

# 確率的ニューラルネットワーク計算の並列高速化アーキテクチャとその画像認識システムへの適用

相部 範之<sup>†</sup> 安永 守利<sup>††</sup>

確率的ニューラルネットワーク (PNN: Probabilistic Neural Network) はその認識精度の高さから、近年多くの実用的なパターン認識問題 (顔画像認識, 気象画像認識など) に適用されている。一方、高い認識精度を得るためには、非常に多くのサンプルパターンに対してネットワーク計算を行う必要がある。さらに、ネットワークのパラメータ決定 (学習) では、この計算を繰り返し実行する必要がある。このため PNN 計算の高速化には専用設計されたハードウェアが不可欠である。本論文では、PNN 計算を高速に実行するための専用プロセッサとこれをベースとした並列アーキテクチャを提案する。さらに、これらの提案方式に基づく小型 (1 ボード) 画像認識システムを試作する。そして試作システムの実行時間を測定し、ビデオレート時間 (33.3 ms) 内に大規模パターン認識処理が終了することを示す。また、実測値からの推定により 10 秒程度で大規模パターン認識問題の学習が終了することを導き、本提案手法とそれに基づく画像認識システムの有効性を示す。

## An Architecture for Parallelized High-speed Probabilistic-Neural-Network Calculation and Its Application to Image Recognition Systems

NORIYUKI AIBE<sup>†</sup> and MORITOSHI YASUNAGA<sup>††</sup>

Recently, probabilistic neural networks (PNN) are being applied to the practical pattern recognition applications such as face recognition and satellite-image recognition. In order to obtain high recognition accuracy, however, PNN requires a large number of sample patterns to be processed in it. Furthermore, the PNN calculation must be repeated so many times in its learning-stage to determine a network parameter precisely. Specialized hardware is thus indispensable for the high speed PNN calculation. In this paper, we propose a specialized processor and a parallel architecture using the processors for high speed PNN calculation. And we develop an image-recognition prototype system in a small single board. Measurements of the system show its high-speed recognition in a video-rate period of 33.3ms and a high-speed learning less than 10 seconds is estimated using the measured result. Those results demonstrate effectiveness of the proposed hardware for the practical pattern recognition problems.

### 1. はじめに

確率的ニューラルネットワーク (PNN: Probabilistic Neural Network) は Specht によって提案された 3 層構造のフィード・フォワード・ネットワークで、ベイズの識別定理に基づいた、統計的なパターン認識手法の 1 つである<sup>1)~4)</sup>。近年、その応用事例として、Mao らの顔認識問題への適用<sup>5)</sup>、Tian らの衛星からの気象画像の認識問題への適用<sup>6),7)</sup> など、実用的な大規模問

題への適用事例が数多く報告されており、その認識精度 (認識率) の高さが注目されている。PNN による認識器が高い認識精度を示す理由は、PNN がサンプルパターンから形成されるカーネル関数を重ね合わせることで、各カテゴリ間の真の確率密度分布の関係を高精度に近似するからである。したがって、サンプルパターンの数が増すほど、その認識精度はベイズ統計に従う理想的な値に近付き、高い認識精度を持つ認識器が実現できる。

一方 PNN では 1 つの未知パターンを認識するために、毎回非常に多くのカーネル関数計算を必要とする。このため、画像認識などの大規模パターン認識では多大な計算量が発生する。一般に画像処理ではビデオレート (33.3 ms) に追従することが 1 つの基準となるが、PNN によるビデオレートに追従した認識のた

<sup>†</sup> 筑波大学大学院博士課程システム情報工学研究科  
Graduate School, Doctoral Program of Systems and Information Engineering, University of Tsukuba

<sup>††</sup> 筑波大学電子・情報工学系  
Institute of Information Sciences and Electronics, University of Tsukuba

めには多大な計算量を高速に処理するための専用ハードウェアが必要となる。さらに、PNN のパラメータ決定(学習)ではカーネル計算を繰り返す必要があり、その計算量は膨大なものとなる。このため、頻繁に生成/更新されるサンプルパターンに追従してパラメータを更新するオンライン学習においても、専用ハードウェアが必要となる。このように PNN では、その認識処理と学習の両方において高速化が望まれている。本論文の目的は、PNN の高速画像認識と高速学習のためのハードウェア(回路とアーキテクチャ)提案を行い、その有効性を開発した試作システムにより実証することである。

以降、2章では PNN のアルゴリズムと専用プロセッサについて説明し、3章では並列学習アーキテクチャを、4章ではそれらの回路構成を、5章では開発した試作システムとそれを用いた認識、および学習の評価結果を述べる。

## 2. 確率的ニューラルネットワークと高速化のための専用プロセッサ

### 2.1 確率的ニューラルネットワーク

PNN は図 1 に示すように、3 層のフィード・フォワード・ネットワークから成る。PNN によるパターン認識の基本的な考え方は、図 1 中のカーネル層(Kernel Layer)のカテゴリごとのニューロンの出力値を加算層(Summation Layer)で重ね合わせることで各カテゴリの確率密度分布を計算し、その大小関係を決定層(Decision Layer)において判定することで未知入力パターンのカテゴリを推定するものである。

未知入力パターン(特徴ベクトル)  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  は、カーネル層のカテゴリごとのニューロン群へ入力され、各ニューロンはサンプルパターン  $S_j^i$  (カテゴリ  $C_i$  の第  $j$  番目のサンプルパターン) を基に定義された関数値  $K(\mathbf{X} - S_j^i)$  を出力する。この  $K(\mathbf{X} - S_j^i)$  をカーネル関数とよび、この具体的な関数としてガウシアン関数や一様関数

$$K(\mathbf{X} - S_j^i) = \begin{cases} 1: & |x_k - (s_j^i)_k| \leq \frac{\sigma}{2} \\ 0: & \text{otherwise} \end{cases} \quad (1)$$

(for all  $k, k = 1, 2, \dots, n$ )

が用いられる。ただしここで、

$$\mathbf{X} = (x_1, x_2, \dots, x_n),$$

$$S_j^i = ((s_j^i)_1, (s_j^i)_2, \dots, (s_j^i)_n)$$

で、 $x_k, (s_j^i)_k$  はそれぞれ第  $k$  番目の成分を表し、 $n$  はパターンの次元数を表す。また、式(1)中の  $\sigma$  はカーネル関数の広がりを表すパラメータである。この

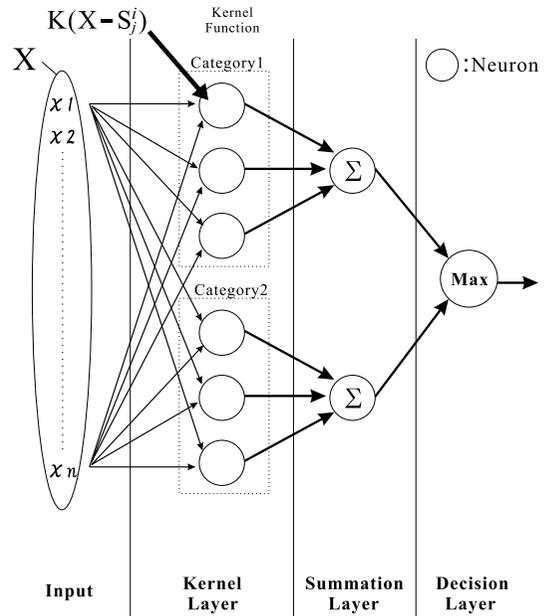


図 1 確率的ニューラルネットワークの構成  
Fig.1 Network configuration of PNN.

