

深度学习框架Keras的安装

Keras是Python语言中基于原始深度学习框架Tensorflow或Theano的封装框架。那么如果准备使用Keras首先必须准备安装Tensorflow或Theano

[Keras中文文档地址](#)

0. CPU运行版本的安装

如果需要直接安装（即使用CPU实现程序运算），仅仅需要在安装好的Python环境下，在终端或者命令行下，输入：

```
pip install theano;  
#pip install tensorflow;  
pip install keras
```

即可完成keras的安装

之后可以验证keras是否安装成功,在命令行中输入Python命令进入Python变成命令行环境：

```
>>>import keras  
Using Theano(Tensorflow) backend.  
>>>
```

那么Keras就已经成功安装了

1. Windows环境下GPU运行版本的安装

Windows本身不具备非常良好的开发环境，所以我们需要准备如下几个要素来驱动GPU运行Keras：

- **基础工具 Microsoft Visual Studio 2010 - 2013**

（2015版本仅仅支持CUDA8.0，不建议安装）

这里推荐到[MSDN我告诉你](#)下载各个版本的，在关闭360等杀毒软件后，进行安装，软件将自动配置环境，不需要更多设置。

- **Python环境 - 推荐Anaconda**

Anaconda是Python众多发行版中非常适用于科学计算的版本，里面已经集成了很多优秀的科学计算Python库。

对于搞科学计算与深度学习的朋友们，建议安装Anaconda2.7版本，如果需要做文本处理，建议3.5

下载地址：[Anaconda](#)

同样关闭360等杀毒软件的屏蔽软件，安装时同意默认Anaconda作为Anaconda作为默认

python路径，那么环境变量不需要再次配置了。

- 关键的gcc/g++编译器

gcc/g++是Windows环境与Linux环境非常大的一个差别点。

然而Keras采用GPU进行编译，gcc/g++是必不可少的，这里提供两种解决方案：

- Mingw

Anaconda官方库中集成了软件包Mingw，里面包含了gcc/g++等编译工具。

打开命令行直接输入：

```
conda install mingw libpython
```

- MSYS2

一部分读者自己本身已经具有了Python环境，再安装Anaconda会造成很大的不便，那么本文推荐安装[MSYS2](#)，网站上有详细的如何安装的说明，本文不再赘述。

在安装好后确认安装目录中存在mingw文件夹及文件夹中的各类文件。

- 核心工具 CUDA Toolkit

[CUDA Toolkit](#)是NVIDIA公司面向GPU编程提供的基础工具包，也是驱动显卡计算的核心技术工具。

该工具目前仅仅面向NVIDIA公司所生产的各类显卡，不支持AMD公司或英特尔公司的显卡产品，如果没有NVIDIA公司的显卡，那么只能使用基于CPU版本的Keras深度学习框架。

目前NVIDIA显卡中支持CUDA包含GeForce\TESLA\QUADRO三个系列，市面上常见的系列显卡GTX、GT、M开头的都支持CUDA，包括笔记本类显卡。

在近期上市的GeForce GTX 1080、GeForce GTX 1070仅仅只能使用[CUDA Toolkit 8.0](#)，建议采用7.5版本，按照自己的系统进行选择性的下载。

在下载之后，按照步骤安装，**不建议新手修改安装目录**，同上，环境不需要配置，安装程序会自动配置好。

- 底层框架Theano/Tensorflow也Keras

这里不加赘述。

```
pip install theano;  
#pip install tensorflow;  
pip install keras
```

或者想要加速开发版本，用（前提是你有git：conda install git）

```
pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git
```

- （可选）加速库CuDNN

从[官网下载](#)需要注册账号申请，两三天批准。网盘搜索一般也能找到最新版。

Windows目前就是cudnn-7.0-win-x64-v5.0-prod.zip。

下载解压出来是名为cuda的文件夹，里面有bin、include、lib，将三个文件夹复制到安装CUDA的地方覆盖对应文件夹，默认文件夹在：

```
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA
```

- 重中之重：环境配置

在我的电脑上右键->属性->高级->环境变量->系统变量中的path，添加

```
C:\Anaconda2;C:\Anaconda2\Scripts;C:\Anaconda2\MinGW  
\bin;C:\Anaconda2\MinGW\x86_64-w64-mingw32\lib;
```

注意 本文将Anaconda安装至C盘根目录，根据自己的情况进行修改；另外在之前安装gcc/g++时采用MSYS2方式安装的，修改并重新定位MinGW文件夹，并做相应修改。

之后并新建变量PYTHONPATH，并添加

```
C:\Anaconda2\Lib\site-packages\theano;
```

在用户目录，也就是C:\Users\当前用户名\，新建.theanorc.txt。这个路径可以通过修改Theano的configparser.py来改变。Theano装在Anaconda\Lib\site-packages里 .theanorc.txt的内容：

```
[global]  
openmp=False  
device = gpu  
optimizer_including=cudnn #不用cudnn的话就不要这句，实际上不用加，只要刚刚配置到位就行  
floatX = float32  
allow_input_downcast=True  
[lib]  
cnmem = 0.8  
[blas]  
ldflags=  
[gcc]  
cxxflags=-ID:\Anaconda2\MinGW  
[nvcc]  
fastmath = True  
--flags=-LD:\Anaconda2\libs #改成自己装的目录  
--compiler_bindir=D:\Microsoft Visual Studio 12.0\VC\bin #改成自己装的目录  
#最后记得把汉字全删了
```

至此安装完成，转入本文结尾《GPU加速测试》部分，验证安装效果。

[Keras中文文档地址](#)

