2層ニューラルネットワークの一括学習アルゴリズム

山本　祥弘

知能情報工学科

(1996年8月28日受理)

On Batch Learning Algorithms of Two Layered Neural Network

by

Yoshihiro YAMAMOTO

Department of Information and Knowledge Engineering

(Received August 28, 1996)

Recently, the author proposed a new learning algorithm (EBP) derived algebraically which is composed of the error back propagation to give a fictitious teacher to each hidden layer and the update rule of each connections. The recurrent type of the algorithm has already been examined for XOR problem by using an orthonormalization method.

Two algorithms are proposed in this paper for two layered neural network and are examined their effectiveness for majority functions.

Some discussions of logic functions are also presented to give an another perspective of nonlinear unit of a neural network

Key words : Batch learning algorithm, Error back propagation, Neural network, Logic function

１.はじめに

ニューラルネットワークの研究は、Rumelhartらにより誤差逆伝搬法（ＢＰ法）[1]が1986年に発表されて以来急激な発展を遂げ、文字・音声認識、画像処理など様々な分野[2]に応用されてきた。近年は、制御問題への応用[3]も議論されている。ところでＢＰ法にも多くの問題点が指摘されており、一つはネットワークの構成を如何にするかに、他の一つはＢＰ法の改良にその努力が払われている。しかしながら、学習アルゴリズムをどのように改良しても、勾配法に基づく限りは根本的解決にはならないものと考えている。

そこで最近、筆者は多層ニューラルネットワーク（ＮＮ）にたいする新しい学習アルゴリズムとして、勾配法ではなく代数学的に導かれるＥＢＰ法[4]−[6]を提案している。それは次の二つから構成される。

A）各中間層への仮の教師信号を、ＮＮの出力誤差から順に決定する（誤差逆伝搬法）。

B）与えられた仮の教師信号（出力層は真の教師信号）を零とするように重みパラメータを（入力層から順に）決定する（重み修正法）。

A)の手順は、ＢＰ法が微分係数の逆伝搬であるのに対し、ＥＢＰ法は誤差そのものを逆伝搬しており、文字どおり誤差逆伝搬となっている。またB)の手順は形式的には、各層それぞれ独立に行われるが、具体的にどのような方法をとるかによって、ニューラルネット全体の学習の成否に影響する。

さらに用いるデータに応じて以下のような分類が可能である。

1）各ステップで入力した結果得られるデータのみを利用可能とする方法。これは従来の逐次修正法と対応しており、その学習速度はもっとも遅い。

2）各ステップで得られるデータのみでなく過去のデータも利用可能とする方法。この方法の一つとして、正規直交化法を用いた学習法が有効であり、さらに改良を現在検討中である。

3）各ステップで重み一定のまま複数の入力の組を用いる方法（部分一括法）。これは従来の一括法をその特別な場合とするものであり、一括するデータの量を可変とするのが特徴である。

本論では、3)の部分一括法にたいする有効なアルゴリズムの開発を目指して、ＮＮを２層に限定した結果を報告する。さらにその結果の考察から、多数決関数、パリテイ検査の論理関数が、ニューロンの拡大解釈から２層ＮＮ（回路）で簡単に実現できることを示す。

２.ニューラルネットワーク

いま、重みが$\mathbf{w}_{p-1}=(w_{0p-1}, w_{1p-1}, \ldots, w_{Np-1})^T$、であるとき、ｐ番目の学習データ$\mathbf{b}_p$に対する応答を

$$c_p = f(z_p), \quad z_p = \mathbf{w}_{p-1}^T \mathbf{b}_p \tag{1}$$

とする。また、学習データ$\mathbf{b}_p$に対応する教師信号をd_pとする。ここにw_0はしきい値であり、対応する入力b_0は$b_0 = 1$とする。この入出力データを重み一定のままＭ組一括するために以下の記号を導入する。

$$\mathbf{c}_p = f(\mathbf{z}_p), \quad \mathbf{z}_p = \mathbf{w}_{p-1}^T \mathbf{B}_p \tag{2}$$

