

VIP买一年送一年再赠羽绒服/加绒卫衣，抽万元壕礼！

抢先加购

CSDN

首页 博客 专栏课程 下载 问答 社区 插件 认证

郑维亮: 你好，非常感谢您的文章。我对自己采集的语音数据转换成频谱图后，得 ...

您愿意向朋友推荐“博客详情页”吗？
强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章
python日志记录系列教程，内
置logging模块（三）——三种配置日志记
录的方式

python日志记录系列教程，内
置logging模块（二）——logging日志进阶教
程（五大核心组件）

python日志记录系列教程，内
置logging模块（一）——直接使
用logging模块的基础日志记录

2020年 28篇 2019年 170篇
2018年 134篇 2017年 25篇
2016年 3篇

LoveMiss-Y

已关注

60 22 200

专栏目录

免费资源，更有超多电子书、面试真题！

前言：tensorboard是一个非常强大的工具，不仅仅可以帮助我们可视化神经网络训练过程中的各种参数，而且可以帮助我们更好的调整网络模型、网络参数，这一块内容后面会讲到，不管是 tensorflow、keras、还是pytorch，tensorboard都提供了非常好的支持，本文是系列文章的第一篇，详细介绍基于keras+tensorboard如何进行网络的可视化。写这篇文章的初衷很简单，网上很多方面类似的文章，但是还没有找到一篇真正满意的，很多都是互相转载，很多要么是太过简单，不具备什么参考价值，本文会从浅入深的来说明。

一、keras是如何使用tensorboard的

keras使用tensorboard是通过回调函数来实现的，关于什么是keras的“回调函数”，这里就不再赘述了，所以Tensorboard也是定义在keras.callbacks模块中的，通过构造一个Tensorboard类的对象，然后在训练的时候在fit里面指定callbacks参数即可，keras使用的一般格式为：

```
1 # 构造一个Tensorboard类的对象
2 tbCallBack = TensorBoard(log_dir='./model')
3
4 # 在 fit 里面指定callbacks参数
5 history=model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, shuffle=True, verbose=2, validation_data=(x_test, y_test), callbacks=[tbCallBack])
```

但是Tensorboard远远不止这么简单，它的参数众多，每一个参数都代表不同的含义。下面将分别来介绍。

在介绍之前，先说一下本文的案例，本文以手写字识别为案例，搭建的模型如下所示：

```
1 model = Sequential()
2 model.add(Conv2D(32, (5,5), activation='relu', input_shape=[28, 28, 1])) #第一卷积层
3 model.add(Conv2D(64, (5,5), activation='relu')) #第二卷积层
4 model.add(MaxPool2D(pool_size=(2,2))) #池化层
5 model.add(Flatten()) #平铺层
6 model.add(Dropout(0.5)) #dropout层
7 model.add(Dense(128, activation='relu')) #Dense层
8 model.add(Dropout(0.5)) #dropout层
9 model.add(Dense(10, activation='softmax')) #Dense层
```

会员中心 收藏 动态 消息 创作

51 订阅 85 篇文章 订阅专栏

?

