

ニューラルネットワークを用いた手書き文字認識

高橋 裕之, 橋場 参生,
波 通隆, 長尾 信一

Hand-written Character Recognition using Artificial Neural Networks

Hiroyuki TAKAHASHI, Mitsuo HASHIBA,
Michitaka NAMI, Shinichi NAGAO

抄 録

生体の優れた情報処理能力に注目し, その機構や構造をモデル化したニューロコンピュータシステムを構築して手書き文字認識応用技術への検討を行った。処理系は, 前処理, 特徴抽出と認識部から成る階層構造を有する。特徴抽出では, 文字の大局的特徴として, 文字パターンの垂直・水平射影演算処理結果と注目点における線分の方向パターンを抽出し, 認識部ではその特徴情報から認識処理を行った。線分の方向パターンと認識処理にニューラルネットワークを応用し, そのネットワークに3層構造のバックプロパゲーションモデルを用いた。ソフトウェアによるシミュレータを作成し, 手書き数字文字と英大文字に対する認識試験を行った。数字と英大文字では, ユニット数が異なる認識処理ネットワークを用いたが, 数字で約97%, 英大文字で約92%以上の認識率が得られた。

1. はじめに

近年の社会の情報化は, 文字を介してコンピュータシステムへ直接アクセスする用例や, データベースを構築するために, 膨大な量の文字データをOCRで読み取る必要性を生み出した。さらに, ワークステーションやパソコンの普及は既に存在する文書を読み取り, 再利用を図るためのフレキシブルな入力装置を求めており, 手書きパターン認識技術は不可欠な技術となってきた。

また, 生体の優れた情報処理能力に注目して, その機構や構造をモデル化したニューロコンピュータ技術は, 学習により自ら問題解決の手段の構築を行い, 従来のコンピュータ技術とは異なった新たなコンピュータ技術を

もたらすと考えられている。そこで, ニューロコンピュータ技術を手書きパターン認識に応用することにより, これまでとは異なった特徴を有する処理系が望め, 各々の特徴を活かしたシステム開発が可能と思われる。

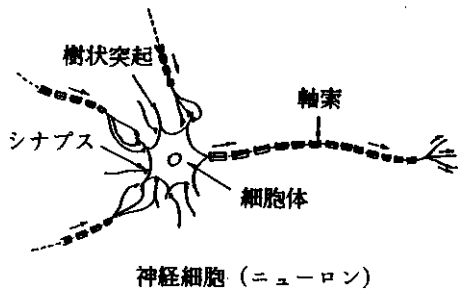
本研究では, 手書きパターン認識システムの開発を目的として, ニューラルネットワークを用いた手書き文字認識の検討を行い, 手書き数字と手書き英大文字に対して認識試験を行った。

2. ニューロコンピュータ

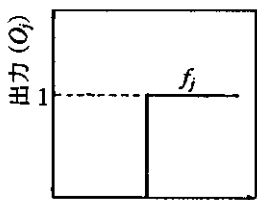
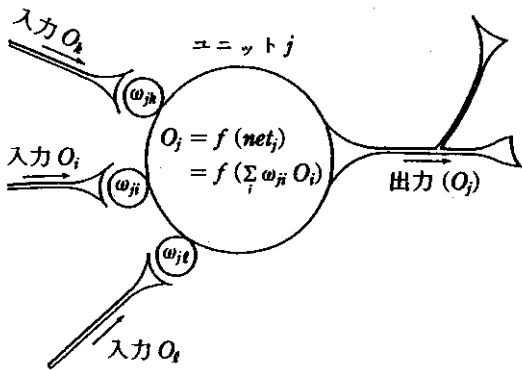
現在のコンピュータにおける情報処理は, 主に対象となる問題を手続きに従った操作に分解し, 各操作を順に

実行して問題を解いていく。このコンピュータは様々な作業を行うことができ、非常に有効なツールである。しかし、曖昧なデータや手続きが明確でない場合の処理等に対する課題はある。

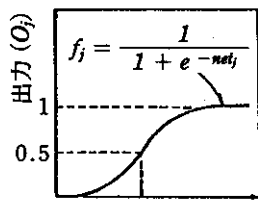
近年、これらのコンピュータ理論とは別の並列情報処理の原理に基づく数理モデルが注目を浴びている。これは、生体の情報処理、とくに脳の情報処理に学んだものであり、現在のコンピュータに対する課題を補おうとするものである。具体的には、図1のような脳の基本素子であるニューロン(ユニット)を多入力1出力の工学的モデルに表現し、それをシナプス荷重(重み)により多数個結合してネットワークを構成し、このネットワーク



神経細胞 (ニューロン)



しきい値処理



Sigmoid 関数

図1 ニューロンのモデル

の中で、同時かつ並列に情報をやりとりすることで情報処理を進め、重みを変えることにより学習を行う。これらは、神経回路網モデル、ニューラルネットワークモデル、コネクショニストモデルなどと呼ばれている。こうしたモデルを利用したコンピュータをニューロコンピュータと呼び、脳の持つ優れた情報処理能力を模範として工学的応用を行うものである。

現在、ニューラルネットワークモデルとして、バックプロパゲーションモデル、ポプフィールドモデル、ボルツマンマシンなどが提案されている。バックプロパゲーションモデルはニューラルネットワークの代表的なモデルであり、学習に大きな特徴がある。

2.1 バックプロパゲーション (BP) モデル

入力層、出力層とその間の任意の中間層(隠れ層)から成る階層型ネットワーク構造(図2)の代表的学習則がバックプロパゲーション学習則(BP法)であり、このBP法によるモデルをバックプロパゲーション(BP)モデルと呼ぶ。このモデルは入力層以外の各層は前層の全ユニットと結合しており、入力層に任意の入力パター

