ニューラルネットの学習アルゴリズムに関する一考察

—慣性項を用いた BP 法に対する階層構造確率オートマトンの適用—

馬場 則夫*・佐藤 剣*

A Consideration on the Learning Algorithm of Neural Network —Utilization of Hierarchical Structure Stochastic Automata for the Back Propagation Method with Momentum—

Norio BABA* and Ken SATO*

Back propagation method with momentum has often been applied to adapt artificial neural networks for various pattern classification problems. However, an important limitation of this method is that its learning performance depends heavily upon the selection of the values of momentum factor and step size. In this paper, it is shown that hierarchical structure stochastic automata are quite helpful for finding appropriate values of the parameters involved in the BP method with momentum.

Key Words: neural network, BP method with momentum, momentum factor, step size, hierarchical structure stochastic automata

1. はじめに

ニューラルネットの学習アルゴリズムとして最も良く知られているものに、Rumelhart らによって提案された逆誤差伝搬法 (BP 法) がある¹⁾。この手法は、さまざまな分野においてインテリジェントシステムの構築に多大の貢献を果してきたが、総誤差関数の局所的最小値へ落ち込みやすい等々の幾つかの問題点を有している^{2)~5)}。

BP 法の問題点を克服するために、これまで、慣性項を用いた BP 法^{1) 2)}、忘却を考慮したアルゴリズム⁶⁾、ハイブリッドアルゴリズム⁴⁾等の改良型 BP 法が幾つか提案されてきた。中でも、慣性項を用いた BP 法はかなり強力であり、最近では、元々の BP 法よりもこの手法が利用される場合が遥かに多くなっているのが実状である。

ところで、この慣性項を用いた BP 法にも一つの大きな問題点がある。それは、慣性項のパラメータの値並びにステップ幅を定める値にその学習性能が大きく依存するということである。

本論文では、階層構造確率オートマトン^{7) 8)}を活用することによって、慣性項を用いた BP 法に含まれる 2 つのパラメータを自動的に選択しニューラルネットの学習の高速化に役立てることができることを示す。

本論文の構成の概略について述べよう。まず、第 2 節で慣性項を用いた BP 法について紹介する。更に、第 3 節で階層構造確率オートマトン (HSSA) の概略を述べる。そして、第 4 節では、HSSA を利用して (慣性項を用いた BP 法に含まれる) 慣性項のパラメータ α の値並びにステップ幅を定めるパラメータ η の値を自動的に選択するアルゴリズムを提案する。ところで、BP 法に含まれる η (ステップ幅を定めるもの) や慣性項のパラメータ α の値を定める方法については、Yu⁹⁾らの研究が知られている。第 5 節では、シミュレーション例により、本提案手法と慣性項を用いた BP 法との比較を行うと共に、Yu⁹⁾らの研究との比較も行う。最後の第 6 節では、本提案手法についての若干の考察を行うと共に、今後の課題について検討する。

2. 慣性項を用いた BP 法

Rumelhart らによって提案された BP 法¹⁾ (これを以下では元々の BP 法と呼ぶ) は、一次元探索^{4) 5)}を行わないステップ幅固定の最急降下法を用いており、 k ステップ目における結合強度ベクトル $w^{(k)}$ は以下の式により $(k+1)$ ステップ目における結合強度ベクトル $w^{(k+1)}$ に変更される^{1) 5)}。

$$w^{(k+1)} = w^{(k)} - \eta \frac{\partial E_p(w)}{\partial w} \Big|_{w=w^{(k)}} \quad (1)$$

ここで、 $E_p(w)$ は第 p 入力パターンに対する誤差関数であり、 $\frac{\partial E_p(w)}{\partial w} \Big|_{w=w^{(k)}}$ は、 $w = w^{(k)}$ における $E_p(w)$ の勾配ベクトルを表している。(勾配ベクトル $\frac{\partial E_p(w)}{\partial w}$ の各成分は、Rumelhart らによって提案された逆誤差伝搬法によって求められる¹⁾。)

* 大阪教育大学教育学部 柏原市旭ヶ丘 4-698-1

* Faculty of Art and Science, Osaka Educational Univ., Kasiwara

(Received December 26, 1996)

(Revised November 25, 1997)

