

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2019年8月22日(22.08.2019)



(10) 国際公開番号

WO 2019/159784 A1

- (51) 国際特許分類:
G06N 3/08 (2006.01) G06N 3/04 (2006.01)
- (21) 国際出願番号: PCT/JP2019/004214
- (22) 国際出願日: 2019年2月6日(06.02.2019)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2018-025942 2018年2月16日(16.02.2018) JP
- (71) 出願人: 日本電信電話株式会社 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目5番1号 Tokyo (JP).
- (72) 発明者: 川合 健治(KAWAI, Kenji); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). 加藤 順一(KATO,

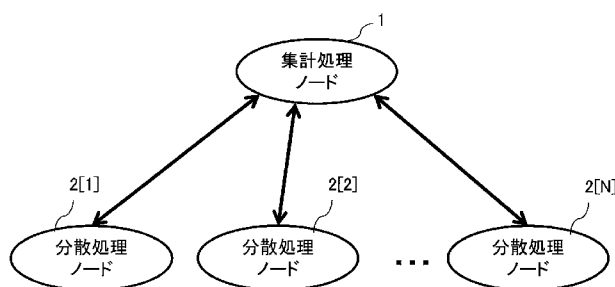
Junichi); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). ゴーフィクー(NGO, Huycu); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). 有川 勇輝(ARIKAWA, Yuki); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). 伊藤 猛(ITO, Tsuyoshi); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). 坂本 健(SAKAMOTO, Takeshi); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP).

(74) 代理人: 山川 茂樹, 外(YAMAKAWA, Shigeki et al.); 〒1006104 東京都千代田区永田町2丁目11番1号 山王パークタワー4階 山川国際特許事務所内 Tokyo (JP).

(54) Title: DISTRIBUTED PROCESSING SYSTEM AND DISTRIBUTED PROCESSING METHOD

(54) 発明の名称: 分散処理システムおよび分散処理方法

[図1]



1 Aggregation processing node
2[1], 2[2], 2[n] Distributed processing node

(57) Abstract: In the present invention, learning efficiency of a neural network is improved. Distributed processing nodes (2)[n] ($n=1, \dots, N$) packetize, in an m -numbered sequence, respective distributed data elements $D[m, n]$ for corresponding M weights $w[m]$ ($m=1, \dots, M$) of a learning-target neural network, transmit the packetized distributed data elements $D[m, n]$ to an aggregation processing node (1), receive packets transmitted from the aggregation processing node (1), acquire aggregate data elements $R[m]$ in an m -numbered sequence, and update the weights $w[m]$ of the neural network on the basis of the respective aggregate data elements $R[m]$. The aggregation processing node (1) receives the packets transmitted from the distributed processing nodes (2)[n], acquires the distributed data elements $D[m, n]$ in an m -numbered sequence, generates the aggregate data elements $R[m]$ respectively obtained by aggregating, for the corresponding weights $w[m]$, the distributed data elements $D[m, n]$ of all the distributed processing nodes, packetizes the aggregate data elements $R[m]$ in an m -numbered sequence, and transmits the packetized aggregate data elements $R[m]$ to the distributed processing nodes (2)[n].

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告(条約第21条(3))

(57) 要約: ニューラルネットワークの学習効率を向上させる。各分散処理ノード(2) [n] ($n = 1, \dots, N$) は、学習対象のニューラルネットワークのM個の重みw [m] 毎 ($m = 1, \dots, M$) の分散データD [m, n] を番号mの順番にパケット化して集計処理ノード(1)に送信し、集計処理ノード(1)から送信されたパケットを受信して集計データR [m] を番号mの順番に取得して、集計データR [m] に基づいてニューラルネットワークの重みw [m] を更新する。集計処理ノード(1)は、各分散処理ノード(2) [n] から送信されたパケットを受信して分散データD [m, n] を番号mの順番に取得し、全分散処理ノードの分散データD [m, n] を重みw [m] 毎に集計した集計データR [m] を生成して、集計データR [m] を番号mの順番にパケット化して各分散処理ノード(2) [n] に送信する。

明 細 書

発明の名称：分散処理システムおよび分散処理方法

技術分野

[0001] 本発明は、集計処理ノードと複数の分散処理ノードとを連携させてニューラルネットワークの学習を行う分散処理システムおよび分散処理方法に関するものである。

背景技術

[0002] 深層学習では、多層のニューロンモデルからなる学習対象について、各ニューロンモデルの重み（前段のニューロンモデルが出力した値に乘じる係数）を、入力したサンプルデータに基づいて更新することにより、推論精度を改善する。

[0003] 通常、推論精度を改善する手法には、ミニバッチ法が用いられている。ミニバッチ法では、サンプルデータ毎に前記重みに対する勾配を計算する勾配計算処理と、複数の異なるサンプルデータについて前記勾配を集計する（サンプルデータ毎に得られた勾配を重み別に合算する）集計処理と、各重みを前記集計された勾配に基づいて更新する重み更新処理と、を繰り返す。

[0004] これらの処理、特に勾配計算処理は、多数回の演算を必要とするが、推論精度を向上させるために、重みの個数や入力するサンプルデータの個数が増加すると、深層学習に要する時間が増大するという、課題がある。

[0005] 勾配計算処理を高速化するため、分散処理の手法が用いられている。具体的には、複数の分散処理ノードを設け、各ノードは、各々異なるサンプルデータについて勾配計算処理を行う。これにより、ノード数に比例して単位時間に処理できるサンプルデータ数を増加させることが可能となるため、勾配計算処理を高速化できる（非特許文献1参照）。

[0006] 深層学習の分散処理において、集計処理を行うためには、分散処理ノード毎に得られたデータ（分散データ）を集計処理ノードに集約するための、各分散処理ノードから集計処理ノードへの通信（集約通信）と、集計処理ノード

ドでの全ノード集計処理と、集計処理ノードが集計したデータ（集計データ）を各分散処理ノードに転送するための、集計処理ノードから分散処理ノードへの通信（分配通信）と、が必要となる。

[0007] 図12に、従来技術による、深層学習の分散処理のシーケンスを示す。分散処理ノード100[n] ($n=1, \dots, N$) は、Iの期間においてサンプルデータ入力と勾配計算処理とノード内集計処理とを行い、分散データを集計処理ノード101に送信する。IIの期間では、このような各ノードからの送信が行われるが、各ノードが同時に分散データを送信するとは限らない。

[0008] IIIの期間では、集計処理ノード101が各ノードから得られた勾配を重み別に合算する全ノード集計処理を行い、IVの期間において集計データを各分散処理ノード100[n] に送信する。Vの期間では、各分散処理ノード100[n] が重み更新処理を行う。

こうして、分散処理が行われることにより、深層学習には、集約通信（II）と全ノード集計処理（III）と分配通信（IV）の各処理時間が加わる。

[0009] このような処理時間は、深層学習を単一ノードで実施するシステムでは不要であり、深層学習の分散処理を行う上で、処理速度を低下させる要因となっている。

近年、深層学習がより複雑な問題に適用されるようになってきており、重みの総数が増加する傾向にある。このため、分散データや集計データのデータ量が増大し、集約通信時間と分配通信時間が増大している。

[0010] このように、深層学習の分散システムは、集約通信時間と分配通信時間の増大によって、各分散処理ノード数を増加させることにより、深層学習の高速化の効果が低下するという問題があった。図13は、従来の分散処理システムにおける分散処理ノード数と深層学習の処理性能との関係を示しており、200は分散処理ノード数と処理性能の理想的な関係（性能 $\propto$ ノード数）を示し、201は分散処理ノード数と処理性能の実際の関係を示している。

先行技術文献

非特許文献

- [0011] 非特許文献1：秋葉 拓哉，“分散深層学習パッケージ ChainerMN 公開”，プリファードインフラストラクチャー (Preferred Infrastructure)，2017年，インターネット<<https://research.preferred.jp/2017/05/chainermn-beta-release/>>

発明の概要

発明が解決しようとする課題

- [0012] 本発明は、上記のような事情を考慮してなされたものであり、その目的は、集計処理ノードと複数の分散処理ノードとを備える分散処理システムにおいて、ニューラルネットワークの学習効率を向上させることができる分散処理システムおよび分散処理方法を提供することにある。

課題を解決するための手段

- [0013] 本発明の分散処理システムは、集計処理ノードとN個（Nは2以上の整数）の分散処理ノードとを備え、各分散処理ノードは、学習対象のニューラルネットワークのM個（Mは2以上の整数）の重み $w[m]$ 毎（ $m=1, \dots, M$ ）の分散データ $D[m, n]$ （ $n=1, \dots, N$ ）を重み $w[m]$ の番号mの順番にパケット化して前記集計処理ノードに送信し、前記集計処理ノードから送信されたパケットを受信して集計データ $R[m]$ を番号mの順番に取得して、この集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新し、前記集計処理ノードは、各分散処理ノードから送信されたパケットを受信して分散データ $D[m, n]$ を番号mの順番に取得し、全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した前記集計データ $R[m]$ を生成して、この集計データ $R[m]$ を番号mの順番にパケット化して各分散処理ノードに送信することを特徴とするものである。
- [0014] また、本発明の分散処理システムの1構成例において、各分散処理ノードは、前記分散データ $D[m, n]$ を番号mの順番にパケット化して前記集計

処理ノードに送信する送信部と、前記集計処理ノードから送信されたパケットを受信して前記集計データ $R[m]$ を番号 m の順番に取得する受信部と、前記集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新する重み更新処理部とを備えることを特徴とするものである。

また、本発明の分散処理システムの 1 構成例において、前記集計処理ノードは、各分散処理ノードから送信されたパケットを受信して前記分散データ $D[m, n]$ を番号 m の順番に取得する受信部と、全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した前記集計データ $R[m]$ を生成する集計処理部と、前記集計データ $R[m]$ を番号 m の順番にパケット化して各分散処理ノードに送信する送信部とを備えることを特徴とするものである。

[0015] また、本発明の分散処理システムの 1 構成例において、各分散処理ノードは、前記ニューラルネットワークの学習用のサンプルデータが入力されたときに、前記ニューラルネットワークの重み $w[m]$ の各々について、前記ニューラルネットワークの損失関数の勾配をサンプルデータ毎に計算する勾配計算処理部と、サンプルデータ毎の前記勾配を集計した数値である前記分散データ $D[m, n]$ を重み $w[m]$ 毎に生成して保持するノード内集計処理部とを、さらに備えることを特徴とするものである。

