BML Document

2020-11-10

百度智能云

Contents

Product Description

Introduction

Baidu Machine Learning (BML) is an AI development platform with complete features, and it provides a one-stop service set for building features of artificial intelligence models. While oriented towards users and providing machine learning and in-depth learning environment, it realizes full-cycle service capabilities of AI construction from data source management, data annotation, data set storage, data preprocessing, model training and production to model management, prediction service management and full-service monitoring.

The platform provides a high-performance cluster training environment, massive algorithm frameworks and model cases, as well as convenient prediction service tools. Users can focus on the models and algorithms themselves, and obtain high-quality models and prediction results.

Product Feature

Data processing engine

The data processing engine provides functional modules such as data source management, data import, data annotation, data exploration and processing and data feature engineering. It reasonably and efficiently manages the data resources required for producing models.

Model production engine

The model production engine provides interactive, visual, automatic, intelligent vision modeling production methods and supports distributed scheduling resources for model training.

Model application engine

The model application engine first orderly stores and manages the models. The prediction service supports control and scheduling of access traffic to provide stable and efficient prediction service for upper-level model application.

Advantages

Mainstream framework support

The BML platform provides machine learning and in-depth learning development capability, supports mainstream development frameworks and supports the installation of third-party software packages to custom other frameworks. PaddlePaddle is an in-depth learning framework self-developed by Baidu AI Cloud platform. The BML platform always maintains support for its latest version.

Multiple modeling methods

BML platform provides multiple modeling methods, such as, interactive (Notebook), visualization (drag and drop components), task-based (task modeling), production line (intelligent visual), which is suitable for users with different R & D capabilities to quickly implement model training, evaluation and prediction.

Full-cycle model management

The BML platform provides full-cycle service for model production and management, including data storage, data annotation, data processing, model training, model prediction, service monitoring, management of operation and maintenance of refined scheduling and user management. It provides model training platforms and mature vertical model production lines to support quick development and supports introduction and management of third-party models.

Open product feature interface

The BML platform provides Open API/SDK interface to facilitate the seamless connection of upper-level business applications. Product design has clear levels and interfaces are open, which is conductive to effective connection with existing IT work environments such as customers' private cloud environments, local servers, big data platforms and operation and maintenance platforms.

Domestic Kunlun chip

Kunlun chip adopts the advanced AI architecture of Baidu, which is well-suited for cloud computing requirements of commonly used deep learning and machine learning algorithms, adaptive to the computing needs of various terminal scenarios. BML launched the Kunlun chip package in Suzhou area, seamlessly supporting Baidu PaddlePaddle framework by providing excellent performance and efficiency.

Interpretations of Terms

TensorFlow

TensorFlow is an open source software library that uses data flow diagrams for numerical computation. It is widely used in the programming of various machine learning algorithms. It is developed and maintained by Google Brain, a Google artificial intelligence team.

PaddlePaddle

PaddlePaddle is an open source deep learning framework launched by Baidu, which supports advanced algorithms such as machine vision, natural language processing and recommendation system, etc. Paddle (Parallel Distributed Deep Learning) has the characteristics of ease of use, flexibility, high efficiency and expandability. See Official Website for details.

BML currently supports Paddle Fluid v1.5 deep learning framework

Baidu Object storage(BOS)

BOS (Baidu Object storage) provides stable, secure, efficient and highly extended storage services, and supports any type of data storage, such as text, multimedia, binary and so on, with a single file of up to 5TB. BML accesses data through BOS, saves training results and logs to the designated BOS address, so when opening BML service, authorization is required to access user's BOS address.

Notebook

Notebook provides Jupyter, a visual code running environment with built-in algorithms such as TensorFlow, PyTorch, Keras, Caffe, Mxnet, Chainer and PaddlePaddle, for data processing and modeling.

Job Modeling

Provide a high-performance computing environment for large-scale distributed model training and optimization. Including deep learning job, machine learning job, AutoDL job, AutoML job.

Prediction Service

Model prediction service feature module provided by BML platform. According to the demand of the model application, reasonably configure and schedule the service resources, build and deploy the highly available online prediction cluster services.

Prediction Model

Prediction model: it is the information set of model data, deployment image and configuration logic required for deployment of prediction service.

Online Prediction

According to the online API generated by the user model, users can send requests to the API for data prediction, or they can

establish multi-version services for small traffic experiments.

Container Image

The software environment required for service operation, including OS, basic library, ML framework, prediction service SDK and user-defined logic, etc.

Use Limitation

- The BOS path used by Notebook instance cannot be the same as the BOS path used by job modeling, otherwise, the content on the job modeling BOS will be overwritten by Notebook. If Notebook uses bos:/test/xxxxx, it will synchronize data with bos:/test/xxxxx regularly. For job modeling, bos:/test/xxxxx and its subdirectories cannot be used. It is recommended to use bos:/test/yyyyyy to distinguish the BOS paths of the two.
- BML currently only supports reading or writing "Bucket" in North China-Beijing region. When creating "Bucket", you need to select North China Beijing ("Bucket" has region attribute and can only be located in one region. After "Bucket" name is created, the region to which it belongs cannot be modified).

Customer Case

BML can help enterprises and developers to realize various business scenarios such as image recognition, video analysis, voice recognition, recommendation and prediction and empower many industries such as finance, education, medicine, security, retail and industrial manufacturing.

Typical customer case:

Internet lending risk control

Establish a risk control model for banks' Internet lending business to predict whether the Internet lending applicant will breach the contract or not. Based on the data in the bank, the sample size is 60,000, the ratio of positive and negative samples is 10:1, and there are more than 50 feature fields. With the GBDT algorithm, the accuracy rate and recall rate of the Internet lending risk control admission model have been greatly improved. BML facilitates the intelligence of the finance industry, builds a risk prevention and control system based on artificial intelligence and big data, controls business risks, improves capital returns and actively adapts to the development trend of new finance.

Precision marketing

Predicting potential customers of financial products for precision marketing is a typical classification problem. When the ratio of positive and negative samples is extremely unbalanced (1:5000), you can sue the GBDT algorithm and AutoML parameters for automatic tuning, let the models to learn the optimal parametric solutions, and achieve a high accuracy rate on the test set. BML uses artificial intelligence technology to support the rapid implementation of various innovative businesses in Internet finance and assists companies to quickly enter the market segment with high-efficient and personalized financial product, to expand the asset scale.

Industrial routing inspection

Have regular inspections of equipment (such as transmission lines) to ensure the safe production of power systems. Since the lines pass through various areas, some areas have bad environments and it's difficult for staff to access to it, routing inspection by unmanned aerial vehicles is adopted. The unmanned aerial vehicle flies according to the predetermined satellite positioning route, shoots transmission lines and returns images to the backend. Through recognition and analysis of the images by BML, suspicious failures can be discovered and then reported to the staff for further judgment and analysis. Quality of the photos is judged, and photos of low quality are filtered out. Algorithms of object detection and segmentation are then applied to detect components such as wires and insulators and other parts where problems may occur to position and alarm. During training, image data enhancement and parameter optimization are used to improve the training effect.

Information extraction

Extract key elements required by customers from credit granting proposals, such as company name, credit limit, credit validity period and credit type, for subsequent business approval and handling. It is a typical series annotation matter in NLP. With the ERNIE pre-training model based on Baidu data training, the accuracy of recognizing various information has reached a top level, which greatly improves the efficiency of business review.

Product Pricing

Product Pricing

Billing Mode

BML products support two billing methods: pay per use (post-paid) and purchase package (prepaid).

Pay per use supported areas: North China-Beijing Package purchase supported areas: North China-Beijing

Pay per Use

BML products are billed per use. According to the selected resource package configuration and the number of running instances, the real-time billing is based on the usage time (minute-level). The specific charging rules are as follows:

- Charge by minute, less than 1 minute is counted as 1 minute.
- Charge by hour, i.e. hourly deducting fees and generating bills on Beijing time. The billing time is within 1 hour after the end of the current billing cycle. For example, the bill during 10:00-11:00 is generated before 12:00. The specific time depends on the system billing time.
- Before using BML, you need to ensure that the account has no arrears.

Billing formula
Fee = package unit price x number of package instances x service time

Time measurement method: the use time of workspace module, training module and prediction module only includes the statistical time when the task status is "running".

Package

BML products support the purchase of resource package in the form of usage (prepaid). After purchasing the package, your BML consumption will be deducted from the package first, and the excess will be paid per use. If more than one package is purchased, the package purchased first will be deducted in priority according to the purchase order. Currently, the resource package only supports two package configuration types used by the workarea feature module: GPU Instance_P4_12 Core 32GB Memory x1 card and GPU Instance_V100_12 core 50GB Memory x1 Card.

You can purchase package directly on the "BML - > Package" page of Baidu AI Cloud management console. When purchasing, and select the area, package configuration, purchase duration and effective time according to the prompts.

Pricing

The package price of BML product pay per use is as follows:

Modules	Resource package configuration (instance)	Unit price (CNY)/minute/piece	Unit price (CNY)/hour/piece	Duration of discount
Workarea	CPU Instance_4 Core_8GB Memory	0.021	1.26	Free for the first 72 hours of each month
Workarea	GPU Instance_ Deep learning development card_6 Core 32GB Memory x1 Card	0.081	4.86	——

Workarea	GPU Instance_K40_6 Core 32GB Memory x1 Card	0.104	6.24	——
Workarea	GPU Instance_V100_12 Core 50GB Memory x1 Card	0.185	11.10	——
Workarea	GPU Instance_V100_48 Core 200GB Memory x4 Card	0.736	44.16	——
Workarea	GPU Instance_P4_12 Core 32GB Memory x1 Card	0.11	6.06	Free for the first 72 hours of each month
Workarea	GPU Instance_P4_48 Core 128GB Memory x4 Card	0.376	22.56	——
Workarea	CDS high performance storage 5G	0.00	0.00	Free
Workarea	CDS high performance storage 40G	0.0003267	0.019602	——
Workarea	CDS high performance storage 100G	0.0008167	0.049002	——
Workarea	CDS high performance storage 200G	0.0016333	0.097998	——
Workarea	CDS high performance storage 500G	0.0040833	0.244998	——
Train	CPU Instance_8 Core_32GB Memory	0.061	3.66	Free for the first 72 hours of each month
Train	GPU Instance_ Deep learning development card_6 Core 32GB Memory x1 Card	0.081	4.86	——
Train	GPU Instance_K40_6 Core 32GB Memory x1 Card	0.104	6.24	——
Train	GPU Instance_K40_24 Core 128GB Memory x4 Card	0.438	26.28	——
Train	GPU Instance_V100_12 Core 50GB Memory x1 Card	0.185	11.10	——
Train	GPU Instance_V100_48 Core 200GB Memory x4 Card	0.736	44.16	——
Train	GPU Instance_P4_12 Core 32GB Memory x1 Card	0.11	6.06	Free for the first 72 hours of each month
Train	GPU Instance_P4_48 Core 128GB Memory x4 Card	0.375	22.5	——
Prediction	CPU Instance_4 Core_8GB Memory	0.021	1.26	Free for the first 72 hours of each month
Prediction	CPU Instance_8 Core_32GB Memory	0.061	3.66	Free for the first 72 hours of each month
Prediction	GPU Instance_P4_12 Core 32GB Memory x1 Card	0.11	6.06	Free for the first 72 hours of each month
Prediction	GPU Instance_P4_48 Core 128GB Memory x4 Card	0.375	22.5	——

The supported package and price of BML product purchase consumption package are as follows:

Module	Package Configuration	Hour	Unit (CNY)/package
Workarea	GPU Instance_P4_12-core 32GB RAM x 1 Card	20	73
		50	182
		100	364
		200	728
		300	1090
		500	1818
Module	Package Configuration	Hour	Unit (CNY)/package
workarea	GPU Instance_V100_12-core 50GB RAMx1 Card	20	133
		50	333
		100	666
		200	1332
		300	1998
		500	3330

Arrears Rules

Low Balance Reminders and Arrears

Reminder of Insufficient Balance: Whether your account balance (including voucher available) is sufficient to pay the bill in 3 days is judged according to your bill amount of the latest 3 days. If not sufficient, system sends the renewal reminder. Whether your account balance (including voucher available) is sufficient to pay the bill in next day is judged according to your bill amount of the latest day. If not sufficient, system sends the renewal reminder.

