

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7537213号
(P7537213)

(45)発行日 令和6年8月21日(2024.8.21)

(24)登録日 令和6年8月13日(2024.8.13)

(51)国際特許分類 F I
G 0 6 N 3/04 (2023.01) G 0 6 N 3/04

請求項の数 7 (全19頁)

(21)出願番号	特願2020-164456(P2020-164456)	(73)特許権者	000002369 セイコーエプソン株式会社 東京都新宿区新宿四丁目1番6号
(22)出願日	令和2年9月30日(2020.9.30)	(74)代理人	110000028 弁理士法人明成国際特許事務所
(65)公開番号	特開2022-56611(P2022-56611A)	(72)発明者	倉沢 光 長野県諏訪市大和3丁目3番5号 セイ コーエプソン株式会社内
(43)公開日	令和4年4月11日(2022.4.11)	審査官	渡辺 一帆
審査請求日	令和5年7月19日(2023.7.19)		

最終頁に続く

(54)【発明の名称】 情報処理装置、演算方法、及び、コンピュータプログラム

(57)【特許請求の範囲】

【請求項1】

情報処理装置であって、
ベクトルニューラルネットワーク型の機械学習モデルを記憶するメモリと、
前記機械学習モデルを用いた演算を実行するプロセッサと、
を備え、
前記機械学習モデルは、
複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、
前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあ
るベクトルニューロン層を下位層と呼ぶとき、
前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として
、1つの出力ベクトルを出力するように構成されており、
前記上位層の任意のノードである対象ノードは、
(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求
め、
(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベ
クトルを求め、
(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求め、
(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによっ
て前記対象ノードの前記出力ベクトルを求める、

10

20

ように構成されている、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、

前記正規化係数は、前記上位層における前記正規化係数の総和が 1 になるように前記ノルムを正規化関数で正規化することによって得られる、情報処理装置。

【請求項 3】

請求項 1 又は 2 に記載の情報処理装置であって、

前記予測行列は複数準備されており、

前記上位層の各ノードの前記出力ベクトルの演算に使用される前記下位層の前記複数のノードの範囲は、前記複数の予測行列を複数の要素とするカーネルを用いた畳み込みによって制限され、

10

前記複数の予測行列は、前記機械学習モデルの学習によって決定される、情報処理装置。

【請求項 4】

請求項 1 ~ 3 のいずれか一項に記載の情報処理装置であって、

前記メモリーは、学習済みの前記機械学習モデルに複数の教師データが入力されたときに前記複数のベクトルニューロン層のうちの少なくとも 1 つの特定層の出力から得られた既知特徴スペクトル群を格納しており、

前記機械学習モデルは、学習済みの前記機械学習モデルに新たな入力データが入力されたときに前記特定層の出力から得られる特徴スペクトルと、前記既知特徴スペクトル群との類似度を演算する類似度演算部を有する、情報処理装置。

20

【請求項 5】

請求項 4 に記載の情報処理装置であって、

前記特定層は、第 1 軸と第 2 軸の 2 つの軸で規定された平面に配置されたベクトルニューロンが、前記 2 つの軸とは異なる方向の第 3 軸に沿って複数のチャンネルとして配置されている構成を有し、

前記特徴スペクトルは、

(i) 前記特定層のうちの 1 つの平面位置におけるベクトルニューロンの出力ベクトルの複数の要素値を、前記第 3 軸に沿った前記複数のチャンネルにわたって配列した第 1 種の特徴スペクトルと、

(i i) 前記第 1 種の特徴スペクトルの各要素値に、前記正規化係数を乗じることによって得られる第 2 種の特徴スペクトルと、

30

(i i i) 前記特定層のうちの 1 つの平面位置における前記正規化係数を、前記第 3 軸に沿った前記複数のチャンネルにわたって配列した第 3 種の特徴スペクトルと、

のうちのいずれかである、情報処理装置。

【請求項 6】

ベクトルニューラルネットワーク型の機械学習モデルを用いて演算処理を実行する演算方法であって、

前記機械学習モデルは、

複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、

前記複数のベクトルニューロン層のうちの 1 つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、

40

前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1 つの出力ベクトルを出力するように構成されており、

前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める工程は、

(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求める工程と、

(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求める工程と、

(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求める工程と、

(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによ

50

て前記対象ノードの前記出力ベクトルを求める工程と、
を含む、演算方法。

【請求項 7】

ベクトルニューラルネットワーク型の機械学習モデルを用いる演算処理をプロセッサ
に実行させるコンピュータプログラムであって、

前記機械学習モデルは、

複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、

前記複数のベクトルニューロン層のうちの 1 つを上位層と呼び、前記上位層の下位にあ
るベクトルニューロン層を下位層と呼ぶとき、

前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として
、 1 つの出力ベクトルを出力するように構成されており、

前記コンピュータプログラムは、

前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める処理として、

(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求
める処理と、

(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベ
クトルを求める処理と、

(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求める処理と、

(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによっ
て前記対象ノードの前記出力ベクトルを求める処理と、

を前記プロセッサに実行させる、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、機械学習モデルを用いる情報処理装置、演算方法、及び、コンピュータプ
ログラムに関する。

【背景技術】

【0002】

特許文献 1 , 2 には、ベクトルニューロンを用いる機械学習モデルとして、カプセルネ
ットワークと呼ばれるものが開示されている。ベクトルニューロンとは、入出力がベクト
ルであるニューロンを意味する。カプセルネットワークは、カプセルと呼ばれるベクトル
ニューロンをネットワークのノードとする機械学習モデルである。カプセルネットワー
クでは、各層の出力ベクトルを求める際に、動的ルーティングを複数回繰り返すことによ
って内部パラメータを探索する。

【先行技術文献】

【特許文献】

【0003】

【文献】米国特許第 5 2 1 0 7 9 8 号公報

【文献】国際公開 2 0 1 9 / 0 8 3 5 5 3 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、カプセルネットワークでは、動的ルーティングを複数回繰り返す必要が
あるため、演算の速度が遅いという問題があった。

【課題を解決するための手段】

【0005】

本開示の第 1 の形態によれば、情報処理装置が提供される。この情報処理装置は、ベク
トルニューラルネットワーク型の機械学習モデルを記憶するメモリーと、前記機械学習モ
デルを用いた演算を実行するプロセッサと、を備える。前記機械学習モデルは、複数の
ノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロ

10

20

30

40

50

ン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記上位層の任意のノードである対象ノードは、(a)前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求め、(b)前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求め、(c)前記和ベクトルのノルムを正規化することによって正規化係数を求め、(d)前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求める、ように構成されている。

【0006】

本開示の第2の形態によれば、ベクトルニューラルネットワーク型の機械学習モデルを用いて演算処理を実行する演算方法が提供される。前記機械学習モデルは、複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める工程は、(a)前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求める工程と、(b)前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求める工程と、(c)前記和ベクトルのノルムを正規化することによって正規化係数を求める工程と、(d)前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求める工程と、を含む。

