岡山理科大学紀要 第36号 A pp. 155-164(2000)

効率的なバックプロパゲーション学習の研究

橋本　麻希・井上　浩孝*・成久　洋之**
岡山理科大学大学院工学研究科修士課程情報工学専攻
*岡山理科大学大学院工学研究科博士課程システム科学専攻
**岡山理科大学工学部情報工学科

(2000年11月1日　受理)

1. はじめに

ニューラルネットワーク（Neural Network：以下，NNと記す）は学習，識別，パターン認識等の知的処理の分野で広範囲に応用されようとしている．これは人間の脳細胞の情報処理形態を模倣したもので，人工的にニューロンを結合させて構成したものである．

NNでの標準的学習則としては逆伝播則が一般的に使用されている．これは多層構造からなるfeed-forward型NNにおいて適用される学習則で，バックプロパゲーション（Back Propagation：以下，BPと記す）とも呼ばれている．BP学習則の原理は単純で強靱性に富んだものであるが，取り扱うデータ構造やデータ数によって学習効率が影響されやすく，NNの規模が大きくなると処理に時間がかかり過ぎる欠点がある．

本研究はBP則の効率化のため，Epsilon-BP法[1)]やAdaptive step algorithms[2)]を使用した場合の有効性につき検討するものである．

2. BP学習法

BP学習法とは教師あり学習の1つで，入力信号から得られた出力と教師データ（または出力目標データと言う）の情報を逆伝播することによって，図1の重み（または結合荷重と言う）w_{ji}を更新する．つまり，重みの繰り返し最適化を図ることがNNでの学習に対応する．次にBP学習法の原理と慣性項を含むBP学習法について記述する．

2.1 BP学習法の原理

BP学習では入力データX_iと出力目標データD_iとの対$(X_i, D_i)(i = 1, \ldots, p)$をニューラルネット$A$に与えることで，入出力関数$F_{W,A}$が入力空間を覆うように重み（または結合荷重と言う）Wを探索しようとするものである．学習は式(1)に示すように，目標値と出力値の誤差に関するコスト関数（評価関数または平均二乗誤差関数とも言う）

$$E(W) = \frac{1}{p}\sum_{i=1}^{p} \mid F_{W,A}(X_i) - D_i \mid^2 \tag{1}$$

上での下降勾配を考慮することで重みの繰り返し最適化を図るものである．つまり出力目標データとNN出力の誤差を小さくする重み係数を，重み係数空間での微分方向に探索するというものである．よって，結合荷重の変化量$\Delta_{ji}(W_n)$には，次式のような関係が成り立つ．

