

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6393058号
(P6393058)

(45) 発行日 平成30年9月19日(2018.9.19)

(24) 登録日 平成30年8月31日(2018.8.31)

(51) Int.Cl.

F I

G 0 6 N 99/00 (2010.01)

G 0 6 N 99/00 1 5 0

請求項の数 9 (全 19 頁)

(21) 出願番号	特願2014-73728 (P2014-73728)	(73) 特許権者	000001007
(22) 出願日	平成26年3月31日(2014.3.31)		キヤノン株式会社
(65) 公開番号	特開2015-197702 (P2015-197702A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成27年11月9日(2015.11.9)	(74) 代理人	100076428
審査請求日	平成29年3月31日(2017.3.31)		弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 情報処理装置、情報処理方法

(57) 【特許請求の範囲】

【請求項 1】

複数の乱数シードの候補のそれぞれについて、該候補を用いて乱数列を生成し、該乱数列から基底ベクトルを生成し、該基底ベクトルと、複数の学習画像のそれぞれから予め抽出された特徴ベクトルと、の内積演算の結果から、該候補に対する評価値を求める評価手段と、

前記複数の乱数シードの候補のうち前記評価値が高い順に上位から、変換元ベクトルを変換する変換先ベクトルの次元数に対応する個数の候補をメモリに登録する登録手段と、

前記変換元ベクトルを取得する取得手段と、

前記メモリに格納されている複数の乱数シードを用いて複数の乱数列を生成する生成手段と、

前記複数の乱数列から複数の基底ベクトルを生成し、該生成した複数の基底ベクトルを用いて前記変換元ベクトルに対する乱数射影を行うことで、該変換元ベクトルをより低次元の前記変換先ベクトルに変換する変換手段と

を備えることを特徴とする情報処理装置。

【請求項 2】

前記変換手段は、前記複数の基底ベクトルのそれぞれと、前記変換元ベクトルと、の内積演算を行うことで、前記変換元ベクトルをより低次元の前記変換先ベクトルに変換することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

10

20

前記変換手段は、前記複数の乱数列を、前記複数の基底ベクトルとすることを特徴とする請求項 1 又は 2 に記載の情報処理装置。

【請求項 4】

前記変換手段は、前記複数の乱数列から平均乱数列を求め、該求めた平均乱数列から前記基底ベクトルを取り出すことを特徴とする請求項 1 又は 2 に記載の情報処理装置。

【請求項 5】

前記基底ベクトルは、コンボリューション演算で用いるフィルタ係数であることを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理装置。

【請求項 6】

前記基底ベクトルは、コンボリューショナルニューラルネットワークで用いるフィルタ係数であることを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理装置。

【請求項 7】

前記基底ベクトルは、階層型ニューラルネットワークにおける 1 つのニューロン値を算出するための前階層ニューロン値に対する結合係数列であることを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理装置。

【請求項 8】

情報処理装置が行う情報処理方法であって、

前記情報処理装置の評価手段が、複数の乱数シードの候補のそれぞれについて、該候補を用いて乱数列を生成し、該乱数列から基底ベクトルを生成し、該基底ベクトルと、複数の学習画像のそれぞれから予め抽出された特徴ベクトルと、の内積演算の結果から、該候補に対する評価値を求める評価工程と、

前記情報処理装置の登録手段が、前記複数の乱数シードの候補のうち前記評価値が高い順に上位から、変換元ベクトルを変換する変換先ベクトルの次元数に対応する個数の候補をメモリに登録する登録工程と、

前記情報処理装置の取得手段が、前記変換元ベクトルを取得する取得工程と、

前記情報処理装置の生成手段が、前記メモリに格納されている複数の乱数シードを用いて複数の乱数列を生成する生成工程と、

前記情報処理装置の変換手段が、前記複数の乱数列から複数の基底ベクトルを生成し、該生成した複数の基底ベクトルを用いて前記変換元ベクトルに対する乱数射影を行うことで、該変換元ベクトルをより低次元の前記変換先ベクトルに変換する変換工程と

を備えることを特徴とする情報処理方法。

【請求項 9】

コンピュータを、請求項 1 乃至 7 の何れか 1 項に記載の情報処理装置の各手段として機能させるためのコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、乱数を用いた射影演算を行うための技術に関する。

【背景技術】

【0002】

ランダムプロジェクション（乱数射影）に基づく次元削減・識別技術が注目されている。例えば、特許文献 1 では、乱数に基づく効率的なハッシュ関数の生成に関する技術が開示されている。また、非特許文献 1 では、ランダムプロジェクションを画像データやテキストデータの次元削減に適用した例が開示されている。また、非特許文献 2 では、乱数に基づくフィルタを利用した物体認識に関する技術が開示されている。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2013 - 68884 号公報

【非特許文献】

10

20

30

40

50

【0004】

【非特許文献1】“Random projection in dimensionality reduction: Applications to image and text data”, Bingham, KDD2001

【非特許文献2】“Beyond simple features: A large-scale feature search approach to unconstrained face recognition”, Nicolas Pinto, David Cox, Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on

10

【発明の概要】

【発明が解決しようとする課題】

【0005】

ランダムプロジェクションでは、乱数に基づいて生成した基底ベクトルを用いて特徴ベクトルをより低次元の空間に射影する。一般的に射影演算を実現する場合、特許文献1に開示されている様に、記憶装置に格納された基底ベクトル（別途予め算出された基底ベクトル）を参照しながら処理する。

【0006】

ランダムプロジェクションを利用した従来の識別装置の構成例について、図2のブロック図を用いて説明する。201は識別処理をアクセラレートするハードウェアであり、特徴抽出処理部202、内積演算処理部203、識別処理部204からなる。内積演算処理部203は、特徴抽出処理部202で生成した特徴ベクトルをより低次元の射影ベクトルに変換する（ここでは乱数を用いたランダムプロジェクションにより処理する）。内積演算処理部203はシステムバス205を介して外部メモリ206に格納されている基底ベクトルを参照しながら射影演算を実行する。外部メモリ206はDRAM（Dynamic Random Access Memory）等の低速・大容量のメモリで構成される。廉価なシステムでは、システムバス205の帯域（単位時間当たりのデータ転送量）や外部メモリ206へのアクセス遅延の制約により、外部メモリ206に格納されている基底ベクトルの転送に要する時間が処理時間を超えてしまう場合がある。また、外部メモリ206は通常図示しない他の処理と共有されるため、システムバス205に必要な帯域は狭い事が望まれる。

20

30

【0007】

基底ベクトルの転送コスト（必要帯域）を削減するために、図3に示す様に、乱数生成器302をアクセラレータハード301に内蔵する手法も考えられる。全ての基底ベクトルが一連の乱数系列からなる場合はこのような実装手法でも良いが、特許文献1や非特許文献2に記載されている様に複数の乱数列の中から所望の条件で選択した乱数列を基底ベクトルとする場合、図3に示したような手法は適用できない。

【0008】

本発明はこのような問題に鑑みてなされたものであり、複数の乱数列から選択された乱数列を用いた射影処理をより廉価且つ高速に実現するための技術を提供する。

40

【課題を解決するための手段】

【0009】

本発明の一様態は、複数の乱数シードの候補のそれぞれについて、該候補を用いて乱数列を生成し、該乱数列から基底ベクトルを生成し、該基底ベクトルと、複数の学習画像のそれぞれから予め抽出された特徴ベクトルと、の内積演算の結果から、該候補に対する評価値を求める評価手段と、

