

制御つきホップフィールド・ニューラルネットワークの 形式言語による特徴化 Language Theoretic Characterization of Controlled Hopfield Neural Network

林原香織 山下雅史 阿江 忠
Kaori Hayashihara, Masafumi Yamashita, Tadashi Ae

広島大学 工学部
Faculty of Engineering, Hiroshima University

[あらまし] 本稿では、ホップフィールド・ニューラルネットワークと有限制御機械が相互作用するマシンを2種類のニューラルネットワーク・マシンとして定義し、その能力の形式言語による特徴化を試みた。1つは、逐次決定過程モデルから、もう1つは、カウンタ・オートマトンモデルから見た定義であるが、いずれのマシンの受理する言語のクラスもCFLの部分クラスであり、さらに、後者のマシンの直積はその個数に関して、無限の階層をなすことが判明した。

[Abstract] *We model a neuro computer, in which Hopfield neural network and the finite control machine interact with each other, as two kinds of neural network machines. One is based on the sequential decision process, and the other on the counter machine. Also we characterize their power by the formal languages, then the class of the language accepted by each machine is a proper subset of CFL. Furthermore, we show that the direct products of the latter machines constructs an infinite hierarchy depending on the number of machines.*

1. まえがき

ニューラルネットワークは脳の神経回路網を模擬した新しい情報処理モデルとして脚光を浴びているが^[1]、その中でもホップフィールド型はTSP(Travelling Salesman Problem)などの組合せ最適化問題の高速解法器としても注目を集めている^[2]。このような背景から、人工的な神経回路網を実現して新しいコンピュータ・システムを構築しようとする研究も盛んである^[3]。

一方、ニューラルネットワークは時系列の処理を不得意とし、ホップフィールド・モデルにおいても局所安定状態(local minima)に陥った場合、何らかの外部入力で制御する必要がある。また、困難な問題に対して、安定状態の集合が正確に解集合に一致するホップフィールド・モデルが構築可能か否かという問題は

否定的と言わざるを得ない^[4]。このような観点から、筆者らはホップフィールド・ニューラルネットワークを通常のノイマン型コンピュータで制御するニューロコンピュータを既に提案し、そのプロトタイプを試作している^[5]。本稿では制御する側のノイマン型コンピュータを有限状態機械に制限した場合のニューロコンピュータを2種類のニューラルネットワーク・マシンとして近似的に定義することを試みた。その結果、それぞれの能力は形式言語により特徴化できる。

一つのマシン・モデルはコスト関数をもつ有限オートマトン(逐次決定過程)であり^[6,7]、もう一つはカウンタ・オートマトン・モデルに基づいている^[10]。前者は、有理数の加法性コストをもつ有限オートマトンの受理言語のクラスに一致し、後者は前者のクラスを

含み、カウンタ・オートマトンの受理言語のクラスに一致する。更に、後者については、その直積がマシンの個数に対して、無限の階層をなすことを示す。これは、後者のマシンに相当する小さい単位のニューロコンピュータを多数個、用意する場合、個数が多いほど能力が高くなることを表している。

なお、本稿では特徴化の第一歩として、ホップフィールド・ニューラルネットワークはデジタル化したものを対象としており、そのサイズも有限に限定している。

2. 制御つきホップフィールド・ニューラルネットワーク

本稿で対象とするニューロコンピュータは図1に示されるようにホップフィールド・ニューラルネットワーク(H-NN)を制御(Ctrl)部で制御するものを想定する。

2.1. H-NN部

H-NNはニューロンの相互結合により構築される。一般的には、各ニューロンは $[0, 1]$ 区間の連続値をとるがここではデジタル化されたニューラルネットワークを想定しているので、各ニューロンは0または1のいずれかの値のみをとるものとする。また、全ニューロンを同期的に観測できるものとし、観測時の各ニューロンの値から作られる要素数 m のベクトルをH-NNの状態とよぶ。ニューロン数 m 個から構成されるH-NNは $n=2^m$ 個の状態をもつ。よって、本稿では、H-NNは n 個の状態を頂点としてもつ超立方体からなる状態空間を作る。ただし、空間の内部は定義されず、頂点のみの離散的状態空間をもつ。H-NNは系全体の特性量としてエネルギー関数をもつ。エネルギー関数は状態により定義され、状態の遷移は時間に関してエネルギー関数が非増加となるように行なわれる。これは超立方体の頂点にエネルギー関数を付随させたとき、ある初期状態を表す頂点から、ハミング距離1のエネルギー関数非増加な頂点へと移動することに対応する。図2(a)に $m=3$ の例を示す。この遷移は、局所安定状態といわれるエネルギー関数の極小値を与える状態までニューロン相互の作用により自律的に行なわれる。デジタル化したホップフィールド・ニューラルネットワークでは入力トリガが与えられ、局所安定状態に落ち着くまでの全過程を1ステップと見ることがもできるが、こ

で議論の対象としているH-NNでは局所安定状態に陥るまでの途中の状態遷移、各々を1ステップと考える。

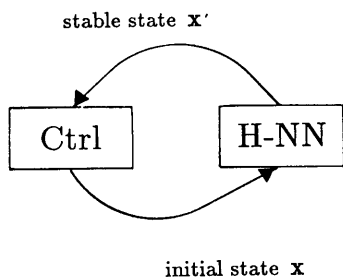
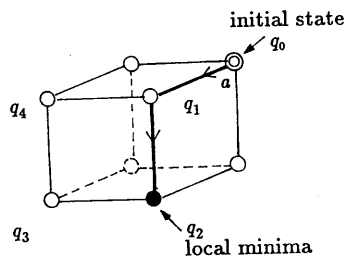
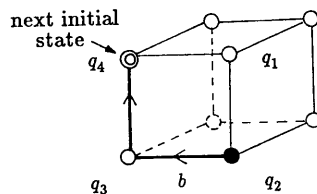


