

ΔΣ 変調に基づくパルスニューラルネットワークによる BP 学習の実現とそのハードウェア実装

奥田 忠義[†] 道木 慎二^{††} 石田 宗秋[†]

Realization of Back Propagation Learning for Pulsed Neural Networks Based on Delta-Sigma Modulation and Its Hardware Implementation

Tadayoshi OKUDA[†], Shinji DOKI^{††}, and Muneaki ISHIDA[†]

あらまし ニューラルネットワーク本来の並列処理をいかした高速処理を実現するためには、個々のニューロンをハードウェアに実装し並列動作させることが肝要である。ニューラルネットワークのハードウェア実現手段として、筆者らもこれまでに実装に適した“ΔΣ 変調に基づくパルスニューラルネットワーク（以下、DSM-PNN）”を提案してきた。DSM-PNN はニューロン間の信号伝達が 1 本で済み、かつパルス生成法が ΔΣ 変調に基づくため、小規模で高精度な信号処理が可能である。これまでに、線形ニューロンを用いた学習に対しての有効性が確認されている。本論文では、ニューロンの非線形演算を伴う BP 学習を実現する DSM-PNN について検討を行う。非線形な活性化関数については、ΔΣ 変調に基づき構成する手法を提案する。更に、提案する活性化関数を用いて DSM-PNN をハードウェア上に構成し、BP 学習を実現することで有効性を示す。

キーワード ΔΣ 変調, パルスニューラルネットワーク, ハードウェア実装, BP 学習

1. ま え が き

近年のノイマン型計算機の発展は、大容量の情報をより高速に扱うことを可能としてきた。しかし、我々の日常生活に頻繁に表れる文字や人の顔の認識といった処理においては、いまだ人間の能力には大きく及ばない。ニューラルネットワークは、人の情報処理装置である脳のもつ日常的情報処理能力の高さに着目し、工学的見地から脳の動作を数式モデルとして実現したものであり、認識、制御、予測などをはじめ幅広く応用されている。しかしながら、ニューラルネットワークの実装形態を見ると多くはソフトウェア上での直列演算により実装されるのが通例である。ニューロンの計算を逐次処理によって行うこの方法では、大規模なニューラルネットワークを実現しようとすると、演算時間は規模に応じて増大してしまう。人間の脳には数

百億個のニューロンが存在するといわれ、それらが非同期かつ並列に動作することにより高速な情報処理を実現している。このような脳の並列性に起因する高速処理を実現するためには、ニューラルネットワークをハードウェア上に実装し並列演算することが望まれており、これまでに様々なニューラルネットワークモデルが提案されている [1] ~ [4]。

大別すると電圧電流の振幅波形を用いて情報を伝送するアナログ方式と情報を何らかの形でデジタル化した上で扱うデジタル方式に大別することができるが、近年の FPGA などの書換え可能なデバイスの性能の向上やハードウェア記述言語等の環境ツールの整備により、低コストでかつ短期間でデジタル回路設計を行うことが可能になっており、このような半導体技術の恩恵を受けるためにはデジタル回路での実装が望ましいと我々は考えている。デジタル方式では、ニューロンの信号処理を模擬するデジタル回路によるプロセッサエレメント (PE) を複数用意し、並列に動作させる。ここで注意しなければならないのは、ハードウェア実装においては、ソフトウェア実装と異なり、利用可能なデバイスの面積に制限が存在し、PE

[†] 三重大学, 津市
Mie University, 1515 Kamihama-cho, Tsu-shi, 514-8507 Japan

^{††} 名古屋大学, 名古屋市
Nagoya University, Furo-cho, Nagoya-shi, 464-8603 Japan

間の相互結合のために物理的な領域（配線領域）を消費するということである。そして PE 間を結ぶデジタル信号の配線領域は、ネットワークの規模の増加に伴い、PE の領域に比して著しく増大し、集積度の限界を生じることが予想される。バスを用意し、時分割により信号伝送を行うことにより、配線領域を削減する方式 [5] も提案されているが、引換えに真の並列動作を失う。

これらの問題を解決する手段として、生物のパルスによる信号処理を模倣し、ニューロン間の信号結線を 1 本としたパルスニューロンモデルがある。パルスニューロンモデルでは、ニューロン間の信号結線が 1 本となるため、ハードウェア実装した際、配線領域や回路規模の削減が期待できる。アナログ回路方式の場合、ニューロンが扱う信号（情報）をそのまま信号線上のアナログ電圧・電流振幅とすればよいわけであるが、本論文のようにデジタル回路方式を選択する場合、信号（情報）を、1 本の信号線上の 1 ビットのデジタル値（H, L の 2 値）により、どのような形で伝えるかにより、様々な手法（パルス密度変調：PWM, パルス周波数変調：PFM, パルス位相変調：PPM, パルス密度変調：PDM, パルス符号変調：PCM 等）が提案されている [1], [6]。

筆者らはパルスニューロンモデルにおいて、そのパルスの生成法や演算を $\Delta\Sigma$ 変調に基づき処理する“ $\Delta\Sigma$ 変調に基づくパルスニューラルネットワーク（以下、DSM-PNN）”を提案してきた [7]。 $\Delta\Sigma$ 変調は高精度なパルス密度信号へ変換するデジタル信号処理技術であり、ニューラルネットワークを信号処理的な観点から解析、設計することができる。

これまでに、GHA (Generalized Hebbian Algorithm) 学習則 [8] をもつ DSM-PNN が FPGA 上に実装し評価されている [9]。GHA 学習則は、Hebb 則における重みの発散という問題点を回避するために制約項を導入した Oja 則の多次元拡張版であり、主成分分析 (PCA) をニューラルネットワークの学習という形で行うものであり、四則演算のみで実現することが可能である。この結果から、DSM-PNN が精度良く四則演算を行うことが確認されている。

本論文では、階層型ニューラルネットワークの代表的学習則である BP 学習を実現するための DSM-PNN について検討を行う。BP 学習は、その適用範囲の広さや応用事例の多さからハードウェア実装の有用性は大きくこれまでも様々な実現方法 [10] ~ [12] が提案

