正規化と細線化を用いたニューラルネットワークの文字認識への応用

正規化と細線化を用いたニューラルネットワークの文字認識への応用

片山　登揚*　　田中 正浩*

An Application of a Neural Network to Characters Recognition Using Normalization and Thinning

Noriaki KATAYAMA　and　Masahiro TANAKA

ABSTRACT

Intellectual systems which enable pattern recognition, prediction etc. are needed for the development of Robotics. One of the realization methods is a neural network system which is modeled on the structure of human brain. In this research, a neural network model for characters recognition is built by using Excel VBA. BP method is used for learning of this model. By making use of this model, Hiragana characters and alphabets shifted vertically and horizontally can be recognized correctly.

Key Words: Neural Network, BP Method, Characters Recognition, Normalization, Thinning

1．はじめに

近年，ロボット工学やコンピュータ技術の急速な発達に伴って，人間らしい動作を行うロボットが登場してきた．しかし，そのほとんどがプログラム通り正確に素早く処理を行うことはできても，人間のようにパターンを認識したり，経験から予測したりすることは困難である．そこで，より高度で柔軟な情報処理系を目指すためには，知能的なシステムを構築する必要がある．

その実現方法のひとつとして，ニューラルネットワークシステムがある．ニューラルネットワークシステムは，学習によって知能を獲得することができ，その応用分野は計測制御をはじめ多方面におよんでいる [1]．そこで，本研究ではニューラルネットワークの応用の一分野として，文字認識を行うニューラルネットワークシステムを構築した．特に，与えられた文字を一定の大きさに拡大する正規化と文字の太さを一様にする細線化の前処理を行うことにより，認識率を向上させることができた．

本小論の構成は，以下のようである．まず，第2節でニューラルネットワークシステムについて簡単に述べる．第3節では，アルファベット文字とひらがな文字の認識における正規化および細線化の前処理と，認識結果について述べる．第4節では，得られた結果と今後の課題をまとめとして述べる．

2003年4月9日 受理

*システム制御工学科 (Department of Systems and Control Engineering)

2．ニューラルネットワークについて

本節では，ニューラルネットワークの基本的考え方とパターン認識問題への応用例を示す．

2.1 ニューラルネットワークと学習アルゴリズム

ニューラルネットワークとは，生物の神経系のもつ特徴的な機能に注目したモデル化手法の一つであり，人間的な並列処理を得意とするアーキテクチャである．

ところで，生物の神経回路網を構成しているニューロンは，細胞体，樹状突起，軸索，シナプスといった成分からなり，入出力端子を持った情報処理素子であるといえる．これをモデル化したものが図 1 のようなニューロンモデルである[2]．図 1 において，X_1，X_2は入力，W_1，W_2は結合強度，θはしきい値，Yは出力を表している．このようなニューロンのモデルを多数組み合わせ，ネットワークを構成していくことで，生物の神経系をシミュレーションすることができ高度な処理が可能になる．

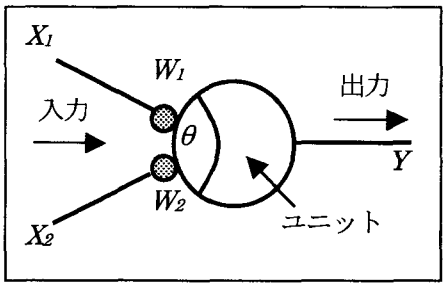

図1 ニューロンのモデル

本研究では，このニューロンを階層的に結合させ，図2に示すような階層構造ニューラルネットワークを構築する．

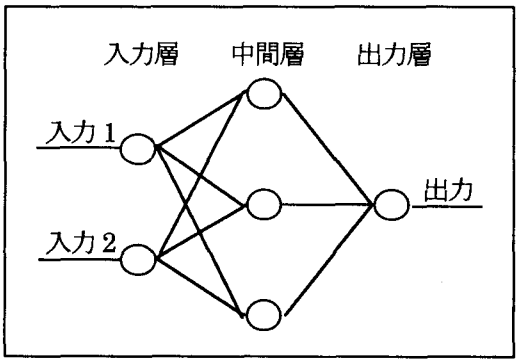

図2　階層構造ニューラルネット

さて，ニューラルネットワークにおける学習とは，定められたアルゴリズムに従ってユニット間の結合強度を変更していくことである．そのアルゴリズムとして様々な手法が提唱されているが，本研究では最もよく用いられているBP(Back Propagation)法を用いる．

