

入力領域適応型ニューラルネットワーク —変換に対して耐性のあるニューラルネットワーク—

麻 生 川 稔[†]

分類問題を扱うことのできる多くのニューラルネットワークが提案されているが、これらはすべて入力データが適当に切り出され正規化されていることを前提としている。実際の応用では、入力データが適当に切り出し、正規化することが可能でないために、誤認識の原因となる。

本論文では、affine 変換が重畠した入力パターンに対し、認識時に最適な affine 変換を弛緩法を用いて 1 つ推定し、補正された入力データについてニューラルネットワークを用い認識する入力領域適応型ニューラルネットワークを提案した。入力領域適応型ニューラルネットワークはパターンを補正する入力領域と、ニューラルネットワークで構成されている。入力領域にはパターンのサンプリングセルが空間的な制約の基に配置されている。変形したパターンが入力領域に提示された場合には、出力層での 2 乗誤差を減少させるために、入力領域の各サンプリングセルは変形を補正し、同時にサンプリングセル間の制約を満たすようにパターン上を動く。移動したサンプリングセルによって得られた入力により、新たに出力が決まる。この手続きを繰り返すことにより最適な affine 変換を 1 つ推定する。また、この手続きは必ず収束し、発散することがないことを証明した。入力領域適応型ニューラルネットワークを手書き数字認識に適用した。階層型ニューラルネットワークで誤認識された 10.83% の手書き文字を正しく認識することができた。

Adaptive Input Field Neural Network Which can Recognize Transformed Patterns

MINORU ASOGAWA[†]

For machines to perform classification tasks, such as speech and character recognition, appropriate handling of deformed patterns is a key to achieving high recognition performance. The authors present a new type of classification system, an Adaptive Input Field Neural Network (AIFNN), which includes a simple pre-trained neural network and an elastic input field attached to an input layer. By using an iterative method, AIFNN can determine an optimal affine transformation for the elastic input field to compensate for deformations of original patterns. The convergence of the AIFNN algorithm is proven. AIFNN is applied to handwritten numeral recognition. Consequently, 10.83% of originally misclassified patterns are correctly categorized and total performance is improved, without modifying the neural network.

1. はじめに

分類問題を扱うことのできる多くのニューラルネットワークが提案されているが、これらはすべて入力データが適当に切り出され正規化されていることを前提としている。実際の応用では、入力データをつねに適切に切り出し、正規化することが可能ではないために誤認識の原因となる。たとえば、階層型ニューラルネットワークを文字認識等に応用したとき、入力パターンは長いベクトルとして扱われ、画像データで上下あるいは左右といった隣接関係は失われる。その結

果、平行移動したパターンはまったく別のベクトルとして表現される。平行移動したパターンは、入力ベクトルは本来のものと比較して、シフトしたものになるので、ニューラルネットワークの重みが入力ベクトルのシフトに対して耐性がないと、出力が異なってしまう可能性が高い[☆]。

このことを図 1 と図 2 を用いて説明する。図 1 の(a)にはカテゴリ A とカテゴリ B があり、パターン A_1 に対して affine 変換されたパターンの集合が多様体を形成することを示す。このことを用いると、カテゴリ A とカテゴリ B が affine 変換によって拡大され、

[†] 日本電気(株) C&C 研究所コンピュータシステム研究部
C&C Research Laboratories, NEC Corporation

[☆] もちろん、学習パターンは平行移動したものが含まれていると、平行移動に対して耐性のある認識が期待できる。

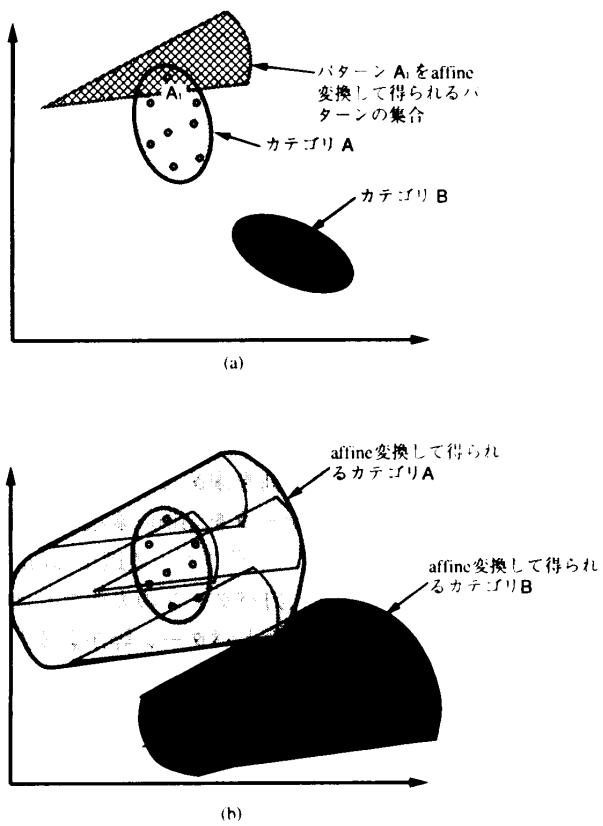


図 1 affine 変換とカテゴリの領域
Fig. 1 Affine transformation and categories.

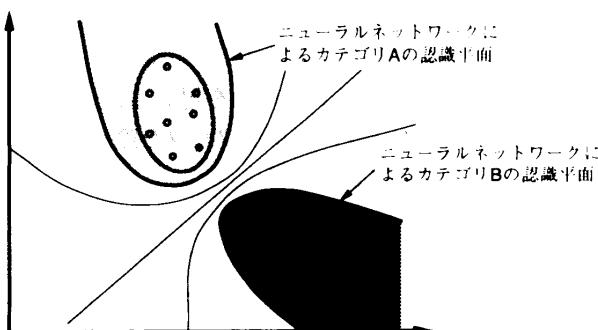


図 2 学習されたニューラルネットワークの認識面
Fig. 2 Learned segregation hyperplane.

図 1 の (b) のようになる。したがって、affine 変換に対して不变の認識を行うためには、図 1 の (b) に示す領域に基づいて行わなければならない。

図 2 には affine 変換されていないパターンで学習したニューラルネットワークの認識平面を示す。これは、学習データに affine 変換されたものが含まれていないので、図 1 の (a) のカテゴリ A とカテゴリ B とはほぼ同じような形を形成する。したがって、ニューラルネットワークの認識領域と、affine 変換された図 1 の (b) とが異なるために誤認識の原因となる。

認識率を向上させるために、入力データに対して適切な affine 変換を適用し変換後のデータを評価することによって高い認識率を達成できることが知られている^{1)~4)}。これらの方法は基本的にはノルムを距離としたテンプレートマッチング法である。しかしながら、適切な affine 変換を見つけることが難しいので affine 変換の線形近似を用いたり³⁾、パターンをスプライン関数で近似したりする¹⁾。

本論文で説明する入力領域適応型ニューラルネットワークは、学習済みのニューラルネットワークを用いて、最適な affine 変換行列を弛緩法を用いて求め、認識する方法であり、すべての affine 変換されたデータに対して認識が可能である。また、この弛緩法による手続きは必ず収束する証明を 4 章に示した。

次の章では、上記した従来方法について紹介し、問題点を明らかにする。

2. 従来方法

認識率を向上させるために、ニューラルネットワークを用いてすべての affine 変換されたデータを正しく認識するためには、

- (1) 学習データから可能なすべての affine 変換されたデータを作成し、学習に用いる。
- (2) 認識時に、入力パターンから可能なすべての affine 変換されたデータを作成し、認識に用いる。
- (3) 認識時に、入力パターンに最適な affine 変換行列を 1 つ推定し、補正したデータを認識に用いる。
- (4) affine 変換に対して不变の特徴量を用いる。
- (5) affine 変換に対して不变な局所的な特徴のトポロジを用いる。

等の方法が考えられる⁵⁾。

しかしながら、(1), (2) の方法では 1 つのデータから、可能なすべての affine 変換したデータを作成する必要がある。すべてのもっともらしい affine 変換を実現することは難しいし、また学習データが増えるために学習に時間がかかる等の問題点がある。

(3) は (2) の方法の改善方法であるが、最適な affine 変換を見つけるためには、入力パターンの正しいカテゴリとそのカテゴリに典型的なテンプレートパターン等が必要となる。認識する以前なので、正しいカテゴリはあらかじめ分かっていない、したがって入力パターンに適用すべきである最適な affine 変換行列を求めることはできない。本論文で提案する入力領域適応型ニューラルネットワークは、この方法に属するが、

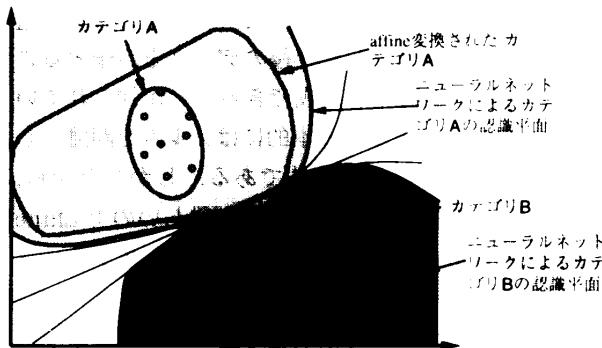


図3 affine変換データによる学習

Fig. 3 Learned segregation hyperplane with affine transformed data.

テンプレートではなくて学習済みのニューラルネットワークを用いる。

(4)についてはaffine変換に対して不变の特徴量をあらかじめ考えることは難しい。また、(5)についてはaffine変換のいくつかの自由度に対しては不变にできるが、すべてのaffine変換の自由度に対応することはできない。

以下、上記した従来方法について紹介し、問題点を明らかにする。

2.1 affine変換された学習データを作成する方法

この方法では、学習データから可能なすべてのaffine変換されたデータを作成し、新しく学習データとし、ニューラルネットワークに学習させる（図3参照）。この方法では大量の学習データができるので、学習に時間がかかる。

別のアプローチとして、Simardらが提案しているTangent Propがある³⁾。この方法では、入力データに対してaffine変換したものについても出力が変わらないことを学習させる。具体的には、入力データの各affineパラメータに対しての、偏微分係数(Tangent)を数値計算で近似する。その偏微分係数を入力に加えても、出力が変動しないように重みを学習する。この学習方法は、誤差逆伝搬法(バックワードプロパゲーション法)⁶⁾を変更することによって実現できる。しかしながら、この方法でも大量の学習データが生成されるため、学習に時間がかかる。

2.2 入力データをaffine変換し認識する方法

この方法では、学習には通常のデータを用い、認識時に入力データに対していくつかのaffine変換されたデータを作成し、ニューラルネットワークで認識する^{7),8)}（図4参照）。認識時に、入力データに対して可能なすべてのaffine変換されたデータを作成して、それらすべてについて認識処理を行う必要があるので、計算量が膨大となる。

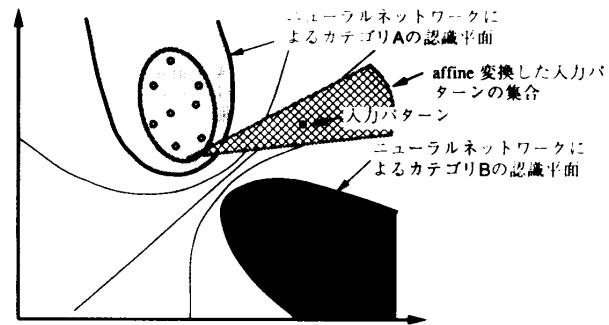


図4 認識時に入力データをaffine変換

Fig. 4 Applying an affine transformation at the recognition phase.

