

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7623398号  
(P7623398)

(45)発行日 令和7年1月28日(2025.1.28)

(24)登録日 令和7年1月20日(2025.1.20)

(51)国際特許分類

F I

G O 6 N 3/084(2023.01) G O 6 N 3/084

G O 6 N 3/063(2023.01) G O 6 N 3/063

G O 6 N 20/00 (2019.01) G O 6 N 20/00

請求項の数 20 (全29頁)

|                   |                             |          |                        |
|-------------------|-----------------------------|----------|------------------------|
| (21)出願番号          | 特願2022-561510(P2022-561510) | (73)特許権者 | 314015767              |
| (86)(22)出願日       | 令和3年2月9日(2021.2.9)          |          | マイクロソフト テクノロジー ライセン    |
| (65)公表番号          | 特表2023-521975(P2023-521975  |          | シング, エルエルシー            |
|                   | A)                          |          | アメリカ合衆国 ワシントン州 9 8 0 5 |
| (43)公表日           | 令和5年5月26日(2023.5.26)        |          | 2 レッドモンド ワン マイクロソフト    |
| (86)国際出願番号        | PCT/US2021/017215           |          | ウェイ                    |
| (87)国際公開番号        | WO2021/211193               | (74)代理人  | 100118902              |
| (87)国際公開日         | 令和3年10月21日(2021.10.21)      |          | 弁理士 山本 修               |
| 審査請求日             | 令和6年1月25日(2024.1.25)        | (74)代理人  | 100106208              |
| (31)優先権主張番号       | 16/851,847                  |          | 弁理士 宮前 徹               |
| (32)優先日           | 令和2年4月17日(2020.4.17)        | (74)代理人  | 100196508              |
| (33)優先権主張国・地域又は機関 | 米国(US)                      |          | 弁理士 松尾 淳一              |
|                   |                             | (74)代理人  |                        |
|                   |                             |          | 中村 彰吾                  |
|                   |                             | (72)発明者  | シー, ジンウェン              |

最終頁に続く

(54)【発明の名称】 メモリ要件が低減されたデュアルモーメンタム勾配最適化

(57)【特許請求の範囲】

【請求項1】

勾配(gradient)オブティマイザと、L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法であって、Lは1より大きい整数であり、  
選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットを前記メモリから取り出すステップと、  
前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットを前記メモリから取り出すステップと、  
前記選択された記憶フォーマットを有するモーメンタム値の前記第1のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、  
前記選択された記憶フォーマットを有するモーメンタム値の前記第2のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換するステップと、  
前記勾配オブティマイザを用いて、モーメンタム値の前記第3のセットおよびモーメンタム値の前記第4のセットを使用して勾配最適化を実行するステップと、  
を含み、前記勾配オブティマイザは、連続するバーストサイクルが前記勾配オブティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成される、

方法。

【請求項 2】

前記選択された記憶フォーマットは、縮小単精度(reduced single precision)フォーマットを含む、

請求項 1 に記載の方法。

【請求項 3】

前記訓練フォーマットは、単精度フォーマットまたは倍精度(double precision)フォーマットを含む、

請求項 1 に記載の方法。

【請求項 4】

勾配(gradient)オブティマイザと、L 層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法であって、L は 1 より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第 1 のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第 2 のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第 1 のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 3 のセットに変換するステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第 2 のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 4 のセットに変換するステップと、

前記勾配オブティマイザを用いて、モーメンタム値の前記第 3 のセットおよびモーメンタム値の前記第 4 のセットを使用して勾配最適化を実行するステップと、

を含み、前記選択された記憶フォーマットを有するモーメンタム値の前記第 1 のセットを、前記訓練フォーマットを有するモーメンタム値の前記第 3 のセットに変換するステップは、単精度フォーマットのモーメンタム値を形成するために余分なゼロビットをパディング(padding)するステップを含む、

方法。

【請求項 5】

勾配(gradient)オブティマイザと、L 層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法であって、L は 1 より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第 1 のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第 2 のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第 1 のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 3 のセットに変換するステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第 2 のセットを、前記勾配オブティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 4 のセットに変換するステップと、

前記勾配オブティマイザを用いて、モーメンタム値の前記第 3 のセットおよびモーメンタム値の前記第 4 のセットを使用して勾配最適化を実行するステップと、

を含み、

勾配最適化の次の反復のためのモーメンタム値の第 5 のセットおよびモーメンタム値の第 6 のセットを生成するステップと、モーメンタム値の前記第 5 のセットおよびモーメンタム値の前記第 6 のセットのそれぞれを記憶する前に、それぞれのモーメンタム値のそれ

10

20

30

40

50

それに関連する符号ビットおよび7つの最上位ビットのみを記憶することによってモーメンタム値の前記第5のセットおよびモーメンタム値の前記第6のセットのそれぞれを前記記憶フォーマットへ変換するステップと、

をさらに含む、方法。

【請求項6】

勾配最適化を実行するステップは、アダプティブモーメンタム推定アルゴリズムを実装するステップを含む、

請求項1に記載の方法。

【請求項7】

前記勾配オプティマイザは、フィールドプログラマブルゲートアレイ（FPGA）を使用して実装される、

請求項6に記載の方法。

【請求項8】

勾配オプティマイザを含むシステムであって、

L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリを含み、Lは1より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第1のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第2のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換し、

前記勾配オプティマイザを使用して、モーメンタム値の前記第3のセットとモーメンタム値の前記第4のセットを用いて勾配最適化を実行する、

ように構成され、前記勾配オプティマイザは、連続するバーストサイクルが前記勾配オプティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成される、

記載のシステム。

【請求項9】

前記選択された記憶フォーマットは、縮小単精度フォーマットを含む、

請求項8に記載のシステム。

【請求項10】

勾配オプティマイザを含むシステムであって、

L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリを含み、Lは1より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第1のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第2のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換し、

前記勾配オプティマイザを使用して、モーメンタム値の前記第3のセットとモーメンタム

10

20

30

40

50

値の前記第4のセットを用いて勾配最適化を実行する、  
ように構成され、

前記訓練フォーマットは、単精度フォーマットまたは倍精度フォーマットを含み、

単精度フォーマットのモーメンタム値または倍精度フォーマットのモーメンタム値を形成するために、余分なゼロビットをパディングするようにさらに構成される、

システム。

【請求項11】

勾配オプティマイザを含むシステムであって、

L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリを含み、Lは1より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットを前記メモリから取り出し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第1のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換し、

前記選択された記憶フォーマットを有するモーメンタム値の前記第2のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換し、

前記勾配オプティマイザを使用して、モーメンタム値の前記第3のセットとモーメンタム値の前記第4のセットを用いて勾配最適化を実行する、

ように構成され、

勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成し、モーメンタム値の前記第5のセットおよびモーメンタム値の前記第6のセットのそれぞれを記憶する前に、それぞれのモーメンタム値のそれぞれに関連する符号ビットおよび7つの最上位ビットのみを記憶することによってモーメンタム値の前記第5のセットおよびモーメンタム値の前記第6のセットのそれぞれを前記記憶フォーマットへ変換する、ようにさらに構成される、

システム。

【請求項12】

アダプティブモーメンタム推定アルゴリズムを実装するように構成される、

請求項8に記載のシステム。

【請求項13】

前記勾配オプティマイザは、フィールドプログラマブルゲートアレイ（FPGA）を使用して実装される、

請求項8に記載のシステム。

【請求項14】

勾配オプティマイザと、L層を含むニューラルネットワークモデルに関連する重みおよびモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法であって、Lは1より大きい整数であり、

選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応するモーメンタム値の第1のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有する、前記ニューラルネットワークモデルの層に対応するモーメンタム値の第2のセットを前記メモリから取り出すステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第1のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、

前記選択された記憶フォーマットを有するモーメンタム値の前記第2のセットを、前記勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4の

10

20

30

40

50

セットに変換するステップと、

前記勾配オプティマイザを用いて、モーメンタム値の前記第3のセットおよびモーメンタム値の前記第4のセットを使用して勾配最適化を実行するステップと、

勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成するステップと、

符号ビットのみを記憶し、モーメンタム値の前記第5のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の前記第5のセットを、前記記憶フォーマットを有するモーメンタム値の第7のセットに変換するステップと、

符号ビットのみを記憶し、モーメンタム値の前記第6のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の前記第6のセットを、前記記憶フォーマットを有するモーメンタム値の第8のセットに変換するステップと、

を含む、方法。

【請求項15】

前記勾配オプティマイザは、連続するバーストサイクルが前記勾配オプティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成される、

請求項14に記載の方法。

【請求項16】

前記選択された記憶フォーマットは、縮小単精度(reduced single precision)フォーマットを含む、

請求項14または15に記載の方法。

【請求項17】

前記訓練フォーマットは、単精度フォーマットまたは倍精度(double precision)フォーマットを含む、

請求項14または15に記載の方法。

【請求項18】

勾配最適化を実行するステップは、アダプティブモーメンタム推定アルゴリズムを実装するステップを含む、

請求項14または15に記載の方法。

【請求項19】

前記訓練フォーマットは、32ビット浮動小数点フォーマットを含み、前記記憶フォーマットは、8ビット浮動小数点フォーマットを含む、

請求項14または15に記載の方法。

【請求項20】

前記勾配オプティマイザは、フィールドプログラマブルゲートアレイ(FPGA)を使用して実装される、

請求項14または15に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本願発明の一実施例は、例えば、メモリ要件が低減されたデュアルモーメンタム勾配最適化に関する。

【背景技術】

【0002】

[0001]ニューラルネットワークモデルを含む訓練されたモデルは、読解、言語翻訳、画像認識、または音声認識などの複雑なタスクを実行するために使用される。自然言語処理(NLP)、回帰型ニューラルネットワーク(RNN)、畳み込みニューラルネットワーク(CNN)、長短期記憶(LSTM)ニューラルネットワーク、またはゲート回帰ユニット(GRU)に基づくものなどの機械学習サービスは、そのような複雑なタスクを実行するために展開されている。これらのタイプのモデルは様々なアーキテクチャおよび手法を用いて訓練されているが、訓練を行うための基盤となるアーキテクチャおよび手法には

10

20

30

40

50

引き続き改良が求められている。

【発明の概要】

【発明が解決しようとする課題】

【0003】

本願発明の一実施例は、例えば、メモリ要件が低減されたデュアルモーメンタム勾配最適化に関する。

【課題を解決するための手段】

【0004】

[0002]一例において、本開示は、勾配オプティマイザと、L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法に関し、Lは1より大きい整数である。本方法は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットをメモリから取り出すステップと、を含んでもよい。本方法は、選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換するステップと、をさらに含んでもよい。本方法は、勾配オプティマイザを用いて、モーメンタム値の第3のセットおよびモーメンタム値の第4のセットを使用して勾配最適化を実行するステップをさらに含み得る。

【0005】

[0003]別の例では、本開示は、勾配オプティマイザを含むシステムに関する。システムは、L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリを含んでもよく、Lは1より大きい整数である。本システムは、(1) 選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットをメモリから取り出し、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットをメモリから取り出し、(2) 選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換し、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換し、(3) 勾配オプティマイザを使用して、モーメンタム値の第3のセットとモーメンタム値の第4のセットを用いて勾配最適化を実行する、ように構成されてもよい。

