

(43) リカレントニューラルネットワークモデル による路面性状値予測の検討

奥田 知之¹・鈴木 康豊²・神武 直彦³

¹正会員 慶應義塾大学大学院 システムデザイン・マネジメント研究科
(〒223-8526 横浜市港北区日吉 4-1-1 協生館)
E-mail:t.okuda@keio.jp

²株式会社パスコ 中央事業部 技術センター インフラマネジメント部
(〒153-0043 東京都目黒区東山 1-1-2 東山ビル 5階)

³慶應義塾大学大学院 システムデザイン・マネジメント研究科
(〒223-8526 横浜市港北区日吉 4-1-1 協生館)

本研究では、ニューラルネットワークモデルの一種であり、不定間隔の時系列データを学習する事が出来るリカレントニューラルネットワークモデルにより路面性状値の予測を行い、舗装劣化予測へ適用する際の有効性を調べた。

複数の出力ユニットがあるニューラルネットワークにおいては、教師データの一部に欠損がある場合が問題になる。本研究では欠損データに対応する出力ユニットの誤差を 0 にすることで、正常に学習できる方法を構築した。そのモデルの 12 年後までの予測結果と、実測値との一致率をわだち掘れのランク評価で算出した。その結果 12 年後までの予測全てで実測との一致率が 50~60%の間になり、特に長期間の予測精度が低下しない結果が得られた。

Key Words: recurrent neural network, artificial intelligence, pavement condition servay, road management

1. 背景と目的

近年、少子高齢化の進展や経済成長の停滞と同時に、社会資本の老朽化が進行していることにより、限られた予算の中で適切な維持管理を実施する必要がある¹⁾。その一つの方策として予防保全が挙げられる。予防保全は、社会資本の劣化が進む前に最適な補修を行い、長寿命化する事でライフサイクルコストを削減するという考えである。最適な予防保全を実現するには、いつ、どこで、何に、どの様な補修を行うとどの程度ライフサイクルコストを削減できるか、将来を予測する事が不可欠である。

社会資本の中でも舗装は、現況を把握するために、ひび割れ・わだち掘れ・平坦性の 3 性状を測定・評価する路面性状値測定により、同一評価基準の定量データが 1970 年代から蓄積されてきた。その過年度の路面性状値を用いて、未調査区間の補間等のため、重回帰モデル²⁾や生成モデル³⁾によって舗装の劣化を予測する試みが行われてきた。それらの方法では、モデルを構築する際

に決める仮定に従ったデータであれば、比較的少ない学習データ数から最適なパラメータを決定する事が出来る。しかし、仮定に合わないデータや、複雑な非線形性を持つデータの特徴を学習する事は難しい。そこで本研究では少ない計算量で複雑な非線形データの特徴を学習する事が出来るニューラルネットワークモデル(以下 NN と称す)を舗装の劣化予測に用いることとした。

路面性状値は同じ路線を数年おきに計測した時系列データであるため、NN の中でも、特に過去の時系列を全て考慮して予測する事が出来るリカレントニューラルネットワークモデル(以下 RNN と称す)を用いた。RNN を用いる事で、過去の舗装の状態変化を全て考慮した上で、現実の様々な舗装の特徴を捉えて劣化予測を行うことが期待できる。

先行事例として路面性状値に対して NN を適用した例^{4,5)}および、アメリカで使われる PSI(Pavement Serviceability Index)という舗装の状態を表す指標に対して RNN を適用した例はあるが⁶⁾、いずれもデータの評価区間数が少な

く、NN の長所を活かせていない。また、路面性状値に対して RNN を適用した事例は無い。そこで本研究では、川崎市の平成 10 年～平成 26 年の路面性状値に対して重回帰モデル、NN、RNN を適用し、RNN の舗装劣化予測に対する有効性を評価した。路面性状値の 3 性状間は相互に関係がある事を考慮して、NN と RNN 各 1 つのモデルで 3 性状を学習した。その際に、路面性状値は NN や RNN が通常用いられている画像、音声や文章データ等とは異なり、欠損がある事が問題になる。そこで欠損データに対応する出力ユニットの誤差を 0 にすることで、欠損データ以外の出力ユニットについて正しく学習が行える方法を構築した。

2. 方法

(1) 使用データの準備

路面性状値には、ひび割れ・わだち掘れ・平坦性の 3 性状値以外にも大型車交通量などのデータも含まれる。今回使用したデータを表-1 に示した。

川崎市では基本的に 3 年間隔で同じ評価区間を調査しているので、12 年後までを 3 年間隔で予測を行うための学習データを作成した(図-1)。例えば 6 年後の予測を行うための学習データは、間の 1 回分の実測データを時系列から削除して、時系列データを作成した。9 年後の予測データは、実測の時系列から間の 2 回分のデータを削除して、時系列データを作成した。なお、予測したい間隔の実測データが存在しない場合は、その時系列データは作成しない。

