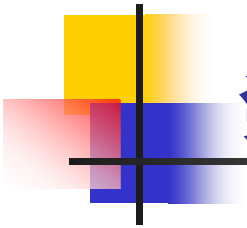
ニューラルネットによる 逐次的関数近似法の提案

沼田研究室
4401073
福井　健一

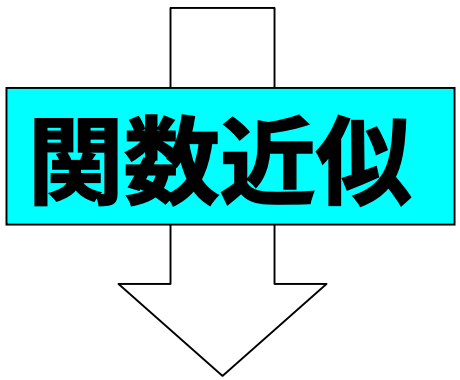


発表構成

1. はじめに
2. ニューラルネットワーク
3. 逐次学習手法
4. 実験
5. 結果と考察
6. 今後の課題
7. 参考文献

1．はじめに

世の中の事象の中には，ある関数にしたがった動き方をしているものがある



未来の情報を予測し各種の利益をもたらすことが考えられる．

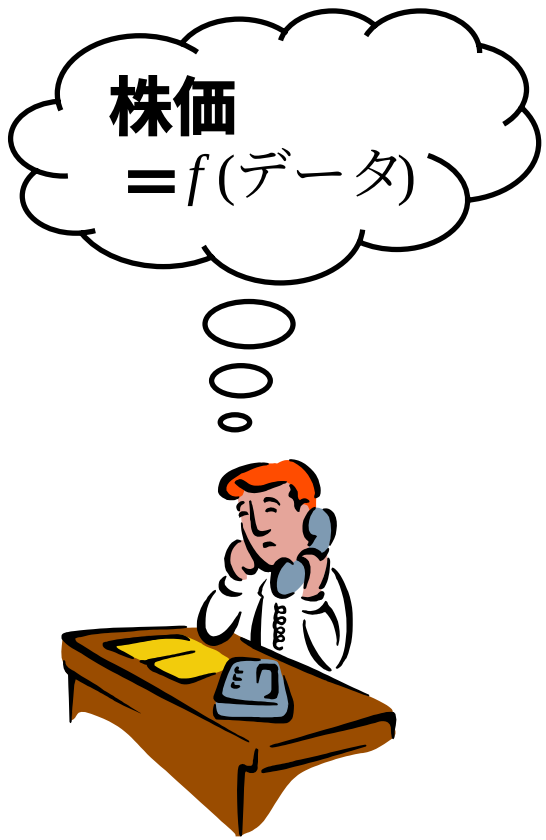
例 「株価予測」[5]

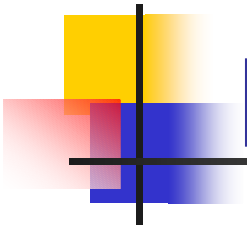
株価は様々な条件で変動している

株価をデータによる関数であるとみなす

関数近似

予測された結果から売買の判断を行う





関数近似

関数近似をする際に大別して2つのアプローチが考えられる

Ⅰ．解析的手法

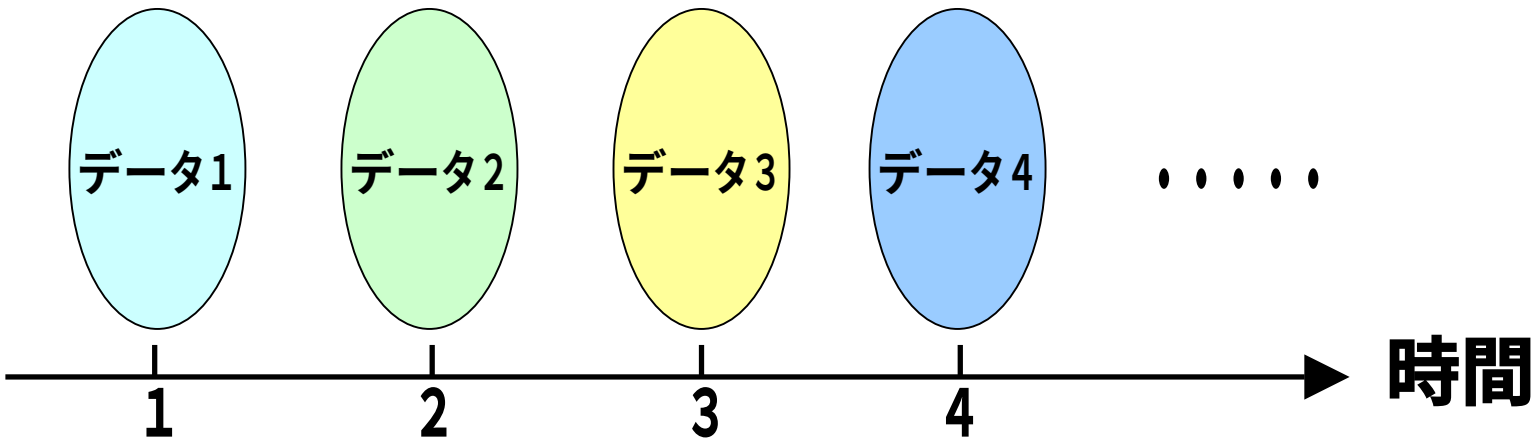
関数形が予測できるデータに用いられる

Ⅱ．機械学習法

関数形が予測できないデータに用いられる

機械学習法の問題点

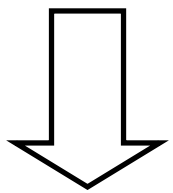
逐次的にデータが得られた場合



計算量が膨大に！ ➡ **逐次的な学習**

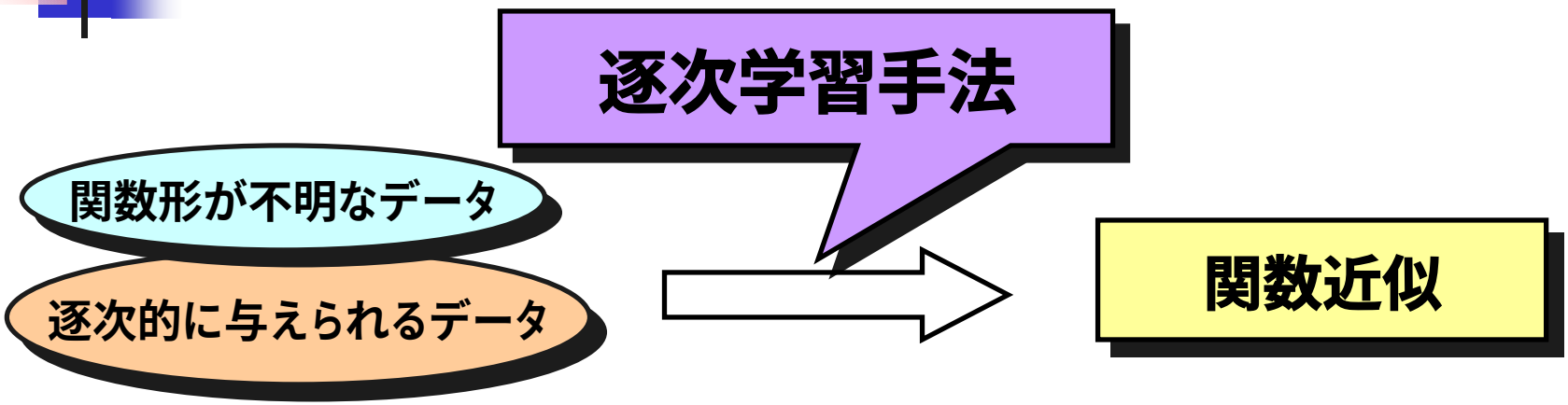
先行研究[3]

