

Reservoir 自己組織化マップによる時系列解析

松尾隆平¹ 堂園浩²

概要 : IoT 機器が普及している中、それに伴い IoT 機器から得られた時系列データの処理を行うクラウドサーバーの負担が増加することが予想されている。そこで本稿では RNN の一種であり、学習時間が短い特徴を持つ Reservoir Computing を自己組織化マップと組み合わせることで、時系列データの予測精度の向上を目的とした。それを実現することにより、現在サーバーの処理で使われている深層学習の負荷を減らす方法の一つとして応用出来ると考えられる。

キーワード : ニューラルネット, 回帰, その他

Time series analysis with Reservoir self-organizing map

RYUHEI MATSUO^{†1} HIROSHI DOZONO^{†2}

Abstract: With the spread of IoT devices, it is expected that the load on cloud servers that process time-series data obtained from IoT devices will increase. In this paper, we aimed to improve the prediction accuracy of time series data by combining Reservoir Computing, which is a type of RNN and has a feature of short learning time, with a self-organizing map. By realizing it, it can be applied as an alternative to deep learning currently used in server processing.

Keywords: Neural net, Regression, Other

1. はじめに

モノとインターネットが繋がる IoT が普及し、スマートフォンやパソコンだけではなく従来インターネットに接続されていなかった家電製品なども、ネットワークを通じてサーバーやクラウドサービスに接続されている。モノとインターネットが接続されることで、得られたデータをクラウドサーバー上で処理や分析を行い相互に情報を交換することで、より良い価値やサービスを提供することを可能にした。しかし、それによりデータの量が増大する一方で、IoT 機器から得られる時系列データを高速に処理しなければならない問題点がある。そのため、サーバーの負担が増加することが考えられる。また、サーバーに集められた時系列データは、階層型ニューラルネットワークと再帰型ニューラルネットワークを組み合わせた深層学習を行いデータの処理をしている。深層学習は精度が良い一方で、消費電力が大きいことや、学習時間がかかることが挙げられる。[1]

そこで、上記の問題点を解決する手段の一つとして再帰型ニューラルネットワークの一種である Reservoir Computing がある。Reservoir Computing の特徴として、重みを調整する箇所が少ないことから学習時間が短いことや、省電力などが挙げられる。

本稿では、Reservoir Computing の状態の解析を行い、自己組織化マップと組み合わせた Reservoir 自己組織化マップを作成し、時系列データの予測精度の向上を目的とした。

それを実現することにより、現在クラウドサーバーの処理で使われている深層学習の負荷軽減が出来ると考えられる。

2. 研究方法

2.1 Python 環境構築

本研究において用いた言語は Python のため、参考文献[2]の Anaconda 公式サイトから PC に Anaconda の 64bit 版をインストールした。Python のバージョンは 3.7.3 を用いた。また、Mackey Glass 方程式を利用するために Signalz というライブラリを Anaconda 上のコマンドプロンプトでインストールした。

Anaconda とは、科学計算（データサイエンス、機械学習アプリケーション、大規模データ処理、予測分析など）のための Python および R 言語の無料のオープンソースディストリビューションである。ディストリビューションとはコンパイルしてある設定済みのソフトウェアの集合体を指す。

2.2 Reservoir Computing の作成と検証

Python を用いて、Reservoir Computing のプログラムを参考文献[3]を参考に作成した。

¹ 佐賀大学
Saga University

² 佐賀大学
Saga University

Reservoir Computing とは、RNN(Recurrent Neural Network) と呼ばれる、過去の出力を現在の入力として扱えるようにすることで、時系列などの順番を扱えるようにしたニューラルネットワークの一種である。Reservoir Computing の概要図を図 1 に示す。

