

| | |
|----|----|
| 寄贈 | |
| 新田 | 平成 |
| 徹 | 年 |
| 氏 | 月 |
| | 日 |

DB
1090
1994
(HG)

工学博士 学位論文

論文題目：高次元化されたパラメータを持つ
階層型ニューラルネットワークの研究

1995年3月

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
新田 徹

目次

| | | |
|-------|---------------------------------|----|
| 1 | 序論 | 1 |
| 2 | 複素バックプロパゲーション学習アルゴリズム | 5 |
| 2.1 | はじめに | 5 |
| 2.2 | 複素学習識別モデル | 5 |
| 2.3 | 学習収束定理 | 6 |
| 2.4 | 複素BPネットワーク | 8 |
| 2.5 | 複素BP学習アルゴリズムの導出 | 10 |
| 2.6 | 関連研究 | 11 |
| 2.7 | むすび | 11 |
| 3 | 複素バックプロパゲーション・ネットワークにおける決定表面の構造 | 13 |
| 3.1 | はじめに | 13 |
| 3.2 | 実ニューロンの重みパラメータ | 13 |
| 3.3 | 複素ニューロンの重みパラメータ | 14 |
| 3.4 | 複素BPネットワークにおける決定表面の直交性 | 16 |
| 3.4.1 | 単一複素ニューロンの場合 | 16 |
| 3.4.2 | 3層ネットワークの場合 | 18 |
| 3.5 | シミュレーション | 20 |
| 3.6 | むすび | 21 |
| 4 | 複素バックプロパゲーション学習アルゴリズムの学習特性 | 25 |
| 4.1 | はじめに | 25 |
| 4.2 | 学習パラメータの修正量の構造 | 25 |
| 4.3 | 学習の高速性の要因 | 29 |
| 4.4 | シミュレーション | 31 |
| 4.4.1 | 例題1 | 32 |
| 4.4.2 | 例題2 | 34 |
| 4.4.3 | 考察 | 38 |
| 4.5 | むすび | 38 |

| | | |
|-------|---|----|
| 5 | 複素バックプロパゲーション学習アルゴリズムの2次元アフィン変換学習能力 | 39 |
| 5.1 | はじめに | 39 |
| 5.2 | 実験の諸条件 | 39 |
| 5.3 | 単純な変換 | 40 |
| 5.3.1 | 回転 | 40 |
| 5.3.2 | 相似変換 | 40 |
| 5.3.3 | 平行移動 | 41 |
| 5.3.4 | 誤差の評価 | 44 |
| 5.3.5 | 学習パラメータの値 | 47 |
| 5.4 | やや複雑な変換 | 48 |
| 5.4.1 | 回転 | 48 |
| 5.4.2 | 相似変換 | 50 |
| 5.4.3 | 平行移動 | 51 |
| 5.5 | 考察 | 52 |
| 5.6 | コンピュータビジョンへの応用 | 53 |
| 5.7 | むすび | 55 |
| 6 | 複素バックプロパゲーション学習アルゴリズムの2次元アフィン変換学習能力の数学的解析 | 57 |
| 6.1 | はじめに | 57 |
| 6.2 | 複素BPネットワークモデル | 57 |
| 6.3 | 1つの変換率の学習能力の解析 | 58 |
| 6.3.1 | 回転の学習 | 58 |
| 6.3.2 | 相似変換の学習 | 63 |
| 6.3.3 | 平行移動の学習 | 65 |
| 6.4 | 2つの変換率の学習能力の解析 | 67 |
| 6.5 | 数値例 | 76 |
| 6.6 | 考察 | 79 |
| 6.7 | むすび | 80 |
| 7 | 3次元ベクトル・バックプロパゲーション学習アルゴリズム | 81 |
| 7.1 | はじめに | 81 |
| 7.2 | 3次元ベクトルBP | 81 |
| 7.2.1 | 3次元ベクトル・ニューロン | 81 |
| 7.2.2 | 3次元ベクトル・ニューラルネットワーク | 83 |
| 7.2.3 | 学習アルゴリズム | 84 |
| 7.3 | 3次元アフィン変換学習能力 | 85 |
| 7.3.1 | 回転 | 86 |
| 7.3.2 | 相似変換 | 89 |
| 7.3.3 | 平行移動 | 90 |

| | | |
|-------|--------------------------------------|-----|
| 7.3.4 | 非線形変換 | 92 |
| 7.4 | コンピュータビジョンへの応用 | 93 |
| 7.5 | むすび | 95 |
| 8 | 外積に基づいた3次元化パラメータ・バックプロパゲーション学習アルゴリズム | 97 |
| 8.1 | はじめに | 97 |
| 8.2 | 外積BP | 97 |
| 8.2.1 | 外積ニューロン | 97 |
| 8.2.2 | 外積ニューラルネットワーク | 98 |
| 8.2.3 | 学習アルゴリズム | 99 |
| 8.3 | 学習特性 | 99 |
| 8.4 | むすび | 101 |
| 9 | 結論 | 103 |
| | 謝辞 | 106 |
| | 参考文献 | 107 |

第 1 章

序論

ニューラルネットワークは生物の神経細胞 (ニューロン) の動作を情報処理の観点からモデル化したものであり、人間に近い高度で柔軟な情報処理能力の実現を目指したものである。現在までに学習能力、汎化能力、連想能力などを有することが確認されており、様々な分野に応用されている [5]。

ニューラルネットワークモデルには、大きく分けて、信号の流れが一方方向である階層型と、信号がネットワークの中を何度も流れることができる相互結合型が提案されている。学習方式としては、学習を行なう際に望ましい出力パターン (教師パターンと呼ばれる) を提示する教師付き学習方式と、教師パターンを提示することなく学習を進める教師無し学習方式の 2 種類の学習方式がある。

階層型ニューラルネットワークにおける教師付きの学習アルゴリズムとしては、1986 年に提案されたバックプロパゲーション学習アルゴリズム (以下、実 B P と呼ぶ) がパーセプトロンの限界を打ち破るものとして注目され [53]、近年、様々な分野に盛んに応用されている。

実 B P の応用分野としては、たとえば、パターン認識、人工知能、信号処理、画像処理、音声認識等であるが、分野によっては、フーリエ変換などにより、その処理過程で複素数が現れることがある。あるいは 3 次元情報を陽に扱うことがある。また、人間の脳においては、異なったパルスパターンが存在し、しかもパルス間の距離は異なっているという。このことは位相と絶対値で表現される複素数をニューラルネットワークに導入することの有益性を示唆している。

本論文は階層型ニューラルネットワークの重みや閾値といったパラメータの高次元化 (複素数化 (2 次元化) あるいは 3 次元化) の試みについて論じたものである。高次元化されたパラメータを持つ階層型ニューラルネットワークを高次元化パラメータ・階層型ニューラルネットワークと呼ぶ。この試みの主な目的は、高次元情報 (例えば、複素数、3 次元情報など) を自然に表現することができ、素直に処理することのできる階層型ニューラルネットワークを提供し、その特性を明らかにすることである。

階層型ニューラルネットワークのパラメータの複素数への拡張に関する研究には本研究に先んじて次の 2 つがある。ひとつは Widrow らによるもの (1975 年)[60]、もうひとつは Kim らによるものである (1990 年)[13]。本研究ではこれらとは異なった方法で階層型ニューラルネットワークのパラメータの複素数化を行ない、その特性を調べる。本研究で提案す

る複素数化された階層型ニューラルネットワークは、Widrowら、Kimらの階層型ニューラルネットワークが持っていた収束性に関する問題点を解消した形のものになっている。また、木ノ内らは、本研究の肩の上に乗る、自己フィードバックを持ったタイプの階層型ニューラルネットワークのパラメータを複素数化し、時系列パターンの学習実験を行ない、学習特性の向上について報告している(1994年)[14]。相互結合型ニューラルネットワークのパラメータの複素数化に関しては、記憶容量などに関する議論を行なった根本ら [20](1991年)、広瀬 [9](1992年)による研究がある。

ロボットの動作計画(ロボットを目標配置に至らしめる問題)においては、ロボットの配置(姿勢)の表現に4元数[8]が有効に利用されている[7, 10, 11, 15]。よく知られているオイラー角を使ってロボットの姿勢を表現する方法の場合、姿勢を表すパラメータが一意に決まらない、特異姿勢が存在するという問題がある。また、姿勢の表現が代数的形式でないことから、代数的形式が要請されるいくつかの動作計画法が利用できないという問題もある。一方、4元数を使う方法では、オイラー角の表現が持っていたような特異姿勢は存在しない。4元数にはパラメータが4つあるが、ロボットの姿勢を表す際に用いられているのは、単位4元数であり、大きさが1という制約式があるため、一つは冗長なパラメータで、実質的には3つである。4元数による姿勢の表現を使った動作計画法は、非常に広範囲の動作計画問題を解くことができ、しかも解が必ず得られるという保証(完全性)がある。しかしながら、莫大な計算量を必要とするという問題がある。一方、ニューラルネットワークは、学習後の処理に要する時間は非常に短いうえ、汎化能力も備えている。これらの点で、完全性は失うが、計算量を減少させ、実用的なアルゴリズムを得るという目的で、本論文で提案する3次元化されたパラメータを持つニューラルネットワークを用いることも考えられる。

本論文の構成は次のようになっている。

まず第2章では、Widrowら[60]、Kimら[13]とは異なった方法で、従来の階層型ニューラルネットワークの結合の重みと各ニューロンが持つ閾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(複素BPと呼ぶ)を導出する。学習収束性については、実BPの理論的な基礎となっている甘利による学習識別理論[4]を複素数に拡張することによって保証される。

第3章では、ネットワークアーキテクチャの観点から、複素BPの基本特性を解析的、実験的に調べ、従来のBPとの差異を明確にする。主要な結果は次のとおりである。(1) 複素BPが適用される階層型ニューラルネットワーク(複素BPネットワーク)における重み係数は、従来のBPネットワーク(実BPネットワーク)におけるそれとは異なり、2次元アフィン変換に関係した制約を持っており、学習は基本的にその制約を保ちつつ行われる。(2) 複素ニューロン(複素BPネットワークの構成要素であるニューロン)の実部の決定表面と虚部の決定表面とは互いに直交しており、決定領域を均等に4つに分割する。3層の複素BPネットワークにおける決定表面は基本的にこの構造を内包しており、中間ニューロンへの総入力十分大きいときに直交する。この意味で、一般には複素BPは複素パターンに対する自然な学習アルゴリズムであると考えられる。これらの特性は意図されたものではなく、複素数への拡張の結果として自ずと現れてきたものである。

第4章では、複素BPの学習特性について述べる。複素BPは次の学習特性を持つ。(1) 複素BPにおける誤差逆伝播は2次元アフィン変換に基づいた構造を有している。(2) 複素BPは、ネットワークを流れる複素信号を1つの処理単位として学習を進めるものである。(3) 複素BPでは、学習則における実部要因と虚部要因の補完構造により、学習停滞状態の発生が押えられる。その結果、通常のBPに比べて、複素パターンに対する学習が数倍速くなっている。しかも、そのときに必要となる学習パラメータ数は約半分で済む。複素BPの基本特性に関しては、重みパラメータの2次元アフィン変換構造、決定表面の直交性が第3章で述べられるが、これらは主としてネットワークアーキテクチャからのアプローチにより得られた結果であるのに対し、本章の結果は学習アルゴリズムの観点から得られるものである。

第5章では、複素BPには、実BPには見られない2次元アフィン変換学習能力(図形変換能力)が備わっていることを確認し、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。また、コンピュータビジョンへの応用についても触れる。

第6章では、複素BPの著しい性質である2次元アフィン変換学習能力の数学的解析を行ない、その定性的な性質を明らかにする。主要な結果は次のとおりである。(1) 1種類の変換率(たとえば、1種類の回転角度)を学習した複素BPネットワークのふるまいを明らかにする。変換率に関する汎化能力の(距離に関する)誤差は、未学習パターンと学習パターンの偏角の差に関する正弦関数によって表現される。(2) 2種類の変換率を学習した複素BPネットワークのふるまいを回転角度を例にとって解析し、次の結果を得た。(a) 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の(距離に関する)誤差の評価式を与えた。(b) 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の『(角度に関する)誤差が無視できるほど小さな値であるための十分条件』を求めた。最後に、計算機実験を通じて、これらの数値例を示し、解析結果の妥当性を確認する。

第7章では、ニューラルネットワークのパラメータの3次元化を試みる。前章までと同様に、階層型ニューラルネットワークモデルおよびそれに付随する最も一般的な学習アルゴリズムとして知られているバックプロパゲーション学習アルゴリズムを拡張の対象とする。本章で提案するアルゴリズムは、第2章で提案した複素BPを群論に基づいて3次元パラメータ化したものであり、重みパラメータが3次直交行列、閾値、入力信号および出力信号がすべて3次元実ベクトルである階層型ニューラルネットワークに適用されるものである。さらに、提案するアルゴリズムには、従来のニューラルネットワークには見られない3次元アフィン変換学習能力が備わっていることを確認する。この能力は複素BPが備えている2次元アフィン変換学習能力(第5,6章)に対応するものである。最後に、コンピュータビジョンへの応用について述べる。

第8章では、第7章とは異なった方法でバックプロパゲーション学習アルゴリズムを3次元パラメータ化する。本章で提案するアルゴリズムは、重み、閾値、入出力信号がすべて3次元実ベクトルであり、外積演算が行なわれる階層型ニューラルネットワークに適用される。

このアルゴリズムは3次元ベクトルを自然な方法で学習するのに使うことができる。有効性を確かめるために、3次元ベクトルから成るパターンを使って計算機実験を行なう。

最後に第9章では、結論として本研究で得られた結果を要約し、また将来に対する課題について述べる。

第 2 章

複素バックプロパゲーション学習アルゴリズム

2.1 はじめに

本章では、従来のバックプロパゲーション学習アルゴリズム (実BP)[53] の 複素数版 である複素バックプロパゲーション学習アルゴリズム (複素BP) を提案する [21, 23, 26, 34, 36, 38]。複素BPはニューロン間の結合の重みおよび各ニューロンが持つ閾値がすべて複素数である階層型ニューラルネットワークに適用されるものであり、複素パターンに対する自然な学習を可能にする。収束性に問題のある Widrow ら [60], Kim ら [13] の定式化と異なる点は、各ニューロンの出力関数にある。

本章では、まず、実BPの理論的な基礎となっている学習識別理論 [4] を複素数に拡張することにより複素学習識別理論を用意し、複素BPの理論的基礎とする。複素BPはこの複素学習識別理論から導出され、その学習の収束性についても、これにより保証される。

2.2 複素学習識別モデル

本節では、実BPの理論的基礎となっている学習識別理論 [4] におけるモデルに複素数要素を加味したモデル (複素学習識別モデル) 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 $\mathbf{x} \in C^n$, 情報源 2 から複素パターン $\mathbf{y} \in C^m$ が未知なる同時確率 $P(\mathbf{x}, \mathbf{y})$ で発生するものとする ($\mathbf{x} \in C^n, \mathbf{y} \in C^m$ であることに注意。 C は複素数の全体をあらわす)。ここで、 $\{(\mathbf{x}, \mathbf{y})\}$ は階層型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 \mathbf{x} が与えられたときに、情報源 2 からの複素パターン \mathbf{y} を推定できるようになることである。推定を与える関数を $z(\mathbf{w}, \mathbf{x}) : R^p \times C^n \rightarrow C^m$ とする (R は実数の全体をあらわす)。 $\mathbf{w} \in R^p$ は階層型ニューラルネットワークにおける結合の重みと各ニューロンの閾値のそれぞれの実部と虚部をすべてまとめたものに対応し、推定関数 z の出力値は階層型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、 $r(\mathbf{y}', \mathbf{y}) : C^m \times C^m \rightarrow R^+$ (R^+ は 0 以上の実数の全体をあらわす) を、複素パターン \mathbf{y} を複素パターン \mathbf{y}' で推定したときの損失関

数とし、 $R(\mathbf{w}) : \mathbf{R}^p \rightarrow \mathbf{R}^1$ をパラメータ \mathbf{w} を用いたときの平均損失とする。すなわち、

$$R(\mathbf{w}) \stackrel{\text{def}}{=} \sum_{\mathbf{x}} \sum_{\mathbf{y}} r(\mathbf{z}(\mathbf{w}, \mathbf{x}), \mathbf{y}) P(\mathbf{x}, \mathbf{y}) \quad (2.1)$$

と定義する。 $R(\mathbf{w})$ は階層型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。

2.3 学習収束定理

本節では、2.2 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 [4] の複素数版である。

離散時間パラメータ n を導入し、 $(\mathbf{x}_n, \mathbf{y}_n)$ 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 \mathbf{w} を

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta \mathbf{w}_n \quad (2.2)$$

に従って修正するものとする。 $R_n(\mathbf{w}_n)$ を時刻 n における平均損失とする。また、 $R(\mathbf{w})$ を最小または極小ならしめるパラメータ \mathbf{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 1 十分小さな正数 ε と正定値行列 \mathbf{A} に対して、パラメータ \mathbf{w} を修正量

$$\Delta \mathbf{w}_n = -\varepsilon \mathbf{A} \nabla r(\mathbf{z}(\mathbf{w}_n, \mathbf{x}_n), \mathbf{y}_n) \quad n = 0, 1, \dots \quad (2.3)$$

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

(証明)

[4] の論理がそのまま適用できる。以下、[4] の論理に従って証明する。

n 回の学習後のパラメータ \mathbf{w}_n は、それまでの入出力パターンの系列 $(\mathbf{x}_0, \mathbf{y}_0), \dots, (\mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ に依存する。これらは確率変数であるので、 \mathbf{w}_n は確率変数である。その確率密度関数を $q_n(\mathbf{w}_n)$ とすると、平均損失 $R_n(\mathbf{w}_n)$ の \mathbf{w}_n に関する期待値 $\hat{R}_n = E_{\mathbf{w}_n}[R_n(\mathbf{w}_n)]$ は

$$\hat{R}_n = \int q_n(\mathbf{w}_n) R_n(\mathbf{w}_n) d\mathbf{w}_n, \quad \text{ただし、} d\mathbf{w}_n = d\mathbf{w}_n^{(1)} \dots d\mathbf{w}_n^{(p)} \quad (2.4)$$

と表される。また、 $\Delta \hat{R}_n$ を n 回学習を行なった後、1 回の学習により、パラメータ \mathbf{w}_n を $\Delta \mathbf{w}_n$ だけ修正したときの \hat{R}_n の変化量とする。つまり、

$$\Delta \hat{R}_n \stackrel{\text{def}}{=} \hat{R}_{n+1} - \hat{R}_n \quad (2.5)$$

とする。また、 $\Delta R_n(\mathbf{w}_n)$ を n 回学習を行なった後、1 回の学習により、パラメータ \mathbf{w}_n を $\Delta \mathbf{w}_n$ だけ修正したときの $R_n(\mathbf{w}_n)$ の変化量とする。すなわち、

$$\Delta R_n(\mathbf{w}_n) \stackrel{\text{def}}{=} R_{n+1}(\mathbf{w}_{n+1}) - R_n(\mathbf{w}_n) \quad (2.6)$$

とする。ここで、 $w_{n+1} = w_n + \Delta w_n$ であり、 Δw_n は式 (2.3) からわかるように、 x_n, y_n, w_n に依存して決まる。よって、 $\Delta R_n(w_n)$ は x_n, y_n, w_n の関数である。さらに、 $\overline{\Delta R_n(w_n)}$ を $\Delta R_n(w_n)$ の (x_n, y_n) に関する平均とする。つまり、

$$\overline{\Delta R_n(w_n)} \stackrel{\text{def}}{=} E_{x_n, y_n} [\Delta R_n(w_n)] \quad (2.7)$$

とする。このとき、 $\Delta \hat{R}_n$ は ΔR_n を x_n, y_n, w_n に関して平均したものであるから、

$$\begin{aligned} \Delta \hat{R}_n &= E_{x_n, y_n, w_n} [\Delta R_n] \\ &= E_{w_n} [\overline{\Delta R_n(w_n)}] \\ &= \int q_n(w_n) \overline{\Delta R_n(w_n)} dw_n \end{aligned} \quad (2.8)$$

と表せる。ここで、式 (2.6) で定義された $\Delta R_n(w_n)$ は、テイラー展開により、

$$\Delta R_n(w_n) = {}^t \Delta w_n \cdot \nabla R_n + \frac{1}{2} {}^t \Delta w_n \cdot \nabla \nabla R_n \cdot \Delta w_n + \dots, \quad (2.9)$$

ただし、 ${}^t \mathbf{x}$ はベクトル \mathbf{x} の転置をあらわす。そこで、式 (2.9) の (x_n, y_n) に関する期待値をとると、

$$\begin{aligned} \overline{\Delta R_n(w_n)} &= E_{x_n, y_n} [\Delta R_n(w_n)] \\ &= E_{x_n, y_n} [{}^t \Delta w_n \cdot \nabla R_n] + \frac{1}{2} E_{x_n, y_n} [{}^t \Delta w_n \cdot \nabla \nabla R_n \cdot \Delta w_n] + \dots \\ &= E_{x_n, y_n} [{}^t \Delta w_n] \cdot \nabla R_n + \frac{1}{2} \varepsilon^2 \cdot \text{tr}[\mathbf{A} \mathbf{Q} {}^t \mathbf{A} \cdot \nabla \nabla R_n] + O(\varepsilon^3), \end{aligned} \quad (2.10)$$

ただし、 tr は行列のトレースを意味する、また、

$$\mathbf{Q} \stackrel{\text{def}}{=} \sum_{x_n} \sum_{y_n} \nabla r(z(w_n, x_n), y_n) \cdot {}^t \nabla r(z(w_n, x_n), y_n) P(x_n, y_n) \quad (2.11)$$

とおいた。よって、式 (2.8) および式 (2.10) により、

$$\Delta \hat{R}_n = \int q_n(w_n) \left[E_{x_n, y_n} [{}^t \Delta w_n] \cdot \nabla R_n \right] dw_n + O(\varepsilon^2). \quad (2.12)$$

ここで、

$$\begin{aligned} E_{x_n, y_n} [\Delta w_n] &= E_{x_n, y_n} [-\varepsilon \mathbf{A} \nabla r(z(w_n, x_n), y_n)] && \text{(式 (2.3) による)} \\ &= -\varepsilon \mathbf{A} \sum_{x_n} \sum_{y_n} \nabla r(z(w_n, x_n), y_n) P(x_n, y_n) \\ &= -\varepsilon \mathbf{A} \nabla R_n. \end{aligned} \quad (2.13)$$

よって、式 (2.12) および式 (2.13) により、 A は正定値行列であることに注意すると、

$$\begin{aligned}\Delta \hat{R}_n &= -\varepsilon \int q_n(\mathbf{w}_n) ({}^t \nabla R_n A \nabla R_n) d\mathbf{w}_n + O(\varepsilon^2) \\ &\leq 0 + O(\varepsilon^2).\end{aligned}\tag{2.14}$$

ε^2 の項を無視することにより、 $\Delta \hat{R}_n \leq 0$ であるから、 \hat{R}_n は単調減少列である。また、 $r(\mathbf{z}(\mathbf{w}_n, \mathbf{x}_n), \mathbf{y}_n) \geq 0$ であるから、 $\hat{R}_n \geq 0$ である。つまり、 $\{\hat{R}_n; n = 0, 1, \dots\}$ は下に有界な単調減少列である。故に、 $\{\hat{R}_n\}$ は収束する。よって、 $\lim_{n \rightarrow \infty} \Delta \hat{R}_n = 0$ 。故に、

$$\begin{aligned}\lim_{n \rightarrow \infty} \Delta \hat{R}_n &= -\varepsilon \lim_{n \rightarrow \infty} \int q_n(\mathbf{w}_n) ({}^t \nabla R_n A \nabla R_n) d\mathbf{w}_n && \text{(式 (2.14) による)} \\ &= 0.\end{aligned}\tag{2.15}$$

つまり、

$$\lim_{n \rightarrow \infty} \int q_n(\mathbf{w}_n) ({}^t \nabla R_n A \nabla R_n) d\mathbf{w}_n = 0.\tag{2.16}$$

従って、 $\nabla R_n \neq 0$ なるすべての \mathbf{w}_n (つまり、最適でないパラメータ) に対して、 $\lim_{n \rightarrow \infty} q_n(\mathbf{w}_n) = 0$ である。すなわち、学習によって最適でないパラメータの生起する確率は 0 に収束する。つまり、 ε^2 の項を無視すれば、 \mathbf{w}_n は最適なパラメータに収束する。

(証明終)

2.4 複素BPネットワーク

本節では、複素BPネットワークを定式化する。

まず、複素ニューロンを定式化する (図 2.1)。入力信号、重み、閾値および出力信号はすべて複素数とする。複素ニューロン j の内部ポテンシャル z_j は、実BPにおいて用いられるニューロン (実ニューロン と呼ぶ) の内部ポテンシャルの類推で、

$$z_j = \sum_k w_{jk} s_k + t_j\tag{2.17}$$

と定義する。ここで、 w_{jk} は複素ニューロン j と複素ニューロン k との間の重みを表す複素数、 s_k は複素ニューロン k から出力された複素ニューロン j への入力信号を表す複素数、 t_j は複素ニューロン j の閾値を表す複素数である。出力信号 $f_C(z_j)$ は次のように定義される：

$$f_C(z_j) = f_R(x_j) + i f_R(y_j), \quad z_j = x_j + i y_j,\tag{2.18}$$

ただし、 $f_R(u) = 1/(1 + \exp(-u))$ 、 $i = \sqrt{-1}$ である。すなわち、複素ニューロンの出力値の実部および虚部は、それぞれ、複素ニューロンへの入力の総和 z_j の実部、虚部に関するシグモイド関数であり、明らかに、 $0 \leq \text{Re}[f_C], \text{Im}[f_C] \leq 1$ 、 $|f_C(z_j)| \leq \sqrt{2}$ を満たす ($\text{Re}[z]$ 、 $\text{Im}[z]$ はそれぞれ複素数 z の実部、虚部)。上記の定式化において、出力関数 f_C に

はいろいろなものを考えることができ、それらに応じて複素ニューロンにはそれら固有の性質が備わることになる。

次に、上で定式化した複素ニューロンを使って、複素BPネットワークを定式化する。簡単のために、3層のネットワークを例にとる(図2.2)。 w_{ji} を入力複素ニューロン i と中間複素ニューロン j との間の結合の重み、 v_{kj} を中間複素ニューロン j と出力複素ニューロン k との間の結合の重み、 θ_j を中間複素ニューロン j が持つ閾値、 γ_k を出力複素ニューロン k が持つ閾値とする。また、 I_i 、 H_j 、 O_k を、それぞれ、入力複素ニューロン i 、中間複素ニューロン j 、出力複素ニューロン k の出力値とし、 U_j 、 S_k をそれぞれ中間複素ニューロン j 、出力複素ニューロン k の内部ポテンシャルとする。すなわち、 $U_j = \sum_i w_{ji} I_i + \theta_j$ 、 $S_k = \sum_j v_{kj} H_j + \gamma_k$ 、 $H_j = f_C(U_j)$ 、 $O_k = f_C(S_k)$ と定義する。また、出力複素ニューロン k に対する教師パターン(出力学習パターン) T_k と出力複素ニューロン 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、複素パターン 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。

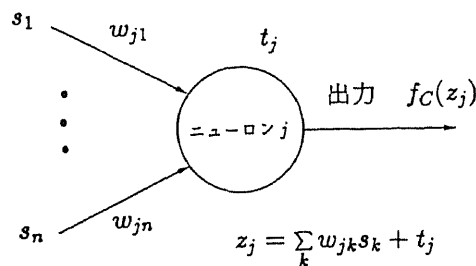


図2.1 複素ニューロン. $s_1, \dots, s_n, w_{j1}, \dots, w_{jn}, t_j, f_C(z_j)$ はすべて複素数.

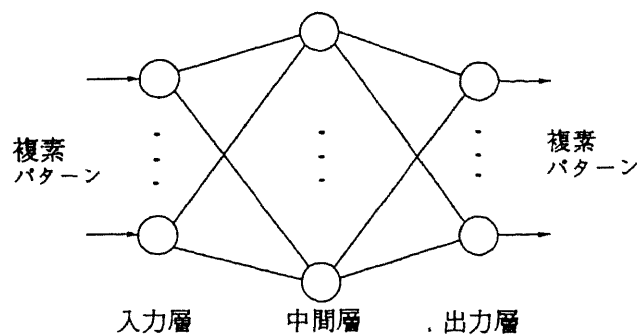


図2.2 3層複素BPネットワーク

2.5 複素BP学習アルゴリズムの導出

本節では、2.2 節および2.3 節で述べた複素学習識別理論を2.4 節で定義した複素BPネットワークに適用することにより、複素BP学習アルゴリズムを導出する。

学習率 $\varepsilon > 0$ を十分小さくとり、正定値行列 \mathbf{A} を単位行列として、2.3 節で示した定理1を適用すると、結合の重みおよび閾値の修正量は、

$$\Delta v_{kj} = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[v_{kj}]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[v_{kj}]}, \quad (2.19)$$

$$\Delta \gamma_k = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[\gamma_k]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[\gamma_k]}, \quad (2.20)$$

$$\Delta w_{ji} = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[w_{ji}]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[w_{ji}]}, \quad (2.21)$$

$$\Delta \theta_j = -\varepsilon \frac{\partial E_p}{\partial \text{Re}[\theta_j]} - i\varepsilon \frac{\partial E_p}{\partial \text{Im}[\theta_j]} \quad (2.22)$$

とすれば良いことになり、結局、

$$\Delta v_{kj} = \overline{H_j} \Delta \gamma_k, \quad (2.23)$$

$$\Delta \gamma_k = \varepsilon \left\{ \text{Re}[\delta^k](1 - \text{Re}[O_k])\text{Re}[O_k] + i\text{Im}[\delta^k](1 - \text{Im}[O_k])\text{Im}[O_k] \right\}, \quad (2.24)$$

$$\Delta w_{ji} = \overline{I_i} \Delta \theta_j, \quad (2.25)$$

$$\begin{aligned} \Delta \theta_j = \varepsilon & \left[(1 - \text{Re}[H_j])\text{Re}[H_j] \right. \\ & \sum_k \left\{ \text{Re}[\delta^k](1 - \text{Re}[O_k])\text{Re}[O_k]\text{Re}[v_{kj}] + \text{Im}[\delta^k](1 - \text{Im}[O_k])\text{Im}[O_k]\text{Im}[v_{kj}] \right\} \\ & \left. - i(1 - \text{Im}[H_j])\text{Im}[H_j] \right. \\ & \left. \sum_k \left\{ \text{Re}[\delta^k](1 - \text{Re}[O_k])\text{Re}[O_k]\text{Im}[v_{kj}] - \text{Im}[\delta^k](1 - \text{Im}[O_k])\text{Im}[O_k]\text{Re}[v_{kj}] \right\} \right] \end{aligned} \quad (2.26)$$

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におけるパラメータの修正量を次に記しておく。式(2.23) - (2.26)と式(2.27) - (2.30)は互いに似通った形をしていることがわかる。ただし、式(2.27) - (2.30)においては、 δ^k , I_i , H_j , O_k , v_{kj} , γ_k , w_{ji} , θ_j はすべて実数である。

$$\Delta v_{kj} = H_j \Delta \gamma_k, \quad (2.27)$$

$$\Delta \gamma_k = \varepsilon \delta^k (1 - O_k) O_k, \quad (2.28)$$

$$\Delta w_{ji} = I_i \Delta \theta_j, \quad (2.29)$$

$$\Delta \theta_j = \varepsilon (1 - H_j) H_j \sum_k \{ \delta^k (1 - O_k) O_k v_{kj} \}. \quad (2.30)$$

2.6 関連研究

第1章でも述べたように、パラメータを複素数化した階層型ニューラルネットワークの他の定式化として次の2つがある。(1) ひとつは、Widrowらによるニューロンの出力関数が正則な複素関数 $f_C(z) = z$, ただし, $z = x + iy$ であるもの [60]。この場合, 非線形な要因が含まれていないことになる。また, この定式化では, 2層のネットワークに限っていた。

(2) もうひとつは, Kimらによる出力関数が正則な複素関数 $f_C(z) = 1/(1 + \exp(-z))$, ただし, $z = x + iy$ であるものである [13]。この定式化では閾値は常数 $1+i$ を使っている。これら2つの定式化は, 2.2 節および 2.3 節で扱った複素学習識別理論の範疇で扱うことができるものである。

我々は, 当初, Kimら [13] とは独立に, 出力関数に彼らと同じ $f_C(z) = 1/(1 + \exp(-z))$, ただし, $z = x + iy$ を使った定式化を行なった。この定式化が最も自然であると思われたし, この関数は正則関数であることから, 興味深い結果が出てくるであろうと予想したからである。しかし, 実験の結果, 我々の行なった実験の範囲内では収束しなかった (発散した)。そこで, 出力関数が有界でなかったことに着目し, 正則ではないが, 有界な関数 (式 (2.18)) を用いたという経緯があった。正則かつ有界な出力関数を用いることを考えると, リュービルの定理 (たとえば, [12]) により, 全複素平面 C で正則かつ有界な関数は定数だけとなる。よって, 出力関数を有界にしようとする, どうしても正則性を破らざるを得ない。そのひとつの結果が, 式 (2.18) で与えた正則ではないが有界な関数である。ちなみに, 有界でない出力関数を持つ Widrowら [60] による定式化に従って実験を行なってみたが, 我々の行なった範囲では学習はすべて発散した。

以上のように, 本研究で提案した複素BPは, 既存の定式化が持っていた収束性に関する問題点を解消した形のものとなっている。ちなみに, Widrowら, Kimらにより提案された複素化パラメータを持つ階層型ニューラルネットワークのその後の研究の進展はみられない。

2.7 むすび

本章では, 複素パターンを自然に学習することのできる複素バックプロパゲーション学習アルゴリズム (複素BP) を提案した。従来のバックプロパゲーション学習アルゴリズム (実BP) は学習識別理論がその理論的基礎となっていた。そこで, 本章では学習識別理論を複素数に拡張した複素学習識別理論を準備し, それをもとに複素BPを導いた。得られた複素BPの学習アルゴリズムは, 実BPにおける学習アルゴリズムと形において似通ったものとなっている。実BPは実数 (1次元) を処理の対象としていたが, 複素BPは複素数 (2次元) を扱うという意味で実BPの2次元版であると言える。本章で提案した複素BPは, Widrowら [60], Kimら [13] による定式化が持っていた収束性に関する問題点を解決している。

第 3 章

複素バックプロパゲーション・ネットワークにおける決定表面の構造

3.1 はじめに

第 2 章において提案した複素 BP には、実 BP には見られない様々な性質が備わっていることが予想される。

本章では、ネットワークアーキテクチャの観点から、複素 BP の基本特性を解析的、実験的に調べ、実 BP との差異を明確にする [22, 24, 30, 45, 48]。主要な結果は次のとおりである。(1) 複素 BP ネットワークにおける重み係数は、実 BP ネットワークにおけるそれとは異なり、2次元アフィン変換 に関係した制約を持っており、学習は基本的にその制約を保ちつつ行われる。(2) 複素ニューロンの実部の決定表面と虚部の決定表面とは 互いに直交 しており、決定領域を均等に 4 つに分割する。3 層の複素 BP ネットワークにおける決定表面は基本的にこの構造を内包しており、中間ニューロンへの総入力十分大きいときに直交する。

3.2 節から 3.4 節では、複素 BP ネットワークの解析を行い、決定表面の性質などの基本特性について調べる。それらの計算機実験の結果を 3.5 節で示し、最後に 3.6 節で結論を述べる。

3.2 実ニューロンの重みパラメータ

まず、実ニューロンの入力の結合重みの基本的な構造について確認しておく。重み $w_k \in \mathbf{R}^1$ ($1 \leq k \leq n$) および閾値 $\theta \in \mathbf{R}^1$ を持つ n 入力実ニューロンを考える。出力関数 $f_R : \mathbf{R}^1 \rightarrow \mathbf{R}^1$ は $f_R(u) = 1/(1 + \exp(-u))$ とする。このとき、入力 $x_k \in \mathbf{R}^1$ ($1 \leq k \leq n$) が与えられると、実ニューロンは $f_R(\sum_{k=1}^n w_k x_k + \theta)$ を出力する。つまり、実ニューロンは実数直線 (1次元) 上の n 個の点 x_1, \dots, x_n を原点との距離に関して、それぞれ、 w_1, \dots, w_n 倍し、それらを 1次元実ベクトルとみなして加えた結果得られるベクトルの示す点を更に θ だけ移動させたと解釈することができる (図 3.1)。その後、非線形変換 f_R を施したものを出力値としている。このように、実ニューロンは 基本的に 1次元上の点列の移動を司っており、重みパラメータ w_1, \dots, w_n は互いに独立であって、何らの影響も与え合わない。ここで、実

ニューロンの出力値は非線形変換 f_R が施されることによって得られるため、実ニューロンの出力値に関して上記の幾何学的解釈が 厳密に 成り立つのは非線形変換 f_R が線形変換とみなせるとき、すなわち、実ニューロンへの入力 x_k の総和の絶対値 $|\sum_{k=1}^n w_k x_k + \theta|$ が小さな値であるときに限ることに注意しなければならない。

