

明 細 書

発明の名称：

データ処理装置、データ処理システムおよびデータ処理方法

技術分野

[0001] 本発明は、ニューラルネットワークの構成に関する情報が符号化された符号化データを生成するデータ処理装置、データ処理システムおよびデータ処理方法に関する。

背景技術

[0002] 入力データの分類（識別）問題および回帰問題を解決する方法として機械学習がある。機械学習には、脳の神経回路（ニューロン）を模擬したニューラルネットワークという手法がある。ニューラルネットワーク（以下、NNと記載する）では、ニューロンが相互に結合されたネットワークによって表現された確率モデル（識別モデル、生成モデル）によって、入力データの分類（識別）または回帰が行われる。

[0003] また、NNは、大量のデータを用いた学習によってNNのパラメータを最適化することで、高性能化することができる。ただし、近年のNNは大規模化しており、NNのデータサイズが大容量化の傾向にあり、NNを用いたコンピュータの計算負荷も増加している。

[0004] 例えば、非特許文献1には、NNの構成を示す情報であるエッジの重み（バイアス値を含む）を、スカラ量子化した上で符号化する技術が記載されている。エッジの重みをスカラ量子化した上で符号化することで、エッジに関するデータのデータサイズが圧縮される。

先行技術文献

非特許文献

[0005] 非特許文献1: Vincent Vanhoucke, Andrew Senior, Mark Z. Mao, “Improving the speed of neural networks on CPUs

”, Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2011.

発明の概要

発明が解決しようとする課題

[0006] データ伝送ネットワークを介して、複数のクライアントが、サーバと繋がったシステムにおいて、サーバ側で学習されたNNの構造を示すデータを符号化して、符号化データをクライアント側で復号することで、複数のクライアントのそれぞれが、サーバで学習されたNNを用いてデータ処理を行うものがある。従来のシステムにおいては、NNの構造を更新する場合、更新されたレイヤに関する情報に加え、更新されなかったレイヤに関する情報についてもクライアントへ伝送される。このため、伝送されるデータサイズの削減ができないという課題があった。

[0007] 本発明は上記課題を解決するものであって、NNの構造を示すデータのデータサイズを削減することができるデータ処理装置、データ処理システムおよびデータ処理方法を得ることを目的とする。

課題を解決するための手段

[0008] 本発明に係るデータ処理装置は、NNを学習するデータ処理部と、NNのモデルを識別するためのモデルヘッダ情報、NNのレイヤを識別するためのレイヤヘッダ情報およびレイヤ単位のエッジの重み情報が符号化された符号化データを生成する符号化部とを備え、符号化部は、ニューラルネットワークのレイヤ構造を示すレイヤ構造情報と、符号化される各レイヤが参照モデルのレイヤからの更新であるか新規レイヤであるかを示す新規レイヤフラグを符号化する。

発明の効果

[0009] 本発明によれば、符号化部が、NNのレイヤ構造を示すレイヤ構造情報と、符号化される各レイヤが参照モデルのレイヤからの更新であるか新規レイ

ヤであるかを示す新規レイヤフラグを符号化する。NNの構造を示すデータのうち、更新されたレイヤに関する情報のみが符号化されて伝送されるので、NNの構造を示すデータのデータサイズを削減することができる。

図面の簡単な説明

[0010] [図1]実施の形態1に係るデータ処理システムの構成を示すブロック図である。

[図2]NNの構成例を示す図である。

[図3]実施の形態1に係るデータ処理装置（エンコーダ）の構成を示すブロック図である。

[図4]実施の形態1に係るデータ処理装置（デコーダ）の構成を示すブロック図である。

[図5]実施の形態1に係るデータ処理装置（エンコーダ）の動作を示すフローチャートである。

[図6]実施の形態1に係るデータ処理装置（デコーダ）の動作を示すフローチャートである。

[図7]実施の形態1における符号化データの例を示す図である。

[図8]実施の形態1における符号化データの別の例を示す図である。

[図9]実施の形態1における1次元データの畳み込み処理の例を示す図である。

[図10]実施の形態1における2次元データの畳み込み処理の例を示す図である。

[図11]NNの1層目のレイヤにおけるノードごとのエッジの重み情報のマトリクスを示す図である。

[図12]NNの1層目のレイヤにおけるノードごとのエッジの重み情報の量子化ステップのマトリクスを示す図である。

[図13]畳み込み層におけるエッジの重み情報のマトリクスを示す図である。

[図14]畳み込み層におけるエッジの重み情報の量子化ステップのマトリクスを示す図である。

[図15]実施の形態1に係るデータ処理装置（エンコーダ）の変形例の構成を示すブロック図である。

[図16]実施の形態1における符号化データの更新の概要を示す図である。

[図17]図16に示す符号化データの更新に対応したネットワークモデルの構成を示す図である。

[図18]モデル情報ヘッダに含まれるレイヤ構造情報の例を示す図である。

[図19]モデル情報ヘッダに含まれるレイヤ構造情報に対応するレイヤ識別情報の例を示す図である。

[図20]図20Aは、実施の形態1に係るデータ処理装置の機能を実現するハードウェア構成を示すブロック図である。図20Bは、実施の形態1に係るデータ処理装置の機能を実現するソフトウェアを実行するハードウェア構成を示すブロック図である。

発明を実施するための形態

[0011] 実施の形態1.

図1は、実施の形態1に係るデータ処理システムの構成を示すブロック図である。図1に示すデータ処理システムにおいて、サーバ1は、データ伝送ネットワーク2を介して、クライアント3-1, 3-2, ..., 3-Nと繋がっている。Nは、2以上の自然数である。サーバ1は、大量のデータを用いた学習によってNN（ニューラルネットワーク）のパラメータを最適化して、高性能なNNを生成するデータ処理装置であり、図1に示すデータ処理システムが備える第1のデータ処理装置である。

[0012] データ伝送ネットワーク2は、サーバ1とクライアント3-1, 3-2, ..., 3-Nとの間でやり取りされるデータが伝送されるネットワークであり、インターネットまたはイントラネットである。例えば、データ伝送ネットワーク2には、NNを生成するための情報が、サーバ1からクライアント3-1, 3-2, ..., 3-Nへ送信される。

[0013] クライアント3-1, 3-2, ..., 3-Nは、サーバ1によって学習されたNNを生成し、生成されたNNを用いたデータ処理を行う機器である。

。例えば、クライアント3-1, 3-2, ..., 3-Nは、パーソナルコンピュータ（PC）、カメラ、またはロボットといった、通信機能およびデータ処理機能を有した機器である。クライアント3-1, 3-2, ..., 3-Nのそれぞれは、図1に示すデータ処理システムが備える第2のデータ処理装置である。

[0014] 図1に示すデータ処理システムにおいて、クライアント3-1, 3-2, ..., 3-Nのそれぞれは、NNのモデルおよびパラメータのデータサイズおよび適切な性能を示す値が異なる。このため、非特許文献1に記載された技術を用いてNNのモデルおよびパラメータを符号化しても、クライアント3-1, 3-2, ..., 3-Nのそれぞれに適したデータサイズに圧縮する必要があり、符号化の処理負荷が高くなる。

[0015] そこで、実施の形態1に係るデータ処理システムでは、サーバ1が、NNのモデルを識別するためのモデルヘッダ情報と、NNのレイヤを識別するためのレイヤヘッダ情報と、レイヤ単位のバイアス値を含んだエッジの重み（以降、特に明記しない限り、エッジの重みはバイアス値を含むものとする）の情報とが符号化された符号化データを生成し、データ伝送ネットワーク2を介してクライアント3-1, 3-2, ..., 3-Nに送信する。クライアント3-1, 3-2, ..., 3-Nのそれぞれは、データ伝送ネットワーク2を介してサーバ1から伝送された符号化データのうち、必要なレイヤに関する情報のみを復号することが可能である。これにより、サーバ1における符号化の処理負荷が低減され、サーバ1からデータ伝送ネットワーク2へ伝送されるデータサイズを削減することができる。

[0016] ここで、NNの構成について説明する。図2は、NNの構成例を示す図である。図2に示すように、入力データ（ $x_1, x_2, \dots, x_{N_1}$ ）は、NNが有するそれぞれの層で処理され、処理結果（ $y_1, \dots, y_{N_L}$ ）が出力される。 N_l （ $l=1, 2, \dots, L$ ）は、 l 層目のレイヤのノード数を示しており、 L は、NNのレイヤ数を示している。NNは、図2に示すように、入力層、隠れ層および出力層を有しており、これらの層のそれぞれには、複

数のノードがエッジで繋がれた構造になっている。複数のノードのそれぞれの出力値は、エッジによって繋がれた前の層のノードの出力値と、エッジの重みおよび層ごとに設定された活性化関数とから算出することができる。

[0017] NNには、例えば、全結合層 (Fully-connected layer) だけでなく、畳み込み層 (Convolutional layer) とプーリング層 (Pooling layer) とを有する畳み込みNN (CNN; Convolutional Neural Network) がある。CNNでは、データのフィルタ処理を実現するネットワークなど、分類および回帰以外のデータ処理を実現するネットワークが生成可能である。

