

自由結合ネットワークと創発的学習手法

Mutually Connected Neural Network and Emergent Learning Algorithm

橋本武法*¹
Takenori Hashimoto

中山功一*¹
Koichi Nakayama

*¹ 佐賀大学大学院 工学系研究科 知能情報システム学専攻
Graduate School of Science and Engineering, Saga University

This paper describes Mutually Connected Neural Network (MCNN) and Emergent Learning Algorithm that is applied for MCNN. MCNN has no restriction on number of nodes in it, and links in MCNN are not restricted by layers. The Emergent Learning Algorithm uses evaluation value of each node, each link, and whole network. The experimental results show that the MCNN obtains optimal logic circuit and memory circuit by Emergent Learning Algorithm.

1. はじめに

人間の脳は数百億といわれる神経細胞がネットワークを構成して、高度な知能を実現している。計算機の進歩に伴い、数百億の神経細胞ネットワークのシミュレートが可能になりつつある。一方、そのような膨大な数の神経細胞を、特定の目的に向けて最適化(学習)することは困難である。

筆者らの取り組んでいる創発的学習手法[中山 02]では、システムを構成する要素の自律分散的な最適化と、全体システムの最適化を組み合わせる。これにより、最適化性能がネットワークの規模に依存しないという意味でスケールフリーであり、また、部分最適ではなくネットワーク全体にとっての最適化を実現する手法の構築を目指している。

これまでの最適化手法では、解空間(探索空間)の増大により、計算時間が指数的に増加する欠点があった。解空間の増大に対し、計算時間の増加が線形となる創発的学習手法の実現により、数百億の神経細胞からなる人間の脳のような巨大なネットワークの最適化が実現でき、知能の創発へとつながると期待される。本稿では、創発的学習手法の実現に向けた第一歩として、自由結合ネットワークを創発的学習手法により最適化し、結果について考察する。

2. 自由結合ネットワーク

2.1 ネットワークの構成

本研究で用いる自由結合ネットワークは、一般的なニューラルネットワークであるパーセプトロンと同様に、入力層/中間層/出力層からなり、各層には複数のノードが存在する。ただし、中間層に存在するノード(以下、中間ノード)同士は結合(ループを含む)を許すという点でパーセプトロンとは異なる。

全てのノード i は、発火の有無 P_i 、入力信号の合計 S_i 、閾値 T_i を持つ。また、全ての中間ノードは、評価値 EN_i を持つ。全てのリンク j は、重み W_j を持つ。また、中間ノードから出る全てのリンクは、評価値 EL_j を持つ。

本稿のネットワークでは、入力層に存在するノード(以下、入力ノード)を全ての中間ノードに、中間ノードを全ての出力層に存在するノード(以下、出力ノード)にリンクする。中間ノード同士のリンク数は、初期化時に、そのネットワークに存在する中間ノードの半数が平均となる二項分布に従い決定する。決定したり

リンク数に応じて、ランダムに選択した中間ノード同士をリンクする。全てのリンクは指向性を持つ。このノードとリンクの集合を 1 個体とする。各個体 k は、評価値 EA_k を持つ。

初期化時に、全ての P_i および全ての S_i を 0 とする。また、全てのリンクの重み W_j を、-1 から 1 までの連続値の間でそれぞれランダムに決定する。全てのノード、リンクおよび個体の評価値を 0 とする。

全ての入力ノードの閾値は、常に 0.5 とする。全ての出力ノードの閾値は、常に 0.99 とする(受けた信号の合計値が 1 以上の場合に発火させるため)。中間ノードの閾値は、初期化時に、-1 から 1 までの連続値の間でそれぞれランダムに決定する。

2.2 ネットワークの入出力

ネットワークへの入出力は、以下の(1)から(3)を 1 ステップとして実行する。

(1) 入力ノードへの伝達

入力ノード i は、受けた信号が閾値を超えた場合に発火する ($P_i=1$)。受けた信号が閾値以下である場合には発火しない ($P_i=0$)。

(2) 入力ノードから中間ノードへの伝達

発火した全ての入力ノードから、全ての中間ノードへ、リンクの重みに従って信号を伝達し、中間ノードの S_i に加算する。中間ノード i は、 S_i が閾値を超えた場合、発火する ($P_i=1$)。全ての入力ノードからの伝達完了後に、全ての中間ノードについて、 $S_i=0$ とする。

(3) 中間ノードから中間ノードおよび出力ノードへの伝達

発火している全ての中間ノード i は、それぞれのリンクの重みに従ってリンク先の中間ノードおよび出力ノードへ信号を伝達する。各中間ノード i は、中間ノードから受けた信号を合計値 S_i に累積する。 S_i の値は、次のステップへ持ち越される。