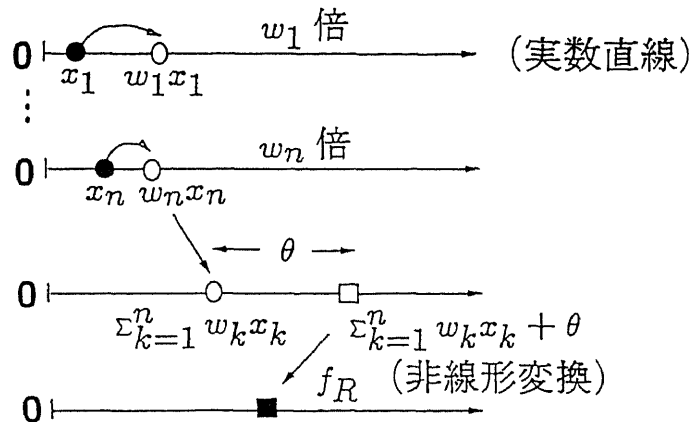


図 3.1 実ニューロンにおける処理のイメージ

3.3 複素ニューロンの重みパラメータ

次に、複素ニューロンの入力の結合重みの基本的な構造について調べる。重み $w_k = w_k^r + iw_k^i \in \mathbb{C}^1$ ($1 \leq k \leq n$) および閾値 $\theta = \theta^r + i\theta^i \in \mathbb{C}^1$ を持つ n 入力複素ニューロンを考える。入力 $x_k + iy_k \in \mathbb{C}^1$ ($1 \leq k \leq n$) が与えられると、この複素ニューロンの出力 $X + iY$ は、

$$\begin{aligned} X + iY &= f_{\mathbb{C}}\left(\sum_{k=1}^n (w_k^r + iw_k^i)(x_k + iy_k) + (\theta^r + i\theta^i)\right) \\ &= f_R\left(\sum_{k=1}^n (w_k^r x_k - w_k^i y_k) + \theta^r\right) + if_R\left(\sum_{k=1}^n (w_k^i x_k + w_k^r y_k) + \theta^i\right) \end{aligned} \quad (3.1)$$

となる。故に、 n 入力複素ニューロンは図 3.2 に示す 2 つの n 入力実ニューロンと等価である。複素ニューロンの出力の実部 X に相当する実ニューロンを“実部”ニューロン、虚部 Y に相当する実ニューロンを“虚部”ニューロンと呼ぶことにする。ここで、

$$\begin{aligned}
\begin{bmatrix} X \\ Y \end{bmatrix} &= F_C \left(\begin{bmatrix} w_1^r & -w_1^i & \cdots & w_n^r & -w_n^i \\ w_1^i & w_1^r & \cdots & w_n^i & w_n^r \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right) \\
&= F_C \left(|w_1| \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 \\ \sin \alpha_1 & \cos \alpha_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \cdots + |w_n| \begin{bmatrix} \cos \alpha_n & -\sin \alpha_n \\ \sin \alpha_n & \cos \alpha_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right), \tag{3.2}
\end{aligned}$$

ただし、 $F_C \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} f_R(x) \\ f_R(y) \end{bmatrix}$, $\alpha_k = \arctan(w_k^i/w_k^r)$ ($1 \leq k \leq n$) である。

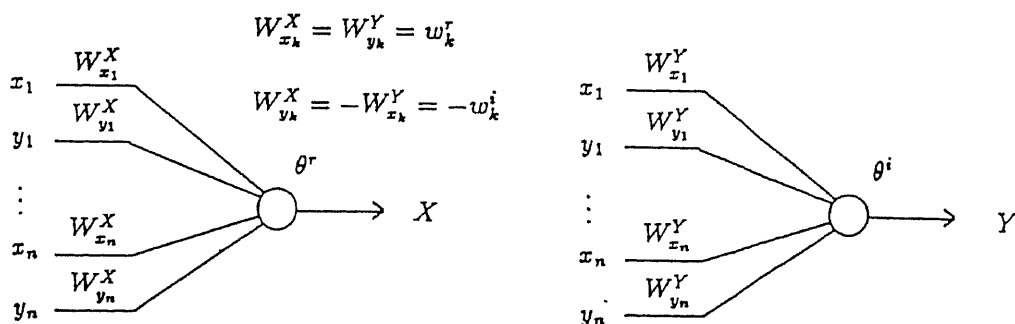


図 3. 2 複素ニューロンと等価な実ニューロン

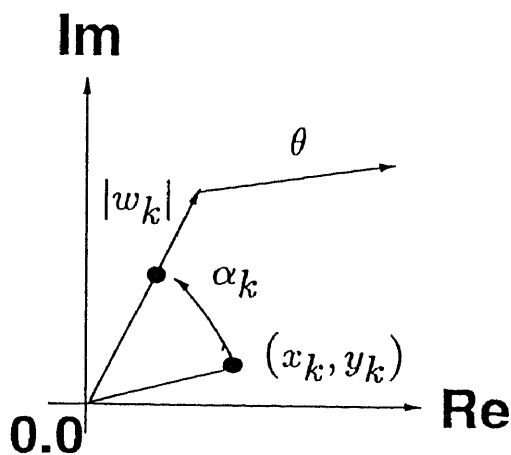


図 3. 3 複素信号に対する 2 次元アフィン変換のイメージ

式(3.2)において、 $|w_k|$ は2次元平面における点 (x_k, y_k) の原点との距離に関する縮小・拡大、 $\begin{bmatrix} \cos \alpha_k & -\sin \alpha_k \\ \sin \alpha_k & \cos \alpha_k \end{bmatrix}$ は原点を中心にした正の方向（反時計回り）への α_k 度の回転、 $[\theta^r \ \theta^i]$ は平行移動を意味する。このように、 n 入力複素ニューロンは、入力された2次元情報（複素数）それぞれに対して“アフィン変換”を施している（図3.3）。すなわち、式(3.2)は2次元平面上の“アフィン変換”を基本的に含んでいると言える。ただし、複素ニューロンの出力値 $[X \ Y]$ は、非線形変換 F_C が施されることによって得られるため、 $[X \ Y]$ に関して上記の幾何学的解釈が厳密に成り立つのは、非線形変換 F_C が線形変換とみなせるとき、すなわち、複素ニューロンへの入力の総和（複素数）の実部および虚部のそれぞれの絶対値が共に小さいときに限ることに注意しなければならない。

前節において見たように、実ニューロンは基本的に実数直線（1次元）上の点列の移動を司っており、実ニューロン間の重みパラメータは互いに独立であって、何らの影響も与え合わない。複素ニューロンは先に見たように基本的に2次元平面上の“アフィン変換”を司っており、学習においてはそのアフィン変換の調整を行うものと解釈することができる。それに伴って、複素ニューロンには、

$$\begin{aligned} & (\text{“実部”ニューロンへの入力値の実部 } x_k \text{ に対する重み}) \\ = & (\text{“虚部”ニューロンへの入力値の虚部 } y_k \text{ に対する重み}), \\ & (\text{“実部”ニューロンへの入力値の虚部 } y_k \text{ に対する重み}) \\ = & - (\text{“虚部”ニューロンへの入力値の実部 } x_k \text{ に対する重み}) \end{aligned}$$

という制約が重みに課されており（図3.2参照）、学習はその制約を保ちつつ行われる。見方を変えると“実部”ニューロンと“虚部”ニューロンはそれらの重みパラメータを通じて互いに影響を与え合っていると言える。このように、基本的に数直線（1次元）上の移動を司っていた実BPは、複素数に拡張されることにより、平面（2次元）上のアフィン変換に関した構造を有するものへと変質していることがわかった。次節で述べるように、重みパラメータのこの構造は決定表面の直交性として現れる。

3.4 複素BPネットワークにおける決定表面の直交性

決定表面（decision boundary）とは、学習識別系（たとえば、実BP）がパターンを切り分ける境界線のことであり、一般に超曲面から成るものである。実ニューロンから構成される階層型ニューラルネットワークの決定表面については、Lippmannにより、中間層の個数が0～2個の3種類に分けて実験的に調べられている[16]。そこで、本節では複素BPネットワークの決定表面について解析的に調べる。

3.4.1 単一複素ニューロンの場合

まず、単一の複素ニューロン（中間層0個）の決定表面について調べる。

重みを $\mathbf{w} = {}^t[w_1 \cdots w_n] = \mathbf{w}^r + i\mathbf{w}^i$, $\mathbf{w}^r = {}^t[w_1^r \cdots w_n^r]$, $\mathbf{w}^i = {}^t[w_1^i \cdots w_n^i]$, 閾値を $\theta = \theta^r + i\theta^i$ とする。出力関数 $f_1: C^1 \rightarrow C^1$ は、簡単のために、

$$f_1(z) = 1(x) + i1(y), \quad z = x + iy, \quad (3.3)$$

ただし, $1(u) = \begin{cases} 1 & (\text{if } u \geq 0) \\ 0 & (\text{if } u < 0) \end{cases}$ とする。このとき, 複素パターン $\mathbf{z} = {}^t[z_1 \cdots z_n] = \mathbf{x} + i\mathbf{y}$, $\mathbf{x} = {}^t[x_1 \cdots x_n]$, $\mathbf{y} = {}^t[y_1 \cdots y_n]$ が入力されると複素ニューロンは、

$$1\left({}^t[\mathbf{w}^r \quad -\mathbf{w}^i] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^r\right) + i1\left({}^t[\mathbf{w}^i \quad \mathbf{w}^r] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^i\right) \quad (3.4)$$

を出力する。故に、複素ニューロンの出力値の 実部 は ${}^t[\mathbf{w}^r \quad -\mathbf{w}^i]$ を法線ベクトルとする超平面 (決定表面) ${}^t[\mathbf{w}^r \quad -\mathbf{w}^i] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^r = 0$ を境にして定まる。つまり、超平面 ≥ 0 ならば 1, そうでなければ 0 となる。同様に、複素ニューロンの出力値の 虚部 は ${}^t[\mathbf{w}^i \quad \mathbf{w}^r]$ を法線ベクトルとする超平面 (決定表面) ${}^t[\mathbf{w}^i \quad \mathbf{w}^r] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^i = 0$ を境にして定まる。ここで、これら 2 つの決定表面の法線ベクトルの内積をとると ${}^t[\mathbf{w}^r \quad -\mathbf{w}^i] \begin{bmatrix} \mathbf{w}^i \\ \mathbf{w}^r \end{bmatrix} = 0$ となり、2 つの決定表面は直交しているということがわかる。

次に、式 (2.18) で示される出力関数を持つ場合について調べる。複素パターン $\mathbf{z} = {}^t[z_1 \cdots z_n] = \mathbf{x} + i\mathbf{y}$, $\mathbf{x} = {}^t[x_1 \cdots x_n]$, $\mathbf{y} = {}^t[y_1 \cdots y_n]$ が入力されると複素ニューロンは、

$$X + iY = f_R\left({}^t[\mathbf{w}^r \quad -\mathbf{w}^i] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^r\right) + if_R\left({}^t[\mathbf{w}^i \quad \mathbf{w}^r] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^i\right) \quad (3.5)$$

を出力する。ここで、常数 $C^R, C^I \in (0, 1)$ を任意に選び、

$$X(\mathbf{x}, \mathbf{y}) = f_R\left({}^t[\mathbf{w}^r \quad -\mathbf{w}^i] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^r\right) = C^R, \quad (3.6)$$

$$Y(\mathbf{x}, \mathbf{y}) = f_R\left({}^t[\mathbf{w}^i \quad \mathbf{w}^r] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^i\right) = C^I \quad (3.7)$$

とおく。式 (3.6) は n 入力複素ニューロンの出力値の 実部 に関する決定表面である、すなわち、入力信号 $(\mathbf{x}, \mathbf{y}) \in \mathbf{R}^{2n}$ は、超曲面 (式 (3.6)) を境界線にして、2 つの決定領域 (decision region) $\{(\mathbf{x}, \mathbf{y}) \in \mathbf{R}^{2n} | X(\mathbf{x}, \mathbf{y}) \geq C^R\}$, $\{(\mathbf{x}, \mathbf{y}) \in \mathbf{R}^{2n} | X(\mathbf{x}, \mathbf{y}) < C^R\}$ に分類される。同様にして、式 (3.7) で示される超曲面は n 入力複素ニューロンの 虚部 に関する決定表面である。そこで、式 (3.6) および式 (3.7) のそれぞれの法線ベクトル $H^R(\mathbf{x}, \mathbf{y}), H^I(\mathbf{x}, \mathbf{y})$ を求めると、

$$H^R(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial X}{\partial x_1} \quad \cdots \quad \frac{\partial X}{\partial x_n} \quad \frac{\partial X}{\partial y_1} \quad \cdots \quad \frac{\partial X}{\partial y_n} \right)$$

$$= f'_R\left([{}^t\mathbf{w}^r \quad -{}^t\mathbf{w}^i] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^r\right) \cdot [{}^t\mathbf{w}^r \quad -{}^t\mathbf{w}^i], \quad (3.8)$$

$$\begin{aligned} H^I(\mathbf{x}, \mathbf{y}) &= \left(\frac{\partial Y}{\partial x_1} \cdots \frac{\partial Y}{\partial x_n} \frac{\partial Y}{\partial y_1} \cdots \frac{\partial Y}{\partial y_n} \right) \\ &= f'_R\left([{}^t\mathbf{w}^i \quad {}^t\mathbf{w}^r] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^i\right) \cdot [{}^t\mathbf{w}^i \quad {}^t\mathbf{w}^r] \end{aligned} \quad (3.9)$$

となり、それらの内積は0となるから、実部に関する決定表面と虚部に関する決定表面は互いに直交していることがわかる。

実ニューロンは入力された実パターンを2つの (0 および 1 という値で代表される) クラスに分類する。一方、複素ニューロンは入力された複素パターンを4つの (0, 1, i , $1+i$ という値で代表される) クラスに分類する。先の議論により、複素ニューロンの決定表面は2つの直交した超曲面 (超平面) から構成されており、領域を4つに均等に分割しようとする。つまり、一般には複素ニューロンは複素パターンに対する自然な決定表面を有していると言える。

3.4.2 3層ネットワークの場合

次に、3層ネットワーク (中間層1個) の決定表面について調べる。入力ニューロン L 個、中間ニューロン M 個、出力ニューロン N 個の3層ネットワークを考える。 $w_{ji} = w_{ji}^r + iw_{ji}^i$ を入力ニューロン i と中間ニューロン j との間の結合の重み、 $v_{kj} = v_{kj}^r + iv_{kj}^i$ を中間ニューロン j と出力ニューロン k との間の結合の重み、 $\theta_j = \theta_j^r + i\theta_j^i$ を中間ニューロン j が持つ閾値、 $\gamma_k = \gamma_k^r + i\gamma_k^i$ を出力ニューロン k が持つ閾値とする。出力関数は、式 (2.18) で示されるものとする。このとき、複素パターン $\mathbf{z} = {}^t[z_1 \cdots z_L] = \mathbf{x} + i\mathbf{y}$, $\mathbf{x} = {}^t[x_1 \cdots x_L]$, $\mathbf{y} = {}^t[y_1 \cdots y_L]$ が入力されると、中間ニューロン j への入力の総和 U_j は、

$$\begin{aligned} U_j &= U_j^r + iU_j^i \\ &= \left[\sum_{i=1}^L (w_{ji}^r x_i - w_{ji}^i y_i) + \theta_j^r \right] + i \left[\sum_{i=1}^L (w_{ji}^i x_i + w_{ji}^r y_i) + \theta_j^i \right] \end{aligned} \quad (3.10)$$

となる。よって、その出力 H_j は、

$$H_j = H_j^r + iH_j^i = f_R(U_j^r) + if_R(U_j^i) \quad (3.11)$$

である。さらに、出力ニューロン k への総入力 S_k は、

$$\begin{aligned} S_k &= S_k^r + iS_k^i \\ &= \left[\sum_{j=1}^M (v_{kj}^r H_j^r - v_{kj}^i H_j^i) + \gamma_k^r \right] + i \left[\sum_{j=1}^M (v_{kj}^i H_j^r + v_{kj}^r H_j^i) + \gamma_k^i \right] \end{aligned} \quad (3.12)$$

である。故に、その出力 O_k は、

$$O_k = O_k^r + iO_k^i = f_R(S_k^r) + if_R(S_k^i) \quad (3.13)$$

となる。ここで、常数 $C^R, C^I \in (0, 1)$ を任意に選び、

$$O_k^r(\mathbf{x}, \mathbf{y}) = C_R, \quad (3.14)$$

$$O_k^i(\mathbf{x}, \mathbf{y}) = C_I \quad (3.15)$$

とおく。式 (3.14) および式 (3.15) は、それぞれ、3層の複素BPネットワークにおける出力ニューロン k の実部および虚部に関する決定表面である。これら2つの超曲面の法線ベクトル $H^R(\mathbf{x}, \mathbf{y}), H^I(\mathbf{x}, \mathbf{y})$ は、それぞれ、

$$H^R(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial O_k^r}{\partial x_1} \dots \frac{\partial O_k^r}{\partial x_L} \frac{\partial O_k^r}{\partial y_1} \dots \frac{\partial O_k^r}{\partial y_L} \right), \quad (3.16)$$

$$H^I(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial O_k^i}{\partial x_1} \dots \frac{\partial O_k^i}{\partial x_L} \frac{\partial O_k^i}{\partial y_1} \dots \frac{\partial O_k^i}{\partial y_L} \right) \quad (3.17)$$

であり、これらの内積は、

$$\begin{aligned} & H^R(\mathbf{x}, \mathbf{y}) \cdot {}^t H^I(\mathbf{x}, \mathbf{y}) \\ &= \frac{\partial O_k^r}{\partial x_1} \cdot \frac{\partial O_k^i}{\partial x_1} + \dots + \frac{\partial O_k^r}{\partial x_L} \cdot \frac{\partial O_k^i}{\partial x_L} + \frac{\partial O_k^r}{\partial y_1} \cdot \frac{\partial O_k^i}{\partial y_1} + \dots + \frac{\partial O_k^r}{\partial y_L} \cdot \frac{\partial O_k^i}{\partial y_L} \end{aligned} \quad (3.18)$$

で与えられる。ここで、任意の $1 \leq i \leq L$ に対して、

$$\begin{aligned} & \frac{\partial O_k^r}{\partial x_i} \cdot \frac{\partial O_k^i}{\partial x_i} + \frac{\partial O_k^r}{\partial y_i} \cdot \frac{\partial O_k^i}{\partial y_i} \\ &= \frac{\partial f_R(S_k^r)}{\partial S_k^r} \cdot \frac{\partial f_R(S_k^i)}{\partial S_k^i} \cdot \left[\sum_{j=1}^M \left(v_{kj}^r w_{ji}^r \cdot \frac{\partial f_R(U_j^r)}{\partial U_j^r} - v_{kj}^i w_{ji}^i \cdot \frac{\partial f_R(U_j^i)}{\partial U_j^i} \right) \right] \\ & \quad \cdot \left[\sum_{j=1}^M \left(v_{kj}^i w_{ji}^r \cdot \frac{\partial f_R(U_j^r)}{\partial U_j^r} + v_{kj}^r w_{ji}^i \cdot \frac{\partial f_R(U_j^i)}{\partial U_j^i} \right) \right] \\ & \quad - \frac{\partial f_R(S_k^r)}{\partial S_k^r} \cdot \frac{\partial f_R(S_k^i)}{\partial S_k^i} \cdot \left[\sum_{j=1}^M \left(v_{kj}^r w_{ji}^r \cdot \frac{\partial f_R(U_j^i)}{\partial U_j^i} - v_{kj}^i w_{ji}^i \cdot \frac{\partial f_R(U_j^r)}{\partial U_j^r} \right) \right] \\ & \quad \cdot \left[\sum_{j=1}^M \left(v_{kj}^i w_{ji}^r \cdot \frac{\partial f_R(U_j^i)}{\partial U_j^i} + v_{kj}^r w_{ji}^i \cdot \frac{\partial f_R(U_j^r)}{\partial U_j^r} \right) \right]. \end{aligned} \quad (3.19)$$

よって、法線ベクトルの内積は必ずしも0になるわけではなく、実部と虚部のそれぞれの決定表面は互いに単純に直交するとは言えない。しかしながら、式 (3.19) を注意深くみると、任意の $1 \leq j \leq M$ に対して、

$$\frac{\partial f_R(U_j^r)}{\partial U_j^r} = \frac{\partial f_R(U_j^i)}{\partial U_j^i} \quad (3.20)$$

すなわち、

$$f'_R \left[\sum_{i=1}^L (w_{ji}^r x_i - w_{ji}^i y_i) + \theta_j^r \right] = f'_R \left[\sum_{i=1}^L (w_{ji}^i x_i + w_{ji}^r y_i) + \theta_j^i \right] \quad (3.21)$$

が成り立つならば、内積 = 0 となることがわかる。一般に、 $f'_R(u_1)$ と $f'_R(u_2)$ は $|u_1|$ および $|u_2|$ がある大きな値以上であれば、ほぼ同じ値をとるものとみなせる。故に、任意の $1 \leq j \leq M$ に対して、十分大きな正数 K_1, K_2 が存在して、

$$\begin{aligned} |U_j^r| &= \left| \sum_{i=1}^L (w_{ji}^r x_i - w_{ji}^i y_i) + \theta_j^r \right| \\ &= \left| \sum_{i=1}^L |w_{ji}| |z_i| \cos(\alpha_{ji} + \beta_i) + \theta_j^r \right| \\ &> K_1, \end{aligned} \tag{3.22}$$

$$\begin{aligned} |U_j^i| &= \left| \sum_{i=1}^L (w_{ji}^i x_i + w_{ji}^r y_i) + \theta_j^i \right| \\ &= \left| \sum_{i=1}^L |w_{ji}| |z_i| \sin(\alpha_{ji} + \beta_i) + \theta_j^i \right| \\ &> K_2, \end{aligned} \tag{3.23}$$

ただし、 $\tan(\alpha_{ji}) = w_{ji}^i / w_{ji}^r$, $\tan(\beta_i) = y_i / x_i$ が成り立つとき、2つの決定表面は互いに直交する。つまり、任意の $1 \leq j \leq M$ に対して、中間ニューロン j への総入力（複素数）の実部および虚部の絶対値が共に十分大きいとき、2つの決定表面は互いに直交する。あるいは、さらに、 $|z| = \sqrt{|z_1|^2 + \dots + |z_L|^2}$ が十分大きい値をとる (\mathbf{x}, \mathbf{y}) において、2つの決定表面は互いに直交する 可能性 が高いと言える。

このことを定理および系の形でまとめておく。

定理 2 すべての中間ニューロンへの総入力（複素数）の実部および虚部の絶対値が共に十分大きいとき、実部と虚部のそれぞれの決定表面は互いに直交する。

系 1 $|z|$ が十分大きい値をとる (\mathbf{x}, \mathbf{y}) において、実部と虚部のそれぞれの決定表面は互いに直交する 可能性 が高い。

3.5 シミュレーション

本節では、3層の複素BPネットワークにおける決定表面の特性について計算機実験を通じて調べ、実BPネットワークのものとの比較を行う。出力関数は、シグモイド関数をベースにした複素関数(式(2.18))である。

実験には、表3.1に示す学習パターンを使い、学習パラメータの初期値は -0.3 から $+0.3$ の間の乱数により設定する。

表 3.1 学習パターン

| パターン番号 | 入力パターン | 出力パターン |
|--------|---------|---------|
| 1 | 0 | 0 |
| 2 | i | 1 |
| 3 | 1 | $1 + i$ |
| 4 | $1 + i$ | i |

学習は誤差

$$\sqrt{2 \sum_p E_p} \quad (3.24)$$

が 0.10 となったときに収束（終了）したものとみなし、収束の規範とする（ E_p は 2.4 節で定義したパターン p に対する 2 乗誤差である）。また、表 3.1 に示す 4 種類のパターンをひと通り提示することで 1 回の学習と数えることとする。

ネットワークは、入力 1 出力 1、中間ニューロンは 3 から 6 までの 4 種類、学習率は 0.5 とした。学習終了後、テストパターンとして、実部、虚部、共に、0.0 から 1.0 まで 0.1 刻みの値（合計 $11 \times 11 = 121$ パターン）を入力した。その結果を図 3.4(a)～図 3.7(a) に示す。また、比較のために行った実 BP の結果を図 3.4(b)～図 3.7(b) に示す。複素 BP と異なるのは入力 2 出力 2 というネットワークを使い、入出力データの実部を入出力ニューロン 1、虚部を入出力ニューロン 2 に対応させた点である。図 3.4～図 3.7 において、領域 1 はテストパターンに対するニューラルネットワークの出力の実部がオフ（0.0～0.5）かつ虚部がオフであったことを示す。同様に、領域 2 は実部オン（0.5～1.0）、虚部オフ；領域 3 は実部オン、虚部オン；領域 4 は実部オフ、虚部オンであったことを示す。

図 3.4～図 3.7 において、複素 BP ネットワークの決定表面は実 BP ネットワークのものとは比べて見るとある一定の構造を持っているように見受けられる。特に、図 3.4(a) は実部の決定表面（領域 2 および領域 3 が成す領域とその他の領域との境界線）と虚部の決定表面（領域 3 および領域 4 が成す領域とその他の領域との境界線）とが直交する傾向が見られ（図 3.8 参照）、3.4 節において示した複素 BP ネットワークの決定表面の直交性の性質が反映されているものと思われる。

3.6 むすび

複素 BP の基本特性を、ネットワークアーキテクチャの観点から解析的、実験的に調べ、実 BP との差異を明らかにした。

複素 BP には、複素数への拡張の結果として、次のような性質が生じている。複素 BP ネットワークにおける重みパラメータは互いに独立ではなく 2 次元アフィン変換に基づいた制約を有しており、学習は基本的にその制約のもとで進められる。複素ニューロンの決定表面は 2 つの直交した超曲面（超平面）から構成されている。また、3 層の複素 BP ネットワークにおける決定表面は基本的にこの性質を内包しており、中間ニューロンへの総入力に十分大

きいときに直交する。決定表面の直交性は決定領域を均等に4つに分割しようとするという意味で、一般には $0, 1, i, 1+i$ という4つのクラスを扱う複素パターンに適した性質である。しかし、たとえば、 $0, 0.2, 0.5i, i$ に分類するような問題の場合は有効であるとは限らないので、決定表面の直交性が有効に働くのは問題によると言える。このような意味で複素BPは一般には複素パターンに対する自然な学習アルゴリズムであり、複素パターンを扱う分野において有効に利用され得るものと期待される。

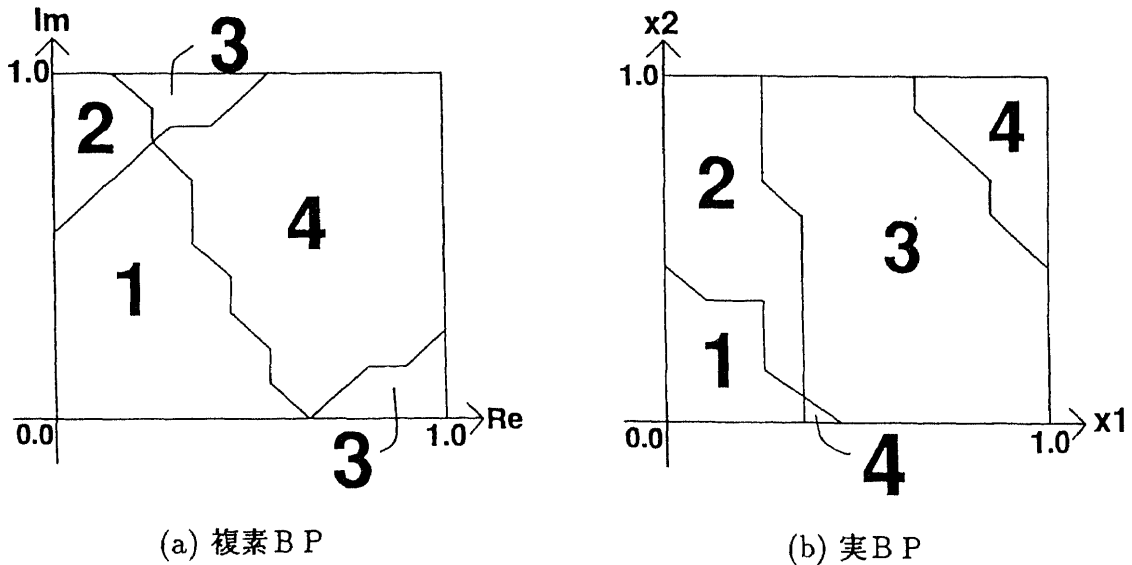


図 3.4 決定表面 (3 中間ニューロン)

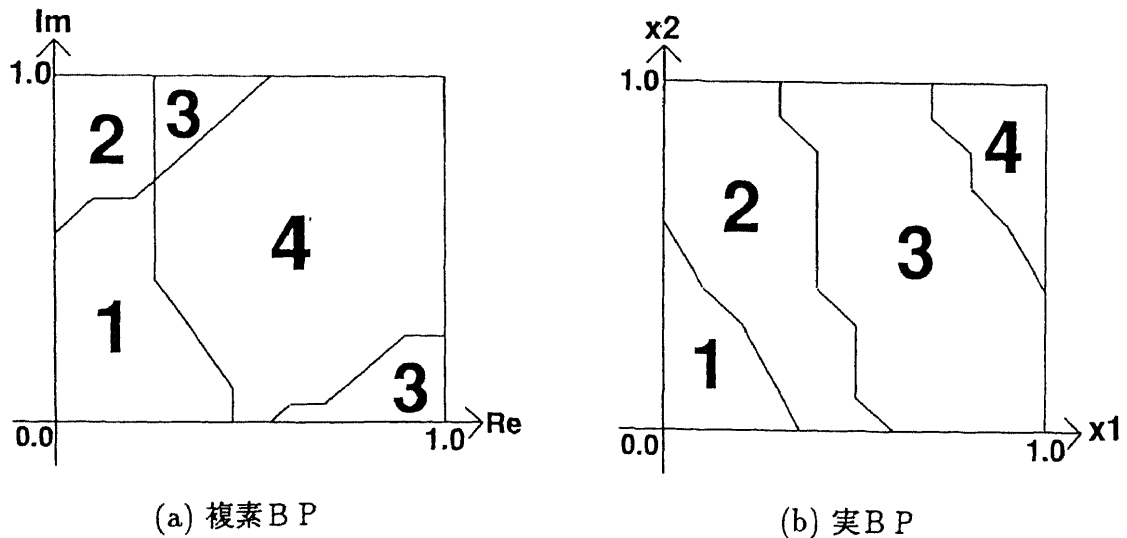
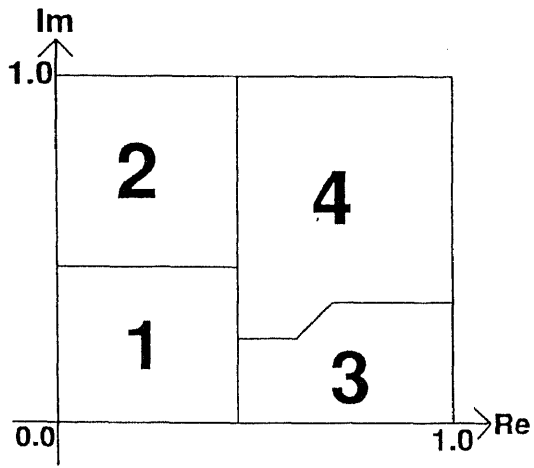
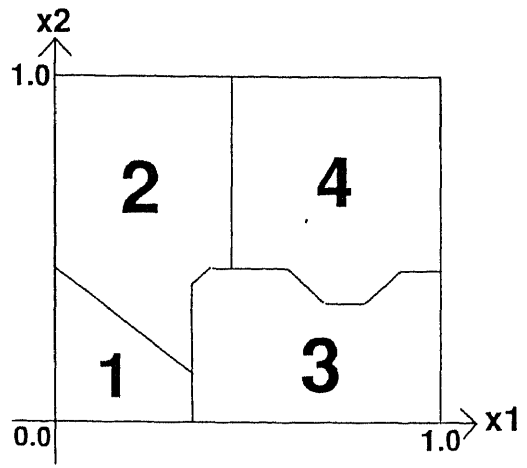


図 3.5 決定表面 (4 中間ニューロン)

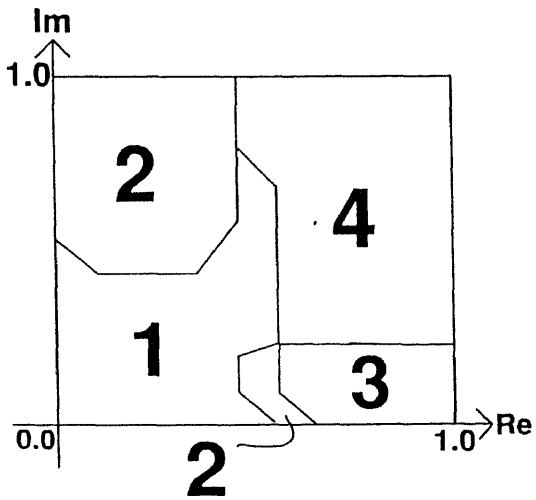


(a) 複素BP

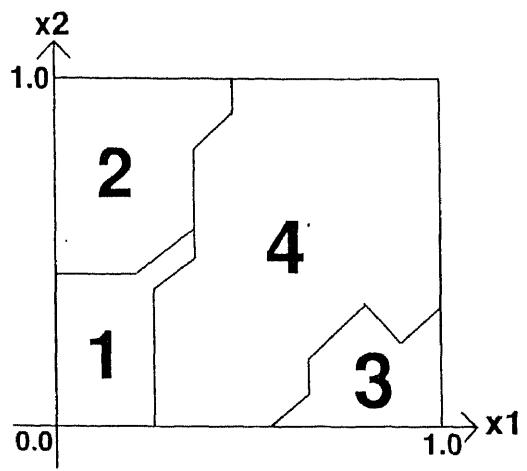


(b) 実BP

図 3.6 決定表面 (5 中間ニューロン)



(a) 複素BP



(b) 実BP

図 3.7 決定表面 (6 中間ニューロン)

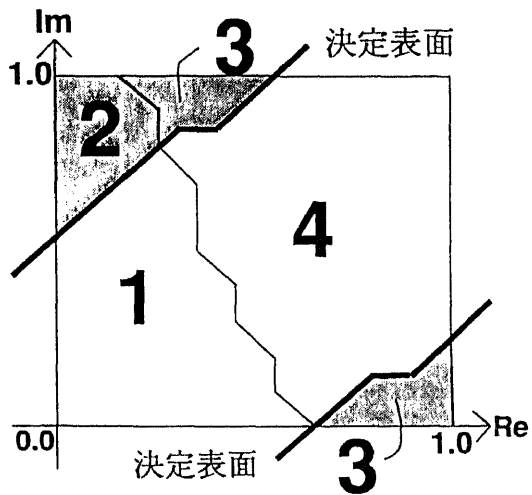
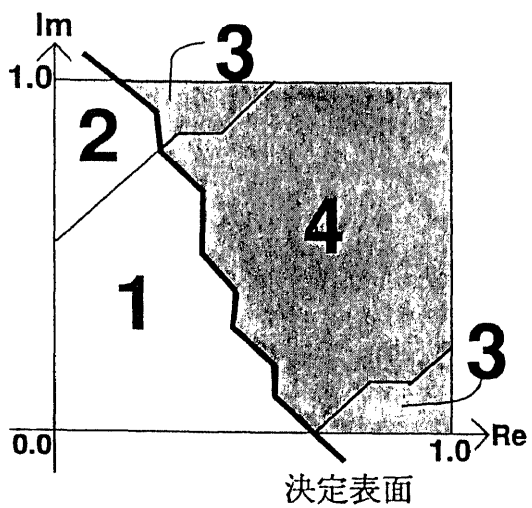


図 3. 4 ~ 3. 8 における数字の意味

- 1 : 実部 0.0 ~ 0.5, 虚部 0.0 ~ 0.5
- 2 : 実部 0.5 ~ 1.0, 虚部 0.0 ~ 0.5
- 3 : 実部 0.5 ~ 1.0, 虚部 0.5 ~ 1.0
- 4 : 実部 0.0 ~ 0.5, 虚部 0.5 ~ 1.0

(a) 出力の実部がオンである領域 (領域 2 + 領域 3)



(b) 出力の虚部がオンである領域 (領域 3 + 領域 4)

図 3. 8 決定表面の直交性

第 4 章

複素バックプロパゲーション学習アルゴリズムの学習特性

4.1 はじめに

本章では主として学習アルゴリズムの観点から複素BPの基本特性を調べる [27, 32, 41, 44, 47]。主な結果は次のとおりである。(1) 複素BPにおける誤差逆伝播は2次元アフィン変換に基づいた構造を有している。(2) 複素BPは、ネットワークを流れる複素信号を1つの処理単位として学習を進めるものである。(3) 複素BPでは、学習則における実部要因と虚部要因の補完構造により、学習停滞状態の発生が押えられる。その結果、実BPに比べて、複素パターンに対する学習が数倍速くなっている(実BPに複素パターンを学習させる方法については後述する)。しかも、そのときに必要となる領域計算量(重みおよび閾値の総数)は約半分済む。これらの意味で、複素BPは複素パターンの学習に適したアルゴリズムであると言える。

以下では、まず、4.2節および4.3節において、複素BP学習アルゴリズムの解析を行い、学習の高速性の要因などの基本特性について調べる。それらの2つの例題を用いた計算機実験の結果を4.4節で示し、結論を4.5節において述べる。

4.2 学習パラメータの修正量の構造

本節では、複素BPにおける学習パラメータの修正量の構造を2.4節で定義した3層ニューラルネットワークを例にとって調べる。すなわち、次のような3層ネットワークを用いる。 w_{ji} を入力ニューロン i と中間ニューロン j との間の結合の重み、 v_{kj} を中間ニューロン j と出力ニューロン k との間の結合の重み、 θ_j を中間ニューロン j が持つ閾値、 γ_k を出力ニューロン k が持つ閾値とする。また、 I_i , H_j , O_k を、それぞれ、入力ニューロン i , 中間ニューロン j , 出力ニューロン k の出力値とする(これらがネットワークを流れる信号である)。そして、出力ニューロン k に対する教師パターン T_k と O_k との誤差を $\delta^k = T_k - O_k$ とする。これらはすべて複素数である。また、 $\varepsilon > 0$ を学習率とする。

