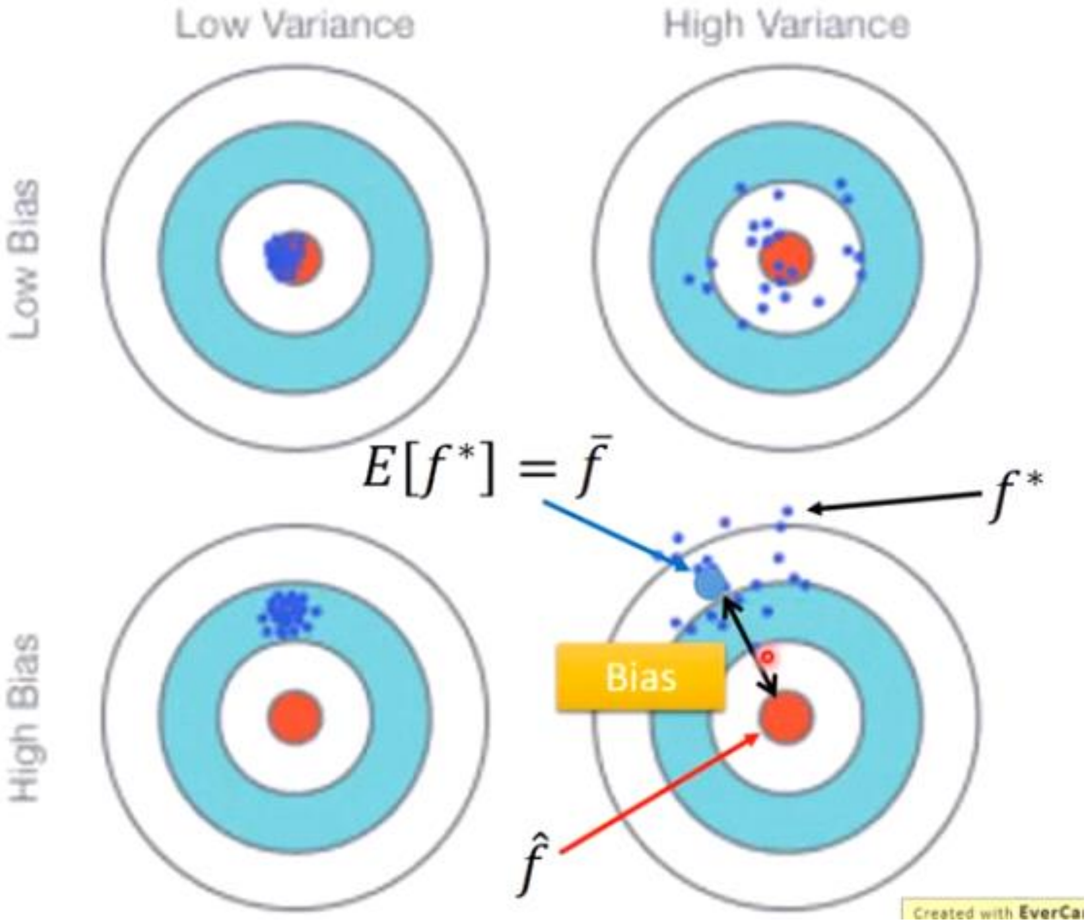
Task03

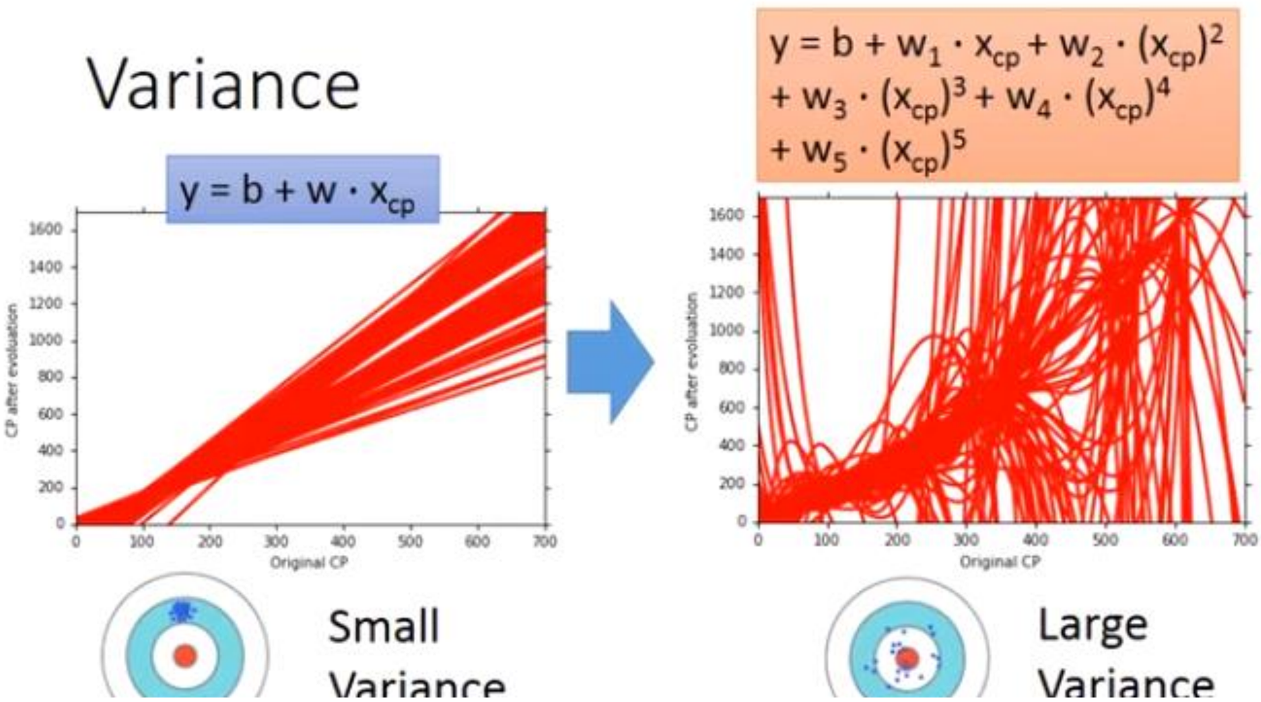
