在本练习中，您将实现线性回归，并看到它在数据上工作。在开始编程之前，我们强烈建议您观看视频讲座并完成相关主题的评论问题。要开始练习，您需要下载启动器  
代码并将其内容解压缩到希望完成练习的目录中。如果需要，在开始本练习之前，请使用Octave / MATLAB中的cd命令切换到此目录。您还可以在课程网站的“环境安装说明”中找到有关安装Octave / MATLAB的说明。  
  
ex1.m - Octave / MATLAB脚本，用于指导您完成练习  
ex1 multi.m - Octave / MATLAB脚本为后续部分的练习  
ex1data1.txt - 具有一个变量的线性回归的数据集  
ex1data2.txt - 具有多个变量的线性回归的数据集  
submit.m - 将您的解决方案发送到我们的服务器的提交脚本  
warmUpExercise.m - Octave / MATLAB中的简单示例函数  
plotData.m - 显示数据集的功能  
computeCost.m - 计算线性回归成本的函数  
gradientDescent.m - 运行梯度下降的功能  
computeCostMulti.m - 多个变量的成本函数  
gradientDescentMulti.m - 多个变量的渐变下降  
featureNormalize.m - 功能规范化功能  
normalEqn.m - 计算正态方程的函数  
指示您需要完成的文件  
表示可选运动  
  
在整个练习中，您将使用脚本ex1.m和ex1 multi.m。这些脚本为问题设置数据集，并调用您将要编写的函数。你不需要修改任何一个。您只需要按照此作业中的说明修改其他文件中的功能。对于这个编程练习，你只需要完成第一个  
用一个变量实现线性回归的一部分练习。练习的第二部分是可选的，包括线性回归  
多个变量。

本课程中的练习使用Octave1或MATLAB，这是一种非常适合数值计算的高级编程语言。 如果您没有安装Octave或MATLAB，请参阅课程网站的“环境设置说明”中的安装说明。 在Octave / MATLAB命令行中，键入帮助后跟功能名称显示内置函数的文档。 例如，帮助情节将提供绘图的帮助信息。 Octave功能的进一步文档可以在Octave文档页面找到。 MATLAB文档可以在MATLAB文档页面找到。 我们也强烈鼓励使用在线讨论与其他学生讨论练习。 但是，请不要看别人写的任何源代码或与他人共享源代码。

1 简单的Octave/Matlab函数

ex1.m的第一部分给你练习Octave / MATLAB语法和作业提交过程。 在文件warmUpExercise.m中，您可以找到Octave/ MATLAB函数的大纲。 修改它返回一个5 x  
5单位矩阵填写下列代码：

A = eye(5);

完成后，运行ex1.m（假设您在正确的目录中，在Octave / MATLAB提示符下键入“ex1”），您应该看到类似于以下内容的输出：

ans =

Diagonal Matrix

1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1

现在ex1.m将暂停，直到您按任意键，然后将运行下一部分作业的代码。 如果你想退出，键入ctrl-c将在程序运行的中间停止程序。

1.1提交  
解决方案完成练习的一部分后，您可以提交您的解决方案  
通过在Octave / MATLAB命令行输入提交进行分级。 提交脚本将提示您登录电子邮件和提交令牌，并询问您要提交哪些文件。 您可以从网页获取作业的提交令牌。  
您现在应该提交您的解决方案。  
您可以多次提交您的解决方案，我们将只考虑最高分。  
  
2线性回归与一个变量  
在本练习的这一部分中，您将使用一个变量实现线性回归，以预测食品卡车的利润。 假设你是餐厅特许经营的首席执行官，正在考虑不同的城市打开一个新的出口。 这个连锁店已经在各个城市拥有卡车，并且有数据可以获得利润和城市人口。

您希望使用此数据来帮助您选择要扩展到下一个城市。 文件ex1data1.txt包含我们线性回归问题的数据集。 第一列是一个城市的人口，第二列是该市的食品卡车的利润。 利润的负值表示亏损。  
ex1.m脚本已经设置为为您加载此数据。

2.1绘制数据  
在开始任务之前，通过可视化来了解数据通常是有用的。 对于此数据集，您可以使用散点图来显示数据，因为它只有两个要绘制的属性（利润和人口）。 （在现实生活中遇到的许多其他问题是多维的，不能在二维图上绘制。）在ex1.m中，数据集从数据文件加载到变量X和Y中：

data = load('ex1data1.txt'); % read comma separated data

X = data(:, 1); y = data(:, 2);

