

バックプロパゲーション学習における安定性と加速性

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

(1999年11月4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, 安定した高速な収束が期待できる BP 学習法が色々と研究されている.

BP の学習過程における安定性と収束の効率化のため, 最近リャプノフ数を利用した手法が提案されている 1). これはカオス状態にならないように, 収束の安定性を保持しながら学習を実施させようとするものである. 本研究はその有効性につき検討するものである.

2. 固定点としての BP における勾配探索学習

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 $(X_i, D_i) (i = 1, \dots, p)$ をニューラルネット A に与えることで, 変換関数 $F_{W,A}$ が入力空間を覆うように重み W を探索しようとするものである. 学習は目標値との誤差に関するコスト関数

$$C(W) = \frac{1}{p} \sum_{i=1}^p |F_{W,A}(X_i) - D_i|^2 \quad (1)$$

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.

図 1 のように c 個の接続をもつニューロン j の n 回目の重み W_n を

$$W_n = (w_{j_1 i_1}(n) \quad w_{j_2 i_2}(n) \quad \dots \quad w_{j_c i_c}(n))$$

とすると, ニューロン j に対する入力は以下ようになる.

$$a_j(n) = \sum_{i \in A(j)} w_{ji}(n) x_i(n) \quad (2)$$

ここで, $A(j)$ はニューロン j の前の層の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,

$$x_j(n) = f(a_j(n)) \quad (3)$$

$$\text{ただし, } f(a) = \frac{1}{1 + \exp(-a)}$$

である.

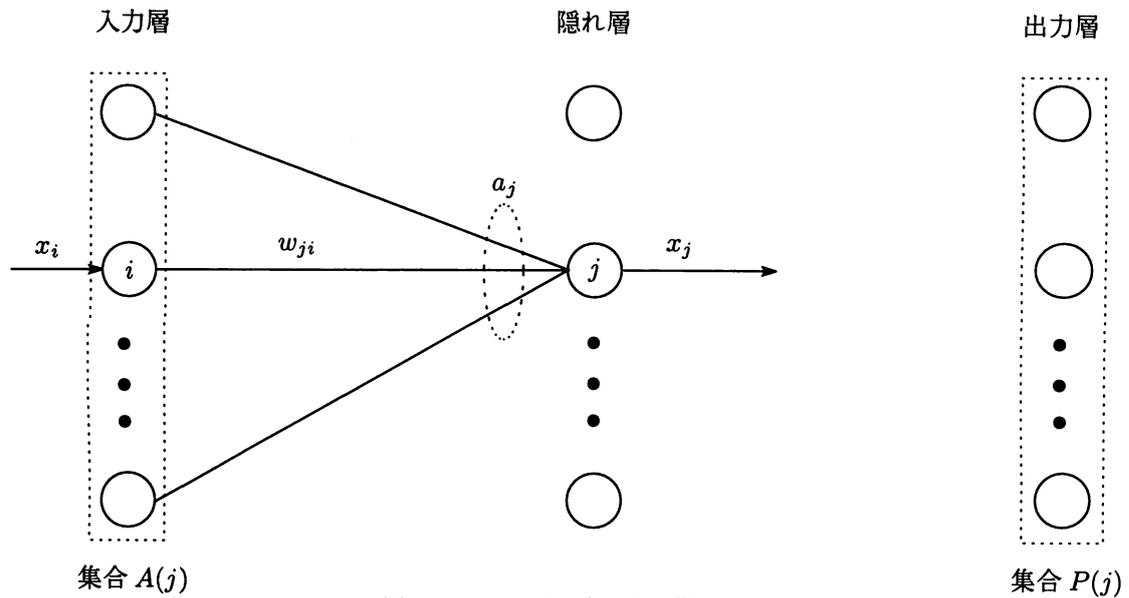


図1 ニューラルネットの構成

結合荷重は、次式によって更新される。

$$w_{ji}(n+1) = w_{ji}(n) + \lambda(n)\Delta_{ji}(W_n) \quad (4)$$

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、 Δ_{ji} は次のようになる。

$$\Delta_{ji} = \frac{\partial x_j(n)}{\partial w_{ji}(n)} K_j(n) \quad (5)$$

$$\text{ニューロン } j \text{ が出力層のとき} \quad K_j(n) = 2(d_j - x_j(n))$$

$$\text{ニューロン } j \text{ が出力層以外のとき} \quad K_j(n) = \sum_{k \in P(j)} \frac{\partial x_k(n)}{\partial x_j(n)} K_k(n)$$

