在自己的电脑上跑BinaryConnect

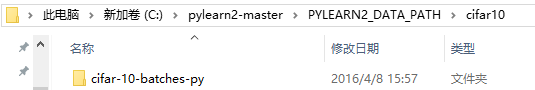
配置数据库

参考资料：<http://www.cnblogs.com/tornadomeet/p/3424520.html>

1. 数据库路径
   1. 系统变量PYLEARN2\_DATA\_PATH:

C:\pylearn2-master\PYLEARN2\_DATA\_PATH

* 1. 这样放置数据库

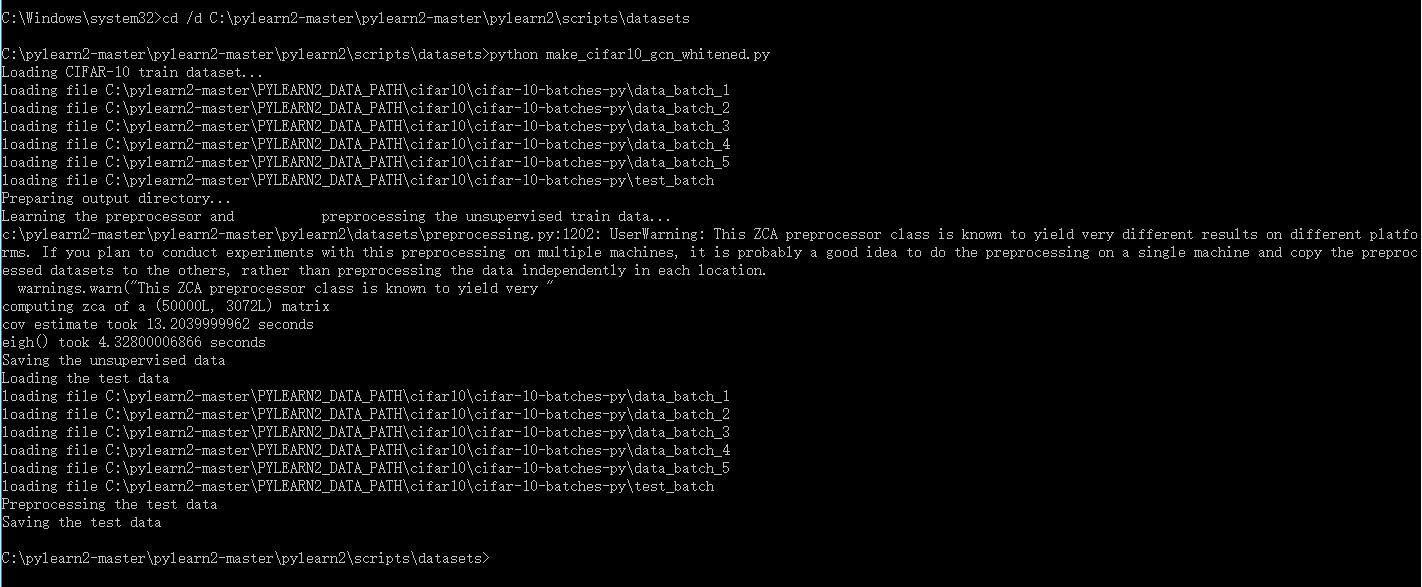


【2】数据库预处理是自己拿pylearn2自带的函数做的，不是BinaryConnect提供的。

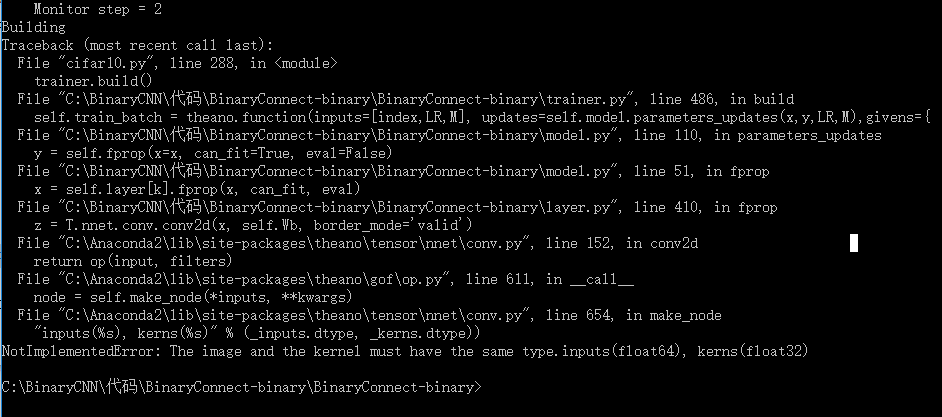
运行

cd /d C:\pylearn2-master\pylearn2-master\pylearn2\scripts\datasets

python make\_cifar10\_gcn\_whitened.py



1. 跑的程序又出错了



Building

Traceback (most recent call last):

