

# 港湾技研資料

TECHNICAL NOTE OF  
THE PORT AND HARBOUR RESEARCH INSTITUTE  
MINISTRY OF TRANSPORT, JAPAN

No. 883 Dec. 1997

港湾環境におけるニューラルネットワークの  
適用性に関する基礎的研究

三 井 正 雄  
橋 本 典 明  
中 川 康 之

運輸省港湾技術研究所



# 目 次

要 旨	4
1. 研究の背景と目的	5
2. ニューラルネットワークとは	5
3. 階層型ニューラルネットワークについて	5
3.1 バックプロパゲーション法	6
3.2 ネットワークパラメータについて	7
4. ニューラルネットワークによる分類処理	9
4.1 曖昧さを含むデータの処理とニューラルネットワークの有効性	9
4.2 非線形分類	10
4.3 XOR問題に見るネットワークの内部状況	10
5. ニューラルネットワークによるフィッティング	17
5.1 直線のフィッティング	17
5.2 曲線のフィッティング	19
5.3 オーバーフィッティング (過学習)	20
6. ニューラルネットワークによる時系列処理	23
6.1 時系列データを扱う際の工夫	23
6.2 フィードバック型ネットワークの構造	23
6.3 非線形時系列データのフィッティング	23
6.4 学習方法	24
6.5 入力データの特徴	24
6.6 予測結果	25
6.7 入力データセットの低減化に伴う予測結果	25
6.8 予測精度向上のための工夫	29
6.9 フィードフォワード型ネットワークとの比較	30
6.10 入力層ニューロン数の決定法	35
6.11 フラクタル次元	35
6.12 予測期間の違いによる出力特性	37
7. ニューラルネットワークの実データへの適用	39
7.1 資料	39
7.2 実験準備	39
7.3 入出力層ニューロン数の決定	40
7.4 学習方法	41
7.5 予測結果	41
7.6 入力層ニューロン数の妥当性	41
7.7 説明変量の追加による予測精度の向上	41

7.8	入力データに事前処理を施すことによる予測精度の向上	44
7.9	説明変量の寄与度	44
8.	おわりに	46
	参考文献	47
	補章A 焼き鈍し法について	48

# The Basic Research on the Potential Application of Neural Network to Environmental Problems Surrounding Harbors

Masao MITSUI \*

Noriaki HASHIMOTO \*\*

Yasuyuki NAKAGAWA \*\*\*

## Synopsis

Neural Networks is an information processing technique by which the signal transmission system of nerve cells in the brain is designed as an artificial model in order to imitate the excellent information processing performance of the human on a computer. As a general characteristic concerning Neural Networks, it is known that the Neural Network is effective to the information processing when there is some vagueness in the target data or when relation between cause and effect cannot be defined clearly.

This report is a basic study to apply Neural Network for various phenomena with complicated relations between each elements of the environment surrounding harbors, using effectively the information processing performance of it. This discussion involves the basic operational characteristics of Neural Networks, output characteristics when varying execution parameters, methods of presenting the study data to obtain highly accurate output, effectiveness of use as feedback type network, and the fractal dimension used as the method of deciding optimum forecast period when the time series data is targeted. Finally, the potential applications of Neural Network to data acquired through actual observations is examined based on results of these investigations, where time variation of chlorophyll-a is forecast with some kinds of water quality data.

**Key Words** : Neural Network, information processing technique, forecast, time series data

---

\*Trainee of Hydrodynamics Laboratory, Marine Environment Div.

(Member of Research & Development Div., Kaijo Corporation)

3-1-5, Sakae, Hamura, Tokyo 205, JAPAN Tel : +81-42-555-6405 Fax : +81-42-555-7176

e-mail : m.mitsui@kaijo.co.jp

\*\*Chief of Hydrodynamics Laboratory, Marine Environment Div. e-mail : hashimoto@cc.phri.go.jp

\*\*\*Member of Hydrodynamics Laboratory, Marine Environment Div. e-mail : nakagawa@cc.phri.go.jp

3-1-1, Nagase, Yokosuka, Kanagawa 239, JAPAN Tel : +81-468-44-5016 Fax : +81-468-44-1274

# 港湾環境におけるニューラルネットワークの 適用性に関する基礎的研究

三井正雄・  
橋本典明\*\*  
中川康之\*\*\*

## 要 旨

ニューラルネットワークとは、人間が有する優れた情報処理能力を計算機中に模倣するために、脳における神経細胞の信号伝達系を人工的なモデルとして考案された情報処理手法である。ニューラルネットワークに関する一般的な特徴としては、対象とするデータが曖昧さを含んでいる場合、あるいは原因と結果の因果関係が明瞭に定義できない場合の情報処理に有効であることが知られている。

本報告は、このニューラルネットワークの情報処理能力に着目して、要素間の因果関係を解明することが困難な港湾環境の複雑な諸現象に対し、これを今後有効的に活用することを目的とした基礎的な研究である。そのためまず始めに、ニューラルネットワークの基本的な動作特性として、実行時に必要なパラメータの違いによる出力特性、高精度な出力を得るための学習データの提示方法、時系列データを対象とする際のフィードバック型ネットワークの有効性、および最適な予測期間の決定方法として、フラクタル次元を利用することの有効性等について議論した。最後にこれらの検討結果を基に、実際に観測された複数の水質データからクロロフィル-a量の予測を行い、ニューラルネットワークの実データに対する適用可能性を検討した。

キーワード：ニューラルネットワーク、情報処理技術、予測、時系列データ

---

\*海洋環境部水理研究室研修生

(株式会社カイジョー研究開発本部)

〒205 東京都羽村市栄町3-1-5 Tel:(042)555-6405 Fax:(042)555-7176

e-mail:m.mitsui@kaijo.co.jp

\*\*海洋環境部水理研究室長 e-mail:hashimoto@cc.phri.go.jp

\*\*\*海洋環境部水理研究室 e-mail:nakagawa@cc.phri.go.jp

〒239 神奈川県横須賀市長瀬3-1-1 Tel:(0468)44-5016 Fax:(0468)44-1274

## 1. 研究の背景と目的

港湾を整備することによって当該海域の水理・環境特性が影響を受けることは、これまでの調査・研究でも指摘されているところである。これらの海域では、構造物や動・植物の生体系を含む種々の要素が複雑に相互作用しており、今日においても不明な部分が多く残されている。このような複雑な現象を解明し、予測し、制御することを目的に各地で種々の実験や観測が行われているが、観測して得られたデータが不完全で、例えば相互に矛盾するような情報を含む場合には、当該現象の誤った判断や解釈がなされる場合もあり得る。ニューラルネットワークは超並列・超分散を基礎とする情報処理技術で、我々人間が活動している現実の社会、即ち多量の情報が並列に存在し、しかも情報が不完全で互いに矛盾する要因を含むような社会において、人間の合理的判断を助ける有用な手法として期待されている技術である。

本研究は、複雑かつ複合的な港湾環境に関する諸現象の評価・予測・制御を最終的な目標として、要素間の因果関係が不明瞭な情報の処理に威力を発揮するニューラルネットワークの基本的な動作特性を検討し、応用上の技術的諸問題を克服しようとする基礎的研究である。

## 2. ニューラルネットワークとは

われわれ人間は、外界（環境）に存在する情報を予め全て知っているわけではなく、与えられた環境の中で生起する様々な状況に「適応」し、遭遇した事例を「学習」することにより、同様な未知なる状況の出現に対しても、「そこそこ」適切に対処可能な準備をしている。ニューラルネットワークとは、生体を持つこのような機能を計算機上に模したものである<sup>1)</sup>。

また、ニューラルネットワークの特徴としては、対象に関する厳密な構造の推定、あるいは解析を行うことなく、事例データを直接ネットワークに提示することにより、変量間の特性を学習可能であるという点である。そのため、特に入出力間の因果関係が不明確な（因果関係の解析が困難な）問題を処理する場合、ニューラルネットワークを利用することは極めて有効な手段である。

一般に、ニューラルネットワークはその学習方式の違いにより「教師あり学習」に代表される階層型（多層）ニューラルネットワークと「教師なし学習」に代表される相互結合型ニューラルネットワークに大別することができる。また、ここでいう「教師あり学習」とは、ある入力に対し、「教師信号」と呼ばれるネットワークの望ましい出力、即ち目標値がネットワークに与えられているもので、既に実用化されているニューラルネットワークの多くは、これに属するものである。

本報告は、この「教師あり学習」を利用した階層型ニューラルネットワークを対象とし、種々の検討を行うものである。

## 3. 階層型ニューラルネットワークについて

通常階層型ニューラルネットワークは、図3.1に示すように、ネットワークへの入力を受け持つ入力層、出力を担う出力層、そしてそれらの間に存在する中間層（隠れ層）により構成されている。このうち、中間層の層数が1層のものを3層ネットワークと呼び、階層型ネットワークの中で最も標準的なものとなっている。また、各層にそれぞれ配置されたニューロンは、結合荷重と呼ばれる線維によって他層のニューロンとつながっている。ここを流れる情報は、入力層から出力層に向かって一方方向に伝達され、各過程で以下のような処理が施される。

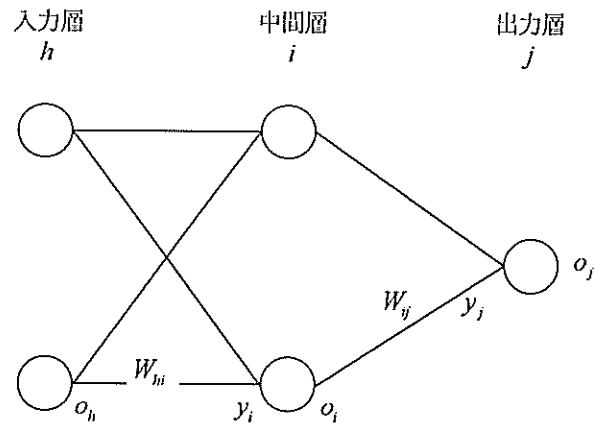


図3.1 階層型ニューラルネットワークの構造

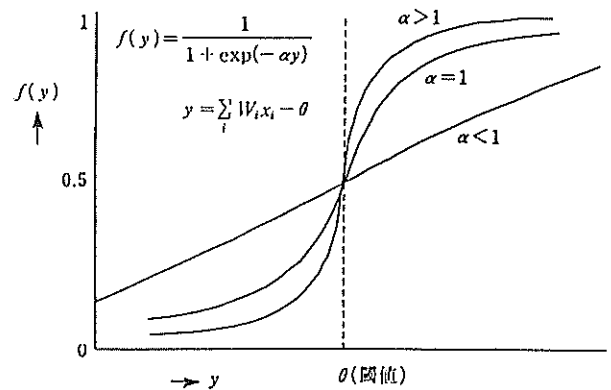


図3.2 シグモイド関数（文献2より抜粋）

入力層：

入力層のニューロンは入力データに対し特別な処理を施すことなく、単に入力データの窓口として、これを中間層へ伝達する。入力層から出力された情報は、入力層（第1層）のニューロン  $h$  と中間層（第2層）のニューロン  $i$  に結合する結合荷重の大きさ（線維の太さ）  $W_{hi}$  に比例して中間層へ伝達される。そのため入力層のニューロン  $h$  に入力されたデータを  $X_h$  とすると、中間層のニューロン  $i$  に伝達される情報はこれらの積である  $W_{hi}x_h$  として表現される。つまり、結合荷重  $W_{hi}$  は入力データに乗ずる重みの大きさを意味するものである。

中間層：

中間層のニューロンは、そこに入力される入力層からの出力と結合荷重との積の総和（ $\sum W_{hi}x_h$ ）がある閾値（ $\theta$ ）を越えたとき、1に近い値を出力し、そうでない場合は0に近い値を出力する。これをニューロンの動作関数と呼び、本報告では図3.2に示したシグモイド関数を採用する。

出力層：

出力層におけるニューロンの動作は、対象としている処理内容によって異なる。分類処理を行う場合は中間層ニューロンの動作と同様である。一方、関数のフィッティングを行う場合、出力は連続的な値が必要であるため、動作関数、即ちシグモイド関数を介することなく、中間層からの出力と結合荷重との積の総和をそのまま出力する。

階層型ニューラルネットワークの学習は、このようにして得られるネットワークの出力と、これに対応する教師信号との差がゼロになるように、結合荷重の修正を繰り返し行うもので、この学習アルゴリズムをバックプロパゲーション法（誤差逆伝搬法）と呼んでいる。

### 3.1 バックプロパゲーション法

ネットワークが望ましい出力を出すようにするためには、ネットワークの出力と教師信号との差、即ち誤差（ $E$ ）の値が小さくなる方向に結合荷重を変化させれば良い。つまり、結合荷重の変化に対する（ $E$ ）の変化が無くなるように結合荷重を修正する。

いま、ニューロン  $j$  への総入力  $y_j$  は、前層のニューロン  $i$  の出力  $o_i$  と、結合荷重  $W_{ij}$  との積の総和として式(3.1)で表される。また、ニューロン  $j$  の出力  $o_j$  は  $y_j$  に動作関数を施したもので式(3.2)により表される。

$$y_j = \sum_i W_{ij} \cdot o_i \quad (3.1)$$

$$o_j = f_j(y_j) \quad (3.2)$$

ここで動作関数  $f(y)$  は、その出力が0~1の範囲に規定されるシグモイド関数

$$f(y) = \frac{1.0}{1.0 + \exp(-ay)} \quad (3.3)$$

である。また、学習の状態を評価するための評価関数（誤差関数）を

$$E = \frac{1}{2} \sum_j (o_j - t_j)^2 \quad t_j : \text{教師信号} \quad (3.4)$$

とすると、学習の目的はこの  $E$  をゼロに近付ける、即ち、 $E$  が小さくなるように結合荷重  $W$  を変化させることである。そこで  $E$  を  $W$  で偏微分すると式(3.5)になる。またこれは、誤差が最も急な斜面の方向に進むことを意味するもので、一般に最急降下法と呼ばれている。

$$\Delta W_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial W_{ij}} \quad (3.5)$$

また、

$$\frac{\partial y_j}{\partial W_{ij}} = o_i \quad (3.6)$$

より、誤差の減少を図る場合、これに対応する結合荷重の変化量  $\Delta W_{ij}$  は、望ましい出力を得るために必要な入力の変化量に、前層の出力を乗じたものとして表され、

$$\frac{\partial E}{\partial y_j} = \delta_{ij} \quad (3.7)$$

と定義すると

$$\Delta W_{ij} = \delta_{ij} o_i \quad (3.8)$$

となる。ここで  $\delta_{ij}$  の意味するところは、誤差をゼロにするためには  $y_j$  をどれだけ動かさなければならないかである。しかしながら  $y_j$  はそもそも結合荷重  $W_{ij}$  と前層の出力  $o_i$  との積であるので、 $y_j$  を動かすためには必然的に  $W_{ij}$  を変化させることになる。即ち  $y_j$  を動かすためには  $\delta_{ij}$  を  $W_{ij}$  とみなして  $\delta_{ij} \times o_i$  としなければならない。これにより結合荷重  $W_{ij}$  の修正量  $\Delta W_{ij} = \delta_{ij} o_i$  が決定されたことになる。

次に、今決定された  $\Delta W_{ij}$  より、現在の  $W_{ij}$  を更新する。ここで注意しなければならないことは、 $\Delta W_{ij}$  は現在の  $W_{ij}$  に対する修正量（ $W_{ij}$  はバイアス分に相当）であるため、新たに更新される結合荷重は  $W_{ij}(\text{new}) = W_{ij} + \Delta W_{ij}$  となる。以上一連の処理手順を数式化すると以下ようになる。

$$\Delta W_{ij} = \frac{\partial E}{\partial W_{ij}}$$

$$= \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial W_{ij}} = \delta_{ij} o_i \quad (3.9)$$

ここで,

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial y_j} \quad (3.10)$$

$$\frac{\partial E}{\partial o_j} = o_j - t_j \quad (3.11)$$

$$\begin{aligned} \frac{\partial o_j}{\partial y_j} &= f_j'(y_j) \\ &= f_j(y_j) \{1.0 - f_j(y_j)\} \alpha \end{aligned} \quad (3.12)$$

より, 中間層-出力層の誤差に対する入力  $y_j$  の修正量  $\delta_{ij}$  は

$$\begin{aligned} \delta_{ij} &= \frac{\partial E}{\partial y_j} \\ &= (o_j - t_j) f_j(y_j) \{1.0 - f_j(y_j)\} \alpha \end{aligned} \quad (3.13)$$

となる. ただしこれは, 分類処理を行う際のもので, 関数のフィッティング処理を行う場合, 中間層-出力層における  $y_j$  の修正量  $\delta_{ij}$  が異なる. なぜならば, 分類処理では最終的な出力を0から1の範囲で出力するため, 出力層にその動作関数としてシグモイド関数を配置しているが, フィッティング処理の場合は最終出力が連続値となる必要があるため, 出力層にシグモイド関数は配置されていない. そのため, フィッティング処理における  $y_j$  の修正量  $\delta_{ij}$  は次式となる.

$$\delta_{ij} = o_j - t_j \quad (3.14)$$

次にこの結果を受けて, 入力層-中間層における入力  $y_i$  の修正量  $\delta_{hi}$  を求める. このとき, 中間層-出力層では誤差を小さくするための情報として, 出力  $o_j$  と教師信号  $t_j$  との差 ( $o_j - t_j$ ) を用いたが, 入力層-中間層では, 中間層-出力層において既に修正された  $\delta_{ij}$  を用いる. つまり, 誤差を減少させるために, 中間層-出力層で求められた修正情報を利用するのである. そのため  $\delta_{hi}$  を求める式とでは修正情報のみ異なることになる. したがって

$$\begin{aligned} \delta_{hi} &= [\text{誤差に対する次層での修正量}] \\ &\quad \times f_i(y_i) \{1.0 - f_i(y_i)\} \\ &= (\sum_j W_{ij} \delta_{ij}) f_i(y_i) \{1.0 - f_i(y_i)\} \end{aligned} \quad (3.15)$$

となる. これらをもう少し一般化した形で表すと

$$\delta_{k-1} = B_{k-1} f'_{k-1}(y_{k-1}) \quad (3.16)$$

となる. ここで  $k$  は層間を表し,  $k-1$  は入力層-中間層,  $k$  は中間層-出力層である. また,

$$B_{k-1} = \sum W_k \cdot B_k \quad (3.17)$$

$$B_k = o_k - t_k \quad (3.18)$$

である. このように, 入力層-中間層の  $\delta_{hi}$  を求めるためには次層の  $\delta_{ij}$  が必要となる. このため, 計算手順としては中間層-出力層の  $\delta_{ij}$  をまず計算し, 次にこれを利用して前層の  $\delta_{hi}$  を求めるといった具合に順次行っていく. このように  $\partial E / \partial y$  (入力に対する誤差の評価) を出力層より逆向きに前の層に伝えて (計算して) いくこの学習法をバックプロパゲーション法という.

また, 誤差に対する修正情報を示す式(3.15)の  $\sum_j W_{ij} \cdot \delta_{ij}$  は,  $\sum_j W_{ij} \cdot X o_j$  として記述すると理解し易い. 即ち  $\delta_{ij}$  は望ましい  $o_j$  を出力するための望ましい  $o_j$  と解釈することができ, この望ましい  $o_j$  を得るためさらに前層からの入力  $y_i$  を修正する. ( $X$  は  $o_j$  が望ましい  $o_j$  になるための係数) このように考えるとバックプロパゲーション (誤差逆伝搬法) と呼ばれる学習アルゴリズムが直感的に理解できる.

これまでの段階で誤差を小さくするために必要な, 各層における入力  $y$  の変化量を求めることができた. そこで次の段階として, 入力  $y$  を動かすために, 前層の出力  $o$  を乗じる. 即ちこれは, 以下のように結合荷重の修正量  $\Delta W$  を求めることにほかならない.

$$\Delta W_{ij} = \delta_{ij} \cdot o_i \quad (3.19)$$

$$\Delta W_{hi} = \delta_{hi} \cdot o_h \quad (3.20)$$

最後に現在の結合荷重  $W$  に対する結合荷重の修正量  $\Delta W$  を用いて, 現在の結合荷重  $W$  を次式のように新たな結合荷重  $W_{(new)}$  に更新する.

$$W_{(new)} = W + \Delta W \quad (3.21)$$

また, 各層間の結合荷重を実際にプログラムで修正する際には次式により行っている.

$$W_{(new)} = \{W - \epsilon \Delta W\} \times 0.75 + W_{(old)} \times 0.25 \quad (3.22)$$

ここで  $\epsilon$  は通常0.1~0.9の値をとる学習係数,  $W_{(old)}$  は前回学習時の結合荷重である. また, 右辺第二項は前回までの学習結果の影響を次回の学習に取り込むことにより, 結合荷重の急激な変更を抑制するもので, このため慣性項と呼ばれている.

以上の手続きを学習データの個数回実行し, これを1サイクル (学習1回) として, 評価関数値が設定値以下になるまで, あるいは実行時に設定した最大学習繰り返し数に達するまで繰り返し実行する.

