(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 968 161 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
25.01.2023 Bulletin 2023/04

(21) Application number: **21164387.9**

(22) Date of filing: **23.03.2021**

(51) International Patent Classification (IPC):
G06N 3/08 *(2006.01)* ***G06F 11/07*** *(2006.01)*

(52) Cooperative Patent Classification (CPC):
G06F 11/0775; G06N 3/08

(54) METHOD AND APPARATUS FOR GENERATING ERROR REPORTING CONTENT OF DEEP LEARNING FRAMEWORK

VERFAHREN UND VORRICHTUNG ZUR ERZEUGUNG VON FEHLERMELDEINHALT EINES TIEFENLERNRAHMENS

PROCÉDÉ ET APPAREIL DE GÉNÉRATION DE CONTENU DE RAPPORTS D'ERREURS DE CADRE D'APPRENTISSAGE PROFOND

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **15.09.2020 CN 202010969161**

(43) Date of publication of application:
16.03.2022 Bulletin 2022/11

(73) Proprietor: **Beijing Baidu Netcom Science and Technology Co., Ltd.**
Beijing 100085 (CN)

(72) Inventors:
- **ZHUO, Wei**
  Beijing, Beijing 100085 (CN)
- **LAN, Xiang**
  Beijing, Beijing 100085 (CN)
- **CHEN, Weihang**
  Beijing, Beijing 100085 (CN)
- **LUO, Tao**
  Beijing, Beijing 100085 (CN)

(74) Representative: **dompatent von Kreisler Selting Werner -**
Partnerschaft von Patent- und Rechtsanwälten mbB
Deichmannhaus am Dom
Bahnhofsvorplatz 1
50667 Köln (DE)

(56) References cited:
EP-A2- 0 735 471 **CN-A- 105 550 059**

EP 3 968 161 B1

Description

Field of the Disclosure

[0001] The present application relates to the field of computer technologies, and particularly to a method and apparatus for generating error reporting content of a deep learning framework, an electronic device, a readable storage medium, and a computer program product in the field of deep learning technologies.

Background of the Disclosure

[0002] As a basic research tool in the field of artificial intelligence, a deep learning framework greatly facilitates developers in related fields. However, due to the complexity of the deep learning framework itself, when using the deep learning framework to train a deep learning model, a developer may call a large number of third-party library application programming interfaces (APIs) to implement corresponding functions.

[0003] A calling failure may occur when the deep learning framework calls the third-party library APIs. However, in the prior art, after the deep learning framework fails to call the API, only an error code returned by a third-party library is displayed to the developer, resulting in that the developer cannot learn more detailed error information, thereby increasing the development cost of the developer and reducing the development efficiency of the developer.

[0004] A patent application EP 0 735 471 A2, published on October 2, 1996, entitled "Method and apparatus for interpreting exceptions in a distributed object system", describes a computer-implemented method for interpreting exceptions received by the host computing system intended for use in a distributed object computing environment, wherein an exception identifier is received that uniquely identifies an exception raised by a remotely located device, and the exception message file is accessed utilizing the message identifier to locate a message string that corresponds to the exception identifier.

[0005] A patent application CN 105 550 059 A, published on May 4, 2016, entitled "Error code conversion method and equipment", relates to Error code conversion, which describes searching the predefine error code matched with described error code and determining described third party system carry out the initiation channel of concluding the business when finding the predefine error code matched with described error code. The described predefine error code is sent to described initiation channel.

Summary of the Disclosure

[0006] A technical solution adopted by the present application to solve the technical problem is to provide a computer-implemented method for generating error reporting content of a deep learning framework including: acquiring an error code returned by a third-party library when an error occurs in calling of a third-party library application programming interface and error information corresponding to the error code, wherein the error code is used for identifying a type of an error when the deep learning framework fails to call the third-party library API, and the error information is used for describing the cause of the error and the solution for the error;determining a programming language used by the deep learning framework, generating an error file which corresponds to the programming language, according to the error code and the error information corresponding thereto, and packaging the error file into the deep learning framework; running the deep learning framework, and in response to the deep learning framework receiving the error code, extracting, from the error file, error information corresponding to the received error code; and generating error reporting content according to the error information,recording the generation time of the error file after the generating the error file according to the error code and the error information corresponding thereto; and updating the error code and the error information corresponding thereto in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

[0007] A technical solution adopted by the present application to solve the technical problem is to provide an apparatus for generating error reporting content of a deep learning framework, including: an acquisition unit configured to acquire an error code returned by a third-party library when an error occurs in calling of a third-party library application programming interface (API) and error information corresponding to the error code, wherein the error code is used for identifying a type of an error when the deep learning framework fails to call the third-party library API, and the error information is used for describing the cause of the error and the solution for the error;a processing unit configured to determine a programming language used by the deep learning framework, generate an error file which corresponds to a programming language, according to the error code and the error information corresponding thereto, and package the error file into the deep learning framework;an error reporting unit configured to run the deep learning framework, and in response to the deep learning framework receiving the error code, extract, from the error file, error information corresponding to the received error code; and a generation unit configured to generate error reporting content according to the error information, record the generation time of the error file after the generating the error file according to the error code and the error information corresponding thereto, and update the error code and the error information corresponding thereto in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

[0008] A non-transitory computer-readable storage medium storing computer instructions is provided,

wherein the computer instructions are used to make the computer to perform the method as described above.

[0009] A computer program product is provided, comprising instructions which, when the program is executed by a computer, cause the computer to perform the method as described above.

[0010] One embodiment in the present application has the following advantages or beneficial effects: the present application can automatically generate error reporting content and make the generated error reporting content include richer information. Since a technical means of extracting, from an error file, error information corresponding to an error code is employed, the technical problem in the prior art that the deep learning framework only obtains an error code returned by a third-party library when failing to call a third-party library API is overcome, and a technical effect of automatically generating error reporting content including richer information is achieved.

[0011] Other effects of the above alternatives will be described below with reference to specific embodiments.

Brief Description of Drawings

[0012] The accompanying drawings are intended to better understand the solution and do not constitute limitations on the present application. In the drawings,

Fig. 1 is a schematic diagram of a first embodiment according to the present application.

Fig. 2 is a schematic diagram of a second embodiment according to the present application.

Fig. 3 is a block diagram of an electronic device configured to implement a method for generating error reporting content of a deep learning framework according to an embodiment of the present application.

