

明 細 書

発明の名称：

ニューラルネットワーク回路装置、ニューラルネットワーク処理方法およびニューラルネットワークの実行プログラム

技術分野

[0001] 本発明は、ニューラルネットワーク回路装置、ニューラルネットワーク処理方法およびニューラルネットワークの実行プログラムに関する。

背景技術

[0002] 古典的な順伝搬型ニューラルネットワーク（F F N N : Feedforward Neural Network）、R B F（Radial Basis Function）ネットワーク、正規化したR B F ネットワーク、自己組織化マップなどがある。R B F Nは、誤差逆伝搬法に用いる活性化関数に放射基底関数を用いる。しかし、中間層が多く取れず高精度認識判定が難しかったり、HW規模が大きく処理時間がかかる、などの問題があり手書き文字認識など応用分野が限定されていた。

近年、A D A S（advanced driver assistance system）用の画像認識や自動翻訳などで注目を集める新方式として畳み込みニューラルネットワーク（C N N : Convolutional Neural Net-work）（層間が全結合でないN N）や再帰型ニューラルネットワーク（双方向伝搬）が登場している。C N Nは、ディープニューラルネットワーク（D N N : Deep Neural Network）に畳込み演算を付加したものである。

[0003] 特許文献1には、誤り訂正符号の検査行列に基づいて、階層型ニューラルネットワークにおける疎結合のノード間で学習された重みの値と入力信号とを用いて、問題を解く処理部を備える処理装置が記載されている。

[0004] 特許文献2には、ニューラルネットの重みパラメータを複数のグループに分割する分割手段と、分割されたグループごとに、重みパラメータをコードブックにより近似して符合化する符合化手段と、を有する情報処理装置が記載されている。重みパラメータは、2値あるいは3値の値を要素とすると記

載されている。CNNの高層の重みパラメータのようにスパースな重みに対しても精度の高い近似が可能になるとしている。

[0005] CNNを用いた多くの機械学習用途には、非常に多くの計算とメモリ帯域幅が必要となる。この必要条件を緩和する一つの方法は、係数をゼロ除去し、係数がゼロのとき計算をスキップすることである。

特許文献3には、畳み込みを実行する方法であって、(a)電子デバイスによって、現在の係数として、カーネルからの係数を選択することと、(b)前記電子デバイスによって、積の集合を得るために、前記現在の係数により、画像の少なくとも一部を乗算することと、(c)前記電子デバイスによって、前記積の集合を蓄積された和の集合に加算することと、(d)(a)から(c)に従って、前記カーネルの全ての係数が処理されるまで、(a)から(c)を繰り返すことと、を含む方法が記載されている。カーネルからの係数を選択する場合、カーネルの非ゼロ係数のみを選択すると記載されている。ゼロ係数スキップと新規の畳み込みエンジンを組み合わせることで、計算とメモリ帯域幅の両方を減少するとしている。

[0006] 既存のCNNは、短精度(多ビット)による積和演算回路で構成されており、多数の乗算回路が必要である。このため、面積・消費電力が多大になる欠点があった。そこで、2値化した精度、すなわち+1と-1(または0と1)のみ用いてCNNを構成する回路が提案されている(例えば、非特許文献1~4参照)。

先行技術文献

特許文献

[0007] 特許文献1: 特開2016-173843号公報

特許文献2: 特開2018-55260号公報

特許文献3: 特開2018-26134号公報

非特許文献

[0008] 非特許文献1: M. Courbariaux, I. Hubara, D. Soudry, R.E. Yaniv, Y. Bengio, "Binarized neural networks: Training deep neural networks with w

eights and activations constrained to +1 or -1,” Computer Research Repository (CoRR), 「2 値化 NN のアルゴリズム」、[online]、2016 年 3 月、[平成 28 年 10 月 5 日検索]、<URL: <http://arxiv.org/pdf/1602.02830v3.pdf> >

非特許文献 2: Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi, “XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks,” Computer Vision and Pattern recognition, 「2 値化 NN のアルゴリズム」、[online]、2016 年 3 月、[平成 28 年 10 月 5 日検索]、<URL: <https://arxiv.org/pdf/1603.05279v4> >

非特許文献 3: Hiroki Nakahara, Haruyoshi Yonekawa, Tsutomu Sasao, Hisashi Iwamoto and Masato Motomura, “A Memory-Based Realization of a Binarized Deep Convolutional Neural Network,” Proc. of the 2016 International Conference on Field-Programmable Technology (FPT), Xi'an, China, Dec 2016 (To Appear).

非特許文献 4: Eriko Nurvitadhi, David Sheffield, Jaewoong Sim, Asit Mishra, Ganesh Venkatesh, Debbie Marr, “Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC,” Proc. of the 2016 International Conference on Field-Programmable Technology (FPT), Xi'an, China, Dec 2016 (To Appear).

発明の概要

発明が解決しようとする課題

[0009] ニューロンを増大したディープニューラルネットワークは、機能を増やすことができ、かつ認識精度を格段に向上させることができる。しかしながら、このようなディープニューラルネットワークは、数十億～数百億の積和演算が必要である。このためには、大量の演算器およびメモリアクセスが不可欠となる。演算器およびメモリアクセスの増大は、いずれも高機能なディープニューラルネットワークを実現する上で大きな障害となっている。演算器については、微細化技術の進展によって増やすことは可能である。これに対

して、外部基板に接続する基板ピン数の物理的制約から、メモリアクセスを増やすことは非常に困難である（実装上限界）。メモリ量を削減してオンチップに配置することは、現在切実な要望となっている。

- [0010] 本発明は、このような事情に鑑みてなされたものであり、メモリ量を削減してオンチップに配置が可能なニューラルネットワーク回路装置、ニューラルネットワーク処理方法およびニューラルネットワークの実行プログラムを提供することを課題とする。

課題を解決するための手段

- [0011] 前記した課題を解決するため、本発明に係るニューラルネットワーク回路装置は、入力層、1 以上の中間層、および、出力層を少なくとも含むニューラルネットワーク回路装置であって、前記中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う畳み込み演算回路と、畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路と、総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路と、を備え、前記畳み込み演算回路は、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行うことを特徴とする。

その他の手段については、発明を実施するための形態のなかで説明する。

発明の効果

- [0012] 本発明によれば、メモリ量を削減してオンチップに配置が可能なニューラルネットワーク回路装置、ニューラルネットワーク処理方法およびニューラルネットワークの実行プログラムを提供することができる。

図面の簡単な説明

- [0013] [図1]ディープニューラルネットワーク（DNN）の構造の一例を説明する図である。

[図2]本発明の実施形態に係るディープニューラルネットワークの非ゼロ畳み込み演算回路を備えるニューラルネットワーク回路の構成を示す図である。

[図3]本発明の実施形態に係るディープニューラルネットワークの非ゼロ畳み

込み演算回路の畳み込み演算の概念を示す図である。

[図4]本発明の実施形態に係るディープニューラルネットワークの非ゼロ畳み込み演算回路の構成を示す図であり、(a)はその回路構成図、(b)はその0スキップの一例を示す図である。

