

4/19/2023

梯度下降.

目标函数  $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$

分别做  $\theta_0$  和  $\theta_1$

$$\frac{\partial J}{\partial \theta_0}$$

$$\frac{\partial J}{\partial \theta_1}$$

寻找最低点

加平方项: 放大效果:  $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (y^i - h_{\theta}(x^i))^2$

批量梯度下降:  $\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m} \sum_{i=1}^m (y^i - h_{\theta}(x^i)) x_j^i$

$$\theta_j' = \theta_j + \frac{1}{m} \sum_{i=1}^m (y^i - h_{\theta}(x^i)) x_j^i$$

速度很慢.

随机梯度下降:  $\theta_j' = \theta_j + (y^i - h_{\theta}(x^i)) x_j^i$

速度快, 但不一定每次都是朝着收敛方向.

小批量梯度下降:  $\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_{\theta}(x^k) - y^k) x_j^k$

↑ batch size = 10

batch size 大: 速度慢, 但更精确

batch size 小: 速度快, 但不那么精确.

learning rate:  $LR = 0.01, 0.03 \dots$  小一点比较好.