2021年8月20日 20:35

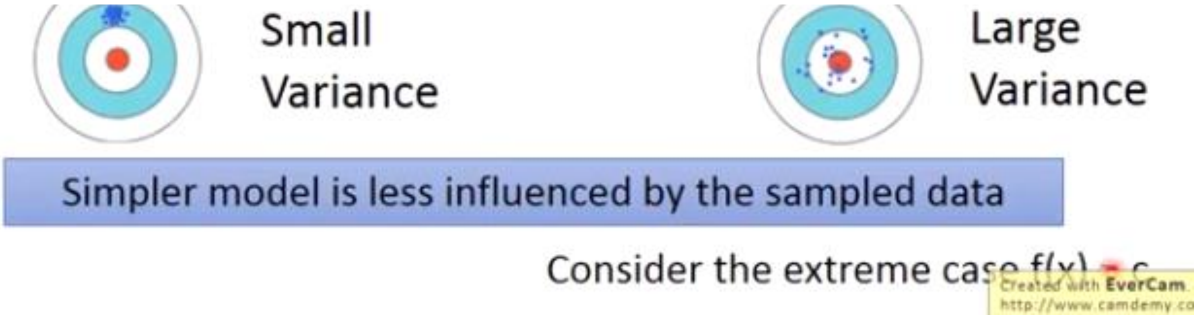
P5 误差从哪来?

