

# プログラミング演習1：線形回帰

## 機械学習

### 前書き

この演習では、線形回帰を実装し、動作を確認します  
データ上。このプログラミング演習を開始する前に、  
ビデオ講義を見て、レビューの質問を完了する  
関連するトピック。

エクササイズを開始するには、スターターをダウンロードする必要があります  
コードを作成し、その内容を完了したいディレクトリに解凍します  
運動。必要に応じて、Octave / MATLABでcdコマンドを使用して、  
この演習を開始する前に、このディレクトリ。

Octave / MATLABのインストール手順は、「En-  
コースWebサイトの「セットアップ手順」を参照してください。

### この演習に含まれるファイル

- ex1.m-演習をステップごとに実行するOctave / MATLABスクリプト
- ex1\_multi.m-演習の後半のOctave / MATLABスクリプト
- ex1data1.txt-1つの変数を使用した線形回帰のデータセット
- ex1data2.txt-複数の変数を持つ線形回帰のデータセット
- submit.m-サーバーにソリューションを送信する送信スクリプト
- [\*] warmUpExercise.m-Octave / MATLABのシンプルなサンプル関数
- [\*] plotData.m-データセットを表示する関数
- [\*] computeCost.m-線形回帰のコストを計算する関数
- [\*] gradientDescent.m-勾配降下を実行する関数
- [†] computeCostMulti.m-複数の変数のコスト関数
- [†] gradientDescentMulti.m-複数の変数の勾配降下
- [†] featureNormalize.m-フィーチャーを正規化する関数
- [†] normalEqn.m-正規方程式を計算する関数

は、完了する必要があるファイルを示します  
†はオプションの演習を示します

演習を通して、スクリプトex1.mおよびex1\_multi.mを使用します。  
 これらのスクリプトは、問題のデータセットを設定し、関数を呼び出します  
 あなたが書くこと。どちらも変更する必要はありません。あなただけです  
 以下の手順に従って、他のファイルの関数を変更する必要があります  
 この割り当て。

このプログラミング演習では、最初の課題のみを完了する必要があります  
 1つの変数で線形回帰を実装する演習の一部。の  
 演習の2番目の部分はオプションで、次の線形回帰を扱います。  
 複数の変数。

## 助けを得る場所

このコースの演習では、Octaveを使用します<sup>1</sup>またはMATLAB、高レベルプログラム-  
 数値計算に適した明言語。お持ちでない場合  
 OctaveまたはMATLABがインストールされています。次のインストール手順を参照してください。  
 コースWebサイトの「環境設定手順」。

Octave / MATLABコマンドラインで、helpに続けてfunc-  
 名前には、組み込み関数のドキュメントが表示されます。たとえば、ヘルプ  
 plotは、プロットに関するヘルプ情報を表示します。の詳細なドキュメント  
 Octave関数は、[Octaveのドキュメントページにあります。](#)。マット-  
 LABドキュメントは、[MATLABドキュメントページ](#)で見つけることができます。

また、オンラインディスカッションを使用して、元の  
 他の生徒と一緒に ただし、書かれたソースコードは見ないでください  
 他人によって、またはソースコードを他人と共有する。

## 1 シンプルなオクターブ/ MATLAB関数

ex1.mの最初の部分では、Octave / MATLAB構文の練習と、  
 宿題提出プロセス。ファイルwarmUpExercise.mでは、  
 Octave / MATLAB関数の概要を見つけます。5 xを返すように変更します  
 5次のコードを入力して単位行列：

```
A = eye (5) ;
```

<sup>1</sup> Octaveは、MATLABの無料の代替です。プログラミング演習では、無料です  
 OctaveまたはMATLABを使用します。

終了したら、ex1.mを実行します（正しいディレクトリにいると仮定します。  
直近、Octave / MATLABプロンプトで「ex1」と入力すると表示されます  
次のような出力：

```
ans =  
  
対角行列  
  
      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
```

ex1.mは、いずれかのキーを押すまで一時停止し、コードを実行します  
割り当ての次の部分。終了する場合は、ctrl-cと入力します  
実行中にプログラムを停止します。