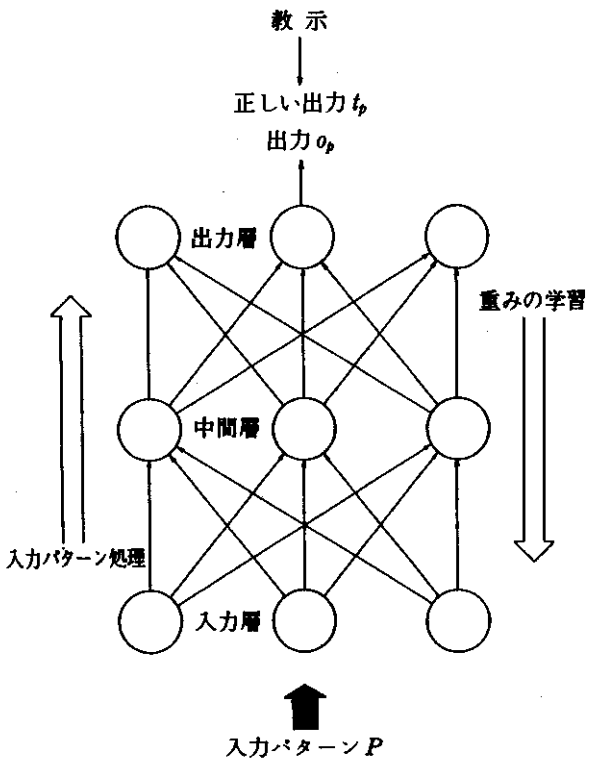


図2 ネットワーク構造

ンを与えると、情報はネットワークの結合の重みに基づき、中間層、出力層と伝播して出力を出す。この出力に対して、本来出力されるべき正しい値を与えることにより、結合の重みを自発的に変える。この繰り返しにより、ネットワークは、正しい答えを出力するようになる。このように結合の重みを変えることを学習という。

2.2 BP 学習則

ユニットは複数の他のユニットからの出力結果とその重みによる入力値により内部状態を変え出力を出す。ネットワークの学習はこの重みを変えていくことである。

あるユニット j の入力値は、複数のユニットからの入力の総和であり、次式で表される。

$$net_{pj} = \sum_i \omega_{ji} o_{pi} \quad (1)$$

ただし ω_{ji} はユニット i からユニット j への重み係数、 o_i はユニット i の出力結果である。

ユニットは入力値により内部状態を変える。その内部関数は各ユニット毎に違っていても構わないが、一般にはしきい値処理や sigmoid 関数を使うことが多く、本研究では sigmoid 関数を用いた。sigmoid 関数は微分可能な、疑似線形関数で

$$f_j = \frac{1}{1 + e^{-net_j}} \quad (2)$$

で表せる。さらに、しきい値を加えて

$$f_j = \frac{1}{1 + e^{-(net_j + \theta_j)}} \quad (3)$$

とする場合もある。

ネットワークは階層構造を採り、入力層、中間層、出力層の方向へ結合し、逆方向の結合や層内での結合は基本的には存在しない。

入力層の各ユニットに与えられた入力データはユニットで変換され中間層を経て出力層から出力される。その出力値と、正しい出力値を比べ、その差を減らすように結合の強さを変える。

BP 法では、あるパターン p を与えたとき、出力ユニット j の実際の出力値 (o_{pj}) と正しい出力値 (t_{pj}) の誤差を次のように定義する。

$$E_p = \frac{1}{2} (t_{pj} - o_{pj})^2 \quad (4)$$

これは出力ユニット j のエラーを表す。 t_{pj} は人間が与える教示データである。学習させるにはこのエラーを減

らすようにすべての結合の強さを変えればいい。ここではパターン p を与えたときの ω の変化量を次式のように定義する。

$$\Delta_p \omega_{ji} \propto - \frac{\partial E_p}{\partial \omega_{ji}} \quad (5)$$

式 (5) の右辺を次式とする。

$$\frac{\partial E_p}{\partial \omega_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial \omega_{ji}} \quad (6)$$

式 (1) より $net_{pj} = \sum_k \omega_{jk} o_{pk}$ であるので式 (6) の右辺の右項は次式のようにになる。

$$\frac{\partial net_{pj}}{\partial \omega_{ji}} = \frac{\partial}{\partial \omega_{ji}} \sum_k \omega_{jk} o_{pk} = o_{pi} \quad (7)$$

また、 δ_{pj} を次式のように定義する。

$$\delta_{pj} = - \frac{\partial E_p}{\partial net_{pj}} \quad (8)$$

式 (7)、(8) から、式 (6) は次式のようにになる。

$$- \frac{\partial E_p}{\partial \omega_{ji}} = \delta_{pj} o_{pi} \quad (9)$$

したがって、式 (5) は次式のように書き直せる。

$$\Delta_p \omega_{ji} = \eta \delta_{pj} o_{pi} \quad (10)$$

ここで、 η は定数である。

式 (8) の δ_{pj} は、そのユニットが出力層か中間層かで異なる。 $o_{pi} = f(net_{pi})$ であるから、出力層の場合は次式のようにになる。

$$\begin{aligned} \delta_{pj} &= - \frac{\partial E_p}{\partial net_{pj}} = - \frac{\partial E_p}{\partial o_{pi}} \cdot \frac{\partial o_{pi}}{\partial net_{pj}} \\ &= - \frac{\partial \{1/2(t_{pj} - o_{pi})^2\}}{\partial o_{pi}} \cdot \frac{\partial o_{pi}}{\partial net_{pj}} \\ &= (t_{pj} - o_{pi}) f'_j(net_{pj}) \end{aligned} \quad (11)$$

