

(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
2020年1月16日(16.01.2020)



(10) 国際公開番号

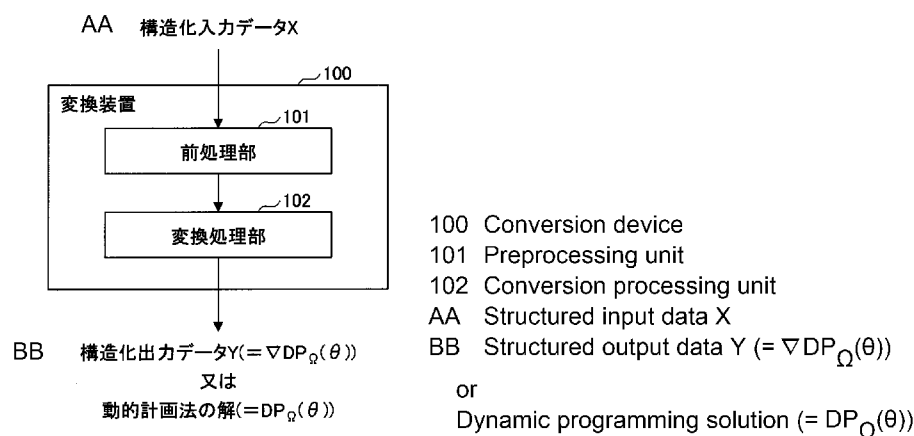
WO 2020/012975 A1

- (51) 国際特許分類:
G06N 3/08 (2006.01) *G06N 99/00* (2019.01)
- (21) 国際出願番号: PCT/JP2019/025636
- (22) 国際出願日: 2019年6月27日(27.06.2019)
- (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 2018-129998 2018年7月9日(09.07.2018) JP
- (71) 出願人: 日本電信電話株式会社 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目5番1号 Tokyo (JP).
- (72) 発明者: ブロンデル マチュー (BLONDEL, Mathieu); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). アルチュール メンシュ (ARTHUR, Mensch); 75013 パリ アルバンキャシュースクエア10 Paris (FR).
- (74) 代理人: 伊東 忠重, 外 (ITO, Tadashige et al.); 〒1000005 東京都千代田区丸の内二丁目1番1号 丸の内 M Y P L A Z A (明治安田生命ビル) 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, CZ, DE, DJ, DK, DM, DO,

(54) Title: CONVERSION DEVICE, LEARNING DEVICE, CONVERSION METHOD, LEARNING METHOD, AND PROGRAM

(54) 発明の名称: 変換装置、学習装置、変換方法、学習方法及びプログラム

[図1]



(57) Abstract: This conversion device using a neural network to convert inputted first data X into second data Y is characterized by having: a calculation means that calculates DP_Ω(θ), which is an approximation of a dynamic programming solution to a problem represented by a weighted directed acyclic graph G, by using third data θ that is acquired by executing a prescribed preprocessing of the first data X and by using a DP_Ω function recursively defined using a max_Ω function, which is a max function into which a strongly convex regularization function Ω has

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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:

- 国際調査報告 (条約第21条(3))

been introduced; and an outputting means that outputs, as the second data Y, at least one of the $DP_{\Omega}(\theta)$ that was calculated by the calculation means or a gradient $\nabla DP_{\Omega}(\theta)$ of said $DP_{\Omega}(\theta)$.

(57) 要約: 入力された第1のデータXをニューラルネットワークにより第2のデータYに変換する変換装置であって、前記第1のデータXに対して所定の前処理を行うことで得られた第3のデータ θ と、max関数に対して強凸正則化関数 Ω を導入したmax $_{\Omega}$ 関数を用いて再帰的に定義される DP_{Ω} 関数とを用いて、重み付き有向非巡回グラフGで表される問題を対象とした動的計画法の解の近似 $DP_{\Omega}(\theta)$ を計算する計算手段と、前記計算手段により計算された前記 $DP_{\Omega}(\theta)$ と、該 $DP_{\Omega}(\theta)$ の勾配 $\nabla DP_{\Omega}(\theta)$ との少なくとも一方を前記第2のデータYとして出力する出力手段と、を有することを特徴とする。

明 細 書

発明の名称：

変換装置、学習装置、変換方法、学習方法及びプログラム

技術分野

[0001] 本発明は、変換装置、学習装置、変換方法、学習方法及びプログラムに関する。

背景技術

[0002] ニューラルネットワークと呼ばれる数学モデルが従来から知られている。古典的なニューラルネットワークは、ベクトルで表現される入力データを、ベクトルやスカラー等で表現される出力データに変換する計算を行う。このようなニューラルネットワークにおける計算は、各層をそれぞれ表す関数を入れ子にした形式で記述することができる。

[0003] ところで、近年では、ニューラルネットワークが様々な分野に応用されるようになってきており、複雑な問題をニューラルネットワークで扱うことも多くなってきている。このような複雑な問題をニューラルネットワークで扱う場合には、入力データや出力データが構造化されたデータ（以降、「構造化データ」とも表す。）であることが多い。ここで、構造化データとは、単なるベクトル等のデータではなく、何等かの構造を有するデータのことであり、例えば、データに含まれる要素間に構造的な関係性があるデータや、他のデータとの間に構造的な関係性があるデータ等のことである。構造化データの具体例としては、例えば、テキスト文書を構成する単語系列や、複数の時系列データ間の対応関係を表すベクトル又は行列等が挙げられる。

[0004] このような構造化データをニューラルネットワークで扱うために、動的計画法（Dynamic Programming）の演算をニューラルネットワークの層として用いる方法が知られている。動的計画法の演算は、対象となる問題を再帰的に複数の部分問題に分割し、分割問題を順次解いていくことで解を求める手法である。ただし、動的計画法の演算は、その多様な表現力ゆえに、微分不可

能な演算となることが多い。

[0005] ここで、ニューラルネットワークのパラメータを誤差逆伝播法等により学習する場合、ニューラルネットワークの予測出力と正解データとに基づいて、所定の損失関数の微分値が計算される。このため、ニューラルネットワークの各層で行われる演算は微分可能である必要がある。

[0006] しかしながら、動的計画法の演算は微分不可能な演算となる場合が多いため、動的計画法を行う層を有するニューラルネットワークでは、パラメータの学習が困難になることがある。これに対して、条件付き確率場（CRF : Conditional Random Field）を用いて、動的計画法の演算を微分可能な演算に変換する方法が提案されている（例えば、非特許文献 1 及び 2）。

先行技術文献

非特許文献

[0007] 非特許文献1 : Lample, Guillaume, Ballesteros, Miguel, Subramanian, San deep, Kawakami, Kazuya, and Dyer, Chris. Neural architectures for named entity recognition. In Proc. of NAACL, pp. 260-270, 2016.

非特許文献2 : Cuturi, Marco and Blondel, Mathieu. Soft-DTW: a Differentiable Loss Function for Time-Series. In Proc. of ICML, pp. 894-903, 2017.

発明の概要

発明が解決しようとする課題

[0008] しかしながら、非特許文献 1 及び 2 で提案されている方法では、動的計画法の演算を行う層の出力データのスパース性が失われてしまうため、出力データの解釈性が低下する場合があった。

[0009] 動的計画法が対象とする問題では、入力された構造化データ（以降、「構造化入力データ」とも表す。）と、出力された構造化データ（以降、「構造化出力データ」とも表す。）との解釈性が重要になる場合が多い。例えば、テキスト文書を構成する単語系列を構造化入力データ、当該単語系列に含ま

れる各単語へのタグ（例えば、単語の品詞やカテゴリ等を表すタグ）付けを表す行列を構造化出力データとした場合、1つの単語に対して1つのタグが対応付けられた構造化出力データが得られることが好ましいことが多い。しかしながら、非特許文献1及び2で提案されている方法では、動的計画法の演算を行う層の出力データのスパース性が失われてしまうため、例えば、1つの単語に対して複数のタグが対応付けられた構造化出力データが得られる場合がある。このため、例えば、1つの単語に対して1つの品詞を特定するといった解釈が困難になることがある。

[0010] 本発明の実施の形態は、上記の点に鑑みてなされたもので、微分可能で、かつ、解釈性が高い動的計画法の演算を実現することを目的とする。

課題を解決するための手段

[0011] 上記目的を達成するため、本発明の実施の形態は、入力された第1のデータ X をニューラルネットワークにより第2のデータ Y に変換する変換装置であって、前記第1のデータ X に対して所定の前処理を行うことで得られた第3のデータ θ と、 $\max$ 関数に対して強凸正則化関数 Ω を導入した $\max_{\Omega}$ 関数を用いて再帰的に定義される DP_{Ω} 関数とを用いて、重み付き有向非巡回グラフ G で表される問題を対象とした動的計画法の解の近似 $DP_{\Omega}(\theta)$ を計算する計算手段と、前記計算手段により計算された前記 $DP_{\Omega}(\theta)$ と、該 $DP_{\Omega}(\theta)$ の勾配 $\nabla DP_{\Omega}(\theta)$ との少なくとも一方を前記第2のデータ Y として出力する出力手段と、を有することを特徴とする。

発明の効果

[0012] 本発明の実施の形態によれば、微分可能で、かつ、解釈性が高い動的計画法の演算を実現することができる。

図面の簡単な説明

[0013] [図1]本発明の実施の形態における変換装置の機能構成の一例を示す図である。

[図2]本発明の実施の形態における学習装置の機能構成の一例を示す図である。

[図3]ビタビアルゴリズムを実現する場合の有向非巡回グラフの一例を示す図である。

[図4]動的時間伸縮法を実現する場合の有向非巡回グラフの一例を示す図である。

[図5]本発明の効果の一例を示す図である。

[図6]本発明の実施の形態における変換装置及び学習装置のハードウェア構成の一例を示す図である。

発明を実施するための形態

[0014] 以下、本発明の実施の形態について説明する。本発明の実施の形態では、構造化入力データを構造化出力データに変換する変換装置100について説明する。このとき、本発明の実施の形態における変換装置100は、微分可能な動的計画法の演算により、構造化入力データを、解釈性が高い構造化出力データに変換する。本発明の実施の形態における変換装置100が実行する動的計画法の演算は、ニューラルネットワークの層として実現される。

[0015] また、本発明の実施の形態では、上記の動的計画法の演算が層として実現されているニューラルネットワークの学習を行う学習装置200についても説明する。

[0016] ここで、構造化入力データを構造化出力データに変換するタスクとしては、例えば、品詞タグ付け（Part-of-Speech Tagging）が挙げられる。品詞タグ付けでは、例えば、テキスト文書を構成する単語系列が構造化入力データ、当該単語系列に含まれる各単語へのタグ（例えば、単語の品詞を表すタグ）付けを表す行列が構造化出力データとなる。この場合、本発明の実施の形態における変換装置100は、テキスト解析装置として機能することになる。

[0017] 構造化入力データを構造化出力データに変換する別のタスクとしては、例えば、翻訳が挙げられる。翻訳では、例えば、原言語のテキスト文書を構成する単語系列が構造化入力データ、当該単語系列を目的言語に翻訳した単語系列が構造化出力データとなる。この場合、本発明の実施の形態における変

