

非同期同調セルオートマトンを用いたレザバー計算

日大生産工 ○與野 弘太郎 日大生産工 浦上 大輔

1 はじめに

セルオートマトンとは、オートマトンをマス上に配置したものをいう。横一列に並べたものを一次元セルオートマトン、縦横の格子状に並べられたものを二次元のセルオートマトンという。ここでの「セル」とは盤のマスのことである。各セルはそれぞれ内部状態をもち、各時刻において隣や周りのセルの状態を入力として次の自分の内部状態を決める。一次元セルオートマトンでは、各セルは左右いくつかのセルの状態を入力とする。一次元のセルオートマトンが優れている点は、一列に並んだセルを時間ごとに縦に並べていくと、セルの時間発展の歴史が一目瞭然に分かることである¹⁾。

一方、レザバー計算というものが注目されている^{2,3)}。レザバーとは入力を受け取る非線形で大自由度の結合力学系のことで、レザバー計算は、レザバーとレザバーの状態から出力を決定する部分で構成される。レザバー計算では、レザバー内の結合は固定で学習による更新は行わず、レザバーの状態と出力の関係のみを学習する。単純ではあるが、レザバーのダイナミクスが十分に多様であれば高い性能を示すことで知られている。

セルオートマトンも結合力学系の一種であり、単純な局所規則を並列的に適用することにより多様な時空間パターンを生成するため、レザバーとして有望であると考えられる⁴⁾。しかし、レザバーとして最適な特性をもつシステムは「クラスIV」とよばれているが、クラスIVに属するものは極めて少なく、「カオスの縁」にのみ存在すると考えられている。一方、非同期同調セルオートマトン (AT_ECA) はカオスの縁とは全く異なるメカニズムでレザバーに適したダイナミクスを生成する^{5,6)}。AT_ECAはオーダーパラメータの調整を経ずに、規則空間に対して普遍的にクラスIVを実現する。したがって、AT_ECAを用いたレザバー計算は、通常のセルオートマトンを用いたものよりも高い学習能力が期待される。

2 初等セルオートマトン

一次元セルオートマトンのなかでも最も単純なものに、初等セルオートマトン (ECA) というものがある。内部状態が0 (白) と1 (黒) の2状態で、次式のようにひとつ前の状態のその左右の3つのセルの状態を入力として次の時間ステップの自分の状態を決める。

$$c_i(t+1) = \varphi(c_{i-1}(t), c_i(t), c_{i+1}(t)) \quad (1)$$

図1は、ルール番号60の局所規則とその適用方法を示したものである。これを繰り返しおこなうことで、図2のような時空間パターンが生成される。ここでは空間サイズと時間ステップ数をそれぞれ100、初期値をランダムとした。

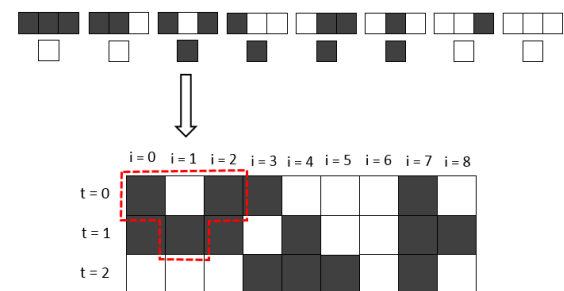


図1 ECA (Rule 60) の局所規則と時間発展

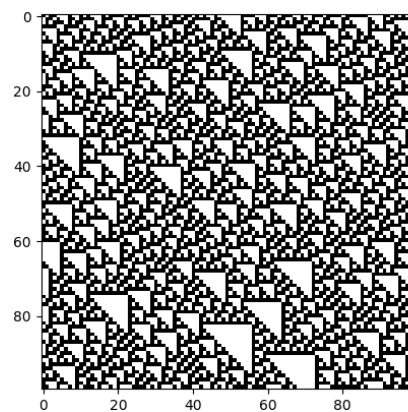


図2 ECA (Rule 60) の時空間パターン

Reservoir Computing using Asynchronously Tuned Elementary Cellular Automata

Kotaro YONO and Daisuke URAGAMI

3 非同期同調セルオートマトン

非同期同調セルオートマトン (AT_ECA) では、非同期つまり一列同時にではなくランダムな順番に一つずつセルが更新される。普通の ECA と違い、AT-ECA は受動態規則と能動的規則という二つの局所規則がある。最初は受動的規則と能動的規則は同じであるが、能動的規則は、受動的規則でセルが更新した場合に変化する。