データが与えられるごとに領域の区分を，少しずつ変形する



区分された各小領域において線形近似することで
関数を近似する

本研究の目的

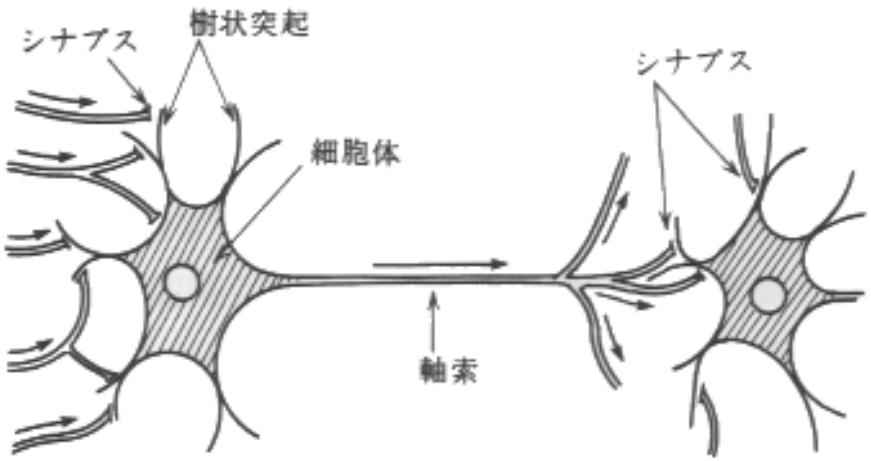


従来研究[3]と違ったアプローチで構築する

ニューラルネットワークの構造を用いて構築

2．ニューラルネットワーク (NN)

