Vectorization -

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

$$= \theta^T x$$

vectorized implementation

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

prediction = $\theta^T * x$

Gradient descent

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

vector

$$\theta := \theta - \alpha \delta \leftarrow \text{vector}$$

vector

$$\text{i.e., } \delta = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

Epinov using augmented reality

→ Matlab uses 'std' function for calculating standard deviation.

X_norm = X 97 × 2 Matrix

mu = zeros (1, 2) 0 0 (mean)

sigma = zeros (1, 2) (std)

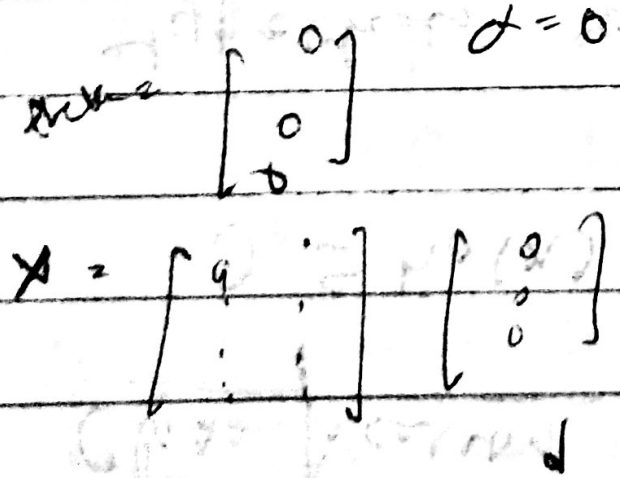
$$\frac{X - \text{mean}}{\text{std.}}$$

2 features

for c = 1 : size (X, 2)

$\alpha = 0.01$ num_iters = 400

[0 0 2]



Cost function

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)} - y^{(i)}) \right)^2$$

Gradient descent.

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

Classification (Logistic regression)

$$0 \leq h_\theta(x) \leq 1$$

Hypothesis representation for linear regression.

$$- h_\theta(x) = \theta^T x.$$

for logistic regression

$$\therefore h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1 + e^{-z}} \quad \text{(sigmoid function)}$$