各出力ノード i について、受けた信号の合計値が閾値を超えた場合、発火する。出力ノード i が発火した場合は 1 を、そうでない場合は 0 を、出力値として返す。

3. 創発的学習手法

本稿で用いる創発的学習手法では、個体のみならず、個体もつ個々の中間ノードおよび中間ノードから信号を伝達する個々のリンクにも評価値を与える。そのため、個体間の評価値の差による学習のみならず、個体内のノード間の評価値の差、およびリンク間の評価値の差による学習が存在する。

連絡先: 中山功一, 佐賀大学, 〒840-8502 佐賀市本庄 1,
TEL: 0952-28-8597, e-Mail: knakayama@is.saga-u.ac.jp

3.1 アルゴリズムの概要

<(1)新規個体の作成>

2.1 節で示したネットワークを持つ n 個体を作成する。

<(2)個体ごとの入出力処理>

2.2 節で示した手順で入出力する。ただし、入力には n 個体全てに対して共通とする。

<(3)評価値の更新>

個体とノードおよびリンクの評価値を更新する。ノードとリンクの評価値の更新の詳細は 3.2 節で、個体の評価値の更新の詳細は 3.3 節で述べる。

<(4)学習>

更新した評価値に基づき、個体とリンクおよびノードをそれぞれを進化的に学習する。リンクとノードの学習の詳細は 3.2 節で、個体の学習の詳細は 3.3 節で述べる。

<(5)繰り返し>

2~4 を 1 世代とし、指定の世代数になるまで繰り返す。

3.2 ノードとリンクの評価値の更新

出力ノードへ信号を伝達する中間ノードの集合を X 、中間ノード X に信号を伝達する中間ノードの集合を Y 、中間ノード Y に信号を伝達する中間ノードの集合を Z とする。図 1 に、中間ノード X, Y, Z およびリンク k, l, m の関係を示す。

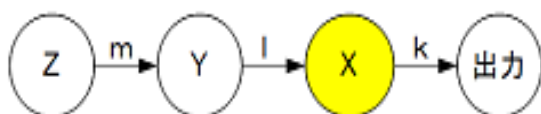


図 1: 中間ノード X, Y, Z およびリンク k, l, m の関係

X の評価値の更新を(A)に、 Y の評価値の更新を(B)に、 Z の評価値の更新を(C)に示す。このとき、処理の対象となるノードが複数存在する場合には、対象となる全てのノードに対して、重複を許して処理する。ただし、以下の式において、ノードの評価値が 0 以下となった場合、その評価値は全て 0 とする。リンクの評価値は 0 以下となり得る。

<A-1: 正解が 1 である出力ノード j に信号を伝達した中間ノード X >

処理: 出力ノード j に信号を伝達した中間ノード X とそのリンク k の評価値に、そのリンク k の重み w_k と重要度補正係数 α の積を加算する(式 1)。

$$\begin{aligned} E_X(t+1) &= E_X(t) + \alpha_j w_k \\ E_k(t+1) &= E_k(t) + \alpha_j w_k \quad (\text{式 1}) \end{aligned}$$

α は、正解の重要度補正係数であり、出力ノード j の正解が 1 である比率を $n(0 < n \leq 1)$ としたとき、 $\alpha = 1/n$ とする。

例えば、100 回中 99 回の正解が 0、1 回の正解が 1 である場合を考え、常に 0 を出力すると正解率が 0.99 となる。一方、1 回しかない 1 を正解した場合の価値は、0 を正解した場合の価値に比べて明らかに高い。具体的には、全部 0 と出力した正解率 0.99 と、1 回しかない 1 を正解した上で 1 回不正解であった

正解率 0.99 では、後者のほうが価値が高いといえる。そこで、正解の重要度を考慮して、更新する値を補正する。

<A-2: 正解が 0 である出力ノード j に信号を伝達した中間ノード X >

処理 1: 出力ノード j に正の出力をした中間ノード X とそのリンク k の評価値から、そのリンク k の重み w_k と正解を 0 としたときの重要度補正係数 α の積を減算する(式 2)。

$$\begin{aligned} E_X(t+1) &= E_X(t) - \alpha_j w_k \\ E_k(t+1) &= E_k(t) - \alpha_j w_k \quad (\text{式 2}) \end{aligned}$$

