

複素バックプロパゲーション学習†

新 田 徹†† 古 谷 立 美††

ニューラルネットワークの学習方式として提案されたバックプロパゲーション学習の応用が、近年、盛んに行われている。応用分野としては、パターン認識、人工知能、信号処理等であるが、分野によっては、たとえば、フーリエ変換などにより、その処理過程において複素数が現れることがある。そこで、本稿では、従来のニューラルネットワークの結合の重みと各ノードが持つしきい値を複素数に拡張し、複素パターンに対する自然な学習を可能にするバックプロパゲーション学習アルゴリズムを提案する。その学習収束性については、学習識別理論を複素数に拡張することによって保証される。まず、学習速度は従来の BP 学習アルゴリズムとほぼ同じであることを実験的に確かめる。次に、提案した複素 BP 学習アルゴリズムには、従来の BP 学習アルゴリズムには見られない図形変換能力があることを確認し、そのふるまいと複素解析における一致の定理との関係について述べる。一般に、実数世界で考えられていた概念などを複素数に拡張した場合、実数世界には存在しない興味深い性質が現れることがよくあるが、複素解析における一致の定理はその典型的な例である。

1. はじめに

階層型ニューラルネットワークの学習アルゴリズムとして提案されたバックプロパゲーション学習²⁾は、様々な分野に応用されている。たとえば、画像処理、音声認識などであるが、分野によってはその処理過程で複素数が現れることがある。

本稿では、従来のニューラルネットワークの結合の重みと各ノードが持つしきい値を複素数に拡張し、複素パターンに対する自然な学習を可能にするバックプロパゲーション学習アルゴリズム（以下、複素 BP 学習アルゴリズムと呼ぶ。また、従来の実数パターンに対するバックプロパゲーション学習アルゴリズムを実 BP 学習アルゴリズムと呼ぶことにする。）を提案する。その学習収束性については、甘利¹⁾による学習識別理論を複素数に拡張することによって保証される。まず、学習速度は実 BP 学習アルゴリズムとほぼ同じであることを実験的に確かめる。次に、提案した複素 BP 学習アルゴリズムには、実 BP 学習アルゴリズムには見られない図形変換能力があることを確認し、そのふるまいと複素解析における一致の定理との関係について言及する。

2. 複素バックプロパゲーション学習

2.1 複素学習識別モデル

実 BP 学習アルゴリズムは、甘利¹⁾による学習識別理論がその理論的基礎となっている。そこで、本稿で

は学習識別理論におけるモデルに複素数要素を加味したモデルを考える。まず、2つの複素パターン情報源を考え、情報源 1 から複素パターン $x \in C^n$ 、情報源 2 から複素パターン $y \in C^m$ が未知なる同時確率 $P(x, y)$ で発生するものとする ($x \in C^n$, $y \in C^m$ であることに注意。 C は複素数の全体をあらわす)。ここで、 $\{(x, y)\}$ はニューラルネットワークにおける学習パターンに対応するものであり、有限個とする。学習の目的は、情報源 1 からの複素パターン x が与えられたときに、情報源 2 からの複素パターン y を推定できるようにすることである。推定を与える関数を $z(w, x): R^p \times C^n \rightarrow C^m$ とする (R は実数の全体をあらわす)。 $w \in R^p$ は、ニューラルネットワークにおける結合の重みと各ノードのしきい値をすべてまとめたものに対応し、推定関数 z の出力値はニューラルネットワークが出力する出力パターンに対応する。また、 $r(y', y): C^m \times C^m \rightarrow R^+$ (R^+ は 0 以上の実数の全体をあらわす) を、複素パターン y を複素パターン y' で推定したときの損失関数とし、 $R(w): R^p \rightarrow R^1$ をパラメータ w を用いたときの平均損失とする。すなわち、

$$R(w) = \sum_x \sum_y^{\text{def}} r(z(w, x), y) P(x, y) \quad (1)$$

と定義する。 $R(w)$ はニューラルネットワークが出力する出力パターンと出力学習パターン（教師パターン）との誤差に対応するものであり、この値が小さいほど良い推定であることになる。

2.2 学習収束定理

本節では、2.1 節で示した複素学習識別モデルにおける学習アルゴリズムを示し、その収束性を確認する。それらは、確率的降下法¹⁾の複素数版である。離

† A Complex Back-propagation Learning by TOHRU NITTA and TATSUMI FURUYA (Computation Models Section, Computer Science Division, Electrotechnical Laboratory).

†† 電子技術総合研究所情報アーキテクチャ部計算機構研究室

散時間パラメータ n を導入し, (x_n, y_n) を時刻 n において発生する複素パターンとし, パラメータ w を

$$w_{n+1} = w_n + \Delta w_n \quad (2)$$

に従って修正するものとする. また, $R(w)$ を最小または極小ならしめるパラメータ w を最適なパラメータと定義する.

このとき, 次の定理が成り立つ.

定理

十分小さな正数 ε と正定値行列 A に対して, パラメータ w を修正量

$$\Delta w_n = -\varepsilon A \nabla r(z(w_n, x_n), y_n) \quad n=0, 1, \dots \quad (3)$$

に従って修正を行うことにより, 平均損失 R に関して最適なパラメータ w がいくらかでも良い精度で得られる (∇ は w に関する gradient).

(証明)

学習識別理論¹⁾の論理がそのまま適用できる. これは以下の理由による. 損失関数 $r(z(w, x), y): \mathbb{C}^m \times \mathbb{C}^m \rightarrow \mathbb{R}^1$ はパラメータ w に関してみれば, 学習識別理論¹⁾と同じ $\mathbb{R}^r$ 上の実数値関数であり, 学習においてはパラメータ w の変量が議論される. また, $r(z(w, x), y)$ は複素パターン $x \in \mathbb{C}^n$, $y \in \mathbb{C}^m$ に関してみると, 複素数上の関数であるが, (x, y) に注目されて直接扱われることはない. 扱われるにしても, 複素数値確率変数 (x, y) に関する期待値 $E[r]$ という形で扱われる. 一般に, 複素数値確率変数は実数値確率変数と同じようにして扱うことができる. よって, 学習識別理論¹⁾の証明の論理に影響はない.