$$\mathbf{c}_p = (c_p, c_{p+1}, \ldots, c_{p+M-1})$$

$$\mathbf{z}_p = (z_p, z_{p+1}, \ldots, z_{p+M-1})$$

$$\mathbf{B}_p = (\mathbf{b}_p, \mathbf{b}_{p+1}, \ldots, \mathbf{b}_{p+M-1})$$

$$\mathbf{d}_p = (d_p, d_{p+1}, \ldots, d_{p+M-1})$$

このとき学習の目的は

$$\mathbf{d}_p = \mathbf{c}_p \tag{3}$$

あるいは等価な

$$f^{-1}(\mathbf{d}_p) = \mathbf{z} = \mathbf{w}^T \mathbf{B}_p \tag{4}$$

を満たすように重みを修正することである。その結果重みの修正量$\Delta \mathbf{w}_{p-1}$は

$$\Delta \mathbf{w}_{p-1} = (\mathbf{B}_p \mathbf{B}_p^T)^{-1} \mathbf{B}_p \mathbf{e}_p^T \tag{5}$$

$$\mathbf{e}_p = f^{-1}(\mathbf{d}_p) - \mathbf{z}_p \tag{6}$$

あるいは

$$\mathbf{w}_p = (\mathbf{B}_p \mathbf{B}_p^T)^{-1} \mathbf{B}_p f^{-1}(\mathbf{d}_p)^T \tag{7}$$

となる[4]−[7]。ただし、$M \geq N+1$とし、$\mathbf{B}_p \mathbf{B}_p^T$の正則性を仮定している。(7)式は$\mathbf{w}_{p-1}$に依存していないことに注意されたい。また、$M < N+1$のときは

$$\Delta \mathbf{w}_{p-1} = \mathbf{B}_p (\mathbf{B}_p^T \mathbf{B}_p)^{-1} \mathbf{e}_p^T \tag{8}$$

$$\mathbf{w}_p = \mathbf{w}_{p-1} + \Delta \mathbf{w}_{p-1} \tag{9}$$

を用いることになる。これらの結果は次のように導かれる。すなわち、(4)式から

$$f^{-1}(\mathbf{d}_p) = \mathbf{z}_p = (\mathbf{w}_{p-1} + \Delta \mathbf{w}_{p-1})^T \mathbf{B}_p \tag{10}$$

となり、$\Delta \mathbf{w}_{p-1}$は

$$B_p^T \varDelta w_{p-1} = (f^{-1}(d_p) - B_p^T w_{p-1})^T = e_p^T \quad (11)$$

を満たすことが必要となる。これは連立方程式を解くこととなり、条件数が未知数よりも多い場合には、最小2乗の意味での解[7),8)]となっている。

以上の準備で次のアルゴリズムを提案する。

［アルゴリズム1］ M＝N＋1と固定し、p＝1,2,..と順次実行する。

このアルゴリズムの意味は、(4)式がパラメータｗに関するN＋1次元空間の一つの超平面であり、それらのN＋1個の超平面の交点を順次探索していくことを示している。これらの交点をすべて調べ上げれば、必ず解の一つに到達するからである。

一方、M＝N＋1と固定したアルゴリズム1とは違って、すべてのデータの組を対象とするアルゴリズムとして、次の方法が有効であると考えられる。

［アルゴリズム2］ p＝1とし全入力データにたいして(7)式からw_1を求める。以降は、(6)式の要素を零としない入力のみをまとめてB_pとし、(5)式または(7)式から新しい重みw_pを求め、以下繰り返す。ただし、(3)あるいは(4)式は等式拘束条件となっているが、実際は不等式であるので、(6)式の要素e_pの零判定には

$$f^{-1}(d_p)e_p \leqq 0 \rightarrow e_p = f^{-1}(d_p) - z_p = 0 \quad (12)$$

を用いるものとする。

アルゴリズム2をFig. 1の平面で解釈する。ここに線分PQRSの上方が解領域とする。3つの直線を拘束条件とするので、(7)式の結果重みは点Aに移動する。このとき2つの直線PQとSRは不等式としては満たしているので次は直線QRで表される条件にたいしてのみ実行すると点Bの解が得られる。

