

3層ニューラルネットワークの 関数学習における誤差領域

田口 功* / 須貝 康雄**

The Error Territory in the Functional Learning of the Three-Layer Neural Networks

Isao TAGUCHI and Yasuo SUGAI

Function learning is well known using the generalization ability in neural networks. Errors increase very much using the general sigmoid function in three-layer neural networks. The error is not improved even if learning times are increased. There is an error even if the operation of standardization is done.

It is shown experimentally that there is a territory of error in the neighborhood of the point of inflection and the territory of large absolute value of inclination of a goal function.

*たぐち・いさお：敬愛大学国際学部助教授 情報処理論

Associate Professor of Computer Science, Faculty of International Studies, Keiai University; information processing.

**すがい・やすお：千葉大学大学院自然科学研究科助教授 ニューラルネットワーク

Associate Professor of Network Science, Graduate School of Technology, Chiba University; neural network.

1. はじめに

ニューラルネットワークを用いて関数を学習することは、よく知られている。ニューラルネットワークの性能を表現するとき、必ず実験のひとつの項目となる(参考文献 [1]) ガウス関数がしばしばそのシュミレーションの対象になる。ガウス関数を実際にニューラルネットワークを用いて学習すると、比較的学習回数を増加させなくとも、精度はよく誤差はそれほど気にならないという報告がある(参考文献 [1])。どのような関数でも学習回数を増加させることによって誤差が減少すれば問題はない。しかし、学習する関数によっては、誤差が大きくなり、学習回数を増加しても誤差が減少しない場合がある。また、誤差に関する研究はあまり行われていない。本稿では、誤差が比較的大きな関数(参考文献 [2]) や \sin 関数をモデルとして実験を行い、誤差の多く発生する領域を実験的に考察した。

2. 3層ニューラルネットワークの構成

最初に、図2.1に示すような3層ニューラルネットワークを考える。ここで、 $\mathbf{x}^{(p)}$ は p 番目の入力パターンベクトル、 $t^{(p)}$ は、それに対応する目標値である。 $z^{(p)}$ は、実際の出力とする。 $V_{ji}^{(1)}$ は、入力層と中間層の間の重み係数、 $W_{kj}^{(h)}$ は、中間層と出力層の間の重み係数である。ここで、 i は入力層の1パターンあたりの入力数、 j は中間層ニューロン数、 k は出力層の1パターンあたりの出力数である。出力層にある素子は、入力 u に対して、シグモイド関数を使用すると、

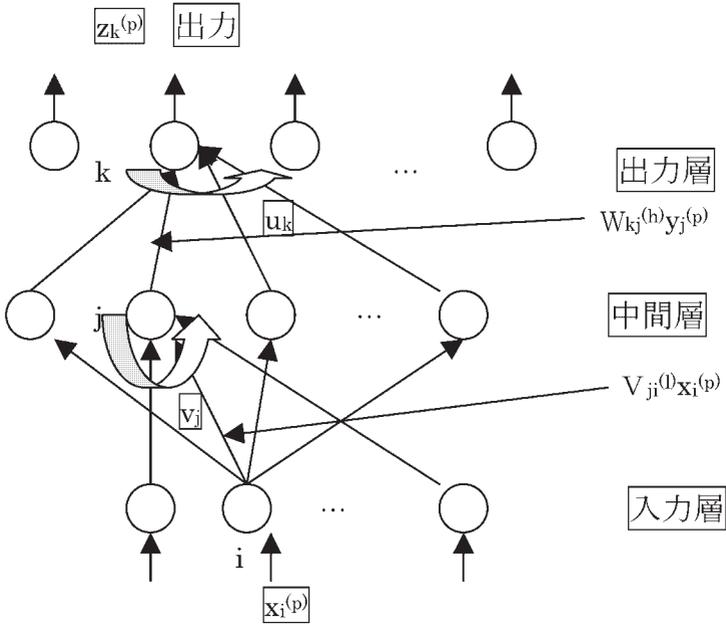
$$g(u) = 1 / (1 + e^{-ku})$$

となる。学習は、

$$E = \sum E^{(p)} = \sum \sum \{(t_k^{(p)} - z_k^{(p)})^2 / 2\}$$

の総誤差 E を最小化することである。ここで $t_k^{(p)}$ は目標値であり $t_k^{(p)} - z_k^{(p)}$ は、 $\mathbf{x}^{(p)}$ に対する3層ニューラルネットワークの出力と目標値の間の誤差

図2.1 3層ニューラルネットワークの構成



となる。する。パターン $x^{(p)}$ が入力として加わった場合、出力誤差 $E^{(p)}$ は、

$$E^{(p)} = \sum \{(t_k^{(p)} - z_k^{(p)})^2 / 2\}$$

となる。この $E^{(p)}$ を減少させるように $W_{kj}^{(h)}$ と $V_{ji}^{(1)}$ を調節すると、

$$\Delta W_{kj}^{(h)} \text{ と、 } \Delta V_{ji}^{(1)} \text{ は、}$$

$$\Delta W_{kj}^{(h)} = -\varepsilon \partial E^{(p)} / \partial W_{kj}^{(h)}$$

と

$$\Delta V_{ji}^{(1)} = -\varepsilon \partial E^{(p)} / \partial V_{ji}^{(1)}$$

となる(本稿では最急降下法を使用)。ここで、 ε は学習係数である。

出力層にある素子 k の入力 u_k は、

$$u_k = \sum W_{kj}^{(h)} y_j^{(p)}$$

となり、出力は、

$$z_k^{(p)} = g(u_k)$$

となる。また、偏微分の連鎖律を考慮すると、

$$\begin{aligned}
-\partial E^{(p)}/\partial W_{kj}^{(h)} &= -\partial E^{(p)}/\partial u_k \cdot \partial u_k/\partial W_{kj}^{(h)} \\
&= -\partial E^{(p)}/\partial z_k^{(p)} \cdot \partial z_k^{(p)}/\partial u_k \cdot \partial u_k/\partial W_{kj}^{(h)} \\
&= (t_k^{(p)} - z_k^{(p)}) \cdot 2m \cdot z_k^{(p)}(1 - z_k^{(p)}) \cdot y_j^{(p)} \quad (2-1)
\end{aligned}$$

が成り立つ。ここで、

$$z_k^{(p)} = g[\sum \{W_{kj}^{(h)} f(V_{ji}^{(l)} x_i^{(p)} + \alpha_j)\}] \quad (2-2)$$

となる。また、出力 $z_k^{(p)}$ の入力に関する微分式は、

$$z_k^{(p)'} = g'(u_k) \cdot [\sum \{W_{kj}^{(h)} f'(V_{ji}^{(l)} x_i^{(p)} + \alpha_j) \cdot V_{ji}^{(l)}\}] \quad (2-3)$$

である。たとえば、入力数 $i=1$ 、中間ニューロン数 $j=3$ 、出力ニューロン数 $k=1$ とすると、

$$\begin{aligned}
z_1^{(p)'} &= g'(u_1) \cdot \{W_{11}^{(h)} f'(V_{11}^{(1)} x_1^{(p)} + \alpha_1) \cdot V_{11}^{(1)} + W_{12}^{(h)} f'(V_{21}^{(1)} x_1^{(p)} + \alpha_2) \cdot \\
&\quad V_{21}^{(1)} + W_{13}^{(h)} f'(V_{31}^{(1)} x_1^{(p)} + \alpha_3) \cdot V_{31}^{(1)}\}
\end{aligned}$$

となる。

ここで、 $\delta_k^{(h)} = (t_k^{(p)} - z_k^{(p)}) \cdot 2m \cdot z_k^{(p)}(1 - z_k^{(p)})$ とおくと、

$$-\partial E^{(p)}/\partial W_{kj}^{(h)} = \delta_k^{(h)} \cdot y_j^{(p)}$$

となる。従って、

$$\begin{aligned}
\Delta W_{kj}^{(h)} &= \varepsilon \delta_k^{(h)} y_j^{(p)} \\
&= \varepsilon \cdot (t_k^{(p)} - z_k^{(p)}) 2m \cdot z_k^{(p)}(1 - z_k^{(p)}) \cdot y_j^{(p)} \quad (2-4)
\end{aligned}$$

となり、出力層と中間層間の重み係数の更新式となる。次に、入力層と中間層との間の重み係数更新を述べる。

中間層にある素子 j の入力 v_j は、

$$v_j = \sum V_{ji}^{(l)} x_i^{(p)}$$

となり、シグモイド関数を使用した場合の出力は、

$$y_j^{(p)} = f(v_j)$$

となる。同様に偏微分の連鎖律を考慮すると、

$$\begin{aligned}
-\partial E^{(p)}/\partial V_{ji}^{(l)} &= -\partial E^{(p)}/\partial v_j \cdot \partial v_j/\partial V_{ji}^{(l)} \\
&= -\partial E^{(p)}/\partial v_j \cdot x_i^{(p)}
\end{aligned}$$

となる。ここで、 $\delta_j^{(l)} = -\partial E^{(p)}/\partial v_j$ とおくと

$$\begin{aligned}
\delta_j^{(l)} &= -\partial E^{(p)}/\partial y_j^{(p)} \cdot \partial y_j^{(p)}/\partial v_j \\
&= -\partial E^{(p)}/\partial y_j^{(p)} \cdot 2m \cdot y_j^{(p)}(1 - y_j^{(p)})
\end{aligned}$$

であり、さらに $\partial E^{(p)} / \partial y_j^{(p)}$ は、

$$\begin{aligned}\partial E^{(p)} / \partial y_j^{(p)} &= \sum \partial E^{(p)} / \partial u_k \cdot \partial u_k / \partial y_j^{(p)} \\ &= \sum \delta_k^{(h)} W_{kj}^{(h)}\end{aligned}$$

となるから、

$$\delta j^{(l)} = (\sum \delta_k^{(h)} W_{kj}^{(h)}) 2m \cdot y_j^{(p)} (1 - y_j^{(p)})$$