学習パラメータ x の修正量を Δx , その実部を Δx^R , 虚部を Δx^I で表すことにすると、

2.5節で導出した複素BPの学習則(式(2.23) - (2.26))は次のように表現できる。

$$\begin{aligned} \begin{bmatrix} \Delta v_{kj}^R \\ \Delta v_{kj}^I \end{bmatrix} &= \begin{bmatrix} \operatorname{Re}[H_j] & \operatorname{Im}[H_j] \\ -\operatorname{Im}[H_j] & \operatorname{Re}[H_j] \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^R \\ \Delta \gamma_k^I \end{bmatrix} \\ &= |H_j| \begin{bmatrix} \cos \beta_j & \sin \beta_j \\ -\sin \beta_j & \cos \beta_j \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^R \\ \Delta \gamma_k^I \end{bmatrix}, \end{aligned} \quad (4.1)$$

$$\begin{bmatrix} \Delta \gamma_k^R \\ \Delta \gamma_k^I \end{bmatrix} = \varepsilon \begin{bmatrix} A_k & 0 \\ 0 & B_k \end{bmatrix} \begin{bmatrix} \operatorname{Re}[\delta^k] \\ \operatorname{Im}[\delta^k] \end{bmatrix}, \quad (4.2)$$

$$\begin{aligned} \begin{bmatrix} \Delta w_{ji}^R \\ \Delta w_{ji}^I \end{bmatrix} &= \begin{bmatrix} \operatorname{Re}[I_i] & \operatorname{Im}[I_i] \\ -\operatorname{Im}[I_i] & \operatorname{Re}[I_i] \end{bmatrix} \begin{bmatrix} \Delta \theta_j^R \\ \Delta \theta_j^I \end{bmatrix} \\ &= |I_i| \begin{bmatrix} \cos \phi_i & \sin \phi_i \\ -\sin \phi_i & \cos \phi_i \end{bmatrix} \begin{bmatrix} \Delta \theta_j^R \\ \Delta \theta_j^I \end{bmatrix}, \end{aligned} \quad (4.3)$$

$$\begin{aligned} \begin{bmatrix} \Delta \theta_j^R \\ \Delta \theta_j^I \end{bmatrix} &= \begin{bmatrix} C_j & 0 \\ 0 & D_j \end{bmatrix} \sum_k \begin{bmatrix} \operatorname{Re}[v_{kj}] & \operatorname{Im}[v_{kj}] \\ -\operatorname{Im}[v_{kj}] & \operatorname{Re}[v_{kj}] \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^R \\ \Delta \gamma_k^I \end{bmatrix} \\ &= \begin{bmatrix} C_j & 0 \\ 0 & D_j \end{bmatrix} \sum_k |v_{kj}| \begin{bmatrix} \cos \varphi_{kj} & \sin \varphi_{kj} \\ -\sin \varphi_{kj} & \cos \varphi_{kj} \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^R \\ \Delta \gamma_k^I \end{bmatrix}, \end{aligned} \quad (4.4)$$

ただし, $A_k = (1 - \operatorname{Re}[O_k])\operatorname{Re}[O_k]$, $B_k = (1 - \operatorname{Im}[O_k])\operatorname{Im}[O_k]$, $C_j = (1 - \operatorname{Re}[H_j])\operatorname{Re}[H_j]$, $D_j = (1 - \operatorname{Im}[H_j])\operatorname{Im}[H_j]$, $\beta_j = \arctan(\operatorname{Im}[H_j]/\operatorname{Re}[H_j])$, $\phi_i = \arctan(\operatorname{Im}[I_i]/\operatorname{Re}[I_i])$, $\varphi_{kj} = \arctan(\operatorname{Im}[v_{kj}]/\operatorname{Re}[v_{kj}])$ である。

まず, Δv_{kj} について見てみる。式(4.1)において, $|H_j|$ は2次元平面上の点の原点との距離に関する縮小・拡大, $\begin{bmatrix} \cos \beta_j & \sin \beta_j \\ -\sin \beta_j & \cos \beta_j \end{bmatrix}$ は原点を中心とした負の方向(時計回り)への β_j 度の回転を意味する。つまり, 式(4.1)では2次元アフィン変換が行われていると言える。よって, 学習パラメータの修正量(複素数)を2次元平面上の点として表現した場合, 中間-出力層の重みの修正量($\Delta v_{kj}^R, \Delta v_{kj}^I$)は, 出力層の閾値の修正量($\Delta \gamma_k^R, \Delta \gamma_k^I$)に上記のアフィン変換を施すことによって得られるというわけである(図4.1)。同様にして, 中間層の閾値の修正量($\Delta \theta_j^R, \Delta \theta_j^I$)は, 出力層の各閾値の修正量($\Delta \gamma_k^R, \Delta \gamma_k^I$)に中間-出力層の重み v_{kj} に関する2次元アフィン変換を施すことによって得られる値に基づいて定まる(式(4.4))。また, 入力-中間層の重みの修正量($\Delta w_{ji}^R, \Delta w_{ji}^I$)は, 中間層の閾値の修正量($\Delta \theta_j^R, \Delta \theta_j^I$)に入力層の出力値 I_i に関する2次元アフィン変換を施すことによって得られる(式(4.3))。要するに, 複素BPにおける誤差逆伝播は 2次元アフィン変換に基づいた構造 を有している

と言える。

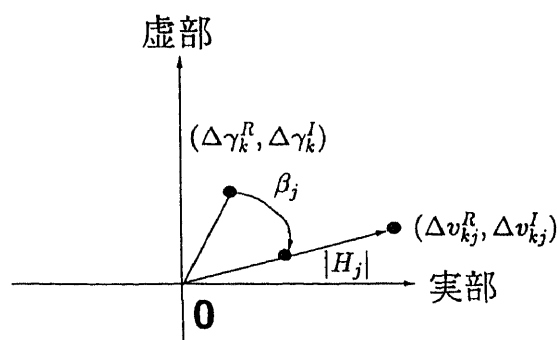


図 4.1 複素BPにおける誤差逆伝播のイメージ

ここで、重みパラメータ自身には、第3章で見たように、2次元アフィン変換に関する制約が課せられており、学習はその制約を保ちながら進められる(ただし、各ニューロンにおいてはシグモイド関数が適用されるので、各ニューロンの出力値に2次元アフィン変換に関する制約が厳密に反映されるのは、シグモイド関数が線形変換とみなせるとき、すなわち、ニューロンへの入力(複素数)の実部および虚部のそれぞれの絶対値が共に小さいときに限る)。よって、複素BPでは、学習パラメータ自身およびその修正量には2次元アフィン変換に関係した構造が備わっていると言える。ただし、学習パラメータ自身に課せられている制約は 正の方向 への回轉變換

$$P = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \quad (4.5)$$

に関係しており、学習パラメータの修正量に課せられている制約は 負の方向 への回轉變換

$$Q = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} \quad (4.6)$$

に関係している。ここで、 P および Q は、共に、直交行列であり、

$$Q^{-1} = {}^tQ = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \quad (4.7)$$

を満たし、 Q の逆行列 Q^{-1} および転置行列 tQ は P と同様な形をしている。

ここで、一方の実BPにおける学習パラメータの修正量の構造について確認しておく。実BPにおける学習パラメータの修正量は、

$$\Delta v_{kj} = H_j \Delta \gamma_k, \quad (4.8)$$

$$\Delta \gamma_k = \varepsilon (1 - O_k) O_k \delta^k, \quad (4.9)$$

$$\Delta w_{ji} = I_i \Delta \theta_j, \quad (4.10)$$

$$\Delta \theta_j = (1 - H_j) H_j \sum_k v_{kj} \Delta \gamma_k \quad (4.11)$$

で与えられる（ただし、各パラメータはすべて実数であり、その定義は複素BPのパラメータに対応する）。学習パラメータの修正量（実数）を1次元直線上の点として表現した場合、中間—出力層の重みの修正量 Δv_{kj} は出力層の閾値の修正量 $\Delta \gamma_k$ を原点との距離に関して H_j 倍することによって得られる（式(4.8), 図4.2）。同様に、中間層の閾値 $\Delta \theta_j$ は、出力層の各閾値の修正量 $\Delta \gamma_k$ をそれぞれ原点との距離に関して v_{kj} 倍した値に基づいて定まる（式(4.11)）。また、入力—中間層の重みの修正量 Δw_{ji} は、中間層の閾値の修正量 $\Delta \theta_j$ を原点との距離に関して I_i 倍することによって得られる（式(4.10)）。つまり、実BPにおける誤差逆伝播は 1次元直線上の点の移動に基づいた構造 をしていると言える。このように、実BPが複素数へ拡張されることによって、誤差逆伝播の構造が 1次元的 なものから 2次元的 なものへと変質していることがわかった。

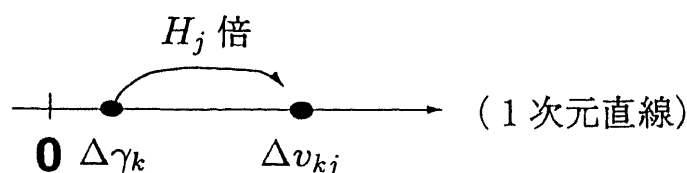


図4.2 実BPにおける誤差逆伝播のイメージ

また、見方を変えると、誤差逆伝播の2次元的な構造に伴い、複素BPはネットワークを流れる複素信号を1つの処理単位として学習を進めているということがわかる。たとえば、 Δv_{kj}^R および Δv_{kj}^I は、ネットワークを流れる信号の実部 ($Re[H_j], Re[O_k]$) および虚部 ($Im[H_j], Im[O_k]$) を共に含んでおり、これらに基づいて値が定まる（式(4.1)）。つまり、 Δv_{kj}^R は複素信号の実部 ($Re[H_j], Re[O_k]$) だけから、 Δv_{kj}^I は虚部 ($Im[H_j], Im[O_k]$) だけから、という具合にお互いに独立して決まるのではなく、ネットワークを流れる信号の実部および虚部の両方の値を通じて依存し合いながら定まるものであると言える。入力—中間層の結合の重み w_{ji} の実部の修正量 Δw_{ji}^R および虚部の修正量 Δw_{ji}^I 、中間層の閾値 θ_j の実部の修正量 $\Delta \theta_j^R$ および虚部の修正量 $\Delta \theta_j^I$ についても同様に、ネットワークを流れる信号の実部、虚部の両方の値を含んでおり、それらを通して影響を与え合っている（式(4.3)および式(4.4)）。また、出力層の閾値 γ_k の実部の修正量 $\Delta \gamma_k^R$ は、直接的には、ネットワークを流れる信号（正確には、ネットワークが出力する信号） O_k の実部 ($Re[O_k]$) だけから定まり、虚部の修正量 $\Delta \gamma_k^I$ は虚部 ($Im[O_k]$) だけから決定される（式(4.2)）。ここで、 $Re[O_k]$ についてみると、

$$Re[O_k] = f_R(Re[S_k]) \quad (4.12)$$

であり、

$$Re[S_k] = \sum_j \{ Re[v_{kj}] Re[H_j] - Im[v_{kj}] Im[H_j] \} + Re[\gamma_k] \quad (4.13)$$

であるから、 $Re[O_k]$ は $Re[H_j]$ と $Im[H_j]$ の関数である (S_k は出力ニューロン k への入力)

総和)。一方,

$$Im[O_k] = f_R(Im[S_k]) \quad (4.14)$$

であり,

$$Im[S_k] = \sum_j \{ Re[v_{kj}] Im[H_j] + Im[v_{kj}] Re[H_j] \} + Im[\gamma_k] \quad (4.15)$$

であるので, $Im[O_k]$ は, やはり, $Re[H_j]$ と $Im[H_j]$ の関数である。故に, $\Delta\gamma_k^R$ と $\Delta\gamma_k^I$ は共に $Re[H_j], Im[H_j]$ の関数である。つまり, $\Delta\gamma_k^R$ と $\Delta\gamma_k^I$ は, $Re[H_j]$ と $Im[H_j]$ を通じて (間接的にはあるが) 依存し合いながら値が定まると言える。

このように, 複素BPにおける学習パラメータの実部および虚部の値の修正は, それぞれ, ネットワークを流れる信号の実部および虚部の両方の値に基づいて, そして, それらを通じて相互に依存し合いながら行われる (図 4.3)。要するに, 複素BPは, ネットワークを流れる信号 (複素数) を1つの処理単位として学習を進めるものであると言える。

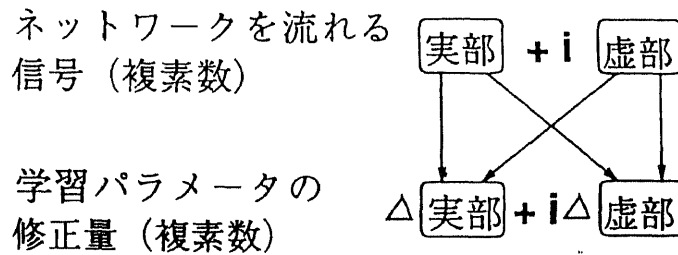


図 4.3 学習パラメータの修正量の決定要因. 矢印の始点は, 矢印の終点の決定要因である。

4.3 学習の高速性の要因

4.2節において見たように, 複素BPにおける誤差逆伝播は2次元アフィン変換に基づいた構造を有しており, その結果, 複素BP学習はネットワークを流れる複素信号を1つの処理の単位としていた。本節では, 更に, このことがその構造において, 学習の高速化を促すものであることを示す。

実BPの学習パラメータの修正量 (式 (4.8) - (4.11)) において, $(1-H_j)H_j \in \mathbf{R}$ および $(1-O_k)O_k \in \mathbf{R}$ は, 各ニューロンの出力関数であるシグモイド関数 $f_R(u) = 1/(1 + \exp(-u))$ の導関数 $(1-f_R(u))f_R(u)$ であり, 図 4.4 のような曲線を描くものである。つまり, ニューロンへの入力の総和 u の絶対値が大きくなるにつれて0に近づいてゆく関数である。よって, ニューロンの出力値を0もしくは1に少しでも近づけるために, ニューロンへの入力の総和 u の絶対値が大きくなるとシグモイド導関数 $(1-f_R(u))f_R(u)$ は微小な値を取ることであり, その結果, 学習は進みにくくなる。その際, 特に, 誤差が大きな値であり, そのような状態が

長く続く場合には、“ローカルミニマムに陥った”と称されることになる。よく知られているように、これが実BPにおける学習停滞のメカニズムである。

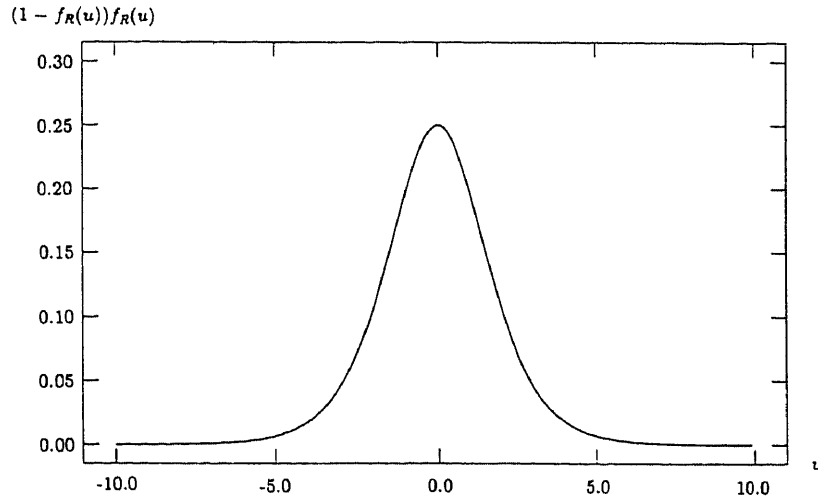


図4.4 シグモイド関数 $f_R(u)$ の導関数

一方、複素BPにおいては、学習パラメータの修正量(式(4.1) - (4.4))には2種類のシグモイド導関数が現れる。1つは出力関数の実部の導関数 $(1 - \text{Re}[O_k])\text{Re}[O_k]$, $(1 - \text{Re}[H_j])\text{Re}[H_j]$, もう一つは虚部の導関数 $(1 - \text{Im}[O_k])\text{Im}[O_k]$, $(1 - \text{Im}[H_j])\text{Im}[H_j]$ であり、複素BPの学習則は基本的にこれらの2つの線形結合

$$\alpha_1(1 - \text{Re}[O_k])\text{Re}[O_k] + \beta_1(1 - \text{Im}[O_k])\text{Im}[O_k], \quad (4.16)$$

$$\alpha_2(1 - \text{Re}[H_j])\text{Re}[H_j] + \beta_2(1 - \text{Im}[H_j])\text{Im}[H_j] \quad (4.17)$$

から構成されている ($\alpha_k, \beta_k \in \mathbf{R}$ ($k=1, 2$))。ここで、式(4.16)が微小な値になるのは、 $(1 - \text{Re}[O_k])\text{Re}[O_k]$ と $(1 - \text{Im}[O_k])\text{Im}[O_k]$ の両方の値が微小になるときである(式(4.17)についても同様)。よって、たとえば、 $(1 - \text{Re}[O_k])\text{Re}[O_k]$ が微小な値をとったとしても、 $(1 - \text{Im}[O_k])\text{Im}[O_k]$ もそうであるとは限らないので、式(4.16)が微小値をとらない可能性があり、この点で、 $(1 - O_k)O_k \in \mathbf{R}$ が微小値をとれば必ず学習パラメータの修正量が微小になってしまう実BPの学習則(式(4.8) - (4.11))とは異なっている。言い換えると、実部要因 $\left((1 - \text{Re}[O_k])\text{Re}[O_k], (1 - \text{Re}[H_j])\text{Re}[H_j] \right)$ と虚部要因 $\left((1 - \text{Im}[O_k])\text{Im}[O_k], (1 - \text{Im}[H_j])\text{Im}[H_j] \right)$ は、学習パラメータの修正量が異常に小さくならないようにお互いに補いつ合っているとも言える。このように、複素BPの学習則は、実BPに比べて、学習の停滞状態が発生しにくい構造を有していると言える(注意：特に、 $(1 - \text{Re}[O_k])\text{Re}[O_k]$ および $(1 - \text{Im}[O_k])\text{Im}[O_k]$ が共に微小な値になることが稀であるときに顕著であろう)。この結果、複素BPは、実BPに比べて、学習が高速化されているものと思われる。また、学習曲線の形についても興味のあるところである。これらのことについては、4.4節において複素パターンの

学習に関する計算機実験を通じて確認する。

従来、学習の高速化手法には、出力関数側からのアプローチと損失関数側からのアプローチが提案されており、それぞれ、効果をあげている。前者には、学習停滞状態に陥ったならば、シグモイド関数の傾きを強制的に修正することにより学習パラメータの修正量を増やすという手法 [61]、あるいは出力関数を区分的線形関数とし、その傾きを学習の対象として取り込む方法 [52] などがある。後者（損失関数側からのアプローチ）には、損失関数に Kullback Divergence を用いる方法 [56]、損失関数にエントロピー項を付加する方法 [3] などがある。複素BPは、これらのように学習の高速化を意図されたものではなかったが、結果的に高速化を促進する構造が自然に備わっているのである。その構造は出力関数側からのアプローチに属するものであり、出力関数の実部と虚部の導関数の線形結合性に基いている。よって、上述の従来の高速化手法のうち、損失関数側からのアプローチのものとは観点が異なっており、複素BPと組み合わせて適用することが可能である。また、複素BPの出力関数の実部と虚部は、実BPの出力関数と同じシグモイド関数である(式(2.18)参照)。よって、上述の従来の高速化手法のうち、出力関数側からのアプローチのものを複素BPの出力関数の実部と虚部に対して、それぞれ、適用することにより、更なる高速化が可能であると思われる。

4.4 シミュレーション

本節では、複素パターンを使った計算機実験を通じて、4.3節において示した複素BPの学習特性の確認を行う。

一般に、学習速度の評価は単純に学習回数だけからではなく、計算量 [50] の観点からトータルに行なう必要がある。計算量には、時間計算量 (time complexity) と 領域計算量 (space complexity) の2種類がある。時間計算量はアルゴリズムの中の特定作業の実行回数のことであり、消費時間の目安にされるものである。一方、領域計算量はアルゴリズムの中で使用される変数の個数のことであり、消費記憶容量の目安にされるものである。

本論文では、アルゴリズムの中で行なわれる(実数の)四則演算の総回数を時間計算量、学習パラメータ数を領域計算量とする。ただし、時間計算量において、(1回の加算の計算量)と(1回の減算の計算量)は等しいものとする。また、一般に、(1回の乗算の計算量) \geq (1回の加算の計算量)であるが、ここでは念の為、(1回の加算の計算量) = (1回の乗算の計算量) および (1回の加算の計算量) = 0 という極端な2つのケースを考える。なお、除算は実BPおよび複素BPでは使用されないのので常に0回である。

次の4.4.1項および4.4.2項では、1回の学習の時間計算量が実BPと複素BPとで等しくなるようにネットワークの構造を調節したうえで、それぞれの収束するまでに必要な平均学習回数を基準にして学習速度の評価を行なう。加えて、そのときの領域計算量も評価する。

実験において、学習パラメータの初期値は、本節を通して、 -0.3 から $+0.3$ の間の乱数により設定する。学習は誤差

$$\sqrt{2 \sum_p E_p} \quad (4.18)$$

が 0.10 となったときに収束(終了)したものとみなし、収束の規範とする (E_p は 2.4 節で

定義したパターン p に対する 2 乗誤差である)。また、学習パターンは、通常、数種類のパターンから構成されているが、その数種類のパターンをひと通り提示することで 1 回の学習と数えることとする。

4.4.1 例題 1

本項では、複素パターンから構成される単純な学習パターン (表 4.1) を使った実験の結果を示す。複素 BP には 1-3-1 ネットワーク、実 BP には 2-6-2 ネットワークおよび 2-7-2 ネットワークを使った。複素 BP および実 BP をこれらのネットワークに適用するときのそれぞれの計算量を表 4.2 に示す。学習 1 回あたりの時間計算量に関して、複素 BP 1-3-1 ネットワークは、(1 回の加算の計算量) = 0 のとき、実 BP 2-6-2 ネットワークに等しく、(1 回の加算の計算量) = (1 回の乗算の計算量) のとき、実 BP 2-7-2 ネットワークにほぼ等しい。

表 4.1 学習パターン [例題 1]

| パターン番号 | 入力パターン | 出力パターン |
|--------|--------|--------|
| 1 | 0 | 0 |
| 2 | i | 1 |
| 3 | 1 | $1+i$ |
| 4 | $1+i$ | i |

表 4.2 複素 BP および実 BP の計算量 [例題 1]

| ネットワーク | 時間計算量 | | | 領域計算量 | | |
|-------------|-------|-----|-----|-------|----|----|
| | 乗除算 | 加減算 | 計 | 重み | 閾値 | 計 |
| 複素 BP 1-3-1 | 78 | 52 | 130 | 12 | 8 | 20 |
| 実 BP 2-6-2 | 78 | 40 | 118 | 24 | 8 | 32 |
| 実 BP 2-7-2 | 90 | 46 | 136 | 28 | 9 | 37 |

実 BP に複素パターンを学習させる方法の 1 つは、入出力層において実部、虚部を担当させるニューロンをあらかじめ決めておくことである。ここでは、入出力される複素数の実部を入出力ニューロン 1、虚部を入出力ニューロン 2 に担当させた。

学習率をパラメータとして 0.1 から 0.6 まで 0.1 刻みで変化させた。各学習率について、それぞれ 50 回の試行を行い、それらの収束に要した学習回数の平均を評価の基準とした。学習は 5 万回で打ち切りとしたが、すべての試行は収束した。図 4.5 がその実験結果である。また、学習曲線 (学習率は 0.1) の一例を図 4.6 に示す (実 BP は 2-7-2 ネットワーク)。

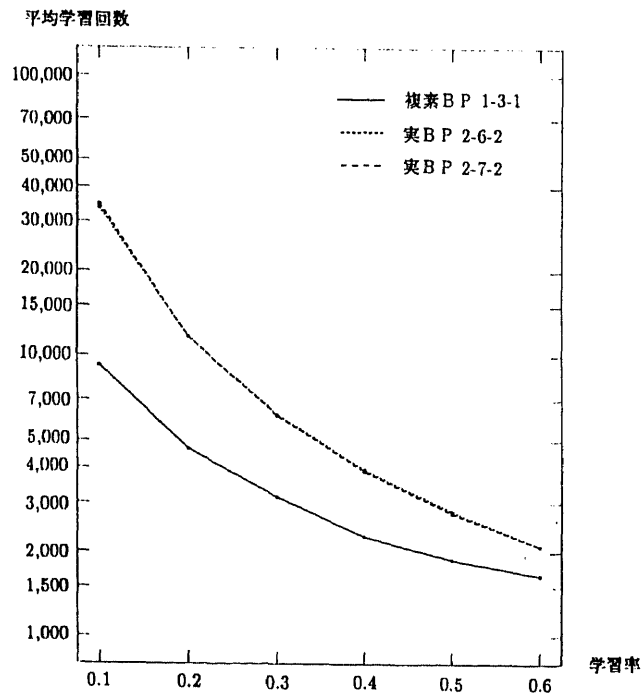


図4.5 平均学習速度(複素BPと実BPの比較)[例題1]

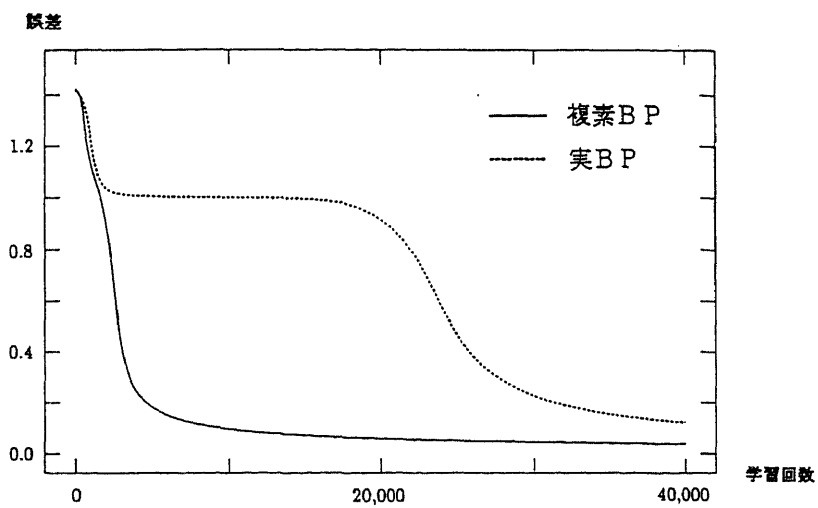


図4.6 学習曲線(複素BPと実BPの比較)[例題1]

さらに、4.3節で述べた「学習の高速性の要因」を裏付けるために、上述の実験における学習パラメータの修正量の絶対値の平均値を調べた。その結果を表4.3に示す。複素BP、実BP共に学習率0.5の場合について、学習回数が750回の時点での値を調べた(ただし、パターン1に対応するデータに基づいている)。なお、実BPネットワークの代表としては2-7-2ネットワークを選んだ。複素BPについては、出力ニューロン1の出力関数の導関数値の実部 $(1 - \text{Re}[O_1])\text{Re}[O_1]$ が0.10以上、かつ、虚部 $(1 - \text{Im}[O_1])\text{Im}[O_1]$ が0.05以下であるような試行についての平均をとった。このようにしたのは、出力関数の導関数値の実部が比較的大きく、かつ、虚部が微小な値となっている試行だけを調査の対象とするためである。そのような試行は50回のうち37回あった。対象とした学習パラメータは中間ニューロン2と出力ニューロン1との間の重みパラメータ w_{12} である。実BPについては、出力ニューロン1の出力関数の導関数値 $(1 - O'_1)O'_1$ が0.10以上、かつ、出力ニューロン2の出力関数の導関数値 $(1 - O'_2)O'_2$ が0.05以下であるような試行についての平均をとった (O'_k は出力ニューロン k の出力値を表す)。そのような試行は50回のうち31回あった。対象とした学習パラメータは中間ニューロン2と出力ニューロン1との間の重みパラメータ w'_{12} および中間ニューロン2と出力ニューロン2との間の重みパラメータ w'_{22} である。表4.3からわかるように、実BPの場合、出力ニューロンの出力関数の導関数値が小さな値をとるとそれに対応する学習パラメータの修正量はそれに応じて小さな値をとってしまう。これに対して、複素BPでは出力関数の導関数値の実部、虚部のどちらか一方が小さな値をとった場合でも、それらに対応する学習パラメータの修正量は、(実BPに比べて収束がある程度進んでいるため値はいくぶん小さいが) 実部、虚部共に同程度の値をとっており、学習停滞状態は発生しにくくなっていることがうかがえる。

表4.3 学習パラメータの修正量の絶対値の平均値 [例題1]

| ネットワーク | 学習パラメータ | 修正量の絶対値の平均値 |
|-------------|---------------------|-------------|
| 複素 BP 1-3-1 | $\text{Re}[w_{12}]$ | 0.00615 |
| | $\text{Im}[w_{12}]$ | 0.00521 |
| 実 BP 2-7-2 | w'_{12} | 0.02330 |
| | w'_{22} | 0.00056 |

4.4.2 例題2

次に、表4.4に示す学習パターンを使った実験を行った。この学習パターンは複素パターンから構成されており、次の2つの規則に従っている。

(1) 入力される複素数1と複素数2が等しいならば、出力される複素数3の実部は1、そうでなければ0。

(2) 入力される複素数2が1または i ならば、出力される複素数3の虚部は1、そうでなければ0。

例題1と異なるのは、複素BPには2-4-1ネットワーク、実BPには4-8-2ネットワークおよび4-9-2ネットワークを使った点である。これらのネットワークが使われる際のそれぞれの計算量を表4.5に示す。学習1回あたりの時間計算量に関して、複素BP 2-4-1ネットワークは、(1回の加算の計算量)=0のとき、実BP 4-8-2ネットワークに等しく、(1回の加算の計算量)=(1回の乗算の計算量)のとき、実BP 4-9-2ネットワークに等しい。

表4.4 学習パターン[例題2]

| パターン番号 | 入力パターン | | 出力パターン |
|--------|--------|-------|--------|
| | 複素数1 | 複素数2 | 複素数3 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | i | i |
| 3 | i | i | $1+i$ |
| 4 | i | 1 | i |
| 5 | 1 | 1 | $1+i$ |
| 6 | i | 0 | 0 |
| 7 | $1+i$ | $1+i$ | 1 |
| 8 | $1+i$ | i | i |

表4.5 複素BPおよび実BPの計算量[例題2]

| ネットワーク | 時間計算量 | | | 領域計算量 | | |
|------------|-------|-----|-----|-------|----|----|
| | 乗除算 | 加減算 | 計 | 重み | 閾値 | 計 |
| 複素BP 2-4-1 | 134 | 92 | 226 | 24 | 10 | 34 |
| 実BP 4-8-2 | 134 | 68 | 202 | 48 | 10 | 58 |
| 実BP 4-9-2 | 150 | 76 | 226 | 54 | 11 | 65 |

実BPに上述の学習パターンを与える際には、例題1と同様にして、入出力層において実部、虚部を担当させるニューロンをあらかじめ決めておいた。すなわち、入力される複素数1の実部、虚部、複素数2の実部、虚部を、この順番で、入力ニューロン1, 2, 3そして4に担当させた。また、出力される複素数3の実部は出力ニューロン1, 虚部は出力ニューロン2に担当させた。

学習は10万回で打ち切りとした。図4.7に実験結果を示す。参考のために、各ネットワークでの収束率(50回の試行のうち収束した試行の割合)を表4.6に示しておく(注意:この場合、収束率に関してみると学習率を0.6以上に選ぶと実BPの方が良い結果が得られる可能性があるように思われる)。また、学習曲線(学習率は0.3)の一例を図4.8に示す(実BPは4-9-2ネットワーク)。

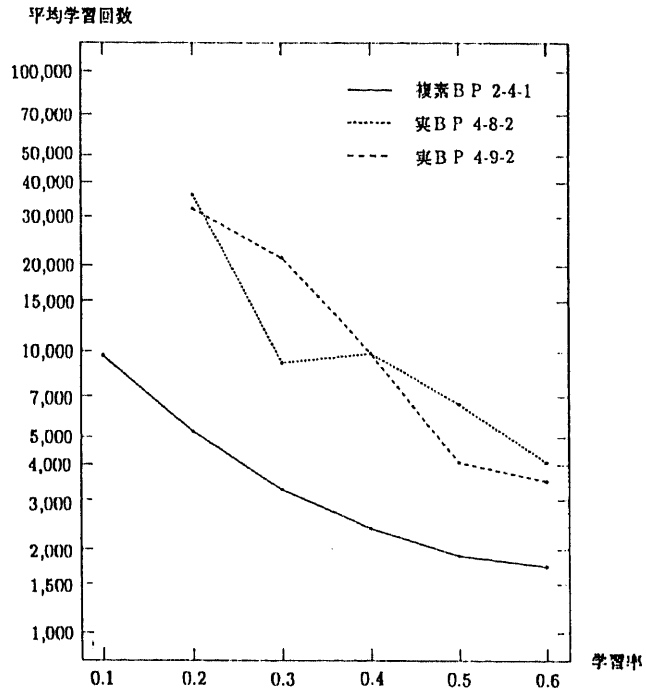


図 4.7 平均学習速度 (複素BP と実BP の比較) [例題 2]

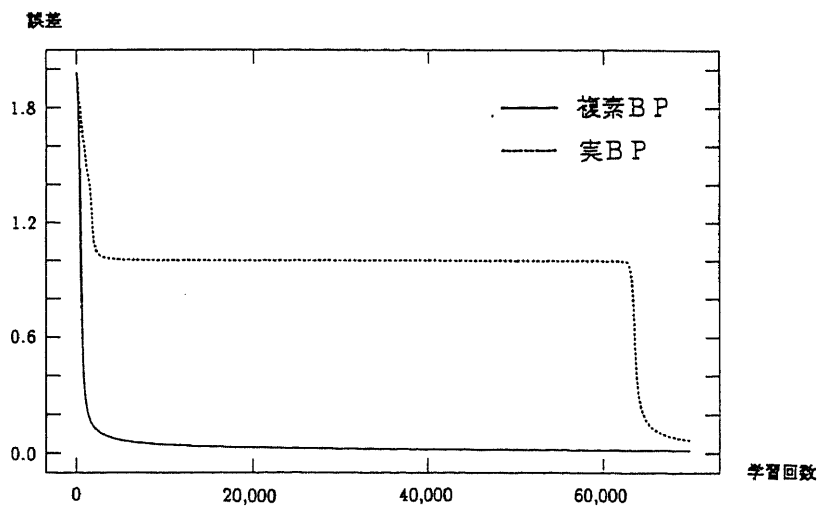


図 4.8 学習曲線 (複素BP と実BP の比較) [例題 2]

さらに、例題1と同様に、4.3節で述べた「学習の高速性の要因」を裏付けるために、上述の実験における学習パラメータの修正量の絶対値の平均値を調べた。その結果を表4.7に示す。複素BP、実BP共に学習率0.5の場合について、学習回数が450回の時点での値を調べた(ただし、パターン6に対応するデータに基づいている)。なお、実BPネットワークの代表としては4-9-2ネットワークを選んだ。複素BPについては、出力ニューロン1の出力関数の導関数値の実部 $(1 - \text{Re}[O_1])\text{Re}[O_1]$ が0.05以下、かつ、虚部 $(1 - \text{Im}[O_1])\text{Im}[O_1]$ が0.10以上であるような試行についての平均をとった。このようにしたのは、出力関数の導関数値の実部が微小であり、かつ、虚部が比較的大きな値となっている試行だけを調査の対象とするためである。そのような試行は50回のうち17回あった。対象とした学習パラメータは中間ニューロン2と出力ニューロン1との間の重みパラメータ w_{12} である。実BPについては、出力ニューロン1の出力関数の導関数値 $(1 - O'_1)O'_1$ が0.05以下、かつ、出力ニューロン2の出力関数の導関数値 $(1 - O'_2)O'_2$ が0.10以上であるような試行についての平均をとった (O'_k は出力ニューロン k の出力値を表す)。そのような試行は50回のうち21回あった。対象とした学習パラメータは中間ニューロン2と出力ニューロン1との間の重みパラメータ w'_{12} および中間ニューロン2と出力ニューロン2との間の重みパラメータ w'_{22} である。表4.7からわかるように、複素BPでは実BPと異なり、出力関数の導関数値の実部、虚部のどちらか一方が小さな値をとった場合でも、それらに対応する学習パラメータの修正量は、例題1と同様に、実部、虚部共に同程度の値をとっており、学習停滞状態は発生しにくくなっていることがうかがえる。

表4.6 収束率 [例題2]

| ネットワーク | 学習率 | | | | | |
|------------|-----|-----|-----|-----|-----|-----|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| 複素BP 2-4-1 | 100 | 96 | 88 | 92 | 90 | 98 |
| 実BP 4-8-2 | 0 | 48 | 70 | 84 | 94 | 94 |
| 実BP 4-9-2 | 0 | 22 | 64 | 78 | 90 | 100 |

表4.7 学習パラメータの修正量の絶対値の平均値 [例題2]

| ネットワーク | 学習パラメータ | 修正量の絶対値の平均値 |
|------------|---------------------|-------------|
| 複素BP 2-4-1 | $\text{Re}[w_{12}]$ | 0.00331 |
| | $\text{Im}[w_{12}]$ | 0.00513 |
| 実BP 4-9-2 | w'_{12} | 0.00038 |
| | w'_{22} | 0.00906 |

4.4.3 考察

4.4.1項および4.4.2項の実験結果により、複素BPには複素パターンの学習に関する以下の特徴が備わっていることがわかる。

(1) 複素BPは、実BPに比べて、学習が数倍速くなっている(図4.5, 図4.7)。しかも、そのときの領域計算量は約半分で済む(表4.2, 表4.5)。

(2) 複素BPの学習曲線には、実BPとは異なり、学習の停滞状態(誤差が大きく、かつ、 x 軸にはほぼ平行な部分)が見当たらない(図4.6, 図4.8)。

これらは4.3節で述べた出力関数の導関数の実部と虚部の(学習則における)線形結合性(式(4.16)および式(4.17))による学習停滞状態の解消構造が、学習速度および学習曲線に現れたものであると考えられる。

