

# 编程练习 1:线性回归

## 机器学习

### 介绍

在本练习中，您将实现线性回归，并了解它对数据的作用。在开始编程练习之前，我们强烈建议观看视频讲座，并完成相关主题的复习题。

要开始练习，您需要下载启动代码并将其内容解压缩到您希望完成练习的目录中。如果需要，在开始这个练习之前，使用 Octave/MATLAB 中的 `cd` 命令切换到这个目录。

您也可以在课程网站的“环境设置说明”中找到安装 Octave/MATLAB 的说明。

### 本练习中包含的文件

`ex1.m` - Octave/MATLAB 脚本，指导您完成练习 `ex1_multi.m` - Octave/MATLAB 脚本，用于练习后面的部分  
`ex1data1.txt` -用于单变量线性回归的数据集  
`ex1data2.txt` -用于多变量线性回归的数据集  
`submit.m` -将您的解决方案发送到我们的服务器的提交脚本  
[A] `warmupexercise . m`-octave/MATLAB 中的简单示例函数[A]  
`plotData.m` -用于显示数据集的函数  
[A] 计算线性回归成本的 `computeCost.m` 函数  
[A] `gradientDescent.m` -运行梯度下降的函数  
[`computeCostMulti . m` --多变量的成本函数  
[`+gradientDescentMulti . m`-多个变量的梯度下降  
[`+feature normalize . m`-归一化特征的函数  
[`+normal eqn . m`-用于计算法线方程的函数

A 表示您需要完成的文件  
表示可选练习

在整个练习中，您将使用脚本 `ex1.m` 和 `ex1_multi.m`。这些脚本为问题建立数据集，并调用您将要编写的函数。您不需要修改它们中的任何一个。你只需要按照本作业中的说明修改其他文件中的函数。对于本编程练习，您只需完成练习的第一部分，即可实现一个变量的线性回归。练习的第二部分是可选的，包括多个变量的线性回归。

## 从哪里获得帮助

本课程中的练习使用八度音阶<sup>1</sup> 或者 MATLAB，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，请参考课程网站“环境设置说明”中的安装说明。

在 Octave/MATLAB 命令行中，键入 `help` 后跟一个函数名会显示内置函数的文档。例如，“帮助绘图”将显示绘图的帮助信息。关于 Octave 函数的更多文档可以在 [Octave 文档页面](#)。MATLAB 文档可以在 [MATLAB 文档页面](#)。

我们也强烈鼓励使用在线讨论与其他学生讨论练习。但是，不要看别人写的任何源代码，也不要和别人分享你的源代码。

---

## 1 简单的倍频程/MATLAB 函数

`ex1.m` 的第一部分让你练习 Octave/MATLAB 语法和作业提交过程。在 `warmUpExercise.m` 文件中，您将找到 Octave/MATLAB 函数的概要。通过填写以下代码，将其修改为返回一个 5 x 5 的单位矩阵：

```
1 Octave 是 MATLAB 的免费替代品。对于编程练习，您可以自由使用 Octave 或 MATLAB。 Oct
```

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` ”)，您应该会看到类似下面的输出：

```
ans =  
  
对角矩阵  
  
   -   0   0   0   0  
   0   -   0   0   0  
   0   0   -   0   0  
   0   0   0   -   0  
   0   0   0   0   -
```

现在 `ex1.m` 将暂停，直到你按下任何键，然后将运行赋值的下一部分的代码。如果你想退出，键入 `ctrl-c` 将会在程序运行过程中停止程序。

## 1.1 提交解决方案

完成部分练习后，您可以通过在 Octave/MATLAB 命令行中键入 `submit` 来提交您的解决方案以进行评分。子任务脚本将提示您输入登录电子邮件和提交令牌，并询问您要提交哪些文件。您可以从作业的网页获取提交令牌。

*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 一元线性回归

在本练习的这一部分中，您将使用一个变量来实现线性回归，以预测食品车的利润。假设你是一家特许经营餐厅的首席执行官，正在考虑在不同的城市开设新的分店。该连锁店已经在各个城市有了卡车，而且你有了这些城市的利润和人口数据。

