

## 再帰型高次結合ニューラルネットワークの計算能力について

田中 賢<sup>†</sup> 長谷川大剛<sup>††</sup>

On Computational Capability of Recurrent Higher Order Neural Networks

Ken TANAKA<sup>†</sup> and Daigo HASEGAWA<sup>††</sup>

あらまし 有限個の連続値出力ニューロンからなるニューラルネットワークは、任意の決定性チューリング機械をシミュレートする能力をもつことが知られている。これまで、いくつかのシミュレートモデルが提案されているが、テープ部を実時間でシミュレートするには膨大な数のニューロンが必要となることが問題となる。本論文では、再帰型高次結合ニューラルネットワークが任意のチューリング機械をシミュレートできることを示し、シミュレート時間とこれに要するニューロン数の下限を明らかにする。まず、しきい値入出力関数と線形入出力関数を用いる有限個のニューロンからなる再帰型高次結合ニューラルネットワークが、任意の決定性チューリング機械を実時間でシミュレートでき、テープのシミュレートに要するニューロン数を6個まで削減できることを示す。また、のこぎり型関数を用いれば、シミュレートに要するニューロン数を4個に削減できることを示す。単純化されたこのシミュレートモデルは、将来ニューラルネットワークによる形式言語の学習を実現する上で有用な枠組みとなる。

キーワード 再帰型ニューラルネットワーク, チューリング機械, 高次結合, 形式言語

## 1. ま え が き

ニューラルネットワークの計算能力を、形式言語の認識能力や計算モデルの模倣能力の点から明らかにし、その学習汎化能力や並列処理能力を記号処理の問題に応用する方法が近年提案されている [4], [7], [9]。とりわけ、高次結合を含む再帰型ニューラルネットワークは、ニューロン出力間の論理的関係により状態遷移を直接導けるという構造上の利点から、形式言語の受理系学習の枠組みとして多く用いられている [1], [2], [6]。

ニューラルネットワークによる受理系の学習を行うには、対象とする言語の受理系をシミュレートできるニューラルネットワークを用いることが必要となる。有限個の連続値出力ニューロンからなるニューラルネットワークは、任意のチューリング機械をシミュレートする能力をもつことが知られている [11]。連続値出力ニューロンからなるニューラルネットワークは句構造言語の認識や学習を行うための条件を満たしていると

いえる。

有限個のニューロンからなるニューラルネットワークがチューリング機械を模倣するためには、無限長テープをニューロンの連続値出力として実現するとともに、テープの読み書き操作をニューラルネットワークの状態更新として実現する必要がある。また、実時間型の学習アルゴリズムを適用するには、シミュレートが実時間で実行できる必要がある。林原ら [11] は、有限個の有理数値ニューロンからなるニューラルネットワークがチューリング機械と等価な任意の2カウンタマシンをシミュレートできることを最初に示した。ここでは、シミュレートに要するニューロン数については明らかにされていない。Siegelmann [5] は、状態数  $s$  スタック数  $p$  のチューリング等価な  $p$  スタック機械が  $s + 12p + 3^p s + 2 + 2p$  のニューロンからなる再帰型ニューラルネットワークによりシミュレートできることを示したが、スタック部のシミュレートに  $3^p s + 2 + 2p$  のニューロンが必要となる点が問題となる。実際の学習においては、学習の成功率を向上させるために解の存在を保証する最少の学習機械を用いることで探索空間を限定することが望ましい。

本論文では、再帰型高次結合ニューラルネットワーク RHON (Recurrent Higher Order Neural Network)

<sup>†</sup> 新潟大学工学部情報工学科, 新潟市

Faculty of Engineering, Niigata University, 8050 2-no-cho Ikarashi, Niigata-shi, 950-2181 Japan

<sup>††</sup> 新潟大学大学院自然科学研究科, 新潟市

Graduate School of Science and Technology, Niigata University, 8050 2-no-cho Ikarashi, Niigata-shi, 950-2181 Japan

が任意のチューリング機械をシミュレートできることを示し、シミュレート時間とこれに要するニューロン数を明らかにする。まず、しきい値入出力関数と線形入出力関数を用いる有限個のニューロンからなる再帰型高次結合ニューラルネットワークが、任意の決定性チューリング機械を実時間でシミュレートでき、状態数  $s$  のチューリング機械のシミュレートに要するニューロン数を、 $s+6$  個まで削減できることを示す。また、のこぎり型関数を用いてテープ記号の削除操作を簡略化することで、シミュレートに要するニューロン数を  $s+4$  個に削減できることを示す。