Handling with the arrear payment: Whether your account balance is sufficient to pay the BML bill is hourly checked up on the Beijing time. For example, whether your account balance is sufficient to pay the bill from 10:00 to 11:00 is checked up on the 11:00. If not sufficient, system confirms the payment arrear and sends the renewal reminder. Service is stopped immediately after arrear. System sends the service suspension notice in arrear. Note: The running workspace instance will be stopped after the service is stopped, and the data under the non-mount point will be released and cannot be saved. The running training job will be forced to stop, the job will fail, and the prediction endpoint in the service will be terminated. Task configuration information in the training module and prediction module consoles will be retained by default. Please recharge and pay in time.

Operation Guide

Overview

Welcome to use Baidu Machine Learning(BML). This article aims to explain the operation of BML's management console and help you quickly apply the machine learning model to your own business. BML consists of three core modules:

- Model Training: Two models training methods are provided. You can choose the appropriate model development method according to your needs.
    - Notebook: A fully managed interactive programming environment Jupyter Lab is built in to implement data processing and code debugging.
    - Job modeling: Support multiple deep machine learning frameworks, launch large-scale training jobs with one key, and maximize training efficiency and effect. Including four types of job: deep learning job, machine learning job, AutoDL job, AutoML job.
- Model warehouse: store and manage the trained models in order according to different model categories, properties,

classifications and versions.

- Prediction service: Quickly deploy the trained model as a highly available online service, flexibly select a variety of computing components to accelerate prediction execution, and complete model test iteration and service operation and maintenance management through A / B test, Beta test upgrade, service monitoring, etc.

Machine learning is a continuous cycle process. Model development-Model management-Release prediction services are used for production deployment. Then, you can combine more business data and retrain the model according to actual usage to improve prediction accuracy.

In addition to these three core modules, the navigation bar on the left side of the console has two modules:

- Item List: Display the item list under the user name. You can add, delete, modify and check items.
- Package: Purchase prepaid package.

Enable the BOS Service and Upload the Data

Activate Object Storage BOS Service

BML needs to connect BOS (Baidu Object Storage) data source, and the training results and logs are stored in the specified BOS address. Therefore, it needs to activate the BOS service and upload the data to the corresponding bucket. Learn more about Baidu Object Storage BOS.

Log in Baidu AI Cloud Management Console, select "storage and CDN - > Object storage BOS" for product service navigation, enter the "Activate object storage BOS" page, and click "Activate now".

Create Bucket

After entering the BOS management console, click "Create bucket" to create a bucket according to the prompts in the pop-up floating layer.

At present, BML only supports reading or writing buckets in the North China- Beijing region. When creating bucket, you need to select North China-Beijing (Bucket has region attribute and can only be located in one region. After Bucket name is created, the region to which it belongs cannot be modified).

The name of each Bucket is globally unique. You can use a prefix to ensure the uniqueness of the name. For example, the name of the organization in which the Bucket is located can be used as the prefix of the Bucket (the name cannot be changed after the bucket is created).

Upload and Download Data

- Upload a file

  Click "Upload a file" to open the local file and select dialog box.

  After selecting the file, select the "Standard" storage type, and then click "upload". You can select multiple files to upload at the same time.

- Download file

  Click the file to download. In the pop-up floating layer, click the download symbol on the right side of "File name".

Data Annotation

Entity-relationship Annotation

Create annotation task

Log into the management console, enter “ Baidu Machine Learning (BML) > Data Annotation”, and click the “Create annotation task” button above the list:

Enter the necessary information in the pop-up window for creating an annotation task:

Contents to be input include:

Name of the annotation task (required): Name of the annotation task is composed of English characters, numbers and underlines. It cannot start or end with underlines, and the length is 2-30 characters.
Annotation scenario (required): Select “entity-relationship” annotation.
Data storage path (required): Storage path of the file to be annotated.
Annotation task description (optional): It is used to introduce specific annotation rules to annotation personnel, and support doc, docx and pdf formats.
Entity type (optional): Entity type used for annotation, such as “person name” and “company name”. If it is blank, you can enter “Tag management” during annotation to add one.
Relationship type (optional): Relationship type used for annotation, such as “father-child” and “husband-wife”. If it is blank, you can enter “Tag management” during annotation to add one.
Review required or not (required): You can select yes or no. If the data do not have to be reviewed after annotation, you can select “No”; if the data have to be reviewed, you can select “Yes” for second confirmation of the annotated information.
After clicking OK, you can see that the annotation task is created successfully.

Upload annotation data

Click “Upload” on the annotation list page or “Upload data” on the annotation task details page to add data for the annotation task.

Support “Local upload” and “Select from BOS” for data upload.
Support “Single file” and “Compressed package” for uploading. When upload a single file, support the file types of txt, doc, docx and pdf. A maximum of 4 files can be uploaded at one time, and each file shall not exceed 2M.

You can also directly select a single file from BOS to upload:

Click OK to enter the annotation task details page. After the data are uploaded and processed, you can see the file.

Conduct data annotation

After entering the annotation task details page, you can click “Annotation” on the data overview page or click the “Manual annotation” tab to enter the annotation page:

The entity-relationship triples are annotated with sentence as the unit. The BML annotation system will automatically segment the sentences in the files uploaded by users, and the default separators are Chinese and English periods, question marks and exclamation marks. ？ ！）
Before annotation, you have to click tag management to add annotation tags.

For example, in this case, we add a “person name” tag in the entity tag and add tags of “uncle-nephew”, “father-child” and “husband-wife” in the “relationship tag”.

During annotation, select the content to be annotated first, and then select the corresponding tag in the pop-up window, for example:

In the annotation process, you have to respectively select entity 1, relationship and entity 2. After this, click “Submit” on the right so that the annotation can be effective.

After annotation, click the “Save and go to next file” button to finish annotation of this file.

Conduct data review

Click the “Result review” tab to review the annotation information which has been completed. If all annotations are correct, click the “Pass” button. If the annotations are incorrect, click the “Fail” button. For failed files, you can annotate them again on the manual annotation page.

View information about annotation tasks

Click “Annotation Task > Task Management” to check the information about annotation tasks, including task progress, annotation task description, upload history of data to be annotated and result export history.

Entity-attribute annotation

Create annotation task

Log into the management console, enter “Baidu Machine learning (BML) > Data Annotation”, and click the “Create annotation task” button above the list

Enter the necessary information in the pop-up window for creating an annotation task:

Contents to be input include:

Name of the annotation task (required): Name of the annotation task is composed of English characters, numbers and underlines. It cannot start or end with underlines, and the length is 2-30 characters.
Annotation scenario (required): Select the “Entity-attribute” annotation.

Data storage path (required): Storage path of the file to be annotated.

Annotation task description (optional): It is used to introduce specific annotation rules to annotation personnel, and support doc, docx and pdf formats.

Entity type (optional): Entity type used for annotation, such as “person name” and “company name”. If it is blank, you can enter “Tag management” during annotation to add one.

Attribute type (optional): Attribute type used for annotation, such as “career” and “nationality”. If it is blank, you can enter “Tag management” during annotation to add one.

Review required or not (required): You can select yes or no. If the data do not have to be reviewed after annotation, you can select “No”; if the data have to be reviewed, you can select “Yes” for second confirmation of the annotated information.

After clicking OK, you can see that the annotation task is created successfully.

Upload annotation data

Click “Upload” on the annotation list page or “Upload data” on the annotation task details page to add data for the annotation task.

Support “Local upload” and “Select from BOS” for data upload. Support “Single file” and “Compressed package” for uploading. When upload a single file, support the file types of txt, doc, docx and pdf. A maximum of 4 files can be uploaded at one time, and each file shall not exceed 2M.

You can also directly select a single file from BOS to upload:

Click OK to enter the annotation task details page. After the data are uploaded and processed, you can see the file.

Conduct data annotation

After entering the annotation task details page, you can click “Annotation” on the data overview page or click the “Manual annotation” tab to enter the annotation page:

The entity-attribute is annotated with sentence as the unit. The BML annotation system will automatically segment the sentences in the files uploaded by users, and the default separators are Chinese and English periods, question marks and exclamation marks. ？ ！）
Before annotation, you have to click tag management to add annotation tags.

For example, in this case, we add tags of “person name” and “apple” in the entity tag and add tags of “nationality”, “career”

and “color” in the “attribute tag”.

During annotation, select the content to be annotated first, and then select the corresponding tag in the pop-up window, for example:

In the annotation process, you have to respectively select entity and attribute. After this, click “Submit” on the right so that the annotation can be effective.

After annotation, click the “Save and go to next file” button to finish annotation of this file.

Conduct data review

Click the “Result review” tab to review the annotation information which has been completed. If all annotations are correct, click the “Pass” button. If the annotations are incorrect, click the “Fail” button. For failed files, you can annotate them again on the manual annotation page.

View information about annotation tasks

Click “Annotation task>Task management” to check the information about annotation tasks, including task progress, annotation task description, upload history of data to be annotated and result export history.

Data Set

The data set is a module to upload, manage and pre-process the data to be used in the modeling process, including user data and common data. The user data are the data you upload, and the common data are the common open-source data set provided by the platform. The user data and common data are continuously updated in the future.

