Linear Regression

Presented by
FCAI-CU-AI-Community

One a day as a ML Engineer your Manager Asked you to work on a problem

Our Problem

Given some attributes, we will discuss them in next slide
We want to predict **Flight Ticket Price**.

ID	airline	flight	source_city	destination_city	arrival_time	departure_time	stops	class	duration	days_left	price
0	SpiceJet	SG-8709	Delhi	Mumbai	Night	Evening	zero	Economy	2.17	1	5953
1	SpiceJet	SG-8157	Delhi	Mumbai	Morning	Early_Morning	zero	Economy	2.33	1	5953
2	AirAsia	I5-764	Delhi	Mumbai	Early_Morning	Early_Morning	zero	Economy	2.17	1	5956

So, What Do you Think?
Is it suitable to solve it with traditional approach or use ML approach

ID	airline	flight	source_city	destination_city	arrival_time	departure_time	stops	class	duration	days_left	price
0	SpiceJet	SG-8709	Delhi	Mumbai	Night	Evening	zero	Economy	2.17	1	5953
1	SpiceJet	SG-8157	Delhi	Mumbai	Morning	Early_Morning	zero	Economy	2.33	1	5953
2	AirAsia	I5-764	Delhi	Mumbai	Early_Morning	Early_Morning	zero	Economy	2.17	1	5956

As you can see, when **arrival_time** and **duration** changed that didn't affect the **price** so using rules will not benfit us.

arrival_time	departure_time	stops	class	duration	days_left	price
Night	Evening	zero	Economy	2.17	1	5953
Morning	Early_Morning	zero	Economy	2.33	1	5953

ML Approach

Questions that need to be answered to select the appropriate algorithm:

1. Is problem **Supervised** or **Unsupervised**?
2. Is it **classification** or **Regression**?

It is a Supervised Problem

because our target exists in data which is **price**

price
5953
5953
5956

But What is the Difference between Classification and Regression?

	Classification	**Regression**
Target	Discrete values	Continuous Values
Example	Cat or Dog, Spam or Ham	House Prices, Score in Exam

It is a Regression Problem

Because price has continues values

price
5953
5953
5956

So, prepare for meeting our first regression model which is

Linear Regression

Remember straight line equation:

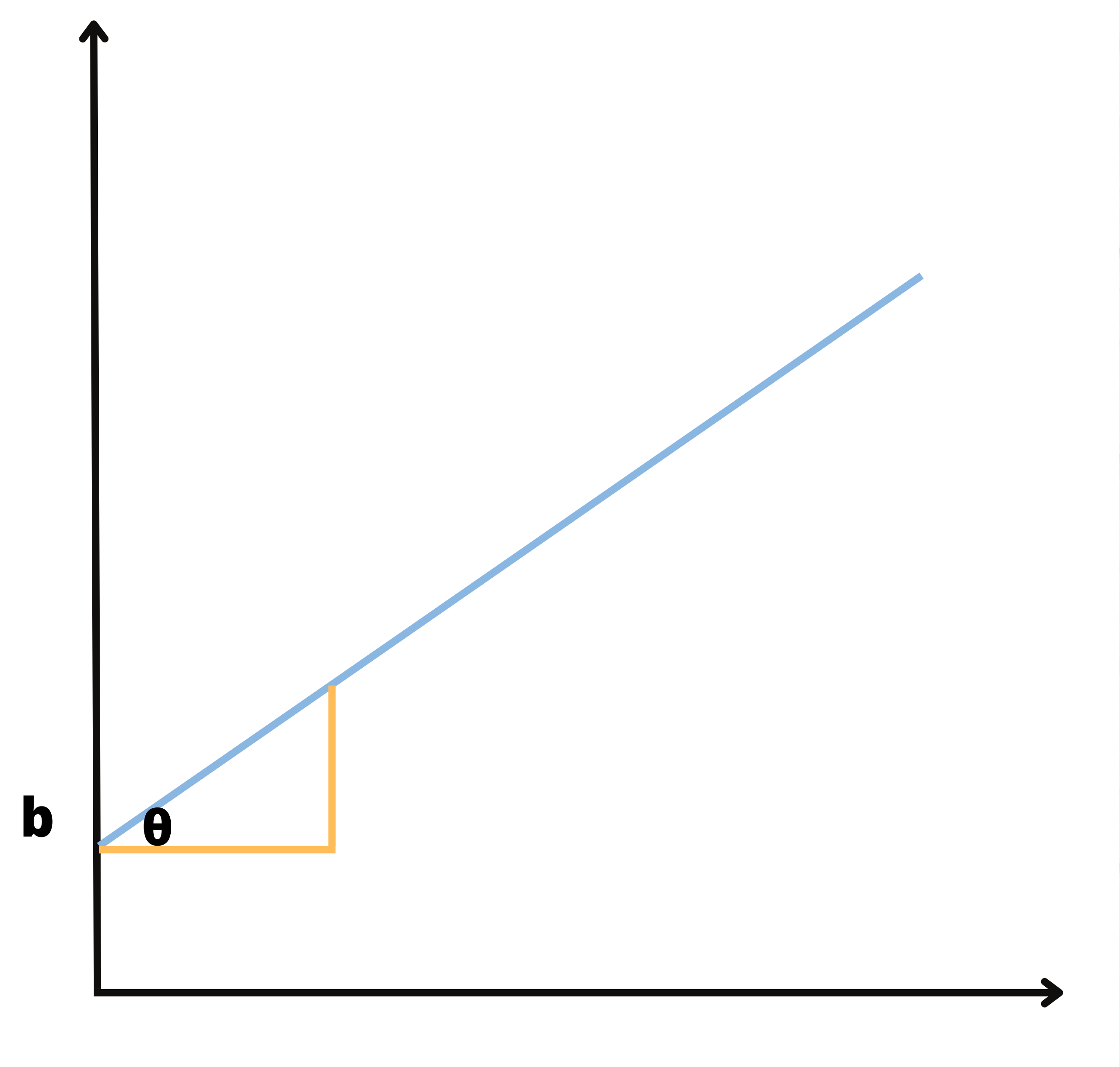
$$Y = W * X + b$$

Where:

