Vol. 38 No. 5 情報処理学会論文誌 May 1997

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

田中 賢† 熊沢逸夫††

文脈自由言語の認識や所属判定は，機械翻訳の実現やプログラミング言語の実装など記号処理において不可欠な技術である．文脈自由言語を認識する受理系には半無限長のスタックメモリが必要となるが，有限状態の素子でこれを構成するには無限個の素子が必要となる．また文脈自由言語の所属判定では，語長 n の文の判定に $O(n^3)$ の時間計算量を必要とし，逐次的な計算機では解析時間が急激に増加することも問題となる．本論文では，再帰型高次結合ニューラルネットワークを用い，これらの問題が解決できることを示す．まず，閾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，任意の決定性文脈自由言語を認識できることを示す．また，閾値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を語長に応じて必要なだけ組み合わせることで，任意の非決定性文脈自由言語に対する所属判定を $O(n^2)$ の時間計算量で実行できることを示す．これらの結果より，提案したニューラルネットワークモデルが形式言語の認識や所属判定における多くの問題を解決できる特性を持つことを明らかにする．

Recognizing Context Free Languages via Recurrent Higher-Order Neural Network

KEN TANAKA† and ITSUO KUMAZAWA††

The recognition of context free languages and the determination of its belonging are the essential problem in the symbol processing. In the recognition of any context free languages, the acceptor must realize infinite stack memory. It is impossible to construct infinite stack memory with the elements of finite states. Moreover, the determination of its belongings requires $O(n^3)$ time complexity. In this paper, we show that RHON (Recurrent Higher-Order Neural Network) can avoid these problems. First of all, we show that RHON composed of finite neurons with linear output function can recognize any deterministic context free languages. Secondly, we show that any non-deterministic context free languages can be determined its belonging using the combination of RHON with threshold output function. Our results show the efficiency of our model for symbol processing.

1. はじめに

ニューラルネットワークの学習汎化能力や並列処理能力に対する期待から，これを記号処理に応用し，形式言語の受理系の学習や生成システムのルールの学習を目指す方法が提案されている[1),2),4),7)~9),11)]．ニューラルネットワークによる正規言語の学習や認識については，これまで多くの方法が提案され，正規言語の受理系である有限オートマトンが学習により獲得可能であることが示されている[1),2),11)]．

文脈自由文法は，正規文法よりも高い記述力を持つことから，プログラミング言語の設計や構文解析の定式化など記号処理全般において不可欠である．しかし，文脈自由言語の認識には，文法の特性上多くの計算機資源や時間計算量を必要とする．文脈自由文法の記述力の高さは入れ子構造を記述できる点にあるが，これに起因して，文脈自由言語の認識を行うプッシュダウンオートマトンには半無限長のスタックメモリが必要となる．しかし，有限状態の素子からなる従来の計算機アーキテクチャでこれを構成するには，無限個の素子が必要となる．また，チューリングマシンで文脈自由言語の所属判定を行うには，$O(n^3)$ の時間計算量[3)]を，またランダムアクセスマシンでこれを行う

† 新潟大学工学部情報工学科
Faculty of Engineering, Niigata University
†† 東京工業大学計算工学専攻
Department of Computer Science, Tokyo Institute of Technology

には，$O(n^3/\log^2 n)$ の時間計算量[5]をそれぞれ必要とし，語長が長くなるにつれ解析時間が急激に増加することが問題となる．これまで，Sun ら[7]は，2 次の結合構造を持つ再帰型ニューラルネットワークモデル CPDA を提案し，これが認識できる文脈自由言語の 1 つを示したが，CPDA が認識できる文脈自由言語のクラスについては明らかにしていない．Santos[6]は，コネクショニストモデル PASL を用い文脈自由言語の構文解析を行っているが，PASL が解析できる文脈自由言語のクラスや解析に要する時間計算量については明らかにしていない．

本論文では，文献 11) で提案した再帰型高次結合ニューラルネットワーク RHON 上で半無限長スタックメモリを実現する方法を提案する．スタックメモリの値を，線形入出力関数を用いる状態ニューロンの出力と対応付けることで，任意の決定性文脈自由言語を認識できる有限のニューロンからなるニューラルネットワークのクラスを示す．また，RHON を用いた CYK アルゴリズム[10]の並列計算法を提案する．判定文字列の各アルファベットを生成する非終端記号集合の計算を，各アルファベットについて同時に行うことで，所属判定に要する時間計算量を $O(n^2)$ に削減できることを示す．

本論文の構成は，次のとおりである．2 章では，RHON の構造を概説する．3 章では，半無限長スタックメモリを RHON 上で実現する方法について考察する．そして，閾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 RHON が，任意の決定性文脈自由言語を認識できることを示す．4 章では，RHON を用いて文脈自由言語の所属判定アルゴリズムの 1 つである CYK アルゴリズムを並列に実行する方法について考察する．そして，閾値入出力関数を用いる有限個のニューロンからなる RHON を必要なだけ組み合わせることで，任意の非決定性文脈自由言語を $O(n^2)$ の時間計算量で判定できることを示す．

2. 再帰型高次結合ニューラルネットワーク

再帰型高次結合ニューラルネットワーク RHON は，入力ユニット，状態ユニットおよびそれらを結ぶ高次結合からなる．RHON の構造を**図 1** に示す．図中，m 個の入力ユニットは入力記号の 1 つ 1 つに対応する．各離散時刻 t における第 k 入力ユニットの出力を $x_k^{(t)}$ で表す．各時刻の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．n 個の状態ユニットの出力により，RHON の状態を表す．各離散時刻 t における第 i 状態ユニットの出力を $O_i^{(t)}$ とする．

