岡山理科大学紀要 第35号 A pp. 153-160(1999)

ニューラルネットにおける重み初期化の有効性について

土屋　秀樹・井上　浩孝*・成久　洋之**

岡山理科大学工学研究科修士課程情報工学専攻

* 岡山理科大学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　まえがき

ニューラルネット (以下 NN と記す) アルゴリズムの一つであるバックプロパゲーション (BP) 学習則はロバスト性を持つ優れた手法であるが，その問題点として学習に多大の時間を要する事が挙げられる．BP 学習則の高速化の方法にアダプティブな手法や 2 次微分を用いるなど諸種の方法が提案されているが，重みの初期化の問題も BP 学習に重大な影響を与えているものと考えられる．本論文は，ニューラルネットの初期化における効果的な結合荷重の使用が学習速度，収束性に与える影響について検討するものである．

2　線形回帰

学習パターン数を n，出力層の出力値 (従属変数) を Z_k，中間層の出力値 (独立変数) を $Y_j(j=1,2,\cdots,m)$ とすると出力層の出力値の予測値 $\hat{Y}$ は，

$$\hat{Z} = w_0 + w_1 Y_1 + w_2 Y_2 + \cdots + w_m Y_m$$

により求められる．ここでの $\hat{Z}$ は教師信号を表しており，実際には実測値 Z との間に誤差

$$e_i = Z_i - \hat{Z}_i$$

が発生する．この誤差は正負の符合を持つのでその 2 乗和 P が最小になるように独立変数にかける重み w_i(偏回帰係数) および定数項 w_0(バイアスウェイト) を決定する．この手法を最小二乗法と呼び，得られる係数を最小二乗推定値と呼ぶ．

$$\begin{aligned} P &= \sum_{i=1}^{n} e_i^2 \\ &= \sum_{i=1}^{n} (Z_i - \hat{Z}_i)^2 \\ &= \sum_{i=1}^{n} Z_i - (w_0 + w_1 Y_{i1} + w_2 Y_{i2} + \cdots + w_m Y_{im})^2 \end{aligned}$$

P を $w_0, w_1, w_2, \cdots, w_m$ で偏微分して 0 とおく．

$$\left\{ \begin{array}{lcl} \frac{\partial P}{\partial w_0} = -2\sum_{i=1}^{n} \{Z_i - (w_0 + w_1 Y_{i1} + w_2 Y_{i2} + \cdots + w_m Y_{im})\} &=& 0 \\ \frac{\partial P}{\partial w_1} = -2\sum_{i=1}^{n} Y_{i1} \{Z_i - (w_0 + w_1 Y_{i1} + w_2 Y_{i2} + \cdots + w_m Y_{im})\} &=& 0 \\ \frac{\partial P}{\partial w_2} = -2\sum_{i=1}^{n} Y_{i2} \{Z_i - (w_0 + w_1 Y_{i1} + w_2 Y_{i2} + \cdots + w_m Y_{im})\} &=& 0 \\ \vdots \\ \frac{\partial P}{\partial w_m} = -2\sum_{i=1}^{n} Y_{im} \{Z_i - (w_0 + w_1 Y_{i1} + w_2 Y_{i2} + \cdots + w_m Y_{im})\} &=& 0 \end{array} \right.$$

これらの偏微分した式に変数 Z，$Y_1, Y_2, \cdots, Y_m$ の平均値を $\bar{Z}, \bar{Y_1}, \bar{Y_2}, \cdots, \bar{Y_m}$ とした時の関係式

