2] Regularization

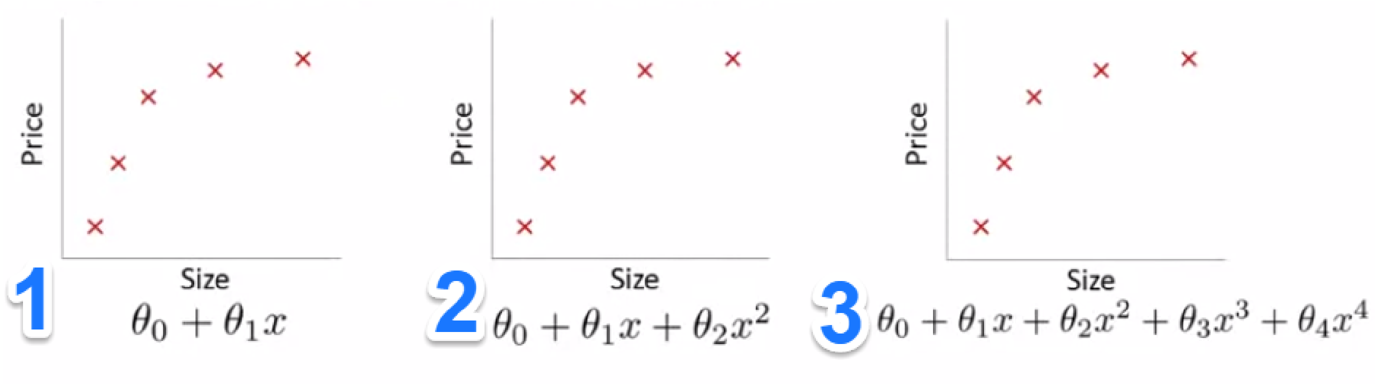
Notebook:	Machine Learning		
Created:	16-Apr-20 5:07 PM	**Updated:**	02-May-20 3:07 PM
Author:	Patel Yas		

What is Regularization ?

Regularization is used to solve the problem of overfitting.

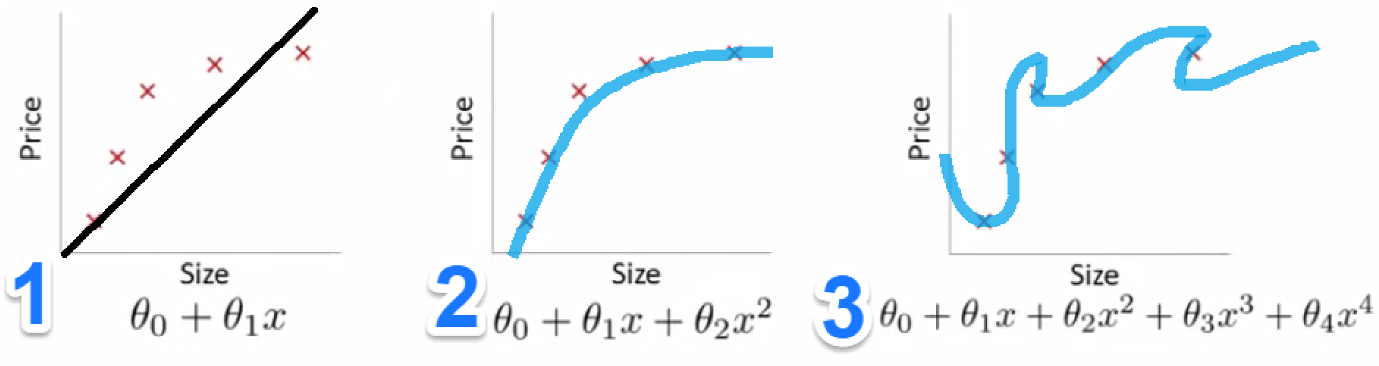
What is overfitting ?

