```python
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
from tensorflow.contrib.tensorboard.plugins import projector
```

```python
#载入数据
mnist = input_data.read_data_sets("MNIST_data",one_hot=True)  #把标签转化为只有0和1的形式
#运行次数
max_steps = 1001
#图片数量
image_num = 3000
#文件路径
DIR = "E:/jupyter/tensorflow/"

#定义会话
sess = tf.Session()

#载入图片
embedding = tf.Variable(tf.stack(mnist.test.images[:image_num]),trainable=False,name='embedding')  #stack为变换矩阵

#参数概要
def variable_summaries(var):
    with tf.name_scope('summaries'):
        mean = tf.reduce_mean(var)
        tf.summary.scalar('mean',mean) #平均值
        with tf.name_scope('stddev'):
            stddev = tf.sqrt(tf.reduce_mean(tf.square(var - mean)))
        tf.summary.scalar('stddev',stddev) #标准差
        tf.summary.scalar('max',tf.reduce_max(var)) #最大值
        tf.summary.scalar('min',tf.reduce_min(var)) #最小值
        tf.summary.scalar('histogram',var) #直方图

#(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')    #[行任意维度，列为784]
    #正确的标签
    y = tf.placeholder(tf.float32,[None,10],name='y-input')    #数字为0-9，则为10

#显示图片
with tf.name_scope('input_reshape'):
    image_shaped_input = tf.reshape(x,[-1,28,28,1])  #-1代表不确定的值，把784转换成28行28列，维度为1
    tf.summary.image('input',image_shaped_input,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
        variable_summaries(W)
    with tf.name_scope('biases'):
        b = tf.Variable(tf.zeros([10]),name='b')     #偏置
        variable_summaries(b)
    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))
    tf.summary.scalar('loss',loss)

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%）
        tf.summary.scalar('accuracy',accuracy)

#产生metadata文件
if tf.gfile.Exists(DIR + 'projector/projector/metadata.tsv'):
    tf.gfile.DeleteRecursively(DIR + 'projector/projector/metadata.tsv')  #如果有这个文件则将其删除
with open(DIR + 'projector/projector/metadata.tsv','w') as f:   #如果没有则采用写的方式生成这个文件
    labels = sess.run(tf.argmax(mnist.test.labels[:],1))   #argmax表示在哪一列元素中，它的哪个位置是最大的，格式为标记为1；如：如果为0则为：1000000000；如果为3则为：0010000000
    for i in range(image_num):
        f.write(str(labels[i] + '\n'))    #将label写入文件中，label间隔一行的格式

#合并所有的summary
merged = tf.summary.merge_all()


projector_writer = tf.summary.FileWriter(DIR + 'projector/projector',sess.graph)   #定义路径，图结构
saver = tf.train.Saver()
```

```python
config = projector.ProjectorConfig()  #定义配置项
embed = config.embeddings.add()
embed.tensor_name = embedding.name
embed.metadata_path = DIR + 'projector/projector/metadata.tsv'
embed.sprite.image_path = DIR + 'projector/data/mnist_10k_sprite.png'
embed.sprite.single_image_dim.extend([28,28])   #按照28*28像素进行切分
projector.visualize_embeddings(projector_writer,config)

for i in range(max_steps):
    #每个批次100个样本
    batch_xs,batch_ys = mnist.train.next_batch(100)
    run_options = tf.RunOptions(trace_level = tf.RunOption.FULL_TRACE)
    run_metadata = tf.RunMetadata()
    summary,_ = sess.run([merged,train_step],feed_dict={x:batch_xs,y:batch_ys},options=run_options,run_metadata=run_metadata)
    projector_writer.add_run_metadata(run_metadata,'step%03d'% i)
    projector_writer.add_summary(summary,i)

    if i%100 == 0:
        acc = sess.run(accuracy,feed_dict={x:mnist.test.images,y:mnist.test.labels})
        print("Iter" + str(i) + ",Testing Accuracy=" + str(acc))

saver.save(sess,DIR + 'projector/projector/a_model.ckpt',global_step=max_steps)
projector_writer.close()
sess.close()

```

```
WARNING:tensorflow:From <ipython-input-2-2569381befff>:2: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data\train-images-idx3-ubyte.gz
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data\train-labels-idx1-ubyte.gz
```

```
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
Extracting MNIST_data\t10k-images-idx3-ubyte.gz
Extracting MNIST_data\t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From D:\anaconda\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From <ipython-input-2-2569381befff>:56: softmax_cross_entropy_with_logits (from tensorflow.python.ops.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:

Future major versions of TensorFlow will allow gradients to flow
into the labels input on backprop by default.

See `tf.nn.softmax_cross_entropy_with_logits_v2`.
```

```

```