Maths secrets behind Supervised Learning

Animated by : Yesmine Makkes

GDSC HICS ISI Ariana

Some notations :

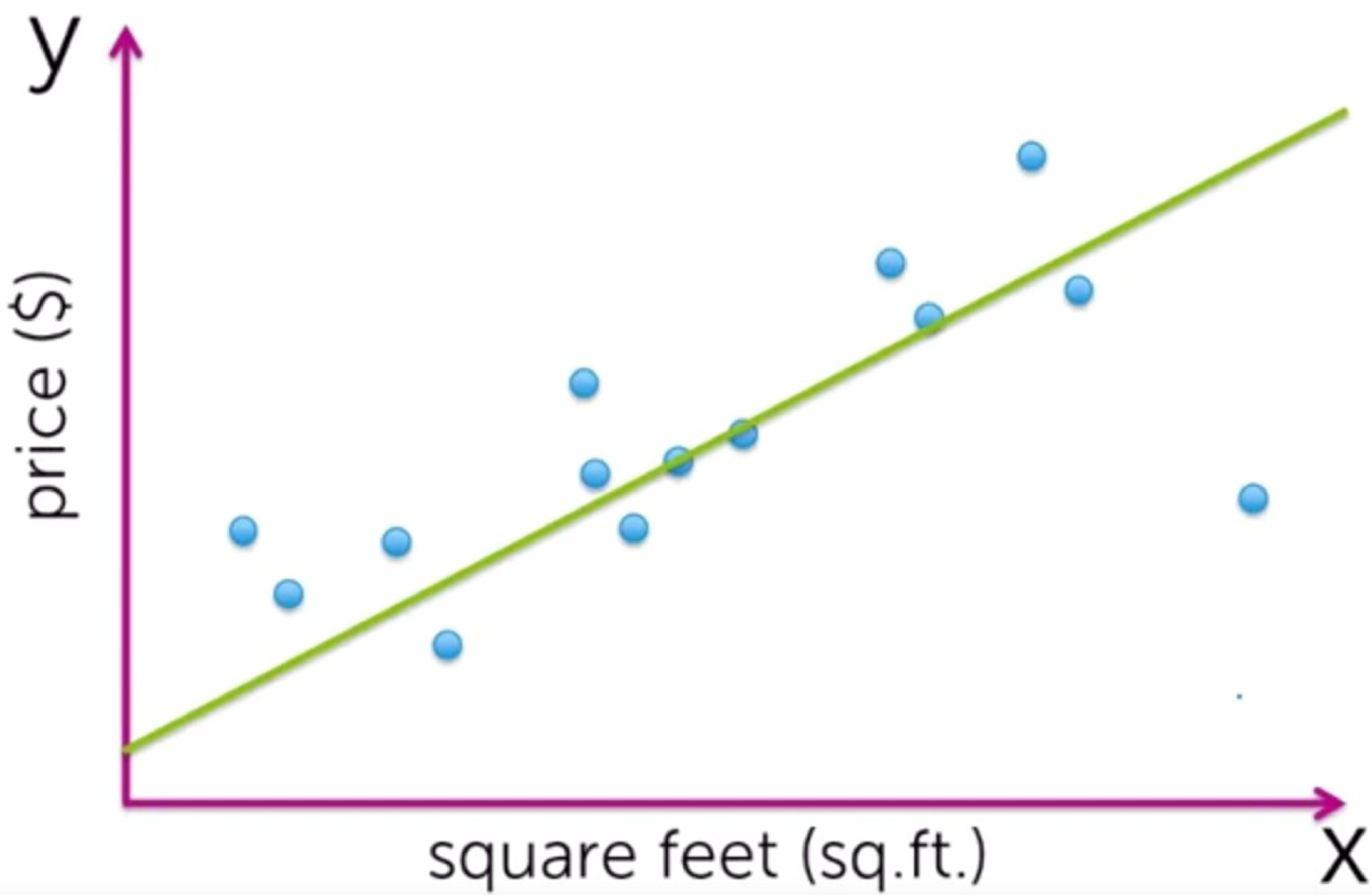
m : number of training examples

x : feature (input)   *

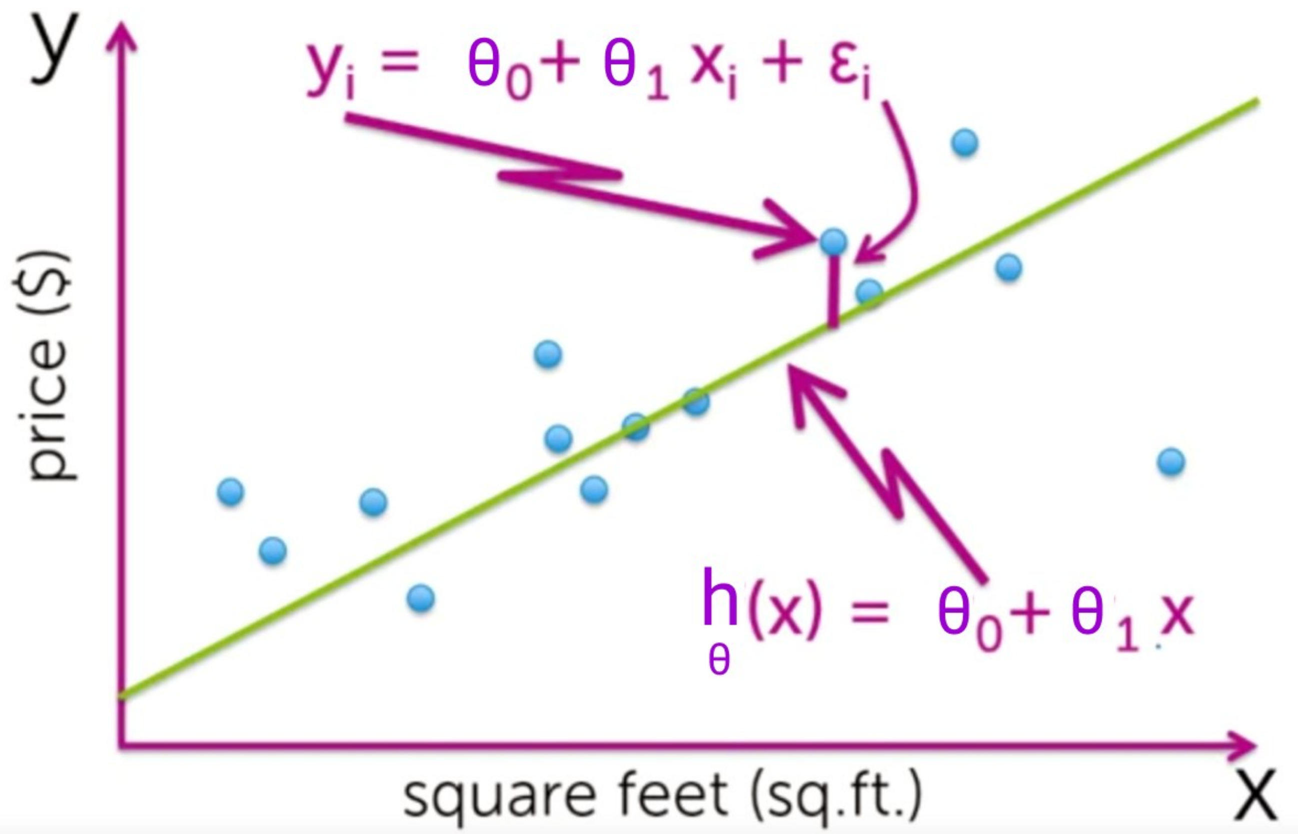
y : target / label (output)