図1. 制御つきホップフィールド・ニューラルネットワーク



(a) H-NNの自律的な遷移



(b) Ctrlによる制御時の遷移

図2. $m=3$ のときの制御つきホップフィールド・ニューラルネットワークの状態遷移

すなわち、1ステップは、超立方体空間の2頂点間の枝に対応し、状態遷移系列は超立方体上の頂点間のパスになっている。1ステップの状態遷移に名前(記号)をつけると、初期状態が与えられたときのH-NNの自律的なふるまい、すなわち、エネルギー非増加な一つの状態遷移は一つの記号系列を与える。従って、H-NNのつくる全ての状態遷移の集合は記号系列の集合、すなわち、言語を生成する。

離散的な状態変化、すなわち、超立方体の頂点間の移動のため、H-NNでは無限の状態変化をもつアナログ・ホップフィールド・ニューラルネットワークに比べ、局所安定状態に陥りやすく、本来のホップフィールド・ニューラルネットワークの能力が過小評価されている。しかも、状態遷移の生成する言語は制御するノイマン型コンピュータの側からの特徴化であるため、ホップフィールド・ニューラルネットワーク自体の特徴化としては弱められている。しかし、本稿のようなアプローチは、ニューラルネットワーク、または、ニューロコンピュータのもつ計算機の処理能力を明らかにする一つの試みといえよう。

2.2. Ctrl部

局所安定状態は必ずしも問題の解に対応するエネルギーの最小値を与えるとは限らない。しかし、H-NNだけではこの状態からの脱出は不可能であるため外部からの制御が必要となる。ここではCtrl部によってそれを行なうが、局所安定状態のもつエネルギーを引き上げて、新たな初期状態を作るまで続けられる。すなわち、Ctrl部による制御時は、H-NNの状態遷移時とは逆に、状態空間において、局所安定状態である頂点から、エネルギー関数非減少方向にハミング距離1の頂点を經由して次の初期状態となる頂点へと遷移していく。図2(b)に $m=3$ の例を示す。

2.3. 制御つきホップフィールド・ニューラルネットワークのモデル

H-NN部および、Ctrl部を融合したモデルとして、次のようなグラフを定義する。

[定義2.1]

ニューラルネットワーク・グラフをコストつき有向グラフ $G = (V, E, C)$ と定義する。

ここで、

$$V = \{v_i = (a_1, a_2, \dots, a_j, \dots, a_m), \mid a_j \in \{0, 1\}, 1 \leq i \leq 2^m\}$$

: ネットワークの状態の集合

$$E = \{(v_i, v_j) \mid \sum_{i=1}^m (a_i \oplus b_i) = 1, \uparrow$$

$$v_i = (a_1, \dots, a_m), v_j = (b_1, \dots, b_m)\}$$

: 状態遷移の集合

$$C: ((v_i, v_j), q) \rightarrow Z \text{ (整数の集合)} : \text{エネルギー差関数.}$$

ただし、

$$(v_i, v_j) \in E,$$

q は制御状態で、次のような非決定性有限オートマトン M の状態である。

$$M = (Q, \Sigma, \delta, q_0, Q_F)$$

ここで、

$$Q = \{q_i \mid v_i \text{ に対応する状態}, 1 \leq i \leq 2^m\}$$

: 制御状態の有限集合

$$\Sigma: \text{制御用入力} \text{の有限集合} (\epsilon \text{ も含む})$$

$$\delta: Q \times \Sigma \times C \rightarrow 2^Q \times C: \text{状態遷移関数}$$

(2^Q は Q の部分集合の集合)

$q \in Q, c \in C$ において、 $a \in \Sigma$ が入力されたとき、 $r \in Q$ へ状態遷移が許され、 $c' \in C$ となる。

$$\Leftrightarrow (r, c') \in \delta(q, a, c).$$

$$q_0 \in Q: \text{初期状態}$$

$$Q_F \subseteq Q: \text{最終状態の集合} \quad \square$$

H-NNなる超立方体状態空間において、2.1で述べたような状態遷移(ハミング距離1の頂点を經由しながらの遷移)は、限られたものとなっているが、その遷移の様子を表現したものがグラフ G である。 G の節点は超立方体の頂点に対応し、H-NN部で状態遷移が起き得るもののみが枝で結ばれる。この遷移はH-NNの自律的な遷移とCtrl部による遷移の両方を意味する。枝にはコストが付けられるが、これは遷移が起きたときのエネルギーの変化に相当する。ただし、状態遷移はすべて制御機械 M によって行なわれるとみなし、 M の制御状態

† v_i, v_j で異なる成分の数。

を Σ の入力によって遷移させると、それに伴い、コストが決まるものとする。一般には、 V の要素数は限定されないが、ここでは有限オートマトンで制御する場合を対象にしているので、ニューロン数は有限に限っている。

この融合モデルにおいて、ある初期状態（初期コストを持つ）から出発するとき、適当な制御用入力列を与えると、H-NN上で問題解決に向かって状態が遷移していく様子を模倣することができる。この入力列を問題を解くための方策とみなす。このとき、方策の決定にコストが陽に関与するとみると、このモデルはカウンタの値を見ながら状態遷移を起こすカウンタ・マシンとして形式化できる。もう一つの見方としてコストが陽に関与しないものが考えられるが、こちらを逐次決定過程として形式化してみる。次節でそれぞれについて詳しく検討していく。