Detailed Description of Preferred Embodiments

[0013] Exemplary embodiments of the present application are described below with reference to the accompanying drawings, including various details of the embodiments of the present application to facilitate understanding, and should be considered as exemplary only. Similarly, for clarity and simplicity, descriptions of well-known functions and structures are omitted in the following description.

[0014] Fig. 1 is a schematic diagram of a first embodiment according to the present application. As shown in Fig. 1, a method for generating error reporting content of a deep learning framework according to the present embodiment specifically includes following steps:

S101: An error code and error information corresponding to the error code are acquired.

S102: An error file is generated according to the error code and the error information corresponding thereto, and the error file is packaged into the deep learning framework.

S103: The deep learning framework is run, and in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, error information corresponding to the received error code is extracted from the error file.

S104: Error reporting content is generated according to the error information.

[0015] According to the method for generating error reporting content of a deep learning framework in the present embodiment, an error file is generated according to an error code and error information corresponding to the error code, so as to obtain a deep learning framework including the error file, so that error information corresponding to an error code returned by a third-party library is extracted from an error reporting file when the deep learning framework fails to call a third-party library API during its running, and error reporting content is then generated according to the error information, which can automatically generate error reporting content and make the generated error reporting content include richer information.

[0016] The error code acquired by performing step S101 in the present embodiment is used to identify an error type when the deep learning framework fails to call the third-party library API. When errors occur due to different reasons in the calling of the API, the errors correspond to different error codes. Since the third-party library may pre-configure different error codes for different error types, an error code may be acquired by accessing the third-party library in the present embodiment.

[0017] After S101 is performed to acquire the error code in the present embodiment, error information corresponding to the error code is acquired. The acquired error information is used to describe in detail the cause of the error, the solution, and other content. In the present embodiment, a webpage, a document, a website and other content corresponding to the error code may be automatically crawled, through crawler code, from the Internet as the error information corresponding thereto, or the error information corresponding thereto may be written manually by a developer.

[0018] It is understandable that the number of the error code acquired by performing S101 in the present embodiment may be one or more. If a plurality of error codes are acquired, error information corresponding to the error codes needs to be acquired respectively in the present embodiment.

[0019] After S101 is performed to acquire the error code and the error information corresponding thereto in the present embodiment, S 102 is performed to generate an error file according to the error code and the error information corresponding thereto and to package the generated error file into the deep learning framework, so that the deep learning framework can parse the error information corresponding thereto.

[0020] Specifically, in the present embodiment, when S102 is performed to generate an error file according to the error code and the error information corresponding thereto, a following implementation is employed: determining a programming language used by the deep learning framework, such as a Python language, a C++ language, or a Java language; and generating an error file corresponding to the determined programming language according to the error code and the error information corresponding thereto.

[0021] In the present embodiment, the error code and the error information corresponding thereto may also be serialized by using an existing protobuf protocol, and the serialized content is taken as the error file.

[0022] In other words, in the present invention, a language of the error file generated by performing S102 corresponds to a programming language used by the deep learning framework, which avoids the failure of error code parsing caused by the inconsistency between the error file and the language of the deep learning framework, and improves the success rate of generating the error reporting content.

[0023] Since the API in the third-party library is often updated, the error code corresponding to the API in the generated error files has certain timeliness. If an updated error code is not included in the error file, the deep learning framework cannot parse error information corresponding to the error code.

[0024] In order to ensure as much as possible that the error file includes the latest error code in the third party-library, in the present embodiment, after the error file is generated by performing S102, following content is further included: recording the generation time of the error file; and updating the error code and the error information corresponding thereto in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

[0025] In addition, in order to avoid the occupation of more storage space in the deep learning framework due to a too large error reporting file, in the present embodiment, after S102 is performed to package the error file into the deep learning framework, a following optional implementation may be employed: compressing the error file; and packaging a compression result of the error file into the deep learning framework, that is, the deep learning framework includes the compressed error file.

[0026] In the present embodiment, steps S101 and S 102 are completed before the running of the deep learning framework, so that when S 103 is performed to run the deep learning framework in the present embodiment, in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, error information corresponding to the received error code can be extracted from the error file.

[0027] In the present embodiment, when S 103 is performed to extract, from the error file, error information corresponding to the received error code, the extraction may be performed immediately after an error code returned by a third-party library is received.

[0028] In order to ensure the stability during the running of the deep learning framework, thee deep learning framework may have a mechanism for multiple retries when calling a third-party library API. That is, if the calling still fails after multiple retries, it is determined that the deep learning framework fails to call the third-party library API, and then error information corresponding to an error code returned by the third-party library due to the failure of the calling is acquired.

[0029] Therefore, in the present embodiment, when S 103 is performed to, in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, extracting, from the error file, error information corresponding to the received error code, a following optional implementation may be employed: determining the number of retries corresponding to the third-party library API currently called by the deep learning framework; and in response to the deep learning framework receiving an error code returned by the third-party library when an error occurs in calling of the third-party library API after the number of retries, extracting, from the error file, error information corresponding to the received error code.

[0030] In other words, in the present embodiment, the deep learning framework may extract the error information corresponding thereto from the error file only if the calling still fails after multiple tries, which can improve the stability of the deep learning framework, avoid the waste of computing resources and improve the accuracy of error information acquisition.

[0031] In addition, in the present embodiment, if the error file is compressed when S102 is performed, when S 103 is performed to extract, from the error file, error information corresponding to the received error code in the present embodiment, a following optional implementation may be employed: determining whether a decompression result of the error file is comprised in the deep learning framework; if yes, extracting, from the decompression result, the error information corresponding to the received error code; and otherwise, decompressing the error file, and then extracting, from a decompression result, the error information corresponding to the received error code.

[0032] For example, in the present embodiment, if the error file is obtained by serializing the error code and the error information corresponding thereto by using a protobuf protocol when S102 is performed, when S 103 is performed in the present embodiment, the error file is deserialized by using the protobuf protocol at first, and then error information corresponding to the received error code is extracted from a deserialization result.

[0033] In the present embodiment, after S 103 is performed to extract, from the error file, error information corresponding to the received error code, S 104 is performed to generate error reporting content according to

the extracted error information and to further display the error reporting content to a user, for example, the error reporting content is printed or directly displayed on a screen.

[0034] It is understandable that in the present embodiment, the extracted error information may be directly taken as error content when S104 is performed. In order to further enrich information included in the error content, in the present embodiment, when S 104 is performed to generate error reporting content according to the error information, a following optional implementation may be employed: splicing preset particular information with the error information, and taking a splicing result as the error reporting content.