元々のBP法は、工学や医学を中心とする様々な分野において利用されてきたが、“総誤差関数の局所的最小値へ落ち込みやすい”という点や“必ずしも総誤差関数の単調減少性を保証しない”等、幾つかの問題点が指摘されている。こうした元々のBP法の問題点を克服するために、これまで幾つかの改良型BP法が提案されてきた。中でも、最近特に良く活用されるアルゴリズムとして、慣性項を用いたBP法がある¹⁾。このアルゴリズムは、結合強度ベクトル $w^{(k)}$ を変更する際に逆誤差伝搬法によって得られた $E_p(w)$ の勾配ベクトルの情報だけでなく、一ステップ前における結合強度ベクトルの変更量をも考慮しようというものであり、次のような結合強度ベクトル変更則に従っている。

$$\Delta w^{(k+1)} = -\eta \left. \frac{\partial E_p(w)}{\partial w} \right|_{w=w^{(k)}} + \alpha \Delta w^{(k)} \quad (2)$$

ここで、 α は正のある定数である。又、 $\Delta w^{(k)}$ は $w^{(k)} - w^{(k-1)}$ を意味しており、 k ステップ目における結合強度ベクトルの変更量を示している。

上で示された慣性項を用いたアルゴリズムは、(著者らの知る限りにおいて) 大域的最小値への収束等の理論的な保証を与えるものではないが、非常に実用的で収束性に優れていることが多くの人々の間で認識されている。しかしながら、このアルゴリズムの学習性能は(2)式に含まれるパラメータ α 及び η の値に依存しており、 α 及び η をどのように選ぶかが非常に重要な問題になっている。

注意1: 本論文では、慣性項を用いたBP法としてRumelhartら¹⁾によって提案されたものを考察の対象とする。(2)式以外に慣性項を用いたBP法としては次の(3)式で定義されるものが知られている²⁾。

$$\Delta w^{(k+1)} = -(1-\alpha)\eta \left. \frac{\partial E_p(w)}{\partial w} \right|_{w=w^{(k)}} + \alpha \Delta w^{(k)} \quad (3)$$

(紙面の都合により、本論文では、慣性項BP法としては、(2)式のみを考察の対象とする。)

3. 階層構造確率オートマトン

Fig.1 Learning mechanism of the hierarchical structure stochastic automata

階層構造確率オートマトン(HSSA)の概念は、ThathacharとRamakrishnanによって提案された⁷⁾⁸⁾¹⁰⁾。以下では、HSSAの基本モデルならびにその学習アルゴリズムについて簡単に紹介する。

Fig.1は、未知環境中で動作する階層構造確率オートマトンの学習メカニズムを示している。ここでは、第1レベルでは $\bar{A}$ 、第2レベルでは r 個の確率オートマトン $\bar{A}_1, \dots, \bar{A}_r$ 、第 s レベル($s=2, \dots, N$)では r^{s-1} 個の確率オートマトン $\bar{A}_{j_1 j_2 \dots j_{s-1}}$ ($j_i=1, \dots, r; i=1, \dots, s-1$)から成り立っている。(HSSA内の各確率オートマトンは、 r 個の出力を持っている)

次に、HSSAの動作についての概略を述べよう。今、時刻 t において $\bar{A}$ が出力確率ベクトル $(p_1(t), \dots, p_r(t))'$ に基づいて出力 α_{j_1} を選んだと仮定しよう。すると、第2レベルでは、 α_{j_1} に対応する確率オートマトン $\bar{A}_{j_1}$ が動作する。そして、 $\bar{A}_{j_1}$ はその出力確率ベクトル $(p_{j_1 1}(t), \dots, p_{j_1 r}(t))'$ に基づいて出力を選択する。次に、 $\bar{A}_{j_1}$ は出力 $\alpha_{j_1 j_2}$ を出したとしよう。すると、第3レベルでは $\alpha_{j_1 j_2}$ に対応する確率オートマトン $\bar{A}_{j_1 j_2}$ が動作する。こうした一連の動作が最下位レベル N まで続けられる。最下位レベルにある確率オートマトンから出力が出されると、環境はトップレベルから最下位レベルまでの一連の出力列に対応する応答を返す。そして、動作した一連の確率オートマトンは、自身が出した出力と環境からの返答に基づいて出力確率ベクトルを変更する。

未知環境からの応答を $\beta(t)$ としたとき、 $\beta(t)$ が0(reward)であるか又は1(penalty)であるかによってHSSAは各レベルの出力確率ベクトルを以下で示されるアルゴリズムに従って変更する⁸⁾：

第1レベル

1) 環境からの応答 $\beta(t)$ が0(reward)ならば、

$$\begin{aligned} p_{j_1}(t+1) &= p_{j_1}(t) + \lambda(t)(1 - p_{j_1}(t)) \\ p_{i_1}(t+1) &= p_{i_1}(t)(1 - \lambda(t)) \quad (i_1 \neq j_1) \end{aligned}$$

