```
# split data
train_input, train_output, test_input, test_output, num_examples = sp.split(input, output, train_size, False)

# set tensorflow Graph Input
X = tf.placeholder(tf.float32, [None, num_input])
Y = tf.placeholder(tf.float32, [None, num_classes])

# set model weights
W = tf.Variable(tf.zeros([num_input, num_classes]))
b = tf.Variable(tf.zeros([num_classes]))

# construct a simple linear model with the softmax function
pred = tf.nn.softmax(tf.matmul(X, W) + b)

# cost function that we need to minimize using cross entropy
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(pred), reduction_indices=1))
# adam optimizer
optimizer = tf.train.AdamOptimizer(learning_rate).minimize(cost)

# evaluate the model
correct_pred = tf.equal(tf.argmax(pred, 1), tf.argmax(Y, 1))
# calculate accuracy
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32))
```