

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7140194号
(P7140194)

(45)発行日 令和4年9月21日(2022.9.21)

(24)登録日 令和4年9月12日(2022.9.12)

(51)国際特許分類	F I		
G 1 0 L 25/51 (2013.01)	G 1 0 L 25/51		
G 0 5 B 23/02 (2006.01)	G 0 5 B 23/02	3 0 2 R	
G 0 1 M 99/00 (2011.01)	G 0 5 B 23/02	3 0 2 Y	
	G 0 1 M 99/00	Z	

請求項の数 14 (全43頁)

(21)出願番号	特願2020-536385(P2020-536385)	(73)特許権者	000004226 日本電信電話株式会社 東京都千代田区大手町一丁目5番1号
(86)(22)出願日	令和1年7月4日(2019.7.4)	(74)代理人	100121706 弁理士 中尾 直樹
(86)国際出願番号	PCT/JP2019/026556	(74)代理人	100128705 弁理士 中村 幸雄
(87)国際公開番号	WO2020/031570	(74)代理人	100147773 弁理士 義村 宗洋
(87)国際公開日	令和2年2月13日(2020.2.13)	(72)発明者	山口 正隆 東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内
審査請求日	令和3年1月21日(2021.1.21)	(72)発明者	小泉 悠馬 東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内
(31)優先権主張番号	特願2018-151412(P2018-151412)		
(32)優先日	平成30年8月10日(2018.8.10)		
(33)優先権主張国・地域又は機関	日本国(JP)		
(31)優先権主張番号	特願2018-209416(P2018-209416)		
(32)優先日	平成30年11月7日(2018.11.7)		
(33)優先権主張国・地域又は機関	日本国(JP)		
特許法第30条第2項適用 (1) 発行日 2018年 最終頁に続く		最終頁に続く	

(54)【発明の名称】 異常検知装置、確率分布学習装置、自己符号化器学習装置、プログラム

(57)【特許請求の範囲】

【請求項1】

異常検知対象機器とは異なる1以上の機器から発せられた正常音の分布を示す第1確率分布と前記異常検知対象機器から発せられた正常音(以下、適応学習用正常音という)との関連付けに基づいて、前記異常検知対象機器から発せられた音(以下、異常検知対象音という)から、当該機器の異常の程度を示す異常度を推定する異常度推定部とを含む異常検知装置。

【請求項2】

請求項1に記載の異常検知装置であって、前記関連付けは、前記適応学習用正常音を用いて、前記第1確率分布を更新することにより、得た前記異常検知対象機器から発せられた正常音の分布を示す第2確率分布であることを特徴とする異常検知装置。

【請求項3】

請求項2に記載の異常検知装置であって、前記第2確率分布 $q_2(x;)$ の変数 x は、前記異常検知対象機器から発せられた正常音から生成された入力データを示す変数であり、前記変数 x は、変換 $f_i (i=1, \dots, K, Kは1以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する)$ と潜在変数 z_0 を用いて、 $x=f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表されるものであり、 $q_0(z_0)$ を潜在変数 z_0 の確率分布とし、

入力データ x の確率密度 $q_2(x; \theta)$ は、入力データ x に対応する潜在変数 $z_0=f_1^{-1}(f_2^{-1}(\dots(f_{K-1}^{-1}(x))\dots))$ の確率密度 $q_0(z_0)$ を用いて計算される

ことを特徴とする異常検知装置。

【請求項 4】

請求項 3 に記載の異常検知装置であって、

前記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも 1 つの変換に対して、その逆変換は、適応バッチ正規化である

ことを特徴とする異常検知装置。

【請求項 5】

請求項 3 に記載の異常検知装置であって、

前記第 2 確率分布 $q_2(x; \theta)$ は、次式で表される

【数 3 9】

$$q_2(x; \theta) = q_0(z_0) \left| \det\left(\frac{\partial f_K(z_{K-1}; \theta_K)}{\partial z_{K-1}}\right) \right| \\ \left| \det\left(\frac{\partial f_{K-1}(z_{K-2}; \theta_{K-1})}{\partial z_{K-2}}\right) \right| \cdots \left| \det\left(\frac{\partial f_1(z_0; \theta_1)}{\partial z_0}\right) \right|$$

(ただし、 $z_1=f_1(z_0; \theta_1)$, $z_2=f_2(z_1; \theta_2)$, ..., $x=f_K(z_{K-1}; \theta_K)$, $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_K^T]^T$ である)

ことを特徴とする異常検知装置。

【請求項 6】

請求項 4 または 5 に記載の異常検知装置であって、

前記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも 1 つの変換に対して、その逆変換は、線形変換であり、

前記線形変換に対応する行列は、下方三角行列と上方三角行列の積、または、下方三角行列と対角行列と上方三角行列の積として表されている

ことを特徴とする異常検知装置。

【請求項 7】

異常検知対象機器とは異なる 1 以上の機器から発せられた正常音（以下、学習用正常音という）から、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の分布を示す第 1 確率分布を学習する学習部と

を含む確率分布学習装置であって、

前記第 1 確率分布 $q_1(x; \theta)$ の変数 x は、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音から生成された入力データを示す変数であり、

前記変数 x は、変換 $f_i(i=1, \dots, K, K$ は 1 以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する)と潜在変数 z_0 を用いて、 $x=f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表されるものであり、

$q_0(z_0)$ を潜在変数 z_0 の確率分布とし、

入力データ x の確率密度 $q_1(x; \theta)$ は、入力データ x に対応する潜在変数 $z_0=f_1^{-1}(f_2^{-1}(\dots(f_{K-1}^{-1}(x))\dots))$ の確率密度 $q_0(z_0)$ を用いて計算されるものであり、

前記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも 1 つの変換に対して、その逆変換は、適応バッチ正規化である

確率分布学習装置。

【請求項 8】

請求項 7 に記載の確率分布学習装置であって、

前記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも 1 つの変換に対して、その逆変換は、線形変換であり、

前記線形変換に対応する行列は、下方三角行列と上方三角行列の積、または、下方三角行列と対角行列と上方三角行列の積として表されている

ことを特徴とする確率分布学習装置。

【請求項 9】

10

20

30

40

50

異常検知対象機器とは異なる 1 以上の機器から発せられた正常音を復元する第 1 自己符号化器と前記異常検知対象機器から発せられた正常音（以下、適応学習用正常音という）との関連付けに基づいて、前記異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器の異常の程度を示す異常度を推定する異常度推定部とを含む異常検知装置。

【請求項 10】

請求項 9 に記載の異常検知装置であって、
前記関連付けは、
前記適応学習用正常音を用いて、前記第 1 自己符号化器を更新することにより、得た前記異常検知対象機器から発せられた正常音を復元する第 2 自己符号化器であることを特徴とする異常検知装置。

10

【請求項 11】

請求項 10 に記載の異常検知装置であって、
前記第 1 自己符号化器は、適応バッチ正規化の計算を実行する AdaBN 層を含むニューラルネットワークとして構成されるものである
ことを特徴とする異常検知装置。

【請求項 12】

異常検知対象機器とは異なる 1 以上の機器から発せられた正常音（以下、学習用正常音という）から、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音を復元する第 1 自己符号化器を学習する学習部と
を含む自己符号化器学習装置であって、
前記第 1 自己符号化器は、適応バッチ正規化の計算を実行する AdaBN 層を含むニューラルネットワークとして構成されるものである
自己符号化器学習装置。

20

【請求項 13】

異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の集合と前記異常検知対象機器から発せられた正常音（以下、適応学習用正常音という）との関連付けに基づいて、前記異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器の異常の程度を示す異常度を推定する異常度推定部と
を含む異常検知装置。

30

【請求項 14】

請求項 1 から 6、または、9 から 11、または、13 のいずれか 1 項に記載の異常検知装置としてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、異常検知技術及びドメイン変換技術に関する。

【背景技術】

【0002】

例えば、故障により、工場に設置された大型の製造機・造型機などの業務用機器の稼働がストップすると、業務に大きな支障をもたらす。そのため、動作状況を日常的に監視し、異常が発生した場合にはただちに対処を行う必要がある。この問題に対応する方法として、業務用機器の管理業者が定期的に現場へ整備員を派遣し、整備員がパーツの摩耗などを確認する方法がある。しかし、この方法は、多大な費用（例えば、人件費や移動費）及び労力がかかるため、すべての業務用機器や工場を対象に実施するのは難しい。そこで、代替手段として、機器の内部に設置したマイクロホンが収音する動作音に基づいて日常的に動作状況を監視する方法がある。具体的には、当該動作音を解析し、異常と思われる音（異常音）を検知した場合にはアラートをあげる。このように、音を利用して監視対象である機器が正常状態にあるか異常状態にあるかを判定する技術を異常音検知技術という。

40

【0003】

50

異常音検知技術において、機器の種類や個体毎に、異常音の種類と検出方法を設定するのはコストがかかる。そのため、異常音を検知するためのルールを自動設計する必要がある。その解決方法の1つとして、統計的手法に基づく異常音検知がよく知られている（非特許文献1）。この統計的手法に基づく異常音検知は、教師あり異常音検知と教師なし異常音検知に大別できる。教師あり異常音検知では、正常音と異常音を学習データとして大量に収集し、識別率（正常音と異常音を識別できる割合）を最大化するように識別器を学習する。一方、教師なし異常音検知では、正常音のみを学習データとして大量に収集し、正常音の発生に関する確率分布（以下、正常モデルという）を学習する。そして、正常モデルを用いて、新しく収集した音（つまり、異常検知対象となる音）が正常音に類似している（尤度が高い）と判定される場合は正常、類似していない（尤度が低い）と判定される場合は異常と判定する。

10

【0004】

産業的応用分野では異常音の学習データを大量に集めることが困難であるため、教師なし異常音検知が採用されることが多い。また、産業的応用分野では、大量の同種の機器に対してそれぞれを異常検知の対象としたい場合がある。例えば、データセンタに存在する膨大なサーバの異常音を監視したい場合である。この場合、各サーバに対してそれぞれ異常検知システムを適用することになるが、各サーバから発せられる音の分布は設置場所や組み立て誤差に起因して微妙に異なることが想定される。そこで、このような大量の同種の機器に対して異常検知システムを適用する方法として、以下のような方法が考えられる。

(1) いくつかの機器から収集した正常音を用いて、これらの機器に共通の正常モデルを学習する。そして、この共通のモデルを用いてすべての機器の異常検知を行う。

(2) 機器ごとに収集した正常音を用いて、機器ごとに異なる正常モデルを学習する。そして、この個々のモデルを用いて各機器の異常検知を行う。

20

【0005】

(1)の方法は、機器ごとに学習を行わないため、監視したい機器が増加しても学習データ収集やモデル学習を行う必要がなく、データ収集や学習に係るコストが抑えられる一方、機器ごとの正常音発生に関する微妙な違いを捉えることができないため、精度が高い異常検知を行えない可能性がある。また、(2)の方法は、各機器から得られた正常音だけを用いて学習を行うため精度の高い正常モデルが生成されることが期待される一方、機器ごとに学習データを収集してモデル学習を行わなければならないため、監視したい機器の増加に伴い、データ収集や学習に係るコストがかかってしまうという問題がある。

30

【0006】

次に、ドメイン変換について説明する。ドメイン変換とは、あるドメインのデータを、当該ドメインとは異なるドメインのデータに変換する技術である。ここで、対象とするデータは、例えば、画像や音である。例えば、参考非特許文献1にあるように、「風景写真の画像」を「風景画の画像」に変換したり、「馬の写真の画像」を「シマウマの写真の画像」に変換したりするものである。

(参考非特許文献1: Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", arXiv:1703.10593v5, <https://arxiv.org/abs/1703.10593v5>)

40

ドメイン変換を実現するためには、ドメインDからドメインD'へのデータ変換器を作ればよい。このようなデータ変換器を作るために、例えば「風景写真」とそれを写実した「風景画」のペアを学習データとして大量に集め、ニューラルネットワークを用いて「風景写真」から「風景画」への変換を学習させるという方法を考えることができる。以下、このように2つのドメインのデータのペアを用いてデータ変換器を学習させる枠組みのことをペアデータありドメイン変換と呼ぶ。ペアデータありドメイン変換は、入力とそれに対する正解となる2つのドメインのデータのペアを学習データとして、比較的簡単に学習によりデータ変換器を構成できるという利点がある。しかし、学習データを多数集めなければならない。先ほどの「風景写真」と「風景画」の例では、まず「風景写真」を集め、その上で（例えば、画家に作成を依頼するなどして）「風景写真」を写実した「風景画」を

50

作成する必要がある。また、「馬の写真」と「シマウマの写真」の例では、同じ構図の写真を撮ることは難しいため、学習データを集めることは現実的には不可能である。

【0007】

そこで、近年、学習データ収集に関する問題を解決するために、ペアデータを用いることなくデータ変換器を学習することができる枠組みが提案されている。この枠組みのことをペアデータなしドメイン変換という。ペアデータなしドメイン変換では、ドメインDのデータとドメインD'のデータを用いて、ドメインDのデータからドメインD'のデータに変換するデータ変換器を学習する。ここで、学習に用いるドメインDのデータとドメインD'のデータは、ペアである必要はない。したがって、「馬の写真」と「シマウマの写真」のようなペアデータを収集することが困難なドメイン間での変換であってもデータ変換器を学習することが可能となる。

10

【0008】

ペアデータなしドメイン変換の例として、例えば、非特許文献2に記載のStarGANという手法が提案されている。

【先行技術文献】

【非特許文献】

【0009】

【文献】井出剛，杉山将，“異常検知と変化検知”，講談社，pp.6-7，2015.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, pp.8789-8797, 2018.

20

【発明の概要】

【発明が解決しようとする課題】

【0010】

まず、第1の課題について説明する。

上述の通り、(1)、(2)の方法には、データ収集や学習に係るコストと異常検知の精度のトレードオフの問題がある。そこで、第3の方法として、以下のような方法も考えられる。

(3)いくつかの機器から収集した正常音を用いて、これらの機器に共通の正常モデルを学習する。次に、機器ごとに収集した正常音を用いて、この共通のモデルから機器ごとに異なる正常モデルを適応的に学習する。そして、この個々の適応済み正常モデルを用いて各機器の異常検知を行う。

30

【0011】

(3)の方法は、上記トレードオフの問題を解決できる可能性があるが、実際にこのような方法を実現するためには、異常検知の対象とする機器ごとに比較的少量の正常音を収集すればよいこと、当該正常音を用いて共通の正常モデルから各機器に適応した正常モデルを効率的に学習することが必要になる。しかし、これまでそのような方法は開発されていなかった。

【0012】

これが第1の課題である。

【0013】

次に、第2の課題について説明する。

非特許文献2のStarGANでは、参考非特許文献2に記載のGenerative Adversarial Networks (GAN)という手法を利用してペアデータなしドメイン変換を実現しているが、学習が不安定であるという問題がある。

(参考非特許文献2: Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, “Generative Adversarial Nets”, Advances in Neural Information Processing Systems 27 (NIPS 2014), 2018.)

