全连接网络基础

MNIST数据集

我们输入的特征是每个像素点



图片出现的标签以一维数组形式给出，每个元素表示对应分类出现的数值的概率



返回各子集样本数的函数

Mnist.train.num\_examples但会训练的样本数量

Mnnist.validation.num\_example

返回标签和数据

Mnist.train.labels[0]

>>>array([0,0,0,0,0,0,0,1,0,0]) 表示这个数字是多少——（即它的标签）

Mnist.train.images[0]

Array([0. 0.])~blah 返回784个像素点

取一小撮数据，喂入神经网络

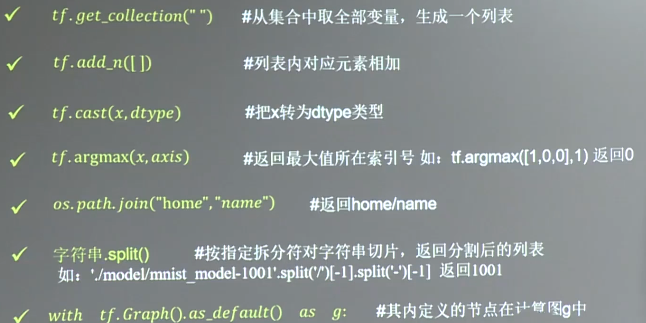
BATCH\_SIZE=200

Xs,ys=minst.train.next\_batch(200)随机读取200组数据进入神经网络

我们来看看 xs的样子，就用xs.shape

，xs.shape:(200,784)

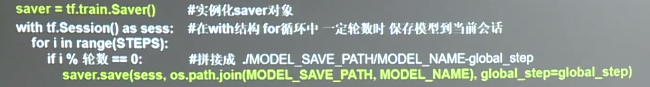
常用函数



这个os的意思为返回路径

保存模型

Saver=tf.train.saver() 实例化一个对象



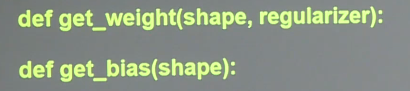
每每循环一定轮数后，开始保存



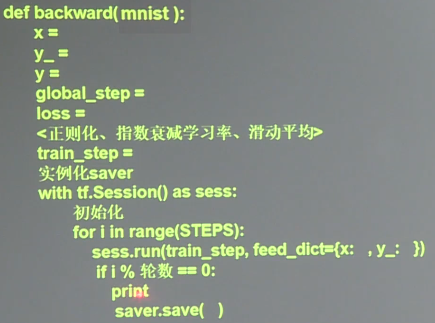
Equal表示相等，就是判断准确率的，我猜的哈哈哈

对于前向传播要搭建出网络

要定义出参数w，b 以及y的推导过程



对于反向传播，要训练网络参数



Ps：正则化,指数衰减学习率,滑动平均

指数衰减学习率是tensorflow提供的一种方法

这个算法实现达到的效果是:随着迭代次数的增加逐步减小学习率，是模型前期训练速度加快，后期训练模型更加稳定而不会出现极优值两边跳动的情况

learning\_rate = tf.train.exponential\_dacay(0.1 , global\_step , 100 , 0.96 , staircase = True)