2) 環境からの応答 $\beta(t)$ が1(penalty)ならば、

$$\begin{aligned} p_{j_1}(t+1) &= p_{j_1}(t) - \mu(t)(1 - p_{j_1}(t)) \\ p_{i_1}(t+1) &= p_{i_1}(t) + \mu(t)p_{i_1}(t) \quad (i_1 \neq j_1) \end{aligned}$$

第 n レベル($n=2, \dots, N$)

1) 環境からの応答 $\beta(t)$ が0(reward)ならば、

$$\begin{aligned} p_{j_1 j_2 \dots j_n}(t+1) &= p_{j_1 j_2 \dots j_n}(t) + \lambda_{j_1 j_2 \dots j_n}(t) \\ &\quad \times (1 - p_{j_1 j_2 \dots j_n}(t)) \\ p_{j_1 \dots j_{n-1} i_n}(t+1) &= p_{j_1 \dots j_{n-1} i_n}(t) \\ &\quad \times (1 - \lambda_{j_1 j_2 \dots j_n}(t)) \quad (i_n \neq j_n) \end{aligned}$$

2) 環境からの応答 $\beta(t)$ が1(penalty)ならば、

$$\begin{aligned} p_{j_1 j_2 \dots j_n}(t+1) &= p_{j_1 j_2 \dots j_n}(t) - \mu_{j_1 j_2 \dots j_n}(t) \\ &\quad \times (1 - p_{j_1 j_2 \dots j_n}(t)) \\ p_{j_1 \dots j_{n-1} i_n}(t+1) &= p_{j_1 \dots j_{n-1} i_n}(t) \\ &\quad \times (1 + \mu_{j_1 j_2 \dots j_n}(t)) \quad (i_n \neq j_n) \end{aligned}$$

ここで,

$$\lambda_{j_1 j_2 \dots j_{n-1}}(t) = \frac{\lambda(t)}{p_{j_1}(t+1) \dots p_{j_1 j_2 \dots j_{n-1}}(t+1)}$$

$$\mu_{j_1 j_2 \dots j_{n-1}}(t) = \frac{\mu(t)}{p_{j_1}(t+1) \dots p_{j_1 j_2 \dots j_{n-1}}(t+1)}$$

但し, HSSA 内の各確率オートマトンの初期出力確率はすべて等しく $1/r$ とする. 又, 動作しなかった HSSA 内の確率オートマトンの出力確率ベクトルは変更されないものとする (注1).

4. 階層構造確率オートマトンを活用した慣性項パラメータ α 並びにステップ幅を定めるパラメータ η の自動調整

既に述べたように, 慣性項を用いた BP 法をニューラルネットの学習に使用した場合, 慣性項を利用する割合を示すパラメータ α の値並びにステップ幅 η の値が, アルゴリズムの学習性能に大きな影響を及ぼす.

そこで, 以下では, 階層構造確率オートマトン (HSSA) を用いることによって慣性項を利用する割合を示すパラメータ α の値並びにステップ幅 η の値を自動的に選択するアルゴリズムを提案する¹¹⁾. まず, 慣性項を利用する割合を示すパラメータ α の値の候補並びにステップ幅の大きさを定めるパラメータ η の値の候補をいくつか選ぶ. そして, 最下位レベルに配置された確率オートマトンから出されるそれぞれの出力と選ばれたそれぞれのパラメータ α 及び η の候補のペアとを一対一に対応させる. そして, HSSA の最下位レベルに配置されたある確率オートマトンが動作して一つの出力を選ぶということを, 対応するパラメータ α 及び η の値を持つ慣性項を用いた BP 法を利用してニューラルネットの学習を何ステップか行うことに関連づける.

Fig. 2 Relation between candidates of the momentum factor α and step size η and outputs from HSSA in the bottom level

参考までに, η の値の候補を3種類と α の値の候補を9種類考えた場合の HSSA の構成例を Fig.2に示す.

ここでは, HSSA は, 第1レベルに3出力を持つ確率オートマトンが配置されており各出力が3種類の η の中の一つの値に対応する. 又, 第2レベル及び第3レベルの確率オートマトンはそれぞれ3出力を持っており, 最下位レベルの確率オートマトンからの各出力が η と α の各ペアに対応する.