40

50

【 0 0 1 4 】

これが第 2 の課題である。

【 0 0 1 5 】

そこで本発明では、正常モデル学習に必要なコストを抑えつつ、精度が高い異常検知技術と、安定したペアデータなし学習が可能となるドメイン変換技術を提供することを目的とする。

【課題を解決するための手段】

【 0 0 1 6 】

本発明の一態様は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の分布を示す第 1 確率分布と前記異常検知対象機器から発せられた正常音（以下、適応学習用正常音という）との関連付けに基づいて、前記異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器の異常の程度を示す異常度を推定する異常度推定部とを含む。

10

【 0 0 1 7 】

本発明の一態様は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音（以下、学習用正常音という）から、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の分布を示す第 1 確率分布を学習する学習部とを含む確率分布学習装置であって、前記第 1 確率分布 $q_1(x; \cdot)$ の変数 x は、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音から生成された入力データを示す変数であり、前記変数 x は、変換 f_i ($i=1, \dots, K$, K は 1 以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する) と潜在変数 z_0 を用いて、 $x=f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表されるものであり、 $q_0(z_0)$ を潜在変数 z_0 の確率分布とし、入力データ x の確率密度 $q_1(x; \cdot)$ は、入力データ x に対応する潜在変数 $z_0=f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x))\dots))$ の確率密度 $q_0(z_0)$ を用いて計算されるものであり、前記変換 f_i ($i=1, \dots, K$) のうち、少なくとも 1 つの変換に対して、その逆変換は、適応バッチ正規化である。

20

【 0 0 1 8 】

本発明の一態様は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音を復元する第 1 自己符号化器と前記異常検知対象機器から発せられた正常音（以下、適応学習用正常音という）との関連付けに基づいて、前記異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器の異常の程度を示す異常度を推定する異常度推定部とを含む。

30

【 0 0 1 9 】

本発明の一態様は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音（以下、学習用正常音という）から、前記異常検知対象機器とは異なる 1 以上の機器から発せられた正常音を復元する第 1 自己符号化器を学習する学習部とを含む自己符号化器学習装置であって、前記第 1 自己符号化器は、適応バッチ正規化の計算を実行する AdaBN 層を含むニューラルネットワークとして構成されるものである。

【 0 0 2 0 】

本発明の一態様は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の集合と前記異常検知対象機器から発せられた正常音（以下、適応学習用正常音という）との関連付けに基づいて、前記異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器の異常の程度を示す異常度を推定する異常度推定部とを含む。

40

【 0 0 2 1 】

本発明の一態様は、第 1 ドメインのドメインデータに対応する入力データから、潜在変数を計算する潜在変数計算部と、前記潜在変数から、第 2 ドメインのドメインデータに対応する出力データを計算する出力データ計算部とを含むデータ変換装置であって、前記潜在変数計算部は、逆関数を有する所定の関数（以下、第 1 関数という）を用いて計算するものであり、前記出力データ計算部は、逆関数を有する所定の関数（以下、第 2 関数という）を用いて計算するものであり、第 1 関数と第 2 関数は、潜在変数 z_0 を変数 x に変換する所定の関数から導出されるものである。

【発明の効果】

50

【 0 0 2 2 】

本発明によれば、正常モデル学習に必要なコストを抑えつつ、精度の高い異常検知が可能となる。また、本発明によれば、安定したペアデータなしの学習が可能となるドメイン変換を実現することができる。

【 図面の簡単な説明 】

【 0 0 2 3 】

【 図 1 】 確率分布学習装置 1 0 0 の構成の一例を示すブロック図である。

【 図 2 】 確率分布学習装置 1 0 0 の動作の一例を示すフローチャートである。

【 図 3 】 確率分布適応学習装置 2 0 0 の構成の一例を示すブロック図である。

【 図 4 】 確率分布適応学習装置 2 0 0 の動作の一例を示すフローチャートである。

10

【 図 5 】 異常検知装置 3 0 0 の構成の一例を示すブロック図である。

【 図 6 】 異常検知装置 3 0 0 の動作の一例を示すフローチャートである。

【 図 7 】 異常度推定部 3 2 0 の構成の一例を示すブロック図である。

【 図 8 】 異常度推定部 3 2 0 の動作の一例を示すフローチャートである。

【 図 9 】 自己符号化器学習装置 4 0 0 の構成の一例を示すブロック図である。

【 図 1 0 】 自己符号化器学習装置 4 0 0 の動作の一例を示すフローチャートである。

【 図 1 1 】 自己符号化器適応学習装置 5 0 0 の構成の一例を示すブロック図である。

【 図 1 2 】 自己符号化器適応学習装置 5 0 0 の動作の一例を示すフローチャートである。

【 図 1 3 】 異常検知装置 6 0 0 の構成の一例を示すブロック図である。

【 図 1 4 】 異常検知装置 6 0 0 の動作の一例を示すフローチャートである。

20

【 図 1 5 】 異常度推定部 6 2 0 の構成の一例を示すブロック図である。

【 図 1 6 】 異常度推定部 6 2 0 の動作の一例を示すフローチャートである。

【 図 1 7 】 確率分布学習装置 1 1 0 0 の構成の一例を示すブロック図である。

【 図 1 8 】 確率分布学習装置 1 1 0 0 の動作の一例を示すフローチャートである。

【 図 1 9 】 データ変換装置 1 2 0 0 の構成の一例を示すブロック図である。

【 図 2 0 】 データ変換装置 1 2 0 0 の動作の一例を示すフローチャートである。

【 図 2 1 】 潜在変数計算部 1 2 2 0 と出力データ計算部 2 3 0 による処理の様子を示す図である。

【 図 2 2 】 データ変換装置 1 3 0 0 の構成の一例を示すブロック図である。

【 図 2 3 】 データ変換装置 1 3 0 0 の動作の一例を示すフローチャートである。

30

【 図 2 4 】 本発明の実施形態における各装置を実現するコンピュータの機能構成の一例を示す図である。

【 発明を実施するための形態 】

【 0 0 2 4 】

以下、本発明の実施の形態について、詳細に説明する。なお、同じ機能を有する構成部には同じ番号を付し、重複説明を省略する。

【 0 0 2 5 】

< 記法 >

$_$ (アンダースコア) は下付き添字を表す。例えば、 x^{y_z} は y_z が x に対する上付き添字であり、 x_{y_z} は y_z が x に対する下付き添字であることを表す。

40

【 0 0 2 6 】

まず、本発明の第 1 実施形態から第 3 実施形態に対する技術的背景と、各実施形態について説明する。

【 0 0 2 7 】

< 技術的背景 >

本発明の実施形態は、複数の同一種類の機器に適用可能な異常検知の枠組みを提供するものである。具体的には、上述の (3) の方法による枠組みを提供する。各機器から発せられる正常音の分布の違いは音に関する特徴量の統計量で表現されるという仮説に基づき、特徴量の 1 次統計量である平均と 2 次統計量である分散を異なる機器間で一致させることにより、異なる機器の正常モデルを一つのモデルから導出できるようにするものである。

50

【 0 0 2 8 】

まず、本発明の実施形態で用いる従来技術について説明する。

【 0 0 2 9 】

《教師なし異常音検知》

異常音検知とは、異常検知対象となる音（観測信号）を発した監視対象機器の状況が正常であるか異常であるかを判定するタスクである。ここで、観測信号から生成される入力データ x には、例えば次式のように観測信号の対数振幅スペクトル $\ln |X_{t,f}|$ を要素とするベクトルを用いることができる。

【 0 0 3 0 】

【数 1】

$$x := (\ln |X_{t-Q,1}|, \ln |X_{t-Q,2}|, \dots, \ln |X_{t-Q,F}|, \ln |X_{t-Q+1,1}|, \ln |X_{t-Q+1,2}|, \dots, \ln |X_{t-Q+1,F}|)^T \quad \dots(1)$$

10

【 0 0 3 1 】

ここで、 $t = \{1, \dots, T\}$ 、 $f = \{1, \dots, F\}$ は、それぞれ時間のインデックス、周波数のインデックスを表す。また、 Q は入力で考慮する過去・未来のフレーム数を表す。

【 0 0 3 2 】

入力データ x は、上記の例に限るものではなく、観測信号から生成される入力データとして、その他の特徴量を用いてもよい。

20

【 0 0 3 3 】

次に、密度推定ベースの異常音検知について説明する。正常音から生成される入力データ（以下、単に正常音の入力データという）は確率分布 $p(x)$ に従って生成されると仮定する。まず、パラメータ θ を持つ確率分布 $q(x; \theta)$ を設計する。次に、確率分布 $p(x)$ から生成された N 個の正常音の入力データの集合 $\{x_i\}_{i=1}^N$ を用いて、 $q(x; \theta)$ が $p(x)$ に最も近くなるようなパラメータ θ^* を求め、 $q(x; \theta^*)$ を $p(x)$ の近似分布とする。そして、異常検知対象となる音の入力データが入力された場合、例えば次式によりその入力データに関する異常度 $A(x; \theta^*)$ を求める。

【 0 0 3 4 】

【数 2】

$$A(x; \theta^*) = -\ln q(x; \theta^*) \quad \dots(2)$$

30

【 0 0 3 5 】

最後に、判定結果 R を、例えば次式により求める。

【 0 0 3 6 】

【数 3】

$$R = H(A(x; \theta^*) - \phi) \quad \dots(3)$$

40

【 0 0 3 7 】

ここで、閾値 ϕ は所定の定数、 $H(\cdot)$ は引数が非負なら1、負なら0を返すステップ関数である。 $R=1$ の場合、当該異常検知対象となる音を発した機器は異常、 $R=0$ の場合、正常と判定する。つまり、異常度 $A(x; \theta^*)$ が事前に設定した閾値 ϕ より大きければ異常と判定する。

【 0 0 3 8 】

上述の通り、密度推定ベースの異常音検知では、(1) 確率分布 $q(x; \theta)$ の設計、(2) パラメータ θ の決定を行う必要がある。(1)の確率分布 $q(x; \theta)$ の設計には、ノーマライジング・フロー(Normalizing Flow)を用いることができる。また、(2)のパラメータ θ の決定には、例えば、損失関数(目的関数)を $L(\theta) = -\sum \log q(x_i; \theta)$ (つまり、正常音

50

の入力データの集合 $\{x_i\}_{i=1}^N$ に対する負の対数尤度の和)とする勾配降下法による学習を用いることができる。

【0039】

《ノーマライジング・フロー(Normalizing Flow)》

ノーマライジング・フローは、データ生成に関する確率分布 $p(x)$ を近似する分布を求める手法である。

【0040】

以下、説明する。 $\{f_i(z)\}_{i=1}^K$ を逆変換を有する K 個の変換(ただし、 $f_i(z):R^D \rightarrow R^D$, R は実数の集合、 D は1以上の整数)とする。また、 $f_i^{-1}(z)$ ($i=1, \dots, K$)を $f_i(z)$ の逆変換とする。

【0041】

ノーマライジング・フローでは、 N 個の入力データの集合 $\{x_i\}_{i=1}^N$ に対して、それぞれ対応する潜在変数 $\{z_{0,i}\}_{i=1}^N$ が存在し、入力データ x_i は、 K 個の変換 $\{f_i(z)\}_{i=1}^K$ と x の潜在変数 z_0 を用いた次式により、対応する潜在変数 $z_{0,i}$ を変換したものであると考える。

【0042】

【数4】

$$x = f_K(f_{K-1}(\dots(f_1(z_0))\dots)) \quad \dots(4)$$

【0043】

つまり、 $i=1, \dots, K$ に対して、次式が成り立つ。

【0044】

【数5】

$$x_i = f_K(f_{K-1}(\dots(f_1(z_{0,i}))\dots)) \quad \dots(4)'$$

【0045】

なお、 $z_1=f_1(z_0)$, $z_2=f_2(z_1)$, ..., $x=f_K(z_{K-1})$ と表すこともある。

【0046】

更に、潜在変数 $\{z_{0,i}\}_{i=1}^N$ は、例えば等方ガウス分布のような、モンテカルロサンプリングが容易な確率分布 $q_0(z_0)$ から生成されると仮定する。このとき、入力データの集合 $\{x_i\}_{i=1}^N$ が従う確率分布 $q(x; \theta)$ (x は入力データを示す変数)は、次の形で表現できる。

【0047】

【数6】

$$q(x; \theta) = q_0(z_0) \left| \det(\partial f_K(z_{K-1}; \theta_K) / \partial z_{K-1}) \right| \\ \left| \det(\partial f_{K-1}(z_{K-2}; \theta_{K-1}) / \partial z_{K-2}) \right| \dots \left| \det(\partial f_1(z_0; \theta_1) / \partial z_0) \right| \dots(5)$$

【0048】

ここで、 $z_0=f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x))\dots))$ である。また、 $\{\theta_i\}_{i=1}^K$ は、変換 $\{f_i(z)\}_{i=1}^K$ に対応するパラメータ、 $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_K^T]^T$ である。

【0049】

なお、確率分布 $q_0(z_0)$ は、モンテカルロサンプリングが容易な分布に限るものではなく、厳密な確率密度推定が容易な分布であればよい。厳密な確率密度推定が容易な分布の一例として、以下の条件を満たす確率分布 $p(x)$ が挙げられる。

(条件1) 確率分布 $p(x)$ に対して、 R^D 上の非負関数 $g(x)$ ($g \geq 0$)が存在し、任意の $x \in R^D$ に対して $p(x) = g(x) / \int g(x) dx$ となる。

(条件2) 関数 $g(x)$ に対して $\int g(x) dx$ を計算することは容易である。

【0050】

条件2を満たす関数として、ガウス分布が挙げられる。一方、条件2を満たさない関数として、 $g(x) = \exp(\sin(x) - x^2)$ が挙げられる。

【0051】

10

20

30

40

50

ノーマライジング・フローでは、確率分布 $q(x; \theta)$ のパラメータ θ を入力データの集合 $\{x_i\}_{i=1}^N$ を用いて学習する。そして学習により得られたパラメータ θ^* を用いた確率分布 $q(x; \theta^*)$ により本来のデータ生成に関する確率分布 $p(x)$ を近似する。

【0052】

ノーマライジング・フローでは、変換 $\{f_i(z)\}_{i=1}^K$ として様々なものを用いることができる。例えば、参考非特許文献3に記載のバッチ正規化(Batch Normalization)やLeaky ReLU(Rectified Linear Unit：正規化線形関数)などを用いることができる。また、参考非特許文献4に記載の以下の線型変換を用いることもできる。

【0053】

【数7】

$$f(z) = LUz \quad \dots(6)$$

10

【0054】

ここで、 $L, U \in \mathbb{R}^{D \times D}$ は、それぞれ下三角行列、上三角行列である。この変換は L と U の対角成分の積の絶対値(つまり、 $|\prod_{i=1}^D L_{ii} U_{ii}|$)によりヤコビアン J の行列式の絶対値 $|det(J)|$ を計算することができるため、入力データ x の確率密度 $q(x; \theta)$ を容易に計算することができる(入力データ x の確率密度 $q(x; \theta)$ の計算コストを抑えることができる)という特徴がある(式(5)参照)。

(参考非特許文献3：S. Ioffe, C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift", ICML 2015, 2015.)

20

(参考非特許文献4：J. Oliva, et al., "Transformation Autoregressive Networks", ICML 2018, 2018.)

【0055】

以下、簡単にバッチ正規化BN: $x \rightarrow y (x, y \in \mathbb{R}^D)$ について説明する。バッチ正規化BNは、入力データの集合 $\{x_i\}_{i=1}^N$ の各次元の要素の平均が0、分散が1になるように調整した後、スケール変換とシフト変換を行う。具体的には、次式により、 $y_i = BN(x_i)$ が計算される。

【0056】

【数8】

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad \dots(7a)$$

30

$$s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - m)^2 \quad \dots(7b)$$

$$\hat{x}_i = \frac{x_i - m}{\sqrt{s^2 + \epsilon}} \quad \dots(7c)$$

$$y_i = \gamma \hat{x}_i + \beta \quad \dots(7d)$$

40

【0057】

ここで、 γ, β は、それぞれスケール変換パラメータ、シフト変換パラメータであり、いずれも学習対象となるパラメータである。また、 ϵ は非負の実数であり、ゼロで割ることを避けたい場合は正の実数、避ける必要がない場合はゼロとすればよい。

【0058】

なお、スケール変換パラメータ γ 、シフト変換パラメータ β を明示するため、 $BN(\cdot)$ を $BN_{\gamma, \beta}(\cdot)$ と表すこともある。

【0059】

また、 K 個の変換すべてが同じ種類の変換である必要はない。したがって、例えば、変換 $f_1(z)$ をバッチ正規化、変換 $f_2(z)$ を線形変換というように、いくつかの種類の変換を組

50

み合わせてもよい。

【0060】

《適応バッチ正規化(AdaBN:Adaptive Batch Normalization)》

ドメイン適応とは、モデル学習に用いる学習データの分布と学習済みモデルを用いた処理の対象であるテストデータの分布が異なる場合に、当該分布の相違により、学習済みモデルを用いた処理の精度が低下しないように、学習済みモデルを調整する技術のことである。ここで、学習データの集合、テストデータの集合がドメインであり、それぞれ学習用ドメイン、テスト用ドメインということもある。

【0061】

深層ニューラルネットワーク(DNN:Deep Neural Network)と組み合わせることができるドメイン適応の手法には様々なものがあるが、ここでは適応バッチ正規化について説明する(参考非特許文献5参照)。適応バッチ正規化は、バッチ正規化における平均・分散の計算と平均・分散の調整(式(7a)~(7d)参照)をドメインごとに行うという手法である。つまり、式(7a)~(7c)による計算を同一のドメインのデータごとに行うようにする。実際にテストする際には、テスト用ドメインの入力データの集合 $\{x_i\}_{i=1}^N$ に対して統計量(平均と分散)を計算して、その統計量を用いて式(7c)、式(7d)により処理結果 y_i を出力する。なお、変換が適応バッチ正規化である場合、AdaBN: $x \rightarrow y(x, y \in \mathbb{R}^D)$ と表すこともある。

(参考非特許文献5:Y. Li, et al., "Revisiting Batch Normalization For Practical Domain Adaptation", ICLR 2017, 2016.)