BP 法は最急降下法を用いて実際の出力と希望の出力（教師信号ともよぶ）との誤差が最小になるように結合強度を変更する方法で，中間に層がいくつあってもこの方法で誤差信号を逆伝搬させ，どの層の結合強度も変更することが可能であるという特徴がある．

2.2 パターン認識問題への応用

本研究では，入力信号や出力信号の入出力についてはExcelのシートを用い，実際の学習処理についてはExcel VBAを用いて行う．VBAを用いたことで，繰り返し計算をシート上で行うよりも飛躍的に高速に行える点と，コードの書き換えによって容易にニューラルネットワークの仕様を変更することができる点が特長である．

VBA の有効性を見るために簡単なパターン認識問題に対するニューラルネットワークシステムを構築する[2]．簡単なパターン認識問題とは，以下に述べる問題である．

<パターン認識問題>

図 3 のデータ群を用意し，与えられた座標(x,y)におけるデータが何であるかを認識せよ．

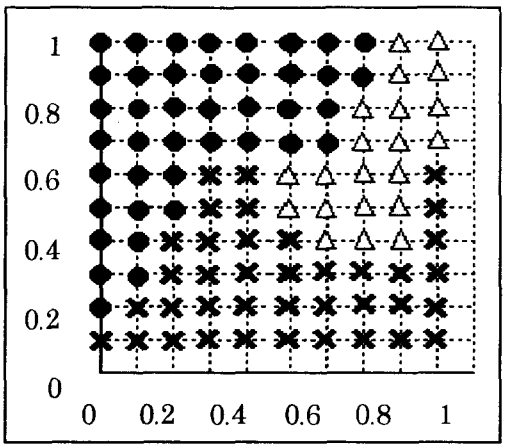

図3 パターン認識問題のデータ群

図3のx座標とy座標を入力値として，その座標の図形データによって出力値を変化させて認識する．出力ユニットを3つ用意し，入力した座標の図形データが"●"の場合は出力ユニットを第1番目から順に0.95，0.05，0.05とし，同様に"△"の場合は0.05，0.95，0.05，"×"の場合は0.05, 0.05, 0.95とするように学習させる．

この例題のパターン認識を実現するニューラルネットモデルの仕様は表1の通りである．

表1 パターン認識問題用ニューラルネットの仕様

ユニット数	3層 入力層ユニット数2 出力層ユニット数3 中間層ユニット数7
入力信号	図3のx座標とy座標
出力信号	入力した座標の図形データによって3パターン(問題参照)
学習方法	BP 法を利用，ステップ幅ηは0.1
備考	1) 結合強度の初期値は0 ～ 1の一様乱数同士の積で決める． 2) 1ステップの学習において，入力パターンの学習順序もランダムに選択する．

作成したモデルで100,000ステップの学習により結合強度ベクトルの修正を行った結果，初期総誤差と比較し

て総誤差は約 15%まで減少した．その様子を横軸ステップ数，縦軸総誤差として図4に示す．また，総誤差は学習前のモデルでの総誤差を 100%とし，その減少を百分率で示している．

