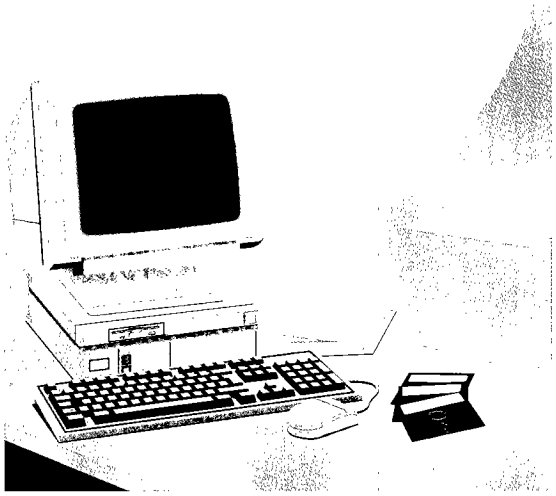
解説

超並列計算機を用いた人工ニューラルネットワーク計算とその応用



筑波大学 電子・情報工学系 安永 守利
yasunaga@is.tsukuba.ac.jp

超並列計算機の位置付け

チェスの世界チャンピオンとなった並列計算機Deep Blueの話題は，まだ記憶に新しい．今後，並列・超並列計算機は，計算物理学や気象予測，構造物設計といったいわゆる“大規模数値計算”だけでなく知識処理の分野でもその威力を発揮すると考えられる．ご多分に漏れず，人工ニューラルネットワークの分野でも，並列計算機の利用技術が研究されている．

人工ニューラルネットワーク研究の大きな目的の1つは，生体の神経回路網をモデル化し，これを工学的応用（パターン認識や最適化問題解法など）に結び付けることである．生体の神経回路網は非常に多くの神経細胞から構成されたネットワークであり，したがってそのモデルも多くの神経細胞から構成される．人工ニューラルネットワークの計算（学習アルゴリズム）は，この多数の神経細胞が持つ多くのパラメータを少しずつ調節する．このため，実用的な応用を対象とした大規模人工ニューラルネットワークの処理には，多大な計算量が伴う．この高速計算のための専用ハードウェアがニューロコンピュータ（ニューロハードウェア）である．一方で，本稿で解説する超並列計算機による高速化のアプローチがある．両者の位置付けは以下であろう．ニューロコンピュータは，超並列計算機をもってしても高速化が困難な大規模問題を高速処理することを目的とする．あるいは，超並列計算機と同等な処理速度をはるかにコンパクトなハードウェアで提供することを目的とする．一方，超並列計算機はその充実した利用環境，プログラミング環境のもとで（ニューロコンピュータより手軽に）高速ニューラルネットワーク計算を提供することを目的とする．

ニューロコンピュータより手軽に利用できるのであれば，超並列計算機を使いたくなる．しかし，現実は，人工ニューラルネットワークをうまく超並列計算機上にマッピング（実装）しないと高い性能が得られない．人工ニューラルネットワークにおける超並列計算機の利用技術の研究は，このマッピング技法の研究といっても過言ではない．

本稿では，2つの人工ニューラルネットワークを例にその超並列計算機上への実装方法と応用例を紹介する．1つは階層型のフィードフォワード型ネットワークであり，もう1つは2次元配列型のネットワークである．この2つを例としてあげた理由は，両者はそれぞれ，逆誤差伝播学習（Back Propagation，以下BPと呼ぶ）と自己組織化特徴地図（SOM: Self-Organizing feature Map，以下SOMと呼ぶ）学習のためのネットワークであり，多くの応用事例が報告されているからである．

