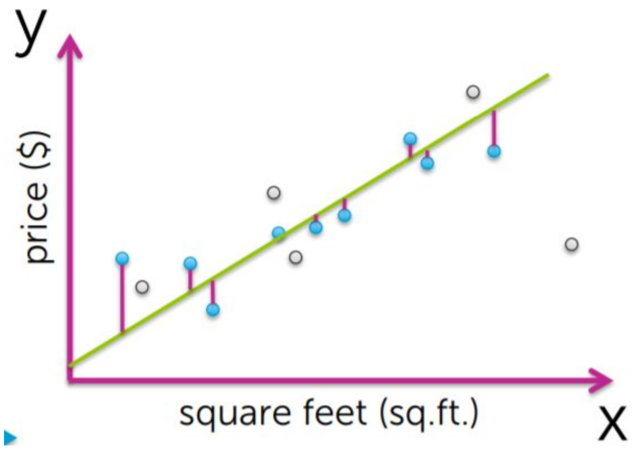
Introduction to Machine Learning: Regression

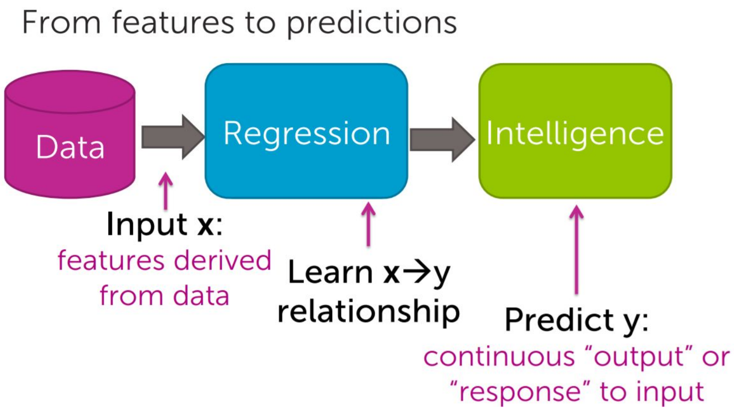
By Shrishty Chandra

What is regression ?

Regression is a **model**, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a combination of the input variables (x).

Train Data + Test Data





Usecase

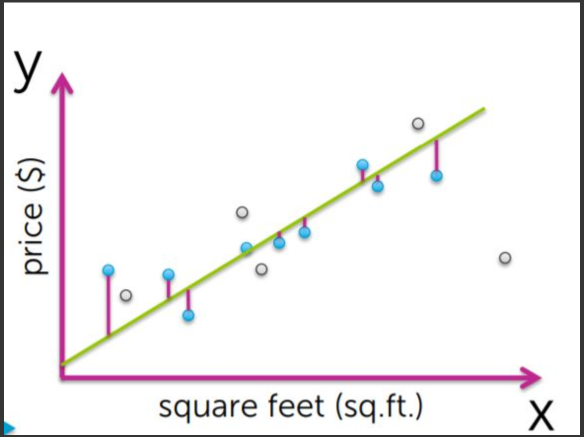
Predicting House Prices: Given features of the house, predict the price of it.

- **f(x=sq.ft, y=$)**
  We will define a function which takes sq.ft of the house and predict the house price.

- **f(x1=sq.ft,x2=#bedrooms, y=$)**
  Given multiple features, predict the price of the house

- **Feature selection and performance (Not talking but Imp)**
  There are 100,000 features in a house, how to choose the correct features

