

ガウシアンマシンと そのアナログ/ディジタル専用アーキテクチャ

Gaussian Machines Neuron Model and its Analog and Digital Architectures

秋山 泰

Akiyama Yutaka

慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

あらまし 筆者らは“ガウシアンマシン”ニューロンモデルを提案している。このモデルの特徴は、入力に対してガウス分布のノイズが印加されるところにある。ガウシアンマシンは、マッカローピッツ・モデル、ホップフィールド・モデル、およびボルツマンマシン・モデルを包含した、より一般的なモデルである。

本論文では、ガウシアンマシンのアナログ、ディジタル両方のアーキテクチャを提案する。このうちアナログ・アーキテクチャはHopfieldのアナログ回路を基礎として、可変コンダクタンス実現のためにスイッチト・レジスタ回路を導入したもとなっている。ディジタル・アーキテクチャは多数のNEU (Neural Execution Unit)を並列動作させるシストリック・アレイとして構成されている。

Abstract

The author and his colleague have proposed an artificial neuron model called “Gaussian machines”. Its significant property is inherently derived from Gaussian random noise appended to the input of each neuron. Gaussian machines model includes McCulloch-Pitts model, Hopfield model, and also Boltzmann machines model as special cases of its definition. In this paper, both analog and digital architecture for Gaussian machines model are discussed. The analog architecture is based on Hopfield’s analog circuit and employs switched resistors in order to achieve variable conductances. And the digital architecture is constructed by a systolic array of the cells called Neural Execution Units (NEUs).

1 ガウシアンマシン・ニューロンモデル

筆者らは、ホップフィールド・モデルやボルツマンマシン・モデル等を包含した、より一般的なニューロンモデルとして“ガウシアンマシン”ニューロンモデルを提案している [1]。

ガウシアンマシンのニューロンをFig. 1に示す。ガウシアンマシンは、a)連続値の出力、b)確率的な決定、c)活性値の時間連続性、の3つの性質によって特徴づけられる。ホップフィールド・モデル [2]はこのうちのa)とc)を満たしているが、その決定論的挙動のためにネットワークの状態が局所的な安定状態に捉えられてしまうという、いわゆる「local minimaの問題」を解決できない。ボルツマンマシン [3]は、b)の性質を満足しlocal minimaの存在する問題空間内での探索にも適するが、出力値が2値で活性値も時間不連続なので、挙動が不安定で定常分布を求めるまでに時間がかかりすぎる。

ガウシアンマシン・モデルの定義は、ほぼホップフ

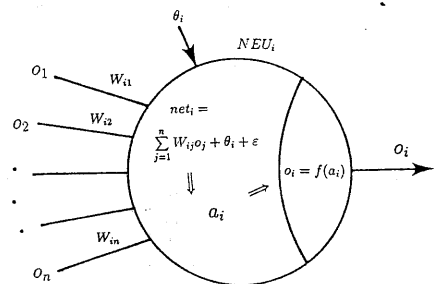


fig.1 ガウシアンマシン・モデルのニューロン

ールド・モデルの定義と同じであるが、入力和 net_i にガウス分布のノイズを印加したためにモデルに非決定性が導入され、local minimaからの脱出が可能となった。

いま i 番目のニューロン NEU_i を考えると、入力 o_j は

$$net_i = \sum_{j=1}^N W_{ij} o_j + \theta_i + \epsilon \quad (1)$$

ただし W_{ij} は NEU_i から NEU_j への結合荷重、 o_j は N

EU_iの出力値、 θ_i はバイアス、 ε はノイズである。

NEU:ニューロンの活性値 a_i は、入力 net_i の影響を受けて、以下の差分方程式で変化する。

$$\frac{\Delta a_i}{\Delta t} = -\frac{a_i}{\tau} + net_i \quad (2)$$

出力 O_i は次のS字状関数により活性値 a_i から求まる。

$$O_i = f(a_i) = \frac{1}{2} \left(\tanh h \left(\frac{a_i}{a_0} \right) + 1 \right) \quad (3)$$

ここで、 a_0 は基準活性値と呼ばれるパラメータで曲線の勾配を制御する働きがある。この値を $a_0=1/4, 2, 4$ と変化した場合のそれぞれの曲線をFig. 2に示す。

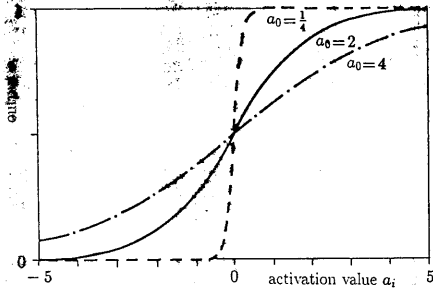


fig.2 ガウシアンマシンの出力関数

ノイズ ε は平均0、分散 σ^2 のガウス分布に従うが、この分散は温度パラメータ T によって制御される。

$$\sigma = kT \quad (4)$$

ただし定数 $k = \sqrt{8/\pi}$ 。(この定数値はボルツマンマシンにおける温度パラメータ T とのCompatibilityを保つ。)

