编程练习 3：

多类分类和神经网络

机器学习

# 介绍

在本练习中，您将实现一个与所有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 recommend观看 the视频讲座并完成相关主题的复习问题。

To 开始练习，you 您需要下载初学者代码并将其内容解压缩到you 您希望完成练习的目录中。如果需要，在cd 启动此exercise之前，请使用Octave/MATLAB中的 cd命令更改为此目录 exe。

find instructions您还可以 also在课程网站的 the “"en- virave/MATLAB"中找到安装 installing八度/MATLAB的说明。

## 本练习中包含的文件

ex3.m - Octave/MATLAB script that steps you through part 1 ex3 nn.m - Octave/MATLAB script that steps you through part 2 ex3data1.mat - Training set of hand-written digits ex3weights.mat - Initial weights for the neural network exercise

提交.m - 提交脚本，将您的解决方案发送到我们的服务器

displayData.m - 有趣的拐点，以帮助可视化数据集fmincg.m - 函数最小化例程（类似于 fminunc） sigmoid.m - Sigmoid 函数

[*\**] lrCost函数.m - 逻辑回归成本函数

[*\**] oneVsAll.m - 训练一个与所有多级分类器

[*\**] 预测在eVAll.m 上 - 使用一个与全部的多分类器预测

[*\**] 预测.m - 神经网络预测功能

*•*指示您需要完成的文件

Throughout the exercise, you will be using the scripts ex3.m and ex3 nn.m. These scripts set up the dataset for the problems and make calls to functions that you will write. You do not need to modify these scripts. You are only required to modify functions in other files, by following the instructions in this assignment.

## 在哪里获得帮助

本课程中的练习使用八角形 1[1](#_bookmark0)或 MATLAB，这是一种高级程序明语言，非常适合数值计算。如果您you 没有have 安装八度或MATLAB，请参阅课程网站的"环境设置说明"中的安装说明。

在八角/MATLAB命令行，typing帮助，然后是 func-

tion名称显示内置函数的文档。例如，帮助绘图将显示用于绘图的帮助信息。有关八度扫描函数的进一步文档，请参阅found at the [八度文档页面](http://www.gnu.org/software/octave/doc/interpreter/)。MAT-LAB 文档可在be [MATLAB 文档页](http://www.mathworks.com/help/matlab/?refresh=true)中找到。

我们还强烈建议 encourage使用 the在线**讨论** to with与其他学生讨论前讨论。但是， do不要查看 at其他人编写by 的任何源代码 code或与他人共享您的源代码。

# 多类分类

在本练习中，您将使用逻辑回归和神经网络来识别手写数字（从 0 到 9）。自动手写数字识别今天被广泛使用 - 从识别邮件信封上的邮政编码（邮政编码）到识别银行支票上写入的金额。本练习将向您展示如何将所学的方法用于此分类任务。

In the first part of the exercise, you will extend your previous implemen- tion of logistic regression and apply it to one-vs-all classification.

1八度是 isMATLAB的 a免费替代品。对于 the编程练习，您可以 are自由使用八度或 MATLAB。

## 数据

在ex3data1.mat中为您提供一个数据集，其中包含 5000ples个手写数字的培训考试。[2](#_bookmark1) .mat 格式表示数据已以本机 Octave/MATLAB 矩阵格式保存，而不是文本 （ASCII） 格式（如 csv 文件）。通过使用加载命令将这些矩阵 c直接读取到程序中load 。加载后，程序的内存中将显示正确尺寸和值的矩阵。矩阵将被命名，因此您无需为其分配名称。

百分比从文件加载加载 sav ed 矩阵（"ex3data1.mat"）;); 负载（

百分比矩阵 X 和 y 现在将位于您的八度环境中

ex3data1.mat中有 5000 个训练示例，其中每个训练示例是数字的 20 像素x 20 像素灰度图像。每个像素由一by 个浮点数表示，指示该位置的灰度强度。 The 20 x 20 of像素网格被"展开"到 a 400 维矢量中。这些培训示例中的每一个都将成为row 数据矩阵 X 中的一行。这为我们提供了一个 5000 x 400 矩阵 X，其中每行都是手写数字图像的培训示例。

·

*X* | 

— (*x*(1))*T* —

— (*x*(2))*T* —

·

·

.