前記複数の乱数シードの候補のうち前記評価値が高い順に上位から、変換元ベクトルを変換する変換先ベクトルの次元数に対応する個数の候補をメモリに登録する登録手段と、

前記変換元ベクトルを取得する取得手段と、

前記メモリに格納されている複数の乱数シードを用いて複数の乱数列を生成する生成手

50

段と、

前記複数の乱数列から複数の基底ベクトルを生成し、該生成した複数の基底ベクトルを用いて前記変換元ベクトルに対する乱数射影を行うことで、該変換元ベクトルをより低次元の前記変換先ベクトルに変換する変換手段と

を備えることを特徴とする。

【発明の効果】

【0010】

本発明の構成によれば、複数の乱数列から選択された乱数列を用いた射影処理をより廉価且つ高速に実現することができる。

【図面の簡単な説明】

10

【0011】

【図1】情報処理装置の機能構成例を示すブロック図。

【図2】ランダムプロジェクションを利用した従来の識別装置の構成例を示す図。

【図3】乱数生成器302をアクセラレータハード301に内蔵する手法を説明する図。

【図4】式(1)に従った演算の様子を模式的に示す図。

【図5】内積演算処理部103の構成例を示すブロック図。

【図6】乱数生成処理部106の構成例を示すブロック図。

【図7】CPU109が行う処理のフローチャート。

【図8】情報処理装置の動作のタイムチャート。

【図9】画素ブロックについて説明する図。

20

【図10】乱数シードを生成して外部メモリ111に登録する処理のフローチャート。

【図11】CNNの構成例を示す図。

【図12】CNNをハードウェアを用いて実現した場合の回路の構成例を示す図。

【図13】CNN回路の動作タイミングを模式的に説明する図。

【図14】外部メモリ111における乱数シードの管理形態を説明する図。

【発明を実施するための形態】

【0012】

以下、添付図面を参照し、本発明の好適な実施形態について説明する。なお、以下説明する実施形態は、本発明を具体的に実施した場合の一例を示すもので、特許請求の範囲に記載した構成の具体的な実施例の1つである。

30

【0013】

[第1の実施形態]

先ず、本実施形態に係る情報処理装置の構成例について、図1のブロック図を用いて説明する。本実施形態では、画像から抽出した n 次元の特徴ベクトル(変換元ベクトル)を乱数射影により、 k ($k < n$) 次元(より低次元)の特徴ベクトルに変換(次元削減)し、該変換した k 次元の特徴ベクトルを用いて該画像に対する認識処理を行う。

【0014】

画像データ入力処理部112、演算部101、DMAC(ダイレクトメモリアクセスコントローラ)108、CPU109、外部メモリ111は何れもシステムバス110に接続されており、システムバス110を介して互いにデータ通信を行うことができる。

40

【0015】

外部メモリ111は、システムバス110に不図示のメモリコントローラを介して接続されるDRAMであり、この外部メモリ111には、乱数発生用のシード(乱数シード)が k 個登録されている。

【0016】

画像データ入力処理部112は、本装置外若しくは本装置内の不図示のメモリから画像のデータを取得する。そしてこの画像は、DMAC108によって、システムバス110を介して外部メモリ111に転送される。

【0017】

その後、DMAC108は、外部メモリ111に転送した画像を画素ブロック単位で演

50

算部 101 に転送するので、演算部 101 は、この転送された画素ブロックから特徴ベクトルを抽出し、該抽出した特徴ベクトルの次元を、乱数射影により削減する。そして演算部 101 は、該次元が削減された特徴ベクトルを用いて識別処理を行う。

【0018】

DMAC (ダイレクトメモリアクセスコントローラ) 108 は、CPU 109 からの指示に応じて、外部メモリ 111 に対するデータの書き込みやデータの読み出しを行う。CPU 109 は、図 1 に示した各機能部の動作制御を行う。

【0019】

次に、演算部 101 についてより詳細に説明する。

【0020】

10

特徴抽出処理部 102 は、外部メモリ 111 から転送された画素ブロックごとに、該画素ブロックから n 次元の特徴ベクトル (要素数が n のベクトル) を抽出する。この特徴ベクトルは、例えば、輝度勾配の方向をピンとしたヒストグラムや複数画像フレーム間の画素の動きベクトル (オプティカルフロー等) の方向をピンとしたヒストグラム等、識別目的に応じて定める様々な特徴量に基づくデータ列である。

【0021】

内積演算処理部 103 は、特徴抽出処理部 102 が求めた n 次元の特徴ベクトルと基底ベクトル生成処理部 105 が生成したそれぞれの基底ベクトルとの内積演算を行うことで、該 n 次元の特徴ベクトルを k 次元の特徴ベクトル (射影ベクトル) に変換する。このような変換処理を画素ブロックごとに行うことで、それぞれの画素ブロックの n 次元の特徴ベクトルを、 k 次元の特徴ベクトルに変換することができる。

20

【0022】

内積演算処理部 103 は、 n 次元の特徴ベクトル X (要素 $x_1 \sim x_n$ から成るベクトル) を k 次元の射影ベクトル P (要素 $p_1 \sim p_k$ から成るベクトル) に変換する場合、以下の式 (1) に従った演算を実行する。

【0023】

【数 1】

$$P = W^T X + B \quad (1)$$

30

【0024】

ここで、 X は上記の n 次元の特徴ベクトル (n 行 1 列のベクトル)、 B は k 次元のバイアス項 (要素 $b_1 \sim b_k$ から成るベクトルであり、 k 行 1 列のベクトル)、 P は上記の k 次元の射影ベクトル (k 行 1 列のベクトル) である。 W は ($n \times k$) 次元の射影行列 (n 行 k 列の行列) であり、 k 個の n 次元基底ベクトル ($E_1 \sim E_k$) により構成される。基底ベクトル $E_1 \sim E_k$ は乱数に基づいて決定したデータ列であり、多数の基底候補となる乱数列から所定の条件で選択された k 個の n 次元ベクトルである。

【0025】

図 4 では、上記式 (1) に従った演算の様子を模式的に示している。例えば、射影ベクトル P (401) の要素 p_1 は、式 (2) に示す如く、射影行列 W (402) における基底ベクトル E_1 と特徴ベクトル X (403) との内積値と、バイアス項 B (404) における要素 b_1 と、の和により算出することができる。

40

【0026】

【数 2】

(2)

$$p_1 = E_1^T X + b_1$$

【0027】

ここで、内積演算処理部 103 の構成例について、図 5 のブロック図を用いて説明する

50

。内積演算処理部 103 に入力された特徴ベクトル X は、シフトレジスタ 501 に格納され、 $m(m-k)$ 個の乗算器 502a ~ 502c のそれぞれに供給される。

【0028】

m 個の乗算器 502a ~ 502c は、射影演算を高速化するために並列で動作するものであり、基底ベクトル生成処理部 105 が生成した基底ベクトルと、シフトレジスタ 501 から供給された特徴ベクトル X と、で対応する要素毎に乗算する。例えば、図 5 において一番上に記されている乗算器から一番下に記されている乗算器までを順に乗算器 1、乗算器 2、... 乗算器 m とし、乗算器 $j(1 \leq j \leq m)$ に入力される基底ベクトルを $E_j = (e_{j1}, \dots, e_{jn})$ とする。このとき、乗算器 j は、 $(e_{j1} \times x_1, e_{j2} \times x_2, \dots, e_{jn} \times x_n)$ を求める。

10

【0029】

m 個の累積加算器 503a ~ 503c は、乗算器と同様に並列で動作するものであり、それぞれ乗算器 502a ~ 502c が求めた乗算結果を合計する。例えば、図 5 において一番上に記されている累積加算器から一番下に記されている累積加算器までを順に累積加算器 1、累積加算器 2、... 累積加算器 m とし、乗算器 j に入力される基底ベクトルを $E_j = (e_{j1}, \dots, e_{jn})$ とする。このとき、累積加算器 j は、 $(e_{j1} \times x_1 + e_{j2} \times x_2 + \dots + e_{jn} \times x_n)$ を求める。

