

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6540725号
(P6540725)

(45) 発行日 令和1年7月10日 (2019.7.10)

(24) 登録日 令和1年6月21日 (2019.6.21)

(51) Int.Cl.		F I			
G06F	7/499	(2006.01)	G06F	7/499	I O I
G06F	9/315	(2006.01)	G06F	9/315	S
G06N	3/063	(2006.01)	G06N	3/063	

請求項の数 9 (全 39 頁)

(21) 出願番号	特願2017-14780 (P2017-14780)	(73) 特許権者	000005223
(22) 出願日	平成29年1月30日 (2017.1.30)		富士通株式会社
(65) 公開番号	特開2018-124681 (P2018-124681A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成30年8月9日 (2018.8.9)	(74) 代理人	100113608
審査請求日	平成30年11月12日 (2018.11.12)		弁理士 平川 明
早期審査対象出願		(74) 代理人	100105407
			弁理士 高田 大輔
		(72) 発明者	伊藤 真紀子
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	伴野 充
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 演算処理装置、方法、およびプログラム

(57) 【特許請求の範囲】

【請求項1】

演算回路と、

前記演算回路による演算結果である複数の第1固定小数点数データ各々についての非符号となる最上位ビットの位置または非符号となる最下位ビットの位置の分布についての統計情報を取得する取得回路と、

前記演算回路による演算結果である第2固定小数点数データをレジスタに格納する際にビット幅を制限するためのビットレンジを前記統計情報に基づいて特定する特定回路と、を備える演算処理装置。

【請求項2】

前記特定回路は、前記統計情報に基づいて、前記複数の第1固定小数点数データの数に対する前記複数の第1固定小数点数データのうち前記ビットレンジから外れる前記非符号となる最上位ビットまたは前記非符号となる最下位ビットを有するデータの数の割合を求め、前記割合が閾値未満になるように前記ビットレンジを特定する、請求項1に記載の演算処理装置。

【請求項3】

前記統計情報は、前記非符号となる最上位ビットの前記位置の前記分布を示す情報であり、

前記特定回路は、前記統計情報に基づいて、前記非符号となる最上位ビットの前記位置の前記分布の最大値と前記非符号となる最上位ビットの前記位置の前記分布の最小値との中

心に前記ビットレンジの中心が一致するように前記ビットレンジを特定する、請求項 1 に記載の演算処理装置。

【請求項 4】

前記取得回路は、前記非符号となる最上位ビットの前記位置を真値で示すビット列を前記複数の第 1 固定小数点数データについて論理和演算によって累積し、累積したビット列中の最上位の真値の位置を取得することを特徴とする請求項 1 から 3 のいずれか 1 項に記載の演算処理装置。

【請求項 5】

前記取得回路は、前記非符号となる最下位ビットの前記位置を真値で示すビット列を前記複数の第 1 固定小数点数データについて論理和演算によって累積し、累積したビット列中の最下位の真値の位置を取得することを特徴とする請求項 1 から 3 のいずれか 1 項に記載の演算処理装置。

【請求項 6】

前記演算回路に対する演算命令から、前記第 2 固定小数点数データを前記レジスタに格納する際にビット幅を制限するために前記統計情報に基づいて特定された前記ビットレンジを指定するための情報を取得するデコーダをさらに備える請求項 1 から 5 のいずれか 1 項に記載の演算処理装置。

【請求項 7】

前記演算処理装置は、ニューラルネットワークの複数の層の演算を行い、

前記取得回路は、前記複数の層のうち指定された特定の層に対し、選択的に前記統計情報を取得することを特徴とする請求項 1 から 6 のいずれか 1 項に記載の演算処理装置。

【請求項 8】

深層学習を実行するコンピュータが、

所定の演算の結果である複数の第 1 固定小数点数データ各々についての非符号となる最上位ビットの位置または非符号となる最下位ビットの位置の分布についての統計情報を取得することと、

前記所定の演算の結果である第 2 固定小数点数データをレジスタに格納する際にビット幅を制限するためのビットレンジを前記統計情報に基づいて特定することと、を実行する情報処理方法。

【請求項 9】

深層学習をコンピュータに実行させるためのプログラムであって、

所定の演算の結果である複数の第 1 固定小数点数データ各々についての非符号となる最上位ビットの位置または非符号となる最下位ビットの位置の分布についての統計情報を取得することと、

前記演算の結果である第 2 固定小数点数データをレジスタに格納する際にビット幅を制限するためのビットレンジを前記統計情報に基づいて特定することと、を実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、演算処理装置、この演算処理装置を含む情報処理装置、方法、およびプログラムに関する。

【背景技術】

【0002】

今日、深層学習へのニーズが高まっている。深層学習においては、乗算、積和演算、ベクトル乗算を含む様々な演算が実行される。ところで、深層学習では、個々の演算精度への要求は、他のコンピュータ処理ほど厳密ではない。例えば、従来の信号処理等では、プログラマは極力桁あふれを発生させないようにコンピュータプログラムを開発する。一方、深層学習では、大きな値がある程度飽和することは許容される。深層学習では、複数の入力データを畳み込み演算するときの係数（重み）の調整が主な処理となり、入力データ

10

20

30

40

50

のうち、極端なデータは重視されないことが多いからである。また、大量のデータを繰り返し用いて係数を調整するため、一度飽和された値も、学習の進行に合わせて桁調整を行なうことで、飽和されずに係数調整に反映できるようになるためである。

【0003】

そこで、このような深層学習の特性を考慮し、深層学習用の演算処理装置のチップ面積の削減、電力性能の向上等を図るため、浮動小数点数を用いずに、固定小数点数による整数演算を用いることが考えられる。浮動小数点数演算よりも整数演算の方が回路構成を簡素にできるからである。

【0004】

しかしながら、固定小数点数は、ダイナミックレンジが狭いので、浮動小数点数より演算精度が劣化する場合がある。したがって、深層学習においても、小さな値を表現する精度、すなわち、有効桁数について、配慮が求められる。このため、固定小数点数を拡張した技術が提案されている。

【0005】

例えば、Mixed Fixed Pointによる処理では、プログラム全体として統一した小数点位置を使うのではなく、変数ごとに適した小数点位置（Qフォーマット）が用いられる。図1は、Q3.12フォーマットを例示する図である。Q3.12フォーマットは、符号ビット1桁、整数部3桁、小数点以下12桁の16ビットデータを定義する。Mixed Fixed Pointでは、変数ごとに、小数点位置、つまり、整数部の桁数と小数点以下の桁数が異なるものとして処理される。

【0006】

他の例として、Dynamic Fixed Point（動的固定小数点数）による処理では、実行中に変数の値域が取得され、一定のタイミングで小数点位置が見直される。したがって、Mixed Fixed Point演算、およびDynamic Fixed Point演算は、浮動小数点演算よりも処理が簡単な固定小数点演算に、浮動小数点演算の側面を加えたものと言える。

【0007】

さらに、Mixed Fixed Point演算、Dynamic Fixed Point演算による処理を実行するためのプログラム向けの機能を有するDigital Signal Processor（DSP）も提案されている。例えば、ブロック・シフト指定付きの演算命令を実行するDSPがある。ブロック・シフト指定付きの演算命令は、変数のビット幅より大きいビット幅で演算を実行し、演算結果から値をシフトして切り出して変数用のレジスタに格納する。この命令では、演算結果から値を切り出すときのシフト量S（例えば、-128～127）が即値/汎用レジスタで指定可能である。例えば、DSPがResult = Saturate (((in1（演算子）in2）>> S), 16)という命令を実行すると、演算結果をSビットシフトし、下位16bitを残し、上位ビットを飽和処理する。S ≥ 0の場合には、DSPは、演算結果を算術右シフトし、すなわち、符号ビットを埋め込んで右シフトし、下位ビットを削除する。一方、S < 0の場合には、DSPは、演算結果を算術左シフト、すなわち、符号ビットを維持して左シフトし、補数での下位ビットを削除する。

【0008】

ブロック・カウント・リーディング・サイン（BCLS）出力を実行するDSPも提案されている。BCLS出力は、DSPが演算結果のcount leading signを取って、結果をレジスタに書き込む機能である。ここで、count leading signとは、正数最上位のビット1の位置（負数で最上位のビット0の位置）をいう。例えば、DSPが、max(block_count_leading_sign(in1（演算子）in2）-1)を実行すると、変数in1と変数in2の演算子による演算結果から、正数最上位のビット1の位置（負数で最上位のビット0の位置）がレジスタに記録される。

【0009】

図2は、深層学習において、単精度浮動小数点数（Single Precision Floating Point）、半精度浮動小数点数（Half Precision Floating Point）、固定小数点数（Fixed point）、Dynamic Fixed Point数を適用した場合の認識エラー率を例示する（非特許文献1

10

20

30

40

50

参照)。なお、図2で、第1行目のデータ(Goodfellow et al.(2013))は、他の論文の評価結果である。また、図2の表で、Propは、認識(フォワード方向)および学習(バックワード方向)におけるデータ伝搬時のビット幅を示す。また、Up.は、重み係数アップデート時のビット幅を示す。さらに、P I M N I S T、M N I S T、C I F A R - 1 0、およびS V H Nは、深層学習に用いられたデータセットを示す。

【0010】

なお、非特許文献1によるDynamic Fixed Point数による処理では、演算または変数の代入ごとに、オーバーフローの有無を記録し、オーバーフロー率(全演算回数に対するオーバーフローした演算の回数、全代入回数に対するオーバーフローした代入回数等)が算出される。そして、この処理では、所定の期間ごとに、各変数の小数点位置を以下の手順1、2で変更する。

10

(手順1) オーバーフロー率が規定値(rmax)より大きい場合、小数点位置を1つ下げる。

(手順2) オーバーフロー率の2倍の値が規定値(rmax)以下の場合、小数点位置を1つ上げる。

【0011】

図2に例示されるように、従来技術による処理では、単精度浮動小数点数、半精度浮動小数点数、固定小数点数、およびDynamic Fixed Point数による結果においても、認識エラー率は大きく増加していない。

20

【先行技術文献】

【特許文献】

【0012】

【特許文献1】特開2016-33754号公報

【特許文献2】特開2012-203566号公報

【特許文献3】特開2010-112878号公報

【非特許文献】

【0013】

【非特許文献1】Courbariaux et al., “TRAINING DEEP NEURAL NETWORKS WITH LOW PRECISION MULTIPLICATIONS” Accepted as a workshop contribution at ICLR 2015, International Conference on Learning Representations(ICLR), 2015年9月23日,

30

【発明の概要】

【発明が解決しようとする課題】

【0014】

上述のように、従来技術のDynamic Fixed Point数による処理結果においても、認識エラー率が大きく増加しない。しかしながら、上記従来の技術においては、小数点位置をオーバーフロー率を基に1ビット調整するため学習結果の精度が劣化する。例えば、k個のデータの学習ごとに小数点位置を調整する場合、小数点位置が適正位置からnビットずれている場合、小数点位置が適切な位置に調整されるまでに、(n * k)回の学習が進む。その結果、小数点位置が適正な位置になるまでに、飽和するデータやアンダーフローするデータが増え、学習が収束しない、あるいは、学習結果の精度劣化につながるおそれがある。

40

1つの側面では本発明の目的は、深層学習を固定小数点数で精度よく実行し、回路規模と消費電力を削減できるようにすることである。

【課題を解決するための手段】

【0015】

本発明の一側面は、以下の演算処理装置によって例示される。すなわち、本演算処理装置は、固定小数点数データに対する命令実行後の前記固定小数点数データ中のビットの分布についての統計情報を取得する取得回路と、前記固定小数点数データの小数点位置を更新する更新回路と、を備える。

【発明の効果】

50

【0016】

本実施形態によれば、深層学習を固定小数点数で精度よく実行することで、回路規模と消費電力を削減することができる。

【図面の簡単な説明】

【0017】

【図1】Q3.12フォーマットを例示する図である。

【図2】深層学習において、単精度浮動小数点数、半精度浮動小数点数、固定小数点数、Dynamic Fixed Point数を適用した場合の認識エラー率を例示する図である。

【図3】ニューラルネットワークにおける深層学習を例示する図である。

【図4】フォワード方向の認識処理および識別処理とともにバックワード方向の学習処理を例示する図である。 10

【図5】比較例に係る深層学習の手順を例示する図である。

【図6】各層の処理の詳細を例示する図である。

【図7】比較例の情報処理装置による学習処理のフローチャートを例示する図である。

【図8A】非符号となる最上位ビット位置の分布データを例示する図である。

【図8B】非符号となる最下位ビット位置の分布データを例示する図である。

【図9】プロセッサの構成を例示する図である。

【図10】プロセッサの回路ブロックを例示する図である。

【図11A】ベクトルユニットの詳細を例示する図である。

【図11B】スカラユニットの構成を例示する図である。 20

【図12】データ変換部の構成を例示する図である。

【図13A】非符号となる最上位ビット位置を取得する統計情報取得部の処理を例示する図である。

【図13B】非符号最上位ビット検出器の真理値表を例示する図である。

【図14】非符号となる最上位ビット位置を取得する統計情報取得部のハードウェア回路の構成を例示する図である。

【図15A】非符号となる最下位ビット位置を取得する統計情報取得部の処理を例示する図である。

【図15B】非符号最下位ビット検出器の真理値表を例示する図である。

【図16】非符号となる最下位ビット位置を取得する統計情報取得部のハードウェア回路の構成を例示する図である。 30

【図17】統計情報集約部の処理を例示する図である。

【図18】統計情報集約部のハードウェア回路の構成を例示する図である。

【図19】統計情報集約部の処理を例示する図である。

【図20】統計情報集約部のハードウェア回路の構成を例示する図である。

【図21】統計情報格納部のハードウェア回路の構成を例示する図である。

【図22】統計情報集約部の変形例である。

【図23】統計情報格納部のハードウェア回路の構成を例示する図である。

【図24】統計情報格納部のハードウェア回路の構成を例示する図である。

【図25】変形例の統計情報格納部の構成を例示する図である。 40

【図26】変形例の統計情報格納部の構成を例示する図である。

【図27】命令フォーマットを例示する図である。

【図28】命令フォーマットを例示する図である。

【図29】情報処理装置が深層学習を実行する場合の各層の処理の詳細を例示する図である。

【図30】情報処理装置による学習処理のフローチャートを例示する図である。

【図31】データが固定小数点数の範囲に収まらない場合の小数点位置更新例である。

【図32】データが固定小数点数の範囲に収まる場合の小数点位置更新例である。

【図33】非符号となる最下位ビットの分布である。

【発明を実施するための形態】

【0018】

以下、図面を参照して一実施形態に係る情報処理装置について説明する。本情報処理装置は、深層学習を実行する。

〔深層学習の実行例〕

【0019】

まず、図3を参照して、ニューラルネットワークにおける深層学習を例示する。ニューラルネットワークは、ハードウェア回路であってもよいし、コンピュータプログラム上で仮想的に構築される階層間を接続するソフトウェアによる仮想的なネットワークであってもよい。ニューラルネットワークは、例えば、画像を認識し、識別するためのフォワード方向の処理と、フォワード方向の処理で使用するパラメータを決定するバックワード方向の処理を実行する。なお、バックワード方向の処理は、バックプロパゲーション (Back Propagation、BP) と呼ばれる。

10

【0020】

図3のニューラルネットワークは、入力画像に対して、畳み込み層 (Convolution Layer) の処理と、プーリング層 (Pooling Layer) の処理とを実行し、画像の特徴を抽出し、画像を識別する。すなわち、図3では、フォワード方向の処理が例示されている。図3では、Input Layerとされている入力画像に対して、畳み込み層の処理とプーリング層の処理が実行され、4フィーチャマップ、6フィーチャマップ等が順次生成される。ここでフィーチャマップは、例えば、画像の異なるチャンネルで授受されるデータ、あるいは、画像のうちのR、G、Bデータ等に相当するデータである。そして、図3のニューラルネットワークは、最終層であるFully Connected multilayer perceptron (MLP) において、識別結果を出力する。なお、プーリング層は、サブサンプリング層 (Sub-sampling Layer) とも呼ばれる。最終層は、全結合層 (Fully Connected Layer) とも呼ばれる。