とおき、まとめると、

$$\begin{aligned}\Delta V_{ji}^{(l)} &= \varepsilon \delta j^{(l)} x_i^{(p)} \\ &= \varepsilon (\sum \delta_k^{(h)} W_{kj}^{(h)}) 2m \cdot y_j^{(p)} (1 - y_j^{(p)}) \cdot x_i^{(p)}\end{aligned}\tag{2-5}$$

となる。

3. 計算機実験および考察

1. 誤差の多い学習関数と学習関数微分および学習 2 回微分関数

本稿でニューラルネットワークを利用して関数学習をするわけであるが、一般的に検査関数としている関数を使用することとした。目標とする関数は、

$$f(X) = \exp(-X^2) \cos(2\pi X)$$

を使用した。この関数は、参考文献 (2) で示した関数の 1 次元版である。関数 $f(X)$ を図3.1.1に示す。

また、その微分関数

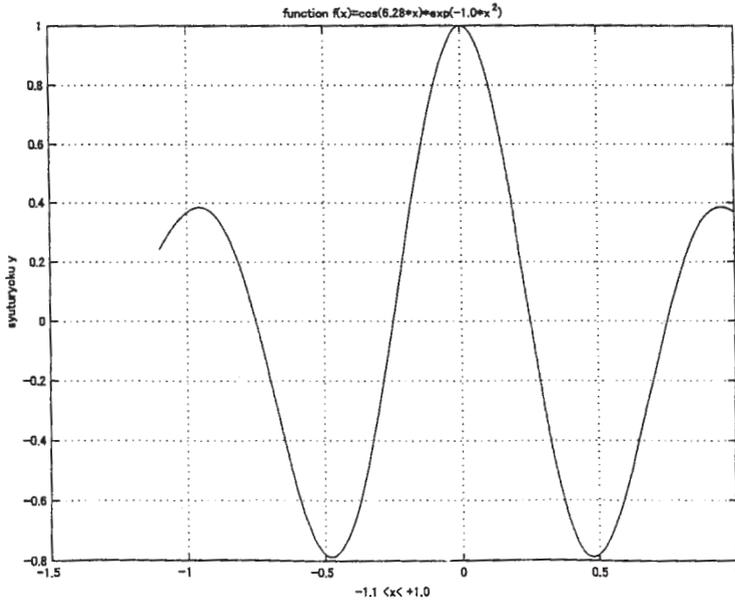
$$f'(X) = 2\pi \sin(2\pi X) \exp(-X^2) - 2X \cos(2\pi X) \exp(-X^2)$$

の関係を示すグラフを図3.1.2に示した。微分関数は、学習関数の変曲点を示すのに都合がよい。また、傾きの最大値も表示できる。さらに、 $f(X)$ の 2 回微分関数

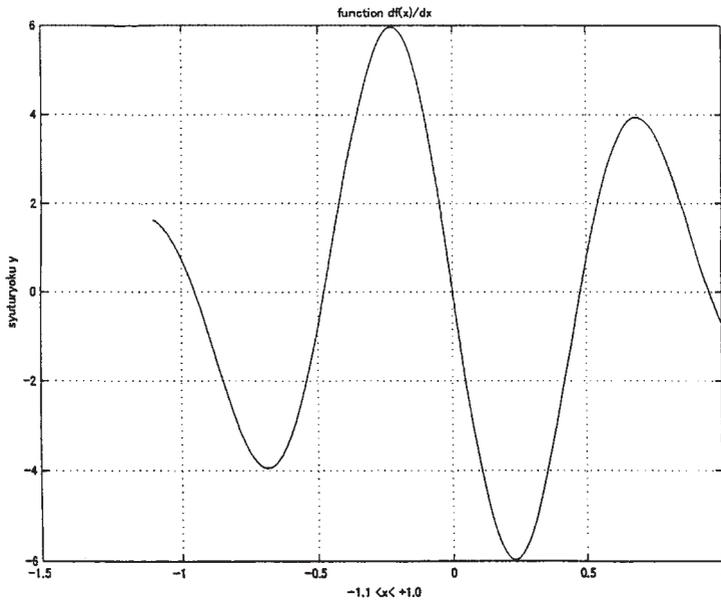
$$\begin{aligned}f''(X) &= \exp(-X^2) \{8.0\pi X \sin(2\pi X) - 2\cos(2\pi X) + 4X^2 \cos(2\pi X) \\ &\quad - 4\pi^2 \cos(2\pi X)\}\end{aligned}$$

を図3.1.3に示す。2 回微分関数は、1 回微分関数の最大値を確認するのに便利である。

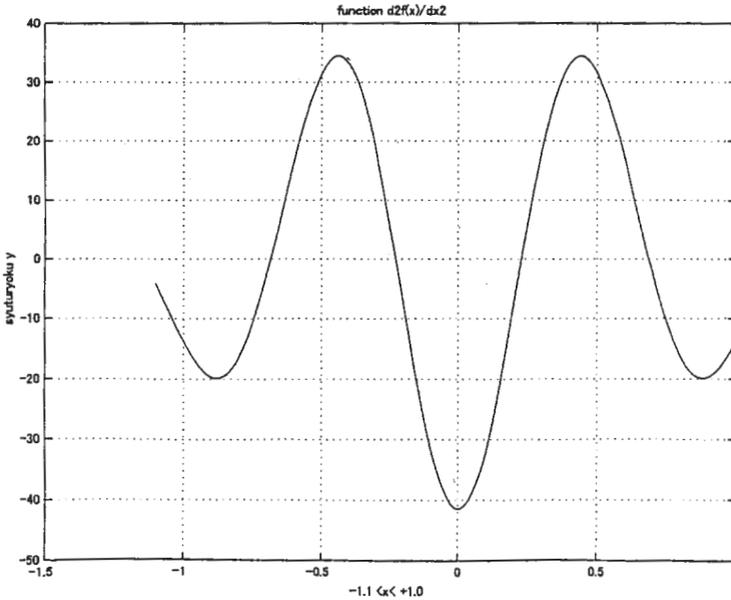
$$\text{Fig. 3.1.1 } f(x) = \exp(-x^2) \cos(2\pi x)$$



$$\text{Fig. 3.1.2 } f(x) = 2\pi \sin(2\pi x) \exp(-x^2) - 2x \cos(2\pi x) \exp(-x^2)$$



$$\text{図3.1.3 } f^*(X) = \exp(-X^2) \{ 8.0 \pi X \sin(2 \pi X) - 2 \cos(2 \pi X) + 4X^2 \cos(2 \pi X) - 4 \pi^2 \cos(2 \pi X) \}$$



2. 出力を規格化しない場合の関数学習 ($f(X) = \exp(-X^2) \cos(2 \pi X)$)

本実験では、1入力1出力、中間層のニューロン数は10個とした。図3.1.1の目標関数を学習する時、一般的なシグモイド関数をそのまま用いた出力は0から1と制限されてしまう。本実験では、一般的シグモイド関数を $f(u) = 1 / (1 + \exp(-u))$ とした場合、出力領域を -1.0 から 1.0 に拡大する目的で $2f(u) - 1.0$ とした。学習率を0.30とし、重み係数は乱数を用いて3種類選択した。学習データとしてのX軸上の値は、 -1.1 、 -0.8 、 -0.5 、 -0.2 、 0.1 、 0.4 、 0.7 、 1.0 の8点を取った。重み係数の初期値を3種類変えて4万回の学習を行った。そのときのニューロ回路の入出力特性と誤差特性を図3.2.1～図3.2.6にそれぞれ示す。図3.2.1、図3.2.3、図3.2.5は、実線で目標学習関数を示している。また、星印の曲線は、4万回の学習が終了した時の重み係数(入力層中間層間および中間層出力層間)を使用し、ニュー

ロ回路の入出力特性を描いたものである。また、図3.2.2、図3.2.4、図3.2.6は、図3.2.1、図3.2.3、図3.2.5における誤差特性を表示したものである。ここで、図3.2.2、図3.2.4、図3.2.6の誤差曲線についてどのような領域に誤差が生じやすいかを観察することとした。その結果を表3.2.1に示す。最初に、目標学習関数の微分特性のグラフ、図3.1.2と比較した。ここで、目標学習関数微分値の絶対値の第一ピーク(+0.25、-0.25)と第二ピーク(+0.7、-0.7)に誤差が多いことがわかる。言い換えれば、図3.1.3の目標学習関数2回微分値の変曲点で誤差が多い。ここで誤差が多いとは、入力誤差特性の絶対値が0.2より大きい領域を目安とした。これは、3個の初期値を変えた場合でも共通している特性となった。

表3.2.1 規格化しない場合の誤差の多い領域

	誤差の多いX軸での値または領域	
1	-0.8、-0.7、-0.2、0.25、0.6	図3.2.2
2	-0.3、-0.2、0.25、0.7	図3.2.4
3	-0.7、-0.1、0.25、0.8	図3.2.6