また、式(4)の λ は gradient step を意味しており、 $P(j)$ はニューロン j の後ろの層のニューロンの集合を表している。

実数 R に関する結合荷重の c 次関数を一般的に R^c とすると、式(4)は

$$R_{ji}(W) = w_{ji} + \lambda \Delta_{ji}(W)$$

となる。繰り返しを考慮したBPアルゴリズムの動作は、次式のように書くことができる。

$$w_{ji}(n+1) = R_{ji}(W_n)$$

これは最終的には

$$W_{n+1} = R(W_n) \quad (6)$$

と、上式のように表現できる。このときベクトル関数 R は、 c 個の w_{ji} を持つ修正関数を表しており、 $R_{ji}(W)$ を意味している。

さらに、もし

$$\bar{x} = F(\bar{x})$$

という関係が成り立つならば \bar{x} は、関数 F の固定点である。これは固定点探索が繰り返し法を示しており、繰り返し関数

$$x_{n+1} = F(x_n)$$

が連続する $\{x_n\}$ の \bar{x} への収束を意味している。式 (6) の関係は、BP アルゴリズムが修正関数 $R(W)$ によって、多層 NN の結合荷重ベクトル \bar{W} を固定点として探索することと同値である。

3. リャプノフ数

NN における BP 学習はネットワークの接続重み W を逐次修正しながら最適な重みに収束させるものである。この収束の状態を示す尺度としてリャプノフ数が提案されている。一般にリャプノフ数が負であれば安定な収束が期待できるものと考えられている。

BP 学習の過程においてネットワークの重み W は繰り返し最適化の結果として $W_0 \rightarrow W_1 \rightarrow \dots \rightarrow W_n \rightarrow \tilde{W}$ (固定点) になるものとする。そこで、 $R(W) = \tilde{W}$ となるようにすると、

$$W_{n+1} = R(W_n) \quad (7)$$

$$R_{ji}(W_n) = w_{ji}(n) + \lambda \Delta_{ji}(W_n) \quad (8)$$

$$\frac{\partial R_{ji}(W)}{\partial w_{ji}} = 1 + \lambda \frac{\partial \Delta_{ji}(W)}{\partial w_{ji}} \quad (9)$$

となる。この偏微分差分方程式において、ニューロン j が c 個の接続ユニットを持つ場合のリャプノフ数 γ を以下のように定義する。

$$\gamma = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{p=0}^{n-1} \ln \left| 1 + \lambda \sum_{k=1}^c \frac{\partial \Delta_{jk^i k}(W_p)}{\partial w_{jk^i k}} \right| = \lim_{n \rightarrow \infty} \gamma_n \quad (10)$$

式 (5), 式 (9) より

$$\begin{aligned} \frac{\partial R_{ji}(W)}{\partial w_{ji}} &= 1 + \lambda \frac{\partial \Delta_{ji}(W)}{\partial w_{ji}} \\ &= 1 + \lambda \left[\frac{\partial^2 x_j}{\partial w_{ji}^2} K_j + \left(\frac{\partial x_j}{\partial w_{ji}} \right)^2 \frac{\partial K_j}{\partial x_j} \right] \\ &= 1 + \lambda L_{ji} | W \end{aligned}$$

ただし、

$$\text{ニューロン } j \text{ が出力層のとき} \quad L_{ji} = 2 \left[\frac{\partial^2 x_j}{\partial w_{ji}^2} (d_j - x_j) - \left(\frac{\partial x_j}{\partial w_{ji}} \right)^2 \right]$$

$$\text{ニューロン } j \text{ が出力層以外のとき} \quad L_{ji} = \frac{\partial^2 x_j}{\partial w_{ji}^2} K_j + \left(\frac{1}{x_j} \frac{\partial x_j}{\partial w_{ji}} \right)^2 \sum_{k \in P(j)} (w_{kj})^2 L_{kj}$$

ここで γ を負の定数 ν として

$$\nu = \frac{1}{n} \sum_{k=0}^{n-1} \ln \left| 1 + \lambda_k \sum_{(i,j) \in C} L_{ji} | W_k \right|$$

$$\exp(\nu) = \left| 1 + \lambda_k \sum_{(i,j) \in C} L_{ji} |W_k| \right|$$

としている。

4. リャプノフ条件付 BP 学習アルゴリズム

BP 学習は与えられた NN の重み系列 $\{W_i\}$ を収束させる過程であると仮定する。この仮定により、安定収束のため、重みに対応するリャプノフ数を常に負にする条件を付加した BP 学習アルゴリズムを、L-BP 学習アルゴリズムと呼ぶことにする。