RHON は、状態ユニット上での状態表現に依存せず任意の有限オートマトンを実現できる [8], [10] ことから、正規言語の学習に適したニューラルネットワークモデルの一つである。本論文で単純化されたシミュレートモデルは、将来ニューラルネットワークによる任意の形式言語の認識や学習を実現する上で有用な枠組みとなる。

## 2. 再帰型高次結合ニューラルネットワーク

本章では、RHON の構造と動作を概説する。

RHON は、図 1 に示すような、入力ユニット、状態ユニット及びそれらの出力間の積を重み付きで計算する高次結合からなる再帰型ニューラルネットワークモデルである。図中、 $m$  個の入力ユニットは異なる  $m$  個の入力記号にそれぞれ対応する。各時刻では、その時刻に入力される入力記号に対応する入力ユニットだけが 1 を出力し、他の入力ユニットは 0 を出力する。

離散化された各時刻  $t$  における第  $k$  入力ユニットの出力を  $x_k^{(t)}$  で表す。  $n$  個の状態ユニットの出力により、RHON の状態を表す。第  $i$  状態ユニットの出力を  $O_i^{(t)}$  で表す。

高次結合は、 $n$  個の状態ユニットのすべての部分集合と入力ユニットとの間に存在し、状態の更新はこれら 2 次から  $n+1$  次までのすべての高次結合を計算することで進行する。  $2 \leq p \leq n+1$  とし、状態ユニット  $j_1, j_2, \dots, j_{p-1}$  と入力ユニット  $k$  から状態ユニット  $i$  への  $p$  次の結合を  $w_{ij_1j_2 \dots j_{p-1}k}$  と表す。図中の小さい黒丸は高次の結合構造を表すものとする。このニューラルネットワークの状態ユニット  $i$  の時刻  $t+1$  における状態  $I_i^{(t+1)}$  を次のように定義する。

$$\begin{aligned}
 I_i^{(t+1)} = & \sum_{j_1=0}^{n-1} \sum_{k=0}^{m-1} w_{ij_1k} O_{j_1}^{(t)} x_k^{(t)} \\
 & + \sum_{j_1=0}^{n-1} \sum_{j_2=0}^{n-1} \sum_{k=0}^{m-1} w_{ij_1j_2k} O_{j_1}^{(t)} O_{j_2}^{(t)} x_k^{(t)} \\
 & \quad \quad \quad j_1 < j_2 \\
 & + \dots \\
 & + \sum_{j_1=0}^{n-1} \sum_{j_2=0}^{n-1} \dots \sum_{j_n=0}^{n-1} \sum_{k=0}^{m-1} \\
 & \quad \quad \quad j_1 < j_2 < \dots < j_n \\
 & w_{ij_1j_2 \dots j_nk} O_{j_1}^{(t)} O_{j_2}^{(t)} \dots O_{j_n}^{(t)} x_k^{(t)}
 \end{aligned}$$

時刻  $t+1$  における状態ユニット  $i$  の出力  $O_i^{(t+1)}$  を、次のように定義する。

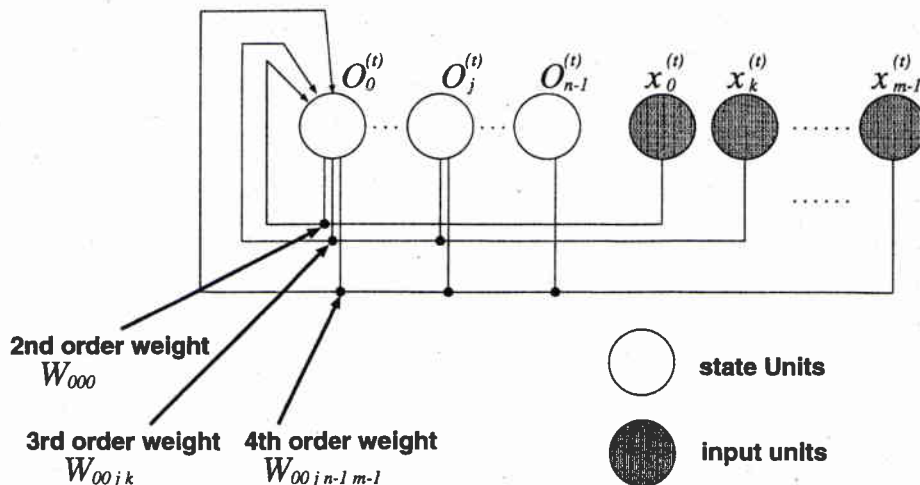


図 1 RHON の構造  
Fig.1 Structure of RHON.