$$\Delta_{ji}(W_n) \propto -\frac{\partial E}{\partial w_{ji}} \tag{2}$$

図1のようにc個の接続をもつニューロンjのn回目における重みW_nを以下のようにすると，

$$W_n = (w_{ji_1}(n) \quad w_{ji_2}(n) \quad \ldots \quad w_{ji_c}(n))$$

（但し，nは学習回数を示し，繰り返し回数とも言う）ニューロンjに対する入力は式(3)のようになる．

$$a_j(n) = \sum_{i \in A(j)} w_{ji}(n) x_i(n) \tag{3}$$

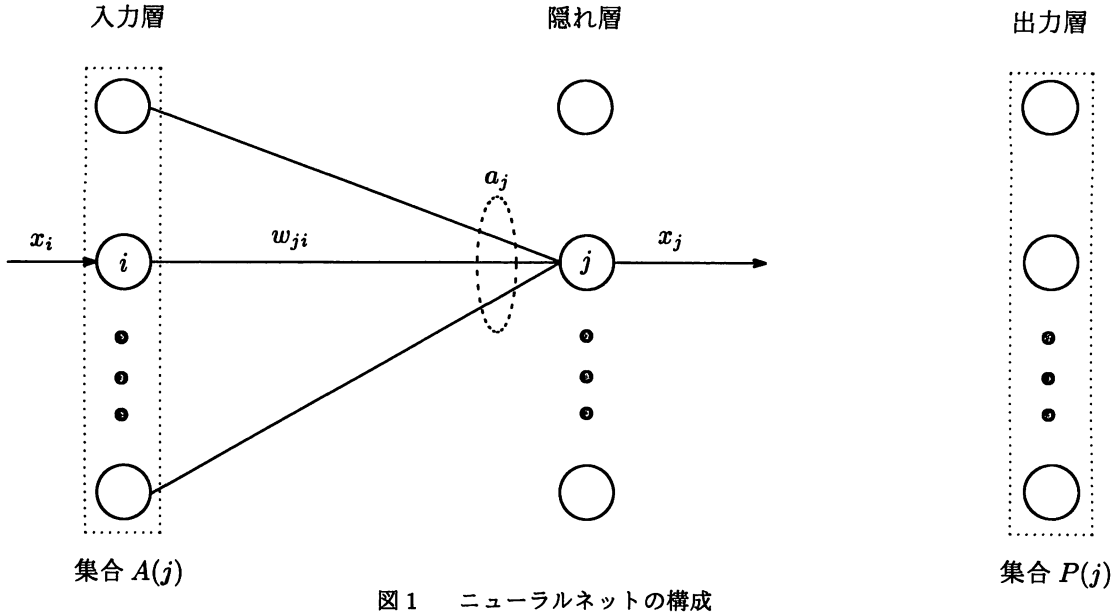

図 1　ニューラルネットの構成

式 (3) の $A(j)$ はニューロン j の前の層にあるニューロンの集合を表している．このとき，ニューロン j の出力は，次式によって得られる．

$$x_j(n) = f(a_j(n)) \tag{4}$$

ここで式 (4) の入出力関数には次式のシグモイド関数を用いた．なお，本研究では式 (5) のシグモイド関数の傾き θ を $\theta = 1.0$ とした．

$$f(a) = \frac{1}{1 + \exp(-\theta a)} \tag{5}$$

次に，結合荷重は以下の式によって更新される．

$$w_{ji}(n+1) = w_{ji}(n) + \Delta_{ji}(W_n) \tag{6}$$

結合荷重は式 (1) を最小になるように変化させればいいので，$\Delta_{ji}(W_n)$ は次のようになる．

$$\Delta_{ji}(W_n) = \eta \delta_{ji} x_j(n) \tag{7}$$

ニューロン j が出力層のとき　$\delta_{ji} = (d_j - x_j(n))x_j(n)(1 - x_j(n))$

ニューロン j が出力層以外のとき　$\delta_{ji} = x_j(n)(1-x_j(n)) \sum_{k \in P(j)} (\delta_{ji}^k w_{ji})$

ここで，式 (7) の η は学習率（または学習係数と言う）を意味しており，この係数が大きい程 1 回の重みの修正量は大きくなる．しかし，学習率を大きくすると学習時に結合荷重の振動を引き起こすことになる．従って，この振動が起こらない範囲で学習係数を大きくとることにより効率的な学習を行うことができる．また，$P(j)$ はニューロン j の後ろの層にあるニューロンの集合を表している．

2.2　慣性項を含む BP 学習法

式 (7)，式 (8) で表される BP 学習法では，$\partial E/\partial w$ に比例した値によって結合荷重の修正量が決定される．一般にこのような勾配降下法では，修正量は微小な方がよいとされている．しかし修正量を小さくすると，学習速度は遅くなってしまう．そこで探索点の動きが振動しない程度に学習速度を上げる方法の 1 つとして，式 (8) の右辺第 2 項のような慣性項を導入して結合荷重の修正を行うことで学習の発散を防ぎ，学習の高速化を図る．

$$\Delta_{ji}(W_{n+1}) = \eta \delta_{ji} x_j(n) + \alpha \Delta_{ji}(W_{n-1}) \tag{8}$$

ただし，α は，現在の結合荷重の変化方向に前回の結合荷重の変化の影響を及ぼさせる為の定数.

3. Epsilon-BP 法

従来の BP 学習では全てのパターンに対して毎回重みを修正してきたが，Epsilon-BP 学習は定数 ϵ を設定する事によって，各パターンごとの誤差が ϵ より大きいときだけ重みを修正する．以下に Epsilon-BP 学習法のアルゴリズムを示す.

Epsilon-BP 学習アルゴリズム

step1. 入力パターンの提示
ある入力パターンを NN に提示し，その入力パターンに対する出力結果 x_j を得る.

$$x_j(n) = f(a_j(n))$$

step2. 各ユニットの誤差の計算
step1. の出力結果と教師信号の誤差を計算する.

