DATA PROCESSING METHOD, TEXT PROCESSING METHOD, IMAGE PROCESSING METHOD, COMPUTING DEVICE, COMPUTER STORAGE MEDIUM AND COMPUTER PROGRAM PRODUCT

The present disclosure claims priority to Chinese Patent Application No. 202410217614.9, filed with the China National Intellectual Property Administration on February 27, 2024 and entitled "DATA PROCESSING METHOD, TEXT PROCESSING METHOD, IMAGE PROCESSING METHOD, COMPUTING DEVICE, COMPUTER STORAGE MEDIUM AND COMPUTER PROGRAM PRODUCT", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

Embodiments of this specification relate to the field of computer technologies, and in particular, to a data processing method, a text processing method, an image processing method, a computing device, a computer storage medium and a computer program product.

BACKGROUND

For a conventional deep learning inference acceleration solution, a static graph optimization technology may be used to greatly enhance model inference efficiency, that is, nodes in a computational graph are fused, and a highly optimized kernel function is selected to perform an operation, thereby reducing computing overheads of computer hardware resources and increasing an execution speed of a model inference process.

However, in view of models having an iterative diffusion process and a dynamic behavior, such as Stable-Diffusion (which is an image generation model, an operating principle of which involves a plurality of iteration processes for gradual conversion from a noisy image to a clear image. In each iteration, parameters or computing procedures need to be dynamically adjusted based on a result of a previous iteration, and it is hard to represent and optimize the model directly by using a static graph due to such a dynamic and recursive property), this static graph solution cannot flexibly deal with the feature that a model dynamically determines a subsequent step according to a previous step in an inference process, and therefore, it is difficult to directly apply this solution and achieve the same acceleration effect. This means that for this type of model, a new optimization technology for adapting to an internal dynamic attribute of the

model needs to be developed.

SUMMARY

In view of this, embodiments of this specification provide a data processing method, a text processing method, an image processing method, a computing device, a computer storage medium and a computer program product, to solve technical defects in the existing technology.

According to a first aspect of the embodiments of this specification, a data processing method is provided, including:

determining a target computational graph of a target model and target data inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target data, where the current node is any one of the plurality of nodes; and

determining a data processing computational graph corresponding to the current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

According to a second aspect of the embodiments of this specification, a data processing apparatus is provided, including:

a data determination module, configured to determine a target computational graph of a target model and target data inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module, configured to determine, from the target computational graph, a current node for processing the target data, where the current node is any one of the plurality of nodes; and

a result obtaining module, configured to: determine a data processing computational graph corresponding to the current node, and process the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

According to a third aspect of the embodiments of this specification, a text processing method is provided, including:

determining a target computational graph of a target model and target text inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target text, where the current node is any one of the plurality of nodes; and

determining a text processing computational graph corresponding to the current node, and processing the target text based on the current node and/or the text processing computational graph, to obtain a text processing result.

According to a fourth aspect of the embodiments of this specification, a text processing apparatus is provided, including:

a text determination module, configured to determine a target computational graph of a target model and target text inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module, configured to determine, from the target computational graph, a current node for processing the target text, where the current node is any one of the plurality of nodes; and

a result obtaining module, configured to: determine a text processing computational graph corresponding to the current node, and process the target text based on the current node and/or the text processing computational graph, to obtain a text processing result.

According to a fifth aspect of the embodiments of this specification, an image processing method is provided, including:

determining a target computational graph of a target model and a target image inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target image, where the current node is any one of the plurality of nodes; and

determining an image processing computational graph corresponding to the current node, and processing the target image based on the current node and/or the image processing

computational graph, to obtain an image processing result.

According to a sixth aspect of the embodiments of this specification, an image processing apparatus is provided, including:

an image determination module, configured to determine a target computational graph of a target model and a target image inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module, configured to determine, from the target computational graph, a current node for processing the target image, where the current node is any one of the plurality of nodes; and

a result obtaining module, configured to: determine an image processing computational graph corresponding to the current node, and process the target image based on the current node and/or the image processing computational graph, to obtain an image processing result.

According to a seventh aspect of the embodiments of this specification, a computing device is provided, including:

a memory and a processor;

where the memory is configured to store a computer program/instruction, and the processor is configured to execute the computer program/instruction, where the computer program/instruction, when executed by the processor, implement the steps of the foregoing data processing method, or text processing method, or image processing method.

According to an eighth aspect of the embodiments of this specification, a computer-readable storage medium is provided, having a computer program/instruction stored therein, where the computer program/instruction, when executed by a processor, implement the steps of the foregoing data processing method, or text processing method, or image processing method.

According to a ninth aspect of the embodiments of this specification, a computer program product is provided, including a computer program/instruction, where the computer program/instruction, when executed by a processor, implement the steps of the foregoing data processing method, or text processing method, or image processing method.

One embodiment of this specification provides a data processing method, including: determining a target computational graph of a target model and target data inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target

computational graph is a computational graph that is unable to be updated when the target model is run; determining, from the target computational graph, a current node for processing the target data, where the current node is any one of the plurality of nodes; and determining a data processing computational graph corresponding to the current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

In the data processing method provided in the embodiments of this specification, the target computational graph of the target model is determined. In a case that the target data is inputted to the target model, since the target computational graph is a computational graph that is unable to be updated when the target model is run, the target computational graph is completely defined and compiled before the target model is run. This helps optimize the compilation of the target computational graph before model inference is performed, thereby reducing an operational load of a processor and enhancing execution efficiency of the processor. Meanwhile, by flexibly invoking the data processing computational graph, it can be ensured that the target model performs an efficient and accurate pre-processing or after-processing operation on the target data, so that dynamic insertion of a computing procedure into the target computational graph is implemented, and secondary compilation is not required, thereby reducing overheads of computational graph compilation, lowering a requirement for and consumption of computer hardware resources, and enhancing model inference efficiency.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram of a scenario of a data processing method according to an embodiment of this specification.

FIG. 2 is a flowchart of a data processing method according to an embodiment of this specification.

FIG. 3 is a schematic diagram of a pre-process table according to an embodiment of this specification.

FIG. 4 is a schematic diagram of a UNET model structure according to an embodiment of this specification.

FIG. 5 is a schematic diagram of a target mapping table according to an embodiment of this specification.

FIG. 6 is a flowchart of a processing process of a single node of a data processing method according to an embodiment of this specification.

FIG. 7 is a flowchart of a text processing method according to an embodiment of this specification.

FIG. 8 is a flowchart of an image processing method according to an embodiment of this specification.

FIG. 9 is a schematic diagram of a structure of a data processing apparatus according to an embodiment of this specification.

FIG. 10 is a schematic diagram of a structure of a text processing apparatus according to an embodiment of this specification.

FIG. 11 is a schematic diagram of a structure of an image processing apparatus according to an embodiment of this specification.

FIG. 12 is a block diagram of a structure of a computing device according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

In the following description, many specific details are described for thorough understanding of this specification. This specification may, however, be embodied in many forms different from that described here. A person skilled in the art can make similar improvements without departing from the connotation of this specification. Therefore, this specification is not limited by the specific embodiments disclosed below.

The terms used in one or more embodiments of this specification are merely intended to describe the specific embodiments but are not intended to limit one or more embodiments of this specification. The terms "a", "said", and "the" in a singular form used in one or more embodiments of this specification and the appended claims are also intended to include a plural form, unless other meanings are clearly indicated in the context. It should be further understood that the term "and/or" used in one or more embodiments of this specification means and includes any or all possible combinations of one or more associated listed items.

