

TensorBoard小试牛刀

导师: GAUSS

目录

1/ TensorBoard简介

2/ Keras训练中使用

3/ 自定义训练中使用

TensorBoard简介

TensorBoard使用

TensorBoard是一个在深度学习中很好的**可视化训练过程**和**模型结构**的工具，那么，要怎样才能TensorFlow2.0中使用它呢？

在TensorFlow2.0中，训练一个神经网络模型主要有两种方式：

- 使用tf.keras模块的**Model.fit()**；
- 使用**tf.GradientTape()**求解梯度，这样可以自定义训练过程。

对于这两种方案，都可以使用TensorBoard，下面以在MNIST数据集上训练一个图像分类模型为例介绍。

Keras训练中使用

Keras训练中使用

Keras在回调函数中内置Tensorboard函数：

```
tf.keras.callbacks.TensorBoard(  
    log_dir='logs',  
    histogram_freq=0,  
    write_graph=True,  
    write_images=False,  
    update_freq='epoch',  
    profile_batch=2,  
    embeddings_freq=0,  
    embeddings_metadata=None  
)
```

Keras训练中使用

参数	解释
log_dir	保存TensorBoard要解析的日志文件的目录的路径。
histogram_freq	频率（在epoch中），计算模型层的激活和权重直方图。如果设置为0，则不会计算直方图。必须为直方图可视化指定验证数据（或拆分）。
write_graph	是否在TensorBoard中可视化图像。当write_graph设置为True时，日志文件可能会变得非常大。
write_images	是否在TensorBoard中编写模型权重以显示为图像。
update_freq	‘batch’ 或 ‘epoch’ 或整数。使用 ‘batch’ 时，在每个batch后将损失和指标(评估函数)写入TensorBoard。这同样适用 ‘epoch’ 。如果使用整数，比方说1000，回调将会在每1000个样本后将指标和损失写入TensorBoard。请注意，过于频繁地写入TensorBoard会降低您的训练速度。
profile_batch	分析批次以采样计算特征。profile_batch必须是非负整数或正整数对的逗号分隔字符串。一对正整数表示要分析的批次范围。默认情况下，它将配置第二批。将profile_batch = 0设置为禁用性能分析。必须在TensorFlow eager模式下运行。
embeddings_freq	可视化嵌入层的频率（以epoch为单位）。如果设置为0，则嵌入将不可见。
embeddings_metadata	字典，它将层名称映射到文件名，该嵌入层的元数据保存在该文件名中。

Keras训练中使用案例

Keras训练中使用案例(ps: windows上使用容易遇到bug)

```
model = MyModel()
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir="keras_logv1",
                                                       histogram_freq=1,
                                                       profile_batch = 100000000)

model.fit(x=x_train,
          y=y_train,
          epochs=1,
          validation_data=(x_test, y_test),
          callbacks=[tensorboard_callback])
```


- ☐ Show data download links
- ☒ Ignore outliers in chart scaling

Tooltip sorting method: **default**

Smoothing

0.6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

- ☒ ☐ train
- ☒ ☐ validation

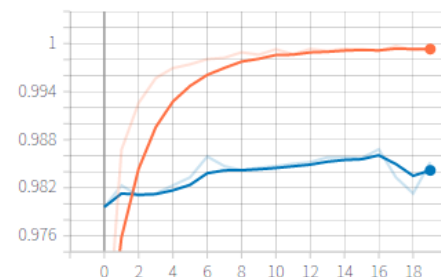
TOGGLE ALL RUNS

keras_logv1/

🔍 Filter tags (regular expressions supported)

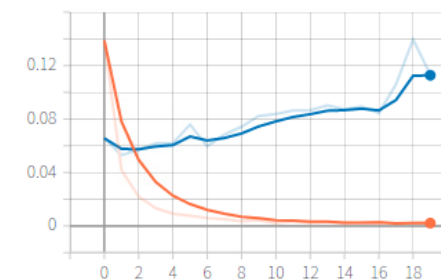
epoch_accuracy

epoch_accuracy



epoch_loss

epoch_loss





Search nodes. Regexes supported.

Fit to Screen

Download PNG

Run (1) train

Tag (1) Default

Upload ☒ Graph☐ Conceptual Graph☐ Profile☐ Trace inputs☐ Show health pillsColor ☒ Structure☐ Device☐ XLA Cluster☐ Compute time☐ Memory☐ TPU Compatibility

colors same substructure

☐ unique substructure

Close legend.

