

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-242808
(P2005-242808A)

(43) 公開日 平成17年9月8日(2005.9.8)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06T 7/00	G06T 7/00 350B	5B056
G06F 17/15	G06F 17/15	5L096
G06N 3/00	G06N 3/00 560A	

審査請求 有 請求項の数 23 O L (全 21 頁)

(21) 出願番号	特願2004-53433 (P2004-53433)	(71) 出願人	504136568 国立大学法人広島大学 広島県東広島市鏡山1丁目3番2号
(22) 出願日	平成16年2月27日 (2004.2.27)	(74) 代理人	100058479 弁理士 鈴江 武彦
		(74) 代理人	100091351 弁理士 河野 哲
		(74) 代理人	100088683 弁理士 中村 誠
		(74) 代理人	100108855 弁理士 蔵田 昌俊
		(74) 代理人	100075672 弁理士 峰 隆司
		(74) 代理人	100109830 弁理士 福原 淑弘

最終頁に続く

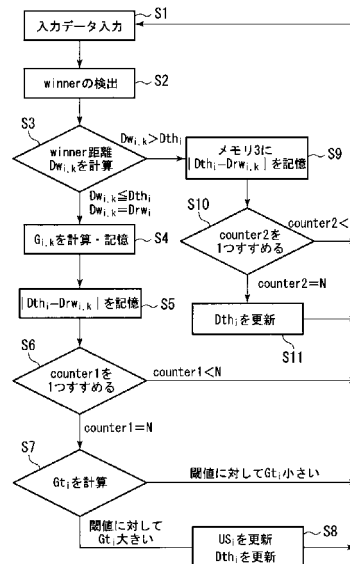
(54) 【発明の名称】 参照データ最適化学習方法とパターン認識システム

(57) 【要約】

【課題】 パターンマッチング処理に不可欠な新規参照データの認識を比較的短時間で自動的に学習させる。

【解決手段】 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(図3中の(a))。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に得ることができない。そこで本発明では、認識winner とされた参照データに対応する一定回数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。また誤認識を避け、認識率を向上させるため、認識領域を縮小、拡大する。このような最適化を施しておくことになり、何らかの影響で入力データがばらついたりした場合にも、その分布の変化に十分対応することができる。

【選択図】 図7



【特許請求の範囲】

【請求項 1】

パターン検索対象である入力データと複数の参照データそれぞれとの間で類似度に相当する距離演算を行い、閾値によって決まる認識領域内に存在する最小距離の参照データを認識データとして選択するパターン認識に用いられ、前記認識データの学習によって参照データを前記パターン認識用に最適化する参照データ最適化学習方法であって、

前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化することを特徴とする参照データ最適化学習方法。

10

【請求項 2】

前記参照データの位置の最適化に際し、当該参照データの認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする請求項 1 記載の参照データ最適化学習方法。

【請求項 3】

前記参照データの位置の最適化に際し、互いに隣接する参照データ間でそれぞれに設定される認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする請求項 1 記載の参照データ最適化学習方法。

【請求項 4】

パターン検索対象の入力データを複数の参照データそれぞれとの間で類似度に相当する距離演算を行い、閾値によって決まる認識領域内で最小距離にある参照データを認識データとして選択するパターン認識に用いられ、前記認識データの学習によって参照データを前記パターン認識用に最適化する参照データ最適化学習方法であって、

20

前記参照データの位置の最適化に際し、前記認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行うことを特徴とする参照データ最適化学習方法。

【請求項 5】

前記閾値の更新量を予め設定された確率分布から選択することを特徴とする請求項 3 または 4 記載の参照データ最適化学習方法。

【請求項 6】

前記閾値の変更は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする請求項 3 または 4 記載の最適化学習方法。

30

【請求項 7】

前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする請求項 1 記載の最適化学習方法。

【請求項 8】

少なくとも、入力パターンの検索データを認識するための複数の参照データを記憶する参照データ記憶手段、前記検索データと最小距離にある参照データを求める最小距離検索手段、及び前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する判断手段を備える連想メモリと、

40

前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化する最適化手段とを具備し、

前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とするパターン認識システム。

【請求項 9】

前記最適化手段は、前記参照データの閾値によって決まる認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする請求項 8 記載のパターン認

50

識システム。

【請求項 10】

前記最適化手段は、互いに隣接する参照データ間でそれぞれの閾値によって決まる認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする請求項 8 記載のパターン認識システム。

【請求項 11】

少なくとも、入力パターンの検索データを認識するための複数の参照データを記憶する参照データ記憶手段、前記検索データと最小距離にある参照データを求める最小距離検索手段、及び前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する判断手段を備える連想メモリと、

10

前記閾値によって決まる認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行う最適化手段とを具備し、

前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とするパターン認識システム。

【請求項 12】

前記最適化手段は、前記閾値の更新量を予め設定された確率分布から選択することを特徴とする請求項 10 または 11 記載のパターン認識システム。

【請求項 13】

前記最適化手段は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする請求項 10 または 11 記載のパターン認識システム。

20

【請求項 14】

前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする請求項 8 または 11 記載のパターン認識システム。

【請求項 15】

前記連想メモリ及び最適化手段は、1 チップ集積回路に組み込まれることを特徴とする請求項 8 または 11 記載のパターン認識システム。

【請求項 16】

30

プログラムに従って演算処理を実行するプロセッサ、少なくとも、入力パターンの検索データを認識するための複数の参照データ、前記プロセッサの演算処理結果、前記演算処理に用いるパラメータを記憶する記憶手段を備え、前記プロセッサが、前記プログラムによって、前記検索データと最小距離にある参照データを求め、前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する演算処理装置と、

前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化する最適化手段とを具備し、

前記最適化された参照データで前記記憶手段に記憶された参照データを更新することを特徴とするパターン認識システム。

40

【請求項 17】

前記最適化手段は、前記参照データの閾値によって決まる認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする請求項 16 記載のパターン認識システム。

【請求項 18】

前記最適化手段は、互いに隣接する参照データ間でそれぞれの閾値によって決まる認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする請求項 16 記載のパターン認識システム。

【請求項 19】

50

プログラムに従って演算処理を実行するプロセッサ、少なくとも、入力パターンの検索データを認識するための複数の参照データ、前記プロセッサの演算処理結果、前記演算処理に用いるパラメータを記憶する記憶手段を備え、前記プロセッサが、前記プログラムによって、前記検索データと最小距離にある参照データを求め、前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する演算処理装置と、

前記閾値によって決まる認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行う最適化手段とを具備し、

前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とするパターン認識システム。 10

【請求項 20】

前記最適化手段は、前記閾値の更新量を予め設定された確率分布から選択することを特徴とする請求項 18 または 19 記載のパターン認識システム。

【請求項 21】

前記最適化手段は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする請求項 18 または 19 記載のパターン認識システム。

【請求項 22】

前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする請求項 16 または 19 記載のパターン認識システム。 20

【請求項 23】

前記演算処理装置及び最適化手段は、1チップ集積回路に組み込まれることを特徴とする請求項 16 または 19 記載のパターン認識システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、パターンマッチング処理のための参照データを追加して認識させる参照データ最適化学習方法と、この方法を用いて参照データの認識・学習を行うパターン認識システムに関する。 30

【背景技術】

【0002】

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。従来システムでは、プロセッサをベースに実現したモデルが最も一般的である。このモデルでは、入力パターンのデータをメモリ内の複数の参照データと逐次比較して、最も類似したパターンを示す参照データを検出する手法をとっている。しかしながら、このモデルの手法には、逐次比較を要することから多くの処理時間を要し、しかも比較データ数の増加に比例して、その処理に要する時間が増大してしまうという問題がある。また、参照データについては、逐次比較されるため、比較順序の入れ替え、新規な参照データの登録において学習する概念はなく、学習による処理時間の短縮効果は得られない。 40