【0007】

本開示の第3の形態によれば、ベクトルニューラルネットワーク型の機械学習モデルを用いる演算処理をプロセッサに実行させるコンピュータープログラムが提供される。前記機械学習モデルは、複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記コンピュータープログラムは、前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める処理として、(a)前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求める処理と、(b)前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求める処理と、(c)前記和ベクトルのノルムを正規化することによって正規化係数を求める処理と、(d)前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求める処理と、を前記プロセッサに実行させる。

【図面の簡単な説明】

【0008】

【図1】実施形態における情報処理装置のブロック図。

【図2】機械学習モデルの構成を示す説明図。

【図3】クラス分類処理の処理手順を示すフローチャート。

【図4】特徴スペクトルを示す説明図。

【図5】既知特徴スペクトル群の構成を示す説明図。

【発明を実施するための形態】

【0009】

図1は、一実施形態における情報処理装置100の機能を示すブロック図である。情報処理装置100は、プロセッサ110と、メモリー120と、インターフェイス回路130と、インターフェイス回路130に接続された入力デバイス140及び表示部150と、を有している。限定されないが例えば、プロセッサ110は、以下で詳述される処理を実行する機能を有するだけでなく、表示部150に、当該処理によって得られるデー

10

20

30

40

50

タ、および当該処理の過程で生成されるデータを表示する機能も有する。情報処理装置 100 は、パーソナルコンピュータなどのコンピュータによって実現可能である。

【0010】

プロセッサ 110 は、入力データのクラス分類処理を実行するクラス分類処理部 112 として機能する。クラス分類処理部 112 は、メモリー 120 に格納されたコンピュータプログラムをプロセッサ 110 が実行することによって実現される。但し、クラス分類処理部 112 をハードウェア回路で実現してもよい。本明細書のプロセッサは、このようなハードウェア回路をも含む用語である。メモリー 120 には、ベクトルニューラルネットワーク型の機械学習モデル 200 と、教師データ TD と、既知特徴スペクトル群 KSG と、被分類データ Di とが格納される。機械学習モデル 200 は、クラス分類処理部 112 による演算に使用される。機械学習モデル 200 の構成例や動作については後述する。教師データ TD は、機械学習モデル 200 の学習に使用されるラベル付きのデータである。既知特徴スペクトル群 KSG は、学習済みの機械学習モデル 200 に教師データ TD を再度入力した際に得られる特徴スペクトルの集合である。特徴スペクトルについては後述する。被分類データ Di は、クラス分類処理の処理対象となる新たな入力データである。なお、教師データ TD は、機械学習モデル 200 の学習時にのみ必要であり、被分類データ Di に対するクラス分類処理の実行時には不要である。また、被分類データ Di は、機械学習モデル 200 の学習時にメモリー 120 に保存されている必要はない。

10

【0011】

上述のように、情報処理装置 100 は、クラス分類処理を行うクラス分類装置としての機能を有する。但し、情報処理装置 100 は、クラス分類処理以外の他の演算処理を行うように構成することも可能であり、例えば、入力データの評価値である連続値を算出する評価値算出処理を実行するように構成することができる。一般に、プロセッサ 110 は、機械学習モデル 200 を用いた演算を実行する演算部として機能する。

20

【0012】

図 2 は、機械学習モデル 200 の構成を示す説明図である。この機械学習モデル 200 は、入力データ IM の側から順に、畳み込み層 210 と、プライマリーベクトルニューロン層 220 と、第 1 畳み込みベクトルニューロン層 230 と、第 2 畳み込みベクトルニューロン層 240 と、分類ベクトルニューロン層 250 とを備える。これらの 5 つの層 210 ~ 250 のうち、畳み込み層 210 が最も下位の層であり、分類ベクトルニューロン層 250 が最も上位の層である。以下の説明では、層 210 ~ 250 を、それぞれ「Conv層 210」、「PrimeVN層 220」、「ConvVN1層 230」、「ConvVN2層 240」、及び「ClassVN層 250」とも呼ぶ。

30

【0013】

図 2 の例では 2 つの畳み込みベクトルニューロン層 230, 240 を用いているが、畳み込みベクトルニューロン層の数は任意であり、畳み込みベクトルニューロン層を省略してもよい。但し、1 つ以上の畳み込みベクトルニューロン層を用いることが好ましい。

【0014】

図 2 の機械学習モデル 200 は、更に、類似度を生成する類似度演算部 260 を有している。類似度演算部 260 は、ConvVN1層 230 と、ConvVN2層 240 と、ClassVN層 250 の出力から、後述する類似度 S_ConvVN1, S_ConvVN2, S_ClassVN をそれぞれ算出することが可能である。但し、類似度演算部 260 を省略してもよい。

40

【0015】

各層 210 ~ 250 の構成は、以下のように記述できる。

<各層の構成の記述>

- ・Conv層 210 : Conv[32,5,2]
- ・PrimeVN層 220 : PrimeVN[16,1,1]
- ・ConvVN1層 230 : ConvVN1[12,3,2]
- ・ConvVN2層 240 : ConvVN2[6,3,1]
- ・ClassVN層 250 : ClassVN[2,4,1]

50

・ベクトル次元 VD : $VD=16$

これらの各層 2 1 0 ~ 2 5 0 の記述において、括弧前の文字列はレイヤー名であり、括弧内の数字は、順に、チャンネル数、カーネルサイズ、及び、ストライドである。例えば、Conv層 2 1 0 のレイヤー名は「Conv」であり、チャンネル数は 3 2、カーネルサイズは 5×5 、ストライドは 2 である。図 2 では、各層の下にこれらの記述が示されている。各層の中に描かれているハッチングを付した矩形は、隣接する上位層の出力ベクトルを算出する際に使用されるカーネルを表している。なお、各層 2 1 0 ~ 2 5 0 の記述で用いたパラメータの値は例示であり、任意に変更可能である。

【 0 0 1 6 】

Conv層 2 1 0 は、スカラーニューロンで構成された層である。他の 4 つの層 2 2 0 ~ 2 5 0 は、ベクトルニューロンで構成された層である。ベクトルニューロンは、ベクトルを入出力とするニューロンである。上記の記述では、個々のベクトルニューロンの出力ベクトルの次元は 1 6 で一定である。以下では、スカラーニューロン及びベクトルニューロンの上位概念として「ノード」という語句を使用する。

10

【 0 0 1 7 】

図 2 では、Conv層 2 1 0 について、ノード配列の平面座標を規定する第 1 軸 x 及び第 2 軸 y と、奥行きを表す第 3 軸 z とが示されている。また、Conv層 2 1 0 の x , y , z 方向のサイズが 1 3 , 1 3 , 3 2 であることが示されている。 x 方向のサイズと y 方向のサイズを「解像度」と呼ぶ。この例では、 x 方向と y 方向の解像度は等しいが、異なっても良い。 z 方向のサイズは、チャンネル数である。これらの 3 つの軸 x , y , z は、他の層においても各ノードの位置を示す座標軸として使用する。但し、図 2 では、Conv層 2 1 0 以外の層では、これらの軸 x , y , z の図示が省略されている。

