Machine Learning Specialization

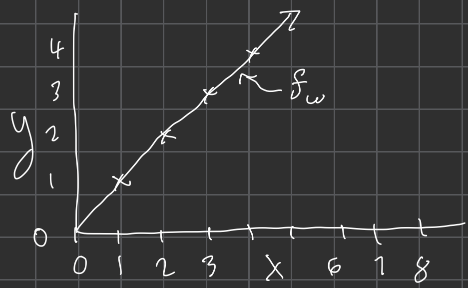
The Cost Function + Gradient Descent

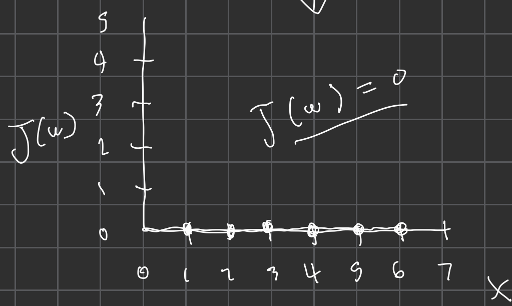
$f_w(x)$ = function of x, fixed w

$J(w)$ = function of w

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_w(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(w x^{(i)} - y^{(i)}\right)^2 = \frac{1}{2m}\left(0^2 + 0^2 + 0^2\right)$$

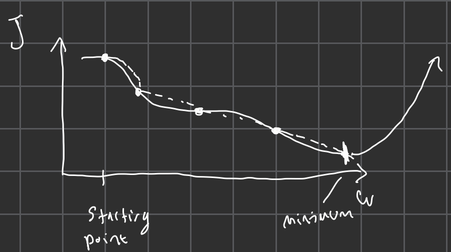




$\min\limits_{w,b} J(w,b)$ minimize the cost fn

$\min\limits_{w_1, w_2, \ldots, w_n, b} J(w_1, w_2, \ldots, w_n, b)$

- Start with some w, b, then iteratively change values to minimize the cost fn



Main update formula: $w = w - \alpha \frac{d}{dw} J(w,b)$

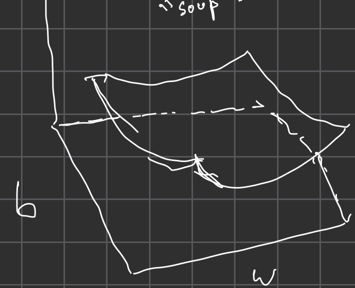
learning rate

$x_i, y_i \in D$

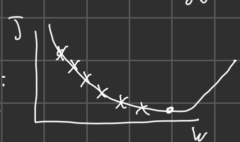
$b = b - \alpha \frac{\partial}{\partial b} J(w,b)$

$J(w,b)$

looks like a "soup bowl"

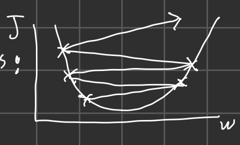


α is small: then lots of small updates



fail to converge and even diverge

α is big, lots of big updates



Multiple Lin Regression

House Prediction

- size (ft)
- bedrooms
- floors
- Age (years)
- price ($)

x_j = jth feature

n = num features

$\vec{x}^{(i)}$ = ith training example

$x_j^{(i)}$ = jth feature of ith sample

$$f_{w,b}(x) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b$$

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b = b + \sum_{j=0}^{N} w_j x_j$$

Previous Notation

params: $w_1, \dots, w_n$

Model: $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + \dots + w_n x_n + b$

cost fn: $J(w_1, w_2, \dots, w_n)$

Vector Notation

$\vec{w} = [w_1, w_2, \dots, w_n]$, b

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

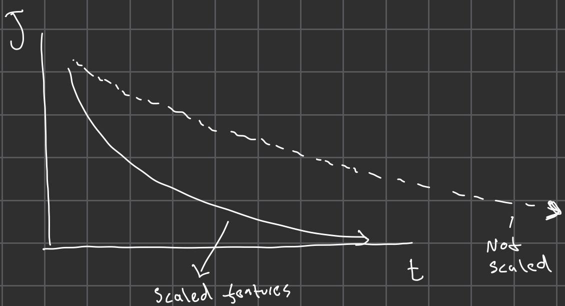
$J(\vec{w}, b)$

Gradient Descent

$$\longrightarrow \quad w_j = w_j - \alpha \frac{\partial}{\partial w_j}\left(J(\vec{w}, b)\right)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

$$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right) x_n^{(i)}$$



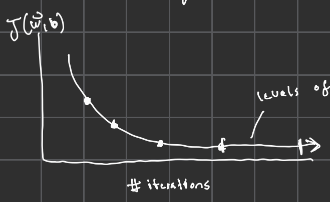
Mean normalization:

$$x_1 = \frac{x_1 - \mu_1}{max - min}$$

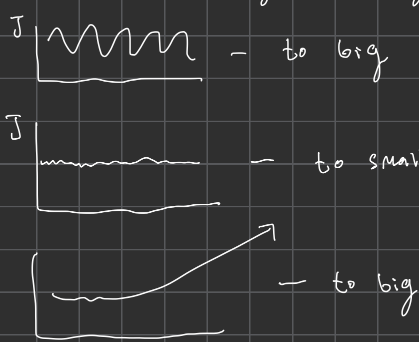
Z-score Normalization:

$$x_1 = \frac{x_1 - \mu_1}{\sigma_1}$$

- make sure gradient descent is working



- How to choose a good learning rate?

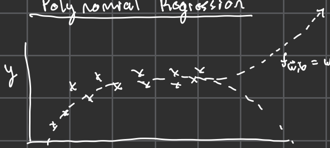


$w_1 = w_1 + \alpha d_1$:(

$w_1 = w_1 - \alpha d_1$:)

- try a range of values and see what produces a good learning graph

Polynomial Regression



$f_{\vec{w},b} = w_3x^3 \ldots\ldots$

$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$