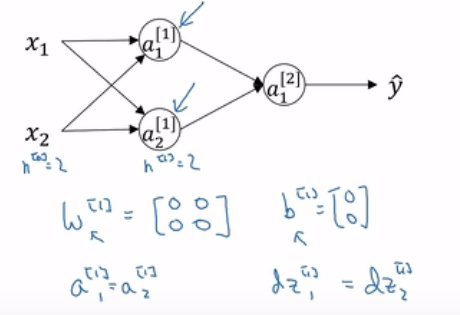
## P37 参数随机初始化



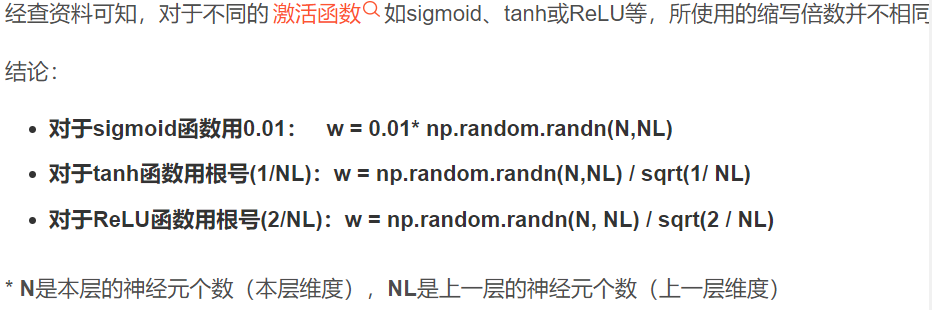
如果不进行随机而将参数w全置为零（b全置0并不影响）。则不同的神经元却进行着相同的计算。反向传播时的Dz1[1] ,Dz2[1]也会完全相同。

无论经过几轮迭代，会发现w的各个行完全相同，等同于一个神经元所实现的功能。

**python实现初始化参数：**

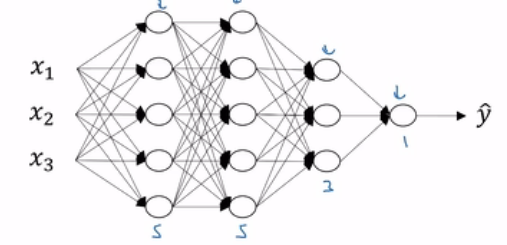
|  |
| --- |
| **w[1] = np.random.randn((2,2)) \* 0.01** |
| **b[1] = np.zero((2,1))** |
| **w[2]等同上进行初始化，如本例**  **w[2] = np.random.randn((1,2)) \* 0.01** |

**补充**：参数过大会导致z计算值也过大，在经过激活函数后斜率接近0。因此常常\*0.01等非常小的量进行初始化。在仅有一层隐藏层的神经网络中，0.01很合适。



## P38 lan Goodfellow访谈

## P39 深层神经网络



深层神经网络的表示：

x：输入特征，也是第0层的激活函数，即x=a[0]；

l：神经网络的层数；

n[l]：第l层神经元的个数；

a[l]：第l层的激活函数，a[l] = g(z[l])，其中最后一层的a[l] = 预测值y\_hat；

w[l]：计算第l层中间值z[l]时的权重；

b[l]：计算第l层中间值z[l]时的偏置参数；

## P40 深层神经网络的前向传播

单个样本x：

z[l] = w[l] a[l-1] + b[l]

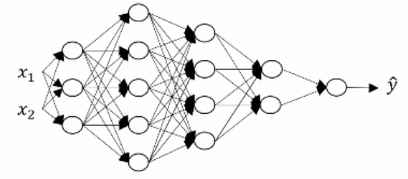
a[l] = g(z[l])

多个样本X：

Z[l] = w[l] A[l-1] + b[l] (w不用大写 因为维度没变，b不用大写因为广播机制)

A[l] = g(Z[l])

