

認識と学習機能を有する連想メモリベースシステムの研究 — 自動的な参照データの学習と最適化、及びその集積化 —

Hans Jürgen Mattausch (ナノデバイス・システム研究センター 教授,
先端研半導体集積科学専攻),

小出 哲士 (ナノデバイス・システム研究センター助教授, 先端研半導体集積科学専攻),
溝上 政弘 (先端研量子物質科学専攻 M2), 白川 佳則 (先端研量子物質科学専攻 M1)

1. 研究目的

パターン認識システムにおいて、パターンマッチング処理は認識処理における重要な要素である。従来のシステムでは、プロセッサをベースに実現したモデルが一般的である。このモデルでは、逐次比較で認識を行うため、参照データ数が増えるにつれ、処理時間が増大してしまう。また新規参照データの登録といった学習に対する概念がなく、学習による処理時間の短縮効果は得られない。パターン認識と学習は、人工知能システムを構築するのに必要不可欠な機能である。そのため、これらの機能を集積回路上で実現することは技術的に非常に重要である。これまでに提案されているパターン認識と学習機能の実現方法は、ニューラルネットワークを用いたものがほとんどである[4]。しかしニューラルネットワークを用いた方法では、各ニューロンの閾値とネットワークの中の各ネットの荷重の更新が必要となり、大量の処理時間がかかってしまうという問題がある。また、効率よくパターンをネットワーク中に記憶する方法がないため、パターンの情報をネットワーク構造自体に取り入れる必要がある。更に、既にネットワークの学習が完了した後で、新たなパターンを認識できるように学習するためには、ネットワーク全体の再学習が必要となり、認識処理をしながら学習を行ってネットの荷重の更新をすることは困難である。

この問題に対し、我々は、認識と学習機能を有する人工知能システムを構築するための新しいメモリベースアーキテクチャ、全並列アーキテクチャによる小面積・高速連想メモリ[1-3]を用いた、短期・長期的な記憶を扱う教師を必要としない自動的な学習アルゴリズム(Fig.1), 及びパターン認識と参照データの学習を同時に行うことのできる最適学習アル

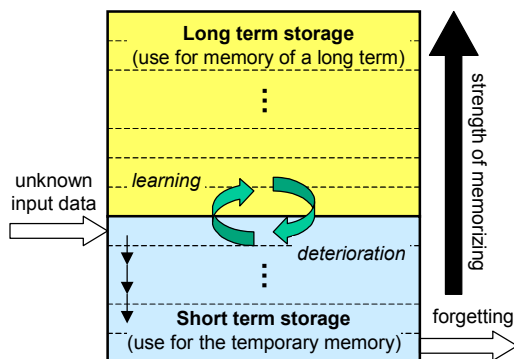


Fig. 1: Learning concept based on a short/long term memory.

ゴリズムとLSI化が可能な連想メモリベース学習システムのアーキテクチャの提案をした。

2. これまでの研究成果概要

2.1. 短期長期記憶に基づくメモリベース学習アルゴリズムの提案

提案アルゴリズムでは連想メモリに記憶された各参照データに対して、どれだけ長い期間記憶しておくかの指標としてrankを与える。そしてrankの順位に応じて参照データの上位を長期記憶、下位を短期記憶と区別する。連想メモリベース認識・学習アルゴリズムのフローチャートを Fig.2 に示す。以下にアルゴリズムの詳細を示す。

[短期・長期記憶に基づく連想メモリベース自動学習アルゴリズム]

Step.1: 入力データと連想メモリ内の参照データの最小距離を距離指標(ハミング距離, マンハッタン距離)に基づき計算し, 最小距離参照データ(Winner と呼ぶ)を決定する。

Step.2: 入力データと Winner 参照データ間の距離Dを計算する。

Step.3(a): $D < \text{しきい値}$ の場合には, 入力データと Winner 参照データは類似していると判断し, 入力データを Winner 参照データと認識する。そして, Winner となった参照データのrankを上げる。この上げ幅は, 検索時の Winner のrankに基づいて決定し, Winner が長期記憶に属していた場合には J_L , 短期記憶に属していた場合には J_S (ただし $J_L > J_S$) とする。