また、本発明の分散処理システムの 1 構成例において、前記集計処理ノードと各分散処理ノードとは、各分散処理ノードがパケット化した分散データ $D[m, n]$ を前記集計処理ノードに送信して、前記集計処理ノードが受信したパケットから前記分散データ $D[m, n]$ を取得する集約通信処理と、前記集計処理ノードが前記集計データ $R[m]$ を生成する全ノード集計処理と、前記集計処理ノードがパケット化した前記集計データ $R[m]$ を各分散処理ノードに送信して、各分散処理ノードが受信したパケットから前記集計データ $R[m]$ を取得する分配通信処理と、各分散処理ノードが前記重み $w[m]$ を更新する重み更新処理とを、それぞれ異なる番号 m について並行して行うことを特徴とするものである。

[0016] また、本発明の分散処理方法は、 N 個（ N は2以上の整数）の分散処理ノードの各々が、学習対象のニューラルネットワークの M 個（ M は2以上の整数）の重み $w[m]$ 毎（ $m=1, \dots, M$ ）の分散データ $D[m, n]$ （ $n=1, \dots, N$ ）を重み $w[m]$ の番号 m の順番にパケット化して集計処理ノードに送信する第1のステップと、前記集計処理ノードが、各分散処理ノードから送信されたパケットを受信して前記分散データ $D[m, n]$ を番号 m の順番に取得する第2のステップと、前記集計処理ノードが、全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した集計データ $R[m]$ を生成する第3のステップと、前記集計処理ノードが、前記集計データ $R[m]$ を番号 m の順番にパケット化して各分散処理ノードに送信する第4のステップと、前記分散処理ノードの各々が、前記集計処理ノードから送信されたパケットを受信して前記集計データ $R[m]$ を番号 m の順番に取得する第5のステップと、前記分散処理ノードの各々が、前記集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新する第6のステップとを含むことを特徴とするものである。

[0017] また、本発明の分散処理方法の1構成例は、前記第1のステップの前に、前記分散処理ノードの各々が、前記ニューラルネットワークの学習用のサンプルデータが入力されたときに、前記ニューラルネットワークの重み $w[m]$ の各々について、前記ニューラルネットワークの損失関数の勾配をサンプルデータ毎に計算する第7のステップと、前記分散処理ノードの各々が、サンプルデータ毎の前記勾配を集計した数値である前記分散データ $D[m, n]$ を重み $w[m]$ 毎に生成して保持する第8のステップとを、さらに含むことを特徴とするものである。

また、本発明の分散処理方法の1構成例は、前記分散処理ノードの前記第1のステップおよび前記集計処理ノードの前記第2のステップと、前記集計処理ノードの前記第3のステップと、前記集計処理ノードの前記第4のステップおよび前記分散処理ノードの前記第5のステップと、前記分散処理ノードの前記第6のステップとを、それぞれ異なる番号 m について並行して行う

ことを特徴とするものである。

発明の効果

[0018] 本発明によれば、各分散処理ノードが、ニューラルネットワークの重み毎の分散データを順番にパケット化して集計処理ノードに送信し、集計処理ノードから送信されたパケットに格納された集計データを順番に取得してニューラルネットワークの重みを更新し、集計処理ノードが、各分散処理ノードから送信されたパケットに格納された分散データを順番に取得し、全分散処理ノードの分散データを集計した集計データをパケット化して各分散処理ノードに送信することにより、各分散処理ノードから集計処理ノードに分散データを送信する処理と、集計処理ノードから各分散処理ノードに集計データを送信する処理とを同時に行うことができ、効果的な分散処理を行うことができ、ニューラルネットワークの学習効率を向上させることができる。

図面の簡単な説明

[0019] [図1]図1は、本発明の第1の実施例に係る深層学習用分散処理システムの構成例を示すブロック図である。

[図2]図2は、本発明の第1の実施例に係る深層学習用分散処理システムの分散処理ノードの構成例を示すブロック図である。

[図3]図3は、本発明の第1の実施例に係る分散処理ノードのサンプルデータ入力処理と勾配計算処理とノード内集計処理を説明するフローチャートである。

[図4]図4は、本発明の第1の実施例に係る分散処理ノードの集約通信処理を説明するフローチャートである。

[図5]図5は、本発明の第1の実施例に係る集計処理ノードの集約通信処理を説明するフローチャートである。

[図6]図6は、本発明の第1の実施例に係る集計処理ノードの全ノード集計処理を説明するフローチャートである。

[図7]図7は、本発明の第1の実施例に係る集計処理ノードの分配通信処理を説明するフローチャートである。

[図8]図8は、本発明の第1の実施例に係る分散処理ノードの分配通信処理を説明するフローチャートである。

[図9]図9は、本発明の第1の実施例に係る分散処理ノードの重み更新処理を説明するフローチャートである。

[図10]図10は、本発明の第1の実施例に係る集計処理ノードと分散処理ノードの処理のシーケンスを示す図である。

[図11]図11は、本発明の第2の実施例に係る集計処理ノードの構成例を示すブロック図である。

[図12]図12は、従来の深層学習の分散処理のシーケンスを示す図である。

[図13]図13は、従来の分散処理システムにおける分散処理ノード数と深層学習の処理性能との関係を示す図である。

発明を実施するための形態

[0020] [第1の実施例]

以下、本発明の実施例について図面を参照して説明する。図1は本発明の第1の実施例に係る深層学習用分散処理システムの構成例を示すブロック図である。図1の分散処理システムは、1個の集計処理ノード1と、ニューラルネットワークのサンプルデータ（学習データ）の集合毎に設けられたN個（Nは2以上の整数）の分散処理ノード2[n]（ $n=1, \dots, N$ ）とを備える。各分散処理ノード2[n]は、双方向の通信が可能なネットワーク3により、集計処理ノード1と接続されている。

[0021] なお、本発明において、「ノード」とは、ネットワーク上に分散配置されているサーバ等の機器を意味する。

[0022] 図2は分散処理ノード2[n]の構成例を示すブロック図である。各分散処理ノード2[n]は、それぞれ図示しないデータ収集ノードから学習用のサンプルデータを受け取るサンプル入力部20と、サンプルデータが入力されたときに、ニューラルネットワークの重みの各々について、ニューラルネットワークの損失関数の勾配をサンプルデータ毎に計算する勾配計算処理部21と、サンプルデータ毎の勾配を集計した数値である分散データを重み毎

に生成して保持するノード内集計処理部 22 と、分散データをパケット化して集計処理ノード 1 に送信する送信部 23 と、集計処理ノード 1 から送信されたパケットを受信して集計データを取得する受信部 24 と、集計データに基づいてニューラルネットワークの重みを更新する重み更新処理部 25 と、ソフトウェア的に構築された数学モデルであるニューラルネットワーク 26 とを備えている。

[0023] 図 3 は分散処理ノード 2 [n] のサンプルデータ入力処理と勾配計算処理とノード内集計処理を説明するフローチャートである。各分散処理ノード 2 [n] ($n = 1, \dots, N$) のサンプル入力部 20 は、図示しないデータ収集ノードから異なる S 個 (S は 2 以上の整数) のサンプルデータ $x[n, s]$ ($s = 1, \dots, S$) をミニバッチ毎に入力する (図 3 ステップ S100)。

[0024] なお、本発明は、データ収集ノードによるサンプルデータの収集方法、および収集したサンプルデータを N 個の集合に振り分けて各分散処理ノード 2 [n] へ分配する方法に限定されるものではなく、これらの方法の如何を問わず適用が可能である。

[0025] 各分散処理ノード 2 [n] ($n = 1, \dots, N$) の勾配計算処理部 21 は、サンプルデータ $x[n, s]$ が入力されたとき、学習対象のニューラルネットワーク 26 の M 個 (M は 2 以上の整数) の重み $w[m]$ ($m = 1, \dots, M$) の各々について、ニューラルネットワーク 26 の損失関数の勾配 $G[m, n, s]$ をサンプルデータ $x[n, s]$ 毎に計算する (図 3 ステップ S101)。

[0026] ニューラルネットワーク 26 を各分散処理ノード 2 [n] にソフトウェアで構築する方法、ニューラルネットワーク 26 の重み $w[m]$ 、ニューラルネットワーク 26 の性能の悪さを示す指標である損失関数、および損失関数の勾配 $G[m, n, s]$ については周知の技術であるので、詳細な説明は省略する。

[0027] 続いて、各分散処理ノード 2 [n] ($n = 1, \dots, N$) のノード内集

計処理部 22 は、サンプルデータ毎の勾配 $G[m, n, s]$ を集計した数値である分散データ $D[m, n]$ を、重み $w[m]$ 毎に生成して保持する（図 3 ステップ S102）。分散データ $D[m, n]$ の計算式は以下のとおりである。

[0028] [数1]

$$D[m, n] = \sum_{s=1, \dots, S} G[m, n, s] \quad \dots (1)$$

[0029] なお、勾配計算処理部 21 による勾配計算処理とノード内集計処理部 22 によるノード内集計処理とは、サンプルデータ単位でパイプライン化する（あるサンプルデータに対して勾配計算処理を行うと同時にその一つ前のサンプルデータから得た勾配を集計するノード内集計処理とを同時に実行する）ことができる。

[0030] 図 4 は分散処理ノード 2 [n] の集約通信処理を説明するフローチャートである。各分散処理ノード 2 [n] ($n=1, \dots, N$) の送信部 23 は、重み $w[m]$ 毎の分散データ $D[m, n]$ ($m=1, \dots, M$) を、重み $w[m]$ の番号 m の順番にパケット化して集計処理ノード 1 に送信する集約通信を行う。

[0031] このとき、各分散処理ノード 2 [n] ($n=1, \dots, N$) の送信部 23 は、保持する M 個の分散データ $D[m, n]$ ($m=1, \dots, M$) を、 L_g (L_g は 1 以上 M 未満の整数) 個ずつ P_g 個 (P_g は 2 以上の整数) の集約通信パケットに振り分けて（図 4 ステップ S103）、全ての集約通信パケットを送信し終わるまで（図 4 ステップ S105 において YES）、 P_g 個の集約通信パケットを順番に集計処理ノード 1 に送信する（図 4 ステップ S104）。すなわち、 p 番目 ($p=1, \dots, P_g$) に送信する集約通信パケット $SP[p, n]$ には、 L_g 個の分散データ $D[i, n]$ ($i = L_g \times (p-1) + 1, i=1, \dots, L_g$) が格納される。

