



(51) 国際特許分類:

G06F 15/173 (2006.01) G06N 3/08 (2006.01)

(21) 国際出願番号:

PCT/JP2019/039449

(22) 国際出願日:

2019年10月7日(07.10.2019)

(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 2018-198230 2018年10月22日(22.10.2018) 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).

ゴーフイクー(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).

(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,

(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

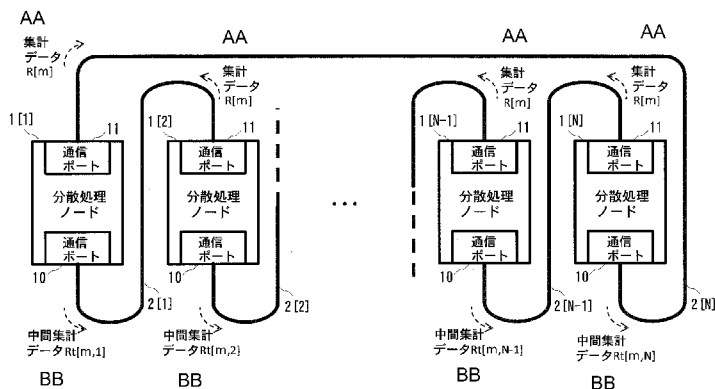


Fig. 1

1[1], 1[2], 1[N-1], 1[N] Distributed processing node

10, 11 Communication port

AA Aggregate data

BB Intermediate aggregate data

(57) Abstract: According to the present invention, a distributed processing node 1[1] transmits distributed data from a communication port 10 to a distributed processing node 1[2] as intermediate aggregate data. A distributed processing node 1[k] ($k=2, \dots, N$) generates updated intermediate aggregate data from received intermediate aggregate data and distributed data and transmits the updated intermediate aggregate data from a communication port 10 to a distributed processing node 1[k⁺] ($k^+=k+1$, provided that $k^+=1$ when $k=N$). Distributed processing node 1[1] transmits intermediate aggregate data received via a communication port 11 from communication port 11 to distributed processing node 1[N] as aggregate data. Distributed processing node 1[k] transmits aggregate data received via communication port 10 from a communication port 11 to distributed processing node 1[k-1]. Each distributed processing node 1 updates the weight of a neural network on the basis of aggregate data. As a result, the present invention can perform effective distributed processing when applied to deep learning.

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: 分散処理ノード1 [1] は、分散データを中間集計データとして通信ポート1 0から分散処理ノード1 [2] に送信する。分散処理ノード1 [k] ($k = 2, \dots, N$) は、受信した中間集計データと分散データとから更新後の中間集計データを生成して、通信ポート1 0から分散処理ノード1 [k^+] ($k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$) に送信する。分散処理ノード1 [1] は、通信ポート1 1を介して受信した中間集計データを集計データとして、通信ポート1 1から分散処理ノード1 [N] に送信する。分散処理ノード1 [k] は、通信ポート1 0を介して受信した集計データを通信ポート1 1から分散処理ノード1 [$k - 1$] に送信する。各分散処理ノード1 は、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。これにより、深層学習に適用した場合に効果的な分散処理を行うことができる。

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、複数の分散処理ノードを備える分散処理システムに係り、特に、各分散処理ノードから数値データを集計して集計データを生成し、各分散処理ノードに集計データを分配する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、各分散処理ノードがサンプルデータ毎に重みに対する勾配を計算する勾配計算処理およびサ

ンプルデータ毎に得られた勾配を重み別に合算するノード内集計処理と、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の間に、分散処理ノード毎に得られたデータ（分散データ）を、集計処理を行うノードに転送するための通信（集約通信）と、集約通信により取得したデータに基づいて集計する処理（ノード間集計処理）と、各分散処理ノードから取得した集計したデータ（集計データ）を各分散処理ノードに分配するための通信（分配通信）と、が必要となる。

[0007] 上記の集約通信や分配通信に要する時間は、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。

[0008] このように、深層学習の分散処理システムでは、集約通信時間と分配通信時間の増大によって、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。

[0009] 図12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。分散処理ノード数に比例してノード間集計処理の入力である分散データの総量は増大するが、実際の処理性能が分散処理ノード数に比例して向上しない理由は、集計処理ノードの通信速度が、このノードの通信ポートの物理速度以下に制限されるため、集約通信に要する時間が増大するためである。

先行技術文献

非特許文献

[0010] 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），201

7 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

- [0011] 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、複数の分散処理ノードを備える分散処理システムにおいて、深層学習に適用した場合に効果的な分散処理を行う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

- [0012] 本発明の分散処理システムは、リング状に配置され、隣接するノードと通信路を介して互いに接続された N 個 (N は2以上の整数)の分散処理ノードを備え、 n 番目 ($n = 1, \dots, N$)の分散処理ノードは、 n^+ 番目 ($n^+ = n + 1$ 、ただし $n = N$ の場合は $n^+ = 1$)の分散処理ノードと双方向の通信が同時に可能な第1の通信ポートと、 n^- 番目 ($n^- = n - 1$ 、ただし $n = 1$ の場合は $n^- = N$)の分散処理ノードと双方向の通信が可能な同時に第2の通信ポートとを備え、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 ($m = 1, \dots, M$)の分散データを生成し、 N 個の分散処理ノードのうち、予め指定された1番目の分散処理ノードは、自ノードで生成された分散データを第1の集計データとして、この第1の集計データを重み $w[m]$ の番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから2番目の分散処理ノードに向けて送信し、 N 個の分散処理ノードのうち、前記1番目を除く k 番目 ($k = 2, \dots, N$)の分散処理ノードは、($k - 1$)番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データと自ノードで生成された分散データとの和を、対応する重み $w[m]$ 毎に求めて更新後の第1の集計データを生成し、この第1の集計データを前記番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから k^+ 番目 ($k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$)の分散処理ノードに向けて送信し、前記1

番目の分散処理ノードは、N番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データを第2の集計データとして、この第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから前記N番目の分散処理ノードに向けて送信し、前記k番目の分散処理ノードは、k+番目の分散処理ノードから自ノードの前記第1の通信ポートを介して受信した第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから(k-1)番目の分散処理ノードに向けて送信し、前記1番目の分散処理ノードは、2番目の分散処理ノードから自ノードの前記第1の通信ポートを介して第2の集計データを受信し、各分散処理ノードは、受信した前記第2の集計データに基づいて前記ニューラルネットワークの重み $w[m]$ を更新することを特徴とするものである。

[0013] 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、前記分散データを生成するノード内集計処理部と、自ノードが前記1番目の分散処理ノードとして機能する場合に、前記第1の集計データを重み $w[m]$ の番号mの順番にパケット化して、自ノードの前記第1の通信ポートから2番目の分散処理ノードに向けて送信し、自ノードが前記k番目の分散処理ノードとして機能する場合に、前記更新後の第1の集計データを前記番号mの順番にパケット化して、自ノードの前記第1の通信ポートからk+番目の分散処理ノードに向けて送信する第1の送信部と、自ノードの前記第2の通信ポートから受信するパケットから前記第1の集計データを取得する第1の受信部と、自ノードが前記1番目の分散処理ノードとして機能する場合に、前記第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから前記N番目の分散処理ノードに向けて送信し、自ノードが前記k番目の分散処理ノードとして機能する場合に、受信した前記第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから(k-1)番目の分散処理ノードに向けて送信する第2の送信部と、自ノードの前記第1の通信ポートから受信するパケットから前

記第2の集計データを取得する第2の受信部と、自ノードが前記k番目の分散処理ノードとして機能する場合に、前記更新後の第1の集計データを生成する集計データ生成部と、受信した前記第2の集計データ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、各分散処理ノードは、前記1番目の分散処理ノードが前記第2の集計データを正常に受信できなかった場合に、前記第1の集計データの送信以降の処理を再度行うことを特徴とするものである。

[0014] また、本発明は、リング状に配置され、隣接するノードと通信路を介して互いに接続された N 個（ N は2以上の整数）の分散処理ノードを備え、 n 番目（ $n=1, \dots, N$ ）の分散処理ノードが、 n^+ 番目（ $n^+=n+1$ 、ただし $n=N$ の場合は $n^+=1$ ）の分散処理ノードと双方向の通信が同時に可能な第1の通信ポートと、 n^- 番目（ $n^-=n-1$ 、ただし $n=1$ の場合は $n^-=N$ ）の分散処理ノードと双方向の通信が可能な同時に第2の通信ポートとを備えたシステムにおける分散処理方法であって、各分散処理ノードが、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（ $m=1, \dots, M$ ）の分散データを生成する第1のステップと、 N 個の分散処理ノードのうち、予め指定された1番目の分散処理ノードが、自ノードで生成された分散データを第1の集計データとして、この第1の集計データを重み $w[m]$ の番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから2番目の分散処理ノードに向けて送信する第2のステップと、 N 個の分散処理ノードのうち、前記1番目を除く k 番目（ $k=2, \dots, N$ ）の分散処理ノードが、（ $k-1$ ）番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データと自ノードで生成された分散データとの和を、対応する重み $w[m]$ 毎に求めて更新後の第1の集計データを生成し、この第1の集計データを前記番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから k^+ 番目（ $k^+=k+1$ 、ただ

し $k = N$ の場合は $k^+ = 1$) の分散処理ノードに向けて送信する第3のステップと、前記1番目の分散処理ノードが、 N 番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データを第2の集計データとして、この第2の集計データを前記番号 m の順番にパケット化して、自ノードの前記第2の通信ポートから前記 N 番目の分散処理ノードに向けて送信する第4のステップと、前記 k 番目の分散処理ノードが、 k^+ 番目の分散処理ノードから自ノードの前記第1の通信ポートを介して受信した第2の集計データを前記番号 m の順番にパケット化して、自ノードの前記第2の通信ポートから $(k - 1)$ 番目の分散処理ノードに向けて送信する第5のステップと、前記1番目の分散処理ノードが、2番目の分散処理ノードから自ノードの前記第1の通信ポートを介して第2の集計データを受信する第6のステップと、各分散処理ノードが、受信した前記第2の集計データ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7のステップとを含むことを特徴とするものである。

[0015] また、本発明の分散処理方法の1構成例において、前記第3のステップは、前記 k 番目の分散処理ノードが、自ノードの前記第2の通信ポートから受信するパケットから前記第1の集計データを取得するステップと、前記更新後の第1の集計データを生成するステップと、前記更新後の第1の集計データを前記番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから k^+ 番目の分散処理ノードに向けて送信するステップとを含み、前記第4のステップは、前記1番目の分散処理ノードが、自ノードの前記第2の通信ポートから受信するパケットから前記第1の集計データを取得するステップと、取得した第1の集計データを第2の集計データとして、この第2の集計データを前記番号 m の順番にパケット化して、自ノードの前記第2の通信ポートから前記 N 番目の分散処理ノードに向けて送信するステップとを含み、前記第5のステップは、前記 k 番目の分散処理ノードが、自ノードの前記第1の通信ポートから受信するパケットから前記第2の集計データを取得するステップと、受信した前記第2の集計データを前記番号 m の順番にパケット