元の有効なデータ数は 18,181 件だが、上記により各予測年数分の時系列データを作成した結果、NN が学習に使用したデータ数は 37,162 件、RNN が学習に使用したデータ数は 109,964 件である。そのうちおよそ 1 割は学習に使用せず、検証のみに使用するテストデータとした。

(2) 使用モデル

a) 重回帰モデル

重回帰モデルは、説明変数と目的変数の関係が線形であり、目的変数の分布は正規分布に従うと仮定したパラメトリックな回帰モデルである。説明変数を大型車交通量と補修後経過年数、目的変数をわだち掘れの年あたりの変化量 ΔRD_t とした。ある時刻 t のわだち掘れ RD_t は、次式(1)

$$RD_t = RD_{t-1} + \sum_r \Delta RD_r \quad (1)$$

のように、前の時刻のわだち掘れ RD_{t-1} に、予測した年あたりの変化量 ΔRD_r を足していく事で予測した。

b) ニューラルネットワークモデル

表-1 入力データと出力データ

(a) 入力データ		(b) 出力データ	
入力	前処理	出力	前処理
補修後経過年数 大型車交通量 CBR(路床の支持力比) 最終調査時路面性状 最終調査後経過年数	正規化	わだち掘れ ひび割れ 平坦性	正規化
車道幅員 調査年度 舗装厚			
交差点 構造物 補修工法		ダミー 変数化	

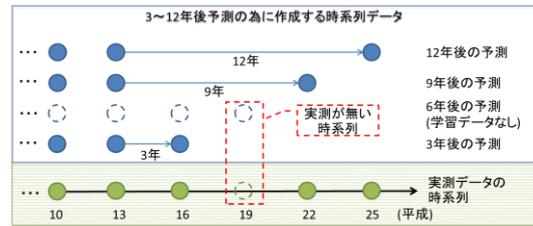


図-1 時系列データの作成方法

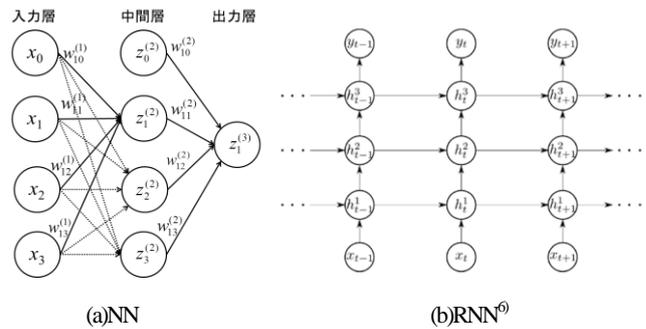


図-2 ニューラルネットワークモデル

NN を構成する 1 つのユニットはニューロンと呼ばれ、人の神経をモデル化したものである。それを並列にかつ多層に接続したモデルが NN である(図-2(a))。n 層目の j 番目のユニットの出力 z_j^n は次式(2)のように計算される。

$$z_j^n = g(a_j^n) \quad (2)$$

ただし、 a_j^n は活性と呼ばれ、

$$a_j^n = \sum_i w_{ij} z_i^{n-1} \quad (3)$$

w_{ij} は、前の層の出力 z_i^{n-1} から、次の層 z_j^n への重みベクトル(パラメータ)である。 $g(x)$ は活性化関数と呼ばれ、次式(4)で表される正規化線形関数を使用した。

$$g(x) = \max(0, x) \quad (4)$$

ニューロンの出力 z_j^n は次の層の入力となる。この計算を入力側 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ から出力側 $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ まで順に行う事を順伝播と呼ぶ。誤差関数には、順伝播の結果と教師データ(実測データ)の 2 乗誤差を用いる。教師データ

を $\mathbf{d}=\{d_1, d_2, \dots, d_r\}$ とすると、誤差関数 E_n は次式(5)で表される。

$$E_n = \frac{1}{2} \|\mathbf{y} - \mathbf{d}\|^2 \quad (5)$$

この2乗誤差を次式(6)の逆誤差伝播公式により、出力層から入力層へ逆方向に伝播する事で、重みベクトルの更新量を求める。

$$\delta_j^{n-1} = g'(a_j^{n-1}) \sum_k w_{kj} \delta_k \quad (6)$$

ここで、 δ_j^n は誤差と呼ばれ、

$$\delta_j^n \equiv \frac{\partial E_n}{\partial a_j^n} \quad (7)$$

とする。

本研究で用いた NN は、学習率や正則化パラメータを調整し、過学習が発生していない事を確認した上で(3(3)参照)中間層は3層、各中間層のニューロン数は50とした。また確率的勾配降下法のアルゴリズムは ADAM⁹⁾を使用した。NN, RNN の構築には NN フレームワークの Chainer⁹⁾を使用した。

