(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2023年10月26日(26.10.2023)

WIPO | PCT

(10) 国際公開番号
WO 2023/204041 A1

(51) 国際特許分類:
G06N 3/0464 (2023.01) *G06F 17/10* (2006.01)

(21) 国際出願番号： PCT/JP2023/014229

(22) 国際出願日： 2023年4月6日(06.04.2023)

(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 2022-069885 2022年4月21日(21.04.2022) JP

(71) 出願人: 株式会社日立製作所(**HITACHI, LTD.**) [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目６番６号 Tokyo (JP).

(72) 発明者: 島村 光太郎(**SHIMAMURA, Kotaro**); 〒1008280 東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内 Tokyo (JP). 池田 尚弘(**IKEDA, Naohiro**); 〒1008280 東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内 Tokyo (JP). 近江 泰志(**OOMI, Taishi**); 〒1008280 東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内 Tokyo (JP).

(74) 代理人: 弁理士法人サンネクスト国際特許事務所(**SUNNEXT INTERNATIONAL PATENT OFFICE**); 〒1400002 東京都品川区東品川二丁目３番１２号 シーフォートスクエア センタービルディング１６階 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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,

(54) **Title:** INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, INFORMATION PROCESSING PROGRAM, SOFTWARE CREATION DEVICE, SOFTWARE CREATION METHOD, AND SOFTWARE CREATION PROGRAM

(54) 発明の名称： 情報処理装置、情報処理方法、情報処理プログラム、ソフトウェア作成装置、ソフトウェア作成方法、及びソフトウェア作成プログラム

図５

```
1  void layer_n(in, H, W, CI, k, CO, out)
2  {
3    convact_n(in, H, W, CI, k, CO, out)
4  }
5
6  void convact_n(in, H, W, CI, k, CO, out)
7  {
8    for(h=0;h<H;h++){
9      for(w=0;w<W;w++){
10       for(co=0;co<CO;co++){
11         d = 0;
12         for(ci=0;ci<CI;ci++){                          ┐
13           d += k[CI * co + ci] * in[CI * (W * h + w) + ci];  ├ CONVOLUTION-n
14         }                                              ┘
15         out[CO * (W * h + w) + co] = f(d);            ┐
16       }                                               ├ ACTIVATION FUNCTION-n
17     }                                                 ┘
18   }
19 }
```

(57) **Abstract:** The present invention reduces the time of neural network processing. This information processing device comprises a processor and a memory. The processor executes neural network processing including: convolution processing of executing multiply-accumulate operations between a plurality of sets of input data and a plurality of sets of weight data to generate a plurality of sets of intermediate data; and activation function processing of calculating the values of a predetermined function, with the plurality of sets of intermediate data serving as an input, to generate a plurality of sets of output data. The processor executes the activation function processing within a loop of the convolution processing.

(57) 要約：ニューラルネットワーク処理の処理時間を短縮する。情報処理装置は、プロセッサとメモリとを有する。プロセッサは、複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、を含むニューラルネットワーク処理を実行する。プロセッサは、活性化関数処理を、畳み込み処理のループ処理内で実行する。

WO 2023/204041 A1

［続葉有］

QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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))

明　細　書

発明の名称 ：

情報処理装置、情報処理方法、情報処理プログラム、ソフトウェア作成装置、ソフトウェア作成方法、及びソフトウェア作成プログラム

技術分野

[0001] 本発明は、情報処理装置、情報処理方法、情報処理プログラム、ソフトウェア作成装置、ソフトウェア作成方法、及びソフトウェア作成プログラムに関する。

背景技術

[0002] 人間の脳の神経細胞の構造を模したニューラルネットワークを用いて情報処理を行うニューラルネットワーク処理装置の第１の従来例として、特許文献１に記載されたものがある。このニューラルネットワーク処理装置は、複数の入力データと複数の重みデータの間の積和演算を行い複数の中間データを生成する畳み込み処理と、複数の中間データを入力として予め定められた関数を計算し複数の出力データを生成する活性化関数処理を実行するレイヤを複数組み合わせて画像等の認識を行う。また、この従来例にはＧＰＵ（Graphics Processing Unit）と呼ばれる多数の演算器を内蔵した回路を使用して畳み込み処理を高速に実行することが記載されている。

[0003] また、ニューラルネットワーク処理装置の第２の従来例として、特許文献２に記載されたものがある。このニューラルネットワーク処理装置は、行列演算部と活性化関数演算部を有する専用の推論演算部を設けることで高速に処理を行うことを可能としている。

先行技術文献

特許文献

[0004] 特許文献1：特開２０１９－８７０７２号公報

特許文献2：特開２０２０－１１２９０１号公報

発明の概要

発明が解決しようとする課題

[0005] ニューラルネットワーク処理装置は、例えば自動車の自動運転装置でカメラで撮影した周囲の画像の認識処理を行うなど、演算リソースの限られた環境で使用する用途が増加している。また、自動車の自動運転装置などでは周囲の広い範囲に対する認識処理が必要なうえ、認識結果の誤りが事故に直結する可能性があるため、高い認識精度が求められる。その様な背景から、ニューラルネットワーク処理は複雑な構成となってきており、かつ、多数の画像に対する認識処理を行う必要があるため、演算量が膨大となる。限られた演算リソースで膨大な演算量の処理を行うと処理時間が期待されるものより多くなり、必要な処理性能が得られないという問題がある。

[0006] 上記第１の従来例はＧＰＵを用いて処理時間を短縮しているが、汎用のハードウェア上のソフトウェアの処理では処理時間短縮の効果は限定的であり、必要な処理性能を達成できないケースが多い。上記第２の従来技術は専用の推論演算部を設けていることから、ＧＰＵを使用するより処理時間を短縮できる可能性があるが、専用の推論演算部を開発するためには膨大な開発コストがかかるため、出荷数量の少ない装置に適用するのは困難である。従って、汎用のハードウェア上のソフトウェアでニューラルネットワーク処理を行う場合の処理時間を短縮したいというニーズがある。

[0007] 本発明はこの様な状況に対応するため、ニューラルネットワーク処理の処理時間を短縮することを目的とする。

課題を解決するための手段

[0008] 上述した課題を解決するため、本発明の情報処理装置は、プロセッサとメモリとを有する情報処理装置であって、前記プロセッサは、複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、を含むニューラルネットワーク処理を実行し、前記活性化関数処理を、前記畳み込み処理のループ処理内で実行することを特徴とする。

発明の効果

[0009] 本発明によれば、中間データの容量が削減されるため、中間データをプロセッシングユニットの内部に格納することが可能となり、プロセッシングユニットと外部メモリとの間のデータ転送が削減されるため、ニューラルネットワーク処理の処理時間を短縮することができる。

図面の簡単な説明

[0010] [図1]実施形態１に係るニューラルネットワーク処理装置の構成を示す図である。

[図2]ニューラルネットワーク処理の構成を示す図である。

[図3]第１の従来技術に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。

[図4]第１の従来技術に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。

[図5]実施形態１に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。

[図6]実施形態１に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。

[図7]第２の従来技術に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。

[図8]第２の従来技術に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。

[図9]実施形態２に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。

[図10]実施形態２に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。

[図11]実施形態３に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。

[図12]実施形態３に係るニューラルネットワーク処理のアルゴリズムの実行

時の処理態様を示す図である。

[図13]実施形態４に係る積和演算処理を行うアルゴリズムのバリエーションと対応するパラメータの例を示す図である。

[図14]実施形態４に係る活性化関数処理を行うアルゴリズムのバリエーションと対応するパラメータの例を示す図である。

[図15]実施形態４に係るニューラルネットワーク処理の各レイヤに対する積和演算処理と活性化関数処理のパラメータの組み合わせの例を示す図である。

[図16]実施形態４に係るパラメータの組み合わせの中から重複を除いたパラメータの組み合わせを選択した結果を示す図である。

[図17]実施形態４に係るソフトウェア作成処理を実行するコンピュータのハードウェア例を示す図である。

[図18]実施形態４に係るソフトウェア作成処理を示すフローチャートである。

発明を実施するための形態

[0011] 以下、図面を参照して本願開示に係る実施形態を説明する。以下の実施形態は、図面も含め本発明を説明するための例示であって、説明の明確化のため、適宜、省略及び簡略化がなされている。本発明は、他の種々の形態でも実施することが可能である。特に限定しない限り、各構成要素は単数でも複数でもよい。

[0012] 以下の実施形態において、プログラムによって実行される処理を説明する場合がある。サーバやクライアントといった計算機は、プロセッサ（例えばＣＰＵ（Central Processing Unit）、ＧＰＵ）によりプログラムを実行し、記憶資源（例えばメモリ）やインターフェースデバイス（例えば通信ポート）等を用いながら、プログラムで定められた処理ステップを行う。そのため、プログラムを実行して行う処理の主体を、プロセッサとしてもよい。同様に、プログラムを実行して行う処理の主体が、プロセッサを有するコントローラ、装置、システム、計算機、ノードでもよい。プログラムを実行して

行う処理の主体は、演算部であればよく、特定の処理を行う専用回路を含んでいてもよい。ここで、専用回路とは、例えばＦＰＧＡ（Field Programmable Gate Array）やＡＳＩＣ（Application Specific Integrated Circuit）、ＣＰＬＤ（Complex Programmable Logic Device）、量子コンピュータ等である。

[0013] プログラムは、プログラムソースから計算機にインストールされてもよい。プログラムソースは、例えば、プログラム配布サーバ又は計算機が読み取り可能な記憶メディアでもよい。プログラムソースがプログラム配布サーバの場合、プログラム配布サーバはプロセッサと配布対象のプログラムを記憶する記憶資源を含み、プログラム配布サーバのプロセッサが配布対象のプログラムを他の計算機に配布してもよい。また、実施形態において、２以上のプログラムが１つのプログラムとして実現されてもよいし、１つのプログラムが２以上のプログラムとして実現されてもよい。

[0014] 以下の実施形態において、「ＹＹＹ（処理）部」、「ＹＹＹステップ」、及び「ＹＹＹ手順」は、装置（またはシステム）、方法、及びプログラムの異なるカテゴリのそれぞれ対応する構成要素である。例えば“ＹＹＹステップ”を実行するプロセッサ又はコンピュータは、“ＹＹＹ処理部”に対応する。また、プロセッサ又はコンピュータに実行させた“ＹＹＹ手順”は、“ＹＹＹステップ”に対応する。

[0015] 以下の実施形態において、同一あるいは同様の機能を有する構成要素が複数ある場合には、同一の符号に異なる添字を付して説明する場合がある。また、これらの複数の構成要素を区別する必要がない場合には、添字を省略して説明する場合がある。

[0016] 以下の実施形態においては、「ソフトウェア」は、プログラム及びライブラリなどのその他のファイルを含むコンポーネントであっても、プログラムそのものであってもよい。