Step.3(b): $D \geq \text{しきい値}$ の場合には, 入力データと Winner 参照データの距離は最小ではあるが, 大きく離れているため, これらのデータは異なるデータと見なし, 入力データを新規データとして認識する。そして, 入力データを一時的な記憶として使用する短期記憶の最上位rank位置に挿入し, 他の短期記憶にある参照データのrankを1つ繰り下げる。

Step.4: 上記 Step.1~3 を入力がある間繰り返す。

2.2. 参照データ最適学習アルゴリズムの提案

本研究では, アルゴリズムをより簡略にし, 簡単な加減算を用いることで, 参照データの最適化を実現する。この学習では, 参照データの学習とその参照

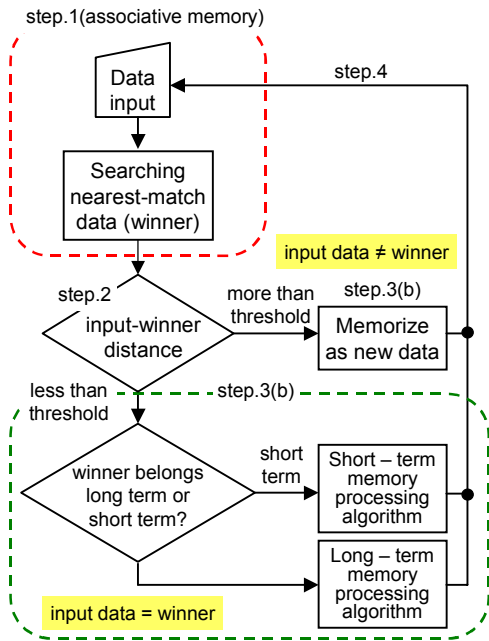


Fig. 2: Flow chart of the memory pattern-learning algorithm based on short/long-term memory. pattern-learning algorithm.

データを認識するためのしきい値 D_{th_i} の認識領域の学習を組み合わせた。しきい値 D_{th_i} は参照データそれぞれに対応した値を持つ。アルゴリズムのフローチャートを Fig.3 に示す。

(1)参照データの学習方法

参照データの学習は以下の要領で行われる。最適な参照データとは、ある参照データが認識 winner となるような全入力データの重心(中心)である。しかし、入力データが連続して入力されるようなオンラインの認識では、このような最適な参照データを一般的に得ることができない。そこで学習アルゴリズムでは、入力データがしきい値の内側であり、参照データが入力データに対応すると認識された winner、認識 winner に対応する一定回数 N 回分の入力データから、それらの重心を計算し、参照データの最適化を行う。具体的には以下の要領で行う。入力データが入力されると、参照データとのマンハッタン距離を次式(1)のようにそれぞれ計算する。

$$D_i = \sum_{j=1}^C |Y_j - X_{ij}| \quad (1)$$

その k 回目に入力される入力データ $Y_{w,k}$ に一番近い参照データ, winner X_w に対応するしきい値 D_{th_w} とマンハッタン距離 $D_{w,k}$ を比較し, D_{th_w} より小さい, または等しいならば, 入力データ $Y_{w,k}$ と winner X_w のベクトル差 $G_{w,k}$ を式(2)のように求める。