Graph (* = expandable)

Namespace* ?

OpNode ?

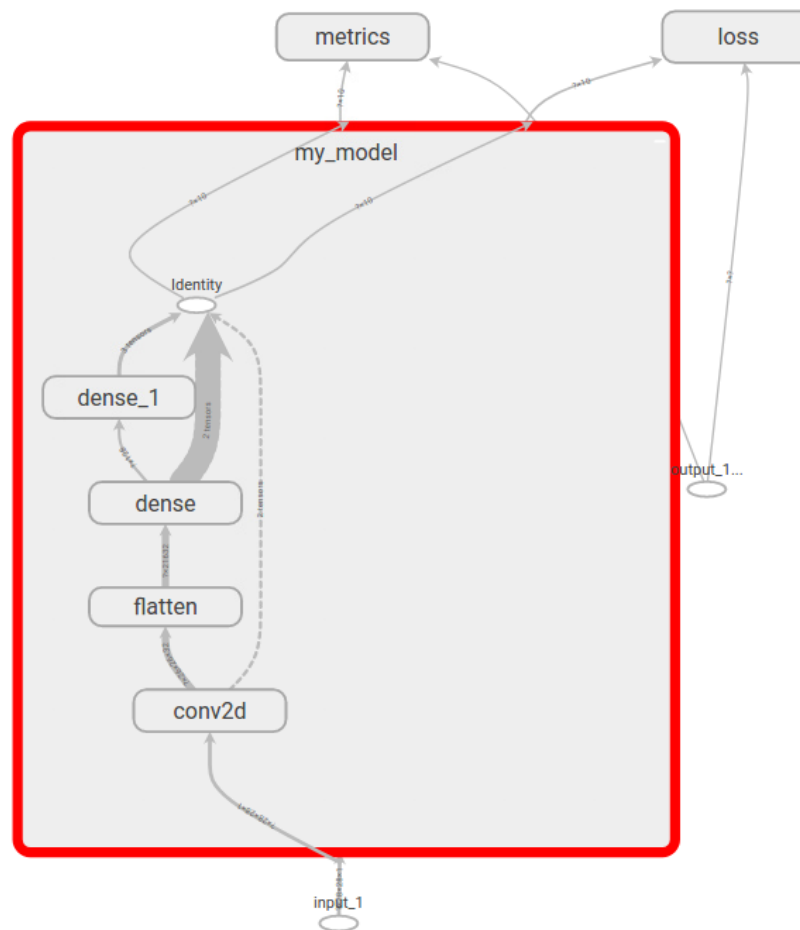
Unconnected series* ?

Connected series* ?

Constant ?

Summary ?

Dataflow edge ?



Horizontal axis

🔍 Filter tags (regular expressions supported)