【0062】

以下、本発明の実施形態における異常検知について説明する。まず、本発明の実施形態における異常検知のための問題設定について説明する。次に、上記従来技術を用いた、本発明の実施形態における異常検知の具体的構成について説明する。

【0063】

《問題設定》

解きたい問題は、「複数の機器から得られた大量の正常音を用いて学習した共通の正常モデルである第1確率分布と、異常検知対象機器から得られた少量の正常音とを用いて、異常検知対象機器のために用いることができる正常モデルである第2確率分布を学習し、この第2確率分布を用いて異常検知対象機器が発した音から異常検知することを可能とする」ことである。したがって、以下のデータを扱うことになる。

(1)学習データ:異常検知対象機器とは異なる1以上の機器から発せられた正常音であり、大量に用意することが可能であると仮定する。学習に用いることから、これらの音のことを学習用正常音という。また、その集合のことを学習用ドメインという。なお、正常音の収集対象となる機器は、異常検知対象機器と種類が同じものが好ましい。

(2)適応学習データ:異常検知対象機器から発せられた正常音であり、少量しか用意できないものと仮定する。適応学習に用いることから、これらの音のことを適応学習用正常音という。なお、その集合は、後述するテスト用ドメインである。

(3)テストデータ:異常検知対象機器から発せられた音であり、この音から機器が正常であるか異常であるかを判定する。そこで、この音のことを異常検知対象音という。また、その集合のことをテスト用ドメインという。

【0064】

以下、学習用正常音を用いて学習するフェーズを学習フェーズ、適応学習用正常音を用いて適応学習するフェーズを適応学習フェーズ、異常検知対象音から異常検知を行うフェーズをテストフェーズ(異常検知フェーズ)という。

【0065】

《ポイント》

本発明の実施形態では、少量の適応学習データから第2確率分布を低い計算量で適応学習できるようにするため、ノーマライジング・フローに適応バッチ正規化を導入する。具体的には、ノーマライジング・フローで用いる K 個の変換 $\{f_i(z)\}_{i=1}^K$ のうち少なくとも1

10

20

30

40

50

つの変換 $f_i(z)$ に対して、その逆変換 $f_i^{-1}(z)$ が適応バッチ正規化であるというものである。なお、適応バッチ正規化の計算のうち、スケール変換及びシフト変換、つまり、式(7d)の計算は、省略したものであってもよい。別の言い方をすると、逆変換 $f_i^{-1}(z)$ が $\gamma_i = 1, \beta_i = 0$ である適応バッチ正規化であるとも表現できる。

【 0 0 6 6 】

《 具体的構成 》

以下、具体的構成について説明する。

(1) 学習フェーズ

まず、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音（つまり、学習データ）から生成される入力データ x を、確率分布 $q_0(z_0)$ に従って生成されているとみなす潜在変数 $z_0 (\sim q(z_0))$ に変換するニューラルネットワークについて説明する。ここでは 5 個の変換 $\{f_i(z)\}_{i=1}^5$ を用いる場合について説明する。つまり、 z_0 は、 $z_0 = f_1^{-1}(f_2^{-1}(f_3^{-1}(f_4^{-1}(f_5^{-1}(x))))))$ として得られることになる。

10

【 0 0 6 7 】

上記 5 個の変換 $\{f_i(z)\}_{i=1}^5$ を次式により定義する。なお、便宜上変換 f_i を示す代わりにその逆変換 f_i^{-1} を示す（ただし、 $z_4 = f_5^{-1}(x)$, $z_3 = f_4^{-1}(z_4)$, $z_2 = f_3^{-1}(z_3)$, $z_1 = f_2^{-1}(z_2)$, $z_0 = f_1^{-1}(z_1)$ ）。

【 0 0 6 8 】

【 数 9 】

$$f_5^{-1}(x) = L_5 D_5 U_5 x \quad \dots(8a)$$

20

$$f_4^{-1}(z_4) = AdaBN_{\gamma_4 \beta_4}(z_4) \quad \dots(8b)$$

$$f_3^{-1}(z_3) = LeakyReLU(z_3) = \max(z_3, \alpha_3 z_3) \quad \dots(8c)$$

$$f_2^{-1}(z_2) = L_2 D_2 U_2 z_2 \quad \dots(8d)$$

$$f_1^{-1}(z_1) = AdaBN_{\gamma_1 \beta_1}(z_1) \quad \dots(8e)$$

30

【 0 0 6 9 】

ここで、 $L_2, L_5 \in \mathbb{R}^{D \times D}$ は、対角成分が 1 の下三角行列であり、上三角部分を除くすべての要素 $L_{2,ij}, L_{5,ij} (i < j)$ が学習対象パラメータ（つまり、パラメータ θ_2 またはパラメータ θ_5 ）である。 $D_2, D_5 \in \mathbb{R}^{D \times D}$ は、対角行列であり、対角成分 $D_{2,ij}, D_{5,ij} (i=j)$ が学習対象パラメータ（つまり、パラメータ θ_2 またはパラメータ θ_5 ）である。 $U_2, U_5 \in \mathbb{R}^{D \times D}$ は、対角成分が 1 の上三角行列であり、下三角部分を除くすべての要素 $U_{2,ij}, U_{5,ij} (i < j)$ が学習対象パラメータ（つまり、パラメータ θ_2 またはパラメータ θ_5 ）である。また、 α_3 (0) は LeakyReLU のパラメータであり、超パラメータとして設定してもよいし、学習対象パラメータ（つまり、パラメータ θ_3 ）としてもよい（ α_3 を学習対象とする場合は、Parametric ReLU と呼ばれる（参考非特許文献 6））。また、 $AdaBN_{\gamma_4 \beta_4}(\cdot)$ 及び $AdaBN_{\gamma_1 \beta_1}(\cdot)$ は上述の適応バッチ正規化であり、 $\gamma_1, \beta_1, \gamma_4, \beta_4$ が学習対象パラメータ（つまり、パラメータ θ_1 またはパラメータ θ_4 ）である。

40

（参考非特許文献 6 : K. He, et al., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", ICCV 2015, pp.1026-1034, 2015.）

また、変換 $\{f_i(z)\}_{i=1}^5$ のヤコビアン行列式の絶対値は、それぞれ次式により計算される（ただし、 $x = f_5(z_4)$, $z_4 = f_4(z_3)$, $z_3 = f_3(z_2)$, $z_2 = f_2(z_1)$, $z_1 = f_1(z_0)$ ）。

【 0 0 7 0 】

【 数 1 0 】

50

$$|\det(\partial f_5(z_4)/\partial z_4)| = 1/|\prod_{i=1}^D D_{5,ii}| \quad \dots(9a)$$

$$|\det(\partial f_4(z_3)/\partial z_3)| = \sqrt{s_4'^2 + \varepsilon}/\gamma_4 \quad \dots(9b)$$

$$|\det(\partial f_3(z_2)/\partial z_2)| = 1/\alpha_3^\delta \quad \dots(9c)$$

$$|\det(\partial f_2(z_1)/\partial z_1)| = 1/|\prod_{i=1}^D D_{2,ii}| \quad \dots(9d)$$

$$|\det(\partial f_1(z_0)/\partial z_0)| = \sqrt{s_1'^2 + \varepsilon}/\gamma_1 \quad \dots(9e)$$

10

【 0 0 7 1 】

ここで、 s_4' は（学習データから生成された入力データ x に対応する） z_4 の標準偏差、 γ_4 は（学習データから生成された入力データ x に対応する） z_3 のうちゼロを下回る要素の数、 s_1' は（学習データから生成された入力データ x に対応する） z_0 の標準偏差である。なお、変換 f_4, f_1 に対するヤコビアン行列式の絶対値 $|\det(\partial f_4(z_3)/\partial z_3)|, |\det(\partial f_1(z_0)/\partial z_0)|$ に関しては、学習時ではなく推論時（つまり、学習済みモデルを用いた処理時）におけるヤコビアンの行列式の絶対値を用いて表記している。

【 0 0 7 2 】

また、先述の通り、確率分布 $q_0(z_0)$ は厳密な確率密度推定が容易な確率分布とする。例えば、確率分布 $q_0(z_0)$ を平均が0、分散が単位行列 I のガウス分布 $N(0, I)$ とすると、確率分布 $q_0(z_0)$ は次式により表される。

【 0 0 7 3 】

【数 1 1】

$$q_0(z_0) = -(2\pi)^{-D/2} \exp(-\|z_0\|_2^2) \quad \dots(10)$$

20

【 0 0 7 4 】

したがって、 $q_1(x_i; \theta)$ を学習データから生成された入力データ x の確率分布とし、式(5)を用いることにより、潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ から入力データ x_i の確率密度 $q_1(x_i; \theta)$ が計算できることがわかる。

30

【 0 0 7 5 】

続いて、パラメータ θ の学習方法について説明する。通常のニューラルネットワークの学習と同様、損失関数 $L(\theta)$ を用いて、例えば、勾配降下法、Momentum SGD(Stochastic Gradient Descent)、ADAM(Adaptive Moment Estimation)、あるいはそれらの組合せにより学習することができる。ノーマライジング・フローを用いる場合、損失関数 $L(\theta)$ として、次式で定義される負の対数尤度の平均を用いることが多い。

【 0 0 7 6 】

【数 1 2】

$$L(\theta) = -1/N \sum_{i=1}^N \log q_1(x_i; \theta) \quad \dots(11)$$

40

【 0 0 7 7 】

なお、上記学習には、ミニバッチと呼ばれる学習データの集合を単位にして行うミニバッチ学習法を用いることができる。ここで、ミニバッチとは、すべての学習データからランダムに選択した複数の学習データのことであり、このミニバッチごとに損失関数 $L(\theta)$ の値を計算することになる。

【 0 0 7 8 】

(2) 適応学習フェーズ

50

次に、異常検知対象機器から発せられた正常音（つまり、適応学習データ）から生成される入力データ x の確率分布 $q_2(x; \cdot)$ の適応学習方法について説明する。例えば、 $z_4=f_5^{-1}(x)$, $z_3=f_4^{-1}(z_4)$, $z_2=f_3^{-1}(z_3)$, $z_1=f_2^{-1}(z_2)$, $z_0=f_1^{-1}(z_1)$ を利用して、以下の手順で学習を実行すればよい。まず、入力データの集合 $\{x_i\}_{i=1}^M$ から $\{z_{4,i}\}_{i=1}^M$ ($z_{4,i}=f_5^{-1}(x_i)$)を計算する。次に、 $\{z_{4,i}\}_{i=1}^M$ の平均、分散を求める。最後に、式(7c)の m, σ^2 を求めた平均と分散で置換する。同様に、入力データの集合 $\{x_i\}_{i=1}^M$ から $\{z_{1,i}\}_{i=1}^M$ ($z_{1,i}=f_2^{-1}(f_3^{-1}(f_4^{-1}(f_5^{-1}(x_i))))$)を計算する。次に、 $\{z_{1,i}\}_{i=1}^M$ の平均、分散を求める。最後に、式(7c)の m, σ^2 を求めた平均と分散で置換する。

【0079】

(3) テストフェーズ

異常検知の方法には、例えば、《教師なし異常音検知》で説明した方法を用いることができる。

【0080】

《効果》

ノーマライジング・フローに適応バッチ正規化を導入することにより、以下の効果が得られる。

(1) 学習データの分布とテストデータの分布の相違を調整することが可能となり、テスト用ドメインにおける異常検知の精度低下を抑えることが可能となる。

【0081】

具体的に説明する。仮に最終層に適応バッチ正規化を用いた場合、ドメイン間のデータの分布の軸方向の平行移動とスケールのずれを是正することが可能となる。また、異なるドメイン間のデータの分布の相違が平行移動と軸方向のスケーリングによって表現される場合は、原理的には1層目に適応バッチ正規化を導入すれば、学習用ドメインにおいて学習済みのパラメータをテスト用ドメインにそのまま適用してもテスト用ドメインでの異常検知が精度よく実行できる。

(2) 第2確率分布を低計算量で適応学習することが可能となる。

【0082】

第2確率分布の適応学習に必要な処理は、上述の通り、基本的には、適応学習データに対して、適応バッチ正規化における統計量を計算するだけである。したがって、適応学習は、通常の学習を繰り返す場合と比べると低い計算量で実行可能であり、場合によっては、オンラインで実行することも可能となる。

【0083】

また、上述の例のように、ノーマライジング・フローに線形変換を導入している場合、当該線形変換に対応する行列 W をLU分解またはLDU分解しておくことにより、以下の効果が得られる。

(3) 確率密度の計算コストが抑制され、学習コストが抑制される。

【0084】

具体的に説明する。ノーマライジング・フローでは、線形変換 f のヤコビアン行列式を逐一計算する必要がある。そのため、行列 W をLU分解またはLDU分解しない形で保持している場合は、行列式 $|W|$ の計算は W の次数を k として $O(k^3)$ の計算量がかかることになる。

しかし、 $W=LU$ または $W=LDU$ のようにLU分解またはLDU分解した形で保持している場合は、 $|W|=|LU|=|L| \times |U|$ （つまり、 L の対角成分と U の対角成分のすべての積）または $|W|=|LDU|=|L| \times |D| \times |U|$ （つまり、 L の対角成分と D の対角成分と U の対角成分のすべての積）により $|W|$ を求めることができるため、非常に高速に計算することができる。

【0085】

<第1実施形態>

同種の機器が2台以上ある状況において、その中の1台の機器（これを異常検知対象機器とする）の異常を検知することを考える。そのために、まずこの異常検知対象機器とは異なる1以上の機器から発せられた正常音（以下、学習用正常音という）から、異常検知対象機器とは異なる1以上の機器から発せられた正常音の分布を示す確率分布（以下、第

10

20

30

40

50

1 確率分布という)を学習する(学習フェーズ)。次に、異常検知対象機器から発せられた正常音(以下、適応学習用正常音という)を用いて、第1確率分布から、異常検知対象機器から発せられた正常音の分布を示す確率分布(以下、第2確率分布という)を適応学習する(適応学習フェーズ)。そして、異常検知対象機器から発せられた音(以下、異常検知対象音という)から、当該機器が異常であるか否かを判定する(テストフェーズ(異常検知フェーズ))。

【0086】

確率分布学習装置100は、学習用正常音から、第1確率分布を学習する。確率分布適応学習装置200は、適応学習用正常音を用いて、第1確率分布から第2確率分布を適応学習する。異常検知装置300は、異常検知対象音から、当該機器が異常であるか否かを判定する。

10

【0087】

以下、図1～図2を参照して確率分布学習装置100を説明する。図1は、確率分布学習装置100の構成を示すブロック図である。図2は、確率分布学習装置100の動作を示すフローチャートである。図1に示すように確率分布学習装置100は、入力データ生成部110と、潜在変数推定部120と、損失関数計算部130と、パラメータ更新部140と、収束条件判定部150と、記録部190を含む。記録部190は、確率分布学習装置100の処理に必要な情報を適宜記録する構成部である。記録部190は、例えば、学習対象となる第1確率分布 $q_1(x; \theta)$ のパラメータ θ を記録する。パラメータ θ の初期値として、例えば乱数を用いて生成された値が記録されている。

20

【0088】

確率分布学習装置100は、学習用正常音記録部910に接続している。学習用正常音記録部910には、事前に用意した学習用正常音が学習データとして記録されている。先述した通り、学習用正常音はできるだけ多く用意しておくことが好ましい。

【0089】

図2に従い確率分布学習装置100の動作について説明する。入力データ生成部110は、入力である学習用正常音 $s_i(i=1, \dots, N)$ から、入力データ $x_i(i=1, \dots, N)$ を生成する(S110)。例えば、式(1)のように、学習用正常音 s_i の対数振幅スペクトルを要素とするベクトルを生成し、これを入力データ x_i とすればよい。なお、学習用正常音 s_i から生成される入力データ x_i として、上記以外の音響特徴量を用いるようにしてもよい。

30

【0090】

潜在変数推定部120は、第1確率分布 $q_1(x; \theta)$ のパラメータ θ を用いて、S110で生成した入力データ $x_i(i=1, \dots, N)$ から、入力データ x_i に対応する潜在変数 $z_{0,i}(i=1, \dots, N)$ を推定する(S120)。なお、ここで用いるパラメータ θ は、学習中の値である。

【0091】

ここで、学習対象となる第1確率分布 $q_1(x; \theta)$ の変数 x は、異常検知対象機器とは異なる1以上の機器から発せられた正常音から生成された入力データを示す変数であり、変数 x は、変換 $f_i(i=1, \dots, K, Kは1以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する)$ と潜在変数 z_0 を用いて、 $x=f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表されるものとする。

【0092】

したがって、入力データ x_i に対応する潜在変数 $z_{0,i}$ は次式により与えられる。

40

【0093】

【数13】

$$z_{0,i} = f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x_i))\dots)) \quad \dots(12)$$

【0094】

また、潜在変数 $z_{0,i}(i=1, \dots, N)$ は、潜在変数 z_0 の確率分布 $q_0(z_0)$ に従って生成されるものとする。ただし、確率分布 $q_0(z_0)$ は厳密な確率密度推定が容易であるという性質を有する。

50

【 0 0 9 5 】

したがって、確率分布 $q_1(x; \theta)$ は、確率分布 $q_0(z_0)$ を用いた次式により表現できる（式(5)参照）。

【 0 0 9 6 】

【数 1 4】

$$q_1(x; \theta) = q_0(z_0) \left| \det(\partial f_K(z_{K-1}; \theta_K) / \partial z_{K-1}) \right| \left| \det(\partial f_{K-1}(z_{K-2}; \theta_{K-1}) / \partial z_{K-2}) \right| \cdots \left| \det(\partial f_1(z_0; \theta_1) / \partial z_0) \right| \cdots (5)'$$

【 0 0 9 7 】

ここで、 θ_i は変換 f_i に対応するパラメータであり、 $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_K^T]^T$ となる。