(注1) ここで提案された HSSA の学習アルゴリズムは, 絶対的良好 (absolute expediency) 等の優れた学習性能を持っていることが知られている^{8) 10)}

環境からの応答としては, 慣性項を用いた BP 法によってニューラルネットの学習が進んだかどうかによって, $\beta(t) = 0$ (reward) 又は $\beta(t) = 1$ (penalty) を HSSA に返すものとする. 言いかえると, 与えられた学習ステップ^(注2)内において総誤差関数 $E(w)$ の値が減少した場合, HSSA は環境から $\beta(t) = 0$ (reward) を受けるものとする. 逆に, $E(w)$ の値が変化しないかしくは増加した場合には, HSSA は環境から $\beta(t) = 1$ (penalty) を受けるものとする. そして, 環境からの応答が reward ($\beta(t) = 0$) か又は penalty ($\beta(t) = 1$) であるかに従って, HSSA は第3節で示したアルゴリズムによって出力確率ベクトルの変更を行う.

5. シミュレーション結果

提案手法の有効性を検討するために, EX-OR 並びに大気汚染物質濃度予測に関するニューラルネットの学習問題を考える^{4) 5)}.

EX-OR に関するニューラルネットの学習問題

各ビットの成分が0または1からなる8ビットの記号列を考える. この記号列は全部で 2^8 種類あるが, そのうち30パターンをランダムに選ぶ. このランダムに選ばれた30パターンのビット列を学習パターンとし, ニューラルネットの学習を行った.

Fig.3, Fig.4, 及び Fig.5は, 3種類の η の値 ($\eta = 1.0, \eta = 0.1, \eta = 0.01$) 及び5種類の α の値 ($\alpha = 0.0, \alpha = 0.1, \alpha = 0.2, \alpha = 0.3, \alpha = 0.5$) のそれぞれに対する慣性項 BP 法を用いた場合の総誤差関数の変化を示している^(注3).

本問題では $\eta = 1.0$ の場合が最も収束スピードが早くなっており, $\eta = 0.01$ の場合は, 総誤差関数の減少がほとんど見られない. つまり, 慣性項 BP 法に含まれるパラメータ η と α の選択如何によって総誤差関数の減少スピードが大きく異なることが分かる.

Fig. 3 Changes in the value of the total error function by the BP method with momentum (EX-OR problem; $\eta = 1.0$)

(注2) 本論文の5節におけるシミュレーション例では, 10ステップ行うものとした.

(注3) パラメータ α が $\alpha = 0.0$ の場合は, 元々の BP 法に相当する. 又, Fig.3~Fig.8 (Fig.12~Fig.17) において, 横軸はステップ数, 縦軸の総誤差関数の値としては, (入力ユニット数) $\times$ (パターン数) で割ったものを用いている.

Fig. 4 Changes in the value of the total error function by the BP method with momentum (EX-OR problem; $\eta = 0.1$)

Fig. 5 Changes in the value of the total error function by the BP method with momentum (EX-OR problem; $\eta = 0.01$)

Fig.6及び Fig.7は、パラメータ η 及び α の値を HSSA によって自動選択させた場合のシミュレーション結果を与えている。 η を3種類^(注4)、 α を9種類^(注5)の中から選んだ場合には、HSSA を活用した方法は、Fig.4($\eta = 0.1$)及び Fig.5($\eta = 0.01$)の結果より優れた性能を示していることがわかる。又、 η 及び α の選択範囲を広げて、 η を5種類^(注6)にして α を125種類^(注7)の中から選んだ場合には、HSSA による方法は、慣性項 BP 法において $\eta = 1.0$ と選んだ場合 (Fig.3)と同等もしくはそれ以上の性能を発揮しているのが理解される^(注8)。

Fig. 6 Changes in the value of the total error function by the proposed method (EX-OR problem; $\eta = 3, \alpha = 9$)

Fig. 7 Changes in the value of the total error function by the proposed method (EX-OR problem; $\eta = 5, \alpha = 125$)

ところで、慣性項を用いた BP 法に含まれるパラメータ η 及び α の決定手法としては、1995 年に発表された Yu⁹⁾らの方法が良く知られている。Fig.8に Yu⁹⁾らの論文に示された5つの手法を用いてニューラルネットの学習を行った結果を示す。Newtonlike Method(NM) 法や Acceptable Line Search(ALS) 法はあまり収束性能が良くないが、Polynomial Approximation(PA) 法、Linear Expansion(LE) 法、あるいは Newtonlike Method with Momentum(NMWM) 法はかなりの収束性能を示している。

Fig. 8 Changes in the value of the total error function by the five methods proposed by Yu et al.⁹⁾ (EX-OR problem)