化して、自ノードの前記第2の通信ポートから $(k-1)$ 番目の分散処理ノードに向けて送信するステップとを含み、前記第6のステップは、前記1番目の分散処理ノードが、自ノードの前記第1の通信ポートから受信するパケットから前記第2の集計データを取得するステップを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構成例において、各分散処理ノードは、前記第6のステップにおいて前記1番目の分散処理ノードが前記第2の集計データを正常に受信できなかった場合に、前記第2のステップ以降の処理を再度行うことを特徴とするものである。

発明の効果

[0016] 本発明によれば、 n 番目 $(n=1, \dots, N)$ の分散処理ノードから n^+ 番目 $(n^+=n+1, \text{ただし } n=N \text{ の場合は } n^+=1)$ の分散処理ノードへの集約通信(第1の集計データを n^+ 番目の分散処理ノードに送信する処理)、 k 番目 $(k=2, \dots, N)$ の分散処理ノードが行うノード間集計処理(受信した第1の集計データと自ノードで生成した分散データとに基づいて更新後の第1の集計データを計算する処理)、 n 番目の分散処理ノードから n^- 番目 $(n^-=n-1, \text{ただし } n=1 \text{ の場合は } n^-=N)$ の分散処理ノードへの分配通信(第2の集計データを n^- 番目の各分散処理ノードに分配する処理)とを、並行してほぼ同時に行うことが可能であり、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。本発明では、各分散処理ノードに第1の通信ポートと第2の通信ポートとを設け、集約通信と分配通信の方向を逆にすることにより、集約通信が完了するまで分配通信の開始を待つ必要がない。また、本発明では、集計処理ノードを設けることなく深層学習の分散処理を行うことができ、集計処理ノードの通信速度によって分散処理の速度が制限されることがなくなる。また、本発明では、 N 個の分散処理ノードが同一のハードウェアを備えるノードであっても、1個の親ノード(1番目の分散処理ノード)を定めて親ノードか否かに応じた設定を各分散処理ノードに対して行うことにより、集約通信処理と

ノード間集計処理と分配通信処理とを実現することが可能となるので、各分散処理ノードに個別に設定を要するシステムと比較して、その管理が極めて簡易になるので、システム管理に要するコストや管理上のミスを削減することができる。

[0017] また、本発明では、各分散処理ノードは、1番目の分散処理ノードが第2の集計データを正常に受信できなかった場合に、第1の集計データの送信以降の処理を再度行う。本発明では、1番目の分散処理ノードにおいて送出した第2の集計データが戻ることで全分散処理ノードの正常な処理が保証されるため、各分散処理ノードの状態監視が不要であり、1番目の分散処理ノードのみで簡易かつ低遅延でデータのインテグリティ（完全性）を保証することができる。

図面の簡単な説明

[0018] [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集約通信処理とノード間集計処理と分配通信処理と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重み更新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2の実施例に係る分散処理ノードの構成例を示すブロック図である。

[図6]図6は、本発明の第2の実施例に係る分散処理ノードの構成例を示すブロック図である。

[図7]図7は、本発明の第2の実施例に係る分散処理ノードの処理の概要を示す図である。

[図8]図8は、本発明の第2の実施例に係る分散処理ノード間の中間集計デー

タおよび集計データの通信のシーケンスを示す図である。

[図9]図9は、本発明の第2の実施例に係る分散処理ノード間の中間集計データおよび集計データの通信のシーケンスを示す図である。

[図10]図10は、本発明の第2の実施例に係る分散処理ノード間の中間集計データおよび集計データの通信のシーケンスを示す図である。

[図11]図11は、本発明の第1、第2の実施例に係る分散処理ノードを実現するコンピュータの構成例を示すブロック図である。

[図12]図12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

[0019] [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、 N 個（ N は2以上の整数）の分散処理ノード1[n]（ $n=1, \dots, N$ ）と、番号 n の分散処理ノード1[n]（ $n=1, \dots, N$ ）が次の番号 n^+ （ $n^+=n+1$ 、ただし $n=N$ の場合は $n^+=1$ ）の分散処理ノード1[n⁺]と互いに双方向に通信するための通信路2[n]（ $n=1, \dots, N$ ）とを備えている。なお、任意の通信路2[n]（ $n=1, \dots, N$ ）には、伝送路の他に、通信を中継する中継処理ノードが任意に介在することも可能である。

[0020] 各分散処理ノード1[n]（ $n=1, \dots, N$ ）は、双方向の通信が同時に可能な通信ポート10と通信ポート11とを備える。通信ポート10は、分散処理ノード1[n]が分散処理ノード1[n⁺]（ $n^+=n+1$ 、ただし $n=N$ の場合は $n^+=1$ ）と双方向の通信を行うための通信ポートであり、通信路2[n]と接続される。また、通信ポート11は、分散処理ノード1[n]が分散処理ノード1[n⁻]（ $n^-=n-1$ 、ただし $n=1$ の場合は $n^-=N$ ）と双方向の通信を行うための通信ポートであり、通信路2[n⁻]と接続される。

[0021] 図2は分散処理ノード1 [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とを説明するフローチャートである。各分散処理ノード1 [n] ($n = 1, \dots, N$) 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2ステップS100)。

[0022] 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1 [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。

[0023] 各分散処理ノード1 [n] ($n = 1, \dots, N$) 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2ステップS101)。

[0024] ニューラルネットワークを各分散処理ノード1 [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の重み $w[m]$ 、ニューラルネットワーク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。

[0025] 続いて、各分散処理ノード1 [n] ($n = 1, \dots, N$) 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 $D[m, n]$ ($m = 1, \dots, M$) を、重み $w[m]$ 毎に生成して保持する (図2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。

[0026] [数1]

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

[0027] なお、ステップS101の勾配計算処理とステップS102の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。

[0028] さらに、各分散処理ノード1 [n] ($n = 1, \dots, N$) は、分散データ $D[m, n]$ ($m = 1, \dots, M$) を生成した後、分散処理ノード間の集約通信を行い、集計データを生成するためのノード間集計処理を行う。

図3は分散処理ノード1 [n] の集約通信処理とノード間集計処理と分配通信処理とを説明するフローチャートである。

[0029] まず、複数の分散処理ノード1 [n] ($n = 1, \dots, N$) のうち、予め定められた1番目の分散処理ノード1 [1] は、自ノードで生成されたM個の分散データ $D[m, 1]$ ($m = 1, \dots, M$) を、中間集計データ $R_t[m, 1]$ として、自ノードの通信ポート10および通信路2 [1] を介して次の番号の分散処理ノード1 [2] に送信する（図3ステップS103, S104）。すなわち、このときの中間集計データ $R_t[m, 1]$ は、分散データ $D[m, 1]$ と同じである。

$$R_t[m, 1] = D[m, 1] \quad \dots (2)$$

[0030] 次に、複数の分散処理ノード1 [n] ($n = 1, \dots, N$) のうち、1番目とN番目とを除く、予め定められた中間の分散処理ノード1 [i] ($i = 2, \dots, N-1$) は、分散処理ノード1 [i-1] から中間集計データ $R_t[m, i-1]$ ($m = 1, \dots, M$) を自ノードの通信ポート11および通信路2 [i-1] を介して受信する（図3ステップS105, S106）。

[0031] 中間の分散処理ノード1 [i] ($i = 2, \dots, N-1$) は、受信した中間集計データ $R_t[m, i-1]$ ($m = 1, \dots, M$) と自ノードで生成された分散データ $D[m, i]$ との和を、対応する重み $w[m]$ 毎に求めることにより、中間集計データ $R_t[m, i]$ を生成する（図3ステップS107）。すなわち、中間集計データ $R_t[m, i]$ は、M個の数値から構成される。中間集計データ $R_t[m, i]$ の計算式は以下のとおりである。

$$R_t[m, i] = R_t[m, i-1] + D[m, i] \quad \dots (3)$$

[0032] そして、中間の分散処理ノード1 [i] ($i = 2, \dots, N-1$) は、自ノードで生成された中間集計データ $R_t[m, i]$ ($m = 1, \dots, M$) を自ノードの通信ポート10および通信路2 [i] を介して次の番号の分散処理ノード1 [i+1] に送信する(図3ステップS108)。

[0033] 複数の分散処理ノード1 [n] ($n = 1, \dots, N$) のうち、予め定められたN番目の分散処理ノード1 [N] は、分散処理ノード1 [N-1] から中間集計データ $R_t[m, N-1]$ を自ノードの通信ポート11および通信路2 [N-1] を介して受信する(図3ステップS109, S110)。

[0034] N番目の分散処理ノード1 [N] は、受信した中間集計データ $R_t[m, N-1]$ ($m = 1, \dots, M$) と自ノードで生成された分散データ $D[m, N]$ との和を、対応する重み $w[m]$ 毎に求めることにより、中間集計データ $R_t[m, N]$ を生成する(図3ステップS111)。すなわち、中間集計データ $R_t[m, N]$ は、M個の数値から構成される。中間集計データ $R_t[m, N]$ の計算式は以下のとおりである。

$$R_t[m, N] = R_t[m, N-1] + D[m, N] \quad \dots (4)$$

[0035] そして、N番目の分散処理ノード1 [N] は、自ノードで生成された中間集計データ $R_t[m, N]$ ($m = 1, \dots, M$) を自ノードの通信ポート10および通信路2 [N] を介して1番目の分散処理ノード1 [1] に送信する(図3ステップS112)。

[0036] このように、式(2)、式(3)、式(4)により計算された、M個の数値から構成される中間集計データ $R_t[m, N]$ は、各分散処理ノード1 [n] ($n = 1, \dots, N$) で生成されたM個の数値から構成される分散データ $D[m, n]$ ($m = 1, \dots, M$) に基づいて計算される。中間集計データ $R_t[m, N]$ の値は以下の式により表すことができる。

[0037] [数2]

$$R_t[m, N] = \sum_{n=1, \dots, N} D[m, n] \quad \dots (5)$$

[0038] 次に、中間集計データ $R_t[m, N]$ ($m = 1, \dots, M$) を集計データとして、各分散処理ノード1 [n] ($n = 1, \dots, N$) に分配する分

