

超並列計算機 nCUBE2 上のニューラルネットワークを用いた顔画像認識法

山森 一人 阿部 亨 堀口 進

北陸先端科学技術大学院大学 情報科学研究科

〒 923-12 石川県能美郡辰口町旭台 15

あらまし

ニューラルネットワークは人工知能分野の1つとして様々な角度から研究がなされてきた。その中で、ニューラルネットワークの学習には膨大な処理時間が必要であることが分かっている。この問題を解決するため、本研究では超並列計算機を用いた並列ニューラルネットワークの学習時間がどの程度短縮可能か評価を行なった。その結果、256PEを持つ超並列計算機 nCUBE2 上では、約100倍を越える高速化が可能であることを明らかにした。

また、並列ニューラルネットワークの応用例として顔画像認識を取りあげ、モザイク顔画像を用いる場合とエッジ顔画像を用いる場合の2手法を評価した。その結果、エッジ顔画像を使用した場合、モザイク顔画像を使用した場合に比べ約 $\frac{1}{3}$ の時間で認識を行なうことができた。

和文キーワード: ニューラルネットワーク, 超並列計算機, 顔画像認識

Face Image Recognition using Neural Network on Massively Parallel Computer nCUBE2

Kunihito Yamamori

Toru Abe

Susumu Horiguchi

Graduate School of Information Science,

Japan Advanced Institute of Science and Technology

Asahi-dai 15, Tatsunokuchi, Nobigun, Ishikawa, 923-12, Japan

abstract

In this paper, we investigate parallel backpropagation algorithms for neural networks on massively parallel computers. We implement the parallel back propagation algorithms on the massively parallel computer nCUBE2 to evaluate the performance of three parallel models; unit-parallel model, learning-set-parallel model and pass-parallel model. It is confirmed that the learning-set-parallel model obtained much higher speed-up than others.

The learning-set-parallel algorithm is applied to the image recognition of human faces. It is shown that, the computation time of edge face recognition becomes 3 times faster than mosaic face image recognition on massively parallel computer, nCUBE2.

英文キーワード: Neural Network, Massively Parallel Computer, Face Image Recognition

1 はじめに

近年、人工知能処理の研究は盛んに行なわれている。その中でも、ニューラルネットワークを用いた情報処理はパターン認識や組み合わせ最適化問題などに広く応用されている。

しかし、学習アルゴリズムとしてバックプロパゲーションを用いるニューラルネットワークを実際的な問題に適用する場合、学習に膨大な時間がかかるため、スーパーコンピュータのような高速な計算機を用いたとしても実用的な時間で解を求めることができない。このため、より高速な処理を目指し、ニューラルネットワークの持つ並列性を利用して、多数のプロセッサを相互に結合した超並列計算機への実装が多数報告されるようになってきた。

Zhang らは、Thinking Machines 社製の超並列計算機 CM-2 を用いて、格子網をベースにしてニューラルネットワークを並列に実行する手法を提案した [1]。Zhang のアルゴリズムを実装した、65,536 個の PE(Processing Element) を備える CM-2 は、約 150MWUPS(Mega Weight Update Per Second) の性能を発揮している。Singer は並列ニューラルネットワークを CM-2 上へ実装した手法について報告を行ない、ニューラルネットワークとその学習アルゴリズムの 1 つであるバックプロパゲーションアルゴリズムの持つ並列性について言及している [2]。

しかしこれらの報告では、ニューラルネットワークをある特定のアーキテクチャで有効なように実装しており、基本的な並列バックプロパゲーションアルゴリズムの性能評価は十分に行なわれていない。本稿では、バックプロパゲーションアルゴリズムが持つ複数の並列性に着目し、同一の超並列計算機上にそれらを実装して性能評価を行なった。

さらに、実用問題を通じて並列ニューラルネットワークの性能を明らかにすることを目的として、並列ニューラルネットワークの応用例として、顔画像認識を採り上げる。

顔画像から個人を同定する場合には、顔の持つ濃淡情報から識別する方法や、画像からエッジを抽出してベクトル量子化しパターンマッチングを行なう方法などが提案されている。本論