20

【0021】

フォワード方向の処理のうち、最終層に至るまでの畳み込み層の処理と、プーリング層の処理を繰り返し実行する演算回路部分を特徴抽出部と呼ぶ。すなわち、特徴抽出部は、入力画像に対して、畳み込み層の処理と、プーリング層の処理を繰り返し実行することで、間引かれた画像を抽出する。畳み込み層の処理は、畳み込み演算とも呼ばれる。プーリング層の処理は、プーリング演算、あるいはサブサンプリング演算とも呼ばれる。

30

【0022】

より具体的には、特徴抽出部は、フォワード方向のある層 (第 $l-1$ 層) において、 N 個 $\times$ N 個の画素を有する画像の情報に対して、例えば、 $m \times m$ 個の重み w_{ab} ($a, b = 0, \dots, m-1$) のフィルタによる畳み込み演算を実行する。特徴抽出部は、この畳み込み演算を実行することで、現在処理中の層 (第 $l-1$ 層) の画像から、次の層 (第 l 層) の画像の情報を作る。さらに、畳み込み演算による処理結果は、非線形活性化関数 $y = \text{sigma}(x)$ によって変換され、発火と呼ばれるモデル化がなされる。非線形活性化関数は、例えば、 $\text{sigma}(x) = \max(0, x)$ である。すなわち、非線形活性化関数 $y = \text{sigma}(x)$ は、 x が負の値の場合、 $\text{sigma}(x) = 0$ であり、 x が正の値の場合、 $\text{sigma}(x) = x$ の値をとる関数である。なお、重み w は畳み込み演算に用いられることから、係数ということもできる。

40

【0023】

<畳み込み層の計算>

今、第 $l-1$ 層における画素 (i, j) からの出力を $y^{l-1}_{i,j}$ とし、第 l 層の画素 (i, j) における畳み込み演算の処理結果を $x^l_{i,j}$ とし、第 l 層における画素 (i, j) の非線形活性化関数 $y = \text{sigma}(x)$ の演算結果を $y^l_{i,j}$ とする。 $x^l_{i,j}$ と $y^l_{i,j}$ は以下の(式1)、(式2)で表すことができる。ただし、(式2)において、関数 sigma は、ギリシャ文字で表している。

【0024】

50

【数 1】

$$x_{ij}^{\ell} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{\ell-1} \quad (\text{式 1}) \quad \text{畳み込み層}$$

【0 0 2 5】

【数 2】

$$y_{ij}^{\ell} = \sigma(x_{ij}^{\ell}) \quad (\text{式 2}) \quad \text{畳み込み層}$$

10

【0 0 2 6】

式 1 の対象となる第 $l-1$ 層の画像の情報のうち、 $m \times m$ 個の画素の領域を以下、単位矩形領域と呼ぶ。なお、単位矩形領域は 1 つのフィーチャマップに限定される訳ではなく、図 1 に例示した 4 フィーチャマップ、6 フィーチャマップが畳み込み演算の対象となる場合に、複数のフィーチャマップを用いて、畳み込み演算が実行されてもよい。また、以下、画素をセルともよぶ。

【0 0 2 7】

プーリング演算は、 N 個 $\times$ N 個の画素を有する画像を N/k 個 $\times$ N/k 個の画素を有する画像に間引き処理である。間引き方としては、例えば、 N 個 $\times$ N 個の画素を k 個 $\times$ k 個ずつの部分画像に区切り、各部分画像に含まれる k 個 $\times$ k 個の画素中の最大の画素値を有する 1 つの画素を選択する処理（以下、`max pooling` と呼ぶ）が例示できる。また、他の例として、 k 個 $\times$ k 個の画素の平均値によって 1 つの画素を生成する処理が例示できる。

20

【0 0 2 8】

識別部は、間引かれた画像中の全画素に対して、それぞれの画素に重みを適用し、畳み込み演算を実行することで、出力値を得る。今、特徴抽出部の最後の層を第 $l-1$ 層として、第 $l-1$ 層の各画素値を y^{l-1}_{ji} とし、全結合による畳み込み演算結果を x^l_i とし、重みを w^{l-1}_{ji} とすると、 x^l_i は以下の（式 3）で表すことができる。

【0 0 2 9】

【数 3】

$$x_i^{\ell} = \sum_j w_{ji}^{\ell-1} y_j^{\ell-1} \quad (\text{式 3}) \quad \text{全結合層}$$

30

【0 0 3 0】

また、識別部における非線形活性化関数 $y = \text{sigma}(x)$ の演算結果を y^l_i とすると、識別部の演算結果 y^l_i は以下の（式 4）で表すことができる。ただし、（式 4）において、関数 `sigma` は、ギリシャ文字で表している。

【0 0 3 1】

【数 4】

$$y_i^{\ell} = \sigma(x_i^{\ell}) + I_i^{\ell} \quad (\text{式 4}) \quad \text{全結合層}$$

40

ここで、 I^l_i はバイアス成分である。

【0 0 3 2】

図 4 に、フォーワード方向の認識処理および識別処理とともにバックワード方向の学習処理を例示する。図 4 のニューラルネットワークにおいても、フォーワード方向の認識処理は、入力画像に畳み込み演算を実行する畳み込み層、間引き処理を実行するプーリング層によって実行される。また、識別結果を出力する識別処理は、全結合層（図 4 では、`Fully connected` と記載）によって実行される。

50

【0033】

全結合層による識別処理の結果は、正解データを比較され、比較結果である差分値がエラーとして出力される。エラーは、学習処理に用いられる。学習処理は、エラーから、フォワード方向の畳み込み層および全結合層における各層でのエラーおよび各層での次の重みを計算する処理である。図2では、現在の重みとして、畳み込み層（1層）における1つの重み w_i と、全結合層（1層）における1つの重み w_j が例示されている。また、次の重みとして、畳み込み層（1層）における1つの重み w_{i+1} と、全結合層（1層）における1つの重み w_{j+1} が例示されている。

【0034】

今、例えば、エラーの評価関数として、識別結果である出力値 y_i と正解データ T_i との差分値の平方和を評価関数として例示すると、（式5）のように定義できる。学習処理は、（式5）で例示されるエラーの評価関数を最小にするための重み w を決定する処理と考えることができる。

【0035】

【数5】

$$E = 1/2 \sum_i (T_i - y_i)^2 \quad (\text{式5})$$

【0036】

次に、（式5）を y_i で偏微分すると（式6）となり、右辺は出力値 y_i と正解データ T_i との差である誤差値となる。エラーの評価関数を（式5）から一般化した場合も、エラーの評価関数を出力値 y_i で偏微分すると、出力値 y_i と正解データ T_i との差である誤差値となると考えることができる。

【0037】

【数6】

$$\frac{\partial E}{\partial y_i^L} = \frac{d}{dy_i^L} E(y^L) \quad (\text{式6})$$

さらに、（式5）を x_i で偏微分し、（式7）の推移律を適用し、さらに、 y_i に式4を代入すると、（式8）が得られる。なお、画素値 x と、非線形活性化関数の出力 y との関係を（式2）で表す場合には、（式9）が得られる。

【0038】

【数7】

$$\partial E / \partial x_i = (\partial E / \partial y_i) (\partial y_i / \partial x_i) \quad (\text{式7})$$

【0039】

【数8】

$$\frac{\partial E}{\partial x_j^\ell} = \sigma'(x_j^\ell) \frac{\partial E}{\partial y_j^\ell} \quad (\text{式8})$$

【0040】

【数9】

$$\frac{\partial E}{\partial x_{ij}^\ell} = \frac{\partial E}{\partial y_{ij}^\ell} \frac{\partial y_{ij}^\ell}{\partial x_{ij}^\ell} = \frac{\partial E}{\partial y_{ij}^\ell} \frac{\partial}{\partial x_{ij}^\ell} (\sigma(x_{ij}^\ell)) = \frac{\partial E}{\partial y_{ij}^\ell} \sigma'(x_{ij}^\ell) \quad (\text{式9})$$

一方、（式5）を y_i^1 で偏微分し、全結合層の画素値 x_i^1 が1つ入力側の層（第1-1層

10

20

30

40

50

）からの出力値 y^{l-1}_j によって（式 3）のように表されることを考慮すると、（式 10）が得られる。なお、第 1 層の画素値 $x^{l-1}_{i,j}$ が 1 つ入力側の層（第 1 - 1 層）からの出力値 y^{l-1}_j によって（式 1）で表される場合には、（式 10）は、（式 11）のようになる。

【0041】

【数 10】

$$\frac{\partial E}{\partial y_i^\ell} = \sum w_{ij}^\ell \frac{\partial E}{\partial x_j^{\ell+1}} \quad (\text{式 10}) \quad \text{全結合層}$$

【0042】

10

【数 11】

$$\frac{\partial E}{\partial y_{ij}^{\ell-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\partial E}{\partial x_{(i-a)(j-b)}^\ell} \frac{\partial x_{(i-a)(j-b)}^\ell}{\partial y_{ij}^{\ell-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\partial E}{\partial x_{(i-a)(j-b)}^\ell} \omega_{ab} \quad (\text{式 11})$$

畳み込み層

$$\frac{\partial x_{(i-a)(j-b)}^\ell}{\partial y_{ij}^{\ell-1}} = \omega_{ab}$$

【0043】

20

（式 10）の左辺は、第 1 層の誤差を示す。また、（式 10）の右辺は、第 1 + 1 層の誤差に、第 1 層の画素 i と第 1 + 1 層の画素 j との間の重みの変数 $w_{i,j}$ を乗算した結果の集計である。この集計は、第 1 層の画素 i に関係する第 1 + 1 層の画素 j について集計したものである。

【0044】

また、（式 5）を重み $w_{i,j}^l$ で偏微分し、推移律を適用し、さらに、（式 3）を代入すると、（式 12）が得られる。ただし、（式 3）は、第 1 - 1 層からの出力 y と、第 1 層の画素値 x の関係となっているが、（式 12）は、第 1 層からの出力 y と、第 1 + 1 層の画素値 x の関係となっている。（式 12）は、エラーの評価関数 E の勾配を示す。最勾降下法によるニューラルネットワークの学習処理においては、エラーの評価関数 E の勾配と、学習係数イータの積が重み W の変化量（現在の重み W_t と次の重み W_{t+1} の差分値）となる。したがって、以下の記述において、エラーの評価関数 E の勾配を重み W の更新量 ΔW と定義する。なお、画素値 x と y の関係が（式 1）で表される場合には、（式 12）は、（式 13）のようになる。以上から、エラーの評価関数 E に最勾降下法を適用すると、（式 14）が得られる。なお、（式 14）では、演算が M 回繰り返される式となっているが、 $M = 1$ でもよい。

30

【0045】

【数 12】

$$\frac{\partial E}{\partial w_{ij}^\ell} = y_i^\ell \frac{\partial E}{\partial x_j^{\ell+1}} \quad (\text{式 12}) \quad \text{全結合層}$$

40

【0046】

【数 13】

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{N-m} \sum_{j=0}^{N-m} \frac{\partial E}{\partial x_{ij}^\ell} \frac{\partial x_{ij}^\ell}{\partial \omega_{ab}} = \sum_{i=0}^{N-m} \sum_{j=0}^{N-m} \frac{\partial E}{\partial x_{ij}^\ell} y_{(i+a)(j+b)}^{\ell-1} \quad (\text{式 13})$$

畳み込み層

【0047】

【数 1 4】

$$W_{t+1} = W_t + \frac{\eta}{|M|} \sum_{m=1}^M \frac{\partial E}{\partial w_t} \quad (\text{式 1 4})$$

(A) 各層の重み

【0 0 4 8】

(式 1 4) は、現在の重み W_t と次回認識処理での重み W_{t+1} との関係を示している。したがって、(式 1 4) の右辺の第 2 項が次回の重みへの増分値となる。また、(式 1 4) でギリシャ文字イータは、エラーの評価関数 E の勾配から重み W_t の増分値を得るための係数であり、例えば、経験的実験的に設定できる。すなわち、学習処理においては、学習回路の各層において、(式 1 2) (あるいは式 1 3) による勾配を求め、エラーの評価関数 E が小さくなる方向に、(式 1 4) の第 2 項の増分値を算出し、次回の重み W_{t+1} を求める処理を実行する。

10

(B) 各層のエラーの評価関数の勾配 (更新量 ΔW の計算)

【0 0 4 9】

ここで、(式 1 4) の第 2 項は、(式 1 2) により、各層 (第 1 層) の画素値 (y^1_i) と、各層 (第 1 層) に対して 1 つ出力側の層 (第 1 + 1 層) のエラー (エラーの評価関数の画素値 x^{1+1}_j による偏微分値) との積から得られる。ただし、現在の層 ((式 1 3) では第 1 - 1 層) と、1 層全結合層側の層 ((式 1 3) では第 1 層) との関係が (式 1) のように積和演算で表される場合、更新量 ΔW は、(式 1 3) で表される。すなわち、更新量 ΔW は、現在の層 (第 1 - 1 層) の画素の出力値 $y^{1-1}_{i,j}$ と関係する第 1 層の画素値 $x^1_{i,j}$ からのエラーへの寄与を集計したものとなっている。

20

(C) 各層のエラー

【0 0 5 0】

一方、(式 1 0) の左辺は、各層 (第 1 層) のエラー (エラーの評価関数の画素値 y^1_i による偏微分値) を示す。また、(式 1 0) の右辺は、現在の層 (第 1 層) に対して 1 つ出力側の層 (第 1 + 1 層) のエラー (エラーの評価関数の画素値 x^{1+1}_j による偏微分値) と、第 1 層の画素 i と第 1 + 1 層の画素 j との間の重みの変数 $w_{i,j}$ との積の集計である。この集計は、第 1 層の画素 i に関する第 1 + 1 層の画素 j についての集計である。

30

[比較例]

【0 0 5 1】

図 5 から図 7 を参照して、比較例に係る情報処理装置での深層学習について説明する。図 5 は、比較例に係る深層学習の手順を例示する図である。比較例の情報処理装置は、コンピュータプログラムにしたがって深層学習を実行する。比較例の情報処理装置は、Dynamic Fixed Point 数の処理を実行可能なプロセッサを有している。

【0 0 5 2】

図 5 のように、比較例の深層学習は、1 層目の畳み込み層 (Conv_1) とプーリング層 (Pool_1)、2 層目の畳み込み層 (Conv_2) とプーリング層 (Pool_2)、全結合層 1 (fc1)、および全結合層 2 (fc2) の処理により実行される。深層学習は、ミニバッチと呼ばれる処理の単位に区切られて実行される。ここで、ミニバッチとは、学習の対象となる入力データの集合 $\{(In_i, T_i), i = 1 \sim N\}$ を複数組 (例えば、 k 個のデータの M 組、 $N = k * M$) に分割した k 個のデータの組み合わせをいう。また、ミニバッチは、このような個々の入力データの組 (k 個のデータ) で実行される学習の処理単位のことをいう。ここで、 In_i は、入力データ (ベクトル)、 T_i は正解データ (ベクトル) である。比較例の情報処理装置は、以下のように、深層学習中の所定数のミニバッチごとに、各層の各変数のオーバーフロー回数を取得してコンピュータプログラム中の変数に蓄積し、深層学習に用いる変数の固定小数点位置を自動調整する。

40

【0 0 5 3】

比較例の情報処理装置は、例えば、浮動小数点数による試行 (ミニバッチ 1 回など) や、

50

ユーザ指定により各変数の初期小数点位置を決定し、学習を開始する（S1）。そして、情報処理装置は、k回のミニバッチを学習中に、各層の各変数のオーバーフローの回数を統計情報として保存する（S2）。このとき、情報処理装置は、学習を実行するプログラム中の各変数に対する演算、代入等がなされる毎に、オーバーフローの有無を判定し、回数を計数し、回数を保持するカウンタ変数に代入する。

【0054】

ミニバッチを学習中に変数にオーバーフローが発生した場合は、変数に対する飽和処理を行ない、学習を続行する（S3）。ここで、飽和処理とは、正の数がオーバーフローした際には正の最大値に、負の数がオーバーフローした際には負の最小値にクリップする処理である。

10

【0055】

そして、情報処理装置は、k組のデータに対するミニバッチ終了後（k回の学習終了後）、カウンタ変数に蓄積したオーバーフロー回数からオーバーフロー率を求め、オーバーフロー率を基に、固定小数点数の小数点位置を調整する（S4）。そして、情報処理装置は、S2の処理に戻り、学習を進める。