配通信を行う。

1 番目の分散処理ノード 1 [1] は、分散処理ノード 1 [N] から中間集計データ $R_t[m, N]$ を自ノードの通信ポート 1 1 および通信路 2 [N] を介して受信する（図 3 ステップ S 1 1 3, S 1 1 4）。

[0039] 1 番目の分散処理ノード 1 [1] は、受信した中間集計データ $R_t[m, N]$ ($m = 1, \dots, M$) を、集計データ $R[m]$ として、自ノードの通信ポート 1 1 および通信路 2 [N] を介して N 番目の分散処理ノード 1 [N] に送信する（図 3 ステップ S 1 1 5）。すなわち、分散処理ノード 1 [1] は、分散処理ノード 1 [N] からの中間集計データ $R_t[m, N]$ を集計データ $R[m]$ として分散処理ノード [N] に戻すことになる。集計データ $R[m]$ は、中間集計データ $R_t[m, N]$ と同じである。

[0040] [数3]

$$R[m] = R_t[m, N] = \sum_{n=1, \dots, N} D[m, n] \quad \dots (6)$$

[0041] 続いて、複数の分散処理ノード 1 [n] ($n = 1, \dots, N$) のうち、1 番目を除く分散処理ノード 1 [k] ($k = N, \dots, 2$) は、次の番号の分散処理ノード 1 [k⁺] ($k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$) から集計データ $R[m]$ ($m = 1, \dots, M$) を自ノードの通信ポート 1 0 および通信路 2 [k] を介して受信する（図 3 ステップ S 1 1 6, S 1 1 7）。

[0042] 分散処理ノード 1 [n] ($n = 1, \dots, N$) のうち、1 番目を除く分散処理ノード 1 [k] ($k = N, \dots, 2$) は、受信した集計データ $R[m]$ ($m = 1, \dots, M$) を自ノードの通信ポート 1 1 および通信路 2 [k - 1] を介して前の番号の分散処理ノード 1 [k - 1] に送信する（図 3 ステップ S 1 1 8）。

[0043] 1 番目の分散処理ノード 1 [1] は、分散処理ノード 1 [2] から集計データ $R[m]$ ($m = 1, \dots, M$) を自ノードの通信ポート 1 0 および通信路 2 [1] を介して受信する（図 3 ステップ S 1 1 9, S 1 2 0）。

[0044] ここで、1番目の分散処理ノード1[1]が、M個の数値から構成される集計データR[m]を正常に受信するためには、他の分散処理ノード1[k] ($k = N, \dots, 2$) が集計データR[m]を正常に受信することが必要である。分散処理ノード間の各通信路2[n] ($n = 1, \dots, N$) は、正常ではない集計データR[m]を正常に戻す機能を有していない。

[0045] したがって、分散処理ノード1[1]が集計データR[m]を正常に受信した場合、全ての分散処理ノード1[n] ($n = 1, \dots, N$) が正常に集計データR[m]を受信できたことが保証される。分散処理ノード1[1]が正常に集計データR[m]を受信できなかった場合は(ステップS120においてNO)、ステップS103に戻って集約通信からやり直すようにすればよい。

[0046] なお、分散処理ノード1[1]が集計データR[m]を正常に受信できたかどうかは、例えばステップS115で送信した集計データR[m]とステップS119、S120で受信した集計データR[m]とを比較することにより、判定することができる。すなわち、送信した集計データR[m]と受信した集計データR[m]とが一致すれば、集計データR[m]を正常に受信できたと判定できる。

[0047] 以上の分配通信により、全ての分散処理ノード1[n] ($n = 1, \dots, N$) は、同一の集計データR[m]を取得することができる。

集約通信は、分散処理ノード1[1] → 分散処理ノード1[2] → $\dots$ → 分散処理ノード1[N] → 分散処理ノード1[1] という経路で行われる。分配通信は、分散処理ノード1[1] → 分散処理ノード1[N] → $\dots$ → 分散処理ノード1[2] → 分散処理ノード1[1] という経路で行われる。

[0048] つまり、集約通信と分配通信とは、互いに通信の方向が逆になる。集約通信と分配通信とは、双方向の通信を同時に行うことが可能な通信ポート10, 11と通信路2[n]とを介して行われるため、集約通信が完了するまで分配通信の開始を待つ必要がない。

- [0049] すなわち、分散処理ノード1 [1] が中間集計データ $R_t[m, 1]$ ($m = 1, \dots, M$) の送信を完了する前に、分散処理ノード1 [1] が中間集計データ $R_t[m, N]$ を受信開始した場合は、この中間集計データ $R_t[m, N]$ を集計データ $R[m]$ とした分配通信を開始できる。
- [0050] 図4は分散処理ノード1 [n] ($n = 1, \dots, N$) 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1 [n] は、集計データ $R[m]$ ($m = 1, \dots, M$) を受信すると（図4ステップS121においてYES）、受信した集計データ $R[m]$ に基づいて、自ノード内の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を行う（図4ステップS122）。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。
- [0051] 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1 [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。
- [0052] 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1 [n] ($n = 1, \dots, N$) は、更新された重み $w[m]$ 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1 [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、自ノードのニューラルネットワークの推論精度を向上させる。
- [0053] 本実施例で示したように、集約通信が完了するまで分配通信の開始を待つ必要がなく、集約通信中であっても、集計を終えたデータの一部から分配通信を開始することが可能であるため、集約通信を完了してから分配通信を開始するという従来技術と比較して、集約通信の開始から分配通信の完了までの時間を短縮することが可能であるため、より高速な深層学習の分散システ

ムを提供することが可能である。

[0054] また、本実施例では、分散処理ノード 1 [1] が集計データ $R[m]$ の取得を完了した時点で他の分散処理ノード 1 [k] ($k = 2, \dots, N$) が集計データ $R[m]$ の取得を完了したことが保証されるため、信頼性の高い深層学習の分散処理システムを提供することが可能である。

[0055] [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 の実施例をより具体的に説明するものである。図 5 は本実施例に係る分散処理ノード 1 [1] の構成例を示すブロック図、図 6 は本実施例に係る分散処理ノード 1 [k] ($k = 2, \dots, N$) の構成例を示すブロック図である。

[0056] 分散処理ノード 1 [1] は、通信ポート 10 (第 1 の通信ポート) と、通信ポート 11 (第 2 の通信ポート) と、中間集計データ $R_t[m, 1]$ ($m = 1, \dots, M$) をパケット化して自ノードの通信ポート 10 に出力する送信部 12 (第 1 の送信部) と、自ノードの通信ポート 10 から受信するパケットから集計データ $R[m]$ を取得する受信部 13 (第 2 の受信部) と、集計データ $R[m]$ をパケット化して自ノードの通信ポート 11 に出力する送信部 14 (第 2 の送信部) と、自ノードの通信ポート 11 から受信するパケットから中間集計データ $R_t[m, N]$ ($m = 1, \dots, M$) を取得する受信部 15 (第 1 の受信部) と、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 16 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 $w[m]$ 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 $G[m, 1, s]$ をサンプルデータ毎に計算する勾配計算処理部 17 と、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 $D[m, 1]$ を重み $w[m]$ 毎に生成して保持するノード内集計処理部 18 と、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を更新する重み更新処理部 20 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 21 とを備えている。

[0057] 分散処理ノード1[k] ($k = 2, \dots, N$) は、通信ポート10 (第1の通信ポート) と、通信ポート11 (第2の通信ポート) と、中間集計データ $R_t[m, k]$ ($m = 1, \dots, M$) をパケット化して自ノードの通信ポート10に出力する送信部12 (第1の送信部) と、自ノードの通信ポート10から受信するパケットから集計データ $R[m]$ を取得する受信部13 (第2の受信部) と、集計データ $R[m]$ をパケット化して自ノードの通信ポート11に出力する送信部14 (第2の送信部) と、自ノードの通信ポート11から受信するパケットから中間集計データ $R_t[m, k-1]$ ($m = 1, \dots, M$) を取得する受信部15 (第1の受信部) と、サンプル入力部16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 $w[m]$ 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 $G[m, k, s]$ をサンプルデータ毎に計算する勾配計算処理部17と、サンプルデータ毎の勾配 $G[m, k, s]$ を集計した数値である分散データ $D[m, k]$ を重み $w[m]$ 毎に生成して保持するノード内集計処理部18と、受信した中間集計データ $R_t[m, k-1]$ ($m = 1, \dots, M$) と自ノードで生成された分散データ $D[m, k]$ との和を、対応する重み $w[m]$ 毎に求めて更新後の中間集計データ $R_t[m, k]$ を生成する集計データ生成部19と、重み更新処理部20と、ニューラルネットワーク21とを備えている。

[0058] なお、分散処理ノード1[1]と分散処理ノード1[k] ($k = 2, \dots, N$) とは、後述のように同一のハードウェアで実現することが可能である。具体的には、外部からの初期設定により、各分散処理ノードの機能を、親ノード (分散処理ノード1[1]) あるいは子ノード (分散処理ノード1[k]) の何れかに指定することが可能である。これにより、本発明では、全ての分散処理ノードを低コストに実現可能である。

[0059] 図2のステップS100で説明したとおり、各分散処理ノード1[n] ($n = 1, \dots, N$) のサンプル入力部16は、データ収集ノードからサンプルデータ $x[n, s]$ ($s = 1, \dots, S$) をミニバッチ毎に入力する

。

[0060] 図2のステップS101で説明したとおり、各分散処理ノード1[n] ($n = 1, \dots, N$)の勾配計算処理部17は、サンプルデータ $x[n, s]$ ($n = 1, \dots, N$)の入力されたとき、ニューラルネットワーク21のM個の重み $w[m]$ ($m = 1, \dots, M$)の各々について、ニューラルネットワーク21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。

[0061] 図2ステップS102で説明したとおり、各分散処理ノード1[n] ($n = 1, \dots, N$)のノード内集計処理部18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 $D[m, n]$ ($m = 1, \dots, M$)を、重み $w[m]$ 毎に生成して保持する。

[0062] 次に、各分散処理ノード1[n] ($n = 1, \dots, N$)の送信部12は、外部からの初期設定により、親ノード(分散処理ノード1[1])用の送信部として動作するか、子ノード(分散処理ノード1[k]、 $k = 2, \dots, N$)用の送信部として動作するかを、設定できるようになっている。