[0032] なお、 M が L_g で割り切れない条件では、 P_g 個目の集約通信パケット $SP[P_g, n]$ には、 $(M - L_g \times (P_g - 1))$ 個の分散データ $D[i, n]$ ($i = L_g \times (P_g - 1) + q, q=1, \dots, M - L_g \times (P_g -$

1)) が格納される。

P_g 番目の集約通信パケット $SP [P_g, n]$ については、 $(M - L_g \times (P_g - 1))$ 個の分散データ $D [i, n]$ の後に、 $\{L_g - (M - L_g \times (P_g - 1))\}$ 個のダミーの数値を追加し、全ての集約通信パケットが等しく L_g 個のデータを格納するようにしてもよい。

[0033] 図5は集計処理ノード1の集約通信処理を説明するフローチャートである。集約通信において、集計処理ノード1は、各分散処理ノード2 $[n]$ が送信した各集約通信パケット $SP [p, n]$ ($p = 1, \dots, P_g$) を受信する(図5ステップS200)。

[0034] 集計処理ノード1は、受信した集約通信パケット $SP [p, n]$ から、分散処理ノード2 $[n]$ が保持していた L_g 個の分散データ $D [i, n]$ ($i = L_g \times (p - 1) + 1, i = 1, \dots, L_g$) を取得する(図5ステップS201)。

[0035] このように、集計処理ノード1は、各分散処理ノード2 $[n]$ ($n = 1, \dots, N$) が保持していた分散データ $D [m, n]$ ($m = 1, \dots, M$) を、重み $w [m]$ の番号 m の順番に取得することができる。

[0036] 図6は集計処理ノード1の全ノード集計処理を説明するフローチャートである。集計処理ノード1は、各分散処理ノード2 $[n]$ ($n = 1, \dots, N$) から重み $w [m]$ の分散データ $D [m, n]$ を取得し終えた後に(図6ステップS202においてYES)、取得した全分散処理ノード2 $[n]$ の分散データ $D [m, n]$ を重み $w [m]$ 毎に集計する全ノード集計処理を行い、集計データ $R [m]$ を生成する(図6ステップS203)。集計データ $R [m]$ の計算式は、以下のとおりである。

[0037] [数2]

$$R[m] = \sum_{n=1, \dots, N} D[m, n] \quad \dots (2)$$

[0038] このように、集計処理は、番号 m の順番に取得した分散データ $D [m, n]$ に基づいて、集計データ $R [m]$ を計算する処理である。このため、集計処理ノード1は、番号 m の順番に集計データ $R [m]$ を生成することができる。

る。

[0039] 図7は集計処理ノード1の分配通信処理を説明するフローチャートである。集計処理ノード1は、重み $w[m]$ 毎の集計データ $R[m]$ ($m=1, \dots, M$)を、重み $w[m]$ の番号 m の順番にパケット化して各分散処理ノード2 $[n]$ ($n=1, \dots, N$)に送信する分配通信を行う。

[0040] このとき、集計処理ノード1は、 M 個の集計データ $R[m]$ ($m=1, \dots, M$)を、 L_s 個 (L_s は1以上 M 未満の整数) ずつ P_s 個 (P_s は2以上の整数)の分配通信パケットに振り分けて(図7ステップS204)、全ての分配通信パケットを送信し終わるまで(図7ステップS206においてYES)、 P_s 個の分配通信パケットを順番に各分散処理ノード2 $[n]$ ($n=1, \dots, N$)に送信する(図7ステップS205)。すなわち、分散処理ノード2 $[n]$ に向けて p 番目 ($p=1, \dots, P_s$)に送信する分配通信パケット $DP[p, n]$ には、 L_s 個の集計データ $R[j]$ ($j=L_s \times (p-1) + k, k=1, \dots, L_s$)が格納される。

[0041] なお、 M が L_s で割り切れない条件では、 P_s 個目の分配通信パケット $DP[P_s, n]$ には、 $(M - L_s \times (P_s - 1))$ 個の集計データ $R[j]$ ($j=L_s \times (P_s - 1) + o, o=1, \dots, M - L_s \times (P_s - 1)$)が格納される。

P_s 番目の分配通信パケット $DP[P_s, n]$ については、 $(M - L_s \times (P_s - 1))$ 個の集計データ $R[j]$ の後に、 $\{L_s - (M - L_s \times (P_s - 1))\}$ 個のダミーの数値を追加し、全ての分配通信パケットが等しく L_s 個のデータを格納するようにしてもよい。

[0042] 図8は分散処理ノード2 $[n]$ の分配通信処理を説明するフローチャートである。分配通信において、各分散処理ノード2 $[n]$ ($n=1, \dots, N$)の受信部24は、集計処理ノード1が送信した各分配通信パケット $DP[p, n]$ ($p=1, \dots, P_s$)を順番に受信する(図8ステップS106)。

[0043] そして、各分散処理ノード2 $[n]$ ($n=1, \dots, N$)の受信部24

は、受信した分配通信パケットDP [p, n] から、集計処理ノード1が生成した L_s 個の集計データR [j] ($j = L_s \times (p - 1) + k$, $k = 1, \dots, L_s$)を取得する(図8ステップS107)。

[0044] このように、各分散処理ノード2 [n] ($n = 1, \dots, N$)は、集計処理ノード1が生成した集計データR [m] ($m = 1, \dots, M$)を、重みw [m]の番号mの順番に取得することができる。

[0045] なお、集計処理ノード1がp番目に送信する分配通信パケットDP [p, n]には、全ての分散処理ノード2 [n]に関して同じ集計データR [j] ($j = L_s \times (p - 1) + k$, $k = 1, \dots, L_s$)が格納されている。したがって、分配通信パケットDP [p, n]の宛先を指定する必要がない場合(例えば図1のように分散処理ノード別に経路が異なる場合や、全分散処理ノード宛にマルチキャストできるネットワークを介する場合)は、全ての分散処理ノード2 [n]に同じ分配通信パケットDP [p]を送信してもよい。

[0046] 図9は分散処理ノード2 [n]の重み更新処理を説明するフローチャートである。各分散処理ノード2 [n] ($n = 1, \dots, N$)の重み更新処理部25は、集計処理ノード1から重みw [m]の集計データR [m]を取得し終えた後に(図9ステップS108においてYES)、取得した集計データR [m]に基づいて、それぞれ自ノード内のニューラルネットワーク26の重みw [m]を更新する重み更新処理を行う(図9ステップS109)。

[0047] 重み更新処理においては、集計データR [m]が示す、損失関数の勾配に基づいて損失関数が最小になるように重みw [m]を番号m毎に更新すればよい。重みw [m]の更新は周知の技術であるので、詳細な説明は省略する。

[0048] このように、重み更新処理は、重みw [m]の番号mの順番に取得した集計データR [m]に基づいて、重みw [m]を更新する処理である。このため、各分散処理ノード2 [n] ($n = 1, \dots, N$)は、重みw [m]に対する重み更新処理を、番号mの順番に行うことができる。

[0049] 重み更新処理の終了により、1回のミニバッチ学習が終了し、各分散処理ノード2[n] ($n = 1, \dots, N$)と集計処理ノード1とは、更新された重みに基づき、次のミニバッチ学習の処理を継続して行う。すなわち、各分散処理ノード2[n]は、次のミニバッチ学習用のサンプルデータを図示しないデータ収集ノードから受け取り、上記で説明したミニバッチ学習の処理を繰り返すことにより、ニューラルネットワーク26の推論精度を向上させる。

[0050] なお、ミニバッチ学習の繰り返しの終了は、(A)ミニバッチ学習の回数が予め指定された値に到達した場合、(B)ニューラルネットワーク26の推論精度(例えば正答が既知の問題をニューラルネットワーク26に推論処理させたときの正答率)が予め指定された閾値を超過した場合、(C)ニューラルネットワーク26の推論精度の向上が停止した場合(予め指定されたミニバッチ学習の回数を繰り返したときに推論精度の上昇が予め指定された閾値を下回った場合)、(D)あるいは、(A)～(C)のうち少なくとも2つの場合の組み合わせが発生した場合、とする。このようなミニバッチ学習の繰り返しの終了は、各分散処理ノード2[n] ($n = 1, \dots, N$)が個別に判断してもよいし、集計処理ノード1が総合的に判断してもよい。

[0051] 図10に、集計処理ノード1と分散処理ノード2[n]の処理のシーケンスを示す。上記のように、各分散処理ノード2[n] ($n = 1, \dots, N$)は、M個の分散データD[m, n] ($m = 1, \dots, M$)を重みw[m]の番号mの順番にパケット化して集計処理ノード1に送信し、集計処理ノード1は、M個の分散データD[m, n] ($m = 1, \dots, M$)を番号mの順番に取得する、という集約通信処理を行う。

[0052] さらに、集計処理ノード1は、重みw[m]の番号mの順番に取得したM個の分散データD[m, n] ($m = 1, \dots, M$)に基づき、集計データR[m] ($m = 1, \dots, M$)を番号mの順番に生成する全ノード集計処理を行う。

[0053] さらに、集計処理ノード1は、重みw[m]の番号mの順番に生成したM

個の集計データ $R[m]$ ($m = 1, \dots, M$) を番号 m の順番にパケット化して各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) に送信し、各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) は、 M 個の集計データ $R[m]$ ($m = 1, \dots, M$) を番号 m の順番に取得する、という分配通信処理を行う。

[0054] さらに、各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) は、番号 m の順番に取得した M 個の集計データ $R[m]$ ($m = 1, \dots, M$) に基づき、 M 個の重み $w[m]$ を番号 m の順番に更新する重み更新処理を行う。

[0055] 本実施例では、集約通信処理と全ノード集計処理と分配通信処理と重み更新処理とを、並行してほぼ同時に（パイプラインで）行うことが可能であり、各通信や各処理が終了するまで、次の処理を開始できなかった従来技術によるシーケンス（図 12）と比較したとき、処理時間の大幅な短縮が可能となる。

[0056] すなわち、各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) の送信部 23 と集計処理ノード 1 とが M 個の重み $w[m]$ の内のある重み $w[m]$ の分散データ $D[m, n]$ について図 4、図 5 で説明した集約通信処理を行っているときに、集約通信処理中の重み $w[m]$ よりも番号 m の小さい重み $w[m]$ の取得済みの分散データ $D[m, n]$ について集計処理ノード 1 が図 6 で説明した全ノード集計処理を行い、全ノード集計処理中の重み $w[m]$ よりも番号 m の小さい重み $w[m]$ の集計処理済みの集計データ $R[m]$ について集計処理ノード 1 と各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) の受信部 24 とが図 7、図 8 で説明した分配通信処理を行い、分配通信処理中の重み $w[m]$ よりも番号 m の小さい重み $w[m]$ の取得済みの集計データ $R[m]$ に基づき各分散処理ノード 2 $[n]$ ($n = 1, \dots, N$) の重み更新処理部 25 が図 9 で説明した重み更新処理を行う。