(証明終)

2.3 複素 BP 学習モデルと学習アルゴリズムの導出

本節では, 2.1 および 2.2 節で示した複素学習識別理論を多層型ニューラルネットワークに適用する. まず, モデルの説明を行う. 3層のニューラルネットワークを考え, 各ユニットの出力関数 $f: \mathbb{C}^1 \rightarrow \mathbb{C}^1$ を

$$f(z) = \frac{1}{1 + \exp(-x)} + i \frac{1}{1 + \exp(-y)}, \quad z = x + iy \quad (4)$$

と定義する. $f(z)$ の実部, 虚部は, それぞれ, z の実部, 虚部に関するシグモイド関数であり, 明らかに, $0 \leq \text{Re}[f], \text{Im}[f] \leq 1, |f(z)| \leq \sqrt{2}$ を満たす ($\text{Re}[z]$, $\text{Im}[z]$ はそれぞれ複素数 z の実部, 虚部). ユニット間の結合の重みおよび各ユニットが持つしきい値はすべて複素数とし, w_{ji} を入力ユニット i と中間ユニッ

ト j との間の結合の重み, v_{kj} を中間ユニット j と出力ユニット k との間の結合の重み, θ_j を中間ユニット j が持つしきい値, γ_k を出力ユニット k が持つしきい値とする. また, I_i, H_j, O_k を, それぞれ, 入力ユニット i , 中間ユニット j , 出力ユニット k の出力値とし, U_j, S_k をそれぞれ中間ユニット j , 出力ユニット k の内部ポテンシャルとする. すなわち, $U_j = \sum_i w_{ji} I_i + \theta_j$, $S_k = \sum_j v_{kj} H_j + \gamma_k$, $H_j = f(U_j)$, $O_k = f(S_k)$ と定義する. また, 出力ユニット k に対する教師パターン (出力学習パターン) T_k と出力ユニット k の出力値 O_k との誤差を δ^k とする. つまり, $\delta^k = T_k - O_k$ と定義する. 最後に, パターン p に対する 2 乗誤差を $E_p = (1/2) \sum_k |T_k - O_k|^2$ と定義する.

次に, 学習アルゴリズムを導出する. $\varepsilon > 0$ を十分小さくとり, 正定値行列 A を単位行列として, 2.2 節で示した定理を適用すると, 結合の重みおよびしきい値の修正量は,

$$\Delta v_{kj} = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[v_{kj}]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[v_{kj}]} \quad (5)$$

$$\Delta \gamma_k = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[\gamma_k]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[\gamma_k]} \quad (6)$$

$$\Delta w_{ji} = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[w_{ji}]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[w_{ji}]} \quad (7)$$

$$\Delta \theta_j = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[\theta_j]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[\theta_j]} \quad (8)$$

とすれば良いことになり, 結局,

$$\Delta v_{kj} = \bar{H}_j \Delta \gamma_k \quad (9)$$

$$\Delta \gamma_k = \varepsilon \{ \text{Re}[\delta^k] (1 - \text{Re}[O_k]) \text{Re}[O_k] + i \text{Im}[\delta^k] (1 - \text{Im}[O_k]) \text{Im}[O_k] \} \quad (10)$$

$$\Delta w_{ji} = \bar{I}_i \Delta \theta_j \quad (11)$$

$$\begin{aligned} \Delta \theta_j = \varepsilon \{ & (1 - \text{Re}[H_j]) \text{Re}[H_j] \\ & \sum_k \{ \text{Re}[\delta^k] (1 - \text{Re}[O_k]) \text{Re}[O_k] \text{Re}[v_{kj}] \\ & + \text{Im}[\delta^k] (1 - \text{Im}[O_k]) \text{Im}[O_k] \text{Im}[v_{kj}] \} \\ & - i (1 - \text{Im}[H_j]) \text{Im}[H_j] \\ & \sum_k \{ \text{Re}[\delta^k] (1 - \text{Re}[O_k]) \text{Re}[O_k] \text{Im}[v_{kj}] \\ & - \text{Im}[\delta^k] (1 - \text{Im}[O_k]) \text{Im}[O_k] \text{Re}[v_{kj}] \} \} \end{aligned} \quad (12)$$

が得られる. ここで, $\bar{z}$ は複素数 z の共役複素数である.

参考のために, 実 BP 学習アルゴリズムにおけるパラメータの修正量を次に記しておく. (9)~(12)と(13)~(16)は互いに似通った形をしていることがわかる. ただし, (13)~(16)においては, δ^k, O_k, H_j, I_i ,

表 1 学習パターン
Table 1 Learning patterns.

ケース	パターン	入力ユニット										出力ユニット							
		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	...	9	
1	0		0.5	$1+i$	$0.5\ i$								i						
	1					0.5	$1+i$	$0.5\ i$						i					
2	0		$0.5\ i$	i	1								1						
	1		0.5	1	i										i				
3	0					0.5	i	$1+i$	0.5			1							
	1	0.8	$0.2\ i$	1									i						
	2		1	1	$1+0.5\ i$									1					
	3				$0.5+i$	i	1								i				

v_{kj} , γ_k , w_{ji} , θ_j はすべて実数である.

$$\Delta v_{kj} = H_j \Delta \gamma_k \quad (13)$$

$$\Delta \gamma_k = \varepsilon \delta^k (1 - O_k) O_k \quad (14)$$

$$\Delta w_{ji} = I_i \Delta \theta_j \quad (15)$$

$$\Delta \theta_j = \varepsilon (1 - H_j) H_j \sum_k \{ \delta^k (1 - O_k) O_k v_{kj} \} \quad (16)$$

2.4 学習速度

2.3 節で提案した複素 BP 学習アルゴリズムの有効性を検証するためにシミュレーションを行った. 使用したマシンは SUN 4, プログラム言語は C である.

準備として, 2つの複素パターン $z = (z_1 \dots z_n)$, $w = (w_1 \dots w_n)$, $z_k, w_k \in \mathbb{C}$ に対して, z と w との間の距離 $d(z, w)$ を

$$d(z, w) \stackrel{\text{def}}{=} \sqrt{\sum_{k=1}^n |z_k - w_k|^2} \quad (17)$$

と定義しておく.

以下に示す3つの単純なケースについて収束の様子を評価した. ここで用いた3層ネットワークは, 入力層 10 ユニット, 中間層 6 ユニット, 出力層 10 ユニットであり, 結合の重みおよびしきい値の実部, 虚部の初期値は 0 と 1 の間の乱数により設定し, 学習率は $\varepsilon = 0.5$ とした. 学習パターンは表 1 のとおりである. ただし, 空欄は $0.0 + i0.0$ を表す.