Regression Model



$$y_i = f(x_i) + e_i$$

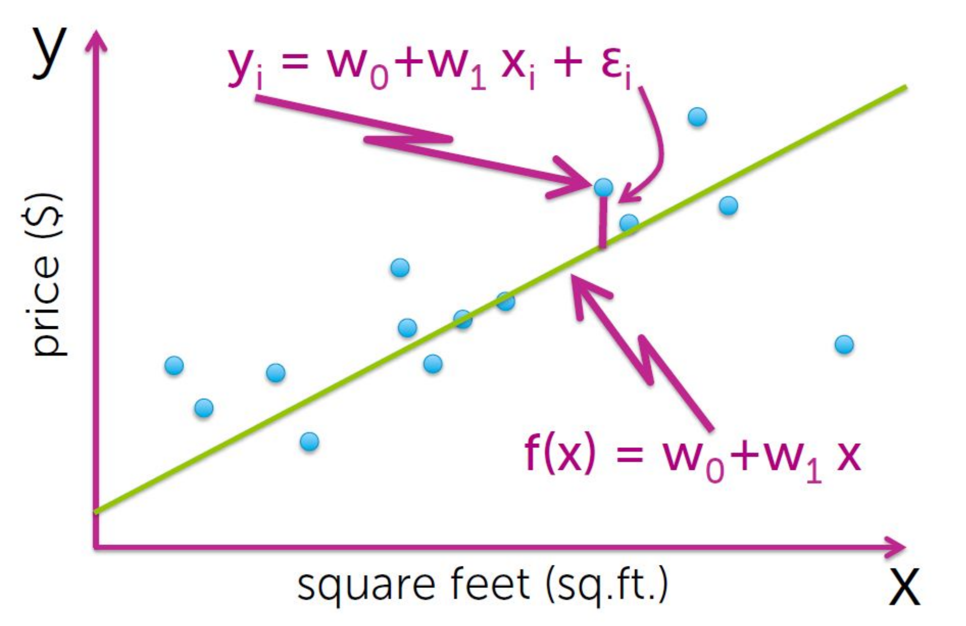
Tip

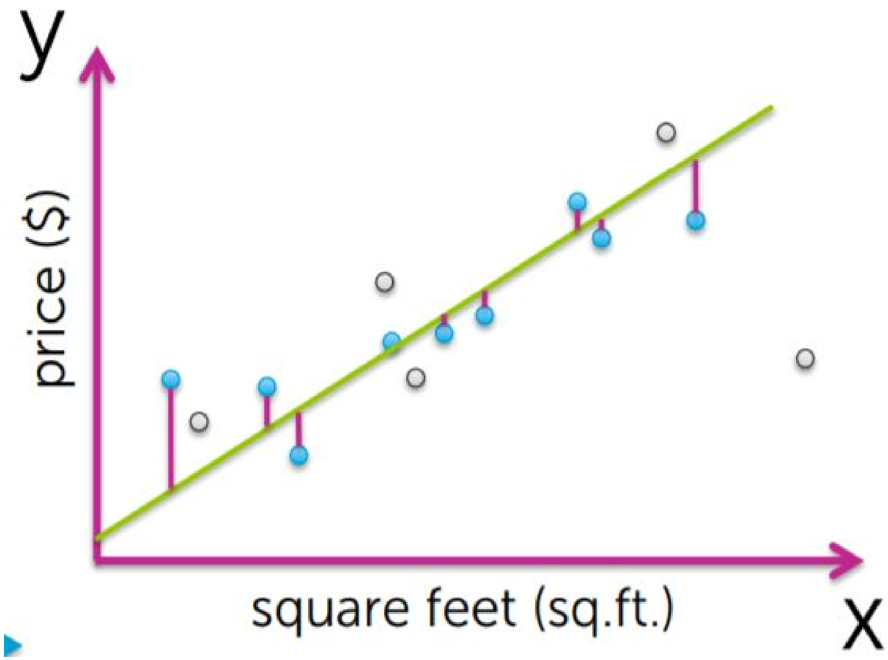
Essentially all models are wrong but some are useful.

George Box, 1987

Simple Linear regression model

- W_0 and W_1 are regression coefficients
- ε_i is the error.
- f(x) is the fitted line in the data.





