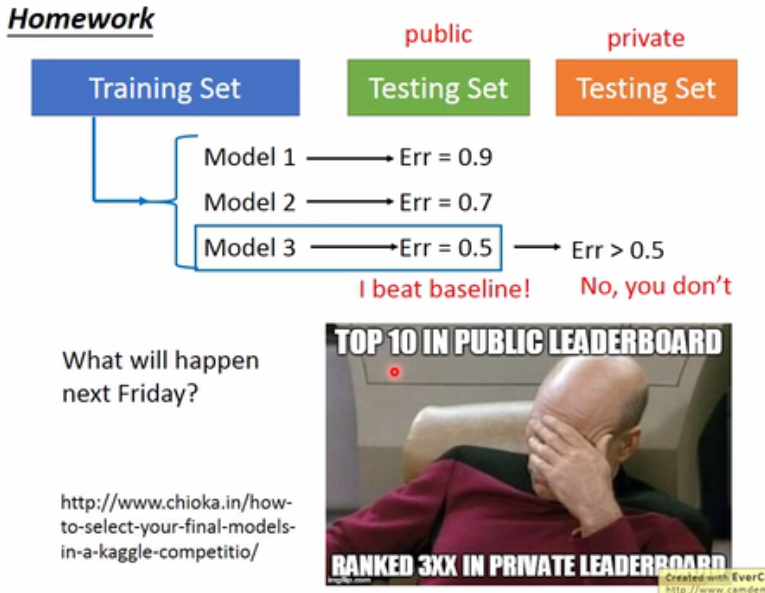
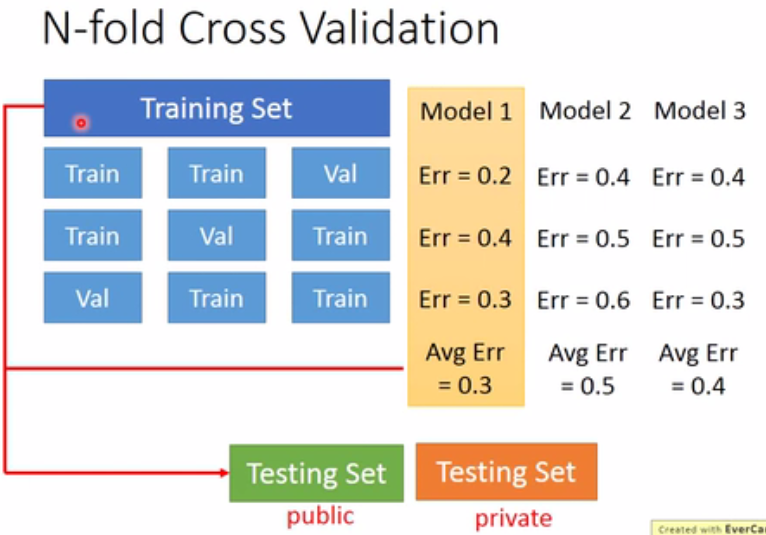
**台大李宏毅—机器学习2017**



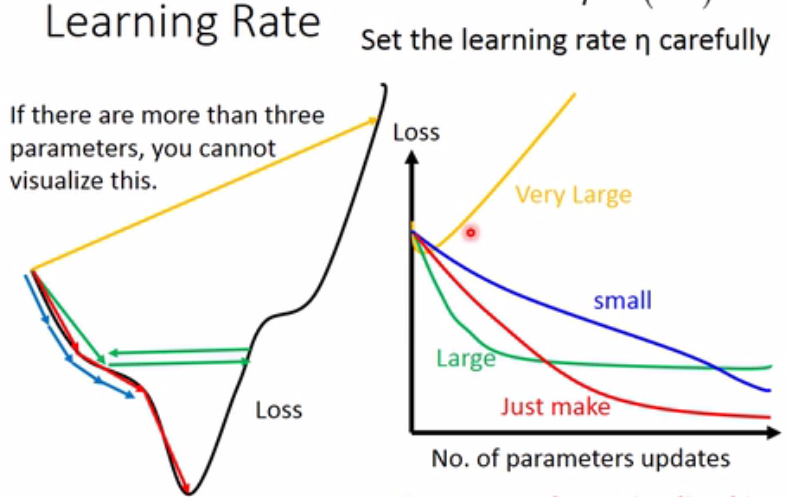
**模型选择的时候，如上，不能直接在training set上训练出模型，然后分别在testing set上看结果选择，这样做的话，例如kaggle上的public排名高，不能保证private一定高，那么应该怎么做呢？---K折交叉验证哈**