学習後のニューラルネットモデルでは，図3のデータを 88%正確に認識することができた．これによって，VBA によるモデル構築の有効性が確認された．

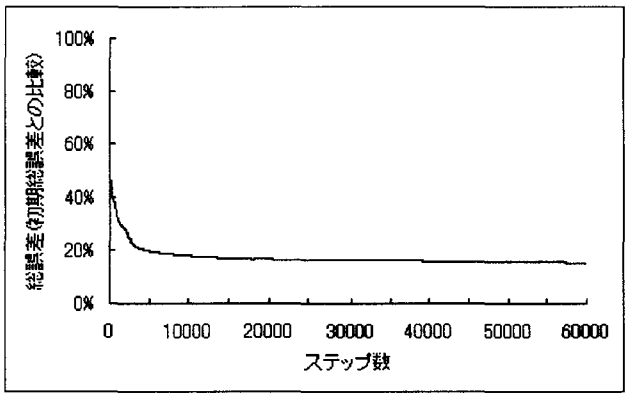

図4　パターン認識問題の学習結果

3．文字認識への応用

本節では，ニューラルネットワークのアルファベット文字とひらがな文字の認識問題への応用について述べる．

3.1 アルファベット文字の認識

まず，認識対象となる文字データは標準的なアルファベット A ～ Z の 26 種である．これらを縦横 16 ピクセルとして作成した学習用文字データの一例を図5に示す．

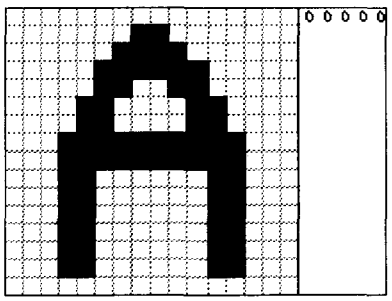

図5　学習用文字データの例 "A"

図5において，文字データの右上にある数値は教師信号である．"A" を 00000(2)とし，1 ずつ加算して "Z" を 11001(2)としている．また，濃淡はセルの値で決定しており，0(淡い) ～ 1(濃い)という条件付きの書式を設定している．

アルファベット文字を認識するニューラルネットモデルの仕様は，表2の通りである．

作成したモデルでおよそ 260 ステップの学習により結合強度ベクトルの修正を行った結果，初期総誤差と比較して総誤差は約 3%まで減少した．その様子を横軸ステップ数，縦軸総誤差として図6に示す．また，総誤差は学習前のモデルでの総誤差を 100%とし，その減少を百分率で示している．

表2 アルファベット文字認識用
ニューラルネットの仕様

ユニット数	3 層 入力層ユニット数 256 出力層ユニット数 5 中間層ユニット数 10
入力信号	入力画像データの画素濃度(全画素を走査)
出力信号	入力した画像によって，対応する文字コード(00000(2) ～ 11001(2))
学習方法	BP 法を利用，ステップ幅 η は 0.1
備考	1) 結合強度の初期値は 0 ～ 1 の一様乱数同士の積で候補を挙げ，予め定めておいた期待値以下の総誤差値が出るまで検討して決める． 2) 1 ステップの学習において，入力パターンの学習順序もランダムに選択する． 3) 学習実行中でも，ユーザ定義割り込みによって処理の中断が可能である．

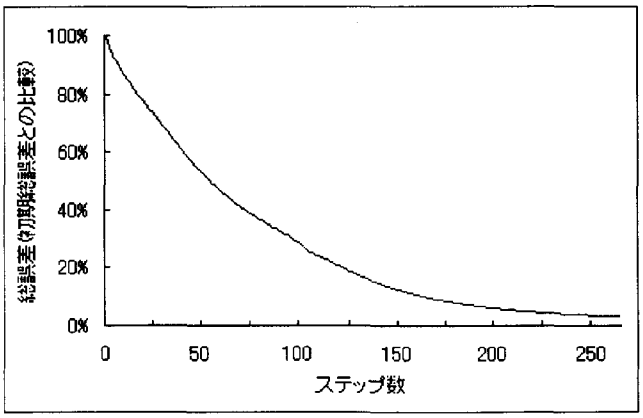

図6　アルファベット文字の学習結果

学習後のニューラルネットモデルでは，学習に使用した 26 種の文字データを 100%正確に認識することができた．

さらに，図7のようなノイズの入った画像や，図8のように，学習させた文字データと多少形状の異なる画像も正しく認識できた．

また，学習用データとして，それぞれのアルファベット文字を90°単位で回転させたデータを加え，26×4で104パターンを学習させることにより，学習に使用した104種の文字データを100%正確に認識できた．ここでは代表例として"A"の学習用文字データを図9に示す．

