

3W-7

リング型マルチプロセッサ用
パイプラインBPアルゴリズム

緑川 博子 青柳 雄大 前畑 隆志 飯塚 肇
成蹊大学 工学部

1. はじめに

バックプロパゲーション (BP) は、様々な用途に用いられ、効果を上げている。現在BPを高速に処理する方法として様々な提案があるが、現実的には、浮動小数点高速処理ボード、あるいはマルチプロセッサによる並列処理が一般的である。本報告では、リング型マルチプロセッサ用のパイプラインBPアルゴリズムを提案し、トランスピュータを用いたリング型マルチプロセッサ上にインプリメントし、性能評価を行った。

2. パイプラインBPアルゴリズム

BPは多層ネットワークを下から上へ、上から下へというように処理が進むため、マルチプロセッサでデータを分割して並列処理する場合、各層をプロセッサに割り当てたのでは並列処理効果がでない。このため本手法では、図1のようにネットワークを縦に分割して、各プロセッサが各層の部分データを持ち、このデータをリング状に結ばれたプロセッサ間でパイプライン的に転送して他のプロセッサに伝えている。各プロセッサは受け取った部分データを用いて計算をし、同時にそのデータを一方のプロセッサに出力し、他方のプロセッサから次の処理のための部分データを受け取る。最終的にはすべての部分データがリングを1周する事により、各プロセッサが他のプロセッサすべてのデータを知って計算することができる。3層のネットワークの場合、アルゴリズムは主要な部分は、以下の6ステップで行われる。

- ①IN ホストプロセッサからリングプロセッサに入力信号I、教師信号Tをロードする。
- ②PH 入力層Iの部分データをリング転送しながら隠れ層Hを算出をする。
- ③HO 隠れ層Hの部分データをリング転送しながら出力層Oを算出をする。
- ④OH 各プロセッサで出力層補正值dOを算出し、バイアスBoを更新し、隠れ層Hの部分データをリング転送しながら重みWoを更新する。
- ⑤HI1 逆伝搬誤差の部分総和をリング転送しながら逆伝搬誤差の総和を算出し、隠れ層補正值dHを算出し、バイアスBhを更新する。
- ⑥HI2 入力層Iの部分データをリング転送しながら重みWhを更新する。

各ステップではリング転送の通信と計算を並列的に動かし、全体としてパイプライン的な処理になっている。

3. リング型マルチプロセッサによる性能評価実験

本実験で用いたシステムは図2のような構成をとる。用いたのはT800である。IBMPCはマンマシン用のホストであり、IBMPCに直結しているT800が処理上での、実際的なホストでBPの収束の判断、入力パターンと教師信号の転送を行っている。

4. 実測性能

T800を複数台用いたときの、1台での処理時間に対する実測相対性能を図3、図4の線(1)に示す。ただし横軸のプロセッサ台数は、実際に処理を分割して並列に処理しているリング要素の数で、ホスト用プロセッサは除外した。また1台の時の値はリング要素プロセッサ1台、ホストプロセッサ1台では、アルゴリズム的に意味がないので、すべての処理を1台のプロセッサで行った値をプロットしている。したがって1台のときの処理時間には通信時間が含まれていない。また処理時間は1パターンあたりの処理時間である。

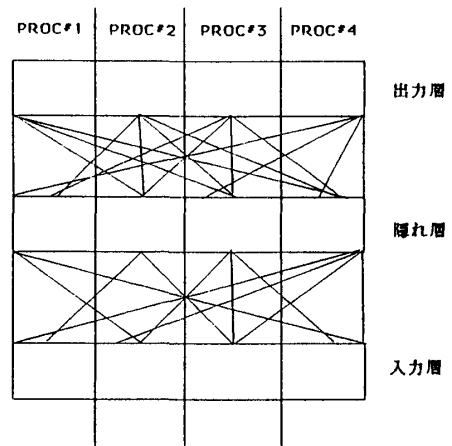


図1 データ分割方式

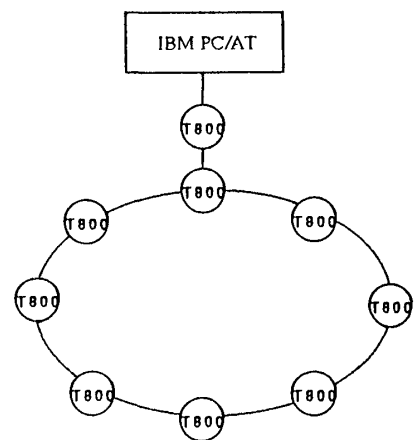


図2 システム構成

Pipeline BP algorithm for ring-shaped multiprocessor

Hiroko MIDORIKAWA Katsuhiko AOYAGI Takashi MAEHATA Hajime IIZUKA
SEIKEI University

用いたネットワークは、図3はNETTALKとほぼ同等なサイズのもの、図4は隠れ層、出力層が少なく、入力層が32x32のサイズのものである。これによると、ほぼ台数に比例して処理性能が向上することがわかる。

