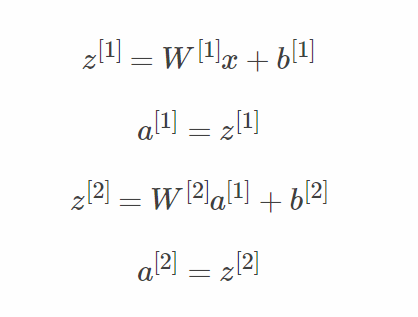
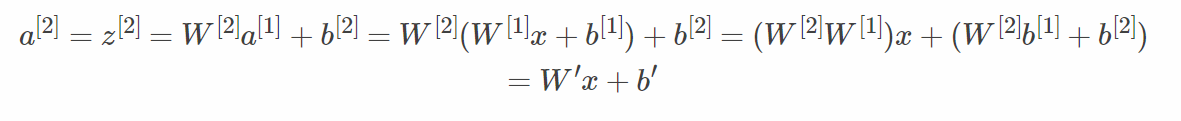
1. **为什么要选择非线性函数：**

对于两层的神经网络（只有一个隐藏层），假设所有的激活函数都是线性的，为了简化计算，我们直接令激活函数g(z)=z，即a=z。那么，浅层神经网络的各层输出为：



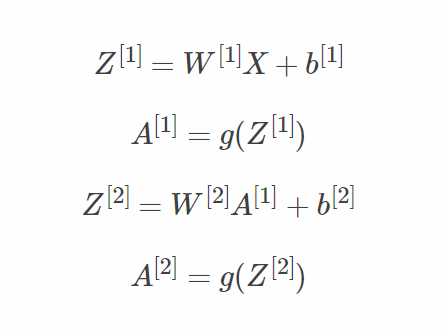
再对上式中a[2]进行化简计算：



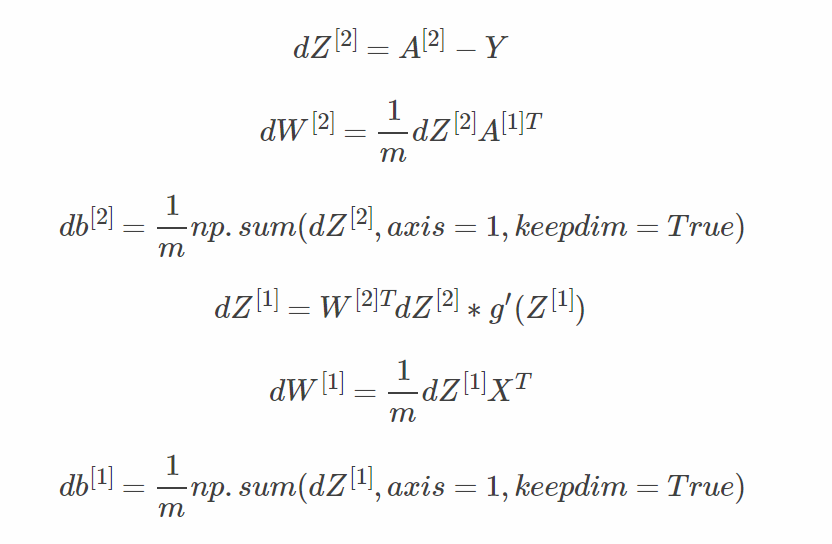
**结论**：经过推导发现a[2]仍是输入变量x的线性组合。这表明，使用神经网络与直接使用线性模型的效果并没有什么两样。即便是包含多层隐藏层的神经网络，如果使用线性函数作为激活函数，最终的输出仍然是输入x的线性模型。这样的话神经网络就没有任何作用了。因此，隐藏层的激活函数必须要是非线性的。

1. **正反向传播公式（m个样本）时向量化：**

**正向公式：**



**反向公式：**

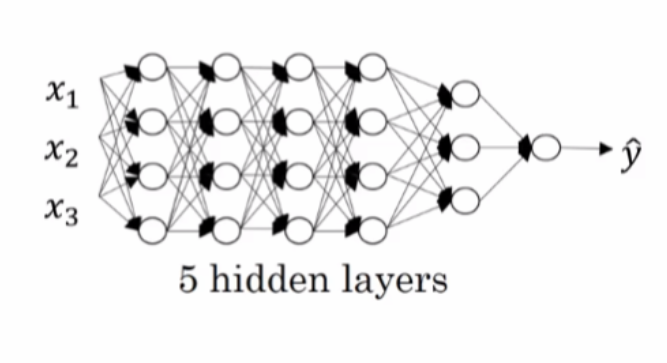


1. **随机初始化：**

**对于w来说**：神经网络模型中的参数W是不能全部初始化为零的。

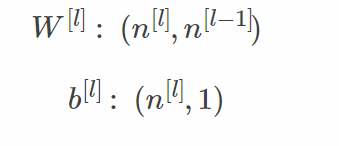
原因在于：设置W[1]和W[2]全为0，这样的结果是隐藏层两个神经元对应的行向量W1[1]和W2[1]每次迭代更新都会得到完全相同的结果，W1[1]始终等于W2[1]，完全对称。这样隐藏层设置多个神经元就没有任何意义了。值得一提的是，参数b可以全部初始化为零，并不会影响神经网络训练效果。