文では顔画像をニューラルネットワークへ入力する方法として、モザイク化するもの [3] とエッジ情報を用いるものとの 2 種類を実装し、その性能を評価する。

2 バックプロパゲーションアルゴリズム

バックプロパゲーションアルゴリズムはニューラルネットワークを学習させる最も代表的な手法の 1 つであり、階層型ニューラルネットワークと組み合わせてごく一般的に用いられている [4]。しかし、バックプロパゲーションアルゴリズムを実用規模の問題に適用するには計算量が膨大すぎて実用的ではない。そのため、近年バックプロパゲーションアルゴリズムそのものを改良したアルゴリズムが提案されるようになってきている。例えば Fahlman は、学習パラメータの変更によるアルゴリズムの動作の違いについて詳しく調べ、quick-propagation と呼ばれる変形バックプロパゲーションアルゴリズムを提案している [5]。また、Shiffmann らは学習の間にパラメータを変更し、学習を高速化する手法について述べている [6]。しかし、アルゴリズムの改良による高速化は、逐次処理型計算機を用いる限り大幅な向上は望めない。そこで本章ではバックプロパゲーションアルゴリズムを並列化する手法について述べる。

2.1 逐次処理型バックプロパゲーションアルゴリズム

ここでは従来の逐次処理型バックプロパゲーションアルゴリズムについて簡単に述べる。図 1 に 3 層の全結合型ニューラルネットワークを示す。

階層型ニューラルネットワークでは、ネットワーク上の各ユニットで入力の総和をとり、その値に対して活性化関数を作用させたものを出力とする。この出力値を重み付きリンクを通じて接続されている次層のユニットへ伝播させていく。学習アルゴリズムとしてバックプロパゲーションアルゴリズムを用いる場合、出力層のユニットは自分自身の出力値と教師信号との誤差

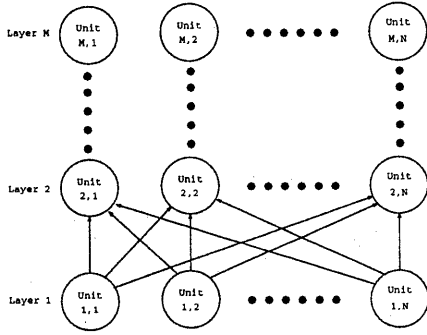


図 1: 3 層全結合型ニューラルネットワーク

をとり、その誤差をリンクを通じて入力層方向に伝播させる。誤差信号を受け取ったユニットは、その誤差にもとづいてリンクの重みを変更する。これをネットワークが平衡状態になるまで繰り返す。

ニューラルネットワークは m 層から成るとし、 k 層の i ユニットへの入力のを I_i^k 、出力を O_i^k とし、 $k-1$ 層の第 j ユニットから k 層の第 i ユニットへの結合重みを $w_{j,i}^{k-1,k}$ とする。このとき、各ユニットの入出力関係を与える関数を f とすると、

$$O_i^k = f(I_i^k) \quad (1)$$

$$I_i^k = \sum_j w_{j,i}^{k-1,k} O_j^{k-1} \quad (2)$$

となる。

誤差の 2 乗を損失関数 r とすると、ある学習パターンの組 $(\mathbf{x}, \mathbf{y})$ がネットワークに入力されたときの r は、

$$r = \sum_j (o_j^m(w, \mathbf{x}) - y_j)^2$$

と書くことができる。ここで、 w はネットワーク中で対応する重みすべてをまとめたものである。 w の修正量を求めるためには、損失関数 r の w に対する gradient を求めればよいことが証明されている [7]。これにより、個々の重みの修正は以下ようになる。

$$\frac{\partial r}{\partial I_i^k} = d_i^k \quad (3)$$

$$\Delta w_{j,i}^{k-1,k} = -\varepsilon d_i^k O_j^{k-1} \quad (4)$$

$$d_i^m = 2(O_i^m - y_i) f'(I_i^m) \quad (5)$$

$$d_i^k = \left(\sum_j w_{j,i}^{k,k+1} d_i^{k+1} \right) f'(I_i^k) \quad (6)$$

$$w(t) = w(t-1) + \Delta w \quad (7)$$

ε は学習率と呼ばれ、1 回の学習による修正の大きさを調節するパラメータである。式 (4) により重みの修正を行なう場合、学習の収束を早め、振動を減らすために、

$$\Delta w_{j,i}^{k-1,k}(t) = -\varepsilon d_i^k O_j^{k-1} + \beta \Delta w_{j,i}^{k-1,k}(t-1) \quad (8)$$