[0017] ［実施形態１］

（実施形態１に係る情報処理装置１０の構成）

図１は、実施形態１に係る情報処理装置１０の構成を示す図である。情報処理装置１０は、マイクロプロセッサ１１、ＲＯＭ（Read Only Memory）１２、ＲＡＭ（Random Access Memory）１３，１５、及びＧＰＵ１４を含んで構成される。

[0018] マイクロプロセッサ１１は、ＣＰＵ１１ａ、及びインタフェース回路１１ｂ１，１１ｂ２，１１ｂ３，１１ｂ４，１１ｂ５を含んで構成される。

[0019] ＲＯＭ１２は、インタフェース回路１１ｂ１を介してマイクロプロセッサ１１と接続される。ＲＡＭ１３は、インタフェース回路１１ｂ２を介してマイクロプロセッサ１１と接続される。ＧＰＵ１４は、ＲＡＭ１５と接続される。また、ＧＰＵ１４は、インタフェース回路１１ｂ３を介してマイクロプロセッサ１１と接続される。

[0020] カメラ１１１ａ，１１１ｂは、インタフェース回路１１ｂ４を介してマイクロプロセッサ１１と接続される。ディスプレイ装置１１２は、インタフェース回路１１ｂ５を介してマイクロプロセッサ１１と接続される。

[0021] 情報処理装置１０は、カメラ１１１ａ，１１１ｂから画像データを入力し、ニューラルネットワーク処理による認識を行った後、認識結果をディスプレイ装置１１２に表示する。認識結果は、スピーカから音声出力されても良い。なお、本実施形態では画像認識を行う例を示すが、画像認識に限定されず、音声認識などの他のニューラルネットワーク処理全般に適用が可能である。

[0022] カメラ１１１ａ，１１１ｂは、それぞれ異なる方向の画像を撮影する。これによって、認識したい領域の全ての画像を取り込んで認識処理を行うことが可能となる。また、カメラ１１１ａ，１１１ｂとして画素数が多く撮影範囲の広いカメラを使用しても良い。これによって、認識したい領域が広い場合でも少ないカメラの台数で全ての領域をカバーすることが可能となる。この場合、１つのカメラで撮影した画像を複数の領域に分割して別々に認識処理を行っても良い。これによって、空などの認識が不要な領域を除外して必要な領域のみに認識処理を行うことが可能となる。

[0023] マイクロプロセッサ１１は、ＣＰＵ１１ａ、及びインタフェース回路１１ｂ１，１１ｂ２，１１ｂ３，１１ｂ４，１１ｂ５を１つのチップに集積したＬＳＩ（Large Scale Integration）である。この構成は一例であり、ＲＯＭ１２、ＲＡＭ１３、ＧＰＵ１４、及びＲＡＭ１５の一部又は全部をマイクロプロセッサ１１内に内蔵しても良い。

[0024] ＣＰＵ１１ａは、インタフェース回路１１ｂ１を経由してＲＯＭ１２に格納されているソフトウェアを読み込んで実行する。なお、ＲＯＭ１２は、ＲＡＭ１３に比べると読書きの速度が低速である場合が多いため、起動時にＲＯＭ１２からＲＡＭ１３にソフトウェアをコピーし、その後はＲＡＭ１３からソフトウェアを読み込む様にしても良い。ＣＰＵ１１ａは、ＲＯＭ１２、又はＲＡＭ１３から読み込んだソフトウェアに従い、以下の一連の処理を行う。

[0025] 先ずＣＰＵ１１ａは、インタフェース回路１１ｂ４を経由してカメラ１１１ａ，１１１ｂから画像データを取得し、インタフェース回路１１ｂ２を経由してＲＡＭ１３に格納する。次にＣＰＵ１１ａは、インタフェース回路１１ｂ２を経由してＲＡＭ１３に格納されている画像データを読み出し、インタフェース回路１１ｂ３を経由してＧＰＵ１４に転送する。

[0026] 次にＣＰＵ１１ａは、インタフェース回路１１ｂ１経由してＲＯＭ１２、もしくはインタフェース回路１１ｂ２を経由してＲＡＭ１３に格納されているソフトウェアを読み出し、インタフェース回路１１ｂ３を経由してＧＰＵ１４に転送する。そしてＣＰＵ１１ａは、ＧＰＵ１４に対してソフトウェアによる演算開始を指示する。

[0027] 次にＣＰＵ１１ａは、インタフェース回路１１ｂ３を経由してＧＰＵ１４から演算終了の通知を受け取ると、インタフェース回路１１ｂ３を経由してＧＰＵ１４から演算結果を取得し、インタフェース回路１１ｂ２を経由してＲＡＭ１３に格納する。

[0028] 次にＣＰＵ１１ａは、インタフェース回路１１ｂ２を経由してＲＡＭ１３からＧＰＵ１４の演算結果を読み込み、所定の処理を行ったうえでインタフ

ェース回路１１ｂ５を経由してディスプレイ装置１１２に表示する。

[0029] ＧＰＵ１４は、インタフェース回路１１ｂ３を経由してＣＰＵ１１ａから画像データを受け取ると、ＲＡＭ１５に格納する。また、ＧＰＵ１４は、インタフェース回路１１ｂ３を経由してＣＰＵ１１ａからソフトウェアを受け取ると、そのソフトウェアを実行し、演算結果をＲＡＭ１５に格納する。

[0030] また、ＧＰＵ１４は、インタフェース回路１１ｂ３を経由してＣＰＵ１１ａから演算結果の読み出し要求を受け取ると、ＲＡＭ１５から演算結果を読み出してインタフェース回路１１ｂ３を経由してＣＰＵ１１ａに出力する。

[0031]（ニューラルネットワーク処理）

図２は、ニューラルネットワーク処理の構成を示す図である。なお、図２は単一の入力画像に対する認識処理を記載したものであり、複数の画像に対して認識処理を行う場合には入力画像毎に図２の処理を実行する。その際、処理の内容は全ての画像に対して同一の構成であっても良いし、認識処理の目的に合わせて画像毎に異なる構成であっても良い。

[0032] 図２に示すニューラルネットワーク処理は、Ｎ層のレイヤ（Layer）ｉ（レイヤ１（Layer１）、レイヤ２（Layer２）、・・・、レイヤＮ（LayerＮ））から構成される。レイヤ１は入力画像（Input Image）を入力し、処理結果をレイヤ２に出力する。レイヤ（ｉ＋１）（ｉ＝１，２，・・・，Ｎ－２）は、１つ前のレイヤｉからデータを受け取り、処理結果をレイヤ（ｉ＋２）に出力する。レイヤＮは、１つ前のレイヤ（Ｎ－１）からデータを受け取り、処理結果を、入力画像の認識結果として出力する。

[0033] Ｎ個の各レイヤｉは、畳み込み処理（Convolution）と活性化関数処理（Activation Function）からなる。畳み込み処理と活性化関数処理の詳細は後述する。

[0034]（比較例である第１の従来技術に係るニューラルネットワーク処理のレイヤ処理）

ここで、実施形態１の比較例である第１の従来技術に係るニューラルネットワーク処理ついて説明する。図３は、第１の従来技術に係るニューラルネ

ットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。第１の従来技術のニューラルネットワーク処理を実行するハードウェアは、実施形態１の情報処理装置１０と同様である。

[0035] ＣＰＵ１１ａは、図３に示す第ｎ層（ｎ＝１，２，・・・，Ｎ）のレイヤ処理（layer_n()）を実行する際、７～２０行目の畳み込み処理（convolution_n()）、及び、２２～３２行目の活性化関数処理（activation_n()）を起動する。畳み込み処理（convolution_n()）では、入力データ（ｉｎ［］）と重みデータ（ｋ［］）の間の積和演算を行い、中間データ（ｄ［］）が生成される。活性化関数処理（activation_n()）では、中間データ（ｄ［］）を入力として、予め定められた所定の関数（ｆ()）の値を計算し出力データ（ｏｕｔ［］）が生成される。

[0036] convolution_n()及びactivation_n()はＧＰＵ１４で実行される。これに限らず、convolution_n()及びactivation_n()はＣＰＵ１１ａで実行されてもよい。

[0037] convolution_n()及びactivation_n()の引数“H”は、画像の縦方向の画素数を示す。図３の９～１９行目の変数ｈに関するループ処理は、画像の縦方向の座標ごとの処理を示す。また、引数“W”は、画像の横方向の画素数を示す。図３の１０～１８行目の変数“ｗ”に関するループ処理は画像の横方法の座標ごとの処理を示す。

[0038] また、引数“ＣＯ”は、出力データの各画素の属性情報の種類数を示す。図３の１１～１７行目の変数“ｃｏ”に関するループ処理は、出力データの単一画素の属性情報の種類毎の処理を示す。引数“ＣＯ”の値は、数十～数百となる場合が多い。

[0039] また、引数“ＣＩ”は、入力データの各画素の属性情報の種類数を示す。図３の１３～１６行目の変数“ｃｉ”に関するループ処理は、入力データの単一画素の属性情報の種類毎の処理を示す。引数“ＣＩ”の値は、入力画像を入力データとするレイヤ１では３（Ｒ（赤），Ｇ（緑），Ｂ（青）の３原色に相当）となるのが一般的である。レイヤ１以外では１つ前のレイヤの出

力を入力とするため、数十～数百となる場合が多い。

[0040] また、引数“ｋ”は、処理対象のレイヤｎの重みデータである。

[0041] 図４は、第１の従来技術に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。図４に示すように、畳み込み処理（convolution_n()）では、入力データ（ｉｎ［］）と重みデータ（ｋ［］）をメモリから読み出し、中間データｄ［］をメモリに書き込む。

[0042] メモリには、プロセッシングユニット（本実施形態ではＧＰＵ１４）に内蔵されたキャッシュメモリとプロセッシングユニットの外部に接続された外部メモリ（本実施形態ではＲＡＭ１５）がある。例えば画像の認識処理などを行う場合には、入力データ（ｉｎ［］）や中間データ（ｄ［］）は容量が大きいため、キャッシュメモリには格納することができず、外部メモリへのアクセスが発生する。

[0043] 本願発明者は、プロセッシングユニットと外部メモリとの間のデータ転送は、キャッシュメモリに比べると時間がかかり、外部メモリとの間のデータ転送の処理時間が演算処理そのものに比して無視できない量となっていることを見出した。活性化関数処理（activation_n()）も同様であり、プロセッシングユニットと外部メモリとの間のデータ転送の処理時間が演算処理そのものに比して無視できない量となっていた。