## 3.2 ネットワークパラメータについて

バックプロパゲーションを学習アルゴリズムとして,



階層型ニューラルネットワークを実行する際、例えば以下に示すような、予め決定しておかなければならないいくつかのパラメータがある。

- ・ 各層におけるニューロン数
- ・ 収束判定値（評価関数値）
- ・ 最大学習繰り返し数
- ・ シグモイド関数の勾配（ $\alpha$ ）
- ・ 学習係数（ $\varepsilon$ ）
- ・ 閾値（ $\theta$ ）

本節ではこれらの内、特に学習に関するパラメータである  $\alpha$ ,  $\varepsilon$ ,  $\theta$  について言及する。

シグモイド関数の勾配（ $\alpha$ ）：

このパラメータは式(3.3)に示したニューロンの動作関数、即ちシグモイド関数の勾配を表すもので、この  $\alpha$  の値によりシグモイド関数は、非線形性の強いステップ関数から、線形関数に近いものまで連続的に変化させることが可能である。

また、この  $\alpha$  の値を適切に設定することにより、学習の進行速度を速めることが可能であり、収束困難な場合にもより低い評価関数値に到達可能な場合がある。本報告では、 $\alpha = 1.0$  を標準として、1.0~4.0の範囲について検討を行っている。

学習係数  $\varepsilon$ ：

このパラメータはバックプロパゲーションによる学習過程において、新たに結合荷重を更新する際に、どれだけ更新させるかを指示する係数である。通常小さな  $\varepsilon$  を設定すると、学習そのものは安定して進行するが、収束値に達するまでに多くの時間を要し、逆に大きな  $\varepsilon$  を設定すると、学習が振動し、極小値を発見できなくなる場合がある。そのため、学習が期待通りに進行しない

場合、まず始めに変更することが望まれるパラメータである。

閾値  $\theta$ ：

バックプロパゲーションによる学習過程において、中間層（出力層）のニューロンは、そこに入力される入力層からの出力と結合荷重との積の総和がある閾値（ $\theta$ ）を越えたとき、1に近い値を出力し、そうでない場合は0に近い値を出力するものであると既に述べた。この定義に従えば、式(3.1)および式(3.2)は次式のように記述されるべきであろう。

$$o_j = f_j(\sum_i W_{ij} \cdot o_i - \theta_j) \quad (3.23)$$

しかしながら、式(3.1)あるいは式(3.2)においても閾値（ $\theta$ ）を表す項は記されていない。この意味するところは、式(3.23)において、 $-\theta_j = -\theta_j \times 1.0$  と考えれば、常に1.0を出力するニューロンが結合荷重 $-\theta_j$ で結合しているものと見なすことが可能であり、この場合閾値（ $\theta$ ）を別途設定する必要はなくなる。このように入力層の1つまたは複数のニューロンが0以外の定数値（通常は1.0）を出力するとき、中間層の各ニューロンにおける閾値（ $\theta$ ）を0とすることが可能であり、この定数値を出力するニューロンを通常バイアスニューロンと呼ぶ。本報告においても特に言及しない限り、このバイアスニューロンを導入するものとする。また、このバイアスニューロンを導入することの目的は、単に数式が単純化できる、あるいはプログラム化が容易である等の理由によるもので、数理モデルとしての特別な意味合いを持つものではない。

#### 4. ニューラルネットワークによる分類処理

通常我々人間が未知なる物質に対し、それが何であるのかを特定する場合、まずいくつかの特徴に注目し、次いでそれらと共通する特徴を持つ物質を検索するという手順を踏む。簡単に判明する場合もあればそうでない場合もある。特に特徴数が多く、それらが互いに関係を持つような場合、複数の候補者から一つに特定することは極めて困難になる。また同様の処理を通常のノイマン型コンピュータを利用して行う場合を考えると、答えを出力するためには必要な条件を全てプログラム中に網羅しておかなければならない。処理を行う以前に予め必要な条件全てを洗い出し、これらをコンピュータに指示することは、我々人間にとって至難の業である。一方ニューラルネットワークでは、学習がノイマン型コンピュータにおけるプログラム作成過程に相当するため、予め必要な条件を準備する必要は無く、ネットワークに対して事例を適当に提示しさえすれば容易に答えを得ることが可能である。

本章では上記のようなニューラルネットワークの有効性に着目し、簡単な例からニューラルネットワークによる分類処理について検討を行う。

##### 4.1 曖昧さを含むデータの処理とニューラルネットワークの有効性

自然現象の観測データ等、実際のデータには外来ノイズあるいは測定誤差などにより、曖昧さが含まれるケースが多い。今、(1, 1, 0)という入力に対し(1, 0)、また(0, 1, 1)という入力に対し(0, 1)をそれぞれ出力する問題を考える。これに曖昧さが加わり、例えば(1, 1, 0)が(0.8, 0.8, 0.1)となった場合、ニューラルネットワークは如何なる結果を出力するであろうか。この問題を調べるため、市川<sup>2)</sup>により紹介されている例題を参考に、以下のような簡単なシミュレーションを行う。

表4.1はネットワークに提示した学習データ、および学習終了後にこの学習結果を利用して、新たな未知デー

タに対するネットワークの出力、即ち予測結果を示したものである。予測結果によると、期待通りニューラルネットワークは分類を完全に成功しており、正解率は100%である。しかも出力値の精度は極めて高い。またこの結果は、ニューラルネットワークがこの学習により、入力データの小数点以下一桁を四捨五入し、その結果が(1, 1, 0)ならば(1, 0)を、また(0, 1, 1)ならば(0, 1)を出力するというシステムを、入力データと教師データの関係から学んだ事実を示している。逆にいうならば、正しい出力を得るためには、0.5以上を1、0.4以下を0とネットワークが認識できるように入力データを与えさえすればよいということである。この程度の処理であればコンピュータで処理する場合でも、その処理に必要な条件の数が少ないため、さほど大きな問題は生じない。ただし、表4.2の結果は両者の能力に関する決定的な違いを示すものである。

表4.2 予測データ

予測データ					
入力データ			予測結果		
0.9	0.1	0.1	1.000	0.000	
0.5	0.4	0.2	0.999	0.001	
0.4	0.4	0.9	0.000	1.000	
0.1	0.3	0.5	0.000	1.000	

表4.2は表4.1の学習データによる学習結果を再度利用して、新たな入力データに対する予測結果を示したものである。これによると、入力データの小数点以下一桁を四捨五入した結果が(1, 0, 0)ならば(1, 0)を、また(0, 0, 1)ならば(0, 1)をネットワークがそれぞれ出力していることが分かる。この結果は表4.1の結果を否定しているようにも見えるが実はそうではなく、ネットワークは入力データの2番目の項目が、目標としている出力に無関係であることを学習によりすでに記憶していたことによるものである。それでは、コンピュータにこれと同じ入力データを与えた時、コンピュータは如何なる値を出力するだろうか。プログラム中に入力データが(1, 0, 0)およ

表4.1 学習データおよび予測データ

学習データ			予測データ			
入力データ		教師データ	入力データ			予測結果
0.9	0.9	0.0	1.0	0.8	0.3	1.000 0.000
0.8	0.7	0.2	1.0	0.6	0.4	0.997 0.003
0.6	0.8	0.1	1.0	0.9	0.0	1.000 0.000
0.5	0.5	0.4	1.0	0.5	0.2	1.000 0.000
0.0	0.9	0.9	0.0	0.1	0.5	0.000 1.000
0.1	0.8	0.8	0.0	0.4	0.6	0.004 0.996
0.4	0.5	0.5	0.0	0.3	0.9	0.000 1.000
0.2	0.6	0.6	0.0	0.2	0.7	0.000 1.000

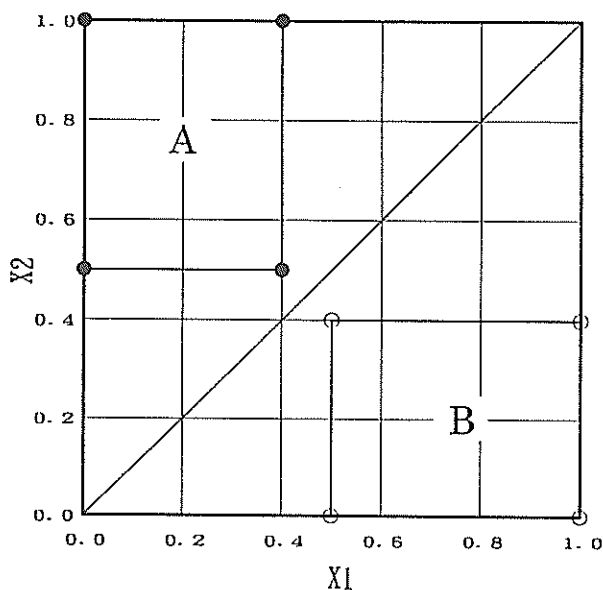


図4.1 ネットワークの学習結果

び(1, 0, 0)の時の処理方法をそれぞれ記述しておかなければ、コンピュータは決して(1, 0)あるいは(0, 1)なる正解を出力することはない。このように、目標関数に対する各入力データ項目の因果関係が、予め十分に分かっていないデータを処理する際、ニューラルネットワークは大変有効であることが理解できる。

#### 4.2 非線形分類

前節では入力データ項目の1番目と3番目の値がそれぞれ(0.0~0.4, 0.5~1.0)の時(0, 1)を、(0.5~1.0, 0.0~0.4)の時(1, 0)を出力する分類問題を処理した。図4.1はこの問題をパターンとして表示したもので、ここに示した2種類のパターンは、例えば図中に示した1本の直線で完全に分離可能である。即ちこの問題は線形分類問題である。そこで本節では、非線形分類問題を対象とする以下のような簡単なシミュレーションを行う。

図4.2は学習および予測に用いた入力データの座標を示したものである。図中に表示した3種類の記号(●, ■, ▲)は、各記号の座標がそれぞれ領域1, 領域2, および領域3の領域であることをネットワークに認識させるための学習用入力データを示している。またアスタリスク(\*)は学習終了後、この結果を利用して各座標の領域がどの領域に属するものをネットワークに予測させる予測用入力データを示している。これら各入力データの値とネットワークの予測結果を表4.3に示す。ここで予測データの番号は図4.2に示したアスタリスク下の番号と対応している。この結果によると、各領域の学習データ付近に存在する予測データは、正確にその領域を認識していることが分かる。また、特に興味深いのは、

各領域の中間点付近に存在する、例えばデータ番号(30, 32, 36, 37)で示した予測データの出力結果である。これらはいずれも二種類の領域に渡り約0.5という値を示しているため、この出力結果からはどちらの領域に属するものか判断することができない。別の言い方をすれば、これらのデータは領域1あるいは領域2に属する、というような曖昧さを、そのまま出力することで、ネットワークはその領域が不明確なデータを無理矢理どちらかの領域とするのではなく、他方の領域に属する可能性を素直に示しているものと思われる。

図4.2中の実線は、表4.3の予測結果に基づき各領域を分離する境界を示したものである。入力データの座標がこの境界線に近付くほど、その予測結果に含まれる曖昧さが増すことになる。また、領域3と領域2は一本の直線で分離可能な、前節同様の線形分離問題である。一方、領域1と領域2は、いかなる直線でも分離不可能な非線形分離問題である。この領域1と領域2を分離するためには、図中に示すような放物型の分離曲線、即ち広い意味での非線形関数を導入しなければならない。通常この非線形関数を知ることは、特に系がより複雑になった場合、大変困難である。しかしながらニューラルネットワークはこの結果が示すように、容易に分離可能である。この非線形分離能力こそニューラルネットワークの大きなセールスポイントの一つである。それでは、ニューラルネットワークはどのようにして非線形分類を行っているのであろうか。この問題について次節で検討する。

#### 4.3 XOR問題に見るネットワークの内部状況

本節ではニューラルネットワークが如何に非線形分離を可能にしているのか、その様子を調べるため、線形分離不可能な典型例として多くの参考書(例えば文献3~5)に取り上げられている、XOR(exclusive OR)問題について検討を行う。

XOR問題とは、表4.4に示すように、0, 1から成る2つの入力データ  $\{x_1, x_2\}$  に対し、同じく0, 1の出力を与える論理関数  $\phi \{x_1, x_2\}$  の設定問題である。実際に  $\phi \{x_1, x_2\}$  を  $x_1, x_2$  の関数として、例えば  $\phi = ax_1 + bx_2$  のような線形和で表現することを考えると、 $a, b$  いずれの係数をどのように設定しようとも、表4.4の論理を実現することはできない。しかしながら、新たな項として  $x_3 = x_1 x_2$  を導入し、さらに  $\phi = ax_1 + bx_2 + cx_3$  とすると、 $a, b, c$  の各係数がそれぞれ  $a=1, b=1, c=-2$  の時、このXOR問題を解くことが可能となる。つまりこの問題を解く鍵は、非線形関数  $x_3$  を見つけることができるか否かであり、これを見つけていなければならない。しかし実際のデータを扱う際、ここでの  $x_3$  のような非線形項を見つけだすことは極めて困難な場合が多い。それではニューラルネットワークはどのようにしてこのXOR問題を解くのであろうか。

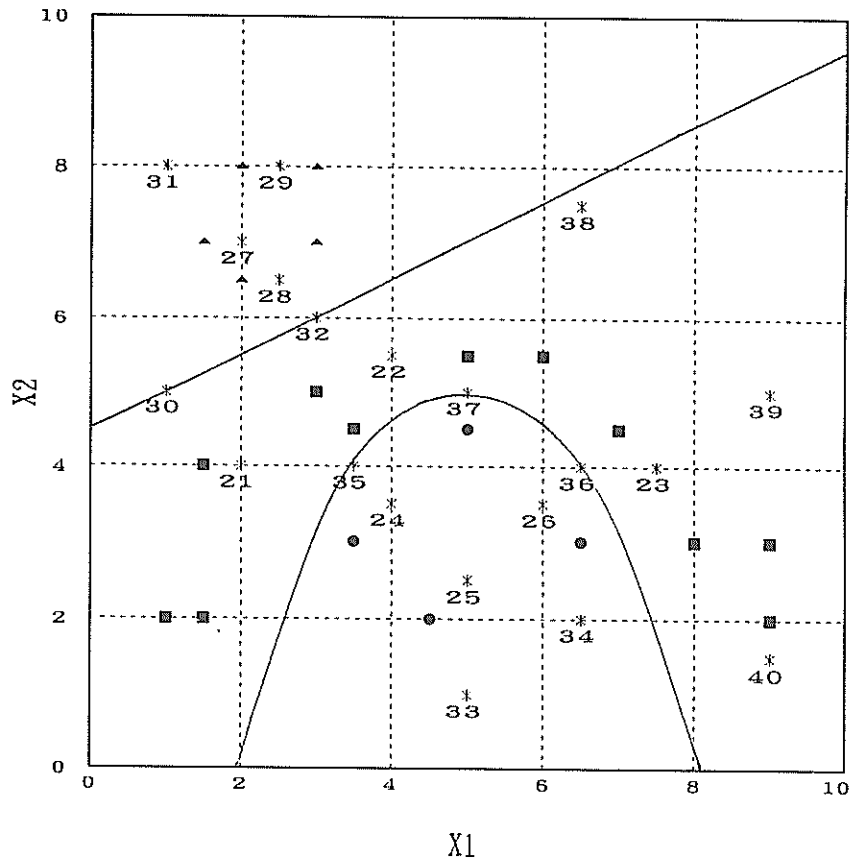


図4.2 分類問題の入力データ座標および分類結果

表4.3 分類問題の学習データおよび予測データ

学習データ			予測データ						
入力データ		領域	番号	入力データ		予測結果			判定
3.5	3.0	1 (●)	21	2.0	4.0	0.000	0.990	0.006	2
4.5	2.0	1 (●)	22	4.0	4.5	0.002	0.980	0.014	2
5.0	4.5	1 (●)	23	7.5	4.0	0.017	0.987	0.000	2
6.5	3.0	1 (●)	24	4.0	3.5	0.980	0.022	0.000	1
1.0	2.0	2 (■)	25	5.0	2.5	0.999	0.001	0.000	1
1.5	2.0	2 (■)	26	6.0	3.5	0.988	0.013	0.000	1
1.5	4.0	2 (■)	27	2.0	7.0	0.000	0.004	0.995	3
3.0	5.0	2 (■)	28	2.5	6.5	0.000	0.033	0.968	3
3.5	4.5	2 (■)	29	2.5	8.0	0.000	0.005	0.998	3
5.0	5.5	2 (■)	30	1.0	5.0	0.000	0.425	0.510	2or3
6.0	5.5	2 (■)	31	1.0	8.0	0.000	0.001	0.998	3
7.0	4.5	2 (■)	32	3.0	6.0	0.000	0.515	0.576	2or3
8.0	3.0	2 (■)	33	5.0	1.0	0.997	0.001	0.000	1
9.0	2.0	2 (■)	34	6.5	2.0	0.996	0.003	0.000	1
9.0	3.0	2 (■)	35	3.5	4.0	0.285	0.849	0.000	2
1.5	7.0	3 (▲)	36	6.5	4.0	0.597	0.444	0.000	1or2
2.0	6.5	3 (▲)	37	5.0	5.0	0.522	0.498	0.000	1or2
2.0	8.0	3 (▲)	38	6.5	7.5	0.000	1.000	0.517	2or3
3.0	7.0	3 (▲)	39	9.0	5.0	0.003	0.998	0.000	2
3.0	8.0	3 (▲)	40	9.0	1.5	0.012	0.984	0.000	2

表4.4 XOR問題の入出力関係

入力 $\{x_1, x_2\}$	出力 $\phi$
0 0	0
1 0	1
0 1	1
1 1	0

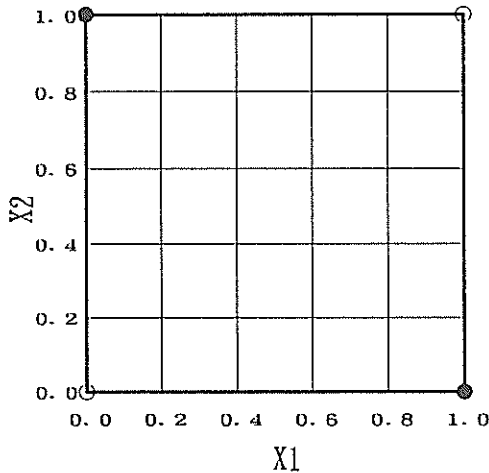


図4.3 XOR問題の入力データ座標

図4.3は表4.4に示した入力データ  $\{x_1, x_2\}$  の座標を示したものである。この図からも明らかなように、入力データをそのまま領域1と領域2に分離可能な直線は存在しない。しかしながら分離に成功したネットワークの出力層ニューロン数は1であり、しかもここでの処理は、単にシグモイド関数を求めているにすぎないことから、シグモイド関数の入力値がすでに線形分離可能な状態になっていることが予想される。(なぜならば、表4.5に示した出力層からの出力値を領域1と領域2のいずれかに判定する境界として0.5を仮定すると、式(3.3)で示したシグモイド関数の解が0.5以上を出力するためには、その入力が0以上、また0.5未満を出力するためには、その入力が0未満でなければならない。)

また、学習により求められた結合荷重行列は、表4.5に示した各入力データに対し不変であるため、この4

表4.5 出力層ニューロンの出力結果

入力 $\{x_1, x_2\}$	出間層出力
0 0	0.0144
1 0	0.9804
0 1	0.9819
1 1	0.0143

表4.6 中間層ニューロンの出力結果

入力 $\{x_1, x_2\}$	中間層出力
0 0	0.9898 0.4674
1 0	0.0062 0.1607
0 1	1.0000 0.8031
1 1	0.9959 0.4710

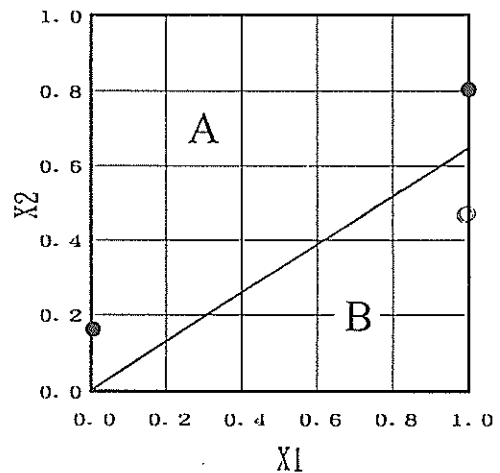


図4.4 中間層の出力結果

ケースそれぞれにおける出力層への入力値の違いは、中間層の出力の違いにほかならない。この問題に関して松葉<sup>1)</sup>は、中間層の果たす役割は、線形分離可能なように入力変数を変換することであると示している。そこで、ネットワーク内部で処理されるため、通常見ることのない中間層の出力について調べた結果を表4.6および図4.4に示す。ここで図4.4は表4.6に示した中間層の出力値を図4.3同様、座標として示したものである。これによると、領域Aと領域Bは、容易に線形分離可能な状態であることが分かる。

以上の結果、中間層は図4.3から図4.4のように、線形分離不可能な状態から、線形分離可能な状態へ入力データを変換している様子を実際に確認することができる。

以上の結果は、学習に必要な各種ネットワークパラメータを表4.7に示した設定値としたときのものである。

表4.7 ネットワークパラメータ

入力層 ニューロン数	2
中間層 ニューロン数	2
出力層 ニューロン数	1
動作関数の勾配 ( $\alpha$ )	1
学習係数 ( $\epsilon$ )	0.5

通常ニューラルネットワークを実際の問題に適用する際、その問題に合わせてネットワークパラメータを適切に設定する必要がある。しかしながら、これらネットワークパラメータの設定法に関しては、有効な手法が確立されているとは言い難く、結局のところ使用者の経験あるいは判断に委ねられることになる。そして厄介なことは、これらパラメータの設定値により、学習結果が大きく左右されるケースが少なくないということである。この様子を具体的に調べるため、ネットワークパラメータの値として、特に $\alpha$ および $\epsilon$ を変化させた時の学習回数に伴う評価関数の挙動を図4.5に示す。

(1) 動作関数 (シグモイド関数) の勾配  $\alpha$

まず始めに動作関数 (シグモイド関数) の勾配  $\alpha$  に着目すると、(f)を除く全ケースにおいて、 $\alpha$ が4の誤差曲線の方が学習回数に対する評価関数値の減少率が大きいことが分かる。つまりこの結果は、 $\alpha$ の値が1よりも4の方が、収束に必要な学習回数が少ないことを示している。また、(c)と(d)を比較すると、(c)では学習回数が増加しても一向に収束の気配を見せないが、(d)では学習回数の増加に伴い、単調に評価関数値が減少している。この様にいくら学習回数を増やしても収束しない場合に、 $\alpha$ の値を大きくすることによって、より小さな評価関数値に到達可能な場合がある。(もちろんこの逆のケースも存在する)ここで示したように、学習の高速化という意味では、 $\alpha$ の値を大きくすることは確かに有効な手段の一つではある。しかしながらこの行為は、同時にニューラルネットワークの大きな特徴の一つである、曖昧な出力を制限するものでもあり、例えば「このデータは領域Aあるいは領域Bに属する」といった回答が期待しにくくなることに注意が必要である。数学的な根拠はないが、これまでの経験によると $\alpha$ の最大値としては、4程度に抑えておくことが安全なようである。

(2) 学習係数  $\epsilon$

次に学習係数  $\epsilon$  に着目する。3.2で述べたように $\epsilon$ の値が小さすぎると学習に多くの時間がかかり、また大きすぎると学習が振動し、収束しにくいことが一般に知られている。図4.5はこの様子を良く表している。例えば(b)と(d)を比較すると、評価関数値が $10^{-3}$ に到達する学習回数は、 $\epsilon$ が0.1の(b)では10000回、また $\epsilon$ がその5倍である0.5の(d)では(b)の1/5の学習回数である2000回となっており、非常に良く対応がとれている。さらに $\epsilon$ の値を0.9に増加させた(f)では、もはや学習の進行を認めることができない。また(g)および(h)は $\epsilon$ を式(4.1)で示すように、評価関数値(E)の関数として、0.1~0.5まで徐々にその値を大きくしたものである。

$$\epsilon_{(new)} = \epsilon_{(old)} + \text{Log}_{10}(E) \times \epsilon_{(old)} \quad (4.1)$$

ここで興味深いのは、 $\epsilon$ を0.5に固定した(c)では評価

関数値を低減することができなかったのに対し、(g)では十分小さな評価関数値を示している。さらにその値は $\epsilon$ を0.1に固定した(a)よりも小さいことが分かる。この結果は補章Aで述べる焼き鈍し法の有効性を示すものと思われる。

以上ネットワークパラメータ $\alpha$ および $\epsilon$ の値によって、学習の収束状況が大きく異なる様子を確認した。また、これまで議論してきたネットワークパラメータ以外にも、学習の収束状況を左右するもう一つのファクターとして、結合荷重行列の初期値が挙げられる。例え同一のネットワーク構造、ネットワークパラメータを用いても、結合荷重行列の初期値の違いにより、学習の収束状況も大きく異なる場合がある。図4.5は表4.8に示す結合荷重行列の初期値を採用して学習を行ったものである。そこで、この結合荷重行列の初期値を表4.9のように変更し、その他の学習条件には一切変更を加えずに、図4.5と全く同一の学習条件で再度学習を行った。この結果を図4.6に示す。

表4.8 結合荷重行列の初期値

中間層 ニューロン	入力層 ニューロン		出力層 ニューロン
	1	2	1
1	-0.50	+0.50	+2.00
2	-1.00	+1.00	-3.00

表4.9 結合荷重行列の初期値

中間層 ニューロン	入力層 ニューロン		出力層 ニューロン
	1	2	1
1	+4.00	-1.00	+1.50
2	+2.00	+2.00	+0.50

これによると全ケースにおいて低い評価関数値が得られていることが分かる。特に注目すべき点は、図4.5では収束が見られなかった(c), (e), (f)に対しても図4.6のそれらは十分低い評価関数値に達していることである。つまりこの結果は、同一のネットワーク構造、同一の学習条件で学習しても、結合荷重行列の初期値により収束する場合と収束しない場合があることを示している。別の言葉で言うならば、ネットワークは結合荷重行列の初期値の違いにより、異なる結果を出力する可能性があるということである。そのため実際の問題に対しニューラルネットワークを実行する場合、数倍の時間を必要とするが、より安定な出力を得るためには、いくつかの異なる

る結合荷重行列の初期値を予め用意し、これらを用いて複数回実行することが望ましいものと思われる。

ただし、ニューラルネットワークの学習は「やってみたら旨くいった」の世界である。そのため種々の学習方法で実行した結果、その中のあるケースが評価データを旨く再現できたならば、統計的にも妥当性が保証されるため、その学習結果を採用することに問題は無いものと思われる。

以上本節では、XOR問題を例題として中間層の役割、ネットワークパラメータの変化に対する出力特性、および出力に対する結合荷重行列の初期値依存性について検討を行った。ここで注意しておかなければならないことは、XOR問題は入力が極端に少なく、また単純なため、

本節で行った様なネットワークの動作特性を確認するためには大変都合がよい。しかしニューラルネットワークの学習目的は、未学習の入力に対し適切な出力を得ることである。つまり予測力を最適にするため、ネットワークの構造、パラメータあるいは学習データ数、結合荷重行列の初期値等これらを適当に調整することになる。従って、未学習の予測データに対する誤差が小さくなることが本来の学習成功であり、学習に使用したデータに対する誤差がいくら小さくなくても、学習に成功したことにはならない。この様な意味で、何かを予測することのないXOR問題は、練習問題の枠を越えるものではなく、単に好ましくない過学習(5.3参照)を行っているもので、学習としては無意味なものと思われる。