### 1.1ソリューションの提出

演習の一部を完了した後、次のソリューションを提出できます。  
Octave / MATLABコマンドラインでsubmitと入力して評価します。サブ  
ミッションスクリプトは、ログイン用の電子メールと送信トークンの入力を求めます  
送信するファイルを尋ねます。提出物を入手できます  
割り当てのWebページからのトークン。

ここで、ソリューションを送信する必要があります。

ソリューションを複数回送信することが許可されています。  
最高のスコアのみが考慮されます。

## 2つの変数を持つ線形回帰

この演習のこの部分では、1つの線形回帰を実装します  
フードトラックの利益を予測する変数。あなたがCEOであると仮定します  
レストランのフランチャイズと新しいを開くためのさまざまな都市を検討しています  
出口。チェーンにはすでにさまざまな都市にトラックがあり、あなたは  
都市からの利益と人口。

このデータを使用して、拡大する都市を選択します。  
次へ。  
ファイルex1data1.txtには、線形回帰問題のデータセットが含まれています。  
レム。最初の列は都市の人口であり、2番目の列は  
その都市のフードトラックの利益。利益の負の値は、  
損失。  
ex1.mスクリプトは、このデータを読み込むために既に設定されています。

2.1データのプロット

タスクを開始する前に、次の方法でデータを理解しておく と 便利 です。  
それを視覚化します。このデータセットでは、散布図を使用して、  
プロットする2つのプロパティ（利益と母集団）しかないため、データ。（た く さ ん の  
実生活で遭遇する他の問題は多次元であり、  
2次元プロットではプロットできません。）  
ex1.mでは、データセットはデータファイルから変数Xにロードされます  
およびy：

```
data = load ('ex1data1.txt') ; %コンマ区切りデータを読み取ります
X = data (:, 1) ; y = data (:, 2) ;
m =長さ (y) ; %トレーニング例の割合
```

次に、スクリプトはplotData関数を呼び出して、次の散布図を作成します。  
データ。あなたの仕事はplotData.mを完成させてプロットを描くことです。変更する  
ファイルして、次のコードを入力します。

```
plot (x、y、'rx'、'MarkerSize'、10) ; %データをプロット
ylabel ('10、000ドルの利益') ; %y軸ラベルを設定します
xlabel ('10、000人の都市の人口') ; %x軸ラベルを設定します
```

さて、ex1.mを実行し続けると、最終結果は次のようになります。  
図1、同じ赤い「x」 マーカーと軸ラベル。  
plotコマンドの詳細については、次の場所にhelp plotと入力できます。  
Octave / MATLABコマンドプロンプトまたはプロットドキュメントのオンライン検索  
刺激。（マーカーを赤い「x」に変更するには、オプション「rx」を使用しました  
plotコマンドと一緒に、つまりplot (...、[your options here]、...、  
'rx') ;)