【0030】

すなわち、乗算器と累積加算器とにより、基底ベクトルと特徴ベクトル X との内積結果を求める。図 5 の場合、乗算器と累積加算器とのセットは m セットあるので、 m 個の基底ベクトルと特徴ベクトル X との内積結果を求めることができる。

20

【0031】

内積演算処理部 103 では、シフトレジスタ 501 に格納する特徴ベクトル X の要素と基底ベクトル $E_1 \sim E_m$ の要素の乗算とその結果の累積加算をクロック毎に順次演算する。即ち、 n クロックサイクルで m 個の基底ベクトルに対応する内積演算を完了する。

【0032】

なお、バイアス項の加算については、上記の内積結果にバイアス項を加える（乗算器 j に入力される基底ベクトル E_j に対応するバイアス項 B の要素を b_j としたとき、累積加算器 j が $e_{j1} \times x_1 + \dots + e_{jn} \times x_n + b_j$ を求める）ようにしても構わない。しかし、以下のようにして上記の内積演算及びバイアス項 B の加算を行うようにしても構わない。

30

【0033】

即ち、特徴抽出処理部 102 は、特徴ベクトル X の要素に要素値 1 の要素を追加し、要素 $x_1 \sim x_n$ 及び要素値 1 を有する要素から成る $(n+1)$ 次元の特徴ベクトル X' を生成する。また、基底ベクトル生成処理部 105 は、生成したそれぞれの基底ベクトル E について、該基底ベクトル E の n 個の要素に、バイアス項において該基底ベクトルに対応する要素を追加して $(n+1)$ 次元の基底ベクトル E' を生成する。そして、乗算器及び累積加算器によりこのような特徴ベクトル X' とそれぞれの基底ベクトル E' との内積演算を行う。

【0034】

従って、バイアス項まで加味すると、 $(n+1)$ クロックで射影ベクトルの要素値 $p_1 \sim p_m$ が算出される。なお、*CLR は特徴ベクトルと基底ベクトルの内積演算及びバイアス項の加算を終了して射影ベクトル要素の出力後に累積加算器を初期化するための信号である。このようにして内積演算処理部 103 は、射影ベクトル P を構成する k 個の要素のうち m 個の要素を並列に算出して出力することができる。

40

【0035】

次に、内積演算処理部 103 に供給される射影行列 W 、すなわち、 k 個の n 次元の基底ベクトルの生成について説明する。この k 個の基底ベクトルは、乱数生成処理部 106 及び基底ベクトル生成処理部 105 によって生成される。

【0036】

50

本実施形態では、外部メモリ 111 には、基底ベクトルごとに乱数シード（固有の乱数列を生成するためのデータ）が格納されている。本実施形態では、基底ベクトルは k 個生成するので、外部メモリ 111 には、 k 個の乱数シードが格納されていることになる。

【0037】

より詳しくは図 14 に示す如く、外部メモリ 111 には、 k 個の乱数シード（Seed 1 ~ Seed k ）と、バイアス項の要素（ $b_1 \sim b_k$ ）と、がセットになって登録されている。Seed 1 ~ Seed k はそれぞれ、基底ベクトル $E_1 \sim E_k$ を生成するための乱数シードであり、 $b_1 \sim b_k$ はそれぞれ、基底ベクトル $E_1 \sim E_k$ に対応するバイアス項である。これらのセットは、DMAC 108 によってシードバッファ 107 に転送される。なお、以下の説明では、バイアス項は乱数シードの一部であるものとして説明する。例えば、バイアス項 b_1 は、Seed 1 の一部であるものとして説明する。

10

【0038】

CPU 109 は、適当なタイミングで DMAC 108 を起動し、外部メモリ 111 に格納されている k 個の乱数シードを、システムバス 110 を介してシードバッファ 107 に転送させる。シードバッファ 107 は高速な SRAM（スタティックランダムアクセスメモリ）により構成する。シードバッファ 107 は、乱数生成処理部 106 による読み出しと DMAC 108 による書き込みとが同時に実行可能な少容量のダブルバッファメモリ等により構成する。

【0039】

乱数生成処理部 106 は、シードバッファ 107 に転送された k 個の乱数シードを用いて、 k 個の乱数列を生成する。乱数生成処理部 106 の構成例について、図 6 のブロック図を用いて説明する。

20

【0040】

乱数生成は M 系列による乱数生成手法等、従来知られている手法で良い。 M 系列の乱数系列長は基底ベクトルの次元数と基底ベクトル選択の候補となる乱数ベクトルの数に基づいて決定する。例えば、“特徴ベクトル次元数 $\times$ 射影ベクトル次元数 $\times$ 基底ベクトル候補数”の周期を超える乱数が生成できる乱数系列長を決定する。ここで基底ベクトル候補とは、基底ベクトルを選択するための候補となるベクトルである。

【0041】

図 6 に示す如く、乱数生成処理部 106 には、図 5 の内積演算器の並列処理数 m に対応して m 個の線形帰還シフトレジスタ（線形帰還シフトレジスタ 1 ~ m ）が備わっており、所定のレジスタ長を有する。線形帰還シフトレジスタ 1 ~ m にはそれぞれ、シードバッファ 107 に転送された乱数シード 1 ~ m が入力され、 M 系列の乱数を並列に生成する。ロード信号は各シフトレジスタ値のロード信号であり、シードバッファ 107 から読み出した乱数シードを所定のタイミングで線形帰還シフトレジスタにロードするための信号である。乱数生成処理部 106 は、内積演算処理部 103 で k 次元の射影ベクトルを生成する場合、 k 個のシードに基づく k 個の乱数列を生成する（基底ベクトルを一樣乱数から生成する場合）。

30

【0042】

図 1 に戻って、基底ベクトル生成処理部 105 は、乱数生成処理部 106 で生成した k 個の乱数列から所定の乱数分布に対応する k 個の基底ベクトルを取り出す。基底ベクトルが一樣乱数からなる場合、乱数シードから生成した一樣乱数列をそのまま基底ベクトルとする。一方、基底ベクトルが正規乱数からなる場合、複数の一樣乱数列の平均乱数列を生成し、該平均乱数列から正規乱数に近似した基底ベクトルを取り出す。また、基底ベクトル生成処理部 105 では、前述した様に、バイアス項に相当するデータを基底ベクトル列に付加する。

40

【0043】

このように、本実施形態では、外部メモリ 111 には基底ベクトルではなく、基底ベクトルに対応する乱数シードを登録しておき、演算部 101 は外部メモリ 111 から乱数シードを取得し、該取得した乱数シードに基づいて基底ベクトルを生成する。これにより、

50

外部メモリ 111 から演算部 101 に転送するデータ量を、外部メモリ 111 に基底ベクトルを登録した場合と比べてより軽減させることができる。

【0044】

そして内積演算処理部 103 は、基底ベクトル生成処理部 105 にて順次生成する基底ベクトル $E_1 \sim E_m$ と、シフトレジスタ 501 に格納されている特徴ベクトル X と、の内積演算を行う。そして、基底ベクトル $E_1 \sim E_k$ と特徴ベクトル X との内積演算が完了した時点で、射影ベクトル P の要素 $p_1 \sim p_k$ が決定するので、射影ベクトル P が確定することになる。内積演算処理部 103 は、このようにして確定した射影ベクトル P を識別処理部 104 に転送する。