$$w_0 = \bar{Z} - w_1 \bar{Y_1} - w_2 \bar{Y_2} - \cdots - w_m \bar{Y_m}$$

及び，独立変数 Y_i, Y_j 間の変動・共変動

$$S_{ij} = \sum_{k=1}^{n} (Y_{ki} - \bar{Y}i)(Y_{kj} - \bar{Y}j)$$

及び，独立変数 Y_i と従属変数 Z の共変動

$$S_{iz} = \sum_{k=1}^{n}(Y_{ki} - \bar{Y}_i)(Z_k - \bar{Z})$$

を代入して整理すると，

$$\begin{cases} w_1 S_{11} + w_2 S_{12} \cdots + w_m S_{1m} = S_{1z} \\ w_1 S_{21} + w_2 S_{22} \cdots + w_m S_{2m} = S_{2z} \\ \qquad\qquad\qquad \vdots \\ w_1 S_{n1} + w_2 S_{n2} \cdots + w_m S_{nm} = S_{nz} \end{cases}$$

が得られる．

この連立方程式 (正規方程式) を解く事により偏回帰係数，つまり初期状態における重みが決定される．

3 Maximum Covariance Method

本研究では，初期化に用いる結合荷重を決定する為に Maximum Covariance Method(MCM) [1] [2] を用いた．MCM は，候補となる中間層ユニットの出力と教師信号と出力層の出力との誤差 (エラー) の共分散の絶対値が最大のものを中間層ユニットとして選択し，そのユニットに繋がる結合荷重を線形回帰によって求める方法である．その手順は以下の通りである．

step1 適当なモデル選択手法を使って学習に要求される中間層の数 q を決定する．

step2 M 個の候補となる中間層ユニット ($M \gg q$) を準備し，入力層と中間層の間の結合荷重 $v_{i,j}$ を $[-4;4]$ でランダムに初期化する．但し、$M = 10q$ とする．

step3 ネットワークの出力が要求される出力シーケンスの平均になるようにバイアスウェイト $w_{0,k}$ を設定する．但し，この時点で候補となる中間層と出力層の間は未接続であり，与えられる結合荷重は $w_{0,k}$ のみである．

step4 候補となる中間層ユニットの出力とエラーの共分散の絶対値 C_j を求める．

$$C_j = \frac{1}{n}\sum_{k=1}^{r}\left|\sum_{e=1}^{n}(y_{j,e} - \overline{y}_j)(\varepsilon_{k,e} - \overline{\varepsilon}_k)\right|,\ j = 1,\ldots,Q$$

ここで $y_{j,e}$，$\varepsilon_{k,e}$ はそれぞれ e 番目の学習パターンに対する j 番目の中間層ユニットの出力及び k 番目の出力層ユニットのエラー，$\overline{y}_j$ は j 番目の中間層ユニットの出力の平均，$\overline{\varepsilon}_k$ は k 番目の出力層ユニットのエラーの平均を表し，r，n はそれぞれ出力層ユニットの数，学習パターンの数を表す．

step5 共分散 C_j の最大値を見つけ，それに対応する中間層と出力層を接続し，$M = M - 1$ とする．

step6 現時点で存在する中間層と出力層の間の結合荷重 $w_{j,k}$ を線形回帰によって与える．これらの重みの数は新しい候補ユニットが出力層と接続される毎に増えていくので，最適化によって出力層のエラーは毎回変化する．

step7 もし，q 個の候補ユニットと出力層が接続されたら初期化を終了する．そうでなければ、残っている候補ユニットに対して **step3-5** を繰り返す．

