

APPARATUS AND METHOD FOR TRAINING AND EVALUATING AI MODEL

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to Chinese Patent Application No. 202311869522.0, filed on December 29, 2023, entitled "Apparatus and Method for Training and Evaluating AI Model," the entire content of which is incorporated herein by reference.

TECHNICAL FIELD

[0002] The present application relates generally to the field of artificial intelligence (AI), and more particularly to an apparatus and method for training and evaluating an AI model.

BACKGROUND

[0003] In recent years, with the rapid development of artificial intelligence technology, an increasing number of AI models have been applied to various fields. The creation of an AI model generally involves two stages: model training and model evaluation.

[0004] Model training refers to the process of training and tuning model parameters using a large amount of training data. Model evaluation refers to the process of testing an AI model generated during training using some pre-prepared and designed test data. After the model evaluation process, some test results of the evaluated model on the test data are usually generated, such as images, metrics, etc. These metrics can reflect the effect of the AI model, thereby helping algorithm personnel select a model with better performance for application in production.

[0005] Currently, the training and evaluation of AI models are typically performed entirely manually by algorithm personnel. For example, after training a model, an algorithm person would temporarily interrupt the training, manually extract the model, and run an evaluation process to evaluate the model. After the evaluation is completed, the algorithm person would decide whether to continue training based on the model evaluation results. Alternatively, after completing all training, the algorithm person would simultaneously extract multiple models generated by the training, run the evaluation process for each model in turn, and after all evaluation processes are completed, manually select the optimal model.

[0006] This fully manual training and evaluation process, due to including too many manual operations, brings about problems such as low evaluation efficiency and high labor costs.

SUMMARY

[0007] In view of the above problems, according to one aspect of the embodiments of the present application, there is provided an apparatus for training and evaluating an AI model, including: a memory for storing program instructions; and a processor coupled with the memory and configured to execute the program instructions to: create a workflow task template applied to a containerized application management platform, the workflow task template including: workflow configuration information associated with a training and evaluation task for the AI model, and stage flags corresponding to respective steps in the training and evaluation task; and in response to receiving a start command for a continuous training and evaluation task, perform the following operations: querying the workflow task template, performing stage splitting on the training and evaluation task based on the workflow configuration information and the stage flags in the workflow task template to construct a training stage task, performing artifact escape analysis based on the training stage task to generate an artifact escape analysis result associated with the training stage task, starting the training stage task based on the artifact escape analysis result associated with the training stage task, and when the training stage task reaches a predetermined training stage node, starting concurrent execution of one or more evaluation stage tasks without interrupting the training stage task.

[0008] According to another aspect of the embodiments of the present application, there is provided a method for training and evaluating an AI model, including: creating a workflow task template applied to a containerized application management platform, the workflow task template including: workflow configuration information associated with a training and evaluation task for the AI model, and stage flags corresponding to respective steps in the training and evaluation task; and in response to receiving a start command for a continuous training and evaluation task, performing the following operations: querying the workflow task template, performing stage splitting on the training and evaluation task based on the workflow configuration information and the stage flags in the workflow task template to construct a

training stage task, performing artifact escape analysis based on the training stage task to generate an artifact escape analysis result associated with the training stage task, starting the training stage task based on the artifact escape analysis result associated with the training stage task, and when the training stage task reaches a predetermined training stage node, starting concurrent execution of one or more evaluation stage tasks without interrupting the training stage task.

[0009] According to yet another aspect of the embodiments of the present application, there is provided a machine-readable medium having stored thereon program instructions which, when executed by a processor, cause the processor to perform the method for training and evaluating an AI model as described above.

[0010] According to yet another aspect of the embodiments of the present application, there is provided an apparatus for training and evaluating an AI model, including means for performing the method for training and evaluating an AI model as described above.

[0011] According to the apparatus, method, device, and machine-readable medium for training and evaluating an AI model according to the embodiments of the present application, the training and evaluation workflow configuration for an AI model is abstracted and solidified, achieving continuous training and evaluation of the AI model, reducing manual participation in the training and evaluation process, solving the problem that model training and model evaluation cannot be performed in parallel, and thus improving the efficiency of AI model development.

[0012] Furthermore, according to some embodiments of the present application, a unified single/continuous training and evaluation method for an AI model can also be implemented. In these embodiments, the platform can respond to a single training and evaluation start command or a continuous training and evaluation start command input by a user, and accordingly start a single training and evaluation procedure or a continuous training and evaluation procedure. That is, after unified training and evaluation configuration, a user can choose to run a single training and evaluation task or choose to run a continuous training and evaluation task without modifying the configuration.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The features, advantages, and technical effects of the exemplary embodiments of the present application will be described below with reference to the accompanying drawings.

[0014] In order to more clearly illustrate the technical solutions in the embodiments of the present application, the accompanying drawings required for use in the description of the embodiments of the present application will be briefly introduced below. For those of ordinary skill in the art, other drawings can also be obtained from these drawings without creative efforts.

[0015] FIG. 1 shows a schematic overall flowchart of a training and evaluation process for an AI model according to some embodiments of the present application;

[0016] FIG. 2 shows an exemplary process of workflow configuration in the training and evaluation process for an AI model according to some embodiments of the present application;

[0017] FIG. 3 shows a single training and evaluation task flow for an AI model according to some embodiments of the present application;

[0018] FIG. 4 shows an overall flowchart of a continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0019] FIG. 5 shows a schematic diagram of a stage splitting process in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0020] FIG. 6 shows a schematic diagram of a process of using a sidecar proxy to facilitate transferring artifact files in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0021] FIG. 7 shows a schematic diagram of non-intrusive Sidecar proxy container injection in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0022] FIG. 8 shows a schematic diagram of concurrent execution of multiple model evaluation tasks in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0023] FIG. 9 shows a detailed flowchart of the continuous training and evaluation task flow for an AI model according to some embodiments of the present application;

[0024] FIG. 10 shows a schematic diagram of a hardware structure of an apparatus for training and evaluating an AI model according to an embodiment of the present application.