図3.2.1 入出力特性 1

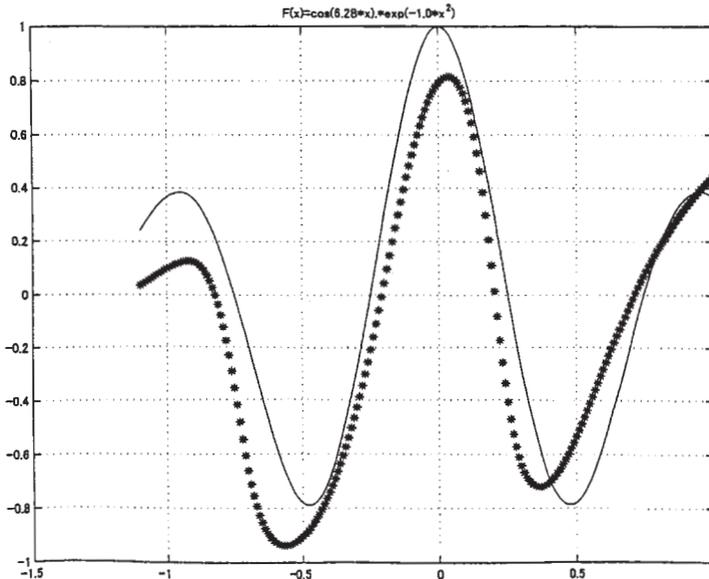


図3.2.2 入力誤差特性 1

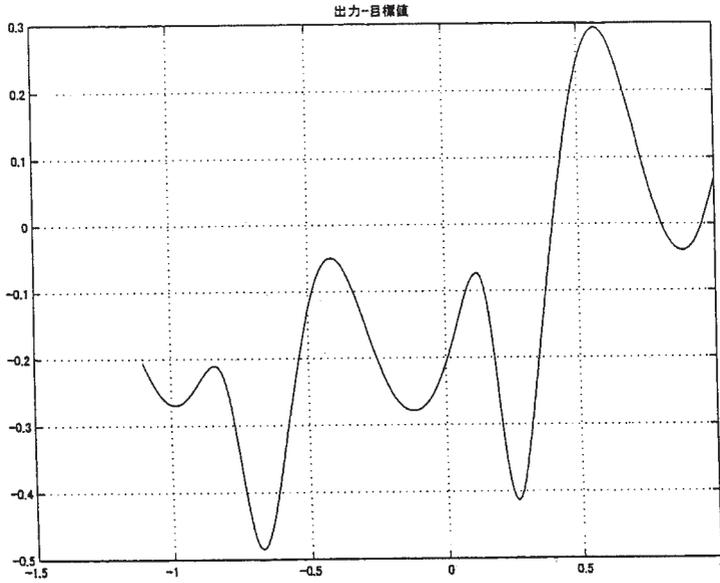


図3.2.3 入出力特性 2

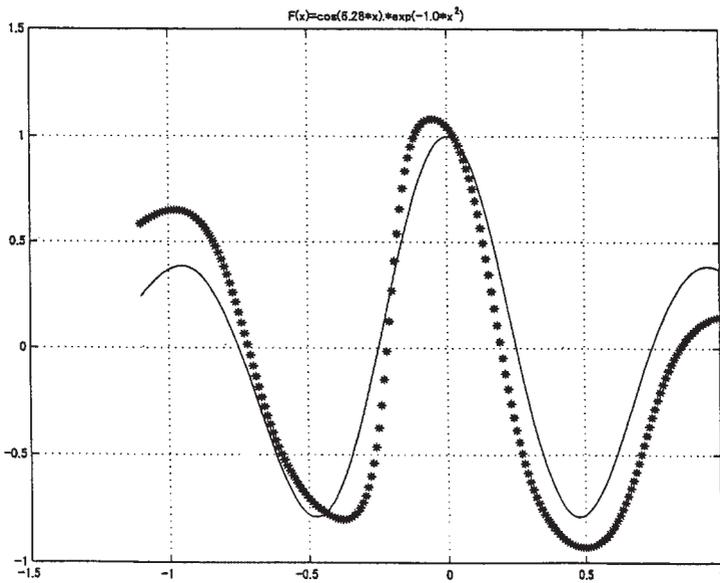


図3.2.4 入力誤差特性 2

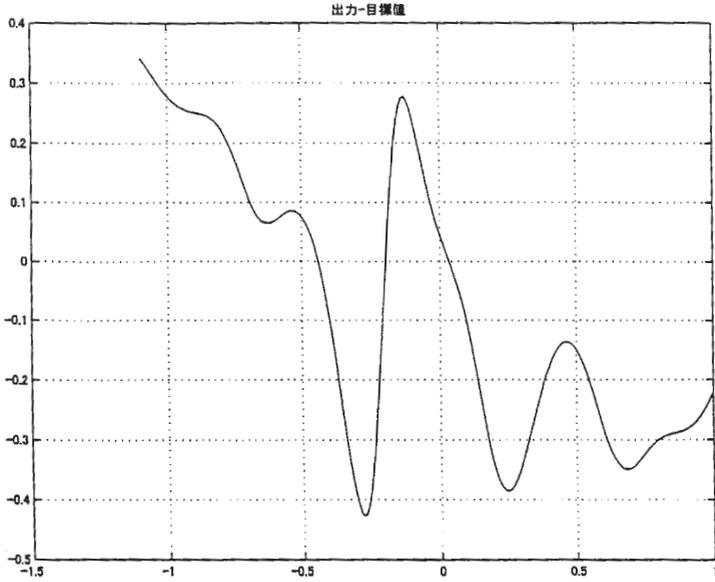


図3.2.5 入出力特性 3

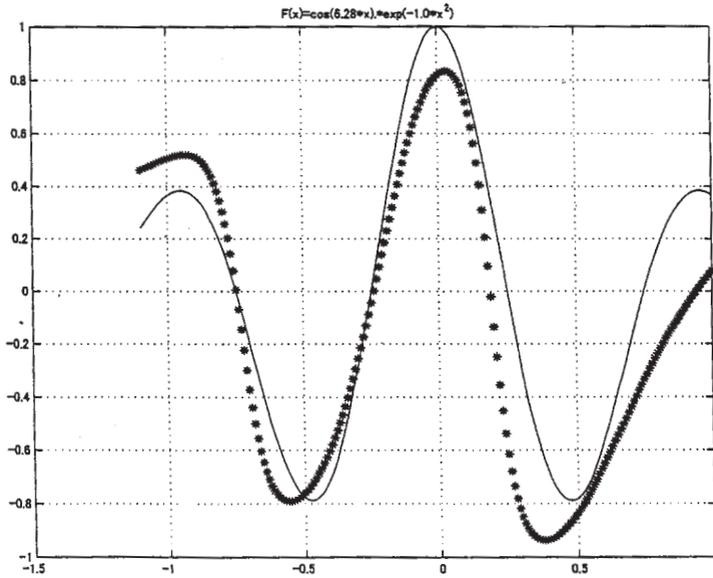
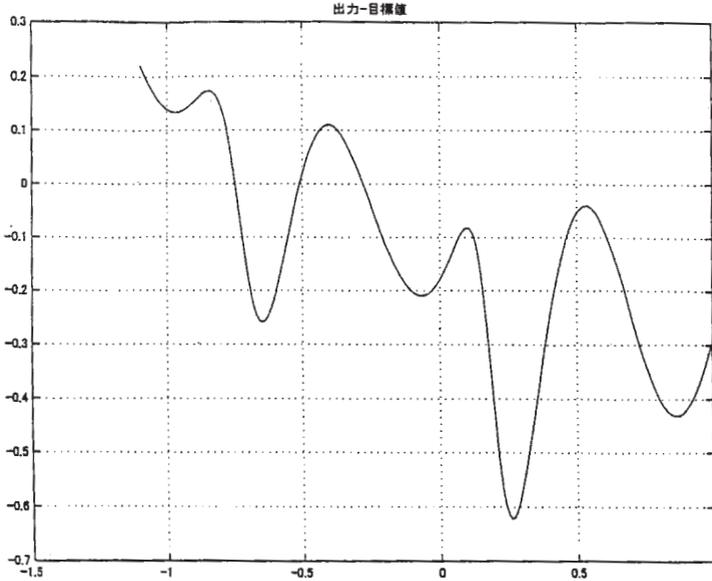


図3.2.6 入力誤差特性 3



3. 出力を規格化した場合の関数学習 ($f(X) = \exp(-X^2) \cos(2\pi X)$)

3.-2. に対して誤差が多いために、ここでは一般的に行われている規格化操作を行った。規格化について簡単に述べる。シグモイド関数の出力範囲は、一般的には0から1である（-1から1の出力を持つ場合もある）。しかし、ニューラルネットワークに期待する出力は、いつも0から1の範囲にあるとは限らない。あれば問題はない。そこでニューラルネットワーク出力に余裕を持たせながら希望の出力を出せるように出力操作をすることになるのである。規格化を行うことで、誤差は減少することが知られているが、その状況はどう変化するのであろうか。出力層の出力を値域が0から1のシグモイド関数を用い学習が行われるように出力を調整する。ここでは、目標学習関数の値域が-0.8から1.0の範囲にあるから、目標学習関数を0.5倍してY軸方向に+0.4平行移動を行った。逆に、4万回の学習終了後、Y軸方向に-0.4平行移動を行い、さらに2倍して元に戻した。

学習率は0.33とした。他の条件は、ほとんど3.-2.と同一である。初期値は乱数を発生させ10種類変えて学習を行った。表3.2.1を見ると初期値を変更して学習を行った結果、 $-0.2 \sim -0.3$ 、0.25および -0.7 のところで誤差が大きくなった。このことは、表3.3.1と比較してみると、 $-0.2 \sim -0.3$ 、0.25、0.5、 -0.7 のいずれかの付近で誤差が大きくなる傾向にあることがわかる。

表3.3.1 規格化した場合の誤差の多い領域

	誤差の多いX軸での値または領域	
1	$-0.2 \sim -0.3$ 、0.25	図3.3.2
2	$-0.2 \sim -0.3$ 、0.25	図3.3.4
3	0.25	図3.3.6
4	$-0.2 \sim -0.3$ 、 -0.7 0.25 0.5	図3.3.8
5	$-0.2 \sim -0.3$ 、0.25	図3.3.10
6	$-0.2 \sim -0.3$ 、0.25	図3.3.12
7	$-0.2 \sim -0.3$ 、0.25 0.5	図3.3.14
8	$-0.2 \sim -0.3$ 、0.25 \sim 0.30	図3.3.16
9	$-0.2 \sim -0.3$ 、0.25	図3.3.18
10	$-0.2 \sim -0.3$ 、0.25	図3.3.20

