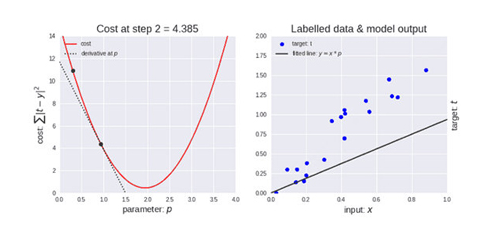
只有在数据很庞大的时候(在机器学习中，几乎任何时候都是)，我们才需要使用 epoch，batch size，迭代这些术语，在这种情况下，一次性将数据输入计算机是不可能的。因此，为了解决这个问题，我们需要把数据分成小块，一块一块的传递给计算机，在每一步的末端更新神经网络的权重，拟合给定的数据。

算法是迭代的，意思是需要多次使用算法获取结果，以得到最优化结果。梯度下降的迭代性质能使欠拟合的图示演化以获得对数据的最佳拟合。



梯度下降中有一个称为学习率的参量。如上图左所示，刚开始学习率更大，因此下降步长更大。随着点下降，学习率变得越来越小，从而下降步长也变小。同时，代价函数也在减小，或者说代价在减小，有时候也称为损失函数或者损失，两者都是一样的。(损失/代价的减小是一件好事)。当一个完整的数据集通过了神经网络一次并且返回了一次，这个过程称为一个 epoch。

batch\_size：中文翻译为批大小（批尺寸）。在深度学习中，一般采用SGD训练，即每次训练在训练集中取batch\_size个样本训练。

iteration：中文翻译为迭代，1个iteration等于使用batch\_size个样本训练一次；一个迭代 = 一个正向通过+一个反向通过。

epoch：迭代次数，1个epoch等于使用训练集中的全部样本训练一次；一个epoch = 所有训练样本的一个正向传递和一个反向传递。

举个例子，训练集有1000个样本，batch\_size=10，那么：训练完整个样本集需要：100次iteration，1次epoch。

**BATCH\_SIZE**

每批数据量的大小。DL通常用SGD的优化算法进行训练，也就是一次（1个iteration）一起训练batch\_size个样本，计算它们的平均损失函数值，来更新参数。

**ITERATION**

1个iteration即迭代一次，也就是用batch\_size个样本训练一次。

**EPOCH**

然而，当一个 epoch 对于计算机而言太庞大的时候，就需要把它分成多个小块。为什么要使用多于一个 epoch?我知道这刚开始听起来会很奇怪，在神经网络中传递完整的数据集一次是不够的，而且我们需要将完整的数据集在同样的神经网络中传递多次。但是请记住，我们使用的是有限的数据集，并且我们使用一个迭代过程即梯度下降，优化学习过程和图示。因此仅仅更新权重一次或者说使用一个epoch是不够的。

随着 epoch 数量增加，神经网络中的权重的更新次数也增加，曲线从欠拟合变得过拟合。1个epoch指用训练集中的全部样本训练一次，此时相当于batch\_size 等于训练集的样本数。

最初训练DNN采用一次对全体训练集中的样本进行训练（即使用1个epoch），并计算一次损失函数值，来更新一次权值。当时数据集较小，该方法尚可。后来随着数据集迅速增大，导致这种方法一次开销大进而占用内存过大，速度过慢。

后来产生了一次只训练一个样本的方法（batch\_size=1）,称作在线学习。该方法根据每一个样本的情况更新一次权值，开销小速度快，但收到单个样本的巨大随机性，全局来看优化性能较差，收敛速度很慢，产生局部震荡，有限迭代次数内很可能无法收敛。

目前常用随机梯度下降SGD来训练，相当于上述两个“极端”方法的折中：将训练集分成多个mini\_batch（即常说的batch）,一次迭代训练一个minibatch（即batch\_size个样本），根据该batch数据的loss更新权值。这相比于全数据集训练，相当于是在寻找最优时人为增加了一些随机噪声，来修正由局部数据得到的梯度，尽量避免因batch\_size过大陷入局部最优。

这种方法存在两对矛盾。由于一次只分析的一小部分数据，因此整体优化效果与batch\_size有关：

batch\_size越小，一个batch中的随机性越大，越不易收敛。然而batch\_size越小，速度越快，权值更新越频繁；且具有随机性，对于非凸损失函数来讲，更便于寻找全局最优。从这个角度看，收敛更快，更容易达到全局最优。

batch\_size越大，越能够表征全体数据的特征，其确定的梯度下降方向越准确，（因此收敛越快），且迭代次数少，总体速度更快。然而大的batch\_size相对来讲缺乏随机性，容易使梯度始终向单一方向下降，陷入局部最优；而且当batch\_size增大到一定程度，再增大batch\_size，一次batch产生的权值更新（即梯度下降方向）基本不变。因此理论上存在一个最合适的batch\_size值，使得训练能够收敛最快或者收敛效果最好（全局最优点）。

根据现有的调参经验，加入正则化项BN后，在内存容量允许的情况下，一般来说设置一个较大的batch\_size值更好，通常从128开始调整。