

ネオコグニトロンの原理を用いた手書き数字認識

Handwritten Numeral Recognition by the Algorithm of the Neocognitron

— マイクロコンピュータによるシステム —

— An Experimental System Using a Microcomputer —

福島邦彦* 三宅誠* 伊藤崇之* 河野隆志**

Kunihiko Fukushima Sei Miyake

Takayuki Ito

Takashi Kouno

* NHK放送技術研究所

** 早稲田大学理工学部

NHK Science and Technical Res. Labs.

Waseda University

Suggested by the mechanism of the brain, the first author previously proposed an algorithm for pattern recognition called the "Neocognitron". It has an ability to recognize patterns, without being affected by distortions in shape or by shifts in the position of the input pattern. Using this algorithm, we have made a program for micro-computer which can recognize handwritten numerals. One numeral can be recognized in about 8 to 35 seconds by our system. This shows that the amount of calculation is not so large, and that we can build a system which works with reasonable speed if we make a special hardware for this purpose.

1. まえがき

従来のパターン認識は入力パターンの変形や位置ずれの影響を避けるために、まず入力パターンの位置や大きさの正規化を行なった後に特徴抽出や識別を行なう方式が多かった。これに対して、筆者らは先に、生物の視覚神経系をヒントにして、高度のパターン認識能力と学習能力を持つシステム“ネオコグニトロン”を開発した⁽¹⁾。この方式は、入力パターンの変形、位置ずれ、ノイズなどに強いパターン認識能力を示す。生物の神経系をヒントにしているため、その反応特性は人間に似ており、人間が似ていると感じるのはネオコグニトロンも似ていると判断する。しかも学習能力を持っているので、あらかじめ学習させておけば、どのようなパターンでも認識させることができる。

我々は既に、ネオコグニトロンによる手書き数字認識システムをミニコンピュータで実現したが⁽²⁾、今回はネオコグニトロンの演算がマイクロコン

ピュータでも実現可能であることを示すとともに、その演算量がどの程度かを一般の技術者に直感的にわかってもらうため、既に広く普及しているマイクロコンピュータPC-9801mを用いてシステムを構成した。プログラムは、Fortranを使用し、出来るだけ高速に動作することを主眼に作成した。なお、このシステムがマイクロコンピュータで実現可能であるということは、専用のハードウェアを用いれば、実用になる速度で働くシステムを作れることを示している。

2. ネオコグニトロンの構造と動作

システムは、実際にはソフトウェアで構成されているのであるが、説明を容易にするために、アナログ型細胞によって構成された多層回路を想定して演算アルゴリズムを説明する。

ネオコグニトロンは図1に示すように、S細胞と呼ばれる細胞の層U₁と、C細胞と呼ばれる細胞の層U₂とを組み合わせた2層の回路を基本単位とし、

この基本単位を縦続的に接続した構造をもっている。S細胞は可変入力結合を持ち、学習により特徴抽出機能を身につけ、ある一種類の特徴にだけ選択的に反応するようになっていく。次のC細胞は同一特徴を抽出するS細胞の中で受容野の位置が少し異なる一群の細胞から興奮性の結合を受け取っており、入力側のS細胞が1個でも出力を出せば、C細胞も出力を出すので、特徴の位置ずれに対して影響を受けにくくなっている。

さて、入力パターンが入力層 U_0 に与えられると、入力層に近い層の各細胞は、それぞれの受け持ちの小さい領域で入力パターンを観測し、局所的な特徴を抽出する。その次の層の細胞は、1段前の層の細胞が抽出した特徴を組み合わせた、より大局的な特徴を抽出する。この特徴の抽出と統合の各過程で、特徴相互間の相対的な位置ずれも少しずつ許容されていくので、最終層のC細胞の反応は入力パターンの位置ずれだけでなく、変形に対しても影響を受けなくなる。なお、最終的な認識結果は、最終段のどのC細胞が出力を出したかによって決定される。

U_s, U_c の各細胞層は、同一特徴を抽出する細胞が面状に並んだ“細胞面”によって構成されている。図1において太線で描いた個々の四角形が1つの細胞面を表わし、細線の長方形が1つ

