

2次元探索共役勾配法を用いた多層神経回路網の高速学習法†

吉田 利信††

多層神経回路網の結合係数を学習する逆伝播学習法は自動微分法の一様である。本論文では、自動微分法および自動微分法を用いた2次元探索共役勾配法の概要が示され、この2次元探索共役勾配法を用いた多層神経回路網の高速学習法が提案されている。本方法は逆伝播法の学習パラメータと慣性パラメータを自動的に制御する共役勾配法であり、最急降下法である逆伝播法よりも少ない回数の繰返しで収束する。本方法にはヘッセ行列の2次形式などの計算が含まれているため、ヘッセ行列を求めてからこれらの値を計算すると、神経回路網の大きさの2乗に比例した計算領域と計算量が必要となる。しかし自動微分法を用いることにより、神経回路網の大きさに比例した計算領域と計算量でこれらの値を計算することが可能である。本論文で提案されている高速学習法は、逆伝播法の高々6倍の計算量で1回の繰返し計算をする共役勾配法である。数値実験において逆伝播法、慣性項をとらぬ逆伝播法、学習パラメータと慣性パラメータを逐次変化させる逆伝播法および本高速学習法を用いた実験結果が比較され、本方法を用いることによって、これらの逆伝播学習法よりもかなり少ない反復回数で学習が収束し、繰返し1回あたりの計算時間が逆伝播法の約3倍となったことが示されている。

1. はじめに

1964年にWengert¹⁾によって示された自動微分法は、Baurら²⁾、Kimら³⁾、Iri⁴⁾、Sawyer⁵⁾によって関数の計算過程を逆にたどる自動微分法(スカラ関数に対しては、変数の数によらずにすべての偏導関数値が高速に得られることから、高速自動微分法とも呼ばれている)へと発展し、すでにこれらの自動微分法を手軽に用いるためのシステム^{7),8),10)}が開発されるまでになってきた。またRumelhartら¹¹⁾が逆伝播法として示した多層神経回路網の学習法は、この関数の計算過程を逆にたどる自動微分法そのものであった。

著者は、自動微分法を用いた2次元探索共役勾配法を提案した¹³⁾。この方法は、最急降下方向と直前の変数の修正方向を含む2次元平面上において、多変数スカラ値の目的関数を2次関数に近似し、その近似関数の極小点を探索し変数を修正する方法であり、勾配ベクトルを用いた共役方向法である。この方法で必要となる目的関数の勾配ベクトル、ベクトルと勾配ベクトルとの内積、およびヘッセ行列の2次形式の値を求めするためには、変数の数の2乗に比例した記憶領域と計算時間が必要であると考えられてきたが、自動微分法を用いることによって目的関数自身の大きさに比例し

た記憶領域と計算時間で求められることが示された。すなわち、この2次元探索共役勾配法では勾配ベクトルを関数の計算過程を逆にたどる自動微分法(以下、TD法と記す)を用いて求め、2次元探索において必要となるベクトルと勾配ベクトルとの内積および2次形式の値を、Wengertの自動微分法(以下、BU法と記す)を用いて求める。

本論文では、2次元探索共役勾配法を多層神経回路網の結合係数の学習に用いる方法を提案する。

逆伝播法では勾配ベクトルをTD法で求め、その勾配ベクトルを η (学習パラメータ)倍したベクトルだけ神経回路網の結合係数を修正しているが、最急降下法であるため一般に収束は遅い。この修正ベクトルに直前の修正ベクトルを α (慣性パラメータ)倍したベクトルを加えることによって収束が加速されることが示されている¹¹⁾。しかし、これらのパラメータは学習の過程において最適な値が変化しており、固定したパラメータで学習を行う場合、これらのパラメータは何回かの試行によって決定されなければならない。これらのパラメータを逐次変化させることによって学習を加速する方法も示されている¹²⁾。本論文で提案する方法は、これらの2つのパラメータ η と α を共役勾配法を用いて自動的に制御する方法であり、単なる最急降下法である逆伝播法よりも、学習の収束に必要な反復回数が少ない。

第2章に自動微分法を、第3章に2次元探索共役勾配法を概観し、第4章では、多層神経回路網の学習法として2次元探索共役勾配法を用いる高速学習法を提

† Rapid Learning Method for a Multilayered Neural Network Using a Conjugate Gradient Method with Two-Dimensional Search by TOSHINOBU YOSHIDA (Department of Electrical and Electronics Engineering, Faculty of Engineering, Chiba University).

†† 千葉大学工学部電気電子工学科

案する。すなわち、神経回路網における TD法を用いた勾配ベクトルの具体的な計算方法、および、BU法を用いたベクトルと勾配ベクトルとの内積、および2次形式の値の具体的な計算方法を示す。第5章では、逆伝播法、慣性項をとまう逆伝播法、学習パラメータと慣性パラメータを逐次変化させる逆伝播法および本高速学習法を用いた実験結果を示し比較する。

2. 自動微分法

自動微分法の詳細については文献1)~10)に示されているので、本論文ではその概略のみを示す。

計算過程は演算を節点に対応させることによって、閉路のない有向グラフに表すことができる。このグラフを計算グラフと呼ぶ。たとえば、

$$x^k_j = f(\sum_i w^k_{ji} x^{k-1}_i) \tag{1}$$

に対して図1に示す計算グラフが対応する。

計算グラフの各辺に要素的偏導関数を対応させる。要素的偏導関数とは、有向辺の終点の節点を v 、始点の節点を u とするとき、 $\partial v/\partial u$ と定義する。図1の各辺にその要素的偏導関数を示す。計算グラフの各節点 v には値 dv を対応させる。ここで dv は、BU法および TD法において以下のように定義される。