$$O_i^{(t+1)} = g(I_i^{(t+1)}) \quad (1)$$

ここで  $g$  は状態ユニット  $i$  の入出力関数を表す。  $g$  としては、シグモイド関数、線形関数、また以下のような  $\theta$  をしきい値としたしきい値関数を用いることができる。

$$g(x) = \begin{cases} 1 & : x \geq \theta \\ 0 & : x < \theta \end{cases} \quad (2)$$

RHON はただか  $n + 1$  次の高次結合を含む。RHON に含まれる高次結合の最大次数が  $2 \leq p \leq n + 1$  なる整数  $p$  であるとき、RHON の次数は  $p$  であるといい、この RHON を  $p$  次の RHON と呼ぶ。

一般に、メモリをもつ受取系をシミュレートするには、有限状態部の状態とメモリ値から演算を行い、メモリ値を更新する仕組みが必要となる [3]。高次の結合は、ニューロン出力間の論理的関係により状態遷移を直接導けるという構造上の利点をもつ。この点に基づき、次章では決定性チューリング機械を実時間でシミュレートする RHON を構成する。

### 3. RHON によるチューリング機械のシミュレート

本章では、任意の決定性チューリング機械に対し、これをシミュレートできる RHON が存在することを構成的に示す。

チューリング機械  $M$  は、状態集合  $Q$ 、入力記号  $\Sigma$ 、テープ記号  $\Gamma$ 、初期状態  $q_0 \in Q$ 、空白記号  $B \in \Gamma$ 、最終状態の集合  $F \subset Q$ 、遷移写像  $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  からなる 7 項組  $\{Q, \Gamma, \Sigma, \delta, q_0, B, F\}$  で定義される。任意のチューリング機械はテープ記号を  $\{A, B\}$  としたチューリング機械  $M'$  によってシミュレートできる。そこで、任意の  $M'$  についてそれをシミュレートできる RHON が存在することを構成的に示すことによって、任意の決定性チューリング機械をシミュレートできる RHON が存在することを示すこととする。

$M'$  の両方向無限テープは、二つの半無限長スタックメモリと等価な記憶能力をもつ。そこで、半無限長スタックメモリを RHON 上で構成するための補題を以下に示す。

[補題 1] 半無限長スタックメモリ値の無限集合  $\Gamma^*$  から状態ユニット出力の部分集合  $\{O_i^{(t)} \mid 0 \leq O_i^{(t)} \leq 1\}$  への 1 対 1 写像が存在する。

(証明) 半無限長スタックメモリの値を、状態ユニットの出力を無限けたまで用いることで記録することを考える。空白記号を  $B$  としてこれを 0 と対応づけ、他の  $\Gamma$  の要素と 10 進符号記号集合の 0 以外の要素を対応づける。ここでは、 $5 \rightarrow A$  と対応づける。次にスタックメモリの値を 10 進符号記号列として表現し、状態ユニットの出力の小数点以下に割り当てる。例えば、 $AABA \rightarrow 0.5505$  と対応づける。これにより、半無限長スタックメモリの任意の値を状態ユニットの出力として 1 対 1 に対応づけることができる。 □

補題 1 を用いて、任意のチューリング機械をシミュレートできる RHON が存在することを構成的に示す。  
[定理 1] 任意の決定性チューリング機械  $M' = (Q, \{A, B\}, \{A, B\}, \delta, q_0, B, F)$  に対し、それをシミュレートできる有限個のニューロンからなる 4 次の RHON が存在する。

(証明)  $M'$  の状態数を  $n_s = |Q|$  とする。1 個の入力ユニットと  $n_s + 6$  個の状態ユニットからなる RHON を考える。チューリング機械のシミュレートを行うために、状態ユニットを三つの部分に分けて考える。状態ユニット 0 から  $n_s - 1$  はチューリングマシンの状態を表す状態部、 $n_s$  と  $n_s + 3$  は左と右の一方無限テープの値を表すスタック部、 $n_s + 1$  と  $n_s + 2$  は左スタックの先頭文字を、 $n_s + 4$  と  $n_s + 5$  は右スタックの先頭文字を読み出す読出し部である。状態ユニットのうち、スタック部は線形入出力関数を用い、その他の状態ユニットはしきい値入出力関数を用いる。入力記号列は、あらかじめチューリング機械のテープ上に与える必要があるため、符号化された記号列をあらかじめ状態ユニットの初期出力値として与えることとする。ただし、RHON の入力ユニットは動作を保持するため常に 1 を出力する。以上のように構成した RHON を図 2 に表す。図中、太丸は線形入出力関数を用いる状態ユニットをそれぞれ表す。小さい白丸は出力の分岐を表す。  $L, R$  はヘッドの左 ( $L$ )、右 ( $R$ ) の移動時のみに用いられる結合構造を表す。状態ユニットの通し番号を各ユニットの右下に示す。

