# ニューラルネットによる手書き文字認識の一手法

A Method of Handwritten Character Recognition Using Neural Networks.

菅 沼　義 昇*

Yoshinori SUGANUMA

（1997 年 1 月 6 日受理）

Abstract: The most important problem in neural networks is their generalization ability. For example, it is to be desired that neural n etworks can recognize all unlearned objects by learning only a few instances. In this paper, I propose a method of handwritten character recognition using neural network, examine its ability, and discuss its problems.

## １．はじめに

ニューラルネットにおいて，汎化能力は非常に重要な問題である．汎化能力がなければ，学習は単なるルート学習になってしまい，実用的な処理を行わせるためには膨大な学習例を記憶する必要が出てくる．例えば，文字認識のような場合においても，少数の学習例からそれらの特徴を抽出し，任意の未学習文字を認識できることが望ましい．ニューラルネットに関する汎化能力増大を目的とした研究も多く行われているが [1)~4)]，決定的な方法は存在しないと言って良い．そこで，本報告では，文字認識を例にとり，ニューラルネットに汎化能力を持たせる一つの試みの結果を報告する．現時点では，満足する結果は得られていないが，今後の研究のため，その現状と問題点について述べる．

人間がパターンをどのように処理し，記憶し，また，一般化しているかは，非常に興味ある問題である．著者は過去に幾つかの方法を試みたが [5,6]，そこでは，ニューラルネットワーク的な処理と外部から与えたシンボリックな情報を併用するような形でモデル化した．つまり，直線や曲線といった概念を属性の組み合わせとして記述し，その各属性の値を与えられたパターンから抽出し，かつ，それらの関係（重み）をニューラルネットワーク的手法で学習するといった方法である．

本報告では，ニューラルネットワークの考え方にできるだけ忠実に，与えられたパターンを学習・記憶する方法について検討する．ニューラルネットを利用した文字認識に関する研究も多く見られるが [7,8]，そのほとんどは，与えられたパターンを直接ニューラルネットに入力せず，パターンから抽出された特徴ベクトルをニューラルネットの入力としている [9)~15)]．しかし，人間の文字認識機能を明らかにしようとする立場から見ると，この方法は非常に不自然に

* 理工学部　知能情報学科

みえる．そこで，本報告では，パターンをそのままニューラルネットの入力として与える方法について検討する．

文字認識等，パターン認識を行う場合，人間は非常に少数の例から一般的な情報を引き出し，様々な変化，

(a) 位置の変化
(b) 回転
(c) 拡大・縮小
(d) 変形（例えば，手書き文字）

等に対応しているように思える．そこで，本報告においては，主として数字の認識を対象とし，一組の数字だけの学習により，どの程度の一般化情報を得，かつ，どの程度の未学習文字を認識できるかについて検討する．ただし，位置の変化に対しては，人間はその位置へ視線を移動して対応しているものと思われ，また，横向きや逆向きの文字が人間にとっても読み難いように，大きく回転された文字に対してはそれを正常の位置に戻すような操作が行われているものと思われる．このように，(a)と(b)に対しては，基本的な視覚処理以外のメカニズムが含まれているものと思われるため，本報告では対象外とする．したがって，入力される文字は，常に入力領域のほぼ中央に書かれているものとする．