4.5 むすび

複素BPの基本特性に対して、学習アルゴリズムの側面からアプローチし、次の結果を得た。

複素数への拡張は複素BPに次のような性質をもたらしている。複素BPにおける誤差逆伝播は、2次元アフィン変換に関係した構造を有している。複素BPは、ネットワークを流れる複素信号を1つの処理単位として学習を進める。複素BPの学習則には、学習停滞状態の発生を抑えるメカニズム(実部要因と虚部要因の補完機構)が含まれている。これにより、複素パターンに対する学習が実BPに比べて数倍速くなっている。しかも、そのときに必要となる領域計算量は約半分で済む。

これらの意味で複素BPは複素パターンを自然に学習することができるものであり、信号処理、電気工学、制御工学などの複素パターンを操作する分野において有効に利用され得るものと期待される。

第 5 章

複素バックプロパゲーション学習アルゴリズムの 2 次元アフィン変換学習能力

5.1 はじめに

本章では、第 2 章で提案した複素 BP が、通常のニューラルネットワークには見られない 2 次元アフィン変換学習能力（図形変換能力） を備えていることを計算機実験によりいくつかの例を通して確認する [21, 23, 26, 34, 36, 38]。また、2 次元アフィン変換を学習した複素 BP ネットワーク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も言及する。一致の定理は複素数世界においてのみ生ずる特異な性質を明らかにしたものである。そして、「誤差の性質」および「学習パラメータの値」についても評価し、さらに、コンピュータビジョンへの応用についても触れる。

以下では、まず、5.2 節において計算機実験の諸条件について述べる。次に、5.3 節および 5.4 節において、それぞれ、単純な変換、やや複雑な変換を通して、2 次元アフィン変換学習能力が備わっていることを示す。5.5 節において一致の定理との類推について触れ、5.6 節においてコンピュータビジョンへの応用について述べる。最後に結論を 5.7 節において述べる。

5.2 実験の諸条件

本節では、複素 BP が 2 次元アフィン変換学習能力を持っていることを確認するために以下で行なう実験の諸条件について述べる。

実験に用いる複素 BP ネットワークは、入力層 1 ニューロン、中間層 6 ニューロン、出力層 1 ニューロンの 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 による実験も行ったが、そのネットワークは入力層 2 ニューロン、中間層 1 2 ニューロン、出力層 2 ニューロンとし、4 章と同様に複素数 z の実部 $\text{Re}[z]$ を入出力ニューロン 1 に、

虚部 $Im[z]$ を入出力ニューロン 2 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、結合の重みおよび閾値は 0 と 1 の間の乱数により設定した。

5.3 単純な変換

本節では、複素BPに2次元アフィン変換学習能力が備わっていることを単純な例を使って示す。さらに、「その際に生じる誤差」および「学習パラメータの値との関係」についても調べる。

5.3.1 回転

まず、直線の回転を行った(図5.1)。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(●印)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パターン(○印)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(▲印)として直線 $y = 0.2$ ($-0.9 \leq x \leq 0.3$) 上の7つの点(図5.1(a))および直線 $y = -x + 0.5$ ($0 \leq x \leq 0.5$) 上の6個の点(図5.1(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(□印)。同様にして、実BP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(■印)を出力した(図5.1)。

次に、斜文字の回転を行った(図5.2, 図5.3)。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(●印)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(○印)とした(図5.2(a))。学習の後、テストパターン(▲印)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および実BP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5.2(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では正の方向に45度回転したが(□印)、実BP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(■印)、SおよびOの形状はくずれた(図5.3)。実BP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り、位置的に近かったからであると思われる。

5.3.2 相似変換

次に、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5.4(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(●印)とし、それらの点と原点との距離を1/2にした点(○印)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12個の点(▲印)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(□印)を出力した。因みに、実B

Pでは直線 $y = x$ 上の点 (■印) 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 5.4(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 (●印) 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 (○印) として与えることにより, 1 辺 0.3 の正方形の拡大を行なった。図 5.4(c) がその結果である。実 BP ネットワークは直線 $y = x$ 上の点列 (■印) を出力したが, 複素 BP 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 (□印)。

5.3.3 平行移動

直線の平行移動を行った (図 5.5(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 (●印) 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 (○印) 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 (▲印) 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 (□印) が得られたが, 実 BP による実験においては平行移動は行われなかった (■印)。

次に, 任意の図形について実験したところ, 図 5.5(b) に示すとおり, 複素 BP ではうまく平行移動させることができた。

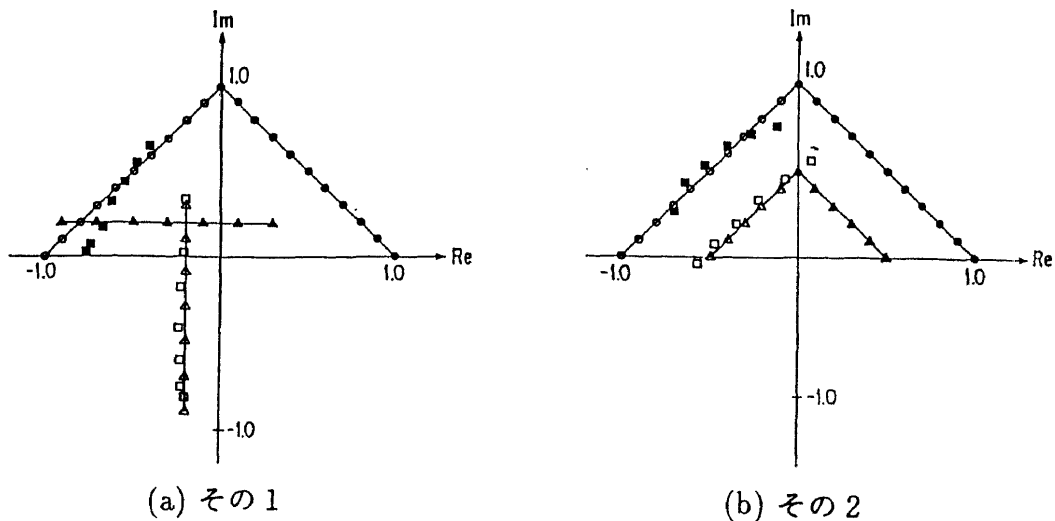
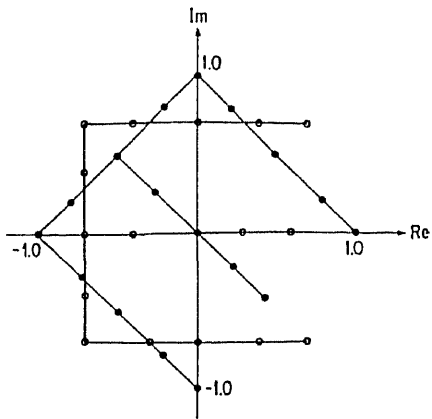


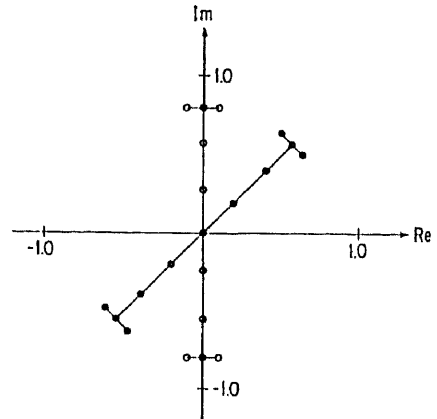
図 5.1 直線の回転

本章の図における記号の意味

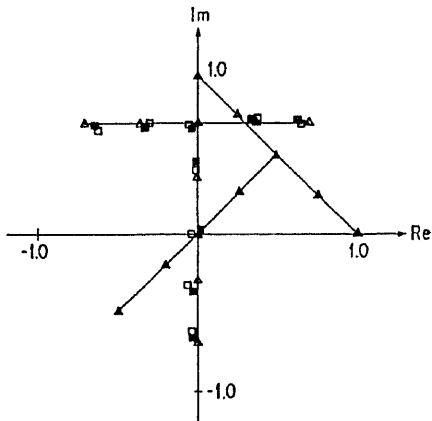
- 入力学習パターン
 - 出力学習パターン
 - ▲ 入力テストパターン
 - △ 期待される出力テストパターン
 - 実 BP 学習による出力テストパターン
 - 複素 BP 学習による出力テストパターン
- 添数はパターンを識別するための番号である。



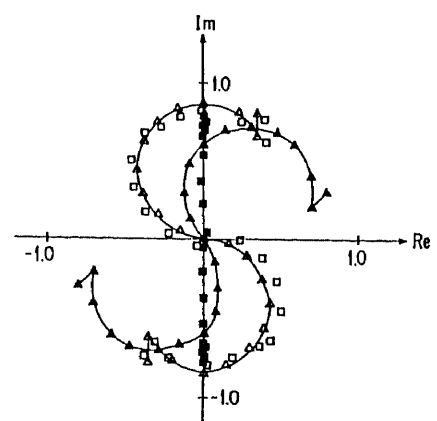
(a) 学習パターン E



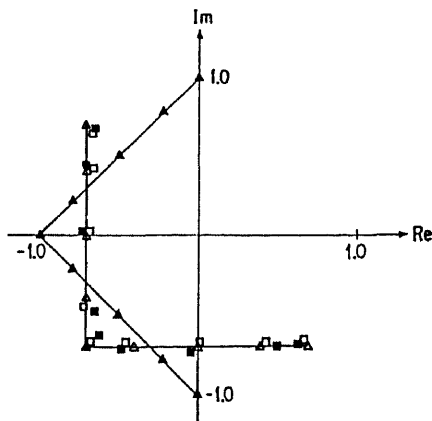
(a) 学習パターン I



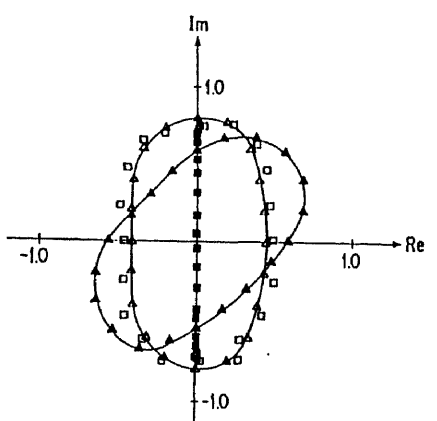
(b) テストパターン T



(b) テストパターン S



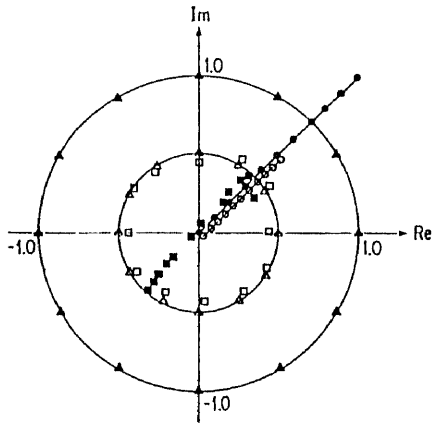
(c) テストパターン L



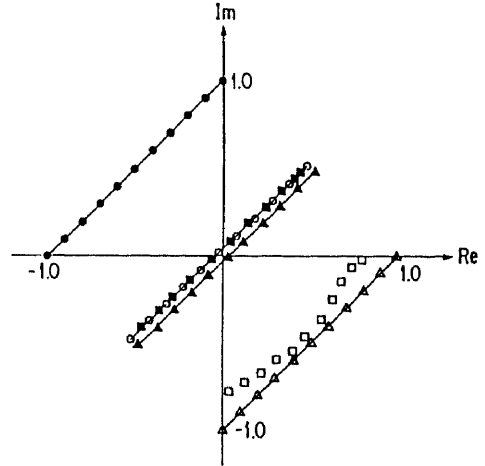
(c) テストパターン O

図 5.2 単語 E T L の回転

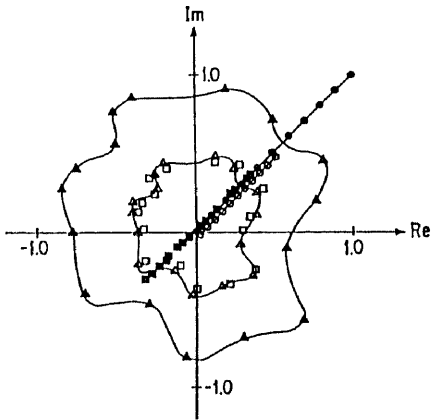
図 5.3 単語 I S O の回転



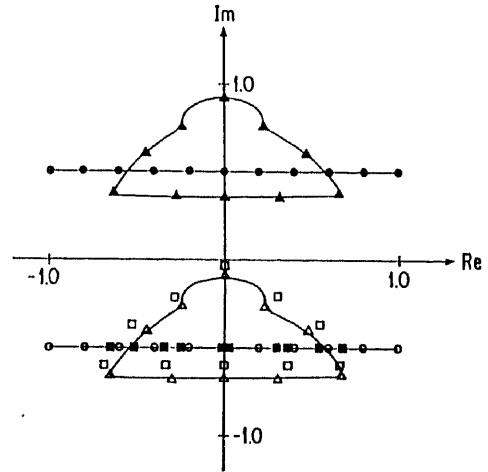
(a) 円の縮小



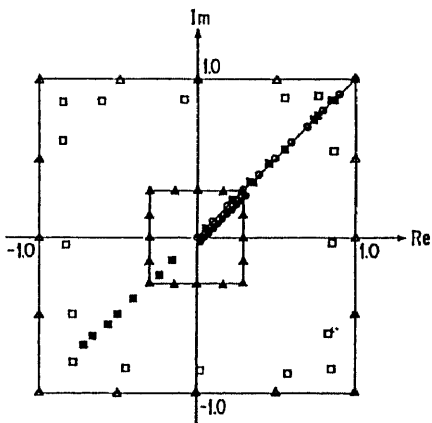
(a) 直線



(b) 曲線の縮小



(b) 曲線



(c) 正方形の拡大

図 5.5 平行移動

図 5.4 相似変換

5.3.4 誤差の評価

5.3.1項–5.3.3項において示した2次元アフィン変換学習能力は、回転角度、相似率(縮小率あるいは拡大率)、平行移動ベクトル(平行移動の方向と距離を表すベクトル)に関する汎化能力と言い換えることもできる。本項では、その汎化能力の誤差について体系的に調べる。

回転、相似変換、平行移動に対して、それぞれ、次のような学習パターンを用いて実験を行った。

1. 回転

入力パターンは図5.6(a)における●印で示された点列、出力パターンは○印で示された点列とした。つまり、出力パターンは入力パターンを原点を中心に正の方向に90度回転させることによって得られるものである。

2. 相似変換

入力パターンは図5.6(b)における●印で示された点列、出力パターンは○印で示された点列とした。つまり、出力パターンは入力パターンを原点との距離に関して2分の1に縮小させることによって得られるものである。

3. 平行移動

入力パターンは図5.6(c)における●印で示された点列、出力パターンは○印で示された点列とした。つまり、出力パターンは入力パターンを右斜め下45度の方向に $1/\sqrt{2}$ の距離だけ平行移動させることによって得られるものである。

上記の3つの場合に対して、それぞれ、学習終了後、テストパターンとして、図5.6(a)–(c)において▲印で示される8つの点を与えた。このテストパターンに対して複素BPネットワークが実際に出力した点と期待された出力点との間にはいくらかのずれ(誤差)が発生した。この誤差を評価するための1つの基準として、複素BPネットワークが実際に出力した点と期待された出力点との間のユークリッド距離を用いた。回転、相似変換、平行移動に対するそのような評価基準(ユークリッド距離)を、それぞれ、 $|E^R(\phi)|$ 、 $|E^S(\phi)|$ 、 $|E^P(\phi)|$ と記し、それぞれ、汎化回転誤差、汎化相似誤差、汎化平行移動誤差と呼ぶことにする。ここで、 $E^R(\phi)$ 、 $E^S(\phi)$ および $E^P(\phi)$ は、それぞれ、誤差を表す複素数である(この詳細については、6章で述べる)。 ϕ はテストパターン(複素数)の偏角と入力学習パターン(複素数)の偏角との差を表す。

実験結果を図5.7(a)–(c)に示す。この実験結果からは、次のような定性的性質が読みとれる。すなわち、汎化能力の誤差はテストパターンが入力学習パターンから離れるにつれて(つまり、 ϕ が大きくなるにつれて)増加する。そして、最も距離が大きくなるあたり($\phi \approx 180$)で誤差が最大となる。さらに、テストパターンが入力学習パターンに近づくにつれて減少する。

上記の実験において、学習パターンはそれぞれ次のとおりであった。

1. 回転

11 個の入力パターンは、直線 $x = 0$ ($0 \leq y \leq 1$) 上に、11 個の出力パターンは $y = 0$ ($-1 \leq x \leq 0$) 上にそれぞれ乗っていた (図 5.6(a))。

2. 相似変換

11 個の入力パターンは、直線 $x = 0$ ($0 \leq y \leq 1$) 上に、11 個の出力パターンは直線 $x = 0$ ($0 \leq y \leq 0.5$) 上にそれぞれ乗っていた (図 5.6(b))。

3. 平行移動

11 個の入力パターンは、直線 $x = 0$ ($0 \leq y \leq 1$) 上に、11 個の出力パターンは直線 $x = 0.5$ ($-0.5 \leq y \leq 0.5$) 上にそれぞれ乗っていた (図 5.6(c))。

また、8 個のテストパターンは円 $x^2 + y^2 = 0.5^2$ の円周上に乗っていた (図 5.6(a)-(c))。

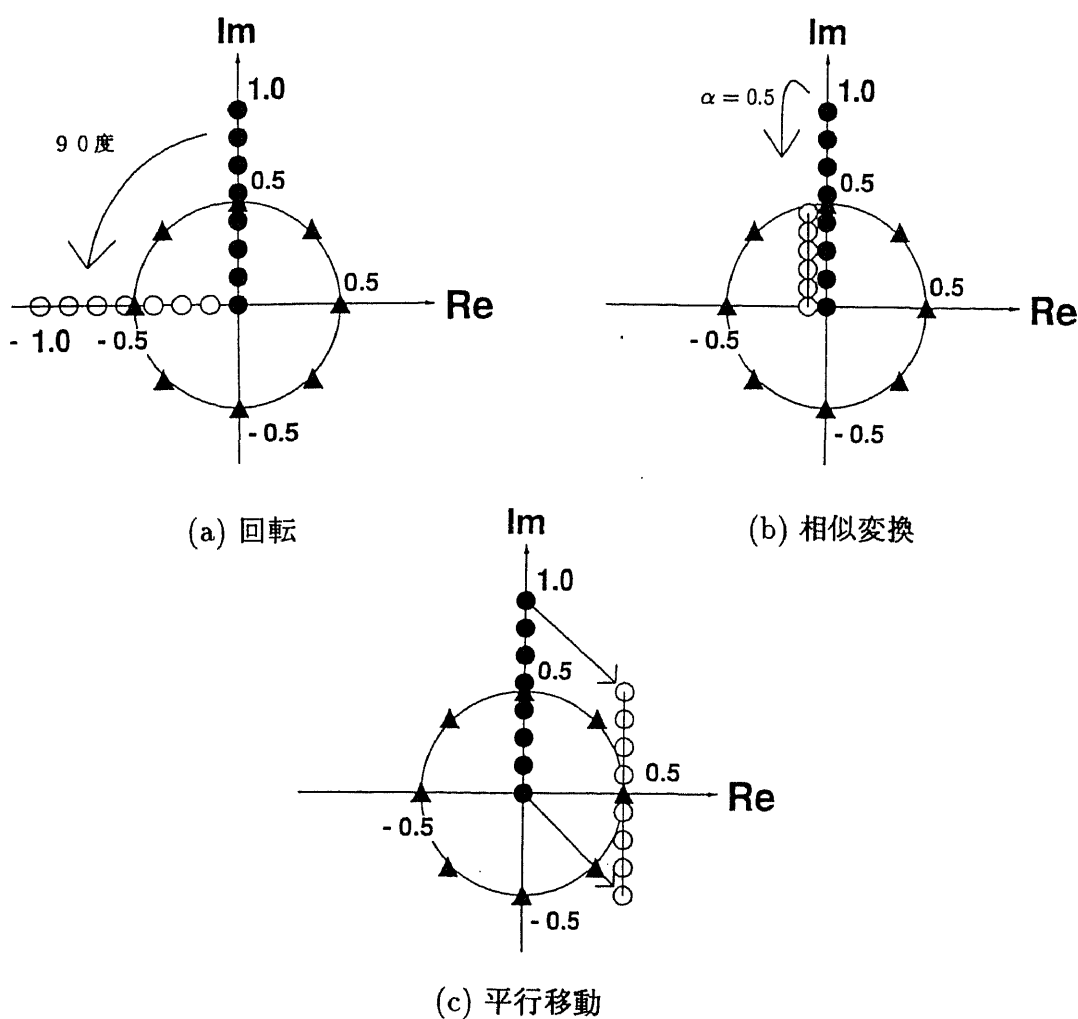
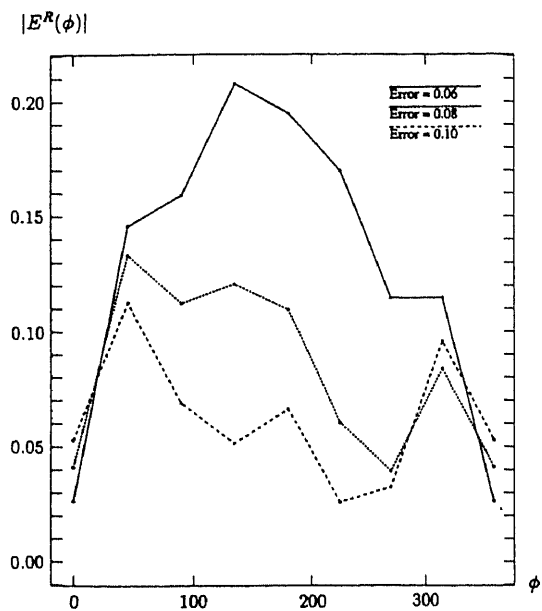
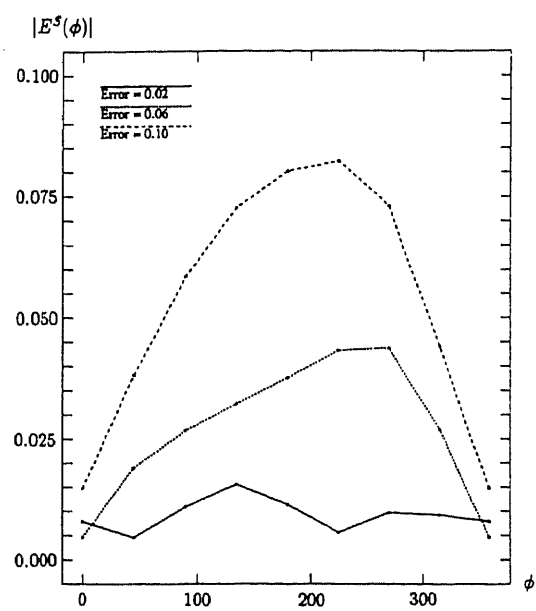


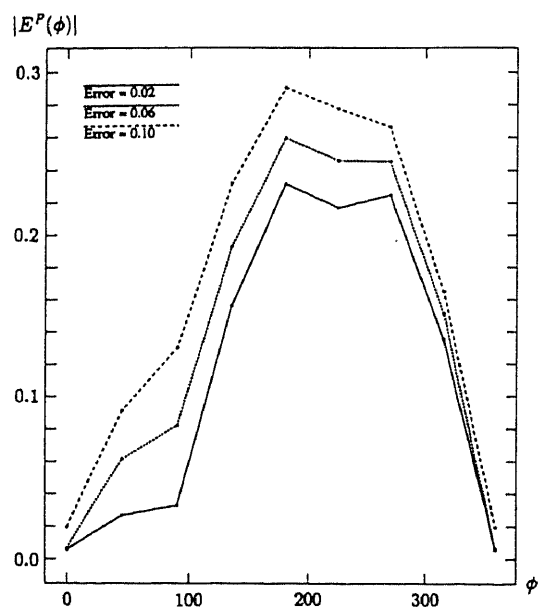
図 5.6 汎化能力の誤差の体系的調査に用いる学習パターンとテストパターン



(a) 回転



(b) 相似変換



(c) 平行移動

図 5.7 汎化能力の誤差の体系的調査の実験結果

5.3.5 学習パラメータの値

本項では2次元アフィン変換学習能力と学習パラメータの値との関係を実験的に調べる。例として回転の学習を取り上げる。

実験において、学習させる回転角度は30度、60度あるいは90度の3種類である(図5.8(a)-(c))。それぞれの学習パターンは11個の点列から成り、出力パターンは入力パターンを原点を中心にして正の方向に30度、60度あるいは90度回転させることによって得られる。入力パターンは直線 $x = 0$ ($0 \leq x \leq 1$) 上に0.1間隔に並んだ11個の点列である。本項では、学習は誤差

$$\sqrt{2 \sum_p E_p} \quad (5.1)$$

が0.10となったときに収束(終了)したもののみなし、収束の規範とする(E_p は2.4節で定義したパターン p に対する2乗誤差である)。

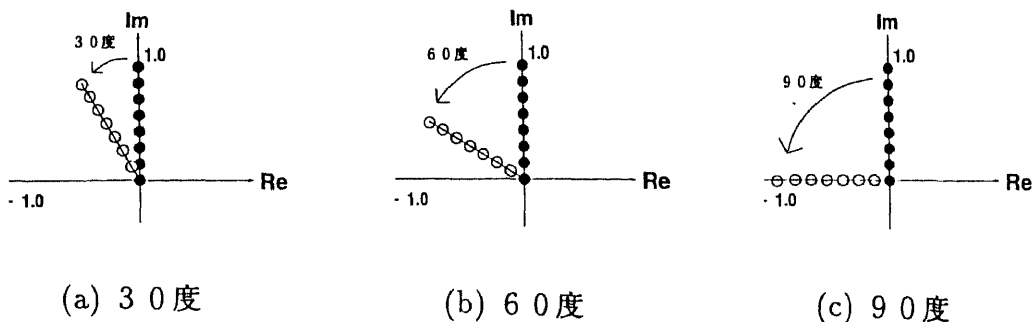


図5.8 学習させる回転角度

学習終了後の学習パラメータの値を調べた。学習パラメータの初期値を設定するための乱数を変えて100回の試行を行った。個々の学習パラメータの値は一見して学習した回転角度とは何らの関係もないように思われるような値であった(表5.1)。そこで、入力-中間層の重みの合計値の平均、中間-出力層の重みの合計値の平均を評価基準としてみた。すなわち、入力-中間層の重みは全部で6個あるが、それらを加えることによって得られる複素数 W (合成重みと呼ぶ) の偏角を100回の試行に関して平均をとるわけである。中間-出力層の合成重み V についても同じである。表5.2がその結果である。絶対値に関しては特に因果関係が見られなかったため、表5.2には偏角のみ示してある。

表5.2からわかるように、学習した回転角度とネットワークの重みの偏角とは明らかに因果関係がある。すなわち、(17度くらいの)誤差はあるものの合成重みの偏角のそれぞれの平均値の合計値 $\overline{W} + \overline{V}$ は学習した回転角度に近い値となっている。

表5.1 学習パラメータの値の一例. 60度を学習したネットワークの入力-中間層の重みの値. w_{j1} は入力ニューロン1と中間ニューロン j との間の重みを表す. 偏角の単位は度.

| 重み | w_{11} | w_{21} | w_{31} | w_{41} | w_{51} | w_{61} |
|-----|----------|----------|----------|----------|----------|----------|
| 絶対値 | 0.6 | 0.4 | 1.6 | 2.8 | 0.9 | 0.6 |
| 偏角 | 14 | 117 | 76 | 36 | 22 | 107 |

表5.2 合成重みの偏角の平均値. 合成重みとは, 入力-中間層 (あるいは中間-出力層) のすべての重みを加えることによって得られる複素数のことをいう. 合成重みは当該層における重みが全体としてどんな値を表しているかを示すものといえる. 回転角度および偏角の単位は度.

| 学習した回転角度 | 30 | 60 | 90 |
|-----------------------------------|----|----|-----|
| 入力-中間層の合成重み W の偏角の平均値 \bar{W} | 25 | 29 | 33 |
| 中間-出力層の合成重み V の偏角の平均値 \bar{V} | 23 | 48 | 75 |
| 合計 $\bar{W} + \bar{V}$ | 48 | 77 | 108 |

5.4 やや複雑な変換

5.3節においては, 単独の変換率 (回転角度, 相似率, 平行移動ベクトル) を用いた2次元アフィン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.

5.4.1 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 (図5.9).

入力学習パターンは, 直線 $x = 0$ ($-1 \leq y \leq 1$) 上の21個の点 (●印) とした. 直線 $x = 0$ ($0 \leq y \leq 1$) 上の11個の点に対応する出力学習パターンは, 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) を出力学習パターンとした (○印). ここで, 直線 $x = 0$ ($0 \leq y \leq 1$) 上の点に対応する学習パターンを学習パターン1, 直線 $x = 0$ ($-1 \leq y \leq 0$) 上の点に対応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 (▲印)。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 (□印)。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 $67.5 = (90 + 45) / 2$ 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。

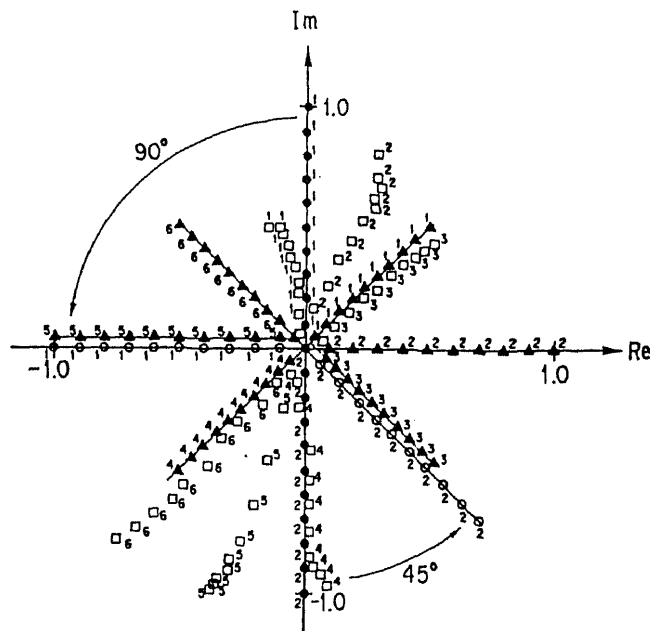


図 5.9 回転

5.4.2 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5.10)。

入学習パターンは直線 $y = x$ ($-1 \leq x \leq 1$) 上の21個の点(●印)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)を出力学習パターンとし、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)を出力学習パターンとした(○印)。ここで、直線 $y = x$ ($0 \leq x \leq 1$) 上の点に対応する学習パターンを学習パターン1、直線 $y = x$ ($-1 \leq x \leq 0$) 上の点に対応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 $x^2 + y^2 = 1$ の円周上の24個の点(▲印)を与えたところ、ニューラルネットワークは図5.10における□印で示される点(出力テストパターン)を出力した。図5.10において、出力テストパターン(□印)は、入学習パターン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る。

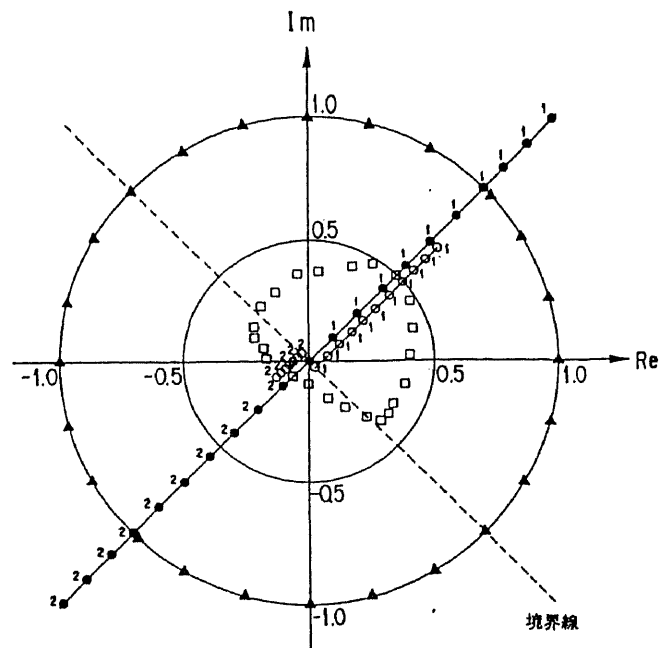


図5.10 相似変換

5.4.3 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5.11)。

入学習パターンは、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(●印)とした。直線 $y = 1$ ($-1 \leq x \leq 1$) 上の21個の点に対する出力学習パターンは、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)を出力学習パターンとした(○印)。ここで、直線 $y = 1$ ($-1 \leq x \leq 1$) 上の点に対応する学習パターンを学習パターン1、直線 $y = 0.2$ ($-1 \leq x \leq 1$) 上の点に対応する学習パターンを学習パターン2と呼ぶことにする。

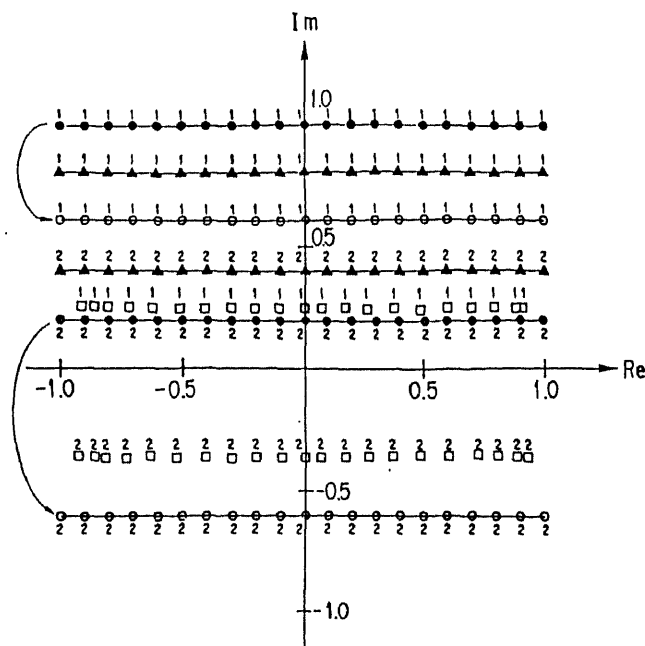


図5.11 平行移動

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(▲印)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(□印)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、入学習パターン1とは0.2、入学習パターン2とは0.6の距離がある。また、入力テストパターン2は、入学習パターン1とは0.6、入学習パターン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、入学習パターン1と入学習パターン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。

5.5 考察

複素BP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2次元アフィン変換学習能力があることがわかった。

実験結果において、正方形が完全に拡大されなかったり(図5.4(c)), 平行移動により直線がいくらかねじ曲がっているが(図5.5(a)), 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、5.3.1項の回転の実験では、点をあらかず複素数 $z_1 = x + iy$ 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5.4(a)に示す縮小は $\alpha = 0.5$ の場合、図5.4(c)の拡大は $\alpha = 3.33$ の場合であり、図5.5(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 $g(z) = ze^{i\theta}$, $g(z) = \alpha z$ あるいは $g(z) = z + w$ 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 g 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(たとえば、[12])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、

定理 3 (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。

平たく言うと、一致の定理は次のようなことを主張している。平野に、高さが複素数値で表されている2つの山 A, B があって、それらの形状が同一であるか否かを知りたいとする。ただし、山 A と B の底面は重なり合わせることができるとする。山 A と B が、それぞれ、正則な複素数値関数 f, g (定義域は同じ)で表現できるとき、私達は山の底面内のある曲線上の山の高さが等しいか否かをチェックしさえすれば、山 A と B の形状は全く同じであるか否かを判定することができる。一方、山 A と B が正則な複素数値関数で表現できな

い場合、一致の定理は適用できないので、私達は山の底面内のすべての点における山の高さが等しいか否かを逐一チェックする必要がある。

そこで、この一致の定理を使って前述のニューラルネットワークのふるまいを解釈すると次のようになる。学習データは、複素関数 $G: [-1, 1] \times [-1, 1] \rightarrow [-1, 1] \times [-1, 1]$ 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 G を推定しようとし、その結果、推定関数 F 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ように定義域 $[-1, 1] \times [-1, 1]$ のすべての点 z に対して真の複素関数 $G(z)$ 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 F 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。

5.6 コンピュータビジョンへの応用

本節では、複素BPのコンピュータビジョンへの応用について述べる。本章で今まで述べてきた2次元アフィン変換学習能力がうまく生かされる。

コンピュータビジョン [6, 11, 54] は、目で外界を見てその状況を認識、理解する人間の視覚情報処理機能を、機械を用いて実現させることを目的とした研究分野である。具体的には、カメラを通じて得られた画像から、外界の物理的対象物に対する明白で意味のある記述を計算機を通して構成することを目的としている。

コンピュータビジョンにおいて、動いている物体の動画像（異なる時刻で得られた画像の集まり）から物体の動きパラメータを推定することは重要な課題である [2]。多くの方法は、動画像からオプティカル・フローと呼ばれる画像上の2次元速度ベクトル場を計算することを前処理として行ない、次にそのオプティカル・フローから画像中の動物体の動きパラメータを推定する。オプティカル・フローから動きパラメータを推定する方法の多くは、オプティカル・フロー中のいくつかのフロー・ベクトルを使って方程式を解いている [1, 55, 57]。しかし、この方法では時間がかかること、雑音に弱いこと、実画像に対する解が容易に得られない、といった解決すべき課題がある。

ニューラルネットワークはパターン変換にしばしば利用され、雑音にそれほど影響を受けないという特徴がある。しかも、学習後の処理に要する時間は短い。よって、これらの点で、動きパラメータの推定に応用すると有効であると考えられる。