BU法

節点 v に入る辺を e_1, \dots, e_k 、辺上の要素的偏導関数を de_1, \dots, de_k 、辺の始点を v_1, \dots, v_k とする。節点 v_i

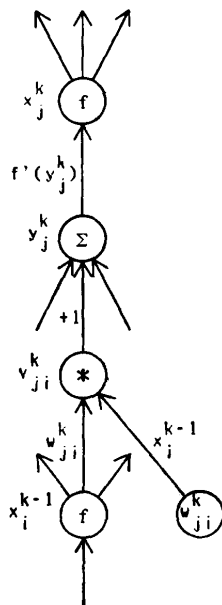


図1 (1)式の計算グラフ
Fig. 1 The computational graph of Eq. (1).

上の値 dv_i に対して、節点 v 上の値 dv を

$$dv = \sum_i de_i dv_i \tag{2}$$

と定義する。

TD法

節点 v から出る辺を e_1, \dots, e_k 、辺上の要素的偏導関数を de_1, \dots, de_k 、辺の終点を v_1, \dots, v_k とする。節点 v_i 上の値 dv_i に対して、節点 v 上の値 dv を

$$dv = \sum_j de_j dv_j \tag{3}$$

と定義する。

独立変数の節点を x_1, \dots, x_n とし、それらの節点にそれぞれ $dx_1=y_1, \dots, dx_n=y_n$ を対応させ、関数の計算の順 (Bottom Up) に上の BU法を行うと、関数の節点 f_1, \dots, f_m にそれぞれ

$$df_j = \sum_i (\partial f_j / \partial x_i) y_i \tag{4}$$

が得られる。つまり、ベクトル関数 f のヤコビ行列 J とベクトル y との積 Jy が得られる。

また、関数の節点を f_1, \dots, f_m とし、それらの節点にそれぞれ $df_1=y_1, \dots, df_m=y_m$ を対応させ、関数の計算とは逆の順 (Top Down) に上の TD法を行うと、独立変数の節点 x_1, \dots, x_n にそれぞれ

$$dx_i = \sum_j y_j (\partial f_j / \partial x_i) \tag{5}$$

が得られる。

スカラ関数の節点 f に $df=1$ を対応させ、TD法を行うことによって、独立変数の数によらずに、関数計算の手間の高々定数倍の手間で関数 f の勾配 ∇f が得られる。神経回路網の逆伝播法はこのTD法である。

またスカラ関数に対して、変数の節点にベクトル y を対応させ BU法を行うと、 $\nabla f \cdot y$ が節点 f に得られる。このスカラ関数 $\nabla f \cdot y$ を計算グラフに表し、変数の節点にベクトル z を対応させ、さらに BU法を行うと $y \cdot \nabla^2 f z$ が得られる。

このように自動微分法を用いることによって、勾配ベクトル ∇f 、ベクトルと勾配ベクトルとの内積 $\nabla f \cdot y$ 、および2次形式の値 $y \cdot \nabla^2 f z$ が、変数の個数によらずに関数計算の手間の高々定数倍の手間で、かつ、関数計算に必要な記憶領域の高々定数倍の領域を用いて求まる。

3. 2次元探索共役勾配法

スカラ関数 $f: R^n \rightarrow R$ の極小値を、点 x での最急降下ベクトル $g = -\nabla f(x)$ と直前の変位ベクトル p とが張る平面上を2次元探索し、これを繰り返すことによって f の極小値を探索する方法が開発されてい

る¹³⁾.

この方法の概略を以下に示す。まず、つぎの記号を定義する。

$$\begin{aligned} dg &= \nabla f(\mathbf{x})' \mathbf{g} \\ dp &= \nabla f(\mathbf{x})' \mathbf{p} \\ ghg &= \mathbf{g}' \nabla^2 f(\mathbf{x}) \mathbf{g} \end{aligned} \quad (6)$$

$$\begin{aligned} ghp &= \mathbf{g}' \nabla^2 f(\mathbf{x}) \mathbf{p} \\ php &= \mathbf{p}' \nabla^2 f(\mathbf{x}) \mathbf{p} \end{aligned}$$

$$\mathbf{b} = \begin{pmatrix} dg \\ dp \end{pmatrix} \quad (7)$$

$$A = \begin{pmatrix} ghg & ghp \\ ghp & php \end{pmatrix} \quad (8)$$

$$\xi = \begin{pmatrix} \eta \\ \alpha \end{pmatrix} \quad (9)$$

点 \mathbf{x} の近傍で $f(\mathbf{x} + \eta \mathbf{g} + \alpha \mathbf{p})$ を η と α の 2 変数の 2 次関数に近似し、次のように表す。

$$\begin{aligned} f(\mathbf{x} + \eta \mathbf{g} + \alpha \mathbf{p}) \\ \approx f(\mathbf{x}) + \mathbf{b}' \xi + \frac{1}{2} \xi' A \xi \end{aligned} \quad (10)$$

A が正定値のときは、 $\xi = -A^{-1} \mathbf{b}$ とおくことによって (10) 式の右辺の極小値を得ることができる。

A が負定値のときは (10) 式の右辺の 2 次関数は極大値を持ち、 $\xi = -A^{-1} \mathbf{b}$ は極大となる位置を示す。そこで、 $\xi = A^{-1} \mathbf{b}$ とおくことによって、 f の値を減少させる方向に進むことができる。

