

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2022年1月20日(20.01.2022)



(10) 国際公開番号

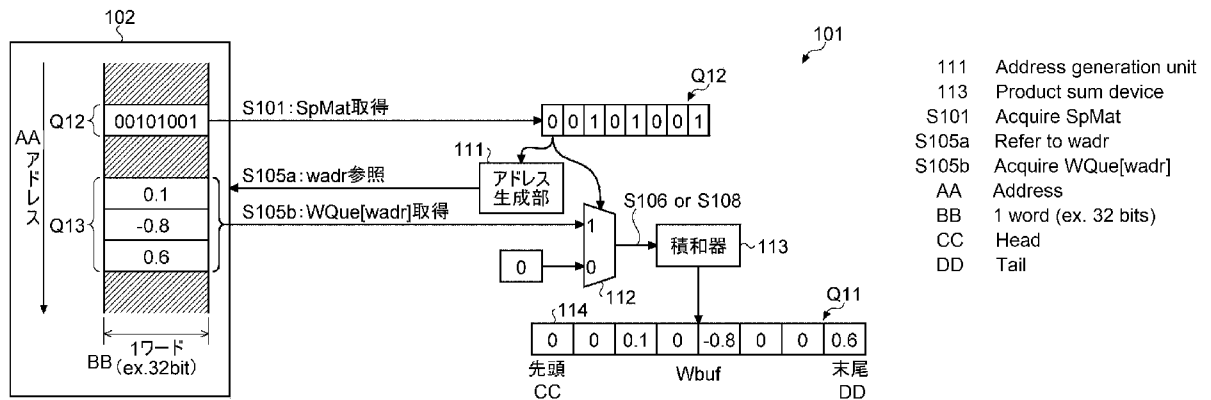
WO 2022/014500 A1

- (51) 国際特許分類:
G06N 3/04 (2006.01) *G06F 17/10* (2006.01)
G06N 3/10 (2006.01)
- (21) 国際出願番号: PCT/JP2021/025989
- (22) 国際出願日: 2021年7月9日(09.07.2021)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2020-123312 2020年7月17日(17.07.2020) JP
- (71) 出願人: ソニーグループ株式会社(SONY GROUP CORPORATION) [JP/JP]; 〒1080075 東京都港区港南1丁目7番1号 Tokyo (JP).
- (72) 発明者: 高木 聡(TAKAGI, Satoshi); 〒1080075 東京都港区港南1丁目7番1号 ソニーグループ株式会社内 Tokyo (JP).
- (74) 代理人: 特許業務法人 酒井国際特許事務所 (SAKAI INTERNATIONAL PATENT OFFICE); 〒1000013 東京都千代田区霞が関3丁目8番1号 虎の門三井ビルディング Tokyo (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ,

(54) Title: NEURAL NETWORK PROCESSING DEVICE, INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING SYSTEM, ELECTRONIC INSTRUMENT, NEURAL NETWORK PROCESSING METHOD, AND PROGRAM

(54) 発明の名称: ニューラルネットワーク処理装置、情報処理装置、情報処理システム、電子機器、ニューラルネットワーク処理方法およびプログラム

【図10】



(57) Abstract: In the present invention, the amount of memory is reduced. A neural network processing device comprises: a decryption unit (41) that decrypts an encrypted coefficient matrix into a zero-coefficient position table that shows the position of a first coefficient for which the value in the coefficient matrix is zero as a first value and shows the position of a second coefficient for which the value in the coefficient matrix is not zero as a second value, and a non-zero coefficient table that holds the second coefficient of the coefficient matrix; and a product-sum circuit (116) that performs a convolution process on a variable matrix and the coefficient matrix decrypted by the decryption unit, the decryption unit storing the second coefficient stored in the non-zero coefficient table in the position shown by the second value of the zero-coefficient position table, whereby the coefficient matrix is decrypted.

NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, WS, ZA, ZM, ZW.

- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

- 一 国際調査報告 (条約第21条(3))

(57) 要約: メモリ量を削減する。ニューラルネットワーク処理装置は、係数行列における値がゼロの第1係数の位置を第1値で示し、前記係数行列における値がゼロでない第2係数の位置を第2値で示すゼロ係数位置テーブルと、前記係数行列における前記第2係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号する復号部(41)と、前記復号部により復号された前記係数行列と、変数行列との畳み込み処理を実行する積和回路(116)とを備え、前記復号部は、前記ゼロ係数位置テーブルにおける前記第2値で示された位置に前記非ゼロ係数テーブルに格納された前記第2係数を格納することで、前記係数行列を復号する。

明 細 書

発明の名称：

ニューラルネットワーク処理装置、情報処理装置、情報処理システム、電子機器、ニューラルネットワーク処理方法およびプログラム

技術分野

[0001] 本技術は、ニューラルネットワーク処理装置、情報処理装置、情報処理システム、電子機器、ニューラルネットワーク処理方法およびプログラムに関する。

背景技術

[0002] 従来、音声認識や話者識別、環境音識別等のオーディオデータに対する識別処理や、画像認識等の画像データに対する識別処理といった自動識別技術が知られている。

[0003] このような識別処理を行うための方法として線形識別や、決定木、サポートベクタマシン、ニューラルネットワークなどを利用した方法が提案されている（例えば、非特許文献1および非特許文献2参照）。

[0004] 例えばニューラルネットワークを利用した識別処理では、予め学習により得られたニューラルネットワーク構造の識別器に対してデータが入力され、識別器によりデータに対する畳み込み処理やプーリング処理、レジデュアル処理等の演算が行われる。そして、識別器による演算結果が、入力されたデータに対する識別結果として出力される。

先行技術文献

非特許文献

[0005] 非特許文献1：Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012

非特許文献2：Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, The MIT Press, 2016

発明の概要

発明が解決しようとする課題

[0006] ところで、近年では自動識別に関する機能をスマートフォンやヘッドフォンなどのポータブル機器に搭載したいという要望もあり、そのためにニューラルネットワーク等を利用した識別処理を行う際に必要となるメモリ量の削減が望まれている。

[0007] 本開示は、メモリ量を削減することが可能なニューラルネットワーク処理装置、情報処理装置、情報処理システム、電子機器、ニューラルネットワーク処理方法およびプログラムを提案する。

課題を解決するための手段

[0008] 実施形態に係るニューラルネットワーク処理装置は、係数行列における値がゼロの第1係数の位置を第1値で示し、前記係数行列における値がゼロでない第2係数の位置を第2値で示すゼロ係数位置テーブルと、前記係数行列における前記第2係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号する復号部と、前記復号部により復号された前記係数行列と、変数行列との畳み込み処理を実行する積和回路とを備え、前記復号部は、前記ゼロ係数位置テーブルにおける前記第2値で示された位置に前記非ゼロ係数テーブルに格納された前記第2係数を格納することで、前記係数行列を復号する。

図面の簡単な説明

[0009] [図1]ニューラルネットワークについて説明する図である。

[図2]畳み込み処理およびプーリング処理について説明する図である。

[図3]フレーム処理について説明する図である。

[図4]第1の実施形態に係るニューラルネットワーク構造の識別器について説明する図である。

[図5]第1の実施形態に係るニューラルネットワーク構造の識別器について説明する図である。

[図6]第1の実施形態に係るフィルタ係数の圧縮符号化について説明する図である。

[図7]第1の実施形態に係るニューラルネットワーク処理装置の構成例を示す図である。

[図8]第1の実施形態に係る識別処理を説明するフローチャートである。

[図9]第1の実施形態に係るスパース表現係数を復号する際の処理の一例を示すフローチャートである。

[図10]第1の実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。

[図11]図10に示す具体例における位相ポインタ及び非ゼロ係数キューポインタの動きを説明するための図である。

[図12]第2の実施形態に係るスパース表現係数を復号する際の処理の一例を示すフローチャートである。

[図13]第2の実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。

[図14]第3の実施形態に係るプライオリティエンコーダを説明するための図である。

[図15]第3の実施形態に係るスパース表現係数を復号する際の処理の一例を示すフローチャートである。

[図16]第3の実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。

[図17]図16に示す具体例における位相ポインタ、非ゼロ係数キューポインタ及び変数キューポインタの動きを説明するための図である。

[図18]第4の実施形態に係るスパース表現係数を復号する際の処理の一例を示すフローチャートである。

[図19]第4の実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。

[図20]図19に示す具体例における位相ポインタ、非ゼロ係数キューポインタ及び変数キューポインタの動きを説明するための図である（その1）。

[図21]図19に示す具体例における位相ポインタ、非ゼロ係数キューポイン

タ及び変数キューポインタの動きを説明するための図である（その２）。

[図22]比較例に係る積和結果のメモリへの書き込みを説明するための図である。

[図23]第5の実施形態に係る符号化部の概略構成例を示す回路図である。

[図24]第6の実施形態に係る符号化部の概略構成例を示す回路図である。

[図25]図24に示す符号化部の動作例を説明するための図である。

[図26]第6の実施形態に係る第1の変形例を説明するための模式図である。

[図27]第6の実施形態に係る第1及び第2の変形例を組み合わせた第3の変形例を説明するための模式図である。

[図28]第1適用例を説明するための図である。

[図29]第1の適用例の変形例を説明するための図である。

[図30]第2の適用例に係る情報処理システムの概略構成例を示すブロック図である。

[図31]第3適用例に係る電子機器の一つである自律ロボットの概略構成例を示す模式図である。

[図32]第4適用例に係る電子機器の一つであるテレビジョンの概略構成例を示す模式図である。

[図33]本技術に係るコンピュータの構成例を示す図である。

[0010] 以下に、本開示の実施形態について図面に基づいて詳細に説明する。なお、以下の各実施形態において、同一の部位には同一の符号を付することにより重複する説明を省略する。

[0011] また、以下に示す項目順序に従って本開示を説明する。

0. 本技術について

1. 第1の実施形態

1. 1 ニューラルネットワーク構造の識別器による識別処理の具体例

1. 2 ニューラルネットワーク構造の予測器を用いた回帰処理の具体例

1. 3 フィルタ係数の圧縮符号化について
1. 4 データ圧縮による処理量削減について
1. 5 学習手法について
1. 6 ニューラルネットワーク処理装置の構成例
1. 7 識別処理の説明
1. 8 係数行列のスパース表現化による効果
1. 9 スパース表現係数の復号に好適な復号部
 1. 9. 1 復号部が実行する復号処理
 1. 9. 2 復号処理の具体例
1. 10 作用・効果
2. 第2の実施形態
 2. 1 スパース表現係数の復号に好適な復号部
 2. 1. 1 復号部が実行する復号処理
 2. 1. 2 復号処理の具体例
 2. 2 作用・効果
3. 第3の実施形態
 3. 1 ゼロ係数に対する積和演算の省略について
 3. 2 スパース表現係数の復号に好適な復号部
 3. 2. 1 復号部が実行する復号処理
 3. 2. 2 復号処理の具体例
 3. 3 作用・効果
4. 第4の実施形態
 4. 1 スパース表現係数の復号に好適な復号部
 4. 1. 1 復号部が実行する復号処理
 4. 1. 2 復号処理の具体例
 4. 2 作用・効果
5. 第5の実施形態
 5. 1 比較例

5. 2 符号化部

5. 3 作用・効果

6. 第6の実施形態

6. 1 符号化部

6. 2 動作例

6. 3 作用・効果

6. 4 変形例

6. 4. 1 第1の変形例

6. 4. 2 第2の変形例

6. 4. 3 第3の変形例

7. 本技術の適用例

7. 1 第1適用例

7. 2 第2適用例

7. 3 第3適用例

7. 4 第4適用例

8. コンピュータの構成例

[0012] 0. 本技術について

本技術は、ニューラルネットワークを利用した識別処理や回帰処理を実現する場合に、入力されたデータに対してフレーム単位で畳み込み処理等の演算処理を行なうときに、現フレームと過去のフレームとの演算処理の処理境界を一致させることでメモリ量を削減することができるようにするものである。

[0013] まず、図1および図2を参照して、ある時間区間のオーディオデータを入力とし、ニューラルネットワークを利用して、そのオーディオデータに対する識別処理を行う識別器について説明する。なお、図1および図2において、互いに対応する部分には同一の符号を付してあり、その説明は適宜省略する。

[0014] 図1は、入力されたオーディオデータに対する識別処理を行い、その識別

結果を出力するニューラルネットワーク構造の識別器の構成を示している。

[0015] 図1では、各四角形はデータの形状を表しており、特に各データにおける図中、縦方向は時間方向を示しており、図中、横方向は次元を示している。また図中、横方向の矢印はデータの変換を表している。ここで、各データのデータ形状は、データの次元数と各次元におけるデータ数（データサンプル数）により定まるものである。

[0016] 例えば図中、左側の端にある四角形の入カデータDT11は、ニューラルネットワーク構造の識別器に入力されるオーディオデータである。

[0017] ここでは入カデータDT11における図中、縦方向は時間方向を示しており、図中、横方向は次元を示している。特に、入カデータDT11における図中、下方向がより新しい時刻の方向（未来方向）となっている。

[0018] 入カデータDT11は、7910サンプル分の時間区間の1チャンネル分のオーディオデータとなっている。換言すれば、入カデータDT11は7910個のサンプル（時間サンプル）のサンプル値からなる1チャンネル分のオーディオデータである。

[0019] したがって、ここではオーディオデータのチャンネルが入カデータDT11の次元に対応しており、入カデータDT11の形状は1次元×7910サンプルとなる。また、入カデータDT11の図中、上側にあるサンプルほど過去のサンプルとなる。

[0020] 図1に示す識別器では、入カデータDT11を入力として5つの各レイヤ（階層）において演算処理としてデータ変換が行われ、図中、右側の端にある識別結果データDT16が出力として得られる。換言すれば、畳み込みレイヤ1から畳み込みレイヤ3までの間で合計5回のデータ変換を行うことで、入カデータDT11から識別結果データDT16が算出される。

[0021] この識別結果データDT16は、識別器による識別結果を示す1×1（1次元×1サンプル）のデータである。より詳細には識別結果データDT16は、例えば入カデータDT11に基づく音声が予め定められた特定の音声である確率を示すデータとなっている。

- [0022] 図1に示す識別器では、まず畳み込みレイヤ1において入力データDT11に対して4種類のフィルタごとにフィルタ処理が行われる。換言すれば、4種類のフィルタごとに、フィルタを構成するフィルタ係数と、入力データDT11との畳み込み処理が行われる。
- [0023] 畳み込みレイヤ1で行われる畳み込み処理で用いられる各フィルタは、 20×1 タップのフィルタとなっている。換言すれば、各フィルタのタップ数、つまりフィルタ係数の数は 20×1 となっている。また、各フィルタでの畳み込み処理では、10サンプル前進のフィルタ処理が行われる。
- [0024] このような畳み込みレイヤ1での畳み込み処理によって、入力データDT11は中間データDT12へと変換される。
- [0025] この中間データDT12は4次元 $\times$ 790サンプルのデータとなっている。すなわち、中間データDT12は図中、横方向に並ぶ4つの各次元のデータからなり、各次元のデータは790個のサンプル、つまり図中、縦方向に並ぶ790個のデータから構成される。
- [0026] 続いて、2番目のレイヤであるプーリングレイヤ1では、畳み込みレイヤ1で得られた中間データDT12に対して、10サンプル幅のデータ領域を対象とし、そのデータ領域から最大値をピック（抽出）する処理が10サンプル前進で行われる。
- [0027] 換言すれば、プーリングレイヤ1では、10サンプル幅のデータ領域を対象として最大値を抽出する10サンプル前進のプーリング処理が中間データDT12に対して行われる。
- [0028] このようなプーリング処理により、中間データDT12が中間データDT13へと変換される。ここでは、中間データDT13は4次元のデータとなっており、各次元のデータは79個のサンプル（データ）からなる。
- [0029] ここで、図2を参照して畳み込みレイヤ1における畳み込み処理と、プーリングレイヤ1におけるプーリング処理について説明する。
- [0030] 例えば図2の左側に示すように、畳み込みレイヤ1における1つのフィルタによる畳み込み処理では、入力データDT11におけるサンプルSP11

やサンプルS P 1 2、サンプルS P 1 3を含む20個のサンプルとフィルタ係数とが畳み込まれる。

[0031] すなわち、データ領域W 1 1内に含まれている、時間方向に連続して並ぶサンプルS P 1 1からサンプルS P 1 3までの20個の各サンプルと、フィルタを構成するフィルタ係数とを畳み込む畳み込み処理が行われる。そして、データ領域W 1 1を対象とした畳み込み処理により得られた値が、中間データD T 1 2の図中、左下のサンプルS P 2 1のサンプル値とされる。

[0032] データ領域W 1 1が1次元×20サンプルの領域とされるのは、畳み込みレイヤ1における畳み込み処理に用いられるフィルタが20×1タップのフィルタだからである。

[0033] また、畳み込みレイヤ1における畳み込み処理のサンプル前進数は10であるので、すなわち10サンプル前進の畳み込み処理であるので、データ領域W 1 1を対象とした畳み込み処理の次に行われる畳み込み処理では、データ領域W 1 2が処理対象の領域とされる。

[0034] このデータ領域W 1 2は、データ領域W 1 1の先頭にあるサンプルS P 1 1から見て、10サンプルだけ前にあるサンプルS P 1 2を先頭とする、連続して並ぶ20個のサンプルからなる領域である。すなわち、データ領域W 1 2は、データ領域W 1 1を過去方向に10サンプルだけずらした領域である。ここでは、データ領域W 1 2は、サンプルS P 1 2からサンプルS P 1 4までの領域となっている。

[0035] データ領域W 1 2を対象とした畳み込み処理では、データ領域W 1 1における場合と同様に、入力データD T 1 1におけるサンプルS P 1 2からサンプルS P 1 4までの20個の各サンプルとフィルタ係数とが畳み込まれる。そして、データ領域W 1 2を対象とした畳み込み処理により得られた値が、中間データD T 1 2の図中、左下から上方向に2番目のサンプルS P 2 2のサンプル値とされる。

[0036] 畳み込みレイヤ1では、4つのフィルタごとに、処理対象とするデータ領域を10サンプルずつずらしていく畳み込み処理が繰り返し行われて、中間

データDT12が生成される。中間データDT12における4つの次元のデータのそれぞれは、畳み込みレイヤ1における4つのフィルタによる畳み込み処理で得られたデータのそれぞれである。

[0037] また、プーリングレイヤ1では、1つ目の次元のサンプルSP21からサンプルSP23までの縦方向に連続して並ぶ10個のサンプルからなるデータ領域W21を処理対象として、最大値を抽出する処理が行われる。

[0038] すなわち、データ領域W21内にある10個のサンプルのサンプル値のうちの最大値が、中間データDT13の図中、左下のサンプルSP31のサンプル値とされる。

[0039] データ領域W21を対象として最大値を抽出する処理が行われると、次にデータ領域W21の先頭にあるサンプルSP21から見て、10サンプルだけ前にあるサンプルを先頭とする、連続して並ぶ10個のサンプルからなる領域が次のデータ領域とされる。すなわち、プーリングレイヤ1におけるプーリング処理のサンプル前進数は10であるので、処理対象とするデータ領域が10サンプル分だけずらされる。

[0040] そして、そのようにして得られた新たなデータ領域を対象として最大値を抽出する処理が行われ、以降においても同様の処理が行われていき、中間データDT12が中間データDT13へと変換される。

[0041] 図1の説明に戻り、プーリングレイヤ1におけるプーリング処理によって中間データDT13が得られると、3番目のレイヤである畳み込みレイヤ2では、8種類のフィルタが用いられて、畳み込みレイヤ1における場合と同様の畳み込み処理が行われる。

[0042] すなわち、畳み込みレイヤ2では、タップ数が 10×4 である、つまり 10×4 タップである互いに異なる8個のフィルタが用いられて、それらのフィルタごとに1サンプル前進の畳み込み処理（フィルタ処理）が行われる。

[0043] これにより、中間データDT13が8次元 $\times$ 70サンプルの中間データDT14へと変換される。すなわち、中間データDT14は8次元のデータとなっており、各次元のデータは70個のサンプル（データ）からなる。

- [0044] 4番目のレイヤであるプーリングレイヤ2では、10サンプル幅のデータ領域を対象として最大値を抽出する10サンプル前進のプーリング処理が中間データDT14に対して行われる。これにより、中間データDT14が8次元×7サンプルの中間データDT15へと変換される。
- [0045] さらに、5番目のレイヤである畳み込みレイヤ3では、1種類のフィルタが用いられて、畳み込みレイヤ1における場合と同様の畳み込み処理が行われる。
- [0046] すなわち、畳み込みレイヤ3では、7×8タップの1個のフィルタが用いられて、1サンプル前進の畳み込み処理（フィルタ処理）が行われる。
- [0047] これにより、中間データDT15が識別器の出力である1次元×1サンプルの識別結果データDT16に変換され、このようにして得られた識別結果データDT16が、入力データDT11に対する識別処理の結果（識別結果）として出力される。
- [0048] 以上のような畳み込みレイヤ1から畳み込みレイヤ3までの5つのレイヤでの演算処理が、ニューラルネットワーク構造の識別器により行われる。
- [0049] ところで、このようなニューラルネットワーク構造の識別器による各レイヤでは、直前のレイヤでの演算処理で得られたデータ全てを対象として、現レイヤでの演算処理を行う必要があるため、演算処理の処理量や演算処理に必要なとなるメモリのメモリ量が多くなってしまう。
- [0050] ここで、図1に示したニューラルネットワーク構造の識別器による処理に対してフレーム処理を適用することを考えるとする。
- [0051] フレーム処理とは、入力データに対してフレームを単位として処理を行うこと、つまり入力データのフレームの先頭または末尾の位置を処理の開始位置として処理を行うことをいう。フレーム処理は、主に演算処理のためのメモリ量を削減したり、例えば入力データ1024サンプルに対して1回の出力を行うなど、ニューラルネットワークの出力を算出する頻度を調整したりすることを目的として適用される。
- [0052] 例えば図3に示すように、図1における場合と同じニューラルネットワー

ク構成で、識別器により行われる処理に対してフレーム処理を適用するとする。すなわち、入力データDT11の1フレーム分である1024サンプルに対して、1回の頻度で識別結果データDT16を算出するようなフレーム処理を行うとする。なお、図3において図1における場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。

[0053] 図3において、入力データDT11における時間的に最も新しいフレームを現フレームとする。現フレームは、入力データDT11における図中、最も下側に連続して並ぶ1024個のサンプルからなる区間となっており、図3では入力データDT11における現フレームの部分には斜線（ハッチ）が施されている。

[0054] 入力データDT11の現フレームに対して1つの識別結果データDT16を算出しようとする、畳み込みレイヤ1ではデータ領域W41を処理対象とした畳み込み処理が行われることになる。

[0055] データ領域W41は、現フレームを構成する1024個のサンプルと、その現フレームの時間的に直前のフレーム（以下、直前フレームとも称する）における最後の6個のサンプルとからなる合計1030個のサンプルからなる区間となっている。

[0056] このとき、入力データDT11における現フレームでの処理対象以外のデータ（区間）に対する畳み込みは、過去のフレームにおいて行われるようにすることができれば、点線で描かれたデータ領域W42部分についての演算処理と、その演算処理のためのメモリ量を削減することができる。

[0057] しかしながら、図3に示すニューラルネットワークの構成では、そのような処理量とメモリ量の削減を行うことができない。これは現フレームの処理境界と、直前フレームの処理境界とが一致しないからである。

[0058] ここでフレームの処理境界とは、例えばそのフレームについての演算処理を開始すべきデータのサンプル位置、つまりフレームの先頭または末尾の位置や、そのフレームについての演算処理が終了した後、次の演算処理を開始すべきデータのサンプル位置である。

- [0059] 例えば現フレームについての処理境界は、現フレームの演算処理が終了した後、次の演算処理を開始すべき入力データDT11のサンプル位置である。このサンプル位置が現フレームと隣接する直前フレームについての演算処理が開始される入力データDT11のサンプル位置、つまり直前フレームの処理境界である、直前フレームの末尾（最後）のサンプル位置と一致しないと演算量やメモリ量を削減することができない。
- [0060] 具体的には、例えば入力データDT11のうち、図3中、最も下側にあるサンプルから n （但し $1 \sim n$ ）番目にあるサンプルを n 番目のサンプルとする。したがって、例えば入力データDT11の図中、最も下側にあるサンプル、つまり最も新しい時刻のサンプルが1番目のサンプルである。
- [0061] この場合、畳み込みレイヤ1では、現フレームにおいて、まず入力データDT11のうちの1番目のサンプルから20番目のサンプルに対して、 20×1 タップのフィルタによる畳み込み処理が行われる。
- [0062] また、畳み込みレイヤ1での前進数（サンプル前進数）は10サンプルであるので、次に、11番目のサンプルから30番目のサンプルに対して 20×1 タップのフィルタによる畳み込み処理が行われる。
- [0063] その後、同様の畳み込み処理が繰り返し行われ、現フレームでは、最終的には入力データDT11のうちの1011番目のサンプルから1030番目のサンプルに対して 20×1 タップのフィルタによる畳み込み処理が行われる。
- [0064] これまでの畳み込み処理の処理対象となったデータ領域が、入力データDT11におけるデータ領域W41であり、現フレームではこのデータ領域W41が処理対象の領域とされて畳み込み処理が行われる。
- [0065] また、1011番目のサンプルから1030番目のサンプルに対する畳み込み処理が行われると、その次は、1021番目のサンプルから1040番目のサンプルが畳み込み処理の対象となるので、畳み込み処理の現フレームの処理境界は1021番目のサンプルとなる。
- [0066] この場合、現フレームの直前にある直前フレームにおいては、1021番

目のサンプルから１０４０番目のサンプルに対して畳み込み処理が行われていなければならないはずである。

[0067] しかし、実際にはそのような畳み込み処理は行われず、フレーム処理を行った場合、つまりフレームを単位として処理を行った場合、直前フレームにおいては、現フレームのフレーム境界である１０２５番目のサンプルから畳み込み処理が行われることになる。

[0068] すなわち、直前フレームの処理境界は１０２５番目のサンプルであり、この１０２５番目のサンプルと、現フレームの処理境界である１０２１番目のサンプルとは一致していない。

[0069] そのため、上述した現フレームでの処理対象以外のデータ（区間）に対する畳み込み、つまりここでは１０２１番目のサンプルからの畳み込みが、直前フレームにおいて行われるようにすることができず、その結果、処理量とメモリ量の削減を行うことができない。

[0070] しかも図３の例では、フレーム処理を適用したために畳み込みレイヤ１での畳み込み処理は、図１を参照して説明した畳み込みレイヤ１での畳み込み処理とは異なる処理となってしまう。すなわち、図１を参照して説明した例とは異なるデータ領域に対する畳み込み処理が行われてしまう。

[0071] そうすると、結局、図１を参照して説明した例と同じ畳み込み処理が行われるようにするためには、１フレーム分である１０２４サンプルごとに、入力データＤＴ１１全てに対して図１を参照して説明した処理を行う必要があり、結果として多くの処理量とメモリ量が必要となる。

[0072] そこで以下の実施形態では、入力データに対してフレーム単位で演算処理を行うフレーム処理を適用し、ニューラルネットワークの構造をフレーム処理において現フレームと過去フレームの処理境界が一致する構造とすることで、少なくともメモリ量を削減できるようにした。

[0073] 具体的には、少なくとも１つのレイヤにおいて、そのレイヤでフレームに対して行われる演算処理の処理単位またはサンプル前進数が、入力データの形状により定まる大きさまたは前進数となっているニューラルネットワーク

構造の識別器が用いられる。

[0074] 換言すれば、以下の実施形態を適用したニューラルネットワーク構造の識別器では、互いに隣接するフレームの処理境界が一致するように、各レイヤにおける演算処理の処理単位やサンプル前進数、フレーム長（フレームのサンプル数）が定められている。

[0075] また、ニューラルネットワーク構造の識別器でフレーム処理が行われ、1フレームの大きさ（フレーム長）が予め定められている場合には、入力データの形状だけでなく1フレームの大きさ（フレーム長）も考慮されてレイヤで行われる演算処理の処理単位やサンプル前進数が定められるようにすればよい。

[0076] ここで、データの形状（データ形状）は、上述したようにデータの次元数と各次元におけるデータ数（データサンプル数）により定まる。

[0077] 例えば図1に示した入力データDT11は、1次元の7910サンプルからなるデータであるので、1次元×7910サンプルのデータ形状のデータとなる。また、例えば図1に示した中間データDT12は、4次元×790サンプルのデータ形状のデータとなる。

[0078] なお、レイヤで行われる演算処理の処理単位の大きさとは、そのレイヤにおいて1フレーム分の演算処理が行われるときの処理の最小単位である。すなわち、例えば1つのレイヤで処理対象とされるデータ領域がずらされながら同じ処理が複数回繰り返し行われるときのデータ領域の大きさ、つまりデータ領域内に含まれるサンプルのサンプル数が処理単位の大きさである。

[0079] 具体的には、例えば図1を参照して説明した畳み込みレイヤ1であれば、処理単位とは1度の畳み込み処理（フィルタ処理）が行われる20個のサンプルからなるデータ領域であり、このデータ領域はフィルタのタップ数により定まる。

[0080] また、例えば図1を参照して説明したプーリングレイヤ1であれば、処理単位とは最大値の抽出対象となる10個のサンプルからなるデータ領域であり、このデータ領域はサンプル幅により定まる。

[0081] このように入力データの形状やフレーム長に対して定まるタップ数や、サンプル幅（データ領域の大きさ）、サンプル前進数でレイヤの演算処理が行われるニューラルネットワーク構造（構成）の識別器を用いた識別処理が行われるようにすれば、互いに隣接するフレームにおける演算処理の処理境界を一致させることができる。これにより、少なくとも演算処理時に一時的に必要なメモリのメモリ量を削減することができ、より効率よく識別を行うことができる。

[0082] 以下の実施形態によれば、例えば数フレーム分の入力データDT11がある状態で、現フレームから時間的に過去方向にフレーム単位で順番に処理を行っていく場合でも、入力データDT11がフレーム単位で逐次入力されて、現フレームから時間的に未来方向にフレーム単位で順番に処理を行っていく場合でも、隣接するフレームにおける演算処理の処理境界を一致させることができる。また、フレームに対する演算処理の開始位置は、フレーム先頭のサンプルであってもよいし、フレーム末尾のサンプルであってもよい。

[0083] なお、ここでは入力データの形状やフレーム長に対してタップ数や、サンプル幅、サンプル前進数が定められる例について説明する。しかし、互いに隣接するフレームにおける演算処理の処理境界が一致するように、入力データの形状に対してデータのフレーム長、すなわちフレームを構成するサンプル数（データ数）が定められるようにしてもよい。そのような場合においても互いに隣接するフレームにおける演算処理の処理境界を一致させ、より効率よく識別を行うことができる。

[0084] 1. 第1の実施形態

以下、本技術の第1の実施形態について、図面を参照して詳細に説明する。

[0085] 1. 1 ニューラルネットワーク構造の識別器による識別処理の具体例

まず、本実施形態に係るニューラルネットワーク構造の識別器による識別処理の具体的な例について説明する。

[0086] 本実施形態に係るニューラルネットワーク構造の識別器では、例えば図4

に示すように各レイヤにおける処理が行われる。

[0087] すなわち、図4に示す例では、識別器による識別処理として5個の各レイヤの処理が順番に行われ、オーディオデータである1次元（1チャンネル）の入力データDT11'から、最終的な識別結果である識別結果データDT16'が算出される。

[0088] 特に、ここではフレームを単位として時間方向で過去のフレームから未来のフレームへと向かって順番に、各フレームについて5個の各レイヤの処理が順番に行われ、フレームごとに識別結果データDT16'が出力されるフレーム処理が行われる。この例では1フレームは1024個のサンプルにより構成されており、1つのフレームの処理では入力データDT11'が識別器の入力とされ、1つの識別結果データDT16'が出力される。

[0089] なお、図4において各四角形はデータを表している。また、各データにおける図中、縦方向は時間方向を示しており、図中、横方向は次元を示している。特に、各データにおける図中、下方向がより新しい時刻の方向（未来方向）となっている。

[0090] さらに、図中、横方向の矢印はデータの変換を表している。ここでは、1番目のレイヤは畳み込みレイヤ1'となっており、2番目のレイヤはプーリングレイヤ1となっており、3番目のレイヤは畳み込みレイヤ2となっており、4番目のレイヤはプーリングレイヤ2となっており、5番目のレイヤは畳み込みレイヤ3となっている。

[0091] これらの5個のレイヤのうち、2番目のプーリングレイヤ1から5番目の畳み込みレイヤ3までの各レイヤの処理は、図1を参照して説明した2番目のプーリングレイヤ1から5番目の畳み込みレイヤ3までの各レイヤの処理と同じであるので、その説明は適宜省略する。

[0092] この例では、入力データDT11'は、図1に示した7910サンプルからなる入力データDT11のうちの最も新しい時刻のサンプルである1番目のサンプルから1036番目のサンプルまでの部分のデータとなっている。

[0093] ここで、1フレームが1024サンプルからなるとすると、入力データD

T 1 1' における 1 番目のサンプルから 1 0 2 4 番目のサンプルまでの区間が 1 つのフレームとなり、以下ではこのフレームを現フレームとすることとする。

[0094] また、入力データ D T 1 1' における 1 0 2 5 番目のサンプルから 1 0 3 6 番目のサンプルまでの各サンプルは、現フレームの時間的に直前にある直前フレームのサンプルとなる。

[0095] 畳み込みレイヤ 1' では、このような現フレームのデータ（サンプル）と、直前フレームの一部のデータとからなる入力データ D T 1 1' が入力とされて畳み込み処理が行われ、入力データ D T 1 1' が中間データ D T 1 2' へと変換される。なお、より詳細には現フレームにおける畳み込みレイヤ 1' の畳み込み処理では、4 次元×7 9 0 サンプルの中間データ D T 1 2' のうちの一部のデータ P D T 1 のみが得られ、残りのデータ P D T 2 の部分は過去のフレームでの畳み込み処理で得られたものとなっている。

[0096] 畳み込みレイヤ 1' での畳み込み処理では、入力データ D T 1 1' に対してタップ数が 2 0 × 1 である、つまり 2 0 × 1 タップである 4 種類のフィルタが用いられて、それらのフィルタごとに 8 サンプル前進の畳み込み処理（フィルタ処理）が行われる。

[0097] 特に畳み込みレイヤ 1' では、フレームを単位とするフレーム処理が行われる。すなわち、畳み込みレイヤ 1' ではフレームごとに、4 種類の 2 0 × 1 タップのフィルタのフィルタ係数を用いた 8 サンプル前進の畳み込み処理（フィルタ処理）が、フレームに対する演算処理として行われる。

[0098] この場合、現フレームに対する処理では、現フレームの全てのサンプルが含まれるデータ領域が対象領域とされて畳み込み処理が行われる。そして、その畳み込み処理の時間的に前のタイミングでは、畳み込みレイヤ 1' の演算処理として、直前フレームの全てのサンプルが含まれるデータ領域が対象領域とされて畳み込み処理が行われている。

[0099] また、2 番目のレイヤであるプーリングレイヤ 1 では、1 0 サンプル幅のデータ領域を対象として最大値を抽出する 1 0 サンプル前進のプーリング処

理が中間データDT12'に対して行われる。これにより、中間データDT12'が4次元×79サンプルの中間データDT13'へと変換される。

[0100] 3番目のレイヤである畳み込みレイヤ2では、タップ数が10×4である互いに異なる8個のフィルタが用いられて、それらのフィルタごとに1サンプル前進の畳み込み処理（フィルタ処理）が行われる。これにより、中間データDT13'が8次元×70サンプルの中間データDT14'へと変換される。