時刻  $t = 1, 2, 3, \dots$  において状態部のただ一つの状態ユニットが 1 を出力するものとする。状態部の出力により  $M'$  の  $n_s$  個の状態を表すことができる。以下では、このような状態表現に基づき、スタックメモリの書換えを実現する結合重みを決定する。

チューリング機械の遷移写像  $\delta$  の定義域における値を特定するために、また後述するヘッドの移動のため

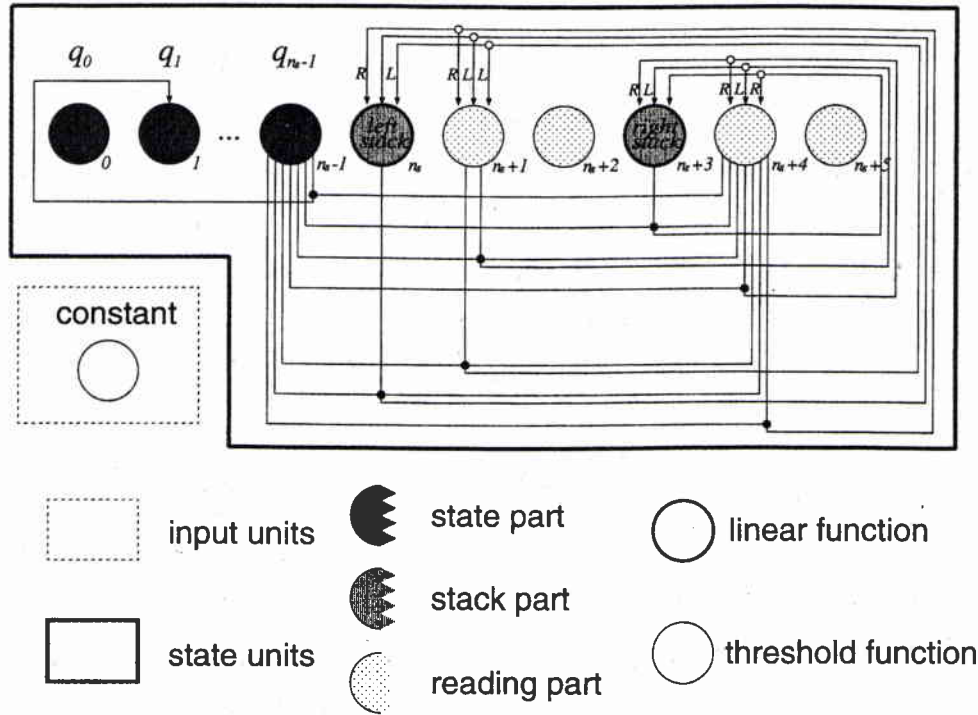


図2  $M'$  をシミュレートする RHON の構造  
Fig. 2 RHON that simulates  $M'$ .

表1 読出し部の入出力関係  
Table 1 input-output relation of reading part.

		左スタックの先頭文字 A	左スタックの先頭文字 B
$n_s + 1$ (しきい値 0.3)	入力の小数第1位以上	0.5	0.0
	出力	1	0
$n_s + 2$ (しきい値 -0.3)	入力の小数第1位以上	-0.5	-0.0
	出力	0	1

に、右スタックメモリと左スタックメモリの出力から最上位の記号に対応する符号を読み出す必要がある。以下のようにすることで、RHONは1回の状態更新で符号を読み出すことができる。左スタックメモリの読出しは、状態ユニット  $n_s + 1$  から  $n_s + 2$  を用いることで実現する。状態ユニット  $n_s + 1$  のしきい値を 0.3、状態ユニット  $n_s + 2$  のしきい値を  $-0.3$  とし、 $n_s + 1$  と  $n_s + 2$  の内部状態が常に逆符号となるように結合構造を定める。これにより、左スタックの更新値の小数第1位が5のとき状態ユニット  $n_s + 1$  が発火し、0のとき状態ユニット  $n_s + 2$  が発火する。更新後のスタック先頭文字に対する読出し部の入出力関係を表1に示す。状態ユニット  $n_s + 1$  と状態ユニット  $n_s + 2$  のどちらかが1を出力し、左スタックの先頭文字が A か B かを表すことができる。ただし初期内部状態として状態ユニット  $n_s + 1$  に初期の左スタック値、状