換装置 100 は、翻訳装置として機能することになる。

[0018] 構造化入力データを構造化出力データに変換する別のタスクとしては、例えば、複数の時系列データ間のアラインメントが挙げられる。アラインメントでは、例えば、複数の時系列データを表すデータが構造化入力データ、これら複数の時系列データ間の対応関係（例えば、複数の時系列データにそれぞれ含まれる各要素間の類似度等）を表すベクトル又は行列等が構造化出力データとなる。この場合、本発明の実施の形態における変換装置 100 は、時系列データのアラインメント装置として機能することになる。

[0019] なお、構造化入力データとしては、上述した単語系列や複数の時系列データ等に限られない。構造化入力データとしては、系列や配列等で表される任意のデータを用いることができる。例えば、画像データ、映像データ、音響信号を表すデータ、生体信号を表すデータ等も構造化入力データとして用いることができる。

[0020] <理論的背景>

以降では、本発明の実施の形態における変換装置 100 及び学習装置 200 が実行する動的計画法を用いた変換や学習等の理論的背景について説明する。本発明の実施の形態では、構造化入力データを X 、構造化出力データを Y で表す。また、構造化出力データ X の集合を

[0021] [数1]

$$X$$

、構造化出力データ Y の集合を

[0022] [数2]

$$Y$$

で表す。

[0023] 構造化入力データ X を構造化出力データ Y に変換する何等かのタスクを行う際には、例えば、以下の式 (1) や式 (2) に示す手順が行われる。

[0024] [数3]

$$X \in \mathcal{X} \xrightarrow{\text{preprocessing}} \theta \in \Theta \xrightarrow{\text{DP}} \text{value} \in \mathbf{R} \longrightarrow \dots \quad (1)$$

[0025] [数4]

$$X \in \mathcal{X} \xrightarrow{\text{preprocessing}} \theta \in \Theta \xrightarrow{\text{DP+backtracking}} Y^* \in \mathcal{Y} \longrightarrow \dots \quad (2)$$

ここで、 θ は実数を要素とする行列又はテンソルであり、 Θ は θ の集合である。また、「太字の $\mathbf{R}$ 」は実数全体を表す。以降では、明細書における表記の便宜上、実数全体を単に「 $\mathbf{R}$ 」とも表す。

[0026] 前処理 (preprocessing) は、動的計画法が対象とする問題に応じて、構造化入力データ X を θ に変換 (射影) するものであり、例えばニューラルネットワークにより実現される。具体的には、例えば、上述した品詞タグ付け (Part-of-Speech Tagging) を行う問題であれば、この前処理は双方向長短期記憶 (BLSTM: bi-directional long short-term memory) により実現される。

[0027] 上記の式 (1) は動的計画法が対象とする問題の目的関数の最適解 (value) を求めるものであり、上記の式 (2) は当該最適解を与える目的関数の引数 (Y^*) を求めるものである。上記の式 (1) では、動的計画法が対象とする問題の目的関数を解くことにより最適解 (value) が得られる。一方で、上記の式 (2) では、当該最適解 (value) を得た後に、バックトラッキングを行うことで目的関数の引数 (Y^*) が得られる。

[0028] 目的関数の最適解 (value) 又は最適解を与える目的関数の引数 (Y^*) の何れが必要となるかは、動的計画法が対象とする問題によって異なる。例えば、上述した品詞タグ付け (Part-of-Speech Tagging) を行う問題や複数の時系列データ間のアラインメントを行う問題であれば、最適解を与える目的関数の引数 (Y^*) が必要となる。なお、目的関数の最適解 (value) と最適解を与える目的関数の引数 (Y^*) との両方が必要になる場合もある。

[0029] 一般に、構造化入力データ X から構造化出力データ Y を得たい場合には上記の式 (2) の手順により、最適解を与える目的関数の引数 (Y^*) を得ることが行われる。このとき、構造化出力データ $Y = Y^*$ である。一方で、構造化入力データ X から構造化出力データ $Y = Y^*$ が得られた場合における何等かの値 (例えば、 Y^* が得られた場合における品詞タグ付けの精度等) を得たい場合には上記の式 (1) の手順により、目的関数の解 (value) を得ることが行われる。

[0030] また、一般に、動的計画法が対象とする問題の目的関数の最適解 (value) を「動的計画法の解」と呼ぶ場合が多いが、最適解を与える目的関数の引数 (Y^*) を「動的計画法の解」と呼ぶ場合もある。本発明の実施の形態では、動的計画法が対象とする問題の目的関数の最適解 (value) を「動的計画法の解」と呼ぶこととする。

[0031] ここで、上述した前処理によって θ が得られたと仮定すると、動的計画法が対象とする問題の目的関数の最適解 (value) を求める処理は、重み付き有向非巡回グラフ (DAG: Directed Acyclic Graph) 上の開始ノードから終了ノードまでの経路のうち、所定のスコアが最大となる経路を見つける問題に定式化することができる。

[0032] そこで、 $G = (\nu, \varepsilon)$ をノード集合 ν とエッジ集合 ε とで構成される重み付き有向非巡回グラフとする。また、ノード数 N は、 $N = |\nu| \geq 2$ であるものとする。エッジ集合 ε に含まれる各エッジは有向エッジであり、或るノードから別の或るノードへの有向エッジが存在する場合、当該或るノード

が「親ノード」、当該別の或るノードが「子ノード」となる。

[0033] 一般性を欠くことなく、何れのノードも子ノードよりも小さい番号となるように、各ノードに対して順番に番号（ID）を付与し、各ノードを順序付けることができる。IDが1のノードを開始ノード、IDがNのノードを終了ノードとする。これにより、

[0034] [数5]

$$\mathcal{V} = [N] \triangleq \{1, \dots, N\}$$

と表すことができる。以降では、IDがnのノードを「ノードn」と表す。なお、「=の上に△が付与された記号」は、この記号の左辺を右辺で定義することを意味する。

[0035] 重み付き有向非巡回グラフGにおいて、ノード1は親ノードを持たない唯一のノードであり、ノードNは子ノードを持たない唯一のノードである。また、重み付き有向非巡回グラフGにおいて、親ノードjから子ノードiへの有向エッジ（i, j）は、重み $\theta_{i,j} \in \mathbb{R}$ を持つものとする。

[0036] 重み付き有向非巡回グラフGの各重み $\theta_{i,j}$ を要素とする行列を $\theta \in \Theta \subseteq \mathbb{R}^{N \times N}$ とする。ただし、エッジ集合 ε に含まれない有向エッジ（i, j）に関する重み $\theta_{i,j}$ は、 $\theta_{i,j} = -\infty$ とする。

[0037] 重み付き有向非巡回グラフGにおいてノード1からノードNに至る全ての経路（パス）の集合を

[0038] [数6]

$$\mathcal{Y}'$$

とすれば、任意の経路

[0039]

[数7]

$$Y' \in \mathcal{Y}'$$

は、 $N \times N$ のバイナリ行列で表すことができる。すなわち、経路 Y' は、 (i, j) 成分の要素を y'_{ij} とした場合に、経路 Y' が有向エッジ (i, j) を通る場合に $y'_{ij} = 1$ 、通らない場合に $y'_{ij} = 0$ となる要素 y'_{ij} で表される行列である。このように表現された経路 Y' は、構造化出力データ Y と1対1に対応する。したがって、以降では、経路 Y' を構造化出力データ Y と同一視して、「経路 Y 」（経路 Y の (i, j) 成分の要素は y_{ij} ）と表す。同様に、経路 Y の集合と構造化出力データ Y の集合とは同一視される。

[0040] このとき、 Y と θ とのフロベニウス内積を $\langle Y, \theta \rangle$ と表した場合、 $\langle Y, \theta \rangle$ は経路 Y に沿って各エッジ (i, j) の重み θ_{ij} を足し合わせた総和に相当する。したがって、フロベニウス内積 $\langle Y, \theta \rangle$ をスコアとして、全ての経路 Y の中からスコアが最大となる経路 $Y = Y^*$ を求めるには、以下の組み合わせ問題 $LP(\theta)$ を計算することになる。

[0041] [数8]

$$LP(\theta) \triangleq \max_{Y \in \mathcal{Y}} \langle Y, \theta \rangle \in \mathbf{R}$$

ここで、

[0042] [数9]

$$\mathcal{Y}$$

の大きさは N に対して指数関数的に増大するが、 $LP(\theta)$ は、動的計画法を用いて、重み付き有向非巡回グラフ G 上の1つの順序付けられた経路 Y について計算することができる。そこで、重み付き有向非巡回グラフ G におけるノード i の親ノードの集合を

[0043] [数10]

$$\mathcal{P}_i$$

として、 $v_i(\theta)$ を以下の式(3)により再帰的に定義する。

[0044] [数11]

$$\begin{aligned} v_1(\theta) &\triangleq 0 \\ \forall i \in [2, \dots, N] : v_i(\theta) &\triangleq \max_{j \in \mathcal{P}_i} \theta_{i,j} + v_j(\theta) \end{aligned} \quad (3)$$

そうすると、最終的に計算される $v_N(\theta)$ が $DP(\theta)$ となる。すなわち

、

[0045] [数12]

$$DP(\theta) \triangleq v_N(\theta)$$

である。

[0046] 動的計画法により計算される解が最適であることは証明することができるため、任意の $\theta \in \Theta$ に対して、 $DP(\theta) = LP(\theta)$ が成り立つ。すなわち、再帰的に定義された上記の式(3)を計算することで、動的計画法の解(上記の式(1)におけるvalue)を求めることができる。

[0047] ここで、上記の式（２）のように、動的計画法の解（目的関数の最適解）が得られた場合に、この最適解を与える目的関数の引数（ Y^* ）を求める問題は、最大スコアを与える経路 Y に対して、

[0048] [数13]

$$Y^*(\theta) \in \operatorname{argmax}_{Y \in \mathcal{Y}} \langle Y, \theta \rangle \quad (4)$$

を求める問題であると言える。上記の式（４）に示す引数（ Y^* ）は、まず、上記の式（３）の再帰的な計算を行った後、バックトラッキングを行うことで求めることができる。

[0049] しかしながら、 $DP(\theta)$ は微分不可能であり、 $Y^*(\theta)$ は不連続関数である。このため、ニューラルネットワークの層として動的計画法の演算を実現する場合、誤差逆伝播法等による微分値（所定の損失関数の微分値）の計算を行うことができないため、例えば勾配降下法（gradient descent）等を用いたニューラルネットワークの学習を行うことができない。

[0050] そこで、本発明の実施の形態では、上記の式（１）及び式（２）に示す手順の代わりに、以下の式（１'）及び式（２'）に示す手順を用いる。

[0051] [数14]

$$X \in \mathcal{X} \xrightarrow{\text{preprocessing}} \theta \in \Theta \xrightarrow{DP_{\Omega}} \text{value} \in \mathbf{R} \longrightarrow \dots \quad (1')$$

[0052] [数15]

$$X \in \mathcal{X} \xrightarrow{\text{preprocessing}} \theta \in \Theta \xrightarrow{\nabla DP_{\Omega}} \text{gradient} \in \operatorname{conv}(\mathcal{Y}) \longrightarrow \dots \quad (2')$$