【0006】

[0004]さらに別の例では、本開示は、勾配オプティマイザと、L層を含むニューラルネットワークモデルに関連する重みおよびモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法に関し、Lは1より大きい整数である。本方法は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメンタム値の第1のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメンタム値の第2のセットをメモリから取り出すステップと、をさらに含んでもよい。本方法は、選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換するステップと、をさらに含んでもよい。本方法は、勾配オプティマイザを用いて、モーメンタム値の第3のセットおよびモーメンタム値の第4のセットを使用して勾配最適化を実

行するステップをさらに含んでもよい。本方法は、勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成するステップをさらに含んでもよい。本方法は、符号ビットのみを記憶し、モーメンタム値の第5のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の第5のセットを、記憶フォーマットを有するモーメンタム値の第7のセットに変換するステップをさらに含んでもよい。本方法は、符号ビットのみを記憶し、モーメンタム値の第6のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の第6のセットを、記憶フォーマットを有するモーメンタム値の第8のセットに変換するステップをさらに含んでもよい。

【0007】

【0005】この発明の概要は、詳細な説明において以下でさらに説明される概念の選択を簡略化した形で紹介するために提供される。この発明の概要は、特許請求される主題の重要な特徴または本質的な特徴を特定することを意図しておらず、特許請求される主題の範囲を限定するために使用することも意図していない。

【0008】

【0006】本開示は、例として説明されており、添付の図によって限定されるものではなく、その中で、同様の参照は同様の要素を示している。図中の要素は、単純化および明確化のために図示されており、必ずしも縮尺通りに描かれていない。

【図面の簡単な説明】

【0009】

【図1】【0007】一実施例によるデュアルモーメンタム勾配オプティマイザ（DMGO）を含む人工知能訓練システム（AITS）を示す図である。

【図2】【0008】一実施例によるデュアルモーメンタム勾配オプティマイザ（DMGO）を示す図である。

【図3】【0009】圧縮されたモーメンタム値が1仮数ビットの切り捨てから23仮数ビットの切り捨てまで掃引されている重み更新誤差の誤差限界曲線（EB（M））を含むグラフである。

【図4】【00010】訓練の精度に対する低減された精度のモーメンタム値の効果を示すシミュレーションされた収束曲線を有するグラフ400を示す図である。

【図5】【00011】一実施例による、ハードウェア支援勾配オプティマイザ（HAGO）を含む訓練システムを示す図である。

【図6】【00012】一実施例による、デュアルモーメンタム勾配オプティマイザ（DMGO）を含むハードウェア支援勾配オプティマイザ（HAGO）のブロック図である。

【図7】【00013】一実施例による、図6のHAGOに関連するタイミング図を示す。

【図8】【00014】他の実施例による勾配最適化システムを示す図である。

【図9】【00015】一実施例によるデュアルモーメンタム勾配オプティマイザ（DMGO）のための方法のフロー図である。

【図10】【00016】一実施例によるデュアルモーメンタム勾配オプティマイザ（DMGO）のための他の方法のフロー図である。

【発明を実施するための形態】

【0010】

【00017】本開示で開示される例は、メモリ要件が低減されたデュアルモーメンタム勾配最適化に関する。特定の例は、より少ないメモリを必要とする、ニューラルネットワークモデルを含むモデルの分散訓練に関する。特定の例の訓練システムでは、各ワーカ（例えば、GPU、DSP、またはASICアクセラレータ）は、訓練データのサブセット（例えば、ミニバッチ）を取り、勾配を計算し、これは、さらに平均化／低減されてモデルのパラメータ（例えば、重さ）を更新する。重みの更新は、勾配を計算するバックプロパゲーションのステップの後に適用される。一例として、ネットワーク（またはネットワークの一部）の重みに関する損失関数の勾配が計算される。そして、その勾配を最適化手法に与え、勾配を利用して、損失関数を最小化するように重みを更新する。バックプロパゲー

10

20

30

40

50

ションの目標は、実際の出力を目標出力に近づけるようにネットワーク内の各重み（または少なくとも重みの一部）を更新し、それによって各出力ニューロンおよびネットワーク全体としての誤差を最小にすることである。

【0011】

[00018]特定の例では、訓練されたモデルは、クラウドコンピューティング環境に展開することができる。クラウドコンピューティングは、構成可能なコンピューティングリソースの共有プールへのオンデマンドネットワークアクセスを可能にするためのモデルを指す場合がある。例えば、クラウドコンピューティングは、構成可能なコンピューティングリソースの共有プールへのユビキタスかつ便利なオンデマンドアクセスを提供するために市場で採用され得る。構成可能なコンピューティングリソースの共有プールは、仮想化によって迅速にプロビジョニングされ、管理作業またはサービスプロバイダとのやりとりが少ない状態でリリースされ、その後適宜スケーリングされる。クラウドコンピューティングモデルは、例えば、オンデマンドセルフサービス、幅広いネットワークアクセス、リソースプーリング、迅速な弾力性、測定サービスなど、様々な特性で構成することができる。クラウドコンピューティングモデルは、例えば、ハードウェア・アズ・ア・サービス（「HaaS」）、ソフトウェア・アズ・ア・サービス（「SaaS」）、プラットフォーム・アズ・ア・サービス（「PaaS」）、インフラストラクチャ・アズ・ア・サービス（「IaaS」）などの様々なサービスモデルの公開に使用され得る。クラウドコンピューティングモデルは、プライベートクラウド、コミュニティクラウド、パブリッククラウド、ハイブリッドクラウドなどの異なる展開モデルを用いて展開することもできる。

【0012】

[00019]回帰ニューラルネットワーク（RNN）、畳み込みニューラルネットワーク（CNN）、長短期記憶（LSTM）ニューラルネットワーク、またはゲート回帰ユニット（GRU）に基づくものなどの様々な機械学習サービスは、本開示で説明する例を使用して実装され得る。一例では、単語、文章、画像、動画、または他のそのようなコンテンツ／情報などのサービス関連コンテンツまたは他の情報は、ベクトル表現に変換されてもよい。ベクトル表現は、RNN、CNN、LSTM、またはGRUなどの技術に対応し得る。一例では、ニューラルネットワークモデルは、多くの層を含んでもよく、各層は、ニューラルネットワークのオフライン訓練を介して得られた係数または定数の形で表される重みの行列またはベクトルとして符号化されてもよい。

【0013】

[00020]LSTMを例にとると、LSTMネットワークは、RNN層または他のタイプの層の繰り返しのシーケンスを含んでもよい。LSTMネットワークの各層は、所定の時間ステップにおける入力、例えば、前の時間ステップからの層の状態を消費してもよく、新しいセットの出力または状態を生成してもよい。LSTMを使用する場合、コンテンツの単一のチャンクは、単一のベクトルまたは複数のベクトルに符号化されてもよい。一例として、単語または単語の組み合わせ（例えば、フレーズ、文、または段落）は、単一のベクトルとして符号化されてもよい。各チャンクは、LSTMネットワークの個々の層（例えば、特定の時間ステップ）に符号化されてもよい。LSTM層は、以下のような一組の式を用いて記述されてもよい。

【0014】

$$\begin{aligned} i_t &= \rho(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \rho(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \rho(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t \tanh(c_t) \end{aligned}$$

[00021]この例では、各LSTM層の内部で、入力と隠れ状態は、ベクトル演算（例えば、ドット積、内積、またはベクトル加算）と非線形関数（例えば、シグモイド、双曲線、および接線）の組み合わせを使用して処理され得る。ある場合には、最も計算集約的な演算は、密な行列－ベクトルおよび行列－行列の乗算ルーチンを使用して実装され得る、



ドット積から生じる可能性がある。一実施例では、ベクトル演算および非線形関数の処理は、並列に実行されてもよい。

【0015】

【0022】勾配オプティマイザは、人工知能に使用される訓練システムにおける重要な構成要素の1つである。勾配オプティマイザは、次の反復の予測をグランドトゥールズに近づけることができるように、モデルのパラメータ（重み）上の勾配を適用してモデルを更新するように設計されている。本開示の特定の例は、2つの単精度（FP32）モーメントがモデルパラメータと共にバッファリングされるデュアルモーメント勾配オプティマイザをサポートするAI訓練フレームワークに関する。各更新ステップは、新しいモーメント値を計算し、それは、次に、新しい重みを生成するために使用される。新しいモーメント値と新しい重みの両方は、次の反復のために保存される。例として、デュアルモーメント適応型モーメント推定（Adam）アルゴリズムは、以下の式で表すことができる。

【0016】

$$\begin{aligned} v_t &:= \beta_1 \cdot v_t + (1 - \beta_1) \cdot dw \\ s_t &:= \beta_2 \cdot s_t + (1 - \beta_2) \cdot dw^2 \end{aligned}$$

【0017】

【数1】

$$w := w - \frac{\alpha v_t}{\sqrt{s_t} + \epsilon}$$

【0018】

ここで、 $v_t$ 、 $s_t$ は、2つのモーメントであり、 $w$ は、モデル重み、 $dw$ は、勾配、 $\alpha$ は、初期学習率であり、 $\epsilon$ はより良い数値安定性のために微小な値である。

【0023】有利には、デュアルモーメントオプティマイザの使用は、SGD（確率的勾配降下法）のようなゼロ／シングルモーメントオプティマイザよりも速く収束するニューラルネットワークモデルを得ることができる。しかし、デュアルモーメントオプティマイザを使用すると、モデル重み以外のモーメントをバッファリングするために、ゼロ／シングルモーメントオプティマイザが必要とするメモリの少なくとも2倍のメモリが必要になる場合がある。この傾向は、特に数百万から数十億のパラメータを含む大規模なモデルで深刻化する可能性がある。オプティマイザがメモリ制限のあるGPU／ASIC上で実行される場合、オプティマイザが要求するメモリ領域は、逆に最大バッチサイズを低減させる。一方、オプティマイザをホストCPUに移すことで、メモリサイズのボトルネックは解消され得る。しかし、メモリ帯域幅のボトルネックは、プロセッサが重み／勾配ペアごとにメモリから2つのFP32モーメントを余分にフェッチし、それらを記憶する必要があるので、オプティマイザの速度を低下させる可能性がある。

【0019】

【0024】さらに、Transformerによる双方向のエンコード表現（BERT）ーラージモデルのようなモデルが、ワーカごとに生成される勾配の1.36ギガバイト（GBs）有することを考えると、モデル全体の勾配のバッファリングは非常に大きな量のメモリを必要とする。このモデルの文脈では、4ワーカの訓練クラスターで勾配をバッファリングするために5.44GBのメモリが必要である。メモリの必要量は、ワーカの数に応じて直線的に増加する。さらに、アダプティブモーメント推定（Adam）のような、勾配ごとに2つの浮動小数点32ビット（FP32）モーメント項のバッファリングを必要とするアルゴリズムと併用する場合、各ワーカのメモリ要件は、BERTに必要なメモリの3倍となる可能性がある。

【0020】

【0025】図1は、一実施例によるデュアルモーメント勾配オプティマイザ（DMGO）150を含む人工知能訓練システム（AITS）100を示す図である。AITS100