与えられた NN のニューロン i とニューロン j との接続子を (i,j) で表し、それらの集合を C とすると、 k 回の繰返しにおける重み W_k に対応するリャプノフ条件は

$$\nu = \frac{1}{n} \sum_{k=0}^{n-1} \ln \left| 1 + \lambda_k \sum_{(i,j) \in C} L_{ji} |W_k| \right| \quad (11)$$

として表わされる。ただし ν は負の実数値とする。これはすべての n について

$$\left| 1 + \lambda_n \sum_{(i,j) \in C} L_{ji} |W_n| \right| = \exp(\nu) \quad (12)$$

と考えられるので、 $\lambda_n > 0$ とすると、

$$\sum_{(i,j) \in C} L_{ji} |W_n| < 0$$

でなければならない。

式(12)より

$$\lambda_n = -(1 \pm \exp(\nu)) \left(\sum_{(i,j) \in C} L_{ji} |W_n| \right)^{-1}$$

となるが、 $\lambda_n > 0$ とするものをとると

$$\lambda_n = -(1 - \exp(\nu)) \left(\sum_{(i,j) \in C} L_{ji} |W_n| \right)^{-1} \quad (13)$$

となる。

以上のことを考慮したアルゴリズムを以下に示す。

L-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。

$$K_j(n) = 2(d_j - x_j(n))$$

step3. 結合荷重の修正量の計算

- gradient step λ_n の計算

$$\lambda_n = -\frac{1 - \exp(\nu)}{B}$$

$$B = \sup_{s \in \Sigma} \left(\sum_{(i,j) \in \Gamma_s} L_{ji} \mid W_n \right)$$

ニューロン j が出力層のとき $L_{ji} = 2 \left[\frac{\partial^2 x_j}{\partial w_{ji}^2} (d_j - x_j) - \left(\frac{\partial x_j}{\partial w_{ji}} \right)^2 \right]$

ニューロン j が出力層以外のとき $L_{ji} = \frac{\partial^2 x_j}{\partial w_{ji}^2} K_j + \left(\frac{1}{x_j} \frac{\partial x_j}{\partial w_{ji}} \right)^2 \sum_{k \in P(j)} (w_{kj})^2 L_{kj}$

- 結合荷重の修正方向 $\Delta_{ji}(W_n)$ の計算

$$\Delta_{ji} = \frac{\partial x_j(n)}{\partial w_{ji}(n)} K_j(n)$$

ニューロン j が出力層のとき $K_j(n) = 2(d_j - x_j(n))$

ニューロン j が出力層以外のとき $K_j(n) = \sum_{k \in P(j)} \frac{\partial x_k(n)}{\partial x_j(n)} K_k(n)$

step4. 結合荷重の修正

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

step5. 収束判定

一定の収束条件を満たすと終了し、そうでなければ step1. に戻り処理を繰り返す。

L-BP アルゴリズムを用いるとき、weight の初期化や gradient step の選択に関係しない。くり返し法において、一定で負のリアプノフ数である gradient step は BP アルゴリズムに関して安定した振る舞いを課す。

5. 実験内容

初期の重みをランダムに与えて、以下の問題を一般的な BP 学習と L-BP 学習で実施し、比較検討を行った。

(a) Exor 問題

Exor 問題の状態表を表 1 に示す。この問題を、

入力層のユニット数 (x_1, x_2) : 2 個 隠れ層のユニット数 : 2 個 出力層のユニット数 (x_j) : 1 個
以上の NN 構造を用いて L-BP 学習では、 ν の値を $\nu = -1, -2, -3$ と変化させ実験を行なった。

表 1 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

(b) 簡単な問題

簡単な問題 1) の入力と出力目標データの関係を表 2 に示す。この問題を、
 入力層のユニット数 (x_1, x_2): 2 個 隠れ層のユニット数: 2 個 出力層のユニット数 (x_j): 1 個
 以上の NN 構造を用いて L-BP 学習では、 ν の値を $\nu = -1, -2, -3$ と変化させ実験を行なった。