It should be understood that although terms "first", "second", and the like "may be used to describe various information in one or more embodiments of this specification, such information should not be limited to these terms. These terms are merely used to differentiate the same type of information. For example, without departing from the scope of the one or more embodiments of this specification, "first" may also be referred to as "second", and similarly, "second" may also be referred to as "first". Depending on the context, the term "if" used herein can be interpreted as "when" or "in response to determining that".

In addition, it should be noted that, user information (including but not limited to user device information and user personal information) and data (including but not limited to data for analysis, stored data, displayed data, and the like) involved in one or more embodiments of this specification are information and data which are authorized by the user or fully authorized by all parties. Furthermore, acquisition, use, and processing of relevant data all comply with relevant laws, regulations, and standards in relevant districts, and corresponding operation entries are provided for the user for authorization or refuse selection.

In one or more embodiments of this specification, a large model refers to a deep learning model having large-scale model parameters, which usually include hundreds of millions, tens of billions, hundreds of billions, trillions, or even tens of trillions of model parameters. The large model may alternatively be referred to as a foundation model (Foundation Model). The large model is pre-trained by using a large-scale unlabeled corpus, to obtain a pre-training model with more than a hundred million parameters. Such a model can adapt to a wide range of downstream tasks, has a relatively good generalization capability, and includes, for example, a large language model (Large Language Model, LLM) or a multi-modal pre-training model (multi-modal pre-training model).

During actual application, the large model may be applied to different tasks by using only a small number of samples to finely tune the pretrained model. The large model may be widely applied to fields such as natural language processing (Natural Language Processing, NLP for short) and computer vision, and specifically may be applied to computer vision field tasks such as visual question answering (Visual Question Answering, VQA for short), image caption (Image Caption, IC for short), and image generation, and natural language processing field tasks such as text-based sentiment classification, text abstract generation, and machine translation. Main application scenarios of the large model include a digital assistant, an intelligent robot, search, online education, office software, e-commerce, intelligent design, and the like.

First, nouns and terms involved in one or more embodiments of this specification are explained.

Stable-Diffusion is a text-to-image generation model. It is mainly used for generating a detailed image based on a text description, and may alternatively be applied to other tasks, such as internal supplementary drawing, external supplementary drawing, and generation of image conversion under guidance of prompt words.

Controlnet is an important matching plug-in in a Stable Diffusion model. Its function is to guide an image generation process by controlling a feature expression, to ensure that a

generated image is more consistent with an inputted text description.

TensorRT is a high-performance deep learning inference SDK (Software Development Kit, software development kit), including a deep learning inference optimizer and a running environment, and providing a low latency and high throughput for deep learning inference applications.

For a modern large model, such as a LLM (Large Language Model, large language model) and a stable-diffusion visual generation model, a series of light-weight control technologies are used in an inference process to precisely adjust an output result. For example, Controlnet (an important matching plug-in in the Stable Diffusion model) designs a parallel structure, and integrates, by using an independent network module for specially processing reference image information, a computation result tensor generated by the independent network module into computation of the Stable Diffusion model, to control and affect an image generated by the Stable Diffusion model. In another method, a specific upsampling-downsampling modules is inserted into the model, to perform style transfer and dynamic style control on a generated image without changing an overall model architecture.

Based on a dynamically adjustable characteristic of a dynamic graph, by replacing a particular operator in a computational graph, one method in the existing technology can adaptively optimize an inference process for different inputs or stages, to implement model inference acceleration. However, this method limits an optimizable manner, and cannot implement operator fusion on a larger scale. Another acceleration method in the existing technology completely uses a static graph for computation. A computational graph cannot be dynamically modified, and a solution such as Controlnet cannot be dynamically supported.

Therefore, an embodiment of this specification provides a data processing method, which remains dynamic modification on a computational graph while optimizing a static graph, so as to implement dynamic modification on the computational graph while saving computer hardware resources.

This specification provides a data processing method. One or more embodiments of this specification also relate to a data processing apparatus, a text processing method, a text processing apparatus, an image processing method, an image processing apparatus, a computing device, a computer-readable storage medium, and a computer program product. Detailed descriptions are made one by one in the following embodiments.

Referring to FIG. 1, FIG. 1 is a schematic diagram of a scenario of a data processing method according to an embodiment of this specification.

Specifically, the data processing method is implemented by using a client 102 and a server

104. The client 102 is configured to transmit target data to the server 104. A target model is deployed in the server 104, and the target model is converted into a static computational graph. Nodes in the static computational graph represent mathematical operations (that is, processing operations of data processing performed on the target model), and edges represent data flows (that is, association relationships between the processing operations). In addition, the static computational graph of the target model may be compiled and optimized. For example, processing efficiency is enhanced by combining repeated operations and reducing redundant computations, and computation can be accelerated by using hardware such as a graphics processing unit (Graphics Processing Unit, GPU). A compiled static computational graph can be efficiently executed.

In a case that target data is inputted to the target model, an inference procedure of the target model is performed, and nodes are computed one by one based on the static computational graph. That is, a current node for processing the target data is determined from the static computational graph of the target model, and a data processing computational graph corresponding to the current node is determined. The data processing computational graph is used for performing pre-processing and/or after-processing on target processed data. The target data is processed based on the current node and/or the data processing computational graph, to obtain a data processing result, and the data processing result is returned to the client 102.

Specifically, by determining whether the data processing computational graph corresponding to the current node exists, and invoking the data processing computational graph corresponding to the current node in a case that the data processing computational graph exists, during actual application, the data processing computational graph including a data pre-process computational graph and a data after-process computational graph, functions of adding, deleting, and modifying data processing computational graphs before and after the current node are achieved.

The client 102 may include a browser, an APP (Application, application), a web application such as a H5 (Hyper Text Markup Language5, hypertext markup language5) application, a light application (also referred to as an applet which is a lightweight application), a cloud application, or the like. The client may be developed and obtained based on a software development kit (SDK, Software Development Kit) of a corresponding service provided by the server, for example, based on a real time communication (RTC, Real Time Communication) SDK. The client may be deployed in an electronic device, and needs to be run depending on the device, a particular APP in the device, or the like. For example, the electronic device may have a display screen and support information browsing, such as a personal mobile terminal

like a mobile phone, a tablet computer, and a personal computer. Generally, the electronic device may be further configured with various other types of applications, such as a human-computer dialog application, a model training application, a data processing application, a webpage browser application, a shopping application, a search application, an instant messaging tool, an email client, and social platform software.

The server end 104 may be understood as a server that provides various services, and include a physical server and a cloud server, such as a server that provides communication services for a plurality of clients, a server that provides support for models used in the clients and is used for background training, or a server that processes data transmitted by the clients. It should be noted that the server 104 may be implemented as a distributed server cluster including a plurality of servers, or may be implemented as a single server. The server 104 may alternatively be a server for a distributed system, or a server in which blockchain is used. The server 104 may alternatively be a cloud server providing basic cloud computing services such as a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a content delivery network (CDN, Content Delivery Network), and a big data and artificial intelligence platform, or an intelligent cloud computing server or intelligent cloud host with an artificial intelligence technology.

It is worth noting that the data processing method provided in the embodiments of this specification may be executed by the server 104. However, in other embodiments of this specification, the target model may be deployed in the client 102, so that the client 102 may have a function similar to that of the server 104, to perform the data processing method provided in the embodiments of this specification. In other embodiments, the data processing method provided in the embodiments of this specification may be further jointly performed by the client 102 and the server 104.