【0003】

パターン認識と学習は、人間の脳と同じような能力を持つ人工知能システムを構築するため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（例えば非特許文献 1 参照）。

【0004】

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 50

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。

【0005】

近年のニューラルネットワークを用いたハードウェアによる実現方法に関する進展は、当初期待されたものよりも遅れており、集積度や消費電力の観点からLSI化が困難な状況にある。このことから、メモリ要素を組み込んで効率よくパターンを記憶可能とする、より優れた新しいハードウェアでの実現方法が要望されている。

10

【0006】

以上の諸事情を考慮し、近年では、全並列アーキテクチャによる小面積・高速連想メモリ（例えば非特許文献2-5、特許文献1-3参照）の開発が進められている。この連想メモリは、認識機能実現に重要な要素である最小距離検索機能を有しており、検索データと各参照データの比較をデジタルによる比較計算とアナログによる最小値計算により行うことにより、全てのデータ間の比較を並列に処理することが実行可能である。このことから、高速なパターンマッチング処理を実現できるものとして注目されている。

【0007】

しかしながら、この連想メモリをベースとしたパターン認識システムにおいても、新たな参照データを認識するように効率よく学習させるための具体的な手法が確立されておらず、参照データの追加または入れ替えのための処理にかなりの時間を要してしまう。

20

【非特許文献1】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。

【非特許文献2】H. J. Mattausch, N. Omori, S. Fukae, T. Koide, and T. Gyoten, "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.

【非特許文献3】T. Koide, Y. Yano, H. J. Mattausch, "An associative memory for real-time applications requiring fully parallel nearest Manhattan-distance-search", Proc. of the Workshop on Synthesis And System Integration of Mixed Technologies, pp.200-205, 2003.

30

【非特許文献4】H.J. Mattausch, T. Gyoten, Y. Soda, T. Koide, "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.

【非特許文献5】Y. Yano, T. Koide, H. J. Mattausch, "Fully parallel nearest Manhattan-distance search memory with large reference-pattern number", Extend. Abst. of the International Conf. on Solid-State Devices and Materials, pp.254-255, 2002.

【特許文献1】特願2001-011760号

【特許文献2】特願2002-159436号

40

【特許文献3】特願2002-165759号

【発明の開示】

【発明が解決しようとする課題】

【0008】

以上述べたように、人工知能システムには、入力環境の変化に対応してパターン認識を行うための機能が必要であり、学習によって幅広い認識が行えるようにすることが重要となるが、従来のシステムでは、プロセッサベースやニューラルネットワークベースの方法で学習を行っている。学習を実現するためにはプロセッサベースシステムでは、何らかの方法で参照データの更新を行う必要があり、ニューラルネットワークベースシステムでは、ネットワーク全体の各ネット荷重を更新する必要がある。そのため、パターン認識と学

50

習を合わせて行う場合には、上述の従来システムでは、消費電力や回路規模の面で集積化が困難といった問題がある。一方、高速マッチング処理の実現を図った連想メモリベースのパターン認識システムが提案されているが、新たに追加される参照データを認識するように効率よく学習させるための具体的な手法が確立されておらず、参照データの認識・学習のための処理にかなりの時間がかかっている。

【0009】

本発明は、上記の事情に鑑みてなされたもので、パターンマッチング処理に不可欠な新規参照データの認識を比較的短時間で自動的に学習させることのできる参照データ最適化学習方法と、この方法を採用してパターン認識に要する時間を短縮可能としたパターン認識システムを提供することを目的とする。

10

【課題を解決するための手段】

【0010】

上記の目的を達成するために、本発明に係る参照データ最適化学習方法は、以下のような特徴を有する。

【0011】

(1) パターン検索対象である入力データと複数の参照データそれぞれとの間で類似度に相当する距離演算を行い、閾値によって決まる認識領域内に存在する最小距離の参照データを認識データとして選択する場合に、前記認識データの学習によって参照データを前記パターン認識用に最適化する参照データ最適化学習方法であって、前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化することを特徴とする。

20

【0012】

(2) (1)において、前記参照データの位置の最適化に際し、当該参照データの認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする。

【0013】

(3) (1)において、前記参照データの位置の最適化に際し、互いに隣接する参照データ間でそれぞれに設定される認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする。

【0014】

(4) パターン検索対象の入力データを複数の参照データそれぞれとの間で類似度に相当する距離演算を行い、閾値によって決まる認識領域内で最小距離にある参照データを認識データとして選択するパターン認識に用いられ、前記認識データの学習によって参照データを前記パターン認識用に最適化する参照データ最適化学習方法であって、前記参照データの位置の最適化に際し、前記認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行うことを特徴とする。

30

【0015】

(5) (3)または(4)において、前記閾値の更新量を予め設定された確率分布から選択することを特徴とする。

40

【0016】

(6) (3)または(4)において、前記閾値の変更は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする。

【0017】

(7) (1)において、前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする。

【0018】

また、本発明に係るパターン認識システムは、以下のような特徴を有する。

50

【0019】

(8) 少なくとも、入力パターンの検索データを認識するための複数の参照データを記憶する参照データ記憶手段、前記検索データと最小距離にある参照データを求める最小距離検索手段、及び前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する判断手段を備える連想メモリと、前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化する最適化手段とを具備し、前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とする。

【0020】

(9) (8)において、前記最適化手段は、前記参照データの閾値によって決まる認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする。

【0021】

(10) (8)において、前記最適化手段は、互いに隣接する参照データ間でそれぞれの閾値によって決まる認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする。

【0022】

(11) 少なくとも、入力パターンの検索データを認識するための複数の参照データを記憶する参照データ記憶手段、前記検索データと最小距離にある参照データを求める最小距離検索手段、及び前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する判断手段を備える連想メモリと、前記閾値によって決まる認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行う最適化手段とを具備し、前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とする。

【0023】

(12) (10)または(11)において、前記最適化手段は、前記閾値の更新量を予め設定された確率分布から選択することを特徴とする。

【0024】

(13) (10)または(11)において、前記最適化手段は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする。

【0025】

(14) (8)または(11)において、前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする。

【0026】

(15) (8)または(11)において、前記連想メモリ及び最適化手段は、1チップ集積回路に組み込まれることを特徴とする。

【0027】

(16) プログラムに従って演算処理を実行するプロセッサ、少なくとも、入力パターンの検索データを認識するための複数の参照データ、前記プロセッサの演算処理結果、前記演算処理に用いるパラメータを記憶する記憶手段を備え、前記プロセッサが、前記プログラムによって、前記検索データと最小距離にある参照データを求め、前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する演算処理装置と、前記認識データとして選択されるべき参照データに対応する入力データを一定回数分取り込み、各入力データの前記参照データとの距離分布の重心を求め、当該重心点に位置するように前記参照データの位置を最適化する最適化手段とを具備し、前記最適化された参照データで前記記憶手段に記憶された参照データを更新することを特徴とする。

10

20

30

40

50

【0028】

(17)(16)において、前記最適化手段は、前記参照データの閾値によって決まる認識領域が隣接参照データの認識領域と重ならないように移動を制限することを特徴とする。

【0029】

(18)(16)において、前記最適化手段は、互いに隣接する参照データ間でそれぞれの閾値によって決まる認識領域が重ならないように閾値を変更して認識領域を拡大または縮小することを特徴とする。