表 2 簡単な問題の状態表

x_1	-1.5				-0.5				0.5				1.5			
x_2	-1.5	-0.5	0.5	1.5	-1.5	-0.5	0.5	1.5	-1.5	-0.5	0.5	1.5	-1.5	-0.5	0.5	1.5
x_j	1	1	1	0	1	1	1	0	1	1	0	0	1	1	0	0

(c) 数字の認識問題

数字の認識問題の入力例を図 2 に、また出力目標データ例を図 3 に示す。この問題を、
 入力層のユニット数: 49 個 隠れ層のユニット数: 49 個 出力層のユニット数: 10 個
 以上の NN 構造を用いて L-BP 学習では、 ν の値を色々変化させ実験を行なった。

“0” の入力データ							“1” の入力データ							“2” の入力データ							
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0
0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	
0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	
0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	
0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	

図 2 数字の学習の入力例

“0” の教師データ										“1” の教師データ										“2” の教師データ																																																																																																
t[0]	=	1	t[0]	=	0	t[0]	=	0	t[1]	=	0	t[1]	=	1	t[1]	=	0	t[1]	=	0	t[2]	=	0	t[2]	=	0	t[2]	=	1	t[2]	=	0	t[3]	=	0	t[4]	=	0	t[5]	=	0	t[6]	=	0	t[7]	=	0	t[8]	=	0	t[9]	=	0																																																															

図 3 数字の学習の出力目標データ例

6. 結果と考察

ここで示されている実験結果は、BP 学習は各問題とも 100 回実行させた時の平均を表しており、L-BP 学習では Exor 問題の場合が実行回数 20 回の平均で、簡単な問題に対しては実行回数 25 回の平均を表している。また、図 4 ~ 図 15 の誤差 Error [db] は以下の式によって求めた値である。

$$20 \log_{10} \frac{E_i}{E_o}$$

E_o : 基準となる誤差の値 (本研究では $E_o = 1$ とする)

E_i : 各学習回数における誤差の値

表3 各問題に対する実験結果

学習方法	Exor 問題			簡単な問題		
	学習時間 [sec]	最終誤差	学習回数	学習時間 [sec]	最終誤差	学習回数
BP	0.0645	0.007903	5000	0.1374	0.000039	2580
L-BP $\nu = -1$	0.045	0.0000	239	0.0144	0.0000	183
(Weight の $\nu = -2$ 初期化あり)	0.003	0.0000	176	0.0084	0.0000	139
$\nu = -3$	0.0000	0.0000	100	0.0096	0.0000	132
L-BP $\nu = -1$	0.0075	0.0000	1623	0.0156	0.0000	193
(Weight の $\nu = -2$ 初期化なし)	0.009	0.0000	2642	0.0104	0.0000	137
$\nu = -3$	0.0115	0.0000	1994	0.008	0.0000	119

まず、図4～図15からわかるように、Exor 問題、簡単な問題ではBP法よりL-BP法の方が学習精度が良いと言える。次に、表3より学習時間はL-BP法の方が速いことがわかった。また、学習回数（ここでは0.0001より誤差が小さくなった時の学習回数を示している。）もBPより良い結果が得られた。

しかし、L-BP学習法は収束したりしなかったりとBP法より確実に学習することができなかった。これは初期の結合荷重をランダムに与えているので、その影響を受けているためと考えられる。したがって、最初の重みに処理を加えて（Weightの初期化）学習させてみた。（図4、図6、図8、図10、図12、図14）これより、Exor問題に対してはWeightの初期化によって実験結果に影響が出ているが、簡単な問題ではほとんど変化が見られなかった。

また、数字の認識問題については、L-BP学習法では初期の結合荷重と ν の値を色々変えたり、Weightの初期化を付加させたりした結果、途中まで学習するケースもあったがきちんと学習することができなかった。

7. おわりに

L-BP学習法はうまく収束した場合、BP法より良い結果が得られているのに対して、収束しないケースが見られる。これは初期の重みによってgradient step λ がうまく収束する値にならないからだと考えられる。また、Weightの初期化によって各問題の結果に影響を受けたり受けなかったりするの、学習する問題の規模によるものではないかと考えられる。つまり、問題が大き過ぎると（数字の認識問題）きちんと学習できなくなり、小さいと（Exor問題）初期の結合荷重に左右されやすくなるのではないと思われる。

これらのことより、L-BP学習法はきちんと収束した場合、BP法よりも精度、計算時間、学習回数の結果有効であると言える。しかし、必ず収束するという学習の安定性の面ではBP法より良いとは考えられない。