## P41 正确的矩阵维度



l = 5；

**（单样本情况下）**：

z[1]是第一级隐藏层的激活矢量 因此为3\*1即（n[1],1）矩阵。

x = a[0] 有两个输入x1，x2因此为 2\*1 即（n[0],1）矩阵。

则w应满足（n[1],1） = （w的维度）\*（n[0],1） 即w维度为（n[1] ,n[0]）。一般化w[l]的维度为（n[l] ,n[l-1]）。

b[1] 为3\*1向量和z[1]保持一致。一般化b[l]的维度为（n[l] ,1）。

反向传播时 dw维度和w维度一致 dz，da维度和z维度一致等等。

**（m样本情况下）**：

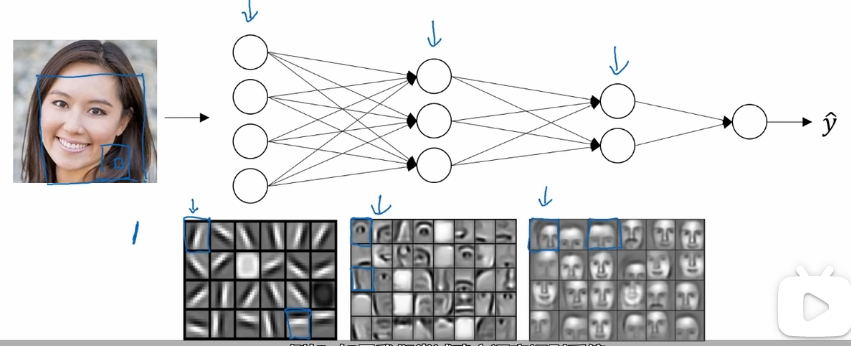
反向传播时 dw维度和w维度依旧一致 dZ，dA维度和Z维度一致。

Z[1]是（n[1],m）矩阵。X为（n[0],m）矩阵。

w[l]维度不变仍为（n[l] ,n[l-1]），b[l]维度不变，由于广播机制仍为（n[l] ,1）。

## P42 为什么深度网络deep networks这么有效

Eg1: CNN中，不同层发挥不同功能，通过多层先进行图像边缘检测，然后进行组合，最后合成出人脸。其他神经网络也类似。

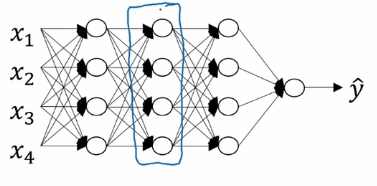


Eg2：语音识别，第一层检测低级音波特征，比如音调是否升高或降低。白噪声还是滑动声（sniffling sound）。然后通过组合低级波形，就可以检测基本的声音单位，专业名词为音素（phonemes），如 cat中 c ，a ，t的发音称为一个音素。再把因素组合在一起，识别出语音中的单词，最后再成句。

如果不进行多层的神经网络，那么浅层网络需要更多，往往是指数级增长的神经元。比如计算（x1，x2，…xn）的异或，单层串行相比多层树形结构O（logn）要耗费更多的神经元。

深度学习原指有很多隐藏层的神经网络，但现在已经泛指很多东西了，因此把有很多隐藏层的神经网络称为深度网络（deep networks）

## P43 构建神经网络

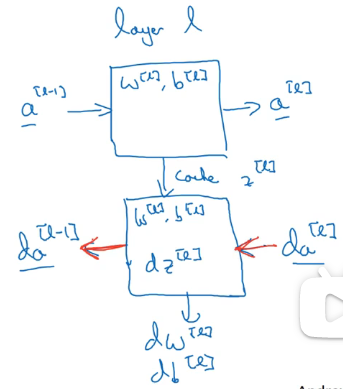


**正向传播时：**对于第L层，有参数w[L],b[L]。和输入a[L-1],以及输出a[L].

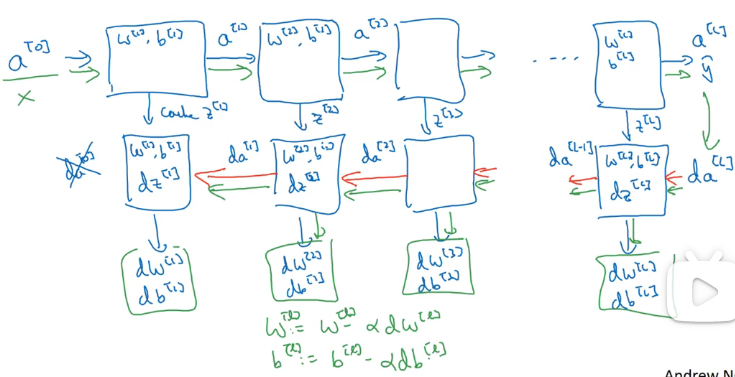
其中z[L] = w[L] a[L-1] + b[L], a[L] = g(z[L]).