[0018] 例えば、画像または音声を入力として、入力信号のノイズ除去または高品質化を実現する画像または音声のフィルタ処理、圧縮音声の高域が失われた音声の高域復元処理、一部の画像領域が欠損した画像の復元処理 (inpainting)、あるいは画像の超解像処理がCNNで実現可能である。CNNでは、生成モデルによって生成されたデータであるか否かを判定する識別モデルを用いてデータの真贋を判定する、生成モデルと識別モデルを組み合わせたNNを構築することもできる。

[0019] 近年では、生成モデルが、識別モデルによって真のデータでないと見破られるデータを生成しないように、識別モデルが、生成モデルによって生成されたデータを真のデータでないと見破れるように敵対的に学習された敵対的生成ネットワーク (Generative Adversarial Network) という新しいNNも提案されている。このNNでは、高精度な生成モデルおよび識別モデルを生成することが可能である。

[0020] 図3は、実施の形態1に係るデータ処理装置 (エンコーダ) の構成を示すブロック図である。図3に示すデータ処理装置は、学習用データセットと評価用データセットを用いてNNを学習し、NNの構成を示すモデル情報 (以下、モデル情報と記載する) の符号化データを生成する第1のデータ処理装置であり、例えば、図1に示したサーバ1である。

- [0021] 図3に示すデータ処理装置は、データ処理部10および符号化部11を備える。データ処理部10は、NNを学習する第1のデータ処理部であり、学習部101、評価部102および制御部103を備える。符号化部11は、学習部101によって学習されたNNのモデルを識別するモデルヘッダ情報、NNのレイヤを識別するレイヤヘッダ情報、およびレイヤ単位のエッジの重み情報が符号化された符号化データを生成する。また、符号化部11は、符号化するレイヤ（符号化レイヤ）のレイヤ構造情報を符号化し、新規レイヤフラグを符号化する。レイヤ構造情報は、NNのレイヤ構造を示す情報である。新規レイヤフラグは、当該レイヤが新たに追加されるレイヤであるか既にあるレイヤを更新したレイヤであるかを識別するためのフラグ情報であり、詳細は後述する。
- [0022] 学習部101は、学習用データセットを用いてNNの学習処理を実施し、学習したNNのモデル情報を生成する。モデル情報は、学習部101から評価部102に出力される。さらに、学習部101は、後述する制御部103によって制御された符号化用モデル情報を持ち、制御部103によって学習完了指示を受けた場合に符号化用モデル情報を符号化部11に出力する。評価部102は、モデル情報を用いてNNを生成し、生成されたNNを用いて、評価用データセットから推論処理を実施する。推論処理の結果として得られた評価指標の値が評価結果であり、評価結果は、評価部102から、制御部103に出力される。評価指標は、評価部102に設定されており、例えば、推論精度または損失関数の出力値である。
- [0023] 制御部103は、評価部102によって評価結果として得られた評価値から、学習部101によって学習されたNNのモデルの更新の有無と、学習部101によるNNの学習の完了可否とを判定し、判定結果に応じて学習部101を制御する。例えば、制御部103は、評価値をモデル更新判定基準と比較して、この比較結果に基づいて当該モデル情報を符号化用モデル情報として更新するか否かを判定する。また、制御部103は、評価値を学習完了判定基準と比較して、この比較結果に基づいて学習部101によるNNの学

習を完了するか否かを判定する。なお、これらの判定基準は、評価値の履歴から決定される。

[0024] 図4は、実施の形態1に係るデータ処理装置（デコーダ）の構成を示すブロック図である。図4に示すデータ処理装置は、図3に示した符号化部11によって生成された符号化データを復号してNNを生成し、生成されたNNを用いて、1以上の評価用データを推論処理する第2のデータ処理装置であり、例えば、図1に示したクライアント3-1, 3-2, ..., 3-Nである。

[0025] 図4に示すデータ処理装置は、復号部201および推論部202を備えている。復号部201は、符号化部11によって生成された符号化データから、モデル情報を復号する。例えば、復号部201は、図4に示すデータ処理装置において必要な情報のみを、符号化データから復号することができる。

[0026] 推論部202は、復号部201によって復号されたモデル情報を用いてNNを生成し、生成されたNNを用いたデータ処理を実施する第2のデータ処理部である。例えば、データ処理は、NNを用いた評価用データに対する推論処理である。推論部202は、NNを用いて評価用データに対する推論処理を実施し、推論結果を出力する。

[0027] 次に、実施の形態1に係るデータ処理システムの動作について説明する。図5は、実施の形態1に係るデータ処理装置（エンコーダ）の動作を示すフローチャートであり、図3に示したデータ処理装置によるデータ処理方法を示している。学習部101がNNを学習する（ステップST1）。例えば、学習部101は、学習用データセットを用いてNNの学習を実施し、この学習によって得られたモデル情報を評価部102に出力する。

[0028] モデル情報は、NNのモデルの構成を示す情報であり、レイヤごとの構成を示すレイヤ構造情報と、レイヤに属する各エッジの重み情報を含んで構成される。レイヤ構造情報には、レイヤ種別情報、レイヤ種別に関わる構成情報、およびエッジの重み以外でレイヤを構成するために必要な情報が含まれている。エッジの重み以外でレイヤを構成するために必要な情報には、例え

ば、活性化関数がある。レイヤ種別情報は、レイヤの種別を示す情報であり、レイヤ種別情報を参照することで、畳み込み層、プーリング層または全結合層といったレイヤの種別を識別することが可能である。

[0029] レイヤ種別に関わる構成情報は、レイヤ種別情報に対応する種別のレイヤの構成を示す情報である。例えば、レイヤ種別情報に対応するレイヤの種別が畳み込み層である場合、レイヤ種別に関わる構成情報には、畳み込みを行うチャネル数、畳み込みフィルタ（カーネル）のデータサイズと形状、畳み込み間隔（ストライド）、畳み込み処理の入力信号の境界に対するパディングの有無、および、パディング有りの場合はパディングの方法がある。また、レイヤ種別情報に対応するレイヤの種別がプーリング層である場合、レイヤ種別に関わる構成情報には、最大プーリングまたは平均プーリングといったプーリング方法、プーリング処理を行うカーネルの形状、プーリング間隔（ストライド）、プーリング処理の入力信号の境界に対するパディングの有無、および、パディング有りの場合はパディングの方法がある。

[0030] 各エッジの重みを示す情報には、全結合層のように各エッジで独立に重みが設定される場合がある。一方、畳み込み層のように、エッジの重みが畳み込みフィルタ（カーネル）単位（チャネル単位）で共通する、すなわち、一つのフィルタでエッジの重みが共通する場合もある。

[0031] 評価部 102 が NN を評価する（ステップ S T 2）。例えば、評価部 102 は、学習部 101 によって生成されたモデル情報を用いて NN を生成し、生成された NN を用いて、評価用データセットから推論処理を実施する。評価結果は、評価部 102 から制御部 103 に出力される。評価結果は、例えば、推論精度または損失関数の出力値である。

[0032] 次に、制御部 103 が、モデル情報を更新するか否かを判定する（ステップ S T 3）。例えば、制御部 103 は、評価部 102 によって生成された評価値がモデル更新判定基準を満たさない場合、学習部 101 が持つ符号化用モデル情報を更新しないと判定し、評価値がモデル更新判定基準を満たす場合、上記符号化用モデル情報を更新すると判定する。

- [0033] モデル更新判定基準の一例としては、評価値が損失関数の出力値である場合、学習開始時からの学習履歴における評価値の最小値よりも今回の学習による評価値が小さいこと、がある。他の一例としては、評価値が推論精度である場合、学習開始時からの学習履歴における評価値の最大値よりも今回の学習の評価値が大きいこと、がある。
- [0034] また、学習履歴の切り替え単位も任意としてもよい。例えば、後述するモデル識別番号 (`model_id`) ごとに学習履歴を持つとする。この場合、当該モデルが、後述する参照モデル識別番号 (`reference_model_id`) を持たない場合は、学習履歴なしとして学習を開始する。すなわち、1 回目のステップ ST3 では、必ずモデル情報を更新することになる。一方、当該モデルが参照モデル識別番号を持つ場合には、当該参照モデル識別番号が指し示すモデルの学習履歴 (履歴 A) を参照する。これにより、当該モデルの学習時に参照モデル識別番号が指し示すモデルより評価値が悪い (推論精度が低い、損失関数の値が大きい、など) モデルに更新されてしまうことを防ぐことが可能となる。このとき、当該モデルのモデル識別番号と参照モデル識別番号が同一である場合、当該モデルの学習を実施する度に、参照モデル識別番号に対応する学習履歴 (履歴 A) が更新されていくことになる。一方、当該モデルのモデル識別番号と参照モデル識別番号が異なる場合は、参照モデル識別番号に対応する学習履歴 (履歴 A) を当該モデルのモデル識別番号の学習履歴 (履歴 B) の初期値としてコピーした上で、当該モデルの学習を実施する度に当該モデルの学習履歴 (履歴 B) が更新されていくことになる。
- [0035] 制御部 103 によってモデル情報を更新すると判定された場合 (ステップ ST3 ; YES)、学習部 101 は、符号化用モデル情報を当該モデル情報に更新する (ステップ ST4)。例えば、制御部 103 は、モデル情報の更新があることを示すモデル更新指示情報を生成し、モデル更新指示情報を含んだ学習制御情報を学習部 101 に出力する。学習部 101 は、学習制御情報に含まれるモデル更新指示情報に従って、符号化用モデル情報を当該モデ