$$W = Tan(\theta) = \frac{y}{x}$$

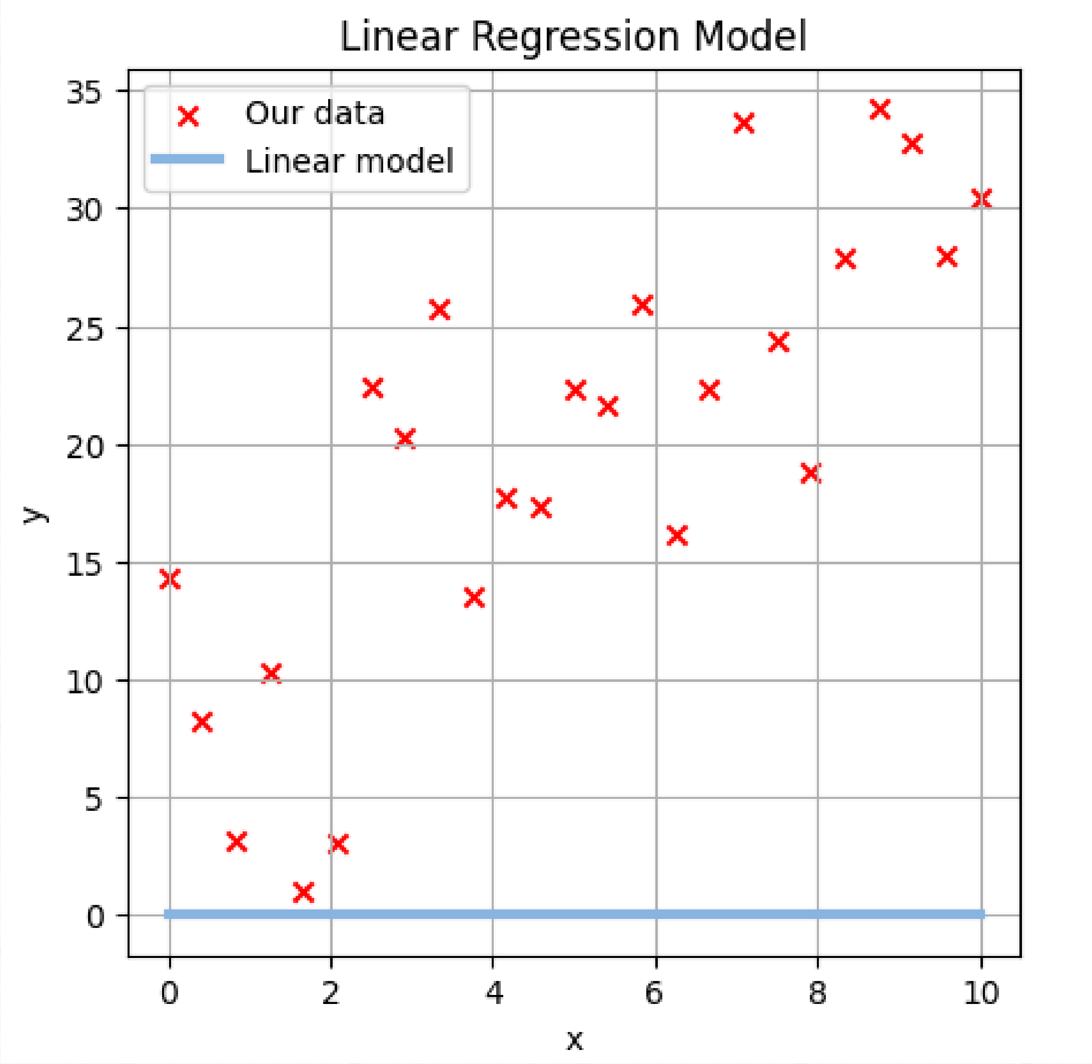
b: is intercepted part from y-axis

Inputs: X
outputs: Y
variables: W, b

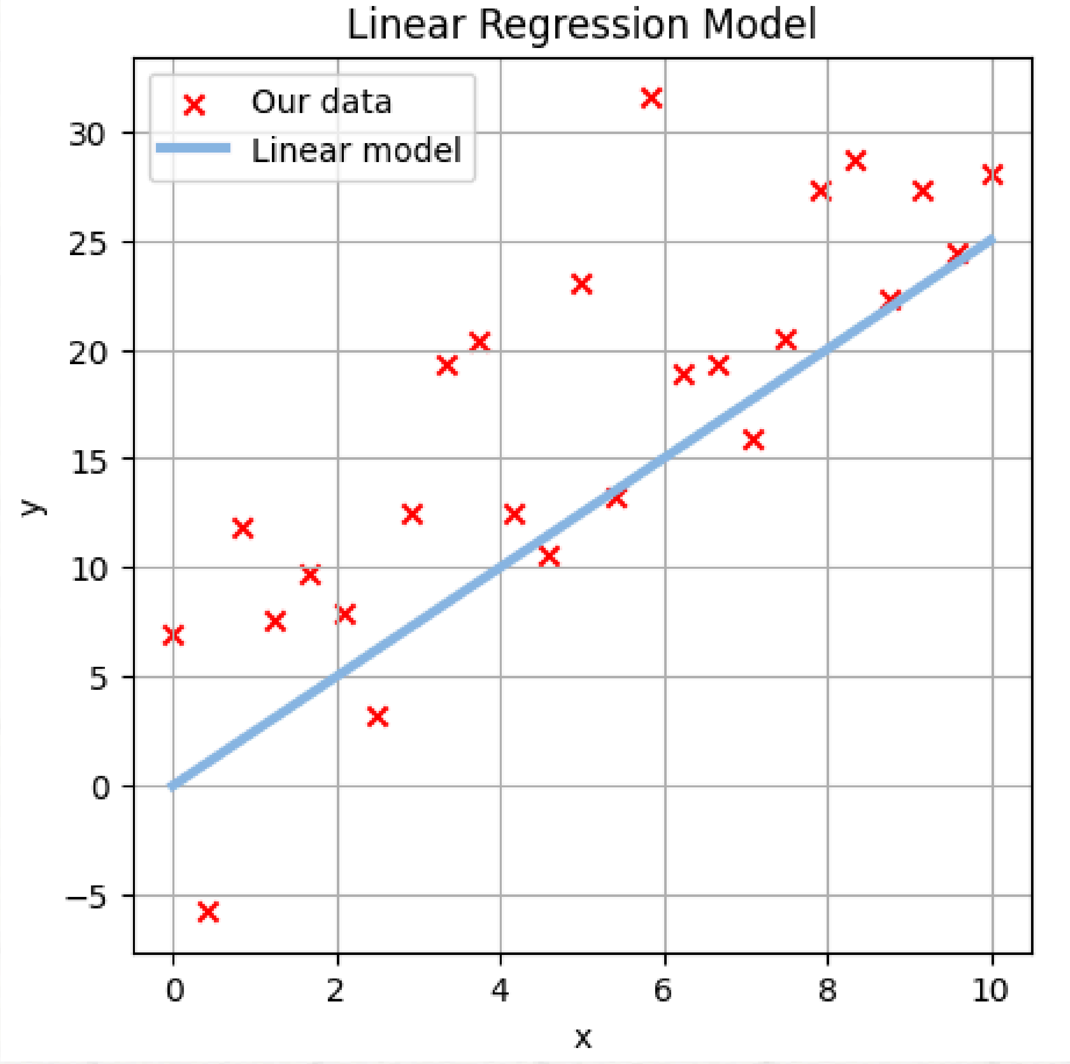


Linear Regression simply is just finding best line to fit our data

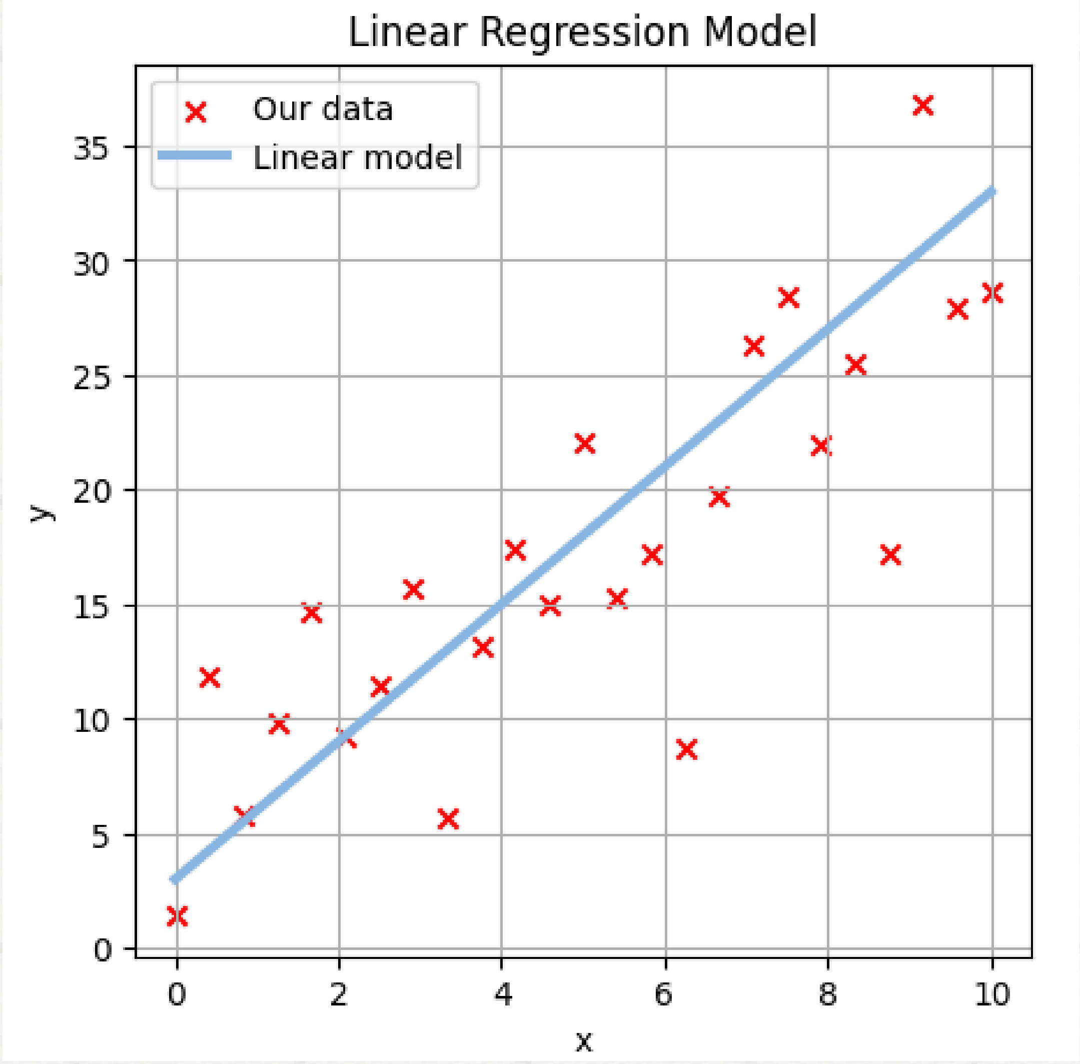
Which line is the best?