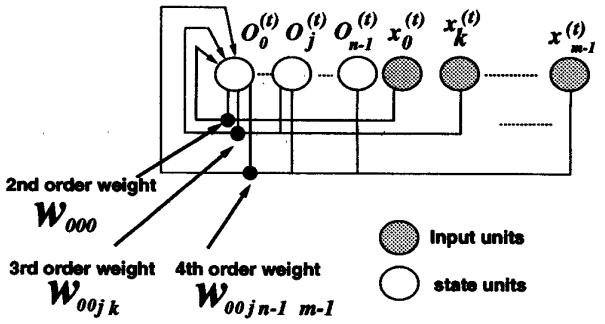

図 1 RHON の構造
Fig. 1 Structure of RHON.

状態の更新は，入力ユニットの出力と状態ユニットの出力の $n+1$ 次までの関係を計算することで進行する．$2 \le p \le n+1$ とし，状態ユニット $j_1, j_2, \cdots, j_{p-1}$ 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 $w_{ij_1j_2\cdots j_{p-1}k}$ 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．ただし，2 次，3 次といった異なる次数の結合構造が同様の黒丸で示してある．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．

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

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．

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

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 θ を閾値とした閾値関数を用いることができる．

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

RHON はたかだか $n+1$ 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 $2 \le p \le n+1$ なる整数 p であるとき，RHON の次数は p で

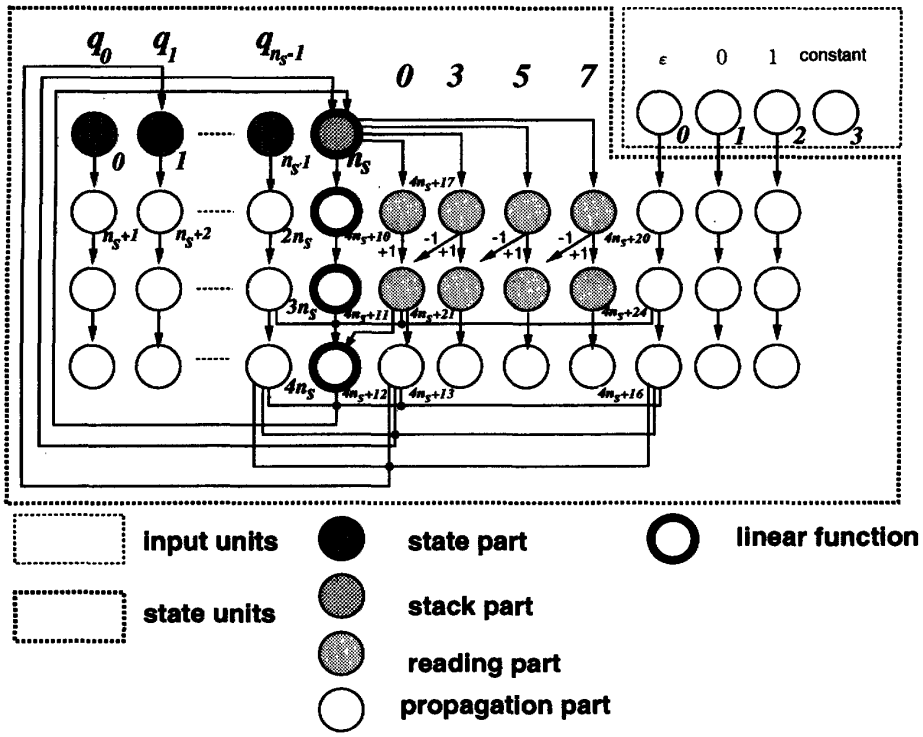

図2 PDA をシミュレートする RHON の構造
Fig. 2 Structure of RHON that simulates PDA.

あるといい，この RHON を p 次の RHON とよぶ．

3. RHON による決定性文脈自由言語の認識

プッシュダウンオートマトンは文脈自由言語を認識できる．よって，任意のプッシュダウンオートマトンをシミュレートできるニューラルネットワークは，文脈自由言語を認識することができる．本章では，任意の決定性プッシュダウンオートマトンに対し，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．

プッシュダウンオートマトンの状態集合を Q，入力アルファベットを Σ，スタックアルファベットを Γ，初期状態を $q_0 \in Q$，初期記号を $Z_0 \in \Gamma$，最終状態の集合を $F \subset Q$ とし，その動作を写像 $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to Q \times \Gamma^*$ で定める．ただし ϵ は空系列である．任意の決定性プッシュダウンオートマトンは，入力アルファベットを $\{0,1\}$，スタックアルファベットを $\{A, B, Z_0\}$ とした決定性プッシュダウンオートマトン M' によってシミュレートできる．そこで，ここでは M' をシミュレートできる RHON の存在を示す．これにより，RHON が任意の決定性文脈自由言語を認識できることが分かる．

半無限長スタックメモリを RHON 上で実現する方法について，以下の補題が成立する．

補題 1 半無限長スタックメモリ値の無限集合 Γ^* から状態ユニット出力の部分集合 $\{O_i^{(t)} : 0 \le O_i^{(t)} \le 1\}$ 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は，状態ユニットの出力を無限桁まで用いることで実現する．まず $\Gamma \cup \{\epsilon\}$ の各要素と 10 進符合記号集合の要素を対応付ける．ここでは，$0 \to \epsilon$，$3 \to A$，$5 \to B$，$7 \to Z_0$，と対応付ける．次に，スタックメモリの値を 10 進符合記号の列として表現し，状態ユニットの出力の小数点以下に割り当てる．たとえば，$AABABAA \to 0.3353533$ と対応付ける．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 に対応付けることができる． □

