4T-4

ＯＳＣＡＲ上での階層型ニューラル・ネットワーク・シミュレーションの並列処理手法

中野恵一、奥田恒久、笠原博徳
早稲田大学 理工学部 電気工学科

1．はじめに

本稿では、バック・プロパゲーション学習法を用いた階層型ニューラル・ネットワーク・シミュレーションを、汎用目的マルチプロセッサ・システムＯＳＣＡＲ[1][2]上で効率良く並列処理する手法を提案する。本手法ではスタティック・マルチプロセッサ・スケジューリング・アルゴリズム[3]を用いることにより、最少数のプロセッサで最小の処理時間を得ることを可能とする。また本手法の有効性は、ＯＳＣＡＲのアーキテクチャを前提とした厳密なシミュレーションにより検証される。

2．階層型ニューラル・ネットワーク・シミュレーション

バック・プロパゲーション学習法[4][5]は、Rumelhart等により提案されたニューラル・ネットワークの学習アルゴリズムである。これにより階層型ネットワークの中間層の学習が可能となり、パターン情報処理を中心とした様々な問題に応用され、有効性が確認されている[6]。

具体的なアルゴリズムは次の通りである。M層からなる、階層的なニューラル・ネットワークを考え、第１層を入力層、第M層を出力層、その他の層を中間層と呼ぶことにする。第ｋ層のユニットの個数をｎ[k]個とし、この層の第ｉ番目のユニットをｕ[k, i]とする。第１層のユニットは入力をそのまま出力するものとし、その他の層のユニットは多入力１出力の素子で、

$$o[k,i] = f\,(i[k,i]) \tag{1}$$

$$i[k+1,j] = \sum_i w[k,i,j]\,o[k,i] - \theta[k+1,j] \tag{2}$$

という準線形の入出力関係を持つものとする。ここで、ｋ層のユニットｕ[k, i]の入力の総和をｉ[k, i]、出力をｏ[k, i]、ユニットの特性関数をｆとする。さらにｗ[k, i, j]は第ｋ層の第ｉユニットから次の第ｋ＋１層の第ｊユニットへの結合の重み、θ[k+1, j]はしきい値である。このしきい値は、各層に常に－１を取る特殊なユニットｕ[k, 0]をおくことで、このユニットからの結合の重みｗ[k, 0, j]として考えることができる。

適当な入力パターンｘ[p]と、これに対応する、望ましい出力パターンｙの第ｊ成分をｙ[j]とする。このとき、次の式にしたがって各結合の重みを修正するのが、バック・プロパゲーション学習法である。

$$\delta[M,j] = (y[j] - o[M,j])\,f'\,(i[m,j]) \tag{3}$$

$k=M,\cdots,2$

$$\Delta w[k-1,i,j] = \alpha\,\Delta w[k-1,i,j] + \varepsilon\,\delta[k,j]\,o[k-1,i] \tag{4}$$

$$w[k-1,i,j] = w[k-1,i,j] + \Delta w[k-1,i,j] \tag{5}$$

$$\delta[k-1,i] = f'\,(i[k-1,i]) \sum_j w[k-1,i,j]\,\delta[k,j] \tag{6}$$

ここでαは学習速度定数、εは学習定数と呼ばれる、小さな正の定数である。

3．並列処理手法

本手法はタスク（プロセッサへの基本割当単位）生成部、タスク・スケジューリング部、およびマシンコード生成部から構成される。ホストコンピュータ上で生成されたコードは、各プロセッサ上のローカル・プログラムメモリにダウンロードされ、並列に実行される。

3．1　タスク生成

タスク生成においては、タスク・グラニュラリティの決定が重要なポイントとなる。バック・プロパゲーション並列処理ではタスク・サイズの小さい方から、１回の算術演算（加減乗除等）、(1)～(6)式のような演算式の計算、ユニットの計算、各層レベルの計算、を単位とする分割が考えられる。本手法はどのタスク・サイズについても同様に適用できるが、本稿では例として演算式レベルのタスク・サイズを使用する場合について述べる。

