

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6689507号  
(P6689507)

(45) 発行日 令和2年4月28日(2020.4.28)

(24) 登録日 令和2年4月10日(2020.4.10)

(51) Int.Cl. F I  
G O 6 N 20/00 (2019.01) G O 6 N 20/00

請求項の数 1 (全 16 頁)

(21) 出願番号	特願2019-542650 (P2019-542650)	(73) 特許権者	517047488
(86) (22) 出願日	平成31年2月20日 (2019.2.20)		株式会社 A B E J A
(86) 国際出願番号	PCT/JP2019/006319		東京都港区白金一丁目17番3号 N B F
(87) 国際公開番号	W02019/163823		プラチナタワー
(87) 国際公開日	令和1年8月29日 (2019.8.29)	(74) 代理人	100205659
審査請求日	令和1年8月6日 (2019.8.6)		弁理士 齋藤 拓也
(31) 優先権主張番号	特願2018-28235 (P2018-28235)	(74) 代理人	100154748
(32) 優先日	平成30年2月20日 (2018.2.20)		弁理士 菅沼 和弘
(33) 優先権主張国・地域又は機関	日本国 (JP)	(72) 発明者	河崎 敏弥
			東京都港区白金一丁目17番3号 N B F
			プラチナタワー 株式会社 A B E J A 内
早期審査対象出願		審査官	今城 朋彬
			最終頁に続く

(54) 【発明の名称】 情報処理システム

(57) 【特許請求の範囲】

【請求項 1】

学習データを入力するデータ入力手段と、  
 学習方法を入力する学習方法入力手段と、  
 前記データ入力手段により入力された所定の前記学習データと、前記学習方法入力手段により入力された所定の前記学習方法の組を選択する学習条件選択手段と、  
 前記学習条件選択手段により選択された前記学習データを含むデータ又は学習方法の少なくともいずれか一方を用いて学習を実行する学習手段と、  
 前記学習手段により実行された学習の結果の採用又は不採用を決定する採用決定手段と、  
 推論方法を入力する推論入力手段と、  
 前記採用決定手段により採用が決定された前記学習の結果と、前記推論入力手段により入力された前記推論方法とを組にしてモデルバージョンとし、前記モデルバージョンを1つ又はそれ以上を、管理するタスク毎に用意するモデル管理手段と、  
 前記モデル管理手段に紐づいた実行選択手段であって、当該モデル管理手段の管理するモデルバージョンと実行環境を選択する実行選択手段と、  
 前記実行選択手段により選択された前記実行環境において、前記実行選択手段により選択された前記モデルバージョンを実行させる実行実施手段と、  
 を備える情報処理システム。

【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、情報処理システムに関する。

## 【背景技術】

## 【0002】

従来より、様々な分野で機械学習が利用されている。機械学習とは、具体的に、複数のデータを用いて反復的に学習し、そこに潜むパターンをコンピュータがを見つけ出すことができる。そして、機械学習を利用した技術が提案されている（例えば特許文献1参照）。

## 【先行技術文献】

## 【特許文献】

10

## 【0003】

【特許文献1】特開2016-192007号公報

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0004】

しかしながら、上述の特許文献1に記載の技術を含む従来技術のみでは、データの収集状況に応じてモデルを更新したり、様々な実行環境で動かすという運用が困難であった。

## 【0005】

本発明は、このような状況に鑑みてなされたものであり、機械学習のモデル管理を容易にすることを目的とする。

20

## 【課題を解決するための手段】

## 【0006】

上記目的を達成するため、本発明の一態様の情報処理システムは、  
学習データを入力するデータ入力手段と、  
学習方法を入力する学習方法入力手段と、  
前記データ入力手段により入力された所定の前記学習データと、前記学習方法入力手段により入力された所定の前記学習方法の組を選択する学習条件選択手段と、  
前記学習条件選択手段により選択された前記学習データを含むデータ又は学習方法の少なくともいずれか一方を用いて学習を実行する学習手段と、  
前記学習手段により実行された学習の結果の採用又は不採用を決定する採用決定手段と

30

、  
推論方法を入力する推論入力手段と、  
前記採用決定手段により採用が決定された前記学習の結果と、前記推論入力手段により入力された前記推論方法とを組にしてモデルバージョンとし、前記モデルバージョンを1つ又はそれ以上を、管理するタスク毎に用意するモデル管理手段と、  
前記モデル管理手段に紐づいた実行選択手段であって、当該モデル管理手段の管理するモデルバージョンと実行環境を選択する実行選択手段と、  
前記実行選択手段により選択された前記実行環境において、前記実行選択手段により選択された前記モデルバージョンを実行させる実行選択手段と、  
を備える。

40

## 【発明の効果】

## 【0007】

本発明によれば、機械学習のモデル管理を容易にすることができる。

## 【図面の簡単な説明】

## 【0008】

【図1】本発明の情報処理システムの一実施形態としての機械学習管理運用システムのハードウェア構成を示すブロック図である。

【図2】図1の機能的構成の一例を示す機能ブロック図である。

【図3】学習ジョブ進行状況確認画面の一例を示す図である。

【図4】学習ジョブ進行状況確認画面の図3とは異なる例を示す図である。

50

【図５】学習ジョブ進行状況確認画面の図３及び図４とは異なる例を示す図である。

【図６】モデルのバージョン管理画面の一例を示す図である。

【図７】モデルのバージョン管理画面の図６とは異なる例を示す図である。

【図８】モデルのバージョン管理画面の図６及び図７とは異なる例を示す図である。

【図９】モデルのバージョン管理画面の図６乃至図８とは異なる例を示す図である。

【発明を実施するための形態】

【０００９】

以下、本発明の実施形態について、図面を用いて説明する。

【００１０】

図１は、本発明の情報処理システムの一実施形態としての機械学習管理運用システムのハードウェア構成例を示すブロック図である。 10

機械学習管理運用システムは、サーバとクライアント等ネットワークを介する複数台の情報装置により構成される実施形態等、各種各様な実施形態を取ることができる。ただし、以下説明の便宜上、本実施形態では、機械学習管理運用システムは、図１に示す情報処理装置で構成されるものとして説明する。

