

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5034041号
(P5034041)

(45) 発行日 平成24年9月26日(2012.9.26)

(24) 登録日 平成24年7月13日(2012.7.13)

(51) Int.Cl.

F I

G O 6 N 3 / 0 6 3 (2006.01)

G O 6 N 3 / 0 6 3

請求項の数 5 (全 11 頁)

<p>(21) 出願番号 特願2006-198797 (P2006-198797) (22) 出願日 平成18年7月20日(2006.7.20) (65) 公開番号 特開2008-27176 (P2008-27176A) (43) 公開日 平成20年2月7日(2008.2.7) 審査請求日 平成21年4月6日(2009.4.6)</p> <p>特許法第30条第1項適用 研究集会名 九州工業大学 大学院生命体工学研究科 脳情報専攻 博士学位論文 公聴会 主催者 国立大学法人九州工業大学 開催日 平成18年1月24日</p>	<p>(73) 特許権者 504174135 国立大学法人九州工業大学 福岡県北九州市戸畑区仙水町1番1号 (74) 代理人 110000154 特許業務法人はるか国際特許事務所 (72) 発明者 山川 烈 福岡県北九州市若松区ひびきの2-4 国 立大学法人九州工業大学内 (72) 発明者 田向 権 福岡県北九州市若松区ひびきの2-4 国 立大学法人九州工業大学内</p> <p>審査官 長谷川 篤男</p>
--	---

最終頁に続く

(54) 【発明の名称】 データ生成回路及びデータ生成方法

(57) 【特許請求の範囲】

【請求項1】

入力データ及び出力データのペアを複数記憶する記憶手段と、

前記記憶手段に記憶される前記複数の入力データ及び出力データのペアのうち少なくとも一部について、それらペアに係る入力データと所与の入力データとの各距離を所定回数だけ右にビットシフトさせて得られる回数だけ、それらペアに係る出力データをそれぞれ右にビットシフトさせる第1のビットシフト手段と、

前記少なくとも一部の入力データ及び出力データのペアについて、それらペアに係る入力データと前記所与の入力データとの各距離を所定回数だけ右にビットシフトさせて得られる回数だけ、1を右にビットシフトさせる第2のビットシフト手段と、

前記第1のビットシフト手段からの出力データの加算値を前記第2のビットシフト手段からの出力データの加算値で除算することにより、前記所与の入力データに応じた出力データを生成する重み付平均演算手段と、

を含むことを特徴とするデータ生成回路。

【請求項2】

請求項1に記載のデータ生成回路において、

前記記憶手段に記憶される前記複数の入力データ及び出力データのペアのそれぞれについて、該ペアに係る入力データと前記所与の入力データとの距離を算出するとともに、算出される距離の上位所定数ビットがすべて零であるか否かを判定する判定手段と、

前記判定手段の判定結果に基づいて、前記少なくとも一部の入力データ及び出力データ

10

20

のペアを選出するセレクト手段と、
をさらに含むことを特徴とするデータ生成回路。

【請求項 3】

請求項 1 又は 2 に記載のデータ生成回路において、
学習用の入力データ及び出力データのペア及び該ペアに対する評価値に基づいて、前記記憶手段に記憶される前記複数の入力データ及び出力データのペアを生成する学習手段をさらに含む、

ことを特徴とするデータ生成回路。

【請求項 4】

請求項 1 乃至 3 のいずれかに記載のデータ生成回路において、
前記第 1 のビットシフト手段は、前記記憶手段に記憶される前記入力データ及び出力データのペアと同数のビットシフト回路を含み、各ビットシフト回路は、該ビットシフト回路に対応する前記入力データ及び出力データのペアについて、該ペアに係る入力データと前記所与のデータとの距離に応じた回数だけ、該ペアに係る出力データを右にビットシフトさせる、

ことを特徴とするデータ生成回路。

【請求項 5】

記憶手段に記憶される複数の入力データ及び出力データのペアのうち少なくとも一部について、それらペアに係る入力データと所与の入力データとの各距離を所定回数だけ右にビットシフトさせて得られる回数だけ、それらペアに係る出力データをそれぞれ右にビットシフトさせるステップと、

