

# 3層ニューラルネットワークの 関数学習の規格化による誤差変化

田口 功\* / 須貝 康雄\*\*

## Error Change by the Standardization of Function Learning of the Three-Layer Neural Networks

Isao TAGUCHI and Yasuo SUGAI

The function learning which generalization ability of the neural network was used for is well known. Even if it increases learning a number of times, the improvement of error may not occur. Therefore, generally the operation of standardization occurs. In this paper, it is shown that an error characteristic varies by the way of standardization, in particular, the degree of the reduction of the goal function. Although considering that errors decrease when a learning interval is made smaller, it was shown that there is a limit.

---

\*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しばしば行われている。ニューラルネットワークの学習でしばしばガウス関数が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（文献〔1〕）。どのような関数でも学習回数を増加させれば誤差が減少すれば問題はない。実際には、学習しにくい関数や学習しやすい関数が存在する。本稿では、誤差が比較的大きな関数（文献〔2〕）をモデルとして実験を行い、出力特性および誤差特性を基にし考察を行う。目標関数の縮小率（以下 *syuku* と表示）と平行移動（以下 *heiko* と表示）操作による規格化の違いにより、出力特性および誤差特性が大きく異なることを示す。規格化とは、出力層のシグモイド関数の出力範囲（0 から 1）と目標関数の出力範囲が異なるときの調整操作を言う。

## 2. 3層ニューラルネットワークの構成

図1に3層ニューラルネットワークの構成図を示す。ここでは、入力層、中間層、出力層からなる3層ニューラルネットワークを考える。ここで、 $x^{(p)}$ 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。中間層、出力層には、シグモイド関数を使用する。

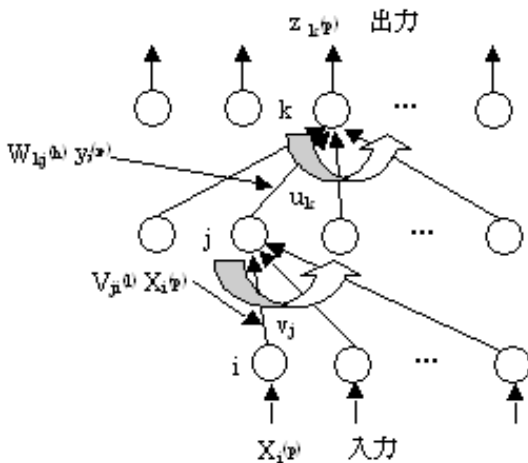
学習は、

$$E = \sum E^{(p)} = \sum \sum \{(t_k^{(p)} - z_k^{(p)})^2 / 2\} \quad (1)$$

の総誤差  $E$  を最小化することである。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、

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

図1 3層ニューラルネットワークの構成



となる。この  $E^{(p)}$  を減少させるように  $W_{kj}^{(h)}$  (中間層と出力層の間の重み係数) と  $V_{ji}^{(h)}$  (入力層と中間層の間の重み係数) を誤差逆伝搬学習により調節することになる。 $\Delta W_{kj}^{(h)}$  および  $\Delta V_{ji}^{(h)}$  は、

$$\Delta W_{kj}^{(h)} = -\varepsilon \partial E^{(p)} / \partial W_{kj}^{(h)} \quad (3)$$

と

$$\Delta V_{ji}^{(h)} = -\varepsilon \partial E^{(p)} / \partial V_{ji}^{(h)} \quad (4)$$

となる (本稿では最急降下法を使用)。ここで、 $\varepsilon$  は学習係数であり本稿では 1 として実験を行った。また、

$$-\partial E^{(p)} / \partial W_{kj}^{(h)} = (t_k^{(p)} - z_k^{(p)}) z_k^{(p)} (1 - z_k^{(p)}) \cdot y_j^{(p)} \quad (5)$$

が成り立つ。ここで、 $\delta_k^{(h)} = (t_k^{(p)} - z_k^{(p)}) \cdot z_k^{(p)} (1 - z_k^{(p)})$  とおくと、

$$-\partial E^{(p)} / \partial W_{kj}^{(h)} = \delta_k^{(h)} \cdot y_j^{(p)} \quad (6)$$

となる。従って、

$$\Delta W_{kj}^{(h)} = \varepsilon \delta_k^{(h)} y_j^{(p)} = \varepsilon \cdot (t_k^{(p)} - z_k^{(p)}) z_k^{(p)} (1 - z_k^{(p)}) y_j^{(p)} \quad (7)$$

となる。同様に偏微分の連鎖律を考慮すると、

$$-\partial E^{(p)} / \partial V_{ji}^{(h)} = -\partial E^{(p)} / \partial v_j \cdot x_i^{(p)} \quad (8)$$

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

$$\delta_j^{(h)} = -\partial E^{(p)} / \partial v_j \cdot \partial y_j^{(p)} / \partial v_j$$

$$= -\partial E^{(p)} / \partial y_j^{(p)} \cdot y_j^{(p)} (1 - y_j^{(p)}) \quad (9)$$

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

$$\begin{aligned} \partial E^{(p)} / \partial y_j^{(p)} &= \sum \partial E^{(p)} / \partial u_k \cdot \partial u_k / \partial y_j^{(p)} \\ &= \sum \delta_k^{(h)} W_{kj}^{(h)} \end{aligned} \quad (10)$$

となるから、

$$\delta_j^{(l)} = (\sum \delta_k^{(h)} W_{kj}^{(h)}) y_j^{(p)} (1 - y_j^{(p)}) \quad (12)$$

とおくと、

$$\begin{aligned} \Delta V_{ji}^{(l)} &= \varepsilon \delta_j^{(l)} x_i^{(p)} \\ &= \varepsilon (\sum \delta_k^{(h)} W_{kj}^{(h)}) y_j^{(p)} (1 - y_j^{(p)}) x_i^{(p)} \end{aligned} \quad (13)$$

となる。

### 3. 計算機実験および考察

#### 1. 目標関数

本稿でニューラルネットワークを利用して関数学習を行う。目標とする関数は、

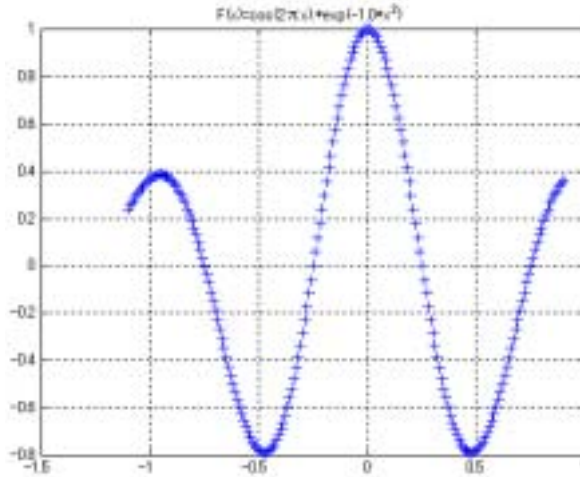
$$f(\mathbf{X}) = \exp(-\mathbf{X}^2) \cos(2\pi \mathbf{X}) \quad (14)$$