[図5]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワーク回路の構成を示す図である。

[図6]本発明の実施形態に係るディープニューラルネットワークの3値化ニューラルネットワーク回路の構成を示す図である。

[図7]本発明の実施形態に係るディープニューラルネットワークの間接メモリアクセスを説明する図であり、(a)は重み／アドレスメモリに格納される非零重みと相対アドレスの一例を示す図、(b)は“0”スキップする非零重み w を示す図、(c)は畳み込みを行う入力値 X の読み出しを示す図である。

[図8]本発明の実施形態に係るディープニューラルネットワークの間接アドレスの表現方法を説明する図であり、(a)は絶対アドレス指定方法を示す図、(b)は相対アドレス指定方法を示す図である。

[図9]本発明の実施形態に係るディープニューラルネットワークの畳み込みニューラルネットワークへの適用を説明する図である。

[図10]本発明の実施形態に係るディープニューラルネットワークの“0”スキップする演算の効果を説明する図である。

[図11]本発明の実施形態に係るディープニューラルネットワークの間接メモリアクセスを説明する図である。

[図12]本発明の実施形態に係るディープニューラルネットワークのビット精度を減らすことでメモリ量を削減する例を説明する図であり、(a)はニューラルネットワーク回路の構成図、(b)はメモリ量を削減できる原理図である。

[図13]本発明の実施形態に係るディープニューラルネットワークの0スキップ演算でメモリ量を削減する例を説明する図であり、(a)はニューラルネ

ットワーク回路の構成図、（b）はメモリ量を削減できる原理図である。

[図14]本発明の実施形態に係るディープニューラルネットワークの比較例のディープニューラルネットワークの学習方法を説明する図である。

[図15]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワークの学習方法を説明する図である。

[図16]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワークの学習方法を説明する図である。

[図17]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワークの認識精度向上を説明する図である。

[図18]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワーク回路と3値ニューラルネットワーク回路の組み合わせによる物体検出実証例を説明する図である。

[図19]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワーク回路と物体認識アルゴリズムYOLOv2との比較を説明する図である。

[図20]本発明の実施形態に係るディープニューラルネットワークの3状態ニューラルネットワーク回路の実装例を説明する図である。

発明を実施するための形態

[0014] 以下、図面を参照して本発明を実施するための形態（以下、「本実施形態」という）におけるディープニューラルネットワークについて説明する。

（背景説明）

図1は、ディープニューラルネットワーク（DNN）の構造の一例を説明する図である。

図1に示すように、ディープニューラルネットワーク（DNN）1は、入力層（input layer）11、任意の数の中間層である隠れ層（hidden layer）12、出力層（output layer）13を有して構成される。

入力層（input layer）11は、複数個（ここでは8）の入力ノード（ニューロン）を有する。隠れ層12は、複数（ここでは3層（hidden layer1, hi

dden layer2, hidden layer3)) である。実際には、隠れ層 1 2 の層数 n は、例えば 20 ~ 100 に達する。出力層 1 3 は、識別対象の数（ここでは 4）の出力ノード（ニューロン）を有する。なお、層数およびノード数（ニューロン数）は、一例である。

ディープニューラルネットワーク 1 は、入力層 1 1 と隠れ層 1 2 のノード間が全て結合し、隠れ層 1 2 と出力層 1 3 のノード間が全て結合している。

[0015] 入力層 1 1、隠れ層 1 2 および出力層 1 3 には、任意の数のノード（図 1 の○印参照）が存在する。このノードは、入力を受け取り、値を出力する関数である。入力層 1 1 には、入力ノードとは別に独立した値を入れるバイアス (bias) ノードがある。構成は、複数のノードを持つ層を重ねることで構築される。伝播は、受け取った入力に対して重み (weight) をかけ、受け取った入力を次層に活性化関数 (activation function) で変換して出力する。活性化関数は、sigmoid関数やtanh関数などの非線形関数、ReLU (Rectified Linear Unit function: 正規化線形関数) がある。ノード数を増やすことで、扱う変数を増やし、多数の要素を加味して値／境界を決定できる。層数を増やすことで、直線境界の組み合わせ、複雑な境界を表現できる。学習は、誤差を計算し、それを基に各層の重みを調整する。学習は、誤差を最小化する最適化問題を解くことであり、最適化問題の解法は誤差逆伝播法 (Backpropagation) を使うのが一般的である。誤差は、二乗和誤差を使うのが一般的である。汎化能力を高めるために、誤差に正則化項を加算する。誤差逆伝播法は、誤差を出力層 1 3 から伝播させていき、各層の重みを調整する。

[0016] 図 1 のディープニューラルネットワーク 1 の構成を 2 次元に展開することで画像処理に適した CNN を構築できる。また、ディープニューラルネットワーク 1 にフィードバックを入れることで、双方向に信号が伝播する RNN (Recurrent Neural Network: 再帰型ニューラルネットワーク) を構成することができる。

[0017] 図 1 の太破線三角部に示すように、ディープニューラルネットワーク 1 は、多層のニューラルネットワークを実現する回路（以下、ニューラルネット

ワーク回路という) 2 から構成されている。

本技術は、ニューラルネットワーク回路 2 を対象とする。ニューラルネットワーク回路 2 の適用箇所および適用数は限定されない。例えば、隠れ層 1 2 の層数 n : 20 ~ 30 の場合、これらの層のどの位置に適用してもよく、またどのノードを入出力ノードとするものでもよい。さらに、ディープニューラルネットワーク 1 に限らず、どのようなニューラルネットワークでもよい。ただし、入力層 1 1 または出力層 1 3 のノード出力には、2 値化出力ではなく多ビット出力が求められるので、ニューラルネットワーク回路 2 は、対象外である。ただし、出力層 1 3 のノードを構成する回路に、乗算回路が残ったとしても面積的には問題にはならない。

なお、入力データに対し学習済のものを評価していくことを前提としている。したがって、学習結果として重み W_i は既に得られている。

[0018] <非ゼロ畳み込み演算回路を備えるニューラルネットワーク回路>

図 2 は、本実施形態に係る非ゼロ畳み込み演算回路 2 1 を備えるニューラルネットワーク回路 2 0 の構成を示す図であり、人工ニューロンモデル (Artificial Neuron (AN) Model) に適用した例である。なお、ゼロは、0、零と表記することがある。

本実施形態のニューラルネットワーク回路 2 0 は、図 1 のディープニューラルネットワーク 1 を構成するニューラルネットワーク回路 2 に適用できる。

[0019] ニューラルネットワーク回路 2 0 は、畳み込みを行う入力値 X_i と、重み W_i を受け取り、非ゼロ畳み込み演算を行う非ゼロ畳み込み演算回路 2 1 (畳み込み演算回路) と、畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路 2 2 と、総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路 2 3 と、を備えて構成される。

