3W-8

ニューロコンピュータAN1における並列処理

三井靖博　相原玲二　山下雅史　阿江忠

広島大学

1. はじめに

Hopfield ニューラルネットワーク（以下HNNと略す）は最適化問題の高速解法を与える1つの手段として非常に興味深い[1]。筆者らはこの求解性をコンピュータ・アーキテクチャに積極的に導入したマシンとしてアルゴリズム駆動ニューロコンピュータAN1を提案し[2]、その試作を行なった[3]。

本稿では並列処理マシンとしてのAN1の特徴、および並列処理例について述べる。

2. AN1の概要

HNNをメモリと見なすと、書込みデータが（一般には）読み出し時には変化するメモリとなる。このメモリを、以後ニューラルメモリNMと呼ぶ。アルゴリズム駆動ニューロコンピュータではNMとアルゴリズムは相互作用を行なうが、これをアルゴリズムを実現するプロセッサがNMを制御していると見なし、このプロセッサをニューラル制御プロセッサNPと呼ぶ。

2.1. AN1の基本動作

AN1の基本的な動作は、図1で表されるようなNMとNPで作られるサイクルで表せる。HNNは、物理系のエネルギに相当する評価関数を最小化することで最適解を求める手法を与える。しかし、真の解ではない極小点に陥ることが多い。そのため、状態読出しの後、アルゴリズム駆動ニューロコンピュータAN1では、アルゴリズムの実行が行なわれる。そして、NMが極小点に陥っているようであれば、再度状態書込み・読出しを行なう。このサイクルを繰り返してAN1は与えられた問題の最適解を求めるように動作する。

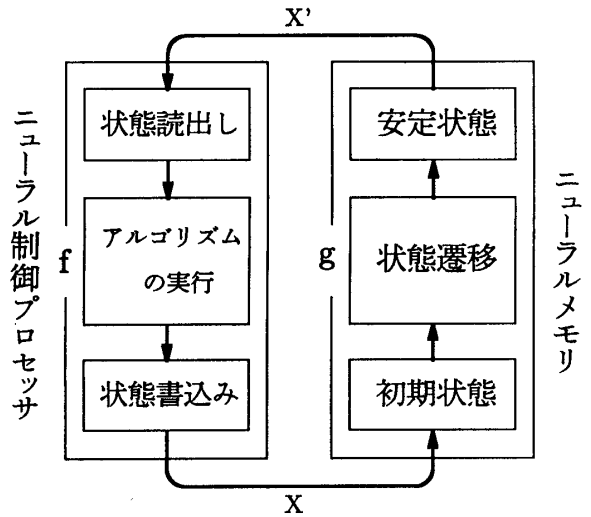

図1.NMとNPで作られるサイクル

2.2. AN1の基本構成

AN1は図2のように以下の部分で構成される。

（1）ニューラルメモリNM
（2）ニューラル制御プロセッサNP
（3）合成アレイSA

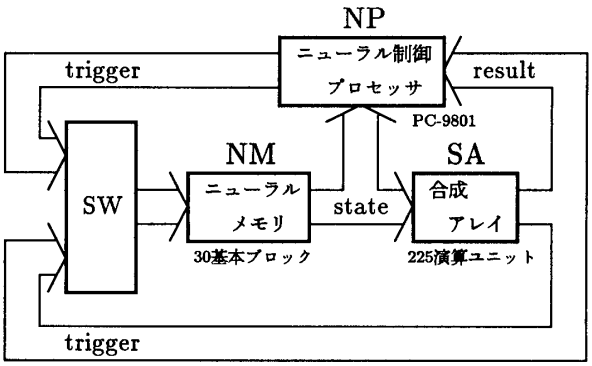

図2.AN1の基本構成

2.3. ニューラルメモリ

NMは基本ブロックの集合であり、1個の基本ブロックはRAMニューロン15個で構成される。

2.3.1. RAMニューロン

HNNをアナログ電子回路モデルにより実現すると、アンプ・抵抗等の素子の調節が複雑となり、集積化に適しているとはいえず、結合重みやしきい値を自由に変更することも困難である。そこでAN1の基本ブロックでは、しきい値素子をRAMによって実現している[3]。RAMを用いたバイナリHNNには、(a) アナログ素子を使用するのに比べて素子値の調節等がない, (b) RAMはオンラインで書き換え可能であり、真理値表を書き換えることによってニューロン間の結合、結合重み、しきい値が簡単に変えられる, などの特徴がある。その反面、(c) 大規模なニューラルネットワークが構成できない, という問題点もある。

2.3.2. 基本ブロック

基本ブロックはしきい値素子がRAMニューロンによって実現され、しきい値素子相互間に帰還のあるバイナリHNNである。RAMには一定の遅延が存在するため、RAMだけで構成した回路は発振する。この発振を押さえるために積分回路を挿入し遅れ補償を行なっている[3]。

Parallelism by Neuro-Computer AN1
Yasuhiro MITSUI, Reiji AIHARA,
Masahumi YAMASHITA, Tadashi AE
Hiroshima University