ニューロン j が出力層のとき $\quad \delta_{ji} = (d_j - x_j(n))x_j(n)(1 - x_j(n))$

ニューロン j が出力層以外のとき $\quad \delta_{ji} = x_j(n)(1-x_j(n)) \sum_{k \in P(j)} (\delta_{ji}^k w_{ji})$

step3. 結合荷重の修正量の計算

$$\Delta_{ji}(W_n) = \eta \delta_{ji} x_j(n)$$

step4. $|d_j - x_j(n)| > \epsilon$ のとき，結合荷重の修正

$$\Delta_{ji}(W_{n+1}) = \eta \delta_{ji} x_j(n) + \alpha \Delta_{ji}(W_{n-1})$$

step5. 全パターンについて step1. から step4. を繰り返す

step6. 収束判定
収束の条件を満たすまで step1. から step5. を繰り返す

4. Adaptive step algorithms

従来の BP 学習法は学習率が一定だった為，収束に時間がかかる場合があった．これに対して Adaptive step algorithms とは学習率を各重みに対して与えることにより，従来の BP 法よりも高速に学習を実施させようとするものである．式 (7) の学習率 η を γ_{ji} とし，以下に Adaptive step algorithms の 1 つである Silva and Almeida's algorithm と Delta-bar-delta の理論と結合荷重修正量の計算式を示す.

4.1 Silva and Almeida's algorithm

Silva and Almeida's algorithm とは学習率を決定する際に，評価関数の現在の状態と過去の状態を比べ，学習率を変動させる事により収束を速くするものである．現在の勾配と 1 つ前の勾配が同符号の場合には，評価関数上に大きな変化がないと考えられる為，学習率を大きくとり収束を加速させる．また現在の勾配と 1 つ前の勾配が異符号の場合には，重みの学習率を小さくして結合荷重が振動するのを防ぎ，従来の BP 法よりも学習を速くする.

以下に Silva and Almeida's algorithm の結合荷重修正量の計算式を示す.

$$\gamma_{ji}(n+1) = \begin{cases} \gamma_{ji}(n)u & \text{if } \nabla_{ji}E(n) \cdot \nabla_{ji}E(n-1) \geq 0 \\ \gamma_{ji}(n)d & \text{if } \nabla_{ji}E(n) \cdot \nabla_{ji}E(n-1) < 0 \end{cases}$$

ただし，$u > 1$，$d < 1$ の定数.

$$\Delta_{ji}(w_n) = -\gamma_{ji}(n) \nabla_{ji}E(n)$$

4.2 Delta-bar-delta

Delta-bar-delta 法は Silva and Almeida's algorithm に学習率の更新基準となる δ を設置したものである．この δ は各反復における勾配の重み付き平均で，平滑化微分と呼ばれ，評価関数の大域的な勾配を表している．この平滑化微分と勾配の符号が異なる場合は，結合荷重が振動してると考えられる為，振動を起こしている重みの学習率を小さくして，その振動を抑制することにより収束を速くする．これに対して平滑化微分と勾配が同符号の場合は，評価関数上に大きな変化がないと考えられる為，学習率を大きくとることにより収束を速くする．

以下に Delta-bar-delta の結合荷重修正量の計算式を示す．

$$\gamma_{ji}(n+1) = \begin{cases} \gamma_{ji}(n) + u & \text{if } \nabla_{ji}E(n) \cdot \delta_{ji}(n-1) > 0 \\ \gamma_{ji}(n)d & \text{if } \nabla_{ji}E(n) \cdot \delta_{ji}(n-1) < 0 \\ \gamma_{ji}(n) & \text{otherwise,} \end{cases}$$

$$\text{ただし,} \quad \delta_{ji}(n) = (1-\phi)\nabla_{ji}E(n) + \phi\delta_{ji}(n-1)$$

$$\Delta_{ji}(w_n) = -\gamma_{ji}(n)\nabla_{ji}E(n)$$

5. 実験内容

従来の BP 法（慣性項付き BP 法を含む）と上記で述べた学習法（Epsilon-BP, Silva and Almeida's algorithm, 及び Delta-bar-delta）を用いて，初期の重みをランダムに変えて，Exor 問題，数字認識問題，アルファベット認識問題に対してそれぞれ 100 回ずつ実験を実施し比較検討を行った．このとき各パラメータを表 1 のように設定し，各学習法とも繰り返し回数を 5000 回とした．ここで表中の Silva は Silva and Almeida's algorithm, Delta は Delta-bar-delta 学習法の事を表す．また，Epsilon-BP 法では ϵ を $\epsilon = 0.001$, 0.005, 0.010, 0.025, 0.050, 0.100, 0.200 と変化させて実験を行った．

表 1　学習パラメータ

学習方法	初期の重みの範囲	学習率
BP	$-1.0 \sim +1.0$	$\eta = 1.5$
慣性項付き BP	$-1.0 \sim +1.0$	$\eta = 0.35,\ \alpha = 0.9$
Epsilon BP	$-0.1 \sim +0.1$ と $-1.0 \sim +1.0$	$\eta = 1.5$
Epsilon 慣性項付き BP	$-0.1 \sim +0.1$ と $-1.0 \sim +1.0$	$\eta = 0.35,\ \alpha = 0.9$
Silva	$-1.0 \sim +1.0$	初期の$\gamma_{ji} = 0.9,\ u = 1.0005,\ d = 0.9995$
Delta	$-1.0 \sim +1.0$	初期の$\gamma_{ji} = 0.9,\ u = 0.02,\ d = 0.995,\ \phi = 0.2$