0 は、メモリ 110（例えば、DRAM または別のタイプのメモリ）、メモリインターフェース 120（例えば、DDR インターフェースまたは別のタイプのインターフェース）、モーメントムフォーマットコンバータ（MFC）130、別のモーメントムフォーマットコンバータ（MFC）140、および DMGO 150 を含んでもよい。メモリ 110 は、バス 112 を介してメモリインターフェース 120 に結合されてもよい。あるいは、メモリ 110 は、チップオンチップパッケージ、システムインチップパッケージ、3D-メモリシステム、スタックダイシステムを介してメモリインターフェース 120 と一体化されてもよい。メモリインターフェース 120 は、モデル重み（W）がメモリバス 122 を介してメモリ 110 から DMGO 150 に転送され得るように、バス 122 を介して DMGO 150 に結合されてもよい。メモリインターフェース 120 は、バス 132 および 134 を介して MFC 130 に結合されてもよい。バス 132 は、第 1 のモーメントムに関連する記憶フォーマット（例えば、vt\_\_fp8 フォーマット）のモーメントム値をメモリ 110 から MFC 130 に転送するために用いられ得る。バス 134 は、第 2 のモーメントムに関連する記憶フォーマット（例えば、st\_\_fp8 フォーマット）のモーメントム値をメモリ 110 から MFC 130 に転送するために使用され得る。一例において、記憶フォーマットは、モーメントム値のための 8 ビット縮小精度フォーマットに対応してもよい。MFC 130 は、第 1 のモーメントム値および第 2 のモーメントム値の両方を処理して、それらのフォーマットを記憶フォーマットから DMGO 150 に関連付けられた訓練フォーマットに変更するように構成されてもよい。一例では、訓練フォーマットは、完全な 32 ビット単精度フォーマットに対応してもよい。MFC 130 は、バス 136 およびバス 138 をそれぞれ介して、第 1 および第 2 のモーメントム値を DMGO 150 に送信するようにさらに構成されてもよい。この例では、MFC 140 は、バス 142 および 144 を介して DMGO 150 に結合されてもよい。MFC 140 は、バス 146 および 148 を介してメモリインターフェース 110 にさらに結合されてもよい。MFC 140 は、訓練フォーマットから記憶フォーマットにモーメントム値を変換してもよい。

【0021】

【0026】さらに図 1 を参照すると、AITS 100 は、性能劣化を導入することなく、Adam のために低減された精度のモーメントムを使用する新しい方法の実装を可能にすることができる。一例として、デュアルモーメントム Adam アルゴリズムは、以下の式で表すことができる。

【0022】

$$\begin{aligned} v_t &:= \beta_1 \cdot v_t + (1 - \beta_1) \cdot dw \\ s_t &:= \beta_2 \cdot s_t + (1 - \beta_2) \cdot dw^2 \end{aligned}$$

【0023】

【数 2】

$$w := w - \frac{\alpha v_t}{\sqrt{s_t} + \epsilon}$$

【0024】

ここで、 $v_t$ 、 $s_t$  は、2 つのモーメントムであり、 $w$  は、モデル重み、 $dw$  は、勾配、 $\alpha$  は、初期学習率であり、 $\epsilon$  はより良い数値安定性のために微小な値である。一例では、モーメントム値の 1 つ（ $v_t$ ）は勾配値の期待値に対応し、モーメントム値の他の 1 つ（ $s_t$ ）は勾配値の分散の期待値に対応し得る。この例では、完全な 32 ビットの単精度のモーメントムを使うのではなく、8 ビットだけを使っている。符号に 1 ビット、モーメントム  $\{v_t, s_t\}$  のダイナミックレンジに 7 ビットである。この例では、 $v_t$ 、 $s_t$  ともに  $(1 - \beta)$  との乗算により 1.0 より小さく、ここで、 $\beta$  は 1.0 に近い値なので、8 ビット指数の最上位ビット（MSB）はゼロ定数となり、保存前に安全に破棄することができる。この例では、モーメントム値の記憶フォーマットは 8 ビットであるが、計算フォーマットはより高精度を実現するために、やはりより高精度である。図 1 は、ある方法

で配置された、ある数の構成要素を含む A I T S 1 0 0 を示しているが、A I T S 1 0 0 は、異なる方法で配置された追加のまたはより少ない構成要素を含んでもよい。一例として、M F C 1 3 0 および M F C 1 4 0 は、チップオンチップパッケージ、システムインチップパッケージ、3 D - メモリシステム、スタックダイシステムを介して D M G O 1 5 0 と統合されてもよい。あるいは、M F C 1 3 0 および M F C 1 4 0 は、（例えば、システムオンチップ（S o C）として）同一チップ上で D M G O 1 5 0 と統合されてもよい。別の例では、メモリ 1 1 0、メモリインターフェース 1 2 0、M F C 1 3 0、M F C 1 4 0、および D M G O 1 5 0 の各々は、チップオンチップパッケージ、システムインチップパッケージ、3 D - メモリシステム、スタックダイシステムを介して統合されてもよい。

【0025】

【0027】図 2 は、一実施例によるデュアルモーメンタム勾配オプティマイザ（D M G O）2 0 0 を示す図である。D M G O 2 0 0 は、モーメンタム伸長器 2 0 2 および別のモーメンタム伸長器 2 0 4 を含んでもよい。D M G O 2 0 0 は、さらに、モーメンタム圧縮器 2 6 2 と別のモーメンタム圧縮器 2 6 4 を含んでもよい。D M G O 2 0 0 は、デュアルモーメンタム・アダプティブモーメント推定（A d a m）アルゴリズムの実装に関連する様々な動作を実行するように構成されたロジックをさらに含んでもよい。この例では、モーメンタム伸長器 2 0 2 およびモーメンタム伸長器 2 0 4 は、図 1 の M F C 1 3 0 に関して前述したのと同様の機能を提供するために使用され得る。同様に、モーメンタム圧縮器 2 6 2 およびモーメンタム圧縮器 2 6 4 は、図 1 の M F C 1 4 0 に関して前述したのと同様の機能を実装するために使用され得る。第 1 のモーメンタム（例えば、v t \_ f p 8）に関連するモーメンタム値は、メモリから取得され、モーメンタム伸長器 2 0 2 に提供されてもよい。第 2 のモーメンタム（例えば、s t \_ f p 8）に関連するモーメンタム値は、メモリから取得され、モーメンタム伸長器 2 0 4 に提供されてもよい。一例では、モーメンタム値は、記憶フォーマット（例えば、8 ビット縮小精度フォーマット）を有してもよい。モーメンタム伸長器 2 0 2 は、第 1 のモーメンタム値を処理して、そのフォーマットを記憶フォーマットから D M G O 2 0 0 に関連する訓練フォーマットに変更するように構成されてもよい。モーメンタム伸長器 2 0 4 は、第 2 のモーメンタム値を処理して、そのフォーマットを記憶フォーマットから D M G O 2 0 0 に関連付けられた訓練フォーマットに変更するように構成されてもよい。

【0026】

【0028】図 2 を引き続き参照すると、D M G O 2 0 0 の一部として含まれるロジックは、加算、乗算、および融合型乗算加算演算などの演算を実行するように構成されたロジックを含んでもよい。ロジックは、デュアルモーメンタム最適化アルゴリズム（例えば、デュアルモーメンタム・アダプティブモーメント推定（A d a m）アルゴリズム）を実装するように構成されてもよい。一例として、新しいモーメンタム値（v t \_ n e w, s t \_ n e w）の計算に関するデュアルモーメンタム A d a m アルゴリズムに対応する計算は、以下の式で表すことができる。

【0027】

$$\begin{aligned} v_{t\_new} &:= \beta_1 \cdot v_{t\_fp32} + (1 - \beta_1) \cdot dw \\ s_{t\_new} &:= \beta_2 \cdot s_{t\_fp32} + (1 - \beta_2) \cdot dw^2 \end{aligned}$$

【0028】

【数 3】

$$w := w - \frac{av_t}{\sqrt{s_t} + \epsilon}$$

【0029】

ここで、v t \_ n e w, s t \_ n e w は、新しいモーメンタム値、w はモデル重み、d w は勾配、a は初期学習率、ε はより良い数値安定性のために使用される微小な値である。したがって、この例では、モーメンタム伸長器 2 0 2 および 2 0 4 は、それぞれのモーメ

ンタム値 ( $v_{t\_fp8}$ ,  $s_{t\_fp8}$ ) の記憶フォーマット (例えば、8ビット圧縮フォーマット) を、DMGO200に関連する訓練フォーマット (例えば、32ビット全精度フォーマット) へ変換する。一例として、勾配最適化が実行される前に、モーメンタム伸長器 (例えば、モーメンタム伸長器202または204) は、符号ビットと指数の最上位ビット (MSB) との間に1つの「0」ビットを挿入し、指数の最下位ビット (LSB) の直後に23の「0」ビットをパディングすることによって、圧縮された8ビットモーメンタム値を32ビット単精度に戻すように変換するであろう。パディングされるビットは「0」ビットである必要はない。一例では、少なくとも少数のビットが非ゼロビットであってもよい。一例として、2つまたは3つのLSBは非ゼロビットであってよい。

【0030】

【0029】さらに図2を参照すると、乗算器210および214ならびに加算器212を含むDMGO200に関連するロジックは、新しいモーメンタム値 ( $v_{t\_new}$ ) のうちの1つを生成することに関連する計算を実行してもよい。同様に、乗算器216、218、および加算器222を含むDMGO200に関連するロジックは、他の新しいモーメンタム値 ( $s_{t\_new}$ ) を生成することに関連する計算を実行してもよい。さらに、この例では、新しいモーメンタム値 ( $v_{t\_new}$ ,  $s_{t\_new}$ ) をメモリに記憶する前に、モーメンタム圧縮器262および264は、新しいモーメンタム値のフォーマットを訓練フォーマットから記憶フォーマットに変換する。新しいモーメンタム値 ( $v_{tc\_new}$ ,  $s_{tc\_new}$ ) は、図2において8ビット記憶フォーマット値として示されている。この例では、モーメンタム圧縮器262および264の各々は、モーメンタム値に関連する指数の最上位ビット (MSB) を除去し、23の仮数ビットのすべてを切り捨てて、32ビット全精度のモーメンタム値 ( $v_{t\_new}$ ,  $s_{t\_new}$ ) を8ビット圧縮モーメンタム値 ( $v_{tc\_new}$ ,  $s_{tc\_new}$ ) に変換するように構成してもよい。これらはメモリに記憶されることになる。また、23ビット未満の仮数ビットを切り捨ててもよい。有利には、モーメンタム値の圧縮により、モーメンタム値を記憶するために必要なバッファサイズを4倍縮小することができる。さらに、最下位ビット (LSB) 切り捨ての使用は、誤差が互いにキャンセルされ得るように同じ丸め誤差方向を導入することによって、訓練性能の損失なしに精度を低下させることができる。

【0031】

【0030】引き続き図2を参照すると、DMGO200は、新しい重み値 (例えば、 $w_{new}$ ) を生成するように構成されたロジックをさらに含んでもよい。この例では、このロジックは、乗算器232および234、加算器242、逆数平方根 (RSQRT) 246、および融合型乗算器 (FMA) 250を含んでもよい。図2は、ある方法で配置されたある数の構成要素を含むDMGO200を示すが、DMGO200は、異なる方法で配置された追加のまたはより少ない構成要素を含んでもよい。

【0032】

【0031】モーメンタム圧縮器によってモーメンタムが圧縮されるとき、その精密ビットは、新しいモーメンタム値 (例えば、それぞれ、 $v_{tc\_new}$ ,  $s_{tc\_new}$ ) に関して  $\delta_1$  および  $\delta_2$  の相対誤差を導入するように切り捨てられる。これを考慮すると、勾配最適化方程式は次のように書き直すことができる。

