

セル・オートマトンを利用した Reservoir Computing の性能比較

岩井さくら (指導教員：オベル加藤ナタナエル)

1 はじめに

再帰ニューラルネットワーク (RNN) は、ニューラルネットワークを拡張し、時系列データを扱えるようにしたものである。RNN は、層を重ね過去の情報を伝搬することでより多くのモデル化が可能になる反面、計算コストが膨大になってしまうという一面もある。

本研究では、その RNN の一種である、Reservoir Computing(RC)[1] という学習モデルに着目した。RC は RNN に比べ、学習コストを削減できるという利点があるからである。この RC をセル・オートマトン [2] というアルゴリズムを利用して構築することを試みた。

1.1 Reservoir Computing

先述した RNN は、入力層・中間層・出力層の 3 つに分かれており、それぞれの間で重みの更新を行うという構造になっている。しかし今回取り上げる RC とは、入力層と中間層 (Reservoir 層) の間は重みをランダムに決定する。そして出力層でのみ簡単な関数を使い学習を行うのである。これにより、膨大になりがちな RNN のコストを大幅に削減することができる。さらに重みの更新を行わない簡単な構造により、低電力での動作することができ、様々な物理システムを使用したハードウェア実装を可能としている。また Reservoir 層でのみ、ノード間の結合を許し層の内部状態を保つことで、モデル内に短期記憶を用意し時系列データの因果関係を扱えるようにしている。つまり、内部で非線形変換が行われていれば Reservoir として使える可能性も秘めているということである。

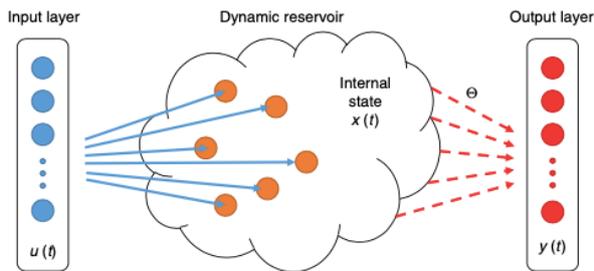


図 1: Reservoir Computing

1.2 セル・オートマトン

セル・オートマトンとは、格子状の計算モデルである。個々のセルの状態を現在の状態と隣接するセルの状態に応じて、一定のルールに従って時間的に更新させていく。ルールは、0 255 までの全 256 種類が存在する。このルールを活用することで Reservoir 層の設定をルール選択のみで行うことができるのである。またセル・オートマトンのもう一つの側面としては、0・1 のバイナリ表現で構成されているという点がある。整数しか使用せず実数を必要としないため、Reservoir として使用することで、学習のための計算が簡単になり、またメモリの使用を抑えられるのではないかと考えられる。

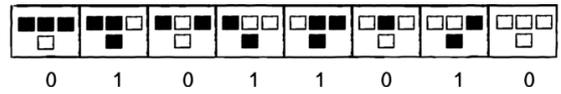


図 2: Rule90

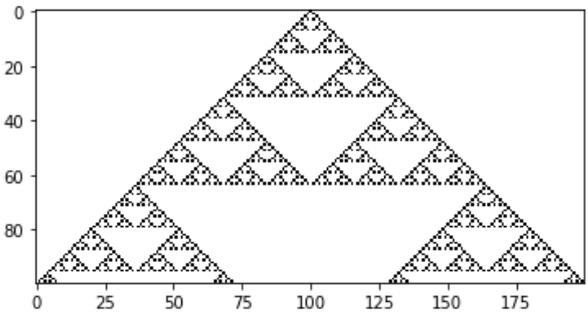


図 3: セル・オートマトン
：中央が 1 のみの状態から Rule90 に従い、セル・オートマトンを更新させたもの

1.3 Classifier

Classifier とは機械学習において、離散的な入力値を決まったクラスに分類することを意味する。そして今回の Classifier には、Support Vector Machine(SVM)[3] を使用した。SVM とは、分類や回帰問題に用いることができる教師あり学習モデルである。2 種類のデータを分割する線に最も近いデータ点 (サポートベクター) からの距離を割り出し、その点から 1 番遠くなる点を選択し境界線を決定していくという仕組みになっている。

2 実験方法

最初に、入力データをセル・オートマトンの初期状態にエンコードさせる。今回は入力データは 5 ビットとした。異なるマッピング方法 (R) に従って、受け取った入力データをランダムに Reservoir にマッピングする。次にそれらをセル・オートマトンの指定したルールに従い、反復回数 (I) でそれぞれ進化させる。セル・オートマトンは、Cellpylib というライブラリを使用した。こうしてできたセル・オートマトンの最終状態を 1 つにまとめ最終データとする。この最終データと最初の入力データを classifier に学習させた。入力データと学習後予測に使用するデータはどちらにも同じ値が入らないように設定している。こうしてできた Reservoir において、数種類のパラメータのベンチマークテストを行った。