A が正定値でも負定値でもないとき、すなわち、(10) 式の右辺の 2 次関数が鞍点を持つとき、 f は A の正の固有値に対応する固有ベクトルの方向に下に凸の 2 次関数であり、負の固有値に対応する固有ベクトルの方向に上に凸の 2 次関数である。正の固有方向の最小を与えるベクトルと負の固有方向の最大値を与えるベクトルの合成ベクトルを $\xi = -A^{-1} \mathbf{b}$ は与えている。そこで、負の固有方向の最大値を与えるベクトルを逆向きにして合成することによって、 f の値を減少させる方向に進むことができる。このベクトルは次の式で表される¹³⁾。

$$c = [(ghg - php)^2 + 4ghp^2]^{1/2} \quad (11)$$

$$\xi = -\frac{1}{c} \begin{pmatrix} ghg - php & 2ghp \\ 2ghp & php - ghg \end{pmatrix} A^{-1} \mathbf{b} \quad (12)$$

A が正則でないとき、計算の最初で $\mathbf{p} = \mathbf{0}$ のとき、および再出発のときは、最急降下方向に進む。もし $ghg > 0$ ならば $\eta = -dg/ghg$, $\alpha = 0$ とし、もし $ghg < 0$ ならば $\eta = dg/ghg$, $\alpha = 0$ とする。 $ghg = 0$ のときは、 $\eta = 1$, $\alpha = 0$ とする。

これらの考え方に基づいた手順を以下に示す。

2次元探索共役勾配法

手順 1. $\mathbf{x} \in R^n$, $\mathbf{p} := \mathbf{0}$, $f_{old} :=$ 十分大きな数, $r := N$ とする。

手順 2. $f := f(\mathbf{x})$, $\mathbf{g} := -\nabla f(\mathbf{x})$ を計算する。

手順 3. $\|\mathbf{g}\| < \varepsilon$ ならば終了。

手順 4. $f < f_{old}$ ならば手順 5 へ。そうでなければ $\mathbf{p} := \mathbf{p}/2$, $\mathbf{x} := \mathbf{x} - \mathbf{p}$, $r = N$ とし、手順 2 へ。

手順 5. \mathbf{b} および A を計算する。

手順 6. $\det A = 0$ かまたは $r \geq N$ ならば手順 7 へ。 A が正定値のとき、 $\xi := -A^{-1} \mathbf{b}$, $r := r + 1$ とし手順 8 へ。

A が負定値のとき、 $\xi := A^{-1} \mathbf{b}$ とし手順 8 へ。

$\det A < 0$ のとき、(11), (12) 式を用いて ξ を定め手順 8 へ。

手順 7. $r := 1$, $\alpha := 0$ とし、

もし $ghg > 0$ ならば、 $\eta := -dg/ghg$,

もし $ghg < 0$ ならば、 $\eta := dg/ghg$,

それ以外るとき、 $\eta := 1$ とする。

手順 8.

$$\mathbf{p} := \eta \mathbf{g} + \alpha \mathbf{p}$$

$$\mathbf{x} := \mathbf{x} + \mathbf{p}$$

とし、手順 2 へ。

ここで ε は収束判定のためのパラメータ、 N は N 回の繰り返しの後、最急降下方向から再出発するためのパラメータである。

関数 f が次のような n 次元ベクトル \mathbf{x} の 2 次関数であるとすると

$$f(\mathbf{x}) = c + \mathbf{b}' \mathbf{x} + \frac{1}{2} \mathbf{x}' H \mathbf{x} \quad (13)$$

ここで H は $n \times n$ の実対称正定値行列、 \mathbf{b} は n 次元ベクトル、 c はスカラーとする。このとき、上の手順によって生成される \mathbf{p} のそれぞれは H に関して互いに共役であること、および、これらの方向ベクトルは Hestenes-Stiefel-Daniel の共役勾配法の公式を用いたときに生成される方向ベクトルと同一であることが示されている¹³⁾。つまり、2 次関数に対しては上の手順は共役勾配法であり、高々 n 回の繰り返しにより極小となる \mathbf{x} を与える。

4. 多層神経回路網の高速学習法

離散時間、連続情報の多層神経回路網モデルとして、次のものを考える。第 0 層を入力層、第 1 層から第 $n-1$ 層を中間層、第 n 層を出力層とし、第 k 層の細胞数を m_k とする。第 k 層 j 番目の細胞の出力を

x^k_j , 第 $k-1$ 層 i 番目の細胞から第 k 層 j 番目の細胞への結合係数を w^k_{ji} とする. 神経細胞の出力関数 f は単調増加 2 回連続微分可能関数とし, モデルの動作は次のように記述されるものとする.

$$y^k_j = \sum_i w^k_{ji} x^{k-1}_i \quad (14)$$

$$x^k_j = f(y^k_j) \quad (15)$$

モデルに学習させる入力と教師信号との組を

$$\begin{aligned} \mathbf{x}_a &= (x_{a1}, x_{a2}, \dots)^t \\ \mathbf{c}_a &= (c_{a1}, c_{a2}, \dots)^t \\ a &= 1, 2, \dots, N_s \end{aligned} \quad (16)$$

とし, 入力 \mathbf{x}^0 を \mathbf{x}_a としたときの出力 \mathbf{x}^n を \mathbf{x}_a^n と書くことにし, 誤差関数 R を次のように定義する.

$$R = \frac{1}{2} \sum_a \sum_j (x_a^n_j - c_{aj})^2 \quad (17)$$

この誤差関数 R は結合係数 w^k_{ji} の関数である. 結合係数全体を \mathbf{w} と表すことにする.