3. ニューラルネットワーク・マシン

本節では2. で提案したニューラルネットワーク・グラフの特徴化のために2種類のモデルを導入し、それぞれ形式化する。ニューラルネットワーク・マシンで問題を解くことはそのための（最適）方策、すなわち、基本的決定（ $\in \Sigma$ ）の系列（ $\in \Sigma^*$ ）が得られることである。H-NNにおいて、方策は、超立方体上の頂点を結ぶパスとみなすことができる。以下では、定義する2種類のニューラルネットワーク・マシンが生成する言語がどの程度になるのかその能力についての検討も行なう。

3.1. 逐次決定過程モデル

[定義3.1.1]

ニューラルネットワーク・マシンNM1を4項組

$$N_1 = (M, h, \xi_0, \theta)$$

と定義する。ここで、

$$M = (Q, \Sigma, \delta, q_0, Q_F) : \text{非決定性有限オートマトン}$$

ここで、

Q : 状態の有限集合

Σ : 入力の有限集合 (ϵ も含む)

$\delta : Q \times \Sigma \rightarrow 2^Q$: 状態遷移関数

(2^Q は Q の部分集合の集合)

$q \in Q$ に $a \in \Sigma$ が入力されたとき、

$r \in Q$ への状態遷移が許される

$$\Leftrightarrow r \in \delta(q, a).$$

$q_0 \in Q$: 初期状態

$Q_F \subseteq Q$: 最終状態の集合

h : コスト関数

($\forall p, q \in Q$) ($\forall a \in \Sigma$) ($q \in \delta(p, a)$) ($\forall \xi_p, \xi_q \in Z$, Z は整数の集合) に対して

$$\xi_q = h(\xi_p, p, q, a) = \xi_p + \psi(p, q, a)$$

$\xi_0 \in Z$: 初期コスト

$\theta \in Z$: しきい値 \square

NM1は直感的にはH-NNの状態を Q で、そのときのエネルギー関数値を ξ で表現したものである。H-NNの遷移前後の2状態のエネルギー差をコスト関数の ψ によって表すと、状態変化に伴うエネルギー関数は、加法性のコスト関数で対応づけることができる。NM1の定義はnd-ra-ap¹の特別な形となっている。なお、遷移を起こすための入力系列は問題を解くための方策で、H-NNの自律的な状態遷移についても遷移ごとに基本的決定がなされたとみなし、その系列も方策の一部をなすものとする。

ところで、 Q_i をH-NNの局所安定状態を表すために Q の部分集合として導入する。H-NNの局所安定状態の中に問題の解が含まれることより、NM1の最終状態の集合 Q_F は Q_i に含まれる必要がある。 Q_i に属す状態へ向かう遷移はH-NNのニューロン相互の作用による自律的な遷移を表す。このときのエネルギー関数は非増加に変化していくが、逆に、 Q_i から出ていく遷移はCtrl部によるH-NNの制御を表し、エネルギーを増加させて次の初期状態へと導く部分である。すなわち、 $a \in \Sigma$ によって、 $q \in Q_i$ に遷移するすべての状態 $p \in Q$ に対して、必ず、 $\psi(p, q, a) \leq 0$ 、逆に、 a によって Q_i から遷移する状態 $r \in Q$ に対しては $\psi(q, r, a) \geq 0$ となっているので、 Q_i に属すすべての状態について ψ を調べることにより有限のステップで Q_i を見つけることができる。本稿では、問題を解くための方策の集合を次のような言語として定義する。

¹ 付録の定義[1]-[6]参照。

[定義3.1.2]

NM1 N_1 の受理集合を次のように定義する.

$$L_1(N_1) = \{x \in \Sigma^* \mid \bar{h}_{q_0}(x) \leq \theta\}$$

ここで[†],

$$\bar{h}_{q_0}(x) = \begin{cases} \min\{\bar{h}_{q_0\sigma}(x) \mid \sigma \in Y(q_0, x), \pi(\sigma) \in Q_i \cap Q_F\} \\ \text{if } \pi(\sigma) \in Q_i \cap Q_F \text{ for some } \sigma \in Y(q_0, x) \\ \infty \text{ otherwise} \end{cases}$$

□

$L_1 = \{L_1(N_1) \mid N_1: \text{NM1}\}$ と記す.

nd-ra-ap と NM1 の受理集合の関係を次の補題に示す.

[補題3.1]

$$L_1 = \Theta_{\text{nd-ra-ap}}$$

(証明)

まず, $L_1 \subset \Theta_{\text{nd-ra-ap}}$ を示す. 任意の NM1

$$N_1 = (M = (Q, \Sigma, \delta, q_0, Q_F), h, \xi_0, \theta)$$

に対して, $L_1(N_1) = A(\Pi)$ ^{††} なる nd-ra-ap Π が構成できることを示せばよい. 最初に, N_1 に対する Q_i を見つける.

$$\Pi = (M' = (Q', \Sigma, \delta', q_0', Q_F'), h', \xi_0', \theta')$$

$$Q' = Q, q_0' = q_0, Q_F' = Q_F \cap Q_i,$$

$$\xi_0' = \xi_0, \theta' = \theta \text{ とし,}$$

$$(\forall p, q \in Q)(\forall a \in \Sigma) \text{ に対して,}$$

$$q \in \delta(p, a) \text{ ならば, } q \in \delta'(p, a),$$

$$\psi'(p, q, a) = \psi(p, q, a),$$

$$h'(p, q, a) = h(p, q, a) = \xi_p + \psi(p, q, a)$$

ξ_p, ψ, θ は整数なので Π は nd-ra-ap となる.