パラメータを高精度に決定することが、PNN の認識精度を決定するうえで重要となる。

次に加算層(Summation Layer)において、このカーネル関数を各カテゴリごとに重ね合わせることで、未知パターン  $\mathbf{X}$  がカテゴリ  $C_i$  に属する確率密度分布  $p(\mathbf{X}|C_i)$  を推定する。すなわち、

$$\hat{p}(\mathbf{X}|C_i) = \frac{1}{N_p} \sum_{j=0}^{N_p} K(\mathbf{X} - S_j^i) \quad (2)$$

と表せる。なお、通常  $N_p$  は各カテゴリで同じとするため、 $1/N_p$  の項は 1 と見なして処理することができる。

最後に決定層(Decision Layer)で最も大きな確率値を得たカテゴリを求めることで、未知パターンのカテゴリが推定できる。

### 2.2 sPNN(serial-PNN) Processor

PNN も含めて、ニューラルネットワークのハードウェア化でニューロン(プロセッサ)設計の中心となるのは、ニューロンの非線形関数計算をいかに効率的に行うかにある。これまでアナログニューロン回路では、非線形関数をデバイスの非線形特性を用いて直接回路化することが行われてきた<sup>8),9)</sup>。一方、デジタルニューロン回路では非線形関数を直接回路化すると回路規模が増大するため得策ではない。このため、これまでの多くの開発事例では、非線形関数を演算器

とマイクロプログラムの組合せや、メモリ (Look-up Table) によって処理してきた<sup>8),10)~12)</sup>。一方、我々の提案する sPNN Processor はデジタルニューロン回路でありながら、非線形関数 (カーネル関数) を単純な組合せ回路で直接実現する点にその特徴がある。直接回路化が可能な理由は、非線形関数を回路化に適した一様関数の重ね合わせで表現することによる。これにより、後述するようにビデオレート (33.3 ms) 以内での高速画像認識を実現することができる。なお、一様関数を用いることの妥当性、および sPNN Processor の回路の詳細 (カーネル関数の具体的な回路化方法) については 4 章で述べる。

### 3. 高速学習のための並列アーキテクチャ

認識精度を高めるためには、多くのサンプルパターンが必要であるだけでなく、PNN のパラメータ  $\sigma$  を適切に決定する必要がある。この  $\sigma$  を最適化することが PNN の学習である。

PNN はサンプルパターンが多いため、1 回のネットワーク計算 (認識処理) だけでも多大な計算量を要する。さらに、前述した顔画像認識や気象画像認識といった実用的な大規模問題では、真の確率密度分布はまったく未知であり、このため  $\sigma$  を決定論的に求めることはできない。したがって  $\sigma$  をその全空間にわたって探索し、適切な値を得る必要がある。このため  $\sigma$  の学習は、その値を少しずつ変えながら、ネットワーク計算を繰り返す必要があり、その計算時間は認識処理時間とは比べものにならないほど多大なものになる。通常、学習はサンプルパターンの追加や変更のたびにを行う必要があり、大規模な画像認識システムでは頻繁にこの追加/変更が行われるため、学習時間の高速化が望まれている。

特に 1 章でも述べたとおり、PNN のオンライン学習では高速な学習が必要となる。たとえば PNN を用いた具体的な報告事例として、文献 6), 7) では衛星 (GOES-8) から転送される気象画像 (1 画像/分) に対して PNN のオンライン学習を行っている。しかし、現在その処理には約 5 分かかっており、学習処理時間が画像の転送スループット時間に追いついていない。雷、竜巻などのスポット予測のために、画像の転送スループットは今後いっそう増加する傾向にあり、PNN の高速学習のための専用ハードウェアが不可欠となっている。またこの事例のほか、たとえば PNN の高速学習をロボットのオンライン運動制御システムに組み込むことで、周囲の環境変化に即座に対応することのできるロボットの実現が可能となる。

我々の提案する高速学習のための並列アーキテクチャを Sigma-Parallel Architecture (SPA) と呼ぶ。SPA は、複数の PNN のネットワークにそれぞれ異なる認識パラメータ  $\sigma$  を与え、それを並列に計算することで学習の高速化を図るものである。すなわち、異なる複数の  $\sigma$  に対して前述の sPNN Processor を多数並列実装する。ここで重要なことは、SPA が VLSI 実装を考慮し、従来より多くのニューラルネットワークで問題となっている VLSI 実装上のピン・ボトルネックを生じない構造となっている点である。さらに複数の sPNN Processor をモジュール化し、これらをニューロン並列 (後述) とすることでさらなる高速化を実現する (これを Hybrid-SPA と呼ぶ)。以下、これらの並列アーキテクチャについて述べる。

#### 3.1 Sigma-Parallel Architecture (SPA)

前述したように、図 1 に示すカーネル層にある各ニューロンは、それぞれカテゴリ  $C_i$  に属する第  $j$  番目のサンプルパターン  $S_j^i$  を基に構成される。サンプルパターン  $S_j^i$  のサイズは、この次元数を  $n$ 、各要素のビット精度を  $q$  bit としたとき、 $qn$  bit となる。このサイズのサンプルパターンが 1 カテゴリにつき  $N_p$  個用いられ、それがさらに  $N_c$  カテゴリ分あるとすると、すべてのサンプルパターンを記憶しておくために必要なメモリの容量は、 $qnN_pN_c$  bit となる。ここで、たとえば実的な例として、 $q = 8$ ,  $n = 256$ ,  $N_p = 512$ ,  $N_c = 8$  とすると必要なメモリ容量は 1MByte となり、さらにカテゴリ数  $N_c$ , 1 カテゴリに含まれるサンプル数  $N_p$  などが増えれば、必要なメモリ容量は数 MByte ~ 数百 MByte と、大幅に増加する。1 つの集積回路内にこれだけの容量のメモリを演算回路 (ランダムロジック) とともに実装することは困難である。このため、これらのサンプルパターンは演算回路とは別に、外部の専用メモリに記憶する必要がある。

したがって、図 2 (a) に示すように、従来よりバックプロパゲーションの並列化などで用いられてきた「ネットワーク上の各ニューロンを専用のプロセッサにハードウェア化し、集積回路内で並列化する手法 (これをニューロン並列と呼ぶ)」を PNN にそのまま適用するためには外部メモリが不可欠となる。この場合、カーネル層のニューロンを計算するプロセッサ (図 2 (a) 中の Neuron in Kernel Layer) は、それぞれ異なるサンプルパターンが同時に必要となるため、外部メモリのデータバスもそれぞれ独立に必要となる。未知入力パターン  $X$  は各ニューロンに集積回路内でブロードキャストされる。認識パラメータ  $\sigma$  も  $X$  と同様に、集積回路内でブロードキャストされ、同じ値が各

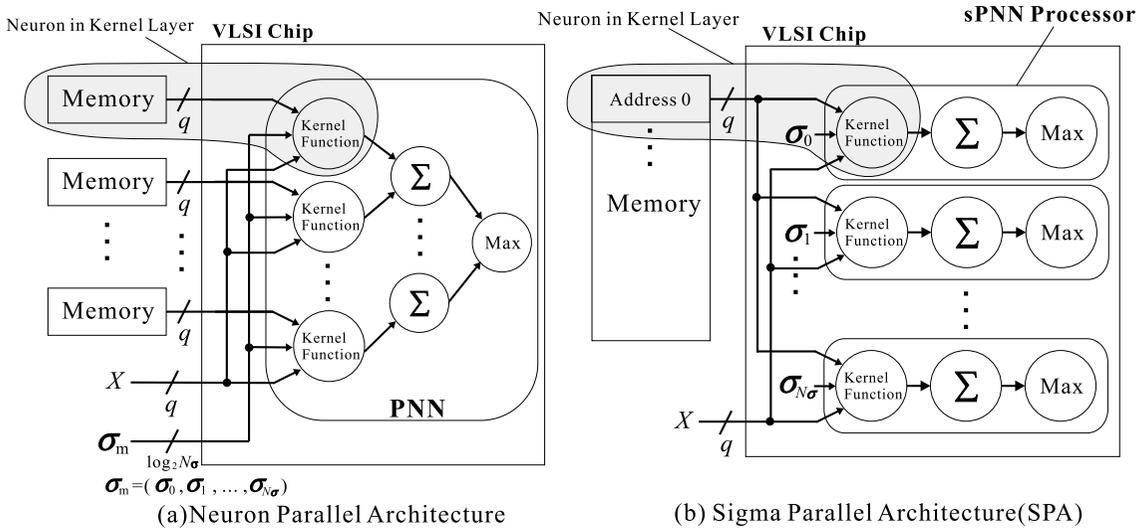


図2 従来手法(ニューロン並列)とSPAの比較  
Fig. 2 Neuron Parallel Architecture vs. Sigma Parallel Architecture.

