必要であり、入力データに対する認識を行う順方向の情報処理と、ネットワークの学習を行う逆方向の情報処理がある。順方向の情報処理は、前層の各ユニット

* 広島大学大学院工学研究課博士課程後期

† 広島大学大学院工学研究課博士課程前期

‡ 広島大学工学部第二類

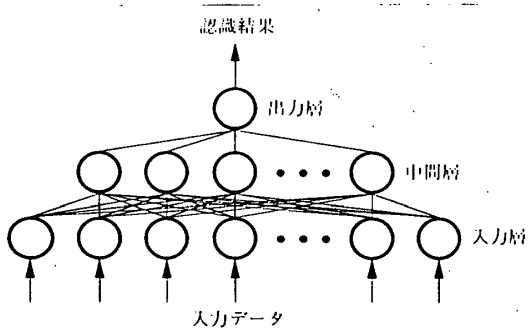


Figure 1: ニューラルネットワークの概念図.

Table 1: 平均 2 乗誤差.

データ区間	95 以前	95 以降	全データ
最尤推定 1	0.08082018	0.58681951	0.37598646
最尤推定 2	0.02662517	0.13870911	0.09200747
Neural Network	0.03798823	0.00853883	0.02080941

からの出力値に結合係数を乗じ、これらの和をとる。そしてこの値でのシグモイド関数値をユニットの出力値とする。一方、認識結果を教師信号に一致させるために、逆方向の情報処理において、出力値と教師信号との二乗誤差を求め、出力層から入力層まで二乗誤差を減らす方向に内部パラメータを順次変更する。

4. 数値実験

JM モデルと上述の 3 層ニューラルネットワークモデルに実際のテストデータ [3] を適用し、累積 MTBF を求める。ここで、JM モデルにおける MTBF の予測は最尤法により行う。特に、ここでは現時点までの全てのデータを用いた通常の最尤推定 (最尤推定 1) と、ニューラルネットワークと同じ入力データ数を用いた最尤推定 (最尤推定 2) の 2 つの方法を用いた。ニューラルネットワークモデルにおいては、教師信号に現時点までの実測値を与えてシミュレーションを行った。

Table 1 は予測結果に基づいた平均 2 乗誤差を各区間で平均して得られた値を表にしたものである。Fig. 2 では実測値と各予測法による予測値の間の平均 2 乗誤差のグラフを示したものである。Fig. 3 は各予測法に対する累積 MTBF を示したグラフである。

Fig. 2 および Fig. 3 において、最尤法による JM モデルの予測が、故障データ数 95 付近以降で急激に悪くなるのがわかる。一方、ニューラルネットワークモデルによる予測に関しては、データ数が増加しても比較的実測データに近い安定した予測結果が得られた。以上のことから、ニューラルネットワークを用いたソフトウェアテスト進捗度予測はパラメトリックモデルよりも有効であるといえる。

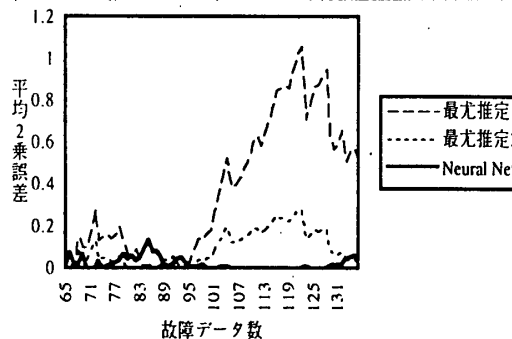


Figure 2: 平均 2 乗誤差のグラフ.

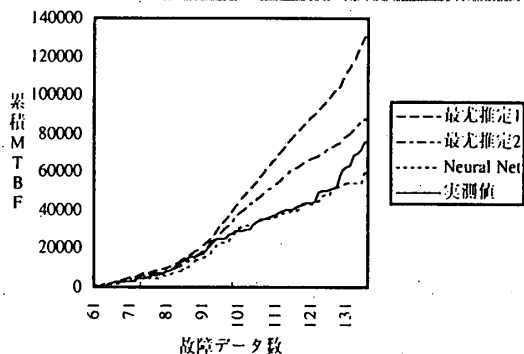


Figure 3: 各予測法に対する累積 MTBF.

5. 今後の課題

ニューラルネットワークモデルにおいて、データ数が十分に得られない場合、未学習の入力データに対して認識結果が補償されないという問題が生じる。この問題に対して、ニューラルネットワークに統計的手法を組み込むことにより、未学習データの認識に対して理論的補償を与える方法が提案されている。今後は、このような内部に統計的構造を有するニューラルネットワークモデルを構成し、実際にソフトウェアのテスト進捗度を評価する。さらに、2. で挙げた JM モデル以外の時間計測型ソフトウェア信頼度成長モデルに対するパフォーマンスの評価を行う予定である。

参考文献

- [1] 山田茂: ソフトウェア信頼性評価技術, HBJ 出版局, (1989).
- [2] Z. Jelinski and P. B. Moranda: "Software Reliability Research", in *Statistical Computer Performance Evaluation*, W. Freiberger (ed.), pp. 465-484, Academic Press, New York, (1972).
- [3] A. A. Abdel-Ghaly, P. Y. Chan and B. Littlewood: "Evaluation of Competing Software Reliability Predictions", *IEEE Trans. Software Eng.*, Vol. SE-12, No. 9. pp. 950-967, (1986).