図3.3.1 入出力特性 1

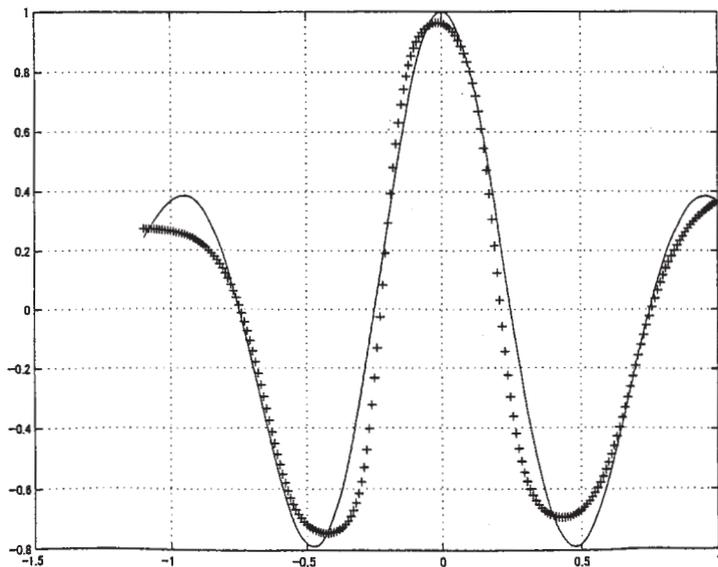


図3.3.2 入力誤差特性 1

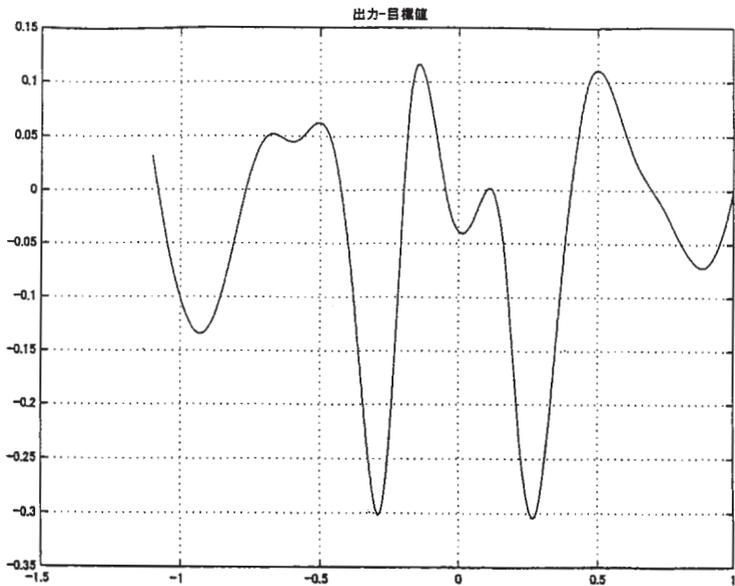


図3.3.3 入出力特性 2

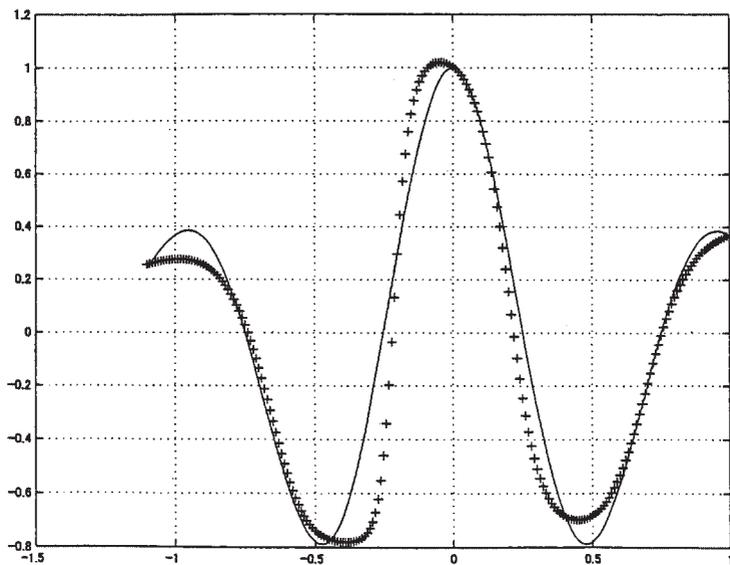


図3.3.4 入力誤差特性 2

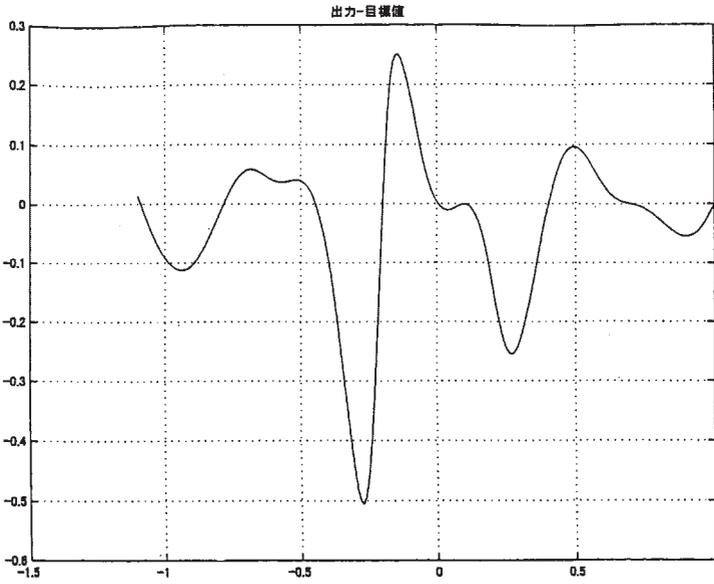


図3.3.5 入出力特性 3

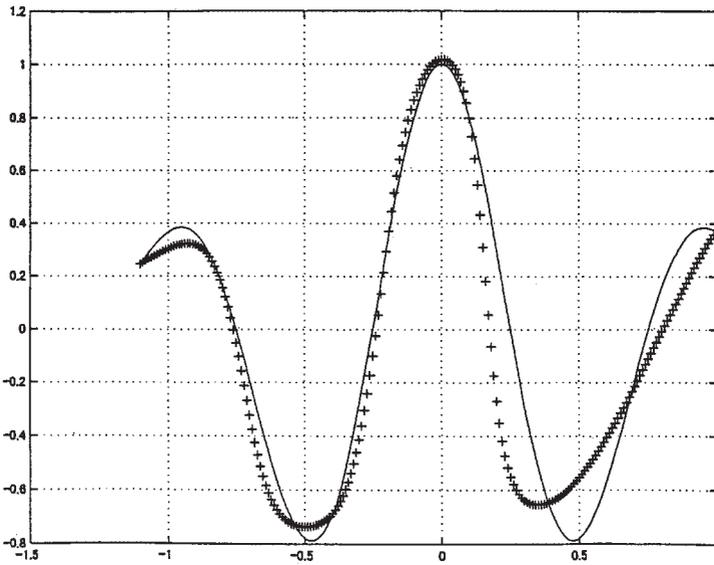


図3.3.6 入力誤差特性 3

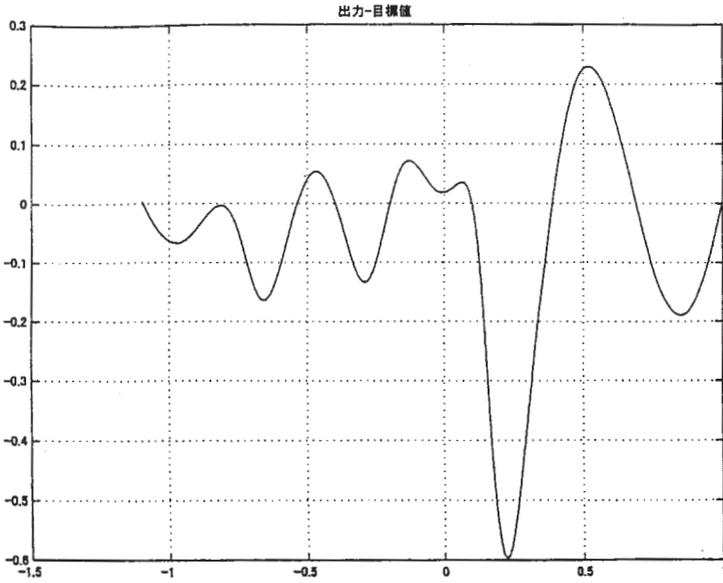


図3.3.7 入出力特性 4

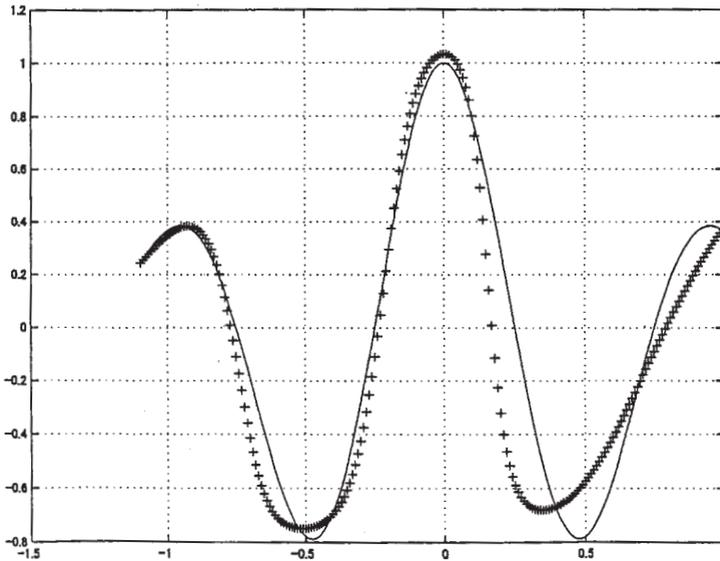


図3.3.8 入力誤差特性 4

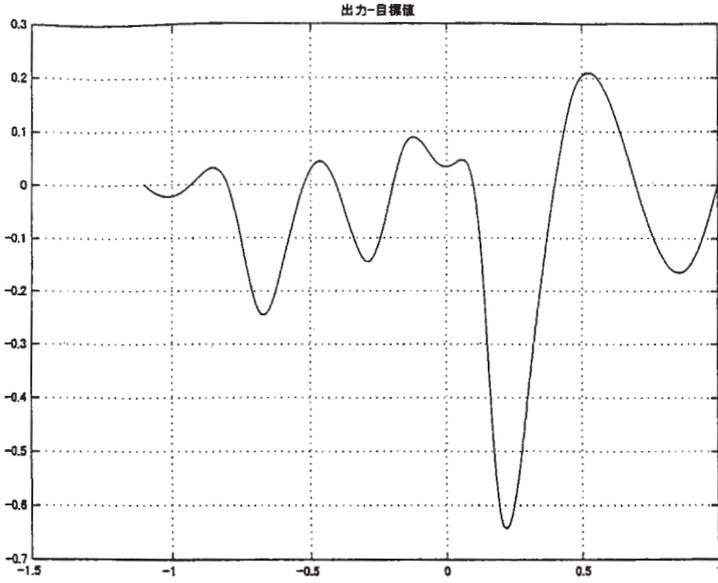


図3.3.9 入出力特性 5

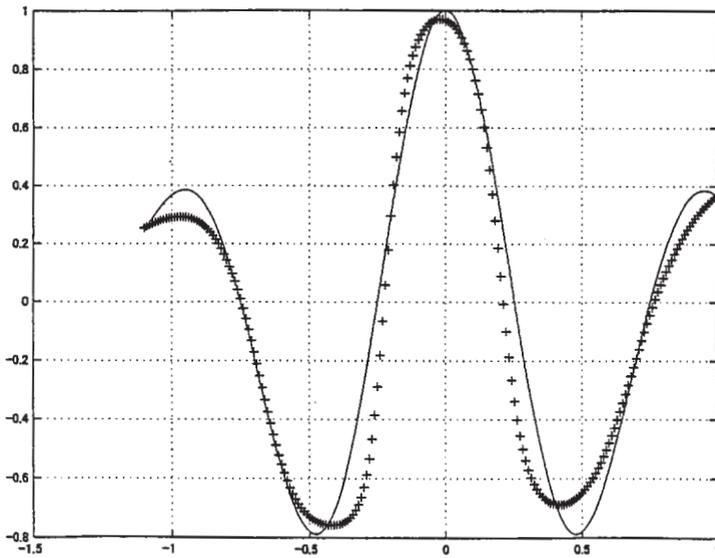


図3.3.10 入力誤差特性 5

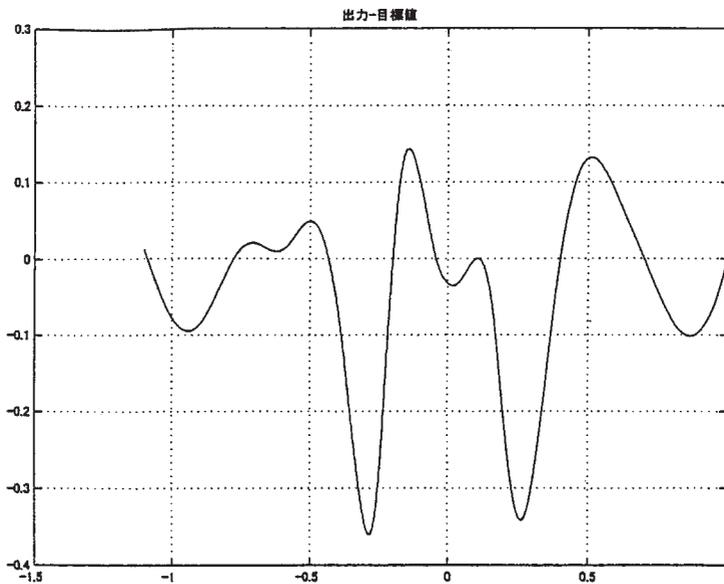


図3.3.11 入出力特性 6

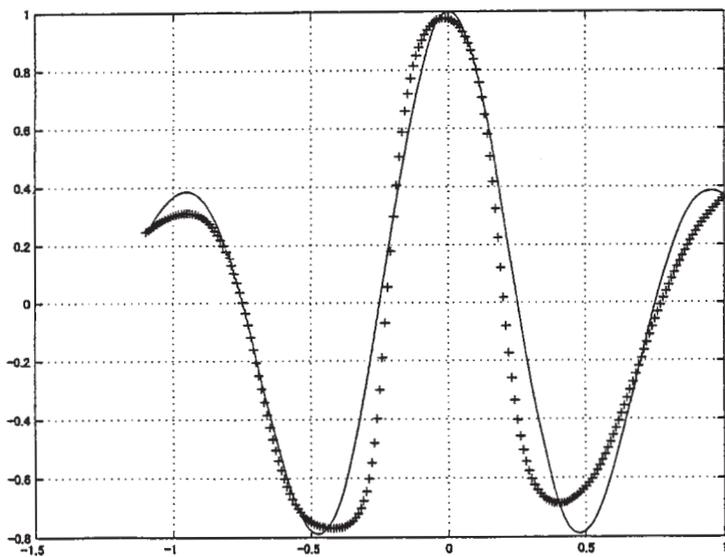


図3.3.12 入力誤差特性 6

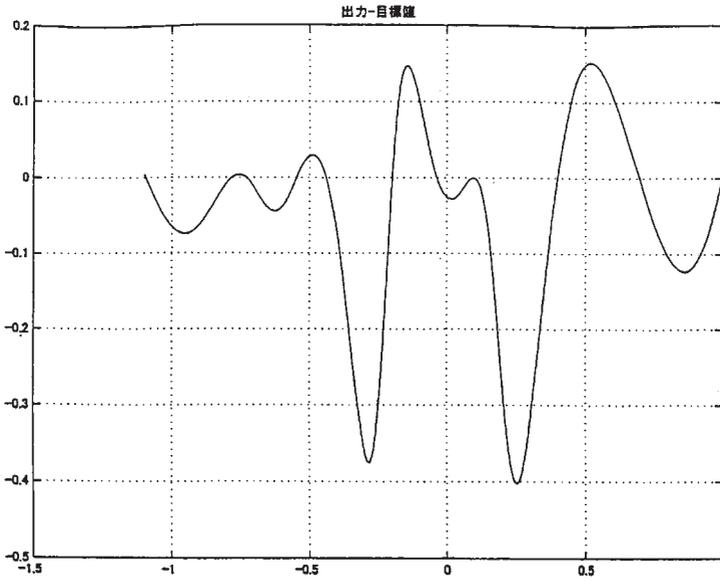


図3.3.13 入出力特性 7

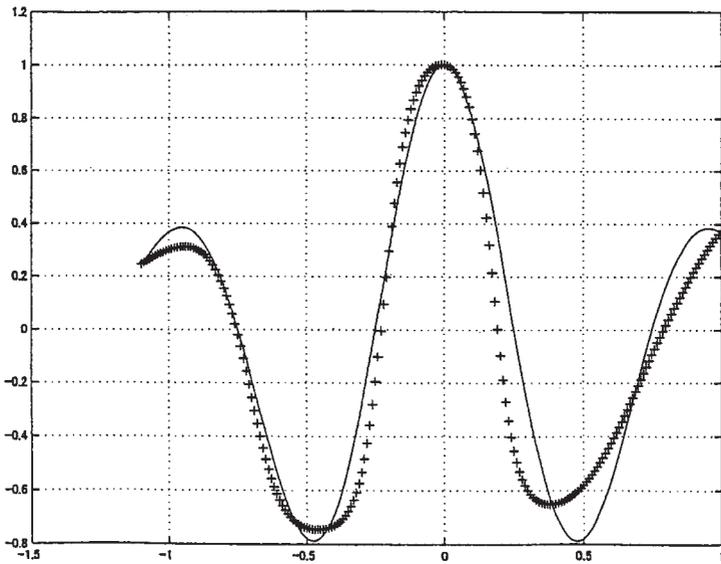


図3.3.14 入力誤差特性 7

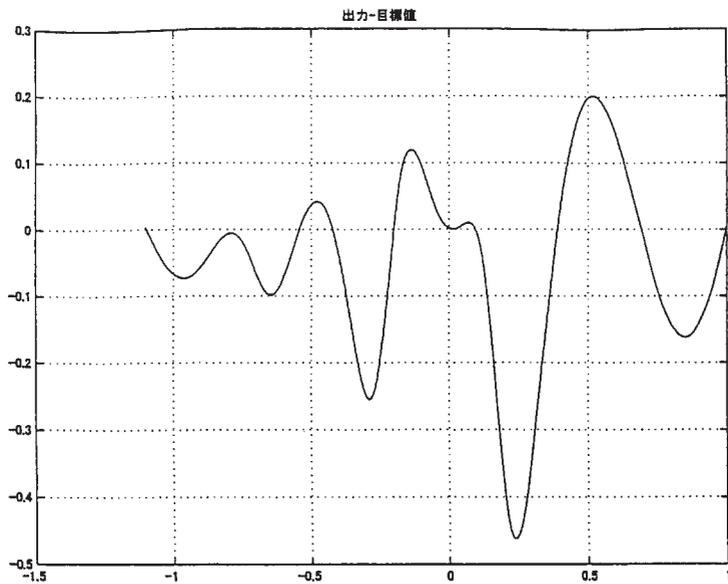


図3.3.15 入出力特性 8