20

【 0 0 1 8 】

よく知られているように、畳み込み後の解像度 W_1 は、次式で与えられる。

$$W_1 = \text{Ceil}\{(W_0 - W_k + 1) / S\} \quad (1)$$

ここで、 W_0 は畳み込み前の解像度、 W_k はカーネルサイズ、 S はストライド、 $\text{Ceil}\{X\}$ は X を切り上げる演算を行う関数である。

図 2 に示した各層の解像度は、入力データ IM の解像度を 29×29 画素とした場合の例であり、実際の各層の解像度は入力データ IM のサイズに応じて適宜変更される。なお、図 2 の例では入力データ IM は 2 次元配列のデータであるが、1 次元配列のデータとしてもよいし、3 次元以上の配列データとしてもよい。

30

【 0 0 1 9 】

PrimeVN層 2 2 0 の各ノードは、Conv層 2 1 0 の $1 \times 1 \times 32$ 個のノードのスカラー出力を 3 2 次元のベクトルとみなして、このベクトルに変換行列を乗ずることによってそのノードのベクトル出力を得る。この変換行列は、 1×1 のカーネルの要素であり、機械学習モデル 2 0 0 の学習によって更新される。なお、Conv層 2 1 0 と PrimeVN層 2 2 0 の処理を統合して、1 つのプライマリーベクトルニューロン層として構成することも可能である。

【 0 0 2 0 】

PrimeVN層 2 2 0 を「下位層 L 」と呼び、その上位側に隣接する ConvVN1層 2 3 0 を「上位層 $L + 1$ 」と呼ぶとき、上位層 $L + 1$ の各ノードの出力は、以下の式を用いて決定される。

40

【数 1】

[Math. 1]

$$\mathbf{v}_{ij} = \mathbf{W}_{ij}^L \mathbf{M}_i^L \quad (2)$$

$$\mathbf{u}_j = \sum_i \mathbf{v}_{ij} \quad (3)$$

$$a_j = F(\|\mathbf{u}_j\|) \quad (4)$$

$$\mathbf{M}_j^{L+1} = a_j \times \frac{1}{\|\mathbf{u}_j\|} \mathbf{u}_j \quad (5)$$

10

ここで、

\mathbf{M}_i^L は、下位層Lにおけるi番目のノードの出力ベクトル、

\mathbf{M}_j^{L+1} は、上位層L+1におけるj番目のノードの出力ベクトル、

\mathbf{v}_{ij} は、出力ベクトル \mathbf{M}_j^{L+1} の予測ベクトル、

\mathbf{W}_{ij}^L は、下位層Lの出力ベクトル \mathbf{M}_i^L から予測ベクトル \mathbf{v}_{ij} を算出するための予測行列、

\mathbf{u}_j は、予測ベクトル \mathbf{v}_{ij} の和、すなわち線形結合、である和ベクトル、

a_j は、和ベクトル \mathbf{u}_j のノルム $\|\mathbf{u}_j\|$ を正規化することによって得られる正規化係数であるアクティベーション値、

20

$F(X)$ は、 X を正規化する正規化関数である。

【0021】

正規化関数 $F(X)$ としては、例えば以下の(4a)式または(4b)式を使用できる。

【数2】

[Math.2]

$$a_j = F(\|\mathbf{u}_j\|) = \text{softmax}(\|\mathbf{u}_j\|) = \frac{\exp(\beta\|\mathbf{u}_j\|)}{\sum_k \exp(\beta\|\mathbf{u}_k\|)} \quad (4a)$$

30

$$a_j = F(\|\mathbf{u}_j\|) = \frac{\|\mathbf{u}_j\|}{\sum_k \|\mathbf{u}_k\|} \quad (4b)$$

ここで、

k は、上位層L+1のすべてのノードに対する序数、

β は、任意の正の係数である調整パラメータであり、例えば $\beta = 1$ である。

【0022】

上記(4a)式では、上位層L+1のすべてのノードに関して和ベクトル \mathbf{u}_j のノルム $\|\mathbf{u}_j\|$ をソフトマックス関数で正規化することによってアクティベーション値 a_j が得られる。

40

一方、(4b)式では、和ベクトル \mathbf{u}_j のノルム $\|\mathbf{u}_j\|$ を、上位層L+1のすべてのノードに関するノルム $\|\mathbf{u}_k\|$ の和で除算することによってアクティベーション値 a_j が得られる。なお、正規化関数 $F(X)$ としては、(4a)式や(4b)式以外の他の関数を用いてもよい。

【0023】

上記(3)式の序数 i は、上位層L+1におけるj番目のノードの出力ベクトル \mathbf{M}_i^{L+1} を決めるために使用される下位層Lのノードに便宜上割り振られるものであり、1~nの値をとる。また、整数 n は、上位層L+1におけるj番目のノードの出力ベクトル \mathbf{M}_i^{L+1} を決めるために使用される下位層Lのノードの数である。従って、整数 n は次式で与えられる。

$$n = N_k \times N_c \quad (6)$$

50

ここで、 N_k はカーネルの要素数であり、 N_c は下位層であるPrimeVN層 2 2 0 のチャンネル数である。図 2 の例では $N_k = 9$ 、 $N_c = 16$ なので、 $n = 144$ である。

【0024】

ConvVN1層 2 3 0 の出力ベクトルを求めるために使用される 1 つのカーネルは、カーネルサイズ 3×3 を表面サイズとし、下位層のチャンネル数 16 を深さとする $3 \times 3 \times 16 = 144$ 個の要素を有しており、これらの要素のそれぞれは予測行列 W_{ij}^L である。また、ConvVN1層 2 3 0 の 12 個のチャンネルの出力ベクトルを生成するためには、このカーネルが 12 組必要である。従って、ConvVN1層 2 3 0 の出力ベクトルを求めるために使用されるカーネルの予測行列 W_{ij}^L の数は、 $144 \times 12 = 1728$ 個である。これらの予測行列 W_{ij}^L は、機械学習モデル 2 0 0 の学習により更新される。

10

【0025】

上述した (2) ~ (5) 式から分かるように、上位層 $L + 1$ の個々のノードの出力ベクトル M^{L+1}_j は、以下の演算によって求められる。

(a) 下位層 L の各ノードの出力ベクトル M^L_i に予測行列 W_{ij}^L を乗じて予測ベクトル v_{ij} を求め、

(b) 下位層 L の各ノードから得られた予測ベクトル v_{ij} の和、すなわち線形結合、である和ベクトル u_j を求め、

(c) 和ベクトル u_j のノルム $|u_j|$ を正規化することによって正規化係数であるアクティベーション値 a_j を求め、

(d) 和ベクトル u_j をノルム $|u_j|$ で除算し、更に、アクティベーション値 a_j を乗じる。

20

【0026】

なお、アクティベーション値 a_j は、上位層 $L + 1$ のすべてのノードに関してノルム $|u_j|$ を正規化することによって得られる正規化係数である。従って、アクティベーション値 a_j は、上位層 $L + 1$ 内の全ノードの中における各ノードの相対的な出力強度を示す指標と考えることができる。(4) 式、(4a) 式、(4b) 式、及び (5) 式で使用されるノルムは、典型的な例ではベクトル長さを表す $L2$ ノルムである。このとき、アクティベーション値 a_j は、出力ベクトル M^{L+1}_j のベクトル長さに相当する。アクティベーション値 a_j は、上述した (4) 式と (5) 式で使用されるだけなので、ノードから出力される必要は無い。但し、アクティベーション値 a_j を外部に出力するように上位層 $L + 1$ を構成することも可能である。