The visual modeling requires the use of data sets. It means that if you want to make modeling by dragging and dropping the components, you should first upload the data in the data set. Currently, only the table formatted data are supported. The platform converts the data of csv\txt\tsv formats into parquet format, and meanwhile makes a simple preprocessing and save the data in your BOS. Then you can use the data in the visual modeling.

Note Currently, the fees are not charged for the data set module. But because the data files are saved in the BOS, the BOS fees are generated.

User Data

Data set list

The page of data set list displays the name, type, status, data volume (for the data list format, namely, the lines of data) creation time, update time and operation of the data set.

Create a data set task and upload data

Click "Create Data Set" button to pop up the window of "Create Data Set". Fill in the data set name, and the data storage path provides the default values. Of course, you can also select the BOS path, click "Confirm" to create the data set task.

Upload the data in BOS in advance. Here takes the open-source iris data set for example, the iris.csv file is previewed as below. The data has no header, and the column separator is the halfwidth comma:

Then click [Upload] button, and the page jumps to the [Upload Data] page. Fill in the upload data list configuration: Upload options (supplement refers to upload of new data or supplement of data of the same dimension, and replacement refers to replacement of data of different dimensions), upload modes (currently, only supporting uploading from BOS), upload path, column separator, whether there is a header, and coded format, as shown in the figure:

Click [Next] to conduct the data preprocessing configuration. Select the abnormal processing mode. Meanwhile, you can modify the column name or data format.

Details of data set

When the data set status becomes successful, click the data set name to enter the page of data set details. You can switch the tags to view the basic information, original data and statistical data.

The statistical data include the simple statistical results of data set, include the number of unique values, number of missing values, mean value, variance, standard deviation, etc. You can drag the slipper for viewing.

Public Data

Currently, the public data set presets the data sets iris and Boston Housing.

click the data set name to enter the page of data set name details. You view the basic information, original data, statistical data and user data.

Raw data:

Granularity data:

Notebook Modeling

Create Notebook Instance

BML provides an interactive code editing and running environment Jupyter Lab with built-in algorithmic frameworks such as TensorFlow, Keras, PyTorch, Caffe, Mxnet, Chainer, CNTK and PaddlePaddle.

Log in to BML Management Console, navigate to select "Model training-> Notebook", click "Create instance", fill in the name of the instance, select the calculation resource package (please see Product Pricing for billing method and package price), select the size of the startup disk, specify the BOS path(optional) to complete the creation (the creation process may take about 1 minute).

Warm Tip: The instance status will continue to incur expenses during "running". Please stop the instance in time when not in use. The instance will be stopped after arrears. If there is a program running, please ensure that the account balance is sufficient. After stopping the instance, all your changes (code, data, installed third-party software package, etc.) will be saved. Click "Run" next time to retrieve it, and then click Jupyter again to continue the previous operation.

Enable Jupyter

Click "Open jupyter" to enter the Jupyter development interface. The platform in the demo folder in the left column provides template code, which is used to introduce how to download data, conduct training, etc., for your learning and reference. Please refer to Appendix for Jupyter operation instructions.

Deep Learning Job

Deep learning job integrates various open source deep learning frameworks. Users can use different frameworks, write codes for multiple rounds of training and iteration, and upload the generated models and various data to BOS storage.

Create a Job

In training job, multiple resource packages and GPU resources of different models are provided to unify resource scheduling, thus improving training speed.

Select "Training > Deep learning job" in the left navigation bar to enter the deep learning job list page. Click "Create Job" to enter the new job process.

When creating a job, you need to submit the running code and complete the corresponding configuration.

To submit the running code, you can input the code in two ways:

1. Edit code directly: copy the debugged code directly to the code edit box to initiate a job.

2. Select code file: upload the code to BOS, fill in the code file path on BOS and initiate cluster job.

- When you select edit code directly, you can input the code directly into the code edit box.

  In addition, you can click "Select code template". Here we provide some code templates for your reference. Note, however, that the selected code template overwrites the code in the code editing area.

- Select "Select code file", select the BOS path of code storage to complete code input.

Job Configuration Item

Configuration name	Required	Description
Job name	Yes	It can only consist of numbers, letters or - and the first can only be a letter
Algorithm or framework	Yes	Support TensorFlow v1.13.1, Python V1.1.0 and PaddlePaddle v1.4.0
Whether to send SMS at the end of job	Yes	Whether to send SMS at the end of job
Output path	Yes	The path where the model output and logs are stored. Put the trained model and data into the output directory of the container, and the platform will automatically upload the contents of the output directory of the container to the path /{job_id}/output, and upload the logs to the path/{job_id}/log
Training data path	No	The platform will automatically download the data under this path to the local train_data directory under the container environment. If the job has multiple containers, each container will only be assigned to download part of the data
Testing data path	No	The platform will automatically download the data under this path to the local test_data directory under the container environment. If the job has multiple containers, each container is assigned to download only part of the data
Computing resources	Yes	BML cluster (or your private CCE cluster)
Resource package	Yes	It includes CPU instance _2 core _4GB memory, CPU instance _8 core _32GB memory, GPU instance_deep learning development card_6 core 40GB memory x1 card, GPU instance _K40_6 core 40GB memory x1 card, GPU instance_V100_6 core 40GB memory x1 card, etc
Number of instances	Yes	Multi-machine configuration
Maximum running time	Yes	After the job runs beyond the maximum running time, it will automatically terminate the job, which may result in no results being generated.

Job Management Related Operations

For jobs that have been submitted, you can do the following:

- Terminate: terminate a job that is currently running or queued. After termination of operation, the job results and job logs will not be uploaded to the specified BOS path.
- Clone: clone the code and configuration items of a job to enter the initiate job page.
- Delete: Delete this job. If the job is still in queue or running at the time of deletion, the queue or running will be terminated before deleting the job.
- View job details: click job name to enter job details, view job configuration information, job code, and job operation details.

- Job operation details: view the current job operation status and startup/end time.
- Resource information list: view the running status of the container used by the current job and the running log. In the running job, you can directly view the running log. For jobs that have finished running, a redirect bos address and a download link to store the running log will be provided for viewing or downloading the running log.
- View log analysis: when there is an error in job execution, you can view the log analysis of the error job here.

View Job Results

After the job runs, the training results and running logs will be stored in the corresponding BOS address according to the output result storage path specified during job configuration and the log storage path.

Go to BOS to view or download the running results of the job, and directly view or download the running log by using the redirect BOS address and download link provided to store the running log. In two cases, the job results and job logs cannot be saved:

1. Terminate the job manually;
2. The job is terminated automatically when it runs out of time.

Machine Learning Job

The machine learning job includes many efficient, and well-proven machine learning algorithms and open-source machine learning algorithms of RAPIDS-cuML GPU version, which are independently developed by Baidu. Among them, BML's efficient distributed computing capability facilitates the user to achieve their work objectives even if there are massive data. These algorithms apply to statistics and analysis of massive data, data mining, model training, business intelligence, and other fields. With the RAPIDS-cuML, the developer can run traditional ML jobs on the GPU without having to learn more about the CUDA programming details.

With the BML algorithm, the user's data must first run the "data standardization" algorithm. The main objective of the data standardization is to achieve the ID of the data samples and reduce the memory occupation in the training process. You can use different algorithms for access to the standardized training data, such as logistic regression dichotomy, and xgboost. That's to say, you can use the standardized data as the input training data for various algorithms. With the RAPIDS-cuML algorithm, you can call cuml and cudf libraries and submit training jobs by directly editing the codes or selecting the code files. You can run the job on a GPU machine. Currently, the BML only supports the single-machine and single-card operation.

Create a Job

On the left navigation bar, select "training--> machine learning jobs" to enter the list page of machine learning jobs. Click the "Create Job" button to enter the new job process.

Data Standardization

The main objective of the BML data standardization is to achieve the ID conversion of the data samples and reduce the memory allocation in the training process. After the user inputs the training data meeting the specified format, the platform counts the number of samples and features, converts the string feature into int numbers, and finally outputs the int numbers to the user's ID-converted data. The user can use this ID-converted data as input for various machine learning algorithms, such as LR, and xgboost. The data standardization is illustrated here so that you can use the standardized data as the input for various algorithms or multiple parameter adjustments. Thus, it is possible to reduce the tedious process of standardizing data for each training.

Configuration Descriptions:

Configuration Name	Required	Description
Job Name	Yes	It can only consist of numbers, letters,-or_ and can only start with letters, with a length of less than 40 characters.
Algorithm or framework	Yes	Data standardization
Send an SMS at the end of job	Yes	Send an SMS by default
Input data format	Yes	Options include sparse data without a weight value, sparse data with a weight value, and dense data. For details, see the following input data format description.
Input data type	Yes	Options include classification and regression. The label of the classification data is a discrete value, and the label of the regression data is a continuous value.
Input data path	Yes	Store the training data meeting the format. Supports entry of a single file or directory. If the input is the directory platform, the input data meeting the format requirements under the path gets standardized, including counting the number of samples and features, and converting the features of the string type to the number of the int types
Output path	Yes	Store the path of output data and log. After the job gets done successfully, store the standardized data in the path /{job_id}/data, and the log in the path/{job_id}/ log. You can use standardized data as the input data of other BML algorithms.
Computing resource	Yes	Currently, the BML only supports BML clusters.
Resource package	Yes	Currently, the BML only supports CPU instance_8 core _32GB memory.
Number of instances	Yes	2-4
Maximum running time	Yes	If the job runs for the maximum running time, the BML stops the job automatically, which may cause a job failure.

Descriptions for the input data format:

Format Type	Format ID	Sample Format	Descriptions
Sparse data without a weight value	SPARSE_ID	No,label,feature1,feature2,.....featureN	In the sample, the weight of the feature that appears is 1, and the weight of the feature that does not appear is 0. The number of features for each sample row may differ.
Sparse with weight value	SPARSE_ID_WEIGHT	No,label,feature1 weight1,feature2 weight2,.....featureN weightN	The weight of the feature in the sample is the corresponding weight value. The number of features in each sample row may differ, and there is a space between the feature and weight.
Dense data	DENSE	No,label,weight1,weight2,weight3.....weightN	According to requirements,the feature number of sample is 0,1,..., n-1, and the corresponding weight is weight0, weight1,... Weightn-1. The number of features in each sample row is N, which must be equal

The general limitations of the input data format are as follows:

- No is the number of the sample in each row. There is no general limitation. It may be null. Note: when No is null, the sample needs to start with a comma, e.g., label, feature_1, feature_2)
- The label is the annotation value of the sample in each row. It may be a discrete value and string. It may be null for unsupervised algorithms.
- featureN n is the specific feature label of each sample row, which may be a discrete value and string.
- weightN is the weight corresponding to the specific feature tag of each sample row, which may be a discrete value and a continuous value.