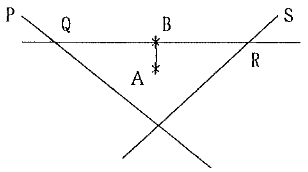

Fig. 1 Interpretation of the algorithm 2

通常、学習データの組はN＋1個よりも多いとしてもよく、従って、少なくとも最初は、(5),(6)式あるいは等価な(7)式を用いることになるので アルゴリズム2はｗの初期値には無関係となる。これは、非常に大きな特徴である。

3．多数決関数

前節で示したアルゴリズムの具体的方法とその有効性を調べるために、線形分離可能と知られている多数決関数を例題とする。その入力数をNとするとき、通常はNを奇数とするが、ここでは2以上の整数とする。ただし、偶数のNに対しては、切り下げ型、3値型、切り上げ型に分類して行った。例えば、N＝2の切り下げ型はAND回路であり、切り上げ型はOR回路である。また3値型は、同数のときに0.5を出力させるものとする。数値実験の結果、N＝2,3,4,..いずれにたいしても基本的に同じ結果であるので、以下N＝3の場合を具体的に示す。

3.1 ニューラルネットワークによる解法

N＝3であるので(1)式は

$$c = f(z) \quad (13a)$$

$$z = w_0 b_0 + w_1 b_1 + w_2 b_2 + w_3 b_3 \quad (13b)$$

である。ここに変数pは省略している。また$w_0 b_0$はしきい値であり、$b_0 = 1$とする。(13a)式の非線形関数はシグモイド関数である。

いま、重みの初期値は何でもよいが、例えばすべて

Table1 : Initial State of Majority function of N＝3

w0＝1.0,			w1＝1.0,	w2＝1.0,		w3＝1.0
b1	b2	b3	z	d	$f^{-1}(d)$	$f^{-1}(d)-z$
0.0	0.0	0.0	1.0	0.1	−2.19722	−3.19722
1.0	0.0	0.0	2.0	0.1	−2.19722	−4.19722
0.0	1.0	0.0	2.0	0.1	−2.19722	−4.19722
0.0	0.0	1.0	2.0	0.1	−2.19722	−4.19722
1.0	1.0	0.0	3.0	0.9	2.19722	−0.80278
1.0	0.0	1.0	3.0	0.9	2.19722	−0.80278
0.0	1.0	1.0	3.0	0.9	2.19722	−0.80278
1.0	1.0	1.0	4.0	0.9	2.19722	−1.80278

1とする。このときのＮＮの各値をTable 1に示す。$p=1$として最初の4組の入力ベクトルを並べたものをB_1とすると、

$$B_1=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&1&1&1\end{bmatrix}\qquad(14)$$

である。$M=N+1$でB_pが正則のとき、(5)式は

$$\Delta \mathbf{w}_{p-1}=(\mathbf{B}_p{}^T)^{-1}\mathbf{e}_p{}^T\qquad(5')$$

と簡単化される。その結果

$$\Delta \mathbf{w}_0=\begin{bmatrix}-1.0\\-1.0\\-1.0\\-3.19722\end{bmatrix},\quad \mathbf{w}_1=\begin{bmatrix}0.0\\0.0\\0.0\\-2.19722\end{bmatrix}\qquad(15)$$

が得られる。$\mathbf{w}_1$の値は(7)式を簡単化した

$$\mathbf{w}_p=(\mathbf{B}_p{}^T)^{-1}\mathbf{f}^{-1}(\mathbf{d}_p)^T\qquad(7')$$

から直接求めることができる。この$\mathbf{w}_1$の値に対するＮＮの状態をTable 2 - 1に示す。次に$p=2$として

$$B_2=\begin{bmatrix}1&0&0&1\\0&1&0&1\\0&0&1&0\\1&1&1&1\end{bmatrix}\qquad(16)$$

とすると、同様な計算からTable 2 - 2の結果を得る。

Table 2 - 1 : N = 3、Algorithm 1、p = 1

w0 = −2.19722,	w1 = w2 = w3 = 0.0					
b1	b2	b3	z	d	$f^{-1}(d)$	$f^{-1}(d)-z$
0.0	0.0	0.0	−2.19722	0.1	−2.19722	0.0
1.0	0.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	1.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	0.0	1.0	−2.19722	0.1	−2.19722	0.0
1.0	1.0	0.0	−2.19722	0.9	2.19722	4.39444
1.0	0.0	1.0	−2.19722	0.9	2.19722	4.39444
0.0	1.0	1.0	−2.19722	0.9	2.19722	4.39444
1.0	1.0	1.0	−2.19722	0.9	2.19722	4.39444

