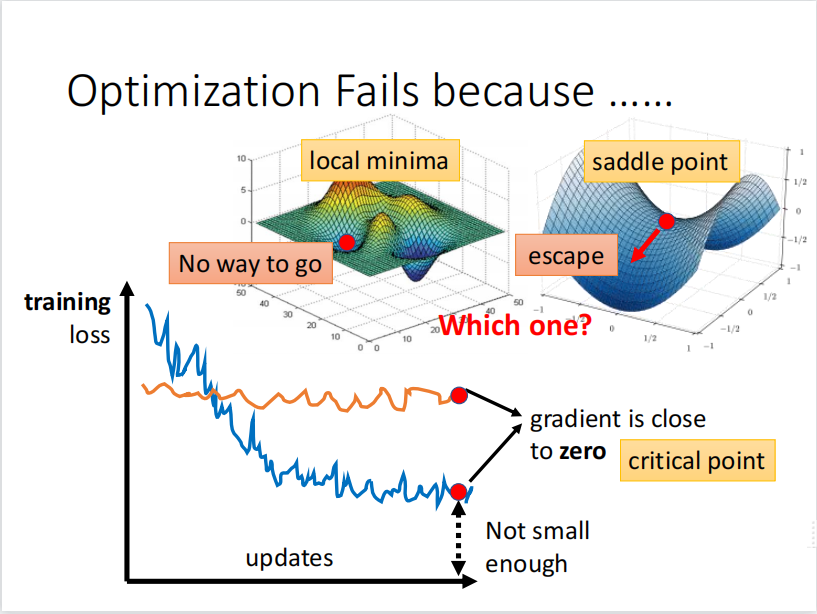
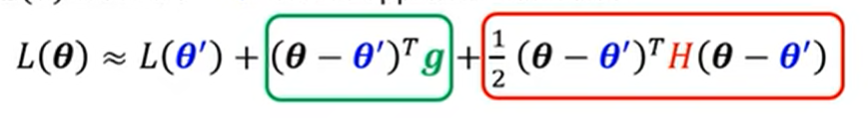
**学习笔记**

**梯度为零问题：**在某些情况下(critical point)，损失函数的梯度趋近于0，使得网络结构很难训练。

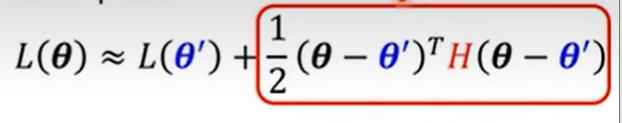
举个例子，如下图所示，当位于local minima和saddle point时，失函数的梯度趋近于0，但在saddle point时还有路径可以使LOSS更小。



