

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7036689号

(P7036689)

(45)発行日 令和4年3月15日(2022.3.15)

(24)登録日 令和4年3月7日(2022.3.7)

(51)国際特許分類

G 0 6 N 3/04 (2006.01)

F I

G 0 6 N 3/04

請求項の数 11 (全32頁)

(21)出願番号	特願2018-151323(P2018-151323)	(73)特許権者	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22)出願日	平成30年8月10日(2018.8.10)	(74)代理人	110001678 特許業務法人藤央特許事務所
(65)公開番号	特開2020-27399(P2020-27399A)	(72)発明者	福田 幸二 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
(43)公開日	令和2年2月20日(2020.2.20)	審査官	中村 信也
審査請求日	令和3年3月5日(2021.3.5)		

最終頁に続く

(54)【発明の名称】 計算機システム

(57)【特許請求の範囲】

【請求項1】

グラフ上の畳み込みニューラルネットワークを実行する、計算機システムであって、
1以上のプロセッサと、
1以上の記憶装置と、を含み、
前記グラフ上の畳み込みニューラルネットワークは、
1以上の畳み込み層と、
1以上のプーリング層と、を含み、
前記1以上の記憶装置は、前記1以上の畳み込み層のカーネルの重みデータを格納し、
前記1以上のプロセッサは、
各畳み込み層において、各ノードの値を、所定ホップ数のサイズを有するカーネルに基づく畳み込み演算によって、更新し、
各プーリング層において、各ノードの値を、各ノードの値及び各ノードから所定ホップ数のプーリング範囲内のノードの値に基づくプーリング処理によって、更新し、
プーリング層の後段の畳み込み層のカーネルのサイズは、前記プーリング層の前段の畳み込み層のカーネルのサイズよりも大きい、計算機システム。

【請求項2】

請求項1に記載の計算機システムであって、
畳み込み層の後段のプーリング層のプーリング範囲は、前記畳み込み層の前段のプーリング層のプーリング範囲よりも広い、計算機システム。

【請求項 3】

請求項 1 に記載の計算機システムであって、

前記 1 以上のプロセッサは、前記所定ホップ数のプーリング範囲のプーリング処理を、1 ホップ範囲のプーリング処理を前記所定ホップ数と一致する回数だけ繰り返すことで行う、計算機システム。

【請求項 4】

請求項 1 に記載の計算機システムであって、

前記 1 以上のプロセッサは、前記グラフ上の畳み込みニューラルネットワークの誤差逆伝播学習において、誤差関数に前記 1 以上の畳み込み層の正則化項を含める、計算機システム。

10

【請求項 5】

請求項 1 に記載の計算機システムであって、

前記 1 以上のプロセッサは、前記プーリング処理において、入力値にスムースマキシマムを適用した後に平均値プーリングを行う、計算機システム。

【請求項 6】

請求項 1 に記載の計算機システムであって、

前記 1 以上のプロセッサは、前記 1 以上の畳み込み層及び前記 1 以上のプーリング層の後段において、全域平均プーリングを実行する、計算機システム。

【請求項 7】

請求項 1 に記載の計算機システムであって、

前記計算機システムは、ネットワークにより接続された複数のサブシステムを含み、
前記複数のサブシステムのそれぞれは、前記複数のサブシステムにおいて隣接するサブシステムとのみ通信を行い、

20

前記複数のサブシステムのそれぞれは、前記グラフ上の畳み込みニューラルネットワークにおいて、同一位置のニューロンからなるカラムの計算を行う、計算機システム。

【請求項 8】

請求項 7 に記載の計算機システムであって、

前記複数のサブシステムのそれぞれは、

前記カラムの値を示す内部状態ベクトルを保持し、

隣接サブシステムを含む他のサブシステムから前記内部状態ベクトルを取得し、

30

自内部状態ベクトル及び前記他のサブシステムの内部状態ベクトルを使用して前記カラムの計算を行う、計算機システム。

【請求項 9】

請求項 7 に記載の計算機システムであって、

前記複数のサブシステムは、前記 1 以上の畳み込み層及び前記 1 以上のプーリング層の後段において、分散で前記カラムの値の平均値を計算する、計算機システム。

【請求項 10】

請求項 7 に記載の計算機システムであって、

前記複数のサブシステムのそれぞれは、

他のサブシステムから独立して、誤差逆伝播による学習によって前記 1 以上の畳み込み層の重みを更新し、

40

前記更新した重みの平均値を前記他のサブシステムとの通信により計算する、計算機システム。

【請求項 11】

計算機システムが、グラフ上の畳み込みニューラルネットワークを実行する、方法であって、

前記計算機システムは、

1 以上のプロセッサと、

1 以上の記憶装置と、を含み、

前記グラフ上の畳み込みニューラルネットワークは、

50

1 以上の畳み込み層と、
1 以上のプーリング層と、を含み、
前記 1 以上の記憶装置は、前記 1 以上の畳み込み層のカーネルの重みデータを格納し、
前記方法は、
各畳み込み層において、各ノードの値を、所定ホップ数のサイズを有するカーネルに基づ
く畳み込み演算によって、更新し、
各プーリング層において、各ノードの値を、各ノードの値及び各ノードから所定ホップ数
のプーリング範囲内のノードの値に基づくプーリング処理によって、更新し、
プーリング層の後段の畳み込み層のカーネルのサイズは、前記プーリング層の前段の畳み
込み層のカーネルのサイズよりも大きい、方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、計算機システムに関する。

【背景技術】

【0002】

社会インフラ及び都市等を効率的に設計、運用するため、実社会及びサイバ空間のデー
タを処理し、社会インフラ等の状態を解析し、若しくは予測し、又は、社会インフラ等を制
御し、若しくは誘導する技術が注目されている。処理されるデータは、温度及び湿度等の
環境のセンシングデータ、自動車等の機械のログデータ、並びにメール及び SNS 等の人
間又は組織のログデータから構成される。

20

【0003】

近年、ニューラルネットワークを使用する機械学習が、データ解析する手法として注目さ
れている。ニューラルネットワークは、高速で正確なデータ解析を容易に実現することが
できる。上述のようなシステムのデータを含む様々な種類のデータは、グラフ構造により
表わすことができるグラフデータである。

【0004】

例えば、国際公開第 2016/174725 号（特許文献 1）は、グラフデータのニュー
ラルネットワークを構築得る技術を開示する。具体的には、特許文献 1 は、一つ以上のニ
ューロンを含む複数の層から構成されるニューラルネットワークを用いた演算処理を実行
する計算機であって、複数のノード及び複数のノード間を接続する一つ以上のエッジから
構成されるグラフデータ、及びニューラルネットワークに入力される一つ以上の値を格納
するサンプルデータを格納する記憶部と、グラフデータを用いて、ニューラルネットワ
ークを構築する構築部と、を備え、構築部は、グラフデータに含まれる前記複数のノードに
基づいて、複数の層の各々の一つ以上のニューロンを生成し、グラフデータに含まれる一
つ以上のエッジに基づいて、複数の層の各々に含まれる一つ以上のニューロン間の接続を
生成することによってニューラルネットワークを構築する、ことを開示する（例えば要約
）。また、非特許文献 1 は、関係グラフ畳み込みネットワークの例を開示する。

30

【先行技術文献】

【特許文献】

40

【0005】

【文献】国際公開第 2016/174725 号

【非特許文献】

【0006】

【文献】M. Schlichtkrull et al., “Modeling Relational Data with Graph Convolutional Networks”, arXiv preprint arXiv:1703.06103, 2017.

【発明の概要】

【発明が解決しようとする課題】

【0007】

様々な種類のニューラルネットワークが知られている中で、畳み込みニューラルネットワ

50

ーク（CNN）は、特に画像処理に有用なニューラルネットワークとして知られている。グラフデータは関係性を有しており、グラフデータをCNNにより解析することができれば有用である。

【0008】

CNNを深いネットワークで構成するいわゆる深層学習の実現には、プーリング、又は、より一般にパラメータの自由度を上層に行くほど小さくする操作が必要である。プーリング層なしで畳み込み層だけを多数重ねると、表現可能な自由度が減らない（正則化がかからない）ままパラメータ数のみが増えることとなり、パラメータの学習時に過学習を引き起こす。

【0009】

従来のグリッドデータのCNN（グリッド上のCNN）は、畳み込み演算（及び非線形活性化関数の適用）の後に、プーリングと呼ばれる操作で画像サイズを縮小することが一般的である。しかし、グラフデータのCNNにおいて、プーリング操作の実現は困難である。

【0010】

グリッドデータにおけるプーリング操作を単純にグラフデータに対応させると、適当にグラフデータのノードを削除してグラフデータのサイズを縮小すればよいと考えられる。しかしながら、どのようにして削除するノードと残すノードを選ぶのかの指針が明らかでない。

【0011】

したがって、グラフ上のCNNにおいてパラメータの自由度を低減するプーリングを実現する技術が望まれる。

【課題を解決するための手段】

【0012】

本開示の一態様は、グラフ上の畳み込みニューラルネットワークを実行する、計算機システムであって、1以上のプロセッサと、1以上の記憶装置と、を含み、前記グラフ上の畳み込みニューラルネットワークは、1以上の畳み込み層と、1以上のプーリング層と、を含み、前記1以上の記憶装置は、前記1以上の畳み込み層のカーネルの重みデータを格納し、前記1以上のプロセッサは、各畳み込み層において、各ノードの値を、所定ホップ数のサイズを有するカーネルに基づく畳み込み演算によって、更新し、各プーリング層において、各ノードの値を、各ノードの値及び各ノードから所定ホップ数のプーリング範囲内のノードの値に基づくプーリング処理によって、更新し、プーリング層の後段の畳み込み層のカーネルのサイズは、前段の畳み込み層のカーネルのサイズよりも大きい。

【発明の効果】

【0013】

本開示の一態様によれば、グラフ上の畳み込み演算において適切にプーリング処理を行うことができる。

【図面の簡単な説明】

【0014】

【図1】計算機システムの構成の一例を示す。

【図2】ニューラルネットワークの一例を示す。

【図3】畳み込みニューラルネットワークの例を示す。

【図4】グラフ上のCNNの構成例を示す。

【図5】グラフデータの例を示す。

【図6】一つのノード及び当該ノードに接続されているノードを示す。

【図7】重みデータの一部であって、ノードの畳み込みに関する値を示す。

【図8】ノードの状態値の更新を模式的に示す。

【図9】グラフ上のCNNの第1畳み込み層、第1プーリング層、第2畳み込み層及び第2プーリング層詳細を示す。

【図10】グラフ上のCNNに入力されるグラフデータの例を示す。

【図11】第1畳み込み層の畳み込み演算の例を示す。

10

20

30

40

50

【図 1 2】グラフデータの属性の例を示す。

【図 1 3】本例の重みデータの一部のデータを示す。

【図 1 4】第 1 畳み込み層によりエッジの状態値の更新を示す。

【図 1 5】第 1 プーリング層によるプーリング処理を模式的に示す。

【図 1 6】図 1 5 に示すグラフ構造を有するグラフデータにおいて、適用可能なラブラシアン行例の例を示す。

【図 1 7】第 2 畳み込み層による、ノードのための畳み込み演算を模式的に示す。

【図 1 8】計算機が実行する学習処理を説明するフローチャートである。

【図 1 9】分散 CNN を実行する計算機システムの例を示す。

【図 2 0】分散 CNN の層及びカラムの例を示す。

10

【図 2 1】分散 CNN の概念的な構成例を示す。

【図 2 2】関数 f の適用ノードの伝播則を示す。

【図 2 3】可算ノードの伝播則を示す。

【図 2 4】乗算ノードの伝播則を示す。

【図 2 5】変数複製ノード（分岐ノード）の伝播則を示す。

【図 2 6】分散 CNN の構成例及びその処理を示す。

【発明を実施するための形態】

【0015】

以下、添付図面を参照して本発明の実施形態を説明する。本実施形態は本発明を実現するための一例に過ぎず、本発明の技術的範囲を限定するものではないことに注意すべきである。各図において共通の構成については同一の参照符号が付されている。

20

【実施例 1】

【0016】

[システム構成]

図 1 は、計算機システムの構成の一例を示す。図 1 に示すように、計算機システム 100 は、複数の計算機 101、及びストレージシステム 102 から構成され、複数の計算機 101 とストレージシステム 102 とは、互いに通信できるように、ネットワーク 103 を介して接続される。本実施例の計算機システム 100 は、3 つの計算機 101 を含む。図 1 は、一つの計算機のみを例として符号 100 で指示する。なお、計算機及びストレージシステムの数は、任意である。計算機システム 100 は、一つの計算機 101 で構成されていてもよい。

30

【0017】

ネットワーク 103 は、WAN (Wide Area Network)、LAN (Local Area Network)、及び SAN (Storage Area Network) 等が考えられる。なお、ネットワーク 103 の種別は限定されない。計算機 101 それぞれとストレージシステム 102 接続するネットワークと、計算機 101 それぞれを接続するネットワークは、異なるネットワークであってもよい。