誤差関数を極小にする \mathbf{w} を求める高速学習法を以下に具体的に示す. 本高速学習法には, 第 3 章に示した 2 次元探索共役勾配法を用いる.

2次元探索共役勾配法の手順 2 における最急降下方向ベクトル $\mathbf{g} = -\nabla R(\mathbf{w})$ を, TD 法すなわち逆伝播法によって R の値とともに次のように求める.

TD 法による \mathbf{g} の導出方法

手順 1. $R := 0, \mathbf{g} := 0$ とし, すべての a に対して手順 2 から手順 6 を繰り返す.

手順 2. 入力 \mathbf{x}^0 を \mathbf{x}_a とし, $k=1, \dots, n$ の順で次の計算を行う.

$$\begin{aligned} y^k_j &:= \sum_i w^k_{ji} x^{k-1}_i \\ x^k_j &:= f(y^k_j) \end{aligned}$$

手順 3. 次の計算を行う.

$$R := R + \frac{1}{2} \sum_j (x^n_j - c_{aj})^2$$

手順 4. 次の計算を行う.

$$dy^n_j := (x^n_j - c_{aj}) f'(y^n_j)$$

手順 5. $k=n, \dots, 2$ の順で次の計算を行う.

$$\begin{aligned} dy^{k-1}_i &:= (\sum_j w^k_{ji} dy^k_j) f'(y^{k-1}_i) \\ dw^k_{ji} &:= dy^k_j x^{k-1}_i \\ g^{k-1}_i &:= g^{k-1}_i - dw^k_{ji} \end{aligned}$$

手順 6. 次の計算を行う.

$$\begin{aligned} dw^{1}_{ji} &:= dy^1_j x^0_i \\ g^{1}_i &:= g^{1}_i - dw^{1}_{ji} \end{aligned}$$

この TD 法による \mathbf{g} の導出方法の手順 5 における計算グラフを図 2 に示す.

2次元探索共役勾配法の手順 5 における \mathbf{b} および A

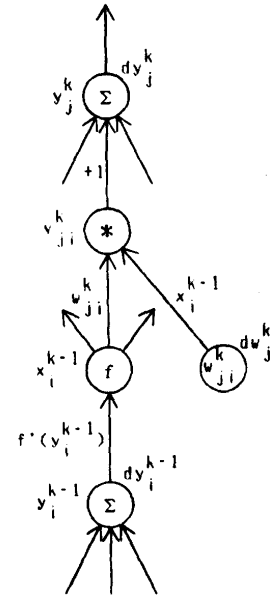


図 2 TD 法による \mathbf{g} の導出方法の手順 5 における計算グラフ

Fig. 2 The computational graph in Step 5 of the derivation procedure of \mathbf{g} by the TD method.

は, (6) 式の dg, dp, ghg, ghp , および php を用いて (7), (8) 式のように定義されている. これらの内積や 2 次形式は, BU 法によって次のように求められる.

BU 法による dg, dp, ghg, ghp および php の導出方法

手順 1. $dg := 0, dp := 0, ghg := 0, ghp := 0, php := 0$ とし, 各 a について手順 2 を繰り返す.

手順 2. $k=1, \dots, n$ の順で手順 3 を繰り返す.

手順 3. $j=1, \dots, m_k$ について以下を計算する.

$$\begin{aligned} y &:= \sum_i w^k_{ji} x^{k-1}_i \\ gy &:= \sum_i (g^k_{ji} x^{k-1}_i + w^k_{ji} g x^{k-1}_i) \\ py &:= \sum_i (p^k_{ji} x^{k-1}_i + w^k_{ji} p x^{k-1}_i) \\ ggy &:= \sum_i (2g^k_{ji} g x^{k-1}_i + w^k_{ji} gg x^{k-1}_i) \\ gpy &:= \sum_i (g^k_{ji} p x^{k-1}_i + p^k_{ji} g x^{k-1}_i \\ &\quad + w^k_{ji} gp x^{k-1}_i) \\ ppv &:= \sum_i (2p^k_{ji} p x^{k-1}_i + w^k_{ji} pp x^{k-1}_i) \\ x^k_j &:= f(y) \\ gx^k_j &:= f'(y) gy \\ px^k_j &:= f'(y) py \\ ggx^k_j &:= f'(y) ggy + f''(y) gy^2 \\ gpx^k_j &:= f'(y) gpy + f''(y) gy py \\ ppv^k_j &:= f'(y) ppv + f''(y) py^2 \end{aligned}$$

$k=n$ のときはさらに以下を計算する。

$$\begin{aligned} u &:= x^n_j - c_{0j} \\ dg &:= dg + u \, gx^n_j; \\ dp &:= dp + u \, px^n_j; \\ ghg &:= ghg + gx^n_j{}^2 + u \, ggx^n_j; \\ ghp &:= ghp + gx^n_j \, px^n_j \\ &\quad + u \, gpx^n_j; \\ php &:= php + px^n_j{}^2 + u \, ppx^n_j; \end{aligned}$$