定义LOSS：



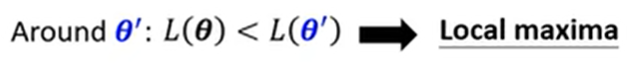
在critical point附近时：



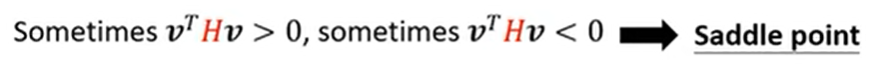
红框中为Hessian，大于0：得：



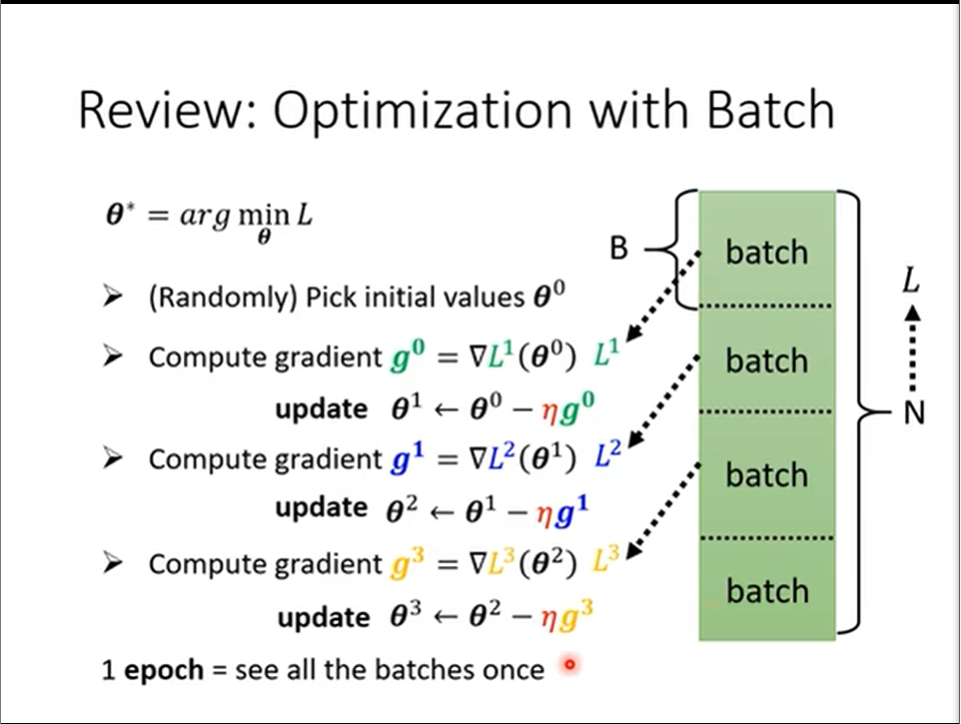
小于0，得：



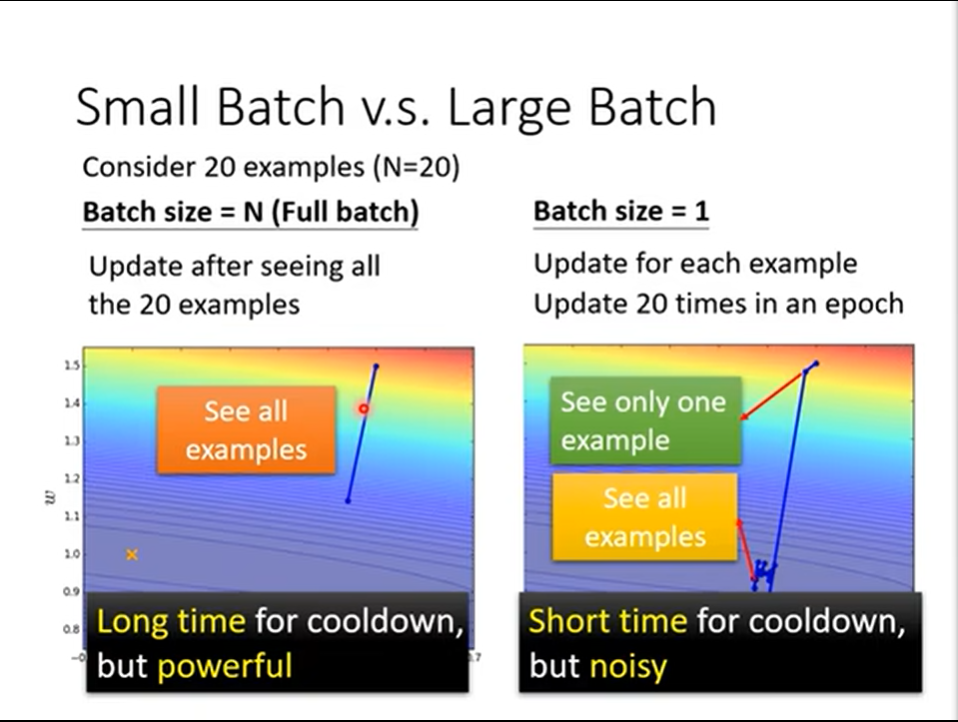