図 1(a) に学習曲線を示す. 縦軸の誤差は, 入力学習パターンに対してニューラルネットワークが出力する出力パターンと教師パターン (出力学習パターン) との間の距離を学習パターンに関して相加平均したものである. ただし, 誤差の計算において, 出力ユニット 4~9 の値は無視している. 1,000 回の学習

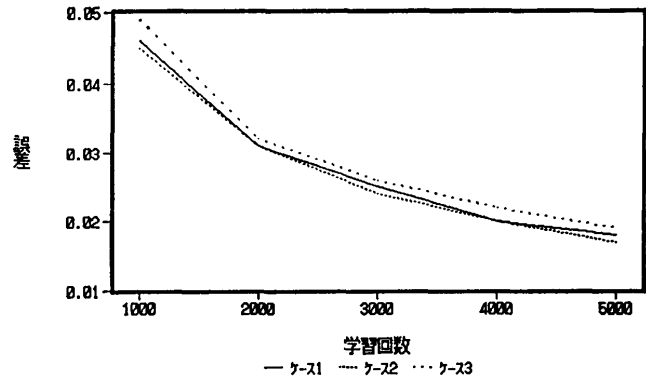


図 1(a) 複素 BP 学習曲線
Fig. 1(a) Convergence behaviors using the complex back-propagation method.

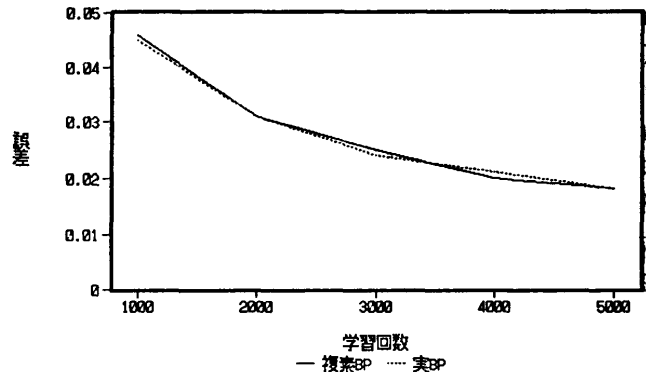


図 1(b) 学習曲線 (実 BP と複素 BP の比較)
Fig. 1(b) Convergence behaviors (a comparison between the standard back-propagation method and the complex back-propagation method).

ではほぼ収束していることがわかる.

次に, ケース 1 の学習パターンを用いて複素 BP 学習と実 BP 学習の学習速度を比較する. 実 BP 学習によって複素パターンを学習する方法の 1 つは, 入出力層において実部, 虚部を担当するユニットをあらかじめ

め決めておくことである。実 BP 学習で用いるネットワークは、複素 BP 学習で用いるネットワークの2倍の大きさとする。つまり、入力層20ユニット、中間層12ユニット、出力層20ユニットであり、入出力層のユニット0~9は実部、ユニット10~19は虚部に割り当てる。たとえば、複素数 $0.5 + i1.0$ に対して、ユニット0に0.5、ユニット10に1.0という具合に使用する。学習率等その他の条件は図1(a)の実験に同じである。図1(b)にシミュレーション結果を示す。複素 BP 学習は実 BP 学習の学習曲線に概ね沿った学習曲線を描くことがわかる。因みに、そのときの実 BP 学習と複素 BP 学習の処理速度(CPU タイム)は図2に示すようにほぼ同じである。

3. 図形変換能力

本章では、前章で示した複素 BP 学習アルゴリズムが図形変換(回転, 相似, 平行移動)を行えることを計算機実験により示し、若干の考察を行う。

実験に用いるネットワークは、入力層1ユニット、中間層6ユニット、出力層1ユニットの3層構造である。つまり、1つの複素数 z_1 を入力し、1つの複素数 z_2 を出力するものである。一般に、1つの複素数 $z = x + iy$ は2次元平面上の1点 (x, y) に対応する。よって、当該ネットワークは2次元平面上のある1点 (x, y) を他の1点 (x', y') に変換するものであると考えることができる。ここでは、2次元平面を $-1 \leq x, y \leq 1$ に制限しておく。ニューラルネットワークが出力する値は、 $0 \leq \text{Re}[z], \text{Im}[z] \leq 1$ であるが、便宜上、以下に示す図では、それらを $-1 \leq \text{Re}[z], \text{Im}[z] \leq 1$ に変換した値で示す。比較のために、実 BP 学習アルゴリズムによる実験も行ったが、そのネットワークは入力層2ユニット、中間層12ユニット、出力層2ユニットとし、2.4節と同様に複素数 z の実部 $\text{Re}[z]$ を入出力ユニット0に、虚部 $\text{Im}[z]$ を入出力ユニット1に対応させた。実験はいずれも、学習率 $\epsilon = 0.5$ 、学習回数1,000回であり、結合の重みおよびしきい値は0と1の間の乱数により設定した。

3.1 回 転

まず、直線の回転を行った(図3)。学習パターンは11個であり、直線 $y = -x + 1$ ($0 \leq x \leq 1$) 上の11個の点を入力パターンとし、それらの点を原点を中心にして正の方向(反時計回り)に90度回転させた点を出力

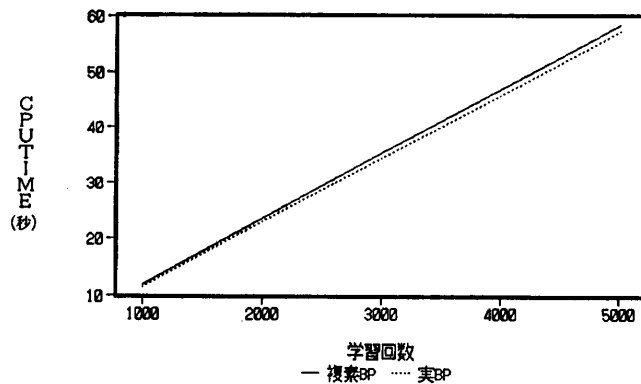


図2 処理速度

Fig. 2 CPU time (a comparison between the standard back-propagation method and the complex back-propagation method).