また、中間層の場合は

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial o_{pj}} \quad (12)$$

とすると、式 (1) より次式のようにになる。

$$\begin{aligned} \frac{\partial E_p}{\partial o_{pj}} &= \sum_k \frac{\partial E_p}{\partial net_{pk}} \cdot \frac{\partial}{\partial o_{pj}} \sum_i \omega_{ki} o_{pi} \\ &= \sum_k \frac{\partial E_p}{\partial net_{pk}} \omega_{kj} \\ &= \sum_k \delta_{pk} \omega_{kj} \end{aligned} \quad (13)$$

$$\begin{aligned} \delta_{pj} &= \frac{\partial E_p}{\partial net_{pj}} = - \frac{\partial E_p}{\partial o_{pj}} \cdot \frac{\partial o_{pj}}{\partial net_{pj}} \\ &= f'_j(net_{pj}) \sum_k \delta_{pk} \omega_{kj} \end{aligned} \quad (14)$$

式 (13) は δ の再帰関数となる。 $\Delta \omega$ の計算は、出力層のユニットから中間層ユニットに向かって進める。中間層ユニットでは、 $\Delta \omega$ はその出力側の $\Delta \omega$ が決まらなると計算できない。したがって、出力層の誤差を初期値として、再帰的に行われることになる。このように、学習は入力データと逆の後向きに進む。これがバックプロパゲーション（誤差逆伝播）法の名前の由来である。

したがって、BP 法による学習は、学習データを入力し、結果を出力する。結果のエラーを減らすように、結合の強さを変える。これを収束するまで繰り返す。

実際に行う場合には、重みの振動を防ぐために慣性項を加える方法が有効であり、

$$\Delta \omega_{ji}(n+1) = \eta \delta_{pj} o_{pi} + \alpha \omega_{ji}(n) \quad (15)$$

のようにする。 n は学習の回数を表す。右辺の第 1 項はいま求めた $\Delta \omega$ であり、第 2 項を加えることにより、重みの変化に一種の慣性が生じ、エラーの曲面の細かな凹凸を無視する効果が得られる。これにより、エラーの振動を防ぎつつ、実効的な重みの変化量を大きくとることができ、学習速度を高速にする効果がある。

3. 手書き文字認識ネットワークモデル

人間の視覚機能は、網膜に投影されたパターン情報を脳に伝達し、その情報から特徴抽出を行い、見たものの形状・色に関する処理が行われ、一方では距離や奥行き

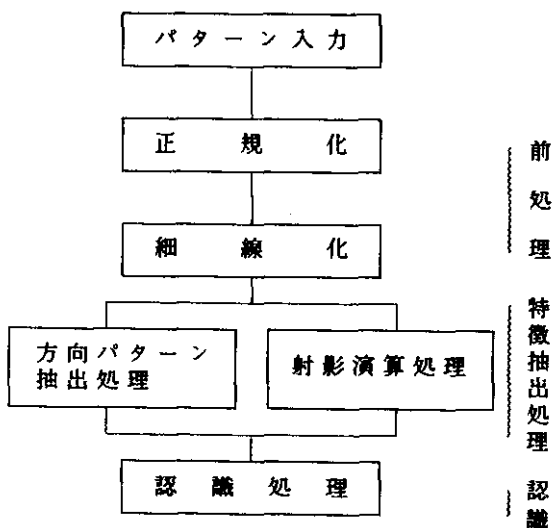


図3 手書きパターン認識フロー

に関する空間的な処理が行われて、ものの認識・識別や運動の指令が行われる。人間はこれらの高度な処理をリアルタイムで行っている。これは、視覚系で特徴抽出、情報圧縮を行い、高度な特徴情報のみを脳に伝達し処理を行っているからであると言われている。

そこで、図 3 に示すフローのような、前処理や特徴抽出を行い、その特徴情報から、認識処理を行う階層構造モデルを構成し、手書き文字認識への応用を検討した。図 4 に英大文字でのブロック図を示す。

3.1 前処理部

文字認識技術では、人間のように自由に書かれた様々な大きさやバランスの文字を認識可能なことが最終的な課題であろうが、現時点での到達は困難である。そこで、文字の大きさやバランスを整え、認識が容易となることを目的に前処理を行った。

手書き文字のパターン情報は、72×72 からなる領域に '0', '1' の情報として直接入力される。この入力パターンに対し、非線形正規化処理を施し、大きさを揃え 32×32 の情報に変換する。さらに、細線化処理を行った。この結果を特徴抽出処理の入力情報とした。

3.1.1 非線形正規化処理

正規化処理は、文字の大きさ、位置を整える操作であり、線形正規化であるアフィン変換が多く用いられている。アフィン変換は、印刷文字のように形が整った文字に適しているが、変形箇所が多い手書き文字では変形部分がそのまま残るので十分ではない。

本研究では、線密度イコライゼーションによる非線形正規化処理を用いた。これは、線の密なところを引伸ばし、粗なところは縮めるという非線形な正規化であり、空間内の線密度が均質化され、空間の有効利用が図れる。具体的には、旧座標の各標本点に特性値として局所的な線密度を定義し、その特性値と各点で可変となる標本化間隔との積が一定になるように変換する方式であり、以下に示す。

