(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7179853号
(P7179853)

(45)発行日　令和4年11月29日(2022.11.29)

(24)登録日　令和4年11月18日(2022.11.18)

(51)国際特許分類

		FI	
G06N	*3/063(2006.01)*	G06N	3/063
G06F	*15/80 (2006.01)*	G06F	15/80

請求項の数　15　（全51頁）

(21)出願番号	特願2020-531932(P2020-531932)
(86)(22)出願日	平成30年12月10日(2018.12.10)
(65)公表番号	特表2021-506032(P2021-506032A)
(43)公表日	令和3年2月18日(2021.2.18)
(86)国際出願番号	PCT/US2018/064777
(87)国際公開番号	WO2019/118363
(87)国際公開日	令和1年6月20日(2019.6.20)
審査請求日	令和2年8月1日(2020.8.1)
(31)優先権主張番号	15/839,017
(32)優先日	平成29年12月12日(2017.12.12)
(33)優先権主張国・地域又は機関	米国(US)
(31)優先権主張番号	15/839,301
(32)優先日	平成29年12月12日(2017.12.12)

最終頁に続く

(73)特許権者　506329306
アマゾン テクノロジーズ インコーポレイテッド
アメリカ合衆国 ９８１０８－１２２６
ワシントン州 シアトル ピーオー ボックス ８１２２６

(74)代理人　100137969
弁理士　岡部 憲昭

(74)代理人　100104824
弁理士　穐場 仁

(74)代理人　100121463
弁理士　矢口 哲也

(72)発明者　ファン，ランディ
アメリカ合衆国，ワシントン州 ９８１０８－１２２６，シアトル，ピー．オー

最終頁に続く

(54)【発明の名称】　オンチップの計算ネットワーク

(57)【特許請求の範囲】

【請求項１】

集積回路であって、

第１の処理エンジンアレイと、

ニューラルネットワークの重み値セットを記憶する複数のメモリバンクを含むオンチップメモリと、を備え、

前記複数のメモリバンクのうちの各バンクは、独立してアクセス可能であり、かつ前記複数のメモリバンクを含むオンチップメモリおよび前記第１の処理エンジンアレイは、同じダイ上にあり、

入力データを受信した時点で、前記集積回路は、前記重み値セットを用いて、前記ニューラルネットワーク用に規定されたタスクを実行するように構成され、前記タスクを実行することは、

前記オンチップメモリから前記第１の処理エンジンアレイに第１の重み値を読み取ることと、

前記入力データを前記第１の処理エンジンアレイに入力することと、

前記入力データおよび前記第１の重み値に基づいて、前記第１の処理エンジンアレイを使用して第１の中間結果を計算することと、

前記第１の中間結果を前記オンチップメモリに書き込むと同時に、前記オンチップメモリから前記第１の処理エンジンアレイに第２の重み値を読み取ることと、

前記第１の中間結果を前記第１の処理エンジンアレイに入力することと、

前記第１の中間結果および前記第２の重み値に基づいて、前記第１の処理エンジンアレイを用いて結果を計算すること、を含み、前記結果は、前記タスク実行の結果に対応する、集積回路。

【請求項２】

前記タスクを実行することは、

前記複数のメモリバンクのうちの異なるメモリバンクから２つ以上の値を同時に読み取ることをさらに含む、請求項１に記載の集積回路。

【請求項３】

前記２つ以上の値は、重み値、入力値または中間結果を含む、請求項２に記載の集積回路。

【請求項４】

前記タスクを実行することは、

前記複数のメモリバンクのうちの第１のメモリバンクへ第１の値を書き込むことと、

前記複数のメモリバンクのうちの第２のメモリバンクから第２の値を読み取ること、をさらに含み、前記第１の値は、前記第２の値が読み取られると同時に書き込まれる、請求項１～３のいずれか一項に記載の集積回路。

【請求項５】

前記第１の値および前記第２の値は、重み値、および入力値、または中間結果を含む、請求項４に記載の集積回路。

【請求項６】

前記重み値セットは、前記ニューラルネットワークの全ての重み値を含む、請求項１に記載の集積回路。

【請求項７】

第２の処理エンジンアレイをさらに備え、前記複数のメモリバンクのうちの第１のメモリバンクセットは、前記第１の処理エンジンアレイによって使用されるように構成され、前記複数のメモリバンクのうちの第２のメモリバンクセットは、前記第２の処理エンジンアレイによって使用されるように構成され、前記第１のメモリバンクセットおよび前記第２のメモリバンクセットは各々、前記ニューラルネットワークの前記重み値の一部分を含み、前記タスクを実行することは、

前記第１の処理エンジンアレイよって第２の中間結果を計算することであって、前記第１の処理エンジンアレイは、前記第１のメモリバンクセットからの重み値を用いて前記第２の中間結果を計算することと、

前記第１の処理エンジンアレイによって、前記第２のメモリバンクセットから追加の重み値を読み取ること、をさらに含み、前記第１の処理エンジンアレイは、前記第２の中間結果および前記追加の重み値を用いて前記結果を計算する、請求項１に記載の集積回路。

【請求項８】

第２のメモリバンクセットに記憶される前記ニューラルネットワークの前記重み値の前記一部分による占有は、前記第２のメモリバンクセットの全てより少なく、前記第２の処理エンジンアレイは、前記重み値セットによって占有されていない前記第２のメモリバンクセットの一部を用いて計算を実行する、請求項７に記載の集積回路。

【請求項９】

前記第１のメモリバンクセットおよび前記第２のメモリバンクセットに記憶される前記重み値の前記一部分は、前記ニューラルネットワークの全ての重み値を含む、請求項７に記載の集積回路。

【請求項１０】

前記複数のメモリバンクの第１の部分は、前記結果を計算するための前記第１の中間結果を記憶するために初めに予約され、前記重み値セットは、前記ニューラルネットワークの全ての重み値より少ない重み値を含む、請求項１に記載の集積回路。

【請求項１１】

前記タスクを実行することは、

前記中間結果を記憶するために必要なメモリの量が減少していると決定することと、
別のメモリから追加の重み値セットを読み取ることと、
前記追加の重み値セットを前記複数のメモリバンクの前記第１の部分に記憶することであって、前記追加の重み値セットは、必要とされる前に記憶される、記憶すること、をさらに含む、請求項１０に記載の集積回路。
【請求項１２】
前記第１の処理エンジンアレイは、処理エンジンセットを含み、前記処理エンジンセットのうちの各処理エンジンは、結果を、前記処理エンジンセットのうちの別の処理エンジンへ直に出力する、請求項１～１１のいずれか一項に記載の集積回路。
【請求項１３】
前記第１の処理エンジンアレイのうちの各処理エンジンは、乗数累算器回路を含む、請求項１～１２のいずれか一項に記載の集積回路。
【請求項１４】
前記ニューラルネットワークは、重み付き有向グラフから導出される複数の重み値と、前記重み付き有向グラフにおける各ノードについて計算を実行するための命令セットとを含み、前記複数の重み値は、既知の入力データを用いて前記タスクを実行することにより、先に決定されている、請求項１～１３のいずれか一項に記載の集積回路。
【請求項１５】
方法であって、
ニューラルネットワーク処理回路のオンチップメモリであって、複数のメモリバンクを有する前記オンチップメモリに、重み値セットを記憶することであって、前記複数のメモリバンクのうちの各バンクは、独立してアクセス可能であり、前記ニューラルネットワーク処理回路は、前記オンチップメモリと同じダイ上に処理エンジンアレイを含み、前記重み値セットは、入力データの受信に先行して記憶される、記憶することと、
入力データを受信することと、
前記重み値セットを用いて、ニューラルネットワーク用に規定されるタスクを実行すること、を含み、前記タスクを実行することは、
前記オンチップメモリから前記処理エンジンアレイに第１の重み値を読み取ることと、
前記入力データを前記処理エンジンアレイに入力することと、
前記入力データおよび前記第１の重み値に基づいて、前記処理エンジンアレイを使用して中間結果を計算することと、
前記中間結果を前記オンチップメモリに書き込むと同時に、前記オンチップメモリから前記処理エンジンアレイに第２の重み値を読み取ることと、
前記中間結果を前記処理エンジンアレイに入力することと、 前記中間結果および前記第２の重み値に基づいて、前記処理エンジンアレイを使用して結果を計算すること、を含み、前記結果は、前記タスク実行の結果に対応する、方法。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001]ニューラルネットワークは、コンピュータ技術を用いて、動物の脳を構成する生体神経回路網により実行される論理的推論を複製しようとするものである。ニューラルネットワークは、ヒトの脳の動作の力学からインスピレーションを得ている。ニューラルネットワークでは、神経細胞がノードで表され、シナプスがノード間の重み付けされた結合によって表される。重みは、入力に対する様々な応答を反映することができる。ニューラルネットワークは、層状に配置されることが可能であって、分析されるべき入力データは、ある入力層へ提供され、かつ各層の出力は、入力を次の層へ提供する。最後の層は、結果を出力することができる。重み値は、訓練を通じて、結果が既知である入力データがニューラルネットワークへ提供される間に決定されることが可能である。
【０００２】
[0002]ニューラルネットワークは、計算を実行するために、中央処理装置（ＣＰＵ）を

用いて実装されることが可能である。しかしながら、ＣＰＵは、並列計算ではなく逐次計算用に最適化される傾向があり、よって、応答時間が長くなる可能性がある。グラフィック処理装置（ＧＰＵ）は、並列計算用に最適化されているが、必ずしも、１つの計算ユニットからの結果を別の計算ユニットへ直に提供するように最適化されたものではない。多くの場合、結果は、まずメモリに書き込まれなければならない。ＧＰＵは、ＣＰＵより優れた応答時間を有するものの、応答時間に遅れが出ることがある。

【０００３】

[0003]専用ニューラルネットワークプロセッサは、並列の連鎖計算用に最適化された計算アレイを含む。あるニューラルネットワークプロセッサでは、計算ユニットは、結果を別の計算ユニットへ直に出力することができ、結果をメモリに書き込む必要がない。結果を、たとえばそのアレイを介して新たな計算サイクルを開始するために、メモリへ書き込む必要がある場合、結果は、計算アレイに対してローカルなメモリに記憶されることが可能である。したがって、ニューラルネットワークプロセッサは、同じ入力データに対してＣＰＵおよびＧＰＵのいずれよりも優れたパフォーマンスを発揮することができる。

【０００４】

[0004]様々な例について、図面を参照して説明する。

【図面の簡単な説明】

【０００５】

【図１】ニューラルネットワークの視覚モデルの一例を示す。

【図２Ａ】順伝播重みおよび回帰重みを含むニューラルネットワークのモデルの一例を示す。

【図２Ｂ】異なる結合型を含むニューラルネットワークのモデルの一例を示す。

【図３Ａ】２次元畳み込みのモデルの一例を示す。

【図３Ｂ】畳み込みニューラルネットワークのモデルの一例を示す。

【図４】ニューラルネットワークの重み値をオフチップメモリではなくオンチップで記憶する効果の一例を示す。

【図５】ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例を示す。

【図６Ａ】ニューラルネットワークを実行するために使用可能なニューラル処理エンジンの一例を示す。

【図６Ｂ】ニューラルネットワークを実行するために使用可能なニューラル処理エンジンの一例を示す。

【図７Ａ】ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例を示す。

【図７Ｂ】ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例を示す。

【図７Ｃ】ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例を示す。

【図８】複数のニューラルネットワーク処理エンジンを有するニューラルネットワークプロセッサの一例を示す。

【図９】ニューラルネットワークの重みが２つのニューラルネットワーク処理エンジンのメモリサブシステムに記憶されるように構成される、ニューラルネットワークプロセッサの一例を示す。

【図１０】ニューラルネットワークの重みが２つのニューラルネットワーク処理エンジンのメモリサブシステムに記憶されるように構成される、ニューラルネットワークプロセッサの一例を示す。

【図１１】ニューラルネットワークの全ての重みをオンチップで記憶するための別のソリューションを示す。

【図１２】ニューラルネットワークプロセッサがニューラルネットワークを実行している場合にニューラルネットワークプロセッサにより実行されることが可能なプロセスの一例

を示す。
【図１３】ニューラルネットワークプロセッサがニューラルネットワークを実行している場合にニューラルネットワークプロセッサにより実行されることが可能なプロセスの一例を示す。
【図１４】ニューラルネットワークプロセッサを含むコンピューティングシステムにより実行されることが可能なプロセスの一例を示す。
【図１５】これまでに論じた様々な例を使用できるネットワークを示す。
【発明を実施するための形態】
【０００６】
[0022]以下の記述では、様々な実装例について説明する。説明を目的として、実施例を完全に理解するために、特定の構成および詳細について述べる。しかしながら、当業者には、実施例が、これらの特定の詳細なしに実践され得ることも明らかであろう。さらに、実施例の説明を不明瞭にしないように、周知の特徴は、省略または簡略化されることがある。
【０００７】
[0023]人工ニューラルネットワークは、コンピュータ技術を用いて、動物の脳を構成する生体神経回路網により実行される論理的推論を複製しようとするものである。人工ニューラルネットワーク（本明細書では、ニューラルネットワークと称する）は、人が意思決定する方法と同じ方法で意思決定することができるインテリジェントマシンの構築を目指す研究およびエンジニアリングの一分野である、人工知能（ＡＩ）分野の一部である。ニューラルネットワークは、機械学習と呼ばれる人工知能の下位分野に属する。機械学習は、コンピュータに明示的なプログラムなしに学習する能力を与えることを探る研究分野である。機械学習アルゴリズムを実装するプログラムは、プログラムがあらゆる可能性を説明するコード、および考えられる全ての挙動を記述するコードを含有する必要なしに、タスクの実行を学習することができる。
【０００８】
[0024]ニューラルネットワークは、ヒトの脳の動作の力学から、これらの動作が理解される限りにおいて、インスピレーションを得ている。脳の様々なモデルによれば、脳の主要な計算要素は、ニューロンである。ニューロンは、幾つかの要素で互いに結合され、ニューロンに入る要素を樹状突起と称し、ニューロンを出る要素を軸索と称する。ニューロンは、樹状突起を介して信号を受け取り、信号を基に計算を実行し、かつ軸索上へ信号を出力する。入力信号および出力信号を、活性化と称する。あるニューロンの軸索は、枝を出して、複数のニューロンの樹状突起へ結合されることが可能である。軸索の枝と樹状突起との結合部を、シナプスと呼ぶ。
【０００９】
[0025]シナプスは、シナプスを通る信号をスケーリングすることができる。スケーリング係数は、重みと称され、脳による学習を可能にする方法と考えられていて、異なる重みは、入力に対する異なる応答から生じる。学習は、重みを変更することができるが、学習を達成するためにニューロンおよびシナプスの編成が変わる必要はない。したがって、脳の静的構造は、プログラムのモデルとして使用されることが可能であり、重みは、プログラムが既に実行することを学習しているタスクを反映することができる。
【００１０】
[0026]ニューラルネットワークは、ニューロンの計算が入力値の加重和を包含するという概念に基づいて動作する。これらの加重和は、シナプスにより実行される値のスケーリング、およびニューロンにおけるこれらの値の結合に対応する。ニューロンでは、結合された入力に対して関数演算が実行される。脳モデルにおいて、演算は、入力がなんらかのしきい値を越えるときにのみニューロンに出力を生成させる非線形関数であるように見える。したがって、類推により、ニューラルネットワークのノードは、ノードに入力される値の加重和へ非線形関数を適用することができる。
【００１１】

[0027]図１は、ニューラルネットワークの視覚モデルの一例１００を示す。本例において、モデル１００は、入力層１０４と、隠れ層と称する場合も多い中間層１０６と、出力層１０８とを含む。各層は、幾つかのノード１０２を含む。本例において、入力層１０４のノード１０２は、隠れ層１０６の各ノード１０２へ連結されている。脳モデルにおいてシナプスと称される連結部は、重み１１０と称される。また、本例では、隠れ層１０６の各ノード１０２も、出力層の各ノード１０２との間に連結部または重み１１０を有する。入力層１０４は、入力を受信することができ、かつこれらの入力を隠れ層１０６へ伝搬することができる。ニューラルネットワークの実装は、複数の隠れ層を包含することができる。隠れ層１０６（または、複数の隠れ層）により計算される加重和は、出力層１０８へ伝搬され、出力層１０８は、最終的な出力をユーザへ提示することができる。ノード１０２の出力は、脳モデルに合わせて、活性化と呼ぶことができる。

【００１２】

[0028]例示的なモデル１００の各層において起こり得る計算の一例は、次の通りである。

$$y_j = f\left(\sum_{i=1}^{3} W_{ij} \times x_i + b\right)$$

【００１３】

[0029]上述の式において、W_{ij}は、重みであり、x_iは、入力活性化であり、y_jは、出力活性化であり、ｆ（）は、非線形関数であり、ｂは、バイアス項である。様々な非線形関数を用いれば、様々な目的を達成することができる。

【００１４】

[0030]モデル１００は、有向重み付きグラフと呼ぶことができる。有向グラフにおいて、ノードへの、またはノードからの各連結部は、方向（たとえば、ノードへ向かう、またはノードから出る方向）を示す。重み付きグラフにおいて、各連結部は、重みを有することができる。ニューラルネットワークの開発ツールは、理解およびデバッグを容易にするために、ニューラルネットワークを有向重み付きグラフとして視覚化することができる。場合によっては、これらのツールは、ニューラルネットワークを訓練しかつ訓練された重み値を出力するためにも使用可能である。ニューラルネットワークの実行は、次に、入力データに対する計算の実行に重みを用いる問題となる。

【００１５】

[0031]４つ以上の層（たとえば、複数の隠れ層）を有するニューラルネットワークは、深層ニューラルネットワークと呼ばれることがある。深層ニューラルネットワークは、たとえば５～１０００超の層を有し得る。

【００１６】

[0032]多層ニューラルネットワークは、より浅いネットワークを上回る複雑さおよび抽象化によって、高レベル特徴の学習が可能である。一例として、ニューラルネットワークは、画像を認識するようにと教わることが可能である。本例では、画像のピクセルをニューラルネットワークの入力層へ供給することができ、第１の層の出力は、線およびエッジなどの、画像における低レベル特徴の存在を示すことができる。後続層では、これらの特徴を組み合わせて、より高いレベルの特徴の存在尤度を測定することができ、つまりは、線を組み合わせて形にすることができ、これらをさらに組み合わせて形の集合にすることができる。この全ての情報を所与として、ニューラルネットワークは、高レベル特徴が特定のオブジェクトまたはシーンを表す蓋然性を出力することができる。たとえば、ニューラルネットワークは、画像に猫が含まれているか、含まれていないか、を出力することができる。

【００１７】

[0033]ニューラルネットワークの学習フェーズは、ニューラルネットワークの訓練と称

される。訓練の間に、ニューラルネットワークは、タスクを実行することを教示される。タスクの学習において、重み（および場合によってはバイアスも）の値が決定される。ニューラルネットワークの基盤となるプログラム（たとえば、ノードの層編成、各層のノード間の連結、および各ノードが実行する計算）を、訓練中に変える必要はない。ニューラルネットワークは、訓練されると、訓練の間に決定された重み値を用いて結果を計算することにより、タスクを実行することができる。たとえば、ニューラルネットワークは、画像が特定のオブジェクトを包含する蓋然性、オーディオシーケンスが特定のワードを包含する蓋然性、画像におけるオブジェクトを囲むバウンディングボックス、または講じるべき対策案を出力することができる。ニューラルネットワークのプログラムの実行は、推論と称される。

【００１８】

[0034]重みの訓練方法は、幾つか存在する。ある方法は、教師あり学習と称される。教師あり学習では、全ての訓練サンプルにラベルが付けられ、よって、各訓練サンプルをニューラルネットワークに入力すると、既知の結果が生成される。別の方法は教師なし学習と称され、訓練サンプルにラベルは付されず、訓練は、データ内の構造またはデータ内のクラスタを見つけることを目的とする。半教師あり学習は、教師あり学習と教師なし学習との中間に属する。半教師あり学習では、訓練データの部分集合にラベルが付される。ラベルなしデータを用いてクラスタ境界を画定し、かつラベル付きデータを用いてクラスタにラベルを付すことができる。

【００１９】

[0035]ニューラルネットワークは、たとえば画像とビデオ、音声と言語、医学、ゲームプレイおよびロボット工学を含む様々なアプリケーションに使用されてきた。画像とビデオでは、画像の分類、オブジェクトの定位および検出、画像のセグメンテーションおよびアクションの認識にニューラルネットワークが使用されてきた。音声と言語では、音声認識、機械翻訳、自然言語処理およびオーディオ生成にニューラルネットワークが使用されてきた。医療分野では、ゲノミクスおよび医用画像化にニューラルネットワークが使用されてきた。ゲームプレイでは、Ｇｏなどの膨大な数の動作可能性を有するゲームを含むビデオゲームおよびボードゲームのプレイにニューラルネットワークが使用されてきた。ロボット工学では、ロボットのモーションプランニング、ビジュアルナビゲーション、制御安定化および自律車両の運転戦略にニューラルネットワークが使用されてきた。

【００２０】

[0036]ニューラルネットワークは、多様なものが開発されてきた。ニューラルネットワークの様々な例は、順伝播型および回帰型という２つの形態に分けることができる。図２Ａは、入力層２０４と隠れ層２０６との間の順伝播重み２１２と、出力層２０８における回帰重み２１４とを含むニューラルネットワークのモデルの一例２１０を示す。順伝播型ニューラルネットワークにおいて、計算は、先行層の出力に対する一連の演算であって、最終層がニューラルネットワークの出力を生成する。図２Ａに示す例では、順伝播が隠れ層２０６によって示され、そのノード２０２は、入力層２０４のノード２０２の出力のみを操作する。順伝播型ニューラルネットワークにはメモリがなく、所与の入力に対する出力は、ニューラルネットワークに与えられた先行入力に関わりなく、常に同じであり得る。多層パーセプトロン（ＭＬＰ）は、順伝播重みのみを有するニューラルネットワークの一種である。

【００２１】

[0037]これに対して、回帰型ニューラルネットワークは、出力に影響する依存関係を許容し得る内部メモリを有する。回帰型ニューラルネットワークでは、中間演算の中に、内部に保存されかつ後の入力の処理に関連して他の演算への入力として使用されることが可能な値を生成できるものがある。図２Ａの例では、回帰が出力層２０８によって示されていて、出力層２０８のノード２０２の出力は、出力層２０８のノード２０２の入力へ連結し戻される。これらのループバック連結を、回帰重み２１４と称し得る。長・短期記憶（ＬＳＴＭ）は、頻繁に使用される回帰型ニューラルネットワークの一変形例である。