【0045】

識別処理部 104 は、この射影ベクトル P を用いて、規定の判定処理を実行することで、画像内のオブジェクトに対する識別処理を行う。例えば、予め学習によって求めた正解データとなる正解射影ベクトルとのハミング距離やユークリッド距離に基づいて判定を行う。或いは SVM (サポートベクターマシン)、ニューラルネットワーク、ブースティング等の判別器を用いて判定しても良い。画像内の特徴量を用いた識別処理については周知の技術であるので、これに係る詳細な説明は省略する。そして識別処理部 104 は、その識別結果 (判定結果) を、システムバス 110 を介して外部メモリ 111 に転送し、そこに格納する。もちろん、識別結果の出力先については外部メモリ 111 に限るものではなく、外部装置に対して転送しても良いし、不図示の表示装置に表示させても良い。

【0046】

以上説明した各処理を実行するために CPU 109 が行う処理について、同処理のフローチャートを示す図 7 を用いて説明する。CPU 109 は、外部メモリ 111 等のメモリに格納されているコンピュータプログラムやデータを用いて処理を実行することで、図 7 のフローチャートに従った処理を実行する。

【0047】

ステップ S701 では、CPU 109 は、情報処理装置を構成する各ハードウェア (図 1 に示した各機能部を含む) の初期化処理を行う。この初期化処理には、各機能部の動作パラメータを設定する処理も含まれる。ステップ S702 では、CPU 109 は、画像データ入力処理部 112、演算部 101、DMAC 108 の各機能部に動作開始指示を送出する。

【0048】

ステップ S703 では、CPU 109 は、画像データ入力処理部 112 が取得した画像を外部メモリ 111 に転送させるべく、DMAC 108 にこの転送を指示するので、DMAC 108 はこの転送を行う。

【0049】

ステップ S704 では、CPU 109 は、DMAC 108 の出力する割り込み信号により画像が転送されたことを検知すると、DMAC 108 に、外部メモリ 111 に格納されている画像の画素ブロック単位の転送及び乱数シードの転送を指示する。DMAC 108 は CPU 109 により指定されたディスクリプタに従って、外部メモリ 111 に格納されている画像を画素ブロック単位で特徴抽出処理部 102 に転送すると共に、外部メモリ 111 に格納されている乱数シードをシードバッファ 107 に転送する。

【0050】

ここで、画素ブロックについて図 9 を用いて説明する。画素ブロック 92 とは、画像 91 を複数の領域に区分した場合に、該領域内の画素から成るブロックである。本実施形態では、画素ブロック 92 内の特徴を代表するベクトルを特徴量 (特徴ベクトル) とし、画素ブロック単位で処理を行うことで、結果として画像全体に対して処理する。然るに、識別処理部 104 による識別結果は、画素ブロックごとに得られることになる。

【0051】

特徴抽出処理部 102 は、外部メモリ 111 から画素ブロックが転送されるたびに、該画素ブロックから特徴ベクトルを抽出し、該抽出した特徴ベクトルを内積演算処理部 10

10

20

30

40

50

3 に送出する。

【 0 0 5 2 】

シードバッファ 1 0 7 に乱数シードが転送されると、乱数生成処理部 1 0 6 はこの乱数シードを用いて乱数列を生成し、基底ベクトル生成処理部 1 0 5 は、この乱数列を用いて基底ベクトルを生成し、該生成した基底ベクトルを内積演算処理部 1 0 3 に送出する。

【 0 0 5 3 】

内積演算処理部 1 0 3 は、特徴抽出処理部 1 0 2 から送出された特徴ベクトルと、基底ベクトル生成処理部 1 0 5 から送出されたそれぞれの基底ベクトルと、の内積演算を行って、射影ベクトルを生成する。

【 0 0 5 4 】

ここで、内積演算処理部 1 0 3 は、図 5 に示す様に m 個の基底ベクトルに対する処理を並列に実行する。従って射影ベクトルの次元を k とすると、天井関数 (k / m) 回の基底ベクトルに対する内積演算の処理時間で射影ベクトルを生成する。

【 0 0 5 5 】

識別処理部 1 0 4 は、内積演算処理部 1 0 3 で算出された射影ベクトルを受け取ると、識別処理を開始する。DMA C 1 0 8 は識別処理部 1 0 4 による識別結果を外部メモリ 1 1 1 に転送する。

【 0 0 5 6 】

そしてDMA C 1 0 8 は、外部メモリ 1 1 1 への識別結果の転送を完了すると、CPU 1 0 9 に対して割り込み信号を出力するので、ステップ S 7 0 5 では、CPU 1 0 9 は、この割り込み信号を受信する。

【 0 0 5 7 】

ステップ S 7 0 6 では、CPU 1 0 9 は、全ての画素ブロックについてステップ S 7 0 4 及び S 7 0 5 の処理を行ったか否かを判断する。この判断の結果、全ての画素ブロックについてステップ S 7 0 4 及び S 7 0 5 の処理を行った場合には、処理はステップ S 7 0 7 に進む。一方、全ての画素ブロックについてステップ S 7 0 4 及び S 7 0 5 の処理を行っていない場合には、未処理の画素ブロックについてステップ S 7 0 4 及び S 7 0 5 の処理が行われる。

【 0 0 5 8 】

ステップ S 7 0 7 では、CPU 1 0 9 は、全てのフレームについてステップ S 7 0 3 ~ S 7 0 6 の処理を行ったか否かを判断する。この判断の結果、全てのフレームについてステップ S 7 0 3 ~ S 7 0 6 の処理を行った場合には、図 7 のフローチャートに従った処理は終了する。一方、全てのフレームについてステップ S 7 0 3 ~ S 7 0 6 の処理を行っていない場合には、未処理のフレームについてステップ S 7 0 3 ~ S 7 0 6 の処理が行われる。

【 0 0 5 9 】

次に、情報処理装置の動作のタイムチャートについて、図 8 を用いて説明する。図 8 では、ある 1 つの画素ブロックに対する動作のタイムチャートを示している。また、図 8 では、内積演算処理部 1 0 3 及び乱数生成処理部 1 0 6 の並列度が $k / m = 2$ となる場合の例を示している。

【 0 0 6 0 】

まずDMA C 1 0 8 によって外部メモリ 1 1 1 に格納されている乱数シードをシードバッファ 1 0 7 に転送する (8 0 1、8 0 3)。特徴抽出処理部 1 0 2 は、この転送に並行して特徴抽出処理を実行する (8 1 5)。特徴抽出処理 (8 1 5) と乱数シードの転送 (8 0 1、8 0 3) を終了すると、シードバッファ 1 0 7 からの乱数シードの読み出し (8 0 6)、乱数生成処理 (8 0 8)、基底ベクトル生成処理 (8 1 0) を順次実行する。生成された基底ベクトルと格納済みの特徴ベクトルに対して内積演算処理を実行する (8 1 2)。ここでは、内積演算処理部 1 0 3 で並列に処理する基底ベクトルの個数分の内積演算が実行される。ここでの一連の処理と並行して、次の内積演算処理 (8 1 3) のための乱数シードの転送を開始する (8 0 2、8 0 4)。内積演算処理 (8 1 2) が終了すると

10

20

30

40

50

、シードバッファ107に格納された乱数シードを参照して乱数生成処理(809)、基底ベクトル生成処理(811)を順次実行する。更に、生成された基底ベクトルと特徴抽出処理815で抽出された特徴ベクトルとの内積演算を実行する(813)。内積演算(813)を終了すると、得られた射影ベクトルに対して識別処理(814)を開始する。この様に本実施形態では、外部メモリに格納する少量の乱数シードに基づいて基底ベクトルを動的に生成しながら射影演算を実行する。