一定の標本化間隔 δ で標本化された 2 値図形を

$$f(x_i, y_j), \quad i = 1, 2, \dots, I, j = 1, 2, \dots, J \quad (16)$$

とする。 (x_i, y_j) は、 x 軸、 y 軸の各々 i 番目、 j 番目の標本点の座標である。この時、標本化される対象となった元の標本点以外の点、すなわち連続座標系上での値は

$$f(x, y) = f(x_i, y_j)$$

$$(x_{i-1} =)x_i - \delta < x \leq x_i, x_0 = 0$$

$$(y_{j-1} =)y_j - \delta < y \leq y_j, y_0 = 0$$

(17)

であるとする。

ここで、各 x_i における X 軸への線の本数の射影

($x = x_i$ なる垂直線と 2 値関数 f との交差数)

$$h_x(x_i) = \sum_{j=1}^J f(x_i, y_j) \cdot \bar{f}(x_i, y_{j-1}) + \alpha$$

$$f(x_i, y_0) = 0$$

$$h_y(y_j) = \sum_{i=1}^I f(x_i, y_j) \cdot \bar{f}(x_{i-1}, y_j) + \alpha$$

$$f(x_0, y_j) = 0$$

(18)

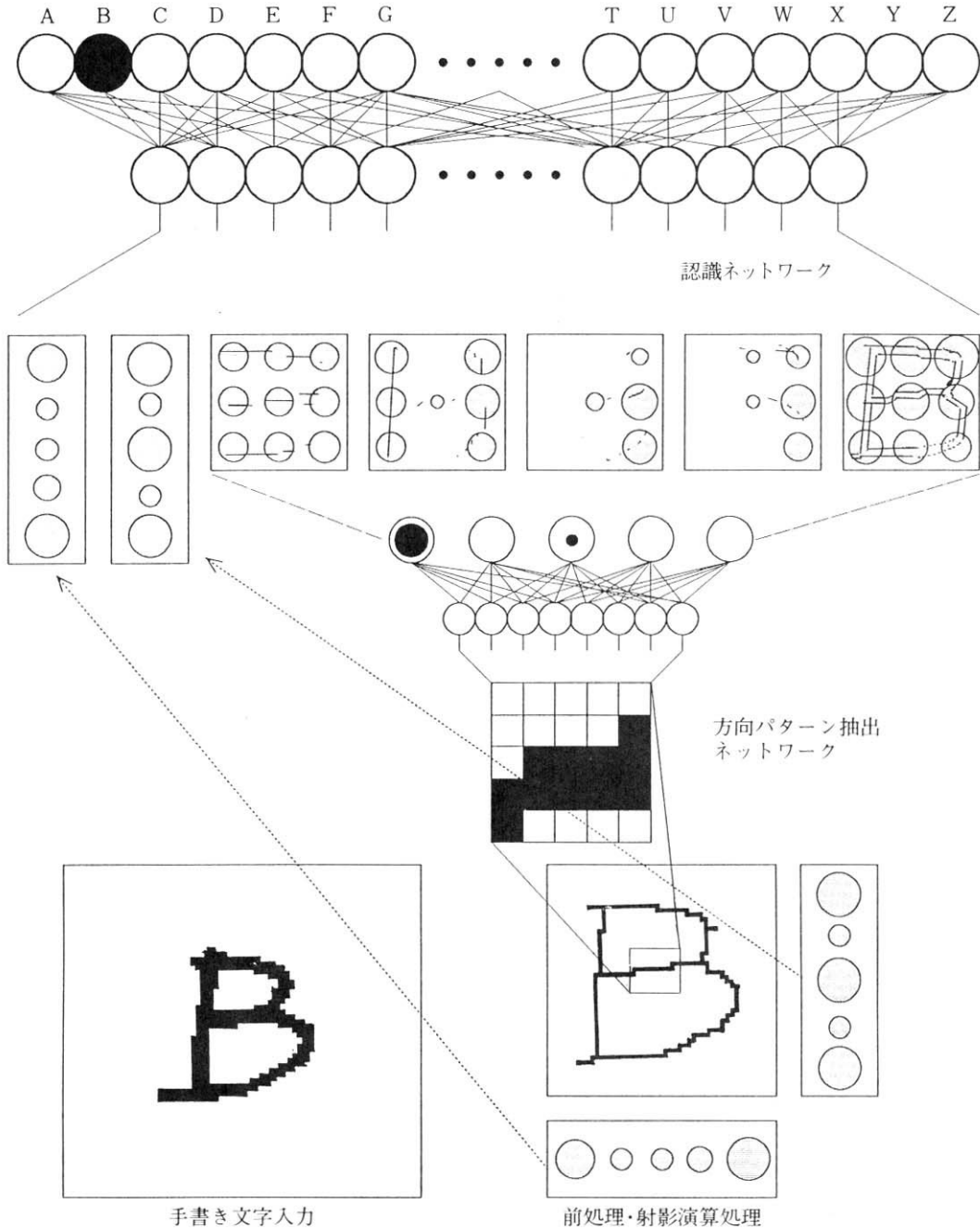


図4 手書き英大文字認識処理ブロック図

を求める。ただし、 \bar{f} は、 f の0-1の反転図形である。また、 α は、外接長方形（文字図形を含む最小の長方形）の外側では $\alpha = 0$ 、内部では > 0 とする。これにより、外部の白地は無視されて、正方形に正規化されるが、内側の白地は、線の本数が1本の部分の $\alpha / (1 + \alpha)$ だけの重みで非線形変換される。なお、 α は大きくすればするほど非線形性が弱まり、内部の α を無限大にすると、従来の外接長方形による線形変換に等しくなる。