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

接下来，脚本调用plotData函数来创建数据的散点图。 你的工作是完成plotData.m绘制情节; 修改文件并填写以下代码：

plot(x, y, 'rx', 'MarkerSize', 10); % Plot the data

ylabel('Profit in $10,000s'); % Set the y−axis label

xlabel('Population of City in 10,000s'); % Set the x−axis label

现在，当您继续运行ex1.m时，我们的最终结果应如图1所示，具有相同的红色“x”标记和轴标签。 要了解有关绘图命令的更多信息，您可以在Octave / MATLAB命令提示符下键入帮助图，或在线搜索绘图文档。 （要将标记更改为红色“x”，我们使用“rx”选项与plot命令，即plot（..，[您的选项]，..，'rx'）;）

图1：训练数据的散点图

2.2梯度下降

在这部分中，您将使用梯度下降将线性回归参数θ适用于我们的数据集。

2.2.1更新方程  
线性回归的目的是最小化成本函数

J(θ)

其中假设hθ（x）由线性模型给出

**hθ(x) = θTx**

回想一下，您的模型的参数是θj值。 这些是您将调整以最小化成本J（θ）的值。 一种方法是使用批量梯度下降算法。 在批量梯度下降中，每次迭代执行更新

**θj**

随着梯度下降的每一步，您的参数θj将更接近实现最低成本J（θ）的最优值。  
  
实现注意：我们将每个示例作为一行存储在Octave / MATLAB中的X矩阵中。 为了考虑截距项（θ0），我们向X添加一个额外的第一列，并将其设置为全部。 这允许我们将θ0视为简单的另一个“特征”。

2.2.2实施  
在ex1.m中，我们已经设置了线性回归的数据。 在以下行中，我们为我们的数据添加了另一个维度来适应θ0截距项。 我们还将初始参数初始化为0，将学习率α初始化为0.01。

**X = [ones(m, 1), data(:,1)]; % Add a column of ones to x**

**theta = zeros(2, 1); % initialize fitting parameters**

**iterations = 1500;**

**alpha = 0.01;**

2.2.3计算成本J（θ）  
当您执行梯度下降以学习最小化成本函数J（θ）时，通过计算成本来监视收敛是有帮助的。 在本节中，您将实现一个函数来计算J（θ），以便您可以检查梯度下降实现的收敛。  
您的下一个任务是完成文件computeCost.m中的代码，这是一个计算J（θ）的函数。 正如你所做的那样，记住变量X和Y不是标量值，而是行代表训练集的例子的矩阵。  
完成功能后，ex1.m中的下一步将使用初始化为0的θ运行computeCost，您将看到打印到屏幕的成本。  
你应该期望看到32.07的成本。  
  
您现在应该提交您的解决方案

2.2.4梯度下降  
接下来，您将在文件gradientDescent.m中实现渐变下降。循环结构已经为您编写，您只需要在每次迭代中向θ提供更新。  
在您编程时，请确保您了解您正在尝试优化的内容以及正在更新的内容。请记住，成本J（θ）由向量θ参数化，而不是X和y。也就是说，我们通过改变矢量θ的值而不是通过改变X或Y来最小化J（θ）的值。如果您不确定，请参阅本讲义中的方程式和视频讲座。  
验证梯度下降是否正确工作的一个好方法是查看J（θ）的值，并检查它是否随着每个步骤而减少。 gradientDescent.m的启动器代码在每次迭代时调用computeCost并打印成本。假设您已经正确地实现了梯度下降和computeCost，则J（θ）的值不应该增加，并且应该在算法结束时收敛到一个稳定的值。  
完成后，ex1.m将使用您的最终参数绘制线性拟合。结果应该如图2所示：  
您的θ的最终价值也将用于对35,000和7万人的利润进行预测。请注意，ex1.m中的以下行使用矩阵乘法而不是显式求和或循环来计算预测。这是Octave / MATLAB中的代码向量化示例。  
  
您现在应该提交您的解决方案

predict1 = [1, 3.5] \* theta;

predict2 = [1, 7] \* theta;

2.3调试  
在实现渐变下降时，请注意以下几点：  
•Octave / MATLAB数组索引从一个开始，而不是零。 如果您将θ0和θ1存储在一个称为θ的向量中，则值将为theta（1）和theta（2）。  
  
