Linear Regression

Linear regression model

$$f_{w,b}(x) = wx + b$$

Cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

Gradient descent algorithm

repeat until convergence {

$$w = w - \alpha \frac{\partial}{\partial w} J(w,b) \rightarrow \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b) \rightarrow \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})$$

}

x^i = input variable / feature
y^i = output variable / target
m = number of training examples
$(x^{(i)},y^{(i)})$ = ith training example

Feature (x) → Model → Prediction (ŷ i.e. estimated y)

Regression model predicts Numbers. There can be infinitely many possible outputs.
Classification model predicts categories. There are small number of possible outputs.

1. Simple Linear Regression

It is also called univariate linear regression.
It means linear regression with one variable (single feature x)

Model: $f_{w,b}(x) = wx + b$
w,b: parameters to improve the model

b = y-intercept (Point where it crosses the y-axis)
w = slope of the line

If w = 0, then f(x) will be equal to b (y-intercept)

$y^{(i)} = f_{w,b}(x^{(i)})$
$f_{w,b}(x^{(i)}) = wx^{(i)} + b$

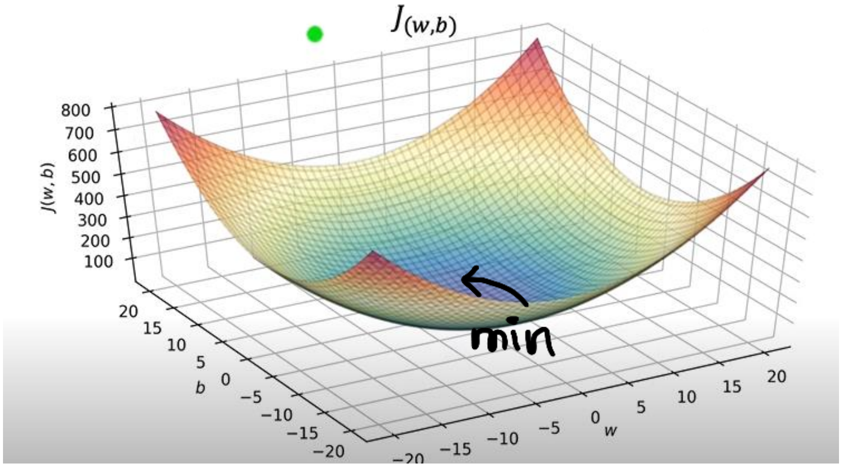
Find w,b such that $\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)},y^{(i)})$

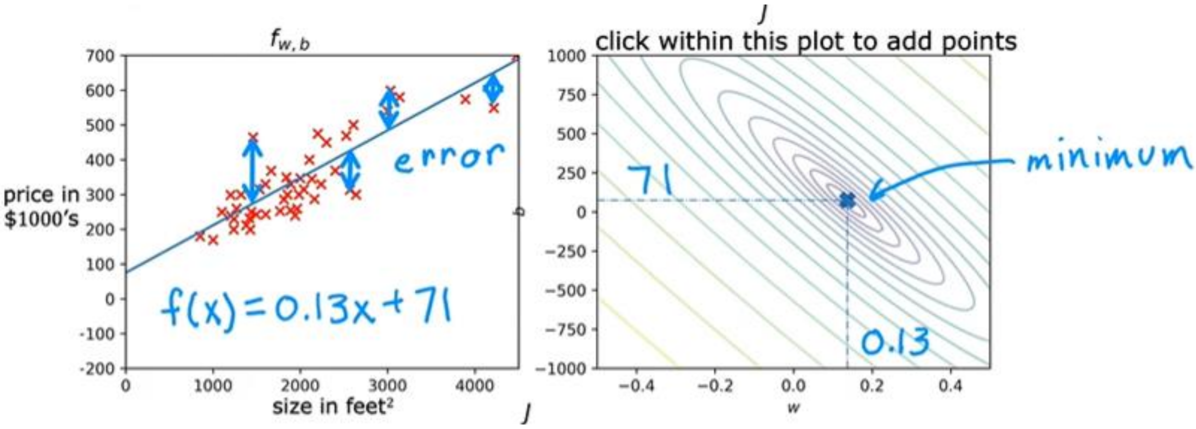
Cost Function

Error = Difference between prediction (ŷ) and target (y) i.e. (ŷ – y)

Formula for Cost Function / Squared Error Cost Function:
$J(w,b) = (1 / 2m) * \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$

Find values of w,b that make Cost Function small.