さらに、この射影された関数、式(18)も、式(17)と同様に、連続座標系上では、

$$\begin{aligned} h_X(x) &= h_X(x_i), \quad x_i - \delta < x \leq x_i \\ h_Y(y) &= h_Y(y_j), \quad y_j - \delta < y \leq y_j \end{aligned} \quad (19)$$

と考える。

次に、それぞれの線の総和を求める。

$$\begin{aligned} N_X &= \sum_{i=1}^I h_X(x_i) \\ N_Y &= \sum_{j=1}^J h_Y(y_j) \end{aligned} \quad (20)$$

以上の準備のもとで、正規化図形

$$g(x'_i, y'_j), \quad i = 1, 2, \dots, I, \quad j = 1, 2, \dots, J \quad (21)$$

の標本点 (x'_i, y'_j) は、次の条件を満足するよう定められる。

$$\begin{aligned} \epsilon_X(i) \cdot h_X(x'_i) &= \text{constant} (= \delta \cdot \frac{N_X}{I}) \\ \epsilon_Y(j) \cdot h_Y(y'_j) &= \text{constant} (= \delta \cdot \frac{N_Y}{J}) \end{aligned} \quad (22)$$

ここで、 ϵ_X, ϵ_Y は、可変な標本化間隔であり

$$\begin{aligned} x'_i - \epsilon_X(i) &= x_{i-1} \\ y'_j - \epsilon_Y(j) &= y_{j-1} \end{aligned} \quad (23)$$

である。すなわち、標本化間隔と線密度の積が一定になるように再標本化される。なお、定数を $\delta \cdot N_X / I$ および $\delta \cdot N_Y / J$ にするのは、正規化図形のメッシュ数も $(I \times J)$ にするためである。

以上が本正規化の意図であるが、これを実行するために、 h の累積関数を定義する。

$$\begin{aligned} H_X(x) &= \int_0^x \frac{h_X(x')}{\delta} dx' \\ H_Y(y) &= \int_0^y \frac{h_Y(y')}{\delta} dy' \end{aligned} \quad (24)$$

ここで、 h は、式(19)で定義された連続座標系上での

関数である。

このようにすれば、新しい標本化図形 $g(x'_i, y'_j)$ の標本点は、

$$\begin{aligned} x'_i &= \{x \mid H_X(x) = i \cdot \frac{N_X}{I}\} \\ y'_j &= \{y \mid H_Y(y) = j \cdot \frac{N_Y}{J}\} \end{aligned} \quad (25)$$

と計算される。

なお、式(24)(25)による正規化は、連続座標系上の元の図形からの再標本化という立場で定式化されている。離散系により、しかも再標本化ではなく2値図形 f から求める場合には、

$$\begin{aligned} i' &= \min \{l \mid \sum_{k=1}^l h_X(x_k) \geq (i - 0.5) \cdot \frac{N_X}{I}\} \\ j' &= \min \{m \mid \sum_{k=1}^m h_Y(y_k) \geq (j - 0.5) \cdot \frac{N_Y}{J}\} \end{aligned} \quad (26)$$

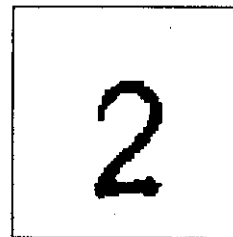
なる i', j' に対し、 $x'_i = x'_{i'}$ 、 $y'_j = y'_{j'}$ として変換する。すなわち、かっこ内の条件を満足する最小の l, m をそれぞれ i', j' とすると、正規化図形の X 軸の i 番目、 Y 軸の j 番目の標本値 $g(x'_i, y'_j)$ は、元の図形のそれぞれ i, j 番目の標本値 $f(x'_i, y'_j)$ として計算される。