ル情報に更新する。

- [0036] 一方、モデル情報を更新しないと判定した場合（ステップST3；NO）、制御部103は、モデル情報の更新がないことを示すモデル更新指示情報を生成し、モデル更新指示情報を含んだ学習制御情報を学習部101に出力する。学習部101は、学習制御情報に含まれるモデル更新指示情報に従って符号化用モデル情報を更新しない。
- [0037] 次に、制御部103は、評価値を学習完了判定基準と比較し、この比較結果に基づいて学習部101によるNNの学習を完了するか否かを判定する（ステップST5）。例えば、学習完了判定基準が評価部102によって生成された評価値が特定の値に達したか否かとする場合、制御部103は、評価部102によって生成された評価値が学習完了判定基準を満たすと、学習部101によるNNの学習が完了したと判定し、評価値が学習完了判定基準を満たしていなければ、学習部101によるNNの学習が完了していないと判定する。あるいは、例えば、連続でM回（Mは1以上の予め定められた整数）、モデル情報の更新なし（ステップST3；NO）が選択されると、学習完了と判定するなどの、学習完了判定基準が直近の学習履歴に基づくものとする場合、制御部103は、学習履歴が学習完了判定基準を満たしていなければ、学習部101によるNNの学習が完了していないと判定する。
- [0038] 制御部103によってNNの学習が完了したと判定された場合（ステップST5；YES）、学習部101が符号化用モデル情報を符号化部11に出力するとともに、ステップST6の処理に移行する。一方、制御部103によってNNの学習が完了していないと判定された場合（ステップST5；NO）、ステップST1からの処理が実行される。
- [0039] 符号化部11は、学習部101から入力された符号化用モデル情報を符号化する（ステップST6）。符号化部11は、学習部101によって生成された符号化用モデル情報を、NNのレイヤ単位で符号化し、ヘッダ情報とレイヤ単位の符号化データから構成された符号化データを生成する。また、符号化部11は、レイヤ構造情報を符号化し、新規レイヤフラグを符号化する

。

[0040] 図6は、実施の形態1に係るデータ処理装置（デコーダ）の動作を示すフローチャートであり、図4に示したデータ処理装置の動作を示している。復号部201は、符号化部11によって符号化された符号化データからモデル情報を復号する（ステップST11）。次に、推論部202は、復号部201によって復号されたモデル情報からNNを生成する（ステップST12）。推論部202は、生成されたNNを用いて、評価用データに対する推論処理を実施し、推論結果を出力する（ステップST13）。

[0041] 次に、図5のステップST6における符号化部11によるモデル情報の符号化について詳細に説明する。符号化部11によるモデル情報の号化には、例えば（1）または（2）の符号化方法を用いることができる。あるいは、（1）または（2）の符号化がパラメータ毎にどちらを用いるか定義されていてもよい。例えば、ヘッダ情報は（1）、重みデータは（2）とすることで、デコーダは、ヘッダ情報を可変長復号することなく容易に解析可能としつつ、符号化データのデータサイズの大部分を占める重みデータは可変長復号によって高い圧縮を実現することができ、符号化データ全体のデータサイズを抑えることができる。

（1）モデル情報に含まれる各情報を構成するパラメータが、パラメータに定義されているビット精度で記述されたビット列そのものが、ヘッダ情報が存在する場合はヘッダ情報を含めて予め設定された順序で並べられたデータを符号化データとする。ビット精度は、例えば、`int`型8ビットあるいは`float`型32ビットといった、パラメータに定義されているビット精度である。

（2）モデル情報に含まれる各情報を構成するパラメータが、パラメータごとに設定された可変長符号化方法によって符号化されたビット列そのものが、ヘッダ情報を含めて予め設定された順序で並べられたデータを符号化データとする。

[0042] 図7は、実施の形態1における符号化データの例を示す図であり、上記（

1) または (2) の符号化データは、図 7 に示す順序で並べてもよい。図 7 に示す符号化データは、データユニットと呼ぶデータの集まりから構成され、データユニットには、非レイヤデータユニットとレイヤデータユニットがある。レイヤデータユニットは、レイヤ単位の符号化データであるレイヤデータが格納されるデータユニットである。

[0043] レイヤデータは、スタートコード、データユニットタイプ、レイヤ情報ヘッダ、および重みデータから構成される。レイヤ情報ヘッダは、NNのレイヤを識別するためのレイヤヘッダ情報が符号化されたものである。重みデータは、レイヤ情報ヘッダが示すレイヤに属するエッジの重み情報が符号化されたものである。なお、図 7 に示す符号化データにおいて、各レイヤデータユニットの並び順は必ずしもNNの各層の並び順と同じでなくてもよく、任意である。これは、後述するレイヤ識別番号 (layer_id) によって、各レイヤデータユニットがNNのどの位置のレイヤであるかを識別可能であるからである。

[0044] 非レイヤデータユニットは、レイヤデータ以外のデータが格納されるデータユニットである。例えば、非レイヤデータユニットには、スタートコード、データユニットタイプ、およびモデル情報ヘッダが格納されている。モデル情報ヘッダは、NNのモデルを識別するためのモデルヘッダ情報が符号化されたものである。

[0045] スタートコードは、データユニットの先頭位置に格納され、データユニットの先頭位置を識別するためのコードである。クライアント 3-1, 3-2, ..., 3-N (以下、復号側と記載する) は、スタートコードを参照することにより、非レイヤデータユニットまたはレイヤデータユニットの先頭位置を特定することが可能である。例えば、スタートコードとして 0x000001 が定義された場合、データユニットに格納されたスタートコード以外のデータは、0x000001 が発生しないように設定される。これにより、スタートコードからデータユニットの先頭位置を特定することができる。

- [0046] 0×000001が発生しないように設定するためには、例えば、0×000000～0×000003の符号化データにおける3バイト目に03を挿入して0×000300～0×000303とし、復号するときに、0×0003を0×0000と変換することにより、元に戻すことができる。なお、スタートコードは、一意に識別可能なビット列であれば、0×000001以外のビット列をスタートコードとして定義してもよい。また、データユニットの先頭位置を識別可能な方法であれば、スタートコードを用いなくてもよい。例えば、データユニットの終端であることを識別可能なビット列をデータユニットの終端に付けてもよい。あるいは、非レイヤデータユニットの先頭のみスタートコードを付けることとし、モデル情報ヘッダの一部として、各レイヤデータユニットのデータサイズを符号化するようにしてもよい。このようにすることで、上記情報から、各レイヤデータユニットの区切り位置を識別することが可能である。
- [0047] データユニットタイプは、データユニットにおいてスタートコードの次に格納されて、データユニットの種類を識別するためのデータである。データユニットタイプは、データユニットの種類ごとに予め値が定義されている。復号側は、データユニットに格納されたデータユニットタイプを参照することで、データユニットが、非レイヤデータユニットであるのか、レイヤデータユニットであるのかを識別でき、さらに、どのような非レイヤデータユニットまたはレイヤデータユニットであるのかを識別することが可能である。
- [0048] 非レイヤデータユニットにおけるモデル情報ヘッダには、モデル識別番号(model__id)、モデル内レイヤデータユニット数(num__layers)および符号化レイヤデータユニット数(num__coded__layers)が含まれる。モデル識別番号は、NNのモデルを識別するための番号である。従って、基本的には、個々のモデルにおいて互いに独立した番号を持つが、もし実施の形態1に係るデータ処理装置(デコーダ)が過去に受信したモデルと同一のモデル識別番号を持つモデルを新たに受信した場合は、当該モデル識別番号を持つモデルが上書きされることになる。モデル内

レイヤデータユニット数は、モデル識別番号で識別されるモデルを構成するレイヤデータユニットの数である。符号化レイヤデータユニット数は、符号化データの中に、実際に存在するレイヤデータユニットの数である。図7の例では、レイヤデータユニット(1)～(n)が存在することから、符号化レイヤデータユニット数はnである。なお、符号化レイヤデータユニット数は、必ず、モデル内レイヤデータユニット数以下になる。