In the data processing method provided in the embodiments of this specification, the target computational graph of the target model is determined. In a case that the target data is inputted to the target model, since the target computational graph is a computational graph that is unable to be updated when the target model is run, the target computational graph is completely defined and compiled before the target model is run. This helps optimize the compilation of the target computational graph before model inference is performed, thereby reducing an operational load of a processor and enhancing execution efficiency of the processor. Meanwhile, by flexibly invoking the data processing computational graph, it can be ensured that the target model performs an efficient and accurate pre-processing or after-processing operation on the

target data, so that dynamic insertion of a computing procedure into the target computational graph is implemented, and secondary compilation is not required, thereby reducing overheads of computational graph compilation, lowering a requirement for and consumption of computer hardware resources, and enhancing model inference efficiency.

Referring to FIG. 2, FIG. 2 is a flowchart of a data processing method according to an embodiment of this specification, specifically including the following steps.

Specifically, the data processing method is applied to computer hardware that supports machine learning model computation.

Step 202: determine a target computational graph of a target model and target data inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run.

The target model may be understood as any data processing model, such as a text processing model or an image processing model, without limitation. The target computational graph may be understood as a static computational graph, is constructed based on a model structure and an operation flow of the target model, and is an invariable computational graph that is defined and compiled before the target model performs an inference flow.

Specifically, the target computational graph of the target model is determined, and the target data inputted to the target model is determined. The inputted target data can be converted into an input node of the target computational graph, data transmission is performed in the target model through model inference, and the data processing result after the target data is processed is obtained.

During actual application, in a case that the static computational graph of the target model is determined, since the static computational graph allows higher-level optimization, such as internal memory pre-allocation, computational graph optimization, and hardware-specific optimization, than the dynamic computational graph, in a deployment and large-scale production environment, the target model compiled by using the static computational graph can implement higher performance and higher execution efficiency than the dynamic computational graph.

In one or more embodiments of this specification, since operation and construction of the dynamic computational graph are performed simultaneously, that is, values of previous nodes in the dynamic computational graph are calculated first, and then a subsequent dynamic computational graph is constructed based on these obtained values. However, in a case that the

constructed dynamic computational graph needs to be modified, the entire dynamic computational graph needs to be re-constructed, that is, the dynamic computational graph needs to be re-compiled, and a large number of computer hardware resources will be consumed in a compiling process. Therefore, to reduce the consumption of computer hardware resources caused by compilation, the static computational graph of the target model is determined. A specific implementation is as follows.

The determining the target computational graph of the target model and the target data inputted to the target model includes:

determining, based on a hardware resource of the computer hardware that supports the machine learning model computation, the target computational graph of the target model and the target data inputted to the target model.

The hardware resource includes, but is not limited to, a CPU (Central Processing Unit, central processing unit), an internal memory, a storage, a network, and the like.

Specifically, before an actual inference flow is performed by using the target model, a computational graph corresponding to the target model needs to be compiled, that is, a source code is converted into a computer-executable target code. During actual application, in a case of compiling a computational graph, a large number of computer hardware resources need to be consumed for the compilation of both the static computational graph and the dynamic computational graph.

For example, in a compiling process, the CPU is required to complete a large number of computations, such as lexical analysis and grammar analysis. The internal memory is required to store data, including source codes, intermediate codes, target codes, and the like. Storage resources need to be used to read and write a large number of files, such as source code files and header files. In addition, in a case that a dependency item or a library needs to be downloaded from a network, network resources need to be used.

Because of a characteristic that a dynamic computational graph is constructed while being computed, in a case that a constructed dynamic computational graph needs to be modified, the dynamic computational graph may be reconstructed. In this case, secondary compilation needs to be performed, that is, a newly constructed dynamic computational graph is compiled again. This causes a problem of consuming a large quantity of computer hardware resources.

The data processing method provided in the embodiments of this specification is applied to the computer hardware that supports the machine learning model computation, and determines the target computational graph of the target model based on the hardware resource of the computer hardware that supports machine learning model computation. That is, in a case

that the hardware resource of the computer hardware is limited and insufficiently configured, the target model can be converted into a static computational graph. In a case that the static computational graph needs to be modified, the static computational graph is modified by dynamically invoking the data processing computational graph. Therefore, there is no secondary compilation situation, that is, the static computational graph does not need to be recompiled. This can reduce overheads of compiling a large number of computational graphs and reduce a requirement for and consumption of computer hardware resources.

In one or more embodiments of this specification, to accurately construct the target computational graph of the target model, in a case that the target model is obtained, the nodes and the edges in the target computational graph are determined based on the target model. A specific implementation is as follows.

The determining the target computational graph of the target model includes:

obtaining the target model, using processing operations of data processing performed on the target model as the nodes, and using association relationships between the processing operations as the edges; and

constructing the target computational graph based on the nodes and the edges.

A node may be understood as a specific mathematical operation or data processing operation. An edge is used for representing a dependency relationship between a direction of a data flow and a data processing operation.

Specifically, a model structure of the target model is first designed and implemented based on a requirement of a machine learning task, to obtain the target model, or a designed target model may be obtained, and processing operations of the target model during data processing are used as the nodes for constructing the target computational graph. For example, in a neural network, a node may represent a matrix multiplication step, an addition step, an activation function application step, a loss function computation step, an optimizer update step, or the like. Furthermore, the association relationships between the processing operations are used as the edges for constructing the target computational graph. For example, in an inference process of the target model, an output obtained after input data is processed by a first convolutional layer is used as an input of a next convolutional layer or pooling layer, and association between the previous and next layers may be expressed by an edge. Thus, the target computational graph is constructed based on the nodes and the edges.

In the data processing method provided in the embodiments of this specification, by constructing the target computational graph, computing steps that can be performed in parallel in the target computational graph can be clearly represented. This helps efficiently perform a

model inference flow on a large-scale dataset. In a distributed environment, a data processing task may further be decomposed to different computing resources based on the structure of the target computational graph, thereby enhancing performance and stability of a computer system.

In one or more embodiments of this specification, an initial computational graph is first constructed based on the nodes and the edges. In a case that the initial computational graph is also a static computational graph, the nodes in the initial computational graph are compiled to implement graph optimization, thus obtaining the target computational graph. A specific implementation is as follows.

The constructing the target computational graph based on the nodes and the edges includes:

constructing an initial computational graph based on the nodes and the edges; and

performing a redundancy elimination operation on a target node and a target edge in the initial computational graph, to obtain the target computational graph of the target model, to run the target model in a target processor in a state of low utilization.

The target node may be understood as a node that needs to be optimized or adjusted, for example, a repeated computing node or a node that can be fused. The target edge may be understood as an edge associated with the target node. For example, in a case that the target node is deleted, the target edge associated with the target node also needs to be deleted. The target processor may be understood as a processor in which the target model is run, and is configured to provide a computing resource in the model inference flow of the target model, for example, a central processing unit or a graphics processing unit, without limitation.

Specifically, in a case that the initial computational graph is constructed based on the nodes and the edges, since the initial computational graph is the static computational graph, the nodes in the initial computational graph may be compiled, to implement graph optimization on the initial computational graph, including but not limited to the redundancy elimination operation, such as constant folding, operator fusion, and dead code deletion, thereby obtaining a simpler and more efficient target computational graph.

In a case of obtaining the target computational graph by eliminating unnecessary repeated computations or combining consecutive operations, computing steps and an internal memory access count in the inference process of the target model can be reduced. In addition, after the target node is deleted, an additional internal memory does not need to be allocated for these invalid or repeated computations. This helps reduce memory occupation and implements that the target processor can run the target model in the state of low utilization.

In the data processing method provided in the embodiments of this specification, the redundancy elimination operation is performed on the nodes in the initial computational graph,

so that execution performance of the target model can be enhanced. The simpler and more efficient target computational graph also helps deployment and migration of the target model on different hardware platforms.

