# Win7配置深度学习环境

目前，深度学习在解决图像分类，语音识别等问题上获得了已知的最优结果，该系列算法越来越受到学术界和工业界的重视。何为深度学习？一个直观的解释是如果一个机器学习算法在建模的过程中使用了多层的自动特征表示，则该机器学习算法可以称之为深度学习算法，也就是该机器学习算法可以自动地计算特征的特征表示。而卷积神经网络（以下简称为CNN）则是深度学习中最基础且有效的算法，CNN虽然最早由KunihikoFukushima提出，但是却因YannLeCun和Yoshua Bengio等人的改进实现而为人熟知，据说其实现的LeNet-5至今仍然被用于手写数字的识别（LeNet-5的简单介绍请参考博文blog.csdn.net/xuanyuansen/article/details/41800721）。为什么要提到这两位大牛呢？因为二人目前都是深度学习领域的执牛耳者，YannLeCun目前供职于FaceBook人工智能实验室，而YoshuaBengio仍然在学术界，既然提到了这二位，就不能不提到Google的GeoffreyE. Hinton和百度的AndrewNg，而Andrew Ng和YoshuaBengio都是Michael I. Jordan的学生，Jordan大神的实验室真可谓是机器学习领域的豪门（豪门盛宴请见[www.cs.berkeley.edu/~jordan/sail2010.jpg](http://www.cs.berkeley.edu/~jordan/sail2010.jpg" \t "_blank)）。

下面进入正题，目前CNN的开源实现很多，而且随着硬件技术的进步（此处应向Nvidia公司致敬，总裁是华人哦），使用GPU来进行神经网络算法的训练及实现已经成为主流。本文将简单介绍三种基于GPU的开源实现（均在Ubuntu14.04X64环境中成功运行）以及LeNet-5的个人简单实现。三种基于GPU的实现分别是DeepNet，Caffe和Theano，注意DeepNet和Theano均为Python实现，Caffe也大量利用了Python的各种库（其余用C/C++实现）。这三种开源环境均包含了很多机器学习算法（包含深度学习算法），CNN只是其一。

CUDA安装。

CUDA的下载地址在[https://developer.nvidia.com/cuda-downloads](https://developer.nvidia.com/cuda-downloads" \t "_blank)，安装的步骤请见[http://blog.csdn.net/xuanyuansen/article/details/43152311](http://blog.csdn.net/xuanyuansen/article/details/43152311" \t "_blank)，此处需要特别注意的是要删除Ubuntu系统自带的开源显卡驱动，以及要彻底删除原系统自带的Nvidia相关的包。

一、DeepNet安装

DeepNet的下载地址是[https://github.com/nitishsrivastava/deepnet](https://github.com/nitishsrivastava/deepnet" \t "_blank)，利用git命令可以直接checkout下来，该源码包的作者是nitishsrivastava（GeoffreyE. Hinton的印度学生），其安装过程按照源码包中的INSTALL.txt说明即可，需要注意的该源码包的演示所用的数据不能用原始的MNIST数据，而要使用经过处理的MNIST数据，下载地址为[http://www.cs.toronto.edu/~nitish/deepnet/mnist.tar.gz](http://www.cs.toronto.edu/~nitish/deepnet/mnist.tar.gz" \t "_blank)。其它需要注意的是需确认Ubuntu系统的python环境是否包含了Numpy和Scipy这两个包，如果没有的话需要提前安装。DeepNet实现了六种模型，分别是：Feed-forwardNeural Nets；RestrictedBoltzmann Machines；DeepBelief Nets；Autoencoders；DeepBoltzmann Machines；ConvolutionalNeural Nets。

二、Caffe安装

Caffe的安装较为复杂，而且坑也比较多（OMG）。Caffe的下载地址为[https://github.com/BVLC/caffe](https://github.com/BVLC/caffe" \t "_blank)，安装的详细过程参见Caffe的官方安装指南[http://caffe.berkeleyvision.org/installation.html](http://caffe.berkeleyvision.org/installation.html" \t "_blank)，以及某位牛人的博客文章（写的非常好）[http://www.cnblogs.com/platero/p/3993877.html](http://www.cnblogs.com/platero/p/3993877.html" \t "_blank)，关于CUDA安装的部分可以直接略过，这里需要注意的有三个地方，一是IntelMKL的安装，该软件包获得很麻烦，可以直接使用OpenBLAS或者atlas代替（个人使用了OpenBLAS），一定记得在Caffe的配置文件中正确配置该项。二是OpenCV的安装，用原始的源码包会出错，解决方法在：[http://code.opencv.org/issues/3814](http://code.opencv.org/issues/3814" \t "_blank)，下载NCVPixelOperations.hpp 替换掉opencv2.4.9内的文件，重新build。三是安装Python的Anaconda包后Caffe编译会出错，解决方法是：删除anaconda/lib/下面的libm，命令为sudo rm -rf libm\*（参见博客链接[http://blog.csdn.net/ccemmawatson/article/details/42004105](http://blog.csdn.net/ccemmawatson/article/details/42004105" \t "_blank)）。