W_{in} が入力層の重み、 W_{res} が Reservoir 層の重み、 W_{out} が出力層の重みである。Reservoir Computing の特徴として、更新をする重みは W_{out} のみであり、他の重みは更新を行わない点である。Reservoir 層の内部状態と、出力層の重みを調整することによって学習が行われる。そのため、学習が出力層のみで行われるため、他の RNN と比較して学習コストの削減ができ、学習時間が少なくなるといった利点がある。

また、Reservoir 層の内部状態の式は次式で表される。

$x(t) = f[(1 - \delta)x(t - 1) + \delta\{W_{in}u(t - 1) + W_{res}x(t - 1)\}]$
 f は活性化関数であり、本研究では $f = \tanh$ で行った。 δ は漏れ率を表す。式より δ の値が小さいほど第一項の値が大きくなり、第二項の値が小さくなるため過去の状態を長く覚えていれるが、直前の入力の影響が小さくなる。逆に δ が大きくなるほど直前の入力の影響は大きいが、過去の状態をあまり覚えることができない状態になる。

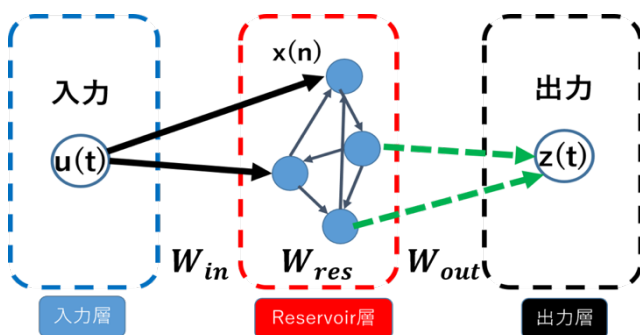


図 1 Reservoir Computing の概要図

Figure 1 Schematic diagram of Reservoir Computing.

条件としてノード数はそれぞれ入力層 1 個、Reservoir 層 150 個、出力層 1 個で行った。

そして作成した Reservoir Computing に時系列データを入力した。

2.3 自己組織化マップを用いた Reservoir 層の状態解析

自己組織化マップ(SOM)とは、1982 年にコホネン(Teuvo Kohonen)によって提案されたニューラルネットワークの一つで、教師なし学習の代表的なアルゴリズムである。高次元データを低次元マップへ写像し、入力データ間の類似度をマップ上での距離で表現する。類似度が高いほど、マップ上での距離が近くなり、類似度が低いほど距離が遠くなるため、データの可視化に利用されている。この簡単な例として図 2 を示す。

Reservoir 層の出力の重みをかける前の値の状態の解析を行うために自己組織化マップを用いて出力値をクラスター

ングした。自己組織化マップの学習回数は 2000 回とした。図 3 に簡単に概要図を示す。

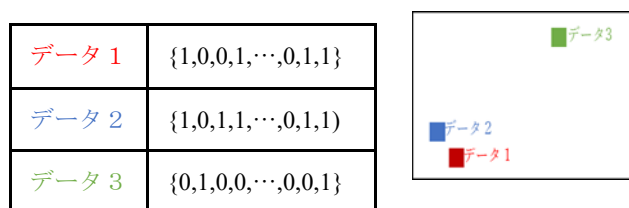


図 2 SOM の概要図

Figure 2 Schematic diagram of SOM.

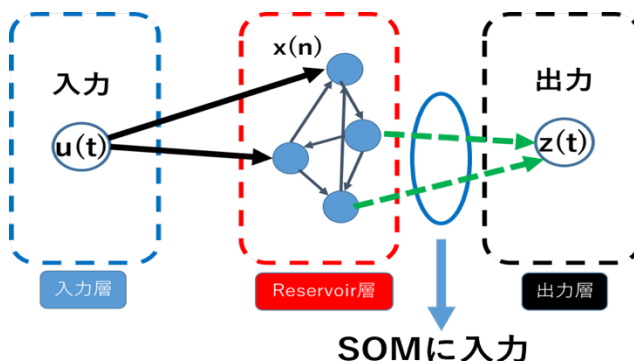


図 3 SOM に入力した概要図

Figure 3 Schematic diagram entered in SOM.