なお，本稿は一般読者を対象とした解説記事であり，さらに最近の研究事例を知りたい読者のために参考文献としてたとえば文献1)，2）をあげる．

Artificial Neural Networks

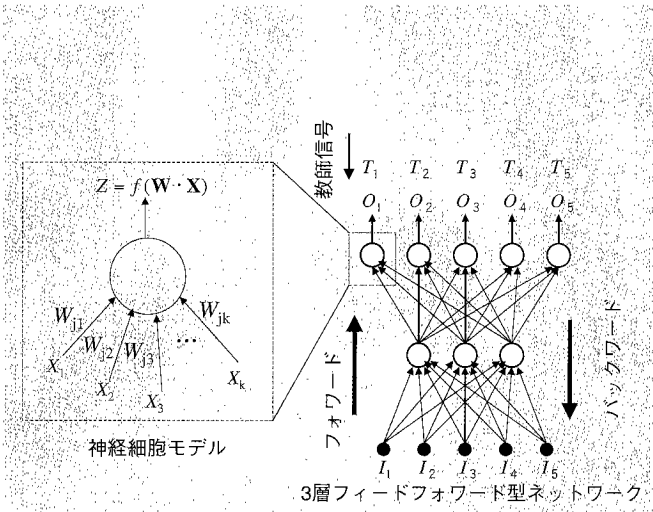


図-1　神経細胞モデルとフィードフォワード型ネットワーク

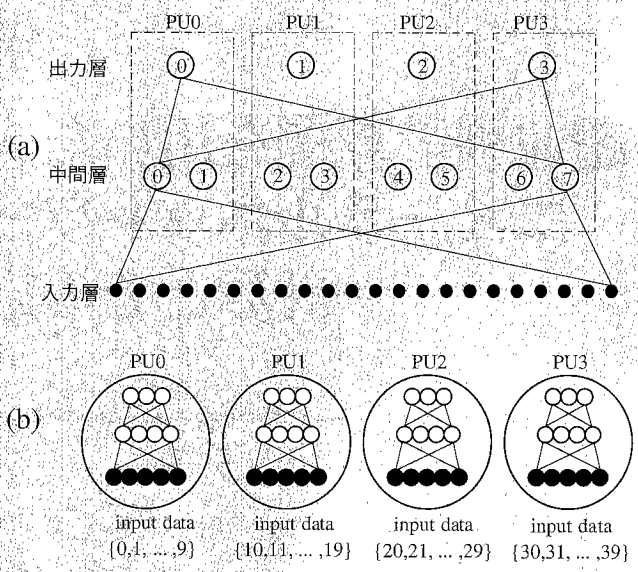


図-2　(a) ノードパラレル方式と (b) データパラレル方式

BPと階層フィードフォワード型ネットワーク

BPは代表的な教師有り学習であり，工学的応用に最も多く適用されている学習アルゴリズムの1つである．BPは，階層フィードフォワード型のネットワークを用いる．図-1に最も典型的な3層のフィードフォワード型ネットワークとニューロン（神経細胞モデル）を示す（入力層をデータとみなして，2層のフィードフォワード型と呼ぶ場合もある）．入力信号をX，ニューロンの重みをWとすると，各ニューロンはXとWの積和演算（入力ベクトルと重みベクトルの内積演算）を行った後，その非線型変換fを出力（Z）とする．ニューラルネットワークに入力データ（I）を入力し，その出力（O）を計算する（フォワード計算）．その出力が望ましい出力である教師信号（T）と異なっていれば，その差分（$O-T$）を出力側から入力側に伝播させ（バックワード計算），次回からその差分が縮まるようにパラメータ（重みと非線型関数のいくつかの定数）を繰り返し調整する．これがBPの基本的なアルゴリズムである．

これまでにBPを用いた多くの成功事例が報告されてきた．一方，厄介な点はその計算量にある．BPは，繰り返し計算により多くのパラメータの微調整をするため，非常に計算量の多いアルゴリズムである．次章で解説する「英単語の発音学習問題」は1980年代にBPを実用的な大規模問題に適用した典型的な例である．学習が終了するまでの経過を記録してデモしたところ，“赤ん坊が発音を学びながら大人に成長するように計算機が学習した”と一躍脚光を浴びた．一方，その学習計算の時間は膨大で，当時のスーパーミニコンを1週間以上回し続ける必要があった．BPに限らず，一般に大規模ニューラルネットワークの計算量は多大なものとなる．これが大きな要因となって，1980年代後半からニューロコンピュータや並列計算機を用いてニューラルネットワーク計算を高速化しようとする研究が活発化した．特に，BPの高速化を目指した研究・開発者の間では，1秒間に可能なパラメータの更新回数を表すCUPS（Connections Updated Per Second）という単位が性能指標に使われるようになった．