Caffe安装好后就可以按照开发文档搭建CNN网络了，参见官方的Tutorial，[http://caffe.berkeleyvision.org/tutorial/](http://caffe.berkeleyvision.org/tutorial/" \t "_blank)。

三、Theano安装

Theano的安装非常简单，使用也非常方便，分分钟就可以搞定，文档及下载地址 <http://deeplearning.net/software/theano/>。基于Theano的CNN模型参见官方的教材，其实现非常方便，地址为<http://www.deeplearning.net/tutorial/lenet.html#lenet>。

四、LeNet-5的个人简单Python实现（非GPU模式）

利用Python实现了LeNet-5的简单版本，首先是实现了多层神经网络，然后在此基础上实现了CNN，代码参见以下链接（代码中的BMNN2即多层神经网络的实现）：

1、http://blog.csdn.net/xuanyuansen/article/details/41544153

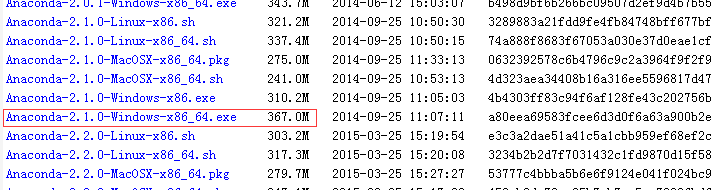
2、<http://blog.csdn.net/xuanyuansen/article/details/41924377>

# 我的实践

对于64位系统，所用软件统一为64位，强烈32位系统建议升级至64位。此外，由于caffe配置太过繁琐，本文使用基于python的theano作为深度学习的工具。

1. 安装python环境

建议使用Anaconda，下载地址，<https://repo.continuum.io/archive/index.htm>，里面集成了theano所需的各种包，由于2.1之后的版本取消了对MinGW的支持，因此这里一定要注意选2.1版本，否则编译不过theano。选择下图所示的版本（<https://repo.continuum.io/archive/Anaconda-2.1.0-Windows-x86_64.exe>）下载并安装，安装过程保持默认设置即可。



1. 安装theano

运行cmd，在命令行中输入pip install theano，下载安装theano。

如果想要卸载的话，在命令行中输入pip uninstall theano。

1. 安装cuda。

这个有我之前的教程，

1. 添加对GPU的支持

在运行cmd时出现的路径中（如C:\Users\lenovo\），新建文件

.theanorc.txt，并写入以下内容：

[global]

openmp=False

[blas]

ldflags=

[global]

device = gpu

floatX = float32

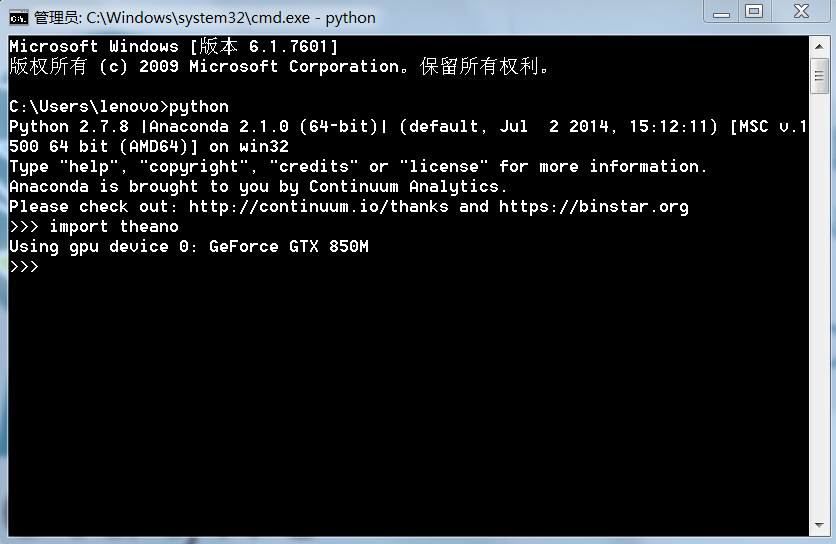
[nvcc]

