(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2021-22050
(P2021-22050A)
(43)公開日　令和3年2月18日(2021.2.18)

(51)Int.Cl.		F I		テーマコード（参考）
G06N 3/08	(2006.01)	G06N	3/08	120

審査請求 未請求　請求項の数 10　ＯＬ　（全 14 頁）

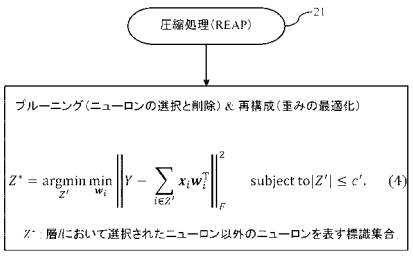
(21)出願番号　特願2019-137019(P2019-137019)
(22)出願日　令和1年7月25日(2019.7.25)

(71)出願人　504145283
国立大学法人　和歌山大学
和歌山県和歌山市栄谷９３０番地
(74)代理人　100111567
弁理士　坂本　寛
(72)発明者　和田　俊和
和歌山県和歌山市栄谷９３０番地　国立大学法人和歌山大学内
(72)発明者　菅間　幸司
和歌山県和歌山市栄谷９３０番地　国立大学法人和歌山大学内

(54)【発明の名称】ニューラルネットワークの圧縮方法、ニューラルネットワーク圧縮装置、コンピュータプログラム、及び圧縮されたニューラルネットワークデータの製造方法

(57)【要約】
【課題】再構成誤差が最小になるようにプルーニングが行われるようにする。
【解決手段】開示のニューラルネットワークの圧縮方法は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、が行われる工程を備える。前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される。
【選択図】図４

図4



【特許請求の範囲】
【請求項１】
選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、が行われる工程を備え、
前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される
ニューラルネットワークの圧縮方法。
【請求項２】
前記プルーニングは、全結合層におけるニューロンのプルーニングである
請求項１に記載のニューラルネットワークの圧縮方法。
【請求項３】
前記プルーニングは、畳み込み層におけるチャネルのプルーニングである
請求項１に記載のニューラルネットワークの圧縮方法。
【請求項４】
前記再構成誤差は、プルーニング対象の挙動ベクトルを、前記プルーニング対象以外の他のプルーニング単位の挙動ベクトルが張る部分空間に射影したときの射影残差に基づいて計算される
請求項１から３のいずれか１項に記載のニューラルネットワークの圧縮方法。
【請求項５】
前記射影残差は、プルーニング対象の挙動ベクトルの双直交基底に基づいて計算される
請求項４に記載のニューラルネットワークの圧縮方法。
【請求項６】
前記射影残差は、グラン・シュミットの直交化計算の反復適用により計算される
請求項４に記載のニューラルネットワークの圧縮方法。
【請求項７】
前記再構成誤差は、並列計算される
請求項１から６のいずれか１項に記載のニューラルネットワークの圧縮方法。
【請求項８】
選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、を行うニューラルネットワーク圧縮装置であって、
前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択されるよう構成されている
ニューラルネットワーク圧縮装置。
【請求項９】
コンピュータを、ニューラルネットワーク圧縮装置として機能させるためのコンピュータプログラムであって、
前記ニューラルネットワーク圧縮装置は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、を行うよう構成され、
前記ニューラルネットワーク圧縮装置において、前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される
コンピュータプログラム。
【請求項１０】
選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、が行われる圧縮工程と、
前記圧縮工程により圧縮されたニューラルネットワークのデータを出力する工程と、
を有し、
前記圧縮工程において、前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される
圧縮されたニューラルネットワークデータの製造方法。

【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、ニューラルネットワークの圧縮に関する。
【背景技術】
【０００２】
ディープニューラルネットワーク（ＤＮＮ）のようなニューラルネットワークの圧縮手法として、プルーニング（Pruning；枝刈り）と、再構成（Reconstruction）と、を行う手法がある。
【０００３】
プルーニングは、全結合型ニューラルネットワーク（ＦＣＮ）においては、ニューロン（とそのニューロンに接続された重み）の削除として行われ、畳み込み型ニューラルネットワーク（ＣＮＮ）においては、チャネルの削除として行われる（非特許文献１参照）。チャネルの削除は、削除されるチャネルに属する重み全体の削除として行われる。
【０００４】
再構成はプルーニング後に行われる。再構成では、プルーニング前の出力に近づくように、ニューラルネットワークの重みパラメータの調整が行われる。例えば、ＦＣＮにおいては、再構成として、ニューロン間の結合の重みパラメータの調整が行われ、ＣＮＮにおいては、再構成として、フィルタ（カーネル）における重みパラメータの調整が行われる。
【先行技術文献】
【非特許文献】
【０００５】
【非特許文献１】Yihui He, Xiangyu Zhang, Jian Sun, “Channel Pruning for Accelerating Very Deep Neural Networks,” Proc. of ICCV2017, 2017
【発明の概要】
【０００６】
プルーニングと再構成とを行う従来の圧縮手法においては、再構成後における誤差（再構成誤差）が最小になるようにプルーニングが行われるわけではない、という課題が存在することを、本発明者らは見出した。以下では、この課題について説明する。なお、以下の説明では、簡単化のため、ＦＣＮを前提として説明する。
【０００７】
プルーニングをする際には、削除されるニューロン（プルーニング対象）を選択する必要がある。削除されるニューロンは、削除されるニューロンが存在する層の次層に与える誤差に着目して、選択される。具体的には、ニューロンの削除によって次層に与える誤差が最小となるニューロンが、削除されるニューロンとして選択される。例えば、ニューロンA_1を削除した場合に、次層に与える誤差がE_1であり、ニューロンA_2を削除した場合に、次層に与える誤差がE_2である場合、誤差E_1が誤差E_2よりも小さければ、ニューロンA_1が、ニューロンA_2よりも優先して、削除されるニューロンとして選択されることになる。
【０００８】
プルーニング後の再構成では、削除されずに残ったニューロンから次層のニューロンへ向かう結合における重みパラメータが、調整される。重みの調整は、ニューロンの削除により次層に与える誤差が最小になるように実行される。例えば、プルーニングにおいてニューロンA_1を削除することで次層に与える誤差がE_1である場合、再構成では、誤差E_1ができるだけ小さくなるように、重みの調整が行われる。重みの調整により最小化された誤差E_1は、再構成誤差E_{1r}と呼ばれる。
【０００９】
以上のような従来の圧縮手法では、再構成誤差E_{1r}が最小になるようにプルーニングが行われるわけではない。例えば、前述のように、ニューロンA_1を削除した場合に次層