分类专栏

二、Tensorboard类详解

该类在keras.callbacks模块中。它的参数列表如下：

- **log_dir**: 用来保存被 TensorBoard 分析的日志文件的文件名。
- **histogram_freq**: 对于模型中各个层计算激活值和模型权重直方图的频率（训练轮数中）。如果设置成 0，直方图不会被计算。对于直方图可视化的验证数据（或分离数据）一定要明确的指出。
- **write_graph**: 是否在 TensorBoard 中可视化图像。如果 write_graph 被设置为 True。
- **write_grads**: 是否在 TensorBoard 中可视化梯度值直方图。histogram_freq 必须要大于 0。
- **batch_size**: 用以直方图计算的传入神经网络输入批的大小。
- **write_images**: 是否在 TensorBoard 中将模型权重以图片可视化，如果设置为True，日志文件会变得非常大。
- **embeddings_freq**: 被选中的嵌入层会被保存的频率（在训练轮中）。
- **embeddings_layer_names**: 一个列表，会被监测层的名字。如果是 None 或空列表，那么所有的嵌入层都会被监测。
- **embeddings_metadata**: 一个字典，对应层的名字到保存有这个嵌入层元数据文件的名字。查看 [详情](#) 关于元数据的数据格式。以防同样的元数据被用于所用的嵌入层，字符串可以被传入。
- **embeddings_data**: 要嵌入在 embeddings_layer_names 指定的层的数据。Numpy 数组（如果模型有单个输入）或 Numpy 数组列表（如果模型有多个输入）。[Learn ore about embeddings](#)。
- **update_freq**: 'batch' 或 'epoch' 或 整数。当使用 'batch' 时，在每个 batch 之后将损失和评估值写入到 TensorBoard 中。同样的情况应用到 'epoch' 中。如果使用整数，例如 10000，这个回调会在每 10000 个样本之后将损失和评估值写入到 TensorBoard 中。注意，频繁地写入到 TensorBoard 会减缓你的训练。

注意的是：该部分来自官方文档，有几个地方说得不是很清楚，所以后面我会逐一介绍。红色的是本文会详细讲到的，黑色的是涉及到嵌入层相关的，本文暂且略过。

这些参数的默认值如下所示：

```
1 | log_dir='./logs', # 默认保存在当前文件夹下的logs文件夹之下
2 | histogram_freq=0,
3 | batch_size=32,
4 | write_graph=True, #默认是True, 默认是显示graph的。
5 | write_grads=False,
6 | write_images=False,
7 | embeddings_freq=0,
8 | embeddings_layer_names=None,
9 | embeddings_metadata=None,
10 | embeddings_data=None,
11 | update_freq='epoch' #默认是“epoch”，这个是很好理解的，这与训练集与验证集的loss、acc曲线相关
```

2.1 最简单的Tensorboard类的构造对象（log_dir,write_graph,update_freq三个参数）

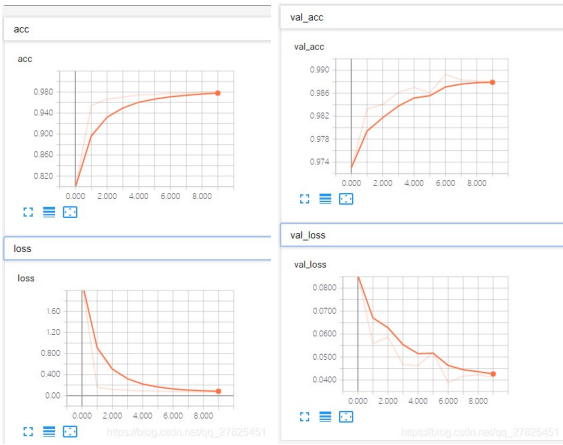
本文以手写字识别为例，因为网络的搭建过程很简单，这里就不给出全部代码了，占地方，只讨论与tensorboard相关的。

最简单的tensorboard对象是不传递任何参数，只是用默认的设置值：如下：

```
1 | tbCallBack = TensorBoard()
2 |
3 | history=model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, shuffle=True, verbose=2, validation_data=(x_test, y_test))
```

这样通过谷歌浏览器启动tensorboard程序，得到如下的结果：

- （1）tensorboard面板中只有scalers和graph两个面板。graph面板是通过默认的write_graph=True来设置的；
- （2）scalers面板中默认给出的是训练数据的loss和accuracy、以及验证集上的loss和accuracy（如果有验证集的话，没有则没有）。本例子训练了10个epochs，而且update_freq="epoch"，是默认值，得到如下所示：