ここで、 DP_{Δ} は DP の近似であり、 DP_{Δ} 以降に続く処理（すなわち、ニューラルネットワークにおいて、動的計画法の演算を行う層以降の層が行う処理）も、 DP を用いた場合と同様に正確に定義することができる。また、 ∇DP_{Δ} は DP_{Δ} の勾配であり、

[0053] [数16]

$\text{conv}(\mathcal{Y})$ は $\mathcal{Y}$ の凸包

である。この凸包は、

[0054] [数17]

$$\text{conv}(\mathcal{Y}) \triangleq \left\{ \sum_{Y \in \mathcal{Y}} \lambda_Y Y : \lambda \in \Delta^{|\mathcal{Y}|} \right\}$$

で定義される。また、 Δ^D は D 次元の単体（simplex）であり、

[0055] [数18]

$$\Delta^D \triangleq \{ \lambda \in \mathbf{R}_+^D : \|\lambda\|_1 = 1 \}$$

で定義される。

[0056] ここで、 DP_{Δ} 及び ∇DP_{Δ} は、 DP 及び Y^* と異なり、微分可能である。また、任意の精度を γ （言い換えれば、 DP_{Δ} と DP との誤差を γ ）とすれば、 DP_{Δ} 及び DP の関係と、 ∇DP_{Δ} 及び Y^* の関係とは、それぞれ以下のように表される。

[0057]

[数19]

$$\begin{aligned} \text{DP}_{\gamma\Omega} &\xrightarrow{\gamma \rightarrow 0} \text{DP} \\ \forall \text{DP}_{\gamma\Omega} &\xrightarrow{\gamma \rightarrow 0} Y^* \end{aligned}$$

上記の式（1'）及び式（2'）により近似した手順を用いて動的計画法の問題を扱うため、max関数を、以下で定義されるmax_Ω関数で置き換えることを考える。

[0058] [数20]

$$\max_{\Omega}(x) \triangleq \max_{q \in \Delta^D} \langle q, x \rangle - \Omega(q)$$

ここで、 $\Omega : \Delta^D \rightarrow \mathbb{R}$ は強凸正則化関数（strongly-convex regularization function）である。

[0059] また、

[0060] [数21]

$$f: \mathcal{Y} \rightarrow \mathbb{R}$$

に関するmax_Ω関数として、便宜的に以下の表記を導入する。

[0061]

[数22]

$$\max_{Y \in \mathcal{Y}} f(Y) \triangleq \max_{\Omega} \left((f(Y))_{Y \in \mathcal{Y}} \right)$$

そして、上記の式（3）を $\max_{\Omega}$ 関数で置き換えることで、以下の式（5）を再帰的に定義することができる。

[0062] [数23]

$$\begin{aligned} v_1(\theta) &\triangleq 0 \\ \forall i \in [2, \dots, N] : \quad v_i(\theta) &\triangleq \max_{j \in \mathcal{P}_i} \theta_{i,j} + v_j(\theta) \end{aligned} \quad (5)$$

以降では、上記の式（5）を、便宜的に、

[0063] [数24]

$$(v_i(\theta))_{i=1}^N$$

とも表す。

[0064] 上記の式（5）により最終的に計算される $v_N(\theta)$ が $DP_{\Omega}(\theta)$ となる。
すなわち、

[0065] [数25]

$$DP_{\Omega}(\theta) \triangleq v_N(\theta)$$

である。

[0066] 以上により、動的計画法の演算を行う層は、以下の2つの層（*Value layer*及び*Gradient layer*）で表現することができる。

[0067] [数26]

Value layer : $DP_{\Omega}(\theta) \in \mathbf{R}$

Gradient layer : $\nabla DP_{\Omega}(\theta) \in \text{conv}(\mathcal{Y})$

なお、動的計画法の解を得たい場合には、ニューラルネットワークの層として、*Value layer*を用いれば良い。一方で、動的計画法の解を与える目的関数の引数を得たい場合には、ニューラルネットワークの層として、*Gradient layer*を用いれば良い。

[0068] *Value layer*である $DP_{\Omega}(\theta)$ は、 $DP(\theta)$ の微分可能な近似として利用することができる。例えば、 $DP_{\Omega}(\theta)$ は、ニューラルネットワークの学習時に、正解出力 Y_{true} と、当該ニューラルネットワークの予測出力 $\nabla DP_{\Omega}(\theta)$ とがどの程度近付いたかを表す損失関数（この損失関数を L_1 と表す。）を定義する際に用いることができる。損失関数 L_1 は、例えば、以下の式（6）により定義される。

[0069] [数27]

$$DP_{\Omega}(\theta) - \langle Y_{\text{true}}, \theta \rangle \in \mathbf{R} \quad (6)$$

この損失関数 L_1 の値が小さい程、正解出力 Y_{true} により近い予測出力 $\nabla DP_{\Omega}(\theta)$ が得られたことを表している。

[0070] ニューラルネットワークの或る層として*Value layer*（すなわち、 $DP_{\Omega}(\theta)$ の演算を行う層）を用いた場合、このニューラルネットワー

クのパラメータを学習するために $DP_{\Omega}(\theta)$ の勾配 $\nabla DP_{\Omega}(\theta)$ を計算する必要がある。この勾配 $\nabla DP_{\Omega}(\theta)$ は、上記の式 (5) を用いて誤差逆伝播法により計算することができる。より具体的には、 $E = (e_{i,j}) \in R^{N \times N}$, $Q = (q_{i,j}) \in R^{N \times N}$, $h = (h_1, \dots, h_N) \in R^N$ として、以下の $Step 1-1 \sim Step 1-3$ の手順により $E = \nabla DP_{\Omega}(\theta) \in R^{N \times N}$ を得ることができる。なお、 $\theta \in R^{N \times N}$ は所与であるものとする。

[0071] $Step 1-1$: 初期化手順として、 $v_1 \leftarrow 0 \in R$, $h_N \leftarrow 1 \in R$, $Q \leftarrow 0 \in R^{N \times N}$, $E \leftarrow 0 \in R^{N \times N}$ とする。なお、「 $\leftarrow$ 」は、左辺に右辺を代入することを意味する。

[0072] $Step 1-2$: 順方向手順として、 $i = 2, \dots, N$ に対して、順に、以下の計算及び代入をそれぞれ行う。

[0073] [数28]

$$v_i \leftarrow \max_{j \in \mathcal{P}_i} \theta_{i,j} + v_j$$

$$(q_{i,j})_{j \in \mathcal{P}_i} \leftarrow \nabla \max_{j \in \mathcal{P}_i} \theta_{i,j} + v_j$$

$Step 1-3$: 逆方向手順として、 $j = N-1, \dots, 1$ に対して、順に、以下の計算及び代入をそれぞれ行う。

[0074] [数29]

$$\forall i \in C_j, e_{i,j} \leftarrow q_{i,j} h_i, h_j \leftarrow \sum_{i \in C_j} e_{i,j}$$

ここで C_j はノード j の子ノードの集合を表す。

以上の手順により最終的に得られた E が $\nabla DP_{\Omega}(\theta)$ である。

[0075] 一方で、Gradient layerである $\nabla DP_{\Omega}(\theta)$ は、上記の式(4)により定義される $Y^*(\theta)$ の微分可能な近似として利用することができる。例えば、 $\nabla DP_{\Omega}(\theta)$ は、ニューラルネットワークの学習時に、正解出力 Y_{true} と、当該ニューラルネットワークの予測出力 $\nabla DP_{\Omega}(\theta)$ とがどの程度近付いたかを表す損失関数(この損失関数を L_2 と表す。)を定義する際に用いることができる。損失関数 L_2 は、例えば、以下の式(7)により定義される。

[0076] [数30]

$$\Delta(Y_{true}, \nabla DP_{\Omega}(\theta)) \quad (7)$$

ここで、 Δ は、例えば、ユークリッド距離やカルバック・ライブラー情報量(Kullback-Leibler divergence)等のダイバージェンスである。この損失関数 L_2 の値が小さい程、正解出力 Y_{true} により近い予測出力 $\nabla DP_{\Omega}(\theta)$ が得られたことを表している。

[0077] ニューラルネットワークの或る層としてGradient layer(すなわち、 $\nabla DP_{\Omega}(\theta)$ の演算を行う層)を用いた場合、このニューラルネットワークのパラメータを学習するために、 $\nabla DP_{\Omega}(\theta)$ のヤコビアン $\nabla \nabla DP_{\Omega}(\theta)$ (すなわち、ヘッセ行列(Hessian) $\nabla^2 DP_{\Omega}(\theta)$)と、与えられた行列 $Z \in R^{N \times N}$ との積を計算する必要がある。これは、以下の参考文献1に開示されているPearlmutter's methodにより計算することができる。

[0078] [参考文献1]

Pearlmutter, Barak A. Fast exact multiplication by the Hessian. Neural computation, 6(1):147-160, 1994.

なお、Gradient layerである $\nabla DP_{\Omega}(\theta)$ を、ニューラルネットワークの注意機構(attention mechanism)として用いることもできる

。

[0079] ここで、 $DP_{\Omega}(\theta)$ 及び $\nabla DP_{\Omega}(\theta)$ に用いられる $\max_{\Omega}$ 関数は、動的計画法が対象とする問題に応じて適宜に設定されれば良いが、以下に $\max_{\Omega}$ 関数の具体例を 2 つ示す。

[0080] ・ $\max_{\Omega}$ 関数の具体例 1

$\max_{\Omega}$ 関数の具体例 1 は、強凸正則化関数 Ω として $negative\ entropy$ を用いたものである。

[0081] $\gamma > 0$ として、

[0082] [数31]

$$\begin{aligned}\Omega(q) &= -\gamma H(q) \\ -H(q) &= \sum_{i=1}^D q_i \log q_i \quad (\text{negative entropy})\end{aligned}$$

とする。すると、 $\max_{\Omega}$ 関数、勾配 $\nabla \max_{\Omega}$ 及びヘッセ行列 $\nabla^2 \max_{\Omega}$ は、それぞれ以下で表される。

[0083] [数32]

$$\begin{aligned}\max_{\Omega}(x) &= \gamma \log \left(\sum_{i=1}^D \exp(x_i/\gamma) \right) \\ \nabla \max_{\Omega}(x) &= \exp(x/\gamma) / \sum_{i=1}^D \exp(x_i/\gamma) \\ \nabla^2 \max_{\Omega}(x) &= J_{\Omega}(\nabla \max_{\Omega}(x))\end{aligned}$$

ここで、

[0084] [数33]

$$J_{\Omega}(q) \triangleq (\text{Diag}(q) - qq^T) / \gamma$$

である。また、 $\text{diag}(q)$ は、対角成分が q の各要素で与えられる正方行列である。なお、 $\gamma = 1$ とすれば、 $\nabla \max_{\Omega}$ は softmax と一致する。

[0085] $\cdot \max_{\Omega}$ 関数の具体例 2

$\max_{\Omega}$ 関数の具体例 2 は、強凸正則化関数 Ω として squared 2-norm を用いたものである。

[0086] $\gamma > 0$ として、

[0087] [数34]

$$\Omega(x) = \frac{\gamma}{2} \|x\|_2^2 \text{ (squared 2-norm)}$$

とする、すると、 $\max_{\Omega}$ 関数、勾配 $\nabla \max_{\Omega}$ 及びヘッセ行列 $\nabla^2 \max_{\Omega}$ は、それぞれ以下で表される。

[0088] [数35]

$$\max_{\Omega}(x) = \langle q^*, x \rangle - \frac{\gamma}{2} \|q^*\|_2^2$$

$$\nabla \max_{\Omega}(x) = \underset{q \in \Delta^D}{\text{argmin}} \|q - x/\gamma\|_2^2 = q^*$$

$$\nabla^2 \max_{\Omega}(x) = J_{\Omega}(\nabla \max_{\Omega}(x))$$

ここで、

[0089] [数36]

$$J_{\Omega}(q) \triangleq (\text{Diag}(s) - ss^T / \|s\|_1) / \gamma$$

である。また、 $s \in \{0, 1\}^D$ は、ベクトル q をサポートするベクトルである。なお、 $\nabla \max_{\Omega}$ は、単体上へのユークリッド射影である。

[0090] 具体例2における $\nabla \max_{\Omega}$ は、以下の参考文献2に記載されている「sparsemax」と一致する。このため、具体例2における $\max_{\Omega}$ 関数を用いた場合、スパース性の高い構造化出力データ Y を得られることが期待できる。

[0091] [参考文献2]

Martins, Andre F.T. and Astudillo, Ramoon Fernandez. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Proc. of ICML, pp. 1614-1623, 2016.

