(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2018-25920
(P2018-25920A)
(43)公開日　平成30年2月15日(2018.2.15)

| (51)Int.Cl. | | | F I | | テーマコード（参考） |
|---|---|---|---|---|---|
| G06N | 3/063 | (2006.01) | G06N | 3/063 | |
| G06N | 3/08 | (2006.01) | G06N | 3/08 | |

審査請求 未請求　請求項の数 8　ＯＬ　（全 17 頁）

(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)

申請有り

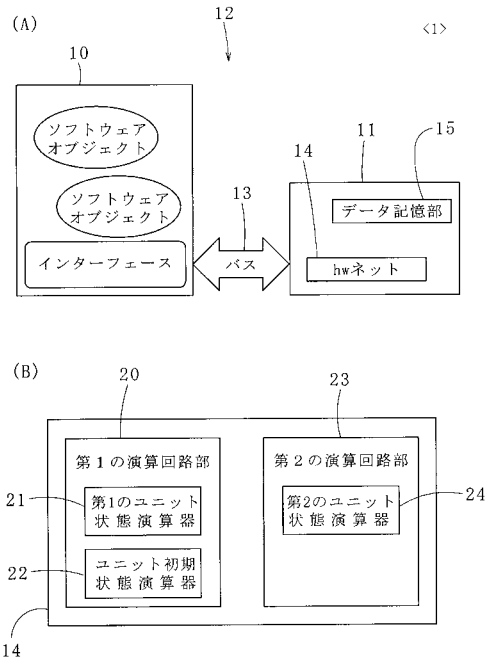
（特許庁注：以下のものは登録商標）
１．ＥＴＨＥＲＮＥＴ

(71)出願人　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
(74)代理人　100090697
弁理士　中前　富士男
(74)代理人　100176142
弁理士　清井　洋平
(74)代理人　100127155
弁理士　来田　義弘
(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
(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

最終頁に続く

(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)【要約】
【課題】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。
【解決手段】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、ユニットＢ毎の発火確率Ｐ（Ｂ）から発火又は非発火の状態を決定して出力する演算回路部１４を設け、演算回路部１４には、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２１、２４を設ける。
【選択図】図１



【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す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。
【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。
【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。

【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。
【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す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、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.
【非特許文献２】Seongwook Park,Kyeongryeol Bong,Dongjoo Shin,Jinmook Lee,Sungpil 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。
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。

【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、

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す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。

【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す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。

【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。

【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。

【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。

【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。

【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。

【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。

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】

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エアモジュール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、ハードウエアモジュール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。

【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）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、・・・、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、・・・、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）。

【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。

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、・・・、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、・・・、ｎ、ｊ＝１、２、・・・、ｍ）と、ｊ番目の隠れユニット１９のバイアス値$b_j$（ｊ＝１、２、・・・、ｍ）を用いて、$b_j+w_{1j}v_1+w_{2j}v_2+w_{3j}v_3+\cdots+w_{nj}v_n=b_j+\Sigma_i w_{ij}v_i$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）。

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Ｒｅｃｔｉｆｉｅｌｄ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、・・・、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。

【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。

【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、・・・、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$v_0$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、・・・、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、・・・、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、・・・、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、・・・、ｎ、ｊ＝１、２、・・・、ｍ）と、ｉ番目の可視ユニット１７のバイアス値$a_i$（ｉ＝１、２、・・・、ｎ）を用いて、$a_i+w_{i1}h_1+w_{i2}h_2+w_{i3}h_3+\cdots+w_{im}h_m=a_i+\Sigma_j w_{ij}h_j$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$v_i$（ｉ＝１、２、・・・、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。

【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。

【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、ＳＩＤＢＡ（Ｓｔａｎｄａｒｄ Ｉｍａｇｅ Ｄａｔａ－ＢＡｓｅ） 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。
【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タを最適値に収束させた。
【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。
【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タの更新を行う事前学習を行った。
【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。

【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。

【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。

【００４４】

（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で使用したのと同一構成のＲＢＭ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１１に示す。

【００４５】

図１１には、比較例１として、実施例３と同一構成のＲＢＭ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エ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１１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、ソフトウエ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。

