计算机学院实验报告

实验题目： 机器学习实验二——多元线性回归		学号：202100130052
日期：2023.3.7	班级：21 智能	姓名：刘欣月
Email：202100130052@mail.sdu.edu.cn		

实验目的：实验二提供多元线性回归，使用梯度下降和正态方程研究多元线性回归，并且观察，代价函数 J(θ)和梯度下降收敛性和学习率α之间的关系。

实验语言介绍： matlab

实验步骤：

1. 下载数据：将数据导入变量 x,y 中。

```
x=load('ex2x.dat');
y=load('ex2y.dat');
```

2. 数据预处理，
   该数据为房价关于住房面积，卧室数量的相关数据，住房面积相当于卧室数量的 1000 倍左右，所以我们需要对数据进行标准化处理。

```
m=length(x);
x=[ones(m,1),x];
sigma=std(x);%标准差
mu=mean(x);%均值
%按标准差进行缩放
x(:,2)=(x(:,2)-mu(2))./sigma(2);
x(:,3)=(x(:,3)-mu(3))./sigma(3);
```

3. 梯度下降
   你对一个单变量回归问题实现了梯度下降。唯一的区别是矩阵 x 中还有一个特征， 假设函数仍然是：

$$h_\theta(x) = \theta^T x = \sum_{i=0}^{n} \theta_i x_i$$

   批量梯度下降更新规则是:

$$\theta_j := \theta_j - a\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \quad (\text{for all } j)$$

4. 选择学习率
   通过进行初始选择、运行梯度下降和观察损失函数，并相应地调整学习率来实现这一点。 回想一下，损失函数被定义为:

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

损失函数也可以写成以下向量化的形式:

$$J(\theta)=\frac{1}{2m}(X\theta-\vec{y})^T(X\theta-\vec{y})$$

首先我们设置学习率α为0.1，进行五十次迭代，

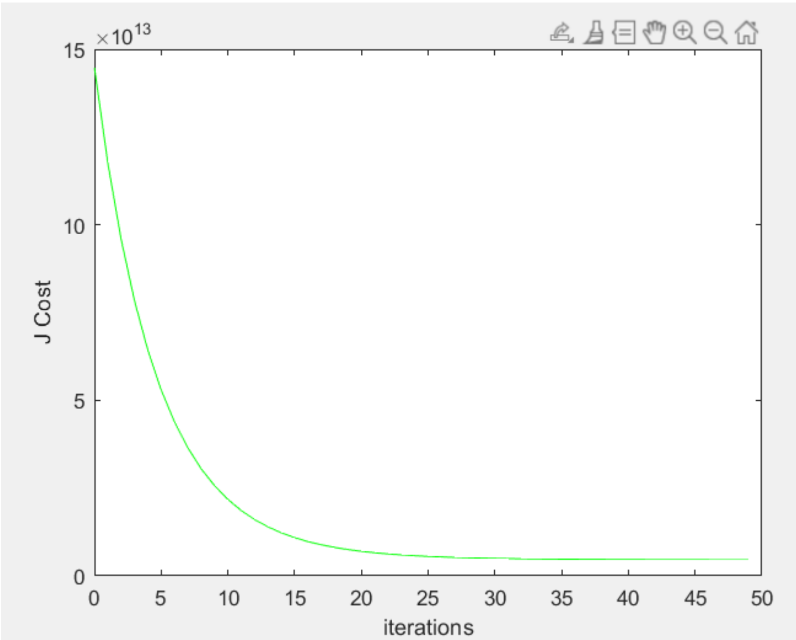
```
theta=zeros(size(x(1,:)))';%1*n
alpha=0.1;
J=zeros(50,1);

for num_iterations = 1:50
    J(num_iterations) = 0.5*m*(x*theta-y)'*(x*theta-y);

    theta = theta-alpha/m*x'*(x*theta-y);
end

figure;
plot(0:49,J(1:50),'g-');
xlabel('iterations');
ylabel('J Cost');
```

得到结果如下:



经尝试了三个不同的 alpha 值，并将成 本存储在里面 J1、J2 和 J3

```
theta = zeros(size(x(1,:)))'; % initialize fitting
parameters
alpha = 0.01;%% Your initial learning rate %%
```