補題 1 を用いて，任意の決定性プッシュダウンオートマトン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 任意の決定性プッシュダウンオートマトン $M' = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ 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 5 次の RHON が存在する．

(証明) M' の状態数を $n_s = |Q|$ とする．4 個の入力ユニットと $4n_s + 25$ 個の状態ユニットからなる RHON を考える．空系列 ϵ と入力記号 0, 1 は，入力ユニット 0, 1, 2 の出力として外部から与えられる．入力ユニット 3 は，つねに 1 を出力するものとし，これを定数入力ユニットとよぶ．状態ユニットは，4 つの部分からなる．状態ユニット 0 から $n_s - 1$ はプッシュダウンオートマトンの状態を表す状態部，n_s は半無限長

スタックメモリの値を表すスタック部，n_s+1 から $4n_s+16$ は他のユニットの出力を保持するための伝達部，$4n_s+17$ から $4n_s+24$ はスタック記号の読み込みを行う読み出し部である．状態ユニットのうち，スタック部と伝達部の一部は線形入出力関数を用い，その他の状態ユニットは閾値入出力関数を用いる．以上のように構成したRHONを，**図2**に示す．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．入力ユニット，状態ユニットの番号を各ユニットの右上，あるいは右下に示す．ただし，定数入力ユニットと状態ユニットの間の高次結合は，図の繁雑さをさけるために略記している．

状態部の状態ユニットが，時刻 $t=0,4,8,12\cdots$ においてただ1つ $O_i^{(t)}=1$ なる値を出力するものとすると，状態部の出力によりプッシュダウンオートマトンの n_s 個の状態を表すことができる．補題1より，M' の状態，スタックメモリの値を各ユニットの出力の組として保持できるRHONが構成できる．

写像 δ は，入力ユニットと状態ユニットの出力の高次の関係を計算することで実現する．δ の左辺の関係を実現するためには，あらかじめスタックユニットの出力から最上位の記号に対応する符合を読み出す必要がある．符合を読み出すには，RHONは2回の状態更新を必要とする．これは，状態ユニット $4n_s+17$ から $4n_s+24$ を用いることで実現する．最上位の4つの記号に対応して，状態ユニット $4n_s+21$ から $4n_s+24$ のいずれか1つのみが発火するように結合重みを定めることを考える．まず，$4n_s+17$ から $4n_s+20$ までの4個のユニットが，スタック部の出力 $O_{n_s}^t$ を受け，それぞれ $O_{n_s}^t\geq 0.0$，$O_{n_s}^t\geq 0.3$，$O_{n_s}^t\geq 0.5$，$O_{n_s}^t\geq 0.7$ のとき発火するためには，スタック部の出力を $4n_s+17$ から $4n_s+20$ までユニットに伝達するために2次の結合構造が定数入力ユニットとスタック部の間に存在すればよい．ここで，$4n_s+17$ から $4n_s+20$ までのユニットには，適切な入出力関数の閾値が与えられるものとする．次に，$4n_s+17$ から $4n_s+20$ までのユニットの出力をもとに，$4n_s+21$ から $4n_s+24$ のいずれか1つのみが発火するような結合重みを考える．このためには，まず $4n_s+17$ から $4n_s+20$ までのユニットの出力を $4n_s+21$ から $4n_s+24$ に伝達するために2次の結合構造が状態ユニットと定数入力ユニットの間に存在し，さらに $4n_s+21$ から $4n_s+24$ までのユニットが排他的に発火するような抑制的な2次の結合重みが，$4n_s+17$ から $4n_s+20$ までの状態ユニットと定数入力ユニットの間に存在すればよい．

スタックメモリの書換えは，最上位の記号をPOPした後 δ により定められるスタック記号列をPUSHすることで行われる．POPおよびPUSH動作は，状態，入力記号，読み込まれたスタック記号に基づいて行われる．スタック記号の読み出しを行う2回の状態更新の間，状態と入力記号，スタックの値を保持する必要がある．このため，伝達部の状態ユニット n_s+1 から $4n_s+11$ を用いる．一般に，ある状態ユニットの出力値と定数入力ユニットの積を他のユニットに伝達し，伝達されたユニットの次の時刻の出力を用いることで状態ユニットの値を間接的に保持することができる．このことから，状態と入力記号，スタックの値を保持するためには，定数入力ユニットと状態部，伝達部の間にそれぞれ2次の結合構造が存在すればよい．

RHONは1回の状態更新でPOPを行う．POP操作は，伝達部を介して保持されたスタックユニットの出力値を上位に1桁シフトし，最上位の符合を0とする2段階の操作で行う．まず，上位へのシフト操作は，状態ユニット $4n_s+11$ により保持されたスタックユニットの出力値を10倍することで実現できる．この操作を，定数入力部，伝達部を介して保持された状態部，入力ユニット，スタック部の出力，および読み出し部の出力の5つのユニットから，状態ユニット $4n_s+12$ へ5次の結合構造を設け，この値を10とすることで実現する．さらに，最上位の符合を0とするために，状態ユニット $4n_s+12$ の出力の整数部を0とするような演算を行うことを考える．整数部を0とするためには，POPされた記号に対応する10進符合語を d としたとき，$-d$ なる値を状態ユニット $4n_s+12$ に加えることで実現できる．これは，定数入力ユニットと状態ユニット $4n_s+21$ から $4n_s+24$ までの間に2次の結合を考え，このうち状態ユニット $4n_s+12$ へ向かう結合重みの値をそれぞれ $0,-3,-5,-7$ とし，状態ユニット $4n_s+11$ との和をとることで実現できる．