[0044] すなわち、第１の従来技術では、畳み込み処理（convolution_n()）の入力データ（ｉｎ［］）の読み出し、中間データ（ｄ［］）の書き込み、活性化関数処理（activation_n()）の中間データ（ｄ［］）の読み出し、及び、出力データ（ｏｕｔ［］）の書き込みで、ＧＰＵ１４とＲＡＭ１５との間のデータ転送が発生し、ＧＰＵ１４といった汎用のハードウェア上のソフトウェアで行うニューラルネットワーク処理のボトルネックとなっていることが見い出された。

[0045]（実施形態１に係るニューラルネットワーク処理のレイヤ処理）

図５は、実施形態１に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。ＣＰＵ１１ａは、図５に示すレイヤ

処理（layer_n()）を実行する際、図３と比較して、畳み込み処理と活性化関数処理を融合した処理（convact_n()）を起動するのみである。

[0046] 図５に示すように、１０～１６行目の変数“ｃｏ”に関するループ処理の中で、１２～１４行目の変数“ｃｉ”に関するループ処理を終了する毎に、中間データ（ｄ）に対して１５行目の活性化関数処理を実施して出力データ（ｏｕｔ［］）が生成される。

[0047] 図６は、実施形態１に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。図５に示した実施形態１に係るニューラルネットワーク処理のアルゴリズムは、図３に示した第１の従来技術のアルゴリズムとは異なり、中間データが配列ではない一次元の変数ｄに格納されているため、ＧＰＵ１４の内部（例えばキャッシュメモリ）に保持することが可能となり、ＲＡＭ１５との間のデータ転送が不要となる。このため、ＧＰＵ１４とＲＡＭ１５との間のデータ転送量が削減され、処理時間を短縮することができる。

[0048] ［実施形態２］

（比較例である第２の従来技術に係るニューラルネットワーク処理のレイヤ処理）

先ず、実施形態２の比較例である第２の従来技術に係るニューラルネットワーク処理ついて説明する。図７は、第２の従来技術に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。図３の第１の従来技術に係るアルゴリズムでは、畳み込み処理は積和演算のみで構成されている。一方、図７の第２の従来技術に係るアルゴリズムでは、図７の１３～３８行目の畳み込み処理に２８～３８行目のバイアス値加算処理が追加されている。バイアス値加算処理は、Ｎ層のレイヤからなるニューラルネットワークの各層毎のバイアス値を積和演算の演算結果に加算する処理である。これによって、認識結果が向上するケースがあることが知られている。

[0049] 図８は、第２の従来技術に係るニューラルネットワーク処理のアルゴリズ

ムの実行時の処理態様を示す図である。図４と比較して、バイアス加算の入力データ（ｂ［］、ｄ１［］）のＲＡＭ１５からの読み出し、及び、バイアス加算の演算結果（ｄ２［］）のＲＡＭ１５への書き込みの処理が追加されていることから、ＧＰＵ１４とＲＡＭ１５との間のデータ転送の処理時間が増加する。

[0050]（実施形態２に係るニューラルネットワーク処理のレイヤ処理）

図９は、実施形態２に係るニューラルネットワーク処理の１つのレイヤを処理するアルゴリズムを示す図である。本実施形態では、図７の積和演算（図７の１３～２６行目のmultiadd_n()）、バイアス加算（図７の２８～３８行目のBiasadd_n()）、活性化関数処理（図７の４０～５０行目のactivation_n()）を１つのループ処理（図９の８～１９行目）で実行する。

[0051] 図９に示すように、１０～１７行目の変数“ｃｏ”に関するループ処理の中で、１２～１４行目の変数“ｃｉ”に関するループ処理を終了する毎に、１５行目のように中間データ（ｄ１）にバイアス値（ｂ［］）を加算して中間データ（ｄ２）が計算される。そして、１６行目のように、中間データ（ｄ２）に対して１５行目の活性化関数処理を実施して出力データ（ｏｕｔ［］）が生成される。

[0052] 図１０は、実施形態２に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。図９に示した実施形態２に係るニューラルネットワーク処理のアルゴリズムは、図７に示した第２の従来技術のアルゴリズムとは異なり、２種類の中間データが配列ではない一次元の変数ｄ１，ｄ２にそれぞれ格納されている。このため、中間データｄ１，ｄ２をＧＰＵ１４の内部（例えばキャッシュメモリ）に保持することが可能となり、ＲＡＭ１５との間のデータ転送が不要となる。よって、ＧＰＵ１４とＲＡＭ１５との間のデータ転送量が削減され、処理時間を短縮することができる。

[0053]［実施形態３］

（実施形態３に係るニューラルネットワーク処理のレイヤ処理）

図１１は、実施形態３に係るニューラルネットワーク処理の１つのレイヤ

を処理するアルゴリズムを示す図である。実施形態３に係るニューラルネットワーク処理のアルゴリズムでは、図１１に示すように、図９の実施形態２のアルゴリズムとは異なり、積和演算のみ異なるループ処理（８～１９行目）で実行し、バイアス値加算処理と活性化関数処理を１つのループ処理（２４～３２行目）で実行する。

[0054] 一般に積和演算は演算量が多く、ニューラルネットワーク処理全体に占める処理時間の比率が多いため、ＧＰＵメーカなどが提供する最適化したライブラリを使用して処理時間を短縮する場合が多い。図９の実施形態２のアルゴリズムの場合は、積和演算と同一のループ処理にバイアス値加算処理や活性化関数処理が組み込まれているため積和演算のライブラリを使用することができず、積和演算の処理時間がライブラリを使用する場合より長くなる可能性がある。一方、図１１の本実施形態のアルゴリズムの場合、積和演算の処理部分のみライブラリに置き換えることが可能となり、積和演算の処理時間増加を防止することができる。

[0055] 図１２は、実施形態３に係るニューラルネットワーク処理のアルゴリズムの実行時の処理態様を示す図である。本実施形態に係るニューラルネットワーク処理は、図８の第２の従来技術と比較して、バイアス値加算処理（biasadd-n()）の処理結果（ｄ２）のメモリへの書き込みとメモリからの読み出しが不要となるため、ＧＰＵ１４とＲＡＭ１５との間のデータ転送量が削減され、処理時間を短縮することができる。また、本実施形態に係るニューラルネットワーク処理は、図１０の実施形態２と比較して、積和演算（multiadd-n()）の処理結果（ｄ１［］）のメモリへの書き込みと読み出しの処理が増えるため、ＧＰＵ１４とＲＡＭ１５との間のデータ転送量は増加し処理時間も増加する。しかし、積和演算（multiadd-n()）にライブラリを使用することが可能となり、積和演算の処理時間の増加を防止できるため、図１０の実施形態２と比較して、トータルの処理時間を削減できる可能性がある。

[0056] ［実施形態４］

本願開示に係る実施形態では、畳み込み処理と活性化関数処理を１つのル

ープ処理で実行する。このため、畳み込み処理と活性化関数処理にそれぞれ複数のバリエーションがあると、ニューラルネットワーク処理のソフトウェアの作成時に、両者のバリエーションの数の積に相当する数の処理を記述する必要があり、ソフトウェアの作成や動作確認に膨大な時間がかかる可能性がある。

[0057] 畳み込み処理に関しては、その一部である積和演算処理のアルゴリズムに多数のバリエーションがあることが知られている。図１３は、実施形態４に係る積和演算処理を行うアルゴリズムのバリエーションと対応するパラメータの例を示す図である。

[0058] 図１３における“INPUT FORM（入力形式）”は、入力データの並び順を示すもので、“ＣＨＷ”と“ＨＷＣ”の２通りがある。“ＣＨＷ”は、まず属性情報の種類の軸でデータを並べ、次に属性情報の種類の軸の値が同じデータ同士で画像の縦方向の軸でデータを並べ、最後に属性情報の種類の軸と画像の縦方向の軸の値が同じデータ同士で画像の横方向の軸でデータを並べる。

[0059] “ＨＷＣ”は、まず画像の縦方向の軸でデータを並べ、次に画像の縦方向の軸の値が同じデータ同士で画像の横方向の軸でデータを並べ、最後に画像の縦方向の軸と画像の横方法の軸の値が同じデータ同士で属性情報の種類の軸でデータを並べる。

[0060] なお、図３、図５、図７、図９、及び図１１に示したアルゴリズムは、入力形式が“ＨＷＣ”に対応したソフトウェアであり、入力形式が“ＣＨＷ”の場合には適用できない。

[0061] 図１３における“WEIGHT FORM（重み形式）”は、重みデータの並び順を示すもので、“ＣＨＷ”と“ＨＷＣ”の２通りがある。“ＣＨＷ”と“ＨＷＣ”の意味は、入力形式と同様である。図３、図５、図７、図９、及び図１１に示したアルゴリズムは、重み形式が“ＨＷＣ”に対応したソフトウェアであり、重み形式が“ＣＨＷ”の場合には適用できない。

[0062] 図１３における“OUTPUT FORM（出力形式）”は、出力データの並び順を

示すもので、“ＣＨＷ”と“ＨＷＣ”の２通りがある。“ＣＨＷ”と“ＨＷＣ”の意味は、入力形式と同様である。図３、図５、図７、図９、及び図１１に示したアルゴリズムは、出力形式が“ＨＷＣ”に対応したソフトウェアであり、出力形式が“ＣＨＷ”の場合には適用できない。

[0063] 図１３における“WEIGHT SIZE（重みサイズ）”は、出力データの各画素に影響を与える入力データの画素の範囲を示す。図３、図５、図７、図９、及び図１１に示したアルゴリズムでは、出力データの各画素は入力データのうちの縦方向と横方向の軸の値が同じ画素からのみ影響を受ける。これを図１３では“１ｘ１”と表記した。これ以外に、縦方向と横方向の軸の値が同じ画素に加えて上下１画素ずつ範囲を広げた３画素四方の入力データから影響を受ける計算方法もあり、これを図１３で“３ｘ３”と表記した。

[0064] 図１３における“ＣＩ”は、入力データの各画素の属性情報の種類数であり、図３で説明した通り、レイヤ１では３、それ以外では数十～数百の値を取りうる。ここでは、スペースの都合で“３”と“３２”の２通りのみ記載した。

[0065] 図１３における“ＣＯ”は、出力データの各画素の属性情報の種類数であり、図３で説明した通り、数十～数百の値を取りうる。ここでは、スペースの都合で“３２”の１通りのみ記載した。

[0066] 図１３で積和演算処理を行うアルゴリズムのバリエーションの全てを網羅しているとは限らないが、それでも３２通りのバリエーションが存在する。

[0067] 図１４は、実施形態４に係る活性化関数処理を行うアルゴリズムのバリエーションと対応するパラメータの例を示す図である。図１４には、それぞれの活性化関数の名称を記載し、関数の計算式は省略している。図１４でバリエーションの全てを網羅しているとは限らないが、それでも１０通りのバリエーションが存在する。