## ２．ネットワークモデル

提案するニューラルネットの概略を図示すれば Fig.1 のようになる．

まず，入力層は，45 × 45 のユニットから成っている．各ユニットは，0 または 1 の値をとる．

第 1 層は角度を検知するユニットの集まりである．これは，8 つの各方向を検知するユニットのグループに分

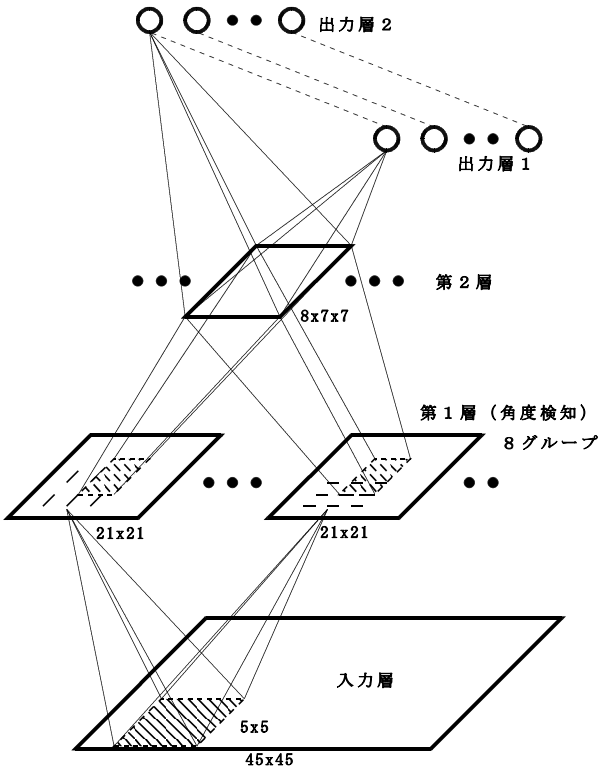


Fig.1　ネットワークモデル

類され，一つの角度グループは，一つ一つが入力層の特定の位置に対応する 21 × 21 個のユニットから成っている．各ユニットは，対応する入力層の 5 × 5 の領域から入力を得ている．この入力層と第 1 層を結ぶリンクの 5 × 5 個の重みは，各角度グループ内では共通であり，以下のようにして学習される．まず，入力層の 5 × 5 個のユニットから構成されるパターンの角度に最も近い角度グループが選択される．次に，重み $w_{ij}$ が次の式で計算され，正規化される．

$$w_{ij}(k+1) = w_{ij}(k) + \rho_1 e_{ij} \qquad i,j=1,\cdots,5 \qquad (1)$$

ただし，

$\rho_1$：パラメータ

$e_{ij}$：入力要素の値（0 または 1）

とする．結局，第 1 層の角度グループ $k$ の$(m,n)$要素の出力 $o_{1kmn}$は，次のようにして計算される．

$$o_{1kmn} = \sum_{i=1}^{5} \sum_{j=1}^{5} w_{ij} e_{ij}, \qquad k=1,\cdots,8, \quad m,n=1,\cdots,21 \qquad (2)$$

第 2 層は，第 1 層の出力結果を出力層へ送るための前処理を行う層であり，複数のグループから成っている．各グループは，第 1 層の各角度グループの中心要素を中心とした 7 × 7 個の要素から入力を得る．したがって，全部で 8 × 7 × 7 個の要素から入力を得ることになる．また，これらを結ぶリンクの重みはすべて 1 である．ただし，第 1 層の一つのユニットが第 2 層の一つのユニットに対応しているわけではなく，グループによって，第 1 層の 2 × 2 または 3 × 3 個の合計出力を 2 または 3 で割った値を一つの入力として取り扱う．これは，拡大・縮小に対応するための処理に相当する．また，中心要素も，位置の微小変動を吸収するため，グループによって異なる．物理的な中心の周囲の各要素を中心とみなして上の処理を行う．従って，第 2 層には，中心要素の数（9 個）×対応関係の種類（1:1，1:2 × 2，1:3 × 3）＝計 27 個のグループが存在する．

第 3 層及び第 4 層は出力層（出力層 1 と出力層 2）であり，それぞれ，分類すべきパターンの種類に対応する個数のユニットからなっている．出力層 1 では，第 2 層の一つのグループを選択し，8 × 7 × 7 個の重み（出力ユニット毎に異なる）を通して入力を得，各出力ユニットの出力を計算する．この処理を第 2 層のグループの数だけ繰り返した後，それらの内，最大値を出力する出力ユニットを含むグループを選択する．そして，この最大値を出力するユニットが認識結果となる．また，どのグループを選んだかの情報（中心要素の位置，ユニットの対応関係）が出力層 2 に送られる．出力層 1 に至る重みの学習は，大きさと入力層が異なるだけで，基本的に(1)式と同じ方法で学習される．したがって，この重みは，あるパターンにおいて入力層で活性化された部分に対応する重みが大きな値を持つように変化する．また，出力値の計算方法も，基本的に，(2)式と同様である．

出力層 2 では，出力層 1 で選択されたグループからの入力によって，最終的な認識が行われる．ただし，重みの学習や出力値の計算方法は今まで述べた方法とは異なる．上で述べたように，出力層 1 に至る経路では，各パターンのどこが活性化されているかといった形で，各パターンの記憶を持つことになる．しかしこの方法には大きな欠点がある．例えば，「一」と「二」のように，その一部として他のパターンを含んでいるようなパターンを識別できない点である．さらに，考え方自体にも大きな問題がある．我々が「もの」を識別する際に重要なの