[0057] したがって、例えば、集約通信処理と全ノード集計処理と分配通信処理と重み更新処理のそれぞれで時間 T を要する場合、従来技術では、これらの全ての処理を終えるのに $4T$ の時間を要したが、本実施例では $T + \alpha$ の時間で

済む。ここで、前記 α は、任意の分散処理ノード2 [n] が任意の分散データD [m, n] を集計処理ノード1に送信した時点から重みw [m] の更新が完了するまでの遅延時間である。本実施例では、重みw [m] の番号mの単位で処理をパイプライン化しているため、時間 α はTと比較すると十分に小さい時間である。したがって、本実施例では、従来技術と比較して、集約通信処理と全ノード集計処理と分配通信処理と重み更新処理とに要する時間を、約1 / 4に短縮することが可能である。

[0058] [第2の実施例]

次に、本発明の第2の実施例について説明する。本実施例は、第1の実施例の深層学習用分散処理システムの構成要素である集計処理ノード1の構成例を説明するものである。図11は集計処理ノード1の構成例を示すブロック図である。

[0059] 集計処理ノード1は、受信部10 [n] ($n=1, \dots, N$) と、受信FIFO (First In, First Out) バッファ11 [n] と、集計処理部12と、送信部13 [n] とを備えている。

[0060] 第1の実施例で説明したように、集計処理ノード1は、集約通信処理において、分散処理ノード2 [n] ($n=1, \dots, N$) の各々から、M個の分散データD [m, n] ($m=1, \dots, M$) を、 L_g 個ずつに振り分けられた P_g 個の集約通信パケットSP [p, n] ($p=1, \dots, P_g$) として受信する。集約通信パケットSP [p, n] ($p=1, \dots, P_g$) には、 L_g 個の分散データD [i, n] ($i=L_g \times (p-1) + 1, i=1, \dots, L_g$) が格納されている。

[0061] また、集計処理ノード1は、分配通信処理において、分散処理ノード2 [n] ($n=1, \dots, N$) の各々に向けて、M個の集計データR [m] ($m=1, \dots, M$) を、 L_s 個ずつに振り分けて P_s 個の集約通信パケットDP [p, n] ($p=1, \dots, P_s$) として送信する。

[0062] 図11に示したように、集計処理ノード1は、各分散処理ノード2 [n] ($n=1, \dots, N$) から集約通信パケットSP [p, n] を受信するた

めの受信部10[n]を、分散処理ノード2[n]別に備える。

[0063] 各受信部10[n]は、図5で説明した集約通信処理を行うものである。具体的には、各受信部10[n]は、対応する分散処理ノード2[n]が送信した集約通信パケットSP[p, n]を受信し、この集約通信パケットSP[p, n]に重みw[m]の番号mの順に格納されたLg個の分散データD[i, n] ($i = Lg \times (p - 1) + 1, 1 = 1, \dots, Lg$)を番号i (iは重みw[m]の番号mの一部)の順に取得して、後段の受信FIFOバッファ11[n]に渡す。

[0064] また、図11に示すように、集計処理ノード1は、受信部10[n]別(分散処理ノード2[n]別)に、受信FIFOバッファ11[n]を備える。さらに、集計処理ノード1は、各受信FIFOバッファ11[n] ($n = 1, \dots, N$)に蓄積されている、番号m ($m = 1, \dots, M$)の分散データD[m, n]を各受信FIFOバッファ11[n]から読み出して集計する集計処理部12を備える。受信FIFOバッファ11[n]と集計処理部12とは、図6で説明した全ノード集計処理を行うものである。

[0065] 具体的には、受信FIFOバッファ11[n]は、対応する受信部10[n]から渡されたLg個の分散データD[i, n] ($i = Lg \times (p - 1) + 1, 1 = 1, \dots, Lg$)を、番号i (iは番号mの一部)の順に蓄積する。この蓄積は各受信FIFOバッファ11[n]が空の状態から開始される。集約通信パケットSP[p, n]の受信と分散データD[i, n]の蓄積とがPg回行われることで、M個の分散データD[m, n]が各受信FIFOバッファ11[n]に蓄積される。

[0066] したがって、各受信FIFOバッファ11[n]に蓄積された分散データを同じ個数ずつ読み出した場合、各受信FIFOバッファ11[n]から読み出された分散データD[m, n]は $m = 1, \dots, M$ の順に並んでいる。

[0067] 各受信FIFOバッファ11[n] ($n = 1, \dots, N$)は、集計処理部12に対して、それぞれ分散データの蓄積の有無を示す蓄積有無信号U[

n] を出力する。

[0068] 集計処理部12は、全ての蓄積有無信号U[n] ($n=1, \dots, N$) が分散データの蓄積有を示す場合に、各受信FIFOバッファ11[n] から分散データを1個ずつ読み出す。なお、各受信FIFOバッファ11[n] は番号mの順番で分散データを蓄積しており、集計処理部12は各受信FIFOバッファ11[n] から同数の分散データを読み出す。このため、各受信FIFOバッファ11[n] から読み出された分散データの番号mは、各受信FIFOバッファ11[n] 間で同じ値となる。したがって、蓄積有無信号U[n] は分散データの番号mを特定する必要はなく、次に読み出すべき分散データが各受信FIFOバッファ11[n] に蓄積されているか否かを知らせるだけでよい。

[0069] ただし、後述するように、集計処理部12において、読み出し済みの分散データD[m, n] に基づいて生成した集計データR[m] を分配通信パケットに格納し、各送信部13[n] ($n=1, \dots, N$) から送出するが、集計処理部12は、分配通信パケットを送出する状態にない場合（例えば別の分配通信パケットを送出中である場合）は、次の分散データD[m, n] の読み出しを、分配通信パケットを送出可能となるまで保留する。

[0070] このため、各送信部13[n] ($n=1, \dots, N$) は、分配通信パケットが送出可能になったときに、分配通信パケットが送出可能であることを示す送出許可信号V[n] を集計処理部12に対して出力する。

[0071] 集計処理部12は、各受信FIFOバッファ11[n] ($n=1, \dots, N$) からの蓄積有無信号U[n] と、各送信部13[n] ($n=1, \dots, N$) からの送出許可信号V[n] ($n=1, \dots, N$) とを受けて、各受信FIFOバッファ11[n] から分散データの読み出しを行うか否かを判定する。

[0072] 具体的には、集計処理部12は、蓄積有無信号U[n] が、次に読み出すべき分散データD[m, n] の蓄積有りを示し、送出許可信号V[n] が、読み出し済みの分散データD[m, n] から生成された集計データR[m]

を含む分配通信パケットの送出可能を示しているとき、分散データ $D[m, n]$ を各受信 FIFO バッファ 11[n] から読み出す。

[0073] さらに、集計処理部 12 は、各受信 FIFO バッファ 11[n] から番号 m の順に読み出した分散データ $D[m, n]$ ($n = 1, \dots, N$) に基づいて、集計データ $R[m]$ を番号 m の順に生成し、集計データ $R[m]$ を番号 m の順に後段の送信部 13[n] に渡す。ここでは、各送信部 13[n] に同一の集計データが渡される。集計データ $R[m]$ の計算式は、式 (2) に示したとおりである。

[0074] 各分散処理ノード 2[n] ($n = 1, \dots, N$) に分配通信パケットを送信するための送信部 13[n] は、分散処理ノード 2[n] 別に設けられている。送信部 13[n] は、図 7 で説明した分配通信処理を行うものである。

[0075] 各送信部 13[n] は、集計処理部 12 から番号 m の順番に渡された集計データ $R[m]$ ($m = 1, \dots, M$) を、 L_s 個ずつ P_s 個の分配通信パケットに振り分けて送信する。すなわち、分散処理ノード 2[n] に向けて p 番目 ($p = 1, \dots, P_s$) に送信する分配通信パケット $DP[p, n]$ には、 L_s 個の集計データ $R[j]$ ($j = L_s \times (p - 1) + k, k = 1, \dots, L_s$) が格納される。上記のとおり、各送信部 13[n] は、分配通信パケット $DP[p, n]$ が送出可能になったときに、集計処理部 12 に対して送出許可信号 $V[n]$ を出力する。

[0076] 第 1 の実施例で説明したように、各送信部 13[n] は、 M が L_s で割り切れない条件では、 P_s 個目の分配通信パケット $DP[P_s, n]$ に、 $(M - L_s \times (P_s - 1))$ 個の集計データ $R[j]$ ($j = L_s \times (P_s - 1) + o, o = 1, \dots, M - L_s \times (P_s - 1)$) を格納する。また、各送信部 13[n] は、 P_s 番目の分配通信パケット $DP[P_s, n]$ について、 $(M - L_s \times (P_s - 1))$ 個の集計データ $R[j]$ の後に、 $\{L_s - (M - L_s \times (P_s - 1))\}$ 個のダミーの数値を追加し、全ての分配通信パケットが等しく L_s 個のデータを格納するようにしてもよい。

[0077] 以上のように、各受信部10[n] ($n=1, \dots, N$) は、分散処理ノード2[n] から受信した集約通信パケットから、重み $w[m]$ の番号 m ($m=1, \dots, M$)の順に分散データ $D[m, n]$ を取り出して、番号 m の順に、分散処理ノード別の受信FIFOバッファ11[n]に格納する。

[0078] 集計処理部12は、分散データ $D[m, n]$ を番号 m の順に各受信FIFOバッファ11[n]から読み出して、読み出した分散データ $D[m, n]$ に基づいて集計データ $R[m]$ を生成する。さらに、各送信部13[n]は、生成された集計データ $R[m]$ を番号 m の順に分配通信パケットに格納して、各分散処理ノード2[n]に送出する。

[0079] 図12で説明した従来技術では、集計処理ノード101が全ての分散データ $D[m, n]$ ($m=1, \dots, M$)を分散処理ノード100[n]から受け、その後、分散データ $D[m, n]$ を集計して全ての集計データ $R[m]$ ($m=1, \dots, M$)を生成し、その後、集計データ $R[m]$ を分散処理ノード100[n]に返していた。

