

## 3H-6 遺伝的アルゴリズムを用いた リカレントニューラルネットの学習について

荻原拓也 中西正和

慶応義塾大学理工学部

### 1. はじめに

ニューラルネットには様々な学習方法があるが、リカレントニューラルネットについてはあまり良い学習方法が見つかっていない。そこで、リカレントニューラルネットの新しい学習法獲得に向けて単純なゲームを用いて実験を行ない、その結果について考察する。

### 2. リカレントニューラルネットとは

#### 2.1 ニューラルネット

一般のニューラルネットにも様々な種類がある。例えば単純パーセプトロンのようなネットでは、排他的論理和などは学習できないことがある<sup>1</sup>。また、このようなフィードフォワードの構造をしているニューラルネットは、関数型の入力および出力に対して有効であるが、時系列的な問題に対して効力を発揮しない。

#### 2.2 リカレントニューラルネット

リカレントニューラルネットは、2.1の例とは違いサイクリックなネットが存在する。そのためにこのネットは時間的な広がりを持つ。そこで、前回の入力および出力が学習の中に採り入れられることになる。

最近リカレントニューラルネットの学習にGAを利用した研究もでてきている[1]。しかし、まだかなり狭い範囲での実験でしかなく、GAの適用もうまくいったとはいえない。そこで、時系列問題で新たに実験を試みる。

### 3. 問題へのアプローチ

#### 3.1 GAを利用する意義

リカレントニューラルネットの既存の学習方法の中にリカレントバックプロパゲーションがあるが、この方法には問題点がある。その問題点を克服するためにGAをうまく適用することによって学習効率を上げる方法について考える。

#### 3.2 ネットワークにおける遺伝的アルゴリズムの応用

##### 3.2.1 結合行列

$$\vec{O} = \left( \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} * W \right) \cdot \vec{X}$$

但し、 $(\{a_{ij}\} * \{b_{ij}\} \stackrel{def}{=} \{a_{ij} \times b_{ij}\})$

図1: 結合行列

ニューロン同士のつながりを1として、つながりがない場合を0とする。図1は、

- $\vec{X}$   
入力層に入ってきたベクトル値
- $W$   
重み行列 (3.2.2)
- $\vec{O}$   
もどる層への入力の状態 (図4)

を表している。また、この行列を結合行列と呼び  $B_m$  と表すことにする。

##### 3.2.2 重み行列

さらに、重みを行列式にして表す (図2)。

結合行列  $B_m$  も考慮するのでこの場合、簡単のため

Training Recurrent Neural Network Using Genetic Algorithms

Takuya OGIHARA  
Masakazu NAKANISI

Faculty of science and technology, Keio University, 3-14-1  
Hiyoshi, Kanagawa Pref., 223, Japan

<sup>1</sup>backpropagationなどが克服するための学習方法として提案されている

$$W = \begin{bmatrix} 0.5 & 0.2 & -0.8 \\ 0.1 & 0.7 & 0.9 \\ -0.4 & 0.8 & -0.2 \\ -0.5 & -0.7 & 0.9 \\ 0.9 & 0.1 & 0.2 \end{bmatrix}$$

図 2: 重み行列  
 $B_m * W$ をBW行列(図3)と呼ぶことにする。

$$B_m * W = \begin{bmatrix} 0 & 0.2 & 0 \\ 0.1 & 0.7 & 0 \\ -0.4 & 0 & -0.2 \\ -0.5 & -0.7 & 0.9 \\ 0 & 0 & 0.2 \end{bmatrix}$$

図 3: BW 行列

### 3.3 計算機上での実現

本研究において用いるネットワークは図4に示すもので、これはモデル規範型制御 (*model based control*) と呼ばれる制御理論の概念に基づいている [3]。入力層と出力層で1組ずつの  $B_m$  及び  $W$ 行列ができています。

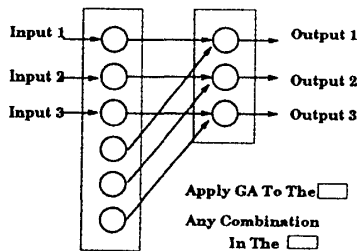


図 4: ネットワークの構造

### 3.4 題材について

単純でわかりやすいものということで、時系列ジャンケンを題材にしてみました。対戦成績を上げていくという方向で進める。対戦相手は、記憶を二段階まで持ち、一定の確率に基づいて次に出す手を決めていく。

### 3.5 GAの実現

- 評価関数

評価値 = 100 × 勝率

- 初期個体行列の生成  
乱数を使用して発生させる。
- 人口  
個体数の変化はなしとする。
- 選択  
親として選択される確率を  $P_s$  とすると、

$$P_s = \frac{\text{遺伝子の評価値}}{\text{世代の評価値の総和}}$$

- 交叉  
 $B_m$ 行列を組み合わせそれぞれ次世代の子どもとして作る。
- 突然変異  
 $B_m$ 行列の0と1を一定の確率で反転させる。

以上のような条件で実装した。

## 4. 結論

本研究による結果は良い結果が出たとはいえませんが、一つの方向性を示している。ネットワークに対するGAの適用法により学習速度がかなり異なることが分かった。本研究で得られた結果をもとにうまくGAを適用すれば、GAの欠点をみとせずその利点を得る方法を獲得することができる。

## 参考文献

[1] 佐藤裕二  
 Technical Paper SIG-PPAI-9401 人工知能学会  
 並列人工知能研究会  
 リカレントニューラルネットワークにおける先  
 読みと共進化, 1994.

[2] D. E. Goldberg  
*Genetic Algorithms in search, optimization and  
 machinelearning*  
 Addison Wesley, 1989.

[3] Nguyen, D., and Widrow, B  
*The truck backer-upper: An example of self-  
 learning in neural networks*  
 Proc. of the Int. Joint Conf. on Neural Net-  
 works,II, 357-363, IEEE Press, New York, June  
 1989.