【0056】

図6は、図5における各層の処理の詳細を例示する図である。図では、1層目の畳み込み層(Conv_1)、プーリング層(Pool_1)、および全結合層2(fc2)が例示されている。この例では、1層目の畳み込み層(Conv_1)において、入力データ $ini=(x_0, \dots, x_n)$ と、重み (W_{ij}) との間で、畳み込み演算が実行され、 $z_0, \dots, z_j, \dots$ 等が計算される。また、 $z_0, \dots, z_j, \dots$ 等に対して、活性化関数が演算され、 $u_0, \dots, u_j, \dots$ 等が計算される。情報処理装置は、この1層目の畳み込み層(Conv_1)での変数 $z_0, \dots, z_j, \dots$ のオーバーフロー回数をカウンタ変数Conv_1_zに蓄積する。また、情報処理装置は、この1層目の畳み込み層(Conv_1)での変数 $u_0, \dots, u_j, \dots$ 等でのオーバーフロー回数をカウンタ変数Conv_1_uに蓄積する。例えば、カウンタ変数Conv_1_uが配列の場合、カウンタ変数Conv_1_u(1)には、変数 $u_0, \dots, u_j, \dots$ のうち、オーバーフローした変数の個数が累積される。また、カウンタ変数Conv_1_u(2)には、変数 $u_0, \dots, u_j, \dots$ のうち、2倍するとオーバーフローする変数の個数が累積される。

20

【0057】

1層目の畳み込み層(Conv_1)での演算結果は、1層目のプーリング層(Pool_1)から、さらに上位層へ伝播される。このとき、情報処理装置は、2層目以上の層でも同様の演算と、オーバーフロー回数の蓄積を実行する。情報処理装置は、以上のような演算を上位層においても実行して演算結果をさらに上位層へ伝播させる。そして、情報処理装置は、最終的には、全結合層2(fc2)において、得られた演算結果と、正解データ T_i との差分値からエラー $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ を算出する。そして、情報処理装置は、エラー $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ を基に、逆方向の伝播を実行する。その結果、情報処理装置は、下位の階層（例えば、結合層1(fc1)）の出力でのエラー $\delta_0^{(6)}, \dots, \delta_j^{(6)}, \dots, \delta_n^{(6)}$ と重みの変数 W_{ij} の次の重みへの差分値 ΔW_{ij} を算出する。このとき、情報処理装置は、エラー $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ およびエラー $\delta_0^{(6)}, \dots, \delta_j^{(6)}, \dots, \delta_n^{(6)}$ でのオーバーフロー回数をカウンタ変数fc2_δに蓄積する。また、情報処理装置は、差分値 ΔW_{ij} でのオーバーフロー回数をカウンタ変数fc2_ΔWに蓄積する。また、情報処理装置は、重みの変数 W_{ij} でのオーバーフロー回数をカウンタ変数fc2_Wに蓄積する。

30

40

【0058】

そして、情報処理装置は、全結合層1(fc1)から1層目の畳み込み層(Conv_1)まで、エラーを逆方法に伝播させながら、重みの差分を計算する。情報処理装置は、以上のような順伝播と逆伝播をk組の入力データを用いて、k回繰り返す。情報処理装置は、このようなk回のミニバッチ終了後、各変数に対応するカウンタ変数のオーバーフロー回数を基に、各変数の固定小数点位置を更新する。

【0059】

図7に、比較例の情報処理装置による学習処理のフローチャートを例示する。この処理

50

では、情報処理装置は、初期小数点位置を決定する（C 1）。そして、情報処理装置は、回数 k を値0で初期化する。そして、情報処理装置は学習終了の条件が充足されたか否かを判定する（C 3）。学習が終了するのは、全結合層(fc2)でのエラーが基準値以下となるか、または、学習回数が規定の最大値に達した場合である。

【0060】

学習終了の条件が充足されない場合、情報処理装置は、次のミニバッチを実行し、変数 k をカウントアップする（C 4）。そして、情報処理装置は、統計情報としてオーバーフロー回数を各カウンタ変数に蓄積する（C 5）。すなわち、情報処理装置は、各層において、オーバーフローした変数の個数、2倍するとオーバーフローする変数の個数を累積する。

10

【0061】

そして、情報処理装置は、回数 k が更新間隔に達したか否かを判定する（C 6）。回数 k が更新間隔に達していない場合、情報処理装置は、C 3の処理に戻る。一方、回数 k が更新間隔に達した場合、情報処理装置は、変数のオーバーフロー回数にしたがって、小数点位置を更新する。例えば、情報処理装置は、非特許文献1と同様、オーバーフロー回数を演算の実行回数で除算したオーバーフロー率を求めればよい。そして、オーバーフロー率が規定値を超えた場合に、情報処理装置は、該当する変数の小数点位置を1つ下げ、整数部を1ビット拡張すればよい。また、オーバーフロー率の2倍の値が規定値以下の場合、情報処理装置は、小数点位置を1つ上げ、整数部を1ビット削減すればよい。そして、情報処理装置1は、C 3の処理に戻る。

20

【0062】

しかし、以上の処理は、各層の演算ごとにオーバーフロー回数を累積し、オーバーフロー率が既定値を越えた場合、および、オーバーフロー率の2倍の値が既定値未満となった場合に固定小数点位置を1ビットずらす処理である。この処理は、オーバーフロー回数あるいはオーバーフロー率に基づく固定小数点位置の更新処理である。情報処理装置は、オーバーフロー回数あるいはオーバーフロー率からは、適正な小数点位置がどこであるかを判断することが困難である。

【0063】

したがって、比較例の処理は、固定小数点位置の更新時、小数点位置を1つ下げる、あるいは上げる等の処理で、適正な位置を試行し、その結果からさらに判断を重ねていくという手順となる。このため、情報処理装置が適正な小数点位置を決定するまでに、小数点位置が複数回更新されることになる。例えば、上記のように、情報処理装置が k 回のミニバッチごとにオーバーフロー率を判定し、1ビットずつ小数点位置を更新する場合には、小数点位置を N ビットずらすために、 $N \times k$ 回（ $\times$ は乗算）分のミニバッチが実行される。このため、小数点位置が不適切な状態で学習処理が繰り返される。つまり、学習中に、望ましくないレベルで固定小数点数データの飽和、あるいは、アンダーフローが発生し、学習結果の精度劣化につながり、収束が遅くなる恐れがある。

30

【0064】

また、比較例の情報処理装置は、単にオーバーフロー回数を取得し、蓄積する処理を深層学習のプログラム中で実行するため、学習時間が増大する可能性がある。つまり、情報処理装置は、適正な小数点位置を決定するための情報をプログラムに提供する機能がないため、プログラムにオーバーフロー回数を累積するコードが組み込まれ、処理時間が増加する恐れがある。

40

〔実施形態〕

【0065】

以下、図8Aから図32により、一実施形態に係る情報処理装置1、情報処理装置1が実行する情報処理方法、および情報処理装置1で実行されるプログラムについて説明する。本実施形態は、一例であり、情報処理装置1は、本実施形態の構成に限定される訳ではない。

<統計情報>

50

【0066】

本実施形態では、情報処理装置1のプロセッサが統計情報を取得することで、統計情報取得のためのプログラム中のオーバーヘッドが削減される。ここで、プロセッサが取得する統計情報は、例えば、以下のいずれか、または、これらの組み合わせをいう。情報処理装置1が実行するアプリケーションプログラムは、プロセッサから統計情報を取得することで、小数点位置を最適化する。アプリケーションプログラムの処理にしたがってプロセッサは、Dynamic Fixed Point演算のための命令を実行する。

(1) 非符号となる最上位ビット位置の分布

【0067】

図8Aに、非符号となる最上位ビット位置の分布データを例示する。図8Aは演算の途中結果が40bitで、固定小数点数の桁あわせのために14bit右シフトされるデータに対する例である。非符号となる最上位ビット位置とは、正数に対しては、ビットが1となっている最上位のビット位置をいう。また、負数に対しては、ビット0となっている最上位のビット位置をいう。非符号となる最上位ビット位置は、例えば、ビットの並びを最上位ビットであるbit[39]から最下位ビットであるbit[0]としたとき、符号ビットbit[39]と異なるbit[k]のうち、インデックスkが最大のものをいう。非符号となる最上位ビット位置の分布が得られると、絶対値としての値の分布範囲が把握可能となる。

【0068】

図8Aで、縦軸は、非符号となる最上位ビット位置の出現個数であり、横軸は最上位のビットの位置 Count Leading Sign (CLS) である。図8Aでは、ビット0の右側に小数点があると仮定する。本実施形態では、情報処理装置1のプロセッサの演算回路および演算回路内のレジスタは、命令のオペランドで指定されるレジスタのビット数（例えば、16ビット）以上のビット幅（例えば、40ビット）を有する。ただし、情報処理装置1のプロセッサの演算回路および演算回路内のレジスタのビット幅が40ビットに限定される訳ではない。また、演算結果は、例えば、16ビットのレジスタ等、演算回路よりも少ないビット幅のレジスタ（命令のオペランドで指定されるレジスタ）に格納される。その結果、演算結果は（例えば、40ビット）、オペランドで指定されるシフト量でシフトし、ビット0未満に該当するビットは所定の丸め処理がなされ、オペランドで指定されるレジスタのビット幅を越えるデータ（ビット15を越えるデータ）は、飽和処理される。

【0069】

また、図8Aの横軸に付与された数値は、固定小数点で表現可能な数値を示す。例えば、情報処理装置1がこの固定小数点数を-2ビットだけシフト（右方向に2ビットシフト）すると、最上位ビットが14の位置にシフトし、飽和される領域が2ビット拡張され、アンダーフローが発生して0となる領域が2ビット減少する。すなわち、情報処理装置1が小数点位置を2ビット左にシフトすると、飽和される領域が2ビット拡張され、アンダーフローが発生する領域が2ビット減少する。また、例えば、情報処理装置がこの固定小数点数を2ビットだけ正方向にシフト（左方向に2ビットシフト）すると、最上位ビットが18の位置にシフトし、飽和される領域が2ビット減少し、アンダーフローが発生する領域が2ビット拡張される。すなわち、情報処理装置1が小数点位置を2ビット右にシフトすると、飽和される領域が2ビット減少し、アンダーフローが発生する領域が2ビット拡張される。

【0070】

情報処理装置1は、学習実行中に、非符号となる最上位のビット位置の分布を得ることで、Dynamic Fixed Point演算における適正なシフト量、つまり、適切な固定小数点位置を直ちに決定できる。例えば、情報処理装置1は、飽和されるデータが指定の割合以下になるように固定小数点位置を決定すればよい。つまり、一例としては、情報処理装置1は、データのアンダーフローが所定の程度となることよりもデータの飽和が所定の程度となること優先して、固定小数点位置を決定できる。

【0071】

非符号となる最上位ビット位置の分布は、情報処理装置1のプロセッサ10（図9参照

10

20

30

40

50

）内の所定のレジスタ（統計情報レジスタともいう）内に積算される。プロセッサ 10 は、当該統計情報レジスタからの分布データの読み出し、書き込み、統計情報レジスタのクリア等の命令を実行する。したがって、統計情報レジスタには、前回のクリア命令の実行から現在までに命令実行の対象となった 1 以上の固定小数点数についての分布データが蓄積される。蓄積された分布データは、読み出し命令によってメモリに読み出される。なお、プロセッサ 10 は、クリア命令に代えて、統計情報レジスタへのロード命令の実行を可能とし、値 0 を統計情報レジスタにロードできるようにしてもよい。

（２）非符号となる最下位ビット位置の分布

【００７２】

非符号となる最下位ビット位置の分布とは、ビットが符号と異なる値となる最下位のビット位置をいう。例えば、ビットの並びを最上位ビットである bit[39] から最下位ビットである bit[0] までの並びとしたとき、符号ビット bit[39] と異なる bit[k] のうち、インデックス k が最小のものをいう。非符号となる最下位ビット位置の分布は図 8 B のように表現される。また、非符号となる最下位ビット位置の分布では、有効なデータが含まれる最下位ビットが把握される。

（３）非符号となる最上位ビット位置の最大値

【００７３】

非符号となる最上位ビット位置の最大値は、前回クリア命令が実行されたときから現在に至るまでに、命令実行の対象となった 1 以上の固定小数点数について符号ビットの値と異なる値となる最上位のビット位置のうちの最大値である。情報処理装置 1 は、非符号となる最上位のビット位置の最大値を Dynamic Fixed Point 演算における適正なシフト量、つまり、適切な小数点位置の決定に利用できる。

【００７４】

プロセッサは、上記統計情報レジスタからの上記最大値の読み出し、統計情報レジスタのクリア等の命令を実行する。したがって、統計情報レジスタには、前回のクリア命令の実行から現在までの最大値が蓄積され、読み出し命令によって最大値がメモリに読み出される。

（４）非符号となる最下位ビット位置の最小値

【００７５】

非符号となる最下位ビット位置の最小値は、前回クリア命令が実行されたときから現在に至るまでの 1 以上の固定小数点数について符号と異なる値となる最下位のビット位置のうちの最小値である。情報処理装置 1 は、非符号となる最下位ビット位置の最小値を Dynamic Fixed Point 演算における適正なシフト量、つまり、適切な小数点位置の決定に利用できる。

【００７６】

プロセッサ 10 は、統計情報レジスタからの上記最小値の読み出し、クリア等の命令を実行する。したがって、統計情報レジスタには、前回のクリア命令の実行から現在までの上記最小値が蓄積され、読み出し命令によってメモリに読み出される。

<構成>

【００７７】

図 9 に、情報処理装置 1 のプロセッサ 10 の構成を例示する。図 9 では、プロセッサ 10 とともに、命令用メモリ（IRAM）21 およびデータ用メモリ（DRAM）22 も例示されている。プロセッサ 10 は、single instruction multiple data（SIMD）型の演算処理装置である。

【００７８】

プロセッサ 10 は、プログラムカウンタ（PC）111、デコーダ（Decoder）112、レジスタファイル 12、ベクトル演算用の演算器 131、スカラ演算用の演算器（Arithmetic Logic Unit（ALU））141、およびベクトル演算用の演算器 131 の結果を加算するアキュムレータ 132 を有している。また、プロセッサ 10 は、ベクトル演算用の演算器 131、スカラ演算用の演算器 141、アキュムレータ 132 等の演算結果、お

よび、データ用メモリ22からの読み出し結果を選択する複数のセクタ101を有している。なお、図では、複数のセクタを総称してセクタ101と呼ぶ。また、複数のベクトル演算用の演算器を総称して演算器131と呼ぶ。

【0079】

さらに、プロセッサ10は、セクタ101によって選択されたデータからの統計情報を取得する統計情報取得部102、統計情報取得部102によって取得された統計情報を格納する統計情報格納部105を有している。統計情報取得部102と統計情報格納部105は、固定小数点数データに対する命令実行後の固定小数点数データ中のビットの分布についての統計情報を取得する取得回路の一例である。なお、図では、複数の統計情報取得部を総称して統計情報取得部102と呼ぶ。

10

【0080】

さらに、プロセッサ10は、セクタ101によって選択されたデータの固定小数点位置を変更するデータ変換部103を有している。また、図では、複数のデータ変換部を総称してデータ変換部103と呼ぶ。

【0081】

図のように、プログラムカウンタ111の指す命令用メモリ21のアドレスから、命令がフェッチされ、デコーダ112がフェッチされた命令をデコードする。なお、図では、命令のフェッチを実行する命令フェッチの制御部が省略されている。

【0082】

デコーダ112が命令をデコードすると、デコード結果に応じて、プロセッサ10の各部が制御される。例えば、デコードの結果がベクトル演算命令であった場合、レジスタファイル12のベクタレジスタのデータがベクトル演算用の演算器131に入力され、ベクトル演算が実行される。ベクトル演算用の演算器131の演算結果は、セクタ101を介して統計情報取得部102およびデータ変換部103に供給される。また、ベクトル演算用の演算器131の演算結果は、アキュムレータ132に入力され、ベクトル演算用の演算器131の演算結果が例えば、カスケードに加算される。アキュムレータ132の演算結果は、セクタ101を介して、統計情報取得部102およびデータ変換部103に供給される。

20

【0083】

また、例えば、デコードの結果、命令がスカラ演算命令であった場合には、レジスタファイル12のスカラ・レジスタのデータがスカラ演算用の演算器141に入力される。演算器141の演算結果は、アキュムレータ132の演算結果と同様、セクタ101を介して統計情報取得部102およびデータ変換部103に供給される。