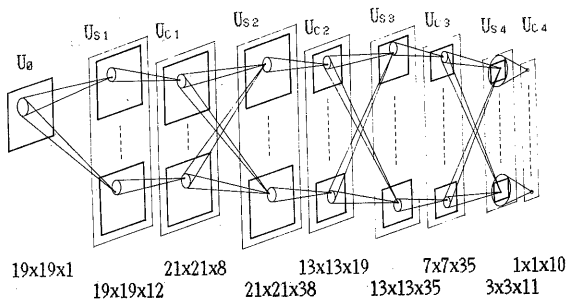


図1：各層の細胞間の結合状態

の細胞層を表わす。図の下に記した数字は各細胞層の細胞数を示す。各層の細胞は縦続的に接続され、個々の細胞の受容野は、後段の細胞ほど大きくなっていき、それにつれて各細胞面内に含まれる細胞の密度は減少する。最終段の U_{c4} 層では10個のC細胞が0から9までの数字パターンと1対1に対応している。

数式で表わすと、第 l 段の k 番目の細胞面内のS細胞の出力は、

$$u_{s_l}(n, k) =$$

$$r_l \cdot \varphi \left[\frac{\sum_{\kappa=1}^{K_{c_{l-1}}} \sum_{\nu \in A_l} a_l(\nu, \kappa) \cdot u_{c_{l-1}}(n+\nu, \kappa)}{1 + \frac{r_l}{1+r_l} \cdot b_l(k) \cdot v_{c_{l-1}}(n)} - 1 \right] \quad k=1, 2, \dots, K_{s_l} \quad \dots(1)$$

但し $\varphi[x] = \max(x, 0)$

で与えられる。ここに n はその細胞の受容野の位置を表わす2次元座標であり、 $u_{c_{l-1}}(n, \kappa)$ はその前段の κ 番目の細胞面内のC細胞の出力を表わす。

式(1)及び次に記す式(2)において、 $l=1$ の場合には、 $u_{c_{l-1}}(n, \kappa)$ は入力層 U_0 の細胞の反応 $u_0(n)$ を表わし、 $K_{c_{l-1}}=1$ であると解釈する。 a_l と b_l は、それぞれ興奮性及び抑制性の可変入力結合の強度である。 r_l は、抑制性入力の効き目を制御する正の定数であり、この値が大きくなると、S細胞が特定の1つの特徴にだけ選択的に反応する傾向が強まる。

この $u_{s_l}(k, n)$ に抑制性信号を送る細胞 $v_{c_{l-1}}(n)$ は、 $u_{s_l}(k, n)$ が受け取るのと同じ細胞群から信号を受け取り、その2乗平均値に等しい出力を出す。

$$v_{c_{l-1}}(n) =$$

$$\sqrt{\sum_{\kappa=1}^{K_{c_{l-1}}} \sum_{\nu \in A_l} c_{l-1}(\nu) \cdot u_{c_{l-1}}(\kappa, n+\nu)^2} \quad \dots(2)$$

ここに $c_{l-1}(\nu)$ は固定入力結合の強度を表わしている。

一方、C細胞の出力は、

$$uc_i(k, n) =$$

$$\psi \left[\sum_{k=1}^{K_{s_i}} j_{i,k}(x, k) \sum_{\nu \in D_i} d_{i,\nu} \cdot us_i(x, n + \nu) \right] \dots (3)$$

$$k=1, 2, \dots; K_{c_i}$$

但し $\psi[x] = \phi[x] / (\alpha_i + \phi[x])$

ここに、 α_i はC細胞の入出力特性の飽和の程度を定める正の定数である。 $d_{i,\nu}$ はS細胞からC細胞への興奮性固定結合の強度を表わす。

$j_{i,k}(x, k)$ は U_{s_i} 層の k 番目の細胞面から U_{c_i} 層の k 番目の細胞面に至る結合が存在する場合に1をとり、存在しない場合に0をとる。

3. 学習

各S細胞の可変入力結合の値は学習によって変化し、ネオコグニトロンは学習パターンを認識する能力を獲得していく。

各S細胞の可変入力結合の値は、次のように強化される。

まず、学習させるべきS細胞面を1つ指定し、その細胞面に学習させるパターンを入力層に呈示すると同時に代表となるべきS細胞の位置を指示する。この代表細胞の可変入力結合は、その結合を通して送り込まれている信号の強度に比例した値だけ増加する。その結果、代表細胞はその時点の入力刺激に最もうまく反応する性質を獲得していく。その細胞面内の、代表以外のS細胞の入力結合は、代表細胞とまったく同一の強化を受ける⁽²⁾。