$$G_{w,k} = Y_{w,k} - X_w \quad (2)$$

$G_{w,k}$ は次の式(3)に示すように, Memory1_w に記憶されている G_m に加えられる(Fig.3(e)). ステップ(f)で

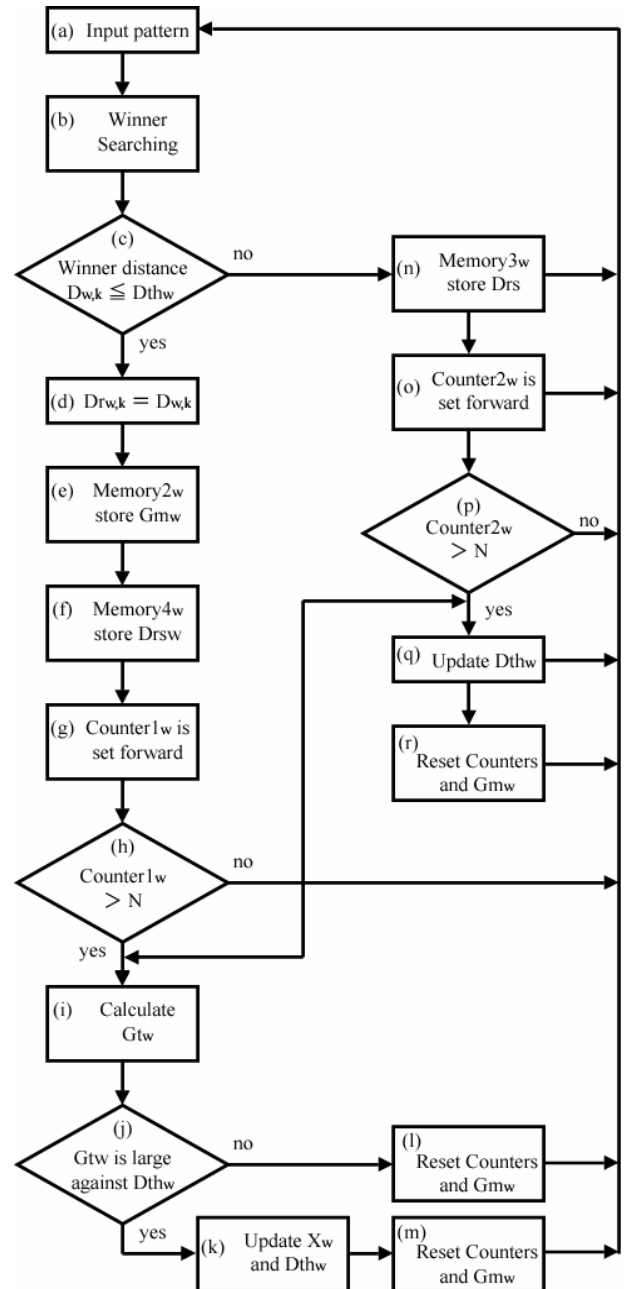


Fig. 3: Flow chart the learning algorithm for reference-pattern and recognition threshold optimization.

はしきい値が更新される。このことは次のセクション 2.2.(3)で具体的に説明する。winner パターンをカウントする $Counter1_w$ は Fig.3(g)のように 1 つ進められ, $k+1$ となる。

$$G_{m_{w,k}} = G_{m_w} + G_{w,k} \quad (3)$$

$k > N$ の場合, 式(4)のように winner パターン X_w が最適化される(Fig.3(k)). ただし Fig.3(j)のような条件が満たされる場合は, セクション 2.2.(3)で説明するよう動作する。

$$X_w = X_w + G_{m_w} / N \quad (4)$$

(2)しきい値の学習方法

しきい値は入力データを認識するために用いら

れる。つまり、もし winner 距離 $D_{w,k}$ が D_{thw} よりも大きかったならば、winner となる参照パターンは入力パターンと大きく異なるということを意味し、認識されないことを意味する。参照パターンを最適化する入力パターンを認識するかを判断するしきい値により、参照データの ローカルミニマが回避されるのである。それゆえ、しきい値の学習、つまり認識領域の学習というものもまた重要になってくる。

しきい値の学習のため、入力パターンがしきい値領域の内側か外側かを区別することが必要である。もし winner に対応する入力パターン全てが、しきい値によって定められた領域の内側であるならば、このしきい値は認識を行うのに効果的である。一方、もし winner に対応する入力データが全て、しきい値によって定められた領域の外側であるならば、この認識領域を拡大する、つまりしきい値を大きくする必要があるのである。本アルゴリズムでは、しきい値学習は2つの要因で実行される。第一の要因は、しきい値 D_{thw} の内側か外側に入力された入力パターンの割合である。これはしきい値 D_{thw} を拡大するか縮小するかを決定するためである。第二の要因は winner 距離 D_w としきい値 D_{thw} の差である。この第二の要因は、しきい値を更新する際、どれほどしきい値を減らす、または増やすかを決定する。我々のしきい値学習アルゴリズムのため、 $Counter2_w$ を付け加える。 $Counter2_w$ は、参照パターン i が winner である場合、winner に対応する入力パターンがしきい値領域の外側 ($>D_{thw}$) のとき、カウントを1つ進める (Fig.3(o))。逆にしきい値の内側に入力された winner に対応する入力パターンの数は、参照データの学習に用いられる (Fig.3(h)) $Counter1_w$ によってカウントされる。