[0063] 親ノードとして設定された分散処理ノード1[1]の送信部12は、自ノードのノード内集計処理部18によって生成されたM個の分散データ $D[m, 1]$ ($m = 1, \dots, M$)を中間集計データ $R_t[m, 1]$ として、この中間集計データ $R_t[m, 1]$ を、重み $w[m]$ の番号mの順番にパケット化し、生成した集約通信パケット $SP[p, 1]$ ($p = 1, \dots, P$ 、Pは2以上の整数)を自ノードの通信ポート10に出力する。この集約通信パケット $SP[p, 1]$ は、通信ポート10から通信路2[1]を介して次の番号の分散処理ノード1[2]に送信される(図3ステップS103、S104)。

[0064] 一方、子ノードとして設定された各分散処理ノード1[k] ($k = 2, \dots, N$)の受信部15は、分散処理ノード1[k-1]から集約通信パケット $SP[p, k-1]$ ($p = 1, \dots, P$)を自ノードの通信ポート11および通信路2[k-1]を介して受信し、受信した集約通信パケット $SP[p, k-1]$ から中間集計データ $R_t[m, k-1]$ ($m = 1, \dots$

、 M) を取得する (図3ステップS105, S106, S109, S110)。

[0065] 子ノードとして設定された各分散処理ノード1[k] ($k=2, \dots, N$) の集計データ生成部19は、自ノードの受信部15によって取得された中間集計データ $R_t[m, k-1]$ ($m=1, \dots, M$) と自ノードのノード内集計処理部18によって生成された分散データ $D[m, k]$ との和を、対応する重み $w[m]$ 毎 (番号 m 毎) に求めることにより、中間集計データ $R_t[m, k]$ を番号 m の順番に生成する (図3ステップS107, S111)。

[0066] そして、各分散処理ノード1[k] ($k=2, \dots, N$) の送信部12は、自ノードの集計データ生成部19によって生成された M 個の中間集計データ $R_t[m, k]$ ($m=1, \dots, M$) を、重み $w[m]$ の番号 m の順番にパケット化し、生成した集約通信パケット $SP[p, k]$ ($p=1, \dots, P$) を自ノードの通信ポート10に出力する。この集約通信パケット $SP[p, k]$ は、通信ポート10から通信路2[k] を介して次の番号の分散処理ノード1[k⁺] ($k^+=k+1$ 、ただし $k=N$ の場合は $k^+=1$) に送信される (図3ステップS108, S112)。

[0067] 次に、各分散処理ノード1[n] ($n=1, \dots, N$) の送信部14は、送信部12と同様に外部からの初期設定により、親ノード (分散処理ノード1[1]) 用の送信部として動作するか、子ノード (分散処理ノード1[k]、 $k=2, \dots, N$) 用の送信部として動作するかを、設定できるようになっている。

[0068] 親ノードとして設定された分散処理ノード1[1] の受信部15は、分散処理ノード1[N] から集約通信パケット $SP[p, N]$ を自ノードの通信ポート11および通信路2[N] を介して受信し、受信した集約通信パケット $SP[p, N]$ ($p=1, \dots, P$) から中間集計データ $R_t[m, N]$ ($m=1, \dots, M$) を取得する (図3ステップS113, S114)。

- [0069] 親ノードとして設定された分散処理ノード1[1]の送信部14は、自ノードの受信部15によって取得された中間集計データ $R_t[m, N]$ ($m = 1, \dots, M$)を集計データ $R[m]$ として、この集計データ $R[m]$ を、重み $w[m]$ の番号 m の順番にパケット化し、生成した分配通信パケット $DP[p, 1]$ ($p = 1, \dots, P$)を自ノードの通信ポート11に出力する。この分配通信パケット $DP[p, 1]$ は、通信ポート11から通信路2[N]を介して N 番目の分散処理ノード1[N]に送信される(図3ステップS115)。
- [0070] 一方、子ノードとして設定された各分散処理ノード1[k] ($k = 2, \dots, N$)の受信部13は、分散処理ノード1[k⁺] ($k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$)から分配通信パケット $DP[p, k^+]$ ($p = 1, \dots, P$)を自ノードの通信ポート10および通信路2[k]を介して受信し、受信した分配通信パケット $DP[p, k^+]$ から集計データ $R[m]$ ($m = 1, \dots, M$)を取得する(図3ステップS116, S117)。
- [0071] 子ノードとして設定された各分散処理ノード1[k] ($k = 2, \dots, N$)の送信部14は、受信部13によって取得された集計データ $R[m]$ ($m = 1, \dots, M$)を、重み $w[m]$ の番号 m の順番にパケット化し、生成した分配通信パケット $DP[p, k]$ ($p = 1, \dots, P$)を自ノードの通信ポート11に出力する。この分配通信パケット $DP[p, k]$ は、通信ポート11から通信路2[k-1]を介して分散処理ノード1[k-1]に送信される(図3ステップS118)。
- [0072] 親ノードとして設定された分散処理ノード1[1]の受信部13は、分散処理ノード1[2]から分配通信パケット $DP[p, 2]$ ($p = 1, \dots, P$)を自ノードの通信ポート10および通信路2[1]を介して受信し、受信した分配通信パケット $DP[p, 2]$ から集計データ $R[m]$ ($m = 1, \dots, M$)を取得する(図3ステップS119, S120)。
- [0073] なお、各分散処理ノード1[n] ($n = 1, \dots, N$)の送信部12は、 M 個の中間集計データ $R_t[m, n]$ を、重み $w[m]$ の番号 m の順番に

L (L は 1 以上 M 未満の整数) 個ずつ P 個 (P は 2 以上の整数) の集約通信パケットに振り分けて、全ての集約通信パケットを送信し終わるまで、 P 個の集約通信パケットを順番に次の番号の分散処理ノード 1 $[n^+]$ ($n^+ = n + 1$ 、ただし $n = N$ の場合は $n^+ = 1$) に送信する。すなわち、 p 番目 ($p = 1, \dots, P$) に送信する集約通信パケット $SP[p, n]$ には、 L 個の中間集計データ $R_t[r, n]$ ($r = L \times (p - 1) + l$ 、 $l = 1, \dots, L$) が格納される。

[0074] M が L で割り切れない条件では、 P 番目の集約通信パケット $SP[P, n]$ には、 $(M - L \times (P - 1))$ 個の中間集計データ $R_t[r, n]$ ($r = L \times (P - 1) + q$ 、 $q = 1, \dots, M - L \times (P - 1)$) が格納される。

[0075] P 番目の集約通信パケット $SP[P, n]$ については、 $(M - L \times (P - 1))$ 個の中間集計データ $R_t[r, n]$ の後に、 $\{L - (M - L \times (P - 1))\}$ 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 個のデータを格納するようにしてもよい。

[0076] また、各分散処理ノード 1 $[n]$ ($n = 1, \dots, N$) の送信部 14 は、 M 個の集計データ $R[m]$ ($m = 1, \dots, M$) を、重み $w[m]$ の番号 m の順番に L 個ずつ P 個の分配通信パケットに振り分けて、全ての分配通信パケットを送信し終わるまで、 P 個の分配通信パケットを順番に分散処理ノード 1 $[n^-]$ ($n^- = n - 1$ 、ただし $n = 1$ の場合は $n^- = N$) に送信する。すなわち、 p 番目 ($p = 1, \dots, P$) に送信する分配通信パケット $DP[p, n]$ には、 L 個の集計データ $R[r]$ ($r = L \times (p - 1) + l$ 、 $l = 1, \dots, L$) が格納される。

[0077] M が L で割り切れない条件では、 P 番目の分配通信パケット $DP[p, n]$ には、 $(M - L \times (P - 1))$ 個の集計データ $R[r]$ ($r = L \times (P - 1) + q$ 、 $q = 1, \dots, M - L \times (P - 1)$) が格納される。

[0078] P 番目の分配通信パケット $DP[P, n]$ については、 $(M - L \times (P - 1))$ 個の集計データ $R[r]$ の後に、 $\{L - (M - L \times (P - 1))\}$ 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 個のデータを格納するようにしてもよい。

[0079] 各分散処理ノード $1[n]$ ($n=1, \dots, N$)の重み更新処理部 20 は、自ノードの受信部 13 によって取得された集計データ $R[m]$ に基づいて、自ノード内のニューラルネットワーク 21 の重み $w[m]$ を更新する重み更新処理を行う(図4ステップ $S122$)。

[0080] 図7に、各分散処理ノード $1[n]$ ($n=1, \dots, N$)の処理の概要を示す。図8～図10に、各分散処理ノード $1[n]$ ($n=1, \dots, N$)間の中間集計データおよび集計データの通信(集約通信と分配通信)のシーケンスを示す。

[0081] なお、図9は、図8の80の部分の処理を示している。また、81は分散処理ノード $1[1]$ におけるノード間集計処理を示している。同様に、図9の90, 91, 92は分散処理ノード $1[\alpha-1]$, $1[\alpha]$, $1[\alpha+1]$ ($\alpha=3, \dots, N-1$)におけるノード間集計処理を示している。図10は、図8の82の部分の処理、すなわち分散処理ノード $1[\beta+1]$, $1[\beta]$, $1[\beta-1]$ ($\beta=N-1, \dots, 3$)の分配通信処理を示している。

[0082] 以上のように、分散処理ノード $1[1]$ を起点・終点とした分散処理ノード $1[n]$ ($n=1, \dots, N$)から分散処理ノード $1[n^+]$ ($n^+=n+1$ 、ただし $n=N$ の場合は $n^+=1$)への集約通信(中間集計データ $R_t[m, n]$ を分散処理ノード $1[n^+]$ に送信する処理)、分散処理ノード $1[k]$ ($k=2, \dots, N$)が行うノード間集計処理(受信した中間集計データ $R_t[m, k-1]$ と自ノードで生成した分散データ $D[m, k]$ とに基づいて中間集計データ $R_t[m, k]$ を計算する処理)、分散処理ノード $1[1]$ を起点・終点とした分散処理ノード $1[n]$ ($n=1, \dots, N$)から分散処理ノード $1[n^-]$ ($n^-=n-1$ 、ただし $n=1$ の場合は $n^-=N$)への分配通信(集計データ $R[m]$ を各分散処理ノード $1[n^-]$ に分配する処理)、の全ては、前記の重み $w[m]$ の番号 m の順番に行われており、番号 m を単

位としたパイプライン化が可能である。

[0083] 本実施例では、図8～図10に示すように、集約通信処理とノード間集計処理と分配通信処理とを、並行してほぼ同時に（番号mを単位としたパイプライン処理で）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と比較したとき、処理時間の大幅な短縮が可能となる。

[0084] また、N個の分散処理ノード1[n]（ $n = 1, \dots, N$ ）が同一ハードウェアを備えるノードであっても、1個の親ノード（分散処理ノード1[1]）を定めて親ノードか否かに応じた設定を各ノードに対して行うことにより、上述する集約通信処理とノード間集計処理と分配通信処理とを実現することが可能となるので、各ノードに個別に設定を要するシステムと比較して、その管理が極めて簡易になる（1個の親ノードを除く各ノードに同じ設定を行えばよい）ので、システム管理に要するコストや管理上のミスを削減できる。