ガウシアンマシン・モデルにおいて各ニューロンの振舞い方を決める重要なパラメータは、1)基準活性値 a_0 、2)温度 T 、及び、3)離散化における時間刻み Δt である。

この3つの基本パラメータによって構成される空間をFig. 3に示す。ただしGM($a_0, T, \Delta t$)の記号はこ

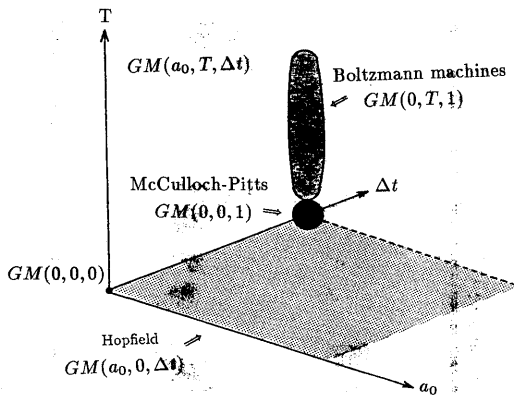


fig.3 ガウシアンマシンのパラメータ空間

のようなパラメータを持つガウシアンマシンを表す。

ここで既存のモデルは、空間内の限られた領域として現れてくる。例えば、マカロッカービットのしきい値モデル[4]はGM($0, 0, 1$)、ホップフィールド・モデルはGM($a_0, 0, \Delta t$)、そしてボルツマンマシン・モデルはGM($0, T, 1$)である。またGM($0, 0, 0$)は、Hopfieldのアナログ回路における高利得極限状態に対応づけられる。ガウシアンマシンは3次元空間内の任意の点を取り得るので、自由度の高いシミュレーテッド・アニーリング手法が適用可能であると考えられる。

2 ガウシアンマシンのアナログ・アーキテクチャ

本章では、ガウシアンマシンのアナログ・アーキテクチャを提案する。ここで紹介するアナログ・アーキテクチャは、Hopfieldが提案したオペアンプと抵抗網で構成されるアナログ回路を、VLSI上に実装する研究から発展したものである。そこで、まずHopfield回路のアナログVLSI実装について述べ、それがどのようにガウシアンマシンと結び付いたかを説明する。

2.1 Hopfieldのアナログ回路

Hopfieldは彼のニューロンモデルと同じ微分方程式で表現されるアナログ回路(Fig. 4)を提案した[2]。

各ニューロンの挙動はオペアンプによってシミュレートされ、各ニューロンの出力値 O_i はオペアンプの出力電圧 V_i の値で表される。またシナプスの結合荷重 W_{ij} はコンダクタンス G_{ij} で表現される。

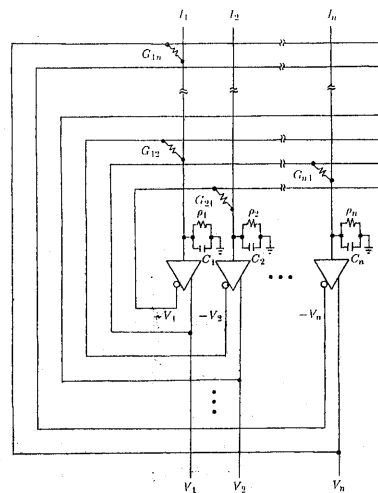


fig.4 Hopfieldのアナログ回路

この回路においてオペアンプへ入力する電流を調べてみると、各コンダクタンスを通じての流入と電流源 I_i からの供給があり、総和は $H_i = \sum G_{ij} V_j + I_i$ となる。ここで電流 I_i を入力バイアス θ_i に対応させれば、この量 H_i は入力荷重 net_i (式 1) に相当する。

オペアンプの入力側には容量 C があり、電流 H_i によって充放電される。このとき、オペアンプの入力電圧 u_i の挙動は次の微分方程式に従う。

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \frac{1}{C} H_i \quad (5)$$

ここで、時定数は $\tau = R_i C$ である。ただし、 R_i は入力側の総抵抗値で次式で定まる。

$$R_i^{-1} = \rho_i^{-1} + \sum_{j=1}^N G_{ij} \quad (6)$$

補償抵抗 ρ_i は各オペアンプの入力側に用意されており、全てのオペアンプで総抵抗値 R を揃えるために用いる。

オペアンプの出力値は、次の S 字状関数で決定される。

$$V_i = g(u_i) = \frac{1}{2} \left(\tanh\left(\frac{u_i}{u_0}\right) + 1 \right) \quad (7)$$

ここで、 u_i は入力電圧、 V_i は出力電圧であり、 u_0 は利得を制御するためのパラメータである。初めに述べたように、この出力電圧 V_i がニューロンからの出力値 O_i を表現する。ところが、コンダクタンスは非負で抑制性の結合を表せないで、符号を反転した出力電圧 $-V_i$ も共に出力し、 $+V_i$ と $-V_i$ の両者を他のオペアンプへのフィードバックに供する必要がある。