2.4 Reservoir 自己組織化マップの作成と検証

2.3 でクラスタリングしたものを基に、SOM のニューロンに Reservoir 層の出力行列を割り当てた。そして、勝者ニューロンを利用して出力し、その出力値を学習させて、通常の Reservoir Computing との予測結果の精度の比較を行った。条件としては、2.2 の Mackey Glass 方程式での漏れ率 $\delta=0.05$ のときの結果と比較を行った。

3. 実験結果

3.1 Reservoir Computing の検証結果

今回入力したデータとして、正弦波と Mackey Glass 方程式を入力とした。正弦波の条件は、周期 $T=20\pi$ 、振幅 $A=1.0$ とした。また、Mackey Glass 方程式とは非線形時間遅延微分方程式であり、カオスを含む複雑なダイナミクスを生成できる方程式の一つである。正弦波と比較して複雑なデータであり、複雑になった場合の変化を検討するために入力データとした。Mackey Glass 方程式の式は以下の式で表される。[4]

$$x_{k+1} = c * x_k + \frac{a * x_{k-d}}{b + x_{k-d}^e}$$

パラメータの値はそれぞれ $a=0.2$, $b=1$, $c=0.9$, $d=17$, $e=10$, 初期値の値を 0.1 として入力した。

そして、Reservoir Computing のパラメータの値の一つであ

る漏れ率 δ の値を変化させた場合の結果の検証を行った。また、どちらの入力の場合も入力データの 60%を訓練データとした。

図 4, 図 5, 図 6 に正弦波を Reservoir Computing に入力させ、漏れ率を変化させた場合の結果を示す。黒色で示しているのが入力データで、赤色が訓練結果、青色が予測結果となっている。どの場合においても、赤で表示されている訓練データは黒の入力データと重なっていることから、訓練データの学習は行えていることが確認できる。しかし、予測結果は図 4 に示す漏れ率 $\delta=0.02$ のときの結果が一番良い結果が得られていることが分かる。

次に、入力が Mackey Glass 方程式で漏れ率 δ を変化させた場合の結果を図 7, 図 8, 図 9 に示す。赤で示す訓練データの結果は漏れ率 δ の値が大きいほど学習がうまくいっていることが分かる。しかし、予測結果は漏れ率 $\delta=0.05$ のときが他の結果と比較して減衰が少なく、一番精度が良いことが分かる。Mackey Glass 方程式の場合は正弦波の場合と違う漏れ率において予測精度が良い。このことから、予測精度を上げるためには、入力に応じて漏れ率を変化させる必要があることが考えられる。また、Mackey Glass 方程式の場合の予測結果は正弦波の時と比較して、最初の方は予測ができていたが、途中から振幅の減衰が大きく見られる傾向がある。そのため、入力の時系列データが正弦波と比較して複雑な場合だと、予測結果の精度が下がるという結果が得られた。

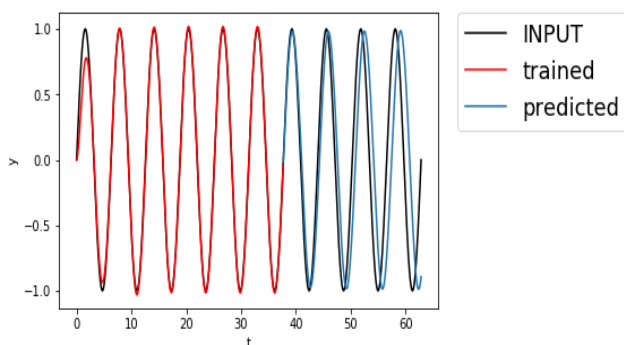


図 4 入力正弦波で漏れ率 $\delta=0.02$ のときの結果

Figure 4 Result when the leakage rate $\delta=0.02$ with the input sine wave.