ニューロンの演算で使われる。学習ではこの共通な  $\sigma$  を  $\sigma_m$  として  $m = 0$  から  $m = N_\sigma$  まで ( $N_\sigma$  は探索する  $\sigma$  の総数) 少しずつ変えて、繰り返し演算を行うことでニューロンの並列性を利用した高速な学習が可能となると考えられる。

しかし、このニューロン並列のアーキテクチャには大きな問題がある。図2(a)では、 $qN_pN_c$  本のサンプルパターン用の入力ピン数(メモリのデータバス幅)と、 $q$  本の未知入力ピン数、合計  $q(N_pN_c + 1)$  本の入力ピン数を必要とする。たとえば先の例、 $q = 8$ 、 $N_p = 512$ 、 $N_c = 8$  のとき、必要なピン数は  $q(N_pN_c + 1) = 32,776$  本と、非現実的な要求ピン数となる。

そこで、我々は「異なる  $\sigma$  をニューロンに与え、各ニューロンの下で並列にテストパターンに対して認識精度を計算する」手法を提案する(図2(b))。これを SPA (Sigma-Parallel Architecture) と呼ぶ。SPAでは前述した sPNN Processor が異なる  $\sigma$  に対してそれぞれ並列に動作する。sPNN Processor は次章でその詳細を述べるように、時分割処理で PNN を計算する。実装上、ニューロンは各層とも1つのみで構成されており、これを繰り返し実行することで1つのネットワーク(PNN)全体を計算する。この sPNN Processor を1つの集積回路内に多数、並列に実装し、それぞれに異なる  $\sigma$  を与えることで、並列化をニューロン並列方式 (Neuron-Parallel Architecture) ではなく、 $\sigma$  並列方式 (Sigma-Parallel Architecture) とすることができる。 $\sigma$  はサンプルパターンのようにデータサイズが大きくないので ( $\log_2 N_\sigma$  は数~数十ビット

ト)、sPNN Processor と同じ1つの集積回路内に容易に実装することができる。

各 sPNN Processor は逐次処理をするため、外部メモリや未知パターンの入力に必要な物理的なピン数も少なくなる。具体的には、サンプルパターンの入力には  $q$  bit、未知パターンの入力も  $q$  bit で合計  $2q$  本の入力ピン数で済む。サンプルパターンの入力は逐次処理となるが、チップ内でパラメータ  $\sigma$  の並列処理ができるため、結果としてピン数の問題を解決しつつ、高速学習が可能となる。

ニューラルネットワークは、多ファン・イン/アウトを持つニューロンが多数結合したネットワークで構成される。このため、ニューラルネットワークの VLSI 実装においては、従来よりピン数不足をいかに解決するかが本質的な問題であり、これまでバックプロパゲーション型や完全結合型、自己組織化型ニューラルネットワーク等々の個々のネットワークごとにこの問題を解決すべく様々なアーキテクチャが提案、開発されてきた<sup>8),12)~16)</sup>。SPA アーキテクチャは PNN を対象として、この問題を解決しつつ高並列処理を実現している。

### 3.2 Hybrid-SPA

図2(b)のSPAは従来のニューロン並列構成と排他的に選択するものではない。SPAで構成した複数の sPNN Processor をモジュール化し(図2(b)中の VLSI Chip 領域を図3中の SPA-Module に示すようにモジュール化する)、これを与えられたピン数制限の範囲で図3のように多重に並列に実装することで、

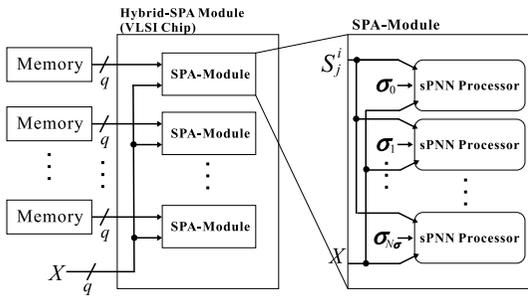


図 3 SPA の多重並列化

Fig. 3 Neuron parallelism based on the SPA.

さらなる高速化が期待できる．このような， $\sigma$  並列を基本としてさらにこれらをニューロン並列化する構成を Hybrid-SPA と呼ぶこととする．現在この Hybrid-SPA に基づく小型画像認識システムを開発中であり，本論文では後半でその試作評価結果を述べる．

#### 4. 回路構成

本章では PNN を実際に SPA に基づいてハードウェア化する際の回路構成について，まず sPNN Processor の回路構成について述べ，次に Hybrid-SPA の全体回路構成について述べる．

##### 4.1 sPNN Processor の回路構成

sPNN Processor の回路構成を図 4 に示す．この回路は入力として未知パターン  $X$ ，サンプルパターン  $S_j^i$ ，認識パラメータ  $\sigma_m$  を持ち，PNN の演算を時分割処理で行う．図 1 の PNN に対応して 3 層から成るが，物理的に存在する各層のニューロンは 1 つずつで，これらを用いて逐次処理することで PNN を構築し，認識処理を行う．未知パターン  $X$  とあらかじめカテゴリごとに外部メモリに記録されているサンプルパターン  $S_j^i$  は，それぞれ  $q$  bit の各要素  $x_1, x_2, \dots, x_n$  および， $((s_j^i)_1, (s_j^i)_2, \dots, (s_j^i)_n)$  ごとに逐次処理される（すなわち，式 (1) の演算を各要素ごとに行う）． $\sigma_m$  は第  $m$  番目の  $\sigma$  で，SPA では各ニューロンごとに異なった値をとる．

ここで，対象とするカーネル関数として，前述した一様関数（式 (1)）を選択した．これは以下に示すように少ない回路規模で関数を直接回路化することができ，高速かつ小型化が可能であるからである．一般に関数を近似するには，一様関数よりガウシアン関数の方が優れているので，PNN においても確率密度分布の近似  $\hat{p}(X|C_i)$  を行う際，ガウシアン関数を用いた方が高い認識精度が得られるように思われる．しかし，一様関数を用いても十分高い認識精度が得られること

がすでに議論されている<sup>17)</sup>．これは， $\hat{p}(X|C_i)$  を基本とするいわゆる統計的（ベイズ統計）パターン認識では， $\hat{p}(X|C_i)$  の値の大小関係から  $X$  のカテゴリを推定するためである．すなわち，各カテゴリの確率密度分布の“形状”を正確に近似することよりも，確率密度分布どうしの大小関係が入れ替わる“交点”を正確に近似することが必要であり，交点を正確に近似する点においては一様関数を用いても十分高い近似精度を得ることができるからである（一様分布関数による参考実験を行い，結果を「付録」に記したので参照されたい）．