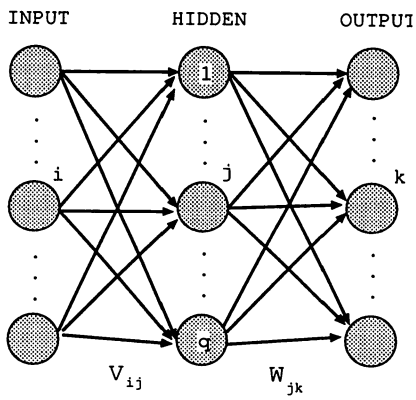

図 1 ニューラルネットの構造

4 学習アルゴリズム

本研究では，ニューラルネットワークの学習アルゴリズムの中でも代表的なバックプロパゲーション学習則を用い，入出力関数としてシグモイド関数

$$f(x) = \frac{1}{1 + exp(-x)}$$

を用いた．

4.1 逐次修正法と一括修正法

バックプロパゲーション学習則の修正法には以下の方法がある．

逐次修正法 ある入力パターンを NN に提示し，それに対する出力結果を得る．次に，出力結果と目標値である教師信号との誤差を逆伝播しながら，各層の各ユニットにおける誤差を求める．さらに，その誤差をもとに結合荷重の修正量を計算し，結合荷重の修正を行なう．以上の処理を全パターンに対して行ない，更に何度も同じ事を繰り返す．

一括修正法 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を得る．次に，その出力結果と教師信号の誤差を逆伝播し，結合荷重の修正量を求め蓄積する．以上の事を全入力パターンに対して行ない，全入力パターンを提示した後に，それまでに蓄積された結合荷重の修正量をもとに結合荷重の修正を行なう．これらの処理を誤差が小さくなるまで繰返し行なう．

本研究では，後者の一括修正法を用いた．

5 実験内容

本研究では以下の 4 つの問題を取り扱った．

- 4 × 4 チェスボード問題
  - $m \times m$ チェス問題は排他的論理和 (XOR) 問題を一般化したものである．2 入力 X, Y があり，○ は 0，× は 1 を表す．学習データ数は 16.
- 2 スパイラル問題
  - 2 入力 X, Y があり，その入力パターンの内，半分は 1 を出力し，残りは 0 を出力する．○ は 0，× は 1 を表す．学習データ数は 194.
- アルファベット 26 文字 (サイズ 10 × 10)
  - 文字のサイズは 10 × 10 で入力ビット数は 100 である．学習データ数は 26.
- アルファベット 52 文字 (大文字，小文字) ＋数字 10 文字 (サイズ 10 × 10)
  - 文字のサイズは 10 × 10 で入力ビット数は 100 である．学習データ数は 62.

便宜上，これらの問題をそれぞれ問題 1，問題 2，問題 3，問題 4 と呼ぶ事にする．ここで，学習回数は全ての学習で 5000 回とした．また，各学習におけるパラメータは表 1 に示す．

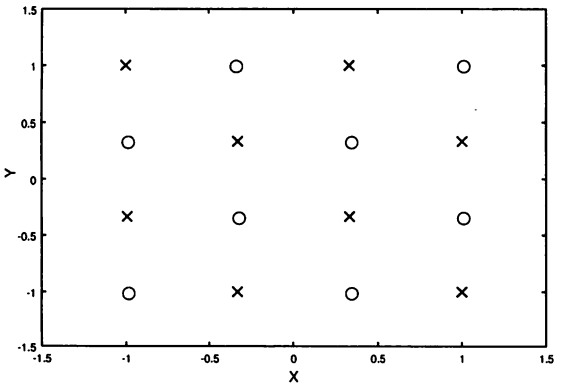

図 2 4 × 4 チェス問題

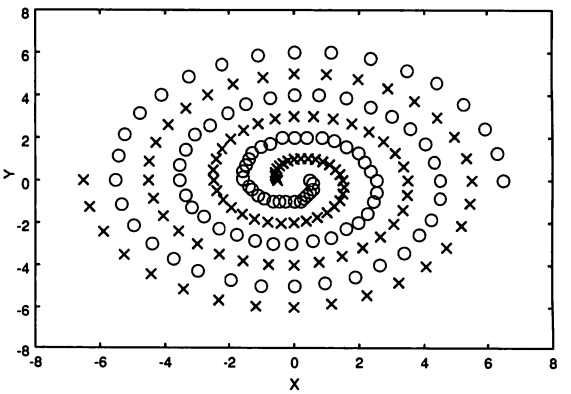

図 3 2 スパイラル問題

表 1 学習パラメータ

問題	学習係数	シグモイド関数の傾き	慣性項	中間層の数
問題 1	0.01	1.0	0.5	6
問題 2	0.01	1.0	0.5	50
問題 3	0.01	1.0	0.5	100
問題 4	0.01	1.0	0.5	200