されてきた。BP 学習では、活性化関数やその導関数といった非線形処理を行う演算器が必要となるため、非線形演算器の実装は、ハードウェア実装のポイントの一つである。

デジタル方式のパルスニューロンにおいては、一般的に積を AND、和を OR とみなしたり、活性化関数をパルスの飽和特性で代用する [13] ~ [16] などの方法がとられてきた。また、学習回路においては、学習アルゴリズムの見直し、活性化関数の導関数の近似などの方法が検討されてきた [10] ~ [12]。

DSM-PNN においても、これまで活性化関数やその導関数は、活性化関数や導関数の近似といった方法によって対処してきた [17] が、他方式同様、近似による演算精度の低下から生じる学習性能の低下は避けられなかった。

本論文では、精度、回路規模の面から、活性化関数とその導関数を $\Delta\Sigma$ 変調に基づき構成する手法を提案する。更に、提案する活性化関数を用いて BP 学習を行う DSM-PNN を構成し、ハードウェア実装を行いその有効性を示す。

以下、2. では、文献 [9] で提案している DSM-PNN の定義、そこで用いられる信号表現とその特徴を紹介し、その基本的演算器について触れる。3. では BP 学習を可能とする活性化関数とその導関数の実現方法を提案する。4. ではそれを用いた BP 学習則の実装方法、5. では実験により有効性を確認する。6. はまとめである。

2. $\Delta\Sigma$ 変調器と 1 ビットパルス信号

本論文で扱うデジタル方式のパルスニューラルネットワークにおいては、情報伝送に $\Delta\Sigma$ 変調されたデジタル信号のパルス列を用いる。本論文では $\Delta\Sigma$ 変調により変換された 1 ビットのパルス密度信号を単に“1 ビットパルス信号”と呼ぶ。本章では、提案に先立ち、文献 [9] に従い、 $\Delta\Sigma$ 変調器、及び 1 ビットパルス信号についての特徴と定義を行い、1 ビットパルス信号の演算事例を示す。

2.1 $\Delta\Sigma$ 変調器と 1 ビットパルス信号の特徴

$\Delta\Sigma$ 変調は、マルチビットやアナログ信号で表されている入力信号を精度良く 1 ビットのパルス密度信号に変換する符号化手法の一つである。 $\Delta\Sigma$ 変調は、構造が簡単でかつ大部分がデジタル回路で構成される点や、原理的に動作速度が直接変換精度（量子化誤差、折返し雑音）を左右する点において、現在の LSI

向きであるといえる．更に変調された 1 ビットパルス信号は低域フィルタ (LPF) を通すのみで容易にもとの波形に復調することができる． $\Delta\Sigma$ 変調のような 1 ビット方式の信号処理技術は、近年の LSI の高速化の恩恵を受けて注目されている技術であり、既に A-D、D-A 変換の手法として、また音響などの D 級アンプとして実用化されている [19] ~ [21]．

$\Delta\Sigma$ 変調の最大の特徴は、符号化に伴い必ず生じる量子化誤差が高周波領域側に偏って発生し低周波領域の量子化誤差が減少することにある． $\Delta\Sigma$ 変調特有のこの性質は、ノイズシェーピングと呼ばれ入力信号帯域 (低周波数帯域) では精度の良い演算が可能となる．

図 1 (a) に振幅 1.0, 周波数 5 kHz の正弦波をサンプリング周波数 $f_s = 1024$ kHz で $\Delta\Sigma$ 変調した際の出力信号のパワースペクトルを示す．同図 (b) には、パルスニューロンモデルにおけるパルス生成法の一つであり、乱数を用いて確率的にパルスを発生させる変調方式である Random Pulse Width Modulation (以下, RPWM) [10], [13], [14] により変調した出力信号のバ

ワースペクトルを示す．

図 1 より, RPWM の場合には $0 \sim f_s/2$ の周波数帯域に均等に量子化誤差成分が分布するのに対し, $\Delta\Sigma$ 変調はノイズシェーピングにより量子化誤差を高周波領域側に形成していることが分かる．

また, 入力信号の周波数を f とすると, 一次の $\Delta\Sigma$ 変調器の SN 比は次式で表現されることが知られている [21]．

$$SNR = 10 \log_{10} \left\{ \frac{9}{16\pi^2} (f_s/f)^3 \right\} \quad (1)$$

マルチビット信号では, 演算精度を向上するためにはビット数を増やさなければならず, それに伴いビット幅など回路を変更する必要がある．一方 1 ビットパルス信号の場合は, 式 (1) よりサンプリング周波数を上昇させることのみで精度を向上させることができる．

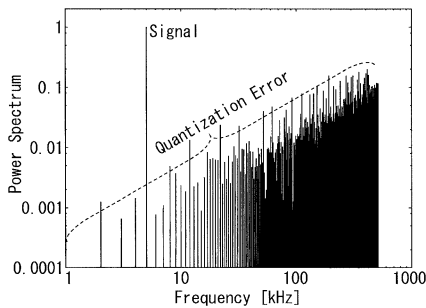
2.2 $\Delta\Sigma$ 変調器の動作原理

図 2 に一次の $\Delta\Sigma$ 変調器の構成を示す．積分器, 比較器, 加算器といった比較的簡単な回路構成により実現できる．本論文では出力が $-1, 1$ となるバイポーラ方式の $\Delta\Sigma$ 変調器を用いる．なお, 実際のハードウェア上では, 2 値デジタル回路における H レベルで $1, L$ レベルで -1 を表現することとなる．この回路の動作は式 (2), 式 (3) で表される．

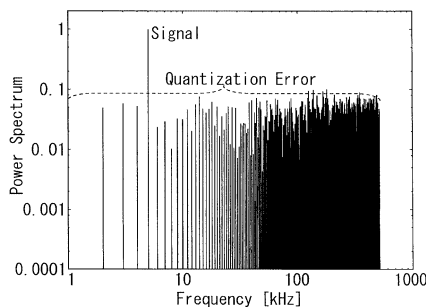
$$w(n) = \frac{z^{-1}}{1 - z^{-1}} \{x(n) - \tau y(n)\} \quad (2)$$

$$y(n) = \begin{cases} 1 & (w(n) \geq 0) \\ -1 & (w(n) < 0) \end{cases} \quad (3)$$

式 (2) において, $w(n)$ は入力信号 $x(n)$ と出力信号 $y(n)$ との差の積分, すなわち積分された量子化誤差であることを意味する．なお τ はフィードバック・ゲインで正の定数である．式 (3) において, $y(n)$ は出力信号であり 1 , または -1 のいずれかの値をとる．常

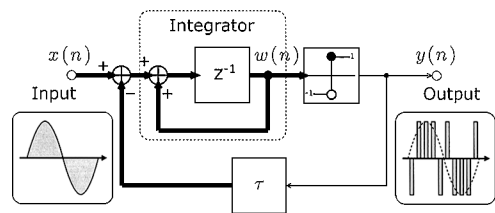


(a) Power spectrum modulated by delta-sigma modulation



(b) Power spectrum modulated by random pulse width modulation

図 1 変調信号のパワースペクトル
Fig. 1 Power spectrum of modulated signal.