以上の構成により,

$$L_1(N_1) = \{x \in \Sigma^* \mid \bar{h}_{q_0}(x) \leq \theta\}$$

$$= \{x \in \Sigma^* \mid \bar{h}_{q_0'}(x) \leq \theta = \theta'\} = A(\Pi)$$

よって, $L_1 \subset \Theta_{\text{nd-ra-ap}}$.

逆に, $\Theta_{\text{nd-ra-ap}} \subset L_1$ を示す. 任意の nd-ra-ap

$$\Pi = (M = (Q, \Sigma, \delta, q_0, Q_F), h, \xi_0, \theta) \text{ に対して,}$$

$A(\Pi) = L_1(N_1)$ なる NM1 N_1 の構成できることを示せばよい.

ここで, Π に現われる ψ は有限個なので, θ, ξ_0, ψ を M/N (M, N は整数) の形で表したときの分母の最小公倍数をそれぞれにかけることで, 元の Π と等価で整数の Q

スト, θ をもつ nd-ra-ap がつくれる. 以降, 整数のコストに限って考えることにする.

$$Q' = Q, q_0' = q_0, \xi_0' = \xi_0, \theta' = \theta \text{ とし,}$$

$$(\forall p, q \in Q)(\forall a \in \Sigma) \text{ に対して,}$$

i) $q \in Q - Q_F$ のとき,

$$q \in \delta(p, a) \Rightarrow$$

$$q \in \delta'(p, a), \psi'(p, q, a) = \psi(p, q, a).$$

ii) $q \in Q_F$ のとき,

$$\forall p \in Q \text{ s.t. } q \in \delta(p, a) \text{ と } \forall r \in Q \text{ s.t. } r \in \delta(q, b)$$

$$(a, b \in \Sigma) \text{ に対して,}$$

$$a) \psi(p, q, a) \leq 0, \psi(q, r, b) \geq 0$$

$$\Rightarrow \text{i) と同様に } \delta', h' \text{ を設定し,}$$

$$Q_F' \leftarrow \{q\}.$$

b) それ以外 \Rightarrow その q に対して新しい状態 s_q を作る.

$$Q' \leftarrow \{s_q\}, Q_F' \leftarrow \{s_q\}.$$

$$\delta'(q, \varepsilon) = \{s_q\}, \psi(q, \varepsilon, a) = -\infty$$

$$\delta'(s_q, \varepsilon) = \{q\}, \psi(s_q, q, \varepsilon) = \infty$$

$$(\infty \text{ は } |\theta| \text{ に比へ十分大きい値})$$

$$h'(p, q, a) = \xi_p + \psi'(p, q, a) \text{ とすると,}$$

NM1 $N_1 = (M' = (Q', \Sigma, \delta', q_0', Q_F'), h', \xi_0', \theta')$ が構成できる. また, N_1 の Q_i は Π の Q_F と等しい.

$$A(\Pi) = \{x \in \Sigma^* \mid \bar{h}_{q_0}(x) \leq \theta\}$$

$$= \{x \in \Sigma^* \mid \bar{h}_{q_0'}(x) \leq \theta, \pi(\sigma) \in Q_F\}$$

$$= \{x \in \Sigma^* \mid \bar{h}_{q_0'}(x) \leq \theta = \theta', \pi(\sigma) \in Q_i\}$$

$$= L_1(N_1)$$

よって, $\Theta_{\text{nd-ra-ap}} \subset L_1$

以上より, $\Theta_{\text{nd-ra-ap}} = L_1$ □

3.2. カウンタ・オートマトン・モデル

NM1 では, H-NN のエネルギー関数を nd-ra-ap のコスト関数に対応させた形式化を行なったが, 本節では, 別の視点から H-NN を表現し, その能力について考察する.

[定義3.2.1]

ニューラルネットワーク・マシン NM2 を 8 項組

$$N_2 = (Q, \Sigma, C, \delta, h, q_0, c_0, Q_F)$$

と定義する. ここで,

Q : 状態の有限集合

Σ : 入力記号の有限集合 (ε を含む)

C : カウンタの値を表す非負整数の集合

[†] σ, π, Y は付録の定義 [2]-[4] 参照.

^{††} 付録の定義 [5] 参照.

$\delta: Q \times \Sigma \times C \rightarrow 2^Q \times C$: 動作関数

($\forall p, q \in Q$) ($\forall a \in \Sigma$) ($\forall c, c' \in C$) に対して,
 $(q, c') \in \delta(p, a, c) \Leftrightarrow$
 状態 p , カウンタ c で a が入力されると,
 状態 q への遷移が許され, それに応じた
 カウンタ値 c' が決まる.

q_0 : 初期状態

c_0 : 初期カウンタ値

Q_F : 最終状態の集合 \square

NM2は, H-NNのエネルギー関数を1方向片無限テープ(カウンタ・テープ)の値により表現したものである. NM1と同様, H-NNの状態遷移前後のエネルギー差をカウンタ値を移動させることで表す. ただし, カウンタ・テープの左端より, 左への移動は定義されない. また, NM2はカウンタの左端にきたことのみ判断できる. カウンタの左方向への移動は, H-NNの自律的な遷移時のエネルギー変化を, 右方向への移動はCtrl部の制御によるH-NNの状態遷移時のエネルギー変化に対応している.

初期状態 q_0 , 初期カウンタ値 c_0 から出発し, 入力系列 $x \in \Sigma^*$ が加えられた遷移後の状態を $\bar{\alpha}(q_0, x, c_0)$, カウンタ値を $\bar{\kappa}(q_0, x, c_0)$ とかくとする.