DETAILED DESCRIPTION

[0025] The features and exemplary embodiments of various aspects of the present application will be described in detail below. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present application. However, it will be apparent to those skilled in the art that the present application may be practiced without some of these specific details. The following description of the embodiments is merely intended to provide a better understanding of the present application by illustrating examples thereof. The present application is in no way limited to any specific configurations set forth below, but covers any modifications, substitutions, and improvements in elements, components, and algorithms without departing from the spirit of the present application. In the accompanying drawings and the following description, well-known structures and techniques are not shown in order to avoid unnecessarily obscuring the present application.

[0026] To solve the problems of high manual involvement, low evaluation efficiency, and high labor costs caused by using fully manual training and evaluation workflows in artificial intelligence model training and evaluation, the present application proposes a continuous training and evaluation process for an AI model that can be implemented on a containerized application management platform (e.g., a Kubernetes-based platform). This process abstracts and solidifies the training and evaluation workflow configuration for an AI model, solving the problem that model training and model evaluation cannot be performed in parallel. Furthermore, the platform implementing the continuous training and evaluation process for an AI model according to the present application can also conveniently accommodate a single training and evaluation process for an AI model. Thus, according to some embodiments of the present application, a unified single or continuous training and evaluation process for an AI model can be implemented. The overall flow of this unified training and evaluation process will be described below with reference to FIG. 1, followed by a detailed description of the process of abstracting and solidifying the training and evaluation workflow configuration for an AI model, as well as the specific flows of the single training and evaluation process and the continuous training and evaluation process with reference to FIGS. 2 to 9, with an emphasis on the implementation flow of the continuous training and evaluation process.

[0027] FIG. 1 shows a schematic overall flowchart of a training and evaluation process for an AI model according to some embodiments of the present application. As shown in FIG. 1, the overall flow of training and evaluation for an AI model may include a workflow configuration process, a training start process, and a single training and evaluation task flow or a continuous training and evaluation task flow started according to a user's selection.

[0028] In general, the training and evaluation process can be implemented on a containerized application management platform. First, the platform can construct a task step composition of a training and evaluation task for the AI model based on domain knowledge associated with the AI model. The task step composition includes respective steps required to complete the training and evaluation task for the AI model. For example, for the image detection field, a task step composition including main steps such as an image preprocessing step, a model training step, a model evaluation step, and a post-processing step can be constructed. The platform can solidify these task steps into a workflow task template through software methods, so as to subsequently run the training and evaluation task based on the workflow task template. Then, based on the solidified workflow task template, the platform can respond to a training and evaluation start command input by a user to start a single training and evaluation task or a continuous training and evaluation task for the AI model, thereby achieving automation and parallelization of the training and evaluation functions for the AI model, and providing a unified way of storing and displaying models and reports.

[0029] In terms of specific implementation, for example, by using relevant technical means such as a containerized application management engine (e.g., Kubernetes), sidecar mechanism, object storage, process engine, and software development kit (SDK), the training and evaluation task flow for the AI model is configured in a templated manner, while mechanisms such as stage splitting and artifact escape analysis are used, so that a single training and evaluation task flow or a continuous training and evaluation task flow can be selected according to user needs. If the user chooses to start a single training and evaluation task flow, a single complete flow including the training and evaluation stages of the AI model will be executed, similar to the fully manual method, but the various steps in the training and evaluation flow will be automatically performed sequentially without much manual intervention. If the user chooses the continuous training and evaluation task flow, stage splitting and artifact escape

analysis will be automatically performed on the process task. The training stage task (also simply referred to as the training task in this text) is started based on the results of the stage splitting and artifact escape analysis, and when the training stage task reaches a predetermined training stage node, one or more evaluation stage tasks (also simply referred to as evaluation tasks in this text) are started concurrently without interrupting the training stage task, thereby achieving parallel execution of the training stage task and the evaluation stage tasks.

[0030] The process of configuration of the training and evaluation flow for an AI model, as well as the specific flows of the single training and evaluation process and the continuous training and evaluation process, will be described in detail below with reference to FIGS. 2 to 9, with an emphasis on the specific implementation flow of the continuous training and evaluation process with reference to FIGS. 4 to 9.

[0031] FIG. 2 shows an exemplary process of workflow configuration in the training and evaluation process for an AI model according to some embodiments of the present application. According to the embodiments of the present application, workflow configuration refers to, through software methods, mapping relevant operational codes, configuration, operating environment, etc., in the workflow of an algorithm expert in the AI-related field to abstracted operational steps (for example, for the image detection field, may include an image preprocessing step, a model training step, a model evaluation step, and a post-processing step), i.e., solidifying the workflow of the algorithm expert in the AI-related field into a workflow task template.

[0032] For example, the technical solution according to the embodiments of the present application can be implemented on a platform based on a containerized application management engine (e.g., Kubernetes). All training and evaluation tasks are ultimately scheduled and run in the form of containers (Pods). Therefore, in specific implementation, the relevant configuration of the algorithm expert's workflow in the AI field can be mapped to parameters such as container start commands, container images, stage artifacts, configuration files, etc., and then these parameters are used at the software level to generate a workflow task template, which is solidified into the training and evaluation process for the AI model.