ところで、上で述べたシミュレーション結果はすべてステップ数^(注9)を横軸にとったものである。それぞれの方法によって1ステップの計算時間が大幅に異なることを考えると、CPU time を横軸として総誤差関数の変化を調べてみることも必要ではないかと考えられる。こうした見地から、本論文における提案手法と Yu⁹⁾らの方法を比較した結果を Fig.9～Fig.11に示す。(これらの図では、横軸の目盛りとしてミリ秒を用いている。)

計算時間という観点から見ると、提案手法の方が Yu⁹⁾らの方法と比較して優れているのが理解される。

Fig. 9 Changes in the value of the total error function by the proposed method (EX-OR problem; $\eta = 3, \alpha = 9$)

(注4) $\eta = 1.0, \eta = 0.1, \eta = 0.01$ の3種類。

(注5) $\alpha = 0.066$ から $\alpha = 0.333$ までを8等分割し、9種類の α の値を定めた。 $(\alpha = 0.066$ 及び $\alpha = 0.333$ を含む)

(注6) $\eta = 0.01, \eta = 0.05, \eta = 0.1, \eta = 0.5, \eta = 1.0$ の5種類。

(注7) $\alpha = 0.0$ から $\alpha = 0.5$ までを124等分割し、それぞれの分割点に対応する α の値を125種類の各 α の値とした。 $(\alpha = 0.0$ 及び $\alpha = 0.5$ を含む)

(注8) HSSA は4レベルから構成されており、各レベルに5出力を持つ確率オートマトンが配置されているものを用いた。

(注9) 1ステップを1エポックとして考えた。

Fig. 10 Changes in the value of the total error function by the proposed method (EX-OR problem; $\eta = 5, \alpha = 125$)

Fig. 11 Changes in the value of the total error function by the five methods proposed by Yu et al.⁹⁾ (EX-OR problem)

大気汚染物質濃度予測のためのニューラルネットの学習問題

Fig.12, Fig.13, 及び Fig.14は、3種類の η の値($\eta = 1.0, \eta = 0.1, \eta = 0.01$)及び5種類の α の値($\alpha = 0.0, \alpha = 0.1, \alpha = 0.2, \alpha = 0.3, \alpha = 0.5$)のそれぞれに対する慣性項BP法を用いた場合の総誤差関数の変化を示している(注3)。

$\eta = 1.0$ の場合は、5種類のすべてのパラメータ α の値に対して局所的最小値に落ち込んでしまっている。 $\eta = 0.1$ の場合は、 α の値によっては(例えば $\alpha = 0.3$)収束性がやや改善されている。又、 $\eta = 0.01$ の場合は、全体的に収束スピードが遅い。慣性項を用いたBP法によって学習を行う場合、パラメータ η と α の値の選択が(運悪く)適切でない場合、なかなか総誤差関数の減少が見込まれない可能性があることがこのシミュレーション例から理解される。

慣性項を用いたBP法に含まれる2つのパラメータ η 及び α の値をHSSAにより自動選択させた場合の学習の様子をFig.15及びFig.16に示す。Fig.12~Fig.14の学習結果と比較して総誤差関数の減少のスピードが加速しているのが理解されるであろう。

Fig. 12 Changes in the value of the total error function by the BP method with momentum (Air pollution problem; $\eta = 1.0$)

Fig. 13 Changes in the value of the total error function by the BP method with momentum (Air pollution problem; $\eta = 0.1$)

Fig. 14 Changes in the value of the total error function by the BP method with momentum (Air pollution problem; $\eta = 0.01$)

Fig.15のシミュレーションでは、3種類の η の値及び9種類の α の値の中からHSSAによって自動選択を行った(注4)(注5)。又、Fig.16のシミュレーションでは、5種類の η (注6)及び125種類の α (注7)の値からHSSAにより自動選択を行った(注8)。

Fig. 15 Changes in the value of the total error function by the proposed method (Air pollution problem; $\eta = 3, \alpha = 9$)

Fig. 16 Changes in the value of the total error function by the proposed method (Air pollution problem; $\eta = 5, \alpha = 125$)

Fig.17より, Polynomial Approximation 法 (PA) 及び Newtonlike Method with Momentum 法 (NMWM) がかなり良い学習結果を与えているのがわかる。

Fig. 17 Changes in the value of the total error function by the five methods proposed by Yu et al.⁹⁾ (Air pollution problem)