[0020] ニューラルネットワーク回路 2 0 は、基本演算が、入力値 X_i と重み W_i の積和演算を行う人工ニューロン (AN) である。

x_i : Input signal (入力値)

w_0 : Bias (バイアス)

w_i : Weight (重み)

u : Internal state (中間値)

$f(u)$: Activation function (活性化関数、例えばSigmoid, ReLU, etc.)

y : Output signal (出力値)

[0021] 非ゼロ畳み込み演算回路21は、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う。

非ゼロ畳み込み演算回路21は、3状態 {−多ビット, 0, +多ビット} の重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う。

非ゼロ畳み込み演算回路21は、3値化 {−1, 0, +1} または2値化 {−1, +1} の重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う。

[0022] <閾値 ρ と関数 $h(x)$ の組み合わせによる状態表現>

本発明は、重みの閾値 ρ と活性化関数 $f(u)$ の組み合わせで様々な状態のニューラルネットワークを表現することを特徴とする。 ρ が閾値、 W_i が重みである。重み W_i が ± 1 のとき、3値化ニューラルネットワーク、重み W_i が $\pm W$ (多ビット精度) のとき、3状態ニューラルネットワークである。

(1) $\rho = 0$, $h(x)$ が $\text{sign}(x)$ 符号関数の場合、ニューラルネットワーク回路20は、2値ニューラルネットワーク回路を構成する。

[0023] [数1]

$$W_i = \begin{cases} -1 & W_{hid} < 0 \\ +1 & 0 \leq W_{hid} \end{cases} \quad \dots(1)$$

[0024] (2) $\rho \neq 0$, $h(x)$ が $\text{sign}(x)$ 符号関数の場合、ニューラルネットワーク回路20は、3値化ニューラルネットワーク回路を構成する。

[0025] [数2]

$$W_i = \begin{cases} -1 & W_{hid} < \rho \\ 0 & -\rho \leq W_{hid} < \rho \\ +1 & \rho \leq W_{hid} \end{cases} \quad \dots(2)$$

[0026] (3) $\rho \neq 0$, $h(x)$ が $\text{sign}(x)$ 符号関数の場合、ニューラルネットワーク回路 20 は、3 状態ニューラルネットワーク回路を構成する。

[0027] [数3]

$$W_i = \begin{cases} W_{hid} & W_{hid} < \rho \\ 0 & -\rho \leq W_{hid} < \rho \\ W_{hid} & \rho \leq W_{hid} \end{cases} \quad \dots(3)$$

ただし、

W_{hid} : 多ビット重み W_i

ρ : 閾値

[0028] 特に、上記数式 (3) の閾値 ρ と活性化関数 $f(u)$ の組み合わせを用いることで、非ゼロ畳み込み演算回路 21 を、3 状態畳み込み演算回路で構成する。これにより、ニューラルネットワーク回路 20 は、3 状態ニューラルネットワーク回路を構成する。

[0029] また、上記数式 (2) の閾値 ρ と活性化関数 $f(u)$ の組み合わせを用いることで、非ゼロ畳み込み演算回路 21 を、3 値化畳み込み演算回路で構成する。これにより、ニューラルネットワーク回路 20 は、3 値化ニューラルネットワーク回路を構成する。

上記 3 状態ニューラルネットワーク回路と 3 値化ニューラルネットワーク回路は、いままでにない本発明のニューラルネットワークである。

[0030] <3 状態ニューラルネットワーク回路と 3 値化ニューラルネットワーク回路の比較>

3 値化ニューラルネットワーク回路は、2 値化 $\{-1, +1\}$ 畳み込み演算

回路よりも省面積かつ高速である。

3 値化ニューラルネットワーク回路は、重み W_i が低ビット（ -1 , 0 , 1 ）であるので、省面積かつ高速な回路である。ただし、3 状態ニューラルネットワーク回路に比べて認識精度が低くなるため用途は限られる。

[0031] 3 状態ニューラルネットワーク回路は、重み W_i が多ビット（例えば $8 \sim 16$ ビットの $\pm W_i$ と “0”）であるので、認識精度は高く用途（応用範囲）も広い。3 状態ニューラルネットワーク回路は、3 値化ニューラルネットワーク回路よりも認識精度が高い。ただし、3 値化ニューラルネットワーク回路に比べて面積（特にメモリ量）が増える。また、3 値化ニューラルネットワーク回路よりも処理速度が遅い。

[0032] 3 状態ニューラルネットワーク回路と重み 3 値化ニューラルネットワーク回路の組み合わせ、または、3 状態ニューラルネットワーク回路と重み 2 値化ニューラルネットワーク回路の組み合わせが可能である。

3 値化ニューラルネットワーク回路は、3 状態ニューラルネットワーク回路に比べて認識精度が低いので用途が狭いものの、3 状態ニューラルネットワーク回路と併用することで高速化・面積削減が可能になる。3 状態ニューラルネットワーク回路と 3 値化（または 2 値化）ニューラルネットワーク回路を組み合わせると、面積と認識精度のバランスが取れた AI 回路を実現可能である。

[0033] <非ゼロ畳み込み演算回路 21>

図 3 は、非ゼロ畳み込み演算回路 21 の畳み込み演算の概念を示す図である。

非ゼロ畳み込み演算回路 21 は、CNN において 3 状態 $\{-W(\text{多ビット}), 0, +W(\text{多ビット})\}$ または 3 値化 $\{-1, 0, +1\}$ 畳み込み演算を行う。

図 3 に示すように、非ゼロ畳み込み演算回路 21 を備えるニューラルネットワーク回路 20 は、物体検出 (Dilated Convolution) する際に、画像データと重み関数の畳み込み演算を行う。例えば、Input Feature Map に画像（行

列の各要素はそれぞれ画像の1ピクセルに対応)を入力する。この入力画像に対して、カーネル(kernel)(この例では $K=3$)の値と行列の値を要素毎に掛け合わせ、それらの値を合計する。この操作をカーネルをスライドさせながら各要素に対して行い全体の畳み込み演算を行う。

図3に示すように、非ゼロ畳み込み演算回路21は、重み関数に、0が多い場合は、0をスキップする回路構成とする(後記)。

[0034] ディープラーニングは学習と推論の2つのフェーズがあり、通常は画像や音声のデータを使って学習を行い、重みを確定する。その後、推論用のデバイスに学習済みデータを変換(ここでは低ビット化と枝刈り(ゼロ化))する。したがって、非零の重みとその位置は学習後に判明することになる。学習後、推論のみの場合は、位置や値の更新は行われない。なお、再学習が必要な時は、位置や値の変更が行われてしまうことになる。

[0035] 図4は、非ゼロ畳み込み演算回路21の構成を示す図であり、図4(a)はその回路構成図、図4(b)はその0スキップの一例を示す図である。

図4(a)に示すように、非ゼロ畳み込み演算回路21は、非ゼロの重みと畳み込みを行う入力値 X_i の相対アドレスを格納する重み/アドレスメモリ213(第1記憶手段)と、加算器214と、入力値 X_i のアドレスを格納するアドレス/データメモリ215(第2記憶手段)と、積算回路216と、を備える。