人間の脳の構造を模して作った情報処理機構



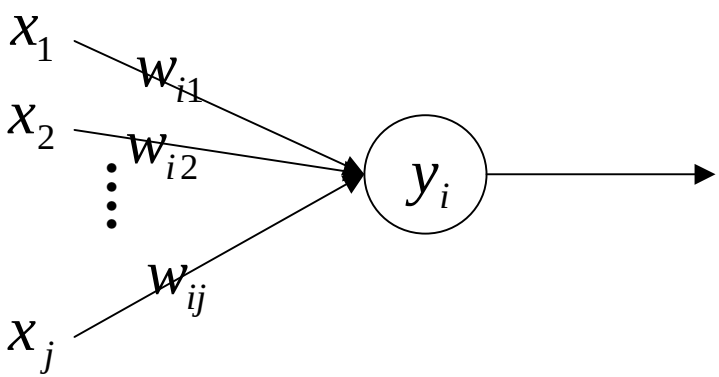
ニューロンの概略

脳＝ニューロンの集合

この構造をモデル化！

人工ニューロン

生物の神経系であるニューロンをモデル化したもの



$$y_i = h[\sum_j w_{ij} x_j - \theta_i]$$

x_j ：ニューロンj からの出力

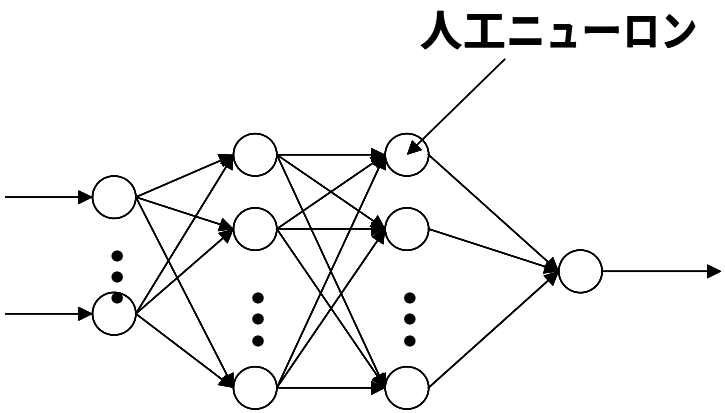
w_{ij} ：ニューロンij　間の結合強度

θ_i ：ニューロンi の閾値

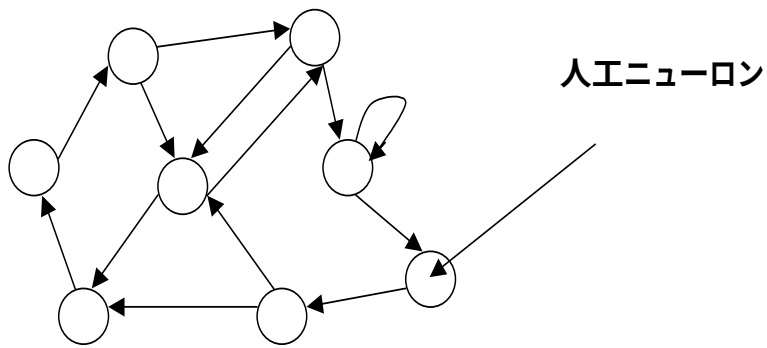
$h[...]$ ：閾値関数

ニューラルネットのモデル

人工ニューロンを結合 ＝ **ニューラルネットワーク**



階層型ニューラルネットモデル



相互型ニューラルネットモデル

3．逐次学習手法

逐次的に与えられたデータに対してシステムを逐次改善し
関数近似結果を出力する

従来研究[3]：領域を動的に区分し，区分された小領域に
おいて線形近似することで関数近似を行う

本研究　：領域内をNNの構造を用いて学習し関数近
似を行う

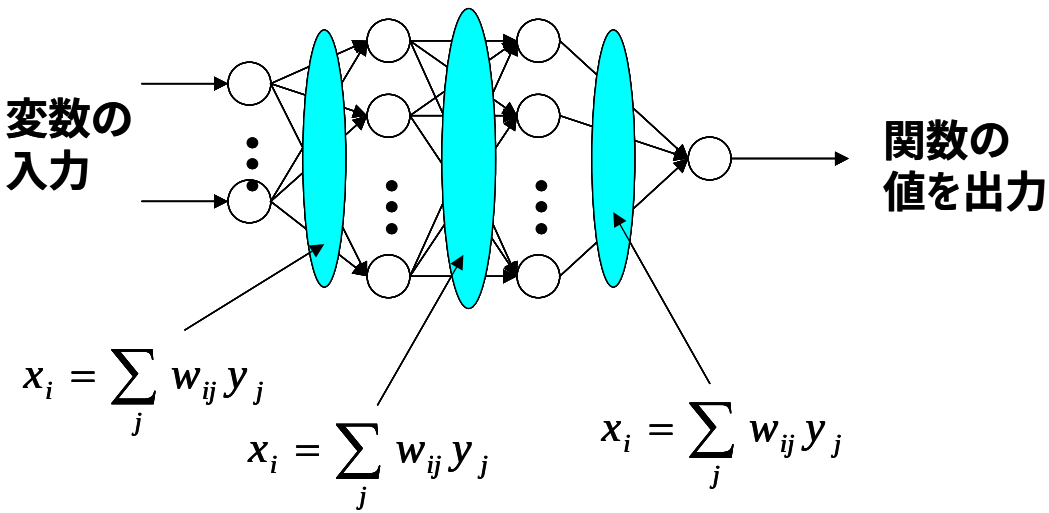
閾値関数を用いないネットワーク

閾値関数を用いるネットワーク　　の3つの方式を比較した

閾値ありの区分学習

Ⅰ：閾値関数を用いないネットワーク

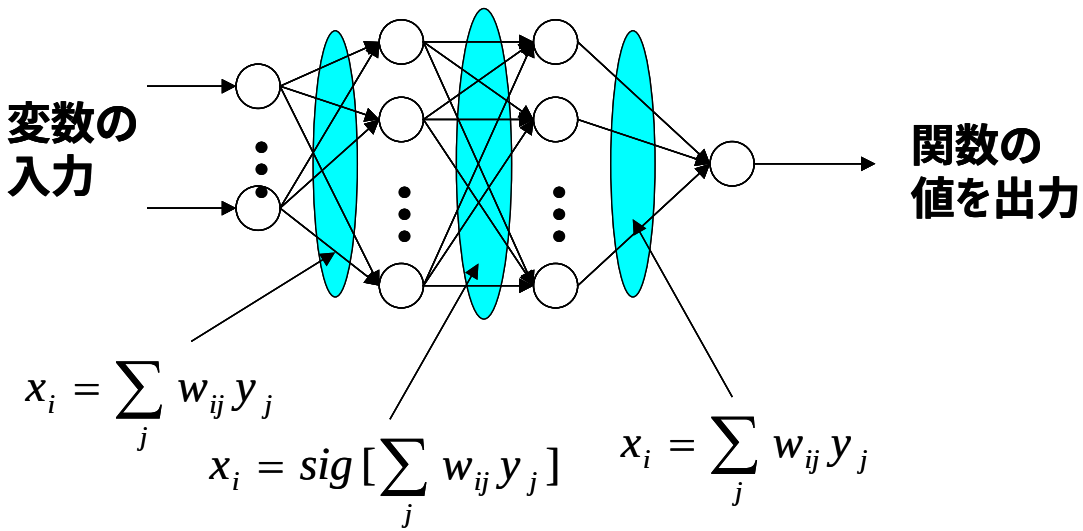
入力層→中間層→中間層→出力層の間が全て入力と重みとの線形結合のみによって表されるネットワーク



ニューロンの発火という概念を無視して構成している

II：閾値関数を用いるネットワーク

中間層→中間層において閾値関数としてシグモイド関数を用い，他のニューロンからの出力は全て入力と結合強度との線形結合によってあらわす



III：閾値ありの区分学習

与えられたデータの定義域から固定的に領域を分割する

区分された小領域について，それぞれ学習をおこなう

問題点

分割領域が多い

→ 近似は容易

学習量は膨大に！

4．実験

実験で近似する関数

$$1:\ f(x_1,x_2)=\frac{1}{9}\left(x_1+x_2\right)$$

$$2:\ f(x_1,x_2)=\frac{2}{27}\left({x_1}^2+x_1\times x_2+{x_2}^2\right)$$

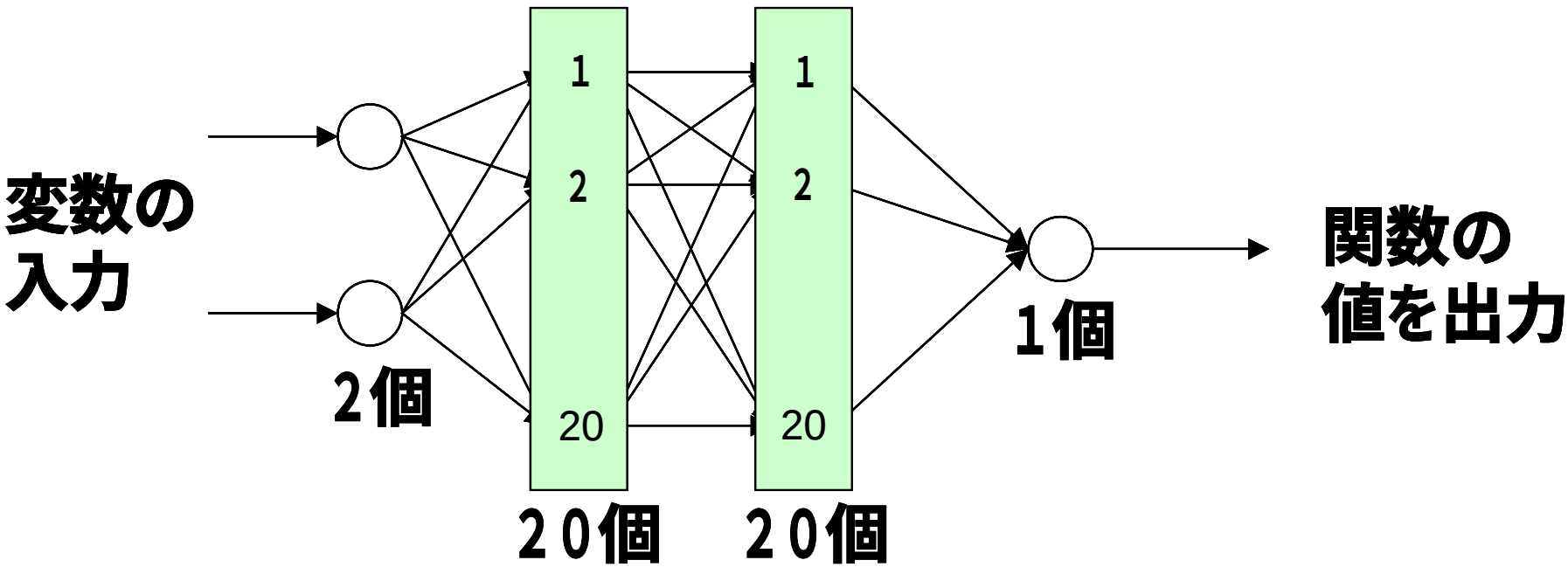
$$3:\ f(x_1,x_2)=\frac{1}{18}\left({x_1}^3+{x_1}^2x_2+x_1{x_2}^2+{x_2}^3\right)$$

$$4:\ f(x_1,x_2)=\frac{\tanh(9x_2-9x_1)+1}{9}$$ [3]