[0035] The preset particular information in the present embodiment may be call stack information, call number information and call time information for the deep learning framework to call the third-party library API this time.

[0036] With the above method according to the present embodiment, when the deep learning framework receives an error code returned by a third-party library when an error occurs in calling of a third-party library API, the error code can be automatically parsed according to an error file included in the deep learning framework, so as to acquire richer error reporting content including error information, which greatly reduces the development cost of a developer and improves the development efficiency of the developer.

[0037] Fig. 2 is a schematic diagram of a second embodiment according to the present application. As shown in Fig. 2, an apparatus for generating error reporting content of a deep learning framework in the present embodiment includes:

an acquisition unit 201 configured to acquire an error code and error information corresponding to the error code;

a processing unit 202 configured to generate an error file according to the error code and the error information corresponding thereto, and packaging the error file into the deep learning framework;

an error reporting unit 203 configured to run the deep learning framework, and in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, extract, from the error file, error information corresponding to the received error code; and

a generation unit 204 configured to generate error reporting content according to the error information.

[0038] The error code acquired by the acquisition unit 201 is used to identify an error type when the deep learning framework fails to call the third-party library API. When errors occur due to different reasons in the calling of the API, the errors correspond to different error codes. Since the third-party library may pre-configure different error codes for different error types, the acquisition unit 201 may acquire an error code by accessing the third-party library.

[0039] After the acquisition unit 201 acquires the error code, error information corresponding to the error code is acquired. The acquired error information is used to describe in detail the cause of the error, the solution, and other content. The acquisition unit 201 may automatically crawl, through crawler code, a webpage, a document, a website and other content corresponding to the error code from the Internet as the error information corresponding thereto, or the acquisition unit 201 may acquire error information corresponding to the error code through manual writing by a developer.

[0040] It is understandable that the number of the error code acquired by the acquisition unit 201 may be one or more. If a plurality of error codes are acquired, the acquisition unit 201 needs to acquire error information corresponding to the error codes respectively.

[0041] After the acquisition unit 201 acquires the error code and the error information corresponding thereto in the present embodiment, the processing unit 202 generates an error file according to the error code and the error information corresponding thereto and packages the generated error file into the deep learning framework, so that the deep learning framework can parse the error information corresponding thereto.

[0042] Specifically, when the processing unit 202 generates an error file according to the error code and the error information corresponding thereto, a following implementation is employed: determining a programming language used by the deep learning framework; and generating an error file corresponding to the determined programming language according to the error code and the error information corresponding thereto.

[0043] The processing unit 202 may also serialize the error code and the error information corresponding thereto by using an existing protobuf protocol, and the serialized content is taken as the error file.

[0044] In other words, a language of the error file generated by the processing unit 202 can correspond to a programming language used by the deep learning framework, which avoids the failure of error code parsing caused by the inconsistency between the error file and the language of the deep learning framework, and improves the success rate of generating the error reporting content.

[0045] Since the API in the third-party library is often updated, the error code corresponding to the API in the generated error files has certain timeliness. If an updated error code is not included in the error file, the deep learning framework cannot parse error information corresponding to the error code.

[0046] In order to ensure as much as possible that the error file includes the latest error code in the third party-library, after the error file is generated by the processing unit 202, following content is further included: recording the generation time of the error file; and updating the error code and the error information corresponding there-

to in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

[0047] In addition, in order to avoid the occupation of more storage space in the deep learning framework due to a too large error reporting file, after the processing unit 202 packages the error file into the deep learning framework, a following optional implementation may be employed: compressing the error file; and packaging a compression result of the error file into the deep learning framework, that is, the deep learning framework includes the compressed error file.

[0048] When the error reporting unit 203 extracts, from the error file, error information corresponding to the received error code, the extraction may be performed immediately after an error code returned by a third-party library is received.

[0049] In order to ensure the stability during the running of the deep learning framework, thee deep learning framework may have a mechanism for multiple retries when calling a third-party library API. That is, if the calling still fails after multiple retries, it is determined that the deep learning framework fails to call the third-party library API, and then error information corresponding to an error code returned by the third-party library due to the failure of the calling is acquired.

[0050] Therefore, when the processing unit 203, in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, extracts, from the error file, error information corresponding to the received error code, a following optional implementation may be employed: determining the number of retries corresponding to the third-party library API currently called by the deep learning framework; and in response to the deep learning framework receiving an error code returned by the third-party library when an error occurs in calling of the third-party library API after the number of retries, extracting, from the error file, error information corresponding to the received error code.

[0051] In other words, in the present embodiment, the deep learning framework may extract the error information corresponding thereto from the error file only if the calling still fails after multiple tries, which can improve the stability of the deep learning framework, avoid the waste of computing resources and improve the accuracy of error information acquisition.

[0052] In addition, if the processing unit 202 compresses the error file, when the error reporting unit 203 extracts, from the error file, error information corresponding to the received error code, a following optional implementation may be employed: determining whether a decompression result of the error file is comprised in the deep learning framework; if yes, extracting, from the decompression result, the error information corresponding to the received error code; and otherwise, decompressing the error file, and then extracting, from a decompression result, the error information corresponding to the received error code.

[0053] After the error reporting unit 203 extracts, from the error file, error information corresponding to the received error code, the generation unit 204 generates error reporting content according to the extracted error information and further displays the error reporting content to a user.

[0054] It is understandable that the generation unit 204 may directly take the extracted error information as error content. In order to further enrich information included in the error content, when the generation unit 204 generates error reporting content according to the error information, a following optional implementation may be employed: splicing preset particular information with the error information, and taking a splicing result as the error reporting content.

[0055] The preset particular information in the generation unit 204 may be call stack information, call number information and call time information for the deep learning framework to call the third-party library API this time.

[0056] According to embodiments of the present application, the present application further provides an electronic device and a readable storage medium.

[0057] As shown in Fig. 3, it is a block diagram of an electronic device for a method for generating error reporting content of a deep learning framework according to an embodiment of the present application. The electronic device is intended to represent various forms of digital computers, such as laptops, desktops, workbenches, personal digital assistants, servers, blade servers, mainframe computers and other suitable computers. The electronic device may further represent various forms of mobile devices, such as personal digital assistant, cellular phones, smart phones, wearable devices and other similar computing devices. The components, their connections and relationships, and their functions shown herein are examples only, and are not intended to limit the implementation of the present application as described and/or required herein.