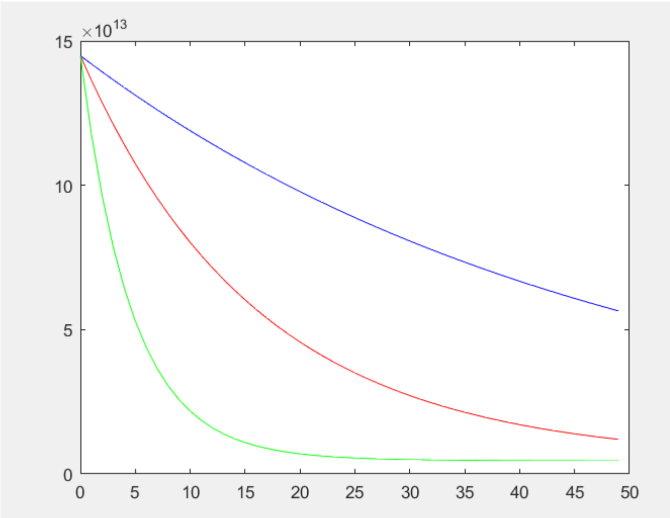
```
    J1 = zeros(50, 1);
    %迭代 50 次
    for num_iterations = 1:50
        J1(num_iterations) = 0.5*m*(x*theta-y)'*(x*theta-y);

        theta = theta-alpha/m*x'*(x*theta-y);
    end
    theta = zeros(size(x(1,:)))'; % initialize fitting
parameters
    alpha = 0.03;%% Your initial learning rate %%
    J2= zeros(50, 1);
    %迭代 50 次
    for num_iterations = 1:50
        J2(num_iterations) = 0.5*m*(x*theta-y)'*(x*theta-y);

        theta = theta-alpha/m*x'*(x*theta-y);
    end

    theta = zeros(size(x(1,:)))'; % initialize fitting
parameters
    alpha = 0.1;%% Your initial learning rate %%
    J3= zeros(50, 1);
    %迭代 50 次
    for num_iterations = 1:50
        J3(num_iterations) = 0.5*m*(x*theta-y)'*(x*theta-y);

        theta = theta-alpha/m*x'*(x*theta-y);
    end
    plot(0:49, J1(1:50), 'b-');
    hold on;
    plot(0:49, J2(1:50), 'r-');
    plot(0:49, J3(1:50), 'k-');
```
结果如下：



回答下列问题：

a. 观察随着学习率的变化，损失函数的变化。当学习率太小时会发生什么？太大呢？

b. 使用你发现的最佳学习率，运行梯度下降，直到收敛以找到

(A) θ 的最终值

(B) 一套 1650 平方英尺和 3 间卧室的房子的预测价格。 当你做出这个预测时，不要忘记缩放你的特征.

计算当学习率等于 0.1,迭代 50 次的 θ 值和预测的 Y 值：

```
theta = zeros(size(x(1,:)))'; % initialize fitting arameters
alpha = 0.1;%% Your initial learning rate %%

for num_iterations = 1:50
    theta = theta-alpha/m*x'*(x*theta-y);
end
format long
disp(theta);
predic_X = [1,(1650- mu(2))/ sigma(2),(3 - mu(3))/sigma(3)];
predic_Y = predic_X*theta;
disp(predic_Y);
```

```
>> ex2
   1.0e+05 *

   3.386582492492952
   1.041275155968978
  -0.001722053337078

     2.927480852321537e+05
```

5. 正态方程。

在正规方程的中，你学过最小二乘拟合的封闭解是

$$\theta = \left(X^T X\right)^{-1} X^T \vec{y}.$$

使用这个公式不需要任何特征缩放，而且你将在一次计算中得到一个精确的解:不存在像梯度下降那样的“直到收敛为止的循环”。

a.在你的程序中，用上面的公式计算θ。 请记住，虽然你不需要缩放你的功能，但你仍然需要添加一个拦截项。

b.一旦你从这个方法中找到θ，就用它来预测一个 1650 平方英尺的房子，有 3 间卧室。 你得到的价格和你通过梯度下降发现的一样吗？

```
h=@(x,theta) x*theta;
u=(x'*x)\x'*y;
disp(u);
t1=(1650-mu(2))./sigma(2);
t2=(3-mu(3))./sigma(3);
disp(h([1,t1,t2],u));
```

```
  1.0e+05 *

  3.404126595744681
  1.106310502788461
 -0.066494742708198

    2.930814643348962e+05
```