5. シミュレーションによる分析

本アルゴリズムは、BPのネットワーク構成とプロセッサ台数から各プロセッサの計算量、通信量を計算でき、用いるプロセッサの単位操作あたりの処理時間を用いて、処理時間の推定、処理時間の内部成分の分析ができる。アルゴリズムには逐次処理部Ts、並列処理部Tpがあり、各処理部に計算処理部と通信処理部がある。したがって、各部分処理あたりの処理時間Tは以下の式で決まる。

$$T = Ts + Tp = Tscal + Tscm + MAX(Tpca, Tpcm)$$

図3、図4にシミュレーションによる総処理時間(2)、並列処理部の総計算時間(3)、並列処理部の総通信時間(4)を示す。

逐次処理時間は台数に依らず一定で、ステップ①の通信時間とステップ⑤の計算時間が主で、その総和は図3、4の線(2)(3)の差分に対応する。

並列処理時間はステップ②③④⑥のそれぞれで通信時間と計算時間の大きい方を総和した時間で決まる。通信時間は、実験により通信開始時の微小固定時間と送るデータ量に比例する部分からなることがわかっており、

$$Tcom = Tset + Nd * Tunit$$

で表わされる。リング転送処理をN台のプロセッサを用いると1回のデータ転送で送られるデータ量はほぼ1/Nになるが、総通信時間はプロセッサ台数分を乗じるのでほぼ一定となる。ただし、TsetはN倍に増えるので、図中の線(4)のように微小量増加していく。

一方、計算量はほぼ1/N²になるので、図中の線(3)のように、プロセッサ台数Nを乗じても総計算時間はほぼ1/Nに比例して減少していく。

したがって並列処理時間は、ネットワーク構成が決まればその総データを通信するための時間が求まり、この下限値まで、ほぼ台数に比例して減少する。図4では、全体の計算量が少ないので、8台使用時に、ほぼ通信時間と計算時間が等しくなっている。

従って総処理時間は、ネットワーク構成によって定まるTsとTpcmの総和を下限値とする時間まで、プロセッサ台数に応じて、性能向上が見込める。

実測値(1)とシミュレート値(2)に差があるが、T800特有の以下のような要因で計算時間が10-20%変動することが実験で確かめられている。

- ①入力パターンに0を多く含む場合とそうでない場合
- ②定義された配列の形式
- ③式の評価のしかた(括弧のつけかた)

6. 他のマシンとの性能比較

図3のネットワークを処理した時の、他のマシンとの処理時間の比較を表1に示す。T800 1台でVAX8200のほぼ7倍、T800 8台では45倍近い性能が観測された。

7. おわりに

今回の実験、性能分析により、パイプラインBPアルゴリズムは、ほぼネットワークデータ量の通信時間と固定逐次処理時間で決まる下限値までほぼ台数に応じた性能が得られることがわかった。ネットワークがn倍大きくなれば、通信量はO(n)、計算量はO(n²)で増えるため、まずTpcmが問題になることはなく、ますます逐次処理BPに対して有利なる。また用意できるプロセッサ台数、扱うネットワークの違いによるプログラムの変更はほとんどいらず、拡張性に富んだアルゴリズムであるといえる。

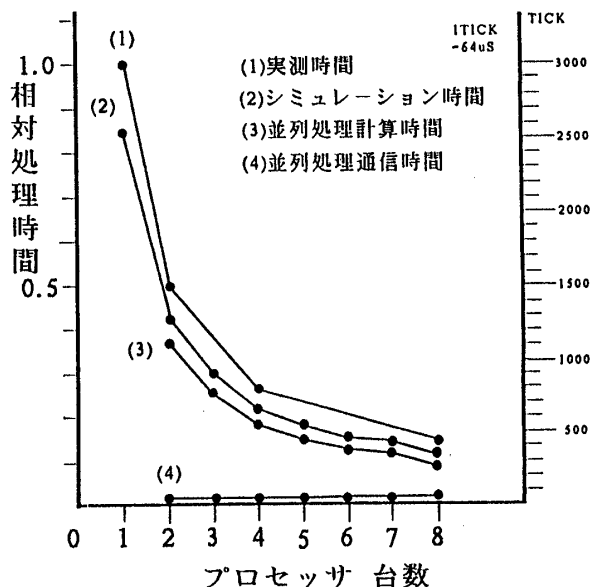


図3 プロセッサ台数と処理時間 (ネットワーク 64-64-208)

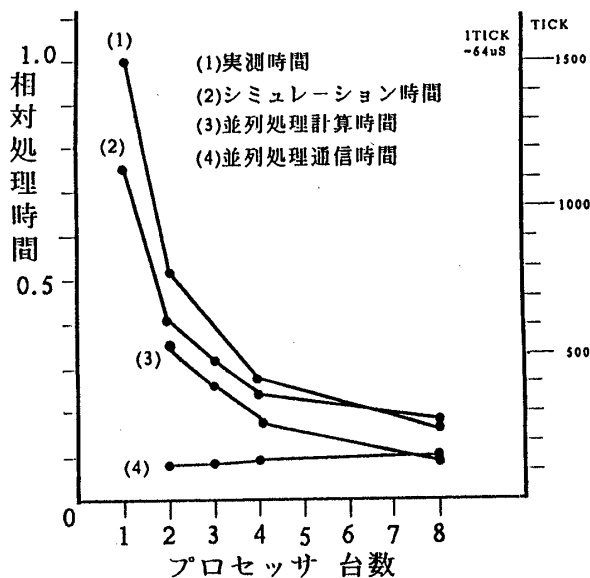


図4 プロセッサ台数と処理時間 (ネットワーク 8-8-1024)

マシン名	処理時間 (ms)	時間比
VAX8200	1359	47.4
SUN3/140	1111	38.7
T800 1個	202	7.0
T800 8個	29	1

表1 他のマシンとの学習時間の比較 (ネットワーク64-64-208、1パターンあたり)