

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。在开始此编程练习之前,我们强烈建议您观看视频讲座并完成相关主题的复习题。

要开始练习,您需要下载起始代码并将其内容解压缩到您希望完成练习的目录。如果需要,在开始本练习之前使用 Octave/MATLAB 中的 `cd` 命令切换到该目录。

您还可以在课程网站的“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 - 指导您完成练习的 Octave/MATLAB 脚本
ex1_multi.m - 练习后面部分的 Octave/MATLAB 脚本
ex1data1.txt - 具有一个变量的线性回归数据集
ex1data2.txt - 具有线性回归的数据集多个变量
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 - 显示数据集的函数
[?] computeCost.m - 函数计算线性回归的成本
[?] gradientDescent.m - 运行梯度下降的函数
[+] computeCostMulti.m - 多个变量的成本函数
[+] gradientDescentMulti.m - 多个变量的梯度下降
[+] featureNormalize.m - 函数归一化特征
[+] normalEqn.m - 计算正规方程的函数

?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`。这些脚本为问题设置数据集并调用您将编写的函数。您不需要修改其中任何一个。您只需按照本作业中的说明修改其他文件中的函数。

对于此编程练习,您只需完成练习的第一部分即可使用一个变量实现线性回归。练习的第二部分是可选的,涵盖了具有多个变量的线性回归。

在哪里获得帮助

本课程的练习使用 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` 后跟一个函数名称会显示内置函数的文档。例如,`help plot` 会调出绘图的帮助信息。Octave 函数的更多文档可以在[Octave 文档页面找到](#)。MATLAB 文档可以在[MATLAB 文档页面中找到](#)。

我们也强烈鼓励使用在线讨论与其他学生讨论练习。但是,请勿查看他人编写的任何源代码或与他人共享您的源代码。

1 个简单的 Octave/MATLAB 函数

`ex1.m` 的第一部分让您练习 Octave/MATLAB 语法和作业提交过程。在文件 `warmUpExercise.m` 中,您将找到 Octave/MATLAB 函数的概要。通过填写以下代码修改它以返回一个 5 x 5 单位矩阵:

```
A = 眼睛 (5) ;
```

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

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”) ,您应该会看到输出类似于以下内容:

```

年 =

对角矩阵

    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,然后运行代码对于下一部分的作业。如果你想退出,输入 ctrl-c 将在运行过程中停止程序。

1.1 提交解决方案

完成部分练习后,您可以提交解决方案。通过在 Octave/MATLAB 命令行输入 submit 进行评分。提交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。您可以获得提交来自网页的用于分配的令牌。

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,我们将采取只考虑最高分。

2 一个变量的线性回归

在本练习的这一部分中,您将使用一个变量来预测食品卡车的利润。假设你是一家公司的 CEO 餐厅特许经营,并正在考虑在不同的城市开设新的出口。该连锁店已经在各个城市拥有卡车,您有数据来自城市的利润和人口。

您想使用这些数据来帮助您选择下一步要扩展的城市。

文件 `ex1data1.txt` 包含我们的线性回归问题的数据集。第一列是一个城市的人口,第二列是该城市的食品卡车的利润。利润为负值表示亏损。

`ex1.m` 脚本已设置为为您加载此数据。

2.1 绘制数据

在开始任何任务之前,通过可视化数据来理解数据通常很有用。对于此数据集,您可以使用散点图来可视化数据,因为它只有两个要绘制的属性(利润和人口)。(您在现实生活中会遇到的许多其他问题是多维的,无法绘制在二维图上。)

在 `ex1.m` 中,数据集从数据文件加载到变量 `X` 和 `y` 中:

```
数据=加载( 'ex1data1.txt' ); X = 数据(:, 1); y          % 读取逗号分隔的数据
= 数据(:, 2); m = 长度(y);                               % 训练样例数
```

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` 来绘制情节;修改文件,填写如下代码:

```
情节(x,y, 'rx', 'MarkerSize', 10);          % 绘制数据
ylabel( '10,000 美元的利润' );               % 设置y轴标签xlabel( 'Population of City
in 10,000s' ); % 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类似于图 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,您可以在 Octave/MATLAB 命令提示符下键入 `help plot` 或在线搜索绘图文档。(为了将标记更改为红色的“x”,我们将选项 `rx` 与 `plot` 命令一起使用,即 `plot(...,[your options here],..., 'rx');`)