【0030】

(19)プログラムに従って演算処理を実行するプロセッサ、少なくとも、入力パターンの検索データを認識するための複数の参照データ、前記プロセッサの演算処理結果、前記演算処理に用いるパラメータを記憶する記憶手段を備え、前記プロセッサが、前記プログラムによって、前記検索データと最小距離にある参照データを求め、前記最小距離が閾値以上か否かに応じて前記検索データと最小距離の参照データとの同一性を判断する演算処理装置と、前記閾値によって決まる認識領域内にある入力データ数と領域外にある入力データ数をそれぞれカウントし、先に前記一定回数に達する側に前記閾値を変更して前記認識領域の拡大または縮小を行う最適化手段とを具備し、前記最適化された参照データで前記参照データ記憶手段に記憶された参照データを更新することを特徴とする。

10

【0031】

(20)(18)または(19)において、前記最適化手段は、前記閾値の更新量を予め設定された確率分布から選択することを特徴とする。

20

【0032】

(21)(18)または(19)において、前記最適化手段は、前記入力データの集団の中心と現在の参照データとの距離を示す値を基準値と比較して、基準値を越えるときのみ前記閾値の変更を実施することを特徴とする。

【0033】

(22)(16)または(19)において、前記距離演算に供される距離の指標には、ユークリッド距離、マンハッタン距離、ハミング距離、マハラノビス距離のいずれかを用いることを特徴とする。

【0034】

(23)(16)または(19)において、前記演算処理装置及び最適化手段は、1チップ集積回路に組み込まれることを特徴とする。

30

【発明の効果】

【0035】

本発明によれば、パターンマッチング処理に不可欠な新規参照データの認識を比較的短時間で自動的に学習させることのできる参照データ最適化学習方法と、この方法を採用してパターン認識に要する時間を短縮可能としたパターン認識システムを提供することができる。

【発明を実施するための最良の形態】

【0036】

以下、図面を参照して、本発明を実施するための最良な形態を詳細に説明する。

40

【0037】

本発明に係る方法が適用される連想メモリは、図1に示すように、予めメモリ内に蓄えられた複数の参照データ(テンプレートとも呼ばれる)と入力データ(検索データとも呼ばれる)を、予め指定した距離指標に基づいて類似度で比較し、入力データと最も類似する(最も距離が小さい)参照データ(winner)を検出することでパターンマッチングを行う機能メモリである。

【0038】

図2は上記連想メモリの一例として、高速処理・小面積・低消費電力の実現に向けて開発が進められている、全並列最小距離検索を行うタイプ(全並列型連想メモリ)の構成例

50

を示すブロック図である。図において、メモリコア部分は、メモリ領域 1 0 0、ウィナー・ラインナップ増幅回路（以下、W L A 回路（Winner Line-Up Amplifier））2 0 0、全ウィナー取得回路（以下、W T A 回路（Winner Take All Circuit））3 0 0 で構成されており、また周辺回路として行デコーダ及び R / W 回路（Column Decoder and Read/Write Circuit（K x W Column））1 1 0、列デコーダ（Row Decoder（R Rows））1 2 0、検索データ保存回路（Search Data（K bit x W））1 3 0 を持つ。

【 0 0 3 9 】

メモリ領域 1 0 0 は、参照データをユニット（k ビット）単位で保存するための S R A M セルで形成される W x R 個のユニット保存回路（Unit Storage：U S）、ユニット毎に参照データと検索データの差の絶対値（後述のマンハッタン距離）を計算する W x R 個のユニット比較回路（Unit Comparison：U C）、計算した距離をアナログ電圧（あるいは電流）に変換する R 個のワード重み比較回路（Word Comparison：W C）により構成されている。

10

【 0 0 4 0 】

ワード重み比較回路 W C で生成された信号 C_i （Comparison Signal）は W L A 回路 2 0 0 に入り、W L A 回路 2 0 0 がこの信号 C_i を回路自身のバランスにより制御し、最初の段階で各行の電圧の差を最も大きく増幅する。W L A 回路 2 0 0 と W T A 回路 3 0 0 は、面積増加の割合が行数 R に対して線形 O（R）の小面積で実現可能であるという特徴がある。

【 0 0 4 1 】

W T A 回路 3 0 0 は W L A 回路 2 0 0 により増幅された各行の電圧出力 $L A_i$ の差を更に増幅する機能を持つ。W T A 回路 3 0 0 の出力 M_i においては、winner（ウィナー）行が“ 1 ”、その他の loser（ルーザ）行が“ 0 ”のデジタル信号となる。W L A 回路 2 0 0 は、内部のフィードバック信号 F（Feedback Signal）を増幅点の winner 行が最大の増幅を得られるようにするために用い、ワード重み比較回路 W C に返す際に、当該 W L A 回路 2 0 0 に内蔵された電圧フォロワ（Voltage Follower）回路を用いることにより、フィードバックを高速化している。

20

【 0 0 4 2 】

ここで、上記マンハッタン距離 D は、ベクトル $a = (a_1, a_2, \dots, a_i, \dots, a_N)$ 、 $b = (b_1, b_2, \dots, b_i, \dots, b_N)$ があるとき、次式のように表される。

30

【 数 1 】

$$D = \sum_{i=1}^N |a_i - b_i|$$

【 0 0 4 3 】

図 3 は本発明に係る最適化学習アルゴリズムの概念図を示すもので、ここではパターンを色調と形状の 2 次元で分類する場合を例にとって表す。図 3 中の左上（a）において、入力データ（八角形のマーク）と予め蓄えられた 4 つの参照データ A、B、C、D のうち、最小のマンハッタン距離 1 を持つものを最も類似しているパターン（以下、winner と呼ぶ）として選択する。この場合の winner は A の八角形のマークである。このときの最小マンハッタン距離を「winner 距離 $D w_i$ 」とする。

40

【 0 0 4 4 】

winner 距離 $D w_i$ が参照データ i に対して予め指定した閾値距離 $D t h_i$ 以下（ $D w_i < D t h_i$ ）であれば、winner は認識された winner としてみなされる。この winner 距離 $D w_i$ を「認識 winner 距離 $D r w_i$ 」と呼ぶ。このように、最も類似しているパターンが予め指定していた閾値内に存在する場合に、連想メモリによってパターンが認識される。これに対して $D w_i$ が $D t h_i$ より大きければ（ $D w_i > D t h_i$ ）、その入力データは該当する参照データと類似していないと見なされ、入力データはどの参照データにも類似していないと見なされ、認識されないことになる。但し、連想メモリでは参照データと入力データ

50

を比較することでwinner を検索し、winner 距離 $D w_i$ を求めるため、予め最適な参照データを用意しておくことが効果的な認識のためには重要である。例えば、図 3 中の中央 (b) において、入力データの分布が図のような場合には、参照データ B に示した入力データ集団の中心が最適な参照データ B である方が望ましい。

【 0 0 4 5 】

また、「認識winner 距離 $D r w_i$ 」は閾値 $D t h_i$ とwinner 距離 $D w_i$ とを比較することで認識されるか否かを判断するため、最適な閾値 $D t h_i$ の設定も認識の向上のためには重要である。例えば、閾値 $D t h_i$ について、ある参照データ i に対応する (類似している) 入力データを全て (1 0 0 %) 認識可能とする閾値を $D m a x_i$ と仮定すると、 $D t h_i$ が $D m a x_i$ に比べて小さ過ぎる場合には認識率が低下してしまい、また、逆に大き過ぎて他の参照データと一致してしまうと誤認識してしまう場合がある (図 3 中の (c) , (d))。そのため参照データに閾値 $D t h_i$ の領域 (認識領域) が重なり合うと、その重なり度合いから間違った認識を行ってしまう可能性があるため、認識領域の重なり合いを解消することが必要である。

