**TensorFlow｜0｜安装**

1. 基本环境

|  |  |
| --- | --- |
| 程序名称 | segWord.py |
| 硬件环境 | MacOSX |
| 软件环境 | Python3, Spyder, Anaconda |

1. 核心问题：安装Anaconda，安装TensorFlow CPU版本

为了使用TensorFlow框架来训练模型，我们需要了解TensorFlow的安装方式。TensorFlow的安装方式没有Theano那么直接，因为它并不是全部由Python写成的库，底层有很多C++ 乃至CUDA的代码，因此需要编译安装。

1. 解决过程：

通常安装TensorFlow分为两种情况，一种是只使用CPU，安装相对容易；另一种是使用GPU，这种情况还需要安装 CUDA 和cuDNN，情况相对复杂。我们使用TensorFlow的CPU版本的安装，把Anaconda作为Python环境，因为可以避免大量的兼容性问题。我们使用Python 3.6作为Python的基础版本，进行在MacOSX下安装TensorFlow的过程。

**3.1安装Anaconda**

Anaconda是Python的一个科学计算发行版，内置了数百个Python经常会使用的库，也包括我们做机器学习或数据挖掘的库，包括Scikit-learn、NumPy、SciPy和Pandas等，其中还有一些是TensorFlow的依赖库。我们安装这些库时，Anaconda提供了一个编译好的环境可以直接安装。Anaconda是目前最好的科学计算的Python环境，方便了安装，也提高了性能。

在Anaconda的官网上(www.continuum.io/downloads)下载Anaconda3 4.2.0版，根据操作系统下载对应版本的64位的Python 3.5版。

* 在Anaconda的下载目录执行命令，根据下载的文件替换对应的文件名。

bash Anaconda3-4.2.0-Linux-x86\_64.sh

* 接下来我们会看到安装提示，回车键确认进入下一步。然后我们会进入Anaconda的 License文档，这里按q键跳过，然后输入yes确认。接下来输入Anaconda3的安装路径，安装就自动开始了。
* 安装完成后，程序提示我们是否把Anaconda3的binary路径加入到.bashrc，添加之后，python和ipython命令就会自动使用Anaconda Python3.5的环境了。

**3.2 TensorFlow CPU版本的安装**

TensorFlow的CPU版本相对容易安装，一般分为两种情况：第一种情况，安装编译好的release版本；第二种情况，使用1.0.0-rc0分支源码编译安装，如果gcc版本比较新，或者不支持编译好的release版本，才推荐第二种安装。

第一种情况，安装编译好的release版本，我们执行下面的命令。Python的默认包管理器是pip，直接使用pip来安装TensorFlow。对于Mac或Windows 系统，可在TensorFlow的GitHub仓库上的Download and Setup 页面查看编译好的程序的地址。

export TF\_BINARY\_URL=https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.0.0rc0-cp35m-linux\_x86\_64.wh1

pip install –upgrade $TF\_BINARY\_URL

第二种情况，使用1.0.0-rc0分支的源码编译安装。

首先跟运行系统安装gcc。为了编译TensorFlow，还需要有Google自家的编译工具bazel(github.com/bazelbuild/bazel),根据其安装教程(www.bazel.io/versions/master/docs/install.html)直接安装它的v0.43 release版本即可，不需要使用最新的dev版本的功能。

在安装完gcc和bazel之后，接下来我们正式开始编译安装TensorFlow，首先先下载TensorFlow 1.0.0-rc0的源码：

wget https://gitthub.com/tensorflow/tensorflow/archive/v1.0.0-rc0.tar.gz

tar –xzvf v1.0.0-rc0.tar.gz

完成下载之后，进入TensorFlow代码仓库的目录，然后执行下面的命令进行配置：

cd tensorflow-1.0.0-rc0

./configure

接下来的输出要选择python路径，确保时anaconda的python路径即可：

Please specify the location of python. [Default is /home/wenjian/anaconda3/bin/python]:

这里选择CPU编译优化选项，默认的-march=native将选择本地CPU能支持的最佳配置，比如SSE4.2、 AVX等。一般选择默认值。

Please specify optimization flags to use during compilation [Defaut is –march=native]:

选择是否使用jemalloc作为默认的malloc实现，建议选择默认设置。

Do you wish to use jemalloc as the malloc implementation?(linux only) [Y/n]

然后它会让我们选择是否开启对Google Cloud Platform的支持，这个在国内一般是访问不到的，有需要的用户可以选择支持，通常选N即可：

Do you wish to build TensorFlow with Google Cloud Platform support? [y/N]

它会询问是否需要支持Hadoop File System，如果有读取HDFS数据的需求，请选y选项，否则就选默认的N即可：

Do you wish to build TensorFlow with Hadoop File System support? [y/N]

选择是否开启XLA JIT编译编译功能支持。这里XLA 是TensorFlow目前实验性的JIT(Just In Time)、AOT(Ahead Of Time)编译优化功能。

Do you wish to build TensorFlow with the XLA just-in-time compiler (experimental)? [y/N]

然后它会让我们选择python的library卢靖，这里选择anaconda的路径：

Please input the desired Python library path to use. Default is [/home/wenjian/anaconda3/lib/python3.5/site-packages]

接着选择不需要使用GPU，即OpenCL和CUDA全部选N：

Do you wish to build TensorFlow with OpenCL support? [y/N]

Do you wish to build TensorFlow with CUDA support? [y/N]

之后需要下载一些依赖库文件，完成后confihure就顺利结束了，接下来使用编译命令执行编译：

bazel build –copt=-march=native –c opt //tensorflow/tools/pip\_package:build\_pip\_package

编译结束后，使用下面的命令生成pip安装包：

bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package/tmp/tensorflow\_pkg

最后，使用pip命令安装TensorFlow：

pip install /tmp/tensorflow\_pkg/tensorflow-1.0.0rc0-cp35-cp35m-linux\_x86\_64.wh1

4.结果

成功安装TensorFlow深度学习框架，可以用于训练模型，例如手写数字模型(MNIST)等。