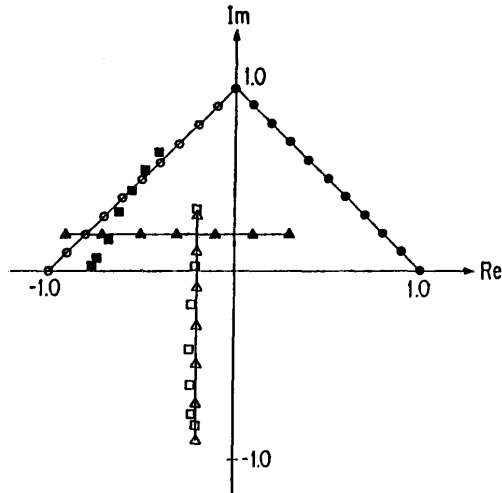
パターンとした。出力パターンは直線 $y = x + 1$ ($-1 \leq x \leq 0$) 上の点になっている。学習の後、テストパターンとして直線 $y = 0.2$ ($-0.9 \leq x \leq 0.3$) 上の7つの点(図3(a))および直線 $y = -x + 0.5$ ($0 \leq x \leq 0.5$) 上の6個の点(図3(b))を与えたところ、ニューラルネットワークはそれらをほぼ正の方向に90度回転させた点を出力した。同様にして、実 BP 学習アルゴリズムに従って実験したところ、当然ではあるが、ニューラルネットワークは教師パターン(出力学習パターン)に位置的に近い点を出力した(図3)。

次に、斜文字の回転を行った(図4, 5)。3文字からなる単語 ETL が負の方向に45度回転させられているとする。このとき、学習パターンとして、最初の1文字E上の点17個を入力パターンとし、それらの点を正の方向に45度回転したものを出力パターンとした(図4(a))。学習の後、テストパターンとして残る2文字TおよびL上の点を与えたところ、複素 BP 学習および実 BP 学習ともにニューラルネットワークはテストパターンをほぼ正の方向に45度回転させた点を出力した(図4(b), (c))。同様にして、単語 ISO について行ったところ、テストパターンSおよびOは複素 BP 学習では正の方向に45度回転したが、実 BP 学習では、出力学習パターンに位置的に近い点を出力したため、SおよびOの形状はくずれた(図5)。実 BP 学習が単語 ETL をうまく回転させることができたのは、テストパターンT, L上の点列が学習パターンE上の点列に含まれていた、位置的に近かったからであると思われる。

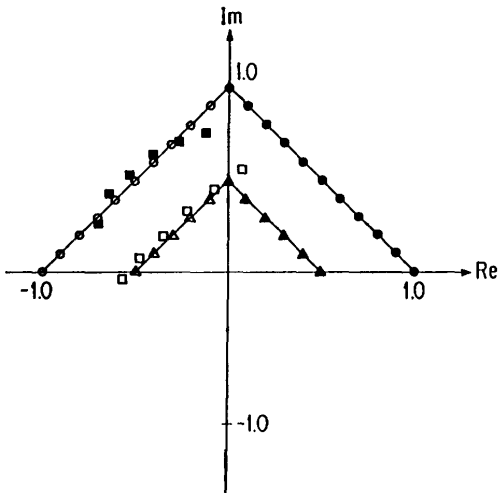
3.2 相 似 変 換

まず、縮小の例として、円 $x^2 + y^2 = 1$ から円 $x^2 +$

$y^2=(1/2)^2$ への相似変換を行った (図 6 (a)). 学習パターンは 11 個であり, 直線 $y=x$ ($0 \leq x \leq 1$) 上の 11 個の点を入力パターンとし, それらの点と原点との距離を $1/2$ にした点を出力パターンとした. 出力パターンは直線 $y=x$ ($0 \leq x \leq 0.5$) 上の点になっている. 学習の後, テストパターンとして, 円 $x^2+y^2=1$ 上の



(a) その 1
(a) No. 1.



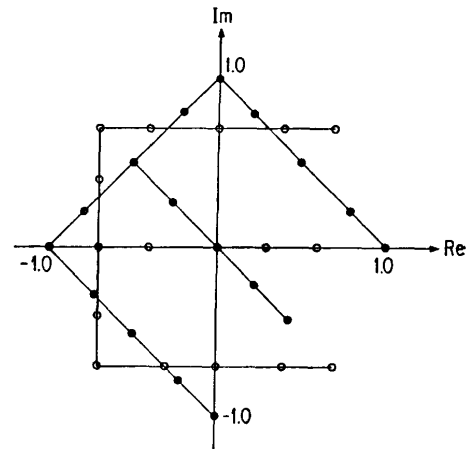
(b) その 2
(b) No. 2.

図 3 直線の回転

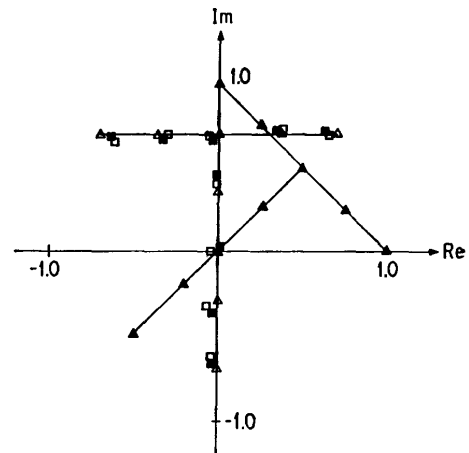
Fig. 3 Rotation of a straight line.

図 3~10 における記号の意味

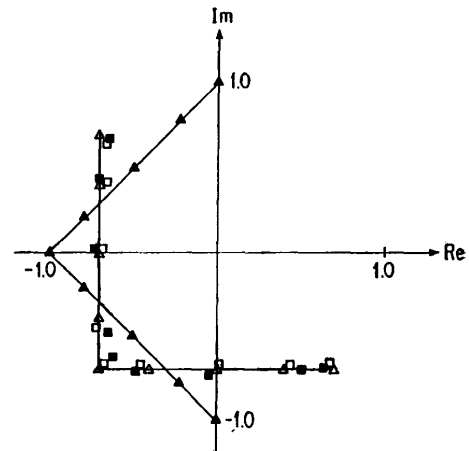
- 入学習パターン
 - 出学習パターン
 - ▲ 入力テストパターン
 - △ 期待される出力テストパターン
 - 実 BP 学習による出力テストパターン
 - 複素 BP 学習による出力テストパターン
- 添数はパターンを識別するための番号である.



(a) 学習パターン E
(a) Learning pattern E.



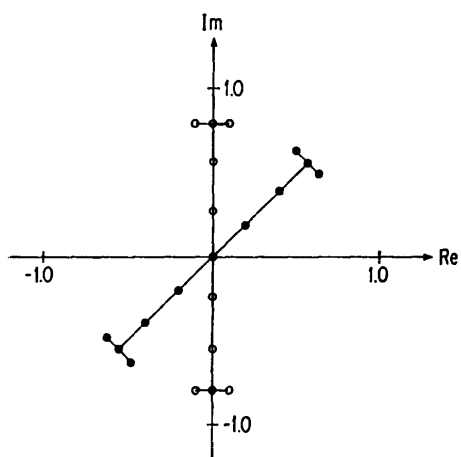
(b) テストパターン T
(b) Test pattern T.