[0033] Specifically, in the workflow configuration stage, the training and evaluation flow for the AI model can be mapped and parameter-configured to generate a workflow task template

for running on the containerized application management platform and associated with the training and evaluation task for the AI model. The workflow task template may include workflow configuration information associated with the training and evaluation task for the AI model. As shown in FIG. 2, for example, for an image detection model, the workflow configuration information may include: an image detection training set, training set preprocessing configuration, an image detection test set, and test set preprocessing configuration associated with the image preprocessing step; image configuration, start command configuration, resource configuration, artifact configuration, and input file configuration associated with the model training step; an image configuration, start command configuration, resource configuration, artifact configuration, and input file configuration associated with the model evaluation step; and model evaluation report analysis and model post-processing configuration associated with the post-processing step.

[0034] Furthermore, as described above, according to some embodiments of the present application, when the user chooses to start the continuous training and evaluation task flow, the platform needs to automatically perform stage splitting and artifact escape analysis on the process task, start the training stage task based on the results of the stage splitting and artifact escape analysis, and when the training stage task reaches a predetermined training stage node, start concurrent execution of one or more evaluation stage tasks without interrupting the training stage task.

[0035] In these embodiments, it is proposed that when creating the workflow task template, based on domain knowledge associated with the AI model, a task step composition of the training and evaluation task for the AI model is constructed, the task step composition including respective steps required to complete the training and evaluation task for the AI model; and for each step in the task step composition, a stage to which the step belongs is allocated to obtain a stage flag corresponding to the step. The stage flags corresponding to the respective steps may be included in the workflow task template to provide a basis for the construction of subsequent stage tasks.

[0036] For example, for a training and evaluation task of an image detection model, the task step composition typically includes an image preprocessing step, a model training step, a model evaluation step, and a post-processing step. These steps can be divided into stages based on

domain knowledge of image detection, i.e., allocating a stage to each step to provide a basis for the construction of subsequent stage tasks. For example, the training and evaluation task of an image detection model can be divided into two stages: a model training stage and a model evaluation stage. Stage flags 1 and 2 are allocated to these two stages respectively, and then corresponding stage flags are assigned to each step in the task step composition. For example, since the image preprocessing step is required for both the training stage task and the evaluation stage task, the image preprocessing step should be assigned stage flags 1 and 2, while the model training step is assigned only stage flag 1, and the model evaluation step is assigned only stage flag 2.

[0037] Therefore, the workflow task template created by the platform includes not only the workflow configuration information associated with the training and evaluation task for the AI model, but also the stage flags corresponding to the respective steps in the training and evaluation task. In this way, when the platform needs to start a stage task corresponding to a certain stage flag, it can construct the corresponding stage task to be started based on the stage flags corresponding to the respective steps in the workflow task template. For example, when it is necessary to start the model training stage task (corresponding to stage flag 1), the training stage task can be constructed based on the steps in the task step composition whose assigned stage flags include stage flag 1, so as to subsequently start the training stage task.

[0038] As shown in FIG. 1, after completing the configuration of the workflow task template for the training and evaluation flow of the AI model, the training and evaluation task can be started with one click. When starting the training and evaluation task, the user can choose to perform a single training and evaluation task or a continuous training and evaluation task as needed. After the training and evaluation task is started, the platform will run the model training container and the model evaluation container based on Kubernetes according to the workflow task template created in the workflow configuration stage, and monitor their running status.

[0039] According to the embodiments of the present application, if the user chooses to start a single training and evaluation task when starting training, the platform will directly run a single training and evaluation task for the AI model, i.e., sequentially execute a single training stage task and a single evaluation stage task for the AI model based on the workflow task

template to complete the single training and evaluation task. FIG. 3 shows a single training and evaluation task flow for an AI model according to some embodiments of the present application.

[0040] As shown in FIG. 3, using an image detection model as an example for illustration, the single training and evaluation task flow for the image detection model may include a preprocessing stage, a training stage, and an evaluation stage. Specifically, in the preprocessing stage of the training and evaluation of the image detection model, the following main operations are performed: automatically pulling the image data required for training and testing from the object storage on demand according to the image pulling conditions configured by the image detection domain expert, and simultaneously automatically preprocessing the pulled image data according to the image preprocessing algorithm configured by the algorithm expert to generate preprocessed artifacts (image data). After completing the pulling of image data and the preprocessing of image data required for the training and evaluation task of the image detection model, the training stage container is automatically executed. In the training stage, the training code of the image detection model is executed, and the preprocessed image data is used to train the image detection model. The training stage will generate some artifact files, the most important of which include the model file. Operations such as archiving these artifact files may be automatically completed in software level. After the training stage is completed, the evaluation stage of the image detection model is executed sequentially. The basic model files used in the training stage and the model files generated after training are automatically transferred to the evaluation stage container, and the relevant image test sets selected by the algorithm expert in the image detection field are also automatically transferred to the evaluation stage. The evaluation stage container generates some model report files by running the evaluation code. For an image detection model, the report typically includes metrics such as image segmentation result graphs, target detection graphs, target detection true positive rate, target detection false positive rate, and target detection false negative rate. In this stage as well, operations such as archiving these artifact files may be automatically completed in software level.

[0041] According to some embodiments of the present application, if the user chooses to start a continuous training and evaluation task when starting training, the platform will automatically perform stage splitting and artifact escape analysis on the training and evaluation task, start the

training stage task based on the results of the stage splitting and artifact escape analysis, and when the training stage task reaches a predetermined training stage node, start concurrent execution of one or more evaluation stage tasks without interrupting the training stage task, so as to perform continuous training and evaluation. FIG. 4 shows an overall flowchart of a continuous training and evaluation task flow for an AI model according to some embodiments of the present application.