10

【 0 0 4 6 】

このような重なり合いの解消は、図 4 中の下部 (c) に示すように参照データをずらしたり (C C , D D)、図 5 中の下部 (c) のように参照データ i の互いの閾値 $D t h_i$ を縮小したりすることで実現することができる。このように、連想メモリベースシステムにおいて認識を正しく行うための各参照データ i に対する閾値 $D t h_i$ の最適化による認識領域の最適化が必要である。

20

【 0 0 4 7 】

そこで本発明の最適化学習とは、参照データを最適なものに更新する「参照データ最適学習」と、各参照データの持つ閾値を最適に更新する「認識領域最適学習」の 2 つを意味する。以下、図 6 乃至図 9 を参照してそれぞれの学習アルゴリズムの詳細を説明する。

【 0 0 4 8 】

尚、以下の説明において、 $U S$ 、 $S B$ 、 G はそれぞれ参照データ、入力データ、及び参照データと入力データの差のベクトルを意味するものとする。また、図 6 に示すように、連想メモリ内部に保存している参照データの集合を $U S = \{ U S_1, U S_2, \dots, U S_i, \dots, U S_R \}$ (R は連想メモリのデータ数 (行数)) とする。各参照データはそれぞれ k ビットの W ワードで構成されるものとする。よって、 $U S_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iW})$ (W は連想メモリのベクトル数 (列数)) と表すことができる。参照データ $U S_i$ ($1 \leq i \leq R$) に対して、認識winner として認識された N 個の入力データの集合を $S B_i = \{ S B_{i,1}, S B_{i,2}, \dots, S B_{i,k}, \dots, S B_{i,N} \}$ ($1 \leq k \leq R$) と定義し、さらに入力データは、 $S B_{i,k} = (y_{i1,k}, y_{i2,k}, \dots, y_{ij,k}, \dots, y_{iW,k})$ ($1 \leq j \leq R$) と定義する。

30

【 0 0 4 9 】

図 7 は本発明の参照データ及び認識領域の最適化学習アルゴリズムを示すフローチャートである。

【 0 0 5 0 】

まず、参照データの最適化学習アルゴリズムについて説明する。本アルゴリズムでは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) であるとする (図 3 中の (a))。しかし、入力データが時間とともに連続して入力されるようなオンラインの認識では、最初に全ての入力データの情報が一度にわからないため、一般に最適な参照データを得ることが難しい。そこで本発明では、認識winner とされた参照データに対応する一定回数分の入力データに基づいて、それらの重心を計算し、参照データの最適化を行う。このような処理を行うことにより、何らかの影響で入力データの分布がばらついたりした場合にも、その分布の変化に十分対応することができる。

40

【 0 0 5 1 】

参照データを最適に更新する学習アルゴリズムを、図 7 のフローチャートに従って説明

50

する。まず、USに入力データ $SB_{i,k}$ を入力して(ステップS1)、winnerの検出を実行すると(ステップS2)、 US_i がwinnerになる。次に、winner距離 $D_{wi,k}$ を計算し、閾値 D_{th_i} との比較を行う(ステップS3)。比較の結果、 $D_{wi,k} < D_{th_i}$ であるとき、 $D_{rwi} = D_{wi,k}$ と認識し、入力データ $SB_{i,k}$ と参照データ US_i のベクトル差 $G_{i,k}$ を(1)式により計算して記憶する(ステップS4)。

【数2】

$$G_{i,k} = SB_{i,k} - US_i \quad (1)$$

【0052】

ここで、N回認識winnerとなった(ステップS6, S7)参照データに対応する入力データに基づいて、参照データ US_i を以下の(2)式により更新する(ステップS8)。

【数3】

$$US_i = US_i + \frac{1}{N} \sum_{k=1}^N G_{i,k} \quad (2)$$

【0053】

入力データが入力されるたびに認識が行われ、N回認識winnerとなった(ステップS6, ステップS7)参照データに対し、 $G_{i,k}$ がN回分計算されることになる。 $G_{i,k}$ は入力データと認識winnerとなった参照データの間の差がどの程度あるかを表すベクトルである。N回分の $G_{i,k}$ から式(2)の第二項が求まり、参照データと入力データの重心との差の大きさがわかる。その差が大きければ、その差を小さくするように式(2)から参照データを最適に更新する(ステップS8)。但し、差の大小の判断基準は、N回分の入力データと参照データとの差から求まる距離(例えばマンハッタン距離)と閾値(例えばマンハッタン距離)の割合である。尚、式(2)の第二項に示す大小の判断基準は後述の更新処理で用いられる式(7)と同じである。参照データ更新と閾値更新の手続は同じ処理であり、入力データの重心と参照データの差が大きい場合、参照データと閾値を更新することになる。

【0054】

次に各参照データ US_i が持つ閾値 D_{th_i} を最適化する学習アルゴリズムについて説明する。

【0055】

まず、閾値を変更することは、認識領域を縮小もしくは拡大することに相当する。そのためには、入力データが認識領域内、外にどれほどの割合で入力されるかを判断する必要がある。本発明では、図8に示すように点線で表された認識領域の内側($< D_{th_i}$)、もしくは外側($> D_{th_i}$)に存在する入力データ数をそれぞれカウントし(ステップS6, S10)、予め指定した一定回数入力データがN回存在する方へ認識領域を縮小もしくは拡大する。

【0056】

ここで、ある参照データ US_i がwinnerとなったとき(ステップS2)、すなわちwinner距離が閾値よりも小さい場合には、winner距離を認識winner距離 $D_{rwi,k}$ (1, 2, ..., k, ..., N)とし(ステップS3)、これらの入力データの集合を $SB_i = \{ SB_{i,1}, SB_{i,2}, \dots, SB_{i,k}, \dots, SB_{i,N} \}$ とする。一方、閾値よりも大きい場合、winner距離をwinner距離 $D_{wi,k}$ (1, 2, ..., k, ..., N)とし、これらの入力データの集合を $SBe_i = \{ SBe_{i,1}, SBe_{i,2}, \dots, SBe_{i,k}, \dots, SBe_{i,N} \}$ とする。これらは閾値の更新量 Dc を得るためのパラメータである。 D_{th_i} と D_{wi} もしくは D_{rwi} の差 Dg_i を以下の(3), (4)式により求める(ステップS11)。 Dg_i はN回分の入力データが閾値からどの程度離れているを示し、閾値更新はこの離れを小さくすることである。

10

20

30

40

【数 4】

$$\left\{ \begin{array}{l} Dg_i = \frac{1}{N} \sum_{k=1}^N |Dth_i - Drw_{i,k}| \quad (\text{先に認識領域の内側にN回認識された場合}) \quad (3) \\ Dg_i = \frac{1}{N} \sum_{k=1}^N |Dth_i - Dw_{i,k}| \quad (\text{先に認識領域の外側にN回認識された場合}) \quad (4) \end{array} \right.$$

【0057】

10

閾値の更新量 Dc_i は図 9 に示されるようにあらかじめ設定された確率分布、例えば三角分布の確率分布から選ばれる。尚、図 9 は閾値の更新量 Dc_i の選択される確率分布 (Dc_i) を示しており、A が $Dg_i = 4$ の場合、B が $Dg_i = 3$ の場合を示している。

【数 5】

$$p(Dc_i) = -\frac{2}{(Dg_i + 1)^2} Dc_i + \frac{2}{Dg_i + 1} \quad (5)$$

【0058】

これより閾値は以下のように学習される。

【数 6】

20

$$\left\{ \begin{array}{l} Dth_i = Dth_i - Dc_i \quad (\text{閾値を大きくする場合}) \quad (6) \\ Dth_i = Dth_i + Dc_i \quad (\text{閾値を小さくする場合}) \quad (7) \end{array} \right.$$