[定義3.2.2]

NM2 N_2 の受理集合を次のように定義する.

$$L_2(N_2) = \{x \in \Sigma^* \mid \bar{\alpha}(q_0, x, c_0) \in Q_F, \bar{\kappa}(c_0, q_0, x) = 0\}$$

\square

NM2の受理集合は $L_2 = \{L_2(N_2) \mid N_2: NM2\}$ とかく.

H-NNの局所安定状態の中に, 問題の解を与えるものがあることより, NM2の受理集合はそれに属す入力系列によって, 初期状態から最終状態の集合に遷移し, かつ, そのときのカウンタの値が0になっていることを受理条件としている. この定義において, 同じ形のNM2でも初期カウンタ値により, 受理集合が異なってくる. NM1においても, 初期コスト, しきい値によって同様のことが起こる. これはホップフィールド・ニューラルネットワークが初期状態の与え方により, 解が異なるのと同じようなふるまいを示しているともみることができる. 初期カウンタ値, 初期コストなどはマシンの定義ごとに設定するものとする.

次に L_2 と 1CL の関係を補題3.2に示す.

[補題3.2]

$$L_2 = 1\text{CL}$$

(証明)

任意のNM2 $N_2 = (Q, \Sigma, C, \delta, h, q_0, c_0, Q_F)$ を

1CM $M = (Q', \Sigma, B, \delta', q_0', Z, Q_F')$ でシミュレートできることを示せばよい.

i) 状態 Q' : $Q' = Q \cup \{q_0'\}$

ii) 初期状態 q_0' : N_2 での初期カウンタ値 c_0 を表すため,

$$(q_0, B^{c_0}) \in \delta'(q_0', \varepsilon, Z) \text{ とする.}$$

iii) 動作関数:

($p, q \in Q$) ($a \in \Sigma$) ($c_p, c_q \in C$) に対し, (q, c_q) $\in \delta(p, a, c_p)$ のとき,

case1) $c_q \geq c_p$ のとき

$$(q, B^{c_q - c_p}) \in \delta'(p, a, B) \text{ とする.}$$

case2) $c_q < c_p$ のとき,

$$\delta'(p, a, B) = \{(q, \varepsilon)\}, \quad \delta'(q, a, B) = \{(q, \varepsilon)\} \text{ とする.}$$

iv) 最終状態: $Q_F' = Q_F$

($q' \in Q_F'$) ($a \in \Sigma$) に対し, $\delta'(q, a, B)$ とし,

q' でカウンタは必ず左端となるようにする.

以上の構成により, 任意のNM2をシミュレートする1CMが存在する. 逆に, 任意の1CMに対してそれをシミュレートするNM2が存在することも示せる. よって, NM2と1CMは等価である. \square

3.3. NM1とNM2の関係

本節では, 2通りに定義をおこなったマシンの受理集合の関係について考察を行なう.

[補題3.3]

$$U = \{a^i b^j \mid i \geq 0\} \notin L_1$$

(証明)

文献[7]より,

$$U \notin \Theta_{nd-ra-op}$$

補題3.1より, 明らかに $U \notin L_1$ \square

† 付録の定義[7],[8]参照.

[注意] NM1において, $N_1 = (M, Q_i, h, \xi_0, \theta)$ の値自体は本質的ではない^[7]. なぜなら, 任意の θ' に対し, 次のように構成された $N_1' = (M, Q_i, h', \xi_0', \theta')$ は $L_1(N_1) = L_1(N_1')$ を満たす.

$$\xi_0' = \xi_0 + (\theta' - \theta)$$

$$h'(\xi', p, q, a) = h(\xi' - (\theta' - \theta), p, q, a) + (\theta' - \theta)$$

よって, L_1 と L_2 の関係を議論するとき, NM2の受理集合における条件, カウンタ値0とNM1の受理集合の θ に関する条件を対応づけやすくするため, NM1については, すべて $\theta = 0$ として構成されるものと仮定する.

[定理3.1]

$$L_1 \not\subseteq L_2$$

(証明)

$$NM1 \ N_1 = (M = (Q, \Sigma, \delta, q_0, Q_F), h, \xi_0, 0)$$

に対して, $L_1(N_1) = L_2(N_2)$ なる

$$NM2 \ N_2 = (Q', \Sigma, C, \delta', h', q_0', c_0, Q_F')$$

が構成できることを示す. 最初に N_1 に対する Q_i を見つけておく.

$$(i) \ Q' = Q^+ \cup Q^-$$

ここで, $Q^+ = \{q^+ \mid q \in Q\}$, $Q^- = \{q^- \mid q \in Q\} \cup \{r_q^-\}$

(ii) q_0', c_0 について

$$\xi_0 \geq 0 \text{ のとき, } q_0' = q_0^+, c_0 = \xi_0$$

$$\xi_0 < 0 \text{ のとき, } q_0' = q_0^-, c_0 = -\xi_0$$

$$(iii) \ Q_F' = \{q^+ \in Q^+ \mid q \in Q_F \cap Q_i\} \cup \{r_q^- \in Q^-\}$$

(iv) δ' について

a) $p, q \in Q, \xi_p, \xi_q \in \mathbb{Z}, a \in \Sigma, q \in \delta(p, a)$, p でのカウンタ値 c_p ,