実際の学習に用いた学習パターンを図2～4に示す。代表細胞の位置は、図3では、○印の場所であるが、その他の図では、各パターンの中心である。各細胞面に対し、1～4個の変形した学習パターンを呈示しているが、これによって、変形した特徴でも抽出できる能力を与えるためである。

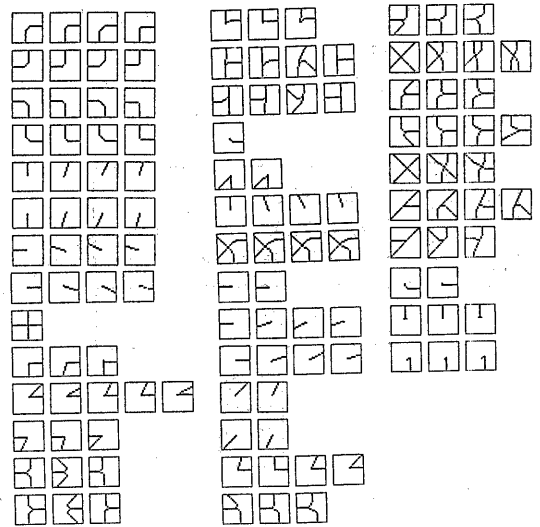


図2：U_{s2}層の細胞面の学習に用いた学習パターン

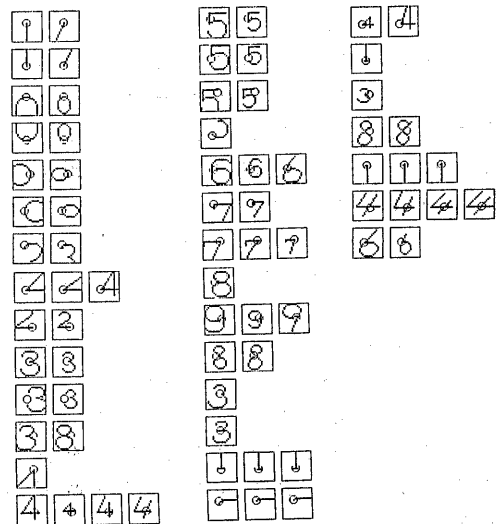


図3：U_{s3}層の細胞面の学習に用いた学習パターン（“○”は中心座標）

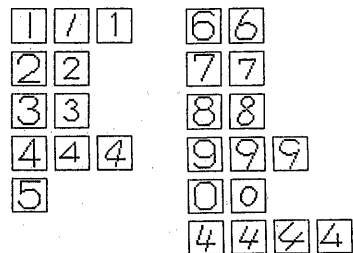


図4：U_{s4}層の細胞面の学習に用いた学習パターン

4. マイクロコンピュータによる 認識実験

今回の実験に用いたハードウェア・システムの概要を図5に示す。

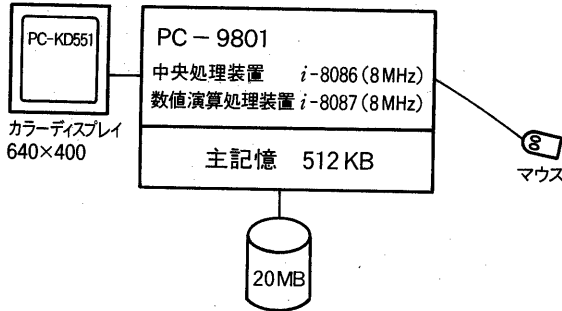


図5：ハードウェア・システム

4.1 認識時間の短縮

まず、認識時間の短縮について述べる。

認識の演算において大部分の時間を要しているのは回路内S細胞の出力を求める式(1)の分子の計算である。ちなみに、式(1)の演算に用いる配列が最大の大きさになるのは $l=2$ の場合で、その大きさは次の通りである。

$$\begin{aligned} u_{s_i}(n, k) &\cdots 21 \times 21 \times 38 \\ u_{c_i-1}(n, k) &\cdots 21 \times 21 \times 8 \\ a_i(l, k, k) &\cdots 5 \times 5 \times 256 \\ b_i(k) &\cdots \cdots \cdots 38 \end{aligned}$$

そこで、プログラミングに際してはこの式(1)の分子の計算に重点を置き、以下のような手法を用いて認識時間の短縮を図った。

(1) 配列の分割

CPU8086は広いメモリ空間をサポートするために、メモリ空間を64KBごとにセグメント化している。そのため、セグメントサイズ以上の大きな配列をアクセスするとオーバーヘッドが大きくなる。S細胞出力の入っている配列 $u_{s_i}(n, k)$ のように、その大きさがセグメントサイズ(64KB)以上の大きな配列は、1つの配列が64KB以下にな