2.3 最適なaffine変換行列を求める方法

この方法は、2.2節の入力データをaffine変換し認識する方法の改善方法である^{1),2),4),9)~11)}。最適なaffine変換を見つけるためには、入力パターンのカテゴリとカテゴリに典型的なテンプレートパターン等が必要となるが、当然カテゴリはあらかじめ分かっていないので、典型的なテンプレートパターンも決めることができない等の問題がある。

この場合、距離尺度として入力パターンとテンプレートのノルムが用いられることが多い。しかしながら、ノルムを距離尺度とすると、ニューラルネットワークのように特徴に注目した認識ができないため、認識能力がニューラルネットワークに比べて劣る²⁾。本論文で提案する入力領域適応型ニューラルネットワークは、この方法に属するが、テンプレートではなくて学習済みのニューラルネットワークを用いる。以下に、入力領域適応型ニューラルネットワークを除く代表的な3つの方法について説明する。

2.3.1 Simardらの方法

Simardらは不完全ながらも、入力パターンをaffine変換した後に認識する方式を提案している²⁾。ここで、すべてのもともらしいaffine変換は計算量が膨大になるので、線形に予測することによって防いでいる。認識はテンプレートをaffine変換したものと、入力データをaffine変換したものとのノルムを用いる。

実験は郵便番号の認識で行っている。テンプレートパターン数は9709個で、認識実験パターン数は2007個である。実験の結果、誤認識率は表1のようになった。

比較実験のk-nearest neighborとSimardらの手法との違いは、Simardらの手法では入力パターンとテンプレートパターンに対して適切なaffine変換した後の距離を用いるが、k-nearest neighborでは補正されていない両パターンの距離を用いている。認識率が改善されていることに注目すると、入力データを適切に

表 1 Simard らの方法による認識結果
Table 1 Recognition results obtained by Simard²⁾.

Human	2.5%
本手法	2.6%
Neural Network	3.3%
k-nearest neighbor	5.9%

affine 変換することにより、認識率を向上させることができると結論できる。また、実験結果によると最小距離を構成するときにはさほど大きな変形は必要ないことが分かる。

本手法の問題点は、距離の尺度として入力パターンとテンプレートパターンのノルムを用いていることである。その結果、入力パターンのうちカテゴリの特徴を構成しない部分（たとえばノイズ）に対しても影響を受ける。また、ノルムを距離尺度として用いている k-nearest neighbor の結果よりニューラルネットワークの結果が優れていることから、ニューラルネットワーク学習によってカテゴリの特徴が抽出され、ノルムを距離尺度として用いるよりも高い確度で分離可能であることが分かる。したがって、認識率の向上のため、距離尺度にはノルムよりもニューラルネットワークに関連するものを用いるべきであることが分かる。

2.3.2 Bodenhausen の方法

Bodenhausen は音声認識に特化したものを提案している¹¹⁾。周波数軸方向には固定であるが、時間軸方向について適応可能なモデルである。具体的には、各周波数にサンプリングセルを 1 つずつ置く（学習の結果、複数個置くときもある）、各サンプリングセルは時間軸方向に対して移動可能で、サンプリングできる範囲を可変としている。サンプリングはセルの時間軸に対してガウスカーネルを用いている（図 5 参照）。学習は、ガウスカーネルの平均と分散とセルから出力セルに対する重みを、認識誤差が減少するように山登り法で修正する。分散が大きいときには、ガウスカーネルを 2 つのカーネルで近似し、セル数を動的に増加する。

この方法では、パターン強度が 0 でない部分の平均と分散を学習の対象とするので、パターン強度が 0 となる部分を学習の特徴とできない。つまり、認識時に、パターン強度が 0 であることを積極的に認識の手がかりとしていない。また特徴抽出は 2 層のネットワークで行われるために、複雑なカテゴリを学習できないことが問題点である。

2.3.3 Hinton らの方法

Hinton らの方法の方法では、手書き文字認識を対象としている。このモデルでは、文字を 8 点のスプライン曲線でモデル化している¹⁾。8 点のスプラインで

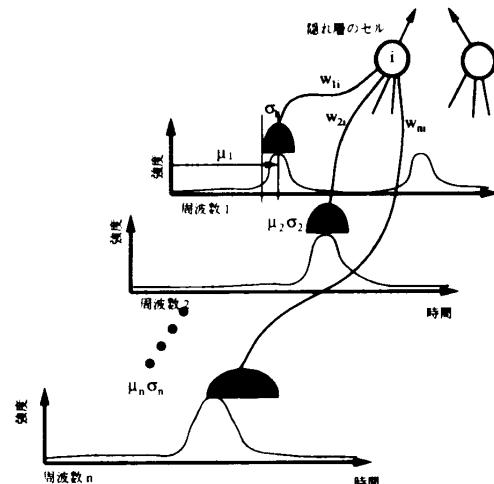


図 5 Bodenhausen の方法
Fig. 5 Bodenhausen's method.

決まる曲線と、入力データのノルムを認識に用いる。8 つの座標は、入力データとのノルムが小さくなるように山登り方を用いて移動させる。その結果として、8 つの点に対して最適な affine 変換行列を求める方法である。

Hinton らの距離尺度には、入力データと 8 点のスプライン曲線とのノルムを用いているが、単にノルムだけを認識の距離尺度に用いるには問題である。また、モデルとしてスプライン曲線を用いているので、線描イメージに対してのみ適応可能である。

2.3.4 Buhmann らの方法

Buhmann らの方法では、データベースの各画像より得られた Gabor フィルタを用い、入力データとのマッチングをダイナミック・リンクとピラミッド型マッチングの手法を用い、入力画像の各点とデータベース画像の各点の対応を求めている⁴⁾。

これから分かるように、この手続きには学習のフェーズではなく、階層型ニューラルネットワークでは可能であるところの画像中の特定の領域に注意を払うといったことができない。

2.4 affine 変換に対して不变の特徴量を用いる方法

affine 変換のすべてのパラメータに対して不变の特徴量は線形システムでは求めることができないので、その一部分を実現したものがいくつか提案されている。Li らは、認識物体の重心を求め、Hough 変換で求めた物体の端と重心との距離を極座標表示する。その後、最大距離への方向を使って規格化し、極座標表示された特徴量を RCE を使って認識を行っている¹²⁾。

この方法では、回転に対して汎化能力があるように思えるが、物体の重心が特徴ベクトルの根本的な要素になっているので、雑音によって重心が変化すると特

微ベクトルが異なってしまうために回転に対しても耐性がなくなる。また、剪断変形によって最大距離への方向も変化するので、その場合も正しく認識されない。さらに、回転すると同じとなるが異なるカテゴリに属するパターン(6と9等)の区別がない等の問題点がある。

2.5 局所的な特徴のトポロジを用いる方法

局所的な特徴のトポロジを用いる方法としてワイベルらが提案している TDNN (Time Delayed Neural Network)¹³⁾、追江らが提案している DNN (Dynamic Neural Network)¹⁴⁾と、Fukushima らが提案しているコグニトロン (Cognitron)¹⁵⁾がある。TDNN と DNN は音声認識に用いられ、時間的な伸縮を補正し認識するものである。しかしながら、この方法を、そのまま affine 変換に適用できない。また、DNN は DP マッチングをベースにしたもので、マッチングの評価をニューラルネットワークで行うものである。DNN は、従来のベクトル間のノルムを評価関数として用いる方法に比べて高い認識率を達成している¹⁴⁾。このことから、認識率向上のためには、評価関数として、ノルムよりもニューラルネットワークが適していることが分かる。

Cognitron は文字認識に用いられたモデルで、入力データの局所的な特徴をトポロジを考慮しつつ大局部的に統合し、特徴量とする。さらに得られた特徴についてまた同様の処理を繰り返すことによって、スケールや位置に不变の認識をする。しかしながら、機構的に回転や剪断変形に対しては、特徴として同様にとらえることができない。したがって、すべての affine 変換に適用するためには、他の手段が必要となる。

3. 入力領域適応型ニューラルネットワーク

入力領域適応型ニューラルネットワークは分類型問題のためのニューラルネットワークで、図 6 に示すように、入力領域と学習が完了したニューラルネットワークで構成されている（ここでは、階層型ニューラルネットワークを用いた例で説明する[☆]）。入力領域には入力パターンのサンプリングセルが空間的な制約のもとに配置されている。階層型ニューラルネットワークの学習は、たとえば誤差逆伝播法（バックワードプ

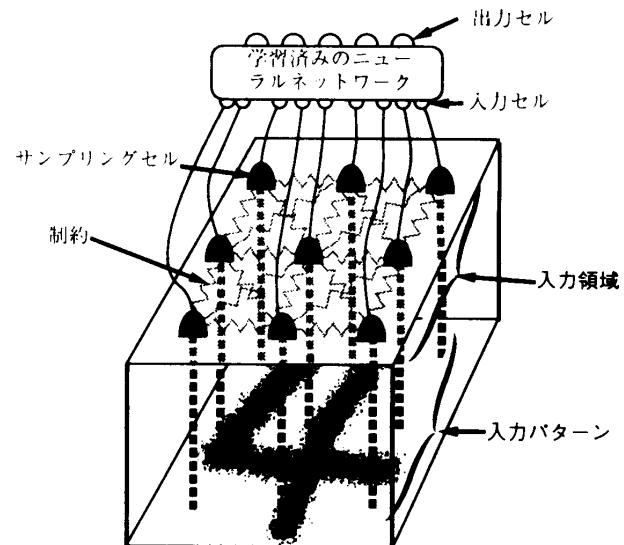


図 6 入力領域適応型ニューラルネットワークの構成

Fig. 6 Architecture of an adaptive input field neural network.