STEP

RELATIVE

WALL

Runs

Write a regex to filter runs

☒ ☐ train☒ ☐ validation

TOGGLE ALL RUNS

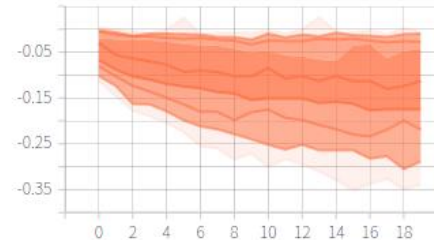
keras_logv1/

my_model

6

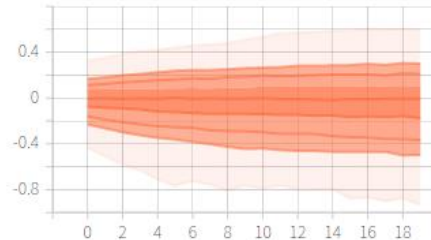
my_model/conv2d/bias_0

train



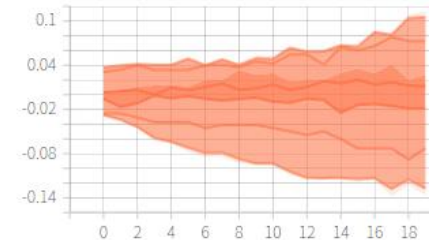
my_model/conv2d/kernel_0

train



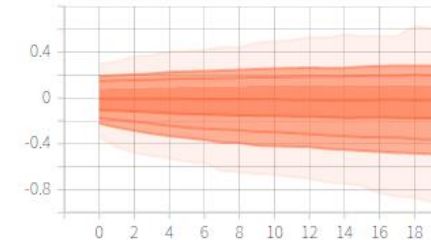
my_model/dense_1/bias_0

train



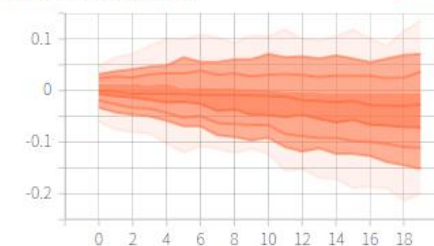
my_model/dense_1/kernel_0

train



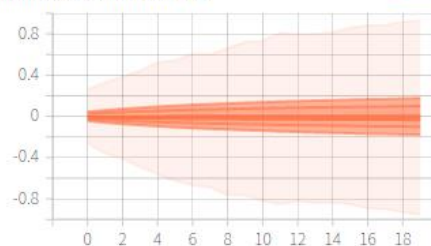
my_model/dense/bias_0

train



my_model/dense/kernel_0

train





Histogram mode

OVERLAY

OFFSET

Offset time axis

STEP

RELATIVE

WALL

Runs

Write a regex to filter runs

☒ train

☒ validation

TOGGLE ALL RUNS

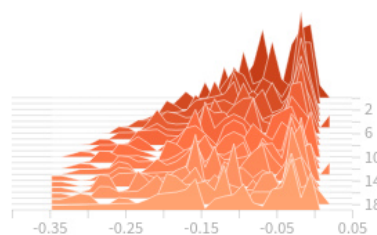
keras_logv1/

Filter tags (regular expressions supported)

my_model

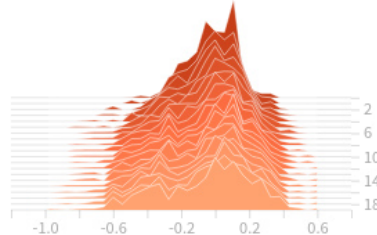
my_model/conv2d/bias_0

train



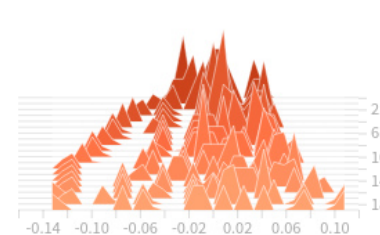
my_model/conv2d/kernel_0

train



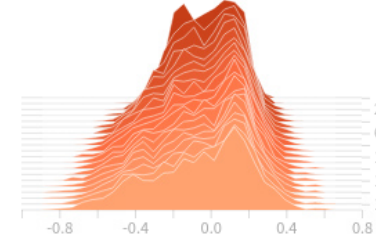
my_model/dense_1/bias_0

train



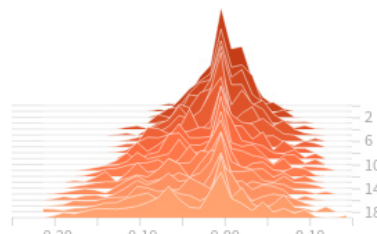
my_model/dense_1/kernel_0

train



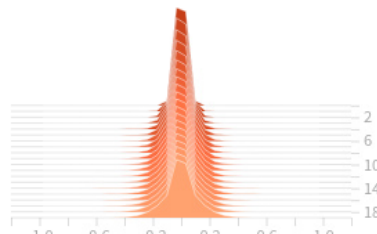
my_model/dense/bias_0

train



my_model/dense/kernel_0

train



Tensorboard界面解释

Scalars : 显示了如何将loss与每个时间段改变。还可以使用它来跟踪训练速度, 学习率和其他标量值。

Graphs: 进行可视化模型。在这种情况下, 将显示层的Keras图, 这可以帮助你确保模型正确构建。

Distributions 和 **Histograms** : 显示张量随时间的分布。这对于可视化权重和偏差并验证它们是否以预期的方式变化很有用。

自定义训练中使用

创建 summary writer 实例

```
tf.summary.create_file_writer(  
    logdir, max_queue=None, flush_millis=None, filename_suffix=None, name=None  
)
```

logdir	文件夹路径。
max_queue	最多在缓存中暂存max_queue个数据，当超过max_queue个时，flush更新到日志文件中并清空缓存。默认为10。
flush_millis	至少flush_millis毫秒内进行一次flush。默认为120,000毫秒。
filename_suffix	日志文件的后缀。默认为.v2。
name	(tf1.0残余特性，请忽略它) 本操作的名称。