**Training set进一步分为训练集和验证集，分别训练然后在验证集上查看结果，最后选择平均错误率最小的model**

# 关于梯度下降

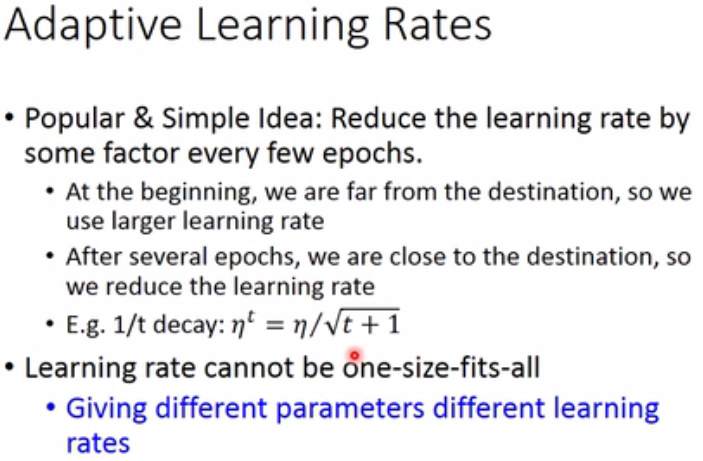
## Learning rate



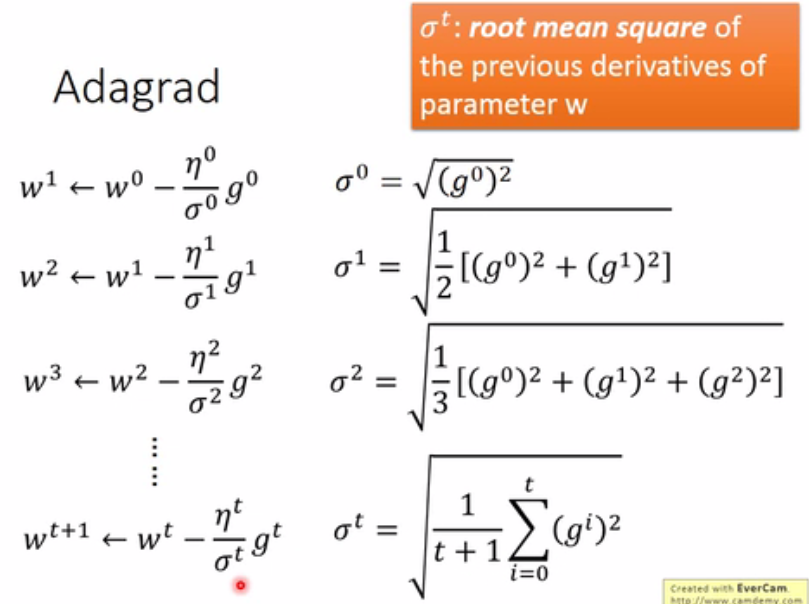
Learning rate对于梯度下降非常重要，如上图是不同的learning rate对于loss function的影响，如果learning rate设置太小，那么loss下降的会很慢；如果learning rate设置的很大，那么会出现先下降的很快，过一会便不怎么变化的情况，如绿线；如果设置很大很大，会出现loss直接变大的情况，如黄线；正确的情况是图中红线哈