Step 204: determine, from the target computational graph, a current node for processing the target data, where the current node is any one of the plurality of nodes.

The current node may be understood as a specific node at which the target model performs an operation based on a structural sequence of the target computational graph.

Specifically, when the target model performs the inference process, an engine performs computation on the nodes one by one based on the target computational graph. In a deep learning framework, the engine may be understood as a kernel part that processes the target computational graph. It can parse the structure of the target computational graph, schedule a node execution sequence, and perform parallel computation and internal memory management by using underlying computer hardware.

During actual application, after the target data is inputted to the target model, the target data may be first transmitted to a 1st node in the target computational graph for processing. This node may be a node corresponding to an input layer of the target model, and is responsible for performing preliminary conversion on the target data. As the computing flow advances, each node receives, based on a processing operation and an associated edge that are defined by the node, an output of a previous node as an input, and generates a new output.

Step 206: determine a data processing computational graph corresponding to the current node, and process the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

The data processing computational graph may be understood as a static computational graph for data processing except the target computational graph.

Specifically, the data processing computational graph corresponding to the current node may be determined, and the target data is processed by using the current node and/or the data processing computational graph based on a requirement for whether to perform pre-processing and/or after-processing on the target processed data, to obtain the data processing result.

During actual application, in a case that the data processing computational graph is also a static computational graph, a varying dynamic computing logic can be implemented by using a computational graph framework implemented by combining the target computational graph with the data processing computational graph, and the data processing computational graph can be adaptively invoked at any time according to an actual requirement. It is greatly flexible.

In one or more embodiments of this specification, a data pre-process computational graph

corresponding to a target structural unit is determined by determining whether the target structural unit of the target model corresponding to the current node is in a pre-process table. A specific implementation is as follows.

The determining the data processing computational graph corresponding to the current node includes:

determining a target structural unit of the target model corresponding to the current node based on a target mapping table; and

determining a data pre-process computational graph corresponding to the target structural unit from a pre-process table in a case that the target structural unit is in the pre-process table, where the pre-process table includes a mapping relationship between the target structural unit and the data pre-process computational graph, and the data pre-process computational graph is used for performing pre-processing on target processed data.

The target mapping table may be understood as a data structure for describing a correspondence relationship between a node in the target computational graph and a structural unit of the target model. The structural unit of the target model may be understood as a basic composition element for constructing the target model. For example, in a neural network, the structural unit may be understood as a network layer, and includes, but is not limited to, an input layer, a hidden layer (a fully-connected layer, a convolutional layer, or a recurrent layer), an output layer, and the like. The target structural unit may be understood as a structural unit that is determined based on the target mapping table and corresponds to the current node.

The data pre-process computational graph may be understood as a computational graph in which target data is processed before the current node is executed.

Specifically, the target structural unit of the target model corresponding to the current node is first determined based on the target mapping table. Whether the target structural unit is in the pre-process table is searched. If the target structural unit is in the pre-process table, the data pre-process computational graph corresponding to the target structural unit is determined from the pre-process table. If the target structural unit is not in the pre-process table, target data is processed based on the current node.

As shown in FIG. 3, FIG. 3 is a schematic diagram of a pre-process table according to an embodiment of this specification.

For example, in Controlnet, input data is generally added with a constant value outside the target computational graph, to process the input data. In this case, based on the target mapping table, it is determined that the target structural unit of the target model corresponding to the current node is a structural unit 1. By using the pre-process table shown in FIG. 3, the

structural unit 1 points to an external addition computational graph (that is, the data pre-process computational graph), where input [0] = getInputs () (input [0] = getInputs ()) indicates obtaining the input data; input [1] = Constant () (input [1] = Constant ()) represents setting a constant value; output [0] = input [0] + input [1] (output [0] = input [0] + input [1]) represents adding the input data and the constant value to obtain an output result; and return output [0] (return output [0]) indicates returning a computed output result. Therefore, by the external addition computational graph, the input data is added with external data before the current node, to complete insertion of the external addition computational graph.

In the data processing method provided in the embodiments of this specification, in a case that the target structural unit determined based on the target mapping table is in the pre-process table, the data pre-process computational graph corresponding to the target structural unit is determined from the pre-process table, so as to process the target data by using the data pre-process computational graph, thereby implementing dynamic insertion of the computational graph before the current node.

In one or more embodiments of this specification, the target mapping table is constructed based on mapping relationships between the nodes in the target computational graph and the structural units of the target model. A specific implementation is as follows.

Before the determining the target structural unit of the target model corresponding to the current node based on the target mapping table, the method further includes:

determining structural units for constructing the target model, where the structural units are network layers of the target model;

determining structural units corresponding to the nodes in the target computational graph; and

constructing the target mapping table based on mapping relationships between the nodes in the target computational graph and the structural units of the target model.

Specifically, a UNET model including a plurality of structural units is used as an example. UNET (U-Net) is a convolutional neural network (Convolutional Neural Network, CNN) architecture, and a name of U-Net derives from its unique U-shaped structure. As shown in FIG. 4, FIG. 4 is a schematic diagram of a UNET model structure according to an embodiment of this specification. A structure of the UNET model mainly includes two parts: an encoder block (Encoder block) and a decoder block (Decoder block).

The encoder block generally includes a multilayer structure, and each layer further includes a self-attention mechanism (Self-Attention), a feedforward neural network (Feedforward Neural Network, FFN for short), and the like. Therefore, each encoder block

includes a sub-module. Similarly, a decoder block includes a sub-module.

In a process of converting the target model into the target computational graph, the target model may be converted into a static computational graph including a plurality of nodes. Positions and indexes of the nodes in the target computational graph in the structural units of the target model are recorded as the target mapping table, that is, a table of correspondence relationships between the nodes and each block, as well as the submodule of the block. As shown in FIG. 5, FIG. 5 is a schematic diagram of a target mapping table according to an embodiment of this specification. A node 1 corresponds to a structural unit 0.0; a node 2 corresponds to a structural unit 0.1; a node 3 corresponds to a structural unit 0.2; a node 4 corresponds to a structural unit 1.0, until a node n corresponds to a structural unit n.n.

In the data processing method provided in the embodiments of this specification, by constructing the target mapping table, a specific structural unit in the target model corresponding to the current node is found based on the current node in the target computational graph, and relevant data processing and operations are accurately performed. Therefore, the entire target model computing flow is more clearly known and controlled, and it is ensured that data can be processed based on a target model design requirement when passing through each node in the target computational graph.

In one or more embodiments of this specification, data processing is accurately performed based on an actual situation through the data pre-process computational graph, the current node, and the data after-process computational graph, and functions such as insertion and deletion of a node in the target computational graph are achieved through the data pre-process computational graph and the data after-process computational graph. A specific implementation is as follows.

The processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result includes:

processing the target data based on the data pre-process computational graph, to obtain pre-processed data;

processing the pre-processed data based on the current node in a case of determining, based on the pre-processed data, to execute the current node, to obtain current processed data;

determining a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process table, where the after-process table includes a mapping relationship between the target structural unit and the data after-process computational graph, and the data after-process computational graph is used for performing after-processing on target processed data; and

processing the current processed data based on the data after-process computational graph, to obtain the data processing result.

The pre-processed data may be understood as processed data obtained by processing the target data by using the data pre-process computational graph. The current processed data may be understood as a processing result obtained by processing, by using the current node, the pre-processed data outputted by the data pre-process computational graph.