我々はまた、しきい値領域の内側にある入力データの winner による winner 距離 $D_{w,k}$ を特別に $Dr_{w,k}$ とした (Fig.4)。

$Counter1_w$ がカウントされるとき、認識 winner 距離 $Dr_{w,k}$ としきい値 D_{thw} の差を計算し、その差を式(5)に示すような Drs_w に、足し合わせて記憶させる。一方、winner 距離 $D_{w,k}$ がしきい値 D_{thw} よりも大きかつ

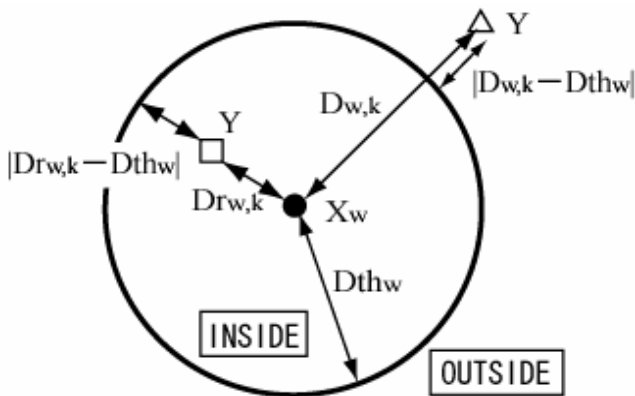


Fig. 4: The location of winner, input pattern and threshold value.

た場合、 $Counter2_w$ は一つカウントが進む。非認識 winner 距離 $D_{w,k}$ としきい値 D_{thw} の差の和は式(6)のように Ds_w に足し合わせて記憶される。

$$Drs_w = Drs_w + |Dr_{w,k} - D_{thw}| \quad (5)$$

$$Ds_w = Ds_w + |D_{w,k} - D_{thw}| \quad (6)$$

$Counter1_w$, $Counter2_w$ の値のどちらかが一定値 N よりも大きくなると、しきい値学習は始まる (Fig.3(h),(p))。 $Counter1_w=N$ であつ、 $Counter2_w=0$ の場合、全ての入力データはしきい値 D_{thw} の内側にあるので、しきい値 D_{thw} は小さくされる。このため、我々は認識 winner の差の平均である $Drg_w (=Drs_w/N)$ を計算する。 Drg_w に基づき、しきい値 D_{thw} の更新量 Dc_w が決まる。

しかし、本アルゴリズムでは一定回数分のデータだけでしきい値を変更することになり、しきい値が最適にならない場合もある。そこで確率的にしきい値変更量 Dc_w を選択することで、この問題を解決する。今回は式(7)、Fig.5 に示すような三角分布を用いた。

$$p_T(Dc_w) = -2Dc_w / (Drg_w + 1)^2 + 2 / (Drg_w + 1) \quad (7)$$

一方、認識領域内外で winner に対応する入力パターンが多い場合や、 $Counter2_w$ 同様 ($Counter1_w + Counter2_w$) が大きい場合、しきい値 D_{thw} は大きくなる。この目的のため、我々は非認識 winner の差の平均 $Dg_w (=Ds_w/N)$ を計算する。 Dc_w の増加量はまた、式(7)によって確率的に選択される。それぞれの学習ステップが終了すると、 $Counter1_w$, $Counter2_w$, Drs_w , Ds_w , Drg_w , Dg_w はリセットされる。

(3)学習の制御方法

マンハッタン距離 G_{tw} がしきい値と比較して小さい場合、参照パターンとしきい値の学習は共に行われない。 G_{tw} は $G_{m_w} = (G_{m_w,1}, G_{m_w,2}, \dots, G_{m_w,j}, \dots,$

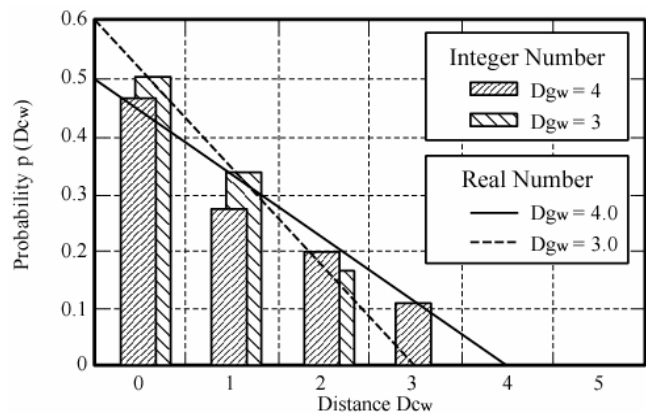


Fig. 5: The triangle distribution for updating threshold value.

