**6(a)**

train\_data = csvread('regression\_train.csv');

x = train\_data(:,1);

y = train\_data(:,2);

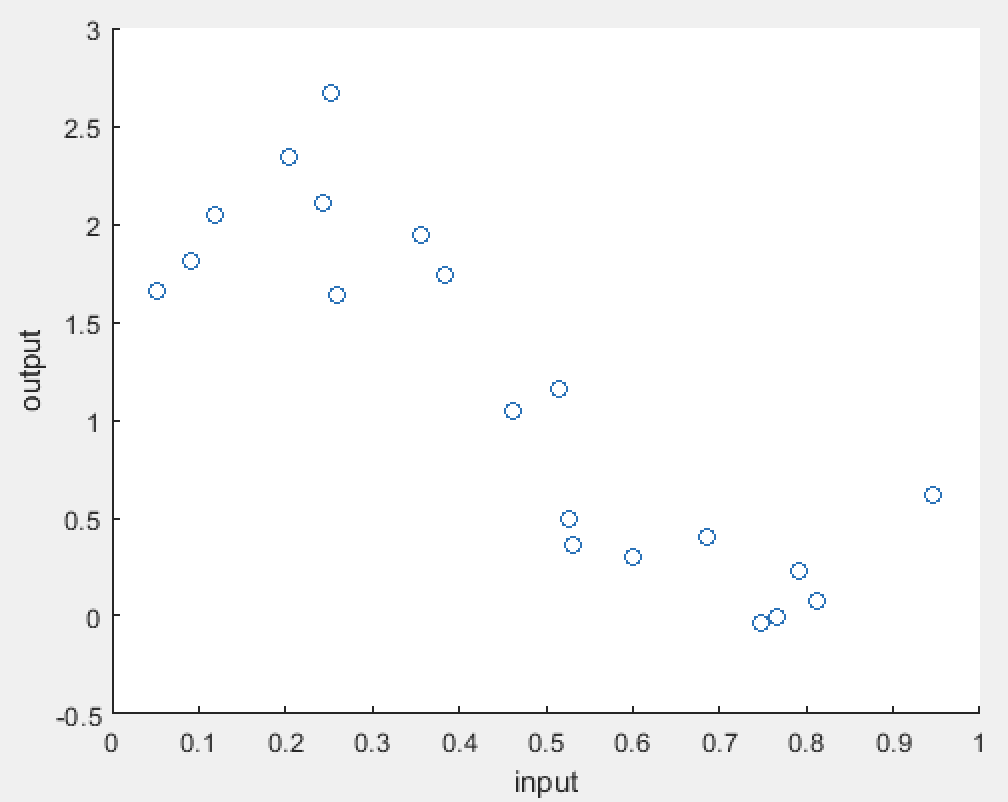
format long;

scatter(x,y);

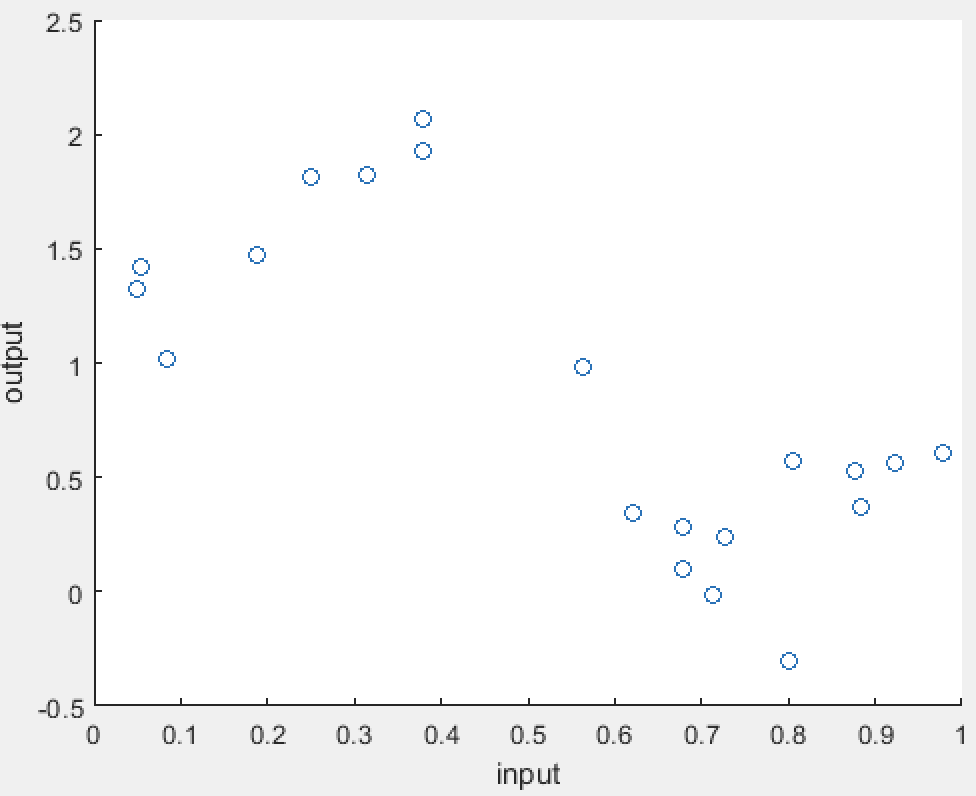
xlabel('input');