【００１１】

情報処理装置１は、ＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）１１と、ＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）１２と、ＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）１３と、ＧＰＵ（Ｇｒａｐｈｉｃｓ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）１４と、バス１５と、入出力インターフェース１６と、出力部１７と、入力部１８と、記憶部１９と、通信部２０と、ドライブ２１とを備えている。 20

【００１２】

ＣＰＵ１１及びＧＰＵ１４は、ＲＯＭ１２に記録されているプログラム、又は、記憶部１９からＲＡＭ１３にロードされたプログラムに従って各種の処理を実行する。

ＲＡＭ１３には、ＣＰＵ１１又はＧＰＵ１４が各種の処理を実行する上において必要なデータ等も適宜記憶される。

【００１３】

ＣＰＵ１１、ＲＯＭ１２、ＲＡＭ１３、及びＧＰＵ１４は、バス１５を介して相互に接続されている。このバス１５にはまた、入出力インターフェース１６も接続されている。入出力インターフェース１６には、出力部１７、入力部１８、記憶部１９、通信部２０、及びドライブ２１が接続されている。 30

【００１４】

出力部１７は液晶ディスプレイ等で構成され、各種情報を出力する。

入力部１８は、各種ハードウェア鉛等で構成され、各種情報を入力する。

記憶部１９は、ＤＲＡＭ（Ｄｙｎａｍｉｃ　Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）等で構成され、各種データを記憶する。

通信部２０は、インターネットを含むネットワークＮを介して他の装置との間で行う通信を制御する。

【００１５】

ドライブ２１は、必要に応じて設けられる。ドライブ２１には磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリ等よりなる、リムーバブルメディア３１が適宜装着される。ドライブ２１によってリムーバブルメディア３１から読み出されたプログラムは、必要に応じて記憶部１９にインストールされる。またリムーバブルメディア３１は、記憶部１９に記憶されている各種データも、記憶部１９と同様に記憶することができる。 40

【００１６】

図２は、機械学習管理運用システム（図１の情報処理装置１）の機能的構成の一例を示す機能ブロック図である。

【００１７】

図２の情報処理装置１のＣＰＵ１１やＧＰＵ１４においては、学習データ入力部１０１ 50

と、学習方法入力部 102 と、学習条件選択部 103 と、学習部 104 と、採用決定部 105 と、推論入力部 106 と、モデル管理部 107 と、実行選択部 108 と、実行実施部 109 とが機能する。

また、図 1 の学習処理装置の記憶部 19 においては、学習データが記憶される学習データ DB 200 と、モデルの各バージョンが記憶されるモデル DB 300 及びモデル DB 400 と、実行環境データが記憶される実行環境データ DB 500 とが設けられる。

【0018】

学習データ入力部 101 は、例えば学習対象を被写体を含む画像及びその学習対象に関する情報を含むデータ等学習に用いるデータ（以下、「学習データ」と呼ぶ）を入力する。学習データは、学習データ DB 200 に記憶される。

10

【0019】

学習方法入力部 102 は、学習方法を入力する。

【0020】

学習条件選択部 103 は、所定のデータ（学習データ DB 200 に記憶された 1 以上の学習データのうち、ユーザにより選択されたデータを含む）と学習方法の組を選択する。

【0021】

学習部 104 は、学習条件選択部 103 により選択されたデータ又は学習方法の少なくともいずれか一方を用いて学習を実行する。

【0022】

採用決定部 105 は、学習部 104 により出力された学習結果の採用又は不採用を決定する。

20

【0023】

推論入力部 106 は、推論方法を入力する。

【0024】

モデル管理部 107 は、採用決定部 105 により決定された学習結果と、推論入力部 106 により入力された推論方法を組にしてモデルバージョンとし、前記モデルバージョンを 1 つ又はそれ以上を管理するタスク毎に用意する。

即ち、図 2 の例では、モデル DB 300 やモデル DB 400 の夫々は、所定の 1 つのタスクについて、1 つ又はそれ以上のモデルバージョン（図 2 の例ではいずれも 3 つのモデルバージョン）を記憶する。そこで、モデル管理部 107 は、このようなモデル DB 300 やモデル DB 400 等のタスク毎のモデル DB を用いて、各モデルバージョンをタスク毎に管理することもできる。

30

【0025】

モデル管理部 107 に紐づいた実行選択部 108 は、当該モデル管理部 107 の管理するモデルバージョンと実行環境を選択する。

【0026】

実行実施部 109 は、実行環境において前記モデルバージョンを実行させる。

【0027】

このように、本発明が適用される機械学習管理運用システムは、次に示す課題を解決することができ、その結果として、次に示す効果を奏することができる。

40

即ち、データの収集状況に応じてモデルを更新したり、様々な実行環境で動かすという運用が従来大変であったという課題が存在した。

本発明が適用される機械学習管理運用システムは、モデル管理の容易化を図ることができるため、このような課題を解決可能である、という効果を奏することができる。

【0028】

換言すると、本発明が適用される機械学習管理運用システムは、タスク毎の学習結果のバージョンと、実行環境の両方をクラウド上で一元的に管理することができるので、このような課題を解決可能である、という効果を奏することができる。