RHONは1回の状態更新でPUSH操作を行う．POP操作を行う1回の状態更新の間，読み出されたスタック記号を保持するため，状態ユニット $4n_s+13$ から $4n_s+16$ を用いる．状態と入力記号，スタックの保持と同様の考察により，これらを保持するためには，定数入力ユニットと伝達部の間に2次の結合構造が存在すればよい．

PUSH操作は，POPされたスタック値を下位にシフトしスタック記号列を加算することで行う．まず，下位へのシフトを行うには，シフトする桁数を c としたとき，伝達部の状態ユニット $4n_s+12$ の出力を n_s に伝える際の結合重みを，10^{-c} とすることで実現で

きる．そこで，定数入力ユニット，伝達部を介して保持された状態とPOPされたスタック値，入力ユニット，スタック部の5つの出力からスタック部 n_s へ向かう5次の結合重みを 10^{-c} とすることで実現する．記号列の加算は，PUSHするスタック記号列をスタックと同様に10進符合記号の列として表現し，これをシフトされた値に加算することで行う．書き込み記号列は，PDAの状態，入力記号，およびスタック最上位の記号により決定される．そこで，それらの値を保持している3つの伝達ユニットと定数入力ユニットからスタック部へ向かう4次の結合重みを書き込み記号列に対応する値とし，下位へシフトしたスタック記号列との和をとることで書き込み操作を実現することができる．

PDAの状態の遷移は，現在の状態，入力記号，スタックの読み出し記号によって決定される．状態部の値の更新は，PDAのこれらの3つの値を保持する伝達部と定数入力ユニットから状態部へ向かう4次の結合重みを考え，3つの値の条件が満たされたときに対応する状態部のユニットが発火するようにすることで実現できる．

以上より，M' をシミュレートできる有限のユニットからなるRHONの存在を構成的に示した．ここで用いた結合重みのうち，最大次数のものは上位へのシフトと下位へのシフトにおいて用いた5次の結合重みである．このことから，M' のシミュレートには，5次の結合重みが十分である． □

RHONにより決定性文脈自由言語の認識を行う場合，語の長さに依存してニューラルネットワークの構造や素子数を変更する必要がない．このことから，5次のRHONは文脈自由言語の認識や学習に適した構造を持つといえる．

4. RHONによる所属判定アルゴリズムの並列化

ニューラルネットワークの並列処理性を用いることで，判定に要する時間計算量を減らすことができる．本章では，任意の非決定性文脈自由言語に対して，$O(n^2)$ の時間計算量で所属判定を行うRHONが存在することを示す．

4.1 FSMによるCYKアルゴリズムの実行

まず，有限の記憶装置を持つ有限状態機械FSM（Finite State Machine）を定義し，これを格子状に配置することでCYKアルゴリズムを並列に実行できることを示す．そして，FSMと同型なRHONが存在することを示し，RHONが任意の非決定性文脈自由言語を $O(n^2)$ の時間計算量で認識できることを示す．

文脈自由文法 $G =< \Sigma, \Gamma, \delta, S >$ と，長さ n の文字列 $x_1 x_2 \cdots x_n, x_i \in \Gamma, 1 \leq i \leq n$ が与えられたとき，式(3)で定義する $P(i,j) = \{N | N \in \Sigma\}, 1 \leq i \leq j \leq n$ を再帰的に計算し，$P(1,n)$ を求めるアルゴリズムをCYKアルゴリズムという．

$$P(i,j) = \begin{cases} \{A | A \in \Sigma, A \to x_i \in \delta\} & i = j \\ \bigcup_{k=i}^{j-1} P(i,k) \otimes P(k+1,j) & i \neq j \end{cases} \tag{3}$$

ここで，Σ は非終端記号の集合，Γ は終端記号の集合，δ は生成規則の集合，S は開始記号で $S \in \Sigma$ である．また，$P(i,k) \otimes P(k+1,j) = \{A | A \in \Sigma, A \to BC \in \delta, B \in P(i,k), C \in P(k+1,j)\}$ とする．

FSM M は，6項組 $(Q, \Sigma, \gamma, \delta, P, B, q_0, F)$ で定義される，1外部入力2入力2出力の有限状態機械である．ここで，Q は状態の集合，Σ は入出力記号の有限アルファベットである．Λ で Σ のべき集合 $\wp(\Sigma) = \{p | p \subseteq \Sigma\}$ を表すものとする．$\gamma : Q \times \Sigma \Rightarrow Q \times \Lambda$ は外部入力を受けた場合の状態遷移関数，$\delta : Q \times \Lambda \times \Lambda \Rightarrow Q \times \Lambda \times \Lambda$ は外部入力を受けない場合の状態遷移関数である．P および B は有限の記憶装置で $P, B \in \wp(\Sigma)$ なる値をとる．$q_0 \in Q$ は初期状態，$F \in Q$ は終了状態の集合を表す．FSMは，外部入力 $\sigma \in \Sigma$ を受けた場合にのみ γ に従って状態遷移し，同一の集合 $\lambda \in \Lambda$ を2出力に出力する．それ以外の場合は，2入力からそれぞれ集合 $\lambda \in \Lambda$ を入力した後 δ に従って状態遷移し，2出力にそれぞれ集合を出力する．FSMの構造を**図3**に示す．図中，点線は外部入力を，細線は2入力を，太線は2出力を表す．