总结：默认的Tensorboard类回调只会显示graph图像和几个训练的scalers，而且日志文件并不是很大，本文中例子只有301K。

2.2 histogram_freq参数

```
1 | tbCallBack = TensorBoard(log_dir="./model", histogram_freq=1)
2 | ...
3 | 该参数用于设置tensorboard面板中的histograms和distributions面板，默认是0，即不显示，设置为大于0的数，本例设置为1，则显示这
4 | ...
```

设置了histogram_freq参数之后，则会显示histograms和distributions面板这两个面板，需要注意的是：

keras中没一个卷积层、全连接层都会默认有权值矩阵，称之为kernel，默认有偏置项bias，所以在这两个面板中会有kernel和bias这两个图形。

DISTRIBUTIONS	HISTOGRAMS	DISTRIBUTIONS	HISTOGRAMS
Filter tags (regular expressions supported)		Filter tags (regular expressions supported)	
conv2d_1		conv2d_1	
conv2d_1_out		conv2d_1_out	
conv2d_2		conv2d_2	
conv2d_2_out		conv2d_2_out	
dense_1		dense_1	
dense_1_out		dense_1_out	
dense_2		dense_2	
dense_2_out		dense_2_out	
dropout_1_out		dropout_1_out	
dropout_2_out		dropout_2_out	
flatten_1_out		flatten_1_out	
max_pooling2d_1_out		max_pooling2d_1_out	

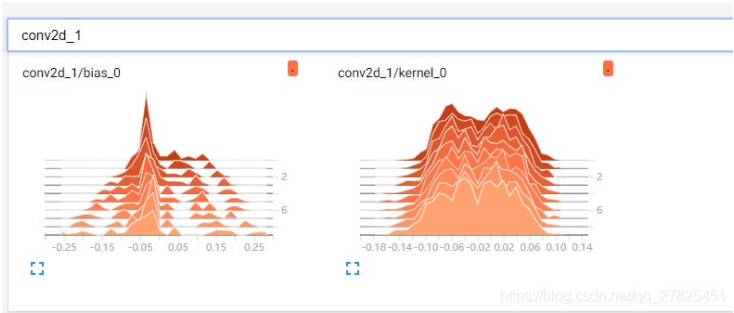
上面是收缩了的面板，我们发现所有的层都会显示出来的，本例中的两个卷积层、两个全连接层、两个dropout层、一个flatten层、一个maxpooling层都显示出来了，但是顺序不是模型中的顺序，会发生变化，是按照英文字母的顺序显示。不仅如此，对于每一个层的输出数据也会被统计出来，对应于上面那些以“out”结尾的名称，表示的是输出。对于像conv卷积层和dense这样的全连接层，除了本身的权阵kernel、偏置项bias外，还有他们的输出，但是像dropout、maxpooling、flatten这样的层本身没有kernel和bias，只有输出，所以只显示输出的统计数据，同样也是以“out”结尾。

注意：本例中每一个层使用的是默认名称，在实际项目里面为了更加清晰的查看，我们最好是自己给每一个层自定义一个名字。

那么如何查看histograms和distributions面板这两个面板的图标呢？这其实是一个很基础的问题，这里通过一个例子来说明吧。

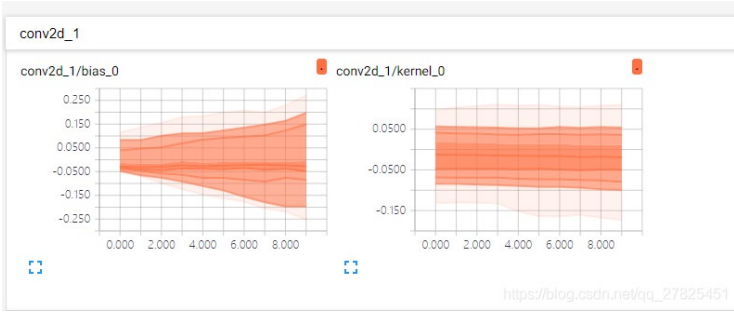
为什么histograms和distributions面板是**相伴而生的**，主要是因为它们其实是同一批数据统计分布的**“不同视角”**呈现的结果，怎么理解？看下图，以第一个卷积层为例conv2d_1。