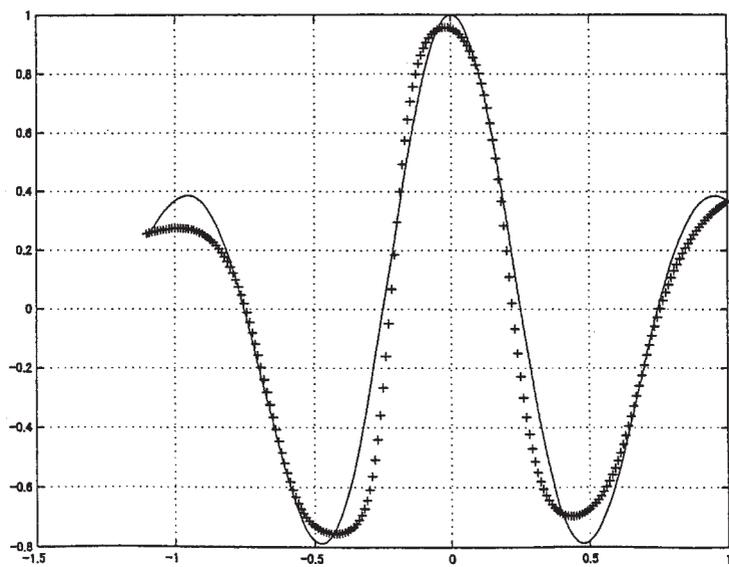


図3.3.16 入力誤差特性 8

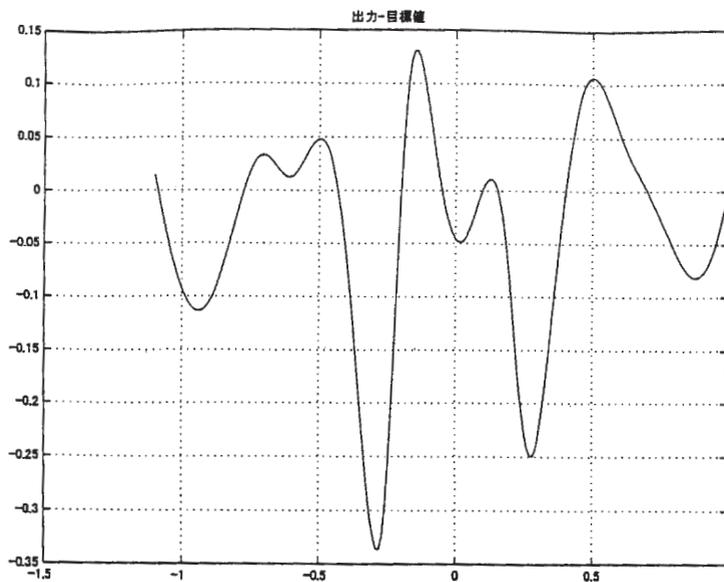


図3.3.17 入出力特性 9

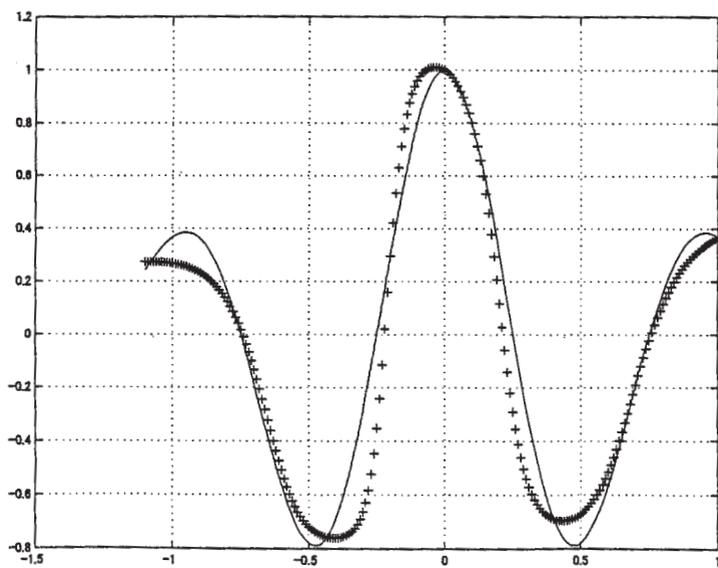


図3.3.18 入力誤差特性 9

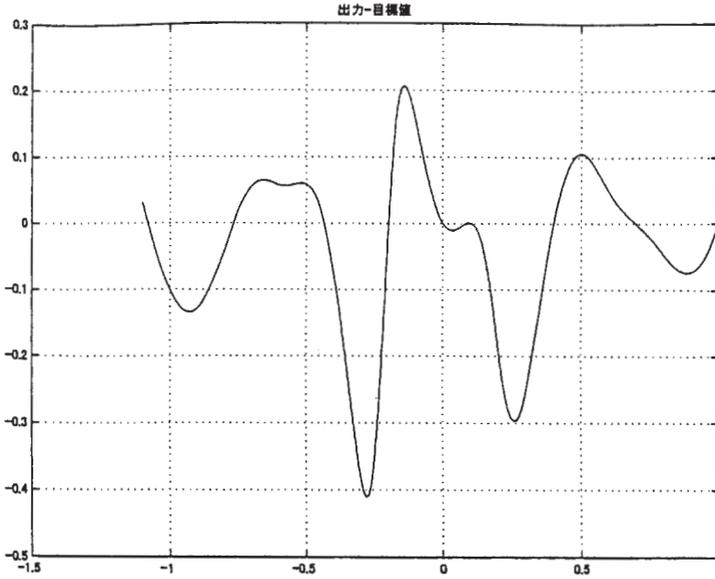


図3.3.19 入出力特性10

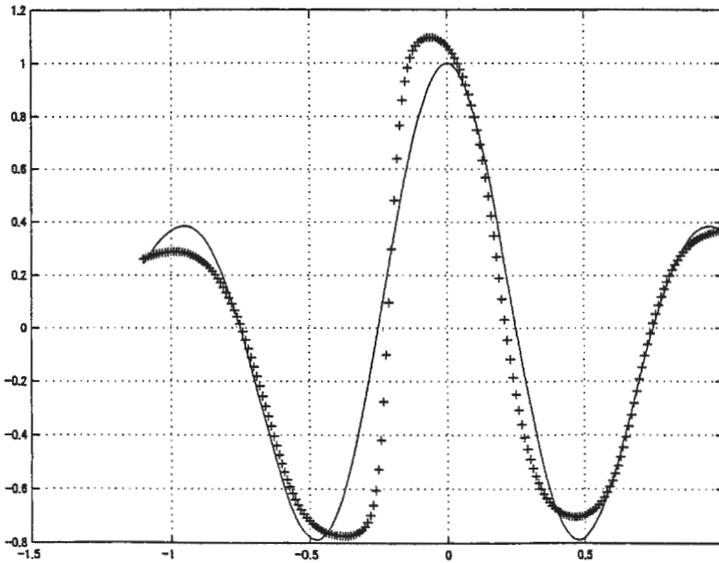
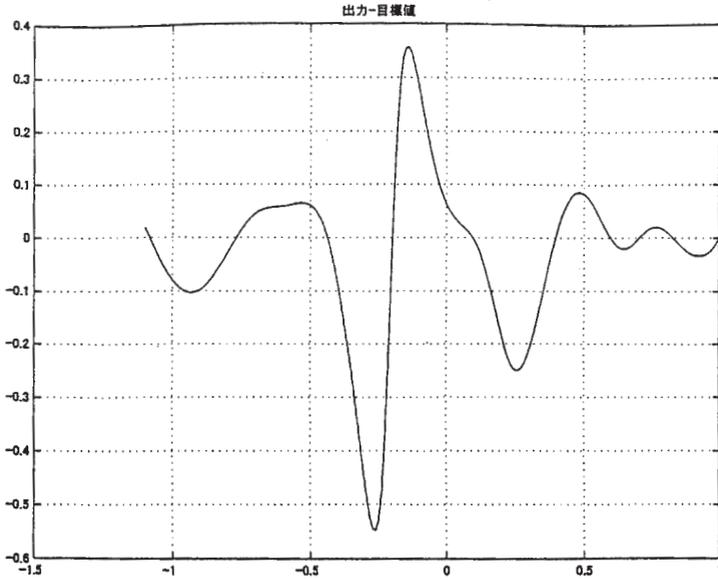


図3.3.20 入力誤差特性10



次に、規格化を行い、条件を同じとし、学習データ数を変えたときの誤差の生ずる領域が変化するかどうかを確認するために、学習するデータ数を7個、9個、11個、15個と変化させた。その結果を表3.3.2～表3.3.5に示す。学習点を変えても誤差の多い領域は、 $0.20 \sim 0.30$ 、 $-0.20 \sim -0.30$ で、どの学習においても誤差が多いという結果となった。また、学習点数を増加させると誤差は減少するけれども、領域は変曲点付近すなわち関数の微分値が正から負に変わる領域(± 0.5 近辺)に多いという共通結果となった。また、これは、学習点数を増加させることによってより明瞭となった。

表3.3.2 規格化した場合の誤差の多い領域

7点学習-1.1、-0.75、-0.40、-0.5、0.3、0.65、1.0	
誤差の多いX軸での値または領域	
1	0.8、0.48、-0.25、-0.55、-0.90
2	0.45、-0.23、-0.55、-0.55
3	0.8、0.48、0.18、-0.25、-0.5、-0.95
4	0.8、0.50、0.20、-0.25、-0.48
5	0.8、0.48、0.20、-0.25、-0.55、-0.9