W = 0, b = 0



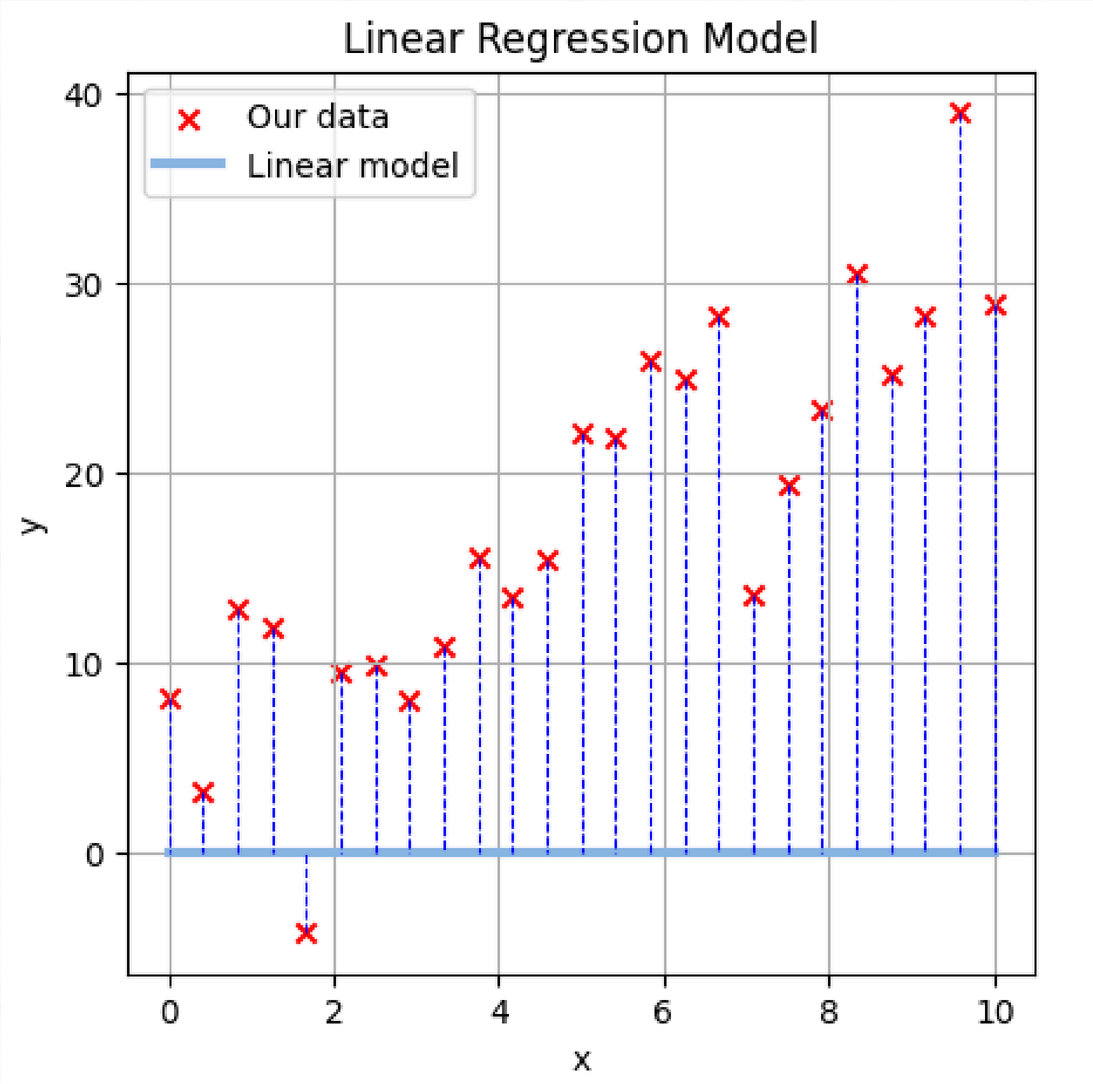
W = 2.5, b = 0



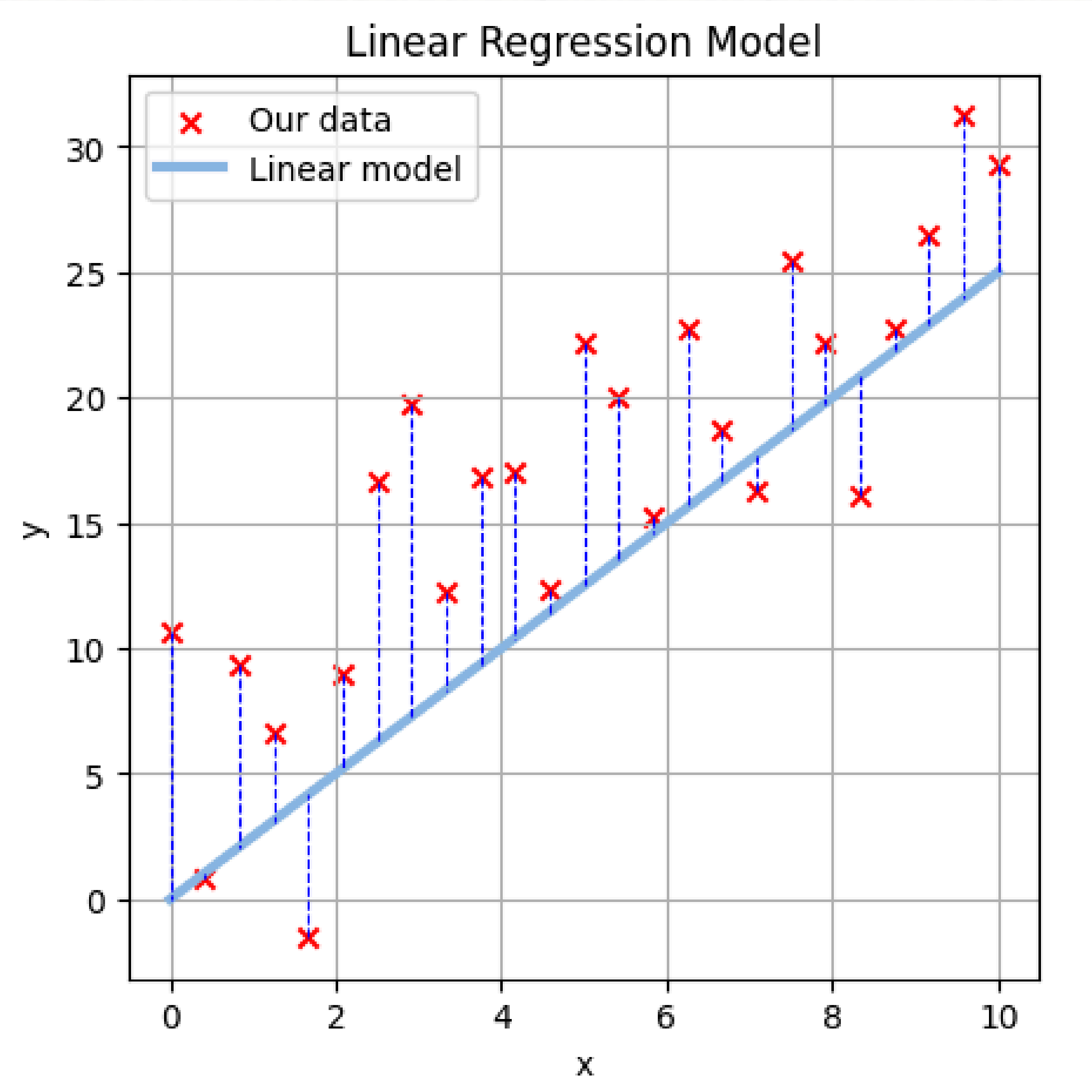
W = 3, b = 3

As humans, we can choose the best line, but how can a machine do that?