[0042] Table 1 below shows some key operations involved in the continuous training and evaluation task flow for an AI model according to an example embodiment of the present application. The sequence numbers in Table 1 correspond to the sequence numbers of these operations shown in FIG. 4.

No.	Operation	Description
1	Call SDK to update dataset artifacts	During continuous training, the algorithm training code logic can use the SDK as needed to update the preprocessing artifact files of the dataset used in training.
2	SDK calls Sidecar proxy interface	The SDK interacts with the Sidecar proxy via an interface, notifying it to update dataset preprocessing artifact files or start concurrent AI model evaluation tasks.
3	Sidecar updates dataset artifact files	The Sidecar proxy actually performs the update of local dataset preprocessing artifact files.
4	Training produces artifacts	After model iteration during the training phase, related training artifacts such as AI model files are generated.
5	Call SDK to notify model generation and start parallel evaluation	The algorithm training code uses the SDK to announce the generation of training artifacts and start parallel model evaluation tasks.
6	Sidecar proxy retrieves training artifacts	The Sidecar proxy obtains the training artifact files from the notification by sharing the file system with the training container.
7	Store artifacts to object storage	The Sidecar proxy completes the upload process of local artifact files to object storage according to the object storage path of the artifact files.

8	Sidecar proxy calls a platform interface to start parallel model evaluation	Based on artifact escape analysis, the Sidecar proxy organizes the parameters required for model evaluation and calls the platform interface to start parallel model evaluation tasks.
9	Platform creates model evaluation tasks	The platform service creates corresponding model evaluation tasks based on the received parameters.
10	Model evaluation task retrieves input files	The model evaluation task retrieves the required input files from object storage according to its own configuration and runs the model evaluation task.

[0043] As described above in conjunction with FIG. 1, the continuous training and evaluation task flow involves various aspects such as stage splitting, artifact escape analysis, artifact storage and transfer, and parallel execution of model training tasks and model evaluation tasks. The operations shown in Table 1 are actually operations related to these aspects. A more detailed description of operations related to these aspects will be given below in conjunction with FIGS. 5 to 9.

[0044] FIG. 5 shows a schematic diagram of a stage splitting process in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application. As shown in FIG. 5, stage splitting refers to automatically splitting the evaluation stage task out of the training and evaluation task for the AI model, i.e., initially starting only the training stage task. Specifically, when the platform receives the start command for the continuous training and evaluation task, the platform will query the workflow task template, and perform stage splitting on the training and evaluation task based on the workflow configuration information and the stage flags in the workflow task template, to construct a training stage task for the AI model.

[0045] As previously mentioned, according to the embodiments of the present application, it is proposed that when creating the workflow task template, based on domain knowledge associated with the AI model, a task step composition of the training and evaluation task for the AI model is constructed, and for each step in the task step composition, a stage to which the step belongs is allocated to obtain a stage flag corresponding to the step. The stage flags

corresponding to the respective steps may be included in the workflow task template to provide a basis for the construction of subsequent stage tasks. In this way, when the platform needs to start a stage task corresponding to a certain stage flag, it can construct the corresponding stage task to be started based on the stage flags corresponding to the respective steps in the workflow task template.

[0046] Based on the stage flags assigned in the workflow task template, when it is necessary to start the training stage task corresponding to stage flag 1, during the stage splitting process, the training stage task can be constructed based on the steps in the task step composition whose assigned stage flags include stage flag 1.

[0047] Furthermore, according to the embodiments of the present application, after the platform performs stage splitting and constructs the training stage task, it also needs to perform artifact escape analysis. Specifically, the algorithm expert can configure corresponding artifacts (e.g., including input artifacts and output artifacts) for each step involved in the training and evaluation task. These artifacts need to be transferred and used between multiple steps. However, because stage tasks have been split, the original artifact transfer method applied between steps in a single task may no longer be applicable. Therefore, it is necessary to analyze and specially handle the situation where artifacts cross stages to transfer, which is also referred to as performing artifact escape analysis herein.

[0048] Specifically, when a certain stage task is to be started, the workflow task template is first obtained. The workflow task template contains the input artifacts and output artifacts of each step configured by the algorithm expert. The input artifacts and output artifacts of these steps are traversed, and the transfer relationships of these artifacts between the steps are analyzed (e.g., artifact 1 is generated by step 1 and used as an input in step 2). After obtaining the transfer relationships of the artifacts between the steps, artifacts whose transfer relationships cross different stages in the training and evaluation task are filtered out (e.g., an artifact needs to be transferred between step 1 and step 2, where step 1 belongs to the training stage with stage flag 1, and step 2 belongs to the evaluation stage with stage flag 2). These artifacts are then treated as escaped artifacts. When escaped artifacts exist, the platform first cancels the original processing logic for these escaped artifacts in the single task, and then, based on the stage flags corresponding to the respective steps in the stage task, establishes

transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the stage task.

[0049] It should be noted that when starting the training stage task and the evaluation stage task, artifact cross-stage transfer may occur in both cases, and therefore, corresponding artifact escape analysis is required for both. Specifically, the artifact escape analysis performed when starting the training stage task includes: based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the training stage task between the respective steps; filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, to be treated as escaped artifacts; and based on stage flags corresponding to the respective steps in the training stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the training stage task. Correspondingly, the artifact escape analysis performed when starting the evaluation stage task includes: based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the evaluation stage task between the respective steps; filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, to be treated as escaped artifacts; and based on stage flags corresponding to the respective steps in the evaluation stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the evaluation stage task.