また，Minchin らは現実の認識問題を対象にガウシアン関数と一様関数を用いた PNN の認識精度の比較実験を行っている<sup>18)</sup>．その報告では，特徴抽出を行った石の表面画像（9 次元ベクトル）を 4 カテゴリに分類する認識問題において，ガウシアン関数を用いた場合と一様関数を用いた場合の認識率を比較し，ガウシアン関数を用いた場合の認識率は 97.0% で，一様関数を用いた場合は 94.2% であったと報告している．また，プリキ製の 5 種類の形状（円形，三角形，四角形，五角形，六角形）の鐘を鳴らしたときの音響周波数スペクトル（48 次元ベクトル）を対象とした認識問題でも比較実験を行い，ガウシアン関数と一様関数でそれぞれ認識率が，95.5%，および 94.0% であったと報告している．この実験結果では一様関数の方が認識率が幾分（1.5～2.8%）低い値となったが，実用的に問題はないことが示されている．さらに，高島らは，一様関数を用いて一般的なシーンの画像の中から顔を検出するシステムを試作し，その有効性を示している<sup>19)</sup>．以上のような報告や実験結果から本論文では，一様関数を用いても十分実用的な認識システムを構築できるものと考えた．

図 4 において，外部メモリから読み込まれたカテゴリ  $C_i$  の第  $j$  番目のサンプルパターンの第 1 番目の要素  $(s_j^i)_1$  は図中の“Kernel Size Adjuster”で  $\sigma_m/2$  の加算と減算を行い，各演算結果を出力する．次に“Kernel Size Comparator”で，この各値， $(s_j^i)_1 + \sigma_m/2$  および， $(s_j^i)_1 - \sigma_m/2$  と未知入力パターン  $X$  の第 1 番目の要素  $x_1$  との値の大小関係を比較する（これが式 (1) に示す一様関数の  $|x_k - (s_j^i)_k| \leq \sigma/2$  の比較に対応する）．“Kernel Size Comparator”は， $((s_j^i)_1 - \sigma_m/2) \leq x_1 \leq ((s_j^i)_1 + \sigma_m/2)$  であった場合に 1 を出力し， $x_1$  がこの範囲にないときは 0 を出力する．

この演算をサンプルパターン  $S_j^i$  のすべての要素に対して行う．この出力を後段のフリップ・フロップで監

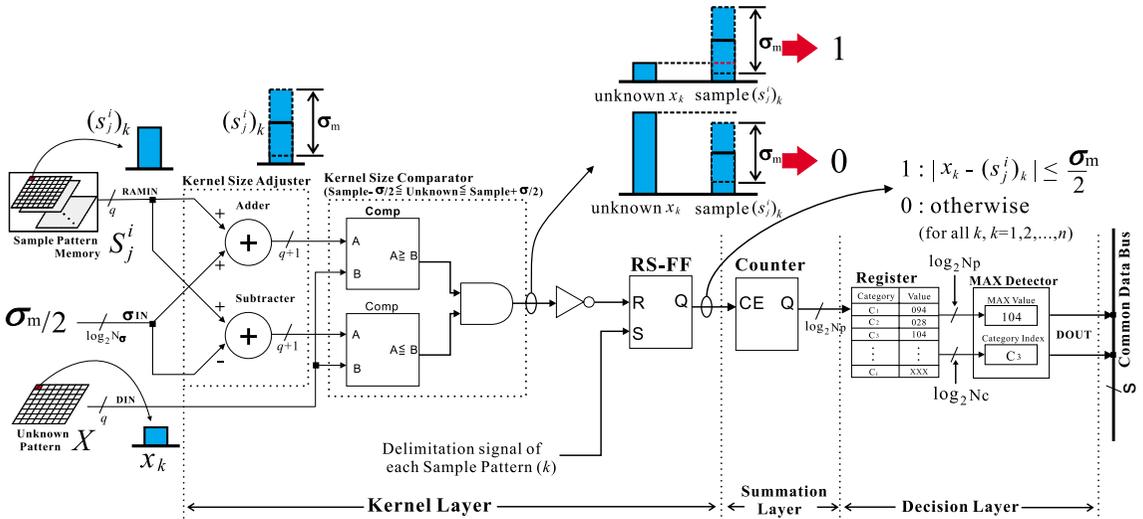


図 4 sPNN Processor の回路構成  
Fig. 4 Circuit configuration of sPNN Processor.

視し、すべての要素において演算結果が 1 となったとき、つまり未知入力カーネルに入った場合 (式 (1) の右辺 1) の数をさらに後段の加算層 (Summation Layer) にあるカウンタでカウントする。

加算層の出力値は決定層 (Decision Layer) でカテゴリごとにレジスタ (Register) にスタックされる。次に最大値検出器 (Max Detector) でそのレジスタにスタックされた値のうち、最も大きな値と、そのカテゴリが検出される。ここで検出されたカテゴリが認識結果である。

実際に、1 個の sPNN Processor を Xilinx 社製の FPGA (XCV1000E-6HQ240C) にインプリメントしたところ、使用した SLICE 数 (1 SLICE は約 100 ゲートと換算できる) は 120 SLICE であった。XCV1000E は 12,288 SLICE を有する<sup>20)</sup> ので、この規模の FPGA でも 64 個以上の sPNN Processor が他の制御回路などを含め実装できると見積もれた。したがって ASIC などの専用集積回路を用いなくても十分実用的なシステムを実現することができる。sPNN Processor はこのようにカーネル関数を簡単な組合せ回路で直接実現しているため、逐次処理でありながらも結果として後述するような高速処理を実現している。

4.2 SPA の全体回路構成

図 5 に現在開発中の小型画像認識システム (5 章で詳細を述べる) で用いている Hybrid-SPA に基づいた PNN ハードウェアの全体回路構成を示す。全体は 2 つの FPGA (Xilinx 社製 XCV1000E-6HQ240C および、XCV300E-6PQ240C) と、サンプルパターン

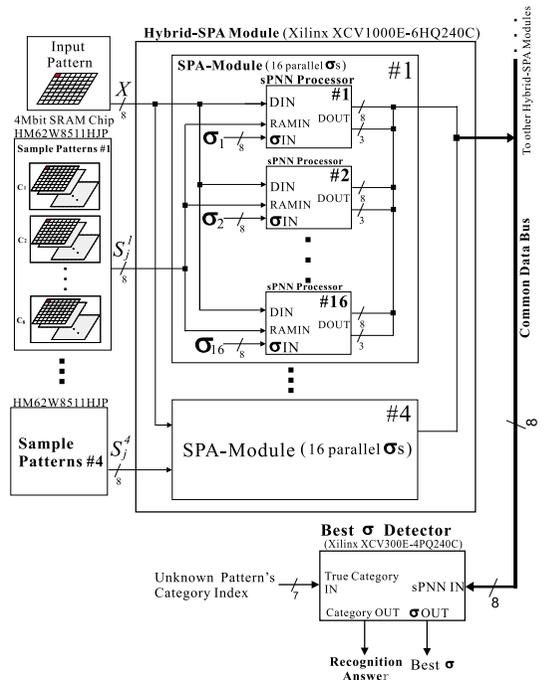


図 5 Hybrid-SPA に基づくシステムの回路構成  
Fig. 5 System configuration using the Hybrid-SPA.