関数1～4の定義域： $\left[0\ ,\ \frac{2}{9}\right]$

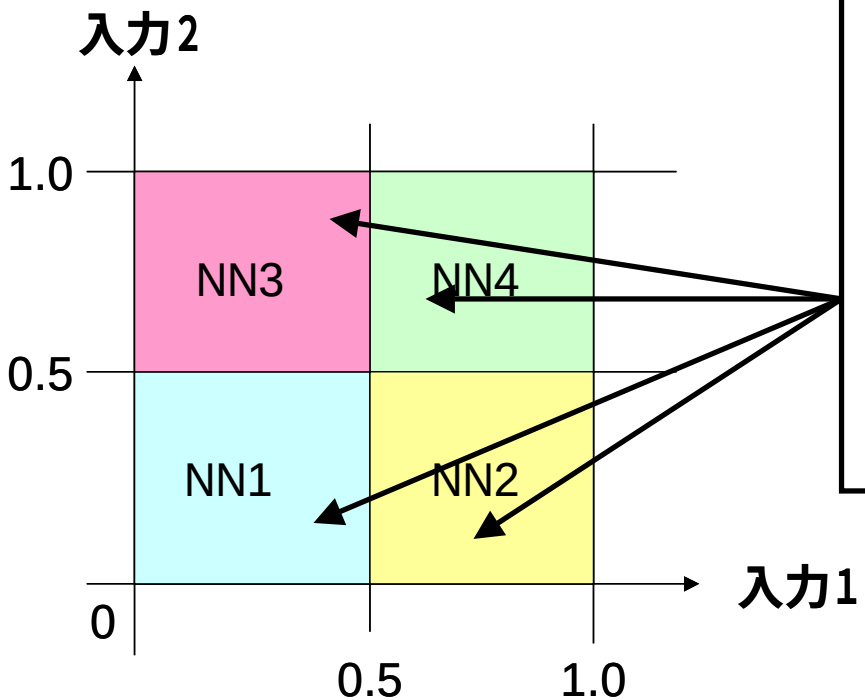
設定　ネットワークのニューロン個数

ネットワーク：入力層2個，中間層20個，出力層1個

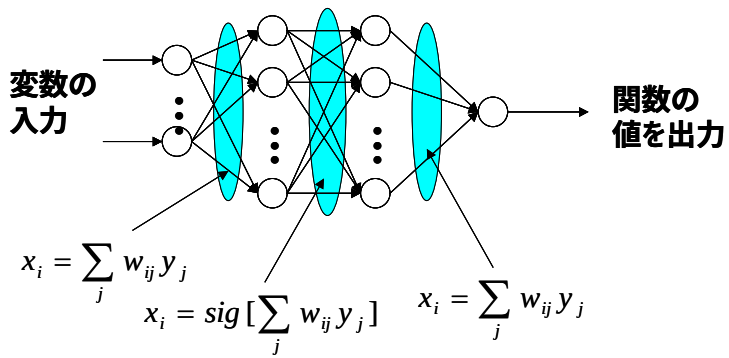


設定　手法III：区分学習

均等に4分割

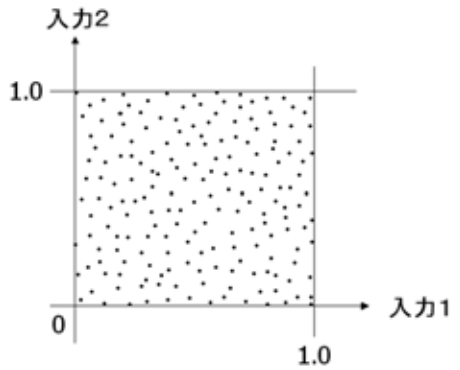


手法II：閾値関数を用いるネットワーク



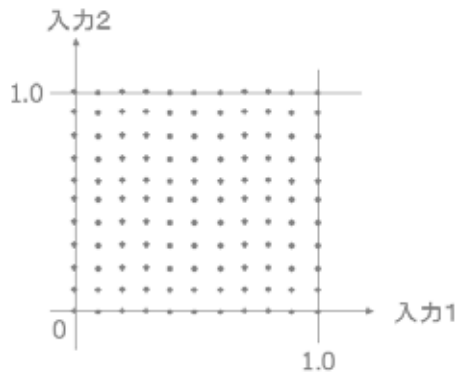
実験方法

Ⅰ：各関数の訓練データ $y = f_i(x_1, x_2)$ は単位正方形 $S = \{x = (x_1, x_2) | x_1, x_2 \in [0,1]\}$ からランダムに選び生成する



Ⅱ：規定回数の訓練データを学習する

Ⅲ：S 上の10×10のサンプル点での関数値 $y = f_i(x_1, x_2)$ と実装したシステムによる推定値との差(誤差)をとる

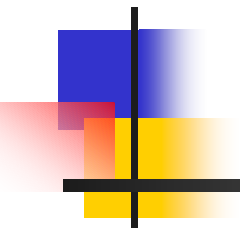


Ⅳ：システムを評価する指標として，平均二乗誤差を計算し，考察をおこなう

（平均二乗誤差：各サンプル点における二乗誤差の平均値）

実行画面

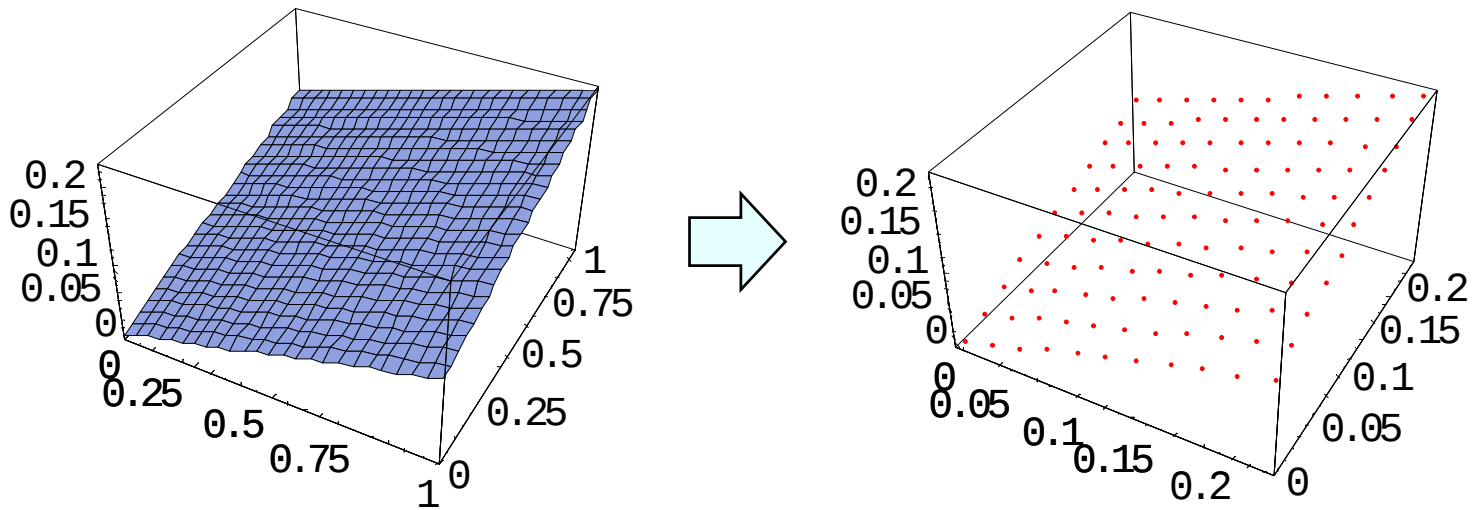

Form1
実験開始
実行時間
ステップ終了時の関数値:
実際値
ステップ終了時の予測値:
予測値
Panel1
ステップ数増加
リセット
ステップ数
総誤差平均
終了