前記少なくとも一部の入力データ及び出力データのペアについて、それらペアに係る入力データと前記所与の入力データとの各距離を所定回数だけ右にビットシフトさせて得られる回数だけ、1 を右にビットシフトさせるステップと、

ビットシフト済みの出力データの加算値をビットシフト済みの 1 の加算値で除算することにより、前記所与の入力データに応じた出力データを生成するステップと、

を含むことを特徴とするデータ生成方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデータ生成回路及びデータ生成方法に関し、特に、入力データ及び出力データのペアとその評価値を用いた学習により抽出されたルールのうち、入力データに適合する 1 又は複数のルールを用いて、該入力データに応じた出力データを生成するデータ生成回路及びデータ生成方法に関する。

【背景技術】

【0002】

データを用いて学習を行う装置及び入力データから出力データを生成する装置は数多く提案されているが、中でも、下記非特許文献 1 及び特許文献 1 に開示された自己組織化関係ネットワークは、入力データ及び出力データのペアとその評価値から制御対象を安定に制御するルール（入出力関係）を学習により抽出するとともに、抽出したルールのうち、入力データに適合する 1 又は複数を選出して、それらを用いて入力データに応じた出力データを生成するものであって、好ましい（評価の高い）入出力関係を学習により容易に獲得でき、活用できることから非常に優位性が高い。

【非特許文献 1】山川烈・堀尾恵一，「自己組織化関係ネットワーク」，電子情報通信学会論文誌，社団法人電子情報通信学会，1999年8月1日，vol. E82-A，No. 8，pp. 1674-1678

【特許文献 1】特開 2000-122991 号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

10

20

30

40

50

しかしながら、上記自己組織化関係ネットワークは、プログラムにより実現されるものであって、実現にはパーソナルコンピュータを要するので装置の規模が大きいという問題がある。さらに、自己組織化関係ネットワークは学習やデータ生成に膨大な時間を要するという問題がある。特に、入力データに適合する1又は複数のルールを選出する際に、入力データと各ルール（入力データとそれに応じた好ましい出力データのペア）との類似度を演算することになるが、この類似度を演算するためには、通常ルックアップテーブルが必要であり、その実装面積が膨大であるという問題がある。また、入力データから出力データを生成する際に用いる重み付き平均は、類似度と各ルールに対応する出力データとの乗算を伴うため、その演算量が膨大となるという問題がある。

【0004】

10

本発明は上記課題に鑑みてなされたものであって、その目的は、自己組織化関係ネットワークをコンパクト且つ高速に実現するためのデータ生成回路及びデータ生成方法を提供することにある。

【課題を解決するための手段】

【0005】

上記課題を解決するために、本発明に係るデータ生成回路は、入力データ及び出力データのペアを複数記憶する記憶手段と、前記記憶手段に記憶される前記複数の入力データ及び出力データのペアのうち少なくとも一部について、それらペアに係る入力データと所与の入力データとの各距離に応じた回数だけ、それらペアに係る出力データをそれぞれ右にビットシフトさせる第1のビットシフト手段と、前記第1のビットシフト手段からの出力データの加算値に基づいて、前記所与の入力データに応じた出力データを生成する重み付平均演算手段と、を含むことを特徴とする。

20

【0006】

本発明では、第1のビットシフト手段により、記憶手段に記憶される入力データと前記所与の入力データとの距離が大きいほど、該入力データに対応する出力データが小さくなる。すなわち、記憶手段に記憶される入力データと前記所与の入力データとの距離に応じた回数を s とすると、出力データ u は、 $u \times 2^{-s}$ となる。そして、こうしてビットシフトされた出力データを加算することにより、記憶手段に記憶される出力データの重み付平均が得られ、これが前記所与の入力データに応じた出力データとして利用される。本発明によると、ビットシフトという演算量の少ないデジタルデータ処理により、記憶手段により記憶される出力データの重み付平均を得ることができ、自己組織化関係ネットワークをコンパクト且つ高速に実現することができるようになる。