は，識別する必要性である．そして，識別する際に重要なのは，個々の「もの」が持っている属性値自身の絶対的な値でなく，それらの間の関係である．例えば，四角形だけで構成された世界に住んでいれば，直線，曲線，円，三角形，四角形の識別は必要なく，これらの概念は全く無意味になる．四角形という概念が存在するのは，三角形，円等，四角形と異なるものとして識別すべき他の概念が存在するからである．

そこで，出力層 2 に至る重みは，パターン $p$ とパターン $q$ が異なる場合，以下のようにして学習される．

$$w_{2p,kmn}(k+1) = w_{2p,kmn}(k) + \rho_2 \left| w_{1p,kmn} - w_{1q,kmn} \right|$$
$$w_{2q,kmn}(k+1) = w_{2q,kmn}(k) + \rho_2 \left| w_{1p,kmn} - w_{1q,kmn} \right|$$
$$m,n = 1,\cdots,7, \quad k = 1,\cdots,8$$

ただし，

$w_{2p,kmn}$：第 2 層の$(k,m,n)$要素から出力層 2 のユニット $p$ に至る重み

$w_{1p,kmn}$：第 2 層の$(k,m,n)$要素から出力層 1 のユニット $p$ に至る重み

$\rho_2$：パラメータ

とする．つまり，第 2 層から出力層 1 に至る重みの内，他のパターンと識別するのに重要な部分の重みを増加させるものである．

また，出力層 2 のパターン $p$ に対応する出力ユニットの出力 $o_{2p}$の計算は，以下のようにして行う．

$$o_{2p} = w_{2p,kmn} \exp(-x^2)$$
$$x = \alpha(w_{1p,kmn} - o_{2kmn})$$

ただし，

$o_{2kmn}$：第 2 層の$(k,m,n)$要素の出力を正規化した値

$\alpha$：定数

とする．第 2 層と出力層 1 を結ぶ重みは，第 2 層の出力の大きさに応じてなされるので，学習時とほとんど同じパターンが入力されれば，上の式の $x$ の値はほぼ 0 になるはずである．このように，$x$（及び，$\exp(-x^2)$）は記憶しているパターンとの違いの程度を表している．ただし，重み$w_{2p,kmn}$は，各パターン間の差から学習されるので，重みが小さい部分において学習パターンと大きな隔たりがあっても無視されることになる．

## ３．シミュレーション

すべての層を同時に学習させることも可能であるが，非常に効率の悪いものとなる．そこで，ここでは，各層毎独立に学習を行わせることにする．まず，入力層と第 1 層間を学習させるため Fig.2 のパターンを 2 回与えた．ここで学習した重みは，以下のすべてのシミュレーションで利用した．

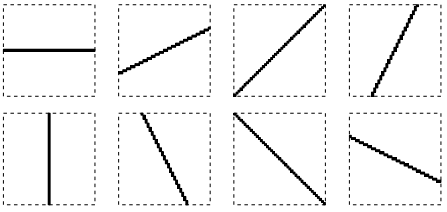

Fig.2 8 つの方向線分

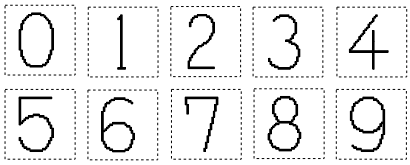

Fig.3 学習した数字

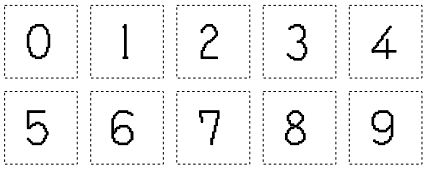

Fig.4 小さな数

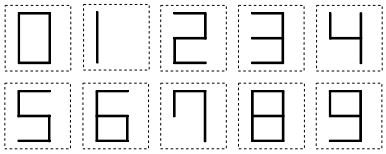

Fig.5 ディジタル数字

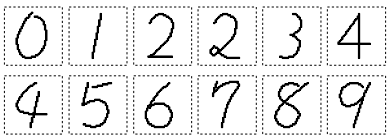

Fig.6 変形した数字

次に Fig.3 に示す数字を学習させた．ここで学習した重みを固定し，小さい数字（Fig.4），ディジタル数字（Fig.5），及び，変形した数字（Fig.6）に対し識別を行った．その結果は Table 1 に示す通りであった．この表において，もっとも大きな数字を出力するユニットが認識結果である．この結果からも明らかなように，このニューラルネットは，ある程度の汎化能力を所有しているように思われる．

