KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Intelektikos Pagrindai (P176B101)

Pirmojo laboratorinio darbo ataskaita

Atliko:

IFF – 6/8 gr. studentas

Tadas Laurinaitis

2019 m. balandžio 1 d.

Priėmė:

Lekt. Germanas Budnikas

KAUNAS 2019

Scenarijus nr. 1 (1-14 užduotis)

Kodas:

% 1-3 uzduotis

load("sunspot.txt")

% 4 uzduotis

figure(1)

plot(sunspot(:,1), sunspot(:,2),'r-\*')

xlabel("Metai")

ylabel("Demes")

title ("Saules demiu aktyvumo uz 1700-2014 metus grafikas")

% 5 uzduotis

L = length(sunspot); % duomen? kiekis

P = [sunspot(1:L-2,2)'; % ?vesties duomen?

sunspot(2:L-1,2)']; % matrica

T = sunspot(3:L,2)'; % išvesties duomen? vektorius

%duomenu isvedimas i console

disp('Matricos P dydis')

size(P)

disp('Matricos P elementai')

P

disp('Matricos T dydis')

size(T)

disp('Matricos T elementai')

T

% 6 uzduotis

figure(2)

plot3(P(1,:), P(2,:), T, 'bo')

xlabel('Demes n-2 metais');

ylabel('Demes n-1 metais');

zlabel('Demes n metais');

title("Matricu P ir T duomenys")

% 7 uzduotis

Pu = P(:, 1:200);

Tu = T(:, 1:200);

disp('Matricos Pu dydis')

size(Pu)

disp('Matrica Pu')

Pu

disp('Matricos Tu dydis')

size(Tu)

disp('Matrica Tu')

Tu

% 8 uzduotis

% sukuriam neurona ir apskaiciuojam jo svoriu reiksmes tiesioginiu metodu

net = newlind(Pu, Tu);

% 9 uzduotis

disp('neurono svorio koeficientai:' )

disp(net.IW{1})

disp(net.b{1})

%priskiriam svoriu reiksmes pagalbiniam kintamiesiems

w1 = net.IW{1}(1)

w2 = net.IW{1}(2)

b = net.b{1}

% 10 uzduotis

% neurono veikimo imitacija

Tsu = sim(net,Pu)

figure(3)

hold on;

grid on;

plot(sunspot(3:202, 1), Tu, 'r-o');

plot(sunspot(3:202, 1), Tsu, 'b-o');

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-1901 metais');

% 11 uzduotis

% neurono veikimo imitacija

Ts = sim(net,P)

figure(4)

hold on;

grid on;

plot(sunspot(3:315, 1), T, 'r-o')

plot(sunspot(3:315, 1), Ts, 'b-o')

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-2014 metais');

% 12 uzduotis

e = (T-Ts)';

figure(5)

grid on;

plot(sunspot(3:315), e, 'b-o')

xlabel('Metai');

ylabel('Tikrosios ir prognozuojamos prognozes skirtumas');

title('Prognozes klaidu grafikas');

% 13 uzduotis

figure(6);

hist(e);

xlabel('Prognozes klaidos reiksme');

ylabel('Daznis');

title('Prognozes klaidu histograma');