図１に示す排他的論理和の学習過程の計算をこのグラニュラリティで分割し、タスク集合を生成する。このタスク間には、タスク間のデータ依存による実行順序関係の制約（先行制約）が生じる。これらの先行制約は図２に示すタスク・グラフと呼ぶ無サイクル有向グラフで表現できる。この際各タスクの１プロセッサ上での処理時間の推定値も算出する。

3．2　タスク・スケジューリング

タスク・グラフ表示されたタスク集合のプロセッサへの最適割当ておよび実行順序決定問題は、実行終了時間最小マルチプロセッサ・スケジューリング問題と定義することができる。しかしこの問題は20年近くにわたり活発な研究が続けられてきたにもかかわらず、効率よいアルゴリズムは開発できず、最近では実用的なアルゴリズムの開発と実マルチプロセッサ上への適用はほとんどあきらめられていた。これに対して筆者等は、ヒューリスティック・アルゴリズムCP/MISF（Critical Path/Most Immediate Successors First）法、それをデータ転送を考慮できるように拡張したCP/MISF/DT、CP/DT/MISF法と、実用的な最適スケジューリング・アルゴリズムDF/IHS（Depth First/Implicit Heuristic Search）法をすでに開発している[3]。さらに、これらのアルゴリズムを用いて種々のアプリケーションの並列処理を実マルチプロセッサ上で実現できることを初めて示した[2][7]。したがって本稿で述べるバック・プロパゲーション学習法計算でもこれらのアルゴリズムを適用して並列処理することが可能である。

3．3　マシンコード生成

次にタスク・スケジューリングの結果に基づき、使用するマルチプロセッサ・システム中の各プロセッサ上で実行されるマシンコードを生成する。各プロセッサ用のマシンコードは、そのプロセッサに割り当てられたタスク用のマシンコードを実行順序に従って並べたものに、タスク間同期を取るマシンコードを挿入した形となっている。本手法ではこのマシンコード生成の際に、同期オーバーヘッドの最小化、各プロセッサ内レジスタ利用の最適化を行う。

A Parallel Processing Scheme for the Layered-Neural-Network Simulation on OSCAR

Keiichi NAKANO, Tsunehisa OKUDA, Hironori KASAHARA

WASEDA University

4．手法の有効性評価

提案した並列処理手法の有効性をＯＳＣＡＲの１プロセッサ・クラスタ[1][2]を前提としたシミュレーションにより評価した結果について述べる。

ＯＳＣＡＲの１クラスタは、16台のプロセッサ・エレメント（ＰＥ）を３本のバスで結合した構造となっており、各ＰＥは64個の汎用レジスタを持つ32bitＲＩＳＣライク・プロセッサ、２kwのＰＥ間データ転送用２ポートメモリ、256kwローカル・データメモリ、128kw×２バンクのインストラクションメモリ等から構成される。そしてメモリアクセス命令、加・減・乗算等の浮動小数点演算等すべての命令を１クロック（200［ns］）で処理する。３本あるバスのデータ転送速度は最大60Mbyte/sである。また各ＰＥ間のデータ転送モードとして、センダ・プロセッサがレシーバ・プロセッサの２ポートメモリ上にデータを書き込む１対１転送、ブロードキャスト、共有メモリを介した１対複数転送（３本のバスから同時に各共有メモリへアクセスできる）を用意している。スケジューリング結果を基に、これら３モードデータ転送を最適に使い分け、オーバーヘッドの低い並列処理を行う。

このシステムを前提とし、タスク間のデータ転送時間、同期に要する時間、バス・アクセス競合による待ち時間を含んだ非常に厳密なシミュレーションを行なった。例えば５層のネットワークによる３項関係の学習計算の場合961個のタスクが生成され、その処理時間は図３のようになる。すなわち１台で約6.5［ms］かかる計算が、プロセッサ16台で約0.79［ms］に短縮できるというように、本手法で効率よい並列処理が実現できることが確かめられた。