【0018】

計算機 101 は、プロセッサ 110、メモリ 111、通信インタフェース 112、及び入出力インタフェース 118 を含み、各構成はバス 114 を介して互いに接続される。また、入出力装置 119 は、入出力インタフェース 118 を介して他の装置と接続される。プロセッサ 110 は、演算処理を実行する 1 つ以上の CPU 115 を含む。図 1 において、一つの CPU のみが符号 115 で指示されている。CPU 115 は、メモリ 111 に格納されるプログラムを実行することによって、計算機 101 が有する機能を実現する。また、計算機 101 上で実行される処理は 1 つ以上の CPU 115 によって実行される。なお、1 つの CPU 115 が複数の処理を実行してもよい。なお、CPU 115 は、FPGA 又は GPU 等の演算器であってもよい。

40

【0019】

メモリ 111 は、プロセッサ 110 が実行するプログラム及び当該プログラムによって使用される情報を格納する。また、メモリ 111 は、プロセッサ 110 が実行する 1 つの処

50

理に対して割り当てられるメモリ空間を含む。なお、当該メモリ空間は、複数のメモリ 1 1 1 のメモリ領域上に確保されてもよいし、また、1 つのメモリ 1 1 1 のメモリ領域上に確保されてもよい。また、メモリ 1 1 1 は、複数の処理のメモリ空間を含んでもよい。メモリ 1 1 1 に格納されるプログラム及び情報については後述する。

【0020】

通信インタフェース 1 1 2 は、ネットワーク 1 0 3 を介して外部装置と通信する。プロセッサ 1 1 0 は、通信インタフェース 1 1 2 を介して、他の装置にアクセスする。入出力インタフェース 1 1 8 は、ネットワーク 1 0 3 を介して、他の装置と入出力装置 1 1 9 との通信を媒介する。

【0021】

ストレージシステム 1 0 2 は、ディスクインタフェース 1 3 3、及び複数の記憶デバイス 1 1 7 を含み、各装置はバス 1 3 5 を介して互いに接続される。ディスクインタフェース 1 3 3 は、複数の記憶ドライブ 1 3 4 と接続するためのインタフェースである。記憶ドライブ 1 3 4 は、各種データを格納する記憶装置であり、例えば、HDD や SSD である。図 1 において、一つの記憶ドライブのみが符号 1 3 4 で指示されている。

【0022】

メモリ 1 1 1 は、データ処理部 1 2 0 を実現するプログラムを格納する。データ処理部 1 2 0 を実行するプロセッサ 1 1 0 は、逆誤差伝播により学習処理、及び順伝播による分析処理を実行する。学習処理では、プロセッサ 1 1 0 は、学習データ 1 2 2 を使用して、構築したニューラルネットワーク内のエッジの重みを決定する。分析処理では、プロセッサ 1 1 0 は、分析対象であるデータを、構築されたニューラルネットワークに入力することによって、分類や回帰等の所定の分析を行う。

【0023】

なお、データ処理部 1 2 0 は、複数のプログラムモジュールから構成されてもよい。例えば、データ処理部 1 2 0 は、ニューラルネットワークを構築する構築モジュール、学習処理を実行する学習処理モジュール、及び分析処理を実行する分析処理モジュールを含んでもよい。また、それぞれのプログラムモジュールを別々の計算機 1 0 1 が実行してもよい。

【0024】

[ニューラルネットワーク]

図 2 は、ニューラルネットワークの一例を示す。図 2 に示すニューラルネットワーク 2 0 0 は、入力層 2 0 1、中間層 2 0 2、及び出力層 2 0 3 の 3 層から構成される。各層は 1 つ以上のニューロン 2 1 1 から構成される。また、各層のニューロン 2 1 1 は、他の層の少なくとも 1 つのニューロン 2 1 1 と接続する。具体的には、入力層 2 0 1 のニューロン 2 1 1 は、中間層 2 0 2 の少なくとも 1 つのニューロン 2 1 1 と接続し、また、中間層 2 0 2 のニューロン 2 1 1 は、出力層 2 0 3 の少なくとも 1 つのニューロン 2 1 1 と接続する。入力層 2 0 1 の側が前段側であり、出力層 2 0 3 の側が後段側である。

【0025】

エッジ 2 1 2 はニューロン 2 1 1 間のデータの出力を表す。ニューラルネットワーク 2 0 0 は、分析処理において、入力されたデータを入力層 2 0 1 から、中間層 2 0 2 を介して、出力層 2 0 3 に順伝播する。各ニューロン 2 1 1 に入力された値は、エッジに割り当てられている重みと掛け算され、エッジ 2 1 2 によって接続される他の層のニューロン 2 1 1 に出力される。入力 2 1 1 への入力はバイアスを含んでもよい。重みは、ニューロン 2 1 1 間の接続の強さを表し、後述する学習処理によって決定される。学習処理は、損失信号の誤差逆伝播により各層の重みが更新される。

【0026】

図 1 に戻って、メモリ 1 1 1 は、グラフデータ 1 2 1、学習データ 1 2 2、及び重みデータ 1 2 3 を格納する。グラフデータ 1 2 1 は、任意の要素に対応するノード、及びノード間を接続するエッジから構成されるグラフ構造を有するデータである。学習データ 1 2 2 は、ニューラルネットワークを用いた学習処理において用いられるデータである。重みデータ 1 2 3 は、ニューラルネットワークの学習処理の処理結果である重みを管理する情報

10

20

30

40

50

である。

【 0 0 2 7 】

グラフデータ 1 2 1 及び学習データ 1 2 2 は、例えば、ストレージシステム 1 0 2 に格納されており、プロセッサ 1 1 0 が、ストレージシステム 1 0 2 からグラフデータ 1 2 1 及び学習データ 1 2 2 を取得し、取得されたグラフデータ 1 2 1 及び学習データ 1 2 2 をメモリ 1 1 1 にロードする。

【 0 0 2 8 】

[グリッド上の C N N]

図 3 は、畳み込みニューラルネットワーク (C N N) の例を示す。本開示の計算機システム 1 0 0 は、グラフ上の C N N を実行する。図 3 を参照して、典型的なグリッド上の C N N 3 0 0 を説明する。グリッド上の C N N は、画像データのようなマトリックス状 (グリッド状) の入力データを処理する。

【 0 0 2 9 】

図 3 の例において、 28×28 のグレースケールの手書きの数字の画像 3 0 1 が入力される。C N N 3 0 0 は、 5×5 の畳み込みフィルタ及び R e L U 関数を適用して (3 0 2) 、6 つの 24×24 の特徴マップ 3 0 3 を生成する。C N N 3 0 0 は、これら特徴マップに 2×2 の最大値プーリングを適用して (3 0 4) 、画像サイズを半分に縮小された特徴マップ 3 0 5 を生成する。

【 0 0 3 0 】

C N N 3 0 0 は、さらに、 5×5 の畳み込みフィルタ及び R e L U 関数を実行して (3 0 6) 、1 0 の 8×8 の特徴マップ 3 0 7 を生成する。C N N 3 0 0 は、これら特徴マップに 2×2 の最大値プーリングを適用して (3 0 8) 、画像サイズを半分に縮小された特徴マップ 3 0 9 を生成する。その後、2 層の全結合層よりなるパーセプトロン 3 1 0 が、最終的に 1 0 次元のベクトルを出力する。出力された 1 0 次元ベクトルは S o f t M a x 関数によりワンホットベクトル表現に変換され、入力画像が 0 ~ 9 までのどの数字であるかの尤もらしさを表す。

【 0 0 3 1 】

[グラフ上の C N N (順伝播)]

以下において、本開示のグラフ上の C N N について説明する。計算機システム 1 0 0 は、グラフ上の C N N を実行する。図 4 は、グラフ上の C N N の構成例 4 0 0 を示す。図 4 は、C N N 4 0 0 における順伝播 (分析処理) を示し、データ処理部 1 2 0 により実行される。グラフ上の C N N 4 0 0 は、各層において、共通のグラフ構造上でデータを処理する。

【 0 0 3 2 】

グラフ構造は、複数のノードと、それぞれ二つノードを接続する複数のエッジで構成されている。グリッド構造はグラフ構造の一つであり、各ノードが隣接するノードそれぞれとエッジで接続されている。本開示におけるグラフ構造は、任意の複数ノードと、任意数のエッジとで構成される。一つのノードは、1 以上の任意の数のエッジに接続される。

【 0 0 3 3 】

グラフデータは、グラフ構造を有するデータであって、ノードそれぞれに状態値が割り当てられている。状態値は、1 次元以上のベクトルで表わすことができる。図 4 に示すように、グラフデータ 1 2 1 が、グラフ上の C N N 4 0 0 に入力される。上述のように、グラフデータ 1 2 1 の各ノードは、状態値を有しており、ノード間のエッジが定義されている。

【 0 0 3 4 】

図 4 の C N N 4 0 0 は、図 3 を参照して説明したグリッド上の C N N 3 0 0 のように、2 つの畳み込み層と 2 つのプーリング層を含む。具体的には、C N N 4 0 0 は、第 1 畳み込み層 4 1 1、第 1 プーリング層 4 1 2、第 2 畳み込み層 4 1 3、第 2 プーリング層 4 1 4 及び全域平均プーリング (G l o b a l A v e r a g e P o o l i n g : G A P) 層 4 1 5 を含む。第 1 畳み込み層 4 1 1 は、グラフデータ 1 2 1 のグラフ構造上で、入力されたグラフデータ 1 2 1 の畳み込み演算を行う。これにより、2 つの特徴マップ 4 0 2 が生成される。

10

20

30

40

50

【 0 0 3 5 】

第 1 プーリング層 4 1 2、グラフデータ 1 2 1 のグラフ構造上で、特徴マップ 4 0 2 それぞれにプーリング処理を実行し、表現の自由度が低下した特徴マップ 4 0 3 に変換する。第 2 畳み込み層 4 1 3 は、グラフデータ 1 2 1 のグラフ構造上で、特徴マップ 4 0 3 の畳み込み演算を行う。これにより、3 つの特徴マップ 4 0 4 が生成される。

【 0 0 3 6 】

第 2 プーリング層 4 1 4、グラフデータ 1 2 1 のグラフ構造上で、特徴マップ 4 0 4 それぞれにプーリング処理を実行し、表現の自由度が低下した特徴マップ 4 0 5 に変換する。特徴マップ 4 0 5 は全域平均プーリング層 4 1 5 に入力される。全域平均プーリング層 4 1 5 は、チャンネル毎に全てのニューロンの値の平均値をとる。

10

【 0 0 3 7 】

畳み込み層及びプーリング層の処理の詳細は後述するが、畳み込み層及びプーリング層は、それぞれ、グラフデータの各ノードの状態値を更新する。したがって、各層に入力されたデータのグラフ構造は、出力データにおいて維持されている。具体的には、入力グラフデータ 1 2 1、特徴マップ 4 0 2 ~ 4 0 5 のグラフ構造は共通である。各層において、ノードそれぞれに対応するニューロンが存在し、ニューロンは、対応するノードの新たな値を出力する。なお、他の例において、1 又は複数の層における入力データのグラフ構造が出力データのグラフ構造と異なってもよい。

【 0 0 3 8 】

図 5 は、グラフデータ 1 2 1 の例を示す。図 5 に示すグラフデータ 1 2 1 は、エッジの構造情報 5 0 0 及びノードの構造情報 5 1 0 を含む。エッジの構造情報 5 0 0 は、エッジ ID 5 0 1、ノード ID 5 0 2、及びエッジ属性 5 0 3 を含む。エッジ ID 5 0 1 は、グラフデータのエッジそれぞれの ID を示す。ノード ID 5 0 2 は、グラフデータそれぞれのノードの ID を示す。エッジ属性 5 0 3 は、エッジそれぞれに付与されている属性情報を示す。エッジ属性 5 0 3 は、エッジの向き及び種類の情報を含む。

20

【 0 0 3 9 】

ノードの構造情報 5 1 0 は、ノード ID 5 1 1、エッジ ID (OUT) 5 1 2、エッジ ID (IN) 5 1 3、及びノード属性 5 1 4 を含む。ノード ID 5 1 1 は、ノード ID 5 0 2 と同一のである。エッジ ID (OUT) 5 1 2 は、ノードから流出するエッジの ID を示す。ノードは、エッジ ID (OUT) 5 1 2 が示すエッジの始点ノードである。エッジ ID (OUT) 5 1 3 は、ノード ID 5 1 1 が示すノードに流入するエッジの ID を示す。ノードは、エッジ ID (IN) 5 1 3 が示すエッジの終点である。ノード属性 5 1 4 は、ノードの属性情報を示す。ノード属性 5 1 4 は、ノードの種類の情報を含む。

