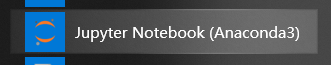
**1、首先建议卸掉原来自己从python官网装的python**

避免后面一些不必要的麻烦。

**2、然后下载anaconda（官网下载） 进行安装。**

一方面，因为他可以自动安装好一些常用的必要的工具包，不用手动去pip安装，而且安装好之后使用命令”conda install XXX”安装包的时候，他好像可以自动安装目标的依赖包。

另一方面，他还会装上“jupyter”,这是个很方便的东西，浏览器运行终端，代码模块化执行，和ipython有的一拼。下图 jupyter



安装过程没什么好说的，一路无脑默认安装。只是经过我的测试，在安装的时候，将其添加到path环境变量会好用很多（比如cmd键入python ；普通cmd就可以代替他专门的anaconda prompt，这样的好处显然是随时可以调出CMD，而不用可以去开始菜单找这个玩意）。虽然官方他并不建议这样做。下图 anaconda prompt



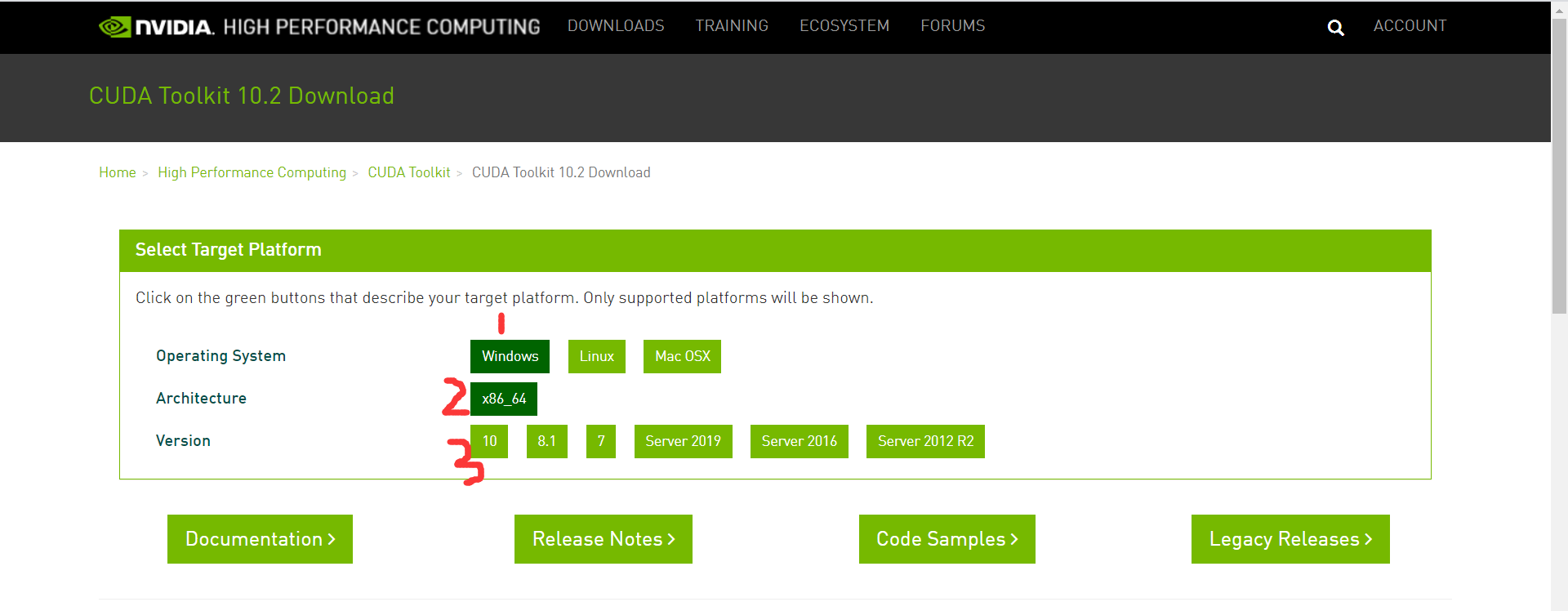
更多好处用法自行查阅资料。

**3、VStools的安装**

注意，这并不意味着我们需要使用Visual Studio，我们只是需要他提供的VC++，去官网下载 <https://www.microsoft.com/en-us/download/details.aspx?id=48145> 这个是2015版本，我这边现在在跑的是2014的。