30

【0084】

さらに、例えば、デコードの結果、命令がロード命令であった場合には、データ用メモリ22からデータが読み出され、セクタ101を介して、統計情報取得部102およびデータ変換部103に供給される。データ変換部103でデータ変換された結果は、レジスタファイル12のレジスタに格納される。

【0085】

また、デコードの結果、命令がDynamic Fixed Point演算を実行する命令であった場合、デコーダ112は、シフト量がデータ変換部103に供給されるように指示する。シフト量は、例えば、命令のオペランド（即値）、オペランドで指定されたレジスタ、オペランドで指定されたアドレス・レジスタの示すアドレスのデータ用メモリ22等から取得され、データ変換部103に供給される。図9の処理は、デコーダが命令から、固定小数点数データの小数点位置の指定（シフト量）を取得することの一例である。

40

【0086】

データ変換部103は、ベクトル演算の結果、スカラ演算の結果、アキュムレータ132の演算結果、あるいは、データ用メモリ22からの読み出し結果等で得られる固定小数点数データを、指定されたシフト量Sだけシフトする。また、データ変換部103は、シフトとともに、上位ビットの飽和処理および下位ビットの丸めを実行する。図12にデー

50

タ変換部の具体的構成を例示する。データ変換部103は、例えば入力を40ビットの演算結果とし、下位Sビットを小数部として丸めを行な丸め処理部と、算術シフトを実行するシフタと飽和処理を行なう飽和処理部を有する。

【0087】

丸め処理部は、下位S bitを小数部として丸めを行なう。Sが負の場合には、丸め処理部は、何も行なわない。丸めとしては、最近接丸め、0への丸め、正の無限大への丸め、負の無限大への丸め、乱数丸めなどが例示される。図でシフト量は、例えば、図9に例示されるように、デコーダが命令から取得するシフト量である。

シフト部は、Sが正の場合はS bit算術右シフト、Sが負の場合は、算術左シフト、すなわち、-S bit算術左シフトを行なう。飽和部は、シフト結果が $2E15-1$ （正の最大値）以上の場合は $2E15$ 、 $-2E15$ （負の最小値）以下の場合は $-2E15$ 、それ以外は入力の下位16bitを出力する。ここで、 $2E15$ は、 2 の 15 乗を表す。

【0088】

そして、データ変換部103は、左シフト時上位ビットの符号を維持し、符号ビット以外を飽和処理、すなわち、上位ビット廃棄し、下位ビットに0を埋め込む。また、データ変換部103は、右シフト時、上位ビット（符号ビットより下位のビット）に符号ビットを埋め込む。そして、データ変換部103は、上記のように丸め、シフト、飽和処理によって得られるデータを、レジスタファイル12のレジスタ等と同一のビット幅（例えば、16ビットのレジスタ）で出力する。データ変換部は固定小数点数データの小数点位置を更新する更新回路の一例である。

【0089】

したがって、プロセッサ10で実行されるコンピュータプログラムがDynamic Fixed Point演算を実行する命令のオペランドにシフト量を指定することで、プロセッサ10は、プログラム実行中に、固定小数点数の小数点位置を指定されたシフト量だけ更新する。

【0090】

また、デコードの結果、命令が統計情報取得を指示する命令（統計情報取得機能付き命令という）である場合、統計情報取得部102において、統計情報が取得され統計情報格納部105に格納される。ここで、統計情報は、上記で述べたように、（1）非符号となる最上位ビット位置の分布、（2）非符号となる最下位ビット位置の分布、（3）非符号となる最上位ビット位置の最大値、（4）非符号となる最下位ビット位置の最小値、またはこれらの組み合わせである。

【0091】

図10に、図9のプロセッサ10の回路ブロックを例示する。プロセッサ10は、制御ユニット11、レジスタファイル12、ベクトルユニット13、スカラユニット14を含む。制御ユニット11は、プログラムカウンタ111とデコーダ112を含む。レジスタファイルは、ベクトルレジスタファイル、ベクトル演算用アキュムレータレジスタ（Vector ACC）、スカラレジスタファイル、スカラ演算用アキュムレータレジスタ（ACC）を含む。ベクトルユニット13は、ベクトル演算用の演算器131、統計情報取得部102、データ変換部103を含む。スカラユニット14は、スカラ演算用の演算器141、統計情報取得部102、データ変換部103を含む。

【0092】

また、図10の構成例では、複数の統計情報取得部102からの統計情報を集約する統計情報集約部104が追加されている。また、統計情報格納部105はレジスタファイル12の一部となっている。また、命令用メモリ21は、メモリアンターフェース（Memory I/F）を介して制御ユニット11に接続される。また、データ用メモリ22は、メモリアンターフェース（Memory I/F）を介してベクトルユニット13およびスカラユニット14に接続される。

【0093】

図11Aにベクトルユニット13の詳細を例示する。図では、統計情報集約部104も例示されている。ベクトルユニット13は、ベクトル・レジスタVector Reg0、Vector Re

10

20

30

40

50

glのデータをベクトル演算用の演算器131-1で演算する。ベクトル演算用の演算器131-1の演算結果は、積和演算用のアキュムレータ132と、ベクトル演算用の演算器131-2に入力される。

【0094】

積和演算用のアキュムレータ132は、ベクトル演算用の演算器131-1の演算結果をスカラ加算し、スカラ演算用アキュムレータレジスタ(ACC)に格納する。ベクトル演算用の演算器131-2は、命令によって指定される演算モードに従って、ベクトル演算用の演算器131-1の演算結果、あるいは、ベクトル演算用アキュムレータレジスタ(Vector ACC)のデータ、あるいは、それらを加算した結果を出力する。

【0095】

セクタ101は、ベクトル演算用の演算器131-2の出力結果と、データ用メモリ22からの読み出し結果(Read Data 0, ..., Read Data 0)のいずれかを選択し、統計情報取得部102およびデータ変換部103に入力する。統計情報取得部102で取得された統計情報は、統計情報集約部104に入力される。また、データ変換部103でデータ変換されたデータは、図示しないセクタを介して、データ用メモリ22(Write Data0, ..., Write Data n)にストアされるか、ベクトルレジスタ(Vector Reg2)に保持される。

【0096】

図11Bにスカラユニット14の構成を例示する。スカラユニット14は、即値(Immediate)オペランドで得られるデータとスカラ・レジスタScalar Reg1からのデータの一方を選択するセクタ142と、セクタ142の選択結果とスカラ・レジスタScalar Reg0のデータを演算するスカラ用の演算器141を有する。スカラ用の演算器141の演算結果は、メモリインターフェース(Memory I/F)を介してデータ用メモリ22のアドレス(例えば、Addr)にストアされる。また、スカラ用の演算器141の演算結果は、セクタ101を介して、統計情報取得部102およびデータ変換部103に入力される。

【0097】

セクタ101は、スカラ用の演算器141の演算結果、スカラ・レジスタScalar Reg2のデータ、スカラ演算用アキュムレータレジスタ(ACC)のデータ、およびメモリインターフェース(Memory I/F)を介して読み出されたデータ(Read Data)の1つを選択する。セクタ101は、選択したデータを、統計情報取得部102およびデータ変換部103に入力する。統計情報取得部102は、セクタ101から入力されたデータから統計情報を取得し、統計情報集約部104に入力する。

【0098】

以下、統計情報取得部102のうち、非符号となる最上位ビット位置を取得するものを統計情報取得部102Aと呼ぶことにする。また、統計情報取得部102のうち、非符号となる最下位ビット位置を取得するものを統計情報取得部102Bと呼ぶことにする。また、統計情報集約部104のうち、統計情報取得部102によって取得されてビット位置を計数してビット位置に対するビットの分布を取得するものを統計情報集約部104Aと呼ぶことにする。また、統計情報集約部104のうち、ビット位置の最大値と最小値を取得するための前段階として統計情報取得部102によって取得されてビット位置を論理和演算するものを統計情報集約部104Bと呼ぶことにする。

【0099】

図13Aに、非符号となる最上位ビット位置を取得する統計情報取得部102Aの処理を例示する。図では、統計情報取得部102Aに含まれる非符号最上位ビット検出器による処理が例示されている。非符号最上位ビット検出器は、例えば、図13Bの真理値表で定義される論理回路である。図13Aの例では、符号ビットが0、「非符号となる最上位ビット位置」として示される位置のビットが1、このビットより上位側のビットがすべて0、かつ、このビットより下位側のビットはbの入力データ(in(40bit))が例示されている。ここで、bは0または1のいずれかである。統計情報取得部102Aは、この入力データを処理し、非符号となる最上位ビット位置を1とし、他のビット位置を0とする出

10

20

30

40

50

力データ (out(40bit)) を生成する。入力データが全ビット0、または、全ビット1の場合は、39bit目に1を、38bit以下に0を出力する。なお、以下の説明では、プロセッサ10の演算回路のデータは40ビットのデータで例示されるが、プロセッサ10の演算回路のデータが40ビットに限定される訳ではない。

【0100】

図13Bに非符号最上位ビット検出器の真理値表を例示する。図13Bのように、非符号最上位ビット検出器は、正数に対しては、ビットin[38]から下位ビット方向にビットをサーチして、最初に1になったビット位置の番号を2進数のビットパターンで出力する。また、非符号最上位ビット検出器は、負数に対しては、in[38]から下位ビット方向にビットをサーチして、最初に0になったビット位置の番号を2進数のビットパターンで出力する。また、入力データが全ビット0、または、全ビット1の場合は、非符号最上位ビット検出器は、39bit目に1を、38bit以下に0を出力する。

10

【0101】

図14に、非符号となる最上位ビット位置を取得する統計情報取得部102Aのハードウェア回路の構成を例示する。この回路では、符号ビットin[39]と他のビット (in[38]からin[0]) との排他論理和 (EXOR) が実行される。そうすると、符号ビットin[39]と同じ値を有するビットによる排他論理和値は0となり、符号ビットin[39]と異なる値を有するビットによる排他論理和値は1となる。

【0102】

今、仮にin[38]がin[39]と異なる値であった場合、排他論理和によって出力データのout[38]は1となる。一方、出力データのout[37]には、in[39]とin[37]の排他論理和値がANDゲートを介して入力される。このANDゲートの一方の入力には、in[39]とin[38]の排他論理和値を反転したビット値が入力されるため、in[39]とin[38]とが不一致の場合、in[39]とin[37]の排他論理和値によらず、ANDゲートの出力は0となる。

20

【0103】

同様に、出力データのout[36]には、in[39]とin[36]の排他論理和値が上記と同様のANDゲートを介して入力される。このANDゲートの一方の入力には、in[39]とin[38]の排他論理和値と、in[39]とin[37]との排他論理和値という2つの排他論理和値の論理和値 (ORゲートの出力) を反転したビット値が入力される。このため、in[39]とin[38]とが不一致の場合、in[39]とin[36]の排他論理和値によらず、ANDゲートの出力は0となる。以下、同様に、in[39]とin[i] (iは37以下) の排他論理和値によらず、ANDゲートの出力は0となる。

30

【0104】

一方、仮にin[38]がin[39]と同じ値であった場合、in[39]とin[38]の排他論理和値は0となり、出力データのout[38]は0となる。このため、in[39]とin[37]の排他論理和値が入力されるANDゲートは、in[39]とin[37]の排他論理和値に依存して、1または0を出力する。以下、同様に、out[i] (iは37以下) が出力されるANDゲートの論理否定付き入力、in[39]とin[j] (jはi+1以上、38以下) の排他的論理和が全て0の場合に0となり、in[39]とin[i] (iは37以下) の排他論理和値が1になると、out[i]には1が設定される。そのビット位置 (i) より下位のビットでは、out[i]が出力されるANDゲートの論理否定付き入力が1となるため、out[i]には0が設定される。したがって、図14の回路によって、非符号となる最上位ビット位置に1が設定され、他のビットが0に設定された出力データout(40bit)が取得される。図13、図14の非符号となる最上位ビット位置を取得する統計情報取得部102Aは、命令実行後の固定小数点数データ中の符号ビットと一致しない最上位のビットの位置を取得する回路の一例である。また、非符号となる最上位ビット位置に1が設定され、他のビットが0に設定された出力データout(40bit)は、符号ビットと一致しない最上位のビットの位置を真値 (1) で示すビット列の一例である。

40

【0105】

図15Aに、非符号となる最下位ビット位置を取得する統計情報取得部102Bの処理を例示する。図では、統計情報取得部102Bに含まれる非符号最下位ビット検出器によ

50

る処理が例示されている。非符号最下位ビット検出器は、例えば、図15Bの真理値表で定義される論理回路である。図15Aの例でも、正数での処理が例示される。この例では、符号ビットin[39]=0、「非符号となる最下位ビット位置」として示される位置のビットが1、このビットよりより下位側のビットがすべて0、このビットより上位側のビット（符号ビットを除く）はbの入力データが例示されている。ここで、bは0または1のいずれかであり、入力データは、in(40bit)である。統計情報取得部102Bは、この入力データを処理し非符号となる最下位ビット位置を1とし、他のビット位置を0とする出力データ(out(40bit))を生成する。入力データが全ビット0、または、全ビット1の場合は、39bit目に1を、38bit以下に0を出力する。

【0106】

10

図15Bに非符号最下位ビット検出器の真理値表を例示する。図15Bのように、非符号最下位ビット検出器は、正数に対しては、in[0]から上位ビット方向にビットをサーチして、最初に1になったビット位置の番号を2進数のビットパターンで出力する。また、非符号最下位ビット検出器は、負数に対しては、in[0]から上位ビット方向にビットをサーチして、最初に0になったビット位置の番号を2進数のビットパターンで出力する。また、また、入力データが全ビット0、または、全ビット1の場合は、非符号最下位ビット検出器は、39bit目に1を、38bit以下に0を出力する。

【0107】

図16に、非符号となる最下位ビット位置を取得する統計情報取得部102Bのハードウェア回路の構成を例示する。符号ビットin[39]が0の場合には、統計情報取得部102Bは、最下位ビットin[0]から上位側に向かって、ビットが1であるビット位置を探索すればよい。一方、符号ビットin[39]が1の場合には、データは補数となっているので、統計情報取得部102Bは、最下位ビットin[0]から上位側に向かって、ビットが0であるビット位置を探索すればよい。

20

【0108】

すなわち、この回路では、符号ビットin[39]と他のビット(in[0]からin[38])との排他論理和(EXOR)が実行される。そうすると、符号ビットin[39]と同じ値を有するビットによる排他論理和値は0となり、符号ビットin[39]と異なる値を有するビットによる排他論理和値は1となる。

【0109】

30

今、仮にin[0]がin[39]と異なる値であった場合、排他論理和によって出力データのout[0]は1となる。一方、出力データのout[1]には、in[39]とin[1]の排他論理和値がANDゲートを介して入力される。このANDゲートの一方の入力には、in[39]とin[0]の排他論理和値を反転したビット値が入力される。このため、in[39]とin[0]の排他論理和値が1の場合、in[39]とin[1]の排他論理和値によらず、ANDゲートの出力は0となる。

【0110】

同様に、出力データのout[2]には、in[39]とin[2]の排他論理和値が上記と同様のANDゲートを介して入力される。このANDゲートの一方の入力には、in[39]とin[0]の排他論理和値と、in[39]とin[1]の排他論理和値という2つの排他論理和値の論理和値(ORゲートの出力)を反転したビット値が入力される。このため、in[39]とin[0]の排他論理和値が1の場合、in[39]とin[2]の排他論理和値によらず、出力データのout[2]に値を出力するANDゲートの出力は0となる。以下、同様に、in[39]とin[i](iは1以上)の排他論理和値によらず、出力データのout[i]に値を出力するANDゲートの出力は0となる。

40

【0111】

一方、仮にin[0]がin[39]と同じ値であった場合、排他論理和によって出力データのout[0]は0となるため、in[39]とin[1]の排他論理和値が入力されるANDゲートは、in[39]とin[1]の排他論理和値に依存して、1または0を出力する。以下、同様に、out[i](iは1以上)が出力されるANDゲートの論理否定付き入力、in[39]とin[j](jは0以上、i-1以下)の排他的論理和が全て0の場合に0となり、in[39]とin[i](iは1以上)の排他論理和値が1になると、out[i]には1が設定される。また、そのビットより上位の出力データout[i]には0