[0068] 図１３に示す積和演算処理のバリエーションと、図１４に示す活性化関数処理のバリエーションを掛け合わせると３２０通りとなる。これは、畳み込み処理と活性化関数処理を１つのループ処理で実行するためには、３２０通

りの処理を記述することが必要となることを意味する。このため、図１３のバリエーションを持つ積和演算処理及び図１４のバリエーションを持つ活性化関数処理を含むニューラルネットワーク処理のソフトウェアの作成や動作確認に膨大な時間がかかると考えられる。

[0069] この課題に対応するため、アプリケーションに依存して必要な組み合わせのみ処理を記述する方法が考えられる。以下にその開発フローを説明する。

[0070] 図１５は、実施形態４に係るニューラルネットワーク処理の各レイヤに対する積和演算処理と活性化関数処理のパラメータの組の例を示す図である。複雑な認識処理を行う場合には、１００以上のレイヤを実行する場合も多いが、ここではスペースの都合でレイヤ数が３０の例を記載した。図１５からわかる通り、パラメータの組み合わせが同一のレイヤが複数存在する。

[0071] 図１６は、実施形態４に係るパラメータの組み合わせの中から重複を除いたパラメータの組み合わせを選択した結果を示す図である。図１６に示すように、図１５のパラメータの組み合わせの中から重複を除いたパラメータの組み合わせを選択した結果として、パラメータの組み合わせは３種類に絞られることが確認できる。よって、畳み込み処理と活性化関数処理を１つのループ処理で実行する場合、３通りのソフトウェアを記述すれば良いことがわかる。

[0072] ＧＰＵメーカがライブラリを提供する場合、畳み込みと活性化関数のどの組み合わせをユーザが使用するかが不明のため、３２０通りの全てのバリエーションに対応する記述のソフトウェアを作成する必要がある。この様な膨大なバリエーションがボトルネックとなり、ライブラリの作成が困難となる可能性がある。一方、ユーザの立場で本願開示に係る実施形態を実施する場合、アプリケーションに依存して必要な組み合わせのみ処理を記述すれば良いため、図１６に従って例えば３通りのパラメータの組み合わせに絞ってソフトウェアを作成すれば良いので、ソフトウェアの記述作成に要する時間を大幅に短縮することが可能である。

[0073] （実施形態４に係るソフトウェア作成方法を実行するコンピュータ１００の

ハードウェア）

図１７は、実施形態４に係るソフトウェア作成方法を実行するコンピュータ１００の構成例を示すハードウェア図である。

[0074] コンピュータ１００は、バス等の内部通信線１０９を介して相互に接続されたＣＰＵをはじめとするプロセッサ１０１、主記憶装置１０２、補助記憶装置１０３、ネットワークインタフェース１０４、入力装置１０５、及び出力装置１０６を備えるコンピュータである。

[0075] プロセッサ１０１は、コンピュータ１００全体の動作制御を司る。また主記憶装置１０２は、例えば揮発性の半導体メモリから構成され、プロセッサ１０１のワークメモリとして利用される。補助記憶装置１０３は、ハードディスク装置、ＳＳＤ（Solid State Drive）、またはフラッシュメモリ等の大容量の不揮発性の記憶装置から構成され、各種プログラムやデータを長期間保持するために利用される。

[0076] 補助記憶装置１０３に格納されたソフトウェア作成プログラム１０３ａがコンピュータ１００の起動時や必要時に主記憶装置１０２にロードされ、主記憶装置１０２にロードされたソフトウェア作成プログラム１０３ａをプロセッサ１０１が実行することにより、ソフトウェア作成方法を実行するソフトウェア作成装置が実現される。

[0077] なお、ソフトウェア作成プログラム１０３ａは、非一時的記録媒体に記録され、媒体読み取り装置によって非一時的記録媒体から読み出されて、主記憶装置１０２にロードされてもよい。または、ソフトウェア作成プログラム１０３ａは、ネットワークを介して外部のコンピュータから取得されて、主記憶装置１０２にロードされてもよい。

[0078] ネットワークインタフェース１０４は、コンピュータ１００をシステム内の各ネットワークに接続する、あるいは他のコンピュータと通信するためのインタフェース装置である。ネットワークインタフェース１０４は、例えば、有線ＬＡＮ（Local Area Network）や無線ＬＡＮ等のＮＩＣ（Network Interface Card）から構成される。

[0079] 入力装置１０５は、キーボードや、マウス等のポインティングデバイス等から構成され、ユーザがコンピュータ１００に各種指示や情報を入力するために利用される。出力装置１０６は、例えば、液晶ディスプレイ又は有機ＥＬ（Electro Luminescence）ディスプレイ等の表示装置や、スピーカ等の音声出力装置から構成され、必要時に必要な情報をユーザに提示するために利用される。

[0080]（実施形態４に係るソフトウェア作成処理）

図１８は、実施形態４に係るソフトウェア作成処理を示すフローチャートである。

[0081] 先ずステップＳ１１では、コンピュータ１００は、対象のニューラルネットワークの複数のレイヤ処理のそれぞれについて畳み込み処理と活性化関数処理のパラメータの組み合わせのリストを作成する第１ステップを実行する。

[0082] 次にステップＳ１２では、コンピュータ１００は、ステップＳ１１で作成したパラメータのリストの中から重複を除外したパラメータの組み合わせを選択する第２ステップを実行する。

[0083] 次にステップＳ１３では、コンピュータ１００は、作成するソフトウェアが推論用か否かを判定する。コンピュータ１００は、作成するソフトウェアが推論処理用である場合（ステップＳ１３ＹＥＳ）にステップＳ１４へ処理を移し、学習処理用である場合（ステップＳ１３ＮＯ）にステップＳ１５へ処理を移す。

[0084] ステップＳ１４では、コンピュータ１００は、ステップＳ１３によって選択されたパラメータの組み合わせに対応する畳み込み処理と活性化関数処理に関して、活性化関数処理を、畳み込み処理のループ処理内で実行するプログラムを作成する第３ステップを実行する。

[0085] 一方、ステップＳ１５では、コンピュータ１００は、ステップＳ１３によって選択されたパラメータの組み合わせに対応する畳み込み処理と活性化関数処理に関して、活性化関数処理を、畳み込み処理とは異なるループ処理内

で実行するプログラムを作成する第４ステップを実行する。

[0086] なお、上記ステップＳ１１からＳ１５の一部または全てを人手で実行してもよい。

[0087] 本実施形態によれば、ニューラルネットワーク処理のソフトウェアの作成時に、パラメータの組み合わせを絞ってソフトウェアを作成すれば良いので、ソフトウェアの作成及び動作テストに要する時間を大幅に短縮することが可能である。

[0088] （その他の実施形態）

なお、ソフトウェアの記述作成に要する時間を短縮するその他の実施形態として、ニューラルネットワーク処理が使用される用途に着目する観点もある。一般にニューラルネットワーク処理は、学習処理と推論処理という２種類の用途に用いられる。学習処理は、入力データとして既知のデータを入力し、出力データが期待される値に近付く様に重みデータの調整を行う。一方、推論処理は、入力データとして未知のデータを入力して出力データを認識結果として使用する。学習処理は通常は計算リソースの豊富なサーバで行われる場合が多く、かつ、処理時間の制約は緩い。従って、処理時間短縮の必要性は比較的少ない。

[0089] 一方、推論処理は、例えば自動車などに搭載される機器で実行されるケースがあり、計算リソースが限定されているケースがある。また、自動運転などの様に認識結果を機器の制御に使用する場合には、タイムリーに機器を制御できる様にするため、処理時間の制約が比較的厳しい。従って、処理時間短縮の必要性が比較的高い。そこで、推論処理に使用するソフトウェアには本願開示に係る実施形態を適用して処理時間を短縮し、学習処理に使用するソフトウェアには本願開示に係る実施形態を適用せずに既存のソフトウェアを使用することで、ニーズを満たしつつソフトウェアの作成に要する時間を短縮することができる。

[0090] 本願開示に係る実施形態では、推論の畳み込み演算のループの中で活性化処理を実行するソフトウェアの作成がユーザ負担となる場合がある一方、自

動車などの車載装置といったエッジ端末での処理速度及び認識精度の両立という効果を奏する。本開示に係る実施形態では、実施形態４にて重複するパラメータの組み合わせを絞り込んで、作成するソフトウェアのアルゴリズムの数を減らすプログラム作成方法を提供することで、ソフトウェアの作成のユーザ負担を軽減することができる。

[0091] 本発明は上述の実施形態に限定されるものではなく、様々な変形例を含む。例えば、上記した実施形態は本発明を分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、矛盾しない限りにおいて、ある実施形態の構成の一部を他の実施形態の構成で置き換え、ある実施形態の構成に他の実施形態の構成を加えることも可能である。また、各実施形態の構成の一部について、構成の追加、削除、置換、統合、又は分散をすることが可能である。また、実施形態で示した構成及び処理は、処理効率又は実装効率に基づいて適宜分散、統合、または入れ替えることが可能である。

符号の説明

[0092] １０：情報処理装置、１００：コンピュータ

請求の範囲

[請求項1] プロセッサとメモリとを有する情報処理装置であって、

前記プロセッサは、

複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、

前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、を含むニューラルネットワーク処理を実行し、

前記活性化関数処理を、前記畳み込み処理のループ処理内で実行する

ことを特徴とする情報処理装置。

[請求項2] 請求項１に記載の情報処理装置であって、

前記プロセッサは、

前記ループ処理の中で、前記積和演算を行って前記複数の中間データのうちの１つの中間データを生成する都度、前記活性化関数処理を行って該１つの中間データを入力とする前記所定の関数の値を計算して前記複数の出力データのうちの１つの出力データを生成する