【 0 0 9 8 】

潜在変数推定部 1 2 0 は、入力データ x から潜在変数 z_0 を計算するニューラルネットワークNNを用いて実現できる。なお、計算式は以下の通りである。

【 0 0 9 9 】

【数 1 5】

$$z_0 = f_1^{-1}(f_2^{-1}(\cdots(f_K^{-1}(x))\cdots)) \cdots (12)'$$

【 0 1 0 0 】

つまり、ニューラルネットワークNNは、入力データ x を入力とし、逆変換 $f_K^{-1}, f_{K-1}^{-1}, \dots, f_2^{-1}, f_1^{-1}$ の順に計算を進めていき、最終的に潜在変数 z_0 を出力するものである。

【 0 1 0 1 】

上記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも1つの変換 f_{i_0} (ただし、 i_0 は $1 \leq i_0 \leq K$ を満たす整数)に対して、その逆変換 $f_{i_0}^{-1}$ は適応バッチ正規化となるようにする。このようにすることで、比較的少量の適応学習正常音を用いて第2確率分布 $q_2(x; \theta)$ を適応学習できるようにする（《ポイント》参照）。

【 0 1 0 2 】

また、変換 $f_i(i=1, \dots, K)$ のうち、少なくとも1つの変換 f_{i_1} (ただし、 i_1 は $1 \leq i_1 \leq K$ を満たす整数)に対して、その逆変換 $f_{i_1}^{-1}$ は、線形変換であり、当該線形変換に対応する行列が、下方三角行列と上方三角行列の積、または、下方三角行列と対角行列と上方三角行列の積として表されているようにしてもよい。このようにすることで、第1確率分布 $q_1(x; \theta)$ の学習時に必要となる確率密度の計算（後述する損失関数計算部 1 3 0における計算）を低コストで実行できるようになる（《効果》の(3)参照）。

【 0 1 0 3 】

具体的な変換 $f_i(i=1, \dots, K)$ の例として、例えば $K=5$ として、逆変換が式(8a)～式(8e)で表される5つの変換を用いることができる。

【 0 1 0 4 】

損失関数計算部 1 3 0 は、S 1 2 0で推定した潜在変数 $z_{0,i}(i=1, \dots, N)$ から、第1確率分布 $q_1(x; \theta)$ のパラメータの最適化に用いる損失関数 $L(\theta)$ の値を計算する（S 1 3 0）。損失関数 $L(\theta)$ は、例えば、式(11)のように負の対数尤度の平均として定義することができる。その際、入力データ $x_i(i=1, \dots, N)$ の確率密度 $q_1(x_i; \theta)$ を計算する必要があるが、入力データ x_i の確率密度 $q_1(x_i; \theta)$ は入力データ x_i に対応する潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ を用いて計算することができる。例えば、確率分布 $q_0(z_0)$ がガウス分布 $N(0, I)$ である場合、潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ は、次式で計算できるので、

【 0 1 0 5 】

【数 1 6】

$$q_0(z_{0,i}) = -(2\pi)^{-D/2} \exp(-\|z_{0,i}\|_2^2) \cdots (10)'$$

10

20

30

40

50

【0106】

式(5)'を用いて、上記計算した潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ から入力データ x_i の確率密度 $q_1(x_i; \theta)$ を計算することができる。

【0107】

パラメータ更新部140は、S130で計算した損失関数 $L(\theta)$ の値を最適化(最小化)するように、第1確率分布 $q_1(x; \theta)$ のパラメータ θ を更新する(S140)。パラメータの更新には、例えば、勾配降下法を用いるとよい。

【0108】

収束条件判定部150は、パラメータ更新の終了条件として事前に設定された収束条件を判定し、収束条件が満たされた場合はS140で更新したパラメータ θ を用いた第1確率分布 $q_1(x; \theta)$ を出力し、収束条件が満たされない場合はS110~S140の処理を繰り返す(S150)。収束条件として、例えばS110~S140の処理の実行回数が所定の回数に達したかという条件を採用することができる。出力に際して、S140で更新したパラメータ θ (このパラメータのことを学習済みパラメータともいう)、逆変換 $f_{K-1}^{-1}(x; \theta)$, $f_{K-1}^{-1}(z_{K-1}; \theta_{K-1})$, ..., $f_2^{-1}(z_2; \theta_2)$, $f_1^{-1}(z_1; \theta_1)$ を出力するようにしてもよい。

【0109】

なお、入力データ生成部110、潜在変数推定部120、損失関数計算部130、パラメータ更新部140、収束条件判定部150を含む構成部を学習部105という。つまり、学習部105は、学習用正常音から第1確率分布を学習する構成部である。

【0110】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を容易に調整することを可能とする第1確率分布 $q_1(x; \theta)$ を学習することができる。

【0111】

<変形例>

確率分布学習装置100は、異常検知対象機器とは異なる1以上の機器から発せられた正常音である学習用正常音から、1つの第1確率分布 $q_1(x; \theta)$ を学習するものとして説明したが、異常検知対象機器とは異なる1以上の機器の数を W (W は1以上の整数)とし、 W 個の機器の各々から発せられた正常音の分布を示す第1確率分布 $q_1^{(1)}(x; \theta_1)$, ..., $q_1^{(W)}(x; \theta_W)$ を学習するようにしてもよい。この場合、(ミニバッチに相当する)入力データ x_i ($i=1, \dots, N$)の中で同一の機器から発せられた学習用正常音から生成した入力データごとに適応バッチ正規化における平均と分散を計算することにより、潜在変数推定部120は潜在変数の推定処理を実行する。つまり、適応バッチ正規化における計算において、1組の平均と分散を用いる代わりに、 W 組の平均と分散を用いることになる。ただし、学習済みパラメータ θ は、 W 個の第1確率分布 $q_1^{(1)}(x; \theta_1)$, ..., $q_1^{(W)}(x; \theta_W)$ で共通となるため、1組である。

【0112】

<第2実施形態>

以下、図3~図4を参照して確率分布適応学習装置200を説明する。図3は、確率分布適応学習装置200の構成を示すブロック図である。図4は、確率分布適応学習装置200の動作を示すフローチャートである。図3に示すように確率分布適応学習装置200は、入力データ生成部110と、パラメータ更新部240と、出力部250と、記録部190を含む。記録部190は、確率分布適応学習装置200の処理に必要な情報を適宜記録する構成部である。記録部190は、例えば、確率分布学習装置100を用いて学習した第1確率分布 $q_1(x; \theta)$ のパラメータ θ (つまり、学習済みパラメータ)を記録しておく。この学習済みパラメータが、第2確率分布 $q_2(x; \theta)$ のパラメータ θ の初期値となるものである。なお、第2確率分布 $q_2(x; \theta)$ の変数 x は、異常検知対象機器から発せられた正常音から生成された入力データを示す変数である。

【0113】

確率分布適応学習装置200は、適応学習用正常音記録部920に接続している。適応学習用正常音記録部920には、事前に用意した適応学習用正常音が適応学習データとし

10

20

30

40

50

て記録されている。先述した通り、適応学習用正常音は、学習用正常音に比べると、比較的少量用意しておくのでよい。

【0114】

図4に従い確率分布適応学習装置200の動作について説明する。入力データ生成部110は、入力である適応学習用正常音 $s_i(i=1, \dots, M)$ から、入力データ $x_i(i=1, \dots, M)$ を生成する(S110)。ここで、適応学習用正常音の数 M は、基本的には学習用正常音の数 N を超えない整数であるが、超える整数であってもかまわない。

【0115】

パラメータ更新部240は、S110で生成した入力データ $x_i(i=1, \dots, M)$ を用いて、第2確率分布 $q_2(x; \cdot)$ のパラメータを更新する(S240)。具体的には、逆変換が適応バッチ正規化である変換 f_{i_0} に対して、その計算に用いる平均と分散を入力データ $x_i(i=1, \dots, M)$ から計算される逆変換 $f_{i_0}^{-1}$ への入力データ $z_{i_0,i}(=f_{i_0}^{-1}(f_{i_0}^{-2}(\dots(f_{i_0}^{-1}(x_i))\dots)))$ の平均と分散で更新すればよい。つまり、式(7c)の m と s^2 を入力データ $z_{i_0,i}(i=1, \dots, M)$ の平均と分散で置換する。

【0116】

出力部250は、S240で更新したパラメータを用いた第2確率分布 $q_2(x; \cdot)$ を出力する(S250)。また、出力に際して、S240で更新したパラメータ(このパラメータのことを学習済みパラメータともいう)、逆変換 $f_K^{-1}(x; \kappa)$, $f_{K-1}^{-1}(z_{K-1}; \kappa-1)$, \dots , $f_2^{-1}(z_2; \kappa-2)$, $f_1^{-1}(z_1; \kappa-1)$ を出力するようにしてもよい。

【0117】

なお、入力データ生成部110、パラメータ更新部240、出力部250を含む構成部を適応学習部205という。つまり、適応学習部205は、適応学習用正常音を用いて第1確率分布から第2確率分布を適応学習する構成部である。

【0118】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を調整することにより、第2確率分布 $q_2(x; \cdot)$ を学習することができる。

【0119】

<変形例>

確率分布学習装置100が W 個の第1確率分布 $q_1^{(1)}(x; \cdot)$, \dots , $q_1^{(W)}(x; \cdot)$ を学習する場合においても、学習済みパラメータは1組しかない。確率分布適応学習装置200はこの1組のパラメータを用いて適応学習をする。

【0120】

<第3実施形態>

以下、図5~図6を参照して異常検知装置300を説明する。図5は、異常検知装置300の構成を示すブロック図である。図6は、異常検知装置300の動作を示すフローチャートである。図5に示すように異常検知装置300は、入力データ生成部110と、異常度推定部320と、異常判定部330と、記録部390を含む。記録部390は、異常検知装置300の処理に必要な情報を適宜記録する構成部である。記録部390は、例えば、確率分布適応学習装置200を用いて学習した第2確率分布 $q_2(x; \cdot)$ のパラメータ(つまり、学習済みパラメータ)を記録しておく。

【0121】

つまり、この学習済みパラメータを用いた第2確率分布 $q_2(x; \cdot)$ を学習済みの第2確率分布として、異常検知装置300は異常検知を実行する。

【0122】

図6に従い異常検知装置300の動作について説明する。入力データ生成部110は、入力である異常検知対象音 s から、入力データ x を生成する(S110)。

【0123】

異常度推定部320は、学習済みの第2確率分布 $q_2(x; \cdot)$ に基づいて、S110で生成した入力データ x から、機器の異常の程度を示す異常度を推定する(S320)。以下、図7~図8を参照して異常度推定部320について説明する。図7は、異常度推定部320

10

20

30

40

50

の構成を示すブロック図である。図 8 は、異常度推定部 3 2 0 の動作を示すフローチャートである。図 7 に示すように異常度推定部 3 2 0 は、潜在変数計算部 3 2 1 と、異常度計算部 3 2 2 を含む。

【 0 1 2 4 】

図 8 に従い異常度推定部 3 2 0 の動作について説明する。潜在変数計算部 3 2 1 は、S 1 1 0 で生成した入力データ x から、入力データ x に対応する潜在変数 z_0 を計算する (S 3 2 1)。具体的には、ニューラルネットワーク NN のパラメータを第 2 確率分布の学習済みパラメータとしたニューラルネットワークを用いて計算することができる。

【 0 1 2 5 】

異常度計算部 3 2 2 は、S 3 2 1 で計算した潜在変数 z_0 から、入力データ x に関する異常度 $A(x; \theta)$ を計算する (S 3 2 2)。異常度は、例えば次式により計算できる。

【 0 1 2 6 】

【数 1 7】

$$A(x; \theta) = -\ln q_2(x; \theta) \quad \dots(2)'$$

【 0 1 2 7 】

確率分布 $q_0(z_0)$ が、厳密な確率密度推定が容易な分布である場合、入力データ x の確率密度 $q_2(x; \theta)$ は、次式により計算することができる。

【 0 1 2 8 】

【数 1 8】

$$q_2(x; \theta) = q_0(z_0) |\det(\partial f_K(z_{K-1}; \theta_K) / \partial z_{K-1})| \\ |\det(\partial f_{K-1}(z_{K-2}; \theta_{K-1}) / \partial z_{K-2})| \dots |\det(\partial f_1(z_0; \theta_1) / \partial z_0)| \dots(5)''$$

【 0 1 2 9 】

式(5)'' の計算に際して、例えば、確率分布 q_0 がガウス分布 $N(0, I)$ 、逆変換が式(8a) ~ 式(8e) で表される 5 つの変換である場合は、式(10) や式(9a) ~ 式(9e) を用いればよい。

【 0 1 3 0 】

異常度判定部 3 3 0 は、S 3 2 0 で推定した異常度 $A(x; \theta)$ から、機器が異常であるか否かを示す判定結果を生成する (S 3 3 0)。例えば、式(3)を用いて、 $R=1$ の場合は異常を示す判定結果、 $R=0$ の場合は正常を示す判定結果を生成すればよい。

【 0 1 3 1 】

つまり、異常検知装置 3 0 0 は、異常検知対象機器とは異なる 1 以上の機器から発せられた正常音の分布を示す第 1 確率分布と異常検知対象機器から発せられた正常音 (適応学習用正常音) との関連付けに基づいて、異常検知対象機器から発せられた音 (異常検知対象音) から、当該機器の異常の程度を示す異常度を推定する異常度推定部 3 2 0 とを含むものであると言える。関連付けの一例は、適応学習用正常音を用いて第 1 確率分布を更新することにより得た、異常検知対象機器から発せられた正常音の分布を示す第 2 確率分布である。

【 0 1 3 2 】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を調整した第 2 確率分布を用いて異常検知することにより、精度のよい異常検知が可能となる。つまり、異常検知精度の低下を抑えることが可能となる。

【 0 1 3 3 】

なお、第 1 実施形態から第 3 実施形態では、ノーマライジング・フローを用いて異常度 $A(x; \theta)$ を計算する方法を説明してきたが、その他の統計モデルを用いて異常度を求めることもできる。例えば、参考非特許文献 7 に記載の自己符号化器 (AE: autoencoder) を用いることもできる。自己符号化器とは、符号化器 (エンコーダ) と復号器 (デコーダ) の組である。

(参考非特許文献 7 : Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harad

10

20

30

40

50

a, “Unsupervised Detection of Anomalous Sound based on Deep Learning and the Neyman-Pearson Lemma,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol.27-1, pp.212-224, 2019.)

以下、本発明の第4実施形態から第6実施形態に対する技術的背景と、各実施形態について説明する。

【0134】

<技術的背景>

自己符号化器を利用する場合、異常度は次式により計算できる。

【0135】

【数19】

$$A(x; \theta) = \|x - D(E(x, \theta_E), \theta_D)\|^2 \quad \dots(13)$$

10

【0136】

ここで、 $\|\cdot\|_2$ はユークリッドノルム、EとDはそれぞれ符号化器と復号器、 θ_E と θ_D はそれぞれ符号化器Eのパラメータと復号器Dのパラメータを表す。つまり、 $\theta = \{\theta_E, \theta_D\}$ である。

【0137】

符号化器E、復号器Dはいずれもニューラルネットワークとして構成できる。この場合、例えば、 θ は正常音の学習データの再構成（復元）誤差を最小化するように学習する。

【0138】

【数20】

$$\theta \leftarrow \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N A(x_i; \theta) \quad \dots(14)$$

20

【0139】

ここで、 x_i はi番目の正常音の学習データ、Nは正常音の学習データのサンプル数である。

【0140】

自己符号化器を用いた適応学習を実現するためには、適応バッチ正規化(AdaBN)を用いたニューラルネットワークとして、符号化器Eと復号器Dの両方、もしくはそのどちらか一方を構成すればよい。つまり、符号化器Eと復号器Dの両方、もしくはそのどちらか一方の計算の途中に適応バッチ正規化を利用する。例えば、符号化器Eを $E(x; \theta_E) = W_2[\sigma(W_1x + b_1)] + b_2$ という3層ニューラルネットワークとして構成する代わりに、符号化器EをAdaBN層を挿入した次式を計算するニューラルネットワークとして構成すればよい。

30

【0141】

【数21】

$$E(x, \theta_E) = W_2 \{AdaBN[\sigma(W_1x + b_1)]\} + b_2 \quad \dots(15)$$

【0142】

ここで、 W_1 、 W_2 は重み行列、 b_1 、 b_2 はバイアスベクトル、 σ は活性化関数を表す。AdaBN層とは、例えば式(8b)や式(8e)のように、AdaBN（適応バッチ正規化）の計算を実行する層のことである。

40

【0143】

<第4実施形態>

同種の機器が2台以上ある状況において、その中の1台の機器（これを異常検知対象機器とする）の異常を検知することを考える。そのために、まずこの異常検知対象機器とは異なる1以上の機器から発せられた正常音（以下、学習用正常音という）から、異常検知対象機器とは異なる1以上の機器から発せられた正常音を復元する自己符号化器（以下、第1自己符号化器という）を学習する（学習フェーズ）。次に、異常検知対象機器から発

50

せられた正常音（以下、適応学習用正常音という）を用いて、第1自己符号化器から、異常検知対象機器から発せられた正常音を復元する自己符号化器（以下、第2自己符号化器という）を適応学習する（適応学習フェーズ）。そして、異常検知対象機器から発せられた音（以下、異常検知対象音という）から、当該機器が異常であるか否かを判定する（テストフェーズ（異常検知フェーズ））。

