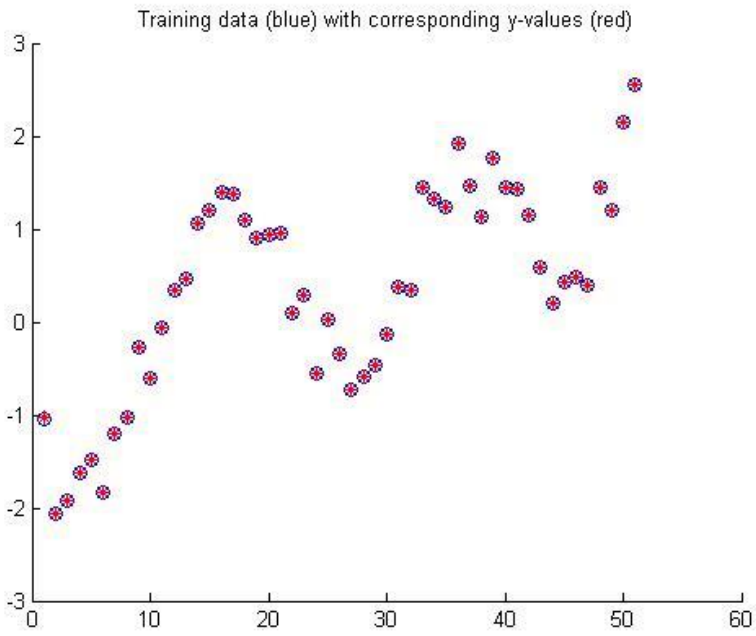
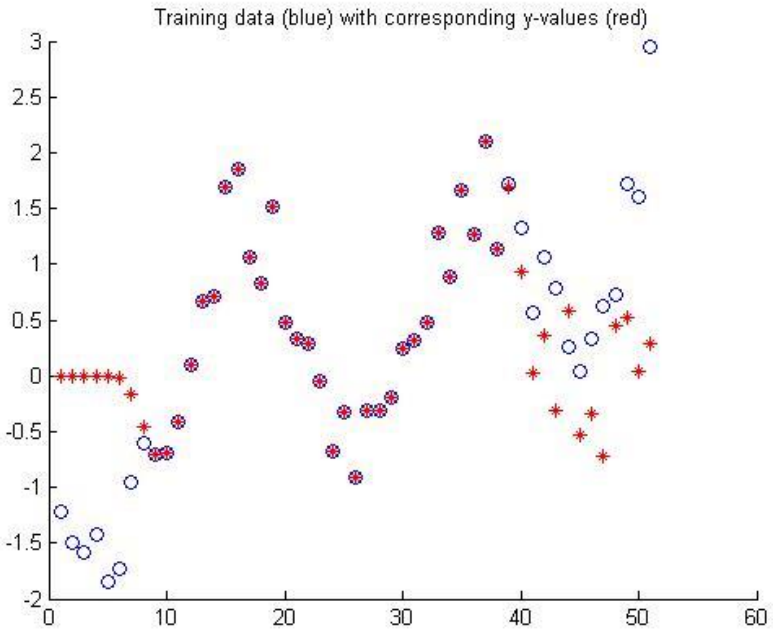
Machine Intelligence I UE 5

Visualization:

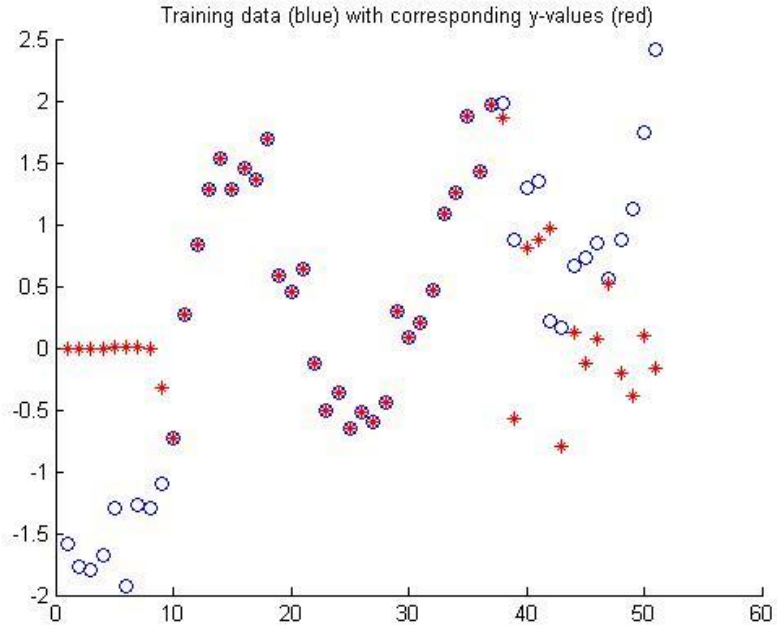
Degree 1



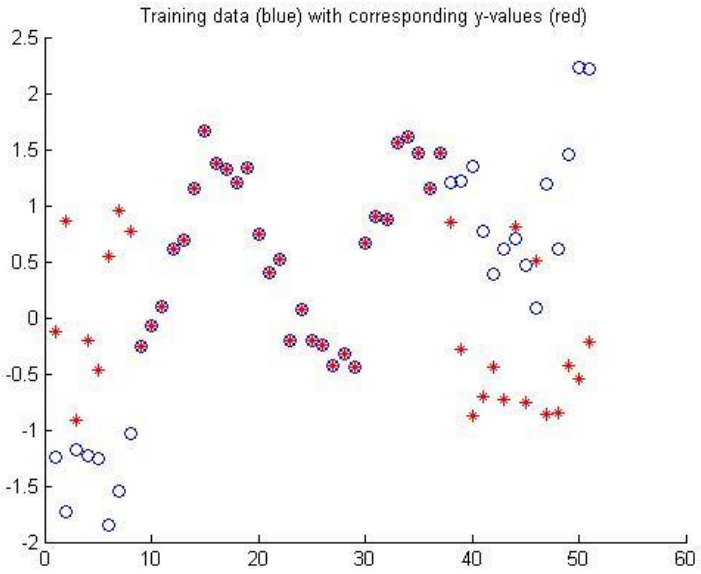
Degree 5



Degree 9



Degree 14



Code:

```
%%%%%%%%%%
%
% Machine Intelligence I UE 5
%
% bias & variance tradeoff & nested CV
%
% Patik Bey
```

```
%
%%%%%%%%


% create data
% x
for i = 1:51
    dat(i) = 0+(i-1)*0.02;
end
dat = dat';
%
% y labels
%
% repeat 40 times
%
for run = 1:40
    for i = 1:51
        y(i,run) = -sin(5*pi*dat(i))+2*(2*dat(i)-1)^3 + rand(1);
    end
end
%
% learning rate eta
%

%%%%
%
% input monomial definition
%
% 1. degree
%
%%%%
eta = 0.5;
w = zeros(length(dat),1000);
w(:,1) = rand(1,length(dat))-0.5;

x = (dat-mean(dat))/std(dat);
for run = 1:40
    for i = 1:1000
        for in = 1:length(x)
            if i == 1
                n(in,run) = tan(x(in))*w(in,i)+1; % f(x)*w-w_0(w_0 == 1)
            else
                n(in,run) = tanh(x(in))*w(in,i)+w(in,i-1);
            end
        w(in,i+1) = w(in,i) - eta*(n(in,run)-y(in,run))*(1-tanh(x(in))^2);
        %w(in,i+1) = w(in,i) - eta*(n(in,run)-y(in,run))'*(n(in,run)-
y(in,run));
        end
        %w(i+1) = w(i) - eta*(w(i)-x*y(:,run));
        e(i,run) = (1/2)*(w(:,i)'*n(:,run)-y(:,run))'*(w(:,i)'*n(:,run)-
y(:,run));

        if abs(e(i,run)) < 0.0001
            break
        end

    end
end

%%%%%
```

```
% b)
% compute average prediction and variance over 40 runs
%
%%%%%%
avg_p = mean(n');
var_p = var(n');

%% bias
bias = (avg_p' - x)'*(avg_p' - x);
%variance
%Mvar = <(y-<y>)^2>, but y := n und <y> as mean(n);!? dim(n) = 51 40 -> not
%fitting

%%%%%%
%
% visualization
%
%%%%%%

figure(1)
plot(n)
hold on
plot(y)

figure()
scatter(1:51,y(:,run))
hold on
scatter(1:51,n(:,run),'r*')
title('Training data (blue) with corresponding y-values (red)')

plot(avg_p)
hold on
plot(var_p,'.')
plot(mean(y),'r')
title('average prediction (blue), its variance (--) and true y values
(red)')
```

4.2

Did not get the CV to run properly.

```
%%%%%%%%%%
%
% Machine Intelligence I UE 5
%
% bias & variance tradeoff & nested CV
%
% Patik Bey
%
%%%%%%%%%

% 10-fold CV

%read Data
clear all

dat = csvread('trainingRidge.csv',1');
```

```
x = dat(:,1:2);

for i = 1:length(x)
    x_neu(i) = (x(i,1)^4)*(x(i,2)^5);
end
clear x
x = x_neu';
x = x-mean(x(:));
x = x/std(x(:));

y = dat(:,3);


%initialization
w = rand(length(x),1)-0.5;

%lambda loop
lambda = 0.001;
eta = 0.01;

%cv loop
cvind = 1:length(x);
index =
[1:20;21:40;41:60;61:80;81:100;101:120;121:140;141:160;161:180;181:200];

%linear connectionist neuron
for ind = 1:10
%cv loop
    for i = 1:3000
        for in = cvind(setdiff(1:end,index(ind,:)))
            if i == 1
                n(in) = w(in,i)*x(in)+1;
            else
                n(in) = w(in,i)*x(in)+w(in,i-1);
            end
            w(in,i+1) = w(in,i) - eta*(n(in)-y(in))*(1-tanh(x(in))^2) -
lambda*0.5*w(in,i)^2;

        end
        e(i) = (1/2)*sum(w(:,i))^2;
            if abs(e(i)) < 0.0001
                break
            end
    end

    %perform prediction
pred(1:length(ind)) = w(index(ind,:),length(w)-1)'*x(index(ind,:));


%compute general performance error
Eg = sum((pred-y(ind))^2);


end
```