可大可小得：



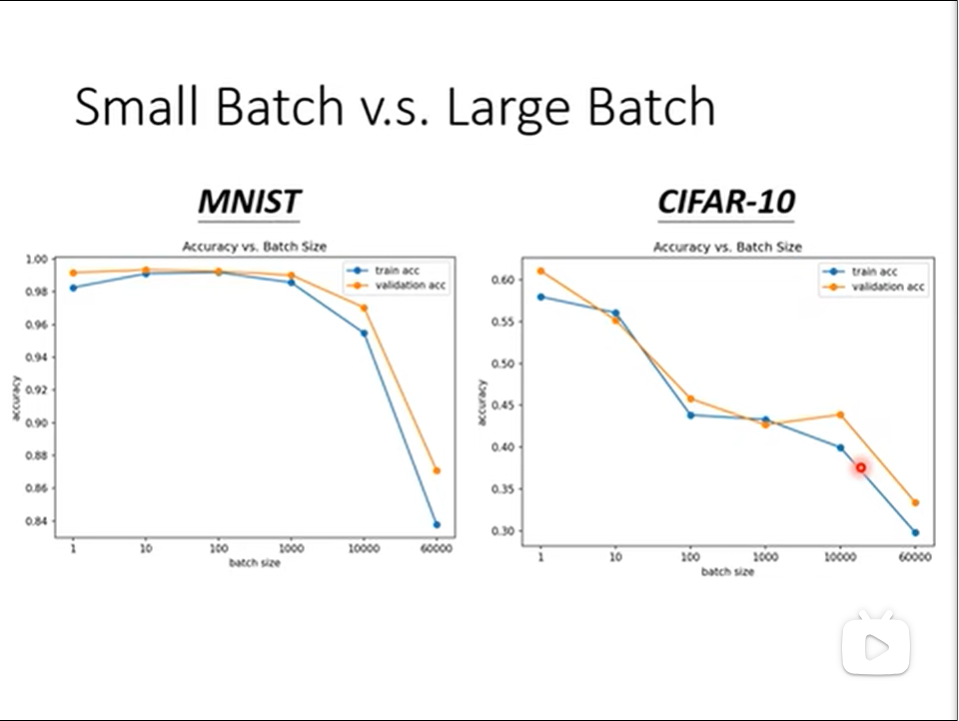
**划分为batch：**

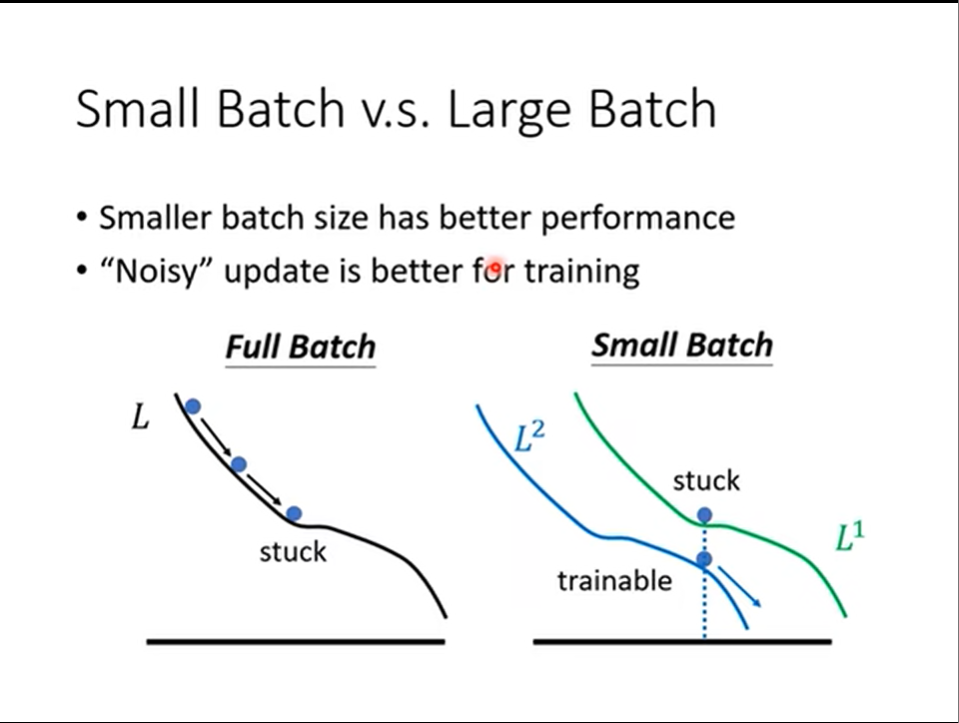


每进行一次成为一次epoch，其中Batch可小可大，batch小则N大，以下使用N=20来对比batch小和batch大两种情况，其中N等于20代表有20个examples，则训练时必须将20个examples都训练后才能更新一次参数。