Table 2 - 2 : N = 3、algorithm 1、p = 2

w0 = −6.59166,	w1 = w2 = w3 = 4.39444					
b1	b2	b3	z	d	$f^{-1}(d)$	$f^{-1}(d)-z$
0.0	0.0	0.0	−6.59166	0.1	−2.19722	4.39444 = 0
1.0	0.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	1.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	0.0	1.0	−2.19722	0.1	−2.19722	0.0
1.0	1.0	0.0	2.19722	0.9	2.19722	0.0
1.0	0.0	1.0	2.19722	0.9	2.19722	0.0
0.0	1.0	1.0	2.19722	0.9	2.19722	0.0
1.0	1.0	1.0	6.59166	0.9	2.19722	−4.39444 = 0

Table 2 - 2の最初と最後の行の右端の0は(12)式の意味での0であり、与えられた問題の不等式を満たしていることになる。以上がアルゴリズム1の計算法である。学習回数(収束回数)は用いる入力データの組の順列に影響されるが、この例題の場合はすべて2、3回で終了している。

次にアルゴリズム2の方法で同じ問題を解いてみる。$p=1$のときは全入力の組を対象とすることになる。すなわち、$M=2^3=8$として、(5)式あるいは(7)式をもちいることになる。そこでB_1を

$$B_1=\begin{bmatrix}0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&1\\0&0&0&1&0&1&1&1\\1&1&1&1&1&1&1&1\end{bmatrix}\qquad(17)$$

と定めると、

$$\Delta \mathbf{w}_0=\begin{bmatrix}1.19722\\1.19722\\1.19722\\-4.29583\end{bmatrix},\quad \mathbf{w}_1=\begin{bmatrix}2.19722\\2.19722\\2.19722\\-3.29583\end{bmatrix}\qquad(18)$$

なる値が得られ、このときのＮＮの値がTable 3 - 1である。(12)式の条件を考慮すると、$p=2$に対

Table 3 - 1 : N = 3、Algorithm 2、p = 1

w0 = −3.29582,	w1 = w2 = w3 = 2.19722					
b1	b2	b3	z	d	$f^{-1}(d)$	$f^{-1}(d)-z$
0.0	0.0	0.0	−3.29583	0.1	−2.19722	1.09861 = 0
1.0	0.0	0.0	−1.09861	0.1	−2.19722	−1.09861
0.0	1.0	0.0	−1.09861	0.1	−2.19722	−1.09861
0.0	0.0	1.0	−1.09861	0.1	−2.19722	−1.09861
1.0	1.0	0.0	1.09861	0.9	2.19722	1.09861
1.0	0.0	1.0	1.09861	0.9	2.19722	1.09861
0.0	1.0	1.0	1.09861	0.9	2.19722	1.09861
1.0	1.0	1.0	3.29583	0.9	2.19722	−1.09861 = 0

してM＝6となり、従って

$$B_2=\begin{bmatrix}1&0&0&1&1&0\\0&1&0&1&0&1\\0&0&1&0&1&1\\1&1&1&1&1&1\end{bmatrix} \quad (19)$$

とおく。このときも$M \geqq N+1$であるので、やはり、(5)式または(7)式を用いる。その結果

$$\Delta \mathbf{w}_1=\begin{bmatrix}2.19722\\2.19722\\2.19722\\-3.29583\end{bmatrix},\quad \mathbf{w}_2=\begin{bmatrix}4.39444\\4.39444\\4.39444\\-6.59166\end{bmatrix} \quad (20)$$