【0029】

機械学習はそれ自体が利益を生み出すものではなく、導入現場で施策を打ったりして効

50

果検証をすることが重要である。即ち、機械学習自体の精度は参考程度の情報である。ここで、バージョンと実行環境を対にして管理しておくことで、機械学習のどのモデルが、実際にどのように顧客の価値につながったかが紐付けられる。前記参考程度の情報しか分からないバージョン管理機構と、実行環境を選択する機構の夫々が有している個別の効果からは、直接的には上記効果は類推できるものではなく、これらが有機的に結びつくことによってはじめて、上記効果は実感できるものである。

#### 【 0 0 3 0 】

図 3 は、学習ジョブ進行状況確認画面の一例を示す図である。

図 3 の例の学習ジョブ進行状況確認画面は、図 1 の出力部 1 7 に表示される。なお、この点は、後述する図 4 の例及び図 5 の例でも同様である。

10

ユーザは、図 3 の例の学習ジョブ進行状況確認画面を用いることで、学習部 1 4 ( 図 2 ) で実行される学習についてのひとつの学習定義 ( Job Definition ) にたいして、複数の学習ジョブの夫々を相互に独立して個別に実行することができる。

ここで、学習定義は、例えば Ver. 1 1 や Ver. 1 2 等の複数のバージョンで管理されており、所定のバージョンの学習定義が上述の「ひとつの学習定義」に相当する。

学習ジョブには、学習部 1 0 4 で学習が実行される際に用いられるもの、即ち、学習条件選択部 1 0 3 により選択されたデータ又は学習方法の少なくともいずれか一方が含まれている。

ユーザは、図 3 の例の学習ジョブ進行状況確認画面を用いることで、複数のジョブの夫々について、「ステータス」、「コメント」、「実行ユーザ」、「進行状況」、「対応する学習定義のバージョン」、「任意で定義した項目と数値」、「経過時間」、「実行開始日時」、「ジョブ実行時のログ」、「学習状況詳細確認」の確認、及び「学習結果のダウンロード」をすることができる。

20

#### 【 0 0 3 1 】

図 4 は、学習ジョブ進行状況確認画面の図 3 とは異なる例を示す図である。

図 4 の例の学習ジョブ進行状況確認画面は、図 3 の例の学習ジョブ進行状況確認画面のうち下側の「Main」に相当するものであるが、図 3 の例とは異なる状況（実行ユーザがコメントを記載している状況）のものである。

例えば実行ユーザは、図 4 の例の学習ジョブ進行状況確認画面を用いることで、各学習ジョブの夫々にコメントを残すことができる。その結果、機械学習管理運用システム（図 1 の情報処理装置 1 ）は、当該学習ジョブを実行した経緯や結果へのコメントを記録することができる。一定期間をおいてからの確認、又は実行ユーザ以外のユーザが学習ジョブを調査するときには有用である。

30

#### 【 0 0 3 2 】

図 5 は、学習ジョブ進行状況確認画面の図 3 及び図 4 とは異なる例を示す図である。

図 5 の例の学習ジョブ進行状況確認画面は、図 3 の例の学習ジョブ進行状況確認画面のうち下側の「Main」に相当するものであるが、図 3 及び図 4 の各例とは異なる状況（実行ユーザがコメントを記載している状況）のものである。

ユーザは、図 5 の例の学習ジョブ進行状況確認画面を用いること（特に同図の太線枠で囲まれている箇所を視認すること）で、「任意で定義した項目と数値」、「ステータス」、及び「進行状況」の指標により、複数ある学習ジョブのうちからもっとも効果が高いと思われる学習ジョブを選択し、又は順序づけをして、それらを図 1 の出力部 1 7 に表示する機能を機械学習管理運用システム（図 1 の情報処理装置 1 ）に発揮させることができる。

40

#### 【 0 0 3 3 】

図 6 は、モデルのバージョン管理画面の一例を示す図である。

図 6 の例のモデルのバージョン管理画面は、図 1 の出力部 1 7 に表示される。なお、この点は、後述する図 7 の例及び図 8 の例でも同様である。

ユーザは、図 6 の例のモデルのバージョン管理画面を用いることで、モデル管理部 1 0 7 により管理されている（モデル DB 3 0 0 やモデル DB 4 0 0 等に記憶されている）モ

50

デルバージョンの更新履歴及び更新毎に使用したソースコードを管理できる。ユーザは、モデルバージョンの夫々の更新において、そのバージョン番号 (Version) 及び、更新時のコメントを記録させることができる。ユーザは、コメントに、更新の経緯や、デプロイした結果のメモ等記録させることができる。

【0034】

図7は、モデルのバージョン管理画面の図6とは異なる例を示す図である。

図7の例のモデルのバージョン確認画面は、図6の例のモデルのバージョン管理画面のうち右上の領域に相当するものであるが、図6の例とは異なる状況 (ユーザが新たなモデルバージョンの作成を所望している状況) のものである。

ユーザは、図7の例のモデルのバージョン管理画面において、“Create Version”のソフトウェアボタンを押下操作することにより、新たなモデルバージョン (モデルの新規バージョン) を作成することができる。

10

【0035】

図8は、モデルのバージョン管理画面の図6及び図7とは異なる例を示す図である。

上述の図7の例のモデルのバージョン管理画面において、“Create Version”のソフトウェアボタンの押下操作がなされると、図1の出力部17に表示される画面が、図8の例のモデルのバージョン管理画面に遷移する。

