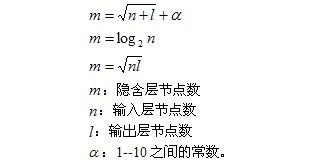
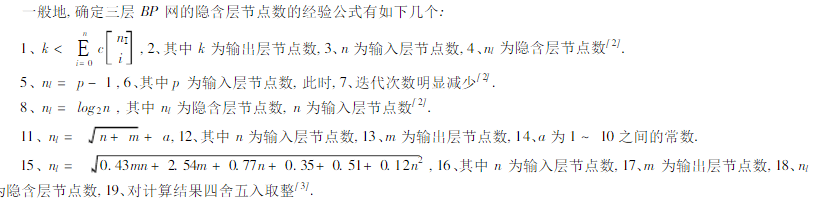
======================================================

网络配置（初始化）

1、查看隐含层节点数确定 经验公式 ----网页。





①一种情况是，只有一个节点。那么这个方程就变成了一个非常简单的线性方程y=ax+b的形式。其中a和b就是权值和偏置。在这种情况下，如果你有一个非线性的系统，那么肯定，这个预测的偏差就会非常大。无法完全的呈现出你样本的关联特性

另外一种情况，节点的数量无穷多。因为你的样本数量是有限的，那么如果想要达到一个比较好的优化，就会出现无限多的权值和偏重组合。呈现出极大的随机性。

②对不同问题应该分别对待，隐层数目不一定越多越好，比如预测问题，是希望网络有误差以增强泛化能力的。

实例结果表明:双隐层网络的两个隐层节点数相近时,网络训练效果最好. 进行工程试凑时可以使用构造法或删除法。

2、权值矩阵初始化 ，这句python代码的意思是否是使矩阵元素在 -0.1~0.1 之间。

w1 = 0.2\*np.random.random((inp\_num, hid\_num))- 0.1 # 初始化输入层权矩阵

注：random.random()用于生成一个0到1的随机符点数: 0 <= n < 1.0

3、此处偏置向量是否就是阈值theta

hid\_offset = np.zeros(hid\_num) #隐含层偏置向量

解答：是的（观察代码，网络训练，权重跟新）

4、

inp\_lrate = 0.2 #输入层-隐含层 的权重学习率

hid\_lrate = 0.2 #隐含层-输出层 的权重学习率

根据什么设置？

答：学习率η∈（0，1）控制着沿反梯度方向下降的步长，若步长太大则下降太快容易产生震荡，若步长太小则收敛速度太慢，一般地常把η设置为0.1，有时更新权重时会将输出层与隐含层设置为不同的学习率。

5、激活函数

sigmoid()函数

根据公式推导，连接权值W、b等的优化步长不仅与学习率有关，还和激活函数的导数有关。误差函数（误差=实际输出（激活函数（输入））- 预测输出）求导，则权值的优化快慢与误差函数的选取、激活函数的选取有关。

详见 印象笔记中 18校招中BP算法内容。

6、统计训练准确率时、

if np.argmax(out\_act) == label[count]:

temp+=1

print('Train\_Set Error is: %.2f%%'%((1-float(temp)/len(sample))\*100))