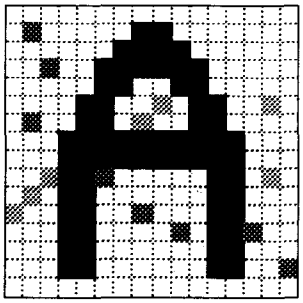

図7 ノイズの入ったデータの例

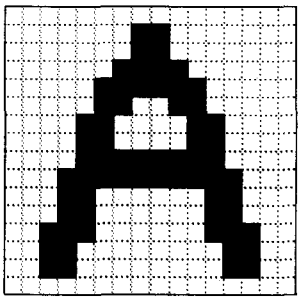

図8 学習データと差異のあるデータの例

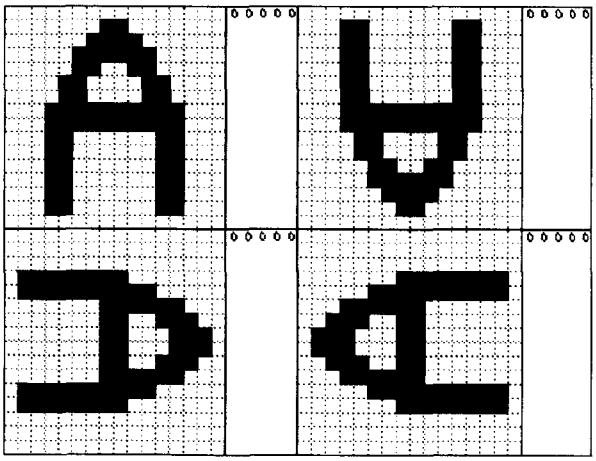

図9 "A"の学習用文字データ(含回転画像)