5．結果と考察

数値実験結果：関数1　手法III

$$1:\ f(x_1, x_2) = \frac{1}{9}(x_1 + x_2)$$



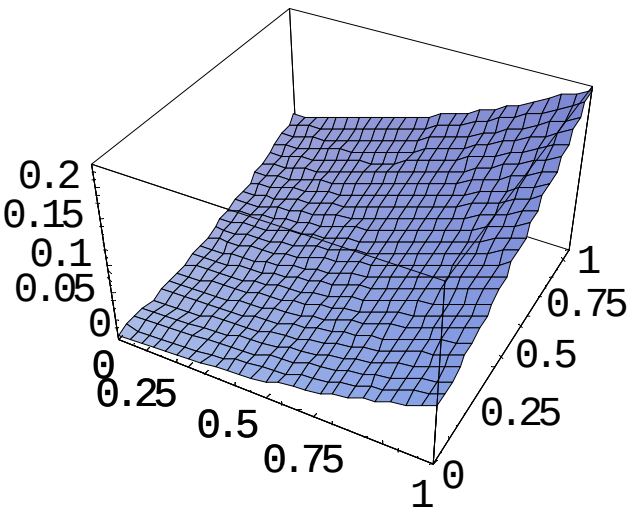
関数1

学習後の推定値

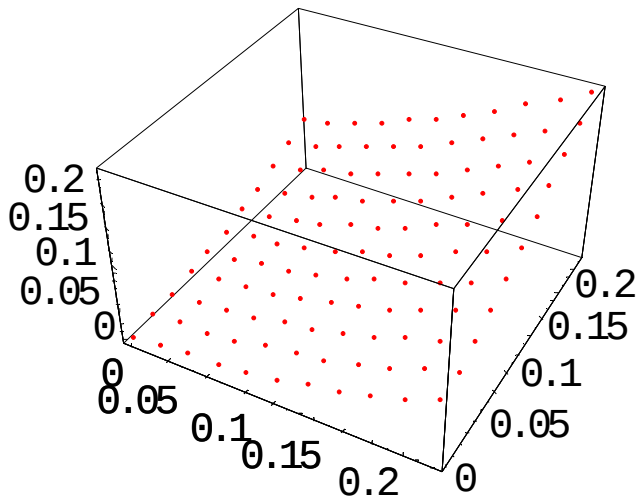
（学習回数：1.0×10^8）

数値実験結果：関数2　手法III

$$\mathbf{2}:\ f(x_1, x_2) = \frac{2}{27}\left({x_1}^2 + x_1 \times x_2 + {x_2}^2\right)$$



関数2

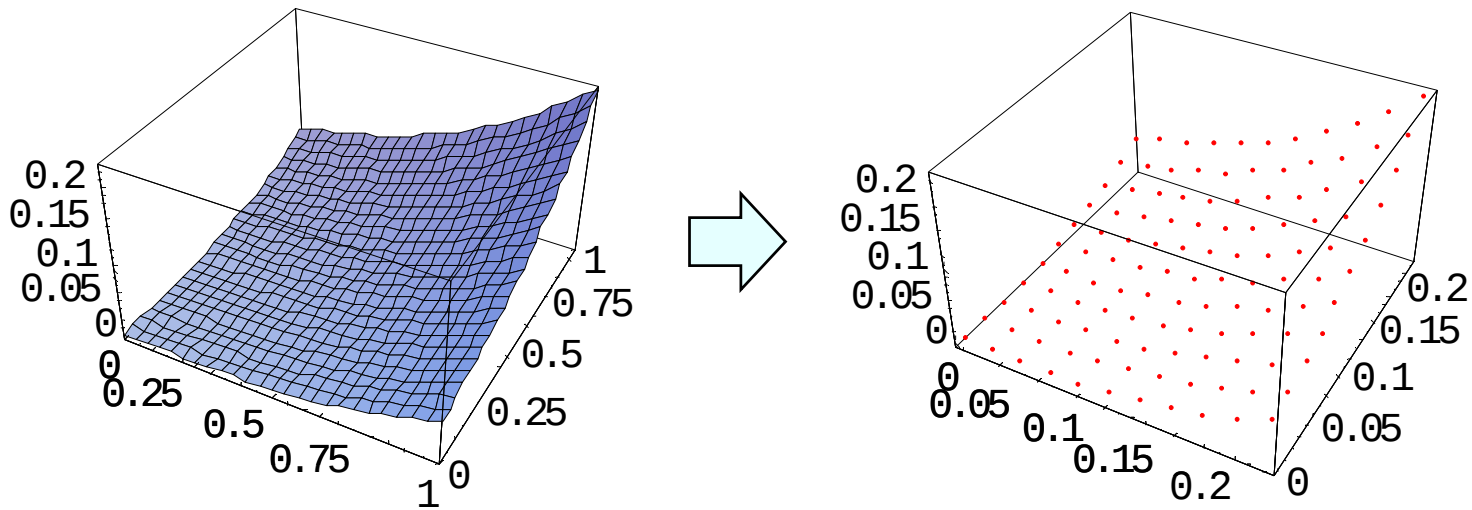


学習後の推定値

(学習回数：1.0×10^8)

数値実験結果：関数3　手法III

$$\mathbf{3}:\ f(x_1, x_2) = \frac{1}{18}\left({x_1}^3 + {x_1}^2 x_2 + x_1 {x_2}^2 + {x_2}^3\right)$$