【００２２】
[0038]図２Ｂは、異なる結合型を含むニューラルネットワークのモデルの一例２２０を示す。このモデル例２２０において、入力層２０４および隠れ層２０６は、２２２で完全に結合された層である。完全に結合された層では、全ての出力活性化が、重み付けされた入力活性化で構成される（たとえば、入力層２０４における全てのノード２０２の出力は、隠れ層２０６の入力の全てに結合される）。完全に結合された層は、大量のストレージおよび計算を要する可能性がある。多層パーセプトロンニューラルネットワークは、完全に結合されているニューラルネットワークの一種である。
【００２３】
[0039]アプリケーションによっては、活性化間の幾つかの結合を、たとえば、これらの結合の重みをゼロに設定することにより、出力の精度に影響を与えることなく削除することができる。結果は、図２Ｂの２２４で示す、隠れ層２０６と出力層２０８との間の重みによる、疎に結合された層である。２２４で示す疎に結合された層を実現できる方法の別の例に、プーリングがある。プーリングでは、ノードのクラスタの出力を、たとえば、最大値、最小値、平均値または中央値を見出すことによって組み合わせることができる。
【００２４】
[0040]ニューラルネットワークの動作効率は、幾つかの異なる方法でさらに向上され得る。たとえば、出力に寄与する重みの数は、出力を固定サイズの入力窓のみの関数とすることによって制限されることが可能である。全ての出力の計算に同じ重みセットを用いれば、さらなる効率の獲得が可能である。同じ重み値の反復使用は、重み共有と称され、重みのストレージ要件を大幅に減らすことができる。
【００２５】
[0041]ニューラルネットワーク層におけるウィンドウ処理および重み共有は、各ノードで実行される計算を畳み込みとして構造化することによって達成され得る。図３Ａは、画像処理に適用された２次元畳み込みのモデルの一例３１０を示す。このモデル例において、フィルタ平面３０４は、高さＲおよび幅Ｓを有する行列に配置された重みの集合である。フィルタ平面３０４は、たとえば、要素同士の乗算３１２を用いて、入力画像に適用可能であり、入力画像のデータは、入力特徴マップ３０６と称され得る。フィルタ平面３０４の高さＲおよび幅Ｓは、共に、入力特徴マップ３０６の高さＨおよび幅Ｗより小さく、よって、フィルタ平面３０４を入力特徴マップ３０６へ適用する結果、入力活性化３１６の小さい近傍が計算されることになる（たとえば、近傍を超える重みは、ゼロに設定可能である）。入力活性化３１６は、たとえば部分和の累積３１４を用いて組み合わされ、出力特徴マップ３０８における出力活性化３１８を生成することができる。出力特徴マップ３０８は、入力特徴マップ３０６のより高レベルの抽象化を表し、高さＥおよび幅Ｆを有する。このモデル３１０では、全ての出力に関して同じ重み集合を共有することができる（たとえば、フィルタ空間は、不変である）。
【００２６】
[0042]図３Ｂは、画像処理に適用された、畳み込みニューラルネットワークのモデルの一例３２０を示す。畳み込みニューラルネットワークは、複数の畳み込み層を包含することができる。畳み込みニューラルネットワークでは、各層は、入力データの（すなわち、入力特徴マップ３２６の）連続的高レベルの抽象化を生成することができる。畳み込みニューラルネットワークは、深い階層の層を採用することによって、極めて高いパフォーマンスを実現することができる。
【００２７】
[0043]図３Ｂの例で示すように、畳み込みニューラルネットワークの各畳み込み層は、高次元の畳み込みから構成される。このモデル３２０において、一層の入力活性化３３６は、各々がチャネルＣと称される２次元入力特徴マップ３２６の集合として構造化されている。各チャネルは、各チャネルに１つのフィルタを有するフィルタスタック３２４からの特定の２次元フィルタで畳み込まれる。フィルタスタック３２４は、単一の３次元フィルタと呼ぶことができる。各点の畳み込みの結果は、全てのチャネルに渡って合計され、

合わせて出力特徴マップ３２８のうちの１つのチャネルＭを形成する出力活性化３３８を生成する。この同じ入力上では、出力チャネルの数に対応する追加的な３次元フィルタＭを用いて、追加の出力チャネルを生成することができる。フィルタ重みの再使用を増進するために、図示の例において１～Ｎでラベル付けされている複数の入力特徴マップ３２６をバッチ処理することができる。

【００２８】

[0044]畳み込みニューラルネットワークは、５～１０００超の範囲の層を包含することができる。例によっては、分類を目的として、畳み込み層の後へ１～３層などの少数の完全に結合された層を付すことができる。完全に結合された層は、入力特徴マップにフィルタを付すこともできるが、これらのフィルタのサイズは、入力特徴マップと同じである。したがって、完全に結合された層には、畳み込み層の重み共有特性がない。

【００２９】

[0045]ニューラルネットワークの訓練は、オンラインで、すなわち、ニューラルネットワークが動作していてユーザが利用できる状態において生じ得る。しかしながら、より多くの場合、訓練は、オフラインで、かつニューラルネットワークが稼働される前に生じる。訓練サンプルセットは、膨大であり得、よって、訓練には数時間または数日を要することがある。オフライン訓練は、潜在的に、より正確な結果も生み出し得る。

【００３０】

[0046]ニューラルネットワークは、訓練されると、訓練中に決定された重みと、ネットワークの各層またはノードで実行されるべき計算を記述する命令セットとを包含する。例によっては、重みの数は、約５百万～１億になることがある。例によっては、重み値を３２ビットの数値を用いて表すことができるが、この場合、５百万～１億個の重みは、約２０メガバイト（ＭＢ）～４００ＭＢの記憶容量を必要とし得る。例によっては、重みの数が僅か１５０万であることもある。

【００３１】

[0047]ニューラルネットワークの動作（たとえば、推論の実行）は、上述のモデルが示すように、入力データまたは入力活性化をフェッチすることと、層内の各ノードについて積和演算を並列して実行することと、出力活性化を提供すること、を含む。応答時間で測定されるニューラルネットワークの最適パフォーマンスは、ハードウェアアーキテクチャが高度に並列化された計算を実行できる場合に達成可能である。汎用処理装置と呼ばれることもある中央処理装置（ＣＰＵ）は、複数のコア（たとえば、２～６４個またはこれを超えるコア）を有し得、かつ複数の実行スレッドの使用によって並列処理を高めることができる。しかしながら、ＣＰＵコアは、逐次処理用に最適化される傾向がある。たとえば、コアの計算エンジン（たとえば、算術論理演算装置（ＡＬＵ））は、メモリからオペランドを取得して結果をメモリに書き込み、よって、逐次計算にはメモリ操作が必要である。この例において、各メモリ操作は、ＣＰＵの制御論理による管理を必要とすることがある。したがって、このために、かつ他の理由により、ＣＰＵは、ニューラルネットワークに関して推論を実行する場合に応答時間が遅くなる傾向がある。

【００３２】

[0048]ＣＰＵとは対照的に、グラフィック処理装置（ＧＰＵ）は、具体的には並列計算を実行するように構成される何千もの小さく効率的なコアを備えることによって並列処理を実現する。したがって、ＧＰＵは、ニューラルネットワークの実行に際して、ＣＰＵより遙かに優れたパフォーマンスを達成することができる。しかしながら、個々のＧＰＵ計算エンジンは、本質的にやはり逐次的である可能性があり、よって、ある計算エンジンの出力を別の計算エンジンの入力へ提供するには、メモリ操作が必要である。

【００３３】

[0049]ニューラルネットワークを実行する場合、ＣＰＵおよびＧＰＵの双方が遭遇し得るパフォーマンスのボトルネックは、メモリへのアクセスに存する。積和演算は、各々が重み値、入力特徴マップの活性化および部分和をフェッチする３つのメモリ読取りと、更新された部分和を記憶するための１つのメモリ書込みとを必要とし得る。最悪の事例では

、全てのメモリトランザクションがオフチップメモリへ、すなわちプロセッサとは別のダイ上かつ別のパッケージ内に位置決めされるメモリへ送られる。このメモリは、プロセッサメモリまたはメインメモリと称されることもあり、プロセッサが専ら、プロセッサによってアクティブに作用されているデータを一時的に記憶するためのものであり得る。プロセッサメモリには、大容量かつ低コストであるという理由で、ダイナミック・ランダム・アクセス・メモリ（ＤＲＡＭ）またはＤＲＡＭの変形がよく使用される。しかしながら、プロセッサメモリの読取りおよびこれへの書込みは、計算エンジンの動作より何桁分も低速である。したがって、ニューラルネットワークの速度は、オフチップメモリの待ち時間によって制限される可能性がある。

【００３４】

[0050]専用ニューラルネットワークプロセッサは、ニューラルネットワークの実行に際して、ＣＰＵおよびＧＰＵの双方より優れたパフォーマンスを達成することができる。ニューラルネットワークプロセッサは、計算エンジンが処理チェーンを形成しかつある計算エンジンから別の計算エンジンへ直にデータを渡すことができる空間アーキテクチャを採用することができる。これにより、メモリトランザクションの数を大幅に減らすことができる。例によっては、ニューラルネットワークプロセッサは、プロセッサメモリから読み取られる値を記憶し得る、かつ値をプロセッサ内の複数の計算エンジンへ配信できるオンチップバッファも含むことができる。計算エンジンは、さらに、中間結果を記憶するための小型ローカル・レジスタ・ファイル（たとえば、小型メモリ）を含むことができる。オンチップメモリ階層の保有は、メモリ待ち時間の短縮により、ニューラルネットワークの動作効率を高めることができる。

【００３５】

[0051]それでもやはり、ニューラルネットワークプロセッサは、ニューラルネットワークの重み値がオフチップで記憶される場合に、メモリ帯域幅が制限されることになり得る。ニューラルネットワークプロセッサの計算行列が計算を実行できる速度は、メモリから重み値および活性化を読み取ることができる速度をすぐに超える可能性がある。たとえば、計算行列は、クロックサイクルごとに１０，０００回の積和演算を実行することができ、よって、サイクル当たり３０，０００個の入力値を必要とする。プロセッサメモリバスのクロック速度は、たとえば数千メガヘルツ（ＭＨｚ）の範囲内であり得るのに対して、プロセッサのクロック速度は、ギガヘルツ（ＧＨｚ）の倍数であり得る。したがって、ニューラルネットワークプロセッサの計算速度は、プロセッサメモリのデータ供給能力をすぐに追い越してしまう可能性がある。

【００３６】

[0052]重み値の再使用は、メモリ帯域幅の制限を回避し得る１つの方法である。再使用は、畳み込みニューラルネットワークでは一般的であって、重み値を、たとえば平均１３００回再使用することができる。後に詳述するように、重み値を頻繁に再使用するニューラルネットワークは、潜在的にメモリ帯域幅の制限を回避することができるが、代わりにプロセッサの計算速度が制限される可能性がある。

【００３７】

[0053]長・短期記憶ニューラルネットワークおよび多層パーセプトロンニューラルネットワークでは、重み値の再使用率は、たとえば平均２回などと遙かに低い。

【００３８】

[0054]重み値の再使用を増やすために使用される１つのソリューションが、バッチ処理である。バッチ処理は、一度に２つ以上の入力データセットをニューラルネットワークに入力することを含む。入力データセットが関連づけられる必要はない。バッチ処理では、ニューラルネットワークにたとえば１０セットの入力データが提供される場合、各重みは、メモリから一旦読み取られると、その後２０回（たとえば、入力データセットにつき２回）再使用が可能である。

【００３９】

[0055]しかしながら、数学モデルは、ニューラルネットワークプロセッサが可能な最大

限のパフォーマンスを実現するには、高い再使用率が必要であることを示唆している。たとえば、一部の例は、約１０００の再使用率が必要であることを示唆している。バッチ処理に際しては、たとえば一度に５０～６０セットの入力データを収集することが可能である場合があるが、５００セットの入力データを収集すると、他の問題が生じることがある。たとえば、ニューラルネットワークのユーザは、たとえば機械翻訳または画像識別を要求する場合には迅速な応答を期待する。ニューラルネットワーク処理システムが、結果の計算を開始するまでに５００もの要求の保有を待機する場合、応答時間に悪影響が及ぶ可能性がある。

【００４０】

[0056]様々な実装において、ニューラルネットワーク処理システムは、ニューラルネットワークの重みをオンチップメモリに記憶することによって、メモリ帯域幅の制限を減らすことができ、よって最適効率に近づくことができる。オンチップとは、メモリが、計算行列と同じダイ上および／または同じパッケージ（たとえば、ダイの物理エンクロージャ）内にあることを意味する。ニューラルネットワークプロセッサは、中間結果を記憶するためのオンチップメモリを有し得る。様々な実装において、プロセッサのメモリサブシステムは、オンチップメモリが中間結果および重み値の双方を記憶できるように設計されることが可能である。ニューラルネットワークプロセッサは、それでもメモリバウンドである場合があるが、オンチップメモリを、たとえばオフチップメモリより１０～５０倍は速く読み取ることが可能であり得る。メモリの遅延をこれだけ減らせば、ニューラルネットワークの動作をプロセッサの計算速度限界まで近づけることが可能であり得る。

【００４１】

[0057]事例によっては、特に小型ニューラルネットワークの場合、ニューラルネットワークの重み値を全てオンチップメモリに記憶することが可能であり得る。しかしながら、単一のモノリシックメモリを用いる場合は、単一のメモリが保有する読取りおよび書込みチャネルセットがたとえば１つまたは２つでしかないことがあり、よって一度に１つまたは２つの値しか読み取れないという理由で、やはりメモリ遅延が生じる場合がある。様々な実装では、１つの大型メモリの代わりに、ニューラルネットワークプロセッサに複数のメモリバンクを装備することができ、これらは個々にアクセス可能であり得る。独立してアクセス可能であることにより、同時に２つ以上のメモリバンクを読み取ることが可能であり得る。

【００４２】

[0058]ニューラルネットワークの処理エンジンでは、計算行列を処理エンジンアレイとして実装することができる。ニューラルネットワークの処理エンジンは、さらに、処理エンジンアレイに対してローカルなメモリバンクセットを含むことが可能であり、ここで、ローカルとは、処理エンジンアレイに物理的に近い、かつ／または処理エンジンアレイによって直にアクセス可能であることを意味し得る。先に述べたように、ローカル・メモリ・バンクは、ニューラルネットワークの処理エンジンにより、中間結果を記憶するために使用可能である。事例によっては、特にニューラルネットワークが小型である場合、ニューラルネットワークの重み値を全てニューラルネットワーク処理エンジンのメモリバンクに記憶することもできる。これらの事例では、処理エンジンアレイは、全てのクロックサイクルにおける完全な利用を持続することが可能であり得る。

【００４３】

[0059]例によっては、ニューラルネットワークの重み値が全てニューラルネットワーク処理エンジンのメモリバンクに収まるわけではない。たとえば、メモリバンクには、重み値の半分に足るスペースを有し得、残りのスペースは、入力データセットの処理中に計算される中間結果を記憶するために必要とされる。しかしながら、中間結果のサイズは、結果を計算する過程で減少する可能性がある。さらに、重み値の中には、一度使用されると不要になり得るものがある。したがって、実装によっては、計算が進行してメモリ空間が利用可能になるにつれて、ニューラルネットワークの処理エンジンは、追加の重みを利用可能なスペースにロードすることができる。事例によっては、重みは、オフチップメモリ

から取得されることが可能である。事例によっては、重みは、オンチップメモリから、たとえば、別のニューラルネットワーク処理エンジンのメモリバンクから、取得されることが可能である。

【００４４】

[0060]実装によっては、ニューラルネットワークプロセッサは、各々が独立した処理エンジンアレイとローカル・メモリ・バンクとを有する複数のニューラルネットワーク処理エンジンで構築されることが可能である。これらの実装では、各ニューラルネットワーク処理エンジンが１つのニューラルネットワークを実行でき、よって、複数のニューラルネットワークが同時に実行されることが可能である。実装によっては、あるニューラルネットワークに関する重み値は、そのうちの１つがこのニューラルネットワークを処理するためのエンジンとして指定されている２つ以上のニューラルネットワーク処理エンジンのメモリバンクに記憶されることが可能である。指定されたニューラルネットワーク処理エンジンが、別のニューラルネットワーク処理エンジンによって記憶されている重みを必要とする場合、これらの重みは、他のニューラルネットワーク処理のメモリバンクから読み取られ、かつ指定されたニューラルネットワーク処理エンジンのメモリバンクへロードされることが可能である。他のニューラルネットワーク処理エンジンは、その固有のメモリバンクにおける残りのあらゆる利用可能なスペースを他の動作用に用いることができる。

【００４５】

[0061]実装によっては、あるニューラルネットワークプロセッサから別のニューラルネットワークプロセッサへ重みを移動させる代わりに、計算を移動させることができる。たとえば、中間結果（たとえば、ある層からの出力活性化）および状態（たとえば、計算された最後の層）は、１つのニューラルネットワーク処理エンジンから第２のニューラルネットワーク処理エンジンへコピーされることが可能であり、この場合、第２のニューラルネットワーク処理エンジンは、そのメモリバンクに、計算の継続に必要な次の重み値セットを有する。第２のニューラルネットワーク処理エンジンは、計算を再開し、かつおそらくは、計算をさらに別のニューラルネットワーク処理エンジンへ引き渡すことができる。

【００４６】

[0062]進行中の計算の、あるニューラルネットワーク処理エンジンから別のニューラルネットワーク処理エンジンへの転送は、実装によっては、個々のニューラルネットワークプロセッサ間での転送を含む。これらの実装では、個々のニューラルネットワークプロセッサは、異なるダイ上および／または異なるパッケージ内に存在し得る。同じくこの実施例において、ニューラルネットワークプロセッサは、ホストバスまたはプロセッサバスを用いて通信することができる。ニューラルネットワーク処理エンジンが同じダイ上に存在する場合のように、中間結果および状態をコピーすると、あるニューラルネットワークプロセッサから別のニューラルネットワークプロセッサへ計算を移動させることができる。

【００４７】

[0063]様々な実装において、あるニューラルネットワーク処理エンジンから別のニューラルネットワーク処理エンジンへの重みのコピー、ニューラルネットワーク処理エンジン間および／または物理ニューラルネットワークプロセッサチップ間での進行中の計算の移動は、ニューラルネットワークの重み値をオンチップで可能な限り多く記憶することを目的として、様々な組合せで使用されることが可能である。重み値をオンチップで有することにより、計算は、オフチップメモリの比較的長い待ち時間によって制限される代わりに、単にオンチップメモリの比較的短い待ち時間によって制限され得る。その結果、ニューラルネットワークの動作を遙かに効率的なものにすることができる。

【００４８】

[0064]図４は、ニューラルネットワークの重み値をオフチップメモリではなくオンチップで記憶する効果の一例を示す。図４に示すグラフ４００は、ルーフラインモデルと呼ばれるもののアプリケーションを示している。ルーフラインモデルは、コンピューティングシステムのパフォーマンス推定を提供するために使用可能な性能モデルである。ルーフラインモデルは、固有のハードウェア限度および最適化の潜在的利点を捉えることができる

。図４の例において、ルーフラインモデルは、ニューラルネットワークプロセッサの、メモリから読み取られる重み当たりの演算回数に関する性能を示すために使用されている。縦軸は、１秒間に実行できる兆単位の演算数（ｔｅｒａｏｐｓ）を示す。横軸は、重み値当たりで実行される演算または計算の回数を示す。重み値当たりで実行される演算の回数は、重み固有の再使用（たとえば、ニューラルネットワークの構造は、重みの再使用に繋がる）、またはバッチ処理、すなわち複数のデータセットをニューラルネットワークへ同時に、またはパイプライン式に入力すること、のいずれかを介して増加し得る。

【００４９】

[0065]図４の例において、グラフ４００に描かれている実線４１０は、重み値をオフチップメモリに記憶するニューラルネットワーク処理システムの性能の一例を示している。このようなシステムでは、重み値がプロセッサメモリに記憶されていて、ニューラルネットワークプロセッサがホストバスまたはプロセッサバス上で重み値を読み取る。重み値を別個のメモリに記憶することにより、ニューラルネットワーク処理システムは、メモリから重み値が読み取られるたびに必ず遅延を被ることになる。

【００５０】

[0066]実線４１０の急勾配部４０２では、重み値当たりで実行される演算回数に対して、１秒間に実行できるテラ演算数がほぼ線形的に増加する。実線４１０の急勾配部４０２では、１秒当たりのテラ演算数を増やすために、所与の重みの再使用が増加されなければならない。逆に言えば、実線４１０の急勾配部４０２では、所与の再使用値において、１秒当たりのテラ演算数は、オフチップメモリから重み値を読み取れる速度によって制約される。したがって、ニューラルネットワーク処理システムは、実線４１０の急勾配部４０２においてメモリバウンドであるとされる。

【００５１】

[0067]実線４１０の平坦部４０４では、１秒当たりのテラ演算数が最大に達している（図示の例では、１秒当たり約１２４兆回の演算）。実線４１０の平坦部４０４では、システムがハードウェアの最大計算速度に達していることから、重み値の再使用が増加しても、１秒当たりのテラ演算数は増加しない。実線４１０の平坦部４０４において、システムは、コンピュートバウンドであるとされる。

【００５２】

[0068]したがって、このルーフラインモデルは、ニューラルネットワーク処理システムの場合、ハードウェアで可能な最大パフォーマンスを実現できることを示している。しかし、グラフ４００に示すように、重みがプロセッサとは別のチップに記憶されているシステムの場合、各重み値を何度も再使用すること（図示の例では、約１６００回）が必要である。先に述べたように、畳み込みニューラルネットワークは、高い重み再使用率を有し得るが、多層パーセプトロンおよび回帰型ニューラルネットワークなどの他の多くのニューラルネットワークの重み再使用率は、極めて低い。これらのタイプのニューラルネットワークの入力データセットのバッチ処理は、再使用を増やし得るが、１１００個の入力データセット（各重み値に固有の再使用を２回と想定）をバッチ処理すると、入力データを提供するユーザに対する応答時間の遅れなどの他の有害な影響が生じる可能性がある。

【００５３】

[0069]図４の例における破線４２０は、重み値がオンチップで記憶されるニューラルネットワーク処理システムに適用されたルーフラインモデルを示す。この事例において、重み値は、システムが入力データを受信する前、およびおそらくはシステム動作中のあらゆる時点で、ニューラルネットワークプロセッサのメモリに記憶される。重み値をオンチップで記憶することは、メモリからの重み値の読取りに対する遅延が大幅に短縮されることを意味する。破線４２０の急勾配部４１２が示すように、（本例が提供する）僅か約１００回の再使用でメモリの待ち時間を克服することができる。したがって、ニューラルネットワーク処理システムは、ほとんどの時間および全ての型式のニューラルネットワークタイプについて、破線４２０の平坦部４１４で動作することができる。