を記憶しておくメモリ LSI 群 (日立製 4M-bit SRAM HM62W8511HJP) から構成されている。中心となる FPGA (XCV1000E) には、図 3 の Hybrid-SPA に基づいて構成された 64 個の sPNN Processor が実装される。この 64 個の sPNN Processor は 4 個の SPA-Module から成り (4 ニューロン並列)、さらに 1 つの

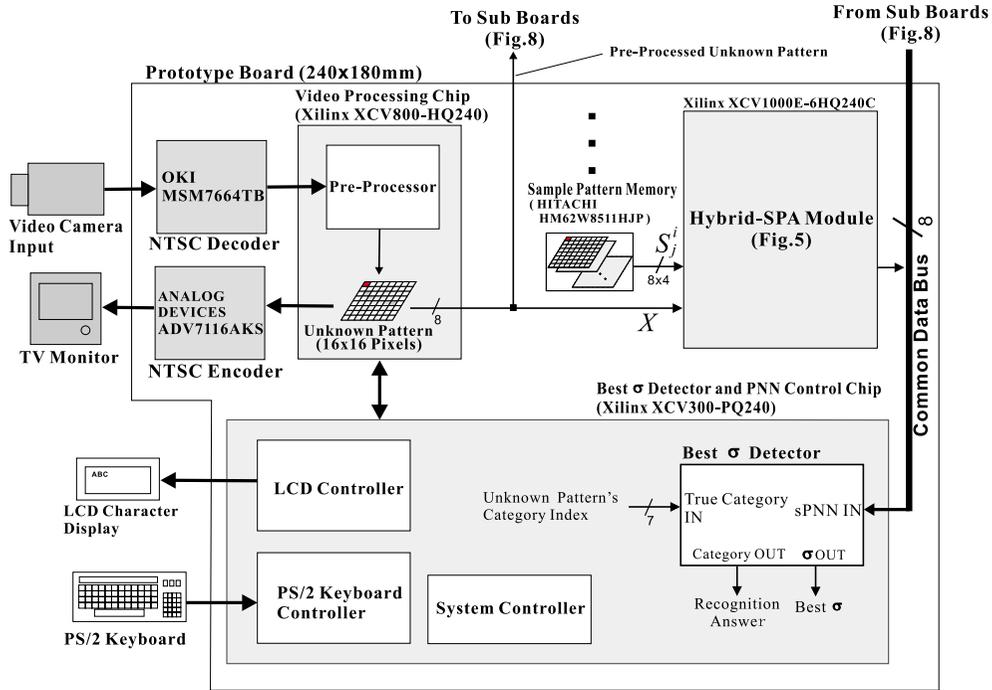


図 6 小型高速画像認識システムの構成

Fig. 6 Configuration of image recognition system.

SPA-Module は 16 個の sPNN Processor から構成される (16 $\sigma$  並列)。

これら 64 個の sPNN Processor の出力 (各 Processor の認識結果) は共有データバス (図 5 中の Common Data Bus) を介してもう 1 つの FPGA (XCV300E) に入力される。この FPGA には最も高い精度を得る  $\sigma$  を求める “Best  $\sigma$  Detector” が実装されており、これを学習時に使用する。Best  $\sigma$  Detector は認識処理においても、各 sPNN Processor の認識結果の中からさらに最終的な認識結果を得るための最大値検出器として働く。すなわち、複数の sPNN Processor の認識結果は、8 bit の共有データバスに結合された Best  $\sigma$  Detector へ逐次転送され、ここでさらに最大値を得たカテゴリが検出される。ここで得られたカテゴリが最終的な認識結果である。

なお、sPNN Processor と Best  $\sigma$  Detector をバスで接続することにより、sPNN Processor を容易に増設できる (次章に詳細を述べる)。したがって、アプリケーションごとに要求数が異なるカテゴリ数やサンプル数、あるいは要求される学習時間に応じて、sPNN Processor を増設することができる。

## 5. 試作システムの評価と考察

### 5.1 試作システム

我々は図 5 に示す構成をベースに、これに画像前処理回路を加えた小型な高速画像認識システムを開発中である。ここで “システム” とは入力画像の特徴抽出から認識処理、学習処理までをすべて実行できる装置を指す。また、“小型” とは 1 つの基板上にこのシステムを実装し、ホストコンピュータ (パーソナルコンピュータによる制御など) を必要としないという意味である。このような小型な高速画像認識システムは、自動車やロボットなど、自律移動体の制御や、工場の生産ラインなどで強く望まれており、本開発システムはこれらの用途をターゲットにしている。

図 6 にシステムの全体構成図を示す (図中の Hybrid-SPA Module は図 5 を参照のこと)。また、表 1 にその諸元を、図 7 に試作システムの写真を示す。試作システムは sPNN Processor を実装する FPGA として前述した XCV1000E の代わりに XCS30XL-4PQ240C (約 30,000 ゲート規模) を用いている点以外は、現在開発中の最終システムとほぼ同等である。なお、XCS30XL には 3 個の sPNN Processor が実装可能である。

図 6 に示すように、ビデオカメラからの入力信号

表 1 試作システムの諸元  
Table 1 Specification of prototype system.

基本動作周波数	27 MHz (sPNN Processor:13.5 MHz)
プロセッサ数	3 sPNN Processors (Max)
メモリ容量	48 Mbit (4 Mbit×12)(Max)
メモリバス幅	96 bit (8 bit×12)(Max)
基板サイズ	240×180 mm (4 層基板)
インタフェース	NTSC コンポジット入出力, PS/2 キーボード, ATA, Analog-RGB 出力, RS-232C

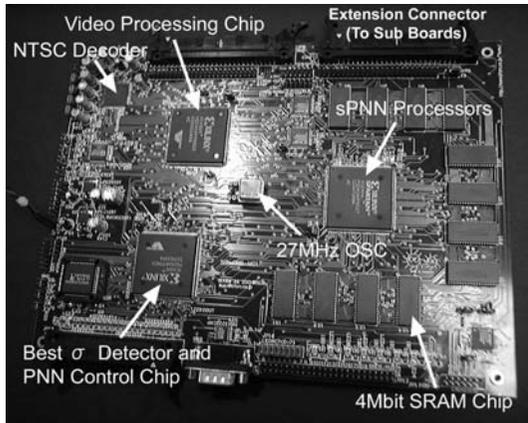


図 7 試作システムの写真  
Fig. 7 Photograph of prototype system.

は NTSC デコーダ ( OKI 製 MSM7674 ) で A/D 変換され, 前処理用の FPGA ( Xilinx 社製 XCV800-4HQ240C ) に入力される. 前処理では特定の固定領域 ( 今回は画面の中心付近とした ) を  $16 \times 16$  セグメントにモザイク化 ( 複数画素の平均化 ) することで情報量の低減を行っており, 今回はこれを特徴抽出処理とした. 前処理の結果は同じく XCV800-4HQ240C に接続された NTSC エンコーダ ( AnalogDevices 製 ADV7176AKS ) によりテレビモニタに表示して確認することができる.  $16 \times 16$  セグメントにモザイク化された未知入力画像は sPNN Processors に送られる. sPNN Processors では, サンプルパターンと比較を行い, その結果から最も正しいと推測される未知入力パターンのカテゴリ・インデックス番号が出力される. サンプルパターンを記憶しておくためのメモリとして日立製の 4Mbit-SRAM ( HM62W8511HJP-12 ) を 12 個搭載しており, メモリの各データバスはすべて独立に XCS30XL-4PQ240C と接続されている ( メモリの各データバス幅は  $8 \text{ bit} \times 12$  で合計 96 bit である ). 前述の Best  $\sigma$  Detector と全体の制御を行う回路は Xilinx 社製 XCV300-4PQ240C に実装している.

また SPA の各プロセッサ間接続ネットワークはバス接続形態であるため, 図 8 に示すように Main Board ( 図 6 ) にプロセッサ増設用拡張基板 ( 図 8 中 Sub Board ) を増設し, 容易に sPNN Processor 数 ( Hybrid-SPA Module ) を増設することができる. Main Board と Sub Board を結び主な信号線は, 特徴抽出後の未知パターンをブロードキャストする信号線 ( 図 8 中の Pre-Processed Unknown Pattern Input で指した信号線 ) と, 各 Hybrid-SPA Module の出力を受ける共有データバス ( 図中の Common Data Bus ) の 2 つのみである. これにより, システム全体のサイズは多少大きくなるものの, 認識および学習処理速度の高速化, パターン規模 ( 次元数, サンプル数, カテゴリ数など ) の拡大に対応することができる.