[0058] As shown in Fig. 3, the electronic device includes: one or more processors 301, a memory 302, and interfaces for connecting various components, including high-speed interfaces and low-speed interfaces. The components are connected to each other by using different buses and may be installed on a common motherboard or otherwise as required. The processor may process instructions executed in the electronic device, including instructions stored in the memory or on the memory to display graphical information of a graphical user interface (GUI) on an external input/output apparatus (such as a display device coupled to the interfaces). In other implementations, a plurality of processors and/or buses may be used together with a plurality of memories, if necessary. Similarly, a plurality of electronic devices may be connected, each of which provides some necessary operations (for example, as a server array, a set of blade servers, or a multiprocessor system). One processor 301 is taken as an example is Fig. 3.

[0059] The memory 302 is the non-instantaneous computer-readable storage medium according to the present application. The memory stores instructions executable by at least one processor to make the at least one processor perform the method for generating error reporting content of a deep learning framework according to the present application. The non-instantaneous computer-readable storage medium according to the present application stores computer instructions. The computer instructions are used to make a computer perform the method for generating error reporting content of a deep learning framework according to the present application.

[0060] The memory 302, as a non-instantaneous computer-readable storage medium, may be configured to store non-instantaneous software programs, non-instantaneous computer executable programs and modules, for example, program instructions/modules corresponding to the method for generating error reporting content of a deep learning framework in the embodiment of the present application (e.g., the acquisition unit 201, the processing unit 202, the error reporting unit 203, and the generation unit 204 shown in Fig. 2). The processor 301 runs the non-instantaneous software programs, instructions and modules stored in the memory 302 to execute various functional applications and data processing of a server, that is, to implement the method for generating error reporting content of a deep learning framework in the above method embodiment.

[0061] The memory 302 may include a program storage area and a data storage area. The program storage area may store an operating system and an application required by at least one function; and the data storage area may store data created according to use of the electronic device. In addition, the memory 302 may include a high-speed random access memory, and may further include a non-instantaneous memory, for example, at least one disk storage device, a flash memory device, or other non-instantaneous solid-state storage devices. In some embodiments, the memory 302 optionally includes memories remotely disposed relative to the processor 301. The remote memories may be connected, over a network, to the electronic device for the method for generating error reporting content of a deep learning framework. Examples of the network include, but are not limited to, the Internet, intranets, local area networks, mobile communication networks and combinations thereof.

[0062] The electronic device for the method for generating error reporting content of a deep learning framework may further include: an input apparatus 303 and an output apparatus 304. The processor 301, the memory 302, the input apparatus 303 and the output apparatus 304 may be connected through a bus or in other manners. In Fig. 3, the connection through a bus is taken as an example.

[0063] The input apparatus 303 may receive input numerical information or character information, and generate key signal input related to user setting and function control of the electronic device for the method for generating error reporting content of a deep learning framework, for example, input apparatuses such as a touch screen, a keypad, a mouse, a trackpad, a touch pad, a pointer, one or more mouse buttons, a trackball, and a joystick. The output apparatus 304 may include a display device, an auxiliary lighting apparatus (e.g., an LED) and a tactile feedback apparatus (e.g., a vibration motor). The display device may include, but is not limited to, a liquid crystal display (LCD), a light-emitting diode (LED) display, and a plasma display. In some implementations, the display device may be a touch screen.

[0064] Various implementations of the systems and technologies described herein may be implemented in a digital electronic circuit system, an integrated circuit system, an application-specific integrated circuit (ASIC), computer hardware, firmware, software, and/or combinations thereof. The various implementations may include: being implemented in one or more computer programs. The one or more computer programs may be executed and/or interpreted on a programmable system including at least one programmable processor. The programmable processor may be a special-purpose or general-purpose programmable processor, receive data and instructions from a storage system, at least one input apparatus and at least one output apparatus, and transmit the data and the instructions to the storage system, the at least one input apparatus and the at least one output apparatus.

[0065] The computing programs (also referred to as programs, software, software applications, or code) include machine instructions for programmable processors, and may be implemented by using high-level procedural and/or object-oriented programming languages, and/or assembly/machine languages. As used herein, the terms "machine-readable medium" and "computer-readable medium" refer to any computer program product, device, and/or apparatus (e.g., a magnetic disk, an optical disc, a memory, and a programmable logic device (PLD)) configured to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions serving as machine-readable signals. The term "machine-readable signal" refers to any signal for providing the machine instructions and/or data to the programmable processor.

[0066] To provide interaction with a user, the systems and technologies described here can be implemented on a computer. The computer has: a display apparatus (e.g., a cathode-ray tube (CRT) or an LCD monitor) for displaying information to the user; and a keyboard and a pointing apparatus (e.g., a mouse or trackball) through which the user may provide input for the computer. Other kinds of apparatuses may also be configured to provide interaction with the user. For example, a feedback provided for the user may be any form of sensory feedback (e.g., visual, auditory, or tactile feedback); and input from the user may be received in any form (including sound input, voice input, or tactile input).

[0067] The systems and technologies described herein can be implemented in a computing system including background components (e.g., as a data server), or a computing system including middleware components (e.g., an application server), or a computing system including front-end components (e.g., a user computer with a graphical user interface or web browser through which the user can interact with the implementation mode of the systems and technologies described here), or a computing system including any combination of such background components, middleware components or front-end components. The components of the system can be connected to each other through any form or medium of digital data communication (e.g., a communication network). Examples of the communication network include: a local area network (LAN), a wide area network (WAN), and the Internet.

[0068] The computer system may include a client and a server. The client and the server are generally far away from each other and generally interact via the communication network. A relationship between the client and the server is generated through computer programs that run on a corresponding computer and have a client-server relationship with each other.

[0069] With the technical solutions according to the embodiments of the present embodiment, when the deep learning framework receives an error code returned by a third-party library when an error occurs in calling of a third-party library API, the error code can be automatically parsed according to an error file included in the deep learning framework, so as to acquire richer error reporting content including error information, which greatly reduces the development cost of a developer and improves the development efficiency of the developer.

