Machine Learning Online Class - Exercise 1: Linear Regression

exercise1:Warm up exercise

exercise2:Compute cost for one varible

exercise3:Gradient descent for one variable

exercise4:Feature normalization

exercise5:Compute cost for multiple variables

exercise6:Gradient descent for multiple variables

# Warm up exercise

function A = Warmupexercise()

A = eye(5);

A;

end

# Compute cost for one varible

function J = Computecost(X, Y, theta)

m = length(Y);

J=0;

J = sum((X\*theta-Y).^2)/2\*m;

end

# Gradient descent for one variable

function [theta,J\_all]=Gradientdescent(X,Y,theta,alpha,all\_iters)

m = length(Y);

theta\_s=theta;

J\_all=zeros(all\_iters,1);

for i=1:all\_iters

theta(1)=theta(1)-alpha/m\*sum(X\*theta\_s-y);

theta(2)=theta(2)-alpha/m\*sum((X\*theta\_s-y).\*X(:,2));

theta\_s=theta;

J\_all(i)=Computecost(X, Y, theta);

end

J\_all;

end

# Feature normalization

function [X\_norm,mu,sigma]=Featurenorm(X)

X\_norm=X;

mu=zeros(1,size(X,2));

sigma=zeros(1,size(X,2));

mu=mean(X);

sigma=std(X);

X\_norm=(X-repmat(mu,size(X,1),1))./repmat(sigma,size(X,1),1);

end

# Compute cost for multiple variables

function J = Computecostmult(X, Y, theta)

m = length(Y);

J=0;

J = sum((X\*theta-Y).^2)/2\*m;

end

# Gradient descent for multiple variables

function [theta,J\_all]=Gradientdescentmult(X,Y,theta,alpha,all\_iters)

m = length(Y);

J\_all=zeros(all\_iters,1);

for i=1:all\_iters

theta=theta-alpha/m\*X'\*(X\*theta-Y);

J\_all(i)=Computecostmult(X, Y, theta);

end

end