[0101] さらに、4番目のレイヤであるプーリングレイヤ2では、10サンプル幅のデータ領域を対象として最大値を抽出する10サンプル前進のプーリング処理が中間データDT14'に対して行われる。これにより、中間データDT14'が8次元×7サンプルの中間データDT15'へと変換される。

[0102] 最後に5番目のレイヤである畳み込みレイヤ3では、タップ数が7×8である1個のフィルタが用いられて1サンプル前進の畳み込み処理が行われ、中間データDT15'が1次元×1サンプルの識別結果データDT16'へと変換される。このようにして得られた識別結果データDT16'が、入力データDT11'に対する識別処理の結果（識別結果）として出力される。

[0103] ここで、畳み込みレイヤ1'での畳み込み処理と、図1を参照して説明した畳み込みレイヤ1での畳み込み処理とを比較すると、畳み込みレイヤ1ではサンプル前進数が10であったのに対して、畳み込みレイヤ1'ではサンプル前進数が8とされている。

[0104] このようにサンプル前進数を8とすると、現フレームでは1番目のサンプルから20番目のサンプルまでのデータ領域が対象とされて畳み込み処理が行われ、続いて9番目のサンプルから28番目のサンプルまでのデータ領域が対象とされて畳み込み処理が行われる。

[0105] そして、それ以降においてもデータ領域が8サンプルずつずらされながら畳み込み処理が行われていき、最後に1017番目のサンプルから1036番目のサンプルまでのデータ領域が対象とされて畳み込み処理が行われる。

[0106] この場合、最後の畳み込み処理が行われるデータ領域の先頭は1017番

目のサンプルであるから、そのサンプルから時間方向に8サンプル前にある1025番目のサンプルは現フレームにおける処理境界となる。

[0107] ここでは1フレームは1024サンプルであり、この1025番目のサンプルは直前フレームの末尾（最後）のサンプルとなるから、1025番目のサンプルは直前フレームにおける処理境界となる。したがって、図4の例では互いに隣接する現フレームにおける処理境界と、直前フレームにおける処理境界とが一致することになる。特にこの例では、フレームの処理境界の位置はフレームの境界位置となる。これにより、現フレームでの処理対象以外のデータ（区間）に対する畳み込みは、過去のフレームにおいて行われるようになる。

[0108] より具体的には、この例ではフレーム処理が行われるので、入力データDT11の1024サンプルの入力に対して、1つの識別結果データDT16'が算出される。

[0109] この場合、現フレームにおいては入力データDT11のうちの入力データDT11'の部分が処理対象のデータ領域とされて畳み込みレイヤ1'の畳み込み処理が行われる。

[0110] そのため、現フレームでは、入力データDT11における点線で描かれたデータ領域W51の部分については畳み込み処理を行う必要がなく、その分だけ畳み込み処理の処理量と、その畳み込み処理のために入力データDT11のデータ領域W51部分のデータ等を保持しておくメモリのメモリ量とを削減することができる。

[0111] なお、この場合、現フレームにおける畳み込みレイヤ1'の畳み込み処理では、中間データDT12'における図4中、下側の4次元×128サンプルの部分のデータPDT1のみが得られ、中間データDT12'の残りの部分のデータPDT2は得られない。

[0112] しかし、データPDT2については、現フレームよりも時間的に過去の複数のフレームにおける畳み込みレイヤ1'の畳み込み処理で既に得られているので、それらの畳み込み処理で得られたデータPDT2を保持しておけば

よい。そうすれば現フレームにおける畳み込みレイヤ 1' の畳み込み処理が終了した時点で中間データ D T 1 2' を得ることができるため、その後、直ちにプーリングレイヤ 1 の処理を開始することができる。

[0113] また、図 4 では畳み込みレイヤ 1' のみ、隣接する各フレームでの処理境界が一致するようなニューラルネットワーク構造の識別器を用いる場合について説明した。しかし、畳み込みレイヤ 1' の後のレイヤにおいても隣接する各フレームでの処理境界が一致するようにしてもよい。

[0114] そのような場合、本実施形態に係るニューラルネットワーク構造の識別器では、例えば図 5 に示すように各レイヤにおける処理が行われる。なお、図 5 において図 4 における場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。

[0115] 図 5 に示す例では、識別器による識別処理として 5 個の各レイヤの処理が順番に行われ、オーディオデータである 1 次元（1 チャネル）の入力データ D T 1 1' から、最終的な識別結果である識別結果データ D T 2 5 が算出される。

[0116] 特に、ここではフレームを単位として時間方向で過去のフレームから未来のフレームへと向かって順番に、各フレームについて 5 個の各レイヤの処理が順番に行われ、フレームごとに識別結果データ D T 2 5 が出力されるフレーム処理が行われる。

[0117] なお、図 5 において各四角形はデータを表している。また、各データにおける図中、縦方向は時間方向を示しており、図中、横方向は次元を示している。特に、各データにおける図中、下方向がより新しい時刻の方向（未来方向）となっている。

[0118] さらに、図中、横方向の矢印はデータの変換を表している。ここでは、1 番目のレイヤは畳み込みレイヤ 1' となっており、2 番目のレイヤはプーリングレイヤ 1' となっており、3 番目のレイヤは畳み込みレイヤ 2' となっている。また、4 番目のレイヤはプーリングレイヤ 2' となっており、5 番目のレイヤは畳み込みレイヤ 3' となっている。

- [0119] これらの5個のレイヤのうち、1番目のレイヤである畳み込みレイヤ1'では、図4を参照して説明した畳み込みレイヤ1'における場合と全く同じ処理が行われるので、その説明は適宜省略する。
- [0120] 図5に示すニューラルネットワーク構造は、図1に示したニューラルネットワーク構造における各レイヤでのサンプル前進数やタップ数を微弱に変更することで、各レイヤにおいて隣接するフレームにおける処理境界が一致するようにしたものである。
- [0121] このように図5に示すニューラルネットワーク構造は、図1に示したニューラルネットワーク構造を微弱に変更しただけのものであるため、このようなニューラルネットワーク構造の変化が識別性能に与える影響は殆どない。
- [0122] この例では、図4を参照して説明した場合と同様に、畳み込みレイヤ1'では入力データDT11'を入力としてフレーム処理が行われる。
- [0123] すなわち、畳み込みレイヤ1'では現フレームの処理として、入力データDT11'に対して、タップ数が 20×1 である4種類のフィルタが用いられて、それらのフィルタごとに8サンプル前進の畳み込み処理が行われる。これにより、入力データDT11'が4次元 $\times 128$ サンプルの中間データDT21へと変換される。この中間データDT21は、図4に示したデータPDT1そのものである。
- [0124] このような畳み込みレイヤ1'での畳み込み処理では、図4における場合と同様に、フレーム処理を適用しない場合と比較してデータ領域W61の分だけ処理量とメモリ量を削減することができる。
- [0125] また、2番目のレイヤであるプーリングレイヤ1'では、8サンプル幅のデータ領域を対象として最大値を抽出する8サンプル前進のプーリング処理が中間データDT21に対して行われる。これにより、中間データDT21が4次元 $\times 16$ サンプルの中間データPDT21へと変換される。
- [0126] ここでは、プーリングレイヤ1'でのサンプル幅とサンプル前進数が、入力データDT11'のデータ形状やフレーム長、畳み込みレイヤ1'での処理、つまり畳み込みレイヤ1'の構成（構造）に対して定まるものとされて

いる。そのため、プーリングレイヤ1'では中間データD T 2 1における互いに隣接するフレームの処理境界を一致させることができ、その結果、プーリングレイヤ1'ではフレーム処理を適用しない場合と比較してデータ領域W 6 2の分だけ処理量とメモリ量を削減することができる。

[0127] 3番目のレイヤである畳み込みレイヤ2'では、タップ数が 10×4 である互いに異なる8個のフィルタが用いられて、それらのフィルタごとに2サンプル前進の畳み込み処理が行われる。

[0128] そのため、畳み込みレイヤ2'では、入力となるデータの形状、換言すれば処理対象となるデータ領域は、4次元 $\times$ 24サンプルとなるが、現フレームのプーリングレイヤ1'での処理で得られた中間データP D T 2 1は4次元 $\times$ 16サンプルのデータである。

[0129] しかし、現フレームの時間的に1つ前の直前フレームにおいてプーリングレイヤ1'で得られた中間データのうちの半分のデータである4次元 $\times$ 8サンプルの中間データP D T 2 2を保持しておけば、中間データP D T 2 1と中間データP D T 2 2から4次元 $\times$ 24サンプルの中間データD T 2 2を得ることができる。

[0130] 畳み込みレイヤ2'では、このような中間データD T 2 2に対して、タップ数が 10×4 である互いに異なる8個のフィルタが用いられて、それらのフィルタごとに2サンプル前進の畳み込み処理が行われる。これにより、中間データD T 2 2が8次元 $\times$ 8サンプルのへ中間データD T 2 3と変換される。

[0131] 畳み込みレイヤ2'でのタップ数とサンプル前進数は、入力データD T 1 1'のデータ形状やフレーム長、畳み込みレイヤ2'よりも前の各レイヤの構成（構造）に対して定まるものとされている。そのため、畳み込みレイヤ2'では中間データD T 2 2における隣接するフレームの処理境界を一致させることができ、その結果、畳み込みレイヤ2'ではフレーム処理を適用しない場合と比較してデータ領域W 6 3の分だけ処理量とメモリ量を削減することができる。

- [0132] 4番目のレイヤであるプーリングレイヤ2'では、8サンプル幅のデータ領域を対象として最大値を抽出する8サンプル前進のプーリング処理が中間データDT23に対して行われる。これにより、中間データDT23が8次元×1サンプルの中間データPDT31へと変換される。
- [0133] ここでは、プーリングレイヤ2'でのサンプル幅とサンプル前進数が、入力データDT11'のデータ形状やフレーム長、プーリングレイヤ2'よりも前の各レイヤの構成（構造）に対して定まるものとされている。そのため、プーリングレイヤ2'では、中間データDT23における互いに隣接するフレームの処理境界を一致させることができ、その結果、プーリングレイヤ2'ではフレーム処理を適用しない場合と比較してデータ領域W64の分だけ処理量とメモリ量を削減することができる。
- [0134] 5番目のレイヤである畳み込みレイヤ3'では、タップ数が8×8である1つのフィルタが用いられて、1サンプル前進の畳み込み処理が行われる。
- [0135] そのため、畳み込みレイヤ3'では、入力となるデータの形状、換言すれば処理対象となるデータ領域は、8次元×8サンプルとなるが、現フレームのプーリングレイヤ2'での処理で得られた中間データPDT31は8次元×1サンプルのデータである。
- [0136] しかし、現フレームの時間的に前の7つの過去フレームにおいてプーリングレイヤ2'で得られた中間データからなる8次元×7サンプルの中間データPDT32を保持しておけば、中間データPDT31と中間データPDT32から8次元×8サンプルの中間データDT24を得ることができる。
- [0137] 畳み込みレイヤ3'では、このような中間データDT24に対して、タップ数が8×8であるフィルタが用いられて、1サンプル前進の畳み込み処理が行われる。これにより、中間データDT24が1次元×1サンプルの識別結果データDT25へと変換される。このようにして得られた識別結果データDT25が、入力データDT11'に対する識別処理の結果（識別結果）として出力される。
- [0138] 畳み込みレイヤ3'でのタップ数とサンプル前進数は、入力データDT1

1' のデータ形状やフレーム長、畳み込みレイヤ 3' よりも前の各レイヤの構成（構造）に対して定まるものとされている。そのため、畳み込みレイヤ 3' では中間データ D T 2 4 における互いに隣接するフレームの処理境界を一致させることができる。

[0139] 以上のように図 5 を参照して説明したニューラルネットワーク構造の識別器を用いて識別処理を行えば、図 1 における場合と比較して、略同等の性能で処理量とメモリ量を約 6 分の 1 程度に削減することができる。

[0140] なお、図 5 では 1 0 2 4 サンプル（1 フレーム）に対して 1 つの識別結果データ D T 2 5 が出力される場合を例として説明したが、図 5 に示したニューラルネットワーク構造の識別器における識別結果データ D T 2 5 の出力頻度は、数フレームに 1 度などとしてもよい。この場合においてもデータ領域 W 6 1 乃至データ領域 W 6 4 の分だけ、処理量とメモリ量を削減することができる。

[0141] また、図 4 や図 5 では、説明を簡単にするため、比較的単純で規模の小さいニューラルネットワークを利用する場合を例として説明したが、ニューラルネットワークの規模、すなわちレイヤ数等はどのような規模であってもよい。本実施形態では、ニューラルネットワークが複雑で大規模になるほど処理量とメモリ量の削減率は高くなる。

[0142] さらに、ここではニューラルネットワークのレイヤで行われる処理として、畳み込み処理とプーリング処理を例として説明したが、その他、レジデュアル処理など、どのような演算処理が行われてもよい。レジデュアル処理は、現レイヤの入力とするデータにその現レイヤよりも前のレイヤの出力を加算して現レイヤの出力とするデータ変換処理である。

[0143] その他、本実施形態は、音声認識や話者識別、環境音識別等のオーディオデータの自動識別技術のみならず、ニューラルネットワークを用いた種々の技術に適用可能である。

[0144] 1. 2 ニューラルネットワーク構造の予測器を用いた回帰処理の具体例
例えば以上においてはニューラルネットワーク構造の識別器を用いた識別

処理に本実施形態を適用する例について説明したが、ニューラルネットワーク構造の予測器を用いた予測処理、すなわち回帰処理に本実施形態を適用することも可能である。

[0145] そのような場合、予測器の出力が回帰処理による予測結果を示す確率となるが、予測器の各レイヤでは、識別器の各レイヤにおける場合と同様の処理が行われる。

[0146] 具体的な例として、例えば本実施形態はオーディオ信号（オーディオデータ）の低域信号に基づき、オーディオ信号の高域信号をニューラルネットワークを用いて生成する帯域拡張技術に適用可能である。

[0147] 帯域拡張技術として、例えば国際公開第2015/79946号（以下、参照特許文献と称する）には、オーディオ信号から得られる複数の低域サブバンドの低域サブバンドパワーから、そのオーディオ信号の高域成分である高域信号を生成することが開示されている。

[0148] すなわち、参照特許文献では、オーディオ信号の低域サブバンドパワーが複数の低域サブバンドごとに求められ、それらの複数の低域サブバンドパワーに基づいて、オーディオ信号の高域側の複数の高域サブバンドの高域サブバンドパワーの推定値が疑似高域サブバンドパワーとして求められる。そして、それらの疑似高域サブバンドパワーに基づいて、高域信号が生成される。

[0149] ここで、参照特許文献では疑似高域サブバンドパワーの算出は、参照特許文献の式（2）に示されるように、複数の低域サブバンドパワーを説明変数とし、複数の高域サブバンドパワーを被説明変数とした、最小二乗法を用いた回帰分析により予め求められた係数が用いられて行われていた。

[0150] これに対して、本実施形態を疑似高域サブバンドパワーの算出に適用すれば、ニューラルネットワークを用いることで、より高精度に疑似高域サブバンドパワーを算出することが可能である。

[0151] そのような場合、ニューラルネットワーク構造の予測器の入力は、オーディオ信号の複数フレームのそれぞれにおける複数の低域サブバンドパワーな

どとされる。また、ニューラルネットワーク構造の予測器の出力は、1フレーム分のオーディオ信号における複数の高域サブバンドの高域サブバンドパワー（疑似高域サブバンドパワー）などとされる。

[0152] より具体的には、例えばニューラルネットワーク構造の予測器の入力は、16フレームの4つの低域サブバンドの低域サブバンドパワーなどとされ、また、予測器の出力は、1フレームの8つの高域サブバンドの高域サブバンドパワーなどとされる。

[0153] この場合、予測器の入力は16行4列の行列、つまり4次元×16サンプルの入力データとなり、入力データの各サンプルは1つの低域サブバンドにおける低域サブバンドパワーを示している。また、予測器の出力は1行8列の行列、つまり8次元×1サンプルの識別結果データとなり、識別結果データの各サンプルは1つの高域サブバンドにおける高域サブバンドパワーを示している。

[0154] このように、ニューラルネットワークを用いた帯域拡張技術においても、本実施形態を適用することが可能である。これと同様に、ニューラルネットワークを用いた画像や映像の認識技術や物体領域の抽出技術等にも本実施形態は適用可能である。

[0155] 1. 3 フィルタ係数の圧縮符号化について

ところで、ニューラルネットワーク構造の識別器の畳み込みレイヤの処理を行う場合、予め定められたフィルタのフィルタ係数が用いられるため、それらのフィルタ係数を保持しておくためのメモリが必要となる。

[0156] 具体的には、畳み込みレイヤでは、タップ数×フィルタ種類数分の数だけフィルタ係数を保持しておく必要がある。すなわち、畳み込みを行う1または複数のフィルタごとにタップ数分のフィルタ係数を保持しておく必要がある。

[0157] そのため、畳み込みレイヤの数や、畳み込みレイヤで用いるフィルタの数が増えると、その分だけフィルタ係数を保持しておくためのメモリ量も多くなってしまふ。

- [0158] そこで、本実施形態ではフィルタ係数を圧縮符号化することで、フィルタ係数を保持しておくためのメモリのメモリ量を削減するようにした。
- [0159] 本実施形態では、値が0であるフィルタ係数、つまりゼロ係数の位置を示すゼロ係数位置テーブルと、ゼロ係数以外のフィルタ係数の値を示す非ゼロ係数テーブルとの2つのテーブルを保持することでフィルタ係数のデータ量の圧縮が実現される。
- [0160] 具体的には、例えば図6の矢印Q 1 1に示すように、 4×4 タップのフィルタ係数からなる係数行列があるとする。この例では、矢印Q 1 1に示す係数行列における四角形がフィルタ係数を表しており、その四角形内の数値がフィルタ係数の値を示している。
- [0161] 本実施形態では、このような係数行列が圧縮符号化されて、矢印Q 1 2に示すゼロ係数位置テーブルと、矢印Q 1 3に示す非ゼロ係数テーブルとに変換される。
- [0162] 矢印Q 1 2に示すゼロ係数位置テーブルは、もとの係数行列のタップ数と同じ 4×4 のテーブルとなっており、ゼロ係数位置テーブルにおける四角形内の数値は、その四角形と同じ位置関係にある係数行列のフィルタ係数の値が0（ゼロ）であるか否かを示している。
- [0163] すなわち、ゼロ係数位置テーブルにおける四角形内の数値が0である場合、その四角形と同じ位置関係にある係数行列のフィルタ係数の値は0であることを示している。したがって、例えばゼロ係数位置テーブルの図中、左上の四角形内の値「0」は、同じ位置関係にある、係数行列の図中、左上のフィルタ係数の値が「0」であることを示している。
- [0164] これに対して、ゼロ係数位置テーブルにおける四角形内の数値が1である場合、その四角形と同じ位置関係にある係数行列のフィルタ係数の値は0でないことを示している。したがって、例えばゼロ係数位置テーブルの図中、最も左側の上から2番目の四角形内の値「1」は、同じ位置関係にある、係数行列の図中、最も左側の上から2番目のフィルタ係数の値が「0」でないことを示している。

- [0165] また、矢印Q 1 3 に示す非ゼロ係数テーブルは、もとの係数行列におけるゼロ係数ではないフィルタ係数の値を示すテーブルとなっており、非ゼロ係数テーブルにおける1つの四角形内の値は、係数行列におけるゼロ係数ではない1つのフィルタ係数の値を示している。
- [0166] この例では、非ゼロ係数テーブルにおいては、係数行列の函中、左上のフィルタ係数を始点としたラスタスキャン順に、ゼロ係数ではないフィルタ係数の値、つまりゼロ係数位置テーブルにおける値が「1」であるフィルタ係数の値が並べられている。
- [0167] 例えば非ゼロ係数テーブルにおける図中、左端の四角形内の数値「3」は、係数行列の最上段の左端から3番目にあるフィルタ係数の値が「3」であることを示している。
- [0168] このように係数行列そのものを保持するのではなく、係数行列を圧縮符号化して得られたゼロ係数位置テーブルと非ゼロ係数テーブルとを保持することで、フィルタ係数を得るための情報を保持しておくためのメモリのメモリ量を削減することができる。なお、以下の説明では、「係数行列（フィルタ係数）」を圧縮符号化してゼロ係数位置テーブルと非ゼロ係数テーブルとを生成することを「係数行列（フィルタ係数）をスパース表現する」とも称し、この「圧縮符号化により得られたゼロ係数位置テーブル及び非ゼロ係数テーブル」を「スパース表現された係数行列（フィルタ係数）」とも称する。また、「ゼロ係数位置テーブル」を「スパース行列」とも称し、「非ゼロ係数テーブル」を「非ゼロデータキュー」とも称する。
- [0169] 例えば図6に示す例では、1つのフィルタ係数が8ビットであるとする、係数行列は16個のフィルタ係数から構成されるので、係数行列のデータ量は128（ $=16 \times 8$ ）ビットとなる。
- [0170] これに対して、ゼロ係数位置テーブルにおいて、1つのフィルタ係数がゼロ係数であるか否かを示す値が1ビットであるとする、ゼロ係数位置テーブルのデータ量は16（ $=16 \times 1$ ）ビットとなる。
- [0171] また、非ゼロ係数テーブルにおいて、1つのフィルタ係数の値が8ビット

であるとする、ここではゼロ係数でないフィルタ係数の数は5個であるので、非ゼロ係数テーブルのデータ量は40 ($= 5 \times 8$) ビットとなる。

[0172] そうすると、ゼロ係数位置テーブルのデータ量と非ゼロ係数テーブルのデータ量の合計は56 ($= 40 + 16$) ビットとなるので、もとの係数行列のデータ量である128ビットと比べてデータ量、つまりメモリ量が約2分の1に圧縮されていることが分かる。フィルタ係数からなる係数行列の圧縮率は、ゼロ係数の数が多くなるほど高くなる。

[0173] 1. 4 データ圧縮による処理量削減について

さらに、本実施形態では係数行列を圧縮符号化することで、畳み込みレイヤでの処理を行う場合における処理量も削減することが可能である。

[0174] 例えば係数行列を用いた畳み込み処理では、その係数行列を用いた行列演算となる。そのため、フィルタ係数がゼロ係数であるか否かによらず、全てのフィルタ係数について、フィルタ係数と畳み込み対象のデータとの積（乗算処理）や、それらの積の和（加算処理）の演算が行われることになる。

[0175] 一方、係数行列を圧縮符号化した場合、まずはゼロ係数位置テーブルと非ゼロ係数テーブルに基づく復号処理が行われ、その結果得られたフィルタ係数が用いられて畳み込み処理が行われることになる。

[0176] このとき、復号処理ではゼロ係数位置テーブルの値が0でない位置についてのみ、非ゼロ係数テーブルからフィルタ係数をロードすればよいので、例えばハフマン符号化などの他の圧縮方式と比較すると復号時の処理量が少なく済む。

[0177] また、復号処理後の畳み込み処理においても、ロードしたフィルタ係数についてのみ畳み込み対象のデータとの積（乗算処理）と、それらの積の和（加算処理）の演算を行えばよいので、畳み込み処理自体の処理量を削減することができる。

[0178] 結果として、ゼロ係数が多ければ、復号処理と畳み込み処理の演算の合計の処理量は、係数行列をそのまま用いたときの畳み込み処理の処理量よりも少なくなり、ゼロ係数の数が多くなればなるほど、処理量の削減率は高くな

る。

[0179] 1. 5 学習手法について

なお、係数行列におけるゼロ係数の数は、ニューラルネットワークの係数の学習において、例えばL1正則化やL2正則化、それらに準ずる正則化手法により増やすことが可能である。したがって、ニューラルネットワーク構造の識別器の学習時において、正則化によりゼロ係数の数を適切に調整しておけば、フィルタ係数を圧縮符号化することにより、処理量およびメモリ量を削減することができる。

[0180] なお、ニューラルネットワークの係数の学習時における正則化については、例えば「Ian Goodfellow, Yoshua Bengio, and Aaron Courville、Deep Learning、The MIT Press、2016」などに詳細に記載されている。

[0181] 1. 6 ニューラルネットワーク処理装置の構成例

次に、以上において説明した本実施形態に係るニューラルネットワーク処理装置について説明する。

[0182] 図7は、本実施形態に係るニューラルネットワーク処理装置の一実施の形態の構成例を示す図である。

[0183] 図7に示すニューラルネットワーク処理装置11は、入力データの形状やフレーム長に対して定まるタップ数や、サンプル幅、サンプル前進数でレイヤの演算処理が行われるニューラルネットワーク構造の識別器からなる情報処理装置である。

[0184] ニューラルネットワーク処理装置11は、供給された入力データに基づいてニューラルネットワークの各レイヤの演算処理を行うことで、入力データを対象とする識別処理を行い、その識別結果を示す識別結果データを出力する。

[0185] なお、ここではニューラルネットワーク処理装置11が図5を参照して説明したニューラルネットワーク構造の識別器である例について説明する。

[0186] ニューラルネットワーク処理装置11は、畳み込み処理部21、プーリン

グ処理部 2 2、畳み込み処理部 2 3、プーリング処理部 2 4、および畳み込み処理部 2 5 を有している。

[0187] この場合、畳み込み処理部 2 1 乃至畳み込み処理部 2 5 によって、図 5 を参照して説明したニューラルネットワーク構造の識別器が構成されることになる。換言すれば、畳み込み処理部 2 1 乃至畳み込み処理部 2 5 によりニューラルネットワークが構成されている。

[0188] 畳み込み処理部 2 1 は、供給された入力データに対して畳み込みレイヤ 1' の処理を行い、その結果得られた中間データをプーリング処理部 2 2 に供給する。

[0189] 畳み込み処理部 2 1 は、復号部 4 1、メモリ 4 2、および係数保持部 4 3 を有している。

[0190] 例えば係数保持部 4 3 は不揮発性のメモリからなる。この係数保持部 4 3 には、予め学習により得られたタップ数が 20×1 である互いに異なる 4 つの種類のフィルタについて、フィルタのフィルタ係数を圧縮符号化して得られたゼロ係数位置テーブルと非ゼロ係数テーブルが保持されている。

[0191] 復号部 4 1 は、各フィルタのゼロ係数位置テーブルと非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行い、各フィルタのゼロ係数でないフィルタ係数を得る。メモリ 4 2 は、例えば揮発性のメモリからなり、供給された入力データや演算中のデータを一時的に保持する。

[0192] プーリング処理部 2 2 は、畳み込み処理部 2 1 から供給された中間データに対してプーリングレイヤ 1' の処理を行い、その結果得られた中間データを畳み込み処理部 2 3 に供給する。プーリング処理部 2 2 はメモリ 5 1 を有している。メモリ 5 1 は、例えば揮発性のメモリからなり、畳み込み処理部 2 1 から供給された中間データや演算中のデータを一時的に保持する。

[0193] 畳み込み処理部 2 3 は、プーリング処理部 2 2 から供給された中間データに対して畳み込みレイヤ 2' の処理を行い、その結果得られた中間データをプーリング処理部 2 4 に供給する。

[0194] 畳み込み処理部 2 3 は、復号部 6 1、メモリ 6 2、および係数保持部 6 3

を有している。

[0195] 例えば係数保持部63は不揮発性のメモリからなる。この係数保持部63には、予め学習により得られたタップ数が 10×4 である互いに異なる8つの種類のフィルタについて、フィルタのフィルタ係数を圧縮符号化して得られたゼロ係数位置テーブルと非ゼロ係数テーブルが保持されている。

[0196] 復号部61は、各フィルタのゼロ係数位置テーブルと非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行い、各フィルタのゼロ係数でないフィルタ係数を得る。メモリ62は、例えば揮発性のメモリからなり、プーリング処理部22から供給された中間データや演算中のデータを一時的に保持する。

[0197] プーリング処理部24は、畳み込み処理部23から供給された中間データに対してプーリングレイヤ2'の処理を行い、その結果得られた中間データを畳み込み処理部25に供給する。プーリング処理部24はメモリ71を有している。メモリ71は、例えば揮発性のメモリからなり、畳み込み処理部23から供給された中間データや演算中のデータを一時的に保持する。

[0198] 畳み込み処理部25は、プーリング処理部24から供給された中間データに対して畳み込みレイヤ3'の処理を行い、その結果得られた識別結果データを出力する。

[0199] 畳み込み処理部25は、復号部81、メモリ82、および係数保持部83を有している。

[0200] 例えば係数保持部83は不揮発性のメモリからなる。この係数保持部83には、予め学習により得られたタップ数が 8×8 であるフィルタのフィルタ係数を圧縮符号化して得られたゼロ係数位置テーブルと非ゼロ係数テーブルが保持されている。

[0201] 復号部81は、フィルタのゼロ係数位置テーブルと非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行い、フィルタのゼロ係数でないフィルタ係数を得る。メモリ82は、例えば揮発性のメモリからなり、プーリング処理部24から供給された中間データや演算中のデータを一時的に保持する。

。

[0202] なお、ここではニューラルネットワーク処理装置 11 が、図 5 を参照して説明したニューラルネットワーク構造の識別器により構成される例について説明した。しかし、ニューラルネットワーク処理装置 11 が、図 4 を参照して説明したニューラルネットワーク構造の識別器により構成されるようにしてもよい。

[0203] そのような場合、畳み込み処理部 21 により畳み込みレイヤ 1' の処理が行われ、プーリング処理部 22 によりプーリングレイヤ 1 の処理が行われ、畳み込み処理部 23 により畳み込みレイヤ 2 の処理が行われ、プーリング処理部 24 によりプーリングレイヤ 2 の処理が行われ、畳み込み処理部 25 により畳み込みレイヤ 3 の処理が行われる。

[0204] 1. 7 識別処理の説明

次に、ニューラルネットワーク処理装置 11 の動作について説明する。すなわち、以下、図 8 のフローチャートを参照して、ニューラルネットワーク処理装置 11 による識別処理について説明する。なお、この識別処理は入力データのフレームごとに行われる。

[0205] ステップ S 11 において、畳み込み処理部 21 の復号部 41 は、係数保持部 43 からゼロ係数位置テーブルと非ゼロ係数テーブルを読み出して、それらのゼロ係数位置テーブルおよび非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行う。また、畳み込み処理部 21 のメモリ 42 は、供給された入力データを一時的に記録する。

[0206] ステップ S 12 において、畳み込み処理部 21 は、ステップ S 11 の処理により得られた 4 つのフィルタのフィルタ係数に基づいて、メモリ 42 に記録されている入力データに対して、フィルタごとに 8 サンプル前進の畳み込み処理を行う。例えばステップ S 12 では、ゼロ係数ではないフィルタ係数のみが用いられて畳み込み処理が行われる。

[0207] 畳み込み処理部 21 は、畳み込み処理により得られた中間データを、プーリング処理部 22 のメモリ 51 に供給し、一時的に記録させる。

- [0208] このような畳み込み処理により、図5のデータ領域W61の分だけ、畳み込み処理の処理量とメモリ42のメモリ量を削減することができる。また、ゼロ係数の分だけ演算処理の処理量をさらに削減することができる。
- [0209] ステップS13において、プーリング処理部22は、メモリ51に記録されている中間データに対して、8サンプル幅のデータ領域を対象として最大値を抽出する8サンプル前進のプーリング処理を行う。プーリング処理部22は、プーリング処理により得られた中間データを畳み込み処理部23のメモリ62に供給し、一時的に記録させる。
- [0210] このようなプーリング処理により、図5のデータ領域W62の分だけ、プーリング処理の処理量とメモリ51のメモリ量を削減することができる。
- [0211] ステップS14において、畳み込み処理部23の復号部61は、係数保持部63からゼロ係数位置テーブルと非ゼロ係数テーブルを読み出して、それらのゼロ係数位置テーブルおよび非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行う。
- [0212] ステップS15において、畳み込み処理部23は、ステップS14の処理により得られた8つのフィルタのフィルタ係数に基づいて、メモリ62に記録されている中間データに対して、フィルタごとに2サンプル前進の畳み込み処理を行う。例えばステップS15では、ゼロ係数ではないフィルタ係数のみが用いられて畳み込み処理が行われる。
- [0213] 畳み込み処理部23は、畳み込み処理により得られた中間データを、プーリング処理部24のメモリ71に供給し、一時的に記録させる。
- [0214] このような畳み込み処理により、図5のデータ領域W63の分だけ、畳み込み処理の処理量とメモリ62のメモリ量を削減することができる。また、ゼロ係数の分だけ演算処理の処理量をさらに削減することができる。
- [0215] ステップS16において、プーリング処理部24は、メモリ71に記録されている中間データに対して、8サンプル幅のデータ領域を対象として最大値を抽出する8サンプル前進のプーリング処理を行う。プーリング処理部24は、プーリング処理により得られた中間データを畳み込み処理部25のメ

メモリ 82 に供給し、一時的に記録させる。

[0216] このようなプーリング処理により、図 5 のデータ領域 W 6 4 の分だけ、プーリング処理の処理量とメモリ 71 のメモリ量を削減することができる。

[0217] ステップ S 17 において、畳み込み処理部 25 の復号部 81 は、係数保持部 83 からゼロ係数位置テーブルと非ゼロ係数テーブルを読み出して、それらのゼロ係数位置テーブルおよび非ゼロ係数テーブルに基づいてフィルタ係数の復号処理を行う。

[0218] ステップ S 18 において、畳み込み処理部 25 は、ステップ S 17 の処理により得られた 1 つのフィルタのフィルタ係数に基づいて、メモリ 82 に記録されている中間データに対して 1 サンプル前進の畳み込み処理を行う。例えばステップ S 18 では、ゼロ係数ではないフィルタ係数のみが用いられて畳み込み処理が行われる。このような畳み込み処理により、ゼロ係数の分だけ演算処理の処理量を削減することができる。

[0219] 畳み込み処理部 25 は、畳み込み処理により得られた識別結果データを出し、識別処理は終了する。

[0220] 以上のようにしてニューラルネットワーク処理装置 11 は、入力データの形状やフレーム長に対して定まるタップ数や、サンプル幅、サンプル前進数で各レイヤの演算処理を行い、識別結果データを生成する。このようにすることで、処理量とメモリ量を削減し、より効率よく識別を行うことができる。

[0221] また、ニューラルネットワーク処理装置 11 は、予め保持しているゼロ係数位置テーブルおよび非ゼロ係数テーブルに基づいて復号処理を行い、その結果得られたフィルタ係数を用いて畳み込み処理を行う。このようにすることで、フィルタ係数を保持しておくメモリのメモリ量を削減するとともに、ゼロ係数部分の積和演算を省略して畳み込み処理の処理量も削減することができる。

[0222] なお、ニューラルネットワーク処理装置 11 が、上述した帯域拡張のための高域サブバンドパワー（疑似高域サブバンドパワー）を予測する予測器と

されるようにしてもよい。

[0223] そのような場合、畳み込み処理部 21 には入力データとして低域サブバンドパワーが入力され、畳み込み処理部 25 は、畳み込み処理の処理結果（演算結果）を、高域サブバンドパワーの予測結果として出力する。但し、この場合、各レイヤでは図 5 を参照して説明した処理とは異なる処理が行われる。

[0224] 1. 8 係数行列のスパース表現化による効果

上述では、ニューラルネットワークの学習過程においてゼロを多く含むような係数行列を発生させる学習手法（1. 5 学習手法について）及びその圧縮符号化方法（1. 3 フィルタ係数の圧縮符号化について）、並びに、データ圧縮による処理量の削減手法（1. 4 データ圧縮による処理量削減について）について説明した。

[0225] ここで、本実施形態では、係数行列の圧縮符号化は、係数行列をゼロ係数位置テーブルと非ゼロ係数テーブルとに分離して記憶すること、すなわち、係数行列をスパース表現することによって実現していた。この手法を実際のニューラルネットワークを利用した推論課題に適用した場合、係数行列にゼロ係数を多く含むネットワークが形成され、その推論処理の際には係数行列のメモリ量が $1/4$ 以下に削減されることを確認することができる。

