

ニューラルネットワークを用いた文字認識の一検討

西正明* · 河本敏志**

An Examination of Character Recognition using Neural Networks

Masaaki NISHI* and Satoshi Kawamoto**

(Received December 1, 1997)

キーワード：ニューラルネットワーク、バックプロパゲーション、パタン認識

1 諸言

脳の情報処理様式を解明しようと盛んに研究が行われている [1]。脳の情報処理様式を真似たものにニューラルネットワークがあり、さまざまなモデルが提案されている。手書き文字認識などのパタン認識では、階層型ニューラルネットワークを誤差逆伝播法 (error back-propagation) で学習するのが有効である。誤差逆伝播法は出力層で出力すべき情報として教師信号を与え、出力と教師信号との差に基づきニューラルネットワークのシナプス結合を徐々に更新して誤差を小さくしていく方法である [2]。

本論文では、イメージキャナで読み取った文字を誤差逆伝播法で学習して認識するシミュレーションシステム”IM-READnr” [3] を用いて、どの程度の認識ができるのかを評価したので報告する。

2 ニューラルネットワークの構成と学習アルゴリズム

本論文で扱うニューラルネットワークの構成と学習アルゴリズムを示す。

2.1 ニューラルネットワーク構成

本論文では、シミュレーションシステム”IM-READnr”を用いる。このシステムで使用しているニューラルネットワークの構成を図1に示す。図1は3層の階層型ニューラルネットワークであり、入力層で576ユニット、中間層で40ユニット、出力層で30ユニット設けている。

* 山口大学教育学部

** 東洋ハイブリット株式会社

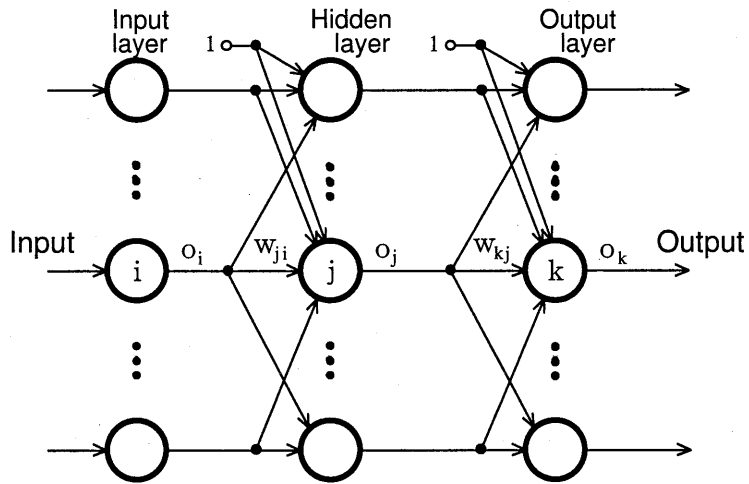


図 1: ニューラルネットワークの構成

2.2 学習アルゴリズム

図 1 に示したネットワーク構成で、 j 番目のユニットの出力 o_j は内部状態を s_j とすれば次式で示される。

$$o_j = f[s_j] = \frac{1}{1 + \exp(-s_j)} \quad (1)$$

$$s_j = \sum_i w_{ji} o_i \quad (2)$$

ここで、 w_{ji} は結合荷重、 o_i は前の層の i 番目ユニット出力である。結合荷重の修正量はあるパターンに対する誤差関数を E とすれば、学習係数を α とし次式で求められる。

$$\Delta w_{ji} = -\alpha \frac{\partial E}{\partial w_{ji}} \quad (3)$$

$$E = \frac{1}{2} \sum_k \{d_k - o_k\}^2 \quad (4)$$

ここで、 o_k は出力層の k 番目のユニット出力、 d_k はその出力に対する教師信号である。学習は学習パターンを与える毎に行うが、出力層での出力があらかじめ決められたユニットだけ 0.5 以上で、他のユニットでは 0.5 未満であれば重みの修正はしない。そうでない場合には、式 (3) に示される誤差の重みについての微分値を求めて与えたパターン毎に重みを逐次修正する。全ての学習パターンに対して正しく出力するようになれば、学習完了である。

学習は式 (3) に基づいて厳密に実行しようとするれば、重み修正を無限回繰り返さなければならなくなる。学習係数 α を大きくすれば一回の重み修正量は大きくできるが、振動する可能性が高くなる。そのため、学習係数の値は振動が起こらない範囲でできるだけ大きくする

が必要になる。本論文では、学習係数を大きくし、なおかつ重みが振動するのを防ぐために、一時刻前の重み修正量を慣性項として利用するモーメント法を使用する。モーメント法では、一時刻前の重み修正量を $\Delta w_{ji}(n-1)$ 、一時刻前の重み修正量の取り入れる程度を決定するモーメント係数を β とすれば、最終的な重み修正量 $\Delta w_{ji}(n)$ は次式で示される。

$$\Delta w_{ji}(n) = -\alpha \frac{\partial E}{\partial w_{ji}} + \beta \Delta w_{ji}(n-1) \quad (5)$$