そこで，一般的な手書き文字認識を行ってみることにした．手書き文字データとしては，電子技術総合研究所のデータベース ETL1（手書き文字）と ETL6（手本を示した場合の手書き文字）の中の数字を使用した．ただし，認識に入る前に以下に述べるような多少の前処理を行った．ま

Table 1 各種の数字の認識結果

| | | 各出力ユニット（出力層２）の出力 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 学習文字 | 0 | 1.000 | 0.086 | 0.205 | 0.303 | 0.124 | 0.372 | 0.359 | 0.194 | 0.409 | 0.480 |
| | 1 | 0.198 | 0.987 | 0.467 | 0.425 | 0.424 | 0.294 | 0.370 | 0.314 | 0.290 | 0.293 |
| | 2 | 0.282 | 0.273 | 1.000 | 0.512 | 0.269 | 0.229 | 0.277 | 0.314 | 0.335 | 0.334 |
| | 3 | 0.424 | 0.223 | 0.450 | 1.000 | 0.224 | 0.355 | 0.374 | 0.342 | 0.481 | 0.320 |
| | 4 | 0.152 | 0.304 | 0.315 | 0.261 | 1.000 | 0.260 | 0.397 | 0.288 | 0.270 | 0.361 |
| | 5 | 0.575 | 0.160 | 0.208 | 0.339 | 0.201 | 1.000 | 0.475 | 0.261 | 0.442 | 0.391 |
| | 6 | 0.517 | 0.196 | 0.232 | 0.336 | 0.303 | 0.500 | 1.000 | 0.252 | 0.462 | 0.250 |
| | 7 | 0.340 | 0.204 | 0.350 | 0.411 | 0.263 | 0.310 | 0.294 | 1.000 | 0.310 | 0.431 |
| | 8 | 0.542 | 0.150 | 0.297 | 0.426 | 0.208 | 0.407 | 0.407 | 0.226 | 1.000 | 0.177 |
| | 9 | 0.276 | 0.057 | 0.137 | 0.152 | 0.126 | 0.168 | 0.181 | 0.126 | 0.161 | 1.000 |
| 縮小文字 | 0 | 0.702 | 0.150 | 0.224 | 0.289 | 0.283 | 0.431 | 0.451 | 0.251 | 0.459 | 0.362 |
| | 1 | 0.196 | 0.650 | 0.457 | 0.424 | 0.419 | 0.289 | 0.344 | 0.308 | 0.287 | 0.290 |
| | 2 | 0.251 | 0.279 | 0.621 | 0.393 | 0.364 | 0.245 | 0.293 | 0.261 | 0.329 | 0.242 |
| | 3 | 0.400 | 0.226 | 0.395 | 0.644 | 0.259 | 0.325 | 0.395 | 0.386 | 0.392 | 0.337 |
| | 4 | 0.182 | 0.302 | 0.352 | 0.281 | 0.694 | 0.326 | 0.417 | 0.266 | 0.288 | 0.380 |
| | 5 | 0.425 | 0.198 | 0.229 | 0.304 | 0.282 | 0.644 | 0.540 | 0.322 | 0.413 | 0.383 |
| | 6 | 0.552 | 0.197 | 0.262 | 0.345 | 0.302 | 0.417 | 0.674 | 0.252 | 0.543 | 0.223 |
| | 7 | 0.315 | 0.222 | 0.402 | 0.371 | 0.314 | 0.325 | 0.310 | 0.605 | 0.352 | 0.339 |
| | 8 | 0.262 | 0.234 | 0.328 | 0.394 | 0.356 | 0.264 | 0.376 | 0.229 | 0.529 | 0.162 |
| | 9 | 0.238 | 0.185 | 0.319 | 0.272 | 0.270 | 0.225 | 0.324 | 0.399 | 0.295 | 0.427 |
| ディジタル文字 | 0 | 0.468 | 0.163 | 0.254 | 0.425 | 0.237 | 0.445 | 0.438 | 0.325 | 0.421 | 0.391 |