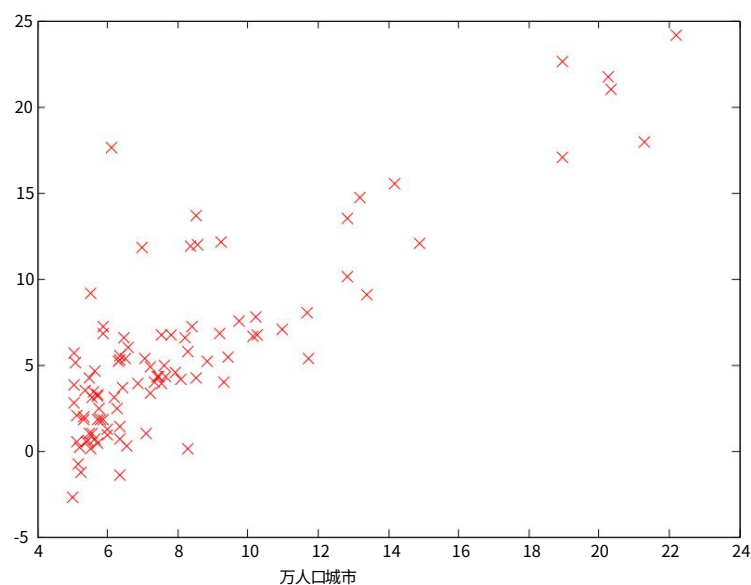


图 1:训练数据的散点图

2.2 梯度下降

在这一部分中,您将线性回归参数 θ 拟合到我们的数据集使用梯度下降。

2.2.1 更新方程

线性回归的目标是最小化成本函数

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

其中假设 $h_{\theta}(x)$ 由线性模型给出

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

回想一下,您的模型的参数是 θ_j 值。这些是您将调整的值以最小化成本 $J(\theta)$ 。一种方法是使用批量梯度下降算法。在批量梯度下降中,每个迭代执行更新

$$1 \theta_j := \theta_j - \alpha \sum_{i=1}^m x_i (h_{\theta}(x^{(i)}) - y^{(i)}) x_j \quad (\text{同时更新所有 } j \text{ 的 } \theta_j)。$$

随着梯度下降的每一步,您的参数 θ_j 更接近将实现最低成本 $J(\theta)$ 的最优值。

实施说明:我们将每个示例作为一行存储在 Octave/MATLAB 的 X 矩阵中。为了考虑截距项(θ_0),我们向 X 添加了额外的第一列并将其设置为全1。这使我们可以将 θ_0 视为简单的另一个“特征”。

2.2.2 实施

在 ex1.m 中,我们已经为线性回归设置了数据。在以下几行中,我们为数据添加了另一个维度以适应 θ_0 截距项。我们还将初始参数初始化为 0,将学习率 alpha 初始化为 0.01。

```
X = [ones(m, 1), data(:,1)]; % 添加一列到 x
theta = zeros(2, 1); % 初始化拟合参数

迭代次数 = 1500;阿尔法 = 0.01;
```

2.2.3 计算代价 $J(\theta)$

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很有帮助的。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。在执行此操作时,请记住变量 X 和 y 不是标量值,而是行表示训练集中示例的矩阵。

完成函数后,ex1.m 中的下一步将使用初始化为零的 θ 运行一次 computeCost,您将看到成本打印到屏幕上。

您应该会看到成本为 32.07。

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,您只需在每次迭代中提供对 θ 的更新。

在您进行编程时,请确保您了解您要优化的内容以及正在更新的内容。请记住,成本 $J(\theta)$ 由向量 θ 参数化,而不是 X 和 y 。也就是说,我们通过改变向量 θ 的值来最小化 $J(\theta)$ 的值,而不是通过改变 X 或 y 。如果您不确定,请参阅本手册中的方程式和视频讲座。

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着每一步而减小。
`gradientDescent.m` 的起始代码在每次迭代时调用 `computeCost` 并打印成本。假设你已经正确地实现了梯度下降和 `computeCost`,你的 $J(\theta)$ 值永远不会增加,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来绘制线性拟合。结果应该类似于图 2:您的 θ 最终值也将用于预测 35,000 和 70,000 人区域的利润。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的方式。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

在实现梯度下降时,请记住以下几点:

- Octave/MATLAB 数组索引从一开始,而不是零。如果您将 θ_0 和 θ_1 存储在称为 θ 的向量中,则值将是 $\theta(1)$ 和 $\theta(2)$ 。
- 如果您在运行时发现许多错误,请检查您的矩阵运算以确保您正在添加和相乘兼容维度的矩阵。使用 `size` 命令打印变量的尺寸将帮助您进行调试。