ユーザは、図8の例のモデルのバージョン管理画面を用いることで、新たなモデルバージョン (モデルの新規バージョン) の作成 (その指示操作) ができると共に、「バージョン番号 (Version)」や「利用する実行環境」の設定その他の設定を指定できる。

20

【0036】

図9は、モデルのバージョン管理画面の図6乃至図8とは異なる例を示す図である。

図9の例のモデルバージョン管理画面は、図8の例のモデルのバージョン管理画面のうち下側の領域に相当するものである。換言すると、ユーザは、新たなモデルバージョン (モデルの新規バージョン) の作成 (その指示操作) をしている際に、スクロール操作等を行うことで、図1の出力部17に表示される画面を、図8の例のモデルのバージョン管理画面から図9の例のモデルのバージョン管理画面に遷移させることができる。

ユーザは、図9の例のモデルのバージョン管理画面において、“Source Code”の下方の灰色の領域に対して、ソースコード (アイコン) をドラッグ&ドロップさせる操作をするか、又は、選択ダイアログから該当ファイル (ソースコード) を選択する操作をすることができる。

30

【0037】

以上、本発明の情報処理システムの一実施形態としての機械学習管理運用システムについて説明した。機械学習管理運用システムは、次の(1)のような情報処理システムに適用されたり、又は次の(2)乃至(11)の夫々のような各種機能を有するようにすることもできる。

(1) クラウドでのモデル学習のデータを利用したAIコンサルティング推薦システム。

(2) 学習結果と共にモデルのソースコードを独立して管理する機能。

(3) Super Awesome Model Platform (s a m p) の機能。

40

(4) モデル学習の達成度に応じた課金プランの機能。

(5) 精度改善を保証したBlue / Greenデプロイメントの機能。

(6) Web API (Application Programming Interface) の確認用インターフェースを有する機能。

(7) 同じモデルをデプロイメントという概念を用いて、用途毎に管理する機能。

(8) 推論のリクエスト (生データ) と推論のレスポンス (推論結果) を自動保存的に保存する機能。

(9) 推論結果に関しての可視化 (モニタリング、グラフ化)、及びアラートの機能。

(10) 推論結果の可視化の機能。

(11) 推論結果の真値チェックの機能。

50

## 【 0 0 3 8 】

( 1 ) クラウドでのモデル学習のデータを利用した A I コンサルティング推薦システム ( 以下、単に「推薦システム」と略記する ) の特徴について、以下に記載する。

推薦システムは、クラウドのみで完結しているので、ユーザの行動履歴を把握できている。

例えば、これくらいのデータとこれくらいのカテゴリに対して、これくらいの精度しか出てないのはおかしい、となった場合に、推薦システムは、コンサルタントに相談しよう、とユーザに推薦することができる。

ユーザは、推薦システムを利用することで、ソースコードを解析して利用しているネットワーク等がわかる。

推薦システムは、チャットボットと比較しての独創性があるものである。

推薦システムは、そこから適切なコンサルティングを推薦できる。

推薦システムは、行動履歴に応じて、困ったポイントをあらかじめ理解した状態でコンサルタントに依頼を出すことができる。

推薦システムは、日本、インド、U S、世界各国に対応できる。

## 【 0 0 3 9 】

推薦システムは、機械学習 / 深層学習のノウハウがないと、学習を実行しても思ったように精度を達成できない、という課題を解決することができる。

その結果、推薦システムは、ユーザの行動履歴及びモデルの学習に用いるソースコードを解析することにより、適切なタイミングで適切な A I コンサルティングを推薦することが可能になる、という効果を奏することができる。

## 【 0 0 4 0 】

ここで、従来において、コンサルティングとのチャンネルを提供するチャットボットのようなソリューションがあったに過ぎない。これに対して、推薦システムは、クラウド上に保管された行動履歴やソースコードを解析した結果から推薦する仕組みを有するものである。

## 【 0 0 4 1 】

推薦システムは、機械学習 / 深層学習の需要にたいして、ノウハウをもつ企業は少ないため、効率的なコンサルティングの必要性は非常に高い、という基本性を有している。

## 【 0 0 4 2 】

また、推薦システムにおいて、コンサルティングとのコンタクトをとるためになんらかのチャンネル ( W e b インターフェースやメール等 ) は顕在化しているはずである。

## 【 0 0 4 3 】

次に、( 2 ) 学習結果と共にモデルのソースコードを独立して管理する機能 ( 以下、「機能 ( 2 ) 」と呼ぶ ) について説明する。

## 【 0 0 4 4 】

機能 ( 2 ) の特徴について、以下に記載する。

従来において、ローカルでの開発だとディレクトリによる管理等、手運用になってしまう。

これに対して、機能 ( 2 ) により、学習に使用したデータセット、コード、パラメータ、ログがバージョン管理されたモデルのソースコードがクラウド上で紐づけられている。

その結果、学習結果とソースコードを別管理できるので、ライセンスの問題を解決できる。

さらに、機能 ( 2 ) は、ソフトウェア著作権とモデルの権利を分けて管理できる。

## 【 0 0 4 5 】

機能 ( 2 ) は、モデル学習の結果として生成される成果物 ( ネットワークの重み等 ) と推論を実行するソースコードの権利関係は異なる可能性がある、という課題を解決することができる。

その結果、機能 ( 2 ) は、モデル学習の結果として生成される成果物と推論を実行するソースコードを別々のコンポーネントとして管理することで、夫々に異なる権利 / ランセ