| | 1 | 0.330 | 0.370 | 0.320 | 0.270 | 0.379 | 0.389 | 0.443 | 0.315 | 0.318 | 0.254 |
| | 2 | 0.337 | 0.257 | 0.429 | 0.445 | 0.174 | 0.376 | 0.372 | 0.283 | 0.438 | 0.279 |
| | 3 | 0.345 | 0.186 | 0.279 | 0.448 | 0.135 | 0.373 | 0.374 | 0.300 | 0.420 | 0.347 |
| | 4 | 0.341 | 0.204 | 0.215 | 0.349 | 0.225 | 0.311 | 0.348 | 0.348 | 0.453 | 0.335 |
| | 5 | 0.332 | 0.205 | 0.236 | 0.343 | 0.248 | 0.506 | 0.504 | 0.231 | 0.436 | 0.326 |
| | 6 | 0.486 | 0.152 | 0.207 | 0.253 | 0.249 | 0.539 | 0.586 | 0.226 | 0.419 | 0.314 |
| | 7 | 0.385 | 0.211 | 0.267 | 0.426 | 0.226 | 0.392 | 0.341 | 0.496 | 0.363 | 0.427 |
| | 8 | 0.368 | 0.153 | 0.250 | 0.381 | 0.193 | 0.397 | 0.409 | 0.269 | 0.429 | 0.337 |
| | 9 | 0.346 | 0.151 | 0.235 | 0.375 | 0.168 | 0.398 | 0.385 | 0.267 | 0.431 | 0.359 |
| 変形文字 | 0 | 0.674 | 0.118 | 0.211 | 0.322 | 0.178 | 0.343 | 0.350 | 0.219 | 0.422 | 0.398 |
| | 1 | 0.279 | 0.480 | 0.478 | 0.367 | 0.361 | 0.283 | 0.384 | 0.227 | 0.361 | 0.201 |
| | 2 | 0.288 | 0.228 | 0.605 | 0.499 | 0.229 | 0.229 | 0.238 | 0.343 | 0.307 | 0.401 |
| | 2 | 0.296 | 0.232 | 0.542 | 0.414 | 0.262 | 0.203 | 0.256 | 0.282 | 0.328 | 0.365 |
| | 3 | 0.314 | 0.232 | 0.378 | 0.588 | 0.230 | 0.268 | 0.284 | 0.360 | 0.362 | 0.352 |
| | 4 | 0.165 | 0.271 | 0.344 | 0.290 | 0.747 | 0.261 | 0.385 | 0.303 | 0.292 | 0.339 |
| | 4 | 0.127 | 0.350 | 0.350 | 0.313 | 0.490 | 0.264 | 0.353 | 0.254 | 0.264 | 0.296 |
| | 5 | 0.423 | 0.227 | 0.247 | 0.316 | 0.321 | 0.598 | 0.543 | 0.346 | 0.381 | 0.397 |
| | 6 | 0.516 | 0.291 | 0.299 | 0.545 | 0.288 | 0.400 | 0.492 | 0.211 | 0.575 | 0.142 |
| | 7 | 0.291 | 0.266 | 0.464 | 0.396 | 0.324 | 0.255 | 0.296 | 0.512 | 0.339 | 0.305 |
| | 8 | 0.447 | 0.220 | 0.305 | 0.372 | 0.299 | 0.429 | 0.487 | 0.266 | 0.690 | 0.156 |
| | 9 | 0.283 | 0.210 | 0.357 | 0.331 | 0.265 | 0.318 | 0.362 | 0.508 | 0.440 | 0.305 |

ず，16 階調で表現されている ETL のデータを，各文字における最大値と最小値を求め，その中間値より大きいか否かによって 2 値データに変換した．次は，大きさの問題である．ETL は，64（x 軸）× 63（y 軸）個のピクセルからなっているが，ざっと見てみたところ，ほとんどが数字が 45 X 45 の範囲に含まれたので，各文字を 45 X 45 の窓で切り取った．