<機能構成>

以降では、本発明の実施の形態における変換装置100及び学習装置200の機能構成について説明する。

[0092] (変換装置100)

まず、本発明の実施の形態における変換装置100の機能構成について、図1を参照しながら説明する。図1は、本発明の実施の形態における変換装置100の機能構成の一例を示す図である。

[0093] 図1に示すように、本発明の実施の形態における変換装置100は、前処理部101と、変換処理部102とを有する。これら各機能部は、例えば、変換装置100にインストールされた1以上のプログラムが、CPU (Central Processing Unit) 等の演算装置に実行させる処理により実現される。

[0094] 前処理部101及び変換処理部102は、構造化入力データ X を構造化出

力データ $Y (= \nabla D P_{\theta}(\theta))$ に変換する。又は、前処理部 101 及び変換処理部 102 は、構造化入力データ X を、動的計画法の解 $(= D P_{\theta}(\theta))$ に変換する。なお、上述したように、 $D P_{\theta}(\theta)$ は、正確には動的計画法の解 $D P(\theta)$ の近似である。

[0095] 前処理部 101 及び変換処理部 102 は、1 以上のニューラルネットワークにより実現される。例えば、上述したように、前処理部 101 は、双方向長短期記憶 (B L S T M) 等のニューラルネットワークで実現され、変換処理部 102 は、動的計画法の演算を行う層を有するニューラルネットワークにより実現される。

[0096] なお、前処理部 101 及び変換処理部 102 は、前処理部 101 を実現するニューラルネットワークと、変換処理部 102 を実現するニューラルネットワークとを組み合わせたニューラルネットワークにより実現されていても良い。この場合、前処理部 101 及び変換処理部 102 を実現するニューラルネットワークは、構造化入力データ X を θ に変換する層（前処理部 101 の演算を行う層）と、 θ を構造化出力データ $Y (= \nabla D P_{\theta}(\theta))$ 又は動的計画法の解 $(= D P_{\theta}(\theta))$ に変換する層（変換処理部 102 の演算を行う層）とを有することになる。

[0097] 前処理部 101 は、学習済のニューラルネットワークにより、上記の式 (1') 又は式 (2') における前処理 (preprocessing) を行う。すなわち、前処理部 101 は、構造化入力データ X を θ に変換する。この前処理は、動的計画法が対象とする問題に応じて決定される所定の前処理である。例えば、上述したように、動的計画法が対象とする問題が品詞タグ付け (Part-of-Speech Tagging) である場合、この前処理は、双方向長短期記憶 (B L S T M) により実現される。

[0098] なお、変換装置 100 が前処理部 101 を有する代わりに、変換装置 100 とは異なる別の装置が前処理部 101 を有しても良い。この場合、当該別の装置の前処理部 101 で構造化入力データ X を θ に変換した後、この θ を変換装置 100 に入力するようにしても良い。

- [0099] 変換処理部102は、学習済のニューラルネットワークにより、上記の式(1')又は式(2')における DP_{Ω} 又は ∇DP_{Ω} に相当する演算を行う。すなわち、変換処理部102は、前処理部101による前処理で得られた θ を、構造化出力データ $Y (= \nabla DP_{\Omega}(\theta))$ 又は動的計画法の解 $(= DP_{\Omega}(\theta))$ に変換する。この変換結果 $(DP_{\Omega}(\theta)$ 又は $\nabla DP_{\Omega}(\theta))$ は、所定の出力先に出力される。所定の出力先としては、例えば、ディスプレイ等の表示装置、補助記憶装置等の記憶装置、他のプログラム、他の装置、ニューラルネットワークにおける次の層等が挙げられる。
- [0100] DP_{Ω} に相当する演算を行う場合、変換処理部102は、上記の式(5)で再帰的に定義された演算を行えば良い。これにより、 $DP_{\Omega}(\theta) = v_N(\theta)$ が得られる。
- [0101] 一方で、 ∇DP_{Ω} に相当する演算を行う場合、変換処理部102は、上記のStep1-1~Step1-3の手順で示される演算を行えば良い。これにより、 $\nabla DP_{\Omega}(\theta)$ が得られる。
- [0102] なお、上述したように、変換処理部102による変換結果として、 $DP_{\Omega}(\theta)$ 又は $\nabla DP_{\Omega}(\theta)$ の何れを求めたいかは、動的計画法の対象とする問題に応じて決定される。なお、変換処理部102による変換結果として、 $DP_{\Omega}(\theta)$ と $\nabla DP_{\Omega}(\theta)$ との両方を求めても良い。
- [0103] (学習装置200)
- 次に、本発明の実施の形態における学習装置200の機能構成について、図2を参照しながら説明する。図2は、本発明の実施の形態における学習装置200の機能構成の一例を示す図である。
- [0104] 図2に示すように、本発明の実施の形態における学習装置200は、学習用データ入力部201と、前処理部101と、変換処理部102と、パラメータ更新部202とを有する。これら各機能部は、学習装置200にインストールされた1以上のプログラムが、CPU等の演算装置に実行させる処理により実現される。
- [0105] なお、学習装置200の前処理部101及び変換処理部102は、上述し

た変換装置 100 の前処理部 101 及び変換処理部 102 と同様である。ただし、学習装置 200 の前処理部 101 及び変換処理部 102 を実現するニューラルネットワークのパラメータには、例えば、予め決められた初期値が設定されている。これらのパラメータが学習によって更新される。

[0106] 学習用データ入力部 201 は、学習用データセットを入力する。学習用データセットとは、学習に用いられる構造化入力データ X_{train} と、この構造化入力データ X_{train} に対応する正解出力 Y_{true} との組で構成される学習用データの集合である。

[0107] 前処理部 101 及び変換処理部 102 は、学習用データ入力部 201 が入力した学習用データに含まれる各構造化入力データ X_{train} に対して前処理（preprocessing）及び変換処理（ DP_{Ω} に相当する演算又は ∇DP_{Ω} に相当する演算）をそれぞれ行って、変換結果として、 $DP_{\Omega}(\theta)$ 又は $\nabla DP_{\Omega}(\theta)$ を計算する。

[0108] パラメータ更新部 202 は、変換処理部 102 による変換結果 $DP_{\Omega}(\theta)$ 又は $\nabla DP_{\Omega}(\theta)$ と、前処理及び変換処理を行った構造化入力データ X_{train} に対応する正解出力 Y_{true} とに基づいて、所定の損失関数の微分値を計算し、この計算結果を用いてニューラルネットワークのパラメータを更新する。損失関数の微分値は、例えば、誤差逆伝播法等を用いて計算される。また、損失関数は、変換処理部 102 による変換結果が $DP_{\Omega}(\theta)$ である場合には上記の式（6）が、変換処理部 102 による変換結果が $\nabla DP_{\Omega}(\theta)$ である場合には上記の式（7）が用いられる。

[0109] このとき、パラメータ更新部 202 は、所定の条件を満たすまで、ニューラルネットワークのパラメータを繰り返し更新する。この所定の条件は、ニューラルネットワークの学習が収束したか否かを判定するものであり、例えば、損失関数の値が所定の閾値以下となったか否か、所定の繰り返し回数に達したか否か等である。

[0110] パラメータ更新部 202 は、所定の条件を満たした場合、例えば、ニューラルネットワークのパラメータの値を出力して処理を終了する。

[0111] <変換装置100の動作の具体例1>

以降では、変換装置100の動作の具体例1として、変換処理部102がビタビアルゴリズム (Viterbi Algorithm) に相当する演算を行う場合について説明する。ビタビアルゴリズムは、動的計画法に用いられるアルゴリズムの中で最も有名な例の1つであり、各時刻において或る状態から別の状態に所定の確率で遷移するような状態遷移モデルの下で、入力系列に対する状態系列のうち、尤もらしい状態系列を出力系列として求めるアルゴリズムである。各状態をノード、状態から状態への遷移を有向エッジ、状態から状態に遷移する確率を重みとすれば、上記の状態遷移モデルは、重み付き有向非巡回グラフ (DAG) で表すことができる。この場合、状態系列は、重み付き有向非巡回グラフ上の開始ノードから終了ノードまでの経路として表すことができる。

[0112] したがって、構造化入力データ X を入力系列 $X = (x_1, x_2, \dots, x_T)$ とした場合、ビタビアルゴリズムでは、例えば、この入力系列 X に対する状態系列 $y = (y_1, y_2, \dots, y_T)$ のうち、尤もらしい状態系列 y (すなわち、有向非巡回グラフ上の尤もらしい経路 y) を解 (出力系列) として求めることになる。ここで、各 x_t ($t = 1, 2, \dots, T$) は D 次元の実ベクトル、各 y_t ($t = 1, 2, \dots, T$) は $[S]$ の元であるとする。なお、 $[S]$ は、集合 $\{1, \dots, S\}$ を表す。

[0113] 具体的には、例えば、入力系列 X として各 x_t の各々が単語である単語系列 X を、出力系列 y として各 x_t に対応するタグ y_t の系列を考えることができる。この場合、ビタビアルゴリズムは、入力系列 X に対して品詞タグ付け (Part-of-Speech Tagging) を行う処理と考えることができる。

[0114] ここで、時刻 t においてノード j からノード i へ遷移する場合を $y_{t,i,j} = 1$ 、それ以外の場合を $y_{t,i,j} = 0$ とすれば、状態系列 y は、 (t, i, j) 成分の要素が $y_{t,i,j}$ である $T \times S \times S$ のバイナリテンソル Y で表すことができる。

[0115] また、時刻 t においてノード j からノード i に遷移する確率を $\theta_{t,i,j}$ とし

て、 (t, i, j) 成分の要素が $\theta_{t, i, j}$ である $T \times S \times S$ の実テンソルを θ とする。この θ は、例えば BLSTM 等を用いて前処理部 101 により求められる。すなわち、この場合、変換装置 100 の前処理部 101 は、例えば BLSTM により $T \times S \times S$ の実テンソル θ を求めることになる。