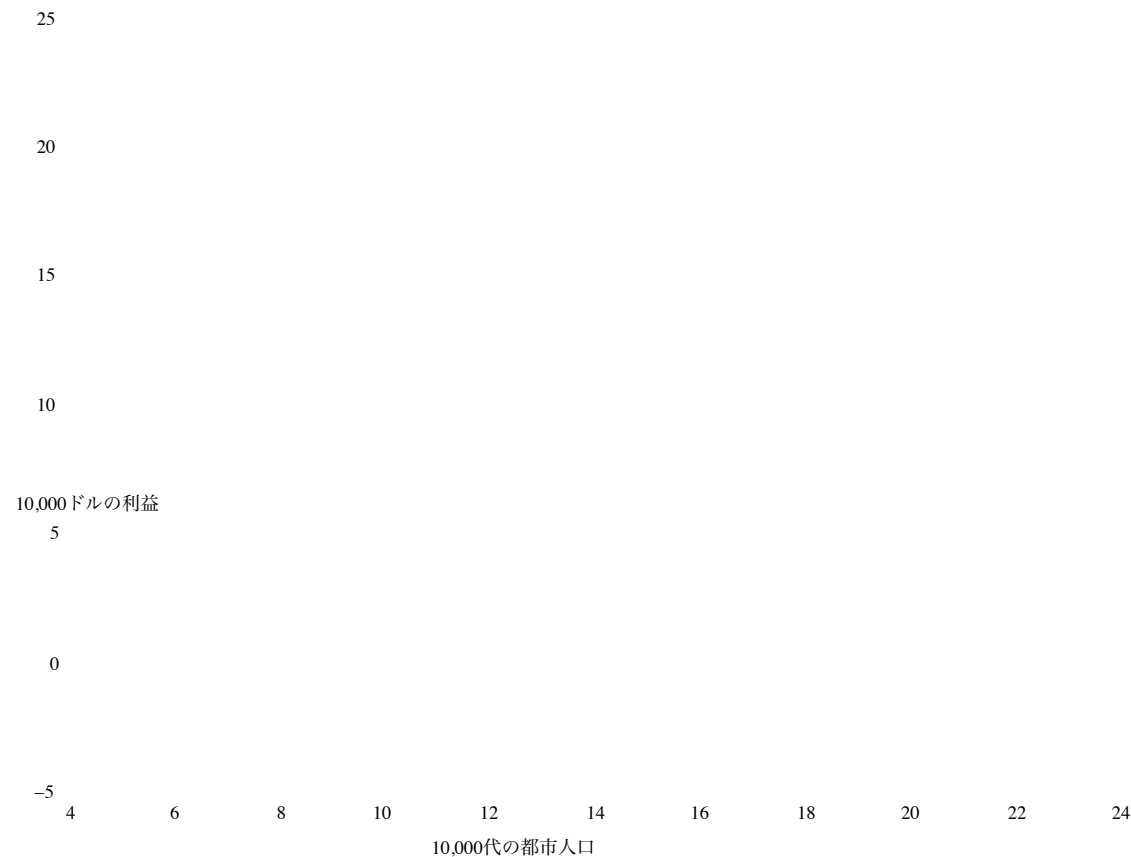


図1：トレーニングデータの散布図

## 2.2 勾配降下

このパートでは、線形回帰パラメーター $\theta$ をデータセットに適合させます。勾配降下を使用します。

### 2.2.1 方程式の更新

線形回帰の目的は、コスト関数を最小化することです。

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

目的関数

ここで、仮説 $h_{\theta}(x)$ は線形モデルで与えられます。

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

仮説関数

お使いのモデルのパラメータが $\theta$ であることを思い出して、 $J(\theta)$ の値。これらはコスト $J(\theta)$ を最小化するために調整する値。これを行う1つの方法はバッチ勾配降下アルゴリズムを使用します。バッチ勾配降下では、それぞれ反復は更新を実行します。

# 最適＝最適

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$
（すべてのj に対して  $\theta_j$  を同時に更新します）。

勾配降下の各ステップで、あなたのパラメータ  $\theta_j$  は近い来ます  
最低コスト  $J(\theta)$  を達成する最適値。

実装上の注意：各サンプルをXの行として保存します  
Octave / MATLABの行列。アカウントに切片項（ $\theta_0$  取るために、 $\theta_0$  を）、  
Xに最初の列を追加して、すべて1に設定します。これにより  
治療のために私たちは、 $\theta_0$  を単に別の「機能」と。

## 2.2.2実装

ex1.mでは、線形回帰のデータを既に設定しています。の中に  
次の行に、データに別の次元を追加して、  
 $\theta_0$  インターセプト用語。また、初期パラメーターを0に初期化し、  
学習率アルファ0.01。

```
X = [ones (m, 1) , data (:, 1) ]; %1の列をxに追加します
theta = zeros (2, 1) ; %フィッティングパラメーターの初期化
```

```
反復= 1500;
アルファ= 0.01;
```