Attributes				Decision
Length	**Height**	**Width**	**Weight**	**Quality**
4.7	1.8	1.7	1.7	high
4.5	1.4	1.8	0.9	high
4.7	1.8	1.9	1.3	high
4.5	1.8	1.7	1.3	medium
4.3	1.6	1.9	1.7	medium
4.3	1.4	1.7	0.9	low
4.5	1.6	1.9	0.9	very-low
4.5	1.4	1.8	1.3	very-low

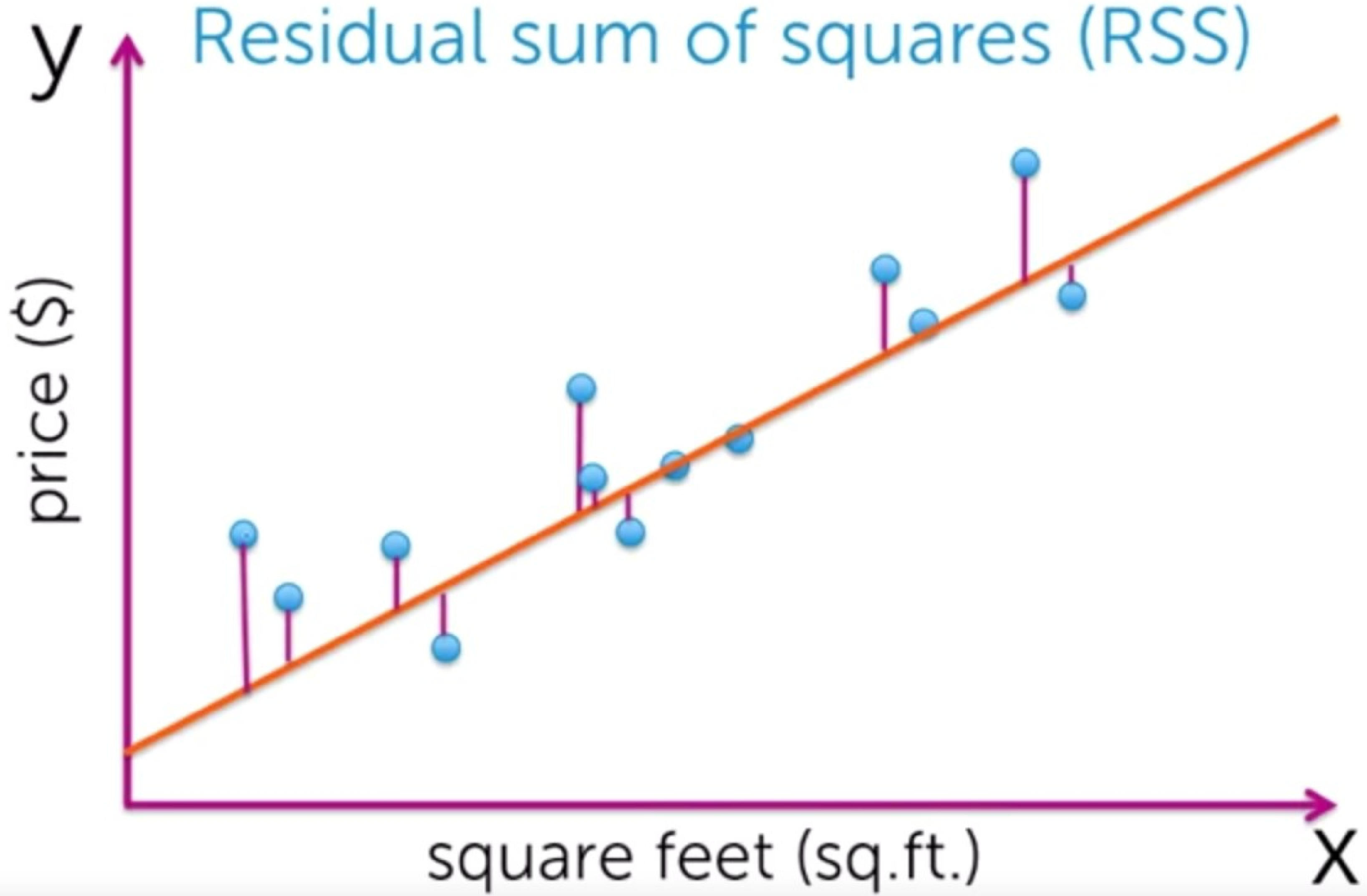
Simple linear regression model



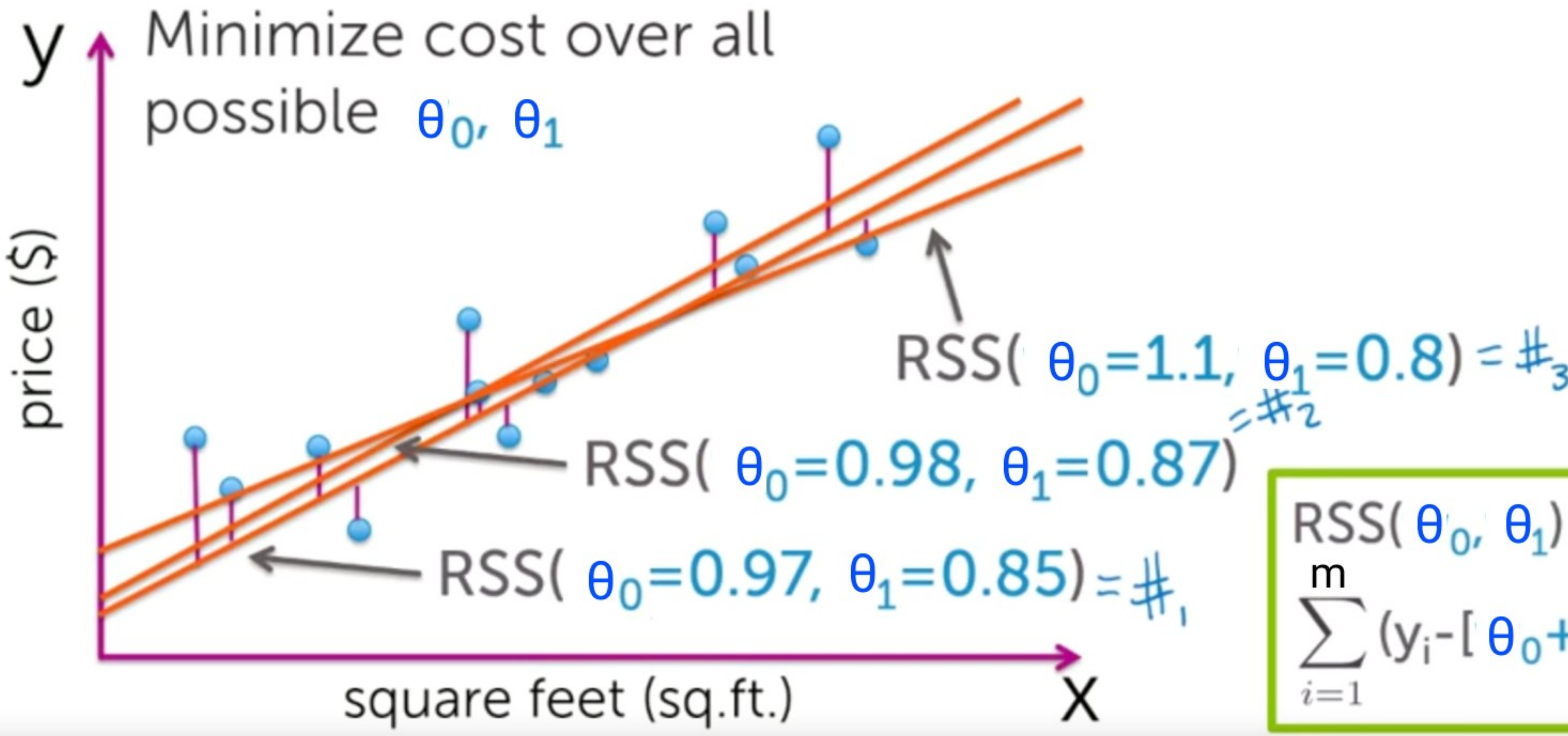
Simple linear regression model