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

#### 2. 規格化について

規格化とは出力層の出力範囲が限定されているために、出力層の出力範囲と目標関数の出力範囲を拡大したり、縮小したり、平行移動したりし、調整することである。目標学習 ( $f(\mathbf{X}) = \exp(-\mathbf{X}^2) \cos(2\pi \mathbf{X})$ ) をそのまま学習する場合、色々問題が生じる。ネットワークの出力層の出力範囲は0から1である。従って、目標関数の定義域を-1.1から1.0とすると、図1からわかるように値域は、-0.8から+1.0であるから、単純に考えてもそのままでは学習は困難である。文献(3)にはこのことについて報告を行っ

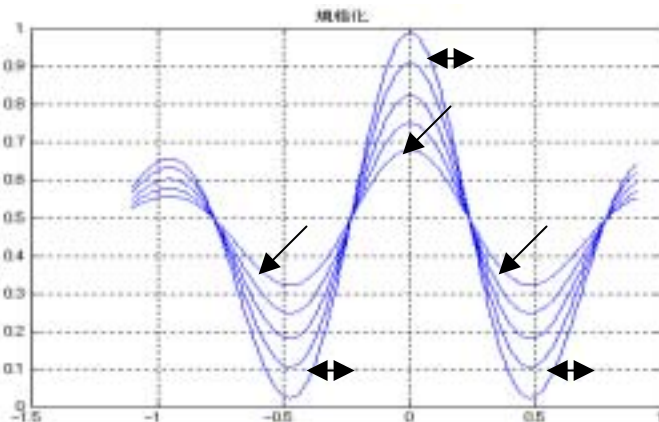
図2 目標関数



た。すなわち、本目標関数は、規格化を行わないとき誤差が非常に多くなり、学習回数を増加させてもそれほど減少しなかったが、規格化を行った結果、誤差が大分改善されたことである。同時に、誤差の生じる範囲も考察した。本稿では、さらに進んで、規格化の仕方によってどの程度誤差が変化するかを実験的に考察する。規格化は縮小率を0.20、0.28、0.36、0.45、0.54の5種類行ったグラフを図3に示す。また、縮小した結果、その中央値が0.5になるように上下方向に平行移動を行った。それらの値は、それぞれ0.48、0.47、0.464、0.46、0.45とした。syuku=0.20のとき、heiko=0.48が対応する。このうち出力層の出力範囲いっばいに使用した規格化は、syuku=0.54、heiko=0.45である。図3において、値域に対して、0から1に一番近い特性がそれを示す。なおここでは、syuku=0.20、syuku=0.36、syuku=0.54の3種について取り上げる。

ここで、図3において、5本の曲線は、出力層の出力範囲0から1の中心0.5が、目標関数の地域の中心(0.1)になるように調整を行った。一方方向矢印のついている曲線が縮小率0.20、平行移動量0.48の規格化曲線である(syuku=0.20、heiko=0.48)。逆に、出力層出力範囲いっばいに使用できるように規格化した特性は、両方向の矢印をつけた(syuku=0.54、

図3 目標関数の規格化



heiko=0.45)。

## 4. 計算機実験

### 1. 縮小率0.20、平行移動量0.48のとき (syuku=0.20、heiko=0.48)

ここでは、学習回数は、すべて2000回とした。また、学習率は、1.0一定とし重み係数の更新操作を行った。1入力1出力とし、中間層のニューロン数は、10個とし固定した。目標関数を学習するとき、入力の刻み幅を0.02、0.03、0.04、0.05、0.06、0.07、0.08、0.1、0.12、0.14、0.15、0.16、0.18、0.20と計14通りの実験を行い、その変化を調べた。刻み幅を0.02とした場合の2000回学習後の入出力特性を図4.1に、誤差特性を図4.2に示す。同様に、刻み幅を0.2まで変化させた場合の入出力特性および誤差特性を図4.27および図4.28までに示す。2000回の学習では、刻み幅0.05のときが一番良いが、まだ良好とはとても言えない。Syuku=0.20では、刻み幅を色々変化させても良好な学習は行われなかった。

図4.1 入出力特性 (刻み幅0.02)

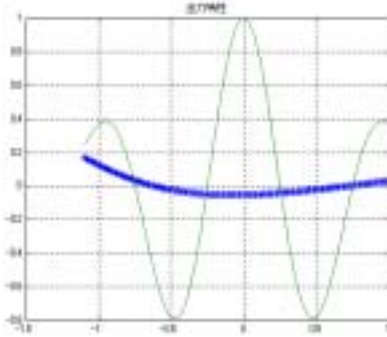


図4.2 誤差特性 (刻み幅0.02)

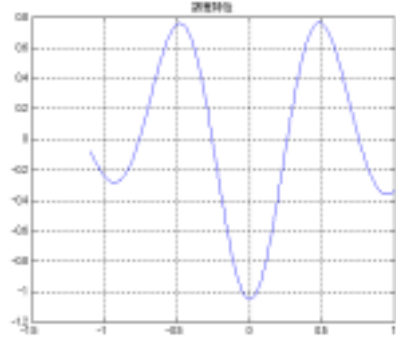


図4.3 入出力特性 (刻み幅0.03)

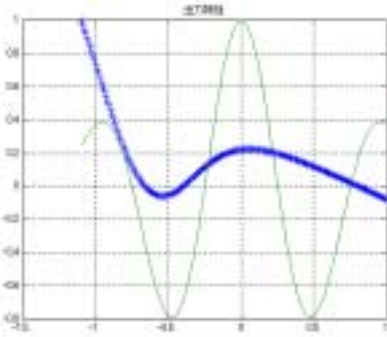


図4.4 誤差特性 (刻み幅0.03)

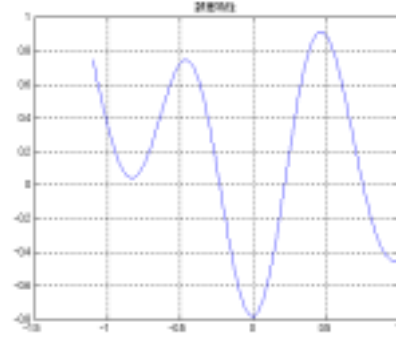


図4.5 入出力特性 (刻み幅0.04)

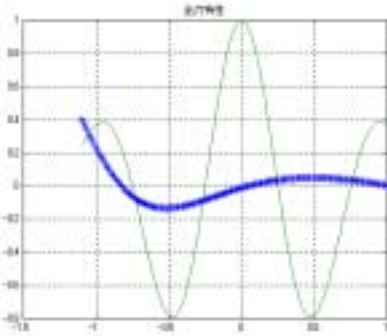


図4.6 誤差特性 (刻み幅0.04)

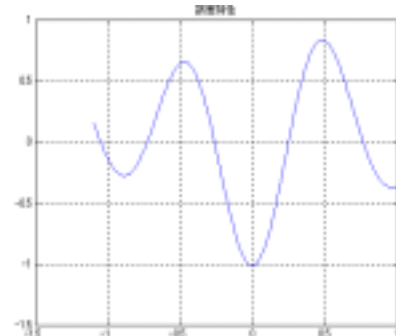


図4.7 入出力特性 (刻み幅0.05)

