

並列計算機“砂丘”による

4B-7

ニューラルネットワークシミュレーションについて

水船 博一 井上 倫夫 小林 康浩

鳥取大学工学部

1. はじめに

我々が現在開発をすすめている資源共有型並列計算機“砂丘”の並列処理の効果の検証を、ニューラルネットワークシミュレーションをとおして行った。本報告では、“砂丘”のメインメモリユニット(MU)へニューラルネットワークを実装し、実際に並列稼働させたときの効果について報告する。

2. 並列計算機“砂丘”の構成

“砂丘”は、最大64台のマイクロプロセッサを共有バスにより結合した、資源共有型の並列計算機である。図1は砂丘の構成図である。 [1] [2]

リードパスはそのブロック内のMUに対してのみつながるものが2本あり、ライトパスはオメガネットワーク(OMN)を通して全ブロックのMUに接続されている。このOMNの切り替えにより、全てのPUから、全ブロックのMUへの一斉放送・個別・パイプライン接続が可能である。

PUへの連絡は割り込みの形式で行う。その種類として、全てのPUに対する一斉割り込み・グループ単位のマスク割り込み・特定のPUに対する個別割り込みがある。

3. ニューラルネットワーク

今回採用したのは、バックプロパゲーションにより学習を行う階層型の、大規模なニューラルネットワークである。入力カメラから取り込んだ連続画像(画素数 p , 画面数 l)を用い、出力はその変化の種類を学習させる。図2にネットワークの構造を示す。

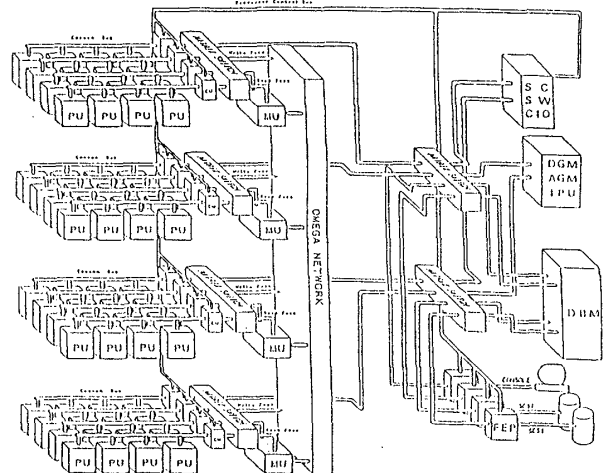


図1. “砂丘”の構成図

各層のニューロンの値を I_i^l, H_j, O_k , 重みを w_{ji}^l, v_{kj} , 閾値を θ_j, γ_k , 教師信号を T_k , 出力層の誤差を δ_k , 重み, 閾値の修正量を $\Delta v_{kj}, \Delta \gamma_k$, 中間層の誤差を λ_j , 重み, 閾値の修正量を $\Delta w_{ji}^l, \Delta \theta_j$ とする。ニューロンの演算は次式の通りである。

$$\begin{aligned}
 f(u) &= \frac{1}{\{1 + \exp^{-u}\}} & \delta_k &= O_k - T_k \\
 U_j^H &= \sum_{i=0}^{p-l} I_i^l w_{ji}^l + \theta_j & \Delta v_{kj} &= -\alpha \delta_k O_k (1 - O_k) H_j \\
 H_j &= f(U_j^H) & \Delta \gamma_k &= -\beta \delta_k O_k (1 - O_k) \\
 U_k^O &= \sum_{j=0}^{N_H-1} H_j v_{kj} + \gamma_k & \lambda_j &= \sum_{k=0}^{N_O-1} v_{kj} O_k (1 - O_k) \\
 O_k &= f(U_k^O) & \Delta w_{ji}^l &= -\alpha \lambda_j H_j (1 - H_j) I_i^l \\
 & & \Delta \theta_j &= -\beta \lambda_j H_j (1 - H_j)
 \end{aligned}$$

(a) 出力値の計算

(b) 修正量の計算

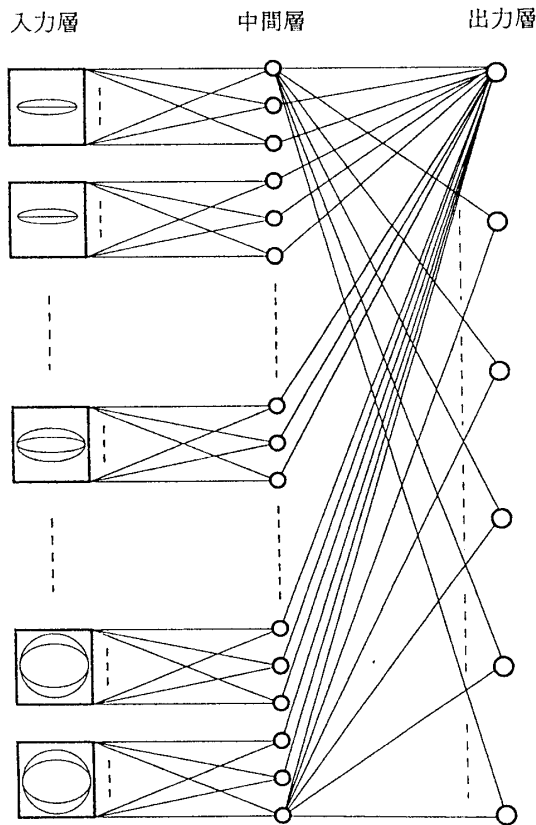


図2. ネットワークの構造

4. シミュレーション結果

シミュレーションに用いたニューラルネットワークは、1画素を1ニューロンに対応させて、入力ニューロン数(p)64×64、画面数(I)16、中間層48(3×16)、出力層4とした。中間層の1ニューロンの演算時間の実測値を表1に示す。LM(ローカルメモリ)、MU、DBM(データバッファメモリ)はニューラルネットワークのパラメータを格納するメモリである。各欄の数値は、アクセス競合によるバスの衝突が発生していない状態での演算時間である。

表1. 1ニューロンの演算時間の実測値

	LM	MU	DBM
演算時間(ms)	49.28	52.65	62.34

MUにネットワークのデータ(ニューロンの出力値・重みなど)を格納し、全PUがそれぞれの層において各ニューロンの処理を一斉に行う。中間層のニューロンの演算は対応する入力画像のみに対して積和計算を行う。

画像の判定時は、それぞれの層のニューロンの演算ごとに並列稼働を行う。

学習時は、判定時と同様のニューロンの出力値演算、および誤差逆伝播法によるニューラルネットワークの修正を行う。修正の段階では、出力誤差の計算、出力層の修正、中間層の修正ごとに並列稼働を行う。

実際の並列稼働の方法について述べる。FEPをマスタプロセッサとし、割り込みによりPUに一斉に並列稼働を行う^[3]。PU側は、Fetch&Addにより得られた値(プロセス番号)に従い領域分割した対象の処理を行う。ニューラルネットワークの場合はニューロンの出力値演算・パラメータの修正量を領域分割する。

5. 参考文献

- [1]武田他:資源共有型並列計算機“砂丘”のアクセス競合緩和法(そのアーキテクチャについて),情報処理学会第45回全国大会講演論文集vol.6, pp89-90, 1992
- [2]金崎他:資源共有型並列計算機“砂丘”のアクセス競合緩和法(その効果に関する検証),情報処理学会第45回全国大会講演論文集vol.6, pp91-92, 1992
- [3]河野他:資源共有型並列計算機“砂丘”の応用について,情報処理学会第47回全国大会講演論文集vol.6, pp45-46, 1993