## 2.2.3コストJ (θ) の計算

勾配降下を実行して学習すると、コスト関数  $J(\theta)$  が最小になります。  
コストを計算して収束を監視すると役立ちます。これで  
セクションでは、 $J(\theta)$  を計算する関数を実装して、  
勾配降下実装の収束。

次のタスクは、ファイルcomputeCost.mのコードを完成させることです。  
は、 $J(\theta)$  を計算する関数です。これを行っているときに、  
変数Xおよびyはスカラー値ではなく、行が表す行列  
トレーニングセットの例。

関数を完了すると、ex1.mの次のステップが実行されます  
ゼロに初期化された $\theta$ を使用してcomputeCostを1回実行すると、コストが表示されます  
画面に印刷されます。

32.07のコストが発生することが予想されます。

ここで、ソリューションを送信する必要があります。



2.2.4勾配降下

次に、ファイルgradientDescent.mに勾配降下を実装します。  
ループ構造はあなたのために書かれており、あなただけを提供する必要があります  
各反復内のθの更新。

プログラムを作成する際には、最適化しようとしていることを理解してください。  
mizeと何が更新されているか。コストJ (θ) はパラメータであることに注意してください  
Xとyではなく、ベクトルθによってターライズされます。つまり、J (θ) の値を最小化します  
Xまたはyを変更するのではなく、ベクトルθの値を変更する。を参照してください  
不明な場合は、この配布資料とビデオレクチャーの方程式を参照してください。

勾配降下が正しく機能していることを確認する良い方法は、  
J (θ) の値で、各ステップで減少していることを確認します。の  
gradientDescent.mのスターターコードは、反復ごとにcomputeCostを呼び出します  
コストを出力します。勾配降下法を実装しており、  
computeCostが正しく、J (θ) の値が増加することはありません。  
アルゴリズムの終わりまでに安定した値に収束します。

終了したら、ex1.mは最終パラメーターを使用して、  
線形フィット。結果は図2のようになります。

θの最終値は、利益の予測にも使用されます  
35,000人と70,000人のエリア。次の行が  
ex1.mは、明示的な加算またはループではなく、行列乗算を使用します。  
ing、予測を計算します。これは、次のコードベクトル化の例です。  
オクターブ/MATLAB。

ここで、ソリューションを送信する必要があります。

```
predict1 = [1, 3.5] * theta;  
predict2 = [1, 7] * theta;
```

2.3デバッグ

勾配降下を実装する際に留意すべき点があります。

- Octave / MATLAB配列のインデックスは、ゼロではなく1から始まります。保存している場合  
θINGの0及びθ1シータと呼ばれるベクトルで、値は、シータ (1) であり、そして  
theta (2) 。
- 実行時に多くのエラーが発生している場合は、マトリックス操作を調べます  
互換性のある行列を追加および乗算していることを確認します  
可能な寸法。サイズ付き変数の寸法を印刷する  
コマンドはデバッグに役立ちます。