【００５４】

[0070]様々な実装において、ニューラルネットワークプロセッサは、重み値をプロセッサ上へ記憶することをサポートするように構築されることが可能である。図５は、ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例５０２を示す。様々な実装において、この例示的なニューラルネットワーク処理エンジン５０２は、ニューラルネットワークプロセッサに包含され得る集積回路である。ニューラルネットワークプロセッサは、追加的なニューラルネットワーク処理エンジン５０２を含む他のコンポーネントを有し得る集積回路である。様々な実装において、ニューラルネットワーク処理エンジン５０２は、メモリサブシステム５０４と、処理エンジンアレイ５１０とを含み得る。動作中（たとえば、入力データ５５０のセットの結果を計算しているとき）、処理エンジンアレイ５１０は、メモリサブシステム５０４から重み５０６および状態５０８の値を読み取ることができる。処理エンジンアレイ５１０は、計算結果を結果バッファ５１２へ出力することができる。事例によっては、この例示的なニューラルネットワーク処理エンジン５０２は、結果がメモリサブシステム５０４へ書き込まれる前に、計算結果に対して、活性化ブロック５１６を用いる活性化関数、および／またはプーリングブロック５１８を用いるプーリングを実行することができる。

【００５５】

[0071]重み５０６は、本例では、ニューラルネットワークの重み値である。様々な実装において、重み５０６は、重み５０６の値が先に決定されていることを意味する訓練後の重みである。状態５０８は、本例では、計算開始時の入力データ５５０だけでなく、進行中の計算を反映する値も包含することができる。状態５０８は、たとえば、動作中のニューラルネットワークのカレント層である処理エンジンアレイ５１０によって決定される部分和、および／または処理エンジンアレイ５１０に対する命令を包含し得、ここで、該命令は、特定の一層に関連づけられてもよい。重み５０６および状態５０８は、処理エンジンアレイ５１０によって作用されるべくメモリサブシステム５０４から読み取られ得る。

【００５６】

[0072]様々な実装において、メモリサブシステム５０４は、複数のメモリバンク５１４を含み得る。これらの実装において、各メモリバンク５１４は、独立してアクセス可能であるが、これは、１つのメモリバンクの読取りが別のメモリバンクの読取りに依存しないことを意味する。同様に、あるメモリバンクへの書込みが、異なるメモリバンクへの書込みに影響を与えることも、これを制限することもない。事例によっては、各メモリバンクの読取りおよび書込みを同時に行うことができる。独立してアクセス可能なメモリバンク５１４を有するためには、様々な技術を用いることが可能である。たとえば、各メモリバンクは、少なくとも１つの読取りチャネルを有することができ、かつ少なくとも１つの別個の書込みチャネルを有してもよい（事例によっては、メモリバンクは、読取りチャネルと書込みチャネルとの組合せを有することがある）。これらの例において、メモリサブシステム５０４は、複数のメモリバンクの読取りチャネルまたは書込みチャネルへの同時アクセスを許容することができる。別の例として、メモリサブシステム５０４は、たとえば複数のメモリバンク５１４の出力間のアービトレーションによって２つ以上のメモリバンクの出力が認可されることになるように、アービトレーション論理を包含することができる。これらの例および他の例において、各メモリバンクは、全体としてはメモリサブシステム５０４によって管理されるものの、他のメモリバンクとは独立して動作されることが可能である。

【００５７】

[0073]メモリバンク５１４を独立してアクセス可能とすることにより、ニューラル処理エンジン５０２の効率は、高まり得る。たとえば、重み５０６および状態５０８は、同時に読み取られて処理エンジンアレイ５１０の各ロウへ提供されることが可能であり、よって、処理エンジンアレイ５１０全体を１サイクルで起動することができる。別の例として、重み５０６および状態５０８は、中間結果がメモリサブシステム５０４へ書き込まれると同時に読み取られることが可能である。これに対して、単一のメモリは、重み５０６および状態５０８を処理エンジンアレイ５１０へなおもオフチップメモリより高速で提供で

きるものの、一度に提供できるものは１つの読取りまたは書込みであることがある。単一のメモリの場合、たとえば、処理エンジンアレイ５１０を開始できるまでに処理エンジンアレイ５１０の各ロウの重みを読み取るべく、複数のクロックサイクルが必要とされる可能性がある。

【００５８】

[0074]様々な実装において、メモリサブシステム５０４は、処理エンジンアレイ５１０、活性化ブロック５１６、プーリングブロック５１８、およびチップ相互接続５２０上でメモリサブシステム５０４へアクセスするあらゆる外部クライアントを含む複数のクライアントへ同時にサービスを提供するように構成されることが可能である。実装によっては、複数のクライアントへサービスを提供できることは、メモリサブシステム５０４が少なくとも、存在するクライアントの数と同数のメモリバンクを有することを意味し得る。事例によっては、処理エンジンアレイ５１０の各ロウは、別々の読取りクライアントと見なされ得る。これらの事例において、重み５０６および状態５０８は、別々に記憶され、よって２つの読取りを要する場合もあれば、連結されて共に記憶され、よって１つの読取りを要する場合もある。事例によっては、処理エンジンアレイ５１０の各カラムは、中間値を出力することができ、よって、各カラムが別々の書込みクライアントと見なされ得る。事例によっては、処理エンジンアレイ５１０からの出力は、メモリバンク５１４に書き込まれることが可能であり、メモリバンク５１４は、続いて、処理エンジンアレイ５１０に入力データを提供することができる。メモリバンク５１４は、たとえば、スタティック・ランダム・アクセス・メモリ（ＳＲＡＭ）を用いて実装されることが可能である。

【００５９】

[0075]様々な実装において、メモリサブシステム５０４は、制御論理を含み得る。制御論理は、たとえば、メモリバンク５１４の各々のアドレス空間を把握し、必要に応じて、読み取る、または書き込むべきメモリバンク５１４を識別し、かつ／または必要であればメモリバンク５１４間でデータを移動することができる。実装によっては、メモリサブシステム５０４は、特定のクライアントへどのメモリバンクを出力するか、かつ／または特定のクライアントからどのメモリバンクの入力を受け入れるかを選択するためのマルチプレクサを包含することができる。これらの実装において、制御論理は、メモリバンク５１４の一部または全てが各クライアントにサービスを提供することを可能にし得る、マルチプレクサのための選択信号を生成することができる。実装によっては、メモリバンク５１４は、特定のクライアントへ結線で接続されることが可能である。たとえば、メモリバンク５１４のセットは、重み５０６および状態５０８を提供すべく処理エンジンアレイ５１０のロウへ結線で接続されることが可能である。これらの例において、制御論理は、たとえば、中間結果が書き込まれるメモリバンク５１４から、次の計算ラウンドのために中間結果が読み取られるメモリバンク５１４へ中間結果を移動するために、データをメモリバンク５１４間で移動させることができる。

【００６０】

[0076]処理エンジンアレイ５１０は、ニューラル処理エンジン５０２の計算行列である。処理エンジンアレイ５１０は、たとえば、とりわけ並列積分、畳み込み、相関および／または行列乗算を実行することができる。処理エンジンアレイ５１０は、ロウおよびカラムに配列される複数の処理エンジン５１１を含み、よって、１つの処理エンジン５１１により出力される結果は、別の処理エンジン５１１へ直に入力されることが可能である。したがって、処理エンジンアレイ５１０の外縁上にない処理エンジン５１１は、メモリサブシステム５０４ではなく、他の処理エンジン５１１から作用すべきデータを受け入れることができる。

【００６１】

[0077]様々な例において、処理エンジンアレイ５１０は、シストリック実行を用い、この場合、データは、各処理エンジン５１１に異なる方向から定間隔で到着する。例によっては、入力データは、処理エンジンアレイ５１０に左から流れ込むことができ、重み値は、上部にロードされることが可能である。例によっては、重みおよび入力データは、左か

ら流れることができ、かつ部分和は、上から下へ流れることができる。これらの例および他の例において、積和演算は、処理エンジンアレイ５１０を介して対角波面として移動し、データは、アレイを右へ、かつ下へ移動する。制御信号は、重み５０６と同時に左側で入力されることが可能であって、計算と共に横へかつ下へ流れることができる。

【００６２】

[0078]様々な実装において、処理エンジンアレイ５１０におけるカラムの数は、処理エンジンアレイ５１０の計算能力を決定し、かつロウの数は、処理エンジンアレイ５１０の最大利用を達成するために必要なメモリ帯域幅を決定する。処理エンジンアレイ５１０は、たとえば、６４のカラムおよび２５６のロウ、または他の何らかの数のカラムおよびロウを有し得る。

【００６３】

[0079]図５には、処理エンジン５１１の一例が示されている。この例が示すように、処理エンジン５１１は、乗数累算器回路を包含することができる。左側からの入力には、たとえば、入力データｉおよび重み値ｗが含まれ得、ここで、入力データは、入力データのセットから取得される値であり、重み値は、ニューラルネットワークの一層を次の層へ連結する重み値セットからの値である。入力データのセットは、たとえば、とりわけ識別またはオブジェクト認識のために提出される画像、音声認識のために提供されるオーディオクリップ、自然言語処理または機械翻訳のためのテキスト文字列、または次の移動を決定するために分析を要求するゲームのカレント状態であってもよい。例によっては、入力データおよび重み値は、次の処理エンジン５１１への入力として右側へ出力される。

【００６４】

[0080]図示の例において、上記からの入力は、別の処理エンジン５１１から、または処理エンジンアレイ５１０による先の計算ラウンドから、のいずれかで提供される部分和ｐ＿ｉｎを包含し得る。新しい入力データセットの計算を開始する場合、処理エンジンアレイ５１０の最上ロウは、ゼロなどのｐ＿ｉｎの固定値を受け取ることができる。本例が示すように、ｉおよびｗは互いに乗算されて結果がｐ＿ｉｎと合計され、別の処理エンジン５１１へ入力されることが可能な新しい部分和ｐ＿ｏｕｔが生成される。処理エンジン５１１の実装は、他にも様々なものが可能である。

【００６５】

[0081]処理エンジンアレイ５１０の最終ロウからの出力は、一時的に結果バッファ５１２に記憶されることが可能である。結果は、中間結果であり得、これは、追加計算のために処理エンジンアレイ５１０へ提供されるべくメモリバンク５１４へ書き込まれることが可能である。あるいは、結果は、最終結果であり得、これは、メモリバンク５１４へ書き込まれると、メモリサブシステム５０４からチップ相互接続５２０上で、該システムにより出力されるべく読み取られることが可能である。

【００６６】

[0082]実装によっては、ニューラル処理エンジン５０２は、活性化ブロック５１６を含む。これらの実装において、活性化ブロック５１６は、処理エンジンアレイ５１０からの結果を組み合わせて１つまたは複数の出力活性化にすることができる。たとえば、畳み込みニューラルネットワークの場合、複数のチャネルからの畳み込みを合計して、単一チャネルのための出力活性化を生成することができる。他の例では、ニューラルネットワーク内の単一ノード用の出力活性化を生成するために、処理エンジンアレイ５１０内の１つまたは複数のカラムからの結果を累積することが必要とされる場合がある。実装によっては、活性化ブロック５１６がバイパスされてもよい。

【００６７】

[0083]実装によっては、ニューラル処理エンジン５０２は、プーリングブロック５１８を含む。プーリングは、ニューラルネットワークの一層からのノードクラスタの出力を組み合わせることである。組み合わされた出力は、次の層へ提供されることが可能である。組合せは、たとえば、最大値、最小値、平均値、中央値、またはノードクラスタの出力から決定される他の何らかの値を計算することを含み得る。様々な実装において、プーリン

グ５１８は、特定のニューラルネットワークにとっての必要性に応じて、選択的に活性化されることが可能である。

【００６８】

[0084]入力データ５５０は、チップ相互接続５２０を介して到着することができる。チップ相互接続５２０は、ニューラル処理エンジン５０２を、入力／出力（Ｉ／Ｏ）デバイス、ストレージドライブまたはネットワークインタフェースから入力データ５５０を入手できるダイレクト・メモリ・アクセス（ＤＭＡ）エンジンなどのニューラルネットワークプロセッサの他のコンポーネントへ接続することができる。入力データ５５０は、たとえば、文字列または数列などの１次元データ、または画像のピクセル値、またはオーディオ信号の経時的な周波数および振幅値のアレイなどの２次元データであり得る。例によっては、入力データ５５０は、たとえば、自動運転車によって使用される状況情報の場合がそうであり得るように、３次元的であり得る。実装によっては、メモリサブシステム５０４は、入力データ５５０用に別個のバッファを包含することができる。実装によっては、入力データ５５０は、重み５０６と共にメモリバンク５１４に記憶されることが可能である。

【００６９】

[0085]様々な実装において、メモリサブシステム５０４に記憶される重み５０６は、１つまたは複数のタスクを実行するようにニューラルネットワークを訓練することによって決定されている可能性がある。入力データ５５０は、実行すべきタスク（たとえば、画像処理、音声認識、機械翻訳、他）を示す命令を含み得る。様々な実装において、ニューラル処理エンジン５０２は、ニューラルネットワークを訓練するためではなく、推論を実行する（たとえば、タスクを実行する）ように構成される。実装によっては、ニューラル処理エンジン５０２が訓練に使用されることがあるが、おそらくは、記憶された重み５０６を更新する際にはソフトウェアから支援される。

【００７０】

[0086]様々な実装において、メモリサブシステム５０４は、ニューラルネットワークの中間結果および全ての重み値の双方を記憶するに足るメモリを包含し得る。メモリサブシステム５０４は、最小でも、メモリサブシステム５０４内に中間結果を記憶するに足るメモリを有するべきであるが、多くの場合、メモリサブシステム５０４は、単に中間結果を記憶するために必要とされるものより遙かに多いメモリバンク５１４を含み得る。この追加スペースは、ニューラルネットワークの全ての重み値を記憶するために使用されることが可能である。たとえば、ニューラルネットワークは、１５０万個の重みを有することがあるが、その各々が３２ビットで表される場合、これにより、約６ＭＢのメモリが必要となり得る。中間結果は、たとえば、最大１０ＭＢの記憶スペースを要し得る。２０ＭＢのオンチップメモリは、適度なサイズであり、よって、先の例では、重み値、中間結果、およびニューラル処理エンジン５０２が動作中に必要とし得る他のあらゆるデータに容易に対応することができる。

【００７１】

[0087]事例によっては、ニューラルネットワークの重みが全てニューラル処理エンジンのメモリに収まるわけではない。これらの事例では、重みの一部がニューラル処理エンジンのメモリに記憶され、かつ残りの重みは、計算の過程でロードされることが可能である。図６Ａ～６Ｂは、ニューラルネットワークを実行するために使用可能なニューラル処理エンジンの一例６０２を示す。様々な実装において、ニューラルネットワーク処理エンジン６０２は、メモリサブシステム６０４と、処理エンジンアレイ６１０とを含み得る。動作中（たとえば、入力データ６５０のセットの結果を計算しているとき）、処理エンジンアレイ６１０は、メモリサブシステム６０４から重み６０６および状態６０８の値を読み取ることができ、ここで、状態６０８の値は、当初、入力データ６５０を含み得る。図示されていないが、処理エンジンアレイ６１０は、処理エンジンアレイを包含し得、これらの処理エンジンの各々が、たとえば積和演算を実行することができる。処理エンジンアレイ６１０は、計算結果を結果バッファ６１２へ出力することができる。事例によっては、この例示的なニューラルネットワーク処理エンジン６０２は、結果がメモリサブシステム

６０４へ書き込まれる前に、計算結果に対して、活性化ブロック６１６を用いる活性化関数、および／またはプーリングブロック６１８を用いるプーリングを実行することができる。

【００７２】

[0088]動作において、ニューラルネットワーク処理エンジン６０２は、メモリサブシステム６０４のメモリバンク６１４から重み６０６および状態６０８を読み取り、重み６０６および状態６０８を処理エンジンアレイ６１０へ入力する。処理エンジンアレイ６１０は、中間結果を結果バッファ６１２へ出力することができ、中間結果は、結果バッファ６１２からメモリサブシステム６０４へ書き込まれることが可能である。複数の中間結果を組み合わせて１つの出力活性化にする場合は、活性化ブロック６１６が選択的に有効化され得る。また、プーリングブロック６１８も、ノードクラスタの出力を組み合わせて１つの出力にする場合に、選択的に起動され得る。重み６０６および状態６０８を読み取って結果を計算するサイクルは、ニューラルネットワークの最終層に到達して処理エンジンアレイ６１０が最終結果を出力するまで、追加的に生じ得る。

【００７３】

[0089]様々な実装において、メモリバンク６１４のセットは、当初、中間結果（たとえば、状態６０８の値）を記憶するために予約され得る。たとえば、図６Ａに示すように、メモリバンク６１４の半分は、即時的な結果を記憶するために予約され得る。この例において、重み６０６は、残りのメモリバンク６１４に記憶され得る。重み６０６は、ニューラルネットワーク処理エンジン６０２が入力データ６５０を受信するより前に記憶されることが可能であり、よって、入力データ６５０が受信されるとすぐに重み６０６を使用可能である。他の例では、メモリバンク６１４を、状態６０８の記憶と重み６０６の記憶との間で比率を変えて分割することができる。実装によっては、個々のメモリバンク６１４を状態６０８の記憶と重み６０６の記憶との間で分割することができる。

【００７４】

[0090]図６Ａの例において、メモリバンク６１４に記憶される重み６０６は、ニューラルネットワークの全ての重みより少ない。残りの重みは、ニューラルネットワーク処理エンジン６０２の外部に位置決めされる追加のメモリに記憶される。追加のメモリ６３０は、たとえば、オンチップキャッシュ内などのオンチップであり得る。あるいは、または追加的に、追加のメモリ６３０は、たとえば、プロセッサメモリ内などのオフチップであり得る。先に論じたように、重み６０６は、ニューラルネットワークの層に対応する。様々な実装において、重み６０６は、層に従って分割されることが可能であり、よって、最初のｎ個の層の重み６０６は、入力データ６５０が受信される前にメモリサブシステム６０４に記憶され、残りのｍ個の層の重み６０６は、追加のメモリ６３０に記憶される。

【００７５】

[0091]ニューラルネットワーク処理エンジン６０２が入力データ６５０の処理を開始すると、状態６０８の値を記憶するために必要なメモリの量が減る可能性がある。さらに、一部のニューラルネットワークでは、重み６０６の中には、一度使用されるとその後は不要になり得るものがある。たとえば、先に論じたように、フィードフォワード重みは、ある層の入力値の計算に一度しか使用されないことがある。重みの中には、一度使用されると二度と必要とされないものがあることから、入力データ６５０のセットを処理する過程で、これらの重みを記憶するために使用されるメモリは、別の目的で使用されることが可能である。

【００７６】

[0092]二度と必要とされない状態６０８の値および重み６０６を記憶するために先に使用されたメモリバンク６１４内のスペースは、追加の重み６０６の値を記憶するように別の目的で使われることが可能である。図６Ｂに示すように、計算が進むにつれて、より多くのメモリバンク６１４が重み６０６を記憶するように指定され得る。

【００７７】

[0093]たとえば、あらゆるクロックサイクル上で、重み６０６のセットは、様々なメモ

リバンク６１４から読み取られることが可能であって、中間結果は、状態６０８の値として記憶され得る。メモリサブシステム６０４は、制御論理を用いて、まだ使用されたことのない重み６０６により使用されるメモリと、状態６０８の値によってその時点で占有されているスペースの量とを把握することができる。あるいは、または追加的に、ソフトウェアは、メモリサブシステム６０４におけるスペースの割当てを管理することができる。状態６０８を記憶するために必要なメモリ空間の方が少ないことから、メモリサブシステム６０４（またはソフトウェア）は、重み６０６を記憶するためにより多くのメモリ空間を割り当てることを決定し得る。さらに、様々な時点で、メモリサブシステム６０４はチップ相互接続６２０を介して、追加のメモリ６３０からより多くの重みが読み取られるように要求を送信することができる。１つの入力データセットを処理する過程で、ニューラルネットワーク処理エンジン６０２は、ニューラルネットワークの重み６０６の全てをメモリバンク６１４へ徐々に移動することができる。

【００７８】

[0094]実装によっては、１つの入力データセットの処理が完了に近づくにつれて、いまだ必要とされる重み６０６の数は、さらに少なくなっていく可能性がある。残りの重み６０６の数が減少するにつれて、重み６０６または状態６０８へのメモリの割当ては、逆転される可能性があり、よって、徐々に、状態６０８を記憶するためにより多くのメモリが蓄えられていく。計算が終わるまでには、メモリを、図６Ａに示すように再び分割することができる。

【００７９】

[0095]先に述べたように、実装によっては、重み６０６は、一度使用されると、メモリサブシステム６０４から削除され得、この場合、これらの重み６０６によって占有されていたメモリは、依然として必要とされる重み６０６を記憶することに引き渡され得る。これらの実装では、入力データセットの処理が終わりに近づくにつれて、初期の層からの重み６０６は、メモリサブシステム６０４へリロードされることが可能である。したがって、入力データの処理が完了すると、ニューラルネットワーク処理エンジン６０２は、次の入力データを処理できる態勢となり得る。

【００８０】

[0096]メモリサブシステム６０４に現行で存在しない重み６０６が、プロセッサメモリにある場合、プロセッサメモリから重み６０６を読み取ると、処理エンジンアレイ６１０が、重み６０６がロードされるのを待つ間にストールされる可能性がある。様々な実装において、プロセッサメモリからの読取りに由来するストールを回避するための１つの技法は、プロセッサメモリに対する読取りトランザクションを、重み６０６が必要とされる遙か前にキューイングするというものである。別の技法は、重み６０６をオンチップキャッシュにプリフェッチする。別の技法は、別のニューラルネットワーク処理エンジンのメモリを用いることであり、これについては、後に詳述する。様々な実装では、所与のシステムの要件に合わせて、これらの、および他の技法を組み合わせることができる。

【００８１】

[0097]様々な実装では、ニューラルネットワーク処理エンジンのメモリサブシステムにおいて、計算の過程で利用可能となるスペースは、別のニューラルネットワークの重みを記憶するために使用されることが可能である。この第２のニューラルネットワークは、ニューラルネットワーク処理エンジンにもともとロードされているニューラルネットワークとは異なるタスクを実行するように訓練されている場合もある。ニューラルネットワーク処理エンジンに記憶される重みを、第１のニューラルネットワークの重みから第２のニューラルネットワークの重みへ切り替えることにより、ニューラルネットワーク処理エンジンは、異なるタスクを順次実行することができる。

【００８２】

[0098]ニューラルネットワーク処理エンジンを、異なるニューラルネットワーク間で切り替えて異なるタスクを順次実行するように構成することは、所定の状況において有用であり得る。たとえば、単一のユーザからの入力は、たとえば、自然言語理解、これに続く

機械翻訳など、複数の動作を必要とする可能性がある。ニューラルネットワークは、自然言語理解などの複雑なタスクを実行する能力を有し得るものの、１つのニューラルネットワークが、自然言語理解と共に機械翻訳をも実行することはできない場合がある。したがって、多くの場合、１つのニューラルネットワークを用いて自然言語理解が実行され、かつ別のニューラルネットワークを用いて機械翻訳が実行されることが可能である。

