

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-20444

(P2010-20444A)

(43) 公開日 平成22年1月28日(2010.1.28)

(51) Int.Cl.

F I

テーマコード (参考)

G06N 3/08 (2006.01)

G06N 3/08

Z

G06N 3/04 (2006.01)

G06N 3/04

F

審査請求 有 請求項の数 9 O L (全 31 頁)

(21) 出願番号 特願2008-178805 (P2008-178805)
 (22) 出願日 平成20年7月9日 (2008.7.9)

(71) 出願人 000002185
 ソニー株式会社
 東京都港区港南1丁目7番1号
 (74) 代理人 100082131
 弁理士 稲本 義雄
 (74) 代理人 100121131
 弁理士 西川 孝
 (72) 発明者 伊藤 真人
 東京都港区港南1丁目7番1号 ソニー株
 式会社内
 (72) 発明者 青山 一美
 東京都港区港南1丁目7番1号 ソニー株
 式会社内
 (72) 発明者 野田 邦昭
 東京都港区港南1丁目7番1号 ソニー株
 式会社内

(54) 【発明の名称】 学習装置、学習方法、およびプログラム

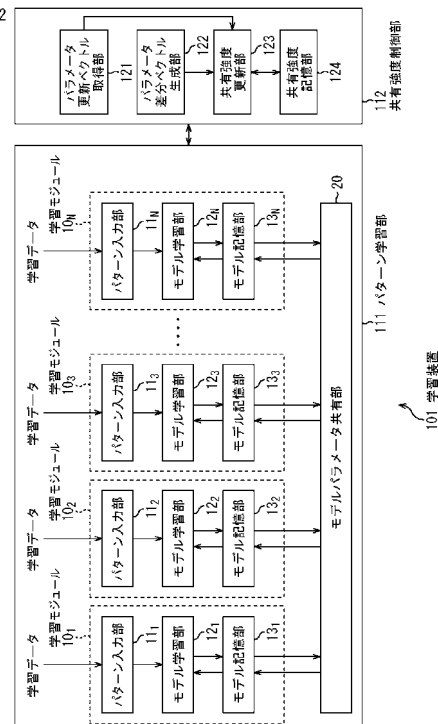
(57) 【要約】

【課題】規模拡張性があり、かつ、汎化特性を有するパターン学習モデルを得ることができるようにする。

【解決手段】学習モジュール10₁ないし10_Nは、入力データを用いて、パターン学習モデルの複数のモデルパラメータを更新する更新学習を行う。モデルパラメータ共有部20は、学習モジュール10₁ないし10_Nに、モデルパラメータを共有させる。共有強度制御部112は、学習モジュール10₁ないし10_Nそれぞれについて、学習モジュール10_jから学習モジュール10_iへの影響の強さを表す共有強度β_{ij}を、更新学習により複数のモデルパラメータを更新したとき学習モジュール10_iの学習誤差E_iを最小化するように更新する。本発明は、例えば、時系列パターンの学習等に適用できる。

【選択図】図12

図12



【特許請求の範囲】**【請求項 1】**

パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行う複数の学習モジュールと、

前記複数の学習モジュールの 2 以上の学習モジュールに、前記モデルパラメータを共有させるモデルパラメータ共有手段と、

前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新する共有強度更新手段とを備える学習装置。

【請求項 2】

前記共有強度更新手段は、前記複数の学習モジュールのうちの第 1 の学習モジュールから第 2 の学習モジュールへの影響の強さを表す前記共有強度を更新する場合、前記第 2 の学習モジュールの学習時のパラメータ更新ベクトルと、前記第 1 の学習モジュールのモデルパラメータから前記第 2 の学習モジュールのモデルパラメータを引き算することにより得られるモデルパラメータ差分ベクトルとの内積により、共有強度を更新する

請求項 1 に記載の学習装置。

【請求項 3】

前記パターン学習モデルは、時系列パターン、又はダイナミクスを学習するモデルである

請求項 1 に記載の学習装置。

【請求項 4】

前記パターン学習モデルは、HMM、RNN、FNN、SVR、又はRNNPBである

請求項 1 に記載の学習装置。

【請求項 5】

前記モデルパラメータ共有手段は、複数の学習モジュールのすべて、又は一部に、前記モデルパラメータを共有させる

請求項 1 に記載の学習装置。

【請求項 6】

前記モデルパラメータ共有手段は、前記複数の学習モジュールの 2 以上の学習モジュールに、前記複数のモデルパラメータのすべて、又は一部を共有させる

請求項 1 に記載の学習装置。

【請求項 7】

前記モデルパラメータ共有手段は、前記 2 以上の学習モジュールの各学習モジュールが更新したモデルパラメータを、前記 2 以上の学習モジュールそれぞれが更新したモデルパラメータの重み付け平均値によって補正することにより、前記 2 以上の学習モジュールに、その 2 以上の学習モジュールそれぞれが更新したモデルパラメータを共有させる

請求項 1 に記載の学習装置。

【請求項 8】

複数の学習モジュールそれぞれにおいて、パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行い、

前記複数の学習モジュールの 2 以上の学習モジュールに、前記モデルパラメータを共有させ、

前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新する

ステップを含む学習方法。

【請求項 9】

コンピュータを、

パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行う複数の学習モジュールと、

前記複数の学習モジュールの 2 以上の学習モジュールに、前記モデルパラメータを共有

10

20

30

40

50

させるモデルパラメータ共有手段と、

前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新する共有強度更新手段と

して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、学習装置、学習方法、およびプログラムに関し、特に、規模拡張性があり、かつ、汎化特性を有するパターン学習モデルを得ることができるようにする学習装置、学習方法、およびプログラムに関する。

10

【背景技術】

【0002】

パターンを学習するパターン学習モデルとしては、例えば、RNN(Recurrent Neural Network)やRNNPB(Recurrent Neural Net with Parametric Bias)等がある。このようなパターン学習モデルの学習のスキームは、「局所表現」スキームと「分散表現」スキームとに分類される。

【0003】

「局所表現」スキームでは、複数のパターンが、パターン学習モデルの学習（パターン学習モデルが有するモデルパラメータの更新）を行う複数の学習モジュールにおいてそれぞれ学習され、これにより、1つの学習モジュールで、1つのパターンが記憶される。

20

【0004】

また、「分散表現」スキームでは、複数のパターンが、1つの学習モジュールにおいて学習され、これにより、1つの学習モジュールで、複数のパターンが同時に記憶される。

【0005】

「局所表現」スキームでは、1つの学習モジュールで、1つのパターンが記憶される、つまり、1つのパターン学習モデルが、1つのパターンを学習するので、ある学習モジュールと、他の学習モジュールとの間で、パターンの記憶の干渉が少なく、パターンの記憶の安定性が高い。そして、「局所表現」スキームは、学習モジュールを追加することにより、新たなパターンを学習することが容易に行うことができるという規模拡張性に優れる。

30

【0006】

但し、「局所表現」スキームでは、1つのパターン学習モデルが、1つのパターンを学習する、つまり、パターンの記憶が、複数の学習モジュールでそれぞれ独立に行われるため、複数の学習モジュールそれぞれでのパターンの記憶間の関係性の構造化（共通化）による汎化特性を得ること、すなわち、例えば、ある学習モジュールに記憶されたパターンとも異なり、他の学習モジュールに記憶されたパターンとも異なるが、それらの両方のパターンの、いわば中間のパターンを生成するようなこと等はできない。

【0007】

一方、「分散表現」スキームでは、1つの学習モジュールで、複数のパターンが記憶される、つまり、1つのパターン学習モデルが、複数のパターンを学習するので、1つの学習モジュール内での複数のパターンの記憶間の干渉によって、その記憶間の共通化による汎化特性を得ることができる。

40

【0008】

但し、「分散表現」スキームでは、パターンの記憶の安定性が低いため、規模拡張性がない。

【0009】

ここで、特許文献1には、あるパターンを学習するRNNと、そのパターンと相関がある他のパターンを学習するRNNとの2つのRNNのコンテキストどうしの誤差に基づいて、その2つのRNNのコンテキストを変化させて、RNNの学習を行うこと、及び、その学習後の2つのRNNのうちの一方向のコンテキストを、他方のRNNのコンテキストとして用いて、つまり、

50

一方のRNNのコンテキストを、他方のRNNのコンテキストに影響させて、出力データを生成すること（RNNの入力層に入力データを入力し、その入力データに対応する出力データを、RNNの出力層から出力すること）が記載されている。

【0010】

また、非特許文献1には、言語のパターンを学習するRNNPBと、動作のパターンを学習するRNNPBとの2つのRNNPBのPBどうしの差に基づいて、その2つのRNNPBをPBを変化させて、RNNPBの学習を行うこと、及び、その学習後の2つのRNNPBのうちの一方のPBを、他方のPBに影響させて、出力データを生成することが記載されている。

【0011】

【特許文献1】特開2002-024795号公報

【非特許文献1】Yuuya Sugita, Jun Tani, "Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes", Adaptive Behavior, Vol. 13, No. 1, 33-52 (2005)

【発明の開示】

【発明が解決しようとする課題】

【0012】

以上のように、従来のパターン学習モデルの学習では、規模拡張性があるパターン学習モデルを得るか、又は汎化特性を有するパターン学習モデルを得ることはできるが、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることは困難である。

【0013】

本発明は、このような状況に鑑みてなされたものであり、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることができるようにするものである。

【課題を解決するための手段】

【0014】

本発明の一側面の学習装置は、パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行う複数の学習モジュールと、前記複数の学習モジュールの2以上の学習モジュールに、前記モデルパラメータを共有させるモデルパラメータ共有手段と、前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新する共有強度更新手段とを備える。

【0015】

本発明の一側面の学習方法は、複数の学習モジュールそれぞれにおいて、パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行い、前記複数の学習モジュールの2以上の学習モジュールに、前記モデルパラメータを共有させ、前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新するステップを含む学習方法である。

【0016】

本発明の一側面のプログラムは、コンピュータを、パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習を行う複数の学習モジュールと、前記複数の学習モジュールの2以上の学習モジュールに、前記モデルパラメータを共有させるモデルパラメータ共有手段と、前記更新学習により前記複数のモデルパラメータを更新したときの学習誤差が最小化されるように、前記学習モジュール間の共有強度を更新する共有強度更新手段として機能させる。

【0017】

本発明の一側面においては、複数の学習モジュールそれぞれにおいて、パターンを学習するパターン学習モデルが有する複数のモデルパラメータを、入力データを用いて更新する更新学習が行われ、複数の学習モジュールの2以上の学習モジュールにおいて、モデルパラメータが共有される。また、更新学習により複数のモデルパラメータを更新したときの学習誤差が最小化されるように、学習モジュール間の共有強度が更新される。

【発明の効果】

【0018】

本発明の一側面によれば、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることができる。

【発明を実施するための最良の形態】

【0019】

図1は、本発明を適用する学習装置の基本となる学習装置の一実施の形態の構成例を示している。

【0020】

図1において、学習装置は、複数であるN個の学習モジュール 10_1 ないし 10_N と、モデルパラメータ共有部20とから構成される。

10

【0021】

学習モジュール 10_i ($i=1, 2, \dots, N$)は、パターン入力部 11_i 、モデル学習部 12_i 、及びモデル記憶部 13_i から構成され、入力データを用いて、パターン学習モデルの複数のモデルパラメータ（学習リソース）を更新する更新学習を行う。

【0022】

すなわち、パターン入力部 11_i には、モデル記憶部 13_i に記憶されたパターン学習モデルに獲得（学習）させるパターン（カテゴリ）の入力データが、パターン学習モデルの学習に用いる学習データとして供給される。

【0023】

20

パターン入力部 11_i は、そこに供給される学習データを、パターン学習モデルの学習に適切な形のデータにする処理をして、モデル学習部 12_i に供給する。すなわち、例えば、学習データが時系列のデータである場合には、パターン入力部 11_i は、例えば、その時系列のデータを、固定の長さに区切って、モデル学習部 12_i に供給する。

【0024】

モデル学習部 12_i は、パターン入力部 11_i からの学習データを用いて、モデル記憶部 13_i に記憶されたパターン学習モデルの複数のモデルパラメータを更新する更新学習を行う。

【0025】

モデル記憶部 13_i は、複数のモデルパラメータを有し、パターンを学習するパターン学習モデルを記憶する。すなわち、モデル記憶部 13_i は、パターン学習モデルの複数のモデルパラメータを記憶する。

30

【0026】

ここで、パターン学習モデルとしては、例えば、時系列のパターンである時系列パターンや、時間変化する力学系を表すダイナミクスを学習（獲得）（記憶）するモデル等を採用することができる。

【0027】

時系列パターンを学習するモデルとしては、例えば、HMM(Hidden Markov Model)等があり、ダイナミクスを学習するモデルとしては、例えば、RNN, FNN(Feed Forward Neural Network), RNNPB等のニューラルネットワークや、SVR(Support Vector Regression)等がある。

40

【0028】

例えば、HMMについては、HMMにおいて状態が遷移する確率を表す状態遷移確率や、状態が遷移するときに、HMMからある観測値が出力される確率を表す出力確率、又は確率密度を表す出力確率密度関数が、HMMのモデルパラメータである。

【0029】

また、例えば、ニューラルネットワークについては、ニューロンに相当するユニット（ノード）において、他のユニットからの入力に付されるウェイト（重み）が、ニューラルネットワークのモデルパラメータである。