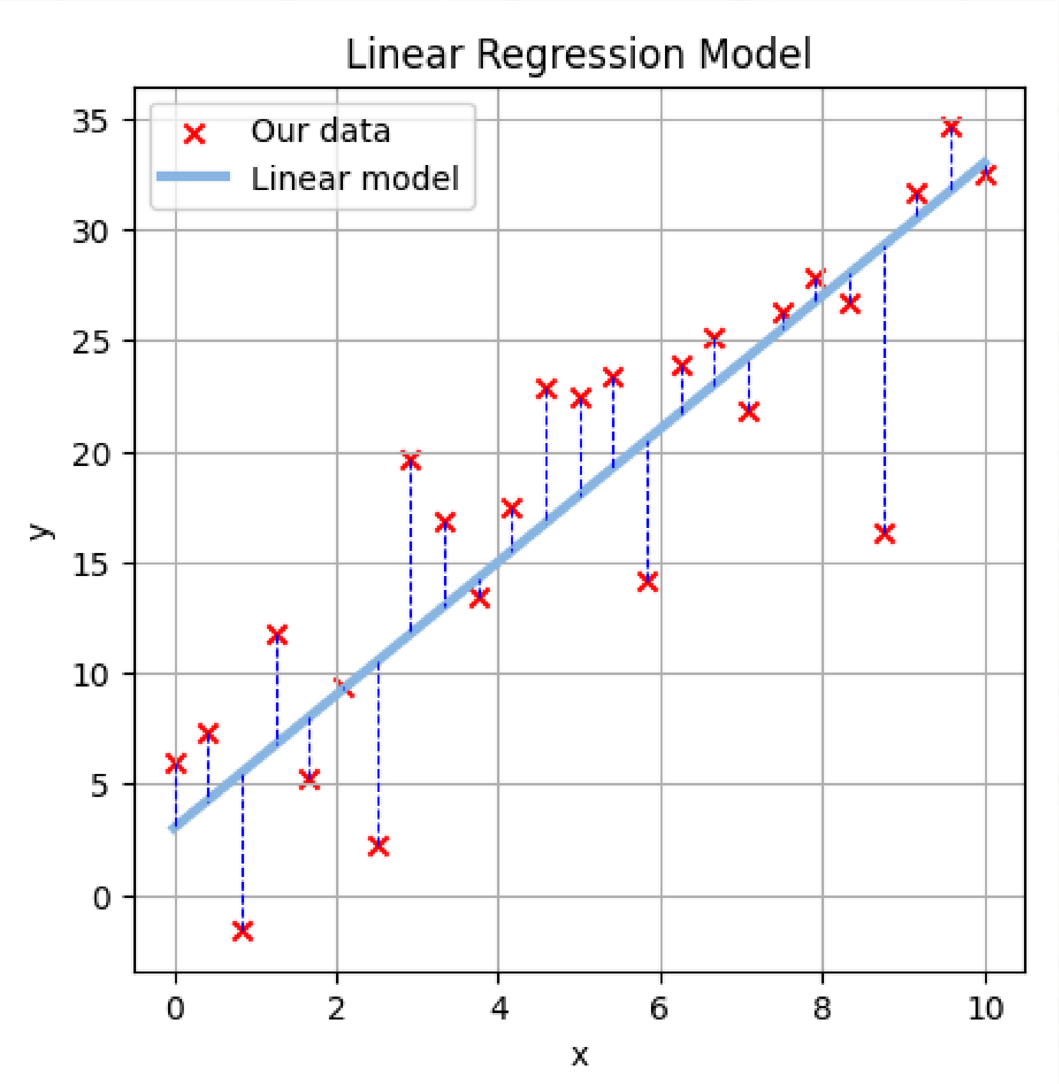
Is measuring distance between actual and predict can help us?



W = 0, b = 0



W = 2.5, b = 0



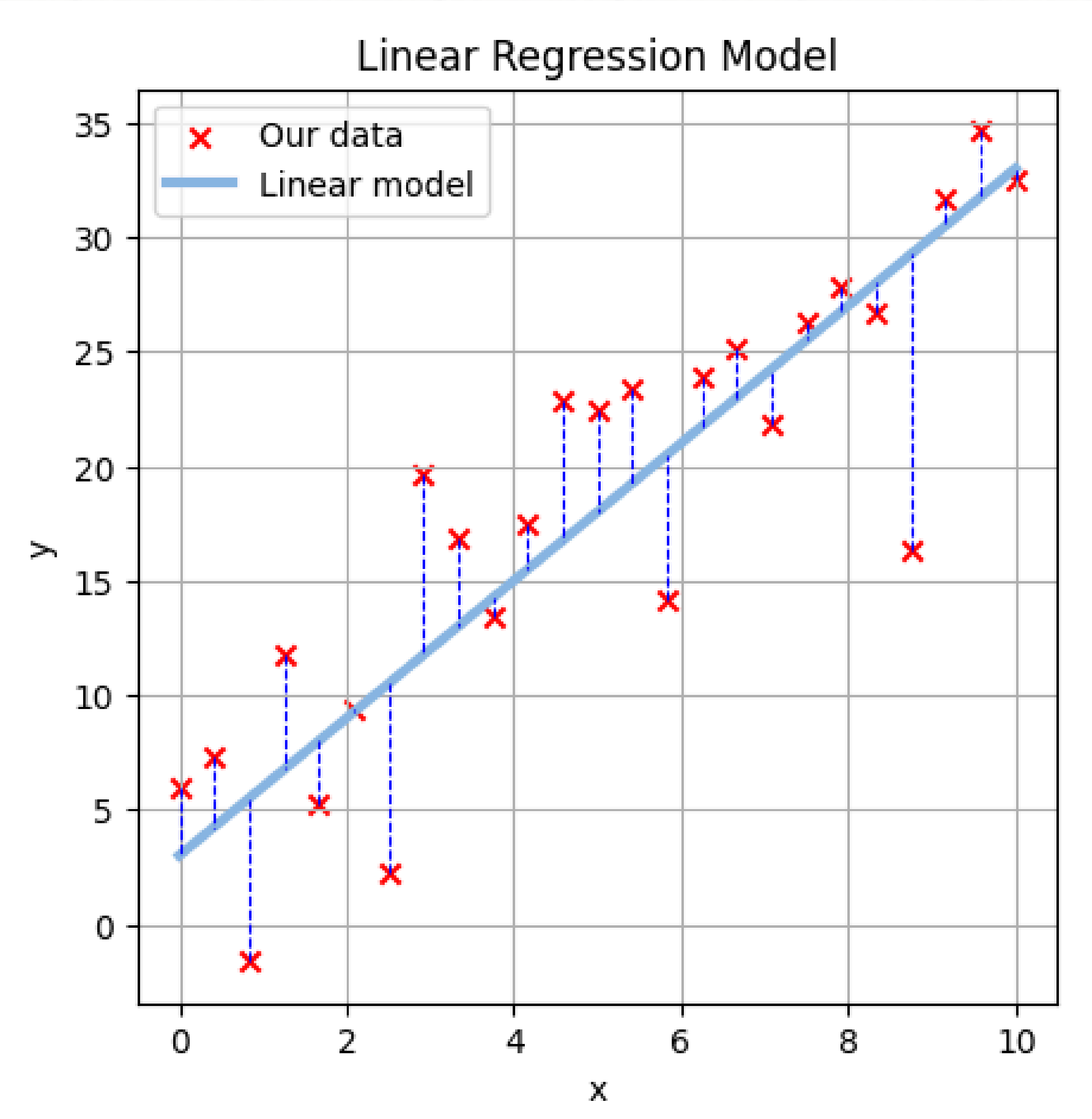
W = 3, b = 3

Lets define some symbols:

$$y : Actual\ data\ value$$

$$\hat{y} : Predicted\ data\ value$$

$$\hat{y} = W * X + b$$



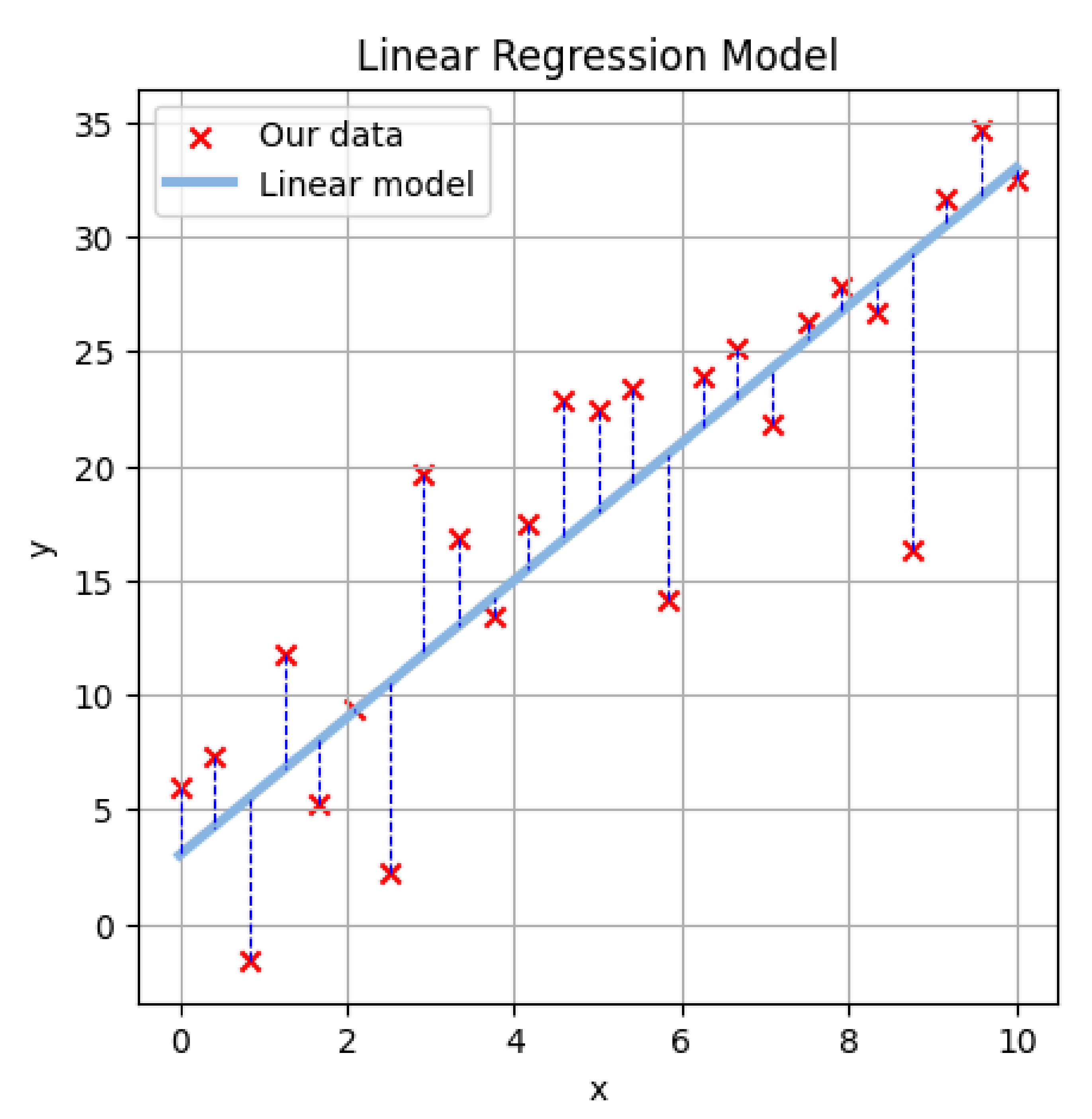
$$\hat{y} = W * X + b \quad y$$

$$distance = \hat{y} - y$$

$$distance = (\hat{y} - y)^2$$

$$Total\,distance = \sum_{i=0}^{m} (\hat{y}_i - y_i)^2$$

$$MSE = \frac{1}{2m} \sum_{i=0}^{m} (\hat{y}_i - y_i)^2$$



Now that we have calculated MSE what will we do with it?

Of course we need to minimize it but how?

So as we can see in J(W, b) if we want to minimize its value we need to change either W, b, or both