を使う場合が多い。ここで、 β は慣性項と呼ばれる小さい正の定数、 t は修正の回数を表わしている。

2.2 並列バックプロパゲーションアルゴリズム

逐次型バックプロパゲーションアルゴリズムには、3 つの並列性を見ることができる。1 つ目は、同一層内のユニットは相互に結合していないので独立に、すなわち並列に動作させることができることである。以下、この性質を基にした並列化をユニット並列モデルと呼ぶ。2 つ目は、式 (7) で示したように、各重みの修正は各学習パターンによる誤差修正量の線形和であることを利用したものである。以下、この性質を用いた並列化を学習セット並列モデルと呼ぶ。3 つ目は、入力層から隠れ層に向かって式 (1)~式 (2) に従って計算している間に、出力層から隠れ層に向かって式 (3)~式 (6) に従って誤差を逆伝搬させることができることである。以下、この性質を用いた並列化をパス並列モデルと呼ぶ。

ユニット並列モデルでは、並列計算機上の各プロセッシングエレメント (以下 PE とする) をニューラルネットワークのユニットに対応づける。このモデルは単一の PE で行なう演算が少なく、直感的に分かり易い実装法である。しか

し、1ユニット=1PEとなるので大規模なネットワークの構築には不向きであり、PE間通信が多く発生するという欠点を持つ。また、割り当てられたユニットにより演算量が異なるので、負荷分散という点でも不利である。

学習セット並列モデルでは、並列計算機上のPEに同一のニューラルネットワークを構築し、各PEで異なった学習セットを用いて学習を行った後で全体の誤差修正量を合計し、それに基づいて重みを更新する。このモデルで重みの更新をバッチ更新とした場合、PE間通信は各PEが担当する学習セットを全て処理した後のみ必要となり、ユニット並列モデルと比較して大幅にPE間通信を減らすことができる。

バス並列モデルでは上の2つのモデルと異なり、原理上単独で大幅に高速化することができない。このため、本論文では学習セット並列モデルと組み合わせて使用している。この場合、2つPEを組にして、1つがフォワードパスを、他のPEがバックワードパスを担当することになる。

3 学習速度評価

2.2章で述べた各並列バックプロパゲーションアルゴリズムは超並列計算機nCUBE2に実装し、その学習速度の高速化率について調べた。

3.1 超並列計算機nCUBE2

nCUBE2は、nCUBE社により商用化された、MIMDのメッセージパッシング型並列計算機である。各PEは64bitのCPU、FPU、メモリ、ネットワークインターフェースを持つ。本稿で用いたnCUBE2はPEを256台搭載しており、これがハイパーキューブ網で結合されている。PEのうち16台はホストとのI/Oを行なう機能を持ち、メモリが16MB搭載されている。残りの240台のPEが持つメモリは4MBである。

ソフトウェア環境については、標準でANSI CとFortran 77をサポートしており、ライブラリとしてホストプログラミングを行なうncube、パラレルライブラリnparaなどが用意されている。表1にnCUBE2の諸元をまとめる。

表 1: nCUBE/2 諸元

PE 数	256PE
CPU	64bit, 10MIPS
FPU	32bit 3.3MFLOPS 64bit 2.4MFLOPS
メモリ	0 ~ 15: 16MB 15 ~ 255: 4MB
通信チャネル	シリアル 2.2MB/s
通信オーバーヘッド	send 140(μ sec) receive 55(μ sec)
ディスク	16GB, SCSI-2
言語	C, Fortran, Assembler

3.2 学習速度の測定

2.2で述べた3種類の並列バックプロパゲーションアルゴリズムを超並列計算機nCUBE2上に実装し、学習速度について測定した。各並列モデルの評価には、Encode/Decode問題を使用した。ユニット並列モデルではnCUBE2のメモリ上の制限のため、parity問題を使用し、学習回数も少ないものとなっている。また、本稿では学習速度の計測を主眼としたため、各並列モデルとも収束状況に関わらず学習回数は250Epochとした。

3.2.1 ユニット並列モデル

nCUBE2のメモリ上の制限により、ユニット並列モデルでは1024パターンのparity問題を10, 20, 30Epoch回学習を行ない、その時の実処理時間とWUPS値を表2に示す。このときのニューラルネットワークの構成は、入力ユニット10、隠れユニット5、出力ユニット1である。

