1 Linear Regression

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} w^{(i)}(y^{(i)} - \theta^T x^{(i)})^2$$

For optimizing J(θ) batch gradient descent is used. Update rule for

$$\theta_j = \theta_j - \eta\frac{\partial}{\partial\theta_j}J(\theta)$$

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

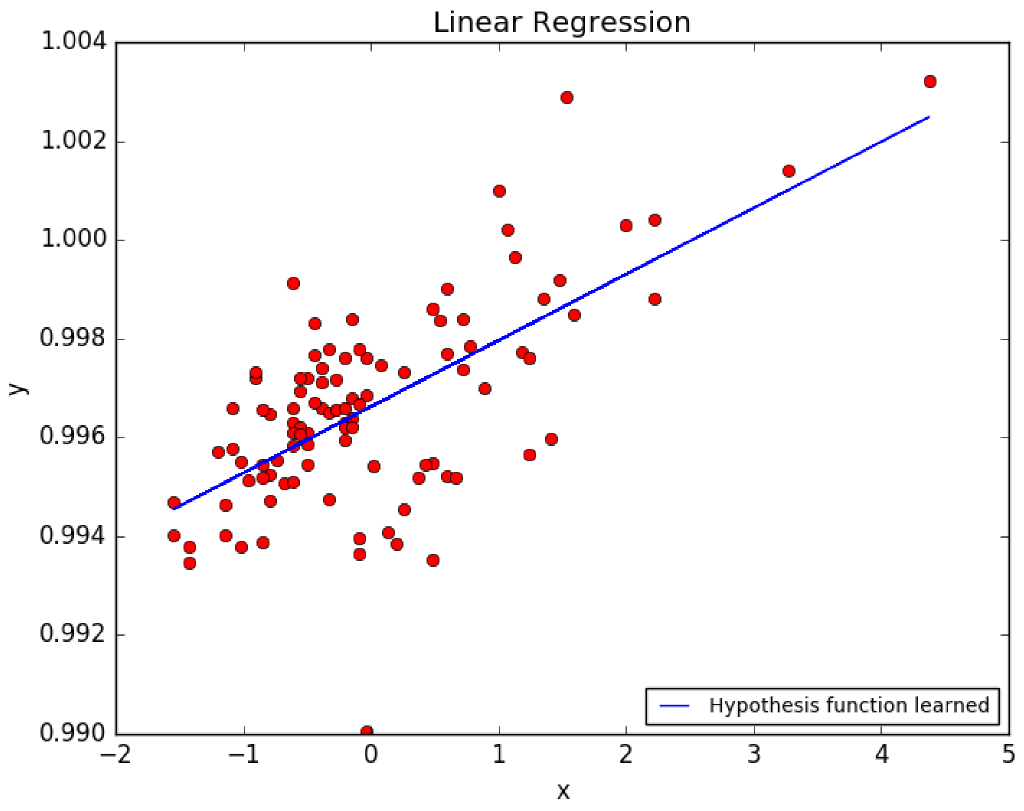
a) Final set of parameters :

$$LearningRate = 0.005$$

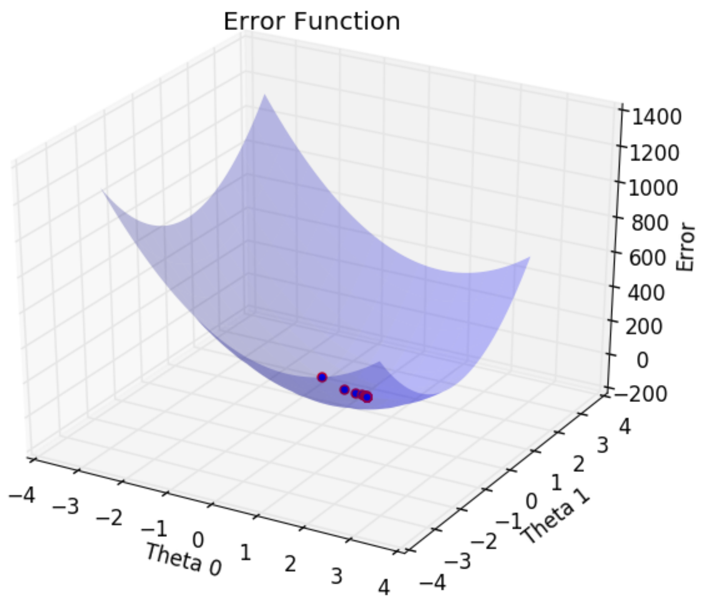
$$StoppingCriteria = \mid J(\theta)_{old} - J(\theta)_{new} \mid < 10^{-12}$$

$$\begin{bmatrix}\theta_0\\ \theta_1\end{bmatrix} = \begin{bmatrix}0.99661981\\ 0.0013402\end{bmatrix}$$