【0030】

50

なお、HMMの状態遷移確率や、出力確率、又は出力確率密度関数、ニューラルネットワークのウェイトは、いずれも複数存在する。

【0031】

モデルパラメータ共有部20は、N個の学習モジュール10₁ないし10_Nのうちの、2以上の学習モジュールに、モデルパラメータを共有させる共有処理を行う。モデルパラメータ共有部20が共有処理を行うことにより、N個の学習モジュール10₁ないし10_Nのうちの、2以上の学習モジュールは、モデルパラメータを共有する。

【0032】

なお、以下では、説明を簡単にするため、モデルパラメータ共有部20は、N個の学習モジュール10₁ないし10_Nのすべてに、モデルパラメータを共有させる共有処理を行うこととする。

10

【0033】

次に、図2のフローチャートを参照して、図1の学習装置が行う、パターン学習モデルを学習する学習処理について説明する。

【0034】

ステップS11において、学習モジュール10_iのモデル学習部12_iは、モデル記憶部13_iに記憶されたモデルパラメータを、例えば、乱数等によって初期化して、処理は、ステップS12に進む。

【0035】

ステップS12では、学習モジュール10_iが、その学習モジュール10_iで学習すべき学習データが供給（入力）されるのを待って、その学習データを用いて、モデルパラメータを更新する更新学習を行う。

20

【0036】

すなわち、ステップS12では、学習モジュール10_iにおいて、パターン入力部11_iが、学習モジュール10_iに供給された学習データを、必要に応じて処理し、モデル学習部12_iに供給する。

【0037】

さらに、ステップS12では、モデル学習部12_iが、パターン入力部11_iからの学習データを用いて、モデル記憶部13_iに記憶されたパターン学習モデルの複数のモデルパラメータを更新する更新学習を行い、その更新学習によって得られた新たな複数のモデルパラメータによって、モデル記憶部13_iの記憶内容を更新する（上書きする）。

30

【0038】

ここで、ステップS11及びS12の処理は、N個の学習モジュール10₁ないし10_Nのすべてで行われる。

【0039】

ステップS12の後、処理は、ステップS13に進み、モデルパラメータ共有部20は、N個の学習モジュール10₁ないし10_Nのすべてに、モデルパラメータを共有させる共有処理を行う。

【0040】

すなわち、学習モジュール10_iが有する複数のモデルパラメータのうちの、例えば、m番目のモデルパラメータに注目すると、モデルパラメータ共有部20は、N個の学習モジュール10₁ないし10_Nそれぞれのm番目のモデルパラメータに基づいて、学習モジュール10₁のm番目のモデルパラメータを補正する。

40

【0041】

さらに、モデルパラメータ共有部20は、N個の学習モジュール10₁ないし10_Nそれぞれのm番目のモデルパラメータに基づいて、学習モジュール10₂のm番目のモデルパラメータを補正し、以下、同様にして、学習モジュール10₃ないし10_Nそれぞれのm番目のモデルパラメータを補正する。

【0042】

以上のように、モデルパラメータ共有部20が、学習モジュール10_iのm番目のモデル

50

パラメータを、N個の学習モジュール 10_1 ないし 10_N それぞれの m 番目のモデルパラメータに基づいて補正することで、N個の学習モジュール 10_1 ないし 10_N の m 番目のモデルパラメータのそれぞれは、N個の学習モジュール 10_1 ないし 10_N の m 番目のモデルパラメータのすべての影響を受ける (N個の学習モジュール 10_1 ないし 10_N の m 番目のモデルパラメータのそれぞれに、N個の学習モジュール 10_1 ないし 10_N の m 番目のモデルパラメータのすべてに影響させる)。

【0043】

このように、複数の学習モジュールのモデルパラメータすべてを、その複数の学習モジュールのモデルパラメータのそれぞれに影響させること (複数の学習モジュールのモデルパラメータのそれぞれが、その複数の学習モジュールのモデルパラメータすべての影響を受けること) が、複数の学習モジュールによるモデルパラメータの共有である。

10

【0044】

モデルパラメータ共有部 20 は、ステップ S 13 において、学習モジュール 10_i のモデル記憶部 13_i に記憶された複数のモデルパラメータのすべてを対象に、共有処理を行い、その共有処理によって得られたモデルパラメータによって、モデル記憶部 13_1 ないし 13_N の記憶内容を更新する。

【0045】

ステップ S 13 の後、処理は、ステップ S 14 に進み、図 1 の学習装置は、学習の終了条件が満たされているかどうかを判定する。

【0046】

ここで、ステップ S 14 での学習の終了条件としては、例えば、学習の回数、つまり、ステップ S 12 及び S 13 が繰り返された回数が、あらかじめ定められた所定の回数となったことや、あらかじめ用意された学習データのすべてを用いて、ステップ S 12 の更新学習が行われたこと、あるいは、ある入力データに対して出力されるべき出力データの真値が分かっている場合に、その入力データに対してパターン学習モデルから出力される出力データの、真値に対する誤差が所定値以下であること、等を採用することができる。

20

【0047】

ステップ S 14 において、学習の終了条件が満たされていないと判定された場合、処理は、ステップ S 12 に戻り、以下、同様の処理が繰り返される。

【0048】

また、ステップ S 14 において、学習の終了条件が満たされていると判定された場合、処理は終了する。

30

【0049】

なお、ステップ S 12 とステップ S 13 は、その処理の順番を逆にすることも可能である。即ち、N個の学習モジュール 10_1 ないし 10_N のすべてに、モデルパラメータを共有させる共有処理を行った後、モデルパラメータを更新する更新学習を行うようにすることもできる。

【0050】

次に、図 3 は、パターン学習モデルとして、RNNPB を採用した場合の、図 1 の学習装置の構成例を示している。

40

【0051】

なお、図 3 においては、学習モジュール 10_i のパターン入力部 11_i 及びモデル学習部 12_i の図示を省略してある。

【0052】

モデル記憶部 13_i には、RNNPB (を定義するモデルパラメータ) が記憶されている。ここで、モデル記憶部 13_i に記憶された RNNPB を、以下、適宜、RNNPB#i と記載する。

【0053】

RNNPB は、入力層、隠れ層 (中間層)、及び出力層により構成されている。入力層、隠れ層、及び出力層は、それぞれ任意の数の、ニューロンに相当するユニットにより構成されている。

50

【 0 0 5 4 】

RNNPBでは、入力層の一部のユニットである入力ユニットに、時系列データ等の入力データ x_t が入力（供給）される。ここで、入力データ x_t としては、例えば、画像や音声の特徴量や、ロボットの手や足に相当する部分の動きの軌道等を採用することができる。

【 0 0 5 5 】

また、入力層の、入力データ x_t が入力される入力ユニット以外のユニットの一部であるPBユニットには、PB(Parametric Bias)が入力される。PBによれば、同一の状態のRNNPBに対して、同一の入力データ x_t が入力されても、PBを変更することにより、異なる出力データ x_{t+1}^* を得ることができる。

【 0 0 5 6 】

入力層の、入力データ x_t が入力される入力ユニット以外のユニットの残りであるコンテキストユニットには、出力層の一部のユニットより出力される出力データが、内部状態を表すコンテキストとしてフィードバックされる。

【 0 0 5 7 】

ここで、時刻 t の入力データ x_t が入力層の入力ユニットに入力されるときに入力層のPBユニットとコンテキストユニットに入力される時刻 t のPBとコンテキストを、それぞれ、 P_{B_t} と c_t と記載する。

【 0 0 5 8 】

隠れ層のユニットは、入力層に入力される入力データ x_t 、 P_{B_t} 、コンテキスト c_t を対象として、所定のウェイト（重み）を用いた重み付け加算を行い、その重み付け加算の結果を引数とする非線形関数の演算を行って、その演算結果を、出力層のユニットに出力する。

【 0 0 5 9 】

出力層の一部のユニットからは、上述したように、次の時刻 $t+1$ のコンテキスト c_{t+1} となる出力データが出力され、入力層にフィードバックされる。また、出力層の残りのユニットからは、例えば、入力データ x_t に対応する出力データとして、その入力データ x_t の次の時刻 $t+1$ の入力データ x_{t+1} の予測値 x_{t+1}^* が出力される。

【 0 0 6 0 】

ここで、RNNPBでは、ユニットへの入力が重み付け加算されるが、この重み付け加算に用いられるウェイト（重み）が、RNNPBのモデルパラメータである。RNNPBのモデルパラメータとしてのウェイトには、入力ユニットから隠れ層のユニットへのウェイト、PBユニットから隠れ層のユニットへのウェイト、コンテキストユニットから隠れ層のユニットへのウェイト、隠れ層のユニットから出力層のユニットへのウェイト、及び、隠れ層のユニットからコンテキストユニットへのウェイトの5種類がある。

【 0 0 6 1 】

パターン学習モデルとして、以上のようなRNNPBを採用した場合、モデルパラメータ共有部20には、RNNPBのモデルパラメータとしてのウェイトを、学習モジュール10₁ないし10_Nに共有させるウェイトマトリクス共有部21が設けられる。

【 0 0 6 2 】

ここで、RNNPBのモデルパラメータとしてのウェイトは、複数あるが、その複数のウェイトをコンポーネントとするマトリクスを、ウェイトマトリクスという。

【 0 0 6 3 】

ウェイトマトリクス共有部21は、モデル記憶部13₁ないし13_Nに記憶されたRNNPB#1ないしRNNPB#Nの複数のモデルパラメータとしてのウェイトマトリクスすべてを、学習モジュール10₁ないし10_Nのそれぞれに共有させる。

【 0 0 6 4 】

すなわち、RNNPB# i のウェイトマトリクスを w_i と表すこととすると、ウェイトマトリクス共有部21は、ウェイトマトリクス w_i を、 N 個の学習モジュール10₁ないし10_Nそれぞれのウェイトマトリクス w_1 ないし w_N のすべてに基づいて補正することで、ウェイトマトリクス w_i に、ウェイトマトリクス w_1 ないし w_N のすべてを影響させる共有処理を行う。

10

20

30

40

50

【 0 0 6 5 】

具体的には、ウエイトマトリクス共有部 2 1 は、例えば、次式 (1) に従い、RNNPB#i のウエイトマトリクス w_i を補正する。

【 0 0 6 6 】

【 数 1 】

$$w_i = w_i + \Delta w_i \quad \dots (1)$$

【 0 0 6 7 】

ここで、式 (1) において、 w_i は、ウエイトマトリクス w_i を補正する補正成分であり、例えば、式 (2) に従って求められる。

10

【 0 0 6 8 】

【 数 2 】

$$\Delta w_i = \alpha_i \sum_{j=1}^N \beta_{ij} (w_j - w_i) \quad \dots (2)$$

【 0 0 6 9 】

式 (2) において、 β_{ij} は、RNNPB#i のウエイトマトリクス w_i に、RNNPB#j (j=1, 2, ..., N) のウエイトマトリクス w_j を影響させる度合いを表す係数 (固定値) である。

【 0 0 7 0 】

したがって、式 (2) の右辺のサメーション $\sum \beta_{ij} (w_j - w_i)$ は、係数 β_{ij} を重みとした、RNNPB#i のウエイトマトリクス w_i に対する RNNPB#1 ないし RNNPB#N のウエイトマトリクス w_1 ないし w_N それぞれの偏差 (差分) の重み付け平均値を表し、 α_i は、その重み付け平均値 $\sum \beta_{ij} (w_j - w_i)$ を、ウエイトマトリクス w_i に影響させる度合いを表す係数である。

20

【 0 0 7 1 】

係数 α_i 及び β_{ij} としては、例えば、0.0 より大で 1.0 より小の値を採用することができる。

【 0 0 7 2 】

式 (2) によれば、係数 α_i が小であるほど、いわば共有が弱くなり (ウエイトマトリクス w_i が受ける重み付け平均値 $\sum \beta_{ij} (w_j - w_i)$ の影響が小さくなり)、係数 α_i が大であるほど、いわば共有が強まる。

30

【 0 0 7 3 】

なお、ウエイトマトリクス w_i の補正の方法は、式 (1) に限定されるものではなく、例えば、式 (3) に従って行うことが可能である。

【 0 0 7 4 】

【 数 3 】

$$w_i = \alpha'_i \cdot w_i + (1 - \alpha'_i) \cdot \sum_{j=1}^N \beta'_{ij} \cdot w_j \quad \dots (3)$$

【 0 0 7 5 】

ここで、式 (3) において、 β'_{ij} は、RNNPB#i のウエイトマトリクス w_i に、RNNPB#j (j=1, 2, ..., N) のウエイトマトリクス w_j を影響させる度合いを表す係数である。

40

【 0 0 7 6 】

したがって、式 (3) の右辺の第 2 項におけるサメーション $\sum \beta'_{ij} \cdot w_j$ は、係数 β'_{ij} を重みとした、RNNPB#1 ないし RNNPB#N のウエイトマトリクス w_1 ないし w_N の重み付け平均値を表し、 α'_i は、その重み付け平均値 $\sum \beta'_{ij} \cdot w_j$ を、ウエイトマトリクス w_i に影響させる度合いを表す係数である。

【 0 0 7 7 】

係数 α'_i 及び β'_{ij} としては、例えば、0.0 より大で 1.0 より小の値を採用することができる。

【 0 0 7 8 】

50

式(3)によれば、係数 α_{ij} が大であるほど、共有が弱くなり(ウエイトマトリクス w_{ij} が受ける重み付け平均値 $\sum \beta_{ij} w_{ij}$ の影響が小さくなり)、係数 α_{ij} が小であるほど、共有が強まる。