"Cost" of using a given line

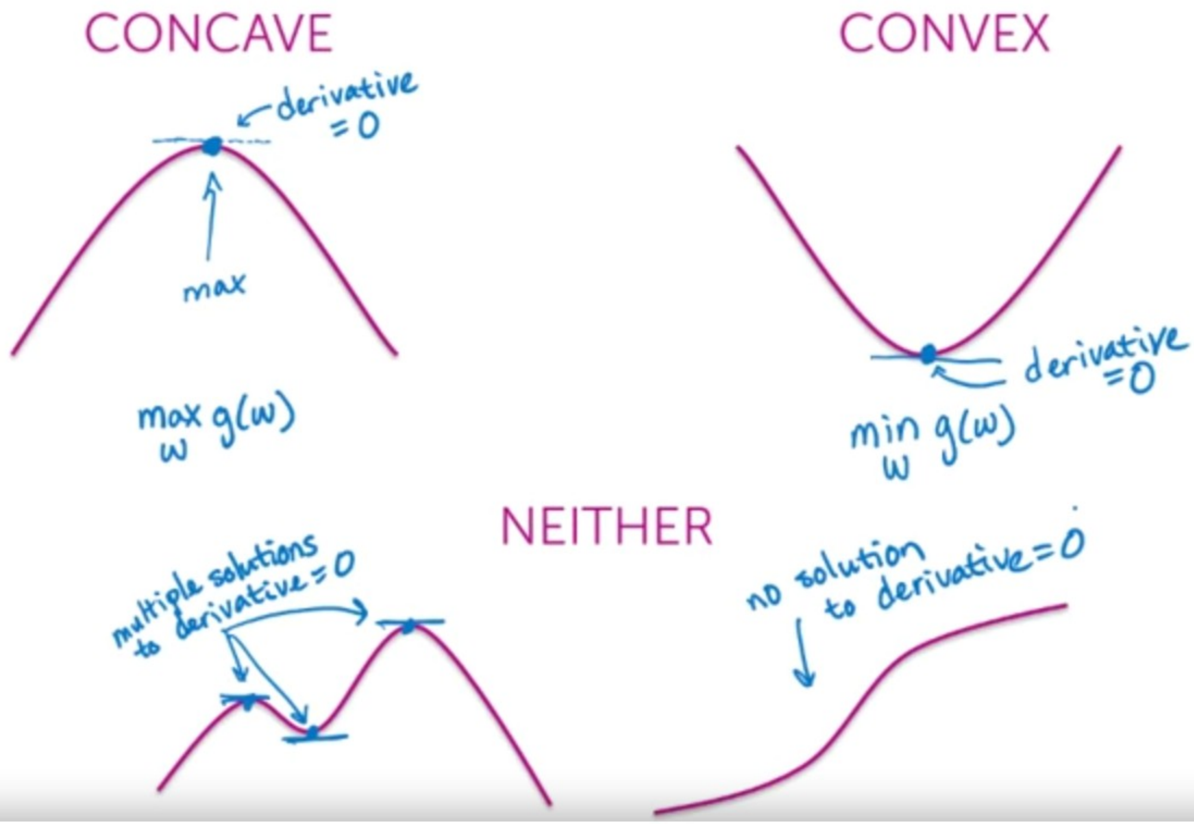


Find "best" line



$$RSS(\theta_0, \theta_1) = \sum_{i=1}^{m} (y_i - [\theta_0 + \theta_1 x_i])^2$$

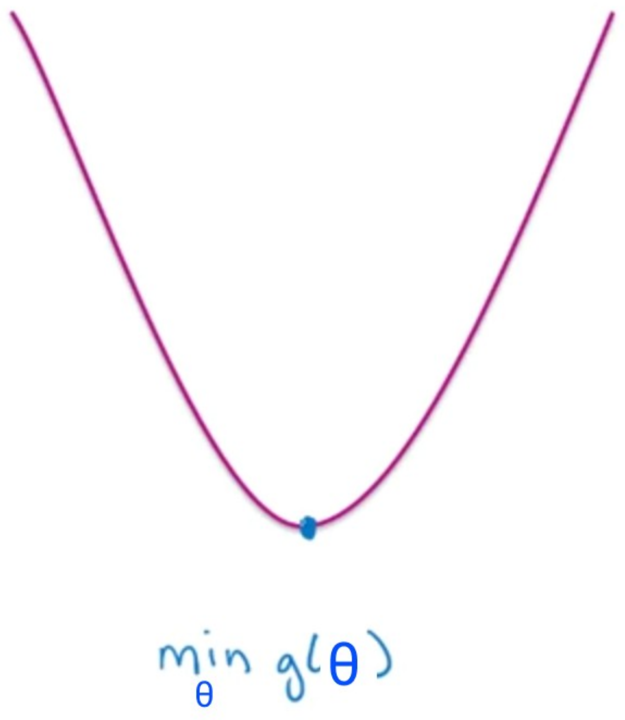
Finding the max or min analytically



Example:

$g(w) = 5-(w-10)^2$

Finding the min via hill descent

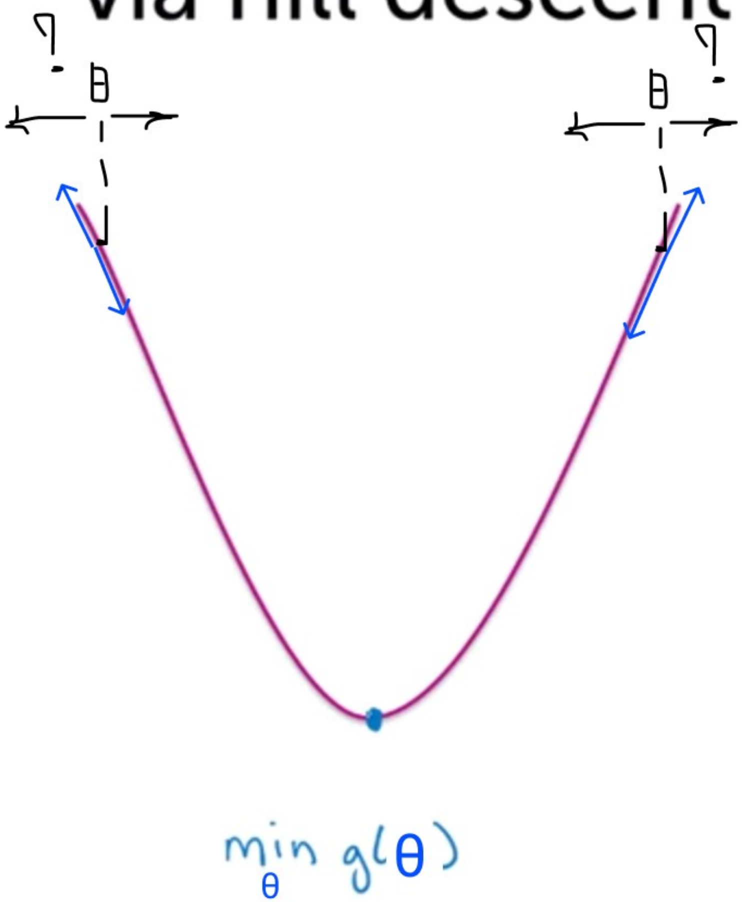


Algorithm:

while not converged