ことを特徴とする情報処理装置。

[請求項3] 請求項１に記載の情報処理装置であって、

前記畳み込み処理は、前記積和演算の結果にバイアス値を加算するバイアス値加算処理を含み、

前記バイアス値加算処理を、前記活性化関数処理と同一のループ処理内で実行する

ことを特徴とする情報処理装置。

[請求項4] プロセッサとメモリとを有する情報処理装置であって、

前記プロセッサは、

複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、

前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、を含むニューラルネットワーク処理を実行し、

前記畳み込み処理は、前記積和演算の結果にバイアス値を加算するバイアス値加算処理を含み、

前記プロセッサは、

前記バイアス値加算処理を、前記活性化関数処理と同一のループ処理内で実行し、

前記バイアス値加算処理及び前記活性化関数処理を、前記畳み込み処理とは異なるループ処理内で実行する

ことを特徴とする情報処理装置。

[請求項5] 請求項１～４の何れか１項に記載の情報処理装置であって、

前記複数の入力データは、複数の範囲に対応する複数の画像データであり、

前記出力データは、前記複数の画像データに対する画像認識の処理結果である

ことを特徴とする情報処理装置。

[請求項6] プロセッサとメモリとを有する情報処理装置が実行する情報処理方法であって、

前記プロセッサが、複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理ステップと、

前記プロセッサが、前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理ステップと、

を含むニューラルネットワーク処理ステップを有し、

前記プロセッサが、前記活性化関数処理ステップを、前記畳み込み処理ステップのループ処理内で実行する

ことを特徴とする情報処理方法。

[請求項7] プロセッサとメモリとを有する情報処理装置が実行する情報処理方法であって、

前記プロセッサが、複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理ステップと、

前記プロセッサが、前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理ステップと、

を含むニューラルネットワーク処理ステップを有し、

前記畳み込み処理ステップは、前記プロセッサが、前記積和演算の結果にバイアス値を加算するバイアス値加算処理ステップを含み、

前記プロセッサが、

前記バイアス値加算処理ステップを、前記活性化関数処理ステップと同一のループ処理内で実行し、

前記バイアス値加算処理ステップ及び前記活性化関数処理ステップを、前記畳み込み処理ステップとは異なるループ処理内で実行する

ことを特徴とする情報処理方法。

[請求項8] プロセッサとメモリとを有するコンピュータに、

複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理手順と、

前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理手順と、

を含むニューラルネットワーク処理手順を実行させ、

前記活性化関数処理手順を、前記畳み込み処理手順のループ処理内で実行させる

ことを特徴とする情報処理プログラム。

[請求項9] プロセッサとメモリとを有するコンピュータに、

複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理手順と、

前記複数の中間データを入力とする所定の関数の値を計算して複数の出力データを生成する活性化関数処理手順と、

を含むニューラルネットワーク処理手順を実行させ、

前記畳み込み処理手順は、前記積和演算の結果にバイアス値を加算するバイアス値加算処理手順を含み、

前記プロセッサに、

前記バイアス値加算処理手順を、前記活性化関数処理手順と同一のループ処理内で実行させ、

前記バイアス値加算処理手順及び前記活性化関数処理手順を、前記畳み込み処理手順とは異なるループ処理内で実行させる

ことを特徴とする情報処理プログラム。

[請求項10] 複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、前記複数の中間データを入力として所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、をそれぞれ含む複数のレイヤ処理を組み合わせて前記入力データの認識処理を行うニューラルネットワーク処理を情報処理装置に実行させるソフトウェアを作成するソフトウェア作成装置であって、

各前記レイヤ処理の前記畳み込み処理と前記活性化関数処理のパラメータの組み合わせのリストを作成する第１処理部と、

前記リストの中から重複を除外したパラメータの組み合わせを選択する第２処理部と、

前記第２処理部によって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理のループ処理内で実行するソフトウェアを作成する第３処理部と

を有することを特徴とするソフトウェア作成装置。

[請求項11] 請求項１０に記載のソフトウェア作成装置であって、

前記第３処理部は、

前記複数の入力データとして未知のデータを入力して前記複数の出力データを認識結果として使用する推論処理のソフトウェアの作成時には、前記第２処理部によって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理のループ処理内で実行するソフトウェアを作成し、

前記複数の入力データとして既知のデータを入力し、前記複数の出力データが期待される値に近付く様に前記複数の重みデータの調整を行う学習処理のソフトウェアの作成時には、前記第２処理部によって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理とは異なるループ処理内で実行するソフトウェアを作成する

ことを特徴とするソフトウェア作成装置。

[請求項12] プロセッサとメモリとを有するコンピュータが、複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、前記複数の中間データを入力として所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、をそれぞれ含む複数のレイヤ処理を組み合わせて前記入力データの認識処理を行うニューラルネットワーク処理を情報処理装置に実行させるソフトウェアを作成するソフトウェア作成方法であって、

前記コンピュータが、各前記レイヤ処理の前記畳み込み処理と前記活性化関数処理のパラメータの組み合わせのリストを作成する第１ステップと、

前記コンピュータが、前記リストの中から重複を除外したパラメータの組み合わせを選択する第２ステップと、

前記コンピュータが、前記第２ステップによって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理のループ処理内で実行するソフトウェアを作成する第３ステップと

を有することを特徴とするソフトウェア作成方法。

[請求項13] プロセッサとメモリとを有するコンピュータに、

複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、前記複数の中間データを入力として所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、をそれぞれ含む複数のレイヤ処理を組み合わせて前記入力データの認識処理を行うニューラルネットワーク処理を情報処理装置に実行させるソフトウェアを作成させるためのソフトウェア作成プログラムであって、

前記コンピュータに、

各前記レイヤ処理の前記畳み込み処理と前記活性化関数処理のパラメータの組み合わせのリストを作成する第１手順と、

前記リストの中から重複を除外したパラメータの組み合わせを選択する第２手順と、

前記第２手順によって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理のループ処理内で実行するソフトウェアを作成する第３手順と

を実行させることを特徴とするソフトウェア作成プログラム。

[請求項14] 複数の入力データと複数の重みデータの間の積和演算を行って複数の中間データを生成する畳み込み処理と、前記複数の中間データを入力として所定の関数の値を計算して複数の出力データを生成する活性化関数処理と、をそれぞれ含む複数のレイヤ処理を組み合わせて前記入力データの認識処理を行うニューラルネットワーク処理を情報処理

装置に実行させるソフトウェアを作成するソフトウェア作成方法であって、

各前記レイヤ処理の前記畳み込み処理と前記活性化関数処理のパラメータの組み合わせのリストを作成する第１ステップと、

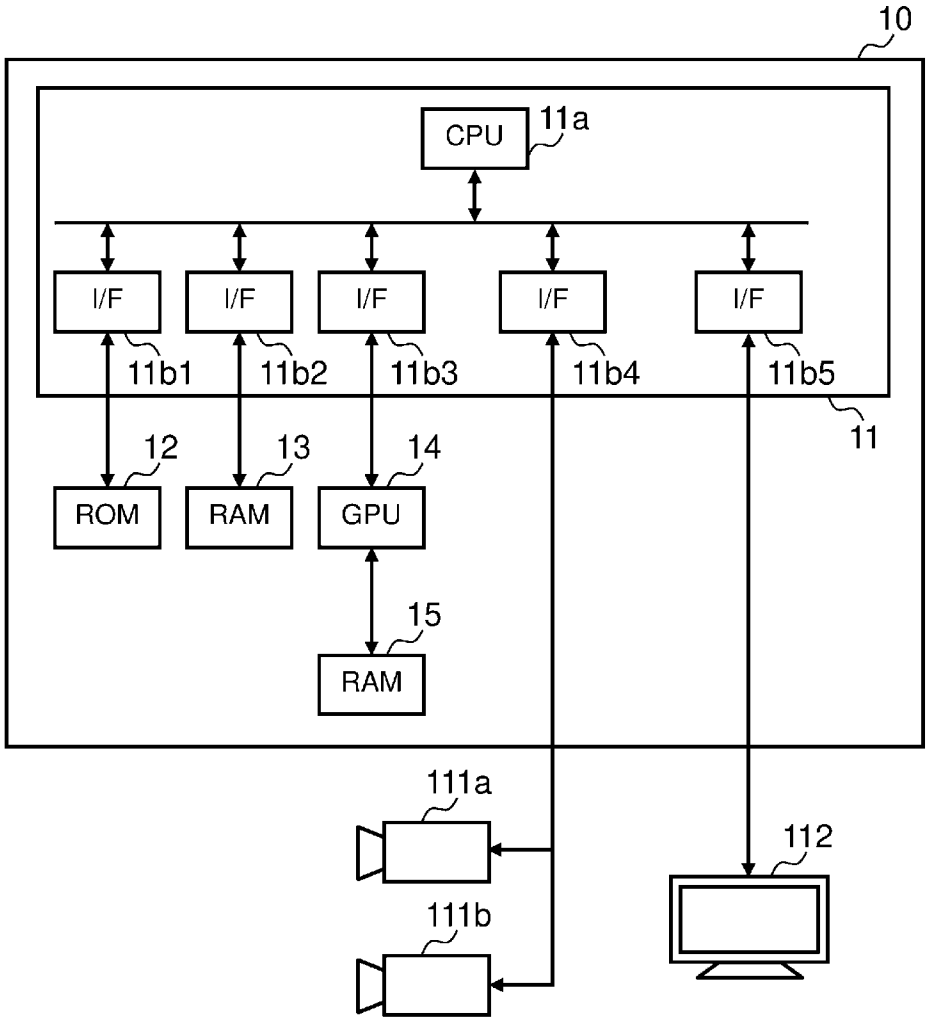
前記リストの中から重複を除外したパラメータの組み合わせを選択する第２ステップと、

前記第２ステップによって選択された前記パラメータの組み合わせに対応する前記畳み込み処理と前記活性化関数処理に関して、該活性化関数処理を、該畳み込み処理のループ処理内で実行するソフトウェアを作成する第３ステップと

を有することを特徴とするソフトウェア作成方法。

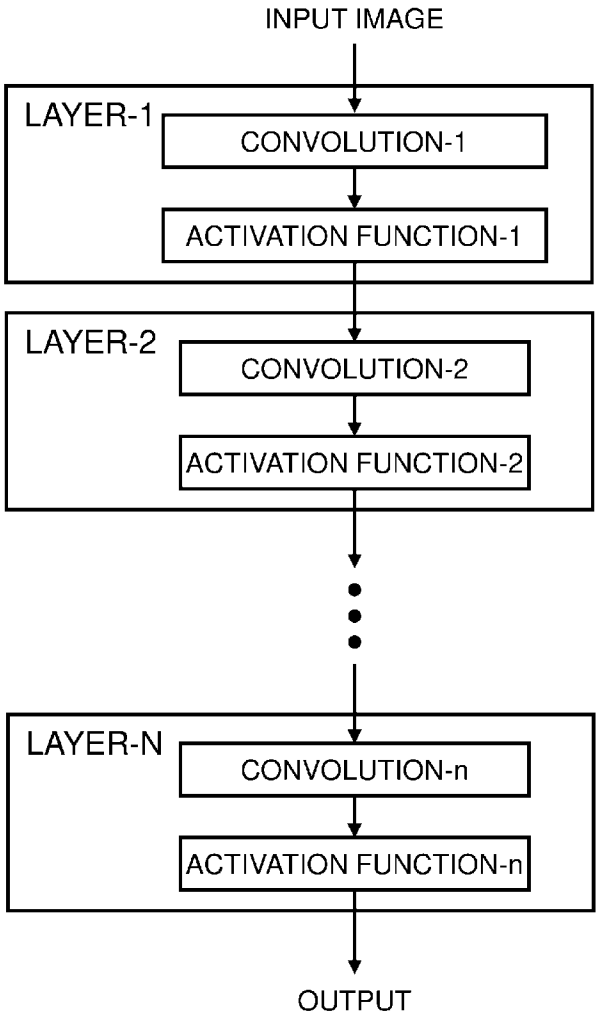
[図1]