先看histograms：



histogram本质上就是在没一个epoch上，权重数据kernel和偏执数据bias的统计分布图，看起来很清楚，可以非常方便的看到在每一个epoch，数据的分布变化状况。

再看一下distributions：



仔细分析其实可以发现，distributions其实就是histograms的一个**“俯视图”**，因为histograms中图像是三维的，所以可以更直观的看着起来哪一个范围内的分布更多（即峰值更高），那distributions中变为了一个平面图是不是就不能看了呢？当然不是，依然可以，它可以**通过颜色的深浅来看峰值**，比如上面两图中，在histograms中的bias图中，峰值集中在-0.05附近，我们看下面的distributions中的bias，发现-0.05附近的颜色更深，其实这就是对应的峰值。其实右边的图也是一样的分析。

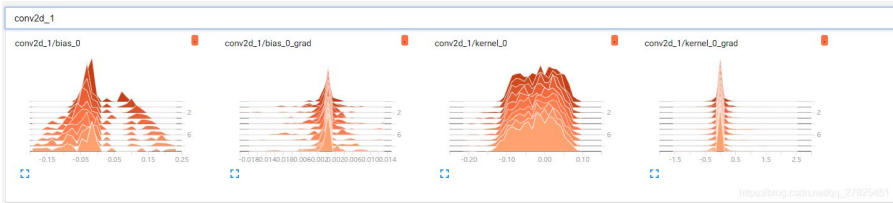
总结：histogram_freq参数会在面板中添加histograms面板和distributions面板，会将每一个层的权值、偏置、输出值的统计分布以不同的视觉角度显示出来，因为权重等信息的存在，现在日志文件变为了111M，增大了很多。

2.3 write_grads参数（默认为False）

将梯度信息也在distributions面板和histograms面板中显示出来，因此它需要依赖 histogram_freq参数，只有当histogram_freq参数大于0（即设置的前提下），然后设置write_grads=True参数才有意义，如下：

```
1 | tbCallback = TensorBoard(log_dir="./model", histogram_freq=1,write_grads=True)
2 | ...
3 | 将梯度信息也在distributions面板和histograms面板中显示出来，因此它需要依赖 histogram_freq参数，只有当histogram_freq参数大于0
4 | ...
```

现在那些可以求梯度的层将会增加梯度的统计过分布信息，包括卷积层conv、全连接层dense等，但是像dropout、flatten、maxpooling层本身没有求梯度，所以相比与之前没什么变化。



上图是第一个卷积层，增加了对权值kernel和偏置bias的梯度统计分布信息。

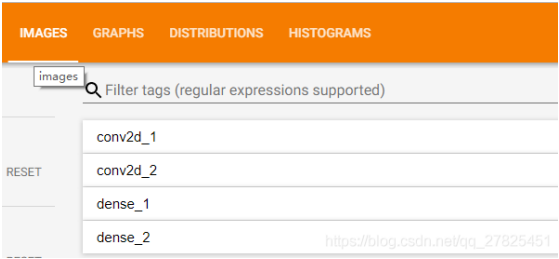
总结：将梯度信息也在distributions面板和histograms面板中显示出来，因此它需要依赖 histogram_freq参数，只有当histogram_freq参数大于0（即设置的前提下），然后设置write_grads=True参数才有意义，由于添加了梯度数据，现在的日志文件更大了，达到了221M。

