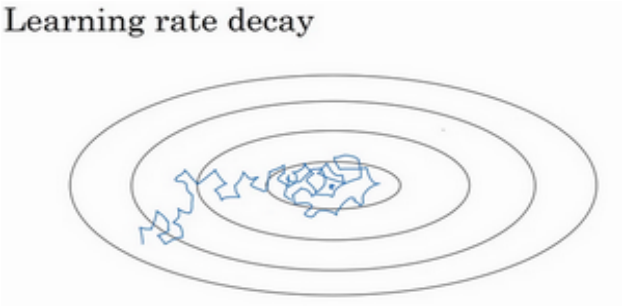
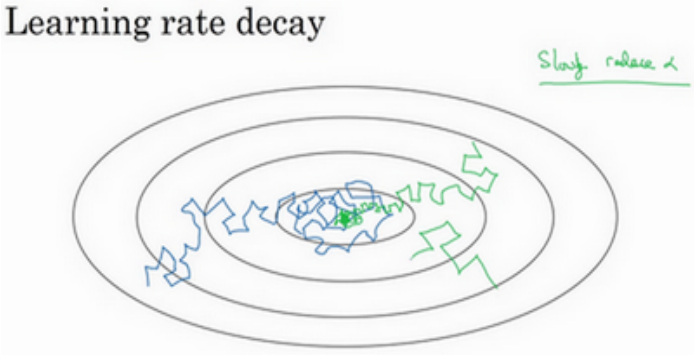
## 2.9 学习率衰减(Learning rate decay)

加快学习算法的一个办法就是随时间慢慢减少学习率，我们将之称为学习率衰减，我们来看看如何做到，首先通过一个例子看看，为什么要计算学习率衰减。



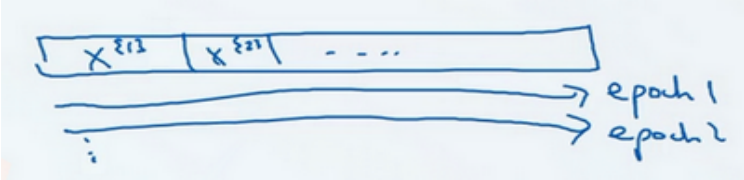

假设你要使用 **mini-batch** 梯度下降法，**mini-batch** 数量不大，大概 64 或者 128 个样本，在迭代过程中会有噪音（蓝色线），下降朝向这里的最小值，但是不会精确地收敛，所以你的算法最后在附近摆动，并不会真正收敛，因为你用的$a$是固定值，不同的 **mini-batch** 中有噪音。



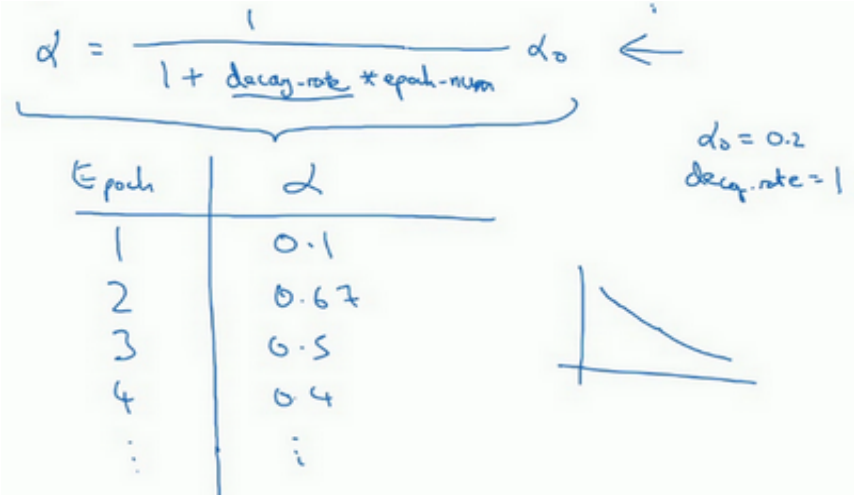

但要慢慢减少学习率$a$的话，在初期的时候，$a$学习率还较大，你的学习还是相对较快，但随着$a$变小，你的步伐也会变慢变小，所以最后你的曲线（绿色线）会在最小值附近的一小块区域里摆动，而不是在训练过程中，大幅度在最小值附近摆动。

所以慢慢减少$a$的本质在于，在学习初期，你能承受较大的步伐，但当开始收敛的时候，小一些的学习率能让你步伐小一些。

你可以这样做到学习率衰减，记得一代要遍历一次数据，如果你有以下这样的训练集:



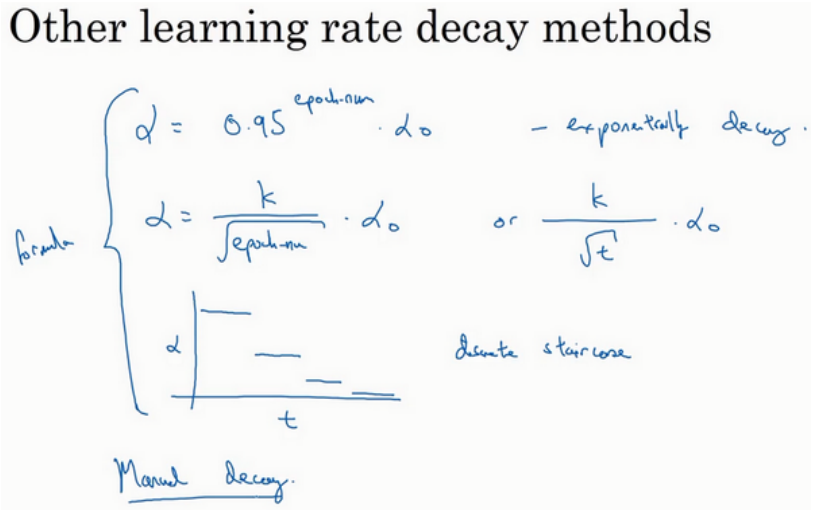
你应该拆分成不同的 **mini-batch**，第一次遍历训练集叫做第一代。第二次就是第二代，依此类推，你可以将$a$学习率设为$a = \frac{1}{1+decayrate*\text{epoch}-\text{num}}a_0$（**decay-rate** 称为衰减率，**epoch-num** 为代数，$a_0$为初始学习率），注意这个衰减率是另一个你需要调整的超参数。




这里有一个具体例子，如果你计算了几代，也就是遍历了几次，如果$a_0$为 0.2，衰减率 **decay-rate** 为 1，那么在第一代中，$a = \frac{1}{1+1}a_0 = 0.1$，这是在代入这个公式计算

$a = \frac{1}{1+decayrate*\text{epoch}-\text{num}}a_0$，

此时衰减率是 1 而代数是 1。在第二代学习率为 0.67，第三代变成 0.5，第四代为 0.4 等等，你可以自己多计算几个数据。要理解，作为代数函数，根据上述公式，你的学习率呈递减趋势。如果你想用学习率衰减，要做的是要去尝试不同的值，包括超参数$a_0$，以及超参数衰退率，找到合适的值，除了这个学习率衰减的公式，人们还会用其它的公式。




比如，这个叫做指数衰减，其中$a$相当于一个小于 1 的值，如$a = 0.95^{\text{epoch}-\text{num}}a_0$，所以你的学习率呈指数下降。

人们用到的其它公式有$a = \frac{k}{\sqrt{\text{epoch}-\text{num}}}a_0$或者$a = \frac{k}{\sqrt{t}}a_0$（$t$为 **mini-batch** 的数字）。

有时人们也会用一个离散下降的学习率，也就是某个步骤有某个学习率，一会之后，学习率减少了一半，一会儿减少一半，一会儿又一半，这就是离散下降（**discrete stair cease**）的意思。

到现在，我们讲了一些公式，看学习率$a$究竟如何随时间变化。人们有时候还会做一件事，手动衰减。如果你一次只训练一个模型，如果你要花上数小时或数天来训练，有些人的确会这么做，看看自己的模型训练，耗上数日，然后他们觉得，学习速率变慢了，我把$a$调小一点。手动控制$a$当然有用，时复一时，日复一日地手动调整$a$，只有模型数量小的时候有用，但有时候人们也会这么做。

所以现在你有了多个选择来控制学习率$a$。你可能会想，好多超参数，究竟我应该做哪一个选择，我觉得，现在担心为时过早。下一周，我们会讲到，如何系统选择超参数。对我而言，学习率衰减并不是我尝试的要点，设定一个固定的$a$，然后好好调整，会有很大的影响，学习率衰减的确大有裨益，有时候可以加快训练，但它并不是我会率先尝试的内容，但下周我们将涉及超参数调整，你能学到更多系统的办法来管理所有的超参数，以及如何高效搜索超参数。

这就是学习率衰减，最后我还要讲讲神经网络中的局部最优以及鞍点，所以能更好理解在训练神经网络过程中，你的算法正在解决的优化问题，下个视频我们就好好聊聊这些问题。