さて，本題としての並列計算機上へのBP実装（マッピング）であるが，基本的なアプローチは，データパラレル方式かノードパラレル方式のどちらかである．

＜ノードパラレル方式＞

ノードパラレル方式では，階層型ニューラルネットワークをPU（プロセッサユニット）数で分割する．分割の

方法としては，当然，各プロセッサユニット（PU）間の通信オーバーヘッドを減らす分割方法が良く，このため図-2（a）のように縦型に分割する．図では，PU0に中間層ニューロン0と1，出力層ニューロン0，PU1に中間層ニューロン2と3，出力層ニューロン1が割り当てられている．すべてのPUは，それぞれが完全な学習データセットを必要とする（すべてのPUが完全な学習データセットをローカルメモリにあらかじめロードしていることが望ましい）．各PUは，それぞれに割り当てられた中間層，および出力層ニューロンのフォワード計算，バックワード計算を行う．計算中にPUの担当外ニューロンの出力値等を他のPUとの間でお互いにやりとりする（データ通信する）．この方式では，フォワード計算，バックワード計算のどちらにおいても全対全ブロードキャスト通信が通信処理のほとんどを占める．このため，広域データ通信が主流となる．したがって，並列計算機の通信ネットワーク性能が低かったり，ブロードキャスト通信に適さないネットワークトポロジであると，通信オーバーヘッドが大きくなり，このため高い並列効果が得られないことがある．

ここで，（並列計算機とは直接関係なく）BP自体の計算方法としての逐次修正法と一括修正法を説明する．逐次修正法にせよ，一括修正法にせよ，学習データセットを何回も繰り返して学習することに変わりはないが，以下の点が異なる．逐次修正法では，各入力データごとにパラメータを修正（調整）する．一方，一括修正法では入力データごとにパラメータ修正を行わず，各入力データごとの修正量を記憶しておき，一通りの修正量の計算が終わった後に修正量の総和を1回だけ修正する．一括修正法は，各入力データごとの修正が不要なために高速学習が可能である．一方，入力データごとに“きめ細やかな修正”が行われないぶん，学習データセットの繰り返し数が増加する．一括と逐次のどちらが良いかは，対象問題，並列計算機のPU性能，およびネットワーク性能（通信性能）による．このため，一般にどちらが良いとは言い切れない．ノードパラレル方式の場合，逐次修正法と一括修正法の両方が可能である．

＜データパラレル方式＞

各プロセッサユニットPUにそれぞれ完全な3層のフィードフォワード型ネットワークを持たせ，学習データセット（入力データとその教師データのセット）をPU数で分割した学習データセットのサブセットを各PUに割り当てる．たとえば，図-2（b）では，40個の学習パターンデータを4つのPUにそれぞれ10個ずつ割り当てている．それぞれのPUは割り当てられた学習データのサブセットについて修正量を計算し，最後に修正量の総和を

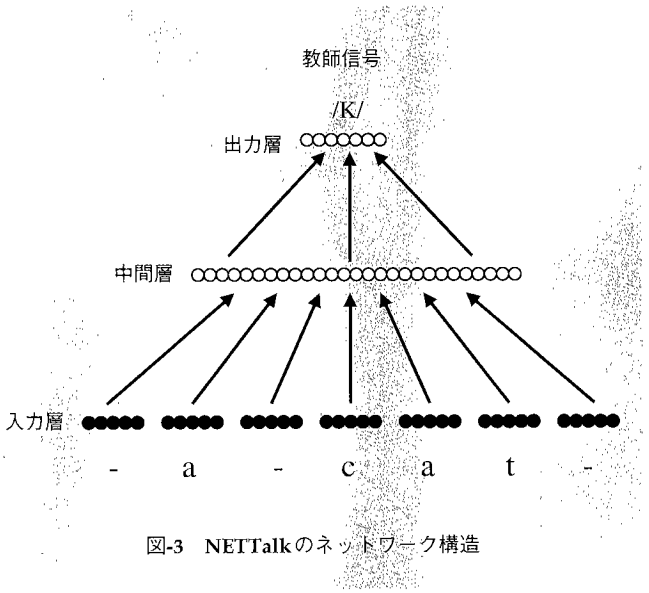


図-3　NETTalkのネットワーク構造

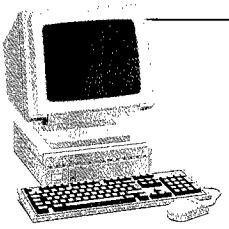
それぞれのPUごとに計算し，パラメータを一括修正する．このため，前述した逐次修正法は適用できない．PU間の通信は，一括修正前に各PUごとの修正量を全対全ブロードキャストする通信となる．このため，ノードパラレルと同様に，通信ネットワーク性能が低かったり，ブロードキャスト通信に適さないネットワークトポロジであると，通信オーバーヘッドが大きくなり，高い並列効果が得られないことがある．

