

# リカレントニューラルネットワークを用いた ソフトウェアフォールト発見数予測

非会員 東京都立大学 \*座間 渉 ZAMA Wataru  
05000041 東京都立大学 肖 霄 XIAO Xiao

## 1. はじめに

現在、システムの多くがソフトウェアによって構成されている。しかし、それらはバグなどに起因する不具合を引き起こすことがあり、種々の問題を引き起こす要因となっている。ソフトウェアの信頼性を定量的に評価することはこのような問題の解決策となり、システム開発者が現在および将来におけるソフトウェアの信頼性を正しく認識することが可能となる。そのような利点は、システム出荷時期やテストに必要な人員の正確な見積もりを実現する。本稿では、リカレントニューラルネットワーク (Recurrent Neural Network; RNN) を用いてソフトウェアフォールト発見数の時間的挙動を高精度に予測する手法の提案を行う。

## 2. 従来手法

ニューラルネットワークは、ある入力に対して期待される出力を予め学習させることにより予測を行うことができる。Wang and Zhang [1] は RNN を用いたフォールト数予測手法を提案した。時刻  $i$  における累積フォールト数の観測データ  $x_i$  ( $i = 1, 2, \dots, m$ ) が得られているとする。このとき、入力サイズが 3、出力サイズが 1 の RNN において、学習データを表 1 のように定める。これを学習させた後、時刻  $i = m + 1$  以降の累積フォールト数を出力させ、その値を予測値とすることができる。

## 3. 提案手法

Wang and Zhang [1] は累積フォールト数の観測データをそのまま RNN の学習データとしていた。しかし、観測データはその特徴量とは無関係な誤差項を含むと考えられるため、誤差項も学習してしまうことになる。そこで、本稿では誤差項を除去した推定値を学習データとして RNN に学習させることで、予測精度の向上を目指す。

### 3.1. 離散型非同次ポアソン過程モデル

ソフトウェア信頼性の振る舞いを記述するモデルとして、離散型非同次ポアソン過程モデル (Discrete Non-homogeneous Poisson Process; D-NHPP) があり、テ

入力データ	出力データ
$[x_1, x_2, x_3]$	$[x_4]$
$[x_2, x_3, x_4]$	$[x_5]$
$[x_3, x_4, x_5]$	$[x_6]$
$\vdots$	$\vdots$
$[x_{m-3}, x_{m-2}, x_{m-1}]$	$[x_m]$

スト時刻  $i$  における累積フォールト数  $N_i$  ( $i = 1, 2, \dots$ ) の確率関数は、

$$\Pr\{N_i = k\} = \frac{\{\Lambda_i\}^k}{k!} \exp\{-\Lambda_i\}, \quad k = 1, 2, \dots \quad (1)$$

と表される。ここで、 $\Lambda_i = E[N_i]$  ( $i = 1, 2, \dots$ ) は平均値関数と呼ばれる。また、 $\lambda_i = E[Y_i] = \Lambda_i - \Lambda_{i-1}$  ( $i = 0, 1, 2, \dots$ ) は強度関数とよばれ、時刻  $i$  におけるフォールト数  $Y_i$  ( $i = 1, 2, \dots$ ) の期待値である。なお、一般性を失うことなく、 $N_0 = Y_0 = \Lambda_0 = \lambda_0 = 0$  とする。

### 3.2. ウェーブレット縮小推定

ウェーブレット縮小推定 (Wavelet Shrinkage Estimation; WSE) [2] とは、D-NHPP に基づいたソフトウェア信頼性評価手法であり、解析対象関数がノイズを含むときそのノイズを除去し期待値を推定する。具体的には、 $Y_i$  ( $i = 1, 2, \dots$ ) はその期待値、すなわち強度関数  $\lambda_i$  と誤差項  $\eta_i$  を用いて、

$$Y_i = \lambda_i + \eta_i \quad (2)$$

と表すことができ、WSE は  $Y_i$  から誤差項  $\eta_i$  を除去し強度関数の推定値  $\hat{\lambda}_i$  を得る。  $Y_i$  の実現値  $y_i$  に対してウェーブレット変換を施し得られたウェーブレット係数は、解析対象関数の変化の情報を保持する。よって、これに対し適切な閾値処理を行うことで微小なノイズ成分を除去し、推定値  $\hat{\lambda}_i$  ( $i = 1, 2, \dots$ ) を得ることができる。