30

【0027】

ベクトルニューラルネットワークの構成は、カプセルネットワークの構成とほぼ同じであり、ベクトルニューラルネットワークのベクトルニューロンがカプセルネットワークのカプセルに相当する。但し、ベクトルニューラルネットワークで使用される上述の (2) ~ (5) 式による演算は、カプセルネットワークで使用される演算と異なる。両者の最も大きな違いは、カプセルネットワークでは、上記 (3) 式の右辺の予測ベクトル v_{ij} にそれぞれ重みが乗じられており、その重みが、動的ルーティングを複数回繰り返すことによって探索される点である。一方、本実施形態のベクトルニューラルネットワークでは、上述した (2) ~ (5) 式を順番に 1 回計算することによって出力ベクトル M^{L+1}_j が得られるので、動的ルーティングを繰り返す必要が無く、演算がより高速であるという利点がある。また、本実施形態のベクトルニューラルネットワークは、カプセルネットワークよりも演算に必要とするメモリー量がカプセルネットワークより少なく、本開示の発明者の実験によれば、約 $1/2 \sim 1/3$ のメモリー量で済むという利点もある。

40

【0028】

ベクトルを入出力とするノードを使用するという点では、ベクトルニューラルネットワークはカプセルネットワークと同じである。従って、ベクトルニューロンを使用する利点もカプセルネットワークと共通している。また、複数の層 2 1 0 ~ 2 5 0 は、上位に行くほどより大きな領域の特徴を表現し、下位に行くほどより小さな領域の特徴を表現する、という点は、通常の畳み込みニューラルネットワークと同じである。ここで、「特徴」とは、ニューラルネットワークへの入力データに含まれている特徴的な部分を意味する。ベ

50

クトルニューラルネットワークやカプセルネットワークでは、或るノードの出力ベクトルが、そのノードが表現する特徴の空間的な情報を表す空間情報を含む点で、通常の畳み込みニューラルネットワークよりも優れている。すなわち、或るノードの出力ベクトルのベクトル長さは、そのノードが表現する特徴の存在確率を表し、ベクトル方向がその特徴の方向やスケール等の空間情報を表している。従って、同じ層に属する2つのノードの出力ベクトルのベクトル方向は、それぞれの特徴の位置関係を表す。あるいは、当該2つのノードの出力ベクトルのベクトル方向は、特徴のバリエーションを表わすとも言える。例えば、「目」の特徴に対応するノードなら、出力ベクトルの方向は、目の細さ、吊り上がり方、などのバリエーションを表し得る。通常の畳み込みニューラルネットワークでは、プーリング処理によって特徴の空間情報が消失してしまうとされている。この結果、ベクトルニューラルネットワークやカプセルネットワークは、通常の畳み込みニューラルネットワークに比べて入力データを識別する性能に優れているという利点がある。

10

【0029】

ベクトルニューラルネットワークの利点は、以下のように考えることも可能である。すなわち、ベクトルニューラルネットワークでは、ノードの出力ベクトルが、入力データの特徴を連続空間内の座標として表現すること、に利点がある。従って、ベクトル方向が近ければ特徴が似ている、というように出力ベクトルを評価できる。また、入力データに含まれている特徴が教師データではカバーできていなくても、補間してその特徴を判別できる、などの利点もある。一方、通常の畳み込みニューラルネットワークは、プーリング処理によって無秩序な圧縮がかかるため、入力データの特徴を連続空間内の座標として表現できない、という欠点がある。

20

【0030】

ConvVN層240とClassVN層250の各ノードの出力も、上述した(2)~(5)式を用いて同様に決定されるので、詳細な説明は省略する。最上位層であるClassVN層250の解像度は 1×1 であり、チャンネル数は2である。ClassVN層250のチャンネル数は、通常は教師データで使用されるラベルの数と等しくなるように設定される。

【0031】

各層210~250のノードの出力を求める方法は、以下のように説明することも可能である。入力データIMに対して、 5×5 のカーネルをストライド「2」で適用することで、入力データIMの範囲のうちでConv層210の一つのノードに出力を与える部分範囲が決まる。入力データIMに対して適用するカーネルの数は32である。これにより、Conv層210は、第1軸xと第2軸yがそれぞれ13個に区切られた領域を有するように構成される。また、Conv層210の深さであるチャンネル数はカーネルの数と同数の32で構成される。「部分範囲」とは、入力データIM上の領域であって、第1軸xの位置と第2軸yの位置で特定される1つの領域である。ただし、以下の説明から明らかなように、「部分範囲」の大きさは、「部分範囲」に対応する1つ以上のノード又は当該1つ以上のノードから構成される「部分領域Rn」が、ベクトルニューロン層220, 230, 240, 250のどれに属するかに応じて異なる。他方、「部分領域Rn」とは、ベクトルニューロン層において第1軸xの位置と第2軸yの位置とで特定される領域である。ベクトルニューロン層における各「部分領域Rn」は、上記第1軸x、第2軸x、および第3軸zに対応する「Width」×「Height」×「Depth」の次元を有する。本実施形態では、1つの「部分領域Rn」に含まれるノードの数は「 $1 \times 1 \times$ デプス数」、すなわち「 $1 \times 1 \times$ チャンネル数」である。本明細書において、部分領域Rnの添え字「n」には、ベクトルニューロン層220, 230, 240, 250に応じて、「220」、「230」、「240」、「250」の数値を代入する。例えば、部分領域R220は、PrimeVN層220における領域を示す。

30

40

【0032】

Conv層210に $1 \times 1 \times 32$ のカーネルをストライド「1」で適用することで、Conv層210の部分領域R210のうちから、PrimeVN層220の一つのノードに出力する部分領域R210が決まる。ここでは、同じサイズ、同じストライドで、16種類のカーネ

50

ルが用いられることから、PrimeVN層 2 2 0 において、Conv層 2 1 0 の一つの部分領域 R 2 1 0 に対応するノードの数は 1 6 である。Conv層 2 1 0 のノードからPrimeVN層 2 2 0 のノードへの出力の生成には、変換行列が用いられており、上記の式 (2) ~ 式 (5) で示される出力決定アルゴリズムが用いられていない。なお、ベクトルニューロン層に畳み込むためのカーネルの次元は、チャンネル数およびベクトルの要素数も考慮に入れる場合には、「Width」×「Height」×「Depth」×「ベクトルの要素数」と表現されることもある。この表現にしたがうと、Conv層 2 1 0 からPrimeVN層 2 2 0 への畳み込みに用いられるカーネルの次元は、 $1 \times 1 \times 3 \times 16$ である。