Analog or Multi-bit \rightarrow 1 bit \rightarrow

図 2 $\Delta\Sigma$ 変調器の構成
Fig. 2 $\Delta\Sigma$ modulator.

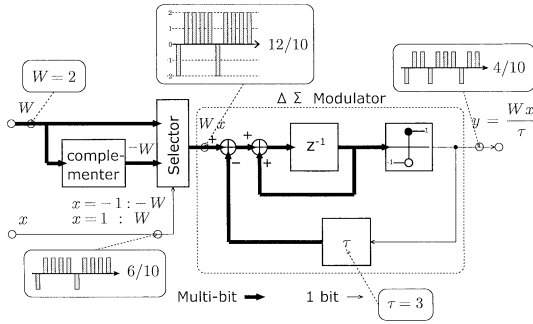


図 3 重み乗算器の構成
Fig. 3 Weight multiplier.

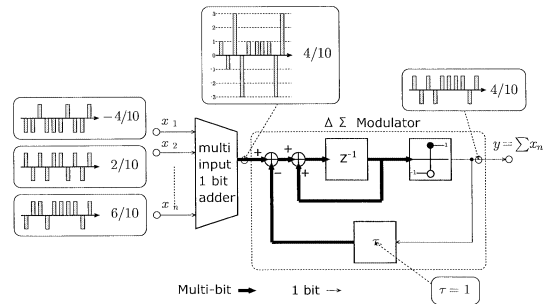


図 4 多入力加算演算器の構成
Fig. 4 Multi input 1 bit adder.

に誤差が 0 となる方向にフィードバックが働くことにより、結果的に量子化誤差の期待値は 0 となり、 $\Delta\Sigma$ 変調の出力パルス列は入力信号と等しく出力される。

2.3 1 ビットパルス信号の演算

本節では、ニューラルネットワークを $\Delta\Sigma$ 変調された 1 ビットパルス信号によって統一的な信号処理を実現するために必要な演算器を示す。ここでは、ニューロンの積和演算、1 ビット信号の乗算について説明するが、詳細は文献 [9] を参考にされたい。

2.3.1 重み乗算器の構成

ニューロンの入力信号とシナプス荷重値との乗算や、1 ビットパルス信号とマルチビット信号の乗算は、図 3 に示す $\Delta\Sigma$ 変調に基づく重み乗算器により実現する。図 3 より重み乗算器は、 $\Delta\Sigma$ 変調器、補数回路及びセレクタから構成される。1 ビットパルス信号 x はセレクタの制御信号に入力され、 x の値 1, -1 によって重み係数を W , $-W$ に切り換えることにより $\Delta\Sigma$ 変調器には Wx が入力される。そして、 $\Delta\Sigma$ 変調器で再び 1 ビットパルス信号に変換される。重み乗算器は $\Delta\Sigma$ 変調器内部のフィードバックゲイン τ と入力 W によって 1 ビットパルス信号を式 (4) のように W/τ 倍にすることができる。

$$y = \frac{W}{\tau}x \quad (4)$$

2.3.2 多入力加算器の構成

1 ビット信号同士の加算やシナプス荷重値により重み付けされたニューロンの入力信号の多次元加算に際しては、図 4 に示す多入力加算器を用いる。入力される 1 ビットパルス信号は n 入力加算部で一時的に $\log_2 n$ ビットのマルチビット信号として扱われるが、 $\Delta\Sigma$ 変調器に入力することにより 1 ビットパルス信号に再び戻される。加算を OR などの論理素子で実現す

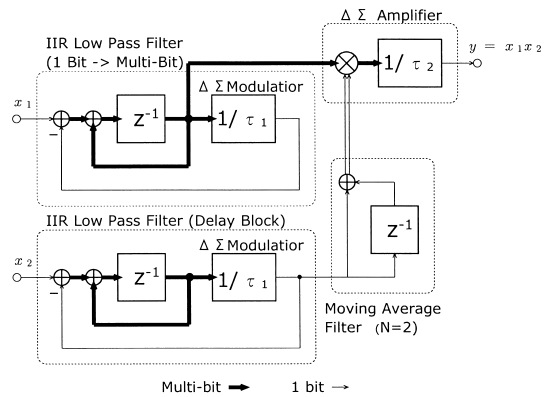


図 5 1 ビット乗算器の構成
Fig. 5 1 bit multiplier.

る方法も提案されている [10] が、パルスが疎、若しくは入力次元数が少ない場合でなければ、加算として近似することができない [13], [14]。 $\Delta\Sigma$ 変調に基づく多入力加算器では、入力次元数を増加させた場合でも誤差を生じず精度良く演算を行うことができる。

2.3.3 1 ビット乗算器の構成

1 ビットパルス信号同士の乗算を実現する手法の一つに、AND や XNOR などの論理回路の演算で実装するモデルが提案されている [10]。しかし、入力信号のパルスパターンによって演算結果がばらつき、精度の良い演算を行うことができない。本論文では、1 ビットパルス信号同士の乗算を図 5 に示す乗算器により実現する。 x_1 側から入力される信号を IIR 型 LPF により τ_1 倍のマルチビット信号 $\tau_1 x_1$ に変換し、図 3 重み乗算器に重み W として入力する。片側の信号をマルチビット信号に変換することで、マルチビット信号と 1 ビット信号との乗算の形としてとらえることができ、図 3 に示す重み乗算器により演算を行うことがで