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{dg}{d\theta}\bigg|$$

Finding the min via hill descent



Algorithm:

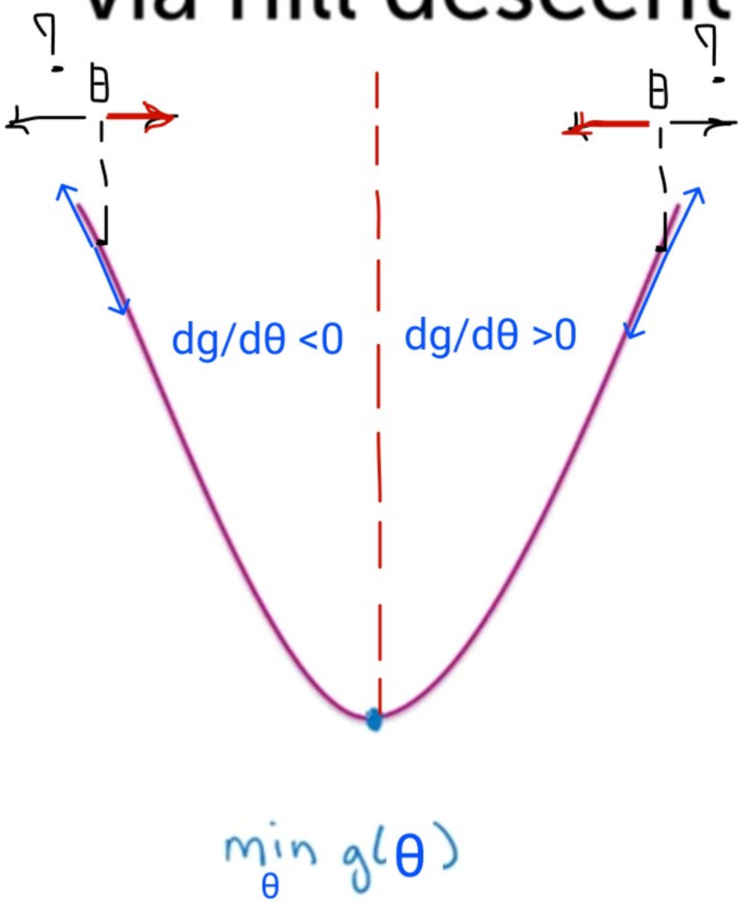
while not converged

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{dg}{d\theta}$$

iteration t

step size

Finding the min via hill descent



when derivative is positive we want to decrease θ , when the derivative is negative we want to increase θ