非ゼロ畳み込み演算回路21は、ゼロの重みをスキップする演算の際、重み/アドレスメモリ213から該当する非ゼロの重みとその相対アドレスを読み込み、読み込んだ相対アドレスと現在のアドレスからアドレス/データメモリ215を参照して次の畳み込みを行う入力値 X_i を読み出し、アドレス/データメモリ215から読み出した入力値 X_i と該当する非ゼロの重みをもとに畳み込み演算を行う(図8(b)参照)。

[0036] 具体的には、重み/アドレスメモリ213は、インデックス $i \times \text{毎}$ に非零重み $w_1, \dots$ と相対アドレス $adr_1, \dots$ とを格納する。重み/アドレスメモリ213は、入力されたカウンタ値に従って、該当するインデックス i

d_x の非零重み W_i と相対アドレス adr を読み出す。加算器 214 は、ベースアドレス X_{tmp} に、相対アドレス adr を加算してアドレスを作成し、アドレス／データメモリ 215 に出力する。図 4 (b) の矢印に示すように、非ゼロ畳み込み演算回路 21 は、ベースアドレス X_{tmp} を基点とし、そこから相対アドレス adr_1 , adr_2 先のアドレスを作成する。

[0037] アドレス／データメモリ 215 は、アドレスとそのアドレスに対応するデータ X_1 , ... を格納する。非ゼロ畳み込み演算回路 21 は、アドレス／データメモリ 215 から、加算器 214 から出力されたアドレスに対応するデータ (入力値 X_i) を読み出す。

積算回路 216 は、重み／アドレスメモリ 213 から読み出した非零重み W_i とアドレス／データメモリ 215 から読み出した入力値 X_i とを乗算して出力値 Y_i を出力する。

[0038] このように、非ゼロ畳み込み演算回路 21 は、重みが 0 でない相対アドレスを指定することで、1 つの演算が終わったら次の相対アドレスにジャンプして必要な箇所だけを畳み込み演算する。すなわち、非ゼロ畳み込み演算回路 21 は、重みが 0 でない相対アドレスによる演算スキップを行う。

前記図 3 に示すように、非ゼロ畳み込み演算回路 21 は、0 をスキップしながら、カーネルの値 (畳み込みを行う入力値 X) と行列の値 (非零重み w) を要素毎に掛け合わせる。

畳み込み演算を行う重み関数が疎 (ゼロが多い) である特徴を生かして、それを相対アドレス表示して重みが非ゼロの箇所のみを演算処理する回路構成にすることによって、計算時間の高速化、メモリ面積の削減が可能となる。

なお、本実施形態では、学習の最適化により、90~95% のスキップが可能になっている。

[0039] 非ゼロ畳み込み演算回路 21 は、非ゼロの重みと畳み込みを行う入力値 X_i の絶対アドレスを格納するメモリ (第 3 記憶手段) (図示省略) を有する。そして、非ゼロ畳み込み演算回路 21 は、ゼロの重みをスキップする演算

の際、このメモリから絶対アドレスを指定して該当する非ゼロの重みと入力値 X_i を読み出し、読み出した入力値 X_i と該当する非ゼロの重みをもとに畳み込み演算を行うものであってもよい（図8（a）参照）。

[0040] [実施形態の構成]

〈3状態ニューラルネットワーク回路200〉

図5は、本発明の実施形態に係るニューラルネットワークの3状態ニューラルネットワーク回路200の構成を示す図である。図4と同一構成部分には、同一符号を付している。

本実施形態の3状態ニューラルネットワーク回路200は、ディープニューラルネットワークへの実装技術を提供する。

3状態ニューラルネットワーク回路200（ニューラルネットワーク回路装置）は、図1のニューラルネットワーク回路2に適用できる。

図5に示すように、3状態ニューラルネットワーク回路200は、カウンタ値を出力するカウンタ211と、ベースアドレスを出力するレジスタ212と、重み／アドレスメモリ213と、加算器214と、アドレス／データメモリ215と、積算回路216と、レジスタ217と、加算器218と、バイアス値を出力するレジスタ219と、加算器220と、活性化関数回路221Aと、を備える。

上記乗算器216、レジスタ217、加算器218、レジスタ219と、加算器220、および活性化関数回路221Aは、Sequential MAC Unit222に搭載される。

上記カウンタ211、ベースアドレスを出力するレジスタ212、重み／アドレスメモリ213、加算器214、アドレス／データメモリ215、および積算回路216は、非ゼロ畳み込み演算回路21を構成する。

[0041] 3状態ニューラルネットワーク回路200は、3状態（Tri-state） $\{-w, 0, +w\}$ 畳み込み演算回路（非ゼロ畳み込み演算回路21）を備える。

活性化関数回路221は、Leaky ReLU符号関数を用いる。

上記重み／アドレスメモリ213の一部と加算器214には、間接メモリ

アクセス（図5破線囲み参照）を導入している（図11で後述）。

[0042] 3状態ニューラルネットワーク回路200は、非ゼロ畳み込み演算回路21を備えることで、計算時間の高速化、メモリ面積の削減が可能となったり。従来技術では、オンチップに配置することは、極めて困難であった。本実施形態によれば、非ゼロ畳み込み演算回路21を備えることで、畳み込み演算部分の積和演算の絶対数を劇的に減らして、オンチップ配置を実現した。

[0043] <3値化ニューラルネットワーク回路200A>

図6は、本発明の実施形態に係るニューラルネットワークの3値化ニューラルネットワーク回路200Aの構成を示す図である。図4と同一構成部分には、同一符号を付している。

本実施形態の3値化ニューラルネットワーク回路200Aは、3状態ニューラルネットワーク回路200と同様にディープニューラルネットワークへの実装技術を提供する。また、3値化ニューラルネットワーク回路200Aは、3状態ニューラルネットワーク回路200と併用して好適である。

3状態ニューラルネットワーク回路200（ニューラルネットワーク回路装置）は、図1のニューラルネットワーク回路2に適用できる。

[0044] 図6に示すように、3値化ニューラルネットワーク回路200Aは、カウンタ値を出力するカウンタ211と、ベースアドレスを出力するレジスタ212と、重み／アドレスメモリ213と、加算器214と、アドレス／データメモリ215と、XNOR（Exclusive NOR：否定排他的論理和）論理を取る複数のXNORゲート回路216Aと、レジスタ217と、加算器218と、バイアス値を出力するレジスタ219と、加算器220と、活性化関数回路221Aと、を備える。

上記XNORゲート回路216A、レジスタ217、加算器218、レジスタ219と、加算器220、および活性化関数回路221Aは、Sequential MAC Unit 222Aに搭載される。