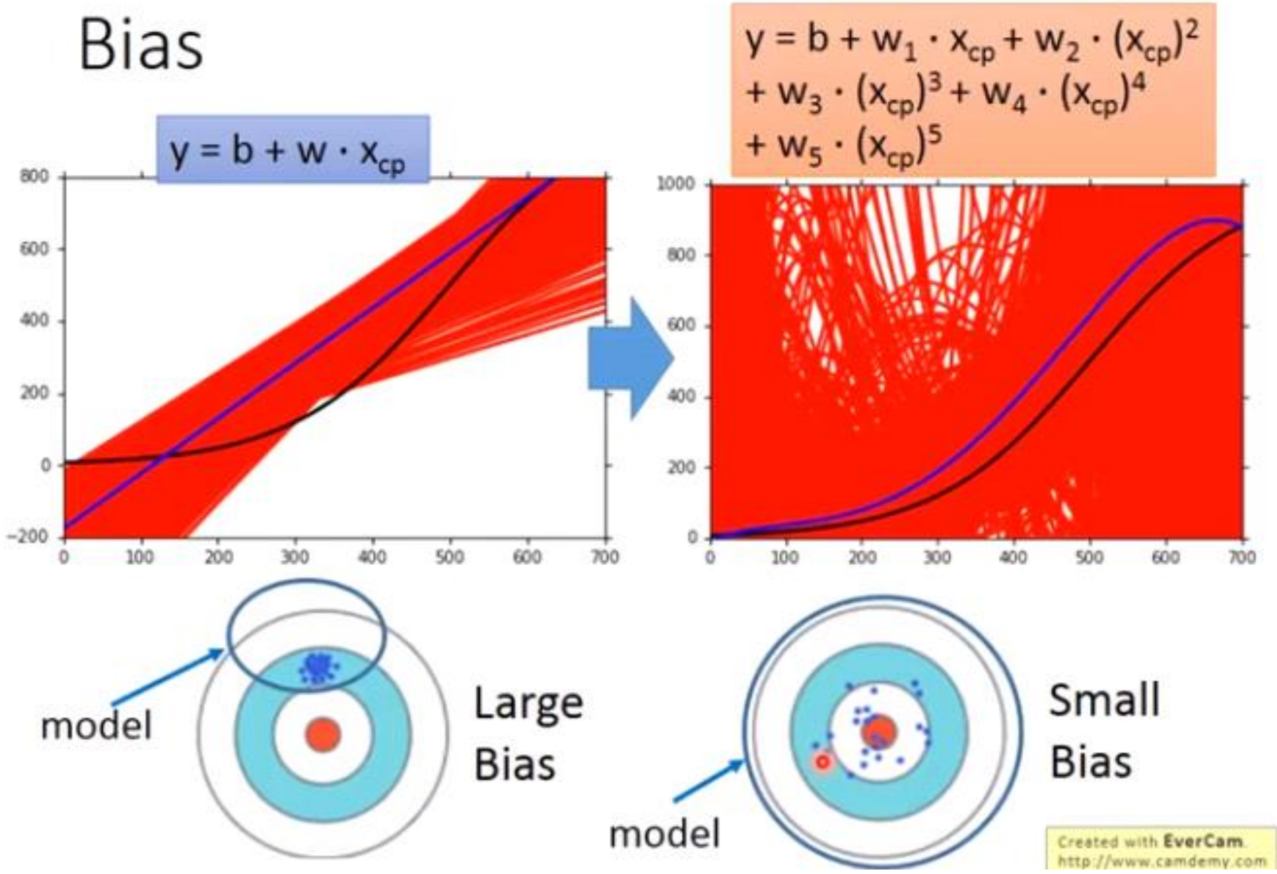
- 误差来源 bias + variance（图中f*表示计算得出的function，f^是实际最好的function



