# python使用训练好的模型进行预测

python使用训练好的模型进行预测流程比较简单，说明如下：

1. 引入必要的包并添加环境变量：

import os

import sys

import numpy as np

import matplotlib.pyplot as plt

caffe\_root = '/home/hy17003/caffe/'

sys.path.insert(0,caffe\_root + 'python')

import caffe

1. 创建Net对象，载入模型

使用caffe.Classifier函数创建分类器，并初始化，分别传入网络结构文件prototxt和训练结果文件caffemodel

1. 设置CPU模式

使用caffe.set\_mode\_cpu()函数

1. 载入测试图像

使用caffe.io.load\_image函数载入图像

1. 预测并输出结果

prediction=net.predict([input\_image],oversample=False)

print 'predicted class: ',prediction[0].argmax()

1. 源码：

import os

import sys

import numpy as np

import matplotlib.pyplot as plt

caffe\_root = '/home/hy17003/caffe/'

sys.path.insert(0,caffe\_root + 'python')

import caffe

MODEL\_FILE = '/home/hy17003/caffe/examples/mnist/lenet.prototxt'

PRETRAINED = '/home/hy17003/caffe/examples/mnist/lenet\_iter\_10000.caffemodel'

IMAGE\_LIST = ['001.bmp','002.bmp','003.bmp','004.bmp','005.bmp','006.bmp',

'007.bmp','007.bmp','008.bmp','009.bmp','000.bmp']

IMAGE\_PATH = '/home/hy17003/caffe/examples/mnist/testImg/'

net=caffe.Classifier(MODEL\_FILE,PRETRAINED)

caffe.set\_mode\_cpu()

for image in IMAGE\_LIST:

input\_image = caffe.io.load\_image(IMAGE\_PATH + image,color=False)

prediction=net.predict([input\_image],oversample=False)

print 'predicted class: ',prediction[0].argmax()

im=input\_image[:,:,0]

plt.gray()

plt.imshow(im)

plt.show()