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2.py", line 288, in <module>

trainer.build()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 486, in build

self.train\_batch = theano.function(inputs=[index,LR,M], updates=self.model.parameters\_updates(x,y,LR,M),givens={

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\model.py", line 110, in parameters\_updates

y = self.fprop(x=x, can\_fit=True, eval=False)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\model.py", line 51, in fprop

x = self.layer[k].fprop(x, can\_fit, eval)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\layer.py", line 410, in fprop

z = T.nnet.conv.conv2d(x, self.Wb, border\_mode='valid')

File "C:\Anaconda2\lib\site-packages\theano\tensor\nnet\conv.py", line 152, in conv2d

return op(input, filters)

File "C:\Anaconda2\lib\site-packages\theano\gof\op.py", line 611, in \_\_call\_\_

node = self.make\_node(\*inputs, \*\*kwargs)

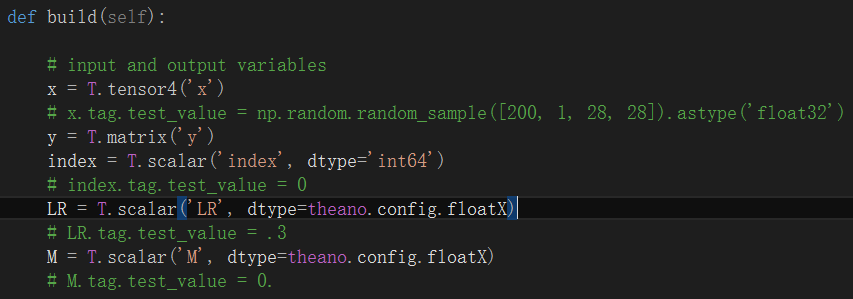
File "C:\Anaconda2\lib\site-packages\theano\tensor\nnet\conv.py", line 654, in make\_node

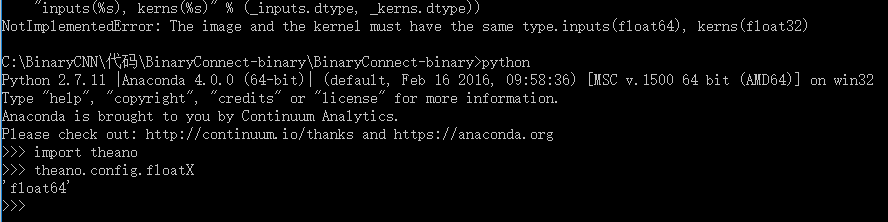
"inputs(%s), kerns(%s)" % (\_inputs.dtype, \_kerns.dtype))

**NotImplementedError: The image and the kernel must have the same type.inputs(float64), kerns(float32)**

好像是因为theano 目前只支持float32，至少是用到gpu的话。

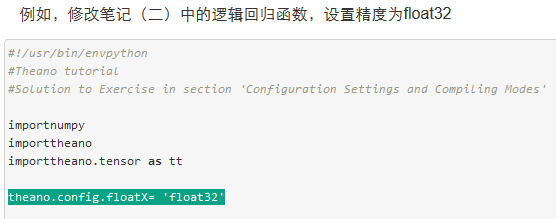
我的系统是win7 64的。参照theano 文档 config一节设置后这个问题就解决了。<http://deeplearning.net/software/theano/library/config.html>





如何修改theano.config?

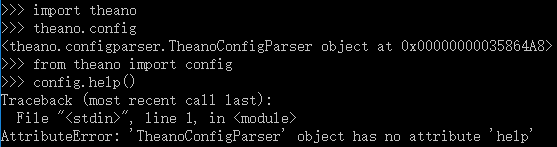
直接修改。。。



[**config**](http://deeplearning.net/software/theano/library/config.html#module-config) **– Theano Configuration**

**Guide**

The config module（config是一个module）



contains many attributes（vt.认为…是; 把…归于; 把…品质归于某人; 认为某事[物]属于某人[物] n.属性; （人或物的）特征; 价值; [语法学]定语） that modify Theano’s behavior. Many of these attributes are consulted during the import of the theano module and many are assumed to be read-only.

*As a rule, the attributes in this module should not be modified by user code.*

Theano’s code comes with default values for these attributes, but you can override（vt. 覆盖; 推翻，无视; 践踏; 优先于;n. 佣金; 超驰控制装置; 撤消，推翻;[网络] 重写; 重载; 方法重写;[例句]） them from your .theanorc file, and override those values in turn by the [THEANO\_FLAGS](http://deeplearning.net/software/theano/library/config.html#envvar-THEANO_FLAGS) environment variable.