Let's take **linear regression** (housing prices example)
following are different dataset is given.



so we are trying to fit the hypothesis into this dataset and want to predict new example's (dataset's) housing price.

so here are the predicted hypothesis according to dataset.



here, in this above hypothesis :
dataset : 1) **unfit / High bias**

- fitted hypothesis is good
- but hypothesis doesn't contain all dataset.
- it's doesn't good for future prediction.

dataset : 2) **just right**

- fitted hypothesis is right
- it contain 80% of our dataset,
- by this type of continues hypothesis we can predict new examples

dataset : 3) **overfit / highvariance**

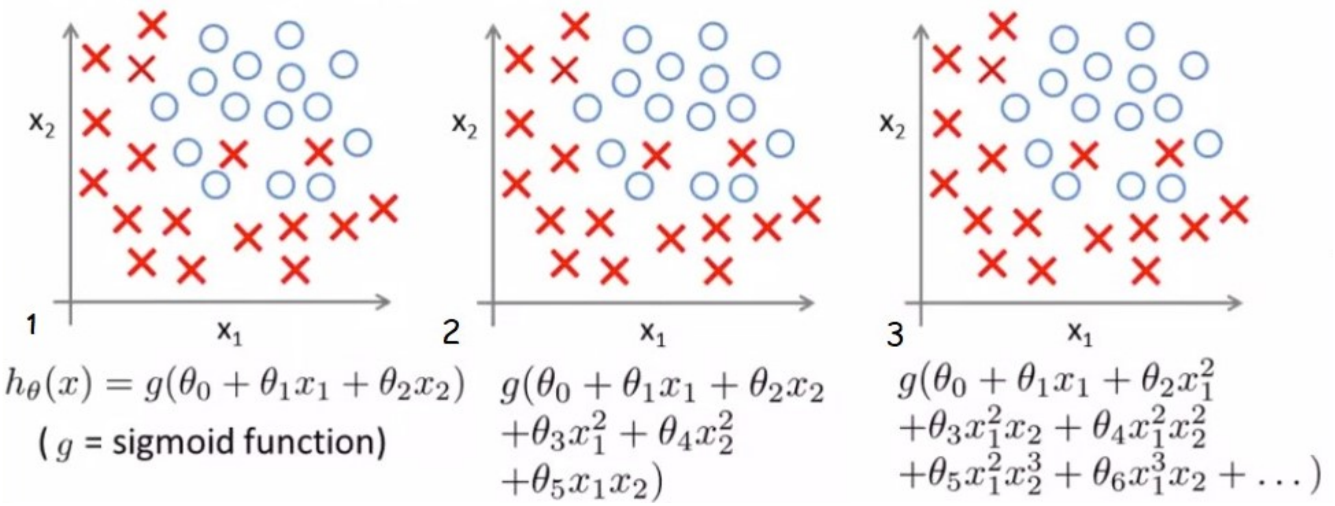
- fitted hypothesis is perfectly good
- it contain all dataset but it is hard to locate continues hypothesis.
- we can not use it for future prediction.

What is Exact Problem of overfit ?

when given dataset it's fit for all of the dataset so good
but it's fail generate for new examples0 / predict prices for new examples.

Let's take **logistic regression** (housing prices example)

following are different dataset is given.



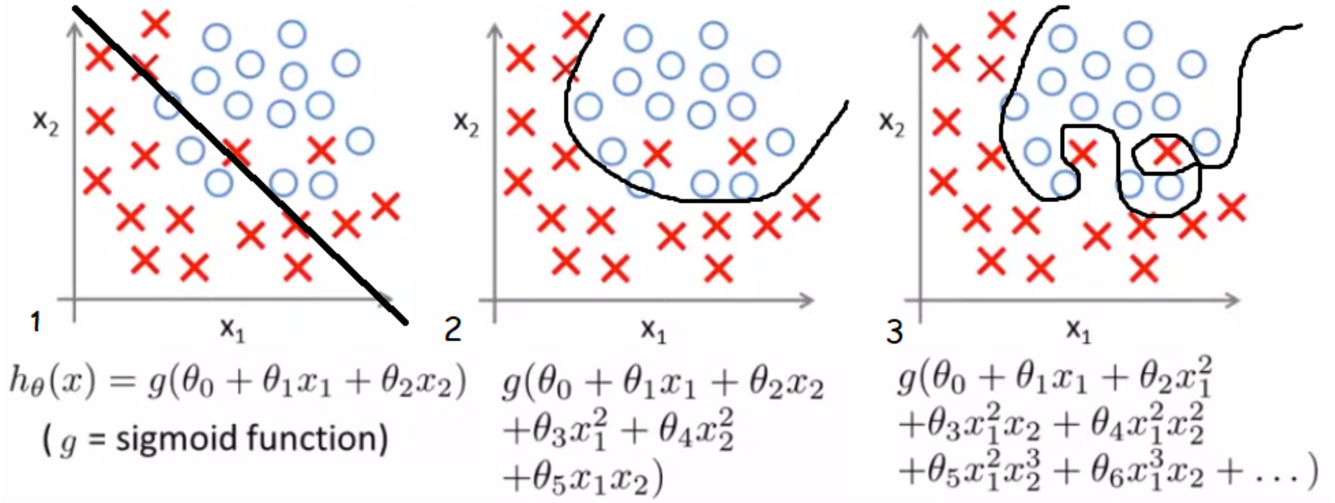
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ (g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

so we are trying to fit the hypothesis into this dataset and want to predict new example's (dataset's) housing price.

so here are the predicted hypothesis according to dataset.