Notices:

- Each field in the user data cannot contain comma and space separating the field. If there are such two characters in the user's original data, you should escape them in advance.
- All rows that do not meet the format are ignorable, which significantly affects the effect of the model.
- When the data is in sparseID format, the content between commas is regarded as a string, and the space is free from the check. You should know and handle it in advance.

Example Configuration:

The training data is the SUSY data downloaded from the Internet. A comma has been added at the beginning of each row of the data. Taking sed-i s / ^ /, / g yourfile as an example, it is cut and stored on the public BOS. After downloading the data, you can divide it into training data, evaluation data, and prediction data and store them on your BOS for data standardization. You can also directly use our public BOS data for training.
Input data format: Dense data
Input data type: Classification
Input data path: bos:/bml-public/automl-demo/data/susy-train/
Th output data path is to configure your BOS path.

Click "OK" to submit the job.

Descriptions for output data format:

- Dataset.info is a data set information file, which records some statistical attributes of the data set, including the number of samples, number of features, and number of labels.

- Under the preprocess_out directory is the ID-converted data.
- Under the preprocess_dictionary is the ID-converted dictionary information and file information. In the feature dictionary featureIDMap file, the first row is the feature number feature_num, and the remaining feature_num rows are "feature ID original data feature string", separated by a space.
- Under the preprocess_summary directory are the statistics information of the features and label. Of which, feature_summary is the feature statistics file, in the format of the original feature_flag (means the filtered flag, with 0 filtered.) feature-id feature_count feature_weight_avg feature_weight_max feature_weight_min. The label_summary is a label statistical file in the format of label_flag label_id label_count.

Logistic regression dichotomy

The BML logistic regression dichotomy is a method to realize the dichotomy model for the standardized training data. The algorithm can output a dichotomy model when providing standardized training data, use the evaluation data to evaluate the model and calculate evaluation indexes when providing standardized evaluation data, and use the prediction data to make predictions and save the prediction results to the BOS when providing standardized prediction data.

Configuration descriptions:

Configuration Name	Required	Description
Job name	Yes	It can only consist of numbers, letters,-or _ and can only start with letters, with less than 40 characters in length
Algorithm or framework	Yes	Logistic Regression Dichotomy
Send an SMS at the end of the job	Yes	Text by default
L1 regularization coefficient	Yes	0 <= L1 <= 1 , floating-point number, scientific counting supported
L2 regularization coefficient	Yes	0 <= L2 <= 1 , floating-point number, scientific counting supported
Convergence condition	Yes	0 < termination <= 0.1, floating-point number, scientific counting supported
Maximum number of iterations	Yes	When the model training reaches the maximum number of iterations or meets the convergence condition, the training stops. The maximum value is within 20 <= maxIter <= 200, and is a positive integer.
Training data path	Yes	Store the training data path after data standardization. That's to say, the output data path in data standardization job details). The algorithm uses this training data to train the model.
Evaluation data path	No	Store the evaluation data path after data standardization. That's to say, the output data path in the details of the data standardization job. The algorithm uses this evaluation data to evaluate the model. It stores the evaluation results in the model output path. / {job_id}/evaluate. If not entered, output the model directly, without an evaluation result.

		model directly, without an evaluation result.
Prediction data path	No	Store the prediction data path after data standardization (i.e., the output data path in the details of data standardization job). The algorithm uses the model to predict the prediction data, and stores the prediction results in the model output path / {job_id}/predict. If not entered, no prediction gets done.
Output path	Yes	Store the model and log. After the job is successful, store the model in the path /{job_id}/model, and the log in the path /{job_id}/log.
Computing resource	Yes	Currently, only BML clusters are supported
Resource package	Yes	Currently, only CPU instance _8 core _32GB memory supported
number of instances	Yes	2-4
Maximum running time	Yes	If you run the job for the maximum running time, BML automatically stops the job, which may cause a job failure

Descriptions for training data format:

You need to standardize the input data of the logistic regression dichotomy algorithm, and then use the standardized data as the input data of the logistic regression dichotomy algorithm.

Example configuration:

The training data is the SUSY data downloaded from the Internet. A comma has been added at the beginning of each row of data. For example, sed-i s / ^ /, / g yourfile, the data is cut and stored on the public BOS. After downloading the data, you can divide it into training data, evaluation data, and prediction data and store them on your own BOS. You can also directly use our public BOS data for training. It is noteworthy that you can use the standardized data as the input data of the logistic regression dichotomy algorithm.
Training data path: bos:/bml-public/automl-demo/data/susy-train/
Evaluation data path: bos:/bml-public/automl-demo/data/susy-test/
Prediction data path: bos:/bml-public/automl-demo/data/susy-all/
First, standardize the data of the above three paths and submit three data standardization jobs. After the jobs are all successful, copy the **Data output path** on the job details page, which is used to enter the input data path of the logical regression dichotomy. For example, copy the path in the red box.

The configuration of a new job in logical regression dichotomy is as follows:

Click "OK" to submit the job.

Descriptions for output data format:

- The output model is mainly the weight parameters corresponding to each feature dimension in the lr model
- The output is in plain text format. Each row represents a feature dimension. There are three fields divided by space, which are the weight parameter of the feature, the internal ID of the feature in the parameter adjustment algorithm, and the original name of the feature.

KMeans clustering

The main objective of the BML KMeans clustering is to realize the clustering model for the standardized training data. It can train the clustering model when providing standardized training data. And, it can use the trained model and prediction data to output prediction results and save them to the BOS when providing standardized prediction data.

Configuration instructions:

Configuration name	Required?	Description
Job name	Yes	It can only consist of numbers, letters,-or_ and can only start with letters, with less than 40 characters in length.
Algorithm or framework	Yes	KMeans Clustering
Send an SMS at the end of the job	Yes	Whether to send an SMS to inform the user after the job gets finished.
Number of clusters	Yes	Means the number of clusters, which is a positive integer within the range of [2, 3000].
Maximum number of iterations	Yes	If the model training reaches the maximum number of iterations or meets the convergence condition, the training stops. It is a positive integer within the range of [6, 10000].
Convergence condition	Yes	If the change rate of the sum of the distance from each point to the center point of the cluster is less than the convergence condition for 5 consecutive rounds, the training gets stopped. It is within the range of [0, 0.5]. It supports scientific counting, and uses a floating-point number
Initialization method of cluster center	Yes	Only INITCLUSTER_RANDOM is supported, i.e., you can select the starting point.
Distance calculation method	Yes	DISTANCE_EUCLIDEAN（Euclidean distance） DISTANCE_SQUAREEUCLIDEAN（SEUCLID） DISTANCE_MANHATTAN（manhatton distance) DISTANC\E_COSINE（cosine distance） DISTANCE_TANIMOTO（jaccard distance）
Center point storage mode	Yes	True indicates sparse storage and false indicates dense storage. Among them, the sparse storage reduces the computing efficiency. Thus, when the feature dimension is not high, try to use the dense storage mode. If the dimension is too high, and you use the dense storage mode, the job may fail because the memory is too small. Thus, when the dimension is too high, try to use the sparse storage mode.
Output training set clustering results or not	Yes	If true, save the training data clustering results in the model output path. /{jobid}/cluster; if false, do not save the training data clustering results, but output the model only.
Training data path	Yes	Store the standardized training data path (i.e., the output data path for the data standardization). The algorithm uses this training data to train the clustering model.
Prediction data path	No	Store the standardized prediction data (i.e., the output data path in the details of data standardization job). The algorithm uses the model to predict the prediction data. It stores the prediction results in the model output path / {job_id}/predict. If not entered, no prediction gets done.
Output	Yes	Means the path to store the model and log. After the job is successful, store the model in the path

path		/{job_id}/model, and the log in the path /{job_id}/log
Computing resources	Yes	Currently, it only supports the BML clusters.
Resource package	Yes	Currently, it only supports the CPU instance _8 core _32GB memory.
number of instances	Yes	2-4
Maximum running time	Yes	If you run the job for the maximum running time, the BML automatically stops the job, which may cause a job failure

Descriptions for training data format:

It is necessary to standardize the input data of the KMeans clustering algorithm. You can use the standardized data as the input data of the KMeans clustering algorithm.

Example configuration:

The training data is the Iris Flower Data Set downloaded from the Internet. It is divided into training data and prediction data and stored in the public BOS. After downloading the data, you can divide it into training data and prediction data and store them on your own BOS. You can also directly use our public BOS data for training. It noteworthy that you can use the standardized data as the input data of the KMeans clustering algorithm. The standardized training and prediction data are as follows:
Training data path: bos:/bml-public/ml-demo/data/iris-train-standardized/
Prediction data path: bos:/bml-public/ml-demo/data/iris-predict-standardized/
KMeans clustering create job example configuration is as follows:

Click "OK" to submit the job.

Introductions for output data format:

- In the model file, the cluster_info is the center point vector, and distance_type is the model-related information, including distance formula and feature dimension, which are used for batch prediction.
- When setting "output training set clustering result or not" to true, you can save the clustering result of the training data in the model output path / {jobid} / cluster. The format of each row in the file is: original sample data; category number.

RAPIDS-cuML

The cuML is a library, which implements the machine learning algorithm in the data science ecosystem of RAPIDS. The cuML enables developers to run traditional ML jobs on the GPU without having to learn more about the details of CUDA programming. With the cuML library, the user can program and call various algorithms (such as KMeans, and xgboost) in the cuML to realize the machine learning.

Configuration description:

Configuration Name	Required?	Description
Job name	Yes	It can only consist of numbers, letters,-or_ and can only start with letters, with less than 40 characters in length.
Algorithm or framework	Yes	RAPIDS-cuML
Send an SMS at the end of the job	Yes	Whether to send an SMS to inform the users after the job gets finished.
Input code	Yes	If desired to edit the code directly, the user can select the code template, modify and write the code directly in the black box, which is applicable to the scenario having one training file. If desired to select the code file, the user can enter the file path and starting command on the BOS, which is applicable to the scenario having multiple training files.
python Version	Yes	Only python3 is supported in cuML
Output path	Yes	Means the path to store the model output and log. You should put the trained model and data in the output directory of the container. The platform automatically uploads the content in the output directory of the container to the path / {job _id} / output, and the log to the path / {job _id} / log
Training data path	No	The platform automatically downloads the data under this path to the train_data directory under the container environment. If the job has multiple containers, assign each container download part of the data only.
Test data path	No	The platform automatically downloads the data under this path to the test_data directory under the container environment. If a job has multiple containers, assign each container to download part of the data only.
Computing resource	Yes	Currently, only BML clusters are supported
Resource package	Yes	Deep learning and development card and other GPU single card packages
Number of instances	Yes	1
Maximum running time	Yes	If you run the job for the maximum running time, BML automatically stops the job, which may cause a job failure

Descriptions for training data format:

There are no restrictions on the format of training data. You can write a reader function to process the input training data.