Claims

1. A computer-implemented method for generating error reporting content of a deep learning framework, comprising:

   acquiring (S101) an error code returned by a third-party library when an error occurs in calling of a third-party library application programming interface (API) and error information corresponding to the error code, wherein the error code is used for identifying a type of an error when the deep learning framework fails to call the third-party library API, and the error information is used for describing the cause of the error and the solution for the error;
   determining (S102) a programming language used by the deep learning framework, generating (S102) an error file which corresponds to the programming language, according to the error code and the error information corresponding thereto, and packaging (S102) the error file into the deep learning framework;
   running (S103) the deep learning framework, and in response to the deep learning framework receiving the error code, extracting (S103), from the error file, error information corresponding to the received error code; and
   generating (S104) error reporting content according to the error information,
   recording the generation time of the error file after the generating the error file according to the error code and the error information corresponding thereto; and
   updating the error code and the error information corresponding thereto in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

2. The method according to claim 1, wherein the packaging the error file into the deep learning framework comprises:

   compressing the error file; and
   packaging a compression result of the error file into the deep learning framework.

3. The method according to claim 1, wherein the in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, extracting, from the error file, error information corresponding to the received error code comprises:

   determining the number of retries corresponding to the third-party library API currently called by the deep learning framework; and
   in response to the deep learning framework receiving an error code returned by the third-party library when an error occurs in calling of the third-party library API after the number of retries, extracting, from the error file, error information corresponding to the received error code.

4. The method according to claim 2, wherein the extracting, from the error file, error information corresponding to the received error code comprises:

   determining whether a decompression result of the error file is comprised in the deep learning framework;
   if yes, extracting, from the decompression result, the error information corresponding to the received error code; and
   otherwise, decompressing the error file, and then extracting, from a decompression result, the error information corresponding to the received error code.

5. The method according to claim 1, wherein the generating error reporting content according to the error information comprises:
splicing preset particular information with the error information, and taking a splicing result as the error reporting content.

6. An apparatus for generating error reporting content of a deep learning framework, comprising:

an acquisition unit (201) configured to acquire an error code returned by a third-party library when an error occurs in calling of a third-party library application programming interface (API) and error information corresponding to the error code, wherein the error code is used for identifying a type of an error when the deep learning framework fails to call the third-party library API, and the error information is used for describing the cause of the error and the solution for the error;
a processing unit (202) configured to determine a programming language used by the deep learning framework, generate an error file which corresponds to a programming language, according to the error code and the error information corresponding thereto, and package the error file into the deep learning framework;
an error reporting unit (203) configured to run the deep learning framework, and in response to the deep learning framework receiving the error code, extract, from the error file, error information corresponding to the received error code; and
a generation unit (204) configured to generate error reporting content according to the error information, record the generation time of the error file after the generating the error file according to the error code and the error information corresponding thereto, and update the error code and the error information corresponding thereto in the error file in response to a time difference between the current time and the previous generation time reaching a preset time interval.

7. The apparatus according to claim 6, wherein when packaging the error file into the deep learning framework, the processing unit (202) is specifically configured to:

compress the error file; and
package a compression result of the error file into the deep learning framework.

8. The apparatus according to claim 6, wherein when in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library API, extracting, from the error file, error information corresponding to the received error code, the error reporting unit (203) is specifically configured to:

determine the number of retries corresponding to the third-party library API currently called by the deep learning framework; and
in response to the deep learning framework receiving an error code returned by the third-party library when an error occurs in calling of the third-party library API after the number of retries, extract, from the error file, error information corresponding to the received error code.

9. A non-transitory computer-readable storage medium storing computer instructions, wherein the computer instructions are used to make the computer to perform the method according to any one of claims 1 to 5.

10. A computer program product comprising instructions which, when the program is executed by a computer, cause the computer to perform the method according to any one of claims 1-5.

Patentansprüche

1. Computerimplementiertes Verfahren zum Erzeugen eines Fehlerberichtsinhalts eines Deep-Learning-Frameworks, mit den folgenden Schritten:

Erfassen (S101) eines Fehlercodes, der von einer Drittpartei-Bibliothek zurückgesendet wird, wenn ein Fehler beim Aufruf einer Drittpartei-Bibliothek-Anwendungsprogrammierschnittstelle (API) auftritt, und von dem Fehlercode entsprechenden Fehlerinformationen, wobei der Fehlercode zur Identifizierung eines Typs eines Fehlers verwendet wird, wenn das Aufrufen der einer Drittpartei-Bibliothek-API durch das Deep-Learning-Framework fehlschlägt, und die Fehlerinformationen zur Beschreibung der Ursache des Fehlers und der Lösung für den Fehler verwendet werden;
Bestimmen (S102) einer von dem Deep-Learning-Framework verwendeten Programmiersprache, Erzeugen (S102) einer der Programmiersprache entsprechenden Fehlerdatei gemäß dem Fehlercode und den diesem entsprechenden Fehlerinformationen, und Packen (S102) der Fehlerdatei in das Deep-Learning-Framework;
Ausführen (S103) des Deep-Learning-Frameworks und, in Reaktion auf das Empfangen des Fehlercodes durch das Deep-Learning-Framework, Extrahieren (S103) von Fehlerinformationen aus der Fehlerdatei, welche dem empfan-

genen Fehlercode entsprechen; und
Erzeugen (S104) eines Fehlerberichtsinhalts gemäß den Fehlerinformationen;
Aufzeichnen des Erzeugungszeitpunkts der Fehlerdatei nach dem Erzeugen der Fehlerdatei gemäß dem Fehlercode und den diesem entsprechenden Fehlerinformationen; und
Aktualisieren des Fehlercodes und der diesem entsprechenden Fehlerinformationen in der Fehlerdatei in Reaktion auf das Erreichen eines vorgegebenen Zeitintervalls durch einen Zeitunterschied zwischen dem aktuelle Zeitpunkt und dem vorhergehenden Erzeugungszeitpunkt.

2. Verfahren nach Anspruch 1, bei welchem das Packen der Fehlerdatei in das Deep-Learning-Framework die folgenden Schritte aufweist:

Komprimieren der Fehlerdate; und
Packen eines Kompressionsergebnisses der Fehlerdatei in das Deep-Learning-Framework.

3. Verfahren nach Anspruch 1, bei welchem das in Reaktion auf das Empfangen des Fehlercodes durch das Deep-Learning-Framework von einer Drittpartei-Bibliothek-API erfolgende Extrahieren (S103) von Fehlerinformationen aus der Fehlerdatei, welche dem empfangenen Fehlercode entsprechen, die folgenden Schritte aufweist:

Bestimmen der Anzahl der erneuten Versuche entsprechend der aktuell von dem Deep-Learning-Framework aufgerufenen Drittpartei-Bibliothek-API; und
in Reaktion auf das Empfangen eines Fehlercodes durch das Deep-Learning-Framework, der von der Drittpartei-Bibliothek zurückgesendet wird, wenn nach der Anzahl der erneuten Versuche ein Fehler beim Aufruf des Drittpartei-Bibliothek-API auftritt, Extrahieren von Fehlerinformationen aus der Fehlerdatei, welche dem empfangenen Fehlercode entsprechen.