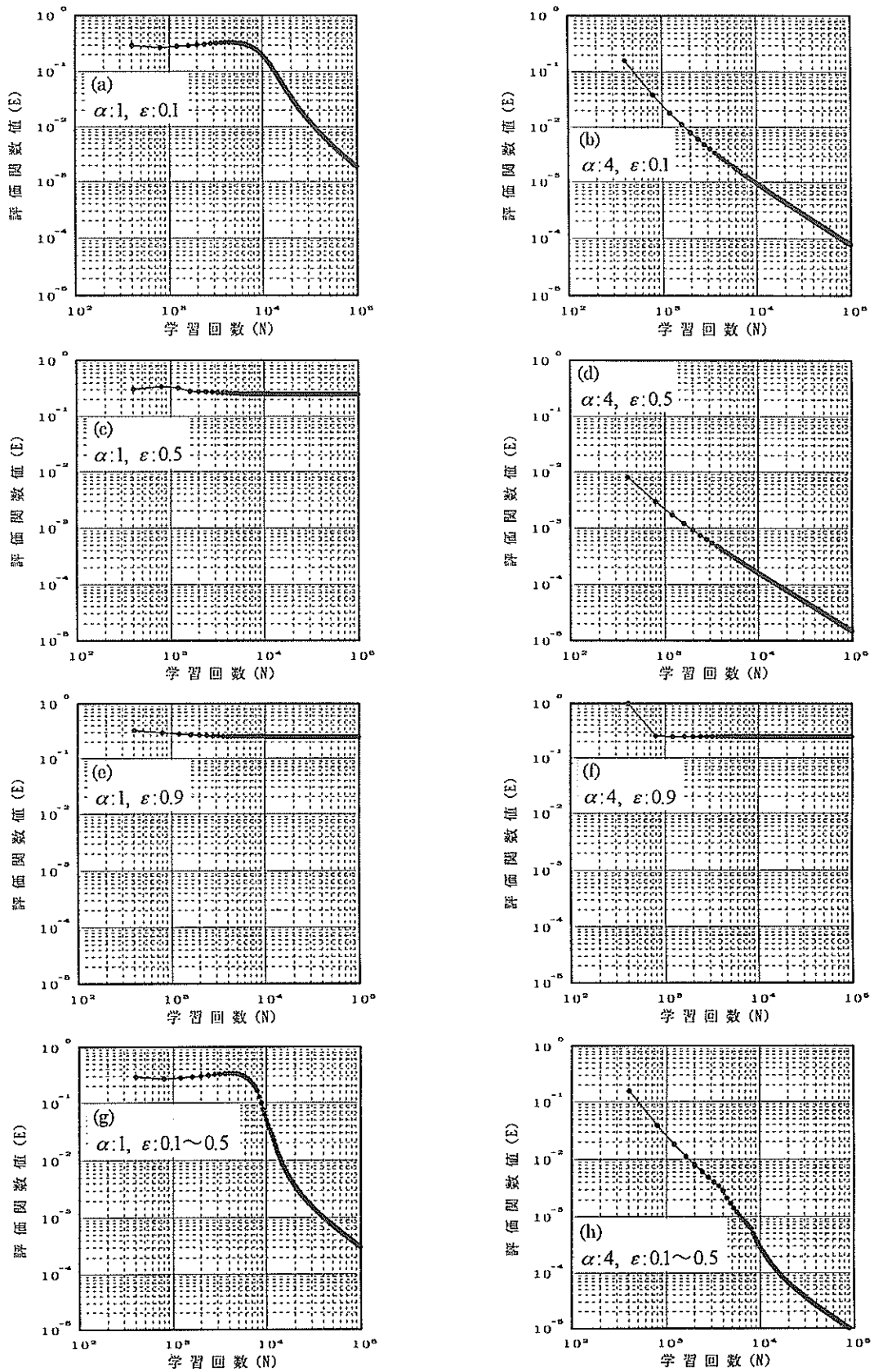


図4.5 ネットワークパラメータの違いによる評価関数値



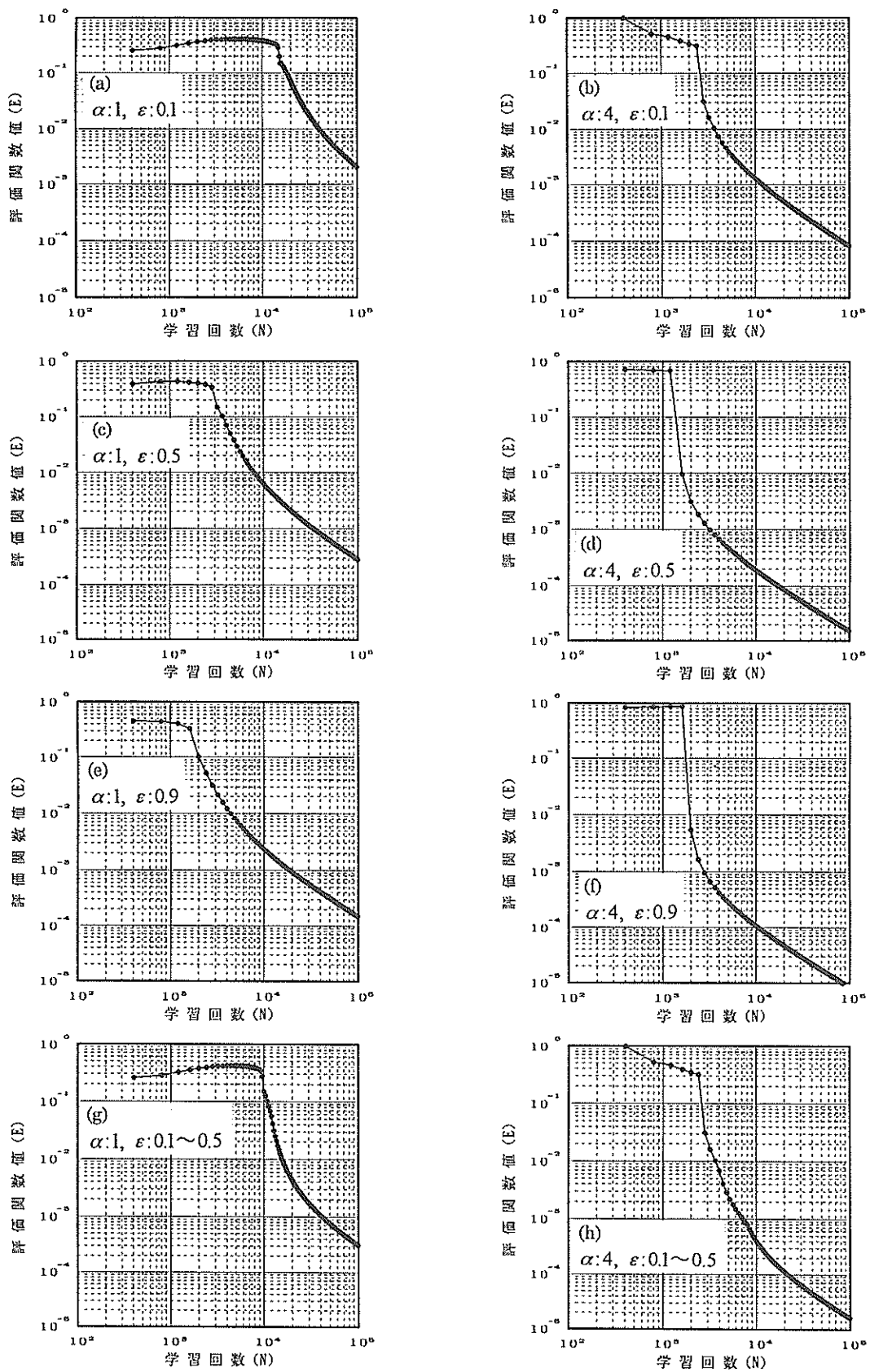


図4.6 結合荷重の初期値変更後の評価関数値

表5.1 ネットワークパラメータ

5. ニューラルネットワークによるフィッティング

前章ではニューラルネットワークによる分類処理について検討を行い、線形分類のみならず、非線形分類をも可能とするニューラルネットワークの高い分類能力を確認した。本章では階層型ニューラルネットワークを利用することのもう一つの目的である、関数のフィッティングについて検討を行う。また本章の目的は、関数のフィッティング処理に関するニューラルネットワークの出力特性を明らかにすることである。そのため、対象とする関数については特殊な関数ではなく、日頃接する機会の多いものとして、市川<sup>2)</sup>の例題としても紹介されている  $y = x$  および  $y = x^2$  を使用する。

入力層 ニューロン数	2
中間層 ニューロン数	2, 10
出力層 ニューロン数	1
動作関数の勾配 ( $\alpha$ )	4
学習係数 ( $\epsilon$ )	0.1

5.1 直線のフィッティング

本節では関数のフィッティングとして最も単純な線形関数  $y = x$  について、ネットワークの出力特性を検討する。学習に用いる入力データは0~10までの偶数値 (0, 2, ..., 10) とし、学習終了後、0~10までの全整数値 (0, 1, ..., 10) を入力し、これらの入力に対する予測を行った。なお、学習に採用したネットワークパラメータを表5.1に示す。

図5.1は中間層ニューロン数が2(a)、および10(b)の2つのネットワークによる各学習段階(評価関数  $E = 10^{-1}$ ,  $10^{-2}$ ,  $10^{-4}$ ) における出力結果を示したものである。両者を比較すると、学習の未成熟時(評価関数  $E = 10^{-1}$ ,  $10^{-2}$ ) に多少の違いが見られるが、学習の進行に伴って予測値が徐々に真値(目標関数)に近付いている様子を確認することができる。また、学習成立時(評価関数  $E = 10^{-4}$ ) の出力結果には、注目すべき大きな差は見られず、全体的にはほぼ同程度の精度といえよう。ただし、学習成立時までの学習回数は両者で大きく異なり、中間

層ニューロン数が2のネットワークは、中間層ニューロン数が10のネットワークに対し、約11倍の学習回数を要する結果となった。表5.2に学習成立時の学習回数、および両ネットワークによる出力結果を示す。このように学習成立後のネットワークの出力は、実用上十分と思われる精度で  $y = x$  の線形関数を予測、即ちフィッティング可能であることが確認できた。また、中間層ニューロン数が異なる両ネットワークの学習回数が示しているように、中間層のニューロン数を変化させることにより、場合によっては学習速度を速めることが可能である。同時に、このような時には、予測精度が向上することがこれまでの経験で多く見られた。

次に、上記同様  $y = x$  の線形関数を対象としたネットワークの内挿能力について検討する。

学習に使用する入力データは2点 (□;  $x = 0, 10$ ) および3点 (●;  $x = 0, 5, 10$ ) の2ケースとし、学習終了後 0~10までの全整数値を予測した。この結果を図5.2に示す。なお学習は図5.1の結果を参考に、評価関数値が  $10^{-4}$  に達するまでを行い、その他の学習条件の変更は行っていない。これによると2点および3点の入力データを用いて学習させた両ケースとも、学習点以外の区間は、概ね比例的に内挿が行われていることが分かる。また、図中直線で示した真値と予測値との差は2点学習より3点学習の方が小さく、この結果は直感と良く合致している。

表5.2 学習成立時の学習回数とネットワークの出力結果

		中間層ニューロン数 2		中間層ニューロン数 10	
学習データ		学習回数 : 7284		学習回数 : 658	
x	y	出力	誤差	出力	誤差
0	0.0	0.084	0.084	-0.003	-0.003
1	1.0	0.949	-0.051	0.951	-0.049
2	2.0	1.881	-0.119	1.960	-0.040
3	3.0	2.870	-0.130	2.990	-0.010
4	4.0	3.902	-0.098	4.033	-0.033
5	5.0	4.959	-0.041	5.077	0.077
6	6.0	6.020	0.020	6.111	0.111
7	7.0	7.066	0.066	7.123	0.123
8	8.0	8.076	0.076	8.102	0.102
9	9.0	9.033	0.033	9.037	0.037
10	10.0	9.923	-0.077	9.919	-0.081

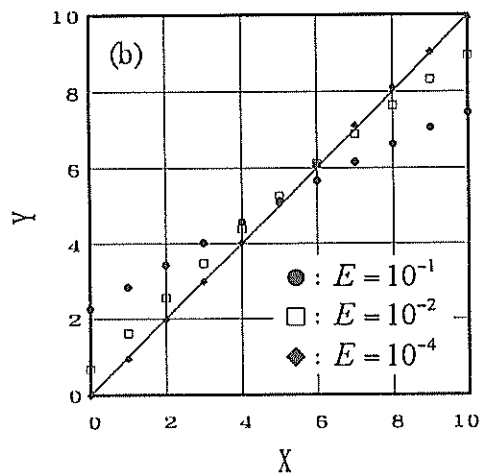
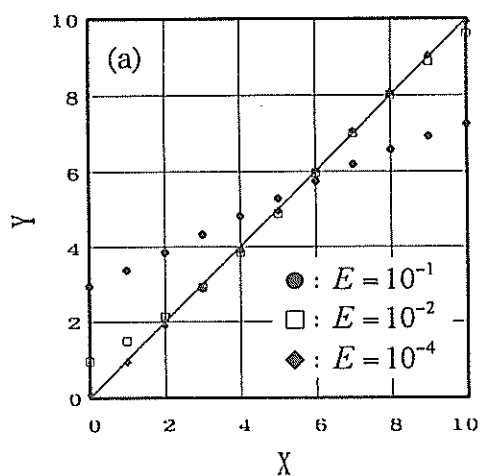


図5.1 各評価関数値 (E) に対する線形関数の予測結果  
(a : 中間層ニューロン数2, b : 中間層ニューロン数10)

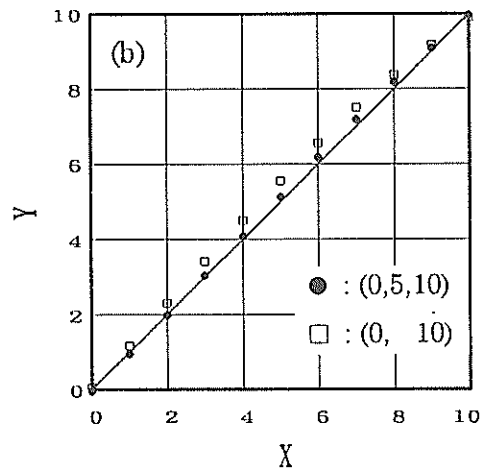
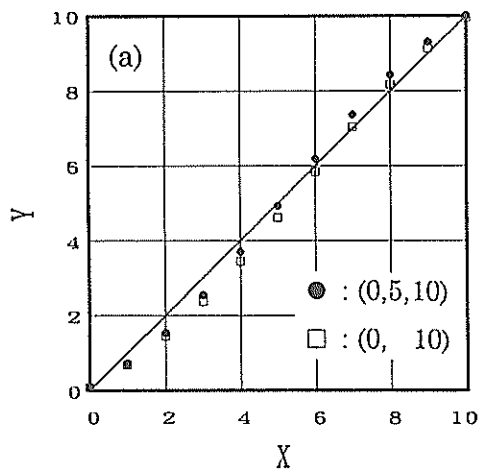


図5.2 学習データの違いによる予測 (内挿) 結果  
(a : 中間層ニューロン数2, b : 中間層ニューロン数10)

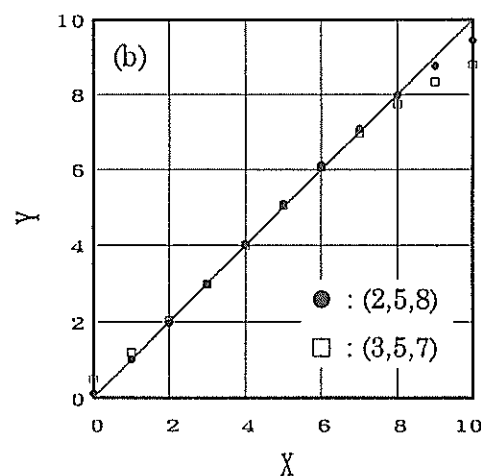
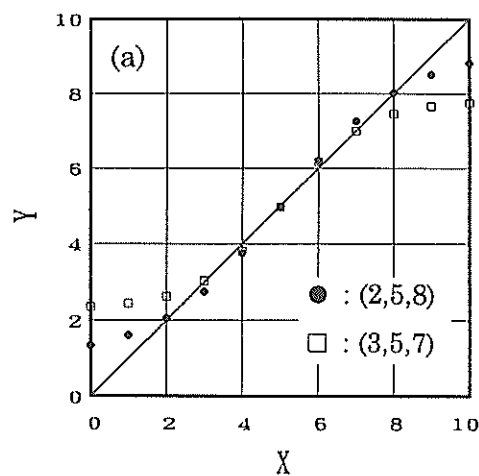


図5.3 学習データの違いによる予測 (外挿) 結果  
(a : 中間層ニューロン数2, b : 中間層ニューロン数10)

最後に同じく  $y = x$  の線形関数を対象としたネットワークの外挿能力について検討する。

学習に使用する入力データは3点 (□;  $x = 3, 5, 7$ ) および (●;  $x = 2, 5, 8$ ) の2ケースとし、学習終了後 0~10までの全整数値を予測した。学習条件は内挿の場合と全く同様である。この結果を図5.3に示す。これによると同数の学習点を用いて学習させた場合でも、学習用入力データの領域が広い ( $x = 2, 5, 8$ ) 方が外挿精度は高く、真値からのズレが小さいことが分かる。また中間層ニューロン数が2の(a)に着目すると、外挿値、即ち学習点の領域を外れた入力値に対する予測値は、学習点の領域内における最大値、あるいは最小値に近付こうとする傾向が確認できる。つまり、学習に用いた入力データの最小値よりも小さな入力に対しては、その最小値に、反対に学習に用いた入力データの最大値よりも大きな入力に対しては、その最大値にそれぞれ予測値が引っぱられる結果となった。この結果は学習データの情報を基に、データの予測を行うニューラルネットワークの学習方法の特性上、妥当な結果のように思われる。一方、中間層ニューロン数が10の(b)に着目すると、(a)に比べ学習に使用した入力データの違いに伴う予測結果の差が減少し、また小さな入力値に対する外挿値より、大きな入力値に対する外挿値の方が真値である線形関係からのズレが顕著となっている。このように図5.3は、同様の評価関数値まで学習を行った場合でも、中間層ニューロン数の違いによりネットワークの出力特性は大きく異なり、また、中間層ニューロン数を十分にとることで、外挿精度を向上させることが可能となった例を示すものである。

本節の結果をまとめると、ニューラルネットワークの学習は、学習データの情報から未知データに対する予測を行うため、本来的に外挿は不得意である。そのため予測精度の向上を図るためには、予測に用いる入力データが学習データに対し内挿の形、即ち予測データは学習データの範囲内であることが望まれる。特に大きな値に対しては、十分注意を払う必要がある。

## 5.2 曲線のフィッティング

前節ではニューラルネットワークによる関数のフィッティングとして、その基本的な出力特性を検討するため  $y = x$  なる線形関数を対象とした。しかし実際の現象を対象とする場合、その対象が線形関数で表現されることは少なく、通常不規則な変動を有する非線形、あるいは非線形的な関数であることが多い。そこで本節では、 $y = x^2$  なる非線形関数を対象にニューラルネットワークによるフィッティングについて検討を行う。

学習に用いる入力データは-10~10までの偶数値(-10, -8, ..., 0, ..., 8, 10)とし、学習終了後、-10~10までの全整数値(-10, -9, ..., 0, ..., 9, 10)を入力し、これらの入

力に対する予測を行った。なお、学習に採用したネットワークパラメータを表5.3に示す。

表5.3 ネットワークパラメータ

入力層 ニューロン数	2
中間層 ニューロン数	10
出力層 ニューロン数	1
動作関数の勾配 ( $\alpha$ )	4
学習係数 ( $\epsilon$ )	0.15

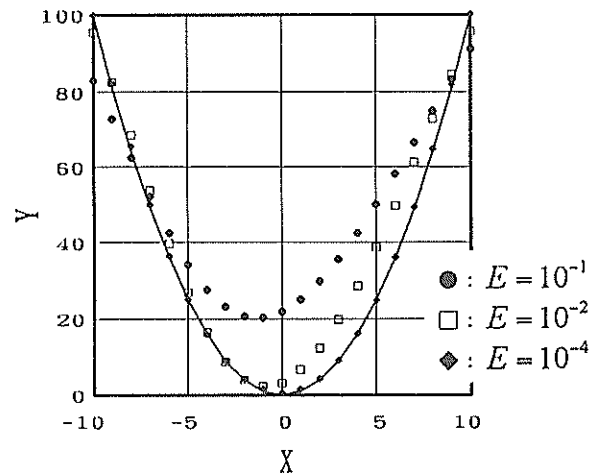


図5.4 各評価関数値に対する非線形関数の予測結果

図5.4は各学習段階(評価関数  $E = 10^{-1}, 10^{-2}, 10^{-4}$ )における出力結果を示したものである。予測結果は学習の進行に伴い徐々に目標関数に近付き、学習成立時(評価関数  $E = 10^{-4}$ )には、ほぼ完全に目標関数のフィッティングに成功していることが分かる。このようにニューラルネットワークによる関数のフィッティングは、非線形関数に対しても有効であることが確認できた。また特別な前処理等を行うことなく、入出力の関係を直接ネットワークに提示することで、任意の関数を近似可能なニューラルネットワークは、自然現象等実際のデータを扱う際大いに有効であるものと思われる。

次に、非線形関数を対象とした場合のニューラルネットワークの内挿能力について検討を行う。学習に用いた入力データは5点 (●;  $x = -10, -5, 0, 5, 10$ ) およびこれらから0を除いた4点 (□;  $x = -10, -5, 5, 10$ ) の2ケースとし、評価関数値が  $10^{-4}$  に達するまで学習を行った。学習終了後、これら2ケースそれぞれの学習情報を基に、-10~10までの全整数値 (-10, -9, ..., 0, ..., 9, 10) に対す

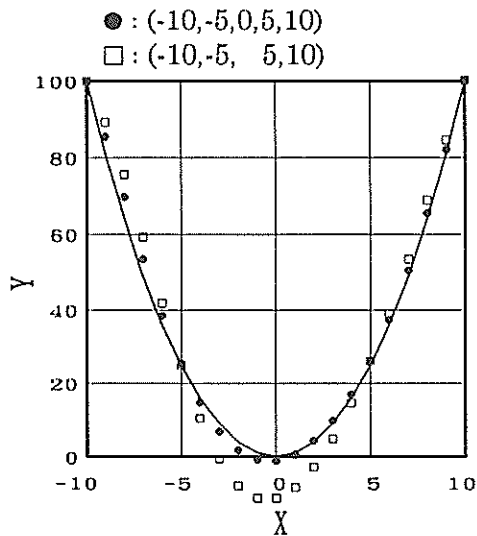


図5.5 学習データの違いによる予測結果  
(中間層ニューロン数10)

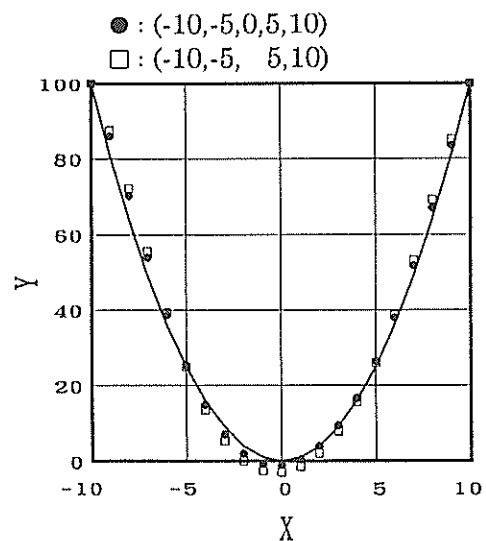


図5.6 学習データの違いによる予測結果  
(中間層ニューロン数3)

る予測を行った。この結果を図5.5に示す。これによると4点で学習を行ったものより、5点で学習を行ったものの方が明らかに予測精度が高い。特に注目すべき点は、 $x = 0$  付近における両ケースの出力差が顕著となっていることである。つまりこの結果は、変曲点付近のデータを学習に採用することが予測精度の向上を図るためには、必要であることを示しているものと思われる。

図5.6は図5.5と全く同様の学習条件で、中間層のニューロン数のみを3に変更した場合の予測結果を示したものである。4点学習による予測結果を図5.5と比較すると、図5.6の方が明らかに目標関数に対する一致度が高いことが分かる。通常この種の問題では、中間層のニューロン数をある程度増加させることにより、予測精度も向上する場合が多い。しかしここで示したように、中間層のニューロン数を減少させることで、予測精度が向上する場合もある。やはりそこには、対象にあったネットワークサイズが存在するようである。同時に、図5.5の方が予測精度が低くなった原因として、次節で検討するオーバーフィッティング（過学習）の可能性が考えられる。