きる．ここで，IIR 型 LPF の除算 $1/\tau_1$ は $\Delta\Sigma$ 変調器を用いて構成することにより除算の際の回路規模を小さく設計できる．もう片側の LPF は，二つの信号の位相をそろえるために接続しているが，BP 学習回路の構成により省略することも可能である．また同時にタップ数が 2 である移動平均フィルタを用いているが，オフセット誤差の原因となるナイキスト周波数成分の量子化誤差を低減するために挿入している．

3. $\Delta\Sigma$ 変調による活性化関数の実現

ニューロンの活性化関数には，2 値をとるしきい値関数やシグモイド関数等様々な関数を用いられる．BP 学習においては，活性化関数として特にシグモイド関数が一般的に使用されるが，これはシグモイド関数が微分可能な連続関数であるためである．しかしハードウェア実装を考えた場合，シグモイド関数を直接ハードウェア上に実装するためには，指数演算や乗除算を行うための演算器が必要となる．指数演算や乗除算などの演算器は一般に回路規模が大きくなり，活性化関数はハードウェア実装における問題点の一つとされている．このため，関数の入出力に対応するテーブルをハードウェア上に実装する手法などが用いられる [18]．テーブル演算により実現する場合，多くのメモリ領域を必要としそのビット幅 N に応じてメモリ領域が 2^N のオーダで増加する．また， $\Delta\Sigma$ 変調という統一した信号処理形態でニューラルネットワークを実現するという観点からは，テーブル演算は望ましくない．

本論文では，回路規模が増加に伴い発生する実装上の問題である論理ゲートの配置，配線の難しさ，信号遅延の影響等を軽減するためには，極力，同様の回路が規則正しくレイアウトすることが望ましいと考え， $\Delta\Sigma$ 変調器という統一した信号処理形態でのニューラルネットワークを目指す．そこで，本章では， $\Delta\Sigma$ 変調に基づき構成される可微分な活性化関数回路，及び BP 学習を実現する上で必要な導関数回路を提案する．

3.1 活性化関数の構成

図 6 に提案する活性化関数の構成を示す．図 6 より，活性化関数回路は $\Delta\Sigma$ 変調器にセクタ，論理素子 XOR を追加するのみで構成される．このため，テーブル等を用いて実現する場合に比べ，小規模実装が可能である． $\Delta\Sigma$ 変調器への入力信号はセクタにより制御する．活性化関数回路への入力信号の符号と， $\Delta\Sigma$ 変調器の内部レジスタの符号が同じであれば $\Delta\Sigma$ 変調器には 0 が入力され，そうでなければ入力信号は

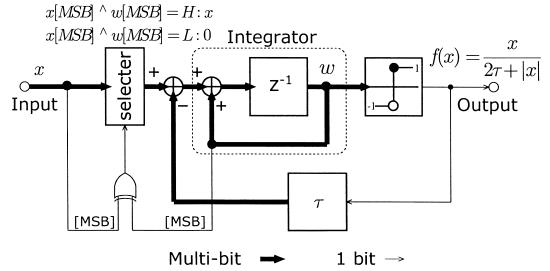


図 6 $\Delta\Sigma$ 変調に基づく活性化関数の構成
Fig. 6 Active function based on delta-sigma modulation.

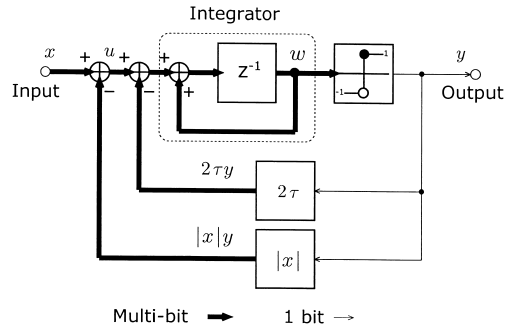


図 7 $\Delta\Sigma$ 変調に基づく活性化関数の導出回路
Fig. 7 Derivation of proposed active function.

そのまま $\Delta\Sigma$ 変調器に入力される．このようにして $\Delta\Sigma$ 変調器に外部回路を付加することにより，式 (5) (図 8) の入出力関係を得ることができる．

$$f(x) = \frac{x}{2\tau + |x|} \quad (5)$$

以下では，その回路構成 (図 6) の導出過程を示す．

通常の $\Delta\Sigma$ 変調器では，その入出力関係はフィードバックゲイン τ を利用し， $y = f(x) = x/\tau$ となる．そこで，図 7 のように入力信号 x の振幅に対応して可変となるゲイン $|x|$ のフィードバックループを追加することにより，式 (5) の関係を得ることができ，図 8 に示すような特性が実現できる．この回路動作を式で記述すると式 (6)，式 (7) 及び式 (8) で表現される． $u(n)$ は $\Delta\Sigma$ 変調器への入力信号， $w(n)$ は内部レジスタの出力信号， $y(n)$ は出力信号である．

$$u(n) = x(n) - |x(n)|y(n) \quad (6)$$

$$w(n) = \frac{z^{-1}}{1 - z^{-1}} \{u(n) - 2\tau y(n)\} \quad (7)$$

$$y(n) = \begin{cases} 1 & (w(n) \geq 0) \\ -1 & (w(n) < 0) \end{cases} \quad (8)$$

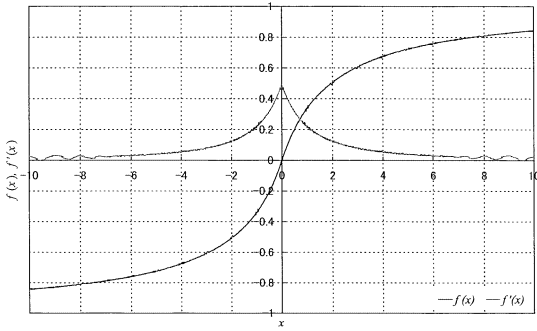


図 8 $\Delta\Sigma$ 変調に基づく活性化関数と導関数の入力出力特性
Fig. 8 Input-output characteristics of proposed active function and derived function.