2. Linux-Ubuntu环境下GPU运行版本的安装

进入Linux系统安装，相对来说容易多了。

对于Ubuntu发行版，本文建议安装Ubuntu 14.04，部分使用帕斯卡系列显卡和CUDA Toolkit8.0的读者，可以使用最新的Ubuntu 16.04发行版。其他发行版中，尤其使用了Ubuntu 15.10的读者，在使用CUDA时会出现，gcc版本过高无法编译的情况，解决办法这里不再详细赘述，可以联系笔者，联系方式在文末。

本文以Ubuntu 14.04作为例子。

- 系统初始配置与依赖库的安装

在系统安装好之后，在 系统设置->软件更新 中更换aliyun镜像源，之后使用快捷键Ctrl+Alt+T打开终端，输入：

```
sudo apt update;
sudo apt upgrade #静待系统更新完成
```

接着安装依赖库：

```
sudo apt install -y python-dev python-pip python-nose gcc g++ git gfortran;
sudo apt install -y libopenblas-dev liblapack-dev libatlas-base-dev;
sudo apt install -y python-numpy python-scipy #Anaconda用户不需要安装
```

部分读者依然喜爱使用Anaconda发行版，可以在该系统下安装，地址依然是官网，按照说明可以自动安装。

- **CUDA Toolkit的安装**

[CUDA Toolkit](#)选择合适的版本进行下载，强烈建议使用.deb格式的下载包。下载完后，终端cd至相应文件夹，输入：

```
sudo dpkg -i cuda-repo-ubuntu1404-7-5-local_7.5-18_amd64.deb;
sudo apt-get update;
sudo apt install cuda
```

安装完毕后，输入：

```
echo 'export PATH=/usr/local/cuda-7.5/bin:$PATH' >> ~/.bashrc;
echo 'export LD_LIBRARY_PATH=/usr/local/cuda-7.5/lib64:$LD_LIBRARY_PATH' >>
~/.bashrc;
source ~/.bashrc
```

CUDA环境也配置完了

验证是否配置成功，可以尝试输入：

```
nvcc -V
```

可以查看到输出nvcc编译器的版本

- **Theano以及Keras的安装**

```
pip install theano;
#pip install tensorflow;
pip install keras
```

- **Theano的配置**

在终端中输入：

```
sudo gedit ~/.theanorc
```

打开文件添加：

```
[global]
floatX=float32
device=gpu
[lib]
cnmem = 0.8
[blas]
ldflags = -lopenblas
[nvcc]
fastmath = True
```

保存退出，至此安装完成，转入本文结尾《GPU加速测试》部分，验证安装效果。

[Keras中文文档地址](#)

3. GPU加速测试

这一部分不分系统，只要配置正确都可以完成。

- 环境测试

在命令行中进入Python环境，输入：

```
import theano #采用tensorflow作为底层的不用使用
```

会出现一系列信息，包括显卡型号、浮点数类型、是否采用CNmem和cuDNN（如果使用了的话）等等，那么恭喜你，环境彻底配置成功。

如果使用了Windows系统的读者，电脑上可能会出现，debug的字样，这是第一次使用，在编译生成运行库，属于正常现象。

- 加速库测试
Python环境下输入：

```
import numpy
id(numpy.dot) == id(numpy.core.multiarray.dot)
```

如果得到的结果为False，说明你的除了gpu加速还得到了数学库blas加速，按照教程顺序配置的Linux用户是一定可以得到False结果的；Windows用户得到True也没有关系，因为Anaconda中已经内置了MKL加速库，如果想使用OPENblas可以按照文末的联系方式联系我。

- 速度测试
新建一个文件test.py，内容为：

```

from theano import function, config, shared, sandbox
import theano.tensor as T
import numpy
import time

vlen = 10 * 30 * 768 # 10 x #cores x # threads per core #这里可以加一两个0, 多
测试一下, 记得去掉汉字
iters = 1000

rng = numpy.random.RandomState(22)
x = shared(numpy.asarray(rng.rand(vlen), config.floatX))
f = function([], T.exp(x))
print(f.maker.fgraph.toposort())
t0 = time.time()
for i in xrange(iters):
    r = f()
t1 = time.time()
print("Looping %d times took %f seconds" % (iters, t1 - t0))
print("Result is %s" % (r,))
if numpy.any([isinstance(x.op, T.Elemwise) for x in
f.maker.fgraph.toposort()]):
    print('Used the cpu')
else:
    print('Used the gpu')

```

在GTX 970显卡下, 输出结果大概是0.21秒, 在一百倍运算量下19秒, 可以进行对比。
想比较速度的话, 可以将刚刚配置中.theanorc/.theanorc.txt文件中的改成

```

[global]
device = cpu

```

理论上, 相比较主频为3.3GHz的CPU, 加速比应该是75倍, 但不同的ssd和内存限制了IO接口传输速度。

- Keras中mnist数据集测试
下载Keras开发包

```

git clone https://github.com/fchollet/keras.git
cd keras/examples/
python mnist_mlp.py

```

程序无错进行, 至此, keras安装完成。

[Keras中文文档地址](#)

声明与联系方式

由于作者水平和研究方向所限，无法对所有模块都非常精通，因此文档中不可避免的会出现各种错误、疏漏和不足之处。如果您在使用过程中有任何意见、建议和疑问，欢迎发送邮件到moyan_work@foxmail.com与中文文档作者取得联系。

本安装教程文档的全部版权归于Keras中文文档作者moyan所有。

本教程不得用于任何形式的商业用途，如果需要转载请与作者或中文文档作者联系，如果发现未经允许复制转载，将保留追求其法律责任的权利。

作者：[SCP-173](#)

E-mail：scp173.cool@gmail.com

如果您需要及时得到指导帮助，可以加微信：SCP-173，酌情打赏即可