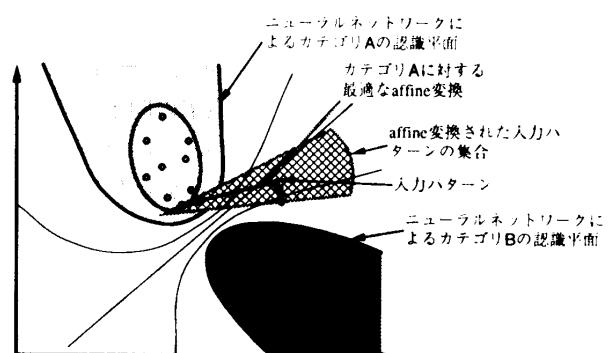


図 7 認識時に最適な affine 変換を推定

Fig. 7 Presume an optimal affine transformation at the recognition phase.

ロパゲーション法)⁶⁾で行う。入力領域適応型ニューラルネットワークの特徴は、認識時に入力パターンが与えられたときに各カテゴリに対して最適な affine 変換行列を求め、入力パターンを再解釈し、カテゴリに属するかを決める点である（図 7 参照）。

図 8 に入力領域適応型ニューラルネットワークの認識アルゴリズムを示す。認識カテゴリを決めるために用いる収束後の出力層での 2 乗誤差と入力領域の歪みの和は、限られた条件ではリヤブノフ関数であることから、認識アルゴリズムの収束性が証明できる。詳しくは、4 章の収束性の証明に示す。

図 8 のステップ 4 で示した入力領域のサンプリングセルの移動ベクトルは、2 乗誤差を小さくし出力が仮定正解カテゴリに近づくようにし、学習時に用いた格子系から affine 変換可能な格子系に近づくようにする。ここで、出力の 2 乗誤差を減少する方に作用するベクトルをデルタ力と呼び、入力領域のサンプリングセル

[☆] これは説明を簡単にするためで、出力に対する入力の偏微分値を計算できるニューラルネットワークならば、どのような方法でもかまわない。これは、解析的に偏微分値が計算できなくても、外乱をえて数値近似できるので、ほとんどすべてのニューラルネットワークが適応可能である。すなわち、ホップフィールド型やボルツマンマシン等も適用可能である。

ステップ 1: まずすべてのカテゴリから、1つカテゴリを選び
仮定正解カテゴリとする。

ステップ 2: 入力パターンをサンプリングセルの位置に従って
サンプリングしニューラルネットワークの入力とする。この入力を学習が完了したニューラルネットワークで処理することによって出力を得る。サンプリングセルの格子系の初期状態は、学習に用いた格子系と一致させる。サンプリングセルの位置が、入力パターンの格子点に一致しないときは、サンプリングした値は補間法を使って近似する。

ステップ 3: 得られた出力と仮定正解カテゴリの代表ベクトルと比較し、2乗誤差を求める。

ステップ 4: 2乗誤差を小さくし、かつサンプリングセルの格子系の歪みが小さくなるようなサンプリングセルの位置を求め、新しい格子系とする。

ステップ 5: サンプリングセルの格子系が収束していなかったら、ステップ 2 に戻る。

ステップ 6: 各仮定正解カテゴリに対して、収束後の出力層の2乗誤差とサンプリングセルの格子系の歪みの和を求め、最小の和を達成した仮定正解カテゴリを認識カテゴリとする。

図 8 入力領域適応型ニューラルネットワークの認識アルゴリズム
Fig. 8 Algorithm of an adaptive input field neural network.

格子の学習時に用いた格子系から affine 変換可能な格子系からの歪みを小さく保つよう作用するベクトルをアドレス力と呼ぶ。各サンプリングセルはデルタ力とアドレス力を合成した方向に移動させる。各サンプリングセルがデルタ力だけに従って移動すると、ニューラルネットワーク出力の2乗誤差が単調に減少し、入力パターンが仮定正解カテゴリのように補正される。各サンプリングセルがアドレス力だけに従って移動すると、移動後の格子系は学習に用いた格子系の affine 変換可能な格子系となる。

以下、デルタ力とアドレス力について詳しく説明する。

3.1 デルタ力

変形された入力パターンが入力領域に提示されたときは、出力層の各セルの出力パターンは学習時の教師信号と一致しないことが多い。入力領域適応型ニューラルネットワークでは、出力セルの1つを仮定正解カテゴリと仮定する[☆]。入力が学習パターンに似ていれば、出力層の仮定正解カテゴリに対応するセルの出力が完全な1.0となり、それ以外のセルの出力が0.0なるはずである^{☆☆}。変形が大きい場合は、どれか1つのセルの出力が1.0でそれ以外が0.0となるような状態にはならず、各セルが中間的な値をとる。したがつ

☆ もちろん、入力パターンがどのカテゴリにも対応しないことはありうる。

☆☆ 説明を簡便にするために、各出力セルがカテゴリに対応し、入力パターンがあるカテゴリに属するとき、対応するセルのみの出力が1.0で、それ以外が0.0となるような教師信号を仮定する。

て、仮定正解カテゴリベクトルと現在の出力層のセルの値に違があることによって、2乗誤差が0にはならない。この2乗誤差を0にするように、デルタ力は各サンプリングセルに作用する^{☆☆☆}。以下に、この各サンプリングセルのデルタ力の導出を示す。

いま、ある入力パターンを提示しニューラルネットワークの出力が決まったとする。ここで、仮定正解カテゴリを1つ決めると、2乗誤差 E^D は以下のように与えられる。

$$E^D = \frac{1}{2N} \sum_{i=1}^N (A_i - O_i)^2 \quad (1)$$

$$\left\{ \begin{array}{l} N : \text{出力層のセルの数} \\ A_i : \begin{cases} 1.0 & i \text{ が仮定正解カテゴリならば} \\ 0.0 & \text{上記以外} \end{cases} \\ O_i : \text{出力層の第 } i \text{ セルの出力} \end{array} \right.$$

バックワードプロパゲーション法を用いることによつて、入力層の第 j セルに対するデルタ値 δ_j を求めることができる。デルタ値 δ_j と出力層での2乗誤差 E^D には、以下に示す関係がある⁶⁾。

$$\delta_j = -\frac{\partial E^D}{\partial net_j} \quad (2)$$

ここで net_j は第 j 入力セルの入力値である。

したがって、 δ_j が正のときは、第 j 入力セルの入力値 net_j が増加すると、2乗誤差 E^D は減少する。つまり、出力はより仮定正解カテゴリベクトルに近づくことになる。このことから、出力をより仮定正解カテゴリに近づけるためには、第 j 入力セルの入力値 net_j を、以下に与えられるような Δnet_j だけ変更されればよい。

$$\Delta net_j = \eta \delta_j \quad (3)$$

ここで、 η は正の数。

この Δnet_j がデルタ力の基本である。このとき、第 j 入力セルの入力値を Δnet_j だけ増加し、出力を計算し、同じ手順で Δnet_j の計算を繰り返すと、第 j 入力セルの入力値は徐々に変化して、最終的にニューラルネットワークの入力は仮定正解カテゴリの標準的なパターンに似る^{16), 17)}。しかしながら、このようにして求められた入力は、現在の入力パターンの affine 変換で得られるとは限らない理想的なパターンを仮定しているので、入力データの分類問題には使えない。

☆☆☆ ここで、デルタ力を導出するのに2乗誤差を用いたが、これは通常ニューラルネットワークの学習に2乗誤差を用いるためであり、別の評価関数を学習に用いた場合は、それをデルタ力の導出に用いるべきである。

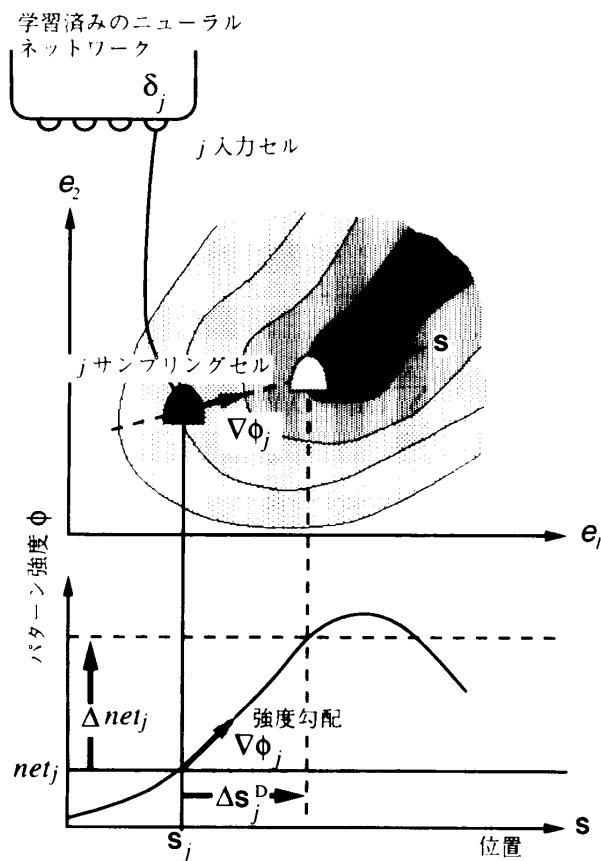


図9 $\nabla\phi_j$ と Δnet_j の関係
Fig. 9 Relation between $\nabla\phi_j$ and Δnet_j .