[0085] 第1、第2の実施例で説明した各分散処理ノード1[n]（ $n = 1, \dots, N$ ）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。

[0086] このコンピュータの構成例を図11に示す。コンピュータは、CPU100と、記憶装置101と、インターフェース装置（以下、I/Fと略する）102とを備えている。I/F102には、例えば通信ポート10, 11を含む通信回路が接続される。CPU100は、記憶装置101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し、本発明の分散処理システムおよび分散処理方法を実現する。

産業上の利用可能性

[0087] 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

[0088] 1…分散処理ノード、2…通信路、10, 11…通信ポート、12, 14…送信部、13, 15…受信部、16…サンプル入力部、17…勾配計算処理部、18…ノード内集計処理部、19…集計データ生成部、20…重み更新処理部、21…ニューラルネットワーク。

請求の範囲

[請求項1] リング状に配置され、隣接するノードと通信路を介して互いに接続

された N 個（ N は2以上の整数）の分散処理ノードを備え、

n 番目（ $n = 1, \dots, N$ ）の分散処理ノードは、 n^+ 番目（ $n^+ = n + 1$ 、ただし $n = N$ の場合は $n^+ = 1$ ）の分散処理ノードと双方向の通信が同時に可能な第1の通信ポートと、 n^- 番目（ $n^- = n - 1$ 、ただし $n = 1$ の場合は $n^- = N$ ）の分散処理ノードと双方向の通信が可能な同時に第2の通信ポートとを備え、

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（ $m = 1, \dots, M$ ）の分散データを生成し、

N 個の分散処理ノードのうち、予め指定された1番目の分散処理ノードは、自ノードで生成された分散データを第1の集計データとして、この第1の集計データを重み $w[m]$ の番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから2番目の分散処理ノードに向けて送信し、

N 個の分散処理ノードのうち、前記1番目を除く k 番目（ $k = 2, \dots, N$ ）の分散処理ノードは、（ $k - 1$ ）番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データと自ノードで生成された分散データとの和を、対応する重み $w[m]$ 毎に求めて更新後の第1の集計データを生成し、この第1の集計データを前記番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから k^+ 番目（ $k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$ ）の分散処理ノードに向けて送信し、

前記1番目の分散処理ノードは、 N 番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データを第2の集計データとして、この第2の集計データを前記番号 m の順番にパケット化して、自ノードの前記第2の通信ポートから前記 N 番目の

分散処理ノードに向けて送信し、

前記 k 番目の分散処理ノードは、 $k+1$ 番目の分散処理ノードから自ノードの前記第 1 の通信ポートを介して受信した第 2 の集計データを前記番号 m の順番にパケット化して、自ノードの前記第 2 の通信ポートから $(k-1)$ 番目の分散処理ノードに向けて送信し、

前記 1 番目の分散処理ノードは、2 番目の分散処理ノードから自ノードの前記第 1 の通信ポートを介して第 2 の集計データを受信し、

各分散処理ノードは、受信した前記第 2 の集計データ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、

前記分散データを生成するノード内集計処理部と、

自ノードが前記 1 番目の分散処理ノードとして機能する場合に、前記第 1 の集計データを重み $w[m]$ の番号 m の順番にパケット化して、自ノードの前記第 1 の通信ポートから 2 番目の分散処理ノードに向けて送信し、自ノードが前記 k 番目の分散処理ノードとして機能する場合に、前記更新後の第 1 の集計データを前記番号 m の順番にパケット化して、自ノードの前記第 1 の通信ポートから $k+1$ 番目の分散処理ノードに向けて送信する第 1 の送信部と、

自ノードの前記第 2 の通信ポートから受信するパケットから前記第 1 の集計データを取得する第 1 の受信部と、

自ノードが前記 1 番目の分散処理ノードとして機能する場合に、前記第 2 の集計データを前記番号 m の順番にパケット化して、自ノードの前記第 2 の通信ポートから前記 N 番目の分散処理ノードに向けて送信し、自ノードが前記 k 番目の分散処理ノードとして機能する場合に、受信した前記第 2 の集計データを前記番号 m の順番にパケット化して、自ノードの前記第 2 の通信ポートから $(k-1)$ 番目の分散処理

ノードに向けて送信する第2の送信部と、

自ノードの前記第1の通信ポートから受信するパケットから前記第2の集計データを取得する第2の受信部と、

自ノードが前記k番目の分散処理ノードとして機能する場合に、前記更新後の第1の集計データを生成する集計データ生成部と、

受信した前記第2の集計データ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、

各分散処理ノードは、前記1番目の分散処理ノードが前記第2の集計データを正常に受信できなかった場合に、前記第1の集計データの送信以降の処理を再度行うことを特徴とする分散処理システム。

[請求項4] リング状に配置され、隣接するノードと通信路を介して互いに接続された N 個（ N は2以上の整数）の分散処理ノードを備え、 n 番目（ $n=1, \dots, N$ ）の分散処理ノードが、 n^+ 番目（ $n^+=n+1$ 、ただし $n=N$ の場合は $n^+=1$ ）の分散処理ノードと双方向の通信が同時に可能な第1の通信ポートと、 n^- 番目（ $n^-=n-1$ 、ただし $n=1$ の場合は $n^-=N$ ）の分散処理ノードと双方向の通信が可能な同時に第2の通信ポートとを備えたシステムにおける分散処理方法であって、

各分散処理ノードが、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（ $m=1, \dots, M$ ）の分散データを生成する第1のステップと、

N 個の分散処理ノードのうち、予め指定された1番目の分散処理ノードが、自ノードで生成された分散データを第1の集計データとして、この第1の集計データを重み $w[m]$ の番号 m の順番にパケット化して、自ノードの前記第1の通信ポートから2番目の分散処理ノードに向けて送信する第2のステップと、

N個の分散処理ノードのうち、前記1番目を除くk番目（ $k = 2, \dots, N$ ）の分散処理ノードが、（ $k - 1$ ）番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データと自ノードで生成された分散データとの和を、対応する重み $w[m]$ 毎に求めて更新後の第1の集計データを生成し、この第1の集計データを前記番号mの順番にパケット化して、自ノードの前記第1の通信ポートから k^+ 番目（ $k^+ = k + 1$ 、ただし $k = N$ の場合は $k^+ = 1$ ）の分散処理ノードに向けて送信する第3のステップと、

前記1番目の分散処理ノードが、N番目の分散処理ノードから自ノードの前記第2の通信ポートを介して受信した第1の集計データを第2の集計データとして、この第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから前記N番目の分散処理ノードに向けて送信する第4のステップと、

前記k番目の分散処理ノードが、 k^+ 番目の分散処理ノードから自ノードの前記第1の通信ポートを介して受信した第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから（ $k - 1$ ）番目の分散処理ノードに向けて送信する第5のステップと、

前記1番目の分散処理ノードが、2番目の分散処理ノードから自ノードの前記第1の通信ポートを介して第2の集計データを受信する第6のステップと、

各分散処理ノードが、受信した前記第2の集計データ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7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5]

請求項4記載の分散処理方法において、

前記第3のステップは、前記k番目の分散処理ノードが、自ノードの前記第2の通信ポートから受信するパケットから前記第1の集計データを取得するステップと、前記更新後の第1の集計データを生成す

るステップと、前記更新後の第1の集計データを前記番号mの順番にパケット化して、自ノードの前記第1の通信ポートからk+番目の分散処理ノードに向けて送信するステップとを含み、

前記第4のステップは、前記1番目の分散処理ノードが、自ノードの前記第2の通信ポートから受信するパケットから前記第1の集計データを取得するステップと、取得した第1の集計データを第2の集計データとして、この第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから前記N番目の分散処理ノードに向けて送信するステップとを含み、

前記第5のステップは、前記k番目の分散処理ノードが、自ノードの前記第1の通信ポートから受信するパケットから前記第2の集計データを取得するステップと、受信した前記第2の集計データを前記番号mの順番にパケット化して、自ノードの前記第2の通信ポートから(k-1)番目の分散処理ノードに向けて送信するステップとを含み、

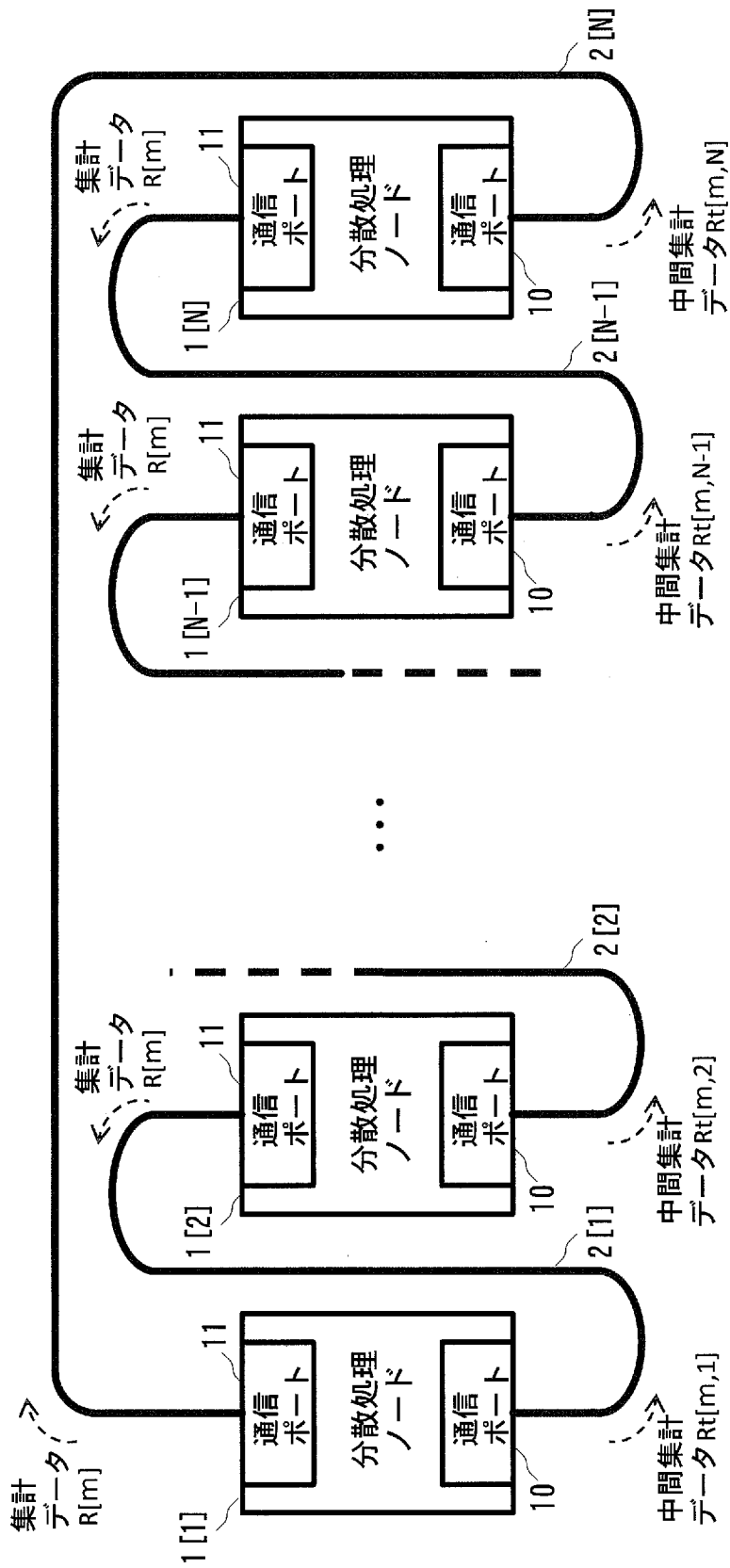
前記第6のステップは、前記1番目の分散処理ノードが、自ノードの前記第1の通信ポートから受信するパケットから前記第2の集計データを取得するステップを含むことを特徴とする分散処理方法。