创建 summary writer 实例

```
tf.summary.image(  
    name, data, step=None, max_outputs=3, description=None  
)  
tf.summary.scalar(  
    name, data, step=None, description=None  
)  
tf.summary.text(  
    name, data, step=None, description=None  
)  
  
tf.summary.histogram(  
    name, data, step=None, buckets=None, description=None  
)  
  
tf.summary.audio(  
    name, data, sample_rate, step=None, max_outputs=3, encoding=None,  
    description=None  
)
```


summary 写入函数

- `tf.summary.scalar()`: 保存单个数值。在tensorboard中将生成折线图。
- `tf.summary.histogram()`: 保存一个张量。直方图或密度图。
- `tf.summary.text()`: 保存一个`tf.string`类型的Tensor。文本。
- `tf.summary.image()`: 保存形状为`[k, h, w, c]`的Tensor。图片。
- `tf.summary.audio()`: 保存形状为`[k, t, c]`的Tensor。音频。

查看 Graph 和 Profile 信息

- `tf.summary.trace_export()`: 停止trace, 并将之前trace记录到的信息写入profiler日志文件。
- `tf.summary.trace_off()`: 停止trace, 并舍弃之前trace记录。
- `tf.summary.trace_on()`: 开始trace以记录计算图和分析信息。



```
stamp = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
import os
logdir = os.path.join("logs/"+stamp)

summary_writer = tf.summary.create_file_writer(logdir)

tf.summary.trace_on(graph=True, profiler=True) # 开启Trace, 可以记录图结构和profile信息
EPOCHS = 20

for epoch in range(EPOCHS):
    for (x_train, y_train) in train_ds:

        train_step(x_train, y_train)

        with summary_writer.as_default(): # 希望使用的记录器
            tf.summary.scalar('loss', train_loss.result(), step=epoch)
            tf.summary.scalar('accuracy', train_accuracy.result(), step=epoch) # 还可以添加其他自定义的变量

#     for (x_test, y_test) in test_ds:
#         test_step(x_test, y_test)
#     with summary_writer.as_default(): # 希望使用的记录器
#         tf.summary.scalar('test_loss', test_loss.result(), step=epoch)
#         tf.summary.scalar('test_accuracy', test_accuracy.result(), step=epoch) # 还可以添加其他自定义的变量

template = 'Epoch {}, Loss: {}, Accuracy: {}, Test Loss: {}, Test Accuracy: {}'
print(template.format(epoch + 1,
                      train_loss.result(),
                      train_accuracy.result() * 100,
                      test_loss.result(),
                      test_accuracy.result() * 100))

# Reset metrics every epoch
train_loss.reset_states()
test_loss.reset_states()
train_accuracy.reset_states()
test_accuracy.reset_states()

with summary_writer.as_default():
    tf.summary.trace_export(name="model_trace", step=3, profiler_outdir=logdir) # 保存Trace信息到文件
```

☐ Show data download links☒ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0.6

Horizontal Axis

STEP

RELATIVE

WALL

Runs

Write a regex to filter runs

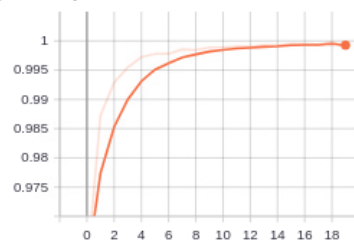
☒ 20200603-222237

TOGGLE ALL RUNS

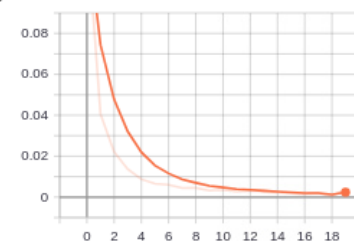
logs/

Filter tags (regular expressions supported)

accuracy

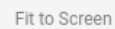
accuracy
tag: accuracy

loss

loss
tag: loss



Search nodes. Regexes supported.