50

が設定される。したがって、図16の回路によって、非符号となる最下位ビット位置に1が設定され、他のビットが0の出力データout(40bit)が取得される。図15、図16の非符号となる最下位ビット位置を取得する統計情報取得部102Bは、符号ビットと一致しない最下位のビットの位置を取得する回路の一例である。また、非符号となる最下位ビット位置に1が設定され、他のビットが0に設定された出力データout(40bit)は、符号ビットと一致しない最下位のビットの位置を真値(1)で示すビット列の一例である。

【0112】

図17は、統計情報取得部102が取得したデータからビットの分布を取得する統計情報集約部104Aの処理を例示する図である。図では、40ビットのデータが8個並列に処理されるSIMDデータからビットの分布を取得する処理が例示される。図17では、ハードウェア回路である統計情報集約部104Aの処理が、擬似コードで記述されている。

10

【0113】

すなわち、入力データは、8(行)×40(ビット)の配列データで例示されている。また、各行の40ビットの入力データは、非符号となる最上位ビット位置(図14の統計情報取得部102Aの出力)または非符号となる最下位ビット位置(図16の統計情報取得部102Bの出力)である。この処理では、40ビットの出力データoutについて、まず、全ビットがクリアされる。次に、入力データの配列in[j][i]の各列iの要素の値が、すべての行(j=0から7)に対して加算される。したがって、図13、図15とは異なり、図17の擬似コードでは、出力データ(配列要素)out[j]は、 $\log_2(\text{SIMDデータ数})$ ビット(図17の例では3ビット)の整数である。なお、図17では、SIMDデータ数(並列処理されるデータ数)は8であると想定したが、SIMDデータ数が8に限定される訳ではない。

20

【0114】

図18に、統計情報取得部102が取得したデータからビットの分布を取得する統計情報集約部104Aのハードウェア回路の構成を例示する。統計情報取得部102が取得したデータ(ここでは、統計取得0から統計取得(SIMDデータ数-1))がbit population count演算によって、8個の統計情報のiビット目(i=0から39)における1の個数がそれぞれカウントされる。入力データは、統計情報取得部102A(図13、図14)によって取得された非符号となる最上位ビット位置である。したがって、統計情報集約部104Aは、統計情報取得部102Aによって取得された非符号となるSIMDデータ個数分の最上位ビット位置に対して、各ビットの'1'の発生回数をカウントすることで最上位ビット位置の発生回数を計数する。統計情報集約部104Aは、計数結果を出力データout0からout39にそれぞれ格納する。

30

【0115】

また、入力データは、統計情報取得部102B(図15、図16)による非符号となる最下位ビット位置とすることもできる。統計情報集約部104Aは、統計情報取得部102Bによって取得されたSIMDデータ個数分の非符号となる最下位ビット位置に対して、各ビットの'1'の発生回数をカウントすることで最下位ビット位置の発生回数を計数する。統計情報集約部104Aは、計数結果を出力データout0からout39にそれぞれ格納する。すなわち、統計情報集約部104Aは、非符号となる最上位ビット位置または非符号となる最下位ビット位置のいずれをも処理可能である。

40

【0116】

また、図18でセクタ(SEL)は、bit population count演算器(Σ)と、スカラユニット14から取得されるデータを選択する。セクタ(SEL)に選択されたデータが、出力データout0からout39に出力される。したがって、統計情報取得部102がスカラユニット14で取得したデータは、一回のスカラユニット14の演算においては、加算されることなくそのまま出力データout0からout39に出力される。out0からout39は、統計情報格納部105に引き渡すデータ(図21の105A、図22の105Bの入力データin0からin39参照)である。図17、図18の統計情報集約部104Aは符号ビットと一

50

致しない最上位のビットの位置を複数の固定小数点数データに対して累積して計数する回路の一例である。図17、図18の統計情報集約部104Aは符号ビットと一致しない最下位のビットの位置を複数の固定小数点数データに対して累積して計数する回路の一例でもある。

【0117】

図19は、統計情報取得部102が取得したデータからビット位置の最大値と最小値を取得する前提として、論理和演算によってビット位置を集約する統計情報集約部104Bの処理を例示する図である。図19においても、図17と同様、40ビットのデータが8個並列に処理されるSIMDデータを処理が例示される。図19では、ハードウェア回路である統計情報集約部104Bの処理が、擬似コードで記述されている。

10

【0118】

この処理では、40ビットの出力データout[i](i=0, ..., 39)には、入力データの配列in[j][i]の各列をすべての行(j=0, ..., 7)について論理和演算(OR演算)した結果が入力される。したがって、図19の擬似コードでは、図17とは異なり、出力データ(配列要素)out[i](i=0, ..., 39)はビット列である。以上の処理の結果、出力データout[i](i=0, ..., 39)では、out[38]から下位ビット方向に向かって最初に1となるビット位置が最大ビット位置である。また、out[0]から上位ビット方向に向かって最初に1となるビット位置が最小ビット位置である。

【0119】

図20に、統計情報取得部102が取得したデータからビット位置の最大値と最小値を取得する前提として、論理和演算によってビット位置を集約する統計情報集約部104Bのハードウェア回路の構成を例示する。統計情報取得部102が取得したデータ(ここでは、統計取得0から統計取得SIMDデータ数-1)がORゲート(40ビット)によってOR演算される。また、図18でセクタ(SEL)は、論理和演算(OR)と、スカラユニット14から取得されるデータを選択する。セクタ(SEL)されたデータが、出力データoutに出力される。したがって、統計情報取得部102がスカラユニット14で取得したデータは、一回の演算においては、論理和演算されることなくそのまま出力データoutに出力される。outは、統計情報格納部105に引き渡すデータである。

20

【0120】

論理和演算によってビット位置を集約する統計情報集約部104Bは、符号ビットと一致しない最上位のビットの位置を真値で示すビット列を複数の固定小数点数データについて論理和演算によって累積する回路の一例である。論理和演算によってビット位置を集約する統計情報集約部104Bは、符号ビットと一致しない最下位のビットの位置を真値で示すビット列を複数の固定小数点数データについて論理和演算によって累積する回路の一例でもある。

30

【0121】

図21に、統計情報格納部105(図10参照)の具体例として、統計情報集約部104Aからの統計情報を専用レジスタに格納する統計情報格納部105Aの構成を例示する。図で、in39からin0は、図18のout39からout0に相当する統計情報集約部104からの統計情報を示す。また、sr39からsr0は、統計情報を格納するレジスタ値である。プロセッサ10は、書き込み命令によってセクタSELを介して、レジスタsr39からsr0のいずれか1つあるいは複数に初期値v39からv0を書き込む。ただし、プロセッサ10は、デコーダからのリセット信号によってレジスタsr39からsr0をリセットしても良い。プロセッサ10は、統計情報取得機能付き命令の実行ごとに、加算器を用いて統計情報を蓄積し、レジスタsr39からsr0に格納する。統計情報集約部104A(図17、図18)と、統計情報集約部104Aからの統計情報を専用レジスタに格納する統計情報格納部105A(図21)の組み合わせは符号ビットと一致しない最上位のビットの位置を複数の固定小数点数データに対して累積して計数する回路の一例である。統計情報集約部104A(図17、図18)と、統計情報集約部104Aからの統計情報を専用レジスタに格納する統計情報格納部105A(図21)の組み合わせは符号ビットと一致しない最下位のビットの位置を複数

40

50

の固定小数点数データに対して累積して計数する回路の一例でもある。また、プロセッサ 10 は、レジスタ (sr39 から sr0) のいずれかの 1 つあるいは複数の値を読み出し、読み出し命令で指定されたデータ用メモリに保存する、あるいは、読み出し命令で指定された汎用レジスタに格納する。

【0122】

図 22 は、統計情報集約部 104 の変形例であり、統計情報の格納先がデコーダ 112 からのインデックスで指定される回路の構成例である。図では、例えば、sr[j][i] (j=0, ..., k, i=0, ..., 39) の領域が確保され、index によって、レジスタファイルの行 j が指定される。

【0123】

プロセッサ 10 は、書き込み命令によってセクタ SEL を介して、index によって指定されるレジスタファイルの行 j の 1 つあるいは複数のレジスタに初期値を書き込む。ただし、プロセッサ 10 はデコーダ 112 からの制御信号により、index によって指定されるレジスタファイルの行 j をリセットしても良い。そして、プロセッサ 10 は、in39 から in0 の統計情報を加算器で蓄積し、index によって指定されるレジスタファイルの行 j に格納する。また、プロセッサ 10 は、デコーダ 112 からの制御信号により、index によって指定されるレジスタファイルの行 j から統計情報を読み出す。また、プロセッサ 10 は、index によって指定されるレジスタファイルの行 j のいずれかの 1 つあるいは複数の値を読み出し、読み出し命令で指定されたデータ用メモリに保存する、あるいは、読み出し命令で指定された汎用レジスタに格納する。

【0124】

図 23 は、図 19、図 20 で例示した統計情報集約部 104 B によって論理和演算されたビット位置の統計情報を蓄積するとともに蓄積された統計情報のビット位置の最大値を読み出す統計情報格納部 105 C のハードウェア回路の構成を例示する図である。統計情報格納部 105 C は、統計情報集約部 104 B によって論理和演算されたビット位置の統計情報を蓄積するレジスタ (sr) を有している。プロセッサ 10 は、書き込み命令 (write) によってセクタ (SEL) を介してレジスタ (sr) に初期値を書き込み可能である。ただし、プロセッサ 10 は、リセット信号によってレジスタ (sr) をリセットできるようにしてもよい。

【0125】

また、統計情報格納部 105 C は、統計情報集約部 104 B の論理和演算結果 (in) と、すでにレジスタ (sr) に蓄積された統計情報の論理和演算を実行し、セクタ (SEL) を介してレジスタ (sr) に論理和演算の結果を蓄積する。

【0126】

また、プロセッサ 10 は、読み出し命令によってセクタ (SEL) を介してレジスタ (sr) の値を読み出し、読み出し命令で指定されたデータ用メモリあるいは読み出し命令で指定された汎用レジスタに保存する。また、統計情報格納部 105 C は、Priority Encoder (MSB 優先) を有してもよい。Priority Encoder (MSB 優先) は、レジスタ (sr) に蓄積されたビット列中の最上位のビット 1 の位置 (−1 から 38) を 2 進数で出力する。例えば、入力データ in として全ビット 0 が入力されると、Priority Encoder (MSB 優先) は、"111111" (−1) を出力する。また、入力データ in として in0=1、他のビットがすべて 0 のデータが入力されると、Priority Encoder (MSB 優先) は、"000000" (0) を出力する。また、入力データ in として in0=x (0 または 1)、in1=1、他のビットがすべて 0 のデータが入力されると、Priority Encoder (MSB 優先) は、"000001" (1) を出力する。同様に、入力データ in として in0~in37 が x (0 または 1)、in38=1 のデータが入力されると、Priority Encoder (MSB 優先) は、"100110" (38) を出力する。プロセッサ 10 は、Priority Encoder (MSB 優先) を介して、統計情報集約部 104 B によって論理和演算されたビット位置の統計情報からビット位置の最大値を 2 進数値で取得できる。図 19、図 20 の論理和演算によってビット位置を集約する統計情報集約部 104 B (図 19、図 20) と統計情報格納部 105 C の組み合わせは、符号ビットと一致しない最上位のビットの位置を真値で

10

20

30

40

50

示すビット列を複数の固定小数点数データについて論理和演算によって累積し、累積したビット列中の最上位の真値の位置を取得する回路の一例である。

【0127】

図24は、図19、図20で例示した統計情報集約部104Bによって論理和演算されたビット位置の統計情報を蓄積するとともに蓄積された統計情報のビット位置の最小値を読み出す統計情報格納部105Dのハードウェア回路の構成を例示する図である。統計情報格納部105Dは、統計情報格納部105CのPriority Encoder(MSB優先)に代えてPriority Encoder(LSB優先)を有している。Priority Encoder(LSB優先)以外の統計情報格納部105Dの構成は、統計情報格納部105Cと同様であるから説明を省略する。

【0128】

Priority Encoder(MSB優先)は、レジスタ(sr)に蓄積されたビット列中の最下位のビット1の位置(−1から38)を2進数で出力する。例えば、入力データinとして全ビット0が入力されると、Priority Encoder(LSB優先)は、“111111”(−1)を出力する。また、入力データinとしてin0=1、他のビットがx(0または1)のデータが入力されると、Priority Encoder(LSB優先)は、“000000”(0)を出力する。また、入力データinとしてin0=0、in1=1、他のビット(in2からin38)がx(0または1)のデータが入力されると、Priority Encoder(LSB優先)は、“000001”(1)を出力する。同様に、入力データinとしてin0~in37が0、in38=1のデータが入力されると、Priority Encoder(LSB優先)は、“100110”(38)を出力する。プロセッサ10は、Priority Encoder(LSB優先)を介して、統計情報集約部104Bによって論理和演算されたビット位置の統計情報からビット位置の最小値を2進数値で取得できる。論理和演算によってビット位置を集約する統計情報集約部104B(図19、図20)と統計情報格納部105D(図24)の組み合わせは、符号ビットと一致しない最下位のビットの位置を真値で示すビット列を複数の固定小数点数データについて論理和演算によって累積し、累積したビット列中の最下位の真値の位置を取得する回路の一例である。

【0129】

図25は、図23の統計情報格納部105Cのレジスタ(sr)をレジスタファイルとした変形例の統計情報格納部105Eの構成を例示する図である。また、図26は、図24の統計情報格納部105Dのレジスタ(sr)をレジスタファイルとする変形例の統計情報格納部105Fの構成を例示する図である。図25、図26では、例えば、sr0,sr1,...,srj,...の領域が確保され、indexによって、レジスタファイルの行jが指定される。プロセッサ10は、デコーダ112からの制御信号により、あるいは、indexによって指定されるレジスタファイルの行jへの初期値の設定、統計情報(入力データin)の蓄積、行jからの統計情報の読み出し等を実行する。

<統計情報取得機能付き命令>

以下、統計情報取得機能付き命令の構成を例示する。ここでは、命令の構成方法として、方法1、方法2を例示する。

【0130】

(方法1) 方法1では、プロセッサ10の命令セットに、統計情報を取得する演算命令、ロード命令などが個別に追加される。以下の命令の構成は例示であり、プロセッサ10が下記命令に限定される訳ではない。

[vmul_s命令]

【0131】

命令の構成: vmul_s vs, vt, vd, imm

vmul_s命令は、プロセッサ10がベクトル・レジスタ vs と vt を乗算し imm ビットシフトし、丸め・飽和を行なって ベクトル・レジスタvd に格納するための命令である。プロセッサ10は、乗算結果のシフト前の統計情報を取得し、統計情報レジスタに累積する。統計情報レジスタは、例えば、図21のsr0からsr39、図22の統計情報レジスタファイルのsrj_i (j=0,1,..., i=0,1,...)、図23、図24のレジスタ(sr)、図25、図26の統計情報レジスタファイルのレジスタsr0,sr1,...等である。統計情報レジスタについて

は以下同様である。

【0132】

なお、プロセッサ10が内部のレジスタとデータ用メモリ22との間で統計情報を授受するためのread / write, load / store 命令も別途用意する。また、プロセッサ10が内部のレジスタをリセットするためのリセット命令を別途用意してもよい。以下、同様である。

[vld_s命令]

【0133】

命令の構成：vld_s ss, rt, rd

vld_s命令は、プロセッサ10がアドレス・レジスタ rs とrt を加算したアドレスからベクトルデータをロードして、ベクトル・レジスタ rd に格納するための命令である。プロセッサ10は、ロードしたデータの統計情報を取得し統計情報レジスタに累積する。

[read_acc_s命令]

【0134】

命令の構成：read_acc_s, rd, imm

read_acc_s命令は、プロセッサ10がアキュムレータ・レジスタ(40bit)のデータに対して、imm ビットシフトし、丸め・飽和を行なってスカラ・レジスタ rd に格納するための命令である。プロセッサ10は、アキュムレータレジスタの統計情報を取得し統計情報レジスタに累積する。