に与える誤差がE_1であり、再構成誤差がE_{1r}であるとする。また、ニューロンA_2を削除した場合に次層に与えられる誤差がE_2であり、再構成誤差がE_{2r}であるとする。この場合において、誤差E_1が誤差E_2よりも小さいとしても、再構成誤差E_{1r}が再構成誤差E_{2r}よりも大きいことがある。すなわち、削除により生じる誤差が最小であっても、再構成誤差が最小になるとの保証はない。

【００１０】

したがって、上記の課題の解決が望まれる。本開示において、上記の課題は、再構成誤差が最小になるようにプルーニングすることにより解決される。更なる詳細は、後述の実施形態として説明される。

【図面の簡単な説明】

【００１１】

【図１】図１は、ニューラルネットワーク圧縮装置及びニューラルネットワーク利用装置の構成図である。

【図２】図２は、ニューラルネットワークの構成及び伝播量Ｙの定式化の説明図である。

【図３】図３は、比較例に係る圧縮処理のフローチャートである。

【図４】図４は、実施形態に係る圧縮処理のフローチャートである。

【図５】図５は、実施形態に係る圧縮処理のためのGreedy Algorithmである。

【図６】図６は、部分空間Uへのx_jの射影を示す図である。

【図７】図７は、残差r_jの計算方法を示す。

【図８】図８は、残差r_jの計算方法を示す。

【図９】図９は、別のニューロンを削除する際の再構成誤差の計算方法を示す。

【図１０】図１０は、別のニューロンを削除する際の再構成誤差の計算方法を示す。

【図１１】図１１は、ＲＥＡＰの畳み込み層への適用を示す。

【図１２】図１２は、グラン・シュミットの直交化計算の適用による射影残差r_jの計算方法を示す。

【図１３】図１３は、実験結果を示す図である。

【発明を実施するための形態】

【００１２】

＜１．ニューラルネットワークの圧縮方法、ニューラルネットワーク圧縮装置、コンピュータプログラム、及び圧縮されたニューラルネットワークデータの製造方法の概要＞

【００１３】

（１）実施形態に係るニューラルネットワークの圧縮方法は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、が行われる工程を備える。前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される。これにより、再構成誤差が最小になるようにプルーニングされる。

【００１４】

（２）前記プルーニングは全結合層におけるニューロンのプルーニングであってもよい。

【００１５】

（３）前記プルーニングは畳み込み層におけるチャネルのプルーニングであってもよい。

【００１６】

（４）前記再構成誤差は、プルーニング対象の挙動ベクトルを、前記プルーニング対象以外の他のプルーニング単位の挙動ベクトルが張る部分空間に射影したときの射影残差に基づいて計算されるのが好ましい。

【００１７】

（５）前記射影残差は、プルーニング対象の挙動ベクトルの双直交基底に基づいて計算されるのが好ましい。

【００１８】

（６）前記射影残差は、グラン・シュミットの直交化計算の反復適用により計算されてもよい。

【００１９】
（７）前記再構成誤差は、並列計算等で高速化されるのが好ましい。
【００２０】
（８）実施形態に係るニューラルネットワーク圧縮装置は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、を行うよう構成されている。ニューラルネットワーク圧縮装置は、前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択されるよう構成されている。
【００２１】
（９）実施形態に係るコンピュータプログラムは、コンピュータを、ニューラルネットワーク圧縮装置として機能させるためのコンピュータプログラムである。前記ニューラルネットワーク圧縮装置は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、を行うよう構成され、前記ニューラルネットワーク圧縮装置において、前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される。
【００２２】
（１０）実施形態に係る圧縮されたニューラルネットワークデータの製造方法は、選択されたプルーニング対象に対するプルーニングと、プルーニングされたニューラルネットワークの再構成と、が行われる圧縮工程と、前記圧縮工程により圧縮されたニューラルネットワークのデータを出力する工程と、を有し、前記圧縮工程において、前記プルーニング対象は、プルーニング及び再構成によって生じる再構成誤差が最小になるように選択される。
【００２３】
＜２．ニューラルネットワークの圧縮方法、ニューラルネットワーク圧縮装置、コンピュータプログラム、及び圧縮されたニューラルネットワークデータの製造方法の例＞
【００２４】
図１は、実施形態に係るニューラルネットワーク圧縮装置（以下、「圧縮装置」という）１０とニューラルネットワーク利用装置（以下、「利用装置」という）１００とを示している。実施形態に係る圧縮装置１０は、ニューラルネットワークＮ１を圧縮して小規模化するための圧縮処理２１を実行する。圧縮処理２１を実行することにより実施される方法は、圧縮されたニューラルネットワークの製造方法又は圧縮されたニューラルネットワークデータの製造方法でもある。
【００２５】
ニューラルネットワークは、複数の人工ニューロン（「ノード」ともいう）が結合した人工的な計算機構である。ニューラルネットワークは、例えば、ディープニューラルネットワーク（ＤＮＮ）である。ＤＮＮは、例えば、全結合型ニューラルネットワーク（ＦＣＮ）であってもよいし、畳み込み型ニューラルネットワーク（ＣＮＮ）であってもよい。以下では、圧縮処理の対象となるニューラルネットワークＮ１を、「原ニューラルネットワーク」といい、圧縮されたニューラルネットワークＮ２を「圧縮ニューラルネットワーク」という。なお、実施形態に係る圧縮装置１０は、原ニューラルネットワークＮ１の機械学習（深層学習）のための処理も実行可能である。圧縮装置１０は、学習済の原ニューラルネットワークＮ１を圧縮する。
【００２６】
圧縮装置１０は、１又は複数のプロセッサ２０及び記憶装置３０を有するコンピュータによって構成されている。１又は複数のプロセッサ２０は、例えば、グラフィックプロセッシングユニット（ＧＰＵ）を含む。１又は複数のプロセッサ２０は、さらにＣＰＵを含んでもよい。ＧＰＵのような大規模並列計算機構は、大規模なニューラルネットワークに関する処理を実行するための大量の計算に適している。
【００２７】
記憶装置３０は、プロセッサ２０によって実行されるコンピュータプログラム３１を記

憶している。プロセッサ２０は、コンピュータプログラム３１を実行することで、圧縮処理２１を行う。圧縮処理２１は、プルーニング（Pruning；枝刈り）と再構成（Reconstruction）とを含む。