6 実験結果及び考察

以下に BP 及び BPM(BP using MCM) の処理結果を示す．これらのグラフは MCM による処理時間を考慮し，BPM の学習開始点を遅らせてプロットしている．この結果より，いずれの学習においても収束率及び学習速度において BPM が優れていると言える (図 4，図 5，図 6，図 7)．しかし，問題 1，問題 2，問題 3，問題 4 と問題のサイズが拡大するにつれ，MCM による初期化の時間が増大する事がわかる (表 2，表 3，表 4，表 5)．これにより学習開始に遅れが生じ，結果的に BPM の学習速度を遅らせる原因となっている．

しかし，一般に学習データのサイズが大きくなるにつれ学習に要する繰返し回数も増加する傾向にある．本研究では学習回数を 5000 回に固定しているが，実際の学習にはもっと多くの学習回数が必要とされる．従って，学習回数に対する初期化の時間の割合が減少し，学習開始の遅れの問題も多少改善されると思われる．

また，中間層の数を変化させ，その時の学習についても検討した．BPM の中間層の数が BP に比べて少ない状態で学習を行なった場合，BPM で BP よりも良い結果が得られる例があった．これは，MCM による初期化によるものと考えられるが，中間層の数がそれぞれ事なる為，同じ学習回数で収束率の良い悪いを判断する事はできない．しかし，少ない数の中間層で得られた収束率が要求される条件を満たしているものであれば，それは良い収束率を持っていると言える．また，中間層の数が減少しただけ初期化に要する時間も減少する為，結果として BPM の学習速度の向上につながっている．