以下，図10～図13は"C"についての認識結果である．

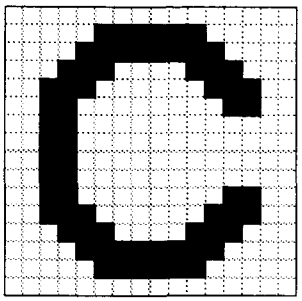

図10 "C"の学習用文字データ

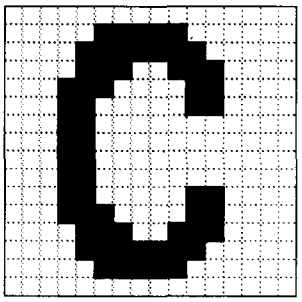

図11 正しく"C"と認識された例1

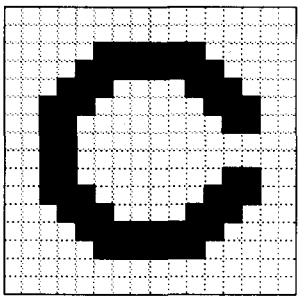

図12 正しく"C"と認識された例2

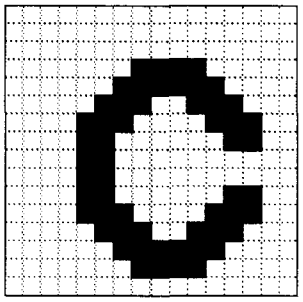

図13 誤って"Q"と認識された例

また，図14～図17は"K"についての認識結果である．

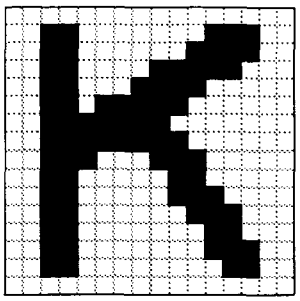

図14　"K"の学習用文字データ

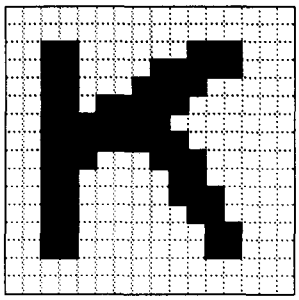

図15 正しく"K"と認識された例1

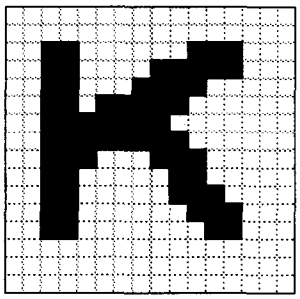

図16 正しく"K"と認識された例2

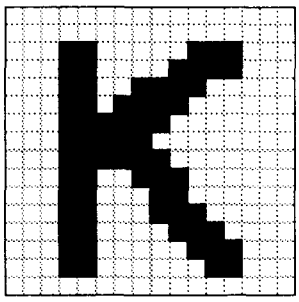

図17 誤って"C"と認識された例

"C" が "Q" , "K" が "C" と誤って認識された例では，入力データの縦横の比が学習させたデータと大きく違うものばかりであった．すなわち，このニューラルネットモデルでは，拡大縮小や平行移動には十分対応できないことが確認された．

3.2 手書きひらがな文字の認識

まず，手書きひらがな文字を想定しペイントツールにひらがな文字を手書きし，Windows API を用いてデバイスコンテキストを取得することでExcelのシート上に手書き文字をトレースさせる．これを学習用文字データとして外部ファイルに蓄積し，BP 法で学習させ，認識させるという一連の手書き文字認識の基本システムを構築した．

また，前節のアルファベット文字の認識方法では以下のような問題があった．

1) 学習に用いたデータを，単純に上下左右方向にシフトして入力しても正しく認識できない．
2) 学習に用いたデータを，単純に拡大縮小して入力しても正しく認識できない．

そこで，これらの問題点を解決し，さらに手書き文字の癖によって発生する入力データの多少のずれに対応するために，次のような手順で処理を行うこととする．

1) 認識対象文字画像を正規化する．
2) 正規化された対象文字画像を細線化する．
3) ニューラルネットワークによって認識する．

この処理内容に合わせて学習用の文字データも予め正規化，細線化し，図 18 のような形式として保存する．つまり，1)，2)はニューラルネットモデルを構築する場合および与えられた文字を認識する場合の前処理である．

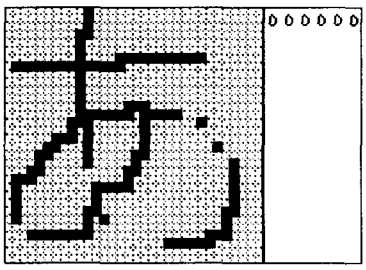

図18 学習用文字データの例"あ"

図 18 において，文字データの右上にある数値は教師信号であり，アルファベットの際と同様に "あ" を $000000_{(2)}$とし，1ずつ加算して "ん" を $101101_{(2)}$としている．

ひらがな文字を認識するニューラルネットモデルの仕様は，表3の通りである．

前処理として行う正規化と細線化について，その手順を述べる．

まず，正規化の手順は次の通りである．

1) 認識対象画像の左端を検出し，枠の左端と合わせるように平行移動させる．
2) 1)で処理された画像の上端を検出し，枠の上端と合わせるように平行移動させる．

3) 2)で処理された画像の右端を検出し，枠の右端と合わせるように拡大処理する．
4) 離散的なエクセルのセル上で拡大すると，値の存在しないセルができてしまうので，3)で処理された画像を補正する．(穴抜け補正)
5) 4)で処理された画像の下端を検出し，枠の下端と合わせるように拡大処理する．
6) 5)で処理された画像の穴抜けを補正する．