【００２８】

記憶装置３０は、圧縮処理２１によって製造された圧縮ニューラルネットワークＮ２を表すデータ（圧縮ニューラルネットワークデータ）Ｎ２０を記憶することができる。圧縮ニューラルネットワークデータＮ２０は、圧縮ニューラルネットワークＮ２を表現する各種のパラメータ（重み、結合関係など）からなるデータである。圧縮装置１０は、圧縮ニューラルネットワークデータＮ２０を、ニューラルネットワークエンジン等へ、出力することができる。圧縮ニューラルネットワークデータＮ２０は、ニューラルネットワークエンジンに読み込まれることで、そのニューラルネットワークエンジンを圧縮ニューラルネットワークＮ２として機能させる。

【００２９】

利用装置１００は、圧縮ニューラルネットワークデータＮ２０を読み込んで、圧縮ニューラルネットワークＮ２として機能するニューラルネットワークエンジンを有する。ニューラルネットワークエンジンは、例えば、プロセッサ２００と記憶装置３００とを備える。プロセッサ２００は、例えば、組み込み系システムにおける低消費電力のＣＰＵでよい。圧縮ニューラルネットワークデータＮ２０は、原ニューラルネットワークＮ１のデータに比べて、サイズが小さいため、低消費電力のＣＰＵによる処理が可能である。

【００３０】

組み込み系システムは、汎用的なコンピュータシステムではなく、特定の用途に向けられたコンピュータシステムであり、例えば、スマートフォン・家電などの家庭用機器、産業用ロボットなどの産業用機器、各種の医療用機器、自動車・ドローンなどのビークル、及びその他の機器におけるコンピュータシステムである。組み込み系システムでは、プロセッサとして、低消費電力のＣＰＵが使われることが多いが、圧縮ニューラルネットワークデータＮ２０は、データサイズが小さいため、実行が容易である。

【００３１】

圧縮ニューラルネットワークＮ２は、例えば、画像・音声の変換、セグメンテーション、識別などの用途に用いられる。より具体的には、例えば、店舗等の客数計測、男女・年齢層分析、車両計数、車種分析など、対象物の画像から必要な情報を抽出するために用いることができる。原ニューラルネットワークＮ１は大規模であり、計算コストが大きいため、組み込み系システムでの実行が困難であるが、圧縮ニューラルネットワークＮ２は、小規模化されているため、組み込み系システムでの実行が容易である。

【００３２】

以下、圧縮処理２１について説明する。以下では、理解の容易のため、まず、全結合型ニューラルネットワーク（ＦＣＮ）を前提に、圧縮処理２１を説明し、その後、同様の圧縮処理２１を、畳み込み型ニューラルネットワーク（ＣＮＮ）に適用できることを説明する。

【００３３】

図２は、原ニューラルネットワークＮ１である全結合型ニューラルネットワーク（ＦＣＮ）における層ｌと、層ｌの次の層である層ｌ＋１と、を示している。図２では、２つの層（ｌ層，ｌ＋１層）が代表的に示されている。ＦＣＮにおける各層は、層状に並べられた人工ニューロン（以下、単に「ニューロン」という）が層間で結合されている全結合層である。各層中における丸印がニューロンである。層ｌに含まれるニューロン数はｃであり、層ｌ＋１に含まれるニューロン数はＣである。

【００３４】

図２中の式（１）は、ニューラルネットワークにデータが与えられた時における、層ｌから次の層ｌ＋１への伝播量Ｙを定式化している。ここでは、Ｙは、層ｌ＋１のＣ個のニューロンの内部活性度を表すＮ×Ｃ行列とする。換言すると、Ｙは、層ｌから与えられる、層ｌ＋１への入力でもある。

【００３５】
Ｎ個のデータ（例えば、Ｎ個の画像データ）を、層ｌのｃ個のニューロンに与えた場合、層ｌの各ニューロンからはＮ個の出力が生じる。層ｌにおけるi番目のニューロンの出力がx_iで表される。x_iは、Ｎ次元のベクトルである。x_iは、i番目のニューロンにＮ個のデータが与えられた時のi番目のニューロンの出力（挙動）を示す。すなわち、x_iはi番目のニューロンの挙動ベクトル（ニューロン挙動ベクトル）でもある。なお、ニューラルネットワークに与えられるデータは、画像以外の他のデータ、例えば、音声データ等であってもよい。画像データ等のデータは、各ニューロンの挙動を把握するために、ニューラルネットワークに与えられる。

【００３６】
図２中のw_iは、ｌ層のi番目のニューロンから、ｌ＋１層のＣ個のニューロンへ向かう結合の重み（重み係数）からなるＣ次元の重みベクトルである。

【００３７】
この場合、次層ｌ＋１への伝播量Ｙは、層ｌにおけるニューロンの出力x_iと、層ｌから次層ｌ＋１への重みベクトルw_iと、によって、図２中の式（１）に示すように定式化される。

【００３８】
実施形態に係る圧縮処理２１の目的は、上記のＹをできるだけ変化させることなく、ニューロンの数を、所望の数ほど減少させることである。ニューロンを減少させても、Ｙの変化が少なければ、原ニューラルネットワークＮ１の性能を維持することができる。つまり、ニューラルネットワークを圧縮しても、精度低下を防止できる。なお、上記のように、ＦＣＮでは、ニューロンがプルーニング単位であるが、ＣＮＮでは、チャネルがプルーニング単位である。なお、プルーニング単位は、削除の単位である。

【００３９】
実施形態に係る圧縮処理２１の説明に先立ち、比較例に係る圧縮処理１２１を説明する。図３は、比較例に係る圧縮処理１２１を示している。図３に示す圧縮処理１２１は、プルーニング工程１２２と、再構成工程１２３と、を有している。比較例においては、プルーニング工程１２２と再構成工程１２３とは完全に分離している。

【００４０】
プルーニング工程１２２では、ある層ｌに含まれる複数のニューロン（複数のプルーニング単位）から削除されるニューロン（プルーニング対象）が選択され、選択されたニューロンの削除が行われる。削除されるニューロンが層ｌの中から選択される場合、次層ｌ＋１への伝播量Ｙに与える影響が最も小さくなるニューロンが、削除されるニューロン（プルーニング対象）として選択される。この選択の際には、Lasso回帰を用いて、図３の式（２）に従ってニューロンが選択される（非特許文献１参照。非特許文献１ではチャネルが選択される）。比較例においては、ニューロンの選択の際には、重みが調整されることはない。