態ユニット  $n_s + 2$  に負にした初期の左スタック値を設定することが必要となる。これにより  $t = 1$  において左スタックの先頭文字を表すことができる。以上より、状態部の状態ユニットと、状態ユニット  $n_s + 1$ 、 $n_s + 2$ 、 $n_s + 4$ 、 $n_s + 5$  の出力値により写像  $\delta$  の定義域の値が特定できる。

状態ユニットの出力値から、高次結合により  $\delta$  を実現する方法を示す。

$\delta$  の左辺の関係を実現するためには、定数入力ユニット、状態部、右スタックの読出し部が結合された3次以上の結合構造を用いる。以下、この3次結合をまとめて次動作結合と呼ぶ。

RHONは1回の状態更新でPOPを行う。POP操作は、スタック部の出力値を上位に1けたシフトし、最上位の符号を0とすることで行う。上位へのシフトを行うには、POPするスタック部の状態ユニットと

次動作結合を結ぶ4次結合を用いる。この結合重みの値を10とすることで、スタック部の出力と重み10の積が計算され、上位に1けたシフトされる。最上位の符号を0とするためには、そのスタックの最上位の符号を表す読出し部の状態ユニットと次動作結合との間の結合構造を用いる。この結合構造は、左スタックのPOP時には4次となるが、右スタックのPOP時には3次の結合構造となる。すなわち、POPされる符号が0, 5のとき、それぞれの結合重みを0, -5とし、上位シフトを行う出力との和をとることで、POP操作を実現できる。

RHONは1回の状態更新でPUSH操作を行う。PUSH操作は、スタック値を下位に1けたシフトし、テープ記号に対応する10進符号記号を小数第1位に加算することで行う。下位へのシフトを行うには、PUSHするスタック部の出力と次動作結合の間の4次の結合構造を用いる。このときの結合重みを0.1とすることでスタック部の出力と重み0.1の積が計算され、下位に1けたのシフトが行える。PUSHするテープ記号は、スタックと同様に10進符号記号として表現する。シフトされたスタック値とともに、これをスタック部に出力して新たなスタック値を得るためには、PUSHするテープ記号を表す読出し部と次動作結合との間に結合構造が存在すればよい。よって左スタックのPUSH時は3次、右スタックのPUSH時は4次の結合構造となる。

チューリング機械はヘッドの移動前に書換えを行うが、RHONは1回の状態更新でテープ記号の書換えとヘッドの移動を同時に行う。書き換える記号はRHONの右スタックの先頭文字に対応し、これは右スタック値の小数第1位に相当する。そのためにヘッドの移動方向により書き換えるスタックとけたが変化することになる。すなわち、左へのヘッド移動時は右スタックがPUSHされ、書き換えるけたは右スタック値の小数第2位になる。右へのヘッド移動時は右スタックはPOP、左スタックがPUSHされ、書き換えるけたは左スタック値の小数第1位になる。この操作は、ヘッドの移動方向により変化する書換え対象のけたを、書き込むテープ記号に対応する10進符号記号に置き換えることで行う。この書換えは、二つの移動方向について、それぞれ書き換えるスタック、書き換えるけた、書換え対象となるテープ記号、書き込むテープ記号に応じて、次動作結合を決定すればよい。実際には、テープ記号のPUSHで用いる重みとテープ記号の書

換えで用いる重みの和をとって一つの重みとすることで、テープ記号の書換えとテープ記号のPUSHを一つの結合関係で実行できる。

このように、POP, PUSH, テープ記号の書換え操作によりスタックの更新値が計算できる。以上の更新関係を理解するために、重みと結合構造の例を付録に示す。同様の更新値を読出し部にも与えることで、スタックの更新と先頭文字の読出しが同時に行える。

状態部の状態の遷移は、遷移先の状態に相当するユニットへの次動作結合の結合重みを正の値とすることで実現できる。以上のように結合重みを与えることで、任意の遷移関数 $\delta$ を実現できることがわかる。

以上より、任意の $M'$ をシミュレートできる有限個のニューロンからなる4次のRHONが存在することが示された。□

#### 4. のこぎり型入出力関数を用いた場合のチューリング機械のシミュレート

ニューラルネットワークモデルでは、理論的な取扱いや実装の容易さを考慮して、しきい値関数、シグモイド関数、線形入出力関数などの単調非減少関数を用いることが一般的である。一方、特殊な入出力関数を用いることで計算モデルとしてのニューラルネットワークの能力を高めることが可能である[11]。ここでは、のこぎり型入出力関数を用いることで、シミュレートに要するニューロン数の削減を行う。