【0033】

【数4】

$$w = w - \frac{\alpha \cdot vt(1 - \delta_1)}{\sqrt{st(1 - \delta_2)}} \approx w - \frac{\alpha \cdot vt}{\sqrt{st}} \cdot \frac{1 - \delta}{\sqrt{1 - \delta}} = w - \frac{\alpha \cdot vt}{\sqrt{st}} \sqrt{1 - \delta}$$

【0034】

32ビット単精度の場合、 $|\delta_1| \leq 2^{m-24}$ ,  $|\delta_2| \leq 2^{m-24}$  となり、 $m$  は切り捨てた精度ビット数である。この例では、 $m$  が大きすぎなければ、 $1 - \delta_1 = 1 - \delta_2$ 、 $1 - \delta$  を安全に近似できる。この誤差限界は、切り捨てられた  $m$  ビットがすべて「1」

10

20

30

40

50

である場合にのみ達成され、このケースは実際のアプリケーションではほとんど起こり得ない。それでも、この極端なケースが起こったとしても、瞬間的なピーク誤差は全体の性能に悪影響を及ぼし得ない。両モーメントを切り捨てた後の更新された重みの相対誤差は、

【0035】

【数5】

$$\tau = 1 - \sqrt{1 - \delta}$$

【0036】

と表すことができる。どのような  $\delta \in [0, 1]$  でも、 $\delta$  の値が小さいと

【0037】

【数6】

$$\sqrt{1 - \delta}$$

【0038】

は  $1 - \delta$  よりも 1 に近くなる。このことは、両モーメントの精度を下げても、更新されたパラメータの精度は線形には下がらないことを示している。この「誤差に対する平方根」の効果は、切り捨てるによる「精度損失」の一部を補償する。

【0039】

[00032]図3は、圧縮されたモーメント値が仮数1ビットの切り捨てから仮数23ビットの切り捨てまで掃引されている状態での重み更新誤差の誤差限界曲線 (EB (M)) を示すグラフ300である。グラフ300に示すように、この例では、切り捨てられた仮数ビットの数が16より少ない場合、完全な単精度 (FP32) モーメント値と比較して、誤差限界値はゼロに近い。この例では、切り捨てられた仮数ビットの数が17を超えると、誤差限界は急速に増加し、仮数ビットのすべてが切り捨てられたときに、最大誤差限界は0.13である。

【0040】

[00033]図4は、精度を下げたモーメント値が訓練の精度に及ぼす影響を示すシミュレーション収束曲線410、420、および420を有するグラフ400を示す。一例として、シミュレーション収束曲線410はモーメント値の1ビット仮数に対応し、シミュレーション収束曲線420はモーメント値の4ビット仮数に対応し、シミュレーション収束曲線430はモーメント値の24ビット仮数に対応する。シミュレーション収束曲線から明らかなように、モーメント値の仮数ビットの切り捨ては、訓練の精度に関して性能を低下させない。しかし、精度を下げたモーメント値はメモリフットプリントの少なくとも50%を節約するので、メモリを読み出すためのサイクルの半分だけが必要となる場合がある。

【0041】

[00034]本開示の一部として説明された技術は、ハードウェアおよび／またはソフトウェアの組み合わせを使用して実装されてもよい。特定の例では、図2のDGM0200は、図5に示されるような訓練システム500の一部として展開されてもよい。この例では、DGM0200は、一例にしたがって、モデルを訓練するための、ハードウェア支援勾配オプティマイザ (デュアルモーメント) 550 (HAGOと呼ばれる) の一部として含まれてもよい。勾配最適化システム500は、HAGO550に高速バスブリッジ520を介して結合された複数のワーカ (例えば、ワーカ\_\_1 502、ワーカ\_\_2 504、およびワーカ\_\_N 506) を含んでもよい。ワーカ (例えば、ワーカ\_\_1 502、ワーカ\_\_2 504、およびワーカ\_\_N 506) の各々は、任意の数のグラフィックス処理ユニット (GPU)、コンピュータ処理ユニット (CPU)、メモリ (例えば、SRAMまたは他のタイプのメモリ)、フィールドプログラマブルゲートアレイ (FPGA)、特定用途向け集積回路 (ASIC)、消去可能および／または複合プログラマブル論理デバイ

10

20

30

40

50

ス（PLD）、プログラマブルアレイロジック（PAL）デバイス、または汎用アレイロジック（GAL）デバイスを使用して実装されてもよい。各ワーカは、勾配最適化システム500を使用して訓練されているニューラルネットワークモデルの全体コピーを担持してもよい。各ワーカは、ニューラルネットワークモデルのコピー全体を担持してもよいが、各ワーカは、モデルに関連付けられた訓練データ（例えば、分類用の画像または他のデータサンプル）のサブセットに対してのみ動作してもよい。各ワーカは、データに対してフォワードパスを実行して誤差を計算し、その後、バックワードパスを実行して各パラメータ（例えば、各重み）に対する勾配を計算してもよい。勾配最適化システム500は、高速バスブリッジ520にバス528を介して結合されたホスト560をさらに含んでもよい。

10

【0042】

[00035]図5を引き続き参照すると、一例では、高速バスブリッジ520は、ピーシーアイエクスプレス（PCIe）ブリッジとして実装されてもよい。ワーカ\_\_1 502は、PCIeリンク510を介して高速バスブリッジ520に接続されてもよい。ワーカ\_\_2 504は、PCIeリンク512を介して高速バスブリッジ520に接続されてもよい。ワーカ\_\_N 506は、PCIeリンク514を介して高速バスブリッジ520に接続されてもよい。HAGO550は、バスリンク522および524を介して高速バスブリッジ520に連結されてもよい。また、他のタイプのバスブリッジおよびバスリンクが使用され得る。さらに、バスブリッジの代わりに、スイッチングファブリックが使用され得る。各ワーカは、訓練データのサブセット（例えば、ミニバッチ）上で勾配を計算し、高速バスブリッジ520を介して更新された勾配をHAGO550に提供してもよい。HAGO550は、各ワーカ（例えば、ワーカ\_\_1 502、ワーカ\_\_2 504、およびワーカ\_\_N 506）から収集した勾配に対して、縮小および最適化を実行してもよい。HAGO550は、データバーストで動作してもよく、一例では、各バーストは、一定数の受信勾配で構成されてもよい。現在の勾配バーストは、前の勾配バーストがその対応する重みと共に最適化される間、バッファリングされてもよい。図5は、ある方法で配置された、ある数の構成要素を含む勾配最適化システム500を示すが、勾配最適化システム500は、異なる方法で配置された追加のまたはより少ない構成要素を含んでもよい。

20

【0043】

[00036]図6は、一実施例による、デュアルモーメンタム勾配オプティマイザを含むHAGO600のブロック図である。一実施例では、図5のHAGO550は、図6のHAGO600として実装されてもよい。HAGO600は、高速バスを介してワーカから更新された勾配を受信してもよい。図6の例では、HAGO600に4つのワーカが結合されていることを想定している。ワーカの1つからの更新された勾配（DW1）は、先入れ先出しブロック（例えば、GFIFO\_\_1 602）を介して受信されてもよい。別のワーカからの更新された勾配（DW2）は、別の先入れ先出しブロック（例えば、GFIFO\_\_2 604）を介して受信されてもよい。別のワーカからの更新された勾配（DW3）は、別の先入れ先出しブロック（例えば、GFIFO\_\_3 606）を介して受信されてもよい。別のワーカからの更新された勾配（DW4）は、別の先入れ先出しブロック（例えば、GFIFO\_\_4 608）を介して受信されてもよい。FIFOは、単一のマルチエントリーFIFOまたは複数のシングルエントリーFIFOとして実装されてもよい。

30

40

【0044】

[00037]図6を引き続き参照すると、HAGO600は、浮動小数点平均化ブロック（例えば、FP-AVGブロック620）をさらに含んでもよい。FP-AVGブロック620は、先入れ先出しブロックからバッファリングされた更新された勾配を受信してもよい。一例として、FP-AVGブロック620は、GFIFO\_\_1 602からバッファリングされた更新された勾配（BDW1）を受信してもよい。FP-AVGブロック620はまた、GFIFO\_\_2 604からバッファリングされた更新された勾配（BDW2）を受信してもよい。FP-AVGブロック620はまた、GFIFO\_\_3 606からバッファリングされた更新された勾配（BDW3）を受信してもよい。FP-AVGブロ

50

ック620はまた、GFIFO\_\_4 608からバッファリングされた更新された勾配（BDW4）を受信してもよい。FP-AVGブロック620は、受信した更新された勾配に対して縮小を実行してもよい。この例では、縮小操作は、受信した勾配を平均化し、出力としてDW\_\_AVGを生成することを含んでもよい。縮小操作は、平均化に限定されない。加算やスケーリングされた加算など、他の演算を使用して縮小を実行してもよい。FP-AVGブロック620からの出力は、図1および図2に関して先に説明したものなどの勾配最適化アルゴリズムを実装するために浮動小数点演算を行う勾配最適化データパス（GODP）ブロック650に提供されてもよい。一例では、FP-AVGブロック620およびGODPブロック650は、FPGAの一部として実装されてもよい。

【0045】

[00038]さらに図6を参照すると、重み（W）およびオプションのモーメンタム（M）のマスターコピーは、メモリ660に記憶されてもよい。一例では、メモリ660は、ダブルデータレート（DDR）ダイナミック・ランダム・アクセス・メモリ（DRAM）を使用して実装されてもよい。メモリ660は、メモリインターフェース662を介して、先入れ先出しブロック（例えば、PFIFOブロック670）に結合されてもよい。メモリインターフェース662は、DDRインターフェースであってもよい。図6に示すように、モデルの重み（W）は、制御有限状態マシン（FSM）（例えば、制御FSM640）の制御下で、PFIFO670にロードされてもよい。さらに、メモリ660は、メモリインターフェース662を介してモーメンタムフォーマット変換器（MFC）684に結合されてもよい。MFC684は、図1のMFC130に関して先に説明したのと同様の機能性を提供してもよい。MFC684の出力は、PFIFO670（または別のFIFO）に結合されてもよい。

【0046】

[00039]一例では、同時に、制御FSM640は、（1）各ワーカがそれぞれのローカルに計算された勾配（例えば、DW1、DW2、DW3、およびDW4）をそれぞれのGFIFO（例えば、GFIFO\_\_1 602、GFIFO\_\_2 604、GFIFO\_\_3 606、およびGFIFO\_\_4 608）に送信すること、および（2）メモリインターフェース662が、メモリ660からPFIFO670へ、オプションのモーメンタムとともに、同じ数のパラメータ（例えば、重み（W））をフェッチすること、の両方を可能にする。したがって、この例では、制御FSM640は、単一のバーストサイクル中にGFIFOに注入される勾配の数と同じ数の重みをPFIFO670にフェッチするようにメモリインターフェース662を制御する。これらのFIFOのそれぞれがバーストサイズに達すると、これらのそれぞれの読み取りポートが制御FSM640によって起動され、それぞれのFIFOの内容を単一サイクルのレイテンシで出力する。この例では、制御FSM640は、これらのバッファからのデータが同じレイテンシでフラッシュされるように、PFIFO670およびGFIFOからの重みの読み出しを可能にする。このように、データの読み出しは完全に調整される。