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ (g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

here, in this above hypothesis same as linear regression :
dataset : 1) **unfit / High bias**
dataset : 2) **just right**
dataset : 3) **overfit / highvariance**

When we have lot's of features with very little training data overfitting will become problem for sure.

How to solve this overfitting problem ?

there are two solution :
1) Reduce number of features

- Manually select which features to keep
- Model Selection algorithm (let algorithm decide which to keep)

2) Regularization

- keep all the features but reduce the magnitude/ values of parameter Θ_j .
- regularization works well when we have lot's of features.
- each of these contributes a bit to predicting y.

Reducing number of features will not work when we have to compulsory keep all the features.
so Regularization will help us to solve the overfitting problem.

WHAT should be our cost function to avoid overfitting ?

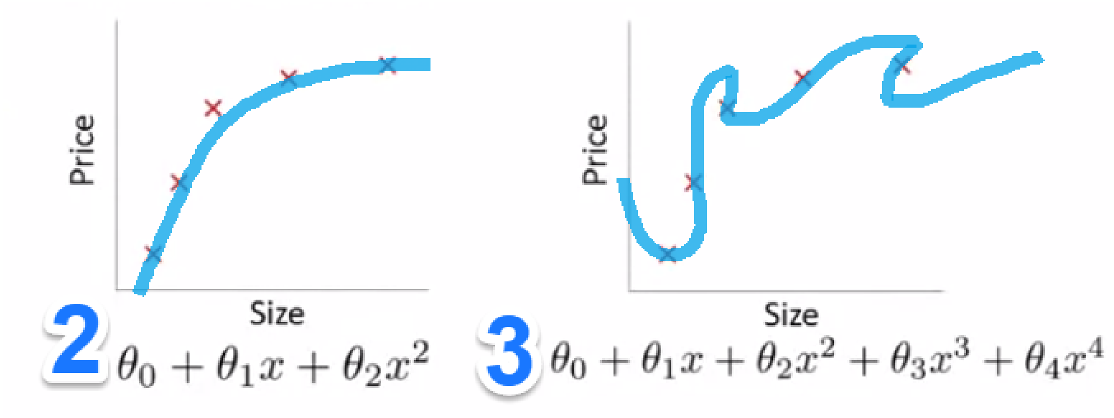
here is the cost function for linear regression :

$$min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

SO here, in this below diagram as mentioned
our hypothesis function for

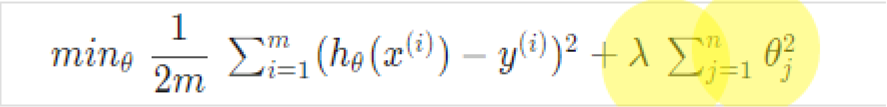
diagram 2 : $\Theta_0 + \Theta_1X + \Theta_2X^2$

diagram 3 : $\Theta_0 + \Theta_1X + \Theta_2X^2 + \Theta_3X^3 + \Theta_4X^4$



So diagram 2 is our best fitting & diagram 3 is our overfitting so we need to reduce this overfitting and make it look like best fitting.
in order to do that observe the both hypothesis function if we reduce the values of Θ_3 & Θ_4 (values like negalatable) .
SO we can transform this eq $\Theta_0 + \Theta_1 X + \Theta_2 X^2 + \Theta_3 X^3 + \Theta_4 X^4$ in this eq $\Theta_0 + \Theta_1 X + \Theta_2 X^2$ by doing reducing the values of Θ_3 & Θ_4 .
AS Θ_3 & Θ_4 values are so small that we can eliminate Θ_3 & Θ_4 it from the equation.