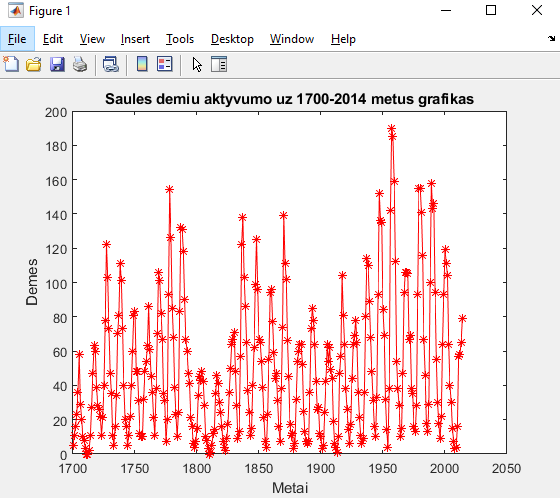
% 14 uzduotis

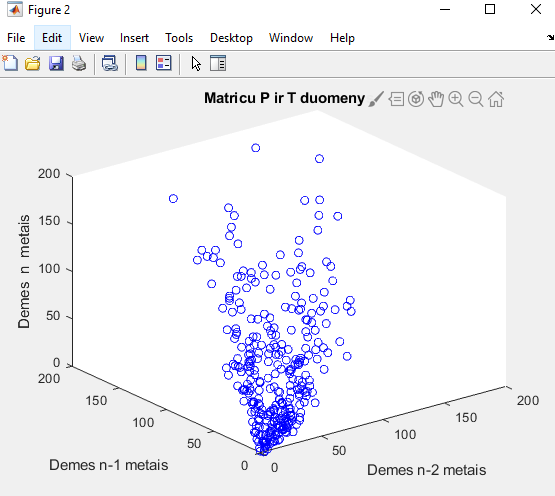
disp('Vidutinës kvadratinës prognozës klaidos reik?më')

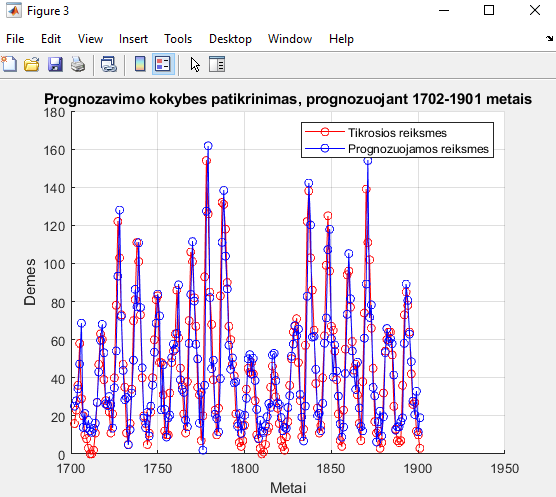
mse\_reiksme = mse(e)

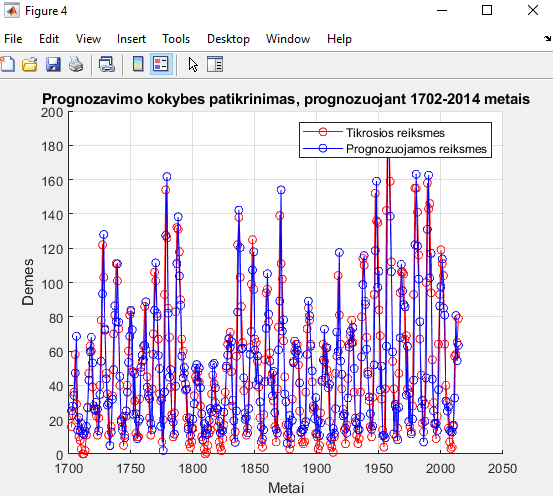
clear()

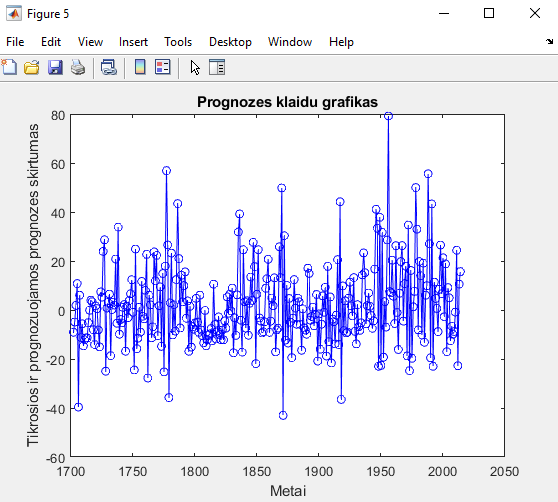
Grafikai:

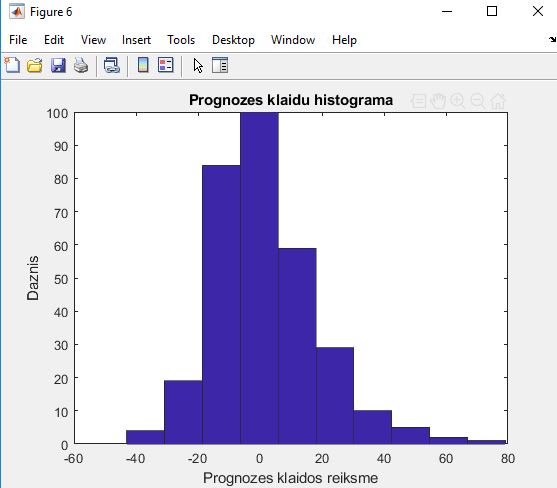












Scenarijus nr. 2 (15-19 užduotis)