次に細線化については Hilditch のアルゴリズムを用いる[3]．これは，認識対象画像の全ての画素を走査し，以下の条件を満たしたときに対象画素を削除するという方法である．

1) 対象の画素値が閾値を超えている．
2) 対象の画素が境界画素である．
3) 対象の画素を削除しても端点を削除しない．
4) 対象の画素を削除しても孤立点を保存する．
5) 対象の画素を削除しても連結性を保存する．

表3 ひらがな文字認識用ニューラルネットの仕様

ユニット数	3層 入力層ユニット数80 出力層ユニット数6 中間層ユニット数10
入力信号	入力画像データの画素濃度
出力信号	入力した画像によって，対応する文字コード(000000(2) ～ 101101(2))
学習方法	BP 法を利用，ステップ幅 η は0.1
備考	1) 結合強度の初期値は0 ～ 1の一様乱数同士の積で候補を挙げ，予め定めておいた期待値以下の総誤差値が出るまで検討して決める． 2) 1ステップの学習において，入力パターンの学習順序もランダムに選択する． 3) 学習実行中でも，ユーザ定義割り込みによって処理の中断が可能である． 4) 認識の前処理として，対象の正規化と細線化を行う．細線化には Hilditch のアルゴリズムを用いる．

図19 ～ 図21に入力データの認識前処理の例として，"あ" の正規化と細線化を示す．

正規化を行うことで，32×32 ピクセルの幅全体に文字が拡大されていることが分かる．また，正規化を行った画像をさらに細線化することで，細さ1ピクセルの画像に変換されていることが分かる．

入力信号は2種類の方法を組み合わせて決める．

まず，縦横8ピクセルのマスクによって画像の左上から順にマスクし，マスクされていない画素の濃度値を全て足し合わせたものを入力値とし，16個の入力データを作成する．これを図22に示す．(図の灰色部分がマスク)

次に，アルファベット文字の認識と同様にそれぞれの画素の濃度値を入力信号とするが，より柔軟なものにするために，縦横4ピクセルごとに足し合わせて1つの入力値とし，64個の入力データを作成する．これを，図23に示す．(太枠内の総和が入力値)

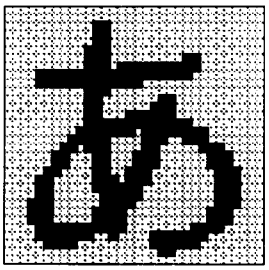

図19 入力データの認識前処理例(元データ)

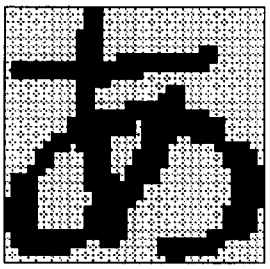

図20 入力データの認識前処理例(正規化)

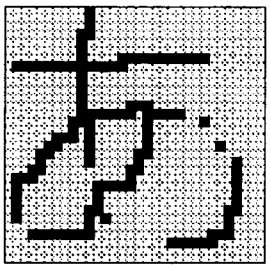

図21 入力データの認識前処理例(細線化)

次に，アルファベット文字の認識と同様にそれぞれの画素の濃度値を入力信号とするが，より柔軟なものにするために，縦横4ピクセルごとに足し合わせて1つの入力値とし，64個の入力データを作成する．これを，図23に示す．(太枠内の総和が入力値)

マスク処理によって算出される16個の入力値と，各画素の濃度値によって算出される64個の入力値の，合わせて80個の入力信号を用いてニューラルネットモデルを構成する．作成したモデルでおよそ3600ステップの学習(結合強度ベクトル修正)を行った結果，初期総誤差と比較して総誤差は約3%まで減少した．その様子を