30

【 0 0 4 0 】

様々な種類の情報がグラフデータで表わすことができ、例えば、地図情報、分子構造情報、ソーシャルネットワークの情報、交通網の情報などが、グラフデータで表わすことができる。例えば、地図の位置がノードで表わされ、道路がエッジで表わされる。土地を表すノードの属性情報は、不動産価値や、駅又は公園等の土地の利用形態などを含む。また、道路を表すエッジの属性情報は、路線名、道路幅、接続する土地 (ノード) の属性、向きなどを含む。土地及び道路の属性情報に応じて、土地及び道路それぞれに属性種類が定義される。例えば、グラフ CNN を既存の不動産価値を表す属性情報を教師データとして学習することで、与えられた土地の不動産価値の推定を行うことができる。

40

【 0 0 4 1 】

グラフ上の畳み込み演算には、大きく分けて、グラフフーリエ変換を用いる手法と、Relational Graph Convolution Network (R-GCN) と呼ばれるより直接的な手法がある。このうち、グラフフーリエ変換を用いる手法は、理論的な背景が明らかという利点があるが、重複エッジのない単純な無向グラフにしか適用することができないという欠点がある。

【 0 0 4 2 】

一方、R-GCN は、グリッド上の畳み込み演算の直感的な拡張になっており、有向グラ

50

フエッジ、又はより一般的にノードやエッジに属性が付与されている場合でも自然に定義することが可能である。本開示の特徴は、いずれの手法にも適用することができるが、以下において、R - G C Nの例を説明する。

【 0 0 4 3 】

[R - G C Nの例]

図 6、7 及び 8 を参照して、R - G C Nの公知の例を説明する。畳み込み演算は、ノードの状態値を、当該ノード及び周囲の他のノードの状態値に基づき、更新する。以下においては、一つのノードの状態値を更新する処理の例を説明する。図 6 は一つのノード 6 0 0 及び当該ノード 6 0 0 に接続されているノード 6 1 1 ~ 6 1 4、6 2 1 ~ 6 2 3 を示す。ノード 6 0 0 及びノード 6 1 1 ~ 6 1 4 それぞれを接続するエッジは属性 R 1 を有する。ノード 6 0 0 及びノード 6 2 1 ~ 6 2 3 それぞれを接続するエッジは属性 R 2 を有する。

10

【 0 0 4 4 】

図 7 は、重みデータ 1 2 3 の一部であって、ノード 6 0 0 の畳み込みに関する値を示す。重みデータ 1 2 3 は、エッジ属性種類 6 0 1 と重み 6 0 2 を含む。エッジ属性種類 6 0 1 は、エッジ属性の種類の識別子を示す。本例において、エッジ属性種類 6 0 1 とグラフデータ 1 2 1 のエッジ属性 5 0 3 で示される種類とは異なる。これらは同一であってもよい。重み 6 0 2 は、エッジ属性種類それぞれの重みを示す。重み 6 0 2 が示す値は、C N N 4 0 0 の学習（訓練）において更新される。なお、本例のノード 6 0 0 の状態値の畳み込み演算のバイアスは 0 とする。

【 0 0 4 5 】

図 6 及び 7 に示すように、エッジの属性種類は、エッジの属性及びノード 6 0 0 に対するエッジの方向により定義されている。具体的には、ノード 6 1 2 ~ 6 1 4 とノード 6 0 0 とを接続するエッジのエッジ属性種類は、R 1 (I N) である。ノード 6 1 1 とノード 6 0 0 とを接続するエッジのエッジ属性種類は、R 1 (O U T) である。ノード 6 2 2 及び 6 2 3 とノード 6 0 0 とを接続するエッジのエッジ属性種類は、R 2 (I N) である。ノード 6 2 1 とノード 6 0 0 とを接続するエッジのエッジ属性種類は、R 2 (O U T) である。エッジ 6 3 1 の始点及び終点はノード 6 0 0 のセルフループであり、そのエッジ属性種類は、S E L F である。

20

【 0 0 4 6 】

図 8 は、ノード 6 0 0 の状態値の更新を模式的に示す。畳み込み演算は、まず、同一エッジ属性種類のノードの状態値の総和を計算する。具体的には、以下の通りである。ノード 6 1 2 ~ 6 1 4 のエッジ属性種類は R 1 (I N) である。畳み込み演算は、ノード 6 1 2 ~ 6 1 4 の状態値 8 0 1 の総和 8 1 1 を計算する。ノード 6 1 1 のエッジ属性種類は R 1 (O U T) である。畳み込み演算は、ノード 6 1 1 の状態値 8 0 2 の総和 8 1 2 を計算する。

30

【 0 0 4 7 】

ノード 6 2 2 及び 6 2 3 のエッジ属性種類は R 2 (I N) である。畳み込み演算は、ノード 6 2 2 及び 6 2 3 の状態値 8 0 3 の総和 8 1 3 を計算する。ノード 6 2 1 のエッジ属性種類は R 2 (O U T) である。畳み込み演算は、ノード 6 2 1 の状態値 8 0 4 の総和 8 1 4 を計算する。畳み込み演算は、セルフループエッジ 6 3 1 に対して、ノード 6 0 0 の状態値 8 0 5 の総和 8 1 5 を計算する。

40

【 0 0 4 8 】

次に、畳み込み演算は、エッジ属性種類の総和と対応する重みの積和を計算する。エッジ属性種類の重みは、図 7 の重みデータ 1 2 3 に示されている。つまり、畳み込み演算は、エッジ属性種類の総和それぞれに、重みデータ 1 2 3 が示す重みを乗算し、それらの総和を計算する（8 2 1）。畳み込み演算は、この積和に R e L U 関数 8 2 2 を適用して得られる状態値 8 3 1 を出力する。

【 0 0 4 9 】

図 6 ~ 8 を参照して説明したグラフ上の C N N の例は、対象ノードの状態値と、当該対象ノードに隣接するノード状態値とから、当該対象ノードの状態値の更新値を決定する。隣

50

接するノードは、対象ノードから1ホップのノードである。ホップ数は、ノード間の距離を表し、ノード間のエッジの数を示す。一つのノードから他のノードまで複数の経路が存在する場合、それぞれの経路のホップ数が定義される。

【0050】

[グラフ上のCNNの各層の処理]

図9は、グラフ上のCNN400の第1畳み込み層411、第1プーリング層412、第2畳み込み層413及び第2プーリング層414の詳細を示す。第1畳み込み層411は、各ノードの状態値の更新値を、当該ノード及び1ホップ内のノードの状態値から、ReLU関数を使用して計算する。第1プーリング層412は、1ホップ内でのプーリング処理を実行する。プーリング処理の詳細は後述する。

10

【0051】

第2畳み込み層413は、各ノードの状態値の更新値を、当該ノード及び2ホップ内のノードの状態値から、ReLU関数を使用して計算する。第2プーリング層414は、2ホップ内でのプーリング処理を実行する。2ホップ内でのプーリング処理を実行するかわりに、1ホッププーリングを2回実行してもよい。

【0052】

図9に示すように、第2畳み込み層413の畳み込み演算の範囲は、第1畳み込み層411の畳み込み演算の範囲よりも広い。また、第1プーリング層412が1ホップ内でのプーリング処理を実行することに対して、第2プーリング層414は、2ホップ内でのプーリング処理を実行する。これは、第2プーリング層414のプーリング処理の範囲が、第1プーリング層412のプーリング処理の範囲よりも広いことを意味する。

20

【0053】

図3を参照して説明したように、従来のグリッド上のCNN300は、畳み込み演算（畳み込み演算＋非線形活性化関数の適用）の後にプーリング処理（通常は最大値プーリング処理）を実行することで、画像サイズを縮小する。これに対して、本例のグラフ上のCNN400のプーリング層412及び414は、グラフ構造を縮小することなくプーリング処理を実行する。後述するように、グラフ上のCNN400におけるプーリング処理は、グラフのサイズを一定に保つ一方で、近隣のノード間で値を近づけるといふ正則化をとり入れることで、実質的にプーリングを行う。

【0054】

第1プーリング層412の直後の第2畳み込み層413は、プーリング層で近隣のノードの値が近づいてしまうため、広い範囲を見る、すなわち、畳み込みフィルタのカーネルサイズを広げる。これにより、プーリング処理後の畳み込み演算がより有効となる。第2畳み込み層413の後のプーリング層414は、第2畳み込み層413の畳み込みフィルタのカーネルサイズが広げられたため、プーリング範囲を広げることで、より有効なプーリング処理を実現する。

30

【0055】

図9に示す例において、第1畳み込み層411の範囲は1ホップ内であるが、第1畳み込み層411の範囲はより広くてもよい。第2畳み込み層413の範囲は、第1畳み込み層411の範囲より広ければ、それらの関係は限定されない。例えば、第2畳み込み層413の範囲は、3ホップ以上であってもよい。

40

【0056】

グラフ上のCNN400の例は、それぞれ3増以上の畳み込み層及びプーリング層を含むことができる。例えば、畳み込み層の畳み込み演算の範囲（ホップ数）は、例えば、一つ前の畳み込み層の定数倍である。ホップ数2倍の例において、畳み込み層の畳み込み演算の範囲は、1ホップ、2ホップ、4ホップ、8ホップと増加する。ホップ数1.5倍の例において、畳み込み層の畳み込み演算の範囲は、1ホップ、2ホップ、3ホップ、4ホップ、6ホップと増加する。プーリング層のプーリング処理の範囲（ホップ数）は、例えば、直前の畳み込み層のホップ数と一致させる。

【0057】

50

以下において、グラフ上のCNN400の処理の例を示す。図10は、グラフ上のCNN400に入力されるグラフデータ121の例を示す。グラフデータ121は、ノードN1～N13と、それぞれノード間を接続する複数のエッジを含む。図10は明示していないが、ノードN1～N13のそれぞれに状態値が定義されており、また、エッジのそれぞれにエッジ属性種類が定義されている。

【0058】

[第1畳み込み層]

図11は、第1畳み込み層411の畳み込み演算の例を示す。図11は、ノードN1の状態値の更新の例を示す。第1畳み込み層411は、1ホップ内で畳み込み演算を実行する。ノードN1から1ホップ内のノード、つまり、ノードN1に一つのエッジで接続されているノードは、ノードN2、N6、N9、N10及びN12である。第1畳み込み層411は、ノードN2、N6、N9、N10及びN12の状態値を集め、それら状態値とノードN1の状態値とから、ノードN1の更新状態値を計算する。

【0059】

図12は、グラフデータ121の属性の例を示す。グラフデータのエッジそれぞれにエッジ属性種類が定義されており、Pと数字からなる符号で指示されている。例えば、ノードN1とノードN12の間のエッジのエッジ属性種類はR1であり、ノードN1とノードN2の間のエッジのエッジ属性種類はR2であり、ノードN1とノードN6の間のエッジのエッジ属性種類はR3である。エッジ属性種類R1のエッジは実線で表わされ、エッジ属性種類R2のエッジは破線で表わされ、エッジ属性種類R3のエッジは2点鎖線で表わされている。

【0060】

図13は、本例の重みデータ123の一部のデータを示す。重みデータ123は、対象ノードの現在状態値の重みW0及び1ホップのエッジ属性種類の重みを定義すると共に、2ホップのエッジ属性種類の重みを定義する。1ホップのエッジ属性種類は、エッジそれぞれのエッジ属性種類R1、R2及びR3と一致する。2ホップのエッジ属性種類は、2つのPと2つの数字で表わされており、二つのエッジのエッジ属性種類の組を示す。

【0061】

より具体的には、2ホップのエッジ属性種類は、対象ノードに近いエッジのエッジ属性種類と対象ノードから遠いエッジのエッジ属性種類を示す。例えば、エッジ属性種類R1R2は、対象ノードと隣接ノードの間のエッジのエッジ属性種類がR1であり、対象ノードの隣接ノードと2ホップ離れたノードとの間のエッジのエッジ属性種類がR2であることを表す。

【0062】

図14は、第1畳み込み層411によりノードN1の状態値の更新を示す。第1畳み込み層411は、対象ノードN1の状態値V1及び、対象ノードN1の1ホップ内のノード、つまり隣接ノードN2、N6、N9、N10、N12の状態値V2、V6、V9、V10、V12を取得する。対象ノードN1と、隣接ノードN2、N6、N9、N10、N12それぞれとの間のエッジのエッジ属性種類は、R2、R3、R1、R2、R1である。

【0063】

図13の重みデータ123が示すように、自ノードの状態値の重みはW0、エッジ属性種類R1、R2及びR3の重みは、それぞれ、W1、W2及びW3である。畳み込み演算の例は、エッジのエッジ属性種類毎に異なる重みをノードの状態値に乗算して足し合わせ、全てのエッジ属性種類の総和を計算する。つまり、ノードN1のための畳み込み演算は、以下の数式で表わされる。なお、バイアスは0とした。

$$V1' = W0 \times V1 + W1 * (V9 + V12) + W2 * (V2 + V10) + W3 * V6$$

【0064】

第1畳み込み層411は、畳み込み演算の結果の状態値V1'を非線形関数であるReLUに入力し、その出力をノードN1の新たな状態値V1として保持する。なお、ReLUと異なる非線形関数(例えばシグモイド関数)を使用してもよい。第1畳み込み層411

