Machine Learning Platform for AI

FAQ

FAQ

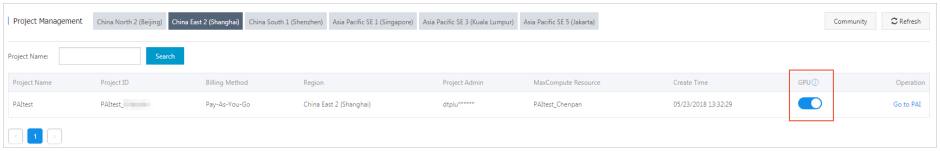
FAQ of deep learning

Contents

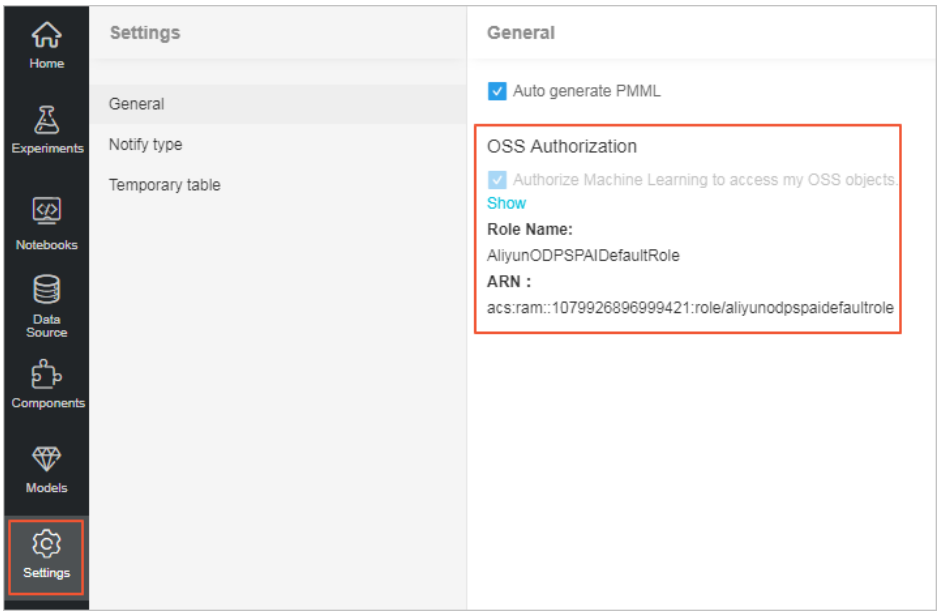
If you are still unable to solve your issue using preceding information, see the Machine learning documentation or send us your logview in a ticket.

How to enable deep learning?

The deep learning feature is currently in beta testing. Three deep learning frameworks: TensorFlow, Caffe, and MXNet are supported. To enable deep learning, log on to your machine learning platform console and enable GPU resources, as shown in the following figure.



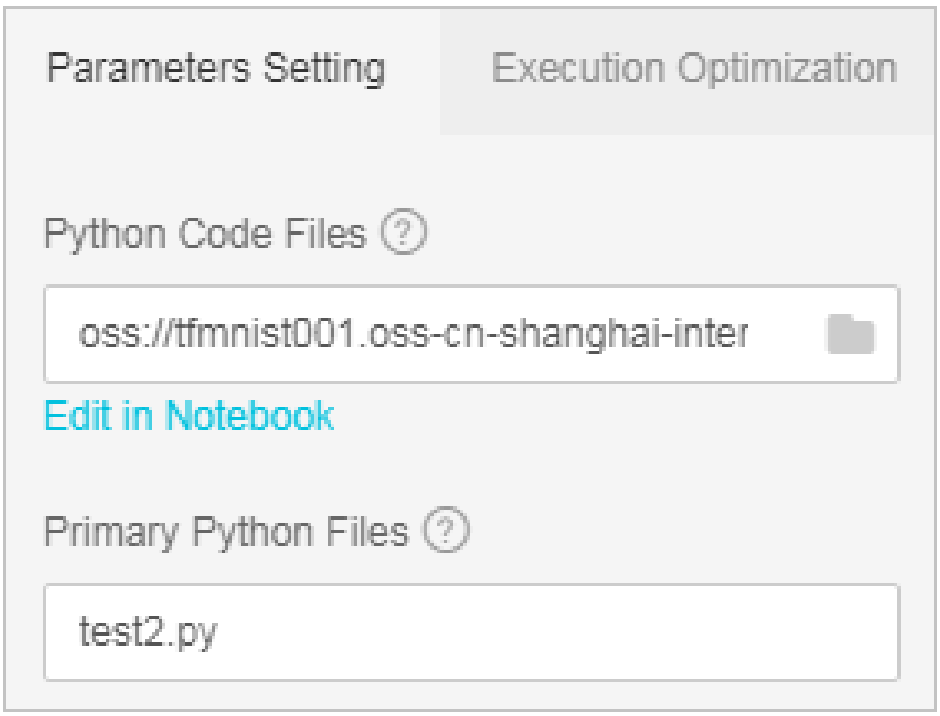
After GPU resources have been enabled, the corresponding projects are allowed to access the public resource pool and dynamically use the underlying GPU resources. You also need to grant OSS permissions to the machine learning platform as follows.