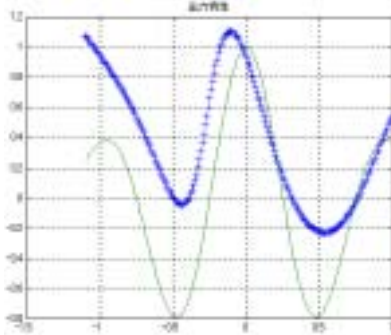


図4.8 誤差特性 (刻み幅0.05)

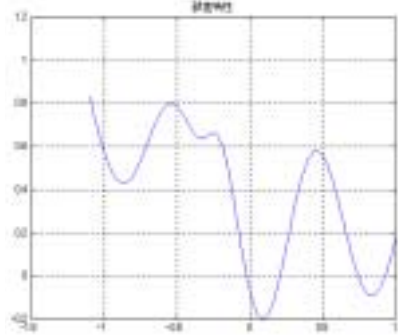


図4.9 入出力特性 (刻み幅0.06)

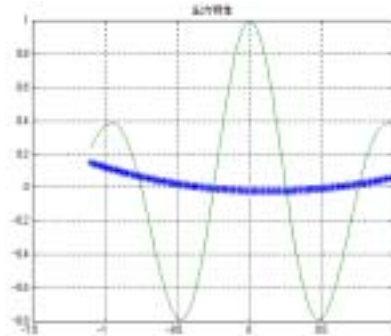


図4.10 誤差特性 (刻み幅0.06)

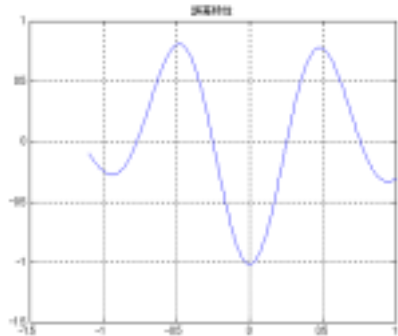


図4.11 入出力特性 (刻み幅0.07)

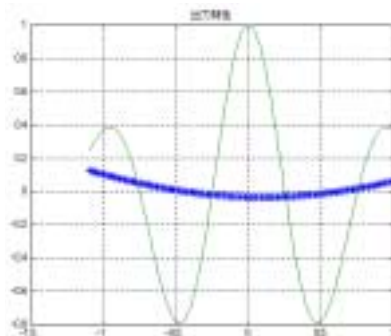


図4.12 誤差特性 (刻み幅0.07)

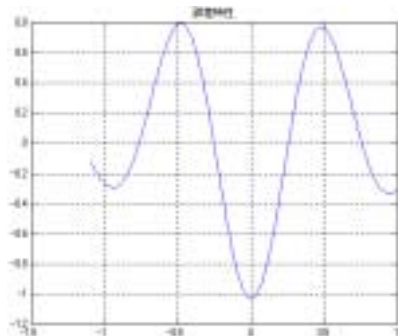




図4.13 入出力特性 (刻み幅0.08)

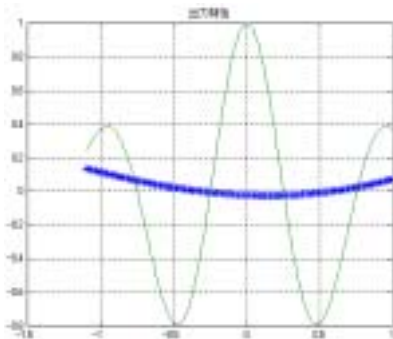


図4.14 誤差特性 (刻み幅0.08)

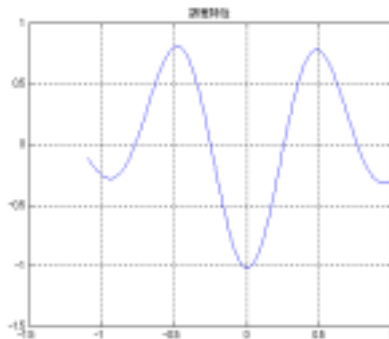


図4.15 入出力特性 (刻み幅0.10)

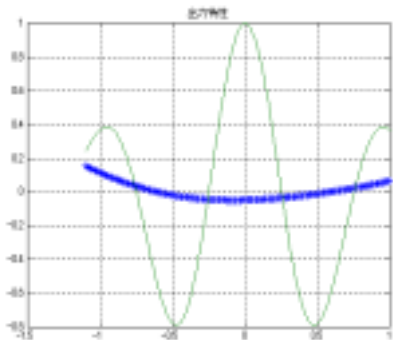


図4.16 誤差特性 (刻み幅0.10)

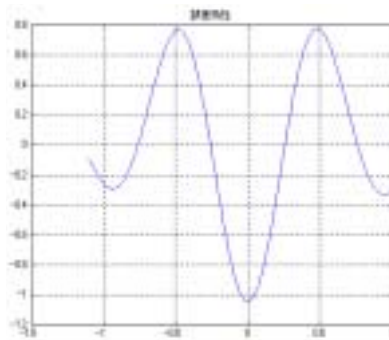


図4.17 入出力特性 (刻み幅0.12)

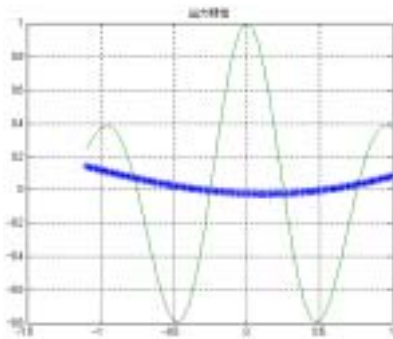


図4.18 誤差特性 (刻み幅0.12)

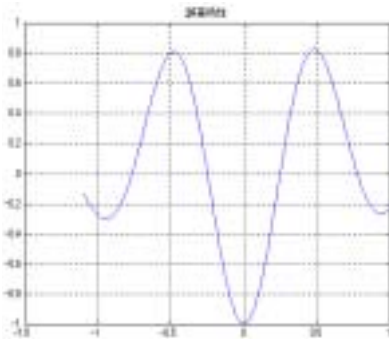


図4.19 入出力特性 (刻み幅0.14)

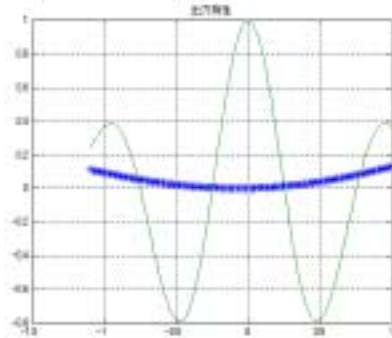


図4.20 誤差特性 (刻み幅0.14)

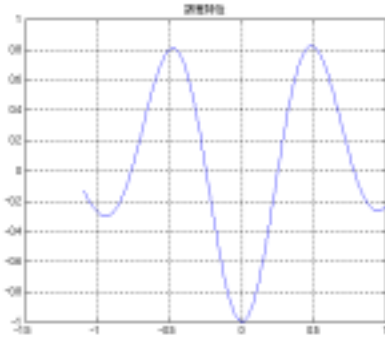


図4.21 入出力特性 (刻み幅0.15)