5.1 Exor 問題

表 2　Exor 問題の状態表

x_1	x_2	x_j
0	0	0
0	1	1
1	0	1
1	1	0

Exor 問題の状態表を表 2 に示す．この問題を，入力層のユニット数 (x_1, x_2) を 3 個，中間層のユニット数を 3 個，及び出力層のユニット数 (x_j) を 1 個とした NN 構造を用いて実験を行った．

5.2 数字認識問題

数字認識問題の入力例と出力目標データ例を図 2 に示す．0 ～ 9 の各数字を 49(7×7) 個の入力データとして，中間層のユニット数を 50 個，及び出力層のユニット数を 10 個とした NN 構造を用いて実験を行った．

5.3 アルファベット認識問題

アルファベット認識問題の入力例と出力目標データ例を図 3 に示す．A ～ Z の各アルファベットを 100(10×10) 個の入力データとして，中間層のユニット数を 101 個，及び出力層のユニット数を 26 個とした NN 構造を用いて実験を行った．

```
0 0 0 0 0 0 0
0 1 1 1 1 1 0
0 1 0 0 0 1 0
0 1 0 0 0 1 0
0 1 0 0 0 1 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0
```

(a) 数字“0”の入力データ

```
t[0] = 1
t[1] = 0
t[2] = 0
t[3] = 0
t[4] = 0
t[5] = 0
t[6] = 0
t[7] = 0
t[8] = 0
t[9] = 0
```

(b) 数字“0”の出力目標データ

図2 数字“0”の場合の入力データ (a) と出力目標データ (b)

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 1 1 1 1 1 1 0 0
0 1 0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
```

(a) アルファベット“A”の入力データ