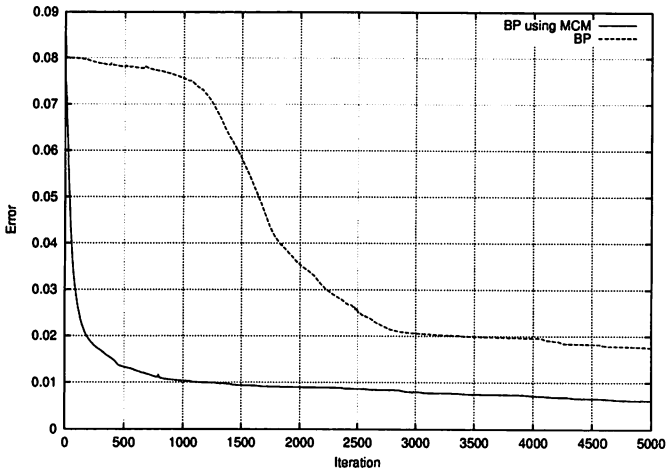

図 4 4 × 4 チェス問題

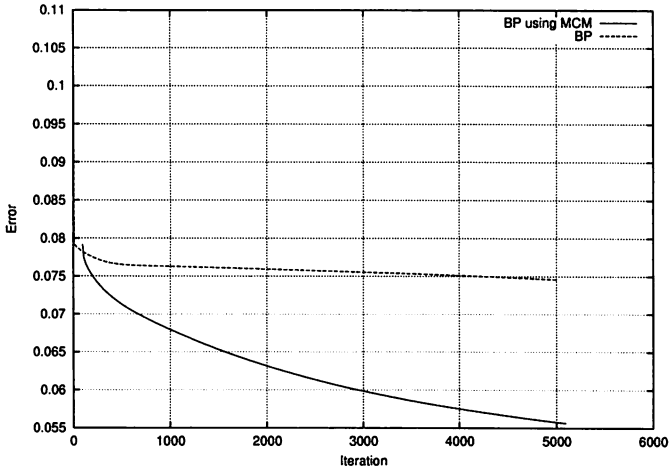

図 5　2 スパイラル問題

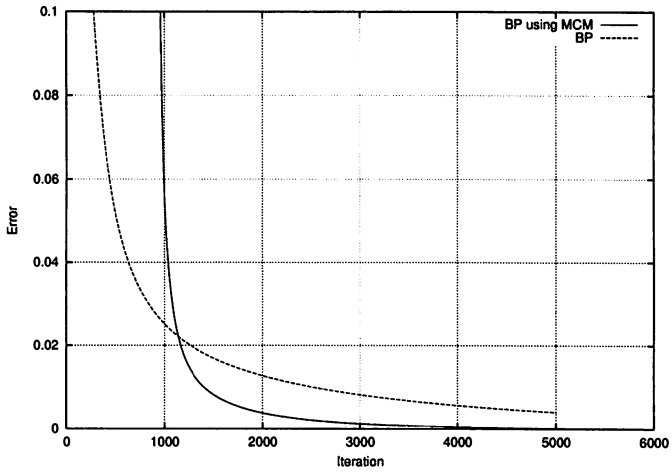

図 6　アルファベット 26 文字

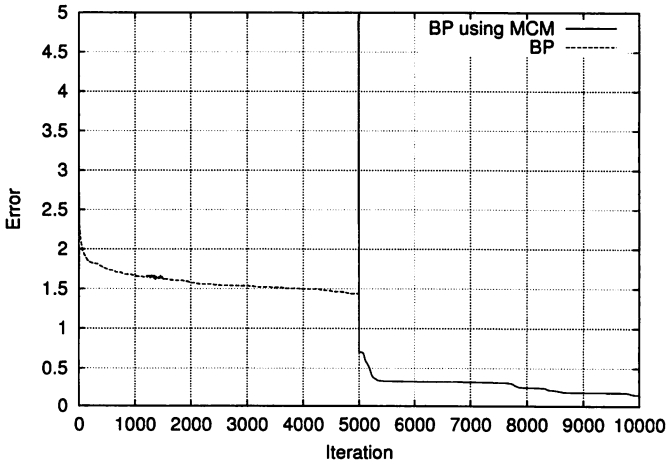

図 7　アルファベット 52 文字＋数字 10 文字

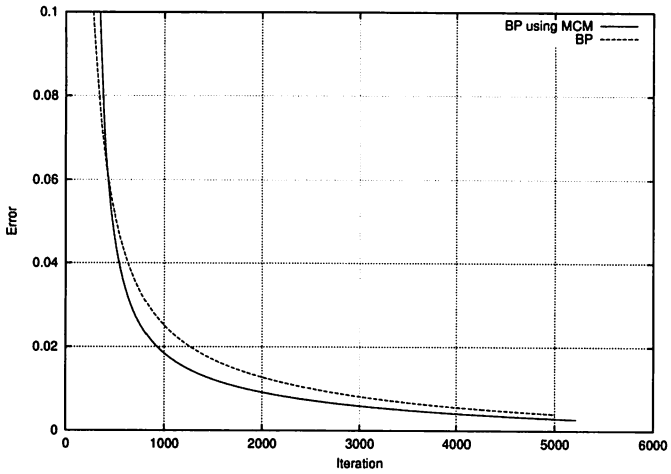

図 8 アルファベット 26 文字

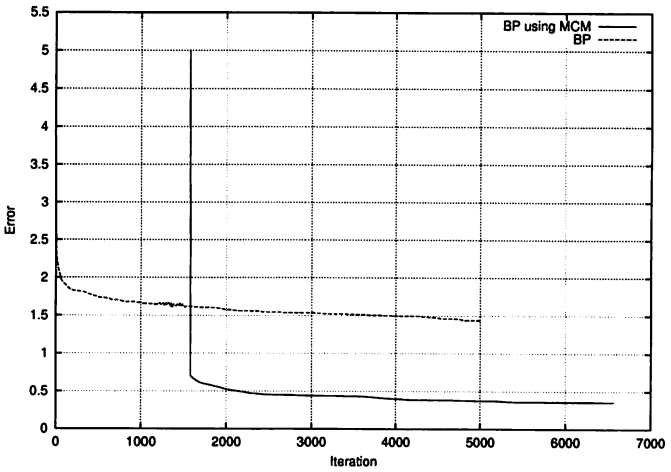

図 9 アルファベット 52 文字＋数字 26 文字

表 2 学習時間 – チェス問題

	BPM			BP	
q	初期化	全処理時間	誤差	全処理時間	誤差
3	0	0.28	0.044645	0.23	0.057922
6	0	0.46	0.006141	0.45	0.018170
12	0.01	0.88	0.000133	0.84	0.001247
30	0.02	2.01	0.009213	1.96	0.004478

表 3 学習時間 – 2 スパイラル問題

	BPM			BP	
q	初期化	全処理時間	誤差	全処理時間	誤差
30	0.26	22.66	0.057485	23.84	0.074580
50	0.93	38.18	0.054124	38.91	0.074733
70	2.2	53.58	0.053220	54.29	0.075086
100	5.64	80.84	0.052787	76.74	0.075495

表 4 学習時間 – アルファベット 26 文字

	BPM			BP	
q	初期化	全処理時間	誤差	全処理時間	誤差
20	0.28	25.62	0.014393	25.12	0.023802
50	2.4	57.2	0.002714	53.67	0.001670
100	19.27	123.7	0.000680	103.0	0.003990
200	171.4	449.7	0.000001	235.2	0.001651

表 5 学習時間 – アルファベット 52 文字+数字 10 文字

	BPM			BP	
q	初期化	全処理時間	誤差	全処理時間	誤差
30	3.99	115.5	0.046816	109.05	0.051883
50	18.0	197.5	0.014266	169.7	0.030721
100	121.6	509.5	0.357803	336.7	0.181740
200	952	1906	0.145685	853.8	1.439263

7 おわりに

本論文は，ニューラルネットの初期化における効果的な結合荷重の使用が学習速度，収束性に与える影響について検討し，MCM によって初期化を行なった BPM は学習速度，収束性両面において非常に効果的な手法である事を示した．

参考文献

[1] M. Lehtokangas, J. Saarinen, P. Salmela, K. Kaski: Weight Initialization Techniques.*Algorithms and Architectures Volume 1 in the NEURAL NETWORK SYSTEMS TECHNIQUES AND APPLICATIONS series Edited by Cornelius T.Leondes*, pp.87–121, ACADEMIC PRESS, 1998.