【0079】

次に、図4のフローチャートを参照して、パターン学習モデルとして、RNNPBを採用した場合の、図1の学習装置の学習処理について説明する。

【0080】

ステップS21において、学習モジュール10_iのモデル学習部12_iは、モデル記憶部13_iに記憶されたRNNPB#iのモデルパラメータであるウエイトマトリクス w_i を、例えば、乱数等によって初期化して、処理は、ステップS22に進む。

10

【0081】

ステップS22では、学習モジュール10_iが、その学習モジュール10_iで学習すべき学習データ x_t が入力されるのを待って、その学習データ x_t を用いて、モデルパラメータを更新する更新学習を行う。

【0082】

すなわち、ステップS22では、学習モジュール10_iにおいて、パターン入力部11_iが、学習モジュール10_iに供給された学習データ x_t を、必要に応じて処理し、モデル学習部12_iに供給する。

【0083】

さらに、ステップS22では、モデル学習部12_iが、パターン入力部11_iからの学習データ x_t を用いて、モデル記憶部13_iに記憶されたRNNPB#iのウエイトマトリクス w_i を更新する更新学習を、例えば、BPTT(Back-Propagation Through Time)法により行い、その更新学習によって得られた新たなモデルパラメータとしてのウエイトマトリクス w_i によって、モデル記憶部13_iの記憶内容を更新する。

20

【0084】

ここで、ステップS21及びS22の処理は、N個の学習モジュール10₁ないし10_Nのすべてで行われる。

【0085】

また、BPTT法については、例えば、特開2002-236904号公報等に記載されている。

【0086】

30

ステップS22の後、処理は、ステップS23に進み、モデルパラメータ共有部20のウエイトマトリクス共有部21は、N個の学習モジュール10₁ないし10_Nのすべてに、ウエイトマトリクス w_1 ないし w_N のすべてを共有させる共有処理を行う。

【0087】

すなわち、ステップS23において、ウエイトマトリクス共有部21は、例えば、式(2)に従い、モデル記憶部13₁ないし13_Nに記憶されたウエイトマトリクス w_1 ないし w_N を用いて補正成分 w_1 ないし w_N をそれぞれ求め、その補正成分 w_1 ないし w_N により、モデル記憶部13₁ないし13_Nに記憶されたウエイトマトリクス w_1 ないし w_N を、式(1)に従ってそれぞれ補正する。

【0088】

40

ステップS23の後、処理は、ステップS24に進み、図1の学習装置は、学習の終了条件が満たされているかどうかを判定する。

【0089】

ここで、ステップS24での学習の終了条件としては、例えば、学習の回数、つまり、ステップS22及びS23が繰り返された回数が、あらかじめ定められた所定の回数となったことや、ある入力データ x_t に対してRNNPB#iが出力する出力データ x_{t+1}^* 、すなわち、入力データ x_{t+1} の予測値 x_{t+1}^* の、入力データ x_{t+1} に対する誤差が所定値以下であること、等を採用することができる。

【0090】

ステップS24において、学習の終了条件が満たされていないと判定された場合、処理

50

は、ステップ S 2 2 に戻り、以下、同様の処理、すなわち、ウエイトマトリクス w_i の更新学習と、共有処理とが交互に繰り返される。

【 0 0 9 1 】

また、ステップ S 2 4 において、学習の終了条件が満たされていると判定された場合、処理は終了する。

【 0 0 9 2 】

なお、図 4 においても、ステップ S 2 2 とステップ S 2 3 は、その処理の順番を逆にすることが可能である。

【 0 0 9 3 】

以上のように、規模拡張性に優れた複数の学習モジュール 10_1 ないし 10_N それぞれにおいて、モデルパラメータを共有しながら、その複数の学習モジュール 10_1 ないし 10_N それぞれのモデルパラメータを更新する更新学習を行うことにより、1つの学習モジュールだけで行われる学習で得られる汎化特性が、複数の学習モジュール 10_1 ないし 10_N の全体で得ることができ、その結果、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることができる。

10

【 0 0 9 4 】

すなわち、多くのパターンを獲得（記憶）することができ、かつ、複数のパターンの共通性を獲得することができる。さらに、複数のパターンの共通性を獲得することで、その共通性に基づいて、未学習のパターンの認識や生成を行うことが可能となる。

【 0 0 9 5 】

具体的には、学習データとして、例えば、N種類の音韻の音声データを、N個の学習モジュール 10_1 ないし 10_N にそれぞれ与えて、パターン学習モデルの学習を行った場合に、そのパターン学習モデルによれば、学習に用いられていない時系列パターンの音声データの認識や生成を行うことができる。さらに、例えば、学習データとして、例えば、ロボットのアームを駆動するための、N種類の駆動データを、N個の学習モジュール 10_1 ないし 10_N にそれぞれ与えて、パターン学習モデルの学習を行った場合に、そのパターン学習モデルによれば、学習に用いられていない時系列パターンの駆動データの生成を行うことができ、その結果、ロボットは、教えられていないアームの動きをすることが可能となる。

20

【 0 0 9 6 】

また、学習後のパターン学習モデルによれば、パターン学習モデル間のモデルパラメータ（リソース）どうしの距離に基づいて、パターン学習モデルどうしの類似性を評価し、類似性が高いパターン学習モデルどうしをクラスタとして、パターンのクラスタリングを行うことができる。

30

【 0 0 9 7 】

次に、図 5 ないし図 9 を参照して、本件発明者が行った、図 1 の学習装置による学習処理（以下、適宜、共有学習処理という）のシミュレーションの結果について説明する。

【 0 0 9 8 】

図 5 は、共有学習処理で学習を行ったパターン学習モデルについての各データを示している。

40

【 0 0 9 9 】

なお、シミュレーションでは、パターン学習モデルとして、2つのPBが入力層に入力され、3つのコンテキストが入力層にフィードバックされる9個のRNNPB#1ないしRNNPB#9を採用し、学習データとして、3つのパターンP#1,P#2,P#3の時系列データのそれぞれに、異なる3つのノイズN#1,N#2,N#3のそれぞれを重畳して得られる9個の時系列データを用いた。

【 0 1 0 0 】

また、RNNPB#1には、パターンP#1の時系列データにノイズN#1を重畳して得られる時系列データを、RNNPB#2には、パターンP#1の時系列データにノイズN#2を重畳して得られる時系列データを、RNNPB#2には、パターンP#1の時系列データにノイズN#3を重畳して得ら

50

れる時系列データを、それぞれ、学習データとして与えた。

【 0 1 0 1 】

同様に、RNNPB#4には、パターンP#2の時系列データにノイズN#1を重畳して得られる時系列データを、RNNPB#5には、パターンP#2の時系列データにノイズN#2を重畳して得られる時系列データを、RNNPB#6には、パターンP#2の時系列データにノイズN#3を重畳して得られる時系列データを、それぞれ、学習データとして与え、RNNPB#7には、パターンP#3の時系列データにノイズN#1を重畳して得られる時系列データを、RNNPB#8には、パターンP#3の時系列データにノイズN#2を重畳して得られる時系列データを、RNNPB#9には、パターンP#3の時系列データにノイズN#3を重畳して得られる時系列データを、それぞれ、学習データとして与えた。

10

【 0 1 0 2 】

なお、更新学習は、入力データ x_t に対してRNNPBが出力する出力データとしての、入力データ x_{t+1} の予測値 x^*_{t+1} の、入力データ x_{t+1} に対する誤差（予測誤差）を小さくするように行った。

【 0 1 0 3 】

図 5 上から 1 番目は、学習後のRNNPB#1ないしRNNPB#9に、学習時に与えられた学習データを入力データとして与えたときに、RNNPB#1ないしRNNPB#9それぞれが出力する出力データ(output)と、その出力データの予測誤差(error)とを示している。

【 0 1 0 4 】

図 5 上から 1 番目において、予測誤差は、ほぼ0になっており、したがって、RNNPB#1ないしRNNPB#9は、入力データ、つまり、学習時に与えられた学習データとほぼ一致する出力データを出力する。

20

【 0 1 0 5 】

図 5 上から 2 番目は、学習後のRNNPB#1ないしRNNPB#9が、図 5 上から 1 番目に示した出力データを出力するときの、3つのコンテキストの時間変化を示している。

【 0 1 0 6 】

また、図 5 上から 3 番目は、学習後のRNNPB#1ないしRNNPB#9が、図 5 上から 1 番目に示した出力データを出力するときの、2つのPB（以下、適宜、2つのPBそれぞれを、PB#1、PB#2と記載する）の時間変化を示している。

【 0 1 0 7 】

図 6 は、学習後のRNNPB#1ないしRNNPB#9のうちの、例えば、5番目のRNNPB#5が、各値のPB#1、PB#2に対して出力する出力データを示している。

30

【 0 1 0 8 】

なお、図 6 では、横軸がPB#1を表し、縦軸がPB#2を表している。

【 0 1 0 9 】

図 6 によれば、RNNPB#5は、PB#1が0.6程度のときに、学習時に与えられた学習データとほぼ一致する出力データを出力しており、これにより、RNNPB#5は、学習時に与えられた学習データのパターンP#2を獲得していることが分かる。

【 0 1 1 0 】

また、RNNPB#5は、PB#1が0.6より小さいときに、RNNPB#1ないしRNNPB#3に学習させたパターンP#1や、RNNPB#7ないしRNNPB#9に学習させたパターンP#3に類似する時系列データを出力しており、これにより、RNNPB#5が、RNNPB#1ないしRNNPB#3が獲得したパターンP#1や、RNNPB#7ないしRNNPB#9が獲得したパターンP#3の影響を受け、いわば、RNNPB#5に対して学習時に与えられた学習データのパターンP#2が、RNNPB#1ないしRNNPB#3が獲得したパターンP#1や、RNNPB#7ないしRNNPB#9が獲得したパターンP#3の方向に変形していくときに現れる中間のパターンをも獲得していることが分かる。

40

【 0 1 1 1 】

さらに、RNNPB#5は、PB#1が0.6より大であるときに、9個のRNNPB#1ないしRNNPB#9のいずれにも学習させていないパターンの時系列データを出力しており、これにより、RNNPB#5が、RNNPB#1ないしRNNPB#3が獲得したパターンP#1や、RNNPB#7ないしRNNPB#9が獲得した

50

パターンP#3の影響を受け、いわば、RNNPB#5に対して学習時に与えられた学習データのパターンP#2が、RNNPB#1ないしRNNPB#3が獲得したパターンP#1や、RNNPB#7ないしRNNPB#9が獲得したパターンP#3の方向とは逆方向に変形していくときに現れるパターンをも獲得していることが分かる。

【0112】

次に、図7は、9個のRNNPB#1ないしRNNPB#9それぞれのウェイトマトリクスどうしの相関としての距離、すなわち、例えば、ウェイトマトリクスを構成する各ウェイトをコンポーネントとするベクトルの空間における、そのベクトルどうしの距離を表す長方形のマップを示している。

【0113】

なお、ウェイトマトリクスどうしの距離が小であるほど、その2つのウェイトマトリクスどうしの相関が高い。

【0114】

図7のマップでは、横軸と縦軸のそれぞれに、9個のRNNPB#1ないしRNNPB#9それぞれのウェイトマトリクスをとって、その横軸のウェイトマトリクスと、縦軸のウェイトマトリクスとの距離が、濃淡で示されており、濃い（黒い）部分ほど、距離が小であることを表す（淡い（白い）部分ほど、距離が大であることを表す）。

【0115】

図7において、横×縦が5×3個のマップのうちの、左上のマップは、学習の回数が0回目であるときのウェイトマトリクスどうしの距離、すなわち、初期化がされたウェイトマトリクスどうしの距離を示しており、マップにおいて、対角線上に並ぶ、同一のRNNPB#iのウェイトマトリクスどうしの距離だけが小になっている。

【0116】

以下、図7では、右に行くほど、そして、下に行くほど、学習が進行したときのマップが示されており、右下のマップが、学習の回数が1400回目であるときのウェイトマトリクスどうしの距離を示している。

【0117】

図7によれば、学習が進行するにつれ、同一のパターンP#1の時系列データを学習したRNNPB#1ないしRNNPB#3のウェイトマトリクスどうしの距離、同一のパターンP#2の時系列データを学習したRNNPB#4ないしRNNPB#6のウェイトマトリクスどうしの距離、及び、同一のパターンP#3の時系列データを学習したRNNPB#7ないしRNNPB#9のウェイトマトリクスどうしの距離が小さくなることが分かる。

【0118】

図8は、図5ないし図7の場合とは別の時系列データを用いて学習を行ったRNNPBのウェイトマトリクスどうしの相関としての距離を表す、図7と同様のマップを示している。

【0119】

なお、図8のマップを作成するシミュレーションでは、図9に示す5種類のパターンP#1, P#2, P#3, P#4, P#5の時系列データのそれぞれに、異なる4つのノイズN#1, N#2, N#3, N#4のそれぞれを重畳して得られる20個の時系列データを用意し、各時系列データを、1つのRNNPBに学習させた。したがって、図8のマップを作成するシミュレーションで用いたRNNPBは、20個のRNNPB#1ないしRNNPB#20である。

