MiniBatch（完整数据学习的问题，是什么，为什么，怎么使用）：

动态梯度下降：

误差曲面，含义（横坐标，纵坐标）：如果是方差代价函数，形状就是碗状的东西。如果是多层的，不是线性的，就不是一个碗状的东西，局部可以近视看成碗状的。

完整数据训练的收敛速度，最陡的并不指向最低的。想要的方向移动小不想要的方向移动大。

学习率过大，导致穿过峡谷，所以应该是斜度小而连续的应该移动快一点，斜度大而不连续的应该移动慢一点。

Mini batch 的好处是，计算更少，同时计算多个训练样例。注意，minibatch 的取样要均衡并且有代表性。

验证数据调整学习（可以知道什么时候开始降低学习率，和什么时候停止训练），训练数据学习，测试数据测试效果。

MiniBatch的优化方法：

* 权值初始化

用小的随机值来初始化权值，而不是使用全0。

* 输入平移，误差曲面是一个圆，会让梯度下降变得简单，
* 伸缩输入，
* 加快学习速度：动量方法，独立的自适应学习率，

动量方法：

斜面会到达一个恒定值？

自适应的学习率（适用于使用full-data 学习）改变学习率：

为什么需要自适应：

不同层的梯度不一样，

有一些神经元并没有获得足够的输入，但也要跟着一些其他的神经元一样改变权值。

怎么自适应：

两次相邻的梯度符号不变的时候，g + 一个小的值 0.05

符号改变的时候，g \* 小于1 且较大的值，0.95

G 是根据权值来变化的。

加上动量的方法，是 累积的梯度于当前的梯度进行 符号的比较。

Rprop改变步长，不管梯度怎么样（为什么要用：因为这样就不用）：

每个weight的Step size 动态改变，只看斜率的符号，然后应用一下的公式：

\*1.2 来增加step size ，通过 \*0.5 （使步长下降的更快）

不适用在minibatch 的原因是：多个+0.1 和 一个负的0.9 我们的期望是权值应该大致不变，但rprop 却大幅增加了权值。

Rmsprop （改进）：

每一个权值保存一个RMS root mean square，这样相邻的mini batch 之间就可以除以一个大致相同的值。

没有固定的准则：

神经网络大不相同

神经网络所要完成的任务不同。