Download PNG

Run (1) 20200603-222237

Tag (1) model_trace

Upload

Graph

☐ Conceptual Graph

☐ Profile

☐ Trace inputs

☐ Show health pills

Color ☒ Structure

☐ Device

☐ XLA Cluster

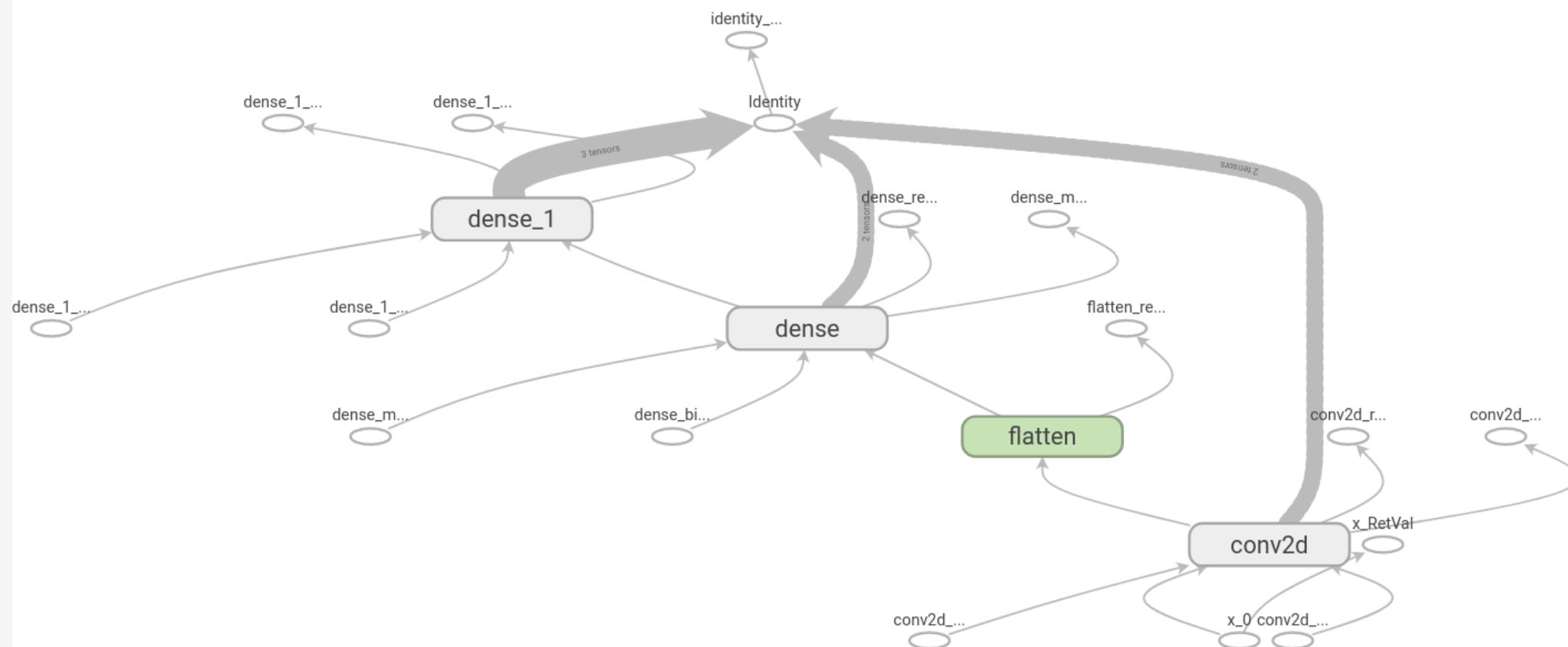
☐ Compute time

☐ Memory

- TPU Compatibility

colors same substructure

 unique substructure



TensorBoard

SCALARS

GRAPHS

PROFILE

INACTIVE



CAPTURE PROFILE

Runs (1)

20200603-222237/2020-06-03_22-30-C ▾

Tools (1)

trace_viewer ▾

Hosts (1)

local ▾

Trace View

M

View Options ▾



▼ /host:CPU (pid 0)

▼ python

forward_
forward_
EagerK_
forward_



本节小结

Summary

TensorBoard 小试牛刀	TensorBoard简介	
	用Keras训练中使用	tf.keras.callbacks.TensorBoard
	自定义训练中使用	tf.summary

结语

——我 说——

看过千万代码，不如实践一把！





深度之眼
deepshare.net

联系我们：

电话：18001992849

邮箱：service@deepshare.net

QQ：2677693114



公众号



客服微信