[0226] このような結果は、精度や速度を維持したままメモリ量を削減できることや、メモリ量の増加を抑制しつつニューラルネットワークのサイズを数倍に拡大できることを示唆している。例えば、外音認識などリアルタイム性が問われるネットワークにおいて、これまではニューラルネットワークのサイズを制限しないと処理のリアルタイム性を確保できなかったものに対して本実施形態を適用することで、メモリ量の増加を抑制しつつ、ニューラルネットワークのサイズを 2 倍～4 倍に拡大することが可能となる。また、メモリ量の決まった組込みシステムに対して本実施形態を適用することで、システムの物理的な構成を変えことなくニューラルネットワークによる推論タスクを追加で実装することも可能になるため、より複雑な認識処理が可能となる。

ようにシステムをアップデートすることが可能となる。

[0227] このように、ニューラルネットワークの学習に正則化手法を取り入れることで、全体としてゼロ係数を多く発生する学習モデルを生成することができる。そして、そのように発生させた係数行列をゼロ係数位置テーブルと非ゼロ係数テーブルとに分けて記録するスパース表現手法を用いることで、係数行列を圧縮してメモリ量を削減することが可能になる。

[0228] 1. 9 スパース表現係数の復号に好適な復号部

スパース表現された係数行列は、ソフトウェアで伸長することも可能であるが、ハードウェアで伸長することも可能である。以下の説明では、ハードウェア構成として実装された復号部について、具体例を挙げて説明する。

[0229] 1. 9. 1 復号部が実行する復号処理

図9は、本実施形態に係る復号部が実行する動作、すなわちスパース表現係数を復号する際の処理の一例を示すフローチャートである。図9では、明確化のため、復号部が実行する復号処理をフローチャート化して説明するが、実際には、図9に係る復号アルゴリズムが図7に例示したニューラルネットワーク処理装置11の復号部41、61及び81にハードウェア構成として実装されてよい。なお、以下の説明では、簡略化のため、復号部41、61及び81を区別しない場合、その符号を“101”とし、メモリ42、62及び82を区別しない場合、その符号を“102”とし、係数保持部43、63及び83を区別しない場合、その符号を“103”とする。

[0230] 図9に示すように、復号部101は、まず、係数保持部103からゼロ係数位置テーブル(SpMat)Q12及び非ゼロ係数テーブルQ13を読み出す(LOAD SpMat)(ステップS101)。読み出されたゼロ係数位置テーブルQ12及び非ゼロ係数テーブルQ13は、例えば、復号部101のメモリ102における既知のアドレスに格納される。

[0231] 次に、復号部101は、ゼロ係数位置テーブルQ12内の係数Wの位置を示す位相ポインタpを初期化する($p \leftarrow 0$)(ステップS102)。

[0232] また、復号部101は、非ゼロ係数テーブルQ13内の位置を示す非ゼロ

係数キューポインタ $wadr$ に、メモリ 102 内の非ゼロ係数テーブル Q13 において先頭に位置する係数 W (非ゼロ係数) を指し示すアドレス (以下、先頭アドレスという) をセットする ($wadr \leftarrow init$) (ステップ S103)。

[0233] 次に、復号部 101 は、メモリ 102 内のゼロ係数位置テーブル Q12 に対して位相ポインタ p が指し示す値が “0” であるか “1” であるかを判定する ($SpMat[p]$) (ステップ S104)。なお、“0” は例えば請求の範囲における第 1 値に相当し得、“1” は例えば請求の範囲における第 2 値に相当し得る。

[0234] 位相ポインタ p が指し示す値が “1” である場合 (ステップ S104 の “1”)、復号部 101 は、メモリ 102 内の非ゼロ係数テーブル Q13 から非ゼロ係数キューポインタ $wadr$ が指し示す非ゼロ係数 $WQue[wadr]$ を読み出し ($LOAD\ WQue[wadr]$) (ステップ S105)、読み出された非ゼロ係数 $WQue[wadr]$ を係数バッファ ($Wbuf$) 114 における所定の位置に格納する ($PUSH\ WQue\ to\ Wbuf$) (ステップ S106)。なお、係数バッファ 114 とは、復号過程において係数行列 Q11 が順次組み立てられるバッファであり、その所定の位置とは、係数行列 Q11 において位相ポインタ p が指し示す位置に相当する位置であってよい。

[0235] その後、復号部 101 は、非ゼロ係数キューポインタ $wadr$ をインクリメント、すなわち、非ゼロ係数キューポインタ $wadr$ に次の非ゼロ係数が格納されているアドレスをセットし ($wadr++$) (ステップ S107)、ステップ S109 へ進む。

[0236] 一方、ステップ S104 において、位相ポインタ p が指し示す値が “0” である場合 (ステップ S104 の “0”)、復号部 101 は、ゼロの値を係数バッファ 114 における所定の位置に格納し ($PUSH\ Zero\ to\ Wbuf$) (ステップ S108)、ステップ S109 へ進む。

[0237] ステップ S109 では、復号部 101 は、位相ポインタ p を 1 インクリメ

ント、すなわち、位相ポインタ p に次の係数 W の位置をセットする ($p++$)。

[0238] その後、復号部 101 は、インクリメント後の位相ポインタ p がゼロ係数位置テーブル $Q12$ の末尾を示す位相ポインタ p の最大値 p_{max} を超えているか否かを判定し ($p > p_{max} ?$) (ステップ $S110$)、超えている場合 (ステップ $S110$ の YES)、本動作を終了する。一方、超えていない場合 (ステップ $S110$ の NO)、復号部 101 は、ステップ $S104$ へリターンし、以降の動作を実行する。

[0239] 以上のように動作することで、係数バッファ 114 内には、係数行列 $Q11$ が復元される。

[0240] 1. 9. 2 復号処理の具体例

続いて、本実施形態に係る復号部 101 が実行する復号処理の具体例を、図 10 及び図 11 を参照しつつ、図 9 に例示したフローチャートに沿って説明する。図 10 は、本実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。図 11 は、図 10 に示す具体例における位相ポインタ及び非ゼロ係数キューポインタの動きを説明するための図である。なお、図 10 及び図 11 には、係数行列 $Q11$ が 1 行 8 列の構成をしており、その 3 列目と 5 列目と 8 列目とに非ゼロ係数が格納されている場合が例示されている。また、3 列目の非ゼロ係数は 0.1 であり、5 列目の非ゼロ係数は -0.8 であり、8 列目の非ゼロ係数は 0.6 であるとする。

[0241] 図 10 に示すように、復号部 101 は、非ゼロ係数キューポインタ $wadr$ を生成及び管理するアドレス生成部 111 と、位相ポインタ p が指し示す値に応じて“非ゼロ係数 $WQueue[wadr]$ ”及び“0”のいずれかを出力するセクタ 112 と、セクタ 112 から出力された値を順次、係数バッファ 114 に格納することで、係数バッファ 114 内に係数行列 $Q11$ を復元する積和器 113 と、復元された係数行列 $Q11$ を格納する係数バッファ 114 とを備える。

[0242] 図 10 に示す構成において、復号部 101 は、まず、ゼロ係数位置テーブ

ルQ12 (= “00101001”) 及び非ゼロ係数テーブルQ13 (= “0.1”、“-0.8”、“0.6”) を係数保持部103から取得する(図9のステップS101に相当)。また、復号部101は、最初の繰返し($itr=0$)では、位相ポインタpを0に初期化するとともに、アドレス生成部111において管理される非ゼロ係数キューポインタwadrに先頭アドレスをセットする(図9のステップS102及びS103に相当)。それにより、図11に示すように、位相ポインタpは、ゼロ係数位置テーブルQ12における一番左(先頭)の“0”を指し示し、非ゼロ係数キューポインタwadrは非ゼロ係数テーブルQ13における先頭の値“0.1”を指し示すことになる。

[0243] 次に、復号部101は、現在、位相ポインタpが指し示す値が“0”であるか“1”であるかを判定する(図9のステップS104に相当)。上述したように、 $itr=0$ では、位相ポインタpは“0”を指し示している。そのため、セクタ112の制御端子には、“0”が入力されている。その場合(図9のステップS104の“0”に相当)、復号部101が非ゼロ係数テーブルQ13から値を取り出すことなく、セクタ112から値“0”が積和器113に入力される。これに対し、積和器113は、入力された値“0”を、先頭の係数Wとして係数バッファ114に格納する(図9のステップS108に相当)。

[0244] その後、復号部101は、位相ポインタpを1インクリメントし(図9のステップS109に相当)、次の繰返し($itr=1$)へ移行する(図9のステップS110のNOに相当)。

[0245] 図11に示すように、 $itr=1$ では、 $itr=0$ と同様、位相ポインタpが指し示す値が“0”である(図9のステップS104の“0”に相当)。そのため、復号部101が非ゼロ係数テーブルQ13から値を取り出すことなく、セクタ112から値“0”が積和器113に入力され、この値“0”が2番目の係数Wとして係数バッファ114に格納される(図9のステップS108に相当)。そして、位相ポインタpが1インクリメントされ(

図9のステップS109に相当)、次の繰返し($itr=2$)が実行される(図9のステップS110のNOに相当)。

[0246] 図11に示すように、 $itr=2$ では、位相ポインタ p が指し示す値が“1”である。そのため、セクタ112の制御端子には、“1”が入力されている。その場合(図9のステップS104の“1”に相当)、復号部101は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル $Q13$ を参照し(図9のステップS105の一部、図10のステップS105aに相当)、非ゼロ係数キューポインタ $wadr$ が指し示す値(非ゼロ係数) $WQue[wadr]$ ($=0, 1$)を非ゼロ係数テーブル $Q13$ から取り出して、セクタ112に入力する(図9のステップS105の一部、図10のステップS105bに相当)。これに対し、セクタ112は、制御端子への入力値が“1”であるため、入力された非ゼロ係数 $WQue[wadr]$ ($=0, 1$)を積和器113に入力する。そして、積和器113は、入力された非ゼロ係数 $WQue[wadr]$ ($=0, 1$)を、3番目の係数 W として係数バッファ114に格納する(図9のステップS106に相当)。

[0247] その後、アドレス生成部111において次の非ゼロ係数 W を指し示すアドレスに非ゼロ係数キューポインタ $wadr$ がインクリメントされるとともに(図9のステップS107に相当)、位相ポインタ p が1インクリメントされ(図9のステップS109に相当)、次の繰返し($itr=3$)が実行される(図9のステップS110のNOに相当)。

[0248] 図11に示すように、 $itr=3$ では、 $itr=0$ と同様、位相ポインタ p が指し示す値が“0”である(図9のステップS104の“0”に相当)。そのため、復号部101が非ゼロ係数テーブル $Q13$ から値を取り出すことなく、セクタ112から値“0”が積和器113に入力され、この値“0”が4番目の係数 W として係数バッファ114に格納される(図9のステップS108に相当)。そして、位相ポインタ p が1インクリメントされ(図9のステップS109に相当)、次の繰返し($itr=4$)が実行される

(図9のステップS110のNOに相当)。

[0249] 図11に示すように、 $itr=4$ では、位相ポインタ p が指し示す値が“1”である(図9のステップS104の“1”に相当)。そのため、復号部101は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブルQ13を参照し(図9のステップS105の一部、図10のステップS105aに相当)、非ゼロ係数キューポインタ $wadr$ が指し示す非ゼロ係数 $WQue[wadr]$ ($=-0.8$)を取り出してセレクタ112に入力する(図9のステップS105の一部、図10のステップS105bに相当)。そして、取り出された非ゼロ係数 $WQue[wadr]$ ($=-0.8$)がセレクタ112を介して積和器113に入力され、5番目の係数 W として係数バッファ114に格納される(図9のステップS106に相当)。その後、非ゼロ係数キューポインタ $wadr$ 及び位相ポインタ p が1インクリメントされ(図9のステップS107及びS109に相当)、次の繰返し($itr=5, 6$)が実行される(図9のステップS110のNOに相当)。

[0250] 図11に示すように、 $itr=5, 6$ では、 $itr=0$ と同様、位相ポインタ p が指し示す値が“0”である(図9のステップS104の“0”に相当)。そのため、 $itr=5, 6$ それぞれでは、復号部101が非ゼロ係数テーブルQ13から値を取り出すことなく、セレクタ112から値“0”が積和器113に入力され、この値“0”が6番目及び7番目の係数 W として係数バッファ114に格納される(図9のステップS108に相当)。そして、位相ポインタ p が1インクリメントされ(図9のステップS109に相当)、次の繰返し($itr=7$)が実行される(図9のステップS110のNOに相当)。

[0251] 図11に示すように、 $itr=7$ では、位相ポインタ p が指し示す値が“1”である(図9のステップS104の“1”に相当)。そのため、復号部101は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブルQ13を参照し(図9のステップS105の一部、図10のステッ

プS 1 0 5 aに相当)、非ゼロ係数キューポインタw a d rが指し示す非ゼロ係数W Q u e [w a d r] (= 0. 6)を取り出してセクタ1 1 2に入力する(図9のステップS 1 0 5の一部、図10のステップS 1 0 5 bに相当)。そして、取り出された非ゼロ係数W Q u e [w a d r] (= 0. 6)がセクタ1 1 2を介して積和器1 1 3に入力され、8番目の係数Wとして係数バッファ1 1 4に格納される(図9のステップS 1 0 6に相当)。

[0252] その後、非ゼロ係数キューポインタw a d r及び位相ポインタpが1インクリメントされるが(図9のステップS 1 0 7及びS 1 0 9に相当)、インクリメント後の位相ポインタpは最大値p m a xを超えることとなる(図9のステップS 1 1 0のY E S)、そのため、復号部1 0 1は、次の繰返しへ移行することなく、本動作を終了する。

[0253] 以上のように動作することで、処理完了後の係数バッファ1 1 4内には、“0”、“0”、“0. 1”、“0”、“-0. 8”、“0”、“0”、“0. 6”から構成された係数行列Q 1 1が復元される。なお、係数バッファ1 1 4に復元された係数行列Q 1 1は、係数バッファ1 1 4から適宜読み出され、復号部1 0 1が実行する畳み込み処理に使用されてよい。

[0254] 以上のような復号アルゴリズムは、上述したように、復号部1 0 1にハードウェア実装され得る。位相ポインタpが1インクリメントされるまでの処理に要するクロック数を1クロック(C L K)とした順序回路を構成した場合、ゼロ係数位置テーブルQ 1 2の読み込みに1 C L K、デコード値“0”、“0”、“0. 1”、“0”、“-0. 8”、“0”、“0”、“0. 6”を得るのに8 C L K、それぞれ要するため、合計9 C L Kで係数行列Q 1 1をデコードすることができる。

[0255] 1. 1 0 作用・効果

以上のように、本実施形態によれば、係数行列Q 1 1がスパース表現により圧縮符号化されてメモリ1 0 2に格納されるため、必要なメモリ量を削減することが可能となる。また、このようなメモリ量の削減を可能にする復号部1 0 1をハードウェアにより実現することが可能となる。

[0256] 2. 第2の実施形態

上述した第1の実施形態に係る畳み込み処理では、演算対象である変数Xと係数Wとの対応関係を保ちつつ、畳み込み演算を行う必要がある。したがって、復号部101をハードウェア構成にて実現する場合、畳み込み対象の変数行列における変数Xの位置と、係数行列Q11における係数Wの位置とを揃えて、積和演算を行う積和回路に入力する必要がある。

[0257] ここで、第1の実施形態では、係数Wがスパース表現された状態でメモリ102（及び係数保持部103）に格納されているのに対し、変数Xは通常表現、すなわち圧縮されていない状態でメモリ102に格納されている。そのため、変数Xと係数Wとの同期を考慮しつつ、係数行列Q11の復号と変数Xの読み出しとを行う必要がある。そこで本実施形態では、変数Xと係数Wとの同期を可能にする復号部について、例を挙げて説明する。

[0258] なお、本実施形態に係るニューラルネットワーク処理装置は、第1の実施形態において図7等を用いて説明したニューラルネットワーク処理装置11と同様の構成であってよいため、ここでは詳細な説明を省略する。また、以下の説明において、畳み込み対象の変数行列Q14は、畳み込み処理に用いられる係数行列Q11と同じデータ長及びデータ構造を備えているものとする。

[0259] 2. 1 スパース表現係数の復号に好適な復号部

変数Xとの同期を図りつつ係数行列を復号する復号部は、ソフトウェアとして実装することも可能である、ハードウェアとして実装することも可能である。以下の説明では、ハードウェア構成として実装された復号部について、具体例を挙げて説明する。

[0260] 2. 1. 1 復号部が実行する復号処理

図12は、本実施形態に係る復号部が実行する動作、すなわちスパース表現係数を復号する際の処理の一例を示すフローチャートである。図12では、明確化のため、復号部が実行する復号処理をフローチャート化して説明するが、実際には、図12に係る復号アルゴリズムが図7に例示したニューラ

ルネットワーク処理装置 11 の復号部 41、61 及び 81 にハードウェア構成として実装されてよい。また、図 12 において、図 9 に例示した動作と同様の動作については、それを引用することで、その詳細な説明を省略する。

[0261] 図 12 に示すように、復号部 101 は、まず、図 9 のステップ S101～S103 と動作を実行することで、係数保持部 103 からゼロ係数位置テーブル (SpMat) Q12 及び非ゼロ係数テーブル Q13 を読み出すとともに、位相ポインタ p を初期化し、非ゼロ係数キューポインタ wadr に先頭アドレスをセットする。

[0262] また、本実施形態では、復号部 101 は、処理対象である変数 X が配列する変数行列 Q14 内の位置を示す変数キューポインタ xadr に、メモリ 102 内の変数行列 Q14 において先頭に位置する変数 X を指し示す先頭アドレスをセットする ($xadr \leftarrow init$) (ステップ S201)。

[0263] 次に、復号部 101 は、図 9 のステップ S104 と同様に、メモリ 102 内のゼロ係数位置テーブル Q12 に対して位相ポインタ p が指し示す値が “0” であるか “1” であるかを判定し、“1” であれば、図 9 のステップ S105～S107 と同様に、メモリ 102 内の非ゼロ係数テーブル Q13 から非ゼロ係数 WQue[wadr] を読み出して係数バッファ Wbuf における所定の位置に格納するとともに、非ゼロ係数キューポインタ wadr をインクリメントし、“0” であれば、図 9 のステップ S108 と同様に、ゼロの値を係数バッファ Wbuf における所定の位置に格納する。

[0264] 次に、復号部 101 は、メモリ 102 内の変数行列 Q14 から変数キューポインタ xadr が指し示す変数 XQue[xadr] を読み出し (LOAD XQue[xadr]) (ステップ S202)、読み出された変数 XQue[xadr] を変数バッファ (Xbuf) 115 における所定の位置に格納する (PUSH XQue to Xbuf) (ステップ S203)。なお、変数バッファ 115 とは、復号過程において変数行列 Q14 が係数行列 Q11 の復元に同期して読み出されるバッファであり、その所定の位置とは、変数行列 Q14 において変数キューポインタ xadr が指し示す位置に

相当する位置であってよい。

[0265] その後、復号部101は、変数キューポインタ $xaddr$ をインクリメント、すなわち、変数キューポインタ $xaddr$ に次の変数 X が格納されているアドレスをセットし($xaddr++$) (ステップS204)、ステップS110へ進む。

[0266] ステップS110では、復号部101は、図9のステップS110と同様に、インクリメント後の位相ポインタ p がゼロ係数位置テーブルQ12の末尾を示す位相ポインタ p の最大値 $pmax$ を超えているか否かを判定し、超えている場合(ステップS110のYES)、本動作を終了する。一方、超えていない場合(ステップS110のNO)、復号部101は、ステップS104へリターンし、以降の動作を実行する。

[0267] 以上のように動作することで、係数バッファ114内には係数行列Q11が復元され、変数バッファ115内には処理対象の変数行列Q14が読み出される。

[0268] 2. 1. 2 復号処理の具体例

続いて、本実施形態に係る復号部101が実行する復号処理の具体例を、図13及び図11を用いつつ、図12に例示したフローチャートに沿って説明する。図13は、本実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。なお、図13には、図10及び図11と同様に、係数行列Q11が1行8列の構成をしており、その3列目と5列目と8列目とに非ゼロ係数が格納されている場合が例示されている。また、3列目の非ゼロ係数は0.1であり、5列目の非ゼロ係数は-0.8であり、8列目の非ゼロ係数は0.6であるとする。さらに、図13では、変数行列Q14が係数 $X_0 \sim$ 係数 X_7 の8つの係数 X が1行8列に配列する構成をしているものとする。

[0269] 図13に示すように、復号部101は、非ゼロ係数キューポインタ $waddr$ を生成及び管理するアドレス生成部111aと、変数キューポインタ $xaddr$ を生成及び管理するアドレス生成部111bと、位相ポインタ p が指し

示す値に応じて“非ゼロ係数WQue [wadr]”及び“0”のいずれかを出力するセクタ112と、セクタ112から出力された値を順次、係数バッファ114に格納することで、係数バッファ114内に係数行列Q11を復元する積和器113と、復元された係数行列Q11を格納する係数バッファ114と、メモリ102から読み出された変数Xを順次格納することで変数行列Q14を保持する変数バッファ115とを備える。

[0270] 図13に示す構成において、復号部101は、まず、ゼロ係数位置テーブルQ12(=“00101001”)及び非ゼロ係数テーブルQ13(=“0.1”、“-0.8”、“0.6”)を係数保持部103から取得する(図12のステップS101に相当)。また、復号部101は、最初の繰返し(i tr=0)では、位相ポインタpを0に初期化するとともに、アドレス生成部111aにおいて管理される非ゼロ係数キューポインタwadrに先頭アドレスをセットする(図12のステップS102及びS103に相当)。それにより、図11に示すように、位相ポインタpは、ゼロ係数位置テーブルQ12における一番左(先頭)の“0”を指し示し、非ゼロ係数キューポインタwadrは非ゼロ係数テーブルQ13における先頭の値“0.1”を指し示すことになる。

[0271] また、最初の繰返し(i tr=0)では、復号部101は、アドレス生成部111bにおいて管理される変数キューポインタxadrに先頭アドレスをセットする(図12のステップS201に相当)。それにより、変数キューポインタxadrは変数行列Q14における先頭の値X₀を指し示すことになる。

[0272] 次に、復号部101は、現在、位相ポインタpが指し示す値が“0”であるか“1”であるかを判定する(図12のステップS104に相当)。上述したように、i tr=0では、位相ポインタpは“0”を指し示している。そのため、セクタ112の制御端子には、“0”が入力されている。その場合(図12のステップS104の“0”に相当)、復号部101が非ゼロ係数テーブルQ13から値を取り出すことなく、セクタ112から値“0

”が積和器113に入力される。これに対し、積和器113は、入力された値“0”を、先頭の係数Wとして係数バッファ114に格納する（図12のステップS108に相当）。

[0273] また、復号部101は、現在の変数キューポインタ $xadr$ に基づいてメモリ102内の変数行列 $Q14$ を参照し（図12のステップS202の一部、図13のステップS202aに相当）、変数キューポインタ $xadr$ が指し示す値 $X_{Que}[xadr]$ （ $=X_0$ ）を変数行列 $Q14$ から取り出して、変数バッファ115に格納する（図12のステップS203に相当）。

[0274] その後、復号部101は、位相ポインタ p を1インクリメントし（図12のステップS109に相当）、次の繰返し（ $itr=1$ ）へ移行する（図12のステップS110のNOに相当）。

[0275] 図11に示すように、 $itr=1$ では、 $itr=0$ と同様、位相ポインタ p が指し示す値が“0”である（図12のステップS104の“0”に相当）。そのため、復号部101が非ゼロ係数テーブル $Q13$ から値を取り出すことなく、セレクタ112から値“0”が積和器113に入力され、この値“0”が2番目の係数Wとして係数バッファ114に格納される（図12のステップS108に相当）。また、現在の変数キューポインタ $xadr$ に基づいてメモリ102内の変数行列 $Q14$ が参照され、これにより取得された変数 $X_{Que}[xadr]$ （ $=X_1$ ）が変数バッファ115に格納される（図12のステップS202～S203に相当）。そして、位相ポインタ p が1インクリメントされるとともに（図12のステップS109に相当）、アドレス生成部111bにおいて次の変数 X_2 を指し示すアドレスに変数キューポインタ $xadr$ がインクリメントされ（図12のステップS204に相当）、次の繰返し（ $itr=2$ ）が実行される（図12のステップS110のNOに相当）。

[0276] 図11に示すように、 $itr=2$ では、位相ポインタ p が指し示す値が“1”である。そのため、セレクタ112の制御端子には、“1”が入力されている。その場合（図12のステップS104の“1”に相当）、復号部1

01は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル $Q13$ を参照し（図12のステップ $S105$ の一部、図13のステップ $S105a$ に相当）、非ゼロ係数キューポインタ $wadr$ が指し示す値（非ゼロ係数） $WQue[wadr]$ （ $=0, 1$ ）を非ゼロ係数テーブル $Q13$ から取り出して、セクタ112に入力する（図12のステップ $S105$ の一部、図13のステップ $S105b$ に相当）。これに対し、セクタ112は、制御端子への入力値が“1”であるため、入力された非ゼロ係数 $WQue[wadr]$ （ $=0, 1$ ）を積和器113に入力する。そして、積和器113は、入力された非ゼロ係数 $WQue[wadr]$ （ $=0, 1$ ）を、3番目の係数 W として係数バッファ114に格納する（図12のステップ $S106$ に相当）。また、現在の変数キューポインタ $xadr$ に基づいてメモリ102内の変数行列 $Q14$ が参照され、これにより取得された変数 $XQue[xadr]$ （ $=X_2$ ）が変数バッファ115に格納される（図12のステップ $S202 \sim S203$ に相当）。

[0277] その後、アドレス生成部111aにおいて次の非ゼロ係数 W を指し示すアドレスに非ゼロ係数キューポインタ $wadr$ がインクリメントされ（図12のステップ $S107$ に相当）、位相ポインタ p が1インクリメントされるとともに（図12のステップ $S109$ に相当）、アドレス生成部111bにおいて次の変数 X_3 を指し示すアドレスに変数キューポインタ $xadr$ がインクリメントされ（図12のステップ $S204$ に相当）、次の繰返し（ $itr=3$ ）が実行される（図12のステップ $S110$ のNOに相当）。

[0278] 図11に示すように、 $itr=3$ では、 $itr=0$ と同様、位相ポインタ p が指し示す値が“0”である（図12のステップ $S104$ の“0”に相当）。そのため、復号部101が非ゼロ係数テーブル $Q13$ から値を取り出すことなく、セクタ112から値“0”が積和器113に入力され、この値“0”が4番目の係数 W として係数バッファ114に格納される（図12のステップ $S108$ に相当）。また、現在の変数キューポインタ $xadr$ に基づいてメモリ102内の変数行列 $Q14$ が参照され、これにより取得された

変数XQue[xadr](=X₃)が変数バッファ115に格納される(図12のステップS202~S203に相当)。そして、位相ポインタpが1インクリメントされるとともに(図12のステップS109に相当)、アドレス生成部111bにおいて次の変数X₄を指し示すアドレスに変数キューポインタxadrがインクリメントされ(図12のステップS204に相当)、次の繰返し(itr=4)が実行される(図12のステップS110のNOに相当)。

[0279] 図11に示すように、itr=4では、位相ポインタpが指し示す値が“1”である(図12のステップS104の“1”に相当)。そのため、復号部101は、現在の非ゼロ係数キューポインタwadrに基づいて非ゼロ係数テーブルQ13を参照し(図12のステップS105の一部、図13のステップS105aに相当)、非ゼロ係数キューポインタwadrが指し示す非ゼロ係数WQue[wadr](=-0.8)を取り出してセクタ112に入力する(図12のステップS105の一部、図12のステップS105bに相当)。そして、取り出された非ゼロ係数WQue[wadr](=-0.8)がセクタ112を介して積和器113に入力され、5番目の係数Wとして係数バッファ114に格納される(図12のステップS106に相当)。また、現在の変数キューポインタxadrに基づいてメモリ102内の変数行列Q14が参照され、これにより取得された変数XQue[xadr](=X₄)が変数バッファ115に格納される(図12のステップS202~S203に相当)。その後、非ゼロ係数キューポインタwadr、位相ポインタp及び変数キューポインタxadrがインクリメントされ(図12のステップS107、S109及びS204に相当)、次の繰返し(itr=5、6)が実行される(図9のステップS110のNOに相当)。

[0280] 図11に示すように、itr=5、6では、itr=0と同様、位相ポインタpが指し示す値が“0”である(図12のステップS104の“0”に相当)。そのため、itr=5、6それぞれでは、復号部101が非ゼロ係数テーブルQ13から値を取り出すことなく、セクタ112から値“0”

が積和器 113 に入力され、この値 “0” が 6 番目及び 7 番目の係数 W として係数バッファ 114 に格納される（図 12 のステップ S108 に相当）。また、現在の変数キューポインタ $xadr$ に基づいてメモリ 102 内の変数行列 Q14 が参照され、これにより取得された変数 $X_{Que}[xadr]$ ($=X_5, X_6$) が変数バッファ 115 に格納される（図 12 のステップ S202 ~ S203 に相当）。そして、位相ポインタ p が 1 インクリメントされるとともに（図 12 のステップ S109 に相当）、アドレス生成部 111b において次の変数 X_7 を指し示すアドレスに変数キューポインタ $xadr$ がインクリメントされ（図 12 のステップ S204 に相当）、次の繰返し ($itr = 7$) が実行される（図 12 のステップ S110 の NO に相当）。

[0281] 図 11 に示すように、 $itr = 7$ では、位相ポインタ p が指し示す値が “1” である（図 12 のステップ S104 の “1” に相当）。そのため、復号部 101 は、現在の変数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル Q13 を参照し（図 12 のステップ S105 の一部、図 13 のステップ S105a に相当）、非ゼロ係数キューポインタ $wadr$ が指し示す非ゼロ係数 $W_{Que}[wadr]$ ($=0.6$) を取り出してセクタ 112 に入力する（図 12 のステップ S105 の一部、図 13 のステップ S105b に相当）。そして、取り出された非ゼロ係数 $W_{Que}[wadr]$ ($=0.6$) がセクタ 112 を介して積和器 113 に入力され、8 番目の係数 W として係数バッファ 114 に格納される（図 12 のステップ S106 に相当）。また、現在の変数キューポインタ $xadr$ に基づいてメモリ 102 内の変数行列 Q14 が参照され、これにより取得された変数 $X_{Que}[xadr]$ ($=X_7$) が変数バッファ 115 に格納される（図 12 のステップ S202 ~ S203 に相当）。

[0282] その後、非ゼロ係数キューポインタ $wadr$ 、位相ポインタ p 及び変数キューポインタ $xadr$ がインクリメントされるが（図 12 のステップ S107、S109 及び S204 に相当）、インクリメント後の位相ポインタ p は最大値 p_{max} を超えることとなる（図 12 のステップ S110 の YES）

。そのため、復号部101は、次の繰返しへ移行することなく、本動作を終了する。

[0283] 以上のように動作することで、処理完了後の係数バッファ114内には、“0”、“0”、“0.1”、“0”、“-0.8”、“0”、“0”、“0.6”から構成された係数行列Q11が復元され、変数バッファ115内には、“ X_0 ”、“ X_1 ”、“ X_2 ”、“ X_3 ”、“ X_4 ”、“ X_5 ”、“ X_6 ”、“ X_7 ”から構成された変数行列Q14が配置される。なお、係数バッファ114に復元された係数行列Q11及び変数バッファ115内に配置された変数行列Q14は、それぞれの先頭から順に順次読み出されて積和回路116に入力されることで、積和演算による畳み込み処理が実行される。

[0284] 以上のような復号アルゴリズムは、復号部101にハードウェア実装され得る。位相ポインタpが1インクリメントされるまでの処理に要するクロック数を1クロック（CLK）とした順序回路を構成した場合、ゼロ係数位置テーブルQ12の読み込みに1CLK、デコード値“0”、“0”、“0.1”、“0”、“-0.8”、“0”、“0”、“0.6”を得るのに8CLK、それぞれ要するため、合計9CLKで係数行列Q11をデコードすることができる。

[0285] 2.2 作用・効果

このように、位相ポインタpを1インクリメントするタイミングで変数キューポインタxadrもインクリメントすることで、変数キューポインタxadrが指し示す変数行列Q14上の位置と、非ゼロ係数キューポインタwadrが指し示す係数行列Q11上の位置とを同期させることが可能となる。それにより、係数バッファ114内の係数W及び変数バッファ115内の変数Xの配列を、演算順序通りの配列とすることが可能となるため、積和回路116への係数W及び変数Xの入力を容易化することが可能となる。その際、係数バッファ114及び変数バッファ115をFIFO（First-In First-Out）として構成することで、係数バッファ114及び変数バッファ115への全ての係数W及び変数Xの格納が完了する前に、積和回路116への

係数W及び変数Xの順次入力が可能となるため、係数行列Q 1 1の復号開始から畳み込み処理の完了までに要する時間をより短縮することが可能となる。

[0286] その他の構成、動作及び効果は、上述した実施形態と同様であってよいため、ここでは詳細な説明を省略する。

[0287] 3. 第3の実施形態

上述した実施形態に係る畳み込み処理では、係数Wがゼロ係数であるか非ゼロ係数であるかを問わず、全ての係数Wに対して変数Xが積和演算された。ただし、ゼロ係数に関しては、いかなる値の変数Xを乗算しても、その結果はゼロである。そこで本実施形態では、ゼロ係数については演算処理を省略することで、全体としてのスループットを向上して、係数行列Q 1 1の復号開始から畳み込み処理の完了までに要する時間をより短縮することを可能する場合について、例を挙げて説明する。

[0288] なお、本実施形態に係るニューラルネットワーク処理装置は、第1の実施形態において図7等を用いて説明したニューラルネットワーク処理装置11と同様の構成であってよいため、ここでは詳細な説明を省略する。また、以下の説明において、畳み込み対象の変数行列Q 1 4は、畳み込み処理に用いられる係数行列Q 1 1と同じデータ長及びデータ構造を備えているものとする。

[0289] 3. 1 ゼロ係数に対する積和演算の省略について

積和演算においてゼロ係数に対する演算を省略して非ゼロ係数に対する演算を実行するための構成としては、例えば、プライオリティエンコーダを用いることができる。図14は、本実施形態に係るプライオリティエンコーダを説明するための図である。

[0290] 一般的なプライオリティエンコーダは、“1”になった入力に対応する値を出力する回路である。本実施形態では、複数の入力が“1”になった場合でも1つの値を出力するように、プライオリティエンコーダが構成されている必要がある。そこで本実施形態では、プライオリティエンコーダ104の

複数の入力に対して優先順位が設定される。

[0291] 図14に示すように、本実施形態において、プライオリティエンコーダ104におけるアクティブな入力の数は、ゼロ係数位置テーブルQ12を構成する“0”及び“1”の数、すなわち係数行列Q11を構成する係数Wの数（図14では8つ）と同数である。各入力へは、ゼロ係数位置テーブルQ12における配列通りに“0”又は“1”の値が入力される。そこで本実施形態では、ゼロ係数位置テーブルQ12における値の配列に従って、プライオリティエンコーダ104の各入力に優先順位を設定する。

[0292] 図14に示す例では、プライオリティエンコーダ104が“a0”～“a7”の8つの入力を備え、ゼロ係数位置テーブルQ12における先頭の値が入力される入力“a0”に対して最も高い優先順位が設定され、末尾の値が入力される入力“a7”に対して最も低い優先順位が設定され、“a”に付属する数字が大きくなるにつれて低くなる優先順位が設定される。なお、本説明では、各入力“a0”～“a7”には、それぞれの“a”に付属する数字に対応する値qが設定されているものとする。