AT-ECA は順番がランダムで更新されるため、あるセルの状態を更新するときに両隣がまだ更新されていない場合、既に両隣が更新されている場合、片方だけが既に更新されている場合の 3 つの場合に分かれる。この 3 つの場合によって更新の仕方が変わる。両隣がともまだ更新されていない場合は、通常の ECA のように能動的規則で更新される (式(2))。また、能動的規則は変化もしない。

$$c_i(t+1) = \varphi^A(c_{i-1}(t), c_i(t), c_{i+1}(t)) \quad (2)$$

両隣が既に更新されている場合は、現在の自分の状態と両隣の更新後のセルの状態に受動的規則を適用して更新される (式(3))。

$$c_i(t+1) = \varphi_i^P(c_{i-1}(t+1), c_i(t), c_{i+1}(t+1)) \quad (3)$$

その後、両隣が更新されていなかった場合の能動的規則の更新を、このときの更新でできた状態と一致するように能動的規則を変更する。

両隣のうち片方だけが既に更新されていた場合は、その更新後の状態を参照して受動的規則式によって更新する (式(4)または式(5))。

$$c_i(t+1) = \varphi_i^P(c_{i-1}(t+1), c_i(t), c_{i+1}(t)) \quad (4)$$

$$c_i(t+1) = \varphi_i^P(c_{i-1}(t), c_i(t), c_{i+1}(t+1)) \quad (5)$$

この場合は、両隣がまだ更新されていなかった場合の能動的規則による更新を、受動的規則の {000} に対する更新と一致するように能動的規則を変更する。

例として、ルール番号 60 について、6 つのセルの {011010} うち内側の 4 つのセルを左から両端 1 つめ、3 つ目、2 つ目、4 つ目の順に更新した場合を図 3 に示す。また、図 4 はルール番号 60 の AT_ECA の時空間パターンである。先に示した ECA の時空間パターン (図 2) と比べて、ローカルなパターンが枝状に成

長する様子が確認できる。

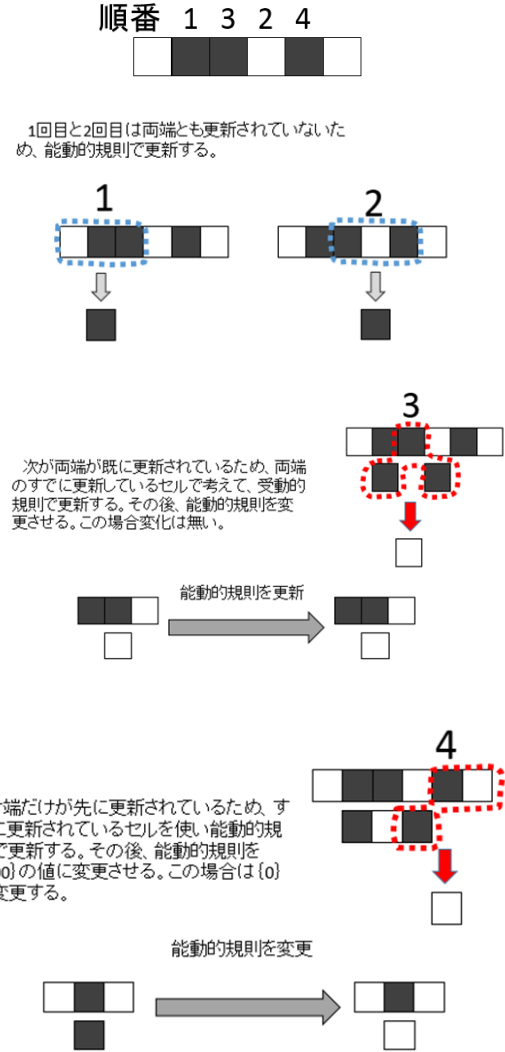


図 3 AT_ECA の時間発展

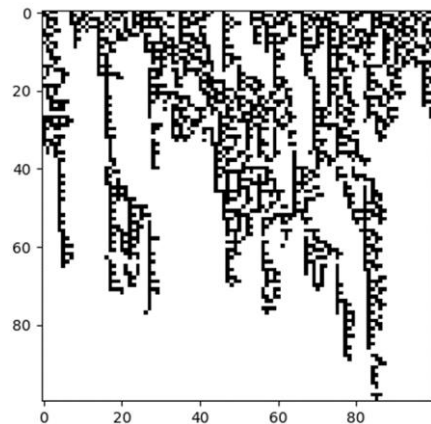


図 4 AT_ECA (Rule 60)の時空間パターン

4 AT_ECAを用いたレザバー計算

提案システムは、図5に示すようにAT_ECAの時空間パターンで実装されるレザバー、入力とレザバーのランダム結合、レザバーから出力を決定するリードアウト (SVM) から構成される。レザバーの大きさを決定するパラメータは L , R , I の3つで、 $L \times R$ はセルの数を、 I はある時刻の入力に対して何回セルを更新するかを意味する。そして、ある時刻の入力に対するレザバーの状態は、 $L \times R \times I$ の大きさの時空間パターンで表現される。

図6は、 $L=20, R=2, I=4$ の例である。まず入力される大きさが4の配列から、各 $L=20$ セルに一つずつランダムに状態をコピーする。この例では、 $R=2$ なので、コピーは2パターンできる。それを初期状態としてAT_ECAの時間発展アルゴリズムにしたがって、 $I=4$ ステップ更新する。このようにして生成された $L \times R \times I$ のサイズの時空間パターンに対してリードアウトによって出力を決定する。次の時間ステップでは、一つ前の時間ステップの最後の行をそのままコピーしたものに、入力された配列の値を最初にランダムに結合したセルと同じセルに上書きする。そしてまた、AT_ECAのアルゴリズムにしたがって時空間パターンを生成し、それから出力を決定する。以上を繰り返しことにより、時系列データの入力に対して出力を決定する。