10

20

30

40

50

ンスを適用できる、という効果を奏することができる。

【0046】

機能(2)は、利用者(ユーザ)である開発者の作成したソースコードと、クラウド上で生成された学習結果の権利関係を明確にすることはプラットフォームにとって重要である、という基本性を有する。

【0047】

機能(2)は、ソースコードと学習結果の取り扱いを見れば分かる、という顕現性を有している。

【0048】

次に、(3) Super Awesome Model Platform (s a m p 10  
)の機能(以下、「機能(3)」と呼ぶ)について説明する。

【0049】

機能(3)の特徴について、以下に記載する。

機能(3)は、入出力データの前処理を利用者(ユーザ)から隠蔽して提供している。

機能(3)は、ユーザのプログラムと実行環境を独立にするための仕組みである。

機能(3)は、HTTP (H y p e r t e x t T r a n s f e r P r o t o c o l  
)である。

機能(3)は、クラウド上でのイベント駆動できる。

機能(3)は、エッジデバイス上での実行できる。

機能(3)は、一ユーザとしての開発者の作業工数の削減ができる。 20

【0050】

次に、(4)モデル学習の達成度に応じた課金プランの機能(以下、「機能(4)」と呼ぶ)について説明する。

【0051】

機能(4)の特徴について、以下に記載する。

機能(4)は、機械学習クラウドオンリーの新しい効果を有する。

機能(4)は、学習結果に満足できない場合はコミットをしないことでコスト削減できる。

機能(4)は、学習に時間がかかる場合は延長できる。

機能(4)は、最終的にキャンセルすることもできる。 30

【0052】

機能(4)は、機械学習/深層学習におけるモデルの学習については「どれくらいの時間」で「どれくらいの精度」が達成できるかが事前に予測困難であり、十分な精度が出ないうちはできるだけコストを抑えたい、という課題を解決することができる。

【0053】

その結果、機能(4)は、次のような効果を奏することができる。

即ち、クラウドでのモデル学習において「オープン」と「コミット」という操作を以下のように定義するものとする。

「オープン」は、学習の開始を表すものとする。「コミット」は、学習を完了し、学習結果からモデルを作成することを表すものとする。 40

この場合、機能(4)は、モデル学習の課金を「オープン」時と「コミット」時に課金されるようにし、「オープン」時の価格を「コミット」時の価格よりも低く抑えることで、顧客(ユーザ)は学習が完了するまでのコストを削減することができる、という効果を奏することができる。

【0054】

ここで、従来において、モデルの学習においては、インフラストラクチャーとして使用するGPUの性能/利用時間によって課金するのが一般的であった。これに対して、機能(4)は、モデル学習の達成度を定義し、それに応じて課金する方式という新しいものである。

【0055】



機能(4)は、機械学習/深層学習のモデル学習がクラウドにシフトする過程においては、GPU14を利用者が意識すること自体が陳腐化すると考えられ、GPU14に応じた課金体系は利用されなくなってくるはずである。

【0056】

機能(4)は、課金プランを確認するという顕現性を有している。

【0057】

次に、(5)精度改善を保証したBlue/Greenデプロイメントの機能(以下、「機能(5)」と呼ぶ)について説明する。

【0058】

機能(5)の特徴について、以下に記載する。

10

機能(5)は、UI(User Interface)上で瞬時に切り替えることができる。

機能(5)は、ひとつ或いは複数のテストデータセットを各エンドポイントにリクエストとして送ることで、異なるバージョン間の精度比較ができる。

機能(5)は、精度比較は可視化されており、ベストなものをプライマリーとして選択できる。

【0059】

機能(5)は、新しいバージョンのAPIにアップデートするときは、以前のバージョンのAPIよりも精度が改善されていることを確認した上で瞬時にアップデート(Blue/Greenデプロイメントによるゼロダウタイムリリース)したい、という課題を解決することができる。

20

その結果、機能(5)は、APIのアップデート時に、同一のデータセットを用いて、現行バージョンと新バージョンの精度の比較をすることができる。比較結果を確認しながら、新バージョンに切り替えるかどうかを決定できる、という効果を奏することができる。

【0060】

ここで、一般的なWebサービスとは異なり、機械学習/深層学習のAPIにおいてはモデルの精度が指標として重要である。

機能(5)は、同一条件下における複数バージョンの精度比較結果をBlue/Greenデプロイメントに用いることができる。

30

【0061】

機能(5)は、上で述べたとおり、機械学習/深層学習のAPIにおいてはモデルの精度が重要であり、一般的なBlue/Greenデプロイメントだけでは十分ではない、という基本性を有する。

【0062】

機能(5)は、WebAPIのデプロイ方法で確認できる、という顕現性を有する。

【0063】

次に、(6)WebAPIの確認用インターフェースを有する機能(以下、「機能(6)」と呼ぶ)について説明する。

【0064】

40

機能(6)の特徴について、以下に記載する。

機能(6)のインターフェースは、入力画像と出力の比較のしやすさを維持しつつ、視認性を上げるために夫々の描画領域を独立させている。

機能(6)は、入力画像の加工ができる。

機能(6)は、アノテーションサービスと紐づいているデータレイクのファイルを入力等に使える。

機能(6)は、結果と比較もできる。

機能(6)は、出力結果を解析してbounding boxを入力画像に重ねたり、出力を工夫できる。

【0065】

50

機能(6)は、機械学習/深層学習モデルの教師データを含む入力と出力の結果が整合していることを、教師データと推論の結果の非構造化テキストデータを照合するのではなく、視覚的に確認したい、という課題を解決することができる。