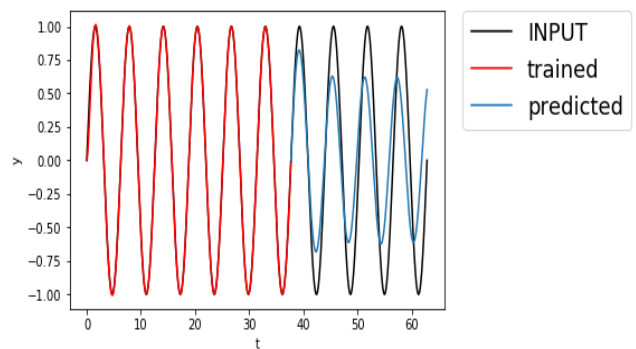


図 5 入力正弦波で漏れ率 $\delta=0.05$ のときの結果

Figure 5 Result when the leakage rate $\delta=0.05$ with the input sine wave.

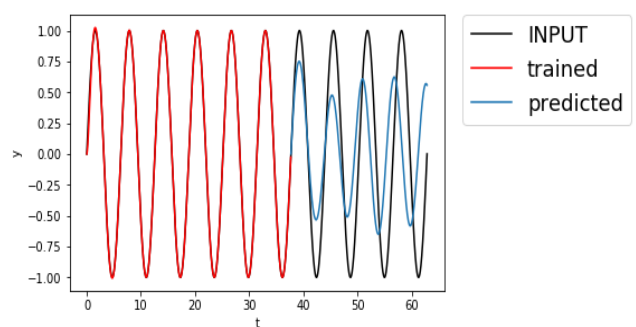


図 6 入力正弦波で漏れ率 $\delta=0.07$ のときの結果

Figure 6 Result when the leakage rate $\delta=0.07$ with the input sine wave.

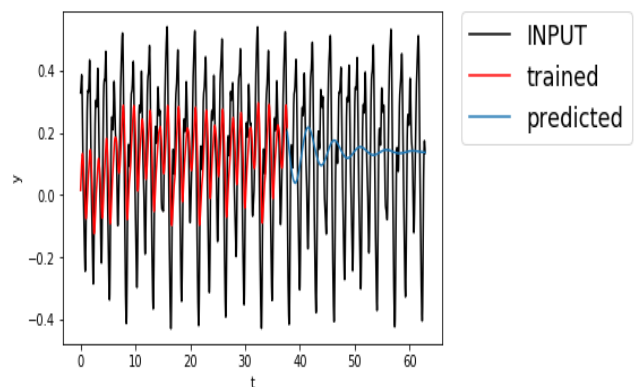


図 7 入力 Mackey Glass 方程式で漏れ率 $\delta=0.02$ のときの結果

Figure 7 Results when the leak rate $\delta=0.02$ with the input Mackey Glass equation.

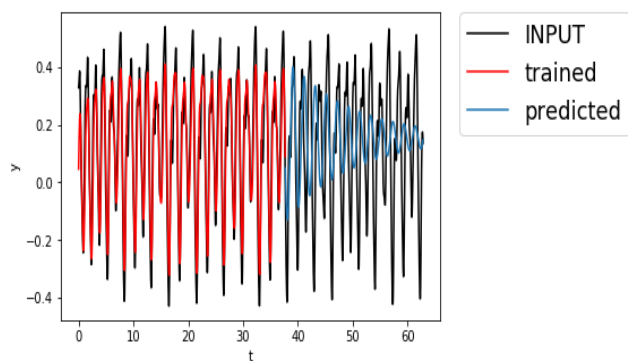


図 8 入力 Mackey Glass 方程式で漏れ率 $\delta=0.05$ ときの結果

Figure 8 Results when the leak rate $\delta=0.05$ with the input Mackey Glass equation.

