

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6831990号
(P6831990)

(45) 発行日 令和3年2月24日(2021.2.24)

(24) 登録日 令和3年2月3日(2021.2.3)

(51) Int. Cl. F 1
 G 0 6 N 3 / 0 6 3 (2 0 0 6 . 0 1) G O 6 N 3 / 0 6 3
 G 0 6 N 3 / 0 8 (2 0 0 6 . 0 1) G O 6 N 3 / 0 8

請求項の数 8 (全 17 頁)

<p>(21) 出願番号 特願2016-156471 (P2016-156471) (22) 出願日 平成28年8月9日(2016.8.9) (65) 公開番号 特開2018-25920 (P2018-25920A) (43) 公開日 平成30年2月15日(2018.2.15) 審査請求日 令和1年6月17日(2019.6.17)</p> <p>特許法第30条第2項適用 2016年2月16日九州工業大学大学において開催された修士論文発表会、2016年2月17日九州工業大学において開催された最優秀修士論文決勝大会、2016年5月16日東京大学において開催されたLSIとシステムのワークショップ2016で発表</p>	<p>(73) 特許権者 504174135 国立大学法人九州工業大学 福岡県北九州市戸畑区仙水町1番1号</p> <p>(74) 代理人 100090697 弁理士 中前 富士男</p> <p>(74) 代理人 100176142 弁理士 清井 洋平</p> <p>(74) 代理人 100127155 弁理士 来田 義弘</p> <p>(72) 発明者 堀 三晟 福岡県北九州市若松区ひびきの2-4 国立大学法人九州工業大学内</p> <p>(72) 発明者 田向 権 福岡県北九州市若松区ひびきの2-4 国立大学法人九州工業大学内</p> <p style="text-align: right;">最終頁に続く</p>
---	--

(54) 【発明の名称】乱数生成器が不要なニューラルネットワークのハードウェア実装の方法及び乱数生成器が不要なニューラルネットワーク

(57) 【特許請求の範囲】

【請求項1】

深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークのハードウェア実装の方法であって、

最下流の前記処理層を除いた任意の処理層Aを形成する複数のユニットAのそれぞれの発火又は非発火の状態を示す状態値Aが、該処理層Aと隣り合う下流側の処理層Bを形成する複数のユニットBにそれぞれ伝達されて決まる該ユニットB毎の状態値Bを固定小数点2進数による演算で求め、該状態値Bを変数とする発火確率関数を用いて前記ユニットB毎の発火確率P(B)を求めて発火又は非発火の状態を決定して出力し、ハードウェアオブジェクトとして利用される演算回路部を設け、前記演算回路部には、前記状態値Bを求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いて、前記ユニットB毎の発火確率P(B)から該ユニットBの発火又は非発火の状態を決定するユニット状態演算器を設けることを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

【請求項2】

請求項1記載の乱数生成器が不要なニューラルネットワークのハードウェア実装の方法において、前記ニューラルネットワークは制限付きボルツマンマシンであって、前記制限付きボルツマンマシンの可視層を形成する複数の可視ユニットのそれぞれの発火又は非発火の状態を示す状態値vが、該可視層と結合する隠れ層を形成する複数の隠れユニットにそれぞれ伝達されて決まる該隠れユニット毎の状態値hを固定小数点2進数によ

る演算から求め、該状態値 h を変数とする発火確率関数を用いて前記隠れユニット毎の発火確率 $P(h)$ を求めて発火又は非発火の状態を決定して出力する順方向学習を行う第1の演算回路部を前記隠れユニット毎に対応させて設け、前記順方向学習で決定された前記隠れユニット毎の状態値 h が複数の前記可視ユニットにそれぞれ伝達されて決まる該可視ユニット毎の状態値 v を固定小数点2進数による演算から求め、該状態値 v を変数とする前記発火確率関数を用いて前記可視ユニット毎の発火確率 $P(v)$ を求めて発火又は非発火の状態を決定する逆方向学習を行う第2の演算回路部を前記可視ユニット毎に対応させて設けて、前記第1の演算回路部と前記第2の演算回路部で前記演算回路部を構成し、前記第1の演算回路部には、前記順方向学習において前記状態値 h を求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 h として用いて、前記隠れユニット毎の発火確率 $P(h)$ から前記隠れユニットの発火又は非発火の状態を決定する第1のユニット状態演算器を設け、前記第2の演算回路部には、前記逆方向学習において前記状態値 v を求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 v として用いて、前記可視ユニット毎の発火確率 $P(v)$ から前記可視ユニットの発火又は非発火の状態を決定する第2のユニット状態演算器を設けて、前記第1のユニット状態演算器と前記第2のユニット状態演算器でユニット状態演算器を構成することを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

10

【請求項3】

請求項2記載の乱数生成器が不要なニューラルネットワークのハードウェア実装の方法において、前記固定小数点2進数のビット幅を超過して切り捨てられるビットは、小数部の下位側のビット又は固定小数点2進数の整数部の上位側ビットであることを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

20

【請求項4】

請求項2又は3記載の乱数生成器が不要なニューラルネットワークのハードウェア実装の方法において、前記乱数 h が前記発火確率 $P(h)$ 以下では前記隠れユニットを発火状態とし、該乱数 h が該発火確率 $P(h)$ を超えると該隠れユニットを非発火状態とし、前記乱数 v が前記発火確率 $P(v)$ 以下では前記可視ユニットを発火状態とし、該乱数 v が該発火確率 $P(v)$ を超えると該可視ユニットを非発火状態とすることを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

30

【請求項5】

請求項2記載の乱数生成器が不要なニューラルネットワークのハードウェア実装の方法において、最初の前記順方向学習における前記可視ユニット毎の初期状態値 v_0 を該可視ユニットにそれぞれ学習値を入力して求め、前記各初期状態値 v_0 が前記隠れユニットにそれぞれ伝達されて決まる該隠れユニット毎の発火確率 $P(h_0)$ から該隠れユニット毎の発火又は非発火の状態を決定する際に用いる乱数 h_0 を、前記各初期状態値 v_0 が前記隠れユニットにそれぞれ伝達されて決まる該隠れユニット毎の状態値 h_0 を前記固定小数点2進数による演算で求める際の非切り捨てビット中の小数部の全ビットを用いて形成することを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

40

【請求項6】

請求項5記載の乱数生成器が不要なニューラルネットワークのハードウェア実装の方法において、前記乱数 h_0 が前記発火確率 $P(h_0)$ 以下では前記隠れユニットを発火状態とし、該乱数 h_0 が該発火確率 $P(h_0)$ を超えると該隠れユニットを非発火状態とすることを特徴とする乱数生成器が不要なニューラルネットワークのハードウェア実装の方法。

【請求項7】

深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークであって、最下流の前記処理層を除いた任意の処理層 A を形成する複数のユニット A のそれぞれの発火又は非発火の状態を示す状態値 A が、該処理層 A と隣り合う下流側の処理層 B を形成する複数のユニット B にそれぞれ伝達されて決まる該ユニット B 毎の状態値 B を固定小数点

50

2進数による演算で求め、該状態値Bを変数とする発火確率関数を用いて前記ユニットB毎の発火確率 $P(B)$ を求めて発火又は非発火の状態を決定して出力し、ハードウェアオブジェクトとして利用される演算回路部を有し、

前記演算回路部には、前記状態値Bを求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いて、前記ユニットB毎の発火確率 $P(B)$ から該ユニットBの発火又は非発火の状態を決定するユニット状態演算器が設けられていることを特徴とする乱数生成器が不要なニューラルネットワーク。