Algorithm:

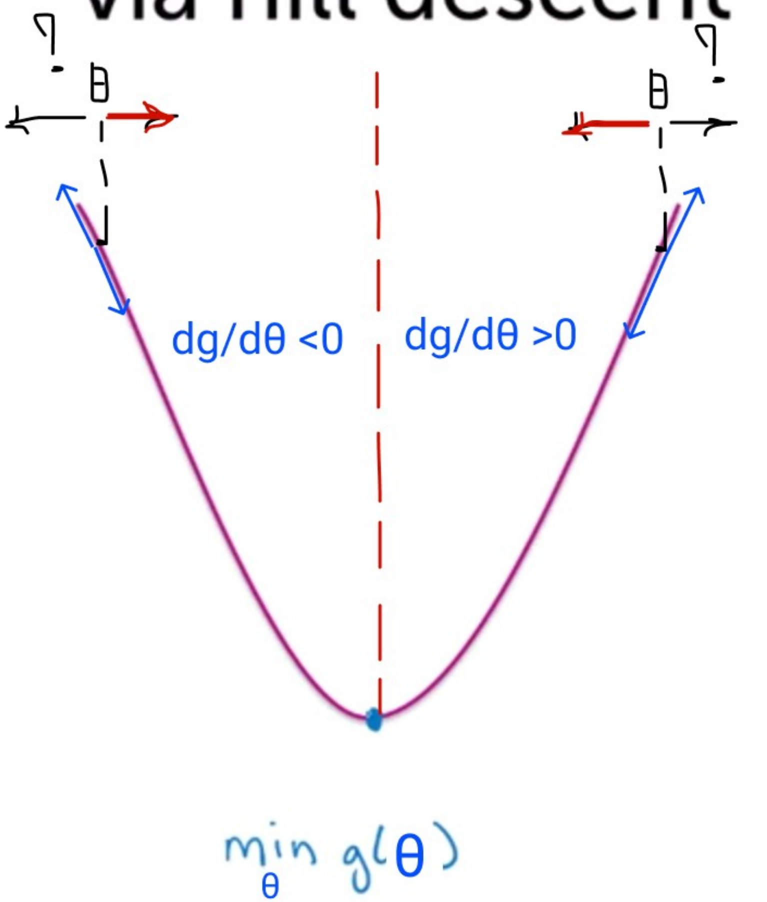
while not converged

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{dg}{d\theta}$$

iteration t

step size

Finding the min via hill descent



when derivative is positive we want to decrease θ , when the derivative is negative we want to increase θ

Algorithm:

while not converged

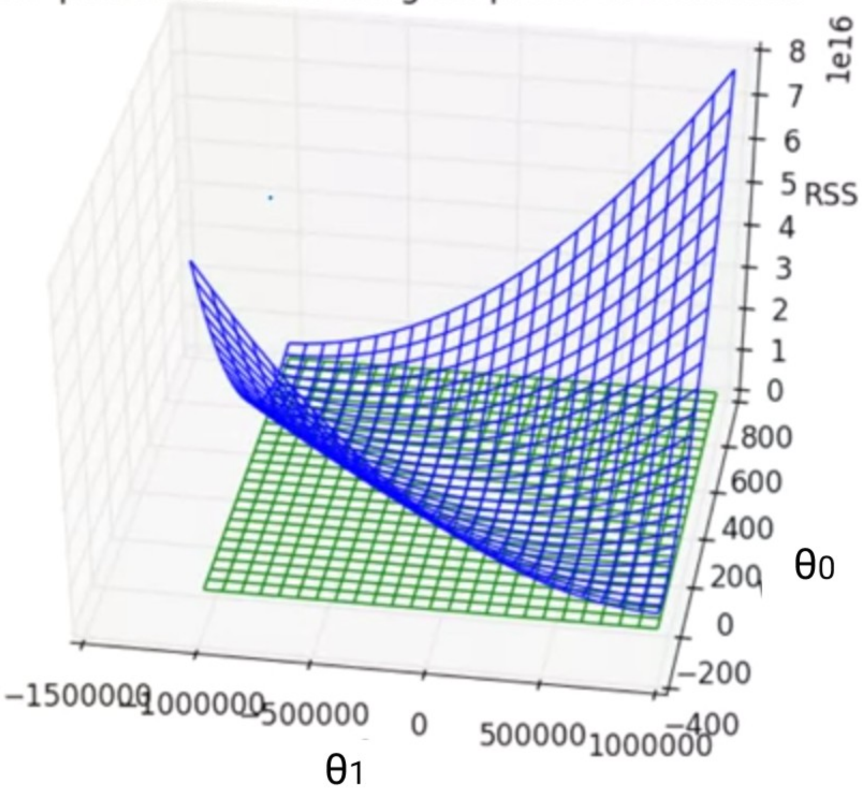
$$\theta^{(t+1)} \leftarrow \theta^{(t)} \boxed{-} \eta \frac{dg}{d\theta}$$