[0116] すると、フロベニウス内積 $\langle Y, \theta \rangle$ は、状態系列 y が表す経路に沿って各エッジの重み $\theta_{t, i, j}$ を足し合わせた総和に相当する。この様子を図 3 に示す。図 3 に示す例では、入力系列を $X = (x_1, x_2, x_3) = (\text{the}, \text{boat}, \text{sank})$ 、状態系列を $y = (y_1, y_2, y_3)$ 、 $y_t \in \{\text{NOUN}, \text{VERB}, \text{DET}\}$ とした場合に、この入力系列 X に対する状態系列 y のうち、時刻 $t = 1$ でノード 1 からノード 3、時刻 $t = 2$ でノード 3 からノード 1、時刻 $t = 3$ でノード 3 からノード 2 に遷移したときの状態系列 y を表している。このとき、図 3 に示すように、フロベニウス内積 $\langle Y, \theta \rangle$ は、 $\langle Y, \theta \rangle = \theta_{1, 3, 1} + \theta_{2, 1, 3} + \theta_{3, 2, 1}$ と表される。

[0117] ここで、このフロベニウス内積 $\langle Y, \theta \rangle = \theta_{1, 3, 1} + \theta_{2, 1, 3} + \theta_{3, 2, 1}$ が最大スコアであれば、図 3 に示す経路 y が最尤経路（すなわち、ビタビアルゴリズムの解となる出力系列）であり、このときの経路 y は、単語 $x_1 = \text{「the」}$ に対して品詞 $y_1 = \text{「DET」}$ （限定詞）、単語 $x_2 = \text{「boat」}$ に対して品詞 $y_2 = \text{「NOUN」}$ （名詞）、単語 $x_3 = \text{「sank」}$ に対して品詞 $y_3 = \text{「VERB」}$ （動詞）が対応付けられることを表している。

[0118] なお、 $\Omega = -H(\text{negative entropy})$ とすれば、以下の参考文献 3 に開示されている `linear-chain CRFs` (conditional random fields) を復元することができる。

[0119] [参考文献 3]

Lafferty, John, McCallum, Andrew, and Pereira, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of ICML, pp. 282-289, 2001.

ビタビアルゴリズムの解を求めるためには、変換装置 100 の変換処理部 102 は、前処理部 101 で求めた $T \times S \times S$ の実テンソル θ に基づき、以

下に定義される $Vit_{\Omega}(\theta)$ と、この $Vit_{\Omega}(\theta)$ から計算される $\nabla Vit_{\Omega}(\theta)$ とを計算すれば良い。

[0120] [数37]

$$Vit_{\Omega}(\theta) \triangleq \max_{\Omega}(v_T(\theta))$$

ここで、 $v_t(\theta)$ ($t = 1, \dots, T$) は、 $v_t(\theta) = (v_{t,1}(\theta), \dots, v_{t,s}(\theta))$ である。また、 $v_t(\theta)$ の i 番目の要素 $v_{t,i}(\theta)$ は、以下で定義される。

[0121] [数38]

$$v_{t,i}(\theta) \triangleq \max_{j \in [S]} v_{t-1,j}(\theta) + \theta_{t,i,j} = \max_{\Omega}(v_{t-1}(\theta) + \theta_{t,i})$$

なお、 $Vit_{\Omega}(\theta)$ は、任意の Ω に対して凸関数となる。

[0122] ここで、 $Vit_{\Omega}(\theta)$ は、以下の Step 2-1 ~ Step 2-3 の手順（順方向の手順）により計算することができる。また、 $\nabla Vit_{\Omega}(\theta)$ は、以下の Step 2-1 ~ Step 2-3 の手順が行われた後に、以下の Step 3-1 ~ Step 3-2 の手順（逆方向の手順）により計算することができる。なお、以下の Step 2-1 ~ Step 2-3 及び Step 3-1 ~ Step 3-2 では、 Q を $(T+1) \times T \times S$ のテンソル、 U を $(T+1) \times S$ の行列、すなわち、

[0123] [数39]

$$Q \triangleq (q)_{t=1,i,j=1}^{T+1,S,S}, \quad U \triangleq (u)_{t=1,j=1}^{T+1,S}$$

とする。また、 $\theta \in \mathbb{R}^{T \times N \times N}$ は所与であるものとする。

[0124] Step 2-1: $v_0 = 0 \in \mathbb{R}^S$ とする。

[0125] Step 2-2: $t = 1, \dots, T$ に対して、順に、各 $i \in [S]$ に対する以下の計算を行う。

[0126] [数40]

$$v_{t,i} = \max_{\Omega}(\theta_{t,i} + v_{t-1})$$

$$q_{t,i} = \nabla \max_{\Omega}(\theta_{t,i} + v_{t-1})$$

Step 2-3: 上記で計算された $v_T = (v_{T,1}, \dots, v_{T,S})$ を用いて、 $\max_{\Omega}(v_T)$ を計算する。この $\max_{\Omega}(v_T)$ が $Vit_{\Omega}(\theta)$ である。また、後述する Step 3-1 ~ Step 3-3 のために、

[0127] [数41]

$$v_{T+1,1} = \max_{\Omega}(v_T)$$

$$q_{T+1,1} = \nabla \max_{\Omega}(v_T)$$

とする。

[0128] Step 3-1: $u_{T+1} = (1, 0, \dots, 0) \in \mathbb{R}^S$ とする。

[0129] Step 3-2: $t = T, \dots, 0$ に対して、順に、各 $j \in [S]$ に対する以下の計算を行う。

[0130]

[数42]

$$\begin{aligned} e_{t,\cdot,j} &= q_{t+1,\cdot,j} \circ u_{t+1} \\ u_{t,j} &= \langle e_{t,\cdot,j}, 1_S \rangle \end{aligned}$$

ここで、 $\circ$ は要素積（アダマール積）を表す。

[0131] 以上の手順により

[0132] [数43]

$$\nabla \text{Vit}_{\Omega}(\theta) = \left(e_{t-1,i,j} \right)_{t=1,i,j=1}^{T,S,S}$$

が得られる。

[0133] また、上記のStep 3-1～Step 3-3の手順を行った後、以下のStep 4-1～Step 4-5の手順により、与えられた $Z \in \mathbb{R}^{T \times S \times S}$ に対して、 $\langle \nabla \text{Vit}_{\Omega}(\theta), Z \rangle$ 及び $\nabla^2 \text{Vit}_{\Omega}(\theta) Z$ を計算することができる。なお、Step 4-1～Step 4-3が順方向の手順、Step 4-4～Step 4-5が逆方向の手順である。

[0134] Step 4-1：まず、

[0135] [数44]

$$\dot{v}_0 = 0_S$$

とする。

[0136] Step 4-2： $t = 1, \dots, T$ に対して、順に、各 $i \in [S]$ に対する以下の計算を行う。

[0137] [数45]

$$\begin{aligned}\dot{v}_{t,i} &= \langle q_{t,i}, z_{t,i} + \dot{v}_{t-1} \rangle \\ \dot{q}_{t,i} &= J_{\Omega}(q_{t,i})(z_t + \dot{v}_{t-1})\end{aligned}$$

Step 4-3 : そして、以下を計算する。

[0138] [数46]

$$\begin{aligned}\dot{v}_{T+1,1} &= \langle q_{T+1,1}, \dot{v}_T \rangle \\ \dot{q}_{T+1,1} &= J_{\Omega}(\dot{q}_{T+1,1})\dot{v}_T\end{aligned}$$

Step 4-4 : 次に、

[0139] [数47]

$$\begin{aligned}\dot{u}_{T+1} &= 0_S \\ \dot{Q}_{T+1} &= 0_{S \times S}\end{aligned}$$

とする。

[0140] Step 4-5 : $t = T, \dots, 0$ に対して、順に、各 $i \in [S]$ に対する以下の計算を行う。

[0141] [数48]

$$\begin{aligned}\dot{e}_{t,\cdot,j} &= q_{t+1,\cdot,j} \circ u_{t+1} + \dot{q}_{t+1,\cdot,j} \circ \dot{u}_{t+1} \\ \dot{u}_{t,j} &= \langle \dot{e}_{t,\cdot,j}, 1_S \rangle\end{aligned}$$

以上の手順により

[0142] [数49]

$$\langle \text{Vit}_{\Omega}(\theta), Z \rangle = \dot{v}_{T+1}$$

$$\nabla^2 \text{Vit}_{\Omega}(\theta) Z = \left(\dot{e}_{t-1, i, j} \right)_{t=1, i, j=1}^{T, S, S}$$

が得られる。

[0143] <変換装置100の動作の具体例2>

以降では、変換装置100の動作の具体例2として、変換処理部102が動的時間伸縮法(DTW: Dynamic Time Wrapping)に相当する演算を行う場合について説明する。動的時間伸縮法は、2つの時系列データの相関(類似度)を分析する場合等に用いられる。

[0144] N_A を時系列データAの系列長、 N_B を時系列データBの系列長とする。また、時系列データAの*i*番目の観測値を a_i 、時系列データBの*j*番目の観測値を b_j とする。

[0145] そして、 a_i と b_j とが類似する場合は $y_{ij} = 1$ 、そうでない場合は $y_{ij} = 0$ として、 y_{ij} を要素とする $N_A \times N_B$ のバイナリ行列Yを考えた場合、このバイナリ行列Yは、時系列データAと時系列データBとの対応関係(類似関係)を表すアラインメントYである。

[0146] また、 θ を $N_A \times N_B$ の行列として、 θ の各要素を θ_{ij} とする。古典的な例としては、或る微分可能な距離尺度*d*を用いて、 $\theta_{ij} = d(a_i, b_j)$ である。この θ は、変換装置100の前処理部101により求められる。なお、 θ は、距離行列とも呼ばれる。

[0147] すると、観測値の組(a_i, b_j)をノードとして、アラインメントYは、重み付き有向非巡回グラフ(DAG)上の経路(パス)を表す。

[0148] ここで、全ての単調アラインメント行列の集合を

[0149] [数50]

$$y$$

とする。単調アラインメント行列とは、行列の左上（1， 1）成分から右下（ N_A ， N_B ）に至る後戻りしないパス、すなわち（ i ， j ）ノードからは右、下、右下の何れかの成分へ向かうパスのみが許容されるような行列である。つまり、 $y_{i,j} = 1$ である場合、 $y_{i+1,j}$ 、 $y_{i,j+1}$ 、 $y_{i+1,j+1}$ の少なくともいずれかは1となる。 $N_A = 4$ 及び $N_B = 3$ とした場合における或る単調アラインメント行列 Y が表す経路（パス）を図4に示す。図4に示す経路は、 a_1 及び b_1 、 a_2 及び b_2 、 a_3 及び b_2 、 a_4 及び b_3 がそれぞれ類似している場合を示している。すなわち、この場合の単調アラインメント行列 Y は、 y_{11} 、 y_{22} 、 y_{32} 及び y_{43} がそれぞれ1で、他の要素が0である行列で表される。

[0150] 上記の単調アラインメント行列 Y を用いることで、フロベニウス内積 $\langle Y, \theta \rangle$ は、単調アラインメント行列 Y が表す経路に沿って各エッジの重み $\theta_{i,j}$ を足し合わせた総和に相当する。すなわち、このフロベニウス内積 $\langle Y, \theta \rangle$ をアラインメントのコストに用いることができる。図4に示す経路では、フロベニウス内積 $\langle Y, \theta \rangle$ は、 $\langle Y, \theta \rangle = \theta_{1,1} + \theta_{2,2} + \theta_{2,3} + \theta_{3,3} + \theta_{3,4}$ と表される。