flags =-LD:\Anaconda\libs

compiler\_bindir=D:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\bin

1. 验证安装成功。

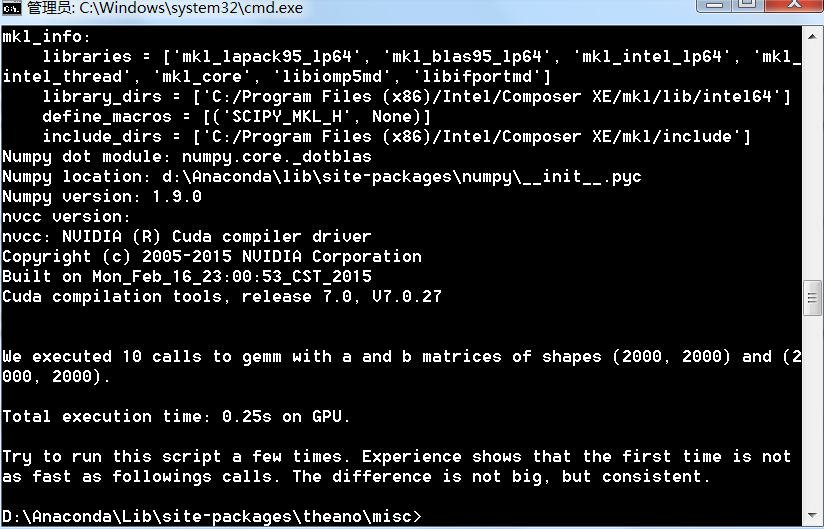
在cmd中输入python进入环境，import theano，出现下列画面表示成功安装theano并且配置了GPU（GPU型号可能不同，我的是GTX850M，搞深度学习的入门级别）：



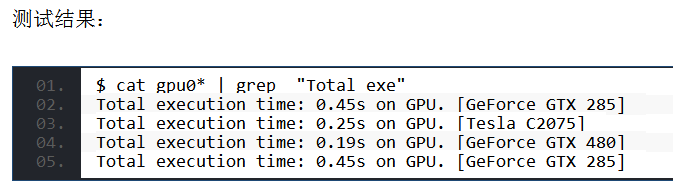
1. 测试性能

安装完成后，测试了下4个不同机器上GPU的性能.

python theano/misc/check\_blas.py



对比下其他显卡的结果，还不算太差。



1. 安装Keras。
2. 安装

在安装好theano和cuda之后，可能cnn，rbm等测试例子还不能跑。这时候要在vs目录d:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\include添加inttypes.h,stdint.h这两个文件，它们都可以在我的百度云里面下载。

到此，我们终于完成了所有的配置，enjoy it。http://images.cnitblog.com/blog/679105/201411/271940579023250.png

**引用：**

**官网安装教程**

<http://www.cnblogs.com/hanahimi/p/4127026.html>

[**http://deeplearning.net/software/theano/install.html#install**](http://deeplearning.net/software/theano/install.html#install)

[**Deep Learning: win7安装Theano,配置GPU**](http://blog.csdn.net/chlele0105/article/details/40557259)

[**http://blog.csdn.net/chlele0105/article/details/40557259**](http://blog.csdn.net/chlele0105/article/details/40557259)

[**手把手教你cuda5.5与VS2010的编译环境搭建**](http://blog.csdn.net/yeyang911/article/details/17450963)

[**http://blog.csdn.net/yeyang911/article/details/17450963**](http://blog.csdn.net/yeyang911/article/details/17450963)

[**64位WIN7上成功安装theano,并运行了g++, GPU**](http://blog.csdn.net/yeyang911/article/details/16357133)

[**http://blog.csdn.net/yeyang911/article/details/16357133**](http://blog.csdn.net/yeyang911/article/details/16357133)

[**Deep Learning 开发工具Theano安装配置**](http://blog.csdn.net/richard2357/article/details/16963187)

[**http://blog.csdn.net/richard2357/article/details/16963187**](http://blog.csdn.net/richard2357/article/details/16963187)