[*x*（*m*）T*T* =

训练集的第二部分是一个 5000 维矢量y，其中包含训练集的标签。为了使内容与Octave/MATLAB 索引（没有零索引）更兼容，我们将have 数字零映射到值 1。因此，"0"数字标记为"10"，而数字"1"到"9" are按 in their其自然顺序标记为 as"1"到"9"。

## 可视化数据

You will begin by visualizing a subset of the training set. In Part 1 of ex3.m, the code randomly selects selects 100 rows from X and passes those rows to the displayData function. This function maps each row to a 20 pixel by 20 pixel grayscale image and displays the images together. We have provided

2 这是 a MNIST手写数字数据集的子集This （http://yann.lecun.com/[http://yann.lecun.com/](http://yann.lecun.com/exdb/mnist/)  [exdb/mnist/](http://yann.lecun.com/exdb/mnist/)）。

显示数据功能，我们鼓励您检查代码以查看 how其工作原理。 you运行此步骤后，应 should看到 an如图 Figure [1](#_bookmark2)所示的图像。



图 1：数据集的示例

## 矢量化逻辑回归

will 您将使用多个一个与所有逻辑回归模型来构建多分类器。由于有 10 个类，you 您需要训练 10 个单独的逻辑回归分类器。为了使这种培训高效，请务必确保您的代码具有良好的矢量化。在本节中，you 您将实现 a that不 not为循环使用任何for 循环的矢量化逻辑回归版本。您可以在最后一个练习中使用代码作为本练习的起点。

### 向量化成本函数

我们将首先编写成本函数的矢量化版本。回想一下，在（非规范化）逻辑回归中，成本函数是

*米*

1 区［ (*i*) (*i*) (*i*) (*i*)

*米*

*Ⅰ*

*Ⅰ*

*.* ( ) log  *−*  *h* (*x* ))－ ))  *−* )  (*x*  *−*  *y* (1 (1 *θ* lo J （A） 和 g （h （ x ） - （1 x y ） g （1 x h （ x ） - .  g

要计算求和中的每个元素，我们必须计算*hθ*(*x*(*i*))

*i*±1

for every example *i*, where *hθ*(*x*(*i*)) = *g*(*θT x*(*i*)) and  *g*(*z*) = 1*−*

1+*e*

*z*

is the

sigmoid 函数。事实证明，我们可以通过使用矩阵乘法快速计算所有示例。让我们将*X* 和 *\** 定义为

·

*X* | 

— (*x*(1))*T* —

— (*x*(2))*T* —

·

·

.

[*x*（*m*）T*T* =

0  *=*1

和 *θ* =

·

·

·

·

*.*

·

.

*否。*

然后，通过计算矩阵产品*X+，*我们有

*X+* =

·

·

·

[*x*（1））*T*  *T =*

[*x*（2））*T*  *T =*

.

[*x*（*m*）T*T*  *=*



 

·

=

*[ TT* （x（*1）*1)]

—*θT* (*x*(2)) —

·

—*θT* (*x*(*m*)) —

*.*

.

在最后一个相等性中，我们使用了*aT*  *b*  = *bT*  *a* a a 和 *b* 是矢量的事实。这使我们能够计算产品*#T*  *x*（*i*） 的所有示例 *i* 在一行代码中。

你的工作是在 to the文件lrCostSins.m 中写入 the非规范化的成本函数，您的实现应该使用上面介绍的策略来编写到 calcu-

晚*=T*  *x*（*i*）. vectorized approach the use of the You should您还应 a对其余

成本函数。完全矢量化版本的lrCost函数.m 不应包含任何循环。

（提示：在编写此函数时，您可能需要使用元乘法运法（.\*） 和总运运和sum (

### 矢量化渐变

回想 t帽子（非规范化）逻辑回归成本的梯度是一个矢量，其中 *j*th 元素定义为

*我不得不*

*θj*我不得不

1

=

*米*

区

*米*

*i*=1

（(*hθ*(*x*(*i*)) *− y*(*i*))*x*(*i*)） *.*

为了在数据集上对此操作进行矢量化，我们从写出所有

*J*

所有*\*j*的部分导数明确，

*"J"* 

*米i*±1

我

·

（(*hθ*(*x*(*i*)) *− y*(*i*))*x*(*i*)） 

1

*±*0

*我不得不*

我

*[ ]*1 ]

*m*

*i*±1

*（h）θ*（x

)*−* *和*

我

·

*i*=1

（(*hθ*(*x*(*i*)) *− y*(*i*))*x*(*i*)）



0

.