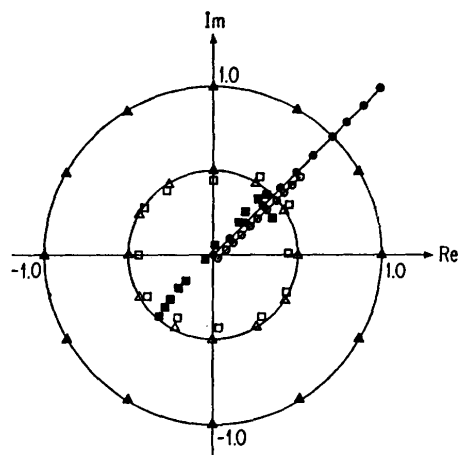
(c) テストパターン L
(c) Test pattern L.

図 4 単語 ETL の回転

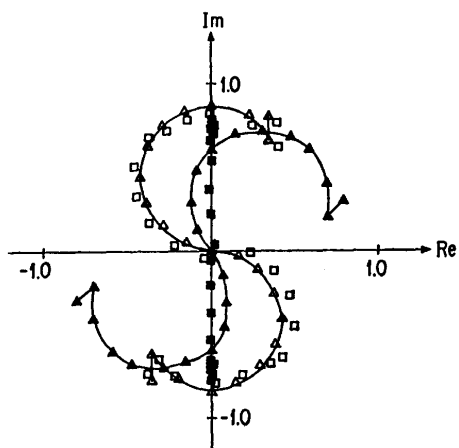
Fig. 4 Rotation of the word "ETL."



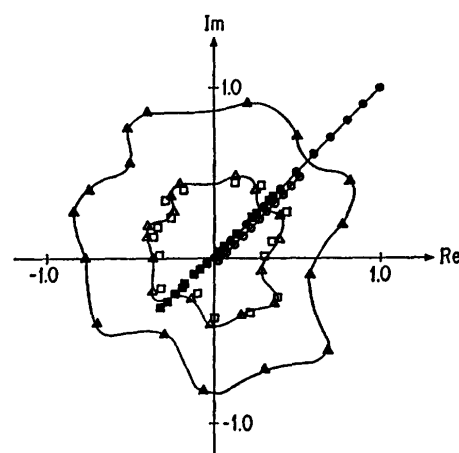
(a) 学習パターン I
(a) Learning pattern I.



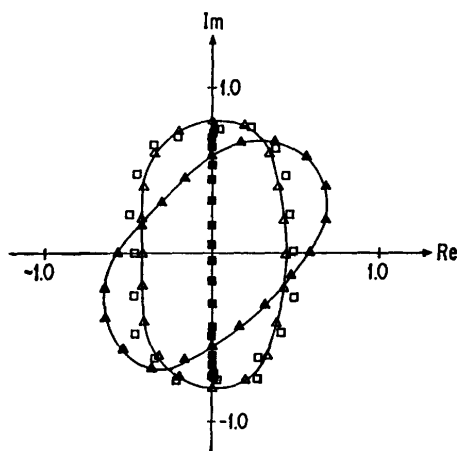
(a) 円の縮小
(a) Reduction of a circle.



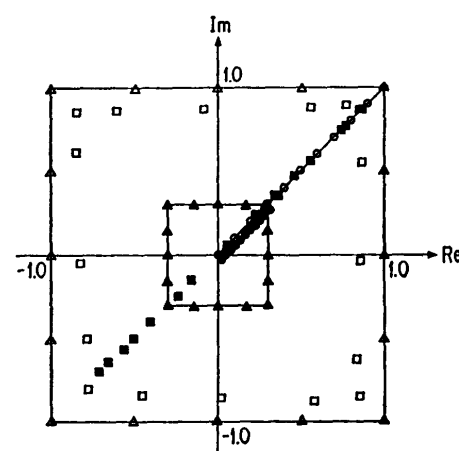
(b) テストパターン S
(b) Test pattern S.



(b) 曲線の縮小
(b) Reduction of a curve.



(c) テストパターン O
(c) Test pattern O.



(c) 正方形の拡大
(c) Enlargement of a square.

図 5 単語 ISO の回転

Fig. 5 Rotation of the word "ISO."

図 6 相似変換

Fig. 6 Similar transformation.

12個の点を与えたところ、ニューラルネットワークは円 $x^2+y^2=1$ を $1/2$ に縮小した円 $x^2+y^2=(1/2)^2$ 上の点を出力した。因みに、実 BP 学習では直線 $y=x$ 上の点を出力し、そのような縮小を行うことはできなかった。さらに、テストパターンとして任意の曲線上の点を与えた結果が図 6 (b) である。円の場合と同様に曲線がその形状を保ったまま $1/2$ に縮小されていることがわかる。

次に、学習パターンとして直線 $y=x$ ($0 \leq x \leq 0.3$) 上の 11 個の点を入力パターンとし、それらの点と原点との距離を 3.33 倍した点を出力パターンとして与えることにより、1 辺 0.3 の正方形の拡大を行った。図 6 (c) がその結果である。実 BP 学習によるニューラルネットワークは直線 $y=x$ 上の点列を出力するが複素 BP 学習によるニューラルネットワークはほぼ 1 辺 1.0 の正方形を出力する傾向がうかがえる。

3.3 平行移動

直線の平行移動を行った (図 7 (a))。学習パターンは 11 個であり、直線 $y=x+1$ ($-1 \leq x \leq 0$) 上の 11 個の点を入力パターンとし、それらの点を右下 45 度の方向に $1/\sqrt{2}$ だけ平行移動させた点を出力パターンとした。出力パターンは直線 $y=x$ ($-0.5 \leq x \leq 0.5$) に乗った点となっている。学習の後、直線 $y=x$ ($-0.5 \leq x \leq 0.5$) 上の点を与えたところ、それらの点をほぼ右下 45 度の方向に $1/\sqrt{2}$ だけ平行移動させた点を得られたが、実 BP 学習アルゴリズムによる実験においては平行移動は行われなかった。

次に、任意の図形について実験したところ、図 7 (b) に示すとおり、複素 BP 学習ではうまく平行移動させることができた。

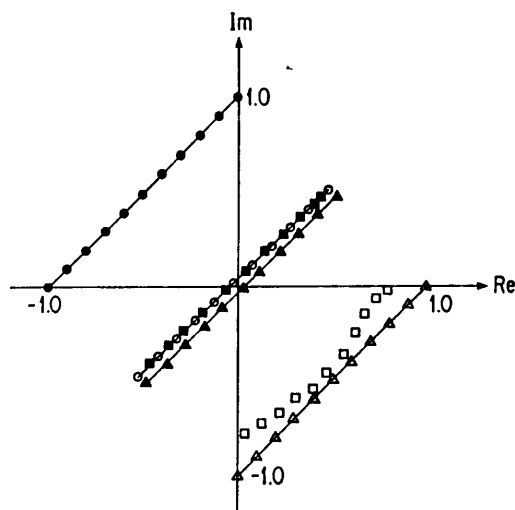
3.4 変換率の組合せ

3.1~3.3 節においては、単独の変換率 (回転角度、相似率、平行移動距離のことを変換率と呼ぶことにする) を用いた図形変換について示した。たとえば、回転については、原点を中心にして正の方向に 90 度などのように 1 つの回転角度だけを学習させ、そのふるまいについて調べた。