現在開発中の Hybrid-SPA Module に使用している XCV1000E は, 1 チップあたり約 64 個の sPNN Processor を実装でき, プロセッサ増設用拡張基板は 1 基板あたり 2 つの XCV1000E を実装する予定であるため, システムに増設する sPNN Processor は 128 Processor 単位となる. たとえば増設する XCV1000E を 3, 5, 7 個とすると, sPNN Processor の総数はそれぞれ 192, 320, 448 Processor となる. また, 次節で述べるようにプロセッサの増加にともない, 共有データバスで生じる逐次処理によるオーバーヘッドは PNN 全体の処理に比べきわめて小さいため, 学習速度もプロセッサ数に対し, ほぼ線形に増加する.

## 5.2 評価結果

### 5.2.1 認識速度および学習速度

NTSC ビデオカメラから取り込んだ 1 パターン (  $16 \times 16 \text{ segment} \times 8 \text{ bit}$  のモザイク画像:  $n = 256$ ,  $q = 8$  ) について, 1 つの sPNN Processor でカーネル関数の演算が完了するまでに要する時間を測定した. 測定結果は  $18.96 \mu\text{s}$  であり, これからビデオレート時間 (  $33.3 \text{ ms}$  ) では約 1,750 パターンを処理できることが分かった. これより 1 カテゴリあたり 256 サンプルパターン (  $N_p = 256$  ) とすると, 1 つの sPNN Processor で 6~7 カテゴリ分 (  $N_c = 6 \sim 7$  ) の処理をビデオレート時間内に実行することができることが分かった.

学習時間の実測は 64 個の sPNN Processor を実装した XCV1000E (  $16\sigma$  並列かつ, 4 ニューロン並列の Hybrid-SPA 構成 ) で行う予定であるが, 上述した演算測定結果と以下の算定式より見積もることができる. 学習時間  $T_{\text{learn}}$  は,

## Sub Board

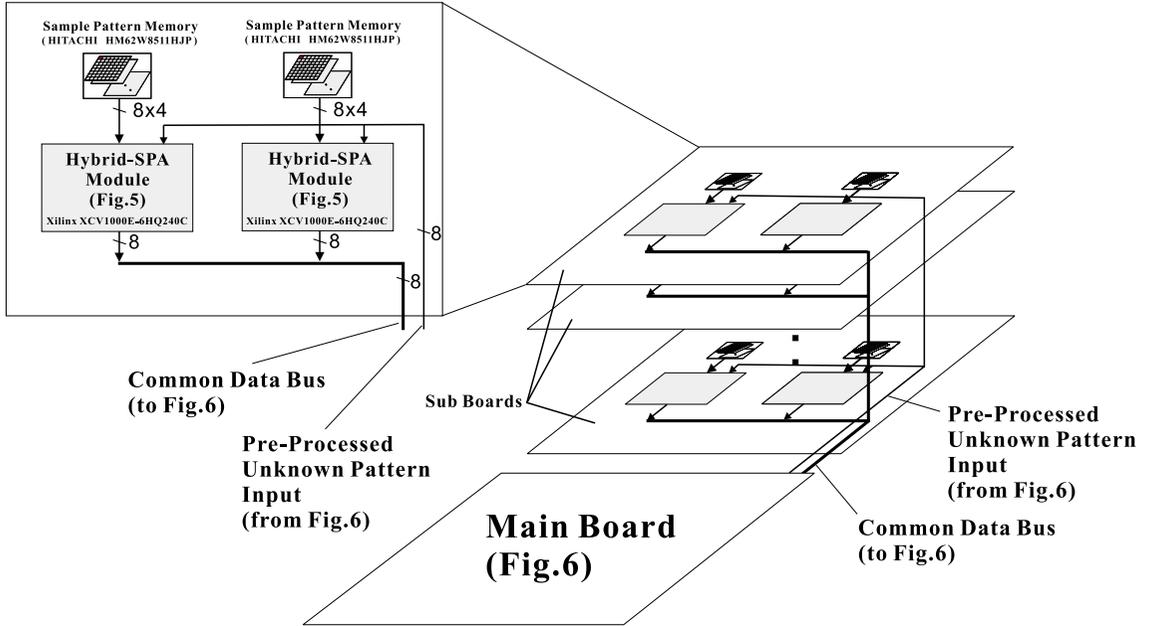


図 8 sPNN Processor 増設時の構成

Fig. 8 Configuration of the extended system using Sub Boards.

$$T_{learn} = \frac{(T_{recog.} + \beta N_{sPNN} N_{Mod.} T_{CLK}) N_{tp} N_{\sigma}}{N_{sPNN}} \quad (3)$$

と算出できる。ただしここで、

- $n$ : パターンの次元数,
- $q$ : 各要素のビット精度,
- $r$ : メモリと sPNN Processor 間のビット幅,
- $s$ : 共有バスのビット幅,
- $N_c$ : カテゴリ数,
- $N_p$ : 1 カテゴリのサンプルパターン数,
- $N_{tp}$ : テストパターン数,
- $N_{\sigma}$ : 探索する  $\sigma$  の総数,
- $N_{sPNN}$ : 1SPA-Module 中の sPNN Processor 数,
- $N_{Mod.}$ : SPA-Module 数,
- $\alpha$ : sPNN Processor 内の決定層で生じるオーバーヘッド,
- $\gamma$ : Best  $\sigma$  Detector 内の決定層で生じるオーバーヘッド,
- $T_{CLK}$ : 基本動作周期

である。また  $T_{recog.}$  は認識時間を表し、

$$T_{recog.} = \left(1 + \frac{qnN_pN_c}{rN_{Mod.}} + \alpha N_c + \beta N_{Mod.}\right) T_{CLK} \quad (4)$$

である。また、

$$\beta = \frac{\log_2 N_p + \log_2 N_c}{s} + \gamma \quad (5)$$

で、これは図 5 に示した各 sPNN Processor と Best  $\sigma$  Detector を結ぶ共有データバス (図 5 中の Common Data Bus) で、データの逐次転送を行うことによって生じるオーバーヘッド・サイクル数である。なお、ここで扱っている “SPA-Module” とは図 5 に示すように、複数の sPNN Processor のまとまりで、“Hybrid-SPA Module” はさらにこの “SPA-Module” をまとめたものである。

認識時間  $T_{recog.}$  の第 1 項目は、パイプライン処理の始めのパターンデータの読み込みである 1 サイクルを表し、第 2 項目の分子  $qnN_pN_c$  は基本動作周期で処理できるデータの総数を表し、分母の  $rN_{Mod.}$  はメモリの総バス幅を表している。ここで、 $n = 256$ ,  $q = r = s = 8$  bit,  $N_c = 1$ ,  $N_p = 256$ ,  $N_{Mod.} = 1$ ,  $\alpha = \gamma = 0$ ,  $T_{CLK} = 1/13.5$  MHz とすると  $T_{recog.} = 4.85$  ms で、これは実際に試作システムで測定した結果と等しく、この推定が正しいことを示している。学習

時間の推定式(式(3))の $\beta$ の項は、すべてのsPNN Processorの出力をBest  $\sigma$  Detectorに逐次転送して最大値検出を行うためのオーバーヘッド・サイクル数を表している。式(3)から、最終システムにおける学習時間は $N_c = 8$ で約2.5sec,  $N_c = 32$ で約10.0secと算出される。これにより、大規模なパターン認識の問題も十分高速に(遅くとも1分以内に)終了することができる。

参考比較のために、同様の学習処理をIntel製PentiumIII-1GHzのPC上でgcc(ANSI C)で作成したソフトウェア・プログラムで実行したところ、処理時間は $N_c = 8$ で約170sec,  $N_c = 32$ で約682secであった。この結果からSPAに基づいて開発したハードウェアがPC上で処理される場合に比べ、1チップ(図5に示した1つのHybrid-SPA Module)あたり約68倍高速であることが導かれる。