【0120】

また、学習では、RNNPB#1ないしRNNPB#4には、パターンP#1の時系列データを、RNNPB#5ないしRNNPB#8には、パターンP#2の時系列データを、RNNPB#9ないしRNNPB#12には、パターンP#3の時系列データを、RNNPB#13ないしRNNPB#16には、パターンP#4の時系列データを、RNNPB#17ないしRNNPB#20には、パターンP#5の時系列データを、それぞれ与えた。

【0121】

図8左の5×3個のマップは、共有が弱い場合、すなわち、20個のRNNPB#1ないしRNNPB#20のウェイトマトリクス w_1 ないし w_{20} それぞれに、その20個のウェイトマトリクス w_1 ないし w_{20} のすべてに影響させる度合いが小さい場合、具体的には、式(2)の係数 α_i が小

10

20

30

40

50

である場合 (α_i がほぼ0である場合) のマップを示している。

【0122】

また、図8右の 5×3 個のマップは、共有が強い場合、すなわち、20個のRNNPB#1ないしRNNPB#20のウェイトマトリクス w_1 ないし w_{20} それぞれに、その20個のウェイトマトリクス w_1 ないし w_{20} のすべてを影響させる度合いが大きい場合、具体的には、式(1)の係数 α_i が小でない場合のマップを示している。

【0123】

共有が弱い場合も強い場合も、学習の回数が0回目であるときの左上のマップでは、対角線上に並ぶ、同一のRNNPB#iのウェイトマトリクスどうしの距離だけが小になっている。

10

【0124】

そして、共有が弱い場合には、図8左に示すように、学習が進行しても、ウェイトマトリクスどうしの距離に、特に傾向は現れないが、共有が強い場合には、図8右に示すように、同一のパターンの時系列データを学習したRNNPBの間で、ウェイトマトリクスどうしの距離が小さくなることが分かる。

【0125】

したがって、共有処理によって、複数の学習モジュールをまたいで分散表現が形成され、複数のRNNPBが汎化特性を有するようになっていくことが分かる。

【0126】

なお、モデル学習部12_iによるモデルパラメータの更新学習の方法、及び、モデルパラメータ共有部20による共有処理の方法は、上述した方法に限定されるものではない。

20

【0127】

また、本実施の形態では、モデルパラメータ共有部20による共有処理において、N個の学習モジュール10₁ないし10_Nのすべてに、モデルパラメータとしてのウェイトマトリクスを共有させるようにしたが、その他、例えば、N個の学習モジュール10₁ないし10_Nのうちの一部だけに、モデルパラメータとしてのウェイトマトリクスを共有させることが可能である。

【0128】

さらに、本実施の形態では、モデルパラメータ共有部20による共有処理において、学習モジュール10_iに、複数のモデルパラメータとしての、ウェイトマトリクスを構成する複数のウェイトすべてを共有させるようにしたが、共有処理では、ウェイトマトリクスを構成する複数のウェイトすべてではなく、そのうちの一部のウェイトだけを共有させるようにすることが可能である。

30

【0129】

また、N個の学習モジュール10₁ないし10_Nのうちの一部だけに、ウェイトマトリクスを構成する複数のウェイトのうちの一部のウェイトだけを共有させることも可能である。

【0130】

なお、図1の学習装置は、モデルパラメータ共有部20が、複数の学習モジュール10₁ないし10_Nに、モデルパラメータを共有させる、つまり、各学習モジュール10_iにおけるパターン学習モデルとしてのRNNPB#iのモデルパラメータであるウェイトマトリクス w_i に、学習モジュール10₁ないし10_NそれぞれにおけるRNNPB#1ないしRNNPB#Nのウェイトマトリクス w_1 ないし w_N を影響させる点で、RNNの学習時に、2つのRNNのコンテキストどうしの誤差に基づいて、その2つのRNNのコンテキストを変化させる、つまり、各RNNのコンテキストに、2つのRNNのコンテキストを影響させる特許文献1に記載の技術と共通する。

40

【0131】

しかしながら、図1の学習装置では、影響を受けるのが、モデルパラメータであるウェイトマトリクスである点で、モデルパラメータではなく、内部状態であるコンテキストが影響を受ける特許文献1に記載の技術と相違する。

50

【 0 1 3 2 】

すなわち、例えば、関数で表現されるパターン学習モデルを例にすれば、パターン学習モデルのモデルパラメータは、そのパターン学習モデルを表現する関数を定義する、学習によって求められる定数（例えば、入力が u と、出力が y と、内部状態が x と、それぞれ表されるシステムをモデル化する状態方程式 $y=Cx+Du$ 、及び $x'=Ax+Bu$ （ x' は x の微分を表す）を例にすれば、 A, B, C, D にあたる）であり、そもそも定数ではない内部状態（状態方程式の例では、内部状態 x ）とは異なる。

【 0 1 3 3 】

同様に、図 1 の学習装置は、各学習モジュール 10_i におけるパターン学習モデルとしての RNNPB# i のモデルパラメータであるウエイトマトリクス w_i に、学習モジュール 10_1 ないし 10_N それぞれにおける RNNPB#1 ないし RNNPB# N のウエイトマトリクス w_1 ないし w_N に影響させる点で、RNNPBの学習時に、2つのRNNPBのPBどうしの差に基づいて、その2つのRNNPBのPBを変化させる、つまり、各RNNPBのPBに、2つのRNNPBのPBに影響させる非特許文献 1 に記載の技術と共通する。

10

【 0 1 3 4 】

しかしながら、図 1 の学習装置では、影響を受けるのが、モデルパラメータであるウエイトマトリクスである点で、モデルパラメータではなく、内部状態である（あるいは、内部状態に相当する）PBが影響を受ける非特許文献 1 に記載の技術と相違する。

【 0 1 3 5 】

すなわち、上述したように、パターン学習モデルのモデルパラメータとは、パターン学習モデルを表現する関数を定義する、学習によって求められる定数であり、定数ではない内部状態とは異なる。

20

【 0 1 3 6 】

そして、モデルパラメータは、パターン学習モデルを表現する関数を定義する、学習によって求められる定数であるがゆえに、学習時は、学習をしようとするパターンに対応する値になるように更新（変更）されるが、出力データを生成するとき（パターン学習モデルとしてのRNNPBの入力層に入力データを入力し、その入力データに対応する出力データを、RNNPBの出力層から出力するとき）には、変更されない。

【 0 1 3 7 】

一方、特許文献 1 に記載の技術が対象としているコンテキスト、及び、非特許文献 1 に記載の技術が対象としているPBは、モデルパラメータとは異なる内部状態であるがゆえに、学習時は勿論、出力データを生成するときも変更される。

30

【 0 1 3 8 】

以上のように、図 1 の学習装置は、特許文献 1 及び非特許文献 1 に記載の技術のいずれとも相違し、その結果、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることができる。

【 0 1 3 9 】

すなわち、図 1 の学習装置では、例えば、図 10 に示すように、RNNPBなどのパターン学習モデルのモデルパラメータを共有させる。

【 0 1 4 0 】

その結果、図 1 の学習装置によれば、図 11 に示すように、規模拡張性に優れるが、汎化特性に欠ける「局所表現」スキームと、汎化特性があるが、規模拡張性に欠ける「分散表現」スキームとの2種類の学習の両方の長所を有する、いわば「中間表現」スキームの学習を行い、規模拡張性があり、同時に、汎化特性を有するパターン学習モデルを得ることができる。

40

【 0 1 4 1 】

ところで、上述した実施の形態においては、パターン学習モデルとしてRNNPBを採用した場合の係数 β_{ij} に相当する、学習モジュール間のモデルパラメータの共有の強度（以下、共有強度とも称する）を、ユーザが決定し、設定する必要があるが、この共有強度を、最適に設定することが難しいという問題がある。

50

【 0 1 4 2 】

すなわち、設定した共有強度が強すぎると、学習モジュール間の共通化が促進され汎化性能は高くなるが、各学習モジュールの独立性は弱くなり学習が進まない（学習に時間がかかる）。反対に、設定された共有強度が弱すぎると、学習モジュールの共通化が進まず、図 1 の学習装置の特徴の 1 つである汎化特性が弱くなり、各学習モジュールが独立に学習することと、さほど差がなくなってしまうという問題がある。

【 0 1 4 3 】

パターンの学習では、理想的には、類似するパターンを学習する学習モジュールどうしについては共有強度を強く設定し、類似しないパターンを学習する学習モジュールとの共有強度は弱く設定したい。しかしながら、パターン間の類似性は、学習する前にはわからないので、そのように設定することが難しいという側面がある。

10

【 0 1 4 4 】

そこで、次に、そのような問題を解決する学習装置について説明する。すなわち、学習モジュール間の共有強度が、類似するパターンを学習する学習モジュールに対しては強く、類似しないパターンを学習する学習モジュールに対しては弱くなるように、学習の過程で動的に変化（制御）させることができるようにした学習装置の実施の形態について説明する。

【 0 1 4 5 】

なお、以下では、パターン学習モデルとして、RNNPBより一般的な、入力層にPBユニットを持たないRNNを採用し、RNNの上述した係数 β_{ij} を共有強度として動的に制御する場合を例にした実施の形態について説明する。RNNは、PBユニットを有していない以外は、更新学習等、上述したRNNPBと同様に行うことができるので、上述したRNNPB#iをRNN#iに置きかえて説明する。

20

【 0 1 4 6 】

図 1 2 は、本発明を適用した学習装置であって、共有強度（係数 β_{ij} ）の動的制御を可能とする学習装置の一実施の形態の構成例を示している。

【 0 1 4 7 】

図 1 2 において、上述した図 1 の学習装置と対応する部分については同一の符号を付してあり、その説明は省略する。

【 0 1 4 8 】

即ち、図 1 2 の学習装置 1 0 1 は、図 1 の学習装置と同様の構成を有するパターン学習部 1 1 1 と、共有強度を制御する共有強度制御部 1 1 2 とにより構成されている。

30

【 0 1 4 9 】

パターン学習部 1 1 1 は、N個の学習モジュール 1 0₁ないし 1 0_Nによって構成され、パターン学習モデルの複数のモデルパラメータ（学習リソース）、即ち、RNN#iのウエイトマトリクス w_i を学習（更新）する更新学習を行う。

【 0 1 5 0 】

共有強度制御部 1 1 2 は、パラメータ更新ベクトル取得部 1 2 1、パラメータ差分ベクトル生成部 1 2 2、共有強度更新部 1 2 3、および共有強度記憶部 1 2 4を備え、N個の学習モジュール 1 0₁ないし 1 0_Nそれぞれが学習したときの学習誤差が最小化されるように共有強度を制御（更新）する。

40

【 0 1 5 1 】

図 1 3 を参照して、共有強度制御部 1 1 2 による共有強度の更新について説明する。

【 0 1 5 2 】

パターン学習モデルとしてRNNを採用した場合の共有強度である係数 β_{ij} は、方向性をもつ係数であって、図 1 3 に示されるように、学習モジュール 1 0_jから 1 0_iへの影響の強さを表す係数である。換言すれば、係数 β_{ij} は、RNN#iのウエイトマトリクス w_i に対する、RNN#jのウエイトマトリクス w_j の影響の強さを表している。

【 0 1 5 3 】

共有強度制御部 1 1 2 は、学習により学習モジュール 1 0_iのモデルパラメータを更新

50

したときの学習誤差 E_i が最小化されるように係数 β_{ij} を更新する。即ち、係数 β_{ij} の更新量を β_{ij} とすると、係数 β_{ij} を更新する式は、次式(4)のように表すことができ、式(4)の更新量 β_{ij} は、式(5)のように表すことができる。

【0154】

【数4】

$$\beta_{ij} = \beta_{ij} + \Delta\beta_{ij} \quad \dots (4)$$

【数5】

$$\Delta\beta_{ij} = \eta \cdot \frac{\partial E_i}{\partial \beta_{ij}} \quad \dots (5)$$

10

【0155】

式(5)において、 $\partial E_i / \partial \beta_{ij}$ は、学習誤差 E_i が最小化される方向(勾配)を表し、 η は、その最小化される方向に沿って1回あたりどれだけ進むかを表す係数(学習率)を表す。

【0156】

ここで、式(5)は、さらに次式(6)のように展開することができる。

【0157】

【数6】

$$\frac{\partial E_i}{\partial \beta_{ij}} = \frac{\partial E_i}{\partial w_i} \cdot \frac{\partial w_i}{\partial \Delta w_i} \cdot \frac{\partial \Delta w_i}{\partial \beta_{ij}} \quad \dots (6)$$

20

【0158】

式(6)の $\partial w_i / \partial w_i$ は、式(1)を w_i で偏微分した結果であり、1に等しい。また、式(6)の $\partial w_i / \partial \beta_{ij}$ は、式(2)を β_{ij} で偏微分した結果であり、実質的には $(w_j - w_i)$ のみで表すことができる。したがって、式(5)は、実質的に、式(7)のように表すことができる。

【0159】

【数7】

$$\Delta\beta_{ij} = \eta \cdot \frac{\partial E_i}{\partial w_i} \cdot (w_j - w_i) \quad \dots (7)$$

30

【0160】

式(7)の $\partial E_i / \partial w_i$ は、学習時のBPTTの過程で求まるウエイトマトリクス w_i を更新するときの更新量(更新量ベクトル)を表す。一方、式(7)の $(w_j - w_i)$ は、ウエイトマトリクス w_j からウエイトマトリクス w_i を引いた差分ベクトルである。従って、係数 β_{ij} の更新量 β_{ij} は、学習によるウエイトマトリクス w_i の更新量ベクトルと、ウエイトマトリクス w_j からウエイトマトリクス w_i を引いた差分ベクトルの相関(内積)で求められることになる。