＜ハイブリッド方式＞

データパラレル，ノードパラレルの両方法ともPU数の増加により並列化効率（スピードアップ比）は低下する．このため，ノードパラレル，あるいはデータパラレルによる単純な並列化では，並列計算機の多くのPU資源を効率的に使用できない．並列化効率の低下を抑えて多くのPUを利用するために，ノードパラレルとデータパラレルを組み合わせる方式がハイブリッド方式である．N台のPUを$N=n\times m$に分割し，n並列のデータパラレルを行うとともに，各データパラレル用のニューラルネットワークをm台のPUでノードパラレルに実行する．我々は次の応用事例で解説するように，ハイブリッド方式を用いてこれまでの報告値を凌ぐ高い学習性能を達成した．

「1秒で辞書を学習」

BPはさまざまな応用に適用されてきたが，その最も有名な適用例の1つは前述した英単語の発音記号学習であろう．これは，1987年にT.J.Sejnowskiらによって開発された[3]（彼らは，この適用例をNETTalkと名付けた）．発音記号が分かっている単語データセットを用いてBPを行い，未知の英単語の発音記号を推論する問題である．



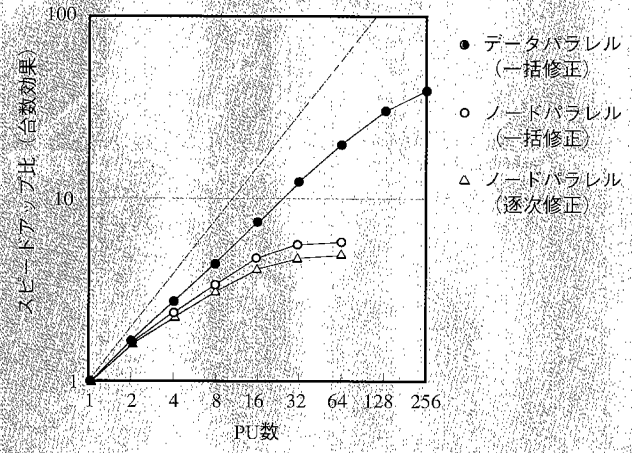


図-4　PU数に対するスピードアップ比

超並列計算機名	一括／逐次学習	PU数	MCUPS	単位時間当たりの学習文字数
CP-PACS [5]	一括	256	1056	57,600文字/秒
CP-PACS [5]	逐次	64	61.8	
AP-1000 [1]	一括	512	82	
AP-1000 [1]	逐次	64	3.0	
CM-2 [2]	一括	64K	40	
CM-2 [2]	逐次	16K	2.8	
VAX11/780FPA [3]	逐次	1	0.037（推定）	2文字/秒