なお、64個のsPNN Processorを実装するXCV1000Eの回路規模は1,569,178 System Gates<sup>20)</sup>であり、これを現在最上位クラスの集積規模のXCV3200E(4,074,387 System Gates<sup>20)</sup>)などに変更することで約128~160個のsPNN Processorを実装できる。ここで、式(3)、(4)中の $\beta$ の項、つまり共有データベースで生じるオーバーヘッドは、処理全体の計算量に対し十分に小さい。これにより、プロセッサ数に対する性能の向上比、つまりスケラビリティは数百sPNN Processor程度では、ほぼ線形増加となる。したがって、使用するFPGAをXCV1000EからXCV3200Eに変更することで、PCの約136~170倍の性能向上が可能と考えられる。

また一例として、各々に2つのXCV1000Eを実装したSub Boardを3枚(64sPNN Processors $\times$ 7 Hybrid-SPA Modules)を用いた場合、sPNN Processorの総数は448で、このときも上述した式(3)、(4)と同様の議論から、PCの約480倍の学習速度を実現可能と推定できる。

今後さらに精度を上げて評価を行う必要があるが、本評価結果により、前述した気象画像認識の問題に適用した場合、入力スループットに追従してオンライン学習が実現できると考えられる。また、使用するFPGAの変更、あるいはSub Boardの増設による、さらなる高速化によりミリ秒オーダーの学習が実現できれば、移動ロボットの動的環境適応などにも適用可能になると考えられる。

### 5.2.2 認識精度

試作システムを用いて、白熱灯(250W 八口ゲンランプを使用)の下で、ビデオカメラの前に任意に提示

された手の動画像から、“ジャンケン”に関する3つのカテゴリ( $C_1$ :グー,  $C_2$ :チョキ,  $C_3$ :パーとした)を認識する実験を行った。手の形状認識のように画面内で位置ずれを生じやすい認識問題の場合、今回使用しているモザイク化のような特徴抽出手法ではなく、より位置ずれに対してロバストな特徴抽出を行う必要があると考えている。しかし、今回の実験は認識回路の動作を確認するために行ったため、容易に入手可能なパターンとして“ジャンケン”画像を選択し、そのモザイク画像を特徴抽出後の認識対象とした。なお、開発中の小型画像認識システムでは、特徴抽出を行うプロセッサもFPGA(Field Programmable Gate Array)を用いているため、認識問題に応じ、適切な特徴抽出を動的に選択することも可能となっている。

現在、学習回路は未実装であるため、使用した $\sigma$ の最適値は別途パーソナルコンピュータにより探索範囲 $\sigma = 0 \sim 256$ ( $N_\sigma = 256$ )で、全探索を行うことにより求めた。最適値は実験ごとにサンプルパターンを更新したため、そのつどパーソナルコンピュータにより再計算を行い、 $\sigma = 80 \sim 90$ を用いた。実験システムの写真を図9に、実験で用いた各カテゴリのサンプルパターンの例を図10に示す。図9中で「Unknown Pattern Field」に提示された未知パターン(今回は特定方向からの手の形状)を「Video Camera」から本認識システムである「Prototype Board」に入力した。カメラにはSONY製の家庭用ビデオカメラDCR-PC110を用いた。画面内の手の位置合わせは「Video Monitors」を見ながら行い、同時に前処理が正しく行われていることを確認した。図10に示すように、画面の中央に16 $\times$ 16セグメントにモザイク化された固定領域があり、この領域に入った手の形状が認識される。各パターンは $q = 8$ ,  $n = 256$ ,  $N_p = N_{tp} = 256$ ,  $N_c = 3$ であり、また、 $N_{sPNN} = N_{Mod.} = 1$ ,  $r = s = 8$ である。実験の結果、画面の中央に特定の向きから提示した、同1人物の同じ手のグー、チョキ、パーの形状を80%以上(10回の提示を連続して行い、毎回8回以上正しく認識できる)の認識率で認識することができた。

## 6. おわりに

本論文では、確率的ニューラルネットワークの認識と学習の高速化のために、カーネル関数を単純な組合せ回路で実現する専用プロセッサ(sPNN Processor)と、異なる学習パラメータを並列に処理する学習アーキテクチャ(SPA)を提案した。このプロセッサとアーキテクチャを用いた試作システムの性能を、ジャンケ

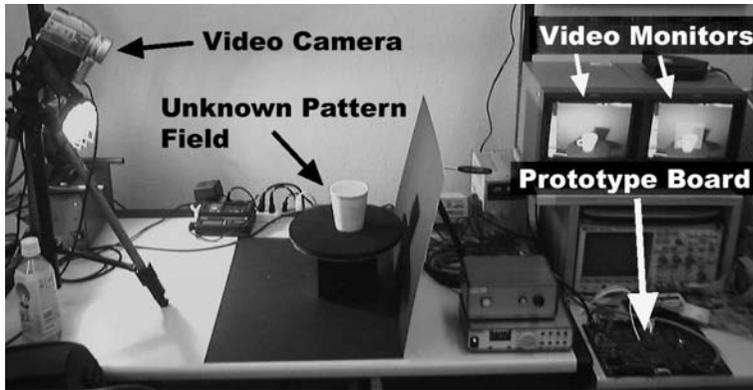


図 9 認識実験システム

Fig.9 Recognition test system.

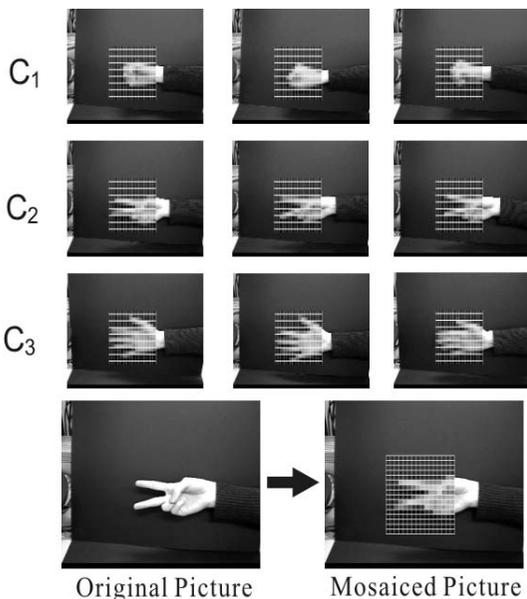


図 10 認識実験に用いた画像パターン

Fig.10 Sample patterns for recognition test.

ン画像を用いた認識実験により評価した結果、認識精度は 80%以上、かつビデオレート時間 (33.3 ms) 以内で処理できることが分かった。また学習においても、256 次元/サンプル、256 サンプル/カテゴリ、32 カテゴリの規模のパターンを約 10.0 秒で処理でき、より大規模なパターンにおける学習処理も十分実用的な時間内で行うことができることが分かった。これにより、提案したプロセッサとアーキテクチャにより、実应用到可能な小型高速画像認識システムを実現できる見通しを得た。今後は最終システムを完成させ、さらに周囲の環境に合わせて動的に学習を行う手法の検討

も行う予定である。

謝辞 本論文の執筆にあたり有益な助言をいただいた宮崎大学工学部情報システム工学科の吉原郁夫教授に感謝する。また、本研究の一部は 2001 年度日本学術振興会科学研究補助金 No.13450163 による。