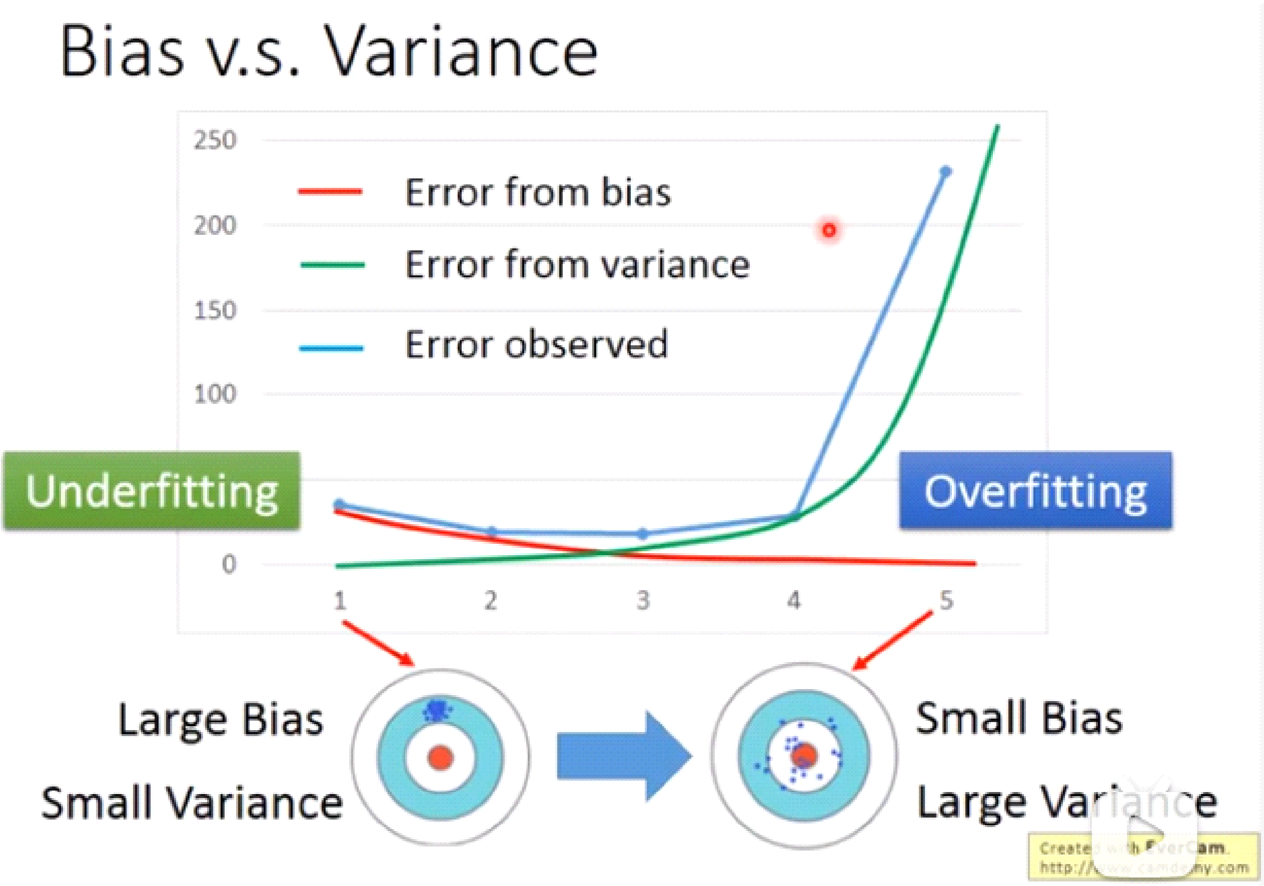
- 简单的model的variance更小，受样本数据的影响较小，反之，如果model复杂，那variance大，function受样本影响较大。







- 简单model的bias较大，复杂model的bias小
- 如果误差主要来源于variance，那属于overfitting即过拟合，如果误差主要来自bias，那属于underfitting。



- 如果bias更大，那需要redesign model，增加变量，使用更复杂的model；如果variance更大，那需要使用更多的数据(甚至手写制造数据)，或者加regularization，使曲线更加平滑，损失是可能会使bias增大，要调整regularization的weight
- 优化方法，n-fold cross方法寻找最好的model

P6 梯度下降

- Learning rate调整：开始设置较大的步长，几步之后慢慢减小，如
- • E.g. 1/t decay: $\eta^t = \eta / \sqrt{t+1}$
- 不同参数设置不同的步长，Tip1，**Adagrad**：

- ## Adagrad

  $$\eta^t = \frac{\eta}{\sqrt{t+1}} \qquad g^t = \frac{\partial C(\theta^t)}{\partial w}$$

  - Divide the learning rate of each parameter by the ***root mean square of its previous derivatives***

