# [Win7配置CUDA并搭建基于Theano框架的GPU加速环境](http://blog.csdn.net/m624197265/article/details/45700619)

分类： [环境配置](http://blog.csdn.net/m624197265/article/category/3215077) [GPU加速](http://blog.csdn.net/m624197265/article/category/3215079) [CUDA](http://blog.csdn.net/m624197265/article/category/3215081)2015-05-13 22:00 181人阅读 [评论](http://blog.csdn.net/m624197265/article/details/45700619#comments)(2) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/m624197265/article/details/45700619#report)

[win7](http://www.csdn.net/tag/win7)[64位](http://www.csdn.net/tag/64%e4%bd%8d)[cuda](http://www.csdn.net/tag/cuda)[gpu](http://www.csdn.net/tag/gpu)[theano](http://www.csdn.net/tag/theano)

目录[(?)[+]](http://blog.csdn.net/m624197265/article/details/45700619)

        今天终于买回来了一块算是较好的显卡，渴望已久的GPU加速终于实现了，于是进行了下面的一系列配置。

## ****一、   硬件环境信息****

        台式电脑：ThinkCentre M8400t（i7 3770/32GB/1TB），CPU 3.4GH 四核8线程

        显卡：NVIDIA GEFORCE GTX 970

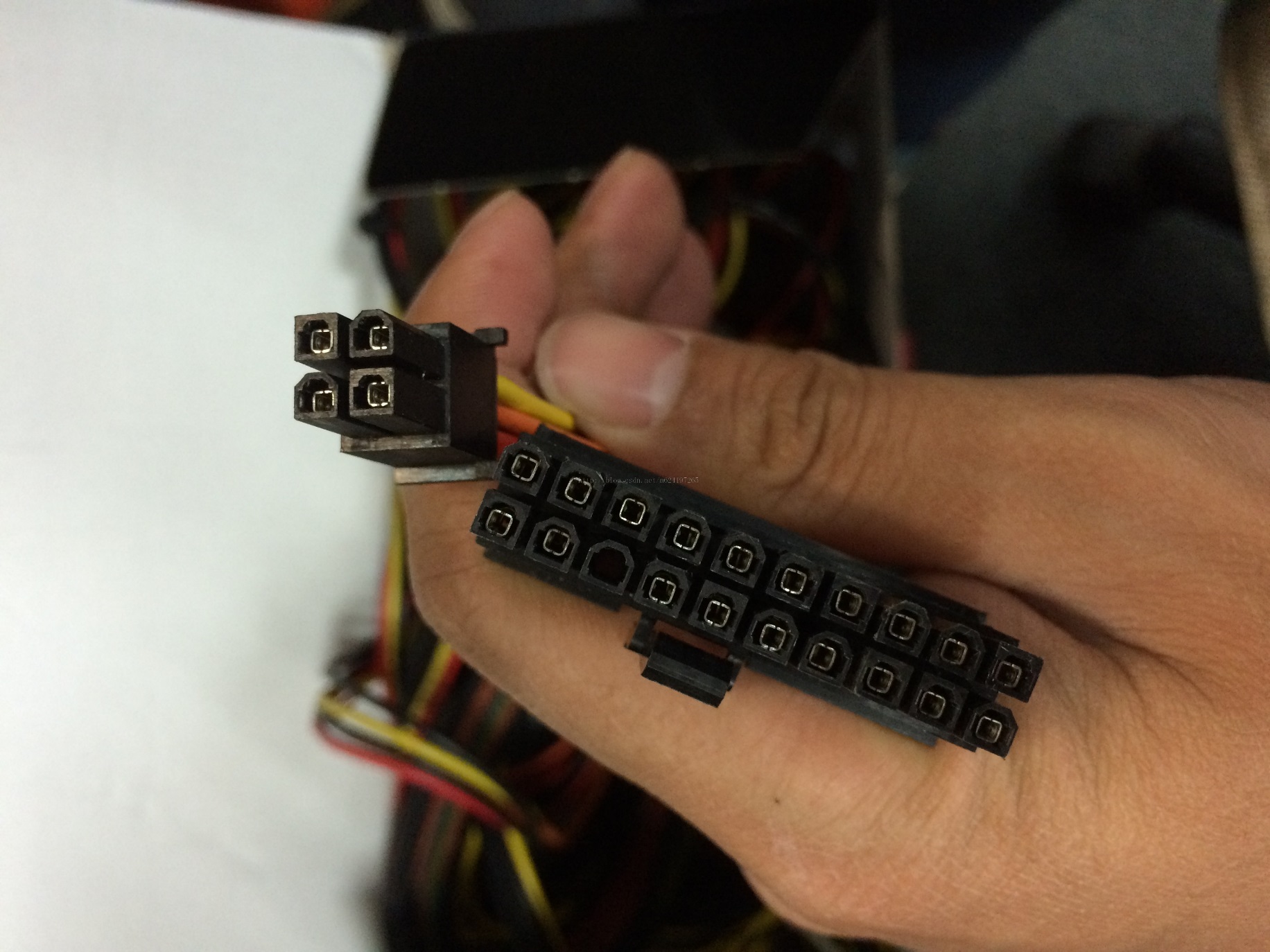
**注意**：机器原来的显卡是AMD Radeon HD 7450，不支持CUDA加速，所以这次花了点小血本买了一块NVIDIA的显卡（如图A），由于新购的NVIDIA GEFORCE GTX 970显卡功率有W，需要单独供电，而原来的主机电源功率比较弱，所以还一并换了一个的Antec 700W超静音主机电源（如图B）。拿到电源与显卡后，突然发现ThinkCentre M8400t这款电脑主板也是比较鸡肋，居然是Intel Q77（这款主板的电源线是14P的，如图D），而新电源只支持20P或24P（如图C），顿时石化了。跑遍了中关村电子市场都没有买到20P或24P转14P的转接线，最后在万能的淘宝上面花了15¥买到了一根转接线，为主机成功装上了电源和显卡。