誤差の曲面が細長い峡谷を形作っているような場合、峡谷の両側の壁面は急であるため、峡谷をまたぐような重みの変化が繰り返されて、峡谷の底の流れに沿って下るような学習は実現し難い。これを防ぐためには、学習係数を小さくして重み修正量を小刻みにすることが考えられるが、学習の収束速度は遅くなってしまう。このような場合にモーメント法を使用すると、重み値の振動を防ぎつつ実効的に重み修正量を大きくすることができる。

3 文字認識

本論文では、イメージスキャナで読み取った文字を3層ニューラルネットワークのシミュレーションシステム”Im-READnr”でどの程度認識ができるのかを評価する。イメージスキャナにはEPSON社製GT6000を用い、コンピュータにはNEC社製PC9801VX、PC9801RAなどを使用する。

3.1 イメージスキャナを用いる場合

使用する文字パターンはアルファベット26種類と一、+、?、!で合計30種類である。これらの文字パターンはシミュレーションシステムの入出力に対応させるために、 24×24 ドットで作成する。まず、 24×24 の0と1のビットパターンを作成する。ビットパターンの一部を図2に示す。これをイメージスキャナで読み込むために必要となる文字パターンに変換して紙面に印刷する。印刷の一部を図3に示す。図3に示すように各文字パターンの大きさは $7\text{mm} \times 7\text{mm}$ であり、外側を枠で囲むことにする。これは、文字データを作成する際に同時にノイズをランダムに付加できるようにしており、図4に示すように外側に枠が無いと文字パターンが分断されてしまう可能性があると考えたためである。”Im-READnr”の仕様上、分断された文字は2文字とみなされることがある。これを防ぐためにノイズが付加されることの無い枠を外側に付加した。

イメージスキャナでの読み取りには付属のユーティリティを用いた。読み取り時の設定はモノクロ、単純二値、解像度120DPI、ズーム100%とし、読み取ったデータはV-RAM形式と呼ばれるデータ形式でファイル保存した。”Im-READnr”はV-RAM形式のままでは扱えないので、さらにシミュレーションシステム付属のユーティリティを用いて”Im-READnr”用原稿ファイルに変換する。このとき、V-RAM形式上の文字パターンの大きさと位置はある程度自動補正される。こうして作成した原稿ファイルを用いて学習、認識のシミュレーションを行った。式(5)における学習係数 α とモーメント係数 β の値はそれぞれ次のように設定した。