【0047】

[00040]図6を引き続き参照すると、この実装態様は、それぞれのFIFOからの4つの勾配出力（例えば、BDW1、BDW2、BDW3、およびBDW4）が同じ位置合わせインデックスを有することを確実にする。FP-AVGブロック620は、次に、これらの勾配（例えば、BDW1、BDW2、BDW3、BDW4）の平均を計算して、縮小勾配（例えば、DW\_\_AVG）を生成し、これは次に、GODPブロック650によってバッファされた重みおよびモーメンタムとともに処理される。この例では、GODPブロック650の出力は、新しい重み（NW）および新しいモーメンタム（NM）の2つの部分を含んでもよく、これらは両方ともメモリインターフェース662を介してメモリ660に書き戻される。新しいモーメンタム値は、図1のMFC140に関して前述したのと同様の機能を実装することができるMFC682に提供されてもよい。したがって、この態様では、圧縮されたモーメンタム値は、より少ない記憶空間を必要とする記憶フォーマットで記憶されてもよい。

## 【 0 0 4 8 】

【00041】さらに、同時に新しい重み（NW）が、新しい重みをワーカに送り返す送信 F I F O（T X F I F O）に注入される。この実施例では、制御 F S M 6 4 0 は、図 5 の高速バスブリッジ 5 2 0 を介したワーカへの重みの流れを制御してもよい。一例として、制御／状態レジスタ 6 3 0 は、様々な動作に関連するタイミングを指定するために使用され得る。さらに、制御／状態レジスタ 6 3 0 は、勾配オプティマイザのハイパーパラメータ（例えば、学習率）を構成するために使用され得る。

## 【 0 0 4 9 】

【00042】それぞれの G F I F O からの読み出しおよび重みの更新の間、勾配の次のバーストは、それぞれの G F I F O に書き込まれる。同様に、次の重みのセットおよびモーメンタムも、P F I F O 6 7 0 からの読み出しおよび重み更新の間、メモリ 6 6 0 から P F I F O 6 7 0 に（例えば、M F C 6 8 4 を介して）フェッチされる。この例では、メモリ 6 6 0 は、読み出しと書き込みの動作の両方によって共有される単一のデータポートだけをサポートするので、P F I F O 6 7 0 のデータ幅は、各 G F I F O のデータ幅の 2 倍であるように構成される。このようにして、この例では、最初のものとは最後のものを除くすべてのデータバーストは、勾配の受信と新しい重みの送信との間にオーバーラップを有することになる。

## 【 0 0 5 0 】

【00043】さらに図 6 を参照すると、H A G O 6 0 0 の動作の特定の態様は、制御 F S M 6 4 0 とともに制御／状態レジスタを使用して制御され得る。以下の表 1 は、制御／状態レジスタ 6 3 0 の例示的な内容を示す。

## 【 0 0 5 1 】

【表 1】

| レジスタ名           | レジスタサイズ | レジスタ説明                                                                     |
|-----------------|---------|----------------------------------------------------------------------------|
| ctrl_opt_mode   | [1:0]   | 最適化アルゴリズムタイプを選択：<br>デュアルモーメンタムアダムまたは他のいくつかの<br>デュアルモーメンタム最適化アルゴリズム         |
| ctrl_burst_size | [3:0]   | バーストサイズを選択： 4'd8=>256, 4'd9=>512,<br>4'd10=>1024, 4'd11=>2048, 4'd12=>4096 |
| ctrl_param_num  | [31:0]  | モデルのパラメータのサイズを設定                                                           |
| ctrl_init_lr    | [31:0]  | オプティマイザの初期学習率（F P 3 2 フォーマット）<br>を設定                                       |
| ctrl_beta1      | [31:0]  | オプティマイザのモーメンタムの係数を設定<br>（F P 3 2 フォーマット）                                   |
| ctrl_beta2      | [31:0]  | オプティマイザの第 2 のモーメンタムの係数を設定<br>（F P 3 2 フォーマット）                              |
| ctrl_is_fp16    | [0:0]   | 入力勾配のフォーマットが FP16 フォーマットの場合に設定                                             |

表 1

## 【 0 0 5 2 】

【00044】表 1 に示すように、c t r l \_ o p t モードレジスタは、任意のデュアルモーメンタム最適化アルゴリズムであってよい最適化アルゴリズムタイプを指定することを可能にする。c t r l \_ バースト \_ サイズレジスタは、各バーストサイクルに関連するバー



ストサイズを選択することができる。この例では、バーストサイズ（例えば、勾配の数で指定される）は、256勾配、512勾配、1024勾配、2048勾配、または4096勾配であり得る。ctrl\_param\_numレジスタは、モデルのパラメータのサイズを指定することができる。ctrl\_init\_lrレジスタは、モデルの初期学習率を指定することができる。一実施例では、初期学習率は、モデルのステップサイズに関連し得る。先に説明したように、一実施例では、最適化アルゴリズムは、以下の式で表すことができるデュアルモーメンタムアダムアルゴリズムであってもよい。

【0053】

$$v_t := \beta_1 \cdot v_t + (1 - \beta_1) \cdot dw$$

$$s_t := \beta_2 \cdot s_t + (1 - \beta_2) \cdot dw^2$$

【0054】

【数7】

$$w := w - \frac{\alpha v_t}{\sqrt{s_t} + \epsilon}$$

【0055】

ここで、 $v_t$ 、 $s_t$ は、2つのモーメンタムであり、 $w$ は、モデル重み、 $dw$ は、勾配、 $a$ は、初期学習率であり、 $\epsilon$ はより良い数値安定性のために微小な値である。ctrl\_beta1レジスタは、重みが更新される速度に関連する加速度を制御することができるモーメンタムの係数を指定することができる。ctrl\_beta2レジスタは、重みが更新される速度に関連する加速度をさらに制御することができる第2のモーメンタムの係数を指定することを可能にする。ctrl\_is\_fp16レジスタは、入力勾配のフォーマットがFP16フォーマットの場合に設定される。表1は、制御／状態レジスタ630の一部として含まれる特定のレジスタのセットを示しているが、追加のまたはより少ないレジスタがHAGO600と共に使用され得る。さらに、制御／状態レジスタ630に含まれる制御情報の少なくとも一部または全部は、HAGO600を用いた勾配最適化に関連する命令に関連するフィールドまたはオペランドを介して指定されてもよい。図6は、ある方法で配置されたある数の構成要素を含むHAGO600を示すが、HAGO600は、異なる方法で配置された追加のまたはより少ない構成要素を含んでもよい。一例として、5つ以上のワーカがHAGO600に結合されてもよい。

【0056】

【00045】図7は、一実施例による、図6のHAGO600に関連するタイミング図700を示す。タイミング図700は、HAGO600の動作をバーストの観点で説明する。タイミング図700は、 $K+1$ バーストサイクルを示し、ここで、 $K$ は正の整数である。最初のバーストサイクル（バースト\_\_1）および最後のバーストサイクル（バースト\_\_ $K+1$ ）は、他のバーストサイクル：バースト\_\_2サイクルからバースト\_\_ $K$ サイクルまで、よりも少ない重複を含む。バースト\_\_1サイクルは、ワーカからの勾配（例えば、DW1）の送信開始からFIFO（例えば、図6のGFIFO\_\_1 602）への勾配の注入までに関わるタイミング（GW）をFIFOのバーストサイズまで含んでいる。複数のワーカ（例えば、図5のワーカ\_\_1 502、ワーカ\_\_2 504、ワーカ\_\_N 506）は、バースト\_\_1サイクルの間に、FIFOのバーストサイズまでFIFOに勾配を注入することができる。バースト\_\_1サイクルはさらに、図6のPFIFO670への重みの注入までの、メモリ（例えば、図6のメモリ660）に記憶されたパラメータ（例えば、モデルの重み）の転送の開始に関与するタイミング（PW）を含む。図7に示すように、重みの転送は、メモリ（例えば、図6のメモリ660）の読み出しレイテンシ（LDDR）に影響される。したがって、バースト\_\_1サイクルにおいて、HAGO600は、FIFOへの勾配の注入と、メモリからの重みのフェッチという2つの動作を重複して行う。

【0057】

【00046】図7を引き続き参照すると、バースト\_\_2サイクルは、FIFO（例えば、図

10

20

30

40

50

6のG F I F O\_\_1 6 0 2)への勾配の注入までのワーカからの勾配の次のセット(例えば、D W 1)の送信開始に関わるタイミング(G W)をF I F Oのバーストサイズまで含んでいる。複数のワーカ(例えば、図5のワーカ\_\_1 5 0 2、ワーカ\_\_2 5 0 4、ワーカ\_\_N 5 0 6)は、バースト\_\_2サイクル中にF I F Oのバーストサイズまで勾配をF I F Oに注入することができる。勾配の次のセットがF I F Oに注入されている間、バースト\_\_2サイクルは、F P-A V Gブロック6 2 0による勾配の平均の計算、およびG O D Pブロック6 5 0によるバッファされたモーメンタムおよびバッファされた重み(B M, B W)と共に縮小勾配の処理、を含む計算サイクルをも含む。この操作の一部として関与するタイミングは、図7においてG P Rとしてラベル付けされている。G O D Pブロック6 5 0から出力された新しいモーメンタムおよび新しい重み(N M, N W)をメモリ6 6 0に書き戻すことに関与するタイミングは、N Pとしてラベル付けされる。バースト\_\_2サイクルは、さらに、T X F I F O 6 8 0を介したワーカへの新しい重みの送信に関与するタイミングを重複して含む。図7のタイミング図7 0 0に示すように、この動作は、G O D Pブロック6 5 0の処理レイテンシ(L D P)によって影響を受ける。バースト\_\_2サイクルはさらに、メモリ(例えば、図6のメモリ6 6 0)に記憶されたパラメータ(例えば、モデルの重み)の転送の開始から、図6のP F I F O 6 7 0への重みの注入までに関わるタイミング(P W)を含む。図7に示すように、重みの転送は、メモリ(例えば、図6のメモリ6 6 0)の読み出しレイテンシ(L D D R)に影響される。したがって、バースト\_\_2サイクルにおいて、H A G O 6 0 0は、勾配最適化の一部として、(1) F I F O(例えば、G F I F Oのいずれか)への勾配の注入、(2) G F I F OおよびP F I F Oからの勾配および重みの読み出し、(3)メモリへの新しい重みおよび新しいモーメンタムの書き戻し、(4)ワーカに対する重みの転送、の4つの動作を少なくとも部分的にオーバーラップさせている。

#### 【0058】

[00047]さらに図7を参照すると、制御F S M 6 4 0は、連続するバーストサイクルがH A G O 6 0 0を通る勾配のストリーミングをもたらすように、ストリーミング様式でH A G O 6 0 0を動作させ続ける。したがって、バースト\_\_3サイクルは、勾配の追加のストリーミングと、勾配最適化の一部として、(1) F I F O(例えば、G F I F Oのいずれか)への勾配の注入、(2) G F I F OおよびP F I F Oからの勾配および重みの読み出し、(3)メモリへの新しい重みおよび新しいモーメンタムの書き戻し、(4)ワーカに対する新しい重みの転送、の4つの動作の少なくとも部分的なオーバーラップを含む複数の動作のオーバーラップ、を含む。同様に、バースト\_\_Kサイクルは、勾配最適化の一部として、(1) F I F O(例えば、G F I F Oのいずれか)への勾配の注入、(2) G F I F OおよびP F I F Oからの勾配および重みの読み出し、(3)メモリへの新しい重みおよび新しいモーメンタムの書き戻し、(4)ワーカに対する新しい重みの転送、の4つの動作の少なくとも部分的なオーバーラップを含む。バースト\_\_K+1サイクルは最後のバーストサイクルに対応する。したがって、バースト\_\_K+1サイクルは、他のバーストサイクル：バースト\_\_2サイクルからバースト\_\_Kサイクルまで、に比べてオーバーラップが少ない。バースト\_\_K+1サイクルの一部として、(1) G F I F OおよびP F I F Oからの勾配と重みの読み出し、(2)新しい重みと新しいモーメンタムのメモリへの書き戻し、(3)新しい重みのワーカへの送信、の3動作がオーバーラップする。