$G_{w,c}$ と”0”ベクトルの差である。

G_{t_w} が小さいということは、参照データもしきい値も誤差の少ない値であるということであり、更新する必要がないことを表している。

$$G_{t_w} = \frac{1}{N} \sum_{j=1}^c |G_{m_{w,j}}| \quad (8)$$

2. 3. シミュレーションによる評価

2. 3. 1. 短期長期記憶に基づくメモリベース学習でのシミュレーション

提案学習アルゴリズムの有効性を検証するためにC言語によりシミュレータを作成し、動作レベルのシミュレーションを行った。連想メモリをデータ長256bit, 参照データ数32として、アルゴリズム中のパラメータを $J_L=6$, $J_S=3$, しきい値=10, 長期記憶数 $N_L=20$, 短期記憶数 $N_S=12$ と設定して、メモリにはあらかじめ参照データが書き込まれている状態とした。そこへ、256bit 長の新たなデータ20個と”1,0”をランダムに選択し作成したノイズデータを入力データとして与えた場合に、長期記憶(上位20個の参照データ)に学習された新たなデータの個数とそれに要した検索回数のシミュレーション結果を Fig.6 に示す。ここで図中の(b)は、短期・長期記憶の概念を用いない方法で、新しいパターンと認識された場合には、最上位rankに新規データを追加する方法に基づく結果である。この結果より、提案アルゴリズムではノイズと見なされたデータは短期記憶として一時的に記憶されるだけで、ある程度の頻度で現われるデータは長期記憶として全て学習され、ノイズデータは長期記憶として学習されることはほとんどないことがわかる。

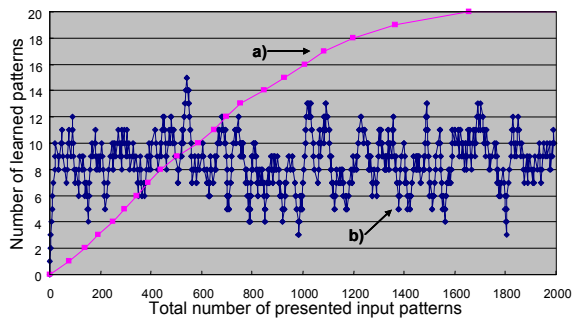


Fig. 6: Simulation result.

a) Learning algorithm based on short/long term storage concept.
b) Without short/long term storage (unknown data is inserted at the top rank).

Fig.7 にテストチップのアーキテクチャを示す。このアーキテクチャは、大まかに連想メモリブロック、ランク構成回路と自動学習回路に分けられる。

上述の提案アルゴリズムを実現したテストチップを0.35 μ m CMOS 技術を用いて試作を行った。Fig.8 にテストチップのレイアウトを、表1に諸元を示す。最小距離検索を行なうマンハッタン距離連想メモリ

とアルゴリズムにおける最小距離検索以外の処理を担当する自動学習回路ブロックから構成される。連想メモリは250nsec以内に最小距離検索を終え、自動学習回路は最大動作周波数166MHz(ゲートレベルシミュレーション)で動作する。

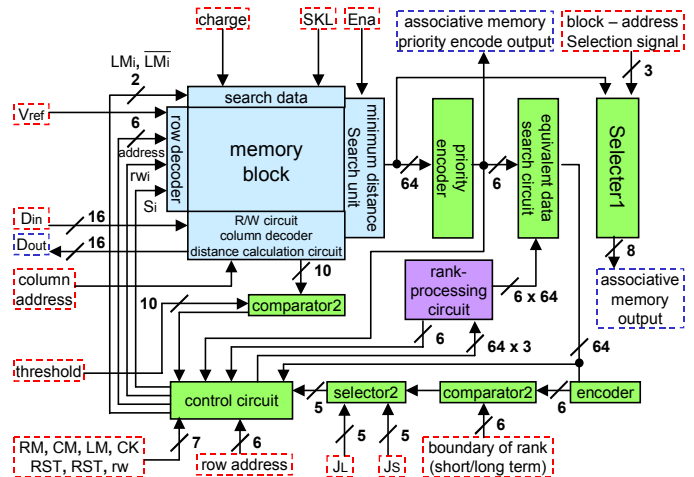


Fig.7: Associative-memory-based automatic pattern learning architecture, with 64 patterns.