）*x*2

·

(

*我不得不* 1

·

·

·

*±*2

.

·

*米*

·

·

=

·

- *米*

(*i*)

(*i*)

(*i*)） 

*我不得不*

*否。*

  L*m*

（(*h* (*x*(*i*)) *− y*(*i*))*x*(*i*)） 

*i*±1 *θ n*

.

*m*

\*（ ）

1

= (*hθ* (*x*(*i*)) *− y*(*i*))*x*(*i*)

*米*

*i*=1

1 *XT* （h  *（x）* *-* ( y *。*)*.*  （1）

其中

*m θ*

*hⅠ*(*x*(1))*− y*(1) 

·

*h* ( ) *−*  *x*  *y*(2) 2)（x （2） ） - y （2） - y

.

*.*

=  *y* *x*) (*θ* h - （x ） - y - "y"  

*hθ* (*x*(1)) *− y*(*m*) 

请注意*，x*（*ii*)）是矢量， 而（*h=*（*x*x ）y *y*（*i*)） 是一个标量 （单个数字）。

自 理解 最后 步 的 推导 让 *βi* = (*hθ*(*x*(*i*)) *y*(*i*))和观察 即：

*•*

区*βix*(*i*) = 

*•*

*i*

(*|*1) (*|*2) (*|m*)

*| | |*

*x x . . . .*  *x*

  *β*1

  .

*\**2

*μm*

 *T*

[ ] X   *=，*

其中值*=i* = （*h=*（*x*（ i*i*）= *−* y （*i*））

上面的表达式允许我们计算所有部分导数，没有任何循环。如果您you 对线性代数感到满意，我们鼓励encourage 您通过上面的矩阵乘法来说服自己矢量化版本执行相同的计算。现在，您应该now 实现公式 [1](#_bookmark3) 来计算正确的 correct矢量化梯度。完成后 you ，通过实现渐变完成函数 lrCost函数.m。

ation, while *θX* is not. Furthermore, if you have a non-vectorized version

of your code, you can compare the output of your vectorized code and non-vectorized code to make sure that they produce the same outputs.

trix *X* of size 100 *×* 20 (100 examples, 20 features) and *θ*, a vector with

dimensions 20 *×* 1, you can observe that *Xθ* is a valid multiplication oper-

**Debugging Tip:** Vectorizing code can sometimes be tricky. One com-

mon strategy for debugging is to print out the sizes of the matrices you are working with using the size function. For example, given a data ma-

### 矢量化正化逻辑回归

后 你 有 实现 矢 量化 对于 物流 回归 你 将 现在向成本函数添加正则。回想一下，对于规范化逻辑回归，t他的成本函数被定义 作为

*N*

1

*J*(*θ*) =

*米*

区［*−y*(*i*) log(*h* (*x*(*i*))) *−* (1 *− y*(*i*)) log(1 *− h* (*x*(*i*)))－ + *λ*

区*θ*2*.*

*j*=1

请注意，您应该*不*正在正规化*θ*0 用于偏置术语。

*米*

*i*±1

*Ⅰ*

*Ⅰ*

2*米*

*J*

相应地，正化逻辑回归成本的部分导数*θj* 定义 作为

*Ⅰ*

*J*

*J* *（）* 1

=

*米*

*2.1.1.2.*

*米*

*i*±1

区

0

区(*h* (*x*(*i*)) *− y*(*i*))*x*(*i*)

对于*J*= 0

*米*

*J* *（）*

=

*θj*我不得不

1

*米*

*i*=1

(*hθ* (*x*(*i*)) *− y*(*i*))*x*(*i*)

