感度調整学習を用いたリザバネットの構築

大分大学　○吉岡晴海　松木俊貴　柴田克成

Construction of Reservoir Network Using Sensitivity Adjustment Learning

Harumi Yoshioka, Toshitaka Matsuki and Katsunari Shibata, Oita University

Abstract: Reservoir networks have an ability to learn time-series data processing easily. However, it has limitations for complex problems because it does not learn and fixes the weight values inside the reservoir. We expect to improve the performance for RNs as well by learning the entire network like other recurrent neural network(RNN) such as Long-Short Term Memory. However, update the weights in the reservoir through learning may cause a decrease in activity and reduce the learning performance. To prevent the decrease, we are considering to apply our proposed Sensitivity Adjusted Learning(SAL) in parallel with the original learning. SAL updates the values of the weights to adjust a local index called sensitivity in each neuron. In this paper, we used a recurrent neural network trained with SAL instead of a reservoir, and the learning performance was compared as a prelude to parallel learning. The results confirm that the learning performance is comparable to that of a reservoir whose weights are set directly.

1　序論

さまざまな分野で Deep Learning が従来の手法を圧倒しその有用性を示している [1]。音声認識や自然言語処理などの時系列データを扱うケースでは、リカレントニューラルネットワーク (RNN) が用いられる [2]。通常のニューラルネットワークの学習には、目標値と出力との誤差を出力から入力に向けて伝播する Back Propagation（BP）が用いられる。RNN はフィードバックを行うため、誤差もフィードバック部分を、時間をさかのぼって伝播する Back Propagation Through Time（BPTT）がよく用られる。しかし、RNN はフィードバックを繰り返すたびに重み値が乗算されるため、入力信号や学習に用いる誤差が小さくなり消失するか大きくなり発散し、うまく学習ができなくなる勾配消失（発散）問題が起こりうる。その問題に対して、Long-Short Term Memory（LSTM）[3] では、過去のデータを線形和で保持する仕組みを備えることで、勾配消失問題を回避している。

一方、Liquid State Machine[4] や Echo State Network[5] のような Reservoir Network(RN) を用いる場合もある。RN が有するリザバは、ランダムに決められ固定された重み値で、相互結合をしたニューロンで構成された RNN である。リザバは、内部ダイナミクスの中に入力情報を長期間保持できることから、リザバから必要な情報を取り出すための出力層の重み値を学習するだけで、より簡単に時系列データ処理を学習することができる [6]。しかし、RN では、リザバの重み値を固定して学習を行わないため、複雑な問題に対しては限界が指摘されている [7]。LSTM などの RNN では学習することによって、適切に時系列データを処理できるようになる。そのため、RN においてもリードアウトだけでなくリザバ内部も学習させることで、現状より良いパフォーマンスが期待できるのではないかと考えた。しかし、RN はリザバ内部を学習させると、その長所である記憶機能の劣化や、カオス性もしくは活性の低減が起こり、むしろ学習に悪影響を与えることが想定されるため、リザバ内部は学習させないという前提であることが一般的である。

この問題を払拭する手立てとして、われわれが提唱した感度調整学習を用いることで、リザバ内のカオス性や活性を維持できれば、リザバ内部の学習が可能となることを期待する。感度調整学習とは、ニューロンごとの入力の微小変化に対する出力の変化の割合である「感度」という、ローカルな指標を基準として重み値を更新する学習法である。先行研究では、RNN に感度調整学習を適用することでカオスダイナミクスを持たせられることを確認した [8]。

そこで、リザバに感度調整学習を適用することで、ダイナミクスを維持しながらリザバ内部を並行して学習できるのではないかと考えた。本論文では、2 つの学習を並行して行う前段階として、疎に相互結合をしたニューロンで構成された RNN にまず感度調整学習を適用したものを、リザバとして重み値を固定して、従来の RN と同様に学習する。そして、従来の方法で作成したリザバとの比較を行うことで、学習によって得られたネットワークをリザバとして使えるかどうかを確認した。

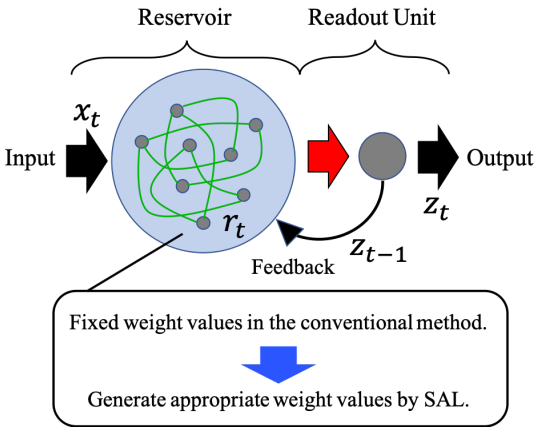


Fig. 1: The network architectures of "Reservoir Network" or "Recurrent Neural Network using Sensitivity Adjustment Learning".

