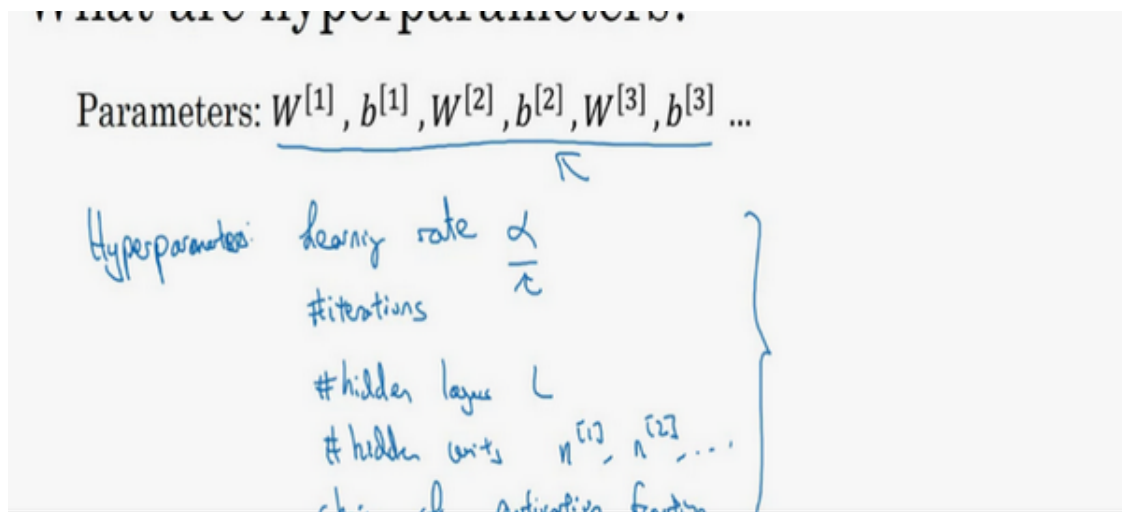


4.7 参数 VS 超参数 (Parameters vs Hyperparameters)

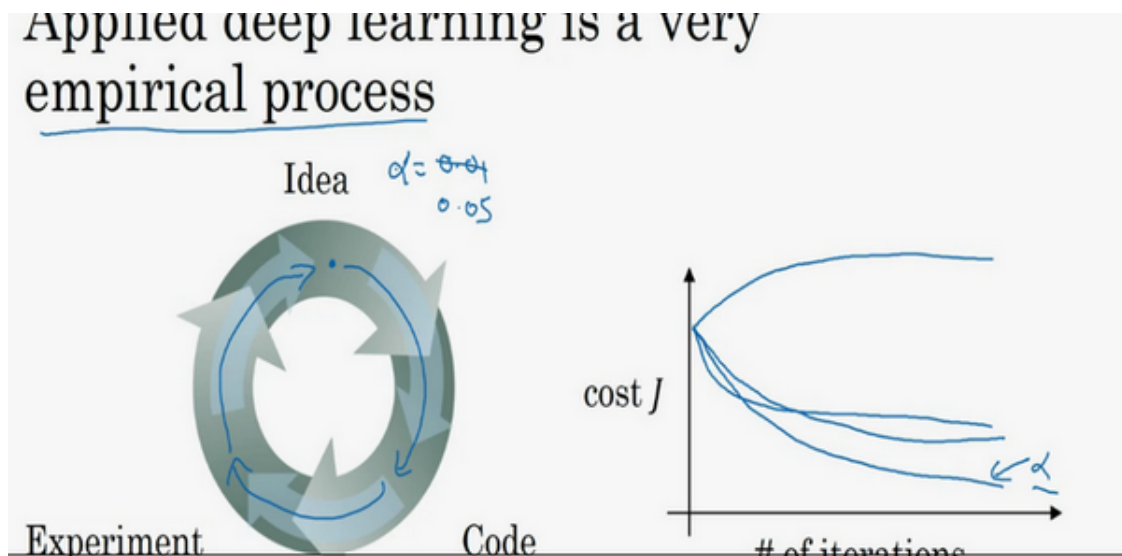
什么是超参数？

比如算法中的 **learning rate** α (学习率)、**iterations**(梯度下降法循环的数量)、 L (隐藏层数目)、 $n^{[l]}$ (隐藏层单元数目)、**choice of activation function** (激活函数的选择) 都需要你来设置，这些数字实际上控制了最后的参数 W 和 b 的值，所以它们被称作超参数。