Example configuration:

The training data of the cuML-xgboost algorithm is the Mortgage Data downloaded from the Internet, and the data downloaded for 2 years are put on the public BOS. You can download data, store it on your own BOS, or use the public BOS data for training directly. Training data path: bos:/bml-public/ml-demo/data/cuml-xgboost/. The output path is to configure your own

BOS path.

The example configuration of creating a job in RAPIDS-cuML is as follows:

Click "OK" to submit the job.

Descriptions for output data format:

- The output model format is the same as the sklearn.Use pickle.dump (model, open (filename, 'wb') to save the model to a file, and pickle.load (open (filename, 'rb) to import the cuML model. Or, you can use joblib.dump (model, filename) to save the model to a file, and joblib.load (filename) to import the cuML model.

Job list-related operations

- Terminate: Terminate the job that is currently running or in the queue. After the operation gets terminated, the system stops to upload the job results and job logs to the specified BOS path.
- Clone: If you select to clone the configuration item of a job, enter the "Create Job" page.
- Delete: This operation is to delete a job. If the job is still in the queue or running at the time of deletion, you need to terminate the queuing or running job, and then delete it. After deletion, the job disappears in the job list.
- View job details: Click the job name to enter the job details, view job information, parameter information, and cluster information.
- View running details: Click the job name, and select the running details tab to enter the running details page. Then, you can view the running status, start and end time, log details, and running curve.

View job results

After the job runs successfully, the model/data/log gets stored in the corresponding BOS address according to the output path specified during job configuration. The user needs to go to the BOS to view or download the job model, standardized data, or log.

Saving the job model/data/log is unsuccessful due to either of the following circumstances:

- Terminate the job manually
- The job gets automatically terminated due to running timeout
- The job gets failed to run

If the user's job gets failed in case of either of the following circumstances:

- The input data does not match the data format
- The BOS address of the input data does not exist or is inaccessible
- The bucket of the output log/model/data does not exist or is not inaccessible
- A training timeout takes place

AutoDL Job

AutoDL is an automatic deep learning product, which uses advanced transfer learning or neural network architecture search technology to provide data for business training. The platform provides a simple and easy-to-understand operation API. It only takes a few steps to train, evaluate and deploy the model, giving users a one-stop experience. AutoDL has a wide range of applicability. Beginners only need to submit data to obtain high-quality models, and experienced engineers can continue to study the high-quality models provided by the platform.

Create a Job

Select "Training-->AutoDLjob" in the left navigation bar to enter the AutoDLjob list page. Click the "Create job" button to enter the Create job process.

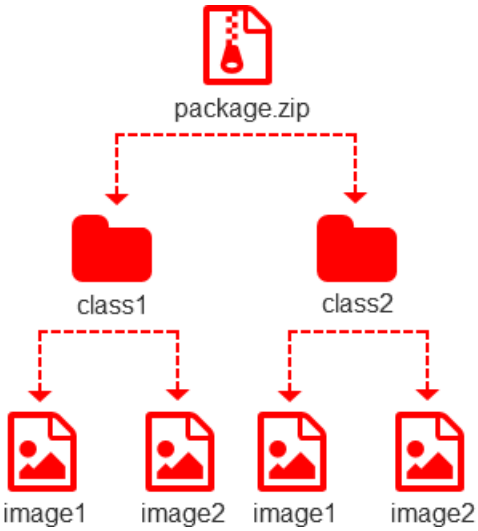
Image Sorting-Transfer Learning

Using trained convolutional neural network (CNN), combining with the training data to fine tune several layers, a model suitable for our training data is obtained, avoiding the long-term training process with large resources, which is the concept of transfer learning. Image sorting-Transfer learning, aiming at hundreds of tensor-level image data, an available model can be trained for users in 10 minutes, and the training process is displayed, including preprocessing results, loss and acc indexes in the training stage. Users can deploy and use this model directly in the prediction service.

Configuration Instructions:

Configuration name	Required	Description
Job name	Yes	It can only consist of numbers, letters,-or _ and can only start with letters, with less than 40 characters in length
Algorithm or framework	Yes	Image Sorting-Transfer Learning
Send SMS at the end of job	Yes	Whether to send SMS to inform users after job completion
Training data path	Yes	Path to store model training data. When filling in zip package, only use this zip package for training; when filling in the folder, use all zip packages under the folder for training. The folder does not support subdirectory iteration. Only the first 10 zip packages are taken from more than 10 zip packages. Currently, only zip packages are supported
Output path	Yes	The path to store the model and log. After the job succeeds, store the model in the path/{job_id}/model, and the log in the path/{job_id}/log
Computing resources	Yes	BML cluster or user self-owned CCE cluster
Resource package	Yes	Deep learning and development card and other GPU single card packages
Number of instances	Yes	1
Maximum running time	Yes	If the job runs for the maximum running time, BML will automatically force the job to stop, which may cause job failure

Notes for Training Data Format:



- All prepared photos need to be sorted into a single folder, and all folders need to be compressed into **.zip format** .
- If there are many photos, it is recommended to divide them into multiple packages, and the maximum support for training is 10 packages.
- If the sorting and naming of multiple packages are consistent, the system will automatically merge the data as a kind of photo.
- The sorting should be named in the form of numbers, letters, and underscores. At present, the Chinese format is not supported. At the same time, please note that there should be no spaces
- Photo description: (1) The expanded-name of photos can support three common types: Jjpg, jpeg, png (2) Photo size: temporarily unlimited (3) number of photos in each category: 20 <= number of photos in each category; Note: the number of photos in each category should be balanced to achieve better model effect (4) number of photo types: 2 categories < = number of photo sortings < = 200 categories (5) total number of photos: total number of photos < = 100,000

Example Configuration:

Training data is cifar10 data downloaded from the Internet. The developer converts the data into an input data format conforming to Image sorting-Transfer learning algorithm (the converted code can refer to transfer_cifar10.py, run in python2 environment and rely on numpy and opencv libraries), i.e. the photos are divided into different directories. As an example, the developer will take 100 photos for each category, totaling 1,000 photos for training. You can download the data and convert it yourself, or you can directly use our public BOS data for training.

Training data path: bos:/bml-public/autodl-demo/data/cifar10-for-transferlearning.zip

Output path configure your own BOS path.

Click "OK" to submit the job.

Notes for Model Output Format:

- The output model is a model in pytorch format, ending with pth suffix, and containing the weight information of the network. The model file is stored in the output path/{jobid}/model specified by the user, which can be used for prediction service.

Activate Prediction Service:

After Image sorting -Transfer learning job runs successfully, copy the model output path on the job details page, such as: bos:/xxx/yyy/autodl-qianyixuexi/job-8c0yiasq02caa6hf/model/

- On the Prediction > Prediction model library > Create model page, add the Image sorting -Transfer learning model, and fill in the path above in the model file path.

- On the Prediction > Template configuration library > Create template page, create a template configuration, such as:
- In the Prediction - > Endpoint management - > Create endpoint page, load the above template and start the prediction service, such as:

Send Prediction Request:

When the prediction endpoint status is in service, the prediction service request can be sent with the following code:

```
 #!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import json
import base64
import requests

IMAGE_PATH = "aaa.jpg"
ENDPOINT_URL = "http://10.181.114.16:8023/v1/endpoints/yyyy/invocations"
PARAMS = "?interface=predict&action=predict"
TARGET_URL = ENDPOINT_URL + PARAMS

def get_request():
    """Construction Request"""
    arr_instances = []
    with open(IMAGE_PATH, "rb") as f:
        data = f.read()
    encoded_data = base64.b64encode(data)
    str_data = str(encoded_data)
    obj_instance = {
        "data": str_data,
    }
    arr_instances.append(obj_instance)
    request = {
        "instances": arr_instances,
    }
    return request

if __name__ == '__main__':
    request = get_request()
    json_request = json.dumps(request)

    headers = {'Content-type': 'application/json'}
    res = requests.post(TARGET_URL, data = json_request, headers = headers)
    result = res.text
    print(result)
```

Modify IMAGE_PATH as the photo position and ENDPOINT_URL as the endpoint URL, and execute the above code to send a prediction request and obtain a photo sorting prediction result. Predicted photo:

Predicted results:

```
 {"result":"[[1.288892149925232, -3.755728244781494, -1.8610944747924805, 6.285534381866455, -
1.8683143854141235, 2.2759482860565186, -3.119175672531128, 0.8390857577323914,
1.8015419244766235, -1.3871209621429443]]"}
```

The results showed that the photo was predicted to be label 3.

Image Sorting - ENAs

Using the advanced Neural Network Architecture Search Technology, training is conducted on the data provided by the service,

and the optimal model search and high-quality model output can be completed in a few hours. According to the classified image data given by the user, a trained model is output to the user, and the training process is displayed, including the preprocessing result, Metric index of each round, neural network structure diagram and tuning result.

Configuration Instructions:

Configuration name	Required	Description
Job name	Yes	It can only consist of numbers, letters,-or _ and can only start with letters, with less than 40 characters in length
Algorithm or framework	Yes	Image Sorting - ENAs
Send SMS at the end of job	Yes	Whether to send SMS to inform users after job completion
Training data path	Yes	Path to store model training data. When filling in zip package, only use this zip package for training; when filling in the folder, use all zip packages under the folder for training. The folder does not support subdirectory iteration. Only the first 10 zip packages are taken from more than 10 zip packages. Currently, only zip packages are supported
Output path	Yes	The path to store the model and log. After the job succeeds, store the model in the path/{job_id}/model, and the log in the path /{job_id}/log
Computing resources	Yes	BML cluster or user self-owned cce cluster
Resource package	Yes	Deep learning and development card and other GPU single card packages
Number of instances	Yes	1
Maximum running time	Yes	If the job runs for the maximum running time, BML will automatically force the job to stop, which may cause job failure

Notes for Training Data Format:

- All prepared photos need to be sorted into a single folder, and all folders need to be compressed into *.zip format *.
- If there are many photos, it is recommended to divide them into multiple packages, and the maximum support for training is 10 packages.
- If the sorting and naming of multiple packages are consistent, the system will automatically merge the data as a kind of photo.

- The sorting should be named in the form of numbers, letters, and underscores. At present, the Chinese format is not supported. At the same time, please note that there should be no spaces

- Photo description: (1) The expanded-name of photos can support three common types: jpg, jpeg, png
  (2) Photo size: 100k < = photo size < = 3M, length-width ratio is within 3:1, with the longest side less than 4096px, the shortest side greater than 30px (3) number of photos per category: 50 <= number of photos per category; Note: The number of photos in each category should be balanced to achieve better model effect (4) Number of photo categories: Category 2 < = number of photos classified < = 200 categories (5) total number of photos: total number of photos < = 100,000

Example Configuration:

Training data is cifar10 data downloaded from the Internet. The developers convert the data into the input data format that conforms to the image sorting - ENAs algorithm (the converted code can refer to the transfer – cifar10.py, which runs in the python2 environment and relies on the numpy and opencv libraries), i.e. classifying the images according to different directories and making them into zip packages. You can download the data and convert it yourself, or you can directly use our public BOS data for training.