图A NVIDIA GEFORCE GTX 970显卡



图B Antec 700W超静音主机电源



图C Antec电源线20P或24P



图D 装好后的整机内部

## ****二、   软件环境信息****

        【1】**操作系统**：Windows 7 （64位）Professional旗舰版。

        【2】**Python环境及相关依赖包**：Anaconda-1.9.2-Windows-x86\_64。

        【3】**Theano包安装**：直接利用pipinstall theano命令即可完成安装。

        【4】**检查电脑显卡是否支持GPU加速**。如果显卡不支持GPU加速，到此就没有必要进行下面两步配置操作了。

        【5】**C++环境安装**：直接安装微软的VisualStudio环境。

        【6】**并行计算架构安装**：直接安装CUDA。

## ****三、   详细配置流程****

        GPU加速环境的配置流程可以简单总结为：**配置开始**→**操作系统安装**→**Python环境及相关依赖包**→**Theano包安装**→**检查显卡是否支持GPU加速**→**C++环境安装**→**并行计算架构安装**→**配置结束**，总共可计6个主要步骤。下面就对每个步骤进行详细描述。

**【1】     操作系统安装**

        一般情况下大家都已经安装了操作系统，这一步基本都不用多大麻烦，这里的配置过程描述均是基于Windows操作系统。但须得注意自己的操作系统是32位还是64位，以下【2】—【6】步骤的配置过程所对应的软件最好是与操作系统的位数保持一致，对于没有保持一致的配置过程没有亲自尝试过。若有同学能够配置成功可以告诉一下。以下的配置过程均是采用的**Windows7（64位）Professional旗舰版**操作系统。

【2】     **Python环境及相关依赖包安装**

        目前Python的版本主要分为Python 2.x和Python 3.x，至于这两种版本的区别在此处就不用过多介绍，可以找度娘。

        由于Theano采用的Pyhton 2.x进行编写的，所以在这里安装Python环境的时候**建议安装Python2.x**，至于具体的何种Python 2.x，建议安装较高的2.x即可。除了安装最基本的Python环境，还应该安装一些其他的依赖包以供Theano使用。在安装Python环境极其相关依赖包的过程中，如果一个一个的安装其他依赖包，这样显得较为麻烦，这里**建议安装一个Python集成开发包**，Python集成开发包种类较多，我这里安装的Anaconda（版本为**Anaconda-1.9.2-Windows-x86\_64**，下载地址：<https://repo.continuum.io/archive/>，该地址包含了所有的Anaconda历史版本），用起来没有任何问题，对Theano的支持很好。在安装Anaconda时建议直接安装在盘的根目录下，如D:\Anaconda。

        在安装完Python后，打开控制台，输入>>**python**命令，即可看到所安装Python版本及其相关信息，这表明Python已经正确安装。如果没有看到任何信息，怎添加环境变量path = D:\Anaconda;D:\Anaconda\Scripts即可。

**注意：**本人第一次安装了个Anaconda-2.2.0（当时最新版），发现没有MinGW，然后自己另外单独安装了MinGW，但是配置了很久都没有配置成功（如有配置成功的可以告诉我），于是换成了**Anaconda-1.9.2**版本（**自带MinGW**）。因为theano的运行需要gcc，所以才会去折腾安装一个MinGW，如果在Linux平台上就不用折腾这一番了。有了Anaconda，其里面的Python包已经足够支持theano的运行了。

【3】     **深度学习框架Theano安装**

        在确保第【2】步中Python环境已经正确配置后就可以安装Theano了。由于Anaconda中的Python依赖包集中已经有了pip工具包，因此在安装Theano时，只需保证网络畅通的前提下，打开**控制台**，然后**输入>>pipinstall theano**命令并**回车**，然后稍等片刻即可自动安装上最新版的Theano深度学习框架。这时可到D:\Anaconda\Lib\site-packages目录下查看是否已经有theano和Theano-0.7.0-py2.7.egg-info两个文件夹，若有则表明已经安装成功。一般情况下，这一步不会出错，很容易操作。

        安装结束后，检查当前用户目录下是否有.theanorc.txt文件（这是theano的一些初始设置信息文件，如不知道当前用户目录位置，可以打开控制台窗口，该窗口里面第一个命令提示行所示的目录位置就是当前用户目录位置，如图1所示中的C:\Users\Mag即为当前用户目录），如果没有可以手动创建一个，然后在文件里面写入下面内容：

        [global]

        openmp = False

        [blas]

        ldflags =

        [gcc]

        cxxflags = -IC:\MinGW\include

        输入完毕后保存.theanorc.txt文件即可。

        现在检查theano是否配置成功，只需打开控制台，然后输入：python回车，再输入import theano回车，若没有其他错误信息输出这表明theano配置正确。如下图1所示。

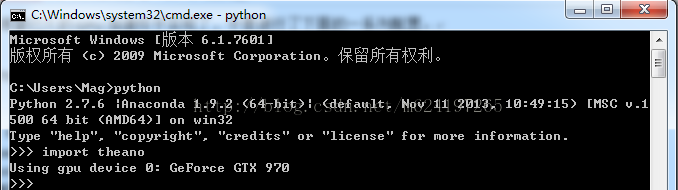


图1 theano配置正确运行截图

        注意：我这里由于已经配置好了CUDA，所以输入import theano回车后有信息输出，在没有配置CUDA之前，输入import theano回车后是没有信息输出的。

至此，大家就可以利用theano框架来编写深度学习的代码，**在CPU上面运行**了。

【4】     **检查电脑显卡是否支持GPU加速**

        这一步至关重要，在确定了电脑显卡支持GPU加速后，再进行后面【5】【6】步操作，不然费时费力而且还没有结果。就目前而言，似乎只有NVIDIA的GF8级别以上的显卡才能支持physx物理加速（即GPU加速，这里所说的GPU加速均默认是CUDA编程），ATI的显卡不支持。

        如何确认自己电脑的GPU是否支持CUDA编程？在设备管理器中找到显示适配器（Display adapters)，找到自己电脑的显卡型号，然后到[http://www.nvidia.com/object/cuda\_gpus.html列表（如图2](http://www.nvidia.com/object/cuda_gpus.html%E5%88%97%E8%A1%A8%EF%BC%88%E5%A6%82%E5%9B%BE2)所示）中进行比对，若在列表中则支持CUDA编程，否则就利用Theano框架所编写深度网络就只能在CPU上面运行了。

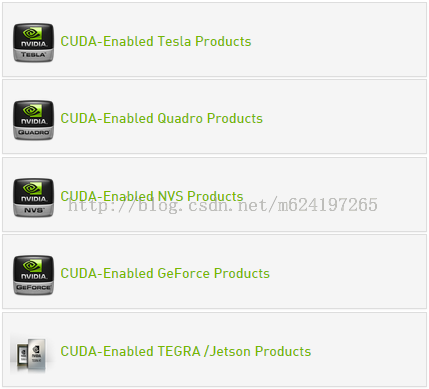


图2 支持CUDA编程的显卡型号

【5】     **C++环境安装**

        在电脑显卡支持GPU加速（即CUDA编程）后，就可以进行【5】和【6】步操作了。

        C++环境的安装，建议直接安装一套微软的Visual Studio环境，建议Visual Studio 2010或更高版本。本人安装Visual Studio 2010与Microsoft Visual Studio 2012两个版本，均能够支持CUDA编程。不过MicrosoftVisual Studio 2012，这一版本的VS体积太过于庞大了，完全安装后有10G+。如果大家不从事其他语言开发，可以考虑在安装VS的时候只选择C++，其他语言如VB，C#，F#都可以不用安装（已试过，没有任何问题，因为CUDA只需C++的支持），这样就能够减少硬盘的使用量。至于VS的具体安装过程这里就不再论述了，有问题找度娘。

【6】     **并行计算架构CUDA的安装**

        在安装CUDA之前最好检查一下自己电脑的显卡驱动版本，版本过老的话，需要更新。有时候安装完毕CUDA之后，运行CUDA所提供的案例程序中的deviceQuery会出错，请优先考虑显卡驱动的问题。这里建议安装NIVDIA官方发布的针对自己显卡型号的驱动。安装CUDA的详细步骤如下：

        （1）     下载NVIDIA  CUDA toolkit，SDK等软件包，由于CUDA 5.0以上版本已经将ToolKit和SDK等整合在一起了，因此只需下载一个安装文件即可。在显卡型号允许的条件下，建议下载CUDA 5.0以上版本，这样比较省事。本人安装的是CUDA 6.5版本。CUDA各版本下载地址：<https://developer.nvidia.com/cuda-toolkit-archive>。注意，下载的时候有些版本需要区分desktop（台式机）和 notebook（笔记本）。

        （2）      下载好了后就可以直接安装了，CUDA的默认安装目录为：C:\Program Files\NVIDIA GPU Computing Toolkit\ 这里推荐使用默认的安装路径，不需更改。安装过程和普通软件没有什么区别。在安装结束时将会出现一个界面（如图3），这表明已经成功安装，从图3可以看出似乎CUDA 6.5至少都需要Visual Studio 2010的C++环境。

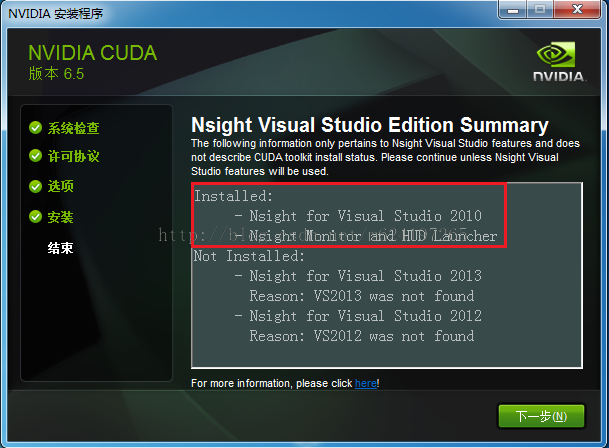


图3 CUDA6.5成功安装结束界面

        （3）     检查是否安装正确。在安装结束后，你会发现在系统环境里面新添加了两个环境变量：CUDA\_PATH\_V5\_0和CUDA\_PATH。

现在，打开cmd控制台命令行，输入命令nvcc –V回车（注意是大写V哟）就可以参考版本信息，如果安装正确则出现图4所示的结果。

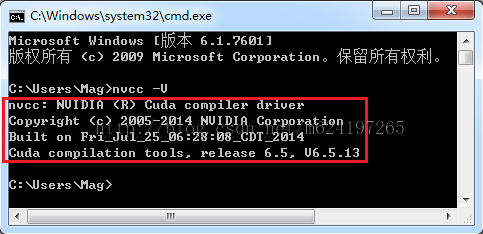


图4 nvcc –V命令执行结果

        在测试CUDA是否能够正确运行时，进入到下面所述目录下，双击运行其目录下的.exe程序即可。不同的版本估计运行案例的路径有所差异，不过可以通过搜索关键字“CUDASamples”或者“particles.exe”就能够找到案例所在目录。

**<1> WindowsXP平台:**

                C:\Documents and Settings\All Users\ApplicationData\NVIDIA Corporation\CUDASamples\v5.0\C\bin\win32\Release

**<2> WindowsVista, Windows 7, Windows 8, Windows Server 2003, and Windows Server2008平台:**

                C:\Program Files\NVIDIACorporation\Installer2\CUDASamples\_6.5\bin\win64\Release

        如果CUDA安装正确，则所有案例都是能够运行的。在本人电脑上，大部分案例都是一闪而过，只有particles.exe这个案例才能看得到绚丽的动画，如图5所示。

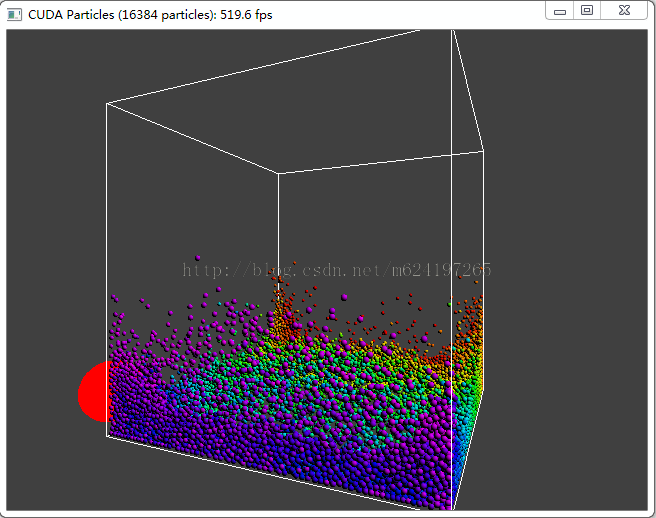


图5 particles.exe案例运行结果

        （4）     如果案例运行正确，可以考虑手动配置环境变量。在系统环境变量中新建如下项（针对win32平台上的CUDA 5.0）：

        CUDA\_SDK\_PATH = C:\ProgramData\NVIDIA Corporation\CUDASamples\v5.0\common

        CUDA\_LIB\_PATH = %CUDA\_PATH%\lib\Win32

        CUDA\_BIN\_PATH = %CUDA\_PATH%\bin

        CUDA\_SDK\_LIB\_PATH = %CUDA\_SDK\_PATH%\common\lib\Win32

        CUDA\_SDK\_BIN\_PATH = %CUDA\_SDK\_PATH%\bin\Win32

        然后在系统环境变量Path 后添加内容：;%CUDA\_LIB\_PATH%;%CUDA\_BIN\_PATH%;%CUDA\_SDK\_LIB\_PATH%;%CUDA\_SDK\_BIN\_PATH%

        当然不同版本的CUDA，估计路径有所差异，上述配置信息是win32版本的CUDA，仅仅作为参考。本人在配置过程中，没有进行这一步手动配置环境变量。

        （5）     CUDA配置结束。若以上案例都运行正常，说明CUDA安装成功。现在就可以利用Theano框架编写深度学习代码，并将代码进行GPU加速了。

**【7】     测试Theano框架是否可以利用所安装的CUDA6.5进行GPU加速**

        在前面第【2】步中，已经通过安装Anaconda将python环境装好，其中Anaconda本身自带了一个称作Spyder的图形界面IDE（如图6所示）。在进行Theano GPU加速测试之前，还得需要配置一下.theanorc.txt文件里面的信息，**红色字体**信息是为了实现GPU加速而**增加的配置信息**，如下：

        [global]

        openmp = False

        device = gpu

        floatX = float32

        allow\_input\_downcast=True

        [blas]

        ldflags =

        [gcc]

        cxxflags = -IC:\MinGW\include

        [nvcc]

        flags = -LC:\Anaconda\libs    #此处一定要和你安装的Python的路径保持一致

        compiler\_bindir = D:\Program Files(x86)\Microsoft Visual Studio 10.0\VC\bin   #此处一定要和你安装的VS的路径保持一致

        fastmath = True

        flags =  -arch=sm\_30    #这里最初设置为sm\_52报错，后设为30或50都没问题，不清楚这是干啥的，有谁知道了可以告诉我

        现在可以启动Spyder，输入表1中的Theano GPU加速测试案例代码并运行，将会出现图7所示的运行结果，其中有一条“Used the gpu”信息输出，表明Theano GPU加速测试成功。

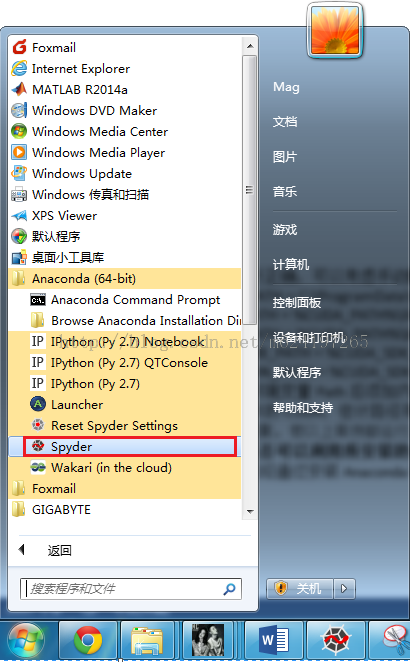


图6 Anaconda自带Python图形界面IDE

|  |
| --- |
| **fromtheanoimport** function, config, shared, sandbox  **importtheano.tensorasT**  **importnumpy**  **importtime**    vlen =10\*30\*768 # 10 x #cores x # threads per core  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'% iters, t1- t0,'seconds'  **print**'Result is', 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' |

表1 Theano GPU加速测试代码

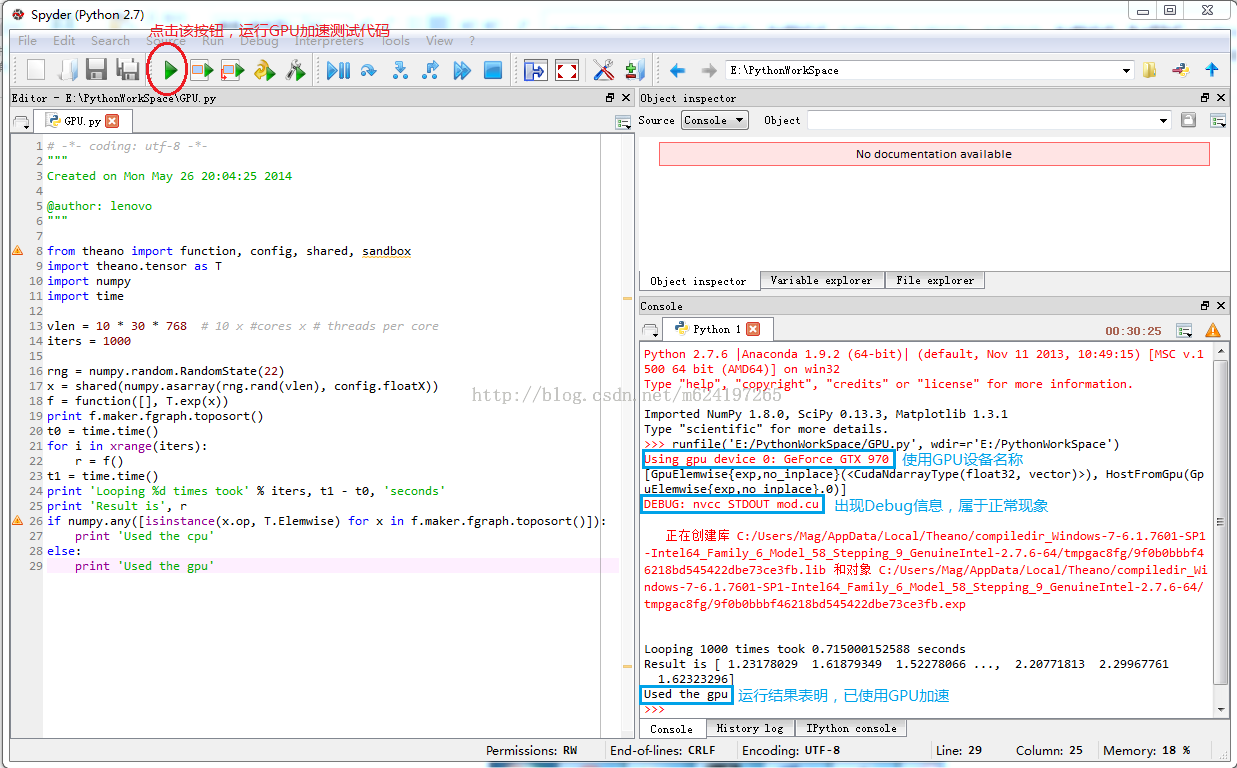


图7 Spyder IDE

【8】     到此，基于CUDA的Theano GPU加速环境已经配置成功。

## ****四、   测试GPU加速与没有GPU加速时的算法执行速度****

        利用Theano框架构建一个深度网络，其层次结构为：Layers = [784, 1050, 4901, 500, 10]；训练配置：Input\_Layer =784，Out\_Put = 10，Batch\_Size = 500，Epoch = 150，网络每一层详细训练速度对比如表2所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 训练类型 | 层次 | 无GPU加速（单位：分钟） | 有GPU加速（单位：分钟） | 加速比（单位：倍） |
| 预训练（Pre-training） | 1050 | 55.5 | 3.8 | 14.6 |
| 4901 | 196.7 | 8.1 | 24.3 |
| 500 | 252.8 | 19.6 | 12.9 |
| 全局调优（Finetunning） | [784, 1050, 4901, 500, 10] | 182.1 | 6.3 | 28.9 |
| 总耗时间 | | 687.1 | 37 | 18.6 |

表2 GPU加速对比

## 五、   总结

        GPU加速果然名不虚传，终于找到提升算法速度的又一大神器啊….