特に、第 j 入力セルに Δnet_j を加えることによって、入力値が負となったり非常に大きな値になる等の不都合が発生する可能性がある。

入力領域適応型ニューラルネットワークでは、第 j 入力セルの入力値を Δnet_j だけ増加するために、第 j 入力セルに 1 対 1 に接続しているサンプリングセルをパターン上で移動する。サンプリングセルが移動しなければならない位置 $s_j + \Delta s_j^D$ は、パターンの強度勾配 $\nabla\phi_j$ と Δnet_j を用いて、図 9 に示すように計算できる。図 9 では、 Δnet_j が正で入力領域においての第 j サンプリングセルの位置が s_j であったとしている。第 j サンプリングセルが移動しなければならない位置 $s_j + \Delta s_j^D$ では、位置 s_j に比較してパターン強度が Δnet_j 大きい。

s_j 近傍のパターンの強度勾配 $\nabla\phi_j$ は、たとえば、4 近傍のパターン強度を元にラクランジエの線形一次補間法で計算する。

以上をまとめて、デルタ力は、 Δnet_j とパターンの強度勾配 $\nabla\phi_j$ を用いて以下のように計算できる。

$$\Delta net_j \frac{1}{|\nabla\phi_j|} \frac{\nabla\phi_j}{|\nabla\phi_j|}$$

上式では、パターンの強度勾配 $\nabla\phi_j \simeq \mathbf{0}$ のときは、デルタ力は無限大となる。しかしながら、この状態ではデルタ力はほぼ $\mathbf{0}$ となるべきである。これは、パターンの強度勾配がないために、その入力セル自身はどちらの方向に移動すべきか決定できないためである。

上記のことを考慮して、デルタ力 Δs_j^D は以下のように決定した。

$$\begin{aligned}\Delta s_j^D &\equiv \Delta net_j \frac{|\nabla\phi_j|}{\epsilon^2 + |\nabla\phi_j|^2} \frac{\nabla\phi_j}{|\nabla\phi_j|} \\ &= \eta \delta_j \frac{|\nabla\phi_j|}{\epsilon^2 + |\nabla\phi_j|^2} \frac{\nabla\phi_j}{|\nabla\phi_j|}\end{aligned}\quad (4)$$

ここで、 ϵ は適当に小さい正の実数である。 ϵ は実問題によって決める必要がある。詳しいことは、7 章の実問題への入力領域適応型ニューラルネットワークの適用で述べる。式(4)では、 $|\nabla\phi_j| = -\epsilon, +\epsilon$ で、各々極大極小 $-1/2\epsilon, 1/2\epsilon$ となる。 $|\nabla\phi_j|$ が ϵ よりも十分小さいときは、ほぼ $|\nabla\phi_j|$ に従うので、強度勾配 $\nabla\phi_j \simeq \mathbf{0}$ のとき、デルタ力 Δs_j^D は無限大にならず、 $1/2\epsilon$ によっておさえられる。また、 $|\nabla\phi_j|$ が ϵ よりも大きいときは、ほぼ $1/|\nabla\phi_j|$ に従って減少する。

3.2 アドレス力

入力領域の各サンプリングセル間には、学習時の格子系から affine 変換可能な範囲に保つように空間的な制約力を導入する。これに基づくのがアドレス力である。したがって、サンプリングセルの格子系が初期のそれと比較して affine 変換可能な範囲にある限り、アドレス力は $\mathbf{0}$ である。

最適な affine 変換行列は、解析的に求めることができない。本手法では、まず各サンプリングセルの位置より適切な affine 変換行列を最小 2 乗法を用いて推定し、その後に推定した affine 変換行列を元にして各サンプリングセルのあるべき位置を決めて、現位置との差をアドレス力とする。したがって、各サンプリングセルをアドレス力に沿って移動させれば、サンプリングセルの格子系は学習時の格子系から affine 変換可能な範囲に保たれる。本章では、現在の各サンプリングセルの位置より適切な affine 変換行列を推定する方法について述べ、アドレス力の導出を行う。

3.2.1 サンプリングセルの位置より affine 変換行列を求める

仮想的に最適なサンプリング格子系 S^N を考える。これは、学習時に用いたサンプリング格子系 S^O に affine 変換 $T(\cdot)$ を適用したものとなる。

$$S^N = T(S^O) \quad (5)$$

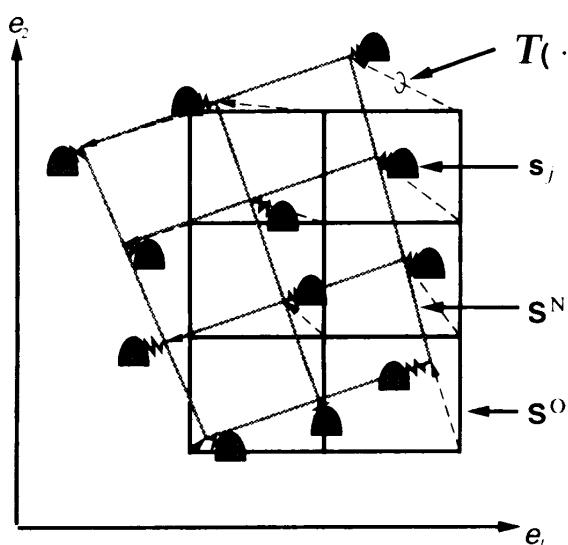


図 10 S^O と S^N の関係
Fig. 10 Relation between S^O and S^N .

このとき、仮想的なサンプリング格子系 S^N と現在の各サンプリングセルの位置 S との距離 $d(S^N, S)$ を以下のように定義する。

$$d(S^N, S) \equiv \sum_i \|s_i^N - s_i\|^2 \quad (6)$$

最適な affine 変換行列は、距離 $d(S^N, S)$ が最小となるように、最小 2 乗法を用いて求める（詳細な求め方は付録に示した）。すなわち、最適なサンプリング格子系 S^N は、初期のサンプリング格子系 S^O の affine 変換したものうち距離 $d(S^N, S)$ を最小とするものである。

3.2.2 affine 変換行列を用いてアドレス力を求める

affine 変換 $T(\cdot)$ を学習時に用いたサンプリング格子系を S^O に適用することによって、サンプリングセルが位置すべき格子系 S^N を決め、現在の各サンプリングセルの位置 S との距離 $d(S^N, S)$ を、歪エネルギー E^A に対応させる。すなわち、各サンプリングセルが格子系 S^N の各位置と自然長さ 0 バネ定数 k のバネで結ばれているモデルを考える^{*}。

$$\begin{aligned} E^A &\equiv \frac{1}{2} k \sum_i \|s_i^N - s_i\|^2 \\ &= \frac{1}{2} k d(S^N, S) \end{aligned} \quad (7)$$

アドレス力 Δs^A は歪エネルギー E^A の s_i に対する偏微分で定義する。

$$\Delta s_j^A \equiv -\mu \frac{\partial E^A}{\partial s_j} \quad (8)$$

ここで、 μ は正の定数である。

すなわち、このアドレス力 Δs^A に沿って各サンプリングセルを移動させると、歪エネルギー E^A が最小になり、すべてのサンプリングセルは格子系 S^N に揃う。つまり、サンプリングセルの格子系は学習時の格子系から affine 変換可能な範囲に保たれる。

3.3 サンプリングセルの新しい位置

サンプリングセルの新しい位置は、デルタ力とアドレス力で決まる合成功力を現在の位置に作用させることによって得る。第 j サンプリングセルの合成功力 Δs_j は

$$\Delta s_j \equiv \Delta s_j^D + \Delta s_j^A \quad (9)$$

である。この合成功力 Δs を各サンプリングセルに適用すると、ニューラルネットワークの出力が仮定正解カテゴリに近づき、かつサンプリングセルの格子系は学習時の格子系から affine 変換可能な範囲に保たれる。

図 8 のアルゴリズムに示すように、出力層での 2 乗誤差とサンプリングセルの格子系の歪みの和が収束しないければ、新しいサンプリングセルの格子系で、入力データをサンプリングしてニューラルネットワークの入力とし、デルタ力とアドレス力を再計算する。収束していたら、この手続きは終了して、和を求める。同様にして、他の仮定正解カテゴリについても和を求め、最小の和を達成した仮定正解カテゴリを認識カテゴリとする。

4. 収束性の証明

この章では、入力領域適応型ニューラルネットワークのアルゴリズムはつねに収束することを、リヤノフ関数の存在を示すことによって証明する。具体的には、出力層でのエラー E^D と入力領域の歪エネルギー E^A の和である認識コスト関数 $L (= E^D + E^A)$ がリヤノフ関数であることを示す。そのため、 L に下限値があり、かつ各サンプリングセルに合成功力 Δs_i を適用し移動させると L が単調に減少することを示す。

まず、出力層でのエラー E^D は、2 乗誤差であるので、 $E^D \geq 0$ である。また、入力領域の歪エネルギーもバネの弾性エネルギーであるため、 $E^A \geq 0$ である。したがって、認識コスト関数 L に下限値があることは明らかである。

次に、 L が各サンプリングセルに合成功力 Δs_i を適用することによって単調減少することを示す。まず、前準備としてデルタ力 Δs_j^A の近似を求める。式 (4)

* バネ定数 k をすべてのサンプリングセルに対して独立に決めることも可能である。これによって、変形が起きそうな部分に重みづけができる。

に示したように、デルタ力は

$$\Delta s_j^A \equiv \eta \delta_j \frac{|\nabla \phi_j|}{\epsilon^2 + |\nabla \phi_j|^2} \frac{\nabla \phi_j}{|\nabla \phi_j|}$$

により決まる。

特に、入力パターンの強度勾配が大きくないとき、つまり $|\nabla \phi_j| \ll \epsilon$ のときは、

$$\frac{\nabla \phi_j}{\epsilon^2 + |\nabla \phi_j|^2} \simeq \nabla \phi_j \quad (10)$$

の近似が成り立つ[☆]。式(10)と式(4)を用いるとデルタ力 Δs_j^A は、以下のように近似できる。

$$\begin{aligned} \Delta s_j^A &= \frac{\nabla \phi_j}{\epsilon^2 + |\nabla \phi_j|^2} \Delta net_j \\ &= \frac{\nabla \phi_j}{\epsilon^2 + |\nabla \phi_j|^2} \left(-\eta \frac{\partial E^D}{\partial net_j} \right) \\ &= -\eta \frac{\nabla \phi_j}{\epsilon^2 + |\nabla \phi_j|^2} \frac{\partial s_j}{\partial net_j} \frac{\partial E^D}{\partial s_j} \\ &\simeq -\eta \nabla \phi_j \frac{\partial s_j}{\partial net_j} \frac{\partial E^D}{\partial s_j} \\ &= -\eta \frac{\partial E^D}{\partial s_j} \end{aligned} \quad (11)$$

各サンプリングに合成功力 Δs_j を適用することによって生ずる、 L の時間変化は、以下のように表現できる。

$$\begin{aligned} \frac{d L}{d t} &= \frac{d E^D}{d t} + \frac{d E^A}{d t} \\ &= \sum_i \left\{ \frac{\partial E^D}{\partial s_i} \frac{d s_i}{d t} \right\} + \sum_i \left\{ \frac{\partial E^A}{\partial s_i} \frac{d s_i}{d t} \right\} \\ &= \sum_i \left\{ \left(\frac{\partial E^D}{\partial s_i} + \frac{\partial E^A}{\partial s_i} \right) \frac{d s_i}{d t} \right\} \end{aligned} \quad (12)$$