特に、複素BPネットワークは入出力パターンが複素パターン（2次元情報）であるので、2次元ベクトルから構成されるオプティカル・フローを自然に扱うことができる。つまり、オプティカル・フロー中の2次元ベクトルを、1つの複素数として、複素ニューロンに素直に入出力させることができる。図5.12に動きの解釈を行なう複素BPネットワークの構成例を示す。入力はおプティカル・フロー、出力は動きパラメータである。ここで、複素BPネットワークには、1つの平行移動や1つの回転といった基本的な動きに関するパターンを教師データとして予め学習させておく。基本的な動きを学習した複素BPネットワークは未学習の動きに対してもうまく解釈を行なうことが期待される。

さらに、実画像のオプティカル・フローは必ずしも完全なものを得られるわけではなく、1部分しか得られないことが多い(図5.13)。このため、正規化を行なう必要があり、この正規化に複素BPの2次元アフィン変換学習能力を利用することが考えられる。つまり、得られたオプティカル・フローの1部分を $1-n-1$ 複素BPネットワークに学習させ、オプティカル・フローの残りの部分を推定させるといった使い方をするわけである。図5.14にオプティカル・フローの正規化を行なう複素BPネットワークの構成例を示す。フロー・ベクトルの始点を表す2次元座標を入力ニューロンに、終点を表す2次元座標を出力ニューロンに対応させて利用する。

実際、宮内らはこの考え方に沿って複素BPネットワークに(実画像も含めた)動きの解釈を行なわせ、その有効性を確認している[17, 18, 19]。

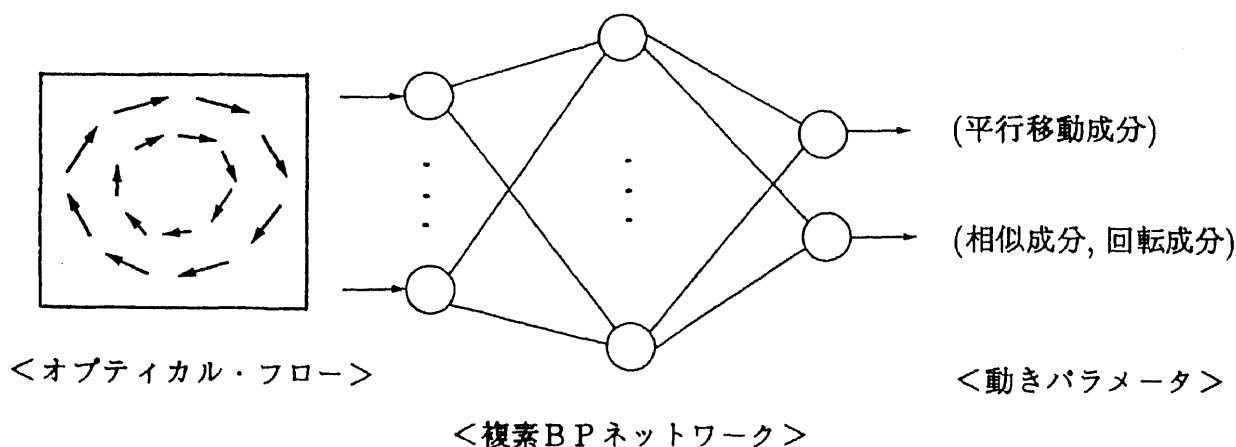


図5.12 動きの解釈を行なう複素BPネットワークの構成例

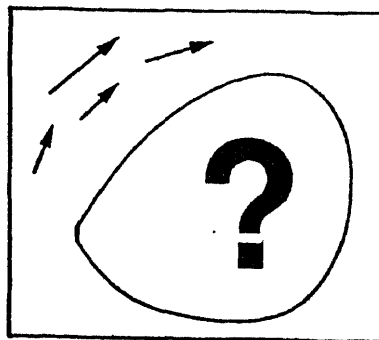
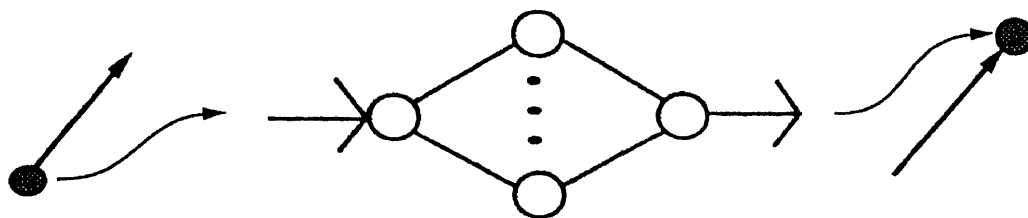


図5.13 1部分だけが得られているオプティカル・フローのイメージ



<フロー・ベクトルの始点> <複素BPネットワーク> <フロー・ベクトルの終点>

図5.14 オプティカル・フローの正規化を行なう複素BPネットワークの構成例

5.7 むすび

複素BPを用いて2次元アフィン変換(図形変換)を学習させたニューラルネットワークのふるまいを計算機実験を通じて評価した。そのふるまいは、実BPには見られない複素BP固有のものであり、図形の変換を行なうものである。そしてそれは一致の定理と関係している。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。また、2次元アフィン変換学習能力の誤差の性質が明らかになった。つまり、2次元アフィン変換学習能力の誤差はテストパターンが入力学習パターンから離れるにつれて増加する。そして、最も距離が大きくなるあたりで誤差が最大となる。さらに、テストパターンが入力学習パターンに近づくと減少する。また、「学習させるアフィン変換」と「学習パラメータの値」との明確な関係が示唆された。回転の学習の場合、合成重みの合計値は学習させた回転角度とほぼ等しくなっている。最後に、コンピュータビジョンへの応用について述べた。第6章では、この2次元アフィン変換学習能力について数学的に解析する。

第 6 章

複素バックプロパゲーション学習アルゴリズムの 2 次元アフィン変換学習能力の数学的解析

6.1 はじめに

本章では、第 5 章において実験的に示した 2 次元アフィン変換学習能力の数学的解析を行う ([25, 28, 29, 33, 46])。得られた主要な結果は次のとおりである。

1. 1 種類の変換率（たとえば、1 種類の回転角度）を学習した複素 BP ネットワークの定性的性質を明らかにした。変換率に関する汎化能力の（距離に関する）誤差は、未学習パターンと学習パターンの偏角の差に関する正弦関数によって表現される。
2. 2 種類の変換率を学習した複素 BP ネットワークのふるまいを回転角度を例にとって解析した結果、次の結果を得た。
 - (a) 2 種類の回転角度を学習した複素 BP ネットワークの回転角度に関する汎化能力の（距離に関する）誤差の評価式を与えた。
 - (b) 2 種類の回転角度を学習した複素 BP ネットワークの回転角度に関する汎化能力の『（角度に関する）誤差が無視できるほど小さな値であるための十分条件』を求めた。

以下、6.2 節において、解析の対象とする複素 BP ネットワークモデルを定義し、6.3 節では 1 種類の変換率を学習した複素 BP ネットワークの変換率に関する汎化能力の解析を行い、その定性的な性質を明らかにする。6.4 節では、2 種類の変換率を学習した場合を扱う。6.5 節において、それらの計算機実験を通じた数値例を示し、6.6 節においては、簡単な考察を行なう。最後に、6.7 節において結論を述べる。

6.2 複素 BP ネットワークモデル

まず、解析の対象とするモデルの説明を行なう。解析を容易にするために、入力ニューロン 1 個、中間ニューロン 1 個、出力ニューロン 1 個から成る 3 層ネットワークを考え、入力

—中間ニューロン間および中間—出力ニューロン間の重みパラメータをそれぞれ $v \exp[iw]$, $c \exp[id]$ とし, 中間ニューロンおよび出力ニューロンの閾値をそれぞれ $s \exp[it]$, $r \exp[il]$ とする。この複素BPネットワークにおいて, 複素BP学習を施した後の学習パラメータを $v^0 \exp[iw^0]$, $c^0 \exp[id^0]$, $s^0 \exp[it^0]$, $r^0 \exp[il^0]$ で表すことにする (これらの記法は, 本章で共通に使われるが, 異なる学習パターンで学習が行なわれる場合, それらの値は異なるものであることに注意)。

また, 次の定数を定義しておく。

$$\begin{aligned}
K &= \frac{1}{(1+\sqrt{2})c^0+2r^0}, \quad G = \frac{kac^0v^0}{2(kav^0+s^0)}, \quad A = \frac{c^0s^0}{2(kav^0+s^0)}, \quad B = \frac{c^0}{\sqrt{2}}, \quad C = r^0, \\
H_R &= A \cos(t^0+d^0) + B \cos\left(d^0 + \frac{\pi}{4}\right) + C \cos(l^0), \\
H_I &= A \sin(t^0+d^0) + B \sin\left(d^0 + \frac{\pi}{4}\right) + C \sin(l^0), \\
M &= 2K\sqrt{H_R^2 + H_I^2}, \\
M(\gamma) &= 2\sqrt{K^2\left[A^2+B^2+C^2+2AB \cos\left(t^0-\frac{\pi}{4}\right)+2AC \cos(t^0+d^0-l^0)+2BC \cos\left(d^0+\frac{\pi}{4}-l^0\right)\right]} \\
&\quad + \frac{\tau^2}{4} - \tau K \left[A \cos(t^0+d^0-\omega) + B \cos\left(d^0 + \frac{\pi}{4} - \omega\right) + C \cos(l^0 - \omega) \right], \\
Q &= -4Kka \sin^2\left(\frac{x-y}{4} + \frac{\alpha-\beta}{2}\right) \cdot \left[H_R \cos\left(\frac{x+y}{2} + \pi + \frac{\alpha+\beta}{2}\right) + H_I \sin\left(\frac{x+y}{2} + \pi + \frac{\alpha+\beta}{2}\right) \right] \\
&\quad + 8K^2G \sin^2\left(\frac{x-y}{4}\right) \cdot \left[H_R \cos\left(\frac{x+y}{2} + \pi + w^0 + d^0\right) + H_I \sin\left(\frac{x+y}{2} + \pi + w^0 + d^0\right) \right] \\
&\quad + M^2. \tag{6.1}
\end{aligned}$$

これらは学習終了後の学習パラメータの値に依存しており, 学習毎に値は異なっていることに注意されたい。

6.3 1つの変換率の学習能力の解析

本節では複素BPの唯一つの変換率の学習能力を解析的に調べ, その定性的な性質を明らかにする。

6.3.1 回転の学習

まず, 回転角度を複素BPにより学習させた場合のニューラルネットワークのふるまいについて調べる。

学習パターンは, 複素平面において実軸と x 度の角度をなす直線上に等間隔に並んだ p 個の点列を原点を中心にして正の方向 (反時計回り) に α 度だけ回転させるものとする (図 6.1)。つまり, 任意の $1 \leq k \leq p$ に対して, 入力パターンは

$$ka \exp[ix] \tag{6.2}$$

であり、それに対応する出力パターンは

$$ka \exp[i(x + \alpha)] \quad (6.3)$$

である。ただし、 $a \in \mathbf{R}^+$ は常数であり、点列の間隔を表す。また、 $k, p \in \mathbf{N}$ (\mathbf{N} は自然数全体の集合を表す); $x, \alpha \in \mathbf{R}$; $0 < pa \leq 1$ である。ここで、式(6.3)で示される出力パターン(-1から1の間の値をとる)は、ニューラルネットワークに与える際には、0から1の間の値へ変換しておく必要がある。なぜならば、複素BPネットワークの出力値が0から1の間の値であるからである。その変換の様子を以下に示す。任意の $1 \leq k \leq p$ に対して、式(6.3)は

$$ka \exp[i(x + \alpha)] = ka \cos(x + \alpha) + ika \sin(x + \alpha) \quad (6.4)$$

と表される。この実部と虚部は共に-1から1の間の値をとる。そこで、それらをそれぞれ2分の1にした後、2分の1を加えることにより、0から1の間の値をとるように変換する:

$$\begin{aligned} & \left[\frac{1}{2}ka \cos(x + \alpha) + \frac{1}{2} \right] + i \left[\frac{1}{2}ka \sin(x + \alpha) + \frac{1}{2} \right] \\ &= \frac{1}{2}ka \left[\cos(x + \alpha) + i \sin(x + \alpha) \right] + \frac{1}{2}[1 + i] \\ &= \frac{1}{2}ka \exp[i(x + \alpha)] + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right]. \end{aligned} \quad (6.5)$$

このように、任意の $1 \leq k \leq p$ に対して、ニューラルネットワークに与えられる出力パターンは

$$\frac{1}{2}ka \exp[i(x + \alpha)] + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right] \quad (6.6)$$

とやや複雑な形で表される。

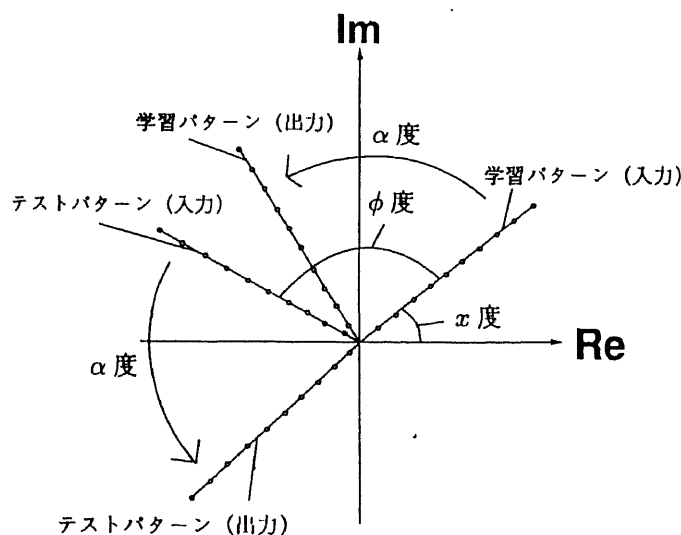


図6.1 学習パターンとテストパターン (α 度の回転)

以下、この学習パターンを学習した複素BPネットワークのふるまいの定性的性質を明

らかにする。準備として、補題を用意しておく。

補題 1 任意の $1 \leq k \leq p$ に対して、近似的に次の等式が成り立つ。

$$K \left[G \cos(x + w^0 + d^0) + H_R \right] + \frac{1}{2} = \frac{1}{2} ka \cos(x + \alpha) + \frac{1}{2}, \quad (6.7)$$

$$K \left[G \sin(x + w^0 + d^0) + H_I \right] + \frac{1}{2} = \frac{1}{2} ka \sin(x + \alpha) + \frac{1}{2}. \quad (6.8)$$

(証明)

任意の $1 \leq k \leq p$ に対して、入力学習パターン $ka \exp[ix]$ に対する複素 BP ネットワークの出力値を求めると、その実部は式 (6.7) の左辺、虚部は式 (6.8) の左辺であることがわかる。その際、各ニューロンが持つ出力関数に現れるシグモイド関数は区分的線形関数で近似しておく。つまり、中間ニューロンにおいては区分的線形関数

$$g(x) = \begin{cases} \frac{x}{2(kav^0 + s^0)} + \frac{1}{2} & \left(-(kav^0 + s^0) \leq x \leq kav^0 + s^0 \right) \\ 1 & \left(kav^0 + s^0 < x \right) \\ 0 & \left(x < -(kav^0 + s^0) \right) \end{cases} \quad (6.9)$$

で、出力ニューロンにおいては区分的線形関数

$$h(x) = \begin{cases} \frac{1}{(1+\sqrt{2})c^0 + 2r^0} x + \frac{1}{2} & \left(-\left(\frac{1+\sqrt{2}}{2}c^0 + r^0\right) \leq x \leq \frac{1+\sqrt{2}}{2}c^0 + r^0 \right) \\ 1 & \left(\frac{1+\sqrt{2}}{2}c^0 + r^0 < x \right) \\ 0 & \left(x < -\left(\frac{1+\sqrt{2}}{2}c^0 + r^0\right) \right) \end{cases} \quad (6.10)$$

で近似しておく。一方、入力学習パターンに対する複素 BP ネットワークの出力値の実部と虚部は、それぞれ、入力学習パターン $(1/2)ka \exp[i(x + \alpha)] + (1/\sqrt{2}) \exp[i(\pi/4)]$ の実部、虚部に等しいから、式 (6.7) および式 (6.8) が得られる。 (証明終)

次に示す定理は、複素 BP の回転角度に関する汎化能力の定性的性質を明らかにするものである。

定理 4 $1 \leq k \leq p$ を任意に固定する。式 (6.2) および式 (6.6) で示される学習パターンを学習した複素 BP ネットワークに対して、『入力学習パターン $ka \exp[ix]$ を原点を中心にして正の方向に任意の角度 ϕ だけ回転させた点 $ka \exp[i(x + \phi)]$ 』を入力テストパターンとして与える (図 6.1)。このとき、複素 BP ネットワークが出力する値は

$$\left[\frac{1}{2} ka \exp \left[i(x + \phi + \alpha) \right] + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right] \right] + E^R(\phi) \in \mathbb{C} \quad (6.11)$$

で与えられる。ここで、式(6.11)の第1項は未学習パターン $ka \exp[i(x + \phi)]$ を原点を中心にして正の方向に『学習した角度 α 』だけ回転させた点に相当する(図6.1)。また、 $E^R(\phi)$ はそのときに生じるずれを表す複素数であり、その絶対値(回転汎化誤差)は、

$$|E^R(\phi)| = M \left| \sin \left(\frac{\phi}{2} \right) \right| \quad (6.12)$$

で与えられる。

(注意)

式(6.12)における M は式(6.1)で定義された常数であり、学習終了後の学習パラメータの値には依存するが、 ϕ には依存しない。

(証明)

次の方針で証明する：補題1で示した式(6.7)および式(6.8)をもとに、未学習パターン $ka \exp[i(x + \phi)]$ に対する複素BPネットワークの出力値を計算し、その出力値を[未学習パターンを原点を中心にして正の方向に α 度回転させた点] + [ずれ] という形に変形する。

まず、入力テストパターン $ka \exp[i(x + \phi)]$ に対する複素BPネットワークの出力値の実部を求める。 $\lambda = \tan \phi$ とおくと、任意の θ に対して、

$$\cos \theta - \lambda \sin \theta = \sqrt{1 + \lambda^2} \cos(\theta + \phi) \quad (6.13)$$

であることを利用して、[式(6.7)] - λ [式(6.8)]を計算することにより、

$$K \left[G \cos(x + \phi + w^0 + d^0) + H_R \right] + \frac{1}{2} = \left[\frac{1}{2} ka \cos(x + \phi + \alpha) + \frac{1}{2} \right] + E_r^R(\phi), \quad (6.14)$$

ただし、

$$E_r^R(\phi) = 2K \sin \left(\frac{\phi}{2} \right) \cdot \left[A \sin \left(t^0 + d^0 + \frac{\phi}{2} \right) + B \sin \left(d^0 + \frac{\pi}{4} + \frac{\phi}{2} \right) + C \sin \left(l^0 + \frac{\phi}{2} \right) \right] \quad (6.15)$$

が得られる。ここで、式(6.14)の左辺は、入力テストパターンに対する複素BPネットワークの出力値の実部を表している。また、その右辺の第1項は、入力テストパターンを原点を中心にして正の方向に、『学習した角度 α 』だけ回転させた点 $ka \exp[i(x + \phi + \alpha)]$ の実部を表している。また、 $E_r^R(\phi)$ は“ずれ”を表す複素数 $E^R(\phi)$ の実部に相当する。

同様に、任意の θ に対して、

$$\lambda \cos \theta + \sin \theta = \sqrt{1 + \lambda^2} \sin(\theta + \phi) \quad (6.16)$$

であることを利用して、 λ [式(6.7)] + [式(6.8)]を計算することにより、

$$K \left[G \sin(x + \phi + w^0 + d^0) + H_I \right] + \frac{1}{2} = \left[\frac{1}{2} ka \sin(x + \phi + \alpha) + \frac{1}{2} \right] + E_i^R(\phi), \quad (6.17)$$

ただし,

$$E_i^R(\phi) = -2K \sin\left(\frac{\phi}{2}\right) \cdot \left[A \cos\left(t^0 + d^0 + \frac{\phi}{2}\right) + B \cos\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + C \cos\left(l^0 + \frac{\phi}{2}\right) \right] \quad (6.18)$$

が得られる。ここで、式(6.17)の左辺は、入力テストパターンに対する複素BPネットワークの出力値の虚部を表している。また、その右辺の第1項は、入力テストパターンを原点を中心にして正の方向に【学習した角度 α 】だけ回転させた点 $ka \exp[i(x + \phi + \alpha)]$ の虚部を表している。また、 $E_i^R(\phi)$ は“ずれ”を表す複素数 $E^R(\phi)$ の虚部に相当する。

よって、式(6.14)および式(6.17)により、入力テストパターンに対する複素BPネットワークの出力値は、

$$\left[\frac{1}{2} ka \exp[i(x + \phi + \alpha)] + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \right] + E^R(\phi), \quad (6.19)$$

ただし,

$$E^R(\phi) \stackrel{\text{def}}{=} E_r^R(\phi) + iE_i^R(\phi) \quad (6.20)$$

で表現される。すなわち、複素BPネットワークは、入力テストパターン $ka \exp[i(x + \phi)]$ を原点を中心にして正の方向に【学習した角度 α 】だけ回転させた点 $ka \exp[i(x + \phi + \alpha)]$ を出力し、そのときに生じる“ずれ”は $E^R(\phi)$ で与えられる。また、式(6.12)は、式(6.15)および式(6.18)から次のようにして得られる：

$$\begin{aligned} E_r^R(\phi)^2 &= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A \sin\left(t^0 + d^0 + \frac{\phi}{2}\right) + B \sin\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + C \sin\left(l^0 + \frac{\phi}{2}\right) \right]^2 \\ &= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A^2 \sin^2\left(t^0 + d^0 + \frac{\phi}{2}\right) + B^2 \sin^2\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + C^2 \sin^2\left(l^0 + \frac{\phi}{2}\right) \right. \\ &\quad + 2AB \sin\left(t^0 + d^0 + \frac{\phi}{2}\right) \cdot \sin\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + 2AC \sin\left(t^0 + d^0 + \frac{\phi}{2}\right) \cdot \sin\left(l^0 + \frac{\phi}{2}\right) \\ &\quad \left. + 2BC \sin\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) \cdot \sin\left(l^0 + \frac{\phi}{2}\right) \right], \quad (6.21) \end{aligned}$$

$$\begin{aligned} E_i^R(\phi)^2 &= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A \cos\left(t^0 + d^0 + \frac{\phi}{2}\right) + B \cos\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + C \cos\left(l^0 + \frac{\phi}{2}\right) \right]^2 \\ &= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A^2 \cos^2\left(t^0 + d^0 + \frac{\phi}{2}\right) + B^2 \cos^2\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + C^2 \cos^2\left(l^0 + \frac{\phi}{2}\right) \right. \\ &\quad + 2AB \cos\left(t^0 + d^0 + \frac{\phi}{2}\right) \cdot \cos\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) + 2AC \cos\left(t^0 + d^0 + \frac{\phi}{2}\right) \cdot \cos\left(l^0 + \frac{\phi}{2}\right) \\ &\quad \left. + 2BC \cos\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right) \cdot \cos\left(l^0 + \frac{\phi}{2}\right) \right]. \quad (6.22) \end{aligned}$$

よって,

$$\begin{aligned}
E_r^R(\phi)^2 + E_i^R(\phi)^2 &= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A^2 + B^2 + C^2 + 2AB \cos\left(t^0 + d^0 + \frac{\phi}{2} - \left(d^0 + \frac{\pi}{4} + \frac{\phi}{2}\right)\right) \right. \\
&\quad \left. + 2AC \cos\left(t^0 + d^0 + \frac{\phi}{2} - \left(l^0 + \frac{\phi}{2}\right)\right) + 2BC \cos\left(d^0 + \frac{\pi}{4} + \frac{\phi}{2} - \left(l^0 + \frac{\phi}{2}\right)\right) \right] \\
&= 4K^2 \sin^2\left(\frac{\phi}{2}\right) \cdot \left[A^2 + B^2 + C^2 + 2AB \cos\left(t^0 - \frac{\pi}{4}\right) + 2AC \cos\left(t^0 + d^0 - l^0\right) \right. \\
&\quad \left. + 2BC \cos\left(d^0 + \frac{\pi}{4} - l^0\right) \right] \\
&= 4K^2 (H_R^2 + H_l^2) \cdot \sin^2\left(\frac{\phi}{2}\right) \\
&= M^2 \cdot \sin^2\left(\frac{\phi}{2}\right). \tag{6.23}
\end{aligned}$$

したがって,

$$\begin{aligned}
|E^R(\phi)| &= \sqrt{E_r^R(\phi)^2 + E_i^R(\phi)^2} \\
&= M \left| \sin\left(\frac{\phi}{2}\right) \right|. \tag{6.24}
\end{aligned}$$

(証明終)

回転汎化誤差(式(6.12))は、未学習パターンが学習パターンから離れる(ϕ が大きくなる)につれて増加し、最も離れた地点($\phi=180$ 度)において最大値 M をとり、更に学習パターンに近づくにつれて減少してゆくことを示している。

6.3.2 相似変換の学習

次に、相似変換を学習した複素BPネットワークの性質をみる。

学習パターンは、複素平面において実軸と x 度の角度をなす直線上に等間隔に並んだ p 個の点列を原点との距離に関して β 倍させるものとする(図6.2)。つまり、任意の $1 \leq k \leq p$ に対して、入力パターンは

$$ka \exp[ix] \tag{6.25}$$

であり、それに対応する出力パターンは

$$ka\beta \exp[ix] \tag{6.26}$$

である。ただし、 $a \in \mathbf{R}^+$ は常数であり、点列の間隔を表す。また、 $k, p \in \mathbf{N}$; $x, \beta \in \mathbf{R}$, $0 < pa\beta \leq 1$ である。ここで、6.3.1項と同様にして、式(6.26)で示される出力パターン(-1 から 1 の間の値をとる)は、ニューラルネットワークに与える際には、 0 から 1 の間の値へ変

換しておく必要がある。変換の結果、ニューラルネットワークに与えられる出力パターンは、任意の $1 \leq k \leq p$ に対して、

$$\frac{1}{2}ka\beta \exp[ix] + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \quad (6.27)$$

とやや複雑な形で表される。

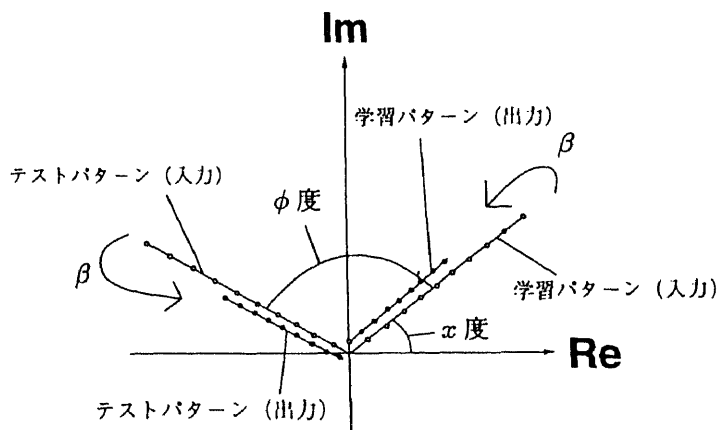


図 6.2 学習パターンとテストパターン (相似率 β の相似変換)

次の定理は複素BPの相似率に関する汎化能力の定性的性質を述べたものである。

定理 5 $1 \leq k \leq p$ を任意に固定する。式 (6.25) および式 (6.27) で示される学習パターンを学習した複素BPネットワークに対して、『入学習パターン $ka \exp[ix]$ を原点を中心に正の方向に任意の角度 ϕ だけ回転させた点 $ka \exp[i(x + \phi)]$ 』を入力テストパターンとして与える (図 6.2)。このとき、複素BPネットワークが出力する値は

$$\left[\frac{1}{2}ka\beta \exp[i(x + \phi)] + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \right] + E^S(\phi) \in \mathbb{C} \quad (6.28)$$

で与えられる。ここで、式 (6.28) の第 1 項は未学習パターン $ka \exp[i(x + \phi)]$ を原点との距離に関して『学習した相似率 β 』だけ相似変換させた点に相当する (図 6.2)。また、 $E^S(\phi)$ はそのときに生じるずれを表す複素数であり、その絶対値 (相似汎化誤差) は、

$$|E^S(\phi)| = M \left| \sin\left(\frac{\phi}{2}\right) \right| \quad (6.29)$$

で与えられる。

(注意)

式 (6.29) における M は式 (6.1) で定義された常数であり、学習終了後の学習パラメータの値には依存するが、 ϕ には依存しない。

証明は、定理 4 と同様にしてできるので省略する。

相似汎化誤差 (式 (6.29)) は、回転汎化誤差 (式 (6.12)) と同様に、未学習パターンが学習パターンから離れるほど大きいことを意味する。

6.3.3 平行移動の学習

最後に、複素 BP ネットワークに平行移動を学習させた場合、どのようなふるまいをするかについて明らかにする。

学習パターンは、複素平面において実軸と x 度の角度をなす直線上に等間隔に並んだ p 個の点列を、複素数 $\gamma = \tau \exp[i\omega]$ で規定される平行移動によって平行移動させるものとする (図 6.3)。 γ は平行移動ベクトルを複素数で表したものであり、 τ は平行移動の距離、 ω は方向を表している。

学習パターンは次のように表現できる。すなわち、任意の $1 \leq k \leq p$ に対して、入力パターンは

$$ka \exp[ix] \quad (6.30)$$

であり、それに対応する出力パターンは

$$ka \exp[ix] + \gamma \quad (6.31)$$

である。ただし、 $a \in \mathbf{R}^+$ は常数であり、点列の間隔を表す。また、 $k, p \in \mathbf{N}$; $x \in \mathbf{R}$; $\gamma \in \mathbf{C}$,

$$-1 \leq \operatorname{Re}[pa \exp[ix] + \gamma], \operatorname{Im}[pa \exp[ix] + \gamma] \leq 1 \quad (6.32)$$

である。式 (6.32) は、学習パターンを複素平面において -1 から 1 の範囲に押えておくためのものである。ここで、6.3.1 項と同様にして、式 (6.31) で示される出力パターン (-1 から 1 の間の値をとる) は、ニューラルネットワークに与える際には、 0 から 1 の間の値へ変換しておく必要がある。変換の結果、ニューラルネットワークに与えられる出力パターンは、任意の $1 \leq k \leq p$ に対して、

$$\frac{1}{2}(ka \exp[ix] + \gamma) + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \quad (6.33)$$

とやや複雑な形で表される。

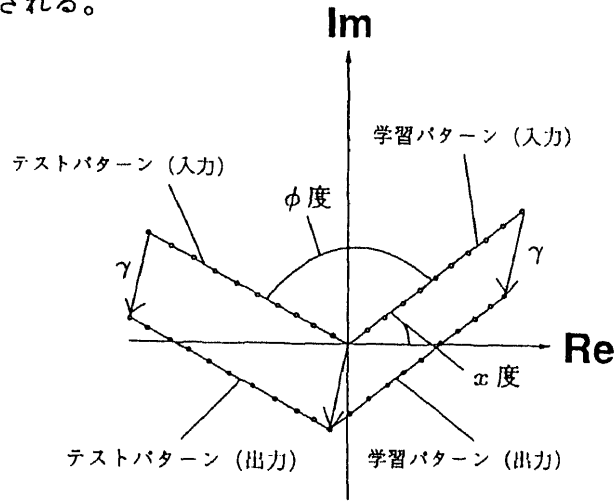


図 6.3 学習パターンとテストパターン (平行移動ベクトル γ の平行移動)

定理 6 は複素 B P の平行移動に関する汎化能力の定性的性質を示す。

定理 6 $1 \leq k \leq p$ を任意に固定する。式 (6.30) および式 (6.33) で示される学習パターンを学習した複素 B P ネットワークに対して, 『入学習パターン $ka \exp[ix]$ を原点を中心にして正の方向に任意の角度 ϕ だけ回転させた点 $ka \exp[i(x + \phi)]$ 』を入力テストパターンとして与える (図 6.3)。このとき, 複素 B P ネットワークが出力する値は

$$\left[\frac{1}{2} \{ ka \exp[i(x + \phi)] + \gamma \} + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right] \right] + E^P(\phi) \in \mathbb{C} \quad (6.34)$$

で与えられる。ここで, 式 (6.34) の第 1 項は未学習パターン $ka \exp[i(x + \phi)]$ を『学習した平行移動ベクトル γ 』に従って平行移動させた点に相当する (図 6.3)。また, $E^P(\phi)$ はそのときに生じるずれを表す複素数であり, その絶対値 (平行移動汎化誤差) は,

$$|E^P(\phi)| = M(\gamma) \left| \sin \left(\frac{\phi}{2} \right) \right| \quad (6.35)$$

で与えられる。

(注意)

式 (6.35) における $M(\gamma)$ は式 (6.1) で定義された常数であり, 学習終了後の学習パラメータの値には依存するが, ϕ には依存しない。

証明は, 定理 4 と同様にしてできるので省略する。

定理 6 は, 定理 4 や定理 5 で示した内容とほぼ同じことを述べている。すなわち, 平行移動汎化誤差 (式 (6.35)) は, 未学習パターンと学習パターンとの離れ具合を示す ϕ に依存して変動する。

6.4 2つの変換率の学習能力の解析

本節では、2つの変換率を複素BPにより学習させたニューラルネットワークのふるまいについて、回転角度を例にとって調べる。

学習パターンは2種類である。1つは複素平面において実軸と x 度の角度をなす直線上に等間隔に並んだ p 個の点列を原点を中心にして正の方向に α 度だけ回転させるもの(学習パターン1)であり、もう1つは実軸と y 度の角度をなす直線上に等間隔に並んだ p 個の点列を原点を中心にして正の方向に β 度だけ回転させるもの(学習パターン2)である(図6.4)。すなわち、任意の $1 \leq k \leq p$ に対して、学習パターン1の入力パターン(入力学習パターン1)は

$$ka \exp[ix] \quad (6.36)$$

($a \in \mathbf{R}^+$ は点列の間隔を表す)であり、それに対応する出力パターン(出力学習パターン1)は

$$ka \exp[i(x + \alpha)] \quad (6.37)$$

である。ここで、6.3節と同様にして、式(6.37)で示される出力パターン(-1から1の間の値をとる)は、ニューラルネットワークに与える際には、0から1の間の値へ変換しておく必要がある。変換の結果、ニューラルネットワークに与えられる出力パターンは、任意の $1 \leq k \leq p$ に対して、

$$\frac{1}{2}ka \exp[i(x + \alpha)] + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \quad (6.38)$$

とやや複雑な形で表される。また、学習パターン2の入力パターン(入力学習パターン2)は

$$ka \exp[iy] \quad (6.39)$$

であり、それに対応する出力パターン(出力学習パターン2)は

$$ka \exp[i(y + \beta)] \quad (6.40)$$

である。ここで、学習パターン1と同様にして、式(6.40)で示される出力パターンを変換すると、任意の $1 \leq k \leq p$ に対して、

$$\frac{1}{2}ka \exp[i(y + \beta)] + \frac{1}{\sqrt{2}} \exp\left[i\frac{\pi}{4}\right] \quad (6.41)$$

となる。ただし、 $k, p \in \mathbf{N}$; $x, y, \alpha, \beta \in \mathbf{R}$; $0 < pa \leq 1$ である。

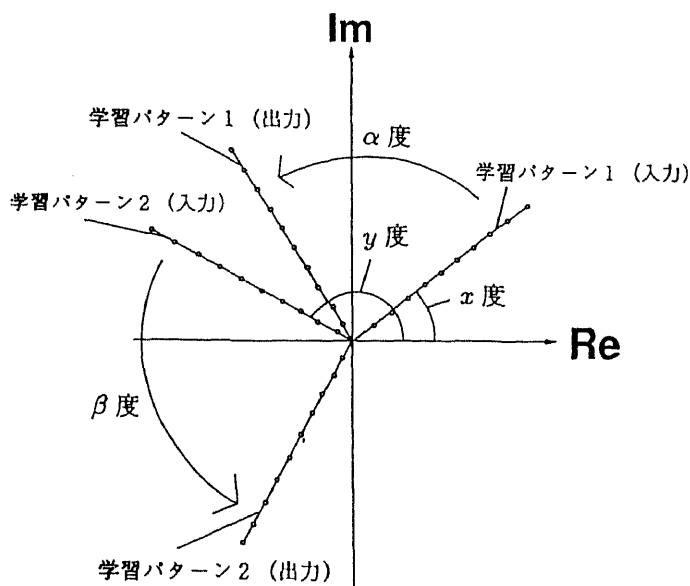


図6.4 学習パターン (α 度と β 度の回転)

そこで、上記の学習パターンを学習した複素BPネットワークのふるまいの定性的性質を明らかにする。準備として、次の補題を用意する。

補題 2 任意の $1 \leq k \leq p$ に対して、近似的に次の等式が成り立つ。

$$K \left[G \cos(x + w^0 + d^0) + H_R \right] + \frac{1}{2} = \frac{1}{2} ka \cos(x + \alpha) + \frac{1}{2}, \quad (6.42)$$

$$K \left[G \sin(x + w^0 + d^0) + H_I \right] + \frac{1}{2} = \frac{1}{2} ka \sin(x + \alpha) + \frac{1}{2}, \quad (6.43)$$

$$K \left[G \cos(y + w^0 + d^0) + H_R \right] + \frac{1}{2} = \frac{1}{2} ka \cos(y + \beta) + \frac{1}{2}, \quad (6.44)$$

$$K \left[G \sin(y + w^0 + d^0) + H_I \right] + \frac{1}{2} = \frac{1}{2} ka \sin(y + \beta) + \frac{1}{2}. \quad (6.45)$$

(証明)

2つの学習パターンに対して、それぞれ、補題1の証明の方法を適用することにより、補題の主張を得る。 (証明終)