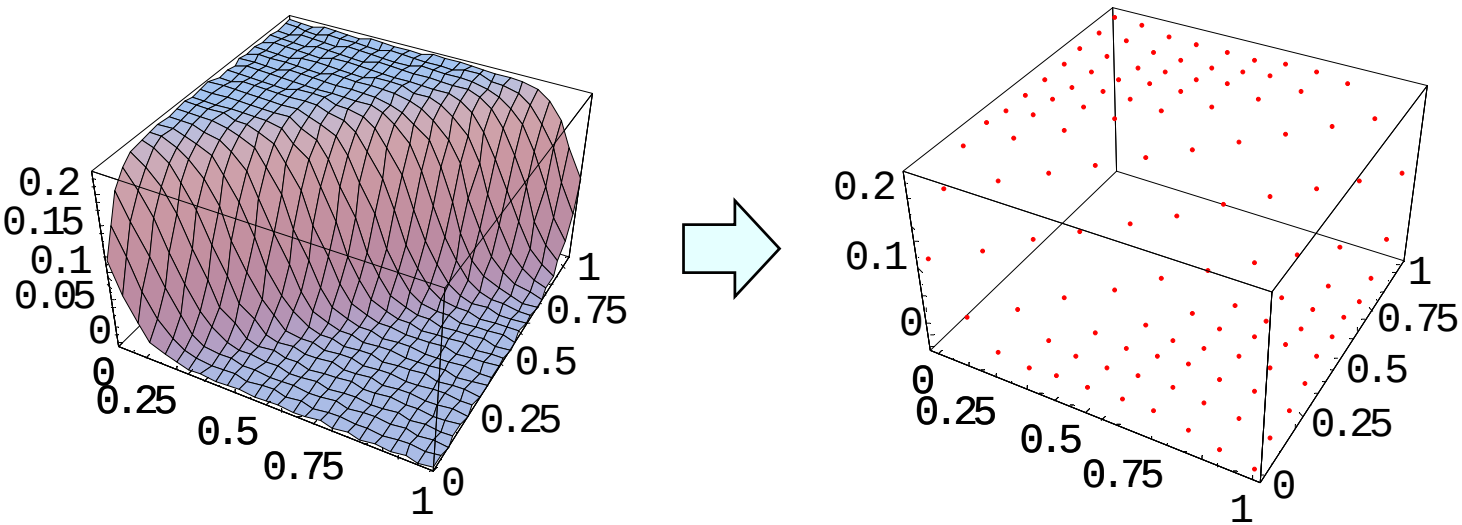
関数3

学習後の推定値

(学習回数：1.0×10^8)

数値実験結果：関数4　手法III

$$\mathbf{4}:\ f(x_1, x_2) = \frac{\tanh(9x_2 - 9x_1) + 1}{9}$$



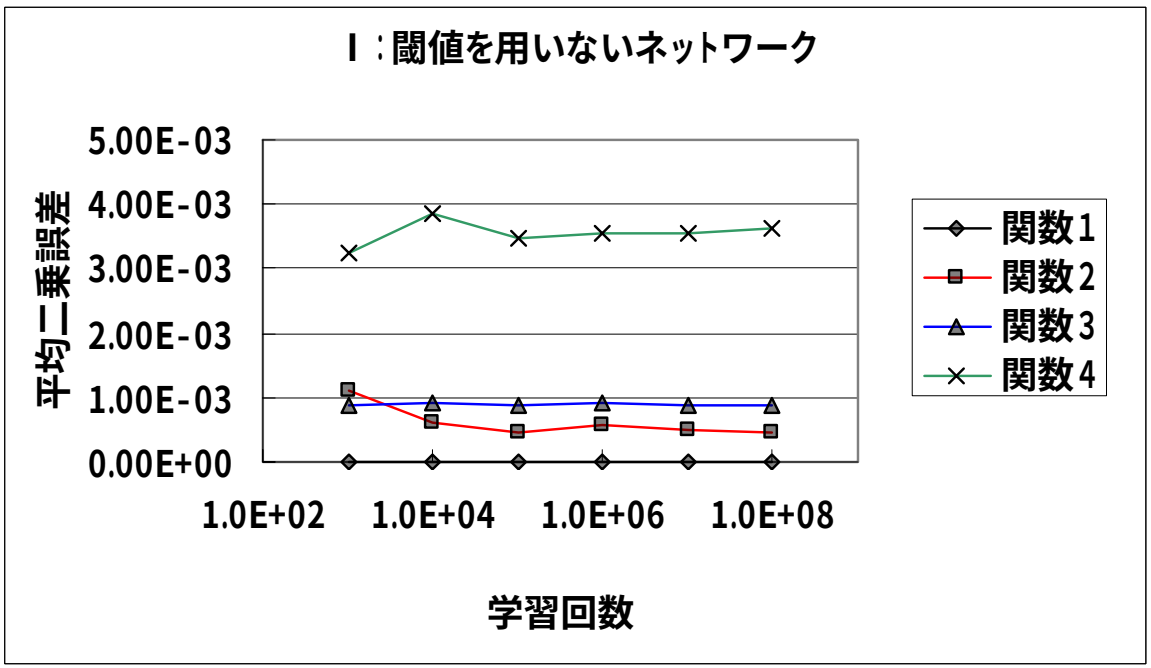
関数4

学習後の推定値

(学習回数：1.0×10^8)

結果　Ⅰ：閾値関数を用いないネットワーク

	1000	10000	100000	1000000	10000000	100000000	学習回数
関数1	4.18E-19	5.32E-33	4.85E-32	5.78E-32	3.37E-32	3.58E-32	
関数2	0.0010987	0.0006103	0.0004624	0.0005651	0.0005131	0.0004671	
関数3	0.0008924	0.0009094	0.0008616	0.0009071	0.0008647	0.0008688	
関数4	0.0052545	0.0038367	0.0034806	0.003556	0.0035389	0.0036134	