[0049] レイヤデータユニットにおけるレイヤ情報ヘッダには、レイヤ識別番号(`layer__id`)およびレイヤ構造情報が含まれる。レイヤ識別番号は、レイヤを識別するための番号である。レイヤ識別番号によってどの層のレイヤか識別できるように、レイヤ識別番号の値の振り方は予め固定的に定義される。例えば、NNの入力層を0、次の層を1というように、入力層に近い層から順に番号を振る、などである。レイヤ構造情報は、NNのレイヤごとの構成を示す情報であって、レイヤ種別情報、レイヤ種別に関わる構成情報、およびエッジの重み以外にレイヤを構成するために必要な情報を含んでいる。例えば、後述する`model__structure__information`と`layer__id__information`の当該レイヤ部分のみの情報である。さらに、レイヤ構造情報として当該レイヤの各エッジの重みのビット精度を示す`weight__bit__length`を持つ。例えば、`weight__bit__length`=8であれば、重みは8ビットのデータであることを示す。したがって、レイヤ単位にエッジの重みのビット精度を設定することができる。これによって、レイヤの重要度(ビット精度が出力結果に影響する程度)に応じてレイヤ単位にビット精度を変更する等の適応制御が可能となる。

[0050] なお、これまでレイヤ構造情報を含むレイヤ情報ヘッダを示したが、モデル情報ヘッダが、符号化データに含まれる全てのレイヤ構造情報(`model__structure__information`)と本レイヤ構造情報に対応するレイヤ識別情報(`layer__id__information`)とを含んでもよい。復号側は、モデル情報ヘッダを参照することで、各レイヤ

識別番号のレイヤの構成を特定することができる。さらに、上記の場合はモデル情報ヘッダを参照することで、各レイヤ識別番号のレイヤの構成を特定することができるため、レイヤ情報ヘッダは、レイヤ識別番号のみを持つようにしてもよい。このようにすることで、レイヤデータユニットのデータサイズが非レイヤデータユニットのデータサイズよりも大きい場合、各レイヤデータユニットのデータサイズを小さくすることができ、符号化データ内のデータユニットの最大データサイズを小さくすることができる。

[0051] レイヤデータユニットにおいて、レイヤ情報ヘッダの次に、レイヤ単位に符号化された重みデータが格納されている。重みデータは、非零フラグおよび非零重みデータを含んでいる。非零フラグは、エッジの重みの値が零か否かを示すフラグであり、対応するレイヤに属する全てのエッジの重みについての非零フラグが設定される。

[0052] 非零重みデータは、重みデータにおいて非零フラグに続いて設定されるデータであり、非零フラグが非零（有意）を示す重みについて、その重みの値が設定されたものである。図7において、それぞれが非零の重みの値を示す重みデータ（1）～重みデータ（m）が非零重みデータとして設定されている。非零の重みデータ数mは、対応するレイヤlの全ての重みの数 M_l 以下である。なお、重みの値が非零のエッジが疎であるレイヤに関する重みデータは、非零重みデータが少なく、ほぼ非零フラグのみとなるため、重みデータのデータサイズが大きく削減される。

[0053] 図8は、実施の形態1における符号化データの別の例を示す図であり、上記（1）または（2）の符号化データは図8に示す順序で並べてもよい。図8に示す符号化データは、重みデータのデータ構成が図7と異なっており、非零重みデータには、対応するレイヤに属する全てのエッジの重みが上位ビットから順にビットプレーンごとにまとめて並べられている。さらに、レイヤ情報ヘッダには、エッジの重みを示す各ビットの先頭位置を示すビットプレーンデータ位置識別情報が設定されている。

[0054] 例えば、エッジの重みに定義されたビット精度がXであると、対応するレ

イヤに属する全てのエッジの重みは、ビット精度 X でそれぞれ記述される。符号化部11は、これらの重みのビット列のうち、1ビット目の非零重みデータである、1ビット目の重みデータ(1), 1ビット目の重みデータ(2), ..., 1ビット目の重みデータ(m)を、1ビット目の各非零重みデータに設定する。この処理は、2ビット目の非零重みデータから X ビット目の非零重みデータまで繰り返される。なお、1ビット目の重みデータ(1), 1ビット目の重みデータ(2), ..., 1ビット目の重みデータ(m)は、1ビット目のビットプレーンを構成する非零の重みデータである。

[0055] 復号側は、ビットプレーンデータ位置識別情報に基づいて、レイヤ単位の符号化データのうち、必要な符号化データを特定し、特定された符号化データを任意のビット精度で復号することができる。すなわち、復号側は、符号化データから必要な符号化データのみを選別でき、復号側の環境に応じたNNのモデル情報を復号することが可能である。なお、ビットプレーンデータ位置識別情報は、ビットプレーンデータ間の区切り位置を識別可能な情報であればよく、各ビットプレーンデータの先頭位置を示す情報であってもよいし、各ビットプレーンデータのデータサイズを示す情報であってもよい。

[0056] NNの構成を示す全ての符号化データを復号側へ伝送するためには、データ伝送ネットワーク2の伝送帯域が十分でない場合に、符号化部11が、当該符号化データのうち、データ伝送ネットワーク2の伝送帯域に応じて伝送する非零重みデータを制限してもよい。例えば、32ビット精度で記述された重み情報のビット列のうち、上位8ビットの非零重みデータを伝送対象とする。復号側は、この非零重みデータの次に並ぶスタートコードから、符号化データにおいて、8ビット目の非零重みデータの後に、次のレイヤに対応するレイヤデータユニットが並んでいることを認識できる。また、復号側は、重みデータにおける非零フラグを参照することで、値が零の重みを正しく復号することができる。

[0057] 復号側で任意のビット精度で重みデータが復号されたときに、そのビット精度での推論精度を改善するため、符号化部11は、各ビット精度で復号さ

れたときの重みに加算するオフセットをレイヤ情報ヘッダに含めてもよい。例えば、符号化部11は、ビット精度で記述された重みのビット列に対してレイヤ単位に一律なオフセットを加算し、最も高精度になるオフセットを求めて、求められたオフセットを、レイヤ情報ヘッダに含めて符号化する。

[0058] また、符号化部11は、NNが備える全てのレイヤにおけるエッジの重みのオフセットをモデル情報ヘッダに含めて符号化してもよい。さらに、符号化部11は、オフセットを含むか否かを示すフラグをレイヤ情報ヘッダまたはモデル情報ヘッダに設定し、例えば、フラグが有効である場合のみ、オフセットを符号化データに含めてもよい。

[0059] 符号化部11は、エッジの重みの値と特定の値の差分を符号化対象としてもよい。特定の値としては、例えば、符号化順が一つ前の重みが挙げられる。また、一つ上位のレイヤ（入力層に近いレイヤ）に属する、対応するエッジの重みを特定の値としてもよいし、更新前のモデルの対応するエッジの重みを特定の値としてもよい。

[0060] さらに、符号化部11は、(A)、(B)および(C)に示す機能を有する。

(A) 符号化部11は、ベース符号化データとエンハンスメント符号化データとに分けて符号化するスケラブル符号化機能を有する。

(B) 符号化部11は、基準のNNにおけるエッジの重みとの差分を符号化する機能を有する。

(C) 符号化部11は、基準のNNにおける部分的な情報（例えば、レイヤ単位の情報）のみを、NNの更新用情報として符号化する機能を有する。

[0061] (A)の例について説明する。

符号化部11は、エッジの重みについて予め定義された量子化手法を用いて、エッジの重みを量子化し、量子化後の重みを符号化したデータをベース符号化データとし、量子化誤差を重みとみなして符号化したデータをエンハンスメント符号化データとする。ベース符号化データとされた重みは、量子化によって量子化前の重みよりもビット精度が低下するため、データサイズ

が削減される。復号側へ符号化データを伝送する伝送帯域が十分でない場合に、実施の形態1に係るデータ処理装置は、ベース符号化データのみを復号側に伝送する。一方、復号側へ符号化データを伝送する伝送帯域が十分な場合、実施の形態1に係るデータ処理装置は、ベース符号化データに加え、エンハンスメント符号化データも含めて復号側に伝送する。

[0062] エンハンスメント符号化データは2つ以上とすることができる。例えば、符号化部11は、量子化誤差をさらに量子化したときの量子化値を、一つ目のエンハンスメント符号化データとし、その量子化誤差を2つ目のエンハンスメント符号化データとする。さらに、2つ目のエンハンスメント符号化データの量子化誤差をさらに量子化した量子化値とその量子化誤差とに分けて目的のエンハンスメント符号化データの数になるように符号化してもよい。このように、スケーラブル符号化を用いることで、データ伝送ネットワーク2の伝送帯域と伝送許容時間とに応じた符号化データの伝送が可能である。

[0063] なお、符号化部11は、図8に示した非零重みデータの上位Mビットまでをベース符号化データとして符号化し、残りのビット列を1以上に分割して1以上のエンハンスメント符号化データとしてもよい。この場合、符号化部11は、ベース符号化データとエンハンスメント符号化データのそれぞれで非零フラグを再び設定する。上位ビットのエンハンスメント符号化データにおいて0となった重みは、必ず0となる。

[0064] (B)の例について説明する。

符号化部11は、学習部101による再学習前のNNのモデルが存在する場合、再学習後のNNのモデルにおけるエッジの重みと、再学習前のモデルにおける対応するエッジの重みとの差分を符号化してもよい。なお、再学習には、転移学習または追加学習がある。データ処理システムにおいて、高い頻度でNNの構成を更新するか、あるいは再学習ごとの学習データの分布の変化が小さい場合、エッジの重みの差分が小さいので、再学習後の符号化データのデータサイズが削減される。