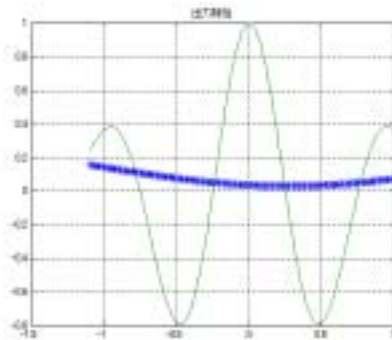


図4.22 誤差特性 (刻み幅0.15)

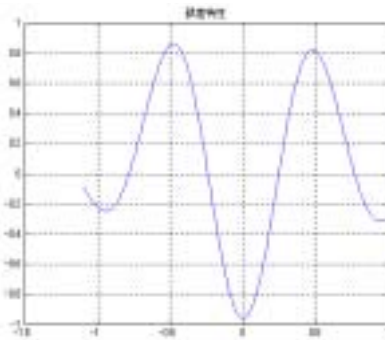


図4.23 入出力特性 (刻み幅0.16)

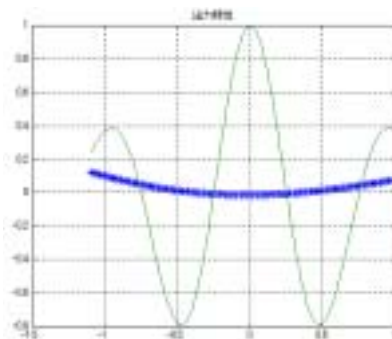


図4.24 誤差特性 (刻み幅0.16)

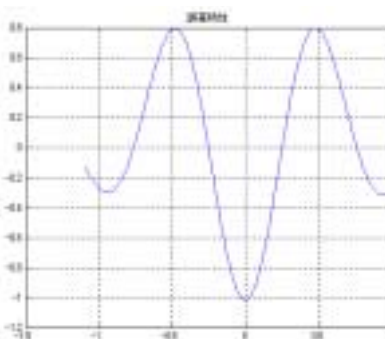


図4.25 入出力特性 (刻み幅0.18)

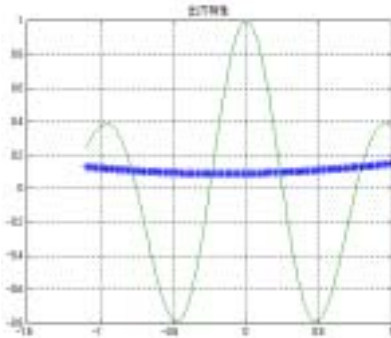


図4.26 誤差特性 (刻み幅0.18)

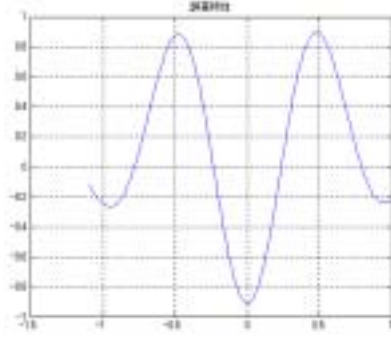


図4.27 入出力特性 (刻み幅0.20)

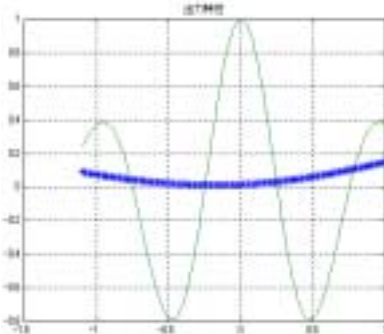
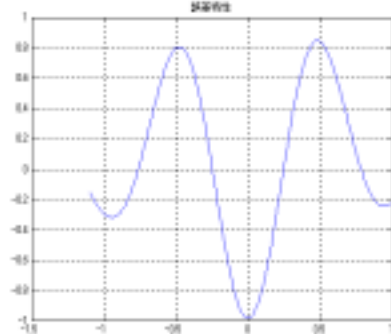


図4.28 誤差特性 (刻み幅0.20)



## 2. syuku=0.36、heiko=0.464のとき

1. と条件は、全く同一とした。2000回学習後の入出力特性および誤差特性を図4.29から図4.56に示す。誤差の絶対値の最大値が0.25以下の刻み幅は、0.04、0.06、0.1であった。とくに、0.06と0.1のときの特性が誤差の生じやすい領域を含んで、ほとんど変化しないのが興味深い。刻み幅が0.02、0.03のときは、syuku=0.20と同様、誤差が大きく学習が進んでいないという結果となった。さらに、刻み幅が0.12以上の場合はほとんど学習が進んでいないという状況となった。

図4.29 入出力特性 (刻み幅0.02)

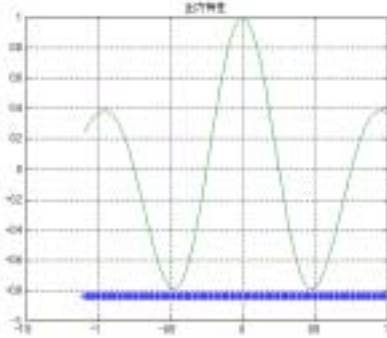


図4.30 誤差特性 (刻み幅0.02)

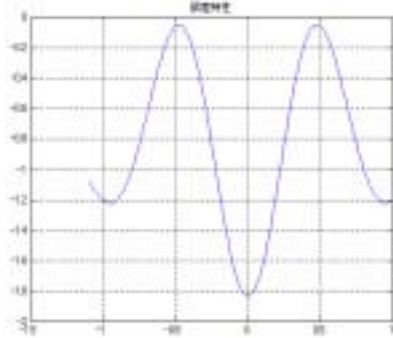


図4.31 入出力特性 (刻み幅0.03)

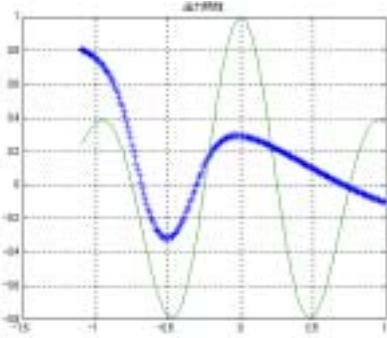


図4.32 誤差特性 (刻み幅0.03)

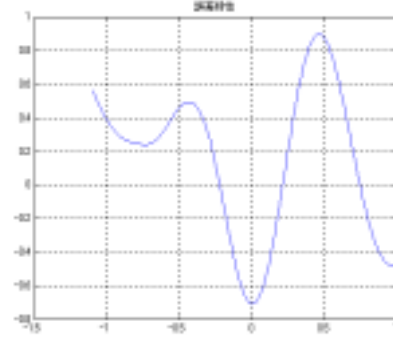


図4.33 入出力特性 (刻み幅0.04)

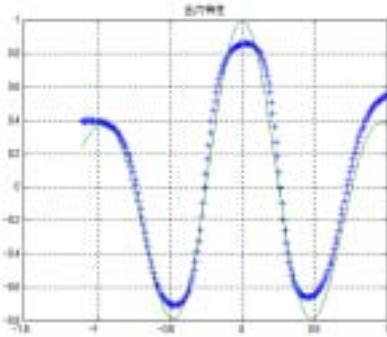


図4.34 誤差特性 (刻み幅0.04)