您希望使用这些数据来帮助您选择下一个要扩展到的城市。

文件 `ex1data1.txt` 包含用于线性回归问题的数据集。第一列是一个城市的人口，第二列是该城市一辆快餐车的利润。利润为负值表示亏损。

`ex1.m` 脚本已经设置好为您加载这些数据。

## 2.1 绘制数据

在开始任何任务之前，通过可视化来理解数据通常是有用的。对于此数据集，您可以使用散点图来可视化数据，因为它只有两个属性可以绘制（利润和人口）。（你在现实生活中会遇到的许多其他问题是多维的，无法在二维图上绘制出来。）

在 `ex1.m` 中，数据集从数据文件加载到变量 `X` 中和 `y`：

---

```
data = load('ex1data1.txt'); % 读取逗号分隔的数据
X = data(:, 1); y = data(:, 2); m = length(y); % 培训示例的百分比
```

---

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` 来绘制绘图；修改文件并填入以下代码：

---

```
plot(x, y, 'rx', 'MarkerSize', 10); y_label = '绘制数据在 1 万美元以内'; % 设置 y 轴标签
xlabel('以万为单位的城市人口'); % 设置 x 轴标签
```

---

现在，当您继续运行 `ex1.m` 时，我们的最终结果应该如图所示 1，具有相同的红色 “x” 标记和轴标签。

要了解有关 `plot` 命令的更多信息，可以在 Octave/MATLAB 命令提示符下键入 `help plot`，或者在线搜索绘图文档。（为了将标记更改为红色 “x”，我们将选项 “rx” 与 `plot` 命令一起使用，即 `plot(.., [您可以在这里选择], .., 'rx');`）

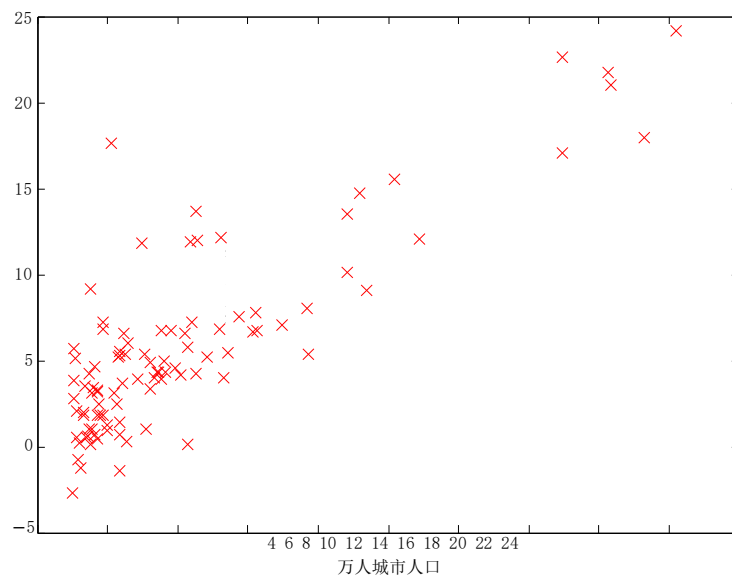


图 1: 训练数据的散点图

## 2.2 梯度下降

在这一部分，您将使用梯度下降将线性回归参数  $\theta$  拟合到我们的数据集。

### 2.2.1 更新方程式

线性回归的目标是最小化成本函数

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

其中假设  $h_{\theta}(x)$  由线性模型给出

$$h_{\theta}(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

回想一下，模型的参数是  $\theta_j$  值。这些是您将调整以最小化成本  $J(\theta)$  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

$$\theta := \theta - \alpha (h(x) - y) x$$

(同时更新所有  $j$  的  $\theta$ )。

随着梯度下降的每一步，参数  $\theta_j$  越来越接近实现最低成本  $J(\theta)$  的最佳值。

### 2.2.2 履行

实现说明：我们将每个例子作为一行存储在 Octave/MATLAB 的  $X$  矩阵中。为了考虑截距项 ( $\theta_0$ )，我们在  $X$  上增加了一个额外的第一列，并将其设置为全数据。这使得我们可以将  $\theta_0$  视为另一个简单的特征，以适应  $\theta_0$  截距项。我们还将初始参数初始化为 0，并将学习率  $\alpha$  初始化为 0.01。

