

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、多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、多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場合、NNのモデルとパラメータのデータサイズおよび適切な性能を示す値は、個々のクライアントの処理性能、記憶容量およびサーバとの間のデータ伝送ネットワークの伝送帯域によって異なる。このため、NNのモデルおよびパラメータを、非特許文献1に記載された技術を用いて符号化しても、多数のクライアントのそれぞれに適したデータサイズに圧縮する必要がある、符号化の処理負荷が高くなるという課題があった。また、クライアントの数が膨大であると、個々のクライアントへ伝送されるデータが圧縮されても、全てのクライアントへ伝送するデータサイズは膨大になるという課題もある。

[0007] 本発明は上記課題を解決するものであって、NNに関する情報を符号化する処理負荷が低減され、伝送される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モデルヘッダ情報と、NNのレイヤを識別するレイヤヘッダ情報と、レイヤ単位のエッジの重み情報と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る。

発明の効果

[0009] 本発明によれば、符号化部が、NNのモデルを識別するモデルヘッダ情報

と、NNのレイヤを識別するレイヤヘッダ情報と、レイヤ単位のエッジの重み情報とが符号化された符号化データを生成する。符号化部は、NNにおいて復号側で必要なレイヤに関する情報を符号化することができるので、NNに関する情報を符号化する処理負荷が低減され、伝送される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]図15A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15B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16]実施の形態2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。

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, $\dots$, 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, $\dots$, 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モデルヘッダ情報と、NNのレイヤを識別するレイヤヘッダ情報と、レイヤ単位のエッジの重み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, $\dots$, 3-Nに送信する。クライアント3-1, 3-2, $\dots$, 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、エッジの重み情報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)

er)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。

[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に予め設定されており、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と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、畳み込みを行うフィルタ（カーネル）のデータサイズと形状（フィルタ係数の数を含む）、当該フィルタの数、畳み込み間隔（ストライド）、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）、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、後述するモデルインデックス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インデックス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S T 3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インデックスを持つ場合は、当該参照モデルインデックスが指し示すモデルの学習履歴（履歴A）を参照する。これによって、当該モデルの学習時に参照モデルインデックス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インデックスと参照モデルインデックスが同一である場合、当該モデルの学習を実施する度に参照モデルインデックス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インデックスと参照モデルインデックスが異なる場合は、参照モデルインデックスに対応する学習履歴（履歴A）を当該モデルのモデルインデックスの学習履歴（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S T 3；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S T 4）。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い、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S T 3；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S T 5）。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した場合、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い場合、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い場合、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の処理に戻る。

[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。

[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符号化には、例えば（1）または（2）

の符号化方法を用いることができる。

(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、後述するレイヤインデックスによって、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] 0x000001が発生しないように設定するためには、例えば、0x000000~0x000003の符号化データにおける3バイト目に03を挿入して0x000300~0x000303とし、復号するときに、0x000003を0x0000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、0x000001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、モデルインデックス、モデル内レイヤデータユニット数および符号化レイヤデータユニット数が含まれる。モデルインデックスは、NNのモデルを識別するためのインデックスである。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インデックスを持つが、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インデックスを持つモデルを新たに受信した場合、当該モデルインデックス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、モデルインデックス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、レイヤインデックスおよびレイヤ構造情報が含まれる。レイヤインデックスは、レイヤを識別するためのインデックスである。レイヤインデックスによってどの層のレイヤか識別できるように、レイヤインデックスの値の振り方は予め固定的に定義される。例えば、NNの入力層をインデックス0、次の層をインデックス1というように、入力層に近い層から順にインデックスを振る、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り、前述したように、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。

[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と本レイヤ

構造情報に対応するレイヤインデックスを含んでもよい。復号側は、モデル情報ヘッダを参照することで、各レイヤインデックス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、各レイヤインデックス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、レイヤインデックス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)、 $\dots$ 、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)、 $\dots$ 、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、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樣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と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] また、符号化部11は、モデルインデックスに加え、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インデックスをモデル情報ヘッダに含めてもよい。(B)の例の場合、上記参照モデルインデックス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を、モデル情報ヘッダに設定してもよい。符号化部11は、この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、モデル情報ヘッダに参照モデルインデックスを設定する。

[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、参照モデルインデックスを参照することで、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。参照モデルインデックス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は、モデル内レイヤデータユニット数よりも少ない。復号側では、モデル情報ヘッダに含まれる参照モデルインデックスと、レイヤ情報ヘッダに含まれるレイヤインデックス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に繋がりが、ノード10-2は、エッジ12-2を介してノード11-1に繋がりが、ノード10-3は、エッジ12-3を介してノード11-1に繋がりが、ノード10-4は、エッジ12-4を介してノード11-1に繋がりが、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に繋がりが、ノード10-4は、エッジ12-7を介してノード11-2に繋がりが、ノード10-5は、エッジ12-8を介してノード11-2に繋がりが、ノード10-6は、エッジ12-9を介してノード11-2に繋がりが、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。ここで、 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、 $j = 1, 2, \dots, N_{l-1}$ 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る。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、 $j' = 1, 2, \dots, K_l$ 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] 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。
- [0080] 図１５Ａ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１５Ａ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１５Ｂ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１５Ｂ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。
- [0081] 上記処理回路が図１５Ａ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。
- [0082] 上記処理回路が図１５Ｂ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。

[0083] メモリ302には、例えば、RAM (Random Access Memory)、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。

[0084] 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。

[0085] 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。

[0086] 上記処理回路が図15A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。

[0087] 上記処理回路が図15B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。

[0088] 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。

[0089] 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、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。

[0090] 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。

[0091] 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。

[0092] 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。

[0093] 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と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。

[0094] 実施の形態2.

図16は、実施の形態2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16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とを用いて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16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。

[0095] 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。

[0096] 学習部 101A は、実施の形態 1 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。

[0097] 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。

[0098] 以上のように、実施の形態 2 に係るデータ処理装置は、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、実施の形態 2 に係るデータ処理装置は、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。

[0099] 実施の形態 3.

実施の形態 3 に係るデータ処理システムは、図 1 と同様の構成を有しており、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る。実施の形態 3 に係る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たデータは、下記の参考文献に記載された画像検索 (r e t r i e v

a l) 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られる。

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

[0100] 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。

[0101] 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。

[0102] 以上のように、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。これにより、量子化によるパラメータデータの圧縮率が高まるので、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。

[0103] 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

[0104] 本発明に係るデータ処理装置は、ニューラルネットワーク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ので、例えば、画像認識技術への利用が可能である。

符号の説明

[0105] 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。