```
t[0]=1   t[10]=0   t[20]=0
t[1]=0   t[11]=0   t[21]=0
t[2]=0   t[12]=0   t[22]=0
t[3]=0   t[13]=0   t[23]=0
t[4]=0   t[14]=0   t[24]=0
t[5]=0   t[15]=0   t[25]=0
t[6]=0   t[16]=0
t[7]=0   t[17]=0
t[8]=0   t[18]=0
t[9]=0   t[19]=0
```

(b) アルファベット“A”の出力目標データ

図3 アルファベット“A”の場合の入力データ (a) と出力目標データ (b)

6. 結果と考察

以下に示す結果は上記で述べた学習法（従来の BP 法，Epsilon-BP，Silva and Almeida's algorithm，及び Delta-bar-delta）を各問題に対してそれぞれ 100 回ずつ実験し，その平均を求めたものである．

表 3〜表 9 及び図 4〜図 8 で示している BP-kansei は慣性項付き BP 法，Silva は Silva and Almeida's algorithm，Delta は Delta-bar-delta 学習法の事を表す．次に表中の収束回数とは，100 回の実行回数うち何回平均二乗誤差が 0.001 より小さくなったかを示したものである．また学習回数とは，収束した時の平均学習回数を表しており，小数点以下を四捨五入したものである．処理時間とは，収束した場合の平均処理時間を表している．そして最終誤差とは，100 回実行した時の最終誤差の平均を求めたものである．

6.1 Epsilon-BP 法による実験結果

Epsilon-BP 法を使用した場合の Exor 問題，数字認識問題，及びアルファベット認識問題に対する実験結果をそれぞれ表 3〜表 5 に示す．また，表 6 に Epsilon-BP 法を用いた場合の各問題に対する最終誤差の平均を表す．ここで表中の W とは初期の重みの範囲を示しており，初期の重みの範囲が $-0.1 \sim +0.1$ にある時を ± 0.1，$-1.0 \sim +1.0$ にある時を ± 1.0 として表している．

学習回数，処理時間，収束回数，及び最終誤差を総合的に考えると，Exor 問題では表 3 及び表 6 より $\epsilon = 0.005$，$W = \pm 1.0$ の時の慣性項付き BP 法が良い結果を得ることができた．数字認識問題の場合は，表 4 及び表 6 より $\epsilon = 0.005$，$W = \pm 1.0$ の場合の BP 法が他のパラメータの値の時よりも比較的に良い結果を得ることができた．またアルファベット認識問題の場合は，表 5 及び表 6 より $\epsilon = 0.005$，$W = \pm 1.0$ の時の BP 法が学習回数，収束回数と一番良い結果を得ることができ，処理時間，及び最終誤差を含めて総合的に考えても他のパラメータよりも良いと言える．また初期の重みの範囲に着目すると，表 3〜表 5 より学習回数の面では Exor 問題は BP 法，慣性項付き BP 法に関わらず $W = \pm 1.0$ の方が $W = \pm 0.1$ よりも良い結果を得た．しかし数字認識問題，及びアルファベット認識問題に対しては BP 法では $W = \pm 1.0$ の方が良いが，慣性項付き BP 法では $W = \pm 0.1$ の方が良い結果を得ることが出来た．これは BP 法が初期の結合荷重に左右されやすいからではないかと考えられる．次に表 3〜表 6 より ϵ に着目すると，ϵ が大きくなると収束回数が減少している．また最終誤差は ϵ が小さいほど良い結果を得る事ができた．これ

は Epsilon-BP 法が各カテゴリ（または入力パターンと言う）で $|d_j - x_j(n)| < \epsilon$ となると重みの更新をしなくなる為だと考えられる．本研究では，学習回数，処理時間，収束回数，及び最終誤差と総合的に見て，各問題とも $\epsilon = 0.005$ の時に比較的に良い結果を得る事が出来た．

6.2 各学習法による実験結果

従来の BP 法，Epsilon-BP 法，Silva and Almeida's algorithm，Delta-bar-delta を使用した場合の Exor 問題，数字認識問題，アルファベット認識問題に対する実験結果をそれぞれ表 7〜表 9 に示す．また，それぞれの問題における各学習法の収束特性（平均二乗誤差を Error として縦軸に，繰り返し回数を Iterations として横軸に示したもの）を図 4〜図 8 に示す．ここで，数字認識問題とアルファベット認識問題は収束特性が似ていた為，図 6，図 8 に Iterations を狭めた収束特性を表す．なお，表 7〜表 9 で示している Epsilon-BP 法は，各問題に対して良い結果が得られ時（各問題とも $\epsilon = 0.005$，$W = \pm 1.0$ で，Exor 問題では慣性項付き BP 法，数字とアルファベット認識問題では BP 法を用いた場合の実験結果）のものを使用している．