[請求項6]

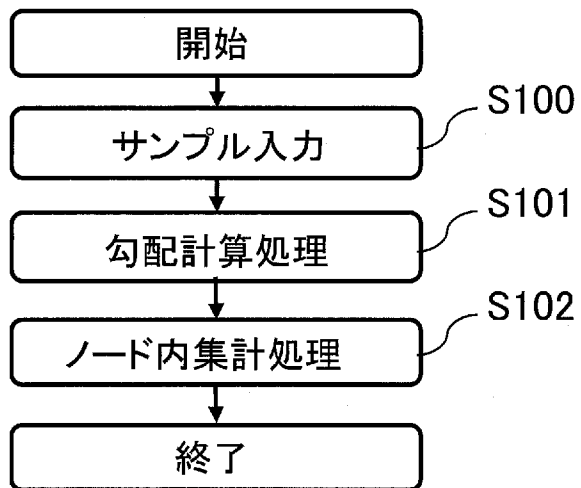
請求項4または5記載の分散処理方法において、

各分散処理ノードは、前記第6のステップにおいて前記1番目の分散処理ノードが前記第2の集計データを正常に受信できなかった場合に、前記第2のステップ以降の処理を再度行うことを特徴とする分散処理方法。

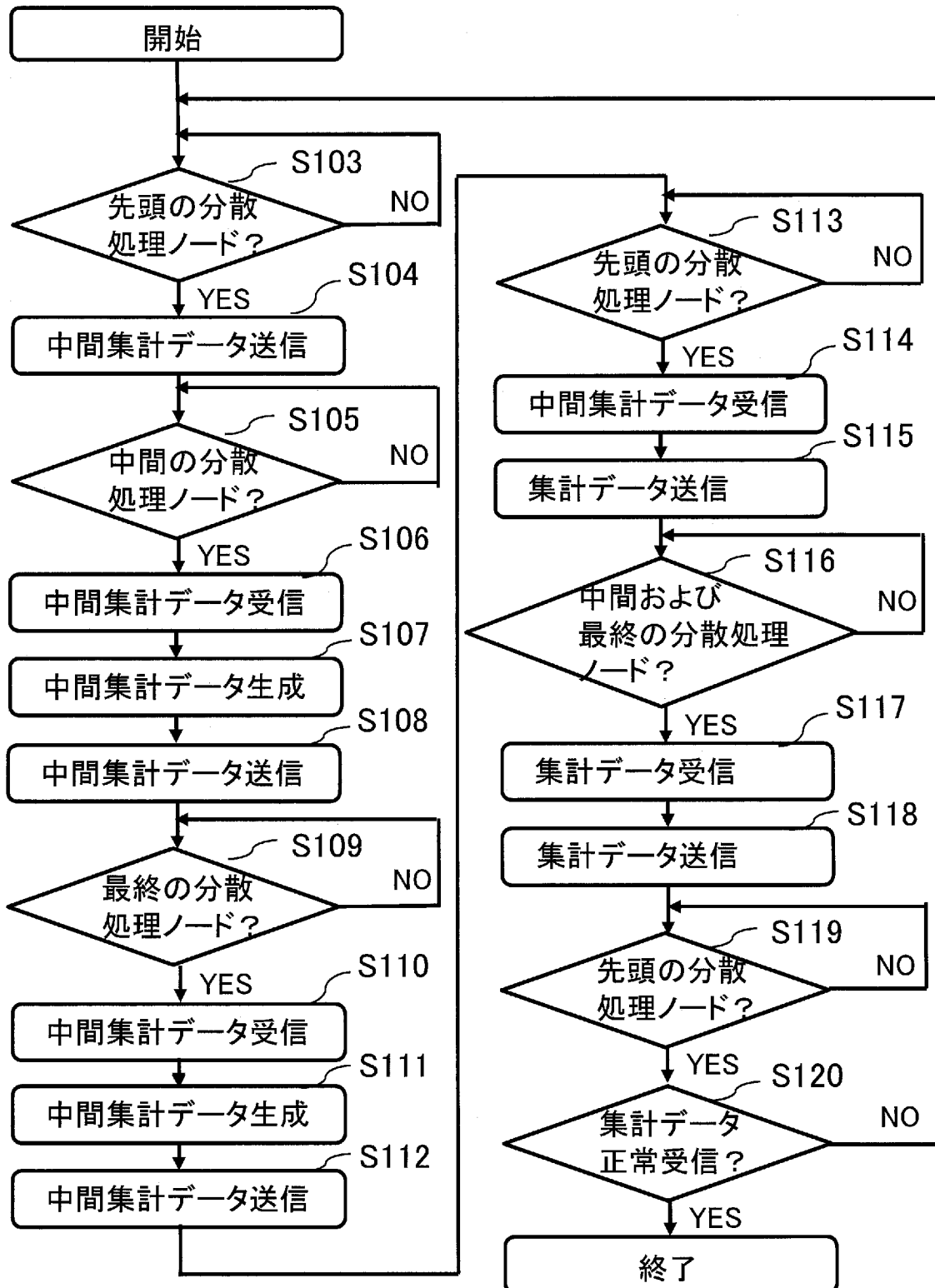
[図1]



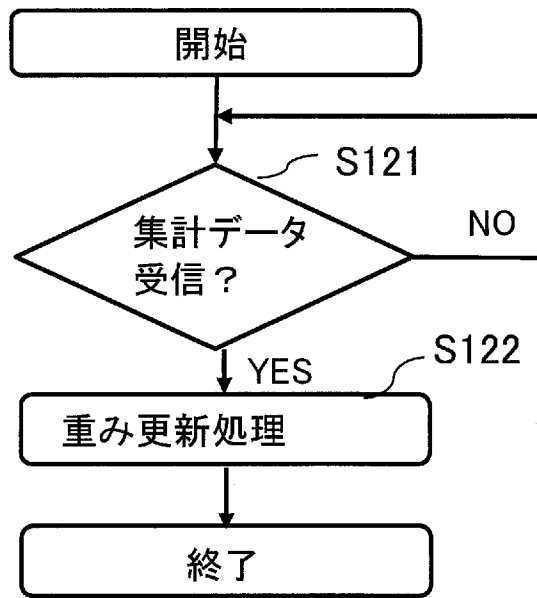
[図2]



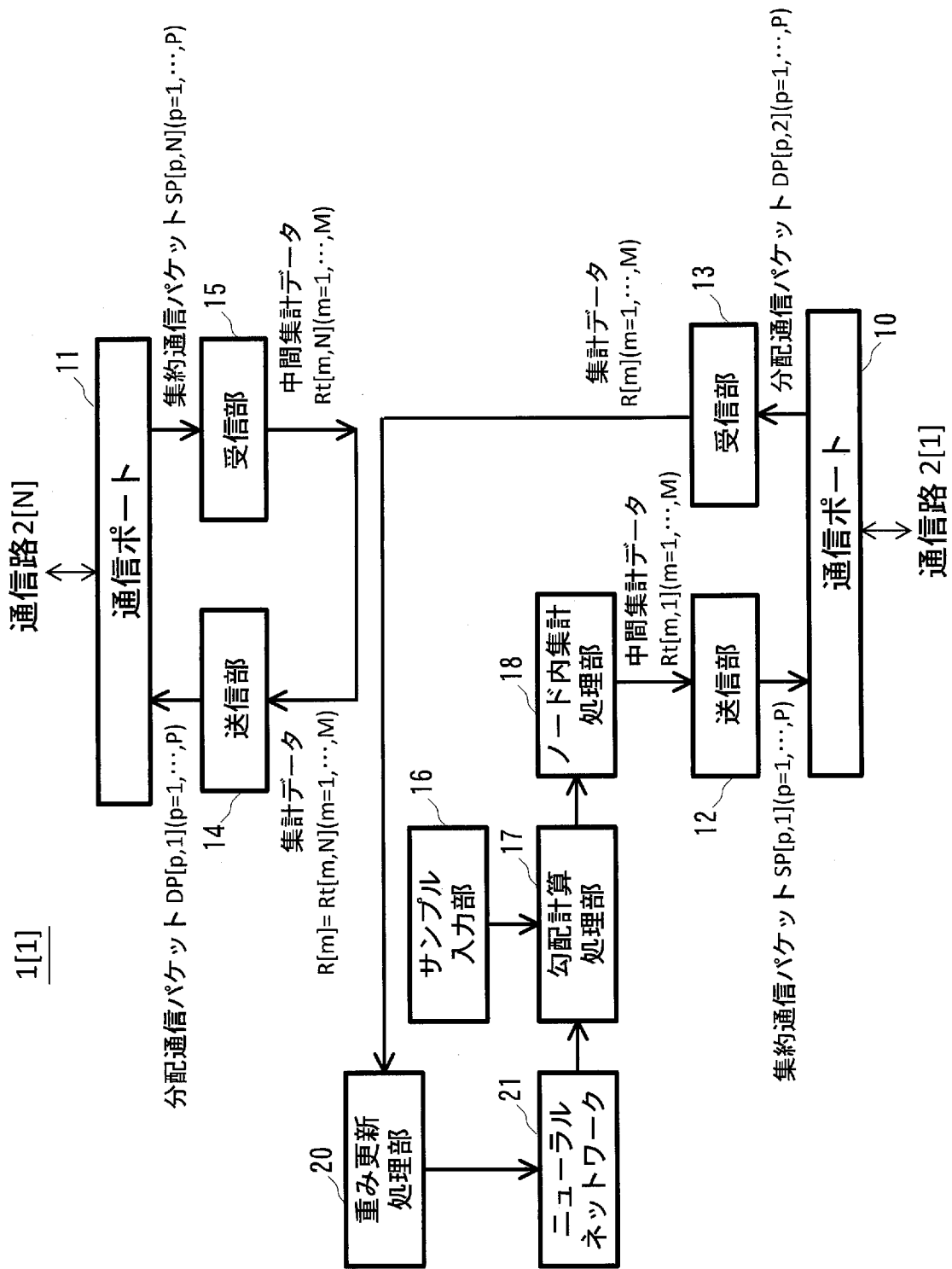
[図3]