30

【0007】

本発明の一態様では、前記少なくとも一部の入力データ及び出力データのペアについて、それらペアに係る入力データと前記所与の入力データとの各距離に応じた回数だけ、1を右にビットシフトさせる第2のビットシフト手段をさらに含み、前記重み付き平均演算手段は、前記第1のビットシフト手段からの出力データの加算値及び前記第2のビットシフト手段からの出力データの加算値に基づいて、前記所与の入力データに応じた出力データを生成する。この態様によれば、第2のビットシフト手段からの出力値の加算値で、第1のビットシフト手段からの出力値の加算値を除算することにより、重み係数を正規化することができる。

40

【0008】

また、本発明の一態様では、前記記憶手段に記憶される前記複数の入力データ及び出力データのペアのそれぞれについて、該ペアに係る入力データと前記所与の入力データとの距離を算出するとともに、算出される距離の上位所定数ビットがすべて零であるか否かを判定する判定手段と、前記判定手段の判定結果に基づいて、前記少なくとも一部の入力データ及び出力データのペアを選出するセレクタ手段と、をさらに含む。算出される距離の上位所定数ビットがすべて零であるか否かは、記憶手段に記憶される各入力データが前記所与の入力データから所定距離内にあるか否かに対応しており、例えばNOR（否定論理和演算）回路により容易に判定することができる。この態様によると、前記所与の入力デ

50

ータから所定距離内にある入力データに係るペアだけを重み付平均演算の対象とすることができ、処理をさらに簡略化することができる。

【0009】

また、本発明の一態様では、学習用の入力データ及び出力データのペア及び該ペアに対する評価値に基づいて、前記記憶手段に記憶される前記複数の入力データ及び出力データのペアを生成する学習手段をさらに含む。こうすれば、学習手段により、好ましい入力データ及び出力データのペアを記憶手段に記憶させることができる。

【0010】

また、本発明の一態様では、前記第1のビットシフト手段は、前記記憶手段に記憶される前記入力データ及び出力データのペアと同数のビットシフト回路を含み、各ビットシフト回路は、該ビットシフト回路に対応する前記入力データ及び出力データのペアについて、該ペアに係る入力データと前記所与のデータとの距離に応じた回数だけ、該ペアに係る出力データを右にビットシフトさせる。こうすれば、各ペアに関するビットシフト演算を並列して実行することが可能となり、自己組織化関係ネットワークをさらに高速に実現できる。

【0011】

また、本発明に係るデータ生成方法は、記憶手段に記憶される複数の入力データ及び出力データのペアのうち少なくとも一部について、それらペアに係る入力データと所与の入力データとの各距離に応じた回数だけ、それらペアに係る出力データをそれぞれ右にビットシフトさせるステップと、ビットシフト済みの出力データの加算値に基づいて、前記所与の入力データに応じた出力データを生成するステップと、を含むことを特徴とする。

【0012】

本発明によると、ビットシフトという演算量の少ないデジタルデータ処理により、記憶手段により記憶される出力データの重み付平均を得ることができ、自己組織化関係ネットワークをコンパクト且つ高速に実現することができるようになる。

【発明を実施するための最良の形態】

【0013】

以下、本発明の一実施形態について図面に基づき詳細に説明する。

【0014】

図1は、本発明の一実施形態に係る自己組織化関係ネットワークを示す図である。同図に示すように、自己組織化関係ネットワークは、それぞれn個、m個、N個のユニットが配置された入力層、出力層及び競合層の3層からなるネットワークである。図1では競合層ユニットは競合層上に1次元に配置されているが、2次元の正方格子状、六角格子状、3次元の球状などに配置されてもよい。入力層には入力ベクトル(入力データ)x、出力層には出力ベクトル(出力データ)yが関連付けられている。入力層、出力層と競合層は結合重みベクトル $v_j = (w_j, u_j)$ で全結合されている。この結合重みベクトルは、好ましい入力ベクトル及び出力ベクトルのペアを示しており、 w_j (以下、「入力部結合重みベクトル」という。)が入力ベクトル、 u_j (以下、「出力部結合重みベクトル」という。)が出力ベクトルに対応している。