処理 2: 出力ノード j に負の出力をした中間ノード X とそのリンク k の評価値から、そのリンク k の重み w_k の 1/3 と重要度補正係数 α の積を加算する(式 3)。

$$\begin{aligned} E_X(t+1) &= E_X(t) + \alpha_j (w_k / 3) \\ E_k(t+1) &= E_k(t) + \alpha_j (w_k / 3) \quad (\text{式 3}) \end{aligned}$$

<B: 中間ノード Y >

処理: 中間ノード X に出力した中間ノード Y とそのリンク l の評価値に、リンク l の重み w_l と中間ノード X の評価値の変化 β_j の積の 1/3 だけ加算する(式 4)。

$$\begin{aligned} \beta_j &= |E_j(t) - E_j(t-1)| \\ E_Y(t+1) &= E_Y(t) + (\beta_j w_l / 3) \\ E_l(t+1) &= E_l(t) + (\beta_j w_l / 3) \quad (\text{式 4}) \end{aligned}$$

<C: 中間ノード Z >

処理: 中間ノード Z とそのリンク m の評価値に、リンク m の重み w_m とリンク l の重み w_l と β_j の積の 1/6 だけ加算する(式 5)。

$$\begin{aligned} E_Z(t+1) &= E_Z(t) + (\beta_j w_m w_l / 6) \\ E_m(t+1) &= E_m(t) + (\beta_j w_m w_l / 6) \quad (\text{式 5}) \end{aligned}$$

3.3 個体の評価値の更新

個体の評価値 E_{Ai} として、各中間ノードの評価値の平均 E_{Avg} に、個体全体の正解率 $C(0 \sim 1)$ の 3 倍を加算し、個体の中間ノード 1 個につき 0.3 だけ減算する(式 6)。ただし、 E_{Ai} が 0 以下のときは、 $E_{Ai} = 0$ とする。

$$\begin{aligned} E_{Avg} &= \left(\sum_{i=1}^n E_i \right) / n \\ E_{Ai} &= E_{Avg} + 3C - (0.3n) \quad (\text{式 6}) \end{aligned}$$

3.4 ノードとリンクの更新

個体の学習には、以下の 5 つの手順からなる進化的手法を用いる。

(1) 個体の順位づけ

個体の評価値に基づき、順位をつける。

(2) ノードの操作

(2-1) 相対的に評価値の低いノードの消去

全個体について、個体の評価値の 4 分の 1 に満たない評価値の中間ノードを全て消去する。そのノードから出るリンクおよびそのノードに入るリンクも消去する。

(2-2)最低評価値ノードの消去

順位が下位 50%である各個体において、評価値が一番低い中間ノードをそれぞれ 1 個、消去する。該当する中間ノードが複数存在する場合、それらの中からランダムに選択した 1 個を消去する。

(2-3)評価値の高いノードの複製

全個体の全中間ノードの中で、評価値が一番高いノードを、順位が下位 50%である全ての個体にそれぞれコピーする。ただし、中間ノード同士のリンクについては、コピー元の個体とコピー先の個体のもつネットワークの構造の違いから、元のままを複製するのは困難である。そのため、コピー先の中間ノードとのリンクは、ランダムに生成する。

(3)リンクの操作

(3-1)評価値の低いリンクの削除

中間ノード同士のリンクについて、評価値がマイナスであるリンクを消去する。

(3-2)評価値の高いリンクの更新

中間ノード同士および中間ノードから出力ノードのリンクについて、評価値がプラスであるリンクの重みを 1.2 倍にする。ただし、重みが-1 以下になる場合は-1 に、1 以上になる場合は 1 とする。

(3-3)評価値の低いリンクの更新

中間ノードから出力ノードへのリンクについて、評価値がマイナスであれば、重みが正の場合は 0.1 減算し、重みが負の場合は 0.1 加算する。

(4)保有するノード数の調整

各個体について、その個体よりも順位の高い個体が保有するノード数の平均値を算出する。この平均値から、その個体の保有するノード数を引いた数をギャップ G とする。

(4-1)ギャップが 1 以上($G \geq 1$)の場合、あるいは、ギャップが 0 以上 1 未満($0 < G < 1$)でかつ正解率が 1 に満たない場合、ランダムに中間ノードを 1 個生成し、その個体に加える。

(4-2)ギャップが 0 未満($G < 0$)の場合、1/2 の確率で、その個体内で評価値の一番低い中間ノードを 1 個消去する。該当する中間ノードが複数存在する場合、ランダムに 1 個を選択し消去する。