ylabel('output');

Training data:



Testing data:



Compared to the training data set, the testing data set are more scattering and forms a boundary between two divisions of data points. The training data set shows a more concentrated pattern that might better fit on a line. Therefore, I guess that the linear regression will better fit the training data set.

**6(b)**

train\_data = csvread('regression\_train.csv');

x = train\_data(:,1);

y = train\_data(:,2);

format long;

scatter(x,y);

xlabel('input');

ylabel('output');

hold on;

X = [ones(length(x),1) x];

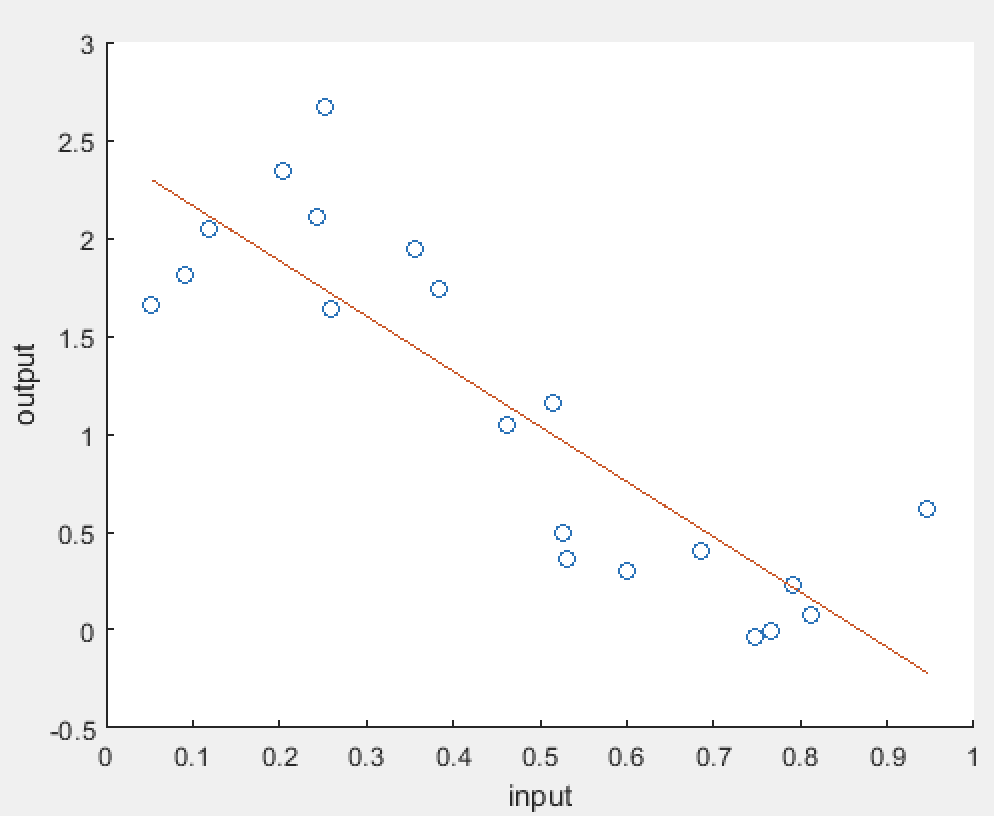
w = X\y;

y1 = X\*w;

plot(x, y1);

% J(w)

norm(X\*w-y,2);



w0 = 2.446407094714721

w1 = -2.816353589568697

J(w) = ∥Xw − y∥2 = 3.912576405791464

**6(c)**

train\_data = csvread('regression\_train.csv');

x = train\_data(:,1);

y = train\_data(:,2);

format long;

X = [ones(length(x),1) x];

w = [0;0]

Jw\_curr = intmax('int32');

Jw\_prev = intmax('int32');

eta = 0.0407

for iter = 1:10000

z1 = 0;

z2 = 0;

for i = 1:length(x)

z1 = z1+(w(1)+w(2)\*x(i)-y(i))\*X(i,1);

z2 = z2+(w(1)+w(2)\*x(i)-y(i))\*X(i,2);

end

w(1) = w(1) - eta\*z1;

w(2) = w(2) - eta\*z2;

Jw\_curr = norm(X\*w-y,2)\*norm(X\*w-y,2);

if abs(Jw\_curr - Jw\_prev) < 0.0001

break;

end

Jw\_prev = Jw\_curr;

end

eta = 0.0407

J(w) = 3.9136

iteration = 104

eta = 0.01

J(w) = 3.9170

iteration = 364

eta = 0.001

J(w) = 3.9578

iteration = 2609

eta = 0.0001

J(w) = 5.4936

iteration = 10000

Higher learning rates cause the algorithm to converge faster and result in a smaller J(w) value that is closer to the optimal J(w) value.

**6(d)**

train\_data = csvread('regression\_train.csv');

x = train\_data(:,1);

y = train\_data(:,2);

format long;

X = [ones(length(x),1) x];

w = [0;0]

Jw\_curr = intmax('int32');

Jw\_prev = intmax('int32');

eta = 0.0407

for iter = 1:40

z1 = 0;

z2 = 0;

for i = 1:length(x)

z1 = z1+(w(1)+w(2)\*x(i)-y(i))\*X(i,1);

z2 = z2+(w(1)+w(2)\*x(i)-y(i))\*X(i,2);

end

w(1) = w(1) - eta\*z1;

w(2) = w(2) - eta\*z2;

Jw\_curr = norm(X\*w-y,2)\*norm(X\*w-y,2);

if abs(Jw\_curr - Jw\_prev) < 0.0001

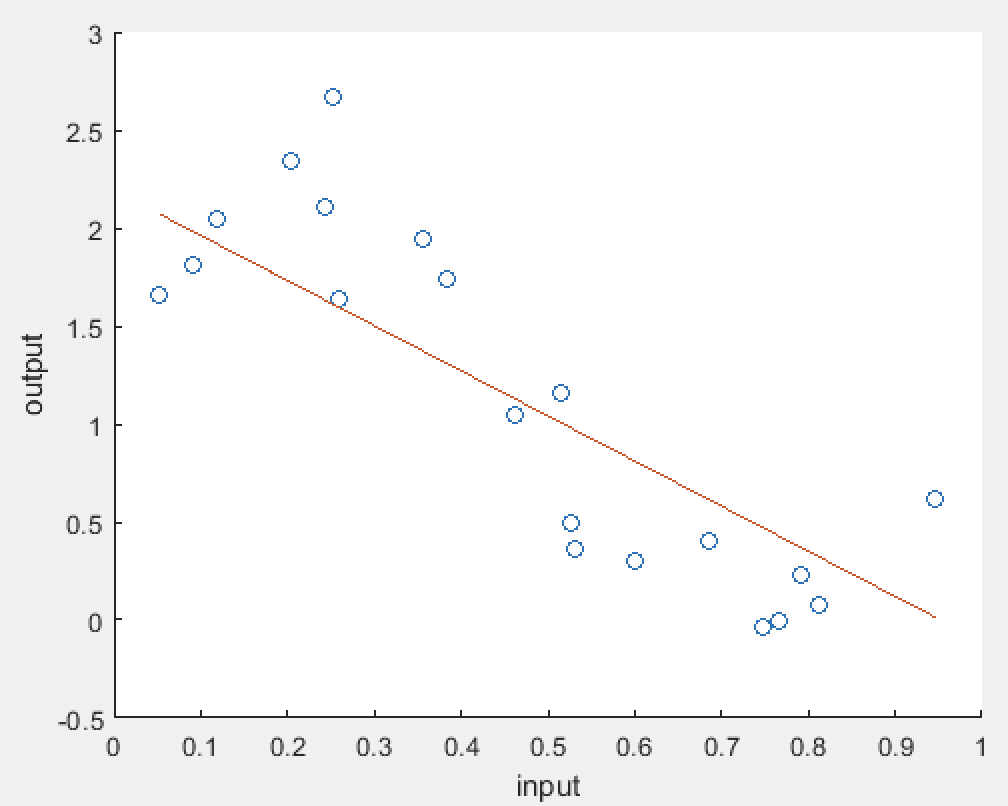
break;

end

Jw\_prev = Jw\_curr;

end

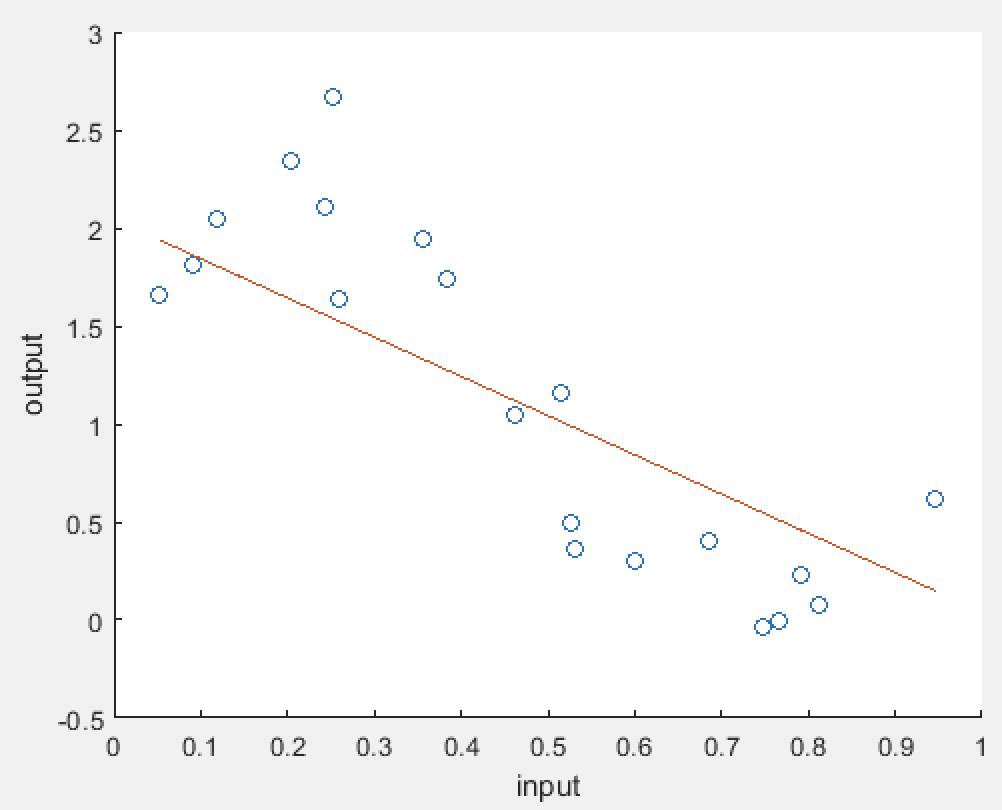
iteration = 40



w = (2.1919, -2.3018)

J(w) = 4.2763

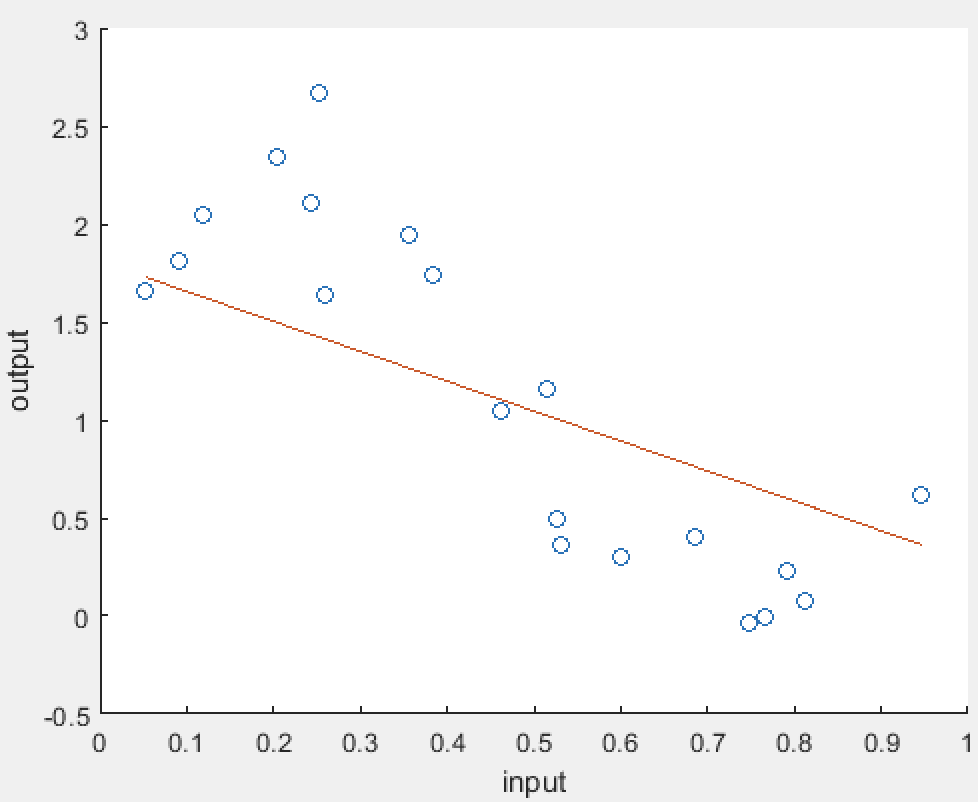
iteration = 30



w = (2.0434, -2.0016)

J(w) = 4.8246

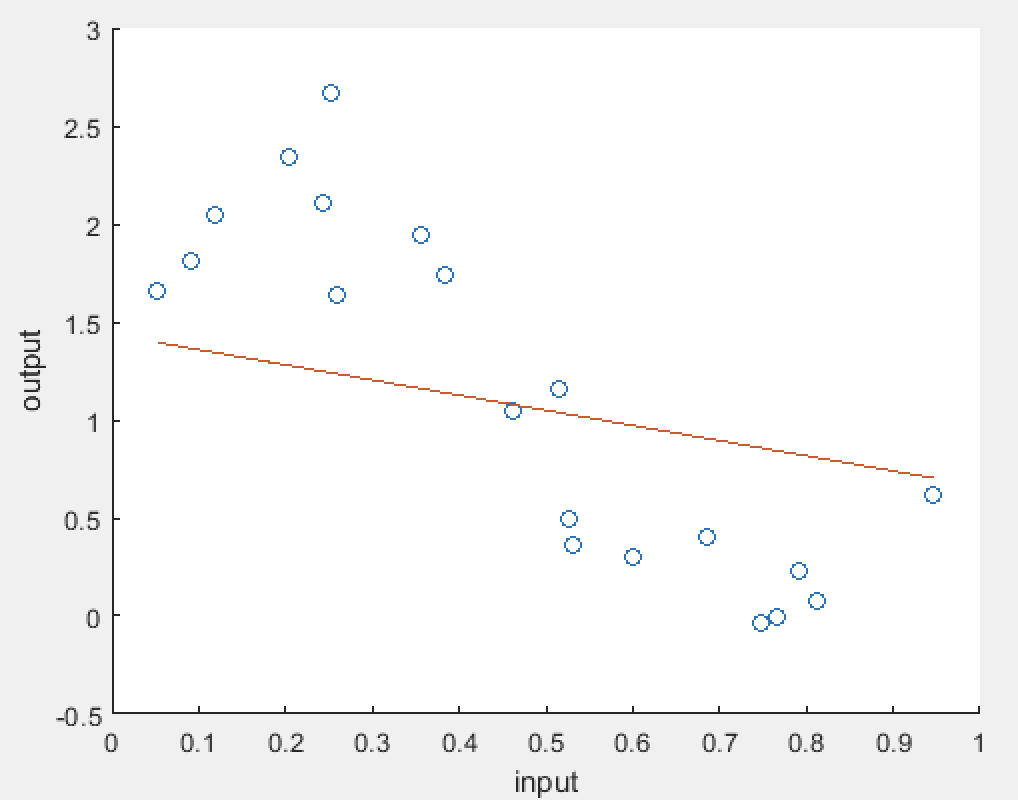
iteration = 20



w = (1.8082, -1.5262)

J(w) = 6.1994

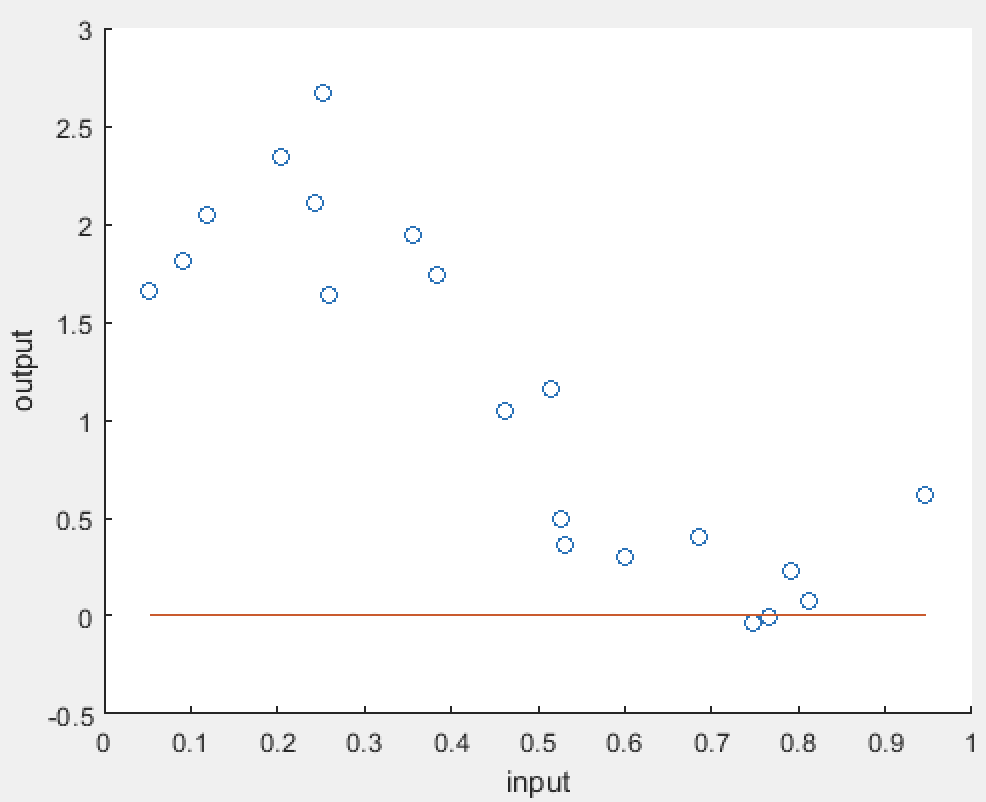
iteration = 10



w = (1.4358, -0.7734)

J(w) = 9.6463

iteration = 0



w = (0, 0)

J(w) = 40.2338

**6(e)**

train\_data = csvread('regression\_train.csv');

x = train\_data(:,1);

y = train\_data(:,2);

test\_data = csvread('regression\_test.csv');

x\_t = test\_data(:,1);

y\_t = test\_data(:,2);

format long;

E\_rms = zeros(11,1);

for m = 0:10

clear X;

clear Z;

for i = 1:20

a = zeros(m+1,1);

b = zeros(m+1,1);

for j = 0:m

a(j+1) = power(x(i),j);

b(j+1) = power(x\_t(i),j);

end

X(i, :) = transpose(a);

Z(i, :) = transpose(b);

end

disp("X:");

disp(X);

w = X\y;

y1 = X\*w;

E\_rms(m+1) = sqrt((norm(X\*w-y,2)\*norm(X\*w-y,2)+norm(Z\*w-y\_t,2)\*norm(Z\*w-y\_t,2))/40);

end

x\_axis = [0;1;2;3;4;5;6;7;8;9;10];

scatter(x\_axis, E\_rms);

xlabel('input');

ylabel('output');