Kodas:

% 1-3 uzduotis

load("sunspot.txt")

% 4 uzduotis

figure(1)

plot(sunspot(:,1), sunspot(:,2),'r-\*')

xlabel("Metai")

ylabel("Demes")

title ("Saules demiu aktyvumo uz 1700-2014 metus grafikas")

% 5 uzduotis

L = length(sunspot); % duomen? kiekis

P = [sunspot(1:L-2,2)'; % ?vesties duomen?

sunspot(2:L-1,2)']; % matrica

T = sunspot(3:L,2)'; % išvesties duomen? vektorius

%duomenu isvedimas i console

disp('Matricos P dydis')

size(P)

disp('Matricos P elementai')

P

disp('Matricos T dydis')

size(T)

disp('Matricos T elementai')

T

% 6 uzduotis

figure(2)

plot3(P(1,:), P(2,:), T, 'bo')

xlabel('Demes n-2 metais');

ylabel('Demes n-1 metais');

zlabel('Demes n metais');

title("Matricu P ir T duomenys")

% 7 uzduotis

Pu = P(:, 1:200);

Tu = T(:, 1:200);

disp('Matricos Pu dydis')

size(Pu)

disp('Matrica Pu')

Pu

disp('Matricos Tu dydis')

size(Tu)

disp('Matrica Tu')

Tu

% 8 uzduotis

% sukuriam neurona ir apskaiciuojam jo svoriu reiksmes tiesioginiu metodu

%net = newlind(Pu, Tu);

% 15 uzduotis

S = 1

lr = 0.0000001;

% 16 uzduotis

net = newlin(Pu,S,0,lr)

disp('neurono svorio koeficientai:' )

disp(net.IW{1})

disp(net.b{1})

%priskiriam svoriu reiksmes pagalbiniam kintamiesiems

w1 = net.IW{1}(1)

w2 = net.IW{1}(2)

b = net.b{1}

% 17 uzduotis

net.trainParam.goal = 100;

net.trainParam.epochs = 1000;

% 18 uzduotis

net = train(net, Pu, Tu)

disp('neurono svorio koeficientai:' )

disp(net.IW{1})

disp(net.b{1})

%svoriu reiksmiu priskyrimas

w3 = net.IW{1}(1)

w4 = net.IW{1}(2)

bb = net.b{1}

% 10 uzduotis

% neurono veikimo imitacija

Tsu = sim(net,Pu)

figure(3)

hold on;

grid on;

plot(sunspot(3:202, 1), Tu, 'r-o');

plot(sunspot(3:202, 1), Tsu, 'b-o');

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-1901 metais');

% 11 uzduotis

% neurono veikimo imitacija

Ts = sim(net,P)

figure(4)

hold on;

grid on;

plot(sunspot(3:315, 1), T, 'r-o')

plot(sunspot(3:315, 1), Ts, 'b-o')

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-2014 metais');

% 12 uzduotis

e = (T-Ts)';

figure(5)

grid on;

plot(sunspot(3:315), e, 'b-o')

xlabel('Metai');

ylabel('Tikrosios ir prognozuojamos prognozes skirtumas');

title('Prognozes klaidu grafikas');

% 13 uzduotis

figure(6);

hist(e);

xlabel('Prognozes klaidos reiksme');

ylabel('Daznis');

title('Prognozes klaidu histograma');

% 14 uzduotis

disp('Vidutines kvadratines prognozes klaidos reiksme')

mse\_reiksme = mse(e)

clear()

% 19 uzduotis

% Klausimai:

% 1. Ka pavaizduoja diagrama, kuri vaizduojama mokymosi proceso metu?

% Diagrama pavaizduoja dirbtinio neurono modeli

% 2. Ar mokymosi procesas yra konverguojantis? Jeigu ne, pamastyti kas gali b?ti priezastimi ir pakeisti atitinkama parametra.

% Nekonveguojantis, galbut pavyzdziu kiekio pakeitimas???