### 5.3 オーバーフィッティング（過学習）

オーバーフィッティングとは、学習データに対し目標関数への近似度を追求するあまり、学習データに対する近似精度は向上するが、未知データに対する予測精度が悪化する現象である。実際のデータには例えば計測誤差等、そこには様々なランダムネスが含まれている場合が少なくない。ニューラルネットワークはその強力なフィッティング能力が故、学習のさせ方によっては、本来不要であるこの微細なランダムネスをも学習してしまう危険性を有している。その結果ネットワークは、目標関数が有する変動の本質を見失ってしまうことになる。こ

の様子を具体的に調べるため、本節では目標関数を  $y = x^2 + e$  ( $x = -5, -4, \dots, 4, 5$ ) に設定した。ここで  $e$  は上記ランダムネスである。ネットワークに提示した入力データ( $x$ )、および教師データ( $y$ )を表5.4に示す。

表5.4 入力および教師データ

x	y	目標値
-5	29.49	25.00
-4	11.43	16.00
-3	12.96	9.00
-2	1.23	4.00
-1	1.32	1.00
0	1.60	0.00
1	3.80	1.00
2	0.15	4.00
3	11.06	9.00
4	12.51	16.00
5	22.82	25.00

表5.5 ネットワークパラメータ

入力層ニューロン数	2
中間層ニューロン数	10
出力層ニューロン数	1
動作関数の勾配 ( $\alpha$ )	4
学習係数 ( $\epsilon$ )	0.1~0.5

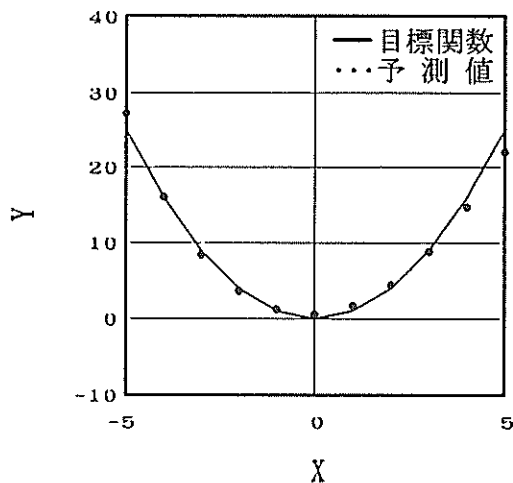


図5.7 2千回学習を行った時の予測結果

図5.7は表5.5に示した学習パラメータを用いて2千回学習を行ったときの各入力に対するネットワークの出力結果を示したものである。これによると、ネットワークの出力は入力に含まれるランダムネスに影響されることがなく、ほぼ真の目標関数である  $y = x^2$  を良く近似していることが分かる。本来ニューラルネットワークはこのように、ランダムネス（ノイズ）に対してロバストな性質を持っている。

図5.8は図5.7と同様の学習条件で、学習回数を10万回に増加させたときの各入力に対するネットワークの出力結果を示したものである。これによるとネットワークの出力結果は、入力と出力の因果関係として、入力に含まれるランダムネスまで学習していることが分かる。即ち入力データに対するあてはめ過ぎであり、このような場合を過学習と呼ぶ。図5.9は上記2ケースそれぞれの学習情

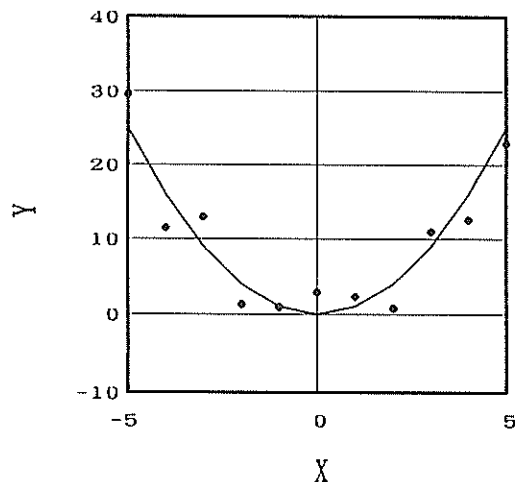


図5.8 10万回学習を行った時の予測結果

報を基に、実際に未学習データとして  $x = -4.5 \sim 4.5$  まで1刻みの入力に対するネットワークの予測結果を示したものである。この図からも明らかなように、学習回数10万回としたときのネットワークの予測結果は過学習により、目標関数から大きく外れる結果となった。このように同一のネットワーク構造でも、必要以上に学習回数を増加させることにより、過学習が起こり得ることが分かった。

また、過学習を回避するための手段としては、中間層ニューロン数を必要最小限にすべきであると一般に言われている。そこで中間層のニューロン数と過学習の関係について検討を行った。学習条件は図5.9と同様で、中間層のニューロン数のみを2に変更し、10万回学習を行なった。図5.10はここで得られた学習情報を基に、図5.9と同様に未学習データとして、 $x = -4.5 \sim 4.5$  まで1

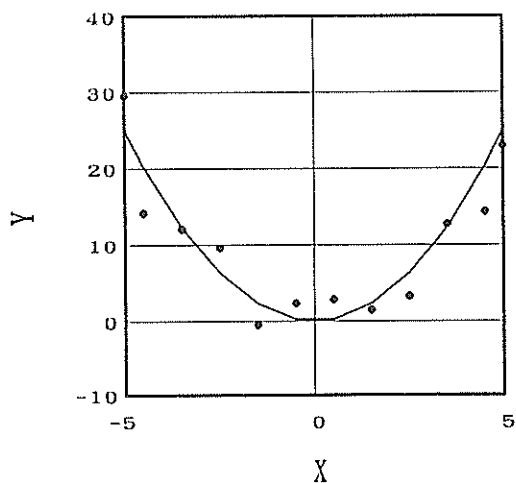


図5.9 未知データに対する予測結果  
(中間層ニューロン数10)

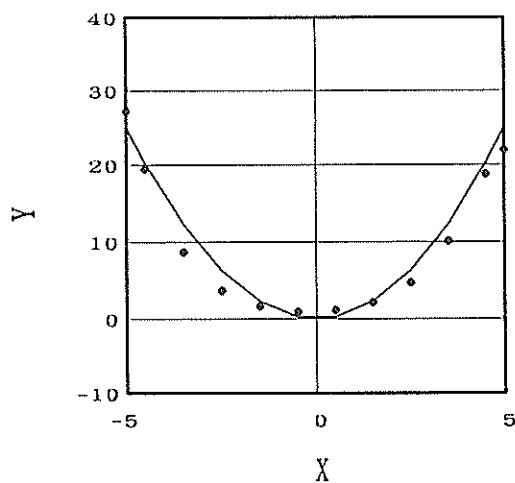


図5.10 未知データに対する予測結果  
(中間層ニューロン数2)

刻みの入力に対するネットワークの予測結果を示したものである。図5.10と図5.9を比較すると、両者の予測結果は大きく異なり、図5.10には図5.9で見られた目標関数との大きなズレは見られず、ほぼ真の目標関数を近似していると言えよう。このように中間層のニューロン数を10に設定し、10万回学習を行った場合、典型的な過学習となったが、同じ学習回数でも中間層のニューロン数を2とした場合、過学習にはなっていないことが分かる。これらの結果より、中間層のニューロン数が過学習に大きく関与していることが分かった。特に中間層のニューロン数をむやみに増加させると過学習に陥る危険性が高まり、逆に中間層のニューロン数が適当に設定されている場合には、学習回数を増加させても過学習にはなりにくいことが確認できた。

本節では学習回数および中間層のニューロン数に関して、これらと過学習の関係について検討を行い、過学習

を回避するためのいくつかの知見を得ることができた。また本来的に、過学習を回避するための最も効果的な手段としては、入力データに含まれているランダムネスを、統計手法等の何らかの手法により取り除くことである。そのような意味で、対象の変動特性を予め知ることは重要であるが、種々の制約によりそれが困難な場合も少なくない。この様なとき、例えば入力データに対し移動平均等の処理を施すことにより、データのランダムウォークを低減させることは有効な手法であるものと思われる。

また特に学習回数に関して、広く知られた過学習の回避方法にクロスバリデーションと呼ばれるものがある。これは全学習データのうち、その一部を評価用に残しておき、それ以外のデータを用いて学習を行う。そして学習の途中で時々評価用データとの誤差をチェックし、評価用データとの誤差が増加し始めたらその時点で学習を打ち切るというものである<sup>61</sup>。

## 6. ニューラルネットワークによる時系列処理

時系列データの特徴を把握するという事は、対象としているデータの生成メカニズムを知ることであり、通常同定することからスタートする。同定がうまくいったということはデータの持つ変動パターン、多変量の場合はさらにデータ間の因果関係等を捉えたことになり、ここではじめて予測の可能性が見えてくる。逆にいうならば、予測がうまくいけば対象の構造が把握できたことになる。

そこでニューラルネットワークが時系列データの特徴を把握することが可能か否か、即ち時系列データに対するニューラルネットワークの適用可能性を検討するため、本章ではニューラルネットワークによる時系列データの予測能力について検討する。

まず初めに、ニューラルネットワークで時系列データを扱う際のいくつかのテクニックを論じた後、計算機上で発生させた疑似時系列データの予測を行う。

### 6.1 時系列データを扱う際の工夫

通常の階層型ニューラルネットワークは入力データと教師データとの静的な写像関係を学習するので、そこに時間要素の入る余地はない。そのため動的な写像関係、即ちデータが有する重要な時間相関が無視され、周期的な変動パターンの予測には使えても、実際の複雑な挙動をする予測などに用いるときには、あまり良い結果が期待できないとの報告<sup>7)</sup>がある。

また、株価等の経済指標データに代表される時系列データの予測を行う場合、通常のフィードフォワード型ネットワークに比べ、時間相関をネットワーク内に取り込むための回帰結合を有する、フィードバック型ネットワークの方がより有効であるとの報告<sup>8, 9, 10)</sup>がある。ここで、単にフィードバック型ネットワークといっても、そこには対象・目的に応じ、階層型および相互結合型に大別される種々様々なタイプのネットワーク（例えば、

① 時間遅れを持つフィードフォワード結合を加えたもの<sup>11)</sup>。

② 出力層から入力層へフィードバック結合を加えたもの<sup>12)</sup>。

③ 中間層から入力層へフィードバック結合を加えたもの<sup>13)</sup>等)が提案されている。これらの中で特にここでは、これまでの研究でいくつかの知見を得ていること、既に一般に広く利用されていること、さらにまたその構造が、相互結合型に対し比較的シンプルであること等の理由から、階層型ネットワークに焦点を絞り、フィードバック型ネットワークの有効性等いくつかの検討を行うこととする。

### 6.2 フィードバック型ネットワークの構造

通常、時間変動を示すデータの将来を予測する場合、 $x(t-N) \sim x(t)$ から $x(t+1) \sim x(t+M)$ を推定する機会が多い。この考え方をニューラルネットワークに適用すると、 $x(t-N) \sim x(t)$ を入力とし、 $x(t+1) \sim x(t+M)$ を教師、即ち出力とすることになる。しかしながら多くの場合、特に変動パターンが複雑になるにつれ、 $x(t+M)$ の決定に対し本来最も強く影響を与え、同時にこれを推定する際、最も必要と思われる直前の $x(t+M-1)$ がこの入力データには存在していない。このようなネットワーク構造では、仮にどのような学習手法をとったとしても、精度良く $x(t+M)$ を予測することは困難と思われる。そこで何とかして入力側にも $x(t+1) \sim x(t+M)$ を設定したい。この問題を解決するためには単純に、入力に $x(t+1) \sim x(t+M)$ を追加し、入力を $x(t-N) \sim x(t+M)$ とすればよいことになる。しかしながらこうすることにより、新たな問題が発生する。 $x(t+1) \sim x(t+M)$ があらかじめ分かっている学習期間中は問題ないが、予測期間では、 $x(t+1) \sim x(t+M)$ が必要となるにも関わらず、これらが未知なためネットワークを構成することができない。それでは入力に $x(t+1) \sim x(t+M)$ ではなく、 $x(t+1) \sim x(t+M)$ に対応したネットワークの出力との差（出力誤差）を入力にフィードバックしたらどうであろうか。学習段階では $x(t-N) \sim x(t)$ に加えて $x(t+1) \sim x(t+M)$ に対応する出力誤差が入力されるため、学習の進行に伴い出力はその教師、即ち $x(t+1) \sim x(t+M)$ に近づく。結果として入力にフィードバックされる出力誤差は学習の進行に伴い、徐々にゼロに近づくことになる。最終的に学習終了時には、入力にフィードバックされる出力誤差はゼロとなり、予測段階ではフィードバックが消滅し、(厳密にはフィードバックが消滅するのではなくゼロがフィードバックされ、結果としてフィードバックが意味をなさなくなる)当初の目的を達成可能な都合の良いネットワークが構築できたことになる。

また、これまで議論してきたフィードバック型ネットワークを実際の時系列データに適用する際の注意点として、学習終了時においてもフィードバックされる値がゼロとなることは現実的にはほとんど無く、そのため学習終了時には、フィードバック値を強制的にゼロと置くことになる。

### 6.3 非線形時系列データのフィッティング

それでは実際に具体的な時系列データを対象に、ニューラルネットワークがデータ内の時間的な相関構造をどの程度抽出できるのか、さらに、果たして本当にフィードバック型ネットワークは有効であるのかどうかを調べることとする。

入力データとしては、松葉<sup>8)</sup>を参考に、 $t$ を時刻、 $x_t$ を状態変数とし、異なる2種類の周期関数で駆動される非線形な動的方程式(式(6.1))を採用した。



$$x_{t+1} = \frac{x_t}{1.0 + x_t^2} + \sin \omega_1 t + \sin \omega_2 t \quad (6.1)$$

ここに、 $\omega_1 = \frac{2\pi}{10}$ 、 $\omega_2 = \frac{2\pi}{24}$ である。

また、本方程式を選定した理由としては、作成したプログラムの出力結果を松葉<sup>8)</sup>により紹介されている結果と比較することにより、その動作確認を行うことを主に狙ったためである。但し、この方程式の初期値、およびネットワークの結合荷重行列の初期値が共に不明であるため、本実験結果と完全な一致は得られないものであることをあらかじめ明示しておく。

#### 6.4 学習方法

ネットワークパラメータは極力松葉<sup>8)</sup>が紹介している学習条件と同じになるように設定し、フィードバック型、フィードフォワード型のそれぞれのネットワークにより処理を行った。具体的なネットワークパラメータの設定値は、入力層ニューロン数が20（フィードフォワード型の場合は10）、出力層ニューロン数が10で、中間層のニューロン数を変数Mとして、1から10まで順に大きくした。予測時にはフィードバック結合をしていた10個の入力層ニューロンに対し、強制的にゼロを入力した。学習係数 $\epsilon$ は0.5とし、シグモイド関数の勾配 $\alpha$ は1.0とした。また最大学習繰り返し数を1万回として、この回数内で評価関数値が $10^{-5}$ 以下に達したとき、これを収束と見なして学習を打ち切った。

入力データの提示方法としては、まず、入力データ $x(t)$ を120個用意して、この内 $t = 1 \sim 80$ を学習用データとして、(実際には $t = 61 \sim 70$ が学習用入力データの最後の60組目で、これに対応する $t = 71 \sim 80$ が60組目の教師データとなる。従って最大60組の入力パターンが設定可能となる)残りの $t = 81 \sim 120$ を予測した。また、学習用データの入力方法は、図6.1に示したように、 $T=t \sim t+10$

入力データ

教師データ

x01, x02, ..., x09, x10, x11, x12, ..., x19, x20  
x02, x03, ..., x10, x11, x12, x13, ..., x20, x21

x60, x61, ..., x68, x69, x70, x71, ..., x78, x79  
x61, x62, ..., x69, x70, x71, x72, ..., x79, x80

図6.1 学習データ

を1組として、これから $T=t+1 \sim t+11$ ,  $T=t+2 \sim t+12$ のように1時刻ずつずらした計60組を用いることとした。

#### 6.5 入力データの特徴

ネットワークに提示して、学習および予測を行う入力データの時系列を図6.2示す。図中横軸の80を境として、前半が学習用データ、後半が予測用データである。式(6.1)で示すように、この関数は異なる2種類の周期関数を含んでいることに起因して、全般に渡り約25程度の大きな周期の上に約10程度の周期が重畳した形で変動しているようすをこの図から読みとることができる。しかしながら、その時間的な振幅の変動パターンは非常に不規則なものとなっている。そのため、波形上のあるピークを基準に、次に訪れる波形の山、あるいは谷の高さ、深さを予測することは大変困難である。この様に非線形な振る舞いをするデータに対し、その関数系があらかじめ分かっている一般のケースにおいて、通常の統計モデルで行われる線形結合では、精度良く将来を予測することはできない場合が多い。この様なとき、特にモデルを設定することなく、任意の関数を近似可能なニューラルネットワークの強力な関数近似能力に期待が高まる。

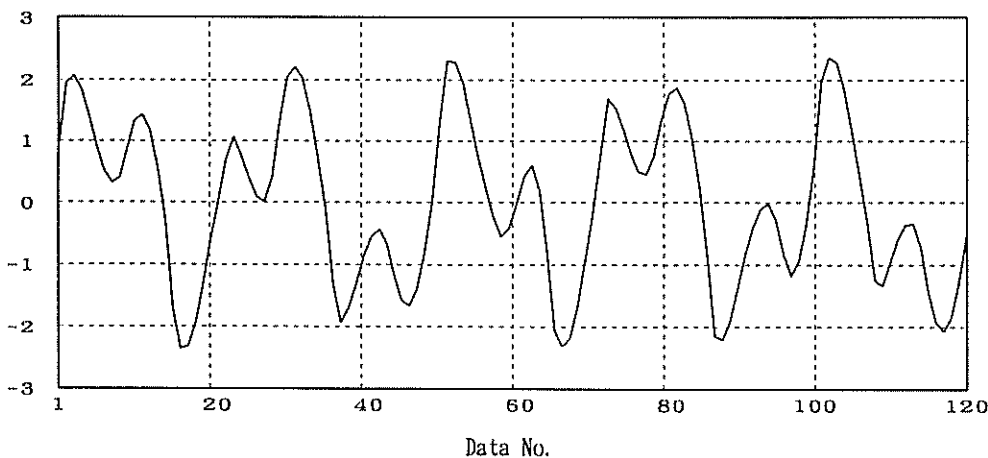


図6.2 ネットワークに提示した入力時系列

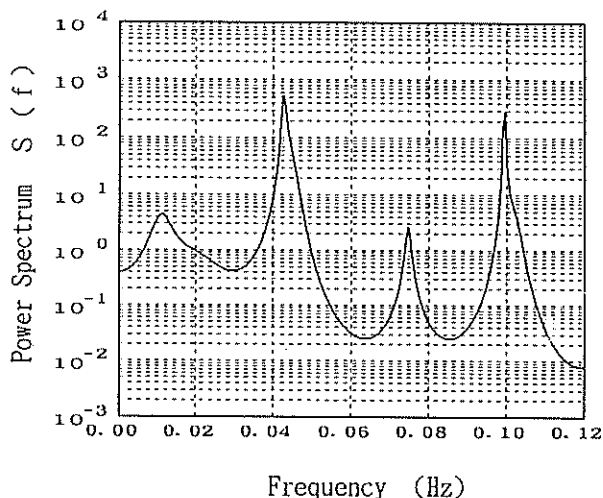


図6.3 入力時系列のスペクトル

図6.3は、図6.2で示した時系列データのより詳細な変動特性を調べるために、MEMで求めたスペクトルである。図6.2で確認したように、式(6.1)の2つの周期関数の変動周期に対応する2つのピークが、明瞭に表れている。さらに、この2つのピーク以外にも、2種類の周期関数が合成された結果生じたものと思われる、他のピークが確認でき、全体としては複雑なスペクトル構造を形成していると言えよう。これらの結果は、ここで対象としている時系列が、複数の周波数成分から構成された見た目ほど単調ではない、複雑な変動パターンを有するものであることを教えている。

### 6.6 予測結果

図6.4は中間層ニューロン数を変化させたときのフィードバック型ニューラルネットワークによる出力結果を示したものである。図中細線が式(6.1)で与えられるモデルからの出力(目標関数)、太線がネットワークによる出力である予測値をそれぞれ表している。ここで確認しておくが、学習データは連続した10個のデータを入力データに、またこれに続く10個のデータを教師データとし、これらを1組と数え、1時刻ずつずらした計60組のデータを用いている。この図より、中間層ニューロン数が4~5個以上になると、位相、振幅共にほぼ目標関数を予測できている様子が分かる。即ちニューラルネットワークが対象時系列の特徴を抽出できたことになる。

### 6.7 入力データセットの低減化に伴う予測結果

前節において、60組の全入力データセットを用いた場合、ニューラルネットワークはほぼ目標関数を予測することができた。そこで、さらに少ないデータセットを用いて学習した場合、どのような出力が得られるのかを検討する。

入力データセットの低減化によるニューラルネットワ

ークの出力を調べることの最大の目的は、入力データセットの数を減らすことによって、学習に要する計算時間を短縮し、学習を効率的に行わせることである。加えて、これまでの経験によると、入力データの数に比例してネットワークの出力である予測精度が向上するとは必ずしも限らず、間雲に入力データ数を増やした結果、予測精度が低下したというケースも少なくない。そこにはやはり、学習を効果的に行うために必要なデータ数というものが存在するようである。大切なことは、対象の特徴を抽出するために、必要最小限のデータをネットワークに提示することである。しかしながら実際問題を扱う際、この最適なデータ数を知ることは、大変困難である場合が多く、結局のところ準備したいいくつかのケースにおいて試行錯誤的な方法をとらざるを得ない。そこで以下に、その試行錯誤的に行った実験の一部を紹介する。

具体的には、前節で作成した60組のデータセットの内、前半の40組およびさらに減少させた20組を入力データセットとして目標関数の予測を行う。学習条件は、入力データにおける組数の違いを除けば、60組の場合と全く同様である。

図6.5は入力データセットを40組とした時のネットワークの出力結果を示したものである。図6.4と同様に細線が目標関数、太線がネットワークによる出力である予測値をそれぞれ表している。これによると入力データセットを40組に減少させたにも関わらず、中間層ニューロン数が6以上の場合においては、図6.4のそれらと比べ、同程度もしくはそれ以上の精度で目標関数を予測できていることがわかる。相違点としては、図6.4では中間層ニューロン数が4~5個以上になると、ほぼ目標関数を予測できているのに対し、図6.5では中間層ニューロン数が6個以上にならないと予測値と目標関数との誤差が大きく、目標関数を再現できていないと言えない点である。このことは、入力データを減少させたことにより、そのデータ内から特徴を抽出するためにネットワークがより多くの中間層ニューロンを必要としたものと思われる。

図6.6は入力データセットを20組とさらに減少させた時の出力結果を示したものである。線種の意味および学習条件は特に言及しない限り上述の通りである。入力データセットが20組になると、何とか目標関数の変動傾向は捉えているものの、中間層ニューロン数が最大の10個でさえ、もはや目標関数を予測できていない。即ち、この入力データでは、目標関数の特徴をネットワークが抽出できなかったことになる。

以上の結果をまとめると、目標関数を精度良く予測するためには、入力データセットが概ね40組以上、中間層ニューロン数が6個以上それぞれ必要ということが分かった。これに従い、主に中間層ニューロン数が5個以上のケースに絞って以降の検討を行うこととする。