[0151] ここで、 $v_{i,j}(\theta)$ をアラインメントの（ i ， j ）成分（セル）のコストであるとすれば、 $v_{i,j}(\theta)$ は、

[0152] [数51]

$$v_{i,j}(\theta) = \theta_{i,j} + \min_{\Omega} (v_{i,j-1}(\theta), v_{i-1,j-1}(\theta), v_{i-1,j}(\theta))$$

と表すことができる。

[0153] また、 $\min_{\Omega}$ 関数、勾配 $\nabla \min_{\Omega}$ 及びヘッセ行列 $\nabla^2 \min_{\Omega}$ をそれぞれ以下で定義及び導入する。

[0154] [数52]

$$\begin{aligned}\min_{\Omega}(x) &\triangleq -\max_{\Omega}(-x) \\ \nabla \min_{\Omega}(x) &= \nabla \max_{\Omega}(-x) \\ \nabla^2 \min_{\Omega}(x) &= -\nabla^2 \max_{\Omega}(-x) \\ &= -J_{\Omega}(\nabla \max_{\Omega}(-x)) \\ &= -J_{\Omega}(\nabla \min_{\Omega}(x))\end{aligned}$$

すると、尤もらしいアラインメント Y を求めるためには、変換装置100の変換処理部102は、前処理部101で求めた $N_A \times N_B$ の距離行列 θ に基づき、以下に定義された $DTW_{\Omega}(\theta)$ と、この $DTW_{\Omega}(\theta)$ から計算される $\nabla DTW_{\Omega}(\theta)$ とを計算すれば良い。

[0155] [数53]

$$DTW_{\Omega}(\theta) \triangleq v_{N_A, N_B}(\theta)$$

ここで、 $DTW_{\Omega}(\theta)$ は、以下のStep 5-1～Step 5-2の手順（順方向の手順）により計算することができる。また、 $\nabla DTW_{\Omega}(\theta)$ は、以下のStep 6-1～Step 6-2の手順（逆方向の手順）により計算することができる。なお、以下のStep 5-1～Step 5-2及びStep 6-1～Step 6-2では、 Q を $(N_A + 1) \times (N_B + 1) \times 3$ のテンソル、 E を $(N_A + 1) \times (N_B + 1)$ の行列、すなわち、

[0156]

[数54]

$$Q \triangleq (q)_{i,j,k=1}^{N_A+1, N_B+1, 3}$$

$$E \triangleq (e)_{i,j=1}^{N_A+1, N_B+1}$$

とする。また、

[0157] [数55]

$$\theta \in \mathbf{R}^{N_A \times N_B}$$

は所与であるものとする。

[0158] Step 5-1: $v_{0,0} = 0$ とする。また、 $i = 1, \dots, N_A$ 、 $j = 1, \dots, N_B$ に対して、 $v_{i,0} = v_{0,j} = \infty$ とする。

[0159] Step 5-2: $i = 1, \dots, N_A$ 、 $j = 1, \dots, N_B$ に対して、順に、以下の計算を行う。

[0160] [数56]

$$v_{i,j} = d_{i,j} + \min_{\Omega}(v_{i,j-1}, v_{i-1,j-1}, v_{i-1,j})$$

$$q_{i,j} = \nabla \min_{\Omega}(v_{i,j-1}, v_{i-1,j-1}, v_{i-1,j}) \in \mathbf{R}^3$$

Step 6-1: 次に、 $i = 1, \dots, N_A$ 、 $j = 1, \dots, N_B$ に対して、

[0161]

[数57]

$$q_{i, N_B+1} = q_{N_A+1, j} = 0_3$$

$$e_{i, N_B+1} = e_{N_A+1, j} = 0$$

$$q_{N_A+1, N_B+1} = (0, 1, 0)$$

$$e_{N_A+1, N_B+1} = 1$$

とする。

[0162] Step 6-2 : $j = N_B, \dots, 1, i = N_A, \dots, 1$ 対して、順に、以下の計算を行う。

[0163] [数58]

$$e_{i, j} = q_{i, j+1, 1} e_{i, j+1} + q_{i+1, j+1, 2} e_{i+1, j+1} + q_{i+1, j, 3} e_{i+1, j}$$

以上の手順により

[0164] [数59]

$$DTW_{\Omega}(\theta) = v_{N_A, N_B}$$

$$\nabla DTW_{\Omega}(\theta) = (e)_{i, j=1}^{N_A, N_B}$$

が得られる。

[0165] また、上記の Step 5-1 ~ Step 5-2 の手順を行った後、以下の Step 7-1 ~ Step 7-4 の手順により、与えられた

[0166]

[数60]

$$Z \in \mathbf{R}^{N_A \times N_B}$$

に対して、 $\langle \nabla D T W_Q(\theta), Z \rangle$ 及び $\nabla^2 D T W_Q(\theta) Z$ を計算することができる。なお、Step 7-1 ~ Step 7-2 が順方向の手順、Step 7-3 ~ Step 7-4 が逆方向の手順である。

[0167] Step 7-1 : まず、 $i = 0, \dots, N_A, j = 1, \dots, N_B$ に対して、

[0168] [数61]

$$\dot{v}_{i,0} = \dot{v}_{0,j} = 0$$

とする。

[0169] Step 7-2 : $i = 1, \dots, N_B, j = 1, \dots, N_A$ に対して、順に、以下の計算を行う。

[0170] [数62]

$$\dot{v}_{i,j} = z_{i,j} + q_{i,j,1} \dot{v}_{i,j-1} + q_{i,j,2} \dot{v}_{i-1,j-1} + q_{i,j,3} \dot{v}_{i-1,j}$$

$$\dot{q}_{i,j} = -J_{\Omega}(q_{i,j})(\dot{v}_{i,j-1}, \dot{v}_{i-1,j-1}, \dot{v}_{i-1,j}) \in \mathbf{R}^3$$

Step 7-3 : 次に、 $i = 0, \dots, N_A, j = 1, \dots, N_B$ に対して、

[0171]

[数63]

$$\begin{aligned}\dot{q}_{i, N_B+1} &= \dot{q}_{N_A+1, j} = 0_3 \\ \dot{e}_{i, N_B+1} &= \dot{e}_{N_A+1, j} = 0\end{aligned}$$

とする。

[0172] Step 7-4: $j = N_B, \dots, 1, i = N_A, \dots, 1$ に対して、順に、以下の計算を行う。

[0173] [数64]

$$\begin{aligned}\dot{e}_{i, j} &= \dot{q}_{i, j+1, 1} e_{i, j+1} + q_{i, j+1, 1} \dot{e}_{i, j+1} + \dot{q}_{i+1, j+1, 2} e_{i+1, j+1} \\ &\quad + q_{i+1, j+1, 2} \dot{e}_{i+1, j+1} + \dot{q}_{i+1, j, 3} e_{i+1, j} + q_{i+1, j, 3} \dot{e}_{i+1, j}\end{aligned}$$

以上の手順により

[0174] [数65]

$$\begin{aligned}\langle \nabla \text{DTW}_\Omega(\theta), Z \rangle &= \dot{v}_{N_A, N_B} \\ \nabla^2 \text{DTW}_\Omega(\theta) Z &= (\dot{e})_{i, j=1}^{N_A, N_B}\end{aligned}$$

が得られる。

[0175] <本発明の効果>

ここで、本発明の一例として上記の具体例2を用いた場合における効果について、図5を参照しながら説明する。図5は、本発明の効果の一例を示す図である。

[0176] 図5(a)は、 Ω としてnegative entropyを用いた場合

における $DTW_{\Omega}(\theta) = -7.49$ と、 $\nabla DTW_{\Omega}(\theta)$ のヒートマップとを示している。一方で、図5(b)は、 Ω として `squared 2-norm` を用いた場合における $DTW_{\Omega}(\theta) = 9.61$ と、 $\nabla DTW_{\Omega}(\theta)$ のヒートマップとを示している。ヒートマップは、セルの色が濃い程その値が大きいことを表し、セルが存在しない場所は値が0であることを表している。また、図5(a)及び図5(b)におけるラインLは、 $\max_{\Omega}$ 関数ではなく、 $\max$ 関数を用いた場合における $DTW(\theta)$ に対応するアラインメントを表している。

[0177] 図5(a)及び図5(b)に示すように、いずれの $\nabla DTW_{\Omega}(\theta)$ も、 $DTW(\theta)$ に対応するアラインメントを高い精度で近似できており、高い解釈性が得られていることがわかる。また、図5(b)では、図5(a)と比べて、より高いスパース性も得られている。したがって、本発明の実施の形態では、微分可能で、かつ、解釈性が高い動的計画法の演算が実現できていることがわかる。

[0178] なお、図5(a)及び図5(b)でヒートマップとして示されている $\nabla DTW_{\Omega}(\theta)$ は、上述したように、誤差逆伝播法によって得られる。これにより、距離行列 θ の学習が可能となる。

[0179] <ハードウェア構成>

最後に、本発明の実施の形態における変換装置100及び学習装置200のハードウェア構成について、図6を参照しながら説明する。図6は、本発明の実施の形態における変換装置100及び学習装置200のハードウェア構成の一例を示す図である。なお、変換装置100及び学習装置200は同様のハードウェア構成で実現可能であるため、以降では、主に、変換装置100のハードウェア構成について説明する。

[0180] 図6に示すように、本発明の実施の形態における変換装置100は、入力装置301と、表示装置302と、外部I/F303と、RAM(Random Access Memory)304と、ROM(Read Only Memory)305と、演算装置306と、通信I/F307と、補助記憶装置308とを有する。これら各ハ

ードウェアは、それぞれがバスBを介して通信可能に接続されている。

[0181] 入力装置301は、例えば、キーボードやマウス、タッチパネル等であり、ユーザが各種操作を入力するのに用いられる。表示装置302は、例えば、ディスプレイ等であり、変換装置100の処理結果を表示する。なお、変換装置100及び学習装置200は、入力装置301及び表示装置302の少なくとも一方を有していなくても良い。

[0182] 外部I/F303は、外部装置とのインタフェースである。外部装置には、記録媒体303a等がある。変換装置100は、外部I/F303を介して、記録媒体303a等の読み取りや書き込みを行うことができる。記録媒体303aには、変換装置100が有する各機能部や学習装置200が有する各機能部を実現する1以上のプログラム等が記録されていても良い。

[0183] 記録媒体303aとして、例えば、フレキシブルディスク、CD (Compact Disc)、DVD (Digital Versatile Disk)、SDメモリカード (Secure Digital memory card)、USB (Universal Serial Bus) メモリカード等が挙げられる。

[0184] RAM304は、プログラムやデータを一時保持する揮発性の半導体メモリである。ROM305は、電源を切ってもプログラムやデータを保持することができる不揮発性の半導体メモリである。ROM305には、例えば、OS (Operating System) に関する設定や通信ネットワークに関する設定等が格納されている。