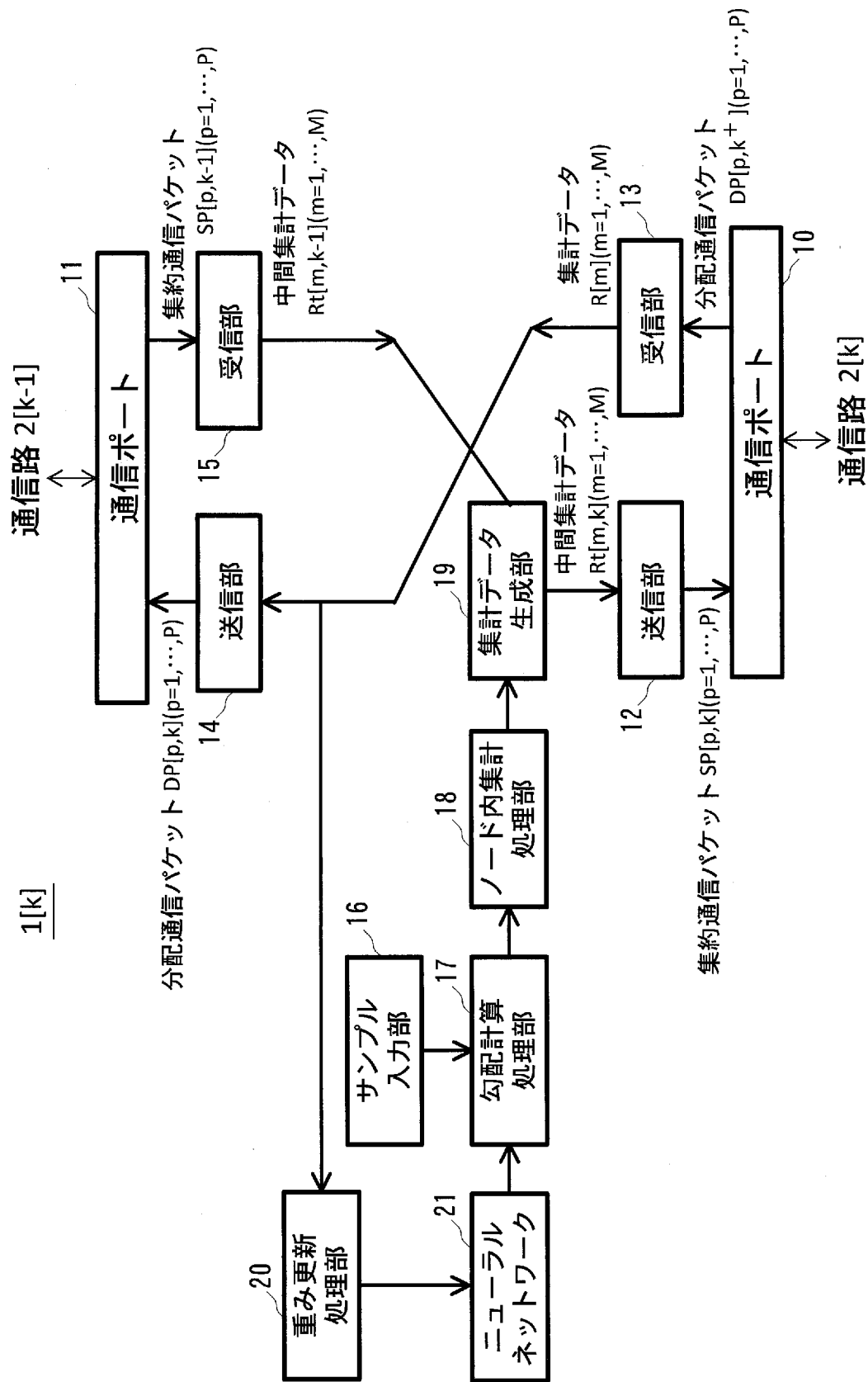
[図4]



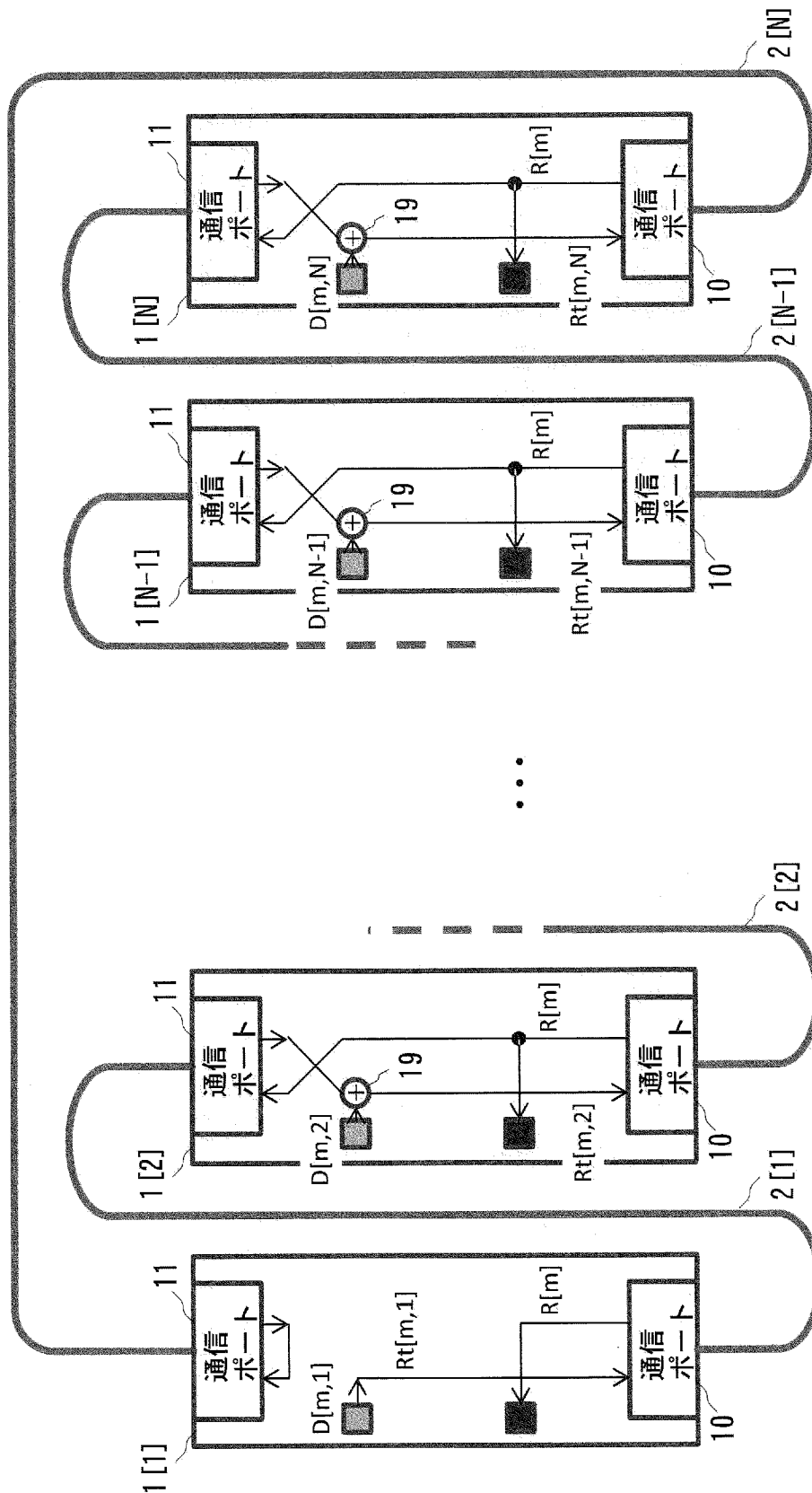
[図5]