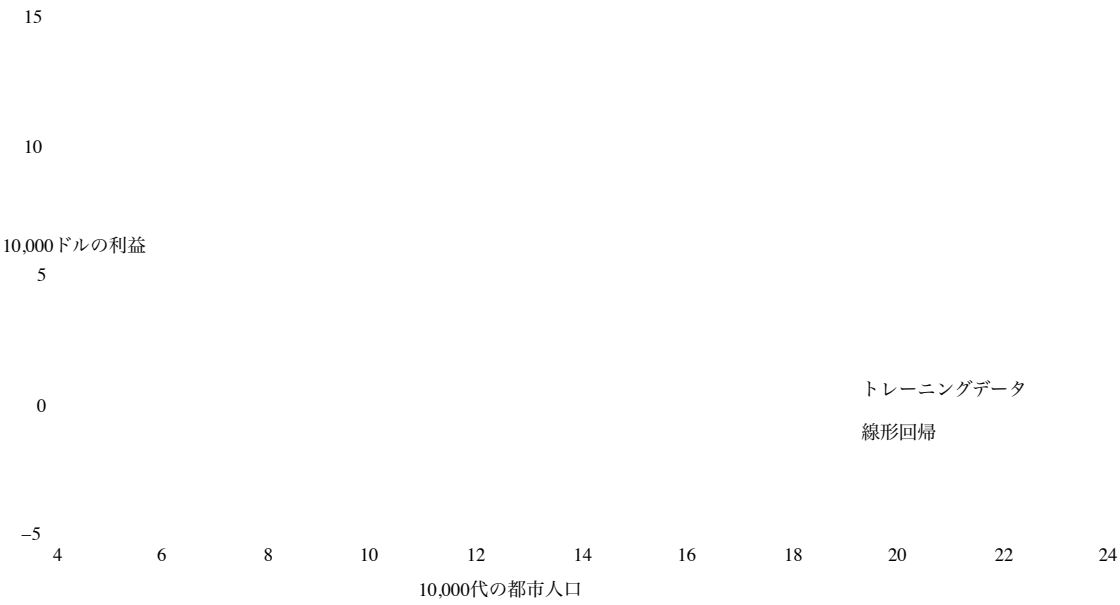


図2：線形回帰近似を使用したトレーニングデータ

- デフォルトでは、Octave / MATLABは数学演算子を行列と解釈します  
演算子。これは、サイズの非互換性エラーの一般的な原因です。もし、あんたが  
行列の乗算が必要ない場合は、「ドット」表記を追加する必要があります  
これをOctave / MATLABに指定するには。たとえば、 $A * B$ は行列を実行します  
乗算しますが、 $A. * B$ は要素ごとの乗算を行います。

## 2.4 J (θ) の視覚化

コスト関数J (θ) をよりよく理解するために、次のコストをプロットします。  
θの2次元グリッドθ<sub>0</sub>とθ<sub>1</sub>の値。何もコーディングする必要はありません  
この部分は新しいが、あなたが書いたコードがどのように  
すでにこれらの画像を作成しています。

ex1.mの次のステップでは、Jを計算するために設定されたコードがあります  
作成したcomputeCost関数を使用した値のグリッド。

```
%J valsを0の行列に初期化します
J vals = zeros (length (theta0 vals) , length (theta1 vals) ) ;

J値を記入する%
for i = 1 : length (theta0 vals)
```



```
以下のための長さ (theta1のヴァルス) : J = 1
t = [theta0 vals (i) ; theta1 vals (j) ];
J vals (i, j) = computeCost (x, y, t) ;
終わり
終わり
```

これらの行が実行されると、J (θ) 値の2次元配列が作成されます。  
スクリプトex1.mは、これらの値を使用して表面と輪郭を生成します  
surfコマンドとcontourコマンドを使用したJ (θ) のプロット。プロットは次のようになります  
図3のようなもの :

