Queue：

* tf.FIFOQueue 按入列顺序出列的队列
* tf.RandomShuffleQueue 随机顺序出列的队列
* tf.PaddingFIFOQueue 以固定长度批量出列的队列
* tf.PriorityQueue 带优先级出列的队列
* tf.FIFOQueue(capacity, dtypes, shapes=None, names=None ...)

tf.train.slice\_input\_producer是一个tensor生成器，作用是按照设定，每次从一个tensor列表中按顺序或者随机抽取出一个tensor放入文件名队列。

slice\_input\_producer(tensor\_list, num\_epochs=None, shuffle=True, seed=None,

capacity=32, shared\_name=None, name=None)

第一个参数 tensor\_list：包含一系列tensor的列表，表中tensor的第一维度的值必须相等，即个数必须相等，有多少个图像，就应该有多少个对应的标签。

第二个参数num\_epochs: 可选参数，是一个整数值，代表迭代的次数，如果设置 num\_epochs=None,生成器可以无限次遍历tensor列表，如果设置为 num\_epochs=N，生成器只能遍历tensor列表N次。

第三个参数shuffle： bool类型，设置是否打乱样本的顺序。一般情况下，如果shuffle=True，生成的样本顺序就被打乱了，在批处理的时候不需要再次打乱样本，使用 tf.train.batch函数就可以了;如果shuffle=False,就需要在批处理时候使用 tf.train.shuffle\_batch函数打乱样本。

第四个参数seed: 可选的整数，是生成随机数的种子，在第三个参数设置为shuffle=True的情况下才有用。

第五个参数capacity：设置tensor列表的容量。

第六个参数shared\_name：可选参数，如果设置一个‘shared\_name’，则在不同的上下文环境（Session）中可以通过这个名字共享生成的tensor。

第七个参数name：可选，设置操作的名称

---------------------

f = tf.train.slice\_input\_producer()产生队列

sess.run(tf.global\_variables\_initializer())初始化

coord = tf.train.Coordinator()线程

threads = tf.train.start\_queue\_runners(sess=sess, coord=coord)启动任务

k = sess.run(f) 可就是获得的tensor

input\_queue = tf.train.slice\_input\_producer([images, label], shuffle=False)

image\_batch, label\_batch = tf.train.batch(input\_queue, batch\_size=batch\_size, num\_threads=1, capacity=64) 获得batch数据