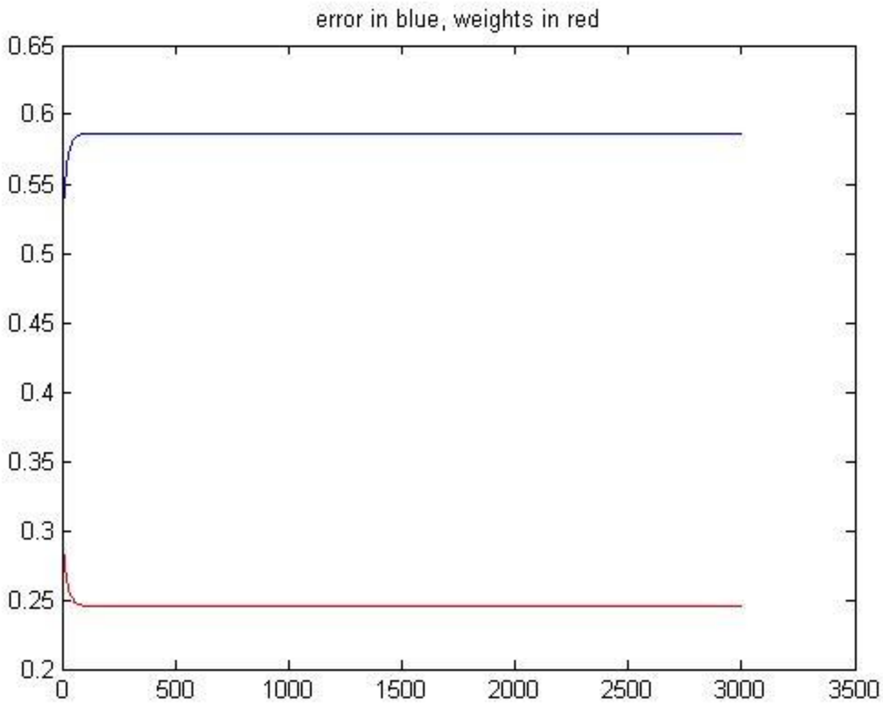
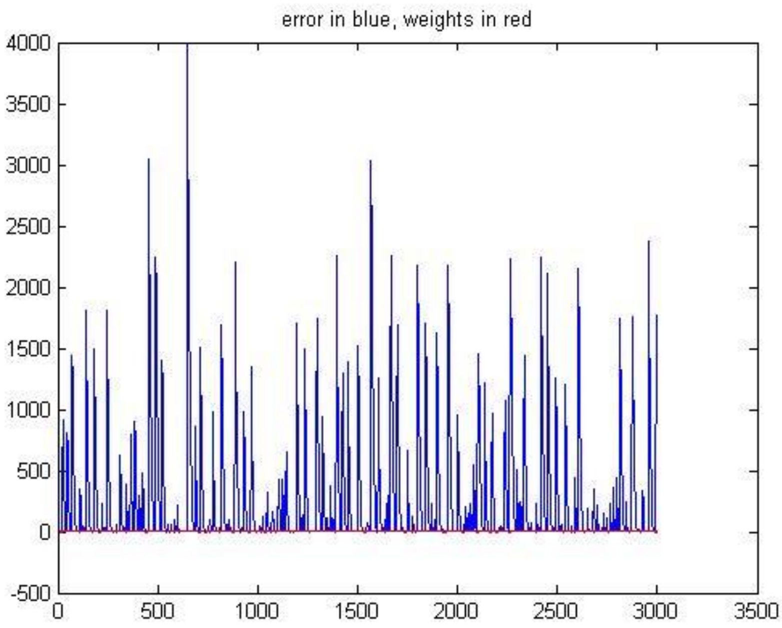
Machine Intelligence I UE 4

Visualization:

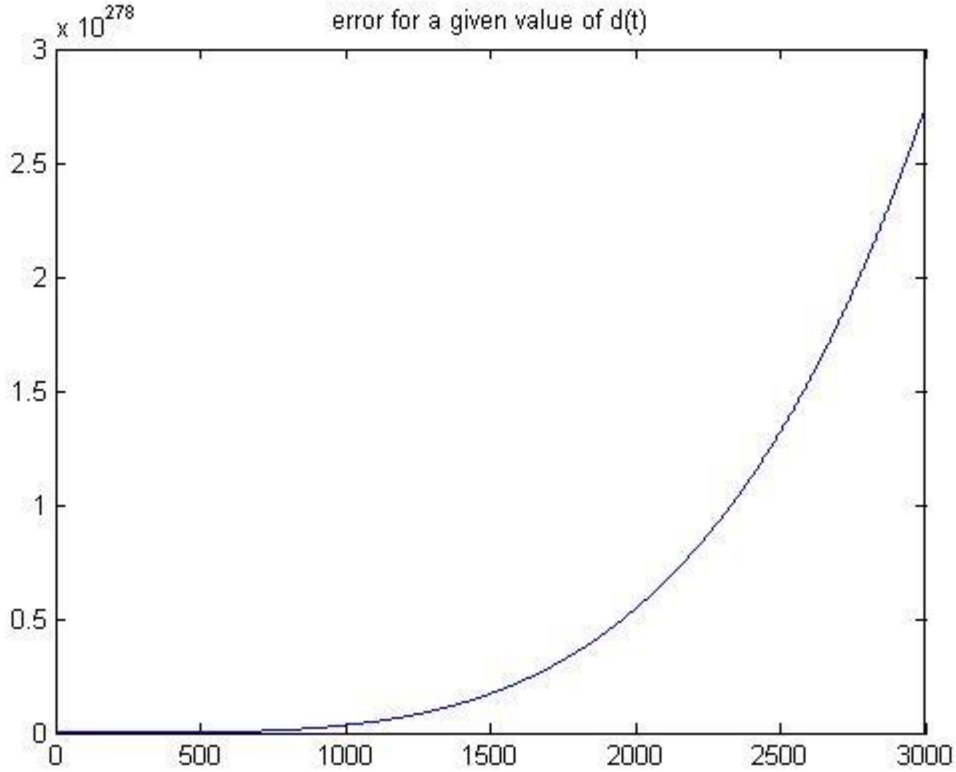
4.1 a)



b)



c)



Obviously these results cannot be correct. Unfortunately I was not able to figure out the problem. I know it has to do with my update of the weights per iteration but cannot find the problem.