[2] M. Lehtokangas, P. Korpisaari, and K. Kaski: Maximum Covariance Method for Weight Initialization of Multilayer Perceptron Networks.*Proceedings of the European Symposium on Artificial Neural Networks, ESANN'96*, pp.243–248, 1996.

[3] S. Chen, P. Grant, and C. Cowan. Orthogonal least-squares algorithm for training multioutput radial basis function networks, *IEE Proc.F* 139:378–384, 1992.

[4] Vladimir Cherkassky, Robert Shepherd: Regularization Effect of Weight Initialization in Back Propagation Networks, *IJCNN'98 Proceedings Volume 3*, pp.2258–2261.

Effect of Weight Initialization in Neural Networks

Hideki TSUCHIYA, Hirotaka INOUE and Hiroyuki NARIHISA*
Graduate School of Engineering
**Department of Information & Computer Engineering*
Faculty of Engineering
Okayama University of Science
Ridai-cho 1-1, Okayama 700-0005,Japan
(Received November 4, 1999)

Artificial neural network has been generally used as an artificial intelligence too in various scientific and engineering fields. Especially,back propagation learning algorithm is widely used in learning , recognitions. However, it has only one drawback; that is to say, it is time–consuming in learning process. In order to conquer this drawback of time–consuming in calculation, a lot of improvement have been proposed.

In this paper, we investigate the Effect of Weight Initialization in Neural Networks.