(5)評価値の低い個体の初期化

10 世代毎に、順位が下位 50%である個体を消去し、再度作成する。

4. 実験

4.1 論理演算回路の学習

学習により、0 または 1 の 2 つの入力から、論理演算の結果を出力するように、ネットワークを最適化する。個体数 $n=10$ 、入力ノード数は 2、出力ノード数は 3 とする。また、各世代において 1 ステップの学習とする。

2 つの入力ノード(以下、入力 1、入力 2)からの入力値に応じて、3 つの出力ノード(以下、出力 1、出力 2、出力 3)へ出力する。出力ノードのうち、出力 1 は入力 1 と入力 2 の論理積、出力 2 は入力 1 と入力 2 の論理和、出力 3 は入力 2 の否定と一致した場合を正解とする。即ち、ある入力 $\{(In1, In2)\}$ に対する正解の出力は、出力 1 が $In1 \wedge In2$ 、出力 2 が $In1 \vee In2$ 、出力 3 が $\neg In2$ である。

実験結果として図 2 に、世代数ごとの、個体の正解率の推移を示す。各世代における、10 個体のうちの最高値を青線で、平均値を赤線で、最低値を黄線で示す。

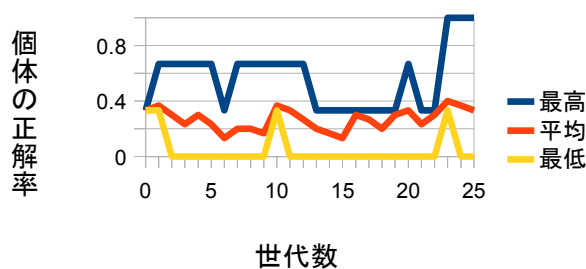


図 2: 個体の正解率の推移

正解率の平均は、0.33(1/3)程度で安定した。正解率の最高は、22 世代目までは 0.33(1/3)~0.67(2/3)であったが、23 世代目に正解率 1 の個体が出現した。

図 3 に、世代数ごとの、個体の評価値の推移を示す。

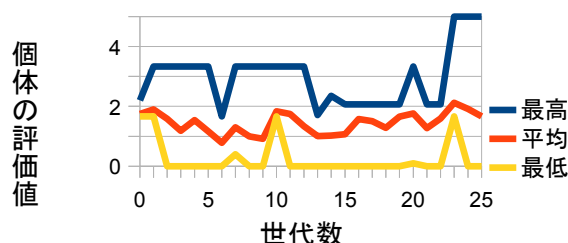


図 3: 個体の評価値の推移

正解率の推移とおおよそ一致した。これは、学習過程でノード数の変化が少ないためであると考えられる。

図 4 に正解率が 1 である個体が学習したネットワーク構造の例を示す。ノードの中の数字は閾値、リンクの横の数字はリンクの重みを表す。

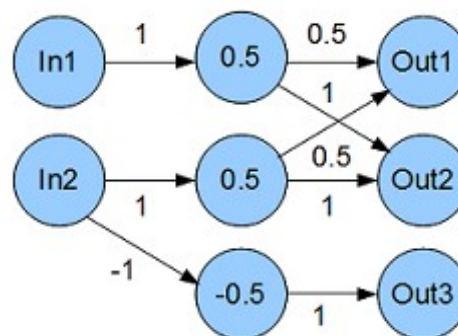


図 4: 正解率が 1 である個体のネットワーク構造の例

以上の結果から、2 つの入力値に対する論理演算の結果を出力するようなネットワーク構造を学習できたと言える。

なお、本実験で得られた論理和、論理積、否定の組み合わせにより、理論上、全ての論理回路が実現できる。すなわち、最適化性能がネットワークの規模に依存しない創発的学習法を実現し、巨大な自由結合型ネットワークを最適化できれば、既存の計算機上で実行可能なあらゆるプログラムを設計できる可能性

が広がる。また、いまだに実現の目途が立たない計算機による人の知能の実現へとつながると期待される。

4.2 メモリ回路(フリップフロップ回路)の学習

本実験では、創発的学習手法によりフリップフロップのようなメモリ回路となるネットワーク構造を得られるように、ネットワークを最適化する。表 1 に、2つの入力に対して正解となる出力を示す。入力 1 がメモリのセット、入力 2 がメモリのリセットに対応する。

学習により、0 または 1 の 2 つの時系列的な入力から、記憶した結果を出力するように、ネットワークを最適化する。個体数 $n=100$ 、入力ノード数は 2、出力ノード数は 1 とする。また、各世代において 100 ステップの学習とする。