2.4 write_images参数（默认为False）

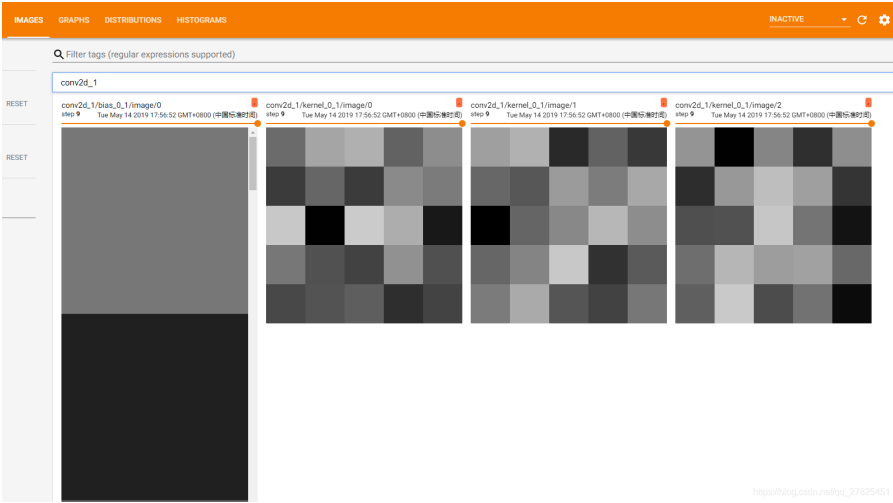
这个参数将那些具有权值kernel和偏置bias的层（如卷积层conv和全连接层dense）的kernel和bias用图片的形式展示出来，所以由于flatten、maxpooling、dropout是没有kernel和bias的，就没有，设定如下：

```
tbCallBack = TensorBoard(log_dir="./model", histogram_freq=1,write_images=True)
```

打开结果如下：



只有两个卷积层，两个全连接层有，因为只有它们有kernel和bias，我们展开一个卷积层查看如下：



因为我的第一个卷积核是5x5的，所以这里的kernel是5x5的，

总结：write_images参数可以将带有权重kernel和bias的网络层的参数用图片的形式显示出来，但是这样做有一个致命的问题，那就是会导致日志文件增加的很庞大，训练过程也会因为这个操作变得更加缓慢，所以非常的不推荐使用。使用该参数之后，之日文件增加到了2.4G，相较于之前增加了10倍之多，这还仅仅是一个很浅层的网络。所以慎用。

从上面可以看出，直接使用keras的回调函数Tensorboard，可以很方便的实现网络的可视化训练，对于一些参数的调优是有帮助的，而且这样的操作更加简单直观，使用起来很方便，当然我完全可以自己定制更加功能强大的实现，这是后面的文章会说到的。

三、该案例的全部代码如下

```
1 import numpy as np
2
3 from keras.models import Sequential # 采用顺序模型
4 from keras.layers import Input, Dense, Dropout, Activation,Conv2D,MaxPool2D,Flatten
5 from keras.optimizers import SGD
6 from keras.datasets import mnist
7 from keras.utils import to_categorical
8 from keras.callbacks import TensorBoard
9
10
11
12 def create_model():
13     model = Sequential()
14     model.add(Conv2D(32, (5,5), activation='relu', input_shape=[28, 28, 1])) #第一卷积层
15     model.add(Conv2D(64, (5,5), activation='relu')) #第二卷积层
16     model.add(MaxPool2D(pool_size=(2,2))) #池化层
17     model.add(Flatten()) #平铺层
18     model.add(Dropout(0.5))
19     model.add(Dense(128, activation='relu'))
20     model.add(Dropout(0.5))
21     model.add(Dense(10, activation='softmax'))
22     return model
23
24 def compile_model(model):
25     #sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True) # 优化函数，设定学习率（lr）等参数
26     model.compile(loss='categorical_crossentropy', optimizer="adam",metrics=['acc'])
27     return model
28
29 def train_model(model,x_train,y_train,batch_size=128,epochs=10):
30
31     #构造一个tensorboard类的对象
32     #tbCallBack = TensorBoard(log_dir="./model", histogram_freq=1, write_graph=True, write_images=True,update_freq=1)
33
34     tbCallBack = TensorBoard(log_dir="./model", histogram_freq=1,write_grads=True)
35     history=model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, shuffle=True, verbose=2, validation_data=(x_test, y_test))
36     return history,model
37
38
39
40 if __name__=="__main__":
41     (x_train,y_train),(x_test,y_test) = mnist.load_data() #mnist的数据我自己已经下载好了
42     print(np.shape(x_train),np.shape(y_train),np.shape(x_test),np.shape(y_test))
43
44     x_train=np.expand_dims(x_train,axis=3)
45     x_test=np.expand_dims(x_test,axis=3)
46     y_train=to_categorical(y_train,num_classes=10)
47     y_test=to_categorical(y_test,num_classes=10)
48     print(np.shape(x_train),np.shape(y_train),np.shape(x_test),np.shape(y_test))
49
50     model=create_model()
51     model=compile_model(model)
52     history,model=train_model(model,x_train,y_train)
```