$P(i,j)$ の計算が，$i \leq k \leq j-1$ の範囲で $P(i,k)$ と $P(k+1,j)$ の間の演算を行うことにより実行されることに着目し，FSMを格子状に配置してCYKアルゴリズムを並列に実行することを考える．**図4**に

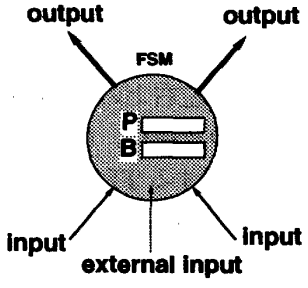

図3 FSMの構造
Fig. 3 Structure of FSM.

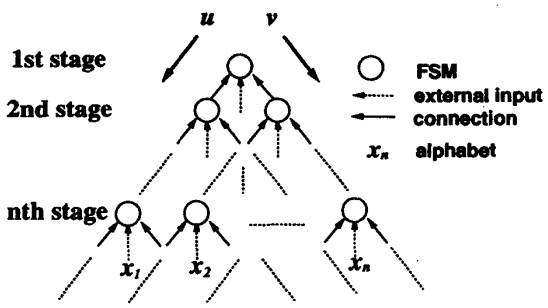

図4 TFSMの構造
Fig. 4 Structure of TFSM.

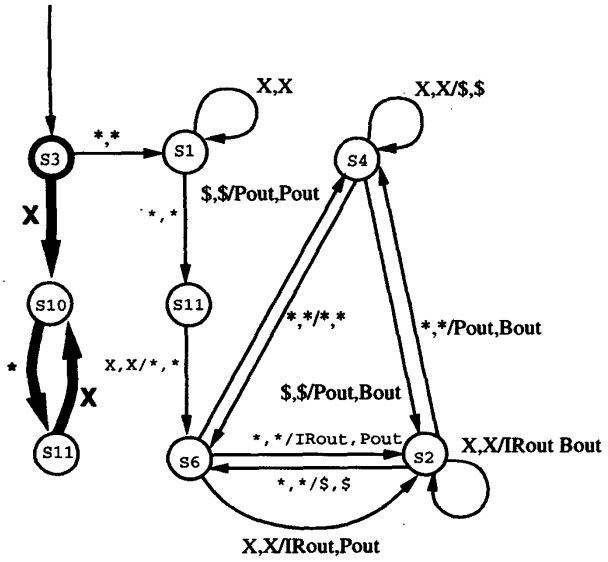

図5 状態遷移図
Fig. 5 State transfer graph.

示した格子状のFSMをTFSM（Trellis FSM）とよぶ．格子の結合方向にそってu軸とv軸をとり，各格子点を座標(u,v)で表す．$1 \leq c$である任意の整数cについて，$u+v-1=c$となる(u,v)に位置するFSMをc段のFSMとよぶ．(u,v)に位置するFSMが$P(v,n-u+1)$の計算を行うために，(u,v)のFSMの2入力を，$(u,v+1)$，$(u+1,v)$のFSMの2出力と結び付ける．図中，丸はFSMを，点線は外部入力を，実線は入力と出力の結び付きを表す．

TFSMは，外部から語を受け認識を行う．長さnの語は，n段の各FSMに与えられる．各FSMは外部入力から各アルファベットを受け取る．1段から$n-1$段までの各FSMは，2入力を通して入力を受け，状態遷移を行うと同時に左右上のFSMに2出力を通じて出力を伝える．語の認識は，$(1,1)$のFSMが行う．

$P(i,j)$の計算を実行するTFSMを構成できるFSMの状態遷移図の1つを図5に示す．

図中，$*$，$ はPを送り出すための同期信号を表す．二重丸は初期状態を表す．太線は初期状態において外部入力を受けた場合の状態遷移を，細線はそれ以外の場合の状態遷移を表す．左右下のFSMからA_1, A_2なる入力を受けたときに，左右上のFSMにB_1, B_2なる出力を伝えて状態遷移することを，$A_1, A_2/B_1, B_2$と表すものとする．

時刻1において外部入力より終端記号Xを受けたN段目のFSMは，時刻1において$N-1$段目のFSMに動作の開始を知らせるために$*$を伝える．以後，偶数時刻においては非終端記号を，奇数時刻においては$*$を，$N-1$段目のFSMに交互に伝える．

時刻Tにおいて2入力より同期信号$*,*$を受けたN段目のFSMは，時刻$T+1$において$N+1$段目のFSMから非終端記号X, Xを受ける．t回非終端記号を受けた後，時刻$T+t$において終了を示す同期信号$*,*$を受け取る．時刻$T+t+1$において$N-1$段目のFSMに対し$*$を出力する．同時に$N+1$段目のFSMからの入力をBに格納する．

4.2 RHONによる文脈自由言語の認識

任意のFSMの状態集合に対し，2次のRHONの状態集合のある部分集合との同型写像が存在することを示す．これにより，RHONが任意の非決定性文脈自由言語を認識できることが分かる．