【００８３】

[0099]ニューラルネットワークプロセッサが様々な（おそらくは無関係の）タスクに対する結果を生成する速度を最適化するために、ニューラルネットワークプロセッサは、高速コンテキストスイッチを実行するように構成されることが可能である。ニューラルネットワーク処理システムに適用されるコンテキストは、ニューラルネットワークの重みと、ニューラルネットワークが訓練されたタスクを実行するために行われる計算とを含み、１つのニューラルネットワークについて記述する。コンテキストスイッチは、ニューラルネットワーク処理エンジンを、第１のニューラルネットワークを実行するように構成された後に第２のニューラルネットワークを実行するようにプログラムし直すことを包含し得る。ニューラルネットワーク処理エンジンをプログラミムし直すことは、第２のニューラルネットワークの重みをニューラルネットワーク処理エンジンのメモリにロードすることを含み得る。

【００８４】

[0100]高速コンテキストスイッチは、ニューラルネットワーク処理エンジンが、第１のニューラルネットワークを用いる計算を完了すると即時、第２のニューラルネットワークを実行できる態勢になる場合に発生し得る。即時とは、たとえば１クロックサイクルまたは数クロックサイクル内を意味し得る。様々な実装では、第１のニューラルネットワークの計算が進行し、ニューラルネットワークプロセッサのメモリにおいてメモリ空間が利用可能になるにつれて、第２のニューラルネットワークの重みをこの利用可能空間に記憶することができる。最初の計算が完了すると、第２のニューラルネットワークの重みの一部または全てがメモリに存在し、これらの重みを用いる計算を開始することができる。

【００８５】

[0101]図７Ａ～図７Ｃは、ニューラルネットワークを実行するために使用可能なニューラルネットワーク処理エンジンの一例７０２を示す。様々な実装において、ニューラルネットワーク処理エンジン７０２は、メモリサブシステム７０４と、処理エンジンアレイ７１０とを含み得る。動作中（たとえば、入力データセットの結果を計算しているとき）、処理エンジンアレイ７１０は、メモリサブシステム７０４から重み７０６および状態７０８の値を読み取ることができ、ここで、状態７０８の値は、当初、入力データを含み得る。図示されていないが、処理エンジンアレイ７１０は、処理エンジンアレイを包含し得、これらの処理エンジンの各々が、たとえば積和演算を実行することができる。処理エンジンアレイ７１０は、計算結果を結果バッファ７１２へ出力することができる。事例によっては、この例示的なニューラルネットワーク処理エンジン７０２は、結果がメモリサブシステム７０４へ書き込まれる前に、計算結果に対して、活性化ブロック７１６を用いる活性化関数、および／またはプーリングブロック７１８を用いるプーリングを実行することができる。

【００８６】

[0102]動作において、ニューラルネットワーク処理エンジン７０２は、メモリサブシステム７０４のメモリバンク７１４から重み７０６および状態７０８を読み取り、重み７０６および状態７０８を処理エンジンアレイ７１０へ入力する。処理エンジンアレイ７１０は、中間結果を結果バッファ７１２へ出力することができ、中間結果は、結果バッファ７１２からメモリサブシステム７０４へ書き込まれることが可能である。複数の中間結果を組み合わせて１つの出力活性化にする場合は、活性化ブロック７１６が選択的に有効化され得る。また、プーリングブロック７１８も、ノードクラスタの出力を組み合わせて１つの出力にする場合に、選択的に起動され得る。活性化７１６もプーリング７１８も有効化されない場合、処理エンジンアレイ７１０からの結果は、結果バッファからメモリサブシ

ステム７０４へ直に移動されることが可能である。重み７０６および状態７０８を読み取って結果を計算するサイクルは、ニューラルネットワークの最終層に到達して処理エンジンアレイ７１０が最終結果を出力するまで、追加的に生じ得る。事例によっては、最終結果は、メモリサブシステム７０４へ書き込まれる前に、活性化ブロック７１６および／またはプーリングブロック７１８によって作用され得る。

【００８７】

[0103]図７Ａに示すように、様々な実装において、メモリバンク７１４の一部は、重み７０６の値を記憶するために使用され得、異なる部分は、状態７０８の値を記憶するために使用され得る。これらの実装形態では、ニューラルネットワークの重み７０６の値は、ニューラルネットワーク処理エンジン７０２が入力データを受信する前に、重み７０６のために取っておかれたメモリバンク７１４の部分に記憶されることが可能である。事例によっては、メモリバンク７１４に記憶される重み７０６は、ニューラルネットワークの重みの全てを包含することができる。事例によっては、記憶される重み７０６は、ニューラルネットワークの重みの全てより少ない量を包含し得、残りの重みは、（先に論じたように）計算が進行しているときに追加のメモリ７３０から読み取られる。同じくこれらの実装では、計算の過程で、中間結果（たとえば、状態７０８の値）を、メモリバンク７１４の、状態７０８の値を記憶するために予約された部分に記憶することができる。メモリバンク７１４の、重み７０６を記憶するために使用される部分と、状態７０８を記憶するために使用される部分と、任意の未使用部分とは、概して重複しない。

【００８８】

[0104]図７Ａの例において、第１の入力データ７５０が受信されるとメモリサブシステム７０４に記憶される重み７０６は、第１のニューラルネットワークの重み７０６である。重み７０６と、第１のニューラルネットワークが実行するように訓練されたタスク（以後の例では、第１のタスクと称する）と、おそらくは処理エンジンアレイ７１０、活性化ブロック７１６および／またはプーリングブロック７１８に対する命令をも含む第１のニューラルネットワークは、ニューラルネットワーク処理エンジン７０２の第１のコンテキストを表し得る。第１の入力データ７５０がニューラルネットワーク処理エンジン７０２によって受信されると、第１の入力データ７５０に対して第１のタスクが実行される。

【００８９】

[0105]先に論じたように、第１のタスクを実行するための計算が進行するにつれて、メモリサブシステム７０４内のスペースは、利用可能になることもあれば、未使用と見なされることもある。たとえば、メモリバンク７１４の、状態７０８の値を記憶するために使用される部分は、中間結果を記憶するために必要とされるメモリの量が少なくなるにつれて、小さくなる可能性がある。一部の例において、メモリサブシステム７０４における論理は、中間結果を記憶するために必要なメモリの量を決定することができる。例によっては、ソフトウェアが、中間結果を記憶するために必要なメモリの量を決定することができ、したがって、メモリバンク７１４の、状態７０８の値を記憶するために使用されている部分を減らすことができる。さらなる例として、メモリバンク７１４の、第１のニューラルネットワークの重み７０６を記憶するために使用される部分は、使用されたことのある重み７０６の値がもはや不要としてマーキングされることに起因して、より小さくなり得る。例によっては、メモリサブシステム７０４における論理は、使用される重みのアドレス位置を利用可能としてマーキングすることができる。例によっては、ソフトウェアは、いまだ必要とされる重み値と、使用されていて不要となった重み値とを把握することができる。

【００９０】

[0106]様々な実装では、メモリサブシステム７０４におけるスペースが未使用または利用可能となるにつれて、図７Ｂに示すように、第２のニューラルネットワークの重み７０７の値を利用可能なスペースに記憶することができる。図７Ｂに示す例では、先に第１のニューラルネットワークの重み７０６を記憶するために使用された幾つかのメモリバンク７１４、および先に状態７０８の値を記憶するために使用された幾つかのメモリバンク７

１４が、第２のニューラルネットワークの重み７０７の値を記憶するために転用されている。第２のニューラルネットワークは、第１のタスクとは異なり得る、以後第２のタスクと称するタスクを実行するように訓練されている場合がある。重み７０７と、第２のタスクと、おそらくは処理エンジンアレイ７１０、活性化ブロック７１６および／またはプーリングブロック７１８に対するコマンドとは、ニューラルネットワーク処理エンジン７０２の第２のコンテキストと見なされ得る。重み７０７は、チップ相互接続７２０を介して追加のメモリ７３０から読み取られることが可能であり、追加のメモリ７３０は、オンチップキャッシュ、プロセッサメモリおよび／または別のニューラルネットワーク処理エンジン７０２のメモリであってもよい。

【００９１】

[0107]様々な実装において、第２のニューラルネットワークの重み７０７を記憶することは、第１の入力データ７５０の結果を計算する間に発生し得る。すなわち、第１の入力データ７５０の結果の計算が進み、メモリバンク７１４においてスペースが利用可能となるにつれて、ニューラルネットワーク処理エンジン７０２は、未使用となったスペースに第２のニューラルネットワークの重み７０７をロードし始めることができる。第２のニューラルネットワークの重み７０７の記憶は、ニューラルネットワーク処理エンジン７０２による第２の入力データ７５１の受信と同時に開始することができ、この時点で第２のタスクが実行される。事例によっては、第２の入力データ７５１は、ニューラルネットワーク処理エンジン７０２が第１の入力データ７５０の結果を計算する過程で受信される。事例によっては、第２のニューラルネットワークの重み７０７の記憶は、第２の入力データ７５１が受信される前または後で開始することができる。例によっては、第２の入力データ７５１の受信は、第２のニューラルネットワークの重み７０７のロードをトリガする。例によっては、第２の入力データ７５１の受信は、第２の入力データ７５１の結果の計算をトリガする。

【００９２】

[0108]事例によっては、第１のニューラルネットワークの重み７０６を記憶するために使用されるメモリバンク７１４のみが、第２のニューラルネットワークの重み７０７を記憶するために転用される。これらの事例において、メモリバンク７１４の、状態７０８を記憶するために使用される部分は、常に、中間結果を記憶するために利用可能である。

【００９３】

[0109]事例によっては、メモリバンク７１４を転用することは、メモリバンク７１４間でデータを移動することを含み得る。たとえば、メモリバンク７１４の、状態７０８を記憶するために使用される部分が小さくなるにつれて、未使用の重み７０６は、この部分へ移動されることが可能である。この例では、メモリバンク７１４の、先に第１のニューラルネットワークの重み７０６により占有されていた部分は、徐々に第２のニューラルネットワークの重み７０７によって占有されることになり得、よって、コンテキストスイッチが完了すると、メモリの割当ては、第１のニューラルネットワークの場合と同じになる。

【００９４】

[0110]様々な実装において、ニューラルネットワーク処理エンジン７０２が第１の入力データ７５０の結果の計算を完了すると、ニューラルネットワーク処理エンジン７０２は、第２の入力データ７５１の計算を実行できる態勢となり得る。図７Ｃに示す例では、ニューラルネットワーク処理エンジン７０２が、第１のニューラルネットワークから第２のニューラルネットワークへのコンテキストスイッチを完了している。この例に示すように、メモリサブシステム７０４には第２のニューラルネットワークの重み７０７がロードされていて、ニューラルネットワーク処理エンジン７０２は、第２の入力データ７５１の結果の計算を開始できる態勢にある。

【００９５】

[0111]様々な実装において、ニューラルネットワーク処理エンジン７０２は、第１の入力データ７５０の計算を完了する１または２サイクルのうちに、第２の入力データ７５１の結果の計算を開始することができる。実装によっては、ニューラルネットワーク処理エ

ンジン７０２は、第２の入力データ７５１の結果の計算を、第１の入力データ７５０の結果が完了する前に開始することができる。たとえば、第２の入力データ７５１の結果を計算するための状態７０８および重み７０７が処理エンジンアレイ７１０に入力されつつある時点で、第１の入力データ７５０の計算の最終サイクルは、まだ処理エンジンアレイ７１０を通って進行中である場合がある。この例では、第１の入力データ７５０の計算および第２の入力データ７５１の計算が処理エンジンアレイ７１０内に同時に存在する場合がある。

【００９６】

[0112]事例によっては、いまやメモリサブシステム７０４に記憶されている重み７０７は、第２のニューラルネットワークの全ての重みを包含する。事例によっては、重み７０７の数は、第２のニューラルネットワークの重みの全てより少なく、残りの重みは、メモリバンク７１４においてスペースが利用可能となるにつれて、追加のメモリ７３０から読み取られ得る。

【００９７】

[0113]事例によっては、図７Ｃに示す例において、メモリサブシステム７０４内になおも第１のコンテキストに関連する幾つかのデータが存在する場合がある。たとえば、メモリバンク７１４には、第１のニューラルネットワークの、最後に使用されたものなどの幾つかの重み７０６の値がまだ存在している場合がある。この例では、第２の入力データ７５１の計算が始まると、これらの重み７０６は、上書きされることが可能である。別の例として、第１の入力データ７５０の計算の最終結果は、チップ相互接続７２０を介する読み出しを待機して、なおもメモリバンク７１４内に存在する場合がある。最終結果は、第２の入力データ７５１の計算が進行している間、メモリサブシステム７０４にとどまる場合もあれば、第２の入力データ７５１の計算が始まる前に読み出される場合もある。

【００９８】

[0114]ニューラルネットワーク処理エンジン７０２が第１のニューラルネットワークの重み７０６値を用いて結果を計算している間に、第２のニューラルネットワークの重み７０７の値をロードすると、ニューラルネットワーク処理エンジン７０２が第１のニューラルネットワークと第２のニューラルネットワークとの間で高速コンテキストスイッチを実行することが有効化される。高速コンテキストスイッチは、ニューラルネットワーク処理エンジン７０２がスループットを高めかつ結果をより速く生成することを有効化し得る。

【００９９】

[0115]ニューラルネットワークプロセッサを介してスループットを高めるための別の技法は、ニューラルネットワークプロセッサに、並行して動作できる複数のニューラルネットワーク処理エンジンを装備するというものである。ニューラルネットワークプロセッサは、たとえば、異なるニューラルネットワーク処理エンジン上で、複数の、おそらくは異なるニューラルネットワークを同時に実行することができる。別の例として、ニューラルネットワークプロセッサは、より多くの入力データセットをバッチ処理することができ、よって重み値の再使用が増加する。

【０１００】

[0116]図８は、複数のニューラルネットワーク処理エンジン８０２ａ～８０２ｎを有するニューラルネットワークプロセッサの一例８００を示す。ニューラルネットワーク処理エンジン８０２ａ～８０２ｎは、各々、メモリサブシステムと、処理エンジンアレイとを含むことができ、かつニューラルネットワークが、そのためにニューラルネットワークがプログラムされたタスクを実行するために必要な計算を実行することができる。図示の例では、ニューラルネットワークプロセッサ８００がｎ個のニューラルネットワーク処理エンジン８０２ａ～８０２ｎを含む。

【０１０１】

[0117]この例示的なニューラルネットワークプロセッサ８００は、さらに、この例ではＤＲＡＭ８３０を用いて実装されるプロセッサメモリと通信するためのＤＲＡＭコントローラ８４２ａ～８４２ｋを含む。図示の例において、ニューラルネットワークプロセッサ

８００は、ｋ個のＤＲＡＭコントローラ８４２ａ～８４２ｋを含み、その各々が、独立したＤＲＡＭバンクセットと通信することができてもよい。他の例では、プロセッサメモリに他のタイプのＲＡＭ技術を使用可能である。ＤＲＡＭコントローラ８４２ａ～８４２ｋは、メモリコントローラと呼ばれることもある。

【０１０２】

[0118]例示的なニューラルネットワークプロセッサ８００は、さらに、システム内のＰＣＩデバイス８３２と通信するためのペリフェラル・コンポーネント・インターコネクト（ＰＣＩ）コントローラ８４４ａ～８４４ｐを含む。本例において、ＰＣＩは、プロセッサバスの実装に使用されているバスプロトコルである。プロセッサバスは、ニューラルネットワークプロセッサ８００を、たとえば、とりわけＩ／Ｏデバイス、メモリコントローラ、ストレージデバイスおよび／またはネットワーク・インタフェース・カードなどのＰＣＩデバイス８３２へ接続することができる。本例において、ニューラルネットワークプロセッサ８００は、ｐ個のＰＣＩコントローラ８４４ａ～８４４ｐを含み、その各々が、別々のルートコンプレックスを含んでもよく、かつＰＣＩデバイス８３２の別々のセットと通信してもよい。他の例では、ウルトラ・パス・インターコネクト（ＵＰＩ）などの他の標準化されたバスプロトコルをホストバスに使用することができる。他の例では、独自仕様のバスプロトコルを使用することができる。

【０１０３】

[0119]この例示的なニューラルネットワークプロセッサ８００は、さらに、ニューラルネットワーク処理エンジン８０２ａ～８０２ｎと、ＤＲＡＭコントローラ８４２ａ～８４２ｋと、ＰＣＩコントローラ８４４ａ～８４４ｐとの間でデータを移動させることができるＤＭＡエンジン８４６ａ～８４６ｄを含む。図示の例では、ニューラルネットワークプロセッサ８００がｄ個のＤＭＡエンジン８４６ａ～８４６ｄを含む。実装によっては、ＤＭＡエンジン８４６ａ～８４６ｄは、ＤＲＡＭコントローラ８４２ａ～８４２ｋからニューラルネットワーク処理エンジン８０２ａ～８０２ｎへのデータの移動、またはＰＣＩコントローラ８４４ａ～８４４ｐとニューラルネットワーク処理エンジン８０２ａ～８０２ｎとの間のデータの移動などの特定のタスクに割り当てられ得る。実装によっては、少なくとも１つのＤＭＡエンジン８４６ａ～８４６ｄを、各ニューラルネットワーク処理エンジン８０２ａ～８０２ｎの専用とすることができる。実装によっては、ＤＭＡエンジン８４６ａ～８４６ｄは、１つの機能またはコンポーネントの専用とされる代わりに、プールとして扱われてもよく、よって、データを移動する必要がある度に、利用可能なＤＭＡエンジン８４６ａ～８４６ｄが使用される。

【０１０４】

[0120]例示的なニューラルネットワークプロセッサ８００では、様々なコンポーネントがチップ相互接続８２０を介して通信することができる。チップ相互接続８２０は、主として、ニューラルネットワークプロセッサ８００のコンポーネント間でデータをルーティングするための配線を含む。事例によっては、チップ相互接続８２０は、データの方向を制御するためのマルチプレクサ、クロックドメイン交差を処理するためのフリップフロップ、およびタイミング論理などの、最小限の論理を包含することができる。

【０１０５】

[0121]図９は、ニューラルネットワークの重み９０６が２つのニューラルネットワーク処理エンジン９０２ａ～９０２ｂのメモリサブシステムに記憶されるように構成されている、ニューラルネットワークプロセッサの一例９００を示す。ニューラルネットワークプロセッサ９００は、図８に示すニューラルネットワークプロセッサ８００に類似するものであり得る。図９に示すように、ニューラルネットワーク処理エンジン９０２ａ～９０２ｎは、各々、メモリサブシステム９０４ａ～９０４ｎと、処理エンジンアレイ９１０ａ～９１０ｎとを含む。様々な実装において、メモリサブシステム９０４ａ～９０４ｎは、各々、ニューラルネットワークの重みを記憶することができ、よって、ニューラルネットワーク処理エンジン９０２ａ～９０２ｎは、各々、独立してニューラルネットワークを実行することができる。

【０１０６】
[0122]事例によっては、第２のニューラルネットワーク処理エンジン９０２ｂなどのニューラルネットワーク処理エンジン９０２ａ～９０２ｎのうちの１つは、ニューラルネットワーク処理エンジン９０２ｂのメモリサブシステム９０４ｂにおけるスペースの全てを必要とはしない場合がある。たとえば、ニューラルネットワーク処理エンジン９０２ｂがニューラルネットワークの重みを記憶するために必要とするスペースは、メモリサブシステム９０４ｂにおける全てのスペースに満たない場合がある。あるいは、別の例として、ニューラルネットワーク処理エンジン９０２ｂは、重み値をＤＲＡＭ９３０のみから、または主としてＤＲＡＭ９３０から入手するように指定されていてもよい（たとえば、ニューラルネットワーク処理エンジン９０２ｂが実行しているニューラルネットワークは、高度の重み再使用を有する）。
【０１０７】
[0123]様々な実装において、第２のニューラルネットワーク処理エンジン９０２ｂ内の利用可能なメモリは、第１のニューラルネットワーク処理エンジン９０２ａにより実行されているニューラルネットワークの重み９０６を記憶するために使用されることが可能である。ニューラルネットワークの重み９０６は、たとえば、第１のニューラルネットワーク処理エンジン９０２ａのメモリサブシステム９０４ａ内に完全に記憶するには多すぎる場合がある。先に述べたように、第１のニューラルネットワーク処理エンジン９０２ａに収まらない重み９０６は、ＤＲＡＭ９３０に記憶されることが可能である。しかしながら、ＤＲＡＭ９３０から重み９０６を読み取ることより、１つのニューラルネットワーク処理エンジン９０２ｂから別のニューラルネットワーク処理エンジン９０２ａへ重み９０６をコピーすることの方が遙かに高速で実行可能であることから、これらの重みを第２のニューラルネットワーク処理エンジン９０２ｂの利用可能なスペースに記憶することの方が遙かに効率的であり得る。
【０１０８】
[0124]先に論じたように、重み９０６は、第１のニューラルネットワーク処理エンジン９０２ａが入力データセットを処理するにつれて、第２のニューラルネットワーク処理エンジン９０２ｂから第１のニューラルネットワーク処理エンジン９０２ａへコピーされることが可能である。事例によっては、使用されて不要となった重み９０６は、入力データの処理が完了するまで、第１のニューラルネットワーク処理エンジン９０２ａから第２のニューラルネットワーク処理エンジン９０２ｂへ移動されてもよい。様々な実装において、ＤＭＡエンジン９４６ａ～９４６ｄは、ニューラルネットワーク処理エンジン９０２ａ～９０２ｎ間でデータをコピーまたは移動させることができ、よって、ニューラルネットワーク処理エンジン９０２ａ～９０２ｎは、ニューラルネットワークの計算の実行に専念することができる。
【０１０９】
[0125]第２のニューラルネットワーク処理エンジン９０２ｂのメモリサブシステム９０４ｂにおける、第１のニューラルネットワーク処理エンジン９０２ａの重み９０６を記憶するために使用されないスペースはいずれも、ニューラルネットワーク処理エンジン９０２ｂがその固有の計算を実行するために使用されることが可能である。
【０１１０】
[0126]実装によっては、ニューラルネットワーク処理エンジン１００２ａ～１００２ｎ間で重みを移動させるのではなく、代わりに進行中の計算を移動させることができる。図１０は、ニューラルネットワークの重み１００６が２つのニューラルネットワーク処理エンジン１００２ａ～１００２ｂのメモリサブシステムに記憶されるように構成されている、ニューラルネットワークプロセッサの一例１０００を示す。ニューラルネットワークプロセッサ１０００は、図８に示すニューラルネットワークプロセッサに類似するものであり得る。図９の例の場合のように、図１０の例において、ニューラルネットワークの重み１００６は、おそらくは、重み１００６の全てをオンチップで記憶するために必要なメモリ量が、ニューラルネットワーク処理エンジン１００２ａ～１００２ｎのどのメモリ量を

も超えるという理由で、２つのニューラルネットワーク処理エンジン１００２ａ～１００２ｂのメモリサブシステム１００４ａ～１００４ｂに記憶される。

【０１１１】

[0127]先に述べたように、重み１００６は、ニューラルネットワークの特定の層に関連づけられ得る。したがって、第１の層セットに関する重み１００６は、第１のニューラルネットワーク処理エンジン１００２ａに記憶されることが可能であり、残りの層に関する重み１００６は、ニューラルネットワーク処理エンジン１００２ｂに記憶されることが可能である。

