

GMDH ネットワークによるソフトウェアのテスト進捗度に対する予測評価

篠原 康秀, 土肥 正, 尾崎 俊治

(広島大学工学部第二類 (電気系))

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件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行程において発見されるフォールト数からソフトウェアの信頼性を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].

従来の信頼度成長モデルに対し, 著者らは階層型ニューラルネットワークを用いた手法を提案し, その有効性を示した。一般に, 階層型ニューラルネットワークは複数の層とユニットによって構成され, その学習は誤差逆伝搬法を用いて行われる。しかし, 誤差逆伝搬法は層数とユニット数の最適な値を一意に決定できないため, ネットワークのサイズを予め数値実験によって決める必要がある。

一方, 時系列データの予測に GMDH (Group Method of Data Handling) を適用した研究が報告されている [2,3]。GMDH は, ニューラルネットワークと同様に複数の層とユニットから構成されているが, ネットワークを構成する総数とユニット数を最適な値に更新出来ることが特徴となっている。

そこで本稿では, ソフトウェアのテスト進捗度に関して, ニューラルネットワークと GMDH (以下 GMDH ネットワーク) を用いて, ソフトウェアの進捗度予測に関するパフォーマンスを比較することを試みる。

2. GMDH ネットワーク

GMDH ネットワークは, 1960 年代後半, A. G. Ivakhnenko らによって提唱された発見的自己組織化 (Heuristic Self-Organization) の原理に基づく手法である。発見的自己組織化とは, まず与えられた入力を元にランダムな結合を作り, 次にその結果発生する部分表現式から最良のものを選択するという過程を繰り返し行い, 最終的には得られた完全表現式を用いて, 要求されている予測値に対するシステムを構築する手続きを自動的に示すことである。

GMDH ネットワークは, システムの完全表現式として, Kolmogorov-Gabor の多項式

$$\phi = a_0 + \sum_i a_i x_i + \sum_i \sum_j a_{ij} x_i x_j + \dots \quad (1)$$

を想定し, システムの部分表現式として, 2 変数の 2 次多項式

$$y_d = b_0 + b_1 x_i + b_2 x_i^2 + b_3 x_i x_j + b_4 x_j^2 + b_5 x_j \quad (2)$$

を用いる。ここで, x_i と b_i はそれぞれ入力値と GMDH ネットワークの荷重の値を表す。

基本型 GMDH ネットワークを構成するユニットは Adaline と呼ばれており, 式 (2) で与えられた加重和を出力する 2 入力 1 出力ユニットである。基本型 GMDH ネットワークは Adaline ユニットの階層構造としてもち, 多入力 1 出力系システムを構成する。

ソフトウェアのテスト行程において, ソフトウェアに内在しているフォールトの発見時間間隔は時系列データとして観測される。そこで本稿では, 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 GMDH ネットワークを用いて推定する。ここで, 使用される GMDH ネットワークの学習アルゴリズムは以下のとおりである。

(ステップ 1): 原データに対し予備的処理を行い, ラベル付けを行う。偶数番号の原データをトレーニングデータ, 奇数番号の原データをセクションデータとする。GMDH ネットワークの処理ユニットの層数は mC_2 となる。ここで, m はデータ数である。

(ステップ 2): トレーニングデータを使用し, GMDH ネットワークの学習を行う。GMDH ネットワークの学習は, 式 (2) から得られた出力値と実測値との平均二乗誤差を計算し, その平均二乗誤差を最小にするように荷重の値を更新することである。平均二乗誤差が減少しなくなるまで, ステップ 2 を繰り返す。

(ステップ 3): GMDH ネットワークの全ての層に対し, ステップ 2 を実行する。

(ステップ 4): ステップ 3 が終了した時点での荷重とセクションデータを用いて, 現在の層の各ユニットにおける平均二乗誤差を計算する。各処理ユニットの平均二乗誤差と平均二乗誤差の最小値との比が, GMDH ネットワークのしきい値よりも大きい場合, その GMDH ネットワークを次候補として確定する。