次の定理は、2つの回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の(距離に関する)誤差の評価式を与えるものであり、汎化能力の一端を示すものである。

定理 7 $1 \leq k \leq p$ を任意に固定する。式 (6.36) から式 (6.41) で示される 2 つの学習パターンを学習した複素 BP ネットワークに対して, 「入学習パターン 1 (式 (6.36)) と入学習パターン 2 (式 (6.39)) との中間に位置するパターン」 $ka \exp[i((x+y)/2)]$ (入力テストパターン 1) および $ka \exp[i((x+y)/2+\pi)]$ (入力テストパターン 2) を与える (図 6.5)。このとき, 入力テストパターン 1 および入力テストパターン 2 に対して複素 BP ネットワークが出力する値は, それぞれ,

$$\left[\frac{1}{2} ka \exp \left[i \left(\frac{x+y}{2} + \frac{\alpha+\beta}{2} \right) \right] + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right] \right] + E^{(1)}(k) \in \mathbb{C}, \quad (6.46)$$

$$\left[\frac{1}{2} ka \exp \left[i \left(\frac{x+y}{2} + \pi + \frac{\alpha+\beta}{2} \right) \right] + \frac{1}{\sqrt{2}} \exp \left[i \frac{\pi}{4} \right] \right] + E^{(2)}(k) \in \mathbb{C} \quad (6.47)$$

で与えられる。ここで, 式 (6.46) および式 (6.47) の第 1 項は入力テストパターンを原点を中心にして正の方向に『学習した角度 α と β の中間的な角度 $(\alpha+\beta)/2$ 』だけ回転させた点に相当する。また, $E^{(1)}(k)$ および $E^{(2)}(k)$ はそのときに生じるずれを表す複素数であり, その絶対値 (回転汎化誤差) は,

$$|E^{(1)}(k)| = \sqrt{\left[2KG \sin^2 \left(\frac{x-y}{2} \right) - ka \sin^2 \left(\frac{x-y}{4} + \frac{\alpha-\beta}{4} \right) \right]^2 + 8kaKG \sin^2 \left(\frac{x-y}{4} + \frac{\alpha-\beta}{4} \right) \cdot \sin^2 \left(\frac{x-y}{4} \right) \cdot \sin^2 \left(\frac{\alpha+\beta}{4} - \frac{w^0+d^0}{2} \right)}, \quad (6.48)$$

$$|E^{(2)}(k)| = \sqrt{|E^{(1)}(k)|^2 + Q} \quad (6.49)$$

で与えられる。

(注意)

(1) 入力テストパターン 1 の回転汎化誤差 (式 (6.48)) と入力テストパターン 2 の回転汎化誤差 (式 (6.49)) との差は Q (式 (6.1)) である。(2) $x=y$, かつ, $\alpha=\beta$ の場合 (つまり, 6.3.1 節におけるモデルと等価), 回転汎化誤差は,

$$|E^{(1)}(k)| = 0, \quad (6.50)$$

$$|E^{(2)}(k)| = M \quad (6.51)$$

となる。ここで, 入力テストパターン 1 は学習パターンそのものであるので, 回転汎化誤差が 0 であることは理に適っている。また, 学習パターンと 180 度の角度をなす入力テストパターン 2 に対する回転汎化誤差が M であることは, 定理 4 の主張と一致する。

(証明)

入力テストパターン 1 に関する主張の証明と入力テストパターン 2 に関する主張の証明は同じ方法によって行なうことができるので, 入力テストパターン 1 に関する主張の証明に

ついでのみ示す。また、その証明は定理4のものと本質的に同じであるので、簡単に述べるに留める。補題2で示した関係式をもとに、[式(6.42)+式(6.44)]/2、[式(6.43)+式(6.45)]/2を計算することにより、入力テストパターン1に対する複素BPネットワークの出力値を求め、その出力値を[入力テストパターン1を原点を中心にして正の方向に $(\alpha + \beta)/2$ 度回転させた点]+[ずれ]という形に変形する。ちなみに、ずれを表す複素数 $E^{(1)}(k)$ の実部 $E_r^{(1)}(k)$ および虚部 $E_i^{(1)}(k)$ は、それぞれ、

$$E_r^{(1)}(k) = \frac{1}{2}ka \cos\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right] - KG \cos\left(\frac{x+y}{2} + w^0 + d^0\right) \cdot \left[\cos\left(\frac{x-y}{2}\right) - 1\right], \quad (6.52)$$

$$E_i^{(1)}(k) = \frac{1}{2}ka \sin\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right] - KG \sin\left(\frac{x+y}{2} + w^0 + d^0\right) \cdot \left[\cos\left(\frac{x-y}{2}\right) - 1\right] \quad (6.53)$$

となる。式(6.48)は、この式(6.52)および式(6.53)から得られる。 (証明終)

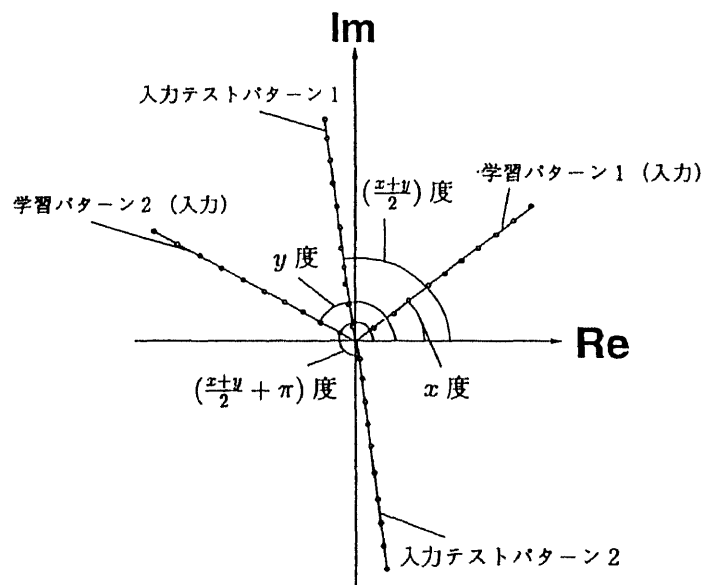


図6.5 テストパターン $(\alpha$ 度と β 度の回転)

次の定理は、2つの回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の『(角度に関する)誤差が無視できるほど小さな値であるための十分条件』を与えるものである。

定理 8 $1 \leq k \leq p$ を任意に固定する。式 (6.36) から式 (6.41) で示される 2 つの学習パターンを学習した複素 BP ネットワークに対して、【入力学習パターン 1 (式 (6.36)) と入力学習パターン 2 (式 (6.39)) との中間に位置するパターン】 $ka \exp[i((x+y)/2)]$ (入力テストパターン 1) および $ka \exp[i((x+y)/2 + \pi)]$ (入力テストパターン 2) を与える。このとき、複素 BP ネットワークは、それぞれの入力パターンに対して、誤差 $E^{(1)}(k), E^{(2)}(k) \in \mathbb{C}$ を伴いつつ、入力テストパターンを原点を中心にして正の方向に【学習した角度 α と β の中間的な角度 $(\alpha + \beta)/2$ 】だけ回転させる。ここで、入力テストパターン 1 に対する複素 BP ネットワークの出力値の回転角度に関する誤差 $E_{angle}^{(1)}(k)$ を

$$E_{angle}^{(1)}(k) \stackrel{\text{def}}{=} \min_{n \in \mathbb{Z}} \left| \frac{x+y}{2} + \frac{\alpha+\beta}{2} + 2n\pi - \angle Output^{(1)}(k) \right|, \quad (6.54)$$

また、入力テストパターン 2 に対するそれを

$$E_{angle}^{(2)}(k) \stackrel{\text{def}}{=} \min_{n \in \mathbb{Z}} \left| \frac{x+y}{2} + \pi + \frac{\alpha+\beta}{2} + 2n\pi - \angle Output^{(2)}(k) \right| \quad (6.55)$$

と定義する (以下、回転汎化角度誤差と呼ぶ。 \mathbb{Z} は整数全体の集合。図 6.6 参照)。ただし、 $\angle Output^{(1)}(k)$ および $\angle Output^{(2)}(k)$ は、それぞれ、入力テストパターン 1 および入力テストパターン 2 に対する複素 BP ネットワークの出力値の偏角を表す。また、

$$0 \leq \alpha - \beta \leq y - x \leq \pi \quad (6.56)$$

とする。

このとき、 KG が十分小さいならば、回転汎化誤差 $|E^{(1)}(k)|$ は、

$$|E^{(1)}(k)| \approx ka \sin^2 \left(\frac{\psi}{4} \right) \quad (-\pi \leq \psi \leq 0) \quad (6.57)$$

で与えられ、かつ、回転汎化角度誤差 $E_{angle}^{(1)}(k)$ は無視できるほど小さい。また、 K および KG が十分小さいならば、回転汎化誤差 $|E^{(2)}(k)|$ は、

$$|E^{(2)}(k)| \approx ka \sin^2 \left(\frac{\psi}{4} \right) \quad (-\pi \leq \psi \leq 0) \quad (6.58)$$

で与えられ、かつ、回転汎化角度誤差 $E_{angle}^{(2)}(k)$ は無視できるほど小さい。ただし、 $\psi \stackrel{\text{def}}{=} (x + \alpha) - (y + \beta)$ 。 ψ は出力学習パターン 1 の偏角と出力学習パターン 2 の偏角の差を表す。

(注意)

式 (6.56) は次の 3 つの式に分けることができ、それらの意味は、それぞれ、次の通りである。

1. $\beta \leq \alpha$

学習パターン 2 の回転角度は、学習パターン 1 の回転角度を越えることはない。

2. $x + \alpha \leq y + \beta$

出力学習パターン1の偏角は、出力学習パターン2の偏角を越えることはない。

3. $0 \leq y - x \leq \pi$

入力学習パターン2の偏角から入力学習パターン1の偏角を差し引いたものは180度以内である。

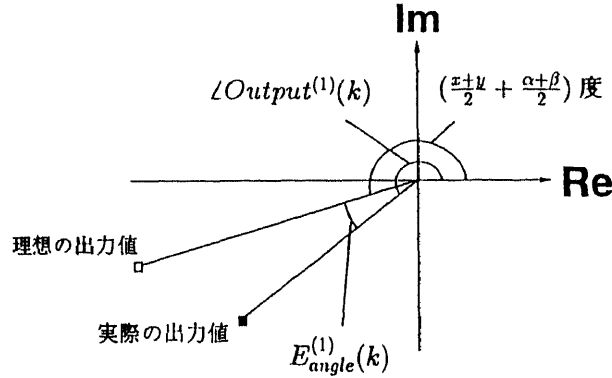


図6.6 回転汎化角度誤差 $E_{angle}^{(1)}(k)$

(証明)

入力テストパターン1に関する主張の証明と入力テストパターン2に関する主張の証明は本質的に同じであるので、入力テストパターン1に関する主張の証明についてのみ示す。まず、定理7により、複素BPネットワークは入力テストパターン1を誤差 $E^{(1)}(k) \in \mathbb{C}$ を伴いつつ、 $(\alpha + \beta)/2$ 度回転させる。ここで、誤差 $E^{(1)}(k)$ の実部 $E_r^{(1)}(k)$ (式(6.52)) および虚部 $E_i^{(1)}(k)$ (式(6.53)) は、それぞれ、 KG が十分小さいから、

$$\begin{aligned}
 E_r^{(1)}(k) &= \frac{1}{2}ka \cos\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right] \\
 &\quad - KG \cos\left(\frac{x+y}{2} + w^0 + d^0\right) \cdot \left[\cos\left(\frac{x-y}{2}\right) - 1\right] \\
 &\approx \frac{1}{2}ka \cos\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right], \quad (6.59)
 \end{aligned}$$

$$\begin{aligned}
 E_i^{(1)}(k) &= \frac{1}{2}ka \sin\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right] \\
 &\quad - KG \sin\left(\frac{x+y}{2} + w^0 + d^0\right) \cdot \left[\cos\left(\frac{x-y}{2}\right) - 1\right] \\
 &\approx \frac{1}{2}ka \sin\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right] \quad (6.60)
 \end{aligned}$$

である。よって,

$$\begin{aligned}\frac{E_i^{(1)}(k)}{E_r^{(1)}(k)} &\approx \frac{\frac{1}{2}ka \sin\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right]}{\frac{1}{2}ka \cos\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right) \cdot \left[\cos\left(\frac{x-y}{2} + \frac{\alpha-\beta}{2}\right) - 1\right]} \\ &= \tan\left(\frac{x+y}{2} + \frac{\alpha+\beta}{2}\right).\end{aligned}\quad (6.61)$$

故に, $E^{(1)}(k)$ の偏角を $\varphi_E \stackrel{\text{def}}{=} \arctan(E_i^{(1)}(k)/E_r^{(1)}(k))$ とおくと,

$$\varphi_E = \frac{x+y}{2} + \frac{\alpha+\beta}{2}\quad (6.62)$$

となる。

ここで, 複素平面を考え,

- (X_0, Y_0) : 誤差 $E^{(1)}(k)$ を 0 としたときの複素BPネットワークの出力値を表す座標 (以下, 理想の出力値と呼ぶ),
- (X_1, Y_1) : 誤差 $E^{(1)}(k)$ を含めた複素BPネットワークの出力値を表す座標 (以下, 実際の出力値と呼ぶ)

とすると,

$$\arctan\left(\frac{Y_0}{X_0}\right) = \frac{x+y}{2} + \frac{\alpha+\beta}{2},\quad (6.63)$$

$$\varphi_E = \arctan\left(\frac{Y_1 - Y_0}{X_1 - X_0}\right)\quad (6.64)$$

$$(E_r^{(1)}(k) = X_1 - X_0, E_i^{(1)}(k) = Y_1 - Y_0 \text{ による})$$

である。故に, 式(6.62)-(6.64)により,

$$\arctan\left(\frac{Y_0}{X_0}\right) = \arctan\left(\frac{Y_1 - Y_0}{X_1 - X_0}\right).\quad (6.65)$$

よって,

$$\frac{Y_0}{X_0} = \frac{Y_1 - Y_0}{X_1 - X_0}.\quad (6.66)$$

そこで, $T = Y_0/X_0$ とおくと,

$$Y_0 = TX_0,\quad (6.67)$$

$$Y_1 - Y_0 = T(X_1 - X_0).\quad (6.68)$$

よって,

$$Y_0 = TX_0, \quad (6.69)$$

$$Y_1 = TX_1 \quad (6.70)$$

となり, 理想の出力値 (X_0, Y_0) および実際の出力値 (X_1, Y_1) は, 共に, 直線 $y = Tx$ 上に乗っていることがわかる。一方, KG は十分小さいから, 式 (6.48) により,

$$\begin{aligned} |E^{(1)}(k)| &\approx ka \sin^2\left(\frac{\psi}{4}\right) \\ &\leq \frac{1}{2}ka. \quad (-\pi \leq \psi \leq 0 \text{ による}) \end{aligned} \quad (6.71)$$

故に, 複素BPネットワークの実際の出力値 (X_1, Y_1) および理想の出力値 (X_0, Y_0) は, 共に, 直線 $y = Tx$ 上に乗っており, それら2つの点の距離は高々 $(1/2)ka$ しか離れていない。そして, 理想の出力値 (X_0, Y_0) と原点との距離は ka である。よって, 実際の出力値 (X_1, Y_1) の位置に関しては, (a) 原点 と 理想の出力値 (X_0, Y_0) との間に (X_1, Y_1) が位置している場合と, (b) (X_1, Y_1) と 原点 との間に (X_0, Y_0) が位置する場合の2通り以外には有り得ない (図 6.7)。故に, (X_0, Y_0) と (X_1, Y_1) との間に原点が位置する場合は有り得ない。従って, ある $n_0 \in \mathcal{Z}$ が存在して,

$$\angle Output^{(1)}(k) \approx \frac{x+y}{2} + \frac{\alpha+\beta}{2} + 2n_0\pi. \quad (6.72)$$

すなわち,

$$\begin{aligned} E_{angle}^{(1)}(k) &\approx \min_{n \in \mathcal{Z}} \left| \frac{x+y}{2} + \frac{\alpha+\beta}{2} + 2n\pi - \left[\frac{x+y}{2} + \frac{\alpha+\beta}{2} + 2n_0\pi \right] \right| \\ &= 0. \end{aligned} \quad (6.73)$$

また, 式 (6.71) により, 回転汎化誤差 $|E^{(1)}(k)|$ は式 (6.57) で与えられることがわかる。

(証明終)

定理 8からは, 次のような性質が読み取れる。

1. 回転汎化誤差 $|E^{(1)}(k)|$ および $|E^{(2)}(k)|$ は, k が大きくなるにつれて増加する。つまり, 原点から離れた点ほど回転汎化誤差は大きくなる。しかし, 回転汎化角度誤差は k の値には無関係に無視できるほど小さい,
2. 回転汎化誤差 $|E^{(1)}(k)|$ および $|E^{(2)}(k)|$ は, 2つの出力学習パターンの偏角の差 ψ に関する正弦関数によって表現される (式 (6.57) および式 (6.58))。つまり, 2つの出力学習パターンの偏角の差 ψ が0のとき最小値0を取り, ψ の絶対値が大きくなるにつ

れて増加し、 $\psi = -\pi$ のとき最大値 $(1/2)ka$ をとる。たとえば、 $x = 90$ 度、 $y = 270$ 度の場合、 $|E^{(1)}(k)| = ka \sin^2((\alpha - \beta - \pi)/4)$ 、ただし、 $0 \leq \alpha - \beta \leq \pi$ 。故に、 $\alpha = \beta$ のとき、 $|E^{(1)}(k)|$ は最大値 $(1/2)ka$ をとり、 $\alpha - \beta$ が増えるにつれて単調に減少し、 $\alpha = \beta + \pi$ のとき最小値 0 をとる (図 6.8)。ただし、これは、 K および KG が十分小さいという仮定のもとでの定性的性質である (ψ の値によって K および KG の値が変動することに注意)。

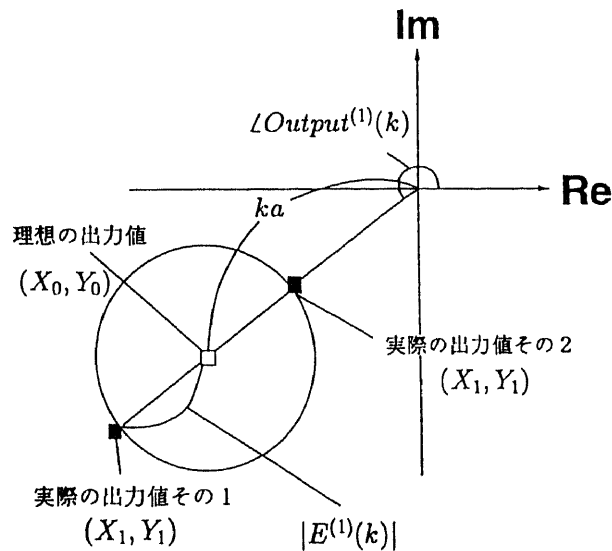
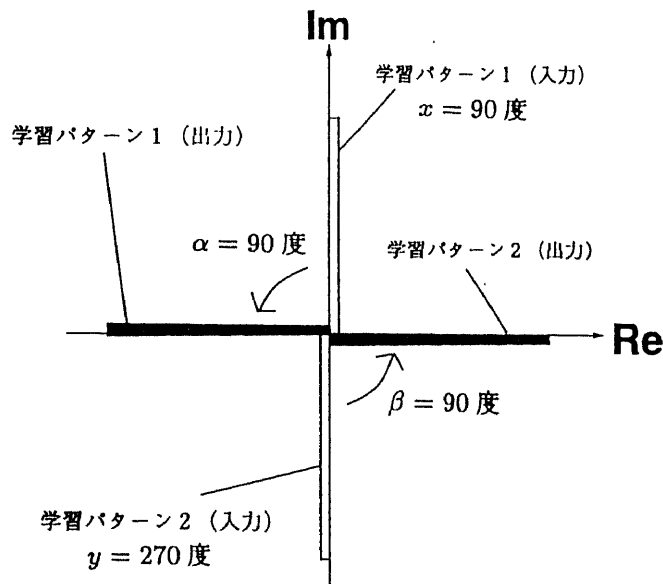
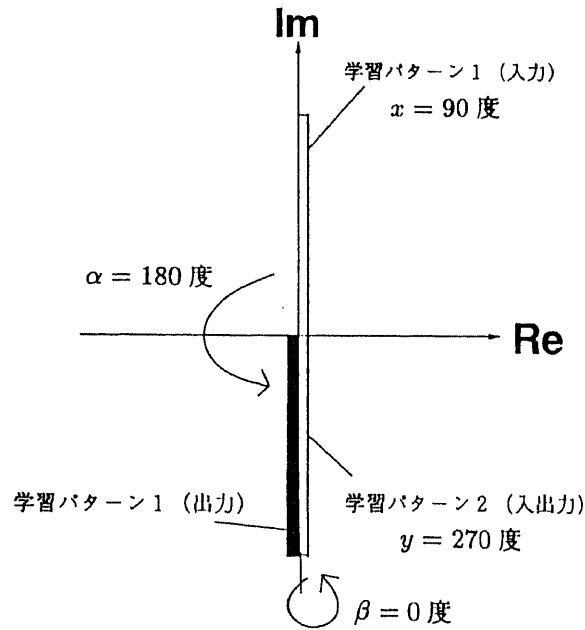


図 6.7 理想の出力値と実際の出力値との位置関係



(a) $|E^{(1)}(k)| = \frac{1}{2}ka$ (最大値)

図 6.8 回転汎化誤差の一例 ($x = 90$ 度、 $y = 270$ 度の場合)



(b) $|E^{(1)}(k)| = 0$ (最小値)

図 6. 8 回転汎化誤差の一例 ($x = 90$ 度, $y = 270$ 度の場合)

6.5 数値例

本節では、6.4節において示した定理7および定理8の数値例を示す。

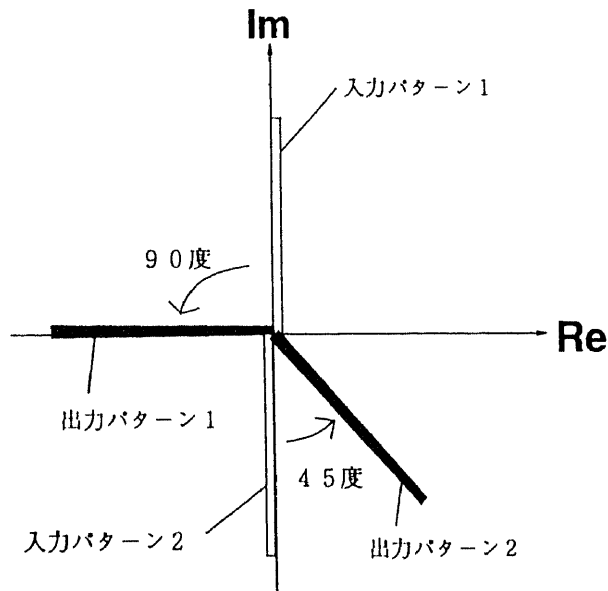
実験では、入力ニューロン1個、中間ニューロン1個、出力ニューロン1個の3層ネットワークを使い、学習パラメータの初期値は、0.0から1.0の間の乱数により設定し、学習率は0.5とする。学習パターンは、通常、数種類のパターンから構成されているが、その数種類のパターンをひと通り提示することで1回の学習と数えることとする。

学習させる2つの回転角度は 90度、45度 とする。90度回転に関する学習パターンを学習パターン1、45度回転に関する学習パターンを学習パターン2とする。入力学習パターン1を $ka \exp[ix]$ ($a = 0.1$; $x = 90$; $k = 1, \dots, 10$) とし、それらを原点を中心にして正の方向に $\alpha = 90$ 度回転させた点列を出力学習パターン1とする (図 6.9(a))。入力学習パターン2を $ka \exp[iy]$ ($a = 0.1$; $y = 270$; $k = 1, \dots, 10$) とし、それらを原点を中心にして正の方向に $\beta = 45$ 度回転させた点列を出力学習パターン2とする (図 6.9(a))。

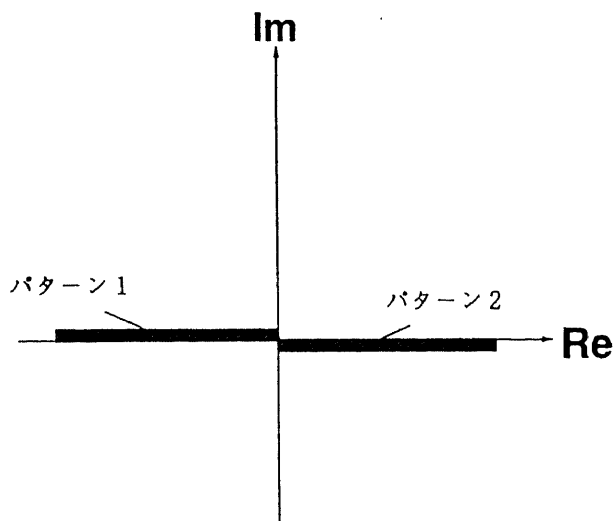
20,000回の学習の後、2つの入力学習パターンの中間に位置する未学習パターン $ka \exp[i180]$ (入力テストパターン1) および $ka \exp[i360]$ (入力テストパターン2) を入力テストパターンとして与えた (図 6.9(b))。実験結果を図 6.10 に示す。図 6.10(a) および図 6.10(b)

は、それぞれ、入力テストパターン1および入力テストパターン2に対する回転汎化誤差を、定理7による理論値と共に示したものである。縦軸は回転汎化誤差、横軸は原点からの離れ具合 $1 \leq k \leq 10$ である。

図6.11は、入力テストパターン1および入力テストパターン2に対する回転汎化角度誤差を示したものである。縦軸は理想の回転角度 $67.5 = (90 + 45)/2$ 度との差、横軸は原点からの離れ具合 $1 \leq k \leq 10$ である。



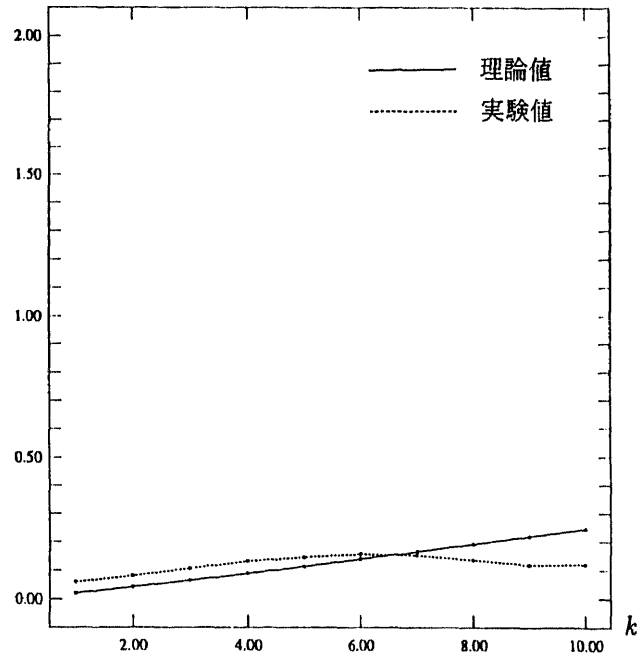
(a) 学習パターン



(b) 入力テストパターン

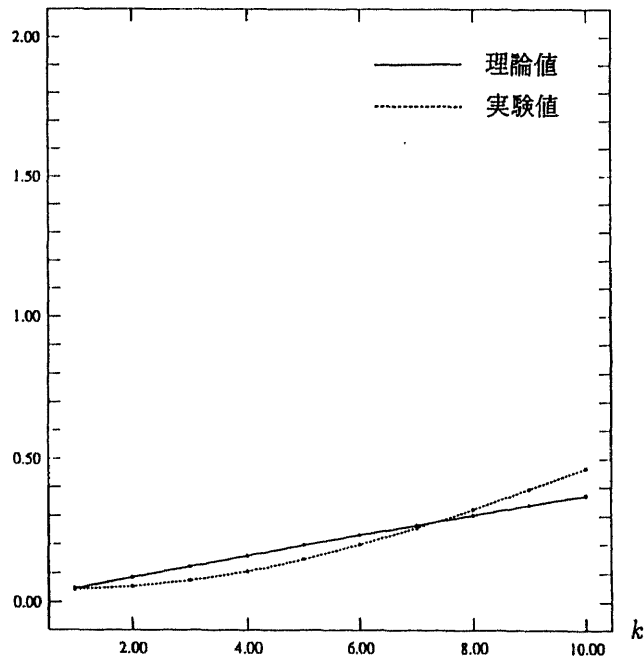
図6.9 90度および45度回転の学習

回転汎化誤差



(a) 入力テストパターン1に対する回転汎化誤差

回転汎化誤差



(b) 入力テストパターン2に対する回転汎化誤差

図6.10 90度および45度回転の学習実験における回転汎化誤差

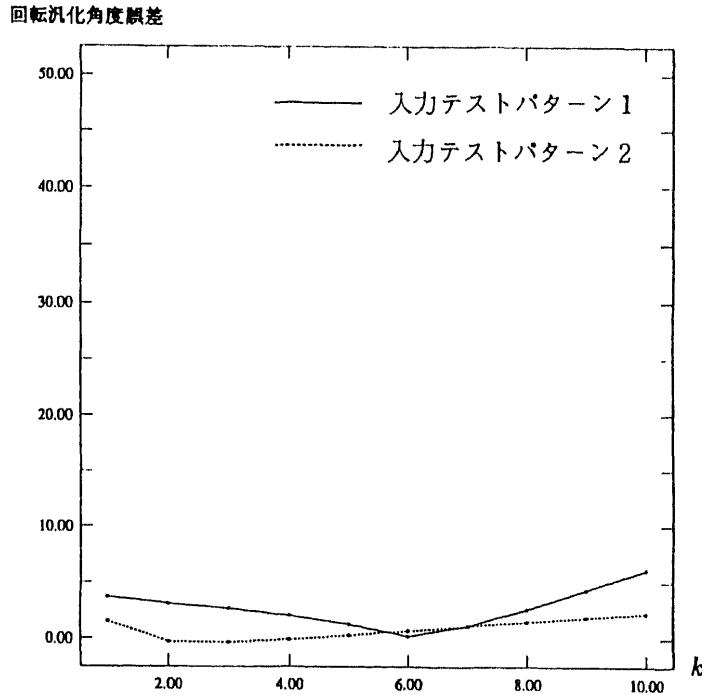


図 6.11 90度および45度回転の学習実験における回転汎化角度誤差

6.6 考察

5.3.4項で示した複素BPの2次元アフィン変換に関する汎化能力の誤差についての数値実験において、回転汎化誤差 $|E^R(\phi)|$ 、相似汎化誤差 $|E^S(\phi)|$ および平行移動汎化誤差 $|E^P(\phi)|$ は、 $\phi = 0$ のとき最も小さく、 ϕ が大きくなるにつれて増加し、 $\phi = 180$ あたりで最も大きくなっており、更に、 ϕ が360に近づくにつれて、概ね、除々に小さくなっている(図5.7)。この傾向は定理4から定理6で示したものとほぼ同じである。若干のずれはあるものの定理4から定理6は複素BPネットワークの定性的なふるまいの大要は捉えているものと言える。

次に、6.5節で示した定理7および定理8に関する数値実験結果について考察する。図6.10からわかるように、回転汎化誤差は、入力テストパターン1および入力テストパターン2、共に、定理7に示した理論値に沿っている。また、式(6.56)は満たされており、 K の値は0.031である。また、 KG の値は表6.1からわかるように十分小さいものである。よって、定理8が適用できる。実験におけるそのときの回転汎化角度誤差は $-1 \sim 6$ 度の間に入っており、ほぼ定理8の主張に沿った結果となっていることがわかる(図6.11)。

表 6.1 90度および45度回転の学習実験における KG の値

| | | | | | | | | | | |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| k | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| KG | 0.010 | 0.018 | 0.025 | 0.031 | 0.037 | 0.042 | 0.046 | 0.050 | 0.053 | 0.057 |

6.7 むすび

第5章において実験的に示した2次元アフィン変換学習能力は複素BPの最も特徴的な性質である。本章ではその2次元アフィン変換学習能力を数学的に解析し、定性的性質を明らかにした。すなわち、

1. 1種類の変換率を学習した複素BPネットワークのふるまいを明らかにした。変換率に関する汎化能力の(距離に関する)誤差は、未学習パターンと学習パターンの偏角の差に関する正弦関数によって表現される。
2. 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の(距離に関する)誤差の評価式を与えた。
3. 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の「(角度に関する)誤差が無視できるほど小さな値であるための十分条件」を求めた。

更に、これらの数値例を示し、解析結果の妥当性を確認した。

第 7 章

3次元ベクトル・バックプロパゲーション学習アルゴリズム

7.1 はじめに

前章までは、通常のバックプロパゲーション学習アルゴリズムの複素数版である複素BPを定式化し、その性質について調べてきた。

複素BPは実BPの2次元化パラメータ版であったわけであるが、本章では、さらに、この複素BPを群論に基づいて自然な形で 3次元パラメータ化した3次元ベクトル・バックプロパゲーション学習アルゴリズム（以下、3次元ベクトルBPと呼ぶ）を提案する ([31, 35, 42])。3次元ベクトルBPは、閾値、入力信号および出力信号がすべて3次元実ベクトル、重みパラメータが3次直交行列である階層型ニューラルネットワークに適用される。さらに、3次元ベクトルBPには、従来のニューラルネットワークには見られない3次元アフィン変換学習能力が備わっていることを確認する ([42, 43, 49])。この能力は複素BPが備えていた2次元アフィン変換学習能力に対応するものである。また、コンピュータビジョンへの応用についても述べる。

7.2節では、3次元ベクトルBPを定式化し、7.3節で3次元ベクトルBPの3次元アフィン変換学習能力について示し、7.4節で応用について述べる。最後に、7.5節で結論を述べる。

7.2 3次元ベクトルBP

本節では、3次元ベクトルBPモデルを定式化し、3次元ベクトルBP学習アルゴリズムの導出を行う。

7.2.1 3次元ベクトル・ニューロン

実BPは実数を扱うという点で 1次元的 であり、複素BPは複素数を扱うという点でいわば 2次元的 である。そのような実BPあるいは複素BPをさらに高次元パラメータ化する方法はいくつか考えられる。ひとつは、実数 α (1次元) を複素数 $z = x + iy$ (2次元)、4元数 $q = a + ib + jc + kd$ (4次元)、8元数 (8次元)、16元数 (16次元)、 \dots へと拡張してゆく方法である。もうひとつは、重みおよび閾値の次元数を1次元あるいは2次元から n 次

元実ベクトルを使って高次元パラメータ化する方法である。本稿では、後者の方法を使って3次元パラメータ化を試みる。

次のようなニューロン(3次元ベクトル・ニューロンと呼ぶ)を考える。入力信号、閾値および出力信号はすべて3次元実ベクトルであり、重みはすべて3次行列である。ニューロン j の内部ポテンシャル A_j は、実BPにおいて用いられるニューロンの内部ポテンシャルの類推で、

$$A_j = \sum_k W_{jk} S_k + T_j \quad (7.1)$$

と定義する。ここで、 W_{jk} はニューロン j とニューロン k との間の重みを表す3次行列、 S_k はニューロン k から出力されたニューロン j への入力信号を表す3次元実ベクトル、 T_j はニューロン j の閾値を表す3次元実ベクトルである。出力信号 $F_3(A_j)$ は次のように定義される：

$$F_3(A_j) = \begin{bmatrix} f_R(a_1) \\ f_R(a_2) \\ f_R(a_3) \end{bmatrix}, \quad \text{ただし, } A_j = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \quad (7.2)$$

$F_3(A_j)$ の各成分は、それぞれ、 A_j のそれぞれの各成分 a_m ($m = 1, 2, 3$)に関するシグモイド関数値である。

上記の定式化において、出力関数 F_3 にはいろいろなもの考えることができ、それらに応じてニューロンにはそれら固有の性質が備わることになる。また、一般に、重みを表す3次行列にはいろいろな制限を与えることができる。たとえば、正則性、対称性、直交性などであるが、これらはニューロンのふるまいに影響を与えることとなる。本稿では、重みは3次直交行列にしておく。この3次直交行列は、複素BPにおいて使われている重みの自然な拡張となっているからである。以下、このことを示す。複素BPネットワークにおけるニューロンの構造は次のとおりである。重み $w_k = w_k^r + iw_k^i \in C$ ($1 \leq k \leq n$)および閾値 $\theta = \theta^r + i\theta^i \in C$ を持つ n 入力のニューロンをひとつ考える。このとき、入力信号 $x_k + iy_k \in C$ ($1 \leq k \leq n$)が与えられると、このニューロンは $X + iY \in C$ を出力する。ここで、

$$\begin{aligned} X + iY &= f_C \left(\sum_{k=1}^n (w_k^r + iw_k^i)(x_k + iy_k) + (\theta^r + i\theta^i) \right) \\ &= f_R \left(\sum_{k=1}^n (w_k^r x_k - w_k^i y_k) + \theta^r \right) + i f_R \left(\sum_{k=1}^n (w_k^i x_k + w_k^r y_k) + \theta^i \right). \end{aligned} \quad (7.3)$$

さらに、

$$\begin{bmatrix} X \\ Y \end{bmatrix} = F_C \left(\left[\begin{array}{cc|ccc} w_1^r & -w_1^i & \cdots & w_n^r & -w_n^i \\ w_1^i & w_1^r & \cdots & w_n^i & w_n^r \end{array} \right] \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right)$$

$$= F_C \left(|w_1| \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 \\ \sin \alpha_1 & \cos \alpha_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \dots \right. \\ \left. + |w_n| \begin{bmatrix} \cos \alpha_n & -\sin \alpha_n \\ \sin \alpha_n & \cos \alpha_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right), \quad (7.4)$$

ここで, $F_C \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} f_R(x) \\ f_R(y) \end{bmatrix}$, $\alpha_k = \arctan(w_k^i/w_k^r)$ ($1 \leq k \leq n$) である。式(7.4)