iteration t

step size

Minimizing the cost

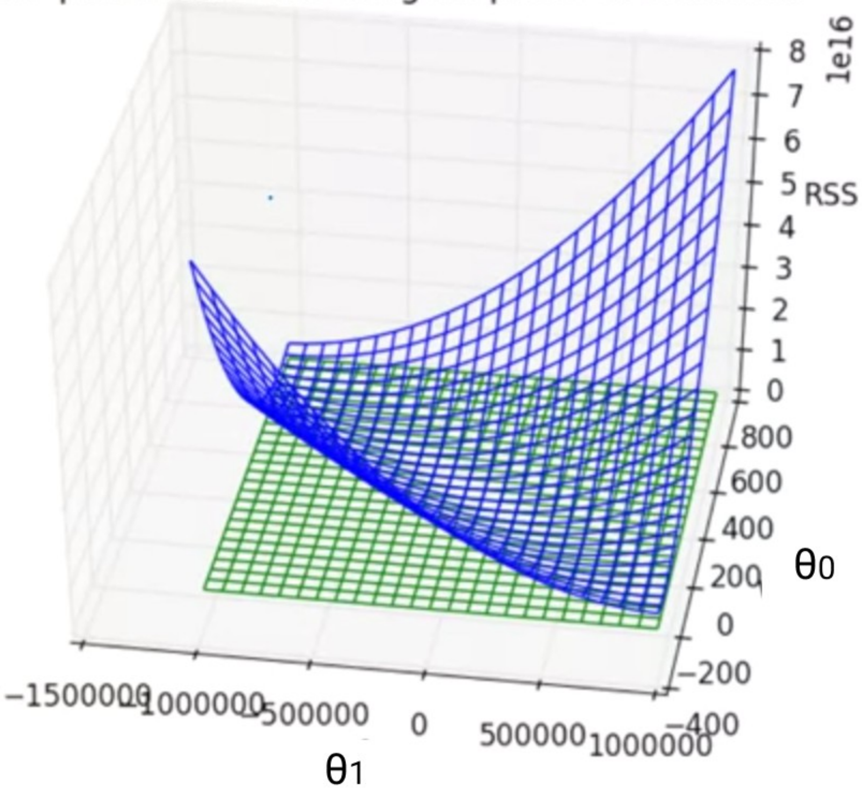
3D plot of RSS with tangent plane at minimum



$$\min_{\theta_0,\ \theta_1} \sum_{i=1}^{m} (y_i - [\,\theta_0 + \theta_1 x_i])^2$$