Training data path: bos:/bml-public/autodl-demo/data/cifar10.zip

Output path configure your own BOS path.

Click "OK" to submit the job.

Notes for Model Output Format:

- The output model is a model in keras HDF5 format, including network structure and weight parameters.

Job List Related Operations

- Terminate: terminate the job that is currently running or queued, no longer queued, no longer running. After termination of operation, the job results and job logs will not be uploaded to the specified BOS path.

- Clone: clone the configuration item of a job to enter create job page.

- Delete: delete the job. If the job is still queued or running at the time of deletion, the queue or running will be terminated first, and then the job will be deleted. After deletion, the job will disappear from the job list.

- View job details: click job name to enter job details, view job information, parameter information and cluster information.

- View operation details: click job name and select operation details tab to enter operation details and view operation status, start and end time, log details, operation curve, etc.

View Job Results

After the job runs successfully, the model and log will be stored in the corresponding BOS address according to the output path specified during job configuration. Users need to go to BOS to view or download the job model or log.

If the job model or log cannot be saved, it may be the following:

- Terminate job manually

- Job runs timeout and is automatically terminated

- Job failed to run

User job failed, possibly due to the following conditions:

- Input data does not match data format requirements

- The BOS address of the input data does not exist or is not accessible

- Bucket of output path does not exist or is not accessible
- Training timeout

AutoML Job

UtoML can simplify the complicated process of algorithm modeling and model parameter debugging, automatically carry out super-parameter learning, and then automatically build high-precision models, saving labor and lowering the threshold of machine learning.

Create a Job

Select "Training-->AutoML job" in the left navigation bar to enter the AutoML job list page. Click the "Create job" button to enter the create job process.

Logistic Regression

Logical regression automatically selects the optimal model training hyper-parameters through the debugging parameter training data and debugging parameter test data given by the user, and outputs a trained model to the user in cooperation with the model training data.

In a debugging parameter experiment, the debugging parameter algorithm will select a group of parameters according to the user specified parameter range and platform debugging parameter algorithm rules. Based on the hyper-parameters and the training/test data set set by the user for debugging parameter, the debugging parameter algorithm completes a model training and evaluation. Debugging parameter algorithm carries out many such debugging parameter experiments, and finally select the hyper-parameters of one experiment with the best effect to train the model.

The number of label types of training data for parameter debugging training/training test/final model must be 2. Currently, automatic hyper-parametric logistic regression only supports dichotomy. If the data is multi-classified, the job may fail.

Configuration Instructions:

Configuration name	**Required**	**Description**
Job name	Yes	It can only consist of numbers, letters,-or _ and can only start with letters, with less than 40 characters in length
Algorithm or framework	Yes	Select Logistic Regression
Send SMS at the end of job	Yes	Text by default
L1 regularization coefficient range	Yes	Floating-point numbers greater than 0 and less than 1, support scientific counting
L2 regularization	Yes	Floating-point numbers greater than 0 and less than 1, support scientific counting

coefficient range		
Number of iterations in a single test	Yes	A positive integer from 10 to 200. In each test, the number of iterations of the algorithm will be selected within this range. In the test, all the training data participating in the parameter debugging once is called a round, also called an epoch
Test times	Yes	A positive integer of 10 to 100 is used for a total of "test times" of tests. Each test selects a group of hyper parameters and combines them with the debugging parameter training data to obtain a model, and uses the debugging parameter test data to evaluate the advantages and disadvantages. After all tests are completed, the platform will select the optimal hyper-parameters, and then output the final model combined with the model training data
Input data format	Yes	Options include: sparse without weight value, sparse with weight value, and dense data. See the algorithm format requirements on the page for details
Training data path for debugging parameter	Yes	Store the training data of debugging parameter, use the training data in each test and conduct model training with a group of hyper-parameters
Parameter test data path	Yes	Store the test data of debugging parameter, use the test data in each test and evaluate the model in combination with the debugging parameter model
Model training data path	Yes	Store the training data of the model, AutoLR selects a set of super parameters with the best evaluation results from all tests, and outputs the final model in combination with the training data
Output path	Yes	The path to store the model and log. After the job succeeds, store the model in the path /{job_id}/model, and the log in the path /{job_id}/log
Computing resources	Yes	Currently, only BML clusters are supported
Resource package	Yes	Currently, only CPU instance_8 core_32GB memory is supported
Number of instances	Yes	2-4
Maximum running time	Yes	If the job runs for the maximum running time, BML will automatically force the job to stop, which may cause job failure

Example configuration:

Training data is the SUSY data downloaded from the Internet. Comma, sed -i s/^/,/g yourfile, has been added at the beginning of each row of data, which is divided and stored on the public BOS. After downloading the data, you can divide and store the debugging parameter training data/debugging parameter test/model training data on your bos, or you can directly

use our public bos data for training. Input data format: Dense data Debugging parameter training data path: bos:/bml-public/automl-demo/data/susy-train/ Parameter testing data path: bos:/bml-public/automl-demo/data/susy-test/ Model training data path: bos:/bml-public/automl-demo/data/susy-all/ Model output path and log output path to configure your own bos path.

Click "OK" to submit the job.

Notes for Model Output Format:

- The output model is mainly the weight parameters corresponding to each feature dimension in the Logistic Regression model
- The output is in plain text format, each row represents a feature dimension, and a total of three fields are divided by spaces,namely the weight parameter of the feature, the internal ID of the feature in the debugging parameter algorithm, and the original name of the feature.
- Only features whose weight parameter is not 0 are output

Job List Related Operation

- Terminate: terminate the job that is currently running or queued, no longer queued, no longer running. After termination of operation, the job results and job logs will not be uploaded to the specified BOS path.
- Clone: clone the configuration item of a job to enter create job page.
- Delete: delete the job. If the job is still queued or running at the time of deletion, the queue or running will be terminated first, and then the job will be deleted. After deletion, the job will disappear from the job list.
- View job details: click job name to enter job details, view job information, parameter information and cluster information.
- View operation details: click job name and select operation details tab to enter operation details and view operation status, start and end time, log details, operation curve, etc.

View Job Results

After the job runs successfully, the model and debugging parameter log will be stored in the corresponding BOS address according to the model output path and log output path specified during job configuration. Users need to go to BOS to view or download the job model and log.

If the job model and job log cannot be saved, the following conditions may occur:

- Terminate job manually
- Job runs timeout and is automatically terminated
- Job failed to run

User job failed, possibly due to the following conditions:

- Training data of debugging parameter training/debugging parameter training test/final model does not match the data format.
- BOS address of training data of debugging parameter training/debugging parameter training test/final model does not exist or is not accessible
- Bucket for output log / model does not exist or is not accessible
- Training timeout

Visual Modeling

The visual modeling is to connect the modeling process, configure the parameters and train the models by dragging and dropping the components.

Create Visual Modeling Task

Select [Visual Modeling] from the navigation bar [Model Training], and click "Create" button.

In the pop-up new task, fill in the experiment name (required) and experiment description (optional). Click "Confirm" to create a task.

Copy Visual Modeling Task

Click the [Copy] button in the operation column in the list to copy tasks, as shown in the figure:

Description of Visual Modeling Canvas

Click the task name to enter the visual modeling canvas. There are a lot of different types of components on the left side of the canvas. You can freely explore, drag and drop the components into the canvas. connect them back and forth to complete a complete modeling process.

The right side shows the configurations to be filled for each component. The configurations may include the parameter setting, field setting and resource setting according to different components.

Click the button on the top of the canvas to start or cancel training, and view the historical record.

The historical record displays the historical process of running, including the running time and running status of historical versions. Meanwhile, we can view the historical versions to make comparison of the versions.

The row of buttons below can achieve zoom-in and zoom-out of the canvas, display of actual size, full screen, box selection and opening of thumbnail.

Right click each component to copy and delete components, start execution here, execute here, execute the node, test run a small data volume and view data.

The loading symbol at the right side of the component indicates that the component is running, and the green check mark at the right side indicates that the component running ends.

The buttons at the upper right corner are respectively "Copy Task", Clear Data", "Save" and "Release".

Please note that if you exit the canvas without saving, the operations in the canvas are not recorded.

Click "Data Clearing" to make relevant settings:

After the data are cleared, the middle result data and logs in the running history are cleared.

If one algorithm component and one prediction component run successfully in the canvas, the "Release" button becomes the clickable status. You can release the well-trained models to the model repository by one key for unified management. After clicking "Release", you can fill in the relevant information in the pop-up window.

After the model is released successfully, you can find the model in "Model Repository>User Model".

Smart Vision

Summary

The BML intelligent vision module provides common image model training solutions. Major scenarios include image classification and object detection.

Create an Image Classification Model

Create an experiment

Click the “Create” button on the intelligent vision list page, enter the experiment name in the pop-up window.

After creation, you can view the creation experiment in the page of intelligent vision experiment list. Click "Experiment Name" or "View" to enter the page of experiment details:

The experiment process has been provided on the interface, and default parameters are available for the component. Usually, training can be completed only by changing data sets.

Modeling process

During the experiment, each component is equipped with the features of "Start Execution Here", "Execute Here" and "Execute the Node" to facilitate debugging.

1.Data selection: It supports selection of all data sets or filtration by multipart.
2.Data segmentation: It supports customization of segment percent of the training set and verification set and customization of the test set source.
3.Statistical node: It is mainly used for statistics of classified data size of the training set/test set/verification set which are used for training.
4.Image classification training: It supports adjustment of algorithm parameters and selection of basic models. For image classification training, only support the basic models of ResNet50, InceptionV4 and DPN131 at present.

Right click “Image classification training” to view the training process of real-time index monitoring model, for example:

5.Model evaluation: Right click “View an evaluation report” to check indexes such as the accuracy of outputting models during model training. Click “View a prediction result” to check the specific performance of each model on the test set, for example:

6.Filtration of prediction results: Filter the prediction results by confidence level, etc.:

7.Model selection: For multiple models generated by the training node, select the optimal model for subsequent use through the model selection component. The selection methods include: Automatic selection (by specifying the highest index in the test set or verification set) or customization. Model name needs to be manually filled in during customization.

8.Model conversion: It is used to convert the checkpoint generated during the modeling process into a model that can be used for prediction. You do not have to define parameters in this step.

9.Start and stop online prediction service: It is used to test whether the model can start the prediction service normally.

Create an Object Detection Model

Create an experiment

Click the “Create” button on the intelligent vision list page, enter the experiment name in the pop-up window, and select “Object detection” for the production line.