表 2: ユニット並列モデルの学習時間

Epoch	処理時間 (秒)	WUPS
10	505.8	101.2
20	1015.5	100.8
30	1505.5	102.2

表2で示したように、ユニット並列モデルはWUPS値が約100程度と非常に低速な学習速度しか得られていない。これはユニット並列モ

デルでは、PE 間通信がリンクの数だけ行なわれること、nCUBE2 ではメッセージの受信が同期して行なわれるために処理がブロックされることなどが原因であると考えられる。

3.2.2 学習セット並列モデル

学習セット並列モデルでは、1024~8192 パターンの Encode/Decode 問題を用いて学習速度を測定した。このときのニューラルネットワークの構成を表3に示す。

表 3: ニューラルネットワークのユニット数

ビット数	入力	隠れ	出力
10	10	5	10
11	11	6	11
12	12	6	12
13	13	7	13

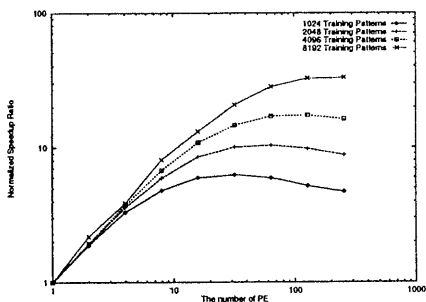


図 2: 学習セット並列モデルの速度向上率

図 2 に Encode/Decode 問題を処理したときの高速化率を示す。図 2 より、学習パターン数が多くなる程高速化の度合いが大きいことが分かる。この場合、学習パターン数 8192 の Encode/Decode 問題を、256 個の PE で処理した時に、1PE 時の約 33 倍の処理速度が得られた。しかし、次のような問題が考えられる。

- 使用した PE 数に比べ、速度向上率が小さい。

- PE 数を増やすことで、速度向上率が低下する場合がある。

これは学習セットを分割することで 1PE 当たりの処理時間は短縮できるものの、相対的に PE 間通信に必要な時間が増大するためと考えられる。この問題点を解決するため、通信を減らすような改良が必要である。

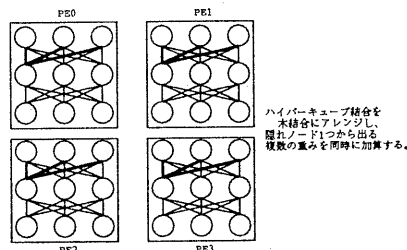


図 3: 複数重みの同時加算

図 3 に示すように複数の重みの値を同時に加算し、その結果をブロードキャストするように変更を加えたときの、Encode/Decode 問題の速度向上率を図 4 に示す。

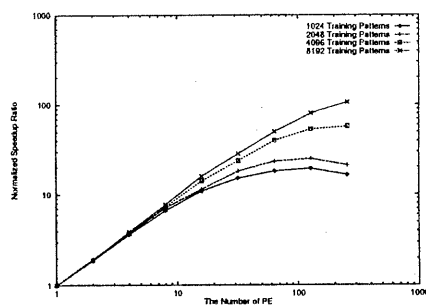


図 4: 複数重み同時加算時の速度向上率

このとき使用したニューラルネットワークの構成は表3と同じである。複数の重みを同時に加算するよう改良した場合、学習パターン数 8192 の Encode/Decode 問題を処理した時、1PE 時に比較して 256PE 時に約 106 倍の高速化が達成できた。このことから、並列計算機上でアプリケーションを実行する場合には、通信量を極力少なくする必要があるといえる。しかし、学

学習セット並列モデルでは、PE 数に対して十分に多い学習パターンを用意しないと、PE の使用効率が低下してしまう。このため、学習セット並列モデルは処理するパターンが多い大規模な問題を並列に処理する場合に適しているといえる。

3.2.3 パス並列モデル

学習セット並列モデルとパス並列モデルを組み合わせ、学習セット並列モデルと同条件で、Encode/Decode 問題を用いて学習時間の変化を調べた。Encode/Decode 問題における速度向上率を図 5 に示す。

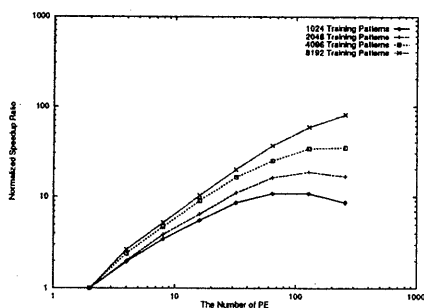


図 5: パス並列モデルの速度向上率

パス並列モデルでは、PE2 つを組にして用いるためにみかけの PE 数が減少する。このため、学習セット並列モデルと同じ PE 数では処理時間が多く必要となるが、単位 PE で処理する学習パターン数を同じにして比較すると約 1.5 倍の高速化となった。