非線形正規化処理結果を図5-(a), (b)の(2)に示す。

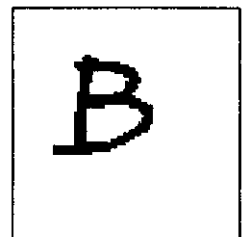
3.1.2 細線化処理

正規化した文字パターンから期待される方向パターンを容易に抽出可能なように細線化処理を施した。

細線化とは、与えられた図形から線幅を細めて幅1の



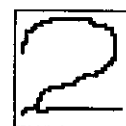
(1) 原画像



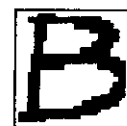
(1) 原画像



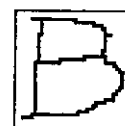
(2) 正規化



(3) 細線化



(2) 正規化



(3) 細線化

図5-(a) 前処理例

図5-(b) 前処理例

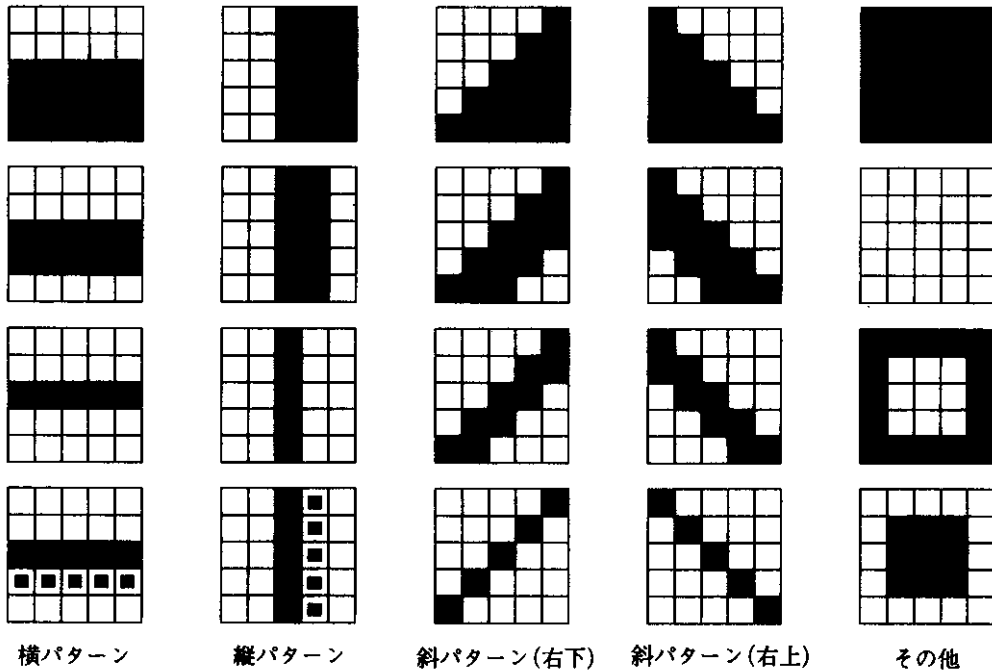


図6 方向パターン学習データ例

中心線を抽出する操作である。細線化の結果からは図形の結合関係の特徴が求められるので、字や図面などの線状図形から線の構造を解析するのに不可欠な処理となっている。

細線化では多くのアルゴリズムが提案されているが、基本的には「画像中の境界点中から、消去可能要素であり、かつ、線の端点ではない画素をすべて消去する」というものである。本モデルでは、Hilditchの逐次型アルゴリズムによる2値図形の細線化処理を行った。

Hilditchのアルゴリズムは、入力2値画像をTVラスタによって繰り返し走査しながら、次の1)~5)の条件をすべて満足する‘1’画素を除去し、1回の画面走査において除去される画素がなくなったとき、処理を終了するというものである。

[各回の画面走査を開始したときの画像の状態(並列状態)における条件]

- 1) 4近傍に‘0’画素が存在する
- 2) 8近傍に2つ以上の‘1’画素が存在する
- 3) 8連結数が1である

[注目画素を走査したときの画像の状態(逐次状態)に対する条件]

- 4) 8近傍に‘1’画素が存在する
- 5) 既走査の4近傍画素の除去により、8連結数が変

化しない

細線化処理結果を図5-(a), (b)の(3)に示す。

3.2 特徴抽出部

文字の特徴量として、ある注目点における線分の方向パターン情報と文字の大局的な特徴を表すものとして射影演算処理結果を用いた。

方向パターン抽出処理では、注目点の5×5の近傍を入力データとし、縦線“|”，横線“—”斜線“/”，“\”に反応するニューラルネットワークを構築し、その方向情報の検出を行った。

ネットワークは3層構造からなり、ユニット数は入力層25，中間層8，出力層は前述の方向パターンに対する4に加えてそれ以外を示すユニットを含めて5とし、BP法により学習を行った。学習パターンの一部を図6に示す。方向パターン抽出は、前処理結果の32×32の領域をTVラスタ走査により、注目点に‘1’画素が存在するか、注目点の8近傍に‘1’画素が2以上存在する点のみについて処理を行った。結果は、5種類の方向パターン各々に36×36の情報として得られるが、特徴データ圧縮と多少の文字ずれや変形の吸収を目的として、3×3のエリアへの頻度情報として0.0~1.0の値に変換した。

また、これまでの研究から方向パターンのみの特徴量では、認識率に限界が生じると考え、認識率向上を目的として、他の特徴量を付加することを検討した。

本研究では、文字の大局的情報として、射影演算処理情報の特徴量として付加した。射影演算処理では、垂直方向、水平方向それぞれの処理結果を5領域に分割し、それぞれの総和が1になるように正規化処理を施した。

以上、本研究では、射影演算処理10、方向パターン情報45の合計55の特徴情報を得て認識処理を行った。

3.3 認識部

認識部も3層のニューラルネットワークで構成した。ユニット数は、入力層には方向パターンと射影演算からの出力結果55、中間層、出力層は数字認識と英大文字認識では異なり、数字認識では中間層12、出力層は0から9の文字に対応した10ユニット、英大文字認識では中間層15、出力層AからZに対応する26ユニットを用いた。

数字、英大文字共にユニット数が異なるだけで、基本的には同じ構造からなる。

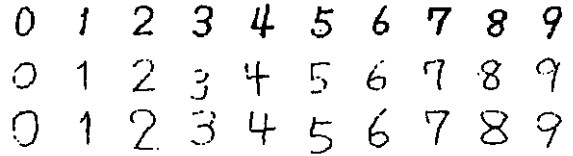
4. ネットワークシミュレータ

上述したモデルのシミュレータをパソコン(PC-9801RA 21)を用いてソフトウェア(C言語)により作成した。プログラムはBP部をライブラリとして作成し、これを基に以下のように機能別に構成した。

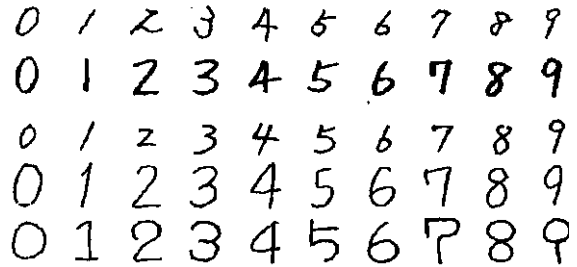
- 1) 方向パターンデータにより、方向パターン抽出ネットワークの学習



図7 手書きパターン認識システム



a) 常用手書き



b) 自由手書き

図8 手書き数字パターン

- 2) 学習用文字データから、前処理を行い、方向パターンおよび射影演算処理による特徴データ出力
- 3) 特徴データにより、認識ネットワークの学習
- 4) 手書き文字データにより、文字認識処理