2 研究方法

2.1 ネットワーク

Fig.1 に本研究で用いる Reservoir Network(RN) と、感度調整学習を施す RNN のネットワーク構造を示す。入力はリザバに入り、リザバ内ニューロンの出力は出力層であるリードアウトユニット（RU）へと送られる。RU はリザバの出力から最終的な出力を計算する。リザバ内は $N = 200$ 個のニューロンで構成され、結合確率 $p = 0.2$ で疎結合している。ある時刻 t でのリザバ内ニューロンの内部状態ベクトル $\boldsymbol{u_t}$ は次式により与えられる、

$$\boldsymbol{u}_t = \lambda \boldsymbol{W}^{\mathrm{rec}} \boldsymbol{r}_{t-1} + \boldsymbol{w}^{\mathrm{in}} x_t + \boldsymbol{w}^{\mathrm{fb}} z_{t-1} \tag{1}$$

ここで、$\boldsymbol{W}^{\mathrm{rec}}$ は、リザバニューロンの相互結合重み値行列であり、$\boldsymbol{r}_t$ はリザバ内部のニューロンの出力である。$\boldsymbol{W}^{\mathrm{rec}}$ の値は平均 0、分散 $1/pN$ のガウス分布によりランダムに決定した後、スペクトル半径 $\rho(\boldsymbol{W}^{\mathrm{rec}})$ で割る。λ はリザバニューロンの相互結合重み値のスケールを決めるパラメータであり、この値が大きいほどネットワーク内部のダイナミクスはよりカオティックになる。入出力が 1 個ずつであるため、$\boldsymbol{w}^{\mathrm{in}}$ はリザバのニューロンへの入力重み値ベクトルであり、$\boldsymbol{w}^{\mathrm{fb}}$ は RU の出力 z_t とリザバ内部のニューロンとの結合重み値ベクトルである。$\boldsymbol{w}^{\mathrm{in}}$ と $\boldsymbol{w}^{\mathrm{fb}}$ の値は -0.5 から 0.5 の一様乱数によってランダムに決定する。

リザバ内の全ニューロンの出力は、リードアウトのニューロンと全結合しており、出力層の出力 z_t は、リザバ内の全ニューロンへとフィードバックされる。入力 x_t は、リザバ内ニューロンに全結合している。また、リザバの全てのニューロンの活性化関数は tanh 関数であり、リザバの出力 $\boldsymbol{r}_t$ は次の式で求める。

$$\boldsymbol{r}_t = f(\boldsymbol{u}_t) = \tanh(\boldsymbol{u}_t) \tag{2}$$

RU により最終的な出力は次のように計算される。

$$z_t = \boldsymbol{w}^{out} \cdot \boldsymbol{r}_t \tag{3}$$

2.2 学習方法

本研究では、感度調整学習で、Fig.1 に示すネットワークの緑色で示す相互結合の重み値を更新し、ネットワークが情報を長期保持できるダイナミクスを獲得する。その後、FORCE Learning と呼ばれる学習法で、Fig.1 に示すネットワークの赤い矢印部分の重み値のみを更新する。

2.2.1 感度調整学習

RNN の学習に用いる感度調整学習の手順を以下に示す。j 番目のニューロンの学習の指標となる感度 $S_{j,t}$ は、各ニューロンの相互結合重み値の大きさと出力の微分を用いて、次の式で計算する。

$$\begin{aligned} S_{j,t} &= |\nabla_{\boldsymbol{r}_{t-1}} r_{j,t}| \\ &= \sqrt{\sum_j \left(\frac{\partial r_{j,t}}{\partial \boldsymbol{r}_{t-1}} \right)^2} \\ &= f'(u_{j,t}) |\boldsymbol{w}^{rec}_{j,t}| \end{aligned} \tag{4}$$

$$\boldsymbol{W}^{rec} = \begin{pmatrix} \boldsymbol{w}^{rec}_1 & \cdots & \boldsymbol{w}^{rec}_n \end{pmatrix}^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{w}^{rec}_j = \begin{pmatrix} w^{rec}_{j,1} & \cdots & w^{rec}_{j,n} \end{pmatrix}^{\mathrm{T}} \tag{6}$$