•如果在运行时看到许多错误，请检查矩阵运算，以确保您正在添加并乘以兼容维度的矩阵。 使用size命令打印变量的维度将有助于您调试。

图2：使用线性回归拟合训练数据

•默认情况下，Octave / MATLAB将数学运算符解释为矩阵运算符。 这是大小不兼容错误的常见来源。 如果不需要矩阵乘法，则需要添加“点”符号以将其指定为Octave / MATLAB。 例如，A \* B进行矩阵乘法，而A. \* B执行元素乘法。

2.4可视化J（θ）  
为了更好地了解成本函数J（θ），现在将绘制θ0和θ1值的二维网格上的成本。 您不需要为此部分编写任何新内容，但您应该了解您已经编写的代码是如何创建这些映像的。  
在ex1.m的下一步中，使用您编写的computeCost函数，设置了一个代码值来计算J（θ）的代码。

**% initialize J vals to a matrix of 0's**

**J vals = zeros(length(theta0 vals), length(theta1 vals));**

**% Fill out J vals**

**for i = 1:length(theta0 vals)**

**for j = 1:length(theta1 vals)**

**t = [theta0 vals(i); theta1 vals(j)];**

**J vals(i,j) = computeCost(x, y, t);**

**end**

**end**

执行这些行后，将有一个二维数组的J（θ）值。 脚本ex1.m然后将使用这些值使用冲浪和轮廓命令来生成J（θ）的曲面和轮廓图。 地块应如图3所示：

图3：成本函数J（θ）

这些图的目的是告诉你J（θ）如何随着θ0和θ1的变化而变化。 成本函数J（θ）是碗状，具有全局最小值。 （这在轮廓图中比在3D曲面图中更容易看到）。 这个最小值是θ0和θ1的最优点，梯度下降的每一步移动到这一点附近。

可选练习  
如果您已成功完成上述材料，请恭喜！ 您现在可以了解线性回归，并且应该可以在自己的数据集上开始使用它。  
  
对于其余的编程练习，我们包括以下可选练习。 这些练习将帮助您更深入地了解材料，如果您能够这样做，我们也鼓励您完成。

3具有多个变量的线性回归  
在这一部分，您将通过多个变量实现线性回归，以预测房屋的价格。 假设你是卖房子，你想知道什么是好的市场价格。 要做到这一点的一个方法是首先收集近期出售房屋的信息，并做出房价模型。  
文件ex1data2.txt包含俄勒冈州波特兰的一套房屋价格。 第一列是房子的大小（平方英尺），第二列是卧室的数量，第三列是房子的价格。  
已经设置了ex1 multi.m脚本，以帮助您逐步完成此练习。

3.1功能规范化  
ex1 multi.m脚本将从此数据集加载和显示一些值开始。 通过查看价值观，请注意，房屋面积约为卧室数量的1000倍。 当特征不同数量级时，首次执行特征缩放可使梯度下降更快地收敛。  
  
你的任务是完成featureNormalize.m中的代码  
  
•从数据集中减去每个要素的平均值。  
  
•减去平均值后，另外按其各自的“标准偏差”对特征值进行标尺（除）。

标准偏差是一种测量特定特征值范围内多少变化的方法（大多数数据点将在平均值的±2标准偏差内）;这是取值范围（max-min）的替代方案。在Octave / MATLAB中，您可以使用“std”函数来计算标准偏差。例如，在featureNormalize.m中，数量X（：，1）包含训练集中x1（房屋大小）的所有值，因此std（X（：，1））计算房屋尺寸的标准偏差。在调用featureNormalize.m时，与x0 = 1对应的1的额外列尚未添加到X中（详见ex1 multi.m）。  
  
您将对所有功能执行此操作，并且您的代码应使用所有大小（任意数量的功能/示例）的数据集。注意，矩阵X的每一列对应于一个特征。  
  
您现在应该提交您的解决方案。  
  
实现注意：在对特征进行归一化时，存储用于归一化的值 - 用于计算的平均值和标准偏差很重要。在从模型中学习参数后，我们经常想预测我们以前没有看到的房子的价格。给定一个新的x值（客厅面积和卧室数），我们必须首先使用我们以前从训练集计算的平均值和标准偏差来归一化x。