【請求項8】

請求項7記載の乱数生成器が不要なニューラルネットワークにおいて、前記処理層Aは制限付きボルツマンマシンの可視層、前記処理層Bは該制限付きボルツマンマシンの隠れ層であって、

前記演算回路部は、前記可視層を形成する複数の可視ユニットのそれぞれの発火又は非発火の状態を示す状態値 v が、該可視層と結合する前記隠れ層を形成する複数の隠れユニットにそれぞれ伝達されて決まる該隠れユニット毎の状態値 h を固定小数点2進数による演算から求め、該状態値 h を変数とする発火確率関数を用いて前記隠れユニット毎の発火確率 $P(h)$ を求めて発火又は非発火の状態を決定して出力する順方向学習を行う第1の演算回路部と、前記順方向学習で決定された前記隠れユニット毎の状態値 h が複数の前記可視ユニットにそれぞれ伝達されて決まる該可視ユニット毎の状態値 v を固定小数点2進数による演算から求め、該状態値 v を変数とする前記発火確率関数を用いて前記可視ユニット毎の発火確率 $P(v)$ を求めて発火又は非発火の状態を決定する逆方向学習を行う第2の演算回路部とを有し、

前記ユニット状態演算器は、前記第1の演算回路部に設けられ、前記順方向学習において前記状態値 h を求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 h として用いて、前記隠れユニット毎の発火確率 $P(h)$ から前記隠れユニットの発火又は非発火の状態を決定する第1のユニット状態演算器と、前記第2の演算回路部に設けられ、前記逆方向学習において前記状態値 v を求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 v として用いて、前記可視ユニット毎の発火確率 $P(v)$ から前記可視ユニットの発火又は非発火の状態を決定する第2のユニット状態演算器とを有していることを特徴とする乱数生成器が不要なニューラルネットワーク。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、深層学習(Deep Learning)に用いる乱数生成器が不要なニューラルネットワークのハードウェア実装の方法及び乱数生成器が不要なニューラルネットワークに関する。

【背景技術】

【0002】

制限付きボルツマンマシン(例えば、非特許文献1参照)は、近年、AI(人工知能)の分野で盛んに研究が行なわれている深層学習に用いられるニューラルネットワークの一つである。

制限付きボルツマンマシンによる学習では、学習データが可視層を構成する複数の可視ユニットに入力されて隠れ層を構成している隠れユニットに伝搬する順方向学習と、隠れ層の隠れユニットで形成される出力データが隠れ層から可視層の可視ユニットに伝搬する逆方向学習を繰り返すことにより、各可視ユニット(隠れユニット)から各隠れユニット(可視ユニット)にデータがそれぞれ伝搬する際の結合荷重(重み)を調整して、可視層に学習データが入力された際の隠れ層からの出力データを、一定誤差の範囲内で目的出力データに一致させることを目的としている。

【0003】

ここで、制限付きボルツマンマシンで構成された多層のニューラルネットワークの学習を

10

20

30

40

50

ソフトウェア上で行なう場合、順方向学習と逆方向学習の繰り返し学習を膨大な回数実行するため、大規模な計算資源と非常に長い演算時間を要することになり、消費電力量も増大するという問題がある。

そこで、制限付きボルツマンマシンによる多層のニューラルネットワークの学習を、より高速かつ低消費電力で行うことを可能にするために、制限付きボルツマンマシンのハードウェア化（チップ化）の研究が行なわれている（例えば、非特許文献2参照）。

【先行技術文献】

【非特許文献】

【0004】

【非特許文献1】Asja Fischer, Christian Igel, "An Introduction to Restricted Boltzmann Machines", Lecture Notes in Computer Science, Vol. 7441, pp. 14-36, 2012. 10

【非特許文献2】Seongwook Park, Kyeongryeol Bong, Dongjoo Shin, Jinmook Lee, Sungpil I Choi, Hoi-Jun Yoo, "A 1.93TOPS/W Scalable Deep Learning/Inference Processor with Tetra-Parallel MIMD Architecture for Big-Data Applications", 47th Annual IEEE/ACM International Symposium on Microarchitecture, 2015.

【発明の概要】

【発明が解決しようとする課題】

【0005】

制限付きボルツマンマシンをハードウェア化する場合、可視ユニットで行われる処理を実行する可視ユニット用の演算回路を可視ユニットの個数だけ、隠れユニットで行われる処理を実行する隠れユニット用の演算回路を隠れユニットの個数だけそれぞれチップ内に設ける必要がある。更に、制限付きボルツマンマシンでは、各ユニットが持つ発火確率によってランダム（乱数による）にそのユニットの発火又は非発火の状態が決定されるため、可視ユニット用の演算回路と隠れユニット用の演算回路にそれぞれ乱数生成器を設ける必要がある。 20

ここで、乱数生成器を形成する場合、ビット長の長いシフトレジスタと複数の論理ゲートが必要となるので、乱数生成器を備えた可視ユニット用と隠れユニット用の演算回路をそれぞれ必要個数だけチップ内に形成すると、チップ内で多量の回路資源（ハードウェア資源）が消費されることになる。このため、チップ内で形成できる制限付きボルツマンマシンの規模に制約が生じることになり、大規模な多層のニューラルネットワークを制限付きボルツマンマシンを用いて実現するためには、膨大なハードウェア資源を有するチップを準備しなければならないという問題が生じる。 30

【0006】

本発明はかかる事情に鑑みてなされたもので、限られたハードウェア資源の中で、ニューラルネットワークのハードウェア実装に必要なハードウェア資源を削減してハードウェア化を実現することが可能な乱数生成器が不要なニューラルネットワークのハードウェア実装の方法及び乱数生成器が不要なニューラルネットワークを提供することを目的とする。

【課題を解決するための手段】

【0007】

前記目的に沿う第1の発明に係る乱数生成器が不要なニューラルネットワークのハードウェア実装の方法は、深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークのハードウェア実装の方法であって、 40

最下流の前記処理層を除いた任意の処理層Aを形成する複数のユニットAのそれぞれの発火又は非発火の状態を示す状態値Aが、該処理層Aと隣り合う下流側の処理層Bを形成する複数のユニットBにそれぞれ伝達されて決まる該ユニットB毎の状態値Bを固定小数点2進数による演算で求め、該状態値Bを変数とする発火確率関数を用いて前記ユニットB毎の発火確率P(B)を求めて発火又は非発火の状態を決定して出力し、ハードウェアオブジェクトとして利用される演算回路部を設け、前記演算回路部には、前記状態値Bを求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いて、前記ユニットB毎の発火確率P(B)から該ユニットBの 50

発火又は非発火の状態を決定するユニット状態演算器を設ける。

【0008】

前記目的に沿う第2の発明に係る乱数生成器が不要なニューラルネットワークは、深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークであって、最下流の前記処理層を除いた任意の処理層Aを形成する複数のユニットAのそれぞれの発火又は非発火の状態を示す状態値Aが、該処理層Aと隣り合う下流側の処理層Bを形成する複数のユニットBにそれぞれ伝達されて決まる該ユニットB毎の状態値Bを固定小数点2進数による演算で求め、該状態値Bを変数とする発火確率関数を用いて前記ユニットB毎の発火確率 $P(B)$ を求めて発火又は非発火の状態を決定して出力し、ハードウェアオブジェクトとして利用される演算回路部を有し、