図3に手順3における計算グラフを示す。この図において、 dw^{kji} を g^{kji} としたときのBU法による gx^k_j の計算過程が図の右側に、 dw^{kji} を p^{kji} としたときのBU法による px^k_j の計算過程が図の左側に示されている。また、 dw^{kji} を g^{kji} とし、BU法を行うことによって、節点 gx^k_j に ggx^k_j が、節点 px^k_j に gpx^k_j が得られる。同様に、 dw^{kji} を p^{kji} とし、BU法を行うことによって、節点 px^k_j に ppx^k_j が得られる。手順3は、これらの計算過程を表したものである。

記憶領域と計算量について以下に考察する。記法 $N(\mathbf{v})$ で \mathbf{v} の要素の数を表すことにする。 \mathbf{x} と \mathbf{w} の要素の数は神経回路網モデルの定義から次のように表される。

$$N(\mathbf{x}) = \sum_i m_i \tag{18}$$

$$N(\mathbf{w}) = \sum_i m_i m_{i-1} \tag{19}$$

まず、TD法による \mathbf{g} の導出方法の記憶領域と計算量について考察する。

R だけを計算するためには、 \mathbf{w} と \mathbf{x} のための記憶領域が必要である。また、 R と \mathbf{g} を計算するためには、 \mathbf{w} 、 \mathbf{g} 、 \mathbf{x} 、 \mathbf{y} 、および $d\mathbf{y}$ の記憶領域が必要である。ただし、 f として、

$$f(u) = 2/(1 + \exp(-u)) - 1 \tag{20}$$

を用いるとき、手順2より

$$f'(y^k_i) = (1 - x^k_i{}^2)/2 \tag{21}$$

となり、 \mathbf{y} を記憶する必要がなくなる。したがって、 $N(\mathbf{w}) = N(\mathbf{g})$ 、 $N(\mathbf{x}) = N(d\mathbf{y})$ より、 R だけを計算するための記憶領域の2倍の記憶領域で R と \mathbf{g} を計算することができる。

手順2における積の演算は $N(\mathbf{w})$ 個、和の演算も $N(\mathbf{w})$ 個、 f の演算が $N(\mathbf{x})$ 個である。これらの演算

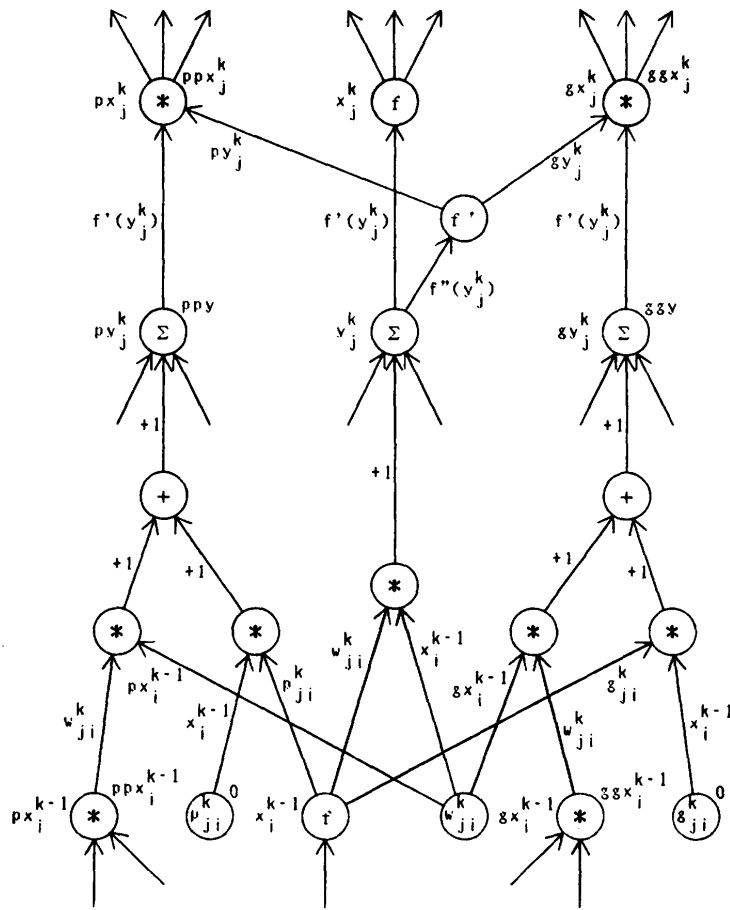


図3 BU法による dg 、 dp 、 ghg 、 ghp および php の導出方法の手順3における計算グラフ
Fig. 3 The computational graph in Step 3 of the derivation procedure of dg 、 dp 、 ghg 、 ghp and php by the BU method.

と手順3のわずかな演算で R が計算される。手順4、5、6では、積の演算は $2N(\mathbf{w}) + N(\mathbf{x}) - m_1 m_0$ 個、和の演算は $2N(\mathbf{w}) + m_n - m_1 m_0$ 個、 f' の演算が $N(\mathbf{x})$ 個である。したがって、 R だけを計算するための計算量のほぼ3倍の計算量で R と \mathbf{g} を計算することができる。

つぎにBU法による dg 、 dp 、 ghg 、 ghp および php の導出方法の記憶領域と計算量について考察する。

f として(20)式を用いた場合、 f' は(21)式で与えられ、 $f'' = -ff'$ となる。したがって、この手順においても \mathbf{y} を記憶する必要がなく、 $N(\mathbf{w})$ の大きさの \mathbf{w} 、 \mathbf{g} 、 \mathbf{p} 、および $N(\mathbf{x})$ の大きさの \mathbf{x} 、 \mathbf{gx} 、 \mathbf{px} 、 \mathbf{ggx} 、 \mathbf{gpx} 、 \mathbf{ppx} のみが用いられている。したがって、 R だけを計算するための記憶領域の3倍強の記憶領域のみで dg から php までの値を計算できる。