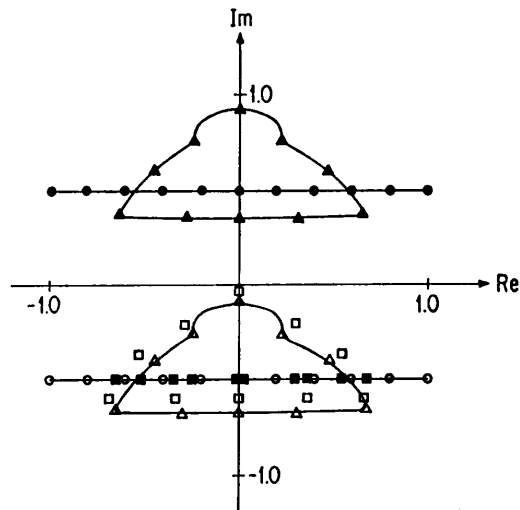
本節では、2 つの変換率を複素 BP 学習アルゴリズムにより学習させた場合のネットワークのふるまいについて調べる。なお、本節に限って学習回数は 7,000 回である。

3.4.1 回 転

異なる 2 つの直線を原点を中心にして正の方向に、それぞれ、90 度、45 度回転させるように学習させた



(a) 直線
(a) A straight line.



(b) 曲線
(b) A curve.

図 7 平行移動

Fig. 7 Parallel displacement.

(図 8). 学習パターンは 21 個であり、直線 $x=0$ ($-1 \leq y \leq 1$) 上の 21 個の点を入力パターンとした。直線 $x=0$ ($0 \leq y \leq 1$) 上の 11 個の点に対する出力パターンは、それらの点を原点を中心にして正の方向に 90 度回転させた点 (直線 $y=0$ ($-1 \leq x \leq 0$) 上にある) とし、直線 $x=0$ ($-1 \leq y \leq 0$) 上の 11 個の点に対しては、それらの点を原点を中心にして正の方向に 45 度回転させた点 (直線 $y=-x$ ($0 \leq x \leq 0.7$) 上にある) を出力パターンとした。ここで、直線 $x=0$ ($0 \leq y \leq 1$) 上の点に対する学習パターンを学習パターン 1、直線 $x=0$ ($-1 \leq y \leq 0$) 上の点に対する学習パターンを学

習パターン2と呼ぶことにする。学習の後、入力テストパターンとして、直線 $y=x$ ($-0.5 \leq x \leq 0.5$)、直線 $y=-x$ ($-0.5 \leq x \leq 0.5$) および直線 $y=0$ ($-1 \leq x \leq 1$) 上の 61 個の点を与えた。ここで、直線 $y=x$ ($0 \leq x \leq 0.5$) 上の点を入力テストパターン1、直線 $y=0$ ($0 \leq x \leq 1$) 上の点を入力テストパターン2、直線 $y=-x$ ($0 \leq x \leq 0.5$) 上の点を入力テストパターン3、直線 $y=x$ ($-0.5 \leq x \leq 0$) 上の点を入力テストパターン4、直線 $y=0$ ($-1 \leq x \leq 0$) 上の点を入力テストパターン5、直線 $y=-x$ ($-0.5 \leq x \leq 0$) 上の点を入力テストパターン6と呼ぶことにする。このとき、ニューラルネットワークは、概ね、原点を中心にして正の方向に、入力テストパターン1を45度、入力テストパターン2を67.5度、入力テストパターン3を90度、入力テストパターン4を45度、入力テストパターン5を67.5度、入力テストパターン6を90度回転させた点を出力した。ここで、入力テストパターン2および入力テストパターン5は、入力学習パターン1と入力学習パターン2とのちょうど境界線に位置しており、それらの出力テストパターンが共に入力テストパターンを67.5度($90+45$)/2度回転させた直線上の点となっていることは興味深い。また、入力テストパターン6および入力テストパターン4は、それぞれ、入力学習パターン1、入力学習パターン2に位置的に近いため、学習パターン1および2の回転角度(90度、45度)の分だけ回転している。ところが、入力テストパターン1および3は、それぞれ、入力学習パターン1(回転角度90度)、入力学習パターン2(回転角度45度)に位置的に近いにもかかわらず、入力テストパターン1は45度、入力テストパターン3は90度回転させられた。

3.4.2 相似変換

相似変換の一例として、異なる2つの直線上の点を原点との距離に関して、それぞれ、1/2、1/10に縮小させるパターンを学習させた(図9)。学習パターンは21個であり、直線 $y=x$ ($-1 \leq x$

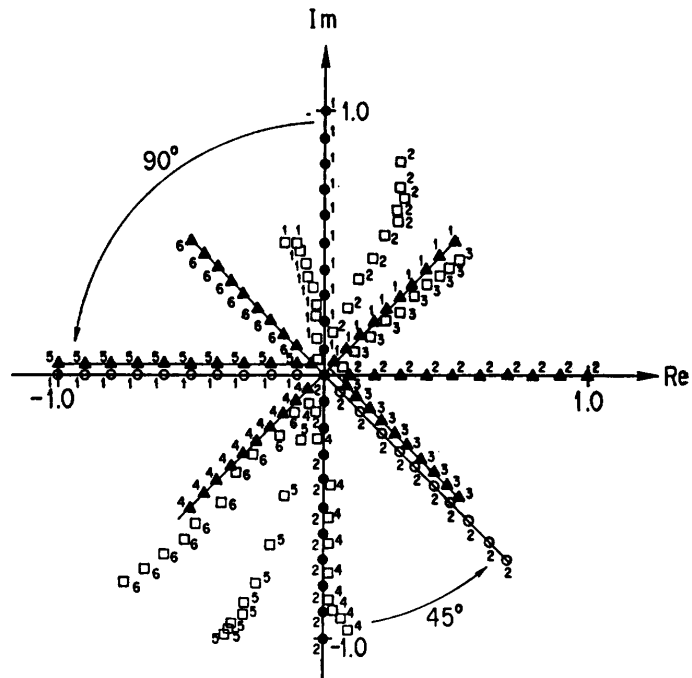


図8 回転
Fig. 8 Rotation.