Long/short-term-memory size, parameters J_L , J_S and the threshold in the algorithm can be set externally.

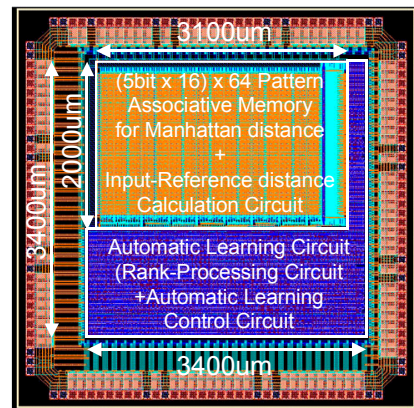


Fig.8: Layout of test chip.

Table. 1: Characteristics of the designed test chip.

Distance Measure	Manhattan Distance (5bit x 16)
Reference Patterns	64
Short Term Storage	24 (Default, Variable)
Long Term Storage	40 (Default, Variable)
Nearest-Match Range	0 to 496
Technology	0.35 μ m, 2-poly 3-metal, CMOS
Supply Voltage	3.3V
Number of Transistors	402,768
Design Area	11.04mm ²
Associative Memory	6.2mm ²
Automatic Learning Circuit	4.84mm ²
Automatic Learning Algorithm Processing Time	< 290nsec (search time 250nsec)
Automatic Learning Circuit Max Operation Frequency	166MHz (gate level simulation)

2. 3. 2. 参照データ最適化学習法のシミュレーション

提案した参照データ最適化学習がいかに有効かを表す指標として、誤差率 α_w [5]を用いる。このパラメータが”0”に近いほど最適化が行われていることを示す。Fig.9とFig.10は入力データが正規分布の場合と一様分布の場合を、それぞれ本学習アルゴリズムとk-means法を比較して表したグラフである。Fig.9から両アルゴリズムも同等の学習効果が得られていることがわかる。この場合本アルゴリズムでは一定回数 $N=8$, k-means法では荷重 $\epsilon=0.5$ とした。提案アルゴリズムは複雑でないマンハッタン距離を適用し、k-means法のような複雑なユークリッド距離を用いなかった点に注目したい。

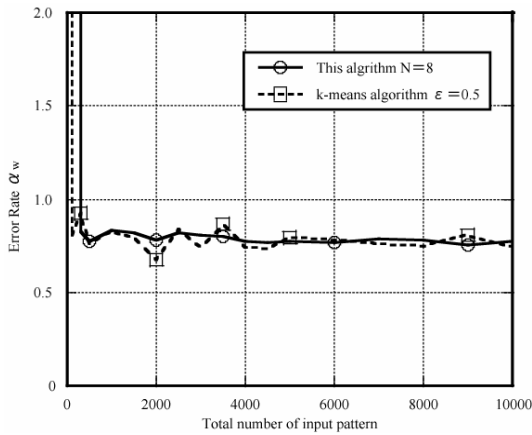


Fig. 9: Performance of the proposed algorithm in comparison to the k-means algorithm for a Gaussian input-pattern distribution.

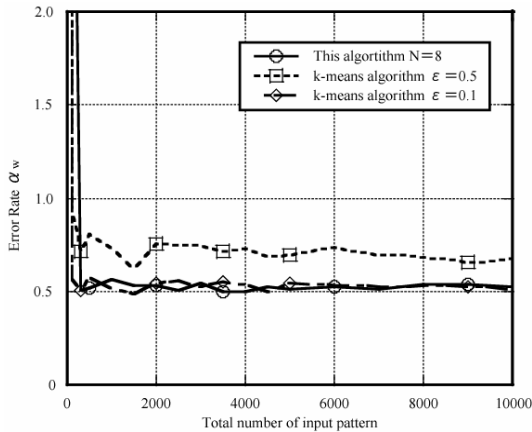


Fig. 10: Performance of the proposed algorithm in comparison to the k-means algorithm for a Homogeneous input-pattern distribution.