【0059】

参照データ及び閾値の両学習において更新処理を行う。今、

【数 7】

$$Gm_i = \sum_{k=1}^N G_{i,k} \quad (8)$$

30

$$= (z_{i1}, z_{i2}, \dots, z_{ij}, \dots, z_{iW}) \quad (9)$$

【0060】

から Gm_i の各次元の平均から求まるマンハッタン距離 Gt_i

【数 8】

$$Gt_i = \frac{1}{N} \sum_{j=1}^W |z_{ij}| \quad (10)$$

40

【0061】

が閾値と比較して小さい場合 (例えば閾値の 2 割程度) は参照データ、閾値ともに更新しない。何故なら、 Gt_i は入力データの集合の中心と現在の参照データそのもののマンハッタン距離、すなわち新しい参照データが現在の参照データ Us_i からどれほど移動するかを表しており、 Gt_i が小さいということは、更新量が少ないため、更新しても認識率の向上はあまり期待できない。そこで、このような更新量が少ない場合には更新を避けることで、更新回数をおさえることとし、効率の良い参照データの最適化を行うこととした。

【0062】

50

次に、上記参照データ及び認識領域の最適化学習アルゴリズムに基づく連想メモリベースシステムのアーキテクチャについて説明する。

【0063】

図10は連想メモリベース最適化学習システムのアーキテクチャを示すブロック図である。アーキテクチャは、連想メモリ装置400に対して、制御回路500、学習回路600、比較回路700、データ保存回路800、保存先選択回路900を備える。

【0064】

連想メモリ装置400は、最小距離検索を行う連想メモリ410、入力データを一時的に保存する入力バッファ420、連想メモリ410で抽出された2進数(winnerのみ"1"、他のアドレス行は"0"を出力する)を行アドレスに変換するエンコーダ(encoder)430を備える。

10

【0065】

制御回路500は、本システム全体の処理を制御する。学習回路600は、連想メモリ装置400の参照データ及び認識領域を決定する閾値を最適な値に学習し更新する。比較回路700は、閾値比較用、更新制御用にそれぞれ比較器710, 720が用意される。データ保存回路800は、閾値記憶用、参照データ学習用、閾値縮小用、閾値拡大用にそれぞれ計算結果を保存するための第1乃至第4のメモリ810~840を備える。保存先選択回路900は、制御回路500及び学習回路600それぞれの処理過程で得られるデータの保存先を選択するための第1及び第2のカウンタ(counter)910, 920を備える。

20

【0066】

図11は上記学習回路600の具体的な構成を示すブロック図である。この学習回路600は、減算器(Subtractor)601、加算器(Adder)602、除算器(Divider)603、乱数発生器604、確率分布格納メモリ605及び各ブロック601~605のデータ入出力に供するレジスタ(REG)611~615、入力バッファ420からの入力データ、エンコーダ430からのwinner行アドレス、データ保存回路800のメモリ810~840からのデータ(以下、メモリ1, メモリ2, メモリ3, メモリ4と記す)を選択的に取り込み、減算器601、加算器602、除算器603に導出する入力選択スイッチ621、レジスタ611~615に取り込まれたデータを連想メモリ装置400、データ保存回路800へ導出する出力選択スイッチ622を備える。

30

【0067】

図10及び図11に示す構成のアーキテクチャにおいて、以下に具体的な動作を説明する。

【0068】

説明を簡単にするため、 $W = 2$ の参照データの*i*行目に $US_i = (80, 110)$ が連想メモリ410に予め蓄えられているとする。まず、閾値 $Dth_i = 10$ が閾値記憶用のメモリ1に蓄えられているとし、入力データ $SB_{i,1} = (75, 112)$ が連想メモリ410に入力されると、連想メモリ410から*i*行がwinnerとなった場合には、*i*行目のみ「1」が出力され、他の行は「0」が出力される。連想メモリ410から $(0, \dots, 1, \dots, 0)^T$ (Tは転置を意味する)がエンコーダ430に入力され、このエンコーダ430から行アドレス*i*が出力される。

40

【0069】

学習回路600に*i*行が入力されると、参照データ*i*行目 $US_i = (80, 110)$ と入力データ $SB_{i,1} = (75, 112)$ の距離を距離指標に基づいて(例えば、マンハッタン距離)計算する。減算器601で各次元の引き算を行う。式(1)より

【数 9】

$$G_{i,k} = SB_{i,k} - US_i \quad (11)$$

$$= (75, 112) - (80, 110) \quad (12)$$

$$= (-5, 2) \quad (13)$$

【0070】

10

が求まる。この計算結果から補数の変換を行った後、加算器602を用いてwinner距離（例えば、マンハッタン距離） $Dw_{i,1} = 7$ を計算する。閾値比較用の第1の比較器710で閾値とwinner距離を比較すると $Dth_i > Dw_{i,1}$ であるので、このwinnerは認識winnerであり、参照データ学習用のメモリ2に $G_{i,1}$ を記憶する。また、 $Dw_{i,1} = Drw_{i,1} = 7$ となり、閾値縮小用のメモリ3に $Drw_{i,1}$ を記憶する。

【0071】

$\{SB_{i,2}, SB_{i,3}, SB_{i,4}\} = \{(86, 112), (81, 114), (77, 105)\}$ と認識winnerとなる入力データが予め決められた一定回数($N = 4$)回入力されると、 $\{G_{i,2}, G_{i,3}, G_{i,4}\} = \{(6, 2), (1, 4), (-3, 5)\}$ と $\{Drw_{i,2}, Drw_{i,3}, Drw_{i,4}\} = \{8, 5, 8\}$ が得られる。参照データ学習用のメモリ2に $G_{m_i} = \{-1, 13\}$ が、閾値縮小用のメモリ2に

【数10】

$$\sum_{k=1}^N |Dth_i - Drw_{i,k}| = 12$$

【0072】

が記憶される。 $N = 4$ 回分入力されたので、減算器601、加算器602、除算器603を用い、式(10)から $Gt_i = 3$ を求める。 $Gt_i = 3$ が更新量として小さいと判断された場合には学習されない。更新前の閾値の2割以下の場合には更新量は小さいと仮定すると、 $Gt_i = 3$ はその値より大きいため、参照データの最適化及び閾値領域の最適化が行われる。式(2)より、参照データは $US_i = (0, 3) + (80, 110) = (80, 113)$ と更新され、閾値は式(3)より $Dg_i = 3$ となる。

【0073】

Dc_i を式(5)で発生させる方法を以下に説明する。図9において、 $Dg_i = 3$ の場合、確率分布格納メモリ605は、 $Dg_i = 3$ に対して $0.00 \sim 0.45$, $0.46 \sim 0.71$, $0.72 \sim 0.91$, $0.92 \sim 1.00$ を予め格納している。このとき、乱数発生器604によって $0.00 \sim 1.00$ の一樣乱数を発生させ、図12に示すように、乱数が $0.00 \sim 0.45$ の場合は $Dc = 0$ 、 $0.46 \sim 0.71$ の場合は $Dc_i = 1$ 、 $0.72 \sim 0.91$ の場合は $Dc_i = 2$ 、 $0.92 \sim 1.00$ の場合は $Dc_i = 3$ が得られる。今、 $Dc_i = 1$ が得られたとすると、式(6)から $Dth_i = 10 - 1 = 9$ と更新される。