$X = [\text{ones}(m, 1), \text{data}(:, 1)];$  %将一列 1 加到  $x$   $\theta = 0(2, 1);$  %初始化拟合参

### 2.2.3 计算成本 $J(\theta)$

当您执行梯度下降来学习最小化成本函数  $J(\theta)$  时，通过计算成本来监控收敛是有帮助的。在本节中，您将实现一个计算  $J(\theta)$  的函数，以便检查梯度下降实现的收敛性。

您的下一个任务是完成文件 `computeCost.m` 中的代码，这是一个计算  $J(\theta)$  的函数。当您这样做时，请记住变量  $X$  和  $y$  不是标量值，而是矩阵，其行表示来自定型集的示例。

一旦完成了函数，`ex1.m` 中的下一步将使用初始化为零的  $\theta$  运行 `computeCost` 一次，您将看到成本打印到屏幕上。

你应该会看到 32.07 英镑的成本。

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` 中实现梯度下降。已经为您编写了循环结构，您只需要在每次迭代中提供对  $\theta$  的更新。

当你编程的时候，确保你明白你要优化的是什么，更新的是什么。请记住，成本  $J(\theta)$  是由向量  $\theta$  确定的，而不是  $X$  和  $y$ 。也就是说，我们通过改变向量  $\theta$  的值来最小化  $J(\theta)$  的值，而不是改变  $X$  或  $y$ 。如果不确定，请参考讲义中的等式和视频讲座。

验证梯度下降是否正常工作的好方法是查看  $J(\theta)$  的值，并检查它是否随每步下降。`gradientDescent.m` 的起始代码在每次迭代时调用 `computeCost` 并打印成本。假设您已经正确实现了梯度下降和 `computeCost`，您的  $J(\theta)$  值应该永远不会增加，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。结果应该类似于图 2：

$\theta$  的最终值也将用于预测 35,000 人和 70,000 人地区的利润。请注意 `ex1.m` 中的下面几行使用矩阵乘法，而不是显式求和或循环来计算预测。这是 Octave/MATLAB 中代码矢量的一个例子。

您现在应该提交您的解决方案。

```
predict1 = [1, 3.5]*theta;  
% 预测 = [1 7]*theta.
```

## 2.3 排除故障

实施梯度下降时，请记住以下几点：

- Octave/MATLAB 数组索引从 1 开始，而不是零。如果你把  $\theta_0$  和  $\theta_1$  存储在一个叫做  $\theta$  的向量中，那么值将是  $\theta(1)$  和  $\theta(2)$ 。
- 如果您在运行时看到许多错误，请检查您的矩阵运算，以确保您正在添加和乘以兼容维度的矩阵。用 `size` 命令打印变量的尺寸将有助于调试。

