Tensorflow 实战Google实战深度学习：

架构：输入，隐藏层，输出。

其中主要是对隐藏层的设计。**隐藏层可以看作是从输入中提取更高纬的特征，这样面对复杂的问题，特征越多判断越准确。**

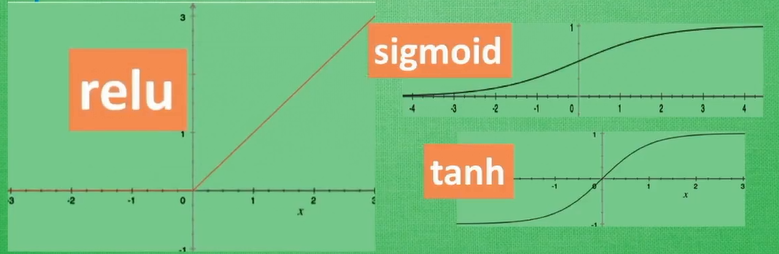
初始完架构，隐藏层各参数值是随机产生。输入测试参数batch，经过隐藏层到输出，输出的值与标准值进行对比得到loss损失值，朝着损失值减小的方向调整参数值（反向传播更新变量），到最后隐藏层各参数值调整到一个较合适的值，使得输出值与标准值接近。

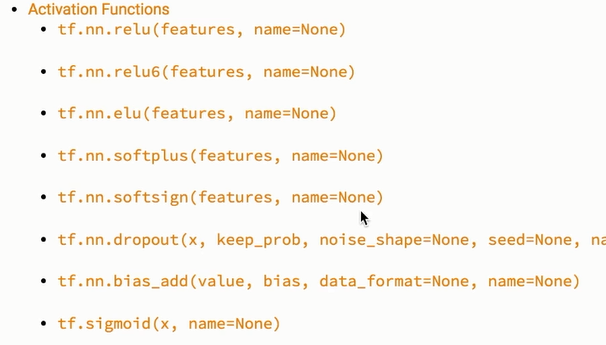
训练后的框架、参数值就是我们要保存使用的。

第四章：深度神经网络

多层与非线性化是深度学习的重要特征。

1.激活函数用来去线性化

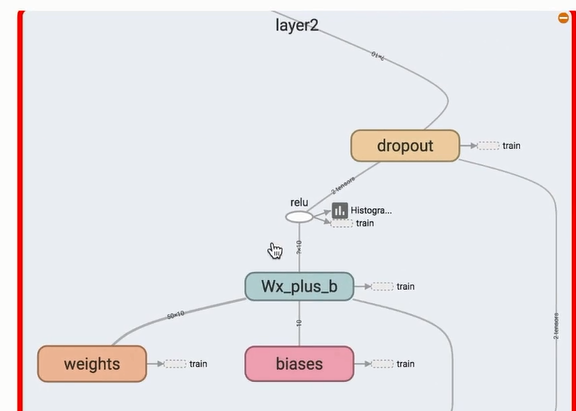




Relu函数把输出小于0的数取值为0，这样输出与输入就不满足线性化，去线性化函数有以上几种，不同情况使用不同去线性化函数。

a = tf.nn.relu(tf.matmul(x, w1) + biases1)

作用点：位于隐藏层layer2输出，筛选输出值到预测值那边



2.损失函数

损失函数由标准值与输出层的预测值比较得到。它的作用是用来给神经网络作为输入，而神经网络的作用是调整隐藏层参数，使损失函数得到的值越来越小。类似人类的学习反馈，从失败中得到经验向成功靠拢。

Cross\_entry = tf.nn.softmax\_cross\_entropy\_with\_logits(y,y\_)

Y为输出值，y\_为标准值。Softmax回归之后的交叉函数损失函数，其中y输入的行序列满足相加为1的概率序列，y\_不用满足，应为softmax会对其操作，使其满足概率分布，适用于分类问题。而回归问题预测，如销量预测，最常用的损失函数是均方差：