学習係数 $\alpha = 0.25$

モーメント係数 $\beta = 0.5$

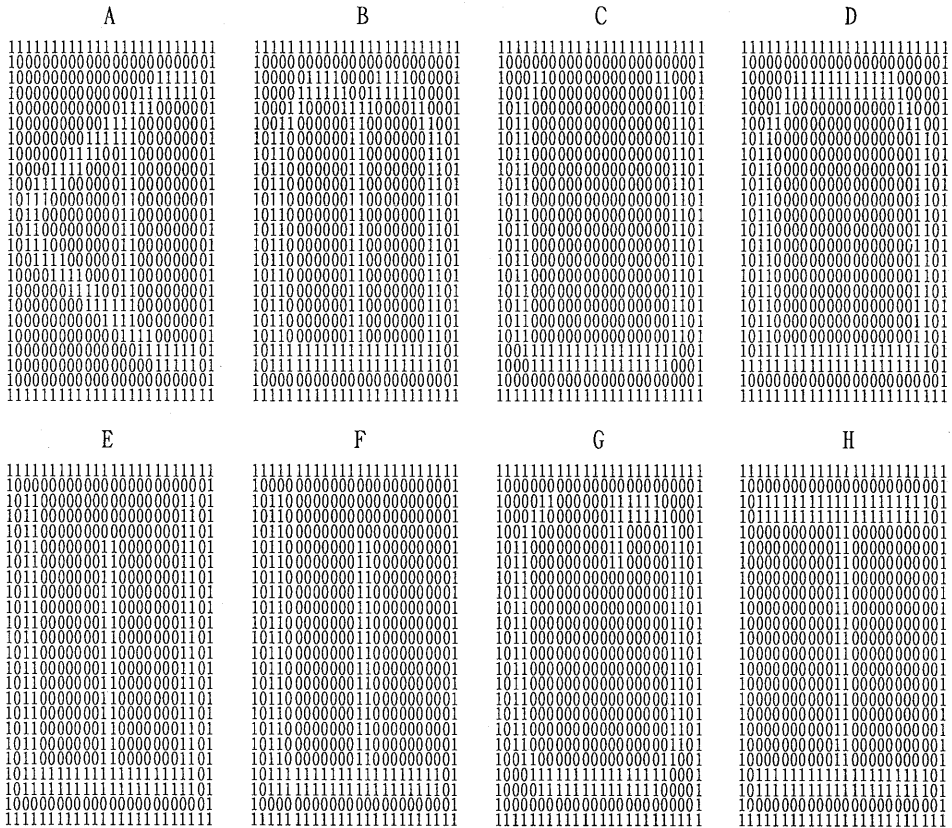


図 2: 24 × 24 の文字ビットパターン

以下にシミュレーション結果を述べる。学習文字が 15 種類 (A~O) と 30 種類全ての二通りの学習データセットを用意した。15 種類の文字学習では 200 個の文字でも学習が完了した。しかし、30 種類の文字学習では 150 個以上の文字数になると学習が完了しないことがあった。例えば、コンピュータに PC9801VX、PC9801RA、PC-H98Smodel8 を用いて 30 種 200 個の文字学習をそれぞれ 48 時間実行させたが、いずれも学習は完了しなかった。そのため、200 個の文字まで学習が完了できた 15 種類の文字学習の場合で評価検討することにする。

図 2 に示した各文字のビットパターン間のハミング距離を調べた。そのうち、文字 A、B、C、D、E について図 5 に示す。ハミング距離とは、ふたつの文字パターン間で対応するドット位置の値が異なる箇所の総和である。評価する認識文字として、他の文字とハミング距離が最も大きい文字と小さい文字を選択することにする。図 5 からわかるように、他の文字とハミング距離が最も大きい文字として A、逆に小さい文字として C を選択することにした。

学習文字数が 15 個、90 個、200 個の 3 通りの場合について、それぞれランダムにノイズを付加して文字認識を行った。文字 A についての認識率を図 6、文字 C についての認識率

を図7にそれぞれ示す。図6、図7において横軸のハミング距離は認識文字に付加したノイズ量を示す。図6から文字Aについては比較的良好に認識が実現されていることがわかる。また、学習文字数が多いと付加するノイズ量が多くなるにつれ認識率が急激に低下していることがわかる。これは学習文字数が多いと他の文字とのハミング距離が小さくなっているためと考えられる。図7からは、文字Cは最大でも50%の認識率であまりよく認識が行われていないことがわかる。

この実験から、イメージスキャナを用いた文字読み取りでは、あまり高い文字認識率は達成されることがわかった。そこで、本論文ではイメージスキャナを用いないで学習認識する場合の認識率と文字パターンを囲む枠の影響、イメージスキャナ自体の読み取り状態を評価検討する。

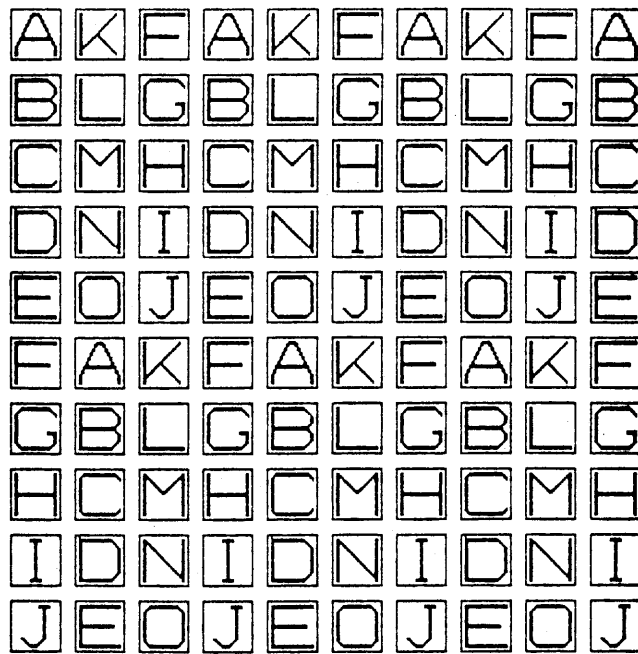


図 3: イメージスキャナ読み取り用文字パターン

3.2 イメージスキャナを用いない場合

紙面に文字パターンを印刷してイメージスキャナで読み取ることを行わずに入力データを直接シミュレーションシステムに与え、ニューラルネットワークシミュレーションシステム”Im-READnr”のシミュレータ本体の文字認識能力を評価検討する。入力データは図2に示した0と1のビットパタンのデータに数値処理を行ってV-RAM形式に変換する。これを、第3.1節と同様に、シミュレーションシステム付属のユーティリティを用いて”Im-READnr”用原稿ファイルに変換して学習、認識のシミュレーションを行う。