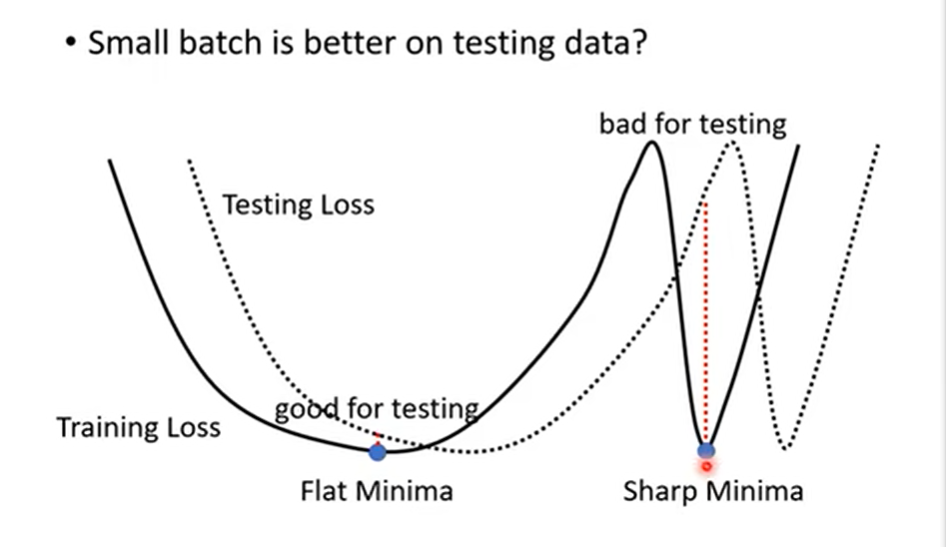
考虑利用GPU平行运算的情况下，在时间运算上基本相等。但batch size越大，实际的acc反而会降低。