ここで、 $\frac{d s_i}{d t}$ は、サンプリングセル j の時間的な位置の変化であるので、合成功力 Δs_i に等しい。

$$\frac{d s_i}{d t} = \Delta s_i^D + \Delta s_i^A \quad (13)$$

したがって、式(11)と式(8)を用い、 $\eta = \mu$ とするとき、 L の時間微分は、

$$\begin{aligned} \frac{d L}{d t} &\simeq -\mu \sum_i \left(\frac{\partial E^D}{\partial s_i} + \frac{\partial E^A}{\partial s_i} \right)^2 \\ &\leq 0 \end{aligned} \quad (14)$$

となる。つまり、認識コスト関数 L は各サンプリングセルに合成功力を適用すると単調に減少することが示された。

ここで、 L に下限値が存在することと、(14)を用いて、認識コスト関数 L がリヤプノフ関数であること

[☆] 実験では、この近似が成り立つように入力データをあらかじめガウスフィルタではさし、 $|\nabla \phi_j|$ を小さくするように工夫している。

が示され、合成功力を繰り返し適用することによって、つねに平衡状態に収束することが証明された。

5. 文字認識実験による入力領域適応型ニューラルネットワークの有効性の確認

入力領域適応型ニューラルネットワークを用いて、手書き数字文字認識実験を行った。手書き文字は、 40×32 のメッシュで 2 値で取り込み、 5×5 のガウスフィルタで処理し、 14×18 とした。ニューラルネットワークは 3 層構造とした。入力層のセルの数は 252 個、中間層のセルの数は 20 個、出力層のセルの数は 10 個とした。ニューラルネットワークの学習中に、不要な重みを取り除くために、重みの簡単化¹⁸⁾を用いた学習を行った。また、シグモイド関数の微分が 0 となり、学習が停止することを避けるために、シグモイド関数の微分に小さな正の値を重畳した¹⁹⁾。学習データは 2500 個で、テスト用データは 2500 個である。学習は 180 回繰り返した。学習後の入力層から出力層の重みを解析したところ、5040 リンク中 2400 リンクの重みはほぼ 0 となっていた。これは、学習に重みの簡単化を行ったため、手書き文字のデータの端には文字が現われないことから、端の入力セルから中間層への重みが、ほぼ 0 になったためである。

学習後に得られたニューラルネットワークの性能を知るために、認識実験を 2500 個のテストパターンについて行った(図 11 参照)。回答カテゴリは、出力層セルの最大活性値を示すものとした。同時に、回答カテゴリのベクトルと出力ベクトルとの 2 乗誤差に基づき、認識不能のパターンを決めた。認識不能の閾値を大きくすると、誤認識率は低くなるが、認識不能パターンが多くなる。閾値を最小(0.0)としたときは、正解率が 95.20% (2380 個)、誤認識率が 4.80% (120 個)、認識不能率が 0.00% (0 個) となった。

ちなみに 2500 個の学習データに関しては、正解率

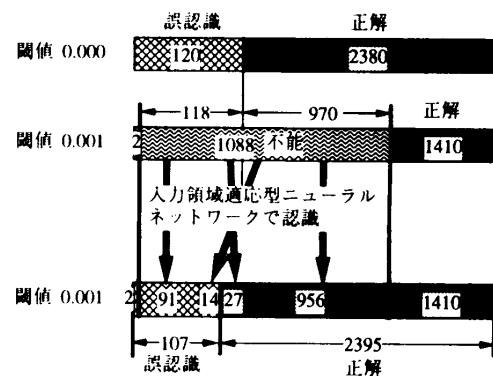


図 11 入力領域適応型ニューラルネットワークを用いた認識結果
Fig. 11 Handwritten numerals recognition result.

が 99.40% (2485 個), 誤認識率が 0.60% (15 個), 認識不能率が 0.00% (0 個) となった.

閾値を 0.001 としたとき, 正解率が 56.40% (1410 個), 誤認識率が 0.08% (2 個), 認識不能率が 43.52% (1088 個) となった.

この認識不能とした 1088 個について入力領域適応型ニューラルネットワークを用いて認識した. その結果, 983 個を正しく認識でき, 105 個を誤認識した. この内訳は, 閾値を 0.001 としたときに誤認識していた 118 個のうち, 入力領域適応型ニューラルネットワークを用いることにより 27 個が正しく認識された. また, 閾値を 0.001 としたときに正しく認識していた 970 個のうち, 入力領域適応型ニューラルネットワークを用いることにより 956 個が正しく認識された. その結果, 全体の認識結果は正解率が 95.800% (2395 個), 誤認識率が 4.200% (107 個), 認識不能率が 0.0% (0 個) となった. したがって, 全体としてテストデータ中で誤認識されていた 120 個のうち, 10.833% にあたる 13 個が正しく認識されて, 誤認識パターンを 4.800% (120 個) から 4.200% (107 個) に減らすことができた.

この実験では, 仮定正解カテゴリに対してサンプリングセルを移動する回数は, 最大 100 回までとした. 1 つの仮定正解カテゴリに対しての計算量は, 従来のニューラルネットワークの前向き伝播が 100 回分, 後ろ向き伝播が 100 回分, ニューラルネットワークの入力値と強度勾配を決めるためのラクランジエ補間を 252 サンプリングセルに対して 100 回必要となる. まとめると, 従来の 3000 倍程度の計算量が必要となる. しかしながら, 並列に計算できる部分も多いので, 実際には 3000 倍もの計算量は必要ない.

図 12 に, 手書き文字の 8 を示したときの, 各仮定正解カテゴリについて, 出力層でのエラー E^D と入力領域の歪エネルギー E^A の和 L の時間変化を示した. このパターンは, ニューラルネットワークで 2 と認識されていたが, 入力領域適応型ニューラルネットワークで仮定正解カテゴリを 8 としたときの方が 2 としたときよりも, エラー E^D と歪エネルギー E^A の和 L が小さい値に収束し, その結果正しく認識された.

図 13~20 に, 手書き文字の 8 を示したときに仮定正解カテゴリが 8 とした場合と 2 とした場合と 1 とした場合の様子を示す. 図中, 右側の 2 つのパターンの下側が入力パターンそのものである. 各位置でのパターン強度を黒丸の大きさで示している. 左側の大きめの格子状の図は, サンプリングセルの格子系を示す. 各黒丸の大きさはサンプリングセルの入力値を表す.隣接サンプリングセルの関係は太めの灰色線で示す.

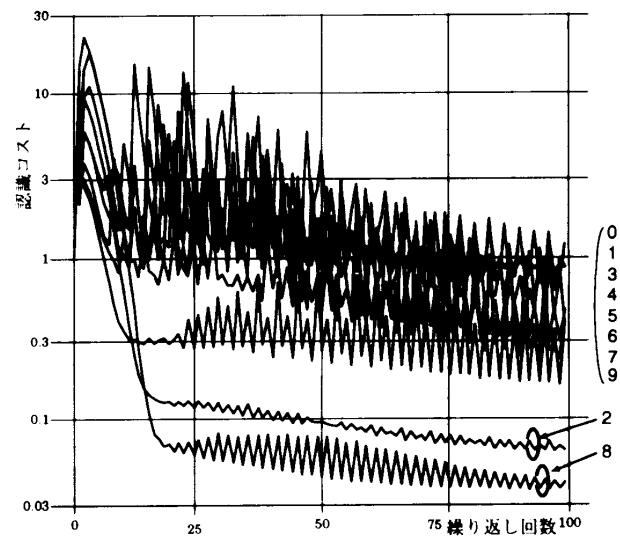


図 12 L の時間変化
Fig. 12 L Transition.

サンプリングセルより突き出ている線分は合成功力を示す. 右上の図は, サンプリングセルの入力を並べ換えたもので, ニューラルネットワークの入力に対応する. 図 13 は初期状態の図であるので, サンプリングセルの格子系は学習時の格子系と同じであるため, 入力パターンとニューラルネットワークの入力は同じである. 図の下側は, ニューラルネットワークの出力を示す. 左側の細い棒が最大活性値を示している. 入力パターンは 8 であるが, カテゴリ 2 についても出力が現われるが, ともに最大値ではない. また, わずかではあるがカテゴリ 2 の出力値が 8 よりも大きい. 従来の方針では, 2 と誤認識するか, 認識不能とするかのどちらかであるが, 入力領域適応型ニューラルネットワークを用いることにより正しく 8 と認識することができた. 図 13~16 は, 図の上にコメントで示すように, 仮定正解カテゴリを 8 とした場合で, 初期状態, 合成功力を 1 回適用した後, 10 回適用した後, 100 回適用した最終状態を示している. 10 回目の図 15 に示すように, サンプリングセルがパターン強度の強いところに集っている. これは, 入力パターンが潰れていたために, その部分を拡大することにより, 8 と解釈しようとしているところである. これは, ニューラルネットワークの入力が入力パターンと比べて, 8 の輪の部分が拡大されていることからも分かる. 図 16 に示す最終状態では, サンプリングセルの格子系が全体的に右にずれ, 8 に対する出力が最大となっている.

また, 図 17 と図 18 は仮定正解カテゴリを 2 としたときの, 10 回目と最終状態である. 合成功の適用に従って, カテゴリ 2 の活性度が増加し, カテゴリ 8 の活性度が減少する. 図 18 に示す最終状態では, カ