【0066】

その結果、機能(6)は、次のような効果を奏することができる。

機能(6)により、Webブラウザで利用可能な確認用のインターフェースが用意できる。

このインターフェースは、入力データ描画エリアと出力データ描画エリアの分かれており、夫々を独立してスクロール可能である。これによって、入力画像と出力の比較のしやすさを維持しつつ、視認性を上げている。

10

入力データとしては以下のデータ(a)乃至(c)が利用可能である。

(a)プラットフォームに組み込まれたサンプル画像。

(b)アップロードした画像。

(c)データレイクに保存された画像と教師データ(非構造化テキストデータ)。

ここで、データ(c)の教師データがある場合は、非構造化テキストデータを解析して、色分けされた矩形として画像に重ね合わせることで視認性を向上させる。

出力データとしては非構造化テキストデータ又は画像を推論結果として取得することができる。

その場合、画像の場合はそのまま描画する。また、非構造化テキストデータを解析し、色分けされた矩形として画像に重ね合わせる。

20

機能(6)は、これらのインターフェースを提供することができるので、入力と出力の比較が容易になるという効果を奏することができる。

【0067】

機能(6)は、入力の教師データ/画像及びAPIの出力の非構造化テキストデータを自動で解析して画像データとして描画するインターフェースは他に類を見ないものである。

【0068】

機能(6)は、デプロイしたWeb APIの確認を容易にすることは重要という基本性を有する。

【0069】

機能(6)は、インターフェースなので確認可能という顕現性を有する。

30

【0070】

次に、(7)同じモデルをデプロイメントという概念を用いて、用途毎に管理する機能(以下、「機能(7)」と呼ぶ)について説明する。

【0071】

機能(7)は、店舗ごと、エッジ毎に分けられるという特徴を有している。

【0072】

機能(7)は、同一のモデルを異なる用途に使い回したいが、用途毎に異なる設定やキャパシティを管理するのは困難である、という課題を解決することができる。

その結果、機能(7)は、あるモデルを複数のインフラストラクチャー上で実行でき、実行に必要なインフラストラクチャーの規模やプログラムの設定を「デプロイメント」という単位で管理できる、という効果を奏することができる。

40

【0073】

機能(7)は、SaaSサービス等では異なる要求に応じて、同一のモデルの性能や設定を変更する必要があるため重要である、という基本性を有する。

【0074】

機能(7)は、開発者が設定するものなので分かる、という顕現性を有する。

【0075】

次に、(8)推論のリクエスト(生データ)と推論のレスポンス(推論結果)を自動保存的に保存する機能(以下、「機能(8)」と呼ぶ)について説明する。

50

## 【 0 0 7 6 】

機能（ 8 ）は、推論して終了ではなく、後ほど推論結果が正しいのか確認するため、精度を向上させるための学習用データとして、その他精度向上させる用途として推論のリクエスト（生データ）と推論のレスポンス（推論結果）を自動保存的に保存する、という特徴を有する。

## 【 0 0 7 7 】

機能（ 8 ）は、次のような課題を解決することができる。

即ち、推論は、100%の精度が出るわけでも、100%正しいわけでもない。そのため、推論したデータと結果を確認しないと正しい精度を測ることができない。さらに精度を向上させるアクションも難しい。あとから精度をチェックするための仕組みが必要となる。

10

機能（ 8 ）は、このような課題を解決することができるので、推論した結果を自動的に保存することで、あとから簡単に精度のチェックができる、という効果を奏することができる。

## 【 0 0 7 8 】

次に、（ 9 ）推論結果に関しての可視化（モニタリング、グラフ化）、及びアラートの機能（以下、「機能（ 9 ）」と呼ぶ）について説明する。

## 【 0 0 7 9 】

機能（ 9 ）は、推論した結果の精度が意図した基準を満たしているのかグラフ化し、満たしていない場合はアラートを送信する、という特徴を有している。

20

## 【 0 0 8 0 】

機能（ 9 ）は、次のような課題を解決することができる。

即ち、推論を行う外部環境で一定水準を満たさないケースが起こりうる。例えばカメラで人数カウントを行う場合、カメラ設置環境の証明を変えるだけでも精度に大きく関わり精度が下がる場合がある。その場合は異常検知して対処する必要がある。

機能（ 9 ）は、このような課題を解決することができるので、異常にいち早く検知し、対処することができる、という効果を奏することができる。

## 【 0 0 8 1 】

次に、（ 1 0 ）推論結果の可視化の機能（以下、「機能（ 1 0 ）」と呼ぶ）について説明する。

30

## 【 0 0 8 2 】

機能（ 1 0 ）は、画像を推論した結果、この画像は何と分類したのかというラベルと画像をサムネイル表示できる。分類以外に、セグメンテーションやバウンディングボックス等も対応する、という特徴を有している。

## 【 0 0 8 3 】

機能（ 1 0 ）は、次のような課題を解決することができる。

即ち、推論した結果が正しいのかを数値では測れないケースにおいて、人の目で確認し精度をチェックする必要がある。確認しやすいようにデータの保存とデータのチェックが必要になる。

機能（ 1 0 ）は、データのチェックを行うための開発やワークフローが不要になる、という効果を奏することができる。

40

## 【 0 0 8 4 】

次に、（ 1 1 ）推論結果の真値チェックの機能（以下、機能（ 1 1 ）と呼ぶ）について説明する。

## 【 0 0 8 5 】

機能（ 1 1 ）の特徴について、以下に記載する。

即ち、推論した結果が正しいのかを数値では測れないケースにおいて、人の目で確認し精度をチェックする必要がある。推論した結果をサンプリングした一部のデータに対して人手のアノテーション代行を行い、真値をチェックする。