FSMとRHONについて，以下の補題が成立する．

補題2 任意のFSMに対し，それと同型な2次のRHONが存在する．

[証明] FSM $M=(Q,\Sigma,\gamma,\delta,P,B,q_0,F)$の$P$，$B$は有限の値をとることから，状態集合$Q$でとりうる$P$，$B$の値を含めた新たな状態集合$Q'$を考えることができる．$p=|\Lambda|$，$q=|Q|$とすると$|Q'|=p^2q$となる．ここで，あらたに定義された$Q'$に対し，有限オートマトン$FA=(Q',\Sigma,\delta' q_0,F)$を考える．ここで，任意の入力系列に対応する$M$の状態遷移系列に応じた状態遷移系列を実現する$\delta'$が存在することより，任意の$M$に対し，それと同型なFAが存在する．

$m=|\Sigma|$個の入力ユニットとn個の状態ユニットからなるRHONを考える．1つの入力ユニットの出力により1つの入力記号を表すことから，このFAの入力記号との全単射な写像が存在する．

各時刻においてその中のただ1つが0よりも大きな値を出力し，それ以外のユニットは0を出力するものとすると，このRHONはn個の状態を表すことができる．そこで，i番めの状態ユニットが0以上の値を出力するときにニューラルネットワークが状態q'_iにあるものと定義する．この状態集合を$Q''=\{q'_0, q'_1, \cdots, q'_{q-1}\}$と定義する．

Q'とQ''の間のある全単射写像をΨとする．RHONの定義より，任意の状態ユニットと任意の入

力ユニットから任意の状態ユニットに向かう $n+1$ 次までの結合重みが存在する．したがって，時刻 t において任意の状態 q'_{j_1} にあるニューラルネットワークが任意の入力 x_i を受けた場合に，時刻 $t+1$ において任意の状態 q'_{j_2} へ遷移するように，2次の結合重み $w_{j_2 j_1 i}$ を設定することができる．このことより，Ψ は同型写像となる．

したがって，任意の FSM $M=(Q,\Sigma,\delta,q_0,F)$ に対し，それと同型な RHON が存在する． □

補題2より，次の定理を得る．

定理2 CYK アルゴリズムを実行できる2次の RHON が存在する．

TFSM による CYK アルゴリズムの実行を理解するために，文脈自由文法の1つである逆ポーランド記法（Reverse Polish notation）を用いた実行例を付録に示す．

4.3 判定に要する時間計算量

CYK アルゴリズムによる判定には，語の長さ n に対して $O(n^3)$ の時間計算量を要する．$P(i,j)$ の計算アルゴリズムを以下に示す．

```
   begin
1) for i := 1 to n do
2)     P(i,i) = {A|A ∈ Σ, A → x_i ∈ δ}
3) for l = 1 to n − 1 do
4)     for i := 1 to n − l do
           begin
5)             j := i + l;
6)             P(i,j) = Φ;
7)             for k := i to j − 1 do
8)                 P(i,j) = P(i,j) ⋃ {A|A ∈ Σ,
                   A → BC ∈ δ, B ∈ P(i,k),
                   C ∈ P(k+1,j)}
           end
   end
```

ステップ3)，4) はそれぞれたかだか n 回繰り返される．さらにステップ7)，8) に $O(n)$ 時間を要することから，全体で $O(n^3)$ の時間計算量となる．

RHON によりこれを並列に実行する場合の時間計算量は次のようになる．まずステップ1)，2) に対応する計算は，文字列の長さによらず一定となり $O(1)$ 時間となる．ステップ3) は，RHON においてもたかだか n 回繰り返される．一方，ステップ4) は，文字列の長さによらず一定となり $O(1)$ 時間となる．このことから，ステップ3)，4) で合わせて $O(n)$ 時間となる．ステップ7) は，たかだか n 回繰り返される．以上より，RHON により CYK アルゴリズムを並列に実行する場合の時間計算量は，$O(n^2)$ となる．

RHON を用いて並列化を行うことで，文字列の各アルファベットを生成する非終端記号の決定と，部分列を生成する非終端記号の組合せの決定に要する時間が，文字列の長さによらず一定となる．この特長を用いることで，文脈自由言語の構文解析に要する時間計算量を削減することが期待できる．

5. 結　　論

本論文では，RHON を用いた文脈自由言語の認識法を提案した．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，任意の決定性文脈自由言語を認識できることを示した．また，文脈自由言語の所属判定法である CYK アルゴリズムを，RHON で並列に実行する方法を提案した．これにより，従来 $O(n^3/\log^2 n)$ の計算量を必要とした語長 n の文の判定が $O(n^2)$ の時間計算量で実行できることを示した．

提案した方法によれば，認識する語の長さに依存してニューラルネットワークの構成を変更する必要が生じない．このことは，文脈自由言語の学習アルゴリズムを構成する際に有用な特長である．無限精度のニューロンは現実には構成できないが，有限精度の素子を用いて言語の構造を学習し，その後に必要な精度を確保することで，実質的に受理系を学習することができる．この点を生かし，今後は，文脈自由言語の語の集合から RHON の重みを決定する学習アルゴリズムを明らかにする必要がある．学習にあたっては，RHON の構造をより単純化する必要があり，このために受理系の状態を連続値として実現すること，またプッシュダウンオートマトンの動作を1動作でシミュレートする方法を明らかにすること，などについて検討する必要がある．

今後，任意の非決定性文脈自由言語を認識でき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のクラスを明らかにする必要がある．また，所属判定に必要なニューロンの数は，入出力記号の数に対して組合せ的に増大することから，メモリの値を連続値で表現することで，RHON の規模を縮小する方法を考案する必要がある．