オペアンプは、固定利得であれば VLSI 内では小さな面積で簡単に作成できる。また、利得を変化させられるオペアンプも、面積消費はやや大きくなるが作成可能である (Takefuji 私信)。

2.2 問題点—プログラマブルなコンダクタンス

Hopfield の回路を実装する上での大きな問題点は、シナプスの結合荷重を表すコンダクタンスをどのように可変にするかということである。

結合荷重はニューラル・ネットワークに解かせる問題ごとに設定が異なるので、コンダクタンス G_{ij} も使用時にその値を設定できることが望まれる。プログラマブルなコンダクタンスを実現しない限りは、汎用の Hopfield 回路チップは作成できない。

Hopfield 回路で必要となるコンダクタンス行列をチップ化する試みは既にいくつか行われてきたが、ほぼ全ての研究が抵抗を固定式としている。例えば、JPL の Lambe は製造後に 1 回だけ高電圧パルスにより結合を書き込めるチップを作成したが抵抗値は 1 種のみである。AT&T の Bell 研ではプロセスの最終マスクで抵抗値を決める方式をとり、22 ニューロン用及び 54 ニューロン用のチップを製造した。C. Mead は、結合荷重を $\{-1, 0, +1\}$ の 3 種類に

限定しながらも、ある学習規則に従って自動的に変更されるチップを作成した。可変抵抗を目指した研究としては MIT の J. Sage ら [5] による電荷結合素子と MNOS 技術を採用したチップがあるが、まだ実用化には至っていない。

本論文では次節以降で、Hopfield 回路における可変コンダクタンスの実現法を提案する。

2.3 スイッチト・レジスタ技法

Hopfield の回路において可変コンダクタンスを実現するために「スイッチト・レジスタ回路」(switched resistor circuits) を利用する。同回路を Hopfield 回路のシナプス結合部に利用しようという基本的アイデアは、米国 University of South Carolina (現在 Case Western Reserve University) の Prof. Takefuji によるものである。筆者は 1987 年夏の滞米中に同氏の指導のもとでスイッチト・レジスタの導入法を研究し、後述する 2 つの回路、DNIC と SNIC を提案した。これらの回路においてノイズを適切に制御すれば、ガウシアンマシンが実現できる。

スイッチト・レジスタ回路は、固定抵抗 R_0 、容量 C 、及びアナログスイッチで構成される (Fig. 5(A))。

スイッチが常に ON である場合は、充電の時定数は $\tau_0 = R_0 C$ となる。しかしスイッチをデューティ比 d で細かく ON/OFF してやると、実効的な時定数 τ は次式のように変化する。

$$\tau = \frac{R_0}{d} C \quad (8)$$

ここで、 $d = t_{on} / (t_{on} + t_{off})$ である。

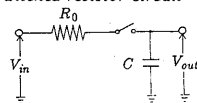
Fig. 5(B) は d が 1.0, 0.5, 0.25 のそれぞれの場合の充電の様子を示す。また Fig. 5(C) はスイッチング・チャートを示す。ここでスイッチング周期: $T_p = t_{on} + t_{off}$ は時定数 τ に比べて十分に小さい必要がある。

時定数 τ の変化は、仮想的にコンダクタンスの変化として考えられる。即ちスイッチが常に ON の場合には $G_0 = 1/R_0$ であったコンダクタンスが、デューティ比 d の操作によって、以下のように可変となる。

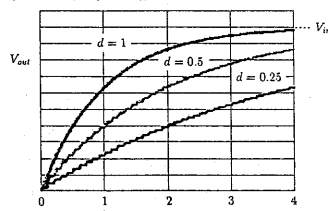
$$G = dG_0 \quad (9)$$

VLSI 上での抵抗 R_0 の実現方法については、 $k \Omega$ 単位の場合には拡散層、数百 Ω の場合にはポリシリコン層を利用できる。今回は $\lambda = 1.5 \mu m$ でチップを設計したが、1 平方 λ 当りの抵抗値は p 及び n 拡散層の場合は 3.5~7.0 $k \Omega$ 、ポリシリコンでは 30~80 Ω の値である。また、容量 C は拡散層と基盤の間に形成され、1 平方 λ あたり約 0.7 fF の値である。VLSI 上で抵抗値や容量値を正確に実現することは大変に困難であり、加工誤差は 20%~30% ほどにも及ぶ。絶対的な誤差はこのように大き

(A) Switched resistor circuit



(B) Duty cycle effects



(C) Switching charts

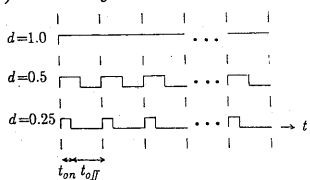


fig.5 スイッチト・レジスタ回路

S F		dij	
S:sign flag F:full-range flag dij:duty cycle			
Representation		Value	
S F	dij	sign	duty cycle
0 0	00000000	+	0/256
0 0	00000001	+	1/256
0 0	00000010	+	2/256
...			
0 0	11111111	+	255/256
0 1	00000000	+	256/256
...			
0 1	11111111	+	256/256
1 0	00000000	-	256/256
...			
1 0	11111111	-	256/256
1 1	00000000	-	256/256
...			
1 1	11111101	-	3/256
1 1	11111110	-	2/256
1 1	11111111	-	1/256

fig.6 デューティ比の表現

V_j (excitatory)