Minimizing the cost

3D plot of RSS with tangent plane at minimum



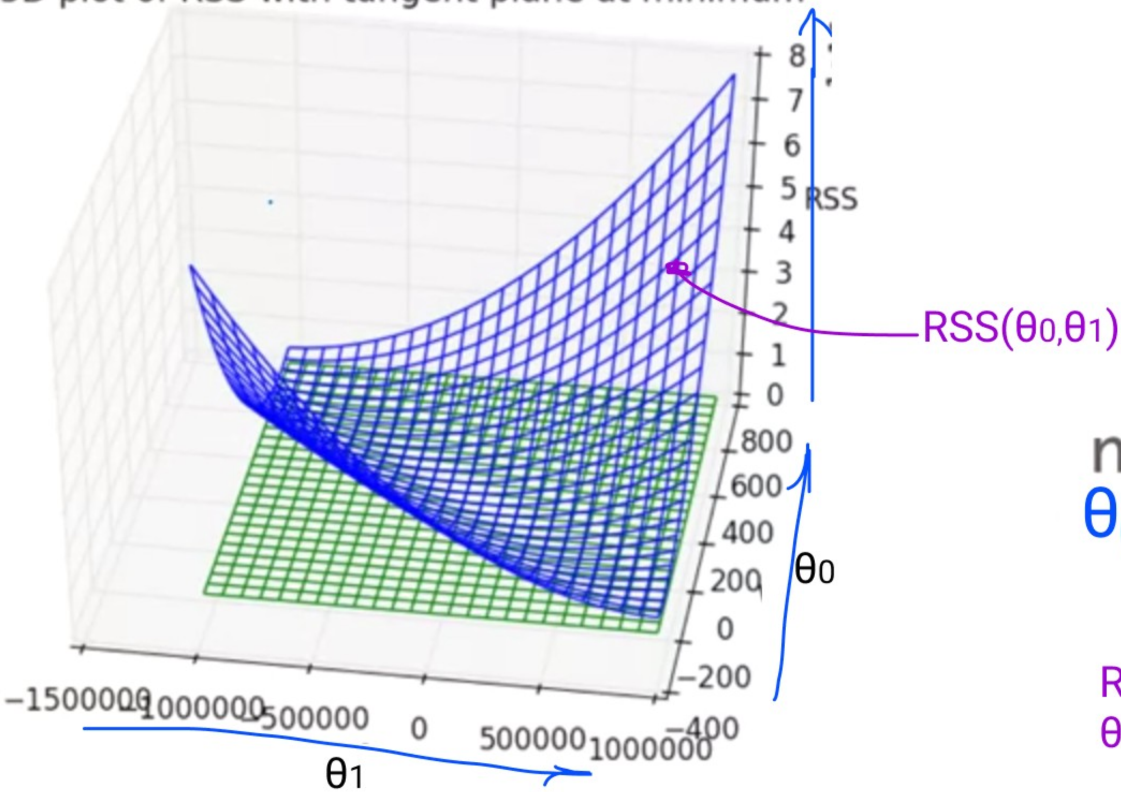
$h(x) = \theta_0 + \theta_1 x_i$

$$\min_{\theta_0, \theta_1} \sum_{i=1}^{m} (y_i - [\theta_0 + \theta_1 x_i])^2$$

RSS(θ_0, θ_1) is a function of 2 variables : θ_0, θ_1

Minimizing the cost

3D plot of RSS with tangent plane at minimum

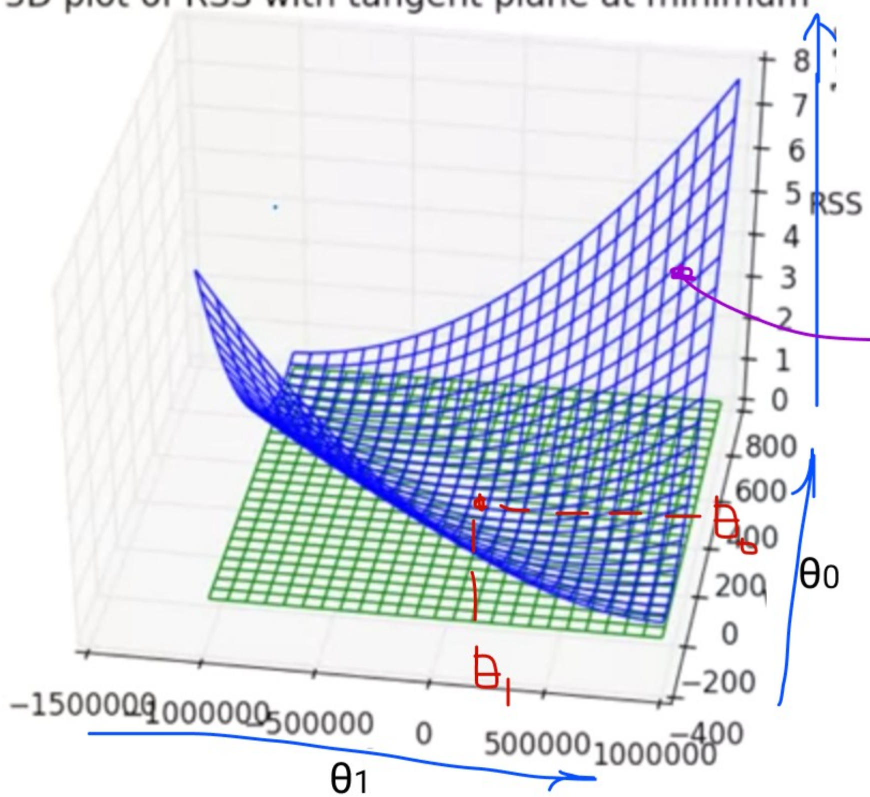


$h(x) = \theta_0 + \theta_1 x_i$

$$\underbrace{\min_{\theta_0, \theta_1} \sum_{i=1}^{m} (y_i - [\theta_0 + \theta_1 x_i])^2}$$