【0061】

以上説明したように、本実施形態では、多次元の基底ベクトルを保持する代わりに基底ベクトルを生成するための少量の乱数シードを外部メモリに保持する事で、外部メモリの利用サイズを数分の1～数百分の1に削減する事ができる。これにより、システムバス110に必要な帯域を数分の1～数百分の1に削減する事ができる。これにより、バス帯域に制約があるシステムでも選択的な乱数列に基づくランダムプロジェクションによる射影演算処理を高速に処理する事が可能になる。

【0062】

[第2の実施形態]

本実施形態では、CPU109が好適な乱数シードを生成(学習)して外部メモリ111に登録する点が、第1の実施形態と異なる。以下では、第1の実施形態との相違点について重点的に説明し、第1の実施形態と同様の点については説明を省略する。乱数シードを生成して外部メモリ111に登録する処理について、図10のフローチャートを用いて説明する。

【0063】

ステップS1001では、CPU109は、演算部101を構成する各機能部を初期化する。この初期化では、例えば、乱数生成処理部106がパラメータにより線形帰還シフトレジスタの乱数周期(乱数系列長)や帰還構成を変更できる場合、その周期や帰還構成に関するパラメータを設定する。乱数系列長は前述した様に「特徴ベクトル次元数×射影ベクトル次元数×基底ベクトル候補数」に基づいて決定する。また、M乱数発生器の場合、帰還構成は数系列長に基づいて定まる。

【0064】

ステップS1002では、CPU109は、十分な個数分の乱数シード候補を生成する。生成方法は如何なる方法でも構わない。また、生成したそれぞれの乱数シード候補に対応するバイアス項を、乱数を用いて生成する。そしてステップS1003では、CPU109は、生成した乱数シード候補のうち未選択の1つを乱数生成処理部106が有する線形帰還シフトレジスタのうち1つに設定し、該設定された乱数シード候補から乱数列を該線形帰還シフトレジスタに生成させる。

【0065】

ステップS1004では、基底ベクトル生成処理部105は、ステップS1003で生成された乱数列から1つの基底ベクトルを生成する。ステップS1005で内積演算処理部103はステップS1004で生成された基底ベクトルと複数の学習画像のそれぞれから予め抽出した特徴ベクトルとの内積演算を行い、該基底ベクトルを生成する為に用いた乱数シード候補に対応するバイアス項を加算する。

【0066】

そしてCPU109は、複数の学習画像のそれぞれに対して求めた「該学習画像の特徴ベクトル $\times$ とステップS1003で選択した乱数シード候補に応じた基底ベクトル E との内積結果にバイアス項 b を加算した結果(値)」を用いて、以下の式(3)により該基底ベクトルを評価する。

【0067】

10

20

30

40

【数 3】

$$\frac{1}{N} \sum_{i=1}^N L(E^T \mathbf{x}_i + b) + \lambda b \quad (3)$$

【0068】

ここで、Nはデータ数、λはバイアス重みパラメータである。また、L(z)は誤差を表す関数であり、例えば、以下の式(4)のように定義される。

【0069】

【数 4】

$$L(z) = \begin{cases} 0 & (z \geq 0) \\ z^2 & (z < 0) \end{cases} \quad (4)$$

【0070】

そしてCPU109は、ステップS1003で選択した乱数シード候補と、ステップS1005での評価結果(評価値)と、をセットにして外部メモリ111に格納する。

【0071】

ステップS1006では、CPU109は、生成した全ての乱数シード候補についてステップS1003～S1005の処理を行ったか否かを判断する。この判断の結果、生成した全ての乱数シード候補についてステップS1003～S1005の処理を行った場合には、処理はステップS1007に進む。一方、生成した全ての乱数シード候補についてステップS1003～S1005の処理を行っていない場合には、処理はステップS1003に進み、未選択の乱数シード候補について以降の処理を行う。

【0072】

ステップS1007では、CPU109は、外部メモリ111に格納されたそれぞれの乱数シード候補のうち評価値が高い順に上位k個の乱数シード候補を「射影行列Wを構成する基底ベクトルE1～Ekを生成するための乱数シード」として選択する。そしてCPU109は改めて、この選択したk個の乱数シード候補と、該乱数シード候補に対応するものとしてステップS1002で生成されたバイアス項と、をセットにして、図14に示すような形式で外部メモリ111に登録する。また、ここで、周期や帰還構成に関するパラメータも関連付けて格納する。

【0073】

そして、本実施形態で外部メモリ111に登録したパラメータや乱数シードを用いて、第1の実施形態で説明した一連の処理を行うことで、学習処理時に評価選択した基底ベクトルと同じ基底ベクトルを乱数シードから生成することができる。

【0074】

なお、基底ベクトルの選択方法は上記の方法に限るわけではなく、適用する用途に応じて様々な選択手法が考えられる。例えば、本実施形態では予め候補となる乱数シードを生成して各乱数列に基づく基底ベクトルを評価する場合について説明したが、他の方法でも良い。例えば、一つの乱数シードに基づく乱数列の中から、所定の長さの乱数列を切り出す事で順次基底ベクトルを生成する方法でも良い。この場合、所定の長さの乱数列を切り出す単位で乱数生成器の内部状態情報(線形帰還レジスタの場合はレジスタ値)を取り出してその情報を基底ベクトルに対応する乱数シードとする。

【0075】

[第3の実施形態]

本実施形態では、第1の実施形態で説明したような射影演算処理をCNN(Convolutional Neural Networks: コンボリュショナルニューラルネットワーク)に適用した場合について説明する。まずは、CNNの構成例について、図11を用いて説明する。本実施形態では、CNNは、入力された画像から特定の物体を検

10

20

30

40

50

出するものとして説明する。

【 0 0 7 6 】

CNNは階層的な特徴抽出処理により構成する。ここでは、第1層1106の特徴数が3、第2層1110の特徴数が2、第3層1113の特徴数が1、の3階層CNNの例を示している。CNNは階層的な特徴抽出処理により対象物の見えの変動等に対して頑健な手法として知られている。

【 0 0 7 7 】

このようなCNNに画像1101を入力する。1103a~1103cは第1階層1106の特徴面を示す。特徴面とは、所定の特徴抽出フィルタ（コンボリューション演算の累積和及び非線形処理）で前階層のデータを走査しながら演算した結果を格納する画像データ面である。特徴面はラスタスキャンされた画像データに対する検出結果であるため検出結果も面で表す。特徴面1103a~1103cは、画像1101を参照して、異なる特徴抽出フィルタにより算出する。特徴面1103a~1103cはそれぞれ模式的に1104a~1104cに対応する2次元のコンボリューションフィルタ演算と演算結果の非線形変換（シグモイド関数等）により生成する（以下、非線形コンボリューション演算と呼ぶ）。なお、1102は非線形コンボリューション演算に必要な参照画像領域を示す。フィルタサイズ（水平方向の長さと垂直方向の高さ）が11×11のコンボリューションフィルタ演算は以下に示すような積和演算により処理する。

【 0 0 7 8 】

【 数 5 】

$$output(x,y) = \sum_{row=-RowSize/2}^{rowSize/2} \sum_{column=-columnSize/2}^{columnSize/2} input(x+column,y+row) \times weight(column,row) \quad (5)$$

input(x,y): 座標(x,y)での参照画素値
 output(x,y): 座標(x,y)での演算結果
 weight(column, row): 座標(x+column,y+row)での重み係数
 columnSize=11, rowSize=11: フィルタサイズ（フィルタタップ数）