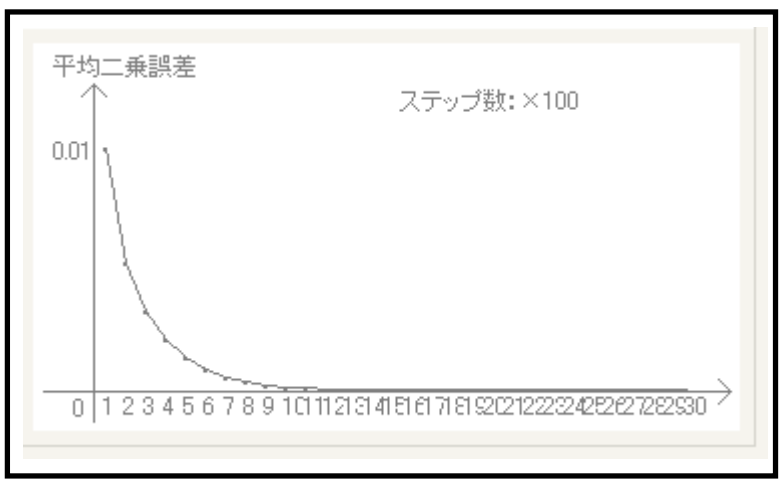


考察　Ⅰ：閾値関数を用いないネットワーク

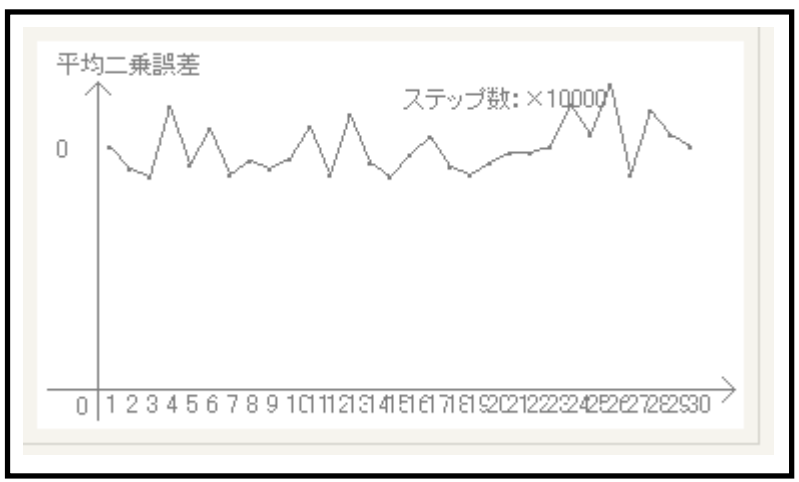
線形関数：学習がおこなえている

非線形関数：学習が行えていない

⇨ **Ⅰの手法では非線形関数の近似は不可能である**



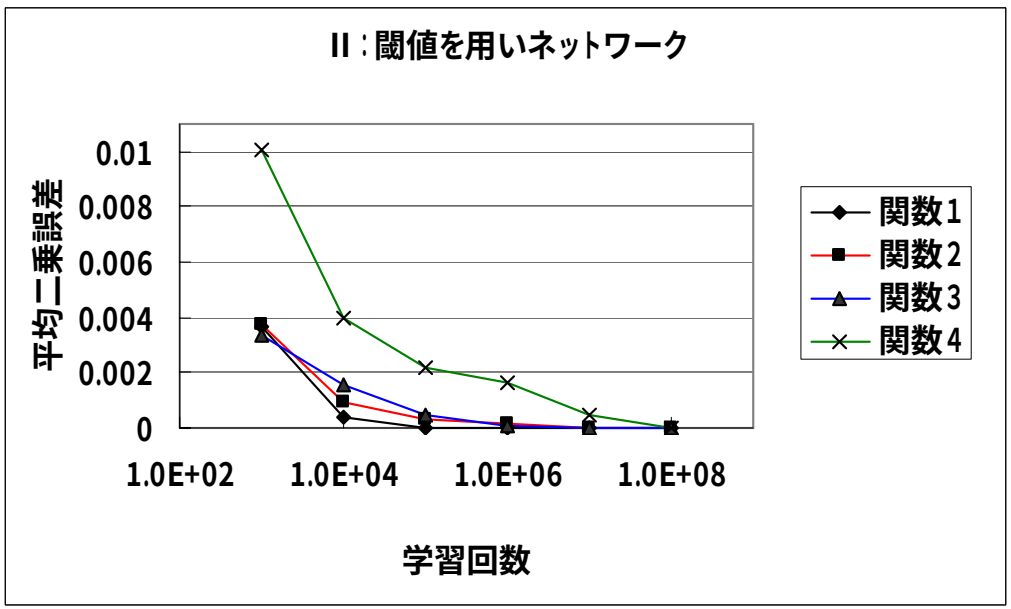
線形関数（関数1）



非線形関数（関数2）

結果　II：閾値関数を用いるネットワーク

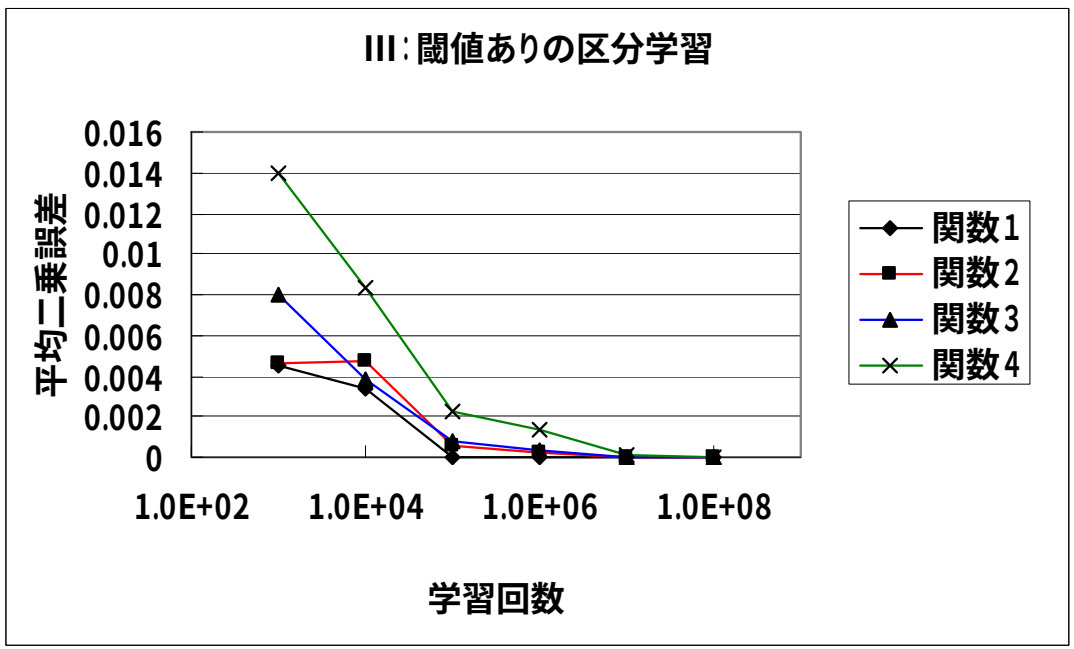
	1000	10000	100000	1000000	10000000	100000000	学習回数
関数1	0.0036532	0.0004185	1.737E-05	6.934E-06	2.386E-06	3.452E-08	
関数2	0.0037294	0.0008985	0.0003496	0.000187	2.273E-05	7.823E-07	
関数3	0.0033801	0.0015517	0.0005068	7.577E-05	2.928E-05	3.608E-06	
関数4	0.0100654	0.003975	0.0021775	0.0016465	0.0004884	4.104E-05	



従来研究[3]
関数4　学習回数：1.0×10^{8}
平均二乗誤差:4.0E-6

結果　III：閾値ありの区分学習

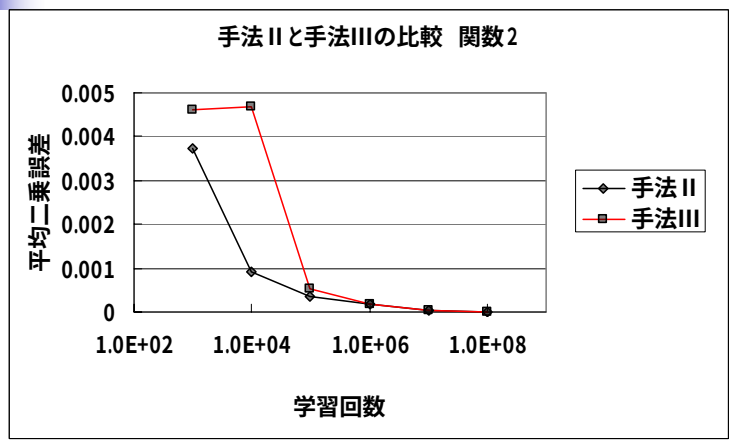
	1000	10000	100000	1000000	10000000	100000000	学習回数
関数1	0.0044686	0.0034036	3.64E-05	1.83E-05	2.18E-06	2.97E-08	
関数2	0.0046237	0.0046988	0.0005222	0.0001798	3.149E-05	3.292E-07	
関数3	0.0080192	0.0037809	0.0008402	0.0003198	9.376E-06	9.18E-07	
関数4	0.0139938	0.0083014	0.0022112	0.0013638	9.582E-05	2.928E-05	