  Vanilla Gradient descent

  $$w^{t+1} \leftarrow w^t - \eta^t g^t$$

  w is one parameters

  Adagrad

  $$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

  σ^t: ***root mean square*** of the previous derivatives of parameter w

- ## Adagrad

  σ^t: ***root mean square*** of the previous derivatives of parameter w

  $$w^1 \leftarrow w^0 - \frac{\eta^0}{\sigma^0} g^0 \qquad \sigma^0 = \sqrt{(g^0)^2}$$

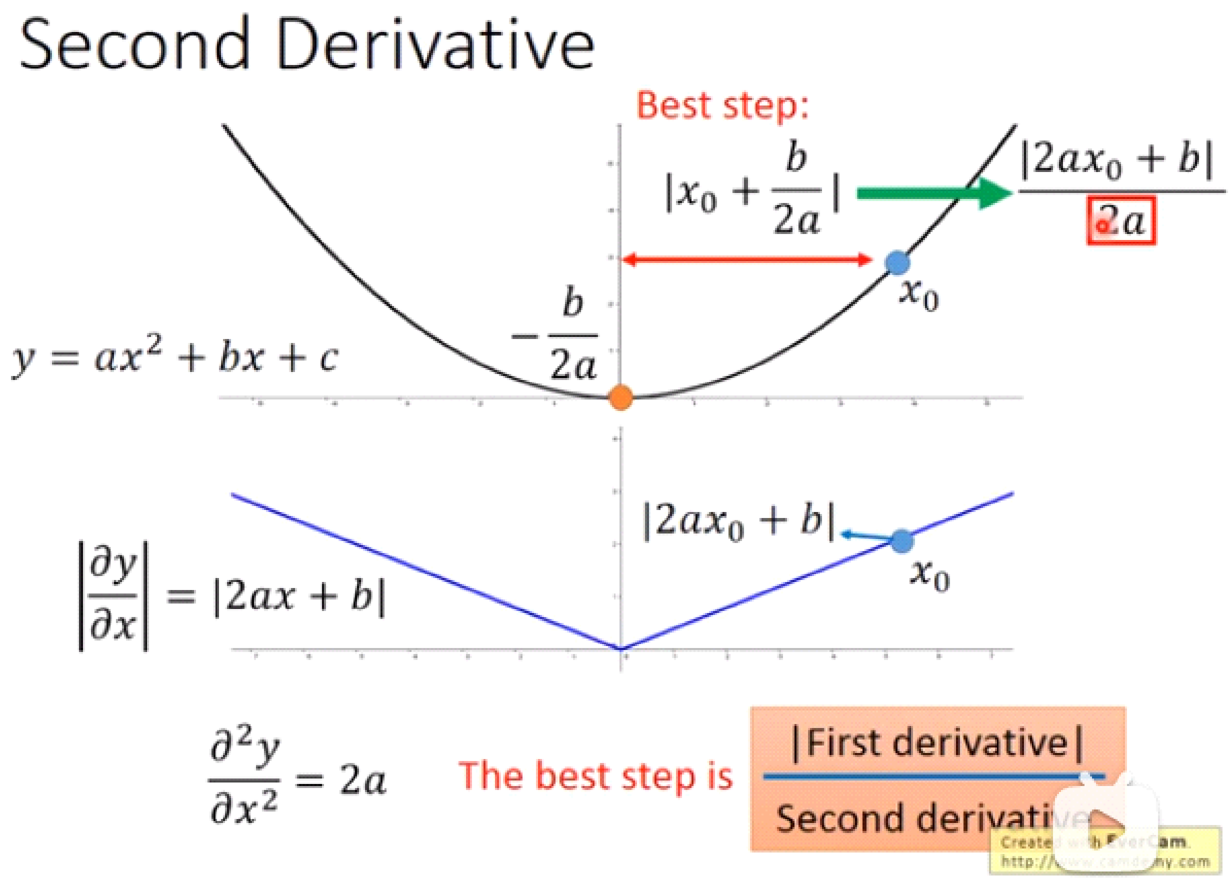
  $$w^2 \leftarrow w^1 - \frac{\eta^1}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\frac{1}{2}[(g^0)^2 + (g^1)^2]}$$

  $$w^3 \leftarrow w^2 - \frac{\eta^2}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\frac{1}{3}[(g^0)^2 + (g^1)^2 + (g^2)^2]}$$