当然，是有梯度自适应算法的；

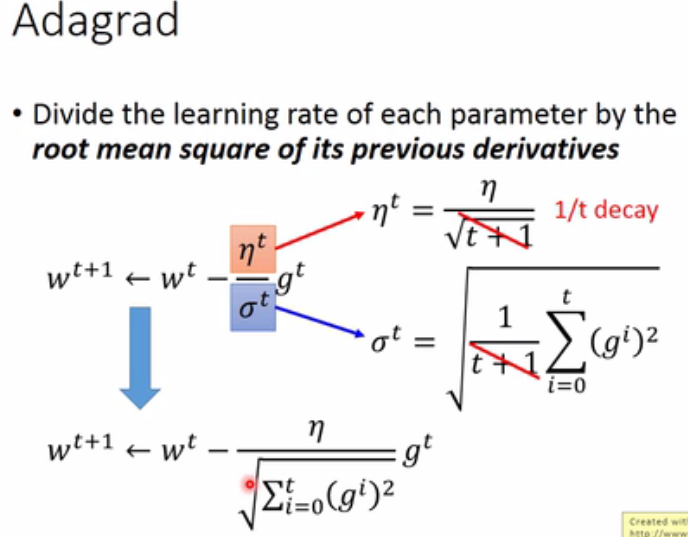
**基本思想：随着参数的不断更新，learning rate应该不断减小，并且应该不同的参数有不同的learning rate 原理如下很清楚：**