#### 【0059】

[00048]図7を引き続き参照すると、各単一バーストサイクル中に、H A G O 7 0 0は勾配バーストで動作してもよく、各勾配バーストは、固定数の勾配を含んでもよい。単一のバーストサイクル中に動作される勾配の数は、図6に関して先に説明したように、制御／状態レジスタ6 3 0を介して構成され得るバーストサイズに対応する。

#### 【0060】

[00049]図8は、別の実施例による、デュアルモーメンタム勾配オプティマイザ(D M G O)のソフトウェアバージョンを含む、勾配最適化システム8 0 0を示す図である。この例では、D M G O(例えば、図2のD M G O 2 0 0)に対応するロジックの実装は、プ

ロセッサによって実行されるとき、メモリに記憶された命令を使用して実装されてもよい。勾配最適化システム 800 は、プロセッサ 802、I/O コンポーネント 804、メモリ 806、プレゼンテーションコンポーネント 808、センサ 810、データベース 812、ネットワーキングインターフェース 814、および I/O ポート 816 を含み、これらはバス 820 を介して相互接続されてもよい。プロセッサ 802 は、メモリ 806 に記憶された命令を実行し得る。I/O コンポーネント 804 は、キーボード、マウス、音声認識プロセッサ、またはタッチスクリーンなどのコンポーネントを含み得る。メモリ 806 は、不揮発性ストレージまたは揮発性ストレージ（例えば、フラッシュメモリ、DRAM、SRAM、または他のタイプのメモリ）の任意の組み合わせであってよい。DMGO に関連する方法に対応する命令は、メモリ 806 または別のメモリに記憶されてもよい。これらの命令は、プロセッサ 802、または他のプロセッサ（例えば、他の CPU または GPU）によって実行されるとき、これらの方法に関連する機能を提供し得る。一例として、右シフト/左シフト命令およびパッキング/アンパッキング命令を含む命令は、モーメントム値を圧縮または伸長するために使用され得る。以下の表 2 は、プロセッサ 802 によって実行されると、DMGO（例えば、図 2 の DMGO 200）に対応する機能性を実行し得る、命令の例示的なセットを示す。

【0061】

【表 2】

| 命令                     | コメント                                                                                     |
|------------------------|------------------------------------------------------------------------------------------|
| LD R0, R1, #オフセット      | addr [R1 + オフセット] をレジスタ R0 へロード                                                          |
| UNPACK32_16 R3, R4, R0 | 2つの16ビットモーメントムとともにR0をそれらのFP32バージョンへアンパック                                                 |
|                        | バージョン (version)                                                                          |
| MUL R3, R3, R5         | 最適化を実行                                                                                   |
| ...                    | ...                                                                                      |
| ADD R3, R4, R5         | 新しいモーメントムがR3およびR6に記憶される                                                                  |
| ADD R6, R4, R7         |                                                                                          |
| PACK16_32 R0, R3, R6   | 2つの浮動小数点32 (FP32) モーメントム値を単精度32ビットワードに切り捨ておよび結合することによって、2つのブロック浮動小数点16フォーマット (BF16) にパック |
| ST R0, R1, #オフセット      | 更新されたモーメントム値を記憶し、メモリに書き戻す                                                                |

表 2

【0062】

[00050] さらに図 8 を参照すると、プレゼンテーション構成要素 808 は、ディスプレイ、ホログラフィックデバイス、または他のプレゼンテーションデバイスを含んでもよい。ディスプレイは、LCD、LED、または他のタイプのディスプレイのような、任意のタイプのディスプレイであってもよい。データベース 812 は、本明細書に記載される方法の実行のために必要に応じてデータを記憶するために使用され得る。データベース 812 は、分散型データベースの集合体として、または単一データベースとして実装されてもよい。ネットワークインターフェース 814 は、イーサネット、セルラー無線、ブルートゥース（登録商標）無線、UWB 無線、または他のタイプの無線若しくは有線通信インターフェースなどの通信インターフェースを含んでもよい。I/O ポート 816 は、イーサネットポート、光ファイバーポート、無線ポート、または他の通信ポートを含んでもよい。図 8 は、ある方法で配置および結合されたある数の構成要素を含むものとして勾配最適化システム 800 を示しているが、異なる方法で配置および結合された、より少ないまた

は追加の構成要素を含むことができる。さらに、システム 800 に関連する機能性は、必要に応じて分散されてもよい。

【0063】

[00051]図 9 は、一実施例による勾配最適化のための方法のフロー図 900 を示す。この方法は、図 1 の AITS100、図 2 の DMGO200、または図 8 の勾配最適化システム 800 を使用して実行されてもよい。したがって、DMGO200 が HAGO600 の一部として実装される場合の例として、制御 FSM640 から受信した制御信号に基づいて、この方法に対応するステップが実行されてもよい。さらに、図 6 には示されていないが、この方法の一部として実行される動作に関連するタイミングを同期させるために、クロック信号が使用され得る。あるいは、メモリ 806 に記憶された命令は、プロセッサ 802 によって実行されるとき、この方法に関連するステップを実行するために使用され得る。さらに、DMGO200 および勾配最適化システム 800 の修正された組み合わせも使用され得る。したがって、一例として、モーメンタム圧縮および伸長ステップは、メモリに記憶された命令を使用して実行されてもよいが、勾配最適化は、DMGO200 に関連するハードウェアロジックを使用して実行されてもよい。ステップ 910 は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第 1 のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第 2 のセットをメモリから取り出すステップと、を含んでもよい。先に説明したように、記憶フォーマットは、縮小精度フォーマットに対応してもよい。

【0064】

[00052]ステップ 920 は、選択された記憶フォーマットを有するモーメンタム値の第 1 のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 3 のセットに変換するステップと、選択された記憶フォーマットを有するモーメンタム値の第 2 のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第 4 のセットに変換するステップと、を含んでもよい。先に説明したように、このステップは、記憶フォーマットにおけるモーメンタム値に余分なゼロビットをパディングすることによってモーメンタム値を伸長することを含んでもよい。追加される余分なゼロビットは、モーメンタム値の圧縮の一部として切り捨てられたビットの数に依存してもよい。

【0065】

[00053]ステップ 930 は、勾配オプティマイザを用いて、モーメンタム値の第 3 のセットおよびモーメンタム値の第 4 のセットを使用して勾配最適化を実行するステップを含み得る。勾配最適化に関連する計算の後、モーメンタム値は、記憶フォーマットに変換されてもよい。先に説明したように、このステップは、単精度フォーマットのモーメンタム値の符号ビットと 7 つの最上位ビット (MSB) のみを記憶するステップを含んでもよい。図 9 は、ある順序で実行されるある数のステップを説明しているが、異なる順序の追加のステップまたはより少ないステップが実行されてもよい。

【0066】

[00054]図 10 は、一実施例による勾配最適化のための方法のフロー図 1000 を示す。この方法は、図 1 の AITS100、図 2 の DMGO200、または図 8 の勾配最適化システム 800 を使用して実行されてもよい。したがって、DMGO200 が HAGO600 の一部として実装される場合の例として、制御 FSM640 から受信した制御信号に基づいて、この方法に対応するステップが実行されてもよい。さらに、図 6 には示されていないが、この方法の一部として実行される動作に関連するタイミングを同期させるために、クロック信号が使用され得る。あるいは、メモリ 806 に記憶された命令は、プロセッサ 802 によって実行されるとき、この方法に関連するステップを実行するために使用され得る。さらに、DMGO200 および勾配最適化システム 800 の修正された組み合わせも使用され得る。したがって、一例として、モーメンタム圧縮および伸長ステップは、メモリに記憶された命令を使用して実行されてもよいが、勾配最適化は、DMGO200

0に関連するハードウェアロジックを使用して実行されてもよい。ステップ1010は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメントタム値の第1のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメントタム値の第2のセットをメモリから取り出すステップと、を含んでもよい。先に説明したように、記憶フォーマットは、縮小精度フォーマットに対応してもよい。

【0067】

[0055]ステップ1020は、選択された記憶フォーマットを有するモーメントタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメントタム値の第3のセットに変換するステップと、選択された記憶フォーマットを有するモーメントタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメントタム値の第4のセットに変換するステップと、を含んでもよい。先に説明したように、このステップは、記憶フォーマットにおけるモーメントタム値に余分なゼロビットをパディングすることによってモーメントタム値を伸長するステップを含んでもよい。追加される余分なゼロビットは、モーメントタム値の圧縮の一部として切り捨てられたビットの数に依存してもよい。先に説明したように、このステップは、プロセッサ（例えば、図8のプロセッサ802）によって実行される場合、（例えば、図8のメモリ806）に記憶された命令を使用して実行されてもよい。代替的に、このステップは、図2のDMGO200に関連するロジックを使用して実行されてもよい。

【0068】

[0056]ステップ1030は、勾配オプティマイザを用いて、モーメントタム値の第3のセットおよびモーメントタム値の第4のセットを使用して勾配最適化を実行するステップを含み得る。先に説明したように、このステップは、プロセッサ（例えば、図8のプロセッサ802）によって実行される場合、（例えば、図8のメモリ806）に記憶された命令を使用して実行されてもよい。代替的に、このステップは、図2のDMGO200に関連するロジックを使用して実行されてもよい。

【0069】

[0057]ステップ1040は、勾配最適化の次の反復のためのモーメントタム値の第5のセットおよびモーメントタム値の第6のセットを生成するステップを含んでもよい。先に説明したように、このステップは、プロセッサ（例えば、図8のプロセッサ802）によって実行される場合、（例えば、図8のメモリ806）に記憶された命令を使用して実行されてもよい。代替的に、このステップは、図2のDMGO200に関連するロジックを使用して実行されてもよい。

【0070】

[0058]ステップ1050は、符号ビットのみを記憶し、モーメントタム値の第5のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメントタム値の第5のセットを、記憶フォーマットを有するモーメントタム値の第7のセットに変換するステップを含んでもよい。先に説明したように、このステップは、単精度フォーマットのモーメントタム値の符号ビットと7つの最上位ビット（MSB）のみを記憶するステップを含んでもよい。先に説明したように、このステップは、プロセッサ（例えば、図8のプロセッサ802）によって実行される場合、（例えば、図8のメモリ806）に記憶された命令を使用して実行されてもよい。代替的に、このステップは、図2のDMGO200に関連するロジックを使用して実行されてもよい。

【0071】

[0059]ステップ1060は、符号ビットのみを記憶し、モーメントタム値の第6のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメントタム値の第6のセットを、記憶フォーマットを有するモーメントタム値の第8のセットに変換するステップを含んでもよい。先に説明したように、このステップは、単精度フォーマットのモーメントタム値の符号ビットと7つの最上位ビット（MSB）のみを記憶するステップを含んでもよい。先に説明したように、このステップは、プロセッサ（例えば、図8のプロセッサ

802)によって実行される場合、(例えば、図8のメモリ806)に記憶された命令を使用して実行されてもよい。代替的に、このステップは、図2のDMGO200に関連するロジックを使用して実行されてもよい。図10は、ある順序で実行されるある数のステップを説明しているが、異なる順序で追加のまたはより少ないステップが実行されてもよい。