Exor 問題に対しては表 7，図 4 より学習回数，処理時間，収束回数，最終誤差，及び収束特性のどれを見ても，Epsilon-BP 法が一番良い結果を得ている．次に数字認識問題は表 8 より，学習回数では Delta-bar-delta，処理時間では BP 法が良い結果を得ることができた．収束回数では慣性項付き BP 法以外の学習法で 100 回全て収束することができた．最終誤差では BP 法及び Delta-bar-delta が一番良い結果を得ることができたが Epsilon-BP 法と Silva and Almeida's algorithm も 0.000002 とほぼ BP 法や Delta-bar-delta と変わらない結果を得た．また図 5 より BP 法，Epsilon-BP 法，Silva and Almeida's algorithm，及び Delta-bar-delta はほぼ同じ収束特性をしていることがわかる．そこで，図 6 で Iterations の範囲を 0〜40 にして，各学習法の収束特性を見ると，Epsilon-BP 法が一番良いと言える．またアルファベット認識問題は図 7，図 8 より，Iterations が 100 回までは BP 法，Epsilon-BP 法，及び Silva and Almeida's algorithm の収束特性は似ているが，1000 回を過ぎたあたりから Silva and Almeida's algorithm の Error が少し増加している．これは過学習，つまり学習をさせ過ぎた事によるものだと思われる．また表 9 より，Delta-bar-delta の学習回数は他の学習法と比較して少ない学習回数で収束しているが，最終誤差は 0.225241 と他の学習法に比べてあまり良くない．これは Delta-bar-delta が収束できた場合（72 ／ 100 回）は少ない学習回数で収束できるが，収束できなかった場合（28 ／ 100 回）はあまり学習できない為に最終誤差が大きくなったものだと考えられる．よって，実験結果を総合的に考えると，BP 法と Epsilon-BP 法が同じくらい良い結果を得た．次に表 8，表 9 より数字認識問題，及びアルファベット認識問題では共に BP 法より Delta-bar-delta の方が学習回数が少ないにも関わらず，処理時間は BP 法の方が速い．これは学習率を各重みごとに設置し，それを更新していく為に BP 法よりも 1 回の学習に時間がかかってしまうからだと考えられる．

7. おわりに

本研究では，従来の BP 法や Epsilon-BP 法，及び Adaptive step algorithms を用いた場合の有効性につき検討した．その結果，Epsilon-BP 法は従来の BP 法よりも良い結果を得ることが出来た．これは Epsilon-BP 法が各カテゴリの情報を生かす為，従来の BP 法よりも比較的速く Error を減少させる事が出来たと考えられる．よって，Epsilon-BP 法は従来の BP 法よりも有効であると言える．しかし Adaptive step algorithms の一つである Silva and Almeida's algorithm や Delta-bar-delta は，学習回数の面から見ると数字認識問題やアルファベット認識問題の場合，BP 法とほぼ変わらない結果を得たが，処理時間及び最終誤差の面ではあまり良い結果は得られなかった．よって，Adaptive step algorithms は従来の BP 法よりも必ずしも有効であるとは考えられないと言える．

参考文献

1) Luis A. Trejo, Carlos Sandoval ; Improving Back-Propagation : Epsilon-Back-Propagation, in *From Natural to Artifical Neural Computation*, ed. Jose' Mira, Francisco Sandoval, pp.427–432, Springer-Verlag, Berlin, 1995.

2) R.Rojas ; Adaptive step algorithms, in *Neural Networks – A Systematic Introduction*, pp.204–209, Springer-Verlag, Berlin, 1996.

3) 安居院 猛, 長橋 宏, 髙橋 裕樹 ; ニューラルプログラム, (株) 昭晃堂, 東京都, 1993.

4) 中野 馨, 石川 眞澄, 戸田 尚宏, 臼井 支朗, 大森 隆司, 倉田 耕治 ; ニューラルネットワーク理論, ニューロ・ファジィ・*AI* ハンドブック, (社) 計測自動制御学会, pp.69–98, (株) オーム社, 1994.

表 3　Epsilon-BP 法を用いた場合の Exor 問題に対する実験結果

ϵ	W	BP			慣性項付き BP		
		学習回数	処理時間 [sec]	収束回数	学習回数	処理時間 [sec]	収束回数
0.001	± 0.1	2470	0.039535	43	840	0.013596	89
	± 1.0	1023	0.016056	71	478	0.008214	84
0.005	± 0.1	2536	0.041622	37	826	0.013297	91
	± 1.0	1049	0.016757	74	405	0.005978	92