【0161】

40

換言すれば、共有強度の更新量は、学習モジュール10_iの学習時のパラメータ更新ベクトル(学習モジュール10_iのモデルパラメータ更新の際の更新量ベクトル)と、学習モジュール10_jのモデルパラメータから学習モジュール10_iのモデルパラメータを引き算することにより得られるモデルパラメータ差分ベクトルの相関(内積)となっている。

【0162】

式(7)は、概念的には、学習モジュール10_iに注目したとき、学習モジュール10_jが、自分(学習モジュール10_i)と同じ学習方向の成分をより多く有している場合には、学習モジュール10_jとの共有度合いを強めて、 β_{ij} を大きくし、学習モジュール10_jが、自分(学習モジュール10_i)と違う学習方向の成分をより多く有している場合には、学習モジュール10_jとの共有度合いを弱めて、 β_{ij} を小さくすることを表している。

50

【 0 1 6 3 】

したがって、共有強度制御部 1 1 2 は、学習によりモデルパラメータが更新されるごとに、各学習モジュールについて、モデルパラメータ更新による更新量ベクトルと、モデルパラメータ差分ベクトルの内積を計算し、その結果を用いて、式 (4) により共有強度を更新する。

【 0 1 6 4 】

図 1 2 に戻り、パラメータ更新ベクトル取得部 1 2 1、パラメータ差分ベクトル生成部 1 2 2、共有強度更新部 1 2 3、および共有強度記憶部 1 2 4 それぞれの処理を説明する。

【 0 1 6 5 】

パラメータ更新ベクトル取得部 1 2 1 は、学習モジュール $1 0_i$ の共有強度の更新量を計算するための、学習モジュール $1 0_i$ のモデルパラメータを学習により更新した際の更新量ベクトルを取得する。RNNを採用したパターン学習モデルでは、パラメータ更新ベクトル取得部 1 2 1 は、RNN#i のBPTTの過程で求めた式 (7) の $\partial E_i / \partial w_i$ を取得する。

【 0 1 6 6 】

パラメータ差分ベクトル生成部 1 2 2 は、学習モジュール $1 0_i$ の共有強度の更新量を計算するための、学習モジュール $1 0_i$ のモデルパラメータと学習モジュール $1 0_j$ のモデルパラメータをパターン学習部 1 1 1 から取得する。そして、パラメータ差分ベクトル生成部 1 2 2 は、学習モジュール $1 0_j$ のモデルパラメータから学習モジュール $1 0_i$ のモデルパラメータを引き算することにより得られるモデルパラメータ差分ベクトルを生成する。RNNを採用したパターン学習モデルでは、パラメータ差分ベクトル生成部 1 2 2 は、ウエイトマトリクス w_i とウエイトマトリクス w_j を取得し、ウエイトマトリクス w_j からウエイトマトリクス w_i を引いた差分ベクトル ($w_j - w_i$) を生成する。

【 0 1 6 7 】

共有強度更新部 1 2 3 は、パラメータ更新ベクトル取得部 1 2 1 が取得したパラメータ更新ベクトルと、パラメータ差分ベクトル生成部 1 2 2 が生成したモデルパラメータ差分ベクトルを用いて共有強度を更新する。RNNを採用したパターン学習モデルでは、共有強度更新部 1 2 3 は、式 (7) により更新量 β_{ij} を求め、さらに式 (4) により更新後の係数 β_{ij} を求める。

【 0 1 6 8 】

共有強度更新部 1 2 3 は、更新後の共有強度 (係数 β_{ij}) をパターン学習部 1 1 1 の各学習モジュール $1 0_1$ ないし $1 0_N$ に供給するとともに、共有強度記憶部 1 2 4 に記憶させる。共有強度記憶部 1 2 4 に記憶された共有強度は、次の更新時に共有強度更新部 1 2 3 により取得され、式 (4) における、現在の共有強度 (係数 β_{ij}) として使用される。

【 0 1 6 9 】

次に、図 1 4 のフローチャートを参照して、共有強度の動的制御も行う、学習装置 1 0 1 の学習処理について説明する。

【 0 1 7 0 】

ステップ S 4 1 において、学習モジュール $1 0_i$ のモデル学習部 1 2_i は、モデル記憶部 1 3_i に記憶されたモデルパラメータを、例えば、乱数等によって初期化する。RNNを採用したパターン学習モデルでは、モデル記憶部 1 3_i に記憶されたRNN#i のモデルパラメータであるウエイトマトリクス w_i が、例えば、乱数等によって初期化される。

【 0 1 7 1 】

なお、ステップ S 4 1 と後述するステップ S 4 3 の処理は、上述した図 2 および図 4 の学習処理と同様に、N個の学習モジュール $1 0_1$ ないし $1 0_N$ のすべてについて行われる。

【 0 1 7 2 】

ステップ S 4 2 において、モデルパラメータ共有部 2 0 は、N個の学習モジュール $1 0_1$ ないし $1 0_N$ のすべてに、モデルパラメータを共有させる共有処理を行う。RNNを採用したパターン学習モデルでは、ウエイトマトリクス共有部 2 1 が、N個の学習モジュール $1 0_1$ ないし $1 0_N$ のすべてに、ウエイトマトリクス w_1 ないし w_N のすべてを共有させる共有処理

10

20

30

40

50

を行う。

【0173】

ステップS43において、学習モジュール10_iが、その学習モジュール10_iで学習すべき学習データが供給（入力）されるのを待って、その学習データを用いて、モデルパラメータを更新する更新学習を行う。

【0174】

RNNを採用したパターン学習モデルでは、ステップS43において、モデル学習部12_iが、パターン入力部11_iからの学習データ x_i を用いて、モデル記憶部13_iに記憶されたRNN#iのウエイトマトリクス w_i を更新する更新学習を、例えば、BPTT(Back-Propagation Through Time)法により行い、その更新学習によって得られた新たなモデルパラメータとしてのウエイトマトリクス w_i によって、モデル記憶部13_iの記憶内容を更新する。

10

【0175】

ステップS44において、共有強度制御部112が、N個の学習モジュール10₁ないし10_Nのすべてについて、学習後のパラメータ更新ベクトルとモデルパラメータ差分ベクトルを用いて、学習モジュール間の共有強度を更新する共有強度の動的制御処理を行う。ステップS44の処理の詳細は、図15を参照して後述する。

【0176】

ステップS44の後、処理はステップS45に進み、学習装置101は、学習の終了条件が満たされているかどうかを判定する。

【0177】

20

ステップS45において、学習の終了条件が満たされていないと判定された場合、処理は、ステップS42に戻り、以下、同様の処理が繰り返される。

【0178】

また、ステップS45において、学習の終了条件が満たされていると判定された場合、処理は終了する。

【0179】

図14の学習処理では、図2の学習処理と比較すると、モデルパラメータの更新とモデルパラメータの共有処理の順序が逆になっている。即ち、図2の学習処理では、ステップS12で、モデルパラメータの更新を行った後、ステップS13で、パラメータの共有処理を行っているが、図14の学習処理では、ステップS42で、モデルパラメータの共有処理を行った後、ステップS43で、モデルパラメータの更新を行っている。

30

【0180】

図2の学習処理では、上述したように、ステップS12とステップS13の処理の順番は、どちらが先であってもよいが、共有強度の動的制御も行う学習処理では、図14に示したように、モデルパラメータの共有処理を行ってから、モデルパラメータの更新を行う方がよい。

【0181】

なぜなら、共有強度の動的制御も行う学習処理では、上述したように、共有強度の更新量の計算に、モデルパラメータの更新直後の更新量ベクトルそのままの値を利用することができるからである。換言すれば、モデルパラメータの更新を行ってから、モデルパラメータの共有処理を行った場合には、共有強度の更新量の計算に、モデルパラメータの更新直後の更新量ベクトルそのままの値ではなく、モデルパラメータの共有処理後（共通化後）の更新量ベクトルを利用することになり、更新直後の更新量ベクトルそのままの値との誤差を、共有強度の更新量の計算に含むことになるからである。なお、その誤差が無視できる場合には、図2の学習処理と同様の順番で処理を行ってもよい。

40

【0182】

図15は、図14のステップS44における共有強度の動的制御処理の詳細なフローチャートを示している。

【0183】

初めに、ステップS61において、パラメータ更新ベクトル取得部121は、学習モジ

50

ジュール 10_i の学習時のパラメータ更新ベクトル（学習モジュール 10_i のモデルパラメータ更新の際の更新量ベクトル）を取得する。RNNを採用したパターン学習モデルでは、パラメータ更新ベクトル取得部 121 は、ステップ S43 のウエイトマトリクス w_i を更新する BPTT の過程で求めた、式 (7) の $\partial E_i / \partial w_i$ を取得する。

【0184】

ステップ S62 において、パラメータ差分ベクトル生成部 122 は、学習モジュール 10_i のモデルパラメータと学習モジュール 10_j のモデルパラメータをパターン学習部 111 から取得し、学習モジュール 10_j のモデルパラメータから学習モジュール 10_i のモデルパラメータを引き算することにより得られるモデルパラメータ差分ベクトルを生成する。RNNを採用したパターン学習モデルでは、パラメータ差分ベクトル生成部 122 は、ウエイトマトリクス w_i とウエイトマトリクス w_j をパターン学習部 111 から取得し、ウエイトマトリクス w_j からウエイトマトリクス w_i を引いた差分ベクトル ($w_j - w_i$) を生成する。

10

【0185】

ステップ S63 において、共有強度更新部 123 は、パラメータ更新ベクトル取得部 121 が取得したパラメータ更新ベクトルと、パラメータ差分ベクトル生成部 122 が生成したモデルパラメータ差分ベクトルを用いて共有強度を更新する。RNNを採用したパターン学習モデルでは、共有強度更新部 123 は、ウエイトマトリクス w_i の更新量ベクトルと、ウエイトマトリクス w_j からウエイトマトリクス w_i を引いた差分ベクトル ($w_j - w_i$) の内積により更新量 β_{ij} を求め、さらに式 (4) により更新後の係数 β_{ij} を求める。

【0186】

20

ステップ S64 において、共有強度更新部 123 は、更新後の共有強度を、次の共有強度の更新のため、共有強度記憶部 124 に記憶させる。RNNを採用したパターン学習モデルでは、共有強度更新部 123 は、更新後の係数 β_{ij} を、共有強度記憶部 124 に供給し、記憶させる。

【0187】

以上の処理が、共有強度の動的制御処理として、図 14 のステップ S44 で行われる。

【0188】

次に、図 16 ないし図 21 を参照して、図 12 の学習装置 101 による共有強度の動的制御処理を検証した実験結果について説明する。

【0189】

30

図 16 は、実験を行った環境を示している。

【0190】

本件発明者は、実験として、図 12 の学習装置 101 を備える移動ロボット 141 に 2 次元平面上の移動を行動パターンとして学習させることを行った。なお、パターン学習モデルには、3 つのコンテキストが入力層にフィードバックされる RNN を採用した。

【0191】

具体的には、本件発明者は、移動ロボット 141 を、図 16 に示される、4 面の壁で囲まれた空間（部屋）161 内に置き、その空間 161 内を移動する 6 つの行動パターン ACT#1 ないし ACT#6 を移動ロボット 141 の各 RNN に学習させた。なお、空間 161 のなかには、障壁 171 ないし 174 とライト 175 が設けられている。また、移動ロボット 141 には、6 つの行動パターン ACT#1 ないし ACT#6 のほかに、壁（障壁）を避ける、光に近づくなどの反射行動が予め入力（学習）されている。

40

【0192】

移動ロボット 141 は、距離センサと光センサを内蔵している。より具体的には、図 17 に示されるように、移動ロボット 141 は、2 次元平面（XY 平面）と並行な所定の面上で、かつ、自分の周囲 360 度を等分割した 8 方向の検出方向（点線で示される方向）の距離または光を検出する距離センサおよび光センサを備える。距離センサの 8 方向の入力値は d_1 ないし d_8 であり、光センサの 8 方向の入力値は h_1 ないし h_8 である。

【0193】

そして、移動ロボット 141 に行動パターンを学習させるとは、移動ロボット 141 に

50

2次元平面上の軌道（位置の軌跡）を学習させるのではなく、移動ロボット141が各軌道を通じたときに、センサが取得したセンサ入力値と、移動ロボット141の駆動部の出力値の時系列データを学習させることを意味する。

【0194】

従って、実験では、移動ロボット141の、距離センサの入力値 d_1 ないし d_8 と光センサの入力値 h_1 ないし h_8 、および、所定のモータ出力に対応する移動ベクトル (m_x, m_y) からなる18次元 $(m_x, m_y, d_1, \dots, d_8, h_1, \dots, h_8)$ のベクトルパターン（時系列データ）を学習させた。

【0195】

なお、本件発明者は、6つの行動パターンACT#1ないしACT#6それぞれについて、互いに軌道を少し異なるものとした5つのバリエーションを学習させたので、学習させた行動パターンのサンプル数は、合計で30個（6つの行動パターン×5バリエーション）となる。