4 顔画像認識

実際の問題を通じて並列ニューラルネットワークの有効性を確認するため、ニューラルネットワークを用いた顔画像認識を行なう。

4.1 モザイク顔画像認識

入力画像として、同一条件で撮影された顔画像を用意する。画像のサイズは 512×512 pixel であり、256 階調の濃度情報を持つグレースケール

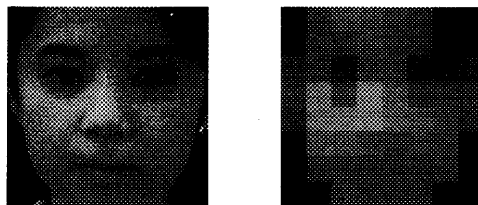


図 6: 切り出した画像とモザイク画像の例

ル画像である。元画像には背景や髪などが含まれている。これらの不要情報を削除してデータ量を減らすため、 256×256 pixel のサイズで顔の中心部を手動で切り出す。この切り出した画像をモザイク化する。モザイク化にあたっては、切り出した画像を 8×8 の 64 ブロックに分割し、各ブロック内の平均輝度値をそのブロックの代表値としてニューラルネットワークへの入力とした。切り出した顔画像と、モザイク化した顔画像の例を図 6 に示す。顔画像は 55 人用意している。

本稿では各個人の識別を目的としているため、各個人に対応する出力ユニットを用意している。このため、ニューラルネットワークの構成は、入力ユニット 64、隠れユニット 40、出力ユニット 55 となる。教師信号には、該当する人物を担当する出力ユニットには 0.9、それ以外の出力ユニットには 0.1 を与えた。入力パターンが分離できたかどうかは、該当出力ユニットの出力のみが 0.7 以上、それ以外のユニットの出力が 0.3 以下であることで判定した。

ここで使用した並列学習モデルは、学習セット並列モデルを使用した。nCUBE2 のネットワークポロジの関係上、学習セット数は 2^n であることが望ましいので、学習セットには 9 人分のデータを 2 度登録して 64 人分とし、64(55) 人分の顔画像認識を行なった。学習率 α は 0.8、慣性項 β は 0.2 として学習を行ない、25000 Epoch で各学習パターンを完全に分離することができた。このときの並列化による速度向上率を表 4 に示す。

本実験で用いた学習セットのパターン数が 55 と比較的少数だったため、速度向上率は 5 倍程

表 4: モザイク顔画像認識での速度向上率

PE 数	時間 (秒)	WUPS	高速化率
1	46650	75456	1.0
2	26166	134526	1.78
4	15723	223876	2.97
8	11059	318293	4.22
16	9273	379597	5.03
32	8936	393912	5.22
64	9317	377804	5.01

度にとどまる結果となった。これをさらに高速化するためには、より多人数の学習セットを必要とするような問題を用いたり、より通信コストの小さい並列計算機が必要となると思われる。

4.2 エッジ顔画像認識

入力画像としては、モザイク顔画像による個人識別と同じように切り出した画像を用いる。画像をエッジ化してニューラルネットワークへ入力するに当たり、平滑化フィルタで画像を処理したあと、ガウシアン-ラプラシアンによりエッジ抽出を行なった。エッジ抽出前後の画像の例を図 7 に示す。なお、図は見やすくするために白黒を反転させている。



図 7: エッジ画像の例

その後、画像を 4×4 のブロックに分割し、図 8 に示すパターンを左下→ 右上の順に走査

し、一致したパターン数を正規化した値をネットワークへの入力としている。

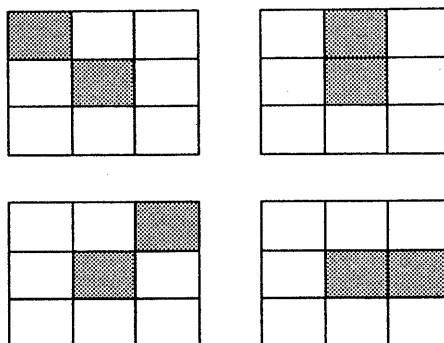


図 8: 検索パターン

画像をブロックに分割することで、エッジの位置をおおまかに把握し、上の 4 つのパターンを 4×4 のブロック内で走査することで、ブロック内のエッジ情報の方向とその長さの概略値が得られる。これにより、エッジ情報をベクトル化してパターンマッチングなどを行なうより簡単に、各顔画像をコード化することができる。