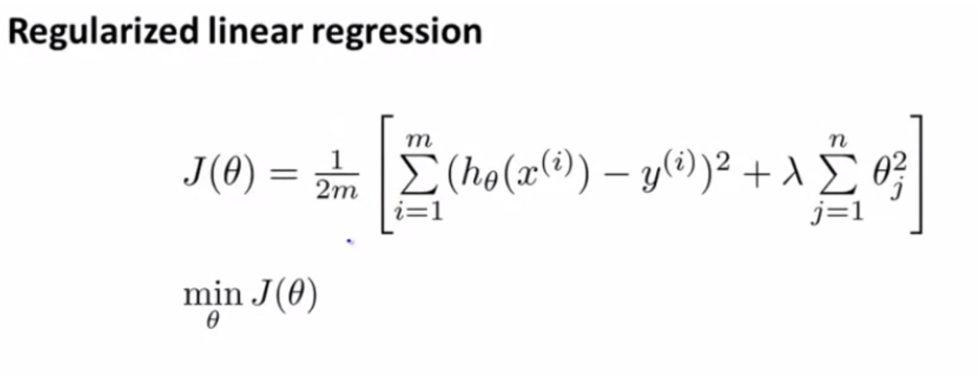
In order to eliminate the influence of Θ_3 & Θ_4 instead changing form of hypothesis or reducing features,
here what we can do is we can modify our cost function like this :



$$min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

here, last added term is called **regularization term.**
λ is called the **regularization parameter**.
λ should be as small as possible to fit the training data.
if we set **λ too large** (perhaps for our problem extremely large, say 10^{10}),
our algorithm results in **underfitting**, it will even fail to fit our whole training dataset.

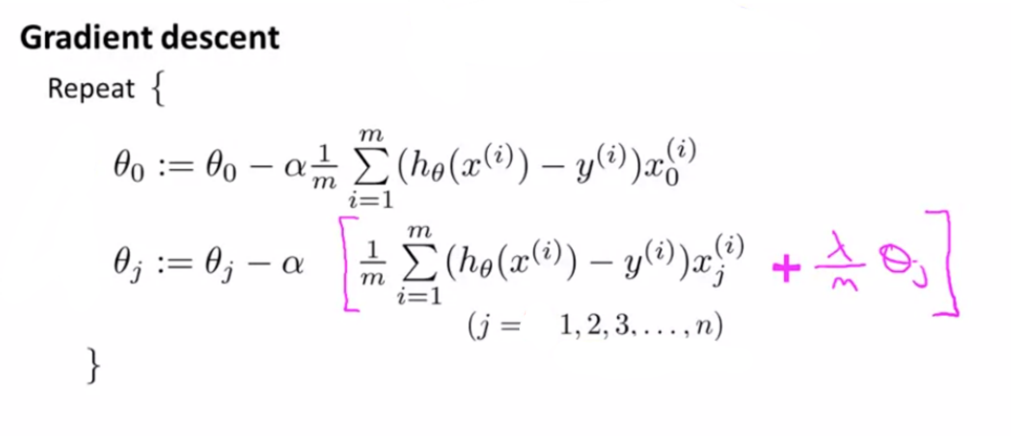
let's solve overfitting problem for linear regression :



Regularized linear regression

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

$$\min_\theta J(\theta)$$



Gradient descent

Repeat {

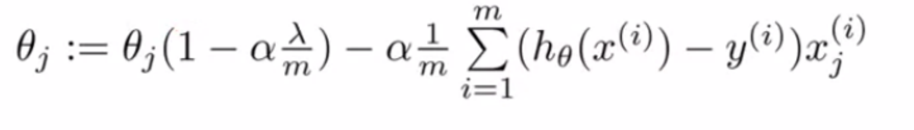
$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