6	0.8、0.48、0.20、-0.25、-0.55、-0.9
7	0.8、0.48、0.20、-0.25、-0.55、-0.9
8	0.48、0.20、-0.25、-0.55、-0.9
9	0.8、0.48、0.20、-0.25、-0.55、-0.9
10	0.8、0.48、0.20、-0.25、-0.55、-0.9

表3.3.3 規格化した場合の誤差の多い領域

9点学習-1.0、-0.75、-0.5、-0.25、0.0、0.25、0.50、0.75、1.00	
誤差の多いX軸での値または領域	
1	0.20~0.30、0.65、-0.20~-0.30、-1.1
2	0.20~0.30、0.65、-1.1
3	0.20~0.30、-0.20~-0.30、-1.1
4	0.20~0.30、-0.20~-0.30、-1.1
5	0.20~0.30、-0.20~-0.30、-1.1
6	0.20~0.30、-0.20~-0.30、-1.1
7	0.20~0.30、-0.20~-0.30、-1.1
8	0.20~0.30、-0.20~-0.30、-1.1

表3.3.4 規格化した場合の誤差の多い領域

11点学習-1.0、-0.8、-0.6、-0.4、-0.2、0.0、0.2、0.4、0.6、0.8、1.0	
誤差の多いX軸での値または領域	
1	0.20~0.30、-0.20~-0.30、-1.1
2	0.20~0.30、-0.20~-0.30、-1.1
3	0.20~0.30、-0.20~-0.30、-1.1
4	0.20~0.30、-0.20~-0.30、-1.1
5	0.20~0.30、-0.20~-0.30、-1.1
6	0.20~0.30、-0.20~-0.30、-1.1
7	0.20~0.30、-0.20~-0.30、-1.1
8	0.20~0.30、-0.20~-0.30、-1.1
9	0.20~0.30、-0.20~-0.30、-1.1
10	-0.20~-0.30、-1.1