【0144】

自己符号化器学習装置400は、学習用正常音から、第1自己符号化器を学習する。自己符号化器適応学習装置500は、適応学習用正常音を用いて、第1自己符号化器から第2自己符号化器を適応学習する。異常検知装置600は、異常検知対象音から、当該機器が異常であるか否かを判定する。

10

【0145】

以下、図9～図10を参照して自己符号化器学習装置400を説明する。図9は、自己符号化器学習装置400の構成を示すブロック図である。図10は、自己符号化器学習装置400の動作を示すフローチャートである。図9に示すように自己符号化器学習装置400は、入力データ生成部110と、復元入力データ推定部420と、損失関数計算部430と、パラメータ更新部440と、収束条件判定部450と、記録部490を含む。記録部490は、自己符号化器学習装置400の処理に必要な情報を適宜記録する構成部である。記録部490は、例えば、学習対象となる第1自己符号化器のパラメータを記録する。パラメータの初期値として、例えば乱数を用いて生成された値が記録されている。

【0146】

自己符号化器学習装置400は、学習用正常音記録部910に接続している。学習用正常音記録部910には、事前に用意した学習用正常音が学習データとして記録されている。先述した通り、学習用正常音はできるだけ多く用意しておくことが好ましい。

20

【0147】

図10に従い自己符号化器学習装置400の動作について説明する。入力データ生成部110は、入力である学習用正常音 $s_i (i=1, \dots, N)$ から、入力データ $x_i (i=1, \dots, N)$ を生成する(S110)。

【0148】

復元入力データ推定部420は、第1自己符号化器のパラメータを用いて、S110で生成した入力データ $x_i (i=1, \dots, N)$ から、入力データ x_i に対応する復元入力データ $y_i (i=1, \dots, N)$ を推定する(S420)。なお、ここで用いるパラメータは、学習中の値である。

30

【0149】

復元入力データ推定部420は、入力データ x_i から復元入力データ y_i を計算するニューラルネットワークを用いて実現できる。なお、計算式は以下の通りである。

【0150】

【数22】

$$y_i = D(E(x_i, \theta_E), \theta_D) \dots (16)$$

【0151】

このニューラルネットワークが第1自己符号化器（ニューラルネットワークNNという）である。

40

【0152】

ここで、 $\theta = \{ \theta_E, \theta_D \}$ （ただし、 θ_E と θ_D はそれぞれ符号化器Eのパラメータと復号器Dのパラメータを表す）である。また、符号化器Eを構成するニューラルネットワーク、復号器Dを構成するニューラルネットワークの少なくとも1つは、AdaBN層を含むものとする。AdaBN層とは、AdaBN（適応バッチ正規化）の計算を実行する層のことである。つまり、ニューラルネットワークNNは、AdaBN層を含む。

【0153】

損失関数計算部430は、S420で推定した復元入力データ $y_i (i=1, \dots, N)$ から、第1自己符号化器のパラメータの最適化に用いる損失関数 $L(\theta)$ の値を計算する(S430)

50

。損失関数 $L(\theta)$ は、例えば、次式により定義される異常度の平均とすることができる。

【0154】

【数23】

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N A(x_i; \theta) \quad \dots(17)$$

【0155】

パラメータ更新部440は、S430で計算した損失関数 $L(\theta)$ の値を最適化(最小化)するように、第1自己符号化器のパラメータを更新する(S440)。パラメータの更新には、例えば、勾配降下法を用いるとよい。

10

【0156】

収束条件判定部450は、パラメータ更新の終了条件として事前に設定された収束条件を判定し、収束条件が満たされた場合はS440で更新したパラメータを出力し、収束条件が満たされない場合はS110~S440の処理を繰り返す(S450)。収束条件として、例えばS110~S440の処理の実行回数が所定の回数に達したかという条件を採用することができる。

【0157】

なお、入力データ生成部110、復元入力データ推定部420、損失関数計算部430、パラメータ更新部440、収束条件判定部450を含む構成部を学習部405という。つまり、学習部405は、学習用正常音から第1自己符号化器(のパラメータ)を学習する構成部である。

20

【0158】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を容易に調整することを可能とする第1自己符号化器を学習することができる。

【0159】

<第5実施形態>

以下、図11~図12を参照して自己符号化器適応学習装置500を説明する。図11は、自己符号化器適応学習装置500の構成を示すブロック図である。図12は、自己符号化器適応学習装置500の動作を示すフローチャートである。図11に示すように自己符号化器適応学習装置500は、入力データ生成部110と、パラメータ更新部540と、出力部550と、記録部490を含む。記録部490は、自己符号化器適応学習装置500の処理に必要な情報を適宜記録する構成部である。記録部490は、例えば、自己符号化器学習装置400を用いて学習した第1自己符号化器のパラメータ(つまり、学習済みパラメータ)を記録しておく。この学習済みパラメータが、第2自己符号化器のパラメータの初期値となるものである。

30

【0160】

自己符号化器適応学習装置500は、適応学習用正常音記録部920に接続している。適応学習用正常音記録部920には、事前に用意した適応学習用正常音が適応学習データとして記録されている。先述した通り、適応学習用正常音は、学習用正常音に比べると、比較的少量用意しておくのでよい。

40

【0161】

図12に従い自己符号化器適応学習装置500の動作について説明する。入力データ生成部110は、入力である適応学習用正常音 $s_i(i=1, \dots, M)$ から、入力データ $x_i(i=1, \dots, M)$ を生成する(S110)。ここで、適応学習用正常音の数 M は、基本的には学習用正常音の数 N を超えない整数であるが、超える整数であってもかまわない。

【0162】

パラメータ更新部540は、S110で生成した入力データ $x_i(i=1, \dots, M)$ を用いて、第2自己符号化器のパラメータを更新する(S540)。具体的には、第1自己符号化器(ニューラルネットワークNN)に含まれる、適応バッチ正規化を計算する層であるAda

50

BN層に対して、その計算に用いる平均と分散を入力データ x_i ($i=1, \dots, M$)から計算される復元入力データ y_i の平均と分散で更新すればよい。

【0163】

出力部550は、S540で更新したパラメータ θ を出力する (S550)。

【0164】

なお、入力データ生成部110、パラメータ更新部540、出力部550を含む構成部を適応学習部505という。つまり、適応学習部505は、適応学習用正常音を用いて第1自己符号化器から第2自己符号化器を適応学習する構成部である。

【0165】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を調整することにより、第2自己符号化器を学習することができる。

10

【0166】

<第6実施形態>

以下、図13~図14を参照して異常検知装置600を説明する。図13は、異常検知装置600の構成を示すブロック図である。図14は、異常検知装置600の動作を示すフローチャートである。図13に示すように異常検知装置600は、入力データ生成部110と、異常度推定部620と、異常判定部630と、記録部690を含む。記録部690は、異常検知装置600の処理に必要な情報を適宜記録する構成部である。記録部690は、例えば、自己符号化器適応学習装置500を用いて学習した第2自己符号化器のパラメータ θ (つまり、学習済みパラメータ)を記録しておく。

20

【0167】

つまり、この学習済みパラメータを用いた第2自己符号化器を学習済みの第2自己符号化器として、異常検知装置600は異常検知を実行する。

【0168】

図14に従い異常検知装置600の動作について説明する。入力データ生成部110は、入力である異常検知対象音 s から、入力データ x を生成する (S110)。

【0169】

異常度推定部620は、学習済みの第2自己符号化器に基づいて、S110で生成した入力データ x から、機器の異常の程度を示す異常度を推定する (S620)。以下、図15~図16を参照して異常度推定部620について説明する。図15は、異常度推定部620の構成を示すブロック図である。図16は、異常度推定部620の動作を示すフローチャートである。図15に示すように異常度推定部620は、復元入力データ計算部621と、異常度計算部622を含む。

30

【0170】

図16に従い異常度推定部620の動作について説明する。復元入力データ計算部621は、S110で生成した入力データ x から、入力データ x に対応する復元入力データ y を計算する (S621)。具体的には、ニューラルネットワークNNのパラメータを第2自己符号化器の学習済みパラメータとしたニューラルネットワークを用いて計算することができる。

【0171】

異常度計算部622は、S621で計算した復元入力データ y から、入力データ x に関する異常度 $A(x; \theta)$ を計算する (S622)。異常度は、例えば式(13)により計算できる。

40

【0172】

異常度判定部630は、S620で推定した異常度 $A(x; \theta)$ から、機器が異常であるか否かを示す判定結果を生成する (S630)。例えば、式(3)を用いて、 $R=1$ の場合は異常を示す判定結果、 $R=0$ の場合は正常を示す判定結果を生成すればよい。

【0173】

つまり、異常検知装置600は、異常検知対象機器とは異なる1以上の機器から発せられた正常音を復元する第1自己符号化器と異常検知対象機器から発せられた正常音 (適応学習用正常音)との関連付けに基づいて、異常検知対象機器から発せられた音 (異常検知

50

対象音)から、当該機器の異常の程度を示す異常度を推定する異常度推定部620を含むものであると言える。関連付けの一例は、適応学習用正常音を用いて第1自己符号化器を更新することにより得た、異常検知対象機器から発せられた正常音を復元する第2自己符号化器である。

【0174】

本実施形態の発明によれば、学習データの分布とテストデータの分布の相違を調整した第2自己符号化器を用いて異常検知することにより、精度のよい異常検知が可能となる。つまり、異常検知精度の低下を抑えることが可能となる。

【0175】

次に、本発明の第7実施形態から第9実施形態に対する技術的背景と、各実施形態について説明する。

10

【0176】

<技術的背景>

本発明の実施形態は、ノーマライジング・フロー(Normalizing Flow)を用いてペアデータなしドメイン変換のデータ変換器の学習を行う。ノーマライジング・フローは、GANに比べて学習を行いやすいという特徴があり、その結果として従来のGANベースのペアデータなしドメイン変換技術(StarGAN)よりも安定して学習することができる。

【0177】

以下、本発明の実施形態で用いる従来技術について説明する。

《ノーマライジング・フロー(Normalizing Flow)》

20

ノーマライジング・フローは、データ生成に関する確率分布 $p(x)$ を近似する分布を求める手法である。

【0178】

$\{f_i(z)\}_{i=1}^K$ を逆変換を有するK個の変換(ただし、 $f_i(z):R^D \rightarrow R^D$, R は実数の集合、 D は1以上の整数、 K は1以上の整数)とする。また、 $f_i^{-1}(z)$ ($i=1, \dots, K$)を $f_i(z)$ の逆変換とする。

【0179】

ノーマライジング・フローでは、N個の入力データの集合 $\{x_i\}_{i=1}^N$ に対して、それぞれ対応する潜在変数 $\{z_{0,i}\}_{i=1}^N$ が存在し、入力データ x_i は、K個の変換 $\{f_i(z)\}_{i=1}^K$ と x の潜在変数 z_0 を用いた式(21)により、対応する潜在変数 $z_{0,i}$ を変換したものであると考える。

30

【0180】

【数24】

$$x = f_K(f_{K-1}(\dots(f_1(z_0))\dots)) \quad \dots(21)$$

【0181】

つまり、 $i=1, \dots, K$ に対して、次式が成り立つ。

【0182】

【数25】

$$x_i = f_K(f_{K-1}(\dots(f_1(z_{0,i}))\dots)) \quad \dots(21)'$$

40

【0183】

なお、 $z_1=f_1(z_0)$, $z_2=f_2(z_1)$, ..., $x=f_K(z_{K-1})$ と表すこともある。

【0184】

更に、潜在変数 $\{z_{0,i}\}_{i=1}^N$ は、例えば等方ガウス分布のような、モンテカルロサンプリングが容易な確率分布 $q_0(z_0)$ から生成されると仮定する。このとき、入力データの集合 $\{x_i\}_{i=1}^N$ が従う確率分布 $q(x; \cdot)$ (x は入力データを示す変数)は、次の形で表現できる。

【0185】

【数26】

50

$$q(x; \theta) = q_0(z_0) \left| \det(\partial f_K(z_{K-1}; \theta_K) / \partial z_{K-1}) \right| \\ \left| \det(\partial f_{K-1}(z_{K-2}; \theta_{K-1}) / \partial z_{K-2}) \right| \cdots \left| \det(\partial f_1(z_0; \theta_1) / \partial z_0) \right| \cdots (22)$$

【0186】

ここで、 $z_0 = f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x))\dots))$ である。また、 $\{f_i\}_{i=1}^K$ は変換 $\{f_i(z)\}_{i=1}^K$ に対応するパラメータ、 $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_K^T]^T$ である。

【0187】

なお、確率分布 $q_0(z_0)$ は、モンテカルロサンプリングが容易な分布に限るものではなく、厳密な確率密度推定が容易な分布であればよい。厳密な確率密度推定が容易な分布の一例として、以下の条件を満たす確率分布 $p(x)$ が挙げられる。

(条件1) 確率分布 $p(x)$ に対して、 \mathbb{R}^D 上の非負関数 $g(x)$ ($g \geq 0$)が存在し、任意の $x \in \mathbb{R}^D$ に対して $p(x) = g(x) / \int g(x) dx$ となる。

(条件2) 関数 $g(x)$ に対して $\int g(x) dx$ を計算することは容易である。

【0188】

条件2を満たす関数として、ガウス分布が挙げられる。一方、条件2を満たさない関数として、 $g(x) = \exp(\sin(x) - x^2)$ が挙げられる。

【0189】

ノーマライジング・フローでは、確率分布 $q(x; \theta)$ のパラメータ θ を入力データの集合 $\{x_i\}_{i=1}^N$ を用いて学習する。そして学習により得られたパラメータ (学習済みパラメータという) θ を用いた確率分布 $q(x; \theta)$ により本来のデータ生成に関する確率分布 $p(x)$ を近似する。

【0190】

ノーマライジング・フローでは、変換 $\{f_i(z)\}_{i=1}^K$ として様々なものを用いることができる。例えば、参考非特許文献3に記載のバッチ正規化(Batch Normalization)やLeaky ReLU(Rectified Linear Unit: 正規化線形関数)などを用いることができる。また、参考非特許文献4に記載の以下の線型変換を用いることもできる。

【0191】

【数27】

$$f(z) = LUz \quad \cdots (23)$$

【0192】

ここで、 $L, U \in \mathbb{R}^D \times \mathbb{R}^D$ は、それぞれ下三角行列、上三角行列である。この変換は L と U の対角成分の積の絶対値 (つまり、 $\prod_{i=1}^D |L_{ii} U_{ii}|$) によりヤコビアン行列式の絶対値 $|\det(\partial f(z; \theta) / \partial z)|$ を計算することができるため、入力データ x の確率密度 $q(x; \theta)$ を容易に計算することができる (換言すれば、入力データ x の確率密度 $q(x; \theta)$ の計算コストを抑えることができる) という特徴がある (式(22)参照)。

【0193】

以下、簡単にバッチ正規化BN: $x \rightarrow y$ ($x, y \in \mathbb{R}^D$) について説明する。バッチ正規化BNは、入力データの集合 $\{x_i\}_{i=1}^N$ の各次元の要素の平均が0、分散が1になるように調整した後、スケール変換とシフト変換を行う。具体的には、次式により、 $y_i = \text{BN}(x_i)$ が計算される。

【0194】

【数28】

10

20

30

40

50

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad \dots(24a)$$

$$s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - m)^2 \quad \dots(24b)$$

$$\hat{x}_i = \frac{x_i - m}{\sqrt{s^2 + \varepsilon}} \quad \dots(24c)$$

$$y_i = \gamma \hat{x}_i + \beta \quad \dots(24d)$$

10

【0195】

ここで、 γ 、 β は、それぞれスケール変換パラメータ、シフト変換パラメータであり、いずれも学習対象となるパラメータである。また、 ε は非負の実数であり、ゼロで割ることを避けたい場合は正の実数、避ける必要がない場合はゼロとすればよい。

【0196】

なお、スケール変換パラメータ γ 、シフト変換パラメータ β を明示するため、BN(\cdot)を $\text{BN}(\gamma, \beta, \cdot)$ と表すこともある。

【0197】

ノーマライジング・フローで用いるK個の変換すべてが同じ種類の変換である必要はない。したがって、例えば、変換 $f_1(z)$ をバッチ正規化、変換 $f_2(z)$ を線形変換というように、いくつかの種類の変換を組み合わせてもよい。

20

【0198】

《適応バッチ正規化(AdaBN: Adaptive Batch Normalization)》

ドメイン適応とは、モデル学習に用いる学習データの分布と学習済みモデルを用いた処理の対象であるテストデータの分布が異なる場合に、当該分布の相違により、学習済みモデルを用いた処理の精度が低下しないように、学習済みモデルを調整する技術のことである。ここで、学習データの集合、テストデータの集合がドメインであり、それぞれ学習用ドメイン、テスト用ドメインということもある。