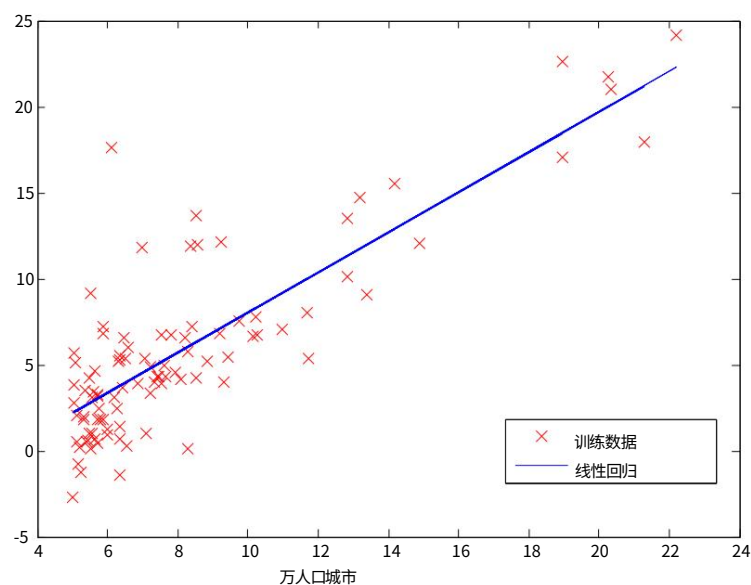


图 2:使用线性回归拟合的训练数据

- 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常见来源。如果你不想矩阵乘法,你需要添加“点”符号
将此指定给 Octave/MATLAB。比如 $A*B$ 做一个矩阵乘法,而 $A.*B$ 进行逐元素乘法。

2.4 可视化 $J(\theta)$

为了更好地理解成本函数 $J(\theta)$,您现在将成本绘制在 θ_0 和 θ_1 值的二维网格。您无需编写任何代码
这部分是新的,但您应该了解您编写的代码是如何编写的
已经在创建这些图像。

在 `ex1.m` 的下一步中,设置了代码来计算 A 上的 $J(\theta)$
使用您编写的 `computeCost` 函数的值网格。


```

% 将 J vals 初始化为 0 的矩阵
J_vals = zeros(length(theta0_vals), length(theta1_vals));

% 填写 J 值
对于 i = 1:length(theta0_vals)
    对于 j = 1:length(theta1_vals) t = [theta0_vals(i); theta1
        vals(j)];
        J_vals(i,j) = 计算成本(x, y, t);
    结束
结束

```

执行完这些行后,您将获得 $J(\theta)$ 值的二维数组。
 然后脚本 ex1.m 将使用这些值使用 surf 和 contour 命令生成 $J(\theta)$ 的曲面和等高线图。这些图应该类似于图 3:

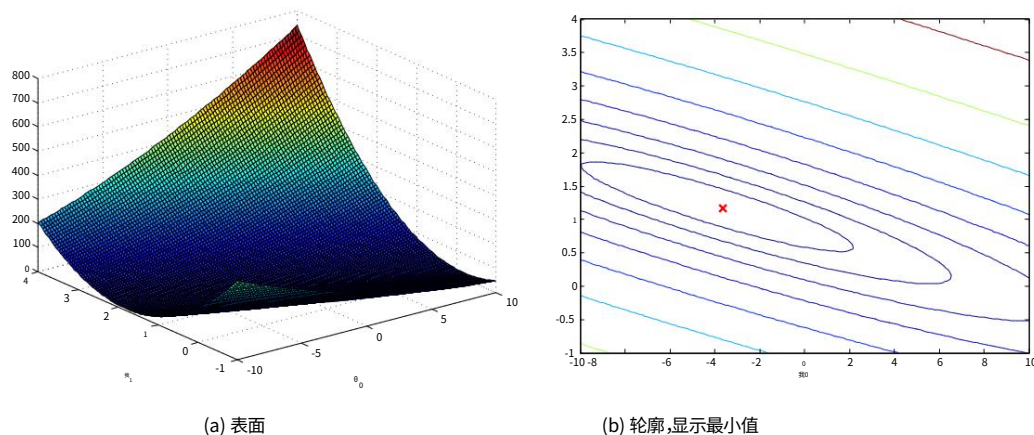


图 3:成本函数 $J(\theta)$

这些图表的目的是向您展示 $J(\theta)$ 如何随着 θ_0 和 θ_1 的变化而变化。成本函数 $J(\theta)$ 是碗形的,具有全局最小值。(这在等高线图中比在 3D 曲面图中更容易看到)。这个最小值是 θ_0 和 θ_1 的最佳点,梯度下降的每一步都向这个点靠拢。

