tf.reshape(tensor, shape, name=**None**)

函数的作用是将tensor变换为参数shape的形式。 其中shape为一个列表形式，特殊的一点是列表中可以存在-1。

转换为一般的shape（也就是不涉及-1的）我这里就不说了，主要说一下对-1的理解。

-1代表的含义是不用我们自己指定这一维的大小，函数会自动计算，但列表中只能存在一个-1。

（当然如果存在多个-1，就是一个存在多解的方程了）

我理解的要点是：首先shape就是reshape变换后的矩阵大小，先不管-1的那一个维度，先看其它维度，然后用原矩阵的总元素个数除以确定的维度，就能得到-1维度的值。

我们来看例子。

M=np.array([[[[1,2,3]]],[[[4,5,6]]],[[[7,8,9]]]]) *#M是[3,1,1,3]的四维矩阵*

我想把M重组成若干个3维的向量，那么直接tf.reshape(M,[-1,3])

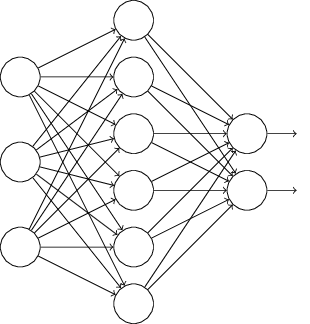
那么会得到几个3维向量呢？ M一共有9个元素，9/3=3，那么得到3个三维向量，那么结果就是[3,3]的矩阵。

那么我想得到若干个[3,3]的矩阵，那么我们tf.reshape(M,[-1,3,3])

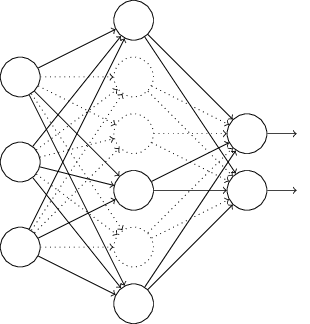
那么结果就是[1,3,3]的矩阵

* **dropout率的选择**
  + 经过交叉验证，隐含节点dropout率等于0.5的时候效果最好，原因是0.5的时候dropout随机生成的网络结构最多。
  + dropout也可以被用作一种添加噪声的方法，直接对input进行操作。输入层设为更接近1的数。使得输入变化不会太大（0.8）
* 训练过程
  + 对参数w的训练进行球形限制(max-normalization)，对dropout的训练非常有用。
  + 球形半径c是一个需要调整的参数。可以使用验证集进行参数调优
  + dropout自己虽然也很牛，但是dropout、max-normalization、large decaying learning rates and high momentum组合起来效果更好，比如max-norm regularization就可以防止大的learning rate导致的参数blow up。
  + 使用pretraining方法也可以帮助dropout训练参数，在使用dropout时，要将所有参数都乘以1/p。

Dropout 技术：Dropout是一个同正则化完全不同的技术，与L1和L2范式正则化不同。dropout并不会修改代价函数而是修改深度网络本身。在我描述dropout的工作机制和dropout导致何种结果前，让我们假设我们正在训练如下一个网络。



特别的。假设我们有一个输入xx并且相关的输入yy的训练。通常的我们将首先通过前馈网络把xx输入我们随机初始化权重后的网络。然后反向传播拿到对梯度的影响。也就是根据误差，根据链式法则反向拿到对相应权重的偏微分。但是，使用dropout技术的话。相关的处理就完全不同了。在开始训练的时候我们随机的（临时）删除一般的神经元。但是输入层和输出层不做变动。对深度网络dropout后。我们将会得到下图中这样类似的网络。注意。下图中的虚线存在的网络就是我们临时删除的。



 我们前向传播输入项xx通过修改后的网络。然后反向传播拿到的结果通过修改后的网络。对此昨晚一个样本化的迷你批次的样本后。我们更新相应的权重和偏置。这样重复迭代处理。首先存储dropout的神经元，然后选择一个新的随机隐层神经元的子集去删除。估计不同样本批次的梯度。最后更新网络的权重和偏置。

通过不断的重复处理。我们的网络将会学到一系列的权重和偏置参数。当然这些参数是在一半的隐层神经元被dropped out（临时删除的）情况下学习到的。当我们真正的运行整个神经网络的时候意味着两倍多的隐层神经元将被激活。为了抵消此影响。我将从隐层的权重输出减半。

dropout处理看起来是奇怪并且没有规律的。为什么我们希望他对正则化有帮助呢。来解释dropout到底发生了什么。我们先不要思考dropout技术。而是想象我们用一个正常的方式训练一个神经网络。特别的。假设我们训练了几个完全不同的神经网络。用的是完全相同的训练数据。当然了。因为随机初始化参数或其他原因。训练得到的结果也许是不同的。当这种情况发生的时候，我们就可以平均这几种网络的结果，或者根据相应的规则决定使用哪一种神经网络输出的结果。例如。如果我们训练了五个网络。其中三个分类一个数字为3，最终的结果就是他是3的可能性更大一些。其他的两个网络也许有些错误。这种平均的架构被发现通常是十分有用的来减少过拟合。（当然这种训练多个网络的代价也是昂贵的。）出现这种结果的原因就是不同的网络也是在不同的方式上过你和。通过平均可以排除掉这种过拟合的。

这种现象与dropout这种技术有什么作用的。启发式的我们发现。dropout不同设置的神经元和我们训练几种不同的神经网络很像。因此，dropout处理很像是平均一个大量不同网络的平均结果。不同的网络在不同的情况下过拟合。因此，很大程度上。dropout将会减少这种过拟合。