【００４１】
比較例では、次層ｌ＋１での内部活性度の誤差に関するペナルティ項に重要度ベクトルβのL_1ノルムを加えている。重要度ベクトルβのL_1ノルムを最小化する重要度ベクトルβを求めると、次層ｌ＋１の活性度の誤差を低く抑えつつゼロ要素の多いβが得られ、削除すべきニューロンを決定できる。すなわち、式（２）の最適化の結果、最適化されたニューロンの重要度ベクトルβ^*が得られるが、そのベクトルのi番目の要素β^*_iが０ならば、i番目のニューロンは不要であり、削除されるニューロンとして選択される。

【００４２】
比較例において、削除されるニューロンの数は、ハイパーパラメータλの微調整によってコントロールされる。λを増加させれば、削除されるニューロンの数が増え、λを減少させれば、削除されるニューロンの数が減る。比較例においては、削除されるニューロンの数は、λによってコントロールされるため、削除されるニューロンの数の制御は難しい。

【００４３】
再構成工程１２３では、層ｌにおいて、プルーニング後に残ったニューロンが、次層ｌ＋１に与えるＹが、プルーニング前におけるＹ（本来のＹ）に近づくように、重みが調整（最適化）される。重みの調整は、図３中の式（３）に従って行われる。式（３）は、再構成誤差を最小化する重みベクトルを求める。ここでの再構成誤差は、プルーニング前のＹと、プルーニング後に重みを調整したときのＹと、の差に基づく。
【００４４】
比較例においては、プルーニング対象は、再構成を行う前の誤差を最小化するように選択されており、再構成後に最も誤差が小さくなるように選択されているわけではない。つまり、比較例では、プルーニング対象の選択は、再構成前の誤差に基づいて行われており、再構成は、再構成後の誤差に基づいて行われており、プルーニングと再構成とが、異なる基準で行われている。また、比較例においては、Lassoを用いており、削除されるニューロンの数をコントロールするには、λの人手による微調整が必要となる。つまり、比較例では、削除されるニューロンの数のコントロールは困難である。
【００４５】
図４は、実施形態に係る圧縮処理２１を示している。以下では、実施形態に係る圧縮処理２１を、ＲＥＡＰ（Reconstruction Error Aware Pruning）という。
【００４６】
比較例では、プルーニングと再構成とが異なる基準で行われていたのに対して、ＲＥＡＰでは、プルーニングと再構成とを同じ基準で行う。すなわち、ＲＥＡＰでは、再構成誤差が最小になるようにプルーニングされるとともに再構成される。ＲＥＡＰでは、図４中の式（４）に従って、削除されるニューロンが決定される。なお、式（４）のＺ＊は、層ｌにおいて、削除して残ったニューロンを示す。式（４）においては、重みベクトルw_iは、Ｚ’をＺ＊に固定する前に最適化される。したがって、式（４）によれば、次層ｌ＋１への伝播量Ｙの再構成誤差を最小化するニューロンの集合が求まる。
【００４７】
ＲＥＡＰでは、再構成誤差が最小になるようにプルーニング対象であるニューロンが選択されるため、ＲＥＡＰは、再構成誤差が最小になるとは限らない比較例に比べて、有利である。
【００４８】
式（４）は、組み合わせ最適化問題であり、グリーディ法（Greedy Algorithm）によって解かれる。図５は、式（４）を解くためのアルゴリズムを示している。まず、ステップＳ１において、層ｌにおけるｊ番目のニューロンを消してみる。ステップＳ２において、層ｌにおいて残ったニューロンのみでＹを再構成して誤差（再構成誤差）を計算する。再構成誤差は、図５中において式（５）として示されるコスト関数Ｐ（Ｚ’）を計算することで求まる。再構成誤差が求まると、一旦、削除したｊ番目のニューロンを元に戻す。ステップＳ３で示される繰り返しループにおいては、ステップＳ１及びステップＳ２がすべてのｊ（j∈Z）について繰り返され、最もＰ（Ｚ’）の値が小さくなるニューロンが、プルーニング対象として選択され、最終的に削除される。
【００４９】
ステップＳ３の繰り返しループによって、一つのニューロンが削除される。ステップＳ４で示される繰り返しループにおいては、残ったニューロンのみで、再度、ステップＳ３の繰り返しループが実行される。再度、ステップ３の繰り返しループが実行されると、別のニューロンが削除される。
【００５０】
層ｌからいくつのニューロンを削除するかは、ステップＳ４の繰り返しループを何回実行するかによって決まる。したがって、所望数Ｄのニューロンを削除したい場合、ステップＳ４の繰り返しループをＤ回実行すればよい。したがって、ＲＥＡＰでは、削除されるニューロンの数のコントロールは容易である。
【００５１】

ＲＥＡＰでは、比較例に比べて、計算量が増加する。すなわち、ＲＥＡＰでは、再構成誤差を求める際に最小二乗法を適用するため、連立方程式を解く必要がある。そして、解く必要のある連立方程式は層内のニューロン数分存在する。例えば、層ｌのニューロン数がｃであり、次層ｌ＋１のニューロン数がＣである場合，重みパラメータ数はｃ×Ｃになる。１個のニューロンを削除する場合、係数行列のサイズが（ｃ－１）Ｃ×（ｃ－１）Ｃとなる。したがって、一つの連立方程式を解くための時間計算量は、Ｏ（c^3C^3）である。この連立方程式をｃ回解く必要があることから、Ｏ（c^4C^3）の時間計算量となる。

【００５２】

ここで、本来解こうとしている最小二乗問題は、あるニューロンの挙動を表す挙動ベクトルx_jがなくなったときに、残りのニューロン集合Ｚ’の挙動ベクトルx_i（ｉ∈Ｚ’）の線形和で、次層ｌ＋１への本来の伝播量Ｙを近似する、という問題である。この近似による誤差は、図６に示すように、x_jをx_i（ｉ∈Ｚ’）の線形和で表現した際の誤差r_jに起因している。したがって、この誤差r_jを最小化すれば、誤差r_jを示すベクトルに、次層ｌ＋１への重みベクトルw_i^Tを掛けるだけで、次層ｌ＋１での活性度（伝播量Ｙ）の誤差が計算できる。