上記カウンタ211、ベースアドレスを出力するレジスタ212、重み／アドレスメモリ213、加算器214、アドレス／データメモリ215、お

よびXNORゲート回路216Aは、非ゼロ畳み込み演算回路21Aを構成する。

[0045] 3値化ニューラルネットワーク回路200Aは、3値{-1, 0, +1}畳み込み演算回路（非ゼロ畳み込み演算回路21A）を備える。

3値化ニューラルネットワーク回路200Aは、積算回路216（図5参照）がXNOR論理を実現するXNORゲート回路216Aに置き換えられている。このため、積算回路216を構成する際に必要であった面積を削減することができる。また、重み W_i は、3値{-1, 0, +1}であるため、多値である3状態と比較してメモリ量を大幅に削減でき、メモリ帯域を向上させることができる。

活性化関数回路221Aは、 $\text{Sign}(x)$ 符号関数を用いる。

上記重み／アドレスメモリ213の一部と加算器214には、間接メモリアクセス（図6破線囲み参照）を導入している（図11で後述）。間接メモリアクセスの導入により“0”スキップを実現している。

[0046] <間接メモリアクセス>

間接メモリアクセスについて説明する。

図7は、間接メモリアクセスを説明する図であり、図7(a)は重み／アドレスメモリ213に格納される非零重みと相対アドレスの一例を示す図、図7(b)は“0”スキップする非零重み w を示す図、図7(c)は畳み込みを行う入力値 X の読み出しを示す図である。

図7(a)に示すように、重み／アドレスメモリ213は、1D毎に非零重み $W(w_1, \dots)$ と相対アドレス $adr_1, \dots$ を指定するインデックス L を格納する。

インデックス L は、(Input Feature Map(入力画像), row(行), column(列))を指定する。

図7に示す間接メモリアクセスでは、下記1. 2. を実行する。

1. 該当する重みと相対アドレスを読み込む。
2. 相対アドレスと現在のアドレスから畳み込みを行う入力値 X_i を読み出

す。

[0047] <間接アドレスの表現>

間接アドレスの表現方法について説明する。

図8は、間接アドレスの表現方法を説明する図であり、図8(a)は絶対アドレス指定方法を示す図、図8(b)は相対アドレス指定方法を示す図である。

図8(a)に示すように、絶対アドレス指定は、{ページ, Y, X}を用いて絶対アドレスを指定する。この例では、図8(a)に示す元の絶対アドレス(0, 0)を基点とし、そこから{ページ, Y, X}で示されるアドレスを指定する。例えば、図8(a)の矢印に示すように、元の絶対アドレス(0, 0)を基点に、{+0, +2, +0}を指定して絶対アドレス(2, 0)の値(図8(a)網掛け参照)を得る。以下同様に、図8(a)の矢印に示すように、元の絶対アドレス(0, 0)を基点に、{+1, +2, +1}を指定して、次のページの画像データの絶対アドレス(2, 1)の値(図8(a)網掛け参照)を得る。

[0048] 絶対アドレス指定方法は、途中でメモリの値が化けても正しく演算できる。ただし、絶対アドレスが長いので、相対アドレス方法と比較してメモリ量が増えてしまうことに留意する。

[0049] 図8(b)に示すように、相対アドレス指定方法は、次のアドレスを相対値で指定する。例えば、図8(b)の矢印に示すように、元のアドレス(ベースアドレス)(0, 0)を基点に、+2を指定して、次のアドレス(1, 3)の値(図8(b)網掛け参照)を得る。そこから、+5を指定して、次のアドレス(3, 2)の値、さらに、+5を指定して、次のページの画像のアドレス(2, 1)の値を得る。

[0050] 相対アドレス指定方法は、相対値を記憶しておけばよいので絶対アドレス方法と比較してメモリ量を減らすことができる。ただし、相対アドレス指定方法は、何らかの理由で、演算途中でメモリの値が化けた時(変化した時)に演算が誤ってしまうことになる。

[0051] 本実施形態では、メモリ量を減らす観点から、0スキップ演算に相対アドレス指定方法を採用する。図4に示す非ゼロ畳み込み演算回路21は、この相対アドレス指定方法を採用して、重み／アドレスメモリ213から相対アドレス *adr* を読み出し、加算器214で、順次、*Xtmp* を加算して先のアドレスを指定する。

[0052] <畳み込みニューラルネットワークへの適用>

畳み込みニューラルネットワークへの適用について説明する。

図9は、畳み込みニューラルネットワークへの適用を説明する図である。

図10は、“0”スキップする演算の効果を説明する図である。

図9に示すように、畳み込みニューラルネットワークは、Input feature mapと、Kernel(Sparse)と、Output feature mapと、を有する。

この畳み込みニューラルネットワークに、3状態ニューラルネットワーク回路200（図5参照）を適用する。

3状態ニューラルネットワーク回路200は、前記したように0スキップ演算を行う非ゼロ畳み込み演算回路21を備える。

[0053] 0スキップ演算を導入することにより、図10の破線囲みに示す演算がスキップされる。これにより、畳み込み演算部分の積和演算の絶対数を劇的に減らすことができ、メモリ量削減、計算時間の高速化を図ることができる。

[0054] また、図9に示す畳み込みニューラルネットワークに、3値化ニューラルネットワーク回路200A（図6参照）を適用してもよい。この場合、3値化ニューラルネットワーク回路200Aは、3値{−1, 0, +1}を用いるため、多値である3状態ニューラルネットワーク回路200と比較してメモリ量を大幅に削減できる。

[0055] <“0”をスキップする演算；間接メモリアクセス>

“0”をスキップする演算、すなわち間接メモリアクセスについて説明する。

図11は、“0”をスキップする演算；間接メモリアクセスを説明する図である。図9に示す畳み込みニューラルネットワークに、図7および図8（

a) の間接メモリアクセスを適用した場合を例に採る。

上述したように、“0”をスキップすることで、メモリ量削減と処理の高速化を図ることを目的とする。

本実施形態では、間接メモリアクセス（相対アドレスによる2回のメモリアクセス）を導入する。具体的には、下記の通りである。

図11に示すように、

1. 図11の実線矢印に示すように、1回目のメモリアクセスで、相対値を読み込む。
2. ベースアドレスと相対値から重み“0”でない入力アドレスを計算する。
3. 2回目のメモリアクセスで、計算したアドレスから入力値を読み込む。
4. 図11の破線矢印に示すように、積算（ニューラルネットワークの演算）を行う。

[0056] <メモリ量を削減できる原理>

メモリ量を削減できる原理について説明する。

図12は、ビット精度を減らすことでメモリ量を削減する例を説明する図であり、図12(a)はニューラルネットワーク回路20（図2参照）の構成図、図12(b)はメモリ量を削減できる原理図である。

図12(a)に示すニューラルネットワーク回路20は、非ゼロ畳み込み演算回路21において、例えば32ビット精度から1ビット精度{-1, +1}にビット数の削減を行う。

これにより、図12(b)に示すように、メモリ量の削減を図ることができる。

[0057] 図13は、0スキップ演算でメモリ量を削減する例を説明する図であり、図13(a)はニューラルネットワーク回路20（図2参照）の構成図、図13(b)はメモリ量を削減できる原理図である。