(ステップ 5): 次に, 層数とユニット数を増加させた GMDH ネットワークを考える。その再構成された GMDH ネットワークに対し, ステップ 2 からステップ 4 までの処理を行う。再構成後の GMDH ネットワークの平均二乗誤差の最小値が次候補よりも小さいならば, その再構成後の GMDH ネットワークを次候補として確定する。ステップ 5 を繰り返し行い, 最終的に得られた GMDH ネットワークをシミュレーションに用いる。

3. 改良型 GMDH ネットワーク

基本型 GMDH では, 生き残った Adaline ユニットの数が最終的なネットワークのサイズに影響を与えることが指摘されている。また, 層数を増加させたネットワークには, 予測値の計算にはまったく関係の無い項が多量に含まれることが予想される。これは, GMDH ネットワークによる予測値の精度を低下させる直接の原因とな

る。よって、ここでは近藤、田村 [4] によって提案された改良型 GMDH ネットワークについて述べる。

目標値との相関が低い項を取り除き、部分表現式を自己選択する改良型 GMDH ネットワークを考える。部分表現式の選択には AIC を用いる。部分表現式 (2) の残差は、平均 0、分散 σ^2 の正規分布に従う確率変数とみなせるため、AIC は以下のように定義される。

$$AIC_m = N \log S_m^2 + 2(m+1) + C. \quad (3)$$

ここで、 S_m^2 は残差の二乗誤差、 $\hat{\phi}_a$ は回帰推定値、 N は入力データとその目標値のデータセット数、 C は m に無関係な定数を表す。

改良型 GMDH の学習は次のようにして行う。まず第 1 層において、 mC_2 個の Adaline ユニットのなかから AIC の値が小さいものを m_1 個選択する。これは、部分表現式 (2) を正規方程式で表現し、第 $(N+1)$ 列目までのすべての非対角要素を消去することで得られる。行列の大きさが 3 を越える場合には、Gauss-Jordan の消去法を用いる。

次に、第 2 層において、部分表現式を 2 段階の手続きにより処理する。ステップ 1 においては、第 1 層で自己選択された Adaline ユニットの出力値を入力変数とする部分表現式を構成し、自己選択を行う。ここで、部分表現式に含まれる項数が m に達した場合、まず、 $m-1$ 個の変数のなかから AIC の値の増加量が最も小さい変数を部分表現式から取り除き、部分表現式の中に含まれていない変数のなかから AIC の減少量が最も大きい変数を部分表現式の中に取り込む。これらの操作を AIC の値が減少しなくなるまで交互に繰り返す。取り除いた変数が部分表現式にすべて取り込まれたら学習を終了し、最終的に得られた部分表現式の中から m_1 個の変数を選択する。さらに、ステップ 2 において、ステップ 1 で選択された Adaline ユニットの出力値と入力データ x_i を入力変数とし、再度部分表現式を構成する。ステップ 1 と同様の手順で AIC の値を求め、 m 個の部分表現式を自己選択する。この段階において、 $(m_1 \times m)$ 個の部分表現式から、AIC の値が小さい Adaline ユニットを m_1 個選択し、次層の学習に移行する。

第 3 層目以降の学習は第 2 層と同様にして行う。ただし、自己選択される前の Adaline ユニット数は、必ず前層のものよりもひとつだけ少ないものとする。前層と現在層の間で AIC の値に変化がない場合、または、前層と現在層の部分表現式の構造が等しい場合には学習を終了し、前層までの Adaline ユニットを用いてシステム全体の部分表現式を構成する。

4. 比較実験と考察

ここでは、実際のソフトウェアテスト工程において観測された 135 個のフォールト発見時間データを使用し、逐次的に次のフォールト発見時刻を推定することを考える。本実験では、推定モデルとして、(i) 基本型 GMDH、(ii) 改良型 GMDH、(iii) 3 層ニューラルネットワーク、(iv) Jelinski and Moranda モデル、(v) Schick and Wolverton モデル、(vi) Moranda モデルをそれぞれ適用し、次降のフォールト発見時間間隔を予測する。