 但这不是overfitting，而是Optimization的问题。



但small batch在测试数据上可能表现更好，这就是overfitting。

Flat Minima和Sharp Minima：

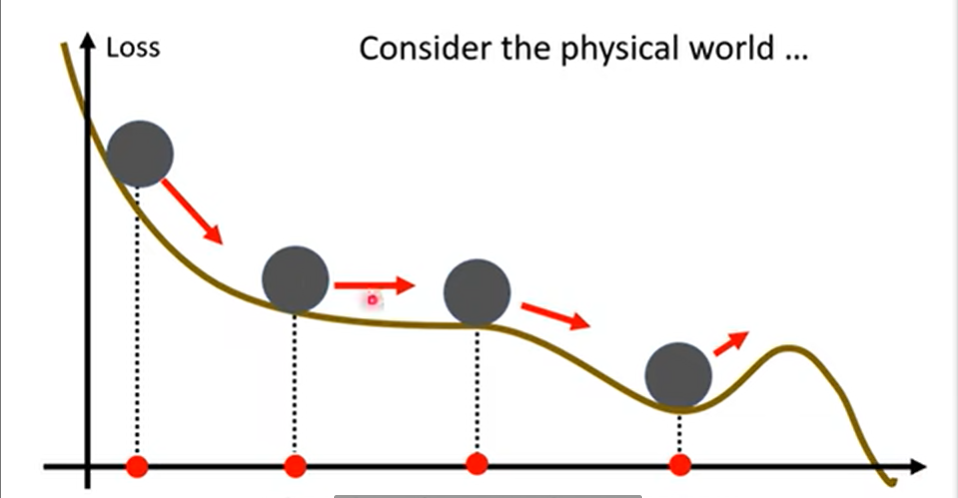


当在training loss和testing loss有上图所示偏差时，Flat Minima的变化不大，但当sharp Minima会变化很大。

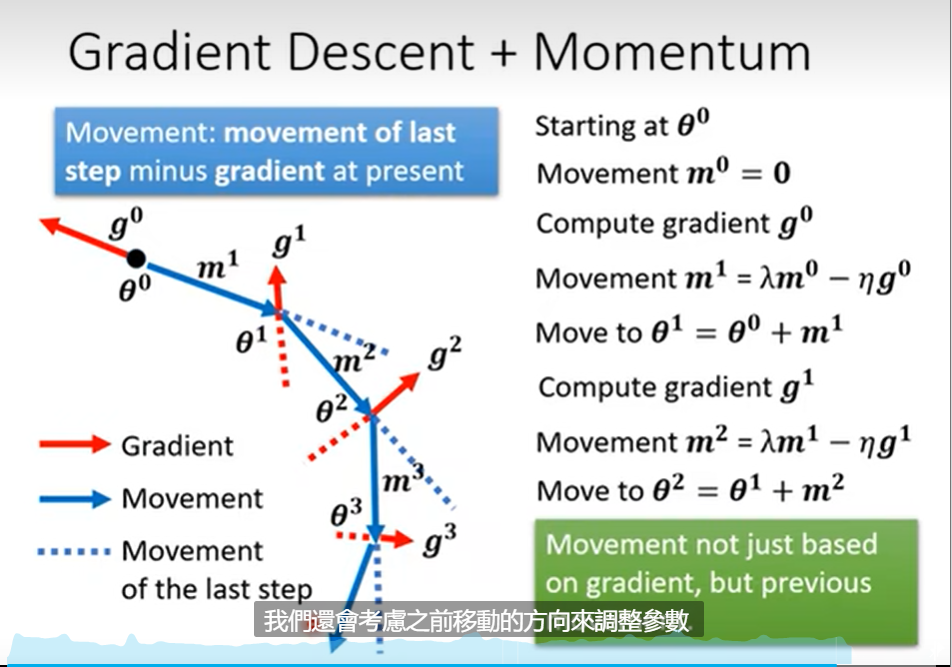
下图是small batch对比large batch：



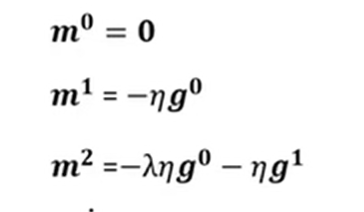
Momentum技术概念,通过动量越过local minima和saddle point：



