COURSERA ML: WEEK 2

Linear Regression w/ Multiple Variables:

eg:

Size	# bedrooms	# floors	age	price

$1 \to m$

4 variables / features $x_1, x_2, x_3, x_4 \longrightarrow 1 \to n$

↳ # features = # variables = n (above $n = 4$)

↳ $x_j^{(i)}$ = val of feature j (col.) in i'th training example (row)

eg: $x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix}$ ← Size, ← # bedrooms, etc. ↓ $\longrightarrow x^{(2)} \in R^4 \ (R^n)$

↳ $x_3^{(2)} = 2$

↳ NOW: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

$(h_\theta(x) = \theta_0 + \theta_1 x_1 \cdots + \theta_n x_n)$

↳ for convenience, $x_0^{(i)} = 1 \longrightarrow x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{n+1}$ (always = 1; n = # features)

NOTE: $J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y)$ good form.

& $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$ $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ 2×1

eg: $X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} \\ x_0^{(2)} & x_1^{(2)} \\ \vdots & \\ x_0^{(m)} & x_1^{(m)} \end{bmatrix}$ $m \times 2$, $n = 1$ here

$\Rightarrow \underline{h_\theta(x) = \theta^T x} \quad (= \theta \cdot x)$

↳ "multivariate linear regression"

$h_\theta(x) = X\theta$

Gradient Descent for Multiple Variables

Hypothesis: $h_\theta(x) = \theta \cdot x = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \dots \theta_n x_n$ ($x_0 = 1$)

Parameters: $\theta_0, \theta_1, \theta_2 \dots \theta_n = \underline{\theta}$ ($n+1$-dimensional vector)

Cost Fnt: $J(\theta_0, \theta_1, \dots \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 = J(\underline{\theta})$

$\Rightarrow$ Gradient Descent: $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\underline{\theta})$ for $j = 0 \dots n$

for $n \geq 1 \rightarrow \frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$

↳ $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ (w/ simultaneous update!)

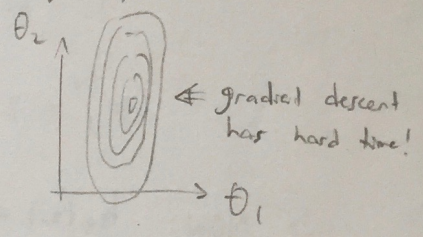
Gradient Descent in Practice I: Feature Scaling & Mean Normalization

$\Rightarrow$ When features are on similar scale, gradient descent converges more quickly

↳ eg 2 features x_1 - size $\rightarrow$ $0 - 2000\ ft^2$

x_2 - # bedrooms $\rightarrow$ $1 - 5$

$\Rightarrow$



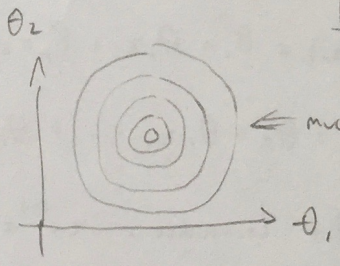
← gradient descent has hard time!

≈ range
or ≈ std. dev.

→ SCALE FEATURES:

$x_1 = \text{size}/2000\ ft^2 \rightarrow 0 - 1$

$x_2 = \#\text{bedrooms}/5 \rightarrow 0 - 1$



← much better

$\Rightarrow$ generally, get every feature into a $\underline{-1 \leq x_i \leq 1}$ range (or close enough)

↳ rule of thumb: $-3 \leq x_i \leq 3$ OK, $-\frac{1}{3} \leq x_i \leq \frac{1}{3}$ OK

$\Rightarrow$ another strategy: Mean Normalization: $x_i \rightarrow \frac{x_i - \mu_i}{s_i}$ ← mean normalization

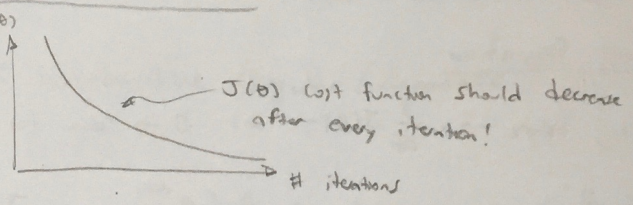
where μ_i = average

s_i = max - min or std. dev

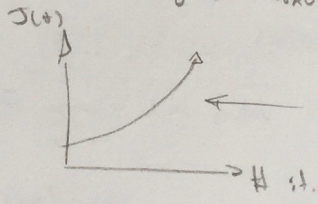
↳ eg. $x_1 = \frac{\text{size} - 1000}{2000}$, $x_2 = \frac{\#\text{bed} - 2}{5}$ ← feature scaling

...dient Descent in Practice II: De-bugging & Chosing α:

=> helpful: plot $J(\theta)$ vs. # iterations



=> declare some convergance threshold, e.g. $J(\theta)$ decreases by less than 10^{-3}

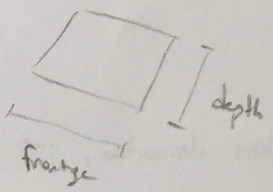
=> if 

likely that α is too big!