基本型 GMDH ネットワークにおいては、フォールト発見時間データの中から入力データ x_i ($i = 1, 2, \dots, 95$) を選び学習に使用する。95 C_2 個の Adaline ユニットのなかで平均二乗誤差が最も少ない $m_1 = 10$ 組の Adaline ユニットを残し、得られた荷重を用いてソフトウェアのフォールト発生時間間隔を推定する。一方、改良型 GMDH ネットワークにおいては、簡単のため式 (3) の任意パラメータ C を 0 とみなして各 m に対する AIC を求め、部分表現式を構成する。

ニューラルネットワークにおいては、BP 学習則の学習係数、層数、ユニット数、入力データとその目標値の

データセットについて様々な組合せを数値的に検討した結果、フォールト発見時間間隔に関する予測精度の最も良いものを採用する。実験に使用したニューラルネットワークは単純な 3 層ネットワークであり、入力層、隠れ層、出力層のユニット数は、それぞれ 10、10、1 と設定した。ニューラルネットワークの学習においては、フォールト発見時間データの中から連続した 11 個のデータを取りだし、最初の 10 個を入力データとして入力層ユニットに与え、残りのデータを入力データに対する目標値とし、学習係数を 0.5 とした。

Jelinski and Moranda モデル、Schick and Wolverton モデル、Moranda モデルにおいては、95 個のデータを用いて最尤推定を行い、各モデルのパラメータを決定した後には MTBF (mean time between failures) を次の予測値として求めた。実際のデータ数に対する累積フォールト発見時間の従属性を図 1 に示す。ここで、実際のデータ (REAL)、基本型 GMDH ネットワーク (GMDH-1)、改良型 GMDH ネットワーク (GMDH-2)、ニューラルネットワーク (NN)、Jelinski and Moranda モデル (JM)、Schick and Wolverton モデル (SW)、Moranda モデル (MR) である。

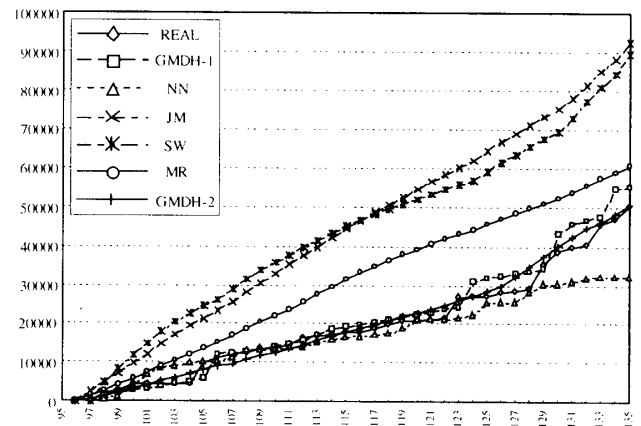


図 1: 累積 MTBF.

この図から、基本型 GMDH、改良型 GMDH、ソフトウェア信頼度成長モデルによる予測値は実データの値よりも大きく、常に楽観的な予測を行っていることが分かる。特に、改良型 GMDH は実行データに対して過剰な予測をすることなく、累積フォールト発見時間の異時的挙動を忠実に予測している。一方、ニューラルネットワークもかなり良好な予測結果を示していることが読みとれるが、実データに対して非観的な予測を行うことが判明した。

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] D. T. Pham and X. Liu: *Neural Networks for Identification, Prediction and Control*, Springer, London, (1995).
- [3] 池田三郎: “GMDH (変数組み合わせ計算法) の基礎と応用”, システムと制御, **23-12**, pp. 710-717, (1979).
- [4] 近藤正, 田村担之: “情報基準 AIC を用いて中間表現式を自己選択する改良形 GMDH”, 計測自動制御学会論文誌, vol. 15, pp. 466-471, (1979).