Residual Sum of Squares

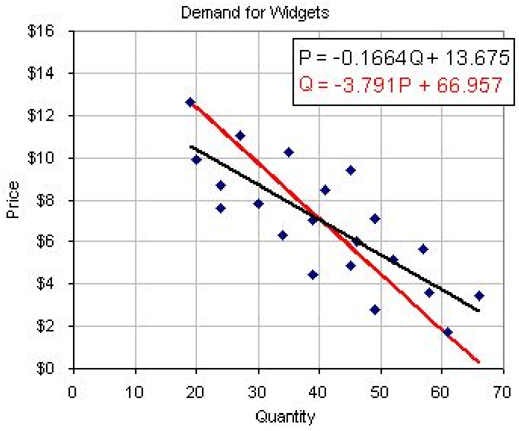
Sum of squares of the errors in predictions.

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

Fitting the best line

We need to minimize the cost over all possible W_0, W_1, to fit the best line.

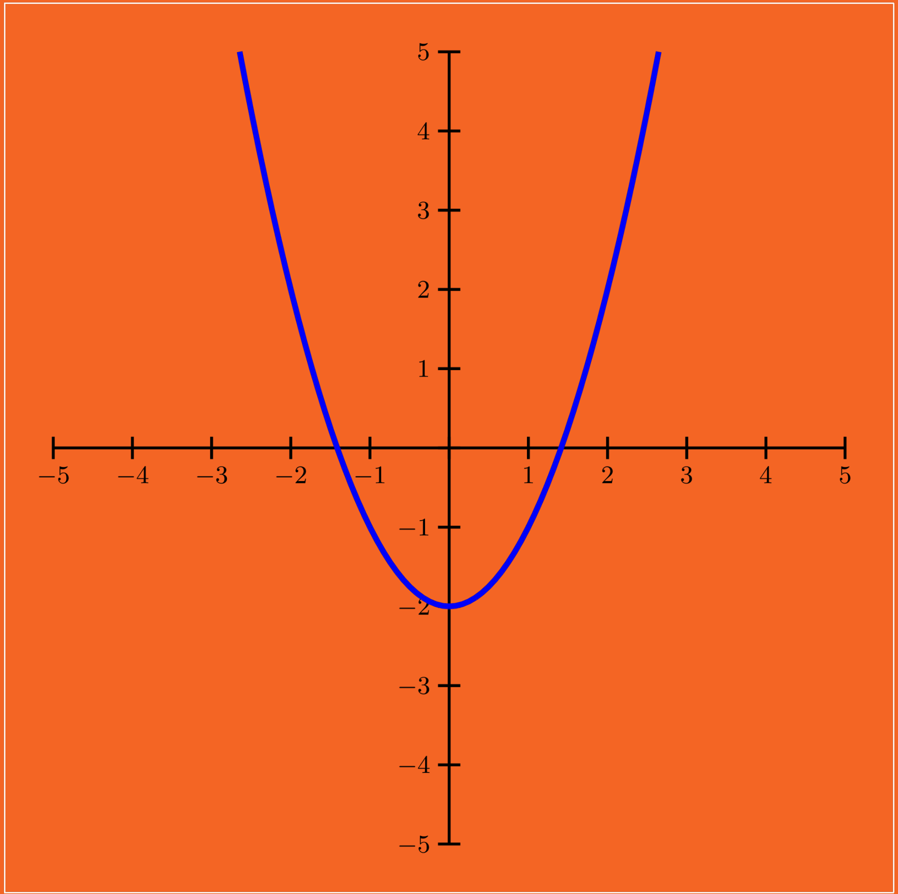


RSS is a quadratic equation with respect to W_0 and also W_1.

When we are calculating W_0 such that it minimizes RSS we will consider all other variables as constant

Similar for W_1

So where do you think this function minimum here ?



