



链滴

deeplab v3+ 在 pascal voc 2012 数据集上进行训练

作者: [vcjmhg](#)

原文链接: <https://ld246.com/article/1598081794092>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

<h2 id="概述">概述</h2>

<p>前边我曾经写了一篇名为《语义分割之 deeplab v3》的文章，在那篇文章中我主要讲了 deeplab v3+ 的原理--当然主要也就是论文上边的内容。此在开始阅读本篇文章之前，建议首先阅读一下上边那篇文章。</p>

<p>本文我主要讲环境搭建以及 pascal_voc_2012 的训练以及可视化相关的内容。关于 deeplab v3+ 迁移学习部分的内容--即如何使用 deeplab v3+ 训练自己的个人数据，我后续后单独写一篇文章来。好，话不多说，我们下边正式开始。</p>

<h2 id="环境搭建">环境搭建</h2>

<p>关于环境搭建部分，其实官方给了一些说明，感兴趣可以参考官方说明《Installation》。虽然说官方给了环境搭建的 指南，但是某些细节部分它并没有具体进行说明因此如果我们只是一味按照官方说明来搭建环境的话，中间也可能会出现这样或者那样的问题。因此在该部分我会以官方的指南作为主体，结合自己在搭建环境过程中的一些细节问题，来记录 deeplab v+ 环境搭建的整个过程。</p>

<h2 id="第一步-创建python环境">第一步 创建 python 环境</h2>

<p>使用 <code>conda</code> 创建 python 环境，应该是我们复现算法的时候必须先做的一步通过 conda 创建的新环境我们事先和本地环境的隔离，可以更摆弄各种变量(:smile:)。</p>

<p>关于 <code>conda</code> 安装的部分内容，我这里就不在进行详述，网上有许多相关的教。在这里我就简单说一下如何使用 conda 创建满足 deeplab v3+ 算法运行的基础环境。</p>

<p>执行如下命令,创建 python3.6 基础环境: </p>

<pre><code class="highlight-chroma">conda create -n deeplab python=3.6</code></pre>

其中 -n 后边的参数指的是环境的名称，此处我使用的是 deeplab，你可以换成任意你喜欢的名。

等号后边的 3.6 指的是创建 python 环境的版本，此处建议使用 python3.6（因为我之前使用 3.7 总出现各种各样的错误）。

<p>执行完成之后会出现如下执行结果</p>

<p></p>

<p>输入 <code>y</code> 之后基础环境就创建完成。</p>

<p>然后通过命令</p>

<pre><code class="language-bash highlight-chroma">conda activate deeplab</code></pre>

<p>进入到刚刚创建的名字为 deeplab 的环境中。</p>

<p>执行的结果如下: </p>

<p></p>

<p>从图中我们可以看到执行完成之后，括号里边的名称由 base 变成了环境名称 (deeplab)。</p>

<p>至此 python 的基础环境搭建完成。</p>

<h2 id="第二步-安装依赖">第二步 安装依赖</h2>

Numpy

Pillow 1.0

- tf Slim (which is included in the "tensorflow/models/research/" checkout)
- Jupyter notebook
- Matplotlib
- Tensorflow (建议使用 1.14, 不推荐使用 2.0 以上版本, 否则后边总出各种问题)

安装 TensorFlow:

执行如下命令 (二选一):

```
# 使用cpu版本
pip install tensorflow==1.14
# 使用GPU版本
pip install tensorflow-gpu
```

安装 python 依赖库 (安装默认版本即可)

```
sudo apt-get install python-pil python-numpy
pip install --user jupyter
pip install --user matplotlib
pip install --user PrettyTable
```

第三步-设置python的依赖路径

由于官方给的 deeplab 代码是在一个 TensorFlow 的 model 库中, 因此它的代码使用了一些其位置的代码文件。如果不进行环境变量的设置, 在运行的过程中会报** "slim 模块或者 deeplab 模块找不到的错误" **。因此我们需要执行如下命令对代码所用的依赖路径进行设置:

```
#进入代码的research目录里边
cd ~/models/research
# 将slim以及deeplab目录添加到python的依赖环境目录中
export PYTHONPATH=$PYTHONPATH:~/models/research/deeplab
pwd
~/models/research/deeplab
```

首先执行命令之前必须要保证在代码的 `research` 目录中。

接着我们执行如下命令, 来对环境进行测试, 看环境是否已经满足要求。

```
# 进入/research/deeplab目录下
cd ~/models/research/deeplab
# 执行model_test.py
python model_test.py
```

</code></pre>

<p>如果执行过程中没有出现错误或者异常，并且最终出现如下结果，则证明基础环境搭建成功。</p>

<p></p>

<h2 id="训练与可视化">训练与可视化</h2>

<p>在前边，我们基础环境搭建完成之后，我们便可以进行训练以及可视化操作。由于 pascal voc 2012 是官方认定的测试集之一，因此官方给了一个脚本，直接运行之后，可以自动下载预训练权重，训练，评估，以及对结果进行可视化等一系列步骤。</p>

<p>执行命令如下：</p>

<pre><code class="language-bash highlight-chroma">#在tensorflow/models/research/deeplab目录下执行

bash local_test.sh

</code></pre>

<p>执行完成之后我们进入 deeplab/datasets 目录下</p>

<p></p>

<p>我们发现多了一个名为 pascal_voc_seg 的目录，该目录里面包含了 pascal 端原始数据集、预训练权重、导出的 checkpoint 文件、以及训练结果等。</p>

<p>下边我们对其常用的目录进行一些说明</p>

<p>tfrecord</p>

<table>

<thead>

<tr>

<th>目录名称</th>

<th>含义</th>

</tr>

</thead>

<tbody>

<tr>

<td>exp</td>

<td>结果文件夹，包含训练后的权重、评估的结果、可视化后的图片等</td>

</tr>

<tr>

<td>init_models</td>

<td>下载的预训练权重位置</td>

</tr>

<tr>

<td>tfrecord</td>

<td>转换成的 tfrecord 文件所在目录</td>

</tr>

<tr>

<td>VOCdevkit</td>

<td>pascal voc 2012 原本的数据集</td>

</tr>

</tbody>

</table>

<h2 id="总结">总结</h2>

<p>本文主要写了关于 deeplab v3+ 使用 pascal voc 2012 数据集进行训练的过程，并且在官方文的基础结合自己在训练过程中所踩的坑来写了这篇文章。希望能给那些训练 deeplab v3+ 模型的伙以帮助。</p>

扩展

同时我们学习一个模型之后，进行迁移学习可能也是一个必不可少的环节。因此在后续我也写一关于**如何使用 deeplabv3+ 训练自己数据集的文章**--《deeplab v3+ 训练个人数据集》，欢迎诸位阅读、批评与指正。

引用

- [https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/installation.md](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Ftensorflow%2Fmodels%2Fblob%2Fmaster%2Fresearch%2Fdeeplab%2Fg3doc%2Finstallation.md)