Fig.15及び Fig.16, Fig.17はステップ数に対する総誤差関数の変化を示しているが, 次に, CPU time で比較してみよう。

Fig.18及び Fig.19に HSSA による総誤差関数の変化を示し, Fig.20に Yu⁹⁾らの 5 つの方法による総誤差関数の変化を示す。

PA 法及び NMWM 法はステップ数の比較においては HSSA とほぼ同程度の性能を示していたが, CPU time で比較となると, HSSA に比べて格段に悪くなっているのがわかる。これは, 様々な計算を必要とするため, かなりの時間を費したからではないかと推察される。

Fig. 18 Changes in the value of the total error function by the proposed method (Air pollution problem; $\eta = 3, \alpha = 9$)

Fig. 19 Changes in the value of the total error function by the proposed method (Air pollution problem; $\eta = 5, \alpha = 125$)

Fig. 20 Changes in the value of the total error function by the five methods proposed by Yu et al.⁹⁾ (Air pollution problem)

以下では, 上で行った 2 種類のシミュレーションに関して幾つかの考察を行う。

考察 1: 2 種類のシミュレーションにおいて, 本提案手法の有効性が確認されたが, その理由としてステップ幅 η と慣性項の割合を示すパラメータ α に関する出力確率分布の変化が適切に作用したのではないかと考えられる。

参考までに, EX-OR 問題における η の分布の変化並びに α の分布の変化を Fig.21並びに Fig.22に示す^(注 10)。

Fig.21より, 初期段階では $\eta = 1.0$ を選択する確率が大きく, 次第に $\eta = 0.1$ を選択する確率が増大しているのが分かる。これは $\eta = 1.0$ によって総誤差関数を減少させ, $\eta = 0.1$ で徐々に大域的最小値を見出していると考えられる。(Fig.3~Fig.5の結果より, 適切な η の選択を HSSA が行ったことが理解される。)

α に関しては, 幾つかの値に対応するものがかなりの確率を持ったまま残存している。これは, Fig.3~Fig.5より明らかのように, α の値の間に総誤差関数の減少という点に関して顕著な差がないことからこのような結果になったのではないかと考えられる。

考察 2: 上の 2 つのシミュレーションは, ランダムに選ばれた一つの (ニューラルネットの) 初期結合強度ベクトルに対して総誤差関数の最小化を実行して行われたものである。ところで, シミュレーションによって提案手法の有効性を主張するためには, 初期結合強度ベクトルをランダムに幾つか選択し, それら全体についての比較を行ってみる必要がある。

そこで, EX-OR 問題に対して, 初期結合強度ベクトルをランダムに 10 種類選び, 以下に示す 3 つの条件で比較を行った。その結果を Table.1(付録を参照されたい)に示す。

条件 1: 総誤差関数の値が初期の値と比較して 1/10 以下になるか, 計算回数が 300 ステップを越えるか, 又は CPU 時間が 10,000 ミリ秒を越えた場合に計算を終了する。

条件 2: 総誤差関数の値が初期の値と比較して 1/30 以下になるか, 計算回数が 900 ステップを越えるか, 又は CPU 時間が 30,000 ミリ秒を越えた場合に計算を終了する。

(注 10) 紙面の都合により大気汚染濃度予測問題に関しては省略する。

条件3: 総誤差関数の値が初期の値と比較して1/50以下になるか, 計算回数が1500ステップを越えるか, 又はCPU時間が50,000ミリ秒を越えた場合に計算を終了する。

Table.1では, それぞれの手法についてまず成功率 (Success Rate)^(注11)を10回のうち何回成功したかという形で比較し, 次に計算を終了した時点での総誤差関数の値を10回の平均で比較し, 更に成功した場合について総誤差関数の値が設定された値以下になるために平均時間としてどれぐらいにかかったかを比較している^(注12)。

大気汚染物質濃度予測問題に関しても同様の比較を行った。その結果をTable.2(付録を参照されたい)に示す^(注12)。

(Table.1及びTable.2において-の印は10回中一度も成功しなかったことを示している。)

Fig. 21 Changes in the value of the probability of selecting each value of η by the proposed method ($\eta: 3$)

Fig. 22 Changes in the value of the probability of selecting each value of α by the proposed method ($\alpha: 9$)

6. 考察並びに今後の課題

本論文では, 慣性項を用いたBP法におけるステップ幅 η と慣性項利用の割合を示すパラメータ α を同時に選択する方法として, 階層構造確率オートマトン(HSSA)を活用することを提案した。そして, EX-OR問題及び大気汚染物質濃度予測問題を例にとり, ニューラルネットの学習に関して本提案手法が有効であることをシミュレーションにより確認した。

(注11)制限ステップ数及び制限CPU時間内に初期総誤差の1/10, 1/30, あるいは1/50に総誤差が減少したかどうか。(減少した場合には成功とする。)