【０１１２】

[0128]様々な実装において、重み１００６に関連づけられるニューラルネットワークの入力データが受信されると、入力データは、第１のニューラルネットワーク処理エンジン１００２ａがニューラルネットワークの初期層の重みを有することから、第１のニューラルネットワーク処理エンジン１００２ａへ提供され得る。第１のニューラルネットワーク処理エンジン１００２ａは、入力データの処理を進めることができ、メモリサブシステム１００４ａから重み１００６を読み取り、処理エンジンアレイ１０１０ａで中間結果を計算しかつ中間結果をメモリサブシステム１００４ａに書き戻す。

【０１１３】

[0129]第１のニューラルネットワーク処理エンジン１００２ａが、第１のニューラルネットワーク処理エンジン１００２ａがその重みを有する各層についての計算を実行すると、第１のニューラルネットワーク処理エンジン１００２ａは、進行中の計算を第２のニューラルネットワーク処理エンジン１００２ｂへ移動させることができる。進行中の計算を移動させることは、状態１００８の値を移動させることを含み得る。状態１００８は、たとえば、中間結果のセットを含み得る。例によっては、状態１００８は、とりわけ、処理された最終層の識別、処理エンジンアレイ１０１０ｂの制御入力、活性化関数の状態値、および／またはプーリング関数の状態値をさらに含み得る。様々な実装において、ＤＭＡエンジン１０４６ａ～１０４６ｄは、状態１００８の値を第１のニューラルネットワーク処理エンジン１００２ａから第２のニューラルネットワーク処理エンジン１００２ｂへコピーすることに従事され得る。

【０１１４】

[0130]様々な実装において、状態１００８が第２のニューラルネットワーク処理エンジン１００２ｂへコピーされると、ニューラルネットワーク処理エンジン１００２ｂは、進行中の計算を再開するための信号を受信することができる。信号は、たとえば、とりわけ、割込み、第２のニューラルネットワーク処理エンジン１００２ｂ内のレジスタへの書込み、または制御バスを介してニューラルネットワーク処理エンジン１００２ａ～１００２ｎ間で送信される命令、の形式をとることができる。事例によっては、第２のニューラルネットワーク処理エンジン１００２ｂは、自動的に計算を開始することができる。たとえば、中間結果は、入力データと同様にして入力されることが可能であり、よって、入力データに対して起動されるはずの第２のニューラルネットワーク処理エンジン１００２ｂ内の同じメカニズムが起動される。

【０１１５】

[0131]様々な実装において、進行中の計算を再開することは、中間層からの重み１００６を、第１のニューラルネットワーク処理エンジン１００２ａからの中間結果と共に処理エンジンアレイ１０１０ｂへ入力することを含み得る。様々な実装において、ニューラルネットワーク処理エンジン１００２ｂは、ニューラルネットワーク処理エンジン１００２ｂが進行中の計算を新しい計算の開始として扱い得ること、または、ニューラルネットワーク処理エンジン１００２ｂは、第１のニューラルネットワーク処理エンジン１００２ａがやめた場所を正確に捕捉できること、のいずれかを理由に、進行中の計算を再開するための特別な、または異なる挙動を必要としない。

【０１１６】

[0132]図９および１０の例では、ニューラルネットワークの全ての重みが処理エンジン

アレイのローカルメモリにおいて保持され得なくなるのはいつであるか、に対するソリューションが提供される。ニューラルネットワーク処理エンジン間でデータを移動する際には、多少の待ち時間が発生し得るが、それでもこの待ち時間は、オフチップメモリから重みを読み取る際に発生する待ち時間より遙かに短いものである。

【０１１７】

[0133]図１１は、重みの全てを単一の処理エンジンアレイに対してローカルに記憶し得ない場合に、ニューラルネットワークの全ての重みをオンチップで記憶するための別のソリューションを示す。図１１は、２つのニューラルネットワークプロセッサ１１０１ａ～１１０１ｂを含むニューラルネットワーク処理システムの一例１１００を示す。本例において、各ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、別々のチップであって、別々のダイを有し、かつ異なるパッケージに封入されている。ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、ホストバスまたはプロセッサバスを用いて、互いに、かつニューラルネットワーク処理システム１１００内の他のコンポーネントと通信することができる。プロセッサバスは、たとえば、ＰＣＩベースのプロトコルを用いて実装されることが可能である。

【０１１８】

[0134]様々な実装において、ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、各々、少なくとも１つの、おそらくはそれ以上のニューラルネットワーク処理エンジン１１０２ａ～１１０２ｂを含む。先に論じたように、ニューラルネットワーク処理エンジン１１０２ａ～１１０２ｂは、各々、処理エンジンアレイ１０１０ａ～１０１０ｂと、メモリサブシステム１１０４ａ～１１０４ｂとを含むことができる。

【０１１９】

[0135]ニューラルネットワーク処理エンジン１１０２ａ～１１０２ｂと、各ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂ内の他のコンポーネントとの間の通信のために、ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、チップ相互接続１０２０ａ～１０２０ｂを含むことができる。チップ相互接続１０２０ａ～１０２０ｂは、主として、ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂのコンポーネント間で信号をルーティングするための配線、およびおそらくは何らかのタイミング論理も含むことができる。

【０１２０】

[0136]ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂの他のコンポーネントは、プロセッサバスと通信するためのＰＣＩコントローラ１１４４ａ～１１４４ｂを含むことができる。様々な実装において、ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、ＤＲＡＭおよびＤＭＡコントローラなどの図示されていないコンポーネントを含むことができる。

【０１２１】

[0137]図１１の例では、ニューラルネットワークの重み１１０６を２つに分割することができ、２つの部分は、同じサイズまたは異なるサイズである。ニューラルネットワークの初期層に対応する重み１１０６の第１のセットは、第１のニューラルネットワークプロセッサ１１０１ａにおけるニューラルネットワーク処理エンジン１１０２ａのメモリサブシステム１１０４ａにロードされることが可能である。重み１１０６の第２のセットは、第２のニューラルネットワークプロセッサ１１０１ｂにおけるニューラルネットワーク処理エンジン１１０２ｂのメモリサブシステム１１０４ｂにロードされることが可能である。この構成では、ニューラルネットワークの入力データが受信されると、入力データは、第１のニューラルネットワークプロセッサ１１０１ａのニューラルネットワーク処理エンジン１１０２ａへ提供され得る。ニューラルネットワーク処理エンジン１１０２ａは、入力データの計算を進めることができる。

【０１２２】

[0138]ニューラルネットワーク処理エンジン１１０２ａが、ニューラルネットワーク処理エンジン１１０２ａがその重み１１０６を有する層の各々についての計算を実行すると

、ニューラルネットワーク処理エンジン１１０２ａは、進行中の計算を第２のニューラルネットワークプロセッサ１１０１ｂにおけるニューラルネットワーク処理エンジン１１０２ｂへ移動させることができる。たとえば、ニューラルネットワーク処理エンジン１１０２ａは、ＤＭＡエンジンに、計算の状態１１０８をＰＣＩコントローラ１１４４ａへコピーするように命令することができるが、状態１１０８は、とりわけ、中間結果を含み得る。本例において、ＰＣＩコントローラ１１４４ａは、第２のニューラルネットワークプロセッサ１１０１ｂへの１つまたは複数の書込みを生成することができ、書き込みは状態１１０８の値を含む。実装によっては、第１のニューラルネットワークプロセッサ１１０１ａは、ニューラルネットワーク処理エンジン１１０２ｂのメモリサブシステム１１０４ｂへ直に書き込むことができる。実装によっては、状態１１０８のデータは、第２のニューラルネットワークプロセッサ１１０１ｂ内のＰＣＩコントローラ１１４４ｂによってキャッシュされる。これらの実装において、ＤＭＡエンジンは、たとえば、状態１１０８のデータをＰＣＩコントローラ１１４４ｂからニューラルネットワーク処理エンジン１１０２ｂへ移動することができる。

【０１２３】

[0139]状態１１０８が第２のニューラルネットワークプロセッサ１１０１ｂ内のニューラルネットワーク処理エンジン１１０２ｂのメモリサブシステム１１０４ｂへロードされると、ニューラルネットワーク処理エンジン１１０２ｂは、進行中の計算を再開することができる。例によっては、第１のニューラルネットワークプロセッサ１１０１ａは、第２のニューラルネットワークプロセッサ１１０１ｂへの割込みを生成することにより、または第２のニューラルネットワークプロセッサ１１０１ｂ内のレジスタへ値を書き込むことにより、第２のニューラルネットワーク処理エンジン１１０２ｂをトリガする。ニューラルネットワーク処理エンジン１１０２ｂが最終結果を計算すると、第２のニューラルネットワークプロセッサ１１０１ｂは、結果を出力することができる。

【０１２４】

[0140]図１１の例において、ニューラルネットワークの重み１１０６は全て、処理エンジンアレイ１０１０ａ～１０１０ｂに対してローカルに記憶され、よって、ニューラルネットワークプロセッサ１１０１ａ～１１０１ｂは、重み１１０６を入手するためにプロセッサメモリを読み取る必要を回避することができる。状態１１０８をニューラルネットワークプロセッサ１１０１ａ～１１０１ｂ間で転送する際には、多少の遅延が生じるものの、この遅延は、プロセッサメモリへの複数回の読取りによって引き起こされる遅延より遙かに少ないものであり得る。

【０１２５】

[0141]様々な実装において、図１０および図１１に示す例は、様々な方法で組み合わされ得る。たとえば、入力データセットの処理は、１つのニューラルネットワーク処理エンジンにおいて開始されることが可能であって、同じニューラルネットワークプロセッサにおける第２のニューラルネットワーク処理エンジンへ転送されてもよい。進行中の計算は、次に、第２のニューラルネットワーク処理エンジンから第３のニューラルネットワーク処理エンジンへ転送されることが可能であり、ここで、第３のニューラルネットワーク処理エンジンは、異なるニューラルネットワークプロセッサ内にある。ニューラルネットワークの全ての重みをオンチップで記憶し、かつプロセッサメモリからの読取りを最小限にする、またはなくすことができるような、様々な構成が可能である。

【０１２６】

[0142]図１２は、ニューラルネットワークプロセッサがニューラルネットワークを実行している場合にニューラルネットワークプロセッサにより実行されることが可能なプロセスの一例１２００を示す。

【０１２７】

[0143]ステップ１２０２において、プロセス１２００は、重み値セットをニューラルネットワーク処理回路の複数のメモリバンクに記憶することを含み、ここで、ニューラルネットワーク処理回路は、複数のメモリバンクと同じダイ上に処理エンジンアレイを含み、

重み値セットは、入力データの受信に先行して記憶される。様々な実装において、ニューラルネットワーク処理回路は、ニューラルネットワークプロセッサの一部である。例によっては、重み値セットは、ニューラルネットワークの全ての重み値を含む。

【０１２８】

[0144]ステップ１２０４において、プロセス１２００は、入力データを受信することを含む。様々な例において、入力データは、画像、オーディオ信号、テキスト文字列、数列または他の何らかの入力であってもよい。

【０１２９】

[0145]ステップ１２０６において、プロセス１２００は、重み値セットを用いて、ニューラルネットワーク用に規定されたタスクを実行することを含む。タスクは、たとえば、画像識別、音声認識、自然言語処理、ゲームプレイ、他を含んでもよい。重み値は、訓練を介して先に決定されている可能性があり、訓練では、たとえば、その出力結果が既知である入力データがニューラルネットワークに入力されている。

【０１３０】

[0146]ステップ１２０６は、タスクを実行するためのさらなるステップを含み得る。ステップ１２０８において、タスクを実行することは、複数のメモリバンクから重み値を読み取ることを含む。例によっては、重み値を読み取ることは、同時に、複数のメモリバンクのうちの第１のメモリバンクから第１の重み値を読み取り、かつ複数のメモリバンクのうちの第２のメモリバンクから第２の重み値を読み取ることを含む。

【０１３１】

[0147]ステップ１２１０において、タスクを実行することは、さらに、重み値および入力データを処理エンジンアレイへ入力することを含み得る。例によっては、処理エンジンアレイは、シストリックアレイである。例によっては、ステップ１２１０は、さらに、処理エンジンアレイを用いて中間結果を決定し、かつ中間結果を複数のメモリバンクにおける１つのメモリバンクに記憶することを含み得る。例によっては、中間結果は、追加の重み値が読み取られると同時に書き込まれてもよい。

【０１３２】

[0148]ステップ１２１２において、タスクを実行することは、さらに、処理エンジンアレイを用いて結果を計算することを含み得、結果は、タスク実行の結果に対応する。

【０１３３】

[0149]実装によっては、ステップ１２０２において記憶される重み値セットは、ニューラルネットワークの全ての重み値の第１の部分を含む。これらの実装において、プロセス１２００は、さらに、複数のメモリバンクが利用可能なスペースを有すると決定することを含み得る。使用可能なスペースは、より少ないスペースを要求する中間結果、および／または不要となった重み値に由来する場合がある。プロセス１２００は、さらに、ニューラルネットワークの全ての重み値の第２の部分を読み取ることを含み得、第２の部分は、追加のメモリから読み取られる。例によっては、追加のメモリは、同じダイ上の第２の処理エンジンアレイに関連づけられ得る。例によっては、追加のメモリは、オフチップであってもよい。プロセス１２００は、さらに、第２の部分を利用可能なスペースへ書き込むことを含み得る。重みの第２の部分は、次に、タスクの実行を継続するために使用され得る。

【０１３４】

[0150]図１３は、ニューラルネットワークプロセッサがニューラルネットワークを実行している場合にニューラルネットワークプロセッサにより実行されることが可能なプロセスの一例１３００を示す。

【０１３５】

[0151]ステップ１３０２において、プロセス１３００は、第１の重み値セットをニューラルネットワーク処理システムの第１のメモリバンクセットに記憶することを含む。

【０１３６】

[0152]ステップ１３０４において、プロセス１３００は、第２の重み値セットをニュー

ラルネットワーク処理システムの第２のメモリバンクセットに記憶することを含み、第１の重み値セットおよび第２の重み値セットは、ニューラルネットワークの全ての重み値を含み、かつ第１の重み値セットおよび第２の重み値セットは、ニューラルネットワーク処理システムによる入力データの受信に先行して記憶される。

【０１３７】

[0153]ステップ１３０６において、プロセス１３００は、入力データを受信することを含む。様々な例において、入力データは、画像、オーディオ信号、テキスト文字列、数列または他の何らかの入力であってもよい。

【０１３８】

[0154]ステップ１３０８において、プロセス１３００は、ニューラルネットワーク用に規定されたタスクを実行することを含む。様々な例において、タスクは、入力データのタイプ（たとえば、入力画像の画像認識、オーディオ信号の音声認識、他）に相関する。

【０１３９】

[0155]ステップ１３０８は、タスクを実行するためのさらなるステップを含み得る。ステップ１３１０において、タスクを実行することは、第１の処理エンジンアレイを用いて中間結果を計算することを含み、第１の処理エンジンアレイは、第１のメモリバンクセットと同じ第１のダイ上に存在し、第１の処理エンジンアレイは、第１の重み値セットおよび入力データを用いて中間結果を計算する。中間結果を計算することは、第１のメモリバンクセットから２つ以上の重み値を同時に読み取ることを含み得る。例によっては、中間結果は、ニューラルネットワークの１つまたは複数の初期層の出力を反映する。

【０１４０】

[0156]ステップ１３１２において、タスクを実行することは、さらに、中間結果および第１のメモリバンクセットからの状態を第２のメモリバンクセットへコピーすることを含み、状態は、第１の処理エンジンアレイにより完了されるタスクのステージを記述する。例によっては、中間結果および状態をコピーできる態勢になると、第１の処理アレイは、入力データに対する演算を完了していて、新しい入力データに対する演算を開始することができる。

【０１４１】

[0157]ステップ１３１４において、タスクを実行することは、さらに、第２の処理エンジンアレイを用いて最終結果を計算することを含み、第２の処理エンジンアレイは、第２のメモリバンクセットと同じ第２のダイ上に存在し、第２の処理エンジンアレイは、第２の重み値セット、中間結果および状態を用いて最終結果を計算し、最終結果は、タスク実行の結果に対応する。最終結果を計算することは、第２のメモリバンクセットから２つ以上の重み値を同時に読み取ることを含み得る。例によっては、最終結果は、ニューラルネットワークの１つまたは複数の最終層の出力を反映する。

【０１４２】

[0158]例によっては、第１のメモリバンクセットおよび第１の処理エンジンアレイを含む第１のダイと、第２のメモリバンクセットおよび第２の処理エンジンアレイを含む第２のダイとは、同じダイの部分である。該ダイは、ニューラルネットワークプロセッサ用であってもよく、ニューラルネットワークプロセッサは、２つ以上のニューラルネットワーク処理エンジンを含む（たとえば、ニューラルネットワーク処理エンジンは、メモリバンクセットおよび処理エンジンアレイを含む）。例によっては、ニューラルネットワークプロセッサは、内部通信ファブリックを含んでもよく、中間結果および状態は、内部通信ファブリックを介してコピーされ得る。

【０１４３】

[0159]例によっては、第１のダイおよび第２のダイは、各々、異なるニューラルネットワークプロセッサの一部であり、よって、異なるパッケージ内に存在する。これらの例において、２つのニューラルネットワークプロセッサは、ホストバスにより、中間結果および状態の一方から他方へのコピーを含む相互通信を可能にされ得る。

【０１４４】

[0160]ステップ１３０４において述べたように、第１の重み値セットおよび第２の重み値セットは、ニューラルネットワークの全ての重み値であり得る。これは、重み値がプロセッサメモリからは読み取られないことを意味し得る。プロセッサメモリは、第１のダイまたは第２のダイとは異なるパッケージ内に存在し、よって、読取り遅延が大きくなる場合がある。

【０１４５】

[0161]図１４は、ニューラルネットワークプロセッサを含むコンピューティングシステムにより実行されることが可能なプロセスの一例１４００を示す。

【０１４６】

[0162]ステップ１４０２において、プロセス１４００は、第１のニューラルネットワークの第１の重み値セットを記憶することを含み、第１のニューラルネットワークは、第１のタスクを実行するように構成され、第１の重み値セットは、ニューラルネットワークプロセッサにおけるメモリの第１の部分に記憶される。第１の重み値セットおよび第１のタスクは、最初のコンテキストと見なされ得る。

【０１４７】

[0163]ステップ１４０４において、プロセス１４００は、第１のタスクに関連づけられる第１の入力データを受信することを含み、第１の入力データは、第１の重み値セットの記憶後に受信される。第１のタスクは、第１の入力データを入力として用いて実行される。様々な実装において、第１の重み値セットは、入力データが受信される前にメモリに記憶される。

【０１４８】

[0164]ステップ１４０６において、プロセス１４００は、ニューラルネットワークプロセッサを用いて第１の結果を計算することを含み、第１の結果を計算することは、第１の重み値セットおよび第１の入力データを用いることを含み、第１の結果は、第１のタスクの実行の結果に対応し、かつ第１の結果を計算する間に、中間結果がメモリの第２の部分に記憶される。メモリの第１の部分および第２の部分は、概して重ならない。

【０１４９】

[0165]ステップ１４０８において、プロセス１４００は、第１の結果を計算する間に、メモリの第３の部分を識別することを含み、第３の部分は、メモリ内に未使用のスペースを含む。未使用のスペースには、以前は第１の部分（たとえば、第１の重み値セットを記憶するために使用される部分）の一部であったがもはやそうではないスペース、および／または以前は第２の部分（たとえば、中間結果を記憶するために使用される部分）の一部であったがもはやそうではないスペースが含まれ得る。第３の部分は、概して、第１の部分または第２の部分と重複しない。

【０１５０】

[0166]ステップ１４１０において、プロセス１４００は、第２の重み値セットを第３の部分に記憶することを含み、第２の重み値セットは、第２のニューラルネットワークに対するものであり、第２のニューラルネットワークは、第２のタスクを実行するように構成される。事例によっては、第２のタスクは、第１のタスクとは異なる。第２の重み値セットは、第１の結果の計算が完了する前に記憶されることが可能である。

【０１５１】

[0167]ステップ１４１２において、プロセス１４００は、第２のタスクに関連づけられる第２の入力データを受信することを含む。第２のタスクは、第２の入力データを入力として用いて実行される。

【０１５２】

[0168]ステップ１４１４において、プロセス１４００は、ニューラルネットワークプロセッサを用いて第２の結果を計算することを含み、第２の結果を計算することは、第２の重み値セットおよび第２の入力データを用いることを含み、第２の結果の計算は、第１の結果の計算が完了した時点で発生し、かつ第２の結果は、第２のタスクの実行の結果に対応する。

【０１５３】
[0169]実装によっては、プロセス１４００は、さらに、メモリの第１の部分のサイズを縮小することを含み得、サイズの縮小は、第１の結果の計算の完了に必要とされない第１の重み値セットからの重み値に対応する。必要とされない重み値には、既に使用されている重み値が含まれ得る。第１の部分のサイズを縮小することは、第１の部分の一部を未使用として指定することと、未使用の一部をもはや第１の部分の一部ではないものとして扱うことを含み得る。
【０１５４】
[0170]実装によっては、プロセス１４００は、さらに、メモリの第２の部分のサイズを縮小することを含んでもよく、該サイズは、中間結果を記憶するために必要なメモリの量に対応する。メモリの量は、第１の結果を計算する過程で減少し得る。第２の部分のサイズを縮小することは、第２の部分の一部をもはや第２の部分の一部ではないとして指定することを含んでもよい。
【０１５５】
[0171]実装によっては、第１の重み値セットは、第１のニューラルネットワークの全ての重み値より少ない数の重み値を含む。これらの実装において、プロセス１４００は、さらに、第１のニューラルネットワークの追加の重み値セットを、コンピューティングシステムの第２のメモリから読み取ることを含んでもよい。第２のメモリは、オフチップ・プロセッサ・メモリ、オンチップキャッシュ、および／または第２のニューラルネットワークプロセッサのメモリであり得る。プロセス１４００は、さらに、追加の重み値セットをメモリの第１の部分に記憶することを含んでもよく、追加の重み値セットは、第２の重み値セットの記憶に先行して記憶される。
【０１５６】
[0172]図１５は、これまでに論じた様々な例を用いることができるネットワーク１５００を示す。図１５の例示的なネットワーク１５００は、ネットワークデバイスを備えるノード、スイッチおよびルータなどの様々な異なるタイプのネットワークデバイスを含む。所定の例において、ネットワーク１５００は、ポイントツーポイント・リンクを備えるスイッチドアーキテクチャに基づくものであってもよい。図１５に示すように、ネットワーク１５００は、複数のスイッチ１１０４ａ～１１０４ｄを含み、これらは、ネットワークに配置されてもよい。事例によっては、スイッチは、Ｃｌｏｓネットワークなどの多層網に配列される。ローカル・エリア・ネットワーク（ＬＡＮ）セグメント間でパケットをフィルタリングしかつ転送するネットワークデバイスを、スイッチと称する場合がある。スイッチは、概して、開放型システム間相互接続（ＯＳＩ）参照モデルのデータリンク層（層２）および時としてネットワーク層（層３）において動作し、かつ幾つかのパケットプロトコルをサポートしてもよい。スイッチ１１０４ａ～１１０４ｄは、複数のノード１１０２ａ～１１０２ｈへ接続され、かつ任意の２つのノード間に複数の経路を提供してもよい。
【０１５７】
[0173]ネットワーク１５００は、他のサブネット、ＬＡＮ、広域ネットワーク（ＷＡＮ）またはインターネットなどの他のネットワーク１５０８との接続のための、ルータ１５０６と称されることもある１つまたは複数のネットワークデバイスも含んでもよい。ルータは、ヘッダおよび転送テーブルを用いてパケットを転送するための最適経路を決定し、かつインターネット制御メッセージプロトコル（ＩＣＭＰ）などのプロトコルを用いて相互に通信しかつ任意の２デバイス間の最良ルートを構成する。
【０１５８】
[0174]例によっては、ネットワーク１５００は、ケーブルネットワーク、インターネット、無線ネットワーク、セルラネットワークおよび他の専用および／または公衆ネットワークなどの多くの異なるタイプのネットワークのうちのいずれか１つまたは組合せを含んでもよい。相互接続されるスイッチ１１０４ａ～１１０４ｄ、もしあればルータ１５０６は、スイッチファブリック、ファブリック、ネットワークファブリック、または単にネッ