[0080] これに対して、本実施例では、集計処理ノード1における集約通信処理と全ノード集計処理と分配通信処理とを互いに異なる番号 m についてパイプライン化することができるので、各分散処理ノード2[n]から分散データ $D[m, n]$ を受けてから、全ノードについて分散データ $D[m, n]$ を集計した集計データ $R[m]$ を、各分散処理ノード2[n]に返すまでの時間を、従来技術と比較して大幅に短縮することができる。

[0081] 例えば、番号 m に関わる各処理に要する時間を t とすると、各分散処理ノード2[n]から分散データ $D[m, n]$ を受けてから、全分散処理ノード2[n]について分散データ $D[m, n]$ を集計した集計データ $R[m]$ を、各分散処理ノード2[n]に返すまでの時間は、本発明では $4t$ （パイプライン段数=4）となる。

[0082] 一方、従来技術では、各処理に M 倍の時間を要するので、各分散処理ノード100[n]から分散データ $D[m, n]$ を受けてから、集計データ $R[m]$

m] を各分散処理ノード 100 [n] に返すまでの時間が $4t \times M$ となる。
このように、本実施例では、 $1/M$ (Mは重み $w[m]$ の個数であり 100, 000, 000 程度の値になり得る) に時間を短縮することができる。

[0083] 分散処理システムの他の構成要素については、第1の実施例で説明したものと同一であるため、本実施例では説明を省略する。

[0084] 第1、第2の実施例で説明した集計処理ノード1と分散処理ノード2 [n] の各々は、CPU (Central Processing Unit)、記憶装置及びインタフェースを備えたコンピュータと、これらのハードウェア資源を制御するプログラムによって実現することができる。集計処理ノード1と分散処理ノード2 [n] の各々のCPUは、各々の記憶装置に格納されたプログラムに従って第1、第2の実施例で説明した処理を実行する。

産業上の利用可能性

[0085] 本発明は、ニューラルネットワークの機械学習を行う技術に適用することができる。

符号の説明

[0086] 1…集計処理ノード1、2…分散処理ノード、10…受信部、11…受信FIFOバッファ、12…集計処理部、13…送信部、20…サンプル入力部、21…勾配計算処理部、22…ノード内集計処理部、23…送信部、24…受信部、25…重み更新処理部、26…ニューラルネットワーク。

請求の範囲

[請求項1] 集計処理ノードとN個（Nは2以上の整数）の分散処理ノードとを備え、

各分散処理ノードは、学習対象のニューラルネットワークのM個（Mは2以上の整数）の重み $w[m]$ 毎（ $m=1, \dots, M$ ）の分散データ $D[m, n]$ （ $n=1, \dots, N$ ）を重み $w[m]$ の番号mの順番にパケット化して前記集計処理ノードに送信し、前記集計処理ノードから送信されたパケットを受信して集計データ $R[m]$ を番号mの順番に取得して、この集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新し、

前記集計処理ノードは、各分散処理ノードから送信されたパケットを受信して分散データ $D[m, n]$ を番号mの順番に取得し、全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した前記集計データ $R[m]$ を生成して、この集計データ $R[m]$ を番号mの順番にパケット化して各分散処理ノードに送信することを特徴とする分散処理システム。

[請求項2] 請求項1記載の分散処理システムにおいて、

各分散処理ノードは、

前記分散データ $D[m, n]$ を番号mの順番にパケット化して前記集計処理ノードに送信する送信部と、

前記集計処理ノードから送信されたパケットを受信して前記集計データ $R[m]$ を番号mの順番に取得する受信部と、

前記集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新する重み更新処理部とを備えることを特徴とする分散処理システム。

[請求項3] 請求項1または2記載の分散処理システムにおいて、

前記集計処理ノードは、

各分散処理ノードから送信されたパケットを受信して前記分散デー

タ $D[m, n]$ を番号 m の順番に取得する受信部と、

全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した前記集計データ $R[m]$ を生成する集計処理部と、

前記集計データ $R[m]$ を番号 m の順番にパケット化して各分散処理ノードに送信する送信部とを備えることを特徴とする分散処理システム。

[請求項4] 請求項 1 乃至 3 のいずれか 1 項に記載の分散処理システムにおいて、

各分散処理ノードは、

前記ニューラルネットワークの学習用のサンプルデータが入力されたときに、前記ニューラルネットワークの重み $w[m]$ の各々について、前記ニューラルネットワークの損失関数の勾配をサンプルデータ毎に計算する勾配計算処理部と、

サンプルデータ毎の前記勾配を集計した数値である前記分散データ $D[m, n]$ を重み $w[m]$ 毎に生成して保持するノード内集計処理部とを、さらに備えることを特徴とする分散処理システム。

[請求項5] 請求項 1 乃至 4 のいずれか 1 項に記載の分散処理システムにおいて、

前記集計処理ノードと各分散処理ノードとは、

各分散処理ノードがパケット化した分散データ $D[m, n]$ を前記集計処理ノードに送信して、前記集計処理ノードが受信したパケットから前記分散データ $D[m, n]$ を取得する集約通信処理と、前記集計処理ノードが前記集計データ $R[m]$ を生成する全ノード集計処理と、前記集計処理ノードがパケット化した前記集計データ $R[m]$ を各分散処理ノードに送信して、各分散処理ノードが受信したパケットから前記集計データ $R[m]$ を取得する分配通信処理と、各分散処理ノードが前記重み $w[m]$ を更新する重み更新処理とを、それぞれ異なる番号 m について並行して行うことを特徴とする分散処理システム

。

[請求項6] N個（Nは2以上の整数）の分散処理ノードの各々が、学習対象のニューラルネットワークのM個（Mは2以上の整数）の重み $w[m]$ 毎（ $m=1, \dots, M$ ）の分散データ $D[m, n]$ （ $n=1, \dots, N$ ）を重み $w[m]$ の番号mの順番にパケット化して集計処理ノードに送信する第1のステップと、

前記集計処理ノードが、各分散処理ノードから送信されたパケットを受信して前記分散データ $D[m, n]$ を番号mの順番に取得する第2のステップと、

前記集計処理ノードが、全分散処理ノードの分散データ $D[m, n]$ を重み $w[m]$ 毎に集計した集計データ $R[m]$ を生成する第3のステップと、

前記集計処理ノードが、前記集計データ $R[m]$ を番号mの順番にパケット化して各分散処理ノードに送信する第4のステップと、

前記分散処理ノードの各々が、前記集計処理ノードから送信されたパケットを受信して前記集計データ $R[m]$ を番号mの順番に取得する第5のステップと、

前記分散処理ノードの各々が、前記集計データ $R[m]$ に基づいて前記ニューラルネットワークの重み $w[m]$ を更新する第6のステップとを含むことを特徴とする分散処理方法。

[請求項7] 請求項6記載の分散処理方法において、

前記第1のステップの前に、前記分散処理ノードの各々が、前記ニューラルネットワークの学習用のサンプルデータが入力されたときに、前記ニューラルネットワークの重み $w[m]$ の各々について、前記ニューラルネットワークの損失関数の勾配をサンプルデータ毎に計算する第7のステップと、

前記分散処理ノードの各々が、サンプルデータ毎の前記勾配を集計した数値である前記分散データ $D[m, n]$ を重み $w[m]$ 毎に生成

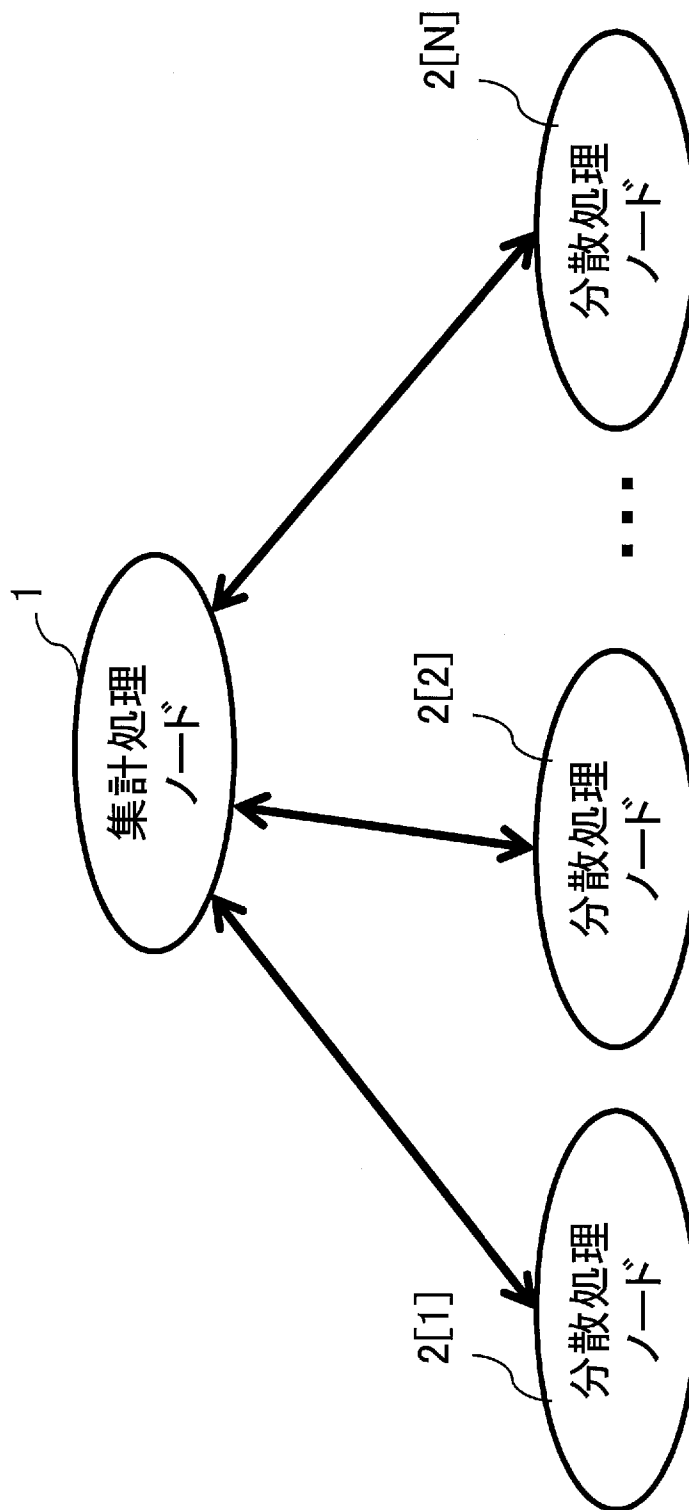
して保持する第8のステップとを、さらに含むことを特徴とする分散処理方法。

[請求項8]