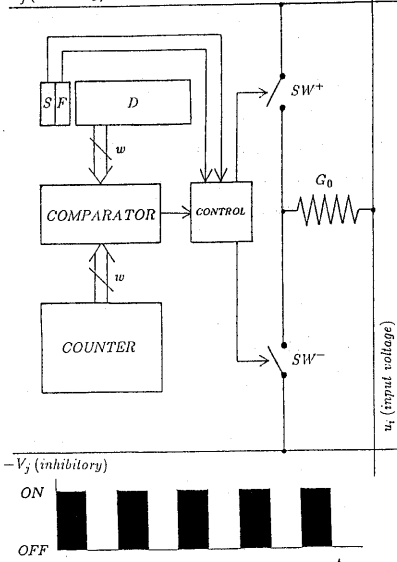


fig.7 DNIC回路と開閉パターン

いが、誤差の方向はチップ内ではほぼ均一なため、相対的な誤差は1%~2%程度に収まる。今回の場合、シナプスの結合荷重を表すためにコンダクタンスの比を変えたいのであるから絶対誤差の値よりは相対誤差に興味があり、別個のチップを接続して使用することを考えない限り、絶対誤差は問題とならない。

可変コンダクタンスを作るための技法は信号処理の分野においてLSIフィルタを作成する目的で長く研究されてきた。本論文で採用したスイッチト・レジスタ技法以外にも、スイッチト・キャパシタ技法や、MOSのゲート抵抗を電圧制御する方法などがVLSI化に適した方法として知られている。

2. 4 専用アナログVLSIチップの基本構成

Hopfield回路内のコンダクタンスは、スイッチト・レジスタ回路によって可変にできる。ここでは、この可変コンダクタンスを用いたHopfield回路の専用VLSIチップの基本構成を述べる。

シナプスを表す結合ごとに1つずつスイッチト・レジスタ回路を用意する。そして結合荷重を表す2進数を格納するレジスタ(Register)をやはり結合箇所ごとに置く。これを“デューティ比レジスタ”と呼ぶ。

ニューロン数がNの場合に、スイッチト・レジスタ及びデューティ比レジスタは N^2 個のみ用意すればよい。これは、コンダクタンスが(荷重の正負に従って) $+V_i$ か $-V_i$ か的一方にしか接続されないからである。この接続の選択は、デューティ比レジスタ内に符号ビット(“S”ビット)を設けることで制御させる。

レジスタのビット幅wは実現されるデューティ比の精度に影響を与える。デューティ比dはwビットの整数D($d = D/2^w$)として表現され、丸め誤差は $0.5/2^w$ 程度になる。ただし、コンダクタンスの精度は抵抗やコンデンサの加工精度によって抑えられてしまうため、wを余り大きくしても意味がない。

ところで、いくつかの応用においては常時結合 $d = 1$ を用いたい場合がある。そこで、フルレンジを表現する“F”ビットを別に用意した。(これはwを1つ増すのに比べてハードの増加量が少ない。)

従って、デューティ比レジスタのビット幅は合計して(w+2)ビットとなる。Fig. 6はビット表現と、極性およびデューティ比の関係を示している。対応は、2の補数表示に基づいているが、値が1で飽和している点が異なっている。

デューティ比レジスタの内容は、計算に先だってホスト計算機からロードする。現在のところ、学習はホスト計算機で計算した新しい荷重を再ロードすることを考えている。これらのロードに際しアドレス機構を設けるのはコストが大きい。そこで、全てのデューティ比レジスタが1つのシフトレジスタを構成するよう連結しておき、ローディングはビットシリアルに行わせる。

以上でチップの基本構成を説明した。残る問題は、いかなる回路で実際にデューティ比を操作するかである。次節以降で、デューティ比制御の2つの方法を提案する。

2. 5 決定論的なデューティ比制御

この方法は、デューティ比制御のため、 w ビットのカウンタと、 w ビットの比較器を用いる。カウンタは周期的に0から 2^w-1 までの値を生成していく。比較器はカウンタの出力 C とデューティ比レジスタ内に格納された値 D を比較して、 $C < D$ であるか否かを判定する。判定結果を LT と表すと、 LT が真の場合にはスイッチ・レジスタのスイッチをONにし、偽の場合にはOFFにする。このとき、 $C = 0$ から $C = D - 1$ までの期間が t_{on} 期間となり、 $C = D$ から $C = 2^w - 1$ までが t_{off} 期間に相当する。従って、望んだデューティ比 d が得られる。

ここで周期 t_p は、カウンタの値を更新させるクロックの周期(t_{sw})と 2^w との積となるが、その時間はアナログ回路側の時定数 τ に比べて充分に小さくなければならない： $t_p = t_{on} + t_{off} = t_{sw} 2^w \ll \tau$