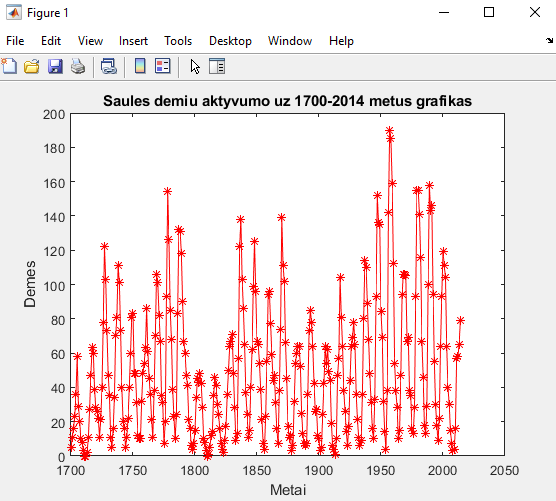
% 3. Kokios yra naujos neurono svoriu koeficientu reiksm?s ?

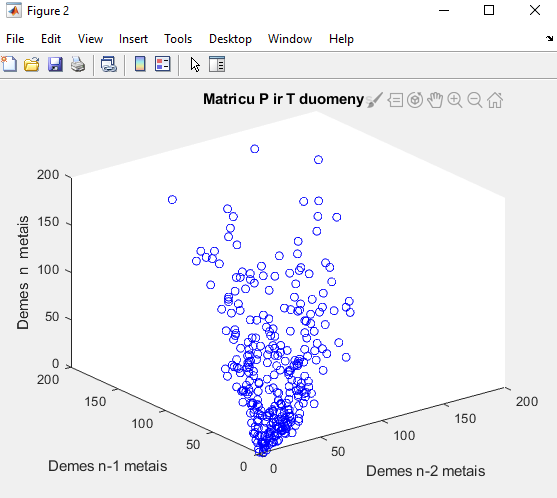
% Naujos gautos reiksmes yra toliau 0 negu senos

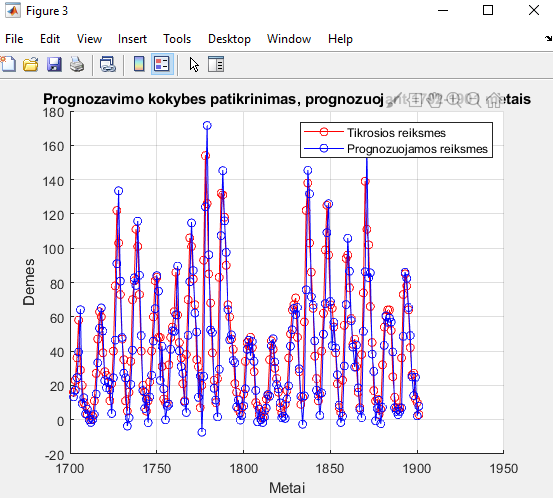
% 4. Kokia yra neurono darbo kokybes ivercio – vidutinis kvadratinis nuokrypis – reiksme ?

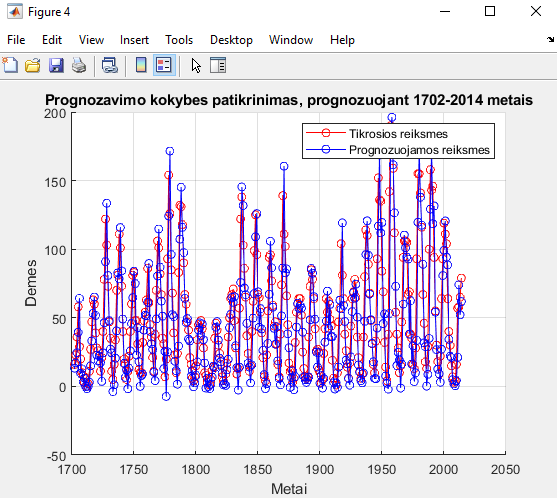
% 357.2350

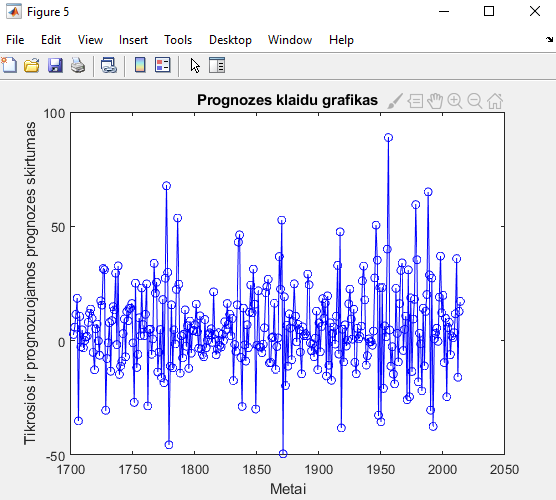
Grafikai:

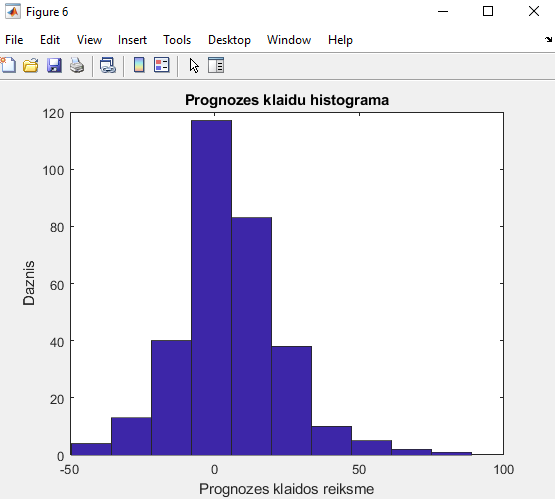












Scenarijus nr. 3 (21 užduotis)

Kodas:

% 1-3 uzduotis

load("sunspot.txt")

% 4 uzduotis

figure(1)

plot(sunspot(:,1), sunspot(:,2),'r-\*')

xlabel("Metai")

ylabel("Demes")

title ("Saules demiu aktyvumo uz 1700-2014 metus grafikas")

% 5 uzduotis

L = length(sunspot) % duomen? kiekis

P = [sunspot(1:L-6,2)'; % n = 6

sunspot(2:L-5,2)';

sunspot(3:L-4,2)';

sunspot(4:L-3,2)';

sunspot(5:L-2,2)';

sunspot(6:L-1,2)'];

T = sunspot(7:L,2)';

%duomenu isvedimas i console

disp('Matricos P dydis')

size(P)

disp('Matricos P elementai')

P

disp('Matricos T dydis')

size(T)

disp('Matricos T elementai')

T

% 7 uzduotis

Pu = P(:, 1:200);

Tu = T(:, 1:200);

disp('Matricos Pu dydis')

size(Pu)

disp('Matrica Pu')

Pu

disp('Matricos Tu dydis')

size(Tu)

disp('Matrica Tu')

Tu

% 8 uzduotis

% sukuriam neurona ir apskaiciuojam jo svoriu reiksmes tiesioginiu metodu

%net = newlind(Pu, Tu);

% 15 uzduotis

S = 1

lr = 0.0000001;

% 16 uzduotis

net = newlin(Pu,S,0,lr)

% 17 uzduotis

net.trainParam.goal = 100;

net.trainParam.epochs = 1000;

% 18 uzduotis

net = train(net, Pu, Tu)

disp('neurono svorio koeficientai:' )

disp(net.IW{1})

disp(net.b{1})

%svoriu reiksmiu priskyrimas

w1 = net.IW{1}(1)

w2 = net.IW{1}(2)

b = net.b{1}

% 10 uzduotis

% neurono veikimo imitacija

Tsu = sim(net,Pu)

figure(3)

hold on;

grid on;

plot(sunspot(7:206, 1), Tu, 'r-o');

plot(sunspot(7:206, 1), Tsu, 'b-o');

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-1901 metais');

% 11 uzduotis

% neurono veikimo imitacija

Ts = sim(net, P)

figure(4)

hold on;

grid on;

plot(sunspot(7:315, 1), T, 'r-o')

plot(sunspot(7:315, 1), Ts, 'b-o')

xlabel('Metai');

ylabel('Demes');

legend('Tikrosios reiksmes', 'Prognozuojamos reiksmes');

title('Prognozavimo kokybes patikrinimas, prognozuojant 1702-2014 metais');

% 12 uzduotis

e = (T-Ts)'

figure(5)

grid on;

plot(sunspot(7:315), e, 'b-o')

xlabel('Metai');

ylabel('Tikrosios ir prognozuojamos prognozes skirtumas');

title('Prognozes klaidos grafikass');

% 13 uzduotis

figure(6);

hist(e);

xlabel('Prognozes klaidos reiksme');

ylabel('Daznis');

title('Prognozes klaidu histograma');

% 14 uzduotis

disp('Vidutines kvadratines prognozes klaidos reiksme')

mse\_reiksme = mse(e)

clear()

% 19 uzduotis

% Klausimai:

% 1. Ka pavaizduoja diagrama, kuri vaizduojama mokymosi proceso metu?