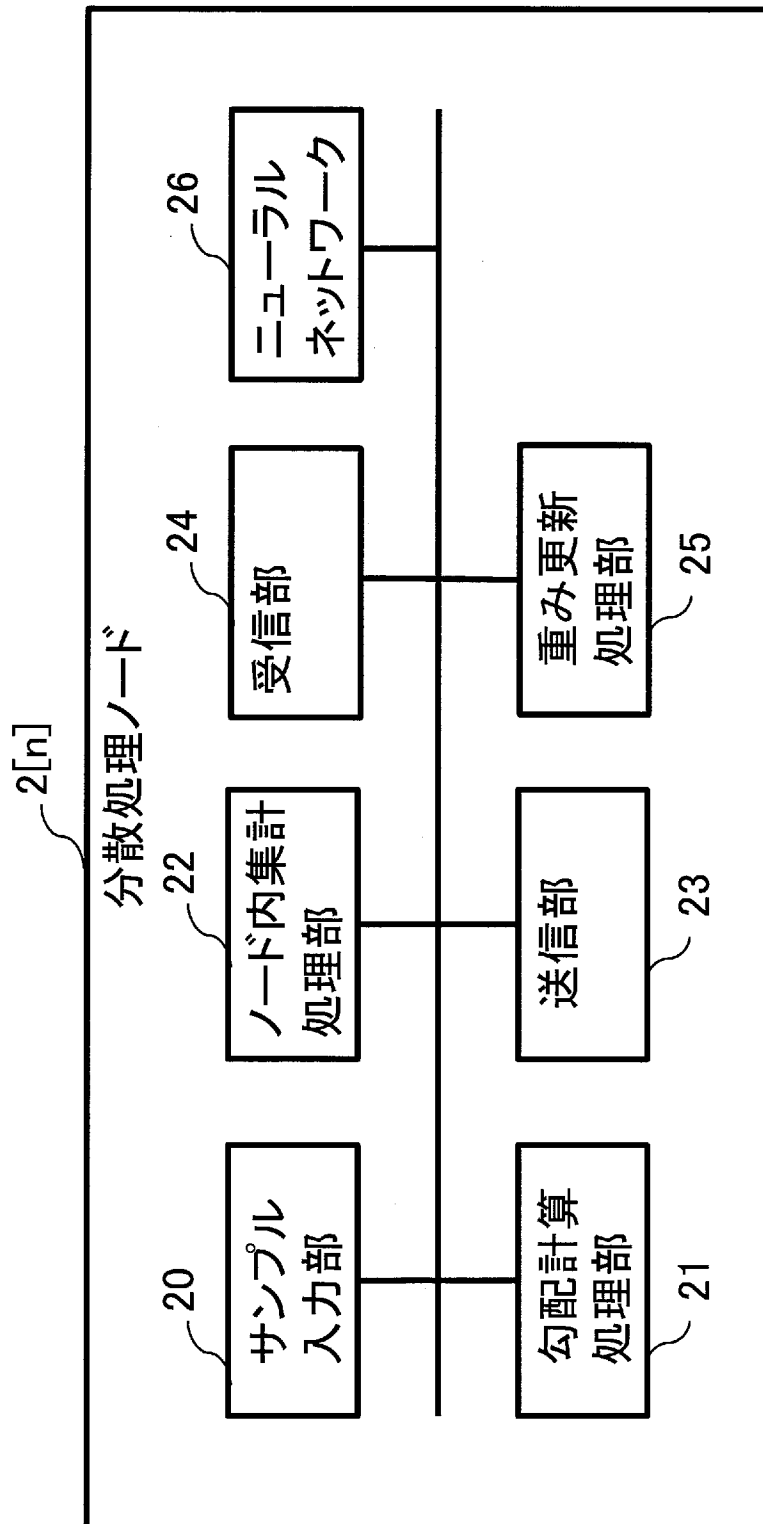
請求項6または7記載の分散処理方法において、

前記分散処理ノードの前記第1のステップおよび前記集計処理ノードの前記第2のステップと、前記集計処理ノードの前記第3のステップと、前記集計処理ノードの前記第4のステップおよび前記分散処理ノードの前記第5のステップと、前記分散処理ノードの前記第6のステップとを、それぞれ異なる番号 m について並行して行うことを特徴とする分散処理方法。

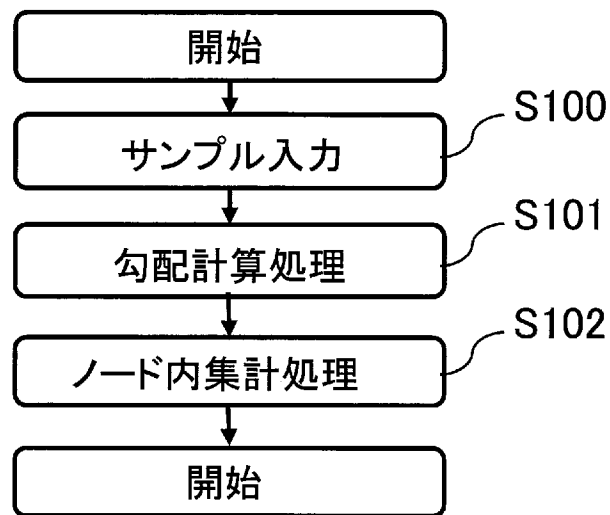
[図1]



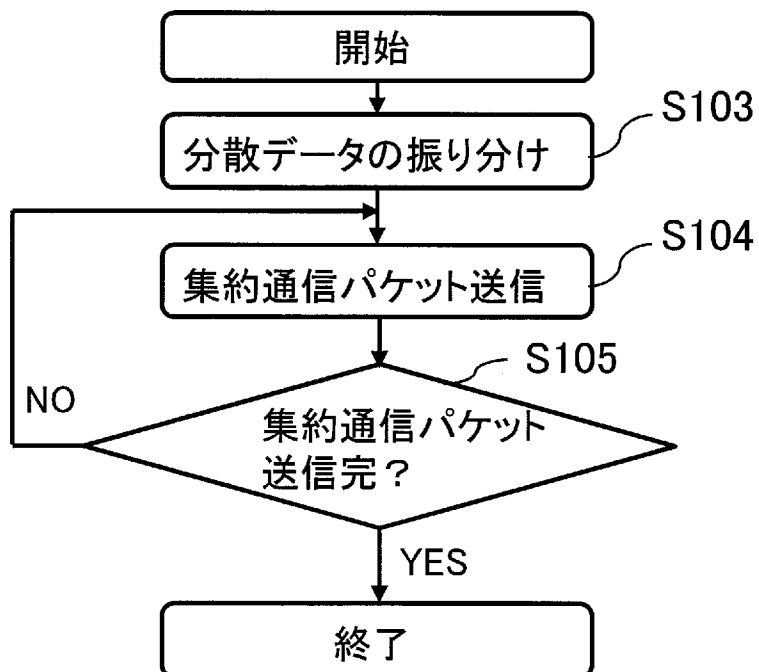
[図2]



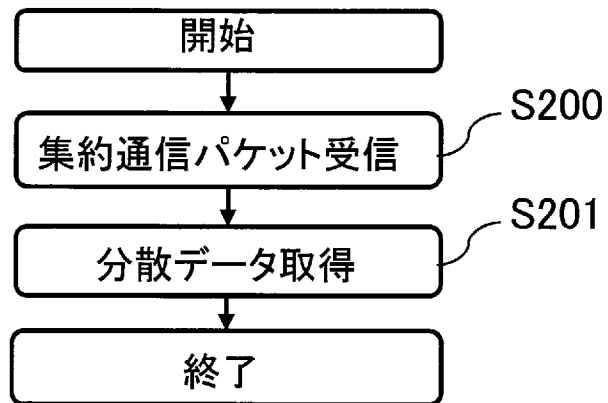
[図3]



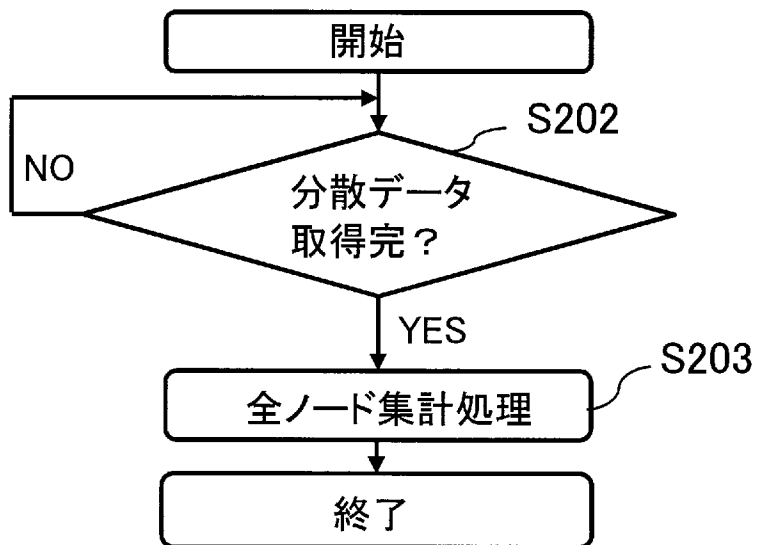
[図4]



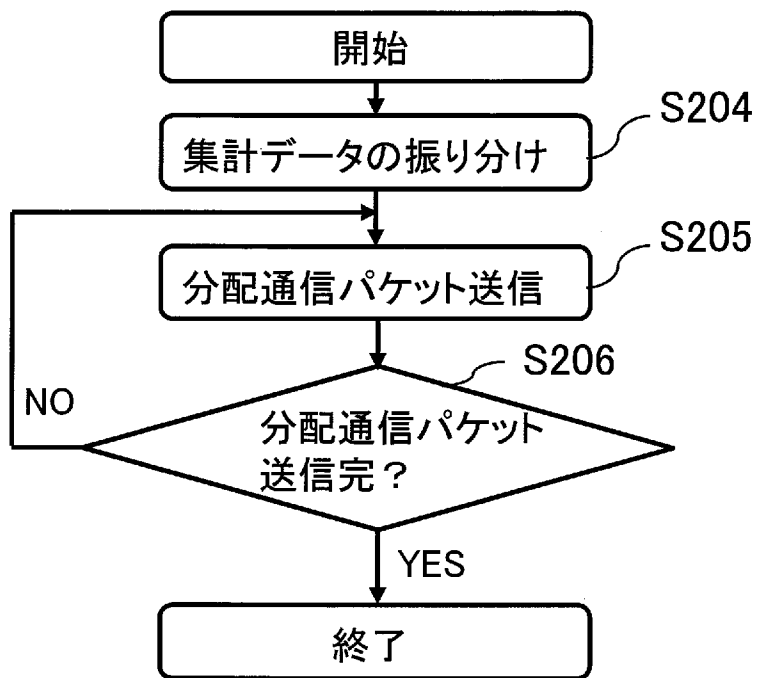
[図5]