学習文字15種200個で学習させた結果を用いて、文字Aと文字Cについてノイズをランダムに付加して認識させた結果を図8に示す。図8から、文字Aと文字Cは共に認識率

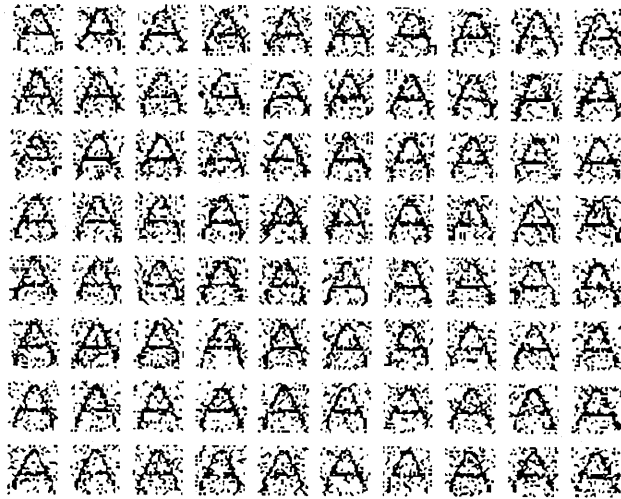


図 4: ノイズを付加したイメージスキャナ読み取り用枠なし文字パターン

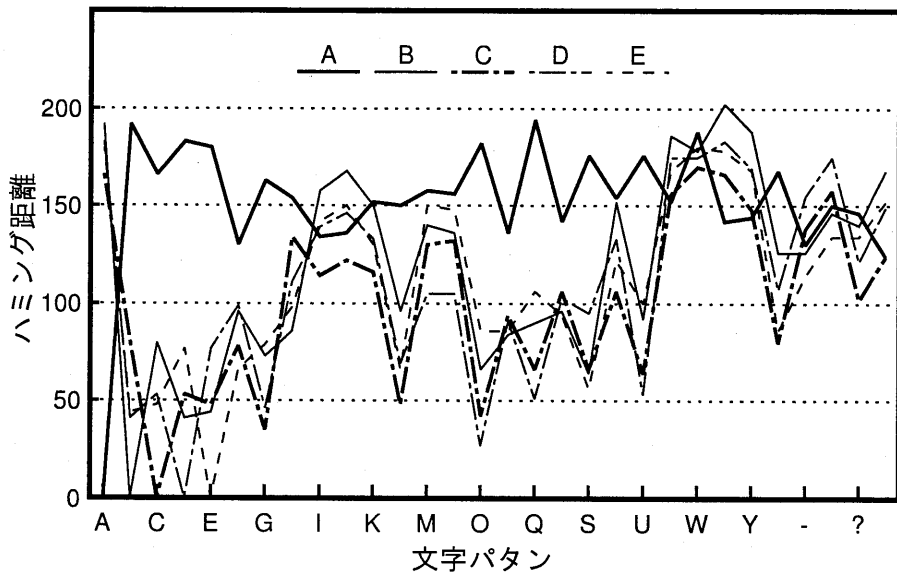


図 5: 文字パターン間のハミング距離

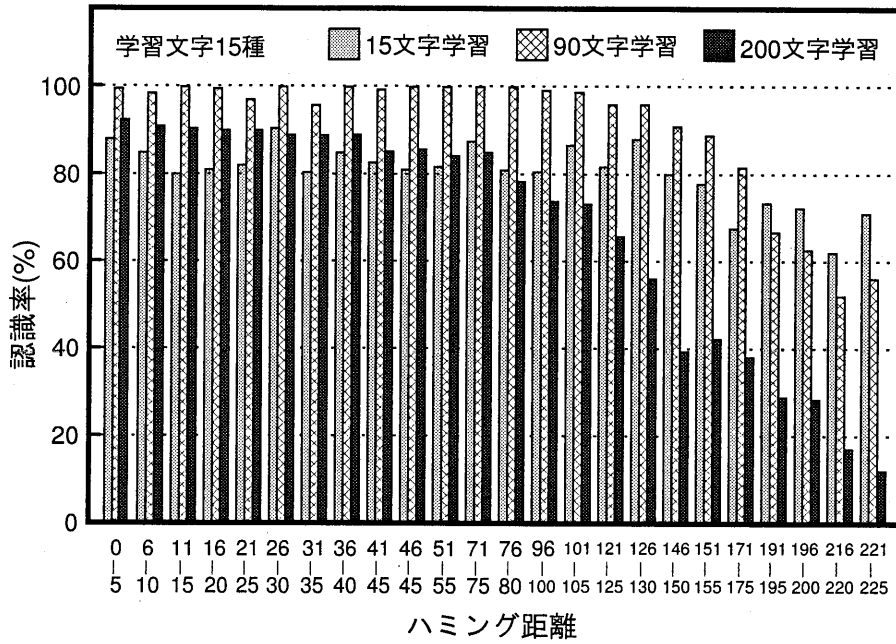


図 6: イメージスキャナ読み込み時の文字 A の認識率

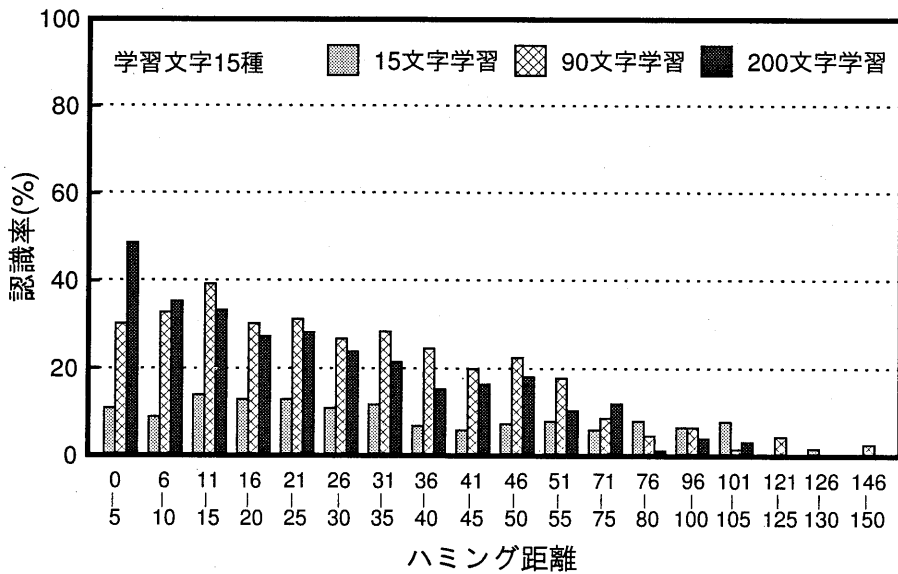


図 7: イメージスキャナ読み込み時の文字 C の認識率

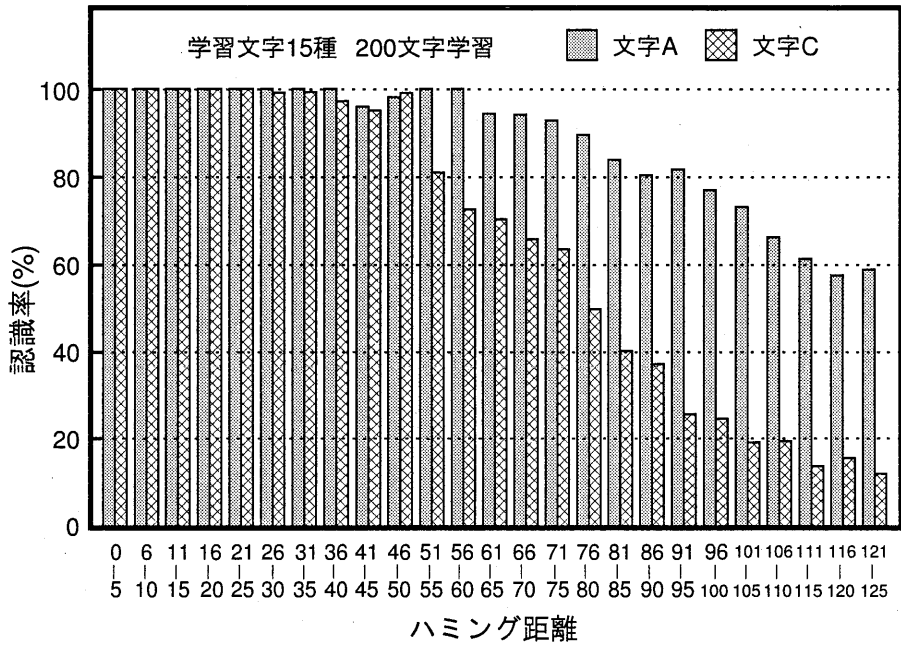


図 8: 文字データ直接入力時の文字 A、C の認識率

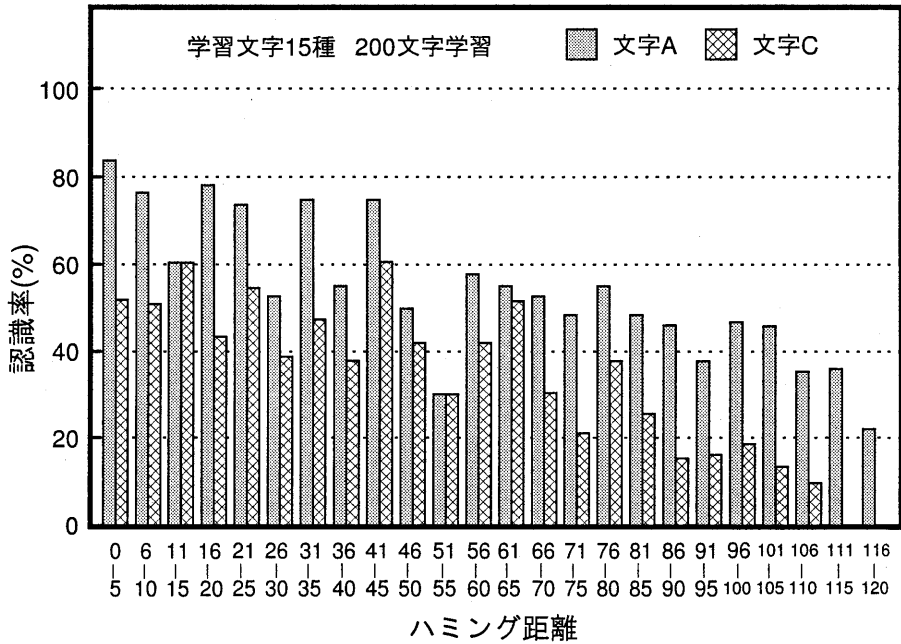


図 9: イメージスキャナ枠無し文字読み込み時の文字 A、C の認識率

が十分高いことがわかる。したがって、ニューラルネットワークシミュレーションシステム”Im-READnr”のシミュレータ本体の文字認識能力は十分高いことがわかった。

3.3 文字枠無しでイメージスキャナを用いる場合