[0293] プライオリティエンコーダ104におけるイネーブル端子enは、入力“a0”～“a7”の少なくとも1つに“1”が入力されたか否かを出力する端子であってよい。例えば、入力“a0”～“a7”への入力が全て“0”である場合、プライオリティエンコーダ104は、例えば“1”のイネーブル信号を出力してもよい。その場合、積和回路116からは、積和演算結果として“0”が出力されてよい。一方、入力“a0”～“a7”の少なくとも1つに“1”が入力された場合、プライオリティエンコーダ104は、例えば“0”のイネーブル信号を出力してもよい。その場合、積和回路116からは、係数バッファ114に格納された係数Wと変数バッファ115に格納された変数Xとを積和演算することで得られた値が出力されてよい。

[0294] このようなプライオリティエンコーダ104に対し、例えば、“00101001”のゼロ係数位置テーブルQ12が入力されると、プライオリティエンコーダ104の8つの入力“a0”～“a7”のうち、入力“a0”、

“a 1”、“a 3”、“a 5”、“a 6”には“0”が入力され、入力“a 2”、“s 5”、“a 8”には“1”が入力される。その場合、入力“a 2”、“s 5”、“a 8”のうちで最も優先順位の高い入力が入力“a 2”であるため、プライオリティエンコーダ104の出力oからは、入力“a 2”に設定された値 $q = 2$ が出力される。

[0295] プライオリティエンコーダ104から出力された値 q は、例えば、変数行列 Q_{14} から読み出す変数 X の位置を示す位相ポインタ q の設定に使用される。例えば、プライオリティエンコーダ104から出力された値が“2”である場合には、位相ポインタ q が変数行列 Q_{14} における先頭から数えて3番目の変数 X_2 を指すように、位相ポインタ q に“2”が設定される。同様に、プライオリティエンコーダ104から出力された値が“4”である場合には、位相ポインタ q が変数行列 Q_{14} における先頭から数えて5番目の変数 X_4 を指すように、位相ポインタ q に“4”が設定される。

[0296] なお、プライオリティエンコーダ104の出力に応じて位相ポインタ q の値を設定した後は、出力oとして使用した入力に対応するゼロ係数位置テーブル Q_{12} 上の値が、例えば復号部101によって“1”から“0”に書き換えられる。例えば、 $itr = 0$ の場合に“00101001”のゼロ係数位置テーブル Q_{12} が入力され、プライオリティエンコーダ104から入力“a 2”に設定された“2”が出力されて位相ポインタ q が設定された場合には、ゼロ係数位置テーブル Q_{12} が復号部101によって“00001001”に書き換えられる。それにより、プライオリティエンコーダ104の入力“a 0”～“a 7”に入力される値が“00001001”となり、次にプライオリティエンコーダ104から出力される値が入力“a 4”に設定された“4”となる。その結果、 $itr = 1$ では、位相ポインタ q が変数行列 Q_{14} における先頭から数えて5番目の変数 X_4 を指すように、位相ポインタ q が更新される。

[0297] 3. 2 スパース表現係数の復号に好適な復号部

本実施形態に係る復号部101も、上述した実施形態に係る復号部101

と同様に、ソフトウェアとして実装することも可能である、ハードウェアとして実装することも可能である。以下の説明では、ハードウェア構成として実装された復号部について、具体例を挙げて説明する。

[0298] 3. 2. 1 復号部が実行する復号処理

図15は、本実施形態に係る復号部が実行する動作、すなわちスプース表現係数を復号する際の処理の一例を示すフローチャートである。図15では、明確化のため、復号部が実行する復号処理をフローチャート化して説明するが、実際には、図15に係る復号アルゴリズムが図7に例示したニューラルネットワーク処理装置11の復号部41、61及び81にハードウェア構成として実装されてよい。また、図15において、図9又は図12に例示した動作と同様の動作については、それを引用することで、その詳細な説明を省略する。

[0299] 図15に示すように、復号部101は、まず、図9のステップS101及びS103と同様に、係数保持部103からゼロ係数位置テーブル(SpMat) Q12及び非ゼロ係数テーブルQ13を読み出すとともに、非ゼロ係数キューポインタwadrに先頭アドレスをセットする。また、復号部101は、図12のステップS201と同様に、変数キューポインタxadrに先頭アドレスをセットする。

[0300] 次に、復号部101によって読み出されたゼロ係数位置テーブルQ12をプライオリティエンコーダ104に入力して評価することで、“1”が入力された入力のうちで最も優先順位が高い入力（すなわち、ゼロ係数位置テーブルQ12における最初の“1”に対応する入力）を特定し(P.E. SpMat)、特定された入力に設定された値を位相ポインタqに設定する($q \leftarrow 1st \ non \ zero$) (ステップS301)。

[0301] 次に、復号部101は、図9のステップS105～S106と同様に、メモリ102内の非ゼロ係数テーブルQ13から非ゼロ係数WQue[wadr]を読み出して係数バッファWbufにおける所定の位置に格納する。

[0302] 次に、復号部101は、メモリ102内の変数行列Q14における、変数

キューポインタ $xadr$ （初期アドレス）から位相ポインタ q 分だけシフトさせたアドレス（ $xadr + q$ ）に格納された変数 $XQue[xadr + q]$ を読み出し（LOAD $XQue[xadr + q]$ ）（ステップ S302）、読み出された変数 $XQue[xadr + q]$ を変数バッファ（ $Xbuf$ ）115 における所定の位置に格納する（PUSH $XQue$ to $Xbuf$ ）（ステップ S303）。

[0303] 次に、復号部 101 は、図 9 のステップ S107 と同様に、非ゼロ係数キューポインタ $wadr$ をインクリメントする。

[0304] 次に、復号部 101 は、ゼロ係数位置テーブル Q12 における、ステップ S301 において特定された入力に対応する値を“0”に書き換える（SpMat[q]）（ステップ S304）。

[0305] その後、復号部 101 は、ステップ S304 で更新後のゼロ係数位置テーブル Q12 の値が全て“0”であるか否かを判定し（SpMat == 0?）（ステップ S305）、全て“0”である場合（ステップ S305 の YES）、本動作を終了する。一方、ゼロ係数位置テーブル Q12 が“1”を含んでいる場合（ステップ S305 の NO）、復号部 101 は、ステップ S301 へリターンし、以降の動作を実行する。

[0306] 以上のように動作することで、係数バッファ 114 内には、係数行列 Q11 から抽出された非ゼロ係数 W が格納され、変数バッファ 115 内には、係数バッファ 114 内に格納された非ゼロ係数 W と対応する変数 X が格納される。それにより、積和回路 116 が実行する係数 W と変数 X との乗算回数を大幅に削減することが可能となるため、係数行列の復号開始から畳み込み処理の完了までに要する時間を大幅に短縮することが可能となる。

[0307] 3. 2. 2 復号処理の具体例

続いて、本実施形態に係る復号部 101 が実行する復号処理の具体例を、図 16 及び図 17 を用いつつ、図 15 に例示したフローチャートに沿って説明する。図 16 は、本実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。図 17 は、図 16 に示す具体例における位相ポイ

ンタ、非ゼロ係数キューポインタ及び変数キューポインタの動きを説明するための図である。なお、図16及び図17には、図10及び図11と同様に、係数行列Q11が1行8列の構成をしており、その3列目と5列目と8列目とに非ゼロ係数が格納されている場合が例示されている。また、3列目の非ゼロ係数は0.1であり、5列目の非ゼロ係数は-0.8であり、8列目の非ゼロ係数は0.6であるとする。さらに、図16及び図17では、図13と同様に、変数行列Q14が係数 $X_0 \sim X_7$ の8つの係数 X が1行8列に配列する構成をしているものとする。

[0308] 図16に示すように、復号部101は、非ゼロ係数キューポインタ $waddr$ を生成及び管理するアドレス生成部111aと、変数キューポインタ $xaddr$ を生成及び管理するアドレス生成部111bと、メモリ102から読み出された係数 W を順次格納することで非ゼロ係数 W よりなる係数行列Q21を保持する係数バッファ114と、メモリ102から読み出された変数 X を順次格納することで非ゼロ係数 W と乗算される変数 X よりなる変数行列Q24を保持する変数バッファ115とを備える。

[0309] 図16に示す構成において、復号部101は、まず、ゼロ係数位置テーブルQ12(=“00101001”)及び非ゼロ係数テーブルQ13(=“0.1”、“-0.8”、“0.6”)を係数保持部103から取得する(図15のステップS101に相当)。また、復号部101は、最初の繰返し($itr=0$)では、アドレス生成部111aにおいて管理される非ゼロ係数キューポインタ $waddr$ 及びアドレス生成部111bにおいて管理される変数キューポインタ $xaddr$ に先頭アドレスをセットする(図15のステップS103及びS201に相当)。それにより、図17に示すように、非ゼロ係数キューポインタ $waddr$ は非ゼロ係数テーブルQ13における先頭の値“0.1”を指し示し、変数キューポインタ $xaddr$ は変数行列Q14における先頭の変数 X_0 を指し示すこととなる。

[0310] 次に、復号部101は、読み出したゼロ係数位置テーブルQ12をプライオリティエンコーダ104に入力して、先頭の“1”の位置、すなわち係数

行列Q 1 1における先頭の非ゼロ係数Wの位置を特定する。上述したように、 $itr = 0$ では、ゼロ係数位置テーブルQ 1 2は“0 0 1 0 1 0 0 1”の行列であるため、プライオリティエンコーダ1 0 4は、入力a 2に設定された値 $p = 2$ を出力oとして出力する。これに対し、復号部1 0 1は、プライオリティエンコーダ1 0 4から出力された値 $q = 2$ を位相ポインタqに設定する（図1 5のステップS 3 0 1に相当）。

[0311] 次に、復号部1 0 1は、現在の非ゼロ係数キューポインタw a d rに基づいて非ゼロ係数テーブルQ 1 3を参照し（図1 5及び図1 6のステップS 1 0 5に相当）、非ゼロ係数キューポインタw a d rが指し示す値（非ゼロ係数）WQ u e [w a d r]（ $= 0, 1$ ）を非ゼロ係数テーブルQ 1 3から取り出して、係数バッファ1 1 4に格納する（図1 5及び図1 6のステップS 1 0 6に相当）。

[0312] また、復号部1 0 1は、現在の変数キューポインタx a d rと位相ポインタqとに基づいて変数行列Q 1 4を参照し（図1 5及び図1 6のステップS 3 0 2に相当）、変数キューポインタx a d rから位相ポインタq分だけ末尾側へシフトさせたアドレスに格納されている変数XQ u e [w a d r + q]（ $= X_2$ ）を変数行列Q 1 4から取り出して、変数バッファ1 1 5に格納する（図1 5及び図1 6のステップS 3 0 3に相当）。

[0313] その後、アドレス生成部1 1 1 aにおいて次の非ゼロ係数Wを指し示すアドレスに非ゼロ係数キューポインタw a d rがインクリメントされるとともに（図1 5のステップS 1 0 7に相当）、ゼロ係数位置テーブルQ 1 2において現在の位相ポインタqが指し示す値“1”が“0”に更新される（図1 5のステップS 3 0 4に相当）。続いて、復号部1 0 1によりゼロ係数位置テーブルQ 1 2の全ての値が“0”であるか否かが判定され（図1 5のステップS 3 0 5に相当）、次の繰返し（ $itr = 1$ ）が実行される（図1 5のステップS 3 0 5のNOに相当）。

[0314] 図1 7に示すように、 $itr = 1$ では、ゼロ係数位置テーブルQ 1 2は“0 0 0 0 1 0 0 1”の行列であるため、プライオリティエンコーダ1 0 4は

、入力 a_4 に設定された値 $p = 4$ を出力 o として出力する。これに対し、復号部 101 は、プライオリティエンコーダ 104 から出力された値 $q = 4$ を位相ポインタ q に設定する（図 15 のステップ S301 に相当）。

[0315] 次に、復号部 101 は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル $Q13$ を参照し（図 15 及び図 16 のステップ S105 に相当）、非ゼロ係数キューポインタ $wadr$ が指し示す値（非ゼロ係数） $WQueue[wadr]$ ($= -0.8$) を非ゼロ係数テーブル $Q13$ から取り出して、係数バッファ 114 に格納する（図 15 及び図 16 のステップ S106 に相当）。

[0316] また、復号部 101 は、現在の変数キューポインタ $xadr$ と位相ポインタ q とに基づいて変数行列 $Q14$ を参照し（図 15 及び図 16 のステップ S302 に相当）、変数キューポインタ $xadr$ から位相ポインタ q 分だけ末尾側へシフトさせたアドレスに格納されている変数 $XQueue[wadr + q]$ ($= X_4$) を変数行列 $Q14$ から取り出して、変数バッファ 115 に格納する（図 15 及び図 16 のステップ S303 に相当）。

[0317] その後、アドレス生成部 111a において次の非ゼロ係数 W を指し示すアドレスに非ゼロ係数キューポインタ $wadr$ がインクリメントされるとともに（図 15 のステップ S107 に相当）、ゼロ係数位置テーブル $Q12$ において現在の位相ポインタ q が指し示す値 “1” が “0” に更新される（図 15 のステップ S304 に相当）。続いて、復号部 101 によりゼロ係数位置テーブル $Q12$ の全ての値が “0” であるか否かが判定され（図 15 のステップ S305 に相当）、次の繰返し ($itr = 2$) が実行される（図 15 のステップ S305 の NO に相当）。

[0318] 図 17 に示すように、 $itr = 2$ では、ゼロ係数位置テーブル $Q12$ は “00000001” の行列であるため、プライオリティエンコーダ 104 は、入力 a_7 に設定された値 $p = 7$ を出力 o として出力する。これに対し、復号部 101 は、プライオリティエンコーダ 104 から出力された値 $q = 7$ を位相ポインタ q に設定する（図 15 のステップ S301 に相当）。

[0319] 次に、復号部101は、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル $Q13$ を参照し（図15及び図16のステップS105に相当）、非ゼロ係数キューポインタ $wadr$ が指し示す値（非ゼロ係数） $WQue[wadr]$ （=0.6）を非ゼロ係数テーブル $Q13$ から取り出して、係数バッファ114に格納する（図15及び図16のステップS106に相当）。

[0320] また、復号部101は、現在の変数キューポインタ $xadr$ と位相ポインタ q とに基づいて変数行列 $Q14$ を参照し（図15及び図16のステップS302に相当）、変数キューポインタ $xadr$ から位相ポインタ q 分だけ末尾側へシフトさせたアドレスに格納されている変数 $XQue[wadr+q]$ （= X_7 ）を変数行列 $Q14$ から取り出して、変数バッファ115に格納する（図15及び図16のステップS303に相当）。

[0321] その後、アドレス生成部111aにおいて次の非ゼロ係数 W を指し示すアドレスに非ゼロ係数キューポインタ $wadr$ がインクリメントされるとともに（図15のステップS107に相当）、ゼロ係数位置テーブル $Q12$ において現在の位相ポインタ q が指し示す値“1”が“0”に更新されるが（図15のステップS304に相当）、この更新により、ゼロ係数位置テーブル $Q12$ の全ての値が“0”となるため（図15のステップS305のYES）、次の繰返しが実行されずに、本動作が終了される。

[0322] 3. 3 作用・効果

以上のように動作することで、処理完了後の係数バッファ114内には、非ゼロ係数 W で構成された係数行列 $Q21$ が復元され、変数バッファ115内には、非ゼロ係数 W に対応する変数 X で構成された変数行列 $Q24$ が配置される。それにより、係数行列 $Q21$ の作成に要するクロック数を係数行列 $Q11$ の復元に要するクロック数（例えば8クロック）よりも少ないクロック数（例えば3クロック）に低減することが可能となる。また、積和演算においてゼロ係数 W と変数 X とを乗算する処理を省略することも可能となる。その結果、全体としてのスループットが向上されるため、係数行列 $Q11$ の

復号開始から畳み込み処理の完了までに要する時間を大幅に短縮することが可能となる。なお、係数バッファ 114 に復元された係数行列 Q21 及び変数バッファ 115 内に配置された変数行列 Q24 は、それぞれの先頭から順に順次読み出されて積和回路 116 に入力されることで、積和演算による畳み込み処理が実行される。

[0323] また、積和演算に必要な係数 W 及び変数 X をメモリ 102 から選択的に読み出すことが可能となるため、バストラフィックの低減や内部バッファ（係数バッファ 114、変数バッファ 115 等）の規模低減などの効果を奏することも可能となる。

[0324] さらに、第 2 の実施形態と同様に、係数バッファ 114 及び変数バッファ 115 を FIFO として構成することで、係数バッファ 114 及び変数バッファ 115 への全ての係数 W 及び変数 X の格納が完了する前に、積和回路 116 への係数 W 及び変数 X の順次入力が可能となるため、係数行列 Q11 の復号開始から畳み込み処理の完了までに要する時間をより短縮することも可能である。ただし、第 3 の実施形態では、係数 W と変数 X とを乗算する回数がゼロ係数位置テーブル Q12 に含まれる “1” の数に依存して変動する。そこで本実施形態では、ゼロ係数位置テーブル Q12 に含まれる “1” の数と同じ回数、積和回路 116 に演算を実行させるための構成や、FIFO 制御においてデータの終了が積和回路 116 に通知するような構成等が付加されてよい。

[0325] その他の構成、動作及び効果は、上述した実施形態と同様であってよいため、ここでは詳細な説明を省略する。

[0326] 4. 第 4 の実施形態

上述した実施形態では、変数行列 Q14 が圧縮符号化されていない場合について例示したが、変数行列 Q14 を圧縮符号化することで、さらなるメモリ量の削減を図ることも可能である。また、係数行列 Q11 と変数行列 Q14 との両方がスパース表現にて圧縮符号化されている場合、係数 W と変数 X との両方が揃っている部分以外の部分の演算を省略することが可能となるた

め、処理量の削減や処理時間の短縮等も可能となる。そこで第4の実施形態では、係数行列 Q_{11} に加えて、変数行列 Q_{14} もスパース表現により圧縮符号化されている場合について、例を挙げて説明する。

[0327] なお、本実施形態に係るニューラルネットワーク処理装置は、第3の実施形態において図7等を引用して説明したニューラルネットワーク処理装置11と同様の構成であってよい。ここでは詳細な説明を省略する。また、以下の説明において、畳み込み対象の変数行列 Q_{14} は、畳み込み処理に用いられる係数行列 Q_{11} と同じデータ長及びデータ構造を備えているものとする。

[0328] さらに、本実施形態では、係数行列 Q_{11} と同様に、変数行列 Q_{14} が、図6に例示したような、ゼロ変数位置テーブル Q_{32} と、非ゼロ変数テーブル Q_{33} とに圧縮符号化されているものとする。なお、ゼロ変数位置テーブル Q_{32} は、もとの変数行列 Q_{14} のタップ数と同じタップ数のテーブルとなっており、各タップの数値は、変数行列 Q_{14} においてそのタップと対応する位置にある変数 X の値が0（ゼロ）であるか否かを示している。また、非ゼロ変数テーブル Q_{33} は、もとの変数行列 Q_{14} におけるゼロ変数ではない変数 X の値を示すテーブルとなっており、非ゼロ変数テーブル Q_{33} における各タップの値は、変数行列 Q_{14} におけるゼロ変数ではない1つの変数 X の値であってよい。

[0329] 4. 1 スパース表現係数の復号に好適な復号部

本実施形態に係る復号部101も、上述した実施形態に係る復号部101と同様に、ソフトウェアとして実装することも可能である、ハードウェアとして実装することも可能である。以下の説明では、ハードウェア構成として実装された復号部について、具体例を挙げて説明する。

[0330] 4. 1. 1 復号部が実行する復号処理

図18は、本実施形態に係る復号部が実行する動作、すなわちスパース表現係数を復号する際の処理の一例を示すフローチャートである。図18では、明確化のため、復号部が実行する復号処理をフローチャート化して説明す

るが、実際には、図18に係る復号アルゴリズムが図7に例示したニューラルネットワーク処理装置11の復号部41、61及び81にハードウェア構成として実装されてよい。また、図18において、図9、図12又は図15に例示した動作と同様の動作については、それを引用することで、その詳細な説明を省略する。

[0331] 図18に示すように、復号部101は、まず、係数保持部103からゼロ係数位置テーブル(SpMatW)Q12及び非ゼロ係数テーブルQ13を読み出すとともに(ステップS401)、所定のメモリ領域に格納されているスパース表現された変数行列(SpMatX)、すなわち、ゼロ変数位置テーブル(SpMatX)32及び非ゼロ変数テーブルQ33を読み出す(ステップS402)。読み出されたゼロ係数位置テーブルQ12及び非ゼロ係数テーブルQ13と、ゼロ変数位置テーブルQ32及び非ゼロ変数テーブルQ33とは、例えば、復号部101のメモリ102における既知のアドレスに格納される。

[0332] 次に、復号部101は、ゼロ係数位置テーブルQ12とゼロ変数位置テーブルQ32とについて、対応する値同士の論理積(AND)を取ることで、係数行列Q11と変数行列Q14との双方において非ゼロの値が存在するタップの位置を示す非ゼロ位置テーブル(SpMat)Q40を生成する(ステップS403)。

[0333] 次に、復号部101は、図9のステップS103と同様に、非ゼロ係数キューポインタwadrに先頭アドレスをセットするとともに、図12のステップS201と同様に、変数キューポインタxadrに先頭アドレスをセットする。

[0334] 次に、復号部101は、ステップS403で生成した非ゼロ位置テーブルQ40をプライオリティエンコーダ104に入力して評価することで、“1”が入力された入力のうちで最も優先順位が高い入力(すなわち、非ゼロ位置テーブルQ40における最初の“1”に対応する入力)を特定し(P.E. SpMat)、特定された入力に設定された値を位相ポインタqに設定

する ($q \leftarrow 1st_nonzero$) (ステップS404)。

[0335] 続いて、復号部101は、図18に示すステップS410～S107の動作と、ステップS420～S426の動作とを実行するが、これらの動作はハードウェア上で並列に実行されてよいため、図18のフローチャートでは、これらを並列に記載する。

[0336] ステップS410～S107の動作では、復号部101は、まず、ステップS401で読み込んだゼロ係数位置テーブル ($SpMatW$) $Q12$ をプライオリティエンコーダ104に入力して評価することで、“1”が入力された入力のうちで最も優先順位が高い入力（すなわち、ゼロ係数位置テーブル $Q12$ における最初の“1”に対応する入力）を特定し ($P.E. SpMatW$)、特定された入力に設定された値を位相ポインタ qw に設定する ($qw \leftarrow 1st_nonzero$) (ステップS410)。

[0337] 次に、復号部101は、ステップS404でセットした非ゼロ位置テーブル $Q40$ に対する位相ポインタ q と、ステップS410でセットしたゼロ係数位置テーブル $Q12$ に対する位相ポインタ qw とが同じ値であるか否かを判定する ($qw == q?$) (ステップS411)。位相ポインタ qw と位相ポインタ q とが同じ値でない場合 (ステップS411の $False$)、復号部101は、非ゼロ係数キューポインタ $wadr$ をインクリメントするとともに ($wadr++$) (ステップS412)、ゼロ係数位置テーブル $Q12$ における位相ポインタ qw が指し示す値を“1”から“0”に書き換え ($SpMatW[qw] \leftarrow 0$) (ステップS413)、ステップS410へリターンする。一方、位相ポインタ qw と位相ポインタ q とが同じ値である場合 (ステップS411の $True$)、復号部101は、図9のステップS105～S107と同様に、メモリ102内の非ゼロ係数テーブル $Q13$ から非ゼロ係数 $WQue[wadr]$ を読み出して係数バッファ $Wbuf$ における所定の位置に格納するとともに、非ゼロ係数キューポインタ $wadr$ をインクリメントし、その後、ステップS431へ移行する。

[0338] これに対し、ステップS410～S107に対して並行して実行されるス

ステップS420～S426の動作では、復号部101は、まず、ステップS402で読み込んだゼロ変数位置テーブル(SpMatX)Q32をプライオリティエンコーダ104に入力して評価することで、“1”が入力された入力のうちで最も優先順位が高い入力（すなわち、ゼロ変数位置テーブルQ32における最初の“1”に対応する入力）を特定し(P. E. SpMatX)、特定された入力に設定された値を位相ポインタqxに設定する($qx \leftarrow 1st \ non \ zero$) (ステップS420)。

[0339] 次に、復号部101は、ステップS404でセットした非ゼロ位置テーブルQ40に対する位相ポインタqと、ステップS420でセットしたゼロ変数位置テーブルQ32に対する位相ポインタqxとが同じ値であるか否かを判定する($qx == q?$) (ステップS421)。位相ポインタqxと位相ポインタqとが同じ値でない場合(ステップS421のFalse)、復号部101は、変数キューポインタxadrをインクリメントするとともに($xadr++$) (ステップS422)、ゼロ変数位置テーブルQ32における位相ポインタqxが指し示す値を“1”から“0”に書き換え($SpMatX[qx] \leftarrow 0$) (ステップS423)、ステップS420へリターンする。一方、位相ポインタqxと位相ポインタqとが同じ値である場合(ステップS421のTrue)、復号部101は、メモリ102内の非ゼロ変数テーブルQ33から非ゼロ変数XQue[xadr]を読み出し(ステップS424)、読み出された非ゼロ変数XQue[xadr]を変数バッファXbufにおける所定の位置に格納するとともに(ステップS425)、非ゼロ変数キューポインタxadrをインクリメントし(ステップS426)、その後、ステップS431へ移行する。

[0340] このように、ステップS410～S107に示す動作と、ステップS420～S426に示す動作とを並列に実行することで、係数バッファ114及び変数バッファ115に、係数Wと変数Xとの両方が揃っている係数W及び変数Xを格納することが可能となる。

[0341] ステップS431では、復号部101は、非ゼロ位置テーブルQ40にお

ける、ステップS404において特定された入力に対応する値を“0”に書き換える(SpMat[q])。そして、復号部101は、ステップS431で更新後の非ゼロ位置テーブルQ40の値が全て“0”であるか否かを判定し(SpMat==0?) (ステップS432)、全て“0”である場合(ステップS432のYES)、本動作を終了する。一方、非ゼロ位置テーブルQ40が“1”を含んでいる場合(ステップS432のNO)、復号部101は、ステップS404へリターンし、以降の動作を実行する。

[0342] 以上のように動作することで、係数バッファ114及び変数バッファ115内には、係数Wと変数Xとの両方が揃っている、すなわち、係数Wと変数Xとのいずれもゼロでないものが格納されるため、メモリ量の削減を図ることが可能である。また、係数Wと変数Xとの両方が揃っている部分以外の部分の演算を省略することが可能となるため、処理量の削減や処理時間の短縮等も可能となる。それにより、積和回路116が実行する係数Wと変数Xとの乗算回数を大幅に削減することが可能となるため、係数行列の復号開始から畳み込み処理の完了までに要する時間を大幅に短縮することが可能となる。

[0343] 4. 1. 2 復号処理の具体例

続いて、本実施形態に係る復号部101が実行する復号処理の具体例を、図19及び図20並びに図21を用いつつ、図18に例示したフローチャートに沿って説明する。図19は、本実施形態に係る復号部が実行する復号処理の具体例を説明するための図である。図20及び図21は、図19に示す具体例における位相ポインタ、非ゼロ係数キューポインタ及び変数キューポインタの動きを説明するための図である。なお、図19～図21には、図10及び図11と同様に、係数行列Q11が1行8列の構成をしており、その3列目と5列目と8列目とに非ゼロ係数が格納されている場合が例示されている。また、3列目の非ゼロ係数は0.1であり、5列目の非ゼロ係数は0.8であり、8列目の非ゼロ係数は0.6であるとする。さらに、図19～図21では、図13と同様に、変数行列Q14が係数 $X_0 \sim X_7$ の8つ

の係数 X が1行8列に配列する構成をしているものとする。

[0344] 図19に示すように、復号部101は、非ゼロ係数キューポインタ $wadr$ を生成及び管理するアドレス生成部111aと、変数キューポインタ $xadr$ を生成及び管理するアドレス生成部111bと、メモリ102から読み出された係数 W を順次格納することで非ゼロ係数 W よりなる係数行列 $Q21$ を保持する係数バッファ114と、メモリ102から読み出された変数 X を順次格納することで非ゼロ係数 W と乗算される変数 X よりなる変数行列 $Q24$ を保持する変数バッファ115とを備える。

[0345] 図19に示す構成において、復号部101は、まず、ゼロ係数位置テーブル($SpMatW$) $Q12$ (=“00101001”)及び非ゼロ係数テーブル $Q13$ (=“0.1”、“-0.8”、“0.6”)を係数保持部103から取得するとともに(図18のステップS401に相当)、ゼロ変数位置テーブル($SpMatX$) $Q32$ 及び非ゼロ変数テーブル $Q33$ を所定のメモリ領域から取得する(図18のステップS402に相当)。そして、復号部101は、ゼロ係数位置テーブル $Q12$ とゼロ変数位置テーブル $Q32$ との論理積を取ることで、係数 W と変数 X との両方が揃っている位置を示す非ゼロ位置テーブル($SpMat$) $Q40$ を生成する(図18のステップS403に相当)。

[0346] また、復号部101は、最初の繰返し($itr=0$)では、アドレス生成部111aにおいて管理される非ゼロ係数キューポインタ $wadr$ 及びアドレス生成部111bにおいて管理される変数キューポインタ $xadr$ に先頭アドレスをセットする(図18のステップS103及びS201に相当)。それにより、図20に示すように、非ゼロ係数キューポインタ $wadr$ は非ゼロ係数テーブル $Q13$ における先頭の値“0.1”を指し示し、変数キューポインタ $xadr$ は非ゼロ変数テーブル $Q33$ における先頭の変数 X_0 を指し示すこととなる。

[0347] 次に、復号部101は、読み出した非ゼロ位置テーブル $Q40$ をプライオリティエンコーダ104に入力して、先頭の“1”の位置、すなわち係数 W

と変数 X との両方が揃った位置における先頭を特定する。上述したように、 $itr = 0$ では、非ゼロ位置テーブル $Q40$ は“00100001”の行列であるため、プライオリティエンコーダ104は、入力 $a2$ に設定された値 $p = 2$ を出力 o として出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値 $q = 2$ を位相ポインタ q に設定する（図18のステップS404に相当）。

[0348] 次に、復号部101は、図18におけるステップS410～S107の動作と、ステップS420～S426の動作とを並列に実行する。

[0349] 図18におけるステップS410～S107の動作では、復号部101は、ステップS401で読み出したゼロ係数位置テーブル $Q12$ をプライオリティエンコーダ104に入力して、先頭の“1”の位置、すなわち係数行列 $Q11$ における先頭の非ゼロ係数 W の位置を特定する。図20に示すように、 $itr = 0$ では、ゼロ係数位置テーブル $Q12$ は“00101001”の行列であるため、プライオリティエンコーダ104は、入力 $a2$ に設定された値 $pw = 2$ を出力 o として出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値 $qw = 2$ を位相ポインタ qw に設定する（図18のステップS410に相当）。

[0350] 続いて、復号部101は、位相ポインタ $qw = 2$ と位相ポインタ $q = 2$ とを比較し（図18のステップS411に相当）、両者が一致していることから（図18のステップS411のTrueに相当）、現在の非ゼロ係数キューポインタ $wadr$ に基づいて非ゼロ係数テーブル $Q13$ を参照し（図18及び図19のステップS105に相当）、非ゼロ係数キューポインタ $wadr$ が指し示す値（非ゼロ係数） $WQueue[wadr]$ （ $= 0, 1$ ）を非ゼロ係数テーブル $Q13$ から取り出して、係数バッファ114に格納する（図18及び図19のステップS106に相当）。そして、復号部101は、アドレス生成部111aにおいて管理されている非ゼロ係数キューポインタ $wadr$ をインクリメントする（図18のステップS107に相当）。

[0351] 一方、図18におけるステップS420～S426の動作では、復号部1

01は、ステップS402で読み出したゼロ変数位置テーブルQ32をプライオリティエンコーダ104に入力して、先頭の“1”の位置、すなわち変数行列Q14における先頭の非ゼロ変数Xの位置を特定する。図20に示すように、 $itr=0$ では、ゼロ変数位置テーブルQ32は“10100111”の行列であるため、プライオリティエンコーダ104は、入力a0に設定された値 $qx=0$ を出力oとして出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値 $qx=0$ を位相ポイントqxに設定する（図18のステップS420に相当）。

[0352] 続いて、復号部101は、位相ポイント $qx=0$ と位相ポイント $q=2$ とを比較し（図18のステップS421に相当）、両者が一致していないことから（図18のステップS421のFalseに相当）、図20に示すように、非ゼロ変数キューポイントxadrをインクリメントするとともに（図18のステップS422に相当）、ゼロ変数位置テーブルQ32における現在の位相ポイントqxが指し示す値を“0”に書き換え（図18のステップS423に相当）、書き換え後のゼロ変数位置テーブルQ32を再度プライオリティエンコーダ104に入力する（図18のステップS420に相当）。

[0353] 図20に示すように、書き換え後のゼロ係数位置テーブルQ12は“00100111”の行列であるため、プライオリティエンコーダ104からは、値 $qx=2$ が出力oとして出力される。これに対し、復号部101は、位相ポイントqxと位相ポイントqとが一致していることから（図18のステップS421のTrueに相当）、現在の非ゼロ変数キューポイントxadrに基づいて非ゼロ変数テーブルQ33を参照し（図18及び図19のステップS424に相当）、非ゼロ変数キューポイントxadrが指し示す値（非ゼロ変数） $X_{Que}[xadr]$ （= X_2 ）を非ゼロ変数テーブルQ33から取り出して、変数バッファ115に格納する（図18及び図19のステップS425に相当）。そして、復号部101は、アドレス生成部111bにおいて管理されている非ゼロ変数キューポイントxadrをインクリメント

する（図18のステップS426に相当）。

[0354] その後、復号部101は、非ゼロ位置テーブルQ40において現在の位相ポインタqが指し示す値“1”が“0”に更新される（図18のステップS431に相当）。続いて、復号部101により非ゼロ位置テーブルQ40の全ての値が“0”であるか否かが判定され（図18のステップS432に相当）、次の繰返し（i t r = 1）が実行される（図18のステップS432のNOに相当）。

[0355] 図21に示すように、i t r = 1では、非ゼロ位置テーブルQ40は“0000001”の行列であるため、プライオリティエンコーダ104は、入力a7に設定された値p = 7を出力oとして出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値q = 7を位相ポインタqに設定する（図18のステップS404に相当）。

[0356] 次に、復号部101は、図18におけるステップS410～S107の動作において、ステップS401で読み出したゼロ係数位置テーブルQ12をプライオリティエンコーダ104に入力して、先頭の“1”の位置を特定する。図21に示すように、i t r = 1では、ゼロ係数位置テーブルQ12は“00001001”の行列であるため、プライオリティエンコーダ104は、入力a4に設定された値p w = 4を出力oとして出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値q w = 4を位相ポインタq wに設定する（図18のステップS410に相当）。

[0357] 続いて、復号部101は、位相ポインタq w = 4と位相ポインタq = 7とを比較し（図18のステップS411に相当）、両者が一致していないことから（図18のステップS411のFalseに相当）、図21に示すように、非ゼロ係数キューポインタw a d rをインクリメントするとともに（図18のステップS412に相当）、ゼロ係数位置テーブルQ12における現在の位相ポインタq wが指し示す値を“0”に書き換え（図18のステップS413に相当）、書き換え後のゼロ係数位置テーブルQ12を再度プライオリティエンコーダ104に入力する（図18のステップS410に相当）。