How to reference multiple Python scripts?

You can use Python modules to organize your training scripts. You can write different pieces of your model in multiple Python files. For example, you can write data preprocessing code in one Python file and use another Python file as the entry file for your algorithm.

The functions in test2.py have referenced specified functions that were defined in test1.py, and test2.py is the entry file for the algorithm. You can compress test1.py and test2.py into a TAR.GZ file and upload this compressed file as follows:



- **Python Code File** Select the compressed TAR.GZ file.
- **Main Python File** Select the entry file for the algorithm.

How to upload data to OSS?

When using deep learning to process your data, you must upload your data to OSS buckets. First, create OSS buckets. The deep learning GPU clusters are only available in the China (Shanghai) region. Therefore, you must choose this region when creating OSS buckets. Your data is then transmitted over the Alibaba Cloud classic network. No traffic fees are incurred when you run your algorithms. After OSS buckets have been created, you can log on to the OSS console to create folders and upload your data.

> OSS supports multiple methods to upload data. For more information, see: https://www.alibabacloud.com/help/doc-detail/31848.html.
>
> OSS also provides many tools to help you make full use of this service. For more information, see: https://www.alibabacloud.com/help/doc-detail/44075.html.

We recommend that you use the ossutil or osscmd tool to upload or download files using command lines. These tools can resume a transmission from breakpoints.

> **Note**: You must specify your Access Key ID to use these tools. You can log on to the Access Key console to create or view Access Keys.

How to read OSS data?

Python does not support OSS data. You cannot run code that contains file operations such as Open(), os.path.exist() to read OSS data.Scipy.misc.imread() and numpy.load() are not supported either.

You can read OSS data in the machine learning platform by using these approaches:

You can use **tf.gfile** functions to perform simple file operations.

```
tf.gfile.Copy(oldpath, newpath, overwrite=False) # Copy a file
tf.gfile.DeleteRecursively(dirname) #Recursively delete all files under the specified directory
tf.gfile.Exists(filename) # Check whether the specified file exists
tf.gfile.FastGFile(name, mode='r') # Non-blocking read
tf.gfile.GFile(name, mode='r') # Read a file
tf.gfile.Glob(filename) # List all files in the specified folder. You can use patterns to filter these files.
tf.gfile.IsDirectory(dirname) #Return whether the specified dirname is a directory
tf.gfile.ListDirectory(dirname) # List all files under the specified dirname
tf.gfile.MakeDirs(dirname) # Create a folder under the specified dirname. If the parent directory does not exist, a parent directory is automatically created.
If the parent directory already exists and is writable, True is returned.
tf.gfile.MkDir(dirname) # Create a folder under the specified dirname
tf.gfile.Remove(filename) #Delete the specified filename
tf.gfile.Rename(oldname, newname, overwrite=False) # Rename
tf.gfile.Stat(dirname) # Return statistical data of the specified dirname
tf.gfile.Walk(top, inOrder=True) # Return the file tree of the specified directory
```

For more information, see the tf.gfile module.

You can use tf.gfile.Glob, tf.gfile.FastGFile, tf.WhoFileReader(),and tf.train.shuffer_batch() to perform batch file operations. You must retrieve the file list before reading file data. To read multiple files, you must create a batch file.