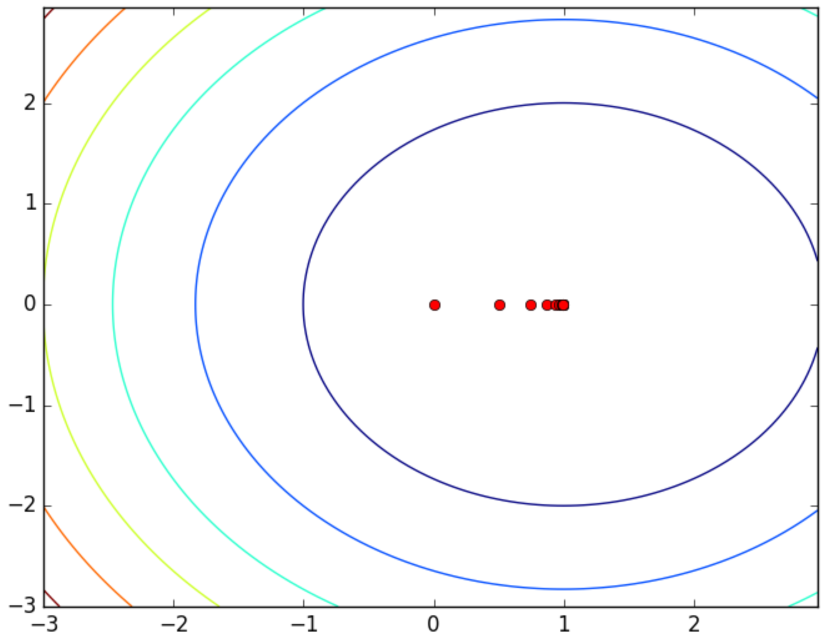
b)



c) 3-d mesh showing error function J(θ) with error value using parameters at each iteration of gradient descent

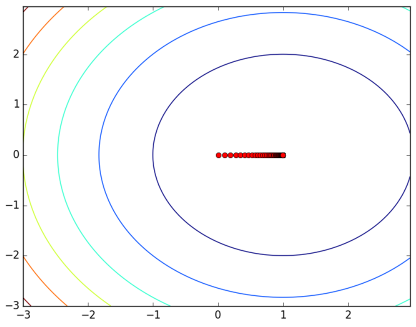


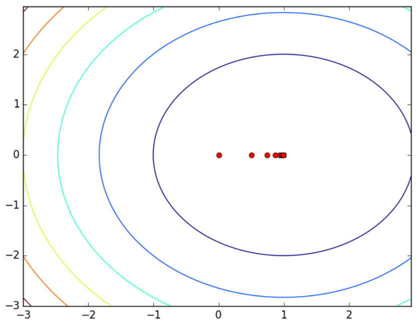
d) Contours of the error function at each iteration of gradient descent

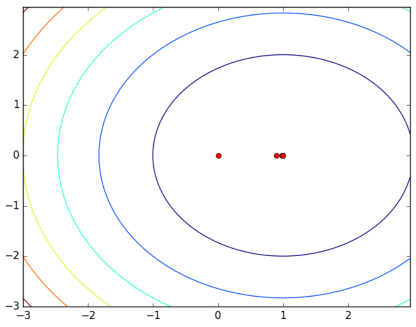


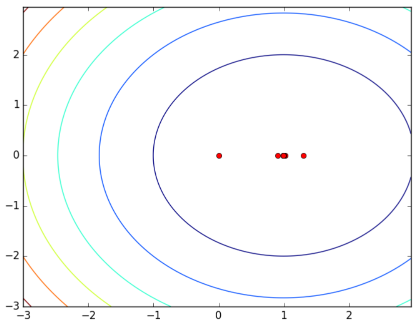
e) Comparative study of different values of η

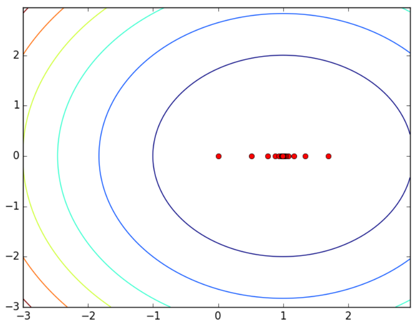
Cont.

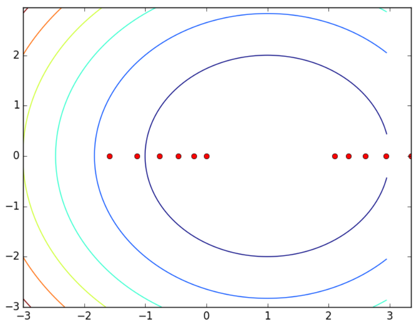

(a) $\eta = 0.001$

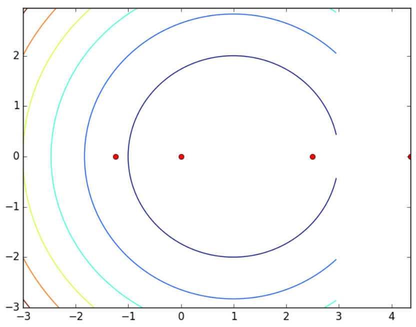

(b) $\eta = 0.005$


(c) $\eta = 0.009$


(d) $\eta = 0.013$


(e) $\eta = 0.017$


(f) $\eta = 0.021$


(g) $\eta = 0.025$

Cont.

η value	No. of iterations to converge
0.001	143
0.005	24
0.009	8
0.013	15
0.017	45
0.021	Do not converge
0.025	Do not converge

We observe that when η value is too small i.e 0.001 it took too many iterations for the algorithm to converge. As we keep on increasing the value from 0.001 to 0.009 no. of iterations required to converge decreases. When we further increase the value to 0.013 and 0.017 we observe that the parameter values after each step oscillate between the optimum value but the error value keeps on decreasing and hence still converges to optimum value. As we increase it further i.e values 0.021 and 0.025 we observe that at each iteration error function increases instead of decreasing and the algorithm never converges. We can summarize it as:
With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point. With a very low learning rate, we move in the direction of the negative gradient but we take small step, so it will take algorithm more time to get to the bottom.

The End.