30

【0199】

深層ニューラルネットワーク(DNN: Deep Neural Network)と組み合わせることができドメイン適応の手法には様々なものがあるが、ここでは適応バッチ正規化について説明する(参考非特許文献5参照)。適応バッチ正規化は、バッチ正規化における平均・分散の計算と平均・分散の調整(式(24a)~(24d)参照)をドメインごとに行うという手法である。つまり、式(24a)~(24c)による計算を同一のドメインのデータごとに行うようにする。実際にテストする際には、テスト用ドメインの入力データの集合 $\{x_i\}_{i=1}^N$ に対して統計量(平均と分散)を計算して、その統計量を用いて式(24c)、式(24d)により処理結果 y_i を出力する。なお、変換が適応バッチ正規化である場合、AdaBN: $x \rightarrow y(x, y \in \mathbb{R}^D)$ と表すこともある。

40

【0200】

《AdaFlow》

AdaFlowは、ノーマライジング・フローに適応バッチ正規化を導入した手法である。具体的には、ノーマライジング・フローで用いるK個の変換 $\{f_i(z)\}_{i=1}^K$ のうち少なくとも1つの変換 $f_{i_0}(z)$ に対して、その逆変換 $f_{i_0}^{-1}(z)$ が適応バッチ正規化であるというものである。なお、適応バッチ正規化の計算のうち、スケール変換及びシフト変換、つまり、式(24d)の計算は、省略したものであってもよい。別の言い方をすると、逆変換 $f_{i_0}^{-1}(z)$ が $\gamma = 1$ 、 $\beta = 0$ である適応バッチ正規化であるとも表現できる。

【0201】

AdaFlowを用いて学習をすると、1つの学習済みモデルから複数ドメインの確率分布の

50

生成が行えるようになる。また、複数ドメイン間でのデータ変換も行えるようになる。

【 0 2 0 2 】

以下、AdaFlowについて具体的に説明する。AdaFlowとは、学習データから生成される入力データ x を、確率分布 $q_0(z_0)$ に従って生成されているとみなす潜在変数 $z_0(\sim q(z_0))$ に変換するニューラルネットワークである。ここでは5個の変換 $\{f_i(z)\}_{i=1}^5$ を用いる場合について説明する。つまり、 z_0 は、 $z_0=f_1^{-1}(f_2^{-1}(f_3^{-1}(f_4^{-1}(f_5^{-1}(x))))))$ として得られることになる。

【 0 2 0 3 】

上記5個の変換 $\{f_i(z)\}_{i=1}^5$ を次式により定義する。なお、便宜上変換 f_i を示す代わりにその逆変換 f_i^{-1} を示す(ただし、 $z_4=f_5^{-1}(x)$, $z_3=f_4^{-1}(z_4)$, $z_2=f_3^{-1}(z_3)$, $z_1=f_2^{-1}(z_2)$, $z_0=f_1^{-1}(z_1)$)。

【 0 2 0 4 】

【数 2 9】

$$f_5^{-1}(x) = L_5 D_5 U_5 x \quad \dots(25a)$$

$$f_4^{-1}(z_4) = \text{AdaBN}_{\gamma_4 \beta_4}(z_4) \quad \dots(25b)$$

$$f_3^{-1}(z_3) = \text{LeakyReLU}(z_3) = \max(z_3, \alpha_3 z_3) \quad \dots(25c)$$

$$f_2^{-1}(z_2) = L_2 D_2 U_2 z_2 \quad \dots(25d)$$

$$f_1^{-1}(z_1) = \text{AdaBN}_{\gamma_1 \beta_1}(z_1) \quad \dots(25e)$$

【 0 2 0 5 】

ここで、 $L_2, L_5 \in \mathbb{R}^{D \times D}$ は、対角成分が1の下三角行列であり、上三角部分を除くすべての要素 $L_{2,ij}, L_{5,ij}(i < j)$ が学習対象パラメータ(つまり、パラメータ θ_2 またはパラメータ θ_5)である。 $D_2, D_5 \in \mathbb{R}^{D \times D}$ は、対角行列であり、対角成分 $D_{2,ij}, D_{5,ij}(i=j)$ が学習対象パラメータ(つまり、パラメータ θ_2 またはパラメータ θ_5)である。 $U_2, U_5 \in \mathbb{R}^{D \times D}$ は、対角成分が1の上三角行列であり、下三角部分を除くすべての要素 $U_{2,ij}, U_{5,ij}(i < j)$ が学習対象パラメータ(つまり、パラメータ θ_2 またはパラメータ θ_5)である。また、 $\alpha_3(0)$ はLeakyReLUのパラメータであり、超パラメータとして設定してもよいし、学習対象パラメータ(つまり、パラメータ θ_3)としてもよい(θ_3 を学習対象とする場合は、Parametric ReLUと呼ばれる(参考非特許文献6))。また、 $\text{AdaBN}_{\gamma_4 \beta_4}(\cdot)$ 及び $\text{AdaBN}_{\gamma_1 \beta_1}(\cdot)$ は上述の適応バッチ正規化であり、 $\gamma_1, \beta_1, \gamma_4, \beta_4$ が学習対象パラメータ(つまり、パラメータ θ_1 またはパラメータ θ_4)である。

【 0 2 0 6 】

また、変換 $\{f_i(z)\}_{i=1}^5$ のヤコビアン行列式の絶対値は、それぞれ次式により計算される(ただし、 $x=f_5(z_4)$, $z_4=f_4(z_3)$, $z_3=f_3(z_2)$, $z_2=f_2(z_1)$, $z_1=f_1(z_0)$)。

【 0 2 0 7 】

【数 3 0】

10

20

30

40

50

$$|\det(\partial f_5(z_4)/\partial z_4)| = 1/|\prod_{i=1}^D D_{5,\bar{i}}| \quad \dots(26a)$$

$$|\det(\partial f_4(z_3)/\partial z_3)| = \sqrt{s_4'^2 + \varepsilon}/\gamma_4 \quad \dots(26b)$$

$$|\det(\partial f_3(z_2)/\partial z_2)| = 1/\alpha_3^\delta \quad \dots(26c)$$

$$|\det(\partial f_2(z_1)/\partial z_1)| = 1/|\prod_{i=1}^D D_{2,\bar{i}}| \quad \dots(26d)$$

$$|\det(\partial f_1(z_0)/\partial z_0)| = \sqrt{s_1'^2 + \varepsilon}/\gamma_1 \quad \dots(26e)$$

10

【 0 2 0 8 】

ここで、 s_4' は（学習データから生成された入力データ x に対応する） z_4 の標準偏差、 δ は（学習データから生成された入力データ x に対応する） z_3 のうちゼロを下回る要素の数、 s_1' は（学習データから生成された入力データ x に対応する） z_0 の標準偏差である。なお、変換 f_4, f_1 に対するヤコビアン行列式の絶対値 $|\det(\partial f_4(z_3)/\partial z_3)|, |\det(\partial f_1(z_0)/\partial z_0)|$ に関しては、学習時ではなく推論時（つまり、学習済みモデルを用いた処理時）におけるヤコビアンの行列式の絶対値を用いて表記している。

【 0 2 0 9 】

また、先述の通り、確率分布 $q_0(z_0)$ は厳密な確率密度推定が容易な確率分布とする。例えば、確率分布 $q_0(z_0)$ を平均が0、分散が単位行列 I のガウス分布 $N(0, I)$ とすると、確率分布 $q_0(z_0)$ は次式により表される。

【 0 2 1 0 】

【数 3 1】

$$q_0(z_0) = -(2\pi)^{-D/2} \exp(-\|z_0\|_2^2) \quad \dots(27)$$

20

【 0 2 1 1 】

したがって、 $q_1(x_i)$ を学習データから生成された入力データ x の確率分布とし、式(22)を用いることにより、潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ から入力データ x_i の確率密度 $q_1(x_i)$ が計算できることがわかる。

【 0 2 1 2 】

続いて、パラメータ θ の学習方法について説明する。通常のニューラルネットワークの学習と同様、損失関数 $L(\theta)$ を用いて、例えば、勾配降下法、Momentum SGD(Stochastic Gradient Descent)、ADAM(Adaptive Moment Estimation)、あるいはそれらの組合せにより学習することができる。ノーマライジング・フローを用いる場合、損失関数 $L(\theta)$ として、次式で定義される負の対数尤度の平均を用いることが多い。

【 0 2 1 3 】

【数 3 2】

$$L(\theta) = -1/N \sum_{i=1}^N \log q(x_i; \theta) \quad \dots(28)$$

40

【 0 2 1 4 】

なお、上記学習には、ミニバッチと呼ばれる学習データの集合を単位にして行うミニバッチ学習法を用いることができる。ここで、ミニバッチとは、すべての学習データからランダムに選択した複数の学習データのことであり、このミニバッチごとに損失関数 $L(\theta)$ の値を計算することになる。

【 0 2 1 5 】

< 第 7 実施形態 >

50

確率分布学習装置 1100 は、P 種類のドメイン D_j ($j=1, \dots, P$) のデータ (以下、ドメインデータという。) から生成される入力データ x を、確率分布 $q_0(z_0)$ に従って生成されているとみなす潜在変数 z_0 ($\sim q(z_0)$) に変換するニューラルネットワークのモデルパラメータを学習する。このモデルパラメータを用いて、確率分布 $q_0(z_0)$ から入力データ x の確率分布 $q(x; \cdot)$ を求めることができる (式(22)参照)。ここで、ドメイン D_j は、 N_j 個のドメインデータを含むものとする。したがって、P 種類のドメインの和集合に含まれるドメインデータの数を N とすると、 $N = \sum_j N_j$ となる。

【0216】

以下、図17～図18を参照して確率分布学習装置 1100 を説明する。図17は、確率分布学習装置 1100 の構成を示すブロック図である。図18は、確率分布学習装置 1100 の動作を示すフローチャートである。図17に示すように確率分布学習装置 1100 は、入力データ生成部 1110 と、潜在変数推定部 1120 と、損失関数計算部 1130 と、パラメータ更新部 1140 と、収束条件判定部 1150 と、記録部 1190 を含む。記録部 1190 は、確率分布学習装置 1100 の処理に必要な情報を適宜記録する構成部である。記録部 1190 は、例えば、入力データ x の確率分布 $q(x; \cdot)$ のパラメータを記録する。パラメータの初期値として、例えば乱数を用いて生成された値が記録されている。

10

【0217】

確率分布学習装置 1100 は、学習用ドメインデータ記録部 1910 に接続している。学習用ドメインデータ記録部 1910 には、事前に用意したドメインデータ s_i ($i=1, \dots, N$) と当該ドメインデータ s_i が含まれるドメインを識別するためのドメイン識別子 j の組 (s_i, j) が学習データとして記録されている。なお、 (s_i, j) のことを学習用ドメインデータともいう。

20

【0218】

図18に従い確率分布学習装置 1100 の動作について説明する。S1110において、入力データ生成部 1110 は、入力である学習用ドメインデータ (s_i, j) ($i=1, \dots, N$, s_i はドメインデータ、 j は s_i を含むドメインのドメイン識別子である) を入力とし、ドメインデータ s_i から、入力データ x_i ($i=1, \dots, N$) を生成し、入力データ x_i とドメイン識別子 j の組を出力する。ドメインデータから入力データを生成する方法は、入力データからドメインデータを復元できるのであれば、どのような方法であってもよい。例えば、ドメインデータが画像や音である場合、ドメインデータ s_i から生成される入力データ x_i として、抽出元である画像や音を復元することができる所定の特徴量を用いるとよい。

30

【0219】

S1120において、潜在変数推定部 1120 は、S1110で生成した入力データ x_i ($i=1, \dots, N$) とドメインデータ s_i を含むドメインのドメイン識別子 j の組を入力とし、確率分布 $q(x; \cdot)$ のパラメータを用いて、入力データ x_i ($i=1, \dots, N$) から、入力データ x_i に対応する潜在変数 $z_{0,i}$ ($i=1, \dots, N$) を推定し、出力する。なお、ここで用いるパラメータは、学習中の値である。

【0220】

ここで、確率分布 $q(x; \cdot)$ の変数 x は、P 種類のドメイン D_j のドメインデータから生成された入力データを示す変数であり、変数 x は、変換 f_i ($i=1, \dots, K$, K は 1 以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する) と潜在変数 z_0 を用いて、 $x = f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表されるものとする。

40

【0221】

したがって、入力データ x_i に対応する潜在変数 $z_{0,i}$ は次式により与えられる。

【0222】

【数33】

$$z_{0,i} = f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x_i))\dots)) \quad \dots(29)$$

50

【 0 2 2 3 】

また、潜在変数 $z_{0,i}(i=1, \dots, N)$ は、潜在変数 z_0 の確率分布 $q_0(z_0)$ に従って生成されるものとする。ただし、確率分布 $q_0(z_0)$ は厳密な確率密度推定が容易であるという性質を有する。

【 0 2 2 4 】

したがって、確率分布 $q(x; \theta)$ は、確率分布 $q_0(z_0)$ を用いた次式により表現できる。

【 0 2 2 5 】

【 数 3 4 】

$$q(x; \theta) = q_0(z_0) \left| \det\left(\frac{\partial f_K(z_{K-1}; \theta_K)}{\partial z_{K-1}}\right) \right| \left| \det\left(\frac{\partial f_{K-1}(z_{K-2}; \theta_{K-1})}{\partial z_{K-2}}\right) \right| \cdots \left| \det\left(\frac{\partial f_1(z_0; \theta_1)}{\partial z_0}\right) \right| \cdots (22)$$

10

【 0 2 2 6 】

ここで、 θ_i は変換 f_i に対応するパラメータであり、 $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_K^T]^T$ となる。

【 0 2 2 7 】

潜在変数推定部 1 1 2 0 は、入力データ x から潜在変数 z_0 を計算するニューラルネットワークNNを用いて実現できる。なお、潜在変数 z_0 の計算式は以下の通りである。

【 0 2 2 8 】

【 数 3 5 】

$$z_0 = f_1^{-1}(f_2^{-1}(\cdots(f_K^{-1}(x))\cdots)) \cdots (29)'$$

20

【 0 2 2 9 】

つまり、ニューラルネットワークNNは、入力データ x を入力とし、逆変換 $f_K^{-1}, f_{K-1}^{-1}, \dots, f_2^{-1}, f_1^{-1}$ の順に計算を進めていき、最終的に潜在変数 z_0 を出力するものである。

【 0 2 3 0 】

上記変換 $f_i(i=1, \dots, K)$ のうち、少なくとも1つの変換 f_{i_0} (ただし、 i_0 は $1 \leq i_0 \leq K$ を満たす整数)に対して、その逆変換 $f_{i_0}^{-1}$ は適応バッチ正規化となるようにする。

【 0 2 3 1 】

また、変換 $f_i(i=1, \dots, K)$ に含まれる変換 f_{i_1} (ただし、 i_1 は $1 \leq i_1 \leq K$ を満たす整数)に対して、その逆変換 $f_{i_1}^{-1}$ は、線形変換であり、当該線形変換に対応する行列が、下方三角行列と上方三角行列の積、または、下方三角行列と対角行列と上方三角行列の積として表されているようにしてもよい。

30

【 0 2 3 2 】

具体的な変換 $f_i(i=1, \dots, K)$ の例として、例えば $K=5$ として、逆変換が式(25a)~式(25e)で表される5つの変換を用いることができる。

【 0 2 3 3 】

また、潜在変数推定部 1 1 2 0 は、逆変換が適応バッチ正規化である変換 f_{i_0} に対して、逆変換 $f_{i_0}^{-1}$ の計算に用いる平均と分散を入力データ $x_i(i=1, \dots, N)$ から計算される逆変換 $f_{i_0}^{-1}$ への入力データ $z_{i_0,i} (= f_{i_0-1}^{-1}(f_{i_0-2}^{-1}(\dots(f_K^{-1}(x_i))\dots)))$ の平均と分散をドメインごとに計算するため、 $(z_{i_0,i}, j)$ (ただし、 j は入力データ x_i の生成元であるドメインデータ s_i のドメイン識別子である)を記録部 1 1 9 0 に記録しておく。以下、逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_j の平均と分散をそれぞれ $m_{i_0,j}$ と $s_{i_0,j}^2$ と表すことにする。なお、平均 $m_{i_0,j}$ と分散 $s_{i_0,j}^2$ を計算するのは、後述するように、収束条件判定部 1 1 5 0 である。

40

【 0 2 3 4 】

S 1 1 3 0 において、損失関数計算部 1 1 3 0 は、S 1 1 2 0 で推定した潜在変数 $z_{0,i}(i=1, \dots, N)$ を入力とし、潜在変数 $z_{0,i}(i=1, \dots, N)$ から、確率分布 $q(x; \theta)$ のパラメータの最適化に用いる損失関数 $L(\theta)$ の値を計算し、出力する。損失関数 $L(\theta)$ は、例えば、式(28)のように負の対数尤度の平均として定義することができる。その際、入力データ $x_i(i=$

50

1, ..., N)の確率密度 $q(x_i; \theta)$ を計算する必要があるが、入力データ x_i の確率密度 $q(x_i; \theta)$ は入力データ x_i に対応する潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ を用いて計算することができる。例えば、確率分布 $q_0(z_0)$ がガウス分布 $N(0, I)$ である場合、潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ は、次式で計算できるので、