% Diagrama pavaizduoja dirbtinio neurono modeli

% 2. Ar mokymosi procesas yra konverguojantis? Jeigu ne, pamastyti kas gali b?ti priezastimi ir pakeisti atitinkama parametra.

% Nekonveguojantis, galbut pavyzdziu kiekio pakeitimas???

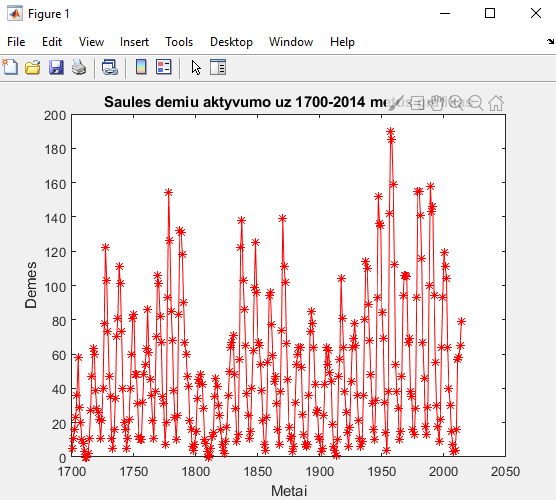
% 3. Kokios yra naujos neurono svoriu koeficientu reiksm?s ?

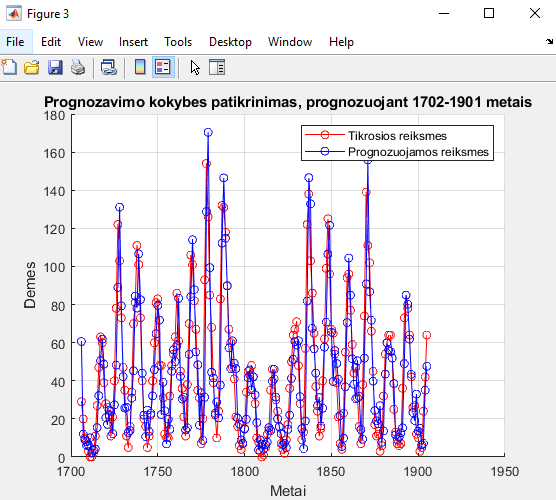
% Naujos gautos reiksmes yra toliau 0 negu senos

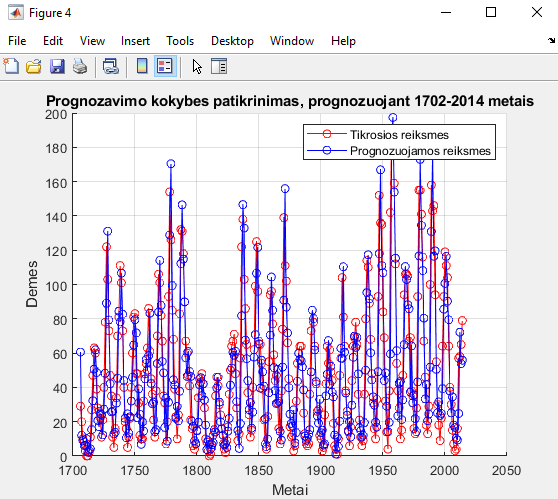
% 4. Kokia yra neurono darbo kokybes ivercio – vidutinis kvadratinis nuokrypis – reiksme ?