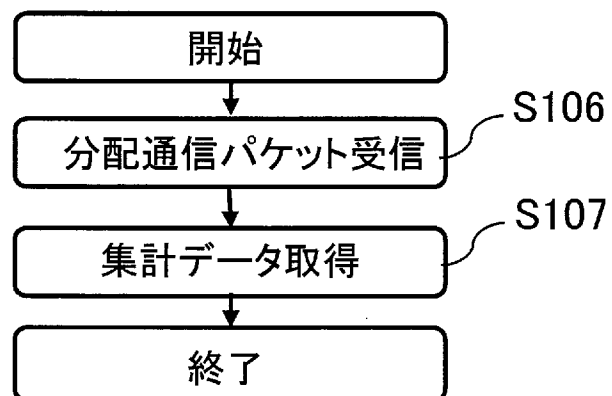
[図6]



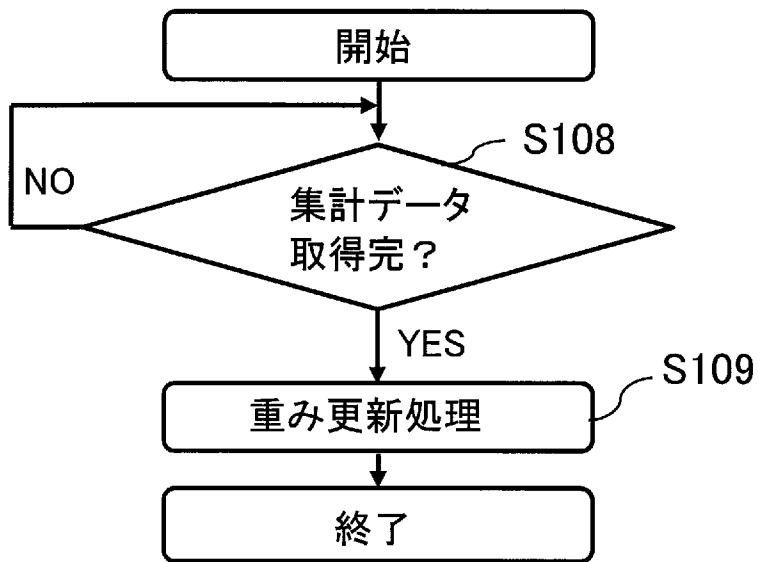
[図7]



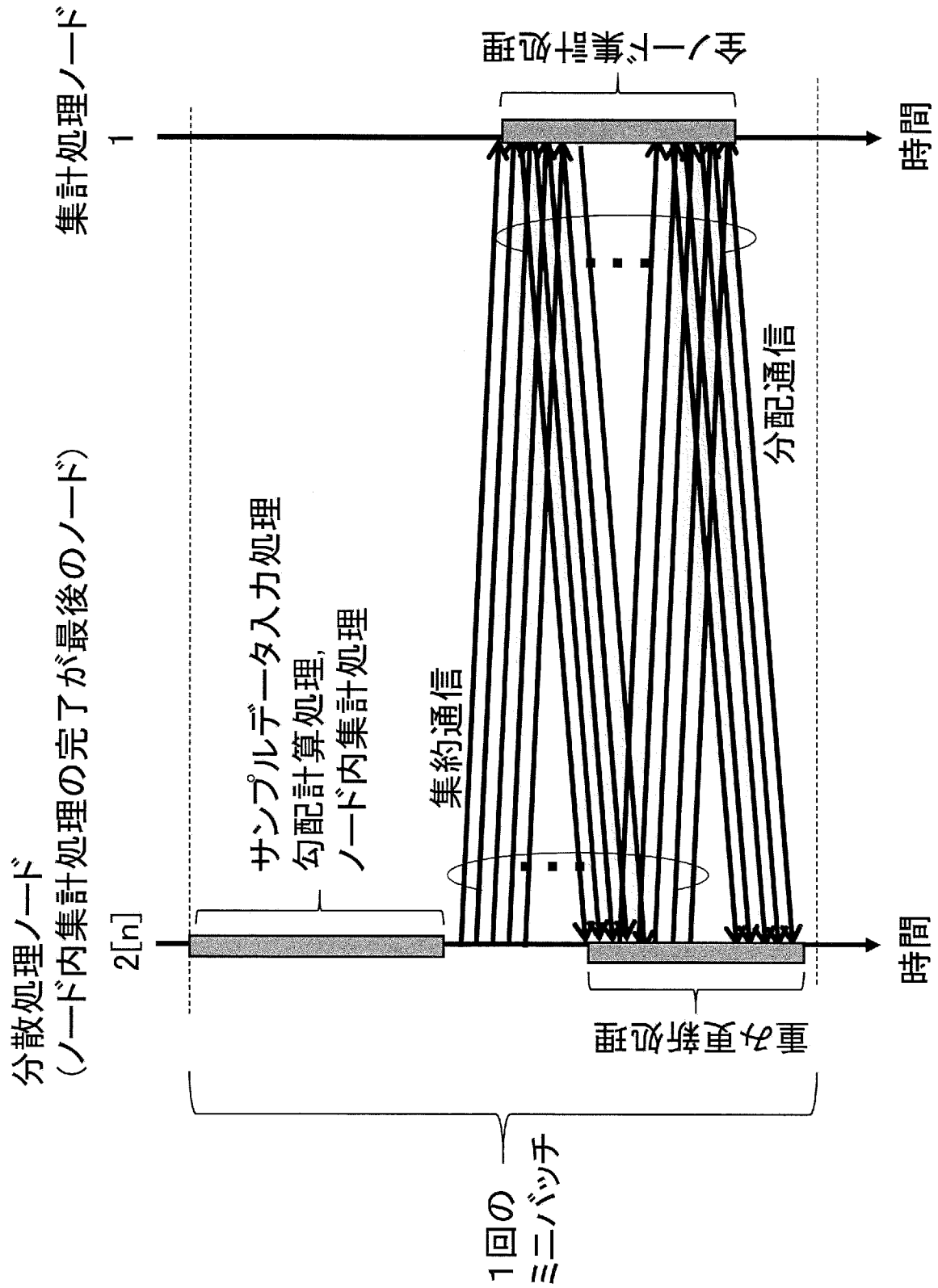
[図8]



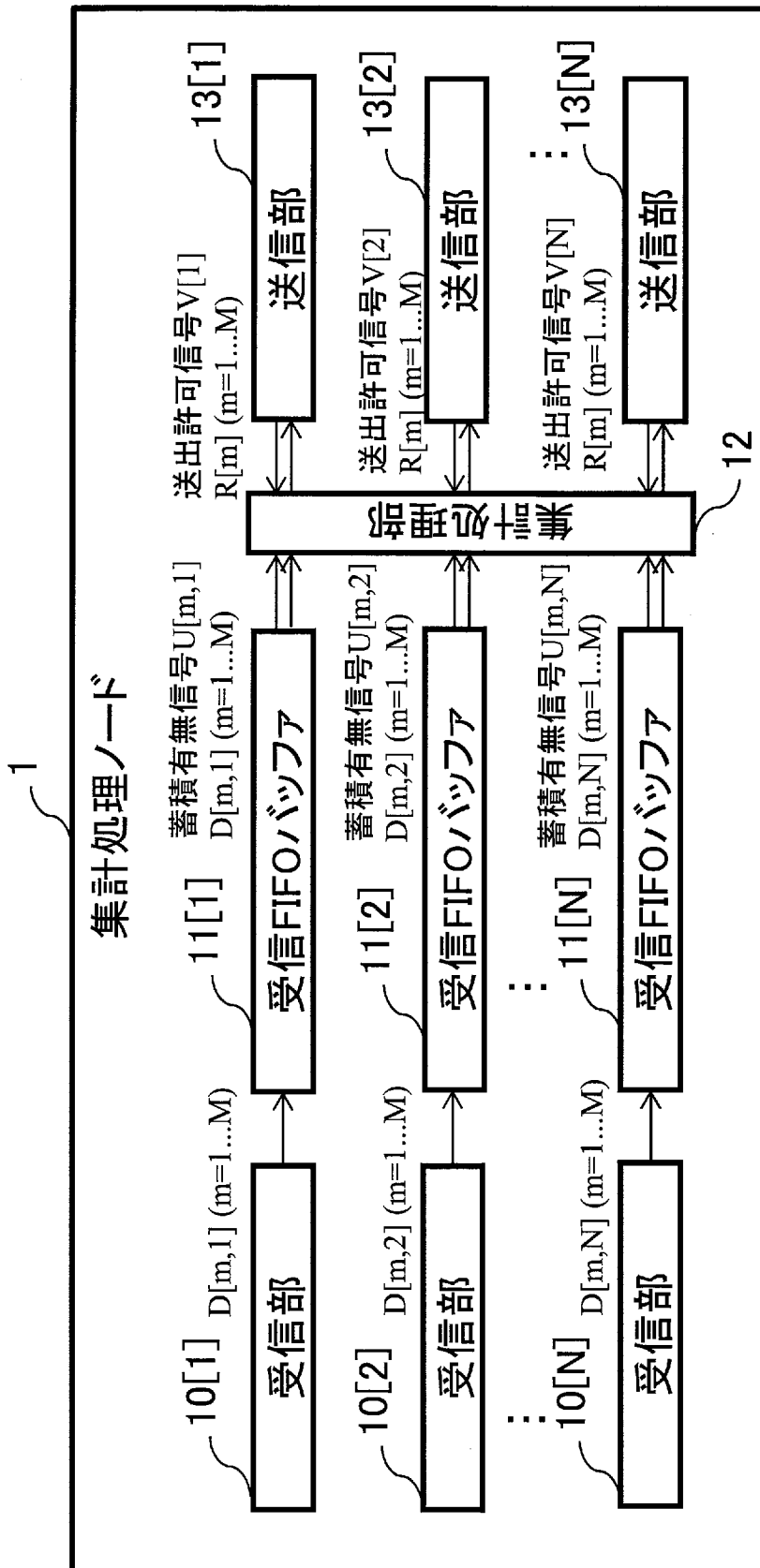
[図9]