## 【 0 0 8 6 】

50

機能(11)は、推論した結果が正しいのかを数値では測れないケースにおいて、どのように精度を確認するのか、という課題を解決することができる。

その結果、機能(11)は、真値チェックするための人のアサイン、管理コストを下げることができる、という効果を奏することができる。

【0087】

以上本発明の一実施形態について説明したが、本発明は、上述の実施形態に限定されるものではなく、本発明の目的を達成できる範囲での変形、改良等は本発明に含まれるものである。

【0088】

例えば、上述した一連の処理は、ハードウェアにより実行させることもできるし、ソフトウェアにより実行させることもできる。

換言すると、図2の機能的構成は例示に過ぎず、特に限定されない。

即ち、上述した一連の処理を全体として実行できる機能が機械学習管理運用システムに備えられていれば足り、機械学習管理運用システムの構成は特に図1の例に限定されず、かつ、この機能を実現するためにどのような機能ブロックを用いるのかは特に図2の例に限定されない。また、機能ブロックの存在場所も、図2に特に限定されず、任意でよい。例えば、機械学習管理運用システムはサーバとクライアントとを含む構成としてもよく、この場合、図1の情報処理装置1の各機能ブロックをサーバ又はクライアントに分散させて移譲させてもよい。

また、1つの機能ブロックは、ハードウェア単体で構成してもよいし、ソフトウェア単体で構成してもよいし、それらの組み合わせで構成してもよい。

【0089】

また例えば、一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、コンピュータ等にネットワークや記録媒体からインストールされる。

コンピュータは、専用のハードウェアに組み込まれているコンピュータであってもよい。

また、コンピュータは、各種のプログラムをインストールすることで、各種の機能を実行することが可能なコンピュータ、例えばサーバの他汎用のスマートフォンやパーソナルコンピュータであってもよい。

【0090】

また例えば、このようなプログラムを含む記録媒体は、ユーザにプログラムを提供するために装置本体とは別に配布される図示せぬリムーバブルメディアにより構成されるだけでなく、装置本体にあらかじめ組み込まれた状態でユーザに提供される記録媒体等で構成される。

【0091】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、その順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムの用語は、複数の装置や複数の手段等より構成される全体的な装置を意味するものとする。

【0092】

以上を換言すると、本発明が適用される情報処理システムは、次のような構成を有していれば足り、各種各様な実施の形態を取ることができる。

即ち、本発明が適用される情報処理システムは、

学習データを入力するデータ入力手段(例えば図2の学習データ入力部101)と、

学習方法を入力する学習方法入力手段(例えば図2の学習方法入力部102)と、

前記データ入力手段により入力された所定の前記学習データと、前記学習方法入力手段により入力された所定の前記学習方法の組を選択する学習条件選択手段(例えば図2の学習条件選択部103)と、

前記学習条件選択手段により選択された前記学習データを含むデータ又は学習方法の少なくともいずれか一方を用いて学習を実行する学習手段（例えば図２の学習部１０４）と

、  
前記学習手段により実行された学習の結果の採用又は不採用を決定する採用決定手段（例えば図２の採用決定部１０５）と、

推論方法を入力する推論入力手段（例えば図２の推論入力部１０６）と、

前記採用決定手段により採用が決定された前記学習の結果と、前記推論入力手段により入力された前記推論方法とを組にしてモデルバージョンとし、前記モデルバージョンを１つ又はそれ以上を、管理するタスク毎に用意するモデル管理手段（例えば図２のモデル管理部１０７）と、

前記モデル管理手段に紐づいた実行選択手段であって、当該モデル管理手段の管理するモデルバージョンと実行環境を選択する実行選択手段（例えば図２の実行選択部１０８）と、

前記実行選択手段により選択された前記実行環境において、前記実行選択手段により選択された前記モデルバージョンを実行させる実行選択手段（例えば図２の実行実施部１０９）と、