表3.3.5 規格化した場合の誤差の多い領域

15点学習-1.0、-0.95、-0.8、-0.65、-0.5、-0.35、-0.2、-0.05、0.1、0.25、0.4、0.55、0.70、0.85、1.0	
	誤差の多いX軸での値または領域
1	0.48、0.20~0.30、-0.20~-0.30、-0.48
2	0.48、0.20~0.30、-0.20~-0.30、-0.48
3	0.48、0.20~0.30、-0.20~-0.30、-0.48
4	0.48、0.20~0.30、-0.20~-0.30、-0.48
5	0.48、0.20~0.30、-0.20~-0.30、-0.48

4. 出力を規格化しない場合の関数学習 ($f(X) = 0.1\sin 8X$)

規格化を行わないにもかかわらず、誤差特性が比較的良好な関数もある。3.3. で行った実験での関数は、文献 (1) にも紹介されているようによく検査関数として使われている。ここでは、それに比べて、出力値の絶対値が比較的小さく、振動周期はそれほど変わらない振動関数を目標関数に選んだ。ここでは、目標関数の値域の大きさ (-0.1から+0.1) によって誤差領域がどうなるかの実験を行った。出力を規格化しない場合の関数学習 ($f(X) = \exp(-X^2)\cos(2\pi X)$) とほとんど同じ条件で実験を行った。関数 $f(X) = 0.1\sin 8X$ は、最小値が -0.1、最大値が 0.1 である。また、微分値の最大値が 0.8、最小値が -0.8 である。さらに、X の学習領域は、-0.7 ~ +0.8 とし、学習点は、X 軸上 -0.7、-0.4、-0.1、0.2、0.5、0.8 の 6 点とした。学習率は 0.4 とし、重み係数の初期値は 5 種類変化させて実験を行った。図 3.4.1 における実線は、目標関数 $f(X) = 0.1\sin 8X$ を示し、星印曲線は、4 万回学習後のニューロ回路の入出力特性を示す。同様に、重み係数の初期値を変えた結果を図 3.4.3、図 3.4.5、図 3.4.7、図 3.4.9 に示す。また、図 3.4.2、図 3.4.4、図 3.4.6、図 3.4.8、図 3.4.10 には、入力誤差特性と、 $f(X) = 0.1\sin 8X$ の 2 回微分関数の 0.01 倍の曲線を描いた。これは、目標関数の変曲点を表示するために描いたものである。図 3.4.1 および図 3.4.3、図 3.4.5、図 3.4.7、図 3.4.9 のみ見ると、誤差の大きさは、比較的小さい。多く生じる領域は図 3.4.2、図 3.4.4、図 3.4.6、図 3.4.8、図 3.4.10 を見ると、表 3.4.1 に示すように、目標関数の変曲点、すなわち -0.2、0.2、-0.6、0.6 を含む領域に誤差が多いことがわかる。ここでは、誤差が比較的小さいために絶対値が 0.02 以上になる領域を誤差の多い領域とした。

表3.4.1 規格化しない場合の誤差の多い領域

	誤差の多いX軸の値または領域	
1	<u>-0.68~-0.45</u> <u>-0.3~-0.2</u>	図3.4.2
2	<u>-0.70~-0.4</u> <u>0.08~0.12</u> <u>0.52~0.72</u>	図3.4.4
3	<u>-0.68~-0.45</u> <u>-0.32~-0.16</u>	図3.4.6
4	<u>-0.7~-0.4</u> <u>0.08~0.15</u> <u>0.5~0.8</u>	図3.4.8
5	<u>-0.65~-0.44</u> <u>-0.28~-0.2</u>	図3.4.10