同时输出z[L]的值作缓存有利于反向传播的计算。

**反向传播时：**对于第L层，输入da[L], 有参数w[L],b[L]并通过w[L],b[L]计算dz[L]，输出­­da[L-1]。输入实际还包含了缓存的z[L]值。如下图所示。

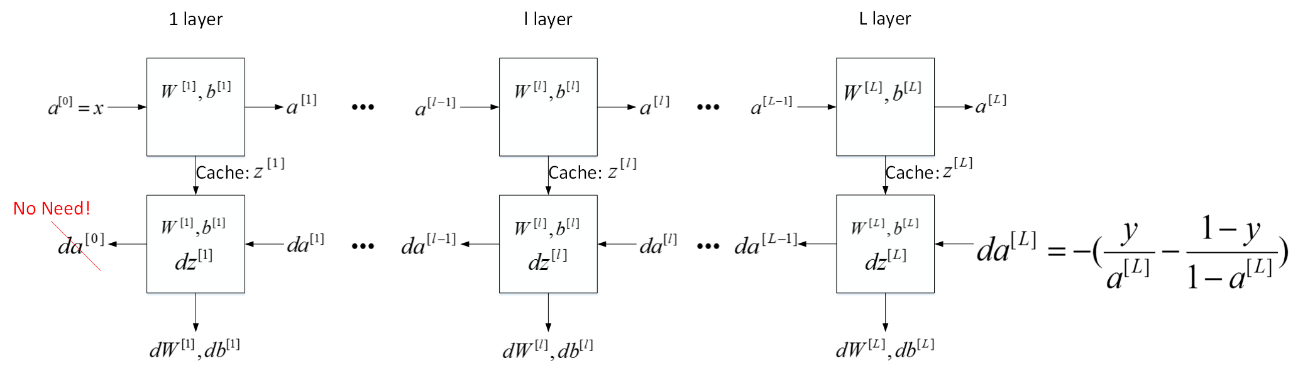


整个过程示意图。



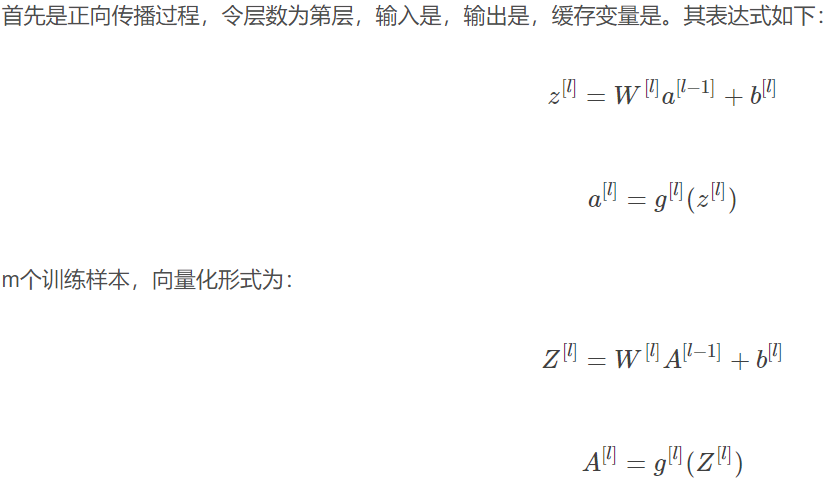
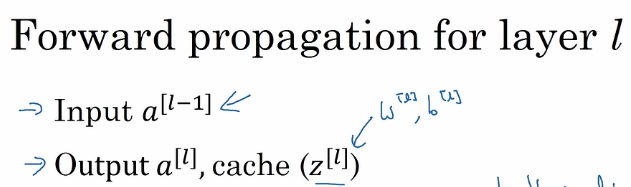
Ps：da[0]是输入特征的导数，没有必要计算。通过Cache 也可以在反向传播中快速得到 w[L],b[L]的值。