c) リカレントニューラルネットワークモデル

一般に、点検データは同じ区間を不定間隔に点検したデータが時系列の履歴として残る。不定間隔の時系列を考慮できる NN に RNN がある。

RNN の概念図を図-2(b)に示した。 \mathbf{x}_t は時系列 t における入力データ、 \mathbf{h}_t^n は n 層のニューロンの内部状態、 y_t は出力を表す。

$$\mathbf{h}_t^n = g(W\mathbf{x}_t + U\mathbf{h}_{t-1}^n) \quad (8)$$

ここで、 U は前の内部状態 \mathbf{h}_{t-1} から \mathbf{h}_t への重みベクトルを表す。あるニューロンの内部状態 \mathbf{h}_t^n は、入力 \mathbf{x}_t と過去のニューロンの内部状態 \mathbf{h}_{t-1}^n から決まる。それにより、過去全ての状態を考慮して、現在の結果を算出する事ができる。各ニューロンには GRU(Gated Recurrent Unit)¹⁰⁾を使用した。RNN の中間層は3層、中間層のニューロン数は50である。

(3) 教師データに欠損がある NN と RNN の学習

通常、教師データ \mathbf{d} の何れかに欠損がある場合、そのデータは学習に使わないか、期待値最大化法等で補完される。本研究では欠損があるデータも効果的に学習で使用するために、次のような方法を提案する。例えば d_r が欠損だとすると、(5)式を計算する前に

$$d_r = y_r \quad (9)$$

とする。これにより、誤差関数に2乗誤差(5)式を使用し、出力層には出力層の活性をそのまま出力値とする恒等写像を用いた場合、出力層の y_r についての誤差(7)式は

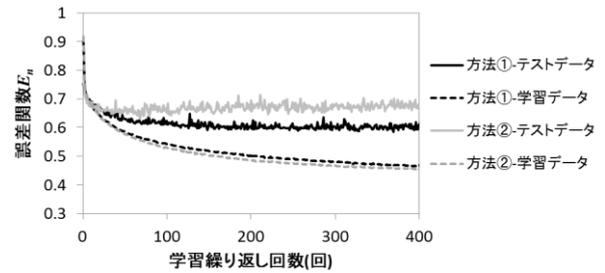


図-3 誤差関数 E_n の変化

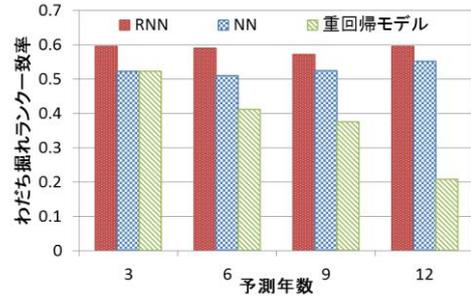


図-4 ランク一致率の比較

$$\delta_r^n \equiv \frac{\partial E_n}{\partial y_r} = y_r - d_r = 0 \quad (10)$$

となる。これにより欠損データに関連する重みは更新しないが、その他の重みだけを更新する事が可能である。

NN において、(9)式によって誤差を0にした方法①と、比較のため、教師データ欠損 d_r の値を0にした方法②の、学習繰り返し回数に対する誤差関数 E_n の変化を図-3に示した。図より、テストデータにおいて方法①の方が E_n が減少しており、教師データに欠損があっても正しく学習できていると言える。また、テストデータの E_n が増加していない事から過学習が発生していないこと、 E_n が一定になる事から学習が収束している事が解る。

3. 結果と考察

(1) ランク一致率

NN と RNN ではわだち掘れ・ひび割れ・平坦性3性状の予測を同時に行っているが、今回は経年変化が比較的定常的に進むわだち掘れの結果に着目して比較した。実測と予測でわだち掘れを5mmごとのランク評価に換算し、その一致率を算出した結果を図-4に示した。横軸の予測年数は、前回の調査(最終調査)から経過した年数を表す。重回帰モデルでは、3年後予測では50%以上の一致率があるものの、年数が経過するごとに一致率が低下し12年後予測ではおよそ20%程度になっている。それと比較し、NN と RNN は予測年数が3~12年全てにお

表-2 学習時系列数ごとのわだち掘れランカー一致率

学習時系列数	ランカー一致率	時系列学習データ数
1	0.578	12,424
2	0.584	10,985
3	0.607	7,549
4	0.632	4,461
5	0.627	1,743