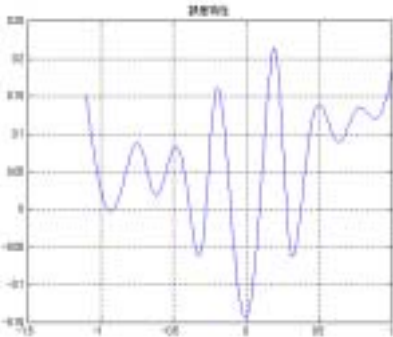


図4.35 入出力特性 (刻み幅0.05)

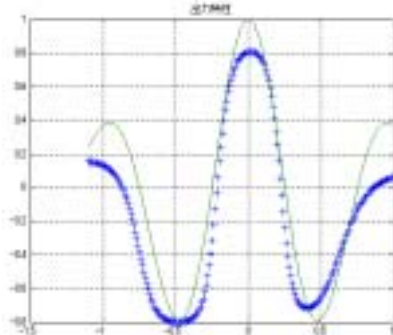


図4.36 誤差特性 (刻み幅0.05)

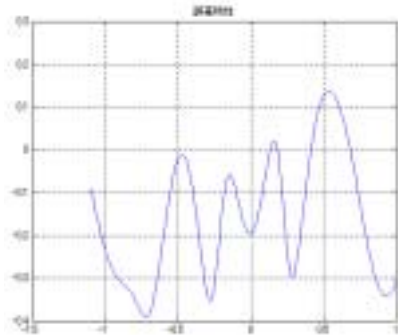


図4.37 入出力特性 (刻み幅0.06)

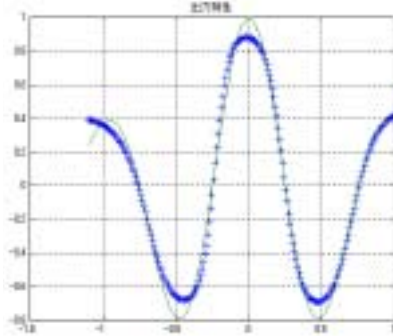


図4.38 誤差特性 (刻み幅0.06)

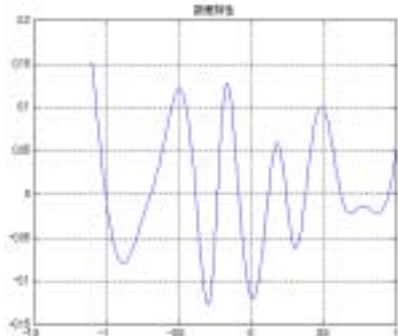


図4.39 入出力特性 (刻み幅0.07)

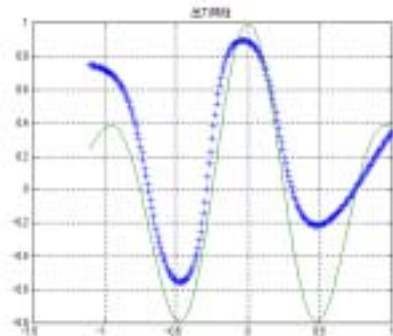


図4.40 誤差特性 (刻み幅0.07)

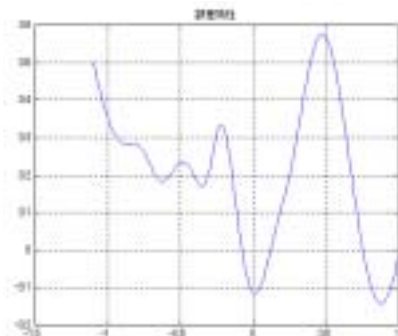


図4.41 入出力特性 (刻み幅0.08)

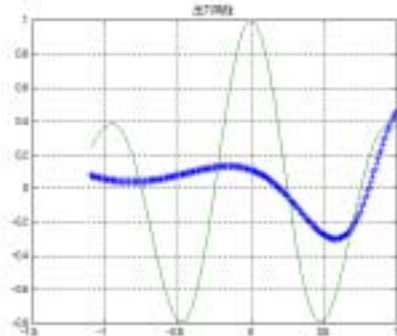


図4.42 誤差特性 (刻み幅0.08)

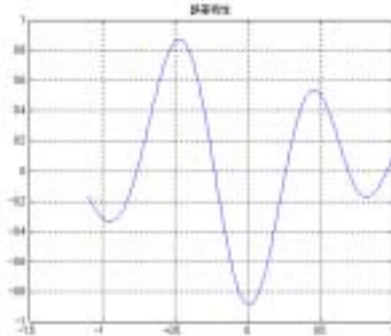


図4.43 入出力特性 (刻み幅0.10)

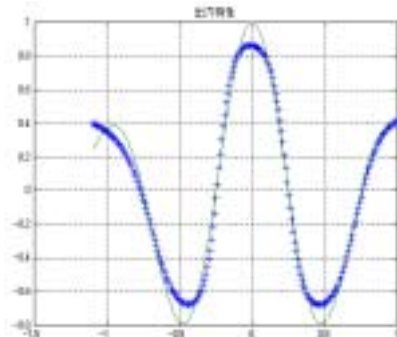


図4.44 誤差特性 (刻み幅0.10)

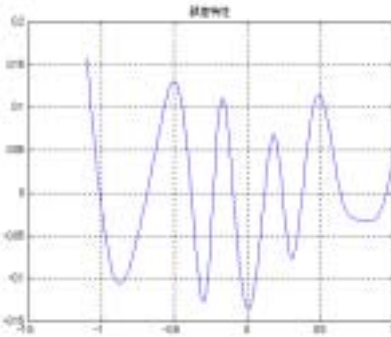


図4.45 入出力特性 (刻み幅0.12)

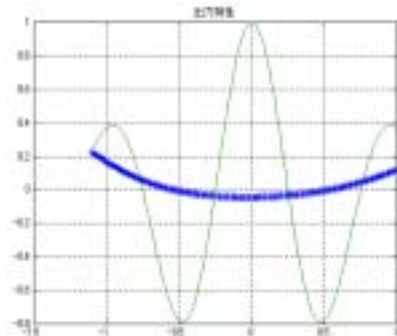


図4.46 誤差特性 (刻み幅0.12)

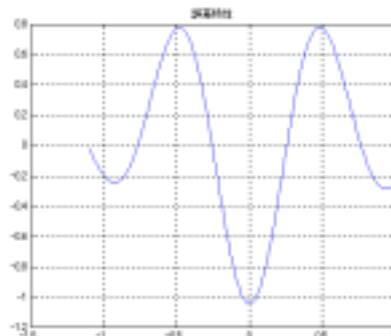


図4.47 入出力特性 (刻み幅0.14)

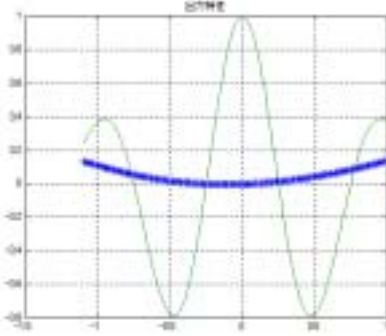


図4.48 誤差特性 (刻み幅0.14)