[0185] 演算装置306は、例えば、CPUやGPU (Graphics Processing Unit) 等であり、ROM305や補助記憶装置308等からプログラムやデータをRAM304上に読み出して処理を実行する演算装置である。

[0186] 通信I/F307は、変換装置100を通信ネットワークに接続するためのインタフェースである。変換装置100が有する各機能部や学習装置200が有する各機能部を実現する1以上のプログラムは、通信I/F307を介して、所定のサーバ装置等が取得 (ダウンロード) されても良い。

[0187] 補助記憶装置308は、例えば、HDD (Hard Disk Drive) やSSD (So

lid State Drive) 等であり、プログラムやデータを格納している不揮発性の記憶装置である。補助記憶装置 308 に格納されているプログラムやデータには、例えば、OS、変換装置 100 が有する各機能部や学習装置 200 が有する各機能部を実現する 1 以上のプログラム等がある。

[0188] 本発明の実施の形態における変換装置 100 及び学習装置 200 は、図 6 に示すハードウェア構成を有することにより、上述した各種処理を実現することができる。

[0189] 本発明は、具体的に開示された上記の実施の形態に限定されるものではなく、特許請求の範囲から逸脱することなく、種々の変形や変更が可能である。

符号の説明

[0190]	100	変換装置
	101	前処理部
	102	変換処理部
	200	学習装置
	201	学習用データ入力部
	202	パラメータ更新部

請求の範囲

[請求項1] 入力された第1のデータXをニューラルネットワークにより第2のデータYに変換する変換装置であって、

前記第1のデータXに対して所定の前処理を行うことで得られた第3のデータ θ と、max関数に対して強凸正則化関数 Ω を導入したmax $_{\Omega}$ 関数を用いて再帰的に定義されるDP $_{\Omega}$ 関数とを用いて、重み付き有向非巡回グラフGで表される問題を対象とした動的計画法の解の近似DP $_{\Omega}(\theta)$ を計算する計算手段と、

前記計算手段により計算された前記DP $_{\Omega}(\theta)$ と、該DP $_{\Omega}(\theta)$ の勾配 $\nabla \text{DP}_{\Omega}(\theta)$ との少なくとも一方を前記第2のデータYとして出力する出力手段と、

を有することを特徴とする変換装置。

[請求項2] 前記DP $_{\Omega}$ 関数は、

[数66]

$$\max_{\Omega}(x) \triangleq \max_{q \in \Delta^D} \langle q, x \rangle - \Omega(q)$$

で定義される前記max $_{\Omega}$ 関数を用いて、 $i = 1, \dots, N$ に対して、

[数67]

$$v_1(\theta) \triangleq 0$$

$$v_i(\theta) \triangleq \max_{j \in \mathcal{P}_i} \theta_{i,j} + v_j(\theta)$$

ここで $\mathcal{P}_i$ はGにおけるノードiの親ノードの集合を表す。

で再帰的に定義される $v_i(\theta)$ によって、 $DP_\Omega(\theta) = v_N(\theta)$ と定義される、ことを特徴とする請求項 1 に記載の変換装置。

[請求項3] 前記強凸正則化関数 Ω は、

$\gamma > 0$ として、

[数68]

$$\Omega(q) = -\gamma H(q)$$

$$-H(q) = \sum_{i=1}^D q_i \log q_i$$

又は

[数69]

$$\Omega(q) = \frac{\gamma}{2} \|q\|_2^2$$

のいずれかである、ことを特徴とする請求項 1 又は 2 に記載の変換装置。

[請求項4] 入力された第 1 のデータ X を第 2 のデータ Y に変換するニューラルネットワークの学習を行う学習装置であって、

前記第 1 のデータ X に対して所定の前処理を行うことで得られた第 3 のデータ θ と、 $\max$ 関数に対して強凸正則化関数 Ω を導入した $\max_\Omega$ 関数を用いて再帰的に定義される DP_Ω 関数とを用いて、重み付き有向非巡回グラフ G で表される問題を対象とした動的計画法の解の近似 $DP_\Omega(\theta)$ を計算する計算手段と、

前記計算手段により計算された前記 $DP_\Omega(\theta)$ と、該 $DP_\Omega(\theta)$ の勾配 $\nabla DP_\Omega(\theta)$ との少なくとも一方を前記第 2 のデータ Y と

して出力する出力手段と、

前記出力手段により出力された前記 $DP_{\Omega}(\theta)$ 又は前記 $\nabla DP_{\Omega}(\theta)$ と、前記第1のデータ X に対する正解データ Y_{true} とを用いた損失関数の微分値に基づいて、前記ニューラルネットワークのパラメータである前記 θ を更新する更新手段と、
を有することを特徴とする学習装置。

[請求項5]

前記損失関数は、

前記出力手段により前記 $DP_{\Omega}(\theta)$ が出力された場合は、 $DP_{\Omega}(\theta) - \langle Y_{true}, \theta \rangle$ であり、

前記出力手段により前記 $\nabla DP_{\Omega}(\theta)$ が出力された場合は、ダイバーバージェンス $\Delta(Y_{true}, \nabla DP_{\Omega}(\theta))$ である、ことを特徴とする請求項4に記載の学習装置。

[請求項6]

入力された第1のデータ X をニューラルネットワークにより第2のデータ Y に変換するコンピュータが、

前記第1のデータ X に対して所定の前処理を行うことで得られた第3のデータ θ と、 $\max$ 関数に対して強凸正則化関数 Ω を導入した $\max_{\Omega}$ 関数を用いて再帰的に定義される DP_{Ω} 関数とを用いて、重み付き有向非巡回グラフ G で表される問題を対象とした動的計画法の解の近似 $DP_{\Omega}(\theta)$ を計算し、

計算された前記 $DP_{\Omega}(\theta)$ と、該 $DP_{\Omega}(\theta)$ の勾配 $\nabla DP_{\Omega}(\theta)$ との少なくとも一方を前記第2のデータ Y として出力する、

処理を実行することを特徴とする変換方法。

[請求項7]

入力された第1のデータ X を第2のデータ Y に変換するニューラルネットワークの学習を行うコンピュータが、

前記第1のデータ X に対して所定の前処理を行うことで得られた第3のデータ θ と、 $\max$ 関数に対して強凸正則化関数 Ω を導入した $\max_{\Omega}$ 関数を用いて再帰的に定義される DP_{Ω} 関数とを用いて、重み付き有向非巡回グラフ G で表される問題を対象とした動的計画法の解

の近似 $DP_{\Omega}(\theta)$ を計算し、

計算された前記 $DP_{\Omega}(\theta)$ と、該 $DP_{\Omega}(\theta)$ の勾配 $\nabla DP_{\Omega}(\theta)$ との少なくとも一方を前記第2のデータ Y として出力し、

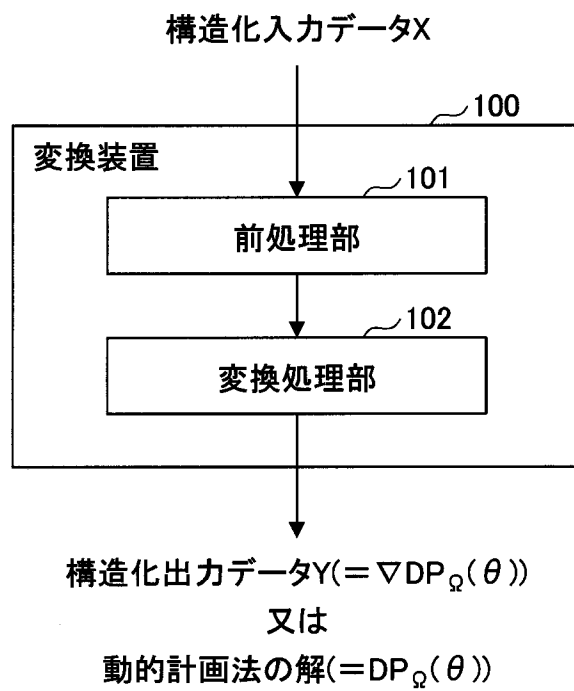
出力された前記 $DP_{\Omega}(\theta)$ 又は前記 $\nabla DP_{\Omega}(\theta)$ と、前記第1のデータ X に対する正解データ Y_{true} とを用いた損失関数の微分値に基づいて、前記ニューラルネットワークのパラメータである前記 θ を更新する、

処理を実行することを特徴とする学習方法。

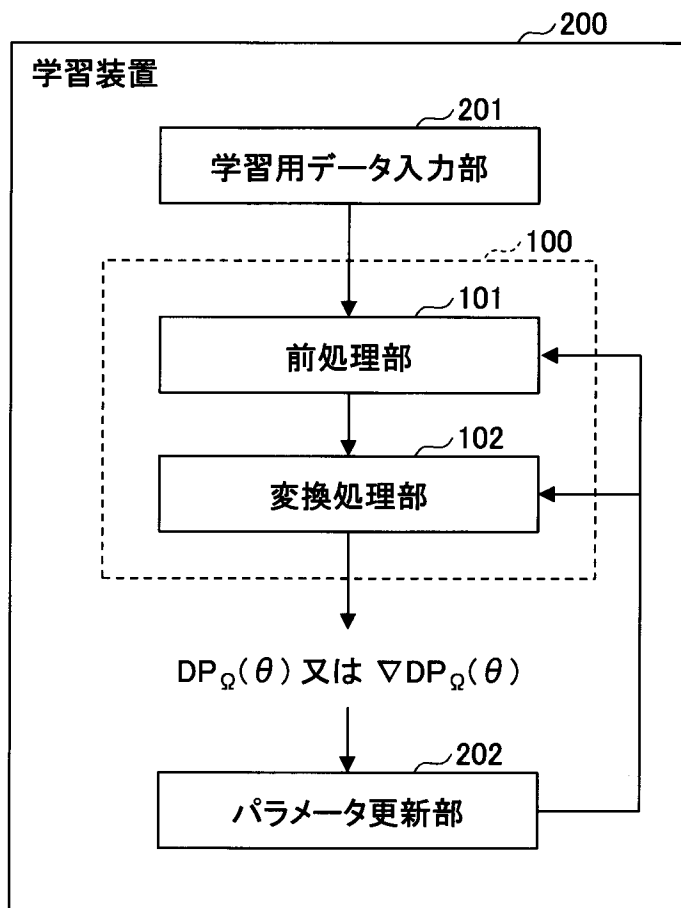
[請求項8]