枯叶绿发: 你好, 按照博主你这样设置特征图write_images=True, tensorboard为什么会没有显示呢? 还有一个就是利用keras 这样生成log文件的名字能不能更改? 谢谢 2月前 回复 **



Abro.: 谢谢! 27天前 回复 **



哗啦啦啦啦: 麻烦问下, 学习率怎么显示呢 6月前 回复 **



介个杀手不太冷 回复:

```
1 class LearningRateLogger(tf.keras.callbacks.Callback):
2     def __init__(self):
3         super().__init__()
4         self._supports_tf_logs = True
5
6     def on_epoch_end(self, epoch, logs=None):
7         if logs is None or "learning_rate" in logs:
8             return
9         logs["learning_rate"] = self.model.optimizer.lr
```

把这个类加入回调列表中即可 4月前 回复 **



aixiayj: 你好, 请问distribution和histogram那里是怎么截图的, 我截图感觉不如您这个漂亮 6月前 回复 **



lytlytly111: 请问一下, 用tensorboard会报这种错误: No module named 'tensorflow.python.types' 是怎么回事呀 8月前 回复 **



unruly_chump 回复: 如果是cpu和gpu都在的话, 建议卸载一个。可能是版本冲突了 昨天 回复 **



helloworldm: 没一个应该是每一个吧, 感谢分享 9月前 回复 **



1535966643: 如果是服务器上运行是不是没有办法实时看 损失曲线呢 10月前 回复 **



介个杀手不太冷 回复: 看我这篇博客: https://mp.csdn.net/mp_blog/creation/editor/118498462 4月前 回复 **



小妖精ing: 博主 所以要怎么启动tensorboard呀 主要是这个不会 10月前 回复 **



仙女也得写代码 回复: cmd输入tensorboard --logdir=my_log_dir 10月前 回复 **



宋yongchao: 请问一下, 我运行代码成功不了, 是不是得自己下载数据集, 你的程序里面没有加载数据集, 1年前 回复 **



不瘦不改名 回复: (x_train,y_train),(x_test,y_test) = mnist.load_data() 这行是下载数据集的, 没有就自己下载了, 如果有就加载 8月前 回复 **



qq_37877721: 你好楼主, 怎么在你的cycle gan的keras代码中添加tensorboard 1年前 回复 **

- keras中TensorBoard 黑米亚的博客 10-5
关于keras中TensorBoard一些东西,突然发现这个好智能。记录1.keras使用tensorboard报错 'Sequential' object has no attribute '_get_distribution_strateg...
- keras+tensorboard使用方法_踏雪飞鸿的博客 11-8
tensorboard --logdir=./logs --port=xxxx# xxxx是四位数,例如0001 那么如何生成tensorboard日志文件呢,因为我用的是keras(比较简单),只用一行命令就可...
- Tensorboard深入详解 (二) ——在keras中自己实现Tensorboard可视化神经网络 Miss-Y的博客 2113
前言: 上一篇文章中《Tensorboard深入详解 (一) ——使用keras结合Tensorboard可视化神经网络详细教程》, 我们可以看出使用keras预定义的回调函...

[illegible]