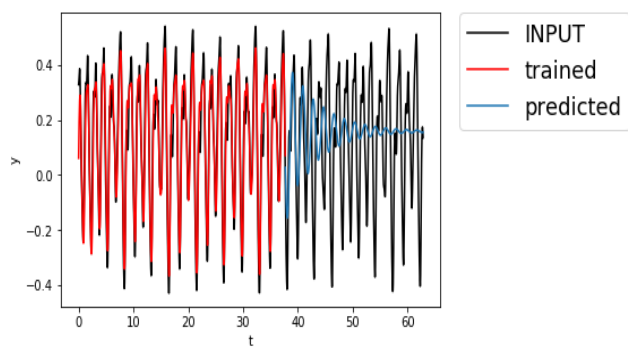


図 9 入力 Mackey Glass 方程式で漏れ率 $\delta=0.07$ ときの結果

Figure 9 Results when the leak rate $\delta=0.07$ with the input Mackey Glass equation.

3.2 自己組織化マップによる Reservoir 層の状態解析結果

まず, Reservoir 層の出力値を表示した際の結果が図 10 の結果になる. これは, Reservoir 層のノード一つ一つの値を表している. 本研究ではノード数 150 個で行っていることから, 150 個の出力値が出力されている. この図 10 から分かるように, ノードごとに値はそれぞれ異なっているということが分かる.

次に, Reservoir 層の出力値に時間をラベル付けして自己組織化マップに学習させた結果を図 11 に表している. つまり, Reservoir 層の出力値の時間ごとの値がクラスタリングされている結果である. また, 図 12 の結果は図 11 の結果と比較を行うために, 見やすくする目的で図 10 の結果の Reservoir 層の出力値の一つだけ取り出したものである. 図 11 の赤丸と黒丸は図 12 の赤点と黒点に対応している. この結果から, 値が近いものは近くに集まっており, 周期的にみてどのように変化しているかを確認できる.

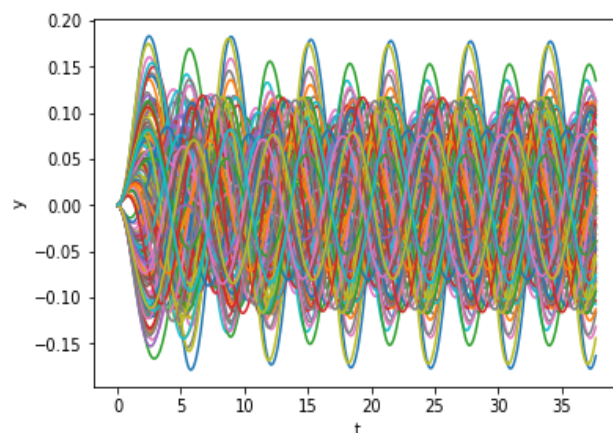


図 10 入力正弦波のときの Reservoir 層の出力値

Figure 10 Output value of Reservoir layer when input sine wave.

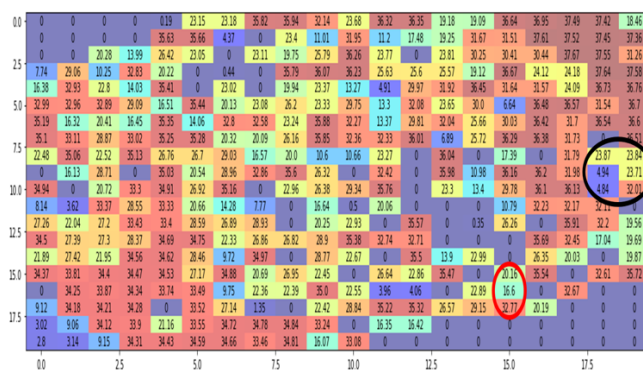


図 11 SOM に時間をラベル付けさせた結果

Figure 11 The result of having SOM label the time.

