# 编程练习1：线性回归

本练习中，您将实施线性回归并查看它对数据的处理。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

要开始练习，您需要下载初学者代码并将其内容解压到您希望完成练习的目录。 如果需要，请在开始本练习之前使用Octave中的cd命令切换到此目录。

## 本练习中包含的文件

ex1.m - matlab脚本，将有助于指导您完成练习

ex1 multi.m - matlab脚本，用于后面拓展部分的练习（后四个）

ex1data1.txt - Dataset for linear regression with one variable

ex1data2.txt - Dataset for linear regression with multiple variables

submit.m - 提交您的解决方案到我们的服务器的提交脚本

warmUpExercise.m - Simple example function in Octave

plotData.m - Function to display the dataset显示数据集的功能

computeCost.m - Function to compute the cost of linear regression计算梯度下降的代价函数

gradientDescent.m - Function to run gradient descent运行梯度下降

computeCostMulti.m - Cost function for multiple variables用于多变量的计算梯度下降

gradientDescentMulti.m - Gradient descent for multiple variables用于多变量的运行梯度下降

featureNormalize.m - Function to normalize features特征值归一化

normalEqn.m - Function to compute the normal equations计算正规方程

# 编程练习2：逻辑回归

在本练习中，您将执行逻辑回归并将其应用于两个不同的数据集。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

ex2.m - Octave script that will help step you through the exercise

ex2 reg.m - Octave script for the later parts of the exercise

ex2data1.txt - Training set for the first half of the exercise

ex2data2.txt - Training set for the second half of the exercise

submit.m - Submission script that sends your solutions to our servers

mapFeature.m - 生成特征多项式的函数

plotDecisionBounday.m - 绘制分类器决策边界的函数

plotData.m - 绘制二维分类数据的函数

sigmoid.m – S型功能

costFunction.m – 逻辑回归代价函数

predict.m – 逻辑回归预测函数

costFunctionReg.m - 正则化的逻辑回归代价函数

在整个练习中，您将使用脚本ex2.m和ex2 reg.m。这些脚本为问题设置数据集并调用您将编写的函数。 你不需要修改其中的任何一个。 您只需按照本作业中的说明修改其他文件中的功能。

# 编程练习3：多类分类和神经网络

在本练习中，您将实施one-vs-all一对多逻辑回归和神经网络来识别手写数字。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

ex3.m - 脚本，将有助于指导您完成第1部分  
ex3\_nn.m - 脚本，有助于引导您完成第2部分  
ex3data1.mat -手写数字的训练样本  
ex3weights.mat - 神经网络的初始权重  
displayData.m - 帮助可视化数据集的程序  
fmincg.m - 函数最小化例程（类似于fminunc）  
sigmoid.m - Sigmoid函数，即S型函数

lrCostFunction.m - 逻辑回归代价函数  
oneVsAll.m - 训练一对多分类器  
predictOneVsAll.m - 使用一对多分类器进行预测  
predict.m - 神经网络预测函数

# 编程练习4：神经网络学习

在本练习中，您将实施神经网络的反向传播算法，并将其应用于手写数字识别任务。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

ex4.m - 脚本，将有助于指导您完成练习

ex4data1.mat – 用于训练的一组手写数字

ex4weights.mat - 练习的神经网络参数

displayData.m - 帮助可视化数据集的函数

fmincg.m – 代价函数最小化例程（高级优化算法）（类似于fminunc）

sigmoid.m - Sigmoid函数

computeNumericalGradient.m – 用于梯度检验中近似计算梯度矩阵

checkNNGradients.m - Function to help check your gradients确认梯度计算是否正确

debugInitializeWeights.m – 调试过程中（梯度检验）用于初始化权重的函数

predict.m - 神经网络预测函数

sigmoidGradient.m – 计算S型函数的梯度

randInitializeWeights.m – 随机初始化权重

nnCostFunction.m – 神经网络代价函数

# 编程练习5：正则化线性回归和偏差与方差

在本练习中，您将执行正则化线性回归并将其用于研究具有不同偏差 - 方差属性的模型。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