【0015】

図2は、上記自己組織化関係ネットワークに関連付けられた学習機能付データ生成回路10のデジタルハードウェアアーキテクチャを示す図である。同図に示すように、自己組織化関係ネットワークのデジタルハードウェアアーキテクチャは、競合層の各ユニットに対応する複数のローカル回路11と、勝者決定回路12、重み付き平均演算回路13、コントローラ14を含んで構成されている。これらは自己組織化関係ネットワークのデジタルハードウェアアーキテクチャを記述したプログラムを公知のコンパイラを用いて論理回路の組み合わせに変換し、これを公知のField Programmable Gate Array (FPGA) や Application Specific Integrated Circuit (ASIC)、論理回路ICなどを用いて実装することで実現される。学習機能付データ生成回路10は、学習モード又は実行モードで動作するものである。

10

20

30

40

50

【 0 0 1 6 】

図 3 は、ローカル回路 1 1 のデジタルハードウェアアーキテクチャを示す図である。同図に示すようにローカル回路 1 1 は、メモリ 2 1、距離演算回路 2 2、メンバシップ関数生成回路 2 3、ビットシフト回路 2 4、2 6、セレクト回路 2 5 を含んで構成されている。メモリ 2 1 は、結合重みベクトル v_j を保存する。また、距離演算回路 2 2 は、学習モードにおいて入力ベクトル及び出力ベクトルのペアと結合重みベクトル v_j との距離を演算する。また、実行モードにおいて入力ベクトルと入力部結合重みベクトル w_j との距離を演算する。メンバシップ関数生成回路 2 3 は、距離演算回路 2 2 から出力される距離信号 d に対して右ビットシフト操作を施して、その結果であるビットシフト回数 s を出力する。さらに、ビットシフト回数 s の上位所定数ビットに対する NOR (否定論理和演算) の結果であるアクティブフラグ ($f l a g$) を出力する。

10

【 0 0 1 7 】

ビットシフト回路 2 6 は、固定のデジタル値 “ 1 ” に対してビットシフト回数 s だけ右ビットシフト操作を施し、これにより入力ベクトルとメモリ 2 1 に記憶された結合重みベクトル v_j の成分 u_j に関する重み係数 z を生成する。また、ビットシフト回路 2 4 は、メモリ 2 1 から読み出される出力部結合重みベクトル u_j に対してビットシフト回数 s だけ右ビットシフト操作を施し、これにより入力ベクトルとメモリ 2 1 に記憶された入力部結合重みベクトル w_j に関するファジィ類似度 (重み係数) z に、メモリ 2 1 に記憶された出力部結合重みベクトル u_j に乗じた値 $z u_j$ を生成する。セレクト回路 2 5 は、メンバシップ関数生成回路 2 3 から出力されるアクティブフラグが 1 の場合、すなわち入力ベクトルとメモリ 2 1 に記憶された入力部結合重みベクトル w_j との距離 d が所定距離内である場合は、ビットシフト回路 2 6 及びビットシフト回路 2 4 から出力される z 及び $z u$ を重み付平均演算回路 1 3 に供給する。また、アクティブフラグが 0 の場合は、 z 及び $z u$ を重み付平均演算回路 1 3 に供給せず、他のローカル回路 1 1 にデータ出力順を明け渡す。

20

【 0 0 1 8 】

なお、本実施形態では、結合重みベクトル v_j を保存するためのメモリ 2 1 を各ローカル回路 1 1 内に分散的に配置しているが、ローカル回路 1 1 の外部に集中配置する形態や、各ローカル回路 1 1 の外部に個別に配置する形態をとってもよい。

【 0 0 1 9 】

まず、学習機能付データ生成回路 1 0 の学習モードにおける動作について説明する。図 4 は、学習モードにおける動作フロー図である。同図に示すように、学習モードでは、まず、ローカル回路 1 1 のメモリ 2 1 に格納された結合重みベクトルを初期化 ($S 1 0 0$) する。次に、入力信号が回路外部から入力される。学習モードの入力信号は、入力ベクトル及び出力ベクトルのペア及びその評価値で、これを学習ベクトルとする。入力信号は全てのローカル回路 1 1 に同時に入力され、これにより学習ベクトルの提示となる ($S 1 0 1$)。学習ベクトルが提示されると、各ローカル回路 1 1 は学習ベクトルとメモリ 2 1 に格納された結合重みベクトル v_j 間の距離を、距離演算回路 2 2 を用いて演算する。用いる尺度は距離の性質を満たしていれば、例えばマンハッタン距離やマハラノビス距離など、どのようなものでもよい。