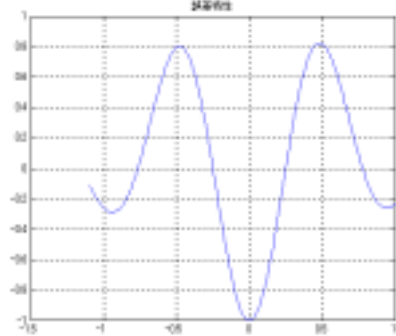


図4.49 入出力特性 (刻み幅0.15)

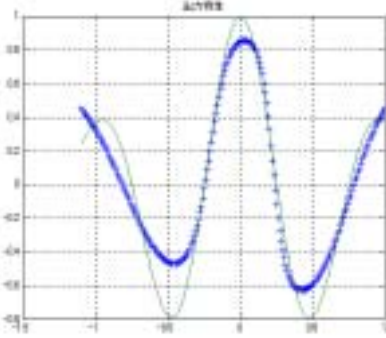


図4.50 誤差特性 (刻み幅0.15)

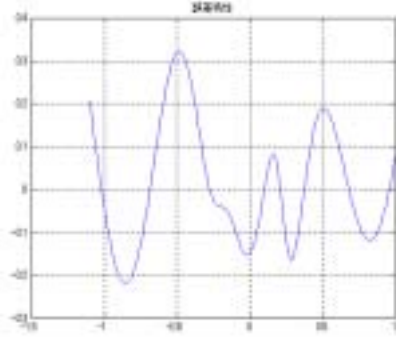


図4.51 入出力特性 (刻み幅0.16)

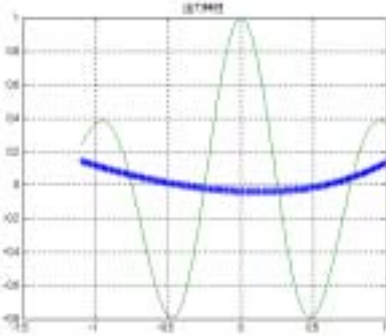


図4.52 誤差特性 (刻み幅0.16)

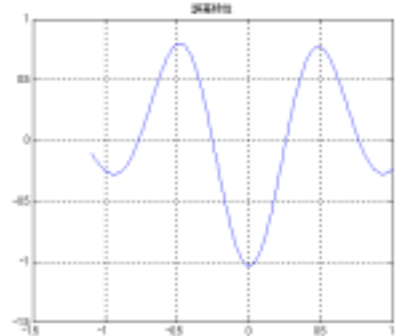


図4.53 入出力特性 (刻み幅0.18)

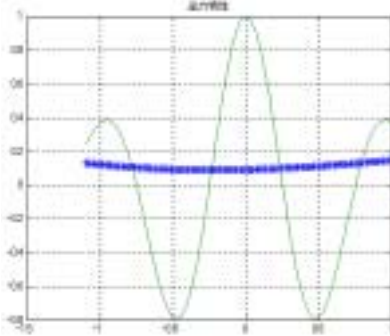


図4.54 誤差特性 (刻み幅0.18)

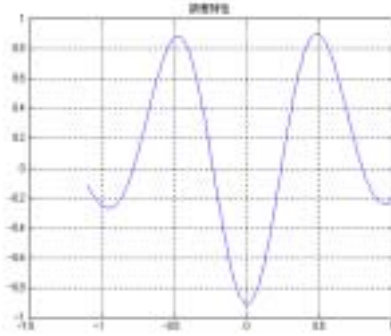


図4.55 入出力特性 (刻み幅0.20)

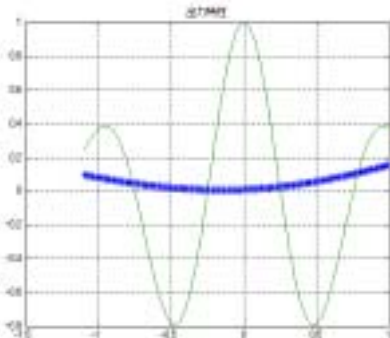
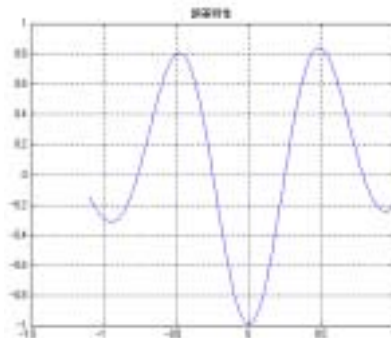


図4.56 誤差特性 (刻み幅0.20)



### 3. syuku=0.54、heiko=0.45のとき

ここでも 1. と条件は、全く同一とした。2000回学習後の入出力特性および誤差特性を図4.57から図4.84に示す。誤差の絶対値の最大値が0.25以下の刻み幅は、0.04、0.06、0.1であった。0.06と0.1のとき誤差の最大絶対値誤差が0.13程度であった。また、誤差の生じやすい領域を含んで、syuku=0.36と同様な結果となったことが興味深い。刻み幅が0.02、0.03のときは、syuku=0.20およびsyuku=0.36と同様、誤差が大きく学習が進んでいないという結果となった。また、syuku=0.15のときは、最大絶対値誤差が、0.32程度となり比較的学習が進んでいた。



図4.57 入出力特性 (刻み幅0.02)

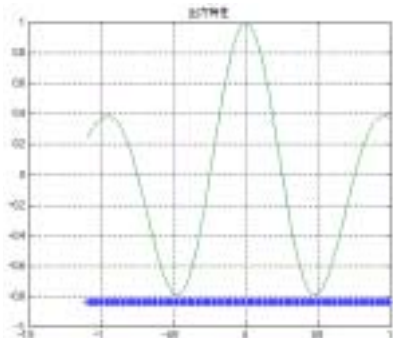


図4.58 誤差特性 (刻み幅0.02)

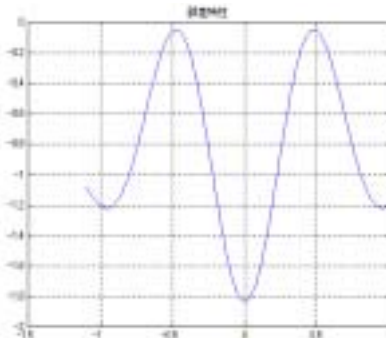


図4.59 入出力特性 (刻み幅0.03)

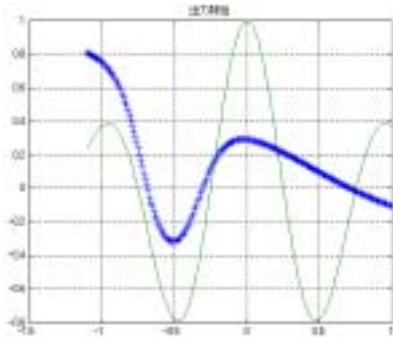


図4.60 誤差特性 (刻み幅0.03)