参考文献

- 1) Bertrand AUGEREAU, Thierry SIMON, Jacky BERNARD, Bernard HEIT: The BP- λ L1 Algorithm: Non-Chaotic and Accelerated Learning in a MLP Network, in From Natural to Artificial Neural Computation, ed. Jose' Mira, Francisco Sandoval, pp.231-238, Springer-Verlag, Berlin (1995).
- 2) 奈良重俊, ピーター・ディビス: カオスと知的処理, (株) ジャストシステム (1994).
- 3) Joseph La Salle, Solomon Lefschetz, 山本稔 訳: リャプノフの方法による安定性理論, 産業図書 (株) (1977).
- 4) 坂本實 訳: 常微分方程式モデル入門, 森北出版 (株) (1996).

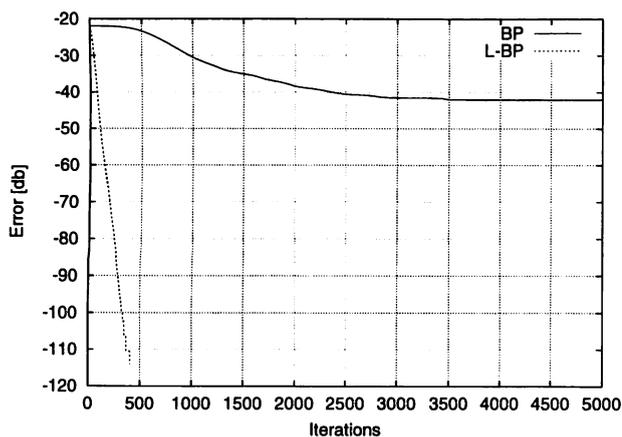


図4 Exor 問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -1$)

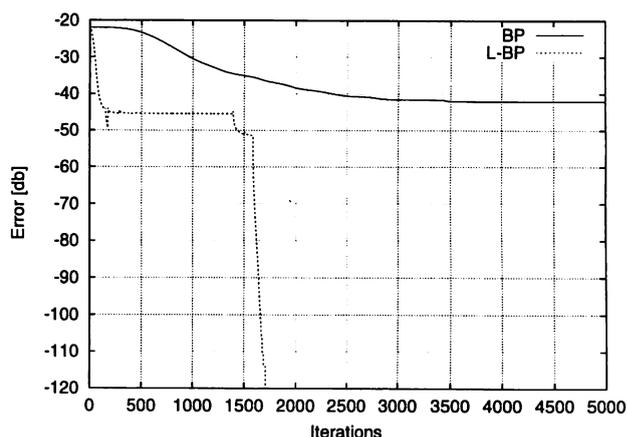


図5 Exor 問題における BP と L-BP の収束特性 ($\nu = -1$)

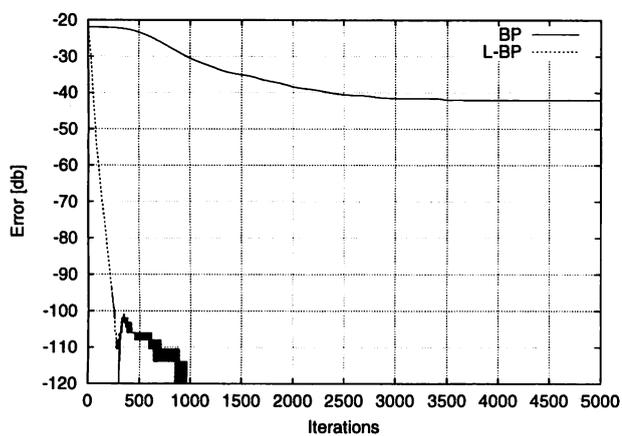


図6 Exor 問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -2$)

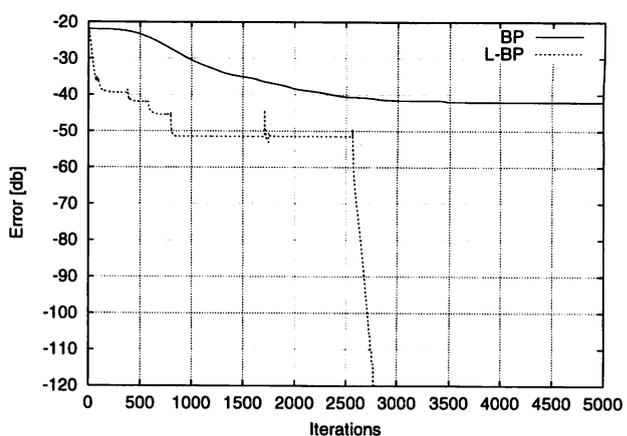


図7 Exor 問題における BP と L-BP の収束特性 ($\nu = -2$)