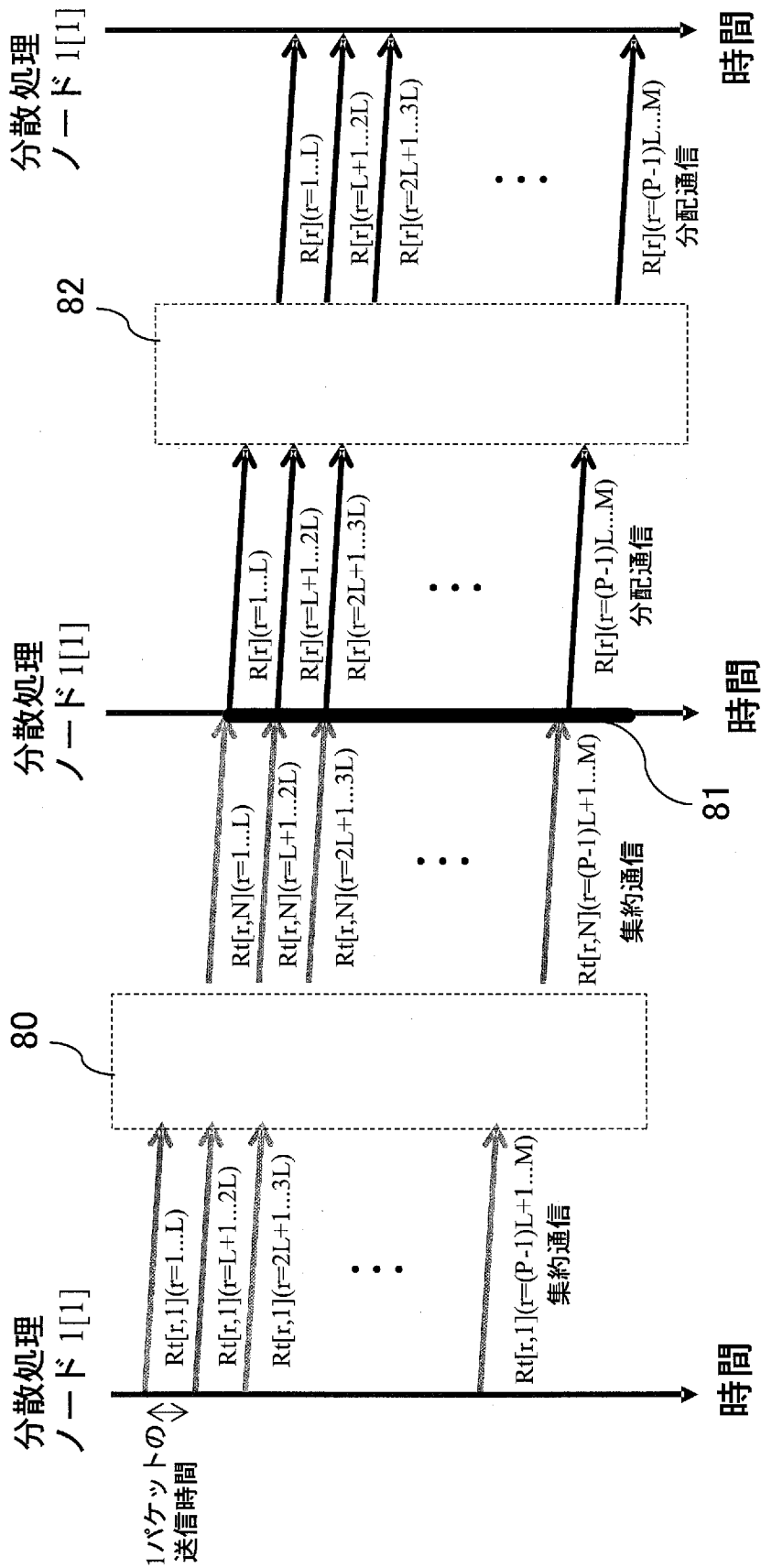
[図6]



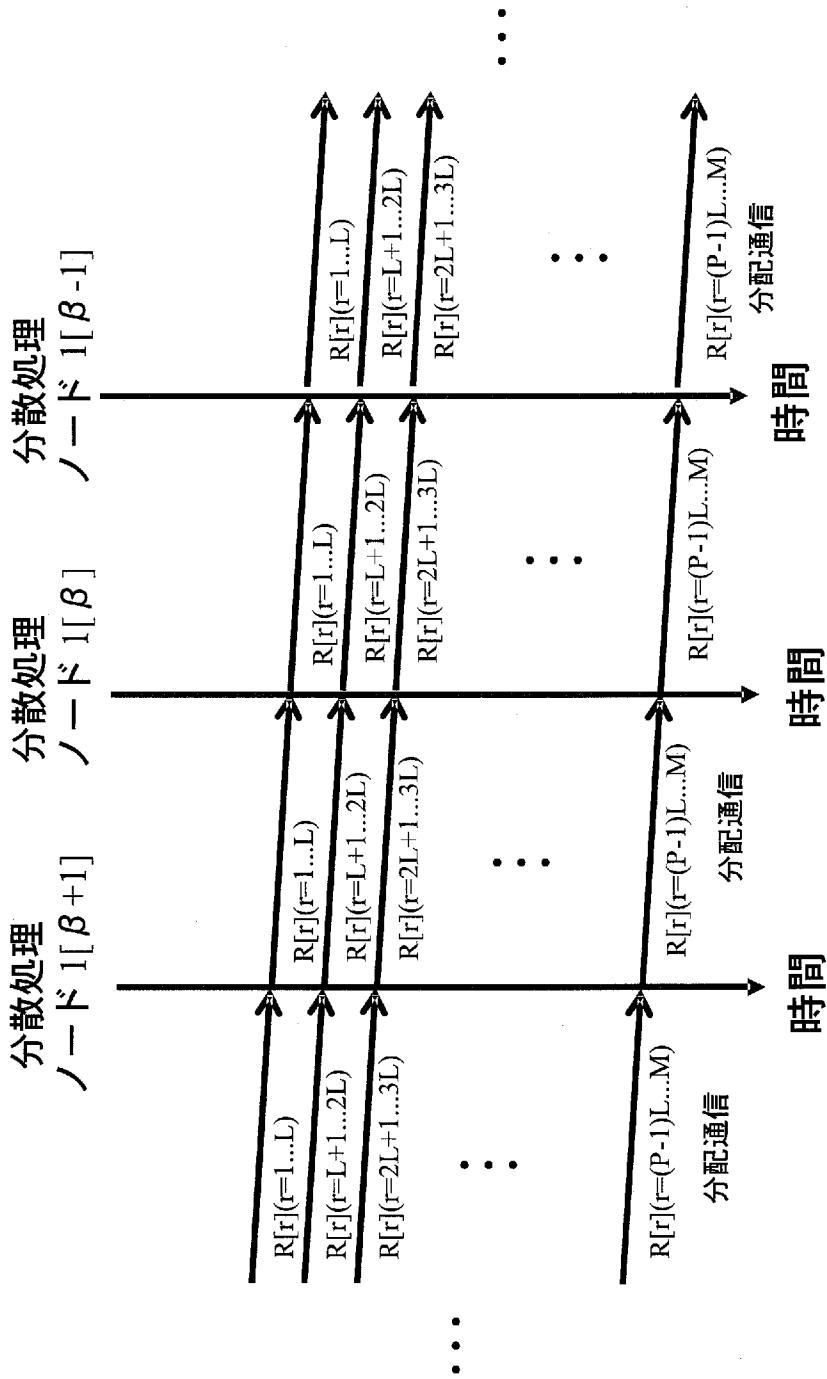
[図7]



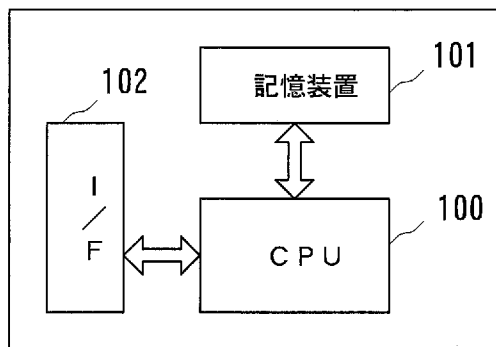
[図8]



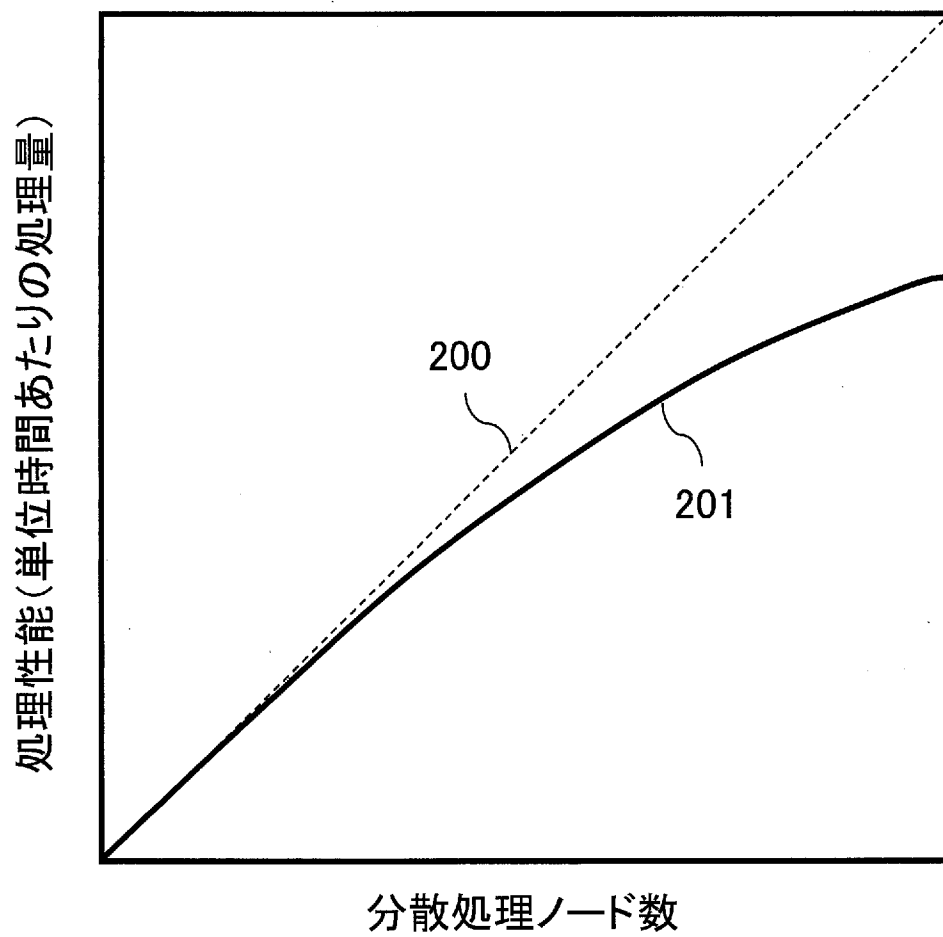
[図10]



[図11]



[図12]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2019/039449

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. G06F15/173 (2006.01) i, G06N3/08 (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. G06F15/173, G06N3/08

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.
A	SERGEEV, Alexander, DEL BALSO, Mike, "Horovod: fast and easy distributed deep learning in TensorFlow," arXiv, 21 February 2018, pp. 1-10 [online], [retrieved on 27 November 2019], Retrieved from the Internet: <URL: https://arxiv.org/abs/1802.05799v3 >	1-6

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"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)

"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

"T" 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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search
27.11.2019

Date of mailing of the international search report
10.12.2019

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

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F15/173(2006.01)i, G06N3/08(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（IPC））

Int.Cl. G06F15/173, G06N3/08

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
A	SERGEEV, Alexander, DEL BALSIO, Mike, "Horovod: fast and easy distributed deep learning in TensorFlow," arXiv, 2018.02.21, pp.1-10 [online], [retrieved on 2019.11.27], Retrieved from the Internet: <URL: https://arxiv.org/abs/1802.05799v3 >	1-6

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 | 「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 |
| 「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 | 「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 |
| 「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） | 「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 |
| 「O」 口頭による開示、使用、展示等に言及する文献 | 「&」 同一パテントファミリー文献 |
| 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願 | |

国際調査を完了した日

27.11.2019

国際調査報告の発送日

10.12.2019

国際調査機関の名称及びあて先

日本国特許庁（ISA/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）

大桃 由紀雄

5 B

5583

電話番号 03-3581-1101 内線 3545