$\xi_q = h(\xi_p, p, q, a) = \xi_p + \psi(p, q, a)$ に対して,

1) $\xi_p \geq 0, \xi_q \geq 0$ のとき

$$\delta'(p^+, a, c_p) = (q^+, c_p + \psi(p, q, a))$$

2) $\xi_p \geq 0, \xi_q < 0$ のとき

$$\delta'(p^+, a, c_p) = (q^-, -c_p - \psi(p, q, a))$$

3) $\xi_p < 0, \xi_q \geq 0$ のとき

$$\delta'(p^-, a, c_p) = (q^+, -c_p + \psi(p, q, a))$$

4) $\xi_p < 0, \xi_q < 0$ のとき

$$\delta'(p^-, a, c_p) = (q^-, c_p - \psi(p, q, a))$$

b) $\{q^- \in Q^- \mid q \in Q_F \cap Q_i\}$ について

$$\delta'(q^-, \varepsilon, c_q) = \{(r_q^-, 0)\}$$

以上のような構成により, コストが正の時は, Q^+ に属す状態をもち, カウンタでそのコストを表す. コストが負のときは, Q^- に属す状態をもち, コストの絶対値をカウンタで表す. さらに, コストが負で $Q_F \cap Q_i$ に属す状態は, 必ず, $L_1(N_1)$ の受理条件を満たすため, その状態に対応する特別な状態

r_q^- を Q^- に作り, カウンタを空にする関数 h' を設定しておく.

$$\begin{aligned} L_1(N_1) &= \{x \in \Sigma^* \mid \bar{\delta}(q_0, x) \in Q_F \cap Q_i, \bar{h}_{q_0}(x) \leq 0\} \\ &= \{x \in \Sigma^* \mid \bar{\delta}'(q_0', x, c_0) \in Q_F', \bar{h}(q_0', x, c_0) = 0\} \\ &= L_2(N_2) \end{aligned}$$

よって, $L_1 \subset L_2$

一方, $U = \{a^i b^i \mid i \geq 0\} \in 1CL$ なので, $U \in L_2$. しかし, 補題3.3より, $U \notin L_1$.

ゆえに, $L_1 \not\subseteq L_2$. \square

4. 複数個のNM2の並列動作について

本節では, NM2に相当する小さい単位のニューロコンピュータを複数個, 並列接続した場合, その能力がどうなるかについて考察を行なう. 並列接続は, NM2の直積として次のように定義される.

[定義4.1]

2個のNM2

$$N_{21} = (Q_1, \Sigma, C_1, \delta_1, h_1, q_{01}, c_{01}, Q_{F1}),$$

$$N_{22} = (Q_2, \Sigma, C_2, \delta_2, h_2, q_{02}, c_{02}, Q_{F2})$$

の直積は N_{2D} として次のように定義される.

$$N_{2D} = (Q, \Sigma, C, \delta, h, q_0, c_0, Q_F)$$

ここで,

$$Q = Q_1 \times Q_2$$

$$C = C_1 \times C_2$$

$$\delta : Q_1 \times Q_2 \times \Sigma \times C_1 \times C_2 \rightarrow 2^{Q_1 \times Q_2} \times C_1 \times C_2$$

$$q_1, q_1' \in Q_1, q_2, q_2' \in Q_2, a \in \Sigma, c_{q_1},$$

$$c_{q_1'} \in C_1, c_2, c_{q_2'} \in C_2 \text{ に対して,}$$

$$i) (q_1', c_{q_1'}) \in \delta_1(q_1, a, c_{q_1}),$$

$$(q_2', c_{q_2'}) \in \delta_2(q_2, a, c_{q_2})$$

$$\Rightarrow (q_1', q_2', c_{q_1'}, c_{q_2'}) \in \delta(q_1, q_2, a, c_{q_1}, c_{q_2})$$

$$ii) (q_1', c_{q_1'}) \in \delta_1(q_1, \varepsilon, c_{q_1}), \delta_2(q_2, \varepsilon, c_{q_2}) = \phi$$

$$\Rightarrow (q_1', q_2, c_{q_1'}, c_2) \in \delta(q_1, q_2, \varepsilon, c_{q_1}, c_{q_2})$$

$$iii) \delta_1(q_1, \varepsilon, c_{q_1}) = \phi, (q_2', c_{q_2'}) \in \delta_2(q_2, \varepsilon, c_{q_2})$$

$$\Rightarrow (q_1, q_2', c_{q_1}, c_{q_2'}) \in \delta(q_1, q_2, \varepsilon, c_{q_1}, c_{q_2})$$

$$q_0 = (q_{01}, q_{02})$$

$$c_0 = (c_{01}, c_{02})$$

$$Q_F = \{(q_1, q_2) \mid q_1 \in Q_{F1}, q_2 \in Q_{F2}\} \quad \square$$

この定義は帰納的に拡張され, k 個のNM2の直積が定義される. これは, 図3で示されるようなNM2の並列接続を表し, 相互に情報を交換することはできない. この定義より次の性質が得られる.