【0135】

(方法1の変形) 上記方法1に加え、命令のオペランドに、統計情報を格納するレジスタ・インデックス(s)を追加し、統計情報の格納先が指定できるようにしてもよい。以下、命令の構成を例示する。この命令のインデックス(s)は、図22、図25、図26等の統計情報レジスタファイルのindexを指定する。

vmul_s vs, vt, vd, imm, s

vld_s ss, st, vd, s

read_acc sd, imm, s

【0136】

(方法2) 図27に方法2による命令フォーマットを例示する。命令フォーマットを拡張し、統計情報を取得するか否かを指定するビットを追加するようにしてもよい。例えば、全命令について統計情報取得機能を追加してもよい。図25に、拡張された命令フォーマットを例示する。図のように、FLG=1は、統計情報の取得を指定する。一方、FLG=0は、統計情報の取得を指定せず、従来と同じ命令を示す。

【0137】

(方法2の変形) 図28に方法2の変形による命令フォーマットを例示する。命令フォーマットを拡張し、統計情報を格納するレジスタ・インデックスを指定するようにしてもよい。図でIDXは、レジスタ・インデックス(図22、図25、図26等のindex)を示す。ただし、IDX=000の場合には、プロセッサ10は統計情報を取得しない。図28の命令フォーマットをデコードし、レジスタ・インデックスIDXを取得する図9のデコーダ112は、統計情報の格納先の指定を取得するデコーダの一例である。

【0138】

情報処理装置1が以上のような統計情報取得機能付き命令を実行するため、統計情報取得機能付き命令を専用サブルーチンで実行するようにすればよい。コンパイラは、専用サブルーチンをコンパイルするときに、統計情報取得機能付き命令を実行形式に組み込めばよい。また、専用サブルーチンは、アセンブリ言語で作成され、統計上取得機能付き命令の実行が指定されるようにしてもよい。情報処理装置1が深層学習を実行するためのコンピュータプログラムにおいて、以上のような専用サブルーチンが呼び出されるようにすればよい。

<実行手順>

【0139】

図29に、本実施形態の情報処理装置1が深層学習を実行する場合の各層の処理の詳細を例示する。図29の処理を実行する情報処理装置1は1以上の層での演算により深層学習を実行する情報処理装置の一例である。図では、1層目の畳み込み層(Conv_1)、プーリング層(Pool_1)、および全結合層2(fc2)が例示されている。この例でも、比較例の図6と同様、1層目の畳み込み層(Conv_1)において、入力データ $ini=(x_0, \dots, x_n)$ と、重み (W_{ij}) との間で、畳み込み演算が実行され、 $z_0, \dots, z_j, \dots$ 等が計算される。また、 $z_0, \dots, z_j, \dots$ 等に対して、活性化関数が演算され、 $u_0, \dots, u_j, \dots$ 等が計算される。情報処理装置1は、この1層目の畳み込み層(Conv_1)での変数 $z_0, \dots, z_j, \dots$ 等での統計情報を内部の統計情報レジスタまたはレジスタ・インデックスで指定される統計情報レジスタファイルに蓄積する。なお、統計情報レジスタあるいはレジスタファイルが不足する場合は、各層の演算終了後に統計情報をメモリに退避し、次の演算開始前に統計情報を復帰させる。また、情報処理装置1は、同様に、この1層目の畳み込み層(Conv_1)での変数 $u_0, \dots, u_j, \dots$ 等での統計情報を内部のレジスタまたはレジスタ・インデックスで指定される統計情報レジスタファイルに蓄積する。本実施形態の統計情報は、比較例のようなオーバーフロー回数とは異なり、上記で説明した(1)非符号となる最上位ビット位置の分布、(2)非符号となる最下位ビット位置の分布、(3)非符号となる最上位ビット位置の最大値、(4)非符号となる最下位ビット位置の最小値、または、これらのいずれか2以上の組み合わせである。

【0140】

1層目の畳み込み層(Conv_1)での演算結果は、1層目のプーリング層(Pool_1)から、さらに上位層へ伝播される。このとき、情報処理装置1は、2層目以上の層でも同様の演算と、統計情報の蓄積を実行する。そして、情報処理装置1は、全結合層2(fc2)において、得られた演算結果と、正解データ T_i との差分値からエラーの変数 $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ を算出する。そして、情報処理装置1は、エラーの変数 $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ を基に、逆方向の伝播処理を実行する。すなわち、情報処理装置1は、下位の階層(例えば、結合層1(fc1))の出力でのエラーの変数 $\delta_0^{(6)}, \dots, \delta_j^{(6)}, \dots, \delta_n^{(6)}$ と重みの変数 W_{ij} の次の重みへの差分値 ΔW_{ij} を算出する。このとき、情報処理装置1は、エラー変数 $\delta_0^{(7)}, \dots, \delta_j^{(7)}, \dots$ およびエラーの変数 $\delta_0^{(6)}, \dots, \delta_j^{(6)}, \dots, \delta_n^{(6)}$ での統計情報を1層目の畳み込み層(Conv_1)と同様に蓄積する。また、情報処理装置1は、差分値 ΔW_{ij} での統計情報も蓄積する。また、情報処理装置1は、重みの変数 W_{ij} での統計情報を蓄積する。

【0141】

そして、情報処理装置1は、全結合層1(fc1)から1層目の畳み込み層(Conv_1)まで、エラーを逆方法に伝播させながら、重みの差分を計算する。情報処理装置1は、以上のような順伝播と逆伝播を k 組の入力データを用いて、 k 回繰り返す。情報処理装置1は、このような k 回のミニバッチ終了後、各変数の統計情報を基に、各変数の固定小数点位置を更新する。

【0142】

図30に、本実施形態の情報処理装置1による学習処理のフローチャートを例示する。この処理では、情報処理装置1は、初期小数点位置を決定する(W E 1)。情報処理装置1は、過去の実験値、実績値、あるいは、ユーザ指定により各変数の初期小数点位置を決定すればよい。

【0143】

そして、情報処理装置1は、回数 k を0に初期化する。また、情報処理装置1は、プログラム内の統計情報を格納する変数を初期化する(W E 2)。そして、情報処理装置1は学習終了の条件が充足されたか否かを判定する(W E 3)。学習が終了するのは、全結合層(fc2)でのエラーが基準値以下となるか、または、学習回数が規定の最大値に達した場合である。

【0144】

学習終了の条件が充足されない場合、情報処理装置1は、次のミニバッチを実行する。このとき、情報処理装置1は、各レイヤの各変数の統計情報を統計情報レジスタあるいは統計情報レジスタファイルに累積する。統計情報を累積は、上記で説明したように、プロセッサ10のハードウェアで実行される。そして、情報処理装置1は、変数kをカウントアップする（WE4）。

【0145】

そして、情報処理装置1は、回数kが更新間隔に達したか否かを判定する（WE5）。回数kが更新間隔に達していない場合、情報処理装置1は、WE3の処理に戻る。一方、回数kが更新間隔に達した場合、情報処理装置1は、各種の統計情報が蓄積された統計情報を統計情報レジスタあるいは統計情報レジスタファイルあるいは統計情報をメモリに退避した領域から統計情報を読み出す。そして、情報処理装置1は、読み出された統計情報を基に各レイヤの各変数の小数点位置を更新する（WE6）。

10

【0146】

このとき、情報処理装置1は、例えば、飽和するデータとアンダーフローするデータの全体のデータ数に対する比率が目標条件を充足するように、小数点位置を決定すればよい。また、情報処理装置1は、飽和するデータの全体のデータに対する比率とアンダーフローするデータの全体のデータに対する比率の両方が目標条件を充足することが困難である場合、一方を優先してもよい。また、情報処理装置1は、飽和が発生しない上限のビット位置と、有効なデータが存在する下限のビット位置の平均を、固定小数点数で表現可能な範囲の中央を位置づけて、小数点位置を設定してもよい。さらに、情報処理装置1は、回数kを0に初期化し、さらに統計情報を格納するレジスタをリセットする。そして、情報処理装置1は、C3の処理に戻る。

20

【0147】

WE4で各レイヤの各変数の統計情報を累積することと、WE6で各種の統計情報が蓄積されたレジスタあるいはレジスタファイルから統計情報を読み出すことは、1以上の層の少なくとも1つの層において前記演算の命令を含む命令実行時の固定小数点数データ中のビットの分布についての統計情報を取得することの一例である。また、WE6で、読み出された統計情報を基に各レイヤの各変数の小数点位置を更新することは、ビットの分布についての統計情報に基づいて固定小数点数データの小数点位置を更新することの一例である。

30

【0148】

図31に、データが固定小数点数の範囲に収まらない場合の小数点位置更新例を示す。図は、例えば、非符号となる最上位ビット位置の分布である。今、固定小数点数がQ5.10（整数部5桁、小数点以下10桁）で記述され、飽和される領域A1、表現可能な領域A2、および、アンダーフローが発生する領域A3が形成されているとする。この例では、飽和される領域A1とアンダーフローが発生する領域A3は、白抜きの度数分布で例示されている。また、表現可能な領域は斜線のハッチングパターンで示されている。また、この例では、アンダーフローが発生する領域の度数分布が飽和する領域の度数分布より高く、バランスが悪い。一方、小数点位置を2ビット下側に移動し、Q3.12（整数部3桁、小数点以下12桁）としても、飽和する領域のデータ数を全データ数で除算した値は、目標の基準値未満となる。そこで、情報処理装置1は、小数点位置をQ5.10からQ3.12設定しなおし、深層学習を継続すればよい。

40

【0149】

図32および図33に、データが固定小数点数の範囲に収まる場合の小数点位置更新例を示す。図32も図31と同様、例えば、非符号となる最上位ビット位置の分布である。また、図33は非符号となる最下位ビットの分布である。図32では、現在のビット精度（Q3.12フォーマット）で表現可能な範囲に当該変数のデータが収まっている。しかしながら、図33では-1ビット目まで有効な値があり、これらの値が丸められていることがわかる。そこで、情報処理装置1は、最上位ビット位置分布でデータ個数が1以上となる最大値の11ビット目と、最下位ビットの分布でデータ個数が1以上となる最小値の-1ビ

50

ット目までが、表現可能な範囲の中心となるよう、小数点位置をQ 3. 1 2 からQ 1. 1 4 設定しなおし、深層学習を継続すればよい。

＜実施形態の効果＞

【0150】

情報処理装置1は、深層学習のミニバッチ実行時、各レイヤの各変数の統計情報をレジスタあるいはレジスタファイルに累積する。そして、情報処理装置1は、蓄積された統計情報を基に各レイヤの各変数の小数点位置を更新することができる。すなわち、プロセッサ10が、ビット分布の統計上情報を取得する。ここで、統計情報は、命令実行時の（1）非符号となる最上位ビット位置の分布、（2）非符号となる最下位ビット位置の分布、（3）非符号となる最上位ビット位置の最大値、（4）非符号となる最下位ビット位置の最小値、またはこれらの組み合わせ等である。したがって、情報処理装置1が深層学習実行時に、データの統計情報取得のための深層学習プログラム中のオーバーヘッドがなくなるため、実用的な時間で動的固定小数点演算を実現可能となる。

10

【0151】

すなわち、本実施形態では、情報処理装置1のプロセッサ10が統計情報取得機能付き命令を実行するとともに、演算結果をビットシフトし、丸め・飽和を行なってレジスタに格納する命令を実行する。したがって、情報処理装置1はビット分布を示す統計情報を取得するオーバーヘッドを少なくできる。また、ビット分布を示す統計情報から直ちに適正なビットシフト、すなわち、小数点位置を決定できる。つまり、情報処理装置1のように、小数点位置を1ビットずつずらして適正な小数点位置を試行し、次の演算で結果を確認するという手順ではなく、ビット分布を示す統計情報から、直ちに小数点位置を決定できる（図31、図32）。したがって、情報処理装置1は、比較例のような小数点位置が不適切な状態で学習処理が繰り返される恐れがすくない。また、情報処理装置1は、学習結果の精度劣化につながり、深層学習の収束が遅くなる恐れが少ない。

20

【0152】

また、情報処理装置1は、深層学習を固定小数点数で精度良く実行できるので、データ転送量や演算器の回路規模が削減できる。その結果として、情報処理装置1は、消費電力を削減できる。削減できる回路規模は以下のように試算できる。例として32bit浮動小数点演算を実行する回路を、16bit固定小数点演算を実行する回路に限定した場合、データ量（メモリ、および、データ転送量）が半減される。また、演算回路規模としては、積和演算での回路規模は約半分となる（FPGAで試算）。

30

【0153】

また、プロセッサ10は、統計情報取得部102Aによって命令実行後の固定小数点数データ中の符号ビットと一致しない最上位のビットの位置を取得できる。また、プロセッサ10は、統計情報集約部104Aによって符号ビットと一致しない最上位のビットの位置を複数の固定小数点数データに対して累積して計数することができる。また、プロセッサ10は、統計情報集約部104Bと、統計情報格納部105C（あるいは105E）によって、符号ビットと一致しない最上位のビットの位置を真値で示すビット列を複数の固定小数点数データについて論理和演算によって累積する。そして、プロセッサ10は、累積したビット列中の最上位の真値の位置を取得することができる。

40

【0154】

また、プロセッサ10は、統計情報取得部102Bによって命令実行後の固定小数点数データ中の符号ビットと一致しない最下位のビットの位置を取得できる。また、プロセッサ10は、統計情報集約部104Aによって符号ビットと一致しない最下位のビットの位置を複数の固定小数点数データに対して累積して計数することができる。また、プロセッサ10は、統計情報集約部104Bと、統計情報格納部105D（あるいは105F）によって、符号ビットと一致しない最下位のビットの位置を真値で示すビット列を複数の固定小数点数データについて論理和演算によって累積する。そして、プロセッサ10は、累積したビット列中の最下位の真値の位置を取得することができる。以上のような構成により、プロセッサ10は、上記統計情報を取得できる。

50

<本実施形態の情報処理装置について>

【0155】

情報処理装置1は、プロセッサ10、命令用メモリ21、およびデータ用メモリ22を備える。情報処理装置1は、命令用メモリ21に実行可能に展開されたコンピュータプログラムにしたがって深層学習を実行する。プロセッサ10は、Central Processing Unit (CPU)、Microprocessor (MPU)とも呼ばれる。プロセッサ10は、は、単一のプロセッサに限定される訳ではなく、マルチプロセッサ構成であってもよい。また、単一のソケットで接続される単一のプロセッサ10がマルチコア構成を有していても良い。深層学習を実行する情報処理装置1の少なくとも一部の処理は、プロセッサ10以外の装置、例えば、Digital Signal Processor(DSP)、Graphics Processing Unit (GPU)、数値演算プロセッサ、ベクトルプロセッサ、画像処理プロセッサ等の専用プロセッサで行われても良い。また、深層学習を実行する情報処理装置1の少なくとも一部の処理は、集積回路(IC)、その他のディジタル回路であっても良い。また、深層学習を実行する情報処理装置1の少なくとも一部には、アナログ回路が含まれても良い。集積回路は、LSI, Application Specific Integrated Circuit (ASIC), プログラマブルロジックデバイス (PLD)を含む。PLDは、例えば、Field-Programmable Gate Array(FPGA)を含む。上記各部は、プロセッサと集積回路との組み合わせであっても良い。組み合わせは、例えば、マイクロコントローラ(MCU)、SoC (System-on-a-chip)、システムLSI、チップセットなどと呼ばれる。

10

<記録媒体>

20

【0156】

コンピュータその他の機械、装置（以下、コンピュータ等）に上記いずれかの機能を実現させるプログラムをコンピュータ等が読み取り可能な記録媒体に記録することができる。そして、コンピュータ等に、この記録媒体のプログラムを読み込ませて実行させることにより、その機能を提供させることができる。

【0157】

ここで、コンピュータ等が読み取り可能な記録媒体とは、データやプログラム等の情報を電氣的、磁氣的、光学的、機械的、または化学的作用によって蓄積し、コンピュータ等から読み取ることができる記録媒体をいう。このような記録媒体のうちコンピュータ等から取り外し可能なものとしては、例えばフレキシブルディスク、光磁気ディスク、Compact Disc (CD) - Read Only Memory (ROM)、CD-Recordable (R)、Digital Versatile Disk (DVD)、ブルーレイディスク、Digital Audio Tape (DAT)、8mmテープ、フラッシュメモリなどのメモリカード等がある。また、コンピュータ等に固定された記録媒体としてハードディスク、ROM (リードオンリーメモリ) 等がある。さらに、Solid State Drive (SSD) は、コンピュータ等から取り外し可能な記録媒体としても、コンピュータ等に固定された記録媒体としても利用可能である。