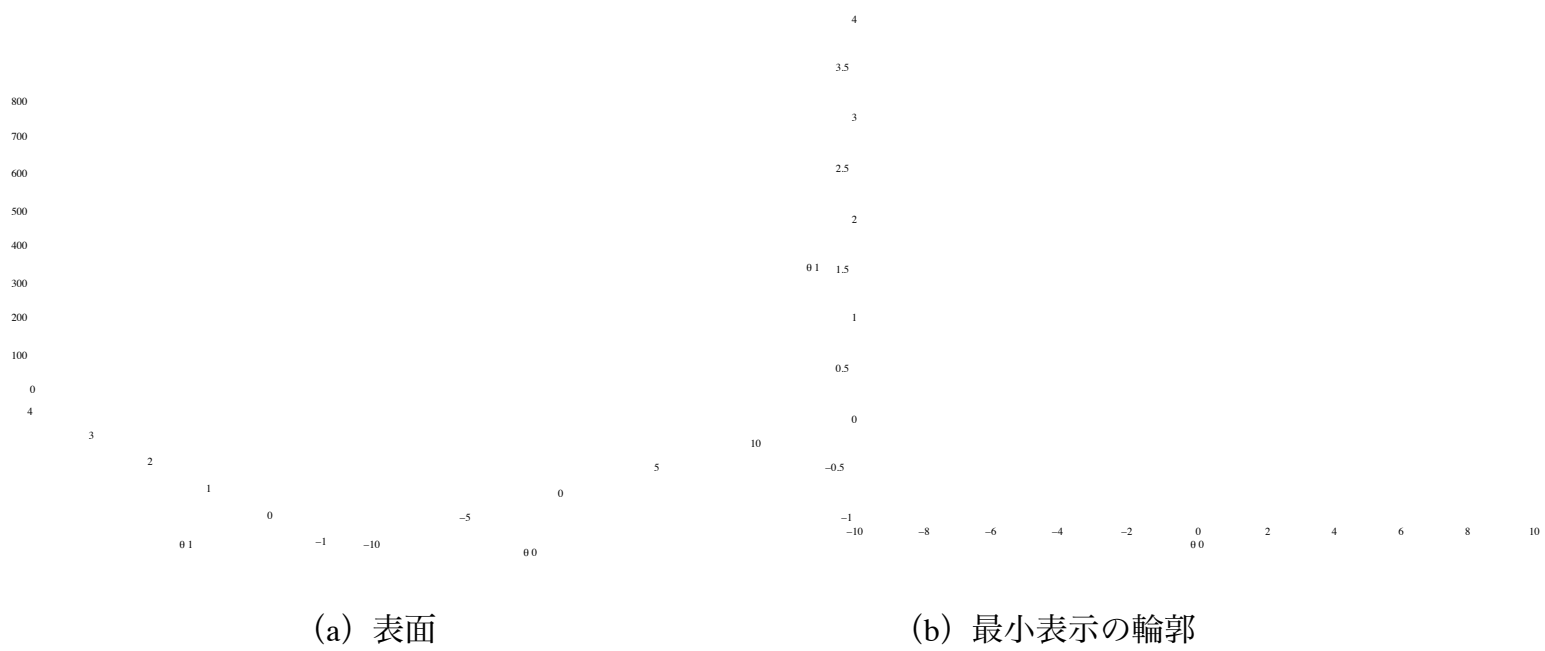


図3：コスト関数J (θ)

これらのグラフの目的は、J (θ) がどのように変化するかを示すことです。  
θの変化<sub>0</sub>とθ<sub>1</sub>。コスト関数J (θ) はボウル型であり、グローバル  
最小（これは、3Dサーフェスよりも等高線プロットで見やすいです。  
プロット）。この最小値は、θに対する最適点で<sub>0</sub>とθ<sub>1</sub>との各ステップ  
勾配降下は、このポイントに近づきます。

## オプションの演習

上記の資料を正常に完了した場合、おめでとうございます！君は今線形回帰を理解し、あなたのそれで使用を開始できるはずです  
独自のデータセット。

このプログラミング演習の残りの部分では、次のものを含めました。  
オプションの演習。これらの演習は、より深い理解を得るのに役立ちます  
材料の、そしてあなたがそうすることができるなら、私たちはあなたが完了することをお勧めします

それらも。

## 3 複数の変数を伴う線形回帰

このパートでは、複数の変数を使用して線形回帰を実装し、家の価格を予測します。あなたがあなたの家を売っていて、あなたが良い市場価格がどうなるか知りたい。これを行う1つの方法は最初に販売された最近の住宅に関する情報を収集し、住宅のモデルを作成します価格。

ファイルex1data2.txtには、Port-の住宅価格のトレーニングセットが含まれています。土地、オレゴン。最初の列は家のサイズ（平方フィート）、2列目は寝室の数、3列目は価格家の。

ex1\_multi.mスクリプトは、これをステップスルーするのに役立つようにセットアップされています運動。

### 3.1 機能の正規化

ex1\_multi.mスクリプトは、いくつかの値を読み込んで表示することから開始しますこのデータセットから。値を見ると、家のサイズは約ベッドルームの1000倍。機能が大きさによって異なる場合nitude、最初に機能スケーリングを実行すると、勾配降下を収束させることができますより迅速に。