【 0 0 3 3 】

PrimeVN層 2 2 0 に $3 \times 3 \times 16$ のカーネルをストライド「 2 」で適用することで、PrimeVN層 2 2 0 の部分領域 R 2 2 0 のうちから、ConvVN1層 2 3 0 の一つの部分領域 R 2 3 0 に含まれるノードに出力を与える部分領域 R 2 2 0 が決まる。ここでは、同じサイズ、同じ次元、同じストライドで、12 種類のカーネルが用いられることから、ConvVN1層 2 3 0 の部分領域 R 2 3 0 に含まれるノードの数は 12 である。PrimeVN層 2 2 0 のノードからConvVN1層 2 3 0 のノードへの出力の生成には、上記の式 (2) ~ 式 (5) で示される出力決定アルゴリズムが用いられる。ここで、下位層 2 2 0 に適用されるカーネルは、上位層 2 3 0 の一つのノードを決めるために用いる下位層 2 2 0 の $3 \times 3 \times 16$ 個のノードを指定する、とも表現される。これは、以下の説明でもあてはまる。

【 0 0 3 4 】

ConvVN1層 2 3 0 に $3 \times 3 \times 12$ のカーネルをストライド「 1 」で適用することで、ConvVN1層 2 3 0 の部分領域 R 2 3 0 のうちから、ConvVN2層 2 4 0 の一つの部分領域 R 2 4 0 に出力を与える部分領域 R 2 3 0 が決まる。ここでは、同じサイズ、同じ次元、同じストライドで、6 種類のカーネルが用いられることから、ConvVN2層 2 4 0 の部分領域 R 2 4 0 に含まれるノードの数は 6 である。ConvVN1層 2 3 0 のノードからConvVN2層 2 4 0 のノードを生成する際には、上記の式 (2) ~ 式 (5) で示される出力決定アルゴリズムが用いられる。

【 0 0 3 5 】

ConvVN2層 2 4 0 に $4 \times 4 \times 6$ のカーネルをストライド「 1 」で適用することで、ConvVN2層 2 4 0 の部分領域 R 2 4 0 のうちから、ClassVN層 2 5 0 の一つの部分領域 R 2 5 0 に出力を与える部分領域 R 2 4 0 が決まる。ここでは、同じサイズ、同じ次元、同じストライドで、2 種類のカーネルが用いられることから、ClassVN層 2 5 0 の部分領域 R 2 5 0 に含まれるノードの数は 2 である。ConvVN2層 2 4 0 のノードからClassVN層 2 5 0 のノードを生成する際には、上記の式 (2) ~ 式 (5) で示される出力決定アルゴリズムが用いられる。

【 0 0 3 6 】

最上位層であるClassVN層 2 5 0 は 1 つの部分領域 R 2 5 0 で構成される。ClassVN層 2 5 0 は、機械学習モデル 2 0 0 に入力された入力データ IM について、予め定められたラベルに分類する。本実施形態では、予め定められたラベルは、ラベル「 0 」とラベル「 1 」とである。ClassVN層 2 5 0 では、2 つのノードのうち、アクティベーション値 a_j が最大となるノードに対応するラベルが出力となる。ClassVN層 2 5 0 から出力されるラベルは、プロセッサ 1 1 0 によって制御されることで表示部 1 5 0 によって出力される。

【 0 0 3 7 】

上記図 2 において、PrimeVN層 2 2 0 の 1 つの部分領域 R 2 2 0 は、入力データ IM の 5×5 ピクセルの部分範囲に対応する。また、ConvVN1層 2 3 0 の 1 つの部分領域 R 2 3 0 は、入力データ IM の 9×9 ピクセルの部分範囲に対応する。また、ConvVN2層 2 4 0 の部分領域 R 2 4 0 は、入力データ IM の 17×17 ピクセルの部分範囲に対応する。また、ClassVN層 2 5 0 の部分領域 R 2 5 0 は、入力データ IM の 29×29 ピクセルの部分範囲に対応する。

【 0 0 3 8 】

図 3 は、クラス分類処理の処理手順を示すフローチャートである。最初の 2 つのステッ

10

20

30

40

50

プS 1 1 0 ~ S 1 2 0 は、教師データTDを用いて機械学習モデル2 0 0の学習を行う準備工程S 1 0 0である、次の3つのステップS 2 1 0 ~ S 2 3 0は、学習済みの機械学習モデル2 0 0を使用して被分類データDiのクラスを判別するクラス分類工程S 2 0 0である。準備工程S 1 0 0とクラス分類工程S 2 0 0を連続して行う必要はなく、クラス分類工程S 2 0 0は、準備工程S 1 0 0の後の任意のタイミングで実行することが可能である。

【0 0 3 9】

図3のステップS 1 1 0では、クラス分類処理部1 1 2が、複数の教師データTDを用いて機械学習モデル2 0 0の学習を実行する。個々の教師データTDには、予めラベルが付与されている。本実施形態では、0と1のいずれかのラベルが、個々の教師データTDに付与されているものと仮定する。

10

【0 0 4 0】

ClassVN層2 5 0の出力は、ラベルの数と等しい個数のクラスに対する複数の判定値に変換されるが、図2ではその図示を省略している。本実施形態では、ClassVN層2 5 0の2つのノードの出力が、2つのラベル「0」「1」に対応する2つのクラスに対する判定値に変換される。これらの判定値は、通常はソフトマックス関数によって正規化された値である。具体的には、例えば、ClassVN層2 5 0の各ノードの出力ベクトルから、その出力ベクトルのベクトル長さを算出し、更に、2つのノードのベクトル長さをソフトマックス関数で正規化する、という演算を実行することによって、2つのクラスに対する判定値を得ることができる。上述したように、上記(4)式で得られるアクティベーション値 a_j は、出力ベクトル M^{L+1}_j のベクトル長さに相当する値であり、正規化されている。従って、ClassVN層2 5 0の2つのノードのそれぞれにおけるアクティベーション値 a_j を出力して、そのまま2つのクラスに対する判定値として使用してもよい。

20

【0 0 4 1】

複数の教師データTDを用いた学習が終了すると、学習済みの機械学習モデル2 0 0がメモリー1 2 0に保存される。図3のステップS 1 2 0では、学習済みの機械学習モデル2 0 0に複数の教師データTDを再度入力して、既知特徴スペクトル群K S Gを生成する。既知特徴スペクトル群K S Gは、以下で説明する特徴スペクトルの集合である。

【0 0 4 2】

図4は、学習済みの機械学習モデル2 0 0に任意の入力データを入力することによって得られる特徴スペクトルSpを示す説明図である。ここでは、ConvVN1層2 3 0の出力から得られる特徴スペクトルSpについて説明する。図4の横軸は、ConvVN1層2 3 0の1つの平面位置(x, y)におけるノードの出力ベクトルの要素番号NDと、チャンネル番号NCとの組み合わせで表されるスペクトル位置である。本実施形態では、ノードのベクトル次元が16なので、出力ベクトルの要素番号NDは0から15までの16個である。また、ConvVN1層2 3 0のチャンネル数は12なので、チャンネル番号NCは0から11までの12個である。