When creating deep learning experiments in the machine learning platform, you must specify the parameters, such as the directory to read data, and the code files on the right side of the page. You can denote these parameters as "—XXX" (XXX represents a string) in **tf.flags**.

```
import tensorflow as tf
FLAGS = tf.flags.FLAGS
tf.flags.DEFINE_string('buckets', 'oss://{OSS Bucket}/', 'The folder where training image data is stored')
tf.flags.DEFINE_string('batch_size', '15', 'batch size')
files = tf.gfile.Glob(os.path.join(FLAGS.buckets,'*.jpg')) #List the paths of all JPG files in the buckets
```

Use tf.gfile.FastGfile() to read a small number of files.

```
for path in files:
file_content = tf.gfile.FastGFile(path, 'rb').read() #You must specify rb when calling this function. Otherwise, errors may occur.
```

```
image = tf.image.decode_jpeg(file_content, channels=3) # In this example, we use JPG images.
```

Use tf.WhoFileReader() to read a large number of files.

```
reader = tf.WholeFileReader()  # Create a reader object
fileQueue = tf.train.string_input_producer(files) # Create a queue for the reader to read
file_name, file_content = reader.read(fileQueue) # Use the reader to read a file from the queue
image_content = tf.image.decode_jpeg(file_content, channels=3) # Decode the file contents into images
label = XXX # Label operations are omitted.
batch = tf.train.shuffle_batch([label, image_content], batch_size=FLAGS.batch_size, num_threads=4,
capacity=1000 + 3 * FLAGS.batch_size, min_after_dequeue=1000)

sess = tf.Session() # Create a Session object
tf.train.start_queue_runners(sess=sess) # Note: You must add this function to start the queue. Otherwise, the thread is blocked.
labels, images = sess.run(batch) # Obtain the results
```

Explanations:

tf.train.string_input_producer: converts the files into a queue. You must use **tf.train.start_queue_runners** to start the queue.

Parameters in tf.train.shuffle_batch are as follows:

batch_size: The batch size. This represents the number of data entries that are returned in each batch operation.

num_threads: The number of threads. This is usually set to 4.

capacity: The maximum number of files that are randomly selected in each batch operation. For example, assume that a dataset contains 10,000 files. Set **capacity** to 5,000 if you want to randomly select files from a maximum of 5,000 files.

min_after_dequeue: The minimum length of the queue. This must not exceed the specified **capacity**.

How to write data to OSS?

Use tf.gfile.FastGFile() to write data to OSS

```
tf.gfile.FastGFile(FLAGS.checkpointDir + 'example.txt', 'wb').write('hello world')
```

Use tf.gfile.Copy() to copy data to OSS

```
tf.gfile.Copy('./example.txt', FLAGS.checkpointDir + 'example.txt')
```

You can write data to OSS by using the preceding functions. The files are stored under the following directory: “Output directory/model/example.txt”.

TensorFlow case studies

How can I perform image classification using TensorFlow?

- Document: https://www.alibabacloud.com/help/doc-detail/75343.html
- Source code and resources: https://www.alibabacloud.com/help/doc-detail/94609.html

Other questions

How do I view Tensorflow logs?

For more information, see: https://www.alibabacloud.com/help/doc-detail/75343.html.

What is the function of **model_average_iter_interval** when two GPUs are used?

If **model_average_iter_interval** is not set, the parallel Stochastic Gradient Descent (SGD) algorithm is used, and the gradient is updated in each iteration.

If **model_average_iter_interval** is larger than one, the model average is calculated at regular intervals. **model_average_iter_interval** specifies the interval.

Two GPUs accelerate the training speed.

FAQ

How do I upload data?

To upload data on the Machine Learning Platform for AI web interface, make sure that the data is less than 20 MB. To upload data that is greater than 20 MB, you must download the MaxCompute client and then use the tunnel command.

How do I set the algorithm parameters?

To set the algorithm parameters, drag an algorithm component to the canvas and click the component. The corresponding parameters are displayed in the right-side pane.

How do I view experiment results?

If Machine Learning Platform for AI has successfully run a component, it marks the component with a green check. You can right-click a component with a green check to view data or evaluation results.

How do I view and download the model generated from an experiment?

To generate a model, you must first select **Setting > General > Auto-generate PMML** from the left-side navigation pane. After successfully running an experiment, you can select Model from the left-side navigation pane to check the corresponding model. To view the model parameters, right-click the model. To download a model, right-click the model and select Download PMML.

What is PMML?

PMML is a standard model description file. A PMML file downloaded from Machine Learning Platform for AI can be applied to open-source engines, such as Spark.