根据dw[L] 和 db[L] 进行对应层参数的更新。

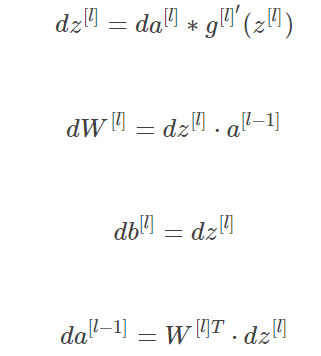


Ps最后一层用的是sigmoid激活函数

## P44 正向传播和反向传播



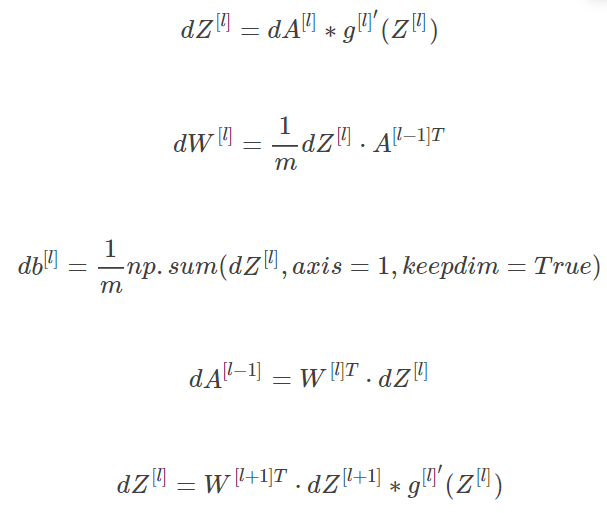
反向传播过程：单样本



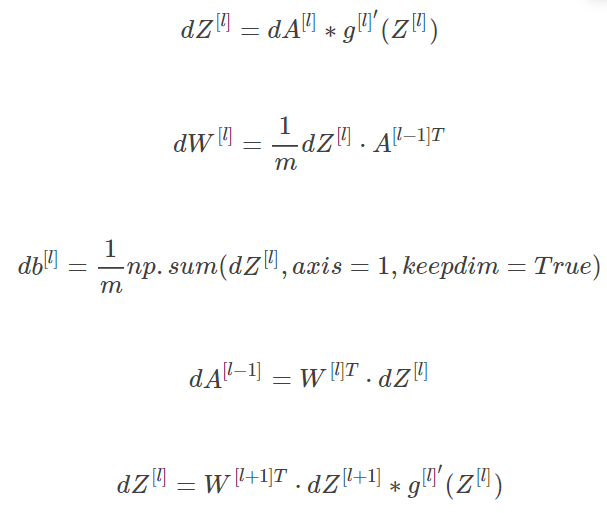
Ps：第二行的a[l-1]少了转置

结合1，4推出。

多样本：



对于m个样本，反向传播向量化后为



## P45 参数和超参数

参数：w，b

超参数：学习率。迭代次数，网络层数，每层的结点数，激活函数的选择。通过超参数影响真正的参数。

如何设置最优的超参数是一个比较困难的、需要经验知识的问题。通常的做法是选择超参数一定范围内的值，分别代入神经网络进行训练，测试cost function随着迭代次数增加的变化，根据结果选择cost function最小时对应的超参数值。

## P46 与人脑的关系

略。。

## Summary

本节课主要介绍了深层神经网络，是上一节浅层神经网络的拓展和归纳。首先，我们介绍了建立神经网络模型一些常用的标准的标记符号。然后，用流程块图的方式详细推导正向传播过程和反向传播过程的输入输出和参数表达式。我们也从提取特征复杂性和计算量的角度分别解释了深层神经网络为什么优于浅层神经网络。接着，我们介绍了超参数的概念，解释了超参数与参数的区别。最后，我们将神经网络与人脑做了类别，人工神经网络是简化的人脑模型。