ここでのタスクは、featureNormalize.mのコードを完成させることです。

- データセットから各フィーチャの平均値を引きます。
- 平均を差し引いた後、特徴値をさらにスケーリング（分割）しますそれぞれの「標準偏差」によって。

標準偏差は、どれだけ変動があるかを測定する方法です特定のフィーチャの値の範囲内（ほとんどのデータポイントは内にあります平均の±2標準偏差）；これは範囲を取る代わりに値の（最大-最小）。Octave / MATLABでは、「std」関数を使用して標準偏差を計算します。たとえば、featureNormalize.m内では、量X（：、1）には、トレーニングのx<sub>1</sub>（家のサイズ）のすべての値が含まれます設定すると、std（X（：、1））は家のサイズの標準偏差を計算します。featureNormalize.mが呼び出された時点で、1の追加列x<sub>0</sub> = 1に対応するXはまだ追加されていません（ex1\_multi.mを参照してください詳細）。

すべての機能に対してこれを行い、コードは

すべてのサイズのデータセット（任意の数の機能/例）。 それぞれに注意してください  
行列Xの列は1つの特徴に対応します。

ここで、ソリューションを送信する必要があります。

実装ノート：機能を正規化する場合、重要です  
正規化に使用される値を保存する-平均値とスタン-  
計算に使用されるダード偏差。 パラメータを学習した後  
モデルから、私たちはしばしば私たちが持っていない家の価格を予測したい  
前に見た。新しいx値（リビングルームの面積とベッド数-  
部屋）、最初に平均と標準偏差を使用してxを正規化する必要があります  
以前にトレーニングセットから計算したこと。

3.2勾配降下

以前は、単変量回帰で勾配降下を実装しました  
問題。唯一の違いは、もう1つの機能があることです  
行列X。仮説関数とバッチ勾配降下更新  
ルールは変更されません。

computeCostMulti.mおよびgradientDescentMulti.mのコードを完了する必要があります。  
線形回帰のコスト関数と勾配降下を実装するには  
複数の変数。前の部分のコード（単一変数）が既に  
複数の変数をサポートしています。ここでも使用できます。

コードが多くの機能をサポートし、適切にベクトル化されていることを確認してください。  
'size (X, 2) 'を使用して、いくつかの機能が存在するかを調べることができます  
データセット。

ここで、ソリューションを送信する必要があります。

実装ノート：多変量の場合、コスト関数は  
また、次のベクトル化された形式で記述されます。

$$J(\theta) = \frac{1}{2m} (X\theta - y)^T (X\theta - y)$$

どこで

$$X = \begin{bmatrix} - & (x_{(1)})^T - \\ - & (x_{(2)})^T - \\ & \dots \\ - & (x_{(m)})^T - \end{bmatrix} \qquad y = \begin{bmatrix} y_{(1)} \\ y_{(2)} \\ \dots \\ y_{(m)} \end{bmatrix} \text{。}$$

ベクトル化されたバージョンは、数値で作業しているときに効率的です  
Octave / MATLABなどのコンピューティングツール。あなたがマトリックスの専門家である場合  
操作では、2つの形式が同等であることを自分で証明できます。

### 3.2.1 オプションの（未定）演習：学習率の選択

演習のこの部分では、異なる学習率を試してみることができます  
データセットを取得し、すぐに収束する学習率を見つけます。あなたは変えられる  
ex1\_multi.mを変更し、  
学習率を設定するコード。

ex1\_multi.mの次のフェーズでは、gradientDescent.m func-を呼び出します。  
選択した学習で約50回反復して勾配降下を実行します  
レート。この関数は、 $J(\theta)$  値の履歴をベクトルで返す必要があります  
J.最後の反復の後、ex1\_multi.mスクリプトはJ値をプロットします  
繰り返しの数。

適切な範囲内の学習率を選択した場合、プロットは同様に見えます  
図4.グラフが大きく異なる場合、特に $J(\theta)$  の値が  
増加したり、爆発したりした場合は、学習率を調整して再試行してください。私たちは  
乗法での対数スケールでの学習率 $\alpha$ の試行値を推奨する  
前の値の約3倍のステップ（0.3、0.1、0.03、0.01など）。  
また、実行している反復の数を調整することもできます。  
曲線の全体的な傾向を確認するのに役立ちます。