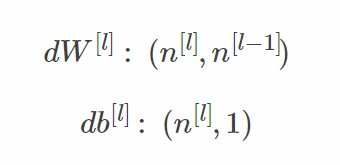
**深层神经网络**

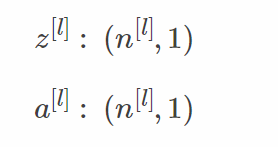


1.例如，该神经网络为有6层的深层神经网络，用n[0]表示第一层输入层的x的维数，n[1]表示该隐藏层神经元数目，以此类推到最后输出层为n[l]表示最后一层神经元数目，l为该神经元层数。

**单个样本，l层的深度神经网络**中参数的**矩阵维度**表示：

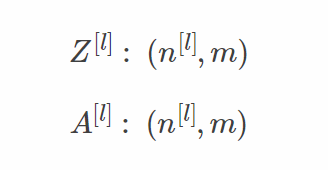






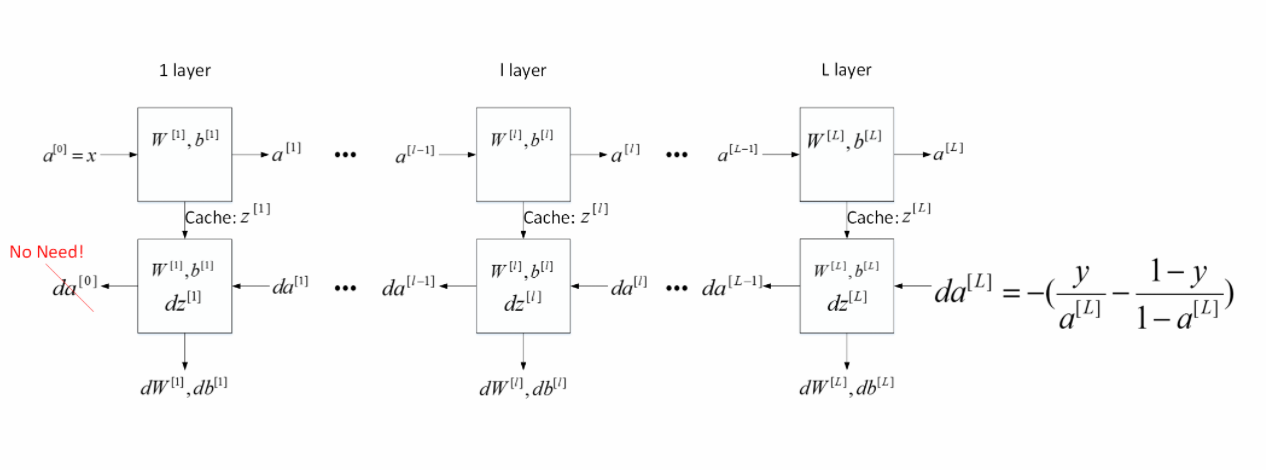
**m个样本时：**

**只有Z和A改变了维度。**



1. **流程图块：**

**正向和反向传播流程图块：**



正向为：输入：a[l−1]

输出：a[l]

参数：W[l],b[l]

缓存变量：z[l]

反向：输入：da[l]

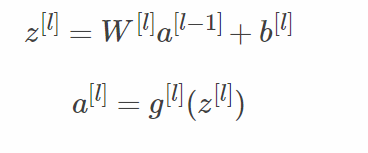
输出：da[l−1],dW[l],db[l]

参数：W[l],b[l]

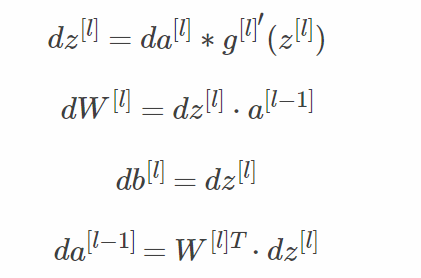
**l层神经网络正反向传播表达式：**

**单个样本时：**

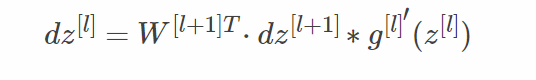
**正向：**



**反向：**

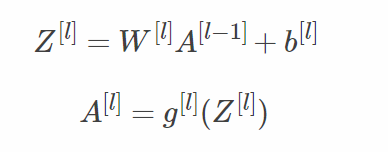


将第四个式子中变为da[l],W[l+1],dz[l+1]时，将其带入一式，可得出：



**m个样本时向量化形式:**

**正向：**



**反向：**