5．おわりに

本稿ではＯＳＣＡＲ上でのバック・プロパゲーション学習法を用いた階層型ニューラル・ネットワーク・シミュレーションの並列処理手法を提案した。本手法を用いることにより、同一のマルチプロセッサ上で任意の規模のニューラル・ネットワーク・シミュレーションを自動的に並列処理することができる。今後は階層型だけではなく、任意のネットワーク形状のシミュレーションも行なうことができるように手法を拡張するとともに、ＯＳＣＡＲ上でインプリメントしていく予定である。

参考文献

[1] 笠原，成田，橋本，”ＯＳＣＡＲ（Optimally Scheduled Advanced Multiprocessor）のアーキテクチャ”，信学論(D)，Vol.J71-D，No.8，pp.1440-1445

[2] H.Kasahara，T.Fujii，H.Nakayama，S.Narita and L.Chua ："A Parallel Processing Scheme for the Solution of Sparse Linear Equations Using Static Optimal Multiprocessor Scheduling Algorithms"，Proc. 2nd Inter-national Conf. on Supercomputing，vol.2，433-442(May 1987)

[3] H.Kasahara and S.Narita ："Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing"，IEEE Trans. Comput.，c-33，1023-1029(Nov. 1984)

[4] D.E.Rumelhart，G.E.Hinton and R.J.Williams ："Learning Representations by Back-Propagating Errors"，Nature，vol.323，533-536(Oct. 1986)

[5] D.E.Rumelhart，J.L.McClelland and the PDP Research Group ："Parallel Distributed Processing"，vol.1&2，MIT Press，(1986)

[6] 稲葉，”ニューラル・ネットをパターン認識、信号処理、知識処理に使う”、日経エレクトロニクス、no.427，115-124，(1987)

[7] 笠原，藤井，本多，成田，”スタティック・マルチプロセッサ・スケジューリング・アルゴリズムを用いた常微分方程式求解の並列処理”，情処論文誌，28-10，(Oct. 1987)

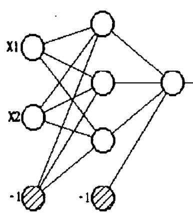

図１ 排他的論理和の学習に用いた３層ネットワークの例

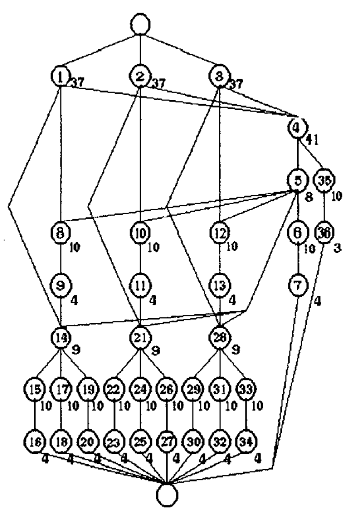

図２　タスク・グラフ

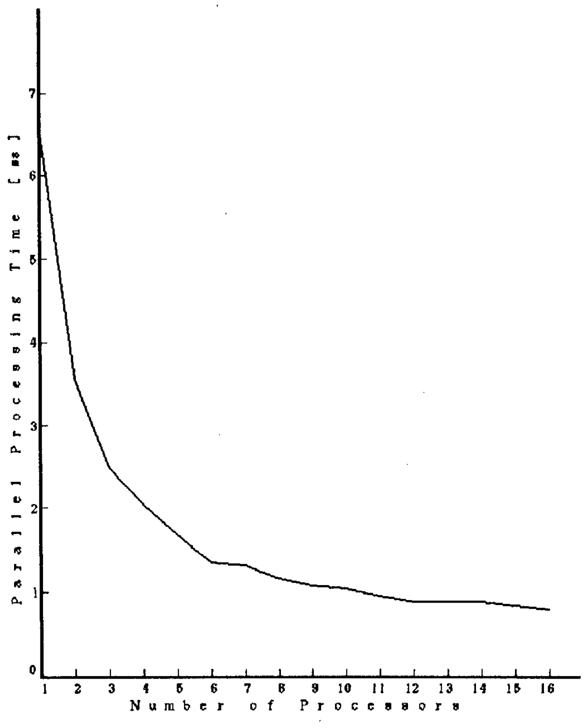

図３：３項関係（文献[4]の一部）学習の並列処理時間　（ループ１回）