【００４６】

図１１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。

【００４７】

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。

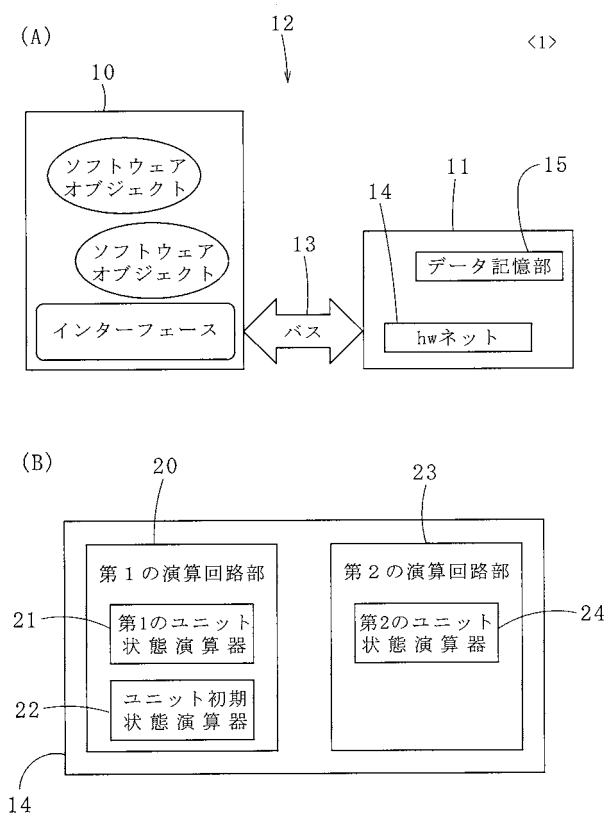
例えば、本実施の形態及び実施例では、固定小数点２進数のビット幅を超過して切り捨てられるビットを用いて形成する数値として、固定小数点２進数の小数部の下位側のビットから形成した数値を用いたが、固定小数点２進数の整数部の上位側ビットから形成した数値を用いることもできる。更に、固定小数点２進数の小数部の下位側のビットから形成される数値と整数部の上位側ビットから形成される数値を組合せて用いることもできる。

【符号の説明】

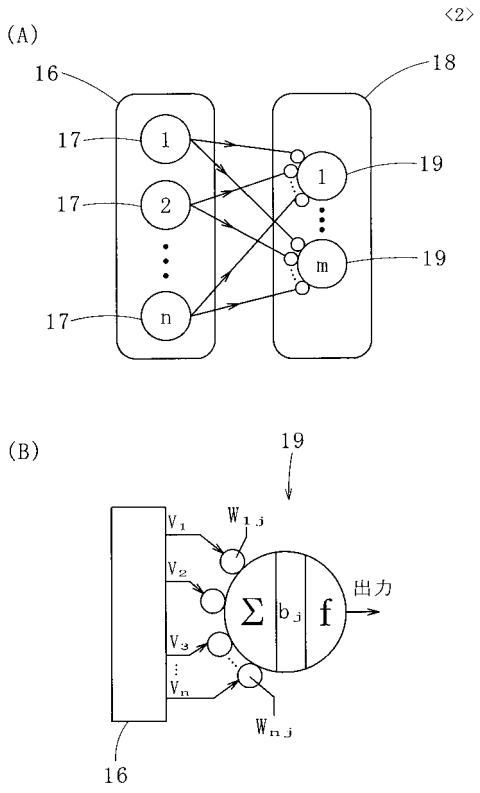
【００４８】

１０：主計算機、１１：ハードウエアモジュール、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

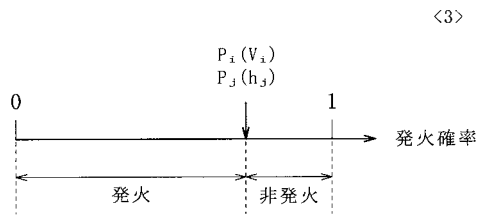
【図１】


(A)
12
<1>
10
ソフトウェア
オブジェクト
ソフトウェア
オブジェクト
インターフェース
13
バス
14
11
15
データ記憶部
hwネット
(B)
20
23
第１の演算回路部
第２の演算回路部
21
第1のユニット
状態演算器
24
第2のユニット
状態演算器
22
ユニット初期
状態演算器
14

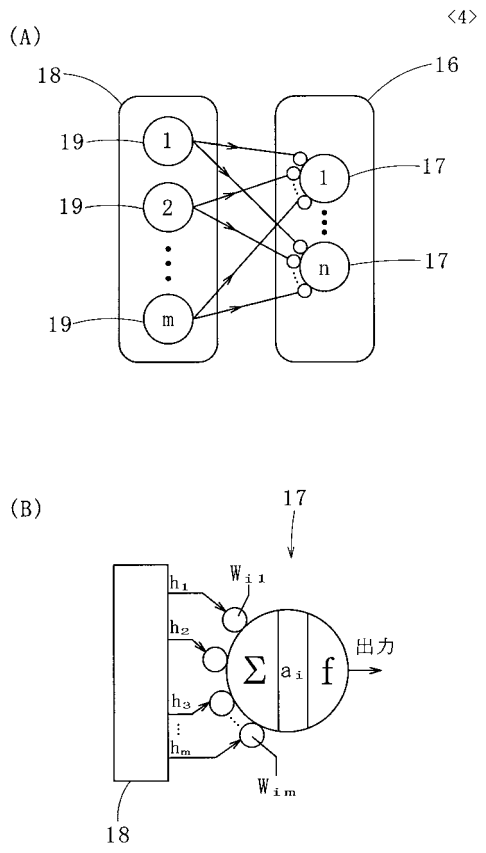
【図２】


<2>
(A)
16
18
17
1
2
n
1
m
19
19
17
17
(B)
19
$V_1$
$W_{1j}$
$V_2$
$V_3$
$V_n$
Σ
$b_j$
f
出力
$W_{nj}$
16

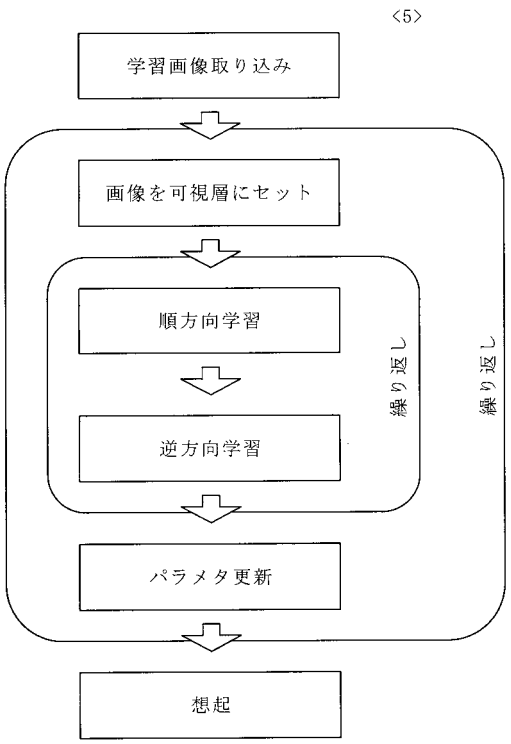
【図３】


<3>
$P_i(V_i)$
$P_j(h_j)$
0
1
発火確率
発火
非発火

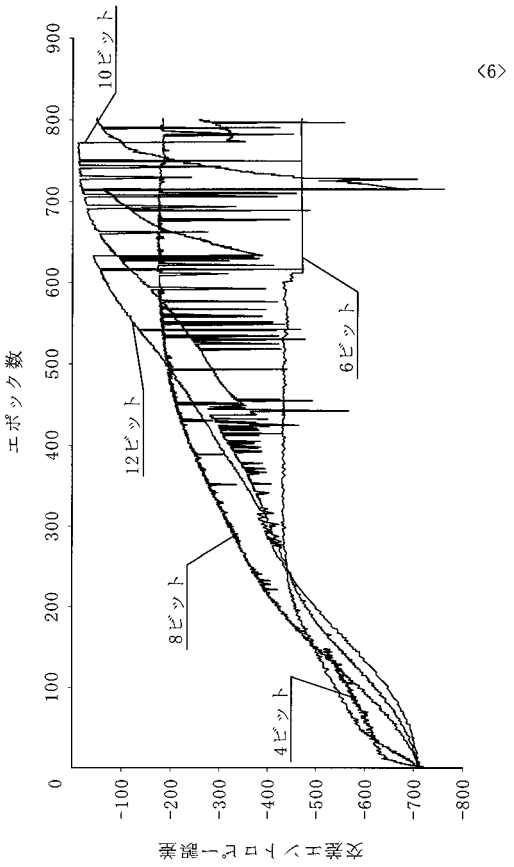
【図４】


<4>
(A)
18
16
19
1
2
m
1
n
17
17
19
19
(B)
17
$h_1$
$W_{i1}$
$h_2$
$h_3$
$h_m$
Σ
$a_i$
f
出力
$W_{im}$
18

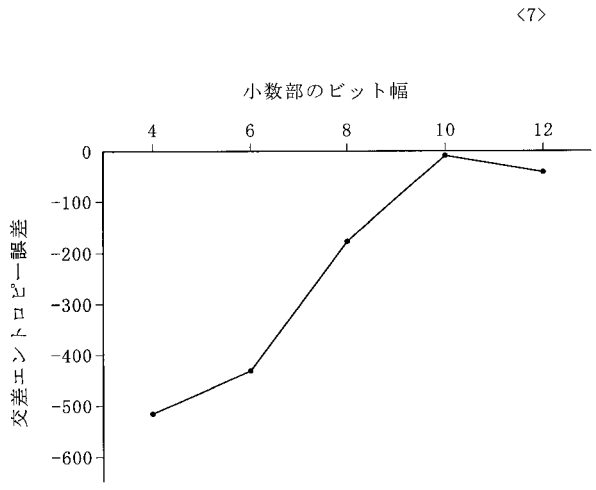
【図５】


〈5〉
学習画像取り込み
画像を可視層にセット
順方向学習
逆方向学習
パラメタ更新
想起
繰り返し
繰り返し

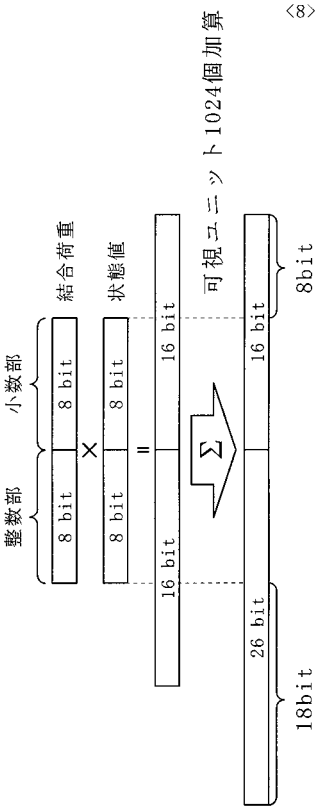
【図６】


〈6〉
エポック数
交差エントロピー誤差
10ビット
12ビット
8ビット
6ビット
4ビット
0
100
200
300
400
500
600
700
800
900
-100
-200
-300
-400
-500
-600
-700
-800