3.において、RHON上でのPOP操作は、スタック部の出力値を上位に1けたシフトし、最上位の符号を0とすることで行われる。線形入出力関数を用いる場合、最上位の符号を0とするためには、あらかじめ読み出された最上位符号に相当する読出し部の状態ユニットの出力を用い、シフトされたスタック部との和を計算する必要がある。のこぎり型入出力関数を用いることでこの計算を簡略化し、シミュレートに必要なニューロンが削減できることを示す。

[定理2] 任意の決定性チューリング機械 $M'$ に対し、それをシミュレートできるのこぎり型入出力関数を用いた $n_s + 4$ 個のニューロンからなるRHONが存在する。

(証明) 1個の入力ユニットと $n_s + 4$ 個の状態ユニットからなるRHONを考える。状態ユニットのうち、 $n_s$ と $n_s + 3$ をスタック部として以下のようなのこぎり型入出力関数を用い、その他の状態ユニットはヘッド部とししきい値入出力関数を用いる。

$$g(x) = \begin{cases} x & : x < 5 \\ x-5 & : x \geq 5 \end{cases} \quad (3)$$

ヘッドよりも左側のテープの内容を  $n_s$ , ヘッドよりも右側のテープの内容を  $n_s + 3$  の状態ユニットの出力として記録する. また, ヘッド先端部のテープ記号は,  $n_s + 1$  の出力が 1 のとき  $A$ ,  $n_s + 2$  の出力が 1 のとき  $B$  として, それぞれ記録する.

ヘッド部の更新は,  $n_s + 1$  と  $n_s + 2$  のしきい値をそれぞれ 0.3, -0.3 とし,  $n_s + 1$  と  $n_s + 2$  の内部状態が逆符号になるように左右のスタック部からの結合重みを決める.

左右スタックメモリの PUSH, POP 操作は, 対応するスタック部と次動作結合の間の重みをそれぞれ 0.1, 10 とすることで実現できる. POP の際, 整数部に現れる記号はのこぎり型入出力関数のエッジにより消去されるため, スタックメモリの先頭記号をあらかじめ特定するためのユニットは不要となる.

テープ記号の書換えは, 次動作結合を用いて PUSH 操作と同時にされる. PUSH 操作されたスタック部の出力と, 次動作結合によって符号化された書込み記号の和をとることで, テープ記号の書換えが行われる. 以上の更新関係を理解するために, 重みと結合構造の例を付録の図 A.3 に示す.

以上より, 任意の決定性チューリング機械  $M'$  をシミュレートできるのこぎり型入出力関数を用いた  $n_s + 4$  個のニューロンからなる RHON が存在することが示された. □

実時間でスタックの書換えを実現するには, POP 操作において 1 ステップで最上位の記号を削除する操作が常に必要となる. しきい値関数や線形入出力関数を用いてこれを実現する場合も, これらを組み合わせたのこぎり型関数のように任意の最上位の記号を削除できる演算を構成する必要がある. このためには, スタック中の最上位の記号を常に特定して同時にスタック部の出力と和を計算する必要がある. このことから, 定理 1 で求めたユニット数は実時間でシミュレートを行うための下限となることがわかる.

定理 2 の結果は, 最上位記号の削減操作を特殊な入出力関数を用いることで簡略化したものである. 特殊な入出力関数を用いる反面, 左右スタックの結合重みが対称構造となり, ネットワーク設計の手続きが簡略化できる.

## 5. むすび

本論文では, ニューラルネットワークの計算能力を形式言語の認識能力ととらえたときの, 再帰型高次結合ニューラルネットワーク RHON の計算能力を明らかにした. 二つのスタックメモリを RHON 上で実現することで,  $M'$  の両方向無限テープを RHON 上でシミュレートし, 有限個のニューロンからなる RHON が任意の決定性チューリング機械をシミュレートできることを示した.

従来提案されたシミュレートモデルは, スタック数  $p$  の冪乗オーダーのニューロンを必要とした. 本論文では, 任意の  $M'$  が  $p$  に依存せず  $n_s + 6$  個のニューロンからなる RHON により実時間でシミュレートできることを示した. また, のこぎり型入出力関数を用いることで, ニューロン数を  $n_s + 4$  個に削減できることを示した.

状態数  $n_s$  からなる  $M'$  のシミュレートに要する RHON のユニット数は  $n_s + 6$  個である.  $n_s + 6$  個のニューロンのうち,  $n_s$  個の出力はチューリング機械の有限状態制御部の状態を表す. これらは, 状態を陽に表現するために必要な最小のニューロンとなる. 残りの 6 個のうち, 2 個のスタックメモリの値はいずれも可算無限個の値をとり得るため, 潜在的には 1 個の連続値ニューロンの出力値として表現できる. その際のテープ記号の読み書き操作の実現には逆に多くのニューロンを必要とすることが予想される. これらのトレードオフについて, 今後明らかにする必要がある.