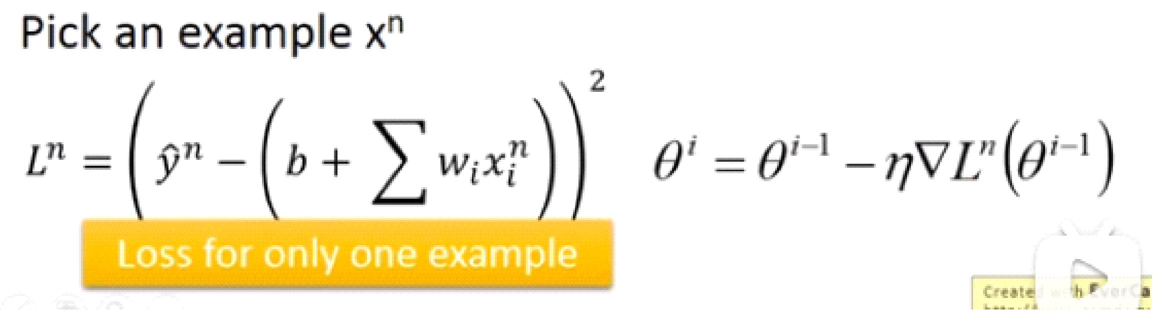
  $$\vdots$$

  $$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}(g^i)^2}$$

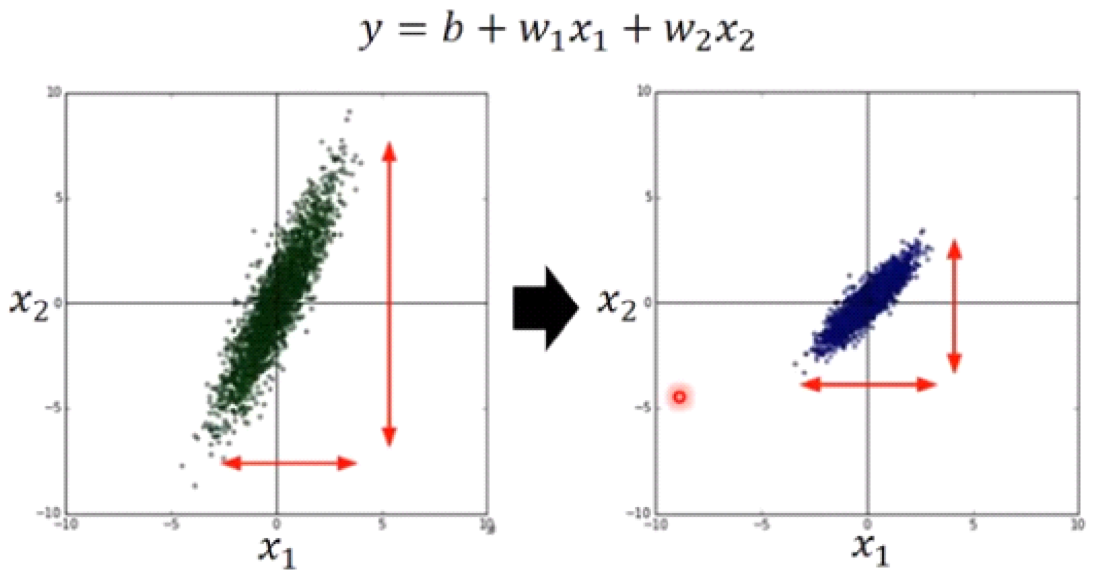
- 只看一个变量时，微分值越大，离原点越远，即步长可以更大，但是两个或多个变量时，不再成立，即两个变量比较时，某个变量微分值越小并不代表离原点更近。这时，要考虑二次微分再计算才能够反应该点到最低点的距离，此时是计算，不再是凭借比例比较。