4. Verfahren nach Anspruch 2, bei welchem das Extrahieren von Fehlerinformationen aus der Fehlerdatei, welche dem empfangenen Fehlercode entsprechen, die folgenden Schritte aufweist:

Feststellen, ob ein Dekompressionsergebnis der Fehlerdatei in dem Deep-Learning-Framework enthalten ist;
wenn ja, Extrahieren der Fehlerinformationen, welche dem empfangenen Fehlercode entsprechen, aus dem Dekompressionsergebnis; und
anderenfalls, Dekomprimieren der Fehlerdatei, und anschließendes Extrahieren der Fehlerinformationen, welche dem empfangenen Fehlercode entsprechen, aus einem Dekompressionsergebnis.

5. Verfahren nach Anspruch 1, bei welchem das Erzeugen des Fehlerberichtsinhalts gemäß den Fehlerinformationen den folgenden Schritt aufweist:
Spleißen vorgegebener bestimmter Informationen mit den Fehlerinformationen, und verwenden eines Spleißungsergebnisses als den Fehlerberichtsinhalt.

6. Vorrichtung zum Erzeugen eines Fehlerberichtinhalts eines Deep-Learning-Frameworks, mit:

einer Erfassungseinheit (201), die dazu ausgebildet ist, einen Fehlercode, der von einer Drittpartei-Bibliothek zurückgesendet wird, wenn ein Fehler beim Aufruf einer Drittpartei-Bibliothek-Anwendungsprogrammierschnittstelle (API) auftritt, und dem Fehlercode entsprechende Fehlerinformationen zu erfassen, wobei der Fehlercode zur Identifizierung eines Typs eines Fehlers verwendet wird, wenn das Aufrufen der einer Drittpartei-Bibliothek-API durch das Deep-Learning-Framework fehlschlägt, und die Fehlerinformationen zur Beschreibung der Ursache des Fehlers und der Lösung für den Fehler verwendet werden;
einer Verarbeitungseinheit (202), die dazu ausgebildet ist, eine von dem Deep-Learning-Framework verwendete Programmiersprache zu bestimmen, eine der Programmiersprache entsprechende Fehlerdatei gemäß dem Fehlercode und den diesem entsprechenden Fehlerinformationen zu erzeugen, und die Fehlerdatei in das Deep-Learning-Framework zu packen;
einer Fehlerberichteinheit (203), die dazu ausgebildet ist, den Deep-Learning-Framework auszuführen und, in Reaktion auf das Empfangen des Fehlercodes durch das Deep-Learning-Framework, Fehlerinformationen, welche dem empfangenen Fehlercode entsprechen, aus der Fehlerdatei zu extrahieren; und
einer Erzeugungseinheit (204), die dazu ausgebildet ist, einen Fehlerberichtsinhalt gemäß den Fehlerinformationen zu erzeugen, den Erzeugungszeitpunkt der Fehlerdatei nach dem Erzeugen der Fehlerdatei gemäß dem Fehlercode und den diesem entsprechenden Fehlerinformationen aufzuzeichnen, und den Fehlercode und die diesem entsprechenden Fehlerinformationen in der Fehlerdatei in Reaktion auf das Erreichen eines vorgegebenen Zeitintervalls durch einen Zeitunterschied zwischen dem aktuelle Zeitpunkt und dem vorhergehenden Erzeugungszeitpunkt zu aktualisieren.

7. Vorrichtung nach Anspruch 6, bei welcher die Verarbeitungseinheit (202) insbesondere dazu ausge-

bildet ist, beim Packen der Fehlerdatei in das Deep-Learning-Framework

die Fehlerdatei zu komprimieren; und
ein Kompressionsergebnis der Fehlerdatei in das Deep-Learning-Framework zu packen.

8. Vorrichtung nach Anspruch 6, bei welcher, wenn in Reaktion auf das Empfangen eines Fehlercodes durch das Deep-Learning-Framework, der von der Drittpartei-Bibliothek zurückgesendet wird, wenn nach der Anzahl der erneuten Versuche ein Fehler beim Aufruf des Drittpartei-Bibliothek-API auftritt, die Fehlerberichteinheit (203) beim Extrahieren von Fehlerinformationen aus der Fehlerdatei, welche dem empfangenen Fehlercode entsprechen, insbesondere dazu ausgebildet ist:

die Anzahl der erneuten Versuche entsprechend der aktuell von dem Deep-Learning-Framework aufgerufenen Drittpartei-Bibliothek-API zu bestimmen; und
in Reaktion auf das Empfangen eines Fehlercodes durch das Deep-Learning-Framework, der von der Drittpartei-Bibliothek zurückgesendet wird, wenn nach der Anzahl der erneuten Versuche ein Fehler beim Aufruf des Drittpartei-Bibliothek-API auftritt, Fehlerinformationen, welche dem empfangenen Fehlercode entsprechen, aus der Fehlerdatei zu extrahieren.

9. Nicht-flüchtiges computerlesbares Speichermedium, das Computerbefehle speichert, wobei die Computerbefehle dazu verwendet werden, den Computer zu veranlassen, das Verfahren nach einem der Ansprüche 1 bis 5 durchzuführen.

10. Computerprogrammprodukt, das Befehle aufweist, welche, wenn das Programm von einem Computer ausgeführt wird, den Computer veranlassen, das Verfahren nach einem der Ansprüche 1 bis 5 durchzuführen.