[0050] Furthermore, regarding the transfer of artifact escape analysis results, according to the embodiments of the present application, it is proposed that the transfer of artifact escape analysis results can be performed in the following manner: configuring an associated proxy container for the stage task, and using the proxy container to store artifact files according to the artifact escape analysis result and to facilitate transferring of the artifact files between the training stage task and the evaluation stage tasks.

[0051] For example, the artifact escape analysis result associated with the training stage task can be passed to the sidecar proxy container in the form of environment variables. At the same time, the sidecar proxy container shares storage with the training container by mounting the same storage, so it can obtain the artifacts generated by the training stage task. After the training task achieves a stage result, the training container notifies the sidecar proxy container to start the model evaluation task via a dedicated software development kit (SDK). The sidecar proxy

container processes all escaped artifacts (e.g., packaging, archiving) based on the artifact escape analysis result obtained from the environment variables, and finally starts the model evaluation task through the platform service interface, passing the archiving information of the current escaped artifacts to the platform service via the interface.

[0052] FIG. 6 shows a schematic diagram of a process of using a sidecar proxy to facilitate transferring artifact files in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application. As shown in FIG. 6, when transferring of an artifact crosses different stage tasks, the artifact needs to be marked as an escaped artifact, and when a subsequent independent evaluation stage task is initiated, the sidecar proxy container can facilitate storing and transferring the artifact files.

[0053] According to some embodiments of the present application, in order to enable the model training container to autonomously complete operations such as updating preprocessed artifacts of the dataset used for training, storing training stage artifacts (models), and starting concurrent model evaluation tasks during the operation of the model training container, a non-intrusive proxy container injection method is adopted in the technical implementation. The proxy container can be, for example, a Sidecar proxy container based on the Sidecar architecture pattern for the containerized application management platform.

[0054] FIG. 7 shows a schematic diagram of non-intrusive Sidecar proxy container injection in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application. Since actions such as initiating the storage of training stage artifacts and updating dataset preprocessed artifacts in the training stage task involve complex and time-consuming operations, and starting concurrent evaluation tasks requires calling a platform service interface, it is inappropriate for the training container itself to implement these operations. To solve this problem, in the embodiments of the present application, an approach combining an SDK and a Sidecar proxy container is proposed, where the Sidecar proxy container performs the complex and time-consuming operations and calls the platform service interface, while the SDK layer only encapsulates a simple interface layer.

[0055] As shown in FIG. 7, the training stage task is configured with an associated Sidecar proxy container. The Sidecar proxy container can interact with the training stage task and the evaluation stage tasks via a dedicated SDK to perform artifact storage and artifact transfer

between the Sidecar proxy container and the training stage task and the evaluation stage tasks. Specifically, by automatically injecting a self-developed Sidecar proxy container into the container to which the training task on Kubernetes belongs, the training container only needs to communicate with the Sidecar proxy container via the local network (this communication will also be encapsulated by a python SDK), while the Sidecar proxy container can share part of the file system with the training container, i.e., complete required file storage or transfer operations synchronously or asynchronously as needed, or call a remote platform service interface to start concurrent execution of multiple model evaluation tasks.

[0056] FIG. 8 shows a schematic diagram of concurrent execution of multiple model evaluation tasks in the continuous training and evaluation task flow for an AI model according to some embodiments of the present application. As shown in FIG. 8, in the embodiments according to the present application, the training task and the evaluation tasks can run in parallel without affecting each other, and multiple evaluation tasks can also run concurrently. For example, when the training stage task reaches a predetermined training stage node, concurrent execution of one or more evaluation stage tasks can be started without interrupting the training stage task. In the embodiments of the present application, the predetermined training stage node may include: a node where the training stage task achieves a stage model training goal, or a node where the training stage task reaches a predefined number of training iterations. For example, when the training stage task reaches a predetermined first training stage node, a first evaluation stage task can be started for the training model obtained at the first training stage node; when the training stage task reaches a predetermined second training stage node, a second evaluation stage task can be started for the training model obtained at the second training stage node; and so on. While starting concurrent execution of multiple evaluation stage tasks, the training stage task can continue. Additionally, as shown in Table 1, the Sidecar proxy container can be used to call the platform service interface to start concurrent execution of multiple model evaluation tasks.

[0057] According to the embodiments of the present application, using the SDK in combination with the Sidecar proxy to implement the storage and transfer of artifact files and the calling of remote platform service interface will have the following advantages: stripping this logic from the training code of the AI model reduces the difficulty of algorithm

implementation and code complexity; the specific implementation is completed by the Sidecar proxy, achieving separation of logic and algorithm, increasing flexibility for subsequent modifications to the implementation logic; and avoiding direct calls to the platform service interface by the training code, thus preventing exposure of the platform service interface.

[0058] Various possible operations and steps involved in the continuous training and evaluation process for an AI model based on a containerized application management platform have been described above with reference to FIGS. 4 to 8. To more clearly illustrate the overall flow of the continuous training and evaluation process, the continuous training and evaluation process is further described below with reference to FIG. 9.