本論文で提案したシミュレートモデルでは, しきい値入出力関数やのこぎり型入出力関数を用いているため, こう配法を利用した学習アルゴリズム [10] をそのまま適用することはできない. また, テープ記号の書き換え結果を教師信号として与える場合, ユニット出力や学習信号の精度を十分確保することが困難となる. 今後, 離散的な探索空間に適用できる学習アルゴリズム若しくは教師信号を必要としない学習アルゴリズムを考案する必要がある.

従来提案されたシミュレートモデルは, 可算無限となるニューロンの出力に着目したものである. 連続無限の場合の計算能力に関する考察は今後の課題である.

## 文 献

- [1] A. Cleeremans, D. Servan-Schreiber, and J. McClelland, "Finite state automata and simple recurrent networks," *Neural Computation*, vol.1, no.3,

pp.372-381, 1989.

[2] C. Giles, C. Miller, D. Chen, H. Chen, G. Sun, and Y. Lee, "Learning and extracting finite state automata with second-order recurrent neural networks," *Neural Computation*, vol.4, no.4, pp.393-405, 1992.

[3] J.E. Hopcroft and J.D. Ullman, オートマトン言語理論計算論 I; サイエンス社, 1984.

[4] R. Liow and J.J. Bidal, "A dual network expert system," *Proc. IJCNN '91 at Singapore*, vol.2, pp.1670-1674, 1991.

[5] H.T. Siegelmann and E.D. Sontag, "On the computational power of neural nets," *J. Comput. Syst. Sci.*, vol.50, pp.132-150, 1995.

[6] G. Sun, H. Chen, C. Giles, and Y. Lee, "Connectionist pushdown automata that learn context-free grammars," *Proc. IJCNN '90*, vol.1, pp.577-580, 1990.

[7] R. Sun, "Neural network models for rule-based reasoning," *Proc. IJCNN '91 at Singapore*, vol.1, pp.503-508, 1991.

[8] K. Tanaka and I. Kumazawa, "Learning regular languages via recurrent higher-order neural networks," *Proc. IJCNN '96 at Washington D.C.*, vol.3, pp.899-907, 1996.

[9] 田中 賢, 熊沢逸夫, "再帰型高次結合ニューラルネットワークによる文脈自由言語の認識," *情処学論*, vol.38, no.5, pp.971-979, 1997.

[10] 田中 賢, 熊沢逸夫, 小川英光, "再帰型高次結合ニューラルネットワークによる正規言語の学習," *信学論 (D-II)*, vol.J79-D-II, no.5, pp.899-907, May 1996.

[11] 林原香織, 山下雅史, 阿江 忠, "シグモイド関数の連続性/離散性とニューラルネットワークのマシン能力について," *信学論 (D-II)*, vol.J73-D-II, no.8, pp.1220-1226, Aug. 1990.

付 録

RHONによる  $M'$  の動作のシミュレートの方法を示す。スタックのPOPとPUSHを行うために、表 A.1、

表 A.1 左スタックを操作する重み  
Table A.1 Weights that operate left stack.

		POP		PUSH		書換え	
		上位へシフト	最上位を0に	下位へシフト	Aに書換え	Bに書換え	
左スタックの先頭文字	A	10	-5	0.1	0.5	0	
	B	10	0	0.1	0.5	0	

表 A.2 右スタックを操作する重み  
Table A.2 Weights that operate right stack.

		POP			PUSH		書換え	
		上位へシフト	最上位を0に	下位へシフト	AをPUSH	BをPUSH	Aに書換え	Bに書換え
右スタックの先頭文字	A	10	-5	0.1	0.5	0	0	-0.05
	B	10	0	0.1	0.5	0	0.05	0

表 A.2 に示した値を重みとして用いる。例として、時刻  $t$  において左スタック部の出力を 0.505, 右スタック部の出力を 0.55 とする。このとき、テープ記号を B に書き換えてヘッドが左に移動する TM の動作をシミュレートするスタックの書換え動作は、図 A.1 のように行われる。左スタック部は POP により 0.505 から 0.05 へ、右スタック部はテープ記号の書換えと PUSH により 0.55 から 0.505 へ書き換えられる。この書換え動作を RHON は高次結合と重みを用いてシミュレートする。その結合構造と重みを図 A.2 に示す。

表 A.1, 表 A.2 中, POP を行う重みが 10 と -5, PUSH のために下位へのシフトを行う重みが 0.1 である。図 A.2 中, テープ記号の PUSH と書換えを一つの結合構造で計算するために, A を PUSH する重み

rewriting tape symbol to B and moving the head left

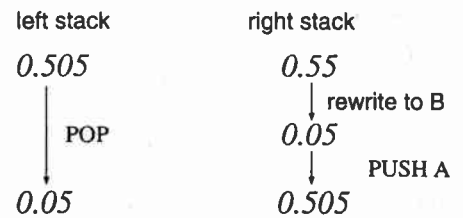


図 A.1  $M'$  の動作をシミュレートするスタックの書換え動作  
Fig. A.1 stack rewriting process in simulating  $M'$ .

