

# ニューラルネットワークを利用した論理型推論に関する密結合化手法の提案

加藤 傑, 村越 英樹†

東京都立科学技術大学電子システム工学科†

武田 有志‡

宇都宮大学大学院工学研究科‡

## 1. はじめに

我々は、記号処理と直感的処理の融合を目的とし、論理型言語の記述を論理演算ととらえ、これを Hopfield 型ニューラルネットワークに変換するアルゴリズムを提案し、実装を行った [1], [2]. しかし、その変換アルゴリズムによって生成されるニューラルネットワークは規模が非常に大きく、解を求めるまでに多くのイテレーション回数を要し、収束率が悪い場合、ニューラルネットワークの能力を生かしきれていない。求解能力が悪い原因としては、2入力1出力の論理回路を扱うネットワークをプリミティブとして、これらを多段結合する方法でニューラルネットワークを生成しているため、ネットワークの規模が大きくなり、信号の伝播に時間がかかることがあげられる。そこで、本論文では多入力に対応したプリミティブを導出し、多段となっているネットワークの細胞を削減するためのアルゴリズムを提案する。その結果イテレーション回数を削減し、収束率を改善することができた。

## 2. 従来のアルゴリズムとその問題点

論理型言語 Prolog のニューラルネットワークへの変換ツール [1] は、図 1 に示すように、Prolog プログラムの論理関係を論理回路とみなし、論理和および論理積の演算を行う 2 入力 1 出力を基本としたニューラルネットワークのプリミティブを割り当てることで変換を行う。各プリミティブは、そのエネルギー関数  $E$  において、論理演算の条件が成り立つとき  $E = 0$ 、成り立たないとき  $E > 0$  となるように構成されており、それらの入出力細胞を重ね合わせる事によって組合せ論理回路を実現する。

$N$  入力 1 出力の論理和や論理積においては、ネットワーク規模を抑えるためにも少ない細胞数で実現できることが望まれる。しかし、 $N > 2$  になるとエネルギー関数の条件が厳しくなり、以前のままの条件ではエネルギー関数を形成する各係数を導出することができない。そのため、2 入力 1 出力を扱うプリミティブを段に組み合わせ、 $\log N + 1$  段、 $2N - 1$  個の細胞を必要としていた。その結果、ネットワークの全ての細胞への信号の伝播が遅くなり、求解に至るまでに多くのイテレーション回数を必要とし、収束率が悪くなってしまいう問題があった。

## 3. 細胞削減のためのアルゴリズム

多段となっている細胞を削減し、高密度なニューラルネットワークを生成するため、 $N$  入力 1 出力の論理和もしくは論理積を 1 段、 $N + 1$  個の細胞で実現するプリミティブを導出する。

Prolog の実行系は、バックトラックによるものであり、結果においては最終的に 1 つのパスで実行するのと同じと見なせる。そのため解が出た時点で、たとえ他に正しい実行パスがあったとしても、それは無視することができる。以

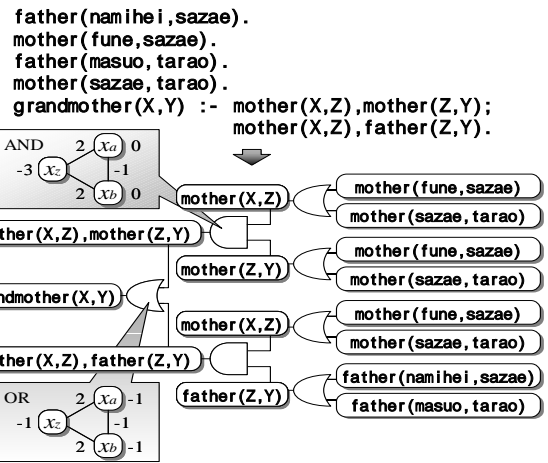


図 1 変換概要

Fig.1 A primary conversion method

上の点から、Prolog における論理和には、排他的論理和を用いることができるので、エネルギー関数を求める際の条件は、表 1 (a) のように設定できる。ここで、 $M$  は入力が 1 である細胞の数、 $x_i$  は出力細胞の状態を表している。 $M > 2$  かつ  $x_z = 1$  のとき  $E > 0$  にしているのは、本来の論理和の条件  $E = 0$  に加え、排他的論理和の条件  $E > 0$  を許容できるからである。