10

20

30

40

50

は、全てのノードについて、上述のような処理を行う。

【 0 0 6 5 】

[第 1 プーリング層]

次に、第 1 プーリング層 4 1 2 の処理を説明する。従来のグリッド上の CNN は、畳み込み演算（＋非線形活性化関数の適用）の後に、プーリング処理により画像サイズを縮小することが一般的である。本開示のグラフ上の CNN 4 0 0 のプーリング層は、入力グラフデータ 1 2 1 のグラフ構造を変更することなく従来のプーリング処理に相当する処理、つまり、グラフ上のプーリング処理を実行する。

【 0 0 6 6 】

プーリング処理は、近隣のノードで値を近づける処理であり、これによって、表現の自由度を低減する。以下において、いくつかのグラフ上のプーリング処理の例を説明する。プーリング処理には、いくつかの手法が知られているが、一般に使用される手法は、平均値プーリング及び最大値プーリングである。そこで、以下においては、グラフ上の平均値プーリング及び最大値プーリングを説明する。

【 0 0 6 7 】

プーリング処理は、畳み込み演算と同様に、グラフ構造を維持しつつ、各ノードの状態値を更新する。プーリング処理は、対象ノード及び対象ノードの近隣のノードの状態値から、対象ノードの新たな状態値を決定する。図 1 5 は、第 1 プーリング層 4 1 2 によるプーリング処理を模式的に示す。図 1 5 において、2 つのノード N 1 及び N 2 が着目されている。第 1 プーリング層 4 1 2 は、ノード N 1 の状態値を、隣接ノード N 2、N 6、N 9、N 1 0、N 1 2 の状態値と近づくように更新する。同様に、第 1 プーリング層 4 1 2 は、ノード N 2 の状態値を、隣接ノード N 1、N 3、N 4、N 5 の状態値と近づくように更新する。

【 0 0 6 8 】

グラフ上の平均値プーリングの例は、対象ノード及び隣接ノードの状態値をもとに、対象ノードの新たな状態値が、対象ノード及び隣接ノードの新たな状態値の平均値になるように、対象ノードの新たな状態値と決定する。例えば、第 1 プーリング層 4 1 2 は、ノード N 1 及びその隣接ノード N 2、N 6、N 9、N 1 0、N 1 2 の現在の（更新前の）状態値を取得し、それらの平均値を計算する。第 1 プーリング層 4 1 2 は、当該平均値をノード N 1 の新たな状態値と決定する。第 1 プーリング層 4 1 2 は、エッジ属性種類に応じて、ノードの状態値に重みを与えた値の平均値を計算してもよい。

【 0 0 6 9 】

グラフ上の平均値プーリングの他の例は、以下の数式に従って対象ノードの新たな状態値を決定する。N はグラフデータ 1 2 1 のノードの集合を示し、E はエッジの集合を示す。 p_i は、ノード i のプーリング処理の出力であり、 v_i は、ノード i のプーリング処理の入力である。 v_i は、直前の畳み込み演算の出力（状態値）である。 β はハイパーパラメータ、I は単位行列、L はラプラシアン行列である。D は、グラフデータ 1 2 1 の全エッジ本数である。右辺の第 2 項を D で除算しているのは、エッジ 1 本あたりの制約の強さをそろえるためであるが、D で割らない（ $D = 1$ とする）構成も考えられる。

【 0 0 7 0 】

【 数 1 】

$$p = \{p_i\} = \underset{\{p_i\}}{\operatorname{argmin}} \left[\sum_{i \in N} (p_i - v_i)^2 + \frac{\beta}{D} \sum_{(i,j) \in E} (p_i - p_j)^2 \right]$$

$$= (I + \beta L)^{-1} v$$

【 0 0 7 1 】

ノード i の第 1 プーリング層 4 1 2 の出力 p_i は、第 1 項より入力 v_i に近い値であり、かつ、第 2 項より隣接するノードの出力値 p_j とも近い値となる。 β が大きくなると第 2 項の寄与が大きくなるため、グラフ上のより広い範囲で値が近づくことになる。すなわち

、 はプーリング処理のカーネルサイズに相当するパラメータである。上記式は、より汎用的に適切な平均値プーリングを行うことができる。

【 0 0 7 2 】

図 1 6 は、図 1 5 に示すグラフ構造を有するグラフデータ 1 2 1 において、適用可能なラプラシアン行列 L の例を示す。上記プーリング処理の式及び図 1 6 に示すラプラシアン行列 L は、エッジ属性種類を無視して、全エッジを一律にプーリング処理する場合に使用される。プーリング処理は、エッジ属性種類を無視して実行してもよい。

【 0 0 7 3 】

これと異なり、エッジ属性種類に基づきプーリング処理を実行してもよい。これにより、より適切なプーリング処理を実行できる。第 1 プーリング層 4 1 2 は、エッジ属性種類毎にプーリング処理を実行する。具体的には、エッジ属性種類に基づくプーリング処理の例は、以下の数式に従う。

【 0 0 7 4 】

【数 2】

$$\begin{aligned} p = \{p_i\} = \underset{\{p_i\}}{\operatorname{argmin}} & \left[\sum_{i \in N} (p_i - v_i)^2 + \beta_1 \sum_{(i,j) \in E_1} (p_i - p_j)^2 \right. \\ & \left. + \beta_2 \sum_{(i,j) \in E_2} (p_i - p_j)^2 + \beta_3 \sum_{(i,j) \in E_3} (p_i - p_j)^2 \dots \right] \\ & = (I + \beta_1 L_1 + \beta_2 L_2 + \beta_3 L_3)^{-1} v \end{aligned}$$

【 0 0 7 5 】

E 1、E 2、E 3 は、それぞれ、エッジ属性種類 R 1、R 2 及び R 3 のエッジの集合を示す。 1、 2、 3 は、それぞれ、エッジ属性種類 R 1、R 2 及び R 3 のハイパーパラメータ（プーリングのカーネルサイズ）である。L 1、L 2、L 3 は、それぞれ、エッジ属性種類 R 1、R 2 及び R 3 のプーリングのラプラシアン行列である。

【 0 0 7 6 】

次に、グラフ上の最大値プーリングの例を説明する。グラフ上の最大値プーリングの例は、プーリング範囲内のノードの状態値の最大値を選択する。例えば、第 1 プーリング層 4 1 2 は、ノード N 2 及びその隣接ノード N 1、N 3、N 4、N 5 の現在の（更新前の）状態値を取得し、それらの内の最大値を選択する。第 1 プーリング層 4 1 2 は、当該最大値をノード N 2 の新たな状態値と決定する。第 1 プーリング層 4 1 2 は、エッジ属性種類に応じた重みに基づき、更新値を与えるノードを選択してもよい。

【 0 0 7 7 】

グラフ上の最大値プーリングの他の例は、入力値（ノードの現在の状態値）に Smooth Maximum を適用した後で、平均値プーリングを行う。Smooth Maximum としていくつかの関数が知られているが、例えば、Log Mean Exp を使用することができる。グラフ上の最大値プーリングの他の例は、以下の数式に従って対象ノードの新たな状態値を決定する。

【 0 0 7 8 】

【数 3】

$$p = \{p_i\} = \frac{1}{\rho} \log \left\{ \underset{\{p_i\}}{\operatorname{argmin}} \left[\sum_{i \in N} (p_i - \exp(\rho v_i))^2 + \frac{\beta}{D} \sum_{(i,j) \in E} (p_i - p_j)^2 \right] \right\}$$

【 0 0 7 9 】

は、Smooth Maximum の強さを表すパラメータであり、 の極限で、Log Mean Exp は最大値関数と一致する。他の変数は、平均値プーリングで説明した通りである。〔数 3〕で表わされる最大値プーリングは、エッジ属性種類を無視して、全エ

10

20

30

40

50

ッジを一律にプーリング処理する場合に使用される。プーリング処理は、エッジ属性種類を無視して実行してもよい。

【 0 0 8 0 】

平均値プーリングの説明と同様に、第 1 プーリング層 4 1 2 は、エッジ属性種類に基づきプーリング処理を実行してもよい。第 1 プーリング層 4 1 2 は、〔数 1〕に代えて、〔数 2〕を使用する。これにより、より適切なプーリング処理を実行できる。第 1 プーリング層 4 1 2 は、エッジ属性種類毎にプーリング処理を実行する。

【 0 0 8 1 】

[第 2 畳み込み層]

次に、第 2 畳み込み層 4 1 3 の処理を説明する。上述のように、第 1 プーリング層 4 1 2 は、グラフ構造のサイズを一定に保つ一方で、近隣のノード間で値を近づけるという正則化をとり入れることで、データの表現の自由度を低減する。第 2 畳み込み層 4 1 3 は、第 1 プーリング層 4 1 2 で近隣のノードの値が近づいてしまう分、広い範囲を参照する、つまり、畳み込みフィルタのカーネルサイズ（畳み込み演算で参照するノードの範囲）を広げる。

【 0 0 8 2 】

図 1 7 は、第 2 畳み込み層 4 1 3 による、ノード N 1 のための畳み込み演算を模式的に示す。同様の畳み込み演算が全てのノードに対して実行される。第 2 畳み込み層 4 1 3 は、ノード N 1 から 2 ホップ範囲内のノードの現在の（更新前の）状態値を取得する。2 ホップ範囲内のノードは、対象ノードから 1 ホップのノード及び 2 ホップのノードで構成される。なお、2 ホップ範囲内のノード、つまり 1 ホップのノード及び 2 ホップのノードを使う代わりに、2 ホップのノードのみを使う構成も考えられる。

【 0 0 8 3 】

図 1 7 に示す例において、ノード N 2、N 6、N 9、N 1 0、N 1 2 は、ノード N 1 から 1 ホップのノードである。ノード N 3 ~ N 5、N 7 ~ N 1 3 は、ノード N 1 から 2 ホップのノードである。ノード N 9、N 1 0、N 1 2 は、ノード N 1 から 1 ホップのノードであると共に、2 ホップのノードでもある。ノード N 9、N 1 0、N 1 2 は、ノード N 1 から 1 ホップのノードと定義し、ノード N 1 から 2 ホップのノードから除外してもよい。

【 0 0 8 4 】

第 2 畳み込み層 4 1 3 は、対象ノード N 1 の状態値、対象ノード N 1 から 1 ホップのノードの状態値及び対象ノード N 1 から 2 ホップのノードの状態値と、重みデータ 1 2 3 でエッジ属性種類毎に定義されている重みの積和を計算する。図 1 3 を参照して説明したように、重みデータ 1 2 3 は、自ノードの重み、1 つのエッジのエッジ属性種類の重み、及び 2 つのエッジの組のエッジ属性種類の重みを定義している。図示していないが、重みデータ 1 2 3 は、さらに、3 種類以上のエッジの組のエッジ属性種類の重みを定義する。第 1 畳み込み層 4 1 1 と同様に、畳み込み演算により得られた対象ノードの状態値は、ReLU に入力され、その出力が対象ノードの新しい状態値として保持される。なお、第 2 畳み込み層 4 1 3 の畳み込み演算は、バイアスを含むことができる。

【 0 0 8 5 】

[第 2 プーリング層]

第 2 プーリング層 4 1 4 は、図 9 に示すように、2 ホップ範囲内でプーリング処理を実行する。例えば、第 2 プーリング層 4 1 4 は、第 1 プーリング層 4 1 2 が実行するプーリング処理を 2 回実行する。これにより、第 2 畳み込み層 4 1 3 の処理範囲に応じたプーリング処理を行うことができる。

【 0 0 8 6 】

[全域平均プーリング層]

次に、全域平均プーリング層 4 1 5 の処理を説明する。グラフ上の CNN 4 0 0 は、ノード数やエッジ数が異なる様々なグラフ構造のグラフデータを取り扱う。従来の CNN における全結合層は、入力層のニューロンの数が固定であることを前提としており、グラフ上の CNN 4 0 0 にそのまま適用することはできない。本開示のグラフ上の CNN 4 0 0 は

10

20

30

40

50

、全域平均プーリングを使用する。なお、入力されるグラフデータのグラフ構造が一定である場合、全域平均プーリング層に代えて、全てのニューロンの値を一ヶ所に集めてアフィン演算することで最終出力する値を計算する全結合層を使用してもよい。

【 0 0 8 7 】

全域平均プーリング層 4 1 5 は、その入力のチャンネル毎に全てのニューロンの値の平均値をとる。したがって、全域平均プーリング層 4 1 5 の出力は入力チャンネル数の次元のベクトルとなる。例えば、全域平均プーリング層 4 1 5 の後段にアフィン変換層（全結合層）を配置し、最終的に必要とする次元の出力を得る。全域平均プーリング層 4 1 5 は全結合層に比べてパラメータ数が大幅に少ないにも関わらず、全結合層と同等以上の性能を示すことができる。