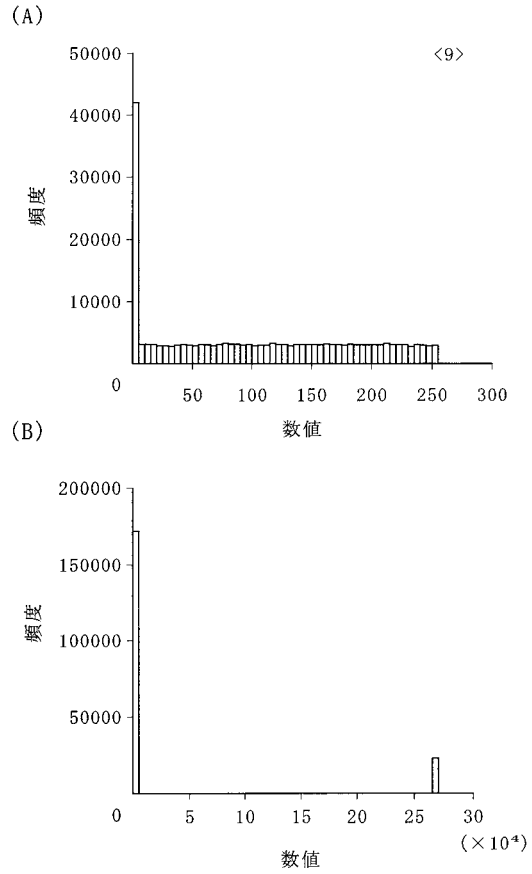
【図７】


〈7〉
小数部のビット幅
交差エントロピー誤差
4
6
8
10
12
0
-100
-200
-300
-400
-500
-600

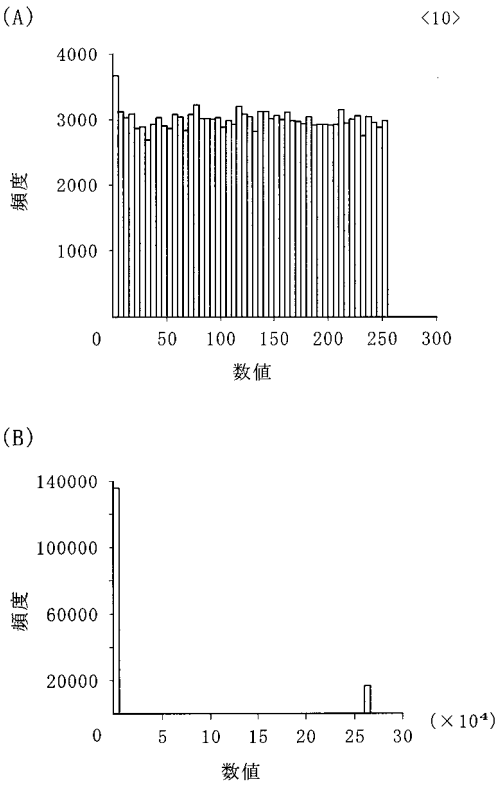
【図８】


〈8〉
可視ユニット1024個加算
結合荷重
状態値
小数部
整数部
8 bit
8 bit
8 bit
8 bit
16 bit
16 bit
16 bit
26 bit
8bit
18bit
×
=
Σ

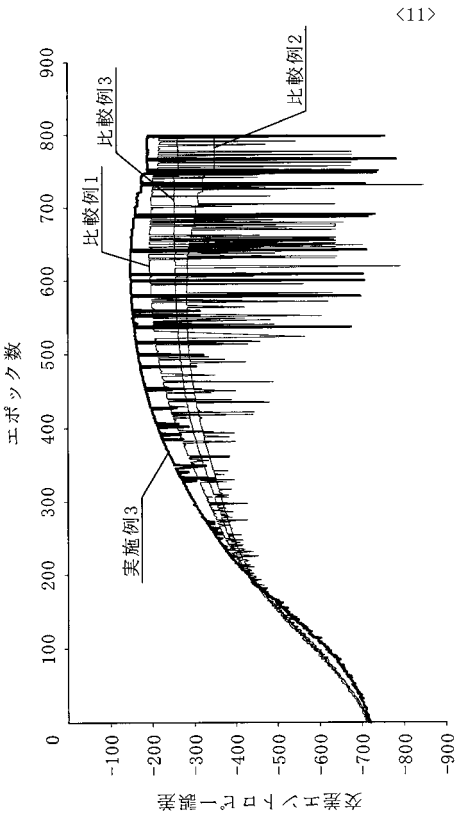
【図９】


(A)
<9>
頻度
数値
(B)
頻度
数値
(×10⁴)

【図１０】


(A)
<10>
頻度
数値
(B)
頻度
数値
(×10⁴)

【図１１】


<11>
比較例2
比較例3
比較例1
実施例3
エポック数
交差エントロピー誤差

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