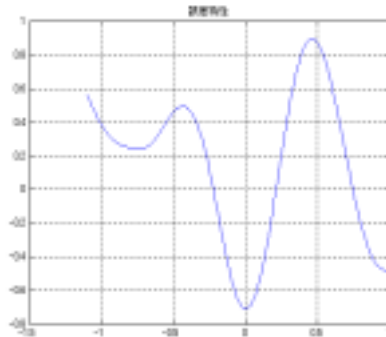


図4.61 入出力特性 (刻み幅0.04)

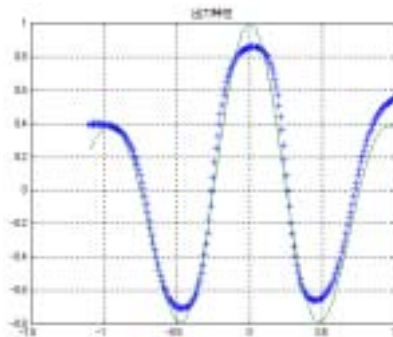


図4.62 誤差特性 (刻み幅0.04)

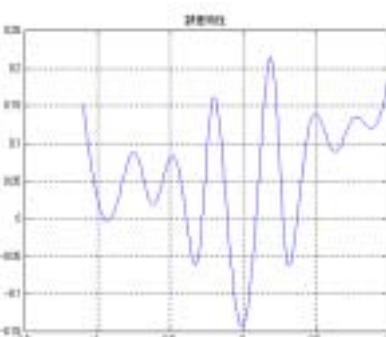


図4.63 入出力特性 (刻み幅0.05)

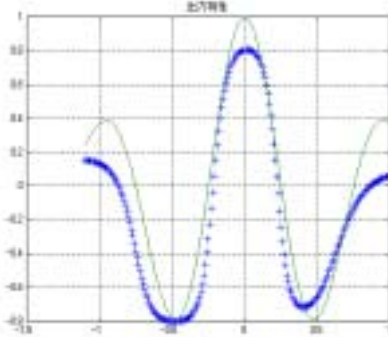


図4.64 誤差特性 (刻み幅0.05)

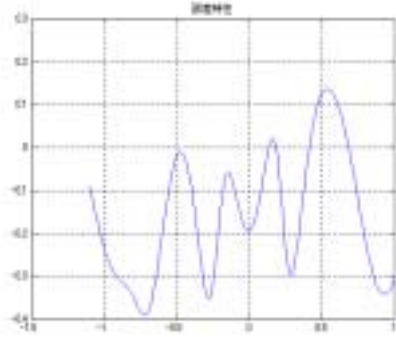


図4.65 入出力特性 (刻み幅0.06)

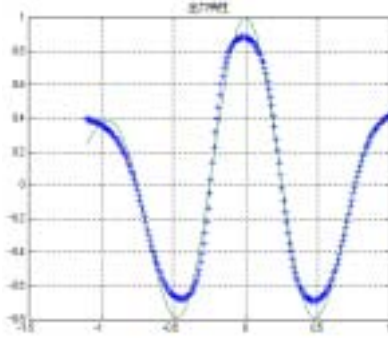


図4.66 誤差特性 (刻み幅0.06)

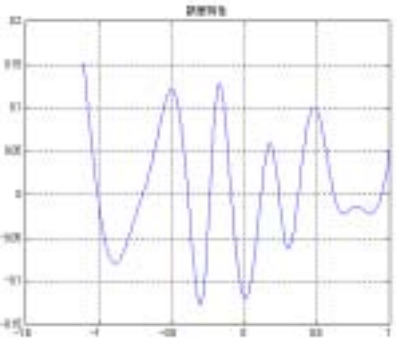


図4.67 入出力特性 (刻み幅0.07)

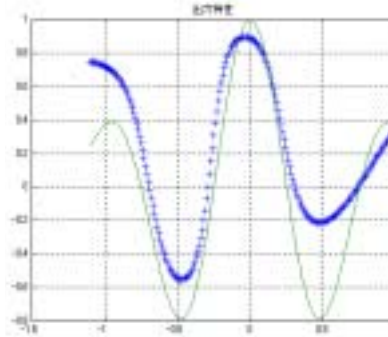


図4.68 誤差特性 (刻み幅0.07)

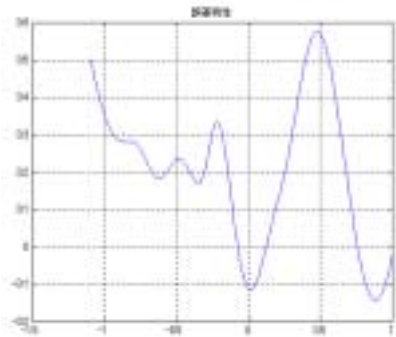


図4.69 入出力特性 (刻み幅0.08)

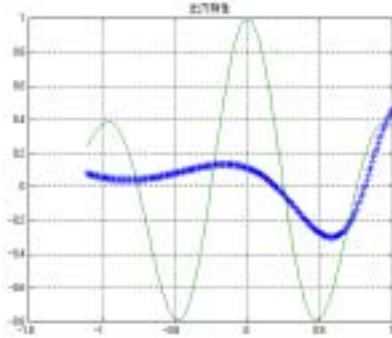


図4.70 誤差特性 (刻み幅0.08)

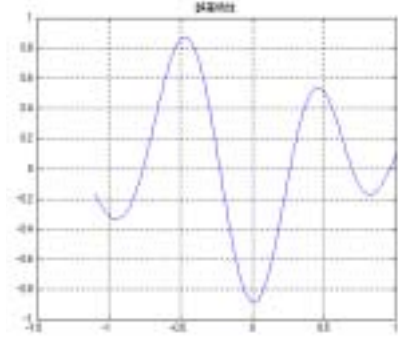


図4.71 入出力特性 (刻み幅0.10)

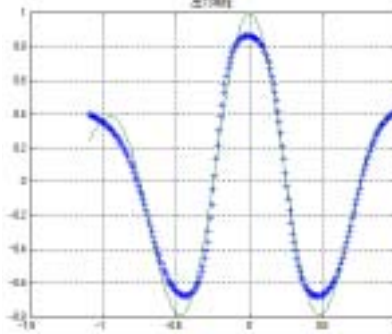


図4.72 誤差特性 (刻み幅0.10)

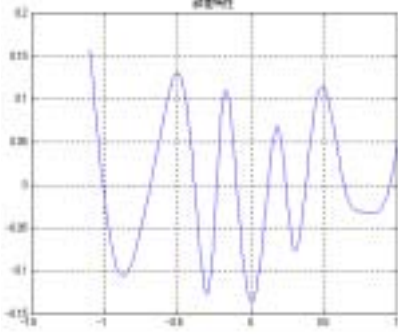


図4.73 入出力特性 (刻み幅0.12)

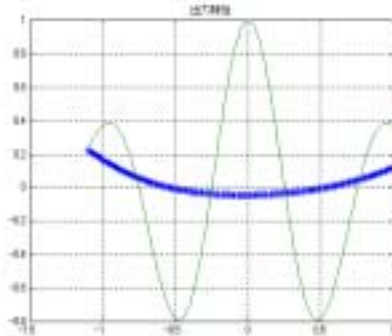


図4.74 誤差特性 (刻み幅0.12)