Revendications

1. Procédé mis en oeuvre par ordinateur permettant de générer un contenu de rapport d'erreur d'un cadre d'apprentissage profond, comprenant :

l'acquisition (S101) d'un code d'erreur renvoyé par une bibliothèque tierce lorsqu'une erreur se produit lors de l'appel d'une interface de programmation d'application (API) de bibliothèque tierce et des informations d'erreur correspondant au code d'erreur, dans lequel le code d'erreur est utilisé pour identifier un type d'erreur lorsque le cadre d'apprentissage profond ne parvient pas à appeler l'API de bibliothèque tierce et que les informations d'erreur sont utilisées pour décrire la cause de l'erreur et la solution pour l'erreur ;
la détermination (S102) d'un langage de programmation utilisé par le cadre d'apprentissage profond, la génération (S102) d'un fichier d'erreur qui correspond au langage de programmation, selon le code d'erreur et les informations d'erreur correspondant à celui-ci, et l'intégration (S102) du fichier d'erreur dans le cadre d'apprentissage profond ;
l'exécution (S103) du cadre d'apprentissage profond, et en réponse à la réception par le cadre d'apprentissage profond d'un code d'erreur, l'extraction (S103), à partir du fichier d'erreur, d'informations d'erreur correspondant au code d'erreur reçu ; et
la génération (S104) d'un contenu de rapport d'erreur selon les informations d'erreur,
l'enregistrement de l'heure de génération du fichier d'erreur après la génération du fichier d'erreur selon le code d'erreur et les informations d'erreur correspondant à celui-ci ; et
la mise à jour du code d'erreur et des informations d'erreur correspondant à celui-ci dans le fichier d'erreur en réponse à une différence de temps entre l'heure actuelle et l'heure de génération précédente atteignant un intervalle de temps prédéfini.

2. Procédé selon la revendication 1, dans lequel l'intégration du fichier d'erreur dans le cadre d'apprentissage profond comprend :

la compression du fichier d'erreur ; et
l'intégration d'un résultat de compression du fichier d'erreur dans le cadre d'apprentissage profond.

3. Procédé selon la revendication 1, dans lequel en réponse à la réception par le cadre d'apprentissage profond d'un code d'erreur renvoyé par une bibliothèque tierce lorsqu'une erreur se produit lors de l'appel d'une API de bibliothèque tierce, l'extraction, à partir du fichier d'erreur, d'informations d'erreur correspondant au code d'erreur reçu comprend :

la détermination du nombre de nouvelles tentatives correspondant à l'API de bibliothèque tierce actuellement appelée par le cadre d'apprentissage profond ; et
en réponse à la réception par le cadre d'apprentissage profond d'un code d'erreur renvoyé par la bibliothèque tierce lorsqu'une erreur se produit lors de l'appel de l'API de bibliothèque tierce après le nombre de nouvelles tentatives, l'extraction, à partir du fichier d'erreur, d'informa-

tions d'erreur correspondant au code d'erreur reçu.

4. Procédé selon la revendication 2, dans lequel l'extraction, à partir du fichier d'erreur, d'informations d'erreur correspondant au code d'erreur reçu comprend :

la détermination du fait qu'un résultat de décompression du fichier d'erreur est compris, ou non, dans le cadre d'apprentissage profond ;
si oui, l'extraction, à partir du résultat de décompression, des informations d'erreur correspondant au code d'erreur reçu ; et
sinon, la décompression du fichier d'erreur, puis l'extraction, à partir d'un résultat de décompression, des informations d'erreur correspondant au code d'erreur reçu.

5. Procédé selon la revendication 1, dans lequel la génération du contenu de rapport d'erreur selon les informations d'erreur comprend :
l'épissage d'informations particulières prédéfinies avec les informations d'erreur, et la prise d'un résultat d'épissage comme contenu de rapport d'erreur.

6. Appareil pour générer un contenu de rapport d'erreur d'un cadre d'apprentissage profond, comprenant :

une unité d'acquisition (201) configurée pour acquérir un code d'erreur renvoyé par une bibliothèque tierce lorsqu'une erreur se produit lors de l'appel d'une interface de programmation d'application (API) de bibliothèque tierce et des informations d'erreur correspondant au code d'erreur, dans lequel le code d'erreur est utilisé pour identifier un type d'erreur lorsque le cadre d'apprentissage profond ne parvient pas à appeler l'API de bibliothèque tierce, et les informations d'erreur sont utilisées pour décrire la cause de l'erreur et la solution pour l'erreur ;
une unité de traitement (202) configurée pour déterminer un langage de programmation utilisé par le cadre d'apprentissage profond, générer un fichier d'erreur qui correspond à un langage de programmation, selon le code d'erreur et les informations d'erreur correspondant à celui-ci, et introduire le fichier d'erreur dans le cadre d'apprentissage profond ;
une unité de rapport d'erreur (203) configurée pour exécuter le cadre d'apprentissage profond, et en réponse à la réception par le cadre d'apprentissage profond du code d'erreur, l'extraction, à partir du fichier d'erreur, d'informations d'erreur correspondant au code d'erreur reçu ; et
une unité de génération (204) configurée pour générer un contenu de rapport d'erreur selon les informations d'erreur, enregistrer l'heure de génération du fichier d'erreur après la génération du fichier d'erreur selon le code d'erreur et les informations d'erreur correspondant à celui-ci, et mettre à jour le code d'erreur et les informations d'erreur correspondant à celles-ci dans le fichier d'erreur en réponse à une différence de temps entre l'heure actuelle et l'heure de génération précédente atteignant un intervalle de temps prédéfini.

7. Appareil selon la revendication 6, dans lequel lors de l'intégration du fichier d'erreur dans le cadre d'apprentissage profond, l'unité de traitement (202) est spécifiquement configurée pour :

compresser le fichier d'erreur ; et
introduire un résultat de compression du fichier d'erreur dans le cadre d'apprentissage profond.

8. Appareil selon la revendication 6, dans lequel lorsqu'en réponse à la réception par le cadre d'apprentissage profond d'un code d'erreur renvoyé par une bibliothèque tierce lorsqu'une erreur se produit lors de l'appel d'une API de bibliothèque tierce, l'extraction, à partir du fichier d'erreur, d'informations d'erreur correspondant au code d'erreur reçu, l'unité de rapport d'erreur (203) est spécifiquement configurée pour :

déterminer le nombre de nouvelles tentatives correspondant à l'API de bibliothèque tierce actuellement appelée par le cadre d'apprentissage profond ; et
en réponse à la réception par le cadre d'apprentissage profond d'un code d'erreur renvoyé par la bibliothèque tierce lorsqu'une erreur se produit lors de l'appel de l'API de bibliothèque tierce après le nombre de nouvelles tentatives, extraire, à partir du fichier d'erreur, les informations d'erreur correspondant au code d'erreur reçu.

9. Support de stockage non transitoire lisible par ordinateur stockant des instructions informatiques, dans lequel les instructions informatiques sont utilisées pour amener l'ordinateur à effectuer le procédé selon l'une quelconque des revendications 1 à 5.

10. Produit programme informatique comprenant des instructions qui, lorsque le programme est exécuté par un ordinateur, amènent l'ordinateur à effectuer le procédé selon l'une quelconque des revendications 1 à 5.