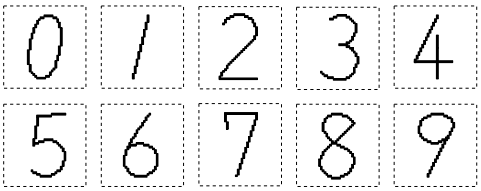

Fig.7 学習パターン

まず，Fig.7 に示すパターンを学習させた．その後，上の処理を行った ETL1 または ETL6 を入力し，認識させた．その結果を Table 2 に示す．この結果からも明らかなように，あまり好ましい結果とはならなかった．そこで，ETL1 及び ETL6 の各々から，ランダムに 100 個を選び，それらを認識させると共に，認識に失敗した文字をその文字の新たな典型例として加えていった（加えるか否かの判断は人間が行った）．この操作により増加した典型例を元に ETL1 及び ETL6 を認識させた結果が Table 3 であり，やや改善されたが未だ不十分であった．

Table 2 ETL1及びETL6に対する認識結果

| 文字 | ＥＴＬ１ | | | | ＥＴＬ６ | | | |
|---|---|---|---|---|---|---|---|---|
| | データ番号 | 個数 | 誤り | 誤り％ | データ番号 | 個数 | 誤り | 誤り％ |
| 0 | 1～ 1421 | 1421 | 681 | 47.92 | 1～ 1383 | 1383 | 159 | 11.50 |
| 1 | 1422～ 2866 | 1445 | 425 | 29.41 | 1384～ 2766 | 1383 | 44 | 3.18 |
| 2 | 2867～ 4310 | 1444 | 958 | 66.34 | 2767～ 4149 | 1383 | 235 | 16.99 |
| 3 | 4311～ 5754 | 1444 | 896 | 62.05 | 4150～ 5532 | 1383 | 193 | 13.96 |
| 4 | 5755～ 7198 | 1444 | 778 | 53.88 | 5533～ 6915 | 1383 | 625 | 45.19 |
| 5 | 7199～ 8642 | 1444 | 1179 | 81.65 | 6916～ 8298 | 1383 | 483 | 34.92 |
| 6 | 8643～10086 | 1444 | 1310 | 90.72 | 8299～ 9681 | 1383 | 228 | 16.49 |
| 7 | 10087～11531 | 1445 | 1180 | 81.66 | 9682～11064 | 1383 | 129 | 9.33 |
| 8 | 11532～12975 | 1444 | 1066 | 73.82 | 11065～12447 | 1383 | 252 | 18.22 |
| 9 | 12976～14419 | 1444 | 221 | 15.30 | 12448～13831 | 1384 | 501 | 36.20 |
| | 全体 | | 8694 | 60.30 | 全体 | | 2849 | 20.60 |

Table 3 ETL1及びETL6に対する認識結果（典型例追加）

| 文字 | ＥＴＬ１ | | | | | ＥＴＬ６ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | データ番号 | 個数 | 誤り | 追加 | 誤り％ | データ番号 | 個数 | 誤り | 追加 | 誤り％ |
| 0 | 1～ 1421 | 1421 | 301 | 3 | 21.18 | 1～ 1383 | 1383 | 79 | 1 | 5.71 |
| 1 | 1422～ 2866 | 1445 | 228 | 4 | 15.78 | 1384～ 2766 | 1383 | 72 | 0 | 5.21 |
| 2 | 2867～ 4310 | 1444 | 750 | 2 | 51.94 | 2767～ 4149 | 1383 | 278 | 0 | 20.10 |
| 3 | 4311～ 5754 | 1444 | 399 | 6 | 27.63 | 4150～ 5532 | 1383 | 164 | 3 | 11.86 |
| 4 | 5755～ 7198 | 1444 | 604 | 5 | 41.83 | 5533～ 6915 | 1383 | 143 | 2 | 10.34 |
| 5 | 7199～ 8642 | 1444 | 671 | 7 | 46.47 | 6916～ 8298 | 1383 | 561 | 0 | 40.56 |
| 6 | 8643～10086 | 1444 | 831 | 3 | 57.55 | 8299～ 9681 | 1383 | 325 | 0 | 23.50 |
| 7 | 10087～11531 | 1445 | 668 | 3 | 46.23 | 9682～11064 | 1383 | 158 | 2 | 11.42 |
| 8 | 11532～12975 | 1444 | 903 | 3 | 62.53 | 11065～12447 | 1383 | 215 | 3 | 15.55 |
| 9 | 12976～14419 | 1444 | 574 | 9 | 39.75 | 12448～13831 | 1384 | 124 | 4 | 8.96 |
| | 全体 | | 5929 | 45 | 41.12 | 全体 | | 2119 | 15 | 15.32 |