**相关链接 懒得收藏了**

**https://developer.nvidia.com/cuda-toolkit-archive**

**https://developer.nvidia.com/cuda-toolkit-50-archive**

**http://docs.nvidia.com/cuda/cuda-getting-started-guide-for-microsoft-windows/index.html#verify-you-have-cuda-enabled-system**

--------------------------------------------如果按照上面的过程你就可以使用了-------------------------------------------

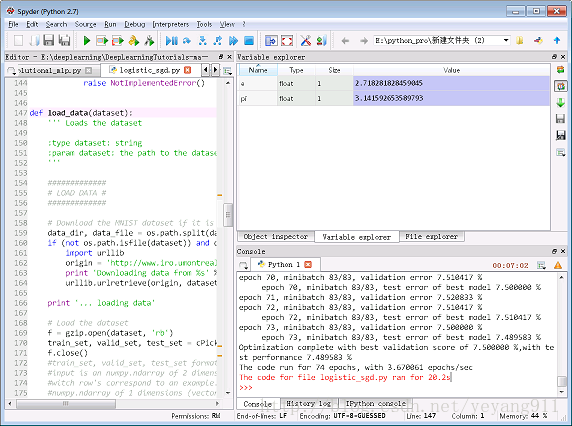
[制作theano可用的训练样本](http://blog.csdn.net/yeyang911/article/details/17523343) <http://blog.csdn.net/yeyang911/article/details/17523343>

--------------------------------------------下面是程序的测试--------------------------------------------------------------------