第3.2節に述べたようにシミュレータ本体の文字認識能力は十分高いにもかかわらず、イメージスキャナを用いた文字認識率が低い。この原因として、ノイズを付加することによって文字パターンが枠とつながってしまっていることが考えられる。例えば、文字Cにおいて、文字の右側が枠と接触すれば、文字Oとの識別が困難になると予想される。そこで、各文字パターンを枠なしにして第3.1節と同様にイメージスキャナを用いた文字認識を行った。

学習文字15種200個で学習させて、ノイズを付加して認識させたときの文字Aと文字Cの認識率を図9に示す。文字Aについては図6と図9から枠が無いと認識率が10~50%小さくなっていることがわかる。文字Cについては図7と図9から枠が無いと認識率が10~50%大きくなっているが、図8に示されたほど認識率は高くなっていないことがわかる。このことから文字パターンを囲む枠の有無は個々の文字の認識率には影響があるが、イメージスキャナを用いることによる認識率低下の主たる要因ではないと考えられる。

3.4 イメージスキャナ自体のノイズ

イメージスキャナを用いたときの文字認識率が低い原因として、イメージスキャナで文字パターンを読み込むときに意図しない大きなノイズが入り込んでいる可能性がある。そこで、イメージスキャナで文字パターンを読み込むときに入り込むノイズ量と、これが文字認識率に与える影響を評価検討する。まず、イメージスキャナの読み取り文字データであるV-RAM形式のデータを調べることにする。V-RAM形式データを0と1のビットパターン文字データに変換する。その一部を図10に示す。