[0065] 符号化部11は、モデル識別番号に加え、参照すべき更新前のモデルを識

別するための参照モデル識別番号 (`reference_model_id`) をモデル情報ヘッダに含む。(B) の例において、上記参照モデル識別番号から再学習前のモデルを識別することが可能となる。さらに、符号化部 11 は、符号化データに参照元があるか否かを示すフラグ (`reference_model_present_flag`) を、モデル情報ヘッダに設定してもよい。このとき、符号化部 11 は、まず上記フラグ (`reference_model_present_flag`) を符号化し、上記フラグがモデルの更新用の符号化データであることを示す場合にのみ、さらにモデル情報ヘッダに参照モデル識別番号を設定する。

[0066] 例えば、図 1 に示したデータ処理システムにおいて、クライアント間で NN の更新頻度が異なるか、互いに異なるモデルの NN を用いてデータ処理を実施する場合であっても、クライアントは、参照モデル識別番号を参照することで、どのモデルに対する更新用の符号化データであるのかを正しく識別することができる。参照モデル識別番号からクライアント側にはないモデルの更新用の符号化データであることが識別された場合には、クライアントが、そのことをサーバ 1 に伝えることも可能である。

[0067] (C) の例について説明する。

学習部 101 は、再学習前の NN のモデルが存在する場合、例えば `Fine-tuning` を目的として、NN の上位 (入力層側) から 1 以上の任意のレイヤを固定し、一部のレイヤのみを再学習することがある。この場合、符号化部 11 は、再学習によって更新されたレイヤの構成を示す情報のみを符号化する。これにより、NN の更新において、復号側へ伝送される符号化データのデータサイズが削減される。なお、符号化データにおける符号化レイヤデータユニット数 (`num_coded_layers`) は、モデル内レイヤデータユニット数 (`num_layers`) 以下となる。復号側では、モデル情報ヘッダに含まれる参照モデル識別番号と、レイヤ情報ヘッダに含まれるレイヤ識別番号とを参照することで、更新すべきレイヤを特定できる。

[0068] 次に、学習部 101、評価部 102 および推論部 202 によるデータ処理を説明する。

図 9 は、実施の形態 1 における 1 次元データの畳み込み処理の例を示す図であり、1 次元データの畳み込み処理を行う畳み込み層を示している。1 次元データには、例えば、音声データ、時系列データがある。図 9 に示す畳み込み層は、前層に 9 つのノード 10-1 ~ 10-9、次層に 3 つのノード 11-1 ~ 11-3 を備えている。エッジ 12-1, 12-6, 12-11 には同じ重みが付与されており、エッジ 12-2, 12-7, 12-12 には同じ重みが付与されており、エッジ 12-3, 12-8, 12-13 には同じ重みが付与されており、エッジ 12-4, 12-9, 12-14 には同じ重みが付与されており、エッジ 12-5, 12-10, 12-15 には同じ重みが付与されている。また、エッジ 12-1 から 12-5 までの重みは全て異なる値となる場合もあるし、複数の重みが同じ値となる場合もある。

[0069] 前層の 9 つのノード 10-1 ~ 10-9 のうち、5 つのノードが、上記の重みで次層の 1 つのノードに繋がっている。カーネルサイズ K は 5 であり、カーネルは、これらの重みの組み合わせによって規定される。例えば、図 9 に示すように、ノード 10-1 は、エッジ 12-1 を介してノード 11-1 に繋がり、ノード 10-2 は、エッジ 12-2 を介してノード 11-1 に繋がり、ノード 10-3 は、エッジ 12-3 を介してノード 11-1 に繋がり、ノード 10-4 は、エッジ 12-4 を介してノード 11-1 に繋がり、ノード 10-5 は、エッジ 12-5 を介してノード 11-1 に繋がっている。カーネルは、エッジ 12-1 ~ 12-5 の重みの組み合わせによって規定される。

[0070] ノード 10-3 は、エッジ 12-6 を介してノード 11-2 に繋がり、ノード 10-4 は、エッジ 12-7 を介してノード 11-2 に繋がり、ノード 10-5 は、エッジ 12-8 を介してノード 11-2 に繋がり、ノード 10-6 は、エッジ 12-9 を介してノード 11-2 に繋がり、ノード 10-7 は、エッジ 12-10 を介してノード 11-2 に繋がっている。カーネルは

、エッジ12-6~12-10の重みの組み合わせによって規定される。

[0071] ノード10-5は、エッジ12-11を介してノード11-3に繋がり、ノード10-6は、エッジ12-12を介してノード11-3に繋がり、ノード10-7は、エッジ12-13を介してノード11-3に繋がり、ノード10-8は、エッジ12-14を介してノード11-3に繋がり、ノード10-9は、エッジ12-15を介してノード11-3に繋がっている。カーネルは、エッジ12-11~12-15の重みの組み合わせによって規定される。

[0072] 学習部101、評価部102および推論部202は、CNNを用いた入力データの処理において、畳み込み層のエッジの重みの組み合わせを用いて、カーネルごとにステップ数の間隔（図9では、 $S=2$ ）で畳み込み演算を実施する。エッジの重みの組み合わせは、カーネルごとに学習によって決定される。なお、画像認識用途のCNNでは、複数のカーネルを有する畳み込み層でNNが構成される場合が多い。

[0073] 図10は、実施の形態1における2次元データの畳み込み処理の例を示す図であり、画像データといった2次元データの畳み込み処理を示している。図10に示す2次元データのうち、カーネル20は、 x 方向のサイズが K_x 、 y 方向のサイズが K_y のブロック領域である。カーネルサイズ K は、 $K=K_x \times K_y$ である。学習部101、評価部102または推論部202は、2次元データにおいて、 x 方向ステップ数 S_x の間隔および y 方向ステップ数 S_y の間隔で、カーネル20ごとのデータの畳み込み演算を実施する。ここで、ステップ S_x 、 S_y は1以上の整数である。

[0074] 図11は、NNの全結合層である l （ $l=1, 2, \dots, L$ ）層目のレイヤにおけるノードごとのエッジの重み情報のマトリクスを示す図である。図12は、NNの全結合層である l （ $l=1, 2, \dots, L$ ）層目のレイヤにおけるノードごとのエッジの重み情報の量子化ステップのマトリクスを示す図である。

[0075] NNにおいては、図11に示すレイヤごとの重み w_{ij} の組み合わせが、ネ

ットワークを構成するデータとなる。このため、ディープニューラルネットワークのような多層のNNでは、一般的に数百Mbyte以上のデータ量となり、大きなメモリサイズも必要となる。 i は、ノードインデックスであり、 $i = 1, 2, \dots, N_l$ である。 j は、エッジインデックスであり、 $j = 1, 2, \dots, N_{l-1} + 1$ (オフセットを含む) である。

[0076] そこで、実施の形態1に係るデータ処理装置では、エッジの重み情報のデータ量を削減するため、重み情報を量子化する。例えば、図12に示すように、量子化ステップ q_{ij} は、エッジの重み w_{ij} ごとに設定される。量子化ステップは、複数のノードインデックスまたは複数のエッジインデックスであってもよいし、複数のノードインデックスとエッジインデックスとが共通化されてもよい。これにより、符号化すべき量子化情報が削減される。

[0077] 図13は、畳み込み層におけるエッジの重み情報のマトリクスを示す図である。図14は、畳み込み層におけるエッジの重み情報の量子化ステップのマトリクスを示す図である。畳み込み層では、1つのカーネルに対するエッジの重みは、全てのノードで共通であり、ノード一つ当たり結合するエッジ数、すなわちカーネルサイズ K を小さくしてカーネルを小領域にすることができ。図13は、エッジの重み $w_{i'j'}$ がカーネルごとに設定されたデータであり、図14は、量子化ステップ $q_{i'j'}$ がカーネルごとに設定されたデータである。なお、 i' はカーネルインデックスであり、 $i' = 1, 2, \dots, M_l$ ($l = 1, 2, \dots, L$) である。 j' はエッジインデックスであり、 $j' = 1, 2, \dots, K_l + 1$ (オフセットを含む) である。

[0078] 量子化ステップは、複数のカーネルインデックス、複数のエッジインデックス、または複数のカーネルインデックスとエッジインデックスで共通化されてもよい。これにより、符号化すべき量子化情報が削減される。例えば、レイヤ内の全ての量子化ステップを共通化して、一つのレイヤで一つの量子化ステップとしてもよいし、モデル内の全ての量子化ステップを共通化して、一つのモデルで一つの量子化ステップとしてもよい。

[0079] 図15は、実施の形態1に係るデータ処理装置(エンコーダ)の変形例の

構成を示すブロック図である。図15に示すデータ処理装置は、学習用データセットと評価用データセットを用いてNNを学習し、NNのモデル情報の符号化データを生成する第1のデータ処理装置であり、例えば、図1に示したサーバ1である。図15に示すデータ処理装置は、データ処理部10A、符号化部11および復号部12を備えている。