【0074】

参照データ更新学習アルゴリズムと、各参照データの持つ閾値更新学習アルゴリズム（認識領域最適化アルゴリズム）をC言語を用いて計算機上に実現し、その有効性を検証するための計算機によるシミュレーションを行った。その結果をそれぞれ図13、図14に示す。ここで学習回数とは、実際に参照データや閾値を更新した回数である。

【0075】

図13のシミュレーション条件は、 $R = 1$, $W = 5$, $N = 4$ とし、入力データのばらつきは正規分布を用いた。正規分布は、図15(a)に示される確率分布であり、中心から

50

のばらつき s は、

【数 1 1】

$$p(s) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{s^2}{2\sigma^2}\right] \quad (14)$$

【0076】

の確率で得られる。このシミュレーションでは $\sigma = 3$ とした。また参照データの最適化が行われていることを確認するため、入力データの中心が変化した場合、その変化に追従し参照データが変化するかを確認した。その方法として、一定回数入力データが入力されると、入力データの中心の値を各次元で「1増やし」少しずつ変化させた。つまり、入力データの中心が初め $(10, 15)$ であった場合には、1回目の変化後は $(11, 16)$ とした。

【0077】

図 14 のシミュレーション条件は、 $R = 1$, $W = 5$, $N = 4$ とし、入力データのばらつきは正規分布と一様分布を用いた。一様分布は図 15 (a) に示される確率分布であり、中心からのばらつき s は、

【数 1 2】

$$p(s) = \frac{1}{2 \times 3\sigma} \quad (15) \quad 20$$

$$-3\sigma \leq s \leq 3\sigma \quad (16)$$

【0078】

の確率で得られる。このシミュレーションでは $\sigma = 3$ とした。またこのシミュレーションでは入力データの変化はさせていない。

【0079】

最適化が行われている度合いを示す指標として、図 13 では誤差率を、図 14 では距離誤差率を用いた。ここで誤差率 α_i は、

【数 1 3】

$$\alpha_i = \frac{|MSE_i - MSE0_i|}{MSE0_i} \quad (17)$$

$$MSE_i = \frac{1}{N} \sum_{k=1}^N \|SB_{i,k} - US_i\|^2 \quad (18)$$

$$MSE0_i = \int \|SB0_i - US0_i\|^2 p(SB0_i) d^W SB0_i \quad (19)$$

【0080】

と表される。 MSE_i は、 i 行の参照データに対する入力データの平均二乗誤差である（一定回数ごと計算される）。 $MSE0_i$ は、理想的な参照データに対する全入力データから得られる推定の平均二乗誤差であり、無限に入力データが入力されると得られる平均二乗誤差 MSE_i となり、生じうる誤差である。つまり、誤差率 α_i は現在の平均二乗誤差 MSE_i が、生じうる誤差 $MSE0_i$ とどの程度離れているかを示す割合である。この誤差率 α_i は「0」に近いほど参照データが最適になっていることを表す。

10

20

30

40

50

【 0 0 8 1 】

距離誤差率は、

【 数 1 4 】

$$\alpha D_i = \frac{|D_{th_i} - D_{max_i}|}{D_{max_i}} \quad (20)$$

【 0 0 8 2 】

と表される。D_{max_i} は認識率が100%となる閾値である。したがって、この距離誤差率は「0」に近いほど認識領域が最適になっていることを表す。図13の結果から、学習回数が増えることで参照データの最適化が行われていることがわかる。また、入力データの分布の変化に対応しても参照データが追従し、入力データの変化に応じた最適な参照データになっていることがわかる。図14では、入力データの入力分布を正規分布と一様分布の2種類を用いており、両分布とも、学習後は認識率が75%以上の閾値となった。ここで図14において正規分布の入力データでの距離誤差率が大きくなっている。その理由は、正規分布の入力では最適な参照データを中心に入力データが偏っているため、閾値は大きくなりやすく、結果として距離誤差率は大きくなったと考えられる。

【 0 0 8 3 】

尚、上記実施形態において、距離指標の例としてマンハッタン距離をあげたが、これ以外にユークリッド距離、ハミング距離、マハラノビス距離なども利用可能である。

【 0 0 8 4 】

また、上記実施形態では、連想メモリ装置をベースとしたパターン認識システムについて説明したが、本発明はこれに限定されるものではなく、例えばプロセッサベースのパターン認識システムにも適用可能である。以下、このプロセッサベースの場合の実施形態について、図16を参照して説明する。

【 0 0 8 5 】

図16は本発明の最適化学習方法を採用するプロセッサベースパターン認識システムの構成を示すブロック図である。図16において、1000はCPU(Central Processing Unit)、1100はメモリ、1300はCPU1000とメモリ1100との間で書き込みデータ、読み出しデータの転送を行うデータバスである。

【 0 0 8 6 】

CPU1000は、ALU(Arithmetic Logic Unit)1010、レジスタ1011~1013、1020...コントローラを備える。コントローラ1020は、予め登録されたプログラムに基づいてALU1010に加減算、比較、除算等を選択的に命令する一方、メモリ1100の読み出し・書き込みを制御する。ALU1010の入出力部にはそれぞれレジスタ1011~1013がバッファとして配置される。

【 0 0 8 7 】

メモリ1100は、入力データS_{B₁} ~ S_{B_R}を一時的に格納する領域1110、参照データU_{S₁₁} ~ U_{S_{RW}}を格納し更新可能とする領域1120、閾値データD_{th₁} ~ D_{th_R}を格納する領域1130、CPU1000の距離計算結果D₁ ~ D_Rを格納する領域1140、Winner距離D_{w_{1,1}} ~ D_{w_{R,N}}を格納する領域1150、Winner距離と閾値との差D_{g₁} ~ D_{g_R}を格納する領域1160、検索データと参照データとの差G_{1,1} ~ G_{R,N}を格納する領域1170、閾値の内側に入力された回数の値(第1カウンタ値)1~Nを格納する領域1180、閾値の外側に入力された回数の値(第2カウンタ値)1~Nを格納する領域1190、確率的な閾値更新量データD_cを格納する領域1200を備え、CPU1000側のコントローラ1020により読み出し・書き込みの制御がなされる。

【 0 0 8 8 】

上記構成において、コントローラ1020の実行プログラムには、先に述べた連想メモリの場合と同様な処理がソフトウェアによって実行される。すなわち、CPU1000において、メモリ1100に対するデータのやりとりにより、パターン検索対象の入力デー

タを複数の参照データそれぞれとの間で類似度に相当する距離演算を行い、閾値によって決まる認識領域内で最小距離にある参照データを認識データとして選択する。

【0089】

ここで、上記コントローラ1020の実行プログラムに、図7に示したフローチャートに基づく参照データの最適化学習処理が組み込んでおく。これにより、メモリ1100内の参照データは、常に最適状態に更新され、パターン認識に要する時間の短縮が図れる。先に述べたように、従来のプロセッサベースシステムでは、参照データの最適化学習による更新処理が組み込まれていなかったため、パターン認識に要する時間の短縮、認識率の向上が困難であったが、本発明の最適化処理を組み込むことによって上記の問題を解決し、パターン認識に要する時間の短縮、認識率の向上を実現することができる。

10

【0090】

尚、上記プロセッサベースシステムにおいても、距離指標の例としてマンハッタン距離以外にユークリッド距離、ハミング距離、マハラノビス距離などを利用できる。

【産業上の利用可能性】

【0091】

本発明は、人工知能システム、自律ロボット、認証システム、IDS（不正侵入検知システム）、データバンクシステム、インターネットルータ、対象認識、電動義手、家事手伝いロボットなどに適用可能である。