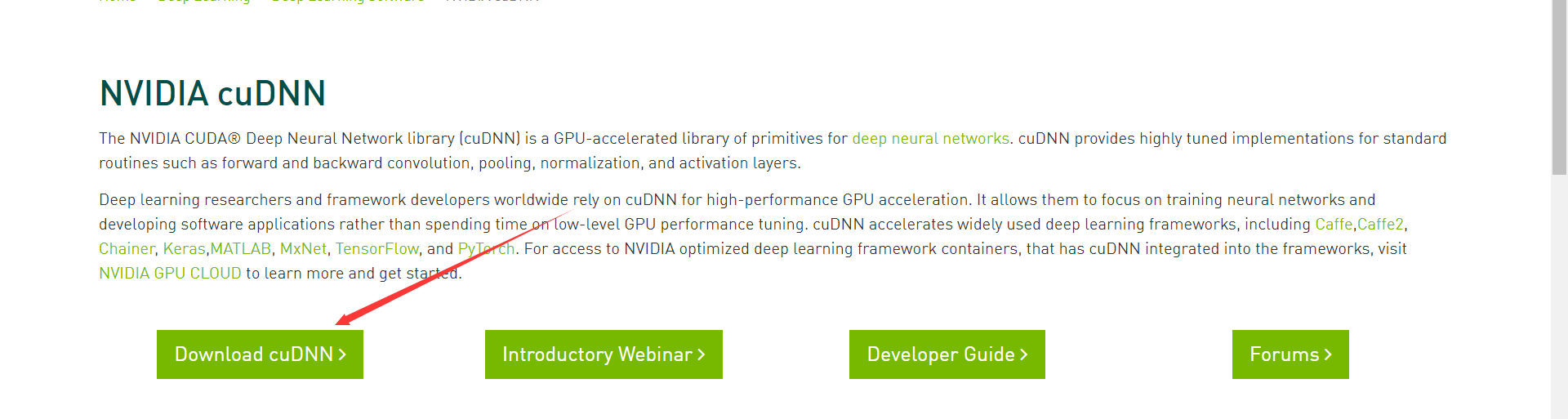
**4、CUDA Toolkit 和CUDNN**

CUDA Toolkit 下载官网：<https://developer.nvidia.com/cuda-downloads>



这是支持10.2版本的CUDA Toolkit，下载的时候一定要确认自己下的CUDA Toolkit是什么版本的。官网：<https://developer.nvidia.com/cudnn>





**5、安装环节。**

下载完成后，进行CUDA Toolkits的安装 双击安装包，然后进行安装，默认就行。当然也有网上说的是自定义安装，选择下面四个包就够了。我忘了我选的啥了。



安装完成后，他会自动配置一些环境变量，但是还有些需要自己配置。举个栗子：

“

CUDA\_SDK\_PATH = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0(这是默认安装位置的路径，经自定义路径后，他的路径为D:\NVIDIA\CUDA Samples)