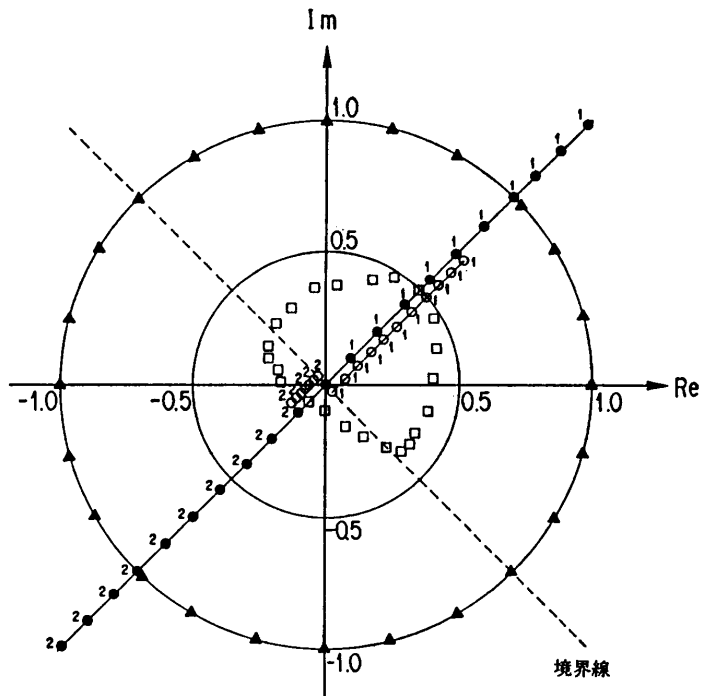


図9 相似変換
Fig. 9 Similar transformation.

≤ 1) 上の 21 個の点を入力パターンとした。直線 $y=x$ ($0 \leq x \leq 1$) 上の 11 個の点に対しては、それらの点

と原点との距離を $1/2$ にした点 (直線 $y=x$ ($0 \leq x \leq 0.5$) 上にある) を出力パターンとし, 直線 $y=x$ ($-1 \leq x \leq 0$) 上の 11 個の点に対しては, それらの点と原点との距離を $1/10$ にした点 (直線 $y=x$ ($-0.1 \leq x \leq 0$) 上にある) を出力パターンとした. ここで, 直線 $y=x$ ($0 \leq x \leq 1$) 上の点に対する学習パターンを学習パターン 1, 直線 $y=x$ ($-1 \leq x \leq 0$) 上の点に対する学習パターンを学習パターン 2 と呼ぶことにする. 学習の後, 入力テストパターンとして, 円 $x^2+y^2=1^2$ 上の 24 個の点を与えたところ, ニューラルネットワークは図 9 における口で示される点 (出力テストパターン) を出力した. 図 9 において, 出力テストパターン (口) は, 入力学習パターン 1 に近い入力テストパターンのものほど円 $x^2+y^2=(1/2)^2$ に近くなっており, 破線で示される学習パターン 1 と学習パターン 2 との境界線を越えると急激に原点との距離を縮めていく.

3.4.3 平行移動

異なる 2 つの直線を下向きに, それぞれ, 0.4, 0.8 だけ平行移動させるように学習させた (図 10). 学習パターンは 42 個であり, 直線 $y=1$ ($-1 \leq x \leq 1$) および直線 $y=0.2$ ($-1 \leq x \leq 1$) 上の 42 個の点を入力パターンとした. 直線 $y=1$ ($-1 \leq x \leq 1$) 上の 21 個の点に対する出力パターンは, それらの点を下方に 0.4 移動させた点 (直線 $y=0.6$ ($-1 \leq x \leq 1$) 上にある) とし, 直線 $y=0.2$ ($-1 \leq x \leq 1$) 上の 21 個の点に対しては, それらの点を下方に 0.8 移動させた点 (直線 $y=-0.6$ ($-1 \leq x \leq 1$) 上にある) を出力パターンとした. ここで, 直線 $y=1$ ($-1 \leq x \leq 1$) 上の点に対する学習パターンを学習パターン 1, 直線 $y=0.2$ ($-1 \leq x \leq 1$) 上の点に対する学習パターンを学習パターン 2 と呼ぶことにする. 学習の後, 入力テストパターンとして, $y=0.8$ ($-1 \leq x \leq 1$), 直線 $y=0.4$ ($-1 \leq x \leq 1$) 上の 42 個の点を与えた. ここで, $y=0.8$ ($-1 \leq x \leq 1$) 上の点を入力テストパターン 1, 直線 $y=0.4$ ($-1 \leq x \leq 1$) 上の点を入力テストパターン 2 と呼ぶことにする. このとき, ニューラルネットワークは, 入力テストパターン 1 を下方に, 0.55, 入力テストパターン 2 を下方に 0.75 だけ平行移動させた点を出力した. つまり, 入力テストパターン 1 および 2 の出力は, それ

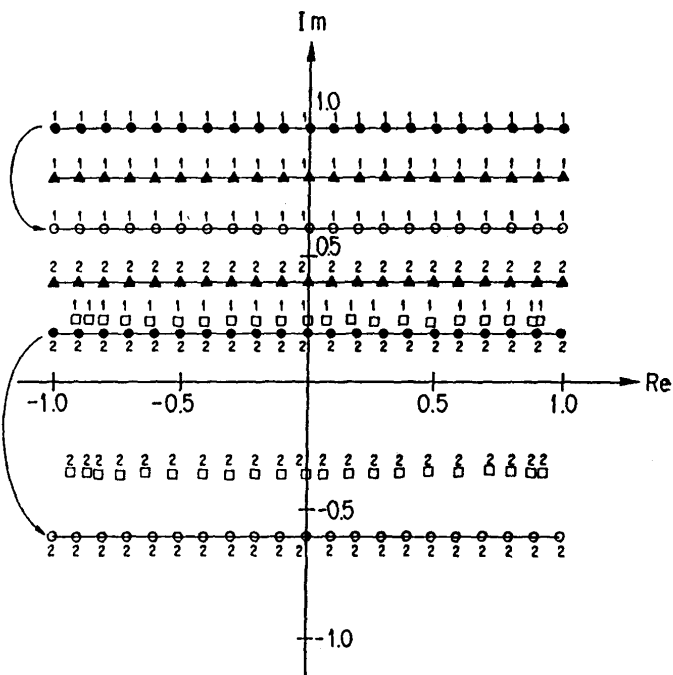


図 10 平行移動

Fig. 10 Parallel displacement.