30

40

【 0 0 2 0 】

次に、次式 (1) に基づき、演算された距離の中から最も小さな値を持つローカル回路を、勝者決定回路 1 2 を用いて勝者ユニットとして決定する ($S 1 0 2$)。ここで、 i はローカル回路を順序付ける番号、 c は勝者ユニットの番号、 I は学習ベクトル、 v_i は i 番目のローカル回路 1 1 に関連付けられた結合重みベクトル、 t は現在時刻を表すものである。勝者ユニットの番号は制御信号としてコントローラ 1 4 に伝えられ、コントローラ 1 4 は勝者ユニットの番号及び競合層ユニットの配置の定義に基づき、近傍ユニット (競合層において勝者ユニットの近傍に配置されたユニット) を決定する。

【 0 0 2 1 】

【数 1】

$$c = \arg \min_i \|I - v_i(t)\| \quad \dots (1)$$

【0022】

その後、勝者ユニット及び近傍ユニットに関連付けられたローカル回路 11 のメモリ 21 を次式 (2) に従って更新する (S103)。ここで、 α は正の評価値 (肯定的評価) の場合における学習係数、 β は負の評価値 (否定的評価) の場合における学習係数、 E は評価値である。このとき、学習係数 α 、 β 及び評価値 E を 2 のべき乗で記述すれば、式 (2) における乗算は全てビットシフトで演算可能であり、ハードウェアをコンパクト化することができ、また演算時間を削減できる。もちろん、これらを 2 のべき乗に限定せずに乗算器を用いて式 (2) を実現してもよい。また、なお、負の評価値に対して、更新式による結合重みベクトル v_i の更新を行った際に、更新後の値があらかじめ定義しておいた値域を超える場合、オーバーフローを防ぐために値域の最大もしくは最小値を取るよう学習量を制限する。

10

【0023】

【数 2】

$$v_i(t+1) = \begin{cases} v_i(t) + \alpha \cdot E(I - v_i(t)) & \text{for } E \geq 0 \\ v_i(t) - \beta \cdot E \cdot 2^{\|I - v_i(t)\|} (I - v_i(t)) & \text{for } E < 0 \end{cases} \quad \dots (2)$$

20

【0024】

学習ベクトルが提示される間、もしくは規定回数が終了するまで同様の手順を繰り返す (S101 ~ S103)。以上の操作によって、学習回路及びデータ生成回路 10 は、学習用の入力データ 出力データ対とその評価値からルール (好ましい入力データ - 出力データ対) を抽出し、これを各ローカル回路 11 のメモリ 21 に獲得する。

【0025】

こうしてメモリ 21 に格納されたルールは、必要に応じて回路外部に読み出すことができる。図 5 は、ルール読み出し処理を示すフロー図である。同図に示すように、外部からローカル回路 11 のアドレスを指定すると (S200)、wired-or もしくは OR 論理回路で結合されている、図 2 に示される結合重みベクトル信号の信号線を通り、指定されたローカル回路 11 の結合重みベクトルの内容を出力する (S201)。以上の操作によって、学習機能付データ生成回路 10 は、学習モードで抽出されたルールを回路外部へ取り出す。

30

【0026】

次に、学習機能付データ生成回路 10 の実行モードにおける動作を説明する。図 6 は、実行モードにおける動作フロー図である。同図に示すように、実行モードでは、実行用の入力データを入力ベクトル x として提示すると (S300)、入力ベクトル x と、全てのローカル回路 11 のメモリ 21 に格納されている入力部結合重みベクトル w_i との間の距離 d_i を、距離演算回路 22 にて、次式 (3) を用いて演算する (S301)。

40

【0027】

【数 3】

$$d_i = \|x - w_i\| \quad \dots (3)$$

【0028】

距離 d_i を示す距離信号はメンバシップ関数生成回路 23 へと入力され、次式 (4) を用いてファジィ類似度 z が算出される (S302)。

【0029】

50

【数4】

$$z_i = \begin{cases} 2^{-s_i}, & \text{if } flag_i = 1 \\ 0, & \text{otherwise} \end{cases} \quad \dots (4)$$