10

前記演算回路部には、前記状態値Bを求める際に前記固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いて、前記ユニットB毎の発火確率 $P(B)$ から該ユニットBの発火又は非発火の状態を決定するユニット状態演算器が設けられている。

【発明の効果】

【0009】

第1の発明に係る乱数生成器が不要なニューラルネットワークのハードウェア実装の方法及び第2の発明に係る乱数生成器が不要なニューラルネットワークにおいては、ニューラルネットワークの処理層Aを形成する複数のユニットAの状態を示す状態値Aから、処理層Aと隣り合う下流側の処理層Bを形成する複数のユニットBの状態を決定して出力する演算回路部をユニットB毎に対応させて設けるので、複数のユニットBの発火又は非発火の状態を、同時にかつ高速で演算して決定することができる。これにより、深層学習を短時間で行うことができ、消費電力の削減を達成することが可能になる。

20

【0010】

更に、ユニットBの状態値Bを固定小数点2進数による演算で求めるので、従来の浮動小数点2進数による演算の場合と比較して、必要な回路資源を削減することができる。また、ユニットBの発火確率 $P(B)$ からユニットBの状態を決定する場合に必要な乱数として、ユニットBの状態値Bを求める際に切り捨てるビットを用いて形成する数値を用いるので、従来必要であった乱数生成器が不要となり、回路資源の節約が可能になる。

その結果、乱数生成器が不要なニューラルネットワークでは、深層学習を行う多層のニューラルネットワークシステムのハードウェア実装をより省資源に実現することが可能になる。更に、乱数生成器が不要なニューラルネットワークでは、ハードウェア化する際の回路資源が従来の手法と比較して削減されるため、稼働時の消費電力の削減を行うことも可能になる。

30

【図面の簡単な説明】

【0011】

【図1】(A)は本発明の一実施の形態に係る乱数生成器が不要なニューラルネットワークのハードウェア実装の方法が適用されたハードウェアとソフトウェアの複合システムの説明図、(B)はハードウェアの演算回路部の説明図である。

【図2】(A)は順方向学習時の可視層と隠れ層の関係を示す説明図、(B)は順方向学習時の隠れユニットの処理内容を示す説明図である。

40

【図3】乱数を用いて隠れユニットの発火確率から隠れユニットの発火又は非発火の状態を決定する方法の説明図である。

【図4】(A)は逆方向学習時の隠れ層と可視層の関係を示す説明図、(B)は逆方向学習時の可視ユニットの処理内容を示す説明図である。

【図5】実施例1における制限付きボルツマンマシンの学習の流れ図である。

【図6】実施例1におけるエポック回数と交差エントロピー誤差の関係を示すグラフである。

【図7】実施例1における交差エントロピー誤差の最大値と固定小数点2進数の小数部のビット幅との関係を示すグラフである。

50

【図 8】実施例 2 における固定小数点 2 進数の演算により発生する切り捨てビットの説明図である。

【図 9】実施例 2 において、(A) は小数部の全ての切り捨てビットを用いて形成した数値のヒストグラム、(B) は整数部の全ての切り捨てビットを用いて形成した数値のヒストグラムである。

【図 10】実施例 2 において、(A) は学習データ入力時のデータを省いて作成した小数部の切り捨てビットを用いて形成した数値のヒストグラム、(B) は学習データ入力時のデータを省いて作成した整数部の切り捨てビットを用いて形成した数値のヒストグラムである。

【図 11】実施例 3 におけるエポック回数と交差エントロピー誤差の関係を示すグラフである。

10

【発明を実施するための形態】

【0012】

続いて、添付した図面を参照しつつ、本発明を具体化した実施の形態につき説明し、本発明の理解に供する。

本発明の一実施の形態に係る乱数生成器が不要なニューラルネットワークのハードウェア実装の方法は、深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークのハードウェア実装の方法であって、最下流の前記処理層を除いた任意の処理層 A を形成する複数のユニット A のそれぞれの発火又は非発火の状態を示す状態値 A が、処理層 A と隣り合う下流側の処理層 B を形成する複数のユニット B にそれぞれ伝達されて決まるユニット B 毎の状態値 B を固定小数点 2 進数による演算で求め、状態値 B を変数とする発火確率関数を用いてユニット B 毎の発火確率 $P(B)$ を求めて発火又は非発火の状態を決定して出力する演算回路部を設けている。更に、演算回路部には、ユニット B の状態値 B を求める際に固定小数点 2 進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いて、ユニット B 毎の発火確率 $P(B)$ からユニット B の発火又は非発火の状態を決定するユニット状態演算器を設けている。

20

以下、乱数生成器が不要な処理層を備えたニューラルネットワークの一例として乱数生成器が不要な制限付きボルツマンマシンのハードウェア実装の方法について具体的に説明する。

【0013】

30

図 1 (A) に示すように、制限付きボルツマンマシンのハードウェア実装の方法では、制限付きボルツマンマシン (以下、RBM ともいう) のソフトウェアが搭載される主計算機 10 にハードウェアモジュール 11 を接続し、RBM のソフトウェアが実行する各種処理の中で、複雑な条件分離等のハードウェア化に向かない処理はソフトウェア (主計算機 10 の CPU) に担当させ、ビット処理や信号処理等のハードウェア化に適する処理はハードウェアモジュール 11 内に形成する専用の演算回路に担当させるといったハードウェアとソフトウェアの複合システム 12 としたハードウェアの構成となっている。

【0014】

ここで、ハードウェアモジュール 11 には、例えば、FPGA (Field Programmable Gate Array) ボードを使用する。これにより、主計算機 10 とハードウェアモジュール 11 を、何らかのインターフェイス (例えば、PCI インターフェイス、PCIe インターフェイス、Ethernet インターフェイス) を用いてバス 13 を介して接続することができ、主計算機 10 の CPU 上で、ソフトウェアによる処理をソフトウェアオブジェクトとして利用するのと同じように、演算回路をハードウェアオブジェクトとして利用することが可能になる。なお、ハードウェアオブジェクトは、ハードウェアモジュール 11 上で実現されるハードウェアである演算回路部 (hw ネット) 14 及びデータ記憶部 15 (深層学習時に使用する学習値や学習結果を保存する部分) と、主計算機 10 の CPU 上で実現されるソフトウェアである入出力関数から構成される。

40

【0015】

本実施の形態に係る制限付きボルツマンマシンのハードウェア実装の方法では、図 2 (A

50

に示すように、RBMの可視層16（処理層Aに相当）を形成する複数（ n 個）の可視ユニット17（ユニットAに相当）のそれぞれの発火又は非発火の状態を示す状態値 v_i （ $i = 1, 2, \dots, n$ 、状態値Aに相当）が、可視層16と結合する隠れ層18（処理層Aと隣り合う下流側の処理層Bに相当）を形成する複数（ m 個）の隠れユニット19（ユニットBに相当）にそれぞれ伝達されて決まる隠れユニット19毎の状態値 h_j （ $j = 1, 2, \dots, m$ 、状態値Bに相当）を固定小数点2進数による演算で求め、状態値 h_j を変数とする発火確率関数 f を用いて隠れユニット19毎の発火確率 $P_j(h_j)$ （発火確率 $P(B)$ に相当）を求めて発火又は非発火の状態を決定して出力する順方向学習を行う第1の演算回路部20を隠れユニット19毎に対応させて演算回路部（ hw ネット）14内に設ける（図1（B）参照）。