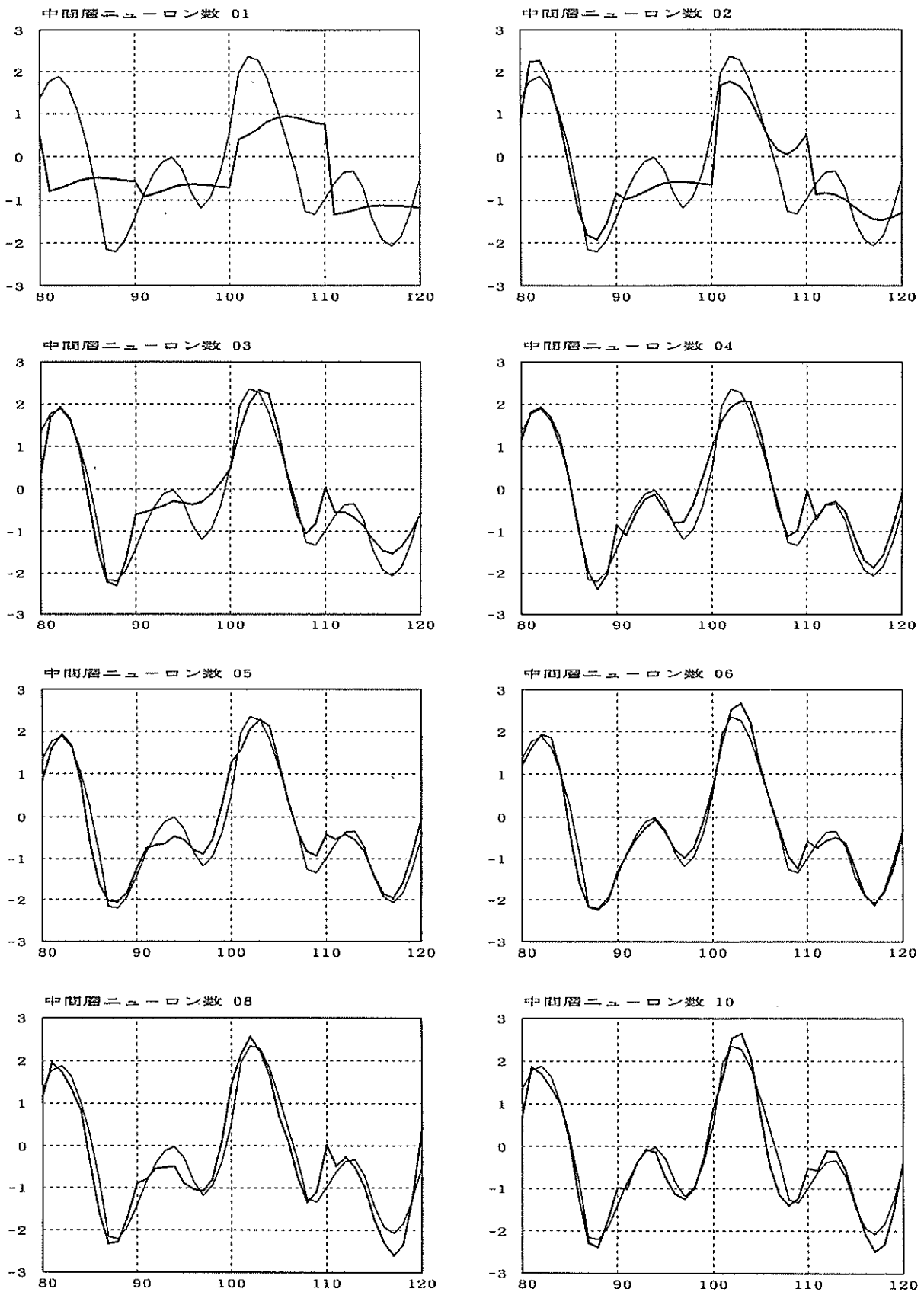


図6.4 入力データセットを60組としたときの各中間層ニューロン数による予測結果

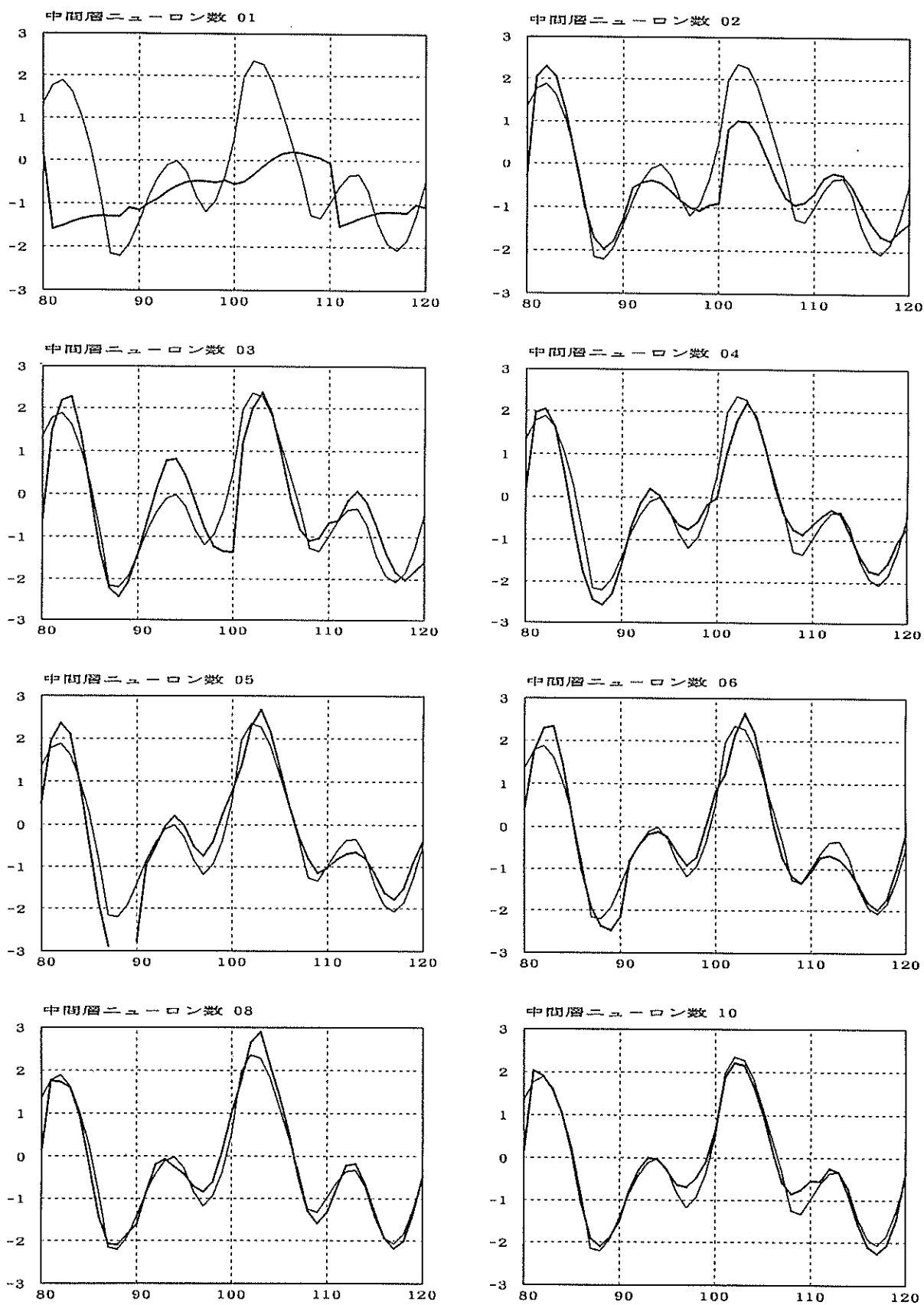


図6.5 入力データセットを40組としたときの各中間層ニューロン数による予測結果

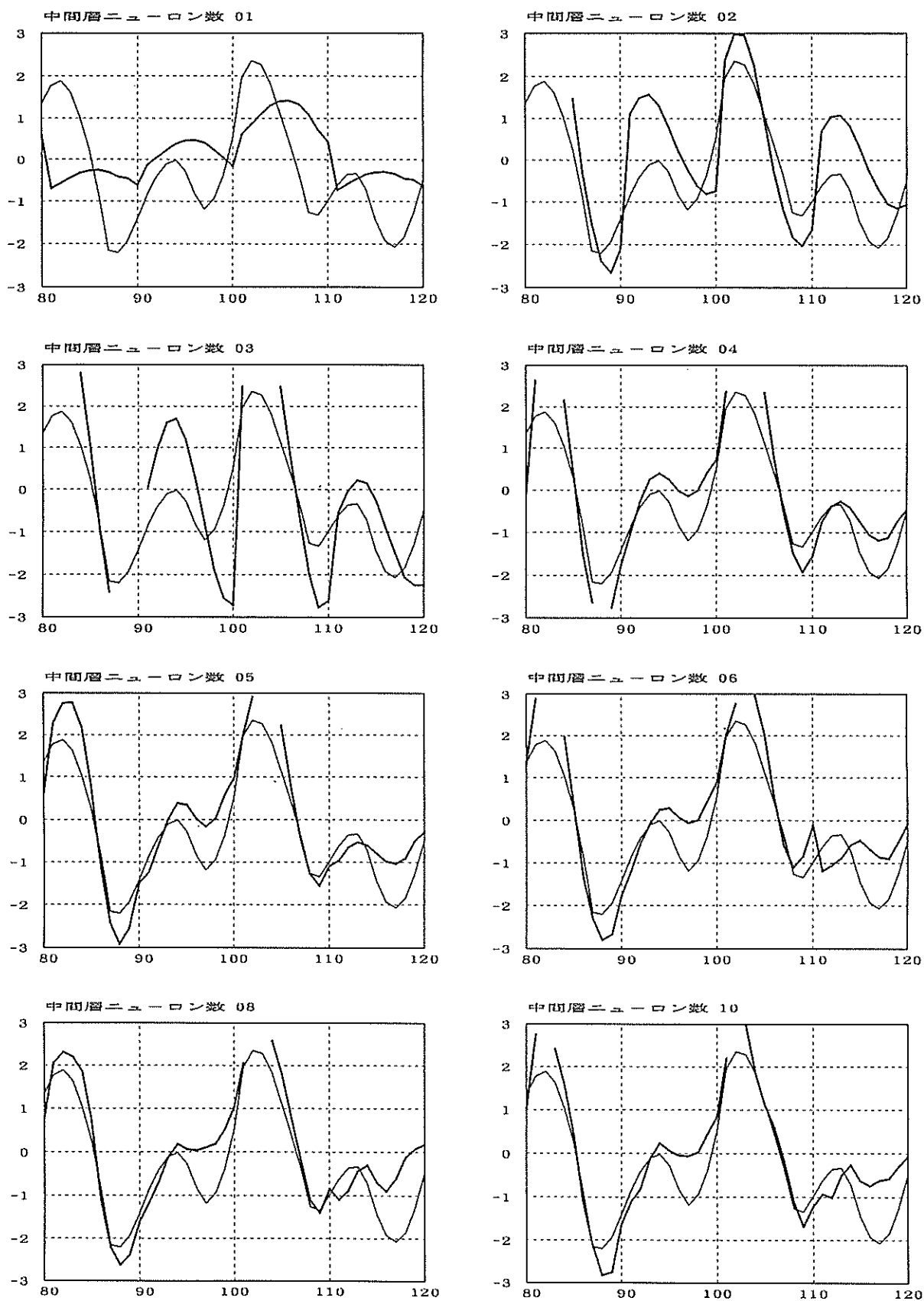


図6.6 入力データセットを20組としたときの各中間層ニューロン数による予測結果

### 6.8 予測精度向上のための工夫

前節の結果によると、入力データセットが20組の場合では、中間層ニューロン数に関係なく、目標関数を予測することはできなかった。その理由は単に入力データセットが少な過ぎたためなのであろうか。ここで、入力されるデータセットの構成について再度確認しておく。1組のデータセットは連続した10個のデータを入力データ、さらにこれに続く10個のデータを教師データとする計20個のデータから構成されている。そしてこのデータセットを僅か1時刻ずらしたものが次のデータセットとなる。そのため20組のデータセットとは、合計200個の入力データが存在しているにも係わらず、そのほとんどは他のデータセットと重複していることになる。実際200個の入力データの内、独立したデータはたかだか30個である。直感的に判断しても、30個の入力データからその変動パターンを獲得し、さらにその将来を予測することなど、本質的に無理があるように思われる。やはり予測精度を向上させるためには、予測期間直前までのデータを用いて、その変動パターンを学習するべきであろう。

それではどのようなデータセットをネットワークに提示することが、より良い予測結果を得ることに繋がるのであろうか。この問題を検討するため新たに、学習期間全体を網羅しつつ、組数を最少とするデータセットを考える。この条件を満足するデータセットの構成は図6.7に示すように、他の組との重複データをなくすことである。このように作成した新たなデータセットは、僅か7組のデータセットで構成されることになる。

入力データ	教師データ
x01, x02, ..., x09, x10	x11, x12, ..., x19, x20
x11, x12, ..., x19, x20	x21, x22, ..., x29, x30
...	...
x51, x52, ..., x59, x60	x61, x62, ..., x69, x70
x61, x62, ..., x69, x70	x71, x72, ..., x79, x80

図6.7 学習データ

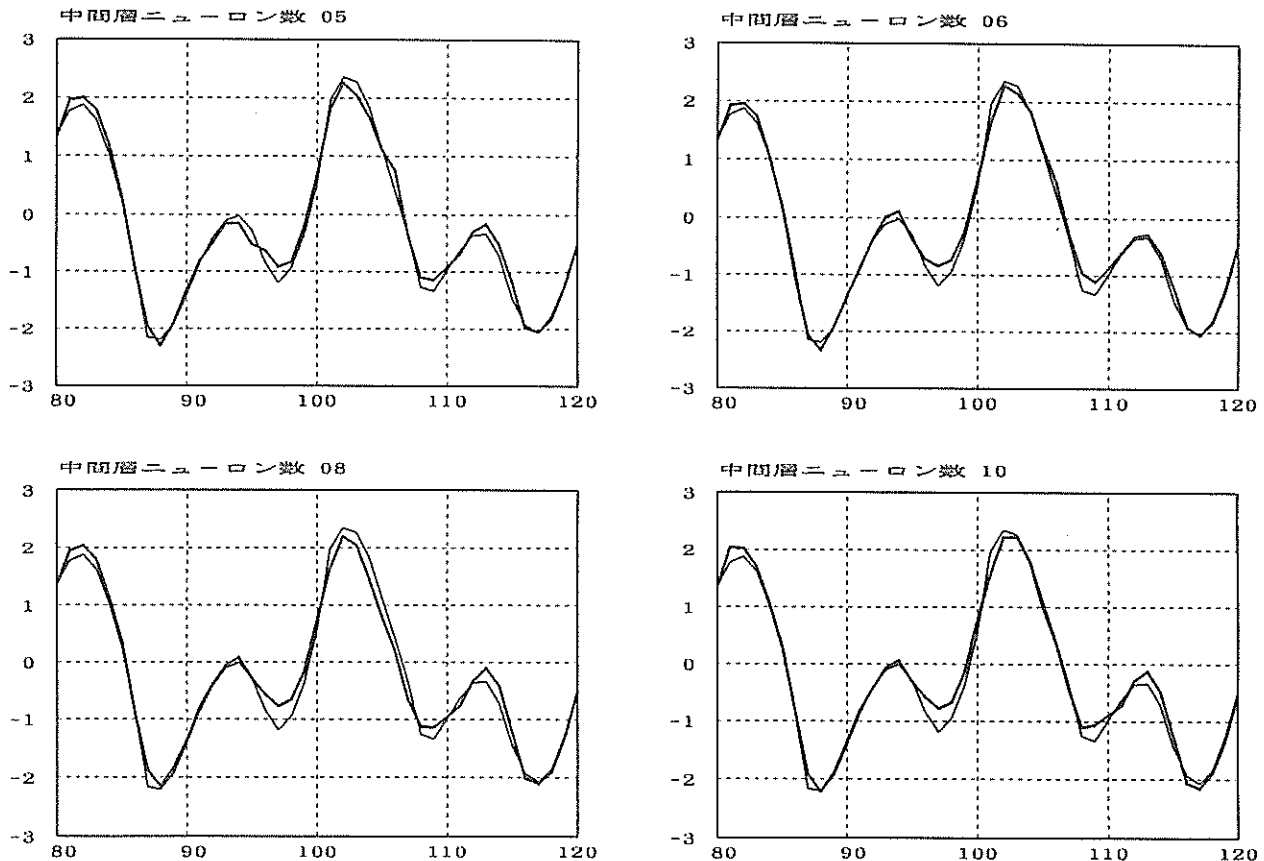


図6.8 重複するデータを除いた7組の入力データセットとしたときの各中間層ニューロン数による予測結果

図6.8は上記のように新たに作成した7組の入力データセットを用いて学習を行ったネットワークの出力結果を示したものである。これによると、本図を拡大して見なければ予測結果と目標関数の違いが分からないほど両者の一致度はよく、ほぼ完全に目標関数が予測されていると言えよう。即ちニューラルネットワークが対象時系列のダイナミクスをほぼ完全に抽出・獲得することに成功したことになる。また、その予測精度の高さは定量的な比較をするまでもなく、これまでに行った計4ケースの中で最高である。さらにここで注目すべきことは、学習に用いた入力データセットが最少の7組でありながら、最高の予測精度を得ることができたという事実である。この事実は、ネットワークが効果的に学習するためには、対象とする現象を説明し得る要点を、簡潔にまとめたものを入力データとして、ネットワークに伝えることが重要であると教えているように思われる。別の言い方をすれば、我々人間がそうであるように、何も知らないネットワークにある約束を覚えさせる時、ネットワークが何が大切であるのか困惑するほど細かな大量の情報を教え込むのではなく、大切なことを手短かに教えることが重要であるということではないだろうか。

#### 6.9 フィードフォワード型ネットワークとの比較

これまで行ってきたいくつかの予測結果に対する検討は、全てフィードバック型ネットワークによるものであった。そこで本節では、時系列データを対象に、ニューラルネットワークで処理する際、通常のフィードフォワード型ネットワークに対し、はたして本当にフィードバック型ネットワークの方が、より有効であるのかといった問題を明らかにする。手法としては、各ケースにおける両者の予測結果および学習時における予測誤差を中心に、比較検討を行う。

図6.9は各入力データセットに対し中間層ニューロン数を変化させた際のフィードバック型およびフィードフォワード型ネットワークそれぞれによる式(3.4)で示した学習データの評価関数値を示したものである。フィードバックの有無を除けば両者の学習条件は全く同様である。また図中、実線がフィードバック型、点線がフィードフォワード型ネットワークによる評価関数値をそれぞれ意味している。これによると重複データをなくした7組の入力データセットを用いた場合を除き、他のほとんどのケースにおいてフィードバック型ネットワークの方がより低い評価関数値を示し、これより同ネットワークの優位性をうかがうことができる。一方重複データをなくした7組の入力データセットを用いた場合、両ネットワークによる評価関数値は中間層ニューロン数によらず同程度であり、フィードバック型ネットワークの明かな優位性は認められない。この理由としては、ここで与えた入力データがフィードフォワード型ネットワークでも

対象の特徴を抽出可能とする極めて良好なデータであり、そのため両ネットワークによる評価関数値に差が生じなかったものと思われる。しかしながら実際には、対象の特徴を十分に説明し得る入力データを知ることは非常に困難であり、それだけでも多大な時間と労力を費やす場合が少なくない。言葉を換えれば、入力データが不十分である場合が多いということである。そういった意味では入力データが不十分なケースとして、例えばここで示した入力データセットを40組あるいは60組とした時、フィードバック型ネットワークの方がより低い評価関数値を示したという結果は、実際の問題を扱う際大変都合が良く、同時に出力結果に対するより高い信頼性を期待することができる。

図6.10は上記図6.9の学習結果を利用して、フィードバック型およびフィードフォワード型ネットワークそれぞれから出力された予測結果を示したものである。図の見方としては、上段から20組、40組、60組、そして最下段が重複データをなくした7組の入力データセットをネットワークに提示した時の予測結果を示し、各段共に上側がフィードバック型ネットワーク、下側がフィードフォワード型ネットワークによるものである。また計3列の各図は、良好な予測結果が得られたケースとして、左から中間層ニューロン数が6個、8個、10個の場合を意味している。

この図からまず分かることは、一目して分かるほどフィードフォワード型およびフィードバック型ネットワークそれぞれによる予測結果の間に大きな差が見られないことである。別の言い方をすれば、フィードフォワード型ネットワークによる予測結果も大局的に見れば十分な精度を有していると言えよう。特に、下段に示したように、入力データセットとして重複データをなくした7組の入力データセットを用いた場合、フィードフォワード型ネットワークも良好な予測結果を示し、結果的にフィードバック型ネットワークの特筆すべき優位性は認められない。しかしながら、例えば、入力データセット40組の中間層ニューロン数8個、10個あるいは入力データセット60組の中間層ニューロン数10個のケースにおいて、両ネットワークによる予測結果を注意深く調べると、大変興味深いことに、フィードフォワード型ネットワークによる予測結果に部分的にはあるが位相のズレを確認することができる。即ち、フィードフォワード型ネットワークを利用する際、本来可能な位相パターンの獲得が、入力データの与え方によっては困難となる場合が生じる可能性を示している。一方、フィードバック型ネットワークによる予測結果にはそのような位相ズレは生じておらず、対象の有する時間的な変動パターンを捉えていることが再確認できる。図6.11はこの様子をさらに詳しく調べるために、図6.10の波形上ではほぼ同一の変動を示した7組の入力データセットを用いた際のフィード

バック型およびフィードフォワード型の両ネットワークによる出力をMEMによりスペクトル解析した結果の一例である。これによると、波形上ではほぼ同一に見える両ネットワークの出力結果も、それらのスペクトルを比較すると、明らかにフィードバック型ネットワークにより得られた出力の方が対象のスペクトル構造を正確に再現している様子が確認できる。同時にこの事実は、未知なるノイズの混入が予想される実際のデータに適用する際、対象のスペクトル構造まで獲得可能なフィードバック型ネットワークが有する強力なフィッティング能力の有効性を示すものと思われる。

一般に、複雑な波形のフィッティングあるいは予測を行い、その良し悪しを判断する際、その判断基準として振幅の絶対値ではなく、むしろ波形の位相、即ち上昇するのか下降するのかといった変動傾向を重要視する場合が少なくない。このような視点では、上で示したように、目標関数と予測結果における位相の一致度といった点で、フィードバック型ネットワークによる予測結果の方が優れているように思われる。