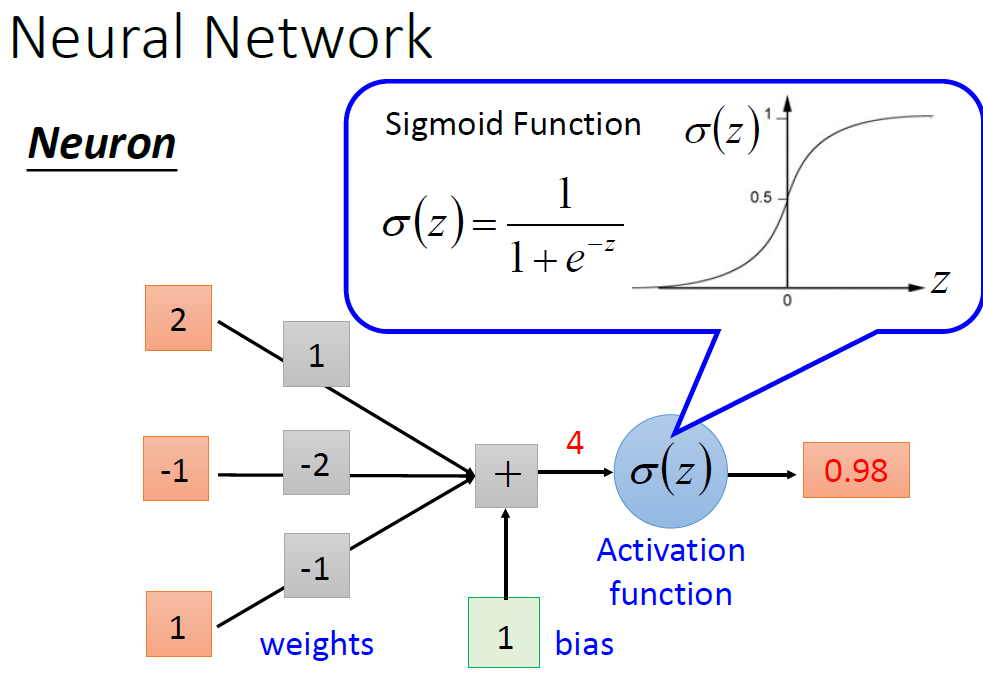


**下面介绍比较著名的自适应learning rate算法：**

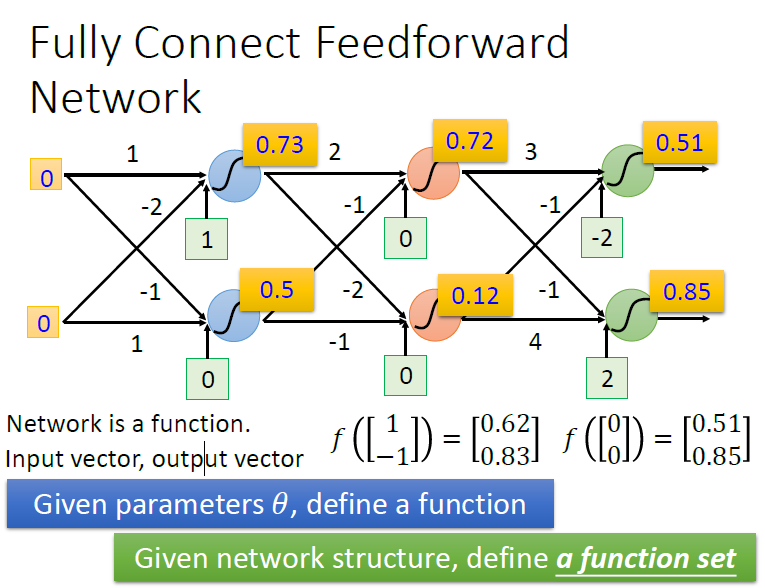


**上图是Adagrad基本思路ηt等于η除以根号t+1**

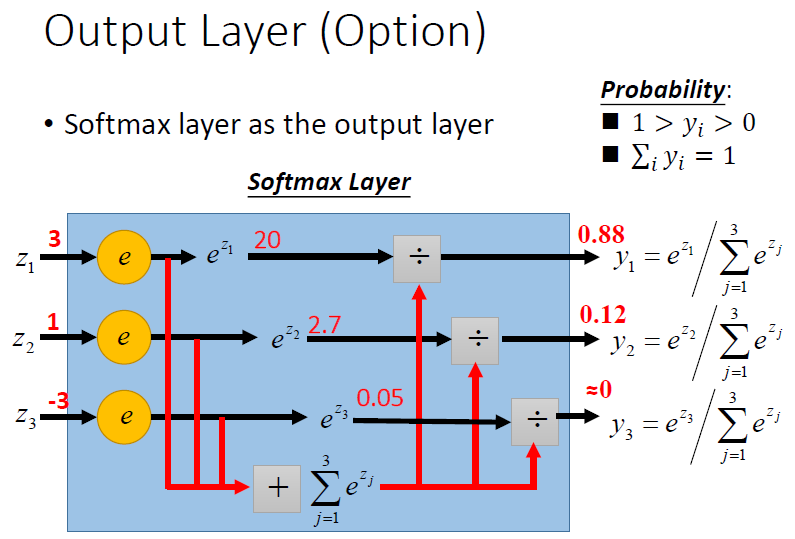




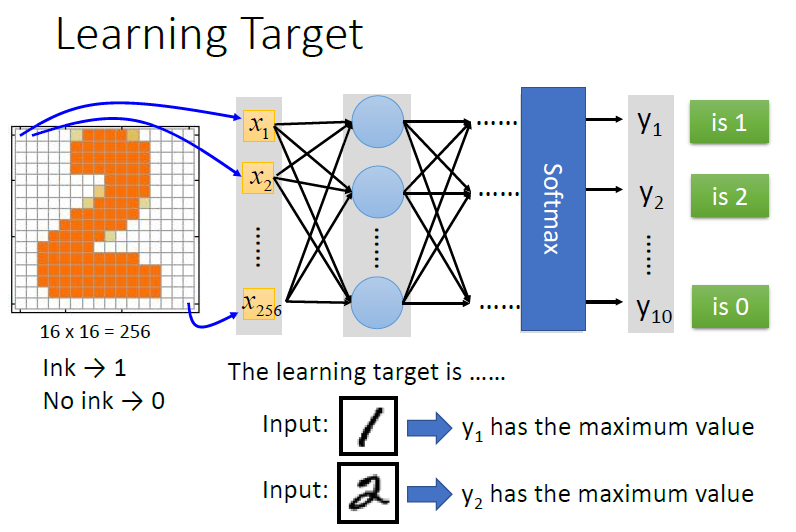
每个神经元包括自己的权重weights(w)和偏执项b(bias)，还有激活函数。