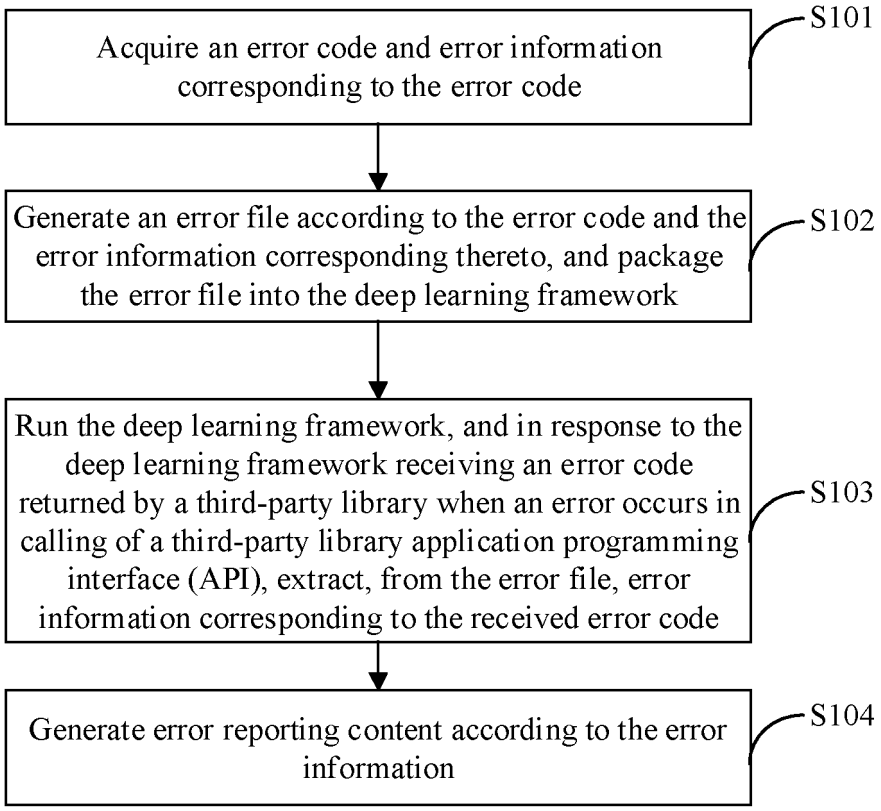
Acquire an error code and error information corresponding to the error code
S101
Generate an error file according to the error code and the error information corresponding thereto, and package the error file into the deep learning framework
S102
Run the deep learning framework, and in response to the deep learning framework receiving an error code returned by a third-party library when an error occurs in calling of a third-party library application programming interface (API), extract, from the error file, error information corresponding to the received error code
S103
Generate error reporting content according to the error information
S104

Fig.1

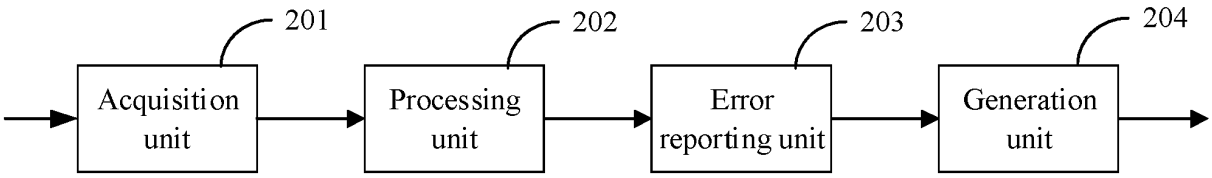
201
202
203
204
Acquisition unit
Processing unit
Error reporting unit
Generation unit

Fig.2

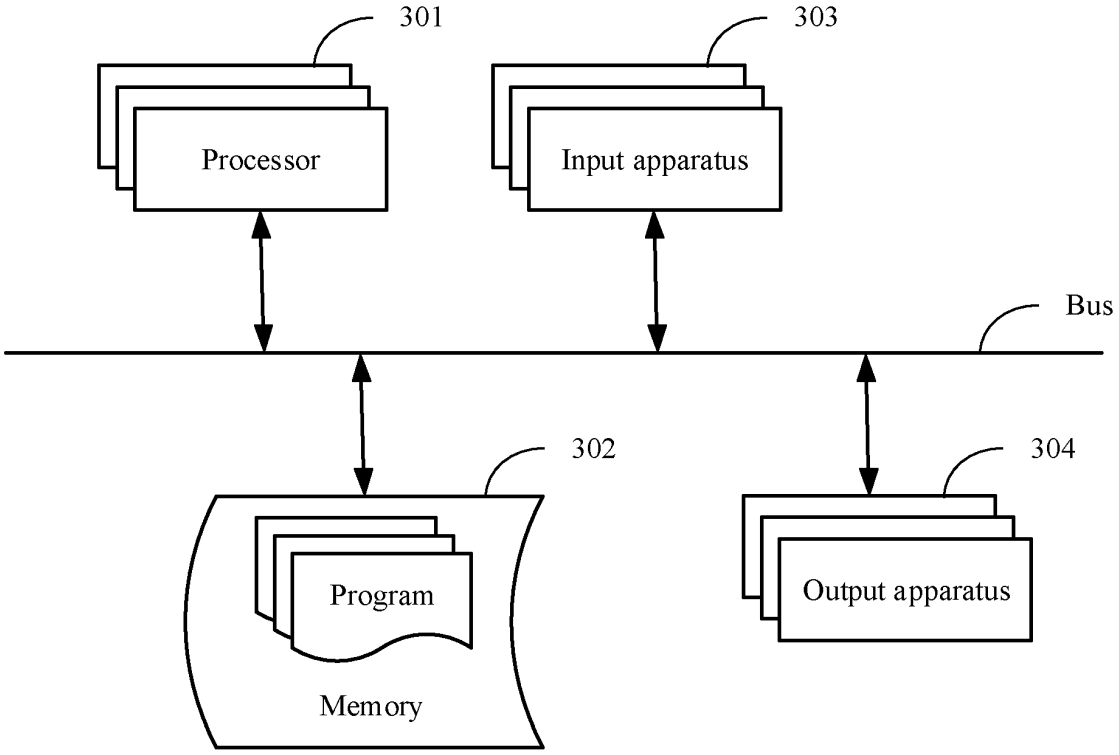
301
303
Processor
Input apparatus
Bus
302
304
Program
Memory
Output apparatus

Fig.3

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 0735471 A2 **[0004]**
- CN 105550059 A **[0005]**