0.010	± 0.1	2224	0.036579	38	811	0.012299	87
	± 1.0	1024	0.016119	67	471	0.007791	86
0.025	± 0.1	2340	0.038293	41	894	0.014022	92
	± 1.0	1006	0.015224	67	402	0.006471	85
0.050	± 0.1	—	—	—	—	—	—
	± 1.0	—	—	—	—	—	—
0.100	± 0.1	—	—	—	—	—	—
	± 1.0	—	—	—	—	—	—
0.200	± 0.1	—	—	—	—	—	—
	± 1.0	—	—	—	—	—	—

表 4　Epsilon-BP 法を用いた場合の数字認識問題に対する実験結果

ϵ	W	BP			慣性項付き BP		
		学習回数	処理時間 [sec]	収束回数	学習回数	処理時間 [sec]	収束回数
0.001	± 0.1	1103	1.040200	100	425	0.456000	100
	± 1.0	199	0.188900	100	1867	2.003478	92
0.005	± 0.1	1053	0.987200	100	404	0.441000	100
	± 1.0	189	0.181000	100	2132	2.267765	85
0.010	± 0.1	1162	1.044400	100	425	0.434300	100
	± 1.0	199	0.187400	100	1897	2.021163	86
0.025	± 0.1	1440	1.166465	99	436	0.419500	100
	± 1.0	189	0.177000	100	2072	2.207126	87
0.050	± 0.1	1636	1.234694	98	506	0.454500	100
	± 1.0	194	0.173400	100	1974	2.095000	88
0.100	± 0.1	—	—	—	—	—	—
	± 1.0	156	0.130000	9	1691	1.782381	21
0.200	± 0.1	—	—	—	—	—	—
	± 1.0	—	—	—	—	—	—

表 5　Epsilon-BP 法を用いた場合のアルファベット認識問題に対する実験結果

ϵ	W	BP			慣性項付き BP		
		学習回数	処理時間 [sec]	収束回数	学習回数	処理時間 [sec]	収束回数
0.001	± 0.1	782	8.861800	100	313	4.620000	100
	± 1.0	279	3.240800	100	714	10.431900	100
0.005	± 0.1	772	8.788100	100	312	4.783700	100
	± 1.0	270	3.067600	100	664	9.598000	100
0.010	± 0.1	795	8.157800	100	316	4.715300	100
	± 1.0	273	3.116900	100	541	7.726400	100
0.025	± 0.1	786	7.501900	100	309	4.336400	100
	± 1.0	285	3.156800	100	639	8.964800	100
0.050	± 0.1	799	7.227700	100	309	4.122700	100
	± 1.0	281	3.024800	100	635	8.988300	100
0.100	± 0.1	1080	8.313700	100	—	—	—
	± 1.0	309	3.089400	100	661	9.159700	100
0.200	± 0.1	—	—	—	—	—	—
	± 1.0	—	—	—	—	—	—

表 6 Epsilon-BP 法を用いた場合の各問題に対する最終誤差

ϵ	W	BP			慣性項付き BP		
		Exor	数字	アルファベット	Exor	数字	アルファベット
0.001	± 0.1	0.032538	0.000245	0.000000	0.005962	0.000062	0.000000
	± 1.0	0.013070	0.000000	0.000000	0.006971	0.004528	0.000000
0.005	± 0.1	0.035944	0.000238	0.000005	0.004877	0.000051	0.000001
	± 1.0	0.011731	0.000002	0.000001	0.003271	0.008560	0.000001
0.010	± 0.1	0.035224	0.000275	0.000008	0.007066	0.000065	0.000004
	± 1.0	0.014619	0.000014	0.000003	0.006039	0.011252	0.000003
0.025	± 0.1	0.033017	0.000274	0.000030	0.004603	0.000102	0.000027
	± 1.0	0.014684	0.000062	0.000023	0.006386	0.007109	0.000021
0.050	± 0.1	0.034341	0.000468	0.000127	0.004912	0.000341	0.000111
	± 1.0	0.012298	0.000253	0.000089	0.003996	0.007428	0.000089