*λ*

• *m=j* j j *=*  *≥* 1

现在修改您的代码lrCost功能 以考虑正规化。

*J*

再次，不应将任何循环放入代码中。

**Octave/MATLAB Tip:** When implementing the vectorization for reg- ularized logistic regression, you might often want to only sum and update certain elements of *θ*. In Octave/MATLAB, you can index into the matri- ces to access and update only certain elements. For example, A(:, 3:5)

= B(:, 1:3) will replaces the columns 3 to 5 of A with the columns 1 to 3 from B. One special keyword you can use in indexing is the end keyword in indexing. This allows us to select columns (or rows) until the end of the matrix. For example, A(:, 2:end) will only return elements from the 2*nd* to last column of A. Thus, you could use this together with the sum and

.^ operations to compute the sum of only the elements you are interested in (e.g., sum(z(2:end).^2)). In the starter code, lrCostFunction.m, we have also provided hints on yet *another* possible method computing the regularized gradient.

*您现在应该提交您的解决方案。*

## 一与全分类

在本练习的这一部分中，您将通过训练多个定期逻辑回归分类器来实现一对全的分类，这些分类针对数据集中的每个 *K*类（图[1）。](#_bookmark2) 在手写数字数据集中of ，K  *K* = 10，但代码应适用于 *K*的任何值。

你应该现在在 中完成代码一VsAll.m 训练一个分类器 每个类。在特别是，您的代码应返回 a 矩阵 Θ R*K×*(*N* +1） , 其中 每个 行 的 Θ 对应 自 学

*·*

一个类的逻辑回归参数。 You您可以使用从 1 到*K* a的"for"循环执行此操作 with，单独训练每个分类器。

请注意，此函数的y 参数是标签从 1 到 10 的矢量，其中我们将数字"0"映射到标签 10（以避免avoid 索引混淆）。

当为类*k*  *k=*  *=*1*、 ...、 K=*训练分类 er 时，您需要一个*m-*尺寸矢量的标签 *y，*其中 *yj*  *=*  0*，1*指示j-thth 训练实例是否属于 k 类 *j* (*（jj* = 1），或者它是否属于其他

类*（yj* = 0）。您可能会发现逻辑数组有助于此任务。

a == b % You should try different values of b here

the expression a == b for a vector a (of size *m×*1) and scalar b will return

a vector of the same size as a with ones at positions where the elements

of a are equal to b and zeroes where they are different. To see how this works for yourself, try the following code in Octave/MATLAB:

a = 1:10; % Create a and b

b = 3;

**Octave/MATLAB Tip:** Logical arrays in Octave/MATLAB are arrays

which contain binary (0 or 1) elements. In Octave/MATLAB, evaluating

此外，您将使用fmincg 进行此练习（而不是 fminunc）。).

fmincg的工作原理 similarly与 fminunc类似，但 is处理大量参数的效率 for更高 more。

正确完成一VsAll.m的代码后，脚本

ex3.m will将继续 to使用你的一个VsAll功能来训练一个多级的CLas-sifier。 sifier.

*您现在应该提交您的解决方案。*

### 一个与一种预测

训练 your一个与全部分类器后，现在 can now可以使用它来预测 the给定图像中包含的数字。对于每个输入，应should使用经过训练的逻辑回归分类器计算 the that它属于 to每个类的"概率"。您的"一个与所有"预测函数将选取 the the相应逻辑回归分类器输出 the最高概率的类 for which，并返回 the类标签 （1，2,... ，或 the prediction for  *K）*)作为输入示例的预测。

现在，您应该在预测OneVsAll.m中完成代码，以便使用一个与全部分类器进行预测。

call your predictOneVsAll function using you are完成后，ex3.m 将使用

\* 的学习价值。您应该看到，训练集的准确率约为 94.9%（即，它正确分类 in the了训练集中 94.9% 的示例）。

*您现在应该提交您的解决方案。*

# 神经网络

在本练习的前一部分中，您实施了多级物流重新访问来识别手写数字。但是，逻辑回归不能形成 more更复杂的假设 as，因为它只是 only线性 linear分类器。[3](#_bookmark4)

在练习的这一部分中，您将实现神经网络，ognize使用与以前相同的训练集来重新访问手写数字。神经网络将能够be 表示形成非线性的模拟的复杂模型hy- potheses。For 本周，您将will be 使用我们已经训练的神经网络的参数。您的目标是实现前馈传播算法，以便使用我们的权重进行预测。在下周的前一个程序，你会写反向传播algorithm学习神经网络network 参数。

The provided script, ex3 nn.m, will help you step through this exercise.

