

MACHINE LEARNING
Linear Regression
Week 1

1. What is Machine Learning,

In this section we explain about machine learning, specially definition of machine learning, we can get two definition the first from Arthur Samuel

- **Arthur Samuel** (1959), Machine Learning is Field of study that gives computers the ability to learn without being explicitly programmed.

- **Tom Mitchell** (1998) Well-posed learning problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

For example we have program to detect email spam or not spam.

- watching you label emails as spam or not spam (E)
- classifying emails as spam or not spam (T)
- the number (or fraction) of email correctly classified as spam/not spam.

2. Kind of Machine Learning Algorithms:

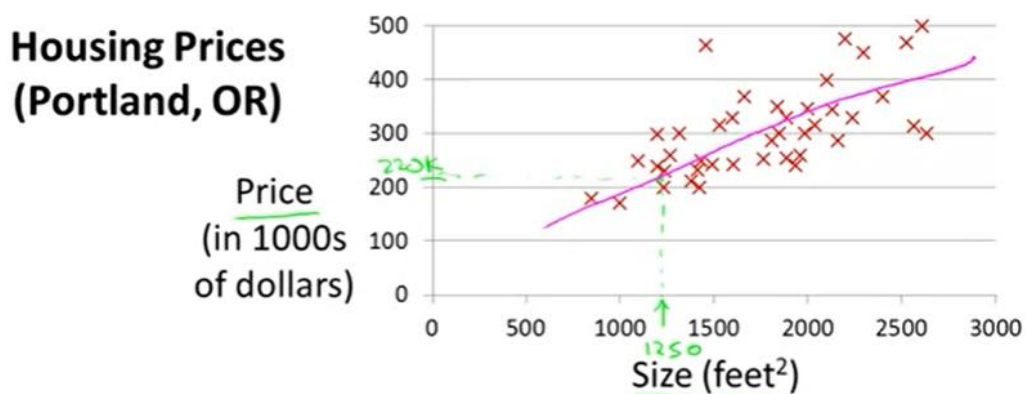
- Supervised Learning
- Unsupervised Learning

Other : Reinforcement learning, recommender systems.

3. Model and Cost Function

A. Model Representation

The model representation is how we can predict some data (y) from another data (x).



this above is example of data, we can try to **predict Price** by **Size of house** in Housing Prices data.

The first we given **several data that representation by point** with color red, **the model is line** that have value approach every data.

B. Cost Function

Cost Function is function can calculate how close is the model with actual data.
Before that we can define function

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

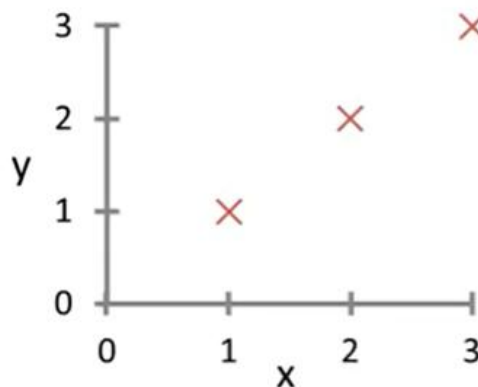
Goal: minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

- **Hypothesis** : model that can predict the data
- **Parameters** : parameters of model, this parameter determine how to close model with actually data
- **Cost Function** : Function that **measure closeness** of model with actual data
- **Goal** : to have good model we must have model that very closely with actual data , so we must know value parameter that have minimum cost function.

C. Example

Example 1 :

- we have data

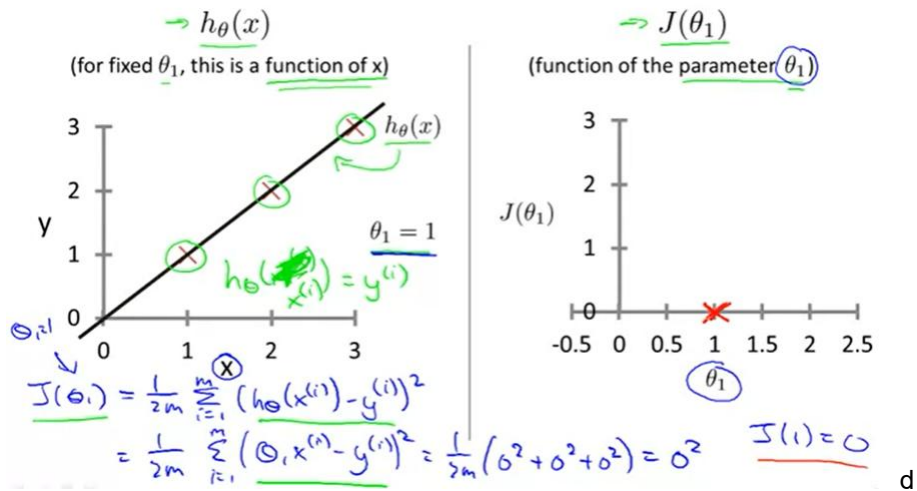


- we have hipotesys function , y =

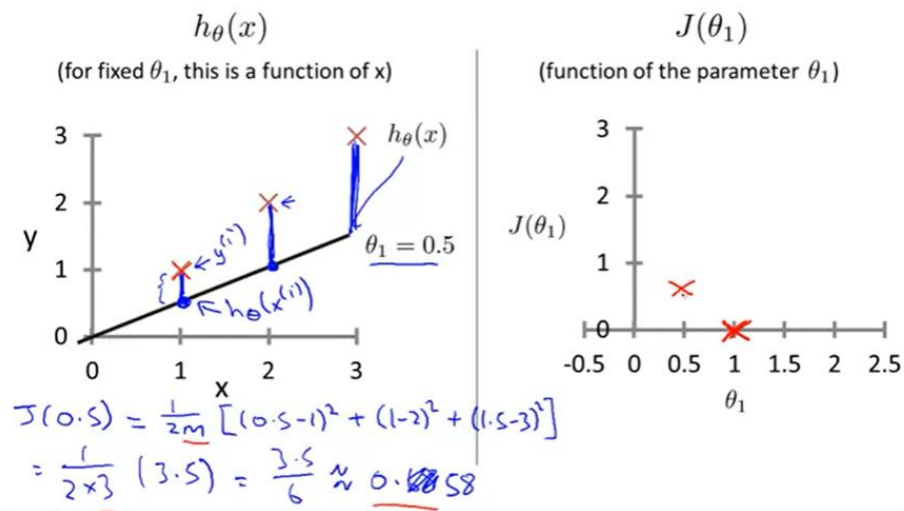
$$h_{\theta}(x) = \theta_1 x$$

- so with can try each of **value parameter** to get minimum cost function. The first we try value parameter 0.5 and 1.

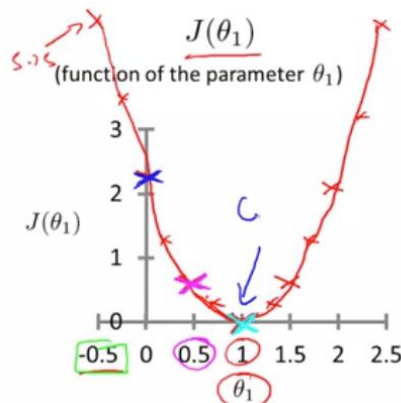
Theta = 1



Theta = 0.5



So if we try to calculate possibility parameter we can the **grafik of cost function** can be like:

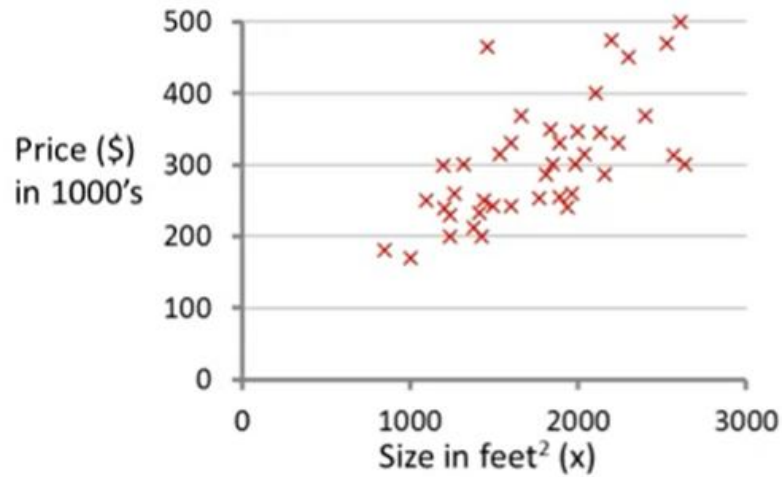


So the goal is find Minimize of Cost Function, in this case the Minimum of Cost function when parameter theta equal 1, which is have Cost Function equal 0.

So the conclusion we can use parameter theta equal 1 for predict the data model.

Example 2

Try to use 2 value parameter

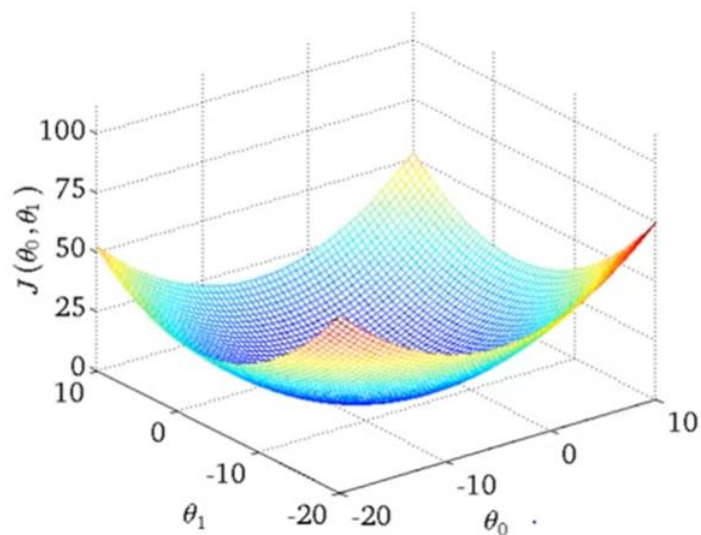


So for predict this data, we can try to use 2 value parameter, with 2 parameter the hypothesis parameter can be:

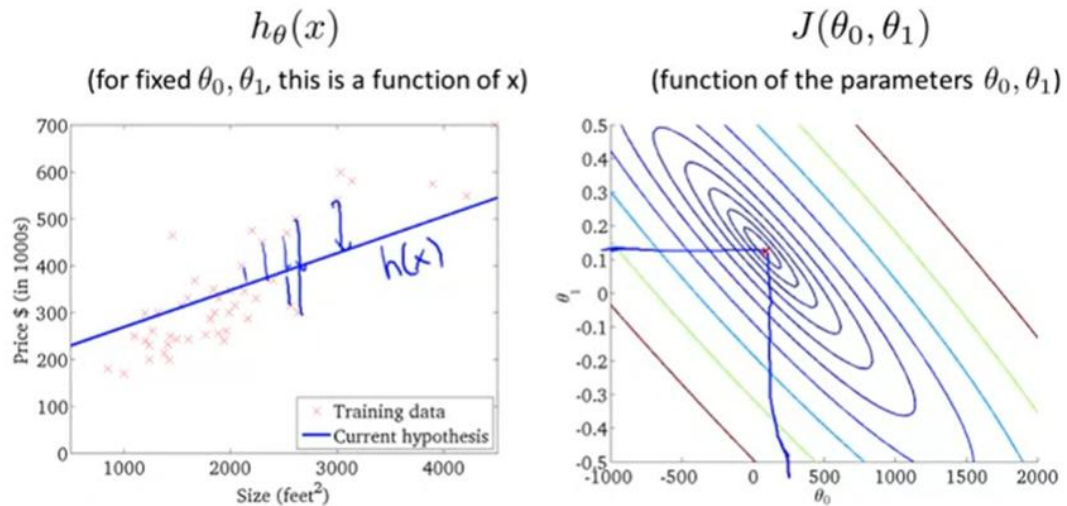
$$\text{Hypothesis: } h_{\theta}(x) = \theta_0 + \theta_1 x$$

Theta 0 and theta 1

So to find Min Cost we must calculate 2 parameter.
So we can have **Cost Function** Grafik in 3 Dimation



We don't use 3 dimation grafik for analytic but we can ty to understand how to user contour plot, the contour is transformation of 3 dimation grafik above



To read contour plot, contour plot have same value of cost function every same line. The minimize of cost function is at center of contour plot, but carefully the center of contour plot can represent of minimum value or maximum value, before determine we must know it.

4. Gradient Descent

Gradient Descent is Algorithm for minimizing the cost function. Gradient Descent not only used in linear regression but used all over the place in machine learning.

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

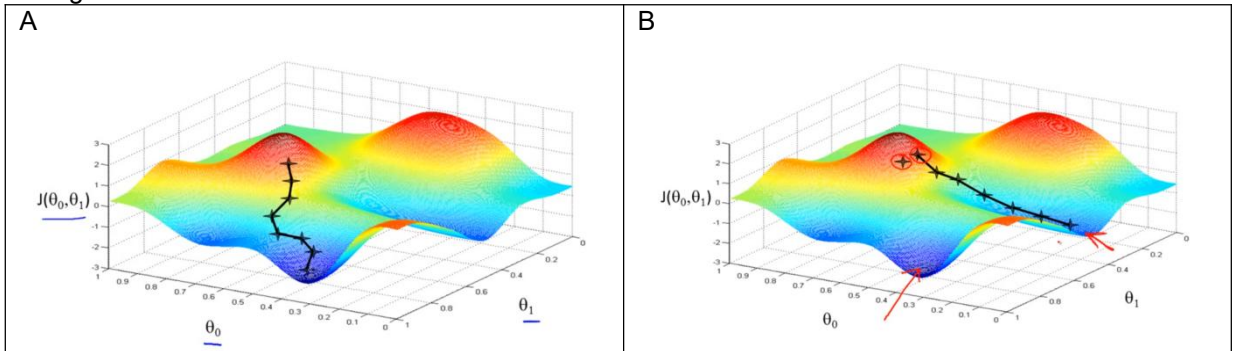
- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

until we hopefully end up at a minimum

Gradient descent is an algorithm for solving this more general problem.

Idea for gradient descent. We can decide the initial guesses for θ_0 and θ_1 . don't really matter what they are, but commonly we choose θ_0 to 0 and θ_1 to 0. And after that gradient descent can keep changing the θ_0 and the θ_1 a little bit to try reduce the function J , until hopefully we find minimum or minimum local.

Intuition how gradient descent work and how different choice **intialisation value** can change the result or the local mininum .



In the picture above, we can see the point move from some place to local minimum (whole with blue color surface) , the gradient descent work some think like this they are find the minimal local in step by step.

Two of picture above, explain about how the different initialisation value can change result of local mininum.

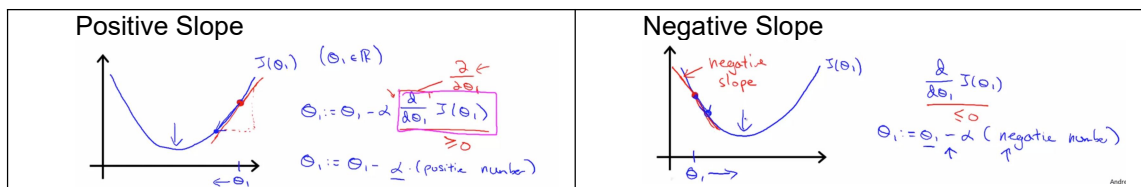
After we know intuition lats look at the math.

Gradient descent algorithm

repeat until convergence {
 $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ (for $j = 0$ and $j = 1$)
}

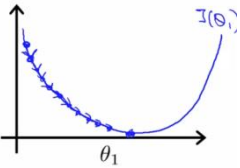
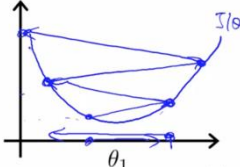
$:=$	Assignment $\alpha := \frac{b}{a}$	Assignment Operator , its mean take value b and use it overwrite whatever value in a
α		Learning Rate , (alpha) Learning Rate does is it basically controls how big a step we take. So if alpha is very large can make very aggressive gradient descent and take huge step downhill. If alpha very small can make taking little, and little baby step downhill.
$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$		Derivative Term , The Derivative can give direction to minimum value.

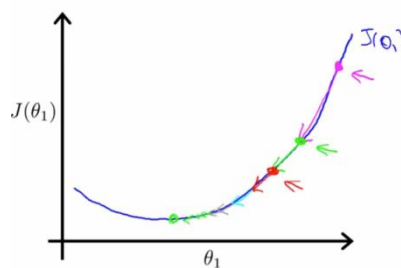
Derivative



In above, explain intuition of derivative. The Deravative value is slope of grafik. So the value always to be downhill and go to local mininum.

Learning Rate

Very Small		Gradient Descent can very long to find local minimum because to many step but a little. That can take a time to compute.
Very Big		Gradient Descent can overshoot, it may be failed to find local minimum or even divergen



Actually for gradient descent will automatically take smaller step. The first maybe can get big step but if was close with local minimul step can be more smaller.

after that, we can explain how to do correct calculation, the update theta0 and theth1, must be simultaneuos update, that mean we can update in same time.

Correct: Simultaneous update

```
temp0 := theta_0 - alpha * d/dtheta_0 J(theta_0, theta_1)
temp1 := theta_1 - alpha * d/dtheta_1 J(theta_0, theta_1)
theta_0 := temp0
theta_1 := temp1
```

Incorrect:

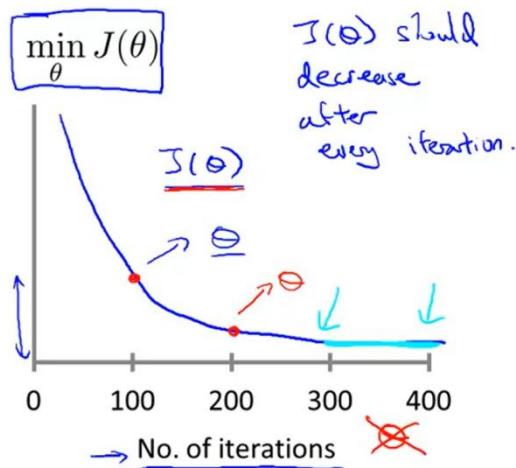
```
temp0 := theta_0 - alpha * d/dtheta_0 J(theta_0, theta_1)
theta_0 := temp0
temp1 := theta_1 - alpha * d/dtheta_1 J(theta_0, theta_1)
theta_1 := temp1
```

In the left side we have correct away, but in right side we have incorrect away because Theta0 was replace from and affect to calculation temp1, it will be wrong.

5. Choice The Great Learning Rate

A. Debugging (How to make sure gradient descent is working correctly)

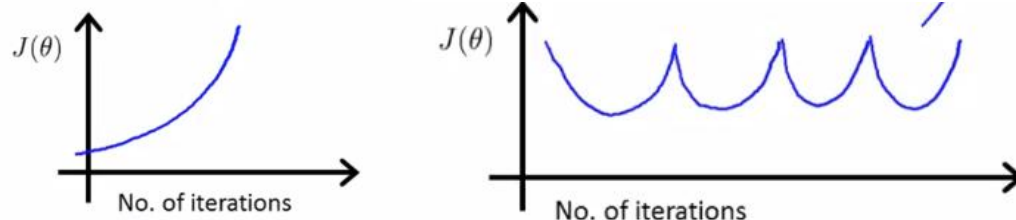
Actually for know your gradient descent is working correctly you can visualization your



If you look at this graph, you can see the data was convergent in 300 No of Iteration to 400 Iterations. If you find something like this your graph is correct.

Actually you can know your iterations were convergent or not use **automatic convergent test**, the test can compare between values of decreases, if the value changing was very small such as 0.0001 (epsilon) that means the data was convergent

But if you find something like this



Your gradient descent not work correctly, you must use smaller alpha (learning rate)

B. How to Choose Learning Rate Alpha

To Choose your Learning Rate Alpha actually you must try every possibility for example you choose value learning rate

0.001 , 0.003 , 0.01, 0.03, 0.1, 0.3, 1

Note:

Sake : demi

Brevity : concise and exact use of words in writing or speech : keringkasan

Concise : give a lot of information clearly and in a few words; brief but comprehensive.

6. Manual Calculation

θ_j	Parameter ,Nilai yang di optimasi)
$:=$	Assignment Operator , Operator menggantikan value , ($a := b$, ini bermaksud b menggantikan value a)
α	Learning Rate ,Mengontrol seberapa besar step yang dilakukan.
$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$	Nilai Turunan / Derivative Term , yang memberi arah menuju minimum/optimal

$:=$	Assignment $\alpha := b$	Assignment Operator , its mean take value b and use it overwrite whatever value in a
α		Learning Rate , (alpha) Learning Rate does is it basically controls how big a step we take. So if alpha is very large can make very aggressive gradient descent and take huge step downhill. If alpha very small can make taking little, and little baby step downhill.
$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$		Derivative Term , The Derivative can give direction to minimum value.

Noted

Calculation Linear Regression

$m = 3$

$Y = [300, 500, 800]$

$X = [1, 2, 4]$

CALCULATION COST FUNCTION

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

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

$$J(180, 100) = \frac{1}{2m} \sum_{i=0}^{m-1} ((180x^{(i)} + 100) - y^{(i)})^2$$

$$J(180, 100) = \frac{1}{2(3)} [((180(1) + 100) - 300)^2 + ((180(2) + 100) - 500)^2 + ((180(4) + 100) - 800)^2]$$

$$J(180, 100) = \frac{1}{2(3)} [(280 - 300)^2 + (460 - 500)^2 + (820 - 800)^2]$$

$$J(180, 100) = \frac{1}{2(3)} [400 + 1600 + 300]$$

$$J(180, 100) = \frac{1}{6} [2400]$$

$$J(180, 100) = 400$$

CALCULATION GRADIENT DESCENT

Repeat Until Convergent {

$$\theta_j := \theta_j + \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

)

Fungsi --> $f_{w,b}(x^{(i)}) = \mathbf{w}x^{(i)} + \mathbf{b}$

Cost Function --> $J(\mathbf{w}, \mathbf{b}) = \frac{1}{2m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})^2$

Gradient Descent will be

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

Then

$$\mathbf{w} := \mathbf{w} - \alpha \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$
$$\mathbf{b} := \mathbf{b} - \alpha \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})$$

Step by step derivatif

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

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})^2 \right)$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{2m} \sum_{i=0}^{m-1} \frac{\partial}{\partial \mathbf{w}} (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{2m} \sum_{i=0}^{m-1} \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}x^{(i)} + \mathbf{b} - y^{(i)})^2$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{2m} \sum_{i=0}^{m-1} (2(\mathbf{w}x^{(i)} + \mathbf{b} - y^{(i)}) x^{(i)})$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=0}^{m-1} (\mathbf{w}x^{(i)} + \mathbf{b} - y^{(i)}) x^{(i)}$$

$$\frac{\partial J(w, b)}{\partial w} = \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$

Untuk yang b

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

$$\frac{\partial J(w, b)}{\partial b} = \frac{\partial}{\partial b} \left(\frac{1}{2m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})^2 \right)$$

$$\frac{\partial J(w, b)}{\partial b} = \frac{1}{2m} \sum_{i=0}^{m-1} \frac{\partial}{\partial b} (wx^{(i)} + b - y^{(i)})^2$$

$$\frac{\partial J(w, b)}{\partial b} = \frac{1}{m} \sum_{i=0}^{m-1} (wx^{(i)} + b - y^{(i)})$$

$$\frac{\partial J(w, b)}{\partial b} = \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})$$

So

$$w := w - \alpha \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$b := b - \alpha \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})$$