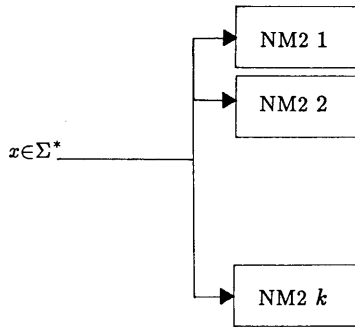


図 3. k個のNM2の直積

[性質4.1]

k個のNM2の直積により受理される言語はそれぞれのNM2により受理される言語の共通集合に等しい。 □

次の定理はNM2のように形式化されたニューロコンピュータを並列接続したとき、その個数が増えるにつれ能力が高くなることを示すものである。

[定理4.1]

k (≥ 2) 個のNM2の直積の受理する言語のクラスに関して、kの値に応じて無限の階層が存在する。

(証明)

補題3.2より、NM2の受理する言語のクラスは1 CMの受理する言語のクラスに等しい。また、NM2の直積の定義より、NM2のk個の直積の受理する言語のクラスはk個の1 CMそれぞれの受理集合の共通集合のつくる言語のクラスであることがいえる。k-CL, k-CFLをそれぞれ、k個の1 CM, PDA (プッシュダウン・オートマトン)の共通集合のつくる言語のクラスとする。次のように定義された言語 $L^{(k)}$ を導入する。

$$L^{(k)} = \{a_1^{i_1} a_2^{i_2} \cdots a_k^{i_k} a_1^{i_1} a_2^{i_2} \cdots a_k^{i_k} \mid i_j \in N, 1 \leq j \leq k\}$$

(Nは非負整数の集合) [9]

文献[8]より、

$$L^{(k)} \in k\text{-CFL}, L^{(k)} \notin (k-1)\text{-CFL}$$

が示されている。

次に $L^{(k)} \in k\text{-CL}$ を示すため、次の言語 L_i を定義する。

$$L_i = \{a_1^i a_2^i \cdots a_n^i \cdots a_k^i a_1^i a_2^i \cdots a_n^i \cdots a_k^i \mid n \in N\}$$

(1 ≤ i ≤ k)

最初に a_i が入力されたときから、カウンタを1増やす動作をn回行なう。それ以外の入力ではカウンタの変動はせず、状態変化のみ行なう。このように構成された1 CMで、その受理集合はカウンタ値が0であることを条件に持つので1 CMで L_i が受理できる。すなわち、

$$L_i \in 1\text{-CL}.$$

また、

$$L^{(k)} = \bigcap_{i=1}^k L_i$$

より、 $L^{(k)}$ はk個の1 CMの受理言語の共通集合として表せる。よって、

$$L^{(k)} \in k\text{-CL}.$$

(k-1)-CL ⊂ (k-1)-CFL で、しかも、 $L^{(k)} \notin (k-1)\text{-CFL}$ が示されているので、

$$L^{(k)} \notin (k-1)\text{-CL}.$$

よって、

$$(k-1)\text{-CL} \subsetneq k\text{-CL}. \quad \square$$

容易に次の系が導かれる。

[系4.1]

k (≥ 2) 個のNM2の直積のつくる言語のクラスはCSLに真に含まれる。

(証明)

$U = \{a^n \mid n \in N\}$ は有限個のCFLの共通集合では表し得ない言語[9]なので、有限個のCLの共通集合でも表せない。しかし、 $U \in CSL$ 。 □

5. むすび

本稿では、ホップフィールド・ニューラルネットワークを有限状態機械で制御するニューロコンピュータを逐次決定過程モデルとカウンタ・オートマトン・モデルとに基づいて、2種類のニューラルネットワーク・マシンとして近似的に定義することを試みた。その結果、それぞれの能力は形式言語により特徴化でき、以下のような結果が得られた。

- (1) 前者のマシンの受理集合は、加法性のコストをもつ有限オートマトン (非決定性モデル) の受理集合のクラスに一致する。

(2) 後者のマシンの受理集合は、カウンタ・オートマトンの受理集合のクラスと一致し、逐次決定過程モデルのマシンの受理集合のクラスを真に含む。

(3) 後者のマシンの $k (\geq 2)$ 個の直積により受理される言語のクラスは k の値に応じて無限の階層ができる。

(3)は、後者のマシンに相当する小単位のニューロコンピュータを並列接続させたとき、接続個数に応じて能力が高くなっていることを示す。既に、筆者らにより提案されたニューロコンピュータAN1では^[5]では、H-NNからなる基本ブロックとその制御用プロセッサを1組としたものを複数個、接続した実現形態をとっているが、各々の組が本稿で形式化した後者のマシンに相当する。よって、AN1全体の能力は、基本ブロックと制御用プロセッサの組の個数が多くなるにつれ、高くなっていくことが示される。

本稿のようにニューロコンピュータを形式言語によって特徴化すると、アナログ・ホップフィールド・ネットワークの無限の状態変化を有限で近似することになり、能力の評価が弱められてしまうことが問題点として残る。今後はこの点の改善を加えたモデリングを行ない、ホップフィールド・ニューラルネットワーク本来の能力評価を検討していきたい。

6. [参考文献]

- [1] IEEE International Conference on Neural Network, San Diego (July, 1987)
- [2] J.J.Hopfield and D.W.Tank; " 'Neural' Computation of decisions in optimization problems", Biological Cybernetics ,52,pp.141 (1985)
- [3] D.E.Rumelhart, J.L.McClelland and the PDP Research Group; Parallel Distributed Processing - Explorations in the Microstructure of Cognition-, Vol.1&2 MIT press
- [4] J.Bruck and J.W.Goodman; "On the power of neural networks for solving hard problems," IEEE Neural Information Processing Systems Conference,(Nov. 1987)
- [5] 阿江,山下,相原,新田;"アルゴリズム駆動型ニューロコンピュータAN1", 信学技報,ICD88-129 (1988)
- [6] 茨木俊秀;"コスト関数をもつ有限オートマトンの受理能力：非決定性モデル",信学論, Vol.J58-D, No.9, pp.562 (1975)