The data after-process computational graph may be understood as a computational graph of processing the target processed data after the current node is executed. In a case that the pre-processed data is obtained and the current node is not executed, the target processed data may be understood as the pre-processed data, that is, the pre-processed data is processed by using the data after-process computational graph, to obtain the data processing result. In a case that the pre-processed data is obtained and the current node is executed, the target processed data may be understood as the current processed data, that is, the current processed data by using the data after-process computational graph, to obtain the data processing result. Alternatively, in a case that the structural unit corresponding to the current node is not in the pre-process table, the target data is processed by using the current node, to obtain a target processing result. In this case, the target processed data may be understood as the target processing result, that is, the target processing result is processed by using the data after-process computational graph, to obtain the data processing result.

Specifically, in a case that the target structural unit is in the pre-process table, the data pre-process computational graph corresponding to the target structural unit is determined from the pre-process table, and the target data is processed by using the data pre-process computational graph, to obtain the pre-processed data. In addition, whether to continue to execute the current node is determined based on the pre-processed data (for example, if the data pre-process computational graph for detecting that an environment is ready is a necessary step before the current node is executed, when a state of the pre-processed data is "ready", a subsequent node is continued to be executed). If it is determined to continue to execute the current node, the pre-processed data is processed based on the current node, to obtain the current processed data. In this case, a computational graph is added before the current node.

After the current node is executed, whether the target structural unit is in the after-process table is determined. If the target structural unit is in the after-process table, the data after-process computational graph corresponding to the target structural unit from the after-process table, and the current processed data is processed based on the data after-process computational graph, to obtain the data processing result. If the target structural unit is not in the after-process

table, the current processed data is determined as the data processing result.

The processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result includes:

processing the target data based on the data pre-process computational graph, to obtain pre-processed data;

in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is in an after-process table, determining a data after-process computational graph corresponding to the target structural unit from the after-process table, and processing the pre-processed data based on the data after-process computational graph, to obtain the data processing result; or

determining the pre-processed data as the data processing result in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is not in the after-process table.

Specifically, in a case that the target structural unit is in the pre-process table, the data pre-process computational graph corresponding to the target structural unit is determined from the pre-process table, and the target data is processed by using the data pre-process computational graph, to obtain the pre-processed data. In a case of determining, based on the pre-processed data, to not execute the current node, whether the target structural unit is in the after-process table is determined. If the target structural unit is in the after-process table, the pre-processed data is processed by using the data after-process computational graph, to obtain the data processing result. If the target structural unit is not in the after-process table, that is, the target structural unit is not in the after-process table, the pre-processed data is determined as the data processing result. In a case that the current node is not continued to be executed, the current node is deleted.

In the data processing method provided in the embodiments of this specification, other computing flows are dynamically inserted into the target computational graph by inserting and determining, in the target model inference process, whether the data pre-process computational graph and the data after-process computational graph exist before and after the execution of the current node.

In one or more embodiments of this specification, in a case that the target structural unit is not in the pre-process table, the target data is processed by using the current node. A specific implementation is as follows.

The determining the data processing computational graph corresponding to the current node includes:

determining a target structural unit of the target model corresponding to the current node based on a target mapping table;

processing the target data based on the current node in a case that the target structural unit is not in a pre-process table, to obtain current processed data; and

determining a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process table.

Specifically, in a case that the target structural unit is not in the pre-process table, the current node is executed based on a sequence in the target computational graph, and the target data is processed by using the current node, to obtain the current processed data. In addition, after the current node is executed, whether the target structural unit is in the after-process table is determined. If the target structural unit is in the after-process table, the data after-process computational graph corresponding to the target structural unit from the after-process table.

In the data processing method provided in the embodiments of this description, data processing processes of structural units in a complex model can be managed in order through the pre-process table and the after-process table, to ensure that data flows and is processed based on a preset rule at different stages.

In one or more embodiments of this specification, after the current node is executed, the data after-process computational graph may be determined in a case that the target structural unit is in the after-process table, and the current processed data is processed by using the data after-process computational graph. A specific implementation is as follows.

The processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result includes:

processing the current processed data based on the data after-process computational graph, to obtain the data processing result.

Specifically, in a case that the target structural unit is in the after-process table, the data after-process computational graph corresponding to the target structural unit is determined from the after-process table, and the current processed data is then processed by using the data after-process computational graph, to obtain the data processing result.

In the data processing method provided in the embodiments of this description, after the current node is executed, whether the target structural unit is in the after-process table is determined. When the target structural unit is in the after-process table, the corresponding data after-process computational graph can be found from the after-process table, to ensure that necessary pre-processing is performed on the data before data enters a next node, thereby

enhancing data transmission accuracy.

In one or more embodiments of this specification, the target computational graph of the target model includes a plurality of nodes. After a computing flow of the entire current node is executed, whether a next node of the current node still exists is determined. If the next node exists, the data processing result is transmitted to the next node, and the next node is determined as the current node. A specific implementation is as follows.

After the obtaining the data processing result, the method further includes:

determining whether a next node of the current node exists;

if the next node exists, determining the next node of the current node as the current node, using the data processing result as the target data, and executing the step of determining the data processing computational graph corresponding to the current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result; and

outputting the data processing result if the next node does not exist.

Whether the current node still has a subsequent node is checked in the target model inference process. The next node is set to be the current node if the current node has the next node. The data processing result obtained by the current node is used as the target data, then the step of determining a data processing computational graph corresponding to the current node based on the newly determined current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain a data processing result is executed, and corresponding processing is continued to be performed on the target data, to obtain the data processing result. This process is repeated until execution of all the nodes in the target computational graph is completed.

In a case that the current node has no next node, it means that the model inference flow has reached the end of the target computational graph. In this case, iteration is not continued to a next node, but a current data processing result is directly outputted.

In the data processing method provided in the embodiments of this description, in this manner, it is ensured that the inputted target data can be operated and processed by the nodes layer by layer based on the design of the target computational graph, to finally obtain a model prediction or inference result.

In the data processing method provided in the embodiments of this specification, the target computational graph of the target model is determined. In a case that the target data is inputted to the target model, since the target computational graph is a computational graph that is unable to be updated when the target model is run, the target computational graph is completely

defined and compiled before the target model is run. This helps optimize the compilation of the target computational graph before model inference is performed, thereby reducing an operational load of a processor and enhancing execution efficiency of the processor. Meanwhile, by flexibly invoking the data processing computational graph, it can be ensured that the target model performs an efficient and accurate pre-processing or after-processing operation on the target data, so that dynamic insertion of a computing procedure into the target computational graph is implemented, and secondary compilation is not required, thereby reducing overheads of computational graph compilation, lowering a requirement for and consumption of computer hardware resources, and enhancing model inference efficiency.

Referring to FIG. 6, FIG. 6 is a flowchart of a processing process of a single node of a data processing method according to an embodiment of this specification, specifically including the following steps.

Step 602: obtain a target structural unit corresponding to a current node based on a target mapping table.

During actual application, a target model implemented by using a dynamic graph architecture is obtained. A dynamic computational graph of the target model is converted into a static computational graph, and a node-> block mapping table (that is, a target mapping table) is constructed based on correspondence relationships between nodes in the static computational graph and structural units in a model structure, where node represents the nodes in the static computational graph, and block represent the structural units in the model structure.

In a case of performing a model inference process, computation is performed node by node based on the static computational graph. For the current node, a block corresponding to the current node is first obtained based on the foregoing node->block mapping table.

Step 604: determine whether the target structural unit is in a pre-process table. If the target structural unit is in the pre-process table, execute step 606. If the target structure unit is not in the pre-process table, execute step 610.