また、この手順における積の演算はほぼ $14N(w) + 11N(x)$ 個、和の演算はほぼ $12N(w) + 3N(x)$ 個、 f , f' , f'' の演算がそれぞれ $N(x)$ 個である。したがって、 R だけを計算するための計算量はほぼ 14 倍の計算量で、 dg から php までの値を計算できる。

以上の考察から、本高速学習法は R だけを計算するための記憶領域の 3 倍強の記憶領域を用いることが示された。また、逆伝播法は R と g の計算を含み、2次元探索共役勾配法は R と g の計算と dg から php までの計算を含むことから、繰り返し 1 回について本高速学習法は逆伝播法の高々 6 倍の計算を含んでいることが示された。

5. 実験例

多層神経回路網モデルとして、細胞数 20 個の入力層、細胞数 10 個の第 1 中間層、第 2 中間層、細胞数 2 個の出力層からなるモデルを用いて実験を行った。神経細胞の出力関数 f には (20) 式を用いた。学習課題は、2 種類の 6 次 AR モデルからの出力を類別する課題で、課題 A では、1 つのモデルにつき長さが 20 のパターンを 3 個、計 6 個のパターンを入力パターンの組とし、課題 B では、1 つのモデルにつき 10 個、計 20 個のパターンを入力パターンの組とした。

学習方法としては逆伝播法、慣性項を伴う逆伝播法、パラメータを逐次変化させる逆伝播法¹²⁾、および本論文で提案した高速学習法を用いた。

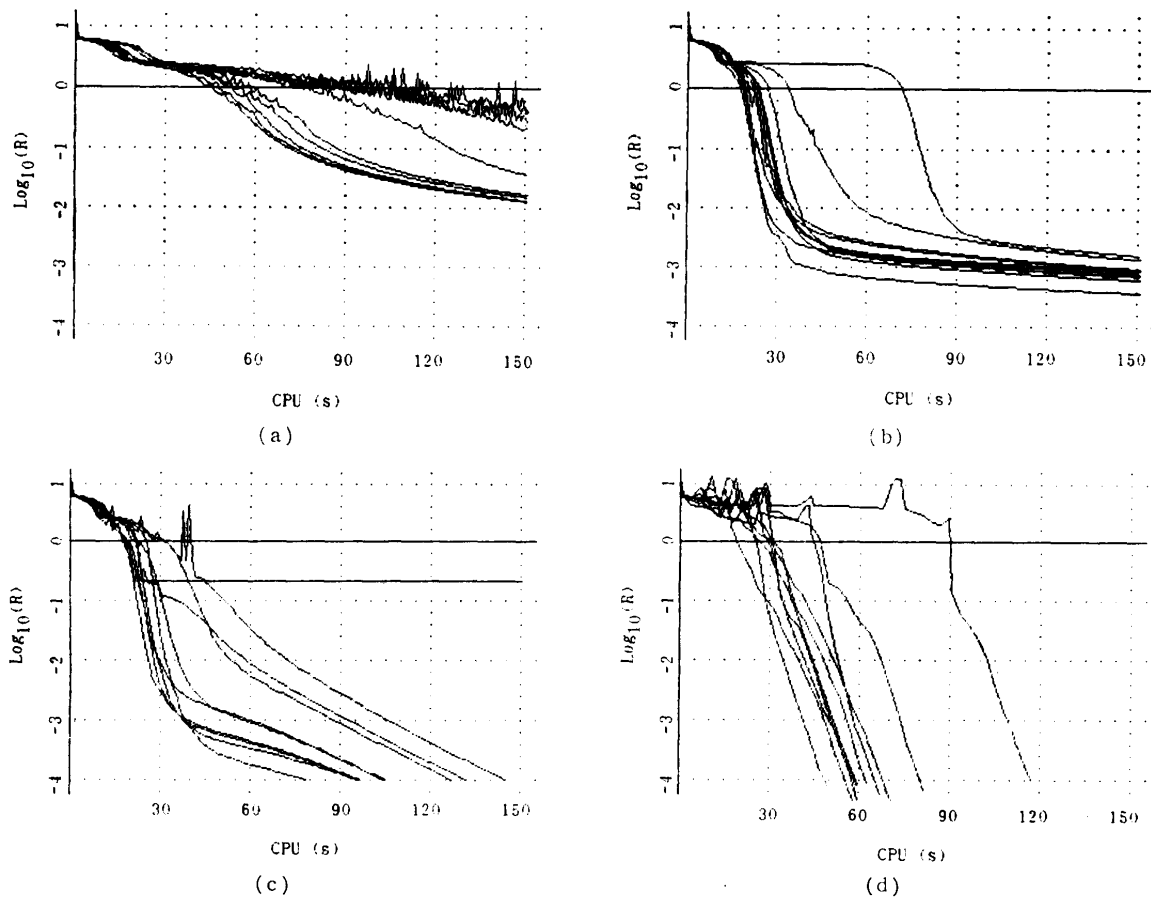


図 4 課題 A の各学習法による学習の収束の様子

- a) 逆伝播法, b) 慣性項をともなう逆伝播法, c) パラメータを逐次変化させる逆伝播法,
d) 高速学習法.

