



# 使用 tfdbg 调试 TensorFlow 模型

2017年3月2日星期四

发布人：工具和基础架构软件工程师 *Shanqing Cai*。

我们怀着激动的心情与大家分享 [TensorFlow 调试程序](#) (`tfdbg`)，这个工具可以简化 TensorFlow 中对机器学习 (ML) 模型的调试。

TensorFlow 是 Google 的开源 ML 内容库，基于[数据流图表](#)。一个典型的 TensorFlow ML 程序包括两个独立的阶段：

- 1. 利用内容库的 [Python API](#) 将 ML 模型设置为数据流图表；
- 2. 利用 `Session.run()` 方法在图表上训练或执行推理。

如果在第二阶段（即 TensorFlow 运行时）出现错误和缺陷，将难以进行调试。

要了解出现这种情况的原因，请注意对标准 Python 调试程序而言，`Session.run()` 调用实际上是单个语句，它并不会公开运行中图表的内部结构（节点及其连接）和状态（节点的输出数组 或 张量）。[gdb](#) 等较低级别的调试程序在组织堆叠框架和变量值时无法令其与 TensorFlow 图表操作产生关联。专业级运行时调试程序是 TensorFlow 用户最常提出的功能请求之一。

`tfdbg` 满足了这一运行时调试需求。让我们通过一段简短的代码来了解 `tfdbg` 的实用效果，这段代码的作用是建立并运行一个简单的 TensorFlow 图表，以通过[梯度下降法](#)拟合一个简单的线性方程。

```
import numpy as np
import tensorflow as tf
import tensorflow.python.debug as tf_debug
xs = np.linspace(-0.5, 0.49, 100)
x = tf.placeholder(tf.float32, shape=[None], name="x")
y = tf.placeholder(tf.float32, shape=[None], name="y")
k = tf.Variable([0.0], name="k")
y_hat = tf.multiply(k, x, name="y_hat")
sse = tf.reduce_sum((y - y_hat) * (y - y_hat), name="sse")
train_op = tf.train.GradientDescentOptimizer(learning_rate=0.02).minimize(sse)

sess = tf.Session()
sess.run(tf.global_variables_initializer())

sess = tf_debug.LocalCLIDebugWrapperSession(sess)
for _ in range(10):
```

Labels

Archive

```
sess.run(train_op, feed_dict={x: xs, y: 42 * xs})
```

正如本例中突出显示的线条所示，会话对象包装成一个用于调试的类 (`LocalCLIDebugWrapperSession`)，因此调用 `run()` 方法会启动 `tfdbg` 的命令行界面 (CLI)。您可以利用鼠标点击或命令执行一遍连续运行调用，检查图表的节点及其属性，通过中间张量列表将图表中所有相关节点完整的执行历史记录可视化。通过使用 `invoke_stepper` 命令，您可以让 `Session.run()` 调用在“步进器模式”下执行，在这种模式下，您可以步进到自己选择的节点，观察并修改其输出，然后再执行进一步的分步调试，其运行方式与调试过程语言（例如 `gdb` 或 `pdb`）类似。

在开发 TensorFlow ML 模型时一类经常遇到的问题是，因溢出、除零、`log(0)` 等错误而导致出现无效数值（无穷大和 `NaN`）。在大型 TensorFlow 图表中，查找此类节点的根源可能既繁琐又耗时。借助于 `tfdbg` CLI 及其条件断点支持，您可以快速找到引发问题的根源节点。下面这段视频演示了如何利用 `tfdbg` 调试神经网络中存在的无穷大/`NaN` 问题：[https://www.youtube.com/watch?time\\_continue=1&v=CA7fjRfduOI](https://www.youtube.com/watch?time_continue=1&v=CA7fjRfduOI)

TensorFlow 调试程序实用效果抓屏，摘自此[教程](#)。

与打印选项等替代性调试选项相比，`tfdbg` 需要改动的代码行数更少，提供的图表覆盖范围更大，并且提供的调试体验交互性更强。它可以加快您的模型开发速度和调试工作流执行速度。它还提供了其他功能，例如[离线调试](#)从服务器环境转储的张量并将其与 `tf.contrib.learn` 集成。首先，请访问此[文档](#)。这篇[研究论文](#)对 `tfdbg` 的设计做了更详尽的展示。

使用 `tfdbg` 时要求安装的最低 TensorFlow 版本为 0.12.1。要报告错误，请在 TensorFlow 的 [GitHub 问题页面](#)上设立问题。如需获得一般使用帮助，请在 StackOverflow 上使用 `tensorflow` 标记发帖提问。

致谢

如果没有 Google TensorFlow 核心/API 团队和应用机器智能团队提供的帮助和反馈，就不可能有这个项目。



中国开发者资源	全球开发技术资源	关注微信
<a href="#">Google 开发者文档</a>	<a href="#">Google 开发者博客</a>	
<a href="#">Android 开发技术文档</a>	<a href="#">Android 开发者博客</a>	
<a href="#">Firebase 开发技术文档</a>	<a href="#">Firebase 博客</a>	



Google

[Google](#) • [Privacy](#) • [Terms](#)