10

【0196】

図18は、30個のRNN#1ないしRNN#30に学習させた行動パターンACT#1ないしACT#6を示している。

【0197】

図18に示されるように、本件発明者は、行動パターンACT#1についての5つのバリエーションをRNN#1ないしRNN#5に学習させ、行動パターンACT#2についての5つのバリエーションをRNN#6ないしRNN#10に学習させ、行動パターンACT#3についての5つのバリエーションをRNN#11ないしRNN#15に学習させ、行動パターンACT#4についての5つのバリエーションをRNN#16ないしRNN#20に学習させ、行動パターンACT#5についての5つのバリエーションをRNN#21ないしRNN#25に学習させ、行動パターンACT#6についての5つのバリエーションをRNN#26ないしRNN#30に学習させた。

20

【0198】

図19は、共有強度の動的制御処理を含む学習処理後の学習モジュール間（RNN間）の共有強度とウエイト距離を濃淡で示した図である。

【0199】

図19Aは、学習処理後の学習モジュール間（RNN間）の共有強度を濃淡で示している。図19Aの横軸と縦軸は、それぞれ30個のRNN#1ないしRNN#30を表す。図19Aでは、横軸のRNN# i のウエイトマトリクス w_i に対する、縦軸のRNN# j のウエイトマトリクス w_j の共有強度である係数 β_{ij} が濃淡で示されており、濃淡は、0.0より大で1.0より小の範囲で、淡い（白い）ほど共有強度が強く（係数 β_{ij} が大きく）、濃い（黒い）ほど共有強度が弱い（係数 β_{ij} が小さい）ことを表す。

30

【0200】

図19Aにおいては、同一のウエイトマトリクスどうしの共有強度のラインである対角線を中心とする数個のRNNどうしの共有強度が強く、それ以外のRNNとの共有強度が弱くなっている。このことは、共有強度の動的制御処理により、同じ行動パターンを学習する学習モジュール（RNN）間で共有強度が強くなるように制御（更新）されたことを表している。即ち、図19Aは、動的制御により、適切に共有強度が設定されたことを表している。

40

【0201】

一方、図19Bは、学習処理後の学習モジュール間（RNN間）のウエイト距離を濃淡で示している。

【0202】

ウエイト距離とは、2つのRNNのウエイトマトリクス間のユークリッド距離であり、例えば、RNN#1のウエイトマトリクス w_1 の各ウエイトを $w_{1,k,l}$ （ $1 \leq k \leq Q, 1 \leq l \leq R$ ）、RNN#2のウエイトマトリクス w_2 の各ウエイトを $w_{2,k,l}$ としたとき、RNN#1とRNN#2のウエイト距離 $D_{\text{weight}}(1,2)$ は、式（8）で表すことができる。

【0203】

50

【数 8】

$$D_{\text{weight}}(1, 2) = \sqrt{\sum_{k=1}^Q \sum_{l=1}^R (w_{1,k,l} - w_{2,k,l})^2} \quad \dots (8)$$

【0204】

図19Bの横軸と縦軸は、それぞれ30個のRNN#1ないしRNN#30を表す。図19Bでは、横軸のRNN#iと、縦軸のRNN#jのウエイト距離 $D_{\text{weight}}(i, j)$ が濃淡で示されており、濃淡は、0.0より大で200.0より小の範囲で、濃い(黒い)ほどウエイト距離が小さく、淡い(白い)ほどウエイト距離が大きいことを表す。

【0205】

図19Bにおいては、同一のウエイトマトリクスどうしのウエイト距離のラインである対角線を中心とする数個のRNNどうしのウエイト距離が小さく、それ以外のRNNとのウエイト距離が大きくなっている。このことは、共有強度の動的制御処理により、同じ行動パターンを学習する学習モジュール間でウエイト距離が小さくなるように制御(更新)されたことを表している。即ち、図19Bも、動的制御により、適切に共有強度が設定されたことを表している。

【0206】

図20は、共有強度の動的制御処理を含む学習処理後の各RNN間のウエイト距離を、2次元空間に射影したサモンマップ(sammon map)を示している。

【0207】

サモンマップは、n次元入力ベクトルを、入力ベクトル空間での関係を維持したまま2次元平面上へ写像することができる手法であり、その詳細は、JOHN W. SAMMON, JR., A Nonlinear Mapping For Data Structure Analysis, IEEE TRANSACTIONS ON COMPUTERS, VOL. C-18, NO. 5, MAY 1969に開示されている。

【0208】

図20において、番号が付されたバツ印(×)は、その番号iに対応するRNN#iのサモンマップ上の位置を表す。そして、バツ印間の距離は、RNN間のウエイト距離を反映している。なお、バツ印を中心とする細線の円は、RNN間のウエイト距離の近さ(RNN間の重なり)をわかりやすくするため、ウエイト距離が10となる範囲を示したものである。

【0209】

また、図20においては、同一の行動パターンACT#2ないしACT#6の学習データを学習した学習モジュール(RNN)群が、それぞれ、太い点線の円で示されている。これにより、行動パターンACT#1を除いて、類似した学習データ(学習サンプル)を学習した学習モジュール間では、ウエイト距離が小さくなっている(所定の範囲内に収まっている)ことが分かる。

【0210】

従って、共有強度の動的制御処理によれば、類似した学習データ(学習サンプル)を学習した学習モジュール間では、ウエイト距離が小さい状態(共通化)を維持しつつも、類似していない学習サンプルを学習した学習モジュール間では、ウエイト距離が大きい状態を維持できている。即ち、共有強度の動的制御処理を行っても、規模拡張性と汎化特性の両立が可能となっていることが分かる。

【0211】

比較のため、30個のRNN#1ないしRNN#30すべての係数 β_{ij} を0.3, 0.1, 0.01の固定値($\beta_{ij} = 0.3$, $\beta_{ij} = 0.1$, $\beta_{ij} = 0.01$)で学習処理を行った場合のサモンマップを図21に示す。

【0212】

図21Aは、係数 β_{ij} が0.3の固定値($\beta_{ij} = 0.3$)の場合であり、図21Bは、係数 β_{ij} が0.1の固定値($\beta_{ij} = 0.1$)の場合であり、図21Cは、係数 β_{ij} が0.01の固定値($\beta_{ij} = 0.01$)の場合である。

【0213】

10

20

30

40

50

なお、図 2 0 と図 2 1 A ないし図 2 1 C は、互いに表示範囲が異なるので、縦軸と横軸のスケールが異なっている。ただし、ウエイト距離が 1 0 となる範囲を示した細線の円は、図 2 0 と図 2 1 A ないし図 2 1 C において共通である。

【 0 2 1 4 】

図 2 1 A ないし図 2 1 C によれば、例えば、図 2 1 C に示されるように、共有強度を小さく設定した場合には、各 RNN が他の RNN のいずれとも独立する（ウエイト距離が大きい状態となる）ように位置し、反対に、図 2 1 A に示されるように、共有強度を大きく設定した場合には、各 RNN が他の RNN のいずれとも共通化する（ウエイト距離が小さい状態となる）ように位置している。図 2 1 B は、図 2 1 A と図 2 1 C の中間的な分布となっている。

【 0 2 1 5 】

また、図 2 1 B を参照すると、類似した学習データ（学習サンプル）を学習した学習モジュール間でウエイト距離が小さく、類似していない学習サンプルを学習した学習モジュール間でウエイト距離が大きいというような位置の傾向は小さい。

【 0 2 1 6 】

従って、図 2 0 と図 2 1 A ないし図 2 1 C とを比較してみても、図 1 2 の学習装置 1 0 1 が、規模拡張性と汎化特性を両立させた状態で、類似するパターンを学習する学習モジュールどうしについては共有強度を強く設定し、類似しないパターンを学習する学習モジュールとの共有強度は弱く設定することができるということが言える。

【 0 2 1 7 】

以上のように、学習装置 1 0 1 によれば、学習モジュール間の共有強度を、ユーザが決定し、設定しなくても（自動的に）、最適に設定することができる。

【 0 2 1 8 】

上述した一連の処理は、ハードウェアにより行うこともできるし、ソフトウェアにより行うこともできる。一連の処理をソフトウェアによって行う場合には、そのソフトウェアを構成するプログラムが、汎用のコンピュータ等にインストールされる。

【 0 2 1 9 】

そこで、図 2 2 は、上述した一連の処理を実行するプログラムがインストールされるコンピュータの一実施の形態の構成例を示している。

【 0 2 2 0 】

プログラムは、コンピュータに内蔵されている記録媒体としてのハードディスク 2 0 5 や ROM 2 0 3 に予め記録しておくことができる。

【 0 2 2 1 】

あるいはまた、プログラムは、フレキシブルディスク、CD-ROM (Compact Disc Read Only Memory)、MO (Magneto Optical) ディスク、DVD (Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体 2 1 1 に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体 2 1 1 は、いわゆるパッケージソフトウェアとして提供することができる。

【 0 2 2 2 】

なお、プログラムは、上述したようなリムーバブル記録媒体 2 1 1 からコンピュータにインストールする他、ダウンロードサイトから、デジタル衛星放送用の人工衛星を介して、コンピュータに無線で転送したり、LAN (Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを、通信部 2 0 8 で受信し、内蔵するハードディスク 2 0 5 にインストールすることができる。

【 0 2 2 3 】

コンピュータは、CPU (Central Processing Unit) 2 0 2 を内蔵している。CPU 2 0 2 には、バス 2 0 1 を介して、入出力インタフェース 2 1 0 が接続されており、CPU 2 0 2 は、入出力インタフェース 2 1 0 を介して、ユーザによって、キーボードや、マウス、マイク等で構成される入力部 2 0 7 が操作等されることにより指令が入力されると、それにしたがって、ROM (Read Only Memory) 2 0 3 に格納されているプログラムを実行する。ある

10

20

30

40

50

いは、また、CPU 2 0 2 は、ハードディスク 2 0 5 に格納されているプログラム、衛星若しくはネットワークから転送され、通信部 2 0 8 で受信されてハードディスク 2 0 5 にインストールされたプログラム、またはドライブ 2 0 9 に装着されたリムーバブル記録媒体 2 1 1 から読み出されてハードディスク 2 0 5 にインストールされたプログラムを、RAM(Random Access Memory) 2 0 4 にロードして実行する。これにより、CPU 2 0 2 は、上述したフローチャートにしたがった処理、あるいは上述したブロック図の構成により行われる処理を行う。そして、CPU 2 0 2 は、その処理結果を、必要に応じて、例えば、入出力インタフェース 2 1 0 を介して、LCD(Liquid Crystal Display)やスピーカ等で構成される出力部 2 0 6 から出力、あるいは、通信部 2 0 8 から送信、さらには、ハードディスク 2 0 5 に記録等させる。

10

【0 2 2 4】

ここで、本明細書において、コンピュータに各種の処理を行わせるためのプログラムを記述する処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理（例えば、並列処理あるいはオブジェクトによる処理）も含むものである。

【0 2 2 5】

また、プログラムは、1のコンピュータにより処理されるものであっても良いし、複数のコンピュータによって分散処理されるものであっても良い。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

【0 2 2 6】

20

なお、本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能である。

【0 2 2 7】

即ち、本発明は、ある特定の空間パターン、時系列シーケンス及びパターンに特化した方法ではない。したがって、コンピュータのユーザインタフェースにおけるユーザ入力のパターン、ロボットのセンサ入力及びモータ出力のパターン、音楽データに関するパターン、画像データに関するパターン、言語処理における音素、単語、センテンスなどのパターンの学習及び学習結果に基づくパターンの予測や分類に適用することができる。