【0235】

【数36】

$$q_0(z_{0,i}) = -(2\pi)^{-D/2} \exp(-\|z_{0,i}\|_2^2) \quad \dots(27)'$$

【0236】

式(22)を用いて、上記計算した潜在変数 $z_{0,i}$ の確率密度 $q_0(z_{0,i})$ から入力データ x_i の確率密度 $q(x_i; \theta)$ を計算することができる。

【0237】

S1140において、パラメータ更新部1140は、S1130で計算した損失関数 $L(\theta)$ の値を入力とし、損失関数 $L(\theta)$ の値を最適化(最小化)するように、確率分布 $q(x; \theta)$ のパラメータ θ を更新し、出力する。パラメータ θ の更新には、例えば、勾配降下法を用いるとよい。

【0238】

S1150において、収束条件判定部1150は、パラメータ更新の終了条件として事前に設定された収束条件を判定し、収束条件が満たされた場合はS1140で更新したパラメータ θ (このパラメータのことを学習済みパラメータという)を用いた確率分布 $q(x; \theta)$ を出力する。その際、収束条件判定部1150は、S1120において記録した $(z_{i_0,i}, j)$ ($i=1, \dots, N$)を用いて、逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_{j_0} の平均 $m_{i_0,j}$ と分散 $s_{i_0,j}^2$ ($j=1, \dots, P$)とを計算し、出力する。一方、収束条件が満たされない場合、収束条件判定部1150は、S1110~S1140の処理を繰り返す。収束条件として、例えばS1110~S1140の処理の実行回数が所定の回数に達したかという条件を採用することができる。なお、S1140で更新したパラメータ θ 、逆変換 $f_K^{-1}(x; \theta)$, $f_{K-1}^{-1}(z_{K-1}; \theta_{K-1})$, ..., $f_2^{-1}(z_2; \theta_2)$, $f_1^{-1}(z_1; \theta_1)$ を出力するようにしてもよい。以下、逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_j の平均 $m_{i_0,j}$ と分散 $s_{i_0,j}^2$ のことをドメイン D_j のドメインデータから算出される統計量という。

【0239】

本実施形態の発明によれば、P種類のドメイン D_j のドメインデータから生成される入力データ x の確率分布 $q(x; \theta)$ を学習することができる。ノーマライジング・フローに基づくAdaFlowを用いて学習することにより、ペアデータなし学習を安定的に行うことが可能となる。

【0240】

<第8実施形態>

データ変換装置1200は、確率分布学習装置1100で学習した学習済みモデルパラメータ θ と、逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_j の平均 $m_{i_0,j}$ と分散 $s_{i_0,j}^2$ ($j=1, \dots, P$)とを用いて、ドメイン D_{j_0} のドメインデータをドメイン D_{j_1} のドメインデータに変換する(ただし、 j_0, j_1 は $1 \leq j_0, j_1 \leq P$ を満たす整数であり、 $j_0 \neq j_1$ である)。以下、ドメイン D_{j_0} のことを変換元ドメイン、ドメイン D_{j_1} のことを変換先ドメインという。

【0241】

以下、図19~図20を参照してデータ変換装置1200を説明する。図19は、データ変換装置1200の構成を示すブロック図である。図20は、データ変換装置1200の動作を示すフローチャートである。図19に示すようにデータ変換装置1200は、入力データ生成部1110と、潜在変数計算部1220と、出力データ計算部1230と、ドメインデータ生成部1240と、記録部1190を含む。記録部1190は、データ変換装置1200の処理に必要な情報を適宜記録する構成部である。記録部1190は、例えば、確率分布学習装置1100を用いて学習した確率分布 $q(x; \theta)$ のパラメータ θ (つま

10

20

30

40

50

り、学習済みパラメータ) を記録しておく。また、記録部 1 1 9 0 は、例えば、逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_j の平均 $m_{i_0,j}$ と分散 $s_{i_0,j}^2$ ($j=1, \dots, P$) を記録しておく。

【 0 2 4 2 】

図 2 0 に従いデータ変換装置 1 2 0 0 の動作について説明する。S 1 1 1 0 において、入力データ生成部 1 1 1 0 は、入力である変換元ドメイン D_{j_0} のドメインデータ s とそのドメイン識別子 j_0 (変換元ドメイン識別子という) を入力とし、ドメインデータ s から、入力データ x を生成し、出力する。

【 0 2 4 3 】

S 1 2 2 0 において、潜在変数計算部 1 2 2 0 は、S 1 1 1 0 で生成した入力データ x と変換元ドメイン識別子 j_0 を入力とし、学習済みパラメータ と逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_{j_0} の平均 m_{i_0,j_0} と分散 s_{i_0,j_0}^2 とを用いて、入力データ x から、入力データ x に対応する潜在変数 z_0 を計算し、出力する。入力データ x に対応する潜在変数 z_0 は、確率分布学習装置 1 1 0 0 で用いた変換 f_i ($i=1, \dots, K$) を用いて、次式により計算される。

10

【 0 2 4 4 】

【数 3 7】

$$z_0 = f_1^{-1}(f_2^{-1}(\dots(f_K^{-1}(x))\dots)) \dots(29)'$$

20

【 0 2 4 5 】

その際、学習済みパラメータ と逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_{j_0} の平均 m_{i_0,j_0} と分散 s_{i_0,j_0}^2 とを用いる。潜在変数計算部 1 2 2 0 は、この点において確率分布学習装置 1 1 0 0 の潜在変数推定部 1 1 2 0 と異なる。

【 0 2 4 6 】

S 1 2 3 0 において、出力データ計算部 1 2 3 0 は、変換先ドメインの識別子である変換先ドメイン識別子 j_1 と S 1 2 2 0 で計算した潜在変数 z_0 とを入力とし、学習済みパラメータ と逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_{j_1} の平均 m_{i_0,j_1} と分散 s_{i_0,j_1}^2 とを用いて、潜在変数 z_0 から、潜在変数 z_0 に対応する出力データ x' を計算し、出力する。潜在変数 z_0 に対応する出力データ x' は、確率分布学習装置 1 1 0 0 で用いた変換 f_i ($i=1, \dots, K$) を用いて、次式により計算される。

30

【 0 2 4 7 】

【数 3 8】

$$x' = f_K(f_{K-1}(\dots(f_1(z_0))\dots)) \dots(21)''$$

【 0 2 4 8 】

その際、学習済みパラメータ と逆変換 $f_{i_0}^{-1}$ への入力データのドメイン D_{j_1} の平均 m_{i_0,j_1} と分散 s_{i_0,j_1}^2 とを用いる。なお、出力データ計算部 1 2 3 0 を実現するニューラルネットワークは、潜在変数計算部 1 2 2 0 を実現するニューラルネットワーク NN の出力を入力に、ニューラルネットワーク NN の出力を出力にしたものとすればよい。

40

【 0 2 4 9 】

潜在変数計算部 1 2 2 0 と出力データ計算部 1 2 3 0 による処理の様子を図 2 1 に示す。図 2 1 は、< 技術的背景 > の説明で用いた 5 個の変換 $\{f_i(z)\}_{i=1}^5$ により、入力データから潜在変数、潜在変数から出力データに変換する様子を示している。ここで、 $f_1^{-1}(j_0)$, $f_1(j_1)$ はドメイン D_{j_0} の平均 m_{i_0,j_0} と分散 s_{i_0,j_0}^2 を、 $f_4^{-1}(j_0)$, $f_4(j_1)$ はドメイン D_{j_1} の平均 m_{i_0,j_1} と分散 s_{i_0,j_1}^2 を用いた関数となる。

【 0 2 5 0 】

S 1 2 4 0 において、ドメインデータ生成部 1 2 4 0 は、S 1 2 3 0 で計算した出力データ x' を入力とし、出力データ x' から、変換先ドメイン D_{j_1} のデータである変換済ドメイ

50

ンデータ s' を生成し、出力する。

【0251】

なお、潜在変数計算部1220、出力データ計算部1230を含む構成部をデータ変換部1205という。つまり、データ変換部1205は、変換元ドメインのドメインデータから生成した入力データを、変換先ドメインのドメインデータの生成元となる出力データに変換する構成部である。

【0252】

本実施形態の発明によれば、ドメイン間でデータを相互に変換することが可能となる。

【0253】

<第9実施形態>

第7実施形態及び第8実施形態では、 P を1以上の整数とし、 P 種類のドメインのドメインデータを用いるものとして説明した。ここでは、 $P=2$ とした場合のデータ変換装置1300について説明する。

【0254】

以下、図22～図23を参照してデータ変換装置1300を説明する。図22は、データ変換装置1300の構成を示すブロック図である。図23は、データ変換装置1300の動作を示すフローチャートである。図22に示すようにデータ変換装置1300は、入力データ生成部1310と、潜在変数計算部1320と、出力データ計算部1330と、ドメインデータ生成部1340と、記録部1190を含む。記録部1190は、データ変換装置1300の処理に必要な情報を適宜記録する構成部である。記録部1190は、例えば、確率分布学習装置1100を用いて学習した確率分布 $q(x; \theta)$ のパラメータ θ （つまり、学習済みパラメータ θ ）を記録しておく。

【0255】

以下、変換元ドメインを第1ドメイン、変換先ドメインを第2ドメインと呼ぶことにし、それぞれ D_1 、 D_2 で表す。また、潜在変数計算部1320には、逆変換 $f_{i_0}^{-1}$ への入力データの第1ドメイン D_1 の平均 $m_{i_0,1}$ と分散 $s_{i_0,1}^2$ を設定しておく。同様に、出力データ計算部1330には、逆変換 $f_{i_0}^{-1}$ への入力データの第2ドメイン D_2 の平均 $m_{i_0,2}$ と分散 $s_{i_0,2}^2$ を設定しておく。

【0256】

図23に従いデータ変換装置1300の動作について説明する。S1310において、入力データ生成部1310は、第1ドメインのドメインデータ s を入力とし、ドメインデータ s から、入力データ x を生成し、出力する。以下、この入力データのことを第1ドメインのドメインデータに対応する入力データともいう。

【0257】

S1320において、潜在変数計算部1320は、S1310で生成した入力データ x を入力とし、学習済みパラメータ θ を用いて、入力データ x から、入力データ x に対応する潜在変数 z_0 を計算し、出力する。

【0258】

S1330において、出力データ計算部1330は、S1320で計算した潜在変数 z_0 を入力とし、学習済みパラメータ θ を用いて、潜在変数 z_0 から、潜在変数 z_0 に対応する出力データ x' を計算し、出力する。

【0259】

S1340において、ドメインデータ生成部1340は、S1330で計算した出力データ x' を入力とし、出力データ x' から、第2ドメインのドメインデータ s' を生成し、出力する。以下、この出力データのことを第2ドメインのドメインデータに対応する出力データともいう。

【0260】

なお、潜在変数計算部1320、出力データ計算部1330を含む構成部をデータ変換部1305という。つまり、データ変換部1305は、第1ドメインのドメインデータに対応する入力データを、第2ドメインのドメインデータに対応する出力データに変換する

10

20

30

40

50

構成部である。

【 0 2 6 1 】

以上の説明からわかるように、潜在変数計算部 1 3 2 0 は、潜在変数を入力データに変換する、逆関数を有する関数（以下、第 1 関数という）を用いて計算するものであり、出力データ計算部 1 3 3 0 は、潜在変数を出力データに変換する、逆関数を有する関数（以下、第 2 関数という）を用いて計算するものであるといえる。そして、第 1 関数と第 2 関数は、潜在変数 z_0 を変数 x に変換する所定の関数から導出されるものである。この所定の関数は、第 1 ドメインと第 2 ドメインの和集合を用いて得られる関数である。より具体的に言えば、当該所定の関数は、変換 f_i ($i=1, \dots, K$ 、 K は 1 以上の整数、変換 f_i に対して逆変換 f_i^{-1} が存在する) を用いて、 $x=f_K(f_{K-1}(\dots(f_1(z_0))\dots))$ と表される関数（ただし、変換 f_i ($i=1, \dots, K$) のうち、少なくとも 1 つの変換 f_{i_0} ($1 \leq i_0 \leq K$) に対して、その逆変換 $f_{i_0}^{-1}$ は適応パッチ正規化である) として、第 1 ドメインと第 2 ドメインの和集合の要素であるドメインデータを学習データとして学習することにより、得られる関数である。また、第 1 関数は、当該和集合に含まれる第 1 ドメインのドメインデータから算出される統計量（具体的には、逆変換 $f_{i_0}^{-1}$ への入力データの第 1 ドメイン D_1 の平均 $m_{i_0,1}$ と分散 $s_{i_0,1}^2$ ）を用いて、上記所定の関数から導出されるものであり、第 2 関数は、当該和集合に含まれる第 2 ドメインのドメインデータから算出される統計量（具体的には、逆変換 $f_{i_0}^{-1}$ への入力データの第 2 ドメイン D_2 の平均 $m_{i_0,2}$ と分散 $s_{i_0,2}^2$ ）を用いて、所定の関数から導出されるものである。

【 0 2 6 2 】

[適用例：教師あり異常検知問題への応用]

データ変換装置 1 3 0 0 を教師あり異常検知問題に応用することができる。ここで、教師あり異常検知とは、多数の正常データと少数の異常データから異常検知モデルを学習し、この異常検知モデルを用いて異常検知する枠組みである。

【 0 2 6 3 】

異常データが多いほど高精度なモデルを学習することができるが、多くの異常データを集めるのは難しい。そこで、異常検知対象としたいドメインを複数用意し、できるだけ多くのデータを集め、確率分布学習装置 1 1 0 0 を用いて当該データからモデルパラメータを学習する。そして、データ変換装置 1 3 0 0 を用いて、第 1 ドメインの異常データを第 1 ドメインとは異なる第 2 ドメインの異常データに変換する。このようにすることにより、収集に困難が伴う異常データを人工的に作り出し、その数を増加させることができ、結果としてより高精度な異常検知モデルを学習することが可能となる。

【 0 2 6 4 】

例えば、第 1 ドメインを異常検知対象機器とは異なる 1 以上の機器から発せられた音の集合、第 2 ドメインを異常検知対象機器の音の集合とし、確率分布学習装置 1 1 0 0 を用いて、パラメータを学習する。次に、データ変換装置 1 3 0 0 を用いて、第 1 ドメインのドメインデータである異常検知対象機器とは異なる 1 以上の機器から発せられた異常音から、第 2 ドメインのドメインデータとして異常検知対象機器の音を生成する。当該生成された異常検知対象機器の音は、異常音であることが期待される。そして、生成された異常検知対象機器の音を学習データとして用いて、第 2 ドメインの異常検知モデルを学習する。

【 0 2 6 5 】

なお、発せられた音の多くは、正常音であると考えられる。学習に際しては、できるだけ多くのデータを用いるのが好ましいため、正常音、異常音を問わず、第 1 ドメインの音を学習に用いるようにするとよい。

本実施形態の発明によれば、ドメイン間でデータを相互に変換することが可能となる。その結果、例えば、異常検知モデル学習に用いる異常データを効率的に生成することができるようになる。

【 0 2 6 6 】

< 実験結果 >

10

20

30

40

50

「風景写真」の集合と「風景画」の集合をドメインとし、ノーマライジング・フローのアーキテクチャとして参考非特許文献 8 に記載の Glow を採用し、Glow の Activation Normalization を AdaBN に置き換えたニューラルネットワーク AdaFlow を用いて学習を行う。風景写真と風景画の画像データをそれぞれ 400 枚ずつ集め、これらを用いてニューラルネットワーク AdaFlow を学習させ、データ変換を行う。具体的には、「風景写真」を「風景画」に変換、または、「風景画」を「風景写真」に変換した。

(参考非特許文献 8 : Diederik P. Kingma, Prafulla Dhariwal, “ Glow: Generative Flow with Invertible 1x1 Convolutions ”, arXiv:1807.03039, <https://arxiv.org/abs/1807.03039>)

この実験、つまり、AdaFlow を用いたペアデータなしドメイン変換により、質のいい画像データが生成されることが確認できた。

【 0 2 6 7 】

< 補記 >

図 2 4 は、上述の各装置を実現するコンピュータの機能構成の一例を示す図である。上述の各装置における処理は、記録部 2 0 2 0 に、コンピュータを上述の各装置として機能させるためのプログラムを読み込ませ、制御部 2 0 1 0、入力部 2 0 3 0、出力部 2 0 4 0 などに動作させることで実施できる。

【 0 2 6 8 】

本発明の装置は、例えば単一のハードウェアエンティティとして、キーボードなどが接続可能な入力部、液晶ディスプレイなどが接続可能な出力部、ハードウェアエンティティの外部に通信可能な通信装置（例えば通信ケーブル）が接続可能な通信部、CPU (Central Processing Unit、キャッシュメモリやレジスタなどを備えていてもよい)、メモリである RAM や ROM、ハードディスクである外部記憶装置並びにこれらの入力部、出力部、通信部、CPU、RAM、ROM、外部記憶装置の間のデータのやり取りが可能なように接続するバスを有している。また必要に応じて、ハードウェアエンティティに、CD-ROM などの記録媒体を読み書きできる装置（ドライブ）などを設けることとしてもよい。このようなハードウェア資源を備えた物理的実体としては、汎用コンピュータなどがある。