表-1　代表的なBP学習性能実測値

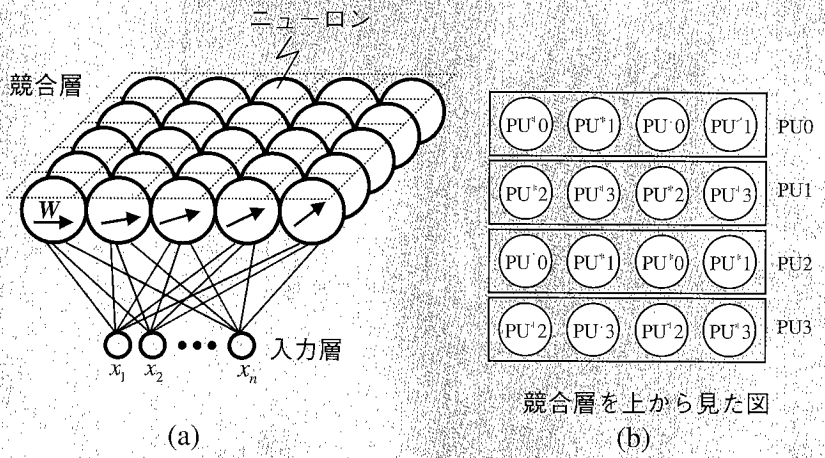


図-5　(a) SOMのネットワーク（2次元競合層）と (b) SOM競合層の分割とPU割当て例

英単語と発音記号の間の規則の抽出が難しく，したがって，従来のルールベースによるアプローチには難しい問題である．学習用のデータセットとして英単語（入力データ）とその発音記号（教師データ）を用意し，これを3層のフィードフォワードネットワークに入力しBPでネットワークを学習させた（図-3）．入力層，中間層，出力層のニューロン数は，それぞれ，203，80，26である．学習が進むにつれて，徐々に単語に対して正しい発音記号を出力する割合（正解率）が増加した．さらに，未知の単語（学習に使わなかった単語）に対しても，正しい発音記号を当てるようになった．

一方，このBPによる学習に当時のハイエンドマシン（浮動少数点アクセラレータ付きのVAX-11/780）で2文字／秒程度，10万語の学習には1～2週間を要した．

我々は並列化方式として前述したハイブリッド方式を用い，これを超並列計算機CP-PACS[4),5)]上に実装することでNETTalkの高速化を試みた[6)]．CP-PACSは，2,048台のPU（プロセッサユニット）から構成されるMIMD（Multiple Instruction stream Multiple Data stream）型の超並列計算機である．ノードパラレル方式とデータパラレル方式を用いたときのスピードアップ比（台数効果）の測定結果を図-4に示す．図中の鎖線は，PU台数に比例して台数効果が得られる場合，すなわち，理想的な場合を示す．これに対していずれの方式でも，PU台数が増加

するとともにそのスピードアップ比（台数効果）は徐々に飽和する．

ハイブリッド方式は，ノードパラレル方式とデータパラレル方式を組み合わせることにより，それぞれスピードアップ比の飽和を抑えてN台のプロセッサをより効果的に使用することが目的である．ここでハイブリッド方式の課題は，N台のプロセッサをnデータパラレル×mノードパラレルに最適分割する（$N=n\times m$）組合せの決定方法である．

我々は，この課題を並列化効率の近似式から予測する手法を提案し，実験結果とよく一致することを示した[6]．この予測から256台のPUを32データパラレル×8ノードパラレルに分割し，1,000MCUPSを超す高速BPを実現した．**表-1**に代表的なBPの処理速度の報告値を示す．NETTalkが提案された当時2文字／秒であった学習速度は，1,000MCUPSを超えることで57,600字／秒という当時では想像もつかなかった学習速度を達成できるに至った．「BPは素晴らしい学習アルゴリズムであるが，大規模問題に対しては時間がかかりすぎる」という“固定観念”は過去のものとなったといってよいであろう．

SOMと2次元配列型のネットワーク

SOM[7]は代表的な教師無し学習アルゴリズムである．その特長は，高次元データ（たとえば，ある人の健康状態が血液検査の結果からGOT, GPT, 総コレステロール，アルブミンの4つの量で表されているとすると，このデータの次元は4次元である）のデータ間の距離関係（似ている度合い）を2次元マップ上に射影することができる点である．つまり，高次元空間でのデータの位相関係を2次元マップ（地図）としてビジュアルに表示することができる．工学や医学への応用だけではなく，最近は，経済分析や人事適性配置といったビジネス戦略への応用も試みられている．