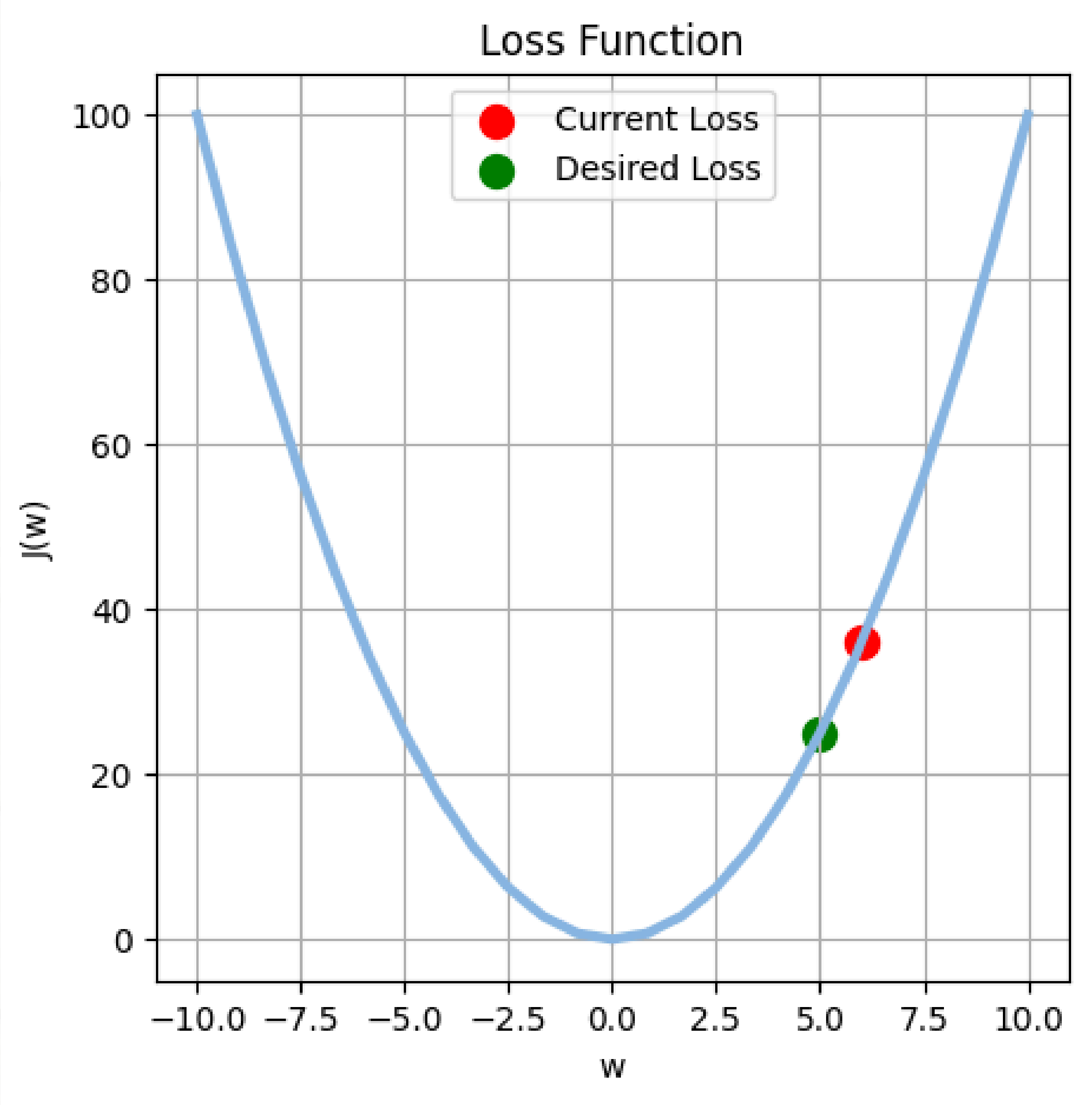
$$\hat{y} = W * X + b \qquad y$$

$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}(\hat{y}_i - y_i)^2$$

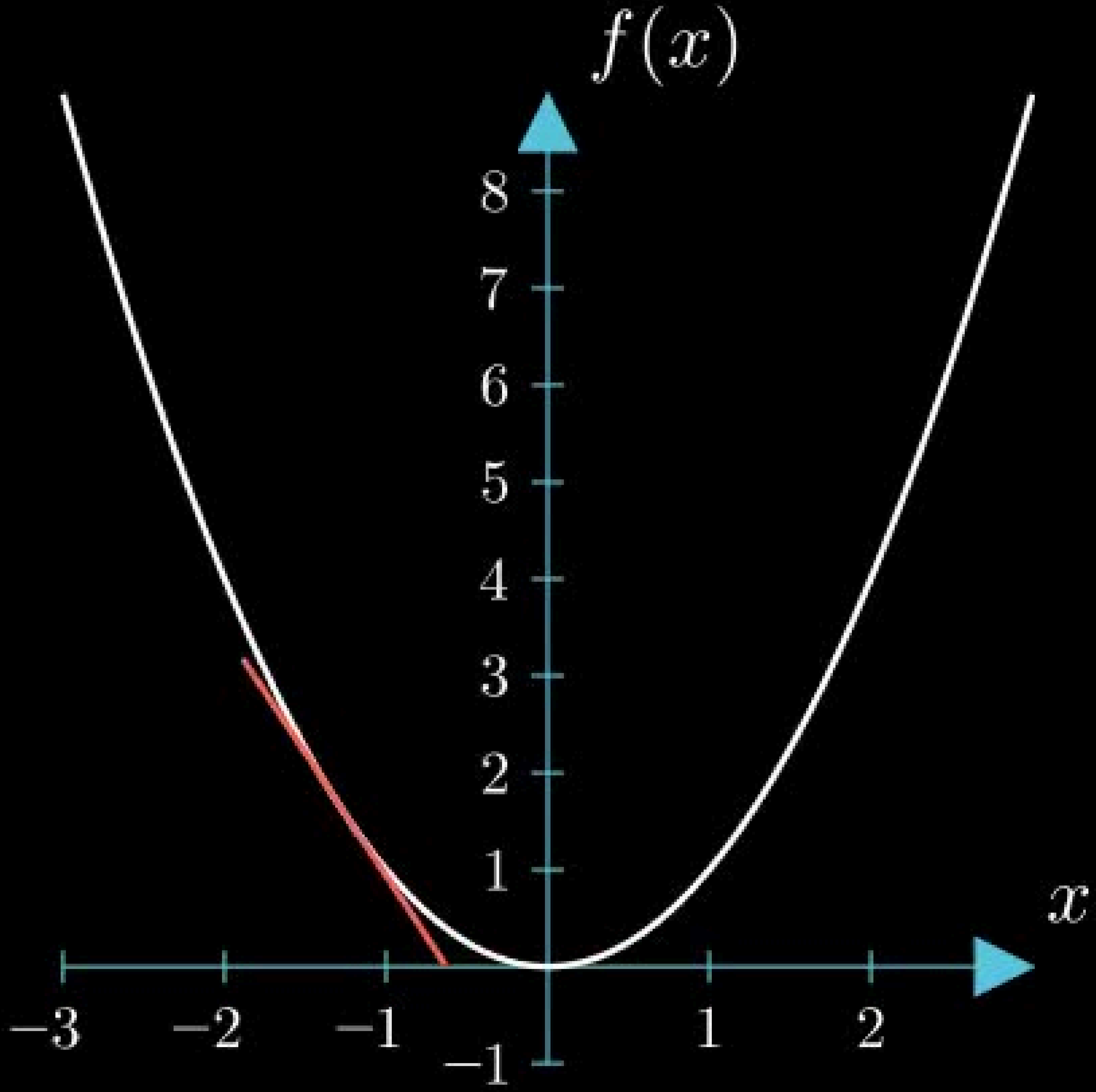
$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}((w * x_i + b) - y_i)^2$$

Lets assume our Loss function: J(W, b)
Has only one input which is W and b = 0

$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}((w * x_i) - y_i)^2$$



How we can move from the red dot to the green dot?



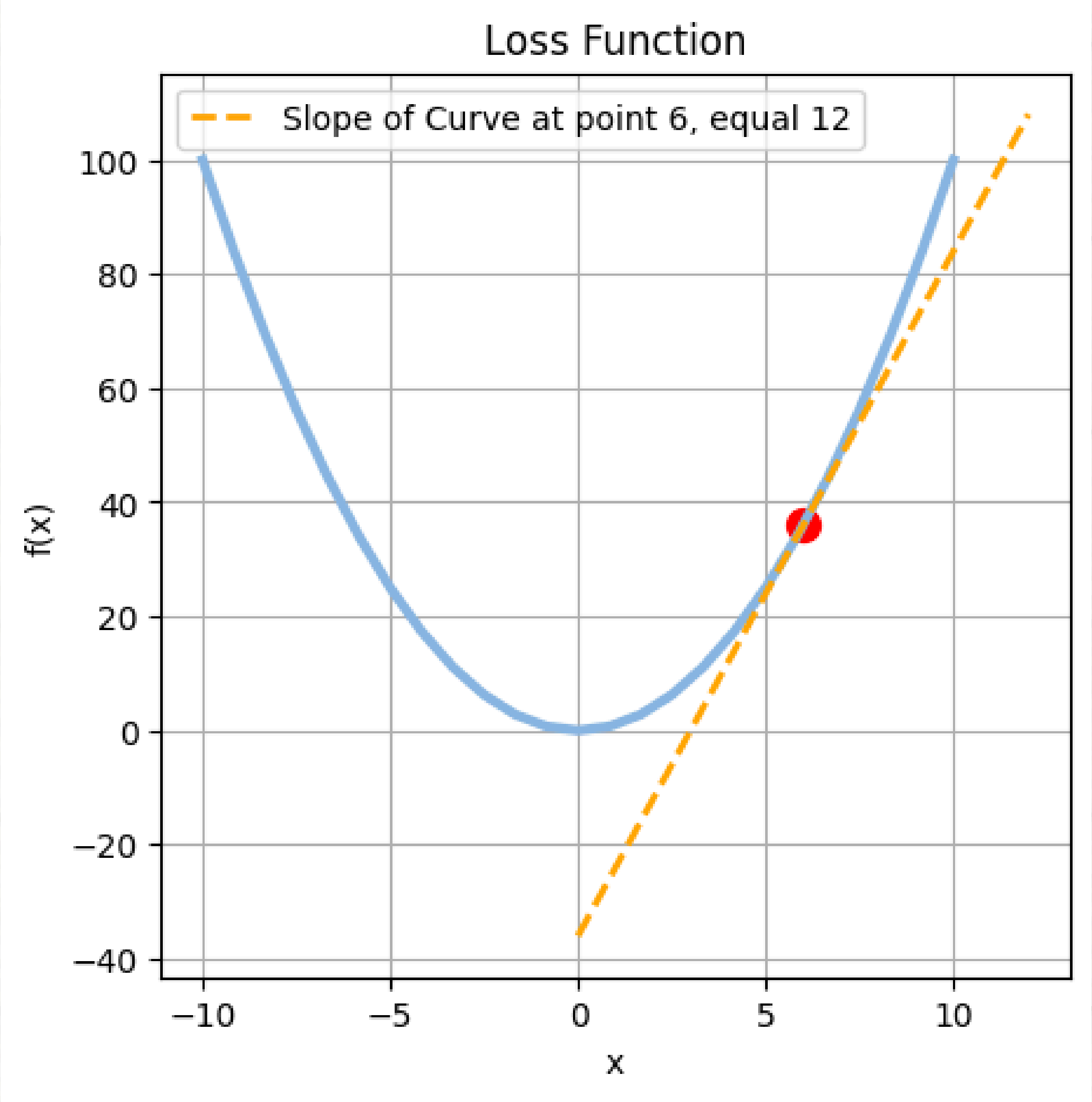
x: -0.80
Slope: -1.60

We can use the slope of the loss function to update the weights

But actually, what is slope:

- Slope have different interpretations but one of its interpretations is that it is the derivative of function.
- Another interpretation is that it is tan(θ) , θ is angle made with x-axis

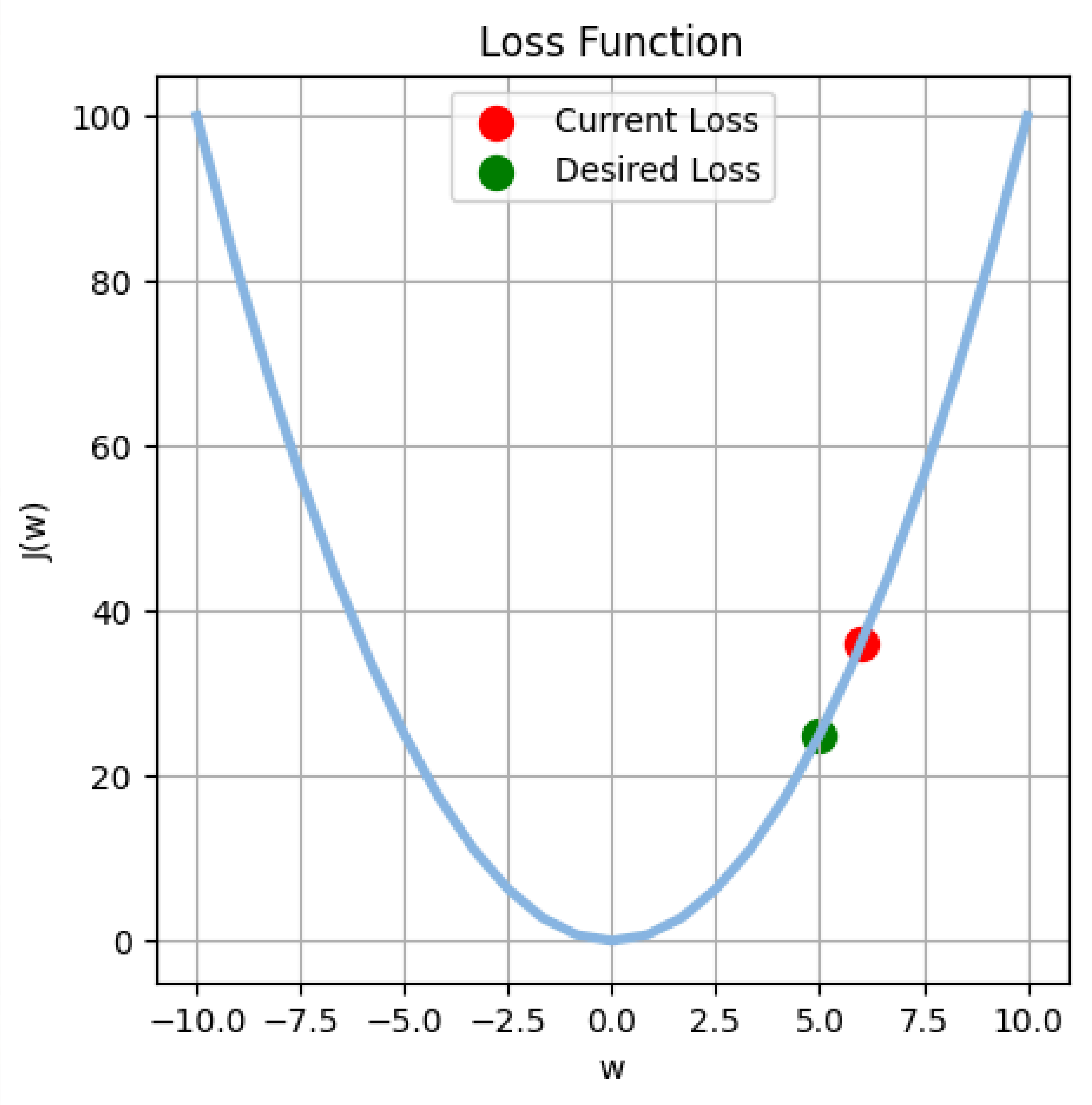
$$f(x) = x^2, \frac{df}{dx} = 2x$$



$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}((w * x_i) - y_i)^2$$

$$\frac{dJ}{dw} = \frac{2}{2m}\sum_{i=0}^{m}((w * x_i) - y_i) * x_i$$

$$\frac{dJ}{dw} = \frac{1}{m}\sum_{i=0}^{m}((w * x_i) - y_i) * x_i$$

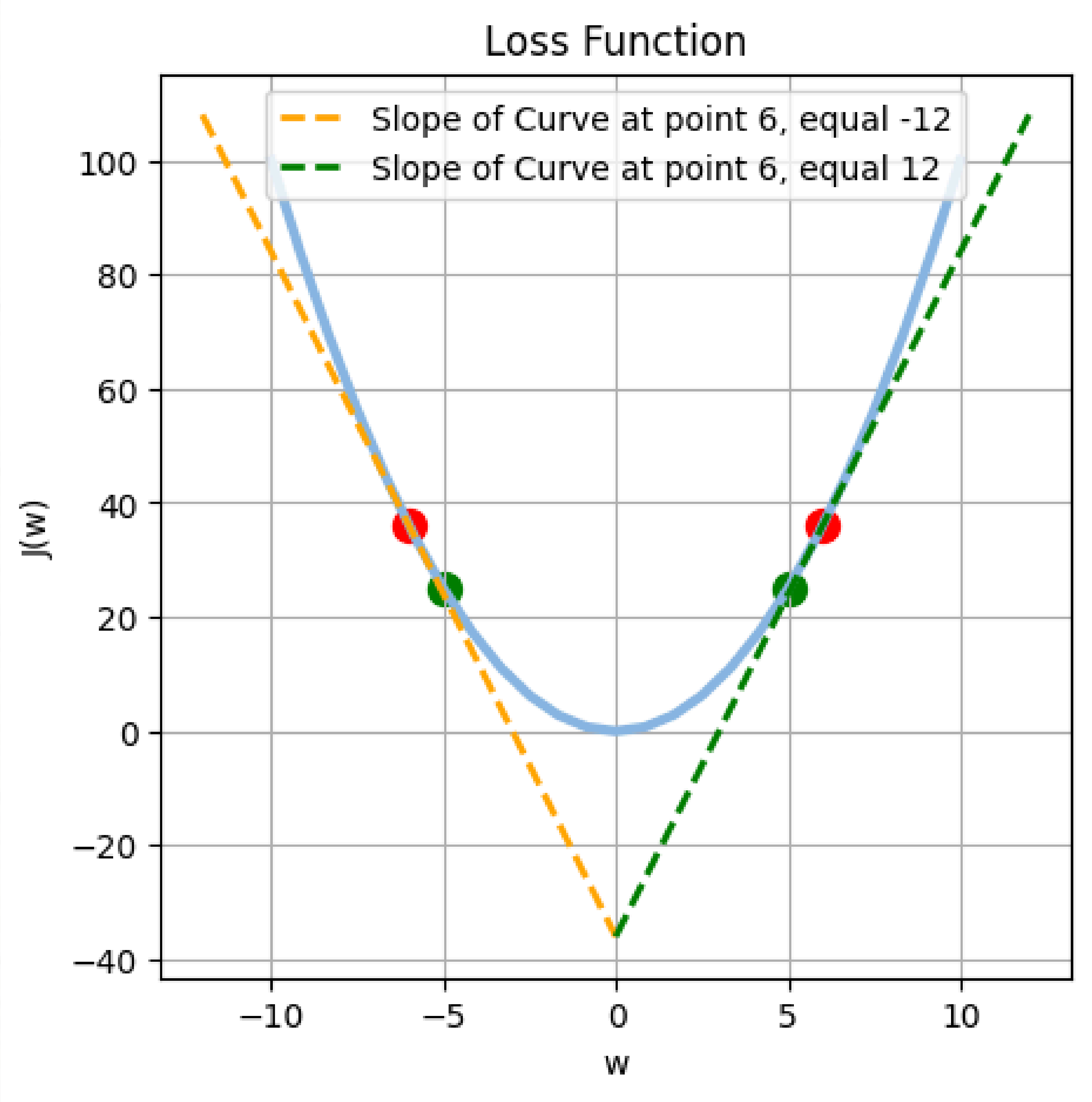


How can we use this $\frac{dJ}{dw} = \frac{1}{m}\sum_{i=0}^{m}((w * x_i) - y_i) * x_i$

To Update W

$$\frac{dJ}{dw} = \frac{1}{m}\sum_{i=0}^{m}((w * x_i) - y_i) * x_i$$

$$w_{new} = w_{old} - \alpha * \frac{dJ}{dw}$$



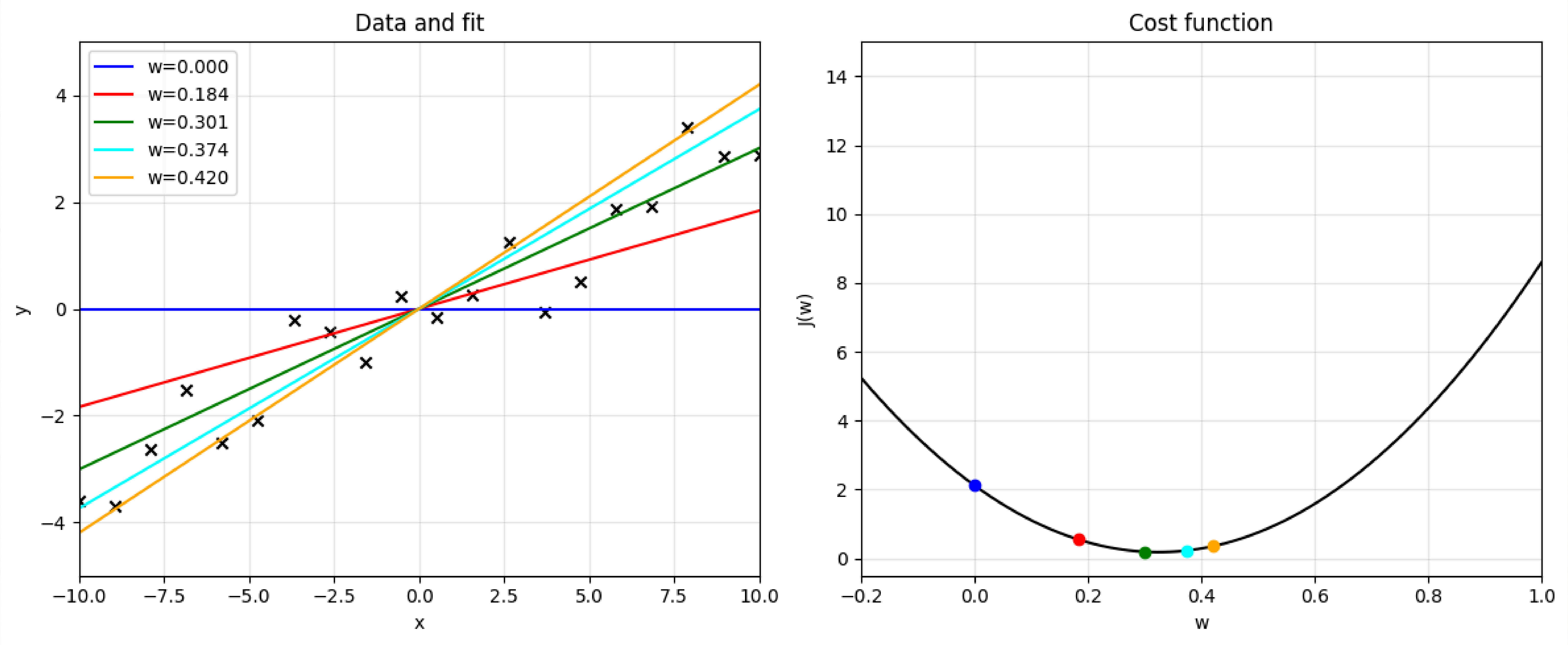
Repeat those three steps until convergence or reaching max iterations

$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}((w * x_i) - y_i)^2$$

$$\frac{dJ}{dw} = \frac{1}{m}\sum_{i=0}^{m}((w * x_i) - y_i) * x_i$$

$$w_{new} = w_{old} - \alpha * \frac{dJ}{dw}$$

How slope of Line change by changing w

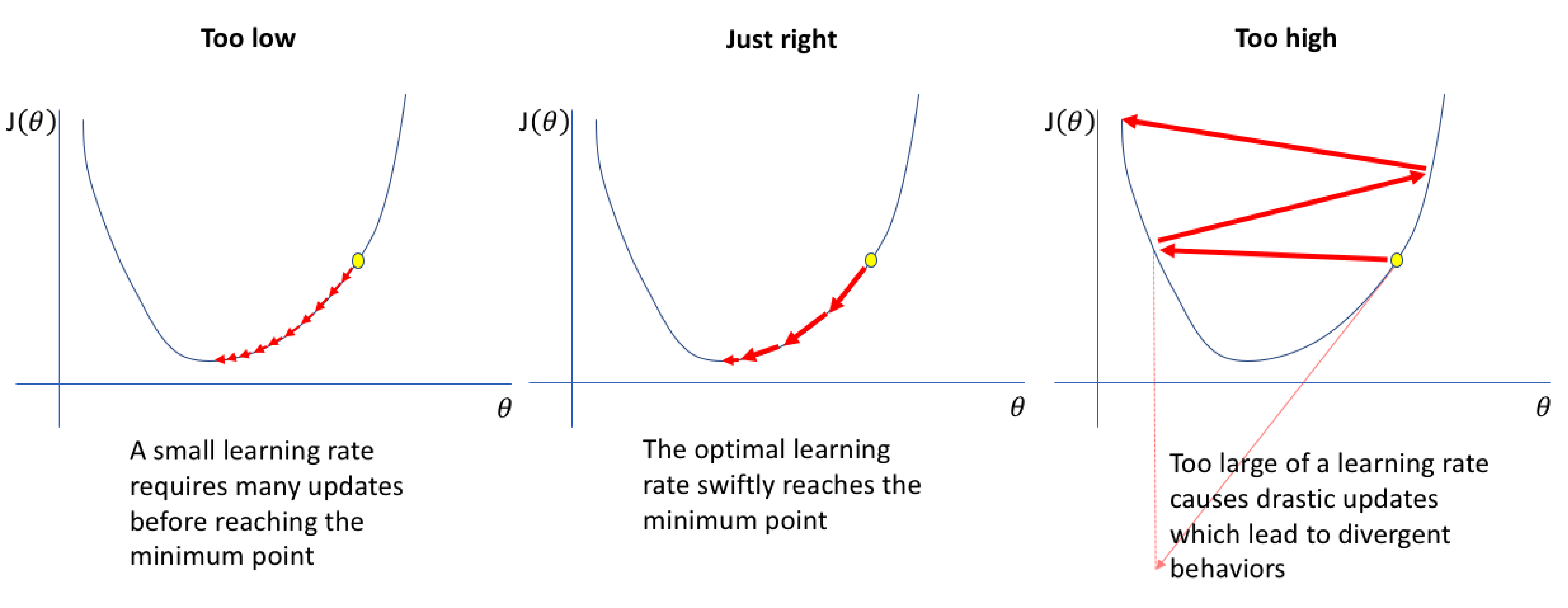


So, we made what we wanted but, what is α?

α is Called learning rate

- It is first hyperparameter you meet.
- It is used to control step of gradient towards global minima.
- usually ranges between 0.1 to 0.0001

Learning rate values effect:



Lets sum up the whole idea

Gradient descent steps:

1- Start with random w and b

2- compute cost $J(w,b) = \frac{1}{2m}\sum_{i=0}^{m}((w*x_i+b)-y_i)^2$

3- compute derivative of cost with respect to w and b

3.1- $\frac{dJ}{dw} = \frac{1}{m}\sum_{i=0}^{m}((w*x_i+b)-y_i)*x_i$

3.2- $\frac{dJ}{db} = \frac{1}{m}\sum_{i=0}^{m}((w*x_i+b)-y_i)$

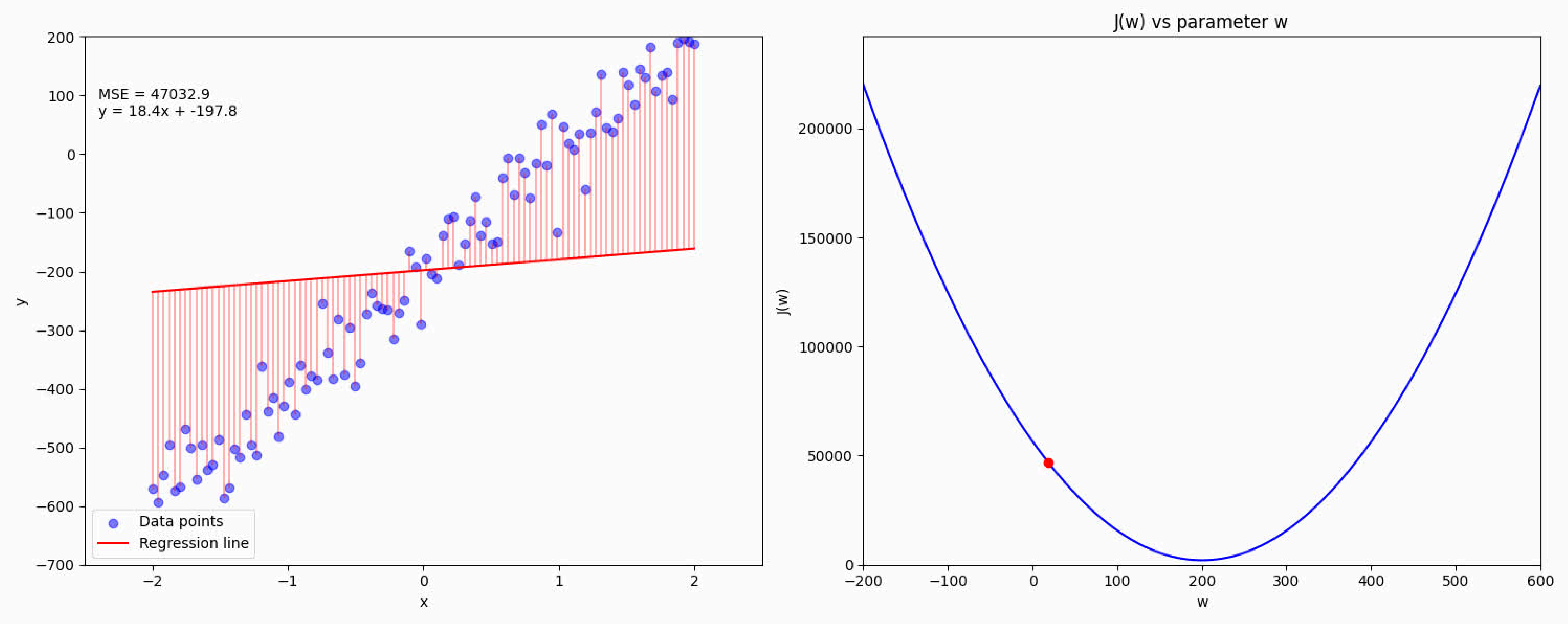
4- use this derivative to update weight and bias

4.1- $w_{new} = w_{old} - \alpha\frac{dJ}{dw}$

4.2- $b_{new} = b_{old} - \alpha\frac{dJ}{db}$

5- repeat 2,3,4 until converge or for some number of iterations

Gradient descent in action



Thank you very much!