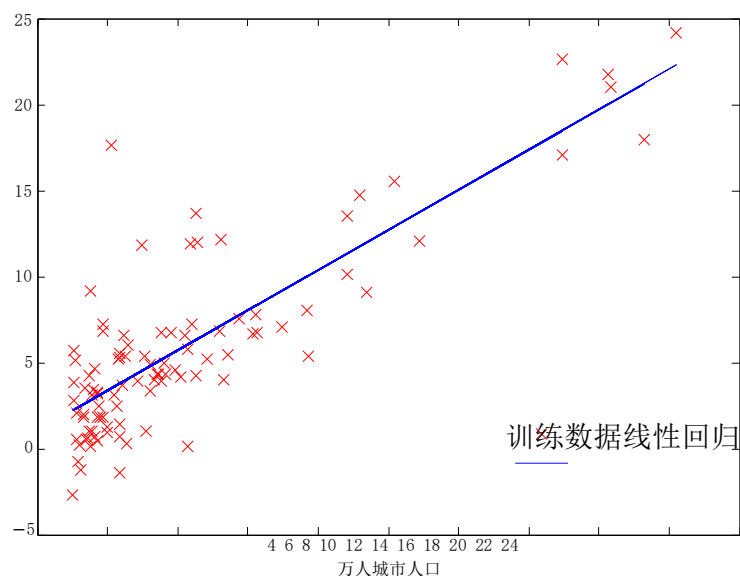


图 2:线性回归拟合的训练数据

- 默认情况下，Octave/MATLAB 将数学运算符解释为矩阵运算符。这是尺寸不兼容错误的一个常见来源。如果你不想要矩阵乘法，你需要添加“点”符号来指定 Octave/MATLAB。例如， $A*B$  执行矩阵乘法，而  $A.*B$  执行元素级乘法。

## 2.4 可视化 $J(\theta)$

为了更好地理解成本函数  $J(\theta)$ ，现在将在  $\theta_0$  和  $\theta_1$  值的二维网格上绘制成本。您不需要为这个部分编写任何新的代码，但是您应该理解您已经编写的代码是如何创建这些图像的。

在 `ex1.m` 的下一步中，使用您编写的 `computeCost` 函数设置代码来计算数值网格上的  $J(\theta)$ 。



%将 J 值初始化为 0 的矩阵

```
J_vals = zeros(长度(theta_0_vals), 长度(theta_1_vals));
```

%填写 J 值

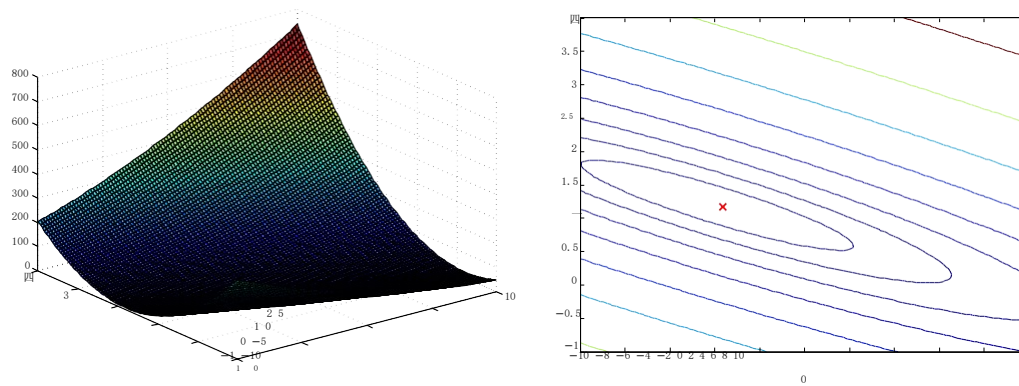
```
对于 i = 1:长度(theta_0_vals) 对于 j = 1:长度  
(theta_1_vals)
```

```
    t = [theta_0_vals(i); theta_1_vals(j)]; J_vals(i, j) = computeCost(x, y, t);
```

结束

结束

执行完这些行之后，您将得到一个  $J(\theta)$  值的二维数组。然后，脚本 `ex1.m` 将使用 `surf` 和 `contour` 命令使用这些值生成  $J(\theta)$  的表面和等值线图。这些情节应该看起来像图 3：



(a) 表面 (b) 轮廓，显示最小值

图 3: 成本函数  $J(\theta)$

这些图表的目的是向您展示  $J(\theta)$  如何随着  $\theta_0$  和  $\theta_1$  的变化而变化。成本函数  $J(\theta)$  是碗形的，并且具有全局最小值。（这在等高线图中比在 3D 表面图中更容易看到）。这个最小值就是  $\theta_0$  和  $\theta_1$  的最优点，梯度下降的每一步都向这个点靠拢。

## 可选练习

如果你已经成功完成了上面的材料，恭喜你！您现在已经理解了线性回归，应该能够开始在自己的数据集上使用它了。

对于本编程练习的其余部分，我们包括了以下可选练习。这些练习将帮助你对材料有更深入的理解，如果你能做到，我们鼓励你也完成它们。

---

## 3 多元线性回归

在这一部分中，您将实现多个变量的线性回归来预测房价。假设你正在出售你的房子，你想知道一个好的市场价格是多少。一种方法是首先收集最近出售的房屋信息，并制作一个房价模型。