表 1 $x - |x|y$ の演算結果
Table 1 Result of $x - |x|y$.

x	y	$x - x y$
Pos.	1	0
Pos.	-1	$2x$
Neg.	1	$2x$
Neg.	-1	0

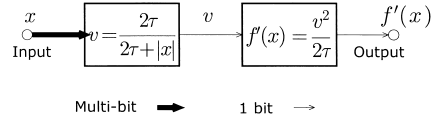
しかし、図 7 より、活性化関数回路は $\Delta\Sigma$ 変調器以外に、加算器、絶対値演算回路、重み乗算器が必要となり、単純にそのまま実装すると回路構成が大きくなる。そこで、以下のような回路変形を行う。

ここで、 $u(n)$ 、 $x(n)$ 等は 1 ビット信号ではなくマルチビット信号で取り扱われることと、式 (6) における $x(n) - |x(n)|y(n)$ の演算に着目すると、表 1 より $\Delta\Sigma$ 変調器への入力信号 u は 0 または $2x$ の 2 通りとなることが分かる。

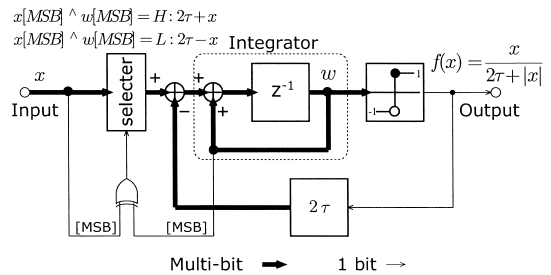
このため、 $u(n)$ は入力 $x(n)$ 、出力 $y(n)$ との XNOR を求め、0、 $2x$ をセレクタで制御することで実現できる。更に $\Delta\Sigma$ 変調器の出力信号 $y(n)$ は内部レジスタの符号ビットを反転させたものであるため、クリティカルパスを考慮すると、出力信号 $y(n)$ の代わりに内部レジスタ $w(n)$ の符号ビットを用いる方が適切である。よって、 $x(n) - |x(n)|y(n)$ は入力信号の符号とレジスタの符号ビットとの XOR によりセレクタを制御することによって実現でき、また τ と x の係数は導出過程で相殺されるため、最終的に図 6 の回路構成が導かれる。

3.2 活性化関数の導関数の構成

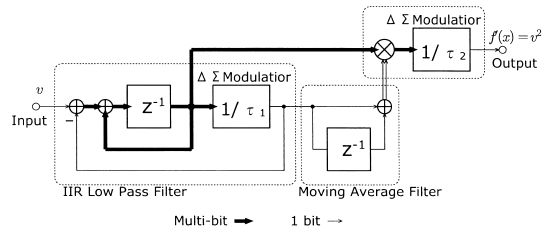
BP 学習においては、逆伝搬処理に必要な活性化関数の導関数についても検討が必要である。活性化関数の導関数は、式 (5) を微分することで次式で与え



Multi-bit \rightarrow 1 bit \rightarrow
図 9 導関数回路の構成
Fig. 9 Derived function based on delta-sigma modulation.



Multi-bit \rightarrow 1 bit \rightarrow
図 10 $\frac{2\tau}{2\tau + |x|}$ を実現する演算器
Fig. 10 Operational unit of $\frac{2\tau}{2\tau + |x|}$.



Multi-bit \rightarrow 1 bit \rightarrow
図 11 1 ビット 2 乗器の構成
Fig. 11 Square multiplier.

られる。

$$f(x)' = \frac{2\tau}{(2\tau + |x|)^2} = \left(\frac{2\tau}{2\tau + |x|} \right)^2 \cdot \frac{1}{2\tau} \quad (9)$$

式 (9) に従い、導関数を実現する回路を構成すると図 9 となり、その特性は図 8 となる。図 9 に示すように、導関数回路は二つの演算器から構成される。各々の演算器は、 $\frac{2\tau}{2\tau + |x|}$ の演算を行う演算器と、2 乗演算を行う演算器から構成される。 $\frac{2\tau}{2\tau + |x|}$ の演算は、図 10 のように構成される。これは図 6 に示す $\Delta\Sigma$ 変調に基づく活性化関数回路において、セレクタからの出力信号を図 10 に示すように変更することで実現可能である。また 2 乗を行う演算は、図 11 に示すように図 5 に示す乗算器を 1 入力为实现した 2 乗演算器により実現する。

4. DSM-PNN による BP 学習の実現

4.1 ニューロンの構成

図 12 に本論文で使用するニューロンの構成を示す。ニューロンにより実行される演算は次式のように表される。

$$h_{jp} = \sum_{k=1}^n w_{kj} g_{kp} \quad (10)$$

$$g_{jp} = f(h_{jp}) \quad (11)$$

$$g'_{jp} = f'(h_{jp}) \quad (12)$$

h_{jp} はパターン p に対するニューロン j の内部電位、 g_{jp} はパターン p に対するニューロン j の出力である。また、 $f(\cdot)$ は提案する活性化関数、 $f'(x)$ はその導関数である。

ニューロンの積和演算は、それぞれ図 3 及び図 4 に示す重み乗算器、多入力加算器により実現される。図 6 に示す活性化関数回路では、絶対値演算を行う必要がありその際に符号情報を必要とする。しかし、1 ビットで表現されたパルス信号中の 1、-1 からはもとの信号の符号情報をそのまま読みとることができない。そこで、加算された 1 ビットパルス信号から符号情報を得るため、図 5 の 1 ビット乗算器でも用いた IIR 型 LPF を挿入することで、セルボディの内部にて一時的にマルチビット信号へと信号変換を行う。その後 $\Delta\Sigma$ 変調に基づく活性化関数により演算し次のニューロンに出力される。

また、ハードウェア実装する際の信号処理の都合上導関数回路をセルボディ内に実装し、セルボディからは活性化関数の出力及びその導関数の出力が得られるようにニューロンの拡張を行った。

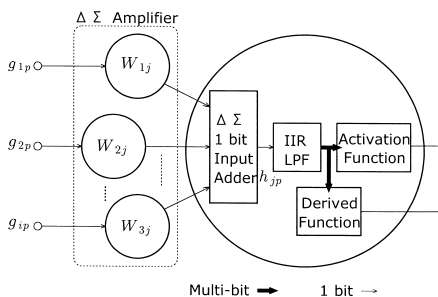


図 12 $\Delta\Sigma$ 変調に基づくパルスニューロンモデルの構成
Fig. 12 Neuron model based on delta-sigma modulation.