实际上深度学习有很多不同的超参数，之后我们也会介绍一些其他的超参数，如 **momentum**、**mini batch size**、**regularization parameters** 等等。



如何寻找超参数的最优值？



走 **Idea—Code—Experiment—Idea** 这个循环，尝试各种不同的参数，实现模型并观察是否成功，然后再迭代。

今天的深度学习应用领域，还是很经验性的过程，通常你有个想法，比如你可能大致知道一个最好的学习率值，可能说 $\alpha = 0.01$ 最好，我会想先试试看，然后你可以实际试一下，训练一下看看效果如何。然后基于尝试的结果你会发现，你觉得学习率设定再提高到 0.05 会比较好。如果你不确定什么值是最好的，你大可以先试试一个学习率 α ，再看看损失函数 J 的值有没有下降。然后你可以试一试大一些的值，然后发现损失函数的值增加并发散了。然后可能试试其他数，看结果是否下降的很快或者收敛到在更高的位置。你可能尝试不同的 α 并观察损失函数 J 这么变了，试试一组值，然后可能损失函数变成这样，这个 α 值会加快学习过程，并且收敛在更低的损失函数值上（箭头标识），我就用这个 α 值了。

在前面几页中，还有很多不同的超参数。然而，当你开始开发新应用时，预先很难确切知道，究竟超参数的最优值应该是什么。所以通常，你必须尝试很多不同的值，并走这个循环，试试各种参数。试试看 5 个隐藏层，这个数目的隐藏单元，实现模型并观察是否成功，然后再迭代。这页的标题是，应用深度学习领域，一个很大程度基于经验的过程，凭经验的过程通俗来说，就是试直到你找到合适的数值。

另一个近来深度学习的影响是它用于解决很多问题，从计算机视觉到语音识别，到自然语言处理，到很多结构化的数据应用，比如网络广告或是网页搜索或产品推荐等等。我所看到过的就有很多其中一个领域的研究员，这些领域中的一个，尝试了不同的设置，有时候这种设置超参数的直觉可以推广，但有时又不会。所以我经常建议人们，特别是刚开始应用于新问题的人们，去试一定范围的值看看结果如何。然后下一门课程，我们会用更系统的方法，用系统性的尝试各种超参数取值。然后其次，甚至是你已经用了很久的模型，可能你在做网络广告应用，在你开发途中，很有可能学习率的最优数值或是其他超参数的最优值是会变的，所以即使你每天都在用当前最优的参数调试你的系统，你还是会发现，最优值过一年就会变化，**因为电脑的基础设施，CPU 或是 GPU 可能会变化很大。所以有一条经验规律可能每几个月就会变。**如果你所解决的问题需要很多年时间，只要经常试试不同的超参数，勤于检验结果，看看有没有更好的超参数数值，相信你慢慢会得到设定超参数的直觉，知道你的问题最好用什么数值。

这可能的确是深度学习比较让人不满的一部分，也就是你必须尝试很多次不同可能性。但参数设定这个领域，深度学习研究还在进步中，所以可能过段时间就会有更好的方法决定超参数的值，也很有可能由于 **CPU、GPU、网络和数据**都在变化，这样的指南可能只会在这段时间内起作用，只要你不断尝试，并且尝试保留交叉检验或类似的检验方法，然后挑一个对你的问题效果比较好的数值。

近来受深度学习影响，很多领域发生了变化，从计算机视觉到语音识别到自然语言处理到很多结构化的数据应用，比如网络广告、网页搜索、产品推荐等等；有些同一领域设置超参数的直觉可以推广，但有时又不可以，特别是那些刚开始研究新问题的人们应该去尝试一定范围内的结果如何，甚至那些用了很久的模型得学习率或是其他超参数的最优值也有可能改变。

在下个课程我们会用系统性的方法去尝试各种超参数的取值。有一条经验规律：经常试试不同的超参数，勤于检查结果，看看有没有更好的超参数取值，你将会得到设定超参数的直觉。

Note that the formulas shown in the next video have a few typos. Here is the correct set of formulas.

$$dZ^{[L]} = A^{[L]} - Y$$

$$dW^{[L]} = \frac{1}{m} dZ^{[L]} A^{[L-1]T}$$

$$db^{[L]} = \frac{1}{m} np.sum(dZ^{[L]}, axis = 1, keepdims = True)$$

$$dZ^{[L-1]} = W^{[L]T} dZ^{[L]} * g'^{[L-1]}(Z^{[L-1]})$$

Note that * denotes element-wise multiplication)

⋮

$$dZ^{[1]} = W^{[2]} dZ^{[2]} * g'^{[1]}(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]T}$$

Note that $A^{[0]T}$ is another way to denote the input features, which is also written as X^T

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

4.8 深度学习和大脑的关联性(What does this have to do with the brain?)