Fig. 7は、この決定論的な結合回路(DNIC: Deterministic Neural Interconnection Circuit)のブロック図を示す。回路中には2つのスイッチ SW^+ と SW^- があり、興奮性入力(+ V_j)と抑制性入力(- V_j)のそれぞれへの接続を開閉する。デューティ比レジスタ内の“ S ”ビットが0のときには SW^+ 、1のときには SW^- が選択される。“ F ”ビットは、 $F \neq S$ の場合にフルレンジを表し、比較結果 LT に係わらずスイッチをONにする。

2. 6 確率的なデューティ比制御

決定論的な結合回路ではカウンタが大きな面積を消費している。そのカウンタを w ビットの乱数発生器で置き換えたものが確率的な結合回路(SNIC: Stochastic Neural Interconnection Circuit)である(Fig. 8)。ここでは比較器は、乱数 R とデューティ比レジスタの内容 D とを比較し、 $R < D$ か否かを結果 LT として出力する。

DNICでは、デューティ比は表現精度の範囲内では必ず望んだ値が実現された。しかし、SNICではデューティ比は望んだ値の周囲にガウス分布をとる。

SNICの有する非決定性は、ネットワークがローカルミナマの状態から脱出するのに寄与する。

乱数 R は $(0, 1)$ に一様分布する。すると“試行” $R < D$ の成功および失敗は二項分布として表現される。成功確率は $p = D / 2^w$ 、失敗確率は $q = 1 - D / 2^w$ となり、

この試行を m 回行ったときの成功の回数は平均 $\mu = mp$ 、分散 $\sigma^2 = mpq$ となる。いま興味のあるのはデューティ比なので成功の比率 d について考えてみると、その平均 μ_d および分散 σ_d^2 は、

$$\mu_d = p = d_{av}, \quad \sigma_d^2 = \frac{pq}{m} = \frac{d_{av}(1-d_{av})}{m} \quad (10)$$

ただし $d_{av} \cong D / 2^w$ である。この式から、分散 σ_d^2 は試

行回数 m のみならず望まれるデューティ比 d_{av} に依存することがわかる。 $d_{av} = 0.5$ のときに、分散は最大値 $\max \sigma_d^2 = (4m)^{-1}$ をとる。

試行数 m が大きくなると二項分布はガウス分布で近似できる。ここで、デューティ比の誤差の絶対値をある指定値 e 以下に抑えるために必要な試行数 m を求めてみる。対象が確率的であるため誤差を100%の確率で抑え込むことは不可能である。そこで95%以上の確率で誤差の絶対値が e 以下になるような m を求めるものとする。ガウス分布の確率密度関数より、およそ $2 \times \max \sigma_d^2 \leq e^2$ となるように設定すれば良いから、結果として次式を得る。

$$m \geq \frac{1}{e^2} \quad (11)$$

例として $w = 8$ ビットの場合を考えると、丸め誤差が $0.5/2^8 \cong 0.002$ であるから、誤差 e は 10^{-3} 程度に抑えたい。上式より必要試行数 m は $m = 10^6$ となる。スイッチング間隔 t_{sw} が10nsの場合、試行に10msかかるが、アナログ回路側の時定数 τ をこの程度に設定すればよい。

一様分布を生成するためには、 M 系列の乱数発生器を用いる。 M 系列は多次元の一様性が証明されている。 M 系列の乱数発生器はシフトレジスタとXORゲートから構成される[6]。例えば周期が 2^{521} であるGF(2)上の生成多項式： $G(x) = x^{521} + x^{32} + 1$ を採用すると、必要なシフトレジスタの長さは521ビットとなり排他的論理和ゲートは1個となる。この発生器では521ビットの乱数が生成されるが、このうち w ビットを使用する。このとき、チップ内に521ビットのシフトレジスタを分散的に配置し、

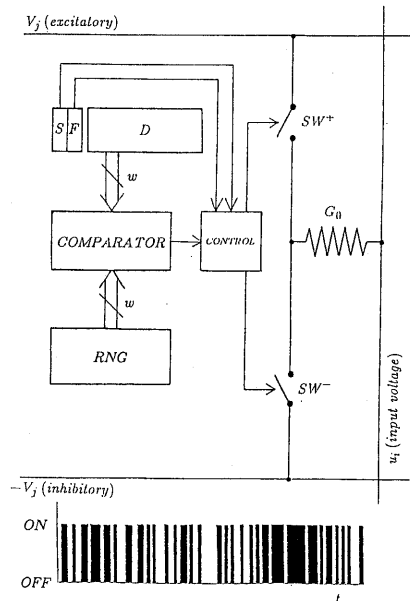


fig.8 SNIC回路と開閉パターン

各SNICがそのうちのwビットずつを用いるようにする。w=8の場合、65個までのSNICに分配が可能である。ただし乱数は使用する毎にwビットずつシフトさせてやる。この方法では、各ニューロン間での純粋な独立性は失われるが、乱数の一様性は保たれる。