更に入力パターンが一様分布である場合 (Fig.10), k-means法では正規分布と同様の荷重 $\epsilon=0.5$ では、本アルゴリズムほど良い結果を得られていない。本アルゴリズムと同程度の誤差率 α_w になるには、k-means法の荷重は $\epsilon=0.1$ でなければならない。結果として、提案アルゴリズムは未知の入力パターンの分布において、k-means法よりも良い結果を得られる。

本提案アルゴリズムの最大の利点は、マンハッタン距離検索全並列型連想メモリを用いたVLSIでの実現のしやすさである。ユークリッド距離のため、k-means法は大変遅く、複雑なハードウェアを必要とする。Fig.11に本アルゴリズムのVLSIアーキテクチャのブロック図を示す。ただし、このアーキテクチャでは整数のみを用いたアーキテクチャである。更に学習回路のアーキテクチャの詳細図をFig.12に示す。これには、連想メモリ、比較器、メモリ、カウンタ、減算器、加算器、乗算器、乱数発生器、確率分布メモリ、REG(レジスタ)、スイッチが必要である。

しきい値学習に用いられる乱数発生器と確率分布メモリの説明を簡単にする。しきい値更新量は、確率分布の値を記憶したコードブックメモリ (Fig.13) から乱数が選択される。

例えば”0”から”100”の数のうち、乱数発生器から”50”という数が選択されると、しきい値更新量 $D_{cw}=1$ は確率分布メモリから選択される (Fig.13)。このように、提案アルゴリズムのしきい値学習部は簡単にハードウェア化できる。

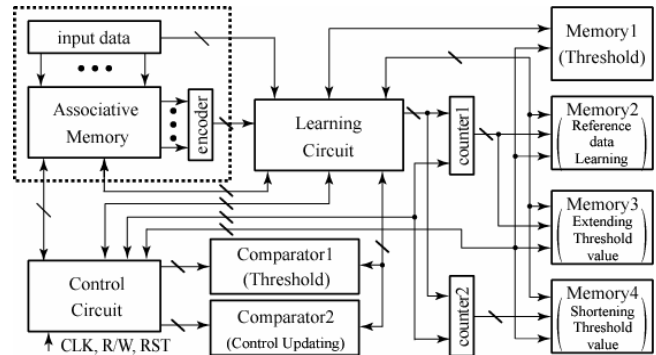


Fig. 11: The associative memory with learning function architecture.

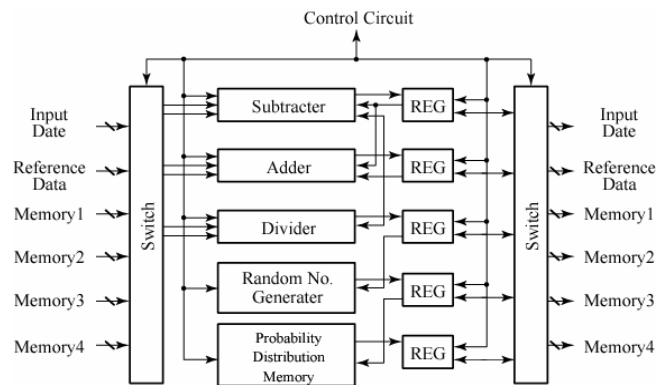


Fig. 12: The learning circuit architecture.

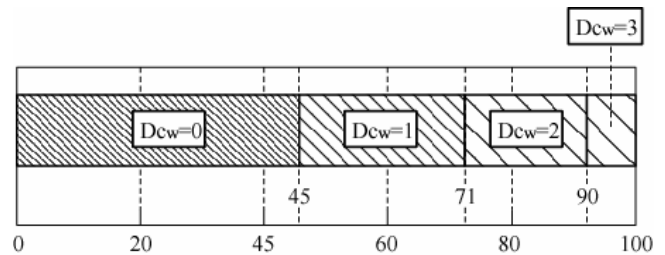


Fig. 13: Contents for random number generation in the range "0" to "100".