10

【 0 0 8 8 】

[誤差逆伝播による学習（訓練）]

以下において、グラフ上の CNN 4 0 0 の誤差逆伝播によるパラメータの学習を説明する。学習処理は、データ処理部 1 2 0 により実行される。誤差逆伝播は、順伝播におけるグラフのノードを逆にたどる。上述のように、第 2 畳み込み層 4 1 3 は、畳み込みフィルタのカーネルサイズを広げる。しかし、第 1 プーリング層 4 1 2 で近隣のノードの値が近づいているので、単純にカーネルサイズを広げるのみでは、パラメータ設定の自由度が大きくなりすぎて過学習を引き起こす、又は、不良設定問題となりパラメータの学習が全くできない可能性が存在する。

【 0 0 8 9 】

20

そこで、プーリング後の第 2 畳み込み層 4 1 3 に対して、畳み込みフィルタのカーネルサイズを大きくすると同時に、直前の第 1 プーリング層 4 1 2 と同様に、畳み込みフィルタ上の近隣の重みを近づけるという正則化を行う。こうすることで、第 1 プーリング層 4 1 2 と直後の第 2 畳み込み層 4 1 3 の自由度が揃うことになり、畳み込み演算とプーリング処理のセットを何層も積み上げる、いわゆる深層学習のグラフ上の CNN をより適切に構成することができる。

【 0 0 9 0 】

ここでは、グラフ上の CNN 4 0 0 のパラメータの学習（誤差逆伝播）は、畳み込み層における複数の重みを、1 段前のプーリング層と同じ正則化項を含む式により最適化する。より具体的には、CNN 4 0 0 全体の誤差関数 $E(w)$ に畳み込み層の正則化項を含める。

30

【 0 0 9 1 】

【 数 4 】

$$E(w) = E_0(w) + \frac{\beta}{2} \sum_{(i,j) \in E} (w_{2,i} - w_{2,j})^2 + \frac{\beta^2}{2} \sum_{(i,j) \in E} (w_{3,i} - w_{3,j})^2 + \dots$$

【 0 0 9 2 】

上記式は、3 以上の畳み込み層を含む構成の誤差関数 $E(w)$ を示す。 $E_0(w)$ は、畳み込み層の正則化項を含まない通常の誤差関数である。右辺の第 2 項は、第 2 畳み込み層の正則化項であり、第 3 項は第 3 畳み込み層の正則化項である。 $w_{2,i}$ 、 $w_{2,j}$ は、あるノードから 2 ホップ離れたノードにおいて隣接している（一つのエッジで結合されている）ノードを示す。 $w_{3,i}$ 、 $w_{3,j}$ は、あるノードから 3 ホップ離れたノードにおいて隣接しているノードを示す。はプーリング層で使用される と同一である。誤差関数 $E(w)$ は、2 層目の畳み込み層以降の畳み込み層の正則化項を含む。

40

【 0 0 9 3 】

上述のように、第 2 畳み込み層、第 3 畳み込み層と、後段にいくにしたがって畳み込みフィルタのカーネルサイズを大きくすることになる。上記式の右辺の第 2 項と第 3 項で にかかる次数を増やして正則化項の効果を大きくしていくことで、各畳み込み層における実質的なパラメータ設定の自由度を一定に保つ。

【 0 0 9 4 】

図 1 3 を参照して説明したように、重みデータ 1 2 3 は、自ノードの重み、1 つのエッジ

50

のエッジ属性種類の重み、及び 2 つのエッジの組のエッジ属性種類の重み、3 種類以上のエッジの組のエッジ属性種類の重み、を定義している。このとき、例えば、右辺の第 2 項、すなわち、第 2 畳み込み層の正則化項は、全ノードそれぞれから 2 ホップのノードにおいて隣接するノード間の重みの差の 2 乗の和を計算する。

【 0 0 9 5 】

ノード N 1 を例として、右辺の第 2 項を説明する。ノード N 1 から 2 ホップのノードは、ノード N 3 ~ N 5、N 7 ~ N 1 3 である。w_{2, i} は、これらノードの全ての重みがそれぞれ代入される。w_{2, j} は、w_{2, i} のノードに隣接する全てのノードの全ての重みが代入される。ノード N 1 の第 2 項の計算 (/ 2 を省略) は以下のように表わされる。

【 0 0 9 6 】

【 数 5 】

$$\begin{aligned} \sum_{(i,j) \in E} (w_{2,i} - w_{2,j})^2 = & \frac{(W_{12} - W_{22})^2}{N_3} + \frac{(W_{12} - W_{11})^2}{N_4} + \frac{(W_{12} - W_{12})^2}{N_3} + \frac{(W_{12} - W_{22})^2}{N_{10}} \\ & + \frac{(W_{23} - W_{22})^2}{N_3} + \frac{(W_{23} - W_{11})^2}{N_4} + \frac{(W_{23} - W_{12})^2}{N_3} + \frac{(W_{23} - W_{22})^2}{N_{10}} \\ & + \frac{(W_{23} - W_{22})^2}{N_3} + \frac{(W_{23} - W_{11})^2}{N_4} + \frac{(W_{23} - W_{12})^2}{N_3} + \frac{(W_{23} - W_{22})^2}{N_{10}} \\ & + \frac{(W_{22} - W_{12})^2}{N_4} + \frac{(W_{22} - W_{23})^2}{N_3} + \frac{(W_{22} - W_{23})^2}{N_4} \\ & + \frac{(W_{21} - W_{11})^2}{N_9} + \frac{(W_{21} - W_{12})^2}{N_{10}} \\ & + \dots \end{aligned}$$

【 0 0 9 7 】

ノード N 3 の例を説明する。ノード N 1 からノード N 3 を見て、2 つのエッジの組のエッジ属性種類の重みは、W_{1 2}、W_{2 3} (ノード N 2 を介する)、W_{2 3} (ノード N 1 0 を介する) である。従って、ノード N 3 の w_{2, i} は、W_{1 2}、W_{2 3}、W_{2 3} である。ノード N 1 から 2 ホップのノードにおいて、ノード N 3 に隣接するノードは、ノード N 4、N 1 0、N 1 2 である。したがって、w_{2, j} は、ノード N 1 からこれらノードを見た重みそれぞれが代入される。

【 0 0 9 8 】

例えば、ノード N 1 からノード N 4 を見て、2 つのエッジの組のエッジ属性種類の重みは、W_{2 2} である。ノード N 1 からノード N 1 0 を見て、2 つのエッジの組のエッジ属性種類の重みは、W_{1 1}、W_{1 2} である。ノード N 1 からノード N 1 2 を見て、2 つのエッジの組のエッジ属性種類の重みは、W_{2 2} である。

【 0 0 9 9 】

第 2 項は、ノード N 1 について、ノード 3 についての上記説明と同様の処理を、ノード N 4、N 5、N 7 ~ N 1 3 についても計算する。さらに、ノード N 1 以外の他のノードについても同様の計算を行う。第 3 項は、第 2 項と同様に、全ノードそれぞれから 3 ホップのノードについて計算を行う。

【 0 1 0 0 】

他の例において、正則化項は、グラフ構造に代えて、エッジ属性種類の関係に基づく計算を行ってもよい。例えば、第 2 項は、対象ノードに近いエッジのエッジ属性種類が同一の重みが近づくような計算を行ってもよい。例えば、重みグループ (W_{1 1}、W_{1 2}、W_{1 3})、(W_{2 1}、W_{2 2}、W_{2 3})、(W_{3 1}、W_{3 2}、W_{3 3}) それぞれにおいて、重みが近づくように、第 2 項が定義される。第 3 項以下についても、3 ホップ以上のエッジ属性種類に応じて同様に定義される。

【 0 1 0 1 】

10

20

30

40

50

第 2 畳み込み層の勾配値は、以下の数式のように、誤差関数 $E(w)$ を w で微分することにより得られる。

【 0 1 0 2 】

【数 6】

$$\frac{\partial E(w)}{\partial w_{i,2}} = \frac{\partial E_0(w)}{\partial w_{i,2}} + \beta L w_{i,2}$$

【 0 1 0 3 】

〔数 6〕における右辺の第 1 項は、上層からの誤差信号（通常の誤差逆伝播）を示す。右辺の第 2 項は、正規化項の微分を示す。他の畳み込み層の勾配値も同様の式で表わすことができる。学習処理は、この勾配値にしたがって、畳み込み層の重み w を更新する。例えば、SGD (stochastic gradient descent)、ADAM などの公知の手法を使用することができる。このように、誤差逆伝播による学習は、上層から逆伝播されてきた通常の誤差信号に、正則化項の微分の項（ $L w$ 、 n ）を加算した値を勾配として、重み w を更新する。これにより、畳み込み層（フィルタ）に対して、直前のプーリングに応じた制約を加えることができる。

【 0 1 0 4 】

次に、学習処理の全体について説明する。図 18 は、計算機 101 が実行する学習処理の一つの例を説明するフローチャートである。プロセッサ 110 は、ニューラルネットワークの構築処理を実行する（S201）。ニューラルネットワークの構築処理は後述する。

【 0 1 0 5 】

次に、プロセッサ 110 は、ストレージシステム 102 から学習データ 122 を取得し、メモリ 111 に格納する（S202）。学習データ 122 は複数のサンプルを含む。次に、プロセッサ 110 は、誤差逆伝播処理のループ処理を開始する（S203）。本ループ処理では、プロセッサ 110 は、ステップ S203 からステップ S207 の処理を繰り返し実行する。例えば、本ループ処理内で実行される誤差逆伝播処理によって算出される誤差が予め設定された閾値以下になった場合、又は、予め決められた回数だけ誤差逆伝播処理が実行された場合に、ループ処理が終了する。

【 0 1 0 6 】

プロセッサ 110 は、サンプルのループ処理を開始する（ステップ S705）。サンプルのループ処理では、ステップ S202 において取得された複数のサンプルに対し、サンプル毎にループ内部の処理を実行する。さらに、プロセッサ 110 は、1 つのサンプルに対して誤差逆伝播処理を実行する（S205）。

【 0 1 0 7 】

誤差逆伝播処理では、プロセッサ 110 は、あるサンプルを入力とし、その入力に対するグラフ上の CNN 400 の出力結果と当該サンプルに対応する教師データとを比較し、2 つのデータの誤差を削減するように、CNN 400 の上層から順に重みを更新する。

【 0 1 0 8 】

サンプルのループ処理では、プロセッサ 110 が、複数のサンプルの各々に対して前述した処理を実行する。ステップ S205 の誤差逆伝播処理では、任意のサンプルデータに対する重みの更新が 1 回だけ行われる。一方、誤差逆伝播処理のループ処理では、誤差を最小にするためには、誤差逆伝播処理が複数回実行される。具体的には、プロセッサ 110 が、所定の条件を満たすまで、繰り返し、複数のサンプルの各々に対して誤差逆伝播処理を実行する。また、サンプルのループ処理は、複数のサンプルに対する重みの更新が行われる。

【 0 1 0 9 】

次に、プロセッサ 110 は、全てのサンプルデータに対して誤差逆伝播処理が実行されたか否かを判定する（S206）。全てのサンプルデータに対して誤差逆伝播処理が実行されていないと判定された場合、プロセッサ 110 は、ステップ S204 に戻り、同様の処

10

20

30

40

50

理を実行する。全てのサンプルデータに対して誤差逆伝播処理が実行されたと判定された場合、プロセッサ 110 は、所定の条件を満たしたか否かを判定する (S207)。

【0110】

所定の条件を満たしていないと判定された場合、プロセッサ 110 は、ステップ S203 に戻り、同様の処理を実行する。所定の条件を満たしたと判定された場合、プロセッサ 110 は、学習結果をメモリ 111 又はストレージシステム 102 に格納する (ステップ S208)。学習結果は、構築されたニューラルネットワークの構造を示す情報及び学習により得られた重みの情報等が含まれる。重みは、エッジ属性種類それぞれに対して得られ、分析処理において、重みデータ 123 として使用される。

【0111】

[グラフ上の CNN の構築]

グラフ上の CNN 400 は、ノード数やエッジ数が異なる様々なグラフ構造のグラフデータを取り扱う。データ処理部 120 は、分析対象のグラフデータ 121 に応じた CNN 400 を構築する。データ処理部 120 は、グラフデータ 121 のグラフ構造に従って、畳み込み層及びプーリング層それぞれを構築する。データ処理部 120 は、学習結果からエッジ属性種類それぞれに対する重みを取得して、重みデータ 123 を構成する。

【0112】

以上のように、本実施例によれば、グラフ上の CNN においてプーリングを行うことで、表現可能自由度 (パラメータの自由度) を上層に行くほど小さくすることができ、パラメータの学習時の過学習や不良設定問題の蓋然性を低下させることができる。また、CNN の層の数を増加させていわゆる深層学習を行うグラフ上の CNN を構成することが可能となる。