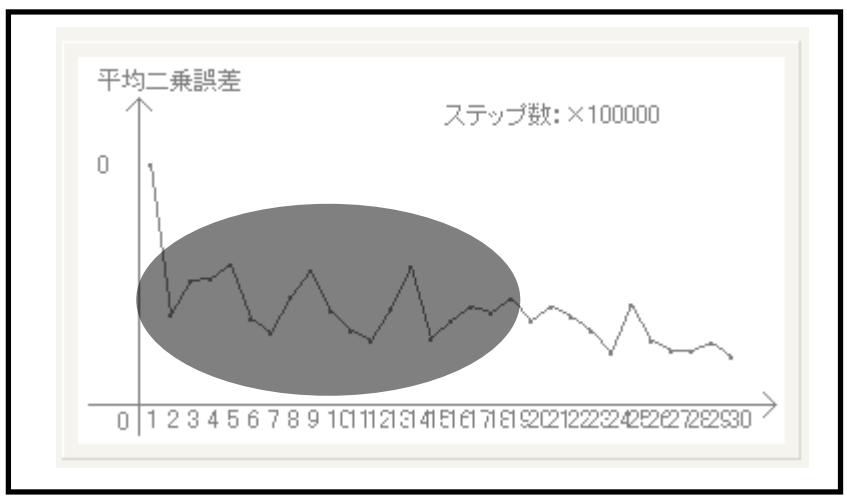


100000000
3.452E-08
7.823E-07
3.608E-06
4.104E-05

手法II

考察　III：閾値ありの区分学習





誤差の減少が遅い

誤差の値が振動している

区分した領域の数だけ学習回数は分散されている

結論

・先行研究[3]と比べると近似性能は劣っている

・手法II，手法III：逐次学習手法として用いることが可能

・手法IIIの関数近似の精度

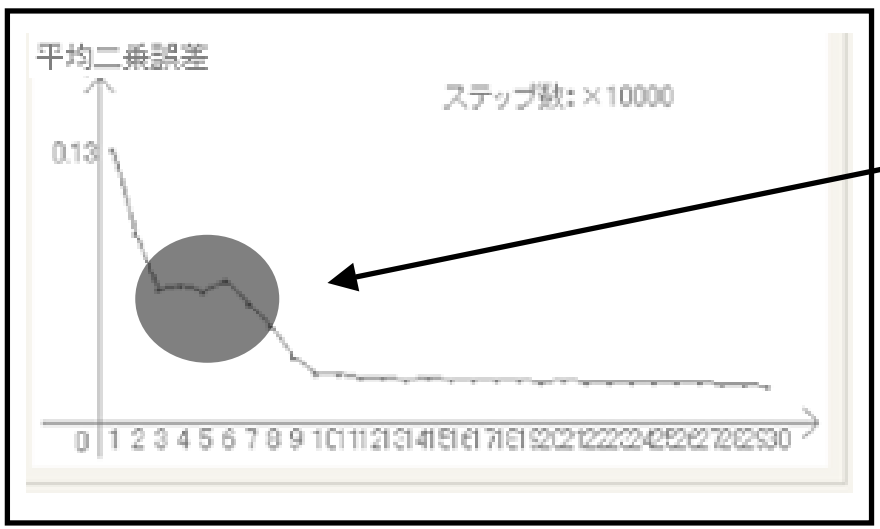
平均二乗誤差：3.0×10^{-5}の限界点

この誤差を許容できるときは有効である！

6．今後の課題

学習法：ステップ幅固定の最急降下法を用いている

局所最適解に陥って誤差が減少しなくなっている可能性



局所最適解に陥ったと思われる箇所

ランダム探索法などを用いる

参考文献

[１]．馬場則夫，小島史夫，小澤誠一：ニューラルネットの基礎と応用；共立出版株式会社（1994）

[２]．茨木俊秀，福島雅夫：最適化の手法；共立出版株式会社（1993）

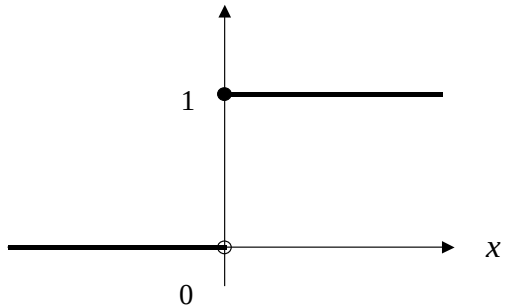
[３]．黒木修一：競合連想ネットの漸近最適性と非線形関数の逐次学習への応用；電子情報学会論文誌　　（pp.１～11）（2002）

参考URL：

[４]http://www.si.hirosaki-u.ac.jp/~slame/neuro/neuron.html

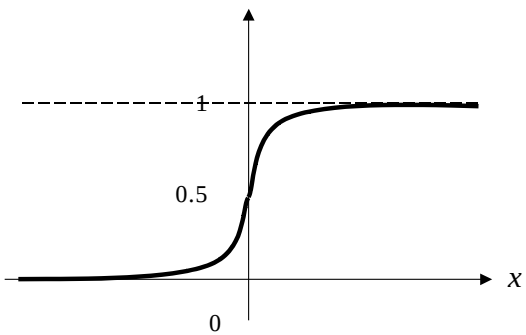
[５]http://www.dc.u-tokai.ac.jp/3laboratory/dc14fujimori/yosoku.html

付録：閾値関数



ステップ関数

$$1[x] = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$



シグモイド関数

$$sig[x] = \frac{1}{1+\exp(-x)}$$

付録：学習方法(BP法)

第P入力に対する第$S-1$層の第jニューロンの出力を y_{jp}

第S層の第iニューロンの出力を x_{ip}としたとき，

ニューロン間の入出力関係を以下のように与える．

$$x_{ip} = h_i(z_{ip}) \qquad z_{ip} = \sum_j w_{ij} y_{jp}$$

(w_{ij}：$S-1$層の第jニューロンとS層の第iニューロン間の結合強度　，$h[]$：閾値関数)

このとき第P入力に対する誤差関数　$E_p(w)$は，第P入力の教師信号を d_{ip}としたとき，誤差関数を $E_p(w) = \frac{1}{2}\sum_i (\overline{x_{ip}} - d_{ip})^2$であらわし，総誤差関数を $E(w) = \sum_p E_p(w)$で示すものとする．この総誤差関数最小化問題を解くために，$E_p(w)$を最小化することを考える．そのため$E_p(w)$の勾配ベクトルを計算し，その逆方向に決められたステップの分だけ w_{ij}を変化させていく．

付録　ニューラルネットのシステム[1]