The order of precedence is:

1. an assignment to theano.config.<property>
2. an assignment in [THEANO\_FLAGS](http://deeplearning.net/software/theano/library/config.html#envvar-THEANO_FLAGS)
3. an assignment in the .theanorc file (or the file indicated in [THEANORC](http://deeplearning.net/software/theano/library/config.html#envvar-THEANORC))

You can print out the current/effective configuration at any time by printing theano.config. For example, to see a list of all active configuration variables, type this from the command-line:

python -c 'import theano; print(theano.config)' | less

**Environment Variables**

THEANO\_FLAGS

This is a list of comma-delimited key=value pairs that control Theano’s behavior.

For example, in bash, you can override your [THEANORC](http://deeplearning.net/software/theano/library/config.html#envvar-THEANORC) defaults for <myscript>.py by typing this:

THEANO\_FLAGS='floatX=float32,device=gpu0,lib.cnmem=1' python <myscript>.py

If a value is defined several times in THEANO\_FLAGS, the right-most definition is used. So, for instance, if THEANO\_FLAGS='device=cpu,device=gpu0', then gpu0 will be used.

THEANORC

The location[s] of the .theanorc file[s] in ConfigParser format. It defaults to $HOME/.theanorc. On Windows, it defaults to $HOME/.theanorc:$HOME/.theanorc.txt to make Windows users’ life easier.



Here is the .theanorc equivalent to the THEANO\_FLAGS in the example above:

[global]

floatX = float32

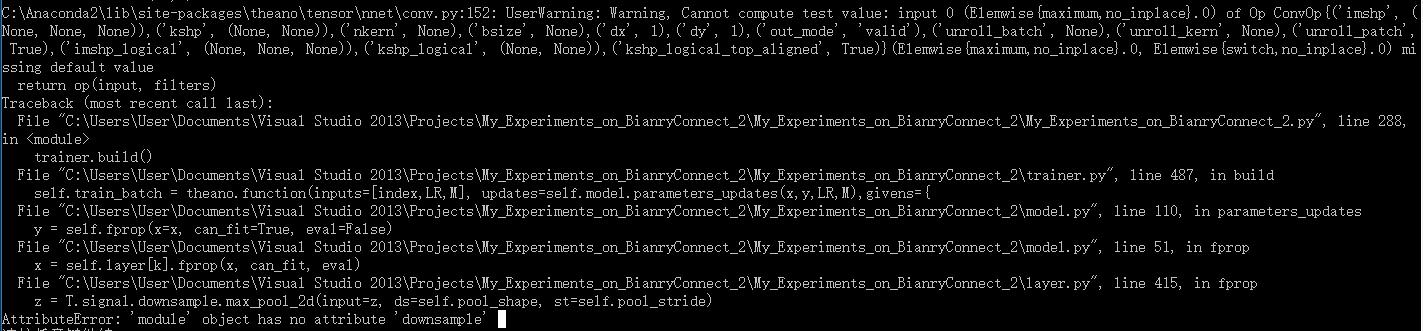
device = gpu0

[lib]

cnmem = 1

Configuration attributes that are available directly in config (e.g. config.device, config.mode) should be defined in the [global] section. Attributes from a subsection of config (e.g. config.lib.cnmem, config.dnn.conv.algo\_fwd) should be defined in their corresponding section (e.g. [nvcc], [dnn.conv]).

【之后的一个报错】



C:\Anaconda2\lib\site-packages\theano\tensor\nnet\conv.py:152: UserWarning: Warning, Cannot compute test value: input 0 (Elemwise{maximum,no\_inplace}.0) of Op ConvOp{('imshp', (None, None, None)),('kshp', (None, None)),('nkern', None),('bsize', None),('dx', 1),('dy', 1),('out\_mode', 'valid'),('unroll\_batch', None),('unroll\_kern', None),('unroll\_patch', True),('imshp\_logical', (None, None, None)),('kshp\_logical', (None, None)),('kshp\_logical\_top\_aligned', True)}(Elemwise{maximum,no\_inplace}.0, Elemwise{switch,no\_inplace}.0) missing default value

return op(input, filters)

Traceback (most recent call last):

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\**My\_Experiments\_on\_BianryConnect\_2.py**", line **288**, in <module>

trainer.build()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\**trainer.py**", line **487**, in build

self.train\_batch = theano.function(inputs=[index,LR,M], updates=self.model.parameters\_updates(x,y,LR,M),givens={

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\**model.py**", line **110**, in parameters\_updates

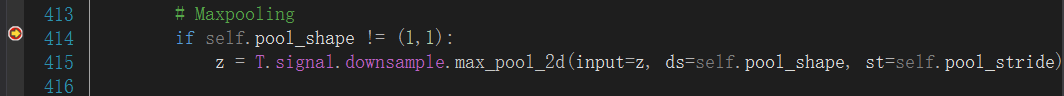
y = self.fprop(x=x, can\_fit=True, eval=False)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\**model.py**", line **51**, in fprop

x = self.layer[k].fprop(x, can\_fit, eval)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\**layer.py**", line **415**, in fprop