4.2 BP 学習回路の構成

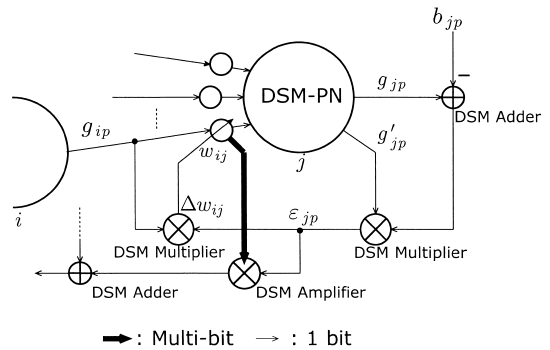
本論文で使用する BP 学習法は、次式で表される最急降下法に基づき重み更新を行う。

$$w'_{ij} = w_{ij} - \rho \cdot \Delta w_{ij} = w_{ij} - \rho \cdot \varepsilon_{jp} \cdot g_{ip} \quad (13)$$

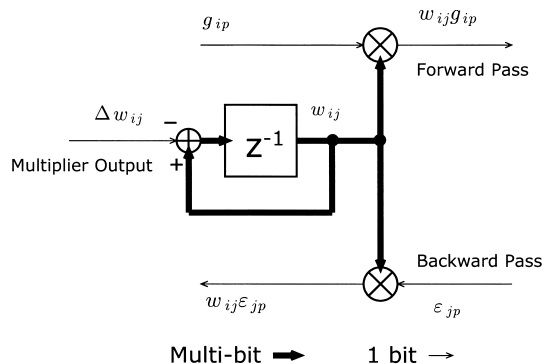
$$\varepsilon_{jp} = \begin{cases} g_{jp} - b_{jp} \cdot f'(h_{jp}) & (\text{ニューロン } j \text{ が出力層}) \\ \left(\sum_k \varepsilon_{kp} w_{jk} \right) \cdot f'(h_{jp}) & (\text{ニューロン } j \text{ が中間層}) \end{cases} \quad (14)$$

ρ は学習係数、 b_{jp} はパターン p に対する教師信号、 $f'(\cdot)$ は提案する導関数である。

図 13 に式 (13)、式 (14) により構成される BP 学習回路を組み込んだネットワークの構成を示す。図 13 は出力層における学習回路の構成である。式 (14) の乗算は、図 5 に示した $\Delta\Sigma$ 変調に基づく 1 ビット乗



Multi-bit \rightarrow 1 bit \rightarrow
図 13 BP 学習回路の構成
Fig. 13 Back propagation learning circuit.



Multi-bit \rightarrow 1 bit \rightarrow
図 14 重み更新回路の構成
Fig. 14 Weight update circuit.

算器により行う。また、学習係数の変更は図 13 中の Δw_{ij} を求める 1 ビット乗算器内部の $\Delta\Sigma$ 変調器のフィードバックゲイン τ_2 を変更することにより設定することができる。また、重み更新回路の構成を図 14 に示す。重み更新回路はレジスタで構成されており、式 (13) に従い重み更新量 Δw_{ij} を表す 1 ビットパルス信号が 1 の場合、レジスタの値を 1 加算、-1 の場合、レジスタの値から 1 減算を繰り返す。

5. 実機実験

5.1 実験概要

提案した BP 学習を行う DSM-PNN によって、FPGA 上にハードウェア実装し実験を行う。DSM-PNN は Verilog-HDL を用いてハードウェアの記述を行う。実験システムの構成を図 15 に示す。実験システムには、ALTERA 社の 20 万ゲート相当の FPGA (APEX20KE EP20K200EFC484-2) を搭載した Exculibur ボードを使用する。FPGA 上には同社が提供するソフト CPU コア (Nios CPU Core 2.2), BP 学習を行うための DSM-PNN, 1 ビットパルス信号変換用の $\Delta\Sigma$ 変調器を実装する。CPU コアからは、DSM-PNN の重みの初期値の設定, 学習データの提示, 学習 ON/OFF 及びリセットなどの制御信号の入力を行う。学習用データは CPU コアより出力される 12 ビットの信号を $\Delta\Sigma$ 変調器により 1 ビットパルス信号に変換し、DSM-PNN に入力する。DSM-PNN と CPU は FPGA の外部からの 33 MHz のクロックにより動作する。また、DSM-PNN に入力される学習データ、及び DSM-PNN の出力信号は、外部に設置した遮断周波数 20 kHz の RC 型の一次遅れ LPF を通して復調し、オシロスコープにより観測する。

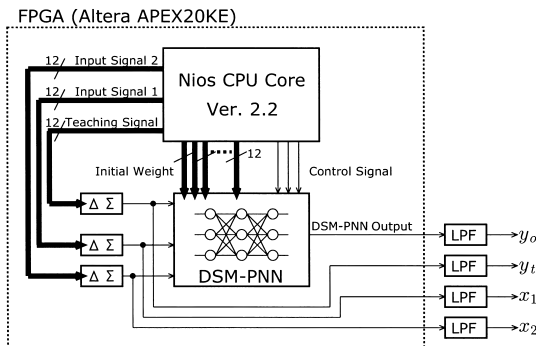


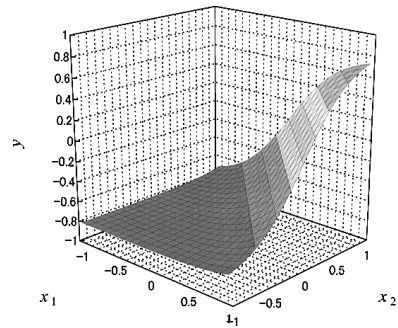
図 15 実験システムの構成
Fig. 15 Experimental system.