10

【0016】

隠れユニット19の状態値 h_j を固定小数点2進数による演算で求めるので、従来の浮動小数点2進数による演算の場合と比較して、必要な回路資源を削減することができる。また、隠れユニット19の状態値 h_j を求める第1の演算回路部20は隠れユニット19の個数だけ存在するので、順方向学習において、各隠れユニット19の状態値 h_j を求める演算を同時にかつ高速で実行することが可能になる。そして、第1の演算回路部20に要する回路資源が削減され、演算に要する時間が短縮されるため、低消費電力の削減が可能になる。

【0017】

ここで、 i 番目の可視ユニット16が発火の状態であるとき状態値 v_i を1とし、非発火の状態であるときの状態値 v_i を0とする。

20

また、図2（B）に示すように、 j 番目の隠れユニット19の状態値 h_j は、各可視ユニット17の状態値 v_i （ $i = 1, 2, \dots, n$ ）が j 番目の隠れユニット19に伝達される際の結合荷重（重み） w_{ij} （ $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ ）と、 j 番目の隠れユニット19のバイアス値 b_j （ $j = 1, 2, \dots, m$ ）を用いて、 $b_j + w_{1j}v_1 + w_{2j}v_2 + w_{3j}v_3 + \dots + w_{nj}v_n = b_j + \sum_i w_{ij}v_i$ となる。

そして、例えば、発火確率関数 f にシグモイド関数（ロジスティック関数）を用いると、 j 番目の隠れユニット19の発火確率 $P_j(h_j)$ は、 $P_j(h_j) = 1 / (1 + \exp(-h_j/T))$ となる（ T はパラメータ）。

30

なお、発火確率関数 f には、シグモイド関数の代わりに、 \tanh 関数又は Rectified 線形関数を使用することもできる。

【0018】

図1（B）に示すように、第1の演算回路部20には、順方向学習において状態値 h_j （ $j = 1, 2, \dots, m$ ）を求める際に、固定小数点2進数のビット幅を超過するビット（整数部の上位側ビットと小数部の下位側ビット）は切り捨てられる。そこで、ユニット状態演算器として、切り捨てられるビットの中で、小数部の下位側ビットを用いて形成する数値を乱数 h として用いて（乱数の代わりに用いて）、隠れユニット19毎の発火確率 $P_j(h_j)$ から、 j 番目の隠れユニット19の発火又は非発火の状態を決定する第1のユニット状態演算器21が設けられている。乱数生成器を使用しないため、第1のユニット状態演算器21に要するハードウェア資源を削減することができ、より省資源にRBMをハードウェア実装することが可能になる。

40

【0019】

ここで、第1のユニット状態演算器21は、図3に示すように、切り捨てた小数部の下位側ビットから形成する数値が発火確率 $P_j(h_j)$ 以下の場合では、 j 番目の隠れユニット19は発火状態であると判定し、隠れユニット19から1を出力する。従って、 j 番目の隠れユニット19の状態値 h_j は1となる。

一方、切り捨てた小数部の下位側ビットから形成する数値が発火確率 $P_j(h_j)$ を超える場合では、 j 番目の隠れユニット19は非発火状態であると判定し、隠れユニット19から0を出力する。従って、 j 番目の隠れユニット19の状態値 h_j は0となる。

50

【0020】

第1の演算回路部20には、最初の順方向学習における可視ユニット17毎の初期状態値 v_{i0} ($i = 1, 2, \dots, n$) を、可視ユニット17にそれぞれ学習値(2進数のため1又は0)を入力することにより求める機能を備えたユニット初期状態演算器22が設けられている。

【0021】

ここで、初期状態値 v_{i0} は1又は0となるため、各初期状態値 v_{i0} が隠れユニット19にそれぞれ伝達された際の隠れユニット19毎の状態値 h_{j0} を求める際に切り捨てビットは発生しない。そこで、ユニット初期状態演算器22は、各初期状態値 v_{i0} が隠れユニット19にそれぞれ伝達されて決まる隠れユニット19毎の発火確率 $P_j(h_{j0})$ から隠れユニット19毎の発火又は非発火の状態を決定する際に用いる乱数 h_0 を、各初期状態値 v_{i0} が隠れユニット19にそれぞれ伝達されて決まる隠れユニット19毎の状態値 h_0 を固定小数点2進数による演算で求める際、非切り捨てビット中の小数部の全ビットを用いて形成する数値とする機能を更に備えている。

そして、乱数 h_0 が発火確率 $P_j(h_0)$ 以下では隠れユニット19を発火状態とし、乱数 h_0 が発火確率 $P_j(h_0)$ を超えると隠れユニット19を非発火状態とする。

【0022】

また、制限付きボルツマンマシンのハードウェア実装の方法では、図4(A)に示すように、順方向学習で決定された隠れユニット19毎の状態値 h_j ($j = 1, 2, \dots, m$) が各可視ユニット17にそれぞれ伝達されて決まる可視ユニット17毎の状態値 v_i ($i = 1, 2, \dots, n$) を固定小数点2進数による演算で求め、状態値 v_i を変数とする発火確率関数 f を用いて可視ユニット17毎の発火確率 $P_i(v_i)$ を求めて発火又は非発火の状態を決定する逆方向学習を行う第2の演算回路部23を可視ユニット17毎に対応させて演算回路部(hwネット)14内に設ける(図1(B)参照)。

【0023】

なお、図4(A)に示すように、逆方向学習では処理の方向が隠れ層18から可視層16に向かうため、隠れ層18が処理層Aに相当し、可視層16が処理層Aと隣り合う下流側の処理層Bに相当する。同様に、隠れユニット19がユニットAに、可視ユニット17がユニットBに相当する。また、発火確率 $P(v_i)$ が発火確率 $P(B)$ に相当する。

【0024】

可視ユニット17の状態値 v_i を固定小数点2進数による演算で求めるので、従来の浮動小数点2進数による演算の場合と比較して、必要な回路資源を削減することができる。また、可視ユニット17の状態値 v_i を求める第2の演算回路部23は可視ユニット17の個数だけ存在するので、逆方向学習において、各可視ユニット17の状態値 v_i を求める演算を同時にかつ高速で実行することが可能になる。そして、第2の演算回路部23に要する回路資源が削減され、演算に要する時間が短縮されるため、低消費電力の削減が可能になる。

【0025】

ここで、 j 番目の隠れユニット19が発火の状態であるとき状態値 h_j を1とし、非発火の状態であるときは状態値 h_j を0とする。

また、図4(B)に示すように、 i 番目の可視ユニット17の状態値 v_i は、各隠れユニット19の状態値 h_j ($j = 1, 2, \dots, m$) が i 番目の可視ユニット17に伝達される際の結合荷重(重み) w_{ij} ($i = 1, 2, \dots, n, j = 1, 2, \dots, m$) と、 i 番目の可視ユニット17のバイアス値 a_i ($i = 1, 2, \dots, n$) を用いて、 $a_i + w_{i1}h_1 + w_{i2}h_2 + w_{i3}h_3 + \dots + w_{im}h_m = a_i + \sum_j w_{ij}h_j$ となる。

そして、発火確率関数 f にシグモイド関数を用いると、 i 番目の可視ユニット17の発火確率 $P_i(v_j)$ は、 $P_i = 1 / (1 + \exp(-v_j / T))$ となる。