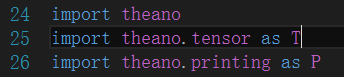
z = T.signal.downsample.max\_pool\_2d(input=z, ds=self.pool\_shape, st=self.pool\_stride)

AttributeError: 'module' object has no attribute 'downsample'  


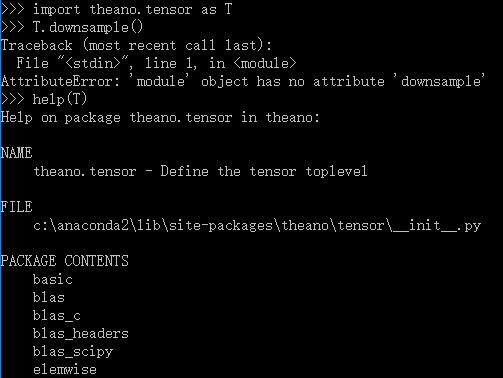
# Maxpooling

if self.pool\_shape != (1,1):

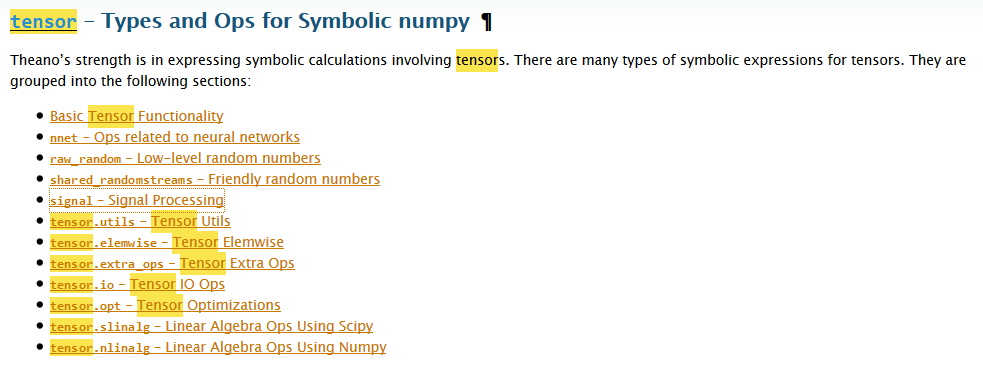
z = T.signal.downsample.max\_pool\_2d(input=z, ds=self.pool\_shape, st=self.pool\_stride)



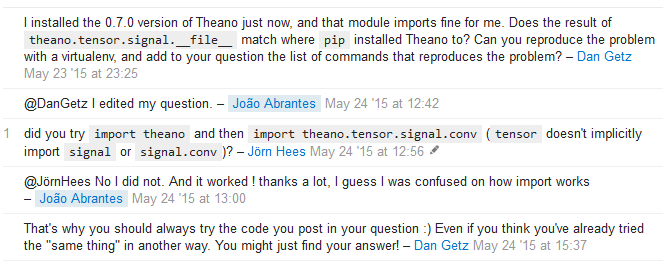
import theano.tensor as T



嗯嗯，很好，只是貌似tensor里面没有“signal”？



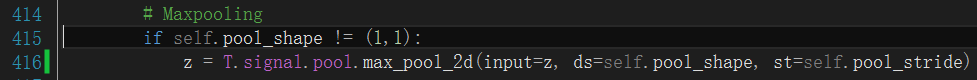
<http://stackoverflow.com/questions/30369287/theano-is-missing-signal-conv-module>



did you **try import theano and then import theano.tensor.signal.conv (tensor doesn't implicitly import signal or signal.conv)?** – Jörn Hees May 24 '15 at 12:56

@JörnHees No I did not. And it worked ! thanks a lot, I guess I was confused on how import works – João Abrantes May 24 '15 at 13:00

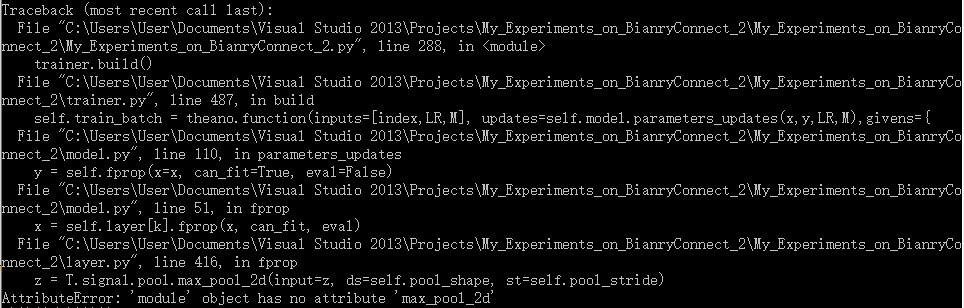
That's why you should always try the code you post in your question :) Even if you think you've already tried the "same thing" in another way. You might just find your answer! – Dan Getz May 24 '15 at 15:37