それでは、位相の一致はそれほど問題ではなく、例えば波形のピークといった振幅の絶対値を重要視する場合はどちらのネットワークがより有効となるのであろうか。

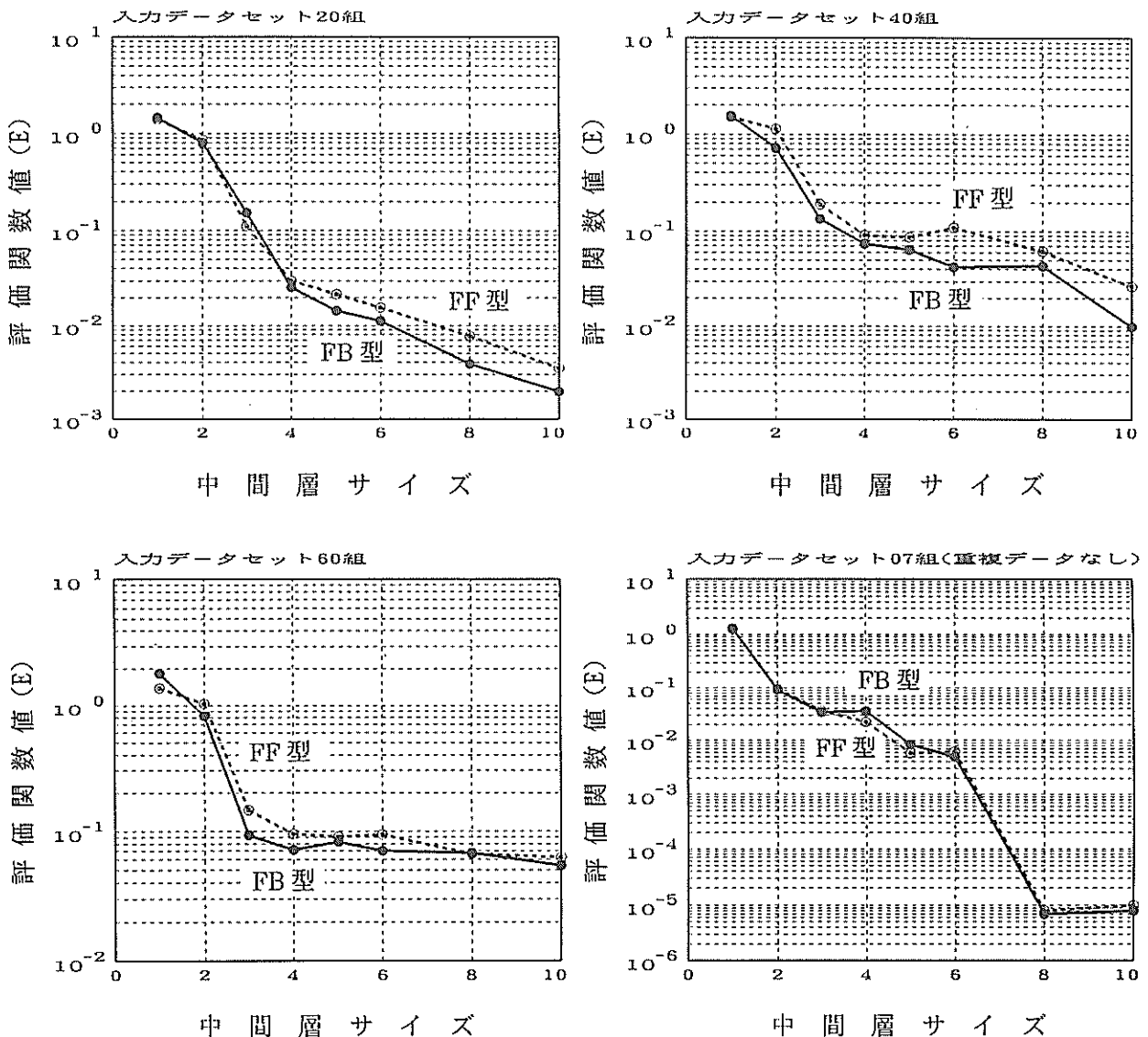


図6.9 中間層ニューロン数の変化に伴うフィードバック型およびフィードフォワード型ネットワークの評価関数値



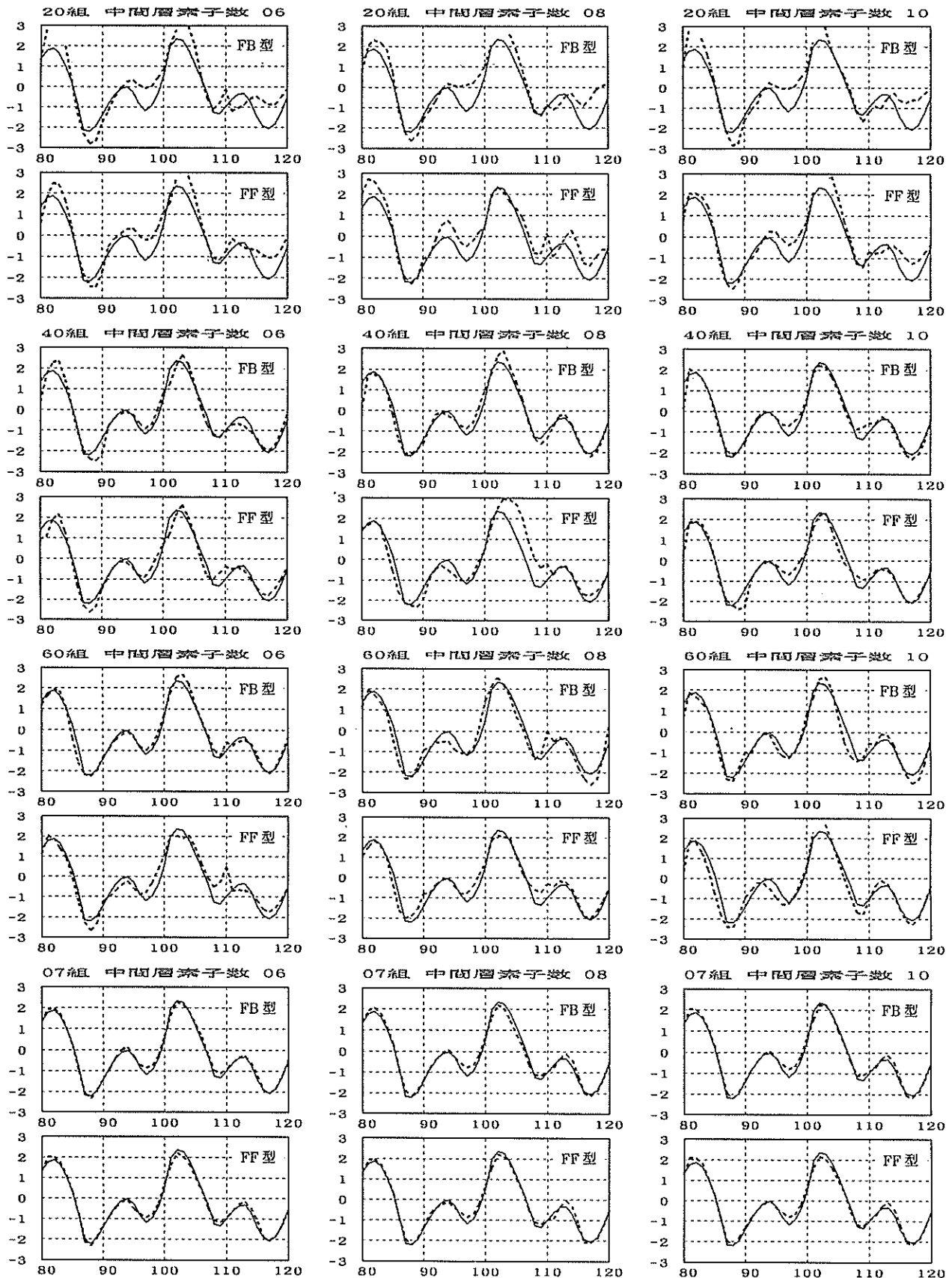


図6.10 入力データセットおよび中間層ニューロン数を変化させたときの  
フィードバック型およびフィードフォワード型ネットワークによる予測結果

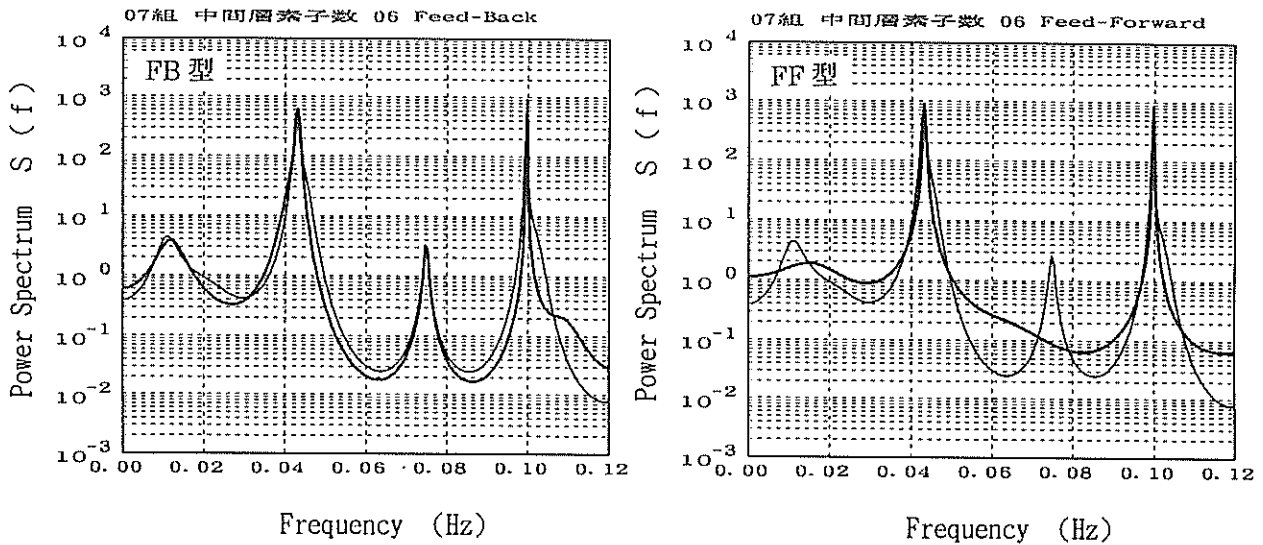


図6.11 フィードバック型およびフィードフォワード型ネットワークによる予測結果に対するスペクトル

ここで入力データセットが20組の場合に着目すると、今対象としている目標関数の振幅に関する一致度では、図6.10に示すようにフィードフォワード型ネットワークの方が、フィードバック型ネットワークより予測精度が高いように見える。そこでこの振幅に関する両ネットワークの予測精度を定量的に把握するため、予測期間の各時刻における目標関数と予測値の差の自乗和を図6.12に示す。入力データセットが20組の場合を除けば、図6.9で示した各ケースにおいて、両ネットワーク間の学習誤差の差が、これらに対応するそれぞれの予測誤差に良く反映されていることが分かる。またこのように位相を考慮せず、単に振幅で比較しても、最小の予測誤差はいずれもフィードバック型ネットワークにより得られており、この結果は同ネットワークの優位性を示すものである。一方、入力データセットが20組の場合に着目すると、他の入力データセットの場合と異なり、特徴的にフィードバック型ネットワークによる予測誤差の方が大きくなっている。この結果は、ある程度最適な入力データが与えられている間は、フィードバック型ネットワークによる予測結果の方がその精度は高いことを示している。しかし入力データを変化させ、最適な情報量からの差が大きくなるに従って、両者の差は縮まり、やがて精度の低下は認められるものの、相対的にフィードフォワード型ネットワークの方が予測誤差は小さくなる。即ち、上の結果はフィードフォワード型ネットワークの方が、入力データの変化に対しロバストであることを示すものであると思われる。

これらの結果から、ニューラルネットワークの使用に関し、極端な言い方をすれば、当たらずとも遠から

ず的な安全性の高い出力を期待するならばフィードフォワード型ネットワークを利用し、振幅位相共に高精度な出力を要求するならばフィードバック型ネットワークを利用することが賢明であるように思われる。但し後者の場合、データの提示方法等、その学習方法については十分な注意が必要である。

また、両ネットワークによる出力特性の比較検討を目的としている本節の論点からは多少逸脱するが、学習誤差と予測誤差を比較することにより、両者の関係について検討を行う。

まず入力データセットが20組の場合、図6.9では、特に中間層ニューロン数4以上の時、フィードバック型ネットワークによる学習誤差の方が一様に小さいにも関わらず、図6.12では上述したようにフィードバック型ネットワークによる予測誤差の方が大きい。つまり、両ネットワークにおける学習誤差と予測誤差とはその大小関係が互いに逆の関係になっていることが分かる。加えて入力データセットが7組の場合、各中間層ニューロン数に対する学習誤差と予測誤差の変動パターンは大きく異なっており、明らかな関連性は認められない。以上これらの結果は、中間層ニューロン数を変化させるなどにより得られた小さな学習誤差が、予測精度の向上には必ずしも結びつくものではないことを示唆し、同時に学習誤差低減化は、あくまで学習の一手法であることを具体的に認識させる一例である。

以上本節の結果をまとめると、予測結果を精度良く得るための学習に必要なことは、適切なネットワーク構造

および学習パラメータを決定した上で、対象の特徴を十分抽出可能な入力データをネットワークに提示することである。即ち、学習誤差の低減化に神経を注ぐのではなく、如何に大切な情報を如何に簡潔に与えるかがキーポイントのようである。そして適切な入力データがネットワークに与えられた時、その出力はネットワークの構造に対する依存度が弱まり、結果として適当なネットワークを構築するために要する時間と労力を抑えることが可能となる。

また現時点では、その検討が不十分ではあるものの、非常に興味深いニューラルネットワークの性質に関するいくつかの知見を得ることができた。その知見とは、ニューラルネットワークが良好な出力を可能とする入力データの範囲とネットワーク構造の関係についてである。具体的には、フィードフォワード型ネットワークの方が入力データに含まれる情報の低下あるいは増加に対する反応が鈍く、その結果、より広い範囲の入力データに対し、ある程度の予測精度を保つことが出来るということ

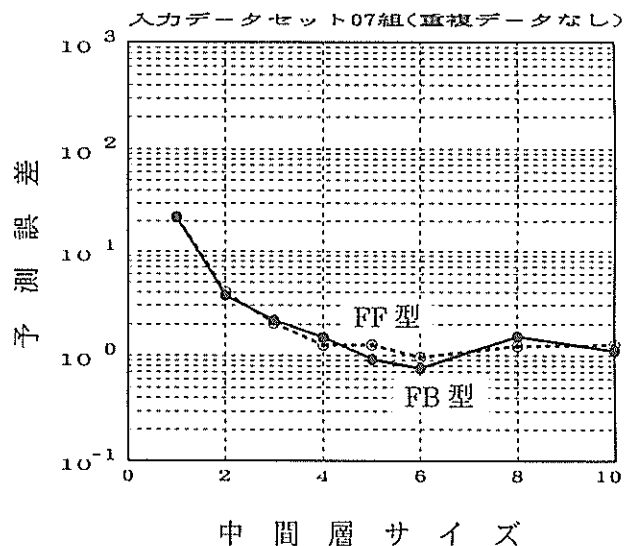
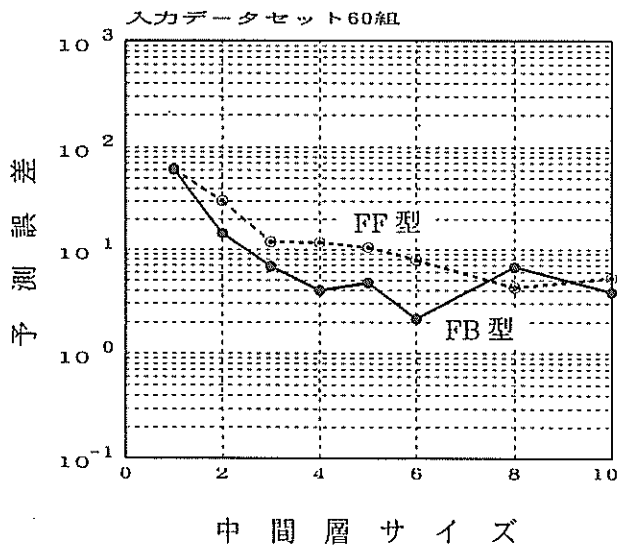
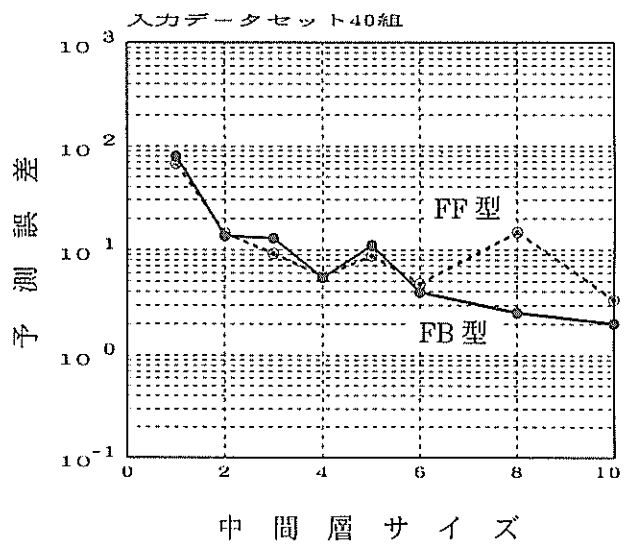
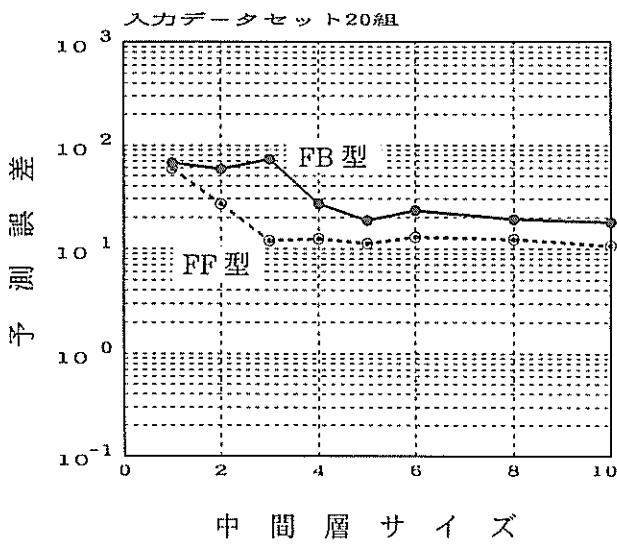


図6.12 中間層ニューロン数の変化に伴うフィードバック型およびフィードフォワード型ネットワークによる予測誤差

である。これに対しフィードバック型ネットワークは、入力データの変化に敏感に反応し、良好な予測精度を確保可能な入力データの範囲が狭い。ただし、入力データの最適度を増加させ、ひとたび入力データがフィードバック型ネットワークの守備範囲圏内に進入すると、この状況は逆転し、フィードバック型ネットワークは高精度な予測結果を出力するようになる。特にこの場合、通常困難な対象の位相構造が獲得可能となり、例えフィードフォワード型ネットワークが位相ズレを起こしているケースにおいても、そのような位相ズレは発生せず、安定して高い予測精度が維持される。解釈を容易にするため別の表現をするならば、フィードフォワード型はオールマイティーな虫メガネ、フィードバック型は狭い範囲ではあるが微細な構造まで映し出す顕微鏡に対応するよう思われる。

#### 6.10 入力層ニューロン数の決定法

通常、階層型ニューラルネットワークを利用することの目的は、パターン分類および目的関数のフィッティングに大別される。分類作業を行う場合、入力項目の種類および分類する数により、入出力層のニューロン数は必然的に決定される。しかしながら時系列データを扱う場合、入出力層のニューロン数は対象時系列のデータに内在する周期性等、そのダイナミクスにより本来決定されるべきものであるが、確立された決定法が未だ存在していない。そのため実際のところ、入出力層のニューロン数は使用者の判断に委ねられることになる。しかもこのように決められたネットワークサイズにより、その出力である予測結果は大きく変わることが予想されるため、その決定にはある程度の経験が必要となる。実際にニューラルネットワークを動作させる際、予め決定しなければならぬネットワーク構造、あるいは学習パラメータの自由度は大変大きく、例え完全ではないにしろ、これらの値の目安を知る方法があるならば、それは時間と労力を大幅に節約可能なものとする。そこで本節以降では、例えば回帰モデルで対象時系列の次数決定にしばしば用いられるAIC、MDL等の情報量基準に対応する統計的手法として、いくつかの報告書<sup>11,12)</sup>でその有効性が確認されているフラクタル次元を導入することにより、過去どれだけのデータから将来何時刻先までを予測すべきであるか、即ち入出力層ニューロン数の決定法について検討する。

これまで時系列データを対象として行ったフィードバック型およびフィードフォワード型ネットワークによる出力特性、あるいはまた、入力データセットの構成や中間層ニューロン数の変化に伴う予測精度等の検討は、いずれも入出力層のニューロン数を10に固定している。この10という数値は根拠無く選択したものではなく、上述した対象時系列のフラクタル次元から求めたものである。

#### 6.11 フラクタル次元

フラクタル次元とは図形の持つ複雑さを表す指標であり、その基本的な考え方は自己相似性の定量化である<sup>14)</sup>。例えば地図上のある複雑な海岸線の距離を大きさの異なる物差しで測る場合を考える。海岸線の測定に使用する物差しの長さを大きいものから小さくするに従い、測定される海岸線の距離は徐々に長くなる。また海岸線の形状が複雑になればなるほどその距離は物差しの長さ強く依存するようになる<sup>15)</sup>。これを時系列データに置き換えると、海岸線が対象時系列の波形に、物差しの長さが時系列データのサンプリング間隔にそれぞれ対応する。図6.13はこの様子を示したもので、サンプリング間隔の増加に伴う波形の単調化が確認できる。言うまでもなく、仮にサンプリング間隔20の波形が原データであったならば、ここで示した全ケースとも同一の波形となる。このように曲線上のある区間を測定する場合、この距離を測る物差しの長さを変化させた時、これに応じてどのような割合で対象区間の距離が変化するかといった、この度合いに対応するものがフラクタル次元である。またフラクタル次元の特徴として、通常正の整数値として表現される次元に対し、例えば1.5次元というような連続した実数値を許容している点が挙げられる。これによりこれまで縦横の2次元で表されていた曲線データの次元数を、その複雑さの度合いにより、1.0~2.0次元の範囲で表現できることになる。これら値の意味するところは、対象時系列データの時間変化が直線的であればその次元は1.0となり、カオスのように全くランダムであれば2.0となる。したがって、次元が2.0に近付くほどその時間変化がランダムとなり予測が困難となる。

図6.14はこれまで対象としてきた図6.2に示す時系列データが、どれくらいのタイムスケールでみた時、どのような変動傾向を示すものかを調べるために、各サンプリング間隔に対する変動の大きさを表したもので、この勾配がフラクタル次元の値(D)となる。これによると、およそサンプリング間隔4および10付近を境にして3種類のフラクタル次元の存在を確認することができる。具体的には、 $1 \leq t \leq 3$ の時 $D=1.11$ 、 $4 \leq t \leq 10$ の時 $D=1.76$ 、そして $t \geq 10$ の時 $D=1.99$ であった。この結果は、時間間隔10以上でサンプリングしたものは、そのフラクタル次元が2.0に近いことからほぼランダムであり、10時刻以上先の予測を行うことは極めて困難であることを示している。一方、時間間隔10以下でサンプリングしたものは、そのフラクタル次元が1.0と2.0の間にあることから、対象時系列にある程度の時間相関が内在していることを示し、ニューラルネットワークによるこの変動特性の抽出可能性を示唆している。

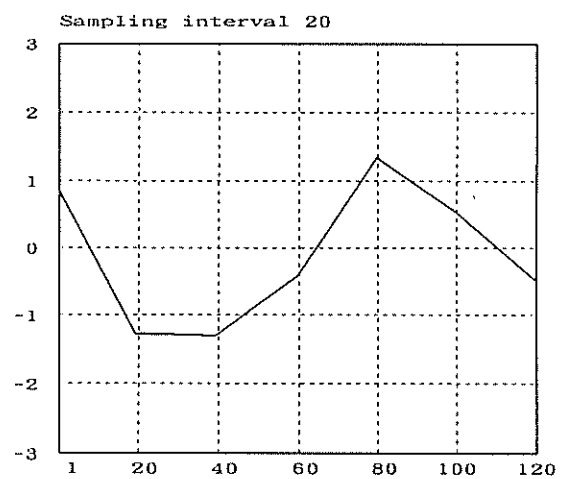
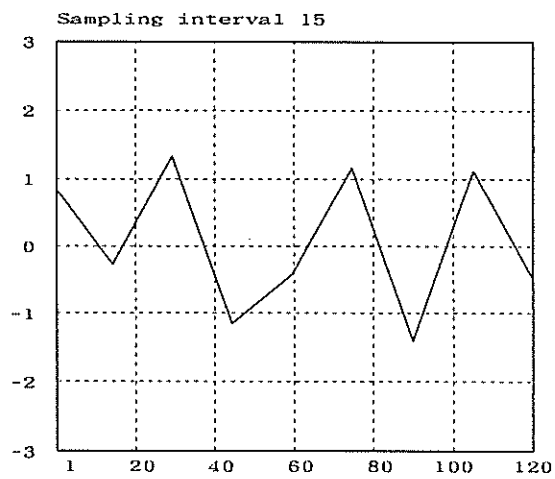
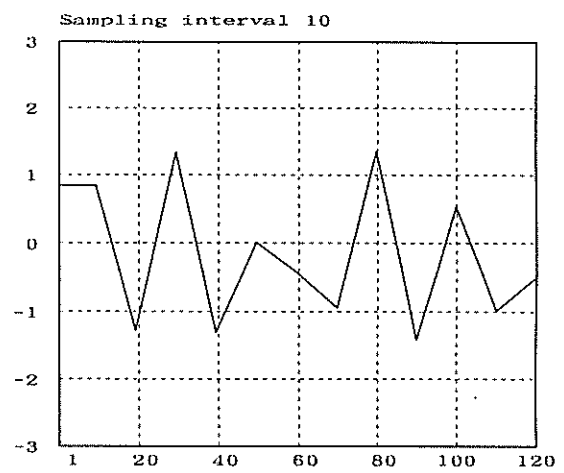
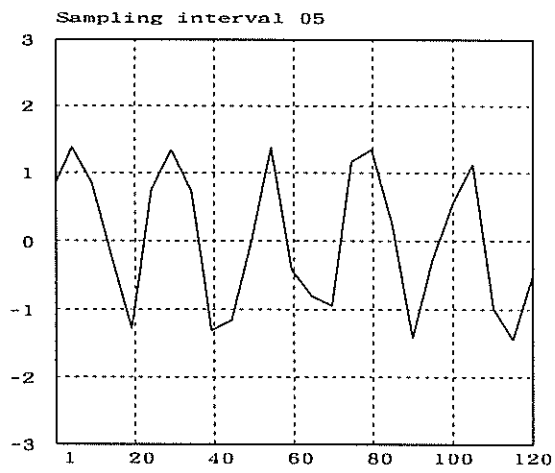
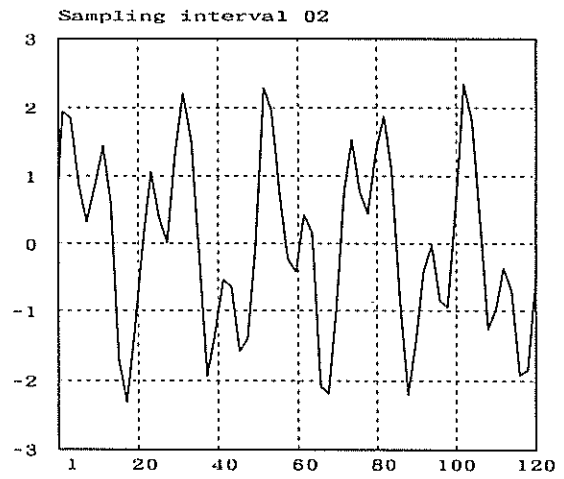
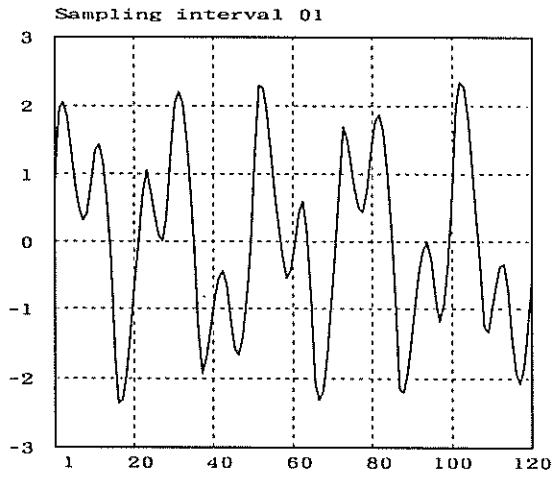


