

Profile Tensorflow using Tensorboard

- [Overview](#)
- [Code example](#)
- [Local profiling on your own computer](#)
- [Remote Profiling on HAL system](#)

Overview

This guide will show our users how to use the TensorFlow Profiler to profile the execution of your TensorFlow code.

Code example

Copy and paste the following code into **tf-profile.py**.

```
from datetime import datetime
import os
import tensorflow

from tensorflow.keras.datasets import mnist
from tensorflow import keras
from tensorflow.keras import layers

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

model = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="softmax")
])

model.compile(optimizer="rmsprop",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])

train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype("float32") / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype("float32") / 255

# Create a TensorBoard callback
logs = "logs/" + datetime.now().strftime("%Y%m%d-%H%M%S")

tboard_callback = tensorflow.keras.callbacks.TensorBoard(log_dir = logs,
                                                         histogram_freq = 1,
                                                         profile_batch = '10,20')

model.fit(train_images,
          train_labels,
          epochs=10,
          batch_size=128,
          callbacks = [tboard_callback])
```

The `tensorflow.keras.callbacks.TensorBoard` command will create a **tensorboard callback** and `profile_batch` will pick batch number **10** to batch number **20**.

Local profiling on your own computer

1. Run the code with command

```
python tf-profile.py
```

2. Compress the **logs** folder

```
tar -zcvf ./logs.tar.gz ./logs
```

3. Download the tarball file with **sftp** and/or **hal-ondemand**.
4. Decompress the tarball file

```
tar -zxvf ./logs.tar.gz
```

5. Install the tensorboard profile plugin in your python environment.

```
pip install tensorboard_plugin_profile
```

6. Launch the tensorboard with profiler installed.

```
tensorboard --logdir ./logs
```

7. Open the tensorboard dashboard in your web browser. (Google Chrome is strongly recommended)

TensorBoard

SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS PROFILE

UPLOAD

CAPTURE PROFILE

Runs (4)
20210617-150725/train/2021_06_...

Tools (8)
input_pipeline_analyzer

Hosts
RZB15

WARNINGS

- No step marker observed and hence the step time is unknown. This may happen if (1) training steps are not instrumented (e.g., if you are not using Keras) or (2) the profiling duration is shorter than the step time. For (1), you need to add step instrumentation; for (2), you may try to profile longer.

Summary of input-pipeline analysis
No step time measured. Therefore we cannot tell where the performance bottleneck is.

Recommendation for next step:
You may skip the rest of this page.

Host-side analysis details

Breakdown of input processing time on the host

Component	Time (ms)	Percentage
Data preprocessing (in ms)	0.307	100%
Enqueuing data		
Reading data from files in advance		
Other data r		

What can be done to reduce above components of the host input time:

- Enqueuing data: you may want to combine small input data chunks into fewer but larger chunks.
- Data preprocessing: you may increase `num_parallel_calls` in `Dataset.map()` or preprocess the data OFFLINE.
- Reading data from files in advance: you may tune parameters in the following tf.data API (`prefetch_size`, `interleave_cycle_length`, `reader_buffer_size`)

Remote Profiling on HAL system

Coming soon...