最后的结果就是：改成pool就好了233333

if self.pool\_shape != (1,1):

z = T.signal.pool.max\_pool\_2d(input=z, ds=self.pool\_shape, st=self.pool\_stride)

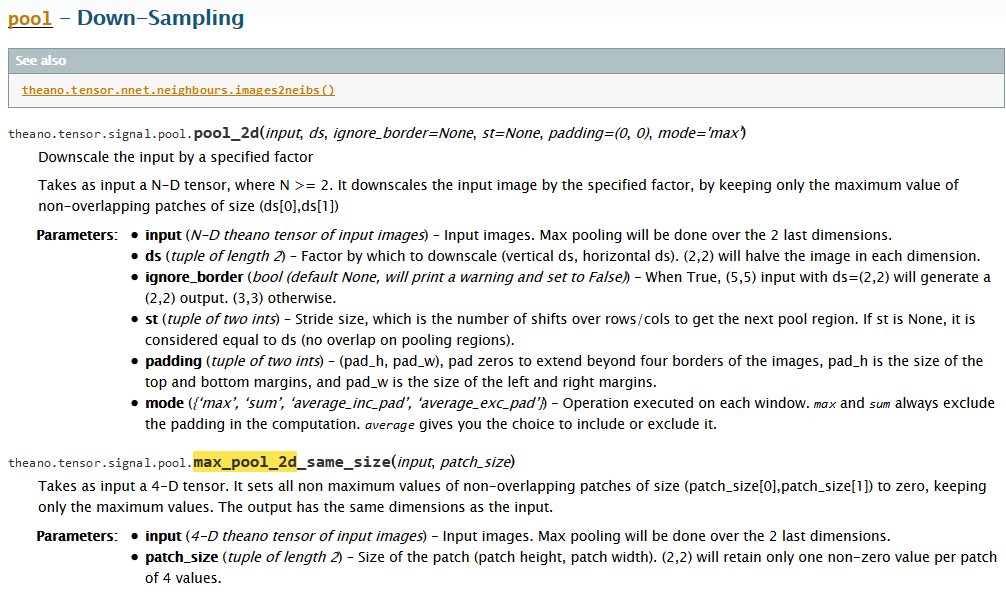


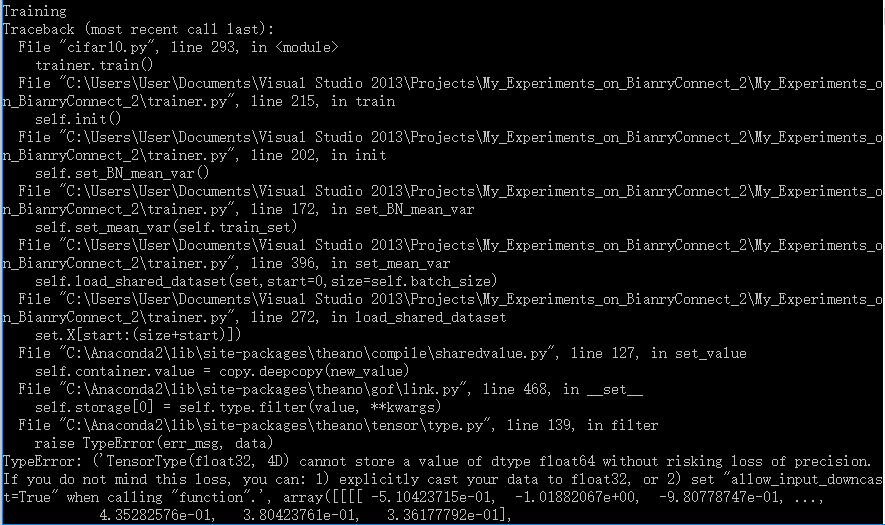
<http://deeplearning.net/software/theano/library/tensor/signal/pool.html?highlight=max_pool_2d#module-pool>

改成这样：

if self.pool\_shape != (1,1):

z = **T.signal.pool.pool\_2d**(input=z, ds=self.pool\_shape, st=self.pool\_stride, **mode='max'**)



【又报错了。。。】

Training

Traceback (most recent call last):

File "cifar10.py", line 293, in <module>

trainer.train()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 215, in train

self.init()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 202, in init

self.set\_BN\_mean\_var()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 172, in set\_BN\_mean\_var

self.set\_mean\_var(self.train\_set)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 396, in set\_mean\_var

self.load\_shared\_dataset(set,start=0,size=self.batch\_size)

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 272, in load\_shared\_dataset

set.X[start:(size+start)])

File "C:\Anaconda2\lib\site-packages\theano\compile\sharedvalue.py", line 127, in set\_value

self.container.value = copy.deepcopy(new\_value)

File "C:\Anaconda2\lib\site-packages\theano\gof\link.py", line 468, in \_\_set\_\_

self.storage[0] = **self.type.filter**(value, \*\*kwargs)