参 考 文 献

1) Cleeremans, A., Servan-Schreiber, D. and McClelleland, J.: Finite State Automata and Simple Recurrent Networks, *Neural Computa-*

tion, Vol.1, No.3, pp.372–381 (1989).
2) Giles, C., Miller, C., Chen, D., Chen, H., Sun, G. and Lee, Y.: Learning and Extracting Finite State Automata with Second-order Recurrent Neural Networks, *Neural Computation*, Vol.4, No.4, pp.393–405 (1992).
3) Hopcroft, J.E. and Ullman, J.E.: オートマトン言語理論計算論 I, サイエンス社 (1984).
4) Liow, R. and Bidal, J.J.: A Dual Network Expert System, *Proc. IJCNN'91 at Singapore*, Vol.2, pp.1670–1674 (1991).
5) Rytter, W.: Fast Recognition of Pushdown Automaton and Context-free Languages, *Information and Control*, Vol.67, pp.12–22 (1985).
6) Santos Jr., E.: A Connectionist Context-free Parser which is Not Context-free, but then It is Not Really Connectionist Either, *The 9th Annual Conference of the Cognitive Science Society*, pp.70–77 (1987).
7) Sun, G., Chen, H., Giles, C. and Lee, Y.: Connectionist Pushdownautomata that Learn Context-free Grammars, *Proc. IJCNN'90*, Vol.1, pp.577–580 (1990).
8) Sun, R.: Neural Network Models for Rule-based Reasoning, *Proc.IJCNN'91 at Singapore*, Vol.1, pp.503–508 (1991).
9) Tanaka, K. and Kumazawa, I.: Learning Regular Languages via Recurrent Higher-order Neural Networks, *Proc. IJCNN'96 at Washington D.C.*, Vol.3, pp.899–907 (1996).
10) Younger, D.: Recognition and Parsing of Context-free Languages in Time n^3, *Information and Control*, Vol.10, No.2, pp.189–208 (1967).
11) 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 (1996).

付 録

TFSM による CYK アルゴリズムの実行例

逆ポーランド記法は, $\Sigma = \{S\}$, $\Gamma = \{+, ., a\}$, $\delta = \{S \to +SS, S \to .SS, S \to a\}$,, $S \in \Sigma$ としたとき, 4 項組 $G_{RP} =< \Sigma, \Gamma, \delta, S >$ で定義される.

G_{RP} の生成規則を Chomsky 標準形で書き表すと, $\Sigma = \{S, A, B, C\}$, $\Gamma = \{+, ., a\}$, $\delta = \{S \to AB, S \to CB, B \to SS, A \to +, C \to ., S \to a\}$, となる. 以後, G_{RP} は Chomsky 標準形で書き表された生成規則からなるものとする.

例として, $+ \cdot aaa$ なる語が与えられた場合の各 FSM の計算過程を図 6 ～図 9 に示す. 図中, 丸は FSM を表し, 実線矢印は FSM 間の結び付きを表す. 点線矢印は外部入力を表す.

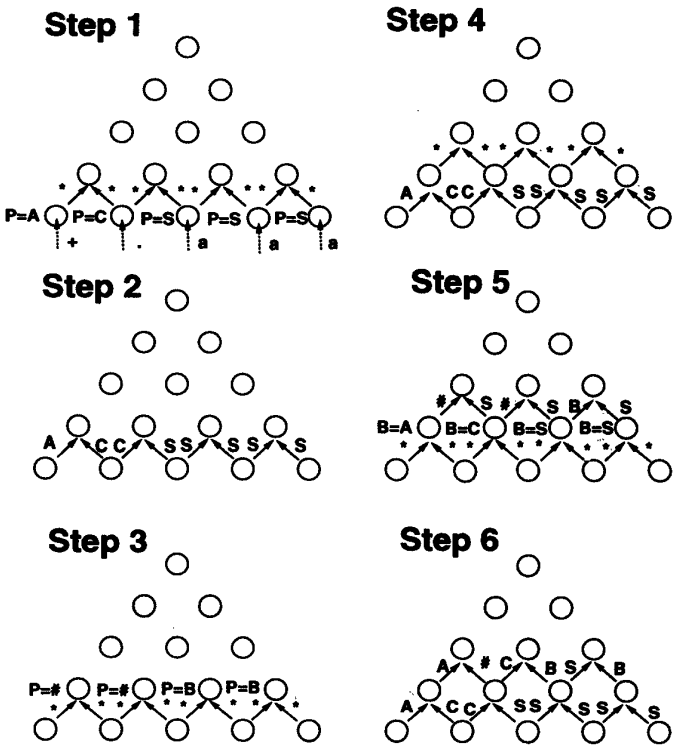

図 6 Step1 から Step6 までの計算過程
Fig. 6 Process of calculations from Step 1 to Step 6.