【００５３】

すなわち、ｊ番目のニューロンの削除により生じる再構成誤差を計算するためには、図６中の式（６－１）に示すコスト関数を計算する必要がある。後述するように、式（６－１）のコスト関数は、式（６－２）のように表される。したがって、x_jをx_i（ｉ∈Ｚ’）が張る部分空間に射影した際の射影残差r_jを、巨大な係数行列を用いることなく、計算することができれば、再構成誤差を効率よく計算することができる。一例として、残差r_jは、式（６－３）のように計算される。

【００５４】

残差r_jの求め方の一例は、図７及び図８に詳しく説明されている。図７及び図８に示す計算方法は、残差r_jが、x_jの双直交基底と線形従属であることを利用したものである。すなわち、残差r_jは、x_jの双直交基底に対する、x_jの射影である。残差r_jは、x_jの双直交基底に基づいて計算される。図７及び図８に示す計算方法によれば、連立方程式の係数行列を使用せずに、残差r_jを効率的に計算できる。また、図９及び図１０は、あるニューロンを削除した後に、別のニューロンを削除するための再構成誤差を効率的に計算する方法を説明している。

【００５５】

高速化のため、再構成誤差の計算は、並列処理で行うのが好ましい。再構成誤差を並列計算することで、高速に再構成誤差を求めることができる。ただし、最小二乗法を解く際に必要となる係数行列を格納するメモリ量は、並列化によって増大する。

【００５６】

したがって、再構成誤差の計算を効率的に行うには、消費メモリ量に相当する空間計算量を削減することが好ましい。ここで、空間計算量（消費メモリ）は一つの連立方程式当たり、Ｏ（c^2C^2）であり、ｃ個並列計算で同時に計算するとＯ（c^3C^2）になる。空間計算量を削減するには、連立方程式の係数行列を使用せずに最小二乗法の計算を行うのが好ましい。

【００５７】

図１２は、並列計算で残差r_jを求める例を示している。図１２では、グラン・シュミット（Gram-Schmit）の直交化計算を反復適用することで、残差r_jを効率的に計算できる。図１１に示す計算方法では、連立方程式の係数行列を使用せずに、残差r_jを効率的に並列計算できる。ニューロン数が非常に多い（実行環境によるが、目安として、4096個以上）の場合は、グラン・シュミットを用いる解法の方が高速である。

【００５８】

図１１は、ＲＥＡＰが畳み込み層におけるプルーニング及び再構成に適用できることを説明している。図１１中の式（１９）に示すように、畳み込み層におけるスライディング

ウィンドウ操作は、行列乗算の和によって表される。式（１９）は、全結合層のための式（１）と同様の形式であることから、畳み込み層においても、全結合層と同様に、ＲＥＡＰを適用できることがわかる。

【００５９】

図１３は、ニューラルネットワークの圧縮をＲＥＡＰ及び比較例によって行った実験結果を示している。実験では、ImageNetデータセットによってトレーニングしたVGG16を、原ニューラルネットワークＮ１として用いた。原ニューラルネットワークＮ１に対する圧縮処理２１としてＲＥＡＰを適用した場合及び比較例を適用した場合それぞれについて、画像の認識精度（正解率）を求めた。

【００６０】

図１３の横軸は、ＦＬＯＰｓ（浮動小数点演算数）を示し、縦軸は、正解率を示す。ＦＬＯＰｓが小さいほど、圧縮ニューラルネットワークＮ２の演算数が少なく、圧縮率が大きいことを示す。図１３に示すように、比較例では、圧縮率を増加（削除されるニューロン数を増加）させると、正解率が０．７（７０％）程度まで下がるのに対して、ＲＥＡＰでは、圧縮率を増加させても、正解率は０．８（８０％）程度までしか下がらなかった。したがって、ＲＥＡＰの方が、圧縮による精度低下を抑制できていることがわかる。

【００６１】

＜３．付記＞

本発明は、上記実施形態に限定されるものではなく、様々な変形が可能である。

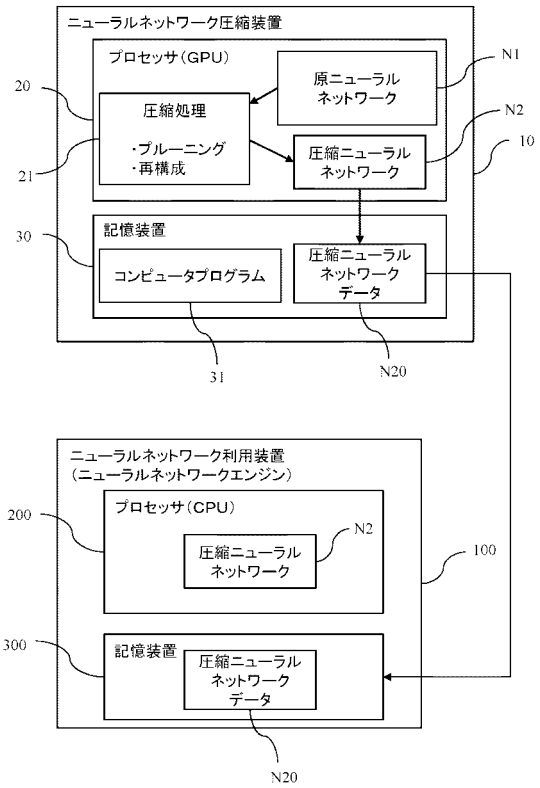
【符号の説明】

【００６２】

１０　：ニューラルネットワーク圧縮装置
２０　：プロセッサ
２１　：圧縮処理
３０　：記憶装置
３１　：コンピュータプログラム
１００　：ニューラルネットワーク利用装置
１２１　：圧縮処理
１２２　：プルーニング工程
１２３　：再構成工程
２００　：プロセッサ
３００　：記憶装置
Ｎ１　：原ニューラルネットワーク
Ｎ２　：圧縮ニューラルネットワーク
Ｎ２０　：圧縮ニューラルネットワークデータ

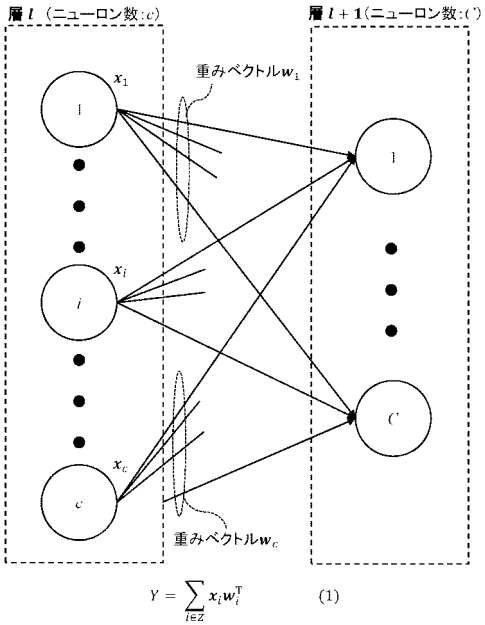
【図１】

図1



【図２】

図2



$$Y = \sum_{i \in Z} \boldsymbol{x}_i \boldsymbol{w}_i^{\mathrm{T}} \qquad (1)$$

Z:層lのニューロンを表す標識集合. $Z=\{1,\ldots,c\}$

Y:N個のデータを層lのc個のニューロンに与えたときに、
層$l+1$のc'個のニューロンの内部活性度を表す$N\times c'$の行列（層$l+1$への伝播量）.