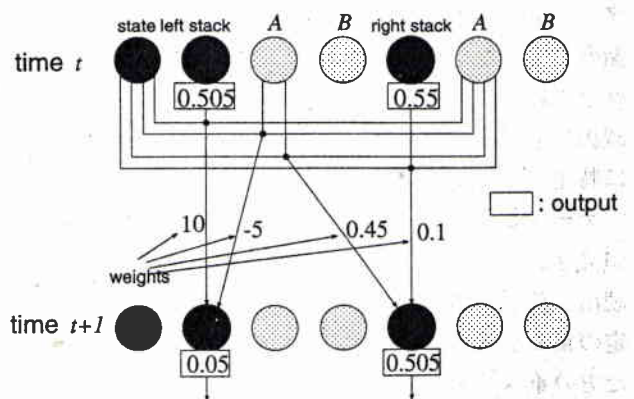


図 A.2 RHON による  $M'$  の動作のシミュレートの例  
Fig. A.2 Example of simulating  $M'$  via RHON.

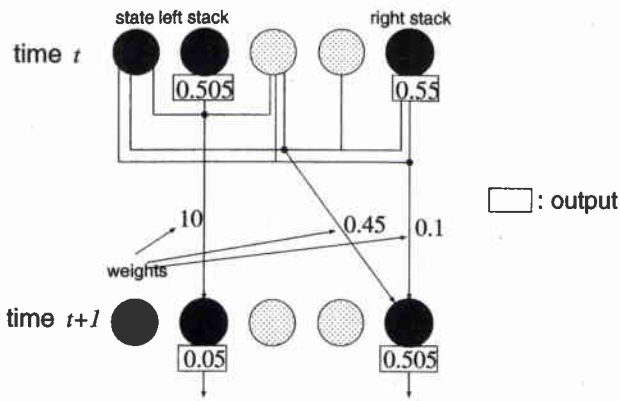


図 A.3 のこぎり型関数を用いた RHON による  $M'$  の動作のシミュレート例  
 Fig. A.3 Example of simulating  $M'$  via RHON with saw type function.



田中 賢 (正員)

平 1 早大・理工・電気卒。平 3 東工大・総合理工・システム科学修士課程了。平 7 東工大・理工・情報工学博士課程単位取得。同年新潟大学大工学部情報工学科助手。ニューラルネットワークの研究に従事。



長谷川大剛 (学生員)

平 9 新潟大・工・情報卒。同大大学院修士課程了。現在、大日本印刷(株)に勤務。ニューラルネットワークの計算論に関する研究に従事。

0.5 とテープ記号を  $B$  に書き換える重み  $-0.05$  との和をとった重みが  $0.45$  である。このように  $M'$  の一つの動作関数をシミュレートするために RHON は四つの重みを用いることが十分である。

各重みに対する結合構造を説明する。まず、TM の動作関数の左辺と対応する次動作結合により、POP するスタックと PUSH するスタック、及び書換えのテープ記号が決定されるため、すべての結合構造は次動作結合を含む。上位と下位へシフトする重みはスタックの先頭文字に関係なく一定で、またシフトするのはスタック部の出力なので、シフトを行う結合構造はスタック部との結合をもつことが十分で、読出し部とは結合をもたなくともよい。最上位を  $0$  にする重みはスタックの先頭文字により変わるので、そのスタックの読出し部と結合構造をもたせ、それぞれの結合の重みに特定の重みを与える。

テープ記号を PUSH する重みは POP する方の先頭文字に対応して変わるので、POP するスタックの読出し部との結合をもたせ、それぞれの結合重みに特定の重みを与える。これにより読出し部の  $1$  を出力した方の重みがいられる。このようにして、結合構造が決定される。その他の TM の動作も同様に RHON は高次結合と重みで模倣できる。以上のようにして  $M'$  の動作が RHON によりシミュレートできる。のこぎり型関数を用いた RHON によるシミュレート例を図 A.3 に示す。

(平成 10 年 8 月 10 日受付, 11 年 1 月 11 日再受付)