可选练习

如果您已成功完成上述材料,恭喜!您现在了解线性回归并且应该能够开始在您自己的数据集上使用它。

对于本编程练习的其余部分,我们包括了以下可选练习。这些练习将帮助您更深入地了解材料,如果您能够这样做,我们鼓励您也完成它们。

3 多变量线性回归

在这一部分中,您将使用多个变量实现线性回归来预测房价。假设您要卖掉您的房子,你想知道一个好的市场价格是多少。一种方法是首先收集有关最近售出房屋的信息并制作房价模型。

文件 `ex1data2.txt` 包含俄勒冈州波特兰市房价的训练集。第一列是房子的大小(平方英尺),第二列是卧室数量,第三列是房子的价格。

`ex1_multi.m` 脚本已设置为帮助您逐步完成此练习。

3.1 特征归一化

`ex1_multi.m` 脚本将首先加载和显示此数据集中的一些值。通过查看这些值,请注意房屋面积大约是卧室数量的 1000 倍。当特征相差几个数量级时,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` 中的代码以

- 从数据集中减去每个特征的平均值。
- 减去平均值后,额外缩放(除)特征值通过它们各自的“标准偏差”。

标准偏差是一种衡量特定特征值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 标准偏差内）；这是取值范围（max-min）的替代方法。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))` 计算房屋大小的标准差。

在调用 `featureNormalize.m` 时，对应于 `x0 = 1` 的额外列 1 尚未添加到 `X`（详见 `ex1_multi.m`）。

您将对所有功能执行此操作，并且您的代码应该适用于所有大小的数据集（任意数量的功能/示例）。请注意，矩阵 `X` 的每一列对应一个特征。

您现在应该提交您的解决方案。

实施注意事项：当对特征进行归一化时，存储用于归一化的值很重要。用于计算的平均值和标准偏差。在从模型中学习参数之后，我们经常想要预测我们以前没见过的房子的价格。给定一个新的 `x` 值（客厅面积和卧室数量），我们必须首先使用我们之前从训练集中计算的均值和标准差对 `x` 进行归一化。

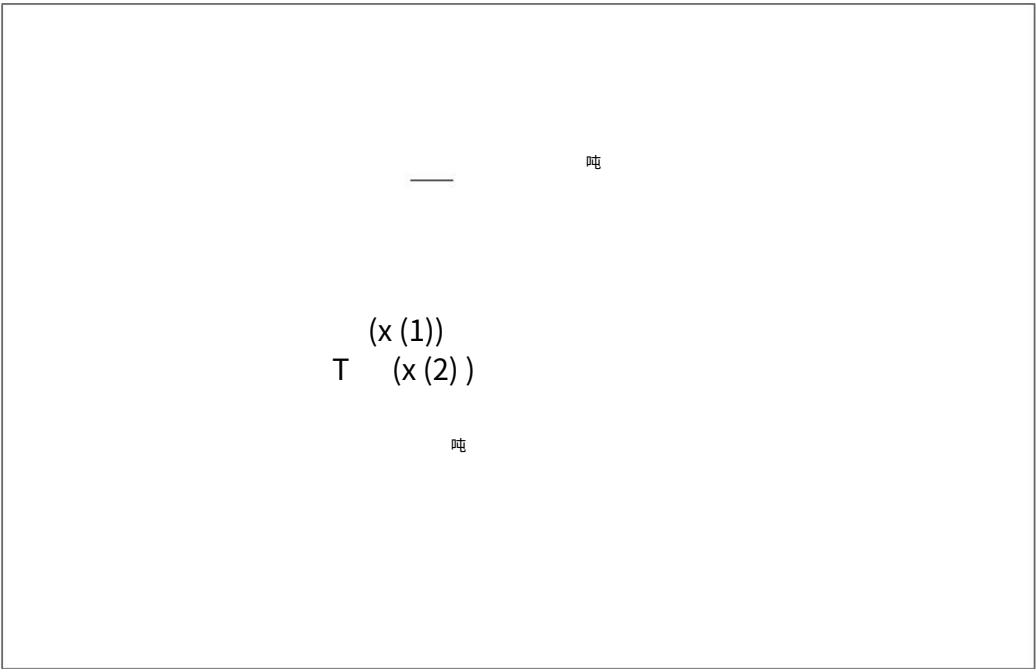
3.2 梯度下降

之前，您在单变量回归问题上实现了梯度下降。现在唯一的区别是矩阵 `X` 中多了一个特征。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` 中的代码实现多变量线性回归的成本函数和梯度下降。如果您在上一部分（单个变量）中的代码已经支持多个变量，您也可以在此处使用它。