図13(a)の×印に示すように、ニューラルネットワーク回路20は、非ゼロ畳み込み演算回路21において、“0”をスキップする（ここでは、

重み w_2 の乗算をスキップ)。すなわち、0の乗算は、0になることが既知であるためスキップする。これにより、メモリ量削減と処理の高速化を図ることができる。

[0058] 上記ニューラルネットワーク回路20は、3状態(0、±多ビット)畳み込み演算を行う3状態ニューラルネットワーク回路200(図5参照)である。3状態ニューラルネットワーク回路200は、低ビット化(図12のメモリ量削減)と0スキップ演算(図13のメモリ量削減)を併せて実行するので、メモリ量をより一層削減することができる。

[0059] [実施形態のディープニューラルネットワークの学習方法]

ディープニューラルネットワークの学習方法について、比較例と本実施形態の3状態ニューラルネットワーク回路200とを比較して説明する。

＜比較例＞

図14は、比較例のディープニューラルネットワークの学習方法を説明する図である。

図14の上段に示すように、比較例のディープニューラルネットワークでは、誤差逆伝搬法により、学習データ(主に画像)を、ニューラルネットワークにおいて順伝搬させる。ニューラルネットワークの出力として、cat88%、dog10%、car5%を得る。

[0060] 図14の下段に示すように、比較例のディープニューラルネットワークでは、教師データcat100%、dog0%、car0%を入力し、順伝搬の出力との誤差を計算する。そして、教師データとの誤差をディープニューラルネットワークの出力層のノードに入力し、ニューラルネットワークを逆伝搬させて重み W_i を更新していく。

上記順伝搬と逆伝搬および重み W_i 更新を何度も繰り返すことで、ディープニューラルネットワークの学習を行う。

学習終了後、丸め処理をしない状態で保っていた重み W_i を0にして、重みが疎なニューラルネットワークを生成する。

[0061] ＜本実施形態＞

図15および図16は、本実施形態の3状態ニューラルネットワークの学習方法を説明する図である。3状態ニューラルネットワークは、例えば図5に示す3状態ニューラルネットワーク回路200を適用する。

3状態ニューラルネットワーク回路200は、ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて重み W_i を更新する、学習を繰り返す場合、学習が終了するまでは少なくとも中間層の中では、重み W_i を丸め処理しない。

図15の上段に示すように、学習データ（主に画像）を、3状態で順伝搬させる。3状態ニューラルネットワーク内部では、 W_i を保持しておく。3状態ニューラルネットワークの出力として、cat88%、dog10%、car5%を得る。

[0062] 図15の下段に示すように、3状態ニューラルネットワークでは、教師データcat100%、dog0%、car0%を入力し、順伝搬の出力との誤差を計算する。そして、教師データとの誤差を3状態ニューラルネットワークの出力層のノードに入力し、3状態ニューラルネットワークを逆伝搬させて重み w を更新していく。 W_i は多ビット値のままである。

上記順伝搬と逆伝搬および重み w 更新を何度も繰り返すことで、3状態ニューラルネットワークの学習を行う。

[0063] 3状態ニューラルネットワーク回路200は、ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて重み W_i を更新する、学習を繰り返す場合、学習終了後、丸め処理をしない状態で保っていた重み W_i を0にして、重みが疎なニューラルネットワークを生成する。

図16の上段に示すように、学習が終わってから、閾値を用いて重み W_i を0にする。

図16の下段の矢印に示すように、重みが疎なディープニューラルネットワークが生成される。

[0064] [実施形態の認識精度]

本実施形態の3状態ニューラルネットワークの認識精度向上について説明

する。

図 17 は、3 状態ニューラルネットワークの認識精度向上を説明する図である。

VGG11で評価した。画像はPascal VOC2007に準拠した。

ニューラルネットワークは、図 17 の左欄の処理、すなわち、Integer Conv2D、Binary Conv2D、Max Pooling、Binary Conv2D×3、Max Pooling、Binary Conv2D×3、Average Pooling およびFully Connect を実行することとした。

[0065] Integer Conv2Dを除く、Binary Conv2D、Max Pooling、Binary Conv2D×3、Max Pooling、Binary Conv2D×3、Average Pooling およびFully Connectを全て2値化 $\{-1, +1\}$ した場合、認識精度は、86.9%であった。

[0066] Binary Conv2D、Max Pooling、Binary Conv2D×3、Max Pooling およびBinary Conv2D×3を2値化 $\{-1, +1\}$ し、Average Pooling およびFully Connectを3状態 $\{-w, 0, +w\}$ した場合、認識精度は、93.47%であった。

[0067] Binary Conv2D、Max Pooling、Binary Conv2D×3、Max Pooling およびBinary Conv2D×2を2値化 $\{-1, +1\}$ し、Binary Conv2D、Average Pooling およびFully Connectを3状態 $\{-w, 0, +w\}$ した場合、認識精度は、97.29%であった。

[0068] [物体検出実証]

本実施形態の3状態ニューラルネットワーク回路と3値ニューラルネットワーク回路の組み合わせについて説明する。

図 18 は、3 状態ニューラルネットワーク回路と3値ニューラルネットワーク回路の組み合わせによる物体検出実証例を説明する図である。

物体検出ニューラルネットワークYOLOv2を3状態ニューラルネットワークで実現した実施例である。

図 18 に示すように、前段は3値化ニューラルネットワーク回路200A、メモリを介して、後段は3状態ニューラルネットワーク回路200とした

。

前段の3値化ニューラルネットワーク回路200Aは、ニューラルネットワークとしてTernaryCNN(Feature extraction)を用いた。後段の3状態ニューラルネットワーク回路200は、Half PrecisionCNN(Localization, classification)を用いた。3状態ニューラルネットワーク回路200は、16bit半精度浮動小数点を利用した。

[0069] 物体検出ニューラルネットワークY0L0v2を3状態ニューラルネットワークで実現した場合、下記を得た。

- (1) 3状態化しても認識精度は、ほぼ同一であることが判明した。
- (2) 重み W_i を約92%スキップする(“0”にすることができた。
- (3) ・メモリ量を92%削減することができた。その結果、36.50FPS (Frames Per Second) の認識速度を達成できた。

[0070] [他のデバイスとの比較例]

本実施形態の3状態ニューラルネットワーク回路と物体認識アルゴリズムY0L0v2との比較について説明する。

図19は、3状態ニューラルネットワーク回路と物体認識アルゴリズムY0L0v2との比較を説明する図である。

本発明の3状態ニューラルネットワーク回路と物体認識アルゴリズムY0L0v2とを比較した。

mCPU (ARM Cortex A57) , mGPU (Nvidia Pascal GPU) : Jetson TX2ボードを使用し、FPGAはZynq Ultrascale+ MPSoC zcu102ボードを使用した。

本発明の3状態ニューラルネットワーク回路は、ARM Cortex-A57と比較して、158.7倍速く、1.1倍少ない電力消費であり、電力性能効率は182.80倍優れていた。