【 0 0 7 9 】

1104a~1104cは夫々異なる係数のコンボリューションフィルタである。なお、階層によってコンボリューションフィルタのサイズも異なる。CNNでは複数のフィルタを画素単位で走査しながら積和演算を繰り返し、最終的な積和結果を非線形変換する事で特徴面を生成する。特徴面1103a~1103cを算出する場合は前階層との結合数が1であるため、フィルタは1つ（1104a）である。一方、特徴面1107a及び特徴面1107bを計算する場合、前階層との結合数が3であるため夫々1108a~1108c及び1108d~1108fに相当する3つのコンボリューションフィルタの演算結果を累積加算する。つまり、特徴面1107aは、コンボリューションフィルタ1108a~1108cの全ての出力を累積加算し、最後に非線形変換処理する事によって得る。1105a~1105cはそれぞれ第2層1110の非線形コンボリューション演算に必要な参照画像領域を示す。即ち特徴面は演算は以下の式に基づいて算出される。

【 0 0 8 0 】

【数 6】

$$feature(x, y) = sigmoid\left(\sum_{k=1}^{feature_number-1} output_k(x, y) + b\right) \quad (6)$$

$feature(x, y)$: 注目対象階層の特徴面の注目位置 (x, y) の値
 $output(x, y)$: 前階層の特徴面に対するコンボリユーション演算結果
 $Feature_number$: 前階層に対する結合の数
 b : バイアス項
 $sigmoid$: シグモイド関数

10

【0081】

第3層（特徴面1111）では、前階層の特徴面1107a、1107bに対する非線形コンボリユーション演算により検出対象の存在尤度を算出する（1112）。ここで、式（6）は一つのコンボリユーションフィルタを基底ベクトルとすると式（1）に示す射影演算の総和（前階層への結合数*i*）に非線形関数演算を追加したものと捉える事ができる（式（7））。つまり、非線形コンボリユーション演算を非線形射影演算と定義できる（射影後の次元は特徴面の数に対応することになる）。

【0082】

【数 7】

$$p = sigmoid\left(\sum_i E_i^T X + b\right) \quad (7)$$

20

E_i : フィルタ係数（基底ベクトル）
 X : 前階層特徴面のフィルタ参照領域
 b : バイアス項
 p : 算出する特徴面の注目位置における非線形コンボリユーション演算結果

【0083】

一般的にコンボリユーションフィルタの係数 $E_i (= weight)$ はバックプロパゲーション等により学習される。一方、非特許文献2に開示されている様に、下位階層（本実施形態の場合、第1層1106、第2層1110）の係数 $Weight$ に乱数を使用しても良い検出性能を示す事が知られている。本実施形態では第1層1106、第2層1110のコンボリユーションフィルタ1104a～1104c、1108a～1108fの係数が乱数から生成するものとする。

30

【0084】

上記のCNNをハードウェアを用いて実現した場合の回路の構成例について、図12を用いて説明する。1201は非線形コンボリユーション演算処理部であり、上記の式（7）に示す非線形射影演算を実行する。非線形コンボリユーション演算処理部1201は内積演算器、累積加算器、非線形変換器等から構成されるものである。1202は特徴面データ保持メモリであり、非線形コンボリユーション演算によって得られる特徴面（1103a～1103c、1107a～1107b、及び1111）を格納するメモリである。1205はシードメモリであり、コンボリユーション演算のフィルタ係数に相当する基底ベクトルを乱数から生成するための乱数シード群及びバイアス項を格納する。シードメモリ1205はROMや高速なSRAM等により構成される。SRAMにより構成する場合、図示しない記憶部に格納する乱数シードをデータバスを介して階層処理毎に予め転送する。1204は乱数生成処理部であり、シードメモリ1205に格納された乱数シードを順次読みだし、該乱数シードに基づいて乱数列を生成する。基底ベクトル生成処理部1203は、乱数生成処理部1204で生成した乱数列からフィルタ係数となる基底ベクトルを生成する。なお、乱数生成処理部1204や基底ベクトル生成処理部1203は、図1に示した乱数生成処理部106や基底ベクトル生成処理部105と同様である。

40

50

【 0 0 8 5 】

図 1 3 は C N N 回路の動作タイミングを模式的に説明する図である。図 1 3 は特徴面 1 1 0 7 a の一つの注目点の値を算出する場合の例を示している。1 3 0 1 a はシードデータの読み出し処理であり、1 1 0 8 a のフィルタ演算の係数に対応する乱数シードのシードメモリ 1 2 0 5 からの読み出しを示す。1 3 0 2 a は乱数生成処理部 1 2 0 4 の動作期間であり、1 3 0 1 a で読みだされた乱数シードに基づいて乱数列を生成する処理を示す。1 3 0 3 a は基底ベクトル生成処理部 1 2 0 3 の動作期間であり、乱数生成処理部 1 2 0 4 で生成された乱数列に基づいてコンボリューションフィルタ 1 1 0 8 a の係数に相当する基底ベクトルを生成する。即ち乱数列から所定の条件でフィルタ係数列を取り出す。1 3 0 4 a は特徴面データ保持メモリ 1 2 0 2 からのデータ読み出し期間を示す。ここでは 1 1 0 5 a に相当する特徴面の領域データを読み出す。1 3 0 5 a は非線形コンボリューション演算処理部 1 2 0 1 の動作期間を示す。ここでは基底ベクトル生成処理部 1 2 0 3 で生成されたフィルタ係数と特徴面データ保持メモリ 1 2 0 2 から読み出された前階層の特徴データとの内積演算を実行し、その結果を累積加算器に格納する。同様に 1 3 0 1 b ~ 1 3 0 5 b のタイミングで前階層の特徴 1 1 0 5 b に対して 1 1 0 8 b に示すフィルタ演算を実行し、その結果を累積加算する。更に同様に 1 3 0 1 c ~ 1 3 0 5 c のタイミングで前階層の特徴 1 1 0 5 c に対して 1 1 0 8 c に示すフィルタ演算を実行し、その結果を累積加算する。1 3 0 6 は 3 つのフィルタ演算の結果の累積和を非線形変換処理するタイミングである。なお、バイアス項は図示しないメモリに階層毎に格納し、その加算も 1 3 0 6 で処理されるものとする。非線形変換処理された累積加算結果は 1 3 0 7 で特徴面データ保持メモリ 1 2 0 2 に格納する。以上の処理をラスタスキャン順に処理する事で特徴面 1 1 0 7 a を算出する。

10

20

【 0 0 8 6 】

この様に C N N は特徴面データ保持メモリ 1 2 0 2 に格納された特徴面データを参照しながら階層的に非線形コンボリューション演算を繰り返す事で物体検出等を実現する。なお、図 1 1 は説明のために簡略化した C N N の構造を示すが、一般的には多数の特徴面を有する多階層のネットワークで構成される。多層かつ多数の特徴面を有するネットワークで構成する場合、コンボリューション演算のフィルタ係数を保持するためのメモリ容量がコスト増大の要因となる。本実施形態では、乱数に基づく C N N 回路において、複数のフィルタカーネル群のフィルタ係数を乱数シードで保持する事により該メモリの回路規模を大幅に削減する事が可能になる。また、係数を外部のメモリ等に保持する場合、転送コストを削減する事ができる。

30

【 0 0 8 7 】

[第 1 乃至 3 の実施形態の変形例について]

上記の実施形態には様々な変形例が考え得る。例えば、上記の実施形態では、情報処理装置 (C N N 回路) は、画像データを対象としていたが、これに限るものではなく、音声データ、テキスト情報、時系列信号データ等、特徴ベクトルを抽出可能な様々なデータを対象とすることができる。