【 0 2 6 9 】

ハードウェアエンティティの外部記憶装置には、上述の機能を実現するために必要となるプログラムおよびこのプログラムの処理において必要となるデータなどが記憶されている（外部記憶装置に限らず、例えばプログラムを読み出し専用記憶装置である ROM に記憶させておくこととしてもよい）。また、これらのプログラムの処理によって得られるデータなどは、RAM や外部記憶装置などに適宜に記憶される。

【 0 2 7 0 】

ハードウェアエンティティでは、外部記憶装置（あるいは ROM など）に記憶された各プログラムとこの各プログラムの処理に必要なデータが必要に応じてメモリに読み込まれて、適宜に CPU で解釈実行・処理される。その結果、CPU が所定の機能（上記、... 部、... 手段などと表した各構成要件）を実現する。

【 0 2 7 1 】

本発明は上述の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で適宜変更が可能である。また、上記実施形態において説明した処理は、記載の順に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的あるいは個別に実行されるとしてもよい。

【 0 2 7 2 】

既述のように、上記実施形態において説明したハードウェアエンティティ（本発明の装置）における処理機能をコンピュータによって実現する場合、ハードウェアエンティティが有すべき機能の処理内容はプログラムによって記述される。そして、このプログラムをコンピュータで実行することにより、上記ハードウェアエンティティにおける処理機能がコンピュータ上で実現される。

10

20

30

40

50

【 0 2 7 3 】

この処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリ等のようなものでもよい。具体的には、例えば、磁気記録装置として、ハードディスク装置、フレキシブルディスク、磁気テープ等を、光ディスクとして、DVD (Digital Versatile Disc)、DVD-RAM (Random Access Memory)、CD-ROM (Compact Disc Read Only Memory)、CD-R (Recordable) / RW (ReWritable) 等を、光磁気記録媒体として、MO (Magneto-Optical disc) 等を、半導体メモリとしてEEP-ROM (Electrically Erasable and Programmable-Read Only Memory) 等を用いることができる。

10

【 0 2 7 4 】

また、このプログラムの流通は、例えば、そのプログラムを記録したDVD、CD-ROM等の可搬型記録媒体を販売、譲渡、貸与等することによって行う。さらに、このプログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することにより、このプログラムを流通させる構成としてもよい。

【 0 2 7 5 】

このようなプログラムを実行するコンピュータは、例えば、まず、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、一旦、自己の記憶装置に格納する。そして、処理の実行時、このコンピュータは、自己の記憶装置に格納されたプログラムを読み取り、読み取ったプログラムに従った処理を実行する。また、このプログラムの別の実行形態として、コンピュータが可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することとしてもよく、さらに、このコンピュータにサーバコンピュータからプログラムが転送されるたびに、逐次、受け取ったプログラムに従った処理を実行することとしてもよい。また、サーバコンピュータから、このコンピュータへのプログラムの転送は行わず、その実行指示と結果取得のみによって処理機能を実現する、いわゆるASP (Application Service Provider) 型のサービスによって、上述の処理を実行する構成としてもよい。なお、本形態におけるプログラムには、電子計算機による処理の用に供する情報であってプログラムに準ずるもの(コンピュータに対する直接の指令ではないがコンピュータの処理を規定する性質を有するデータ等)を含むものとする。

20

30

【 0 2 7 6 】

また、この形態では、コンピュータ上で所定のプログラムを実行させることにより、ハードウェアエンティティを構成することとしたが、これらの処理内容の少なくとも一部をハードウェア的に実現することとしてもよい。

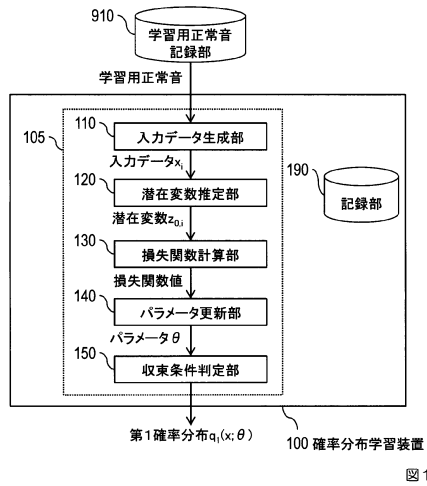
【 0 2 7 7 】

上述の本発明の実施形態の記載は、例証と記載の目的で提示されたものである。網羅的であるという意思はなく、開示された厳密な形式に発明を限定する意思もない。変形やバリエーションは上述の教示から可能である。実施形態は、本発明の原理の最も良い例証を提供するために、そして、この分野の当業者が、熟考された実際の使用に適するように本発明を色々な実施形態で、また、色々な変形を付加して利用できるようにするために、選ばれて表現されたものである。すべてのそのような変形やバリエーションは、公正に合法的に公平に与えられる幅にしたがって解釈された添付の請求項によって定められた本発明のスコープ内である。

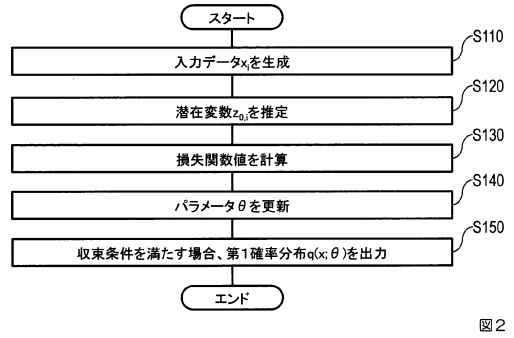
40

【図面】

【図 1】

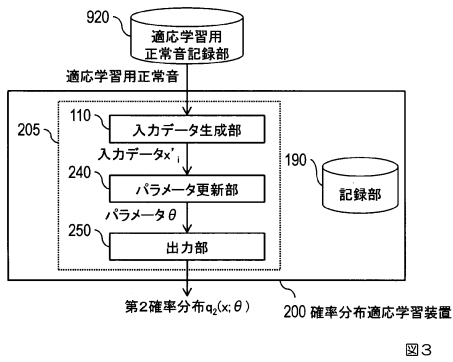


【図 2】

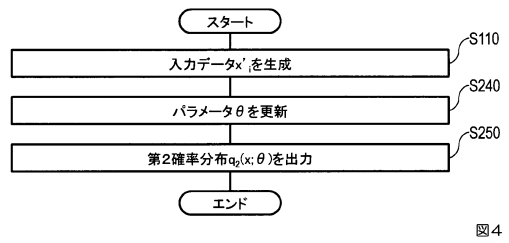


10

【図 3】



【図 4】



20

30

40

50

【 図 5 】

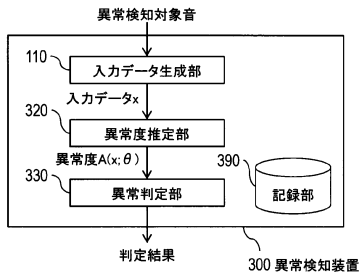


図5

【 図 6 】

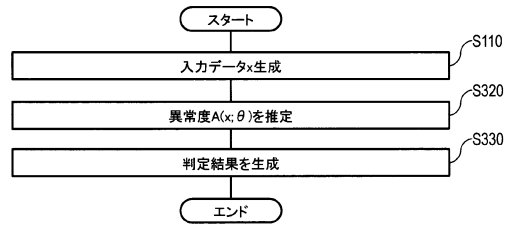


図6

10

【 図 7 】

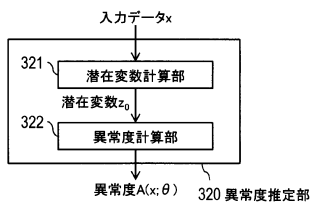


図7

【 図 8 】

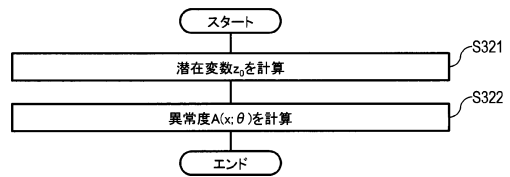


図8

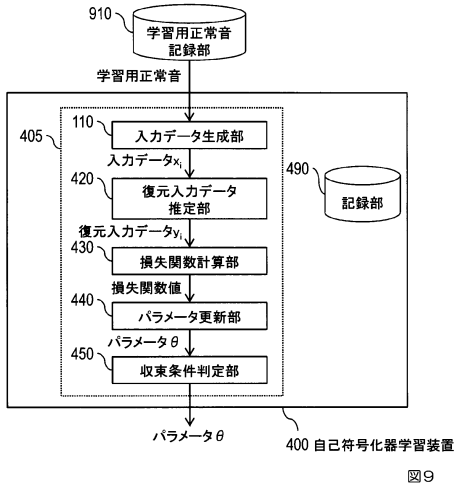
20

30

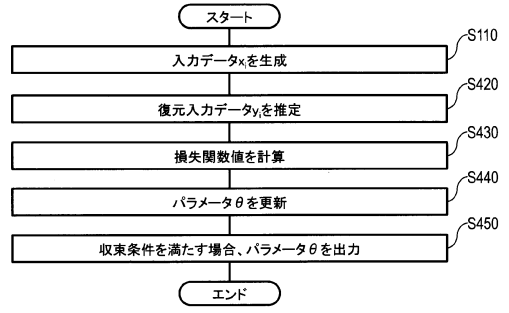
40

50

【 図 9 】

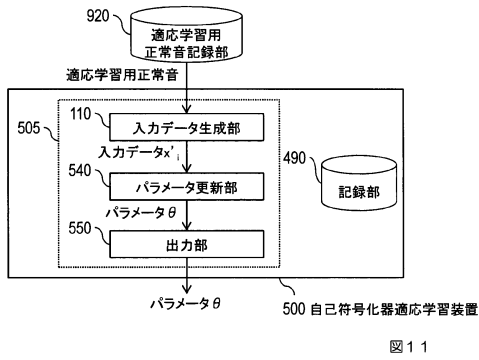


【 図 1 0 】

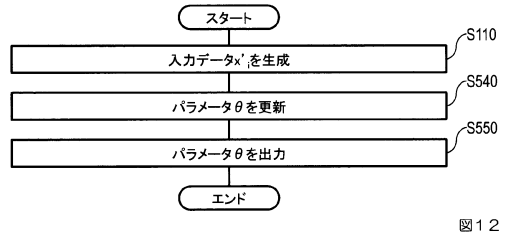


10

【 図 1 1 】



【 図 1 2 】



20

30

40

50

【図13】

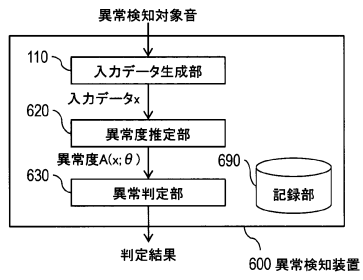


図13

【図14】

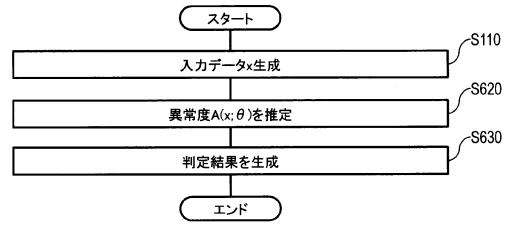


図14

10

【図15】

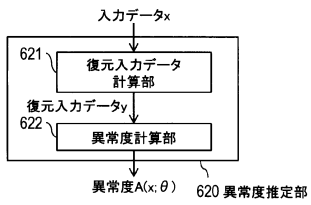


図15

【図16】

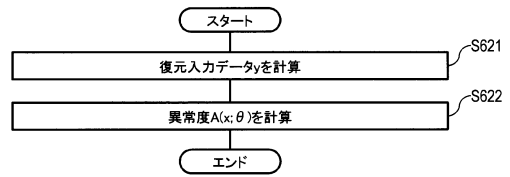


図16

20

30

40

50

【図17】

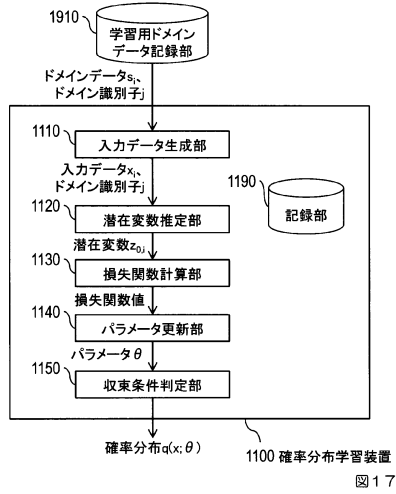


図17

【図18】

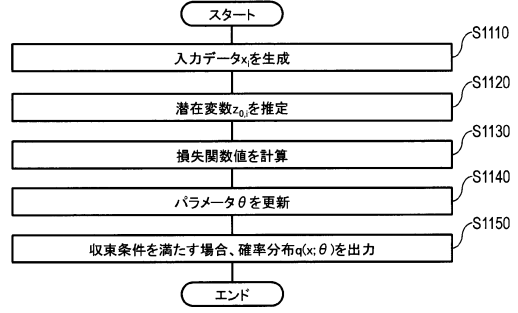


図18

10

【図19】

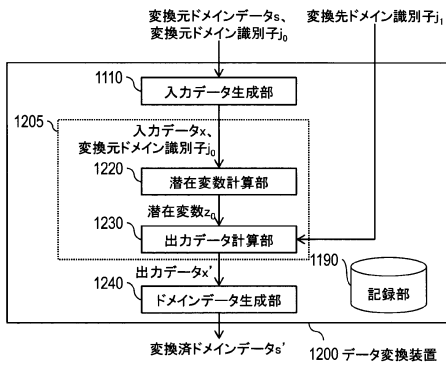


図19

【図20】

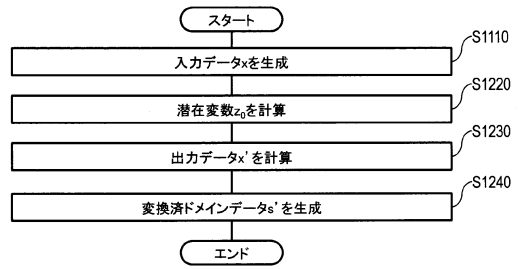


図20

20

30

40

50

【 図 2 1 】

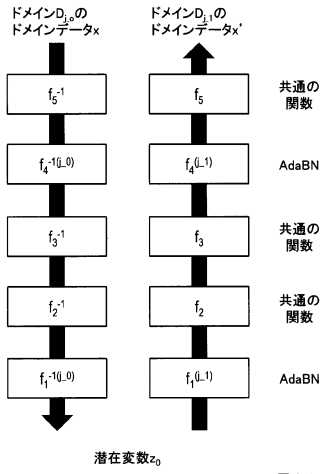


図 2 1

【 図 2 2 】

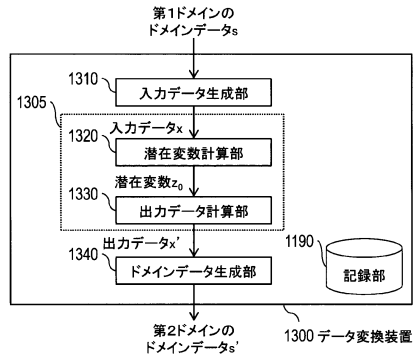


図 2 2

10

20

【 図 2 3 】

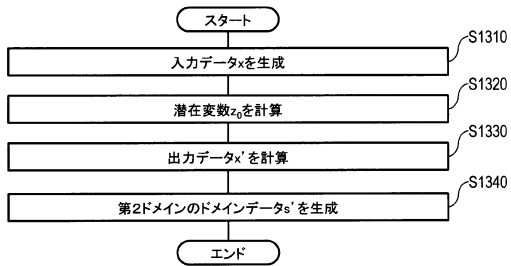


図 2 3

【 図 2 4 】

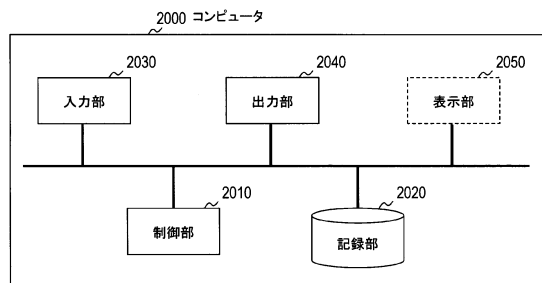


図 2 4

30

40

50

フロントページの続き

8月29日 刊行物 日本音響学会 2018年秋季研究発表会 講演論文集

(72)発明者 原田 登

東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内

審査官 米倉 秀明

(56)参考文献 特開2016-007800(JP, A)

Yunfei Teng, ほか2名, "Invertible Autoencoder for domain adaptation", arXiv.org[online]
, 2018年02月10日, [arXiv:1802.06869v1], [2022年2月8日検索], インターネット <https://arxiv.org/abs/1802.06869> <https://arxiv.org/pdf/1802.06869.pdf>

(58)調査した分野 (Int.Cl., DB名)

G10L 25/51

G05B 23/02

G01M 99/00