

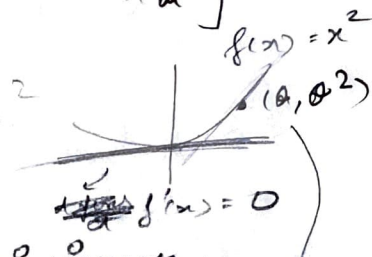
$$h(\theta) = \sum_{j=0}^n \theta_j x_j = \theta^T X \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

$(x^{(i)}, y^{(i)}) \xrightarrow{\text{training example}}$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_0(x) = h(x)$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)})^2$$



(cost func)

Task  $\rightarrow$  Choose  $\theta$  s.t.  $J(\theta)$  is minimum.

$$\frac{d x^2}{d x} = 2x$$

$$2x = 0$$

$$x = 0$$

$$\theta = 0.01 \times 20$$

$$\theta = 0.020$$

$$1.98 \theta = 0.01$$

① LMS Algo (Gradient descent)  
 Starts with  $\theta$  (say  $\theta = \vec{0}$ )

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} (J(\theta))$$

assignment operator

$\alpha$  learning rate  
 (size of step towards a minimum)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_0(x) - y)$$

$$= \frac{1}{2} \cdot 2 (h_0(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left( \sum_{i=0}^n \theta_i x_i - y \right)$$

$$= (h_0(x) - y) \cdot (x_j)$$

$$\theta_j := \theta_j - \alpha (h_0(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

Repeat this until convergence

(for  $j = 0, 1, \dots, n$ ) {

$$\theta_j := \theta_j - \alpha \sum_{i=0}^m (h_0(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

}