【0072】

[0060]結論として、本開示は、勾配オプティマイザと、L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法に関し、Lは1より大きい整数である。本方法は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットをメモリから取り出すステップと、を含んでもよい。本方法は、選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換するステップと、をさらに含んでもよい。本方法は、勾配オプティマイザを用いて、モーメンタム値の第3のセットおよびモーメンタム値の第4のセットを使用して勾配最適化を実行するステップをさらに含み得る。

【0073】

[0061]選択された記憶フォーマットは、縮小された単精度フォーマットを含んでもよい。訓練フォーマットは、単精度フォーマットまたは倍精度フォーマットを含んでもよい。

【0074】

[0062]選択された記憶フォーマットを有するモーメンタム値の第1のセットを、訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップは、単精度フォーマットのモーメンタム値を形成するために余分なゼロビットをパディングするステップを含んでもよい。本方法は、勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成するステップと、モーメンタム値の第5のセットおよびモーメンタム値の第6のセットのそれぞれを記憶する前に、それぞれのモーメンタム値のそれぞれに関連する符号ビットおよび7つの最上位ビットのみを記憶することによってモーメンタム値の第5のセットおよびモーメンタム値の第6のセットのそれぞれを記憶フォーマットへ変換するステップと、をさらに含むことができる。

【0075】

[0063]勾配最適化を実行するステップは、アダプティブモーメンタム推定アルゴリズムを実装するステップを含む。勾配オプティマイザは、フィールドプログラマブルゲートアレイ(FPGA)を使用して実装されてもよく、勾配オプティマイザは、連続するバーストサイクルが勾配オプティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成されてもよい。

【0076】

[0064]別の例では、本開示は、勾配オプティマイザを含むシステムに関する。システムは、L層を含むニューラルネットワークモデルに関連するモーメンタム値を記憶するように構成されたメモリを含んでもよく、Lは1より大きい整数である。本システムは、(1) 選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第1のセットをメモリから取り出し、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応する、モーメンタム値の第2のセットをメモリから取り出し、(2) 選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換し、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモー

メンタム値の第4のセットに変換し、(3) 勾配オプティマイザを使用して、モーメンタム値の第3のセットとモーメンタム値の第4のセットを用いて勾配最適化を実行する、ように構成されてもよい。

【0077】

[00065]選択された記憶フォーマットは、縮小された単精度フォーマットを含んでもよい。訓練フォーマットは、単精度フォーマットまたは倍精度フォーマットを含んでもよい。

【0078】

[00066]システムはさらに、単精度フォーマットのモーメンタム値または倍精度フォーマットのモーメンタム値を形成するために、余分なゼロビットをパディングするように構成されてもよい。システムはさらに、勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成し、モーメンタム値の第5のセットおよびモーメンタム値の第6のセットのそれぞれを記憶する前に、それぞれのモーメンタム値のそれぞれに関連する符号ビットおよび7つの最上位ビットのみを記憶することによってモーメンタム値の第5のセットおよびモーメンタム値の第6のセットのそれぞれを記憶フォーマットへ変換する、ように構成されてもよい。

【0079】

[00067]システムは、アダプティブモーメンタム推定アルゴリズムを実装するように構成されてもよい。勾配オプティマイザは、フィールドプログラマブルゲートアレイ(FPGA)を使用して実装されてもよく、勾配オプティマイザは、連続するバーストサイクルが勾配オプティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成されてもよい。

【0080】

[00068]さらに別の例では、本開示は、勾配オプティマイザと、L層を含むニューラルネットワークモデルに関連する重みおよびモーメンタム値を記憶するように構成されたメモリと、を含むシステムにおける方法に関し、Lは1より大きい整数である。本方法は、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメンタム値の第1のセットをメモリから取り出すステップと、選択された記憶フォーマットを有する、ニューラルネットワークモデルの層に対応するモーメンタム値の第2のセットをメモリから取り出すステップと、をさらに含んでもよい。本方法は、選択された記憶フォーマットを有するモーメンタム値の第1のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第3のセットに変換するステップと、選択された記憶フォーマットを有するモーメンタム値の第2のセットを、勾配オプティマイザに関連付けられた訓練フォーマットを有するモーメンタム値の第4のセットに変換するステップと、をさらに含んでもよい。本方法は、勾配オプティマイザを用いて、モーメンタム値の第3のセットおよびモーメンタム値の第4のセットを使用して勾配最適化を実行するステップをさらに含んでもよい。本方法は、勾配最適化の次の反復のためのモーメンタム値の第5のセットおよびモーメンタム値の第6のセットを生成するステップをさらに含んでもよい。本方法は、符号ビットのみを記憶し、モーメンタム値の第5のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の第5のセットを、記憶フォーマットを有するモーメンタム値の第7のセットに変換するステップをさらに含んでもよい。本方法は、符号ビットのみを記憶し、モーメンタム値の第6のセットの仮数に関連する所定の数のビットを切り捨てることによって、モーメンタム値の第6のセットを、記憶フォーマットを有するモーメンタム値の第8のセットに変換するステップをさらに含んでもよい。

【0081】

[00069]選択された記憶フォーマットは、縮小された単精度フォーマットを含んでもよい。訓練フォーマットは、単精度フォーマットまたは倍精度フォーマットを含んでもよい。本方法において、勾配最適化を実行するステップは、アダプティブモーメンタム推定アルゴリズムを実装するステップを含んでもよい。

【0082】

【00070】訓練フォーマットは、32ビット浮動小数点フォーマットを含んでもよく、記憶フォーマットは、8ビット浮動小数点フォーマットを含んでもよい。勾配オプティマイザは、フィールドプログラマブルゲートアレイ（FPGA）を使用して実装されてもよく、勾配オプティマイザは、連続するバーストサイクルが勾配オプティマイザを通る勾配のストリーミングをもたらすように、バーストモードで動作するように構成されてもよい。

【0083】

【00071】本明細書に描かれた方法、モジュール、および構成要素は、単に例示的なものであることを理解されたい。代替的に、または追加的に、本明細書で説明される機能は、少なくとも部分的に、1つまたは複数のハードウェア論理構成要素によって実行され得る。例えば、限定するものではないが、使用可能なハードウェア論理構成要素の例示的なタイプには、フィールドプログラマブルゲートアレイ（FPGA）、特定用途向け集積回路（ASIC）、特定用途用標準品（ASSP）、システムオンチップシステム（SOC）、複合プログラマブルロジックデバイス（CPLD）等が含まれる。抽象的ではあるが、明確な意味において、同じ機能を達成するための構成要素の任意の配置は、所望の機能が達成されるように効果的に「関連付けられる」。したがって、特定の機能を実現するために組み合わせられた本明細書の任意の2つのコンポーネントは、アーキテクチャまたは媒体間コンポーネントに関係なく、所望の機能が達成されるように互いに「関連」していると見なすことができる。同様に、そのように関連付けられた任意の2つの構成要素は、所望の機能を達成するために互いに「動作可能に接続される」、または「結合される」とも見なすことができる。

【0084】

【00072】本開示で説明されるいくつかの例に関連する機能は、非一時的媒体に記憶された命令も含み得る。本明細書で使用される「非一時的媒体」という用語は、機械を特定の方法で動作させるデータおよび／または命令を記憶する任意の媒体を指す。例示的な非一時的媒体は、不揮発性媒体および／または揮発性媒体を含む。不揮発性媒体には、例えば、ハードディスク、ソリッドステートドライブ、磁気ディスクまたはテープ、光ディスクまたはテープ、フラッシュメモリ、EPROM、NVRAM、PRAM、またはそのような他の媒体、またはこれらの媒体のネットワーク化されたバージョンが含まれる。揮発性媒体は、例えば、DRAM、SRAM、キャッシュなどの動的メモリ、または他のそのような媒体を含む。非一時的媒体は、伝送媒体とは異なるが、伝送媒体と組み合わせて使用することができる。伝送媒体は、データおよび／または命令を機械にまたは機械から伝送するために使用される。例示的な伝送媒体には、同軸ケーブル、光ファイバーケーブル、銅線、および電波のような無線媒体が含まれる。

【0085】

【00073】さらに、当業者は、上述した操作の機能の間の境界は単に例示的なものであることを認識するであろう。複数の操作の機能は、単一の操作に組み合わせられてもよく、および／または、単一の操作の機能は、追加の操作に分散されてもよい。さらに、代替の実施形態は、特定の操作の複数のインスタンスを含んでもよく、操作の順序は、他の様々な実施形態において変更されてもよい。

【0086】

【00074】本開示は特定の例を提供するが、以下の特許請求の範囲に規定されるように、本開示の範囲から逸脱することなく、様々な修正および変更を行うことが可能である。したがって、明細書および図面は、制限的な意味ではなく例示的な意味で見なされ、すべてのそのような修正は、本開示の範囲に含まれることが意図されている。特定の実施例に関して本明細書に記載される任意の利益、利点、または問題に対する解決策は、任意のまたはすべての請求項の重要な、必要な、または必須の特徴または要素として解釈されることを意図していない。

【0087】

【00075】さらに、本明細書で使用される「a」または「an」という用語は、1つまたは複数として定義される。また、請求項における「少なくとも1つ」および「1つ以上」



などの導入句の使用は、不定冠詞「a」または「a n」による別の請求項要素の導入が、同じ請求項に導入句「1つ以上」または「少なくとも1つ」および不定冠詞「a」または「a n」などが含まれていても、当該導入した請求項要素を含む任意の特定の請求項に当該要素を1つだけ含む発明を制限することを意味するものと解釈すべきではない。定冠詞の使用についても同様である。

【0088】

[00076]特に断らない限り、「第1の」および「第2の」のような用語は、そのような用語が説明する要素を任意に区別するために使用される。したがって、これらの用語は、必ずしも、そのような要素の時間的または他の優先順位を示すことを意図していない。

