

我的深度学习开发环境详解：TensorFlow + Docker + PyCharm等，你的呢（附问卷）

2017年06月19日 12:45:25 机器之心

0

选自Upflow.co

作者：Killian

机器之心编译

参与：Nurhachu Null、李亚洲

在这篇文章中，研究员 Killian 介绍了自己的深度学习开发环境：TensorFlow + Docker + PyCharm + OSX Fuse + Tensorboard。但根据自己的预算、语言习惯、开发需求，每个人都会配置不同的开发环境，也遇到过各种各样的难题。因此，我们在文后附上了一份调查问卷，希望能了解众多不同开发者的深度学习环境，最终汇集成一篇文章为大家提供不同的洞见。

在尝试用不同的东西来配置深度学习环境这个过程中，我花费了相当多的时间。因此我想着把自己目前的工作流程整理成文档，希望可以帮助到尝试着做同样事情的人。

目标

在开始创建我的模型之前，我脑海中会有几个清晰的目标，即理想中会使用的开发环境。下面是我会在这篇博文中详细介绍的几个高层次目标：

在本地机器（一个标准的 MacBookPro 笔记本电脑）上用 Pycharm 编辑我的代码

用一个强大的远程机器来训练我的模型

和我的同事们没有任何冲突地使用这台远程机器

在本地和远程机器上的 docker 容器中以开发/产品的模式来运行/调试我的 TensorFlow 代码

当我的模型在远程机器上训练的时候，把模型的性能图形化地实时显示在本地机器上


致谢

我想感谢我的实验室同伴 Chris Saam，因为他给我指明了几个我会在本文中提到的有趣的工具。

一次安装

远程机器上

因此，在做其他任何事情之前，你可能需要做这几件事情。顺便说一下，在这篇文章中我会提及在你的远程机器上（带有所有的 GPU 的附属项目）使用 super duper，在这台远程机器上你计划训练你的深度学习机器模型。

机器之心

专业的人工智能媒体与产业服务平台。

热文排行

- 日榜周榜月榜
- 1

诺基亚：你以为它死了，其实它已重回世..
- 2

它为了美国拒绝中国百亿投资！如今却求..
- 3

残酷真相：被中国人神化的德国制造
- 4

一千万存银行吃利息就行了？真事告诉你..
- 5

估值相当于BAT总和的三倍，这是世界上...
- 6

看完马云、王健林的豪宅，再看任正非，..
- 7

看完马云、王健林的豪宅，再看任正非，..
- 8

别小瞧韩国！这其实是个超出国人预料的..
- 9

扎心！朋友圈流行仅三天可见，背后是你..
- 10

《深夜食堂》遭全网差评，原来它的“老...



安装 Nvidia-docker：你需要做的第一件事情就是安装 Nvidia-docker。Docker 确实是一个很酷的工具，但是它目前并不能让你最有效地使用任何一个 NVIDIA 的 GPU 硬件或者 CUDA 驱动程序，所以不可能拿 docker 来训练你的深度模型。Nvidia-docker 为你解决了这个问题，并且看上去更像一个普通的 docker。在常规的 Docker 命令之上，它还提供了一些选项，可以让你更有效地管理你的 NVIDIA GPU 硬件。