【図面の簡単な説明】

【0092】

【図1】本発明に係る方法が適用される連想メモリ概念図。

【図2】図1の連想メモリの一例とする全並列型連想メモリ構成例を示すブロック図。

【図3】本発明に係る最適化学習方法概念図。

【図4】本発明に係る最適化学習方法概念図。

【図5】本発明に係る最適化学習方法概念図。

【図6】本発明が適用される連想メモリ参照データと入力データの対応関係を示す図。

【図7】本発明に係る参照データ及び認識領域の最適化学習アルゴリズムを示すフローチャート。

【図8】図7の閾値最適化学習アルゴリズムを説明するための認識領域の縮小及び拡大の様子を示す概念図。

【図9】本発明に係る閾値の更新量の選択される確率分布を示す特性図。

【図10】本発明に係る学習機能を付加した連想メモリのアーキテクチャを示すブロック図。

【図11】図10に示す学習回路の具体的な構成を示すブロック図。

【図12】図11に示す確率分布格納メモリが $Dg_i = 3$ の場合に発生させる乱数選択の一例を示す概念図。

【図13】本発明に係る参照データ最適学習に対するシミュレーション結果を示すグラフ。

【図14】本発明に係る閾値の最適化学習（認識領域最適化）に対するシミュレーション結果を示すグラフ。

【図15】図13、図14のシミュレーションに用いた入力データのばらつきの様子を示す波形図。

【図16】本発明の最適化学習方法を採用するプロセッサベースパターン認識システムの構成を示すブロック図。

【符号の説明】

【0093】

100...メモリ領域、200...WLA（Winner Line-Up Amplifier）回路、300...WTA（Winner-Take All-Circuit）回路、110...行デコーダ及びR/W回路（Column Decoder and Read/Write Circuit（ $K \times W$ Column））、120...列デコーダ（Row Decoder（ R Rows））、130...検索データ保存回路（Search Data（ K bit \times W））、

20

30

40

50

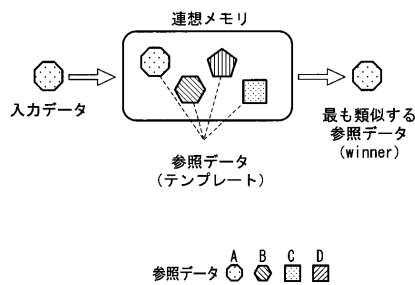
4 0 0 ... 連想メモリ装置、4 1 0 ... 連想メモリ、4 2 0 ... 入力バッファ、4 3 0 ... エンコーダ (encoder)、5 0 0 ... 制御回路、6 0 0 ... 学習回路、7 0 0 ... 比較回路、7 1 0 ... 閾値比較用比較器、7 2 0 ... 更新制御用比較器、8 0 0 ... データ保存回路、8 1 0 ~ 8 4 0 ... 第 1 乃至第 4 のメモリ、9 0 0 ... 保存先選択回路、9 1 0 , 9 2 0 ... カウンタ (counter)、

6 0 1 ... 減算器 (Subtractor)、6 0 2 ... 加算器 (Adder)、6 0 3 ... 除算器 (Divider)、6 0 4 ... 乱数発生器、6 0 5 ... 確率分布格納メモリ、6 1 1 ~ 6 1 5 ... レジスタ (REG)、6 2 1 ... 入力選択スイッチ、6 2 2 ... 出力選択スイッチ、

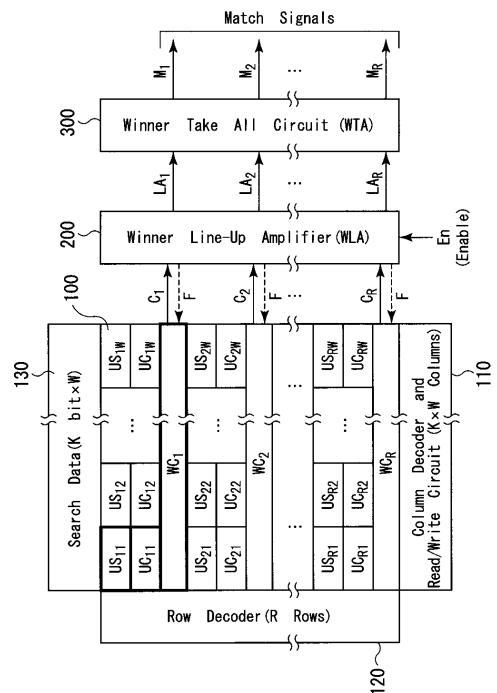
1 0 0 0 ... CPU、1 0 1 0 ... ALU、1 0 1 1 ~ 1 0 1 3 ... レジスタ、1 0 2 0 ... コントローラ、1 1 0 0 ... メモリ、1 1 1 0 ... 入力データ格納領域、1 1 2 0 ... 参照データ格納領域、1 1 3 0 ... 閾値データ格納領域、1 1 4 0 ... 距離計算結果格納領域、1 1 5 0 ... Winner 距離格納領域、1 1 6 0 ... 差分データ格納領域、1 1 7 0 ... 距離差データ格納領域、1 1 8 0 ... 第 1 カウンタ格納領域、1 1 9 0 ... 第 2 カウンタ格納領域、1 2 0 0 ... 閾値更新量データ格納領域、1 3 0 0 ... データバス。