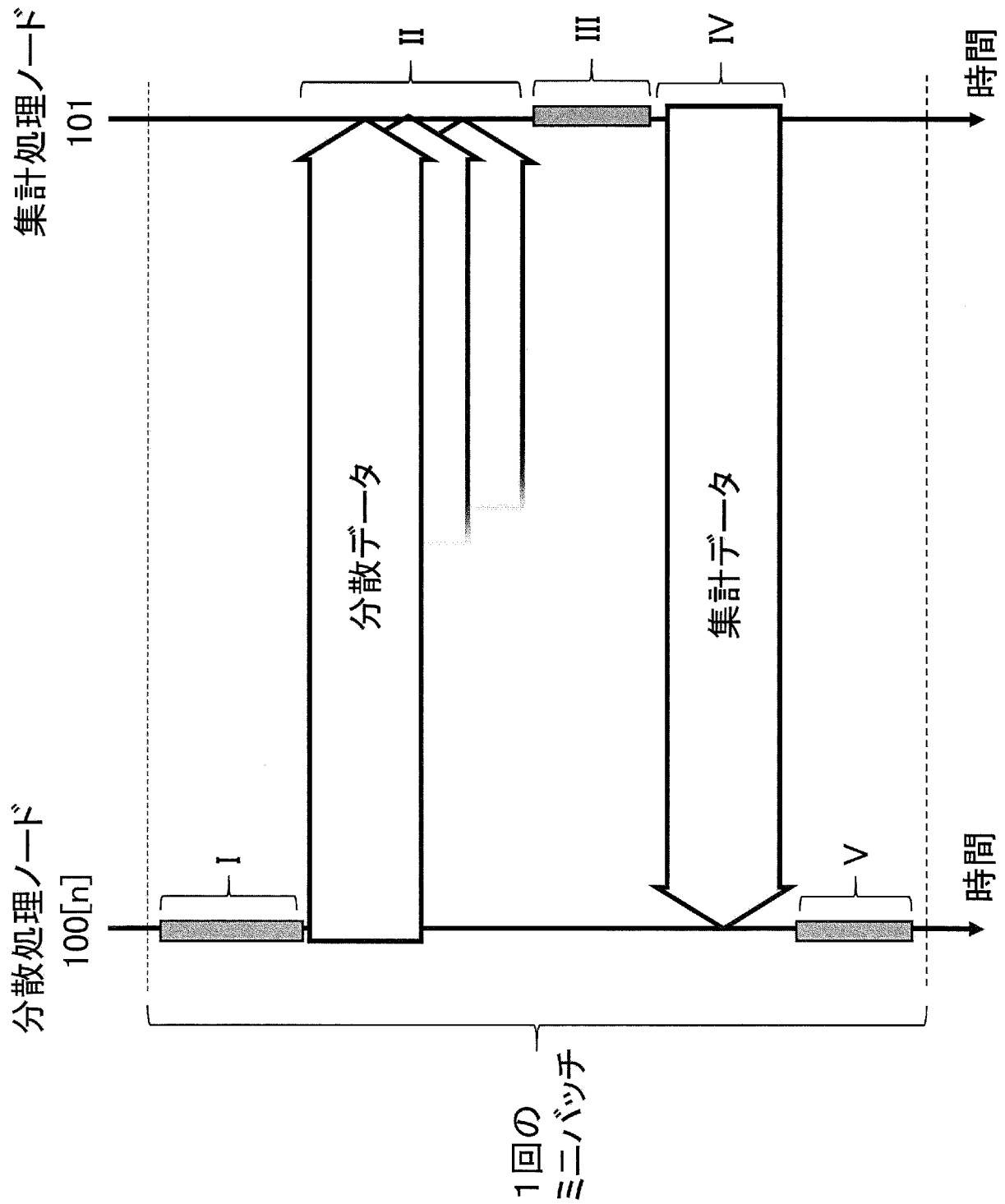
[図10]



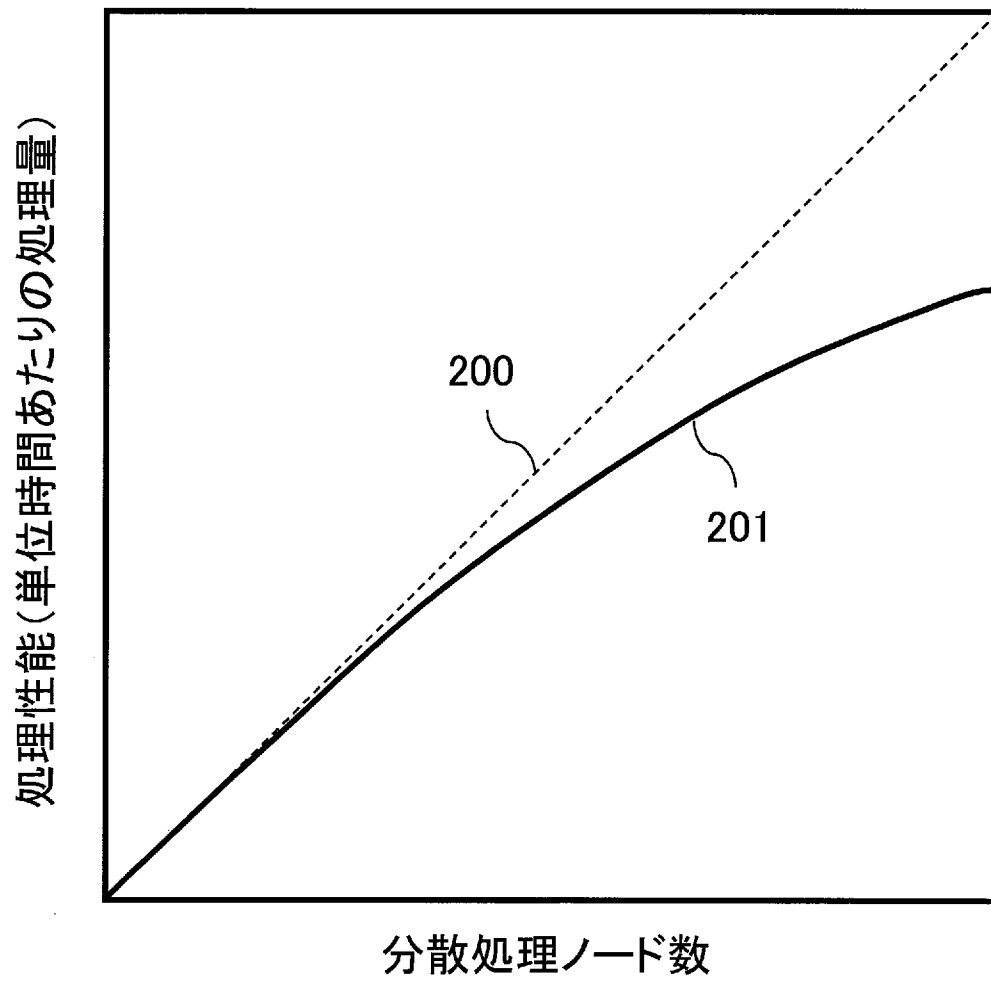
[図11]



[図12]



[図13]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2019/004214

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06N3/08 (2006.01) i, G06N3/04 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06N3/08, G06N3/04

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2019
Registered utility model specifications of Japan	1996-2019
Published registered utility model applications of Japan	1994-2019

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	山崎 雅文 他, 「MPI を用いた Deep Learning 処理高速化の提案」, 情報処理学会研究報告ハイパフォーマンスコンピューティング(HPC), 08 August 2016, vol. 2016-HPC-155, no. 6, pp. 1-8, ISSN:2188-8841, (YAMAZAKI, Masafumi et al., "Accelerating Deep Learning Framework with MPI", IPSJ SIG Technical Report: High-performance computing (HPC))	1-4, 6-8 5

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search
22 April 2019 (22.04.2019)

Date of mailing of the international search report
07 May 2019 (07.05.2019)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2019/004214

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	@sz_dr, 「ニューラルネットワークの順伝播, 逆伝播, 確率的勾配降下法を手計算する」, [online], Increments Inc., 21 March 2016, [retrieval date 22 April 2019], Internet <URL:https://qiita.com/sz_dr/items/800d043f8ff92f49c18c>, in particular, see description entitled, "back propagation" and "Learning by stochastic gradient descent", non-official translation ("Manual calculation of forward propagation, back propagation, and stochastic gradient descent in neural networks")	1-4, 6-8 5
A	LI, Mu et al., "Scaling Distributed Machine Learning with the Parameter Server", Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation, USENIX, 08 October 2014, pp. 583-598, ISBN:978-1-931971-16-4, [retrieval date 22 April 2019], Internet <URL:https://www.cs.cmu.edu/~muli/file/parameter_server_osdi14.pdf>	1-8

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06N3/08(2006.01)i, G06N3/04(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06N3/08, G06N3/04

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2019年
日本国実用新案登録公報	1996-2019年
日本国登録実用新案公報	1994-2019年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y A	山崎 雅文 他, 「M P I を用いた D e e p L e a r n i n g 処理 高速化の提案」, 情報処理学会研究報告 ハイパフォーマン スコンピューティング(H P C), 2016. 08. 08, Vol. 2016-HPC-155 No. 6, 第1頁-第8頁, ISSN:2188-8841	1-4, 6-8 5

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

22.04.2019

国際調査報告の発送日

07.05.2019

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

三坂 敏夫

5 B

4178

電話番号 03-3581-1101 内線 3545

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y A A	@ s z _ d r, 「ニューラルネットワークの順伝播, 逆伝播, 確率的勾配降下法を手計算する」, [online], Increments Inc., 2016.03.21, [検索日 2019.04.22], インターネット<URL: https://qiita.com/sz_dr/items/800d043f8ff92f49c18c>, 特に「逆伝搬」及び「確率的勾配降下法による学習」と題した記載を参照 LI, Mu et al., "Scaling Distributed Machine Learning with the Parameter Server", Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation, USENIX, 2014.10.08, pages:583-598, ISBN:978-1-931971-16-4, [検索日 2019.04.22], インターネット<URL: https://www.cs.cmu.edu/~muli/file/parameter_server_osdi14.pdf>	1-4, 6-8 5 1-8