。

[0358] 図21に示すように、書き換え後のゼロ係数位置テーブルQ12は“000001”の行列であるため、プライオリティエンコーダ104からは、値 $qx = 7$ が出力oとして出力される。これに対し、復号部101は、位相ポインタqwと位相ポインタqとが一致していることから（図18のステップS411のTrueに相当）、現在の非ゼロ係数キューポインタwadrに基づいて非ゼロ係数テーブルQ13を参照し（図18及び図19のステップS105に相当）、非ゼロ係数キューポインタwadrが指し示す値（非ゼロ係数）WQueue[wadr]（=0.6）を非ゼロ係数テーブルQ13から取り出して、係数バッファ114に格納する（図18及び図19のステップS106に相当）。そして、復号部101は、アドレス生成部111aにおいて管理されている非ゼロ係数キューポインタwadrをインクリメントする（図18のステップS107に相当）。

[0359] 一方、図18におけるステップS420～S426の動作では、復号部101は、ステップS402で読み出したゼロ変数位置テーブルQ32をプライオリティエンコーダ104に入力して、先頭の“1”の位置を特定する。図21に示すように、itr=1では、ゼロ変数位置テーブルQ32は“0000111”の行列であるため、プライオリティエンコーダ104は、入力a5に設定された値 $qx = 5$ を出力oとして出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値 $qx = 5$ を位相ポインタqxに設定する（図18のステップS420に相当）。

[0360] 続いて、復号部101は、位相ポインタqx=5と位相ポインタq=2とを比較し（図18のステップS421に相当）、両者が一致していないことから（図18のステップS421のFalseに相当）、図21に示すように、非ゼロ変数キューポインタxadrをインクリメントするとともに（図18のステップS422に相当）、ゼロ変数位置テーブルQ32における現在の位相ポインタqxが指し示す値を“0”に書き換え（図18のステップS423に相当）、書き換え後のゼロ変数位置テーブルQ32を再度プライ

オリティエンコーダ104に入力する（図18のステップS420に相当）。

[0361] 図21に示すように、書き換え後のゼロ係数位置テーブルQ12は“0000011”の行列であるため、プライオリティエンコーダ104は、入力a6に設定された値 $q_x = 6$ を出力oとして出力する。これに対し、復号部101は、プライオリティエンコーダ104から出力された値 $q_x = 6$ を位相ポイント q_x に設定する（図18のステップS420に相当）。

[0362] 続いて、復号部101は、位相ポイント $q_x = 6$ と位相ポイント $q = 2$ とを比較し（図18のステップS421に相当）、両者が一致していないことから（図18のステップS421のFalseに相当）、図21に示すように、非ゼロ変数キューポイントxadrをインクリメントするとともに（図18のステップS422に相当）、ゼロ変数位置テーブルQ32における現在の位相ポイント q_x が指し示す値を“0”に書き換え（図18のステップS423に相当）、書き換え後のゼロ変数位置テーブルQ32を再度プライオリティエンコーダ104に入力する（図18のステップS420に相当）。

[0363] 図21に示すように、書き換え後のゼロ係数位置テーブルQ12は“0000001”の行列であるため、プライオリティエンコーダ104からは、値 $q_x = 7$ が出力oとして出力される。これに対し、復号部101は、位相ポイント q_x と位相ポイント q とが一致していることから（図18のステップS421のTrueに相当）、現在の非ゼロ変数キューポイントxadrに基づいて非ゼロ変数テーブルQ33を参照し（図18及び図19のステップS424に相当）、非ゼロ変数キューポイントxadrが指し示す値（非ゼロ変数） $X_{Que}[xadr]$ （= X_7 ）を非ゼロ変数テーブルQ33から取り出して、変数バッファ115に格納する（図18及び図19のステップS425に相当）。そして、復号部101は、アドレス生成部111bにおいて管理されている非ゼロ変数キューポイントxadrをインクリメントする（図18のステップS426に相当）。

[0364] その後、非ゼロ位置テーブルQ 4 0において現在の位相ポインタqが指し示す値“1”が“0”に更新されるが（図18のステップS 4 3 1に相当）、この更新により、非ゼロ位置テーブルQ 4 0の全ての値が“0”となるため（図18のステップS 4 3 2のYES）、次の繰返しが実行されずに、本動作が終了される。

[0365] 4. 2 作用・効果

以上のように動作することで、処理完了後の係数バッファ1 1 4及び変数バッファ1 1 5内には、係数Wと変数Xとの両方が揃っているものが格納される。それにより、積和演算において実質的に寄与しない係数W及び変数Xが係数バッファ1 1 4及び変数バッファ1 1 5に格納されないため、さらなるメモリ量の削減を図ることも可能である。また、係数Wと変数Xとの両方が揃っている部分以外の部分の演算を省略することが可能となるため、処理量の削減や処理時間の短縮等も可能となる。

[0366] その他の構成、動作及び効果は、上述した実施形態と同様であってよいため、ここでは詳細な説明を省略する。

[0367] 5. 第5の実施形態

上述した実施形態では、スパース表現により圧縮符号化された係数行列（及び変数行列）を復号する場合について例示した。それに対し、第5の実施形態では、畳み込み処理の結果として得られた積和結果、すなわち変数行列をスパース表現により圧縮符号化してメモリに書き込む場合について、例を挙げて説明する。

[0368] なお、本実施形態に係るニューラルネットワーク処理装置は、第1の実施形態において図7を用いて説明したニューラルネットワーク処理装置1 1と同様の構成であってよいため、ここでは詳細な説明を省略する。また、以下の説明において、畳み込み対象の変数行列Q 1 4は、畳み込み処理に用いられる係数行列Q 1 1と同じデータ長及びデータ構造を備えているものとする。

[0369] 5. 1 比較例

図 2 2 は、比較例に係る積和結果のメモリへの書き込みを説明するための図である。図 2 2 に示すように、比較例では、畳み込み処理部 2 1 / 2 3 / 2 5 の積和回路 1 1 6 から積和演算の単位ごとに出力された積和演算結果である変数 $X_0 \sim X_7$ が、各変数 $X_0 \sim X_7$ の値に関わらず、次の段のプーリング処理部 2 2 / 2 4 のメモリ 5 1 / 7 1 に順次書き込まれる。ここで、上述の実施形態において例示した変数行列 Q 1 4 のように、例えば変数 X_1 、 X_3 、 X_4 がゼロであった場合、変数行列 Q 1 4 をスパース表現により圧縮符号化した上でメモリ 5 1 / 7 1 に書き込むことで、メモリ 5 1 / 7 1 上のデータ量を削減することが可能となる。

[0370] 5. 2 符号化部

図 2 3 は、本実施形態に係る符号化部の概略構成例を示す回路図である。なお、図 2 3 に例示する符号化部 2 0 0 は、例えば、図 7 に例示するニューラルネットワーク処理装置 1 1 の各畳み込み処理部 2 1 / 2 3 / 2 5 における積和結果の出力段に配置されてよい。

[0371] 図 2 3 に示すように、本実施形態に係る符号化部 2 0 0 は、例えば、バッファ回路 2 0 1 と、判定回路 2 0 2 と、セクタ 2 0 3 と、遅延回路 2 0 4 と、スパース行列バッファ (S p M a t B u f) (第 1 バッファ) 2 0 5 と、書込みバッファ (第 2 バッファ) 2 0 6 とを備える。

[0372] なお、積和回路 1 1 6 は、上述した実施形態と同様であってよく、例えば、変数 X と係数 W とを乗算する乗算器 1 6 1 と、この乗算結果を加算する加算器 1 6 2 と、この加算結果を積和演算の単位ごとに保持するアキュムレータ 1 3 とを備えている。

[0373] 符号化部 2 0 0 のバッファ回路 2 0 1 には、1 回の積和演算が完了するごとに、アキュムレータ 1 6 3 からその結果が入力される。また、バッファ回路 2 0 1 のイネーブル端子 e_n には、1 回の積和演算が完了する度に、例えば積和回路 1 1 6 又は復号部 1 0 1 から積和終了通知が入力される。バッファ回路 2 0 1 は、積和終了通知が入力されると、保持している積和演算の結果、すなわち変数 X を判定回路 2 0 2 及び書込みバッファ 2 0 6 に出力す

る。

[0374] 判定回路202は、入力された変数Xについて、その値がゼロであるか非ゼロであるかを判定する。変数Xの値がゼロである場合（T：True）、判定回路202は、変数Xがゼロであったことを示す制御信号をセクタ203に出力する。一方、変数Xの値が非ゼロである場合（F：False）、判定回路202は、変数Xが非ゼロであったことを示す制御信号を書込みバッファ206に出力する。

[0375] セクタ203は、判定回路202から入力された制御信号に従って、“0”又は“1”の値をスパース行列バッファ205に入力する。例えば、セクタ203は、変数Xがゼロであったことを示す制御信号が入力されている場合、スパース行列バッファ205に“0”を出力し、この制御信号が入力されていない場合、“1”を出力してもよい。なお、判定回路202は変数Xが非ゼロであったことを示す制御信号をセクタ203に出力し、セクタ203がこの制御信号を受けてスパース行列バッファ205に“1”を出力するように構成されてもよい。

[0376] スパース行列バッファ205には、積和終了通知がタイミングを合わせるための遅延回路204を介して入力される。スパース行列バッファ205は、セクタ203から入力された“0”又は“1”の値を、積和終了通知のレベル遷移に従って順次保持する。これにより、スパース行列バッファ205内には、積和演算結果としてのゼロ変数位置テーブルQ52が構築される。その後、例えば復号部101等からマップ終了通知が入力されると、スパース行列バッファ205に保持されているゼロ変数位置テーブルQ52が、次の段のプーリング処理部22／24のメモリ51／71に書き込まれ、スパース行列バッファ205内のゼロ変数位置テーブルQ52がクリアされる。なお、マップとは、1回の畳み込み処理の対象となるデータ全体を意味してよい。

[0377] 書込みバッファ206には、上述したように、バッファ回路201から出力された変数Xと、判定回路202から出力された制御信号とが入力される

。また、書込みバッファ206には、非ゼロ変数Xを書き込むメモリ領域の先頭アドレスも入力される。書込みバッファ206は、判定回路202から変数Xが非ゼロであったことを示す制御信号が入力されると、バッファ回路201から入力された変数Xを一時保持する。そして、例えば、1つのマップに対する圧縮符号化処理が完了する時点で、保持している変数Xの行列、すなわち非ゼロ変数テーブルQ53を、例えばFIFO制御に従って、先頭アドレスから順番に次の段のプーリング処理部22/24のメモリ51/71内に格納する。これにより、メモリ51/71内には、積和演算結果としての非ゼロ変数テーブルQ53が格納される。なお、書込みバッファ206は、各非ゼロ変数Xの書込み先アドレスを非ゼロ変数Xの値と同期して保持し、その後、書込み先アドレスに従って非ゼロ変数Xをメモリ51/71内に書き込んでもよい。

[0378] 5. 3 作用・効果

以上のような符号化部200を各畳み込み処理部21/23/25の出力段に配置することで、畳み込み処理の結果として得られた積和結果、すなわち変数行列Q14をスパース表現により圧縮符号化してメモリに書き込むことが可能となる。それにより、メモリ51/71上のデータ量を削減することが可能となる。

[0379] その他の構成、動作及び効果は、上述した実施形態と同様であってよいため、ここでは詳細な説明を省略する。

[0380] 6. 第6の実施形態

上述した第5の実施形態において、変数Xをスパース表現化しながらメモリ51/71書き込みする際に、次の畳み込み処理部23/25における畳み込み処理（以下、単に次レイヤと称する）の係数Wのゼロ係数位置を加味することで、さらにメモリ容量の低減を図ることが可能となる。これは、「次レイヤの積和処理において、当該次レイヤのゼロ係数位置テーブルから係数Wがゼロであることが事前に分かっている位置の変数Xについては、当該次レイヤの積和処理において演算処理を省略することができるため、現レイヤ

でのメモリへの書き込みが不要である」という考えに基づいている。これは変数 X をずらしながら参照するCNN (Convolutional Neural Network) においても、後半のフルコネクション層において有効である。そこで、第6の実施形態では、次レイヤのゼロ係数位置テーブルから係数 W がゼロであることが事前に分かっている位置の変数 X についてはメモリへの書き込みを省略することができる構成について、例を挙げて説明する。

[0381] なお、本実施形態に係るニューラルネットワーク処理装置は、第5の実施形態において図7等を参照して説明したニューラルネットワーク処理装置11と同様の構成であってよいため、ここでは詳細な説明を省略する。また、以下の説明において、畳み込み対象の変数行列 Q_{14} は、畳み込み処理に用いられる係数行列 Q_{11} と同じデータ長及びデータ構造を備えているものとする。

[0382] 6. 1 符号化部

図24は、本実施形態に係る符号化部の概略構成例を示す回路図である。図24に示すように、本実施形態に係る符号化部300は、第5の実施形態において図23を用いて説明した符号化部200と同様の構成において、2つのAND回路301及び302と、レジスタ303とが追加された構成を備える。なお、図24では、説明の簡略化のため、畳み込み部21、23、25におけるメモリ102と、プーリング処理部22、24におけるメモリ51、71とが連続するメモリ領域304上に配置されている場合を例示する。

[0383] レジスタ303には、メモリ領域304（次レイヤのメモリ102）から事前に読み出された次レイヤのゼロ係数位置テーブル Q_{62} が格納される。レジスタ303は、積和終了通知の遷移に従って、ゼロ係数位置テーブル Q_{62} を構成する値“0”又は“1”を1つずつ、2つのAND回路301及び302それぞれへ出力する。

[0384] AND回路301は、判定回路202から出力された変数 X がゼロ変数であったことを示す制御信号“1”と、レジスタ303から出力された係数 W

がゼロ係数であるか非ゼロ係数であるかを示す値（“0”又は“1”）との論理積を取り、その結果をセクタ203の制御端子に入力する。

[0385] これに対し、セクタ203は、AND回路301から入力された値が“1”である場合、すなわち、バッファ回路201から入力された変数 X が非ゼロ変数 X であって且つ次レイヤの係数行列における対応する係数 W が非ゼロ係数 W である場合に“1”を出力し、AND回路301から入力された値が“1”である場合、すなわち、変数 X と次レイヤにおける係数 W との少なくとも1つの値がゼロである場合に“0”を出力する。これにより、スパー行列バッファ（SpMat Buf）205内に、次レイヤの積和演算において有効に活用される非ゼロ変数 X の位置を示すゼロ変数位置テーブルQ52が構築される。

[0386] AND回路302は、判定回路202から出力された変数 X が非ゼロ変数であったことを示す制御信号“1”と、レジスタ303から出力された係数 W がゼロ係数であるか非ゼロ係数であるかを示す値（“0”又は“1”）との論理積を取り、その結果を書込みバッファ206に入力する。

[0387] これに対し、書込みバッファ206は、AND回路302から入力された値が“1”である場合に、すなわち、バッファ回路201から入力された変数 X が非ゼロ変数 X であって且つ次レイヤの係数行列における対応する係数 W が非ゼロ係数 W である場合に、バッファ回路201から出力された変数 X を格納する。これにより、書込みバッファ206内又はメモリ領域207（メモリ51／71）内に、次レイヤの積和演算において有効に活用される非ゼロ変数 X で構成された非ゼロ変数テーブルQ53が構築される。

[0388] 6. 2 動作例

続いて、図24に示す符号化部300の動作について説明する。図25は、図24に示す符号化部の動作例を説明するための図である。なお、図25では、説明の都合上、現在処理中のレイヤ（以下、現レイヤという）の変数 X を $X_{0,n}$ （ n は自然数）とし、次レイヤの変数 X を $X_{1,n}$ とする。また、説明の簡略化のため、現レイヤにおける変数行列Q14のゼロ係数位置テーブル

Q 1 2 と係数行列 Q 1 1 のゼロ変数位置テーブルとの非ゼロの値の論理積を取った行列、すなわち変数行列 Q 5 4 の非ゼロの位置を示す行列が “1 0 1 0 0 1 1 1” であるとし、変数行列 Q 1 4 がスパース表現化されていない場合を例示する。ただし、これに限定されず、変数行列 Q 1 4 はスパース表現により圧縮符号化されていてもよい。

[0389] 図 2 5 に示すように、積和回路 1 1 6 からは、現レイヤの変数行列 Q 1 4 と係数行列 Q 1 1 とを乗算することにより得られた変数行列 Q 5 4 が出力される。図 2 4 に示す書込みバッファ 2 0 6 は、AND 回路 3 0 2 から出力された、変数 X_{1n} と次レイヤのゼロ係数位置テーブル Q 6 2 における対応する値との論理積の結果に基づき、次レイヤで有効な変数 X_{1n} をメモリ領域 3 0 4 に書き込む。また、メモリ領域 3 0 4 には、スパース行列バッファ 2 0 5 に保持されているゼロ変数位置テーブル Q 5 2 も格納される。

[0390] 6. 3 作用・効果

以上のような符号化部 3 0 0 を各畳み込み処理部 2 1 / 2 3 / 2 5 の出力段に配置することで、メモリ 5 1 / 7 1 に書き込まれる変数行列のデータ量をさらに削減することが可能となる。

[0391] その他の構成、動作及び効果は、上述した実施形態と同様であってよいため、ここでは詳細な説明を省略する。

[0392] 6. 4 変形例

なお、次レイヤで処理すべきマップが複数ある場合、例えば、読み込むべき次レイヤのゼロ係数位置テーブル Q 6 2 が複数ある場合、以下のようなバリエーションを採ることができる。

[0393] 6. 4. 1 第 1 の変形例

図 2 6 は、第 1 の変形例を説明するための模式図である。図 2 6 に示すように、各レイヤにおける処理対象のデータが複数のマップ（変数行列 Q 1 4 a ~ Q 1 4 c）から構成され、それぞれに対して異なるフィルタ係数（係数行列）が設定されている場合、次レイヤにおける全マップ分のゼロ係数位置テーブル Q 6 2 a ~ Q 6 2 c の論理和を取ることで得られたゼロ係数位置テ

ーブルQ 7 2を図2 4のレジスタ3 0 3に登録して利用することも可能である。

[0394] 6. 4. 2 第2の変形例

また、第1の変形例のように、各レイヤにおける処理対象のデータが複数のマップ（変数行列Q 1 4 a～Q 1 4 c）から構成され、それぞれに対して異なるフィルタ係数（係数行列）が設定されている場合、次レイヤにおける各マップ分のゼロ係数位置テーブルQ 6 2 a～Q 6 2 cを図2 4のレジスタ3 0 3に登録しておくことで、マップごとの積和演算結果を多重に出力するように構成することも可能である。

[0395] 6. 4. 3 第3の変形例

上述した第1及び第2の変形例は、組み合わせることが可能である。図2 7は、第1及び第2の変形例を組み合わせた第3の変形例を説明するための模式図である。図2 7に示すように、第3変形例では、次レイヤにおける複数のフィルタ係数（係数行列）が複数のグループにグループ分けされ、各グループで、ゼロ係数位置テーブルの論理和が取られる。図2 7に示す例では、グループ化された係数行列Q 6 2 b及びQ 6 2 cの論理和が取られてゼロ係数位置テーブルQ 8 2 bが生成され、レジスタ3 0 3に登録される。ただし、グループ内に1つのゼロ係数位置テーブルQ 6 2 aしか存在しない場合、そのゼロ係数位置テーブルQ 6 2 aがそのままレジスタ3 0 3に登録されてよい。

[0396] 7. 本技術の適用例

次に、上述した実施形態に係るニューラルネットワーク処理装置1 1の適用例について、いくつか例を挙げて説明する。

[0397] 7. 1 第1適用例

図2 8は、第1適用例を説明するための図である。図2 8に示すように、第1適用例では、ニューラルネットワーク処理装置1 1を、情報処理装置（プロセッサともいう）1 0 0 0に組み込まれたDNNエンジン（アクセラレータ）1 0 0 1として構成したものを示す。DNNエンジン1 0 0 1は、情

報処理装置１０００におけるプロセッサコア（単にコアともいう）１００２にバス１００３を介して接続される。これにより、プロセッサコア１００２とＤＮＮエンジン１００１とが密に連携した処理を実現することが可能である。

[0398] また、図２９に例示する情報処理装置１０００Ａのように、１つのプロセッサコア１００２に対して複数のＤＮＮエンジン１００１ａ、１００１ｂ、１００１ｃ、…がバス１００３を介して接続されてもよい。

[0399] ７．２ 第２適用例

図３０は、第２適用例に係るＩｏＴ（Internet of Things）エッジＡＩ（Artificial Intelligence）システム（情報処理システムともいう）の概略構成例を示すブロック図である。図３０に示すように、第１適用例に係るニューラルネットワーク処理装置１１を搭載する情報処理装置１０００ａ～１０００ｃは、様々なモノに搭載された様々なセンサ１１１１～１１１４に無線又は有線で接続されたＩｏＴエッジＡＩシステム１１００に組み込まれてもよい。

[0400] ７．３ 第３適用例

図３１は、第３適用例に係る電子機器の一つである自律ロボットの概略構成例を示す模式図である。図３１に示すように、第１適用例に係る情報処理装置１０００は、例えば、自律ロボット１２００の目に相当するイメージセンサ１２１１で取得されたデータを処理する処理部１２０１や、耳に相当するマイクロホン１２１２で取得されたデータを処理する処理部１２０２に適用することができる。また、第２変形例に係るＩｏＴエッジＡＩシステム１１００は、自律ロボット１２００の腕や脚に搭載された各種センサで得られた情報を処理する処理部１２０３に適用することができる。さらに、第２変形例に係るＩｏＴエッジＡＩシステム１１００は、自律ロボット１２００全体を制御する制御部１２０４にも適用することができる。

[0401] なお、イメージセンサ１２１１で取得されたデータを処理する処理部１２０１に適用する場合には、イメージセンサ１２１１のチップと情報処理装置

１０００のチップとが、互いに張り合わされた積層チップ構造を有していてもよい。

[0402] ７．４ 第４適用例

図３２は、第４適用例に係る電子機器の一つであるテレビジョンの概略構成例を示す模式図である。図３２に示すように、第１適用例に係る情報処理装置１０００及び第２適用例に係るＩｏＴエッジＡＩシステム１１００は、テレビジョン１３００が設置された空間（部屋等）の環境認識のためのサブシステムに適用することが可能である。

[0403] 第１適用例に係る情報処理装置１０００は、例えば、テレビジョン１３００の周囲を撮像するイメージセンサ１３１１で取得されたデータを処理する処理部１３０１や、テレビジョン１３００の周囲の状況を取得するセンサ、例えば、ＴｏＦ（Time-of-Flight）センサや超音波センサや低解像度カメラやマイクロホンアレイなどで取得されたデータを処理する処理部１３０２に適用することができる。

[0404] なお、イメージセンサ１３１１で取得されたデータを処理する処理部１３０１に適用する場合には、イメージセンサ１２１１のチップと情報処理装置１０００のチップとが、互いに張り合わされた積層チップ構造を有していてもよい。

[0405] ８．コンピュータの構成例

ところで、上述した実施形態に係る一連の処理は、ハードウェアにより実行することもできるし、ソフトウェアにより実行することもできる。一連の処理をソフトウェアにより実行する場合には、そのソフトウェアを構成するプログラムが、コンピュータにインストールされる。ここで、コンピュータには、専用のハードウェアに組み込まれているコンピュータや、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどが含まれる。

[0406] 図３３は、上述した一連の処理をプログラムにより実行するコンピュータのハードウェアの構成例を示すブロック図である。

- [0407] コンピュータにおいて、CPU (Central Processing Unit) 2001、ROM (Read Only Memory) 2002、RAM (Random Access Memory) 2003は、バス2004により相互に接続されている。
- [0408] バス2004には、さらに、入出カインターフェース2005が接続されている。入出カインターフェース2005には、入力部2006、出力部2007、記録部2008、通信部2009、及びドライブ2010が接続されている。
- [0409] 入力部2006は、キーボード、マウス、マイクロホン、撮像素子などよりなる。出力部2007は、ディスプレイ、スピーカなどよりなる。記録部2008は、ハードディスクや不揮発性のメモリなどよりなる。通信部2009は、ネットワークインターフェースなどよりなる。ドライブ2010は、磁気ディスク、光ディスク、光磁気ディスク、又は半導体メモリなどのリムーバブル記録媒体2011を駆動する。
- [0410] 以上のように構成されるコンピュータでは、CPU 2001が、例えば、記録部2008に記録されているプログラムを、入出カインターフェース2005及びバス2004を介して、RAM 2003にロードして実行することにより、上述した一連の処理が行われる。
- [0411] コンピュータ (CPU 2001) が実行するプログラムは、例えば、パッケージメディア等としてのリムーバブル記録媒体2011に記録して提供することができる。また、プログラムは、ローカルエリアネットワーク、インターネット、デジタル衛星放送といった、有線または無線の伝送媒体を介して提供することができる。
- [0412] コンピュータでは、プログラムは、リムーバブル記録媒体2011をドライブ2010に装着することにより、入出カインターフェース2005を介して、記録部2008にインストールすることができる。また、プログラムは、有線または無線の伝送媒体を介して、通信部2009で受信し、記録部2008にインストールすることができる。その他、プログラムは、ROM 2002や記録部2008に、あらかじめインストールしておくことができ

る。

[0413] なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであってもよいし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであってもよい。

[0414] また、上述した実施形態は、上述した実施の形態に限定されるものではなく、本技術の要旨を逸脱しない範囲において種々の変更が可能である。

[0415] 例えば、本技術は、1つの機能をネットワークを介して複数の装置で分担、共同して処理するクラウドコンピューティングの構成をとることができる。

[0416] また、上述のフローチャートで説明した各ステップは、1つの装置で実行する他、複数の装置で分担して実行することができる。

[0417] さらに、1つのステップに複数の処理が含まれる場合には、その1つのステップに含まれる複数の処理は、1つの装置で実行する他、複数の装置で分担して実行することができる。

[0418] 以上、本開示の実施形態について説明したが、本開示の技術的範囲は、上述の各実施形態そのままに限定されるものではなく、本開示の要旨を逸脱しない範囲において種々の変更が可能である。また、異なる実施形態及び変形例にわたる構成要素を適宜組み合わせてもよい。

[0419] また、本明細書に記載された各実施形態における効果はあくまで例示であって限定されるものではなく、他の効果があってもよい。

[0420] さらに、本技術は、以下の構成とすることも可能である。

[0421] (1)

第1係数行列における値がゼロの第1係数の位置を第1値で示し、前記第1係数行列における値がゼロでない第2係数の位置を第2値で示す第1ゼロ係数位置テーブルと、前記第1係数行列における前記第2係数を保持する第1非ゼロ係数テーブルとに符号化された前記第1係数行列を復号する復号部と、

前記復号部により復号された前記第 1 係数行列と、第 1 変数行列との畳み込み処理を実行する積和回路と、

を備え、

前記復号部は、前記第 1 ゼロ係数位置テーブルにおける前記第 2 値で示された位置に前記第 1 非ゼロ係数テーブルに格納された前記第 2 係数を格納することで、前記第 1 係数行列を復号する

ニューラルネットワーク処理装置。

(2)

前記復号部は、

制御端子に入力された値が前記第 1 値である場合、ゼロを出力し、前記第 2 値である場合、当該第 2 値の前記第 1 ゼロ係数位置テーブル上の位置に対応する前記第 2 係数を出力するセレクトと、

前記セレクトから入力された前記ゼロと前記第 2 係数とを配列することで前記第 1 係数行列を復元する積和器と、

を備え、

前記復号部は、前記第 1 ゼロ係数位置テーブルを構成する値を順に前記セレクトに入力する

前記 (1) に記載のニューラルネットワーク処理装置。

(3)

前記復号部は、前記第 1 ゼロ係数位置テーブルを構成する値を順に前記セレクトに入力する際に、前記セレクトに入力する前記値の前記第 1 ゼロ係数位置テーブル上の位置に対応する前記第 1 変数行列上の位置に格納された変数を取得する

前記 (2) に記載のニューラルネットワーク処理装置。

(4)

前記復号部は、前記第 2 値の前記第 1 ゼロ係数位置テーブル上の位置に対応する前記第 2 係数を取得するとともに、前記第 2 値の前記第 1 ゼロ係数位置テーブル上の位置に対応する前記第 1 変数行列上の位置に格納された変数

を取得し、取得した前記第 2 係数と前記変数とを前記積和回路に入力し、

前記積和回路は、前記復号部から入力された前記第 2 係数と前記変数とを順に乗算して加算することで、前記第 1 係数行列と前記第 1 変数行列との前記畳み込み処理を実行する

前記（１）に記載のニューラルネットワーク処理装置。

（５）

前記復号部は、それぞれ優先順位が設定された複数の入力を有し、前記第 2 値が入力された 1 以上の前記入力のうち最も前記優先順位が高い前記入力に設定された値を出力するプライオリティエンコーダを備え、

前記復号部は、前記複数の入力に対して前記第 1 ゼロ係数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記第 1 非ゼロ係数テーブルから前記第 2 係数を取得する

前記（４）に記載のニューラルネットワーク処理装置。

（６）

前記第 1 変数行列は、当該第 1 変数行列における値がゼロの第 1 変数の位置を前記第 1 値で示し、前記第 1 変数行列における値がゼロでない第 2 変数の位置を前記第 2 値で示す第 1 ゼロ変数位置テーブルと、前記第 1 変数行列における前記第 2 変数を保持する非ゼロ変数テーブルとに符号化されており、

前記復号部は、前記第 1 ゼロ係数位置テーブルを構成する値と前記第 1 ゼロ変数位置テーブルを構成する値とを論理演算し、当該論理演算の結果に基づいて、互いに乗算した結果がゼロとまらない前記第 2 係数と前記第 2 変数とを前記第 1 非ゼロ係数テーブル及び前記非ゼロ変数テーブルから取得し、取得した前記第 2 係数と前記第 2 変数とを前記積和回路に入力する

前記（１）に記載のニューラルネットワーク処理装置。

（７）

前記復号部は、

それぞれ優先順位が設定された複数の入力を有し、前記第 2 値が入力された 1 以上の前記入力のうち最も前記優先順位が高い前記入力に設定された値を出力するプライオリティエンコーダを備え、

前記復号部は、

前記複数の入力に対して前記第 1 ゼロ係数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記第 1 非ゼロ係数テーブルから前記第 2 係数を取得し、

前記複数の入力に対して前記第 1 ゼロ変数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記非ゼロ変数テーブルから前記第 2 変数を取得する

前記（6）に記載のニューラルネットワーク処理装置。

（8）

前記積和回路から出力された第 2 変数行列を、当該第 2 変数行列における値がゼロの第 1 変数の位置を前記第 1 値で示し、前記第 2 変数行列における値がゼロでない第 2 変数の位置を前記第 2 値で示す第 2 ゼロ変数位置テーブルと、前記第 2 変数行列における前記第 2 係数を保持する第 2 非ゼロ係数テーブルとに符号化する符号化部をさらに備える

前記（1）～（7）の何れか 1 つに記載のニューラルネットワーク処理装置。

（9）

前記符号化部は、

入力された値がゼロであるか否かを判定する判定回路と、

前記判定回路により値がゼロであると判定された場合に前記第 1 値を格納し、ゼロでないと判定された場合に前記第 2 値を格納する第 1 バッファと、

前記判定回路により値がゼロでないと判定された場合に前記第 2 変数を格納する第 2 バッファと、

を備え、

前記符号化部は、前記判定回路に前記第 2 変数行列を構成する変数を順に
入力する

前記（8）に記載のニューラルネットワーク処理装置。

（10）

前記符号化部は、次レイヤの畳み込み処理において使用される第 2 ゼロ係
数位置テーブルが格納されるレジスタをさらに備え、

前記第 1 バッファは、前記判定回路によりゼロでないと判定された変数の
前記第 2 変数行列上の位置に対応する前記第 2 ゼロ係数位置テーブル上の位
置に格納された値が前記第 1 値である場合、前記第 2 値に代えて前記第 1 値
を格納し、

前記第 2 バッファは、前記判定回路によりゼロでないと判定された前記変
数のうち、当該変数の前記第 2 変数行列上の位置に対応する前記第 2 ゼロ係
数位置テーブル上の位置に格納された値が前記第 1 値である場合、当該変数
を格納しない

前記（9）に記載のニューラルネットワーク処理装置。

（11）

前記レジスタは、前記次レイヤの畳み込み処理において使用される複数の
前記第 2 ゼロ係数位置テーブルを論理和することで得られた第 3 ゼロ係数位
置テーブルを格納する

前記（10）に記載のニューラルネットワーク処理装置。

（12）

前記レジスタは、前記次レイヤの畳み込み処理において使用される複数の
前記第 2 ゼロ係数位置テーブルを格納する

前記（10）に記載のニューラルネットワーク処理装置。

（13）

前記次レイヤの畳み込み処理において使用される複数の前記第 2 ゼロ係数
位置テーブルは、1 以上のグループにグループ化され、

前記レジスタは、前記グループごとに前記第 2 ゼロ係数位置テーブルを論

理和することで得られた第3ゼロ係数位置テーブルを格納する

前記(10)に記載のニューラルネットワーク処理装置。

(14)

前記(1)～(13)の何れか1つに記載のニューラルネットワーク処理装置と、

前記ニューラルネットワーク処理装置とバスを介して接続されたプロセッサコアと、

を備える情報処理装置。

(15)

請求項14に記載の情報処理装置と、

前記情報処理装置に接続された1以上のセンサと、

を備える情報処理システム。

(16)

前記(14)に記載の情報処理装置を備える電子機器。

(17)

係数行列における値がゼロの第1係数の位置を第1値で示し、前記係数行列における値がゼロでない第2係数の位置を第2値で示すゼロ係数位置テーブルと、前記係数行列における前記第2係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号し、

復号された前記係数行列と、変数行列との畳み込み処理を実行する

ことを含み、

前記係数行列は、前記ゼロ係数位置テーブルにおける前記第2値で示された位置に前記非ゼロ係数テーブルに格納された前記第2係数を格納することで復号される

ニューラルネットワーク処理方法。

(18)

コンピュータに、

係数行列における値がゼロの第1係数の位置を第1値で示し、前記係数行

列における値がゼロでない第2係数の位置を第2値で示すゼロ係数位置テーブルと、前記係数行列における前記第2係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号する復号処理と、

前記復号処理により復号された前記係数行列と、変数行列との畳み込み処理を実行する積和处理と、

を実行させ、

前記復号処理は、前記ゼロ係数位置テーブルにおける前記第2値で示された位置に前記非ゼロ係数テーブルに格納された前記第2係数を格納することで、前記係数行列を復号する

プログラム。

符号の説明

- [0422] 1 1 ニューラルネットワーク処理装置
- 2 1、2 3、2 5 畳み込み処理部
- 2 2、2 4 プーリング処理部
- 4 1、6 1、8 1、1 0 1 復号部
- 4 2、5 1、6 2、7 1、8 2、1 0 2 メモリ
- 4 3、6 3、8 3、1 0 3 係数保持部
- 1 0 4 プライオリティエンコーダ
- 1 1 1、1 1 1 a、1 1 1 b アドレス生成部
- 1 1 2、2 0 3 セレクタ
- 1 1 3 積和器
- 1 1 4 係数バッファ
- 1 1 5 変数バッファ
- 1 1 6 積和回路
- 2 0 0、3 0 0 符号化部
- 2 0 1 バッファ回路
- 2 0 2 判定回路
- 2 0 4 遅延回路

205 スパース行列バッファ

206 書込みバッファ

301、302 AND回路

303 レジスタ

304 メモリ領域

1000、1000A、1000a、1000b、1000c 情報処理
装置（プロセッサ）

1001、1001a、1001b、1001c DNNエンジン（アク
セラレータ）

1002 プロセッサコア

1003 バス

1100 IoTエッジAIシステム

1111～1114 センサ

1200 自律ロボット

1211 イメージセンサ

1212 マイクロホン

1201、1202、1203、1301、1302 処理部

1204 制御部

1300 テレビジョン

請求の範囲

[請求項1]