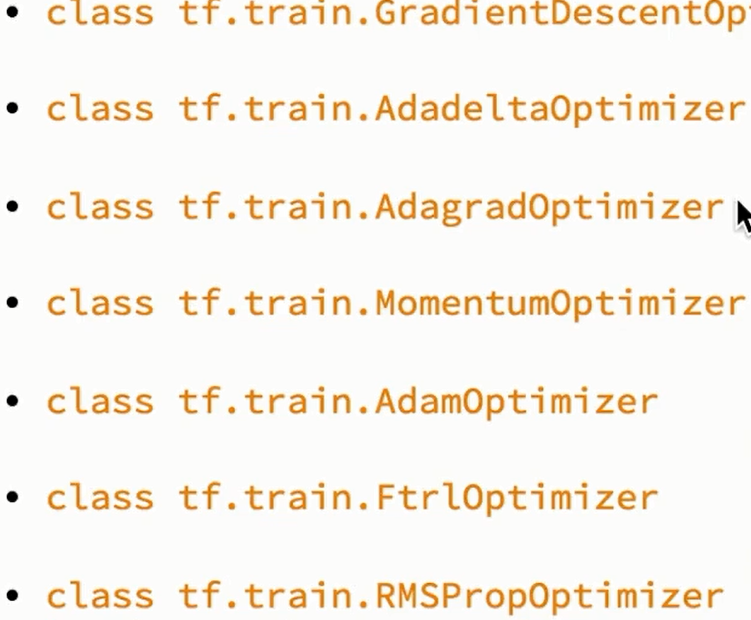
Mse = tf.reduce\_mean(tf.square(y\_-y))。

损失函数也可以自己定义

Loss= tf.reduce\_sum(tf.select(tf.greater(y,y\_),(y-y\_)\*loss\_more,(y-y\_)\*loss\_less))

3.神经网络优化算法

方式有:反向传播算法和梯度下降算法



参数的优化基于学习效率一次改变参数大小的幅度的多少，不同优化函数考虑的因素有差异，但都是基于损失函数的趋势做改变。

Train\_step = tf.train.AdamOptimizer(learn\_reat).minimize(loss)

4.学习效率

神经网络的优化效率由学习效率来控制，如果学习效率一样，那么参数每次变化的幅度将差不多，这将导致优化到正确区域很难让正确值更精进一步，所以学习效率值的大小应随着训练增加而减小，(如同粗调与精调)。

Learn\_reat= tf.train.exponential\_decay(learn\_reat,global\_step,100,0.96,staircase=True)

5.过拟合

就是对未来过度自信。对训练值过度依赖，导致在判断未出现的情况而loss过大。

用正则化可避免该问题。

这就需要我们在加入训练参数时，向其中加入不确定的随机值。这个随机值加在计算出来的损失函数中。

所以损失值有两部分组成

Weight=

tf.get\_variable(“weight”,shape,initializer=tf.truncated\_normal\_initializer(stddev=0.1))

Weight满足正态分布。

Loss = tf.reduce\_mean(tf.square(y\_-y)+tf.contrib.layers.l2\_regularizer(权重)（weight）。

而损失值往往不只有一个，因为每一层都有，所以需要自定义一个损失集合，方便操作

Tf.add\_to\_collection(‘losses’,正则化损失的值)

Loss= tf.reduce\_mean(tf.square(y\_-y) +tf.add\_n(tf.get\_collection(‘losses’)

## if train : fc1 = tf.nn.dropout(fc1,0.5)

##以50%概率随机保留输出值，防止过拟合

6.滑动平均模型

Step = tf.Variable(0,trainable = False)

Ema = tf.train.ExponentialMovingAverage(0.99,step)

Maintain\_averages\_op = ema.apply([v1])

**滑动平均值数学含义及作用不懂**，只知道每次运行maintain\_averages\_op会根据滑动平均值的数学公式来改变该表格里v1的值。

滑动平均模型一定程度上控制模型的更新速度