Attached is also the code I wrote for the problems.

CODE:

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

%

% Machine Intelligence I UE 4

%

% Gradient methods for parameter optimization

%

% Patik Bey
```

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

%define data set

x = [0,-1,0.3,2]';
t = [1,-0.1,0.5,0.5]';


%%%%%%%%%%%%%%%%%%%%%%%%
%
% 4.1 (a) gradient descent
%
%%%%%%%%%%%%%%%%%%%%%%%%

%%% initilisation

% weights
w(1) = rand(1)-0.5;
w_0 = 1;

% learning rate
eta = 0.01;
% x_0
x_0 = 1;
% weight update iterations it = 1:3000 (including convergence criterion)
for it = 1:3000
% input data loop x(i)
        %for i = 1:length(x)
                % neuron output function
                if it == 1
```

```
                n = w(it)*x + w_0;
                else
                n = w(it)*x + w(it-1);
                %n(i) = w(it)*x(i) + w_0;
                %else
                %n(i) = w(it)*x(i) + w(it-1);
            end
                % sum of square errors
            e(it) = 1/2*(w(it)*n-t)'*(w(it)*n-t);
        %end

        %include convergence criteria
        if abs(e(it)) < 0.0001
            break
        end
        %w(it+1) = w(it) - eta*((n*n')*w(it)-n*t);
        w(it+1) = w(it) - eta*((x'*x)*w(it)-x'*t);



end
% weight trace
scatter(w(1:3000),w(2:3001),'*')

figure(1)
plot(e)
hold on
plot(w,'r')
title('error in blue, weights in red')
figure(2)
scatter(1:length(x),t)
hold on
```

```
scatter(1:length(x),n,'r*')

title('training data (blue) with corresponding output values t (red)')



%%%%%%%%%%%%%%%%%%%%%

%

% 4.1 (b) line search

%

%%%%%%%%%%%%%%%%%%%%%


% weights

w(1) = rand(1)-0.5;

w_0 = 1;


% x_0

x_0 = 1;

% weight update iterations it = 1:3000 (including convergence criterion)

for it = 1:3000

% input data loop x(i)

	%for i = 1:length(x)

		% neuron output function

		if it == 1

			n = w(it)*x + w_0;

			else

			n = w(it)*x + w(it-1);

		end

			% sum of square errors

		e(it) = 1/2*(w(it)*n-t)'*(w(it)*n-t);

	%end


	%include convergence criteria
```

```
        if abs(e(it)) < 0.0001
                break
        end
        %al = (((x'*x)*w(it)-x'*t)*((x'*x)*w(it)-x'*t))/(((x'*x)*w(it)-x'*t)*(x'*x)*((x'*x)*w(it)-x'*t));
        al = (((n'*n)*w(it)-n'*t)*((n'*n)*w(it)-n'*t))/(((n'*n)*w(it)-n'*t)*(n'*n)*((n'*n)*w(it)-n'*t));

        %w(it+1) = w(it) - al*((n'*n)*w(it)-n'*t);
        w(it+1) = w(it) - al*((x'*x)*w(it)-x'*t);



end


figure(1)
plot(e)
hold on
plot(w,'r')
title('error in blue, weights in red')
figure(2)
scatter(1:length(x),t)
hold on
scatter(1:length(x),n,'r*')
title('training data (blue) with corresponding output values t (red)')

%%%%%%%%%%%%%%%%%%%%%%%%
%
% 4.1 (c) conjugate gradient
%
%%%%%%%%%%%%%%%%%%%%%%%%



x = [0,-1,0.3,2]';
```

```
t = [1,-0.1,0.5,0.5]';

w_0 = 1;


% weights
g(1) = (x'*x*w_0-x'*t);
w(1) = -g(1);
d(1) = w(1);

for git = 1:20
        for it = 1:3000
        % input data loop x(i)
                %for i = 1:length(x)
                        % neuron output function
                        if it == 1
                                n = w(it)*x + w_0;
                                else
                                n = w(it)*x + w(it-1);
                        end
                                % sum of square errors
                        e(it) = 1/2*(w(it)*n-t)'*(w(it)*n-t);
                %end

                %include convergence criteria
                if abs(e(it)) < 0.0001
                        break
                end

                al = (d(git)*g(git))/(d(git)*(x'*x)*d(git));
                %w(it+1) = w(it) - al*d(t);
```

```
                w(it+1) = w(it) - al*d(git);
        end
g(git+1) = x'*x*w(it)-x'*t;
d(git+1) = g(git+1) + ((g(git+1)*g(git+1))/(g(git)*g(git)))*d(git);
error(git) = e(it);
end

figure(1)
plot(error)
title('error of given iteration for d(t)')
figure(2)
scatter(1:length(x),t)
hold on
scatter(1:length(x),n,'r*')
title('training data (blue) with corresponding output values t (red)')
```