トワークと称されることがある。コンピュータネットワークのコンテキストにおいて、本明細書における「ファブリック」および「ネットワーク」という用語は、同義で使用されてもよい。

【０１５９】

[0175]ノード１１０２ａ～１１０２ｈは、ユーザデバイス、サービスプロバイダ・コンピュータまたは第三者コンピュータを表す、ホストシステム、プロセッサノード、ストレージサブシステムおよびＩ／Ｏシャーシの任意の組合せであってもよい。

【０１６０】

[0176]ユーザデバイスは、アプリケーション１５３２（たとえば、ウェブブラウザまたはモバイルデバイス・アプリケーション）にアクセスするためのコンピューティングデバイスを含んでもよい。態様によっては、アプリケーション１５３２は、コンピューティング・リソース・サービスまたはサービスプロバイダによってホストされ、管理されかつ／または提供されてもよい。アプリケーション１５３２は、ユーザが、たとえばウェブコンテンツ（たとえば、ウェブページ、音楽、ビデオ、他）にアクセスするために、サービスプロバイダ・コンピュータと対話することを可能にしてもよい。ユーザデバイスは、たとえば、携帯電話、スマートフォン、携帯情報端末（ＰＤＡ）、ラップトップコンピュータ、ネットブックコンピュータ、デスクトップコンピュータ、シンクライアントデバイス、タブレットコンピュータ、電子書籍（ｅ－ブック）リーダ、ゲーム機、他などのコンピューティングデバイスであってもよい。例によっては、ユーザデバイスは、他のネットワーク１５０８を介してサービスプロバイダ・コンピュータと通信状態であってもよい。さらに、ユーザデバイスは、サービスプロバイダ・コンピュータによって管理され、制御される分散システムの一部であっても、他に、サービスプロバイダ・コンピュータの一部（たとえば、サービスプロバイダ・コンピュータと統合されたコンソールデバイス）であってもよい。

【０１６１】

[0177]図１５のノード１１０２ａ～１１０２ｈは、１つまたは複数のサービスプロバイダ・コンピュータも表してもよい。１つまたは複数のサービスプロバイダ・コンピュータは、ユーザが対話し得るユーザデバイス上で実行されるように構成されるネイティブアプリケーションを提供してもよい。サービスプロバイダ・コンピュータは、例によっては、クライアントエンティティ、低遅延のデータストレージ、耐久性のあるデータストレージ、データアクセス、管理、仮想化、クラウドベースのソフトウェアソリューション、電子コンテンツのパフォーマンス管理、等々などの、但しこれらに限定されないコンピューティングリソースを提供してもよい。サービスプロバイダ・コンピュータは、また、ウェブホスティング、データベース化、コンピュータアプリケーション開発および／または実装プラットフォーム、これらの、またはこれらに類似するものの組合せをユーザへ提供するように動作可能であってもよい。実施形態によっては、サービスプロバイダ・コンピュータは、ホストされるコンピューティング環境において実装される１つまたは複数の仮想マシンとして提供されてもよい。ホストされるコンピューティング環境には、１つまたは複数の迅速にプロビジョニングおよびリリースされるコンピューティングリソースが含まれてもよい。これらのコンピューティングリソースは、コンピューティング、ネットワーキング、および／またはストレージデバイスを含んでもよい。ホストされるコンピューティング環境は、クラウドコンピューティング環境と称されることもある。サービスプロバイダ・コンピュータは、おそらくはクラスタに配置される１つまたは複数のサーバをサーバファームとして、または互いに関連づけられない個々のサーバとして含んでもよく、かつアプリケーション１５３２および／またはクラウドベースのソフトウェアサービスをホストしてもよい。これらのサーバは、統合された分散コンピューティング環境の一部として構成されてもよい。態様によっては、サービスプロバイダ・コンピュータは、追加的に、または代替として、たとえば、携帯電話、スマートフォン、携帯情報端末（ＰＤＡ）、ラップトップコンピュータ、デスクトップコンピュータ、ネットブックコンピュータ、サーバコンピュータ、シンクライアントデバイス、タブレットコンピュータ、ゲーム機、他な

どのコンピューティングデバイスを含んでもよい。インスタンスによっては、サービスプロバイダ・コンピュータは、１つまたは複数の第三者コンピュータと通信してもよい。

【０１６２】

[0178]ある例示的な構成において、ノード１１０２ａ～１１０２ｈは、少なくとも１つのメモリ１５１８と、１つまたは複数の処理ユニット（または、プロセッサ１５２０）とを含んでもよい。プロセッサ１５２０は、ハードウェア、コンピュータ実行可能命令、ファームウェア、またはこれらの組合わせで実装されてもよい。プロセッサ１５２０のコンピュータ実行可能命令またはファームウェア実装は、記述される様々な関数を実行するために任意の適切なプログラミング言語で書かれたコンピュータ実行可能または機械実行可能命令を含んでもよい。

【０１６３】

[0179]インスタンスによっては、ハードウェアプロセッサ１５２０は、シングルコアプロセッサであっても、マルチコアプロセッサであってもよい。マルチコアプロセッサは、同じプロセッサ内に複数の処理ユニットを含んでもよい。実施形態によっては、マルチコアプロセッサは、バスおよび二次または三次キャッシュなどの所定のリソースを共有してもよい。インスタンスによっては、シングルコアまたはマルチコアプロセッサにおける各コアは、複数の実行論理プロセッサ（または、実行スレッド）も含むことがある。このようなコア（たとえば、複数の論理プロセッサを備えるコア）では、実行パイプラインの幾つかのステージ、およびより低次のキャッシュをも、共有されることがある。

【０１６４】

[0180]メモリ１５１８は、プロセッサ１５２０上でロード可能かつ実行可能なプログラム命令、ならびにこれらのプログラムの実行中に発生されるデータを記憶してもよい。ノード１１０２ａ～１１０２ｈの構成およびタイプに依存して、メモリ１５１８は、揮発性（ＲＡＭなど）であっても、かつ／または不揮発性（ＲＯＭ、フラッシュメモリ、他など）であってもよい。メモリ１５１８は、オペレーティングシステム１５２８、１つまたは複数のデータストア１５３０、１つまたは複数のアプリケーションプログラム１５３２、１つまたは複数のドライバ１５３４、および／または本明細書に開示している特徴を実装するためのサービスを含んでもよい。

【０１６５】

[0181]オペレーティングシステム１５２８は、タスクのスケジューリング、アプリケーションの実行、および／またはコントローラ周辺機器などのノード１１０２ａ～１１０２ｈの基本機能をサポートしてもよい。実装によっては、サービスプロバイダ・コンピュータは、１つまたは複数の仮想マシンをホストしてもよい。これらの実装において、各仮想マシンは、その独自のオペレーティングシステムを実行するように構成されてもよい。オペレーティングシステムの例としては、Ｕｎｉｘ、Ｌｉｎｕｘ（登録商標）、Ｗｉｎｄｏｗｓ、Ｍａｃ　ＯＳ、ｉＯＳ、Ａｎｄｒｏｉｄおよびこれらに類似するものが含まれる。また、オペレーティングシステム１５２８は、独自仕様のオペレーティングシステムである場合もある。

【０１６６】

[0182]データストア１５３０は、オペレーティングシステム１５２８、アプリケーションプログラム１５３２またはドライバ１５３４によって使用されかつ／または作用される永続的または一時的なデータを含んでもよい。このようなデータの例としては、ウェブページ、ビデオデータ、オーディオデータ、画像、ユーザデータ、等々が含まれる。データストア１５３０内の情報は、実装によっては、ネットワーク１５０８を介してノード１１０２ａ～１１０２ｈへ提供されてもよい。事例によっては、データストア１５３０は、追加的に、または代替として、記憶されたアプリケーションプログラムおよび／またはドライバを含んでもよい。あるいは、または追加的に、データストア１５３０は、標準および／または独自のソフトウェアライブラリ、および／または標準および／または独自のアプリケーション・ユーザ・インタフェース（ＡＰＩ）ライブラリを記憶してもよい。データストア１５３０に記憶される情報は、機械可読オブジェクトコード、ソースコード、イン

タープリタ型コードまたは中間コードであってもよい。
【０１６７】
[0183]ドライバ１５３４は、ノード内のコンポーネント間の通信を提供し得るプログラムを含む。たとえば、一部のドライバ１５３４は、オペレーティングシステム１５２８と、追加のストレージ１５２２、ネットワークデバイス１５２４および／またはＩ／Ｏデバイス１５２６との間の通信を提供してもよい。あるいは、または追加的に、一部のドライバ１５３４は、アプリケーションプログラム１５３２とオペレーティングシステム１５２８との間、および／またはアプリケーションプログラム１５３２とサービスプロバイダ・コンピュータへアクセス可能な周辺デバイスとの間の通信を提供してもよい。多くの事例において、ドライバ１５３４は、十分に理解されている機能を提供するドライバ（たとえば、プリンタドライバ、ディスプレイドライバ、ハードディスク・ドライバ、ソリッドステートデバイス・ドライバ）を含んでもよい。他の事例において、ドライバ１５３４は、独自仕様または特殊化された機能を提供してもよい。
【０１６８】
[0184]サービスプロバイダ・コンピュータまたはサーバも、追加のストレージ１５２２を含んでもよく、追加のストレージ１５２２は、取外し可能ストレージおよび／または取外し不能ストレージを含んでもよい。追加のストレージ１５２２としては、磁気ストレージ、光ディスク、ソリッド・ステート・ディスク、フラッシュメモリおよび／またはテープストレージが含まれてもよい。追加のストレージ１５２２は、ノード１１０２ａ～１１０２ｈと同じシャーシ内に収容されてもよく、または外部エンクロージャ内に存在してもよい。メモリ１５１８および／または追加のストレージ１５２２およびこれらに関連づけられるコンピュータ可読媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、およびコンピューティングデバイスに関する他のデータの不揮発性ストレージを提供してもよい。実装によっては、メモリ１５１８は、ＳＲＡＭ、ＤＲＡＭまたはＲＯＭなどの複数の異なるタイプのメモリを含んでもよい。
【０１６９】
[0185]共に取外し可能および取外し不能であるメモリ１５１８および追加のストレージ１５２２は、コンピュータ可読記憶媒体の例である。たとえば、コンピュータ可読記憶媒体としては、情報を記憶するための方法または技術において実装される揮発性または不揮発性、取外し可能または取外し不能な媒体が含まれてもよく、情報としては、たとえば、コンピュータ可読命令、データ構造、プログラムモジュールまたは他のデータが含まれる。メモリ１５１８および追加のストレージ１５２２は、コンピュータ記憶媒体の例である。ノード１１０２ａ～１１０２ｈ内に存在し得る追加のタイプのコンピュータ記憶媒体としては、ＰＲＡＭ、ＳＲＡＭ、ＤＲＡＭ、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリまたは他のメモリ技術、ＣＤ－ＲＯＭ、ＤＶＤまたは他の光ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージまたは他の磁気ストレージデバイス、ソリッド・ステート・ドライブ、または、所望される情報を記憶するために使用されることが可能でありかつノード１１０２ａ～１１０２ｈからアクセスされることが可能な他の何らかの媒体が含まれ得るが、この限りではない。コンピュータ可読媒体としては、１つの媒体タイプの複数のユニットを含む、上述の媒体タイプのいずれかの組合せも含まれる。
【０１７０】
[0186]あるいは、または追加的に、コンピュータ可読通信媒体は、コンピュータ可読命令、プログラムモジュール、または搬送波または他の伝送などのデータ信号内で送信される他のデータを含んでもよい。しかしながら、本明細書で使用するコンピュータ可読記憶媒体という用語は、コンピュータ可読通信媒体を包含しない。
【０１７１】
[0187]ノード１１０２ａ～１１０２ｈは、キーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイス、ディスプレイ、スピーカ、プリンタおよびこれらに類似するものなどのＩ／Ｏデバイス１５２６も含んでもよい。ノード１１０２ａ～１１０２ｈは、１つまたは複数の通信チャネル１５３６も含んでもよい。通信チャネル１５３６は、ノード

１１０２ａ～１１０２ｈの様々なコンポーネントがそれを介して通信できる媒体を提供してもよい。１つまたは複数の通信チャネル１５３６は、バス、リング、スイッチングファブリックまたはネットワークの形式をとってもよい。

【０１７２】

[0188]ノード１１０２ａ～１１０２ｈは、ノード１１０２ａ～１１０２ｈが記憶されたデータベース、別のコンピューティングデバイスまたはサーバ、ユーザ端末、および／またはネットワーク１５００上の他のデバイスと通信できるようにする、ネットワークデバイス１５２４も含んでもよい。

【０１７３】

[0189]実装によっては、ネットワークデバイス１５２４は、ＰＣＩベースのデバイスなどの周辺機器である。これらの実装において、ネットワークデバイス１５２４は、ホストデバイスと通信するためのＰＣＩインタフェースを含む。「ＰＣＩ」または「ＰＣＩベースの」という用語は、原初のＰＣＩ標準、ＰＣＩ－Ｘ、アクセラレーテッド・グラフィックス・ポート（ＡＧＰ）、およびＰＣＩ－Ｅｘｐｒｅｓｓ（ＰＣＩｅ）を含む、バスプロトコルのＰＣＩファミリにおける任意のプロトコル、または、本明細書において論じるＰＣＩプロトコルを基礎とする他の任意の改良または派生プロトコルを記述するために使用される場合がある。ＰＣＩベースのプロトコルは、ローカル周辺機器などのデバイスをホストデバイスへ接続するための標準バスプロトコルである。標準バスプロトコルは、様々な製造業者がそのための仕様を規定して採用しているデータ転送プロトコルである。製造業者は、対応デバイスがバスプロトコルを実装するコンピューティングシステムと互換性があること、およびその逆を保証している。本明細書で使用するＰＣＩベースのデバイスという用語には、不揮発性メモリエクスプレス（ＮＶＭｅ）を用いて通信するデバイスも含まれる。ＮＶＭｅは、コンピューティングシステムへ取り付けられる不揮発性記憶媒体にＰＣＩｅを用いてアクセスするためのデバイスインタフェース仕様である。たとえば、バス・インタフェース・モジュールは、ＮＶＭｅを実装してもよく、よってネットワークデバイス１５２４は、ＰＣＩｅインタフェースを用いてコンピューティングシステムへ接続されてもよい。

【０１７４】

[0190]ＰＣＩベースのデバイスは、１つまたは複数の機能を含んでもよい。「機能」は、ネットワークデバイス１５２４により提供され得る動作を記述する。機能の例としては、とりわけ、大容量ストレージコントローラ、ネットワークコントローラ、ディスプレイコントローラ、メモリコントローラ、シリアル・バス・コントローラ、ワイヤレスコントローラ、および暗号化および復号化コントローラが含まれる。事例によっては、ＰＣＩベースのデバイスは、２つ以上の機能を含んでもよい。たとえば、ＰＣＩベースのデバイスは、大容量ストレージコントローラおよびネットワークアダプタを提供してもよい。別の例として、ＰＣＩベースのデバイスは、２つの異なるストレージリソースを制御するために２つのストレージコントローラを提供してもよい。実装によっては、ＰＣＩベースのデバイスは、最大８つの機能を有してもよい。

【０１７５】

[0191]実装によっては、ネットワークデバイス１５２４は、シングルルートＩ／Ｏ仮想化（ＳＲ－ＩＯＶ）を含んでもよい。ＳＲ－ＩＯＶは、ＰＣＩベースのデバイスに包含され得る拡張機能である。ＳＲ－ＩＯＶは、１つの物理リソース（たとえば、単一のネットワーク・インタフェース・コントローラ）が複数のリソース（たとえば、６４個のネットワーク・インタフェース・コントローラ）として現出することを可能にする。したがって、所定の機能（たとえば、ネットワーク・インタフェース・コントローラ）を提供するＰＣＩベースのデバイスは、ＰＣＩベースのデバイスを利用するデバイスには、同じ機能を提供する複数のデバイスであるように見えることがある。ＳＲ－ＩＯＶ対応のストレージ・アダプタ・デバイスの機能は、物理機能（ＰＦ）または仮想機能（ＶＦ）として分類されてもよい。物理機能は、発見され、管理されかつ操作されることが可能な、デバイスが備える完全な機能である。物理機能は、ストレージ・アダプタ・デバイスを構成または制

御するために使用されることが可能な構成リソースを有する。物理機能は、非仮想化デバイスが有すると思われる同じ構成アドレス空間およびメモリアドレス空間を含む。１つの物理機能は、それに関連づけられる幾つかの仮想機能を有する場合がある。仮想機能は、物理機能に類似するものであるが、概して構成リソースが不足し得る軽量の機能であり、概して、その基礎を成す物理機能の構成によって制御される。物理機能および／または仮想機能は、各々、ホストデバイス上で実行される個々の実行スレッド（たとえば、仮想マシンなど）へ割り当てられてもよい。

【０１７６】

[0192]一連の機能の一部として、様々な例では、ノード１１０２ａ～１１０２ｈのうちの１つまたはそれ以上が、これまでにその様々な例について述べたニューラルネットワーク処理エンジンおよび／またはニューラルネットワークプロセッサを包含することができる。これらの例において、こうして装備されたノード１１０２ａ～１１０２ｈは、機械学習タスクおよびニューラルネットワーク処理タスク用に最適化されることが可能である。

【０１７７】

[0193]本明細書に記述しているモジュールは、ソフトウェアモジュール、ハードウェアモジュール、またはこれらの適切な組合せであってもよい。モジュールがソフトウェアモジュールであれば、モジュールは、非一時的コンピュータ可読媒体上に具現され、かつ本明細書に記述しているコンピュータシステムのいずれかにおけるプロセッサによって処理されることが可能である。記述しているプロセスおよびアーキテクチャが、リアルタイム、またはユーザ対話に先行する非同期モードのいずれかで実行可能であることは、留意されるべきである。モジュールは、図１５において提案されている方法で構成されてもよく、かつ／または本明細書に記載の機能は、別個のモジュールとして存在する１つまたは複数のモジュールによって提供されることが可能であり、かつ／または本明細書に記載のモジュール機能は、複数のモジュールに分散されることが可能である。

【０１７８】

[0194]したがって、本明細書本文および図面は、限定を意味するものではなく、例示的なものと見なされるべきである。しかしながら、これらに対して、特許請求の範囲に記載されている本開示のより広い精神および範囲を逸脱することなく、様々な改変および変更が行われ得ることは、明らかであろう。

【０１７９】

[0195]開示する技術の例示的な実施形態は、以下の条項によって記述することができる。

１．ニューラルネットワーク処理のための集積回路であって、

ニューラルネットワークの重み値セットを記憶する複数のメモリバンクであって、重み値セットは、ニューラルネットワークの全ての重み値を含み、重み値セットは、結果が既知である入力データを用いて先に決定されていて、複数のメモリバンクからの各バンクは、独立してアクセス可能である、複数のメモリバンクと、

第１の処理エンジンアレイであって、各処理エンジンは、乗数累算器回路を含み、第１の処理エンジンアレイは、複数のメモリバンクと同じダイ上にある、第１の処理エンジンアレイと、を備え、

入力データを受信した時点で、集積回路は、重み値セットを用いて、ニューラルネットワークが実行するように訓練されたタスクを実行するように構成され、タスクは、結果が既知である入力データによって規定され、かつ、タスクを実行することは、

複数のメモリバンクから重み値を読み取ることと、

重み値および入力データを第１の処理エンジンアレイに入力することであって、第１の処理エンジンアレイにおける各処理エンジンは、複数の重み値からの１つの重み値と、複数の入力データからの１つの入力値とを用いて加重和を計算することと、

結果を計算すること、を含み、結果を計算することは、第１の処理エンジンアレイからの出力を累積することを含み、結果は、タスク実行の結果に対応する、集積回路。

２．第２の処理エンジンアレイをさらに備え、第１のメモリバンクセットは、第１の処

理エンジンアレイによって使用されるように構成され、第２のメモリバンクセットは、第２の処理エンジンアレイによって使用されるように構成され、かつ重み値セットは、第１のメモリバンクセットおよび第２のメモリバンクセットに記憶される、条項１に記載の集積回路。

３．オフチップメモリとの通信を可能にするメモリコントローラと、

ホストバスとの通信を可能にするバス・インタフェース・コントローラと、

集積回路のコンポーネント間でデータを移動するように構成される管理コントローラと、

複数のメモリバンク、メモリコントローラ、バス・インタフェース・コントローラ、および管理コントローラの間の通信を可能にする通信ファブリックと、をさらに備える、条項１～２に記載の集積回路。

４．集積回路であって、

第１の処理エンジンアレイと、

ニューラルネットワークの重み値セットを記憶する複数のメモリバンクと、を備え、複数のメモリバンクからの各バンクは、独立してアクセス可能であり、かつ複数のメモリバンクおよび第１の処理エンジンアレイは、同じダイ上にあり、

入力データを受信した時点で、集積回路は、重み値セットを用いて、ニューラルネットワーク用に規定されたタスクを実行するように構成され、かつ、タスクを実行することは、

複数のメモリバンクから重み値を読み取ることと、

重み値および入力データを第１の処理エンジンアレイに入力することと、

第１の処理エンジンアレイを用いて結果を計算すること、を含み、結果は、タスク実行の結果に対応する、集積回路。

５．タスクを実行することは、さらに、

複数のメモリバンクのうちの異なるメモリバンクから２つ以上の値を同時に読み取ることを含む、条項４に記載の集積回路。

６．２つ以上の値には、重み値、入力値または中間結果が含まれる、条項５に記載の集積回路。

７．タスクを実行することは、さらに、

複数のメモリバンクのうちの第１のメモリバンクへ第１の値を書き込むことと、

複数のメモリバンクのうちの第２のメモリバンクから第２の値を読み取ること、を含み、第１の値は、第２の値が読み取られると同時に書き込まれる、条項４～６のいずれかに記載の集積回路。