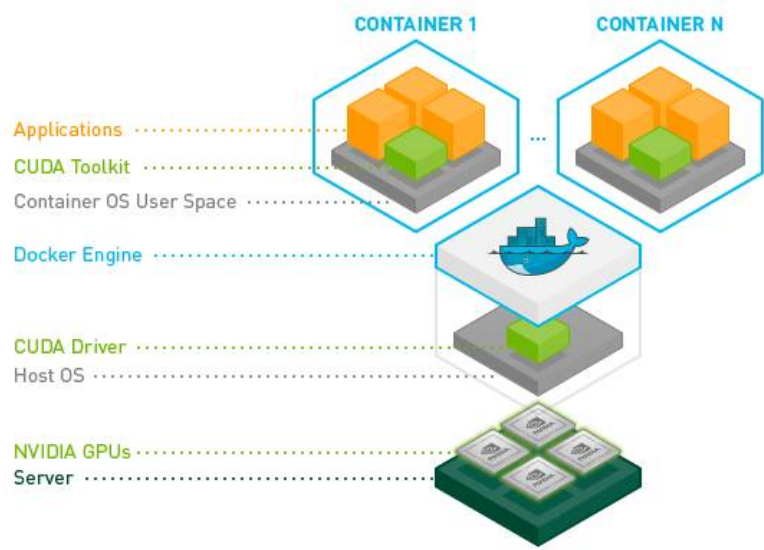


图 1: NVIDIA-Docker (由 NVIDIA-Docker 提供)

安装 Slurm：如果你计划和你的同事共享那个深度学习机器，你也许会想着安装像 SLURM 一样的工具。通过限制默认情况下可以使用的命令集，SLURM 让您对团队同事在机器上的权限拥有更好的控制，并且强制每个成员使用特定的专用 GPU/CPU 资源在「作业」环境中运行他们的代码。如果您希望避免任何因团队同事同时访问这台机器而产生的资源争夺，这确实是有用的。

把文件夹设置标准化：如果您计划和同事共享机器，就可以让成员之间的文件夹结构标准化，这也是一个好主意。我的深度学习机器的设置方式如下：

- /home/myusername 文件夹包含你自己的私有项目代码。
- /data 文件夹包含团队在项目过程中共享的数据集。
- /work 文件夹包含当前实验需要的特定数据集。这个文件夹比 /data 文件夹更低一级，但是它在训练过程中提供了更快的内存访问。

本地机器上

安装 OS X Fuse: 如果你像我一样正在使用最新版本的 OS X, 你可能会想着安装 OS X Fuse。OS X Fuse 可以让你用 SFTP/SSH 在本地 Finder 中从远程机器上挂载文件夹。或者如果你不想花费时间去挂载你的远程/home 文件夹，你可以简单地使用 GIT PUSH/PULL 在本地机器和远程机器之间传送代码，但是这样效率不高。所以在长时间运行的过程中挂载这些文件夹会替你节省大量时间。

设置一个远程的 python 解释器：在本地机器和远程机器上使用同一个 docker 映像是避免以后可能会发生的环境配置问题的另一个方法。Pycharm 有这个很酷的功能，可以让你在 docker 容器中运行代码。在 Pycharm 中进行任何设置之前，请保证你已经获取了正确 TensorFlow 的 docker 映像。在本地机器上，你可能仅仅需要以下步骤就可以获取 TensorFlow 的 docker 映像：

```
# 启动你的 docker 虚拟机
```

```
docker-machine start default
```

```
# 获取最新 TensorFlow CPU 版本的 docker 映像
```

```
docker pull gcr.io/tensorflow/tensorflow:latest
```

当你获取期望的 docker 映像之后，就去设置你的 Pycharm Project Interpreter。在 Pycharm 中，转到 PreferencesProject InterpreterAdd Remote(如下图)。当 docker 虚拟机的实例在你的本地机器上开始运行时，就需要选择 docker 配置（Docker configuration）。一旦它连接到你的 docker 虚拟机，你应该会看到你刚才获取的 TensorFlow 映像已经在可用映像的列表中了。当这个设置好之后，只要 pycharm 连接好了，你就可以开始了。

每日常规程序

本地机器上

挂载远程文件夹：你想做的第一件事情就是确保你可以访问你要在本地机器上运行的脚本。所以你要做的第一件事情就是在你的 Mac 上用 OS X Fuse 挂载 home/myusername 文件夹，并且选择性地挂载深度学习数据。你可能希望为所有这些命令起一些别名，因为它们确实有些长。

```
# 挂载你的远程 home 文件夹
```

```
sshfs -o uid=$(id -u) -o gid=$(id -g)
```

```
myusername@mydeeplearningmachine.com:/home/myusername/  
/LocalDevFolder/MountedRemoteHomeFolder
```

```
# 挂载你的远程数据文件夹 (有选择地)
```

```
sshfs -o uid=$(id -u) -o gid=$(id -g)
```

```
myusername@mydeeplearningmachine.com:/data/myusername/  
/LocalDevFolder/MountedRemoteDataFolder
```

这里使用 uid 和 gid 来映射本地和远程机器的用户和组 ID，因为这些可能会有所不同。

在本地机器上启动 docker：接下来，我们想保证 pycharm 会访问正确的库来在本地编译我们的代码。为了做到这个，仅仅需要在本地启动一个 docker 虚拟机。如果你在设置中没有改变任何地方，TensorFlow 的 CPU 映像应该已经在你的本地 docker 环境中了。

```
docker-machine start default
```

打开 pycharm，并选择你刚才挂载的 home 文件夹中的项目。转到 Project Interpreter 参数选择中，在项目解释器的可用列表中选择你之前就创建好的远程 TensorFlow 解释器，pycharm 应该能够正确地编译你的代码。这时候，你可以随时随地使用你的代码，并且改变任何你想要改变的东西。

远程机器上

Ok，你已经在 pycharm 中用一项新功能更新了你的代码，然后你希望训练/测试你的模型。

用 SSH 远程登录你的机器：你需要做的第一件事就是简单地远程登录你的深度学习机器。

```
ssh myusername@mydeeplearningmachine.com
```

运行一个 SLURM 任务: 在你进行下一步之前，请确保你的团队中没有其他成员正在运行任务。这会阻止你的任务得到它所需要的资源，所以检查一下目前有哪些任务正运行在远程机器上总会是一个不错的做法。使用 SLURM 做到这件事，只需要运行一下 `squeue` 命令即可，它会列出目前正运行在机器上的任务。如果由于某些原因你之前的某个任务仍然在运行，你可以使用 `scancel` 命令来取消它。在确定没有其他任务在运行之后，让我们开始一个新任务吧。你可以通过以下的命令来开始一个新的任务。

```
srn --pty --share --ntasks=1 --cpus-per-task=9 --mem=300G --gres=gpu:15 bash
```

`srn` 命令给出了相当多的选项来让你指定一个特定的任务需要哪些资源。在这个例子中，`cpus-per-task`、`mem` 以及 `gres` 选项让你指定这个任务分别需要的 CPU 的数量、总体内存以及 GPU 的数量。`pty` 选项只是提供一个漂亮的命令行界面。

启动 Nvidia docker：既然你已经得到了为你的任务所分配的资源，那么，启动一个 docker 容器来在正确的环境中运行你的代码吧。与使用常规的 docker 有所不同，这里我们会使用 NVIDIA-Docker 来充分地利用我们的 GPU。另外，为了充分利用你的硬件，请保证你运行的是 TensorFlow 的 GPU docker 映像而不是 docker CPU 映像。别忘了使用 `-v` 选项来在 docker 容器中挂载你的项目文件夹。当你在那个容器中以后，你就可以简单地使用常规的 `python` 命令来运行你的代码了。

```
# 启动你的容器
```

```
nvidia-docker run -v /home/myusername/MyDeepLearningProject/src -it -p 8888:8888  
gcr.io/tensorflow/tensorflow:latest-gpu /bin/bash
```

```
# 别忘记切换到你的源码文件夹
```

```
cd src
```

```
# 运行你的模型
```

```
python myDLmodel.py
```

本地机器上

启动 Tensorboard 可视化：你还差一点点就做完了。你的代码现在正在顺利地运行，然后你想着使用 `tensorboard` 去实时地看一下你的模型中的变量是如何变化的。实际上这是最简单的一部分。首先，确保你知道自己本地 docker 机对应的 IP 地址。你可以使用下面的命令来做这件事：

```
docker-machine ls
```

然后，切换到已经挂载的远程 `home` 文件夹，并启动一个 TensorFlow docker 容器。因为你已经在本地机器上启动了一个 `Tensorflow` docker 容器，所以要确保你正在启动的是 CPU 版本的 docker 容器。如上面所述，不要忘记在 docker 容器中挂载你的项目文件夹。为了在本地机器可视化正在训练的模型，你还需要用 `-p` 选项将 `Tensorboard` 使用的端口号从容器映射到你的本地机器。

```
docker run -v /LocalDevFolder/MountedRemoteHomeFolder/MyDeepLearningProject:/src -p 6006:6006 -it gcr.io/tensorflow/tensorflow:latest /bin/bash
```

一旦你进入 docker 容器，通过制定你的模型保存变量的路径 (更可能是 checkpoint 文件夹的路径) 来启动 Tensorboard：

```
tensorboard—logdir=Checkpoints/LatestCheckpointFolder
```

如果一切进展顺利，你现在需要做的就是使用你最喜欢的浏览器转到到 `http://DOCKER_MACHINE_IP:6006`，

这会显示在 Tensorboard 中显示你在模型中正在跟踪的所有变量。

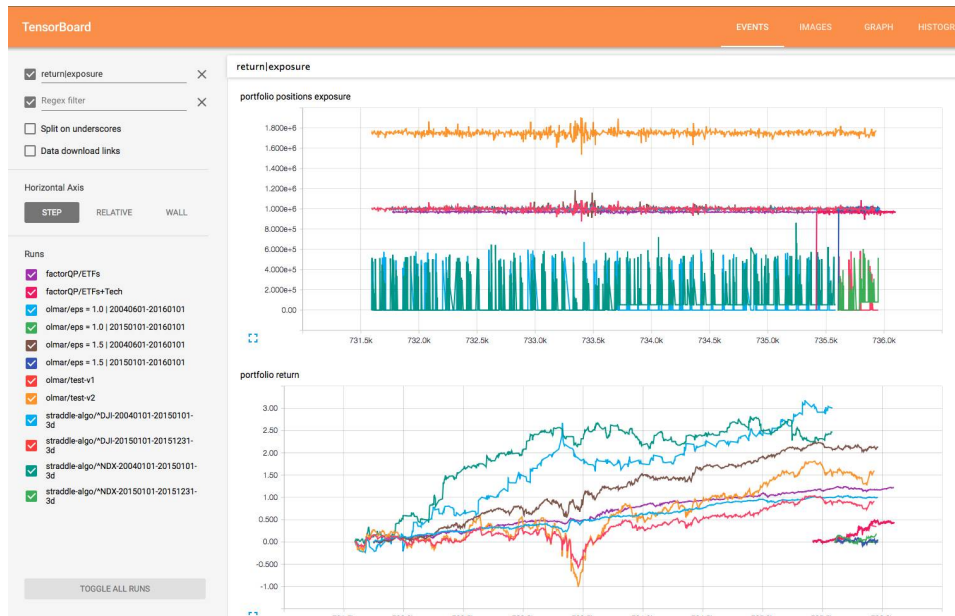


图 2.Tensorboard 可视化 (由 Jimgoo 提供)

深度学习开发环境问卷调查



这份「深度学习开发环境调查问卷」的问题涉及到开发者的基础信息（职业、研究领域等）、选择的硬件、系统、语言、框架等问题。此外，如果各位有兴趣或认为此份调查问卷缺少的问题，也可在问卷中补充。

为了感谢大家的积极参与，我们会选择其中回答最详细的 5 位调查者赠送机器之心礼品一套（包含：机器之心贴纸、马克杯、T 恤、公仔）。

提醒大家，期望获得礼品的读者不要忘了填写联系方式。

参与问卷

点击「阅读原文」填写问卷

原文链接：<http://upflow.co//8T69/blog/posts/post-2016-07-22/post.html>

本文为机器之心编译，转载请联系本公众号获得授权。

加入机器之心（全职记者/实习生）：hr@jiqizhixin.com

投稿或寻求报道：editor@jiqizhixin.com


广告商务合作：bd@jiqizhixin.com

▪

	0			
--	---	--	--	--

作者历史文章


资源 | 谷歌全attention机器翻译模型Transformer的TensorFlow实现



选自GitHub机器之心编译参与：黄小天、Smith谷歌前不久在arXiv上发表论文《AttentionIsAllYouNeed》，提出一种完全基于attent[详细]

2017年 06月19日 12:45


教程 | 斯坦福CS231n 2017最新课程：李飞飞详解深度学习的框架实现与



选自Stanford作者：李飞飞等机器之心编译参与：Smith、蒋思源斯坦福大学的课程CS231n(ConvolutionalNeuralNetworksfor[详细]

2017年 06月19日 12:45

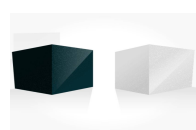
一周论文 | 基于知识图谱的问答系统关键技术研究#4



作者 | 崔万云学校 | 复旦大学博士研究方向 | 问答系统，知识图谱领域问答的基础在于领域知识图谱。对于特定领域，其高质量、结构化的知识往往是不存在，或者是极少的。本章希[详细]

2017年 06月19日 12:45


深度 | 详解首个系统性测试现实深度学习系统的白箱框架DeepXplore



选自TheForetellixBlog作者：YoavHollander机器之心编译参与：吴攀、晏奇五月份，来自哥伦比亚大学和理海大学的几位研究者的论文《Deep[详细]

2017年 06月19日 12:45

业界 | 站在锤子手机背后，小源科技用 AI 打造短信场景服务



机器之心原创作者：藤子毫无疑问，个人短信已经过时，但是，随着移动互联网的发展，企业短信却有增无减。小源科技，就抓住这个商机，用人工智能打造短信上的场景服务。20[详细]

2017年 06月18日 13:45

学界 | MIT提出生成式压缩：使用生成式模型高效压缩图像与视频数据



选自arXiv机器之心编译参与：李亚洲论文地址：
https://arxiv.org/abs/1703.01467摘要传统的图像和视频压缩算法要依赖手动调整的编码[详细]

2017年 06月18日 13:45

深度学习助力前端开发：自动生成GUI图代码（附试用地址）



选自arXiv机器之心编译参与：JaneW、蒋思源哥本哈根的一家初创公司
UlzardTechnologies训练了一个神经网络，能够把图形用户界面的截图转译成代[详细]

2017年 06月18日 13:45

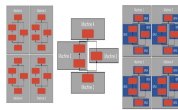
深度 | 神经网络基础：七种网络单元，四种层连接方式



选自THEASIMOVINSTITUTE作者：FJODORVANVEEN机器之心编译参与：
黄小天、李亚洲2016年9月，FjodorVanVeen写了一篇名为《[详细]

2017年 06月18日 13:45

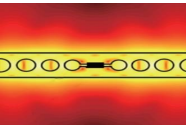
资源 | TensorFlow分布式计算机制解读：以数据并行为重



选自clindatasci作者：NeilTenenholtz机器之心编译参与：JaneW、黄小天
Tensorflow是一个为数值计算（最常见的是训练神经网络）设计[详细]

2017年 06月18日 13:45

前沿 | 面向光量子计算：MIT新研究实现室温下单光子非线性



选自MITNews作者：LarryHardesty机器之心编译参与：Smith、李泽南、吴攀
看起来，MIT最近在光计算上取得了不少的成果。前两天，机器之心报道了[详细]

2017年 06月17日 14:45