【0026】

図1(B)に示すように、第2の演算回路部23には、逆方向学習において状態値 v_i (

10

20

30

40

50

$i = 1, 2, \dots, n$) を求める際に、固定小数点2進数のビット幅を超過するビット(整数部の上位側ビットと小数部の下位側ビット)は切り捨てられる。そこで、ユニット状態演算器として、切り捨てられるビットの中で、小数部の下位側ビットを用いて形成する数値を乱数 v として用いて(乱数の代わりに用いて)、可視ユニット17毎の発火確率 $P_i(v_i)$ から、 i 番目の可視ユニット17の発火又は非発火の状態を決定する第2のユニット状態演算器24が設けられている。乱数生成器を使用しないため、第2のユニット状態演算器24に要するハードウェア資源を削減することができ、より省資源にRBMをハードウェア実装することが可能になる。

【0027】

ここで、第2のユニット状態演算器24には、図3に示すように、切り捨てた小数部の下位側ビットから形成する数値が発火確率 $P_i(v_i)$ 以下の場合では、 i 番目の可視ユニット17は発火状態であると判定し、可視ユニット17から1を出力する。従って、 i 番目の可視ユニット17の状態値 v_i は1となる。

一方、切り捨てた小数部の下位側ビットから形成する数値が発火確率 $P_i(v_i)$ を超える場合では、 i 番目の可視ユニット17は非発火状態であると判定し、可視ユニット17から0を出力する。従って、 i 番目の可視ユニット17の状態値 v_i は0となる。

【0028】

本発明の一実施の形態に係る乱数生成器が不要なニューラルネットワークは、深層学習に用いる乱数生成器が不要な処理層を備えたニューラルネットワークの一例である制限付きボルツマンマシン(図2、図4参照)であって、図1(B)に示すように、順方向学習を行う、即ち、上流側の可視層16(処理層Aに相当)を形成する複数の可視ユニット17(ユニットAに相当)のそれぞれの発火又は非発火の状態を示す状態値 v (状態値Aに相当)が、可視層16と結合する隠れ層18(処理層Aと隣り合う下流側の処理層Bに相当)を形成する複数の隠れユニット19(ユニットBに相当)にそれぞれ伝達されて決まる隠れユニット19毎の状態値 h (状態値Bに相当)を固定小数点2進数による演算から求め、状態値 h を変数とする発火確率関数 f を用いて隠れユニット19毎の発火確率 $P(h)$ (発火確率 $P(B)$ に相当)を求めて発火又は非発火の状態を決定して出力する第1の演算回路部20を演算回路部14に有している。

【0029】

更に、制限付きボルツマンマシンは、図1(B)に示すように、逆方向学習を行う、即ち、順方向学習で決定された隠れ層18(データの処理方向が逆転するため処理層Aに相当、以下同様)の隠れユニット19(ユニットAに相当)毎の状態値 h (状態値Aに相当)が、可視層16(処理層Bに相当)を形成する複数の可視ユニット17(ユニットBに相当)にそれぞれ伝達されて決まる可視ユニット毎の状態値 v (状態値Bに相当)を固定小数点2進数による演算から求め、状態値 v を変数とする発火確率関数 f を用いて可視ユニット17毎の発火確率 $P(v)$ (発火確率 $P(B)$ に相当)を求めて発火又は非発火の状態を決定する第2の演算回路部23を演算回路部14に有している。

【0030】

第1の演算回路部20を設けることにより、順方向学習において、各隠れユニット19の状態値 h を求める演算を同時にかつ高速で実行することが可能になる。そして、第1の演算回路部20では固定小数点2進数による演算が実行されるため、第1の演算回路部20に要する回路資源が削減され、演算に要する時間が短縮されるため、消費電力の削減が可能になる。

また、第2の演算回路部23を設けることにより、逆方向学習において、各可視ユニット17の状態値 v を求める演算を同時にかつ高速で実行することが可能になる。そして、第2の演算回路部23では固定小数点2進数による演算が実行されるため、第2の演算回路部23に要する回路資源が削減され、演算に要する時間が短縮されるため、低消費電力の削減が可能になる。

【0031】

そして、第1の演算回路部20には、順方向学習において、複数の可視ユニット17の各

10

20

30

40

50

状態値 v が隠れ層 18 の複数の隠れユニット 19 にそれぞれ伝達されて決まる隠れユニット 19 毎の状態値 h を求める際に固定小数点 2 進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 h として用いて、隠れユニット 19 毎の発火確率 $P(h)$ から可視ユニット 19 の発火又は非発火の状態を決定する第 1 のユニット状態演算器 21 が設けられている。

【0032】

また、第 2 の演算回路部 23 には、逆方向学習において、複数の隠れユニット 19 の各状態値 h が可視層 16 の複数の可視ユニット 17 にそれぞれ伝達されて決まる可視ユニット 17 毎の状態値 v を求める際に固定小数点 2 進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数 v として用いて、可視ユニット 17 毎の発火確率 $P(v)$ から可視ユニット 17 の発火又は非発火の状態を決定する第 2 のユニット状態演算器 24 が設けられている。

10

【0033】

従来の乱数生成器を使用しないため、第 1、第 2 のユニット状態演算器 21、24 に要するハードウェア資源を削減することができ、より省資源に制限付きボルツマンマシンをハードウェア実装することが可能になる。

【0034】

ここで、順方向学習を開始した際、可視ユニット 17 毎の初期状態値は 1 又は 0 となるため、各初期状態値が隠れユニット 19 にそれぞれ伝達された際の隠れユニット 19 毎の状態値を求める際に切り捨てビットは発生しない。このため、図 1 (B) に示すように、第 1 の演算回路部 20 には、可視ユニット 17 にそれぞれ学習値 (2 進数のため 1 又は 0) を入力することにより最初の順方向学習における可視ユニット 17 毎の初期状態値を求めるユニット初期状態演算器 22 が設けられている。ユニット初期状態演算器 22 では、可視ユニット 17 の各初期状態値が隠れユニット 19 にそれぞれ伝達されて決まる隠れユニット 19 毎の発火確率から隠れユニット 19 毎の発火又は非発火の状態を決定する際に乱数として用いる数値を、各初期状態値が隠れユニット 19 にそれぞれ伝達されて決まる隠れユニット 19 毎の状態値を固定小数点 2 進数による演算で求める際の非切り捨てビット中の小数部の全ビットを用いて形成している。

20

【実施例】

【0035】

(実施例 1)

固定小数点 2 進数による演算において、整数部のビット幅を 8 ビット、小数部のビット幅を 4 ビット、6 ビット、8 ビット、10 ビット、及び 12 ビットの 5 段階に設定して、小数部のビット幅の影響を検討した。

RBM (可視層ノード数が 1024、隠れ層ノード数が 16) に、S I D B A (S t a n d a r d I m a g e D a t a - B A s e) から選んだ数枚の画像 (画像サイズは、縦 32 ピクセル × 横 32 ピクセルの 1024 ピクセル) を学習画像に用いて学習させた。学習の流れを図 5 に示す。

30

【0036】

図 5 に示すように、RBM は全ての学習画像を取り込み二値画像に変換する。そして、学習画像毎に、二値画像の各ピクセルの 1 又は 0 の値を可視層の各可視ユニットに入力し (可視層の各可視ユニットは、学習画像の各ピクセルの値にそれぞれ対応した状態となる)、可視層から隠れ層へ向かう順方向学習と、隠れ層から可視層へ向かう逆方向学習を 1 回繰り返す事前学習を行ってパラメータの初期値を求めた。次いで、全ての学習画像を用いて、可視層から隠れ層へ向かう順方向学習と隠れ層から可視層へ向かう逆方向学習の連携学習 (1 エポックという) を 800 回繰り返す全体学習を行ってパラメータを最適値に収束させた。