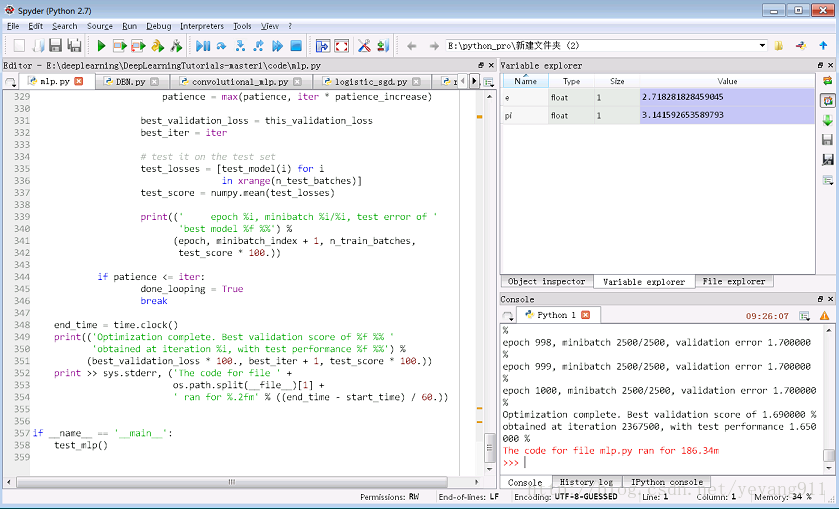
程序下载：

<http://download.csdn.net/detail/yeyang911/6937611>

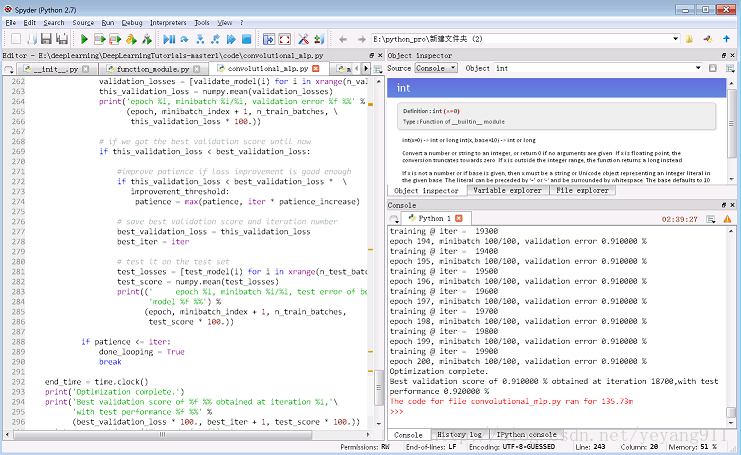
GPU下：logistic\_sgd.py ran for 20.2s



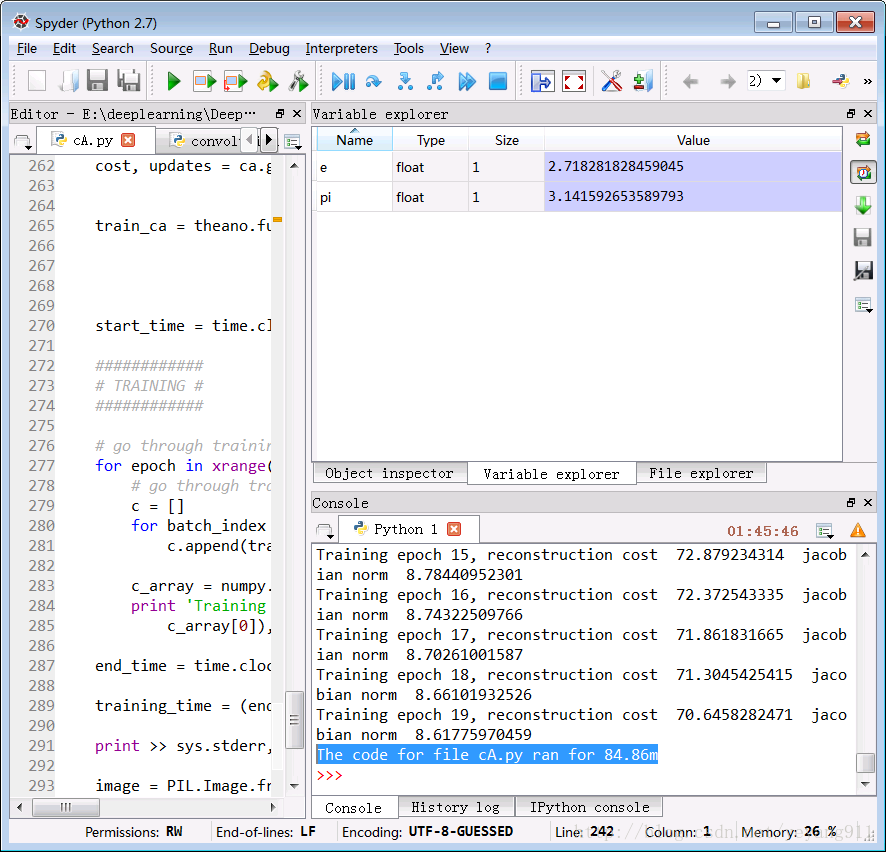
GPU下：mlp



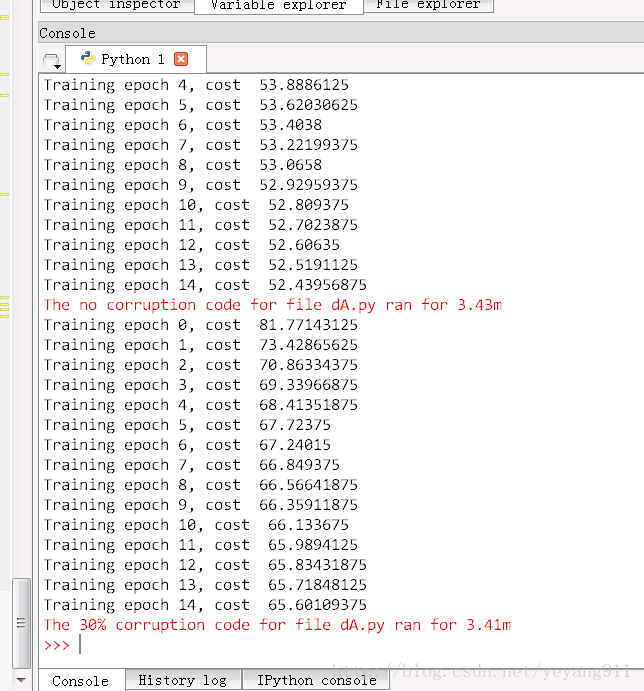
GPU:  CNN(**注：运行这个GPU程序需要下载“inttypes.h","stdint.h" 这两个头文件，然后放到VS2010/VC/include  
的文件下面。)**[**http://download.csdn.net/detail/yeyang911/6750713**](http://download.csdn.net/detail/yeyang911/6750713)**这个可以下载**