第1係数行列における値がゼロの第1係数の位置を第1値で示し、前記第1係数行列における値がゼロでない第2係数の位置を第2値で示す第1ゼロ係数位置テーブルと、前記第1係数行列における前記第2係数を保持する第1非ゼロ係数テーブルとに符号化された前記第1係数行列を復号する復号部と、

前記復号部により復号された前記第1係数行列と、第1変数行列との畳み込み処理を実行する積和回路と、

を備え、

前記復号部は、前記第1ゼロ係数位置テーブルにおける前記第2値で示された位置に前記第1非ゼロ係数テーブルに格納された前記第2係数を格納することで、前記第1係数行列を復号する

ニューラルネットワーク処理装置。

[請求項2]

前記復号部は、

制御端子に入力された値が前記第1値である場合、ゼロを出力し、前記第2値である場合、当該第2値の前記第1ゼロ係数位置テーブル上の位置に対応する前記第2係数を出力するセレクタと、

前記セレクタから入力された前記ゼロと前記第2係数とを配列することで前記第1係数行列を復元する積和器と、

を備え、

前記復号部は、前記第1ゼロ係数位置テーブルを構成する値を順に前記セレクタに入力する

請求項1に記載のニューラルネットワーク処理装置。

[請求項3]

前記復号部は、前記第1ゼロ係数位置テーブルを構成する値を順に前記セレクタに入力する際に、前記セレクタに入力する前記値の前記第1ゼロ係数位置テーブル上の位置に対応する前記第1変数行列上の位置に格納された変数を取得する

請求項2に記載のニューラルネットワーク処理装置。

[請求項4] 前記復号部は、前記第2値の前記第1ゼロ係数位置テーブル上の位置に対応する前記第2係数を取得するとともに、前記第2値の前記第1ゼロ係数位置テーブル上の位置に対応する前記第1変数行列上の位置に格納された変数を取得し、取得した前記第2係数と前記変数とを前記積和回路に入力し、

前記積和回路は、前記復号部から入力された前記第2係数と前記変数とを順に乗算して加算することで、前記第1係数行列と前記第1変数行列との前記畳み込み処理を実行する

請求項1に記載のニューラルネットワーク処理装置。

[請求項5] 前記復号部は、それぞれ優先順位が設定された複数の入力を有し、前記第2値が入力された1以上の前記入力のうち最も前記優先順位が高い前記入力に設定された値を出力するプライオリティエンコーダを備え、

前記復号部は、前記複数の入力に対して前記第1ゼロ係数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記第1非ゼロ係数テーブルから前記第2係数を取得する

請求項4に記載のニューラルネットワーク処理装置。

[請求項6] 前記第1変数行列は、当該第1変数行列における値がゼロの第1変数の位置を前記第1値で示し、前記第1変数行列における値がゼロでない第2変数の位置を前記第2値で示す第1ゼロ変数位置テーブルと、前記第1変数行列における前記第2変数を保持する非ゼロ変数テーブルとに符号化されており、

前記復号部は、前記第1ゼロ係数位置テーブルを構成する値と前記第1ゼロ変数位置テーブルを構成する値とを論理演算し、当該論理演算の結果に基づいて、互いに乗算した結果がゼロとならない前記第2係数と前記第2変数とを前記第1非ゼロ係数テーブル及び前記非ゼロ変数テーブルから取得し、取得した前記第2係数と前記第2変数とを

前記積和回路に入力する

請求項 1 に記載のニューラルネットワーク処理装置。

[請求項7]

前記復号部は、

それぞれ優先順位が設定された複数の入力を有し、前記第 2 値が入力された 1 以上の前記入力のうち最も前記優先順位が高い前記入力に設定された値を出力するプライオリティエンコーダを備え、

前記復号部は、

前記複数の入力に対して前記第 1 ゼロ係数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記第 1 非ゼロ係数テーブルから前記第 2 係数を取得し、

前記複数の入力に対して前記第 1 ゼロ変数位置テーブルを構成する値を並列に入力し、当該入力に対して前記プライオリティエンコーダから出力された値に基づいて前記非ゼロ変数テーブルから前記第 2 変数を取得する

請求項 6 に記載のニューラルネットワーク処理装置。

[請求項8]

前記積和回路から出力された第 2 変数行列を、当該第 2 変数行列における値がゼロの第 1 変数の位置を前記第 1 値で示し、前記第 2 変数行列における値がゼロでない第 2 変数の位置を前記第 2 値で示す第 2 ゼロ変数位置テーブルと、前記第 2 変数行列における前記第 2 係数を保持する第 2 非ゼロ係数テーブルとに符号化する符号化部をさらに備える

請求項 1 に記載のニューラルネットワーク処理装置。

[請求項9]

前記符号化部は、

入力された値がゼロであるか否かを判定する判定回路と、

前記判定回路により値がゼロであると判定された場合に前記第 1 値を格納し、ゼロでないと判定された場合に前記第 2 値を格納する第 1 バッファと、

前記判定回路により値がゼロでないと判定された場合に前記第2変数を格納する第2バッファと、

を備え、

前記符号化部は、前記判定回路に前記第2変数行列を構成する変数を順に入力する

請求項8に記載のニューラルネットワーク処理装置。

[請求項10]

前記符号化部は、次レイヤの畳み込み処理において使用される第2ゼロ係数位置テーブルが格納されるレジスタをさらに備え、

前記第1バッファは、前記判定回路によりゼロでないと判定された変数の前記第2変数行列上の位置に対応する前記第2ゼロ係数位置テーブル上の位置に格納された値が前記第1値である場合、前記第2値に代えて前記第1値を格納し、

前記第2バッファは、前記判定回路によりゼロでないと判定された前記変数のうち、当該変数の前記第2変数行列上の位置に対応する前記第2ゼロ係数位置テーブル上の位置に格納された値が前記第1値である場合、当該変数を格納しない

請求項9に記載のニューラルネットワーク処理装置。

[請求項11]

前記レジスタは、前記次レイヤの畳み込み処理において使用される複数の前記第2ゼロ係数位置テーブルを論理和することで得られた第3ゼロ係数位置テーブルを格納する

請求項10に記載のニューラルネットワーク処理装置。

[請求項12]

前記レジスタは、前記次レイヤの畳み込み処理において使用される複数の前記第2ゼロ係数位置テーブルを格納する

請求項10に記載のニューラルネットワーク処理装置。

[請求項13]

前記次レイヤの畳み込み処理において使用される複数の前記第2ゼロ係数位置テーブルは、1以上のグループにグループ化され、

前記レジスタは、前記グループごとに前記第2ゼロ係数位置テーブルを論理和することで得られた第3ゼロ係数位置テーブルを格納する

請求項 10 に記載のニューラルネットワーク処理装置。

[請求項14]

請求項 1 に記載のニューラルネットワーク処理装置と、
前記ニューラルネットワーク処理装置とバスを介して接続されたプロセッサコアと、
を備える情報処理装置。

[請求項15]

請求項 14 に記載の情報処理装置と、
前記情報処理装置に接続された 1 以上のセンサと、
を備える情報処理システム。

[請求項16]

請求項 14 に記載の情報処理装置を備える電子機器。

[請求項17]

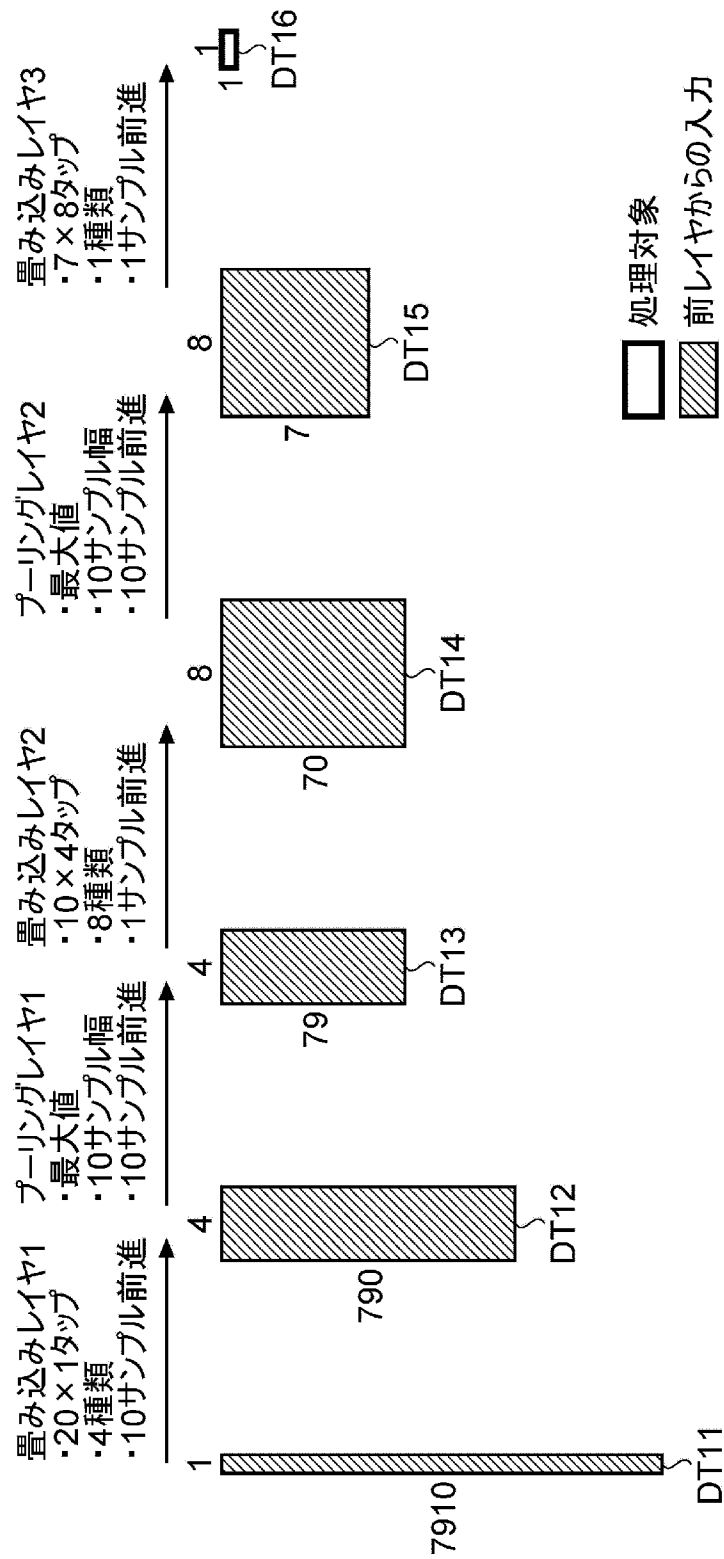
係数行列における値がゼロの第 1 係数の位置を第 1 値で示し、前記係数行列における値がゼロでない第 2 係数の位置を第 2 値で示すゼロ係数位置テーブルと、前記係数行列における前記第 2 係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号し、
復号された前記係数行列と、変数行列との畳み込み処理を実行することを含み、
前記係数行列は、前記ゼロ係数位置テーブルにおける前記第 2 値で示された位置に前記非ゼロ係数テーブルに格納された前記第 2 係数を格納することで復号される
ニューラルネットワーク処理方法。

[請求項18]

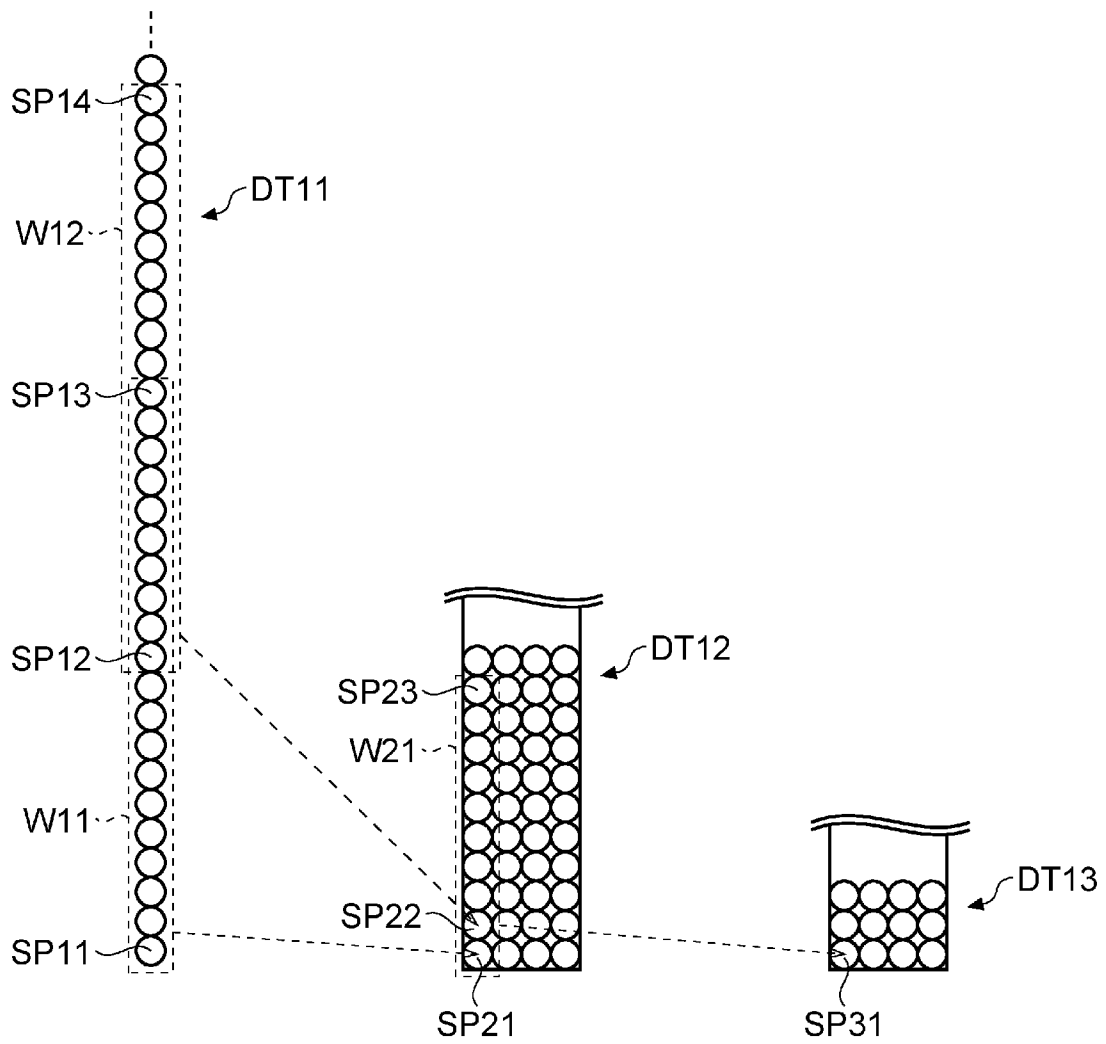
コンピュータに、
係数行列における値がゼロの第 1 係数の位置を第 1 値で示し、前記係数行列における値がゼロでない第 2 係数の位置を第 2 値で示すゼロ係数位置テーブルと、前記係数行列における前記第 2 係数を保持する非ゼロ係数テーブルとに符号化された前記係数行列を復号する復号処理と、
前記復号処理により復号された前記係数行列と、変数行列との畳み込み処理を実行する積和処理と、
を実行させ、

前記復号処理は、前記ゼロ係数位置テーブルにおける前記第 2 値で示された位置に前記非ゼロ係数テーブルに格納された前記第 2 係数を格納することで、前記係数行列を復号するプログラム。

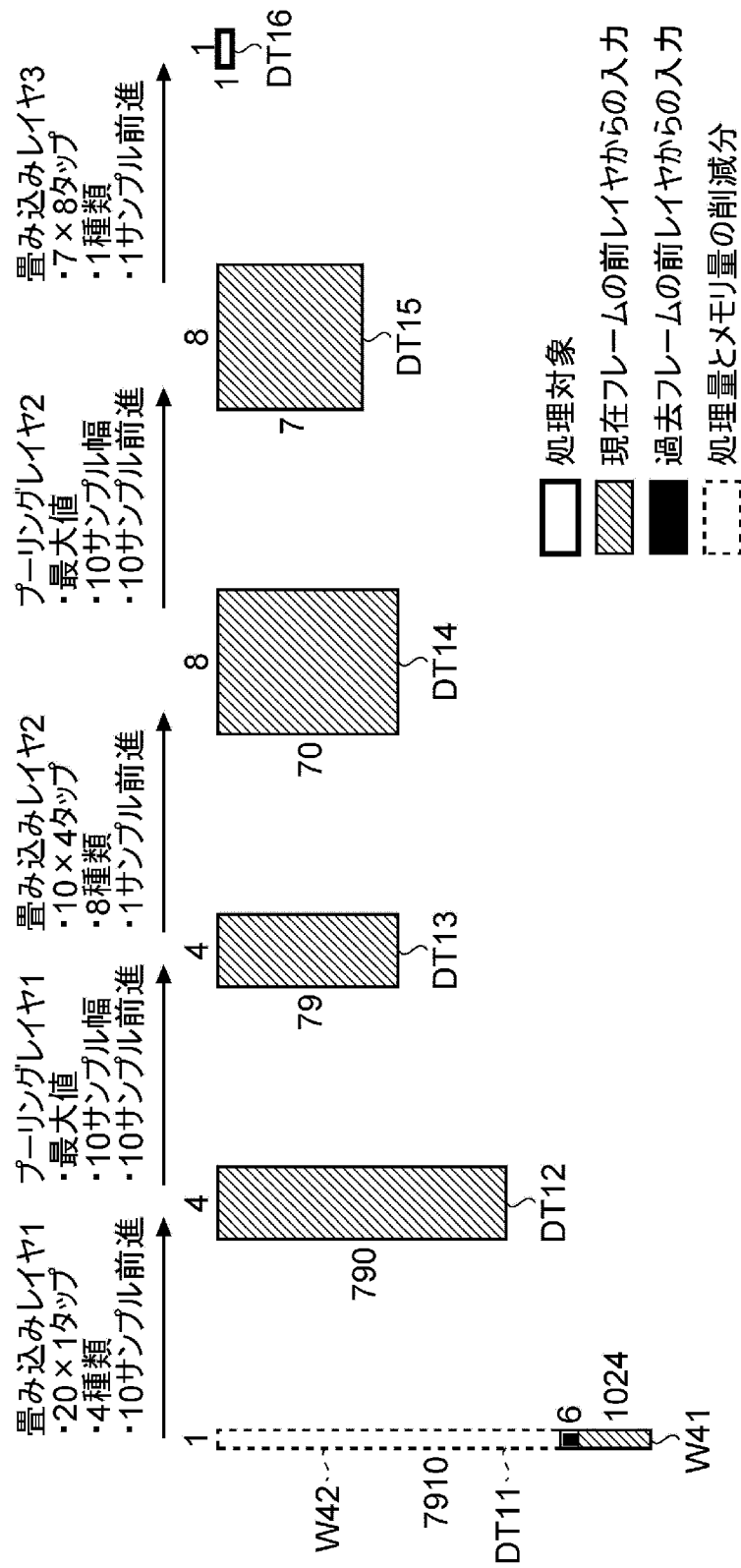
[図1]



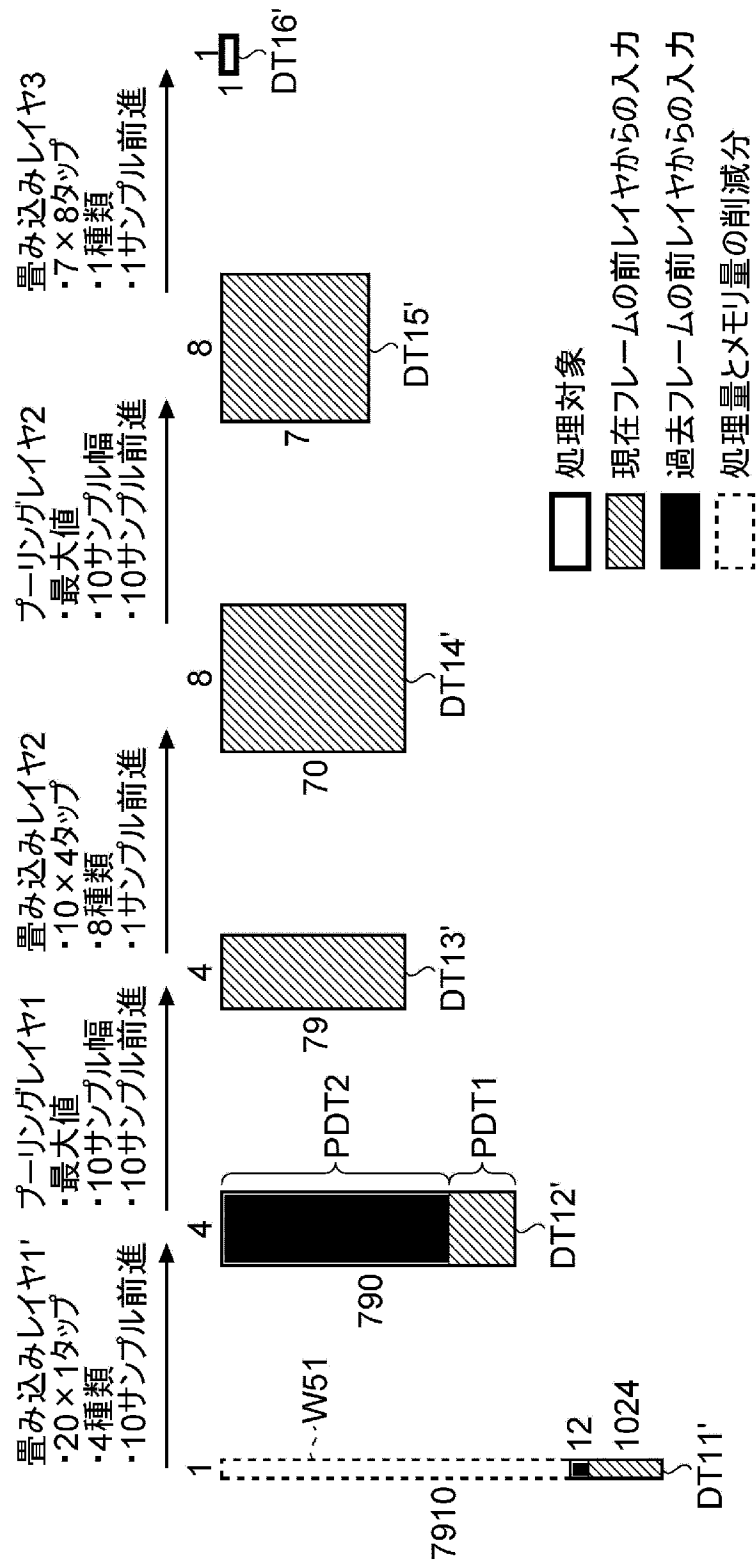
[図2]



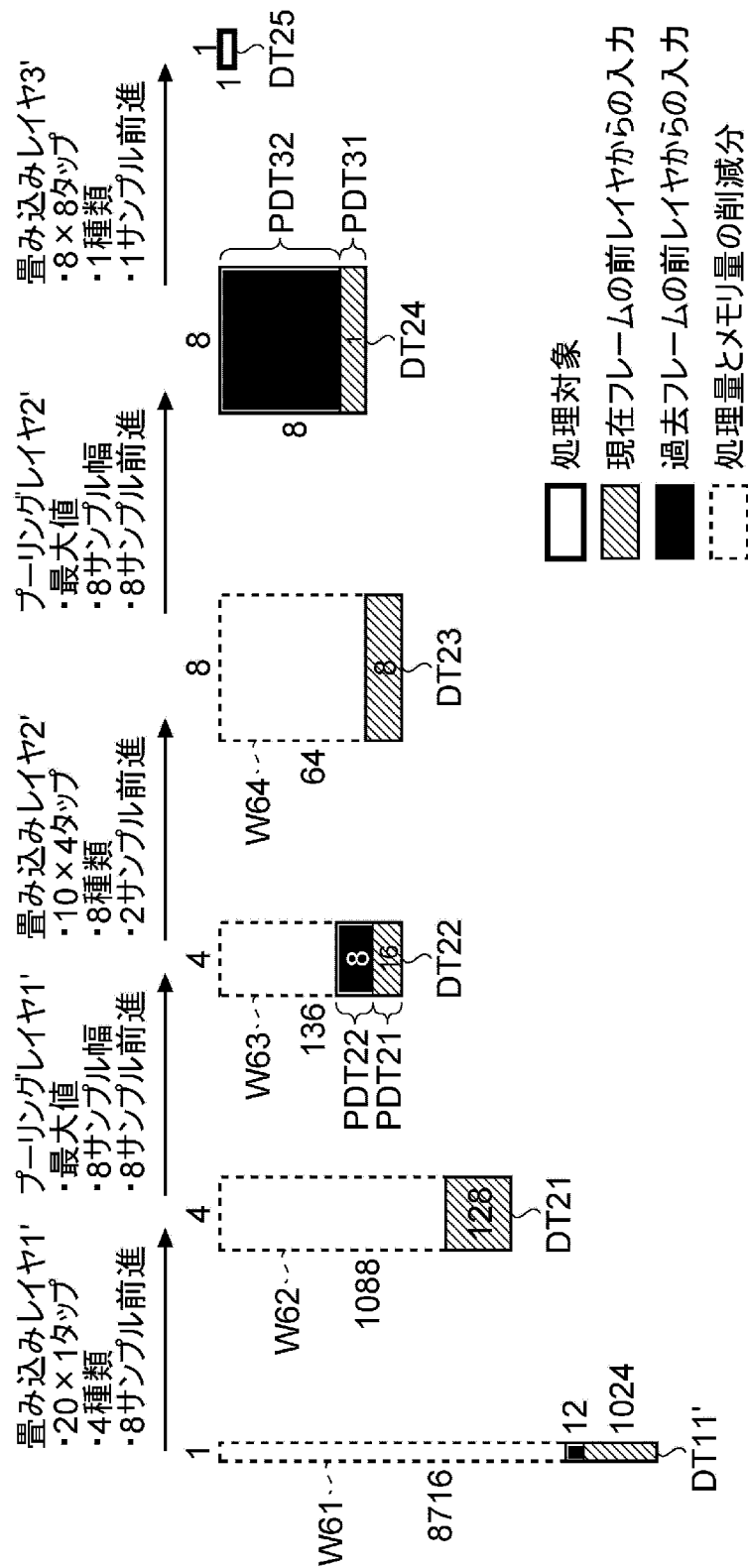
[図3]



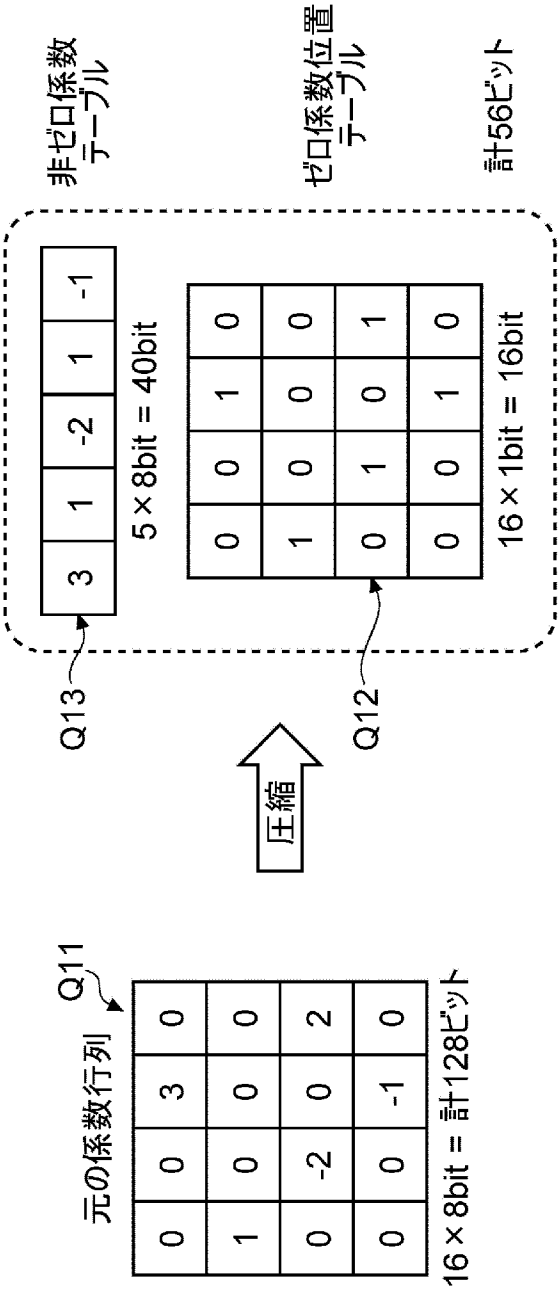
[図4]



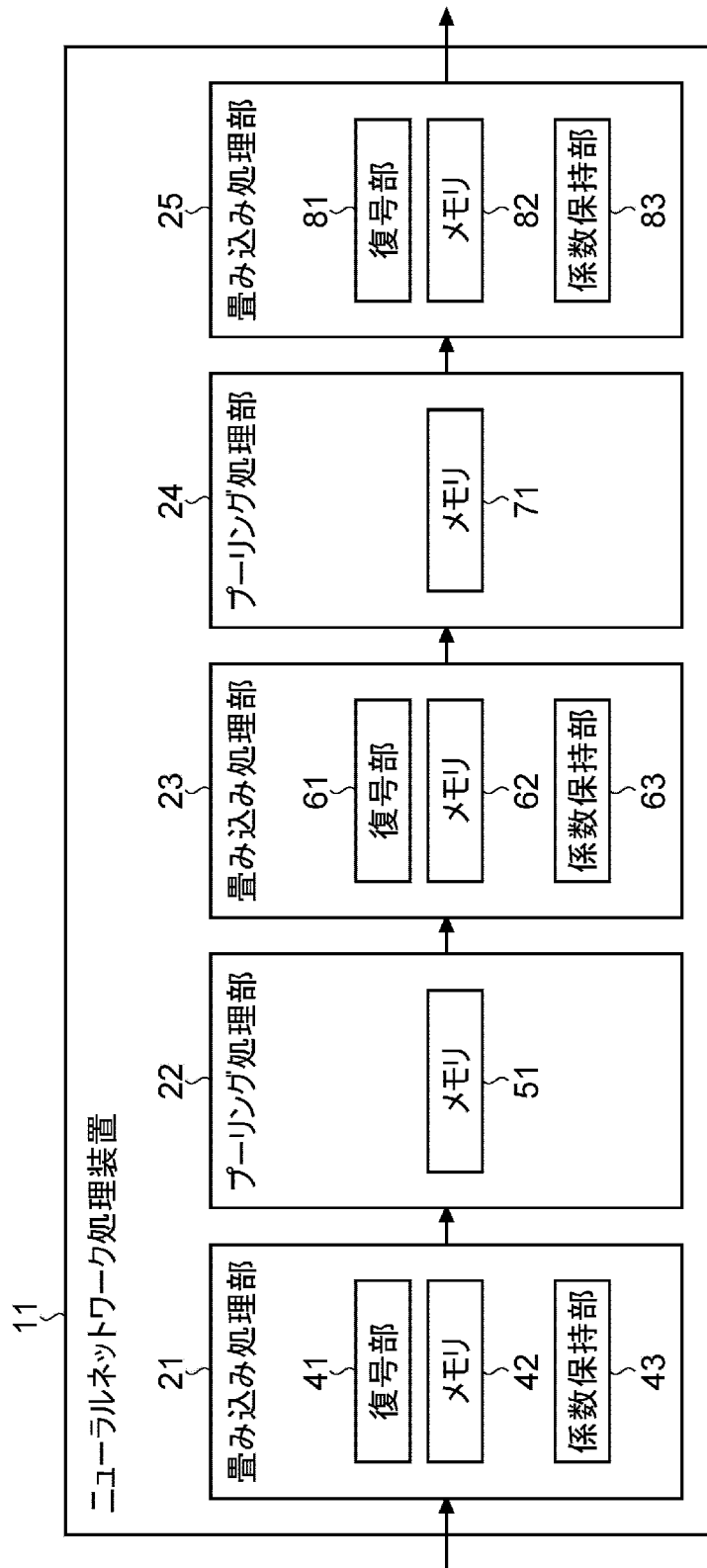
[図5]



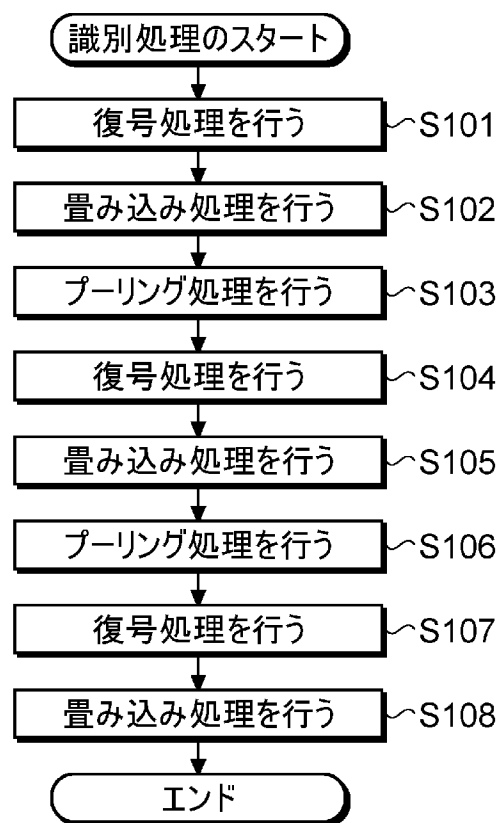
[図6]



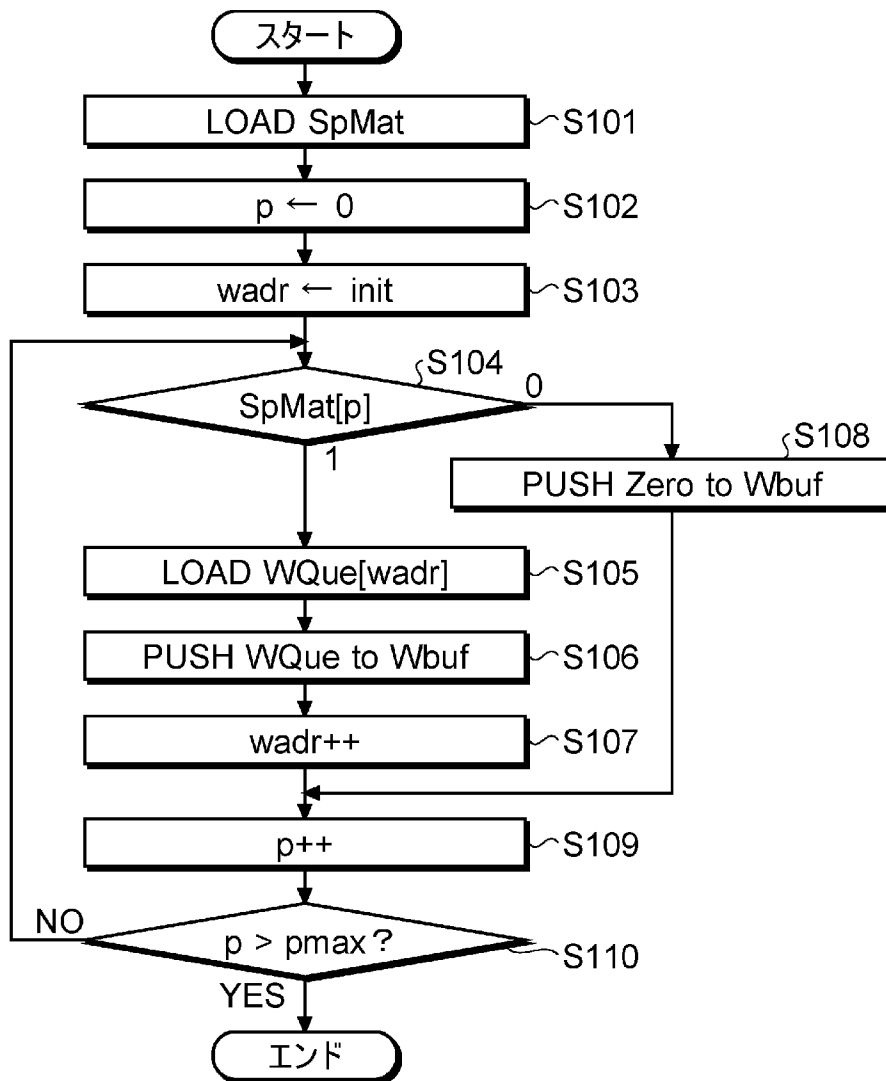
[図7]