40

【0037】

ここで、事前学習と全体学習では、最初の順方向学習において、複数の可視ユニットのそれぞれの状態を示す状態値を、可視ユニットにそれぞれ学習画像の各ピクセルの値 (学習

50

値)を入力して得られる初期状態値とした。また、隠れユニット毎の発火又は非発火の状態を決定する際に用いる乱数を、可視ユニット毎の初期状態値が複数の隠れユニットにそれぞれ伝達されて決まる隠れユニット毎の状態値を用いて形成した数値とした。

【0038】

学習中に学習画像を想起した結果の画像と学習画像との間の交差エントロピー誤差の値を観測した。なお、計算時の値が2進数で表現できる値の最大値もしくは最小値を超過した場合は、表現可能な一定の値に固定した。また、小数点以下の数値でビット精度上表現できない値は0とした。交差エントロピー誤差のエポック毎の変化を図6に示す。なお、小数部のビット幅が4ビットの場合、途中から交差エントロピー誤差の値が発散したので、発散したエポック数の範囲以降では交差エントロピー誤差の値を記載していない。

10

【0039】

図6から求めた交差エントロピー誤差の最大値と小数部のビット幅との関係を図7に示す。図7から、小数部のビット幅が大きくなるほど、交差エントロピー誤差の値が大きくなって(最大値0に近づき)、結果が良くなる傾向を示すことが分かる。その結果、小数部のビット幅は少なくとも8に設定することが好ましいと考えられる。

【0040】

(実施例2)

固定小数点2進数のビット幅を超過して切り捨てられるビットを用いて形成する数値を乱数として用いることの妥当性について検討した。

SIDBAから選んだ3枚の学習画像(画像サイズは、縦32ピクセル×横32ピクセルの1024ピクセル)を用いて、RBM(可視層ノード数を1024、隠れ層ノード数を16、結合荷重と状態値の固定小数点2進数による整数部のビット幅を8ビット、小数部のビット幅を8ビットに設定)に、順方向学習と逆方向学習の2つの学習フェーズを5回繰り返しながらパラメータの更新を行う事前学習を行った。

20

【0041】

結合荷重と状態値の固定小数点2進数では、整数部と小数部がともに8ビットであるため、図8に示すように、結合荷重と状態値との積は、整数部と小数部がそれぞれ8ビットの値の乗算となるから、その結果は整数部16ビット、小数部16ビットとなる。続いて、状態値と重みの積の値をすべての可視ユニット数だけ加算する演算では、可視ノード数は1024個であるから、その加算後のビットの増分は10ビット($\log_2 1024 = \log_2 2^{10}$)となるため、加算後の値のビット幅は、整数部26ビット、小数部16ビットとなる。ここで、増加したビット幅を、元の整数部8ビットと小数部8ビットに戻すためには、整数部26ビットの中で上位側の18ビットと、小数部16ビットの中で下位側の8ビットをそれぞれ切り捨てることになる。

30

【0042】

5回目の事前学習を行った時に切り捨てた小数部の下位側の8ビット分から形成される数値(10進数表示)のヒストグラムを図9(A)に、整数部の上位側の18ビット分から形成される数値(10進数表示)のヒストグラムを図9(B)に示す。なお、データ数はどちらも153600個である。

図9(A)に示すように、小数部の下位側の8ビット分から形成される数値は0が極めて多く出ていることが分かる。これは、最初の順方向学習では、学習データ入力時だけ可視ユニットの状態値が入力された学習画像のピクセル値である1又は0の二値のいずれかに固定されて、切り捨てビットとなる小数点9ビット以下の値が出現しないからである。また、図9(B)に示すように、整数部の上位側の18ビット分から形成される数値は最大値もしくは最小値のみ出現している。

40

【0043】

そこで、学習データ入力時のデータ(初期値)を省いて作成した小数部の下位側の8ビット分から形成される数値のヒストグラムを図10(A)に、整数部の上位側の18ビット分から形成される数値のヒストグラムを図10(B)に示す。図10(A)に示すように、小数部の切り捨てビットから形成される数値に関しては、各数値が略同等の頻度で出現

50

しており、ある程度の白色性を有していることが確認できた。一方、整数部の切り捨てビットから形成される数値に関しては最小値又は最大値のみであった。

その結果、隠れユニット毎の状態値 h を固定小数点 2 進数の演算で求める際に、固定小数点 2 進数のビット幅を超過して切り捨てられる小数部の下位側のビットから形成される数値を乱数として用いることが可能であることが確認できた。

【 0 0 4 4 】

(実施例 3)

学習を開始した最初のステップ、即ち、最初の順方向学習においては、各初期状態値 v_0 が隠れユニットにそれぞれ伝達されて決まる隠れユニット毎の発火確率 $P(h_0)$ から隠れユニット毎の発火又は非発火の状態を決定する際に用いる乱数 h_0 を、各初期状態値 v_0 が隠れユニットにそれぞれ伝達されて決まる隠れユニット毎の状態値 h_0 を固定小数点 2 進数による演算で求める際の非切り捨てビット中の小数部の全ビットを用いて形成する数値としている。

そこで、隠れユニット毎の状態値 h_0 を固定小数点 2 進数による演算で求める際の非切り捨てビット中の小数部の 8 ビット全てを用いて形成される数値を乱数として用いることの妥当性を検討するため、実施例 2 で使用したのと同構成の R B M に、実施例 1 で用いた学習画像による 1 回の事前学習によりパラメータの初期値を求めた後、800 回エポックの全体学習を行い、1 エポック毎に交差エントロピー誤差を求めた。その結果を実施例 3 として図 1 1 に示す。

【 0 0 4 5 】

図 1 1 には、比較例 1 として、実施例 3 と同一構成の R B M に、隠れユニットの発火又は非発火の状態を決定する際に使用する乱数を全てソフトウェア上の乱数生成器で求めながら、1 回の事前学習によりパラメータの初期値を求めた後、800 回エポックの全体学習を行い、1 エポック毎に交差エントロピー誤差を求めた結果を示す。

更に、図 1 1 には、比較例 2 として、可視ユニットの初期状態値 v_0 が伝達された際の隠れユニットの発火又は非発火の状態を決定する際にだけに使用する乱数を、ソフトウェア上の乱数生成器で求めながら、1 回の事前学習によりパラメータの初期値を求めた後、800 回エポックの全体学習を行い、1 エポック毎に交差エントロピー誤差を求めた結果を、比較例 3 として、隠れユニット毎の状態値 h_0 を固定小数点 2 進数による演算で求める際の非切り捨てビット中の小数部の下位側の 4 ビットを用いて形成される数値を乱数として用いながら、1 回の事前学習によりパラメータの初期値を求めた後、800 回エポックの全体学習を行い、1 エポック毎に交差エントロピー誤差を求めた結果をそれぞれ示す。

【 0 0 4 6 】

図 1 1 から、実施例 3 の交差エントロピー誤差の変化は、全ての乱数をソフトウェア上の乱数生成器で求めた比較例 1 の交差エントロピー誤差の変化と同等の結果を示している。これにより、最初の順方向学習において、隠れユニット毎の発火又は非発火の状態を決定する際に用いる乱数として、可視ユニットの初期状態値 v_0 が隠れユニット毎に伝達されて決まる隠れユニット毎の状態値 h_0 を固定小数点 2 進数による演算で求める際の非切り捨てビット中の小数部の 8 ビット全てを用いて形成される数値を用いることの妥当性が確認できた。