また、これらの処理を一括して行うプログラムをEWS (NWS-1460: SONY製)で作成し、処理を行った。図7はシステム図である。

4.1 手書き数字認識処理

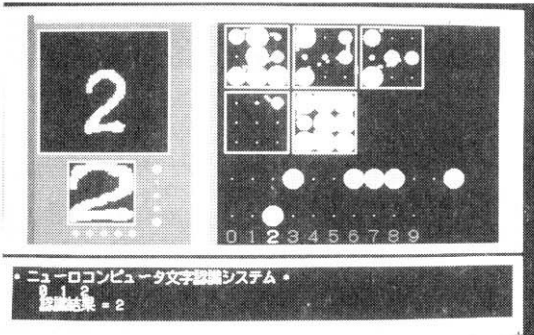
サンプルの手書き数字データは、常用手書きと自由手書きにわけた。常用手書きは、文字の形状に制約を設け

表1 手書き数字認識処理結果

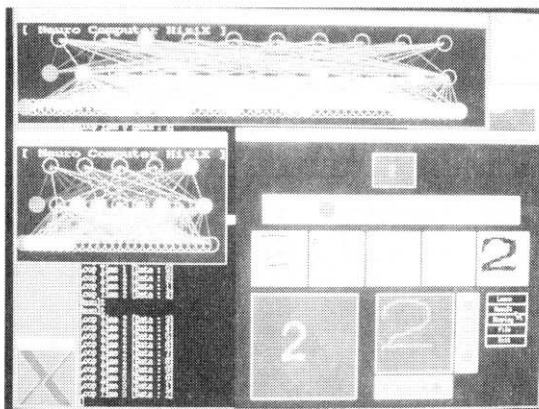
		常用手書き	自由手書き	合計
サンプル数		250 (10×25)	500 (10×50)	750 (10×75)
net 1	学習データ	60	240	300
	未学習データ 認識数	98.9% 188/190	96.2% 250/260	97.3% 438/450
	全認識率	99.2%	98.0%	98.4%
net 2	学習データ	150	150	300
	未学習データ 認識数	100.0% 100/100	96.3% 337/350	97.1% 437/450
	全認識率	100.0%	97.4%	98.3%

たものである。サンプル数は、常用手書き文字 25×10、自由手書き文字 50×10 である。サンプルの一部を図 8 に示す。自由手書き文字では、ある程度パターン揃ったものと、そうでないものに分け、サンプルを 3 カテゴリに分類した。

学習は、始めに少しの学習データで認識させて、認識



a) パソコン



b) EWS

図 9 手書き数字認識処理画面表示

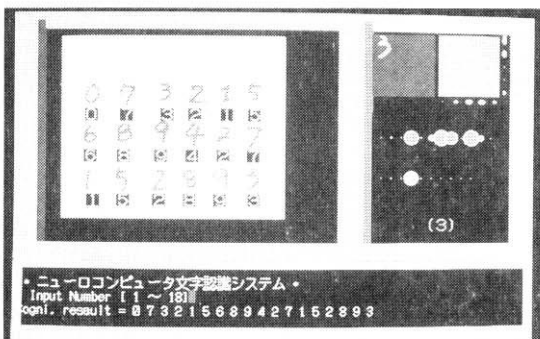
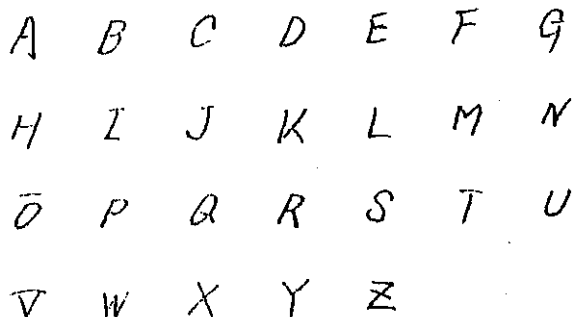


図 10 CCD カメラによる手書き数字認識システム例

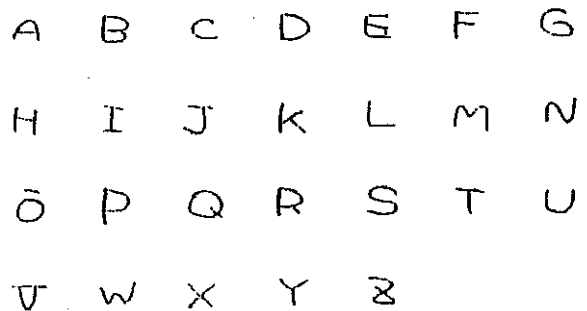
率の悪いデータを学習データとして増やして、改めて学習を行うということを数度繰り返した。最終的には、学習データ数として、300 個用い、残りの未学習データにより認識率を求めた。類似文字を効率良く認識できるように、学習回数は完全に収束させず 100 回とした。

学習は複数回学習データを変えて行った。認識結果を表 1 に示す。このように認識率は約 97% 以上が得られた。処理の表示の様子を図 9 に示す。結果のなかで、常用手書きとパターン揃った自由手書き文字では、学習データに、より多くの文字パターンを含ませるような最適な選択を行うことにより約 100% の認識率を得ることが可能であった。換言すると、このことは、ターゲットを限定することにより、かなり確度の高い認識システムを構築することが可能である。