Two ways to calculate the minimum

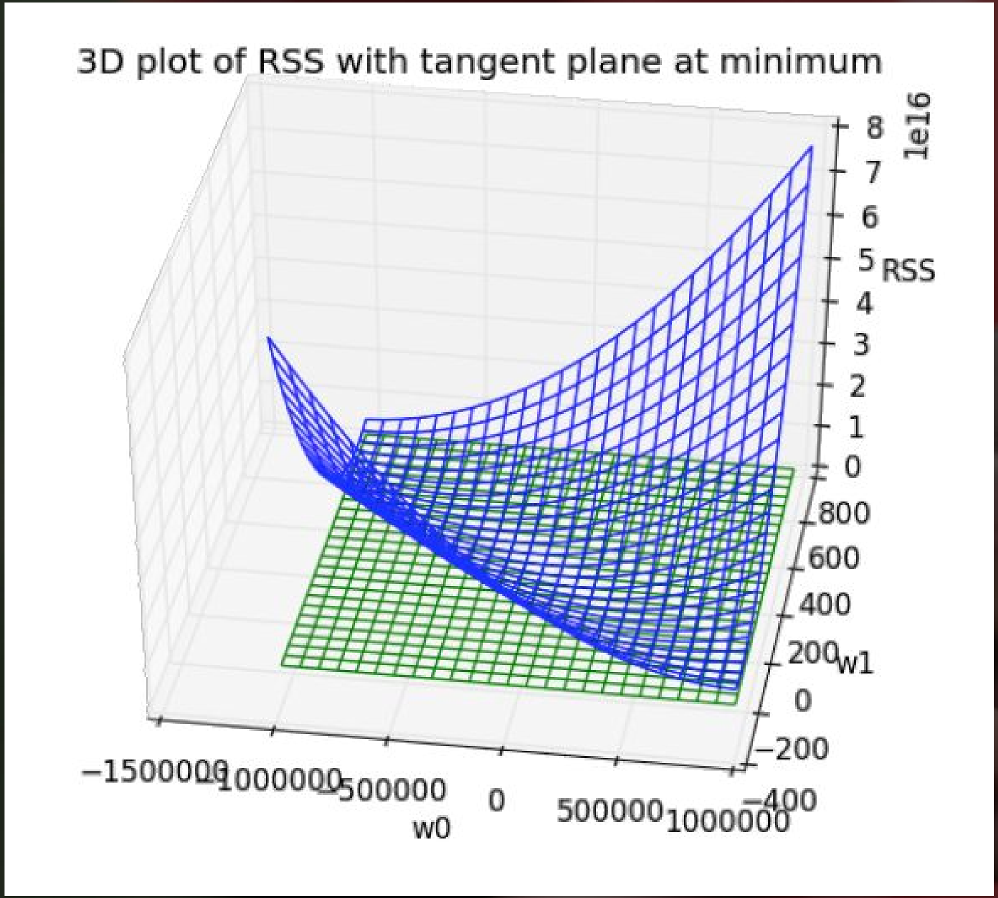
- Find the point where the derivative is zero
- Gradient descent
  - η is the step size here, represents how fast move towards the optimum w
  - Common choices of η are
    - $\eta_t = \boldsymbol{\alpha}/t$
    - $\eta_t = \boldsymbol{\alpha}/\mathrm{sqrt}(t)$

Hill Descent algorithm

Algorithm:

while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$



Multidimensional view of gradients

$g(w) = 5W_0 + 10W_0W_1 + 2W_1^2$

$\partial g/\partial w_0 = 5 + 10W_1$

$\partial g/\partial w_1 = 4W_1 + 10W_0$

$\nabla g(w) = [\partial g/\partial w_0, \partial g/\partial w_1]$

So if,

$$\mathrm{RSS}(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

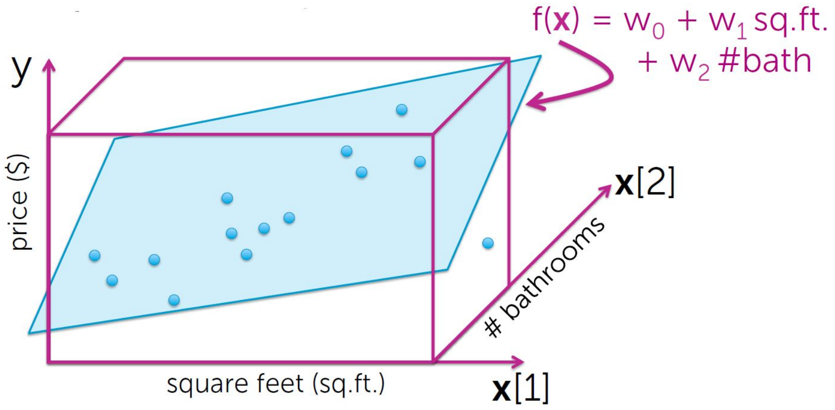
After setting the gradient to zero. We will be able to calculate W_0 and W_1

Then,

$$\nabla \mathrm{RSS}(w_0,w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)] \\ -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)]x_i \end{bmatrix}$$

Comparing the derivative and gradient approach

- Most ML problems cannot solve gradient = 0
- Even if solving gradient = 0 is feasible, gradient descent can be more efficient
- But gradient descent depends on step size and convergence criteria



Multiple Regression

Linear regression with multiple features

There are many possible inputs

- Sqft
- #bedrooms
- #bathrooms etc

General notation

- Output: y (Scalar)
- Inputs: x = (x[1], x[2], x[3], x[4].....)
  - D-dimensional vector
  - D is number of features
- x[j] = jth input (scalar)
- hj(x) = jth feature (scalar)
- xi = input of ith data point (vector)
- xi[j] = jth input of ith data point (scalar)

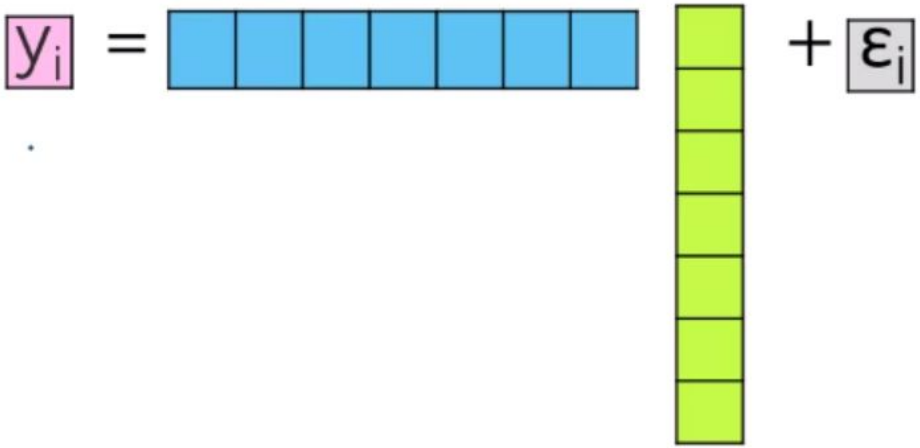
Model:

$$y_i = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) + \varepsilon_i$$

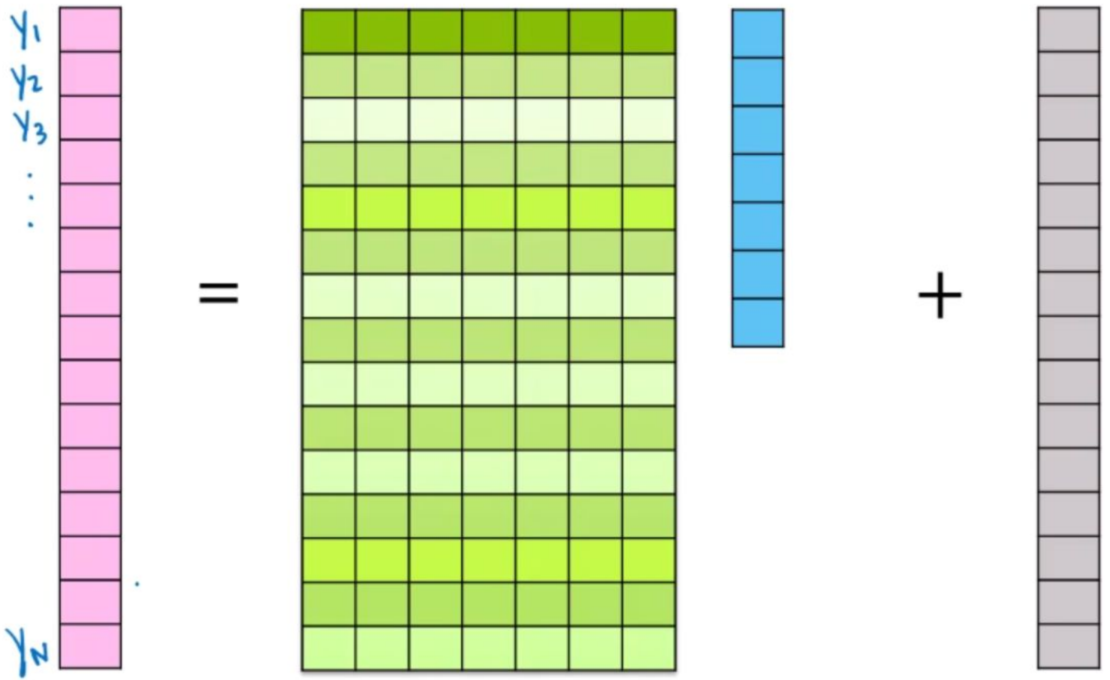
$$= \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i) + \varepsilon_i$$

In Matrices

$$y_i = \sum_{j=0}^{D} w_j \, h_j(\mathbf{x}_i) + \varepsilon_i$$



Calculating all house prices at once



Calculating RSS in Matrix

residual_1	residual_2	residual_3	...	residual_N

residual_1
residual_2
residual_3
...
residual_N

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - h(\mathbf{x}_i)^T \mathbf{w})^2$$

$$= (\mathbf{y} - \mathbf{H}\mathbf{w})^T (\mathbf{y} - \mathbf{H}\mathbf{w})$$

The Gradient

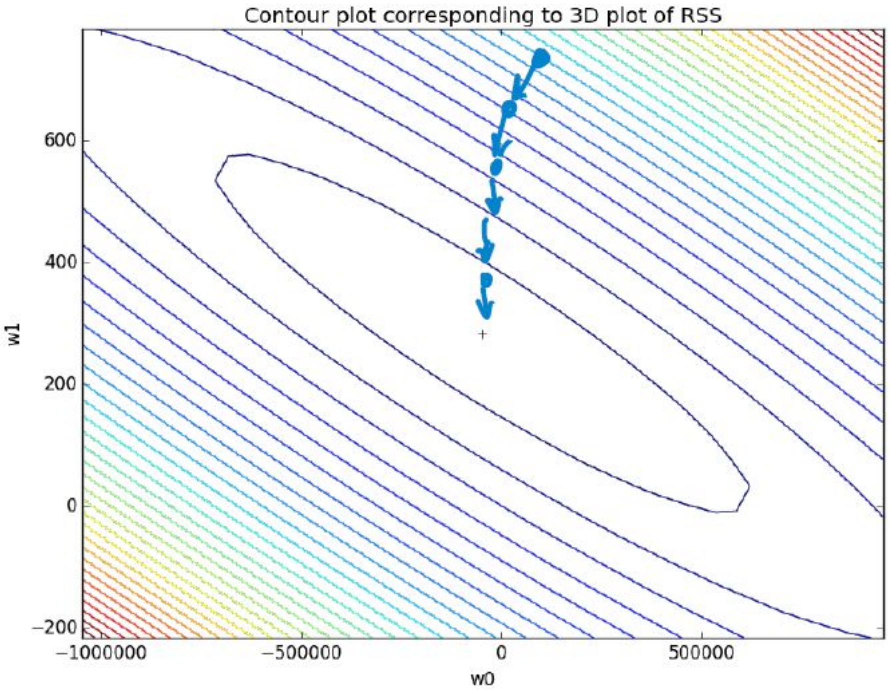
$$\nabla \mathrm{RSS}(\mathbf{w}) = \nabla[(\mathbf{y}-\mathbf{H}\mathbf{w})^T(\mathbf{y}-\mathbf{H}\mathbf{w})]$$