↳ for sufficiently small α, $J(\theta)$ should decrease on every iteration (too small = slow)

Features & Polynomial Regression:

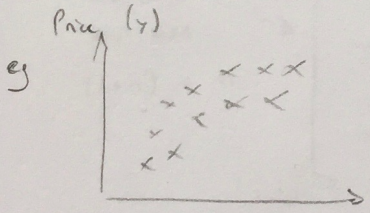
eg. Housing Price Prediction: $h_\theta(x) = \theta_0 + \theta_1 \cdot \text{frontage} + \theta_2 \cdot \text{depth}$



↳ possibly decide area is real feature!

↳ $h_\theta(x) = \theta_0 + \theta_1 x$ where $x = \text{frontage} \cdot \text{depth}$

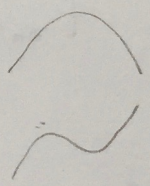
=> Polynomial Regression:



eg

-> fit quadratic: $\theta_0 + \theta_1 x + \theta_2 x^2$

-> fit cubic: $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$



↳ $x = \text{size}$

=> $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ where $x_1 = \text{size}$, $x_2 = \text{size}^2$, $x_3 = \text{size}^3$

↳ say size ranges $1 \leq x \leq 1000$ ← FEATURE SCALING V. IMPORTANT

↳ another possibility: $h_\theta(x) = \theta_0 + \theta_1 \text{size} + \theta_2 \sqrt{\text{size}}$

Computing Parameters Analytically

Normal Equation

Minimizing before → $\frac{d}{d\theta} J(\theta)$ → set $= 0$ → solve for real # θ.

↳ θ is now a vector → $\theta \in \mathbb{R}^{n+1}$, $J(\theta_0, \theta_1, \theta_2 \dots \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

$\Rightarrow \frac{d}{d\theta_j} J(\theta) = \dots = 0$ for each j → solve for $\underbrace{\theta_0, \theta_1, \theta_2 \dots \theta_n}_{\text{values that minimize } J(\theta)}$

ex] $m = 4$ training examples

x_0	size x_1	# beds x_2	# floors x_3	age (yrs) x_4	price y
1	2104	5	1	45	460
.	.	.	.	.	.
.	.	.	.	.	.
1	852	2	1	36	178

$m \times (n+1)$

$$\Rightarrow X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & . & . & . & . \\ 1 & . & . & . & . \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

$$y = \begin{bmatrix} 460 \\ . \\ . \\ 178 \end{bmatrix} \quad m \times 1$$

$\Rightarrow$ Skips derivation, → $\theta_{min} = (X^T X)^{-1} X^T y$

$\Rightarrow$ generally: m training examples $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$ & n features

$$x^{(1)} = \begin{bmatrix} x_0^{(1)} \\ x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} \in \mathbb{R}^{n+1} \Rightarrow X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ \vdots \\ — (x^{(m)})^T — \end{bmatrix}$$

DESIGN MATRIX $m \times (n+1)$

* Here, feature scaling not necessary! Not using gradient descent

$\Rightarrow$ w/ m training examples, n features:

Gradient Descent	Normal Eqn
• need to choose α	• no need to choose α
• need many iterations	• no iterations
• works well, even w/ large n	• slow if n is very large → $(X^T X)^{-1}$ is $n \times n$, $O(n^3)$
• $O(kn^2)$	↳ max $n \approx 10{,}000$

...nal Equation Non-Invertibility:

$\theta = (X^T X)^{-1} X^T y$ -> what if $X^T X$ non-invertible / singular / degenerate?

↳ should happen pretty rarely

↳ octave has pinv & inv -> using pinv guarantees no problem

↳ causes for non-invertibility: • redundant features (eg. x_1 = size in feet, x_2 = size in metres)

↳ linearly dependent

• too many features (eg. $m = 10$, $n = 100$). -> $m \leq n$

↳ $\theta \in \mathbb{R}^{101}$

↳ delete linearly dependent features or excess features / regularization