After creation, you can view the creation experiment in the page of intelligent vision experiment list. Click "Experiment Name" or "View" to enter the page of experiment details:

The experiment process has been provided on the interface, and default parameters are available for the component. Usually, training can be completed only by changing data sets.

Modeling process

The modeling process is similar to that of image classification, in which:

1. Data cleaning node: It supports to filter out untagged data during training.
2. Object detection training node: It supports adjustment of training parameters and automatic adjustment of batch_size and lr. Only FasterRCNN, RetinaNet and YOLOV3 are supported for the detection algorithm at present.

3. Model evaluation node: It supports to view the mAP index in the detection model.

Release the Model to the Model Warehouse

For experiments that have run successfully and with which data are not cleaned, it supports to release the model to the model warehouse. After release the model, further enable the online prediction service for use.

Model Warehouse

Model warehouse, as the name implies, is a warehouse used to manage models, including importing models, viewing, adding versions, retrieving, deleting, etc. To deploy a trained model as a prediction service, it is necessary to first publish the model to a model warehouse and then publish it as a prediction service.

The model repository contains user model and common model. User models are generated by users on the platform by modeling or imported from the BOS. Common models are several sample models provided by the BML platform, which will be updated continuously later.

User Model

Select "Model warehouse" - > "User model" in the navigation column, and the page displays the list of related models created by the user.The list contains the model name, the number of versions of the model, and labels.

You can create a prediction model as follows:

1. Click "Import model" on the page
2. Select and fill in the configuration items in the pop-up window, including selecting import method (select "Import New Model" for importing a new model), filling in the model name, model version, selecting/creating model tag, selecting model type and relevant information of model file in linkage, model path and description, and then click "OK" to complete the model import.

User can also add a new version of the model based on the existing model, as follows:

1. Click "Import model" on the page
2. Select and fill in the configuration items in the pop-up window, select the import mode "import as new version of existing model", complete other configuration items and click "OK". Currently supported model types include: Deep learning type (Tensorflow-v1.13.1, paddle-fluid-v1.5.0, Pytorch-v1.1.0, Caffe2, ONNX), machine learning type (Sklearn-v0.20, GBDT-v0.82, R-v3.5.2, Pyspark-v2.4.3), General-purpose type (PMML, custom), and built-in type (transfer learning - image classification)

After importing the model, return to the model list page. Click a model name on the model list page to go to the model version list page, where you can view multiple versions of the model.

Page Operation Description:

Page [Delete] : Delete all versions of the model.

Page [Refresh]: After creating a new version, you can refresh the page to view the newly created version.

List [Delete]: Delete a specific version of the model.

List [Create Online Prediction]: Click it and fill in the configuration item in the pop-up window. After confirming, create the online prediction service with this version model.

Click a version number to enter the details page of this version, which includes model information and related prediction services. Model information includes model type, related information when creating the model, creation time, update time, model source, model tag, model path and model description. The related prediction services part displays the list of online prediction services created by this version model.

Page Operation Description:

[Create Online Prediction]: Click it and fill in the configuration item in the pop-up window. After confirming, create the online prediction service with this version model.

[Remove]: Delete this version of model

[Refresh] button: Refresh page

Edit [Model Tag]: Click the Edit button to change the tag of the model.

Edit [Model Description]: Click the Edit button to change the description of the model.

Common Module

Select "Model Repository" - > "Common Model" in the navigation bar, and the page will display the list of built-in models of the platform. Here, several sample models are included. Novice users can directly use appropriate models to establish prediction services.

Prediction Service

The prediction service aims to run the user "model" by Web Server, and provide the predicting features. Currently, only the Https access mode is supported. The system assigns an access address to the prediction service running successfully. The user can access the prediction service by the "access address".

The premise of creating the prediction service is to import the well-trained model to the model repository. For the specific mode, please refer to the part of model repository.

Create Online Prediction Service

After the model is trained and released to the model repository, you can deploy the online prediction service by the online prediction service functional module of the BML platform.

In the "Prediction Service"-> "Online Prediction" page in the navigation bar, click "Create Online Prediction". In the page of pop-up box, select "Create New Service", fill in related parameters and click "Confirm".

Parameter description:

Creation mode: You can select to create a new prediction service or add a new version on the original prediction service.

Service name: Name of prediction service

Service version: Version No. of prediction service

Model: If you select "Yes", it means that the model in the model repository is selected to create the prediction service. If you select "No", it means that the custom image is selected to create the prediction service. Currently, the model in the model repository is used by default to create the prediction service. The system automatically matches the service type according to the used model.

Proportion: The platform distributes the traffic as per the proportion, and is used for multiple versions of AB Test. If there is only one version of the prediction, the traffic is 100% regardless of the proportion; if there are multiple versions, the traffic is distributed as per the proportion. For example, if the proportions of the two versions are 3 and 7 respectively, 30% and 70% of the traffic is distributed to the two versions respectively as per the proportion of 3:7.

Number of running duplicates: The prediction service can include multiple duplicates, and each copy is equivalent to ensure the high availability of the service. The fees charged increases proportionally as the number of duplicates increases.

If you want to add a version on one prediction service, you can click [Add Version] in the operation.

Then you can open the pop-up window of [Add Online Prediction Version], and fill in a new four-bit version number. Select a

new model, set the proportion, computing resource and duplicates, and click "Confirm" to add a new version.

Prediction Service List

The information of the prediction service list is composed by the prediction services submitted historically, and displays the basic information of the services.

As shown in the figure, the following operations are supported in the page of prediction service list:

Add version: Add a new version based on one service, namely, the created page of creation is the new version of the existing service.

View the service quality: View the PV, total handling time, total traffic, proportion of successful assess of the service, and other service quality information.

Stop: The prediction service is stopped. After the user doesn't use the related service any more, this operation is available. After the prediction service is not, fees are not charged.

Restart: Re-create the stopped services or the services failing to be created, and recover the corresponding service to the available status.

Testing: Enter the data to be predicted by the interface. Test whether the corresponding prediction results meet the expectations.

Deletion: Delete the corresponding prediction service. The deleted prediction service cannot be recovered. The corresponding record is also deleted and can't be queried. The service can be deleted only when it is under the status of "Stopped", "Creation Failed", etc.

Online Debugging

After the the service is created successfully, you can click『Test』to debug the prediction service.

Details of Prediction Service

After creating the prediction service, you can click the service name to enter the service details. The tags are divided into two switchable tags: basic information and service quality.

The service quality displays : PV (prediction requests), total handling time, total traffic, and the proportion of successful assess. The middle broken line graph/bar graph includes: PV, total traffic (sumTraffic), average traffic of each request (avgTraffic), total handling time (sumHandleTime), average handling time of each request (avgHandleTime) at each time point. The middle pie chart includes: proportions of status codes 5xx(status5Proportion), 4xx (status4Proportion), 3xx(status3Proportion) and 2xx(status2Proportion) at each time point. Support the query of corresponding data according to time.

The basic information includes the log path, access address and version list.

The version list displays the details of multiple versions: Status, model file, resource package, traffic proportion, service type, expectation and number of actual running instances, operation.

If the status is abnormal, it means that the actual number of running duplicates is less than expected number of running duplicates. Fees are charged for the duplicates in the running or abnormal status.

You can click "Edit" to modify the prediction service, including changes of model, traffic proportions of all versions, resource package and number of running duplicates, etc. If you need a high availability, you can properly increase the number of duplicates. If a lot of computation is required, you can change to a package with a higher availability. If you want to adjust the AB Test traffic proportion, you can make the adjustment by modifying the proportion.

If you need to roll back, you can click "Configuration History" to expand the pop-up window of configuration history; select the configuration of a certain historical time and click "Recover" to roll back.

Version Details

In the version list, click the version No. of one service version the expand the details of the version service. The details include the basic information and duplicate information of the service.

Duplicate Details

Click the duplicate ID of the page of version details to enter the page of copy details to view the duplicate information and the container list.

Container Details

Click the container name to display the container details in the page of browser new label. The container details mainly include the container log, stdout and stderr output in the running process of the container. You can turn the pages by the buttons in the page.

Project Management

The project is a logical set of Notebook, job modeling, model warehouse and prediction service under the user name, and is a workspace. After the project is deleted, all data, models and prediction services belonging to the project will be deleted. Please be careful.

List of Items

Click "Project List" in the left navigation bar to enter the project list page, where all the projects under the user name are displayed. Each BML user has a default item, which cannot be deleted.

The project list shows the number of tasks in running and the total number of tasks in the three modules of Notebook modeling, job modeling and prediction service under each project, which is convenient for you to know what services are charged, and then enter the project to close the instance and stop charging.

Create Project

Click "Create Project", fill in the project name, and click OK to create a project. Project name cannot be duplicated.

You can see that the created project also appears in the project list.

Click on the name of the project to enter the Notebook page of the project by default.

Switch Projects

If you want to switch projects, you can click the drop-down box under the current project to select.

Appendix

Instructions on Jupyter

For the official introduction and usage of Jupyter, please see:

https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

FAQs

FAQs

- Technical Support
- What is BOS? Why is BOS enabled when using BML?

- Where can I view running logs of training jobs?
- Does it support the installation of third-party packages? How to install?
- [When installing a third-party package in the workarea, the error “Could not find a version that satisfies the requirement xxx （from versions:） No matching distribution found for xxx”, How to solve it?](#when-installing-a-third-party-package-in-the-workarea-the-error-could-not-find-a-version-that-satisfies-the-requirement-xxx-from-versions-） no-matching-distribution-found-for-xxx--is-reported-how-to-solve-it#）
- How to convert the debugged code in Notebook into. py file and submit training job?
- Why is the training/prediction data of deep learning job not downloaded and the output data not saved?
- How to deal with the error of "invalid region, please check and try again"?
- Does it support subaccount?
- How to download the data from the non-mount bos directory to notebook?

Technical Support

For any problems you encounter in the use process, you are welcome to use QQ to scan the following QR code to join the technical exchange group for consultation and discussion.

Tips: the platform provides a lot of tips on the page. Don't forget to click the small question mark on the right to view related help.

What is BOS? Why is BOS enabled when using BML?

BOS (Baidu Object Storage) provides stable, secure, efficient and highly extended storage services, and supports any type of data storage, such as text, multimedia and binary, with a maximum of 5TB per file. In order to facilitate processing data and codes more securely and efficiently in the cloud, BML has achieved a deep connection with BOS, so you need to open BOS and create "Bucket" before using BML.

Where can I view the running logs of training jobs?

When the training operation status is running, enter "Operation details" and click "Real-time log details" to view the real-time log.

After the job succeeds or fails, enter "Operation details", click "Log details" to enter your "Bucket" Management page, enter the trainer/folder, and trainer.log is the log file of the training job. Currently, BOS does not support preview of files in this format. You need to click the download symbol to the right of "File Name: trainer.log』" and view it after downloading.