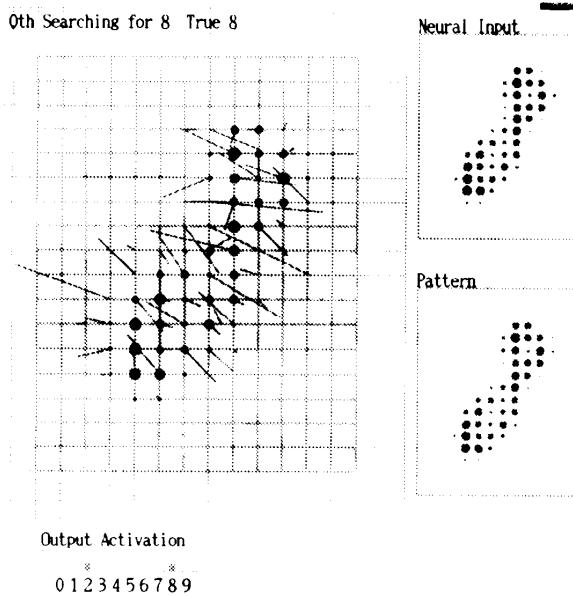


図 13 仮定正解カテゴリが 8 の初期状態
Fig. 13 Initial state of AIFNN with presumption as 8

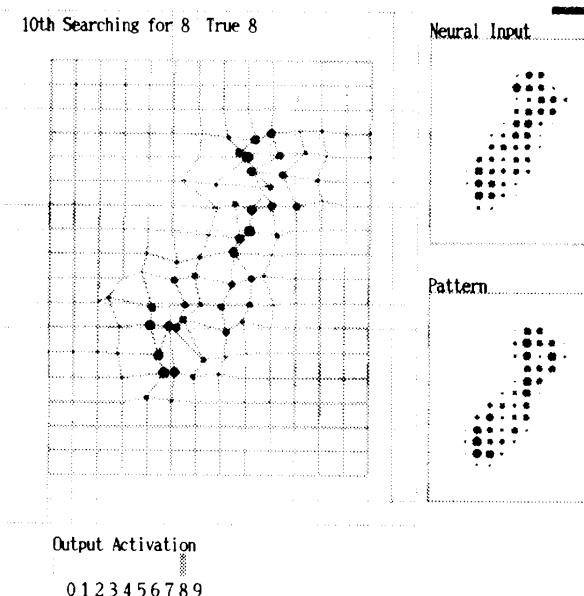


図 15 仮定正解カテゴリが 8 の 10 回目
Fig. 15 After 10th iteration with presumption as 8

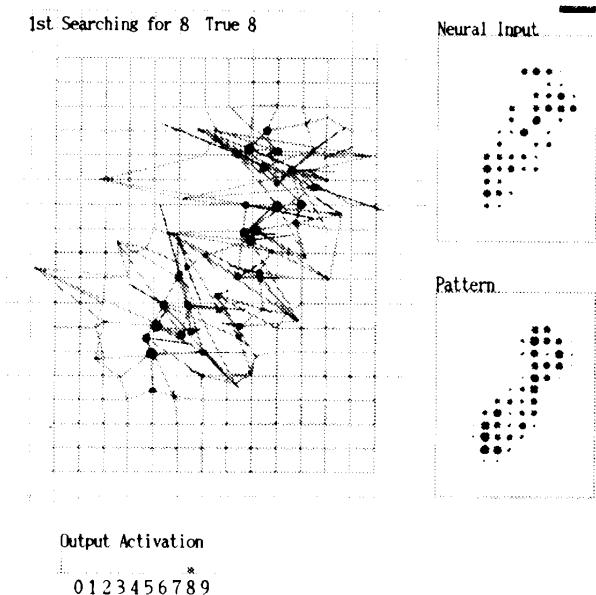


図 14 仮定正解カテゴリが 8 の 1 回目
Fig. 14 First state of AIFNN with presumption as 8.

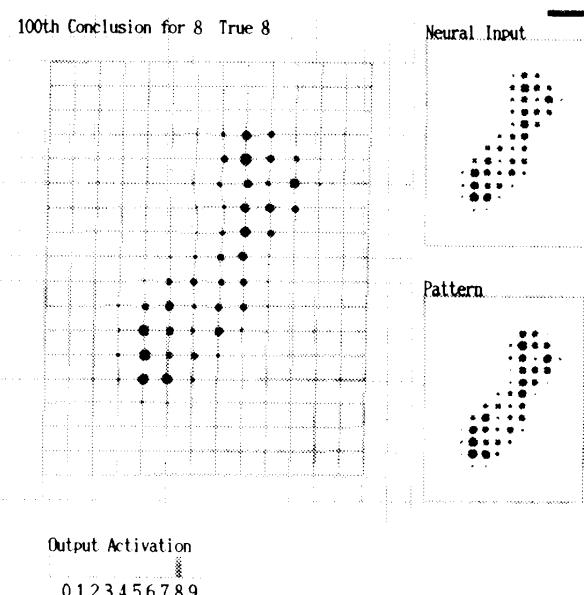


図 16 仮定正解カテゴリが 8 の 100 回目
Fig. 16 Final AIFNN state with presumption as 8.

テゴリ 2 に対してほぼ完全に活性化しているが、サンプリングセルの格子が歪んでいるために全体として L の値が大きくなる。図 19 と図 20 は、仮定正解カテゴリを 1 としたときの、5 回目と最終状態である。図 19 では、入力パターンを 1 と解釈するために、サンプリングセルがパターン強度の強いところから外れている。これによって、ニューラルネットワークの入力パターンは 1 に似る。しかしながら、アドレス力の影響でサンプリングセルは歪みのないように並べられ、最終状態では 8 も活性化する。その結果、 L の値が

大きくなる。

6. 実問題における入力領域適応型ニューラルネットワークの収束性

本手法の収束性は、4 章に示したが、つねに最小の状態に収束する保証はなく、ローカルミニマに収束してしまう可能性がある。ここでの問題となるローカルミニマには、(1) 初期状態の近傍にあるローカルミニマと (2) 初期状態から比較的遠方にあるローカルミニマがある。

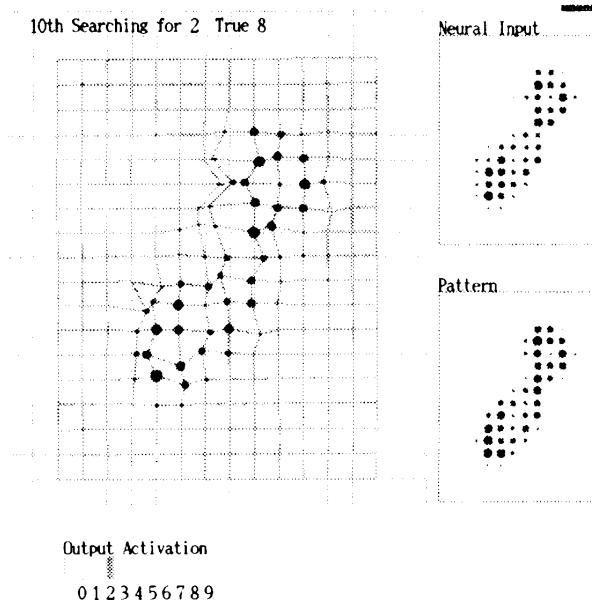


図 17 仮定正解カテゴリが 2 の 10 回目
Fig. 17 First state of AIFNN with presumption as 2.

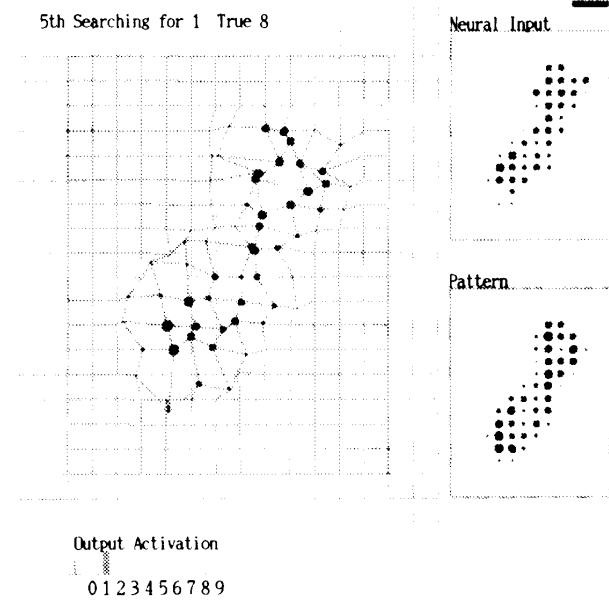


図 19 仮定正解カテゴリが 1 の 5 回目
Fig. 19 After 5th iteration with presumption as 1.

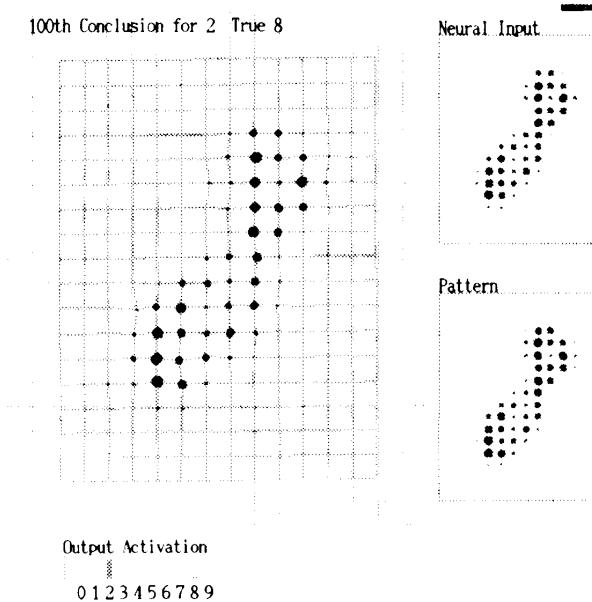


図 18 仮定正解カテゴリが 2 の 100 回目
Fig. 18 Final AIFNN state with presumption as 2.

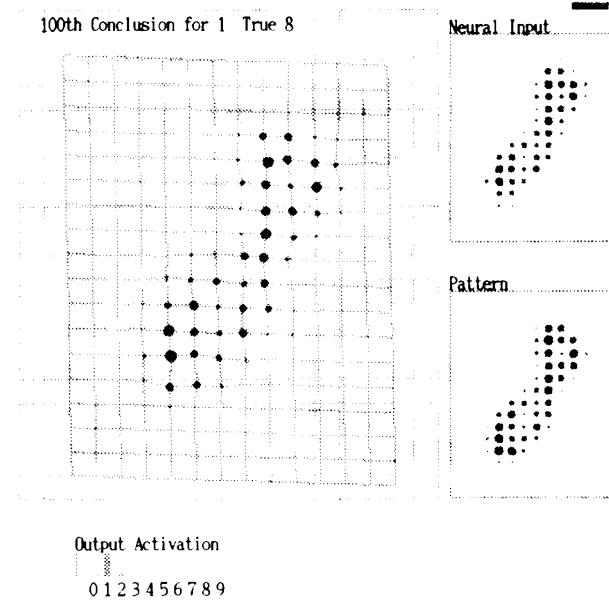


図 20 仮定正解カテゴリが 1 の 100 回目
Fig. 20 Final AIFNN state with presumption as 1.