図4：適切な学習率での勾配降下の収束

実装ノート：学習率が大きすぎる場合、 $J(\theta)$  は寸前で「爆発」し、コンピューターには大きすぎる値が生じる計算。これらの状況では、Octave / MATLABは戻る傾向があります NaN。NaNは「not a number」の略で、多くの場合未定義によって引き起こされます  $-\infty$  および  $+\infty$  を含む演算。

Octave / MATLAB ヒント：学習学習の違いを比較するにはレートは収束に影響するため、いくつかの学習レートについてJをプロットすると役立ちます。同じ図に。Octave / MATLABでは、これはperform-間に「hold on」コマンドを使用して複数回勾配降下を行うプロット。具体的には、3つの異なるアルファ値を試した場合（おそらくこれよりも多くの値を試してください）、コストをJ1、J2、およびJ3では、次のコマンドを使用して同じ図にプロットできます。

```
plot (1:50、J1 (1:50) 、 'b') ;
つかまっている;
plot (1:50、J2 (1:50) 、 'r') ;
plot (1:50、J3 (1:50) 、 'k') ;
```

最後の引数 'b'、 'r'、 および 'k'は、異なる色を指定しますプロット。

学習率が変化すると、収束曲線が変化することに注意してください。学習率が小さい場合、勾配降下は非常に時間がかかることに気付くはずで。最適値に収束するまでに長い時間がかかります。逆に、大規模な学習勾配降下は収束しないか、発散さえするかもしれません！

見つけた最高の学習率を使用して、ex1 multi.mスクリプトを実行します収束するまで勾配降下を実行して、 $\theta$ の最終値を見つけます。次、この $\theta$ の値を使用して、1650平方フィートの家の価格を予測し、3つのベッドルーム。後で値を使用して、の実装を確認します正規方程式。あなたが作るときにあなたの機能を正規化することを忘れないでくださいこの予測！

これらのオプション（未評価）のソリューションを提出する必要はありません。演習。

3.3正規方程式

講義ビデオでは、線形の閉じた形式のソリューション回帰は

$$\theta = (X^T X)^{-1} X^T y。$$

この式を使用すると、機能のスケーリングは必要ありません。  
1回の計算での正確な解：次のような「収束までループ」はありません  
勾配降下。  
上記の式を使用して計算するには、normalEqn.mのコードを完成させます。  
後半のθ。機能をスケーリングする必要はありませんが、  
切片項（θ持つようにX行列への1の列を追加する必要がありますo）。  
ex1.mのコードは、1の列をXに追加します。

ここで、ソリューションを送信する必要があります。

オプションの（未評価の）演習：ここで、これを使用してθを見つけたら  
方法、これを使用して1650平方フィートの家の価格予測を行います  
3つのベッドルーム。値と同じ予測価格が得られるはずです。  
勾配降下を伴うモデル近似を使用して取得しました（セクション3 [2.1](#)）。

## 提出と採点

課題のさまざまな部分を完了したら、必ず送信を使用してください  
ソリューションを当社のサーバーに提出するための機能システム。 以下は  
この演習の各部分の採点方法の内訳。

部	提出されたファイル	ポイント
ウォームアップ運動	warmUpExercise.m	10ポイント
1つの変数のコストを計算する	computeCost.m	40ポイント
1つの変数の勾配降下	gradientDescent.m	50ポイント
合計点		100ポイント
オプションの演習		
部	提出されたファイル	ポイント
機能の正規化	featureNormalize.m	0ポイント
複数の費用を計算する	computeCostMulti.m	0ポイント
変数		
複数の勾配降下	gradientDescentMulti.m	0ポイント
変数		
正規方程式	normalEqn.m	0ポイント

ソリューションを複数回送信することが許可されています。  
最高のスコアのみが考慮されます。