30

【符号の説明】

【0158】

- 1 情報処理装置
- 10 プロセッサ
- 11 制御ユニット
- 12 レジスタファイル
- 13 ベクトルユニット
- 14 スカラユニット
- 21 命令用メモリ
- 22 データ用メモリ
- 101 セレクタ
- 102 統計情報取得部
- 103 データ変換部
- 104 統計情報集約部

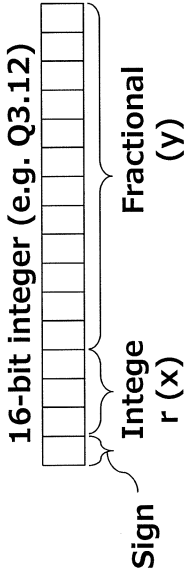
40

50

- 1 0 5 統計情報格納部
- 1 1 1 プログラムカウンタ
- 1 1 2 デコーダ
- 1 3 1 ベクトル演算用の演算器
- 1 3 2 積和演算用のアキュムレータ
- 1 4 1 スカラ用の演算器

【図 1】

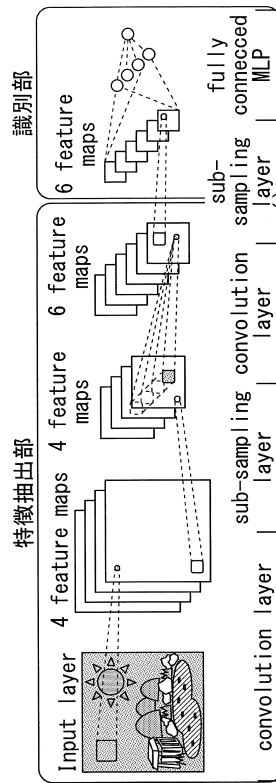
【図 2】



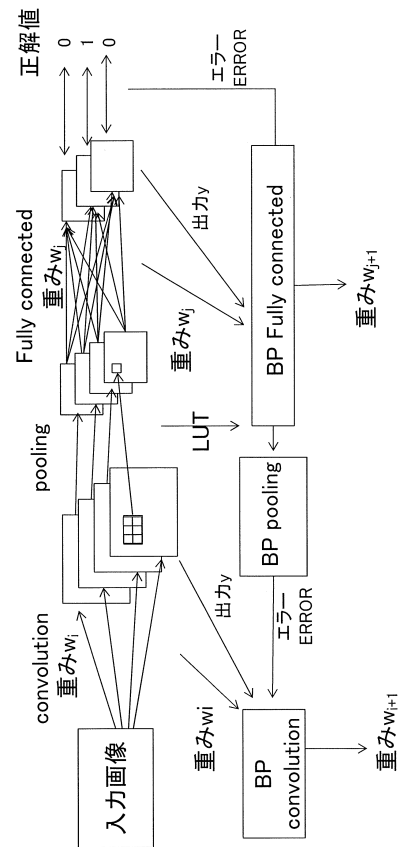
Format	Prop.	Up.	PI MNIST	MNIST	CIFAR-10	SVHN
Goodfellow <i>et al.</i> (2013a)	32	32	0.94%	0.45%	11.68%	2.47%
Single precision floating point	32	32	1.05%	0.51%	14.05%	2.71%
Half precision floating point	16	16	1.10%	0.51%	14.14%	3.02%
Fixed point	20	20	1.39%	0.57%	15.98%	2.97%
Dynamic fixed point	10	12	1.28%	0.59%	14.82%	4.95%