図10からわかるように、元々の文字パターンは24×24ドットであったにもかかわらず、36×36ドットになっている。これはイメージスキャナの読み取り精度と文字パターンの大きさがうまく整合していなかったためと思われる。次に、図2に示した元々の文字パターンと図10の文字パターンとのハミング距離を調べる。このためには両方の文字パターンの大きさを合わせる必要がある。図2に示した文字パターンについて、配列値の奇数行を2行分に、奇数列を2列分にそれぞれ同じ値で拡張して、全体を1.5倍の大きさに拡大することにする。こうして得た文字パターン間のハミング距離を1.5の自乗で割った値を、24×24ドットの文字パターンにおけるハミング距離として扱うことにする。

イメージスキャナで読み込んだ15種類(A~O)の5組分の文字データ75個と、元々の文字データを上述のように加工して求めたハミング距離は平均で48であった。したがって、イメージスキャナを用いて文字パターンを読み込んだ場合の認識率は、イメージスキャナを用いないで文字パターンを直接入力するときのノイズ付加量がハミング距離で48だけ多い場合に相当すると考えられる。学習文字15種200文字としたときの認識文字AとCについて、イメージスキャナを用いる場合と用いない場合の比較を図11に示す。図11では、イメージスキャナを用いない場合は図8のハミング距離が51以上の部分を示す。イメージスキャナを用いる場合は図6と図7をハミング距離で50だけシフトして図11に示す。図11より認識文字がAについては、イメージスキャナを用いないで直接入力する場合とイメージスキャナを用