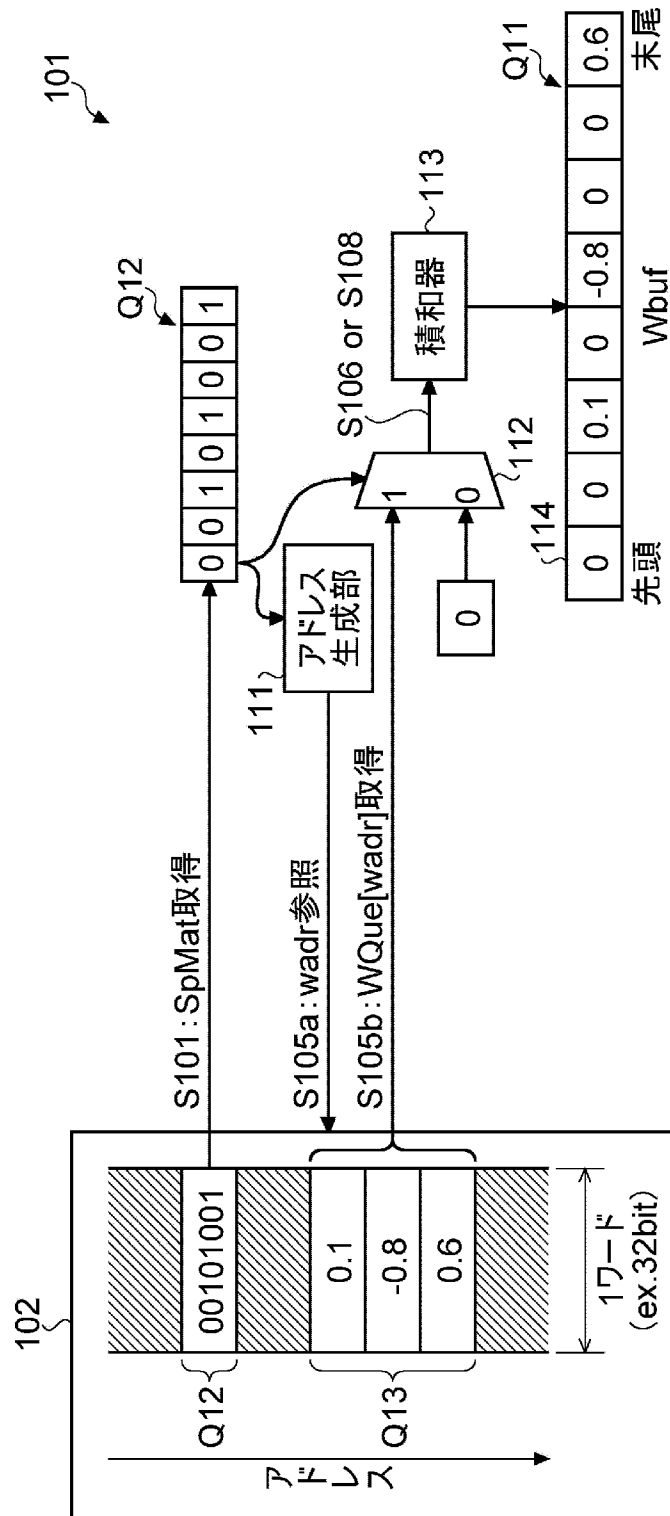
[図8]



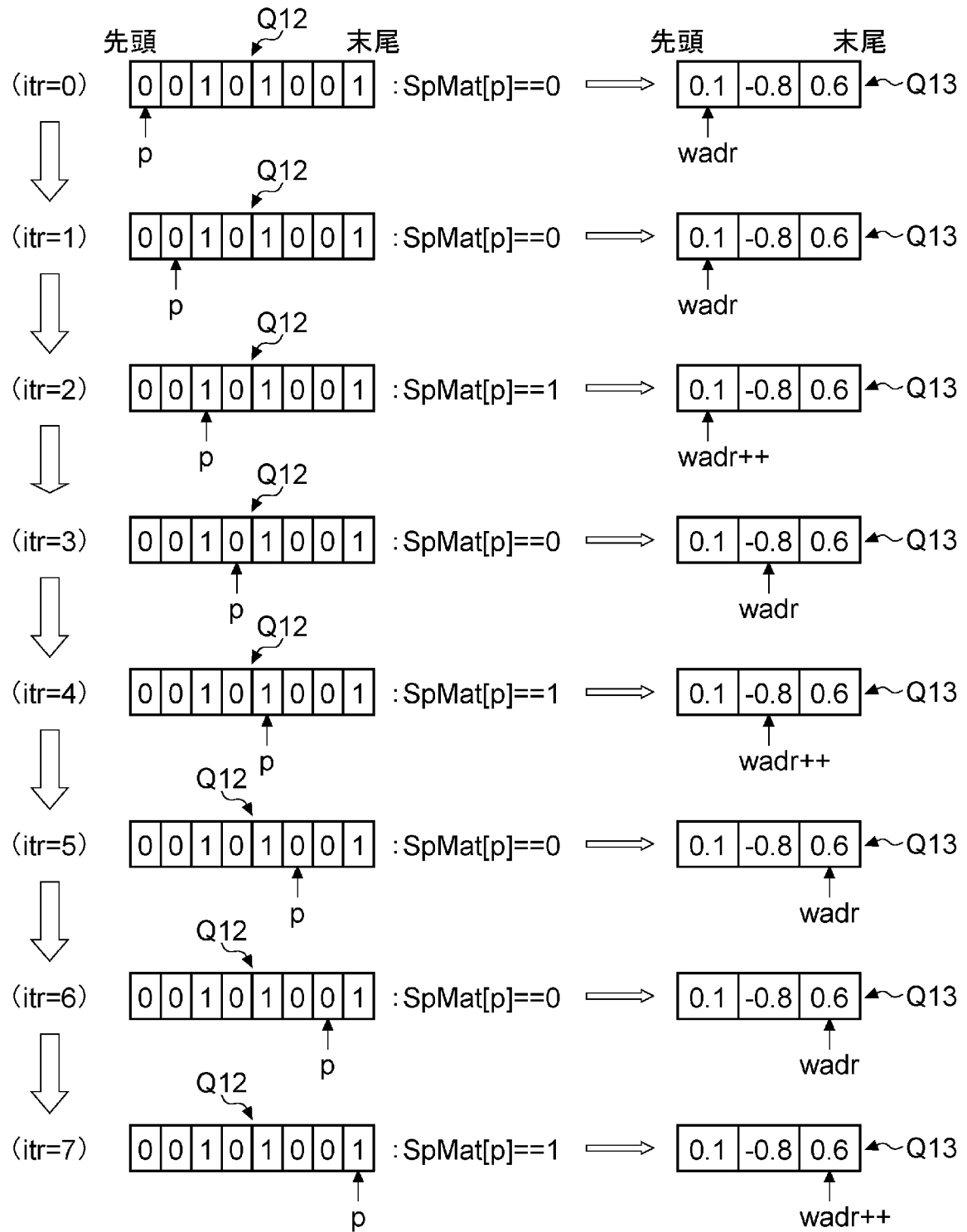
[図9]



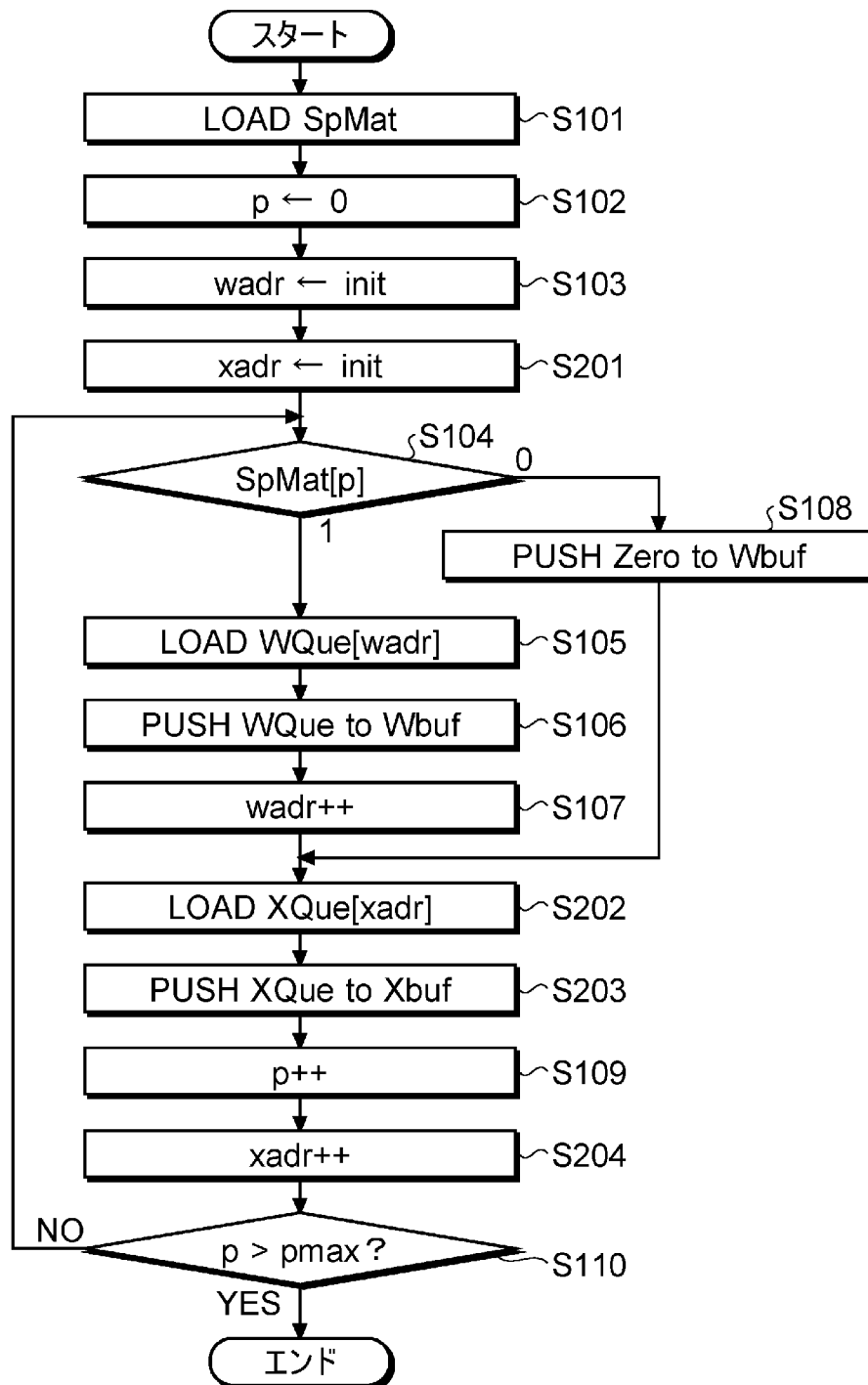
[図10]



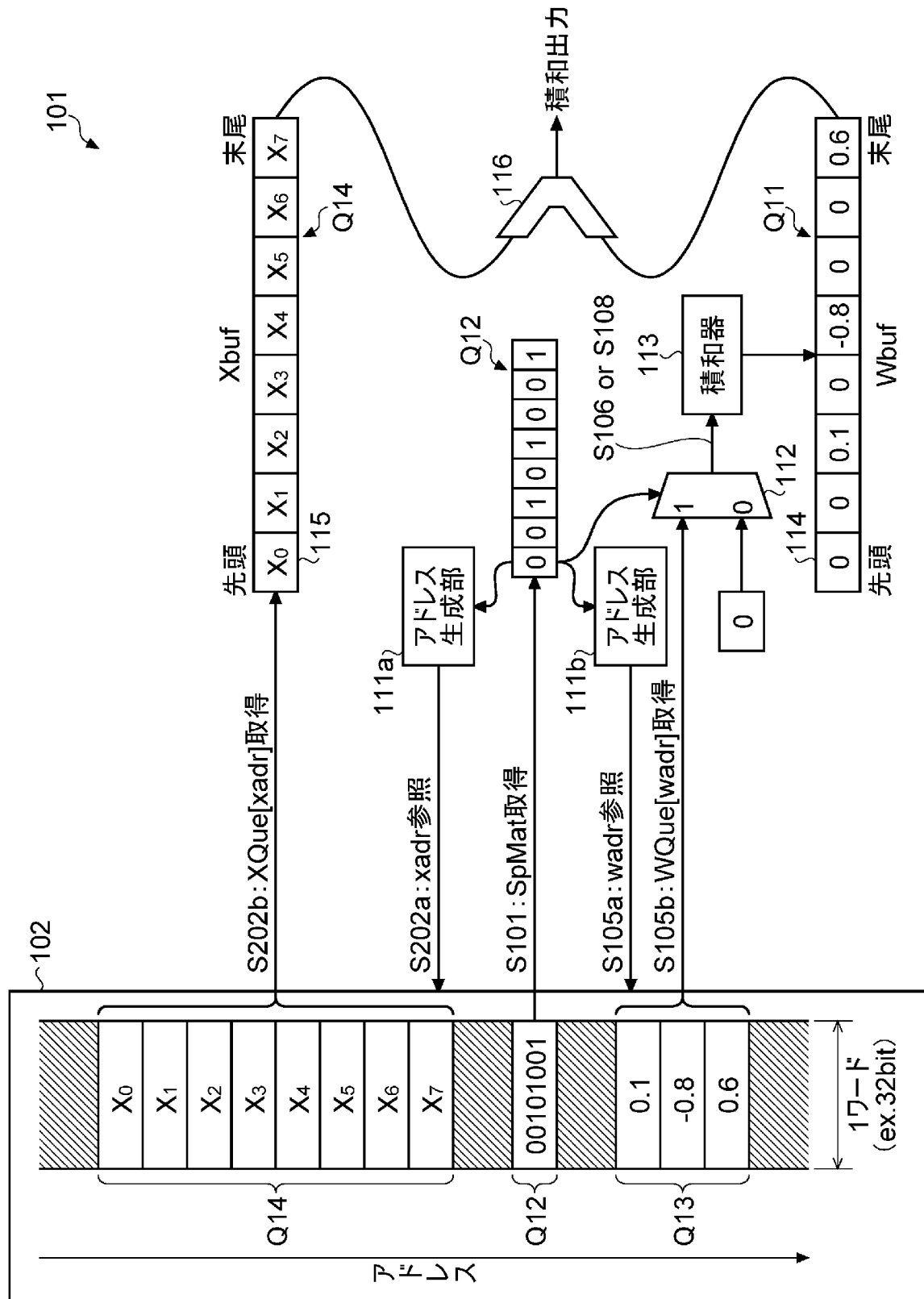
[図11]



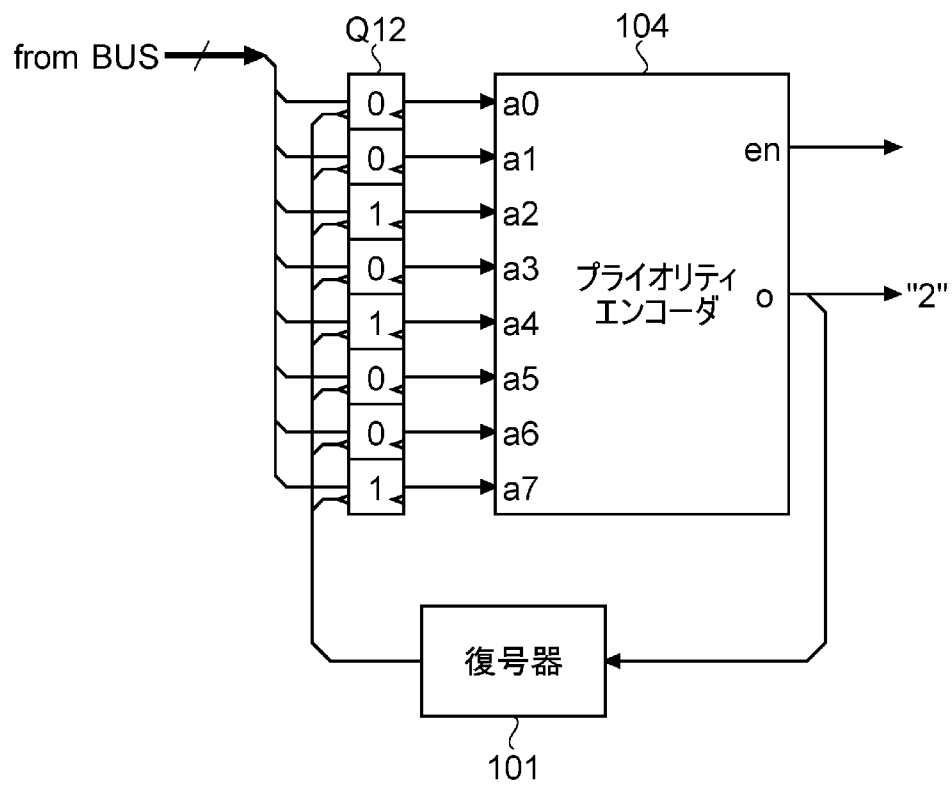
[図12]



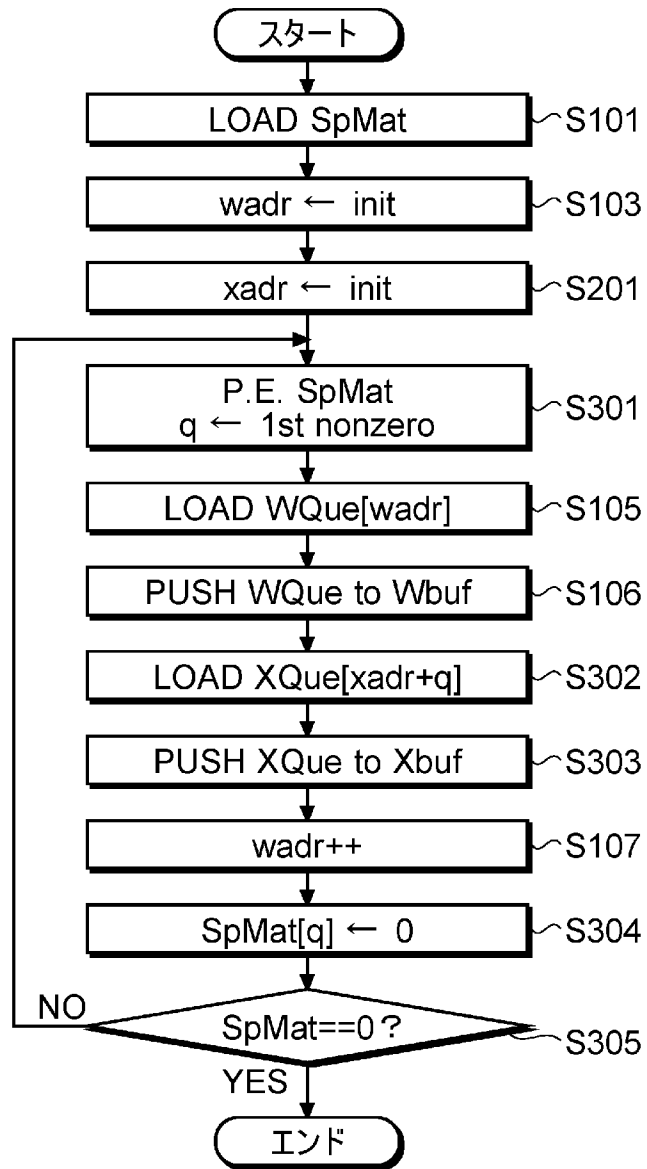
[図13]



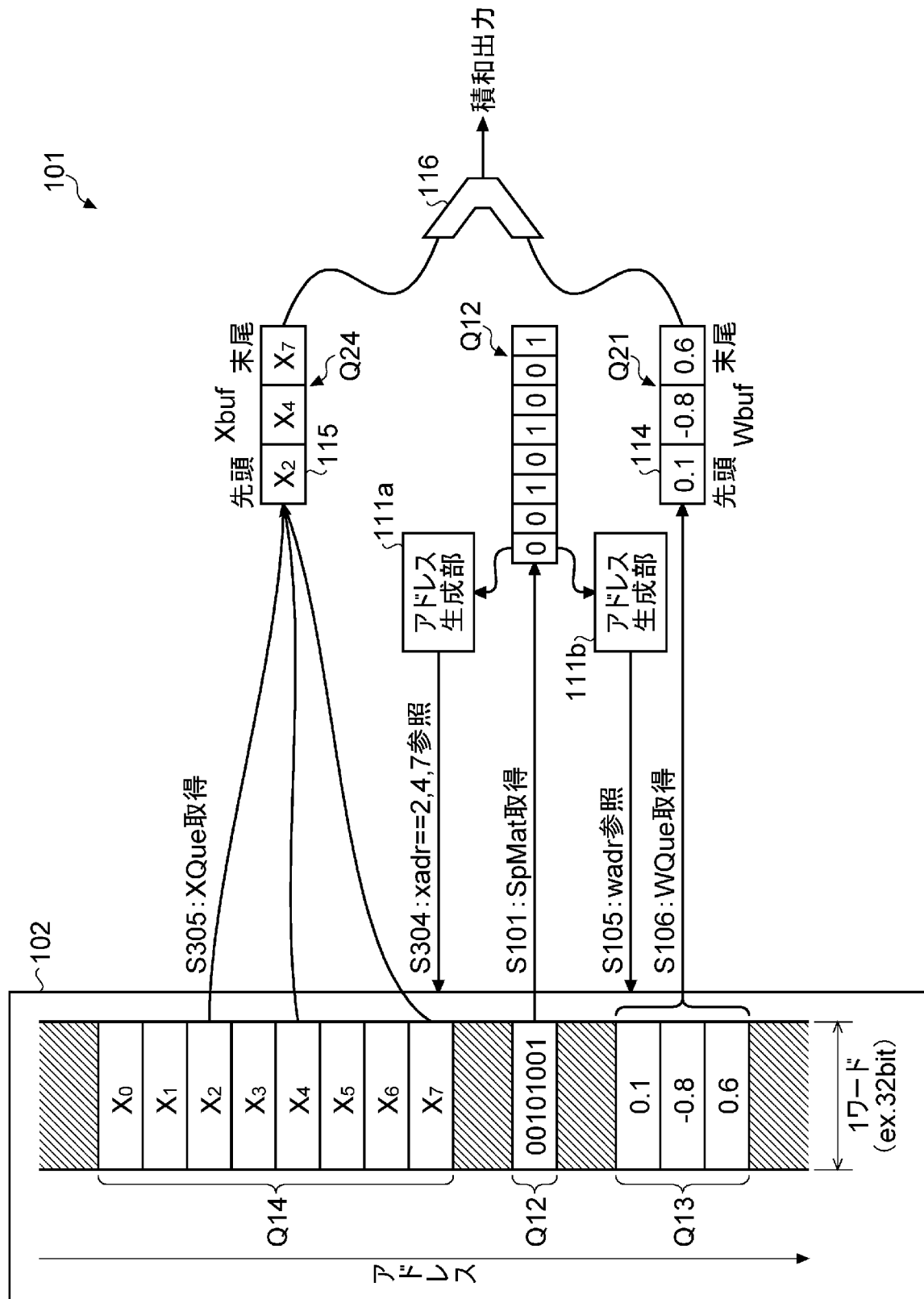
[図14]



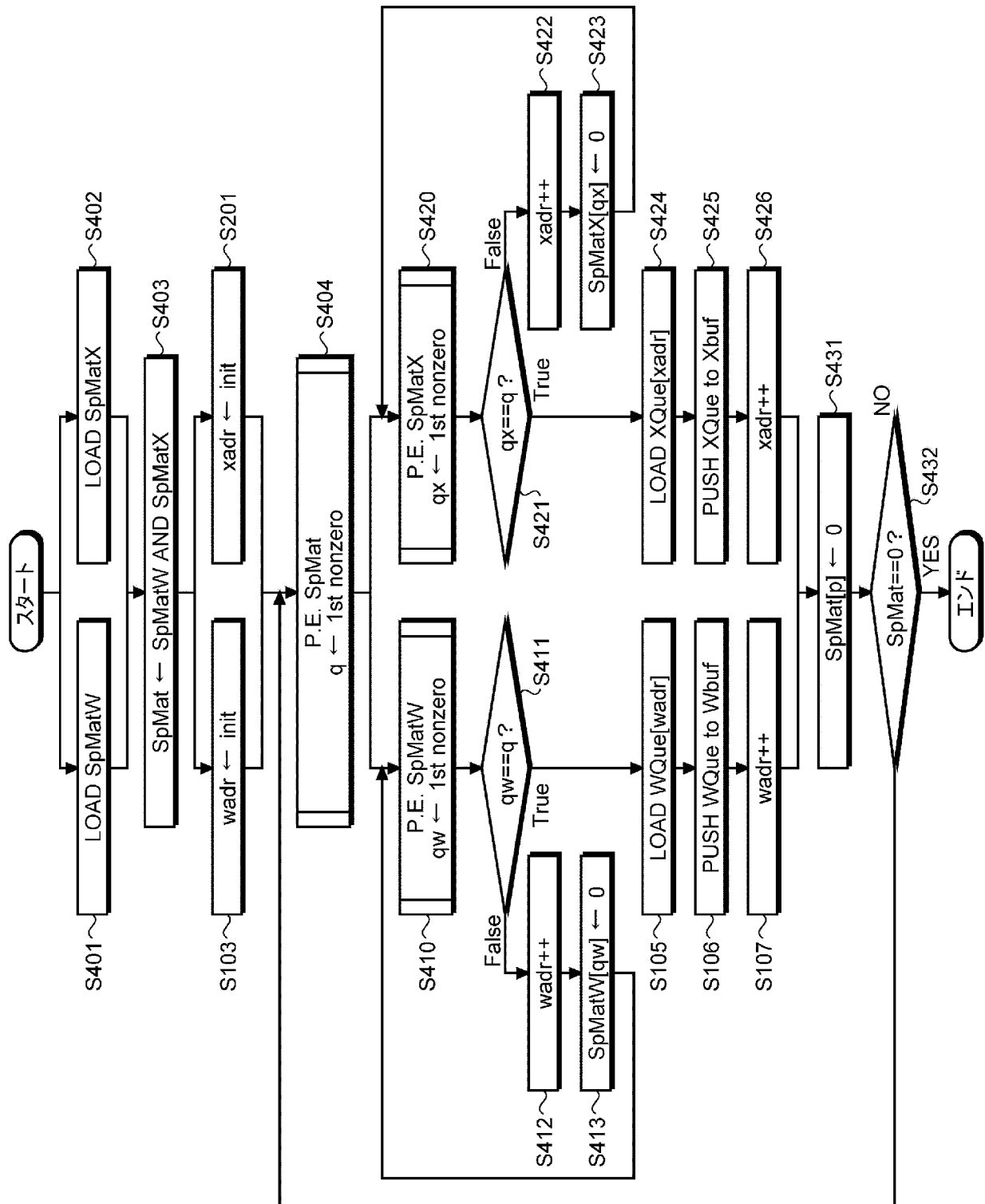
[図15]



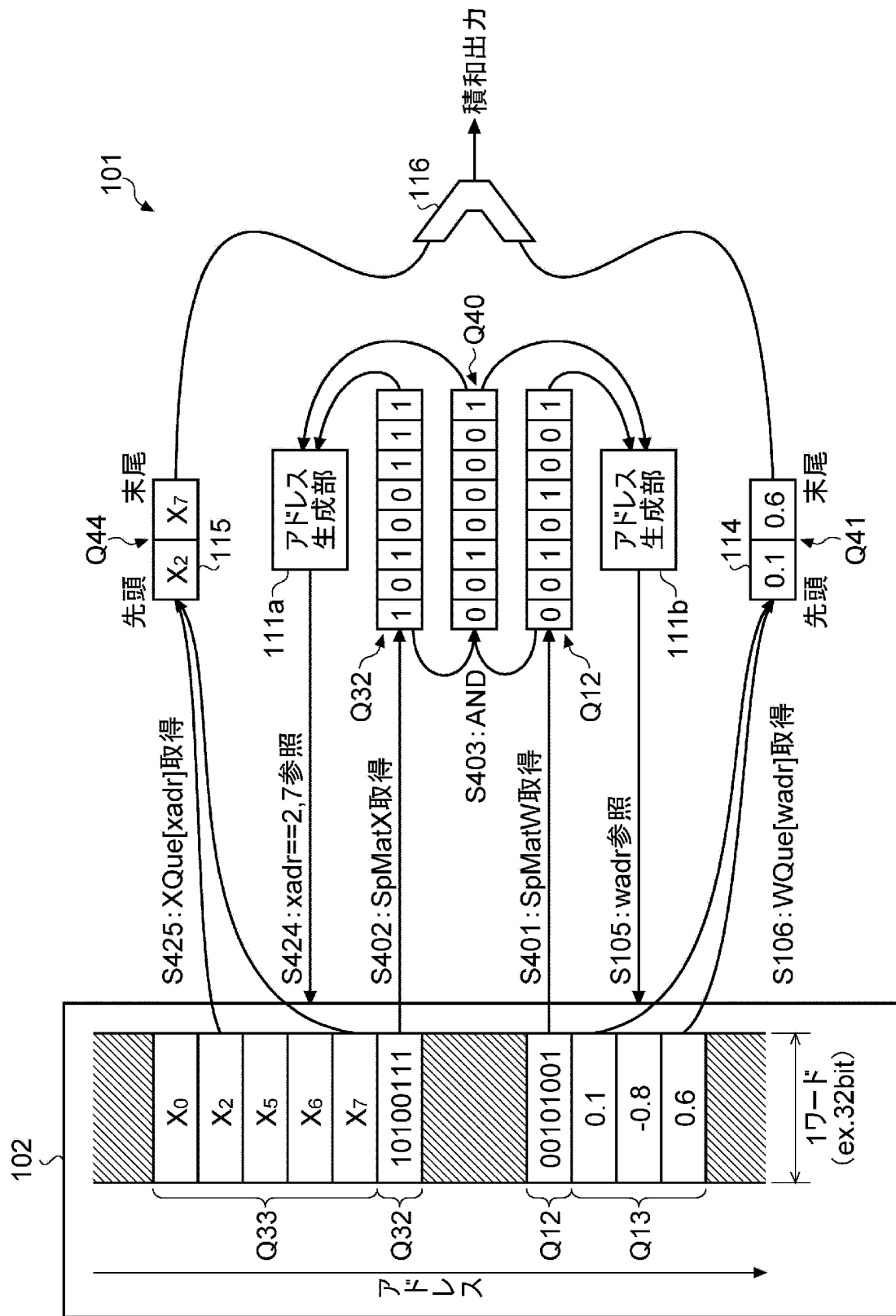
[図16]



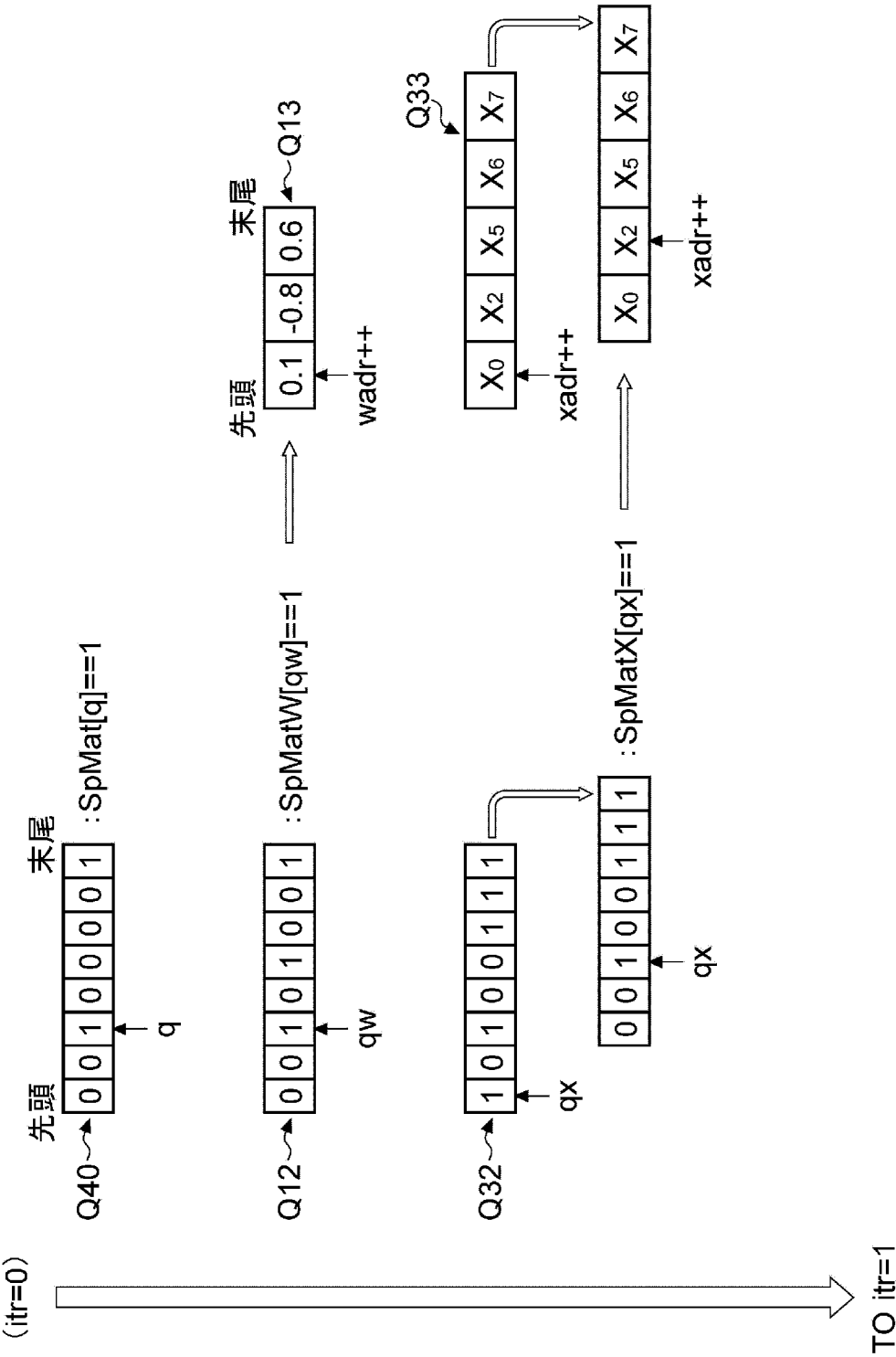
[図18]