として一つの解が得られた。同様にしてN＝2、4、5にたいして求めた結果がTable 4である。

Table 4 : Weights of majority function

N	型	w_0	$w_1 \sim w_N$
2	切下げ型	−6.59166	4.39444
	3 値 型	−2.19722	2.19722
	切上げ型	−2.19722	4.39444
3		−6.59166	4.39444
4	切下げ型	−10.9861	4.39444
	3 値 型	−4.39444	2.19722
	切上げ型	−6.59166	4.39444
5		−10.9861	4.39444

3.2 簡便法

多数決関数は線形分離可能であるので、以下のようにTable 4はより簡単に求めることができる。例えば、N＝3のとき$w=w_1=w_2=w_3$として

$$(0+0+1)w+w_0=f^{-1}(0.1)=-2.19722 \quad (21a)$$

$$(0+1+1)w+w_0=f^{-1}(0.9)=+2.19722 \quad (21b)$$

を解けばよい。またN＝4のとき切下げ型に対し、

$$(0+0+1+1)w+w_0=f^{-1}(0.1)=-2.19722 \quad (22a)$$

$$(0+1+1+1)w+w_0=f^{-1}(0.9)=+2.19722 \quad (22b)$$

3値型に対し

$$(0+0+1+1)w+w_0=f^{-1}(0.5)=0.0 \quad (23a)$$

$$(0+1+1+1)w+w_0=f^{-1}(0.9)=+2.19722 \quad (23b)$$

切上げ型に対し

$$(0+0+0+1)w+w_0=f^{-1}(0.1)=-2.19722 \quad (24a)$$

$$(0+0+1+1)w+w_0=f^{-1}(0.9)=+2.19722 \quad (24b)$$

Table 3-2 : N＝3、Algorithm 2、p＝2

w0＝−6.59166, w1＝w2＝w3＝4.39444						
b1	b2	b3	z	d	$f^{-1}(d)$	$f^{-1}(d)-z$
0.0	0.0	0.0	−6.59166	0.1	−2.19722	4.39444＝0
1.0	0.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	1.0	0.0	−2.19722	0.1	−2.19722	0.0
0.0	0.0	1.0	−2.19722	0.1	−2.19722	0.0
1.0	1.0	0.0	2.19722	0.9	2.19722	0.0
1.0	0.0	1.0	2.19722	0.9	2.19722	0.0
0.0	1.0	1.0	2.19722	0.9	2.19722	0.0
1.0	1.0	1.0	6.59166	0.9	2.19722	−4.39444＝0

をそれぞれ解けばよい。その結果一般に

N:奇数：$w=4.39444$、$w_0=-2.19722N$

N:偶数：$w=4.39444$、$w_0=-2.19722(N+1)$

$w=2.19722$、$w_0=-1.09861N$

$w=4.39444$、$w_0=-2.19722(N-1)$ (25)

となる。偶数は上から切下げ、3値、切上げ型である。ここに3値とは、0〜Nの中間点N/2にたいして0.5を出力させるものとする。この方法は、N＋1次元のパラメータ空間において、条件$w_1=w_2=\ldots=w_N$により制限した2次元平面上で、隣接する独立な2条件から、解の一組を求めている。

4. 論理関数の実現

前節ではNNの重みを求めてきたが、ここでは重み$w_1=\cdot\cdot=w_N=1$とし、非線形関数fを工夫することにより論理関数の実現を計る。いま

$$f(z;k,m)=\frac{1}{1+\exp(-m(z-k))} \quad (26)$$

とする。

4.1 多数決関数：このとき

$$f(z)=f(z;k,m) \quad (27)$$

とし、kを0〜Nの適当な中間点、mを充分大とすればよい。N＝2のときAND回路やOR回路が実現できる。

4.2 矩形関数：これは入力の総和が定められた区間(a, b)の間だけ1を出力させるものであり、

$$f(z)=f(z;a,m)-f(z;b,m) \quad (28)$$

とし、mを充分大とすればよい。例えばN＝２、a＝0.5、b＝1.5 とすればＸＯＲ回路となる。数値例としてはTable５である。

Table５　Numerical values of XOR-problem

	m＝1	m＝5	m＝10	m＝15	m＝20
f (0)	0.1951	0.0753	0.0067	0.0006	0.0000
f (1)	0.2449	0.8453	0.9866	0.9989	1.0000
f (2)	0.1951	0.0753	0.0067	0.0006	0.0000