【 0 0 4 7 】

以上、本発明を、実施の形態を参照して説明してきたが、本発明は何ら上記した実施の形態に記載した構成に限定されるものではなく、特許請求の範囲に記載されている事項の範囲内で考えられるその他の実施の形態や変形例も含むものである。

更に、本実施の形態とその他の実施の形態や変形例にそれぞれ含まれる構成要素を組合わせたものも、本発明に含まれる。

例えば、本実施の形態及び実施例では、固定小数点 2 進数のビット幅を超過して切り捨てられるビットを用いて形成する数値として、固定小数点 2 進数の小数部の下位側のビットから形成した数値を用いたが、固定小数点 2 進数の整数部の上位側ビットから形成した数値を用いることもできる。更に、固定小数点 2 進数の小数部の下位側のビットから形成さ

10

20

30

40

50

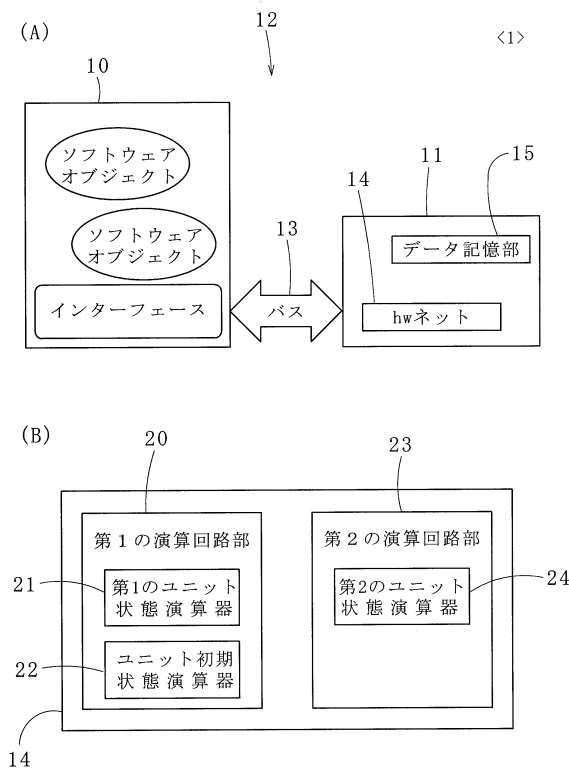
れる数値と整数部の上位側ビットから形成される数値を組合せて用いることもできる。

【符号の説明】

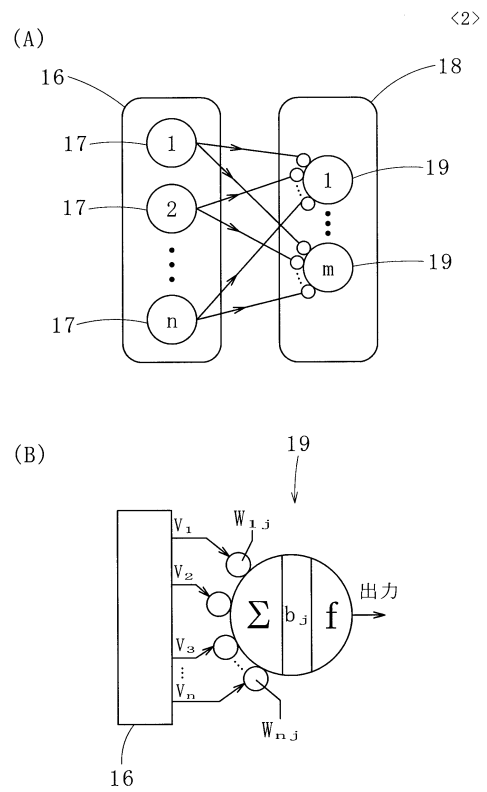
【0048】

10：主計算機、11：ハードウェアモジュール、12：ハードウェアとソフトウェアの複合システム、13：バス、14：演算回路部（hwネット）、15：データ記憶部、16：可視層、17：可視ユニット、18：隠れ層、19：隠れユニット、20：第1の演算回路部、21：第1のユニット状態演算器、22：ユニット初期状態演算器、23：第2の演算回路部、24：第2のユニット状態演算器

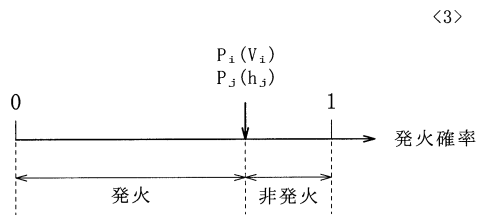
【図1】



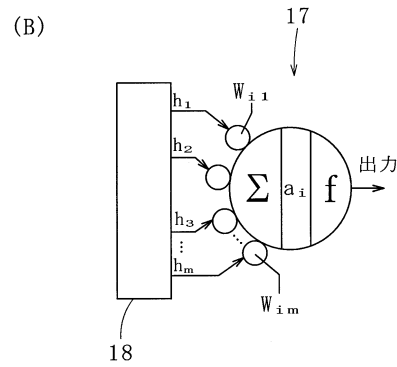
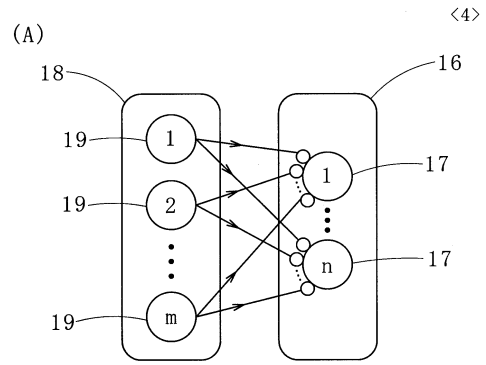
【図2】