CUDA\_LIB\_PATH = %CUDA\_PATH%\lib\x64

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

CUDA\_SDK\_BIN\_PATH = %CUDA\_SDK\_PATH%\bin\win64

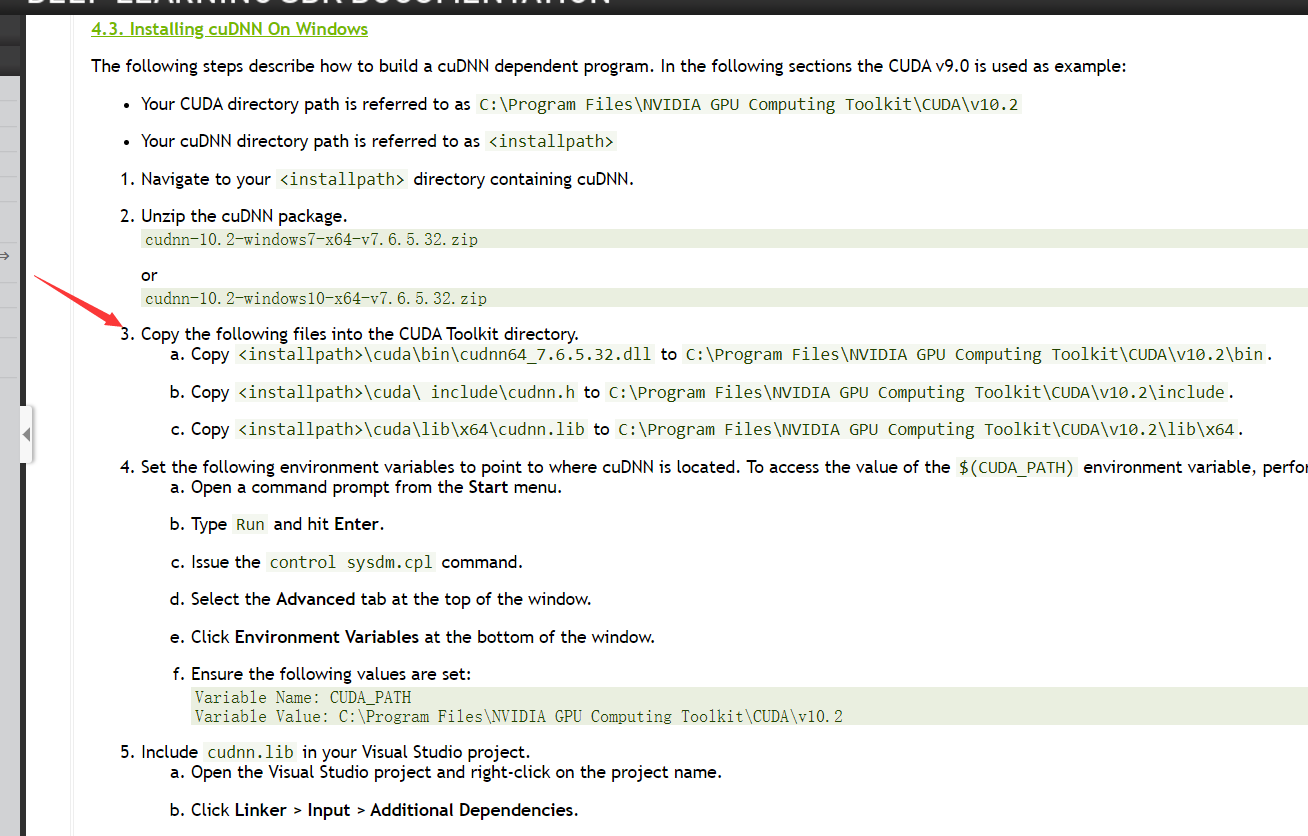
CUDA\_SDK\_LIB\_PATH = %CUDA\_SDK\_PATH%\common\lib\x64

“

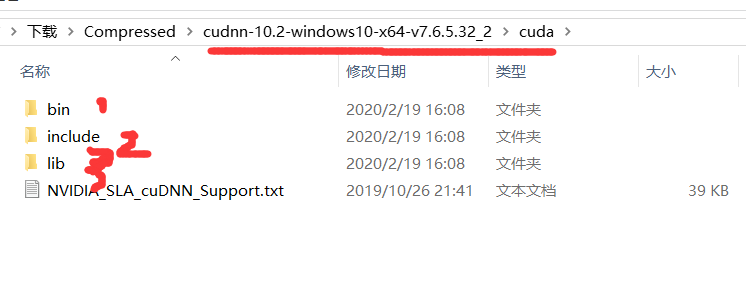
注意上述文件的版本不要直接复制哈。

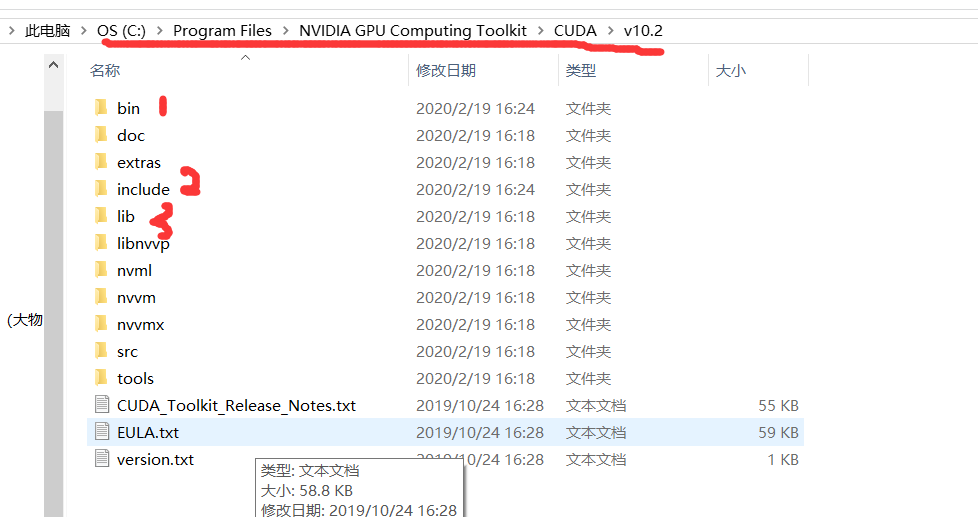
这一步完成后，再安装官网的教程。地址：<https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installwindows>