[0080] データ処理部10Aは、NNを生成して学習するデータ処理部であって、学習部101A、評価部102および制御部103を備える。符号化部11は、学習部101によって生成されたモデル情報を符号化し、ヘッダ情報とレイヤ単位の符号化データから構成された符号化データを生成する。復号部12は、符号化部11によって生成された符号化データからモデル情報を復号する。また、復号部12は、復号済みのモデル情報を学習部101Aに出力する。

[0081] 学習部101Aは、学習部101と同様に、学習用データセットを用いてNNの学習を実施し、学習されたNNの構成を示すモデル情報を生成する。また、学習部101Aは、復号済みのモデル情報を用いてNNを生成し、学習用データセットを用いて、生成されたNNのパラメータを再学習する。

[0082] 上記再学習の際、一部のエッジの重みを固定して再学習することで、符号化データのデータサイズを小さく保ったまま高精度化することが可能である。例えば、非零フラグが0の重みは0に固定した状態で再学習を実施することで、再学習前のエッジの重みに係る符号化データのデータサイズ以上となることを防ぎながら重みの最適化が可能となる。

[0083] データ処理装置が復号部12を備え、データ処理部10Aが、復号部12によって復号された情報を用いてNNを学習する。これにより、例えば、符号化歪みが発生する非可逆符号化を符号化部11が行う場合であっても、当該データ処理装置は、符号化データの実際の復号結果に基づいてNNを生成して学習することができ、符号化データのデータサイズに対する制約を課した状況下で、符号化誤差の影響を最小限に抑えたNNの学習が可能である。

[0084] 図1と同様の構成を有し、サーバ1として、図3に示したデータ処理装置

を備え、クライアント3-1, 3-2, . . . , 3-Nとして、図4に示したデータ処理装置を備えたデータ処理システムにおいて、NNの中間レイヤから出力されるデータは、下記参考文献1に記載された画像検索(retrieval)またはマッチング(matching)を一例とした、画像データおよび音声データに対するデータ処理の特徴量として用いることができる。

(参考文献1) ISO/IEC JTC1/SC29/WG11/m39219, "Improved retrieval and matching with CNN feature for CDVA", Chengdu, China, Oct. 2016.

[0085] 例えば、画像検索、マッチングまたは物体追跡といった画像処理の画像特徴量として、NNの中間レイヤの出力データを用いる場合、従来の上記画像処理で用いられていた画像特徴量であるHOG(Histogram of Oriented Gradients)、SIFT(Scale Invariant Feature Transform)、または、SURF(Speeded Up Robust Features)に対する画像特徴量の置き換えあるいは追加が行われる。これにより、従来の特徴量を用いた画像処理と同じ処理手順で当該画像処理を実現できる。実施の形態3に係るデータ処理システムにおいて、符号化部11は、画像特徴量を出力する中間レイヤまでのNNの構成を示すモデル情報を符号化する。

[0086] さらに、サーバ1として機能するデータ処理装置は、上記データ処理の特徴量を用いて画像検索等のデータ処理を行う。クライアントとして機能するデータ処理装置は、符号化データから中間レイヤまでのNNを生成し、生成されたNNの中間レイヤから出力されたデータを特徴量として用いて、画像検索などのデータ処理を実施する。

[0087] データ処理システムにおいて、符号化部11が、NNの中間レイヤまでの構成を示すモデル情報を符号化することによって、量子化によるパラメータデータの圧縮率が高まり、符号化前の重み情報のデータ量を削減することが

できる。クライアントは、復号部201によって復号されたモデル情報を用いてNNを生成し、生成されたNNの中間レイヤから出力されたデータを特徴量として用いたデータ処理を行う。

[0088] また、実施の形態1に係るデータ処理システムは、図1と同様の構成を有し、サーバ1として、図3または図15に示したデータ処理装置を備え、クライアント3-1, 3-2, ..., 3-Nとして、図4に示したデータ処理装置を備えることができる。この構成を有したデータ処理システムにおいて、符号化データには、新規レイヤフラグ(`new__layer__flag`)が設定されている。新規レイヤフラグが0(無効)である場合は、新規レイヤフラグに対応するレイヤは、参照レイヤを基準として更新されるレイヤである。新規レイヤフラグが1(有効)である場合は、新規レイヤフラグに対応するレイヤは、新規に追加されるレイヤである。

[0089] 新規レイヤフラグが0(無効)である場合、新規レイヤフラグに対応するレイヤに対してチャンネル単位にエッジの重みの更新有無を識別するためのフラグ(`channel__wise__update__flag`)が設定される。このフラグが0(無効)であれば、全てのチャンネルのエッジの重みが符号化される。このフラグが1(有効)であれば、チャンネル単位の重みの更新フラグ(`channel__update__flag`)が設定される。この更新フラグは、チャンネルごとに参照レイヤからの更新の有無を示すフラグである。この更新フラグが1(有効)である場合、チャンネルの重みが符号化され、0(無効)であれば、参照レイヤと同一の重みとされる。

[0090] さらに、レイヤ情報ヘッダとして、レイヤのチャンネル数を示す情報(`num__channels`)、チャンネル単位のエッジの重みの数を示す情報(`weights__per__channels`)が設定される。あるレイヤ l の`weights__per__channels`は、カーネルサイズ $K_l + 1$ あるいは一つ前の層であるレイヤ $l - 1$ からのエッジ数 $N_{l-1} + 1$ となる。

[0091] 符号化データが、前述した新規レイヤフラグを有することで、レイヤデータユニットの符号化データのみから、チャンネル数とチャンネル単位の重みの数

を特定することが可能である。従って、レイヤデータユニットの復号処理として、チャンネル単位の重みの更新フラグを復号することができる。

[0092] また、チャンネル単位に重みの更新有無を識別するためのフラグが1（有効）に設定される場合は、参照レイヤとチャンネル数とが同じであるときに制約される。これは、参照レイヤとチャンネル数が異なる場合、参照レイヤと、上記フラグに対応するレイヤとの間で、各チャンネルの対応関係が不明になるためである。

[0093] 図16は、実施の形態1における符号化データの更新の概要を示す図である。図16において、上側に示すデータは、非レイヤデータユニットと、レイヤデータユニット（1）～（4）とから構成され、図7と同様に、レイヤデータユニット（4）から順番に符号化される。非レイヤデータユニットには、モデルヘッダ情報として、モデル識別番号（`model__id`）=0、モデル内レイヤデータユニット数（`num__layers`）=4、レイヤ構造情報（`model__structure__information`）およびレイヤ識別情報（`layer__id__information`）が設定され、符号化データに参照元があるか否かを示すフラグ（`reference__model__present__flag`）に0（無効）が設定されている。

[0094] レイヤデータユニット（1）において、レイヤ識別番号（`layer__id`）には0が設定され、レイヤのチャンネル（フィルタ、カーネル）数を示す情報（`num__channels`）に32が設定され、チャンネル（フィルタ、カーネル）単位の重みの数（バイアス値を含む）を示す情報（`weights__per__channels`）に76が設定されている。また、レイヤデータユニット（2）において、レイヤ識別番号（`layer__id`）には1が設定され、レイヤのチャンネル数を示す情報（`num__channels`）に64が設定され、チャンネル単位の重みの数を示す情報（`weights__per__channels`）に289が設定されている。

[0095] レイヤデータユニット（3）において、レイヤ識別番号（`layer__i`

d) には2が設定され、レイヤのチャンネル数を示す情報 (num__channels) に128が設定され、チャンネル単位の重みの数を示す情報 (weights__per__channels) に577が設定されている。また、レイヤデータユニット (4) において、レイヤ識別番号 (layer__id) には3が設定され、レイヤのチャンネル数を示す情報 (num__channels) に100が設定され、チャンネル単位の重みの数を示す情報 (weights__per__channels) に32769が設定されている。

[0096] 図16において、下側に示すデータは、レイヤ構造情報、レイヤ更新フラグおよび新規レイヤフラグを用いて、上側に示すデータから更新されたデータであり、非レイヤデータユニットと、レイヤデータユニット (1')、(2)、(3)、(5)、(4') とから構成される。上側に示すデータが伝送されたクライアントに対しては、非レイヤデータユニットと、レイヤデータユニット (1')、(5)、(4') を送信する必要があるが (Need to transmit)、レイヤデータユニット (2) および (3) は更新されておらず、送信する必要がない (No need to transmit)。

[0097] 図16の下側に示す非レイヤデータユニットには、モデルヘッダ情報として、モデル識別番号 (model__id) = 10、モデル内レイヤデータユニット数 (num__layers) = 5、レイヤ構造情報 (model__structure__information) およびレイヤ識別情報 (layer__id__information) が設定され、符号化データに参照元があるか否かを示すフラグ (reference__model__present__flag) に1 (有効) が設定され、参照モデル識別番号 (reference__model__id) に0が設定され、符号化レイヤデータユニット数 (num__coded__layers) に3が設定されている。