#### 参 考 文 献

- 1) Specht, D.: Probabilistic Neural Networks, *J. Neural Networks*, Vol.3, pp.109–118 (1990).
- 2) Specht, D.: Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification, *IEEE Trans. Neural Networks*, Vol.1, No.1, pp.110–111 (1990).
- 3) Müller, Klaus-R., Mika, S., Rätsch, G., Tsuda, K. and Schölkopf, B.: An Introduction to Kernel-Based Learning Algorithms, *IEEE Trans. Neural Networks*, Vol.12, No.2, pp.181–201 (2001).
- 4) Ruiz, A. and López-de Teruel, P.E.: Nonlinear Kernel-Based Statistical Pattern Analysis, *IEEE Trans. Neural Networks*, Vol.12, No.1, pp.16–32 (2001).
- 5) Mao, K.Z., Tan, K.C. and Ser, W.: Probabilistic Neural-Network Structure Determination for Pattern Classification, *IEEE Trans. Neural Networks*, Vol.11, No.4, pp.1009–1016 (2000).
- 6) Tian, B., Azimi-Sadjadi, M.R., Vonder Harr, T.H. and Reinke, D.: Temporal Updating Scheme for Probabilistic Neural Network with Application on Satellite Cloud Classification, *IEEE Trans. Neural Networks*, Vol.11, No.4, pp.903–920 (2000).
- 7) Tian, B., Azimi-Sadjadi, M.R.: Comparison

of Two Different PNN Training Approaches for Satellite Cloud Data Classification, *IEEE Trans. Neural Networks*, Vol.12, No.1, pp.164–168 (2001).

- 8) 岩田, 雨宮: ニューラルネットワーク LSI, 電子情報通信学会 (編), オーム社 (1995).
- 9) Gotarredona, T.S., Barranco, B.L. and Andreou, A.G.: Programmable Kernel Analog VLSI Convolution Chip for Real Time Vision Processing, *Proc. IEEE Int'l Joint Conf. Neural Networks*, CD-ROM (2000).
- 10) Szabo, T., Antoni, L., Horvath, G. and Feher, B.: A full-parallel digital implementation for pre-trained NNs, *Proc. IEEE Int'l Joint Conf. Neural Networks*, CD-ROM (2000).
- 11) Girau, B. and Lorraine, L.: Digital hardware implementation of 2D compatible neural networks, *Proc. IEEE Int'l Joint Conf. Neural Networks*, CD-ROM (2000).
- 12) Yasunaga, M., Masuda, N., Yagyū, M., Asai, M., Yamada, M. and Masaki, A.: Design, Fabrication and Evaluation of A 5-inch Wafer Scale Neural Network LSI Composed of 576 Digital Neurons, *IJCNN '90*, II, pp.527–535 (1990).
- 13) 平井有三, 落合辰男, 安永守利: 1000 ニューロン 100 万シナプスで構成されたニューラルネットワークハードウェアシステム, 電子情報通信学会論文誌, J84-D-II, No.6, pp.1185–1193 (2001).
- 14) Sinencico, E.S. and Lau, C. (Eds.): *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementations*, IEEE Press (1992).
- 15) Kohonen, T.: *Chapter 8 (Hardware for SOM) in "Self-Organizing Maps"*, Vol.1, 2nd edition, chapter 2, pp.371–381, Springer (1997).
- 16) Wawrzyniek, J., Asanovic, K., Kingsbury, B., Johnson, D., Beck, J. and Morgan, N.: Spert-II: A Vector Micro Processor System, *Special Issue of Neural Computing in IEEE COMPUTER*, Vol.29, No.3, pp.79–86 (1996).
- 17) Bishop, C.: *Neural Network for Pattern Recognition*, Oxford Press (1995).
- 18) Minchin, G. and Zaknich, A.: *A Design for FPGA Implementation of the Probabilistic Neural Network*.
- 19) 高島浩二, 中村太郎, 安永守利, 吉原郁夫: 遺伝的アルゴリズムを用いた画像中の顔画像検出とその進化ハードウェア上への実装, 電子情報通信学会技報, PRMU2000-117, pp.71–78 (2000).
- 20) Xilinx, Inc.: *Virtex-E 1.8V Field Programmable Gate Arrays*, DS022-1 (v2.2) (2001).  
<http://www.xilinx.com/partinfo/ds022.pdf>

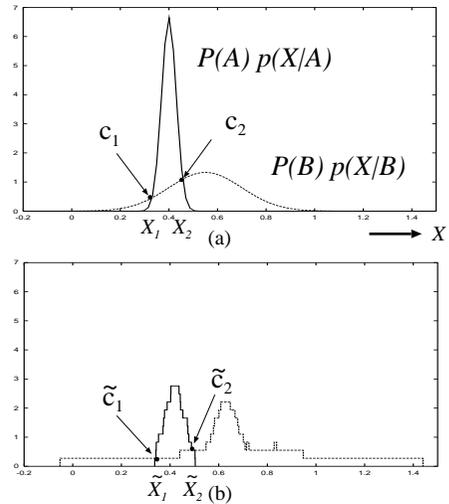


図 11 一様関数を用いた確率密度分布交点近似実験の結果  
 Fig. 11 Experimental result of cross-points approximation between the probabilistic density distributions.

## 付 録

一様関数を用いた確率密度分布近の交点近似実験  
 一様関数を用いて 2 つの確率密度分布の交点がどの程度正確に近似できるか計算機 (PC) による参考実験を行った。2 つの確率密度分布として一次元正規分布  $p(X|A)$  (平均 0.4, 分散 0.03),  $p(X|B)$  (平均 0.55, 分散 0.15) を用いた (図 11 (a))。両分布の交点  $c_1$ ,  $c_2$  の  $X$  値は, それぞれ,  $X_1 = 0.331$ ,  $X_2 = 0.457$  である。

この確率密度分布に従ってそれぞれ 10 個 (合計 20 個) のサンプル値  $S$  を発生し, この 20 個の  $S$  に対する一様関数 (式 (1)) の重ね合わせを行った。結果を図 11 (b) に示す。両近似関数の交点  $\tilde{c}_1$ ,  $\tilde{c}_2$  の  $X$  値は, それぞれ,  $\tilde{X}_1 = 0.337$ ,  $\tilde{X}_2 = 0.488$  となった。使用したサンプル数がわずか 20 個と少ないため, 近似関数の形状は正規分布の形状をよく近似しているとはいいがたい。しかし, わずか 20 個のサンプルにもかかわらず, 両交点を非常に高い精度で近似している。この実験結果からも, 一様関数をカーネル関数として用いても高い認識精度が実現できることが分かる。なお, 本実験において式 (1) の  $\sigma$  は計算時間の関係から,  $\sigma$  を広範囲にわたって変化させる代わりに, 20 個のそれぞれの一様関数において  $\sigma = 0.5, 0.05, 0.005$  の中から交点を実際の交点に最も近くなった組合せを選んだ。

(平成 14 年 1 月 29 日受付)

(平成 14 年 5 月 16 日採録)



相部 範之(学生会員)

平成 11 年工学院大学工学部電子工学科電子工学コース卒業。同年，理化学研究所脳科学総合研究センター入所。平成 12 年同研究所退所後，筑波大学大学院修士課程理工学研究科理工学専攻入学。平成 14 年同大学院修士課程修了。現在，同大学院博士課程システム情報工学研究科コンピュータ・サイエンス専攻に在学中。パターン認識，画像処理，再構成型ハードウェアの研究に従事。



安永 守利(正会員)

昭和 56 年筑波大学第三学群基礎工学類卒業。昭和 58 年同大学院工学研究科修士課程修了。同年(株)日立製作所入社。同社中央研究所において大型計算機ハードウェア，ウェーハスケール集積回路の研究に従事。昭和 63 年より，ニューラルネットワーク，記憶ベース推論，遺伝的アルゴリズムの研究を開始。平成 3~4 年米国カーネギーメロン大学客員研究員を兼任。平成 6 年筑波大学電子・情報工学系助教授。現在，進化ハードウェア，ニューロハードウェア，並列計算機の研究に従事。博士(工学)。電子情報通信学会，IEEE，INNS(国際神経回路学会)各会員。