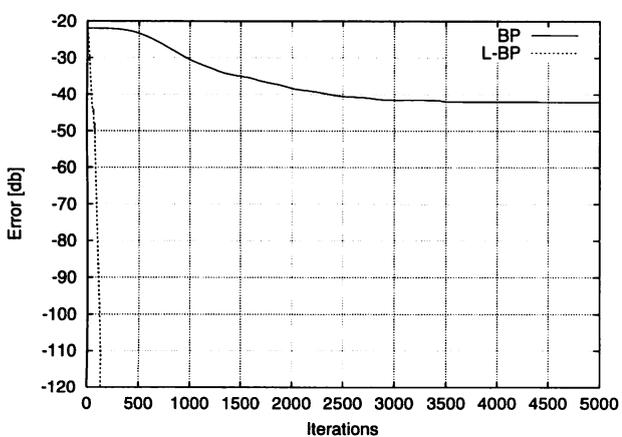


図8 Exor 問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -3$)

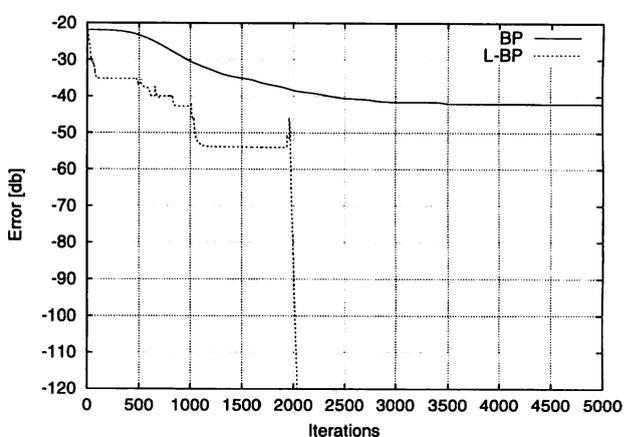


図9 Exor 問題における BP と L-BP の収束特性 ($\nu = -3$)

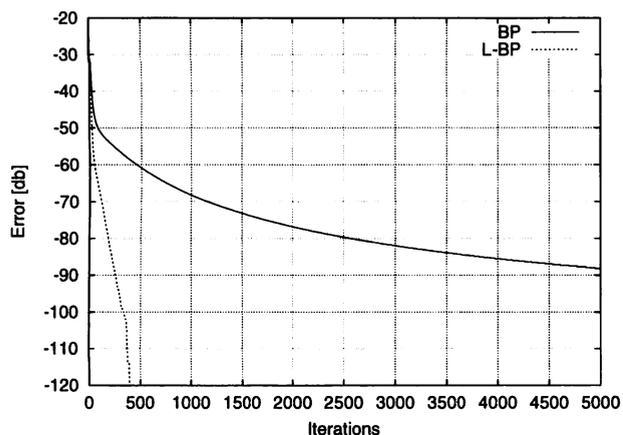


図 10 簡単な問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -1$)

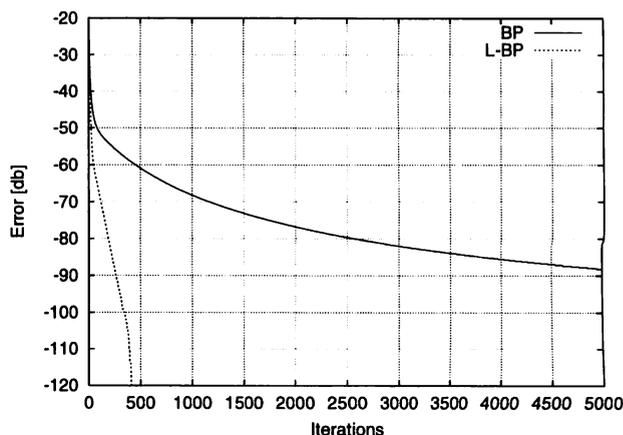


図 11 簡単な問題における BP と L-BP の収束特性 ($\nu = -1$)