【0030】

図7は、メンバシップ関数生成回路23のデジタルハードウェアアーキテクチャを示す図である。同図に示すように、メンバシップ関数生成回路23は、メモリ31、ビットシフト回路32、NORゲート33を含んで構成される。

10

【0031】

s_i は i 番目のローカル回路11で算出される上述したビットシフト回数であり、次式(5)で与えられる。 r はメンバシップ関数の幅を表すパラメタ、 a は演算精度(ビット)を表すパラメタ、 $r - \log_2 a$ はビットシフト回数(符号34)である。幅の広いメンバシップ関数が必要な場合は大きな r を、幅の狭いメンバシップ関数が必要な場合、小さな r を設定することで、様々なメンバシップ関数を生成することが出来る。例えば $r = 4$ 、 $a = 8$ の場合、メンバシップ関数の幅は16となり、 s_i は d_i を右に1ビットシフトすることで算出することが出来る。ビットシフト回路32は、距離演算回路22で得られた距離信号に対して、メモリ31に格納されたビットシフト回数($r - \log_2 a$)だけ、右ビットシフト操作を行うことでビットシフト回数 s_i を算出する。

20

【0032】

【数5】

$$s_i = \frac{d_i}{2^{r - \log_2 a}} \quad \dots (5)$$

【0033】

図8は、式(4)及び式(5)で生成されるメンバシップ関数の形状の一例($r = 3$ 、 $a = 8$ の場合)である。メンバシップ関数の出力値であるファジィ類似度 z が2のべき乗である点が特徴的であり、これによりファジィ類似度 z を出力部結合重みベクトル u_j に乗算する演算を、出力ベクトル u_j をビットシフト回数 s_i だけ右ビットシフト操作する演算により代替することが可能となる。この結果、ハードウェアをコンパクトにし、かつ演算を高速化することが可能となる。

30

【0034】

式(4)の $flag$ は、アクティブユニットかどうかを示すものである。 $flag = 1$ の場合はアクティブユニットとする。また、0の場合はアクティブユニットではないと判断され、その類似度を0とする。 $flag$ の値はメンバシップ関数生成回路23においてビットシフト回路32から出力されるビットシフト回数 s の上位所定数のビットをNORゲート33に入力することで得られる。具体的には、距離 d_i を格納するためのレジスタのビット数を D とすると、 s_i の上位 $D - \log_2 a$ ビットをNORゲート33に入力する。アクティブユニットの場合、NORゲート33の入力35は全て0になるので、出力として1が得られる。NORゲート33の入力35に1が1つでも含まれる場合、出力として0が得られる。これは、メンバシップ関数の幅の外(図7の場合は $s > 8$)に入力があるということの意味し、非アクティブなユニットであると判断される。本実施形態では、以上のように、NORゲート33を用いてアクティブユニットの判定を行う点が特徴的である。なお、 s_i の下位 $\log_2 a$ ビットはメンバシップ関数生成回路30から出力される。

40

【0035】

図3に示すように、ファジィ類似度 z と結合重みベクトルの出力部 u との積 zu は、ビットシフト回路24により演算される。式(4)で、 z が2の s_i 乗で表現されている

50

ため、 $z u$ は、 u をビットシフト回数 s_i だけ右ビットシフト操作することで得ることが出来る。したがって、回路規模の大きい乗算器の代わりにビットシフト回路を用いることが出来るため、コンパクトなハードウェア化が可能となる。また、メンバシップ関数生成回路23から出力されるビットシフト回数 s_i はビットシフト回路26にも入力され、ビットシフト回路26では、デジタル値1を右に s_i ビットシフトすることでファジィ類似度 z を演算する。

【0036】

コントローラ14は、1番目のローカル回路11から順に、 z 及び $z u$ の出力命令を送る。命令を受けたローカル回路11は、セクタ回路25にて $flag$ の有無を確認し、 $flag$ がある場合は z 及び $z u$ を出力し、出力したという信号をコントローラ14へ返す。 $flag$ が無い場合、出力の権利を次のローカル回路11へ移す。出力の権利を受け取ったローカル回路11は、出力の命令を受け取ったローカル回路11と同様の動作を行う。重み付き平均回路13へ向かう z 及び $z u$ の信号線は $wired-or$ もしくはORゲートにて接続されている。これにより、出力を行ったローカル回路11の z 及び $z u$ のみが重み付き平均演算回路13へ送られる。