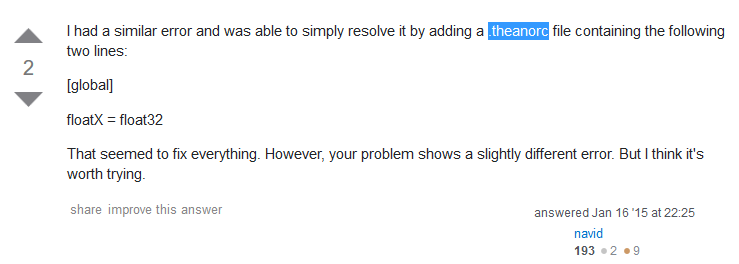
File "C:\Anaconda2\lib\site-packages\theano\tensor\type.py", line 139, in filter

raise TypeError(err\_msg, data)

TypeError: ('TensorType(float32, 4D) cannot store a value of dtype float64 without risking loss of precision. If you do not mind this loss, you can: 1) explicitly cast your data to float32, or 2) set "allow\_input\_downcast=True" when calling "function".',

TypeError: ('TensorType(float32, 4D) cannot store a value of dtype float64 without risking loss of precision. If you do not mind this loss, you can: 1) explicitly cast your data to float32, or 2) set "allow\_input\_downcast=True" when calling "function".', array([[[[ -5.10423715e-01, -1.01882067e+00, -9.80778747e-01, ...,

<http://stackoverflow.com/questions/22914786/error-because-of-theano-and-numpy-variable-types>



注意！！！这个东西必须只能是这个名字！！！注意后缀！！！其他的都不对！！！



我改成了这个：

[global]

openmp=False

floatX = float32

device = gpu0

[blas]

ldflags=

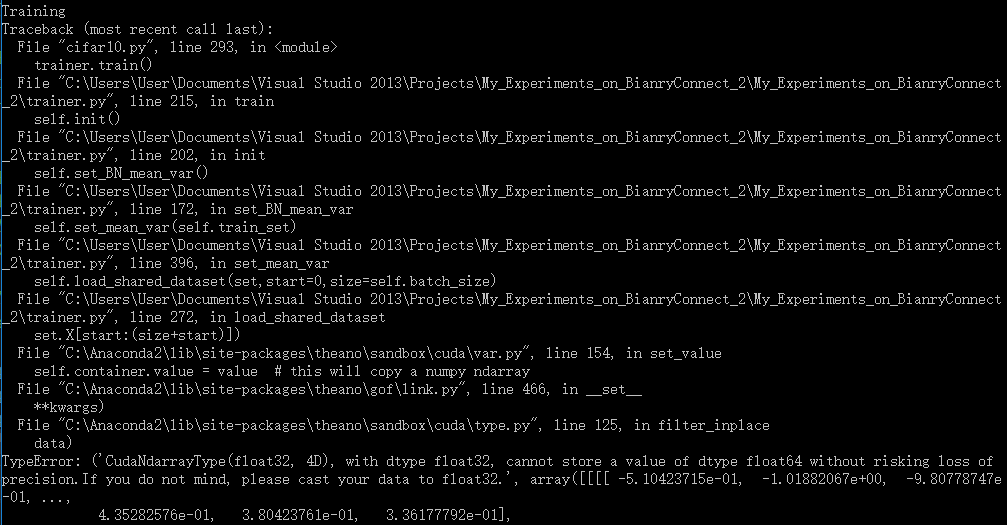
[gcc]

cxxflags=-I C:\Anaconda2\MinGW

import theano 之后是这个样子的：



【这个float32 还是 float 64的错误怎么还没完没了了？？？】



Training

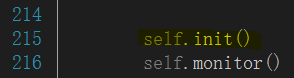
Traceback (most recent call last):

File "cifar10.py", line 293, in <module>

trainer.train()

File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 215, in train

self.init()



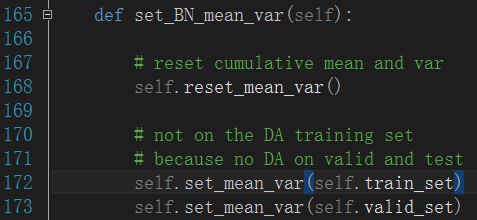
File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 202, in init

self.set\_BN\_mean\_var()



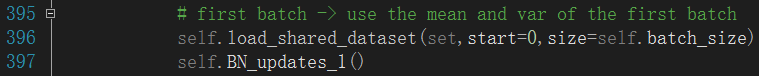
File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 172, in set\_BN\_mean\_var

self.set\_mean\_var(self.train\_set)



File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 396, in set\_mean\_var

self.load\_shared\_dataset(set,start=0,size=self.batch\_size)