ニューラルネットワークの構成は 3 層の階層型とし、それぞれニューロン数は入力層 2, 中間層 3, 出力層 1 とする。

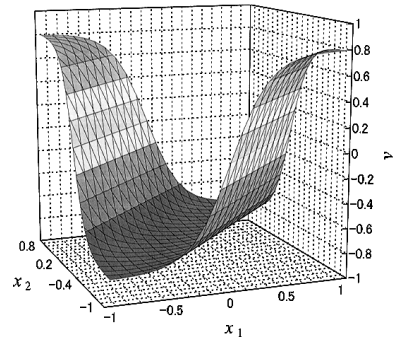
また、回路設計に際して、BP 学習を正しく行うために、各回路の演算精度に注意を払う必要がある。これを決定する各回路内部のビット幅の設定については、様々な検討結果が報告されている [22] が、本論文では、あらかじめ次節で述べる実験にあわせた予備実験の結果から、ニューロンのシナプスの重み乗算器は 12 ビット、乗算器、活性化関数の内部のビット幅は 8 ビットと、回路規模が最小となるように設計した。

なお、これらのビット幅は、提案するシステムが HDL を使用した FPGA 実装であることに加え、すべての信号処理を同一の $\Delta\Sigma$ 変調器を基本に構成しているため、学習対象の要求する演算精度に合わせて、適宜容易に設計変更が可能である。

学習データは図 16 (a), (b) に示す三次元曲面を横軸 x_1 、及び奥行軸 x_2 を入力信号、縦軸を教師信号 y_t とする。具体的には、入力信号 x_1, x_2 として異なる

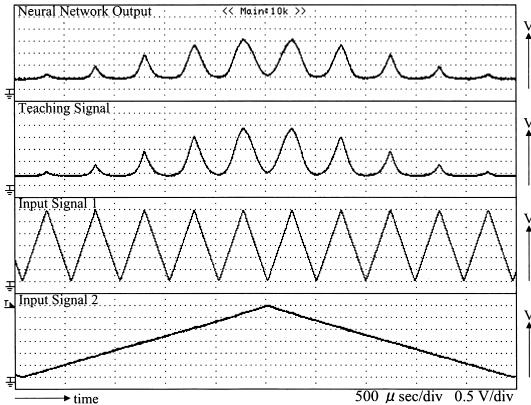


(a) Learning pattern 1

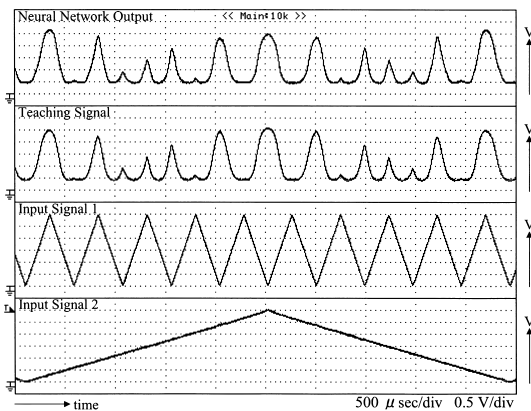


(b) Learning pattern 2

図 16 学習データ
Fig. 16 Learning pattern of DSM-PNN.



(a) Output signal to learning pattern 1
[500 μs/div, 0.5 V/div]



(b) Output signal to learning pattern 2
[500 μs/div, 0.5 V/div]

図 17 DSM-PNN の出力信号と学習データ (上より DSM-PNN の出力信号 y_o , 教師信号 y_t , 入力信号 x_1, x_2)

Fig. 17 Output signal of DSM-PNN and learning pattern.

周期の三角波の時系列信号を設定し、図 16 において点 (x_1, x_2) の時系列に対する曲面の高さ y の時系列データを教師信号とする。図 17 中の下 3 段の時系列データがここで得られる学習データである。重み係数の初期値をランダムに与え、時系列データを 1000 サイクル学習を行う。

5.2 実験結果

図 17 (a), (b) に学習後の DSM-PNN の出力波形を示す。上より、ニューラルネットワークの出力信号 y_o , 教師信号 y_t , 入力信号 x_1, x_2 である。図 17 より、DSM-PNN の出力信号は教師信号と一致し学習が成功していることが分かる。

表 2 DSM-PNN の回路規模と CPUS 値
Table 2 Circuit scale and CUPS.

形状	DSM-PNN		ソフトウェア
	回路規模 [LC]	CUPS [1/s]	CUPS [1/s]
2-2-1	1246	763×10^6	2.27×10^6
2-3-1	1696	1102×10^6	2.45×10^6
2-4-1	2225	1261×10^6	2.49×10^6
2-8-1	4191	2105×10^6	2.69×10^6

表 2 に、中間層数を変化させることによりネットワークの形状を変更した場合の BP 学習回路も含めた回路規模 (1LC 当り約 25 gate) 及び、単位時間当りに更新可能なシナプス荷重数 CUPS (Connections Update Per Second) 値を示す。同表には、演算精度を同等に設定したマルチビット構成でソフトウェア実装されたニューラルネットワークにおける CUPS 値も参考に掲載している。なお、ソフトウェア処理には CPU: Celeron 2 GHz, MEM: 512 MB の PC 上にて実行した。

表 2 より、ニューロン数が増えるに従い、回路規模は約 500 LC ずつ上昇していることが分かる。マルチビットでハードウェア実装されたニューラルネットワークにおいては、マルチビットの結合配線によって、ニューロン数が増えるに従い配線領域増大とされている。しかし、DSM-PNN においてはニューロン間の結合はすべて 1 ビットで接続するため、全体での結線数も少なくすることができ、結果的にニューロン数の増加に伴って線形的に上昇していることが分かる。

また、文献 [1], [4] によれば、デジタル方式のニューラルネットワーク専用 LSI の CUPS 値は、およそ数 G 程度とされている。今回の実装では、CUPS 値は、2-2-1 のネットワークで約 0.8 G, 2-8-1 のネットワークで約 2 G であり、上記文献出版時の LSI 技術では主なマイクロプロセッサが 100 MHz 程度のクロック周波数で動作していたこと、本方式が FPGA 実装でありクロック周波数が 85 MHz にとどまることを考慮すると、動作速度について提案手法の可能性を示すものであるといえる。

一方で、同様に FPGA 上に BP 学習を行うニューラルネットワークを実装した事例 [12] と比較すると、CUPS 値で約 7 万倍と大きく上回っている反面、回路規模は約 2.5 倍に達している。演算精度の設定、使用した FPGA の世代や種類、また論理合成ツール等の条件が異なるため、単純な比較は困難であるが、提案

手法は 4. で述べたように $\Delta\Sigma$ 変調器による統一した信号処理形態にこだわった結果として、動作速度と引換えに回路規模の増加を招いている可能性があり、今後の検討課題である。