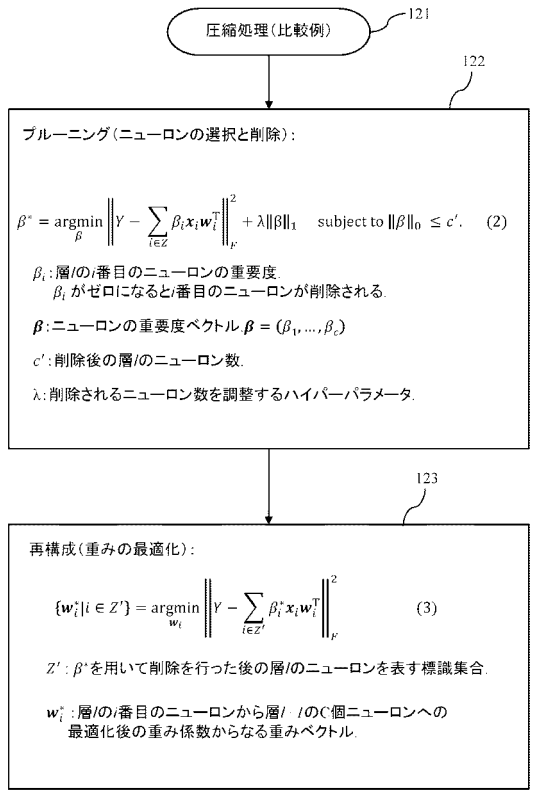
$\boldsymbol{x}_i$:N個のデータを層lのi番目のニューロンに与えたときに、
そのニューロンの出力からなるN次元ベクトル（挙動ベクトル）.

$\boldsymbol{w}_i$:層lのi番目のニューロンから層$l+1$のC個ニューロンへの
重み係数からなる重みベクトル.

$\boldsymbol{x}_i\boldsymbol{w}_i^{\mathrm{T}}$:層$l$の$i$番目のニューロンによって発生させられる
層$l+1$のc'個のニューロンの内部活性度.

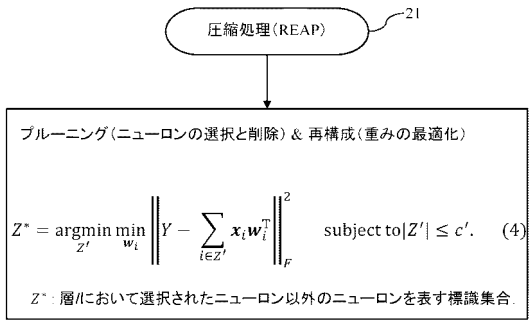
【図３】

図3



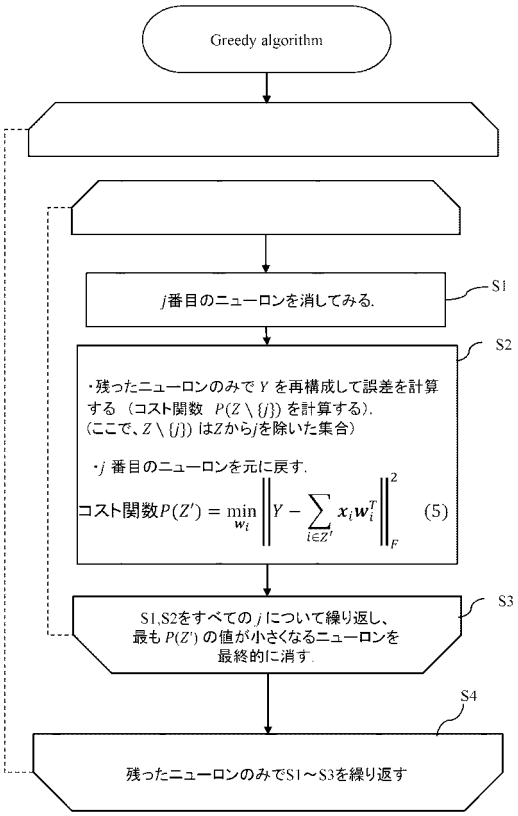
【図４】

図4



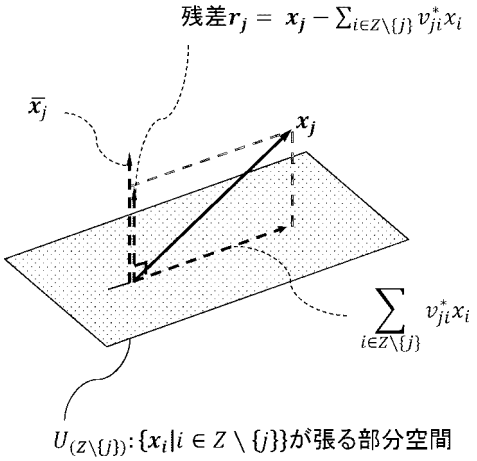
【図５】

図5



【図６】

図6



$U_{(Z \setminus \{j\})}$へのx_jの直交射影の計算は、
$\{x_i | i \in Z \setminus \{j\}\}$から$x_j$を再構成する問題を最小二乗法により解くことと等価である.
残差r_jは、x_jの双対基底である$\bar{x}_j$に線形従属である.

$$P(Z \setminus \{j\}) = \min_{w_i} \left\| Y - \sum_{i \in Z \setminus \{j\}} x_i w_i^T \right\|_F^2 \qquad (6-1)$$

$$= \left\| r_j w_j^T \right\|_F^2 \qquad (6-2)$$

r_jは、x_jの$\bar{x}_j$に対する射影であり、以下のように計算される.