Hopfield 型ニューラルネットワークは、(1) 式で示すように、2 次形式で表された評価関数の最小値を求めることにより、最適化問題を解くことができる。

$$E \equiv -\frac{1}{2} \sum_{ij} w_{ij} x_i x_j - \sum_i v_i x_i + C \quad (1)$$

ここで、ネットワークの各係数には、任意の出力細胞数  $N$  に対応させることを踏まえ、対称性があるものと仮定する。そこで、 $w_{ij}, w_{iz}, v_i$  のそれぞれの値は添字に依存せず、全て同じになるとすると、表 1 (a) で示した論理和の条件と (1) 式から、エネルギー関数の各係数は以下のように帰着できる。

$$\begin{cases} C = 0, & v_z < 0, & v_i < 0 \\ w_{iz} + v_i + v_z = 0, & \frac{M-1}{2} w_{ij} + v_i < 0, \\ M \left( \frac{M-1}{2} w_{ij} + w_{iz} + v_i \right) + v_z \leq 0 \end{cases} \quad (2)$$

$(M = 2, 3, \dots, N)$

これらの条件を満たす解を視察により求めると、 $N$  入力 1 出力の論理和の動作をするプリミティブのエネルギー関数の 1 つとして：

$$E_{or} = \frac{1}{2} \sum_{ij} x_i x_j - 2 \sum_i x_i x_z + \sum_i x_i + x_z \quad (3)$$

が得られる。入力細胞の状態に対して求めた  $E_{or}$  の値を表 1 (a) に示しており、確かに条件を満たしていることがわかる。

\* A Cell Reduction Algorithm for the Logic Inference System using Neural Networks

† Suguru Kato, Hideki Murakoshi

Dept. of Electronic Systems Eng, Tokyo Metropolitan Institute of Technology.

‡ Yuji Takeda

Faculty of Engineering, Utsunomiya Univ.

論理積についても同様にして 条件を設定すると表 1 (b) のようになる . 表 1 (b)で示した論理積の条件と(1)式より , エネルギー関数の各係数は以下のように帰着できる .

$$\left\{ \begin{array}{l} C = 0, \quad v_z < 0, \quad \frac{M-1}{2} w_{ij} + v_i \leq 0, \\ M \left( \frac{M-1}{2} w_{ij} + w_{iz} + v_i \right) + v_z < 0, \\ \frac{N-1}{2} w_{ij} + v_i < 0, \\ N \left( \frac{N-1}{2} w_{ij} + w_{iz} + v_i \right) + v_z = 0 \end{array} \right. \quad (4)$$

(M = 1, 2, \dots, N - 1)

先程と同様に , 条件を満たす解を視察により求めると , N 入力 1 出力の論理積の動作をするプリミティブのエネルギー関数の 1 つとして :

$$E_{and} = -2 \sum_i x_i x_z + \sum_i x_i + N x_z \quad (5)$$

が得られる . 入力細胞の状態に対して求めた  $E_{and}$  の値を表 1 (b)に示しており , 確かに条件を満たしていることがわかる .

論理積が成り立つ条件は , 全入力が 1 のとき出力が必ず 1 に , 全入力が 0 のときは出力が必ず 0 になることである . そのため Prolog の実行がバックトラックであるという特徴から , 素子の入出力に対して細胞を割り当てることなく , 論理積の動作をさせることが可能である . つまり , すべての入力細胞を重ね合わせて 1 つにしても , 論理積の動作を行わせることができる . しかしながら , 変項を割り当てている細胞に対しては解を得るために存在させておく必要があるため , 重ね合わせることはできない . したがって , 変項を割り当てた細胞同士の論理積には (5) 式で示した関係を持つプリミティブを用い , 規則を割り当てた細胞同士の論理積には , 重ね合わせて 1 つの細胞に置き換える .

#### 4 . 評価

提案するアルゴリズムの動作を確認するため , ネットワークコンパイラ [1] に実装し , 正常に動作することを確認した . 従来のアルゴリズムと比較および評価するために , 以下の 3 つのネットワークを対象とし , 問題は n-queen 問題を取り上げる .