以下将Momentum和gradient descent结合：



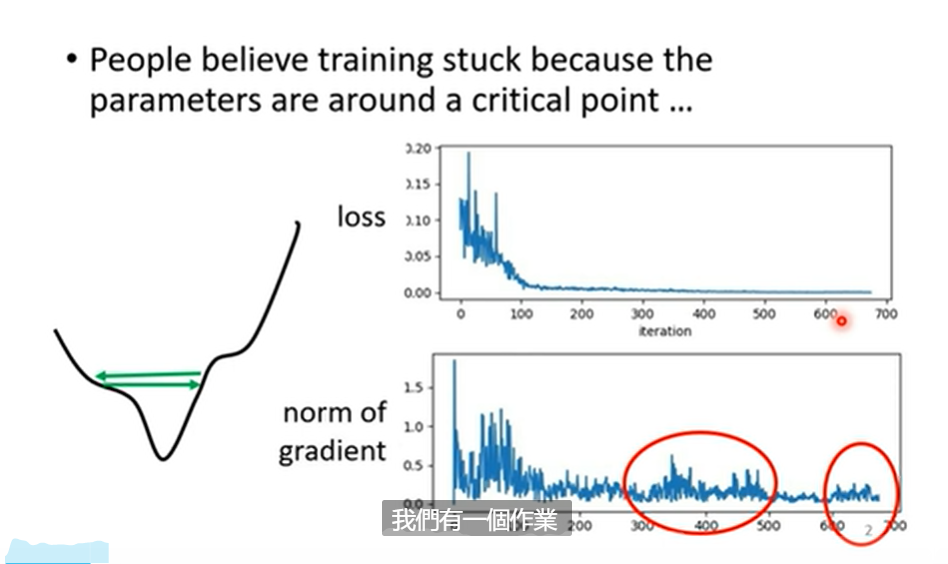
即根据Momentum和gradient descent来确定方向。化简得：



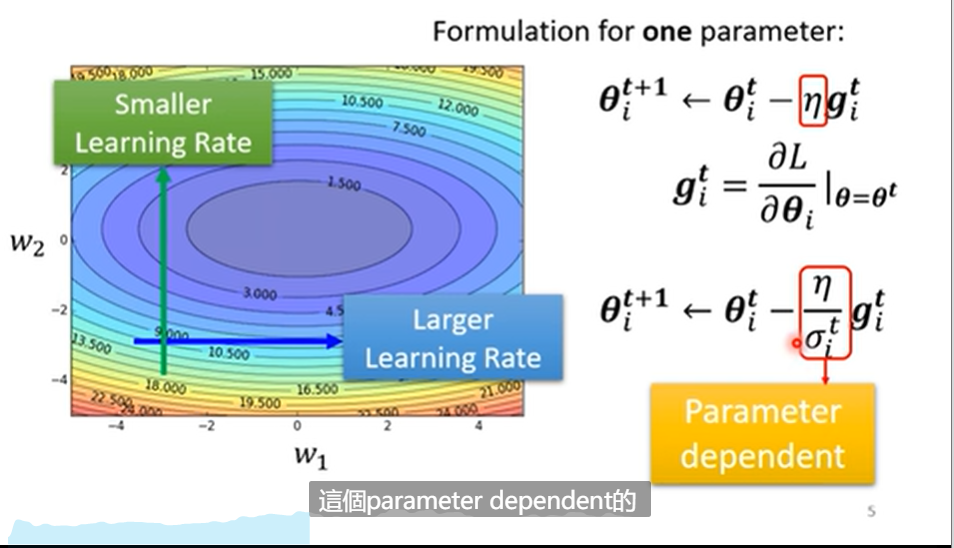
可得实际上是过去gradient descent的总和。

**Adaptive learning Rate技术：**

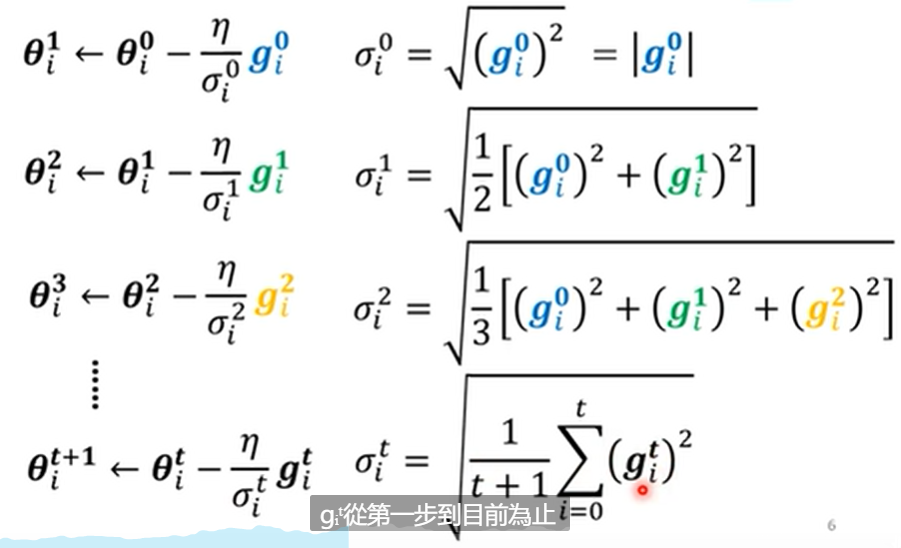
Loss不会再下降时Gradient不一定为零，即不一定卡到local minima或者saddle oint。



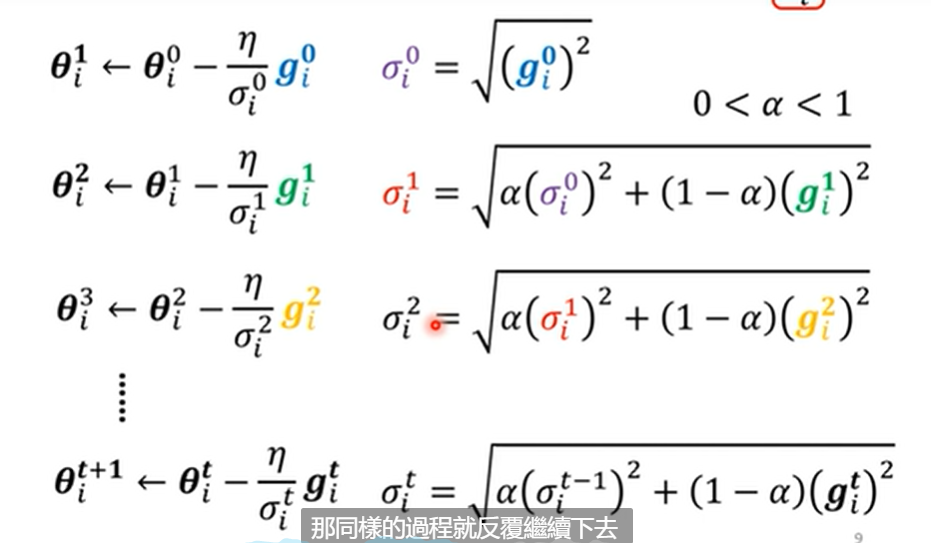
我们期待的是当在某一方向的坡度非常陡时，learning rate可以小一点，当某一方向的learning rate非常大时，我们的learning rate可以大一点，如下一个参数的示例。