Does it support the installation of third-party packages? How to install?

At present, the CPU / GPU instances of workarea and training jobs support users to install third-party packages. We have configured common third-party packages for you, such as “numpy”, “sklearn”, “torch”, etc. You can use pip list to view the currently installed third-party packages and versions, and use pip install to install other third-party packages you need. (you need to distinguish the python version when installing package, pip for python2 and pip3 for python3.)

In the workarea, first, you can use the command `pip install xxx` to install directly in terminal; second, you can use the command `!pip install xxx` to install in notebook.

Training job is different from workarea, and the system command installation needs to be called in the training script. The specific installation method is shown in the following figure.

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  import os
4  os.system('pip install jieba')
5
6  import jieba
7
8  print(jieba.__version__)
```

When installing a third-party package in the workarea, the error "Could not find a version that satisfies the requirement XXX (from versions: ） No matching distribution found for XXX "? Is reported, how to solve it?

The default “pip” source of the workarea is Tsinghua image source (https://pypi.tuna.tsinghua.edu.cn/simple). If the error "could not find a version that satisfies the requirement XXX (from versions: ） No matching distribution found for xxx" is reported, indicating that the software package is not in Tsinghua image source, please use Python's official source image (https://pypi.python.org/pypi).

Taking “kaggle” as an example, you can use `!pip install kaggle -i https://pypi.python.org/pypi` to install the software package.

How to convert the debugged code in Notebook into. Py file and submit training job?

The code file in Notebook is in. ipynb format. If you want to submit a training job, you need to convert it to . py file. You can use `jupyter nbconvert --to python xxx.ipynb` in terminal to convert. ipynb files to. py files with the same name. As shown in the figure below, first input `ll` all the files listed under /mnt/demo path, then convert tensorflow-mnist.ipynb file to tensorflow-mnist.py file through `jupyter nbconvert --to python tensorflow-mnist.ipynb`, and finally input `ll` the validation file and save it successfully.

During conversion, non-code cells will be commented out, and jupyter unique statement that does not conform to python specifications may appear, which need to be corrected manually.

In addition, it should be noted that the input and output paths in the code should be appropriately modified: the training data path should be changed to "./train_data/", the prediction data path should be changed to "./test_data/", and the model and data of training output should be changed to "./output/". See FAQ-Why is the training / prediction data of deep learning job not downloaded and the output data saved?

Why is the training / prediction data of deep learning job not downloaded and the output data not saved?

The following figure is a screenshot of the parameter configuration of deep learning job. You need to fill in the output path (required), training data path (optional) and prediction data path (optional).

The platform will automatically download the BOS data under the training data path to the train_data directory in the container, download the BOS data under the test data path to the test_data directory in the container, and upload the contents under the output directory in the container to the output path during code running.

Therefore, it should be noted that the relative path is correctly filled in the code, the training data path is "./train_data/", the test data path is "./test_data/", and the model and data of the training output is "./output/", so that the train/test data can be downloaded into the container and the output data stored in BOS.

How to troubleshoot the error of "invalid region, please check and try again"?

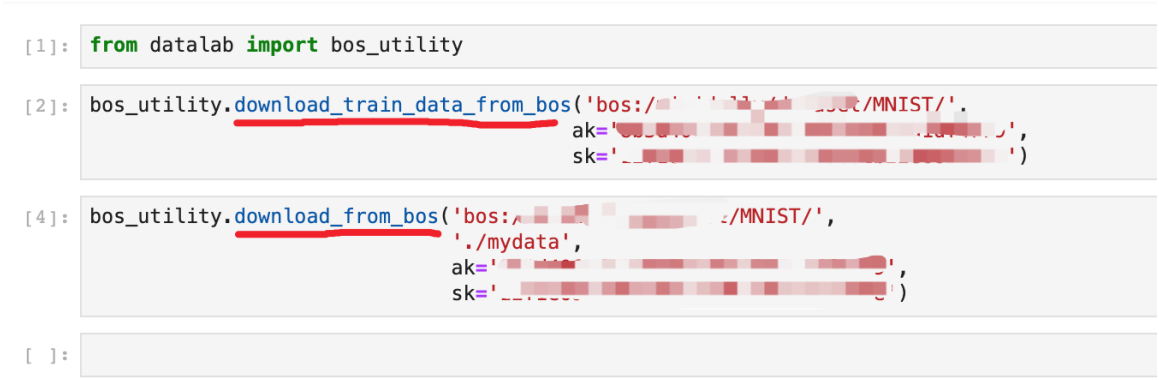
First, check whether the "Bucket" area is North China-Beijing. BML currently only supports "Bucket" reading or writing in North China-Beijing region. If "Bucket" region is North China-Beijing, this error is still reported. It is likely that you have switched to other region in other products, which will cause this error when switching back to BML. The solution is that you can enter Baidu Cloud Compute BCC product on the console, select "North China-Beijing" as the region, and then enter BML product for operation.

Does it support sub-account?

BML has not been connected to the multi-account system at present, so it does not support sub-account now. Please use a normal account. The demand for sub-account is under planning, please look forward to it.

How to download data from unmounted BOS directory to Notebook?

The data under the BOS path selected when starting the workarea will be directly mounted in the "Data" directory shown on the left. When downloading data that is not under the bos path, you can use bos_utility's download_train_data_from_bos or download_from_bos and other methods to download. You need to fill in your own aksk (Get aksk method) in use. The difference is that download_train_data_from_bos downloads data to './train_data' directory, while download_from_bos specifies target_path after downloading, for example, to the'./mydata' directory. The main use method is shown in the figure below.



Service Level Agreement (SLA)

BML Service Level Agreement (SLA) (V1.0)

The Service Level Agreement (hereinafter referred to as "SLA") specifies the service availability level index and compensation scheme of the AI development platform Baidu Machine Learning (hereinafter referred to as "BML") provided by Baidu AI Cloud.

1. Definition

Service Cycle: A service cycle is one natural month. **Instances**: BML product workarea, training module and prediction module, as well as service that selects resource package configuration to run. **Total minutes of single instance service cycle**: Calculated according to the total number of days * 24 (hours) * 60 (minutes) in the single instance service cycle. **Unavailable minutes of single instance service**: The sum of unavailable minutes of a single instance within a service cycle. **Instance not available**: Due

to the failure of Baidu AI Cloud, the BML product workarea module, training or prediction instance cannot be created or service jobs cannot be completed, and the status lasts for more than one minute, which is deemed that the BML product instance is unavailable within that minute. **Monthly service fee**: It is the total service fee paid for a single instance of BML product in a service cycle (i.e. natural month). If the user pays for multiple prepaid packages multiple prepaid packages, the monthly service fee will be calculated according to the actual consumption of that month.

2. Service availability

2.1 Service availability calculation method

The service availability of BML will count the availability of each BML instance according to the service cycle:

$$Service\ availability = \left(\frac{Total\ minutes\ in\ the\ service\ cycle\ of\ a\ single\ instance\ -\ unavailable\ minutes\ of\ a\ single\ instance\ service}{Total\ minutes\ of\ single\ instance\ service\ cycle}\right) \text{ x } 100\%$$

2.2 Service Availability Commitment

Baidu AI Cloud promises that the availability of BML services in a service cycle will not be less than 99.95%;

If BML service fails to meet the above service availability commitment, the customer may obtain compensation in accordance with Article 3 of this agreement. The scope of compensation does not cover the unavailability caused by the following reasons:

(1) System maintenance after Baidu AI Cloud notified users in advance, including handover, maintenance, upgrade and simulated failure drill;

(2) Any network or equipment failure or configuration adjustment other than the equipment to which Baidu AI Cloud belongs;

(3) User's application or data information attacked by hackers;

(4) Loss or disclosure of data and passwords caused by improper maintenance or confidentiality of users;

(5) Negligence of the user or operation authorized by the user;

(6) Force majeure and accidents;

(7) There is a risk of data loss in the local disk of the cloud server stored using the local disk (such as damaged local disk, etc.), and the data on the local disk and the local disk are not available as startup dependencies;

(8) BML services only compensate for BML itself, not other cloud services associated with BML instances;

(9) BML services caused by reasons other than Baidu AI Cloud cannot be used normally.

3. Service Compensation Clause 3.1 Compensation standard
   If the service availability is lower than 99.95%, compensation can be obtained according to the standard in the table below. The compensation method is only limited to vouchers used to purchase BML products, and the total amount of compensation shall not exceed the monthly service fee paid by the customer for the BML instance in the month when the service availability commitment is not met.

Service availability	Amount of vouchers for compensation
Lower than 99.95% but equal to or higher than 99%	10% of monthly service fee
Lower than 99% but equal to or higher than 95%	25% of monthly service fee
Lower than 95%	100% of monthly service fee

3.2 Compensation Application Time Limit

Users can apply for compensation after the fifth (5th) work day of each month for service failing to reach availability last month. Compensation application must be made within two (2) months after the end of the relevant month in which BML does

not reach availability. Compensation application exceeding the time limit shall not be accepted.

Other instructions

(1) Baidu reserves the right of final interpretation of this agreement to the extent permitted by laws and regulations. (2) Baidu AI Cloud has the right to amend the SLA clause once this agreement is published and becomes effective immediately. If there are any changes to the SLA clause, Baidu AI Cloud will notify you by means of website publicity or email. If you do not agree with Baidu AI Cloud's modification to the SLA clauses, you have the right to stop the use of INF service. If you continue to use INF service, it will be deemed that you have accepted the modified SLA clauses. (3) All notices given by Baidu AI Cloud to users under this agreement can be made in web page notice, station letter, email, short message or other forms; such notices shall be deemed to have been delivered to the receipt on the date of delivery. Baidu AI Cloud will not be held responsible for any loss suffered by users due to their failure to timely know the service change or termination clause of Baidu AI Cloud. (4) The conclusion, execution and interpretation of this agreement and the settlement of disputes shall be governed by Chinese laws and shall be subject to the jurisdiction of Chinese courts. In case of any dispute between the two parties regarding the content of this agreement or its execution, the two parties shall try their best to resolve it through friendly negotiation; if the negotiation fails, either party may bring a lawsuit to the Haidian District People's Court, Beijing. (5) This Agreement constitutes the complete agreement between both parties on the matters agreed in this Agreement and other related matters. Except as provided in this Agreement, no other rights are granted to the parties to this Agreement. (6) If any agreement in this agreement is totally or partially invalid or unenforceable for any reason, the remaining agreements in this agreement shall remain valid with binding force. (7) For the terms of user restriction, please see the "Rights and Obligations of Users" in the User Service Agreement for more information. (8) For the exemption clause of service providers, please see the relevant clause of "Disclaimer" in the User Service Agreement for more information.