図 1


10
CPU
11a
I/F
I/F
I/F
I/F
I/F
11b1
11b2
11b3
11b4
11b5
11
12
13
14
ROM
RAM
GPU
15
RAM
111a
111b
112

[図2]

図 2


INPUT IMAGE
LAYER-1
CONVOLUTION-1
ACTIVATION FUNCTION-1
LAYER-2
CONVOLUTION-2
ACTIVATION FUNCTION-2
LAYER-N
CONVOLUTION-n
ACTIVATION FUNCTION-n
OUTPUT

[図3]

図 3

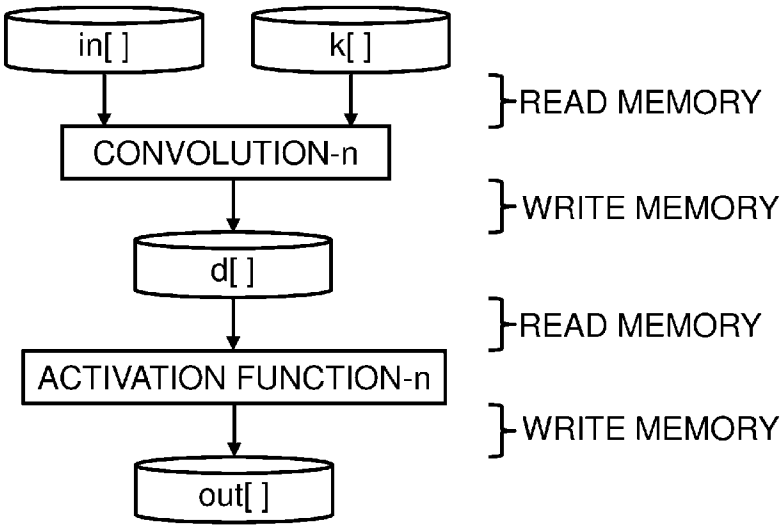
```
1  void layer_n(in, H, W, CI, k, CO, out)
2  {
3    convolution_n(in, H, W, CI, k, CO, d);
4    activation_n(d, H, W, CO, out);
5  }
6
7  void convolution_n(in, H, W, CI, k, CO, d)
8  {
9    for(h=0;h<H;h++){
10     for(w=0;w<W;w++){
11       for(co=0;co<CO;co++){
12         d[CO * (W * h + w) + co] = 0;
13         for(ci=0;ci<CI;ci++){
14           d[CO * (W * h + w) + co] +=
15             k[CI * co + ci] * in[CI * (W * h + w) + ci];
16         }
17       }
18     }
19   }
20 }
21
22 void activation_n(d, H, W, CO, out)
23 {
24   for(h=0;h<H;h++){
25     for(w=0;w<W;w++){
26       for(co=0;co<CO;co++){
27         out[CO * (W * h + w) + co] =
28               f(d[CO * (W * h + w) + co]);
29       }
30     }
31   }
32 }
```

CONVOLUTION-n

ACTIVATION FUNCTION-n

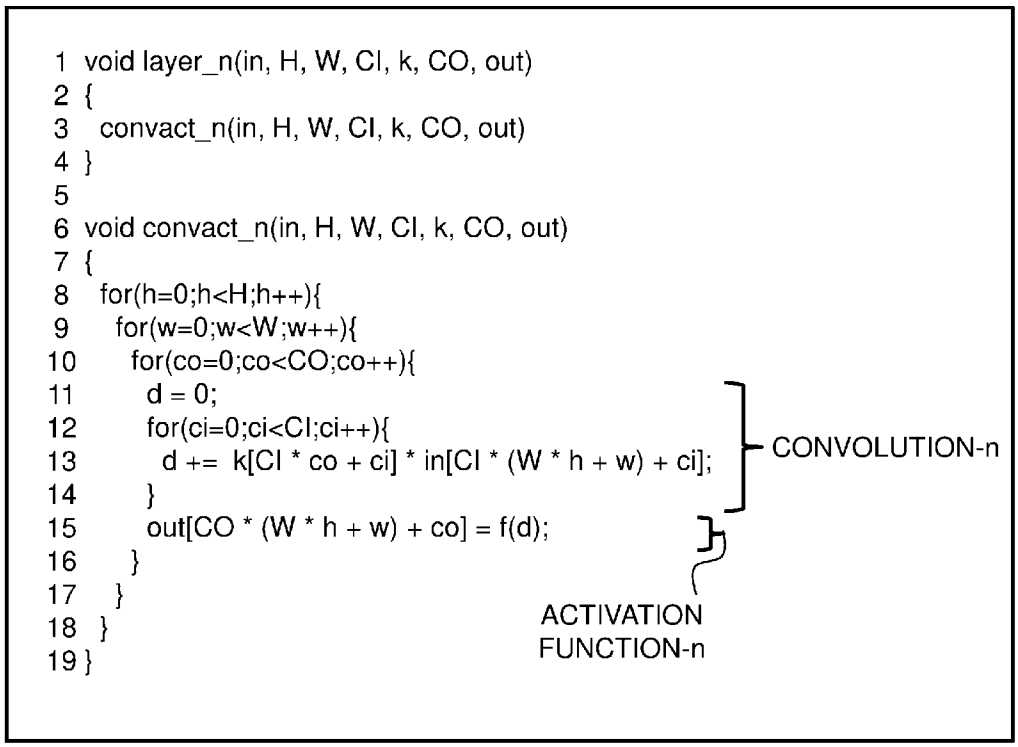
[図4]

図 4


in[ ]
k[ ]
READ MEMORY
CONVOLUTION-n
WRITE MEMORY
d[ ]
READ MEMORY
ACTIVATION FUNCTION-n
WRITE MEMORY
out[ ]

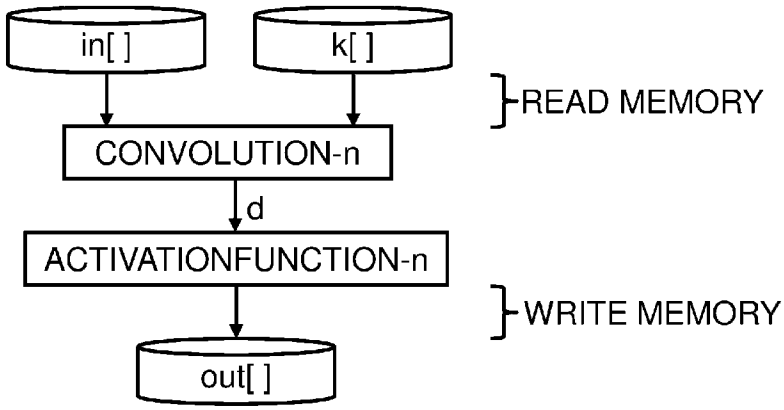
[図5]

図 5


1 void layer_n(in, H, W, CI, k, CO, out)
2 {
3 convact_n(in, H, W, CI, k, CO, out)
4 }
5
6 void convact_n(in, H, W, CI, k, CO, out)
7 {
8 for(h=0;h<H;h++){
9 for(w=0;w<W;w++){
10 for(co=0;co<CO;co++){
11 d = 0;
12 for(ci=0;ci<CI;ci++){
13 d += k[CI * co + ci] * in[CI * (W * h + w) + ci];
14 }
15 out[CO * (W * h + w) + co] = f(d);
16 }
17 }
18 }
19 }
CONVOLUTION-n
ACTIVATION
FUNCTION-n

[図6]

図 6


in[ ]
k[ ]
READ MEMORY
CONVOLUTION-n
d
ACTIVATIONFUNCTION-n
WRITE MEMORY
out[ ]

[図7]

図 7

```
1  void layer_n (in, H, W, CI, k, CO, b, out)
2  {
3    convolution_n(in, H, W, CI, k, CO, b, d2);
4    activation_n(d2, H, W, CO, out);
5  }
6
7  void convolution_n(in, H, W, CI, k, CO, b, d2)
8  {
9    multadd_n(in, H, W, CI, k, CO, d1);
10   biasadd_n(d1, H, W, CO, b, d2);
11 }
12
13 void multadd_n(in, H, W, CI, k, CO, d1)
14 {
15   for(h=0;h<H;h++){
16     for(w=0;w<W;w++){
17       for(co=0;co<CO;co++){
18         d[CO * (W * h + w) + co] = 0;
19         for(ci=0;ci<CI;ci++){
20           d1[CO * (W * h + w) + co] +=
21             k[CI * co + ci] * in[CI * (W * h + w) + ci];
22         }
23       }
24     }
25   }
26 }
27
28 void biasadd_n(d1, H, W, CO, b, d2)
29 {
30   for(h=0;h<H;h++){
31     for(w=0;w<W;w++){
32       for(co=0;co<CO;co++){
33         d2[CO * (W * h + w) + co] =
34           d1[CO * (W * h + w) + co] + b[CO * (W * h + w) + co];
35       }
36     }
37   }
38 }
39
40 void activation_n(d2, H, W, CO, out)
41 {
42   for(h=0;h<H;h++){
43     for(w=0;w<W;w++){
44       for(co=0;co<CO;co++){
45         out[CO * (W * h + w) + co] =
46               f(d2[CO * (W * h + w) + co]);
47       }
48     }
49   }
50 }
```