File "C:\Users\User\Documents\Visual Studio 2013\Projects\My\_Experiments\_on\_BianryConnect\_2\My\_Experiments\_on\_BianryConnect\_2\trainer.py", line 272, in load\_shared\_dataset

set.X[start:(size+start)])



File "C:\Anaconda2\lib\site-packages\theano\sandbox\cuda\var.py", line 154, in set\_value

self.container.value = value # this will copy a numpy ndarray

File "C:\Anaconda2\lib\site-packages\theano\gof\link.py", line 466, in \_\_set\_\_

\*\*kwargs)

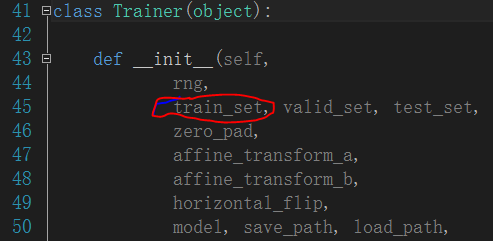
File "C:\Anaconda2\lib\site-packages\theano\sandbox\cuda\type.py", line 125, in filter\_inplace

data)

TypeError: ('CudaNdarrayType(float32, 4D), with dtype float32, cannot store a value of dtype float64 without risking loss of precision.If you do not mind, please cast your data to float32.', array([[[[ -5.10423715e-01, -1.01882067e+00, -9.80778747e-01, ...,

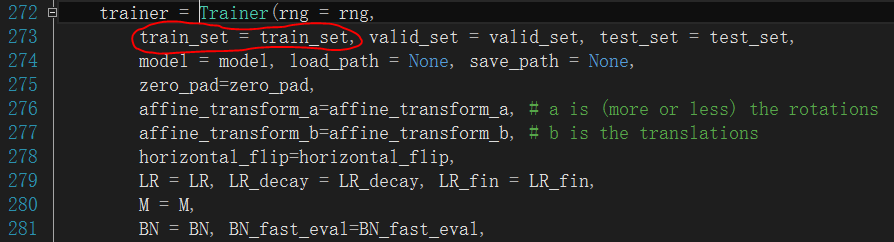
看一下数据的来龙去脉：

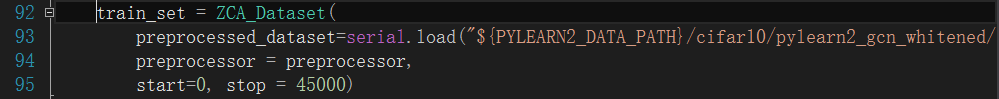
1. 类Trainer



看我们关心的类的引用：







ZCA\_Dataset:



我们来看一看ZCA\_Dataset之后究竟是个啥

import pylearn2

from pylearn2.datasets.zca\_dataset import ZCA\_Dataset

from pylearn2.utils import serial

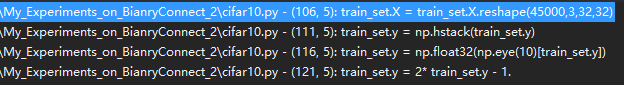
preprocessor = serial.load("${PYLEARN2\_DATA\_PATH}/cifar10/pylearn2\_gcn\_whitened/preprocessor.pkl")

train\_set = ZCA\_Dataset(

preprocessed\_dataset=serial.load("${PYLEARN2\_DATA\_PATH}/cifar10/pylearn2\_gcn\_whitened/train.pkl"),

preprocessor = preprocessor,

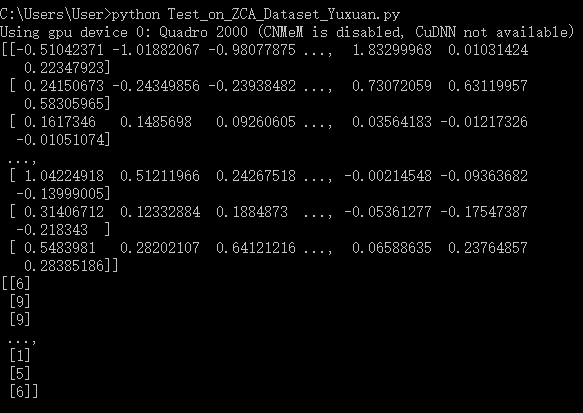
start=0, stop = 45000)

#我想看一看train\_set的各个属性都是什么

#后面只用到了train\_set.X 和 train\_set.y

print train\_set.X

print train\_set.y



看来X是图片矩阵，y是类别标签。

我要显示一下X的数据类型，float32还是float64?