SOMでは一般に2次元配列型のネットワークが用いられる（**図-5 (a)**）．2次元配列されたニューロンの層は，競合層，あるいは出力層と呼ばれる．各ニューロンは入力データ（n次元）と同じ次元の重みWを持つ．SOMの学習は，次の通りである．はじめに，入力データxが各ニューロンに入力される．各ニューロンは，入力データと重みWとの距離計算を行い（たとえば，ユークリッド距離），その値を出力とする．次に，全ニューロンの中で出力が最小のニューロン（入力データに最も近い重みを持っているニューロン）を検出する．検出されたニューロンを勝者ニューロンと呼ぶ．最後に，勝者ニューロンの重みとその近傍のニューロンの重みを入力データに少し近づける．これが，1回の学習計算であり，この処理を全入力データに対して繰り返し行う．単純なこの処理を繰り返すことで，類似した入力データに対する勝者ニューロンは，2次元配列上でも近い位置関係になる（入力データの類似関係がニューロン地図上の位置関係に射影される）．最初，互いにバラバラな方向を向いていたすべてのベクトルWが，学習が進むにつれて図-5 (a) に示すように秩序のある状態に次第に変化する．この特長を利用することで，高次元データの類似関係を2次元平面状に可視化できる（詳しくは，後述の適用事例を参照されたい）．SOMも多数のニューロンの繰り返し計算が基本であり，したがって，ニューロン数が多くなるとBPと同様に学習終了までに（重みベクトル全体が秩序のある状態に収束するまでに）長い計算時間が必要となる．

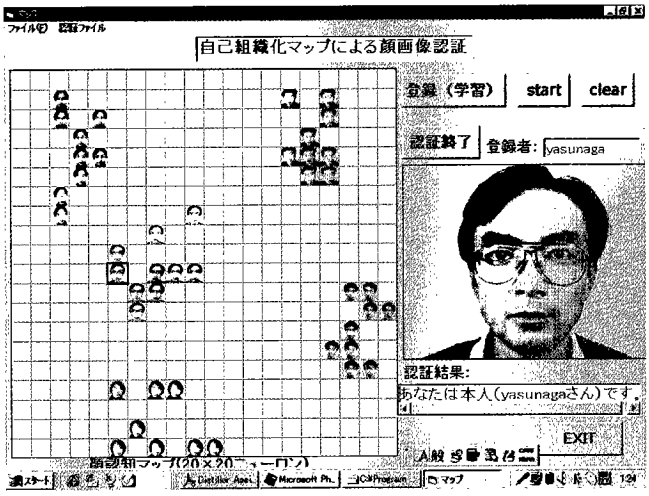
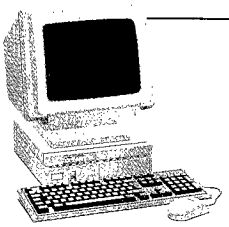


図-6　SOMを用いた顔認識システムのGUI

SOMも並列計算による高速化が可能である．並列（超並列）計算機上へのSOM実装の報告例はBPほど多くないもののいくつか報告されている．最も単純なPU割当ては，競合層を**図-5 (b)** のPU0～PU3で示すように単純分割する方法である．この場合，勝者ニューロンの検出には並列PUによる検出が効果的に働く．しかし，その後の重み修正処理が勝者ニューロン近傍のローカルな処理となるため，負荷の分散がうまくできない欠点がある．そこで，図-5 (b) のPU*0からPU*3に示すように各ニューロンのプロセッサ割当てを順番にずらす“スキュー割当て”によって重み修正処理時の負荷を分散させる手法も提案されている．

「顔画像特徴マップが10秒で完成」

我々は，SOMを用いた顔画像特徴マップを利用し，顔画像による認証システムのプロトタイプを試作した．**図-6**にそのGUI画面を示す．通常，認証システムは「登録」と「認証」の2つの手続きからなる．「登録」は認証対象者の顔画像をあらかじめシステムに登録する手続きであ



る.「認証」は,登録手続き後に認証対象者の顔画像を入力し,この画像が本人か否かを判定する手続きである.本システムでは,SOMを用いて次のように登録を行う.

あらかじめ登録者とは異なる他人の顔画像を複数枚撮影しておく.これらと登録対象者の複数枚の顔画像をまとめて,学習用の入力データセットとする.ここで,各画像は,たとえば8×8画素に平滑化しておく(平滑化することにより,顔画像の多少の平行移動に対しても正しい認証が行われるとともに,画像の次元を低減している).この平滑化後の画像(64次元のベクトル)が重みベクトルであり,前章で説明した高次元データに対応する.ベクトルの各要素の値は,画素の濃度値である.このデータセットを基にSOMの学習を行う.試作したプロトタイプでは,SOMの学習進行状態をGUI画面上で見ることができる.画面左側の碁盤状の配列は競合層を示しており,1つの四角形が1つのニューロンに対応している.図-6は20ニューロン×20ニューロン(合計400ニューロン)の例である.GUI上の表示では,ある学習間隔で,それぞれの入力データ(入力画像)に対する勝者ニューロンの上に対応する縮小した入力画像を表示する.これにより,自己組織化の進行状況(似た入力画像に対する勝者ニューロンが徐々にクラスタを形成する)をモニタすることができる.なお,ここで"クラスタ"とは,お互いに距離が近いニューロン同士の群れを意味する.図-6は学習終了の状態であり,登録者を含めた5人分の画像のクラスタが形成されていることが分かる.本システムではこのようにSOMを用いて各登録者ごとにクラスタ形成を行い,各登録者ごとに競合層ニューロンの重みをハードディスクに保存することが「登録」に相当する.