最後に、デューティ比の誤差がコンダクタンス G_{ij} 、入力電流 H_i 、入力電圧 u_i に与える影響を求める。ここでは、ある単位時間あたりの試行回数をmとおく。

コンダクタンスの(単位時間当りの)分散 $\sigma_{G_{ij}}^2$ は、 $G = dG_0$ の関係より、

$$\sigma_{G_{ij}}^2 = \sigma_d^2 G_0^2 = \frac{G_{ij}(G_0 - G_{ij})}{m} \quad (12)$$

となり、入力電流 H_i の(単位時間当りの)分散は、

$$\sigma_{H_i}^2 = \sum_{j=1}^N (\sigma_{G_{ij}} V_j)^2 \quad (13)$$

また入力電圧 u_i の分散はこれを時間積分して、

$$\begin{aligned} \sigma_{u_i}^2 &= \int_0^\infty \frac{\sigma_{H_i}^2}{C^2} \exp\left(-\frac{t}{\tau}\right) dt \\ &= \frac{\tau}{C^2} \sum_{j=1}^N (\sigma_{G_{ij}} V_j)^2 \end{aligned} \quad (14)$$

で示される。

入力電流 H_i の分散も入力電圧 u_i の分散も単位時間当りの試行回数mに反比例する。したがって、スイッチング周期 t_{sw} をクロックの分周などを用いて変化させ単位時間当りの試行数mを操作することにより、電流 H_i や電圧 u_i へのノイズ、言い換えればニューロンモデルにおける net_i や a_i へのノイズを制御できることがわかる。

2. 7 設計と実装

DNICとSNICの両回路を、米国 University of South Carolinaにおいて設計した。回路設計は筆者及び Prof. Takefujiが行い、レイアウト設計は主に Yong B. Cho (現在Case Western Reserve Univ.) によって行われた。設計は、 $\lambda = 1.5 \mu m$ (3 μ ルール)のCMOSテクノロジーを前提とした。設計後にSNICについては米国の教育プログラムであるMOSISシステムに製造を依頼し、現在までに完成品が入手できた。

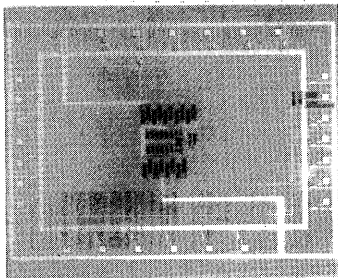


fig.9 SNICの拡大写真

ビット幅wは8ビットとした。このためデューティ比レジスタは $(w+2)=10$ ビットとなり、比較器、DNICのカウンタ、SNICの乱数用シフトレジスタはいずれも8ビットが必要となった。レイアウト面積はDNICがおおよそ $700 \lambda \times 700 \lambda$ 、一方のSNICが $700 \lambda \times 500 \lambda$ となった。

今回実際に製造されたSNICチップは、テストを目的としてSNIC回路1つだけをチップとしたものであり、面積はおおよそ1mm角である。Fig.9に製造されたSNIC回路の拡大写真を示す。回路の上段がデューティ比レジスタ、中段が比較器、下段が乱数用シフトレジスタである。また中段右端の小さな部分がアナログスイッチとその制御回路である。制御回路は、NORゲート2個とNANDゲート2個のみで構成できる(Fig.10)。抵抗 R_0 は左端のパッドの近くに作成されている。完成したチップは、28ピンのDIPにパッケージされている(Fig.11)。

今回のレイアウト設計では最適配置とする努力を省略したので、かなり隙間が開いておりフルカスタム化による面積縮小が期待できる。しかし、より大きな要素としてビット数wの短縮がある。ここで、仮に $w=4$ とし、 $\lambda=0.8 \mu m$ (MOSISにおける最小)、ダイサイズが15mm角(同最大)でどれだけSNICが収納できるかを計算すると、かなり内輪に見積っても $60 \times 60 = 3600$ 個が1チップに収まることになる。これを用いれば60ニューロンまでの任意のシナプス結合が実現可能となる。

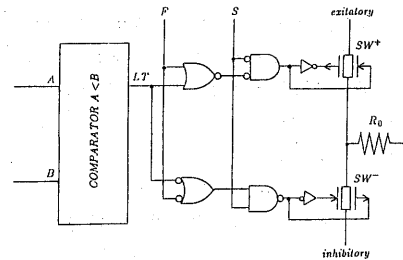


fig.10 スイッチ制御の論理回路

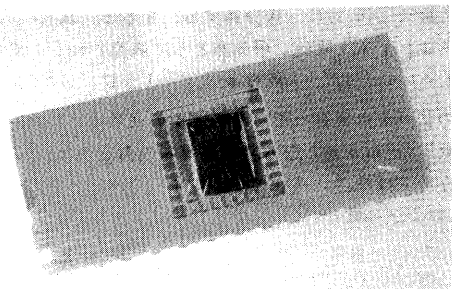


fig.11 SNICチップの外観

2. 8 ガウシアンマシンのアナログアーキテクチャ

以上の節でプログラマブルなコンダクタンスを有するHopfield回路のVLSI実装について述べた。デューティ比制御の方式として提案したDNICとSNICのどちらを採用しても、ガウシアンマシンのアナログアーキテクチャとすることができる。