30

【0 0 4 3】

図4の縦軸は、各スペクトル位置での特徴値 C_V を示す。この例では、特徴値 C_V は、出力ベクトルの各要素の値 V_{ND} である。なお、特徴値 C_V としては、出力ベクトルの各要素の値 V_{ND} と、そのノードのアクティベーション値 a_j とを乗算した値 $V_{ND} \times a_j$ を使用してもよく、或いは、アクティベーション値 a_j をそのまま使用してもよい。後者の場合には、特徴スペクトルSpに含まれる特徴値 C_V の数はチャンネル数に等しく、12個である。

40

【0 0 4 4】

1つの入力データに対してConvVN1層2 3 0の出力から得られる特徴スペクトルSpの数は、ConvVN1層2 3 0の平面位置(x, y)の数に等しいので、 $6 \times 6 = 36$ 個である。同様に、1つの入力データに対して、ConvVN2層2 4 0の出力から16個の特徴スペクトルSpが得られ、ClassVN層2 5 0の出力から1個の特徴スペクトルSpが得られる。

【0 0 4 5】

50

類似度演算部 260 は、学習済みの機械学習モデル 200 に教師データ TD が再度入力されたときに、図 4 に示す特徴スペクトル Sp を算出して、既知特徴スペクトル群 KSG に登録する。

【0046】

図 5 は、既知特徴スペクトル群 KSG の構成を示す説明図である。この例では、既知特徴スペクトル群 KSG は、ConvVN1 層 230 の出力から得られた既知特徴スペクトル群 KSG_ConvVN1 と、ConvVN2 層 240 の出力から得られた既知特徴スペクトル群 KSG_ConvVN2 と、ClassVN 層 250 の出力から得られた既知特徴スペクトル群 KSG_ConvVN1 とを含んでいる。

【0047】

既知特徴スペクトル群 KSG_ConvVN1 の個々のレコードは、レコード番号と、レイヤー名と、ラベル Lb と、既知特徴スペクトル KSp とを含んでいる。また、個々のレコードは、教師データ TD の個別のデータ名や、入力データ IM において特徴スペクトル Sp に対応する部分の左上の座標、などの他の項目を含んでいてもよい。既知特徴スペクトル KSp は、教師データ TD の入力に応じて得られた図 4 の特徴スペクトル Sp と同じものである。図 5 の例では、複数の教師データ TD を学習済みの機械学習モデル 200 に入力することによって、ConvVN1 層 230 の出力から、ラベル Lb = 0 に関連づけられた N1_0max 個の既知特徴スペクトル KSp と、ラベル Lb = 1 に関連づけられた N1_1max 個の既知特徴スペクトル KSp と、が得られて登録されている。N1_0max, N1_1max は、それぞれ 2 以上の整数である。前述したように、ラベル Lb = 0 とラベル Lb = 1 は、互いに異なるクラスに対応する。従って、既知特徴スペクトル群 KSG_ConvVN1 における個々の既知特徴スペクトル KSp は、複数のクラスのうちの 1 つのクラスに関連付けられて登録されていることが理解できる。他の既知特徴スペクトル群 KSG_ConvVN2, KSG_ConvVN1 も同様である。

【0048】

なお、ステップ S120 で使用される複数の教師データ TD は、ステップ S110 で使用された複数の教師データ TD と同じものである必要はない。但し、ステップ S120 においても、ステップ S110 で使用された複数の教師データ TD の一部又は全部を利用すれば、新たな教師データを準備する必要が無いという利点がある。

【0049】

図 3 のステップ S210 では、クラス分類処理部 112 が、機械学習モデル 200 に被分類データ Di を入力して、被分類データ Di のクラス分類処理を実行する。本実施形態では、被分類データ Di が、2 つのラベル Lb = 0, Lb = 1 のいずれのクラスに分類されるかが判定される。ClassVN 層 250 からの 2 つの出力は、ラベル Lb = 0 のクラスに該当する確率を示す第 1 の判定値と、ラベル Lb = 1 のクラスに該当する確率を示す第 2 の判定値である。クラス分類処理部 112 は、これらの 2 つの判定値のうち、予め定められた閾値よりも大きな判定値に対応するクラスを、クラス分類処理の結果としてユーザーに提示する。

【0050】

図 3 のステップ S220 では、類似度演算部 260 が、ConvVN1 層 230 と、ConvVN2 層 240 と、ClassVN 層 250 の出力から、既知特徴スペクトル群 KSG との類似度 S_ConvVN1, S_ConvVN2, S_ClassVN をそれぞれ算出する。以下では、ConvVN1 層 230 の出力から類似度 S_ConvVN1 を算出する方法を説明する。

【0051】

類似度演算部 260 は、既知特徴スペクトル群 KSG との類似度 S_ConvVN1 として、類似度画像 S_ConvVN1_M と、クラス別類似度 S_ConvVN1_C の 2 種類の類似度のいずれかを算出することができる。なお、図 1 では、これらの符号の末尾「_M」「_C」は省略している。

【0052】

類似度画像 S_ConvVN1_M の各画素位置 (x, y) における類似度 S(x, y) は、

10

20

30

40

50

図 5 に示した既知特徴スペクトル群 $K S G$ を用いて、次式に従って求めることができる。

$$S(x,y)=\max[G\{Sp(x,y),KSp(j)\}] \quad (7)$$

ここで、 $G\{a,b\}$ は a と b の類似度を求める関数、 $Sp(x,y)$ は被分類データ D_i に応じて得られる ConvVN1 層 230 の平面位置 (x,y) での特徴スペクトル、 $KSp(j)$ は ConvVN1 層 230 に関連付けられたすべての既知特徴スペクトル、 $\max[X]$ は X の最大値を取る論理演算を示す。すなわち、各画素位置 (x,y) における類似度 $S(x,y)$ は、被分類データ D_i に応じて得られた特徴スペクトル $Sp(x,y)$ と、同じ ConvVN1 層 230 で得られていたすべての既知特徴スペクトル $KSp(j)$ との間の類似度のうちの最大値である。

【0053】

類似度を求める関数 $G\{a,b\}$ としては、例えば、コサイン類似度を求める式や、距離に応じた類似度を求める式を使用できる。なお、各位置 (x,y) での画素値は、類似度 $S(x,y)$ の他に、上記 (7) 式において最大値を与えた既知特徴スペクトル $KSp(j)$ に関連づけられたラベル L_b も含む形で保存される。類似度画像 $S_ConvVN1_M$ の類似度 $S(x,y)$ は、その位置 (x,y) に対応する被分類データ D_i の画素位置に、そのラベル L_b に対応するクラスの特徴が存在する確率を表している。換言すれば、類似度 $S(x,y)$ は、その層の平面位置 (x,y) における特徴が、複数のクラスのうちの一つのいずれかのクラスの特徴と類似する程度を示す指標である。