【実施例 2】

【0113】

以下において、分散 CNN を説明する。分散 CNN は、ネットワークを介して接続された複数のサブシステム (例えば計算機) が、CNN を実行する。サブシステムは、サブシステムをネットワーク上で隣接するサブシステムとのみ通信 (情報交換) が可能である。なお、サブシステム間を接続するネットワーク構造は、固定である必要はなく時間的に変化してもよい。分散 CNN において、各サブシステムは、例えば、それぞれ独立にシステム状態関数の値を推定する。分散 CNN により、全てのサブシステムの推定値が真のシステム状態関数の値に一致する。

【0114】

他の例において、各サブシステムがそれぞれの環境入力と内部状態に応じてアクションを取ってもよい。各サブシステムは、全体システムの状態関数の値を最適化するようなアクションを見出すことができる。例えば、ネットワークを介して接続された工場 (の計算機) が、それぞれ隣接する工場とのみ通信を行い、全工場の総利益を最小化するような工場の電力使用量を決定する。工場又はその計算機は、サブシステムである。

【0115】

このほか、分散 CNN は、車間通信による、車のルート最適化に適用できる。車又はその計算機 (例えばカーナビゲーションシステム) は、サブシステムである。各車は、現在位置と、目的地とから、独立にルートを決断する。各車が近隣の車と通信することで、多数の車が、全体として一つの分散 CNN を構成する。各車は、自車のルートを入力として、分散 CNN によって、平均移動時間を回帰 (推定) する。

【0116】

各車は、独立に、誤差逆伝播によって、分散 CNN の出力値 (= 平均移動時間) を減らす方向に、ルート (= 入力) を更新する。同時に、各車は、独立に、道路交通情報から得た真の平均移動時間と、分散 CNN の出力値との誤差を減らすように、誤差逆伝播で、パラメータ (重み) を更新する。各車は、目的地につけば、分散 CNN から離脱する。

【0117】

[システム構成]

10

20

30

40

50

図 19 は、分散 CNN を実行する計算機システムの例を示す。計算機システムは、複数の計算機 101A ~ 101M を含み、これらはネットワークを介して接続されている。通信可能な隣接計算機は、エッジで接続されている。計算機の構成は実施例 1 で説明した通りである。図 19 のネットワーク構成は一例であって、他の例において、グリッド状に配置された計算機が隣接計算機とのみ通信可能であってもよい。

【0118】

分散 CNN は、CNN を構成するニューロンを層（畳み込み層やプーリング層等）ごとにまとめて扱うのではなく、複数の層の同一位置のニューロン（以下ではカラムと呼ぶ）をまとめて、一つのサブシステムで計算する。結果として、多数のサブシステムが集まったシステム全体が、単一の CNN として動作する。

10

【0119】

図 20 は、分散 CNN の層及びカラムの例を示す。分散 CNN は、畳み込み層やプーリング層等の層 701 ~ 704 を含む。図 20 は、一つのカラム 711 を例として示す。一つのカラムは、一つのサブシステムにより実行される。図 19 の計算機システムにおいて、いずれか一つの計算機が、カラムのニューロンの計算を行う。

【0120】

以上の説明から理解されるように、実施例 1 において説明したグラフ上の CNN は、分散 CNN に適用することができる。図 19 に示す計算機システムのネットワーク構造は、実施例 1 で説明したグラフデータ 121 のグラフ構造と一致している。計算機 101A ~ 101M は、それぞれ、CNN を構成する各層（畳み込み層やプーリング層）において、グラフデータ 121 における対応するノードの演算処理を行う。

20

【0121】

図 21 は、分散 CNN の概念的な構成例を示す。図 21 は、3 つの計算機（サブシステム）101B、101C、及び 101D を例として示す。計算機 101D は、計算機 101B 及び計算機 101C と隣接しており、通信可能である。計算機 101B、101C、及び 101D は、それぞれ、カラム 711B、711C、及び 711D を実行する。カラムを構成する四角はニューロンを示す。

【0122】

各サブシステムは、自装置が担当するカラム上のニューロンの出力値をまとめたベクトルを内部状態として保持する。各サブシステムは、自分の内部状態ベクトルと、隣接サブシステムと通信によって得た隣接サブシステムの内部状態ベクトル（隣接位置のカラム上のニューロンの値）を入力として、各層の演算を行って、自分の内部状態ベクトルを更新する。

30

【0123】

分散 CNN を実行する計算機システムにおいて、システム全体を統括する中央制御システムは存在せず、システム全体でのグローバルな同期機構も存在しない。各サブシステムは、隣接サブシステムとの通信とサブシステムの内部状態ベクトルの更新を、他のサブシステムは非同期に実行する。サブシステムは、他のサブシステムの内部状態ベクトルと自サブシステムの内部状態ベクトルに基づき、自サブシステムの内部状態ベクトルを更新する。

【0124】

隣接サブシステムとの通信とサブシステムの内部状態ベクトルの更新は、同頻度で行われるとは限らず、あるサブシステムでは隣接サブシステムとの通信頻度が高いのに対して、別のサブシステムでは内部状態ベクトルの更新の頻度高いこともある。あるサブシステムが複数の隣接サブシステムをもつとき、それぞれの隣接サブシステムとの通信が行われる頻度は同一でなくてよい。

40

【0125】

分散 CNN を実行する計算機システムは、隣接サブシステム間の通信と内部状態ベクトルの更新処理をそれぞれ非同期に行う。したがって、サブシステムは、内部状態ベクトルの更新を実行する時に、畳み込み演算に必要な隣接カラム（隣接サブシステムが実行するカラム）上のニューロンの値を得ることはできない。

50

【 0 1 2 6 】

そのため、サブシステムは、前回の隣接サブシステムとの通信のときに得られた隣接コラム上のニューロンの値（内部状態ベクトル）をメモリに記憶し、内部状態ベクトルの更新時には記憶しておいた値を用いて畳み込み演算を行う。この構成は、後述するように、誤差逆伝播を非同期で行うことを可能とする。

【 0 1 2 7 】

[畳み込み]

以下において、分散CNNの各層の処理を説明する。計算機システムは、実施例1において説明した、グラフ上のCNN400における各層の処理を実行することができる。まず、畳み込み層の処理を説明する。分散CNNを実行する各サブシステムは、自システムが担当するカラムの畳み込み演算する。各サブシステムは、実施例1で説明した畳み込み層の処理を、担当するカラムにおいて実行する。

10

【 0 1 2 8 】

実施例1において説明したように、畳み込み演算は、対象ノード（ニューロン）の値及び対象ノードから所定ホップ数のノードの値を使用する。分散CNNでは、各サブシステムは、対象ノード（ニューロン）以外のノードの値を、他のサブシステムから収集する。

【 0 1 2 9 】

上述のように、各サブシステムは、隣接サブシステムとの通信時に得た隣接カラム上のニューロンの値を記憶しておき、内部状態ベクトルの更新時には、記憶しておいた値を用いて畳み込み演算を行う。畳み込み演算の重み（カーネルの係数）は、全てのサブシステムで共通である。例えば、サブシステムは、互いに通信を行うことで、同一の重みデータを共有することができる。

20

【 0 1 3 0 】

サブシステムは、畳み込み演算のカーネル範囲（ホップ数）内の全てのカラムの値を必要とする。実施例1で説明したように、プーリング層の後の畳み込み層におけるカーネルサイズは広げられる。第2畳み込み層413の処理のような、2ホップ以上のカーネルサイズの畳み込み演算のため、サブシステムは、隣接していないサブシステムのカラムの値（内部状態ベクトル）を、隣接サブシステムを介したマルチホップ中継により取得する。nホップの情報伝達は、n回の通信が必要である。

【 0 1 3 1 】

[プーリング]

次に、プーリング処理について説明する。分散CNNでは、各サブシステムが、CNNの下層から上層まで同一位置のニューロンをカラムとして扱うため、全ての層でニューロンの数は一定である。これは、実施例1において説明したグラフ上のCNNと同様である。分散CNNにおいて、各サブシステムは、実施例1で説明したプーリング処理と同様に、そこで、ニューロン数（ノード数）を削減することなく、表現の自由度を低下させるプーリング処理を実行する。

30

【 0 1 3 2 】

実施例1において説明した平均値プーリング及び最大値プーリングの双方が、分散CNNにおけるプーリング処理に適用できる。実施例1において説明したように、プーリング処理は、隣接ノードの状態値を必要とする。分散CNNにおいては、サブシステムは、自システムの内部状態ベクトルと隣接するサブシステムから取得した内部状態ベクトルから、自システムの新たな内部状態ベクトルを計算する。

40

【 0 1 3 3 】

[全結合]

次に全結合について説明する。集中型のCNNは、畳み込み演算及びプーリングを何度か繰り返したのち全結合層によって全てのニューロンの値を一ヶ所に集めてアフィン演算することで、最終出力する値を計算する。一方、分散CNNでは、各サブシステムは自律的に動作しており、全サブシステムの値を一ヶ所に集めることはできない。

【 0 1 3 4 】

50

そこで、本開示の計算機システムは、分散で平均値を求める既知のアルゴリズムを利用する。分散で平均値を求めるアルゴリズムは、例えば、分散ADMM (Alternating Direction Method of Multipliers) や平均合意アルゴリズムである。

【 0 1 3 5 】

具体的には、各サブシステムは、まず、自分のカラム上のニューロンの出力値に、対応する全結合層の重み係数を乗算する。次に、全てのサブシステムの乗算結果の値の平均値を、後述のアルゴリズムを用いて分散で計算することで、全結合層の出力値を得ることができる。なお、この方法は、ニューロンの値に重み係数を乗算したものの総和ではなく平均値が出力する。重み係数は学習データ (教師データ) から学習されるため、この違いが問題になることはない。従来の集中型CNNで学習した重み係数を分散CNNで用いる場合には、各サブシステムに保持させる重み係数の値に、あらかじめ全サブシステムの数に乗算しておけばよい。

10

【 0 1 3 6 】

以下において、分散で平均値を計算するアルゴリズムの二つの例を説明する。第1の例は分散ADMMを用いた平均値計算アルゴリズムである。分散ADMMを用いて以下の最適化問題を解くことで、N個の値 x_i の平均値 を分散で計算することができる。

【 0 1 3 7 】

【数7】

$$v = \operatorname{argmin}_{\{v\}} \sum_{i=1}^N \frac{1}{2} (x_i - v)^2$$

20

【 0 1 3 8 】

分散ADMMを用いる方法は、常に、上記式を満たすように最適化を行うため、動作の最中に値 x_i やネットワークトポロジが動的に変化した場合でも、変化に追従してその時点での平均値 を出力することができる。したがって、任意の指定した時点の値に対する平均値を計算することができる。

【 0 1 3 9 】

第2の例は平均合意アルゴリズムである。平均合意アルゴリズムは、各サブシステムで保持する値 x_i の総和を一定に保ちながら、サブシステム間で値 x_i を分配し、最終的に、全てのサブシステムの値 x_i のを同一の値に収束させる。値 x_i の総和が常に一定であることから、収束値は各値 x_i 初期値の平均値となる。いくつかの種類の平均アルゴリズムの中で、最も単純な平均合意アルゴリズムは、例えば、サブシステム i とサブシステム j の通信時に、値 x_i と値 x_j を以下の式に従って更新する。

30

【 0 1 4 0 】

【数8】

$$x_i^{t+1} = x_j^{t+1} = \frac{1}{2} (x_i^t + x_j^t)$$

40

【 0 1 4 1 】

これにより、隣接間通信が行われるたびに、システム全体の値 x_i の総和を一定に保ちながら、値 x_i の分散が減少するため、いずれ、全ての値 x_i が同一の値、すなわち、各値 x_i の初期値の平均値に収束することになる。

【 0 1 4 2 】

平均合意アルゴリズムは、動作の最中に値 x_i の値やネットワークトポロジが動的に変化すると、システム全体での値 x_i の総和が変化するため、各値 x_i の初期値の平均値を正しく計算できない。一方で、平均合意アルゴリズムが持つ収束値が常にその時点でのシステム全体の値 x_i の平均値になっているので、誤差逆伝播による畳み込みフィルタの重み係数の学習に有用である。

50

【 0 1 4 3 】

[誤差逆伝播による学習（訓練）]

上述のように、分散CNNにおいて、サブシステムは、前回の隣接サブシステムとの通信のときに得られた隣接カラム上のニューロンの値を記憶し、記憶しておいた値を用いて畳み込み演算等の順伝播の計算を行うことで内部状態ベクトルを更新する。サブシステムは、誤差逆伝播計算時に順伝播時に用いた値をそのまま使うことで、他のサブシステムとの同期なく、誤差逆伝播によるパラメータ更新を行うことができる。