DNIC回路を用いた場合、チップはホップフィールド・モデルを単純に実現することになる。そこで外部からの供給電流 H_i をノイズにより振らしてやれば、ガウシアンマシン・モデルの特徴である入力ノイズ項を実現することができる。温度パラメータ T はこのノイズの振幅によって制御される。モデルとアーキテクチャの隔たりとして、モデルでは結合荷重 W_{ij} に値域の制限がなかったが、アーキテクチャではコンダクタンスの値が(G_0 を基準として) $[0, 1]$ に制限されている点がある。結合荷重の正規化は、一部の学習規則の適用を困難にする。

SNIC回路を用いた場合は、ガウシアンマシンの本来の定義とは異なり、ノイズが結合荷重の上に印加される。このノイズの大きさは、単位時間あたりのスイッチング回数 m の自乗根に反比例するので、これで温度パラメータ T を制御する。即ち、初めは分周された遅いクロックでスイッチを開閉し、次第にクロックを速くしてスイッチング周期 t_{sw} を短くするのである。

SNICを用いるアーキテクチャは、興味深いガウシアンマシンの変種を示唆している。12式で示されたようにノイズは荷重 W_{ij} が中間的な値を持っている結合、つまり接続するか否かを迷っているシナプスに対して最も大きく影響する。各ニューロンが持つ結合荷重のパターンはばらばらなので、結果としてニューロンごとに異なった温度 T に置かれていることと等価になる。しかも荷重へのノイズは実際に入力される値 O_j と乗算されて初めて意味を持つ。これらの性質の計算モデルとしての良否、または生理学的解釈については現在のところ不明である。しかし、既存のシミュレーテッド・アニーリング手法が系内の温度を常に一定と仮定するのに対して、ニューロンの活動に関係して部分ごとに温度が異なるまま、次第に低温に移行するという方法には興味深いものがある。

3. ガウシアンマシンのデジタル・アーキテクチャ

ガウシアンマシンはデジタル回路によって実装することも可能である。この章では、ガウシアンマシンのシストリック・アーキテクチャを提案する。

3. 1 シストリックNEUアレイ

ガウシアンマシンにおけるニューロン1個の挙動を担当するデジタルプロセッサを設計し、それを N 台並列に

動作させることで、ニューロン N 個で構成されるニューラル・ネットワークを実現する。

ここで問題となるのは N 台のプロセッサの接続方式である。本研究の提案は、プロセッサを1次元の環状ネットワークで接続し、パイプライン的に処理することである。この方式はH. T. Kung[7]らによって“シストリック・アルゴリズム”(systolic algorithms)として研究されてきた。ニューロンのシミュレーションの場合には、処理が全ユニット間で均一かつ時間的にも一定であるため、シストリック・アレイでの実現に大変に適している。

ここでニューロンに相当する各プロセッサを“NEU”(Neural Execution Unit)と呼ぶことにする。各NEUは1つのニューロンの挙動を担当し、他のNEUの出力値を受取って自分の出力値を更新する。NEUはFig. 12(A)のように環状に接続される。

出力値 O_j はNEU j から出力された後、“ステップ”ごとに右に1ユニットずつシフトされる(Fig. 12(B))。受け取ったNEU j は入力総和として入力値 O_j と結合荷重 W_{ij} の積を加えていく。Nステップの後に、値 O_j は送り手のNEU j に戻ってくる。(N+1)ステップ目には更新された出力値が送出される。アレイ内には同時にNニューロン全ての出力値が流れており、それぞれが一周の間にパイプライン的に処理される。このとき送り手への帰還は同一のステップに集中する(Fig. 12(B))。

ガウス分布のノイズ e は外部の乱数発生器によって生成し同様にアレイ内を流す。ただし e は循環させる必要はない。ノイズはNステップおきに必要となるので、供給されたノイズ e は一度ずつしか使用されない。

3. 2 NEUの内部構造と動作

NEU内部のブロック図をFig. 13に示す。

NEUは2つの入力ポートと2つの出力ポートを持つ。また5個の定数レジスタがあり、ニューロン数 N 、減衰因子 α 、バイアス θ 、活性値 a 、活性値 a の初期値を保持する。これらの定数はシステムの初期化に際してポートを通じて設定される。荷重ベクトルは“荷重RAM”に保持し、ダウンカウンタでそのアドレスを発生させる。乗算は“乗算ROM”の読みだしによって値を得る。活性値 a から出力値 O への変換は“応答RAM”の読みだしで実行される。RAM類も初期化時に設定される。

次に動作を説明する。NEUは1ステップ間に次の4種類の“フェーズ”を順に実行していく。

① P_a (減衰フェーズ)