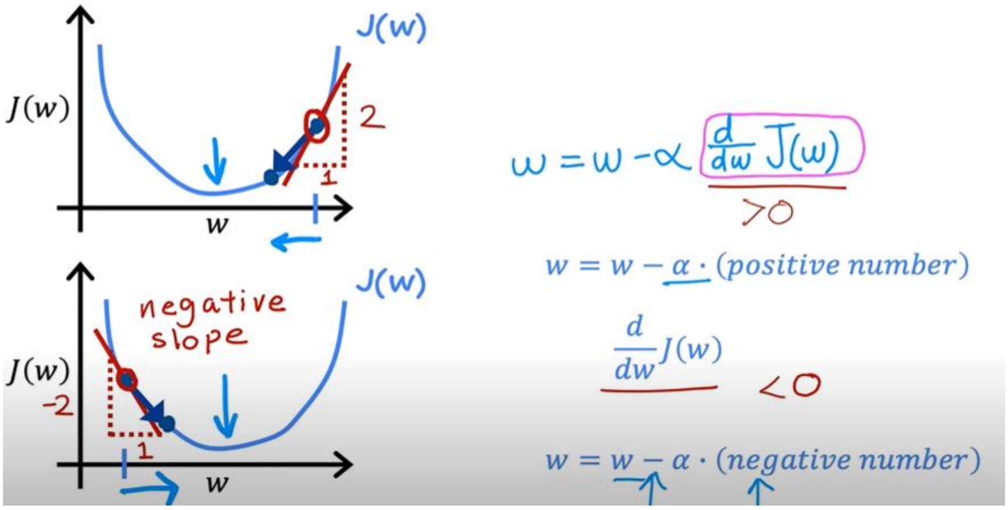


Gradient Descent – It is an algorithm that you can use to try to minimize any function

Goal: $\min_{w_1,\ldots,w_n,b} J(w_1, w_2, \ldots, w_n, b)$

Outline:
Start with some w,b (set w = 0, b = 0)
Keep changing w,b to reduce J(w,b)
Until we settle at or near a minimum

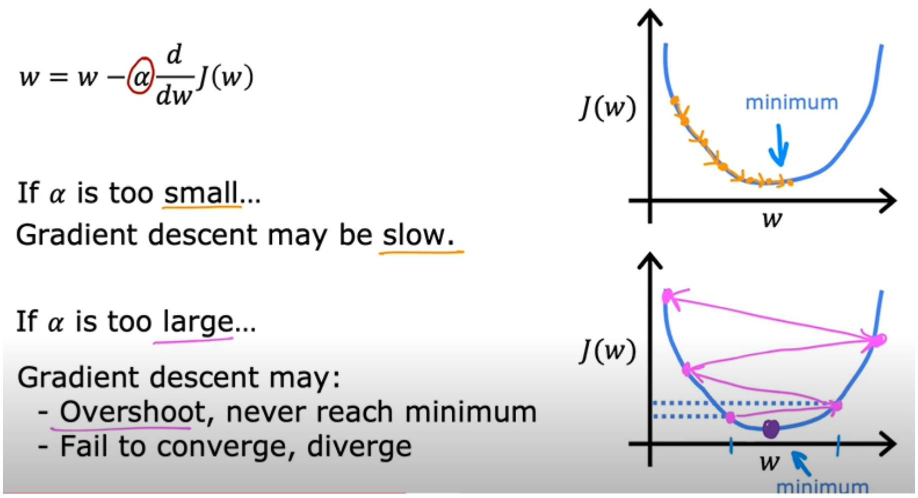
Repeat until convergence:
$w = w - \alpha (\partial / \partial w) J(w,b)$
$b = b - \alpha (\partial / \partial w) J(w,b)$



Learning rate (α)
Range: 0 – 1

Near a local minimum,
- Derivative becomes smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate



2. Multiple Linear Regression

It means linear regression with multiple features and one target variable.

Size in feet^2 X_1	Number of bedrooms X_2	Number of floors X_3	Age of home in years X_4	Price in $1000's X_5
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

X_j = jth feature
n = number of features
$\overline{x}^{(i)}$ = features of ith training example
$x_j^{(i)}$ = value of feature j in ith training example

$f_{w,b}(x) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$
$\overline{w} = [w_1\ w_2\ w_3 \ldots w_n]$
b is a number
$\overline{x} = [x_1\ x_2\ x_3 \ldots x_n]$ (Column Vector)

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

Gradient Descent

n features ($n \geq 2$)

repeat {

j=1

$$w_1 = w_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_1^{(i)}$$

$$\frac{\partial}{\partial w_1}J(\vec{w},b)$$

⋮

j=n

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

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

simultaneously update
w_j (for $j = 1, \cdots, n$) and b

}