【 0 1 4 4 】

誤差逆伝播の計算は、順伝播のグラフのノード（ニューロン）を逆にたどることで行う。図22～25にグラフ上のノードの種類ごとの伝播則の例を示す。具体的には、図22～25は、それぞれ、関数 f の適用ノード771、可算ノード772、乗算ノード773及び変数複製ノード（分岐ノード）774を示す。順伝播の伝播則は実線で示され、逆伝播の伝播則は破線で示されている。順伝播の入力は x 及び y である。分散CNNでは、このうち変数複製ノード774の実現に工夫が必要である。

10

【 0 1 4 5 】

変数複製ノード774は、順伝播の計算時に同一の変数を複数回使用することに相当する。分散CNNでは複数のサブシステムが分散して計算するため、図25の変数複製ノード774の逆伝播則において、上層からの誤差信号 $dx1$ 及び $dx2$ のうち、自サブシステム以外のサブシステムが担当するカラムからの信号を得ることはできない。

【 0 1 4 6 】

そこで、変数複製ノード774の誤差逆伝播では、各サブシステムが、それぞれ自サブシステムが担当するカラム内部で得られる誤差信号のみを使って、独立にパラメータの微係数を計算してパラメータ値を更新する。その後、各サブシステムは、上記平均合意アルゴリズムを用いて全てのサブシステムが保持するパラメータ値の平均を計算する。平均合意アルゴリズムでは、収束値が、常にその時点での全てのサブシステムが持つ値の平均値になっているため、継続的に学習、すなわち、パラメータ値の更新を続けることができる。

20

【 0 1 4 7 】

なお、数学的に正しい微係数が上層からの全ての誤差信号の総和であるのに対して、この方法は、上層からの全ての誤差信号の平均値を微係数と考えてパラメータを更新する。そのため、パラメータ更新に用いる学習アルゴリズムのハイパーパラメータである学習率（ $Learning Rate$ ）に、あらかじめ全サブシステムの数を乗算しておく。

30

【 0 1 4 8 】

逆誤差伝播における分岐ノードの後段（順伝播では前段）にさらに別のノードがある場合も、同様に、分岐ノードでは各サブシステムが自サブシステムが担当するコラムの情報のみを用いて誤差信号を逆伝播させて後段にあるパラメータ更新を独立に実行し、その後にパラメータ値の平均値を平均合意アルゴリズムにより計算する。

【 0 1 4 9 】

[分散CNNの構成例]

以下において、本開示の分散CNNの構成例を説明する。図26は、分散CNNの構成例900及びその処理を示す。以下において、一つのサブシステムが実行する処理を説明する。まず、順伝播を説明する。順伝播において、第1畳み込み層901は、実施例1の第1畳み込み層411と同様に、畳み込み演算及びReLUを実行する。第1畳み込み層901は、自内部状態ベクトルから対象ニューロンの値を取得し、また、予め保持している隣接サブシステムの内部状態ベクトルから、隣接ニューロンの値を取得する。第1畳み込み層901は、これらの値の畳み込み演算及びReLUを実行し、自内部状態ベクトルを更新する。

40

【 0 1 5 0 】

第1プーリング層902は、実施例1の第1畳み込み層411と同様に、プーリング処理を実行する。第1プーリング層902は、自内部状態ベクトルから第1畳み込み層901により更新された対象ニューロンの値を取得し、また、予め保持している隣接サブシステ

50

ムの内部状態ベクトルから、隣接ニューロンの第 1 畳み込み層 9 0 1 の出力値を取得する。第 1 プーリング層 9 0 2 は、これらの値のプーリング、本例において最大値プーリングを実行し、自内部状態ベクトルを更新する。

【 0 1 5 1 】

第 2 畳み込み層 9 0 4 は、実施例 1 の第 2 畳み込み層 4 1 3 と同様に、畳み込み演算及び R e L U を実行する。第 2 畳み込み層 9 0 4 は、自内部状態ベクトルから第 1 プーリング層 9 0 2 に更新された対象ニューロンの値を取得し、また、予め保持している 2 ホップ内のサブシステムの内部状態ベクトルから、隣接ニューロンの第 1 プーリング層の出力値を取得する。2 ホップ離れたサブシステムからのデータは、隣接サブシステムを介して転送される。第 2 畳み込み層 9 0 4 は、これらの値の畳み込み演算及び R e L U を実行し、自内部状態ベクトルを更新する。

10

【 0 1 5 2 】

第 2 プーリング層 9 0 6 は、実施例 1 の第 2 プーリング層 4 1 4 と同様に、プーリング処理を実行する。第 2 プーリング層 9 0 6 は、自内部状態ベクトルから第 2 畳み込み層 9 0 4 に更新された対象ニューロンの値を取得し、また、予め保持している隣接サブシステムの内部状態ベクトルから、隣接ニューロンの値を取得する。第 2 プーリング層 9 0 6 は、これらの値のプーリング、本例において最大値プーリングを実行し、自内部状態ベクトルを更新する。

【 0 1 5 3 】

第 2 プーリング層 9 0 6 は、さらに、自内部状態ベクトルから直前のプーリング処理で更新されたニューロンの値を取得し、予め保持している隣接サブシステムの内部状態ベクトルから、隣接ニューロンの値を取得する。第 2 プーリング層 9 0 6 は、これらの値のプーリング、本例において最大値プーリングを実行し、自内部状態ベクトルを更新する。

20

【 0 1 5 4 】

全結合層 9 0 7 は、分散 A D M M によって平均値を計算する。全結合層 9 0 7 は、自内部状態ベクトルの対象ニューロンの値と、予め保持している他のサブシステムの内部状態ベクトルの対象ニューロンの値から、分散 A D M M によって平均値を計算する。

【 0 1 5 5 】

次に、誤差逆伝播を説明する。誤差逆伝播において、各サブシステムは、教師データを与えられている。各サブシステムは、内部に保持しているデータを使用して、他のサブシステムから独立に、重みデータ（バイアスを含む）を更新する。各サブシステムは、平均合意アルゴリズムによって、更新した重みデータの平均値を計算する。これにより、全てのサブシステムの重みデータが共通化される。

30

【 0 1 5 6 】

なお、本発明は上記した実施例に限定されるものではなく、様々な変形例が含まれる。例えば、上記した実施例は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明したすべての構成を備えるものに限定されるものではない。また、ある実施例の構成の一部を他の実施例の構成に置き換えることが可能であり、また、ある実施例の構成に他の実施例の構成を加えることも可能である。また、各実施例の構成の一部について、他の構成の追加・削除・置換をすることが可能である。

40

【 0 1 5 7 】

また、上記の各構成・機能・処理部等は、それらの一部又は全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリや、ハードディスク、SSD (Solid State Drive) 等の記録装置、または、ICカード、SDカード等の記録媒体に置くことができる。また、制御線や情報線は説明上必要と考えられるものを示しており、製品上必ずしもすべての制御線や情報線を示しているとは限らない。実際には殆どすべての構成が相互に接続されていると考えてもよい。

50

【符号の説明】

【 0 1 5 8 】

1 0 1 計算機、 1 1 0 プロセッサ、 1 1 1 メモリ、 1 2 3 重みデータ、 4 0 0 C
NN、 4 1 1 第 1 畳み込み層、 4 1 2 第 1 プーリング層、 4 1 3 第 2 畳み込み層、 4
1 4 第 2 プーリング層、 9 0 0 分散 CNN