【0054】

一方、クラス別類似度 $S_ConvVN1_C$ は、例えば次式を用いて算出できる。

$$S_ConvVN1_C(Class)=\max[G\{Sp(i,j),KSp(Class,k)\}] \quad (8)$$

ここで、“Class” は複数のクラスに対する序数、 $G\{a,b\}$ は a と b の類似度を求める関数、 $Sp(i,j)$ は被分類データ D_i に応じて得られるすべての平面位置 (i,j) での特徴スペクトル、 $KSp(Class,k)$ は、ConvVN1 層 230 と特定の “Class” とに関連付けられたすべての既知特徴スペクトル、 $\max[X]$ は X の最大値を取る論理演算を示す。すなわち、クラス別類似度 $S_ConvVN1_C$ は、ConvVN1 層 230 のすべての平面位置 (i,j) における特徴スペクトル $Sp(i,j)$ のそれぞれと、特定のクラスに対応するすべての既知特徴スペクトル $KSp(k)$ のそれぞれとの間で算出された類似度のうちの最大値である。このようなクラス別類似度 $S_ConvVN1_C$ は、複数のラベル L_b に対応する複数のクラスのそれぞれに対して求められる。クラス別類似度 $S_ConvVN1_C$ は、被分類データ D_i が、そのクラスの特徴に類似している程度を表している。

【0055】

ConvVN2 層 240 と ClassVN 層 250 の出力に関する類似度 $S_ConvVN2$ 、 $S_ClassVN$ も、類似度 $S_ConvVN1$ と同様に生成される。なお、これらの 3 つの類似度 $S_ConvVN1$ 、 $S_ConvVN2$ 、 $S_ClassVN$ をすべて生成する必要はないが、これらのうちの 1 つ以上を生成することが好ましい。本開示において、類似度の生成に使用された層を、「特定層」とも呼ぶ。

【0056】

図 3 のステップ S230 では、クラス分類処理部 112 が、既知特徴スペクトル群 $K S G$ との類似度 $S_ConvVN1$ 、 $S_ConvVN2$ 、 $S_ClassVN$ をユーザーに提示し、ユーザーは、これに応じてクラス分類処理の結果を確認する。上述したように、既知特徴スペクトル群 $K S G$ との類似度 $S_ConvVN1$ 、 $S_ConvVN2$ 、 $S_ClassVN$ は、被分類データ D_i が、いずれかのラベルに対応するクラスの特徴に類似している程度を表しているため、これらの類似度 $S_ConvVN1$ 、 $S_ConvVN2$ 、 $S_ClassVN$ の少なくとも 1 つから、クラス分類処理の結果の良否を確認することができる。なお、クラス分類工程 S200 の 3 つのステップ S210 ~ S230 は、実際にはほぼ同時に実行される。

【0057】

以上のように、本実施形態では、上位層 $L+1$ の個々のノードの出力ベクトル M^{L+1}_j は、(a) 下位層 L の各ノードの出力ベクトル M^L_i と予測行列 W^{L+1}_{ij} との積に基づく予測ベクトル v_{ij} を求め、(b) 下位層 L の各ノードから得られた予測ベクトル v_{ij} の線形結合であ

10

20

30

40

50

る和ベクトル u_j を求め、(c) 和ベクトル u_j のノルム $|y|$ を正規化することによって正規化係数であるアクティベーション値 a_j を求め、(d) 和ベクトル u_j をノルム $|y|$ で除算し、更に、アクティベーション値 a_j を乗じる、という演算によって求められる。従って、カプセルネットワークのように動的ルーチンを複数回実行する必要がないので、機械学習モデル 200 を用いた演算をより高速に実行できるという利点がある。

【0058】

なお、既知スペクトル群 K S G の生成方法や、ConvVN1 層等の中間層の出力データの生成方法は上記実施形態に限定されるものではなく、例えば、Kmeans 法を用いてこれらのデータを生成してもよい。また、PCA や ICA、Fisher などの変換を用いてこれらのデータを生成してもよい。また、既知スペクトル群 K S G と中間層の出力データの変換方法は異なってもよい。

10

【0059】

・他の実施形態：

本開示は、上述した実施形態に限られるものではなく、その趣旨を逸脱しない範囲において種々の形態で実現することができる。例えば、本開示は、以下の形態 (aspect) によっても実現可能である。以下に記載した各形態中の技術的特徴に対応する上記実施形態中の技術的特徴は、本開示の課題の一部又は全部を解決するために、あるいは、本開示の効果の一部又は全部を達成するために、適宜、差し替えや、組み合わせを行うことが可能である。また、その技術的特徴が本明細書中に必須なものとして説明されていなければ、適宜、削除することが可能である。

20

【0060】

(1) 本開示の第1の形態によれば、情報処理装置が提供される。この情報処理装置は、ベクトルニューラルネットワーク型の機械学習モデルを記憶するメモリーと、前記機械学習モデルを用いた演算を実行するプロセッサと、を備える。前記機械学習モデルは、複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記上位層の任意のノードである対象ノードは、(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求め、(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求め、(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求め、(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求め、ように構成されている。

30

この情報処理装置によれば、カプセルネットワークのように動的ルーチンを複数回実行する必要がないので、機械学習モデルを用いた演算をより高速に実行できる。

【0061】

(2) 上記情報処理装置において、前記正規化係数は、前記上位層における前記正規化係数の総和が1になるように前記ノルムを正規化関数で正規化することによって得られるものとしてもよい。

40

この情報処理装置によれば、簡単な演算で適切な正規化係数を求めることができる。

【0062】

(3) 上記情報処理装置において、前記予測行列は複数準備されており、前記上位層の各ノードの前記出力ベクトルの演算に使用される前記下位層の前記複数のノードの範囲は、前記複数の予測行列を複数の要素とするカーネルを用いた畳み込みによって制限され、前記複数の予測行列は、前記機械学習モデルの学習によって決定されるものとしてもよい。

この情報処理装置によれば、カーネルによって演算の範囲が制限されているので予測行列の数が少なくすみ、また、学習によって適切な予測行列を決定できる。

【0063】

(4) 上記情報処理装置において、前記メモリーは、学習済みの前記機械学習モデルに複

50

数の教師データが入力されたときに前記複数のベクトルニューロン層のうちの少なくとも1つの特定層の出力から得られた既知特徴ベクトル群を格納しており、前記機械学習モデルは、学習済みの前記機械学習モデルに新たな入力データが入力されたときに前記特定層の出力から得られる特徴ベクトルと、前記既知特徴ベクトル群との類似度を演算する類似度演算部を有するものとしてもよい。