MULTIADD-n

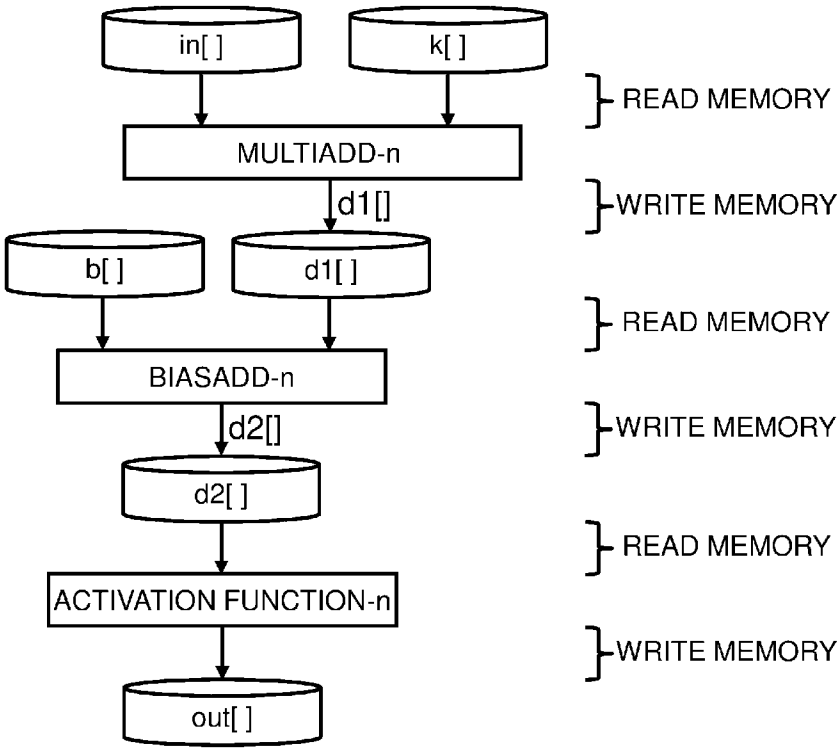
CONVO-LUTION-n

BIASADD-n

ACTIVATION FUNCTION-n

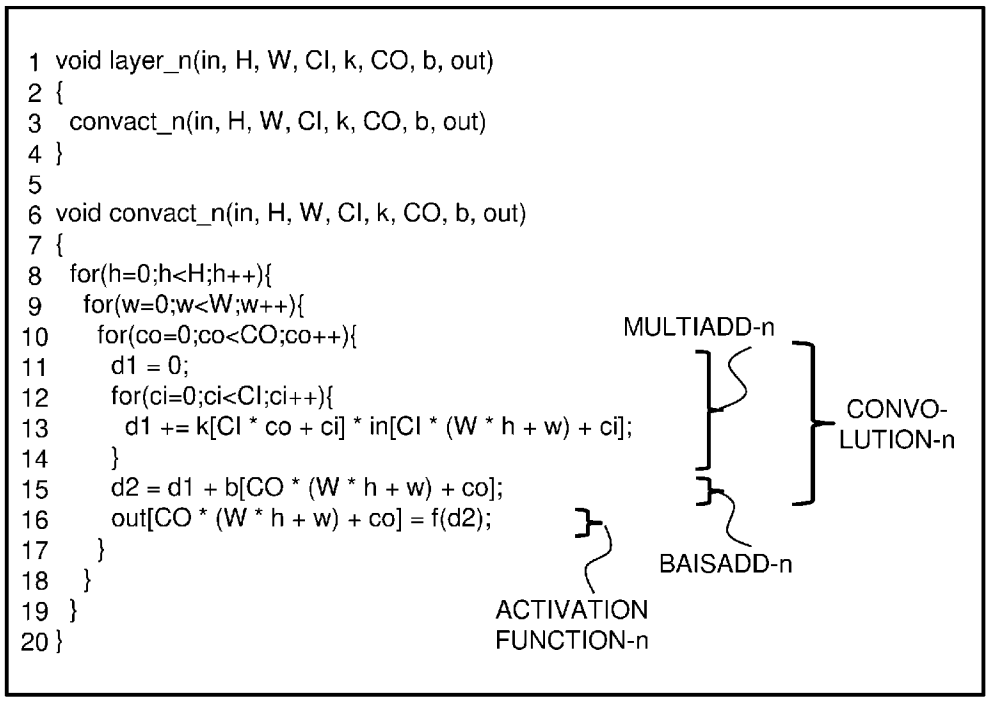
[図8]

図 8


in[ ]
k[ ]
READ MEMORY
MULTIADD-n
d1[]
WRITE MEMORY
b[ ]
d1[ ]
READ MEMORY
BIASADD-n
d2[]
WRITE MEMORY
d2[ ]
READ MEMORY
ACTIVATION FUNCTION-n
WRITE MEMORY
out[ ]

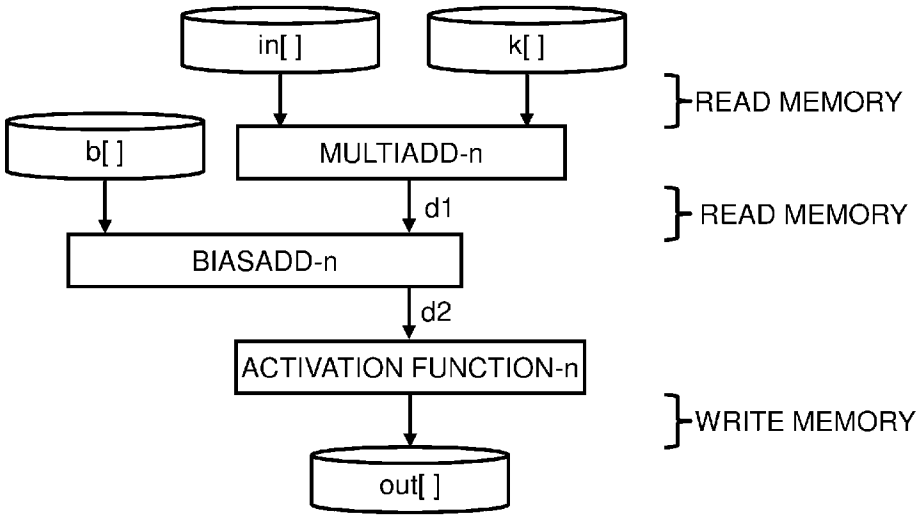
[図9]

図 9


1 void layer_n(in, H, W, CI, k, CO, b, out)
2 {
3 convact_n(in, H, W, CI, k, CO, b, out)
4 }
5
6 void convact_n(in, H, W, CI, k, CO, b, out)
7 {
8 for(h=0;h<H;h++){
9 for(w=0;w<W;w++){
10 for(co=0;co<CO;co++){
11 d1 = 0;
12 for(ci=0;ci<CI;ci++){
13 d1 += k[CI * co + ci] * in[CI * (W * h + w) + ci];
14 }
15 d2 = d1 + b[CO * (W * h + w) + co];
16 out[CO * (W * h + w) + co] = f(d2);
17 }
18 }
19 }
20 }
MULTIADD-n
CONVO-
LUTION-n
BAISADD-n
ACTIVATION
FUNCTION-n

[図10]

図 10


in[ ]
k[ ]
READ MEMORY
b[ ]
MULTIADD-n
d1
READ MEMORY
BIASADD-n
d2
ACTIVATION FUNCTION-n
WRITE MEMORY
out[ ]

[図11]

図 11

```
1   void layer_n (in, H, W, CI, k, CO, b, out)
2   {
3     multadd_n(in, H, W, CI, k, CO, d1);
4     biasact_n(d1, H, W, CO, b, out);
5   }
6
7   void multadd_n(in, H, W, CI, k, CO, d1)
8   {
9     for(h=0;h<H;h++){
10      for(w=0;w<W;w++){
11        for(co=0;co<CO;co++){
12          d[CO * (W * h + w) + co] = 0;
13          for(ci=0;ci<CI;ci++){
14            d1[CO * (W * h + w) + co] +=
15              k[CI * co + ci] * in[CI * (W * h + w) + ci];
16          }
17        }
18      }
19    }
20  }
21
22  void biasact_n(d1, H, W, CO, b, out)
23  {
24    for(h=0;h<H;h++){
25      for(w=0;w<W;w++){
26        for(co=0;co<CO;co++){
27          d2 = d1[CO * (W * h + w) + co]
28               + b[CO * (W * h + w) + co];
29          out[CO * (W * h + w) + co] = f(d2);
30        }
31      }
32    }
33  }
```

MULTI-ADD-n

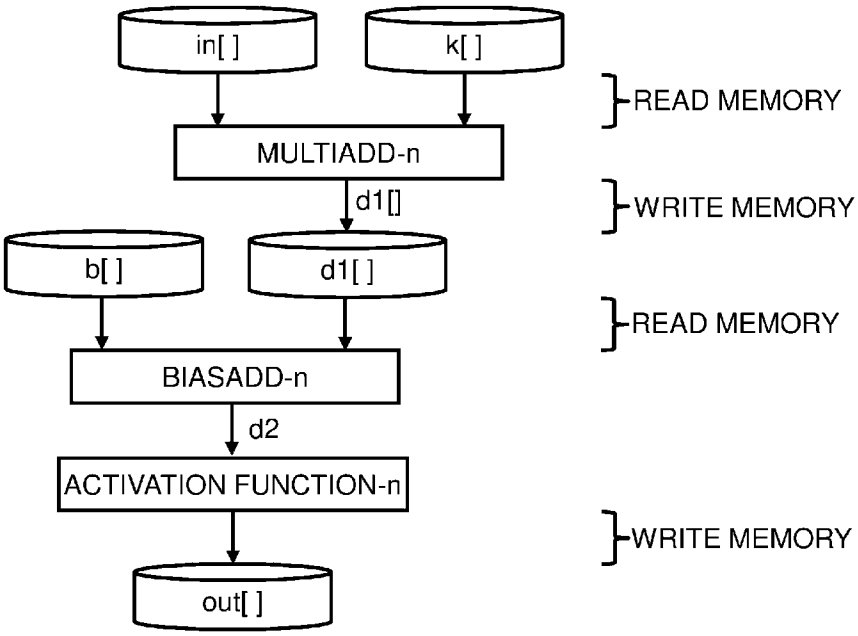
BAIS-ADD-n

ACTIVATION FUNCTION-n

CONVO-LUTION-n

[図12]

図 12


in[ ]
k[ ]
READ MEMORY
MULTIADD-n
d1[]
WRITE MEMORY
b[ ]
d1[ ]
READ MEMORY
BIASADD-n
d2
ACTIVATION FUNCTION-n
WRITE MEMORY
out[ ]

[図13]

図 13

No	INPUT FORM	WEIGHT FORM	OUTPUT FORM	WEIGHT SIZE	CI	CO
1	CHW	CHW	CHW	1x1	3	32
2	CHW	CHW	CHW	1x1	32	32
3	CHW	CHW	CHW	3x3	3	32
4	CHW	CHW	CHW	3x3	32	32
5	CHW	CHW	HWC	1x1	3	32
6	CHW	CHW	HWC	1x1	32	32
7	CHW	CHW	HWC	3x3	3	32
8	CHW	CHW	HWC	3x3	32	32
9	CHW	HWC	CHW	1x1	3	32
10	CHW	HWC	CHW	1x1	32	32
11	CHW	HWC	CHW	3x3	3	32
12	CHW	HWC	CHW	3x3	32	32
13	CHW	HWC	HWC	1x1	3	32
14	CHW	HWC	HWC	1x1	32	32
15	CHW	HWC	HWC	3x3	3	32
16	CHW	HWC	HWC	3x3	32	32
17	HWC	CHW	CHW	1x1	3	32
18	HWC	CHW	CHW	1x1	32	32
19	HWC	CHW	CHW	3x3	3	32
20	HWC	CHW	CHW	3x3	32	32
21	HWC	CHW	HWC	1x1	3	32
22	HWC	CHW	HWC	1x1	32	32
23	HWC	CHW	HWC	3x3	3	32
24	HWC	CHW	HWC	3x3	32	32
25	HWC	HWC	CHW	1x1	3	32
26	HWC	HWC	CHW	1x1	32	32
27	HWC	HWC	CHW	3x3	3	32
28	HWC	HWC	CHW	3x3	32	32
29	HWC	HWC	HWC	1x1	3	32
30	HWC	HWC	HWC	1x1	32	32
31	HWC	HWC	HWC	3x3	3	32
32	HWC	HWC	HWC	3x3	32	32