16 の各ブロックで 4 つのパターンを走査することで、ニューラルネットワークへ入力する値は 64 個となる。よって、エッジ顔画像認識で用いたニューラルネットワークの構成は、モザイク顔画像認識を全く同じものを使用した。また、入力の分離条件もモザイク顔画像認識と同じものを用いた。

認識実験に関しては、モザイク顔画像認識と同じく 55 人分の顔画像にダミーデータを加えた 64 個の学習パターンについて行なった。学習率、慣性項ともにモザイクの場合と同じにそれぞれ 0.8, 0.2 としている。このとき、入力パターンを 100% 分離するまでに 8000 Epoch の学習が必要であった。このときの学習時間と速度向上率を表 5 に示す。

表 5 から分かるように、モザイク画像を用いたときと比較して、ネットワークの構造自体は同じなので速度向上率もほぼ同じ値となっている。しかし、収束に必要な学習回数が $\frac{1}{3}$ となっており、より短時間で学習が可能であることを

表 5: エッジ顔画像認識での速度向上率

PE 数	時間 (秒)	WUPS	高速化率
1	14942	75385	1.0
2	8374	134512	1.78
4	5052	222961	2.96
8	3580	314637	4.17
16	2974	378749	5.02
32	2898	388682	5.16
64	3044	370039	4.91

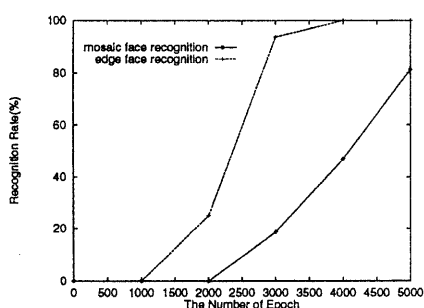


図 9: 学習回数と認識率の比較

示している。モザイク顔画像認識とエッジ顔画像認識とで、同じ学習回数でどれだけの認識率が得られているかを示したものを図 9 に示す。

5 まとめ

超並列計算機 nCUBE2 上に並列バックプロパゲーションアルゴリズムを 3 種類実装し、その速度向上率を計測した。その結果、超並列計算機の PE1 つにニューラルネットワークのユニットを割り当てるユニット並列モデルでは、現状のメッセージパッシング型並列計算機では通信のオーバーヘッドが大きく、実用的な速度は得られなかった。一方、学習セット並列モデルを使用した場合には、学習パターンが増えるほど高速化でき、大規模な問題に適したモデルであることが明らかとなった。

また、並列ニューラルネットワークの応用例として顔画像認識を行なった。その際、顔画像

をモザイク化して認識を行なう方法と、エッジ化して認識を行なう方法の 2 種類について評価を行なった。その結果、エッジ画像を用いた場合、モザイク画像を用いた時に比べて約 $\frac{1}{3}$ の学習時間で、55 人分の顔画像を完全に認識できた。今後の課題としては、ノイズや位置ずれに対するロバストネスや、人数が多くなったときのスケーラビリティについての評価が必要である。

謝辞

本研究の一部は文部省科学研究費を用いて行なわれた。関係各位に感謝する。

参考文献

- [1] Xiru Zhang, David L. Waltz, et al. "The Back-propagation Algorithm On Grid and Hypercube Architecture". Technical report RL90-3, 1990.
- [2] Alexander Singer. "Implementation of Artificial Neural Network on the Connection Machine". Technical report RL90-2, 1990.
- [3] 小杉 信. "モザイクとニューラルネットを用いた顔画像の認識". 信学論 (D-II), J76-D-II(6):pp.1132 - 1139, June 1993.
- [4] D.E.Rumelhart, J.L.McClelland and PDP Research Group. "Parallel Distributed Processing", volume 1. MIT Press, 1986.
- [5] Scott E.Fahlman. "An Empirical Study of Learning Speed in Back-Propagation Network". CMU-CS-88-162, 1988.
- [6] W.Schiffman, M.Joost, and R.Werner. "Optimization of the Backpropagation Algorithm for training Multilayer Perceptrons". 1993.
- [7] 安西 祐一郎. "認識と学習". 岩波講座ソフトウェア科学 16". 岩波書店, 1989.