従来技術 論文

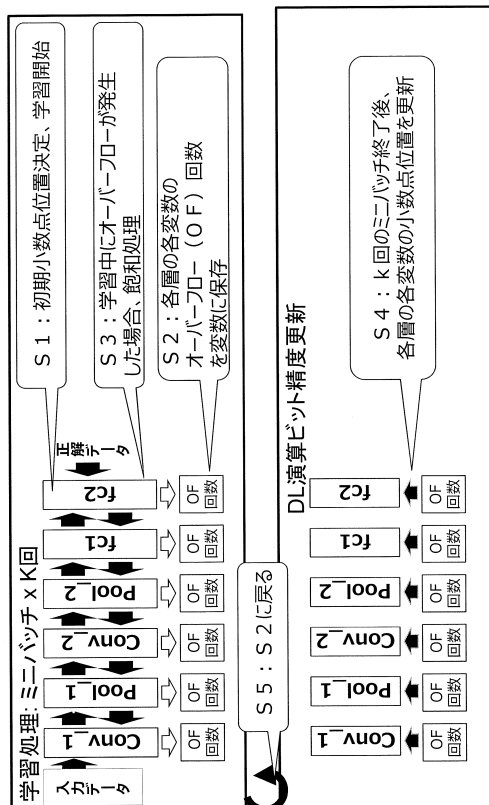
【図 3】



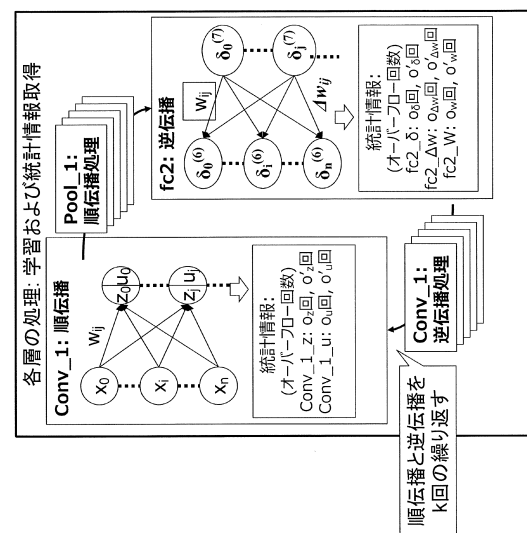
【図 4】



【図 5】

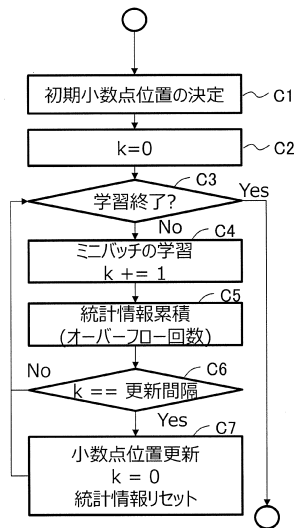


【图 6】

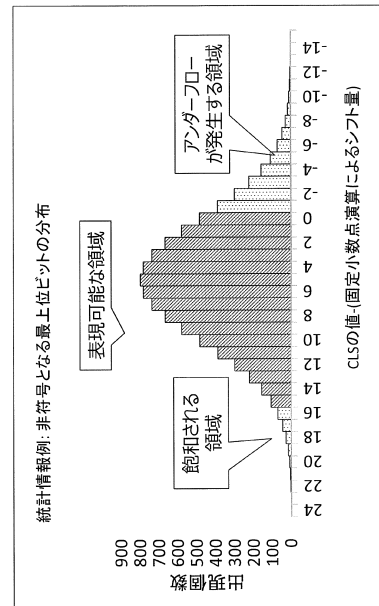


比較例

【図 7】

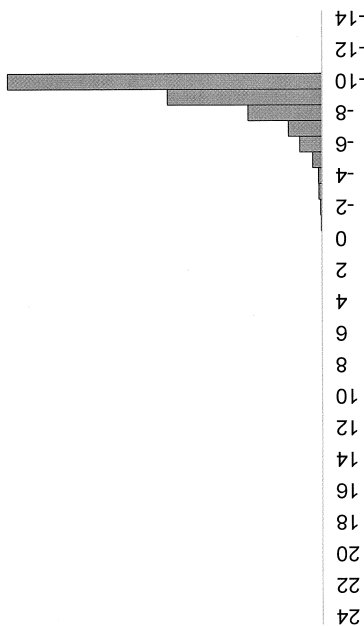


【図 8 A】

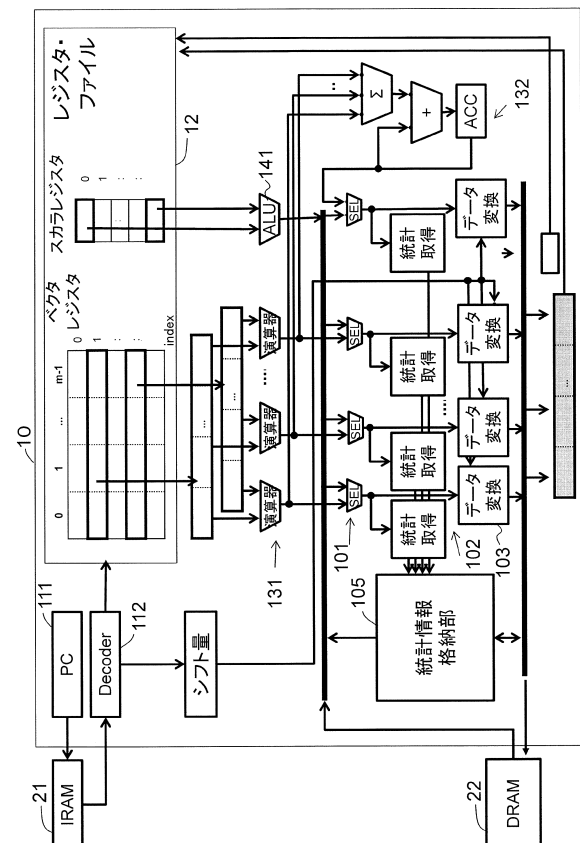


実施形態

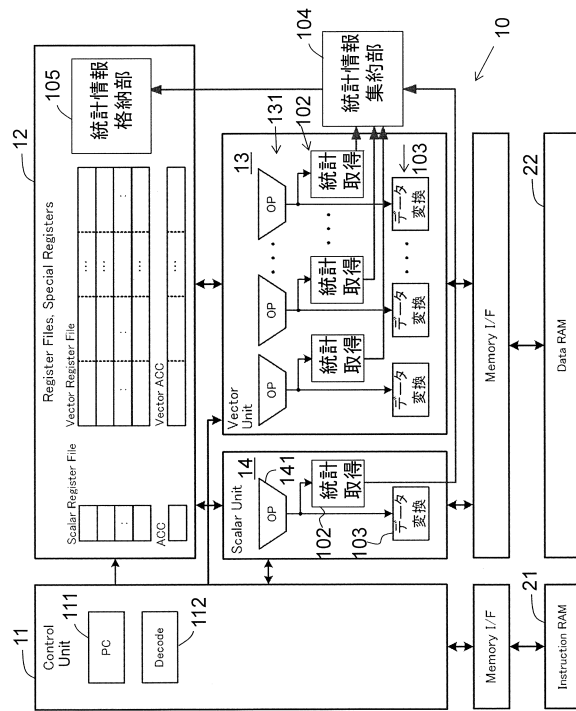
【図 8 B】



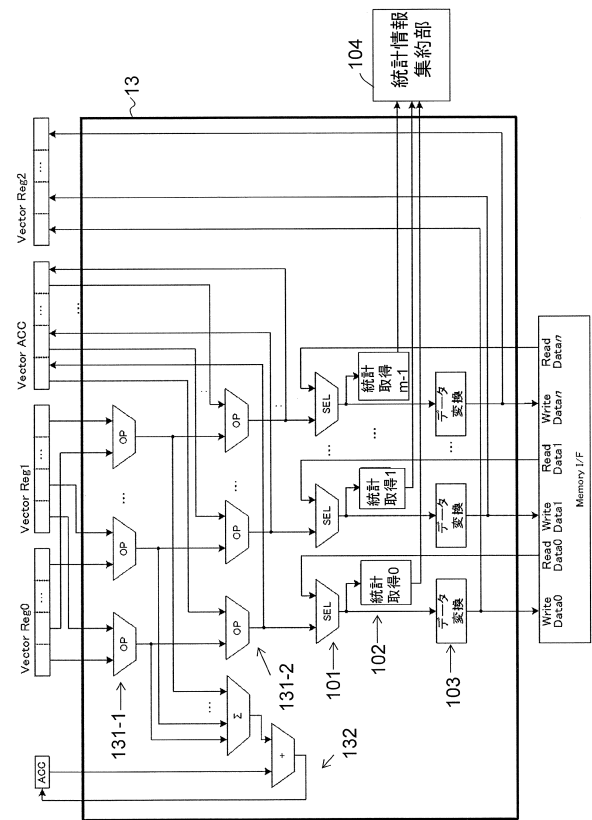
【図 9】



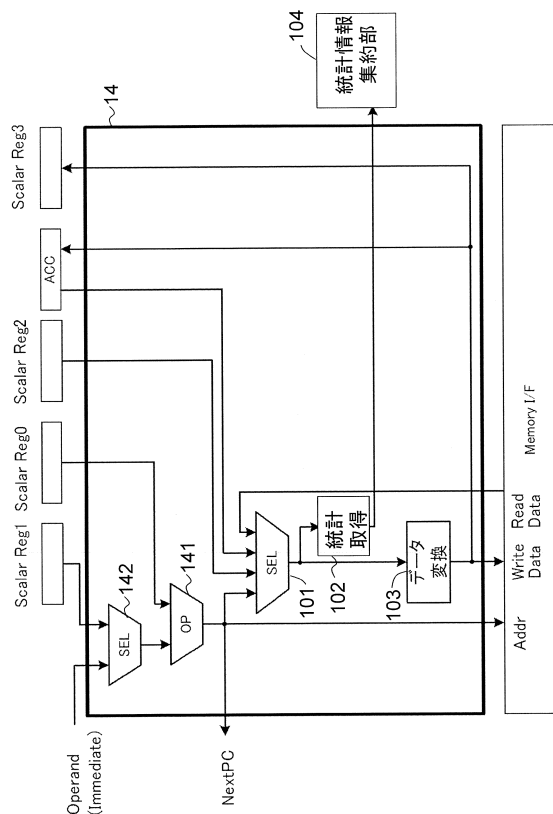
【図 10】



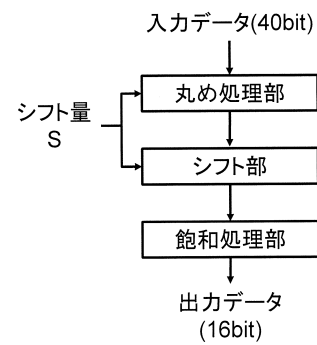
【図 11 A】



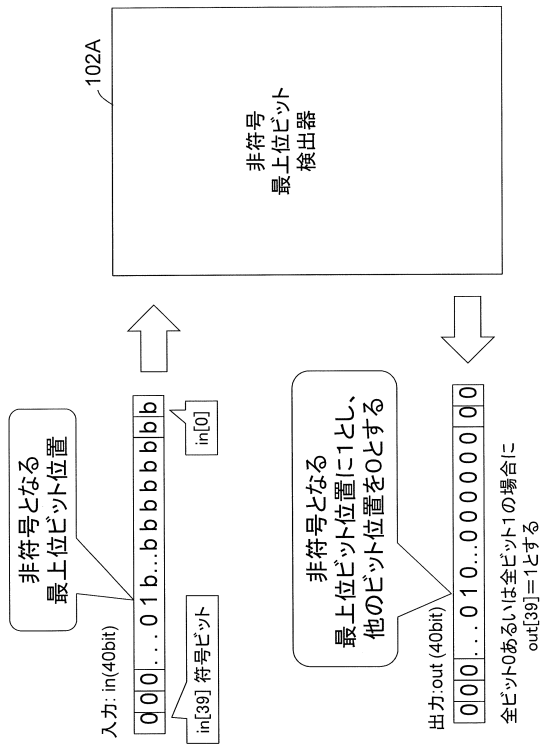
【図 11 B】



【図 12】



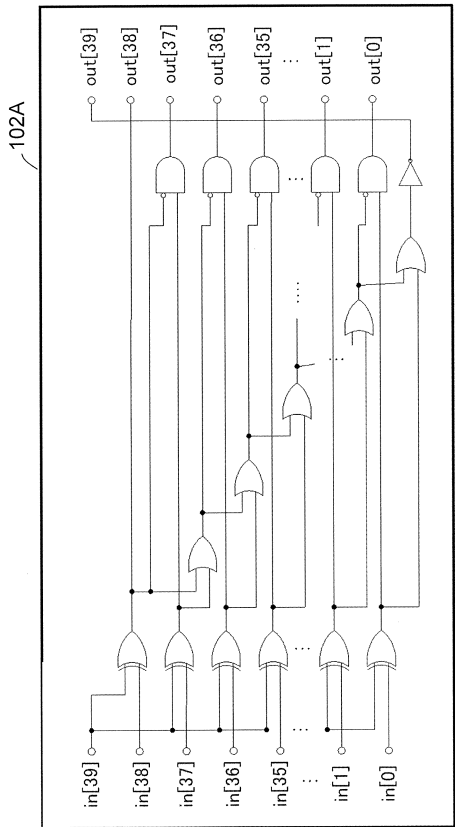
【図 1 3 A】



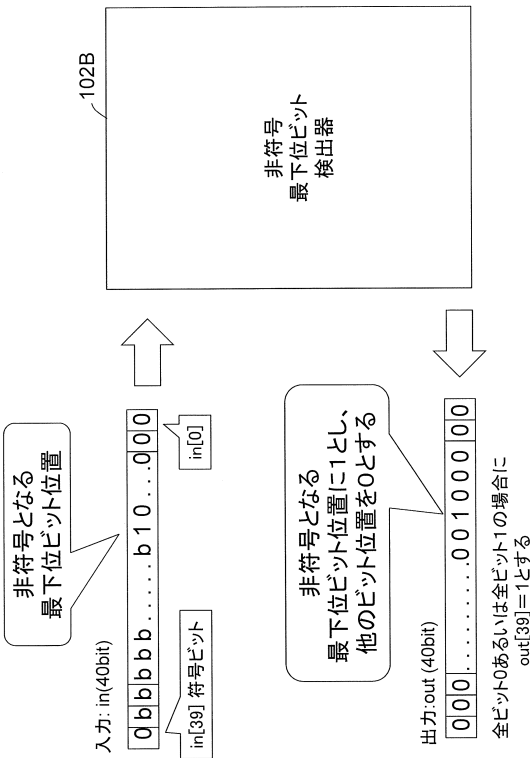
【図 1 3 B】

input				output			
in[39]	in[38]	in[37]	in[36]	...	in[1]	in[0]	out
0	0	0	0	...	0	0	0x8000000000
1	1	1	1	...	1	1	0x8000000000
0	1	b	b	...	b	b	0x4000000000
1	0	b	b	...	b	b	0x4000000000
0	0	1	b	...	b	b	0x2000000000
1	1	0	b	...	b	b	0x2000000000
0	0	0	1	...	b	b	0x1000000000
1	1	1	0	...	b	b	0x1000000000
:	:	:	:	:	:	:	:
0	0	0	0	...	1	b	0x0000000002
1	1	1	1	...	0	b	0x0000000002
0	0	0	0	...	0	1	0x0000000001
1	1	1	1	...	1	0	0x0000000001

【図 1 4】



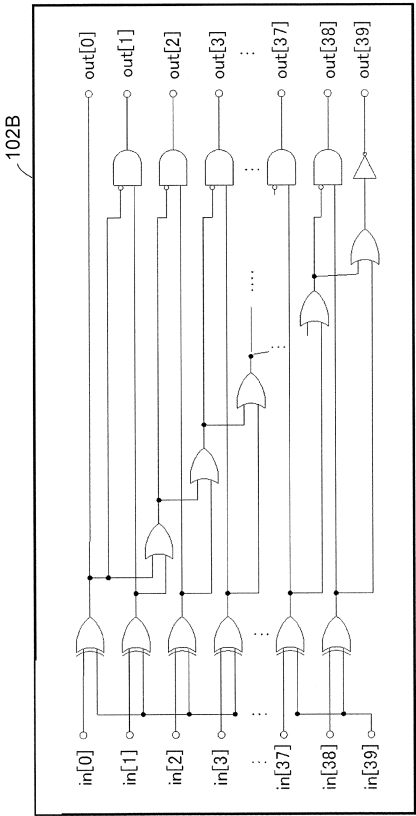
【図 1 5 A】



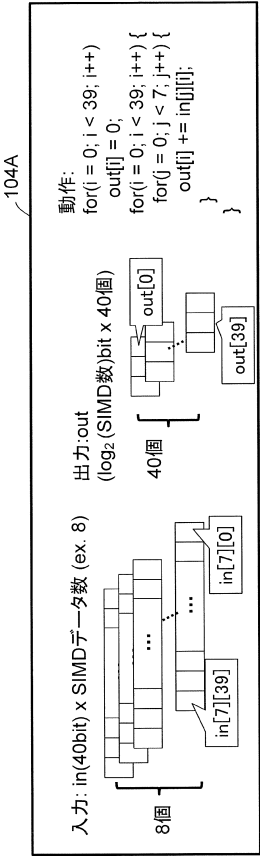
【図 15 B】

input					output		
in[39]	in[38]	in[37]	in[36]	...	in[1]	in[0]	out
0	b	b	b	...	b	1	0x0000000001
1	b	b	b	...	b	0	0x0000000001
0	b	b	b	...	1	0	0x0000000002
1	b	b	b	...	0	1	0x0000000002
:	:	:	:	:	:	:	:
0	b	b	1	...	0	0	0x1000000000
1	b	b	0	...	1	1	0x1000000000
0	b	1	0	...	0	0	0x2000000000
1	b	0	1	...	1	1	0x2000000000
0	1	0	0	...	0	0	0x4000000000
1	0	1	1	...	1	1	0x4000000000
0	0	0	0	...	0	0	0x8000000000
1	1	1	1	...	1	1	0x8000000000

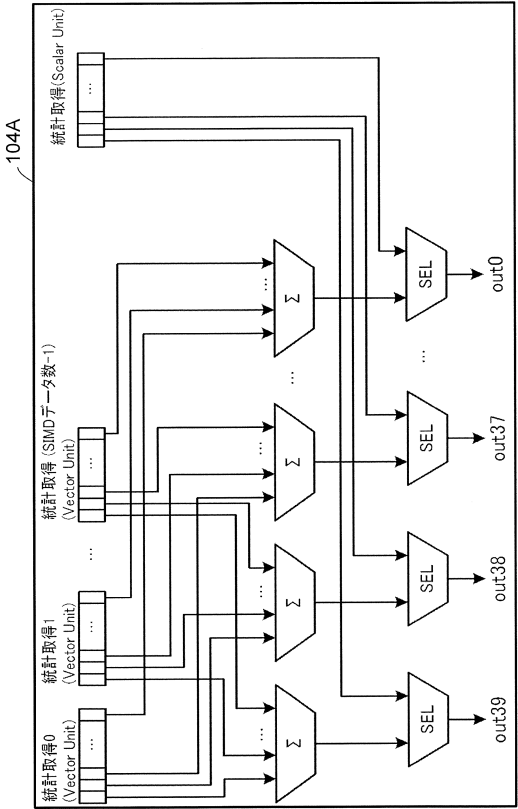
【図 16】



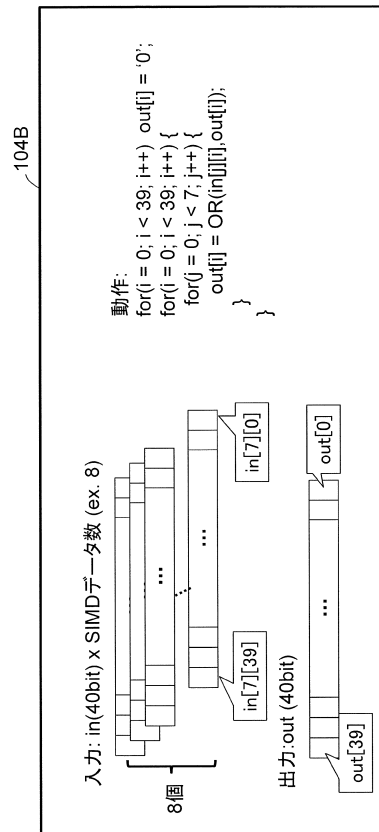
【図 17】



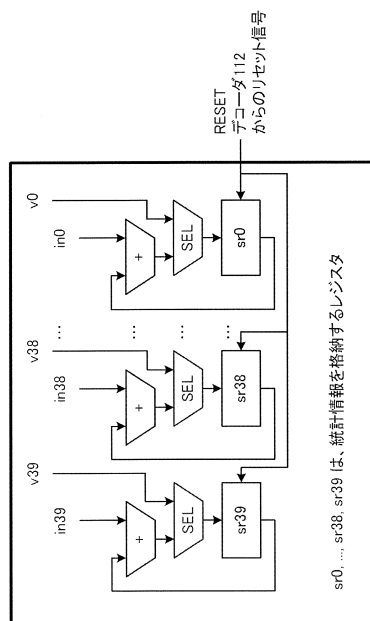
【図 18】



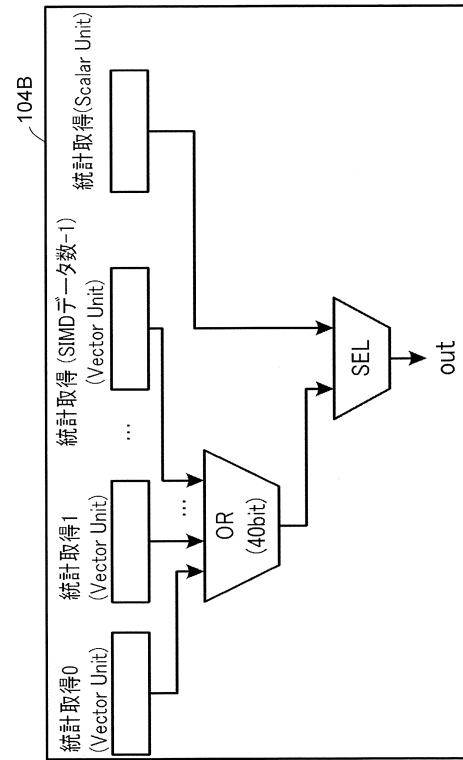
【図 19】



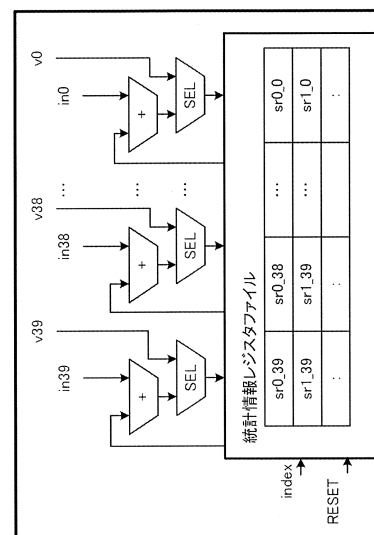
【図 21】



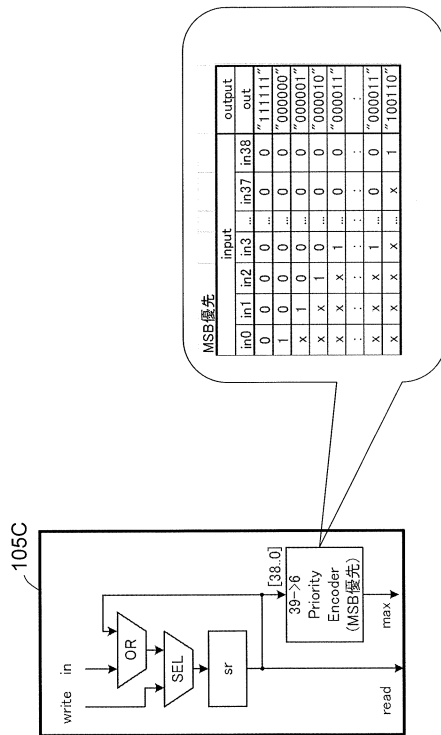
【図 20】



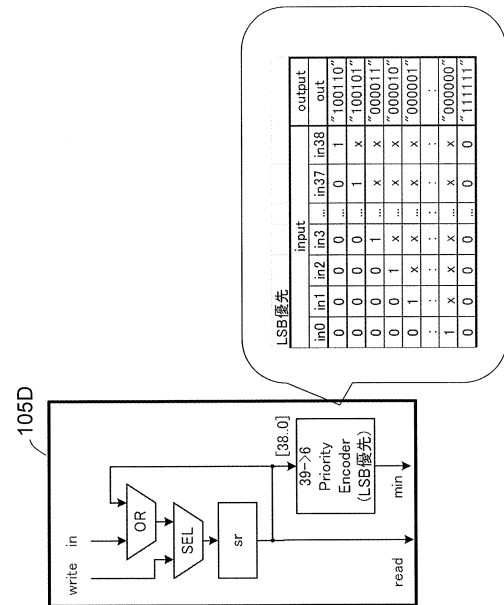
【図 22】



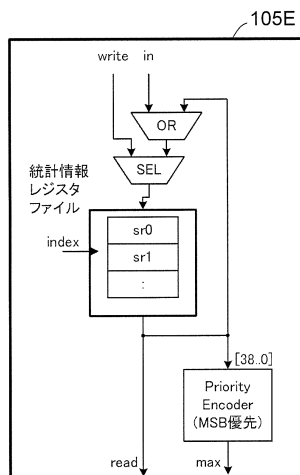
【図 2 3】



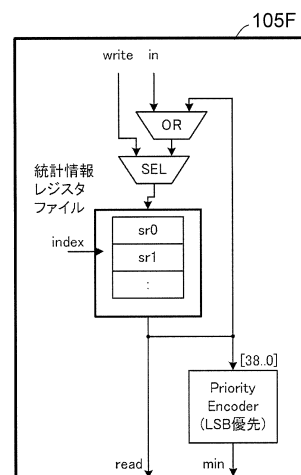
【図 2 4】



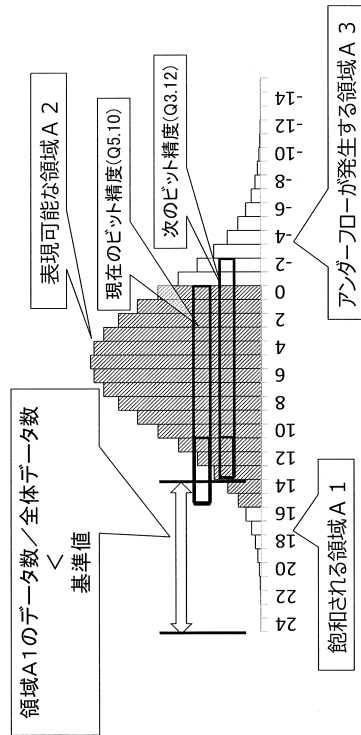
【図 2 5】



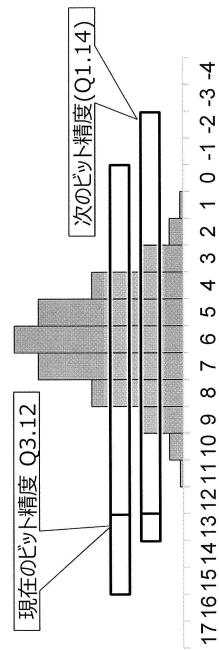
【図 2 6】



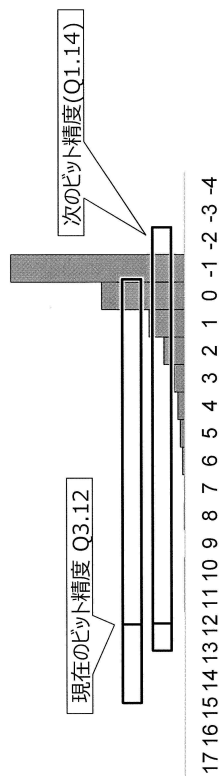
【図 3 1】



【図 3 2】



【図 3 3】



フロントページの続き

- (72)発明者 石原 輝雄
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 依田 勝洋
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 野津 隆弘
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 漆原 孝治

- (56)参考文献 特開平10-333883 (JP, A)
特開2006-048590 (JP, A)
特開平07-084975 (JP, A)
桑高 克直, 汎用並列ニューロコンピュータを用いたニューラルネットワーク演算, 電子情報通信学会技術研究報告, 日本, 社団法人電子情報通信学会, 1996年12月14日, 第96巻, 第430号, pp.79-86

- (58)調査した分野(Int.Cl., DB名)
- | | |
|---------|-----------|
| G 0 6 F | 7 / 4 9 9 |
| G 0 6 F | 9 / 3 1 5 |
| G 0 6 N | 3 / 0 6 3 |