横軸ステップ数，縦軸総誤差として図24に示す．図24は学習前のモデルでの総誤差を100%とし，学習による総誤差の減少を百分率で示している．

学習後のニューラルネットモデルでは，学習に使用した46種の文字データを100%正確に認識することができた．さらに，前処理によって図25に例として示す学習用文字データのオリジナルを上下左右方向に平行移動，縮小した図26のような文字も正しく認識できた．

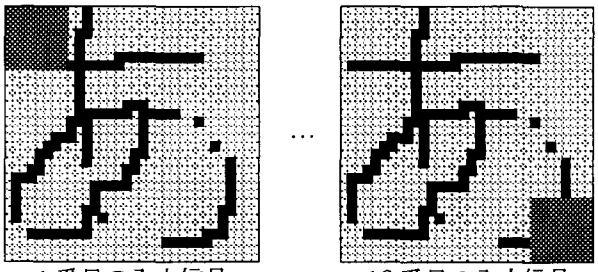

図22 マスク処理による入力値の算出

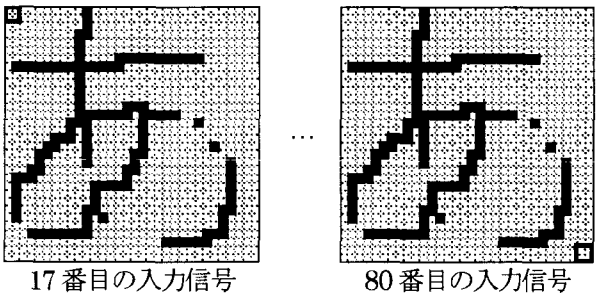

図23 各画素の濃度値による入力値の算出

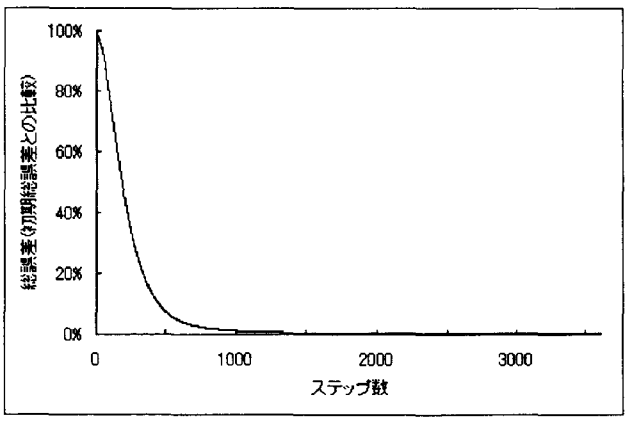

図24　ひらがな文字の学習結果

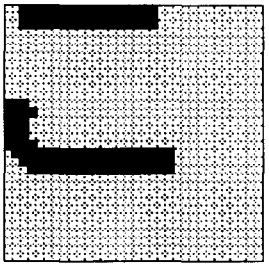

図25 "こ"の学習文字データのオリジナル

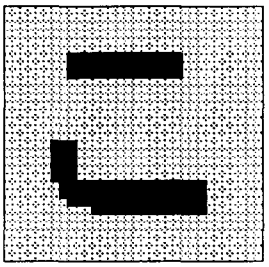

図26 平行移動・縮小した"こ"