コンピュータを、請求項1乃至3の何れか一項に記載の変換装置における各手段、又は、請求項4又は5に記載の学習装置における各手段として機能させるためのプログラム。

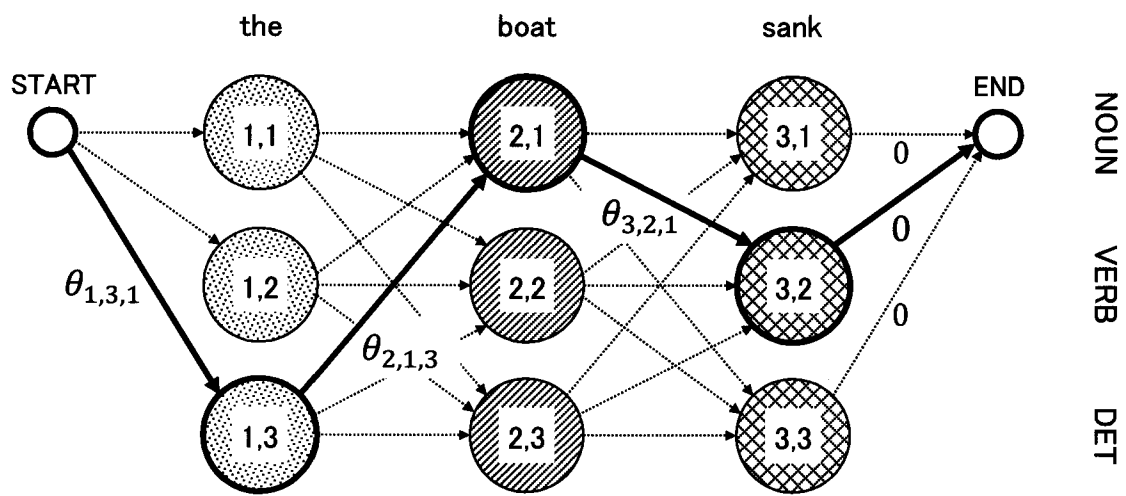
[図1]



[図2]

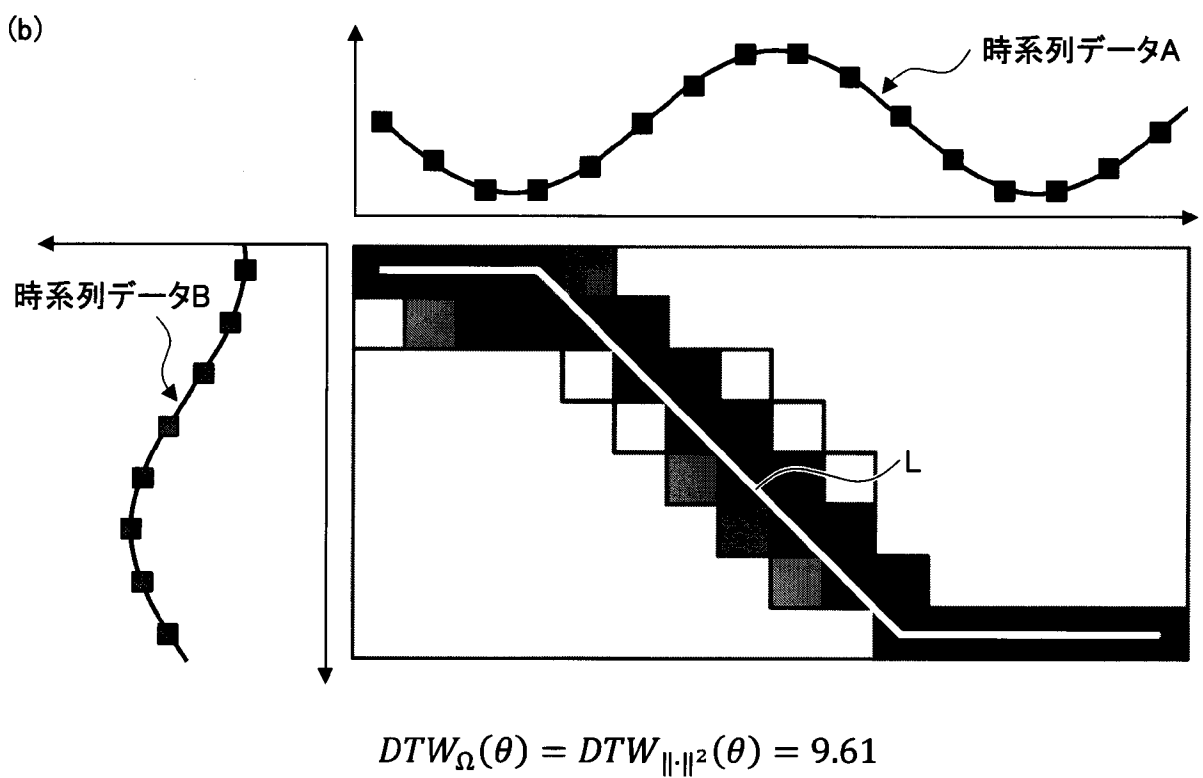
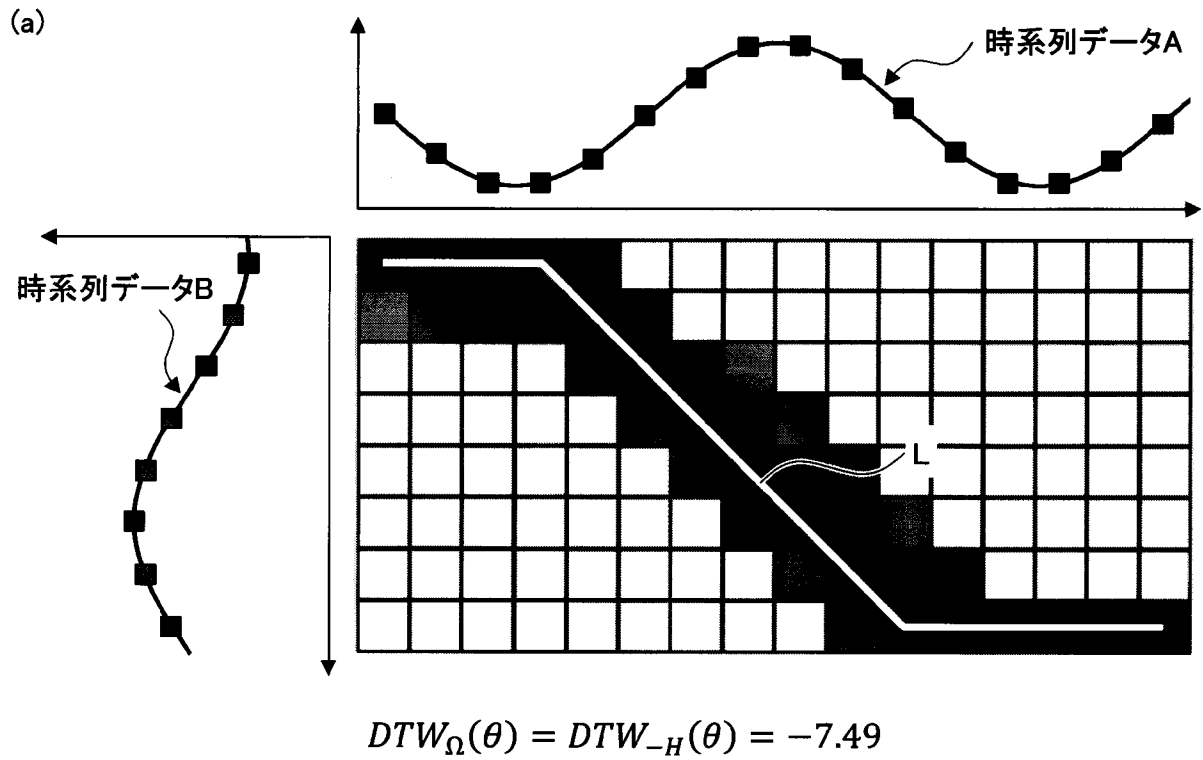


[図3]

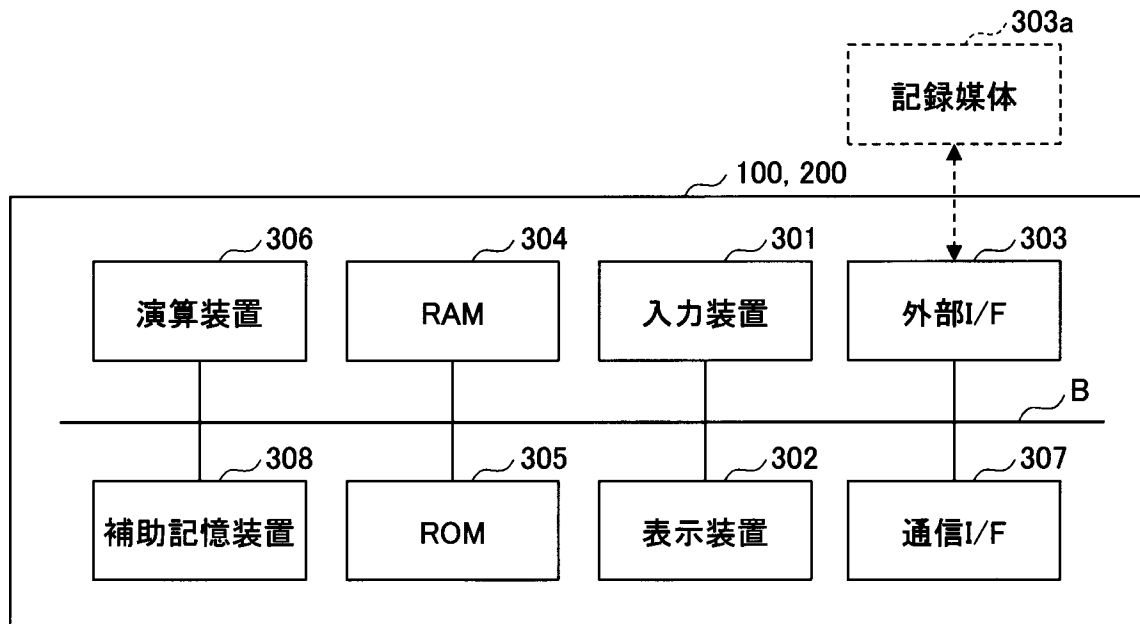


$$\langle Y, \theta \rangle = \theta_{1,3,1} + \theta_{2,1,3} + \theta_{3,2,1}$$

[図5]



[図6]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2019/025636

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06N3/08 (2006.01) i, G06N99/00 (2019.01) i

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)

Int.Cl. G06N3/08, G06N99/00

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	1971-2019
Registered utility model specifications of Japan	1996-2019
Published registered utility model applications of Japan	1994-2019

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.
A	篠原 正太ほか, ニューラルネットワークを用いた系列ラベリングによる単語分割手法, 第9回データ工学と情報マネジメントに関するフォーラム, 日本データベース学会, 08 March 2017, non-official translation (SHINOHARA, Shota et al., "Word segmentation method by sequence labeling using neural network", The 9th forum of data engineering and information management, The Database society of Japan)	1-8

☐ 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

"I" 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
 17 September 2019 (17.09.2019)

Date of mailing of the international search report
 01 October 2019 (01.10.2019)

Name and mailing address of the ISA/
 Japan Patent Office
 3-4-3, Kasumigaseki, Chiyoda-ku,
 Tokyo 100-8915, Japan

Authorized officer

 Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06N3/08(2006.01)i, G06N99/00(2019.01)i

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（I P C））

Int.Cl. G06N3/08, G06N99/00

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	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 9 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 9 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 9 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
A	篠原 正太ほか、ニューラルネットワークを用いた系列ラベリングによる単語分割手法、第9回データ工学と情報マネジメントに関するフォーラム、日本データベース学会、2017.03.08	1-8

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	「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矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「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	「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
「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）	「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
「O」 口頭による開示、使用、展示等に言及する文献	「&」 同一パテントファミリー文献
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願	

国際調査を完了した日

1 7 . 0 9 . 2 0 1 9

国際調査報告の発送日

0 1 . 1 0 . 2 0 1 9

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

塚田 肇

5 B

3 6 5 2

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 5