るようにいくつかに分けて(図6)、オーバーヘッドを小さくした。この手法により、認識時間を半分にできた。

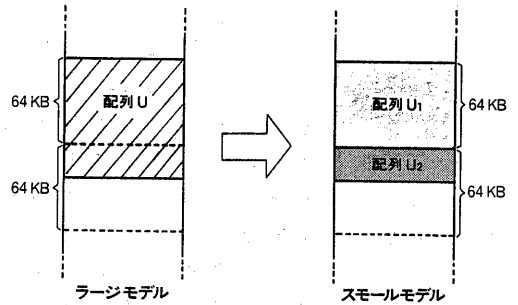


図6：配列の分割

(2) 結合係数データの圧縮

結合係数 $a_i(l, k, k)$ は0のものが多い。そこで、0かどうかを判定して、対応する乗算をスキップすることによって演算時間を短縮できる。しかしながら、0の判断にも時間がかかるので、この判断をしなくてもよいように値0の係数を除いた配列を作り、各係数と細胞との対応は別の配列にポインタとして格納した(図7)。また、配列のアドレス計算を簡単にするため、 $a_i(l, k, k)$ 及び $u_{s_i}(n, k)$ の中の2次元変数 l 及び n を、1次元の変数に置き換えて演算を行なった。これらの手法により、単純に0の判断を行なって演算する場合に比べて、20%ほど認識時間を短縮できた。

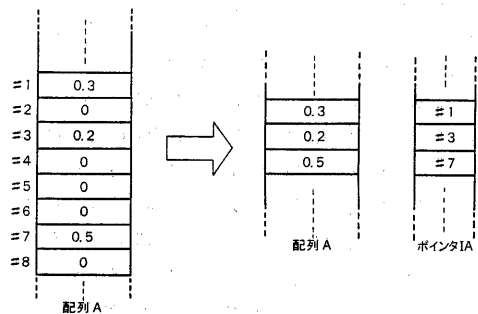


図7：ポインタ参照による結合係数データの圧縮

コンポリューションの計算には次の2つが考えられる。

- (a) S細胞を1つずつ選び、その出力を順番に計算していく。
- (b) S細胞への入力を累積していくバッファをS細胞の個数だけ用意する。次に、C細胞を1つ選び、そのC細胞から送り出される信号を各S細胞のバッファに加算していく。C細胞を1個ずつ順番に選び、このような操作を各C細胞について繰り返すことによって、最終的にS細胞の出力が定まる。

これらの手法を図にしたのが図8(a)、(b)である。

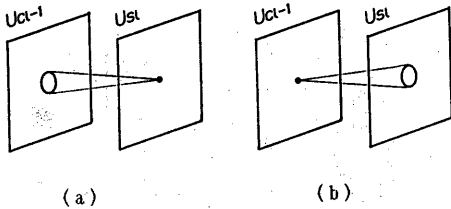


図8：コンポリューションの計算方法

ところで、 U_{c1-1} の値には0のものが多いため、 U_{c1} のゼロスキップを行うには、(b)の方法の方が効果的である。この場合、図9(a)に示すように U_{s1} の計算をする時に U_{c1-1} 面の周辺でそれに対応するS細胞が存在しないことがある。そこで、図9(a)のように U_{s1} 面の境界をはみ出した部分に対して、 U_{c1-1} 面からのデータを送り出さないようにする方法と、図9(b)のように配列 u_{s1} の周辺にダミーの配列を追加することにより、周辺部に対しても中心部と同じ手続きで演算を行なうようにする方法が考えられる。しかし、 m 及び n を1次元変数に展開して演算を行なう場合には、図9(a)の方法では周辺部での処理が複雑になり、演算時間がかえって長くなるので、図9(b)の方法を用いた。

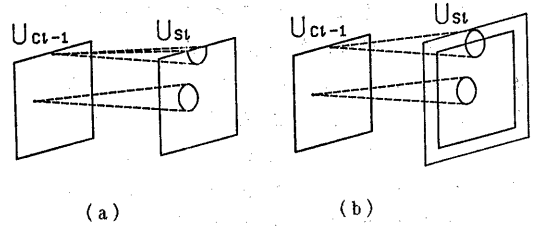


図9：周辺処理

(3) アセンブラ・レベルの最適化

このプログラム全体で一番時間がかかっている部分は、S層の計算である。この部分は、Fortranのアセンブラリストをもとにして、アセンブラルーチンを作成し、次のような最適化を行った。

