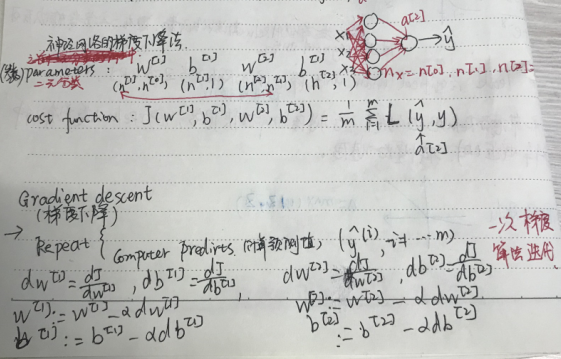
**2019.1.22总结**

**神经网络的梯度下降法：**

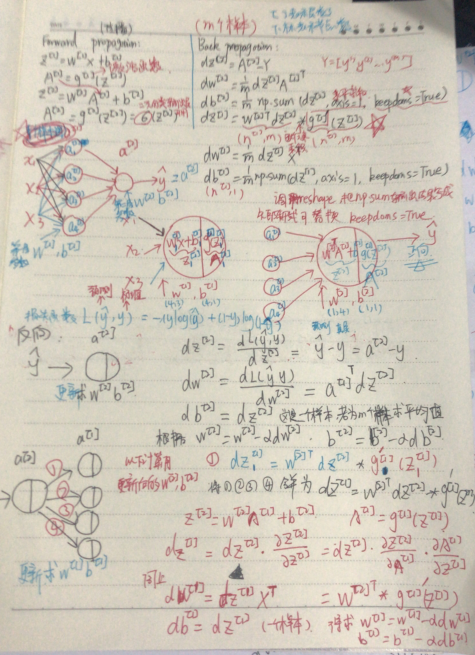
Parameters (参数)



**正向传播与反向传播算法：**

Keepdoms=True用法，保持对应矩阵维度，这里keepdoms是（n,）,使得求和后的矩阵保持（n,1）的维度。

损失函数是机器学习中用来衡量模型预测值F(x)和真实值Y之间的不一致程度的，通常越小越好。



**初始化：**[**https://github.com/stormstone/deeplearning.ai**](https://github.com/stormstone/deeplearning.ai)**课程作业**

 =np.random.randn((2,2))\*0.01 产生参数为（2,2）高斯分布随机变量，\*0.01是为了将权重初始化成很小的随机数。为什么不取\*100等很大的数，因为权重越大，z越大，在tanh，sigmoid函数中，对应斜率越小，梯度算法很慢，学习速度很慢。在单层的网络可以使用0.01，在很多深层的网络还可能用其它常数

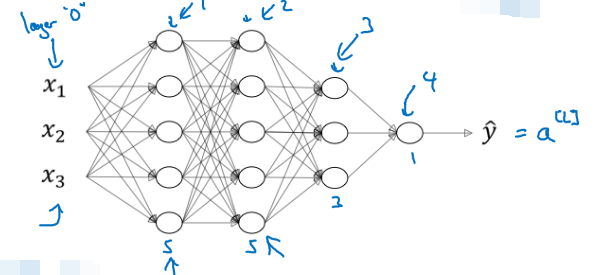
=np.zero((2,1)) b没有对称性问题，b初始化可以为0，因为只要w随机初始化，一开始还是用不同的隐藏单元计算不同的函数。

 =np.random.randn((2,2))\*0.01

=np.zero((2,1))

第四周深层神经网络

4 layer NN

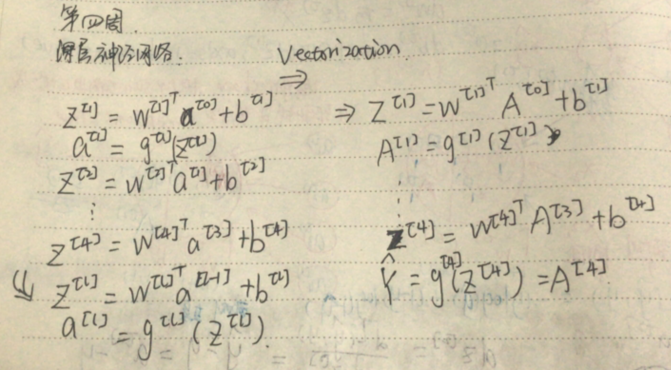


L=4(#layers)

 =#unit in layer l 在l层有几个节点（隐藏单元）

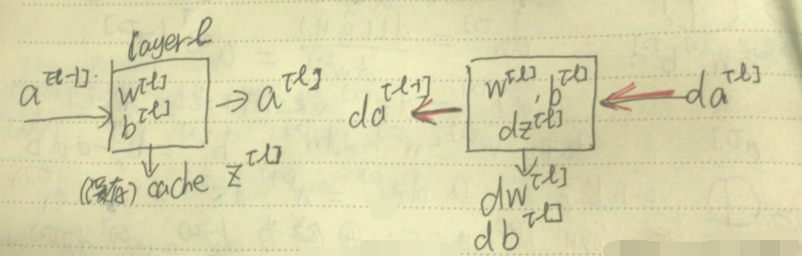
 = activations in layer l 在l层中的激活函数

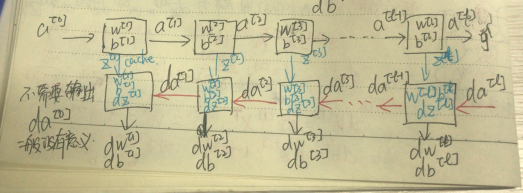
 和前面的定义相同



矩阵的维度： m个样本对应w,b,dw,db不变

**搭建神经网络块：**





每一层w,b都在更新