[0059] FIG. 9 shows a detailed flowchart of the continuous training and evaluation task flow for an AI model according to some embodiments of the present application. The continuous training and evaluation task flow can be implemented on a containerized application management platform. Here, it should be noted that in the previous description, the training and evaluation task flow (including the single training and evaluation task flow and the continuous training and evaluation task flow) provided by the present application was mainly described using the image detection model in the image processing field as an example. However, it is understandable that the training and evaluation task flow can be applied to the training and evaluation of different AI models in various fields. For different AI models in different fields, the platform can create a workflow task template applied to the containerized application management platform according to workflow configuration information associated with the training and evaluation task for the AI model input by algorithm experts in the relevant field. The workflow task template may include: workflow configuration information associated with the training and evaluation task for the AI model, and stage flags corresponding to respective steps in the training and evaluation task. Therefore, there may be multiple workflow task templates associated with different AI models in different fields on the platform. Accordingly, when a certain training or evaluation task is to be started, it is necessary to query the workflow task template to which the training or evaluation task belongs.

[0060] As shown in FIG. 9, in response to receiving a start command for a continuous training and evaluation task, the platform may perform the following operations: querying the workflow task template, performing stage splitting on the training and evaluation task based on the

workflow configuration information and the stage flags in the workflow task template to construct a training stage task, performing artifact escape analysis based on the training stage task to generate an artifact escape analysis result associated with the training stage task, and starting the training stage task based on the artifact escape analysis result associated with the training stage task.

[0061] During the execution of the training stage task, the following operations may be performed: performing model training iterations; updating dataset preprocessed artifacts through the SDK to notify the sidecar proxy as needed during the iterations; when the training stage task reaches a predetermined training stage node (e.g., achieving a stage model training goal or a predefined number of training iterations), notifying the sidecar proxy via the SDK to start an evaluation stage task without interrupting the training stage task; using the sidecar proxy to store escaped artifacts according to the stage operation configuration and the artifact escape analysis result; the sidecar proxy calling the platform service interface to start the evaluation stage task; continuing subsequent model training.

[0062] As shown in FIG. 9, the platform can start a corresponding evaluation stage task each time the training stage task reaches a predetermined training stage node without interrupting the training stage task, thereby achieving concurrent execution of multiple evaluation stage tasks (evaluation task 1 to evaluation task n) on the platform. Moreover, for each evaluation stage task, the following operations may be performed: querying the workflow task template; constructing the evaluation stage task based on the workflow configuration information and the stage flags in the workflow task template, performing artifact escape analysis based on the evaluation stage task to generate an artifact escape analysis result associated with the evaluation stage task, and starting the evaluation stage task based on the artifact escape analysis result associated with the evaluation stage task.

[0063] In summary, using the continuous training and evaluation flow according to the present application, automation and parallelization of training and evaluation functions for various AI models in different fields can be achieved. Furthermore, using the SDK in combination with the Sidecar proxy to implement the storage and transfer of artifact files and the calling of the platform service interface means that the training task container does not need

to directly call the platform service interface, thus avoiding exposure of the platform service interface.

[0064] FIG. 10 shows a schematic diagram of a hardware structure of an apparatus for training and evaluating an AI model according to an embodiment of the present application.

[0065] The apparatus may include a processor 1001 and a memory 1002 storing program instructions.

[0066] Specifically, the processor 1001 may include a central processing unit (CPU), or an application specific integrated circuit (ASIC), or may be configured as one or more integrated circuits implementing the embodiments of the present application.

[0067] The memory 1002 may include mass storage for data or instructions. By way of example and not limitation, the memory 1002 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a universal serial bus (USB) drive, or a combination of two or more of these. Where appropriate, the memory 1002 may include removable or non-removable (or fixed) media. Where appropriate, the memory 1002 may be internal or external to an integrated gateway disaster recovery device. In a specific embodiment, the memory 1002 is a non-volatile solid-state memory.

[0068] In a specific embodiment, the memory 1002 includes read only memory (ROM). Where appropriate, the ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory, or a combination of two or more of these.

[0069] The memory may include read only memory (ROM), random access memory (RAM), magnetic disk storage media devices, optical storage media devices, flash memory devices, electrical, optical, or other physical/tangible storage devices. Thus, in general, the memory includes one or more tangible (non-transitory) readable storage media (e.g., memory devices) encoding software that includes computer-executable instructions, and when the software is executed (e.g., by one or more processors), it is operable to perform operations described with reference to the method for training and evaluating an AI model according to the embodiments of the present application.

[0070] The processor 1001 reads and executes the program instructions stored in the memory 1002 to implement the method for training and evaluating an AI model as described in the above embodiments.

[0071] In one example, the apparatus for training and evaluating an AI model may further include a communication interface 1003 and a bus 1010. As shown in FIG. 10, the processor 1001, the memory 1002, and the communication interface 1003 are connected via the bus 1010 and communicate with each other.

[0072] The communication interface 1003 is mainly used to enable communication between modules, devices, units, and/or equipment in the embodiments of the present application.

[0073] The bus 1010 includes hardware, software, or both, coupling components of the apparatus to each other. By way of example and not limitation, the bus may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a Front Side Bus (FSB), a HyperTransport (HT) interconnect, an Industry Standard Architecture (ISA) bus, an InfiniBand interconnect, a Low Pin Count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCI-X) bus, a Serial Advanced Technology Attachment (SATA) bus, a Video Electronics Standards Association Local (VLB) bus, or another suitable bus, or a combination of two or more of these. Where appropriate, the bus 1010 may include one or more buses. Although the embodiments of the present application describe and illustrate a particular bus, the present application contemplates any suitable bus or interconnect.

[0074] In combination with the method for training and evaluating an AI model in the above embodiments, the embodiments of the present application may be implemented by a machine-readable medium. The machine-readable medium has stored thereon program instructions, which, when executed by a processor, cause the processor to perform the method for training and evaluating an AI model as described in the above embodiments.