入力1	入力2	出力
1	0	1
1	1	0
0	1	0
0	0	前ステップの出力と同値

表 1: 正解である出力のテーブル

実験結果として図 5 に、世代数ごとの、個体の正解率の推移を示す。各世代における、100 個体のうちの最高値を青線で、平均値を赤線で、最低値を黄線で示す。

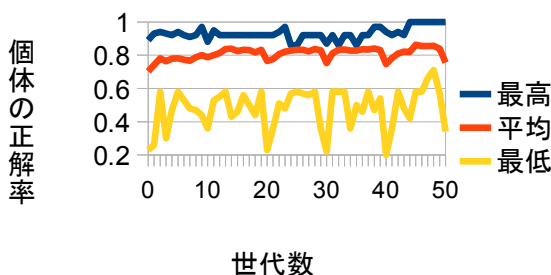


図 5: 個体の正解率の推移

正解率の平均は、初期状態では 0.7 程度であったが、10 世代前後で 0.8 程度になり、安定した。

44 世代目で初めて正解率 1 の個体が出現した。

図 6 に、世代数ごとの、個体の評価値の推移を示す。

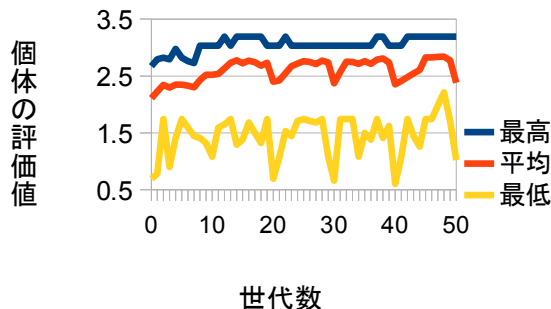


図 6: 個体の評価値の推移

評価値の最高と平均は、10~15 世代目くらいまで上昇し、その後安定した。

図 7 に正解率が 1 である個体が学習したネットワーク構造の例を示す。

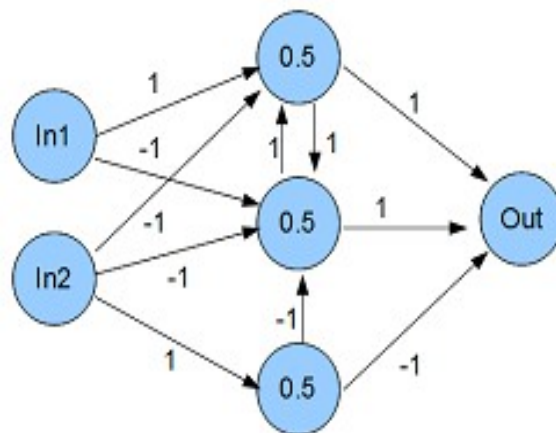


図 7: 正解率が 1 である個体のネットワーク構造の例

以上の結果から、フリップフロップのようなメモリ回路に相当するネットワーク構造を、創発的に学習できたと言える。

既存の計算機では、計算を担当する部分(CPU など)と、記憶を担当する部分(メモリなど)がハードウェア上で区別されている。一方、人間の頭脳では、この両者が、神経ネットワークという共通の要素の相互作用で実現されている。本実験では、事前に機能を設定されていない共通の要素から、計算機能と記憶機能という異なる機能を実現した。この結果は、自由結合ネットワークの創発的学習手法により、人間の頭脳のように単純な要素の相互作用から複雑な機能を生み出す可能性を示唆している。

5. おわりに

本稿では、創発的学習手法を適用した自由結合型ニューラルネットワークについて述べた。個体間の評価値の差による学習と、個体内におけるノードとリンクの評価値の差による進化的学習とを組み合わせた学習手法により、論理演算回路およびメモリ回路を実現した。

本稿の結果からでは、最適化性能がネットワークの規模にどの程度依存しているか(最適化性能のスケールフリー性)は明らかではない。このため、数百億の神経細胞からなる人間の脳のような、解空間が巨大となるネットワークの最適化が実現できるかは未知数である。しかし、少なくとも小さなネットワークに関しては、創発的学習手法により、計算機能と記憶機能が単純かつ均一な中間層の相互作用から実現できることが示された。

また、本稿の結果からでは、創発的学習手法と他の最適化手法との性能の違いは明らかではない。今後は、他手法との比較や、巨大なネットワークの最適化に向けた手法の改良などに取り組む予定である。また、創発的学習手法による知能の創発に向けて、研究を進める予定である。

参考文献

[中山 02] 中山功一他: 動的離隔型 GA(DS-GA)の提案, 情報処理学会論文誌, Vol.43, No. SIG 10 (TOM7), pp95-109, 2002.