3 実験結果

R, I : ランダムマッピングの種類とセル・オートマトンの反復回数を変化させ、実験した。R, I を大きくしていくことにより、おおよそ良い結果が得られるということがわかった。セル・オートマトンの進化の期間が長くなり、異なるランダムなマッピングが多く存在することで、情報が広がり、より多く記憶を保持す

ることを可能とするのである。つまり、 $I \times R$ のサイズが大きくなることで、より正解率を高められるようになると思われる。

	R=8	16	32	64
I=8	24.9%	33.3	31.6	59.9
16	33.3	41.6	34.9	46.6
32	54.9	54.9	46.6	65.0

図 4: R, I

kernel : カーネルとは、データの分類の特徴量を算出する際に、計算を簡単なものに変換してくれる関数のことである。結果は、linear カーネルが1番良く、次に rbf カーネルは結果が得ることができた。反対に、それ以外のカーネルにおいては十分な結果は得られなかった。

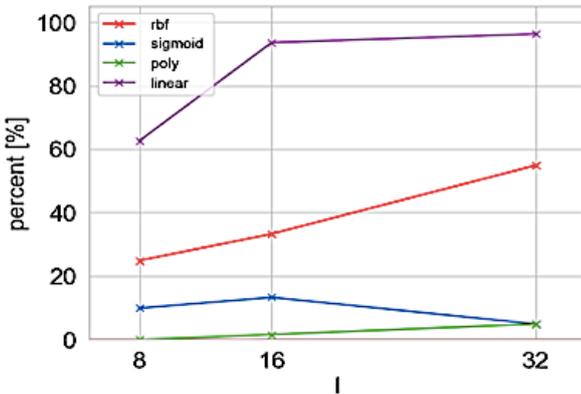


図 5: kernel

Rule : セル・オートマトンのルールを変更して実験した。Rule90 や 150,182 では Reservoir として作用している結果が得られたが、Rule32 や 160 では良い結果が得られなかった。Rule によって、次のセルに伝えることができる情報は違う、つまり時間が経過することで残る情報量も変わってしまい、結果に影響するのだと考えられる。

rule90	R=8	16	32	rule182	R=8	16	32
I=8	24.9	33.3	31.6	I=8	33.3	58.3	58.3
16	33.3	41.6	34.9	16	33.3	33.3	58.3

rule32	R=8	16	32	rule160	R=8	16	32
I=8	0	0	0	I=8	0	0	0
16	0	0	0	16	0	0	0

図 6: Rule

Reservoir の内部状態を変更する : Reservoir 内のノード間の繋がりを追加して実験した。具体的には、前時刻の入力データのセル・オートマトンを作用させた結果と、現時刻の入力データのランダムマッピング後のものを足し合わせた。このようにすることで、より長期的に情報が伝達できるのではないかと考えたが、良い結果は得られなかった。原因の一つとしては、Reservoir の容量に対して一度に多くの情報を追加し

過ぎていてのではないかと考えられる。

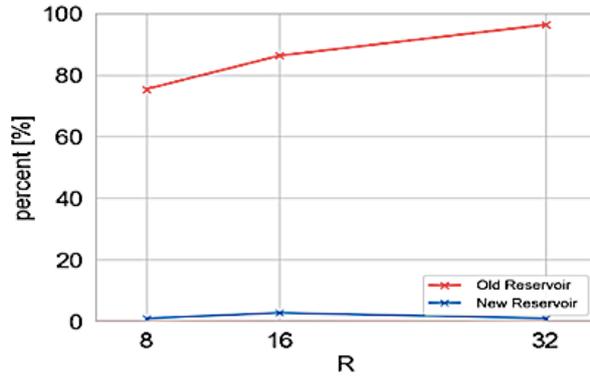


図 7: new/old reservoir

4 まとめと課題

セル・オートマトンを Reservoir として用いることは、今後有望な結果が得られた。特に Reservoir のサイズとルールの選択が重要になっている。しかし、最後の実験では、Reservoir の構造を変えたことで十分な結果が得られなくなってしまった。したがって、より情報を失わずに保持できる構造を考えなければならぬ。さらに、現実で実装するとなるとより多くの課題が残されている。メモリの制約が出てくるためである。また現実問題で考えると、学習させたい情報の他にもノイズのような情報が存在すると考えられる。このような、入力データとは全く関係のない数値を入力する期間をコードに取り入れ、結果がどう変わるのか実験していきたい。

参考文献

- [1] Chao Du, Fuxi Cai, Mohammed A Zidan, Wen Ma, Seung Hwan Lee, and Wei D Lu. Reservoir computing using dynamic memristors for temporal information processing. *Nature communications*, Vol. 8, No. 1, pp. 1–10, 2017.
- [2] Ozgur Yilmaz. Reservoir computing using cellular automata. *arXiv preprint arXiv:1410.0162*, 2014.
- [3] Ashis Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, Vol. 2, No. 8, pp. 82–85, 2012.