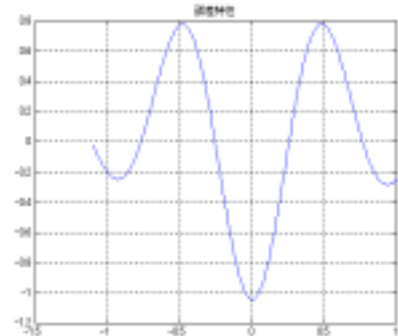


図4.75 入出力特性 (刻み幅0.14)

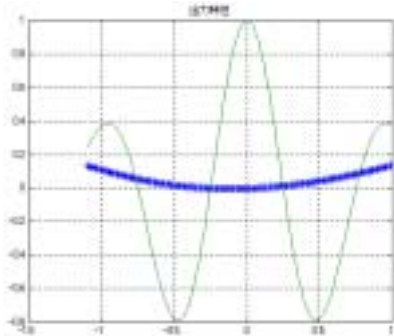


図4.76 誤差特性 (刻み幅0.14)

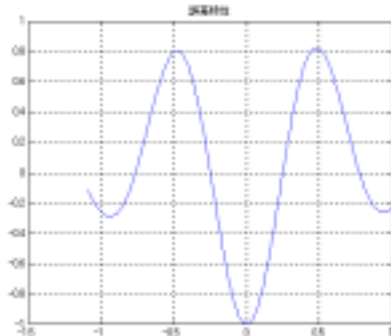


図4.77 入出力特性 (刻み幅0.15)

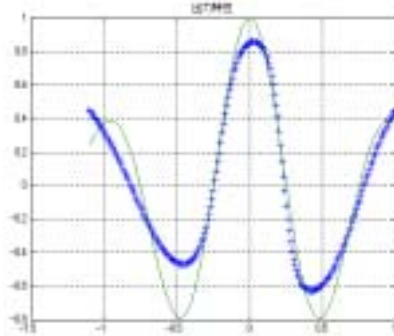


図4.78 誤差特性 (刻み幅0.15)

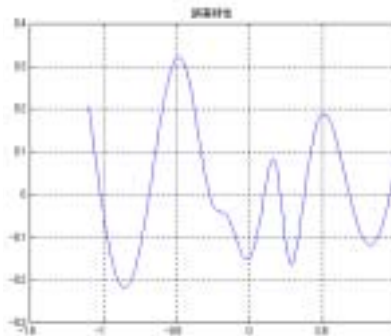


図4.79 入出力特性 (刻み幅0.16)

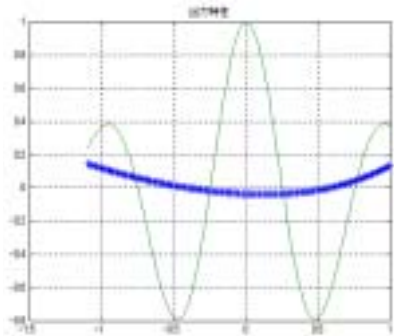


図4.80 誤差特性 (刻み幅0.16)

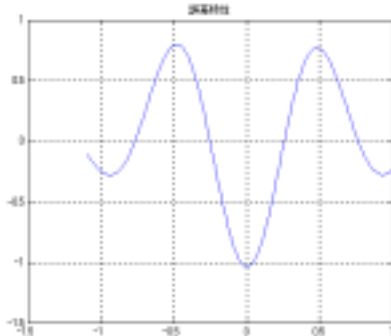


図4.81 入出力特性 (刻み幅0.18)

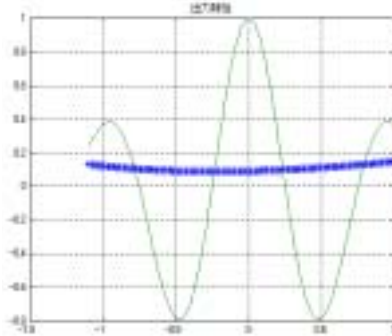


図4.82 誤差特性 (刻み幅0.18)

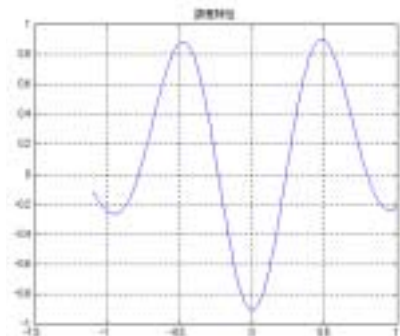


図4.83 入出力特性 (刻み幅0.20)

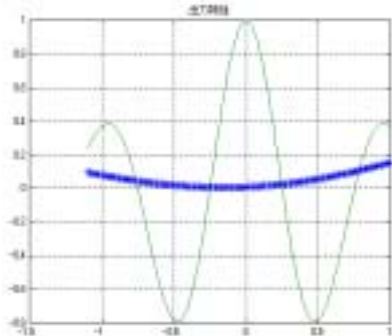
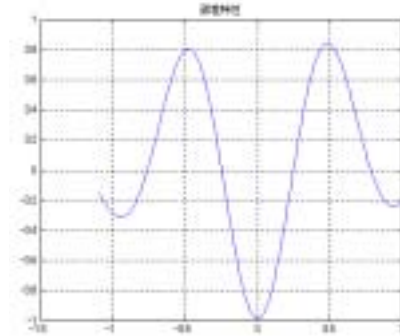


図4.84 誤差特性 (刻み幅0.20)



## 5. 考察

ニューラルネットワーク関数学習では、規格化という操作が行われる。計算機実験 1. 2. 3. を見て共通して言えることは縮小率があまり小さすぎでは学習速度が遅くなるのではないかと考えられる。本稿では、 $syuku=0.20$ 、 $0.36$ 、 $0.54$  に対して  $syuku=0.36$ 、 $0.54$  の場合2000回の学習で学習が進み、誤差特性で良好な結果を持つ誤差特性となった。さらに、目標関数の刻み幅が、 $0.04$ 、 $0.06$ 、 $0.1$  のとき、非常に良好な結果となり、それより刻み幅を大きくしても小さくしても良好な結果とはならなかった。すなわち、一般的によく言われていることは、誤差特性を良好にするため

には、刻み幅を小さくすればよいということである。しかし、小さすぎても良好な結果とはならないことが、本実験から明らかとなった。総合的に考えると、syuku の値を大きめにとり、刻み幅もあまり小さくとらず、0.04、0.06、0.1程度とするとよいという結果となった。今後の課題は、規格化、刻み幅を決めた後で、学習回数を減少させる方法を考察したい。

## 6. おわりに

本稿では、ニューラルネットワークに対する規格化の検討を行った。その結果、規格化するための注意点が得られた。すなわち、縮小率は、あまり小さくすると、学習速度が遅く、良好な結果とはならない。さらに、目標関数の刻み幅も小さすぎても、大きすぎても良好な学習が行われないという結果となった。

(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。
- (3) 田口功・須貝康雄「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」『敬愛大学国際研究』第10号、2002年、63-93ページ。