この $S_{j,t}$ が 1 より大きければ入力の変化より出力の変化が大きく拡大していき、1 より小さければ入力の変化に対して出力の変化が小さくなり収束をしていく。次に、感度 $S_{j,t}$ の時間平均 $\bar{S}_{j,t}$ を求める。

$$\bar{S}_{j,t} = (1-a)\,\bar{S}_{j,t-1} + a S_{j,t} \tag{7}$$

ここで、$a = 0.01$ とした。

次の式を用いて、相互結合重み値の更新量を求め、山登り法で重み値を更新する。

$$\begin{aligned} \Delta \boldsymbol{w}^{rec}_{j,t} &= \eta_s (1 - r^2_{j,t}) \\ &\quad \times \left(\frac{\boldsymbol{w}^{rec}_{j,t}}{|\boldsymbol{w}^{rec}_{j,t}|} - 2 r_{j,t} |\boldsymbol{w}^{rec}_{j,t}| \boldsymbol{r}_{t-1} \right) \end{aligned} \tag{8}$$

$$\boldsymbol{w}^{rec}_{j,t} = \boldsymbol{w}^{rec}_{j,t-1} + \Delta \boldsymbol{w}^{rec}_{j,t} \tag{9}$$

この時、学習係数 $\eta_s = 1 \times 10^{-4}$ とし、以上の学習を感度の時間平均 $\bar{S}_{j,t}$ が目標感度を超えるまで行う。一度目標感度を超えたニューロンは、それ以降は重み値を固定する。

2.2.2 FORCE Learning

リードアウトの学習に用いる FORCE Learning[6] の学習手順を示す。ある時刻 t の目標出力 T と実際のリードアウト出力 z_t との誤差を出す。

$$e_t = T_t - z_t \tag{10}$$

e_t とリザバ出力 $\boldsymbol{r}_t$ を用いて、次の式で重み値の更新量 $\Delta \boldsymbol{w}_t^{out}$ を決める。

$$\Delta \boldsymbol{w}_t^{out} = -\eta_f e_t \boldsymbol{P}_t \boldsymbol{r}_t \tag{11}$$

$$\boldsymbol{P}_t = \boldsymbol{P}_{t-1} - \frac{\boldsymbol{P}_{t-1}\boldsymbol{r}_t\boldsymbol{r}_t^T\boldsymbol{P}_{t-1}}{1+\boldsymbol{r}_t^T\boldsymbol{P}_{t-1}\boldsymbol{r}_t} \tag{12}$$

ここでは文献 [6] に基づき、学習係数 $\eta_f = 0.4$ とし、$\boldsymbol{P}_t$ はニューロンの相関行列の逆行列で、初期値 $\boldsymbol{P}_0$ は $\boldsymbol{I}_N\mu^{-1}$ とした。$\boldsymbol{I}_N$ は N 次の単位行列で、$\mu = 1.5$ である。$\Delta \boldsymbol{w}_t^{out}$ を用いて次式のように重み値の更新を行う。

$$\boldsymbol{w}_t^{out} = \boldsymbol{w}_{t-1}^{out} + \Delta \boldsymbol{w}_t^{out} \tag{13}$$

2.3 タスク設定

ネットワークの学習性能を検証するため、RNN の記憶容量や計算能力を評価するためによく使用され、記憶が必要な NARMA 10 Time Series と呼ばれるタスクを用いて学習の性能を比較する。NARMA とは Nonlinear AutoRegressive Moving Average（非線形自己回帰移動平均）の略である。10 Time Series とあるように、このタスクで正しい出力をするためには、過去 10step 分の出力と 10step 前と 1step 前の入力が必要となる。ある時刻 t での出力は以下の式で求められる [9]。

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta y_{t-1} \sum_{i=1}^{10} y_{t-i} \\ &\quad + \gamma x_{t-10} x_{t-1} + \delta \end{aligned} \tag{14}$$

ここで、$\alpha = 0.3$、$\beta = 0.05$、$\gamma = 1.5$、$\delta = 0.1$ であり、入力 x_t は 0 から 0.5 までの一様乱数によって得られる。