Whether the block corresponding to the current node is in a pre_process table (i.e. the pre-process table in the foregoing embodiment) is determined. If the block is in the pre_process table, step 606 is executed, that is, a computational graph corresponding to the block in the pre_process table is executed. If the block is not in the pre_process table, step 610 is executed, that is, computation of the current node is directly executed.

Step 606: execute a data pre-process computational graph corresponding to the target structural unit in the pre-process table.

In a case that the block corresponding to the current node is in the pre_process table, a

computational graph corresponding to the block in the pre_process table is executed, returned data (that is, the pre-processed data in the foregoing embodiment) and a result of whether to continue the computation are obtained.

Step 608: determine whether to continue the computation. If it is determined to continue the computation, execute step 610. If it is determined to not continue the computation, executing step 612.

In a case that it is determined to continue the computation, the computation of the current node is performed. In this case, it is equivalent to inserting a computational graph before the current node, and a function of adding a computational graph is achieved. In a case that the computation is not continued, the computation of the current node is skipped, and returned data is transmitted to a next node of the current node, that is, a function of deleting the current node is achieved.

Step 610: perform the computation of the current node.

Step 612: determine whether the target structural unit is in an after-process table. If the target structural unit is in the after-process table, execute step 614. If the target structure unit is not in the after-process table, execute step 616.

Similarly, after the execution of the current node is completed, the block of the current node is also obtained based on the node-> block mapping table, and then whether the block is in the after_process table (the after-process table) is searched. If the block is in the after_process table, step 614 is executed, that is, a computational graph corresponding to the block in the after_process table is executed. If the block is not in the after_process table, the entire computing flow of the current node is directly ended.

Step 614: execute a data after-process computational graph corresponding to the target structural unit in the after-process table.

In a case that the block of the current node is in the after_process table, the computational graph corresponding to the block in the after_process table is executed, and returned data (that is, the data processing result in the foregoing embodiment) is obtained.

Step 616: end.

The entire computing flow of the current node is completed, and the obtained data processing result is transmitted to a next node of the current node.

In the data processing method provided in the embodiments of this specification, by flexibly invoking the data processing computational graph, it can be ensured that the target model performs an efficient and accurate pre-processing or after-processing operation on the target data, so that dynamic insertion of a computing procedure into the target computational

graph is implemented, and secondary compilation is not required, thereby reducing overheads of computational graph compilation, lowering a requirement for and consumption of computer hardware resources, and enhancing model inference efficiency.

Referring to FIG. 7, FIG. 7 is a flowchart of a text processing method according to an embodiment of this specification, specifically including the following steps.

Step 702: determine a target computational graph of a target model and target text inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run.

Step 704: determine, from the target computational graph, a current node for processing the target text, where the current node is any one of the plurality of nodes.

Step 706: determine a text processing computational graph corresponding to the current node, and process the target text based on the current node and/or the text processing computational graph, to obtain a text processing result.

Specifically, the text processing method is described in detail by using an example in which the target model is a deep learning text classification model.

The text classification model is constructed in a form of a dynamic graph. First, the text classification model is converted into a static computational graph to facilitate management and optimization of an inference process. According to a model structure (for example, a word embedding layer, a convolutional layer, a recurrent neural network layer, and a fully-connected layer), a node->block mapping table is constructed. Each node represents an operation step in the static computational graph, and a block corresponds to an actual structural unit of the text classification model.

At a model inference stage, in a case that the target text is inputted, node computation is performed in sequence based on the static computational graph. Assuming that a current node corresponds to the word embedding layer (block), it is found in a pre-process table that the block has a text pre-process computational graph. For example, preprocessing such as word segmentation or filling is performed on the inputted target text, to obtain pre-processed data, and whether a condition for continuing computation is satisfied (for example, whether data is valid, has a suitable length, or the like) is determined. If the condition for continuing computation is satisfied, actual computation of the current node (the word embedding layer) is performed, and the pre-processed text is converted into a word vector representation.

After the computation of the current node is completed, whether the current node has a

text pre-process computational graph in an after-process table is further checked. For example, data normalization or another after-process step is executed. If the current node has the text pre-process computational graph, the corresponding data after-process computational graph is executed, and a result is obtained. If the current node does not have the text pre-process computational graph, the computing flow of the current node is directly ended, and a computing result is transmitted to a next node.

In the text processing method provided in the embodiments of this specification, a part of the static computational graph can be flexibly inserted or deleted by invoking the data processing computational graph, to meet different data pre-processing requirements or after-processing requirements, thereby optimizing inference efficiency and effect of the entire target model.

Referring to FIG. 8, FIG. 8 is a flowchart of an image processing method according to an embodiment of this specification, specifically including the following steps.

Step 802: determine a target computational graph of a target model and a target image inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run.

Step 804: determine, from the target computational graph, a current node for processing the target image, where the current node is any one of the plurality of nodes.

Step 806: determine an image processing computational graph corresponding to the current node, and process the target image based on the current node and/or the image processing computational graph, to obtain an image processing result.

Specifically, the image processing method is described in detail by using an example in which the target model is an image classification model.

The image classification model includes a plurality of structural units, such as an input layer, a convolutional layer, a pooling layer, a fully-connected layer, and an output layer. After a dynamic graph is converted into a static computational graph, first, mapping relationships between nodes and the structural units of the model are constructed based on the static computational graph of the image classification model. For example, a node corresponds to computation of a first convolutional layer.

At a model inference stage, in a case that the target image is inputted, node computation is performed node by node in sequence based on the static computational graph, and it is learned, based on a node->block mapping table, that a current node corresponds to the first

convolutional layer.

Whether the convolutional layer is in a pre-process table is determined. If the convolutional layer is in the pre_process table, a computational graph corresponding to the convolutional layer in the pre-process table is executed, and a pre-processing operation is performed on the inputted target image based on the computational graph, to obtain normalized image data. It is determined that computation of the current node needs to be continued, and the computation of the current node (that is, the first convolutional layer) is performed, and feature extraction is performed on the pre-processed image data, to obtain a feature extraction result. Whether the current node is in an after-process table is determined. If the current node is not in the after-process table, the feature extraction result is transmitted to a next node.

In the image processing method provided in the embodiments of this specification, a part of the static computational graph can be flexibly inserted or deleted by invoking the data processing computational graph, to meet different data pre-processing requirements or after-processing requirements, thereby optimizing inference efficiency and effect of the entire target model.

Corresponding to the foregoing method embodiments, this specification further provides an embodiment of a data processing apparatus. FIG. 9 is a schematic diagram of a structure of a data processing apparatus according to an embodiment of this specification. As shown in FIG. 9, the apparatus includes:

a data determination module 902, configured to determine a target computational graph of a target model and target data inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module 904, configured to determine, from the target computational graph, a current node for processing the target data, where the current node is any one of the plurality of nodes; and

a result obtaining module 906, configured to: determine a data processing computational graph corresponding to the current node, and process the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

Optionally, the result obtaining module 906 is further configured to:

determine a target structural unit of the target model corresponding to the current node based on a target mapping table; and

determine a data pre-process computational graph corresponding to the target structural

unit from a pre-process table in a case that the target structural unit is in the pre-process table, where the pre-process table includes a mapping relationship between the target structural unit and the data pre-process computational graph, and the data pre-process computational graph is used for performing pre-processing on target processed data.

Optionally, the result obtaining module 906 is further configured to:

process the target data based on the data pre-process computational graph, to obtain pre-processed data;

process the pre-processed data based on the current node in a case of determining, based on the pre-processed data, to execute the current node, to obtain current processed data;

determine a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process table, where the after-process table includes a mapping relationship between the target structural unit and the data after-process computational graph, and the data after-process computational graph is used for performing after-processing on target processed data; and

process the current processed data based on the data after-process computational graph, to obtain the data processing result.