- [7] 室,茨木,長谷川;"加法性コスト関数をもつ有限オートマトンの受理能力：非決定性モデル",信学論, Vol.J62-D, No.7, pp.475 (1979)
- [8] L.Y.Liu and P.Weiner; "An infinitive hierachy of intersection of context-free languages", Mathematical Systems Theory, Vol.7, pp.185(1973)
- [9] L.Y.Liu ;"Finite-reversal pushdown automata", Ph.D. Disseratation, Princeton University (1968)
- [10] P.C.Fischer, A.R.Meyer, and A.L.Rosenberg; "Counter machines and counter Languages", Mathematical Systems Theory, Vol.2, pp.265 (1968)

付録

逐次決定過程については、茨木らの一連の論文で詳しく定義されているが、ここでは、本稿で引用する主なもののみ挙げておく。

[定義1]^[6]

非決定性の逐次決定過程(Nondeterministic Sequential Decision Process; nd-sdp)はシステム $\Pi=(M, h, \xi_0, \theta)$ である。ただし、

$$M=(Q, \Sigma, \delta, q_0, Q_F)$$

: 非決定性有限オートマトン

ここで、

Q : 状態の有限集合

Σ : 入力 of 有限集合 (ϵ も含む)

$\delta: Q \times \Sigma \rightarrow 2^Q$ (2^Q は Q の部分集合の集合)

:非決定的な状態遷移関数

$q \in Q$ に $a \in \Sigma$ が入力されたとき、 $r \in Q$ への状態遷移が許される

$$\Leftrightarrow r \in \delta(q, a).$$

$q_0 \in Q$: 初期状態

$Q_F \subseteq Q$: 最終状態の集合

$h: E \times Q \times Q \times \Sigma \rightarrow E$ (E : 実数の集合)

: コスト関数 $h(\xi, q, r, a)$ はコスト $\xi \in E$ をもつ、ある $q \in Q$ に $a \in \Sigma$ が与えられたとき、遷移可能な任意の $r \in Q$ の状態遷移後のコストを表す。

$\xi_0 \in E$: 初期コスト

$\theta \in E$: しきい値 □

[定義2]^[6]

nd-sdp Π において, 状態 $q \in Q$ に

$x = a_1 a_2 \cdots a_k \in \Sigma^*$ が与えられたときの状態遷移経路 $Y(q, x)$ は次のように定義されるとする.

$$Y(q, x) = \{ r_1 r_2 \cdots r_k \mid r_1 \in \delta(q, a_1), r_2 \in \delta(q, a_2), \dots, r_k \in \delta(r_{k-1}, a_k), r_i \in Q, 1 \leq i \leq k \} \quad \square$$

[定義3]^[6]

$x \in \Sigma^*$ に対し, $q \in Q$ からコスト ξ_q で出発し, 経路 $\sigma \in Y(q, x)$ に沿って遷移したときのコスト $\bar{h}_{q, \sigma(x)}$ は次のように定義される.

$$\begin{aligned} \bar{h}_{q, \mu}(\varepsilon) &= \xi_q \\ \bar{h}_{q, \sigma}(xa) &= h(\bar{h}_{q, \sigma}(x), \pi(\sigma), r, a) \\ \text{ただし,} \\ x &\in \Sigma^*, a \in \Sigma, r \in \delta(\pi(r), a) \quad (\sigma r \in Y(q, xa)), \\ \mu &\text{は長さ } 0 \text{ の経路.} \\ \text{ここで } \pi(\sigma) &\text{は } \sigma \text{ の最後の状態を表す.} \end{aligned}$$

[定義4]^[6]

$x \in \Sigma^*$ が状態 q に加えられたときの Π におけるコスト $\bar{h}_q(x)$ を次のように定義する. ここで,

$$\bar{h}_q(x) = \begin{cases} \min\{\bar{h}_{q, \sigma}(x) \mid \sigma \in Y(q_0, x), \pi(\sigma) \in Q_F\} & \text{if } \pi(\sigma) \in Q_F \text{ for some } \sigma \in Y(q_0, x) \\ \infty & \text{otherwise} \end{cases}$$

□

[定義5]^[6]

Π によって受理される系列の集合 $A(\Pi)$ を

$$A(\Pi) = \{x \in \Sigma^* \mid \bar{h}_{q_0}(x) \leq \theta\}$$

と定義する. □

[定義6]^[7]

nd-sdp Π のコスト関数 h が $r \in \delta(q, a)$ に対して

$$h(\xi, q, r, a) = \xi + \psi(q, r, a)$$

とかけるとき, nd-加法性過程(nd-Additive Process; nd-ap)という. □

nd-ap Π において, $\psi(q, r, a), \xi_0, \theta$ の値が有理数しか許されない場合を有理nd-ap(Rational nd-ap; nd-ra-ap)という.

また, $A(\Pi)$ の集合を次のようにかく.

$$\Theta_{nd-sdp} = \{A(\Pi) \mid \Pi : nd-sdp\}$$

apでも同様に用いる.

[定義7](カウンタ・オートマトン)^[10]

カウンタ・テープを一本もつ1カウンタ・オートマトン(1CM)を7項組

$$M = (Q, \Sigma, B, \delta, Z, q_0, Q_F)$$

で定義する. ここで,

Q : 状態の有限集合

Σ : 入力 of 有限集合 (ε も含む)

$\Gamma = \{B\}$: カウンタ・テープの記号の集合

$\delta: Q \times \Sigma \times \Gamma \rightarrow 2^Q \times \Gamma^*$

Z : カウンタ・テープの初期記号

(テープの左端を表す)

q_0 : 初期状態

Q_F : 最終状態の集合 □

[定義8]

1CM M で受理される言語を

$$1CL(M) = \{x \in \Sigma^* \mid \bar{\delta}(q_0, x) \in Q_F\} \quad \square$$

$1CL = \{1CL(M) \mid M : 1CM\}$ とかく.