深度学习和大脑有什么关联性吗？

关联不大。

那么人们为什么会说深度学习和大脑相关呢？

当你在实现一个神经网络的时候，那些公式是你在做的东西，你会做前向传播、反向传播、梯度下降法，其实很难表述这些公式具体做了什么，深度学习像大脑这样的类比其实是过度简化了我们的大脑具体在做什么，但因为这种形式很简洁，也能让普通人更愿意公开讨论，也方便新闻报道并且吸引大众眼球，但这个类比是非常不准确的。

一个神经网络的逻辑单元可以看成是对一个生物神经元的过度简化，但迄今为止连神经科学家都很难解释究竟一个神经元能做什么，它可能是极其复杂的；它的一些功能可能真的类似 **logistic** 回归的运算，但单个神经元到底在做什么目前还没有人能够真正可以解释。

深度学习的确是个很好的工具来学习各种很灵活很复杂的函数，学习到从 x 到 y 的映射，在监督学习中学习到输入到输出的映射。

Forward and backward propagation

$$\begin{aligned} Z^{[1]} &= W^{[1]}X + b^{[1]} \\ A^{[1]} &= g^{[1]}(Z^{[1]}) \\ Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\ A^{[2]} &= g^{[2]}(Z^{[2]}) \\ &\vdots \\ A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{y} \end{aligned}$$

$$\begin{aligned} dZ^{[L]} &= A^{[L]} - Y \\ dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]T} \\ db^{[L]} &= \frac{1}{m} \text{np.sum}(dZ^{[L]}, \text{axis} = 1, \text{keepdims} = \text{True}) \\ dZ^{[L-1]} &= dW^{[L]T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\ &\vdots \\ dZ^{[1]} &= dW^{[L]T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\ dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]T} \\ db^{[1]} &= \frac{1}{m} \text{np.sum}(dZ^{[1]}, \text{axis} = 1, \text{keepdims} = \text{True}) \end{aligned}$$

"It's like the brain"

Andrew Ng

但这个类比还是很粗略的，这是一个 **logistic** 回归单元的 **sigmoid** 激活函数，这里是一个大脑中的神经元，图中这个生物神经元，也是你大脑中的一个细胞，它能接受来自其他神经元的电信号，比如 x_1, x_2, x_3 ，或可能来自于其他神经元 a_1, a_2, a_3 。其中有一个简单的临界计算值，如果这个神经元突然激发了，它会让电脉冲沿着这条长长的轴突，或者说一条导线

传到另一个神经元。

所以这是一个过度简化的对比，把一个神经网络的逻辑单元和右边的生物神经元对比。至今为止其实连神经科学家们都很难解释，究竟一个神经元能做什么。一个小小的神经元其实是极其复杂的，以至于我们无法在神经科学的角度描述清楚，它的一些功能，可能真的是类似 **logistic** 回归的运算，但单个神经元到底在做什么，目前还没有人能够真正解释，大脑中的神经元是怎么学习的，至今这仍是一个谜之过程。到底大脑是用类似于后向传播或是梯度下降的算法，或者人类大脑的学习过程用的是完全不同的原理。

所以虽然深度学习的确是个很好的工具，能学习到各种很灵活很复杂的函数来学到从 x 到 y 的映射。在监督学习中，学到输入到输出的映射，但这种和人类大脑的类比，在这个领域的早期也许值得一提。但现在这种类比已经逐渐过时了，我自己也在尽量少用这样的说法。

这就是神经网络和大脑的关系，我相信在计算机视觉，或其他的学科都曾受人类大脑启发，还有其他深度学习的领域也曾受人类大脑启发。但是个人来讲我用这个人类大脑类比的次数逐渐减少了。