において, $\begin{bmatrix} w_k^r & -w_k^i \\ w_k^i & w_k^r \end{bmatrix}$ は 2 次直交群 $O_2(\mathbf{R})$ の元である。このように, 重みが 3 次直交群 $O_3(\mathbf{R})$ の元である 3 次直交行列であることは自然であると言える。さらに, 出力関数 F_3 の形態などについても先に与えられた 3 次元ベクトル・ニューロンの定式化は自然であることがわかる。ちなみに, $\begin{bmatrix} \cos \alpha_k & -\sin \alpha_k \\ \sin \alpha_k & \cos \alpha_k \end{bmatrix}$ は 2 次回転群 $SO_2(\mathbf{R})$ の元であり, $|w_k|$ は $\mathbf{R}^* = \{x \in \mathbf{R} \mid x > 0\}$ (乗法に関する群) の元である。したがって, $\begin{bmatrix} w_k^r & -w_k^i \\ w_k^i & w_k^r \end{bmatrix}$ は群 $\mathbf{R}^* \times SO_2(\mathbf{R})$ の元でもあるわけである。

7.2.2 3次元ベクトル・ニューラルネットワーク

本項では, 3次元ベクトルBPが適用されるネットワークを, 7.2.1項で定義した3次元ベクトル・ニューロンを使って構成する。一般に, 3次元ベクトル・ニューロンから構成されるニューラルネットワークを3次元ベクトル・ニューラルネットワークと呼ぶことにするが, ここで定義するネットワークは, 3次元ベクトルBPが適用されるネットワークという意味で3次元ベクトルBPネットワークと呼ぶことにする。簡単のために, 3層ネットワークを例にとる。

入力ニューロン i と中間ニューロン j との間の重みを $W_{ji} \in O_3(\mathbf{R})$ ($O_3(\mathbf{R})$ は 3 次直交群), 中間ニューロン j と出力ニューロン k との間の重みを $V_{kj} \in O_3(\mathbf{R})$, 中間ニューロン j の閾値を $\Theta_j = {}^t[\theta_j^x \ \theta_j^y \ \theta_j^z] \in \mathbf{R}^3$, 出力ニューロン k の閾値を $\Gamma_k = {}^t[\gamma_k^x \ \gamma_k^y \ \gamma_k^z] \in \mathbf{R}^3$ とする。また, 入力ニューロン i への入力信号を $I_i = {}^t[I_i^x \ I_i^y \ I_i^z] \in \mathbf{R}^3$ とし, 中間ニューロン j および出力ニューロン k の出力信号を, それぞれ, $H_j = {}^t[H_j^x \ H_j^y \ H_j^z] \in \mathbf{R}^3$, $O_k = {}^t[O_k^x \ O_k^y \ O_k^z] \in \mathbf{R}^3$ とする。さらに, $\Delta^k = {}^t[\delta_k^x \ \delta_k^y \ \delta_k^z] = T_k - O_k \in \mathbf{R}^3$ を O_k と出力ニューロン k に対する教師パターン (出力学習パターン) $T_k = {}^t[T_k^x \ T_k^y \ T_k^z] \in \mathbf{R}^3$ との間の誤差とする。パターン p に対する 2 乗誤差を $E_p = (1/2) \sum_{k=1}^N \|T_k - O_k\|^2$, と定義する。

ただし, $\|x\| \stackrel{\text{def}}{=} \sqrt{x_1^2 + x_2^2 + x_3^2}$, $x = {}^t[x_1 \ x_2 \ x_3]$, N は出力ニューロンの総数である。重みは, ここでは, 特に次のように定義する:

$$W_{ji} = \begin{bmatrix} w_{ji}^x & -w_{ji}^y & 0 \\ w_{ji}^y & w_{ji}^x & 0 \\ 0 & 0 & w_{ji}^z \end{bmatrix} \in O_3(\mathbf{R}), \quad \text{ただし, } w_{ji}^z = \sqrt{(w_{ji}^x)^2 + (w_{ji}^y)^2}, \quad (7.5)$$

$$V_{kj} = \begin{bmatrix} v_{kj}^x & 0 & 0 \\ 0 & v_{kj}^y & -v_{kj}^z \\ 0 & v_{kj}^z & v_{kj}^y \end{bmatrix} \in O_3(\mathbf{R}), \quad \text{ただし, } v_{kj}^x = \sqrt{(v_{kj}^y)^2 + (v_{kj}^z)^2}. \quad (7.6)$$

式(7.5)および式(7.6)における W_{ji} , V_{kj} は群 $\mathbf{R}^* \times SO_3(\mathbf{R})$ の元でもある ($SO_3(\mathbf{R})$ は 3 次回転群)。なぜならば、これらは次のようにも表現できるからである：

$$W_{ji} = w_{ji}^z A(\alpha_{ji}) = w_{ji}^z \begin{bmatrix} \cos \alpha_{ji} & -\sin \alpha_{ji} & 0 \\ \sin \alpha_{ji} & \cos \alpha_{ji} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (7.7)$$

$$V_{kj} = v_{kj}^x B(\beta_{kj}) = v_{kj}^x \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta_{kj} & -\sin \beta_{kj} \\ 0 & \sin \beta_{kj} & \cos \beta_{kj} \end{bmatrix}, \quad (7.8)$$

ただし, $\alpha_{ji} = \arctan(w_{ji}^y/w_{ji}^x)$, $\beta_{kj} = \arctan(v_{kj}^z/v_{kj}^y)$. ここで, $B(\beta_{kj})$ は 3 次回転群 $SO_3(\mathbf{R})$ の元の標準形である, つまり, 任意の $X \in SO_3(\mathbf{R})$ に対して, ある直交行列 $Y \in O_3(\mathbf{R})$ が存在して, $YXY^{-1} = B(\beta_{kj})$ 。また, $B(\beta_{kj})$ は X 軸のまわりの右まわりの回転, $A(\alpha_{ji})$ は Z 軸のまわりの右まわりの回転を表している。

上記の定式化では, Y 軸のまわりの回転要因が含まれていないため, 一般性を欠くように思われるが, 一般に, 3次元ユークリッド空間における任意の合同変換 (2点間の距離を変えない変換) はあるユニタリー変換 (たとえば, 式(7.8)における $B(\beta_{kj})$) と平行移動によって一意に表現されるので, 上記の定式化は Y 軸のまわりの回転を (暗に) 含んでいると言える。しかしながら, 偏りのある定式化であることは否めない (これは一例であることに注意されたい)。

7.2.3 学習アルゴリズム

次に, 前項までに定義した 3次元ベクトル BP モデルに対する学習アルゴリズムを導出する。十分小さい学習率 $\varepsilon > 0$ に対して, 最急降下法を使うことにより, 学習パラメータは次の式にしたがって修正すべきであることがわかる。

$$\begin{aligned} \begin{bmatrix} \Delta v_{kj}^y \\ \Delta v_{kj}^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial v_{kj}^y} \\ \frac{\partial E_p}{\partial v_{kj}^z} \end{bmatrix}, & \begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial \gamma_k^x} \\ \frac{\partial E_p}{\partial \gamma_k^y} \\ \frac{\partial E_p}{\partial \gamma_k^z} \end{bmatrix}, \\ \begin{bmatrix} \Delta w_{ji}^x \\ \Delta w_{ji}^y \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial w_{ji}^x} \\ \frac{\partial E_p}{\partial w_{ji}^y} \end{bmatrix}, & \begin{bmatrix} \Delta \theta_j^x \\ \Delta \theta_j^y \\ \Delta \theta_j^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial \theta_j^x} \\ \frac{\partial E_p}{\partial \theta_j^y} \\ \frac{\partial E_p}{\partial \theta_j^z} \end{bmatrix}. \end{aligned} \quad (7.9)$$

式 (7.9) を計算すると、結局、

$$\begin{bmatrix} \Delta v_{kj}^y \\ \Delta v_{kj}^z \end{bmatrix} = \begin{cases} \begin{bmatrix} (v_{kj}^y/v_{kj}^x)H_j^x & H_j^y & H_j^z \\ (v_{kj}^z/v_{kj}^x)H_j^x & -H_j^z & H_j^y \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} & (v_{kj}^x \neq 0 \text{ のとき}) \\ \begin{bmatrix} 0 & H_j^y & H_j^z \\ 0 & -H_j^z & H_j^y \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} & (v_{kj}^x = 0 \text{ のとき}), \end{cases} \quad (7.10)$$

$$\begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} = \varepsilon \begin{bmatrix} (1 - O_k^x)O_k^x & 0 & 0 \\ 0 & (1 - O_k^y)O_k^y & 0 \\ 0 & 0 & (1 - O_k^z)O_k^z \end{bmatrix} \begin{bmatrix} \delta_k^x \\ \delta_k^y \\ \delta_k^z \end{bmatrix}, \quad (7.11)$$

$$\begin{bmatrix} \Delta w_{ji}^x \\ \Delta w_{ji}^y \end{bmatrix} = \begin{cases} \begin{bmatrix} I_i^x & I_i^y & (w_{ji}^x/w_{ji}^z)I_i^z \\ -I_i^y & I_i^x & (w_{ji}^y/w_{ji}^z)I_i^z \end{bmatrix} \begin{bmatrix} \Delta \theta_j^x \\ \Delta \theta_j^y \\ \Delta \theta_j^z \end{bmatrix} & (w_{ji}^z \neq 0 \text{ のとき}) \\ \begin{bmatrix} I_i^x & I_i^y & 0 \\ -I_i^y & I_i^x & 0 \end{bmatrix} \begin{bmatrix} \Delta \theta_j^x \\ \Delta \theta_j^y \\ \Delta \theta_j^z \end{bmatrix} & (w_{ji}^z = 0 \text{ のとき}), \end{cases} \quad (7.12)$$

$$\begin{bmatrix} \Delta \theta_j^x \\ \Delta \theta_j^y \\ \Delta \theta_j^z \end{bmatrix} = \begin{bmatrix} (1 - H_j^x)H_j^x & 0 & 0 \\ 0 & (1 - H_j^y)H_j^y & 0 \\ 0 & 0 & (1 - H_j^z)H_j^z \end{bmatrix} \sum_k \begin{bmatrix} v_{kj}^x & 0 & 0 \\ 0 & v_{kj}^y & v_{kj}^z \\ 0 & -v_{kj}^z & v_{kj}^y \end{bmatrix} \begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} \quad (7.13)$$

が得られる。

7.3 3次元アフィン変換学習能力

複素BPには2次元アフィン変換学習能力が備わっていた(第5章および第6章)。本節では、複素BPの自然な拡張である3次元ベクトルBPに3次元アフィン変換学習能力(3次元図形変換能力)が備わっていることを計算機実験により検証する。

実験に用いるネットワークは、入力層1ニューロン、中間層6ニューロン、出力層1ニューロンの3層構造である。つまり、1つの3次元ベクトル \mathbf{X} を入力し、1つの3次元ベクトル \mathbf{X}' を出力するものである。一般に、1つの3次元ベクトル $\mathbf{X} = [x_1 \ x_2 \ x_3]$ は3次元空間内の1点 (x_1, x_2, x_3) に対応する。よって、当該ネットワークは3次元空間内のある1点 (x_1, x_2, x_3) を他の1点 (x'_1, x'_2, x'_3) に変換するものであると考えることができる。ここでは、対象とする空間を、 x 軸、 y 軸および z 軸の取る値が -1 から 1 までであるような空間に制限しておく。ニューラルネットワークが出力する値は 0 から 1 の間の値であるが、便宜上、以下に示す図では、それらを -1 から 1 の間の値に変換した値で示す。

比較のために、実BPによる実験も行ったが、ネットワークは入力層3ニューロン、中間層15ニューロン、出力層3ニューロンとし、3次元ベクトル \mathbf{X} の第1成分 x_1 を入出力

ニューロン 1 に、第 2 成分 x_2 を入出力ニューロン 2 に、第 3 成分 x_3 を入出力ニューロン 3 に対応させた。

学習率 ϵ は 0.5 とし、学習パラメータの初期値は -0.3 と $+0.3$ の間の値をとる乱数により設定した。学習は誤差

$$\sqrt{2 \sum_p E_p} \quad (7.14)$$

が 0.05 となったときに収束 (終了) したものとみなし、収束の規範とした (E_p は 7.2.2 節で定義したパターン p に対する 2 乗誤差である)。また、学習パターンは、通常、数種類のパターンから構成されているが、その数種類のパターンをひと通り提示することで 1 回の学習と数えることとした。

ちなみに、“1-6-1” 3次元ベクトルBPネットワークと“3-15-3”実BPネットワークの学習 1 回あたりの時間計算量はほぼ等しく、領域計算量 (つまり、学習パラメータ数) は実BPの約半分となっている (表 7.1)。

表 7.1 3次元ベクトルBPと実BPの計算量の比較。時間計算量は 1 回の学習において行なわれる四則演算の総数。領域計算量は学習パラメータ (つまり、重みおよび閾値) の総数。

| ネットワーク | 時間計算量 | | | 領域計算量 | | |
|-----------------|-------|-----|-----|-------|----|-----|
| | 乗除算 | 加減算 | 合計 | 重み | 閾値 | 合計 |
| 3次元ベクトルBP 1-6-1 | 255 | 141 | 396 | 36 | 21 | 57 |
| 実BP 3-15-3 | 264 | 141 | 405 | 90 | 18 | 108 |

7.3.1 回転

まず、直線の回転に関する実験を行った (図 7.1 ~ 図 7.3)。学習パターンは 1 1 個であり、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.15)$$

上の 1 1 個の点 (点列の間隔は 0.1) を入力パターンとし、それらの点を z 軸のまわりを右まわりに 60 度の回転を行ない、その後 x 軸のまわりを右まわりに 45 度の回転を行なった点を出力パターンとした (図 7.1)。出力パターンは直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{6}}{4} \\ \frac{\sqrt{6}}{4} \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.16)$$

の上ののっている。入力パターンと出力パターンのなす角は 60 度である。なお、見やすくするために、図中では点列を省略し、点列を繋いだ直線あるいは曲線のみ示している。

学習の後、テストパターンとして、入力学習パターンと30度の角度をなす4つの直線上の点列を与えた（各直線上には0.1の等間隔で11個の点列がのっている）（図7.2）。各直線は次の式で与えられる。

(1) テストパターン1

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} \frac{\sqrt{3}}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix} \quad (0.0 \leq t \leq 1.0). \quad (7.17)$$

この直線は、入力学習パターン（式(7.15)）をy軸のまわりを左まわりに30度回転させて得られる。

(2) テストパターン2

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} \frac{\sqrt{3}}{2} \\ 0 \\ -\frac{1}{2} \end{bmatrix} \quad (0.0 \leq t \leq 1.0). \quad (7.18)$$

この直線は、入力学習パターン（式(7.15)）をy軸のまわりを右まわりに30度回転させて得られる。

(3) テストパターン3

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \quad (0.0 \leq t \leq 1.0). \quad (7.19)$$

この直線は、入力学習パターン（式(7.15)）をz軸のまわりを右まわりに30度回転させて得られる。

(4) テストパターン4

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \\ 0 \end{bmatrix} \quad (0.0 \leq t \leq 1.0). \quad (7.20)$$

この直線は、入力学習パターン（式(7.15)）をz軸のまわりを左まわりに30度回転させて得られる。

これらのテストパターンに対して、3次元ベクトルBPネットワークは、誤差はあるものの、【学習した回転】に従った点を出力する傾向性が見られた（図7.2～図7.3）。同様にし、実BPを使って実験したところ、当然ながら、実BPネットワークは教師パターン（出力学習パターン）に近い点を出力した（図7.2）。

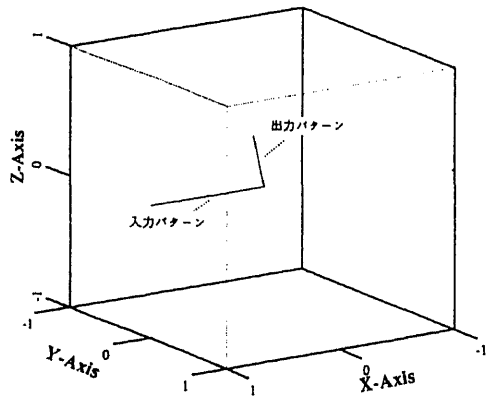


図7.1 学習パターン (回転)

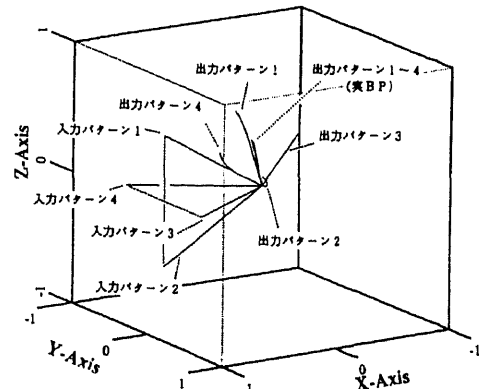
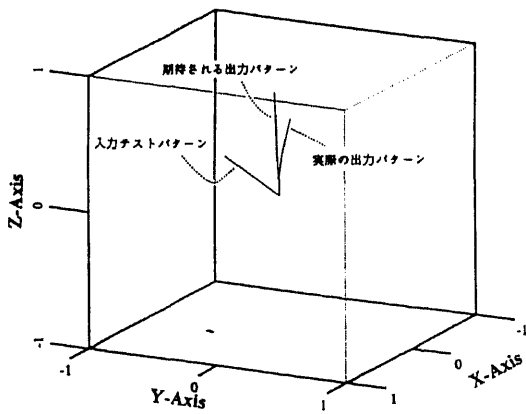
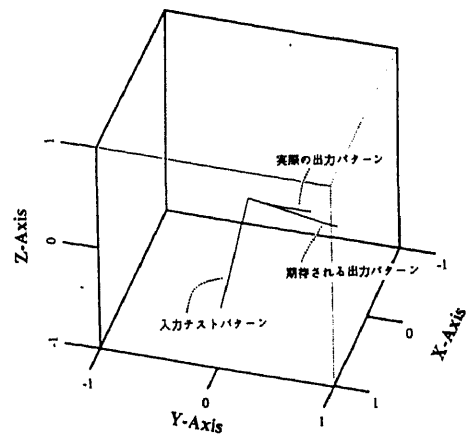


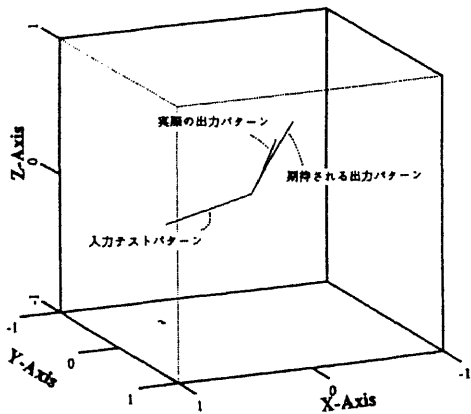
図7.2 テストパターン (回転)



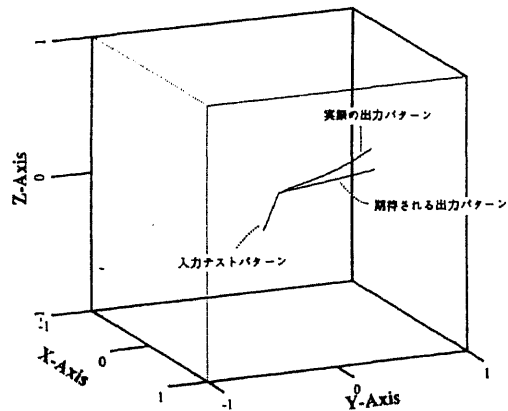
(a) テストパターン1 (回転)



(b) テストパターン2 (回転)



(c) テストパターン3 (回転)



(d) テストパターン4 (回転)

図7.3 テストパターン (回転)

7.3.2 相似変換

次に、相似変換に関する実験を行った(図7.4～図7.5)。学習パターンは11個であり、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.21)$$

上に等間隔に並んだ11個の点(点列の間隔は $\sqrt{3}/10$)を入力パターンとし、それらの点を原点との距離に関して2分の1に縮小した点を出力パターンとした(図7.4)。当然ながら、出力パターンも入力パターンがのっている直線(式(7.21))の上にある。

学習の後、テストパターンとして、3つの正方形のそれぞれの4辺の上に等間隔に並んだ48個の点を与えたところ、ニューラルネットワークはそれらの点を原点との距離に関して2分の1に縮小した点を出力する傾向が見られた(図7.5)。ただし、縮小された正方形は歪んでおり、入学習パターンに近いものほど原点との距離が2分の1に近い値を出力していて、それから離れるに従って徐々に誤差が大きくなっていっている。複素BPの2次元アフィン変換学習能力において、変換率に関する汎化能力の誤差は、未学習パターンと学習パターンの偏角の差に関する正弦関数によって表現される(6.3節)という性質があったが、実験の結果はこの性質が3次的に現れたものであると考えられる。ちなみに、実BPを使った実験では、出学習パターンに近い点が出力された(図7.5)。

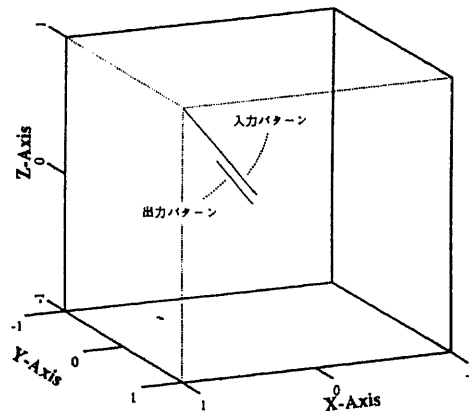
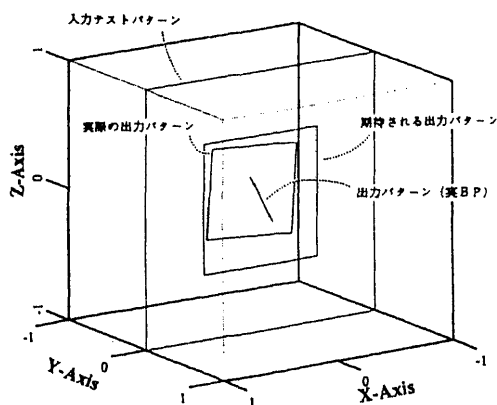
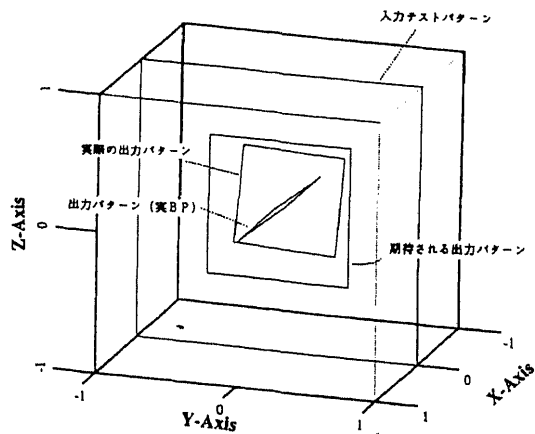


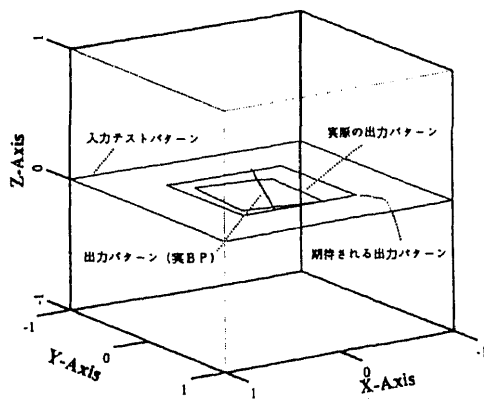
図7.4 学習パターン(相似変換)



(a) テストパターン 1 (相似変換)



(b) テストパターン 2 (相似変換)



(c) テストパターン 3 (相似変換)

図 7.5 テストパターン (相似変換)

7.3.3 平行移動

直線の平行移動を行った (図 7.6 ~ 図 7.7)。学習パターンは 11 個であり、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.22)$$

上に等間隔に並んだ 11 個の点 (点列の間隔は $\sqrt{2}/10$) を入力パターンとし、それらの点を x 軸方向、 y 軸方向、 z 軸方向にそれぞれ -0.5 ずつ平行移動させた点を出力パターンとした (図

7.6)。出力パターンは直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.23)$$

の上のっている。

学習終了後、テストパターンとして、出力学習パターンと同じ点列(テストパターン1), 入学習パターンを x 軸方向, y 軸方向にそれぞれ -0.5 ずつ平行移動して得られる点列(テストパターン2), おなじく入学習パターンを z 軸方向に -0.5 だけ平行移動して得られる点列(テストパターン3)を与えた(図7.7)。これらのテストパターンに対して, 3次元ベクトルBPネットワークは, 誤差はあるものの, 『学習した平行移動』に従った点を出力する傾向性が見られた。同様に, 実BPを使って実験したところ, 当然ながら, 実BPネットワークは教師パターン(出力学習パターン)に近い点を出力した(図7.7)。

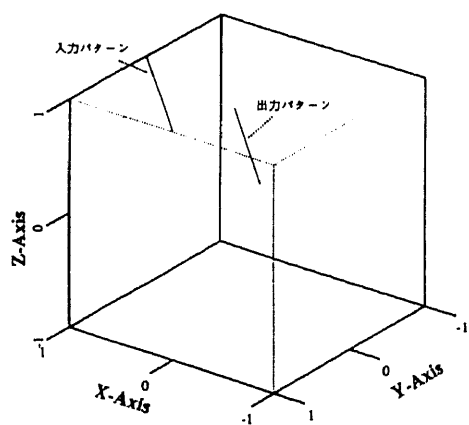
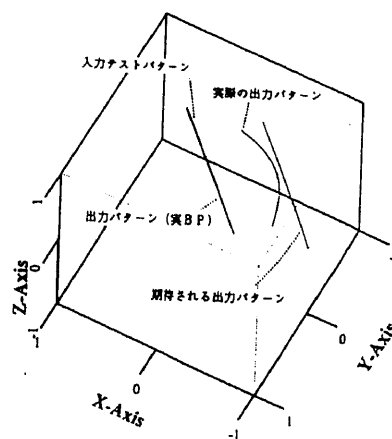
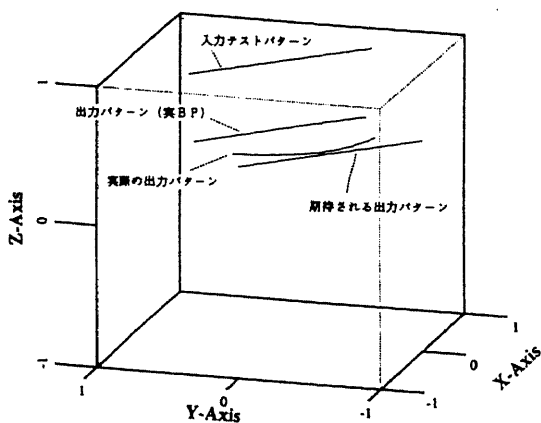


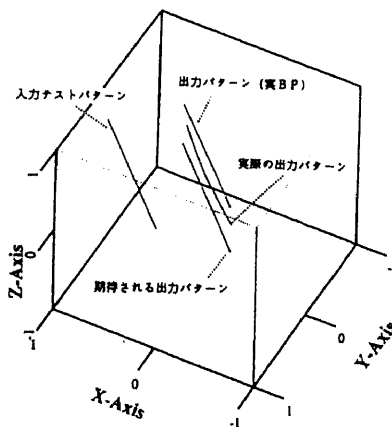
図7.6 学習パターン(平行移動)



(a) テストパターン1(平行移動)



(b) テストパターン2(平行移動)



(c) テストパターン3(平行移動)

図7.7 テストパターン(平行移動)

7.3.4 非線形変換

7.3.1～7.3.3項においては、単純な3次元アフィン変換学習能力について示した。たとえば、“原点との距離に関する2分の1の縮小”などのように単独の縮小率を学習させ、そのふるまいについて調べた。本項では、やや複雑な3次元アフィン変換を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。

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、2分の1、10分の1に縮小させるパターンを学習させた(図7.8)。学習パターンは21個であり、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (-1.0 \leq t \leq 1.0) \quad (7.24)$$

上の21個の点(点列の間隔は $\sqrt{3}/10$)を入力パターンとした。直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (0.0 \leq t \leq 1.0) \quad (7.25)$$

上の11個の点に対しては、それらの点と原点との距離を2分の1にした点を出力パターンとし、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (-1.0 \leq t \leq 0.0) \quad (7.26)$$

上の11個の点に対しては、それらの点と原点との距離を10分の1にした点を出力パターンとした。ここで、前者の学習パターンを学習パターン1、後者の学習パターンを学習パターン2と呼ぶことにする。出力学習パターン1は、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (0.0 \leq t \leq 0.5) \quad (7.27)$$

上に乗っており、出力学習パターン2は、直線

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (-0.1 \leq t \leq 0.0) \quad (7.28)$$

上に乗っていることになる。

学習終了後、入力テストパターンとして、3つの円 $x^2+z^2=1$ 、 $y^2+z^2=1$ 、 $x^2+y^2=1$ の円周上の60個の点を与えたところ、ニューラルネットワークは図7.9に示すようなパター

ンを出力した。図7.9において、入力学習パターン1に近い入力テストパターンほど2分の1に縮小され、入力学習パターン2に近い入力テストパターンほど10分の1に縮小されるという傾向性が見られる。つまり、0.1と0.5という2つの縮小率が反映された結果となっているようである。

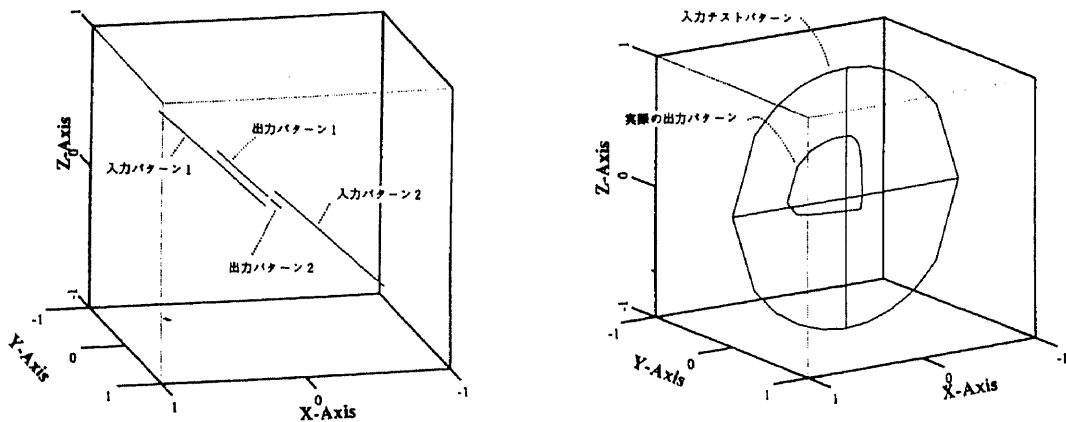
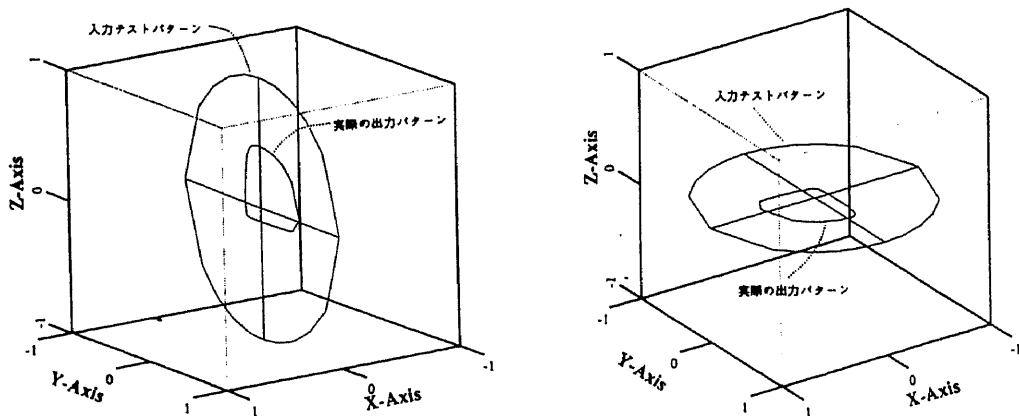


図7.8 学習パターン(やや複雑な変換) (a) テストパターン1(やや複雑な変換)



(b) テストパターン2(やや複雑な変換) (c) テストパターン3(やや複雑な変換)

図7.9 テストパターン(やや複雑な変換)

7.4 コンピュータビジョンへの応用

3次元ベクトルBPのコンピュータビジョンへの応用について述べる。

5.6節では、複素BPの応用として、単眼視(1つのカメラ)によりオプティカル・フローから動物体の(画像平面上の)動きを解釈する方法について述べた。本節では、3次元ベクトルBPの応用として、5.6節で述べた方法の肩の上に乗る、3眼視(3つのカメラ)による動物体の(3次元空間内の)動きの解釈を行なう方法について述べる。本節では、5.6節で述べた単眼視により得られる「画像平面上の動きを規定するパラメータ」を 2次元動きパラメータ

と呼ぶことにする。そして、この2次元動きパラメータを出力する複素BPネットワーク(図5.12)を2次元動き解釈ネットワークと呼ぶことにする。

図7.10に3次元ベクトルBPを利用した3次元空間内の動きの解釈を行なう処理系の構成例を示す。まず、各カメラ画像のオプティカル・フローをそれぞれ2次元動き解釈ネットワークを通して解釈を行ない、2次元動きパラメータを出力する。その結果を3次元ベクトルBPネットワーク(3次元動き解釈ネットワークと呼ぶ)に入力し、その出力を3次元空間内の動きを表すパラメータ(3次元動きパラメータと呼ぶ)の解釈値とする。ここで、3次元動き解釈ネットワークには、1つの並進運動や1つの回転といった基本的な3次元動きに関するパターンを教師データとして予め学習させておく。基本的な3次元動きを学習した3次元動き解釈ネットワークは、学習していない3次元動きに対してもうまく解釈する能力を示すことが期待される。図7.11に示すように、3次元動き解釈ネットワークへの入力となるのは、各カメラ視点ごとに得られる2次元動きパラメータである。つまり、 $\{(dx_2, dy_2, dz_2)\}$ および $(\omega_{z_2}(1), \omega_{z_2}(2), \omega_{z_2}(3))$ といった4つの3次元ベクトルである。ここで、 dx_2 は x 軸方向の並進成分、 dy_2 は y 軸方向の並進成分、 dz_2 は拡大・縮小成分、そして $\omega_{z_2}(i)$ はカメラ i により得られた回転成分である ($i = 1, 2, 3$)。いわば、 $\{(dx_2, dy_2, dz_2)\}$ は並進成分、 $(\omega_{z_2}(1), \omega_{z_2}(2), \omega_{z_2}(3))$ は回転成分である。そして、3次元動き解釈ネットワークは、3次元の動きを表す3次元動きパラメータ、すなわち、対象物体の3次元空間における x, y, z 軸方向への並進成分 (dx, dy, dz) と各軸毎の回転成分 $(\omega_x, \omega_y, \omega_z)$ の2つの3次元ベクトルを出力する。このように、3次元ベクトルBPネットワークは3次元ベクトルから構成される入出力パターンを想定しているので、3次元情報を自然に扱うことができる。

3組のカメラ画像の動きから連立方程式を解くことにより、3次元の動きを計算する一般的な手法の場合、計算量が多く非効率的であるのに対し、3次元ベクトルBPネットワークを使用するこの手法は、学習後の処理に要する時間が短く、しかも雑音に強いと考えられる。

宮内らはこの考え方に沿って3次元ベクトルBPネットワークに3次元の動きの解釈を行なわせ、ノイズの影響にも強いことを含めて、その有効性の確認を行なった[58, 59, 62]。

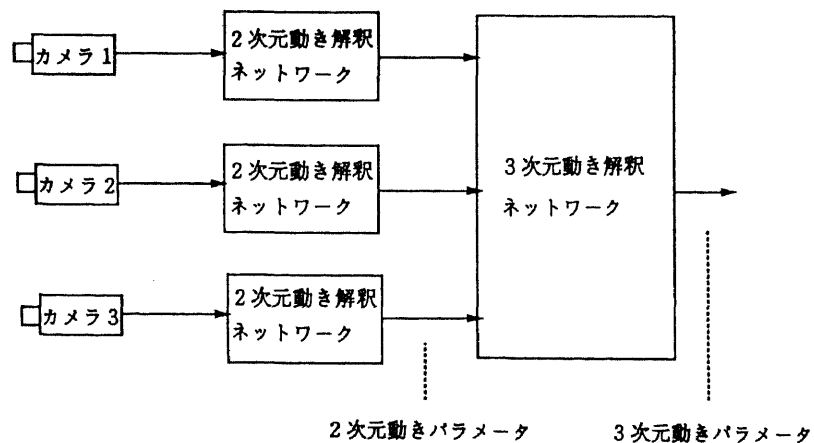


図7.10 3次元動き解釈処理系の構成例

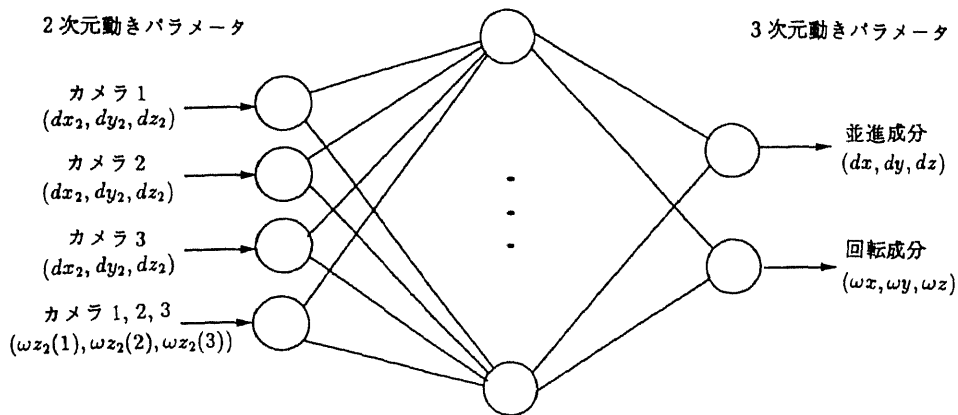


図7.11 3次元動き解釈ネットワークの構成例

7.5 むすび

バックプロパゲーション学習アルゴリズムの3次元化パラメータ版である3次元ベクトル・バックプロパゲーション学習アルゴリズムを提案した。3次元ベクトルBPは入力信号、閾値および出力信号がすべて3次元実ベクトルであり、重みがすべて3次直交行列である階層型ニューラルネットワークに適用される。この新しいアルゴリズムは第2章において提案した複素BPの自然な拡張になっており、その拡張は群論に基づいて行なわれた。さらに、複素BPには2次元アフィン変換学習能力が備わっていたように、3次元ベクトルBPには、従来の階層型ニューラルネットワークには見られない3次元アフィン変換学習能力が備わっていることが示唆された。最後に、応用事例としてコンピュータビジョンへの応用を取り上げた。