８．第１の値および第２の値は、重み値、および入力値、または中間結果を含む、条項７に記載の集積回路。

９．重み値セットは、ニューラルネットワークの全ての重み値を含む、条項４～８のいずれかに記載の集積回路。

１０．第２の処理エンジンアレイをさらに備え、複数のメモリバンクのうちの第１のメモリバンクセットは、第１の処理エンジンアレイによって使用されるように構成され、複数のメモリバンクのうちの第２のメモリバンクセットは、第２の処理エンジンアレイによって使用されるように構成され、第１のメモリバンクセットおよび第２のメモリバンクセットは各々、重み値セットの一部分を含み、かつ、タスクを実行することは、さらに、

第１の処理エンジンアレイよって中間結果を計算することであって、第１の処理エンジンアレイは、中間結果を、第１のメモリバンクセットからの重み値を用いて計算することと、

第１の処理エンジンアレイによって、第２のメモリバンクセットから追加の重み値を読み取ることであって、第１の処理エンジンアレイは、中間結果および追加の重み値を用いて結果を計算すること、を含む、条項４～９のいずれかに記載の集積回路。

１１．重み値セットによる占有は、第２のメモリバンクセットの全てより少なく、第２の処理エンジンアレイは、重み値セットによって占有されていない第２のメモリバンクセットの一部を用いて計算を実行する、条項１０に記載の集積回路。

１２．第１のメモリバンクセットおよび第２のメモリバンクセットに記憶される重み値

セットの部分は、ニューラルネットワークの全ての重み値を含む、条項１０に記載の集積回路。
１３．複数のメモリバンクの第１の部分は、結果を計算するための中間結果を記憶するために予約され、かつ重み値セットは、ニューラルネットワークの全ての重み値より少ない重み値を含む、条項４～１２のいずれかに記載の集積回路。
１４．タスクを実行することは、さらに、
中間結果を記憶するために必要なメモリの量が減少していると決定することと、
別のメモリから追加の重み値セットを読み取ることと、
追加の重み値セットを複数のメモリバンクの第１の部分に記憶することを含み、追加の重み値セットは、結果の計算に必要とされる前に記憶される、条項１３に記載の集積回路。
１５．第１の処理エンジンアレイは、１つの処理エンジンセットを含み、処理エンジンセットのうちの各処理エンジンは、結果を、処理エンジンセットのうちの別の処理エンジンへ直に出力する、条項４～１４のいずれかに記載の集積回路。
１６．第１の処理エンジンアレイのうちの各処理エンジンは、乗数累算器回路を含む、条項４～１５のいずれかに記載の集積回路。
１７．ニューラルネットワークは、重み付き有向グラフから導出される複数の重み値と、重み付き有向グラフにおける各ノードについて計算を実行するための命令セットとを含み、複数の重み値は、既知の入力データを用いてタスクを実行することにより、先に決定されている、条項４～１６のいずれかに記載の集積回路。
１８．方法であって、
ニューラルネットワーク処理回路の複数のメモリバンクに、１つの重み値セットを記憶することであって、ニューラルネットワーク処理回路は、複数のメモリバンクと同じダイ上に１つの処理エンジンアレイを含み、かつ重み値セットは、入力データの受信に先行して記憶されることと、
入力データを受信することと、
重み値セットを用いて、ニューラルネットワーク用に規定されるタスクを実行すること、を含み、タスクを実行することは、
複数のメモリバンクから重み値を読み取ることと、
重み値および入力データを処理エンジンアレイに入力することと、
処理エンジンアレイを用いて結果を計算すること、を含み、結果は、タスク実行の結果に対応する、方法。
１９．重み値セットは、ニューラルネットワークの全ての重み値を含む、条項１８に記載の方法。
２０．重み値セットは、ニューラルネットワークの全ての重み値の第１の部分を含み、かつ、さらに、
複数のメモリバンクが利用可能なスペースを有すると決定することと、
ニューラルネットワークの全ての重み値の第２の部分を読み取ることであって、第２の部分は、追加のメモリから読み取られることと、
第２の部分を使用可能なスペースへ書き込むこと、を含む、条項１８または１９に記載の方法。
２１．追加のメモリは、同じダイ上の第２の処理エンジンアレイに関連づけられる、請求項１８～２０のいずれかに記載の方法。
２２．追加のメモリは、オフチップである、請求項１８～２１のいずれかに記載の方法。
２３．重み値を読み取ることは、同時に、複数のメモリバンクのうちの第１のメモリバンクから第１の重み値を読み取り、かつ複数のメモリバンクのうちの第２のメモリバンクから第２の重み値を読み取ることを含む、条項１８～２２のいずれかに記載の方法。
２４．処理エンジンアレイを用いて中間結果を決定することと、
中間結果を、複数のメモリバンクのうちの１つのメモリバンクに記憶すること、をさらに含み、中間結果は、重み値が読み取られると同時に記憶される、条項１８～２３のいずれかに記載の方法。

２５．ニューラルネットワーク処理システムであって、
第１の処理エンジンアレイであって、各処理エンジンが乗数累算器回路を含む、第１の処理エンジンアレイと、
第１の重み値セットを記憶する第１のメモリバンクセットであって、第１の重み値セットは、結果が既知である入力データを用いて先に決定されており、第１のメモリバンクセットのうちの各バンクは、独立してアクセス可能であり、かつ第１の処理エンジンアレイおよび第１のメモリバンクセットは、同じ第１のダイ上にある、第１のメモリバンクセットと、
第２の処理エンジンアレイであって、各処理エンジンが乗数累算器回路を含む、第２の処理エンジンアレイと、
第２の重み値セットを記憶する第２のメモリバンクセットであって、第２の重み値セットは、結果が既知である入力データを用いて先に決定されており、第２のメモリバンクセットのうちの各バンクは、独立してアクセス可能であり、かつ第２の処理エンジンアレイおよび第２のメモリバンクセットは、同じ第２のダイ上にある、第２のメモリバンクセットと、を備え、第１の重み値セットおよび第２の重み値セットは、ニューラルネットワークの全ての重み値を含み、かつ第１の重み値セットおよび第２の重み値セットは、ニューラルネットワーク処理システムによる入力データの受信に先行して記憶され、
入力データを受信した時点で、ニューラルネットワーク処理システムは、ニューラルネットワークが実行するように訓練されたタスクを実行するように構成され、タスクは、結果が既知である入力データによって規定され、かつ、タスクを実行することは、
第１の処理エンジンアレイを用いて中間結果を計算することであって、第１の処理エンジンアレイにおける各処理エンジンは、第１の重み値セットからの１つの重み値および入力データからの１つの入力値を用いて加重和を計算することと、
第１のメモリバンクセットからの中間結果および状態を、第２のメモリバンクセットへコピーすることであって、状態は、第１の処理エンジンアレイにより完了されるタスクのステージを記述することと、
第２の処理エンジンアレイを用いて最終結果を計算すること、を含み、第２の処理エンジンアレイは、状態を用いてタスクの状態を決定し、第２の処理エンジンアレイにおける各処理エンジンは、第２の重み値セットからの１つの重み値、および中間結果からの１つの値を用いて加重和を計算し、かつ最終結果は、タスク実行の結果に対応する、ニューラルネットワーク処理システム。
２６．第１のダイおよび第２のダイは、同じダイの部分であり、ダイは、通信ファブリックと、ダイレクト・メモリ・アクセス・コントローラとを含み、かつダイレクト・メモリ・アクセス・コントローラは、中間結果および状態を通信ファブリックを介してコピーする、条項２５に記載のニューラルネットワーク処理システム。
２７．第１のダイは、第１のパッケージ内にあり、かつ第２のダイは、第２のパッケージ内にあり、かつ、
第１のダイおよび第２のダイに連結されるバスをさらに備え、中間結果および状態は、バスを介してコピーされる、条項２５または２６に記載のニューラルネットワーク処理システム。
２８．ニューラルネットワーク処理システムであって、
第１の処理エンジンアレイと、
第１の重み値セットを記憶する第１のメモリバンクセットであって、第１の処理エンジンアレイおよび第１のメモリバンクセットは、同じ第１のダイ上にある、第１のメモリバンクセットと、
第２の処理エンジンアレイと、
第２の重み値セットを記憶する第２のメモリバンクセットであって、第２の処理エンジンアレイおよび第２のメモリバンクセットは、同じ第２のダイ上にある、第２のメモリバンクセットと、を備え、第１の重み値セットおよび第２の重み値セットは、ニューラルネットワークの全ての重み値を含み、かつ第１の重み値セットおよび第２の重み値セットは

、ニューラルネットワーク処理システムによる入力データの受信に先行して記憶され、
入力データを受信した時点で、ニューラルネットワーク処理システムは、ニューラルネットワーク用に規定されるタスクを実行するように構成され、かつ、タスクを実行することは、
第１の処理エンジンアレイを用いて中間結果を計算することであって、第１の処理エンジンアレイは、第１の重み値セットおよび入力データを用いて中間結果を計算することと、
第１のメモリバンクセットからの中間結果および状態を、第２のメモリバンクセットへコピーすることであって、状態は、第１の処理エンジンアレイにより完了されるタスクのステージを記述することと、
第２の処理エンジンアレイを用いて最終結果を計算すること、を含み、第２の処理エンジンアレイは、最終結果を、第２の重み値セット、中間結果および状態を用いて計算し、かつ最終結果は、タスク実行の結果に対応する、ニューラルネットワーク処理システム。
２９．第１のダイおよび第２のダイは、同じダイの部分である、条項２８に記載のニューラルネットワーク処理システム。
３０．第１のダイおよび第２のダイは、異なるダイである、条項２８または２９に記載のニューラルネットワーク処理システム。
３１．バスをさらに備え、第１のダイおよび第２のダイは各々、バスに連結され、かつ中間結果および状態は、バスを介してコピーされる、条項３０に記載のニューラルネットワーク処理システム。
３２．タスクを実行することは、さらに、
第１のメモリバンクセットまたは第２のメモリバンクセットのうちの異なるメモリバンクから、２つ以上の値を同時に読み取ることを含む、条項２８～３１のいずれかに記載のニューラルネットワーク処理システム。
３３．２つ以上の値には、重み値、入力値または中間結果が含まれる、請求項３２に記載のニューラルネットワーク処理システム。
３４．タスクを実行することは、さらに、
第１のメモリバンクセットまたは第２のメモリバンクセットのうちの第１のメモリバンクへ第１の値を書き込むことと、
第１のメモリバンクセットまたは第２のメモリバンクセットのうちの第２のメモリバンクから第２の値を読み取ること、を含み、第１の値は、第２の値が読み取られると同時に書き込まれる、条項２８～３３のいずれかに記載のニューラルネットワーク処理システム。
３５．第１の値または第２の値には、重み値、入力値または中間結果が含まれる、請求項３４に記載のニューラルネットワーク処理システム。
３６．タスクを実行する間、プロセッサメモリから重み値は読み取られず、プロセッサメモリは、第１のダイまたは第２のダイとは異なるダイ上にある、条項２８～３５のいずれかに記載のニューラルネットワーク処理システム。
３７．第１の処理エンジンアレイおよび第２の処理エンジンアレイのうちの各処理エンジンは、乗数累算器回路を含む、条項２８～３６のいずれかに記載のニューラルネットワーク処理システム。
３８．ニューラルネットワークは、重み付き有向グラフから導出される複数の重み値と、重み付き有向グラフにおける各ノードについて計算を実行するための命令セットとを含み、複数の重み値は、既知の入力データを用いてタスクを実行することにより、先に決定されている、条項２８～３７のいずれかに記載のニューラルネットワーク処理システム。
３９．ニューラルネットワークを処理する方法であって、
第１の重み値セットを、ニューラルネットワーク処理システムの第１のメモリバンクセットに記憶することと、
第２の重み値セットを、ニューラルネットワーク処理システムの第２のメモリバンクセットに記憶することであって、第１の重み値セットおよび第２の重み値セットは、ニューラルネットワークの全ての重み値を含み、かつ第１の重み値セットおよび第２の重み値セットは、ニューラルネットワーク処理システムによる入力データの受信に先行して記憶さ

れることと、
入力データを受信することと、
ニューラルネットワーク用に規定されるタスクを実行すること、を含み、タスクを実行することは、
第１の処理エンジンアレイを用いて中間結果を計算することであって、第１の処理エンジンアレイは、第１のメモリバンクセットと同じ第１のダイ上にあり、かつ第１の処理エンジンアレイは、第１の重み値セットおよび入力データを用いて中間結果を計算することと、
第１のメモリバンクセットからの中間結果および状態を、第２のメモリバンクセットへコピーすることであって、状態は、第１の処理エンジンアレイにより完了されるタスクのステージを記述することと、
第２の処理エンジンアレイを用いて最終結果を計算すること、を含み、第２の処理エンジンアレイは、第２のメモリバンクセットと同じ第２のダイ上にあり、第２の処理エンジンアレイは、最終結果を、第２の重み値セット、中間結果および状態を用いて計算し、かつ最終結果は、タスク実行の結果に対応する、方法。
４０．第１のダイおよび第２のダイは、同じダイの部分である、条項３９に記載の方法。
４１．第１のダイおよび第２のダイは、異なるパッケージ内にある、請求項３９に記載の方法。
４２．プロセッサメモリから重み値は読み取られず、プロセッサメモリは、第１のダイまたは第２のダイのいずれかとは異なるパッケージ内にある、条項３９～４１のいずれかに記載の方法。
４３．第１のメモリバンクセットから２つ以上の値を同時に読み取ることと、
第２のメモリバンクセットから２つ以上の値を同時に読み取ること、をさらに含む、条項３９～４２のいずれかに記載の方法。
４４．第１のメモリバンクセットから第１の値を読み取ると同時に、第１のメモリバンクセットへ第２の値を書き込むことと、
第２のメモリバンクセットから第３の値を読み取ると同時に、第２のメモリバンクセットへ第４の値を書き込むこと、をさらに含む、条項３９～４３のいずれかに記載の方法。
４５．ニューラルネットワークを処理するための集積回路であって、
処理エンジンアレイと、
第１のニューラルネットワークの第１の重み値セットを記憶する複数のメモリバンクと、を備え、第１のニューラルネットワークは、第１のタスクを実行するように構成され、第１の重み値セットは、結果が既知である入力データを用いて第１のタスクを実行することにより、先に決定されており、複数のメモリバンクからの各バンクは、独立してアクセス可能であり、かつ複数のメモリバンクおよび処理エンジンアレイは、同じダイ上にあり、
集積回路は、
第１のタスクに関連づけられる第１の入力データを受信するように構成され、第１の入力データは、複数のメモリバンクに記憶され、かつ集積回路は、
第１の重み値セットおよび第１の入力データを用いて第１の結果を計算するように構成され、第１の結果は、処理エンジンアレイを用いて計算され、かつ第１の結果は、第１のタスク実行の結果に対応し、かつ集積回路は、
第１の結果を計算する間に、複数のメモリバンクが使用可能なスペースを有することを決定し、
利用可能なスペースに第２の重み値セットを記憶するように構成され、第２の重み値セットは、第２のニューラルネットワーク用であり、第２のニューラルネットワークは、第２のタスクを実行するように構成され、第２のタスクは、第１のタスクとは異なり、かつ、第２の重み値セットを記憶することは、集積回路が第１の入力データに基づく第１のタスクの実行を完了する前に、第２のタスクを実行するように集積回路を準備し、かつ集積回路は、
第２のタスクに関連づけられる第２の入力データを受信するように構成され、第２の入

力データは、複数のメモリバンクに記憶され、かつ集積回路は、
第２の重みセットおよび第２の入力データを用いて第２の結果を計算するように構成され、第２の結果は、処理エンジンアレイを用いて計算され、第２の結果の計算は、第１の結果の計算が完了した時点で発生し、かつ第２の結果は、第２のタスク実行の結果に対応する、集積回路。
４６．複数のメモリバンクが利用可能なスペースを有すると決定することは、
第１の結果を計算するための中間結果を記憶するために必要なスペースの量を決定することと、
スペースの量が減少したと決定すること、を含む、条項４５に記載の集積回路。
４７．複数のメモリバンクが利用可能なスペースを有すると決定することは、
第１の結果を計算するために既に使用されている１つまたは複数の重み値を識別することと、
１つまたは複数の重み値によって占有されるスペースを利用可能としてマーキングすること、を含む、条項４５または４６に記載の集積回路。
４８．第１の重み値セットおよび第１のタスクは、第１のコンテキストに関連づけられる、条項４５～４７のいずれかに記載の集積回路。
４９．第２の重み値セットおよび第２のタスクは、第２のコンテキストを構成する、条項４５～４８のいずれかに記載の集積回路。
５０．集積回路であって、
処理エンジンアレイと、
第１のニューラルネットワークの第１の重み値セットを記憶する複数のメモリバンクと、を備え、第１のニューラルネットワークは、第１のタスクを実行するように構成され、複数のメモリバンクからの各バンクは、独立してアクセス可能であり、かつ複数のメモリバンクおよび処理エンジンアレイは、同じダイ上にあり、
集積回路は、
第１のタスクに関連づけられる第１の入力データを受信し、
第１の重み値セットおよび第１の入力データを用いて第１の結果を計算するように構成され、第１の結果は、処理エンジンアレイを用いて計算され、かつ第１の結果は、第１のタスク実行の結果に対応し、かつ集積回路は、
複数のメモリバンクに第２の重み値セットを記憶するように構成され、第２の重み値セットは、第２のニューラルネットワーク用であり、第２のニューラルネットワークは、第２のタスクを実行するように構成され、かつ第２の重み値セットを記憶することは、第１の結果を計算する間に発生し、かつ集積回路は、
第２のタスクに関連づけられる第２の入力データを受信し、かつ、
第２の重み値セットおよび第２の入力データを用いて第２の結果を計算するように構成され、第２の結果は、処理エンジンアレイを用いて計算され、第２の結果の計算は、第１の結果の計算が完了した時点で発生し、かつ第２の結果は、第２のタスク実行の結果に対応する、集積回路。
５１．第１の重み値セットは、第１の入力データの受信に先行して、複数のメモリバンクに記憶される、条項５０に記載の集積回路。
５２．第１の重み値セットは、第１のニューラルネットワークの全ての重み値を含む、条項５０または５１に記載の集積回路。
５３．第２の結果の計算の完了に先行して、第２のニューラルネットワークの全ての重み値が複数のメモリバンクに記憶される、条項５０～５２のいずれかに記載の集積回路。
５４．第２のニューラルネットワークの追加の重み値セットを記憶するようにさらに構成され、追加の重み値セットは、第２の結果を計算する間に記憶される、条項５０～５３のいずれかに記載の集積回路。
５５．複数のメモリバンクにおける、第１の結果を計算するための中間結果を記憶するために必要なスペースの量を決定し、
第１の結果を計算する間にスペースの量が減少したと決定し、かつ、

複数のメモリバンクにおける、もはや中間結果を記憶するために使用されないスペースを用いて、第２の重み値セットを記憶するように、さらに構成される、条項５０～５４のいずれかに記載の集積回路。

５６．第１の重み値セットのうちの、第１の結果の計算の完了に必要とされない１つまたは複数の重み値を識別し、かつ、

複数のメモリバンクにおける、１つまたは複数の重み値により占有されるスペースを用いて第２の重み値セットを記憶するように、さらに構成される、条項５０～５５のいずれかに記載の集積回路。

５７．複数のメモリバンクのうちの異なるメモリバンクから２つ以上の値を同時に読み取るように、さらに構成される、条項５０～５６のいずれかに記載の集積回路。

５８．２つ以上の値には、重み値、入力値または中間結果が含まれる、条項５０～５７のいずれかに記載の集積回路。

５９．複数のメモリバンクのうちの第１のメモリバンクへ第１の値を書き込み、かつ、

複数のメモリバンクのうちの第２のメモリバンクから第２の値を読み取るように、さらに構成され、第１の値は、第２の値が読み取られると同時に書き込まれる、条項５０～５８のいずれかに記載の集積回路。

６０．第１の値または第２の値は、重み値、入力値、または中間結果を含む、条項５９に記載の集積回路。

６１．処理エンジンアレイは、１つの処理エンジンセットを含み、処理エンジンセットのうちの各処理エンジンは、結果を、処理エンジンセットのうちの別の処理エンジンへ直に出力する、条項５０～６０のいずれかに記載の集積回路。

６２．処理エンジンアレイのうちの各処理エンジンは、乗数累算器回路を含む、条項５０～６１のいずれかに記載の集積回路。

６３．各ニューラルネットワークは、個々の重み付き有向グラフから導出される個々の複数の重み値と、重み付き有向グラフにおける各ノードについて計算を実行するための個々の命令セットとを含み、複数の重み値は、既知の入力データを用いて個々のタスクを実行することにより、先に決定されている、条項５０～６２のいずれかに記載の集積回路。

６４．コンピュータ実行方法であって、

ニューラルネットワークプロセッサを含むコンピューティングシステムにより、第１のニューラルネットワークの第１の重み値セットを記憶することであって、第１のニューラルネットワークは、第１のタスクを実行するように構成され、かつ第１の重み値セットは、ニューラルネットワークプロセッサにおけるメモリの第１の部分に記憶されることと、

第１のタスクに関連づけられる第１の入力データを受信することであって、第１の入力データは、第１の重み値セットの記憶後に受信されることと、

ニューラルネットワークプロセッサを用いて第１の結果を計算することであって、第１の結果を計算することは、第１の重み値セットおよび第１の入力データを用いることを含み、第１の結果は、第１のタスク実行の結果に対応し、かつ第１の結果を計算する間に、メモリの第２の部分に中間結果が記憶されることと、

第１の結果を計算する間に、メモリの第３の部分を識別することであって、第３の部分は、メモリにおける未使用のスペースを含むことと、

第３の部分に第２の重み値セットを記憶することであって、第２の重み値セットは、第２のニューラルネットワーク用であり、第２のニューラルネットワークは、第２のタスクを実行するように構成されることと、

第２のタスクに関連づけられる第２の入力データを受信することと、

ニューラルネットワークプロセッサを用いて第２の結果を計算すること、を含み、第２の結果を計算することは、第２の重みセットおよび第２の入力データを用いることを含み、第２の結果の計算は、第１の結果の計算が完了した時点で発生し、かつ第２の結果は、第２のタスク実行の結果に対応する、コンピュータ実装方法。