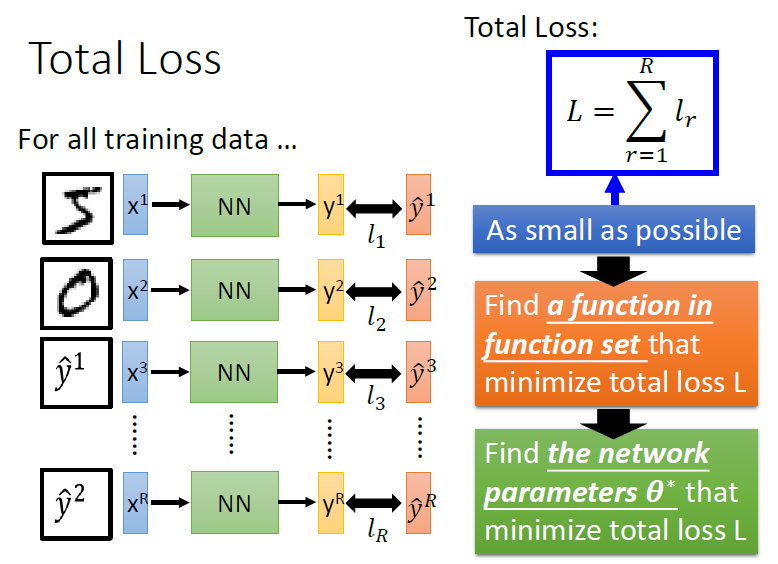
**神经网络整体看就是一个映射啦，输入一个vector，输出一个vector。**