ex5.m - 脚本，将有助于指导您完成练习

ex5data1.mat - 数据集

featureNormalize.m - 特征缩放归一化函数

fmincg.m - 代价函数最小化例程（高级优化算法）（类似于fminunc）

plotFit.m - 绘制多项式拟合

trainLinearReg.m - 使用您的代价函数训练线性回归

linearRegCostFunction.m - 正则线性回归代价函数

learningCurve.m - 生成学习曲线

polyFeatures.m - 将数据映射到多项式特征空间

validationCurve.m - 生成交叉验证曲线

# 编程练习6：支持向量机

在本练习中，您将使用支持向量机（SVM）构建垃圾邮件分类器。 在开始编程练习之前，我们强烈建议您观看视频讲座并完成相关主题的复习问题。

ex6.m - 演习前半部分的脚本

ex6data1.mat - 示例数据集1

ex6data2.mat - 示例数据集2

ex6data3.mat - 示例数据集3

svmTrain.m - 支持向量机训练函数

svmPredict.m - 支持向量机预测函数

plotData.m - 绘制2D数据

visualizeBoundaryLinear.m - 绘制线性边界

visualizeBoundary.m - 绘制非线性边界

linearKernel.m - 用于SVM的线性核函数

gaussianKernel.m - 用于SVM的高斯核函数

dataset3Params.m - 用于数据集3的参数

ex6 spam.m - 练习后半部分的脚本

spamTrain.mat - 垃圾邮件训练样本集

spamTest.mat - 垃圾邮件测试样本集

emailSample1.txt - 示例电子邮件1

emailSample2.txt - 示例电子邮件2

spamSample1.txt - 垃圾邮件示例1

spamSample2.txt - 垃圾邮件样本2

vocab.txt - 词汇表

getVocabList.m - 加载词汇列表

porterStemmer.m – Stemming（枝干）函数

readFile.m - 将文件读入字符串

processEmail.m - 电子邮件预处理

emailFeatures.m - 从电子邮件中提取特征

# 编程练习7：K均值聚类和主成分分析

在本练习中，您将实现K均值聚类算法并将其应用于压缩图像。 在第二部分中，您将使用主成分分析来查找面部图像的低维表示。

ex7.m - K-means第一次练习的Octave / Matlab脚本

ex7 pca.m - 用于PCA（主成分分析）第二次练习的Octave / Matlab脚本

ex7data1.mat - PCA的示例数据集

ex7data2.mat - K-means的示例数据集

ex7faces.mat - 面对的数据集

bird small.png - 示例图像

displayData.m - 显示存储在矩阵中的2D数据

drawLine.m - 在一个现有的图上绘制一条线

plotDataPoints.m - K均值质心的初始化

plotProgresskMeans.m - 绘制K-means的每一步

runkMeans.m - 运行K-means算法

pca.m - 执行主成分分析

projectData.m - 将数据集投影到较低维空间

recoverData.m - 从投影中恢复原始数据

findClosestCentroids.m - 查找最接近的质心（用于K均值）

computeCentroids.m - 计算质心方法（用于K均值）

kMeansInitCentroids.m - K均值质心的初始化

# 编程练习8：异常检测和推荐系统

在本练习中，您将实施异常检测算法并将其应用于检测网络中发生故障的服务器。 在第二部分中，您将使用协作过滤为电影构建推荐系统。

ex8.m - 第一部分练习的Octave / Matlab脚本

ex8 cofi.m - 第二部分练习的Octave / Matlab脚本

ex8data1.mat -用于异常检测的第一个数据集

ex8data2.mat -用于异常检测的第二个示例数据集

ex8 movies.mat - 电影评论数据集

ex8 movieParams.mat - 提供用于调试的参数

multivariateGaussian.m - 计算高斯分布的概率密度函数

visualizeFit.m - 高斯分布和数据集的二维图

checkCostFunction.m - 协同过滤的渐变检查

computeNumericalGradient.m -计算数字渐变

fmincg.m - 函数最小化例程（类似于fminunc）

loadMovieList.m - 将电影列表加载到单元阵列中

电影ids.txt - 电影列表

normalizeRatings.m - 协作过滤的均值归一化

estimateGaussian.m--估计具有对角协方差矩阵的高斯分布的参数

selectThreshold.m - 查找异常检测的阈值

cofiCostFunc.m - 实施协作过滤的代价函数