NEUの持つ活性値 a を、 $(1 - \Delta t / \tau) a$ へと減衰させるフェーズ。ただし減衰因子 $\alpha = -\log_2(\Delta t / \tau)$ が自然数となるような減衰だけに制限する。これは減衰を単なるビットシフトで実現したいためである。

② P_θ (バイアスフェーズ)

あらかじめθレジスタに設定されているバイアスθをaに加算するフェーズ。

③ P_i (入力フェーズ)

他のNEUからの出力をO_{in}ポートより受け取り、それを荷重RAM中のWと乗算し、結果を加算するフェーズ。このフェーズは(N-1)回繰り返して実行される。

④ P_o (出力フェーズ)

e_{in}ポートから入力したノイズεをaに加え、応答RAMで変換した値をO_{out}ポートから出力するフェーズ。

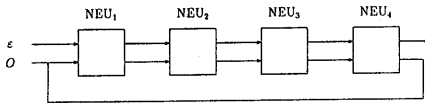
4. むすび

本論文ではガウシアンマシン・ニューロンモデルについて簡単に紹介した上で、そのアナログアーキテクチャとデジタルアーキテクチャをそれぞれ提案した。

アナログアーキテクチャは、スイッチト・レジスタ技法の採用によってコンダクタンスが可変なHopfield回路を実現し、その上でノイズを制御することで得られる。スイッチの制御回路としてDNICとSNICの2種を設計し、SNICについてはチップを製造した。

デジタルアーキテクチャとしては、シストリック・アレイの手法を採用した環状NEUアレイを提案した。

(A) The linear structure of the systolic NEU array



(B) The operation of the NEU array

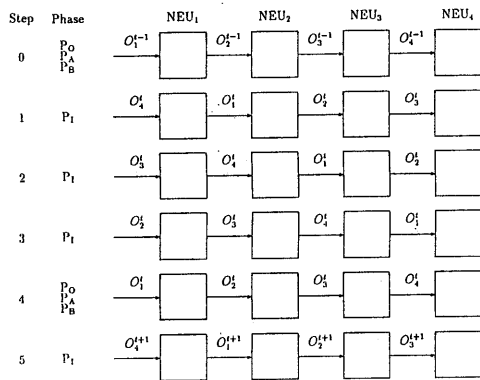


fig.12 ガウシアンマシンのデジタル・アーキテクチャ

References

- [1] Yutaka Akiyama, and Akira Yamashita : "Gaussian Machines: A General Neuron Model", SICE'88, The Society of Instrument and Control Engineers, Aug 1988. (to be appeared)
- [2] J. J. Hopfield : "Neurons with graded response have collective computational properties like those of two-state neurons", Biophysics, Proc. Natl. Acad. Sci. USA, vol. 81, pp.3088-3092, May 1984.
- [3] Geoffrey E. Hinton, Terrence J. Sejnowski, and David H. Ackley : "Boltzmann Machines: Constraint Satisfaction Networks that Learn", CMU-CS-84-119, Carnegie-Mellon University, May. 1984.
- [4] W. S. McCulloch, and W. Pitts : "A Logical Calculus of the Ideas Imminent in Nervous Activity", Mathematical Biophysics, 5, pp.115-133, 1943.
- [5] J. P. Sage, et. al. : "An Artificial Neural Network Integrated Circuit Based On MNOS/CCD Principles", Neural Networks For Computing (Ed. J. Denker), AIP Conf. Proc. 151, 1986
- [6] Herbert S. Bright, and Richard L. Enison : "Quasi-Random Number Sequences from a Long-Period TLP Generator with Remarks on Application to Cryptography", ACM Computing Surveys, vol. 11, No. 4, pp. 357-370, Dec. 1979.
- [7] H. T. Kung: "Why Systolic Architectures?" COMPUTER, pp.37-46, Jan, 1982.

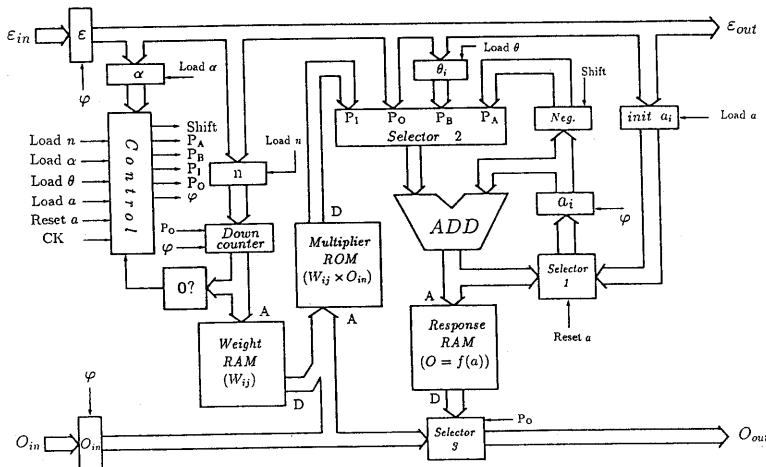


fig.13 NEUのブロック図