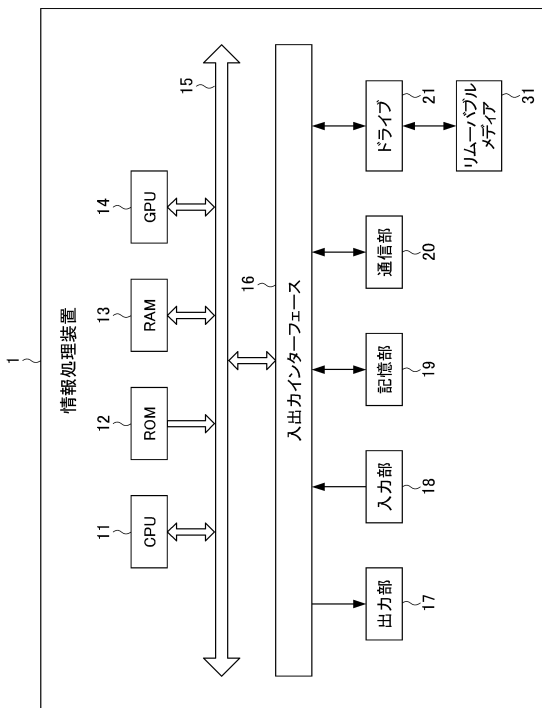
を備える。

【符号の説明】

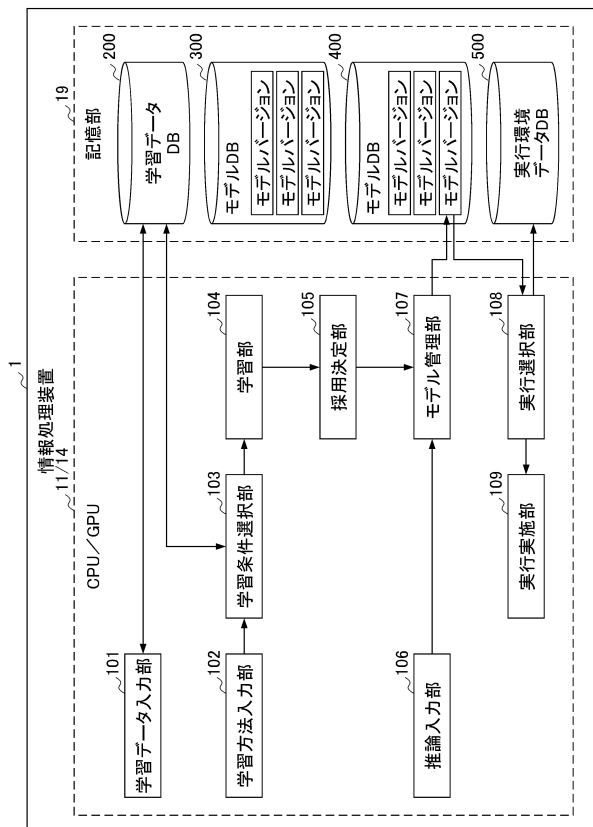
【００９３】

１・・・情報処理装置、１１・・・ＣＰＵ、１４・・・ＧＰＵ、１０１・・・学習データ入力部、１０２・・・学習方法入力部、１０３・・・学習条件選択部、１０４・・・学習部、１０５・・・採用決定部、１０６・・・推論入力部、１０７・・・モデル管理部、１０８・・・実行選択部、１０９・・・実行実施部、２００・・・学習データＤＢ、３００、４００、５００・・・モデルＤＢ、５００・・・実行環境データＤＢ

【図１】



【図２】



10

20

【図 3】

HOME PLATFORM \*\*\*\*\* English \*\*\*\*\*

Home > Training Job Definitions > Job Definition (Job Definition ID: \*\*\*\*\*)

Job Definition (Job Definition ID: \*\*\*\*\*)

Refresh Create Training Job

Job	Creator	TensorBoard	Progress	Validation Accuracy	Validation Loss	When
cat and dog classification(enhanced)	La la la...	TensorBoard ver.12	100.0%	49.2 %	0.0036	24 20:39:46
La la la...	La la la...	TensorBoard ver.12	100.0%	49.3 %	0.0036	24 20:04:7
テストのために実行しました。	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:13:42
cat and dog classification	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:27:13

【図 4】

Main

Job	Creator	TensorBoard	Progress	Validation Accuracy	Validation Loss	When
cat and dog classification(enhanced)	La la la...	TensorBoard ver.12	100.0%	49.2 %	0.0036	24 20:39:46
La la la...	La la la...	TensorBoard ver.12	100.0%	49.3 %	0.0036	24 20:04:7
テストのために実行しました。	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:13:42
cat and dog classification	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:27:13

Main

Job	Creator	TensorBoard	Progress	Validation Accuracy	Validation Loss	When
cat and dog classification(enhanced)	La la la...	TensorBoard ver.12	100.0%	49.2 %	0.0036	24 20:39:46
La la la...	La la la...	TensorBoard ver.12	100.0%	49.3 %	0.0036	24 20:04:7
テストのために実行しました。	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:13:42
cat and dog classification	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:27:13

【図 5】

Main

Job	Creator	TensorBoard	Progress	Validation Accuracy	Validation Loss	When
cat and dog classification(enhanced)	La la la...	TensorBoard ver.12	100.0%	49.2 %	0.0036	24 20:39:46
La la la...	La la la...	TensorBoard ver.12	100.0%	49.3 %	0.0036	24 20:04:7
テストのために実行しました。	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:13:42
cat and dog classification	La la la...	TensorBoard ver.11	100.0%	24.9 %	0.0036	14 03:27:13

【図 6】

Home > Models > default > Model Versions

Model Versions

Search Create Version Refresh

Version	ID	Runtime	Handler	Created
0.1.4	ver-0a0cccd128442f	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.1.3	ver-0a24847081214254	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.1.2	ver-0b0da88d4a14a39	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.1.1	ver-a300cda1912443d	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.1.0	ver-0a24847081214254	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.0.5	ver-1442ca2a3c4c77b	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.0.4	ver-0504e9f4d46af	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.0.3	ver-7a02077801ba4591	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.0.2	ver-7a02077801ba4591	abja-inc/keras0.1.0	mainhandler	2017/09/07 21:08:21 +0900
0.0.1	ver-0853e71c4d4231	abja-inc/minimal0.1.0	mainhandler	2017/09/07 21:08:21 +0900

【図 7】

Create Version

Refresh

【図 8】

Create Model Version

Version \*

1.0.0

Runtime \*

abeja-inc/keras:0.1.0

Model Handler

mainhandler

TRAINING ARTIFACT

Job Definition

Select training job definition

Training Job

Loading artifacts...

SOURCE CODE

Source Code

【図 9】

SOURCE CODE

Source Code

Drag and files or

Select files

☐ Use latest source code instead of uploading source code

Cancel

Create Model Version

---

フロントページの続き

(56)参考文献 特開2018-010475(JP,A)

TensorFlow Servingで機械学習モデルをプロダクション環境で運用する,[online],2017年  
12月23日,all 12 pages,URL,https://web.archive.org/web/20171223064239/https://  
developers.freee.co.jp/serve-ml-model-by-tensorflow-serving

(58)調査した分野(Int.Cl.,DB名)

G06N 20/00