$(j = \quad 1, 2, 3, \dots, n)$

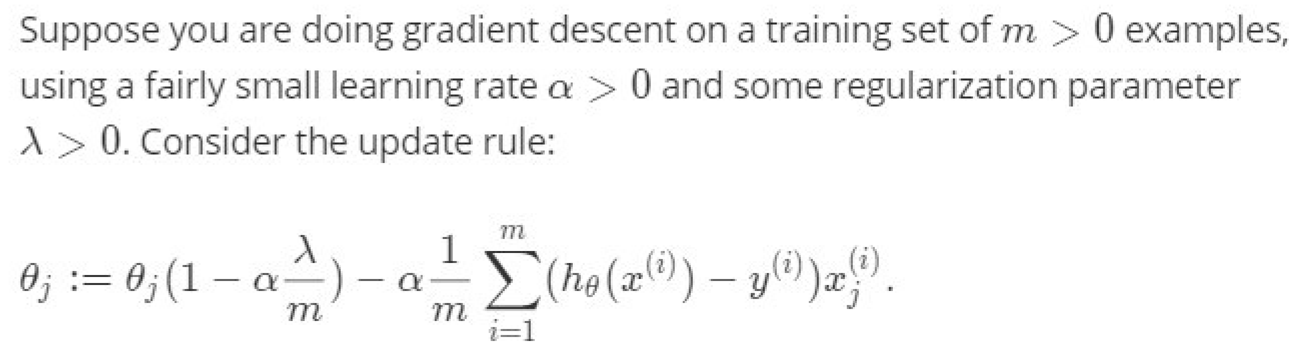
}

WE can also rewrite this Θ_j as follows :



$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

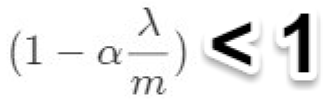
SO



Suppose you are doing gradient descent on a training set of $m > 0$ examples, using a fairly small learning rate $\alpha > 0$ and some regularization parameter $\lambda > 0$. Consider the update rule:

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}.$$

for that following term must be less than 1



$$(1 - \alpha\frac{\lambda}{m}) < 1$$

let's solve overfitting problem for logistic regression :

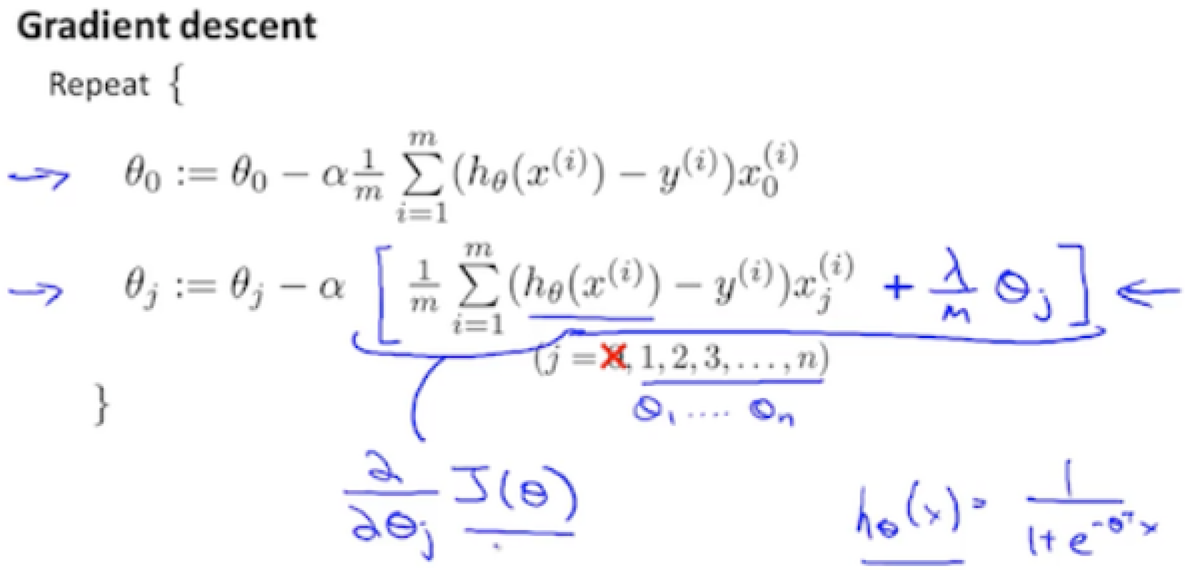
Cost Function

Recall that our cost function for logistic regression was:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1 - y^{(i)})\ \log(1 - h_\theta(x^{(i)}))]$$

We can regularize this equation by adding a term to the end:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1 - y^{(i)})\ \log(1 - h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$



Gradient descent

Repeat {

$$\rightarrow \quad \theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\rightarrow \quad \theta_j := \theta_j - \alpha\ \underbrace{\left[\frac{1}{m}\sum_{i=1}^{m}(\underline{h_\theta(x^{(i)})} - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]}_{\frac{\partial}{\partial\theta_j}\underline{J(\theta)}} \leftarrow$$

$$(j = \cancel{0}, \underbrace{1, 2, 3, \ldots, n}_{\theta_1 \ldots\ \theta_n})$$

}

$$\underline{h_\theta(x)} = \frac{1}{1 + e^{-\theta^T x}}$$