

检查梯度下降是否收敛：

批量梯度下降：先画出迭代次数和代价函数的图像，再进行梯度下降，根据图像判断。

随机梯度下降：先计算代价函数，再更新参数 θ 。每1000此迭代，画出上1000个算法处理的图像，判断是否收敛。

Checking for convergence

→ Batch gradient descent:

→ Plot $J_{train}(\theta)$ as a function of the number of iterations of gradient descent.

$$J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad M = 300,000,000$$

→ Stochastic gradient descent:

$$cost(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad \rightarrow (x^{(i)}, y^{(i)}), (x^{(i+1)}, y^{(i+1)}), \dots$$

→ During learning, compute $cost(\theta, (x^{(i)}, y^{(i)}))$ before updating θ using $(x^{(i)}, y^{(i)})$.

→ Every 1000 iterations (say), plot $cost(\theta, (x^{(i)}, y^{(i)}))$ averaged over the last 1000 examples processed by algorithm.

当减小学习率，函数噪声会更加大。如图一。

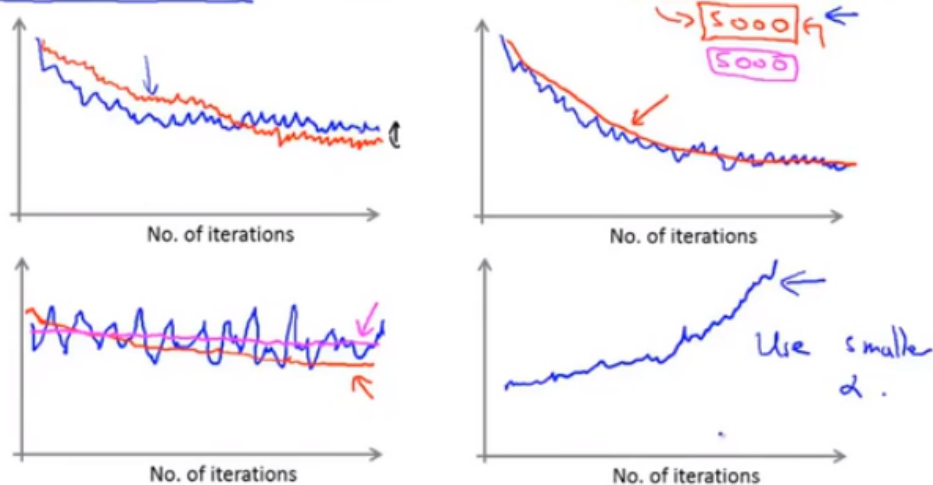
如果函数图像的噪声太大，可以加大每次迭代使用的数据数量，但这样可能造成反馈有延迟。如图二。

图像过于震荡，如图三，但他们的平均值总体是在下降的，那么可以增大训练样本。

当图像并不是在下降趋势，说明算法发散，如图四，可以使用更小的学习率。

Checking for convergence

Plot $cost(\theta, (x^{(i)}, y^{(i)}))$, averaged over the last 1000 (say) examples



Andrew Ng

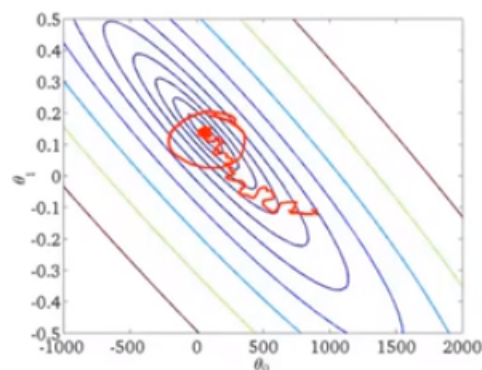
我们也可以调整学习速率，让学习速率随着迭代次数逐渐减小，设置const1和const2，根据公式设置学习速率。

Stochastic gradient descent

$$cost(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2}(h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^m cost(\theta, (x^{(i)}, y^{(i)}))$$

1. Randomly shuffle dataset.
2. Repeat {
 - for $i := 1, \dots, m$ {
 - $\theta_j := \theta_j - \alpha(h_{\theta}(x^{(i)}) - y^{(i)})x_j^{(i)}$ (for $j = 0, \dots, n$)



Learning rate α is typically held constant. Can slowly decrease α over time if we want θ to converge. (E.g. $\alpha = \frac{const1}{iterationNumber + const2}$) $\alpha \rightarrow 0$

Andrew Ng