

# GAN实战

## 环境配置

### Windows

我们选择使用Anaconda来配置环境。

本项目中，我们所使用的Python版本为3.6，TensorFlow版本为tensorflow-gpu 1.11. cuDNN版本为7，CUDA版本为9。

为了创建相应的环境，你可以按照如下步骤进行操作。

首先，我们需要创建运行tensorflow的conda环境。

```
conda create -n tensorflow python=3.6
```

接着，我们需要进入虚拟环境之中。

```
conda activate tensorflow
```

进入虚拟环境之后，我们就可以通过pip来按照TensorFlow了。

如果你只是打算进行实验，采用CPU版本运行的话，你可以直接使用下述命令进行安装，之后便可跳过后面的内容了。如果你打算采用GPU版本运行的话，请不要执行下述命令。

```
pip install tensorflow==1.11
```

当然，只有CPU的运行效率是极低的，我们肯定还是要借助GPU的力量才能最大程度发挥计算机的性能。

所以，让我们一起来试着安装GPU版本的TensorFlow吧。

如果你已经执行了上面那条安装CPU版本TensorFlow的命令的话，你可能需要先把CPU版本的TensorFlow卸载。

```
pip uninstall tensorflow
```

那么，GPU版本TensorFlow的安装正式开始。

我们首先需要看看在目前的Anaconda环境中有哪些可用的 `cuda toolkit` 和 `cudnn`。你可以分别采用下面的命令来查询。

检查当前的cuda版本号:

```
conda search cuda
```

检查当前的cudnn版本号:

```
conda search cudnn
```

在CUDA版本为9，cuDNN版本为7的要求下，你可以从中选择对应的版本进行安装。

这里提供一种安装方案。

```
conda install cudatoolkit=9.0  
conda install cudnn=7.6.5
```

此方案不一定可行，仅供参考。

最后，我们使用pip安装tensorflow-gpu

```
pip install tensorflow-gpu==1.11
```

这一切都安装完成之后，你应该就能够正常使用TensorFlow了。

需要注意的是，如果你要在Windows环境下非Anaconda环境中使用GPU版本的TensorFlow的话，你可能需要对应安装CUDA 10才能够使用。

总之，所有安装的代码罗列如下。

```
conda create -n tensorflow-gpu python=3.6  
conda activate tensorflow-gpu  
conda install cudatoolkit=9.0  
conda install cudnn=7.6.5  
pip install tensorflow-gpu=1.11
```

## Linux

我们选择使用Anaconda来配置环境。

由于在Windows上能够使用的GPU版本的TensorFlow比较有限，所以，我们在Windows上所使用的版本与Linux上的版本稍微有些差异。

本项目中，我们所使用的Python版本为3.8，TensorFlow版本为tensorflow-gpu 1.15. 这是目前唯一在RTX3060及以上版本中还在维护的TensorFlow1.x环境。

为了创建相应的环境，你可以按照如下步骤进行操作。

首先，我们需要创建运行tensorflow的conda环境。

```
conda create -n tensorflow python=3.8
```

接着，我们需要进入虚拟环境之中。

```
conda activate tensorflow
```

进入虚拟环境之后，我们就可以通过pip来按照TensorFlow了。

如果你只是打算进行实验，采用CPU版本运行的话，你可以直接使用下述命令进行安装，之后便可跳过后面的内容了。

```
pip install tensorflow==1.15
```

当然，只有CPU的运行效率是极低的，我们肯定还是要借助GPU的力量才能最大程度发挥计算机的性能。

所以，让我们一起来试着安装GPU版本的TensorFlow吧。

如果你已经执行了上面那条安装CPU版本TensorFlow的命令的话，你可能需要先把CPU版本的TensorFlow卸载。

```
pip uninstall tensorflow
```

那么，GPU版本TensorFlow的安装正式开始。

如果你是在Ubuntu环境下的话，你可以直接在Python中安装NVIDIA相关包来实现，无需再手动安装其他版本的CUDA。

```
# To install the NVIDIA wheels for Tensorflow, install the NVIDIA wheel index
pip install nvidia-pyindex
# To install the current NVIDIA Tensorflow release
pip install --user nvidia-tensorflow[horovod]
```

GPU版本的安装命令如下。

```
pip install tensorflow-gpu==1.15
```

这一切都安装完成之后，你应该就能够正常使用TensorFlow了。

## 环境测试

我们在此目录下提供了一个 `test-env.ipynb` 的测试文件，用于检查你的TensorFlow是否安装成功。你可以打开该Jupyter文件运行以进行测试。

注：

如果有一天你不想使用该环境了，你可以使用以下命令删除该虚拟环境。

首先，我们需要退出虚拟环境。

```
conda deactivate
```

然后，将虚拟环境以及其中所有的包删除即可。

```
conda remove -n tensorflow --all
```