[図14]

図 14

No	ACTIVATIVATION FUNCTION
1	tanh
2	sigmoid
3	ReLU
4	LReLU
5	PReLU
6	ReLU6
7	SELU
8	Swish
9	hard-Swish
10	Mish

[図15]

図 15

LAYER	INPUT FORM	WEIGHT FORM	OUTPUT FORM	WEIGHT SIZE	CI	CO	ACTIVATION FUNCTION
1	HWC	HWC	HWC	1x1	3	32	Mish
2	HWC	HWC	HWC	1x1	32	32	Mish
3	HWC	HWC	HWC	3x3	32	32	Mish
4	HWC	HWC	HWC	1x1	32	32	Mish
5	HWC	HWC	HWC	3x3	32	32	Mish
6	HWC	HWC	HWC	1x1	32	32	Mish
7	HWC	HWC	HWC	3x3	32	32	Mish
8	HWC	HWC	HWC	1x1	32	32	Mish
9	HWC	HWC	HWC	3x3	32	32	Mish
10	HWC	HWC	HWC	1x1	32	32	Mish
11	HWC	HWC	HWC	3x3	32	32	Mish
12	HWC	HWC	HWC	1x1	32	32	Mish
13	HWC	HWC	HWC	3x3	32	32	Mish
14	HWC	HWC	HWC	1x1	32	32	Mish
15	HWC	HWC	HWC	3x3	32	32	Mish
16	HWC	HWC	HWC	1x1	32	32	Mish
17	HWC	HWC	HWC	3x3	32	32	Mish
18	HWC	HWC	HWC	1x1	32	32	Mish
19	HWC	HWC	HWC	3x3	32	32	Mish
20	HWC	HWC	HWC	1x1	32	32	Mish
21	HWC	HWC	HWC	3x3	32	32	Mish
22	HWC	HWC	HWC	1x1	32	32	Mish
23	HWC	HWC	HWC	3x3	32	32	Mish
24	HWC	HWC	HWC	1x1	32	32	Mish
25	HWC	HWC	HWC	3x3	32	32	Mish
26	HWC	HWC	HWC	1x1	32	32	Mish
27	HWC	HWC	HWC	3x3	32	32	Mish
28	HWC	HWC	HWC	1x1	32	32	Mish
29	HWC	HWC	HWC	3x3	32	32	Mish
30	HWC	HWC	HWC	1x1	32	32	Mish

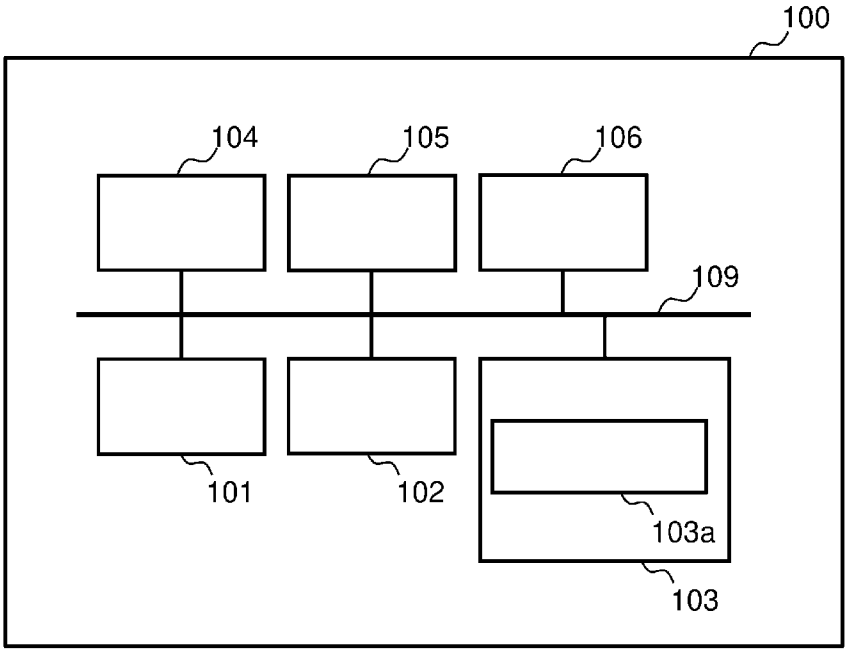
[図16]

図 16

LAYER	INPUT FORM	WEIGHT FORM	OUTPUT FORM	WEIGHT SIZE	CI	CO	ACTIVATION FUNCTION
1	HWC	HWC	HWC	1x1	3	32	Mish
2	HWC	HWC	HWC	1x1	32	32	Mish
3	HWC	HWC	HWC	3x3	32	32	Mish

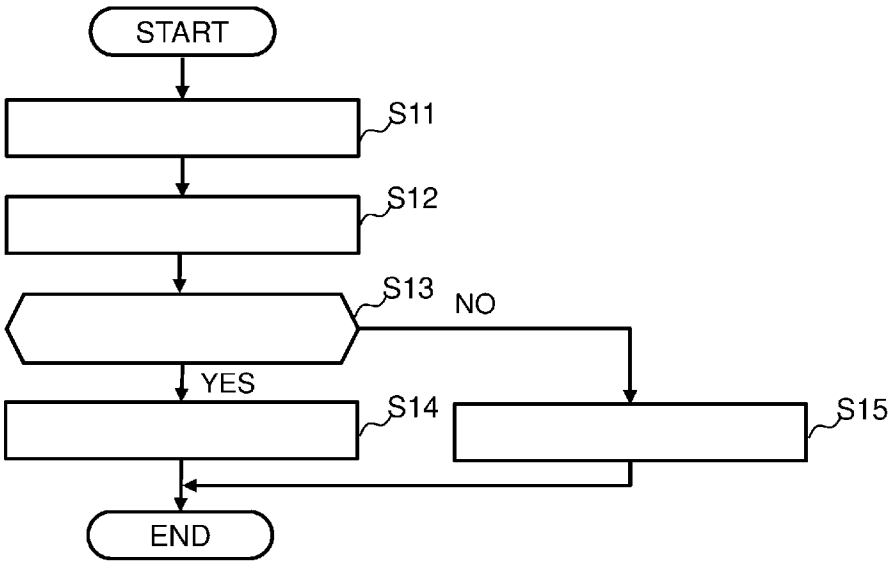
[図17]

図 17



[図18]

図 18



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2023/014229

A. CLASSIFICATION OF SUBJECT MATTER

G06N 3/0464(2023.01)i; ***G06F 17/10***(2006.01)i
FI: G06N3/0464; G06F17/10 A

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)

G06N3/02-3/10; G06F17/10

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 1971-2023
Registered utility model specifications of Japan 1996-2023
Published registered utility model applications of Japan 1994-2023

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.
X	JP 2017-151604 A (DENSO CORP.) 31 August 2017 (2017-08-31) paragraphs [0010], [0018]-[0024], [0032], fig. 6, 7	1, 3, 5-6, 8
A		2, 4, 7, 9-14
A	ONISHI, R et al., Domain Adaptation Method Using Activation Function with Convolution, Proceedings of The 35th Annual Conference of the Japanese Society for Artificial Intelligence [online], 2021, pp. 1-4, [retrieved on 29 May 2023], Retrieved from the Internet: <URL: https://www.jstage.jst.go.jp/article/pjsai/JSAI2021/0/JSAI2021_4G4GS2m05/_article/-char/en>, <DOI: 10.11517/pjsai.JSAI2021.0_4G4GS2m05> entire text, all drawings	1-14
A	JP 2021-517702 A (RECOGNI INC.) 26 July 2021 (2021-07-26) entire text, all drawings	1-14

☐ 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	Date of mailing of the international search report
29 May 2023	**13 June 2023**
Name and mailing address of the ISA/JP **Japan Patent Office (ISA/JP)** **3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915** **Japan**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2023/014229

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP 2017-151604 A	31 August 2017	(Family: none)	
JP 2021-517702 A	26 July 2021	US 2019/0286975 A1 entire text, all drawings WO 2019/177735 A1 KR 10-2020-0140282 A CN 112236783 A	

国際調査報告

国際出願番号
PCT/JP2023/014229

A. 発明の属する分野の分類（国際特許分類（ＩＰＣ））
G06N 3/0464(2023.01)i; G06F 17/10(2006.01)i
FI: G06N3/0464; G06F17/10 A

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））
G06N3/02-3/10; G06F17/10

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　１９７１－２０２３年
日本国実用新案登録公報　１９９６－２０２３年
日本国登録実用新案公報　１９９４－２０２３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
X	JP 2017-151604 A (株式会社デンソー) 31.08.2017 (2017-08-31) 段落[0010],[0018]-[0024],[0032],図6-7	1,3,5-6,8
A		2,4,7,9-14
A	ONISHI, R et al., Domain Adaptation Method Using Activation Function with Convolution, Proceedings of The 35th Annual Conference of the Japanese Society for Artificial Intelligence [online], 2021, pp. 1-4, [retrieved on 2023.05.29], Retrieved from the Internet: <URL: https://www.jstage.jst.go.jp/article/pjsai/JSAI2021/0/JSAI2021_4G4GS2m05/_article/-char/en>, <DOI: 10.11517/pjsai.JSAI2021.0_4G4GS2m05> 全文,全図	1-14
A	JP 2021-517702 A (レコグニ インコーポレイテッド) 26.07.2021 (2021-07-26) 全文,全図	1-14

☐ C欄の続きにも文献が列挙されている。

☑ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー
“A” 特に関連のある文献ではなく、一般的技術水準を示すもの
“E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
“L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
“O” 口頭による開示、使用、展示等に言及する文献
“P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
“T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの
“X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
“Y” 特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
“&” 同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
29.05.2023	13.06.2023

名称及びあて先
日本国特許庁(ISA/JP)
〒100-8915
日本国
東京都千代田区霞が関三丁目４番３号

権限のある職員（特許庁審査官）
渡辺　一帆 5B 4808

電話番号 03-3581-1101 内線 3545

国際調査報告
パテントファミリーに関する情報

国際出願番号
PCT/JP2023/014229

引用文献	公表日	パテントファミリー文献	公表日
JP 2017-151604 A	31.08.2017	（ファミリーなし）	
JP 2021-517702 A	26.07.2021	US 2019/0286975 A1 全文,全図 WO 2019/177735 A1 KR 10-2020-0140282 A CN 112236783 A	