```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data   #手写数字相关的数据包
```

```
# 载入数据集
mnist = input_data.read_data_sets("MNIST_data",one_hot=True)    #载入数据，{数据集包路径，把标签转化为只有0和1的形式}

#定义变量，即每个批次的大小
batch_size = 100    #一次放100章图片进去
n_batch = mnist.train.num_examples // batch_size   #计算一共有多少个批次；训练集数量（整除）一个批次大小

#(在3-2基础上添加)命名空间
with tf.name_scope('input'):
    #定义两个placeholder
    x = tf.placeholder(tf.float32,[None,784],name='x-input')    #[行不确定，列为784]
    y = tf.placeholder(tf.float32,[None,10],name='y-input')    #数字为0-9，则为10

with tf.name_scope('layer'):
    #创建一个简单的神经网络
    with tf.name_scope('wights'):
        W = tf.Variable(tf.zeros([784,10]),name='W')   #权重
    with tf.name_scope('biases'):
        b = tf.Variable(tf.zeros([10]),name='b')     #偏置
    with tf.name_scope('wx_plus_b'):
        wx_plus_b = tf.matmul(x,W) + b
    with tf.name_scope('softmax'):
        prediction = tf.nn.softmax(wx_plus_b)    #预测

with tf.name_scope('loss'):
    #定义二次代价函数
    # loss = tf.reduce_mean(tf.square(y-prediction))
    loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y,logits=prediction))

with tf.name_scope('train'):
    #使用梯度下降法
    train_step = tf.train.GradientDescentOptimizer(0.2).minimize(loss)

#初始化变量
init = tf.global_variables_initializer()

with tf.name_scope('accuracy'):
    with tf.name_scope('correct_prediction'):
        #准确数，结果存放在一个布尔型列表中
        correct_prediction = tf.equal(tf.argmax(y,1),tf.argmax(prediction,1))   #比较两个参数大小是否相同，同则返回为true，不同则返回为false；argmax()：返回张量中最大的值所在的位置
```

```
    with tf.name_scope('accuracy'):
        #求准确率
        accuracy = tf.reduce_mean(tf.cast(correct_prediction,tf.float32))
#cast()：将布尔型转换为32位的浮点型；（比方说9个T和1个F，则为9个1，1个0，即准确率为90%）

#在3-2基础上更改
with tf.Session() as sess:
    sess.run(init)
    writer = tf.summary.FileWriter('logs/',sess.graph)
    for epoch in range(1):
        for batch in range(n_batch):
            batch_xs,batch_ys = mnist.train.next_batch(batch_size)
            sess.run(train_step,feed_dict={x:batch_xs,y:batch_ys})

        acc = sess.run(accuracy,feed_dict=
{x:mnist.test.images,y:mnist.test.labels})
        print("Iter" + str(epoch) + ",Testing Accuracy" + str(acc))

```

```
Extracting MNIST_data\train-images-idx3-ubyte.gz
Extracting MNIST_data\train-labels-idx1-ubyte.gz
Extracting MNIST_data\t10k-images-idx3-ubyte.gz
Extracting MNIST_data\t10k-labels-idx1-ubyte.gz
Iter0,Testing Accuracy0.8764
```