PROGRESS REPORT

(10.12.18)

* Completed all the course material for the first 2 weeks of the Andrew Ng ML course.
* Learnt about Linear regression(for multiple variables too).
* polynomial regression.
* Implemented Linear regression for one variable on MATLAB as part of the programming submission for the course .
* Learnt how to implement linear regression using normal equations and gradient descent also the pros and cons of both methods.
* Also methods to optimize Linear regression(gradient descent) such as feature scaling and mean normalization.

Cost function code:

Theta is the vector of parameters and X is matrix of features fitted with x0 just a column of ones.

A cost function computes the average difference between the predicted outputs of the hypothesis and the given outputs.

J(cost function)= 1/2*m*​*i*=1∑*m*​(*hθ*​(*xi*​)−*yi*​)2

function J = computeCost(X, y, theta)

m = length(y); % number of training examples

ad=ones(m,1); %vector of ones

f=X\*theta;

diff=f-y;

sq=diff.\*diff; %squaring each element

sum=sq'\*ad;

J=sum/(2\*m);

end

Gradient descent code:

You update the values of the parameters in such a way the cost function reaches its minima, hence giving use the most efficient model.

function [theta, J\_history] = gradientDescent(X, y, theta, alpha, num\_iters)

m = length(y); % number of training examples

J\_history = zeros(num\_iters, 1);

for iter = 1:num\_iters

k=alpha/m;

f=X\*theta;

diff=f-y;

res=(diff'\*X)\*k;

theta=theta-(res)';

end

end

Motor code:

* Will start in the coming days.