文件 `ex1data2.txt` 包含俄勒冈州 Port- land 房价的训练集。第一栏是房子的大小(平方英尺)，第二栏是卧室的数量，第三栏是房子的价格。

`ex1_multi.m` 脚本可帮助您逐步完成本练习。

### 3.1 特征标准化

`ex1_multi.m` 脚本将从加载和显示该数据集中的一些值开始。通过查看这些值，注意到房子的大小大约是卧室数量的 1000 倍。当特征按大小顺序不同时，首先执行特征缩放可以使梯度下降收敛得更快。

您在这里的任务是完成 `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，将特征值除以各自的“标准偏差”

标准差是一种测量特定特性的值的范围内有多少变化的方法(大多数数据点将位于平均值的 2 个标准差内);这是取值范围(最大-最小)的替代方法。在 Octave/MATLAB 中,可以使用“std”函数来计算标准差。例如,在 featureNormalize.m 中,数量  $X(:, 1)$  包含训练集中  $x_1$ (房屋大小)的所有值,因此  $\text{std}(X(:, 1))$  计算房屋大小的标准偏差。在 featureNormalize.m 被调用时,对应于  $x_0 = 1$  的 1 的额外列还没有被添加到  $X$  中(参见 ex1\_multi.m 的详情)。

您将对所有要素执行此操作,并且您的代码应适用于所有大小的数据集(任意数量的要素/示例)。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。

**3.2 梯度下降** 实施注意:当标准化特征时,重要的是存储用于标准化的值-用于计算的平均值和标准偏差。在从模型中学习了参数之后,我们经

以前,您在单变量问题中,我们以前没有见过的房子的价格是给定  $x$  值多了一个特征,假设函数和梯度下降更新规则不变。我们必须首先使用之前从训练集中计算出的平均值和标准偏差对  $x$  进行归一化。

您应该完成 computeCostMulti.m 和 gradientDescentMulti.m 中的代码,以实现线性回归的成本函数和梯度下降多个变量。如果前一部分中的代码(单个变量)已经支持多个变量,您也可以在这里使用它。

确保您的代码支持任意数量的特性,并且是良好矢量化的。您可以使用“ $\text{size}(X, 2)$ ”来确定数据集有多少个要素。

您现在应该提交您的解决方案。

---

实施说明：在多变量情况下，成本函数也可以写成以下矢量化形式：

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

在哪里

$$\begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix}^T = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix}^T \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$\rightarrow \mathbf{y}$      $y$  (米)。

当你处理数字时，矢量化版本是有效的  
像 Octave/MATLAB 这样的计算工具。如果你是矩阵运算的专家，你可以证明这两种形式是等价的。

---

### 3.2.1 可选(未评分)练习:选择学习率

在练习的这一部分，您将尝试数据集的不同学习率，并找到一个快速收敛的学习率。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` 的下一阶段将调用 `gradientDescent.m` 函数，并以选定的学习速率运行大约 50 次梯度下降迭代。该函数还应该返回向量中  $J(\theta)$  值的历史记录  
J. 在最后一次迭代之后，`ex1_multi.m` 脚本根据迭代次数绘制 J 值。

如果你在一个很好的范围内选择了一个学习率，你的图看起来会很相似 4. 如果你的图表看起来非常不同，特别是如果你的  $J(\theta)$  值增加甚至爆炸，调整你的学习速度，再试一次。我们建议在对数标度上尝试学习率  $\alpha$  的值，乘法步长约为先前值的 3 倍（即 0.3、0.1、0.03、0.01 等）。如果有助于看到曲线的整体趋势，您可能还想调整正在运行的迭代次数。