里面也不用全看，就看关键部分，我截图了。



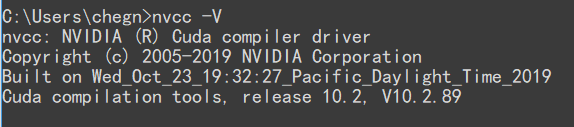
最重要的就是把CUDNN里面的三个文件分别copy到cuda的安装目录里面对应的文件夹以下。





不是把前者的文件夹去覆盖后者哈，而是把前者的对应文件夹的里面的东西copy到后者对应的文件夹下。Copy完了，前者就无用了。

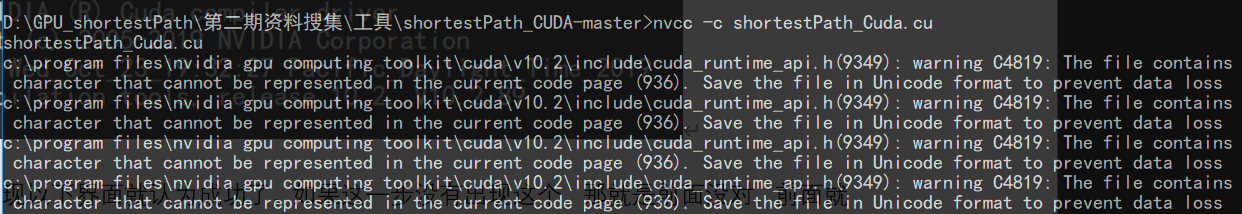
网上很多的验证阶段都是 这是后输入”nvcc -V”



然后出现以下界面就认为成功了。如果这一步没有出现这个，那就是前面没对，前面就错了。如果后面报错什么文件找不到的话，就去VS文件夹里面找，找到了就把他加入环境变量。比如，我之前编译就出现了一个”cl.exe” 找不到报错，然后我全盘搜索，在window这个文件夹下找到一个，加入环境变量，还是报错，经过敬老师确认只是同名文件而已。后来在VS的文件夹下又找到一个，才没报错了。

我的VS路径：C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\amd64

编译CU文件，效果是这样：



跑完了就好了。后面怎用，我还不知道。

**6、安装“pycuda”**

如果第一步你没有加入环境变量，请打开“anaconda prompt”，加入了，就随便找一个cmd但是需要管理员权限运行（这里，如果后面仅仅使用就不需要管理员，需要安装东西就需要管理员权限），然后“conda install pycuda” 如果找不到或者报错那就”pip install pycuda” 途中如果说有什么包确实 你就先”pip install XXX”,手动解决依赖。如果PIP速度比较慢可以更换一个国内的源。方法如下：

在你的/user目录下新建一个pip的文件夹，然后在里面新建一个文件“pip.ini”。copy下列内容，不要包含首尾两个引号。

“

[global]

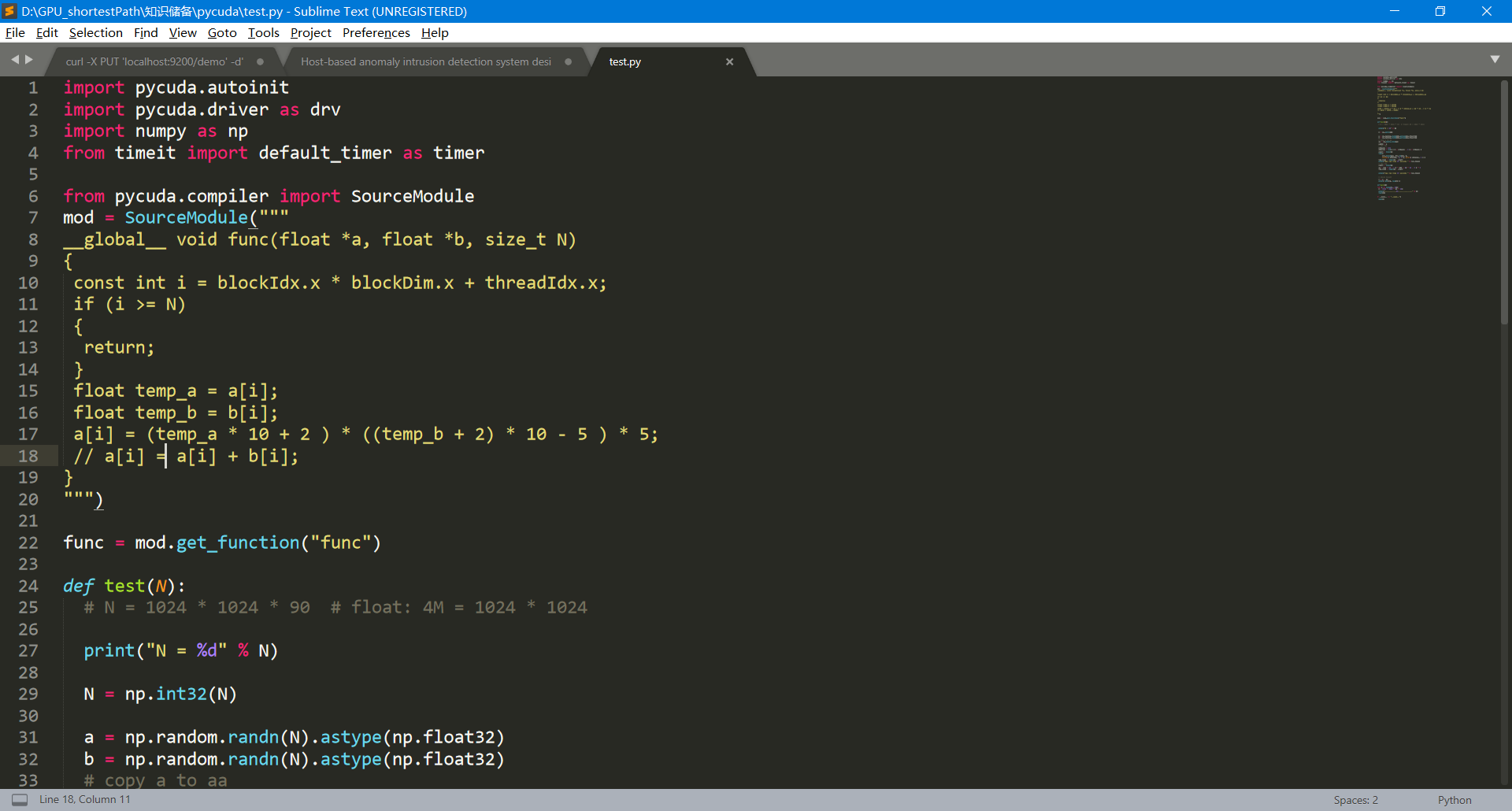
index-url = https://pypi.tuna.tsinghua.edu.cn/simple

”

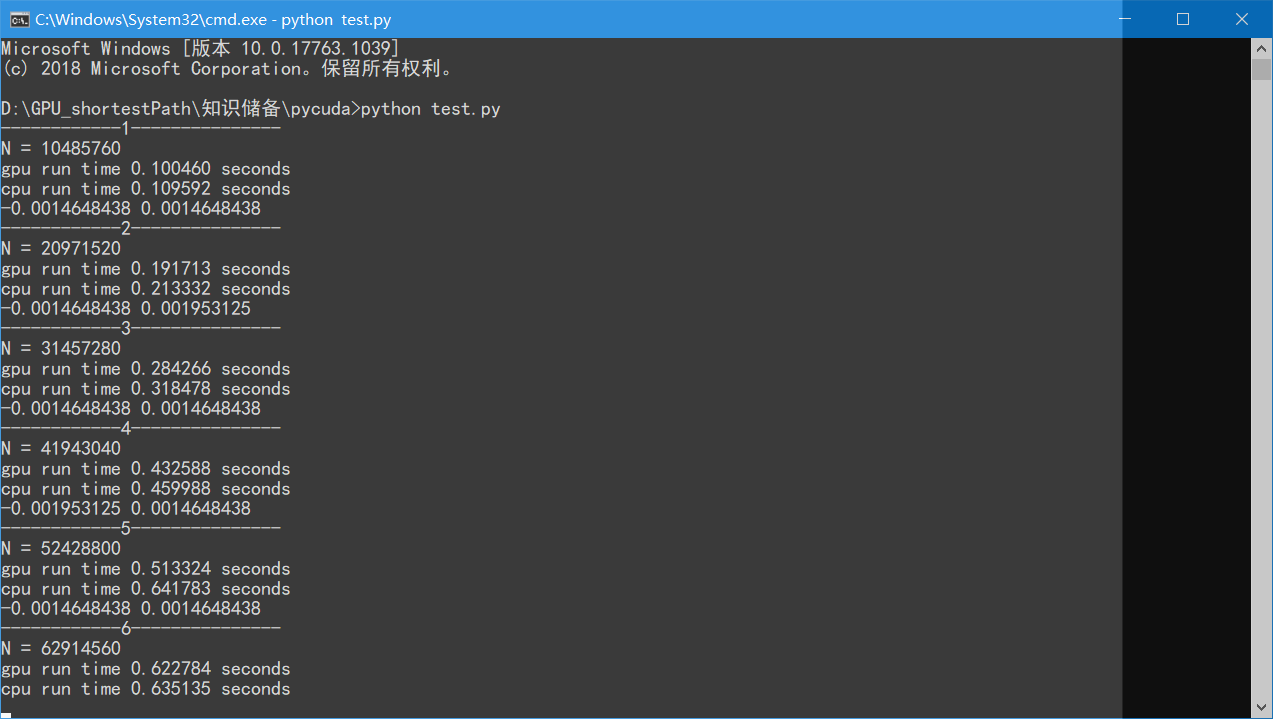
这里是换的清华的源，如果想换其他的，就把http哪个部分修改为对应的源就好了。

**7、安装完成以后，就可以开始使用了。**

下面是一个demo



然后运行它：



这里就可以体现一开始加入path的好处了。否则你想要运行这个demo，你要么需要使用jupyter 在网页中运行，当然习惯了或者开放项目这样是很方便的，但是一个demo就显得不方便了。 要么，你需要打开spider（一个anaconda的IDE），在spider里面运行。并且你还不能在命令行中愉快的ipython。这里你可能会说，你不卸载原来去python下载的python版本不就好了吗？我本来就可以在命令行中使用python。但是哪个python的包和conda install的包不在一起，会有很多麻烦，找不到包之类的。总之，anaconda是一个很不错的工具。还是卸载原来的吧。

好的，目前我就做到这种地方，剩下的就是看文献和网上找pycuda使用教程吧。

Demo源码：

import pycuda.autoinit

import pycuda.driver as drv

import numpy as np

from timeit import default\_timer as timer

from pycuda.compiler import SourceModule

mod = SourceModule("""

\_\_global\_\_ void func(float \*a, float \*b, size\_t N)

{

const int i = blockIdx.x \* blockDim.x + threadIdx.x;

if (i >= N)

{

return;

}

float temp\_a = a[i];

float temp\_b = b[i];

a[i] = (temp\_a \* 10 + 2 ) \* ((temp\_b + 2) \* 10 - 5 ) \* 5;

// a[i] = a[i] + b[i];

}

""")

func = mod.get\_function("func")

def test(N):

# N = 1024 \* 1024 \* 90 # float: 4M = 1024 \* 1024

print("N = %d" % N)

N = np.int32(N)

a = np.random.randn(N).astype(np.float32)

b = np.random.randn(N).astype(np.float32)

# copy a to aa

aa = np.empty\_like(a)

aa[:] = a

# GPU run

nTheads = 256

nBlocks = int( ( N + nTheads - 1 ) / nTheads )

start = timer()

func(

drv.InOut(a), drv.In(b), N,

block=( nTheads, 1, 1 ), grid=( nBlocks, 1 ) )

run\_time = timer() - start

print("gpu run time %f seconds " % run\_time)

# cpu run

start = timer()

aa = (aa \* 10 + 2 ) \* ((b + 2) \* 10 - 5 ) \* 5

run\_time = timer() - start

print("cpu run time %f seconds " % run\_time)

# check result

r = a - aa

print( min(r), max(r) )

def main():

for n in range(1, 10):

N = 1024 \* 1024 \* (n \* 10)

print("------------%d---------------" % n)

test(N)

if \_\_name\_\_ == '\_\_main\_\_':

main()