[0075] It should be clarified that the present application is not limited to the specific configurations and processes described above and shown in the drawings. For the sake of brevity, detailed descriptions of known methods are omitted here. In the above embodiments, several specific steps are described and shown as examples. However, the method processes of the present application are not limited to the specific steps described and shown. Those skilled

in the art can make various changes, modifications, and additions, or change the order of steps, after appreciating the spirit of the present application.

[0076] The functional blocks shown in the structural block diagrams described above can be implemented as hardware, software, firmware, or a combination thereof. When implemented in hardware, it can be, for example, an electronic circuit, an application specific integrated circuit (ASIC), appropriate firmware, a plug-in, a function card, etc. When implemented in software, the elements of the present application are programs or code segments used to perform the required tasks. The programs or code segments can be stored on a machine-readable medium or transmitted over a transmission medium or communication link via a data signal carried by a carrier wave. A "machine-readable medium" may include any medium that can store or transmit information. Examples of machine-readable media include electronic circuits, semiconductor memory devices, ROM, flash memory, erasable ROM (EROM), floppy disks, CD-ROMs, optical disks, hard disks, fiber optic media, radio frequency (RF) links, and the like. The code segments may be downloaded via a computer network such as the Internet, an intranet, or the like.

[0077] It should also be noted that the exemplary embodiments mentioned in the present application describe some methods based on a series of steps. However, the present application is not limited to the order of the steps described; that is, the steps may be performed in the order described in the embodiments, or in a different order than in the embodiments, or several steps may be performed simultaneously.

[0078] Various aspects of the present application are described above with reference to flowcharts and/or block diagrams of methods and apparatuses according to embodiments of the present application. It will be understood that each block of the flowcharts and/or block diagrams, and combinations of blocks in the flowcharts and/or block diagrams, can be implemented by program instructions. These program instructions may be provided to a processor of a general-purpose computer, special-purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions executed by the processor of the computer or other programmable data processing apparatus enable the implementation of the functions/acts specified in one or more blocks of the flowcharts and/or block diagrams. Such a processor may be, but is not limited to, a general-purpose processor,

special-purpose processor, special application processor, or field-programmable logic circuitry. It will also be understood that each block of the block diagrams and/or flowcharts, and combinations of blocks in the block diagrams and/or flowcharts, can also be implemented by special-purpose hardware that performs the specified functions or acts, or by a combination of special-purpose hardware and computer instructions.

[0079] The above description is merely specific embodiments of the present application. Those skilled in the art can clearly understand that for the convenience and brevity of description, the specific working process of the apparatus described above may refer to the corresponding process in the foregoing method embodiments, which will not be repeated here. It should be understood that the protection scope of the present application is not limited thereto. Any equivalent modifications or substitutions that can be easily conceived by those skilled in the art within the technical scope disclosed in the present application shall be covered by the protection scope of the present application.

WHAT IS CLAIMED IS:

1. An apparatus for training and evaluating an artificial intelligence (AI) model, comprising:

a memory for storing program instructions; and

a processor coupled with the memory and configured to execute the program instructions to:

create a workflow task template applied to a containerized application management platform, the workflow task template comprising: workflow configuration information associated with a training and evaluation task for the AI model, and stage flags corresponding to respective steps in the training and evaluation task; and

in response to receiving a start command for a continuous training and evaluation task, perform the following operations:

querying the workflow task template,

performing stage splitting on the training and evaluation task based on the workflow configuration information and the stage flags in the workflow task template to construct a training stage task,

performing artifact escape analysis based on the training stage task to generate an artifact escape analysis result associated with the training stage task,

starting the training stage task based on the artifact escape analysis result associated with the training stage task, and

when the training stage task reaches a predetermined training stage node, starting concurrent execution of one or more evaluation stage tasks without interrupting the training stage task.

2. The apparatus according to claim 1, wherein starting concurrent execution of one or more evaluation stage tasks comprises performing the following operations for each evaluation stage task of the one or more evaluation stage tasks:

querying the workflow task template;

constructing the evaluation stage task based on the workflow configuration information and the stage flags in the workflow task template,

performing artifact escape analysis based on the evaluation stage task to generate an artifact escape analysis result associated with the evaluation stage task, and

starting the evaluation stage task based on the artifact escape analysis result associated with the evaluation stage task.

3. The apparatus according to claim 1, wherein the predetermined training stage node comprises: a node where the training stage task achieves a stage model training goal, or a node where the training stage task reaches a predefined number of training iterations.

4. The apparatus according to claim 1, wherein performing artifact escape analysis based on the training stage task comprises:

based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the training stage task between the respective steps, the artifacts comprising input artifacts and output artifacts;

filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, the filtered artifacts to be treated as escaped artifacts; and

based on stage flags corresponding to the respective steps in the training stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the training stage task.

5. The apparatus according to claim 2, wherein performing artifact escape analysis based on the evaluation stage task comprises:

based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the evaluation stage task between the respective steps, the artifacts comprising input artifacts and output artifacts;

filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, the filtered artifacts to be treated as escaped artifacts; and

based on stage flags corresponding to the respective steps in the evaluation stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the evaluation stage task.

6. The apparatus according to any one of claims 1 to 5, wherein the training stage task is configured with an associated proxy container, and starting the evaluation stage task in the training and evaluation task comprises: using the proxy container to start the concurrent execution of the one or more evaluation stage tasks by calling a service interface of the containerized application management platform.