- ① 不必要な命令(数値演算プロセッサ8087の初期化命令など)は削除する。
- ② 変数やループ・カウンタは出来るだけレジスタに割り付ける。
- ③ ループ内で定数として扱える数値は数値演算プロセッサ8087のスタックに積むようにして、ロードによる時間のロスを防ぐ。
- ④ 配列のアドレス計算などで同じ計算をしている部分は1つにまとめる。
- ⑤ 平方根などの関数はインライン展開する。
- ⑥ 各命令の実行時間を計算し、CPU8086との並列処理時間が出来るだけ長くなるように数値演算プロセッサ8087の命令を移動する。

この最適化により認識時間を40%ほど短縮できた。

4.2 実験結果

パターンへの入力にはマウスを使う。認識を終了すると、画面上には図10~12のような結果が現われる。画面左端は入力パターンを、中間が各 U_c 層の細胞の出力を、右端が最終認識結果を示している。

図10は、変形された手書きパターン“9”に対する回路内の反応を示している。最終層 U_{c4} では細胞“9”が反応しており、正しく認識できたことを示している。また、図11のように位置がずれた変形パターンが与えられても、正しく認識できることがわかる。

図12はノイズがある“4”の変形パターンに対する回路内の反応であり、この認識方式がノイズと変形に強いことを示している。

認識時間は入力パターンによるが、表示時間を含めて8~35秒程度である。

5. あとがき

マイクロコンピュータを用いて手書き数字認識のシミュレーション・システムを完成した。認識時間は表示時間を含めて8~35秒程度であり、リアルタイム・シミュレーションに耐えうると考えられる。なお、マイクロコンピュータでもこの程度の演算が可能であるということは、専用のハードウェアを用いれば、実用になる速度で働くシステムを作れることを示している。

ところで、パターン認識の究極的目標は、人間と同じように認識できることである。ネオコグニトロンのようなバイオサイバネティクスの研究はこの目標を達成する大きな可能性を秘めていると言えよう。

<文献>

- (1) K. Fukushima, S. Miyake: "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position", Pattern Recognition, 15, 6, pp. 455-469 (1982)
- (2) 福島邦彦, 三宅誠, 伊藤崇之: ネオコグニトロンの原理を用いた数字パターン認識, 信学論(D), J66-D, 2, pp. 206-213 (1983)

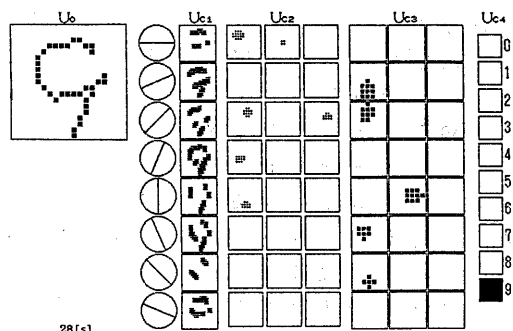


図10: 変形パターン“9”に対する回路内の反応

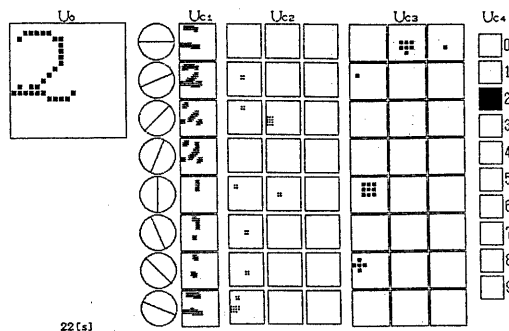


図11: 位置がずれた変形パターン“2”に対する回路内の反応

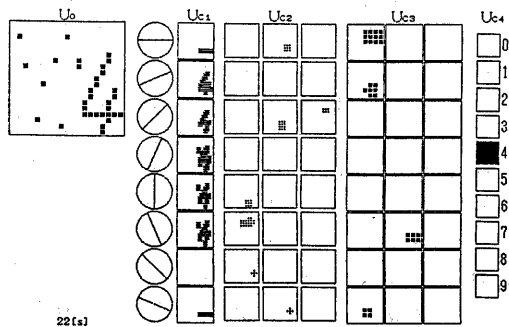


図12: ノイズがある“4”の変形パターンに対する回路内の反応