表 2: 提案手法の学習データ

入力データ	出力データ
$[\hat{\Lambda}_1, \hat{\Lambda}_2, \hat{\Lambda}_3]$	$[\hat{\Lambda}_4]$
$[\hat{\Lambda}_2, \hat{\Lambda}_3, \hat{\Lambda}_4]$	$[\hat{\Lambda}_5]$
$[\hat{\Lambda}_3, \hat{\Lambda}_4, \hat{\Lambda}_5]$	$[\hat{\Lambda}_6]$
$\vdots$	$\vdots$
$[\hat{\Lambda}_{m-3}, \hat{\Lambda}_{m-2}, \hat{\Lambda}_{m-1}]$	$[\hat{\Lambda}_m]$

表 3: PRMSE を用いた予測精度の比較

DS	予測手法	提案	従来
	DS1		4.30
DS2		2.49	2.88
DS3		5.84	6.74
DS4		9.55	9.62
DS5		2.06	2.97
DS6		2.97	3.12

### 3.3. 提案手法について

従来手法では累積フォールト数の観測データ  $x_i$  を学習データとしていたが、これらは 3.1 項の  $N_i$  の実現値とみなすことができる。本稿では WSE により観測データ  $x_i$  から平均値関数  $\Lambda_i$  を推定し、それを学習データとした予測を行う。まず、 $y_i = x_i - x_{i-1}$  ( $i = 1, 2, \dots, m$ ) の関係を用いてフォールト数の実現値  $y_i$  を算出し、WSE を用いて強度関数の推定値  $\hat{\lambda}_i$  を得る。続いて、 $\hat{\Lambda}_i = \sum_{k=1}^i \hat{\lambda}_k$  から平均値関数の推定値  $\hat{\Lambda}_i$  を算出し、これを学習データとして用いる。具体的に、学習データを表 2 のように定める。得られた平均値関数の予測値  $\hat{\Lambda}_i$  ( $i = m+1, m+2, \dots$ ) のそれぞれに  $\Delta = x_m - \hat{\Lambda}_m$  を足し合わせて最終的な予測値とする。ここで  $\Delta$  は時刻  $i = m$  までの誤差項の累積値を表し、累積フォールト数の観測値  $x_i$  ( $i = m+1, m+2, \dots$ ) には誤差が含まれるためにこれを足し合わせる。

### 4. 数値実験

ここでは、平均値関数の推定値とフォールト数の観測値をそれぞれ学習データとして得られた予測値の比較を行う。フォールト数のデータセットは [3][4] より引用した (DS1~DS6)。これらのデータセットのうち、テスト時刻が上位 10% (小数点以下切り捨て) のものを検証に用いるデータ (テストデータ) とし、残りのデータを学習データとする。学習データを用いて学習させた後、検証データに対応した時刻におけるフォールト数予測を行い、得られた予測値と観測値との比較を行う。この際、評価指標として予測二乗誤差 (Prediction Root Mean Squared Error; PRMSE) を用いる。

$$\text{PRMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (3)$$

ただし、 $n$  はテストデータの数、 $y_i$  は時刻  $i$  におけるフォールト数の観測値、 $\hat{y}_i$  は予測値である。PRMSE が小さいほど予測精度は良いとする。RNN の出力値は、

パラメータの初期化がランダムに行われることにより学習毎の予測値が異なるため計 500 回の学習と予測を行い、それぞれの予測値について PRMSE を算出する。表 3 に計 6 つのデータにおける、提案手法と従来手法の PRMSE の平均値を示す。いずれのデータセットにおいても、提案手法の PRMSE は従来手法の PRMSE と比較して小さくなることが確認された。

### 5. 結論と今後の課題

WSE によって得られた推定値を RNN に学習させた予測は、観測データを学習させた予測と比較して高い予測精度を示した。今後はより多くのデータセットを用いて検証を行っていく予定である。また、WSE や RNN のモデル、学習アルゴリズムにも多くの種類が存在する。これらが予測精度に与える影響についても調査を行うことが課題として挙げられる。

### 参考文献

- [1] J. Wang and C. Zhang, "Software reliability prediction using a deep learning model based on the RNN encoder-decoder," Reliability Engineering and System Safety, vol. 170, pp. 73-82, 2018.
- [2] X. Xiao and T. Dohi, "Wavelet Shrinkage Estimation for Non-Homogeneous Poisson Process Based Software Reliability Models," IEEE Transactions on Reliability, vol. 62, no. 1, pp. 211-225, 2013.
- [3] M. R. Lyu, Handbook of Software Reliability Engineering, IEEE Computer Society Press and McGraw-Hill Book Company, 1996.
- [4] Y. Tohma, H. Yamano, M. Ohba and R. Jacoby, "Parameter estimation of the hyper-geometric distribution model for real test/debug data," International Symposium on Software Reliability Engineering, pp. 28-34, 1991.