

Machine Learning Online Class - Exercise 1: Linear Regression

exercise1: Warm up exercise

exercise2: Compute cost for one variable

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 variable

```
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
```