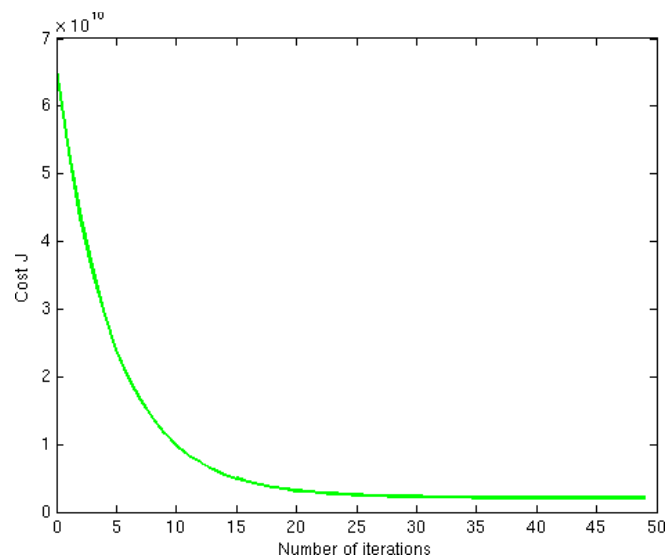


图 4: 具有适当学习速率的梯度下降收敛

实施注意：如果你的学习率太大， $J(\theta)$  会发散和“爆炸”，导致计算机计算的值太大。在这些情况下，

---

**Octave/MATLAB 提示：**比较不同的学习方式如何学习

速率会影响收敛，在同一个图上画出几个学习速率的  $J$  是很有帮助的。在 Octave/MATLAB 中，这可以通过绘图 `(1:50, J1(1:50), 'b')`；

坚持住；

情节 `(1:50, J2(1:50), 'r')`；

情节 `(1:50, JBOY3 乐队(1:50), 'k')`；

---

最后的参数 “b”、“r” 和 “k” 为  
剧情。

注意当学习率改变时收敛曲线的变化。在学习率很小的情况下，你应该会发现梯度下降需要非常长的时间才能收敛到最优值。相反，在学习率很大的情况下，梯度下降可能不收敛，甚至可能发散！

使用您找到的最佳学习率，运行 `ex1_multi.m` 脚本运行梯度下降，直到收敛，以找到  $\theta$  的最终值。接下来，用  $\theta$  的这个值来预测一栋 1650 平方英尺、3 间卧室的房子的价格。稍后您将使用 `value` 来检查法线方程的实现。做这个预测的时候别忘了把你的特征正常化！

*对于这些可选的(未评分的)练习，您不需要提交任何答案。*

### 3.3 正规方程

在讲座视频中，您了解到线性回归的封闭解是

$$\theta = (X^T X)^{-1} X^T y。$$

使用这个公式不需要任何特征缩放，一次计算就能得到精确解：不存在像梯度下降中那样的“循环直到收敛”。

完成 `normalEqn.m` 中的代码，使用上面的公式计算  $\theta$ 。请记住，虽然您不需要缩放要素，但我们仍然需要向  $X$  矩阵添加一列 1，以获得截距项 ( $\theta_0$ )。 `ex1.m` 中的代码会将 1 的列添加到  $X$  中。

*您现在应该提交您的解决方案。*

*可选(未评分)练习:现在，一旦你用这种方法找到  $\theta$ ，就用它来预测一栋 1650 平方英尺、有 3 间卧室的房子的价格。您应该会发现，给出的预测价格与使用梯度下降模型拟合获得的价格相同(在第一节中)3.2.1)。*

## 提交和分级

完成作业的各个部分后，请务必使用提交功能系统将您的解决方案提交到我们的服务器。下面是这个练习的每一部分是如何评分的。

部分	提交的文件	点
热身运动	warmUpExercise.m	10 分
计算一个变量的成本	computeCost.m	40 分
单变量梯度下降	gradientDescent.m	50 分
总分		100 分

### 可选练习

部分	提交的文件	点
特征标准化	featureNormalize.m	0 分
计算多个变量的成本	计算机科学多媒体	0 分
多变量梯度下降		0 分
正规方程	正常设备	0 分

您可以多次提交您的解决方案，我们将只考虑最高分。