## 模型表示

我们的神经网络如图所示[2](#_bookmark5).它有3个层• 输入层 隐藏层和输出层。回想一下，我们的输入是像素值数字图像。由于图像大小为 20 20，这给了我们400个输入层单位（不包括额外的偏置单位，总是输出 #1）。和以前一样 培训 数据 将 是 加载 到 变量 X 和 y.

*X*

You have been provided with a set of network parameters (Θ(1)*,* Θ(2)) already trained by us. These are stored in ex3weights.mat and will be loaded by ex3 nn.m into Theta1 and Theta2 The parameters have dimensions that are sized for a neural network with 25 units in the second layer and 10 output units (corresponding to the 10 digit classes).

百分比 从文件加载中保存的矩阵（"ex3权重.mat"）;); 负载（

% 矩阵 Theta1 和 Theta2 现在将在您的八边形

% 环境

% Theta1 具有 25 x 401 尺寸

% Theta2 具有尺寸 10 x 26

3您可以向逻辑回归添加更多要素（如多项式要素），但

训练费用可能非常昂贵。

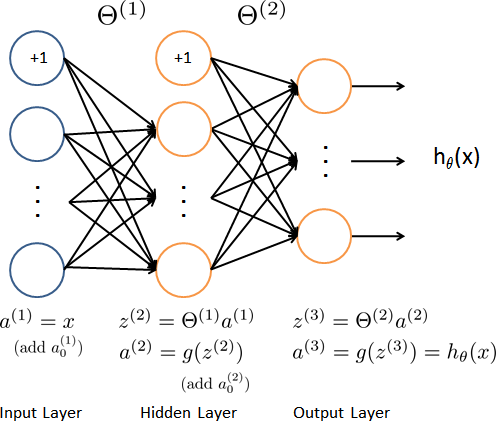


图2：神经网络模型。

## 前馈传播和预测

现在，您将实现神经网络 for的源前传播。您需要在预测中完成代码才能返回神经网络的预测。

您应该实现 the每个示例 *i* that的计算 *h*前馈计算*h +*（*x*（ i*i*），并返回关联的预测。与一对全分类策略类似，来自神经网络的预测将是be the that具有 the最大输出（h*h=*（x*x*）k*k*的标签。

**实现说明：**矩阵 X 包含行中的示例。en 在预测.m中完成代码后，you 您需要将add the 1 的列添加到矩阵中。 矩阵 Theta1 和 Theta2 包含行中每个单元的参数。具体而言row of ，Theta1的第一行对应于第二层中的第一个隐藏单元。在八度/MAT-LAB中，当您you 计算 *z* rrectly （2） = (=（1）*a*（1） 时，请确保索引Θ（如有必要，转置）X， co以便获得 *a*一个*（l）*) 作为列矢量。

完成后，ex3 nn.m将使用Theta1 和 Theta2and Theta2的加载参数集调用预测predict 函数。你应该看到，

准确率约为97.5%。之后，交互式序列将从训练集中一次启动不播放的图像，而控制台将打印出显示图像的预测标签。要停止图像序列，请按Ctrl-C。

*您现在请提交您的解决方案。*

# 提交和分级

完成此作业后， be请务必 to使用 the提交函数将您的解决方案分mit给我们的服务器。 The以下是 is a of how本练习的每个部分的评分的细目。

|  |  |  |
| --- | --- | --- |
| **部分** | **已提交文件** | **点** |
| 正化日志回归 | lrCost函数.m | 30 分 |
| 一个与全部分类器训练  一个与全部分类器预测 | 一VsAll.m  预测一个VsAll.m | 20分  20分 |
| 神经网络预测函数 | 预测.m | 30 分 |
| 总积分 |  | 100 点 |

您可以多次提交您的解决方案，我们将只考虑最高分。