2016.10.29练习：

1. 安装tensorflow和numpy库，运行mnist手写数字识别代码，并且根据代码的注释和ppt尽量理解代码每一步所完成的内容
2. 用tensorflow实现求二次函数最小值对应的解（参考梯度下降法在tensorflow的实现方法），二次函数的a和b通过用户输入

可能用到的函数：

1. tf.Variable()设置tensorflow变量，第一个参数是必要的，可以是一个初始值，也可以是一个数值或者矩阵的tensor。还可以设置：变量类型dtype，形状（生成矩阵时用到）shape，名字（用于可视化）name，是否可训练trainable。

2、tf.constant()设置一个包含常数或者常数矩阵内容的tensor，第一个参数是必要的，可以是数值，矩阵等。还可以填入：变量类型dtype。

3、tf.random\_normal()生成正态分布随机矩阵，第一个参数是必要的，填入要生成的矩阵的形状(形如”[a,b]”，生产一个a行b列的矩阵)。还可以填入：mean是均值，stddev标准差。

4、tf.placeholder()用于设置一个占位符，这个占位符的内容可以之后再填入。可以达到先用这个变量进行运算，后期再根据具体情况填入相应的值。第一个变量是必要的，填入参数类型，这里填入的参数类型一定要是tensorflow中的参数类型，在tensorflow中，,32位浮点数为tf.float32,16位整数为tf.int16，以此类推。第二个参数是矩阵的形状，如果不是矩阵就不填。使用方式是先设置一个placeholder变量，如果需要使用这个变量就直接拿来用（类似函数的形参）。真正运行的时候在run里面设置feed\_dict={}参数来向用placeholder设置的变量填入具体的值

5、tf.train.GradientDescentOptimizer().minimize()这个函数相对比较复杂，实现的是梯度下降的功能，第一个括号内的参数是学习率，用于控制学习的速度，第二个括号内的参数是要最小化的变量。

6、tf.global\_variables\_initializer()或者rf.initialize\_all\_variables()可以用来初始化变量

7、tf.Session()中的run函数用于执行某个步骤，具体的话就是自己设置的操作，比如说train，add等，要获得tensor的值也要通过run来完成，run的第一个参数是要执行的指令或者步骤（之前设置好的），还有一个参数是feed\_dict={a:value\_for\_a,b:value\_for\_b}这样的形式

然后有两个地方容易犯错

1. 所有操作都要在一个tf.Session()中完成
2. 使用变量前要初始化，建议在进入Session后立刻run一次initializer

3、（稍微复杂一点，视情况而定）用tensorflow实现单个数字判断或者数字分类。构建一个神经网络，用mnist手写数字训练集进行训练，然后判断测试集的数字是否是8，并给出判断的准确率。数字分类则是实现判断测试集数字是0~9中的哪一个。

加载图片的方式：

在“代码以及样本”文件夹中有个loadfun.py文件，自己创建文件时把它import进去，里面有两个函数，分别是

lf.loadImageSet()

lf.loadLabelSet()

参数填文件名，文件名要用双引号括起来

如果载入的是训练集图片那么输出将得到一个60000\*784的矩阵。按行取是样本，按列取是特征

相应的载入训练集标签得到的是一个60000\*1的矩阵，标签内容是图片对应的数字。

如果载入的是测试集图片那么输出将得到一个10000\*784的矩阵。按行取是样本，按列取是特征

相应的载入测试集标签得到的是一个10000\*1的矩阵，标签内容是图片对应的数字。