4.3 ３値関数(階段関数)：中間値0.5を区間(ａ、ｂ)で出力させるものとして

$$f(z)=0.5f(z\ ;\ a,m)+0.5f(z\ ;\ b,m) \quad (29)$$

とすればよい。さらに高段にも拡張可能である。例として、ａ＝ｎ－1.5、ｂ＝ｎ＋1.5の結果をTable６に示す。

Table６　Stair function

	m＝1	m＝5	m＝10	m＝15	m＝20
f (n − 3)	0.0967	0.0002	0	0	0
f (n − 2)	0.2034	0.0379	0.0033	0.0002	0.0000
f (n − 1)	0.3491	0.4620	0.4966	0.4997	0.4999
f (n)	0.5	0.5	0.5	0.5	0.5
f (n + 1)	0.6508	0.5379	0.5033	0.5002	0.5000
f (n + 2)	0.7965	0.9620	0.9966	0.9997	0.9999
f (n + 3)	0.9032	0.9997	1	1	1

4.4 パリティ関数：一般的には

$$f(z)=\Sigma_{k}\pm f(z\ ;\ k,m) \quad (30)$$

である。これは先の矩形関数を並べたものに相当し、パリティ関数としては、ｋ＝0.5, 1.5, 2.5,.. とすればよい。例として、Ｎ＝６の場合をTable７に示す。

Table７　Parity-check function of N＝6

	m＝1	m＝5	m＝10	m＝15	m＝20
f (0)	0.2486	0.0753	0.0067	0.0006	0.0000
f (1)	0.3698	0.8488	0.9866	0.9989	0.9999
f (2)	0.4368	0.1506	0.0134	0.0011	0.0001
f (3)	0.4581	0.8494	0.9866	0.9989	0.9999
f (4)	0.4368	0.1506	0.0134	0.0011	0.0001
f (5)	0.3698	0.8488	0.9866	0.9989	0.9999
f (6)	0.2486	0.0753	0.0067	0.0006	0.0000

６．おわりに

本論では、従来のＢＰ法にかわる新しい誤差逆伝搬法による学習アルゴリズムＥＢＰの一括処理法を２層ＮＮにたいし示し、多数決関数に応用した結果を示した。多層ＮＮにたいしては別に報告する。また、２層ＮＮによる論理関数の実現を、非線形関数の工夫により可能であることを示した。ただし、これらの非線形関数をニューロンと呼ぶかどうかは意見の分かれるところであるが、(27)～(30)式が示すように、シグモイド関数の線形和で構成されている。現在、シグモイド関数およびＲＢＦが良く知られているが、さらに種々の関数が有効になるのではないかと考えている。

参考文献

1) Rumelhart, D.E., McClelland, J.L. and the PDP Research Group : *Parallel Distributed Processing*, Vol.1, MIT Press (1986)

2) ニューラルネットワークの応用総合特集号：システム制御情報学会誌、35, 1 (1991)

3) Miller, W.T. et al Eds. : *Neural Network for Control*, MIT Press (1990)

4) 山本：ニューロ回路の学習規則と適応アルゴリズム、システム制御情報学会論文誌, 7, 12 533/535 (1994)

5) 山本：多層ニューロ回路の新しい誤差逆伝搬法とその代数学的性質, システム制御情報学会論文誌、9, 5, 203/211 (1996)

6) 山本：正規直交化法による多層ニューラルネットワークの学習アルゴリズム, システム制御情報学会論文誌、9, 10 (1996)

7) 山本：ある種の極値問題の解法とその適応・学習アルゴリズムへの応用, 鳥取大学工学部研究報告, 26, 1, 45/55 (1995)

8) 山本：最小二乗法の一般化による適応アルゴリズムとその液位プラントへの応用, システム制御情報学会論文誌、7, 3, 77/83 (1994)