【 0 0 8 8 】

また、上記の実施形態では、基底ベクトルに対応する乱数シードを外部メモリ 1 1 1 に格納する場合について説明したが、外部メモリ 1 1 1 の代わりに情報処理装置内に R O M を設け、この R O M に乱数シードを格納するようにしても構わない。この場合、R O M のサイズを大幅に削減する事ができる。

40

【 0 0 8 9 】

また、上記の実施形態では、乱数シードから基底ベクトルを生成する方法として、1 つの乱数シードから 1 つの基底ベクトルを生成する方法や、複数の乱数シードから 1 つの基底ベクトルを生成する方法 (正規乱数の場合) について説明した。しかし、乱数シードから基底ベクトルを生成する方法はこれに限るものではない。例えば、1 つの乱数シードから所定のまとまりの複数の基底ベクトルを生成するようにしても構わない。乱数シードと基底ベクトルとの関係は学習時の基底ベクトル生成方法に依存して任意である。

50

【0090】

また、上記の実施形態では、線形帰還レジスタによりM系列の乱数を生成する方法について説明したが、これに限るものではなく、乱数シードから一意に定まる乱数生成方式であれば如何なる方法を用いても構わない。

【0091】

また、第1の実施形態では、シードバッファ107をダブルバッファメモリで構成する場合について説明したが、例えば、シードバッファ107を設ける事無く、乱数生成処理部106に乱数シードを直接格納するようにしても構わない。また、第1の実施形態では、射影ベクトルPを生成する場合について説明したが、射影ベクトルPの各要素を2値化してHash値として利用しても良い。

10

【0092】

また、第3の実施形態では、CNNに適用した場合について説明したが、乱数に基づく係数による複数のコンボリューション演算処理を有する様々な処理に対応可能である。また、一般的なMulti Layer Perceptron型のニューラルネットワークに適用しても良い。この場合、階層型ニューラルネットワークの一つのニューロン値を算出するための前階層ニューロン値に対する結合係数列が上記の基底ベクトルに対応する。更に、以下に開示されているRecursive Neural Networks等に適用しても良い。

【0093】

Socher, R., Manning, C. D., and Ng, A. Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In Deep Learning and Unsupervised Feature Learning Workshop, 2010.

20

また、演算部101を構成する各部は何れもハードウェアで構成しても構わないが、シードバッファ107を除く各部のうち1以上をソフトウェア（コンピュータプログラム）で構成しても構わない。この場合、このコンピュータプログラムは外部メモリ111に格納され、CPU109により実行されることで、対応する機能部の機能を実現することができる。この場合、特にマルチプロセッサ構成の処理装置で空間分割処理する場合等に有効である。

30

【0094】

（その他の実施例）

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはCPUやMPU等）がプログラムを読み出して実行する処理である。

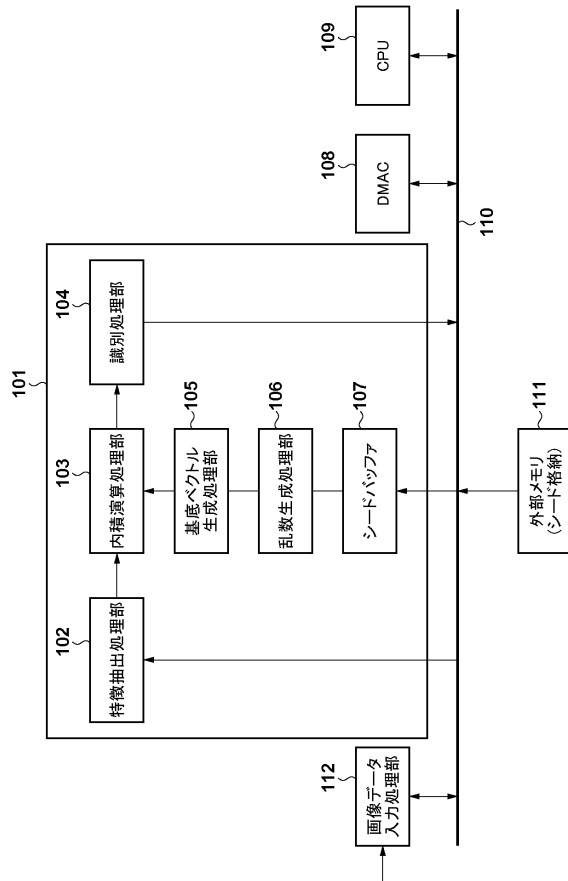
【符号の説明】

【0095】

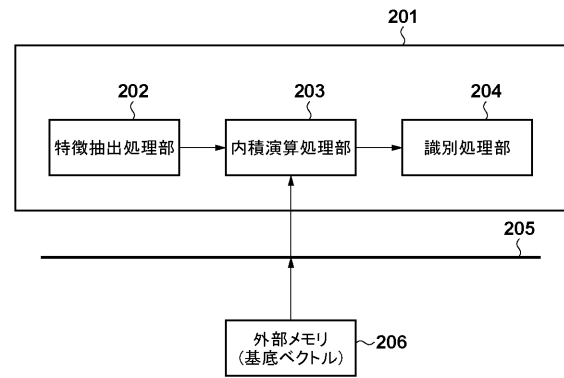
102：特徴抽出処理部 103：内積演算処理部 105：基底ベクトル生成処理部
106：乱数生成処理部