$$r_j = \frac{x_j^\top \bar{x}_j}{\left\| \bar{x}_j \right\|^2} \bar{x}_j = \frac{\bar{x}_j}{\left\| \bar{x}_j \right\|^2} \qquad (6-3)$$

【図７】

図7

残差r_jの計算 (1/2)

j番目のニューロンの削除により生じる再構成誤差を計算するため、
以下のコスト関数$P(Z \setminus \{j\})$を計算する.

$$P(Z \setminus \{j\}) = \min_{w_i} \left\| Y - \sum_{i \in Z \setminus \{j\}} x_i w_i^T \right\|_F^2 \qquad (6-1)$$

部分問題の定義:

$$\{v_{ji}^* | i \in Z \setminus \{j\}\} = \underset{v_{ji}}{arg\min} \left\| x_j - \sum_{i \in Z \setminus \{j\}} v_{ji} x_i \right\|^2 \qquad (7)$$

v_{ji} : x_iからx_jを再構成するための係数

式(7)を解くと、式(6－1)の解は、式(1)より、以下のように得られる.

$$P(Z \setminus \{j\}) = \left\| \left(x_j - \sum_{i \in Z \setminus \{j\}} v_{ji}^* x_i \right) w_j^T \right\|_F^2 \qquad (8)$$

部分問題の効率的な解法:
式(7)は、$\{x_i | i \in Z \setminus \{j\}\}$から再構成された$x_j$の誤差を最小化する問題である.
$\{x_i | i \in Z \setminus \{j\}\}$から再構成された$x_j$の計算は、図6に示す部分空間$U_{(Z \setminus \{j\})}$への$x_j$の直交射影の計算と等価である.

x_jの残差をr_jで示すと、以下の式が得られる.

$$r_j = x_j - \sum_{i \in Z \setminus \{j\}} v_{ji}^* x_i \qquad (9)$$

明らかに、r_jはx_i ($i \in Z \setminus \{j\}$)に直交し、x_jには直交しない.
すなわち、r_jはx_jの双対基底に線形従属である.
このr_jの性質を利用し、計算を効率化する.

【図８】

図8

残差r_jの計算 (2/2)

ここで、
$\{\bar{x}_i | i \in Z\}$を、$\{x_i | i \in Z\}$の双対基底とする.
X $[x_1 \ \ldots \ x_c]$, $\bar{X}$ $[\bar{x}_1 \ \ldots \bar{x}_c]$ とすると、双対基底の定義より、
$\bar{X}$ は次のように与えられる.

$$\bar{X} = (X^g)^\top \qquad (10)$$

X^g : Xの一般化逆行列.

r_jはx_jの双対基底に線形従属であるため、
r_jを得るには、図6に示すように、x_jの$\bar{x}_j$への直交射影を計算すればよい.
したがって、r_jは、以下のように計算される.

$$r_j = \frac{x_j^\top \bar{x}_j}{\left\| \bar{x}_j \right\|^2} \bar{x}_j = \frac{\bar{x}_j}{\left\| \bar{x}_j \right\|^2} \qquad (6-3)$$

ここで、
Dを対角行列とする. この対角行列の(i,i)成分は$1/\|\bar{x}_i\|^2$で与えられる.
すると、
$R=[r_1 \ldots r_c]$は、以下のシンプルな行列乗算によって得られる.

$$R = \bar{X} D \qquad (11)$$

ここで、
V^*を行列とする. この行列の (j,i) 成分は v_{ji}^* $(i \neq j)$ である.
v_{ji}^* $(i \neq j)$は式(7)を解くことで得られる. $v_{ii}^* = 0$である.

式(9)より以下の方程式が成り立つ.

$$X = R + X V^{*\top} \qquad (12)$$

したがって、$V^{*\top} = I - X^g R$ が得られる。ここで、Iは単位行列である.

上記のように、とり得るjそれぞれについて、式(7)を解く代わりに、
いくつかの簡単な行列演算を行うことにより、等価な解が得られる.

【図９】

図9

別のニューロンを削除する際の再構成誤差の計算 (1/2)

ここで、j番目のニューロンが削除されたものとする.
式(4)をgreedy流に解くには、
次のステップとして、次に削除されるニューロンを選択するために、
$k(k \in Z \setminus \{j\})$について$P(Z \setminus \{j,k\})$を計算する.

単純なアプローチは、とり得るkそれぞれについて$Z' = Z \setminus \{j,k\}$とし、
式(5)で定義される最小二乗問題を解くことである.
しかし、このアプローチは計算コストが大きい.

以下に、$P(Z \setminus \{j,k\})$をより効率的に計算する方法を示す.

ここで、
$\boldsymbol{r}'_j$を$\boldsymbol{x}_j$の残差とし、$\boldsymbol{r}'_k$を$\boldsymbol{x}_k$の残差とする.
残差は、$\{\boldsymbol{x}_i | i \in Z \setminus \{j,k\}\}$から計算される.

$Z' = Z \setminus \{j,k\}$の場合、式(5)の解は、以下のように与えられる.

$$P(Z \setminus \{j,k\}) = \left\| \boldsymbol{r}'_j \boldsymbol{w}_j^{\mathrm{T}} + \boldsymbol{r}'_k \boldsymbol{w}_k^{\mathrm{T}} \right\|_F^2 \qquad (13)$$

したがって、
$\boldsymbol{r}'_j$及び$\boldsymbol{r}'_k$が得られれば$P(Z \setminus \{j,k\})$を計算できる.