10

20

30

40

50

【図面】  
【図 1】

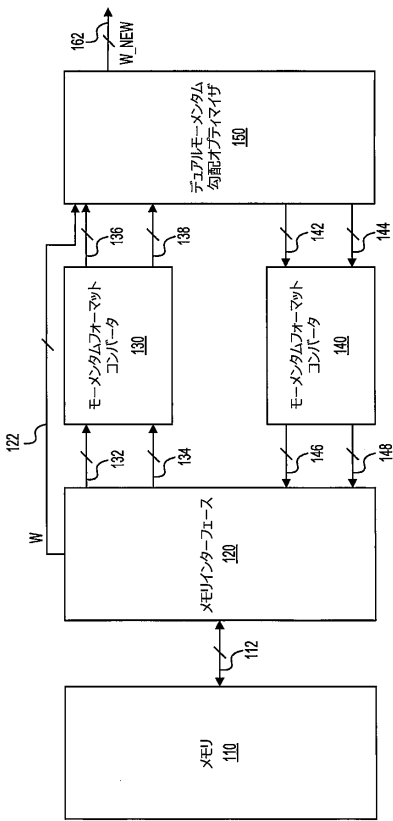


FIG. 1

【図 2】

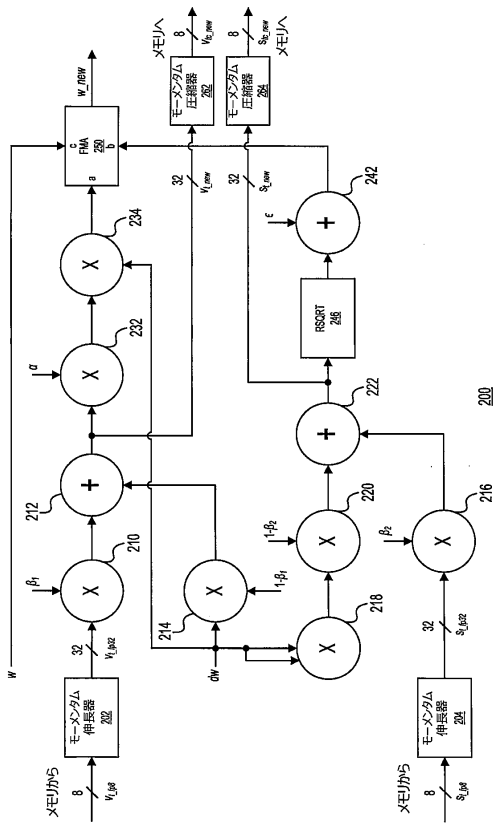


FIG. 2

【図 3】

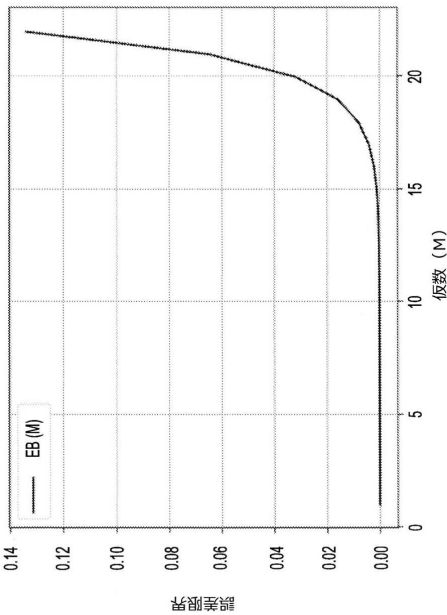


FIG. 3

【図 4】

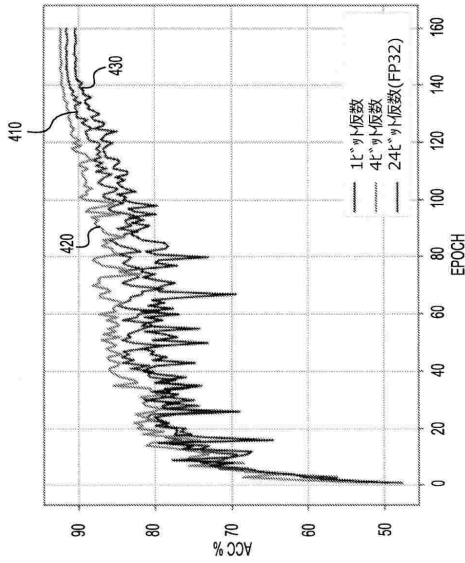


FIG. 4

【図 5】

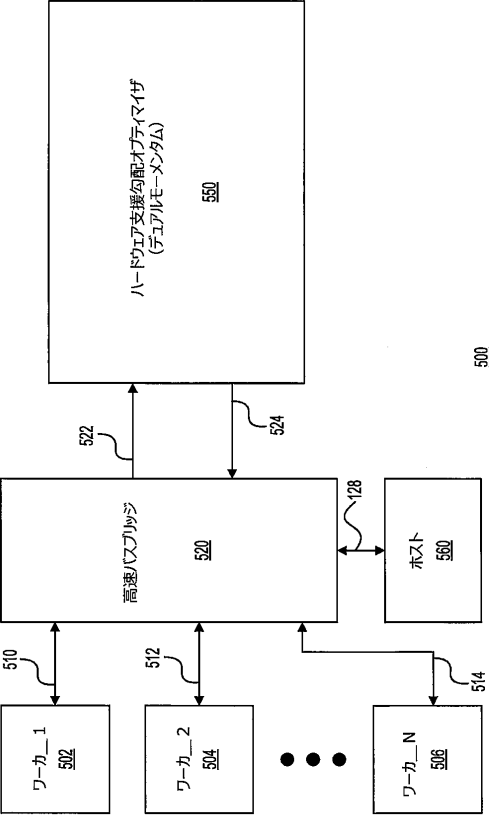


FIG. 5

【図 6】

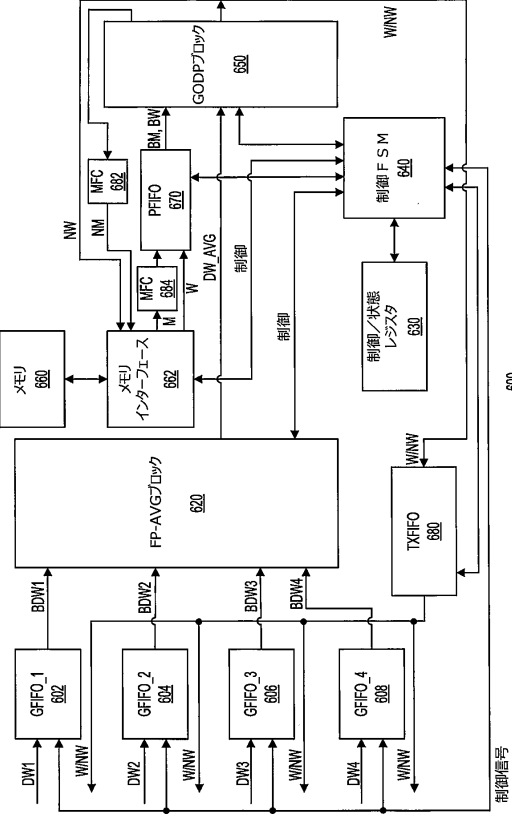


FIG. 6

【図 7】

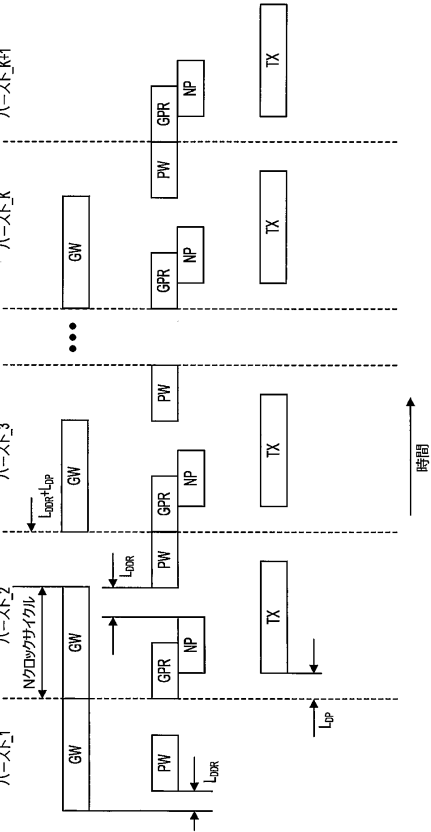


FIG. 7

【図 8】

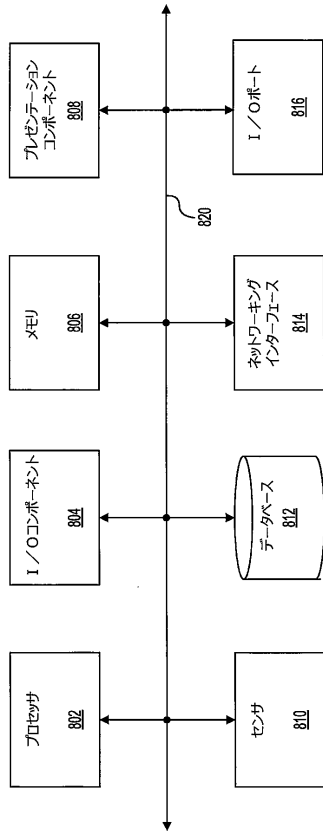
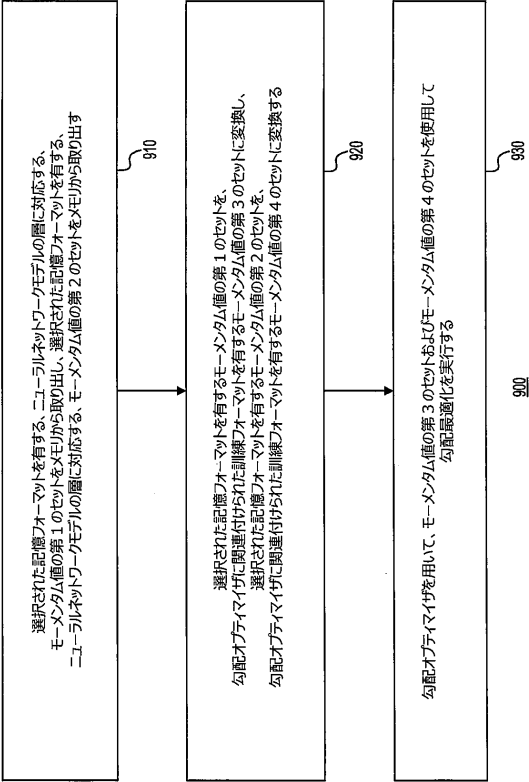


FIG. 8

【図 9】



【図 10】

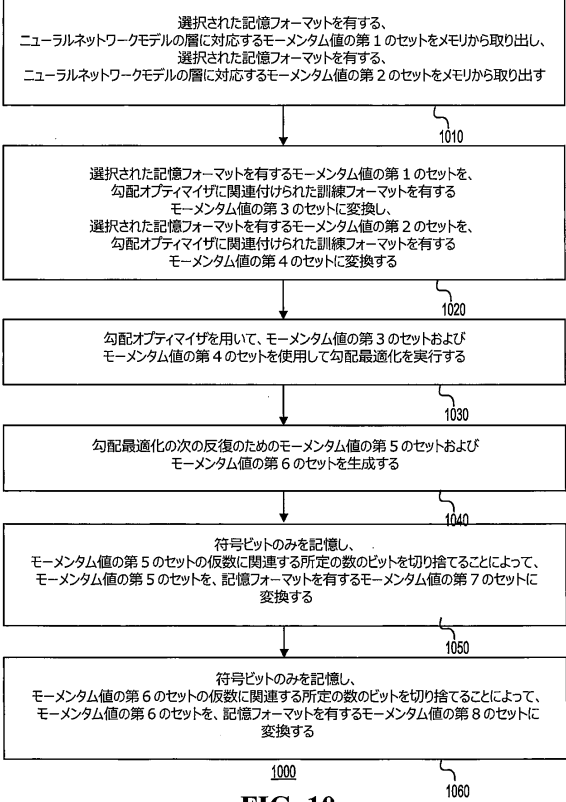


FIG. 10

フロントページの続き

- アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト ウェ  
イ, 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
- (72)発明者 プディペディ, バラドワージ
- アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト ウェ  
イ, 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
- (72)発明者 トレンブレイ, マーク
- アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト ウェ  
イ, 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
- 審査官 新井 則和
- (56)参考文献 米国特許出願公開第2019/0034784 (US, A1)
- 特表2019-512126 (JP, A)
- Sebastian Ruder, An overview of gradient descent optimization algorithms, arxiv.org, [onli  
ne], 2017年06月15日, [検索日 2024.09.09], Retrieved from the Internet: <URL: https://  
arxiv.org/pdf/1609.04747>
- (58)調査した分野 (Int.Cl., DB名)
- G 0 6 N 3 / 0 6 3
- G 0 6 N 3 / 0 8 4
- G 0 6 N 2 0 / 0 0