本モデルに入力装置として CCD カメラを用い、手書き数字認識システムを構成した。処理にはパソコンに挿入したトランスピュータボードを使用した。カメラにより手書き数字が記入されたシートをイメージデータとして取り込み、文字は水平方向に並んでいるという前提条件により文字の切り出しを行い、認識処理を行った。処理画面を図 10 に示す。切り出した文字が 72×72 の入



a) パターン 1



b) パターン 2

図 11 手書き英大文字パターン

表2 手書き英大文字認識処理結果

		全文字数 (A~Z×数)	
サンプル数		1560	(26×60)
net 1	認識数 学習データ	514	: 98.8% (26×20)
	未学習データ 認識数	934	: 89.8% (26×40)
	全認識率	1448	: 92.8%
net 2	認識数 学習データ	505	: 97.1% (26×20)
	未学習データ 認識数	963	: 92.6% (26×40)
	全認識率	1468	: 94.1%

力領域に対して小さかったが、良好な認識結果が得られた。

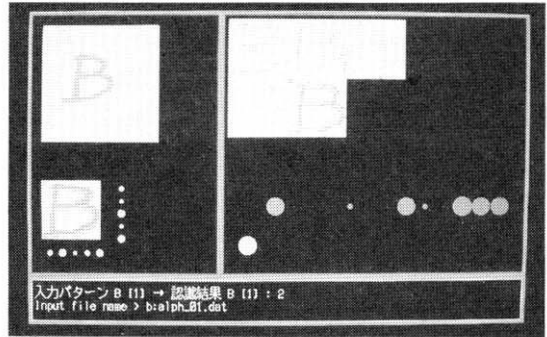
4.2 手書き英大文字認識処理

手書き英大文字データとして、常用手書き文字を用いた。サンプル数は 1560 (26×60) 文字用い、そのうち、520(26×20)文字を学習データとして、1040(26×40)文字に対して認識試験を行った。その一部を図 11 に示す。学習回数は数字と同様に 100 回に限定して行った。結果例を表 2 に、表示画面を図 12 に示す。このうち net 1 は学習データを適当に選定したもので、この学習データから、比較的形が類似したものを除き、認識結果の悪いものを加えて学習データを構成したものが net 2 であり、未学習データの認識率は 92.6% が得られた。このように学習データを最適化していくとさらに認識率の向上は望められると思われる。また、学習データ中の誤認があるが、これは、形が規則に反したもの、大きく傾斜したもの等によるためである。

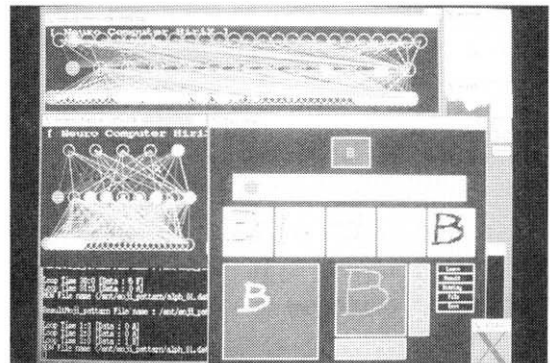
5. まとめ

複数のネットワークを階層的に構成し、手書き文字認識を行うネットワークシミュレータを作成し、手書き数字約 97%、手書き英大文字 92.6% の認識結果が得られた。また、文字種や形状を限定する等、ある程度の制約を設けることにより、100% 近い認識結果が得られることがわかった。

処理時間に関しては、パソコンで機能別に分割処理を



a) パソコン



b) EWS



c) パソコン A~Z 表示

図12 書き英大文字認識処理画面表示

行っており、数字・英大文字では異なるが、特徴データへの出力、認識ネットワークの学習共に約 1~2 時間を要した。EWS では単一のプログラム構成でありパソコンとの単純な比較はできないが、数時間~数十時間を要した。学習の高速化は今後の課題ではあるが、手書き数字と手書き英大文字がほぼ同じシステムで動作するよう

に、学習データを替えることにより、種々のパターン認識システムに対応できる可能性を有していると言える。

今後は、学習方法や処理系の最適化により、学習や認識処理の高速化や認識率の向上の検討、また、数字、英文字さらに手書きシンボル等を統括した処理系への展開と入力機構を含めたシステム構築を検討していきたい。

引用文献

- 1) 朱 小燕, 山内康一郎, 神保孝志, 梅野正義:「階層的ニューラルネットワークを用いた手書き文字認識」, 電子情報通信学会論文誌 D-II Vol.J73-D-II No.1 pp.54-61 1990.1
- 2) 波 通隆, 高橋裕之, 長尾信一, 齊藤 整:「手書き文字認識に関する研究—伝票自動読み取りシステムへの応用—」, 北海道立工業試験場報告, No.289 pp 111-124 (1990)
- 3) 山田博三, 齊藤泰一, 山本和彦:「線密度イコライゼーション—相関法のための非線形正規化法—」, 信学論 (D), Vol.J67-D, No.11, pp.1379-1383 (1984)
- 4) 鈴木 智:「細線化アルゴリズムの高速化に関する考察」, 情報処理学会論文誌, Vol.29 No.10, pp.925-932 (1988)
- 5) D.E. ラメルハート, J.L. マクレランド, PDP リサーチグループ 甘利俊一監訳:「PDPモデル認知科学とニューロン回路網の探索」, 産業図書
- 6) 「ニューラル・ネットをパターン認識, 信号処理, 知識処理に使う」, 日経エレクトロニクス pp.115-124 1987,8,10 (no.427)
- 7) SPIDER マニュアル
- 8) 菊池豊彦:「入門ニューロコンピュータ」, オーム社