今回は FPGA 上への実装であり、速度・回路規模の絶対的な数値は専用 LSI に比べ劣らざるを得ない。しかし、上記の結果は、研究室レベルで容易に実装なニューラルネットワークハードウェアとしての有用性、更に、将来の LSI 化による性能向上の可能性が期待できるものであると考える。

6. む す び

本論文では、BP 学習を実現するための DSM-PNN の実現手法の検討を行った。BP 学習に必要であったニューロンの活性化関数を $\Delta\Sigma$ 変調により実現する手法を提案した。活性化関数も含めニューラルネットワークすべてを $\Delta\Sigma$ 変調に基づき構成することで全体を統一したパルス信号処理形態での信号処理が実現可能である。また、BP 学習を行う DSM-PNN をハードウェアに実装し、動作速度、回路規模の点から評価を行い、提案手法の有効性を示した。

文 献

- [1] 岩田 穆, 雨宮好仁, ニューラルネットワーク LSI, 電子情報通信学会, 1995.
- [2] 塚田 稔, 島 健, ニューラルネットワークとニューロデバイス, 森北出版, 1997.
- [3] 白井支朗, 岩田 彰, ニューロンコンピューティング基礎講座 No.4 競合学習型ニューラルネットワーク, 工学研究社, 1993.
- [4] B.J. Sheu and J. Choi, Neural Information Processing And VLSI, Kluwer Academic Publishers, 1995.
- [5] 山下博司, 黒川恭一, 古賀義亮, “相互結合型バイナリニューラルネットワークのハードウェア化” 信学論 (D-II), vol.J77-D-II, no.10, pp.2130-2137, Oct. 1994.
- [6] A.F. Murray and D.D. Corso, “Pulse-stream VLSI neural networks mixing analog and digital techniques,” IEEE Trans. Neural Netw., vol.2, no.2, pp.193-204, March 1991.
- [7] 村橋善光, 道木慎二, 大熊 繁, “ $\Delta\Sigma$ 変調に基づくパルスニューラルネットワークとその GHA 学習則” 電学論 (C), vol.122-C, no.10, pp.1821-1829, 2002.
- [8] T.D. Sanger, “Optimal unsupervised learning in a single-layer linear feedforward neural network,” Neural Netw., vol.2, pp.459-473, 1989.
- [9] 村橋善光, 太田祐矢, 道木慎二, 大熊 繁, “GHA 学習則による $\Delta\Sigma$ 変調に基づくパルス密度型ニューラルネットワークの FPGA 実装” 信学技報, NC2002-161, March 2002.
- [10] 江口裕俊, 古田俊之, 堀口浩幸, 樗木杉高, “学習機能をもつパルス密度型ニューロンモデルとそのハードウェア” 信学論 (C-II), vol.J74-C-II, no.5, pp.369-376, May 1991.
- [11] 肥川宏臣, “ハードウェア化に適した学習機能付き 3 値多層ニューラルネットワーク” 信学論 (D-II), vol.J81-D-II, no.12, pp.2811-2818, Dec. 1998.
- [12] 肥川宏臣, “パルス位置変調を用いた多層ニューラルネットワーク” 信学論 (D-II), vol.J85-D-II, no.10, pp.1571-1581, Oct. 2002.
- [13] Y. Kondo and Y. Sawada, “Functional abilities of a stochastic logic neural network,” IEEE Trans. Neural Netw., vol.3, no.3, pp.434-443, 1992.
- [14] Y.-C. Kim and M.A. Shanbaltt, “Random noise effects in pulse-mode digital multilayer neural networks,” IEEE Trans. Neural Netw., vol.6, no.1, pp.220-229, 1995.
- [15] 平井有三, 落合辰男, 安永守利, “1000 ニューロン 10 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” 信学論 (D-II), vol.J84-D-II, no.6, pp.1185-1193, June 2001.
- [16] 西澤邦宜, 平井有三, “連続時間 PCA ハードウェアの実時間動作” 信学技報, NC99-110, 2000.
- [17] 奥田忠義, 廣瀬裕哉, 道木慎二, 石田宗秋, “誤差逆伝搬法のハードウェア実装を考慮した近似シグモイド関数の検討” 三重地区計測制御研究講演会講演論文集, B11-1-B11-2, 2002.
- [18] 安藤道則, 太田則一, 渡邊章弘, “高集積化をめざした簡略型ニューロンの設計” 豊田中研 R&D レビュー, vol.31, no.4, pp.13-21, 1996.
- [19] 相良岩男, AD/DA 変換回路入門, 日刊工業, 1991.
- [20] C. Marven, G. Ewers (著), 山口博久 (訳), デジタル信号処理の基礎, 丸善, 1995.
- [21] 山崎芳男, “AD/DA 変換とデジタルフィルタ” 日本音響学会誌, vol.46, no.3, pp.251-257, March 1990.
- [22] 阪上茂生, 香田敏行, 山本浩司, 丸野 進, 木泰治, “パターン認識を学習する誤差逆伝搬法に必要な演算精度の削減” 信学論 (D-II), vol.J76-D-II, no.3, pp.672-679, March 1993.

(平成 16 年 3 月 16 日受付, 8 月 25 日再受付)



奥田 忠義

2002 三重大・工・電気電子卒。2004 同大大学院工学研究科電気電子工学専攻博士前期課程了、現在に至る。



道木 慎二 (正員)

1995 名古屋大学大学院工学研究科電子機械工学専攻博士課程後期課程了。同年同助手, 2000 年 6 月三重大学工学部電気電子工学科助教授, 2004 名古屋大学大学院工学研究科電子情報システム専攻助教授, 現在に至る。工博。主としてモータドライブシステム, パターン情報処理技術の制御への応用に関する研究に従事。1992 IEEE・IECON 論文賞, 2004 電気学会論文賞各受賞。IEEE, 計測自動制御学会各会員。



石田 宗秋

1980 名古屋大学大学院工学研究科電気工学及び電子工学専攻博士課程後期課程了。同年同大学工学部電気工学科助手, 1987 年 6 月三重大学工学部電気工学科助教授, 1996 年 10 月同大学工学部電気工学科教授, 現在に至る。工博。1993 及び 2001 電気学会論文賞, 1999 電気学会著作賞, 2002 計測自動制御学会論文賞各受賞。IEEE, パワーエレクトロニクス研究会, 計測自動制御学会, 日本機械学会各会員。