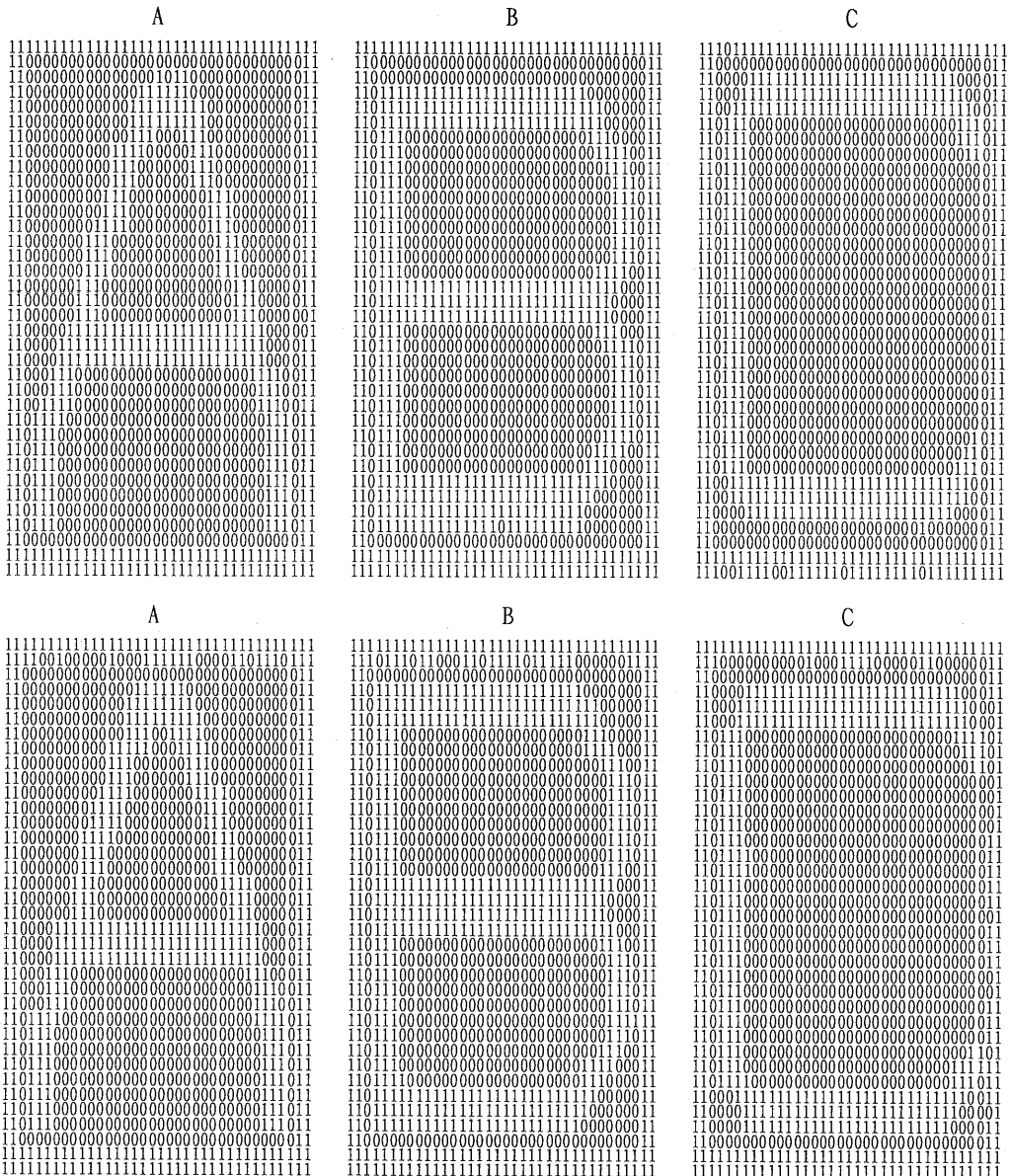


図 10: イメージスキャナの読み取り文字ビットパターン

いる場合はよく一致していることがわかる。しかし、認識文字 C の場合はあまりよく合っていない。これは図 10 からわかるようにイメージスキャナで読み込んだ文字データは同じ文字種でも同じ文字データになっていない。また、図 5 からわかるように、D、E、G、L、O などの文字は文字 C のハミング距離 50 の範囲に入ってしまう。このように文字 C についてはイメージスキャナによるノイズ付加以外にも影響を受ける要因があるためと思われる。