「認証」では,認証対象画像を学習終了後の競合層に入力し,この画像に対する勝者ニューロンを決定する(認証対象画像がGUI画面の右側に示され,かつ,この画像に対する勝者ニューロンの上に縮小した認証対象画像が枠で囲まれてGUI左側のマップ上に表示される).ここで,認証対象画像に対する勝者ニューロンとすべての学習画像に対する勝者ニューロンとのそれぞれの距離(20×20ニューロンのマップ上の距離)を計算する.この結果,認証対象画像の勝者ニューロンに対する最小距離の勝者ニューロンが,登録者の学習画像の勝者ニューロンであった場合,認証対象画像は登録者のものと判定される(本人として認証される).

以上紹介した例では20×20ニューロンの比較的小規模なSOMを用いているが,これをさらに大規模化することで,多くの高次元データの類似度をビジュアルに表現することができる.問題は学習にかかる計算量である.20×20ニューロン程度の規模であれば,高速なパソコンで数分のうちに自己組織化が終了する.しかし,"瞬時自己組織化"(瞬時とはあいまいな表現ではあるが,自己組織化を観測しているユーザが待ちを感じない程度の時間ということで10秒程度)は,現在のところシングルプロセッサでは不可能である.高速化には,前述した並列処理が効果的に働く.我々はCP-PACS(64プロセッサ)を用いて上述した顔画像のSOMによる学習の並列化を試みた.競合層の分割には,図-5(b)に示した並列化の中の単純分割を用いた.その結果,約10秒で学習が終了し,顔画像の特徴マップを作成することができた.さらに大規模並列化を行うことで,10,000×10,000ニューロンのマップによる顔の相関地図を作るといったことも可能であると考えられる(もちろん,どうやって10,000×10,000ニューロンの顔地図を表示するかといった問題もあるが).このようにSOMを並列処理することで,非常に多くの高次元データの類似度を高速にクラスタリングし,ビジュアルに表示することが可能になる.

キラーアプリケーションの登場を期待

大規模人工ニューラルネットワークの計算は時間がかかるという固定観念は,ニューロコンピュータや超並列計算機を利用することにより捨て去ることができる.人工ニューラルネットワーク計算の高速化という"量"の差が"質"の差につながり,80年代後半に登場して我々を驚嘆させたNETTalkやその他のアプリケーションを凌駕する21世紀のキラーアプリケーションが登場することを期待する.

参考文献
1) Sundararajan, N. and Saratchandran, P. (Eds.): Parallel Architectures for Artificial Neural Networks, IEEE Computer Society Press (1998).
2) Serbedzija, N.B.: Simulating Artificial Neural Networks on Parallel Architectures, IEEE Computer, Vol.29, No.3, pp.56-63 (1996).
3) Sejnowski, T.J. and Rosenberg, C.R.: Parallel Networks that Learn to Pronounce English Text, Complex Systems, Vol.1, pp.145-168 (1987).
4) 特集:計算物理学と超並列計算機－CP-PACS計画－,情報処理, Vol.37, No.1, pp.10-42 (Jan. 1996).
5) http://www.rccp.tsukuba.ac.jp/
6) Yasunaga, M. and Yoshida, E.: Optimization of Parallel BP Implementation: Training Speed of 1,056 MCUPS on the Massively Parallel Computer CP-PACS, IEEE and INNS, Proc. Int'l. Joint Conf. Neural Networks (IJCNN'98) (1998).
7) Kohonen, T.: Self-Organizing Maps, Springer-Verlag (1997).

(平成12年12月19日受付)