このように，十分な結果が得られなかった大きな理由は，ネットワークの汎化能力の不足にある．今回の場合，変形の大きさ，線の太さの違い等が大きく影響したものと思われる．

## ４．結論

シミュレーション結果からも明らかなように，ここで提案したニューラルネットはある程度の汎化能力を持っているといっても良い．しかしながら，その一般化能力は，とても十分なものとは言えない．人間であれば，「6 という数字は円形の部分とそれに付加された曲線の部分からなっている．そして，付加された部分が直線であっても構わないし，また円形の部分も必

ずしも円である必要はない」といったような知識を所有しており，その知識が認識にも利用されていると思われる．非常に少数の例から学習し得るためには，少なくとも，このような知識をニューラルネットに獲得させる必要があり，それが今後の大きな課題である．

## 参考文献


1）喜多一, "ニューラルネットワークの汎化能力", システム／制御／情報, Vol.36, No.10 (1992), pp.625-633.

2）H.Yoshii, "Hierarchical Backward Search: A New Classification Tree Using Preprocessing by Multilayer Neural Network", in D.W.Pearson, N.C.Steele, and R.F.Albrecht eds., *Artificial Neural Nets and Genetic Algorithms*, Springer-Verlag (1995), pp.176-179.

3）M.Maclin and J.W.Shavlik, "Combining the Predictions of Multiple Classfiers: Using Competitive Learning to Initialize Neural Networks", *Proc. IJCAI* (1995), pp.524-530.

4）渡辺栄治, "パターン認識問題に対する階層型ニューラルネットワークの汎化能力改善学習法", 電子情報通信学会論文誌, Vol.J79-D-Ⅱ, No.5 (1996), pp.917-923.

5）菅沼義昇, "単一例による学習とパターン認識", 人工知能学会誌, Vol.5, No.1 (1990), pp.67-80.

6）菅沼義昇, 水野りか, "Ｖ＿ＣＯＲＥＳ：視覚情報処理システム －幾何学的錯視と認識", 静岡理工科大学紀要, Vol.2 (1993), pp.89-118.

7）中野康明, "文字認識・文書理解における最近の動向（１）", 人工知能学会誌, Vol.11, No.5 (1996), pp.702-709.

8）中野康明, "文字認識・文書理解における最近の動向（２）", 人工知能学会誌, Vol.11, No.6 (1996), pp.859-864.

9）井藤好克, 大橋健, 江島俊朗, "手書き文字認識における複数特徴を統合する認識器ＥＩＤ３の提案", 情報処理学会論文誌, Vol.37, No.4 (1996), pp.483-489.

10）猿田和樹他, "係数変化型学習法とその手書き文字認識への応用", 電子情報通信学会論文誌, Vol.J78-D-Ⅱ, No.6 (1995), pp.973-981.

11）猿田和樹他, "排他的学習ネット（ＥＬＮＥＴ）を用いた手書き文字認識の細分類手法", 電子情報通信学会論文誌, Vol.J79-D-Ⅱ, No.5 (1996), pp.851-859.

12）S.Skoneczny, J.Szostakowaski, "Advanced Neural Networks Methods for Recognition of Handwritten Characters", in D.W.Pearson, N.C.Steele, and R.F.Albrecht eds., *Artificial Neural Nets and Genetic Algorithms*, Springer-Verlag (1995), pp.140-143.

13）若林哲史他, "特徴量の次元数増加による手書き数字認識の高精度化", 電子情報通信学会論文誌, Vol.J77-D-Ⅱ, No.10 (1994), pp.2046-2053.

14）若林哲史他, "手書き数字認識における特徴選択に関する研究, 電子情報通信学会論文誌, Vol.J78-D-Ⅱ, No.11 (1995), pp.1627-1638.

15）若林哲史他, "非線形正規化と特徴量の圧縮による手書き漢字認識の高精度化, 電子情報通信学会論文誌, Vol.J79-D-Ⅱ, No.5 (1996), pp.765-774.

16）福島邦彦, "位置ずれに影響されないパターン認識機構の神経回路モデル, 電子情報通信学会論文誌, Vol.J62-A, No.10 (1979), pp.658-665.