(1) 初期状態の近傍にあるローカルミニマ

このローカルミニマは認識には問題となる。このローカルミニマは、式(4)と式(8)の η , μ を時間的に変更し、アニーリングのような原理で避けることが可能である。つまり、探索の開始直後では、デルタ力を大きく設定し、各サンプリングセルを必要以上に大きく動かす（図 14 参照）。しかし、この操作は確率的ではなく探索も 1 度だけなので、ローカルミニマを必ず避ける保証はない。

これとは別に、入力の前処理としてガウスフィルタ

を用いることにより、入力パターンを滑らかにしローカルミニマに収束することを避けている。このとき、ガウスカーネルのサイズとサンプリングセルの間隔と入力パターンの空間周波数には密接な関連がある。たとえば、サンプリングセルの間隔は入力パターンの特徴がとらえられる程度に短くしなくてはならない。また、ガウスカーネルのスケールを大きくしすぎると、入力パターンの特徴が曖昧になる等も考慮しなくてはならない。計算量と認識率とのかねあいを考えて、望ましくないローカルミニマに収束しないように、認識

対象によって調整する必要がある。

(2) 初期状態から比較的遠方にあるローカルミニマ

本手法では、初期状態から比較的遠方にあるローカルミニマには収束しない。これは、本手法は初期状態からの山登り法に近いので、比較的遠方にある極小点に収束することはほとんどないからである。

また、実際の認識問題では、比較的遠方にある極小状態に収束することは望ましくない。つまり、認識問題において、初期状態の近傍にあるほどもともらしい認識結果であるからである。これは、 L が最小の状態に収束することと、もっともらしい認識は、必ずしも一致しない立場である。

このことを、入力パターンが V であったとして説明する。本手法を用いると、 V と認識するが、 V の横棒があるために、入力領域に歪みが発生し、 L は極小とはなるが、最小の状態に収束することはできない。 L が最小の状態は、入力領域全体が 180 度回転したときに達成される。つまり、ローカルミニマは V と認識することで、最小状態は A と認識することである。この問題は、6 と 9 の認識ではさらに顕著になる。

7. 実問題への入力領域適応型ニューラルネットワークの適用

入力領域適応型ニューラルネットワークでは、仮定正解カテゴリを決めてからサンプリングセルの格子系の最適な affine 変換行列を求め、収束後の出力層での 2 乗誤差とサンプリングセルの格子系の歪みの和を比較し、最小の値を達成した仮定正解カテゴリを認識カテゴリとしている。仮定正解カテゴリを決めてから後は、各々の仮定正解カテゴリに対する処理は独立に行える。したがって、並列計算機を利用することによって、高速に行うことができる。

また、4 章の収束性の証明でも示したように、認識のときに利用する認識コスト関数 L （出力層でのエラー E^D と入力領域の歪エネルギー E^A の和）は必ず単調に減少し、実問題では L の時間変化は $\Gamma(1)$ 関数のような形になる。したがって、認識コスト関数の時間変化をすべての仮定正解カテゴリに対して比較すれば、どの仮定正解カテゴリがもっともらしいかあらかじめ見当をつけることが可能である。このようにして、並列に実行することによって実行時間を短くすることができる。また、この性質を使った別の方法として、まず 1 つの仮定正解カテゴリを決めて入力領域適応型ニューラルネットワークを実行し、認識コスト関数の時間変化を観察し、認識コスト関数がある閾値以下になかった場合は、その仮定正解カテゴリに対する

処理を打ち切って、別の仮定正解カテゴリを選択する方法がある¹⁰⁾。この方法では、全体の計算量を減らすことができるが、最低の認識コスト関数に対応するカテゴリを選択できなくなる可能性がある。また、この方法は ART との方法とよく似ていて、閾値を ART の Vigilance（覚醒）値と考えることができる。違いは、ART では入力から出力へのリンクと、出力から入力へのリンクは別々であるが、入力領域適応型ニューラルネットワークでは同じであり、それが学習済みのニューラルネットワークであることである。

4 章の収束性の証明でも示したように、入力領域適応型ニューラルネットワークがつねに収束するためには、式 (10) の近似が成り立つことが重要である。

$$\frac{\nabla \phi_j}{\epsilon^2 + |\nabla \phi_j|^2} \simeq \nabla \phi_j$$

したがって、入力パターンの強度勾配 $\nabla \phi_j$ が ϵ より小さいことが重要である。このことを考慮して ϵ とサンプリング関連のパラメータ（ガウスカーネルの大きさ等）を決める必要がある。

ニューラルネットワークの入力値を決めるとき、サンプリングセルの位置は入力パターンの格子点に必ずしも一致しないので、ラクランジエ補間法等を用いる必要がある。また、サンプリングセル近傍のパターンの強度勾配も同様にラクランジエ補間法等を用いる必要がある。これらの補間法でもっともな値を得るために、1 つの方法として、入力データをあらかじめガウスフィルタでぼかしている。また、これによって式 (10) が成立しやすいので都合がよい。

アドレス力の歪エネルギー E^A は式 (7) によって定義されている。

$$E^A \equiv \frac{1}{2} k \sum_i \|s_{N_i} - s_i\|^2$$

したがって、バネ定数 k を変更することにより、合成能力に対するアドレス力の影響を調整できる。バネ定数 k を大きくすれば、アドレス力が強くなり、収束後のサンプリング格子系は学習時の格子系から affine 変換可能な範囲に保たれる。また、小さくすれば、アドレス力が弱くなり、収束後のサンプリング格子系は学習時の格子系から affine 変換可能な範囲からはずれやすくなる。たとえば、 $k = 0$ とすると、アドレス力がまったく働かなくなり、すべてのサンプリングセルはニューラルネットワーク出力の 2 乗誤差が減少する方向のみに動く。収束後のサンプリング格子系が学習時の格子系から affine 変換可能な範囲からどのあたりまで許すのかは、バネ定数 k を試行錯誤で決める必要がある。

デルタ力とアドレス力は、式(4)と式(8)のように定義されていて、パラメータ η , μ を変更することによって合成力に対する影響を調整できる。収束を早くするために、探索の初めではデルタ力を大きく設定し、各サンプリングセルをニューラルネットワークの出力層での2乗誤差を減少するように大きく動かす。また、このように時間的に変更することにより、初期状態の近傍にあるローカルミニマを避けることができる(詳細は6章)。探索の終りの方ではアドレス力を大きく設定し各サンプリングセルを学習に用いた格子系のaffine変換可能な範囲に保つようとする。このように、デルタ力とアドレス力のパラメータの調整により、より早く収束させることが可能であるが、パラメータの設定には試行が必要となる。

入力領域の歪エネルギー E^A は、各サンプリングセルが格子系 S^N の各位置とバネ定数 k のバネの伸びによって決まる。バネ定数をすべてのサンプリングセルに対して独立に決めることも可能である。このときは、カテゴリによってバネ定数 k の分布のモデルを作る必要がある。このようにすることによって、カテゴリごとに変形の起きやすい部分と起きにくい部分を学習できる。このとき、 $\sum_i k_i$ を最小にする等の制約をつけながら、 k_i を学習する必要がある。

3.2節のアドレス力の説明では、affine変換について説明したが、入力データの変形の機構が分かっていて、入力データのサンプリング数よりも十分少ない数のパラメータで表現できるのならば、アドレス力は適用可能で、入力領域適応型ニューラルネットワークが適用可能である。たとえば、1つの入力データ全体に対するゲインやオフセットをパラメータとして補正することができる。また、高所から俯瞰した物体に対する認識に対しても同様に行う適応ができる(図21参照)。

入力領域適応型ニューラルネットワークは仮定正解カテゴリとの2乗誤差を小さくするように各サンプリングセルをパターン上で移動させ、同時にサンプリングセルの格子系が元の格子系からaffine変換可能な範囲に保つようにしている。入力領域適応型ニューラルネットワークを適用した結果、入力領域は、元の入力パターンを変換して仮定正解カテゴリに属する標準的なパターンに可能な限り似せるようになる。つまり、入力領域適応型ニューラルネットワークによって仮定正解カテゴリを正当化するように、入力領域を変更し、入力パターンを再解釈したと見なすことができる。この効果は、メンタルローテーション等とも関連づけられる。また、別な見方をすると、入力領域適応型ニューラルネットワークは出力層に与えられた仮定正

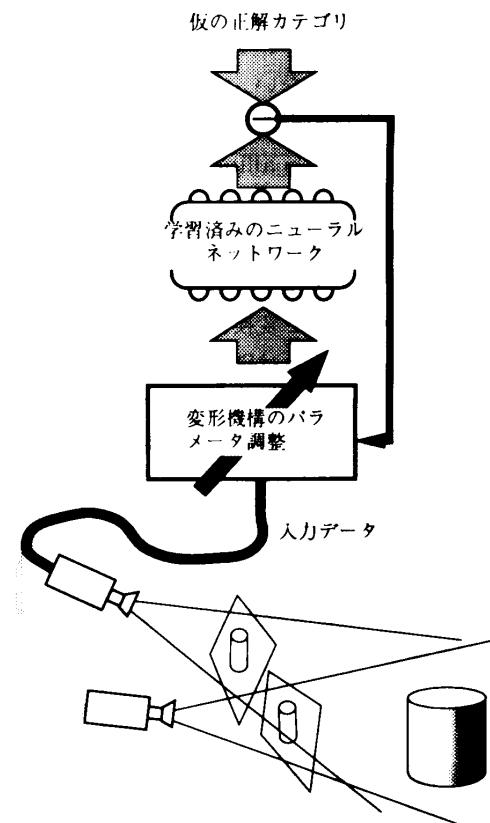


図21 その他の認識への適用
Fig. 21 AIFNN other applications.

解カテゴリをトップダウンに伝播し、入力パターンの再認識をする。同様のことは、ネオコグニトロン¹⁵⁾やART²⁰⁾でも行える。違う点は、トップダウンのために別なニューラルネットワークを持たず、認識に用いるニューラルネットワークを使うことである。

入力領域適応型ニューラルネットワークは学習済みのニューラルネットワークを用いて、カテゴリの認識を行ったが、マルチテンプレートマッチングにも応用できる。これは、デルタ力はサンプリングセルの入力値を変更し仮定正解カテゴリに対する2乗誤差を減少させることができるのであり、この2乗誤差をテンプレートとサンプリングしたパターンとのノルムと置き換えることによって容易に実現できる。具体的には、式(3)の Δnet_j を以下のように変更する。ここで net_j^T はテンプレートの j セルに対応する値である。

$$\Delta net_j = \eta (net_j^T - net_j) \quad (15)$$

ニューラルネットワークの学習に用いた手書き文字のデータには、affine変換の補正を行っていない。そのため、学習済みのニューラルネットワークは学習データに含まれているaffine変換に対しては正しく認識できる。本方式では、入力パターンのaffine変換は入力領域で補正し、非線形な変形のみをニューラル

ネットワークで処理することを目的としている。したがって、ニューラルネットワークに affine 変換を学習させることは好ましくない。そこで、ニューラルネットワークが非線形な変形のみを学習できるように、あらかじめ学習データから affine 変換を取り除く必要がある²¹⁾。