(注12)EX-OR問題における平均初期総誤差関数の値は0.113209であり, 大気汚染物質濃度予測問題における平均初期総誤差関数の値は0.117668である。

ところで, 本提案手法がニューラルネットの学習に対して効果的であったのは何故であろうか。我々は, この問いに対して, 現在の所, 次のような解釈をしている: 非線形計画法の分野における最小化アルゴリズムとしては降下法に属するもの⁽¹⁴⁾と確率的探索法が良く知られている。前者は, 関数の微分情報を主として利用するものであり, 局所的最小値への収束に優れている。一方, 後者は, Matyas型ランダム探索法⁽¹⁵⁾⁽¹⁶⁾や焼きなまし法⁽¹⁷⁾に代表されるようにかなりのステップ数を必要とするが, 大域的最小値を発見するのに優れている。本提案手法は, 上で述べた両者の利点を良い意味で融合しているといえるのではないだろうか。つまり, 慣性項を用いたBP法によって収束スピードを上げ(一般に, ニューラルネットの総誤差関数の局所的最小値への接近を早め), 階層構造確率オートマトンを導入することによって確率的ゆらぎを与え, 探索が局所的最小値に落ち込んでしまうのを防いでいるのではないかと考えられる。

しかしながら, 本手法の有効性を確認するためには, 理論的な収束性の保証を与えるか, または, 非常に多くのシミュレーション例に対してその効果を示す必要がある。既に, 株価予測や気液二相流の同定に用いたニューラルネットの学習問題に対しても, シミュレーションにより, その有効性を確認しているが, 今後更に様々な応用問題に適用することにより本手法の性能を検討する予定である。また, 本提案手法について理論的な収束の保証を与えることができるかどうか調べることも今後の課題として残されており, 我々はこの点についても検討を加えて行く所存である。

参考文献

- 1) D.E.Rumelhart et al: Parallel Distributed Processing, MIT Press (1986)
- 2) R.A.Jacobs: Increased Rates of Convergence through Learning Rate Adaptation, Neural Networks, 1, 295/307 (1988)
- 3) N.Baba: A New Approach for Finding the Global Minimum of Error Function of Neural Networks, Neural Networks, 2, 367/373 (1989)
- 4) N.Baba et al: A Hybrid Algorithm for Finding the Global Minimum of Error Function of Neural Networks and Its Applications, Neural Networks, 7, 1253/1265 (1994)
- 5) 馬場, 小島, 小澤: ニューラルネットの基礎と応用, 共立出版 (1994)
- 6) 石川: 忘却を用いたコネクショニストモデルの構造学習アルゴリズム, 人工知能学会誌, 5, 595/603 (1990)
- 7) M.A.L.Thathachar and K.R.Ramakrishnan: A Hierarchical System of Learning Automata, IEEE Trans. SMC, 11, 236/241 (1981)
- 8) K.R.Ramakrishnan: Hierarchical Systems and Cooperative Games of Learning Automata, Ph.D.Thesis, Indian Institute of Science (1982)
- 9) X.H.Yu et al: Dynamic Learning Rate Optimization of the Backpropagation Algorithm, IEEE Trans. Neural Networks, 6, 669/677 (1995)
- 10) 馬場: 学習オートマトン理論の最近の話題, 計測と制御, 26, 801/808 (1987)
- 11) 馬場他: ニューラルネットの学習アルゴリズムに関する一考察 — 慣性項BP法に対する階層構造確率オートマトンの適用,

第35回 SICE 講演会予稿集, 1, 241/242 (1996)

- 12) N.Baba et al: Utilization of Neural Networks for Constructing a User Friendly Decision Support System to Deal Stocks, Proceedings of ICNN, 818/823 (1995)
- 13) N.Baba et al: New Neural Network Training Algorithm and Its Applications, Journal of Artificial Neural Networks, 2, 37/54(1995)
- 14) 馬場, 坂和: 数理計画法入門, 共立出版 (1989)
- 15) J.Matyas: Random Optimization, Automation & Remote Control, 28, 246/253(1965)
- 16) F.J.Solis and R.J.Wets: Minimization by Random Search Techniques, Mathematics of Operations Research, 6, 19/30 (1981)
- 17) S.Kirkpatrick et al: Optimization by simulated annealing, IBM Thomas J.Watson Research Center Report (1982)

《付録》

Table 1 Simulation results obtained under the three conditions (EX-OR problem)