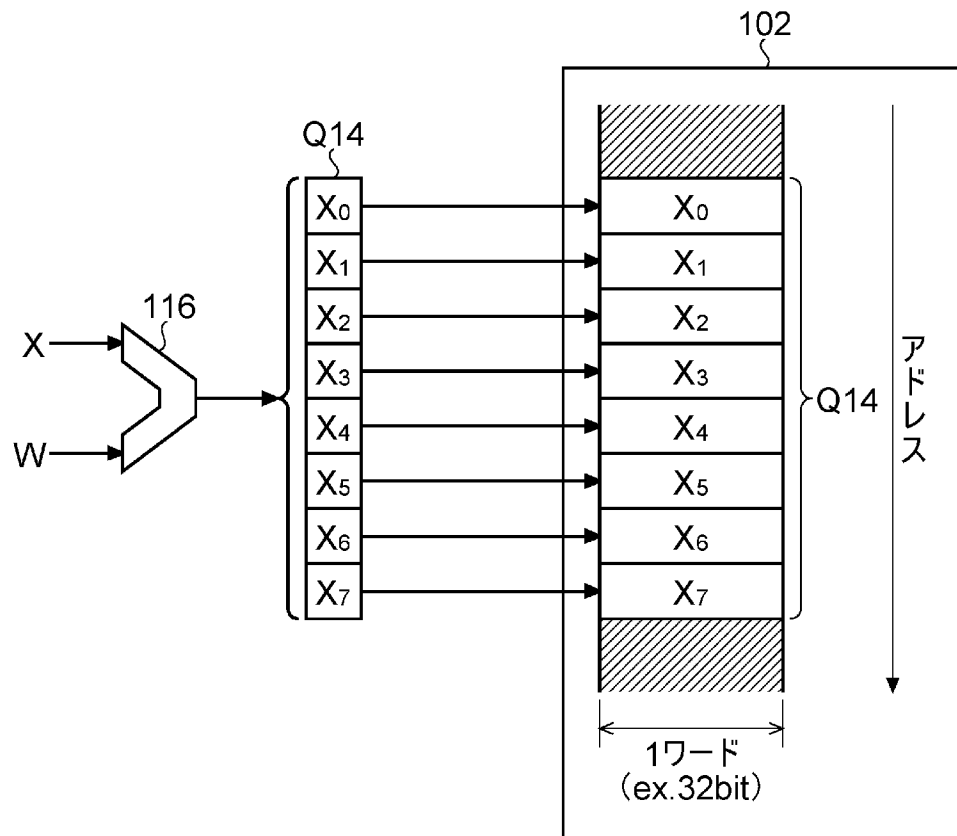
[図19]



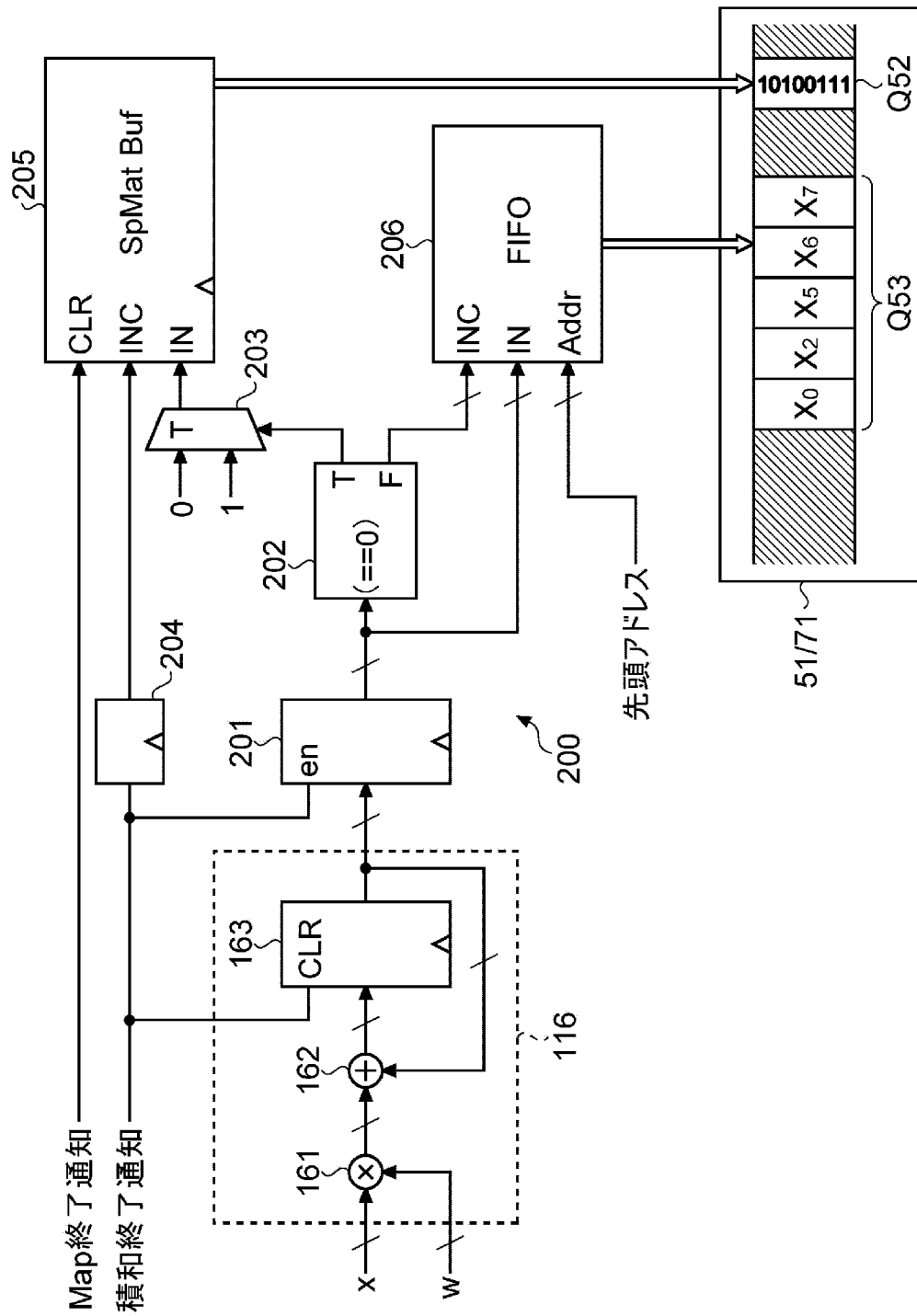
[図20]



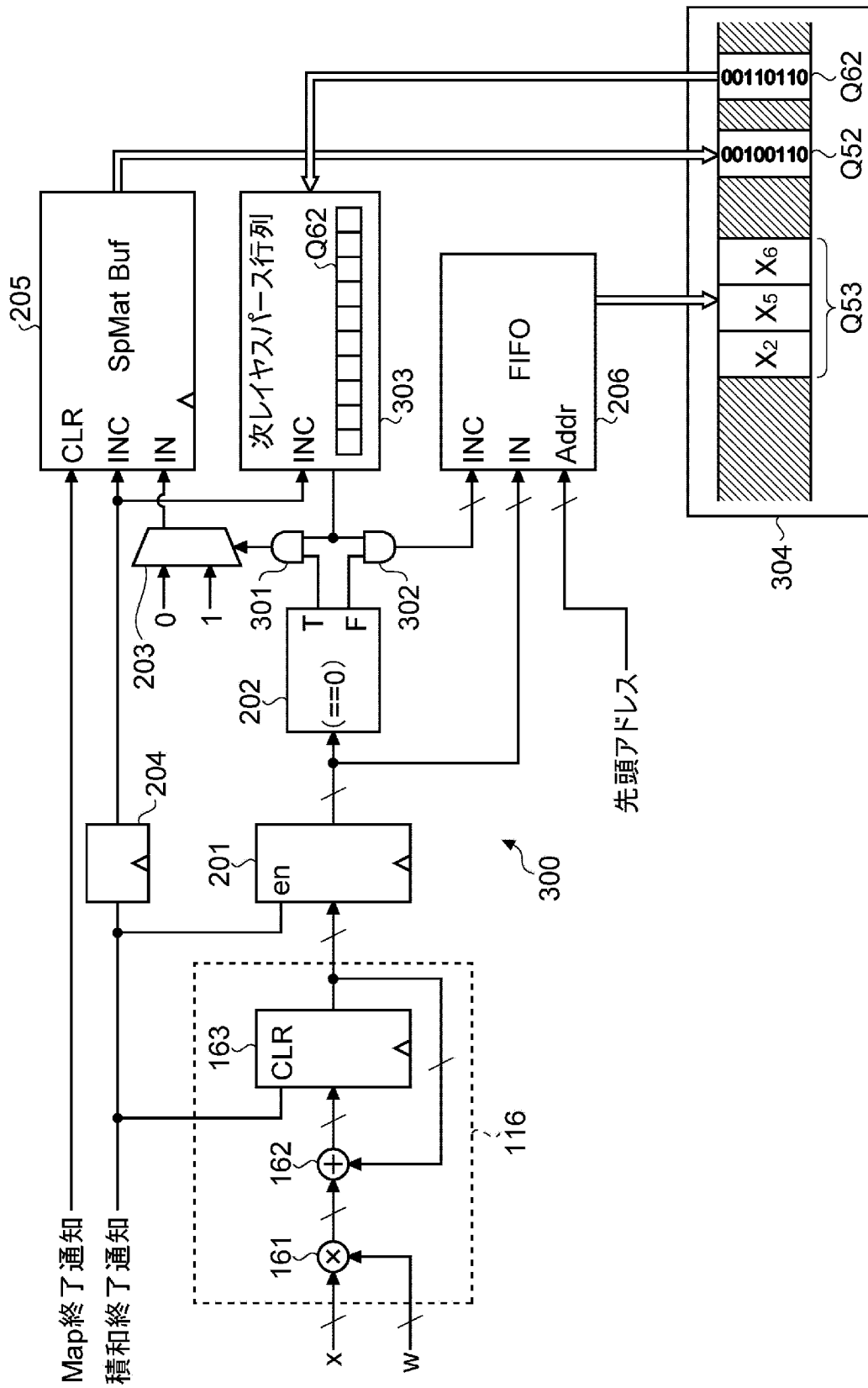
[図22]



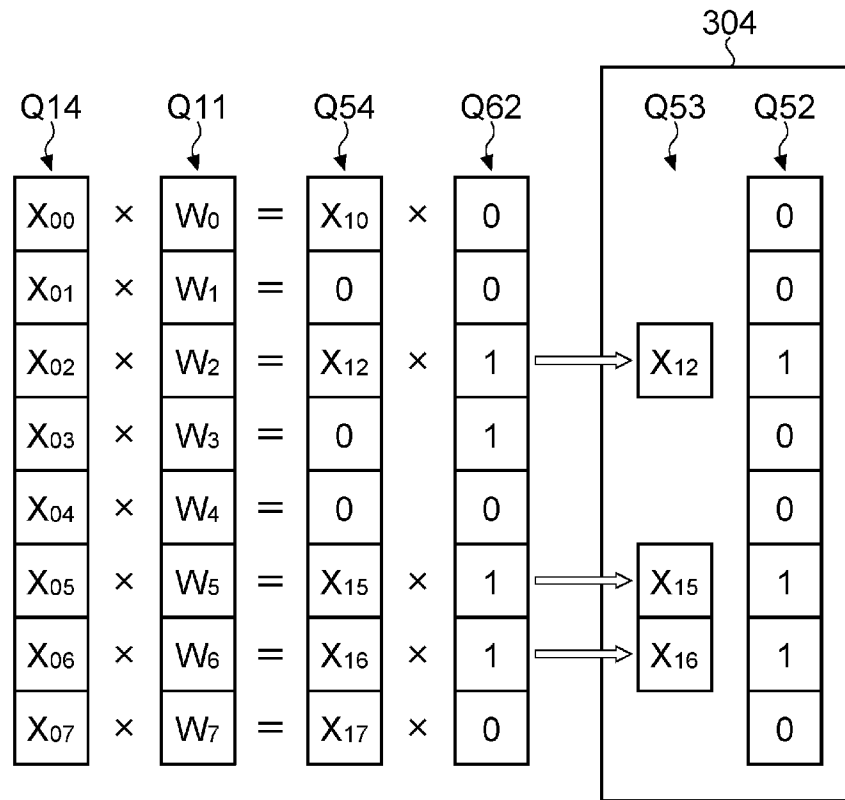
[図23]



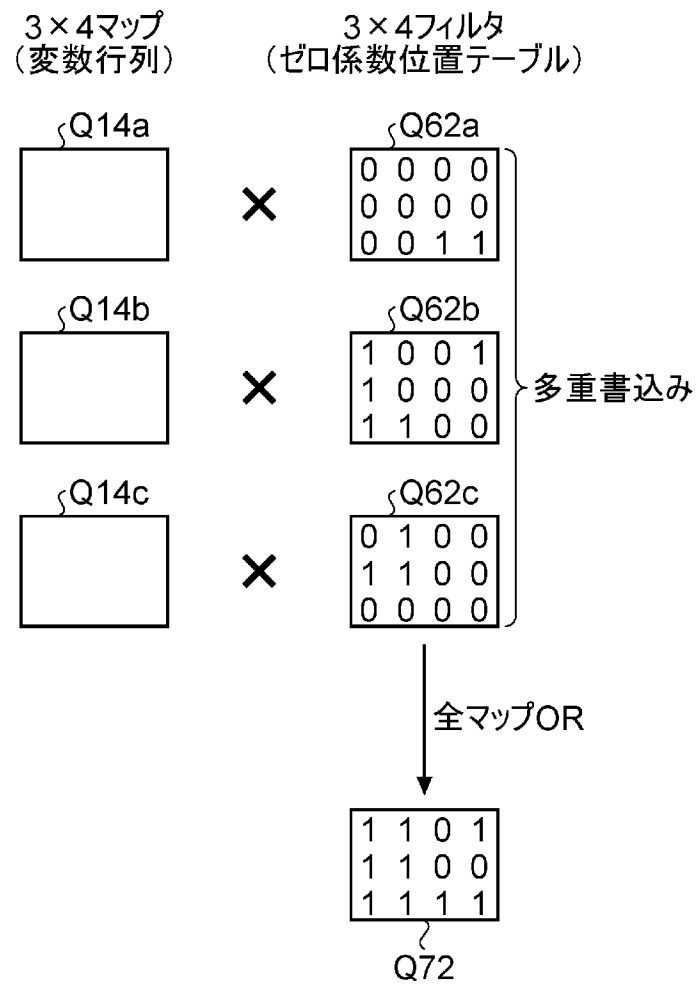
[図24]



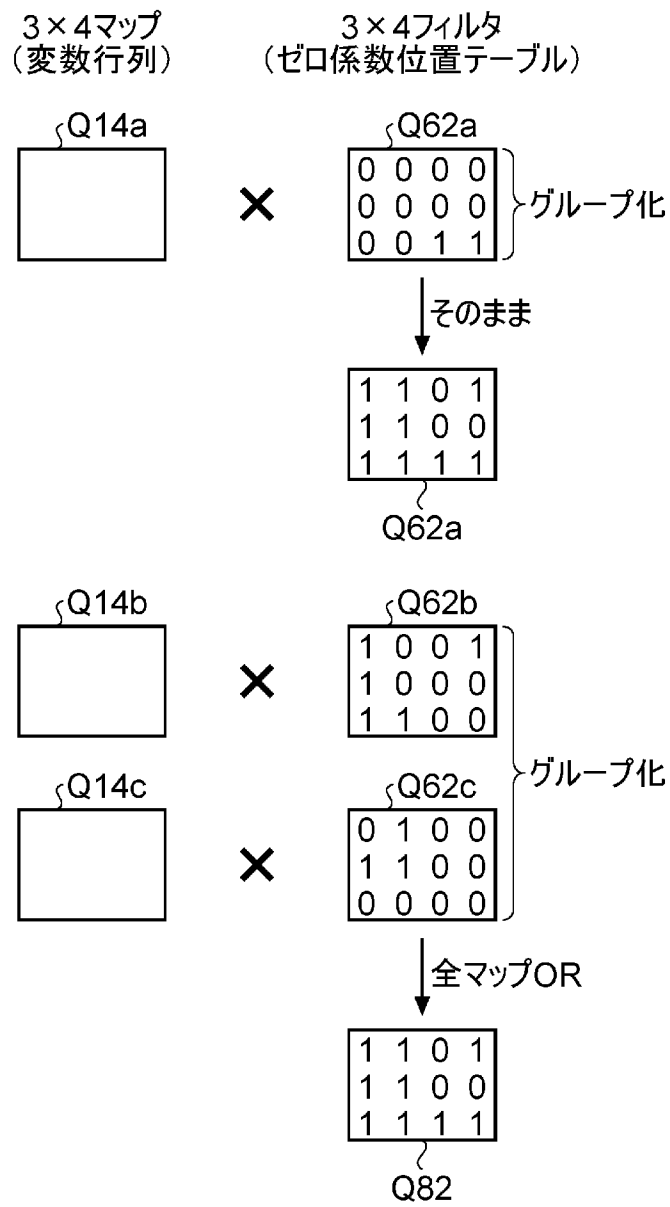
[図25]



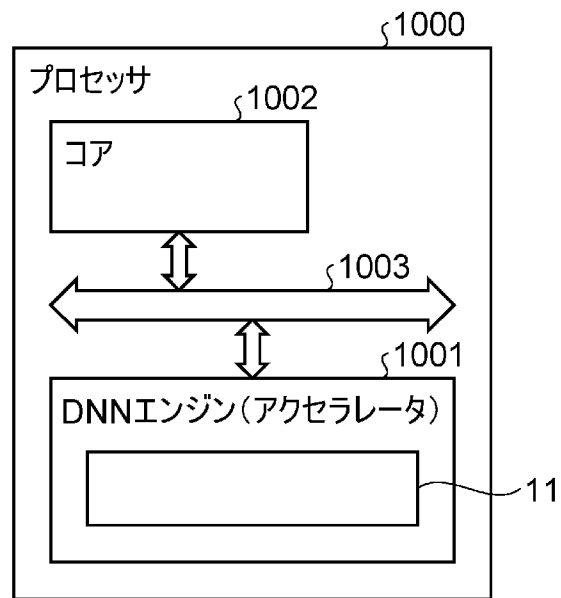
[図26]



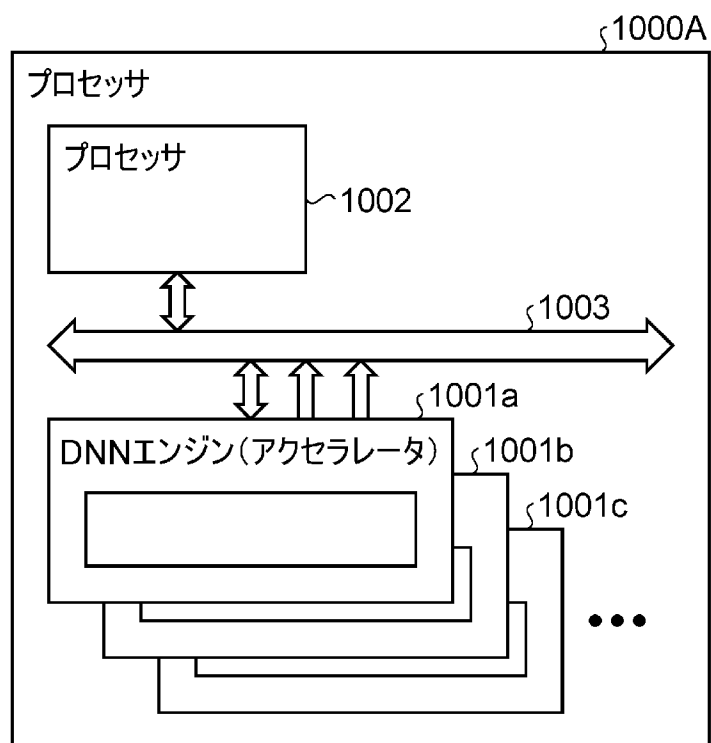
[図27]



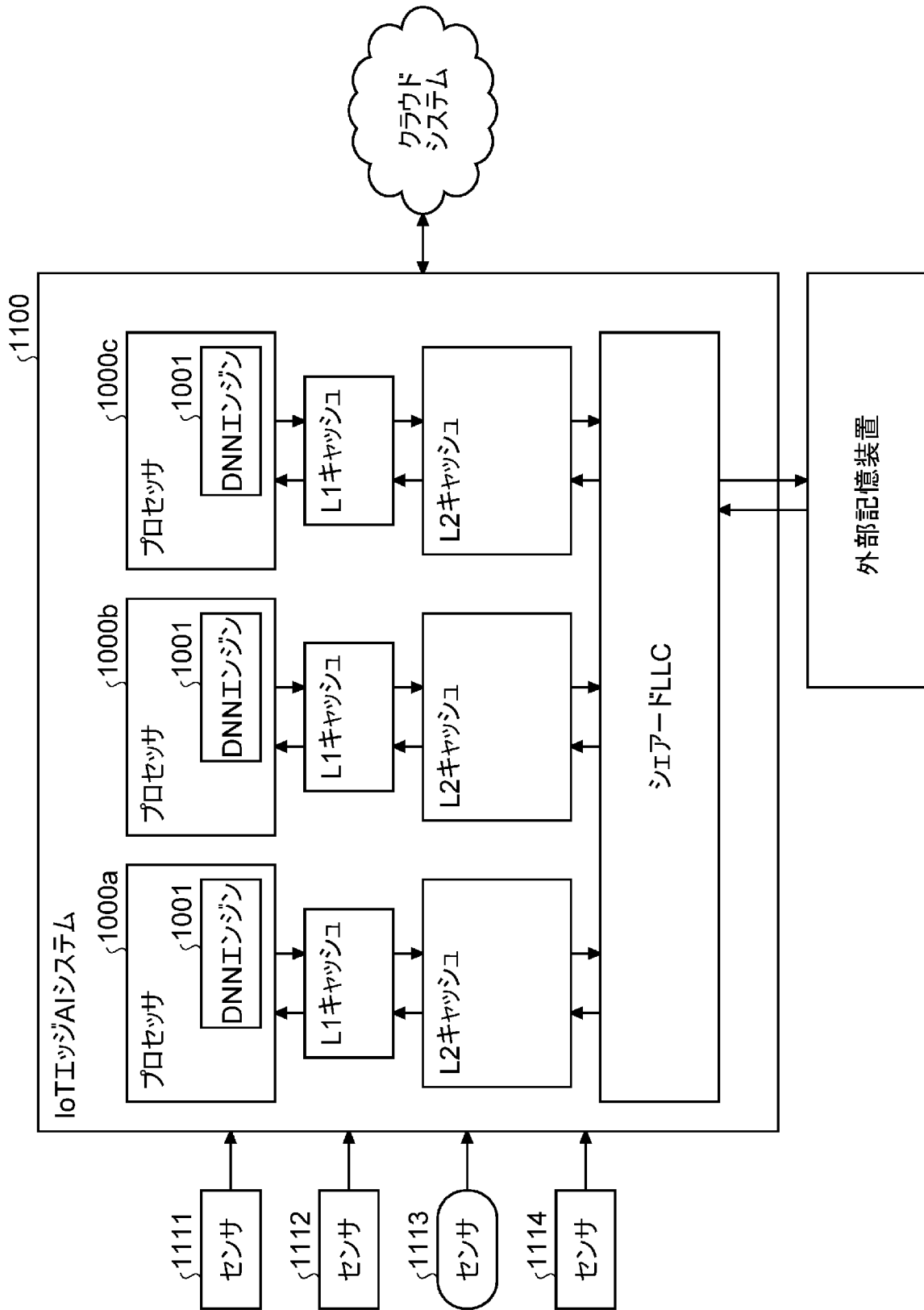
[図28]



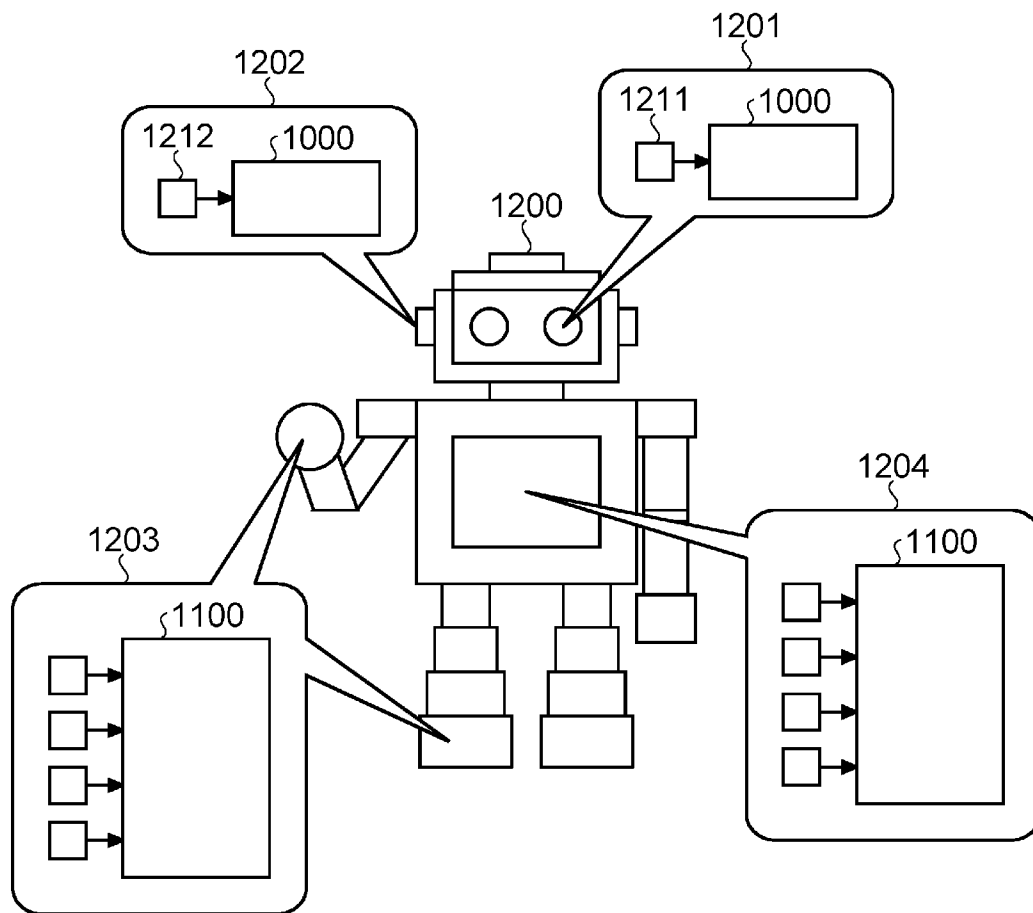
[図29]



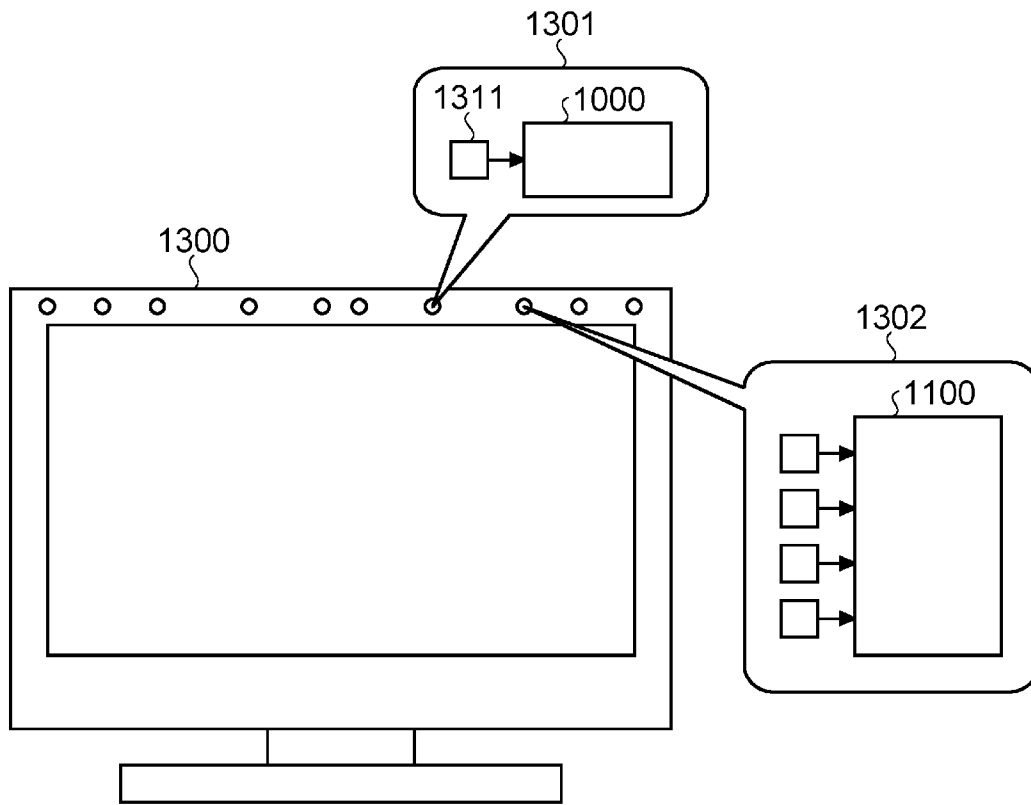
[図30]



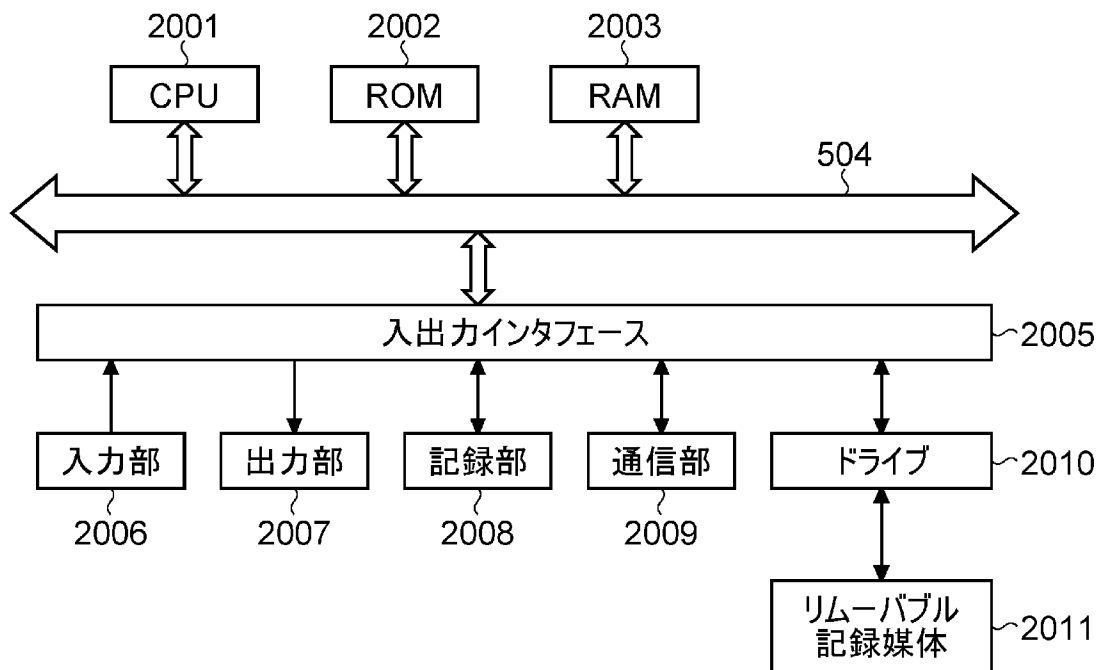
[図31]



[図32]



[図33]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2021/025989

A. CLASSIFICATION OF SUBJECT MATTER

G06N 3/04 (2006.01) i; G06N 3/10 (2006.01) i; G06F 17/10 (2006.01) i
 FI: G06F17/10 A; G06N3/10; G06N3/04

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N3/04; G06N3/10; G06F17/10

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2021
Registered utility model specifications of Japan	1996-2021
Published registered utility model applications of Japan	1994-2021

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y A	WO 2019/146398 A1 (SONY CORP.) 01 August 2019 (2019-08-01) paragraphs [0160]-[0234]	1, 4, 17-18 6, 8-9, 14-16 2-3, 5, 7, 10-13
Y	JP 2020-500365 A (GOOGLE LLC) 09 January 2020 (2020-01-09) paragraphs [0021], [0037]-[0041], [0044], [0049]	6, 8-9, 14-16
Y	JP 2020-517014 A (MICROSOFT TECHNOLOGY LICENSING, LLC) 11 June 2020 (2020-06-11) paragraphs [0025], [0053]-[0054], [0108], [0112], fig. 8	14-16



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
13 September 2021 (13.09.2021)

Date of mailing of the international search report
21 September 2021 (21.09.2021)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/JP2021/025989

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
WO 2019/146398 A1	01 Aug. 2019	(Family: none)	
JP 2020-500365 A	09 Jan. 2020	US 2018/0121377 A1 paragraphs [0025], [0040]-[0044], [0047], [0052] WO 2018/080624 A1 CN 108009626 A KR 10-2019-0053262 A	
JP 2020-517014 A	11 Jun. 2020	US 2018/0300606 A1 paragraphs [0034], [0062]-[0063], [0117], [0121], fig. 8 WO 2018/194998 A1 CN 110520909 A KR 10-2019-0141694 A	

A. 発明の属する分野の分類（国際特許分類（IPC）） G06N 3/04(2006.01)i; G06N 3/10(2006.01)i; G06F 17/10(2006.01)i FI: G06F17/10 A; G06N3/10; G06N3/04		
B. 調査を行った分野		
調査を行った最小限資料（国際特許分類（IPC）） G06N3/04; G06N3/10; G06F17/10		
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1922 - 1996年 日本国公開実用新案公報 1971 - 2021年 日本国実用新案登録公報 1996 - 2021年 日本国登録実用新案公報 1994 - 2021年		
国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X	WO 2019/146398 A1（ソニー株式会社）01.08.2019（2019 - 08 - 01） 段落[0160]-[0234]	1, 4, 17-18
Y		6, 8-9, 14-16
A		2-3, 5, 7, 10-13
Y	JP 2020-500365 A（グーグル エルエルシー）09.01.2020（2020 - 01 - 09） 段落[0021], [0037]-[0041], [0044]及び[0049]	6, 8-9, 14-16
Y	JP 2020-517014 A（マイクロソフト テクノロジー ライセンシング, エルエルシー） 11.06.2020（2020 - 06 - 11） 段落[0025], [0053]-[0054], [0108]及び[0112]並びに[図8]	14-16
☐ C欄の続きにも文献が列挙されている。 ☒ パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー “A” 特に関連のある文献ではなく、一般的技術水準を示すもの “E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの “L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） “O” 口頭による開示、使用、展示等に言及する文献 “P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献 “T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの “X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの “Y” 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの “&” 同一パテントファミリー文献		
国際調査を完了した日 13.09.2021		国際調査報告の発送日 21.09.2021
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号		権限のある職員（特許庁審査官） 坂庭 剛史 5B 9288 電話番号 03-3581-1101 内線 3545

国際調査報告
 パテントファミリーに関する情報

国際出願番号

PCT/JP2021/025989

引用文献			公表日	パテントファミリー文献	公表日
WO	2019/146398	A1	01.08.2019	(ファミリーなし)	
JP	2020-500365	A	09.01.2020	US 2018/0121377	A1
				段落[0025], [0040]-	
				[0044], [0047]及び[0052]	
				WO 2018/080624	A1
				CN 108009626	A
				KR 10-2019-0053262	A
JP	2020-517014	A	11.06.2020	US 2018/0300606	A1
				段落[0034], [0062]-	
				[0063], [0117]及び[0121]	
				並びに図8	
				WO 2018/194998	A1
				CN 110520909	A
				KR 10-2019-0141694	A