10

【0037】

図9は、重み付き平均演算回路13のデジタルハードウェアアーキテクチャを示す図である。同図のように、重み付き平均演算回路13は、加算器41、加算器42、除算器43を含んで構成される。重み付き平均演算回路13を用いて、次式(6)により出力信号を生成する(S303)。すなわち、加算器42により各ローカル回路11から出力されるファジィ類似度 z (重み係数として用いられる。)の合計値 z が計算され、加算器41により各ローカル回路11から出力される出力結合重みベクトルに重み係数を乗じた値 $z u$ の合計値 $z u$ が計算される。そして、除算器43では、 $z u$ を z で割った値が算出され、これが出力される。

20

【0038】

【数6】

$$y = \frac{\sum_{i \in C} z_i u_i}{\sum_{i \in C} z_i} \quad \dots (6)$$

30

【0039】

式(6)は、アクティブユニットのみを用いて重み付き平均演算を行うことを意味する。

すなわち、同式(6)において、 C はアクティブユニットの集合を示しており、非アクティブユニットをスキップし、アクティブユニットのみで、重み付き平均演算を行う点が特徴的である。通常、競合層ユニットの数に比べ、アクティブユニットの数は非常に少ない。したがって、従来の自己組織化関係ネットワークでは N 回の積和演算が必要であったのに対し、本実施形態により C 回の積和演算で重み付き平均が実現できるので、大幅な演算量削減が可能となる。

40

【0040】

以上の操作によって、学習機能付データ生成回路10は、入力信号から出力信号を生成する。

【0041】

以上の学習機能付データ生成回路10によれば、自己組織化マップに関する学習及びデータ生成を高速かつコンパクトに実現可能であり、例えば、ロボットビジョンの処理などのような高速性とデバイスの小型化が要求される応用の実現が可能となる。

【図面の簡単な説明】

【0042】

50

【図1】本発明の実施形態に係る学習機能付データ生成回路によりシミュレートされる自己組織化関係ネットワークを示す概念図である。

【図2】本発明の実施形態に係る学習機能付データ生成回路のデジタルハードウェアアーキテクチャを示す図である。

【図3】ローカル回路のデジタルハードウェアアーキテクチャを示す図である。

【図4】本発明の実施形態に係る学習機能付データ生成回路の学習モードにおける動作フロー図である。

【図5】本発明の実施形態に係る学習機能付データ生成回路のルール取り出し処理を示すフロー図である。

【図6】本発明の実施形態に係る学習機能付データ生成回路の実行モードにおける動作フロー図である。

【図7】メンバシップ関数生成回路のデジタルハードウェアアーキテクチャを示す図である。

【図8】メンバシップ関数生成回路にて得られるメンバシップ関数の形状を示す図である。

。

【図9】重み付き平均演算回路のデジタルハードウェアアーキテクチャを示す図である。

【符号の説明】

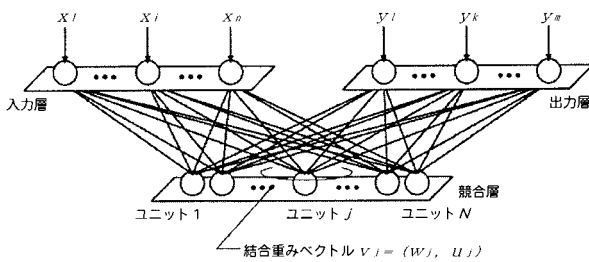
【0043】

10 学習機能付データ生成回路、11 ローカル回路、12 勝者決定回路、13 重み付き平均演算回路、14 コントローラ、21 メモリ、22 距離演算回路、23 メンバシップ関数生成回路、24, 26 ビットシフト回路、25 セレクタ回路、31 メモリ、32 ビットシフト回路、33 NORゲート、41, 42 加算器、43 除算器。