10

20

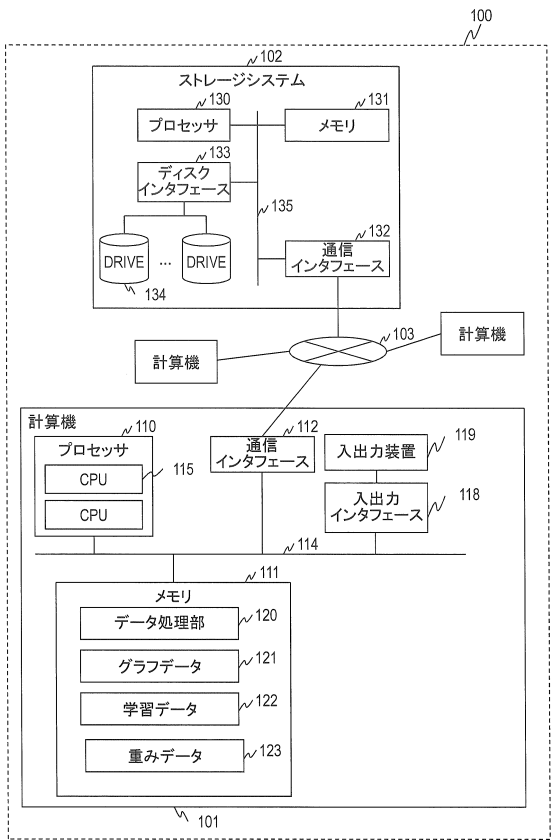
30

40

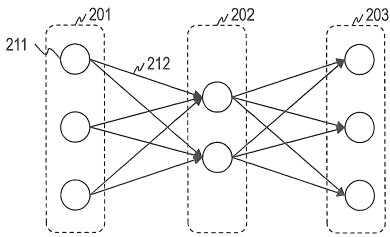
50

【図面】

【図 1】



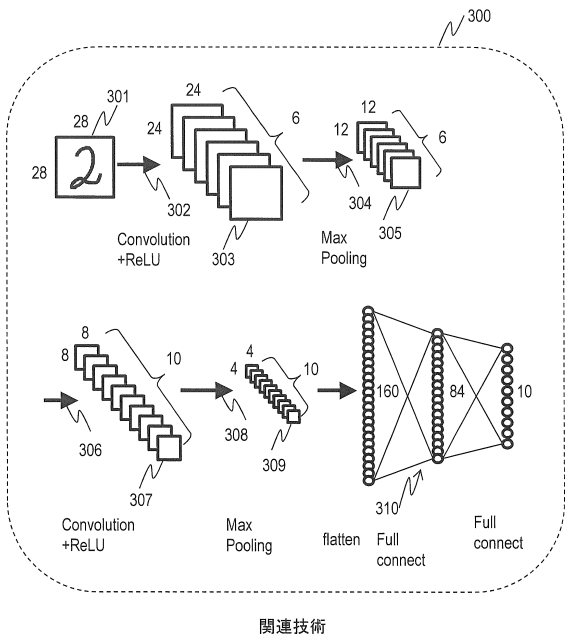
【図 2】



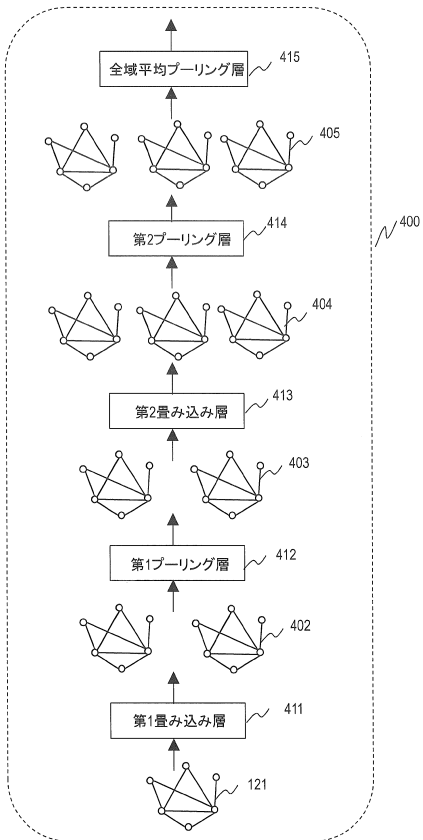
10

20

【図 3】



【図 4】

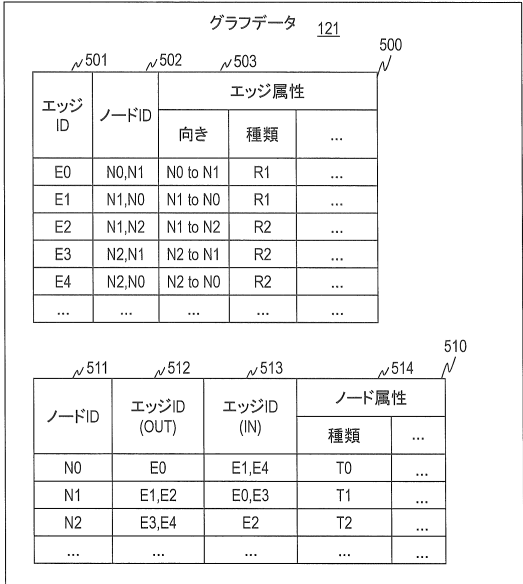


30

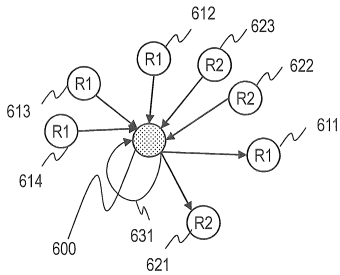
40

50

【 図 5 】

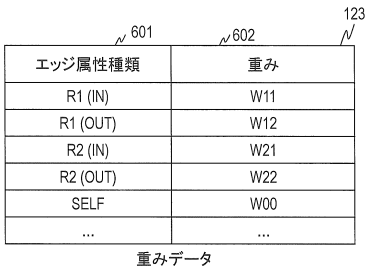


【 図 6 】

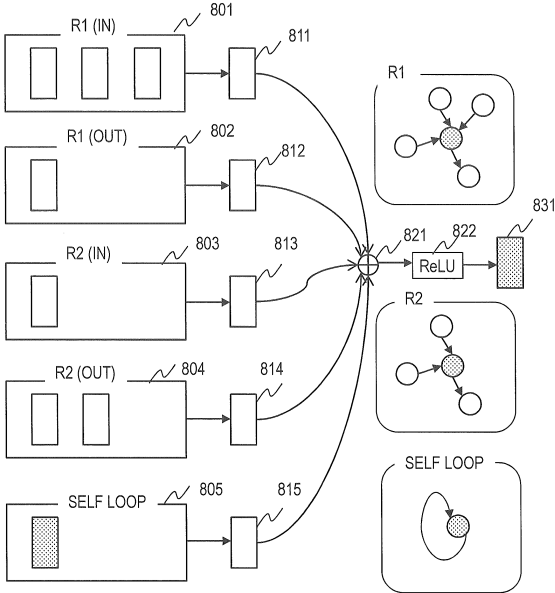


10

【 図 7 】



【 図 8 】



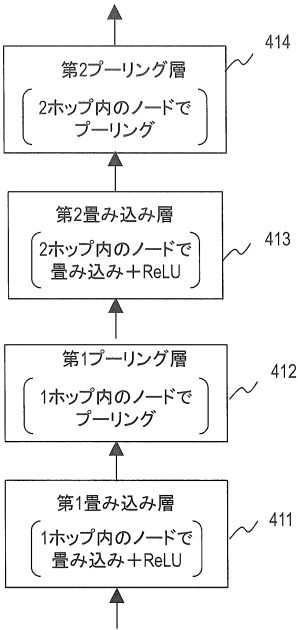
20

30

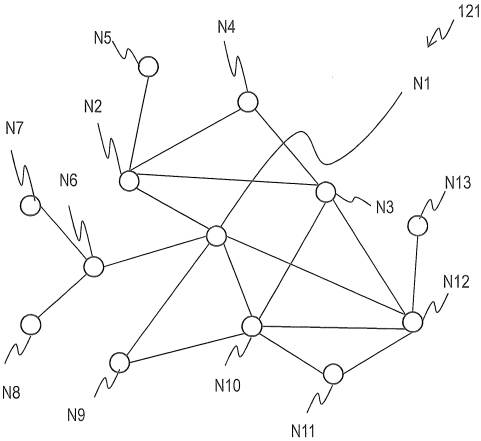
40

50

【図 9】

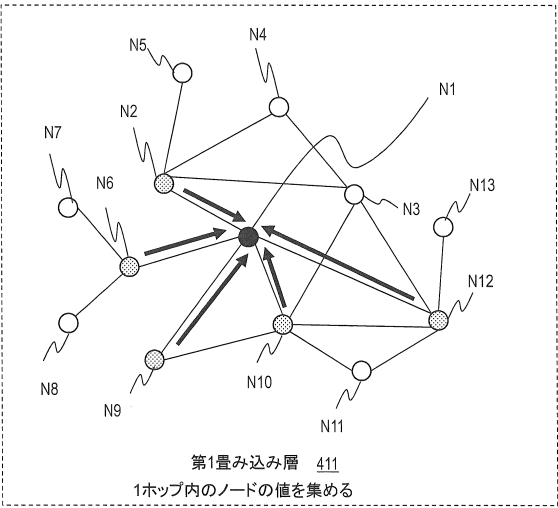


【図 10】

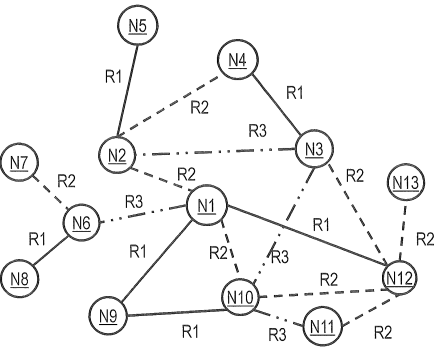


10

【図 11】



【図 12】



30

40

50

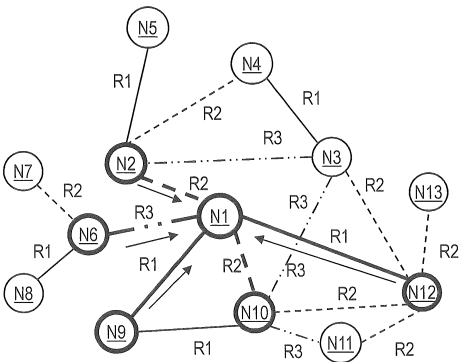
【図 1 3】

601 602 123

エッジ属性 種類	重み
SELF	W0
R1	W1
R2	W2
R3	W3
R1R1	W11
R1R2	W12
R1R3	W13
R2R1	W21
R2R2	W22
...	...

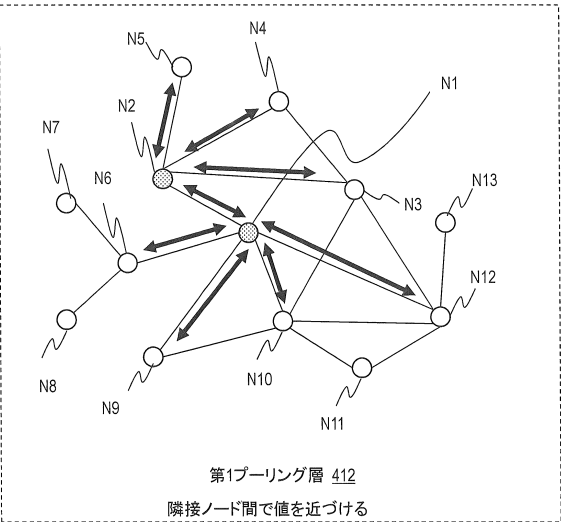
重みデータ

【図 1 4】



10

【図 1 5】



第1ブーリング層 412

隣接ノード間で値を近づける

【図 1 6】

$$L = \begin{pmatrix} 5 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 5 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 5 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}$$

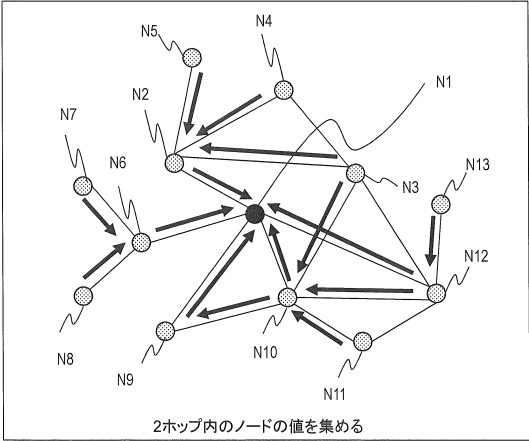
20

30

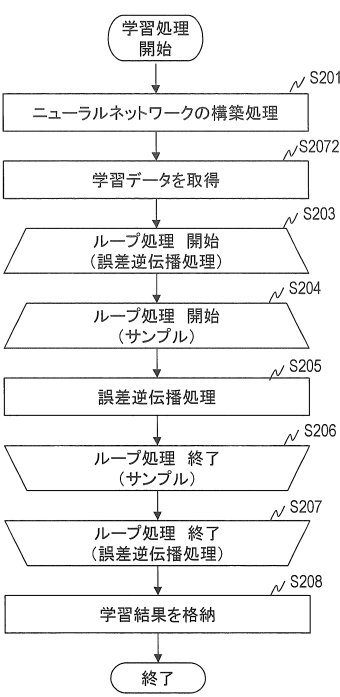
40

50

【図 17】



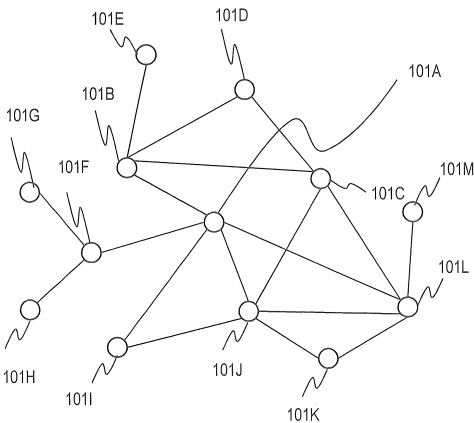
【図 18】



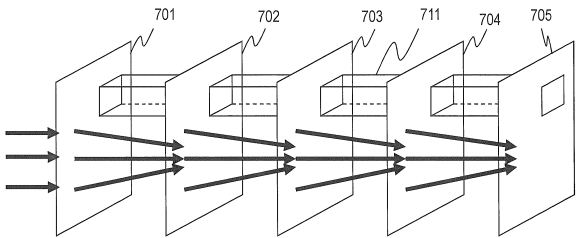
10

20

【図 19】



【図 20】

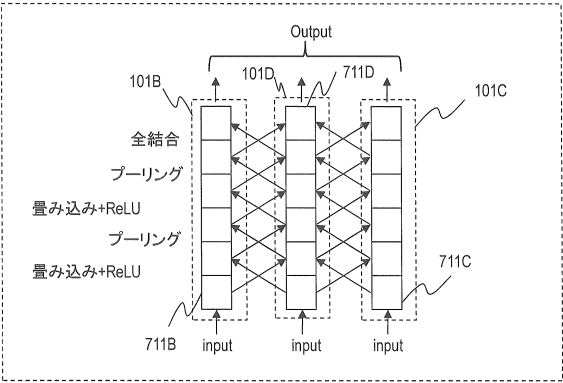


30

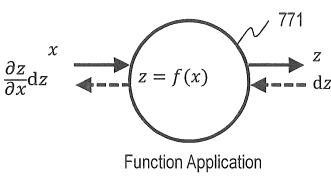
40

50

【図 2 1】

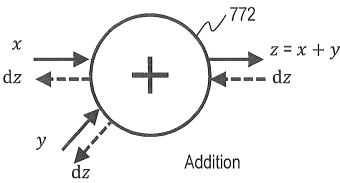


【図 2 2】

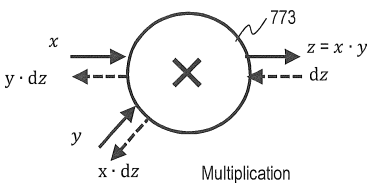


10

【図 2 3】



【図 2 4】



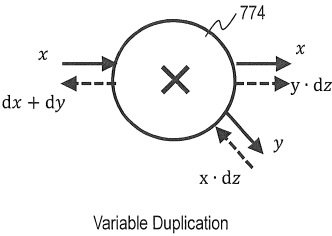
20

30

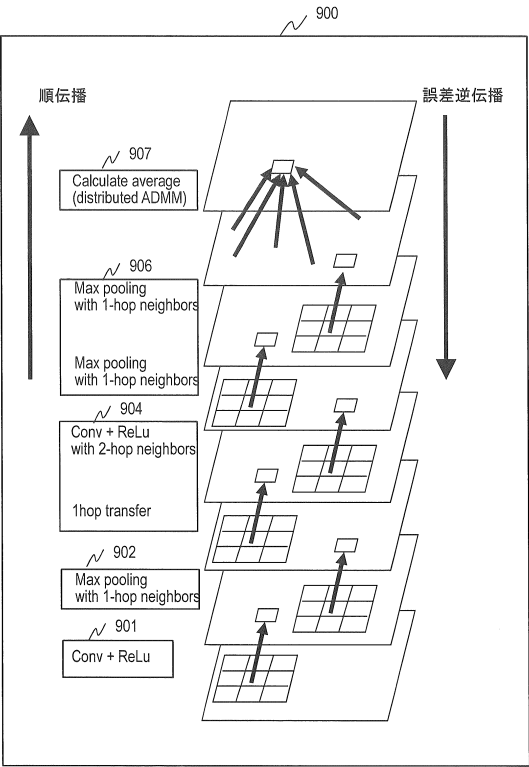
40

50

【 図 2 5 】



【 図 2 6 】



10

20

30

40

50

フロントページの続き

- (56)参考文献 米国特許出願公開第 2 0 1 9 / 0 1 2 2 1 1 1 (U S , A 1)
特開 2 0 1 9 - 0 5 6 9 6 0 (J P , A)
米国特許出願公開第 2 0 1 9 / 0 0 8 7 5 0 9 (U S , A 1)
- (58)調査した分野 (Int.Cl. , D B 名)
- G 0 6 N 3 / 0 0 - 3 / 1 2
G 0 6 N 7 / 0 8 - 9 9 / 0 0
G 0 6 N 5 / 0 0 - 7 / 0 6