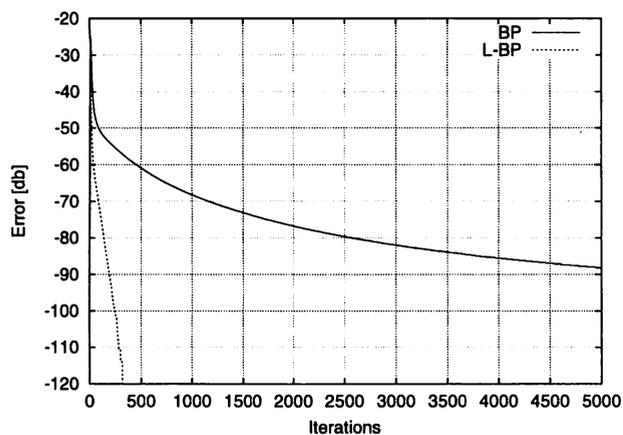


図 12 簡単な問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -2$)

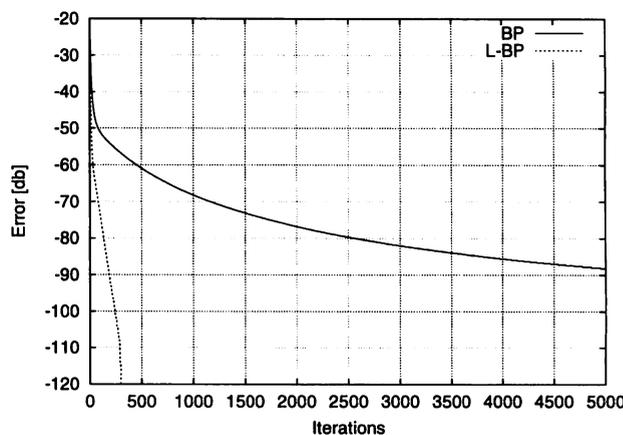


図 13 簡単な問題における BP と L-BP の収束特性 ($\nu = -2$)

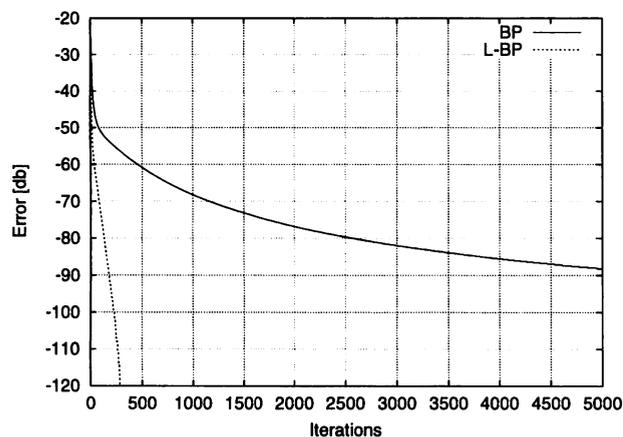


図 14 簡単な問題における BP と Weight の初期化を付加した L-BP の収束特性 ($\nu = -3$)

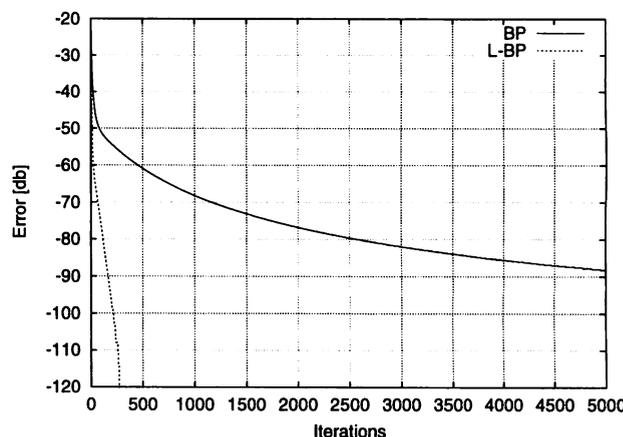


図 15 簡単な問題における BP と L-BP の収束特性 ($\nu = -3$)

Stability and Acceleration in 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 4, 1999)

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 stability and convergent speed when applied to large scale problems. In order to improve these network, many approaches have been proposed by different researchers. Recently, the lyapunov number algorithm was proposed. This algorithm aims at stable convergence by positively using chaotic conditions during the convergence process. In this paper, we investigate the possibility of an application of the lyapunov number algorithm to pattern recognition from the view points of stability and convergence speed.