10

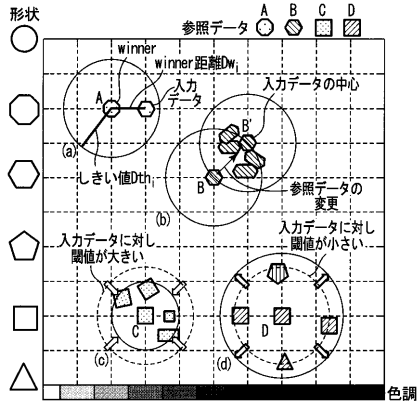
【 図 1 】



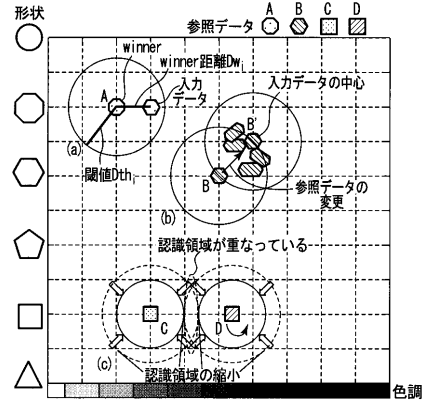
【 図 2 】



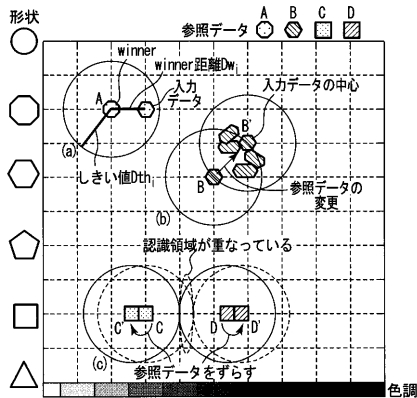
【 図 3 】



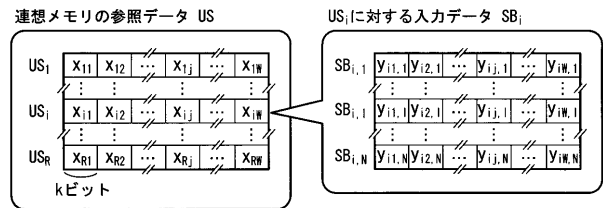
【 図 5 】



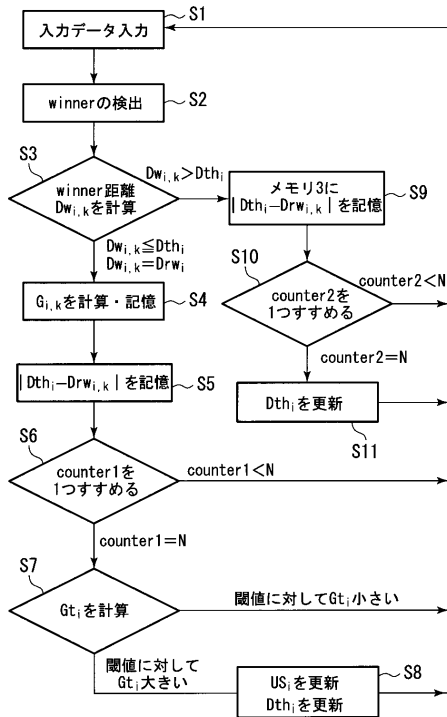
【 図 4 】



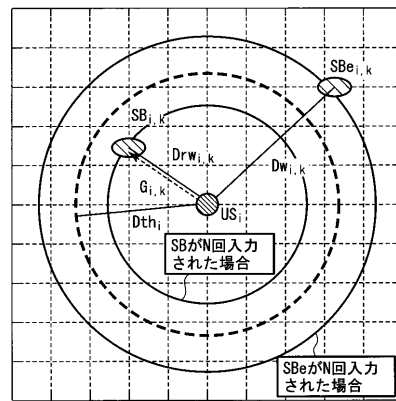
【 図 6 】



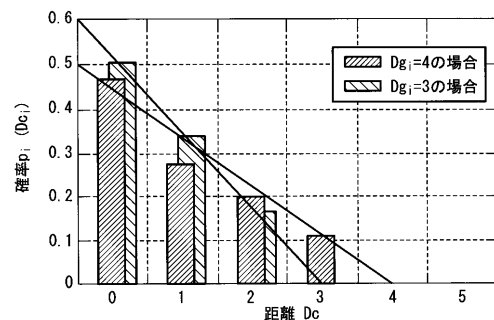
【 図 7 】



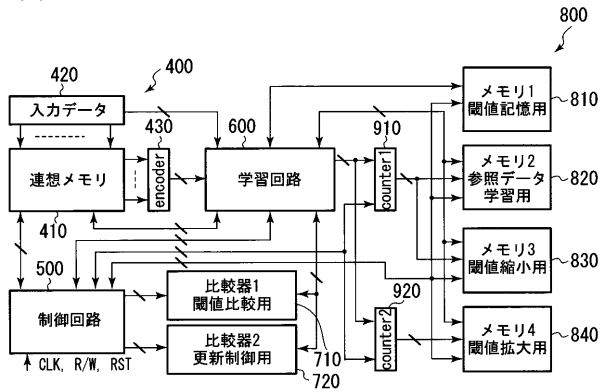
【 図 8 】



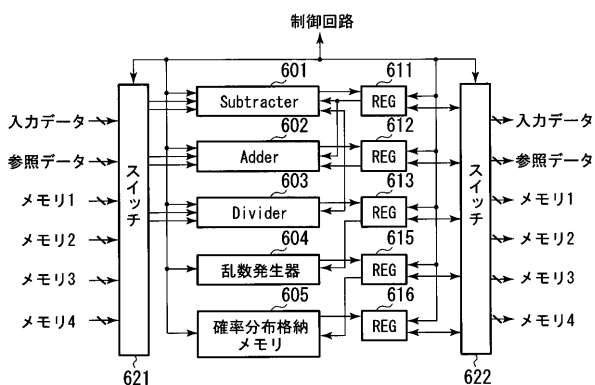
【 図 9 】



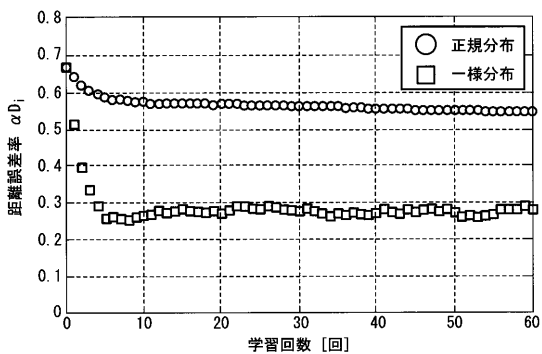
【図10】



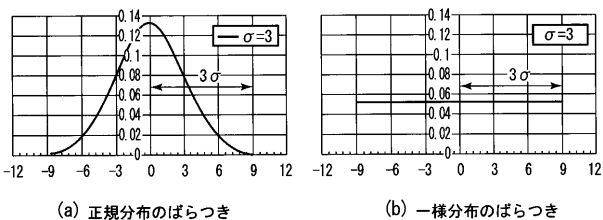
【図11】



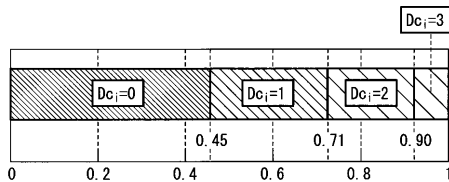
【図14】



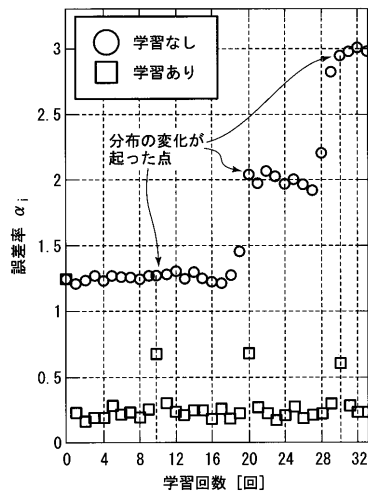
【図15】



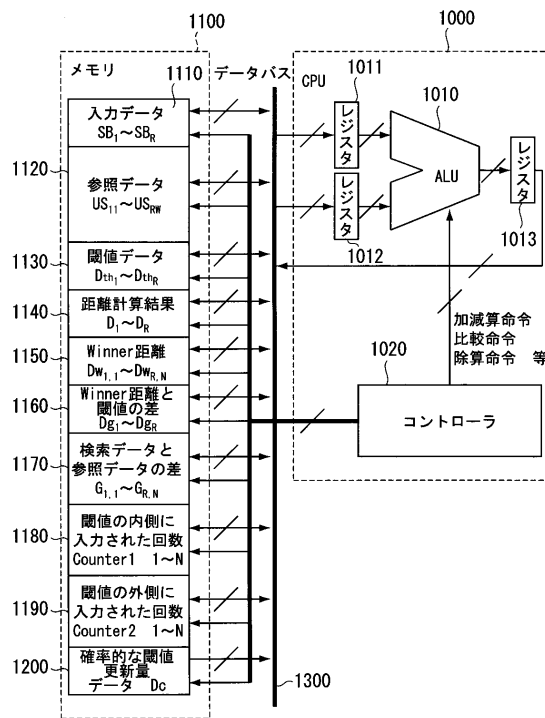
【図12】



【図13】



【図16】



フロントページの続き

(74)代理人 100084618

弁理士 村松 貞男

(74)代理人 100092196

弁理士 橋本 良郎

(72)発明者 マタウシュ・ハンスユルゲン

広島県東広島市西条町大沢 8 4 9 - 2

(72)発明者 小出 哲士

広島県東広島市西条町大字御園字 6 4 7 - 1 5 9

(72)発明者 白川 佳則

広島県広島市南区大洲 5 丁目 2 番 2 6 号 大洲 c o - o p マンション 5 0 5 号

Fターム(参考) 5B056 BB23

5L096 FA39 FA60 FA66 GA51 JA03 JA09 KA04 KA17 LA01 LA05