Optionally, the result obtaining module 906 is further configured to:

process the target data based on the data pre-process computational graph, to obtain pre-processed data;

in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is in an after-process table, determine a data after-process computational graph corresponding to the target structural unit from the after-process table, and process the pre-processed data based on the data after-process computational graph, to obtain the data processing result; or

determine the pre-processed data as the data processing result in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is not in the after-process table.

Optionally, the result obtaining module 906 is further configured to:

determine a target structural unit of the target model corresponding to the current node based on a target mapping table;

process the target data based on the current node in a case that the target structural unit is not in a pre-process table, to obtain current processed data; and

determine a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process

table.

Optionally, the result obtaining module 906 is further configured to:

process the current processed data based on the data after-process computational graph, to obtain the data processing result.

Optionally, the data determination module 902 is further configured to:

obtain the target model, use processing operations of data processing performed on the target model as the nodes, and use association relationships between the processing operations as the edges; and

construct the target computational graph based on the nodes and the edges.

Optionally, the data determination module 902 is further configured to:

construct an initial computational graph based on the nodes and the edges; and

perform a redundancy elimination operation on a target node and a target edge in the initial computational graph, to obtain the target computational graph of the target model, to run the target model in a target processor in a state of low utilization.

The apparatus further includes:

a construction module, configured to: determine structural units for constructing the target model, where the structural units are network layers of the target model;

determine structural units corresponding to the nodes in the target computational graph; and

construct the target mapping table based on mapping relationships between the nodes in the target computational graph and the structural units of the target model.

The apparatus further includes:

an after-processing module, configured to: determine whether a next node of the current node exists;

if the next node exists, determine the next node of the current node as the current node, use the data processing result as the target data, and execute the step of determining the data processing computational graph corresponding to the current node, and process the target data based on the current node and/or the data processing computational graph, to obtain the data processing result; and

output the data processing result if the next node does not exist.

Optionally, the data determination module 902 is further configured to:

determine, based on a hardware resource of computer hardware that supports machine learning model computation, the target computational graph of the target model and the target data inputted to the target model.

The above describes a schematic solution of the data processing apparatus of this embodiment. It should be noted that the technical solution of the data processing apparatus and the technical solution of the foregoing data processing method belong to the same concept. For details not described in detail in the technical solution of the data processing apparatus, refer to the descriptions of the technical solution of the foregoing data processing method.

Corresponding to the foregoing method embodiment, this specification further provides an embodiment of a text processing apparatus. FIG. 10 is a schematic diagram of a structure of a text processing apparatus according to an embodiment of this specification. As shown in FIG. 10, the apparatus includes:

a text determination module 1002, configured to determine a target computational graph of a target model and target text inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module 1004, configured to determine, from the target computational graph, a current node for processing the target text, where the current node is any one of the plurality of nodes; and

a result obtaining module 1006, configured to: determine a text processing computational graph corresponding to the current node, and process the target text based on the current node and/or the text processing computational graph, to obtain a text processing result.

The above describes a schematic solution of the text processing apparatus of this embodiment. It should be noted that the technical solution of the text processing apparatus and the technical solution of the foregoing data processing method belong to the same concept. For details not described in detail in the technical solution of the data processing apparatus, refer to the descriptions of the technical solution of the foregoing data processing method.

Corresponding to the foregoing method embodiments, this specification further provides an embodiment of an image processing apparatus. FIG. 11 is a schematic diagram of a structure of an image processing apparatus according to an embodiment of this specification. As shown in FIG. 11, the apparatus includes:

an image determination module 1102, configured to determine a target computational graph of a target model and a target image inputted to the target model, where the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

a node determination module 1104, configured to determine, from the target computational graph, a current node for processing the target image, where the current node is any one of the plurality of nodes; and

a result obtaining module 1106, configured to: determine an image processing computational graph corresponding to the current node, and process the target image based on the current node and/or the image processing computational graph, to obtain an image processing result.

The above describes a schematic solution of the image processing apparatus of this embodiment. It should be noted that the technical solution of the image processing apparatus and the technical solution of the foregoing data processing method belong to the same concept. For details not described in detail in the technical solution of the image processing apparatus, refer to the descriptions of the technical solution of the foregoing data processing method.

FIG. 12 is a block diagram of a structure of a computing device 1200 according to an embodiment of this specification. Components of the computing device 1200 includes, but are not limited to, a memory 1210 and a processor 1220. The processor 1220 is connected to the memory 1210 through a bus 1230. A database 1250 is used for storing data.

The computing device 1200 further includes an access device 1240. The access device 1240 enables the computing device 1200 to communicate via one or more networks 1260. Examples of these networks include a public switched telephone network (PSTN, Public Switched Telephone Network), a local area network (LAN, Local Area Network), a wide area network (WAN, Wide Area Network), a personal area network (PAN, Personal Area Network), or a combination of communication networks such as the Internet. The access device 1240 may include one or more of any type of network interface (for example, a network interface controller (NIC, network interface controller)) that is wired or wireless, such as an IEEE802.11 wireless local area network (WLAN, Wireless Local Area Network) wireless interface, a worldwide interoperability for microwave access (Wi-MAX, Worldwide Interoperability for Microwave Access) interface, an Ethernet interface, a universal serial bus (USB, Universal Serial Bus) interface, a cellular network interface, a Bluetooth interface, and a near field communication (NFC, Near Field Communication).

In an embodiment of this specification, the foregoing components of the computing device 1200 and other components not shown in FIG. 12 may alternatively be connected to each other, for example, through a bus. It should be understood that the block diagram of the structure of the computing device shown in FIG. 12 is provided by way of example only, and is not intended to limit the scope of this specification. A person skilled in the art may add or replace other

components as needed.

The computing device 1200 may be any type of stationary or mobile computing device, including a mobile computer or mobile computing device (for example, a tablet computer, a personal digital assistant, a laptop computer, a notebook computer, or a netbook), a mobile phone (for example, a smartphone), a wearable computing device (for example, a smart watch or smart glasses), or other types of mobile devices, or a stationary computing device such as a desktop computer or a personal computer (PC, Personal Computer). The computing device 1200 may alternatively be a mobile or stationary server.

The processor 1220 is configured to execute the following computer program/instruction. The computer program/instruction, when executed by the processor, implements the steps of the foregoing data processing method, or text processing method, or image processing method.

The embodiments in this specification are all described progressively, and the same similar parts between the various embodiments can be referred to each other. Each embodiment focuses on differences from other embodiments. Especially, the embodiment of the computing device is basically similar to the embodiment of the data processing method, so that the computing device is described simply. For related parts, refer to partial descriptions in the embodiment of the data processing method.

An embodiment of this specification further provides a computer-readable storage medium, having a computer program/instruction stored therein. The computer program/instruction, when executed by a processor, implements the steps of the foregoing data processing method, or text processing method, or image processing method.

The embodiments in this specification are all described progressively, and the same similar parts between the various embodiments can be referred to each other. Each embodiment focuses on differences from other embodiments. Especially, the embodiment of the computer-readable storage medium is basically similar to the embodiment of the data processing method, so that the computer-readable storage medium is described simply. For related parts, refer to partial descriptions in the embodiment of the data processing method.

An embodiment of this specification further provides a computer program product, including a computer program/instruction. The computer program/instruction, when executed by a processor, implements the steps of the foregoing data processing method, or text processing method, or image processing method.

The above describes a schematic solution of the computer program product of this embodiment. It should be noted that the technical solution of the computer program product and the technical solution of the foregoing data processing method belong to the same concept.