0.100	± 0.1	0.039489	0.002048	0.000666	0.011745	0.001724	0.000552
	± 1.0	0.014090	0.001366	0.000535	0.010759	0.006211	0.000432
0.200	± 0.1	0.041477	0.005354	0.002820	0.023085	0.005181	0.002489
	± 1.0	0.026394	0.003920	—	0.020840	0.018547	0.002866

表 7 Exor 問題における各学習法の実験結果

学習方法	学習回数	処理時間 [sec]	収束回数	最終誤差
BP	980	0.015000	78	0.009671
慣性項付き BP	416	0.006742	89	0.004655
Epsilon-BP	405	0.005978	92	0.003271
Silva	1479	0.026667	78	0.009048
Delta	1095	0.020385	26	0.024516

表 8 数字認識問題における各学習法の実験結果

学習方法	学習回数	処理時間 [sec]	収束回数	最終誤差
BP	198	0.178700	100	0.000000
慣性項付き BP	1969	2.091176	85	0.008543
Epsilon-BP	189	0.181000	100	0.000002
Silva	268	0.602500	100	0.000002
Delta	113	0.337400	100	0.000000

表 9 アルファベット認識問題における各学習法の実験結果

学習方法	学習回数	処理時間 [sec]	収束回数	最終誤差
BP	281	2.307500	100	0.000000
慣性項付き BP	655	9.509800	100	0.000000
Epsilon-BP	270	3.067600	100	0.000001
Silva	362	23.375555	99	0.017974
Delta	245	16.757639	72	0.225241

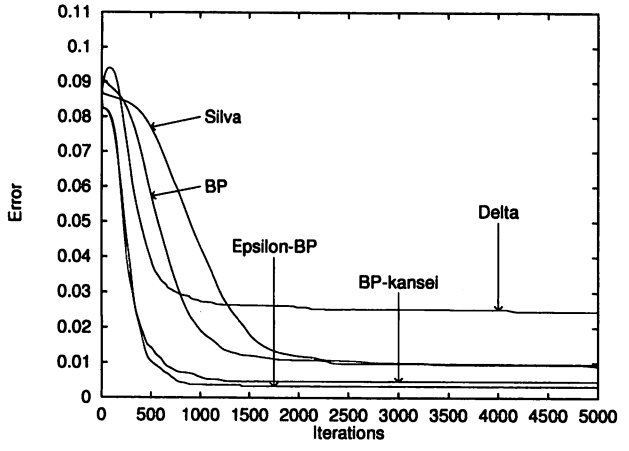

図 4　Exor 問題における各学習法の収束特性

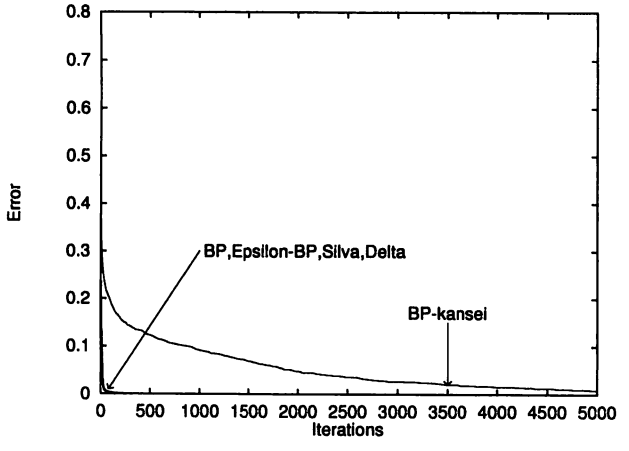

図 5　数字認識問題における各学習法の収束特性

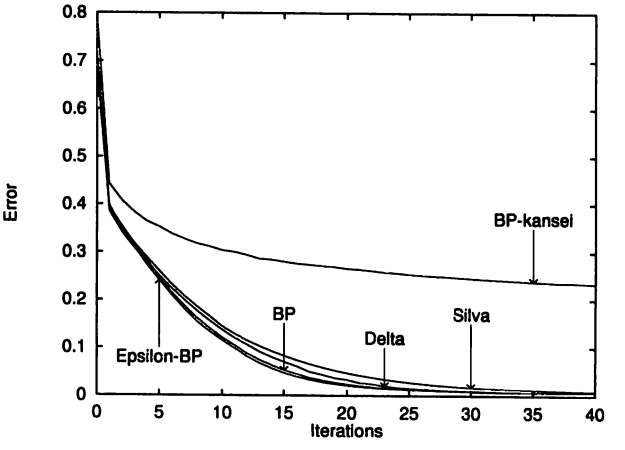

図 6　数字認識問題における各学習法の収束特性

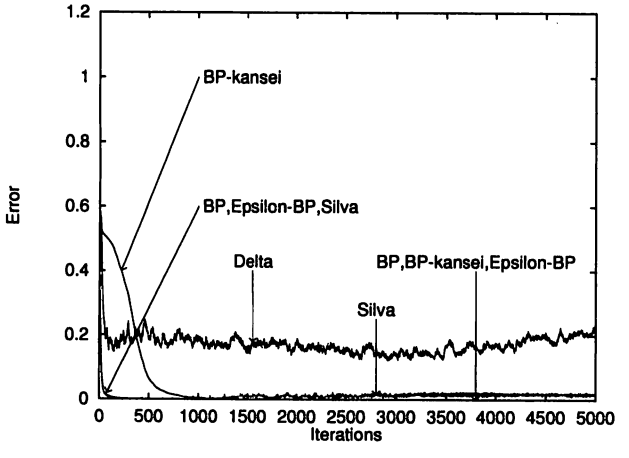

図 7　アルファベット認識問題における
各学習法の収束特性

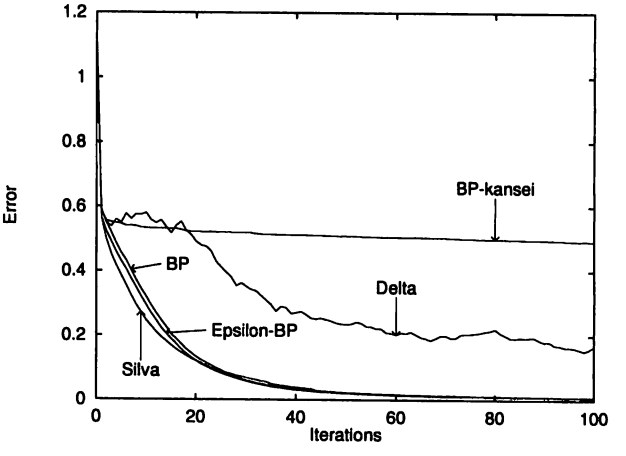

図 8　アルファベット認識問題における
各学習法の収束特性

A study on effectiveness of the Back Propagation Learning Algorithm

Maki HASHIMOTO, Hirotaka INOUE* and Hiroyuki NARIHISA**

Graduate School of Engineering,
**Doctoral Program in System Science,*
Graduate School of Engineering,
***Department of Information & Computer Engineering,*
Faculty of Engineering,
Okayama University of Science,
Ridaicho 1-1, Okayama 700-0005, Japan

(Received November 1, 2000)

Neural networks, which imitate the process of the human brain, have been applied to a wide range of problems including artificial intelligence discrimination, and pattern realization. The back propagation is the most widely used learning algorithm in neural networks, because of its simplicity and performance superiority compared with feed-forward type neural networks. Notwithstanding, the back propagation has some drawbacks in terms of convergent speed when applied to large scale problems. In this paper, we investigate the possibility of an application of Epsilon-BP algorithm and Adaptive step algorithms to pattern recognition from the view points of effectiveness.