図6.13 サンプリング間隔の変化に伴う入力データ波形

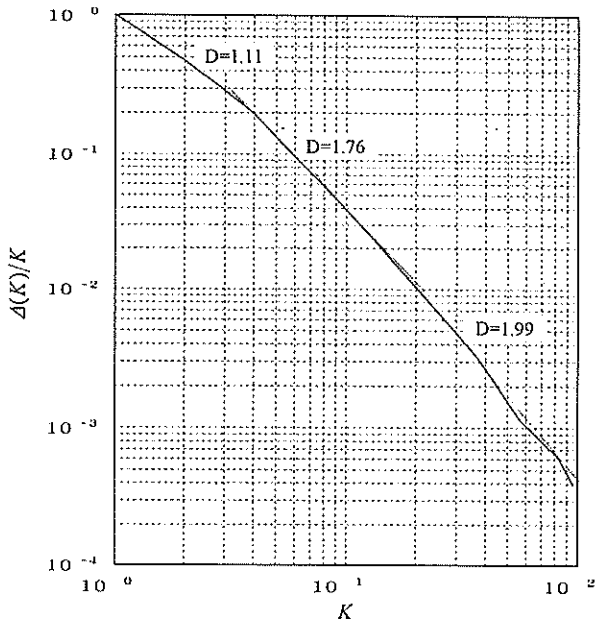


図6.14 入力データのフラクタル次元  
(K: サンプル間隔,  $\Delta(K)$ : 折れ線の長さ)

例えば自然現象等のある時系列データに対し、その全体を眺めた時とある一部分に着目した時とでは、波形の複雑さが異なるというのは珍しいことではなく、むしろ一般的である。このようなケースにおいて将来を予測しようとする場合に、どれ位先まで予測可能であるのか、その妥当性を考えた時、波形の変動パターンが比較的単調な期間を予測の対象期間とすることは自然な考え方である。そのような意味で、予測対象期間の決定に対し、図形の複雑さを定量的に表現するフラクタル次元を導入することは、我々人間の思考と良く合致しているように思われ、直感的にも理解し易い。

以上の議論に従い、本章では入出力層のニューロン数を10に決定した。

### 6.12 予測期間の違いによる出力特性

前節では、ある時刻からどれ位先まで予測すべきであるか、即ち入出力層のサイズを決定する際、その手段としてフラクタル次元を導入し、これまで対象としてきた時系列データに対しては、入出力層のニューロン数を10（ここではフィードバック型ネットワークを使用するため、実際の入力層ニューロン数は入力データ+フィードバックデータ、即ち20となる）とすることが妥当であると議論した。しかしながらそれらは定量的な議論を与えるものではなかった。そこで本節では、入出力層のニューロン数として、はたして本当に10が妥当であるのかどうか、この問題について検討を行う。

図6.15は入力層ニューロン数が異なる4種類のフィードバック型ネットワークそれぞれにより、得られた予測結果の一例として、中間層ニューロン数を6とした場合

のものである。なおこれらの学習条件は全て同一である。ここで与えた具体的な入力層ニューロン数は、本節の目的を考慮して10、およびこの前後として2、5、および20とした。これによると、入力層のニューロン数を10としたネットワークによる予測結果が、明らかに他の3ケースよりその予測精度が高く、最適な入力層のニューロン数はフラクタル次元が示した10であることを物語っている。

表6.1 各ネットワークサイズによる予測誤差

入力層	中間層	出力層	誤差の自乗和
2	6	2	7.90112
5	6	5	9.93988
10	6	10	0.76225
20	6	20	39.13467
2	8	2	8.52272
5	8	5	14.97006
10	8	10	1.50935
20	8	20	66.58823
2	10	2	8.38988
5	10	5	11.23105
10	10	10	1.10510
20	10	20	53.66166

表6.1は図6.15に示した中間層ニューロン数を6とした場合に加え、中間層ニューロン数を8および10としたそれぞれの場合における目的関数と予測結果に関する誤差の自乗和を示したものである。中間層ニューロン数を変化させた3ケースいずれの場合においても、入力層のニューロン数が10の場合に誤差の自乗和が最小になっていることが分かる。

この結果、やはり最適な入力層のニューロン数は10であり、時系列データを対象とする場合、入力層のサイズを決定する手段として、松葉<sup>1)</sup>あるいは船橋<sup>1)</sup>が示しているように、フラクタル次元を利用することは大変有効であることが確認できた。

以上の結果が示すことは、中間層同様、入力層においても最適なニューロン数が存在するということである。入力層のサイズを必要以上に大きくすると、予測期間に影響を与えていない不要なデータが混入することにより、また、入力層のサイズを本来必要なもの以下にすると、予測期間に大きな影響を与えている重要なデータが無視され、結果としていずれの場合も予測精度は低下する。ニューラルネットワークは入力データと教師データとの写像関係を学習するため、入力データの過不足、即ち入力データに不要な情報が多数含まれているものと、必要な情報が欠如しているものとは、その意味としては等価である。それ故最適な入力層のサイズに比べ、それより大きくても、小さくても対象の特徴を正確に獲得することができなかつたものと思われる。

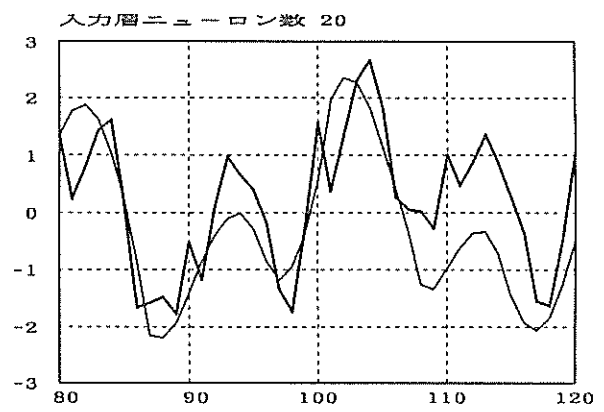
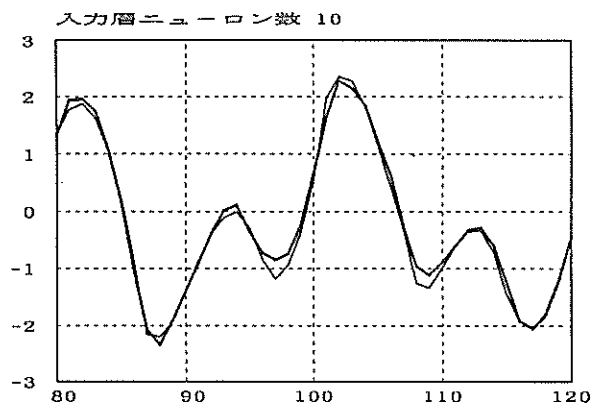
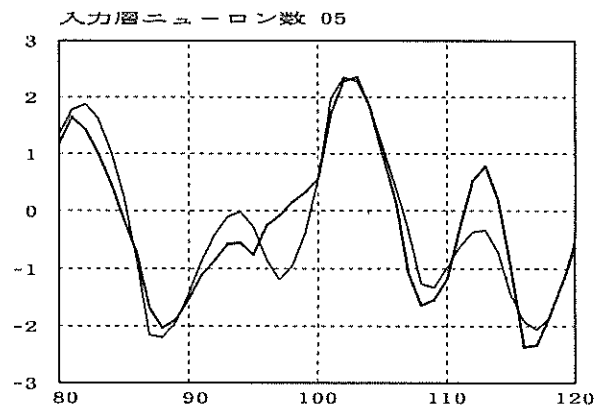
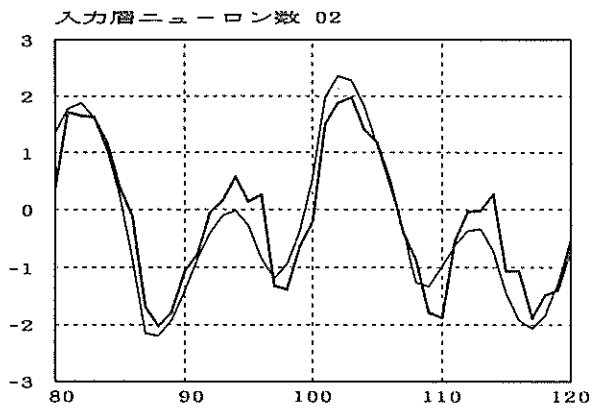


図6.15 予測期間（入力層ニューロン数）の違いによる予測結果

## 7. ニューラルネットワークの実データへの適用

本報告ではこれまでに、ニューラルネットワークの動作特性、あるいはまた、ニューラルネットワークを利用する際の注意点、テクニック等、種々の数値的な検討を行った。そこで本章ではこれらの検討結果を基に、実データへの適用として、ニューラルネットワークによる赤潮の発生予測を行う。

ニューラルネットワークの適用対象として、赤潮の発生予測を選定した理由は、主として以下に示す通りである。

①赤潮は生物学的、物理学的および化学的な要素が複雑かつ密接に関与し合った現象であるため、一般に知られている発生条件が満足されている場合でも、必ずしも発生するとは限らない極度の非線形現象であること。

②赤潮発生のダイナミクスには未だ不明確な点が多く、そのため従来の線形モデルでは、精度良く赤潮の発生を予測することは困難とされている。

また本章、さらには本報告の目的は、ニューラルネットワークを使用する際のノウハウを身につけ、同時にニューラルネットワークの適用可能性を検討することであり、赤潮の発生を予測することではない。あくまで赤潮の発生予測は、ニューラルネットワークの適用可能性を検討するための、一対象であることを予め明示しておく。

### 7.1 資料

本章で赤潮の発生予測を行うために用いた水質データは、運輸省第三港湾建設局神戸機械整備事務所が実施した水質自動観測装置開発に関する調査研究の際に測定されたものである。観測は昭和58年から60年まで計3回実施されているが、本報告では観測期間が最も長く、かつ安定にデータが取得された60年の測定データを使用する。なお、この調査で測定された水質項目、観測条件ならびに測定された水質項目の概要、特性等については村上<sup>16)</sup>を参照されたい。

### 7.2 実験準備

ここでは、通常植物プランクトン量を示す最も良い指標と考えられている、クロロフィル-a量を赤潮の指標値とし、このクロロフィル-a量の予測を間接的ではあるが赤潮の予測とする。また、クロロフィル-a量の予測に使用する説明変数は、村上<sup>16)</sup>の検討結果にならない、水温、塩分、リン、およびアンモニアとした。予測実験に用いる各測定データは図7.1に示すように、クロロフィル-a量の変動に富み、また各測定データが比較的連続して取得された、昭和60年7月13日0時から7月26日6時までの期間の各160データとした。なお、各測定データともサンプリング間隔は2時間である。実際にニューラルネットワークによりクロロフィル-a量の予測を行うに際し、計160個から成る各説明変数の内、前半の120個を学習データ、残りの40個を予測データとした(図7.2)。

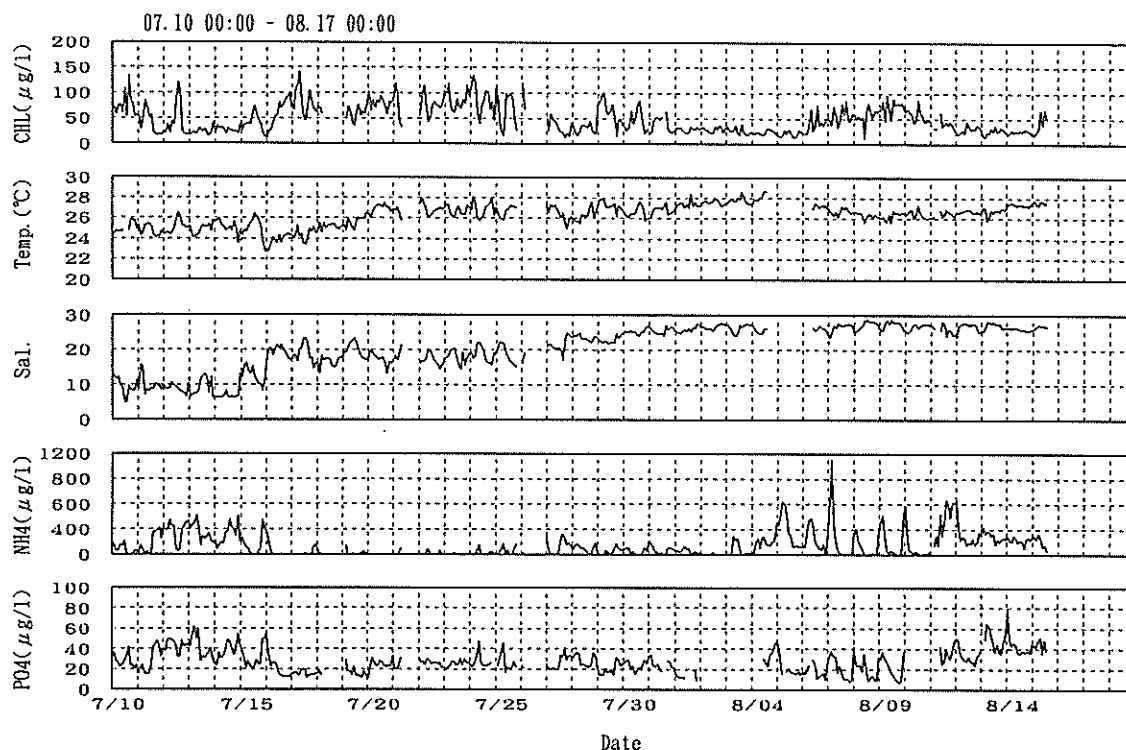


図7.1 各水質データの時系列



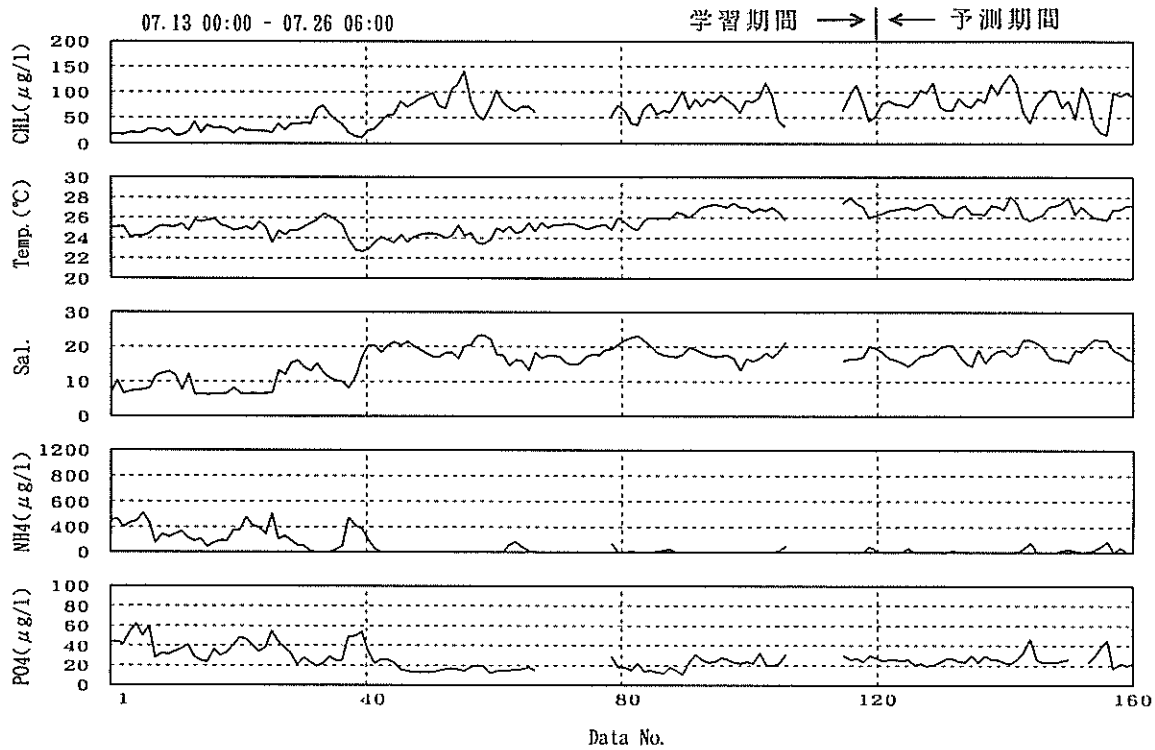


図7.2 学習および予測に使用した水質データの時系列

### 7.3 入出力層ニューロン数の決定

入出力層のニューロン数、即ちクロロフィル-a量の予測を行う際、どれだけ先までの予測を行うことが適当であるか、その決定手法として、前章で検討したフラクタル次元を利用する。図7.3にクロロフィル-a量のフラクタル次元を示す。これによると、 $t=5$ 付近を境にして2種類のフラクタル次元が存在していることが分かる。また、この時のフラクタル次元は、それぞれ $1 \leq t \leq 5$ では $D=1.6$ 、 $5 \leq t$ では $D=2.0$ であった。以上の結果より、サンプリング間隔を10時間（測定データの収録間隔が2時間であるため）以上としたデータは、ほぼランダムと見なされ、ある時刻より10時間以上先までのデータを予測することは本来的に無意味であることを示している。一方、サンプリング間隔が10時間以下の場合、そのフラクタル次元が概ね1.0と2.0の間にあることから、ある時刻より10時間以内までのデータには、ある程度の規則性が存在していることが予想される。同時に、ニューラルネットワークがその規則性を獲得する可能性が生じることになる。そこで、クロロフィル-a量の予測を行うニューラルネットワークの出力層ニューロン数を5に決定し、以後の検討を行う。

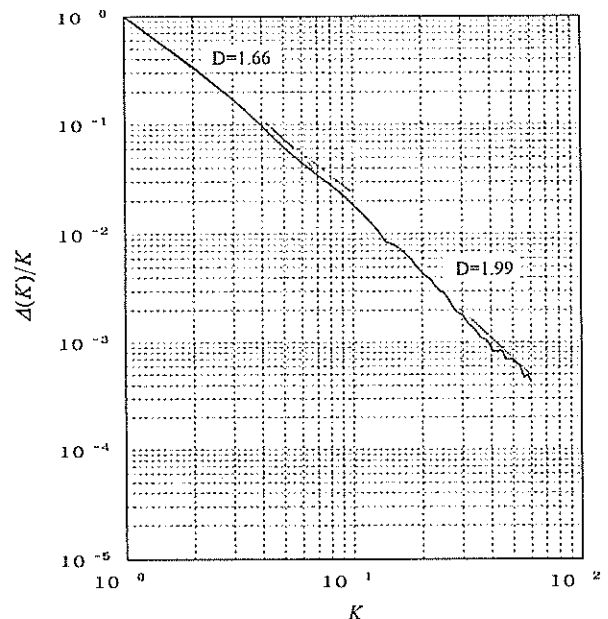


図7.3 クロロフィル-a量のフラクタル次元  
(K:サンプリング間隔,  $\Delta(K)$ :折れ線の長さ)

#### 7.4 学習方法

学習は前章の検討結果において、時系列データの処理にその有効性が確認されたフィードバック型ニューラルネットワークにより行った。また、この学習に採用したネットワークパラメータを表7.1に示す。なお入力データに対しては一切特別な前処理は行っていないが、欠測データに対してのみ、これを線形補間し使用した。学習時の入力層ニューロン数については、予測期間を5期(10時間)先とする前節の検討結果に基づき、説明変数5種類(水温、塩分、リン酸態リン、アンモニア態窒素、およびクロロフィル-a)さらにネットワークの出力と教師データとの差として入力層に回帰されるフィードバック量を合わせた計6種類それぞれに対し、過去5期分のデータを要するため計30個、これにバイアスニューロンを加え、最終的に必要なニューロン数の合計は31個となる。予測時の入力層ニューロン数は、フィードバック量が不要となるため26個となる。

表7.1 ネットワークパラメータ

入力層ニューロン数	31
中間層ニューロン数	8
出力層ニューロン数	5
動作関数の勾配( $\alpha$ )	1
学習係数( $\epsilon$ )	0.5

#### 7.5 予測結果

フィードバック型ニューラルネットワークによるクロロフィル-a量の予測結果を図7.4に示す。図中実線が実測値(目標関数)、一点鎖線がネットワークの予測結果を示している。これによると予測されたクロロフィル-a量は、全体的に実測値の変動傾向を概ね再現していることから、ニューラルネットワークはクロロフィル-a量の変動特性をその学習データより、ある程度獲得することができたものと思われる。しかしながら赤潮の発生予測を行うという目的では、クロロフィル-a量が増加傾向にあるのか、あるいは減少傾向にあるのかといった変動傾向に加え、クロロフィル-a量のピーク値に対する予測結果が問題である。そのような意味において、図7.4に示したクロロフィル-a量の予測結果は、特にピーク値に対する予測精度が不十分であるように思われ、さらなる精度向上のための検討が必要である。(この問題については7.7および7.8で議論する)

#### 7.6 入力層ニューロン数の妥当性

前節で示したクロロフィル-a量の予測結果は、その予測期間をフラクタル次元(6.11参照)より10時間先までとしたものであった。そこで本節では、より良い予測結果を得るための予測期間として、上述の10時間がはたし

て本当に適当であるのか否かについて検討を行なった。検討手法としては、予測期間をこれまでの10時間より短かくした場合、および長くした場合として、6時間および14時間にそれぞれ設定し、各ケースにより得られた予測結果を予測期間10時間とした前節の結果と比較する。表7.2は6時間、10時間および14時間の各予測期間に対する予測値と実測値との相関係数および平均2乗誤差を示したものである。

表7.2 相関係数および平均2乗誤差

予測期間	相関係数	平均2乗誤差
6時間	0.130	950.3
10時間	0.438	804.9
14時間	0.014	1276.7

これによると、予測期間を10時間としたときの相関係数が最も高く、さらに平均2乗誤差も最小となっていることから、クロロフィル-a量の予測期間としてはやはり10時間が適当であることが分かった。同時にこの結果は、予測期間の決定方法として、前章で検討したフラクタル次元が実データに対しても有効的に利用可能であることを示すものであると思われる。参考のため予測期間を6時間、および14時間に設定したときの両予測結果をそれぞれ図7.5および図7.6に示す。これら図7.5および図7.6を図7.4と比較すると、表7.2の結果を見るまでもなく、図7.4の予測値が最も実測値に近いことは明らかである。この結果の示すことは、ある時刻でのクロロフィル-a量の値は、およそ10時間程度の過去の影響を受けているということである。そのため、本来影響を受けている期間のおよそ半分(6時間)のデータからその将来を予測すると、予測値の決定に影響を与えるべきはわずのおよそ半分のデータが無視されることになり、またこれとは逆に、本来影響を受けている期間よりも不必要に長い期間(14時間)のデータからその将来を予測すると、予測値の決定にほとんど影響を与えていない多数のデータが悪影響を及ぼし、結果としてどちらの場合も予測誤差が増大したものである。

#### 7.7 説明変量の追加による予測精度の向上

前節ではニューラルネットワークによるクロロフィル-a量の予測を行なった。その結果、予測値は実測値の変動傾向を概ね再現することができたが、ピーク値に関する予測精度は十分なものではなかった。