$\boldsymbol{r}'_j$及び$\boldsymbol{r}'_k$は、いくつかのシンプルな線形代数のトリックによって得られる.
以下では、$\boldsymbol{r}'_k$の計算方法だけを示す. $\boldsymbol{r}'_j$は$\boldsymbol{r}'_k$と同様の方法で計算される

以下の式(14)及び(15)が得られる.

$$\boldsymbol{x}_j = \boldsymbol{r}_j + v^*_{jk}\boldsymbol{x}_k - \sum_{i \in Z \setminus \{j,k\}} v^*_{ji}\boldsymbol{x}_i \qquad (14)$$

$$\boldsymbol{x}_k = \boldsymbol{r}_k + v^*_{kj}\boldsymbol{x}_j - \sum_{i \in Z \setminus \{j,k\}} v^*_{ki}\boldsymbol{x}_i \qquad (15)$$

【図１０】

図10

別のニューロンを削除する際の再構成誤差の計算 (2/2)

j番目及びk番目のニューロンを削除した後では、
$\boldsymbol{x}_k$の再構成のために$\boldsymbol{x}_j$を利用することができない.
そこで、式(14)を式(15)に代入し、以下の式を得る.

$$\boldsymbol{x}_k = \frac{\boldsymbol{r}_k + v^*_{kj}\boldsymbol{r}_j}{1 - v^*_{jk}v^*_{kj}} + \sum_{i \in Z \setminus \{j,k\}} \frac{v^*_{ki} - v^*_{kj}v^*_{ji}}{1 - v^*_{jk}v^*_{kj}}\boldsymbol{x}_i \qquad (16)$$

明らかに、
$\boldsymbol{x}_q^{\mathrm{T}}\boldsymbol{r}_j = 0 \ (q \in Z \setminus \{j\})$
$\boldsymbol{x}_p^{\mathrm{T}}\boldsymbol{r}_k = 0 \ (p \in Z \setminus \{k\})$
であるから
$\boldsymbol{x}_i^{\mathrm{T}} \frac{\boldsymbol{r}_k + v^*_{kj}\boldsymbol{r}_j}{1 - v^*_{jk}v^*_{kj}} = 0$が、 $i \in Z \setminus \{j,k\}$について成り立つ.
このことは、式(16)の右辺の第1項が、$\{\boldsymbol{x}_i | i \in Z \setminus \{j,k\}\}$から
再構成された$\boldsymbol{x}_j$の残差を示すことを意味する.
したがって、残差$\boldsymbol{r}'_k$は以下のように表される.

$$\boldsymbol{r}'_k = \frac{\boldsymbol{r}_k + v^*_{kj}\boldsymbol{r}_j}{1 - v^*_{jk}v^*_{kj}} \qquad (17)$$

残差$\boldsymbol{r}'_j$は残差$\boldsymbol{r}'_k$と同様に計算される.
$\boldsymbol{r}'_j$及び$\boldsymbol{r}'_k$を式(13)に代入すると、$P(Z \setminus \{j,k\})$が計算される.
$P(Z \setminus \{j,k\})$は、$\{\boldsymbol{x}_i | i \in Z \setminus \{j,k\}\}$から再構成されたYの誤差を示す.

同時に、再構成のための係数を更新する.
$\{v'_{ki} | i \in Z \setminus \{j,k\}\}$は、$\{\boldsymbol{x}_i | i \in Z \setminus \{j,k\}\}$から$\boldsymbol{x}_j$を再構成するための係数を示す.
すると、以下の式が得られる.

$$v'_{ki} = \frac{v^*_{ki} - v^*_{kj}v^*_{ji}}{1 - v^*_{jk}v^*_{kj}} \qquad (18)$$

上記のように、式(5)で定義される最小二乗問題を直接解くことなく、
$P(Z \setminus \{j,k\})$を計算することができる.
したがって、$P(Z \setminus \{j,k\})$を最小化するkを決定し、
k番目のニューロンを削除することができる.
上記と同じ手順を繰り返せば、さらに別のニューロンを削除できる.

【図１１】

図11

REAPの畳み込み層への適用

aは、入力チャネル数を示し、
Aは、出力チャネル数を示し、
t_wは、特徴マップの幅を示し、
t_hは、特徴マップの高さを示し、
s_wは、重みテンソルの幅を示し、
s_hは、重みテンソルの高さを示すものとする.

畳み込み層における演算に関して、
N入力画像に対応する特徴マップを示す$N \times a \times t_w \times t_h$テンソルでの
スライディングウィンドウ操作、及び
重みを示す$A \times a \times s_w \times s_h$テンソルは、次のように表現できる.

$$Y = \sum_{i \in B} \Phi_i \psi_i^T = \sum_{i \in B} \sum_{m \in T} \Phi_{i(m)} \psi_{i(m)}^T \qquad (19)$$

ここで、
$B = \{1, \ldots, a\}$は、入力チャネルの標識集合を示す.
行列$\Phi_i \in \mathbb{R}^{Nt_w t_h \times swsh}$は、$i$番目のreshapeされた入力特徴マップを示す.
Φ_iの各行は元の特徴マップの部分テンソルを示す.
行列$\psi_i \in \mathbb{R}^{A \times swsh}$は、$i$番目のreshapeされた重みテンソルを示す.
$T = \{1, \ldots, s_w s_h\}$は、$\Phi_s$及び$\psi_s$の列標識集合を示す.
$\Phi_{i(m)}$及び$\psi_{i(m)}$は、Φ_i及びψ_iのm番目の列を示す.

畳み込み層に関しては、チャネルレベルでのプルーニング(削除)
が行われる.換言すると、i番目のチャネルの削除のために、
$\Phi_{i(m)}$及び$\psi_{i(m)}$ $(m \in T)$が同時に削除される.
他の手続については、全結合層における手続と同様である.

【図１２】

図12

グラン・シュミット(Gram-Schmit)の直交化計算の適用による射影残差$\boldsymbol{r}_j$の計算

グラン・シュミットの直交化計算で用いられる以下の射影計算を用いる.

$$\boldsymbol{F}(\boldsymbol{r}, \boldsymbol{x}) = \boldsymbol{r} - \frac{(\boldsymbol{r}, \boldsymbol{x})}{\|\boldsymbol{x}\|^2}\boldsymbol{x} \qquad (20)$$

ここでは、$\boldsymbol{x}_j$を$\boldsymbol{x}_j$以外のベクトルに順次射影する$\boldsymbol{F}(\boldsymbol{F}(\boldsymbol{F}(\boldsymbol{x}_i, \boldsymbol{x}_1), \boldsymbol{x}_2), \ldots, \boldsymbol{x}_n)$を
計算すれば、これが$\boldsymbol{r}_j$に漸近するという性質を用いる.
この性質を用いると、並列計算のための共有メモリ領域に$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$を
格納しておけば、以下のように、各射影残差$\boldsymbol{r}_j$を並列計算できる.

$$\boldsymbol{r}_1 = F(F(F(x_1, x_2), x_3), \ldots, x_n)$$

$$\boldsymbol{r}_2 = F(F(F(x_2, x_1), x_3), \ldots, x_n)$$

$$\vdots$$

$$\boldsymbol{r}_n = F(F(F(x_n, x_1), x_n), \ldots, x_{n-1})$$

【図１３】

図13