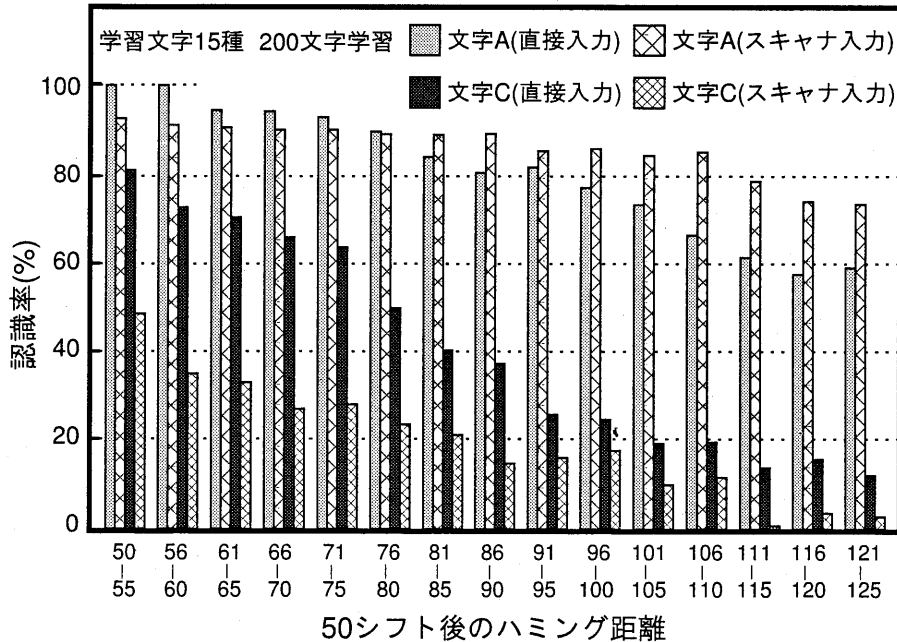


図 11: 文字データ直接入力とイメージスキャナ入力時の文字 A、C の認識率の比較

3.5 学習処理時間

本論文で用いたニューラルネットワークシミュレーションシステム”IM-READnr”の学習処理時間を図 12 に示す。図 12 は文字 1 種類当たり 5 個の文字を学習したときの処理時間である。図 12 から、学習文字種が増えると学習処理時間は学習文字種数に比例する以上に増加する傾向があることがわかる。

4 結言

本論文では、イメージスキャナで読み取った文字を誤差逆伝播法で学習して認識するニューラルネットワークシミュレーションシステム”IM-READnr”を用いて、どの程度の認識ができるのかを評価した。使用したニューラルネットワークは 3 層の階層型ニューラルネットワークであり、入力層で 576 ユニット、中間層で 40 ユニット、出力層で 30 ユニットである。文字は 30 種 200 個を用意したが、学習文字が 150 個以上では学習が完了しないことがあった。

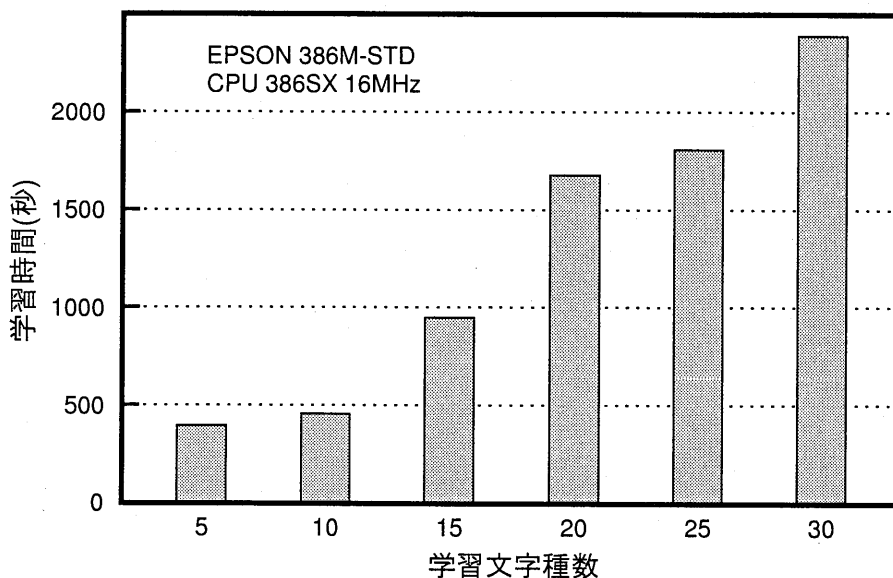


図 12: 学習処理時間

認識率は文字によってばらつきがあつて50~100%であり、あまり良い結果ではなかつた。そこで、イメージスキャナを用いなくて、0と1のビットパターン文字データを直接入力にしてシミュレーションしてみた。その結果、10%程度のノイズを付加してもほぼ100%の文字認識が達成できることが確認された。したがって、イメージスキャナを用いた場合に認識率が低いのは、シミュレーションシステムのシミュレーション本体の精度が低いというより、入力部すなわちイメージスキャナの読みとり精度とそのデータ変換部の精度に問題があると云える。

今後の課題は、イメージスキャナで読み込んだ文字データをニューラルネットワークシミュレーションシステムの入力に正しく対応するようにすることである。そのためには、イメージスキャナの読み取り精度を適切に調整すること、文字の大きさと位置ずれに対する処理を的確に行なうことが必要である。また、本論文では学習用の入力データにはノイズを付加していない文字データを用いたが、認識率を向上させるためにはノイズを付加した文字データも用いるべきと考える。

参考文献

- [1] 松本元, 大津展之, “ニューロコンピューティング”, 培風館(1992).
- [2] D.E. ラメルハート, J.L. マクレランド, “PDP モデル—認知科学とニューロン回路網の探索”, 産業図書(1988).
- [3] 前田民雄, “ニューロシミュレーションによる文字認識”, 山海堂(1990).