Condition 1 (1/10;300Steps;10000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(10/10)	(0/10)	(0/10)	(0/10)	(0/10)	(0/10)	(0/10)	(0/10)	(0/10)	(5/10)
average number of steps	228	—	—	—	—	—	—	—	—	221
average value of the total error function	0.0109	0.0703	0.0987	0.0482	0.0647	0.0908	0.0907	0.0959	0.0397	0.0208
CPU time [Milli seconds]	1064	—	—	—	—	—	—	—	—	3012
Condition 2 (1/30;900Steps;30000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(9/10)	(0/10)	(0/10)	(3/10)	(0/10)	(0/10)	(0/10)	(0/10)	(1/10)	(9/10)
average number of steps	247	—	—	450	—	—	—	—	645	355
average value of the total error function	0.0040	0.0335	0.0954	0.0134	0.0340	0.0645	0.0752	0.0875	0.0190	0.0046
CPU time [Milli seconds]	1147	—	—	25387	—	—	—	—	7490	4761
Condition 3 (1/50;1500Steps;50000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(10/10)	(2/10)	(0/10)	(7/10)	(3/10)	(0/10)	(0/10)	(0/10)	(5/10)	(10/10)
average number of steps	208	1298	—	557	1211	—	—	—	1063	425
average value of the total error function	0.0022	0.0142	0.0896	0.0084	0.0146	0.0533	0.0680	0.0814	0.0114	0.0022
CPU time [Milli seconds]	1547	4945	—	31424	43557	—	—	—	12422	5676

(LE = Linear Expansion, ALS = Acceptable Line Search, PA = Polynomial Approximation, NM = Newtonlike Method, NMWM = Newtonlike Method with Momentum)

Table 2 Simulation results obtained under the three conditions (Air pollution problem)

Condition 1 (1/10;300Steps;10000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(0/10)	(7/10)	(0/10)	(5/10)	(7/10)	(0/10)	(0/10)	(0/10)	(2/10)	(3/10)
average number of steps	—	168	—	96	84	—	—	—	274	130
average value of the total error function	0.0832	0.0162	0.0677	0.0169	0.0125	0.0353	0.0674	0.0633	0.0320	0.0179
CPU time [Milli seconds]	—	640	—	4474	1073	—	—	—	2780	1450
Condition 2 (1/30;900Steps;30000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(0/10)	(6/10)	(0/10)	(4/10)	(8/10)	(2/10)	(1/10)	(2/10)	(3/10)	(4/10)
average number of steps	—	323	—	311	233	169	68	54	407	250
average value of the total error function	0.0849	0.0124	0.0189	0.0095	0.0057	0.0166	0.0343	0.0252	0.0155	0.0118
CPU time [Milli seconds]	—	1118	—	14200	4803	28005	25340	19960	4077	2760
Condition 3 (1/50;1500Steps;50000Milliseconds)										
	BP ($\eta=1.0,$ $\alpha=0.0$)	BP ($\eta=0.1,$ $\alpha=0.0$)	BP ($\eta=0.01,$ $\alpha=0.0$)	LE	ALS	PA	NM	NMWM	HSSA ($\eta=15,$ $\alpha=125$)	HSSA ($\eta=15,$ $\alpha=125$)
success rate	(0/10)	(6/10)	(1/10)	(4/10)	(8/10)	(3/10)	(1/10)	(3/10)	(3/10)	(4/10)
average number of steps	—	383	1487	1518	281	211	85	81	472	294
average value of the total error function	0.0853	0.0114	0.0110	0.0087	0.0046	0.0127	0.0231	0.0161	0.0151	0.0110
CPU time [Milli seconds]	—	1318	4700	17340	5745	36210	31650	29723	4683	3230

(LE = Linear Expansion, ALS = Acceptable Line Search, PA = Polynomial Approximation, NM = Newtonlike Method, NMWM = Newtonlike Method with Momentum)

[著者紹介]

馬場 則 夫 (正会員)

1974年, 京都大学大学院工学研究科博士課程数理工学専攻修了。同年徳島大学工学部情報工学科助手, 同講師, 同助教授を経て, 1988年同知能情報工学科教授, 1990年大阪教育大学教育学部教授, 現在に至る。ニューラルネット, ゲームング, 遺伝的アルゴリズム, 学習オートマトン, 数理計画法などの研究に従事。工学博士。

佐藤 剣

1996年大阪教育大学教育学部情報科学専攻卒業。1998年3月同大学院修士課程修了。1998年4月より京都大学大学院情報科学研究科博士課程システム科学専攻進学。ニューラルネット, 学習オートマトン, 強化学習の研究に従事。