入力領域適応型ニューラルネットワークでは、入力領域の各サンプリングセルを独立に動かし、それらの間に元の格子から affine 変換の範囲内に保たれるようになっている。この方法とは別に、直接 affine 変換パラメータ (t_{ij} , $i = 1, 2$, $j = 1, 2, 3$) を数値計算を用いて推定することも可能である。すなわち、以下の近似で求める。

$$\frac{\partial E^D}{\partial t_{ij}} \simeq \frac{E^D(t_{ij}) - E^D(t_{ij} - \Delta t_{ij})}{\Delta t_{ij}} \quad (16)$$

これによって、affine 変換パラメータを弛緩法で求めることが可能となる。

8. 結 論

入力領域適応型ニューラルネットワークは、認識時に入力データに対して最適な affine 変換を弛緩法を用いて推定し、ニューラルネットワークで認識する新しい認識手法で、従来から問題であった入力データの切り出しと正規化の誤りによる誤認識を減らすことができる。また、認識の評価基準として用いた、出力層でのエラー E^D と入力領域の歪エネルギー E^A の和は、リヤノフ関数でアルゴリズムはつねに収束することが保証されている。3層構造の学習済みのニューラルネットワークを用いて手書き数字文字認識の実験を行った。入力領域適応型ニューラルネットワークを用いることにより、テストデータ中で誤認識されていた 120 パターンのうち、22.50% にあたる 27 パターンを正しく認識することができ、誤認識パターンを 4.800% (120 パターン) から 3.720% (93 パターン) に減らすことができ、入力領域適応型ニューラルネットワークを用いることによりニューラルネットワークの認識性能を向上させることができることを示した。本論文では、affine 変換について説明したが、入力データの変形の機構が明らかである場合に適用可能で、たとえば三次元の物体の認識で視点の位置に不变の認識など応用可能である。

謝辞 研究の機会を与えて下さりご指導して下さった、NEC 欧州小池ジェネラルマネージャ、日本電気(株) C&C 研究所山本所長、C&C 研究所コンピュータシステム研究部中崎部長に感謝いたします。

参 考 文 献

- 1) Hinton, G., Williams, C. and Revow, M.: Adaptive Elastic Models for Hand-Printed Character Recognition, *Advances in Neural Information Processing Systems*, Vol.4 (1992).
- 2) Simard, P., Cun, Y.L. and Denker, J.: Efficient Pattern Recognition Using a New Transformation Distance, *Advances in Neural Information Processing Systems*, Vol.5 (1993).
- 3) Simard, P., Victorri, B., Cun, Y.L. and Denker, J.: Tangent Prop - A Formalism for Specifying Selected Invariance in an Adaptive Network, *Advances in Neural Information Processing Systems*, Vol.4 (1992).
- 4) Buhmann, J., Lange, J. and von der Malsburg, C.: Distortion Invariant Object Recognition by Matching Hierarchically Labeld Graphs, *Proc. of IJCNN '89 at Washington DC.*, Vol.1 (1989).
- 5) Barnard, E. and Casasent, C.: Invariance and Neural Nets, *IEEE Trans. on Neural Networks*, Vol.2, No.5, pp.498-508 (1992).
- 6) Rumelhart, D., McClelland, J. and the PDP Research Group: *Parallel Distributed Processing*, MIT Press, Cambridge, MA (1986).
- 7) 早野克巳, 鷺沢輝芳: サッケード機構を組み込んだ神経回路モデル—手書き文字の位置ずれに対する性能評価—, 情處全大, Vol.5E-7 (1989).
- 8) Widrow, B., Winter, R.G. and Baxter, R.A.: Layered Neural Nets for Pattern Recognition, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol.36, No.7, pp.1109-1118 (1988).
- 9) Asogawa, M.: Adaptive Input Field Neural Network - That can Recognize Rotated and/or Shifted Character, *Proc. of IJCNN '90 at San Diego*, Vol.3, pp.733-738 (1990).
- 10) Asogawa, M.: Adaptive Input Field Neural Network, *Proc. of IJCNN '91 at Singapore*, Vol.1, pp.83-88 (1991).
- 11) Bodenhausen, U.: The Temp Algorithm, *Proc. of IJCNN '90 at Washington D.C.*, Vol.1, pp.597-600 (1990).
- 12) Li, W. and Nasrabadi, N.: Invariant Object Recognition Based on a Neural Network of Cascaded RCE Nets, *Proc. of IJCNN '90 at San Diego*, Vol.2 (1990).
- 13) アレックスワイベル: 時間遅れ神経回路網(TDNN)による音声認識, 信学技報 SP87-100 (1987).
- 14) 追江博昭, 磯健一: ダイナミックニューラルネットワークの検討, 信学技報 SP87-101 (1987).
- 15) Fukushima, K.: Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Recognition Unaffected by Shift in Posistion,

- Biological Cybernetics*, Vol.36, No.4, pp.193-202 (1980).
- 16) 山田敬嗣, 津雲淳: 多層ニューラルネットをもつたパターンの意識的連想起処理方法, 電子情報学会春季全国大会論文集, Vol.6, D-540 (1989).
- 17) Williams, R.J.: Inverting a Connectionist Network Mapping by Backpropagation of Error, *8th Annual Conf. Cognitive Sci. Soc.* (1986).
- 18) Ishikawa, M.: A Structural Learning Algorithm with Forgetting of Link Weights, *Proc. of IJCNN '89 at Washington DC.*, Vol.2 (1989).
- 19) Fahlman, S.: An Empirical Study of Learning Speed in Back-propagation Networks, Technical Report CMU-CS-88-162, Carnegie Mellon University (1988).
- 20) Carpenter, G. and Grossberg, S.: ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns, *Applied Optics*, Vol.26, No.23, pp.4919-4930 (1987).
- 21) Asogawa, M.: *Bootstrap Learning for Invariant Recognition* (in press).

付録 最小 2乗法を用いて最適な affine 変換行列を求める方法

学習時に用いたサンプリング格子系を \mathbf{S}^O での第 j サンプリングセルの位置を (s_{j1}^O, s_{j2}^O) と表す。この位置に, affine 変換を $\mathcal{T}(\cdot)$ 適用した最適なサンプリング格子系 \mathbf{S}^N での第 j サンプリングセルの位置 (s_{j1}^N, s_{j2}^N) は, 以下のように表すことができる。

$$\begin{bmatrix} s_{j1}^N \\ s_{j2}^N \end{bmatrix} = \mathcal{T} \begin{bmatrix} s_{j1}^O \\ s_{j2}^O \end{bmatrix}$$

$$= \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} s_{j1}^O \\ s_{j2}^O \end{bmatrix} + \begin{bmatrix} t_{13} \\ t_{23} \end{bmatrix} \quad (17)$$

ここで, このサンプリング格子系 \mathbf{S}^N と現在の各サンプリングセルの位置 \mathbf{S} との距離 $d(\mathbf{S}^N, \mathbf{S})$ は,

$$d(\mathbf{S}^N, \mathbf{S}) = \sum_i \| \mathbf{s}_i^N - \mathbf{s}_i \| ^2$$

$$= \sum_i \{ (s_{i1}^N - s_{i1})^2 + (s_{i2}^N - s_{i2})^2 \}$$

$$(18)$$

となる。明らかに, 距離 $d(\mathbf{S}^N, \mathbf{S})$ は affine 変換行列の各パラメータ t_{ij} に対して二次式であるので, これを最小とする affine 変換行列は, $d(\mathbf{S}^N, \mathbf{S})$ の t_{ij} に対する偏微分が 0 であることを用いて求めることができる。

$$\frac{\partial d(\mathbf{S}^N, \mathbf{S})}{\partial t_{ij}} = 0, \quad \text{for } i = 1, 2 \quad j = 1, 2, 3 \quad (19)$$

具体的に, $i = 1, j = 1$ について式 (19) の左辺を式 (18) を用いて計算すると,

$$\begin{aligned} \frac{\partial d(\mathbf{S}^N, \mathbf{S})}{\partial t_{11}} &= \sum_i \frac{\partial \{ (s_{i1}^N - s_{i1})^2 + (s_{i2}^N - s_{i2})^2 \}}{\partial t_{11}} \\ &= 2 \sum_i (t_{11}s_{i1}^O + t_{12}s_{i2}^O + t_{13} - s_{i1}) (s_{i1}^O) \\ &= -2 \sum_i s_{i1}s_{i1}^O + 2t_{11} \sum_i s_{i1}^O \\ &\quad + 2t_{12} \sum_i s_{i2}s_{i1}^O + 2t_{13} \sum_i s_{i1}^O \quad (20) \end{aligned}$$

となる。同様に t_{12}, t_{13} について求めて, 式 (20) とまとめると, 以下のようになる。

$$\begin{bmatrix} \sum_i s_{i1}^{O2} & \sum_i s_{i1}^O s_{i2}^O & \sum_i s_{i2}^O \\ \sum_i s_{i1}^O s_{i2}^O & \sum_i s_{i2}^{O2} & \sum_i s_{i2}^O \\ \sum_i s_{i1}^O & \sum_i s_{i2}^O & \sum_i 1 \end{bmatrix} \begin{bmatrix} t_{11} \\ t_{12} \\ t_{13} \end{bmatrix} = \begin{bmatrix} \sum_i s_{i1}^O s_{i1} \\ \sum_i s_{i2}^O s_{i1} \\ \sum_i s_{i1} \end{bmatrix} \quad (21)$$

ここで, 式 (21) の第 1 項の行列を \mathcal{S} と表すと, t_{11}, t_{12}, t_{13} は以下のようにして求めることができる。

$$\begin{bmatrix} t_{11} \\ t_{12} \\ t_{13} \end{bmatrix} = \mathcal{S}^{-1} \begin{bmatrix} \sum_i s_{i1}^O s_{i1} \\ \sum_i s_{i2}^O s_{i1} \\ \sum_i s_{i1} \end{bmatrix} \quad (22)$$

ここで \mathcal{S} は, 学習時に用いたサンプリング格子系 \mathbf{S}^O によってのみ決まることに注意すると, 逆行列の計算はたかだか 1 回計算すればよいことが分かり, 対称行列で要素がすべて実数であるので逆行列は容易に計算できる。式 (22) と同様にして, t_{21}, t_{22}, t_{23} を求めることができる。

(平成 7 年 4 月 3 日受付)

(平成 7 年 11 月 2 日採録)



麻生川 稔（正会員）

1984 年南カリフォルニア大学より
Ms.Sc. (Computer Science). 1986
年東京工業大学理工学研究科電子物
理工学専攻修士課程修了。同年日本
電気（株）入社。1992 年から 93 年
にかけて CMU 客員研究員。エキスパートシステム、
ニューラルネットワーク、遺伝子情報処理等の研究に
従事。神経回路学会会員。1992 年情報処理学会全国
大会奨励賞授賞。