以下，いくつかのデータについて認識結果を示す．図27 ～ 30 は "あ" についての例，図 31 ～34 は "を" についての例である．

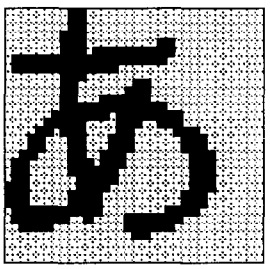

図27 "あ"の学習文字データ

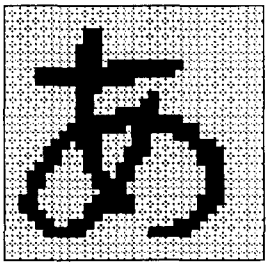

図28 正しく"あ"と認識された例1

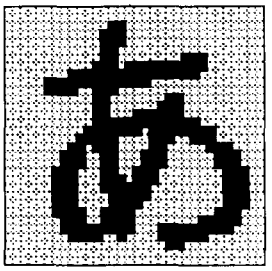

図29 正しく"あ"と認識された例2

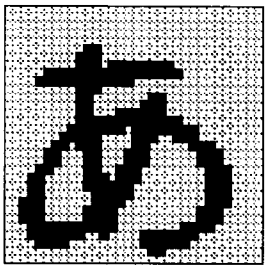

図30 誤って"め"と認識された例

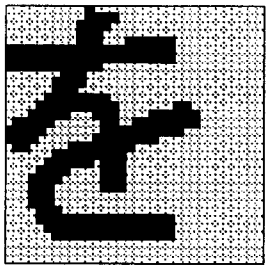

図31 "を"の学習文字データ

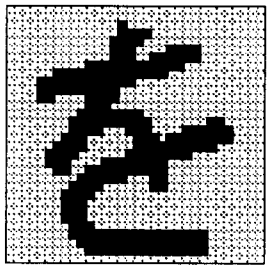

図32 正しく"を"と認識された例1

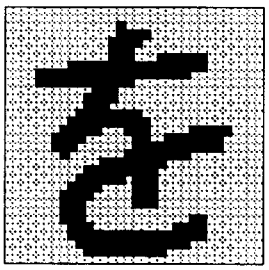

図33 正しく"を"と認識された例2

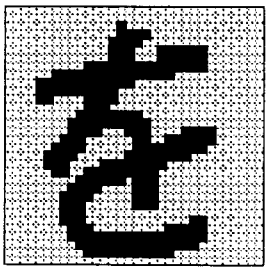

図34 誤って"ち"と認識された例

誤って認識した例のうち，"あ" と "め" については形状が似ている部分がある．しかし，"を" と "ち" は一見すると似ているようには見えない．これは，出力信号の扱いが原因である．本研究では出力信号として文字コードを2進数で扱ったが，これでは6つのビットのみで文字を判別することになるため，このような誤認識が起こってしまったものと考えられる．

4．まとめ

入力値の選び方がニューラルネットの性能に大きく影響することより，単純に座標ごとの画素値を入力値にした場合には柔軟性に欠け，少しのずれで正しく認識できない．これを解決するために，本研究では正規化や細線化，数ピクセルの画素値の総和を入力するなどの工夫を施した．

本研究により，以下のことが確認された．

1)ニューラルネットワークモデルの構築に Excel VBA を用いたことで，効率良くニューラルネットワークモデルの構築ができた．

2) ニューラルネットワークの学習アルゴリズムとして BP 法を用い，満足できる結果が得られた．，誤差を十分減少させるのにかなり時間がかかる場合もあり，中間層の数を増やし入力値を工夫することで，誤差を減少させられることが確認できた．

3) 文字認識への応用に際して，入力データに正規化や細線化といった前処理を施すことにより，認識率が向上した．

さらに，今後の検討すべき課題として次の点が上げられる．

1)文字を認識する場合には，その文字自体の持つ特徴を取り出して入力値とすることが最も望ましいので，入力値の選定を工夫する必要がある．

2)学習アルゴリズムとして BP 法を用いたが，学習アルゴリズムには BP 法以外にもいくつかのアルゴリズムが提唱されているので，これらを適用し比較検討する必要がある．

参考文献

[1] 北村新三，ニューラルネットワーク応用の現状と展望 システム／制御／情報 Vol. 35 No.1 (1991).

[2] 馬場 則夫， 小島 忠男， 小澤 誠一， ニューラルネットの基礎と応用，共立出版(1994)

[3] 大間 大,C 言語による画像処理入門 http://www.katto.comm.waseda.ac.jp/~ohma/b304.pdf 2003 年 1 月 15 日参照