而在图中的新的learning rate计算方法如下。

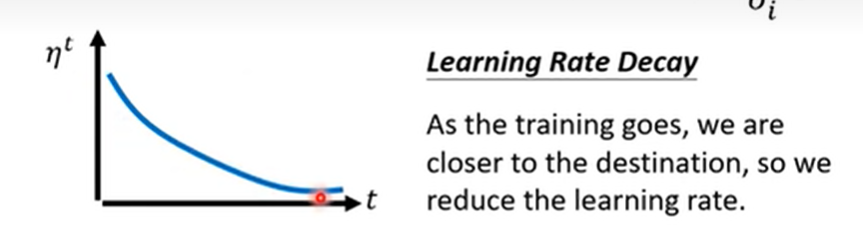


RMSProp方法改良上述后：

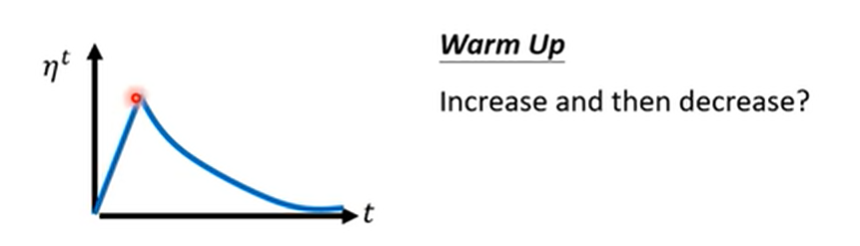


Learning Rate Scheduling：

随着时间的推进，Learning Rate要慢慢下降.(Learning Rate Decay)

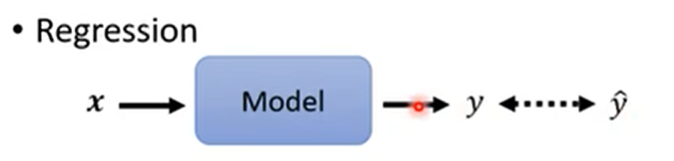


Warm up:

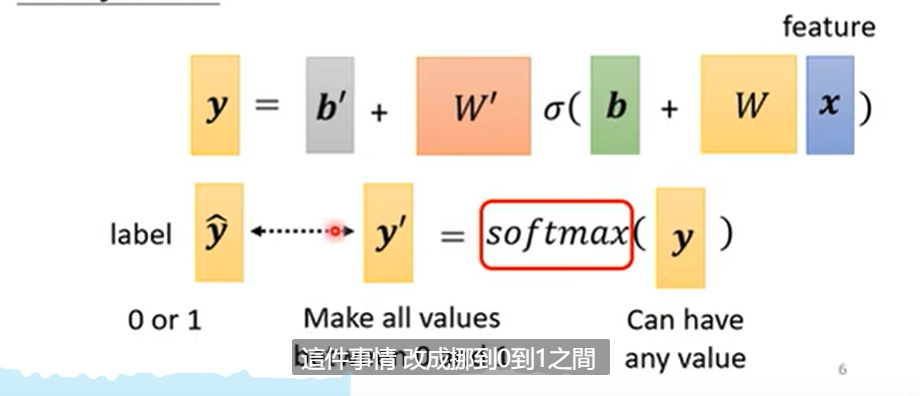


**分类：**

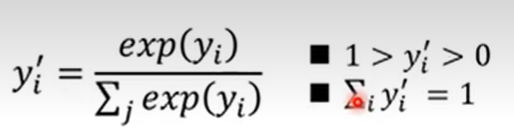
**Regression:**



Classification：



其中siftmax是将所有值置于0到1之间。具体的计算公式如下：



Mean Square Error 和 Cross-entropy计算