确保您的代码支持任意数量的功能并且向量化良好。您可以使用 “`size(X, 2)`” 来找出数据集中存在多少特征。

您现在应该提交您的解决方案。



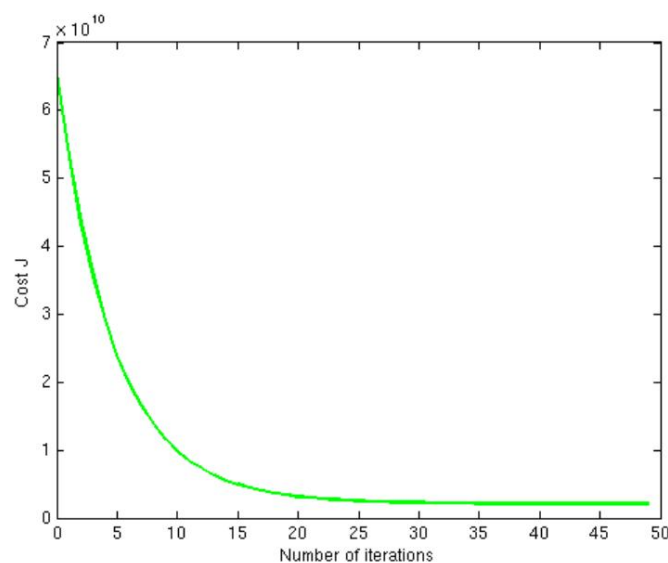


图 4:梯度下降与适当学习率的收敛

实施注意事项:如果您的学习率太大, $J(\theta)$ 可能会发散并“炸毁”,导致值对于计算机计算来说太大。在这些情况下,Octave/MATLAB 将倾向于返回 NaN。NaN 代表“非数字”,通常由涉及 $-\infty$ 和 $+\infty$ 的未定义运算引起。

Octave/MATLAB 提示:要比较不同的学习率如何影响收敛,在同一个图上绘制多个学习率的 J 会很有帮助。在 Octave/MATLAB 中,这可以通过在绘图之间使用“hold on”命令多次执行梯度下降来完成。具体来说,如果您尝试了三个不同的 α 值(您可能应该尝试更多的值)并将成本存储在 $J1$ 、 $J2$ 和 $J3$ 中,您可以使用以下命令将它们绘制在同一个图上:

```

plot(1:50,J1(1:50), 'b');
hold on;
plot(1:50,J2(1:50), 'r');
hold on;
plot(1:50,J3(1:50), 'k');

```

最后的参数“b”、“r”和“k”为绘图指定不同的颜色。

注意随着学习率的变化收敛曲线的变化。

使用较小的学习率,您应该会发现梯度下降需要很长时间才能收敛到最优值。相反,如果学习率很大,梯度下降可能不会收敛甚至发散!

使用您找到的最佳学习率,运行 `ex1_multi.m` 脚本以运行梯度下降直到收敛以找到 θ 的最终值。接下来,使用这个 θ 值来预测 1650 平方英尺和 3 间卧室的房子的价格。稍后您将使用 `value` 来检查正规方程的实现。当你做出这个预测时,不要忘记标准化你的特征!

您无需为这些可选 (未评分) 练习提交任何解决方案。

3.3 正规方程

在讲座视频中,您了解到线性回归的封闭形式解决方案是

$$\theta = (X^T X)^{-1} X^T y$$

使用这个公式不需要任何特征缩放,并且您将在一次计算中得到一个精确的解决方案:没有像梯度下降那样的“循环直到收敛”。

完成 `normalEqn.m` 中的代码以使用上面的公式计算后期 θ 。请记住,虽然您不需要缩放特征,但我们仍然需要在 X 矩阵中添加一列 1 以获得截距项(θ_0)。

`ex1.m` 中的代码将为您将 1 列添加到 X 中。

您现在应该提交您的解决方案。

可选 (未分级) 练习:现在,一旦您使用此方法找到 θ ,请使用它来预测一栋 1650 平方英尺的 3 间卧室房屋的价格。您应该会发现它给出的预测价格与您使用梯度下降拟合模型获得的值相同 (在第 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

功能系统将您的解决方案提交到我们的服务器。以下是一个

此练习的每个部分的评分方式的细分。

部分	提交文件	积分
热身运动	warmUpExercise.m	10 分
计算一个变量的成本	计算成本.m	40 点
一个变量的梯度下降	gradientDescent.m	50 点
总积分		100 分

可选练习		
部分	提交文件	积分
特征归一化	特征标准化.m	0 分
计算多个成本	多计算成本	0 分
变量		
多个梯度下降	gradientDescentMulti.m	0 分
变量		
正规方程	正态方程	0 分

您可以多次提交您的解决方案,我们将采取
只考虑最高分。