7. The apparatus according to claim 6, wherein the operations further comprise: using the proxy container to store artifact files according to the artifact escape analysis result, and to facilitate transferring of the artifact files between the training stage task and the one or more evaluation stage tasks.

8. The apparatus according to claim 6, wherein the training stage task interacts with the proxy container via a dedicated software development kit (SDK) to notify the proxy container to update artifacts associated with the training stage task and to start the concurrent execution of the one or more evaluation stage tasks.

9. The apparatus according to claim 6, wherein the proxy container is a Sidecar proxy container based on a Sidecar architecture pattern for the containerized application management platform.

10. The apparatus according to claim 1, wherein the processor is further configured to execute the program instructions to:

in response to receiving a start command for a single training and evaluation task, sequentially execute a single training stage task and a single evaluation stage task for the AI model based on the workflow task template.

11. The apparatus according to claim 1, wherein creating the workflow task template applied to the containerized application management platform comprises:

based on domain knowledge associated with the AI model, constructing a task step composition of the training and evaluation task for the AI model, the task step composition comprising respective steps required to complete the training and evaluation task for the AI model; and

allocating, for each step in the task step composition, a stage to which the step belongs, to obtain a stage flag corresponding to the step.

12. The apparatus according to claim 11, wherein the AI model comprises an image detection model, and the task step composition comprises an image preprocessing step, a model training step, a model evaluation step, and a post-processing step.

13. A method for training and evaluating an artificial intelligence (AI) model, comprising:
creating a workflow task template applied to a containerized application management platform, the workflow task template comprising: workflow configuration information associated with a training and evaluation task for the AI model, and stage flags corresponding to respective steps in the training and evaluation task; and

in response to receiving a start command for a continuous training and evaluation task, performing the following operations:

querying the workflow task template,

performing stage splitting on the training and evaluation task based on the workflow configuration information and the stage flags in the workflow task template to construct a training stage task,

performing artifact escape analysis based on the training stage task to generate an artifact escape analysis result associated with the training stage task,

starting the training stage task based on the artifact escape analysis result associated with the training stage task, and

when the training stage task reaches a predetermined training stage node, starting concurrent execution of one or more evaluation stage tasks without interrupting the training stage task.

14. The method according to claim 13, wherein starting concurrent execution of one or more evaluation stage tasks comprises performing the following operations for each evaluation stage task of the one or more evaluation stage tasks:

querying the workflow task template;

constructing the evaluation stage task based on the workflow configuration information and the stage flags in the workflow task template,

performing artifact escape analysis based on the evaluation stage task to generate an artifact escape analysis result associated with the evaluation stage task, and

starting the evaluation stage task based on the artifact escape analysis result associated with the evaluation stage task.

15. The method according to claim 13, wherein the predetermined training stage node comprises: a node where the training stage task achieves a stage model training goal, or a node where the training stage task reaches a predefined number of training iterations.

16. The method according to claim 13, wherein performing artifact escape analysis based on the training stage task comprises:

based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the training stage task between the respective steps, the artifacts comprising input artifacts and output artifacts;

filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, the filtered artifacts to be treated as escaped artifacts; and

based on stage flags corresponding to the respective steps in the training stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the training stage task.

17. The method according to claim 14, wherein performing artifact escape analysis based on the evaluation stage task comprises:

based on the workflow task template, analyzing transfer relationships of artifacts of respective steps in the evaluation stage task between the respective steps, the artifacts comprising input artifacts and output artifacts;

filtering out artifacts whose transfer relationships cross different stages in the training and evaluation task, the filtered artifacts to be treated as escaped artifacts; and

based on stage flags corresponding to the respective steps in the evaluation stage task, establishing transfer processing logic for the escaped artifacts as the artifact escape analysis result associated with the evaluation stage task.

18. The method according to any one of claims 13 to 17, wherein the training stage task is configured with an associated proxy container, and starting the evaluation stage task in the training and evaluation task comprises: using the proxy container to start the concurrent execution of the one or more evaluation stage tasks by calling a service interface of the containerized application management platform.

19. The method according to claim 18, further comprising: using the proxy container to store artifact files according to the artifact escape analysis result, and to facilitate transferring of the artifact files between the training stage task and the one or more evaluation stage tasks.

20. The method according to claim 18, wherein the training stage task interacts with the proxy container via a dedicated software development kit (SDK) to notify the proxy container to update artifacts associated with the training stage task and to start the concurrent execution of the one or more evaluation stage tasks.

21. The method according to claim 18, wherein the proxy container is a Sidecar proxy container based on a Sidecar architecture pattern for the containerized application management platform.

22. The method according to claim 13, further comprising:

in response to receiving a start command for a single training and evaluation task, sequentially executing a single training stage task and a single evaluation stage task for the AI model based on the workflow task template.

23. The method according to claim 13, wherein creating the workflow task template applied to the containerized application management platform comprises:

based on domain knowledge associated with the AI model, constructing a task step composition of the training and evaluation task for the AI model, the task step composition comprising respective steps required to complete the training and evaluation task for the AI model; and

allocating, for each step in the task step composition, a stage to which the step belongs, to obtain a stage flag corresponding to the step.

24. The method according to claim 23, wherein the AI model comprises an image detection model, and the task step composition comprises an image preprocessing step, a model training step, a model evaluation step, and a post-processing step.

25. A machine-readable medium having stored thereon program instructions which, when executed by a processor, cause the processor to perform the method for training and evaluating an artificial intelligence (AI) model according to any one of claims 13 to 24.

26. An apparatus for training and evaluating an artificial intelligence (AI) model, comprising means for performing the method for training and evaluating an AI model according to any one of claims 13 to 24.