[0098] レイヤデータユニット (1') において、レイヤ識別番号 (layer__id) は0であり、新規レイヤフラグ (new__layer__flag) に0が設定され、レイヤのチャンネル数を示す情報 (num__channels

）に32が設定され、チャンネル単位の重みの数を示す情報（`weights_per_channels`）に76が設定されている。また、チャンネル単位に重みの更新有無を識別するためのフラグ（`channel_wise_update_flag`）には1（有効）が設定されているので、チャンネル単位の重みの更新フラグ（`channel_update_flag`）が設定されている。

[0099] レイヤ識別番号（`layer_id`）が1であるレイヤデータユニット（2）およびレイヤ識別番号（`layer_id`）が2であるレイヤデータユニット（3）は、更新対象ではないため、符号化データに含まれない。よって、上記モデルヘッダ情報は、モデル内レイヤデータユニット数（`num_layers`）=5、符号化レイヤデータユニット数（`num_coded_layers`）=3が設定されている。

[0100] レイヤデータユニット（5）において、レイヤ識別番号（`layer_id`）は4であり、新規レイヤフラグ（`new_layer_flag`）に1（有効）が設定されている。また、レイヤのチャンネル数を示す情報（`num_channels`）には256が設定され、チャンネル単位の重みの数を示す情報（`weights_per_channels`）に1153が設定されている。

[0101] レイヤデータユニット（4'）において、レイヤ識別番号（`layer_id`）は3であり、新規レイヤフラグ（`new_layer_flag`）に0が設定され、レイヤのチャンネル数を示す情報（`num_channels`）に100が設定され、チャンネル単位の重みの数を示す情報（`weights_per_channels`）に16385が設定されている。また、チャンネル単位に重みの更新有無を識別するためのフラグ（`channel_wise_update_flag`）には0（無効）が設定され、チャンネル単位の重みに更新はない。

[0102] 下側に示すデータでは、上側に示すデータにおけるレイヤデータユニット（1）および（4）が、レイヤデータユニット（1'）および（4'）に更

新されており、さらに、レイヤ識別番号が4であるレイヤデータユニット（5）が追加されている。

[0103] 図17は、図16に示す符号化データの更新に対応したネットワークモデルの構成を示す図である。図17において、左側に示すネットワークモデルが、図16の上側に示すデータを復号して実現されるネットワークモデルである。また、右側に示すネットワークモデルが、図16の下側に示すデータを復号して実現されるネットワークモデルである。

[0104] レイヤデータユニット（1'）は、チャンネル単位に重みの更新有無を識別するためのフラグ（`channel_wise_update_flag`）が1であるので、レイヤデータユニット（1）から、いくつかのチャンネルの重みが更新されている。また、レイヤデータユニット（5）が追加され、レイヤデータユニット（4）からレイヤデータユニット（4'）に更新されたことにより、右側に示すネットワークモデルでは、全結合層（`Fully Connected layer`）までの間に、2D畳み込み層（`2D convolution layer`）および2D最大プーリング層（`2D max pooling layer`）が追加されている。

[0105] 図18は、モデル情報ヘッダに含まれるレイヤ構造情報の例を示す図である。モデル情報ヘッダに含まれる全レイヤ構造情報（`model_structure_information`）として、図18に示すようなテキスト情報を設定してもよい。図18に示すテキスト情報は、NNEF（`Neural Network Exchange Format`）という、参考文献2に記載される標準規格によるモデルのレイヤ構造を示すテキスト情報である。

（参考文献2）“`Neural Network Exchange Format`”，`The Khronos NNEF Working Group`，`Version 1.0`，`Revision 3`，`2018-06-13`。

[0106] 図18において、（A）`model_id=0`のネットワークモデルは、

図16の上側に示したデータに対応するネットワークモデル（図17の左側に示したネットワークモデル）である。（B）`model_id=10`のネットワークモデルは、図16の下側に示したデータに対応するネットワークモデル（図17の右側に示したネットワークモデル）である。

[0107] 図19は、モデル情報ヘッダに含まれるレイヤ構造情報に対応するレイヤ識別情報（`layer_id_information`）の例を示す図であり、図18のレイヤ構造情報に対応するレイヤ識別番号が設定されたレイヤ識別情報を示している。図19において、（A）`model_id=0`のネットワークモデルは、図17の左側に示したネットワークモデルに対応するレイヤ識別情報である。（B）`model_id=10`のネットワークモデルは、図17の右側に示したネットワークモデルに対応するレイヤ識別情報である。各レイヤの重みおよびバイアス値がレイヤ識別番号に割り当てられ、その値は図16に示したデータに対応している。

[0108] 全レイヤ構造情報である`model_structure_information`と全レイヤ構造情報に対応するレイヤ識別番号を示す情報である`layer_id_information`が記載されたファイル等のファイルデータは、モデル情報ヘッダにおいて、それぞれ上記ファイルデータのバイト数を示す情報の後に挿入する形で符号化データを構成する。あるいは、上記ファイルデータの入手先を示すURL（Uniform Resource Locator）をモデル情報ヘッダに含める構成も可能である。さらに、これらの構成のいずれかを選択できるように、いずれの構成であるかを識別するフラグを、モデル情報ヘッダにおける上記ファイルデータまたはURLの前に設定してもよい。上記識別フラグは、`model_structure_information`と`layer_id_information`とで共通でもよいし、個別に持つようにしてもよい。前者であれば、モデル情報ヘッダの情報量を削減でき、後者であれば、使用する際の前提条件に応じて独立に設定できる。

さらに、モデル情報ヘッダは、上記テキスト情報のフォーマットを示す情

報を含む。例えば、NNEFはインデックス0、その他のフォーマットが1以降となるような情報である。これによって、どのフォーマットで記述されているかを識別することができ、正しく復号することができる。

[0109] なお、図18および図19に示したようなテキスト情報で表されたレイヤ構造情報と、レイヤ構造情報に対応するレイヤ識別番号を示す情報は、実施の形態1で示した全てのシステムに適用することが可能である。さらに、`model__structure__information`と`layer__id__information`から、当該符号化データのみから各レイヤデータユニットがモデル内のどのレイヤのデータであるのかを識別することができる。したがって、モデルを更新する場合（`reference__model__present__flag`が有効である場合）、本実施の形態で示す符号化データから生成されていないモデルを参照モデルとすることも可能である。すなわち、本実施の形態で示す符号化データは、モデル情報ヘッダの一部として`model__structure__information`と`layer__id__information`を持つことで、任意のモデルを参照モデルとして設定することができる。ただし、この場合、参照モデル識別番号（`reference__model__id`）と参照モデルとの対応付けは別途定義しておく必要がある。

[0110] 次に、実施の形態1に係るデータ処理装置の機能を実現するハードウェア構成について説明する。実施の形態1に係るデータ処理装置における、データ処理部10および符号化部11の機能は、処理回路により実現される。すなわち、実施の形態1に係るデータ処理装置は、図5のステップST1からステップST6までの処理を実行するための処理回路を備える。処理回路は、専用のハードウェアであってもよいが、メモリに記憶されたプログラムを実行するCPU（Central Processing Unit）であってもよい。

[0111] 図20Aは、実施の形態1に係るデータ処理装置の機能を実現するハードウェア構成を示すブロック図である。図20Aにおいて、処理回路300は

、図3に示したデータ処理装置として機能する専用の回路である。図20Bは、実施の形態1に係るデータ処理装置の機能を実現するソフトウェアを実行するハードウェア構成を示すブロック図である。図20Bにおいて、プロセッサ301およびメモリ302は、信号バスによって互いに接続されている。

[0112] 上記処理回路が図20Aに示す専用のハードウェアである場合、処理回路300は、例えば、単回路、複合回路、プログラム化したプロセッサ、並列プログラム化したプロセッサ、ASIC (Application Specific Integrated Circuit)、FPGA (Field-Programmable Gate Array) またはこれらを組み合わせたものが該当する。なお、データ処理部10および符号化部11の機能を別々の処理回路で実現してもよいし、これらの機能をまとめて1つの処理回路で実現してもよい。

[0113] 上記処理回路が図20Bに示すプロセッサである場合、データ処理部10および符号化部11の機能は、ソフトウェア、ファームウェアまたはソフトウェアとファームウェアとの組み合わせによって実現される。ソフトウェアまたはファームウェアは、プログラムとして記述されて、メモリ302に記憶される。プロセッサ301は、メモリ302に記憶されたプログラムを読み出して実行することによって、データ処理部10および符号化部11の機能を実現する。すなわち、実施の形態1に係るデータ処理装置は、プロセッサ301によって実行されるときに、図5に示したステップST1からステップST6までの処理が結果的に実行されるプログラムを記憶するためのメモリ302を備える。これらのプログラムは、データ処理部10および符号化部11の手順または方法をコンピュータに実行させるものである。メモリ302は、コンピュータを、データ処理部10および符号化部11として機能させるためのプログラムが記憶されたコンピュータ可読記憶媒体であってもよい。