また、本発明の3状態ニューラルネットワーク回路は、Nvidia Pascal 組み込みGPUと比較して、24.6倍高速であり、消費電力は2.0倍少なく、電力効率は49.38倍向上した。

[0071] [実装例]

図20は、本発明の実施形態に係る3状態ニューラルネットワーク回路の実装例を説明する図である。

<STEP1>

まず、与えられたデータセット（今回はImageNet、画像認識タスク用にデータ）を既存のディープニューラルネットワーク用のフレームワークソフトウェアであるChainer（登録商標）を用いてGPU（Graphics Processing Unit）を有するコンピュータ201上で学習を行った。学習は、GPU上で実行する。このコンピュータ201は、ARMプロセッサなどのCPU（Central Processing Unit）と、メモリと、ハードディスクなどの記憶手段（記憶部）と、ネットワークインタフェースを含むI/Oポートとを有する。このコンピュータは、CPU201が、メモリ上に読み込んだプログラム（2値化したニューラルネットワークの実行プログラム）を実行することにより、後記する各処理部により構成される制御部（制御手段）を動作させる。

[0072] <STEP2>

次に、自動生成ツールを用いて、本実施形態の3状態ニューラルネットワーク回路200と等価なC++コードを自動生成し、C++コード202を得た。

[0073] <STEP3>

次に、FPGAベンダの高位合成ツール(Xilinx社SDSoC)（登録商標）を用いて、FPGA（field-programmable gate array）合成用にHDL（hardware description language）203を生成した。例えば、高位合成ツール(Xilinx社SDSoC)では、実現したい論理回路をハードウェア記述言語（Verilog HDL/VHDL）を用いて記述し、提供されたCADツールでビットストリームに合成する。そして、FPGAにこのビットストリームを送信するとFPGAに回路が実現する。

[0074] <STEP4>

次に、従来のFPGA合成ツールVivado（登録商標）を用いて、FPGA上に実現（FPGA合成204）して画像認識タスクの検証を行った。

[0075] <STEP5>

検証後、基板205を完成させた。基板205には、2値化ニューラルネットワーク回路100がハードウェア化されて実装されている。

[0076] 以上説明したように、本実施形態に係る3状態ニューラルネットワーク回路200（図5参照）は、中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う非ゼロ畳み込み演算回路21と、畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路22（図2）と、総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路23と、を備える。非ゼロ畳み込み演算回路21は、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う。

[0077] この構成により、畳み込み演算部分の積和演算の絶対数を劇的に減らすことができ、メモリ量削減、計算時間の高速化を図ることができる。実際の例で確認した結果、畳み込み演算部分の計算時間が95%削減（1/20）となった。

この計算時間の高速化、メモリ面積の削減の効果を活かして初めて、メモリ量を削減してオンチップに配置が可能なニューラルネットワークが実現できた。

3状態ニューラルネットワーク回路200は、重み W が多ビット（例えば8～16ビットの $\pm w$ と“0”）であり、認識精度は高く用途（応用範囲）も広い利点を活かしつつ、メモリ量を劇的に減らして、オンチップ配置を実現した。オンチップ化（ニアメモリ）のメリットとして、広帯域（オフチップより約100倍高速）および低消費電力（オフチップより約33倍低い）の効果を奏する。

[0078] 3状態ニューラルネットワーク回路200は、ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて重み W_i を更新する、学習を繰り返す場合、学習終了後、丸め処理をしない状態で保っていた重み W_i を閾値を用いて0にし

て、重みが疎なニューラルネットワークを生成する。

- [0079] ディープニューラルネットワークのCNNにおいて、畳み込み演算を行う重み関数が疎（ゼロが多い）である特徴を活かして、それを相対アドレス表示して重みが非ゼロの箇所のみを演算処理する回路構成にする。3状態（0、±多ビット）ニューラルネットワークの計算量を大幅に削減する。

本発明を適用することにより、多ビットのアナログ量を扱う一般物体認識、意味的領域分割、自己位置推定などをリアルタイムで実現することが可能となる。

- [0080] また、3値化ニューラルネットワーク回路200A（図6参照）は、3値化により、2値化ニューラルネットワークと比較して約1～2%認識精度を改善しつつ、約95%削減（重みを0）にすることにより約3～4倍の高速化を達成することが可能となる。

- [0081] ちなみに、特許文献2の技術では、係数をゼロ除去するのみであり、非ゼロは多ビット（通常は32ビット浮動小数点精度）になっている。これに対して、本発明は、図12に示すように、係数をゼロにすることと低ビット化（1ビット）を同時に行うことで、メモリ量の大幅な削減を図ることができる。また、本発明のように、ニューラルネットワークを逆伝搬させて重み W_i を更新する、学習を繰り返す場合、学習終了後、丸め処理をしない状態で保っていた重み W_i を閾値を用いて0にして、重みが疎なニューラルネットワークを生成する技術的思想はない。

- [0082] 本発明は上記の実施形態例に限定されるものではなく、特許請求の範囲に記載した本発明の要旨を逸脱しない限りにおいて、他の変形例、応用例を含む。

- [0083] また、上記した実施形態例は本発明をわかりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、ある実施形態例の構成の一部を他の実施形態例の構成に置き換えることが可能であり、また、ある実施形態例の構成に他の実施形態例の構成を加えることも可能である。また、実施形態例は、その他の様

々な形態で実施されることが可能であり、発明の要旨を逸脱しない範囲で、種々の省略、置き換え、変更を行うことができる。これら実施形態やその変形例は、発明の範囲や要旨に含まれるとともに、特許請求の範囲に記載された発明とその均等の範囲に含まれる。

[0084] また、上記実施形態において説明した各処理のうち、自動的に行われるものとして説明した処理の全部または一部を手動的に行うこともでき、あるいは、手動的に行われるものとして説明した処理の全部または一部を公知の方法で自動的に行うこともできる。この他、上述文書中や図面中に示した処理手順、制御手順、具体的名称、各種のデータやパラメータを含む情報については、特記する場合を除いて任意に変更することができる。

また、図示した各装置の各構成要素は機能概念的なものであり、必ずしも物理的に図示の如く構成されていることを要しない。すなわち、各装置の分散・統合の具体的形態は図示のものに限られず、その全部または一部を、各種の負荷や使用状況などに応じて、任意の単位で機能的または物理的に分散・統合して構成することができる。

[0085] また、上記の各構成、機能、処理部、処理手段等は、それらの一部または全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行するためのソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリや、ハードディスク、SSD (Solid State Drive) 等の記録装置、または、IC (Integrated Circuit) カード、SD (Secure Digital) カード、光ディスク等の記録媒体に保持することができる。

また、上記実施の形態では、装置は、ニューラルネットワーク回路装置という名称を用いたが、これは説明の便宜上であり、名称はディープニューラルネットワーク回路、ニューラルネットワーク装置、パーセプトロン等であってもよい。また、方法およびプログラムは、ニューラルネットワーク処理方法という名称を用いたが、ニューラルネットワーク演算方法、ニューラル