输出层一般是softmax layer，即会先结果去e次方，然后进行归一化哈

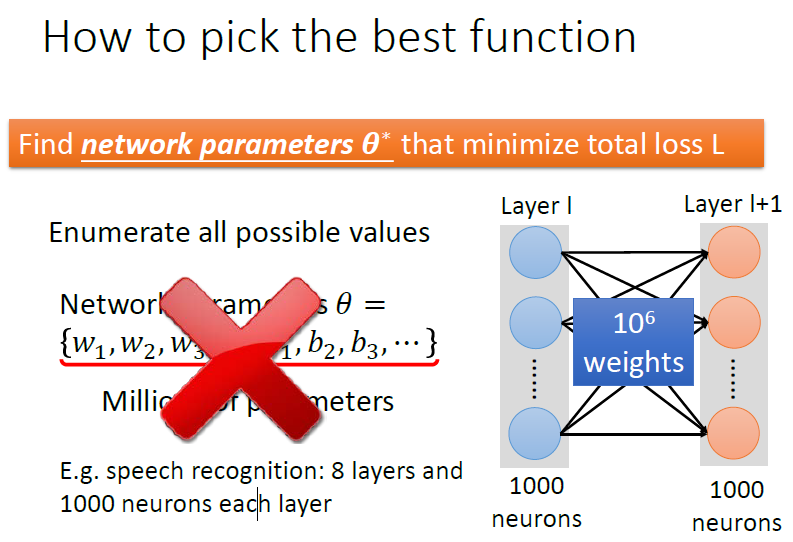


以这个数字图像识别为例，输出即是10维向量，每个维度是对应数字的概率

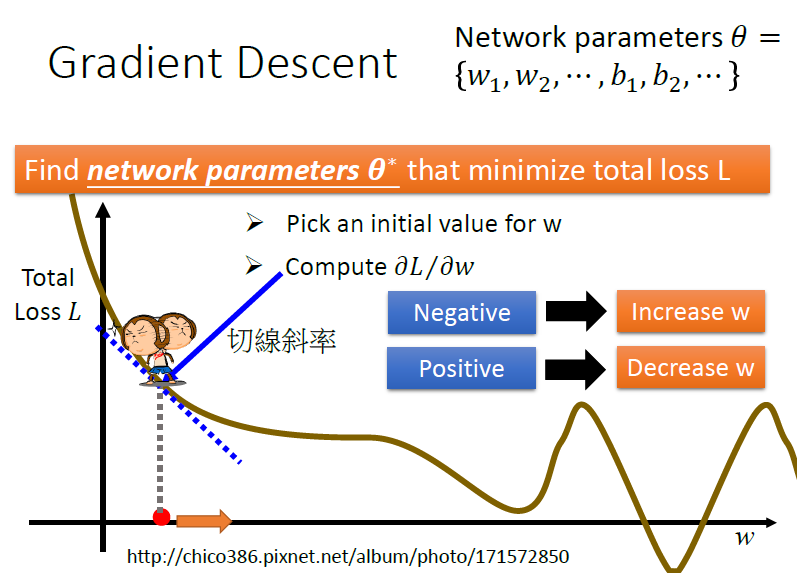


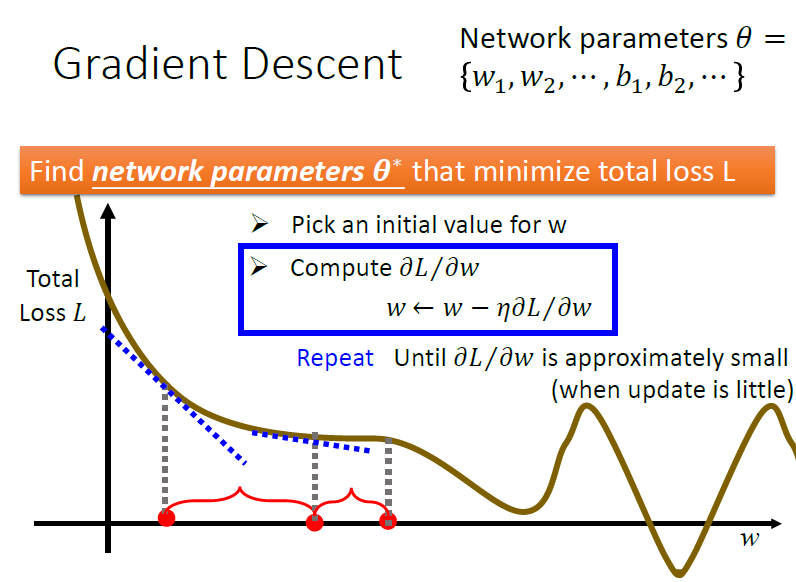
我们目标就是在给定训练集上学出一个total loss最小的神经网络啦

下面就是怎么学的问题，如何学习出神经网络中的各个参数呢（w,b等），一种直观想法是枚举所有可能参数，然后一个一个试试，看谁最小，但是参数量太大，不现实呀。

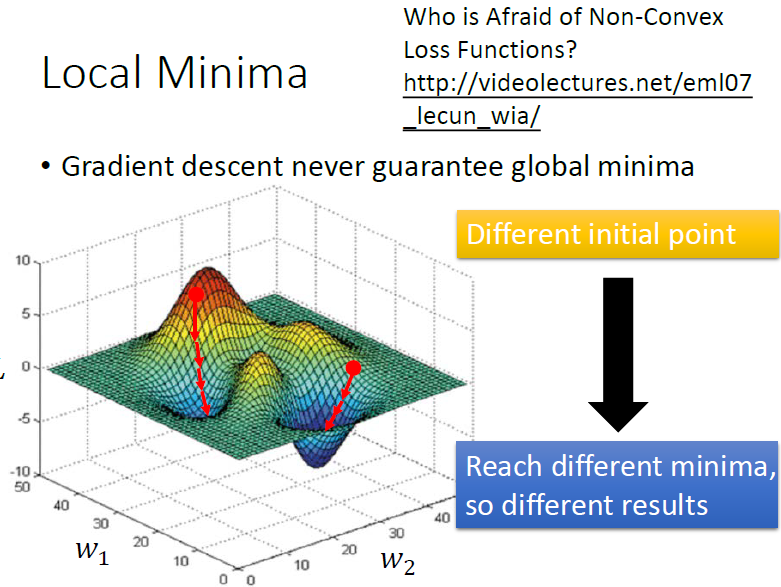