【図面の簡単な説明】

【0 2 2 8】

30

【図 1】本発明を適用する学習装置の基本となる学習装置の一実施の形態の構成例を示すブロック図である。

【図 2】図 1 の学習装置の学習処理を説明するフローチャートである。

【図 3】パターン学習モデルとしてRNNPBを採用した場合の図 1 の学習装置の構成例を示すブロック図である。

【図 4】パターン学習モデルとしてRNNPBを採用した場合の図 1 の学習装置の学習処理を説明するフローチャートである。

【図 5】シミュレーションの結果を示す図である。

【図 6】シミュレーションの結果を示す図である。

【図 7】シミュレーションの結果を示す図である。

40

【図 8】シミュレーションの結果を示す図である。

【図 9】シミュレーションに用いた時系列データを示す図である。

【図 1 0】RNNPBのモデルパラメータの共有を模式的に示す図である。

【図 1 1】「局所表現」スキーム、「分散表現」スキーム、及び「中間表現」スキームの学習の関係を模式的に示す図である。

【図 1 2】本発明を適用した学習装置の一実施の形態の構成例を示すブロック図である。

【図 1 3】共有強度の更新手法を説明するための図である。

【図 1 4】図 1 2 の学習装置の学習処理を説明するフローチャートである。

【図 1 5】共有強度の動的制御処理を説明するフローチャートである。

【図 1 6】実験を行った環境を説明する図である。

50

【図 17】移動ロボットについて説明する図である。

【図 18】図 12 の学習装置に学習させた行動パターンを示す図である。

【図 19】学習処理後の共有強度とウエイト距離を示す図である。

【図 20】学習処理後の RNN 間のウエイト距離を 2 次元空間に射影したサモンマップを示す図である。

【図 21】共有強度が固定値の場合の学習処理後のサモンマップを示す図である。

【図 22】本発明を適用したコンピュータの一実施の形態の構成例を示すブロック図である。

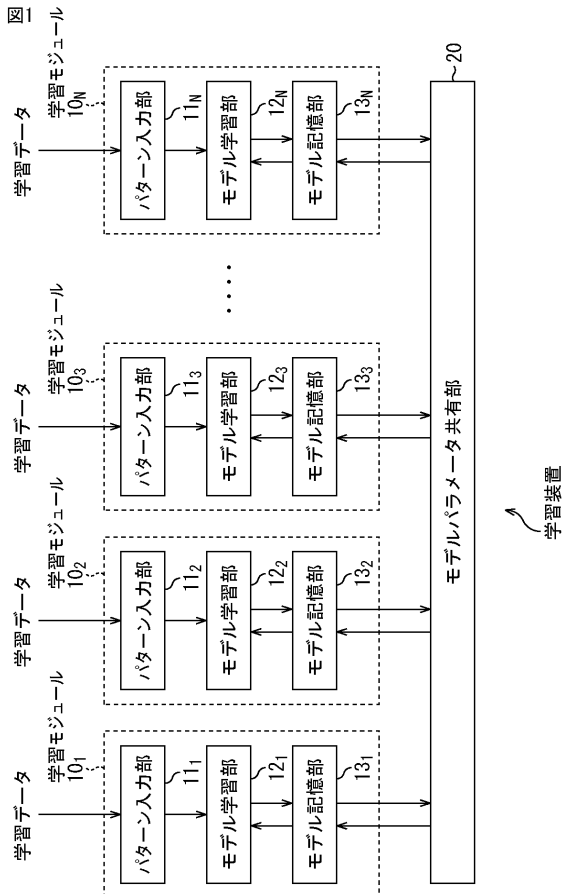
【符号の説明】

【0229】

10

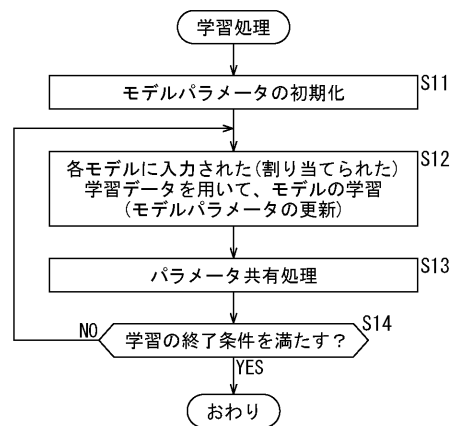
10₁ないし10_N 学習モジュール, 11₁ないし11_N パターン入力部, 12₁ないし12_N モデル学習部, 13₁ないし13_N モデル記憶部, 20 モデルパラメータ共有部, 21 ウエイトマトリクス共有部, 101 学習装置, 111 パターン学習部, 112 共有強度制御部, 121 パラメータ更新ベクトル取得部, 122 パラメータ差分ベクトル生成部, 123 共有強度更新部, 124 共有強度記憶部, 201 バス, 202 CPU, 203 ROM, 204 RAM, 205 ハードディスク, 206 出力部, 207 入力部, 208 通信部, 209 ドライブ, 210 入出力インタフェース, 211 リムーバブル記録媒体

【図 1】

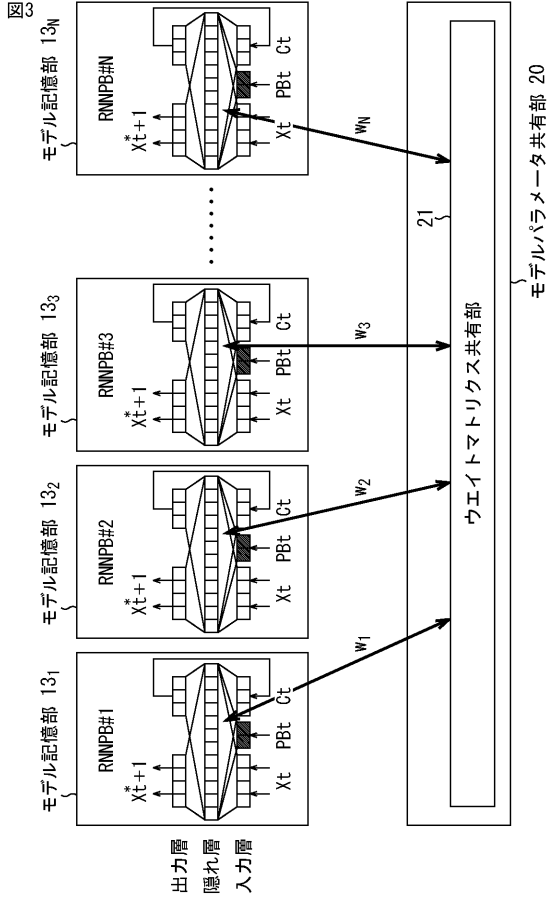


【図 2】

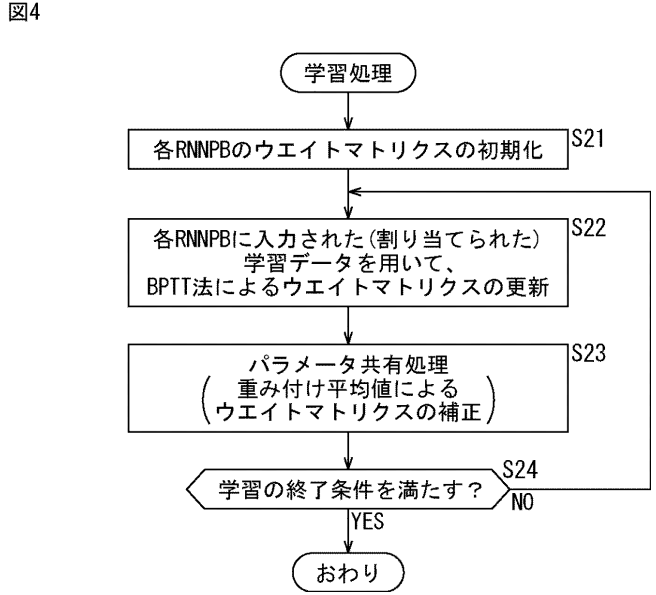
図2



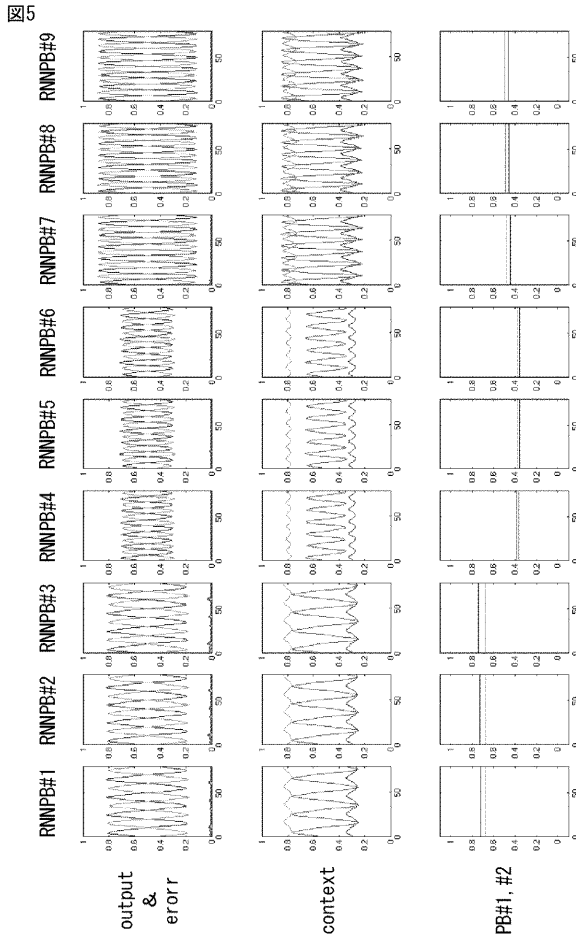
【図 3】



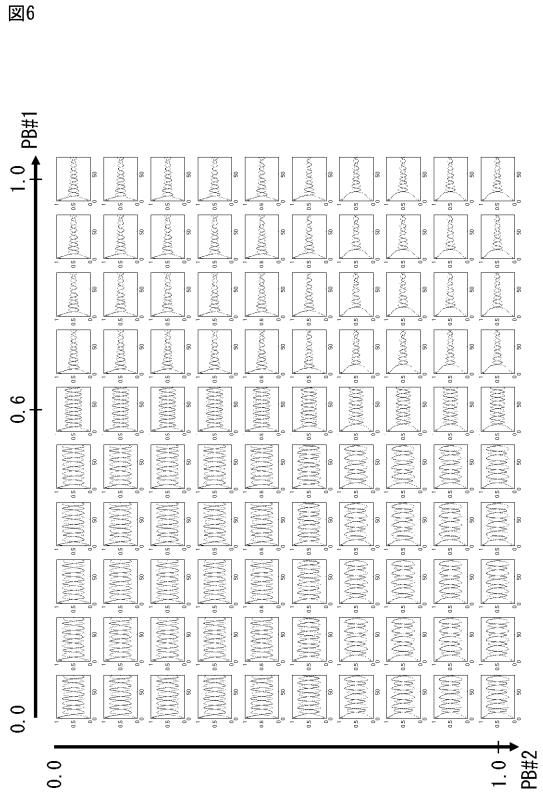
【図 4】



【図 5】

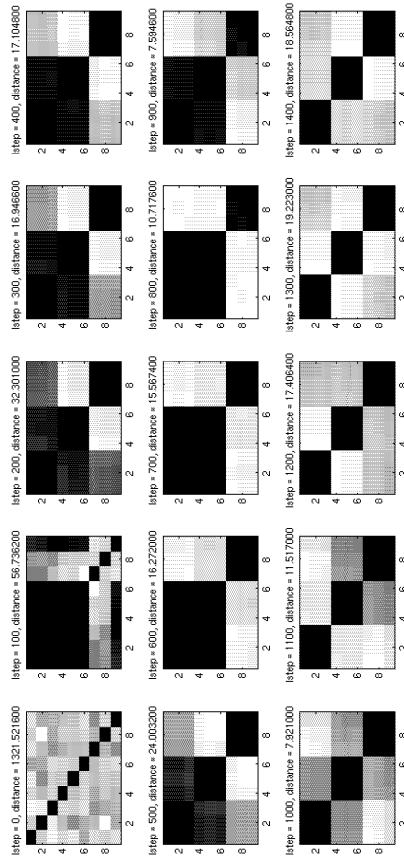


【図 6】



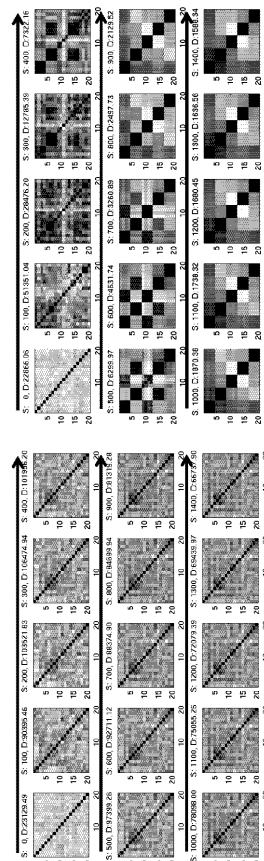
【図 7】

図7



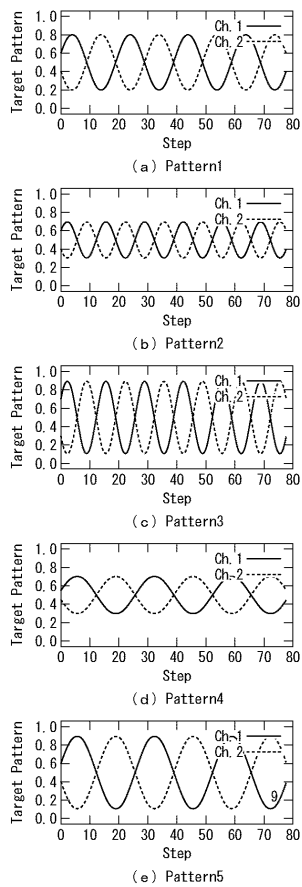
【図 8】

図8



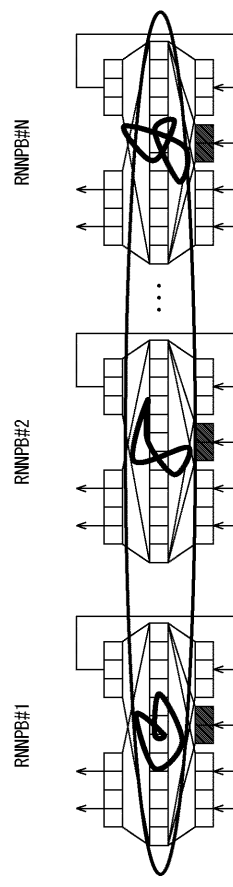
【図 9】

図9



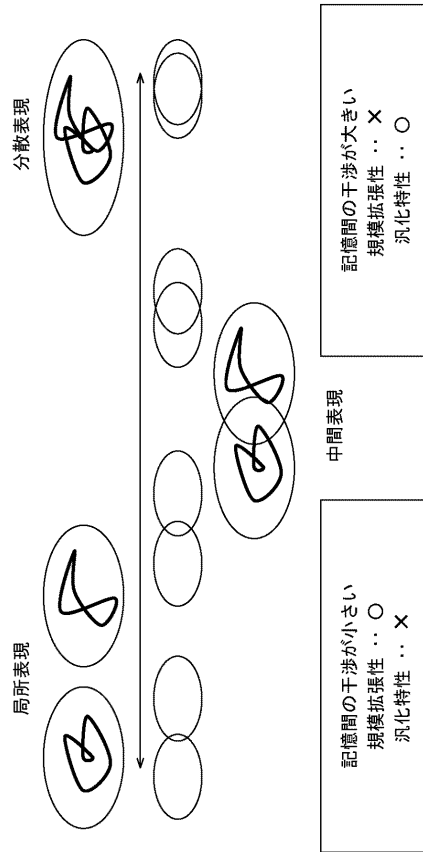
【図 10】

図10



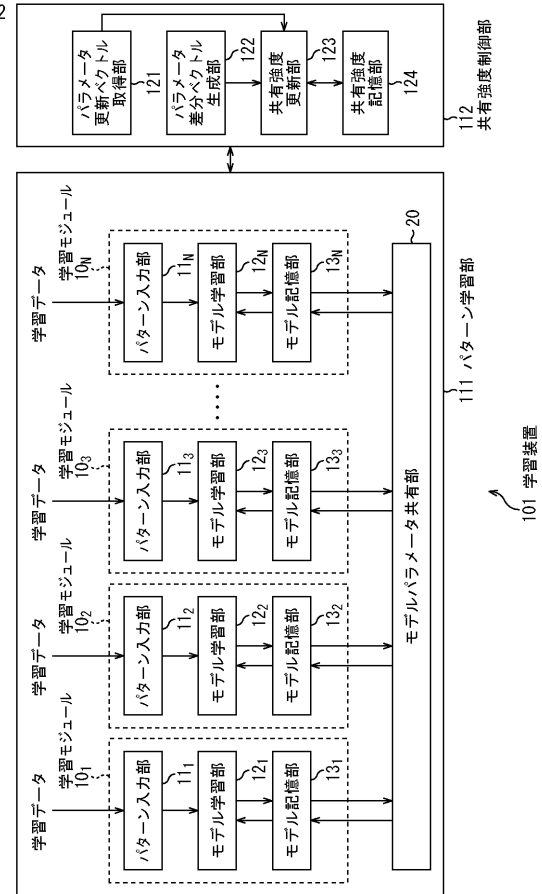
【図 1 1】

図11



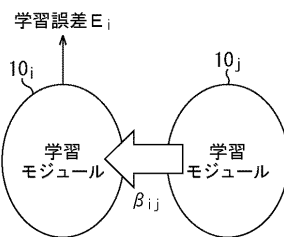
【図 1 2】

図12



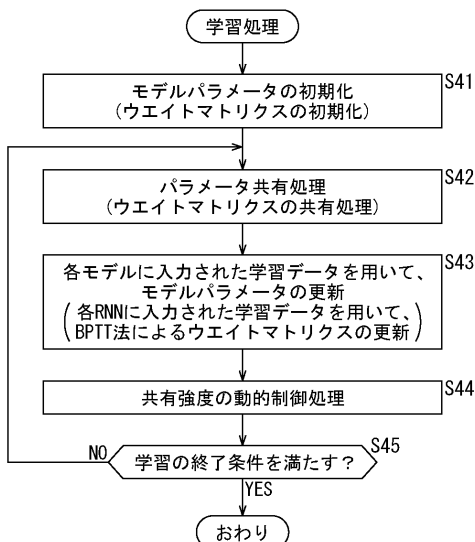
【図 1 3】

図13



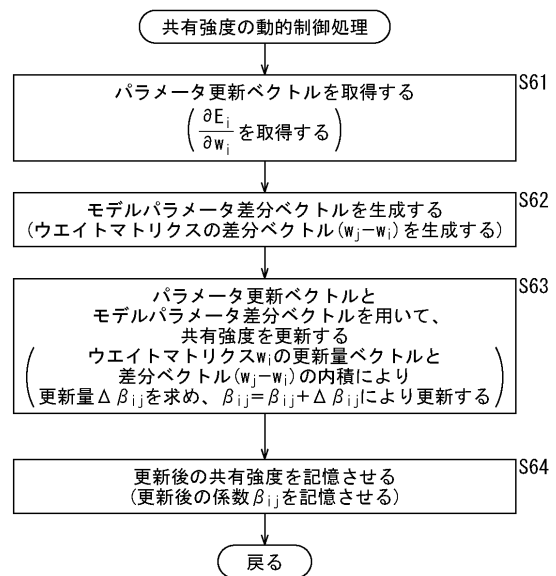
【図 1 4】

図14



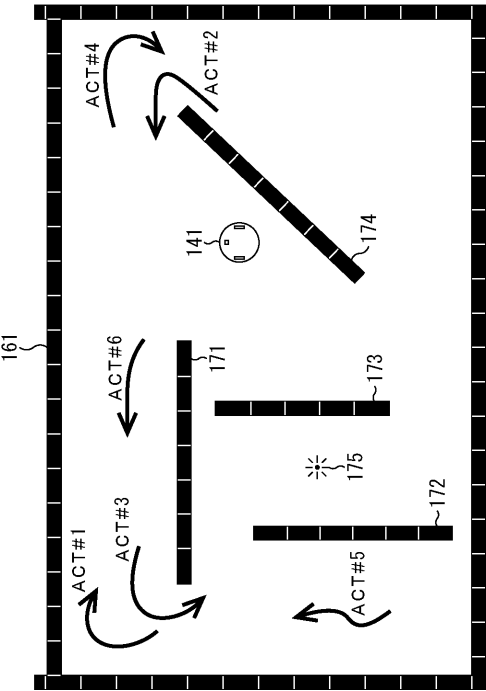
【図 1 5】

図15



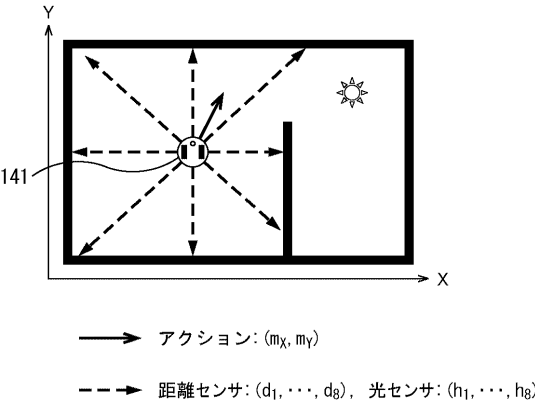
【図 16】

図16



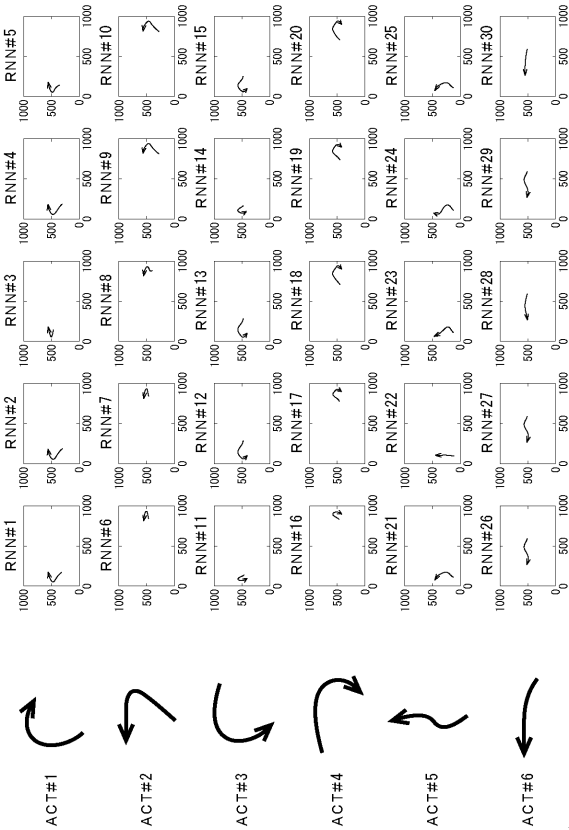
【図 17】

図17



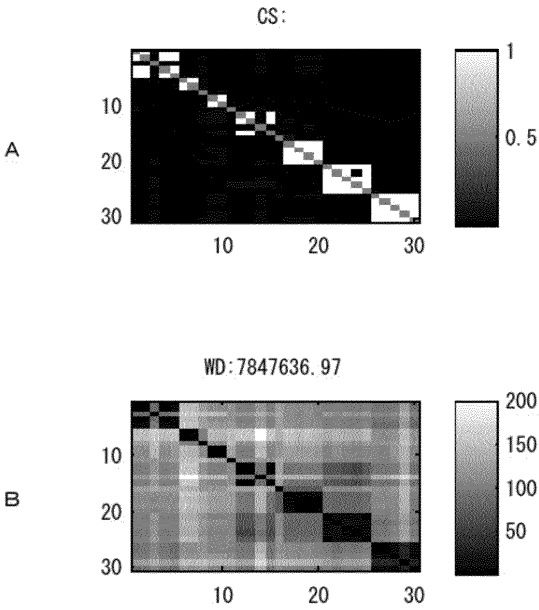
【図 18】

図18



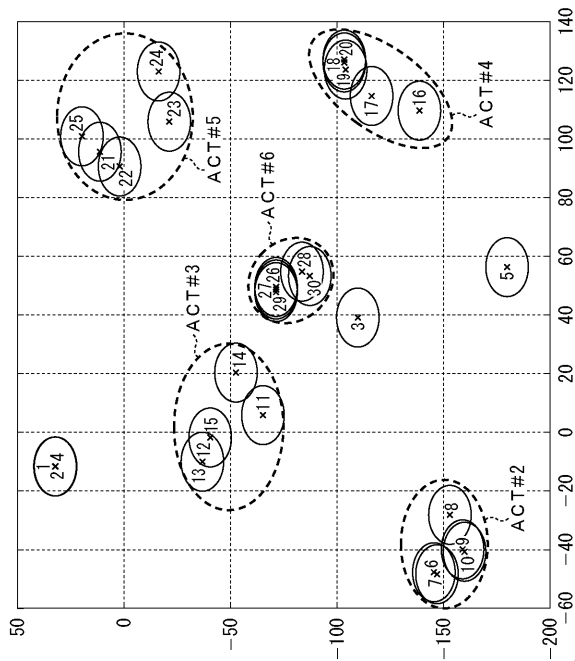
【図 19】

図19



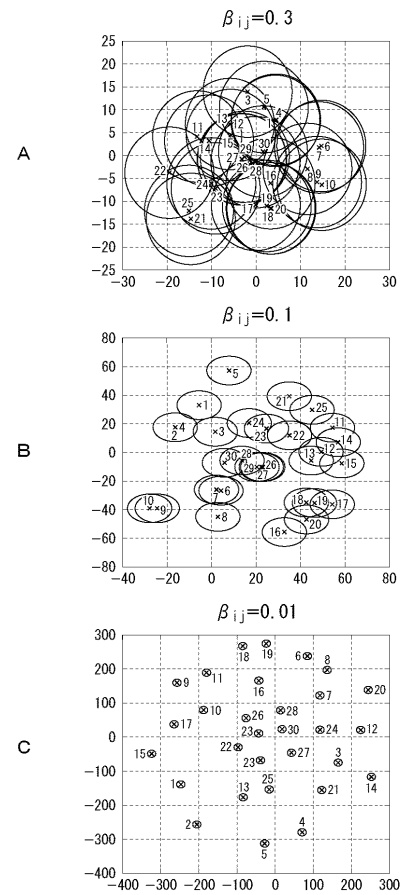
【図 20】

図20



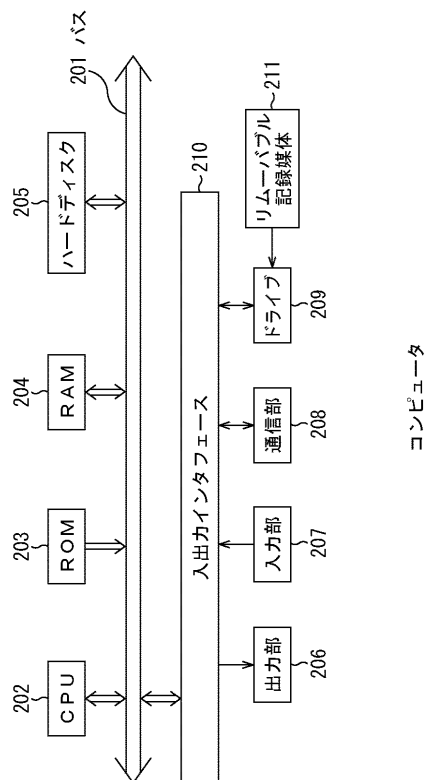
【図 21】

図21



【図 22】

図22



【手続補正書】

【提出日】平成21年7月10日(2009.7.10)

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】0 0 7 0

【補正方法】変更

【補正の内容】

【0 0 7 0】

したがって、式(2)の右辺のサメーション $\sum \beta_{ij}(w_j - w_i)$ は、係数 β_{ij} を重みとした、RNNPB# i のウエイトマトリクス w_i に対するRNNPB#1ないしRNNPB# N のウエイトマトリクス w_1 ないし w_N それぞれの偏差(差分)の重み付け平均値を表し、 α_i は、その重み付け平均値 $\sum \beta_{ij}(w_j - w_i)$ を、ウエイトマトリクス w_i に影響させる度合いを表す係数である。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0 1 0 0

【補正方法】変更

【補正の内容】

【0 1 0 0】

また、RNNPB#1には、パターンP#1の時系列データにノイズN#1を重畳して得られる時系列データを、RNNPB#2には、パターンP#1の時系列データにノイズN#2を重畳して得られる時系列データを、RNNPB#3には、パターンP#1の時系列データにノイズN#3を重畳して得られる時系列データを、それぞれ、学習データとして与えた。