3次元ベクトル・ニューロンの入力、出力は3次元ベクトルであることが要請されるため、本章で提案したアルゴリズムが自然な形で扱うことのできる問題は、入出力空間が $R^3 \times \dots \times R^3$ であるようなクラスの問題に制限される。つまり、階層型ニューラルネットワークのパラメータを3次元化することにより、3次元アフィン変換学習能力という機能が生じたが、一方では、入出力空間が R^n であるような一般的な問題を素直に扱うことはできないという制限が生じたわけである。

第 8 章

外積に基づいた 3 次元化パラメータ・バックプロパゲーション 学習アルゴリズム

8.1 はじめに

第 7 章では、複素 BP の自然な拡張としての 3 次元ベクトル BP を提案した。本章では第 7 章とは異なった観点から実 BP の 3 次元パラメータ化を行なう ([37, 39, 40])。すなわち、外積 を拡張の基軸とする。

まず、次のような階層型ニューラルネットワークを定式化する。重み、閾値および入出力信号がすべて 3 次元実ベクトルであり、各ニューロンにおける計算が 3 次元外積を使って行なわれるというような階層型ニューラルネットワークである。3 次元ベクトル・ニューラルネットワーク (7.2.2 項) と異なるのは、重みが 3 次直交行列から 3 次元実ベクトルになっており、その影響により、3 次元ベクトル・ニューラルネットワークにおいて行なわれていた行列演算が外積演算になっているという点である。次に、このような階層型ニューラルネットワークに適用されるバックプロパゲーション学習アルゴリズム (以下、外積 BP と呼ぶ) を導出する。外積 BP は 3 次元ベクトル BP (第 7 章) と同様、実 BP の 3 次元化パラメータ版の一種である。最後に外積 BP の性能を調べるために、簡単な計算機実験を行ない、実 BP に比べて学習速度および計算量において優れていることを確認する。

8.2 節では、外積 BP を定式化し、8.3 節で外積 BP の学習特性に関する実験を行なう。8.4 節で結論を述べる。

8.2 外積 BP

本節では、外積 BP モデルを定式化し、外積 BP 学習アルゴリズムの導出を行う。

8.2.1 外積ニューロン

第 7 章でも述べたように、実 BP あるいは複素 BP をさらに高次元パラメータ化する方法にはいくつか考えられる。ひとつは、実数 α (1 次元) を複素数 $z = x + iy$ (2 次元, 第 2 章), 4 元数 $q = a + ib + jc + kd$ (4 次元), 8 元数 (8 次元), 16 元数 (16 次元), ... へと拡張

してゆく方法である。もうひとつは、重みおよび閾値の次元数を1次元から n 次元実ベクトルを使って高次元パラメータ化する方法である。後者の方法にはさらに次の2つの方法が考えられる：(1) 重みを n 次行列(第7章)とする方法、(2) 重みを n 次元実ベクトルとする方法。本章では、(2)の方法を使って3次元パラメータ化を試みる。その際、3次元実ベクトルと3次元実ベクトルとから3次元実ベクトルを生成することが必要となる。そのような演算を自由に創ることもできるが、ここでは、数学において標準的に使用されている外積を採用する。

次のようなニューロン(外積ニューロンと呼ぶ)を考える。入力信号、重み、閾値および出力信号はすべて3次元実ベクトルである。ニューロン j の内部ポテンシャル A_j は、実BPにおいて用いられるニューロンの内部ポテンシャルの類推で、

$$A_j = \sum_k (W_{jk} \times S_k) + T_j \quad (8.1)$$

と定義する。ここで、 W_{jk} はニューロン j とニューロン k との間の重みを表す3次元実ベクトル、 S_k はニューロン k から出力されたニューロン j への入力信号を表す3次元実ベクトル、 T_j はニューロン j の閾値を表す3次元実ベクトルである。また、 $\mathbf{x} \times \mathbf{y}$ は $\mathbf{x} = {}^t[x_1 \ x_2 \ x_3]$ と $\mathbf{y} = {}^t[y_1 \ y_2 \ y_3]$ の外積を表している。すなわち、 $\mathbf{x} \times \mathbf{y} = {}^t[x_2y_3 - x_3y_2 \ x_3y_1 - x_1y_3 \ x_1y_2 - x_2y_1]$ である。出力信号 $F_3(A_j)$ は次のように定義される：

$$F_3(A_j) = \begin{bmatrix} f_R(a_1) \\ f_R(a_2) \\ f_R(a_3) \end{bmatrix}, \quad \text{ただし, } A_j = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \quad (8.2)$$

$F_3(A_j)$ の各成分は、それぞれ、 A_j のそれぞれの各成分 a_m ($m = 1, 2, 3$)に関するシグモイド関数値である。この出力関数 F_3 は式(7.2)において定義したものと同一のものである。

8.2.2 外積ニューラルネットワーク

本項では、外積BPが適用されるネットワークを、8.2.1項で定義した外積ニューロンを使って構成する。一般に、外積ニューロンから構成されるニューラルネットワークを外積ニューラルネットワークと呼ぶことにするが、ここで定義するネットワークは、外積BPが適用されるネットワークという意味で外積BPネットワークと呼ぶことにする。簡単のために、3層ネットワークを例にとる。

入力ニューロン i と中間ニューロン j との間の重みを $W_{ji} = {}^t[w_{ji}^x \ w_{ji}^y \ w_{ji}^z] \in \mathbf{R}^3$ 、中間ニューロン j と出力ニューロン k との間の重みを $V_{kj} = {}^t[v_{kj}^x \ v_{kj}^y \ v_{kj}^z] \in \mathbf{R}^3$ 、中間ニューロン j の閾値を $\Theta_j = {}^t[\theta_j^x \ \theta_j^y \ \theta_j^z] \in \mathbf{R}^3$ 、出力ニューロン k の閾値を $\Gamma_k = {}^t[\gamma_k^x \ \gamma_k^y \ \gamma_k^z] \in \mathbf{R}^3$ とする。また、入力ニューロン i への入力信号を $I_i = {}^t[I_i^x \ I_i^y \ I_i^z] \in \mathbf{R}^3$ とし、中間ニューロン j および出力ニューロン k の出力信号を、それぞれ、 $H_j = {}^t[H_j^x \ H_j^y \ H_j^z] \in \mathbf{R}^3$ 、 $O_k = {}^t[O_k^x \ O_k^y \ O_k^z] \in \mathbf{R}^3$ とする。さらに、 $\Delta^k = {}^t[\delta_k^x \ \delta_k^y \ \delta_k^z] = T_k - O_k \in \mathbf{R}^3$ を O_k と出力ニューロン k に対する教師パターン(出力学習パターン) $T_k = {}^t[T_k^x \ T_k^y \ T_k^z] \in \mathbf{R}^3$ との

間の誤差とする。パターン p に対する 2 乗誤差を $E_p = (1/2) \sum_{k=1}^N \|T_k - O_k\|^2$, と定義する。ただし, $\|\mathbf{x}\| \stackrel{\text{def}}{=} \sqrt{x_1^2 + x_2^2 + x_3^2}$, $\mathbf{x} = {}^t[x_1 \ x_2 \ x_3]$, N は出力ニューロンの総数である。

8.2.3 学習アルゴリズム

次に, 前項までに定義した外積BPモデルに対する学習アルゴリズムを導出する。十分小さい学習率 $\varepsilon > 0$ に対して, 最急降下法を使うことにより, 学習パラメータは次の式にしたがって修正すべきであることがわかる。

$$\begin{aligned} \Delta V_{kj} \stackrel{\text{def}}{=} \begin{bmatrix} \Delta v_{kj}^x \\ \Delta v_{kj}^y \\ \Delta v_{kj}^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial v_{kj}^x} \\ \frac{\partial E_p}{\partial v_{kj}^y} \\ \frac{\partial E_p}{\partial v_{kj}^z} \end{bmatrix}, & \Delta \Gamma_k \stackrel{\text{def}}{=} \begin{bmatrix} \Delta \gamma_k^x \\ \Delta \gamma_k^y \\ \Delta \gamma_k^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial \gamma_k^x} \\ \frac{\partial E_p}{\partial \gamma_k^y} \\ \frac{\partial E_p}{\partial \gamma_k^z} \end{bmatrix}, \\ \Delta W_{ji} \stackrel{\text{def}}{=} \begin{bmatrix} \Delta w_{ji}^x \\ \Delta w_{ji}^y \\ \Delta w_{ji}^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial w_{ji}^x} \\ \frac{\partial E_p}{\partial w_{ji}^y} \\ \frac{\partial E_p}{\partial w_{ji}^z} \end{bmatrix}, & \Delta \Theta_j \stackrel{\text{def}}{=} \begin{bmatrix} \Delta \theta_j^x \\ \Delta \theta_j^y \\ \Delta \theta_j^z \end{bmatrix} &= -\varepsilon \begin{bmatrix} \frac{\partial E_p}{\partial \theta_j^x} \\ \frac{\partial E_p}{\partial \theta_j^y} \\ \frac{\partial E_p}{\partial \theta_j^z} \end{bmatrix}, \end{aligned} \quad (8.3)$$

ここで, $\Delta \mathbf{x}$ は学習パラメータ \mathbf{x} の修正量を表す。式(8.3)を計算すると, 結局,

$$\Delta V_{kj} = H_j \times \Delta \Gamma_k, \quad (8.4)$$

$$\Delta \Gamma_k = \varepsilon D_k \Delta_k, \quad (8.5)$$

$$\Delta W_{ji} = I_i \times \Delta \Theta_j, \quad (8.6)$$

$$\Delta \Theta_j = D_j \sum_k (\Delta \Gamma_k \times V_{kj}), \quad (8.7)$$

ただし,

$$D_k = \begin{bmatrix} (1 - O_k^x)O_k^x & 0 & 0 \\ 0 & (1 - O_k^y)O_k^y & 0 \\ 0 & 0 & (1 - O_k^z)O_k^z \end{bmatrix}, \quad (8.8)$$

$$D_j = \begin{bmatrix} (1 - H_j^x)H_j^x & 0 & 0 \\ 0 & (1 - H_j^y)H_j^y & 0 \\ 0 & 0 & (1 - H_j^z)H_j^z \end{bmatrix} \quad (8.9)$$

が得られる。

8.3 学習特性

本節では外積BPの学習特性について調べる。簡単な学習パターンを使って実BPとの比較実験を行なった。学習パターンは表8.1に示すものを用いた。また, ネットワークは,

1-3-1 外積BPネットワーク, 3-7-3 実BPネットワークを用いた。表 8.2 はこれらのネットワークの学習 1 回の時間計算量がほぼ等しいものであることを示している。ここで, 4.4 節と同様に, 1 回の学習で行なわれる (実数の) 四則演算の総回数を時間計算量, 学習パラメータ数を領域計算量としている。ただし, 時間計算量において, 加算, 減算, 乗算および除算のそれぞれの 1 回の計算量はすべて等しいものと仮定している。また, 学習パターンは, 通常, 数種類のパターンから構成されているが, その数種類のパターンをひと通り提示することで 1 回の学習と数えることとしている。学習率は 0.5 とし, 学習パラメータの初期値は -0.3 から $+0.3$ までの間の乱数により設定した。

実験結果を図 8.1 に示す。縦軸は誤差 $\sqrt{2\sum_p E_p}$ である (E_p は 8.2.2 節で定義したパターン p に対する 2 乗誤差である)。外積BP はおよそ 400 回で収束したのに対し, 実BP は 800 回の学習を要した。さらに, 外積BP ネットワークの領域計算量は実BP ネットワークのものに比べて, 約 3 分の 2 程度になっている (表 8.2)。

表 8.1 実験に用いる学習パターン. x, y, z は外積BP ネットワークにおいてはそれぞれ入出力ニューロン 1 の X 成分, Y 成分, Z 成分に与え, 実BP ネットワークにおいてはそれぞれ入出力ニューロン 1, 2, 3 に与える。

| パターン番号 | 入力パターン | | | 出力パターン | | |
|--------|--------|-----|-----|--------|-----|-----|
| | x | y | z | x | y | z |
| 1 | -2 | 0 | 2 | 1 | 0 | 0 |
| 2 | 1 | -1 | 0 | 1 | 0 | 1 |
| 3 | -1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 2 | 1 | -1 | 0 | 1 | 0 |

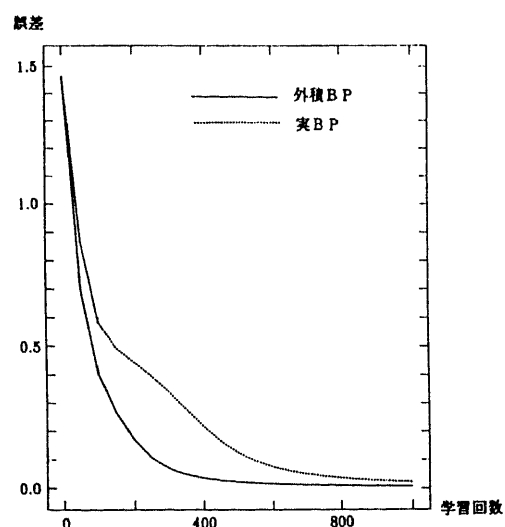


図 8.1 学習曲線 (外積BP と実BP の比較)

表 8.2 外積BPおよび実BPにおける計算量. 時間計算量は1回の学習において行なわれる(実数の)四則演算の総数. 領域計算量は学習パラメータ数(重みおよび閾値の総数).

| ネットワーク | 時間計算量 | | | 領域計算量 | | |
|------------|-------|-----|-----|-------|----|----|
| | 乗除算 | 加減算 | 合計 | 重み | 閾値 | 合計 |
| 外積BP 1-3-1 | 117 | 78 | 195 | 18 | 12 | 30 |
| 実BP 3-7-3 | 128 | 69 | 197 | 42 | 10 | 52 |

8.4 むすび

バックプロパゲーション学習アルゴリズム(BP)の3次元化パラメータ版である外積バックプロパゲーション学習アルゴリズム(外積BP)を提案した。外積BPは入力信号、重み、閾値および出力信号がすべて3次元実ベクトルである階層型ニューラルネットワークに適用される。第7章において提案した3次元ベクトルBPと異なる点は、外積を基軸としてパラメータを3次元に拡張したという点である。さらに、外積BPの学習特性について調べ、実BPに比べて優れているとの感触が得られた。しかし、明確な結論を得るためには、より体系的な実験を要する。また、複素BPには2次元アフィン変換学習能力が、3次元ベクトルBPには3次元アフィン変換学習能力がそれぞれ備わっていたように、外積BPにも何らかの固有の性質が備わっていることが予想される。

第 9 章

結論

階層型ニューラルネットワークのパラメータを高次元に拡張した高次元化パラメータ・階層型ニューラルネットワークを提案し、その固有の特徴、性質を解析的、実験的に調べた。その結果を要約すれば次のようになる。

1. Widrow ら [60], Kim ら [13] とは異なった方法で、従来の階層型ニューラルネットワークの結合の重みと各ニューロンが持つ閾値を複素数に拡張し、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 (複素BP) を提案した。学習識別理論 [4] を複素数に拡張することにより、学習収束性の保証を得た。複素BPは、既存の複素化パラメータを持つ階層型ニューラルネットワーク [60, 13] が持っていた収束性に関する問題点を解消している。
2. 複素BPの基本特性を、ネットワークアーキテクチャの観点から解析的、実験的に調べ、実BPとの差異を明らかにした。複素BPには、複素数への拡張の結果として、次のような性質が生じている。複素BPネットワークにおける重みパラメータは互いに独立ではなく2次元アフィン変換に基づいた制約を有しており、学習は基本的にその制約のもとで進められる。複素ニューロンの決定表面は2つの直交した超曲面から構成されており、決定領域を均等に4つに分割しようとする。一般に、これは $0, 1, i, 1+i$ という4つのクラスを扱う複素パターンに適した性質である。3層の複素BPネットワークにおける決定表面は基本的にこの性質を内包しており、中間ニューロンへの総入力十分大きいときに直交する。
3. 複素BPの基本特性に対して、学習アルゴリズムの側面からアプローチし、次の結果を得た。複素数への拡張は複素BPに次のような性質をもたらしている。複素BPにおける誤差逆伝播は2次元アフィン変換に関係した構造を有している。複素BPは、ネットワークを流れる複素信号を1つの処理単位として学習を進める。複素BPの学習則には、学習停滞状態の発生を抑えるメカニズム (実部要因と虚部要因の補完機構) が含まれている。これにより、複素パターンに対する学習が実BPに比べて数倍速くなっている。しかも、そのときに必要となる領域計算量は約半分済む。

4. 複素BPには2次元アフィン変換学習能力(回転, 相似変換, 平行移動などを学習する能力)が備わっていることを確認した。これは実BPには見られない複素BPが持つ顕著な性質である。また, この2次元アフィン変換学習能力と複素解析における一致の定理との関係を考察した。一致の定理は複素解析においてのみ見られる固有の性質である。さらに, 「学習させたアフィン変換」と「学習パラメータの値」との間には明確な関係があることを明らかにした。また, コンピュータビジョンへの応用の可能性を明らかにした。
5. 複素BPの2次元アフィン変換学習能力の数学的解析を行ない, その定性的性質を明らかにした。得られた主要な結果は次のとおりである。(1) 1種類の変換率(たとえば, 1種類の回転角度)を学習した複素BPネットワークのふるまいを明らかにした。変換率に関する汎化能力の(距離に関する)誤差は, 未学習パターンと学習パターンの偏角の差に関する正弦関数によって表現される。(2) 2種類の変換率を学習した複素BPネットワークのふるまいを回転角度を例にとりて解析し, 次の結果を得た。(a) 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の(距離に関する)誤差の評価式を与えた。(b) 2種類の回転角度を学習した複素BPネットワークの回転角度に関する汎化能力の『(角度に関する)誤差が無視できるほど小さな値であるための十分条件』を求めた。更に, これらの数値例を示し, 解析結果の妥当性を確認した。
6. バックプロパゲーション学習アルゴリズムの3次元化パラメータ版である3次元ベクトル・バックプロパゲーション学習アルゴリズム(3次元ベクトルBP)を提案した。3次元ベクトルBPは入力信号, 閾値および出力信号がすべて3次元実ベクトルであり, 重みがすべて3次元直交行列である階層型ニューラルネットワークに適用される。このアルゴリズムは複素BPの自然な拡張になっており, その拡張は群論に基づいて行なわれた。さらに, 複素BPには2次元アフィン変換学習能力が備わっていたように, 3次元ベクトルBPには, 従来のニューラルネットワークには見られない3次元アフィン変換学習能力が備わっていることが示唆された。最後に, コンピュータビジョンへの応用について述べ, 3次元ベクトルBPの有用性を示した。
7. 3次元ベクトルBPとは異なった観点からバックプロパゲーション学習アルゴリズムを3次元パラメータ化した外積BPを提案した。外積BPが適用されるのは入出力信号, 重みおよび閾値がすべて3次元実ベクトルであり, 各ニューロンにおいて外積計算が行なわれるような階層型ニューラルネットワークである。3次元ベクトルから成るパターンを使って簡単なシミュレーションを行ない, 外積BPの性能を調べた。その結果, 実BPに比べて学習速度, 計算量の点において優れているとの感触を得た。外積ニューロンへの入力には3次元ベクトルであることが要請されるため, 外積BPが自然な形で扱うことのできる問題は, 入出力空間が $\mathbf{R}^3 \times \dots \times \mathbf{R}^3$ であるようなクラスの問題に制限される。複素BPおよび3次元ベクトルBPについても同様のことが言える。

以上の結果から、階層型ニューラルネットワークのパラメータを高次元化し、複素数(2次元情報)あるいは3次元情報などの高次元情報を自然に表現することができ、素直に処理することのできる階層型ニューラルネットワークを提供し、その特性を明らかにするという本論文の目的は、ある程度達成されたものと考えられる。

最後に今後の課題について述べる。まず、複素BPに関してであるが、2次元への拡張方法としては複素数が最も標準的であると考えられる。しかし、複素数にとらわれなければ、2次元化への方法はいろいろ考えられる。また、ニューロンが持つ出力関数にも様々な種類が考えられ、それらに応じてそれら固有の性質が生ずるのであろうと推定される。特に、シグモイド関数の代わりに-1から1の間の値をとる双曲線正接関数(hyperbolic tangent) $\tanh x$ をベースにした複素関数を使うと、2次元アフィン変換を学習する際に出力パターンを「-1から1の間の値」から「0から1の間の値」に変換する必要がなく、素直に処理することができるようになる。複素BPの特性に関して言えば、複素BPネットワークによる複素関数の近似能力の解析、4層以上のネットワークの決定表面の構造、学習速度の高速化、2次元アフィン変換学習能力と一致の定理との関係に関する理論的考察、3種類以上の回転角度を学習した複素BPネットワークの性質の探究、相似変換および平行移動に関する学習能力の数学的解析、「学習させたアフィン変換」と「学習パラメータの値」との関係の体系的調査、ニューロン数が非常に多い場合の特性の考察などが考えられる。また、本論文では決定表面、学習特性および2次元アフィン変換学習能力を個々に調べたわけであるが、これらの相互間の関係について考察することも興味深い問題である。たとえば、決定表面の直交性が学習特性にどのような影響を与えているかという問題である。つまり、3層ネットワークの場合、決定表面がかならずしも直交するとは限らなかったが、直交する場合としない場合とで学習特性がどのように変わってくるかという問題である。応用に関して言えば、文献[17, 18, 19]において行なわれたような画像処理への応用が既に存在するが、画像処理に限らず信号処理、制御工学、音声認識などへの本格的な応用を通じた複素BPの評価を進めてゆく予定である。

次に3次元ベクトルBPについては、上記で述べた複素BPの今後の課題に対応した項目がすべてそのまま今後の課題の対象となる。複素BPに備わっていた基本特性が3次元ベクトルBPにどのような形で現れてくるか、興味のあるところである。特に、3次元アフィン変換学習能力については3種類以上の変換率を学習したネットワークのふるまいを調べてゆきたい。また、外積BPの基本特性について言えば、研究の端緒についたばかりであり、より体系的なアプローチが必須である。複素BP、3次元ベクトルBPにおける2次元アフィン変換学習能力、3次元アフィン変換学習能力のように、外積BPに固有の性質が出てくることを期待している。また、3次元ベクトルBPあるいは外積BPに3次元空間における曲線の曲率を学習させたり、Poggioらが行なったGeneralized Radial Basis Functions Networkによる3次元空間内の曲がりくねった針金の認識実験[51]を行なわせるなどの応用も行なってゆく予定である。

さらに階層型ニューラルネットワークのパラメータを4次元以上に高次元化した場合に、どのような特性が現れてくるのかについても引続き興味のあるところである。

謝辞

本論文は、通商産業省工業技術院電子技術総合研究所に於いて筆者が行なってきた階層型ニューラルネットワークのパラメータの高次元化に関する研究をまとめたものです。

本研究は電子技術総合研究所の多くの方々の支援のおかげで行なうことができました。特に、棟上昭男元情報アーキテクチャ部長(現情報処理振興事業協会理事)、弓場敏嗣前情報アーキテクチャ部長(現電気通信大学大学院教授)、太田公広情報アーキテクチャ部長には、終始多大のご支援とご理解を頂きました。ここに心からの謝意を表します。特に、弓場教授には本論文をまとめる契機を与えて下さり、太田部長には心構えをご教示いただきました。そのご支援とご指導に深く感謝いたします。

また、古谷立美前計算機構研究室長(現東邦大学理学部教授)をはじめ、樋口哲也計算機構研究室長、大東栄夫氏、田中敏雄氏、秋山泰氏(現京都大学化学研究所助教授)、丹羽竜哉氏(現分散システム研究室)、稲吉宏明氏の計算機構研究室の諸氏、赤穂昭太郎氏(情報数理研究室)には、熱心かつ有益なご討論ならびにご協力を頂きました。ここに記して感謝いたします。

第2章の複素BP、第4章の複素BPの学習特性、第5章の複素BPの2次元アフィン変換学習能力は古谷教授との共同研究をまとめたものです。第3章の複素BPネットワークの決定表面の構造は、古谷教授、秋山助教授、赤穂氏との共同研究をまとめたものです。以上の方々に心から感謝いたします。

また、坂根茂幸自律システム研究室長、比留川博久氏(行動知能研究室)には、ロボット分野において4元数が有効に応用されている事例に関する適切な文献を心良くご教示頂きました。厚くお礼申し上げます。

最後に、本論文をまとめるにあたっては、筑波大学電子・情報工学系平井有三教授に適切なご助言とご指導をいただきました。また、筑波大学電子・情報工学系板橋秀一教授、同西原清一教授、同大田友一教授、筑波大学連携大学院併任教授としての大津展之知能情報部長には、本論文の完成に有益なご教示、ご助言をいただきました。ここに、深く感謝の意を表します。

参考文献

- [1] Adiv, G. : “Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects”, IEEE Trans. Pattern Anal. & Machine Intell., PAMI-7 (4), pp.384-401 (1985).
- [2] Aggarwal, J. K. and Nandhakumar, N. : “On the Computation of Motion from Sequences of Images - A Review”, Proceedings of IEEE, 76, 8, pp.917-935 (1988).
- [3] 秋山 泰, 古谷 立美 : “損失関数にエントロピー項を導入したバックプロパゲーション学習則”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC91-6, pp.39-46 (1991).
- [4] Amari, S. : “A Theory of Adaptive Pattern Classifiers”, IEEE Trans. Electronic Computers, EC-16 (3), pp.299-307 (1967).
- [5] 麻生 英樹 : “ニューラルネットワーク情報処理”, 産業図書 (1988).
- [6] Ballard, D. H. and Brown, C. M. : “*Computer Vision*”, Rrentice-Hall, Inc. (1982).
- [7] Canny, J. F. : “*The Complexity of Robot Motion Planning*”, The MIT Press (1988).
- [8] エビングハウス他 (成木 勇夫訳) : “数 (下) ”, シュプリンガー・フェアラーク東京 (1993).
- [9] Hirose, A : “Proposal of Fully Complex-Valued Neural Networks”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN'92-Baltimore, June, Vol.4, pp.152-157 (1992).
- [10] Hirukawa, H., Papegay, Y. and Matsui, T. : “A Motion Planning Algorithm for Convex Polyhedra in Contact under Translation and Rotation”, Proceedings of IEEE International Conference on Robotics and Automation, Vol.4, pp.3020-3027 (1994).
- [11] Horn, B. K. P. : “*Robot Vision*”, The MIT Press (1986).
- [12] 笠原 乾吉 : “複素解析”, 実教出版 (1980).

- [13] Kim, M. S. and Guest, C. C. : “Modification of Backpropagation Networks for Complex-Valued Signal Processing in Frequency Domain”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN'90-SanDiego, June, Vol.3, pp.27-31 (1990).
- [14] 木ノ内 誠, 荻原 将文 : “自己フィードバックを有する複素バックプロパゲーション”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC94-40, pp.63-70 (1994).
- [15] Latombe, J. C. : “*Robot Motion Planning*”, Kluwer Academic Publishers (1991).
- [16] Lippmann, R. P. : “An Introduction to Computing with Neural Nets”, IEEE Acoustic, Speech and Signal Processing Magazine, April, pp.4-22 (1987).
- [17] Miyauchi, M. and Seki, M. : “Interpretation of Optical Flow through Neural Network Learning”, Proceedings of IEEE International Conference on Communication Systems/International Symposium on Information Theory and its Applications, Singapore, Nov., pp.1247-1251 (1992).
- [18] Miyauchi, M., Seki, M., Watanabe, A. and Miyauchi, A. : “Interpretation of Optical Flow through Neural Network Learning”, Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo, Dec., pp.523-528 (1992).
- [19] Miyauchi, M., Seki, M., Watanabe, A. and Miyauchi, A. : “Interpretation of Optical Flow through Complex Neural Network”, Proceedings of International Workshop on Artificial Neural Networks, IWANN'93-Barcelona, Lecture Notes in Computer Science, Vol.686, Springer-Verlag, pp.645-650 (1993).
- [20] 根本 幾, 河野 知志 : “神経回路網の複素数値化の試み”, 電子情報通信学会論文誌, Vol.J74-D-II, No.9, pp.1282-1288 (1991).
- [21] 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”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC90-60, pp.45-52 (1991).
- [22] 新田 徹, 赤穂 昭太郎, 秋山 泰, 古谷 立美 : “複素BPにおける決定表面の構造”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC91-62, pp.105-109 (1991).
- [23] 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 (分冊2), pp.145-146 (1991).
- [24] 新田 徹, 赤穂 昭太郎, 秋山 泰, 古谷 立美 : “複素BPの基本的性質”, 第43回情報処理学会 全国大会 論文集 (分冊2), pp.223-224 (1991).

- [25] 新田 徹, 古谷 立美 : “複素 BP の汎化能力”, 第 4 3 回情報処理学会 全国大会 論文集 (分冊 2), pp.225-226 (1991).
- [26] 新田 徹, 古谷 立美 : “複素バックプロパゲーション学習”, 情報処理学会論文誌, Vol.32, No.10, pp.1319-1329 (1991).
- [27] 新田 徹, 古谷 立美 : “複素 BP 学習の高速性”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC91-127, pp.215-221 (1992).
- [28] 新田 徹 : “複素 BP の回転角度に関する学習能力の誤差解析”, 電子情報通信学会 ニューロコンピューティング研究会 技術研究報告, NC92-34, pp.67-74 (1992).
- [29] 新田 徹 : “複素バックプロパゲーション学習アルゴリズムの非線形 2 次元回転学習能力”, 電子情報通信学会 秋季大会 講演論文集 (分冊 6), p.20 (1992).
- [30] 新田 徹, 赤穂 昭太郎, 秋山 泰, 古谷 立美 : “複素バックプロパゲーション・ネットワークにおける重みパラメータと決定表面の構造”, 情報処理学会論文誌, Vol.33, No.11, pp.1306-1313 (1992).
- [31] Nitta, T. and deGaris, H. : “A 3D Vector Version of the Back-Propagation Algorithm”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN'92-Beijing, Nov.3-6, Vol.2, pp.511-516 (1992).
- [32] 新田 徹, 古谷 立美 : “複素バックプロパゲーション学習アルゴリズムの学習特性”, 情報処理学会論文誌, Vol.34, No.1, pp.29-38 (1993).
- [33] 新田 徹 : “回転を学習した複素 BP ネットワークのふるまい”, 情報処理学会論文誌, Vol.34, No.1, pp.39-51 (1993).
- [34] Nitta, T. : “A Complex Numbered Version of the Back-Propagation Algorithm”, Proceedings of INNS World Congress on Neural Networks, WCNN'93-Portland, July 11-15, Vol.3, pp.576-579 (1993).
- [35] Nitta, T. : “A Three-Dimensional Back-Propagation”, Proceedings of INNS World Congress on Neural Networks, WCNN'93-Portland, July 11-15, Vol.3, pp.572-575 (1993).
- [36] Nitta, T. : “A Back-Propagation Algorithm for Complex Numbered Neural Networks”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN'93-Nagoya, Oct. 25-29, Vol.2, pp.1649-1652 (1993).
- [37] Nitta, T. : “A Back-Propagation Algorithm for Neural Networks Based on 3D Vector Product”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN'93-Nagoya, Oct. 25-29, Vol.1, pp.589-592 (1993).

- [38] Nitta, T. : “A Supervised Learning Algorithm for Neural Networks with Two-dimensional Weights”, Proceedings of CIE/IEEE International Conference on Neural Networks and Signal Processing, ICNNSP’93-Guangzhou, Nov. 2–5, pp.425–430 (1993).
- [39] Nitta, T. : “Proposal of Neural Networks Based on Vector Product”, Proceedings of CIE/IEEE International Conference on Neural Networks and Signal Processing, ICNNSP’93-Guangzhou, Nov. 2–5, pp.397–402 (1993).
- [40] Nitta, T. : “An Extension of the Back-Propagation Algorithm to Three Dimensions by Vector Product”, Proceedings of 5th IEEE International Conference on Tools with Artificial Intelligence, TAI’93-Boston, Nov. 8–11, pp.460–461 (1993).
- [41] Nitta, T. : “Fundamental Structures of the Complex Back-Propagation Algorithm”, Proceedings of International Symposium on Artificial Neural Networks, ISANN’93-Taiwan, Dec. 20–22, pp.30–38 (1993).
- [42] 新田 徹 : “ニューラルネットワークの3次元への拡張”, 情報処理学会論文誌, Vol.35, No.7, pp.1300–1310 (1994).
- [43] Nitta, T. : “Ability of the 3D Vector Version of the Back-Propagation to Learn 3D Motion”, Proceedings of INNS World Congress on Neural Networks, WCNN’94-SanDiego, June 4-9, Vol.3, pp.262–267 (1994).
- [44] Nitta, T. : “An Analysis on the Learning Rule in the Complex Back-Propagation Algorithm”, Proceedings of INNS World Congress on Neural Networks, WCNN’94-SanDiego, June 4-9, Vol.3, pp.702–707 (1994).
- [45] Nitta, T. : “Decision Boundaries of the Complex Valued Neural Networks”, Proceedings of INNS World Congress on Neural Networks, WCNN’94-SanDiego, June 4-9, Vol.4, pp.727–732 (1994).
- [46] Nitta, T. : “Behavior of the Complex Numbered Back-Propagation Network which has Learned Similar Transformation”, Proceedings of INNS World Congress on Neural Networks, WCNN’94-SanDiego, June 4-9, Vol.4, pp.765–770 (1994).
- [47] Nitta, T. : “Structure of Learning in the Complex Numbered Back-Propagation Network”, Proceedings of IEEE International Conference on Neural Networks, ICNN’94-Orlando, June 28-July 2, Vol.1, pp.269–274 (1994).
- [48] Nitta, T. : “An Analysis on Decision Boundaries in the Complex Back-Propagation Network”, Proceedings of IEEE International Conference on Neural Networks, ICNN’94-Orlando, June 28-July 2, Vol.2, pp.934–939 (1994).

- [49] Nitta, T. : “Generalization Ability of the Three-Dimensional Back-Propagation Network”, Proceedings of IEEE International Conference on Neural Networks, ICNN’94-Orlando, June 28-July 2, Vol.5, pp.2895–2900 (1994).
- [50] 野崎 昭弘 : “アルゴリズムと計算量”, 共立出版 (1987).
- [51] Poggio, T. and Edelman, S. : “A Network that Learns to Recognize Three-Dimensional Objects”, Nature, Vol.343:18, pp.263–266 (1990).
- [52] Rezgui, A. and Tepedelenlioglu, N. : “The Effect of the Slope of the Activation Function on the Back Propagation Algorithm”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN’90-Washington-DC, Jan., Vol.1, pp.707–710 (1990).
- [53] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. : “*Parallel Distributed Processing*”, Vol.1, The MIT press (1986).
- [54] 白井 良明編 : “パターン理解, 知識工学講座 9”, オーム社 (1987).
- [55] Subbarao, M. : “Interpretation of Image Flow: A Spatio-Temporal Approach”, IEEE Trans. Pattern Anal. & Machine Intell., PAMI-11 (3), pp.266–278 (1989).
- [56] 丹 康雄, 加藤 喜永, 江島 俊郎 : “誤差評価関数による PDP モデルの高速化”, 電子情報通信学会論文誌, Vol.J73-D-II, No.12, pp.2022–2028 (1990).
- [57] Tsai, R. Y. and Huang, T. S. : “Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces”, IEEE Trans. Pattern Anal. & Machine Intell., PAMI-6 (1), pp.13–27 (1984).
- [58] 渡部 朗, 宮内 新, 宮内 ミナミ : “ニューラルネットワークによるオプティカル・フローの 3 次元動き解釈のための一提案”, 第 4 6 回情報処理学会 全国大会 論文集 (分冊 2), pp.167–168 (1993).
- [59] 渡部 朗, 宮内 新, 宮内 ミナミ : “ニューラルネットワークを用いた物体の 3 次元動き解釈手法の研究”, 第 1 6 回情報理論とその応用シンポジウム (SITA’93) 予稿集 (2), pp.529–532 (1993).
- [60] Widrow, B., McCool, J. and Ball, M. : “The Complex LMS Algorithm”, Proceedings of the IEEE, Vol.63, No.4, pp.719–720 (1975).
- [61] Yamada, K., Kami, H., Tsukumo, J. and Temma, T. : “Handwritten Numeral Recognition by Multi-Layered Neural Network with Improved Learning Algorithm”, Proceedings of IEEE/INNS International Joint Conference on Neural Networks, IJCNN’89-Washington-DC, June, Vol.2, pp.259–266 (1989).

- [62] 矢沢 伸行, 全 へい東, 宮内 新, 宮内 ミナミ : “ニューラルネットワークを用いた3次元動き解釈システムのカメラ配置に関する検討”, 第49回情報処理学会全国大会 論文集 (分冊2), pp.129-130 (1994).