この情報処理装置によれば、特徴ベクトルの類似度を利用して、入力データが複数の教師データのいずれに類似するかを確認できる。

【0064】

(5) 上記情報処理装置において、前記特定層は、第1軸と第2軸の2つの軸で規定された平面に配置されたベクトルニューロンが、前記2つの軸とは異なる方向の第3軸に沿って複数のチャンネルとして配置されている構成を有し、前記特徴ベクトルは、(i) 前記特定層のうちの1つの平面位置におけるベクトルニューロンの出力ベクトルの複数の要素値を、前記第3軸に沿った前記複数のチャンネルにわたって配列した第1種の特徴スペクトルと、(ii) 前記第1種の特徴スペクトルの各要素値に、前記正規化係数を乗じることによって得られる第2種の特徴スペクトルと、(iii) 前記特定層のうちの1つの平面位置における前記正規化係数を、前記第3軸に沿った前記複数のチャンネルにわたって配列した第3種の特徴スペクトルと、のうちのいずれかであるものとしてもよい。

この情報処理装置によれば、特徴ベクトルを容易に求めることができる。

【0065】

(6) 本開示の第2の形態によれば、ベクトルニューラルネットワーク型の機械学習モデルを用いて演算処理を実行する演算方法が提供される。前記機械学習モデルは、複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める工程は、(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求める工程と、(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求める工程と、(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求める工程と、(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求める工程と、を含む。

この演算方法によれば、カプセルネットワークのように動的ルーチンを複数回実行する必要がないので、機械学習モデルを用いた演算をより高速に実行できる。

【0066】

(7) 本開示の第3の形態によれば、ベクトルニューラルネットワーク型の機械学習モデルを用いる演算処理をプロセッサに実行させるコンピュータプログラムが提供される。前記機械学習モデルは、複数のノードをそれぞれ含む複数のベクトルニューロン層を有し、前記複数のベクトルニューロン層のうちの1つを上位層と呼び、前記上位層の下位にあるベクトルニューロン層を下位層と呼ぶとき、前記上位層の各ノードは、前記下位層の複数のノードからの出力ベクトルを入力として、1つの出力ベクトルを出力するように構成されている。前記コンピュータプログラムは、前記上位層の任意のノードである対象ノードの前記出力ベクトルを求める処理として、(a) 前記下位層の各ノードの出力ベクトルと予測行列との積に基づく予測ベクトルを求める処理と、(b) 前記下位層の前記各ノードから得られた前記予測ベクトルの線形結合に基づく和ベクトルを求める処理と、(c) 前記和ベクトルのノルムを正規化することによって正規化係数を求める処理と、(d) 前記和ベクトルを前記ノルムで除算し、更に、前記正規化係数を乗じることによって前記対象ノードの前記出力ベクトルを求める処理と、を前記プロセッサに実行させる。

このコンピュータプログラムによれば、カプセルネットワークのように動的ルーチンを複数回実行する必要がないので、機械学習モデルを用いた演算をより高速に実行できる。

【0067】

10

20

30

40

50

本開示は、上記以外の種々の形態で実現することも可能である。例えば、クラス分類装置の機能を実現するためのコンピュータプログラム、そのコンピュータプログラムを記録した一時的でない記録媒体（non-transitory storage medium）等の形態で実現することができる。

【符号の説明】

【0068】

100 ... 情報処理装置、110 ... プロセッサ、112 ... クラス分類処理部、120 ... メモリー、130 ... インターフェイス回路、150 ... 表示部、200 ... 機械学習モデル、210 ... 畳み込み層、220 ... プライマリーベクトルニューロン層、230 ... 第1畳み込みベクトルニューロン層、240 ... 第2畳み込みベクトルニューロン層、250 ... 分類ベクトルニューロン層、260 ... 類似度演算部

10

20

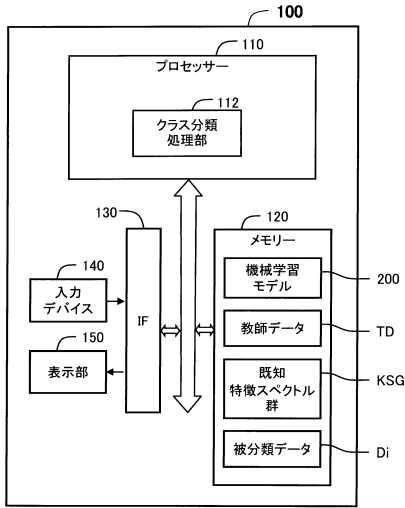
30

40

50

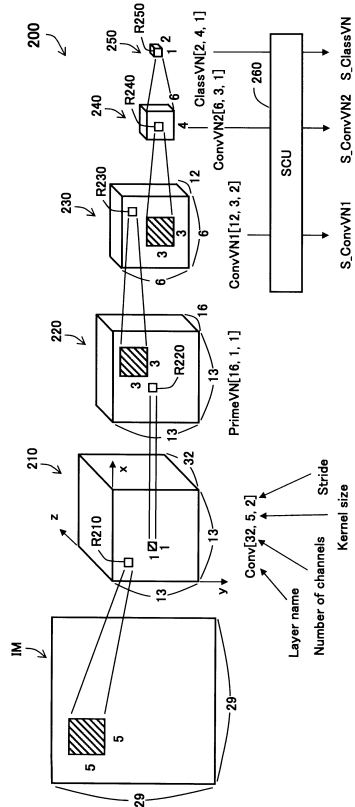
【図面】
【図 1】

Fig.1



【図 2】

Fig.2

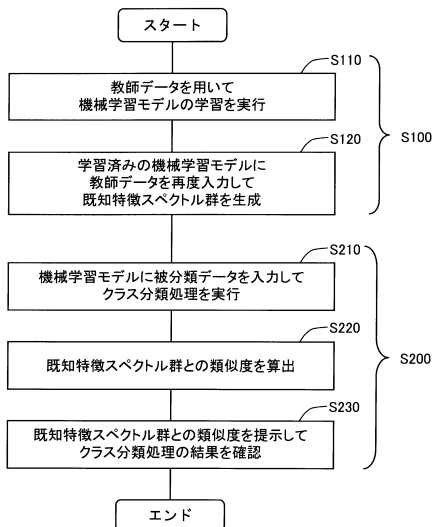


10

20

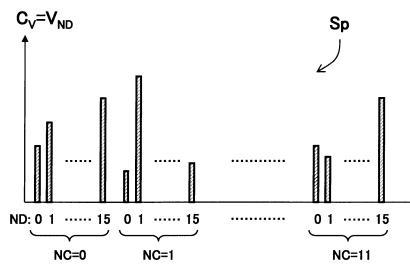
【図 3】

Fig.3



【図 4】

Fig.4



30

40

50

【 図 5 】

Fig.5

No.	レイヤー名	ラベルLb	既知特徴スペクトルKSp
1	ConvVN1	0
2	ConvVN1	0
...
N1_0max	ConvVN1	0
1	ConvVN1	1
2	ConvVN1	1
...
N1_1max	ConvVN1	1

10

20

30

40

50

フロントページの続き

- (56)参考文献 国際公開第2019/083553(WO, A1)
特開2006-164063(JP, A)
特開2020-126662(JP, A)
特表2020-522077(JP, A)
- (58)調査した分野 (Int.Cl., DB名)
G06N 3/02 - 3/10