図3.4.1 入出力特性 1

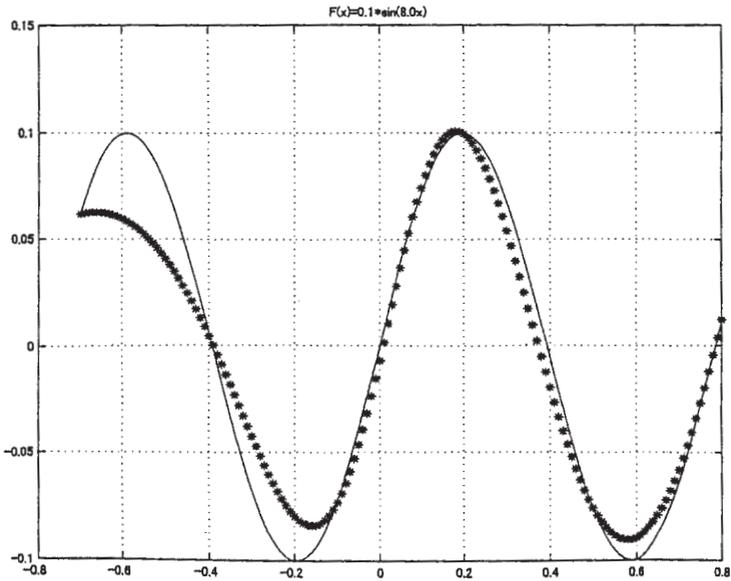


図3.4.2 入力誤差特性 1

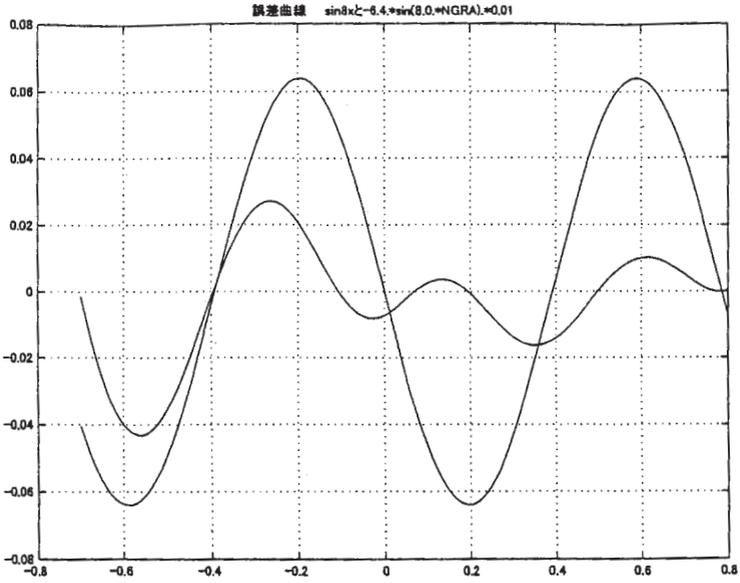


図3.4.3 入出力特性 2

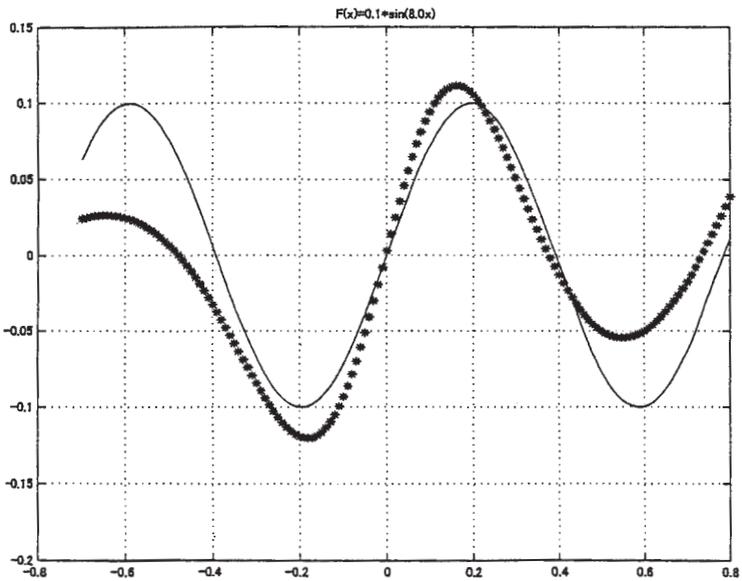


図3.4.4 入力誤差特性 2

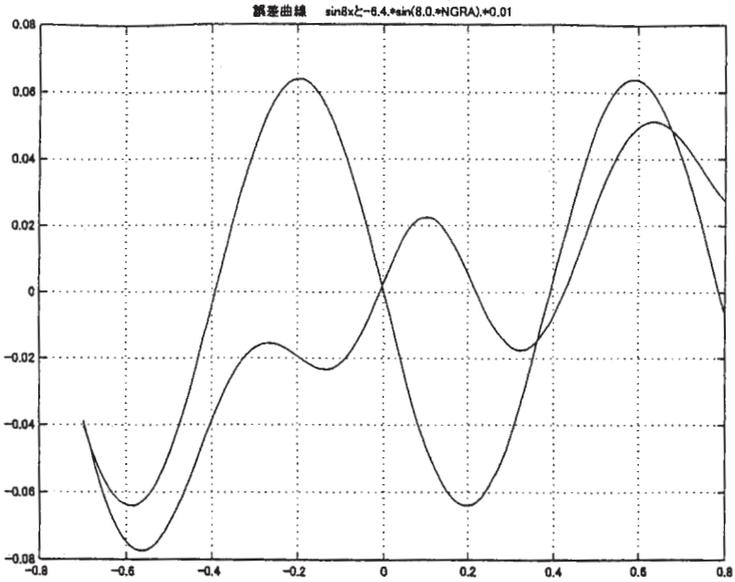


図3.4.5 入出力特性 3

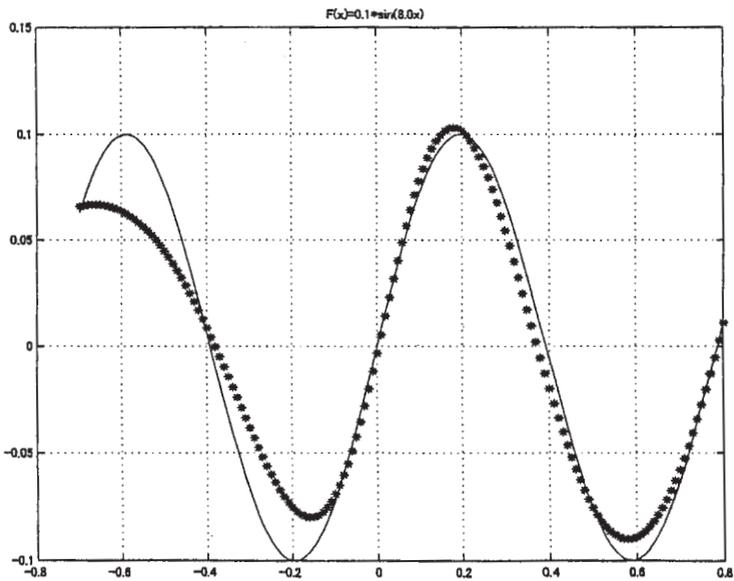


図3.4.6 入力誤差特性 3

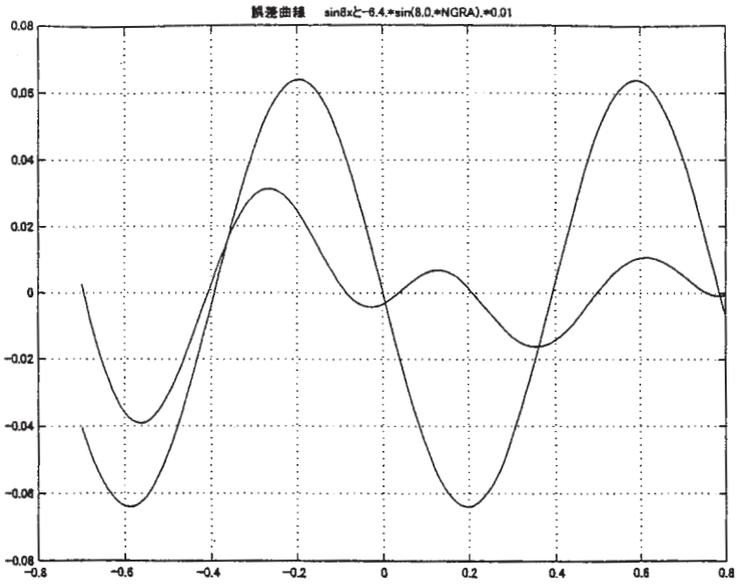


図3.4.7 入出力特性 4

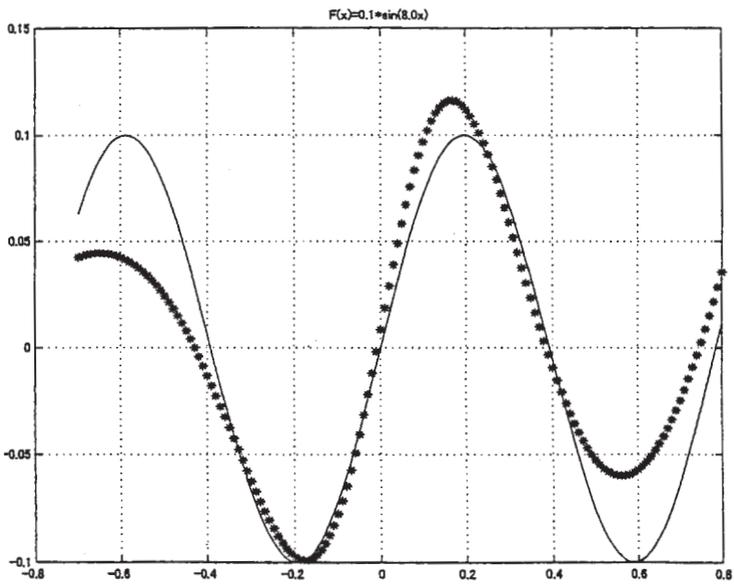


図3.4.8 入力誤差特性 4

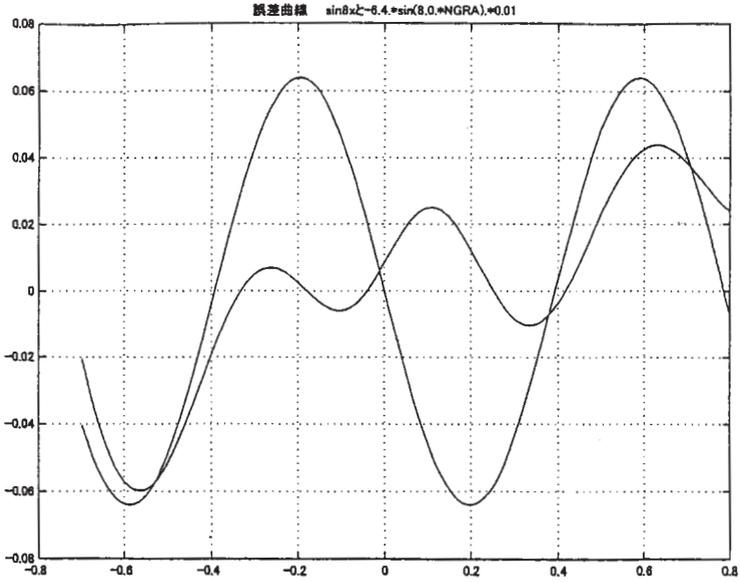


図3.4.9 入出力特性 5

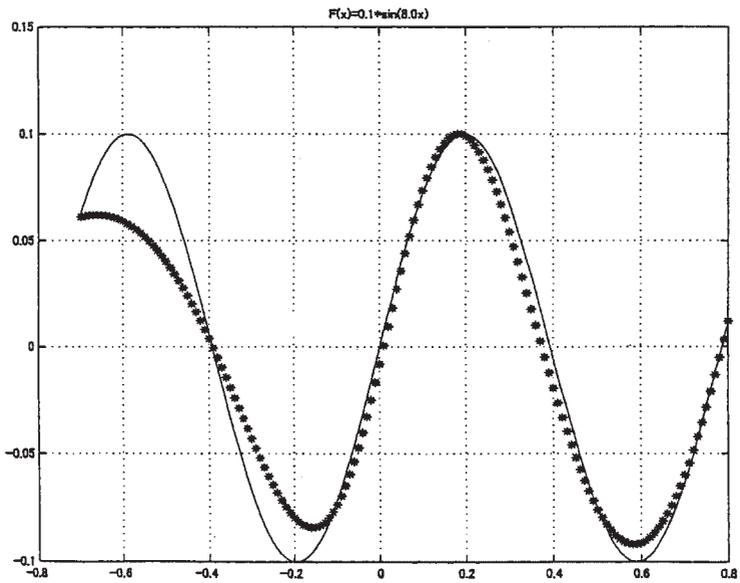
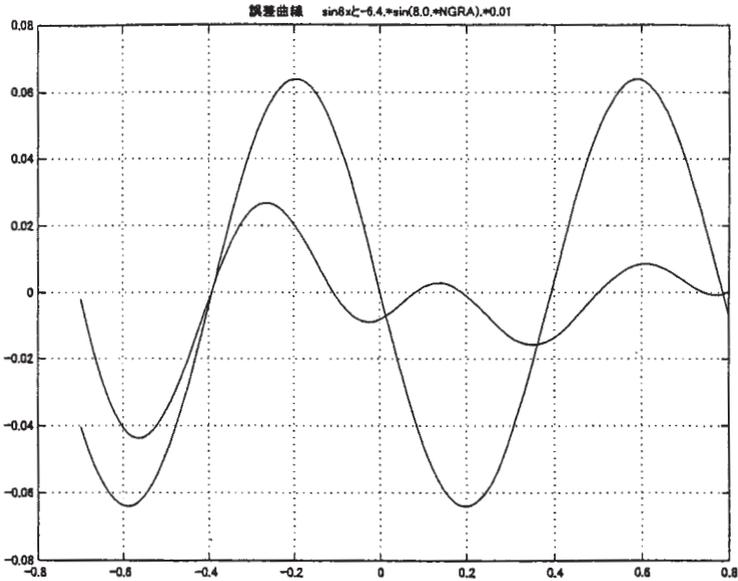


図3.4.10 入力誤差特性 5



4. 考 察

ニューラルネットワークを用いて関数を学習する場合、計算機実験 3.-1.~3.-3. を見て共通して言えることは、変曲点付近すなわち目標関数の傾きの符号が変わる領域で誤差が大きくなる場合が多いという結果となった。また、ニューラルネットワークでの学習では、どのような関数でも学習が可能とされているが、誤差が比較的大きくなり関数特性が、目標関数に対して、ゆがんだ入出力特性が生じてしまう場合がある。この場合は、学習回数を増加させても誤差は一定値以上は減少しなかった。規格化をすると、学習点付近は、誤差が少なくなるが、全体的には、ある程度減少するものの、3.-3. の関数ほど減少はしない。3.-2. と 3.-3. を比較すると、目標関数の微分値が、後者のほうが変化の幅が大きいため追従できないのではないかと考えられる。同じ定義域の幅に対して、値域が急激になると、

ニューラルネットワークでは誤差が大きくなってしまわないかとも考えられる。

また変曲点付近でも、誤差が比較的少なくなっている場合もあった。規格化した場合は、学習点付近は誤差が少なくなっていた。規格化しない場合は、学習点であっても誤差が多い傾向にあった。学習点と変曲点の組み合わせで入力誤差特性は変化するものであるのか。

今後、学習関数の変曲点と学習点の選択によって、さらに誤差の出やすい関数であっても誤差を小さくするための方法を考察したいと考える。

5. おわりに

ニューラルネットワークを用いて関数を学習することはよく知られているが、誤差に対する検討は、あまり行われていない。本稿では、2種類の関数の検討を行った。3.-1. および 3.-2. の誤差の多い関数に対して共通して言えることは、学習関数の変曲点付近では、誤差が大きくなるということがわかった。また、規格化によって、誤差の改善は、良好な結果となった。

しかし、誤差特性の比較的小さい関数に対しての関数学習に対しては、変曲点付近に誤差は多いものの、そうではない領域にも誤差の多い領域があることが確認された。今後、さらなる検討を試みたい。

(参考文献)

- (1) 松井伸之・石見憲一「しきい値ゆらぎをもつニューロンモデルを用いた階層型ニューラルネットワーク」『電気学会論文誌C』第114巻第11号、1994年、1208-1213ページ。
- (2) 王婷・須貝康雄「非線形多変数関数近似のためのウェーブレットニューラルネットワーク」『電気学会論文誌C』第120巻第2号、2000年、185-193ページ。