40

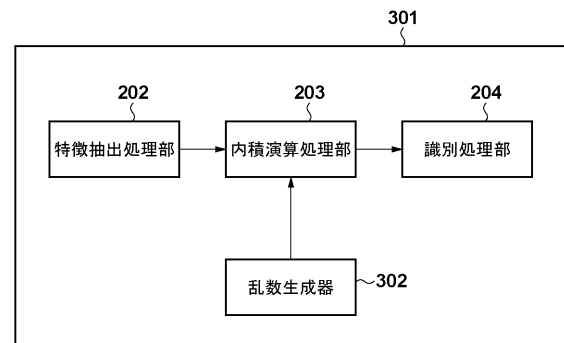
【図 1】



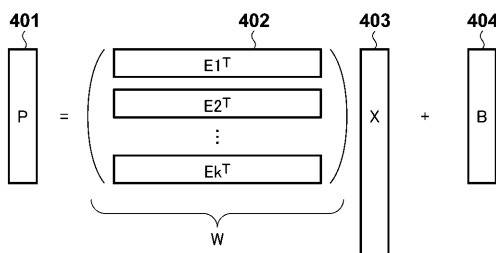
【図 2】



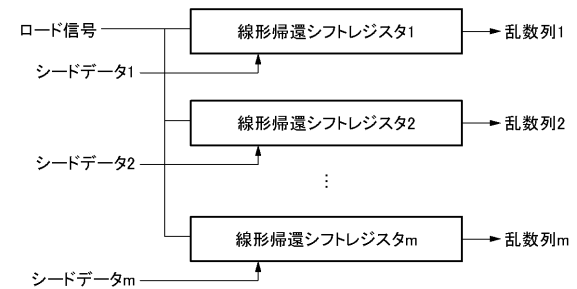
【図 3】



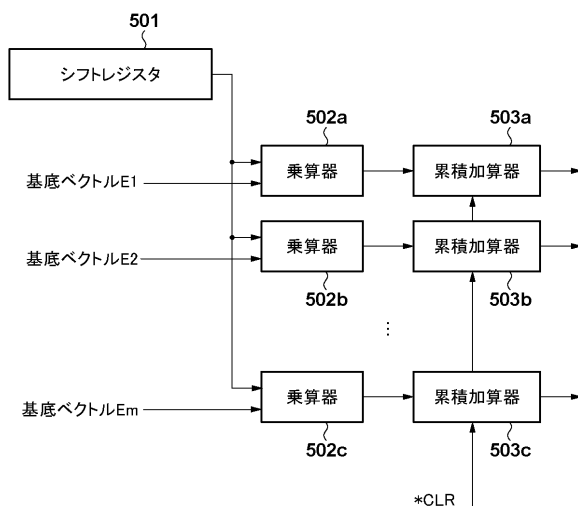
【図 4】



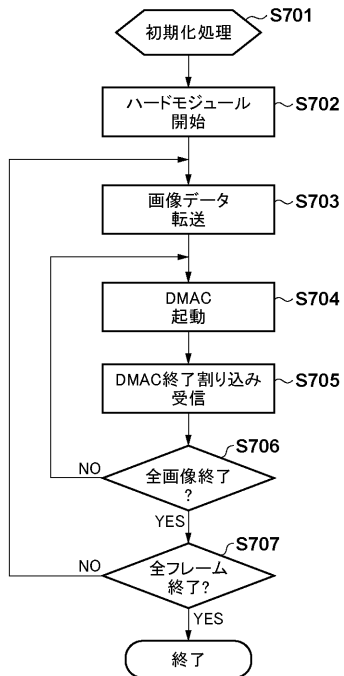
【図 6】



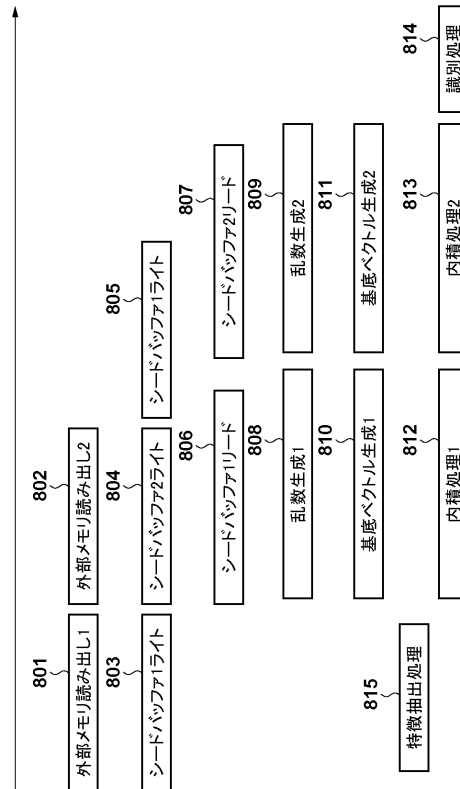
【図 5】



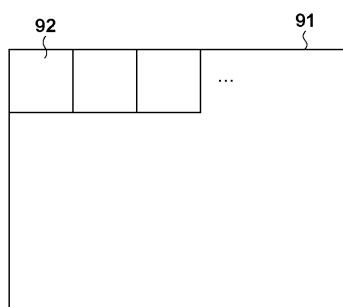
【図 7】



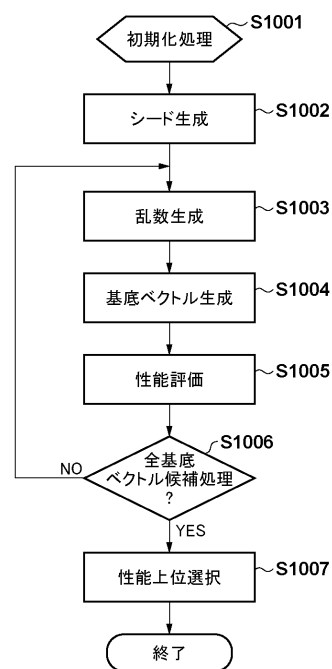
【図 8】



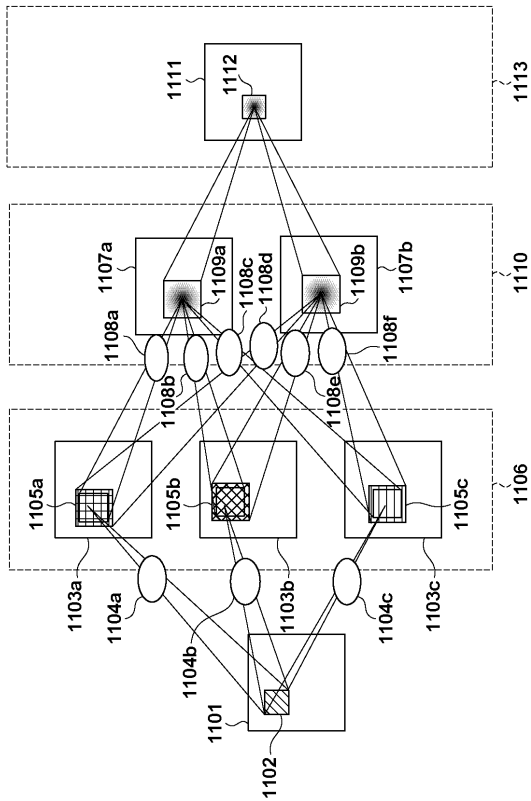
【図 9】



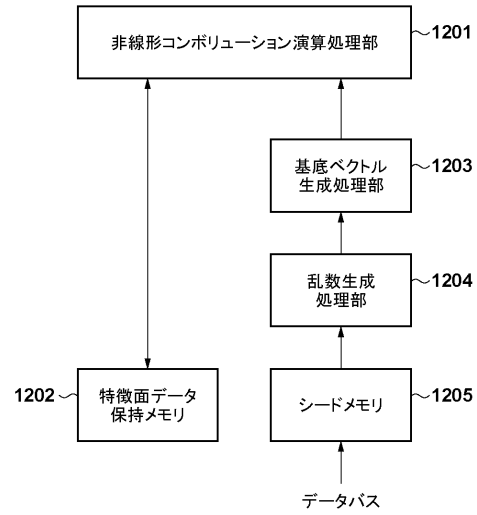
【図 10】



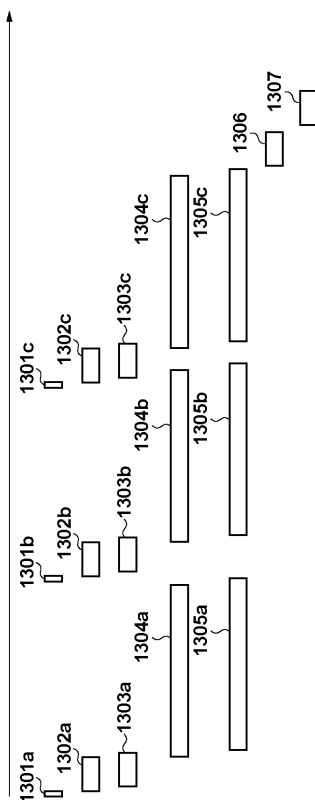
【 図 1 1 】



【圖 12】



【 図 1 3 】



【 図 1 4 】

アドレス0	Seed1	b1
アドレス1	Seed2	b2
	$\vdots$	
アドレスk	Seedk	bk

アドレス0	Seed1	b1
アドレス1	Seed2	b2
	$\vdots$	
アドレスk	Seedk	bk

フロントページの続き

- (72)発明者 加藤 政美
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 八谷 大岳
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 塚田 肇

- (56)参考文献 特開2013-145954(JP,A)
特開平10-313402(JP,A)
特表2007-519983(JP,A)
米国特許出願公開第2005/0213760(US,A1)
米国特許出願公開第2007/0041638(US,A1)
米国特許出願公開第2008/0082468(US,A1)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|---------|-----------|
| G 0 6 N | 9 9 / 0 0 |
| G 0 6 N | 3 / 0 0 |