基本ブロックの操作は大きく分けて2つのモードに分かれる。一つは、基本ブロックをNMとして設定する、すなわち各RAMへの真理値表の書込み・読出しを行なう関数設定モードで、RAMへ書き込まれる真理値表はしきい値関数から作成される。そして真理値表を書き込むことによって、各RAMはRAMニューロンとして機能する。もう一つは、NMとしての動作する、すなわちコンピュテーションを行なう動作モードで、状態の書込み、状態遷移、状態読出し、の3つのフェーズに分かれる。基本ブロックへの状態の書込みは、各ニューロンの出力をトリガによって強制的に0、1、又は不定にすることにより行ない、それが初期状態となる。初期状態を設定後トリガを解除すると、NMは初期状態から安定状態へ遷移するので、状態遷移の後に状態読出しを行ない計算結果を調べる。

3. 並列処理例

組合せ問題の解法器を目的として設計されたAN1ではあるが、パターンマッチング処理を行なわせることもできる。ここでは並列処理の度合いを見るために、パターンマッチングに用いた例を述べる。

3.1. パターンマッチング処理

この処理では、ビット列で表現されるビットパターンの1ビットと1個のニューロンを1対1に対応させる。まず、設定パターン $x = (x_{n-1} \cdots x_1 x_0)$, 但し $x_j \in \{0,1\}$、とノイズによる誤りを考えて許容誤差（ハミング距離）を与える。そして入力パターンが x と同一か、指定されたハミング距離内のパターンであれば、入力パターンと x がマッチしたとして x を出力し、それ以外では x の補パターン $\overline{x} = (\overline{x}_{n-1} \cdots \overline{x}_1 \overline{x}_0)$, 但し $\overline{x}_j = x_j$を反転させた値、を出力するように基本ブロックを設定する。

3.2. 使用方法について

上述の動作を行なうように基本ブロックを設定するために、与えられた設定パターン x から各ニューロンの重み T_{ij} を次のように決める。

$$T_{ij} = \begin{cases} -1 & (\text{if } x_j = 0) \\ +1 & (\text{if } x_j = 1) \end{cases} \quad (1)$$

このとき、許容誤差をI、入力パターン $(O_{n-1} \cdots O_1 O_0)$ とし、T_{ij} と x_j とIから真理値表を作成する。RAM iの出力 O_i は次式で表される。

$$O_i = R(T_{i0}O_0 + T_{i1}O_1 + \cdots + T_{in-1}O_{n-1}) \quad (2)$$

$$R(s) = \begin{cases} \begin{cases} 1 & (s < X) \\ 0 & (s \geq X) \end{cases} & (\text{if } x_i = 0) \\ \begin{cases} 0 & (s < X) \\ 1 & (s \geq X) \end{cases} & (\text{if } x_i = 1) \end{cases}$$

$$X = \sum_{j=0}^{n-1} T_{ij}x_j - I$$

3.3. 実験方法

式(2)から作成した真理値表を基本ブロックの各RAMに書き込み、トリガによって入力パターンを与えて出力パターンを求める。実験は、マッチングを調べるパターンの構成ビット数を変えて基本ブロックを設定し、構成ビット数と出力パターンが求まる時間の関係を調べ、また許容誤差も変えて、許容誤差と出力パターンが求まる時間の関係も調べる。

3.4. 実験結果

図3に示されるように、マッチングを調べるパターンを構成するビット数を変えても出力パターンが一定時間で求まることが確かめられた。許容誤差を変えても時間に変化はなかった。より大きなHNNが構成可能になった場合、パターンを構成するビット数をさらに増加しても同様の結果が得られることが期待できる。パターンマッチングを1ビットずつ比較して行なう場合、パターンがnビットで構成されているとすると、入力パターンと設定パターンの比較は $O(n)$ で行なえる。このことから、AN1を用いてパターンマッチング処理を行なうことは $O(n)$ 倍の速度向上になる。

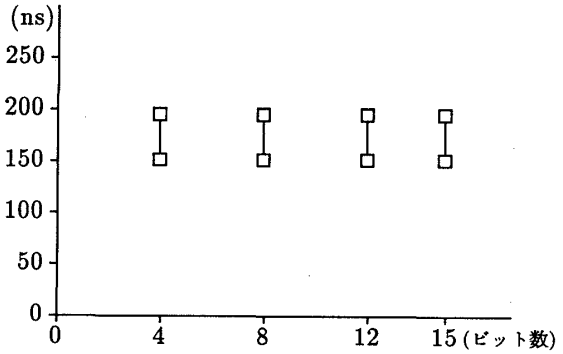

図3. パターンマッチングを行なうのに必要な時間

4. あとがき

本稿では、RAMニューロンで構成したHNNを基本ブロックとするAN1を使用して、パターンマッチング処理について実験し、AN1の有効性について調べた。その結果、基本ブロックのサイズが大きくできればかなり有効であることが確かめられた。今後の課題としては大規模なHNNの実現が挙げられる。そのためにはHNNの専用チップ化が考えられる。

参考文献

[1] J.J.Hopfield and D.W.Tank, " "Neural" Computation of Decisions in Optimization Problems," Biological Cybernetics, 52, pp.141-152 (1985).

[2] 阿江, 山下, 相原, 新田, "アルゴリズム駆動ニューロコンピュータ AN1," 信学技報, ICD88-129, pp.81-88 (Dec.1988).

[3] 阿江, 相原, 新田, 久長, "アルゴリズム駆動ニューロコンピュータ AN1 のハードウェア," 情報処理学会計算機アーキテクチャ研究会 (Jun.1989).