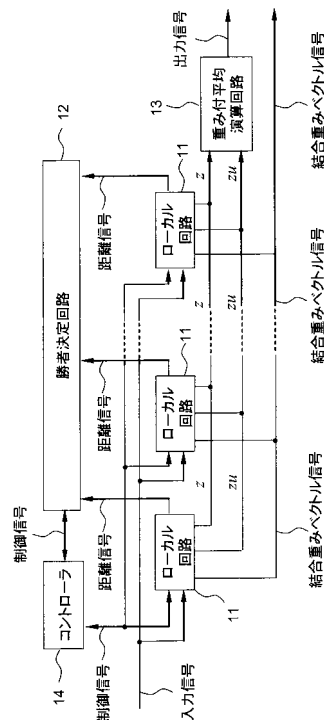
10

20

【図1】

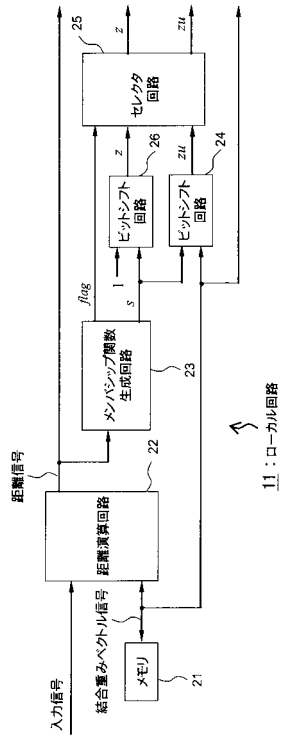


【図2】

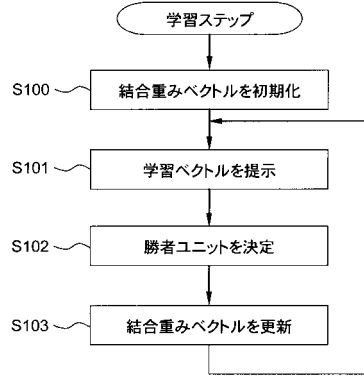


10: 学習回路及びデータ生成回路

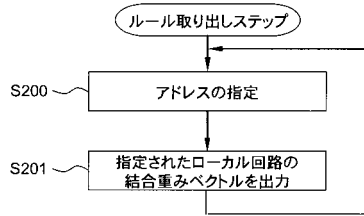
【図3】



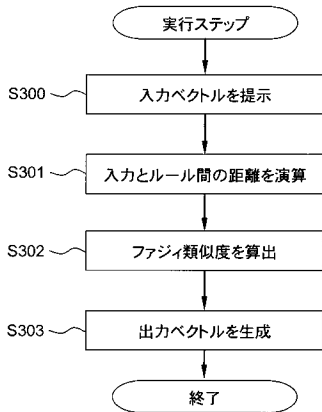
【図4】



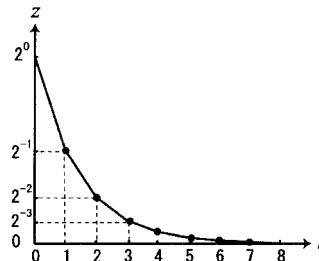
【図5】



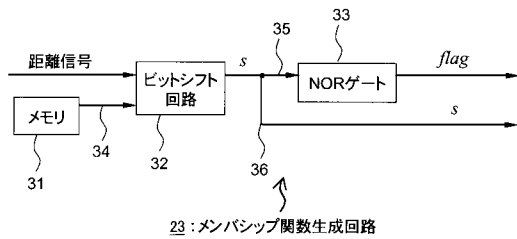
【図6】



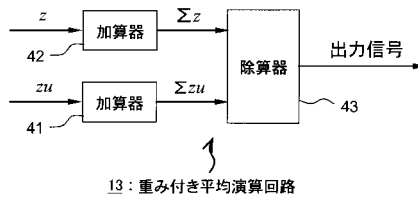
【図8】



【図7】



【図9】



フロントページの続き

(56)参考文献 特開2000-122991(JP,A)
特開平04-051384(JP,A)
特開2004-287782(JP,A)
特開平06-259585(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06N 3/00-3/08
IEEE Xplore
JSTPlus(JDreamII)