ぞれ, 直線 $y=0.25$ ($-1 \leq x \leq 1$), $y=-0.35$ ($-1 \leq x \leq 1$) 上にほぼ乗っている. ここで, 入力テストパターン 1 は, 入力学習パターン 1 とは 0.2, 入力学習パターン 2 とは 0.6 の距離がある. また, 入力テストパターン 2 は, 入力学習パターン 1 とは 0.6, 入力学習パターン 2 とは 0.2 の距離がある. つまり, 入力テストパターン 1 は, 入力学習パターン 1 と入力学習パターン 2 を 1 対 3 に内分する直線, 入力テストパターン 2 は, 入力学習パターン 1 と入力学習パターン 2 を 3 対 1 に内分する直線である. そして, 出力テストパターン 1 は, 出力学習パターン 1 と出力学習パターン 2 をほぼ 1 対 3 に内分する直線, 出力テストパターン 2 は, 出力学習パターン 1 と出力学習パターン 2 をほぼ 3 対 1 に内分する直線となっている. つまり, 出力テストパターンは, 入力テストパターンと 2 つの入力学習パターンとの距離関係を反映したものとなっていると言える.

3.5 考察

複素 BP 学習アルゴリズムによって学習させたニューラルネットワークには, 従来の多層ニューラルネットワークには見られない図形変換能力 (回転, 相似, 平行移動) があるらしいことがわかった.

実験結果において, 正方形が完全に拡大されなかつ

たり (図 6 (c)), 平行移動により, 直線がねじ曲げられる (図 7 (a)) ことが気になるが, これらの現象はシグモイド関数の出力値が完全に 1 にならないために起こるものと考えられるので, 若干の工夫を施すことにより改善できるものと思われる.

一般に, 2次元平面上の点 (x, y) が原点を中心として正の方向に θ 度だけ回転することは, 複素数 $z_1 = x + iy$ に, 動径が 1 で偏角が θ 度の複素数 $z_2 = e^{i\theta}$ を掛け合わせることに対応する. つまり, $z_1 z_2$ は, 点 (x, y) を原点を中心として正の方向に θ 度だけ回転させた点を意味する. たとえば, 3.1 節の回転の実験では, 点をあらわす複素数 $z_1 = x + iy$ に $z_2 = e^{i90}$ または $z_2 = e^{i45}$ が掛け合わされていたことになる. また, 2次元平面上の点 (x, y) の相似変換 (縮小, 拡大) および平行移動は, それぞれ, 複素数 $z_1 = x + iy$ と実数 α との乗算, 複素数 $z_1 = x + iy$ と複素数 w との加算に対応する. 図 6 (a) に示す縮小は $\alpha = 0.5$ の場合, 図 6 (c) の拡大は $\alpha = 3.33$ の場合であり, 図 7 (a) に示す平行移動は $w = 0.5 - 0.5i$ の場合であった. 要するに, ニューラルネットワークは複素関数 $f(z) = ze^{i\theta}$, $f(z) = \alpha z$ あるいは $f(z) = z + w$ を学習したと考えることができる. ここで, 注意すべきことは, 複素関数 f の定義域は $[-1, 1] \times [-1, 1]$ であるにもかかわらず, ニューラルネットワークが学習したのはその領域のある直線上の複数個の点にすぎないということである. 定義域の一部の点列を学習したニューラルネットワークは, その定義域のすべての点に対して, 学習した複素関数に従った出力を行っている. ニューラルネットワークのこのふるまいは複素解析における一致の定理との関係を持っていることが予想される. 一致の定理は, 実解析には見られない複素解析の特徴的な性質を示したものである. すなわち,

一致の定理

f, g は領域 D 上の正則関数とする. このとき, D 内のある曲線上で $f(z) = g(z)$ ならば, D で恒等的に $f(z) = g(z)$ である.

前述のニューラルネットワークのふるまいはいかえると次のようになる. 学習データは, 複素関数 $f: [-1, 1] \times [-1, 1] \rightarrow [-1, 1] \times [-1, 1]$ の定義域内のある曲線上の点から得られたものとする. ニューラルネットワークは, この与えられた学習データをもとに真の複素関数 f を推定しようとし, その結果, 推定関数 g を求める (ニューラルネットワークは少なくとも学習データについては $f(z) = g(z)$ としようとするはず

である). このとき, ニューラルネットワークは一致の定理を満たすかのごとく定義域 $[-1, 1] \times [-1, 1]$ のすべての点 z に対して真の複素関数 $f(z)$ に近い値を出力する. つまり, ニューラルネットワークが推定した複素関数 g は定義域 $[-1, 1] \times [-1, 1]$ 上で恒等的に $f(z) = g(z)$ となっていることが予想される.

4. おわりに

複素パターンに対する自然なバックプロパゲーション学習アルゴリズムを提案し, そのふるまいを計算機実験を通じて評価した. また, 提案した複素 BP 学習アルゴリズムには実 BP 学習アルゴリズムには見られない図形変換能力が備わっていることを確認した. 複素数への拡張を行うだけでこのような性質が現れることは興味深いことである.

今後の課題としては, 複素 BP 学習アルゴリズムの性質の調査, 一致の定理との関係を含めた理論的考察, 学習アルゴリズムの高速化, 画像処理等への応用, 複素関数の近似能力の解析などが考えられる.

謝辞 本研究の機会を与えてくださった元情報アーキテクチャ部長棟上博士 (現情報処理振興事業協会理事), 討論に参加いただいた研究室諸氏, 有益なコメントをくださった審査委員の方々に感謝いたします.

参考文献

- 1) 甘利俊一: 学習識別の理論, 電子通信学会誌, Vol. 50, No. 7, pp. 1272-1279 (1967).
- 2) Rumelhart, D. E. et al.: *Parallel Distributed Processing*, Vol. 1, MIT Press (1986).
- 3) 新田 徹, 古谷立美: 複素バックプロパゲーション学習とその図形変換能力, 信学技報, NC 90-60, pp. 45-52 (1991).
- 4) 新田 徹, 古谷立美: 複素バックプロパゲーション学習, 第 42 回情報処理学会全国大会論文集, 1E-5 (1991).

(平成 3 年 2 月 12 日受付)

(平成 3 年 6 月 13 日採録)

**新田 徹** (正会員)

1960年生。1983年筑波大学第一学群自然科学類卒業。1985年同大学院修士課程修了。同年日本電気(株)入社。1990年同社退社。同年電子技術総合研究所入所。ニューラルネット

ワーク、資源配分理論の研究に従事。

**古谷 立美** (正会員)

昭和22年生。昭和48年成蹊大学大学院修士課程修了。同年電子技術総合研究所入所。現在同所計算機構研究室長。工学博士。IEEE 会員。