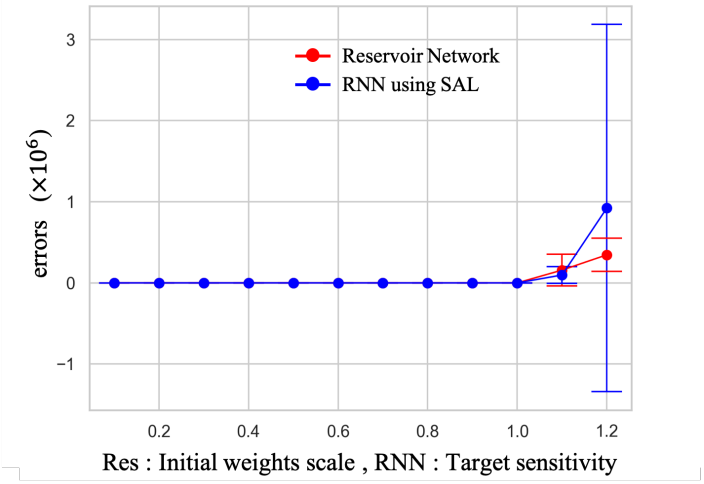
3 結果

従来法で初期化したリザバには 50,000 ステップの FORCE Learning を行い、その後、3,000 ステップのテストを行う。感度調整学習を適用する RNN には、初めの 15,000 ステップで感度調整学習を行い、その後リザバと同様に、50,000 ステップの FORCE Learning と 3,000 ステップのテストを行う。テスト時には目標値と実際の出力の誤差を 3,000 ステップ分で合計する。これを乱数系列 10 パターン分実行し、その誤差の平均を 2 つのネットワークで比較する。

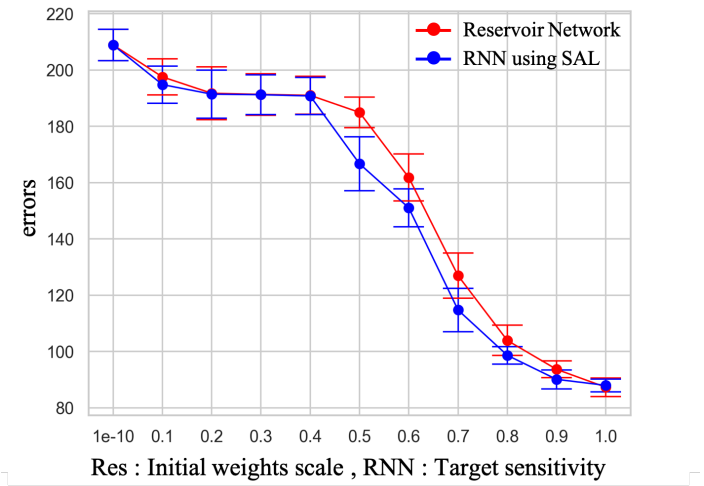
Fig.2 は、リザバの相互結合の重み値スケール λ と、RNN の相互結合重み値の初期値を 10^{-10} とした場合の目標感度を、同時に変化させ学習性能の比較を行った結果である。

Fig.2(a) は 0.1 から 1.2 の間で変化させた場合の結果である。リザバの初期重み値も RNN の目標感度も共に 1.0 を超えると、10^6 のオーダーという非常に大きな誤差となっている。このことから、今回のタスクにおいては、感度調整学習の目標感度の上限を 1.0 とする。

Fig.2(b) は、Fig.2(a) で比較できなった 1.0 以下に、RNN の初期値である 10^{-10} を加え、10^{-10} から 1.0 の間で変化させた場合の結果である。この結果を見ると、



(a) Varied range is from 0.1 to 1.2.



(b) Varied range is from 10^{-10} to 1.0. The scale of the vertical axis is different from (a)

Fig. 2: Comparison of learning results when the weight value scale of the reservoir and the target sensitivity of the RNN are varied together.

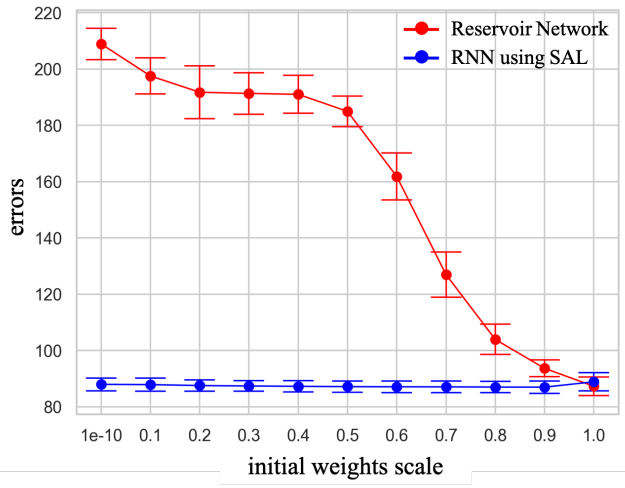


Fig. 3: Comparison of learning results when the weight scales of the reservoir and RNN are set to the same value and the target sensitivity of the RNN is set to 1.0.