ネットプログラム等であってもよい。

符号の説明

- [0086] 1 ディープニューラルネットワーク
- 2, 20 ニューラルネットワーク回路
- 11 入力層
- 12 隠れ層（中間層）
- 13 出力層
- 21, 21A 非ゼロ畳み込み演算回路（畳み込み演算回路）
- 22 総和回路
- 23, 221, 221A 活性化関数回路
- 200 3状態ニューラルネットワーク回路（ニューラルネットワーク回路装置）
- 200A 3値化ニューラルネットワーク回路（ニューラルネットワーク回路装置）
- 211 カウンタ
- 212, 217, 219 レジスタ
- 213 重み／アドレスメモリ（第1記憶手段）
- 214, 222 加算器
- 215 アドレス／データメモリ（第2記憶手段）
- 216 積算回路
- 216A XNORゲート回路
- 222, 222A Sequential MAC Unit
- Xi 入力値
- Wi 重み

請求の範囲

- [請求項1] 入力層、1 以上の中間層、および、出力層を少なくとも含むニューラルネットワーク回路装置であって、
- 前記中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う畳み込み演算回路と、畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路と、総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路と、を備え、
- 前記畳み込み演算回路は、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う
- ことを特徴とするニューラルネットワーク回路装置。
- [請求項2] 入力層、1 以上の中間層、および、出力層を少なくとも含むニューラルネットワーク回路装置であって、
- 前記中間層は、第1 中間層と、第2 中間層とを含み、
- 前記第1 中間層および前記第2 中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う畳み込み演算回路と、畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路と、総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路と、を備え、
- 前記第1 中間層の前記畳み込み演算回路は、3 状態 {−多ビット, 0, +多ビット} の重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行い、
- 前記第2 中間層の前記畳み込み演算回路は、3 値化 {−1, 0, +1} または2 値化 {−1, +1} の重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う
- ことを特徴とするニューラルネットワーク回路装置。

[請求項3] 前記畳み込み演算回路は、3状態 $\{-W(\text{多ビット}), 0, +W(\text{多ビット})\}$ の重み W_i と入力値 X_i との畳み込み演算を行うことを特徴とする請求項1に記載のニューラルネットワーク回路装置。

[請求項4] 前記畳み込み演算回路は、前記重み W_i が、下記式で示されることを特徴とする請求項1に記載のニューラルネットワーク回路装置。

[数3]

$$W_i = \begin{cases} W_{hid} & W_{hid} < \rho \\ 0 & -\rho \leq W_{hid} < \rho \\ W_{hid} & \rho \leq W_{hid} \end{cases} \quad \dots(3)$$

@0028

ただし、

W_{hid} : 多ビット重み W_i

ρ : 閾値

ことを特徴とする請求項1に記載のニューラルネットワーク回路装置。

[請求項5] 前記畳み込み演算回路は、3値化 $\{-1, 0, +1\}$ の重み W_i と入力値 X_i との畳み込み演算を行うことを特徴とする請求項1に記載のニューラルネットワーク回路装置。

[請求項6] 前記畳み込み演算回路は、2値化 $\{-1, +1\}$ の重み W_i と入力値 X_i との畳み込み演算を行うことを特徴とする請求項1に記載のニューラルネットワーク回路装置。

[請求項7] 前記畳み込み演算回路は、非ゼロの重みと畳み込みを行う入力値 X_i の相対アドレスを格納する第1記憶手段と、入力値 X_i のアドレス

を格納する第2記憶手段と、を有し、

ゼロの重みをスキップする演算の際、前記第1記憶手段から該当する非ゼロの重みとその相対アドレスを読み込み、

読み込んだ相対アドレスと現在のアドレスから前記第2記憶手段を参照して次の畳み込みを行う入力値 X_i を読み出し、

前記第2記憶手段から読み出した入力値 X_i と該当する非ゼロの重みをもとに畳み込み演算を行う

ことを特徴とする請求項1または請求項2に記載のニューラルネットワーク回路装置。

[請求項8]

前記畳み込み演算回路は、非ゼロの重みと畳み込みを行う入力値 X_i の絶対アドレスを格納する第3記憶手段を有し、

ゼロの重みをスキップする演算の際、前記第3記憶手段から絶対アドレスを指定して該当する非ゼロの重みと入力値 X_i を読み出し、

読み出した入力値 X_i と該当する非ゼロの重みをもとに畳み込み演算を行う

ことを特徴とする請求項1または請求項2に記載のニューラルネットワーク回路装置。

[請求項9]

ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて前記重み W_i を更新する、学習を繰り返す場合、

前記学習が終了するまでは少なくとも前記中間層の中では、前記重み W_i を丸め処理しない

ことを特徴とする請求項1または請求項2に記載のニューラルネットワーク回路装置。

[請求項10]

ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて前記重み W_i を更新する、学習を繰り返す場合、

前記学習終了後、丸め処理をしない状態で保っていた前記重み W_i

をゼロにして、重みが疎なニューラルネットワークを生成する

ことを特徴とする請求項 1 または請求項 2 に記載のニューラルネットワーク回路装置。

[請求項11] 入力層、1 以上の中間層、および、出力層を少なくとも含むニューラルネットワーク処理方法であって、

前記中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う畳み込み演算ステップと、

畳み込み演算された各演算値とバイアス W_0 との総和を取るステップと、

総和を取った信号 Y を活性化関数 $f(u)$ で変換するステップと、を有し、

前記畳み込み演算ステップでは、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う

ことを特徴とするニューラルネットワーク処理方法。

[請求項12] ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて前記重み W_i を更新する、学習を繰り返す場合、

前記学習が終了するまでは少なくとも前記中間層の中では、重み W_i を丸め処理しない

ことを特徴とする請求項 11 に記載のニューラルネットワーク処理方法。

[請求項13] ニューラルネットワークを順伝搬して得られた出力と学習データとの誤差を計算し、当該ニューラルネットワークを逆伝搬させて前記重み W_i を更新する、学習を繰り返す場合、

前記学習終了後、丸め処理をしない状態で保っていた前記重み W_i をゼロにして、重みが疎なニューラルネットワークを生成する

ことを特徴とする請求項 11 に記載のニューラルネットワーク処理

方法。

- [請求項14] 入力層、1 以上の中間層、および、出力層を少なくとも含むニューラルネットワーク回路装置としてのコンピュータを、
- 前記中間層の中で、畳み込みを行う入力値 X_i と、重み W_i を受け取り、畳み込み演算を行う畳み込み演算回路手段、
- 畳み込み演算された各演算値とバイアス W_0 との総和を取る総和回路手段、
- 総和を取った信号 Y を活性化関数 $f(u)$ で変換する活性化関数回路手段、として機能させ、
- 前記畳み込み演算回路手段は、重み W_i がゼロの重みをスキップし、非ゼロの重みと当該非ゼロの重みに対応する入力値 X_i をもとに畳み込み演算を行う
- ニューラルネットワークの実行プログラム。