**那么如何学习参数呢？咳咳，类似于机器学习，采用梯度下降呀：）（神经网络因为激活函数的存在，基本都是非凸函数，所以无法直接求得全局最小值，直接通过梯度下降，求一个能用的局部最小呀）**





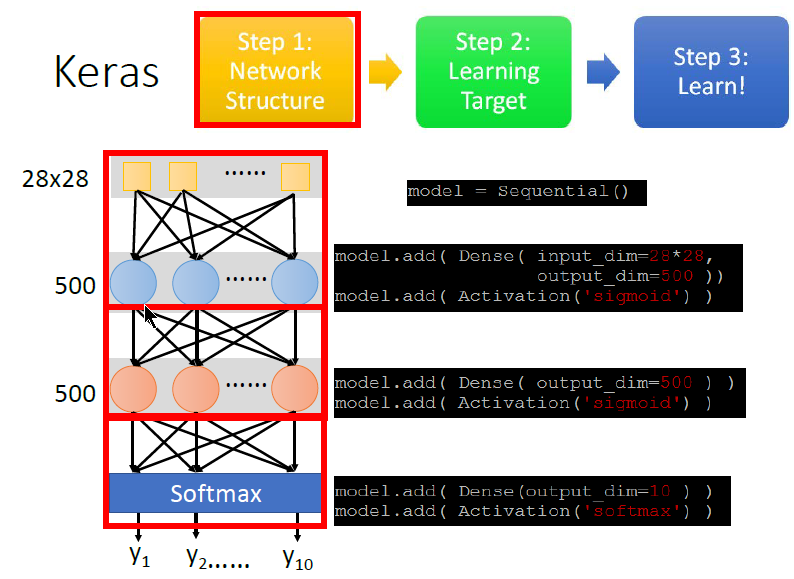
一直迭代，知道梯度值是非常小了，原理与机器学习类似的。



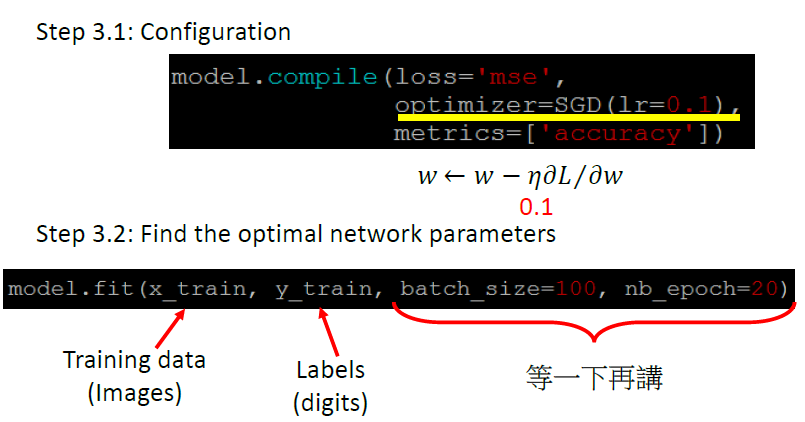
如之前所说，不保证全局最小，因为受到初始值的设定，不同的初始值可能到达不同的局部最小值，所以说结果不同。

Keras简单使用：

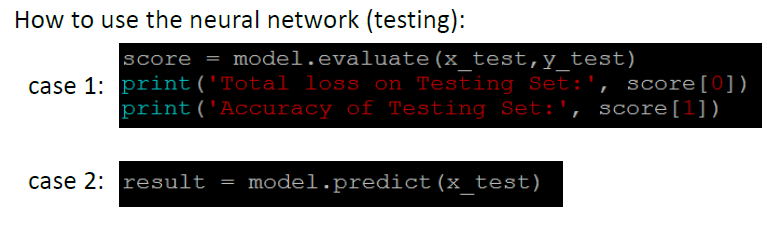
1.搭建网络（“搭积木啦”）



2.配置目标函数和学习率，以及fit数据啦：）



最后就可以predict和看效果啦：



Mini-batch: