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, imporves 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)

- classfiying emails as spam or not spam (T)

- the number (or fraction) of email correctly classified as spam/not spam.

1. **Kind of Machine Learning Algoritms:**

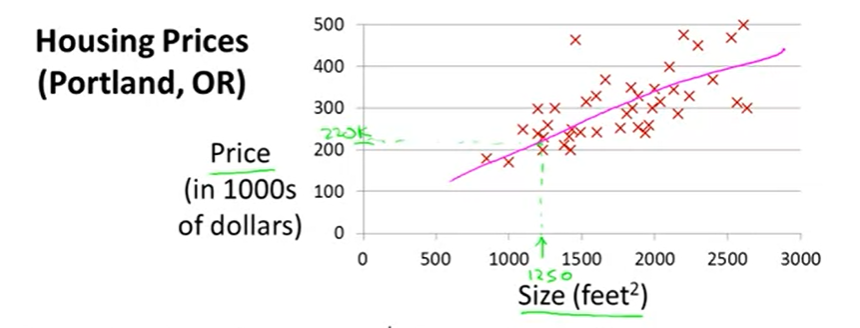
- Supervised Learning

- Unsupervised Learning

Other : Reinforcement learning, recommeder systems.

1. **Model and Cost Function**
2. **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.

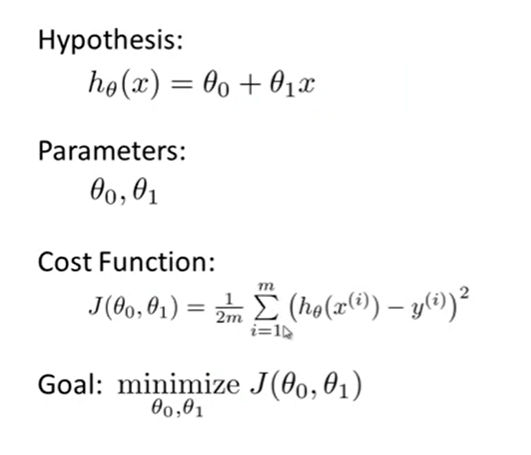
The first we given **several data that repesentation by point** with color red,

**the model is line** that have value approach every data.

1. **Cost Function**

Cost Function is function can calculate how close is the model with actual data.

Before that we can define function



- **Hypothesis :** model that can predict the data

- **Parameters** : parameters of model, this parameter determine how to close model with actually data

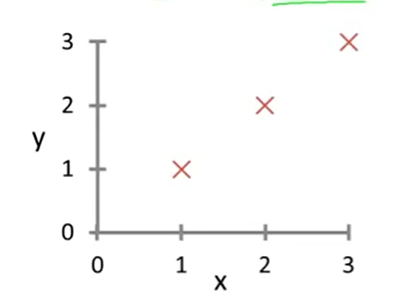
- **Cost Function** : Function that **measure closeness** of model with actual data

- **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.

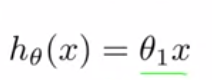
1. **Example**

Example 1 :

- we have data

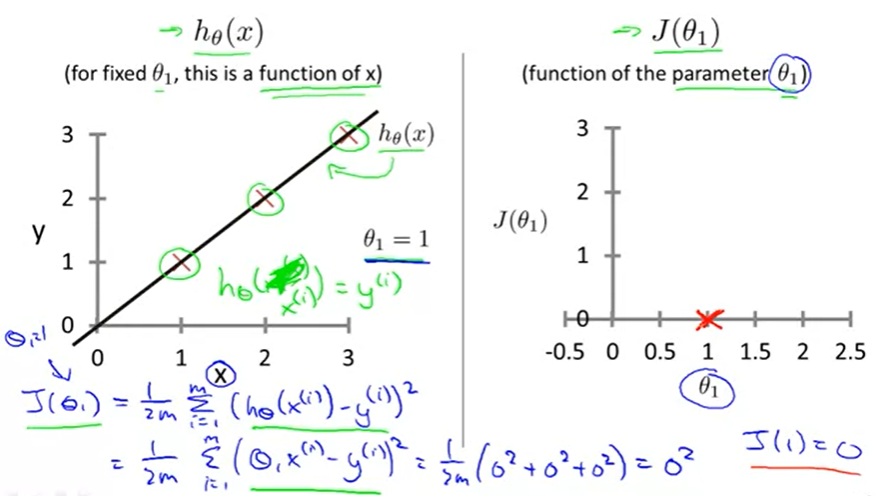


- we have hipotesys function , y =

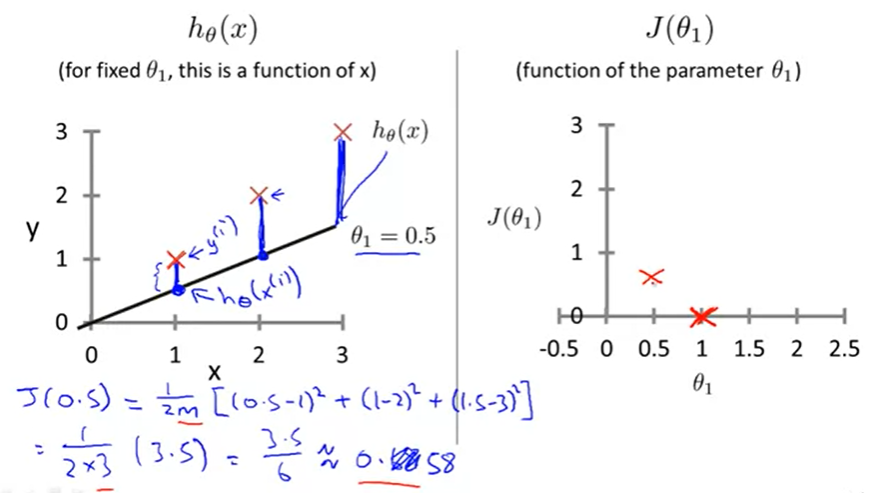


- so with can try each of **value parameter** to get minimum cost function.The first we try value parameter 0.5 and 1.

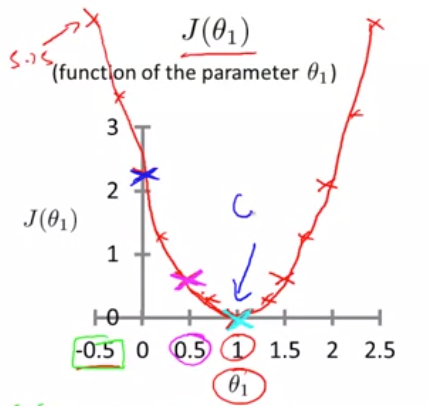
**Theta = 1**

d

**Theta = 0.5**



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

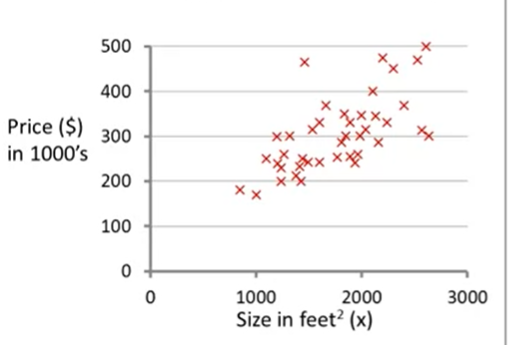


So the goal is find Minimize of Cost Function, in this case the Minimum of Cost function when parameter theta equal 1, which is have Cost Function equal 0.

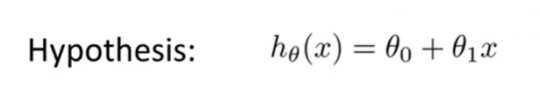
So the conclution we can use parameter theta equal 1 for predict the data model.

**Example 2**

Try to use 2 value parameter



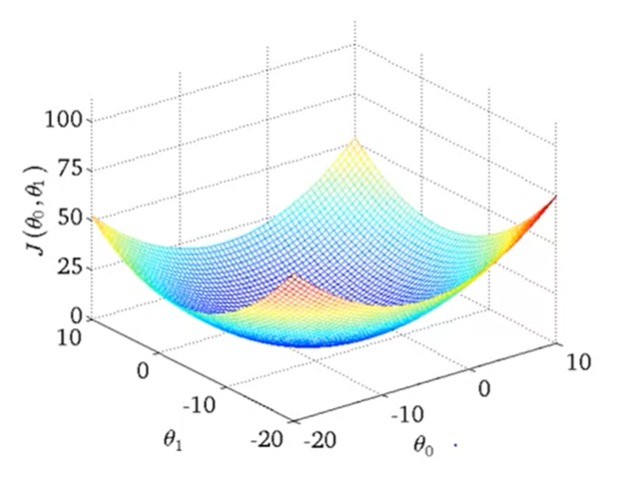
So for predict this data, we can try to use 2 value parameter, with 2 parameter the hypothesis paramater can to be:



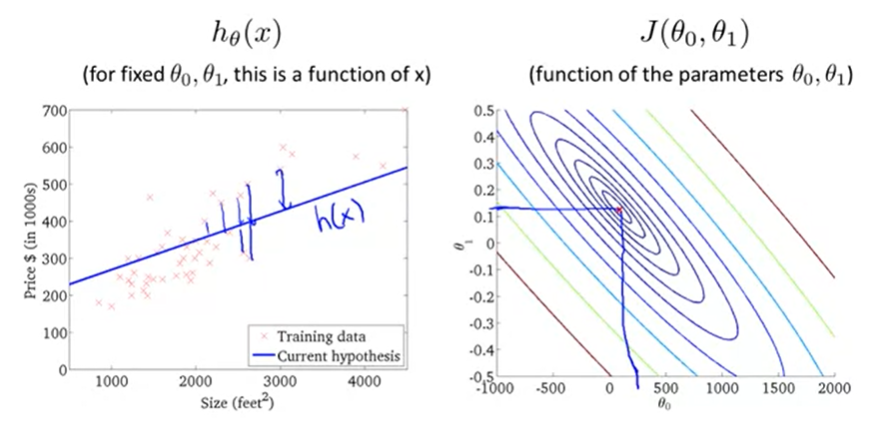
Theta 0 and theta 1

So to find Min Cost we must calculate 2 parameter.

So we can have **Cost Function** Grafik in 3 Dimention



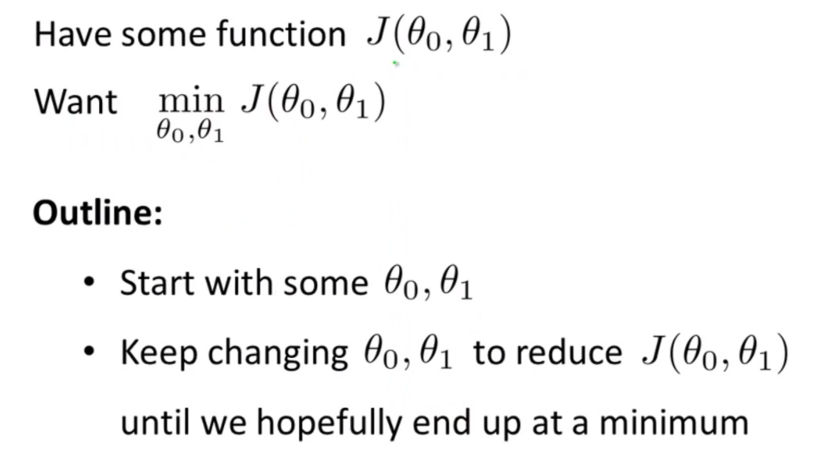
We don’t use 3 dimention grafik for analytic but we can ty to understand how to user contour plot, the contour is transformation of 3 dimention grafik abouve



To read contour plot, contour plot have same value of cost fucntion every same line. The minimize of cosst function is at center of contour plot, but carefully the center of contour plot can repsentation of minimum value or maxsimum value, before determina we must know it.

1. **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.



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

Idea for gradient descent. We can decade the initial guesses for theta 0 and theta 1. don’t really matter what they are, but commonly we choice theta 0 to 0 and theta 1 to 0.

And after that gradient descent can keep changing the theta 0 and the theta 1 a little bit to try reduce the function J , until hopefully we find minimum or minimum local.

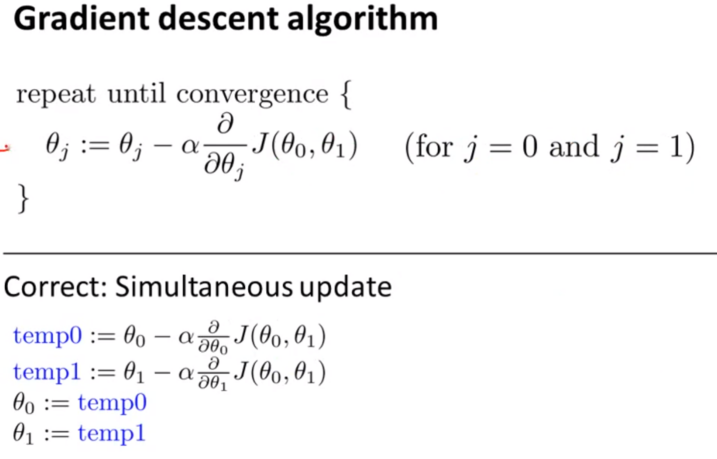
Intuition how gradient descent work and how different choice **intialisation value** can change the result or the local minumum .

|  |  |
| --- | --- |
| A | B |

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 minumum.

After we know intuition lats look at the math.



|  |  |  |
| --- | --- | --- |
|  |  | **Assigment Operator** , its mean take value b and use it overwrite whatever value in a |
|  | | **Learning Rate**,(alpa) Learning Rate does is it basically controls how big a step we take.  So if alpa 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. |
|  | | **Derivative Term,** The Derivative can give direction to minimum value. |

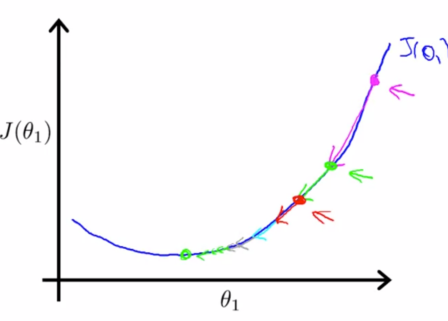
**Derivative**

|  |  |
| --- | --- |
| Positive Slope | Negative Slope |

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

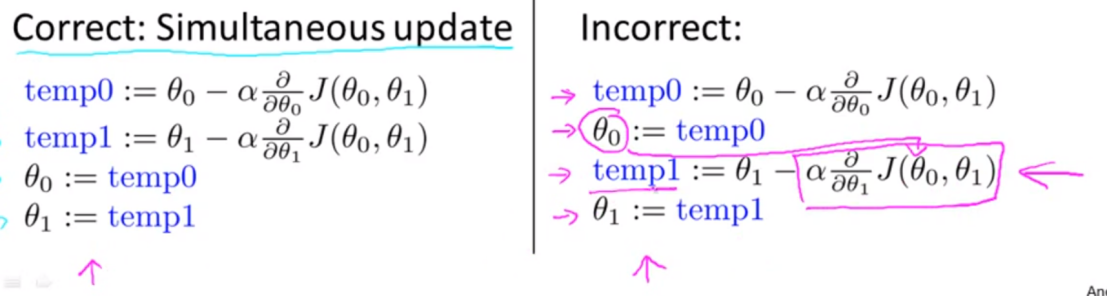
**Learning Rate**

|  |  |  |
| --- | --- | --- |
| 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 overshot, it may be failed to find local minumum 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.

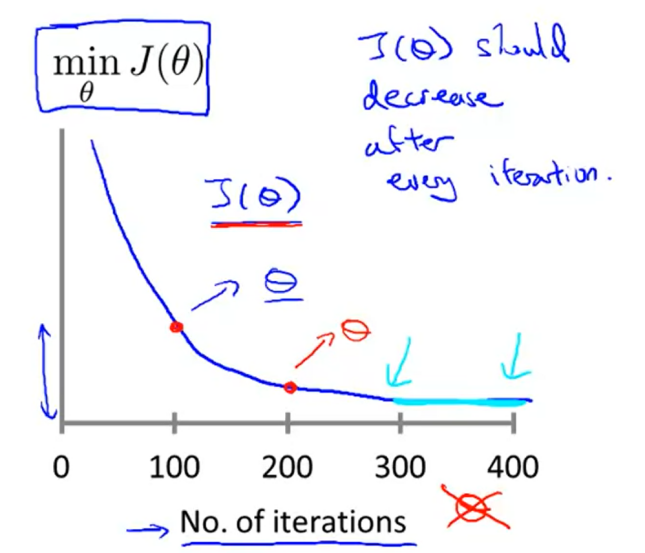
after that, we can explain how to do correct calculation, the update theta0 and theth1, must be simultaneuos update, that mean we can update in same time.



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.

1. **Choice The Great Learning Rate**
2. Debugging (How to make sure gradient descent is working correctly)

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

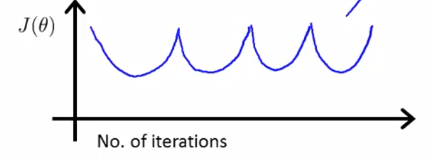
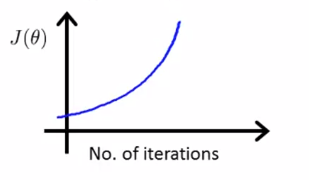


If you looks this graph, you can see the data wa convergent in 300 No of Iteration to 400 Iterations. If you find some like this your graf is correct.

Acctually you can know your iterations was convergent or not use

**automatic convergent test,** the test can compare between value of decreses, if the value changing was very small such as 0.0001 (epsilon) that mean the data was convergent

But if you find something like this



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

1. How to Choose Learning Rate Alpha

To Choose your Learning Rate Alpha acctually you must try every posibility for example you choise 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 compherensive.

1. **Manual Calculation**

|  |  |
| --- | --- |
|  | **Parameter** ,Nilai yang di optimasi) |
|  | **Assigment Operator**, Operator menggantikan value , ( ab , ini bermasud b menggantikan value a) |
|  | **Learning Rate ,**Mengkontrol seberapa besar step yang dilakukan. |
|  | **Nilai Turunan / Derivative Term** , yang memberi arah menuju minimum/optimal |

|  |  |  |
| --- | --- | --- |
|  |  | **Assigment Operator** , its mean take value b and use it overwrite whatever value in a |
|  | | **Learning Rate**,(alpa) Learning Rate does is it basically controls how big a step we take.  So if alpa 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. |
|  | | **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

CALCULATION GRADIENT DESCENT

Repeat Until Convergent {

)

Fungsi -->

Cost Function -->

Gradient Descent will be

Then

Step by step derivatif

Untuk yang b

So