６５．メモリの第１の部分のサイズを縮小することをさらに含み、サイズの縮小は、第１の結果の計算の完了に必要とされない第１の重み値セットからの重み値に対応する、条

項６４に記載のコンピュータ実装方法。
６６．メモリの第２の部分のサイズを縮小することをさらに含み、サイズは、中間結果を記憶するために必要なメモリの量に対応する、条項６４または６５に記載のコンピュータ実装方法。
６７．第１の部分、第２の部分および第３の部分は重複しない、請求項６４～６６のいずれかに記載のコンピュータ実装方法。
６８．第１の重み値セットは、第１のニューラルネットワークの全ての重み値より少ない重み値を含み、かつ、
コンピューティングシステムの第２のメモリから、第１のニューラルネットワークの追加の重み値セットを読み取ることと、
追加の重み値セットをメモリの第１の部分に記憶すること、をさらに含み、追加の重み値セットは、第２の重み値セットの記憶に先行して記憶される、条項６４～６７のいずれかに記載のコンピュータ実装方法。
【０１８０】
[0196]他の変形例も、本開示の精神の範囲内である。したがって、開示している技術に対しては、様々な改変および代替構造が可能であるが、図面に示しかつこれまでに詳述したものは、その所定の例示的な実施形態である。しかしながら、本開示を、開示した特定の１つまたは複数の形式に限定する意図はなく、逆に、添付の特許請求の範囲に規定されている本開示の精神および範囲に含まれる全ての改変、代替構造および同等物をカバーすることが意図される点は、理解されるべきである。
【０１８１】
[0197]開示される実施形態を説明する文脈における（特に、添付の特許請求の範囲の文脈における）不定冠詞および定冠詞および類似の指示語の使用は、本明細書における別段の指摘または文脈による明らかな矛盾のない限り、単数および複数の双方を包含するものと解釈されるべきである。「備える」、「有する」、「含む」および「包含する」という用語は、別段の記載のない限り、制約のない（すなわち、「を含むが、これに限定されない」を意味する）用語として解釈されるべきである。「接続される」という用語は、介在する何かが存在するとしても、部分的または完全に、内部に含まれる、付着される、または互いに接合される、と解釈されるべきである。本明細書における値範囲の列挙は、本明細書において別段の指摘のない限り、単に、その範囲に含まれる別個の各値に個々に言及する簡略法として機能することが意図されるものであって、別個の値は各々、本明細書において個々に列挙されたものであるかのように明細書に組み込まれる。本明細書に記述している全ての方法は、本明細書における別段の指摘または文脈による別段の明らかな矛盾のない限り、任意の適切な順序で実行されることが可能である。本明細書において述べる全ての例または例示的な言語（たとえば、「など」）の使用は、単に、本開示の実施形態をよりよく明らかにすることを意図したものであり、別段の主張のない限り、本開示の範囲を限定するものではない。明細書における文言は、いずれも、請求の範囲に記載されていない要素が本開示の実施に不可欠であることを示すものとして解釈されるべきではない。
【０１８２】
[0198]「Ｘ、ＹまたはＺのうちの少なくとも１つ」という言い回しなどの選言的言語の意図は、別段で具体的な明記のない限り、一般的に使用される文脈において、あるアイテム、用語、他がＸ、ＹまたはＺのいずれか、またはこれらの任意の組合せ（たとえば、Ｘ、Ｙおよび／またはＺ）であり得ることを提示するものと理解されることにある。したがって、このような選言的言語は、概して、所定の実施形態が少なくとも１つのＸ、少なくとも１つのＹ、または少なくとも１つのＺを、各々が存在するために必要とすることを含意するものではなく、また含意すべきではない。
【０１８３】
[0199]本明細書には、発明者らが知るところの本開示を実施するための最良の形態を含む、本開示の様々な実施形態が記述されている。これまでの説明を読めば、一般的な当業者には、これらの実施形態の変形が明らかになり得る。本発明者らは、当業者がこのよう

な変形例を適宜使用することを予期するものであり、かつ発明者らの意図は、本開示が本明細書における具体的な記述以外の方法で実施されることにある。したがって、本開示は、適用法で許可される、添付の特許請求の範囲に列挙された主題の全ての変更および同等物を含む。さらに、これまでに述べた要素の、その全ての可能な変形における組合せは、いずれも、本明細書における別段の指摘または文脈による別段の明らかな矛盾のない限り、本開示に包含される。

【図面】

【図１】

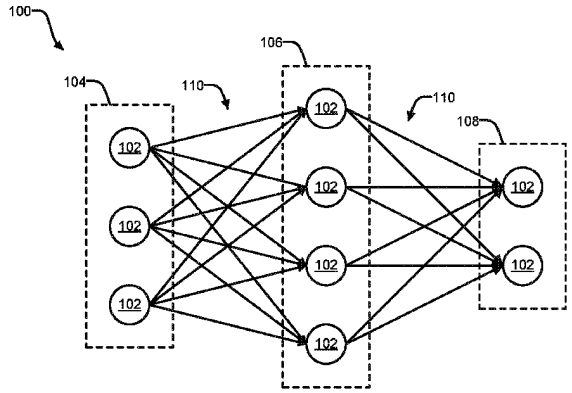

FIG. 1

【図２Ａ】

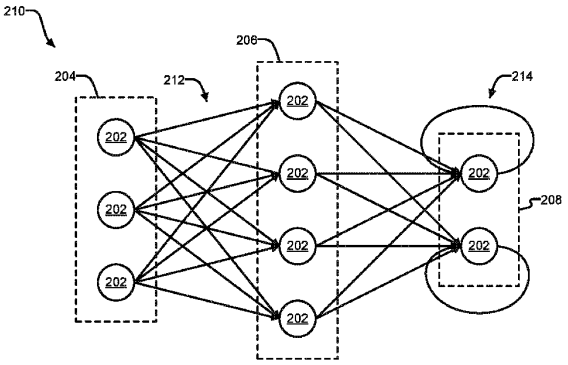

FIG. 2A

【図２Ｂ】

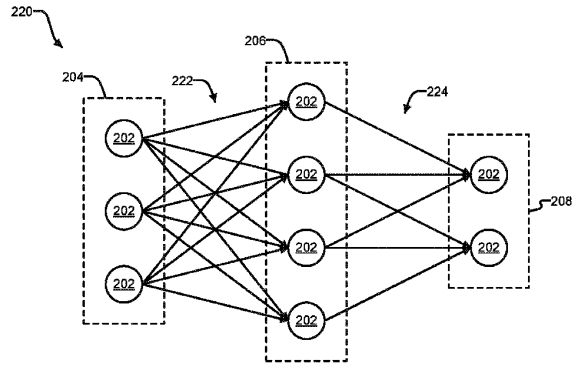

FIG. 2B

【図３Ａ】

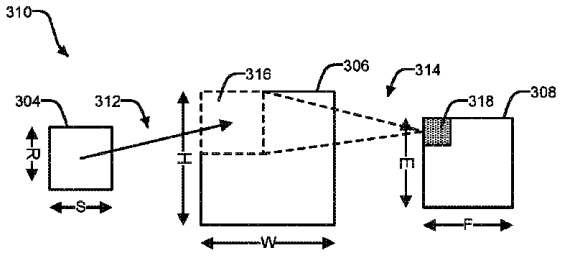

FIG. 3A

【図３Ｂ】

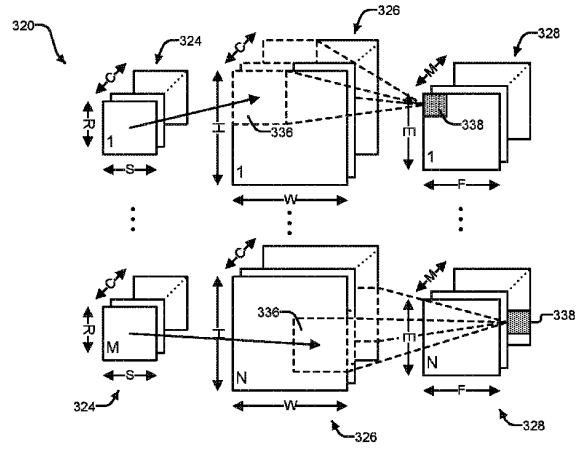

FIG. 3B

【図４】

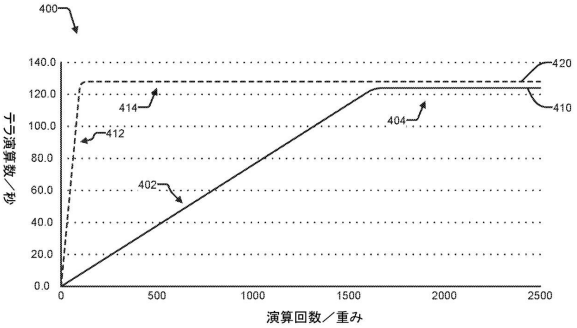

FIG. 4

【図５】

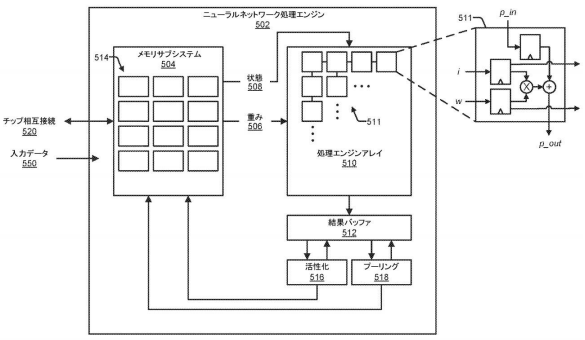

FIG. 5

【図６Ａ】

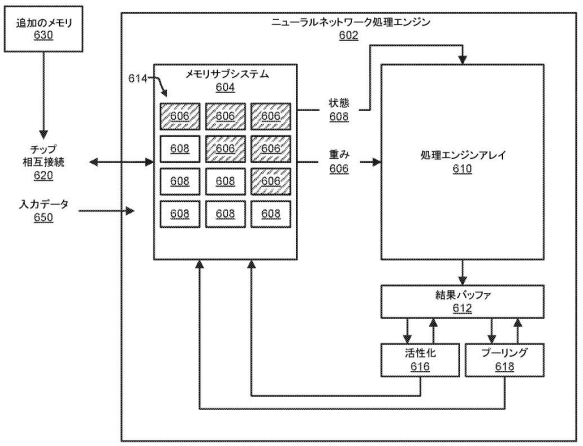

FIG. 6A

【図６Ｂ】

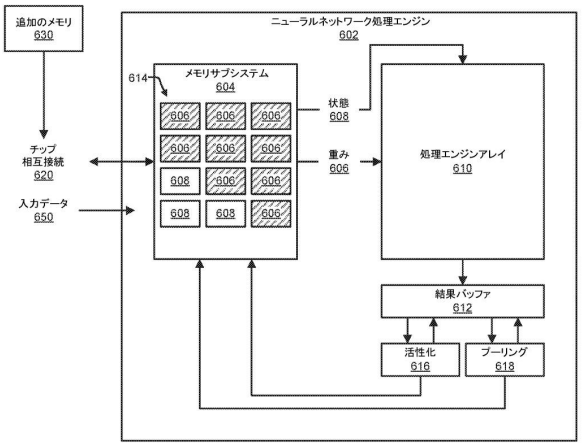

FIG. 6B

【図７Ａ】

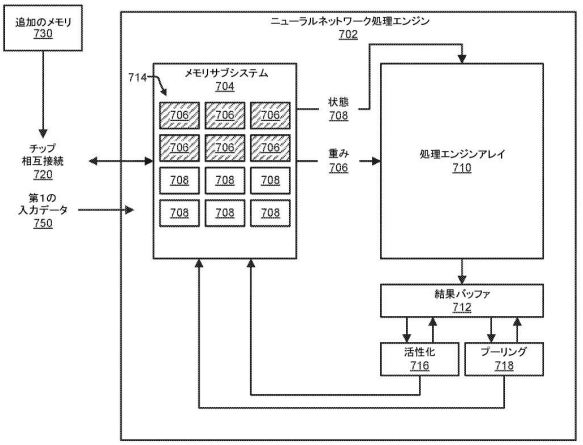

FIG. 7A

【図７Ｂ】

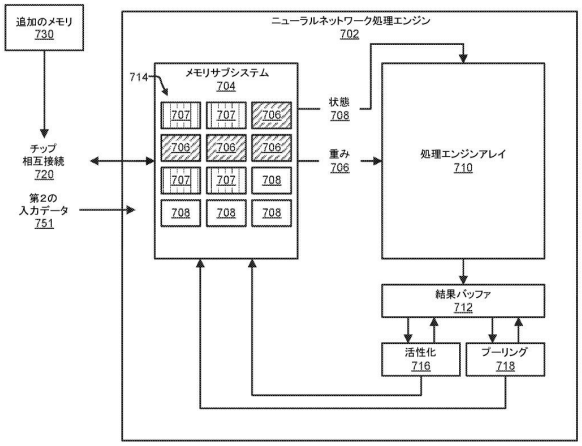

FIG. 7B

【図７Ｃ】

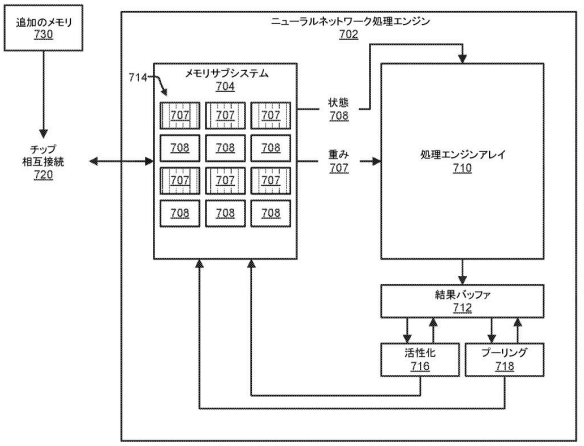

FIG. 7C

【図８】

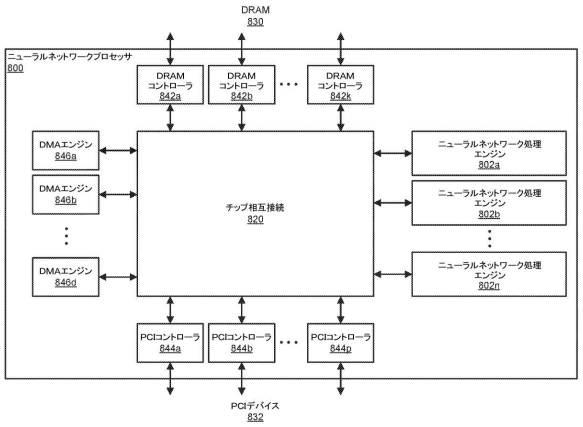

FIG. 8

【図９】

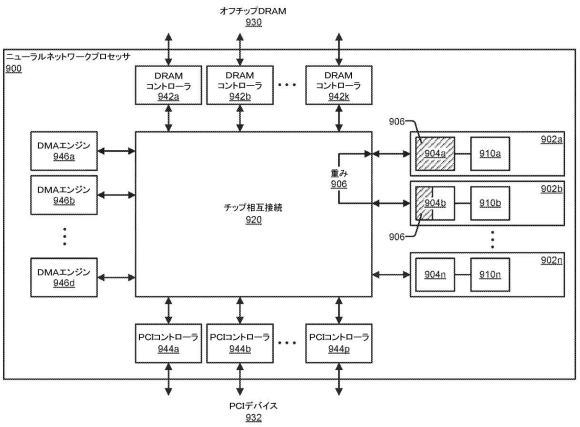

FIG. 9

【図１０】

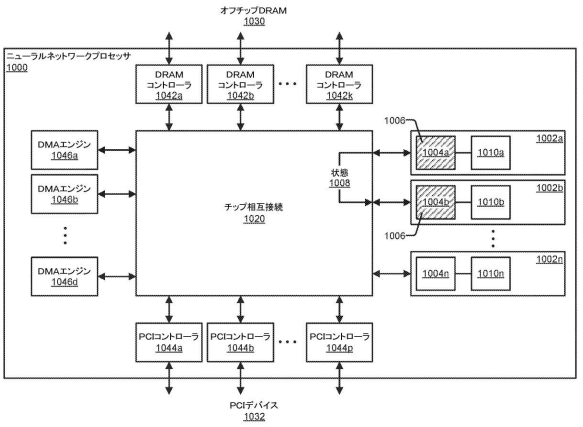

FIG. 10

【図１１】

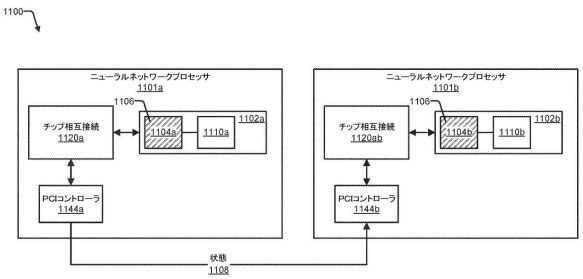

FIG. 11

【図１２】

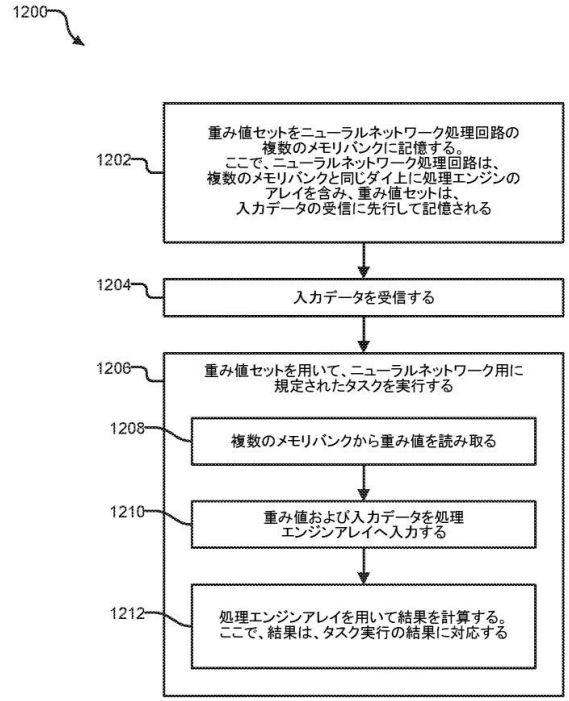

FIG. 12

【図１３】

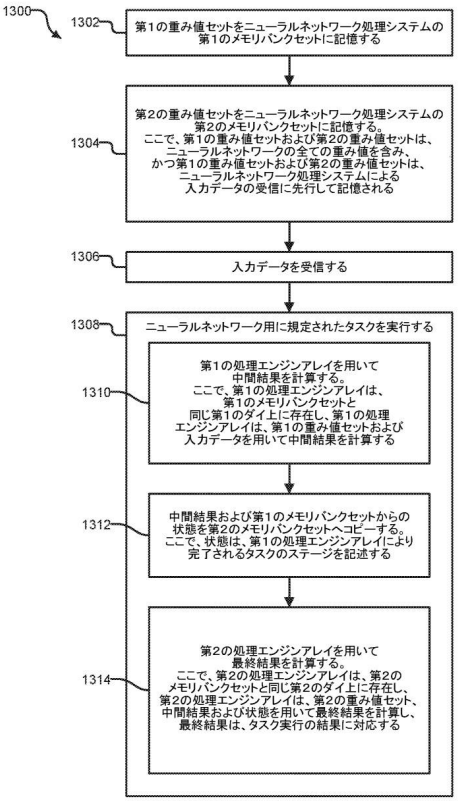

FIG. 13

【図１４】

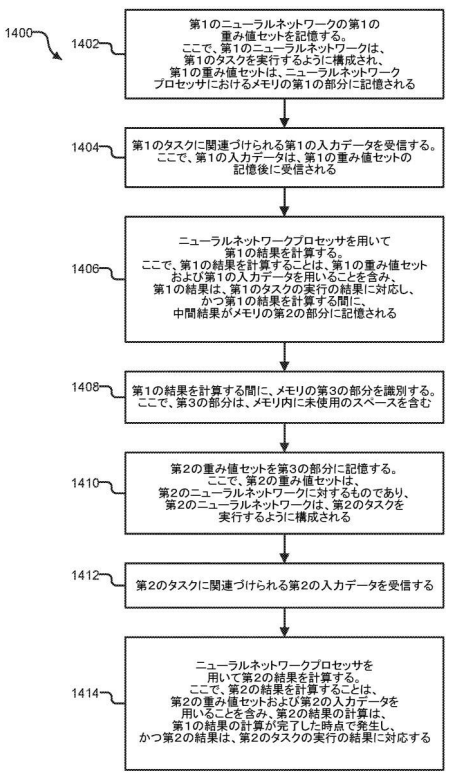

FIG. 14

【図１５】

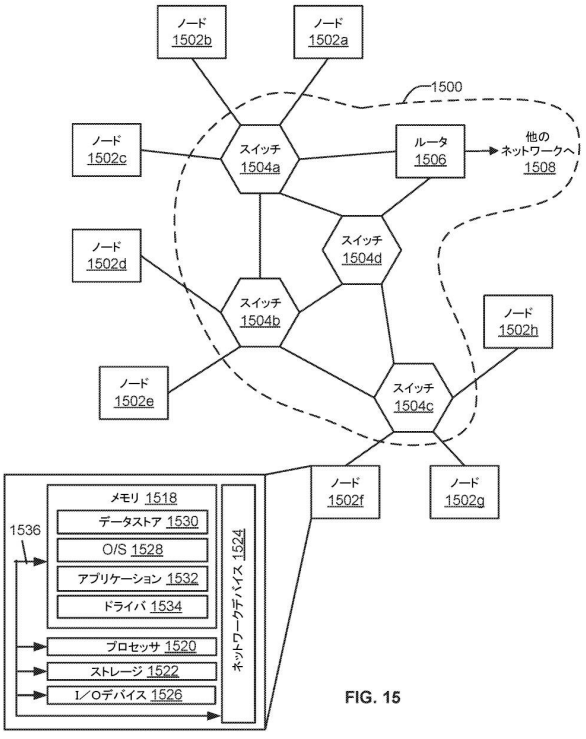

FIG. 15

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号 15/839,157
(32)優先日 平成29年12月12日(2017.12.12)
(33)優先権主張国・地域又は機関
米国(US)

前置審査

．ボックス ８１２２６
(72)発明者 ディアマント，ロン
アメリカ合衆国，ワシントン州 ９８１０８－１２２６，シアトル，ピー．オー．ボックス ８１２２６
(72)発明者 ゼジュダ，ジンドリック
アメリカ合衆国，ワシントン州 ９８１０８－１２２６，シアトル，ピー．オー．ボックス ８１２２６
(72)発明者 ボルコヴィッチ，ドラジェン
アメリカ合衆国，ワシントン州 ９８１０８－１２２６，シアトル，ピー．オー．ボックス ８１２２６

審査官 武田 広太郎
(56)参考文献 特表２０１３－５２９３４２（ＪＰ，Ａ）
特開２００２－１１７３８９（ＪＰ，Ａ）
特開２０１３－１７８２９４（ＪＰ，Ａ）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｎ ３／０６３
Ｇ０６Ｆ １５／８０