****

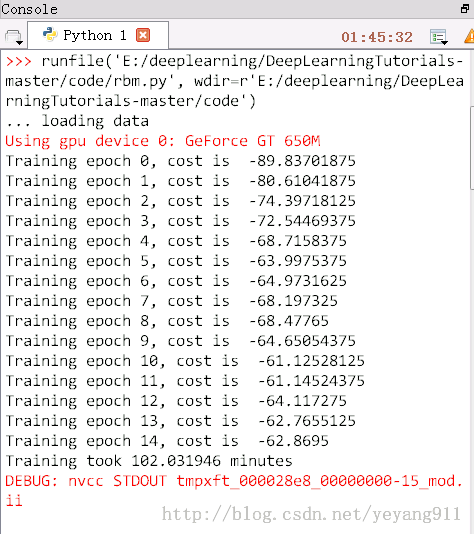
GPU:The code for file cA.py ran for 84.86m



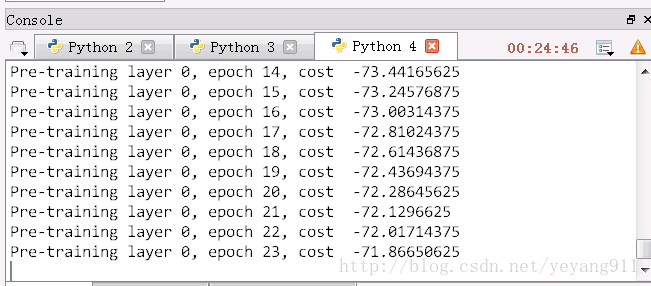
GPU:The no corruption code for file dA.py ran for 3.43m \ The 30% corruption code for file dA.py ran for 3.41m



GPU:rbm



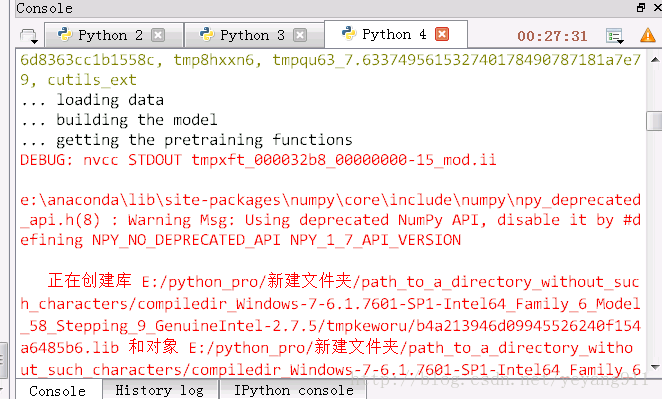




GPU下运行的  DBN  程序   可以发现 不到一分钟可以进行一次pre-train 比在CPU 下快了很多

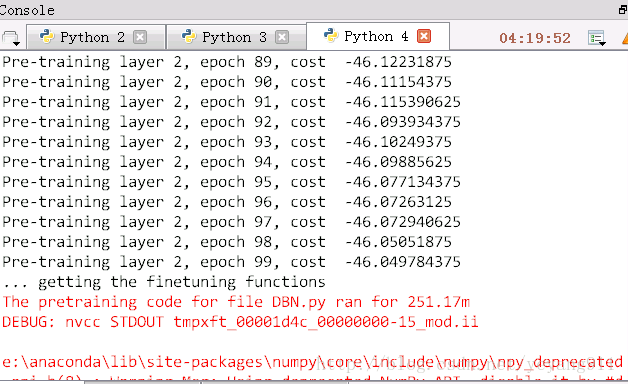
我的GPU是GT650M   因为是笔记本。

当然 ，在进行GPU处理前 好像还做了一些程序翻译：

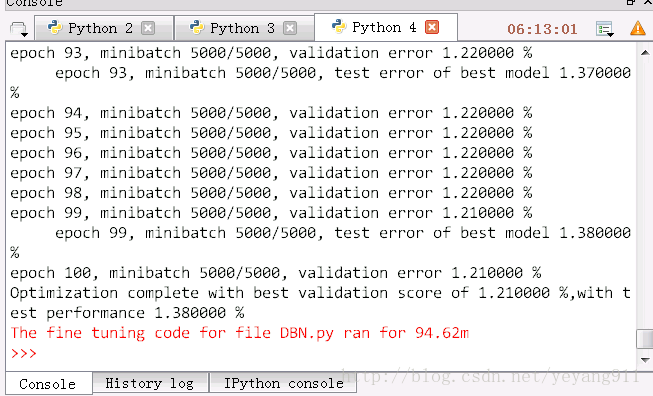


还有很长我就不贴出来了。祝大家成功配置成功。。。。

下面演示的是DBN 代码 运行的过程。。。



pre-train  耗时 251.17 分钟



fine tuning   94.62 分钟

如果在CPU上要跑2,3天吧。。