[0114] メモリ302には、例えば、RAM (Random Access Me

mory)、ROM (Read Only Memory)、フラッシュメモリ、EPROM (Erasable Programmable Read Only Memory)、EEPROM (Electrically EPROM)などの不揮発性または揮発性の半導体メモリ、磁気ディスク、フレキシブルディスク、光ディスク、コンパクトディスク、ミニディスク、DVDなどが該当する。

[0115] なお、データ処理部10および符号化部11の機能について一部を専用のハードウェアで実現し、一部をソフトウェアまたはファームウェアで実現してもよい。例えば、データ処理部10については、専用のハードウェアとしての処理回路でその機能を実現し、符号化部11については、プロセッサ301がメモリ302に記憶されたプログラムを読み出して実行することによってその機能を実現してもよい。このように、処理回路は、ハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせによって、上記機能のそれぞれを実現することができる。

[0116] なお、図3に示したデータ処理装置について説明したが、図4に示したデータ処理装置においても、同様である。例えば、図4に示したデータ処理装置は、図6のステップST11からステップST13までの処理を実行するための処理回路を備える。この処理回路は、専用のハードウェアであってもよいが、メモリに記憶されたプログラムを実行するCPUであってもよい。

[0117] 上記処理回路が図20Aに示す専用のハードウェアである場合、処理回路300は、例えば、単回路、複合回路、プログラム化したプロセッサ、並列プログラム化したプロセッサ、ASIC、FPGAまたはこれらを組み合わせたものが該当する。なお、復号部201および推論部202の機能を、別々の処理回路で実現してもよいし、これらの機能をまとめて1つの処理回路で実現してもよい。

[0118] 上記処理回路が図20Bに示すプロセッサであると、復号部201および推論部202の機能は、ソフトウェア、ファームウェアまたはソフトウェアとファームウェアとの組み合わせによって実現される。ソフトウェアまたは

ファームウェアは、プログラムとして記述されて、メモリ302に記憶される。プロセッサ301は、メモリ302に記憶されたプログラムを読み出して実行することによって、復号部201および推論部202の機能を実現する。すなわち、図4に示したデータ処理装置は、プロセッサ301によって実行されるときに、図6に示すステップST11からステップST13までの処理が結果的に実行されるプログラムを記憶するためのメモリ302を備える。これらのプログラムは、復号部201および推論部202の手順または方法を、コンピュータに実行させるものである。メモリ302は、コンピュータを、復号部201および推論部202として機能させるためのプログラムが記憶されたコンピュータ可読記憶媒体であってもよい。

[0119] なお、復号部201および推論部202の機能について一部を専用のハードウェアで実現し、一部をソフトウェアまたはファームウェアで実現してもよい。例えば、復号部201については専用のハードウェアとしての処理回路でその機能を実現し、推論部202については、プロセッサ301がメモリ302に記憶されたプログラムを読み出して実行することによってその機能を実現してもよい。

[0120] 以上のように、実施の形態1に係るデータ処理装置において、符号化部11が、レイヤ構造情報を符号化し、レイヤ更新フラグを符号化し、レイヤ更新フラグがレイヤ構造の更新を示す場合は、新規レイヤフラグを符号化する。NNの構造を示すデータのうち、更新されたレイヤに関する情報のみが符号化されて伝送されるので、NNの構造を示すデータのデータサイズを削減することができる。

[0121] また、符号化部11が、NNの構成を示す情報を符号化し、ヘッダ情報とレイヤ単位の符号化データから構成された符号化データを生成する。復号側で必要なレイヤに関する情報のみを符号化することができるので、NNの構成に関する情報を符号化する処理負荷が低減され、復号側へ伝送するデータサイズを削減することができる。

[0122] 実施の形態1に係るデータ処理装置において、符号化部11が、NNのレ

イヤに属するエッジの重み情報を、上位ビットからビットプレーン単位で符号化する。これにより、復号側へ伝送する符号化データのデータサイズを削減することができる。

[0123] 実施の形態 1 に係るデータ処理装置において、符号化部 11 が、ヘッダ情報で指定された 1 以上のレイヤに関する情報を符号化する。これにより、復号側で必要なレイヤに関する情報のみが符号化され、復号側へ伝送する符号化データのデータサイズを削減することができる。

[0124] 実施の形態 1 に係るデータ処理装置において、符号化部 11 が、ヘッダ情報で指定されたレイヤに属するエッジの重みの値と特定の値との差分を符号化する。これにより、復号側へ伝送する符号化データのデータサイズを削減することができる。

[0125] 実施の形態 1 に係るデータ処理装置において、符号化部 11 が、エッジの重み情報を、ベース符号化データとエンハンスメント符号化データに分けて符号化する。これにより、データ伝送ネットワーク 2 の伝送帯域と伝送許容時間に応じた符号化データの伝送を実現することができる。

[0126] なお、本発明は上記実施の形態に限定されるものではなく、本発明の範囲内において、実施の形態のそれぞれの自由な組み合わせまたは実施の形態のそれぞれの任意の構成要素の変形もしくは実施の形態のそれぞれにおいて任意の構成要素の省略が可能である。

産業上の利用可能性

[0127] 本発明に係るデータ処理装置は、例えば、画像認識技術への利用が可能である。

符号の説明

[0128] 1 サーバ、2 データ伝送ネットワーク、3-1~3-N クライアント、10、10A データ処理部、10-1~10-9、11-1~11-3 ノード、11 符号化部、12 復号部、12-1~12-15 エッジ、20 カーネル、101、101A 学習部、102 評価部、103 制御部、201 復号部、202 推論部、300 処理回路、301

プロセッサ、302 メモリ。

請求の範囲

- [請求項1] ニューラルネットワークを学習するデータ処理部と、
前記ニューラルネットワークのモデルを識別するためのモデルヘッダ情報、前記ニューラルネットワークのレイヤを識別するためのレイヤヘッダ情報およびレイヤ単位のエッジの重み情報が符号化された符号化データを生成する符号化部と、
を備え、
前記符号化部は、前記ニューラルネットワークのレイヤ構造を示すレイヤ構造情報と、符号化される各レイヤが参照モデルのレイヤからの更新であるか新規レイヤであるかを示す新規レイヤフラグを符号化すること
を特徴とするデータ処理装置。
- [請求項2] 前記符号化部は、レイヤに属するエッジの重み情報を、上位ビットからビットプレーン単位で符号化すること
を特徴とする請求項1記載のデータ処理装置。
- [請求項3] 前記符号化部は、前記レイヤヘッダ情報によって識別される、1以上のレイヤに属するエッジの重み情報を符号化すること
を特徴とする請求項1または請求項2記載のデータ処理装置。
- [請求項4] 前記符号化部は、エッジの重みの値と特定の値との差分を符号化すること
を特徴とする請求項1または請求項2記載のデータ処理装置。
- [請求項5] 前記符号化部は、エッジの重み情報を、ベース符号化データとエンハンスメント符号化データとに分けて符号化すること
を特徴とする請求項1または請求項2記載のデータ処理装置。
- [請求項6] 前記符号化部によって生成された符号化データを復号する復号部を備え、
前記データ処理部は、前記復号部によって復号された情報を用いて、前記ニューラルネットワークを学習すること

を特徴とする請求項 1 または請求項 2 記載のデータ処理装置。

[請求項7]

ニューラルネットワークを学習する第 1 のデータ処理部と、

前記ニューラルネットワークのモデルを識別するためのモデルヘッダ情報、前記ニューラルネットワークのレイヤを識別するためのレイヤヘッダ情報およびレイヤ単位のエッジの重み情報が符号化された符号化データを生成する符号化部と、

を有する第 1 のデータ処理装置と、

前記符号化部によって生成された符号化データから復号する復号部と、

前記復号部によって復号された情報を用いて、前記ニューラルネットワークを生成し、前記ニューラルネットワークを用いたデータ処理を行う第 2 のデータ処理部と、

を有する第 2 のデータ処理装置と、

を備え、

前記符号化部は、前記ニューラルネットワークのレイヤ構造を示すレイヤ構造情報と、符号化される各レイヤが参照モデルのレイヤからの更新であるか新規レイヤであるかを示す新規レイヤフラグを符号化すること

を特徴とするデータ処理システム。

[請求項8]

前記符号化部は、前記ニューラルネットワークの中間レイヤまでに
関する情報を符号化し、

前記第 2 のデータ処理装置は、前記ニューラルネットワークの中間レイヤから出力されたデータを特徴量として用いたデータ処理を行うこと

を特徴とする請求項 7 記載のデータ処理システム。

[請求項9]

データ処理部が、ニューラルネットワークを学習するステップと、

符号化部が、前記ニューラルネットワークのモデルを識別するためのモデルヘッダ情報、前記ニューラルネットワークのレイヤを識別す

るためのレイヤヘッダ情報およびレイヤ単位のエッジの重み情報が符号化された符号化データを生成するステップと、

を備え、

前記符号化部は、前記ニューラルネットワークのレイヤ構造を示すレイヤ構造情報と、符号化される各レイヤが参照モデルのレイヤからの更新であるか新規レイヤであるかを示す新規レイヤフラグを符号化すること

を特徴とするデータ処理方法。