Fig. 4 Convergence behaviors for Problem A using four learning methods.

- a) the back-propagation method, b) the back-propagation method with the momentum factor, c) the back-propagation method with the successively changing learning rate and momentum factor, d) the proposed rapid learning method.

逆伝播法では学習パラメータ η を、課題Aでは 0.3 とし、課題Bでは 0.05 とした。慣性項を伴う逆伝播法では慣性パラメータ α を 0.8、学習パラメータ η を課題Aでは 0.3 とし、課題Bでは 0.05 とした。これらの方法では、誤差関数 R の値が増加したとき、2次元探索共役勾配法の手順4に示したのと同様な方法で逆戻りをする。パラメータを逐次変化させる逆伝播法は、上に示した逆戻りが起こらなかったとき、 η を ϕ 倍し、逆戻りが起こったとき、 η を β 倍しかつ一時的に α を 0 にする方法である。ここで $\phi=1.05$ 、 $\beta=0.7$ とした。また、高速学習法では再出発のパラメータ N を 100 とした。

パーソナルコンピュータ (CPU:80286, コプロセッサ:80287, クロック:10 MHz) を用い、C言語でこれらの学習方法をプログラムし実験を行った。図4, 5に10通りの初期値に対する学習の収束の様子を示す。

課題Aについてみると、逆伝播法では150秒間に

100 から 160 回 (逆戻りを含めて 166 回) の繰返し学習が行われたが、誤差は 10^{-2} 以下にならず収束がきわめて遅い。繰返し1回あたりの計算時間は0.9秒であった。学習パラメータ η を 0.3 より小さくすると逆戻りの回数は減るが収束が遅くなる。また、0.3 より大きくすると逆戻りが頻繁に起こるようになり、収束が遅くなる。慣性項をともなう逆伝播法やパラメータを逐次変化させる逆伝播法では、計算時間30秒前後、繰返し回数30回前後で誤差が 10^{-2} 以下になり、高速学習法よりも早く誤差が減少したが、その後の収束が遅い。これらの逆伝播法の繰返し1回あたりの計算時間は0.95秒であった。高速学習法では、16回から36回の繰返しで誤差が 10^{-4} 以下になり、繰返し1回あたりの計算時間は逆伝播法の約3倍の2.8秒であった。

課題Bについては、学習の過程で最適なパラメータの値が大きく異なるため、パラメータの値を固定した

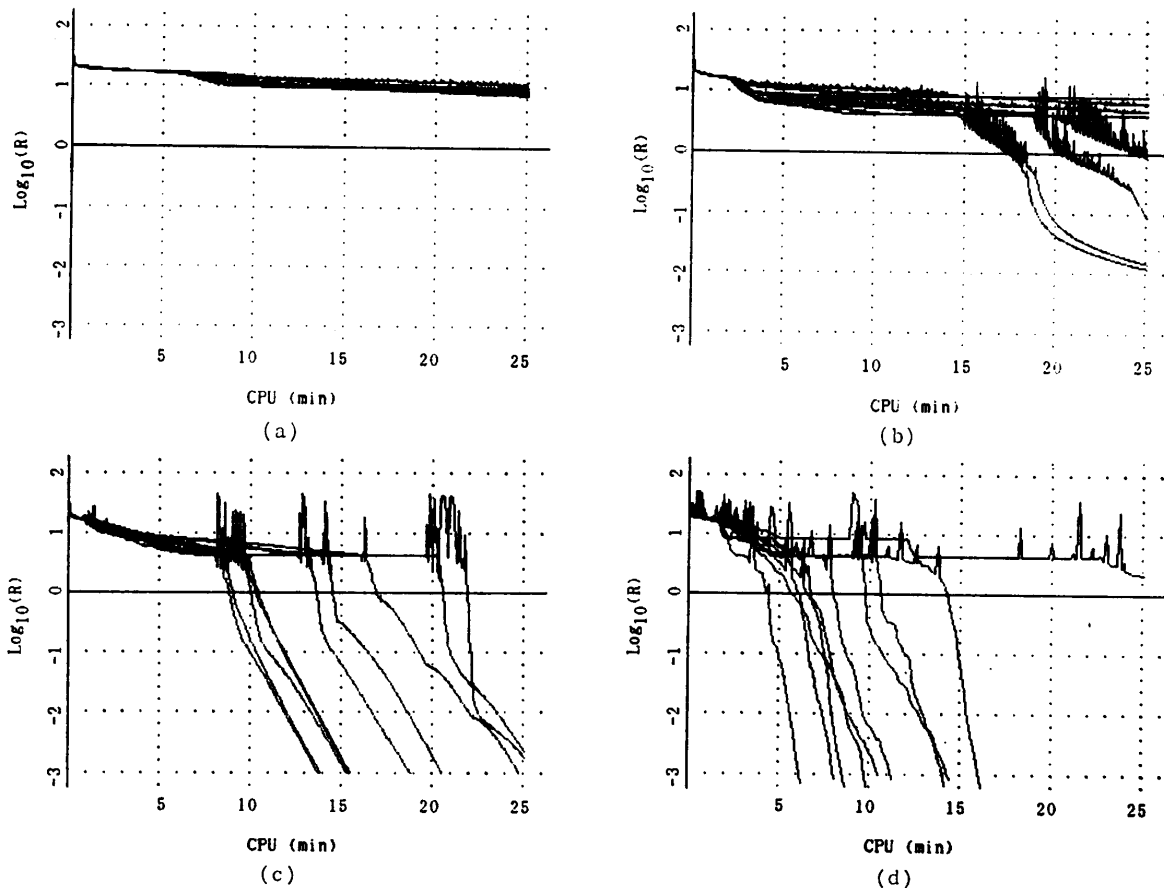


図5 課題Bの各学習法による学習の収束の様子
(a) から (d) は図4に同じ)

Fig. 5 Convergence behaviors for Problem B using four learning methods.
a)-d) are the same methods as in Fig. 4.