また、赤潮の発生に関しては、河川水からの栄養塩類の補給、塩分量の低下、日照および赤潮生物が表層に集積するのを妨げない平穏な海況が発生条件として一般に考えられている(例えば文献16)。これまでの検討ではこれら栄養塩類、水温および塩分量をクロロフィル-a量

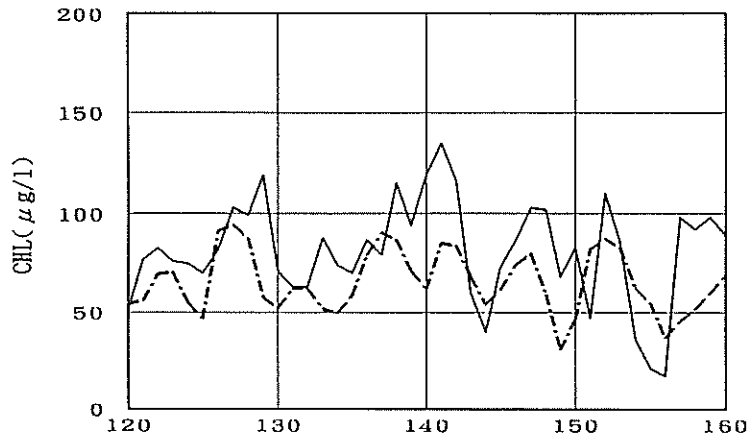


図7.4 クロロフィル-a量の予測結果 (予測期間10時間)

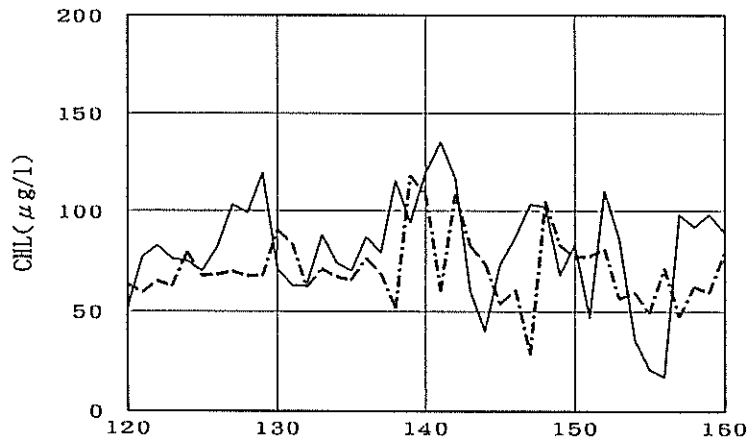


図7.5 クロロフィル-a量の予測結果 (予測期間6時間)

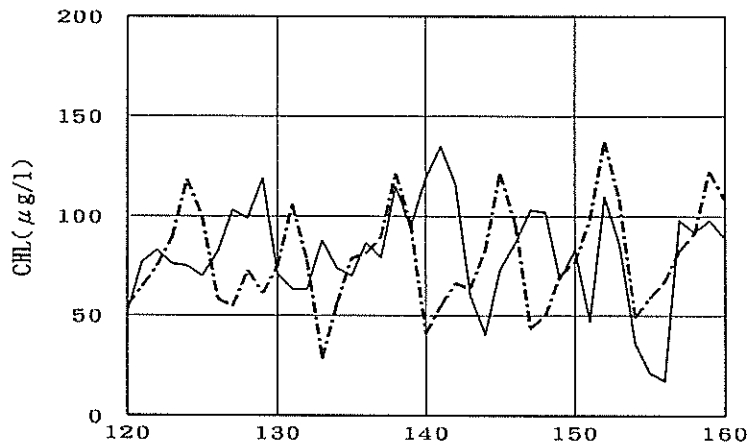


図7.6 クロロフィル-a量の予測結果 (予測期間14時間)

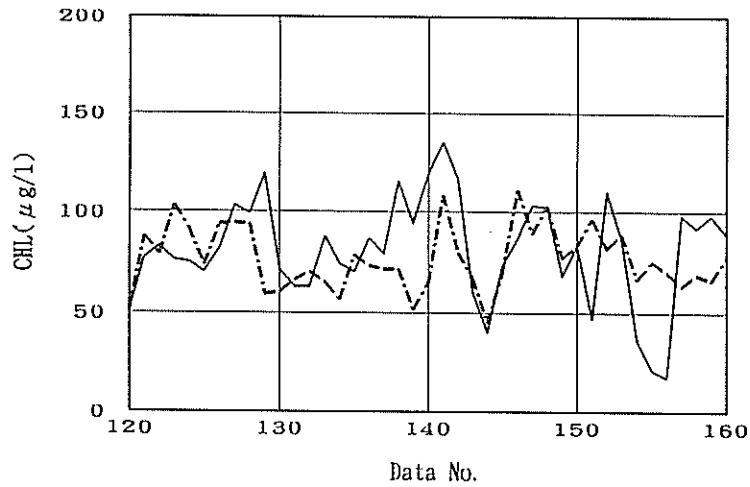


図7.7 入力に有義波高を追加したクロロフィル-a量の予測結果

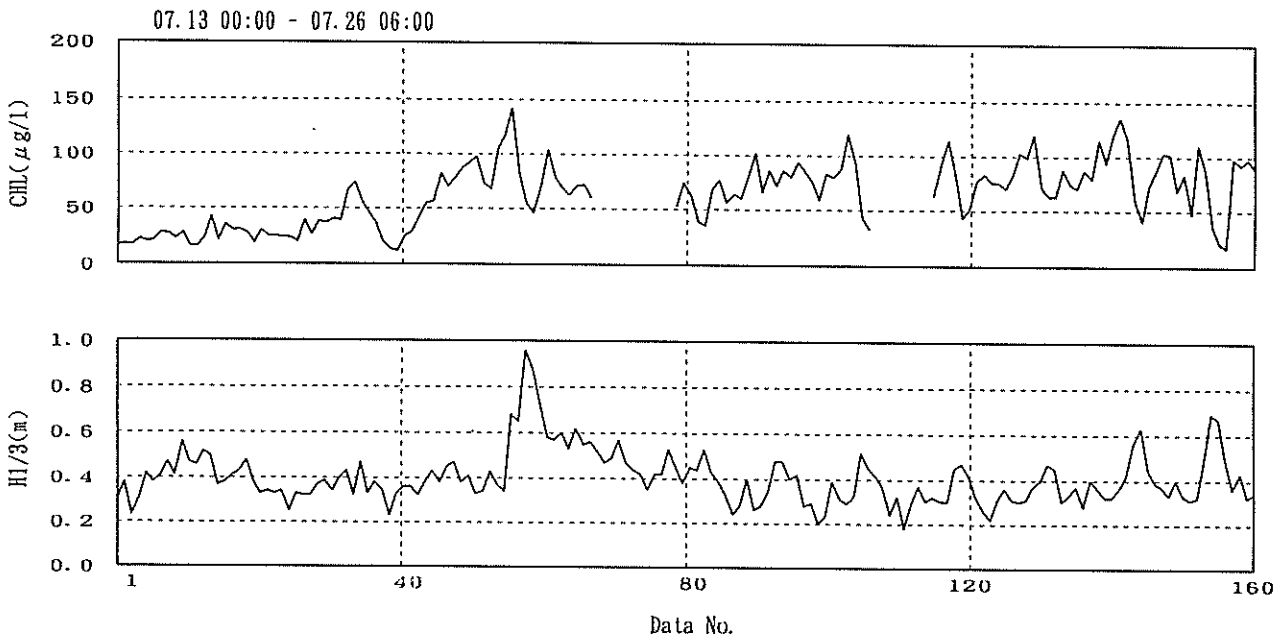


図7.8 クロロフィル-a量と有義波高の時系列

予測の説明変量として使用しているが、上記の「平穏な海況」に関連する物理量は使用していない。そこで本節では、「平穏な海況」を代表する観測物理量として、有義波高を新たに説明変量に追加し、再度クロロフィル-a量の予測を行った。得られた予測結果を図7.7に示す。

この結果と説明変量に有義波高を使用していない図7.4を比較すると、図7.7の方が実測値と予測値のピーク値に関する一致度が高いことが分かる。また、平均2乗誤差もこれまでの中で最も小さい717.3となったことから、ある時刻におけるクロロフィル-a量の決定に、それまでの有義波高が関与しているものと思われる。また、興味

深いのは、予測開始後72時間（データNo.144）の極小値に関する実測値と予測値の良好な一致である。説明変量に有義波高を加えない場合には、ネットワークパラメータあるいはネットワークサイズをどのように調整しようとも、この様な良好な一致を得ることはできなかった。即ちこの結果は、予測開始後72時間に存在するクロロフィル-a量の極小値を表現するためには、説明変量として水温、塩分、栄養塩類に加え波高の情報が必要であることを示すものと思われる。

事実、この時の有義波高は図7.8に示すように極大値を示しており、波高の増大が赤潮発生の制限因子となって

いるものと思われる。

以上本節の結果をまとめると、ニューラルネットワークを利用してある対象の変動を精度良く予測するためには、その変動を十分説明し得る説明変量を準備することが必要である。

### 7.8 入力データに事前処理を施すことによる予測精度の向上

前節では、ニューラルネットワークによるクロロフィル-a量の予測精度を向上させるため、ネットワークに入力する説明変量の種類を増加させ、その効果について検討を行った。本節では、予測精度を向上させるもう一つの手法として、入力データ（各説明変量および目的変量）に簡単な前処理を施し、これにより新たに得られたネットワークの予測結果について検討を行う。

通常観測データには、計測誤差その他の要因により、何らかの誤差が含まれている場合が多い。ニューラルネットワークによりこれら観測データ間の因果関係を学習させる際、変動の本質をネットワークに獲得させ、同時に過学習を抑制させるためにも、本来的には学習データに含まれるこれら誤差を統計手法等により、事前に取り除いておくことが望ましい。

そこで本節では、各説明変数および目的変数内に含まれるランダムウォークを抑制し、これら各変数が有する本質的な変動を強調することを目的に、各変数に対し $t$ および $t \pm 1$ 時刻の計3データ（6時間相当）による移動平均処理を全入力データに施した。この移動平均処理を施した各説明変数および目的変数を新たな学習データとして、再度クロロフィル-a量の予測を行った。図7.9にこの結果を示す。この図7.9と入力データに前処理を施していない図7.4を比較すると、図7.9の方が明らかに予測精度が高いことが分かる。また、この時の実測値と予測値との相関係数はこれまでで最高の0.60を、平均2乗誤差はこれまでで最小の470.8をそれぞれ示した。

以上の結果、ニューラルネットワークで複数の時系列（説明変量）から対象時系列（目的変量）の予測を行う際、例えば本節で検討した移動平均等の事前処理を各入力データに加え、これら各データの本質的な変動特性を明瞭化することにより、予測精度の向上が図れる場合があることが分かった。別の言葉で言うならば、より高精度な予測結果を得るために重要なことは、質の高い情報を必要最小限にネットワークへ提示することであるものと思われる。

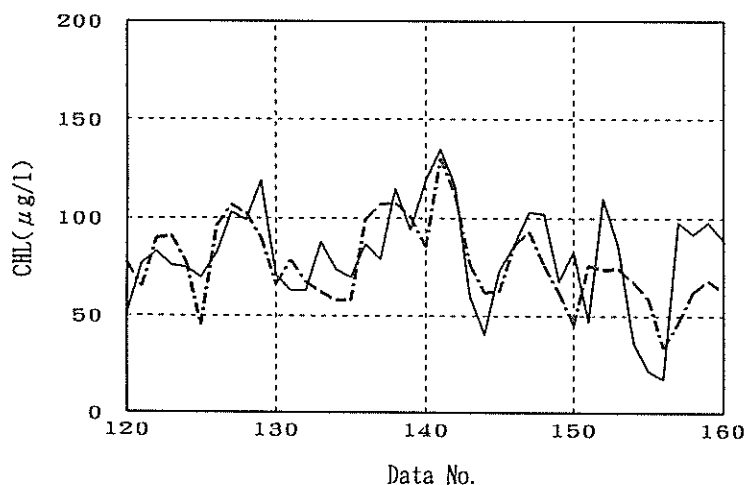


図7.9 入力に移動平均を施した時のクロロフィル-a量の予測結果

### 7.9 説明変量の寄与度

本章でこれまでに行ってきた種々の検討は、いずれもクロロフィル-a量の予測精度の向上に関するものであり、クロロフィル-a量決定に対する各説明変量の影響の程度等には言及していない。そこでここでは、本章の目的からは多少逸脱するが、クロロフィル-a量の予測にどの説明変量が最も影響を与えているのか、この問題について

検討を行う。

手法としては、5種類の入力データ（水温、塩分、アンモニア、リン酸およびクロロフィル-a量）から1種類ずつ削除した4種類の入力データ（例えば、塩分、アンモニア、リン酸、クロロフィル-a量を入力データセットとする）を計5ケース準備し、それぞれのケースにおけるネットワークの予測値を求めた。得られた結果として、

表7.3 各項目を削除したときの相関係数および平均2乗誤差

削除した項目	Temp.	Sal.	NH4	P04	CHL	無し
相 関 係 数	0.439	0.553	0.441	0.454	0.243	0.600
平均2乗誤差	687.8	615.9	655.8	634.7	1329.7	470.8

CHL : クロロフィル-a

7.3に実測値と予測値との相関係数および平均2乗誤差を示す。

表7.3に示したように、計5ケースの中で予測精度が最も低下したものは、入力データからクロロフィル-a量を削除した場合であった。また、クロロフィル-a量以外の項目については、特に目立った大きな差は見られないが、この中で最も予測精度が低下したものは水温を削除した場合で、次いでアンモニアとなった。一方、予測精度の低下が最も少ないものは、塩分を削除した場合であった。

以上の結果より、クロロフィル-a量の予測を行う際に

最も重要な説明変量は、それ自身、即ち過去のクロロフィル-a量、次いで水温、アンモニア、リン酸、塩分の順であることが分かった。クロロフィル-a量を除く各項目の中で、水温が最もクロロフィル-a量の予測に影響を与える項目となった理由の一つとして、赤潮生物の増殖に強く影響を与えていると思われる日射の情報が、水温として陰に反映されたものと考えられる。そこで今後、当該海域の日射情報を入手し、これを入力データに加えることにより、より精度の良い予測結果が得られるものと期待される。

## 8. おわりに

本研究は、従来の数理モデルでは表現することが困難な港湾環境に関する複雑かつ複合的な諸現象の評価、予測、制御といった一連の問題に対し、将来これらを可能とする手段として大きな期待が持たれているニューラルネットワークについて、これを今後有効的に利用することを目的とした基礎的な研究である。そのため、本報告書ではニューラルネットワークに関連した教科書あるいは論文等で通常報じられることが少ない、ニューラルネットワークの基本的な動作特性および使用する際の注意点を議論の中心としており、本来要求されるニューラルネットワークの応用成果については、その適用可能性を提示するに留めていることを御理解頂きたい。

本研究の初期においては、ニューラルネットワークのプログラムを実行しさえすれば、「そこそこ」の出力が期待できるものと認識していた。これに対する答としては、現在でも「Yes」である。しかし、この「そこそこ」から、ある程度納得のできるレベルまで予測精度を高めるためには、ニューラルネットワークおよび対象のダイナミクスについての知識が望まれる。4.3でも示したように、ニューラルネットワークの学習は、基本的には「やってみたら旨くいった」の世界である。この意味するところは、精度の高い出力を可能とする、最適ネットワークの構築手段が未だ確定されていないことに起因して、必然的に試行錯誤的な作業が伴うということである。その理由としては、例えば全く同様なネットワークを利用しても、結合荷重の初期値の与え方一つで、ネットワークの出力が大きく異なるケースなどは決して珍しいことではないばかりか、ニューラルネットワークを実行する際、事前に決定しなければならないネットワーク構造、あるいは学習パラメータ等、それらの自由度が極めて大きいことが挙げられる。それ故、「旨く」いくためにはそれなりの経験と知識がやはり必要となるようである。ましてや本報告で取り扱ったネットワークは、そ

の構造が最も基本的なものであり、文字認識、音声認識等で既に実用化され、評価を得ているネットワークの多くは、非常に複雑かつ巨大なものである。

また、ニューラルネットワークの出力について、特に出力値の精度に関しては疑問視する意見も少なくないようである。実際対象の数理モデルが確立されているものに対しては、ニューラルネットワークの予測精度がこれに勝ることは恐らく希であろう。それでは実際のところ、ニューラルネットワークを利用することのメリットは無いのであろうか。決してそうではない。現実に数理モデルでは表現し得ない問題が、特に環境問題には多数存在していることは事実であり、この種の問題に対しては、現在のところニューラルネットワーク（広い意味での人工知能的）のような処理手法に頼らざるを得ないのも、恐らくまた事実であろう。そのような意味において、今後ニューラルネットワークの研究を進めていくことは大いに意義があるものと考えられる。

最後に、ニューラルネットワークはそのプログラムを所有してさえいれば、誰でも直ちに利用することが可能であるが、これをより有効的に活用するためには、上述のように相当な経験が必要であるものと思われる。しかし、ある対象の入出力関係が明かでなく、これを解明することが非常に困難とされている問題に対しては、それを表現するモデルが確立されるまでの開発時間と、同問題を処理可能とするネットワークの構築に要する時間とを比較すると、恐らく多くの場合、ニューラルネットワークの方がより短時間で、ある程度の成果が期待可能であるものと思われる。ならばそのモデルが開発されるまでの暫定的な処理手法として、あるいはニューラルネットワークにより得られた知見をモデルの開発にフィードバックするといった、いわば先導的な役割としてのニューラルネットワークの利用価値は大いにあるのではないだろうか。微力ながらニューラルネットワークの研究を行ってきた者としては、今後海岸工学の分野に広くニューラルネットワークが利用されていくことを期待したい。

(1997年9月30日受付)

## 参考文献

- 1) システム制御情報学会 編：ニューラルネットと計測制御，朝倉出版，p.228.，1995.
- 2) 市川 紘：階層型ニューラルネットワークー非線形問題解析への応用ー，共立出版，184p.，1993.
- 3) 秋葉巴也 訳：ニューラルネットワークの物理モデル，吉岡書店，p173.，1992.
- 4) 松葉育雄：ニューラルシステムによる情報処理，昭晃堂，p191.，1993.
- 5) 上坂吉則：ニューロコンピューティングの数学的基礎，近代科学社，p241.，1993.
- 6) 和田安弘・川人光男：新しい情報量規準と Cross Validationによる汎化能力の推定，電子情報通信学会論文誌 D-II，Vol.J74-D-II，No.7，pp.955～965，1991.
- 7) 松葉育雄：時系列データ向きニューラルネットワークとその経済問題への応用，平成3年電気・情報関連学会連合大会，1991.
- 8) 松葉育雄：バックプロパゲーションによる特徴抽出，数理科学8月号，サイエンス社，pp.31～37，1991.
- 9) 渡辺 歩・森村信雄・永野俊：リカレントニューラルネットワークによる時系列予測に関する研究，電子情報通信学会技術報告，NC95-118，pp.17～24，1996.
- 10) 木ノ内誠・萩原将文：複素ニューロンによる時系列の学習，電気学会論文誌 C，第116巻，第3号，pp.748～754，1996.
- 11) Waibel, A. et al. : Phoneme Recognition Using Time-Delay Neural Networks, IEEE Trans. ASSP, 37, pp.328 ~ 339, 1989.
- 12) Jordan, M. I. : Serial Order: a parallel distributed processing approach, Technical Report 8604, ICS UC San Diego, 1986.
- 13) Elman, J. L. : Finding Structure in Time, Cognitive Science, 14, 2, pp.179 ~ 211, 1990.
- 14) 船橋誠壽：ニューロコンピューティング入門，オーム社，p152.，p.1992.
- 15) 武者利光：ゆらぎの世界，講談社，p234.，1996.
- 16) 村上和男：港湾技術研究所報告，第26巻，第2号，pp.213～252，1987.
- 17) 和達清夫 監修：海洋大辞典，東京堂出版，pp.1～2，1987.



## 補章A 焼き鈍し法について

最急降下法では一度でも極小解につかまると、そこから脱出する術を持たない。初期値を多数用意して、各初期値ごとに学習を行うことは有効な手段ではあるが大変手間がかかる。これに対して焼きなまし法と呼ばれる確率的なゆらぎを導入することにより、極小解からの脱出を図る手法<sup>1)</sup>がある(式A.1)

$$\Delta W_{ij}(t) = -\varepsilon \frac{\partial E}{\partial W_{ij}(t)} + \xi_{ij}(t) \quad (A.1)$$

通常のバックプロパゲーション法との違いは、右辺にランダム項  $\xi_{ij}(t)$  (平均値0で分散  $T(t)$  の独立な白色ガウスノイズ) が付加されているだけであるが、この項こそが極小解を回避する働きを持っている。バックプロパゲーション法の学習アルゴリズムに従い、誤差曲面を忠実に下っていくと、誤差面が平らになったらそこで止まらざるを得ない。しかしながら、右辺ノイズ項の影響で、その付近を探索しているうちに、その変化量が大きければ極小解の谷間からこぼれ出てくる可能性が高くなる。ただし、ノイズが大きすぎると、重みがノイズに支配されてしまい、新たな谷間に気付くことなく、いつまでたっても山を下れない。また、ノイズが小さすぎるとは極小解から脱出できない。そのためノイズの大きさ(分散)は重要であり、その与え方には注意を要する。

そこで焼きなまし法では、誤差が比較的大きな学習初期に分散を大きくし、学習が進むにつれ分散を小さくする。この意味するところは、学習の進行具合に伴って誤差曲面の勾配も小さくなり、これに応じてノイズも小さくする必要があるということである。誤差曲面の勾配が大きくなうちはある程度ノイズが大きくても誤差曲面を下っていくが、誤差曲面の勾配が小さくなると相対的にノイズの方が大きくなり、そうなるとうちはノイズに振り回されて、誤差曲面を下る学習本来の意味が無くなってしまふ。そのため学習の進行に合わせ、ノイズの分散を徐々に小さくしなければならない。

同じような手続きを我々人間が問題を解決する場合に

も見るができる。例えば優秀なスポーツ選手、あるいは研究者がある日スランプに陥ることがある。これは、極小解に捕捉されてその先にあるであろう下り坂が見えない状態に対応する。このようなとき、我々人間は、ただがむしゃらに下り坂を探せばかりでなく、ときとして周囲のアドバイスに耳をかたむけたり、気分転換を図ったりすることにより、このスランプ、即ち極小解から脱出することができる場合がある。しかもスランプ状態から脱出することができた際には、以前よりも高度な技術を身につけられる場合が多い(より深い谷に到達している)。つまり、問題解決のために山を下る場合、まじめに急な下り坂を探すだけでなく、ランダムに周囲を歩き回る両方の要素が必要である。焼きなまし法とはこのように、学習初期の段階では、はじめに見つけた1つの谷に固執せず、一通りどのような谷があるのか探して回り、その後徐々に気を引き締めて山を下るとしばしば良い結果が得られることを教えている。

また一般に出力誤差最小で定義される学習手法では、必ずしも最適解に到達するとは限らない。このため、極力良い解にたどり着くために、様々な初期値を用意したり、確率的なゆらぎを導入するなどの策をとっている。それでは、我々人間の脳が行っている問題解決の手法はどのようなものであろうか。我々人間が現実社会で問題を解決する際には、例えば締め切り日までに論文を作成する、津波が到達するまでに最低限必要な貴重品を持って高台に避難する等、その多くの場合に時間的な制約が存在する。このような状況下では最適解を探す時間はなく、与えられた時間内に少しでも良い解を探す必要がある。ここでの時間的制約とは、学習での最大繰り返し数に対応する。もちろん、ある程度良好な解を探し出すための時間的制約、即ち学習の最大繰り返し数に対するおおよその目星は必要であるが、得られた解が最適解かどうかにかかわらず、学習本来の目的を見失わないよう、注意を払わなければならない。学習することの本来の目的とは、対象に内在する特徴を抽出・把握することであり、出力誤差を最小にすることではない。あくまで出力誤差最小化は学習の一手段であることを心掛けたい。

港湾技研資料 No.883

1997・12

編集兼発行人 運輸省港湾技術研究所

発行所 運輸省港湾技術研究所  
横須賀市長瀬3丁目1番1号

印刷所 株式会社シーケン

Published by the Port and Harbour Research Institute  
Nagase, Yokosuka, Japan

Copyright © (1997) by P.H.R.I.

All rights reserved. No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the Director General of P.H.R.I.

この資料は、港湾技術研究所長の承認を得て刊行したものである。したがって、本資料の全部又は一部の転載、複写は、港湾技術研究所長の文書による承認を得ずしてこれを行ってはならない。