RSS(θ_0 , θ_1) is a function of 2 variables : θ_0 , θ_1

Minimizing the cost



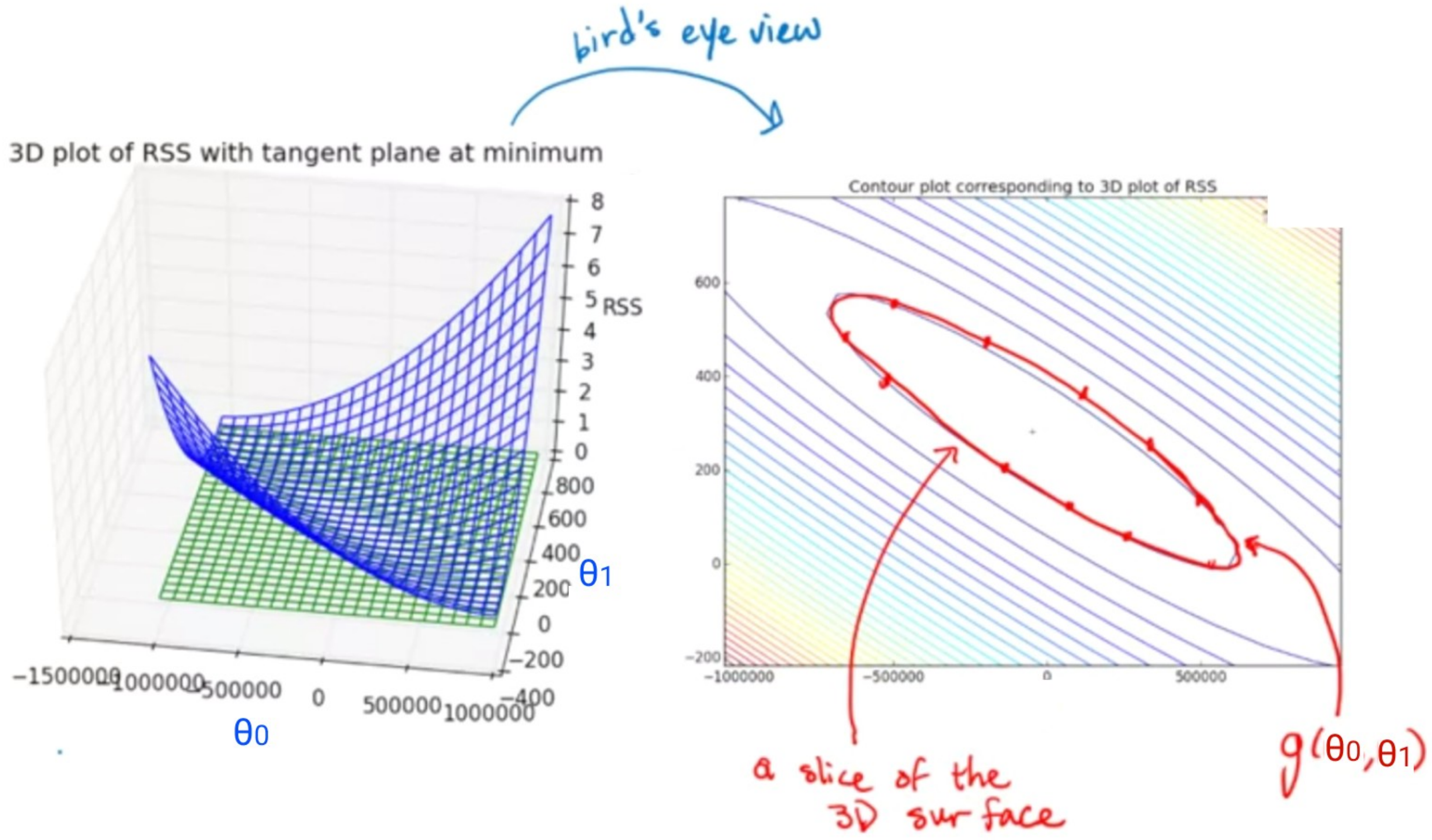
$$h(x) = \theta_0 + \theta_1 x_i$$

minimize RSS over all possible θ_0 , θ_1

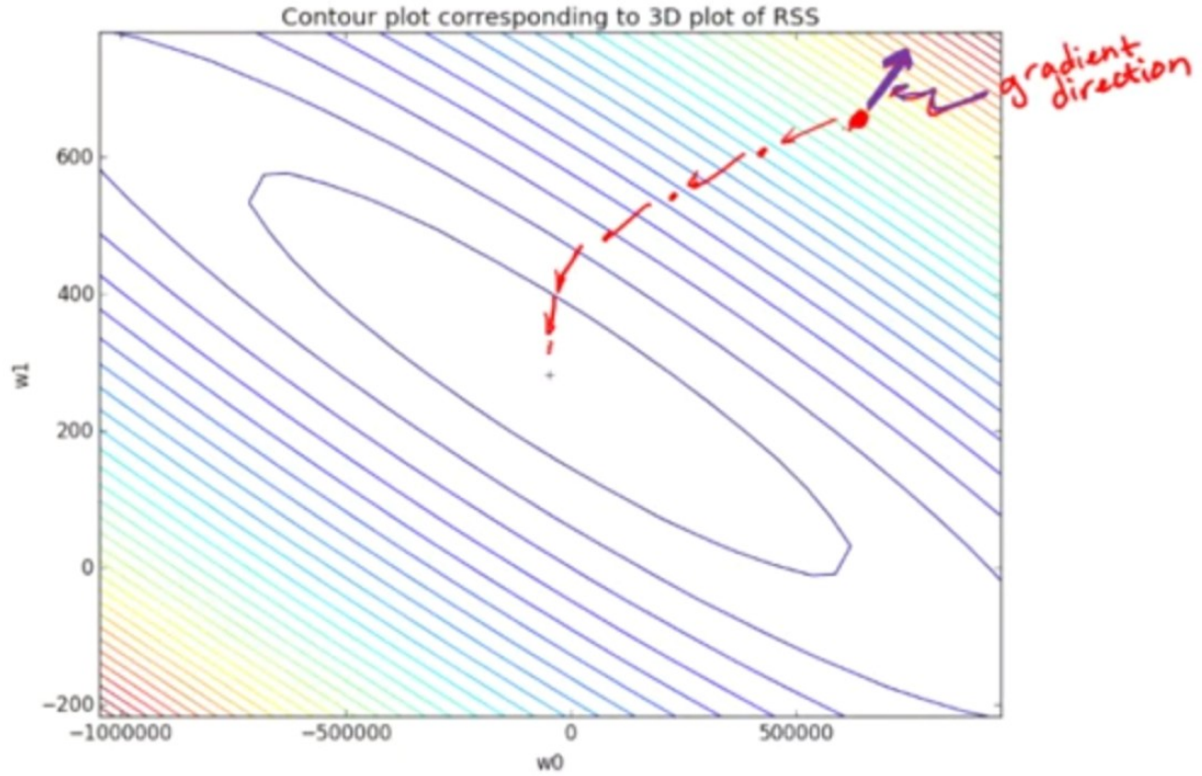
$$\underbrace{\min_{\theta_0,\ \theta_1} \sum_{i=1}^{m} (y_i - [\,\theta_0 + \theta_1 x_i])^2}$$

RSS(θ_0 , θ_1) is a function of 2 variables : θ_0 , θ_1

Contour plots



Gradient descent



Convergence criteria

For convex functions,

optimum occurs when

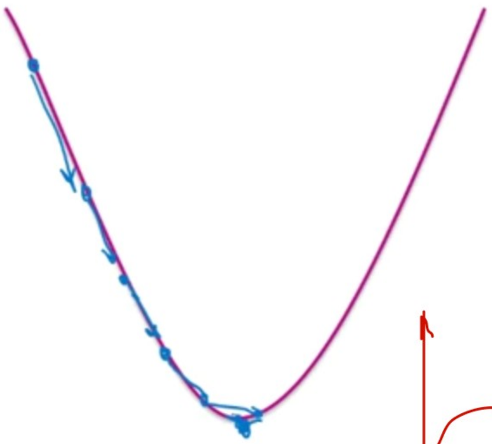
$$\frac{dg(w)}{dw} = 0$$

In practice, stop when

$$\left|\frac{dg(w)}{dw}\right| < \epsilon$$

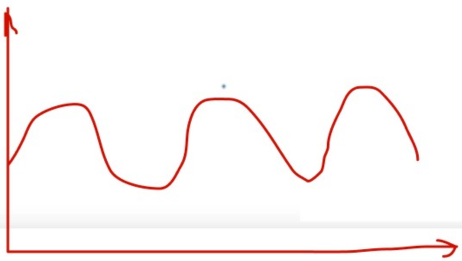
threshold to be set

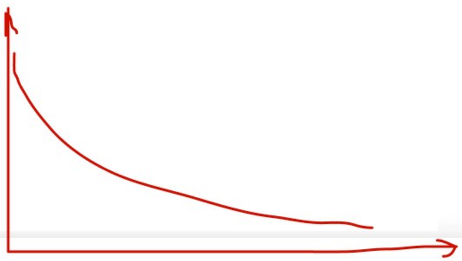
Choosing the stepsize

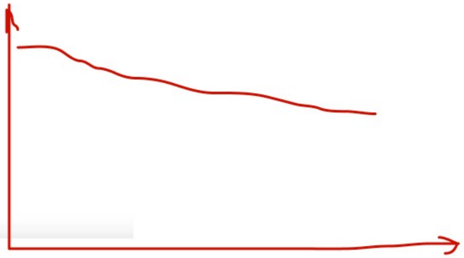


Common choices:

stepsize (learning rate) = 0.001 , 0.003, 0.009, 0.03, ...

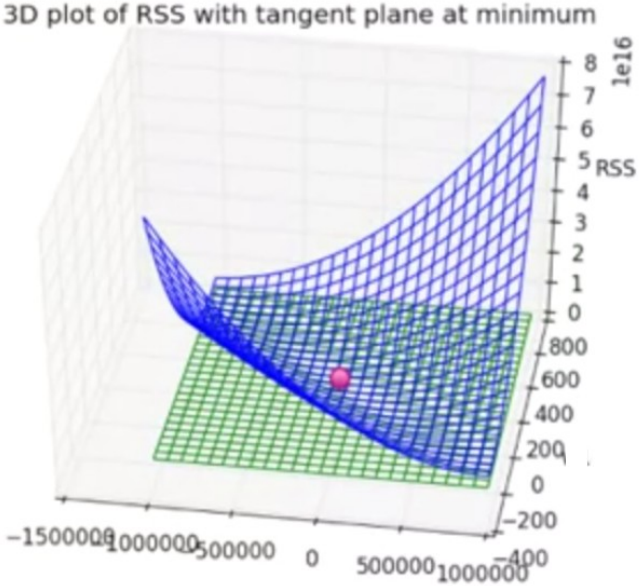






Approach 2 : Set gradient = 0

$$\nabla RSS(\theta_0, \theta_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - (\theta_0 + \theta_1 x_i)] \\ -2\sum_{i=1}^{N}[y_i - (\theta_0 + \theta_1 x_i)]x_i \end{bmatrix}$$



Comparing the approaches

- For most ML problems, cannot solve gradient = 0
- Even if solving gradient = 0 is feasible, gradient descent can be more efficient
- Gradient descent relies on choosing stepsize and convergence criteria

Thank you !