For details not described in detail in the technical solution of the computer program product, refer to the descriptions of the technical solution of the foregoing data processing method.

Particular embodiments of this specification are described above. Other embodiments fall within the scope of the attached claims. In some cases, the actions or steps recorded in the claims may be executed in an order different from that in the aforementioned embodiments and can still achieve the desired results. In addition, the process depicted in the accompanying drawings does not necessarily require a specific order or continuous orders to achieve the desired results. In some implementations, multitasking and parallel processing are also possible or may be advantageous.

The computer instruction includes a computer program code. The computer program code may be in a source code form, an object code form, an executable file form, some intermediate forms, or the like. The computer-readable medium may include: any entity or apparatus that is capable of carrying the computer program code, a recording medium, a USB flash drive, a removable hard disk, a magnetic disk, an optical disc, a computer memory, a read-only memory (ROM, Read-Only Memory), a random access memory (RAM, Random Access Memory), an electric carrier signal, a telecommunication signal and a software distribution medium, and the like. It should be noted that the content included in the computer-readable medium may be appropriately increased or decreased according to the requirements of patent practice. For example, in some regions, according to patent practice, the computer-readable medium does not include an electric carrier signal and a telecommunication signal.

It should be noted that for the foregoing method embodiments, for ease of description, they are all described as a series of action combinations. However, those skilled in the art should be aware that the embodiments of this specification are not limited by a sequence of the described actions, as according to the embodiments of this specification, and some steps can be performed in other sequences or simultaneously. In addition, a person skilled in the art should also know that all the embodiments described in this specification are preferred embodiments, and the actions and modules involved are not necessarily mandatory to the embodiments of this specification.

In the foregoing embodiments, the description of each embodiment has respective focuses. For a part that is not described in detail in an embodiment, reference may be made to related descriptions in other embodiments.

The preferred embodiments of this specification disclosed above are merely used to help explain this specification. Not all details are elaborated in optional embodiments, and the present invention is not limited to only the specific implementations. Clearly, many

modifications and variations can be made based on the content of the embodiments of this specification. These embodiments are selected and specifically described in this specification to better explain the principles and practical applications of the embodiments of this specification, so that a person skilled in the art can understand and use this specification well. This specification is limited only by the claims and the full scope and equivalents thereof.

CLAIMS

1. A data processing method, comprising:

determining a target computational graph of a target model and target data inputted to the target model, wherein the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target data, wherein the current node is any one of the plurality of nodes; and

determining a data processing computational graph corresponding to the current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain a data processing result.

2. The data processing method according to claim 1, wherein the determining the data processing computational graph corresponding to the current node comprises:

determining a target structural unit of the target model corresponding to the current node based on a target mapping table; and

determining a data pre-process computational graph corresponding to the target structural unit from a pre-process table in a case that the target structural unit is in the pre-process table, wherein the pre-process table comprises a mapping relationship between the target structural unit and the data pre-process computational graph, and the data pre-process computational graph is used for performing pre-processing on target processed data.

3. The data processing method according to claim 2, wherein the processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result comprises:

processing the target data based on the data pre-process computational graph, to obtain pre-processed data;

processing the pre-processed data based on the current node in a case of determining, based on the pre-processed data, to execute the current node, to obtain current processed data;

determining a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process table, wherein the after-process table comprises a mapping relationship between the target structural unit and the data after-process computational graph, and the data after-process computational graph is used for performing after-processing on target processed data; and

processing the current processed data based on the data after-process computational graph, to obtain the data processing result.

4. The data processing method according to any one of claims 2 to 3, wherein the processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result comprises:

processing the target data based on the data pre-process computational graph, to obtain pre-processed data;

in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is in an after-process table, determining a data after-process computational graph corresponding to the target structural unit from the after-process table, and processing the pre-processed data based on the data after-process computational graph, to obtain the data processing result; or

determining the pre-processed data as the data processing result in a case of determining, based on the pre-processed data, to not execute the current node and that the target structural unit is not in the after-process table.

5. The data processing method according to any one of claims 1 to 4, wherein the determining the data processing computational graph corresponding to the current node comprises:

determining a target structural unit of the target model corresponding to the current node based on a target mapping table;

processing the target data based on the current node in a case that the target structural unit is not in a pre-process table, to obtain current processed data; and

determining a data after-process computational graph corresponding to the target structural unit from an after-process table in a case that the target structural unit is in the after-process table.

6. The data processing method according to claim 5, wherein the processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result comprises:

processing the current processed data based on the data after-process computational graph, to obtain the data processing result.

7. The data processing method according to any one of claims 1 to 6, wherein the determining the target computational graph of the target model comprises:

obtaining the target model, using processing operations of data processing performed on the target model as the nodes, and using association relationships between the processing

operations as the edges; and

constructing the target computational graph based on the nodes and the edges.

8. The data processing method according to claim 7, wherein the constructing the target computational graph based on the nodes and the edges comprises:

constructing an initial computational graph based on the nodes and the edges; and

performing a redundancy elimination operation on a target node and a target edge in the initial computational graph, to obtain the target computational graph of the target model, to run the target model in a target processor in a state of low utilization.

9. The data processing method according to any one of claims 2 to 8, wherein before the determining the target structural unit of the target model corresponding to the current node based on the target mapping table, the method further comprises:

determining structural units for constructing the target model, wherein the structural units are network layers of the target model;

determining structural units corresponding to the nodes in the target computational graph; and

constructing the target mapping table based on mapping relationships between the nodes in the target computational graph and the structural units of the target model.

10. The data processing method according to claim 3, 4, or 6, wherein after the obtaining the data processing result, the method further comprises:

determining whether a next node of the current node exists;

if the next node exists, determining the next node of the current node as the current node, using the data processing result as the target data, and executing the step of determining the data processing computational graph corresponding to the current node, and processing the target data based on the current node and/or the data processing computational graph, to obtain the data processing result; and

outputting the data processing result if the next node does not exist.

11. The data processing method according to any one of claims 1 to 10, wherein the data processing method is applied to computer hardware that supports machine learning model computation.

12. The data processing method according to claim 11, wherein the determining the target computational graph of the target model and the target data inputted to the target model comprises:

determining, based on a hardware resource of the computer hardware that supports the machine learning model computation, the target computational graph of the target model and

the target data inputted to the target model.

13. A text processing method, comprising:

determining a target computational graph of a target model and target text inputted to the target model, wherein the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target text, wherein the current node is any one of the plurality of nodes; and

determining a text processing computational graph corresponding to the current node, and processing the target text based on the current node and/or the text processing computational graph, to obtain a text processing result.

14. An image processing method, comprising:

determining a target computational graph of a target model and a target image inputted to the target model, wherein the target model is a machine learning model; the target computational graph is constructed by a plurality of nodes and edges for connecting the nodes, and the target computational graph is a computational graph that is unable to be updated when the target model is run;

determining, from the target computational graph, a current node for processing the target image, wherein the current node is any one of the plurality of nodes; and

determining an image processing computational graph corresponding to the current node, and processing the target image based on the current node and/or the image processing computational graph, to obtain an image processing result.

15. A computing device, comprising:

a memory and a processor,

wherein the memory is configured to store a computer program/instruction; the processor is configured to execute the computer program/instruction; and the computer program/instruction, when executed by the processor, implement the steps of the method according to any one of claims 1 to 14.

16. A computer-readable storage medium, storing a computer program/instruction, wherein the computer program/instruction, when executed by a processor, implement steps of the method according to any one of claims 1 to 14.

17. A computer program product, comprising a computer program/instruction, wherein the computer program/instruction, when executed by a processor, implement steps of the

method according to any one of claims 1 to 14.