いてランカー一致率が50~60%から低下しない事が特徴的である。重回帰分析では目的変数を正規分布、目的変数と説明変数間の関係は線形と仮定し、全ての学習データを使って説明変数ごとの重回帰係数を1つのみ学習する。従って、重回帰分析で算出する年あたりのわだち掘れ変化量が、実測と合わない評価区間は多い。したがって、実測と合わないわだち掘れ変化量を年数が進行するごとに加算することになり、実測との差が増加し、ランカー一致率が低下すると考えられる。一方、NN および RNN は各評価区間の特徴を学習し、それに基づき予測した結果、ランカー一致率が低下しなかったと考えられる。

(2) RNNの学習時系列数とランカー一致率の関係

結果(1)で NN よりも RNN のランカー一致率が高かった。その理由として考えられるのは、RNN では過去の時系列を考慮している事である。そのため、予測の際に与えられた時系列データ数が多いほど精度が良くなる事が考えられる。表-2 に RNN で予測する際に与えられた時系列数と、ランカー一致率及び、その時系列数の学習に使用したデータ数を示す。学習時系列数が4までは、時系列数が多いほどランカー一致率がおおよそ2%ずつ向上している。学習時系列数が5の場合、学習時系列数4と比較してランカー一致率が低下しているが、時系列数が5のデータ数が少ないためだと思われる。従って学習時系列数がランカー一致率の向上に寄与しており、RNN は NN よりもランカー一致率が高い要因だと考えられる。

4. まとめと今後の課題

今回得られた知見は次の通りである。

- NN および RNN において、教師データに欠損があった場合でも正常に学習が行えることが解った。
- NN と RNN は重回帰モデルと比較し、各評価区間ごとに特徴を捉えて予測をすることで、3~12年後の予測において、ランカー一致率が50~60%から低下しない事が非常に特徴的であった。
- RNN と NN を比較すると、RNN は過去の時系列データを考慮する事で NN より精度高く予測できることが解った。

今後の課題としては、現状ではモデル構造や学習パラメータの最適化が十分では無いため、より最適なモデルを構築する必要がある。特に学習速度と精度の面で非常に効果が高い Batch Normalization を RNN へ導入する方法については確立されておらず、検討が必要である。

また、他地域は路面性状値の特徴が異なると考えられる一方、NN および RNN は学習データが増加するほど精度を向上させる事が出来るので、同一のモデルを用いて複数地域の路面性状値や気象情報などの多様なデータを学習し、精度を向上する事が出来るのか確認する。その上で最終的には最適な予防保全計画立案への適用を検討したい。

謝辞：本研究を遂行するに当たり、川崎市建設緑政局の貴重なデータを使用させていただいた。ここに記して謝意を表する。

参考文献

- 1) 国土交通省：国土交通白書<2016> 平成 27 年度年度次報告, p156, 2016.
- 2) 森修二, 岳本秀人, 丸山紀美雄：積雪寒冷地における舗装メンテナンスに向けた路面性状予測について, 北海道開発土木研究所月報 No.598, 2003.
- 3) 貝戸清之, 小林潔司, 青木一也, 松岡弘大：混合マルコフ劣化ハザードモデルの階層ベイズ推計, 土木学会論文集 D3 (土木計画学), 68(4), pp.255-271, 2012.
- 4) 堀木賢一, 福田正：ニューラルネットワークによる舗装のパフォーマンスモデル, 土木学会論文集, 496, pp.99-102, 1994.
- 5) 重原大二朗, 西澤辰男, 中源達雄, 平野誠志：ニューラルネットワークによる北陸地方のアスファルト舗装わだち掘れ進行モデル, 土木学会舗装工学論文集, 13, pp25-30, 2008.
- 6) Tabatabaee, N., Ziyadi, M., and Shafahi, Y.: Two-Stage Support Vector Classifier and Recurrent Neural Network Predictor for Pavement Performance Modeling, *Journal of Infrastructure Systems*, Vol.19, No.3, pp.266-274, 2013.
- 7) Chung, J., Gulcehre, C., Cho, K., and Bengio, Y.: Gated feedback recurrent neural networks., *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Vol.37, pp.2067-2075, 2015.
- 8) Kingma, D. P., and Ba, J. L.: Adam: a Method for Stochastic Optimization., *International Conference on Learning Representations*, 2015.
- 9) Tokui, S., Oono, K., Hido, S., and Clayton, J.: Chainer: a Next-Generation Open Source Framework for Deep Learning., *Nips*, 2015.
- 10) Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1724-1734, 2014.