どちらの場合も値が大きくなるにつれ学習結果が良くなっている。また、0.4 付近まではリザバ、RNN 共にほぼ同じ結果である。それ以降では、RNN の感度の目標値とした場合の方が、わずかに良い結果となった後、1.0 で再び同程度になった。これは、感度調整学習で重み値を大きくすると、最終的な λ が目標感度の値より少し大きくなることで、同じ数値を直接 λ とするより学習結果が良くなり、1.0 付近では λ による学習性能の差が小さくなるからであると考えられる。

Fig.3 では、リザバと RNN の両方で λ を 10^{-10} から 1.0 の間で変化させ、RNN では目標値を 1.0 として感度調整学習を行った場合の結果である。感度調整学習を行った RNN では、初期重み値のスケールにかかわらず、リザバで最も学習結果の良い $\lambda = 1.0$ と同程度の結果となった。しかし、RNN で初期 λ を 1.0 とした場合では、他の初期値の場合よりわずかに悪い結果となった。このとき、すでに目標感度に近い状態から感度調整学習を行ったことで、最終的な相互結合の重み値スケール λ が、他の初期重み値から大きくする場合よりわずかに大きくなった。このことが原因となり学習性能が低下したと考えられる。

4 結論

本論文では、疎に相互結合をしたニューロンで構成された RNN に感度調整学習を適用したものと、従来の方法で構築されたリザバで、記憶が必要なタスクを用いて学習結果の比較を行い、感度調整学習でリザバとして適切な重み値の生成ができるかを確認した。RNN に感度調整学習を適用することで、相互結合重み値スケール λ の初期値にかかわらず、リザバにとって適切な重み値を生成できることを確認した。この結果だけであれば、従来通り適切な重み値を直接与えてしまえば良いことになる。しかし、感度調整学習により重み値を適切な値に持っていけるという結果は、他学習との並列学習へつながるものと考えている。

今後、相互結合重み値に限らず、入力に掛かる重み値とフィードバックに掛かる重み値を含めて RNN の重み値全てに感度調整学習を適用し、リザバとして適切な重み値を生成できるかを確認する。また、本来の目的である、他の学習と並行して行い、リザバの学習をしつつ全体の学習に悪影響を及ぼさないか調査する。

謝辞

本研究は JSPS 科研費 (15K00360, 20K11993) および栢森情報科学技術振興財団研究助成金の補助を受けた。ここに謝意を表する。

参考文献

[1] D. Hassabis, et al. : Neuroscience-inspired artificial intelligence. *Neuron* Vol.95, Issue2, pp. 245-258 (2017)

[2] A. Hannun, C. Case and J. Gasper et al. : Deep Speech: Scaling up end-to-end speech recognition, *arXiv*, 1412.5567 (2014)

[3] S. Hochreiter and J. Schmidhuber : Long short-term memory. *Neuralcomputation* 9(8) pp. 1735-1780 (1997):

[4] W. Maass T. Natschlger and H. Markram : Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neuralcomputation* 14.11, pp. 2531-2560 (2002)

[5] H. Jaeger. : The“echo state”approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany : German National Research Center for Information Technology GMD Technical Report* 148 (2001)

[6] D. Sussillo, L.F. Abbott : Generating coherent patterns of activity from chaotic neural networks. *NeuronArticle*, Vol.63, No.4, pp. 544-557(2009)

[7] I. Sutskever : *Training recurrent neural networks*. Ph.D. thesis, Department of computer science, University of Toronto (2013)

[8] 徳丸侑輝, 柴田克成 : リカレントネットにおける感度調整学習でのカオスダイナミクスの生成と維持, 第 38 回計測自動制御学会九州支部学術講演会予稿集, pp. 75-78, (2019)

[9] A. Goudarzi, et al. : A Comparative Study of Reservoir Computing for Temporal Signal Processing, *arXiv*, 1401.2224 (2014)