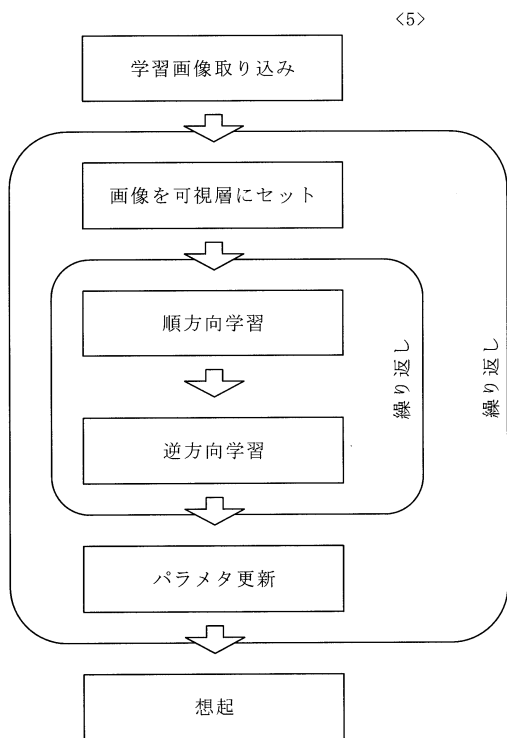
【図3】



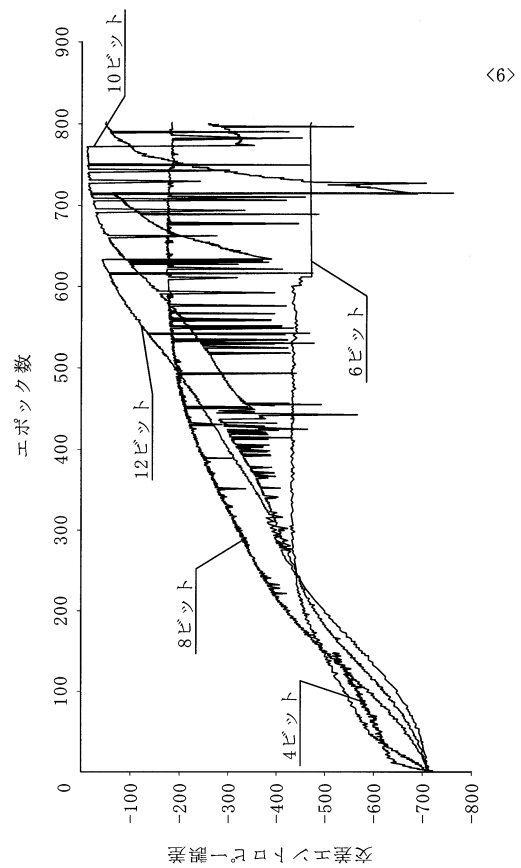
【図4】



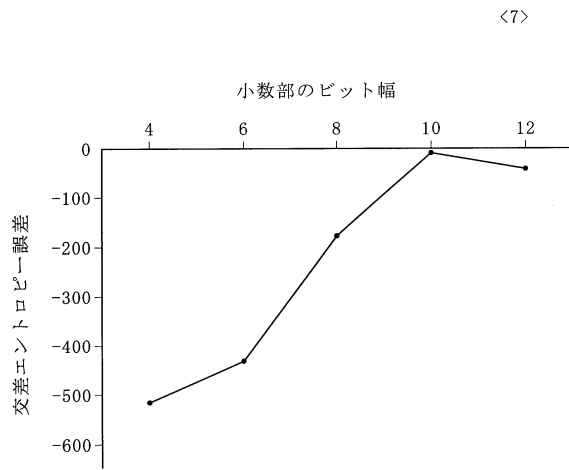
【図5】



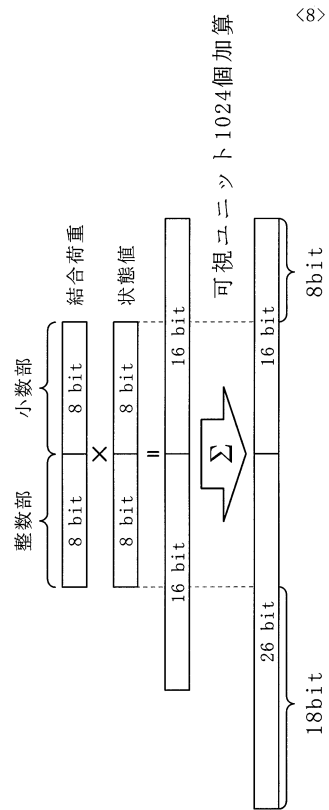
【図6】



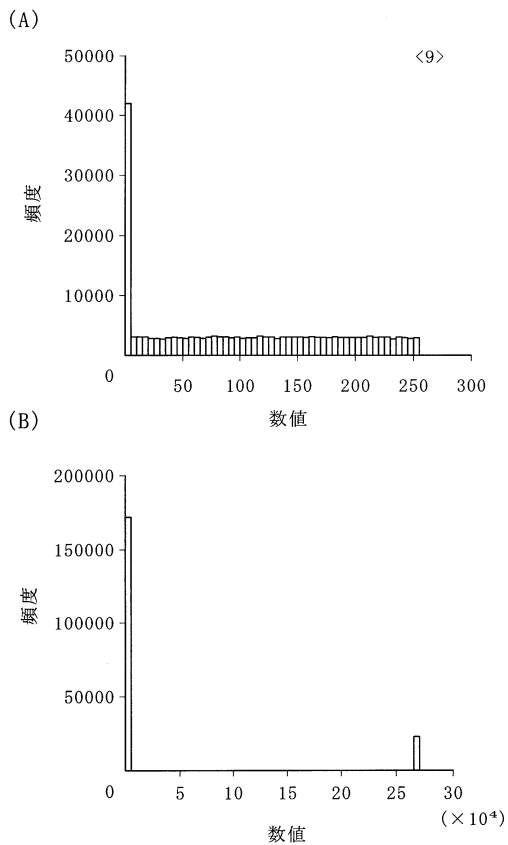
【図7】



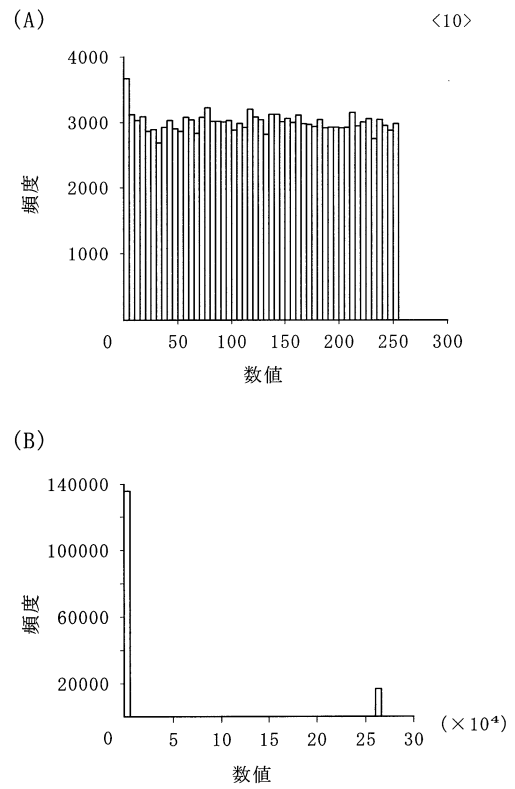
【図8】



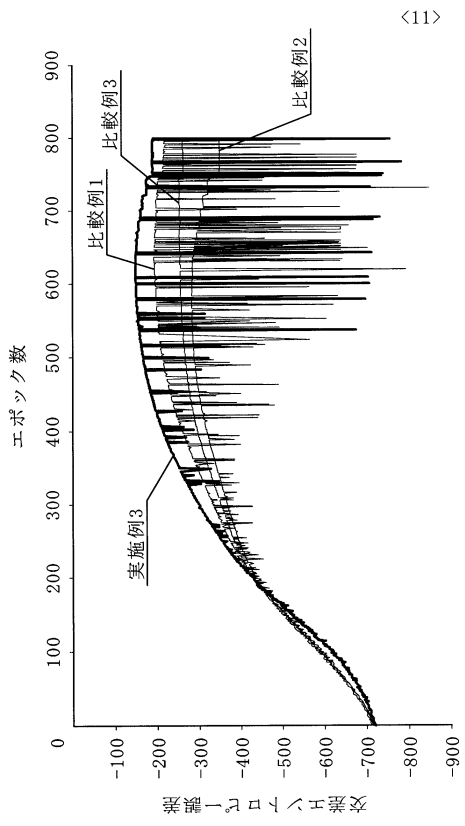
【図9】



【図10】



【図 11】



フロントページの続き

(72)発明者 森江 隆

福岡県北九州市若松区ひびきの2 - 4 国立大学法人九州工業大学内

審査官 渡辺 順哉

(56)参考文献 特開平07 - 248841 (JP, A)

特開2003 - 076272 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06N 3/00 - 99/00