$$= -2\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{w})$$

$$\nabla \mathrm{RSS}(\mathbf{w}) = -2\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{w}) = 0$$

Approach 1

- Set the gradient to zero
- Then solve for W



Approach 2

- Use Gradient descent to calculate the optimum w

while not converged

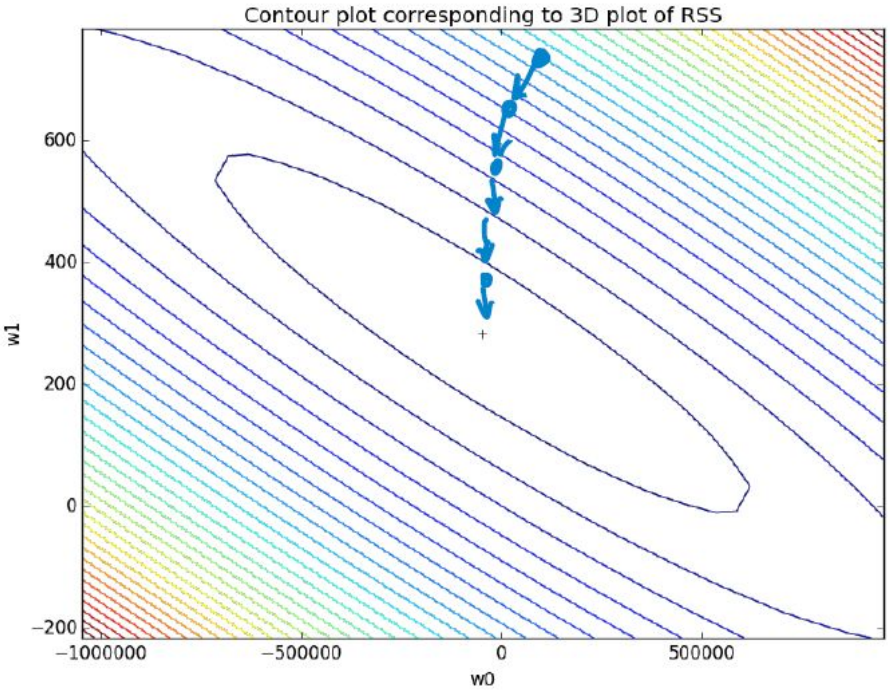
$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \underbrace{\nabla RSS(\mathbf{w}^{(t)})}_{-2\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{w})}$$

Reference

- https://www.coursera.org/learn/ml-regression/home/welcome

Other good resources:

- Machine Learning Recipes: https://www.youtube.com/watch?v=cKxRvEZd3Mw
- Tensorflow, https://www.youtube.com/watch?v=g-EvyKpZjmQTensorflow, https://www.youtube.com/watch?v=g-EvyKpZjmQ
- Neural Network, https://www.youtube.com/watch?v=NfnWJUyUJYU&list=PLwQyV9I_3POsyBPRNUU_ryNfXzgfkiw2p&index=1
- AndrewNG Machine Learning: https://see.stanford.edu/Course/CS229



Lets See the code