逆伝播法では学習が非常に遅い。慣性項をともなう逆伝播法では、1つの初期値に対して、繰返し483回、逆戻り70回、計算時間1,505秒で誤差が 10^{-2} 近くになった。パラメータを逐次変化させる逆伝播法では、繰返し276回、逆戻り27回、計算時間820秒で最も早く誤差が 10^{-3} 以下になった。繰返し1回あたりの計算時間は2.7秒であった。一方、高速学習法では、繰返し40回、逆戻り6回、計算時間375秒から、繰返し98回、逆戻り30回、計算時間970秒までの範囲で誤差が 10^{-3} 以下になった。繰返し1回あたりの計算時間は逆伝播法の約3倍の9.0秒であった。

6. おわりに

多層神経回路網の学習に2次元探索共役勾配法を用いる高速学習法を提案した。この方法は、逆伝播法における学習パラメータと慣性パラメータを自動的に制御する方法である。本方法は、自動微分法を用いて勾配ベクトル、ベクトルと勾配ベクトルとの内積、および2次形式の値を計算しており、これらの計算に必要な記憶領域が神経回路網を表現するのに必要な記憶領域の3倍強であること、および、本方法の繰返し1回あたりの計算量が逆伝播法の高々6倍の計算量であることを示した。

実験の結果、本高速学習法では逆伝播法に比べてかなり少ない繰返し回数で学習が収束した。また、本方法の繰返し1回あたりの計算時間が、逆伝播法の約3倍であった。

パラレル計算機を用いることによって、本方法の逆伝播法に対する繰返し1回あたりの計算時間の比が削減される可能性があり、さらに高速な学習が期待できる。また、2次元探索を正確に行う方法、2次以上の高次元の探索をする方法などを検討することが、今後の課題である。

謝辞 日頃ご指導いただく千葉大学工学部電気電子基礎講座、計算機工学講座、数理科学講座の皆様にご感謝いたします。

参 考 文 献

- 1) Wengert, R. E.: A Simple Automatic Derivative Evaluation Program, *Comm. ACM*, Vol. 7, No. 8, pp. 463-464 (1964).
- 2) Rall, L. B.: Automatic Differentiation: Techniques and Applications, *Lecture Notes in Computer Science*, Vol. 120, Springer-Verlag, Berlin (1981).
- 3) Baur, W. and Strassen, V.: The Complexity

of Partial Derivatives, *Theor. Comput. Sci.*, Vol. 22, pp. 317-330 (1983).

- 4) Kim, K. V., Nesterov, Yu. E., Skokov, V. A. and Cherkasskii, B. V.: An Efficient Algorithm for Computing Derivatives and Extremal Problems, *Ekonomika i matematicheskie metody*, Vol. 20, No. 2, pp. 309-318 (1984).
- 5) Iri, M.: Simultaneous Computation of Functions Partial Derivatives and Estimates of Rounding Errors—Complexity and Practicality, *Jpn. J. Appl. Math.*, Vol. 1, No. 2, pp. 223-252 (1984).
- 6) Sawyer, J. W., Jr.: First Partial Differentiation by Computer with an Application to Categorical Data Analysis, *American Statistician*, Vol. 38, No. 4, pp. 300-308 (1984).
- 7) 岩田憲和: 偏導関数計算の自動化, 東京大学大学院工学系研究科情報工学専門課程修士論文 (1984).
- 8) 久保田光一, 伊理正夫: 高速自動微分法の定式化の試みと利用のためのシステム, 統計数理研究所昭和61年度共同研究報告書, Vol. 61-共会-14, pp. 154-163 (1987).
- 9) 久保田光一, 伊理正夫: 高速自動微分法の定式化と計算複雑度の解析, 情報処理学会論文誌, Vol. 29, No. 6, pp. 551-560 (1988).
- 10) 吉田利信: 偏導関数自動導出システム, 情報処理学会論文誌, Vol. 30, No. 7, pp. 799-806 (1989).
- 11) Rumelhart, D. E., Hinton, G. E. and Williams, R. J.: Learning Representations by Back-Propagating Errors, *Nature*, Vol. 323, pp. 533-536 (1986).
- 12) Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T. and Alcon, D. L.: Accelerating the Convergence of the Back-Propagation Method, *Biological Cybernetics*, Vol. 59, pp. 257-263 (1988).
- 13) Yoshida, T.: Conjugate Gradient Method with Two-Dimensional Search Using Automatic Differentiation, *Journal of Optimization Theory and Applications*, in submission.

(平成元年8月31日受付)

(平成2年4月17日採録)



吉田 利信 (正会員)

1951年生。1978年東京大学大学院博士課程修了(計数工学)。工学博士。千葉大学工学部電気電子工学科助手。言語理解システム、自動微分システムおよび神経回路網の研究に従事。電子情報通信学会、日本音響学会、神経回路学会、日本応用数理学会各会員。