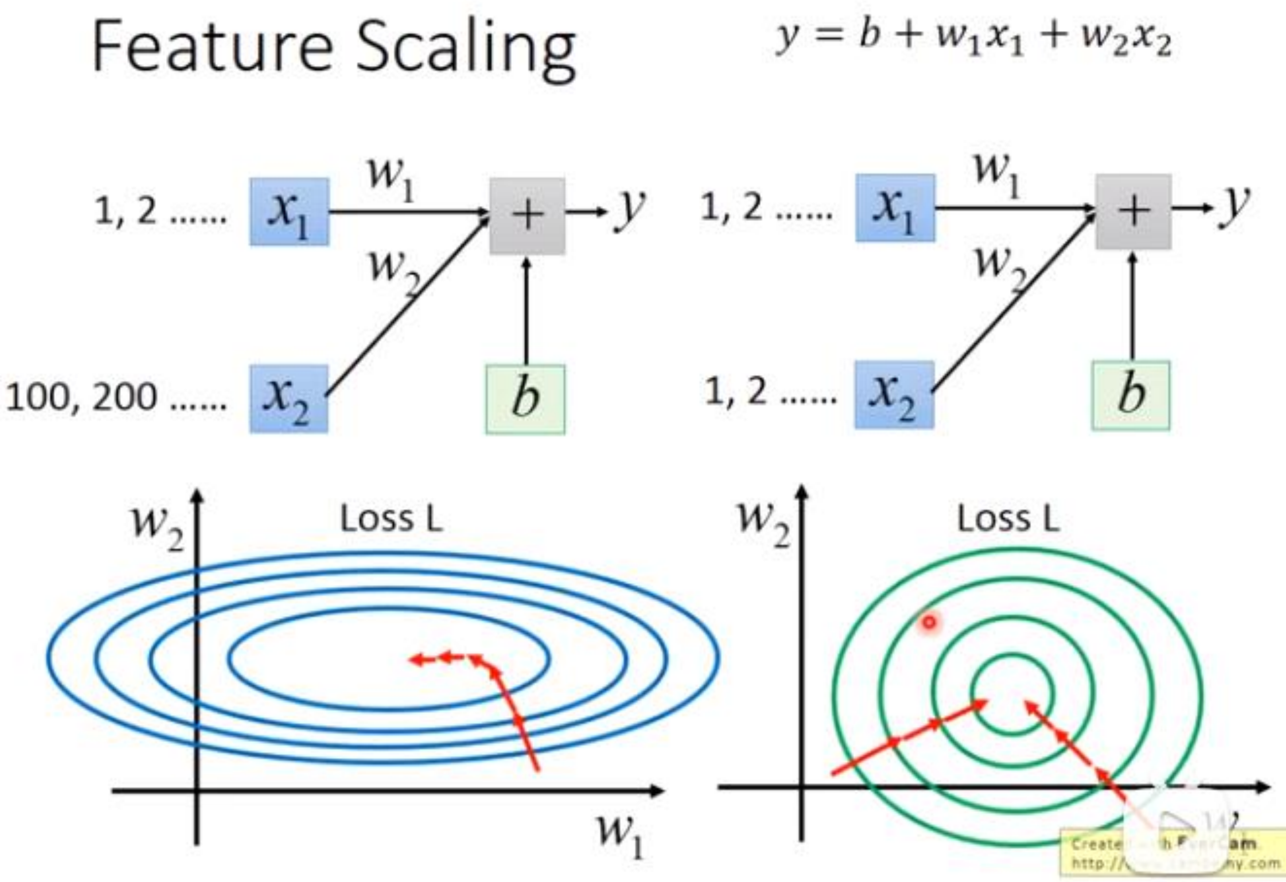
- 

- Tip2，**Stochastic Gradient Desent**

- 

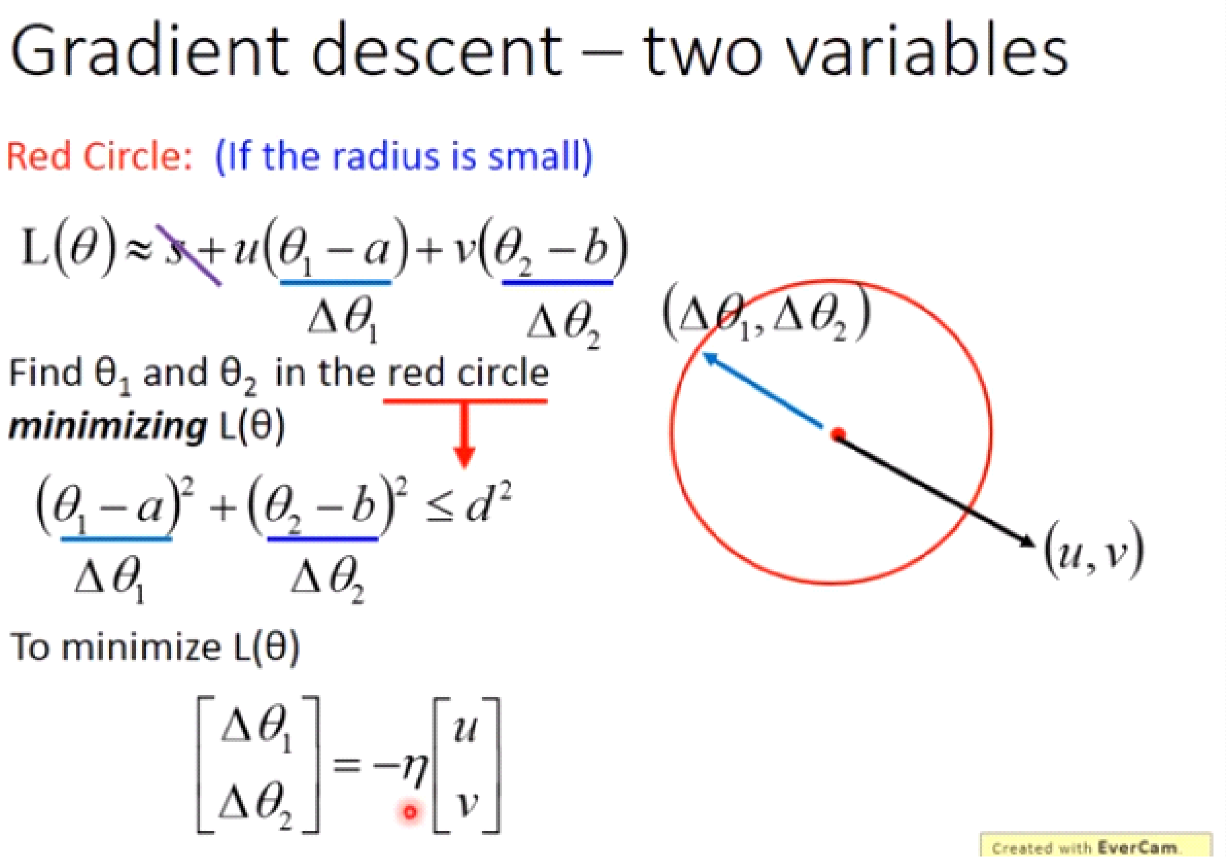
- Tip3，**Feature scaling**

- 

- 

- 第一个需要用Adagrad保证不同的参数有不同的学习率，但转变为图2时使用梯度下降便容易很多，使用标准化，使变量均值为0方差为1

通过圆圈递进靠近原点：



Back to Formal Derivation

Based on Taylor Series:

If the red circle is ***small enough***, in the red circle

constant

$$s = \mathrm{L}(a,b)$$

$$\mathrm{L}(\theta) \approx s + u(\theta_1 - a) + v(\theta_2 - b)$$

$$u = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_1}, v = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_2}$$

Find θ_1 and θ_2 yielding the smallest value of $L(\theta)$ in the circle

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \eta \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial \mathrm{C}(a,b)}{\partial \theta_1} \\ \frac{\partial \mathrm{C}(a,b)}{\partial \theta_2} \end{bmatrix}$$

This is gradient descent.