查一下dtype

**import** pylearn2

**from** numpy **import** **\***

**from** pylearn2**.**datasets**.**zca\_dataset **import** ZCA\_Dataset

**from** pylearn2**.**utils **import** serial

preprocessor **=** serial**.**load**(**"${PYLEARN2\_DATA\_PATH}/cifar10/pylearn2\_gcn\_whitened/preprocessor.pkl"**)**

train\_set **=** ZCA\_Dataset**(**

preprocessed\_dataset**=**serial**.**load**(**"${PYLEARN2\_DATA\_PATH}/cifar10/pylearn2\_gcn\_whitened/train.pkl"**),**

preprocessor **=** preprocessor**,**

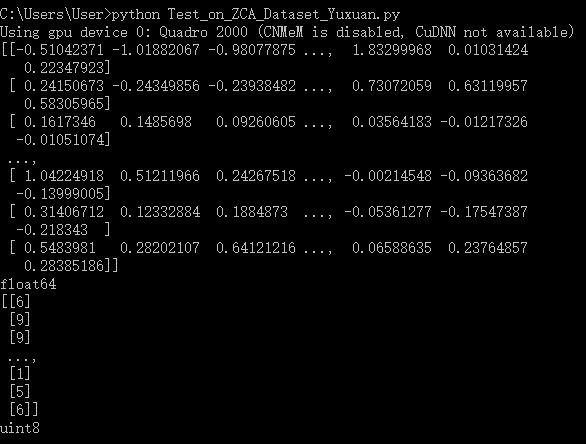
start**=**0**,** stop **=** 45000**)**

**print** train\_set**.**X

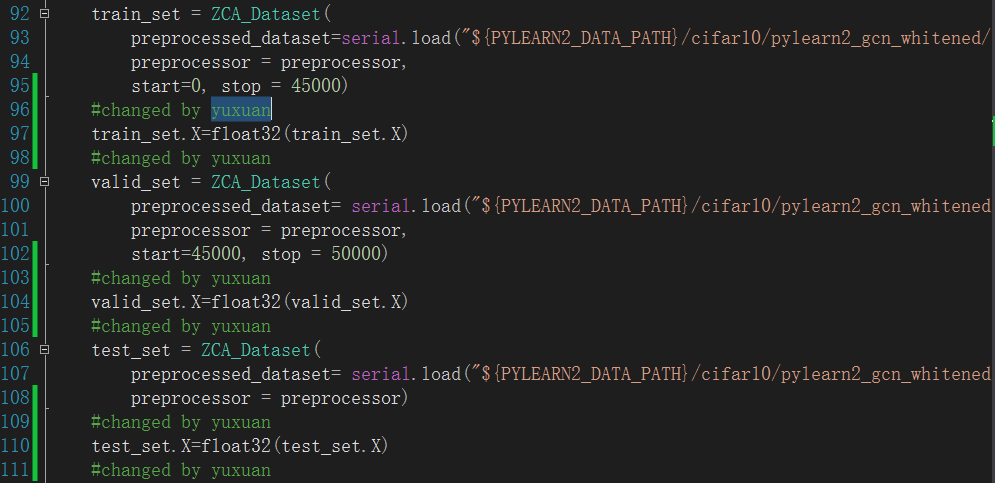
**print** train\_set**.**X**.**dtype

**print** train\_set**.**y

**print** train\_set**.**y**.**dtype

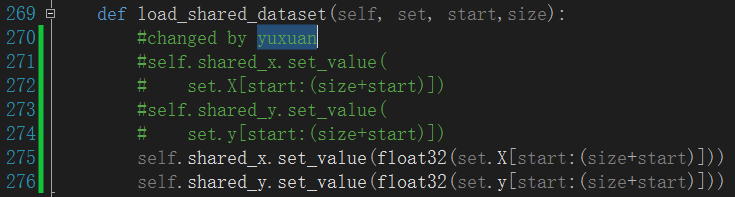


那么是不是直接在这里把数据类型换成float32就可以了？



这样改没成功

再在报错的地方改一下：



GPU跑不开，索性不用了，在config文件里直接删掉那一行。

感觉CPU跑的话慢得难以忍受，我把batch size 改成了50，然后用GPU跑。

添加了几句为了显示进度的指令，删掉之后可能更快。

那些指令可以通过搜索

#Yuxuan\_change\_for\_observision\_delete\_if\_necessary

来找到。

情况不太对：

还是out of memoey

epoch 0:

learning rate 0.300000

momentum 0.000000

validation error rate 89.800000%

test error rate 91.010000%

epoch associated to best validation error 0

best validation error rate 89.800000%

test error rate associated to best validation error 91.010000%

layer 0 weights mean abs = 0.0360811

layer 1 weights mean abs = 0.0322615

layer 2 weights mean abs = 0.0311418

layer 3 weights mean abs = 0.034235

layer 4 weights mean abs = 0.0221068

layer 5 weights mean abs = 0.0242005

layer 6 weights mean abs = 0.0156243

layer 7 weights mean abs = 0.0270642

layer 8 weights mean abs = 0.0382559

Error when trying to find the memory information on the GPU: out of memory

Error allocating 46080000 bytes of device memory (out of memory). Driver report 0 bytes free and 0 bytes total