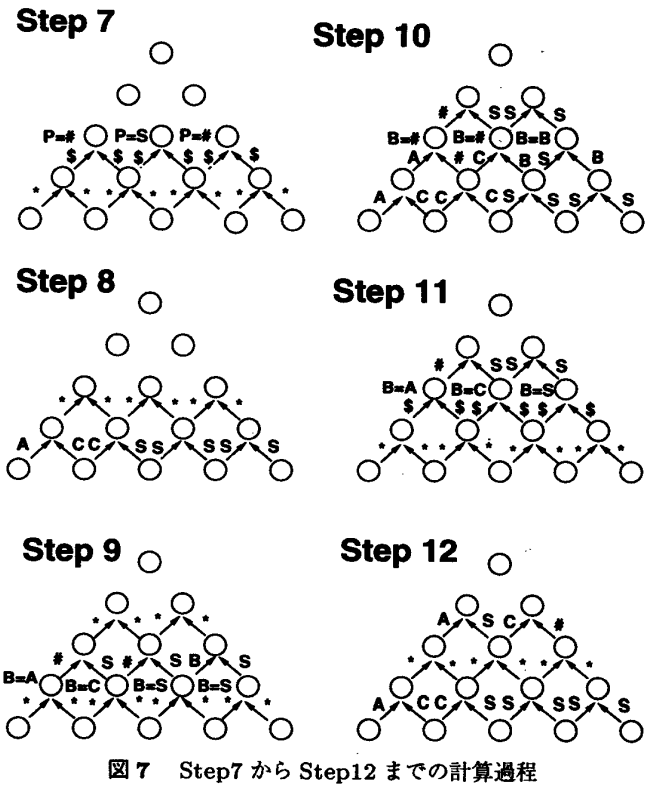

図 7 Step7 から Step12 までの計算過程
Fig. 7 Process of calculations from Step 7 to Step 12.

語 $+ \cdot aaa$ の長さは 5 であることから, 5 つの各アルファベットは 5 段の各 FSM に与えられる. 図中, Step1 で 5 段の各 FSM は入力を受け, 入力に応じて

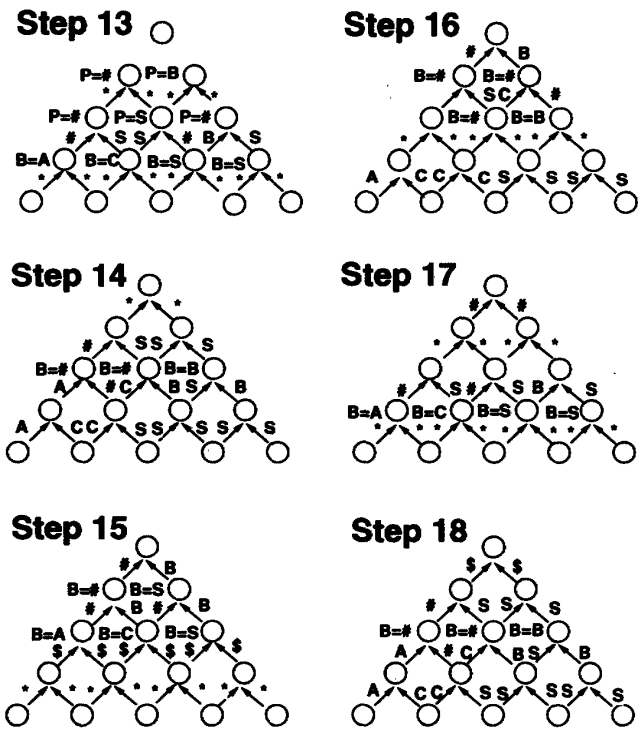

図8 Step13 から Step18 までの計算過程
Fig. 8 Process of calculations from Step 13 to Step 18.

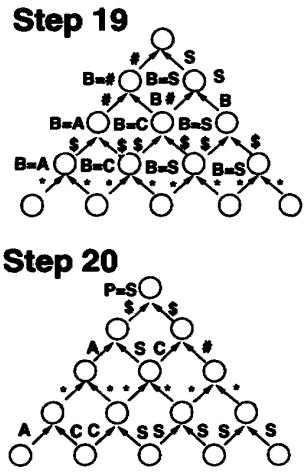

図9 Step19 から Step20 までの計算過程
Fig. 9 Process of calculations from Step 19 to Step 20.

$P(1,1)\cdots P(5,5)$ を計算し P に格納する．同時に4段の各 FSM に動作の開始を伝えるために $*$ を送る．Step2 で5段の各 FSM は P の値を4段目の各 FSM に伝える．Step3 で4段の各 FSM は Step2 の入力に基づき $P(1,2)\cdots P(4,5)$ を計算し P に格納する．Step4 で4段の各 FSM は3段目の各 FSM に動作の開始を伝えるために $*$ を送る．Step5 で4段目の各 FSM は3段目の各 FSM に対し P の要素を1つ伝える．同時に5段目の各 FSM からの入力を B に格納する．Step6 では4段目の各 FSM は3段目の各 FSM に対し P の要素をさらに1つ送り出す．Step7 では4段目の各 FSM は3段目の各 FSM に対し残りの P の要素を伝える．これにより，3段目の各 FSM の P の値が決定され $P(1,3)\cdots P(3,5)$ が計算される．このような動作を繰り返し，Step20 で $P(1,5)$ が計算される．$P(1,5)=S$ であることから，$+\cdot aaa$ は逆ポーランド記法により生成された言語の語であることが分かる．

(平成8年8月22日受付)
(平成9年3月 7 日採録)

田中 賢

平成元年早稲田大学理工学部電気学科卒業．平成3年東京工業大学総合理工学部システム科学修士課程了．平成7年同大学理工学部情報工学博士課程単位取得．同年新潟大学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．

熊沢 逸夫

昭和56年東京工業大学電気電子学部卒業．昭和61年同大大学院博士課程了．同年同大学工学部情報工学科助手．平成2年助教授．パターン認識，信号画像処理，ニューラルネットワークの研究に従事．工学博士．