3. まとめ

本研究では参照データの自動的な学習と最適化のためのアルゴリズム及び、その VLSI アーキテクチャを提案した。これらのアーキテクチャは、連想メモリベースのパターン認識システムに応用する予定である。アルゴリズムの一部の VLSI アーキテクチャについては、ローム(0.35 μ m)CMOS 技術を用いて設計、試作し、アーキテクチャレベルでは動作を確認した。テストチップのシミュレーションから、通常の学習に要するサイクルは約 300ns と大変高速であり、おそらく全てのアプリケーションに有効であると考えられる。

提案したアルゴリズムではマンハッタン距離指標のみを用いたが、パターン学習の性能は、従来のユークリッド距離を用いる複雑な計算による k-means 法と同様に良い。k-means 法では全並列処理可能な効果的なハードウェアが知られていないため、提案アーキテクチャは非常に有効であると言える。

4. 今後の予定

今後の課題は、自動参照パターン学習とその最適化のために、パターン最適化を含めたアーキテクチャ全体のテストチップ設計することである。そしてテストチップの動作を確認し、提案アーキテクチャの有効性を実験的に評価、検証することである。

更に、動体検出と認識システムの応用に対して、提案アーキテクチャの適用を現在行っている。これらの応用において、児童参照パターン学区集アーキテクチャのシステムレベルでの適合性と性能を今後評価する予定である。

参考文献

- [1] H. J. Mattausch et al. , "Compact associative-memory architecture with fully-parallel search capability for the minimum hamming distance", IEEE Journal. of Solid-State Circuits, Vol.37, pp.218-227, 2002.
- [2] H. J. Mattausch et al. , "Fully-parallel pattern-matching engine with dynamic adaptability to Hamming or Manhattan distance", 2002 Symposium on VLSI Circuit Dig. of Tech. Papers, pp.252-255, 2002.
- [3] H. J. Mattausch et al. , "An architecture for Compact Associative Memories with Deca-ns Nearest-Match Capability up to Large Distance", ISSCC Dig. of Tech. Papers, pp.170-171, 2001.
- [4] 岩田 穆, 雨宮 好仁 編著, "ニューラルネットワーク LSI", 電子情報通信学会, 1995.
- [5] T. M. Martinez, S. G. Berkovich, and K. J. Schulten, "'Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction," IEEE Trans. Neural Networks, vol. 4, pp.558-569, 1993.

5. これまでの研究発表、特許等

① 国際会議プロシーディング等

1. Y. Shirakawa, H. J. Mattausch and T. Koide,

"Reference-Pattern Learning and Optimization from an Input-Pattern Stream for Associative-Memory-Based Pattern-Recognition Systems", 47th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'04), in press, 2004.

2. Y. Shirakawa, M. Mizokami, H. J. Mattausch and T. Koide, "Automatic Pattern-Learning Architecture Based on Associative Memory and Short/Long Term Storage Concept", 2004 Solid State Devices and Materials International Conference (SSDM'04), submitted, 2004.

②特許

1. マタウシュ ハンスユルゲン, 小出哲士, 溝上政弘, "参照データ認識・学習方法及びパターン認識システム", 日本特願 2003-434596 (2003.12.26).
2. マタウシュ ハンスユルゲン, 小出哲士, 白川佳則 "参照データ最適化学習方法とパターン認識システム", 日本特願 2004-053433 (2004.2.27).
3. H. J. Mattausch, T. Koide and M. Mizokami, "Reference Data Recognition and Learning Method and Pattern Recognition System," USA, EPC, KOR, TWN, Patent Application No. TBD. (2004, 6. 15).

③その他

1. 溝上 政弘, 白川 佳則, 小出 哲士, H. J. Mattausch, "集積化認識システムのための連想メモリベース自動学習アーキテクチャ," 2004 年度電子情報通信学会総大会, No.C-12-18, p.120, 2004.
2. 溝上 政弘, 矢野 祐二, 本田 稔, H. J. Mattausch and T. Koide, "長い参照パターンを並列可能なデジタル・アナログ混載連想メモリの性能評価," 平成 14 年度電気・情報関連学会中国支部連合大会, 101309, pp.283-284, 2002.