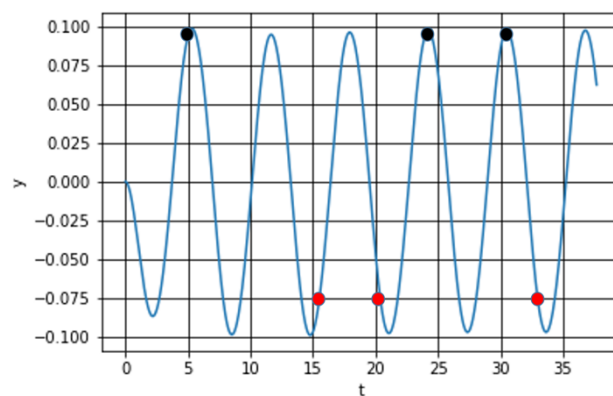


図 12 Reservoir 層のノード 1 個の出力結果

Figure 12 Output result of one node of Reservoir layer.

3.3 Reservoir 自己組織化マップの結果

Reservoir 自己組織化マップで学習させた結果が図 13 の結果になる. 比較対象である図 8 と比較すると, 予測結果の精度が少し向上していることが確認できる. この結果から, Reservoir Computing に自己組織化マップを組み合わせ

ることで精度が良くなり、Reservoir 自己組織化マップの有用性を確認できた。しかし、予測精度の大きな向上は見られず、通常の Reservoir Computing と同様に振幅の減衰が見られる。そのため、この Reservoir 自己組織化マップの更なる研究が必要であると考えられる。

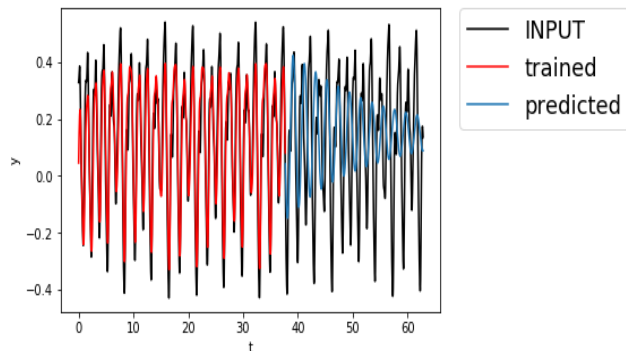


図 13 Reservoir 自己組織化マップの結果

Figure 13 Reservoir self-organization map results.

4. おわりに

本研究では、Reservoir Computing の時系列データの予測精度の向上を目的として実験を行った。Reservoir Computing を作成し、漏れ率を変化させることで予測精度の結果と考察を行った。そして、Reservoir Computing を自己組織化マップと組み合わせることで予測精度の向上ができるということが分かった。しかし、予測精度の大きな変化は見られなかったため、層の数を変化させるなど、さらなる検証が必要である。

謝辞 本研究を行うにあたり、親切に御指導いただいた堂園浩准教授、助言や協力をしていただいたグループ F 研究室の皆様に感謝いたします。

参考文献

- [1] “リザバーコンピューティング –IoT 時代に繋がる新しい情報処理技術” . https://www.library.osaka-u.ac.jp/doc/LS_20191209_reservoir_computing.pdf, (参照 2021-01-21).
- [2] “Anaconda 公式サイト” . <https://www.anaconda.com>, (参照 2021-01-21).
- [3] “ゼロから作る Reservoir Computing” . <https://qiita.com/pokotsun/items/dd8eb48fadeee052110b>, (参照 2021-01-21).
- [4] “Signalz 0.8 ドキュメント マッキーガラス離散方程式” . https://matousc89.github.io/signalz/sources/generators/mackey_glass.html/, (参照 2021-01-21).
- [5] “リザバーコンピューティング –IoT 時代に繋がる新しい情報処理技術” . https://www.library.osaka-u.ac.jp/doc/LS_20191209_reservoir_computing.pdf, (参照 2021-01-21).