- (a) 従来のアルゴリズムによるネットワーク
- (b) 提案するアルゴリズムによるネットワーク
- (c) 手で作成したニューラルネットワーク

まず , 細胞数で評価する . (a) , (b) は Prolog プログラムから生成したネットワークであり , クイーン数 N に対し , それぞれ  $2N^2 - 1$  ,  $N^2 + 1$  個の細胞で構成された . 一方 , (c) はチェス盤のそれぞれのマス目に対して 1 つの細胞を割り当てて作成する一般的なネットワークであり ,  $N^2$  個の細胞で構成できる . よって , (a) は (c) に対して  $N^2 - 1$  個 , (b) は (c) に対して 1 個の差となる . 本手法により , 多段となっている細胞を削減し , 従来と比べてネットワークの規模を十分に抑えることができた .

次に , 収束率 , 平均イテレーション回数で評価する . 本論文では , 比較的単純なボルツマンマシンを採用した [3] . このボルツマンマシンで最適化問題を解く際 , 焼きなまし (Simulated Annealing) と呼ばれる手法が利用される . 本論文では , このとき設定する時刻  $t$  の温度を  $T(t) = T_0 / (1+t)$  とした . ここで ,  $T_0$  は温度の初期値 ,  $t$  はアニーリングにおける冷却の速さを決める定数である .

表 2 はその実行結果である . 収束率は (a) がおよそ 30 ~

表 1 入出力細胞の状態に対する条件

Table 1 Constrains for input and output cell state

#### (a) 論理和

$M^*$	$x_z$	constains	$E_{or}$
0	0	$E = 0$	0
0	1	$E > 0$	1
1	0	$E > 0$	1
1	1	$E = 0$	0
2, 3, \dots, N	0	$E > 0$	$M(M+1)/2$
2, 3, \dots, N	1	$E \geq 0$	$M(M-3)/2 + 1$

#### (b) 論理積

$M^*$	$x_z$	constains	$E_{and}$
0	0	$E = 0$	0
0	1	$E > 0$	N
1, 2, \dots, N-1	0	$E \geq 0$	M
1, 2, \dots, N-1	1	$E > 0$	N-M
N	0	$E > 0$	N
N	1	$E = 0$	0

\*M: number of cells where  $x_i = 1$

表 2 n-queen 問題における収束率とイテレーション回数

Table 2 Convergence rate and average number of iterations in n-queen problems

N	$T_0$		conv.rate (%)			ave.num.of iterations		
			(a)	(b)	(c)	(a)	(b)	(c)
4	1	1	74	100	100	58.4	10.8	9.1
5	1	1	64	100	100	39.0	6.4	6.3
8	10	2	61	100	100	914.3	87.0	81.7
10	10	5	57	100	100	1724.7	232.4	213.8
16	20	1	36	100	100	2058.1	144.6	135.4
32	20	1	30	100	100	2774.3	156.6	150.8

70% であるのに対し (b) はすべて 100% となり , 平均イテレーション回数は従来のアルゴリズムに対して , およそ 5 ~ 17 倍向上し , (c) とほぼ同じになった .

#### 5 . むすび

本論文では , 以前我々が提案したツールにおいて , 論理型言語 Prolog を Hopfield 型ニューラルネットワークに変換する際 , 従来に比べ高密度なニューラルネットワークを生成するためのアルゴリズムを提案した . その結果 , 細胞数およびイテレーション回数を減少させ , 手で作成したニューラルネットワークとほぼ同性能のネットワークを構築できた . 今後の課題としては , 様々な問題への適用 , 解析を行い , より高次の知識処理への組合せによる実用的なツールの実現等があげられる .

#### 参考文献

- [1] 武田 有志 , 川原 英哉 , 菊池 之裕 , 村越 英樹 , 舟久保 登 " 論理型推論のニューラルネットワークによる実現 , " 信学論 (D-II) , vol.J84-D-II, no7, pp.1456-1463, July 2001.
- [2] 加藤 傑 , 村越 英樹 , " Prolog からのニューラルネットワーク生成ツールを用いた生産スケジューリングについて , " 平 13 電学 C 部門大会 , 分冊 1 , no.TC-5-5, pp.45-48, Sep.2001.
- [3] GE.Hinton, T.J. Sejnowski and D.H. Ackley, "Boltzmann Machines: Constraint Satisfaction Networks That Learn," Tech. Rep. CMU-CS-84-119, CarnegieMellon University, 1984.