3.2梯度下降  
以前，您对单变量回归问题实施了梯度下降。 唯一的区别是矩阵X中还有一个特征。假设函数和批量梯度下降更新规则保持不变。  
  
您应该完成computeCostMulti.m和gradientDescentMulti.m中的代码  
实现具有多个变量的线性回归的成本函数和梯度下降。 如果上一部分的代码（单变量）已经支持多个变量，那么也可以在这里使用。  
  
确保您的代码支持任意数量的功能，并且向量化。 您可以使用“size（X，2）”来了解数据集中存在多少功能。  
  
您现在应该提交您的解决方案。

实现注意：在多变量情况下，成本函数也可以以下列矢量化形式写入：  
![图片包含 文字, 白板

已生成极高可信度的说明]()  
  
当您使用数字计算工具（如Octave / MATLAB）时，矢量化版本是高效的。如果您是矩阵操作的专家，您可以向自己证明这两种形式是等效的。  
  
3.2.1可选（未分级）练习：选择学习率  
  
在本练习的这一部分中，您将尝试为数据集尝试不同的学习率，并找到快速收敛的学习率。您可以通过修改ex1 multi.m并更改设置学习率的代码部分来更改学习率。  
ex1 multi.m的下一个阶段将调用您的gradientDescent.m函数，并以选定的学习速率运行梯度下降约50次迭代。该函数还应返回向量J中J（θ）值的历史。在最后一次迭代之后，ex1 multi.m脚本绘制了J值与迭代次数的关系。  
  
如果您选择学习率在一个很好的范围内，你的情节看起来就像图4.如果你的图形看起来很不一样，特别是如果你的J（θ）值增加甚至爆炸，调整你的学习率，再试一次。我们建议在对数刻度上以大约3倍于先前值（即0.3,0.1,0.03,0.01等）的乘法步骤来尝试学习速率α的值。您可能还需要调整运行的迭代次数，以帮助您查看曲线中的整体趋势。

图4：具有适当学习速率的梯度下降趋同

实施注意：如果您的学习率太大，J（θ）可能会发散并“爆炸”，导致计算机计算过大的值。 在这些情况下，Octave / MATLAB将倾向于返回NaN。 NaN代表“不是数字”，并且通常由涉及-∞和+∞的未定义的操作引起。

Octave / MATLAB提示：为了比较不同的学习学习率如何影响收敛，在同一数字上绘制几个学习率的J是有帮助的。 在Octave / MATLAB中，可以通过在地块之间执行“hold on”命令多次执行梯度下降来完成。 具体来说，如果您尝试了三种不同的alpha值（您应该尝试比此更多的值），并将成本存储在J1，J2和J3中，则可以使用以下命令在同一图上绘制：

**plot(1:50, J1(1:50), ‘b’);**

**hold on;**

**plot(1:50, J2(1:50), ‘r’);**

**plot(1:50, J3(1:50), ‘k’);**

注意随着学习率的变化，收敛曲线的变化。 以较小的学习率，您应该发现梯度下降需要很长时间才能收敛到最佳值。 相反，学习速度很快，梯度下降可能不会收敛甚至可能发散！  
使用您发现的最佳学习率，运行ex1 multi.m脚本运行渐变下降直到收敛以找到θ的最终值。 接下来，使用这个值θ来预测1650平方英尺的房子和3间卧室的价格。 您稍后将使用值来检查正常方程的实现。 做这个预测时不要忘记规范化你的功能！  
您不需要为这些可选（未分级）练习提交任何解决方案。

3.3正常方程在演讲视频中，您了解到线性回归的闭式解决方案是

θ=

使用此公式不需要任何特征缩放，您将在一个计算中得到一个精确的解决方案：像渐变下降中没有“循环直到收敛”。  
在normalEqn.m中完成代码以使用上面的公式来计算θ。 请记住，虽然您不需要扩展功能，但我们仍然需要向X矩阵添加一列1以具有截距项（θ0）。 ex1.m中的代码将为您添加1的列。  
  
您现在应该提交您的解决方案。  
  
可选（未分级）练习：现在，一旦您使用这种方法找到θ，就可以使用它为一间拥有3间卧室的1650平方英尺的房子进行价格预测。 您应该发现，使用与梯度下降拟合的模型获得的值相同（在3.2.1节中）。

提交和评分  
完成作业的各个部分后，请务必使用提交功能系统将解决方案提交给我们的服务器。 以下是打破这个练习的每个部分得分。