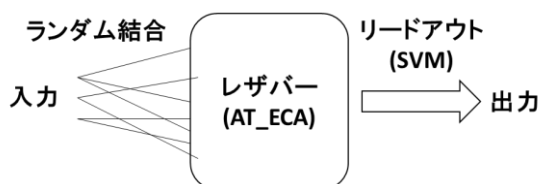


図5 AT_ECAを用いたレザバー計算

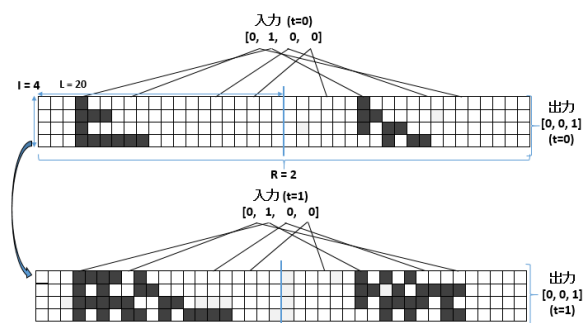


図6 入力とレザバーの結合

5 five-bit taskによる評価

本研究では、図7に示すようなfive-bit taskというテスト課題によって提案システムを評価する。この課題ではある長さのディストラクタを経た後にそれをリピートできるかどうかを問う。最初の5ステップの入力が学習パターンで、左から第1ビットか第2ビットのどちらかが1でそれ以外は0である。したがって学習パターンは32通りある。次に第3ビットのみが1の入力を200ステップ続ける。これがディストラクタとなり、この間に学習パターンを記憶し続ける必要がある。そして、ディストラクタが終わると「キュー」の後に学習パターンの同じパターンを繰り返しパターンとして出力することが学習課題となる。

表1は学習の成功率について、レザバーとしてECAを使用した場合とAT_ECAを使用した場合について比較したものである。両者とも局所規則はルール番号60を使用した。32パターン全てをリピートできた場合に学習が成功したとし、表中の数値はそれぞれ100試行おこなったうちの成功率である。AT_ECAの方がより小さいレザバーサイズ ($R=4, I=4$ および $R=8, I=4$ の場合) でより高い成功率になっていることがわかる。

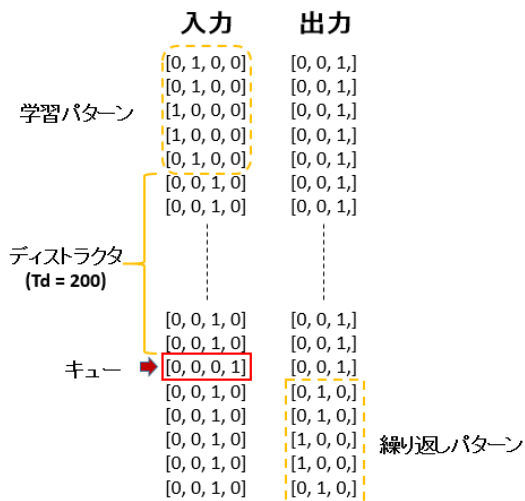


図7 five-bit task

表1 レザバーサイズと学習の成功率

(Rule 60, $L=20$)

	$R=4, I=4$	$R=4, I=8$	$R=8, I=4$	$R=8, I=8$
ECA	0.07	0.37	0.51	1.00
AT_ECA	0.62	0.20	0.97	1.00

6 まとめ

本研究では、レザバー計算のレザバーとして非同期同調セルオートマトン (AT_ECA) を用いた学習システムを提案した。そして、通常のセルオートマトン (ECA) を用いたレザバー計算と提案システムをfive-bit taskをテスト課題として比較した。その結果、提案システムの方が小さいレザバーサイズで成功率が高くなることが示された。また、ECAとAT_ECAのどちらを用いたものもあるレザバーサイズを超えると成功率が1になることが分かった。今後の課題として、ディストラクタの長さを変えたときの影響へ、規則番号を変えるとどうなるかを調査する。

「参考文献」

- 1) Wolfram, S.: A new kind of science, Wolfram Media (2002)
- 2) 伊藤浩之、青柳富誌生： 非線形物理学から見たニューラルネットワークの学習, 人工知能学会誌Vol. 33 No 4, pp. 403-411 (2018)
- 3) 中嶋浩平： やわらかい身体ダイナミクスに計算をアウトソースする, 人工知能会誌Vol. 33 No 5, pp. 507-576(2018)
- 4) O. Yilmaz: Reservoir Computing Using Cellular Automata, arxiv.org/abs/1410.0162 (2014)
- 5) Yukio-Pegio Gunji: Self-Organized Criticality in Asynchronously Tuned Elementary Cellular Automata, Complex Systems, vol.23 (1), pp.55-69 (2014)
- 6) 浦上大輔, 西蔵尚樹, 郡司ペギオ幸夫: セルオートマトンにおける非同期性と $1/f$ ゆらぎ, 計測自動制御学会システムインテグレーション部門講演会 (SI2017)講演論文集, 2B1-08 (2016)