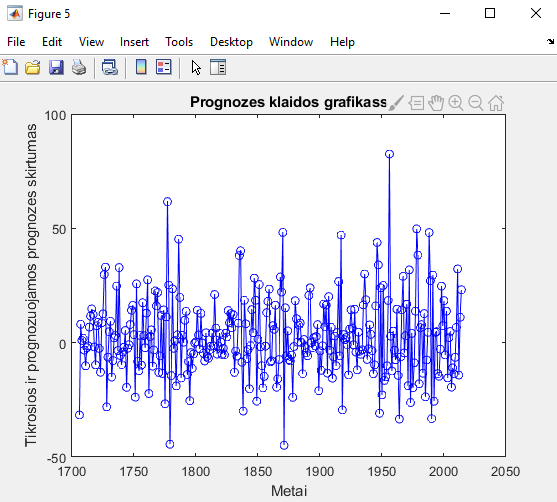
% 357.2350

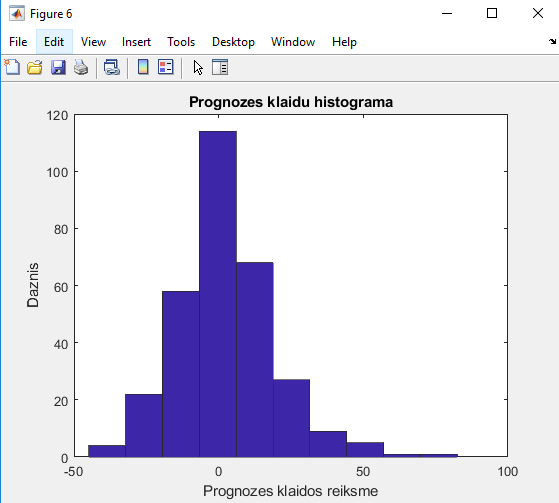
Grafikai:











Scenarijus nr. 4 (Papildoma užduotis)

Kodas:

%% Crab Classification

% This example illustrates using a neural network as a classifier to

% identify the sex of crabs from physical dimensions of the crab.

%% The Problem: Classification of Crabs

% In this example we attempt to build a classifier that can identify the

% sex of a crab from its physical measurements. Six physical

% characteristics of a crab are considered: species, frontallip, rearwidth,

% length, width and depth. The problem on hand is to identify the sex of a

% crab given the observed values for each of these 6 physical

% characteristics.

%

%% Why Neural Networks?

% Neural networks have proven themselves as proficient classifiers and are

% particularly well suited for addressing non-linear problems. Given the

% non-linear nature of real world phenomena, like crab classification,

% neural networks are certainly a good candidate for solving the problem.

%

% The six physical characteristics will act as inputs to a neural network

% and the sex of the crab will be the target. Given an input, which

% constitutes the six observed values for the physical characteristics of a

% crab, the neural network is expected to identify if the crab is male or

% female.

%

% This is achieved by presenting previously recorded inputs to a neural

% network and then tuning it to produce the desired target outputs. This

% process is called neural network training.

%

%% Preparing the Data

% Data for classification problems are set up for a neural network by

% organizing the data into two matrices, the input matrix X and the target

% matrix T.

%

% Each ith column of the input matrix will have six elements representing a

% crab's species, frontallip, rearwidth, length, width, and depth.

%

% Each corresponding column of the target matrix will have two elements.

% Female crabs are represented with a one in the first element, male crabs

% with a one in the second element. (All other elements are zero).

%

% Here the dataset is loaded.

[x,t] = crab\_dataset;

size(x)

size(t)

%% Building the Neural Network Classifier

% The next step is to create a neural network that will learn to identify

% the sex of the crabs.

%

% Since the neural network starts with random initial weights, the results

% of this example will differ slightly every time it is run. The random

% seed is set to avoid this randomness. However this is not necessary for

% your own applications.

setdemorandstream(491218382)

%%

% Two-layer (i.e. one-hidden-layer) feed forward neural networks can learn

% any input-output relationship given enough neurons in the hidden layer.

% Layers which are not output layers are called hidden layers.

%

% We will try a single hidden layer of 10 neurons for this example. In

% general, more difficult problems require more neurons, and perhaps more

% layers. Simpler problems require fewer neurons.

%

% The input and output have sizes of 0 because the network has not yet been

% configured to match our input and target data. This will happen when the

% network is trained.

net = patternnet(10);

view(net)

%%

% Now the network is ready to be trained. The samples are automatically

% divided into training, validation and test sets. The training set is used

% to teach the network. Training continues as long as the network continues

% improving on the validation set. The test set provides a completely

% independent measure of network accuracy.

[net,tr] = train(net,x,t);

nntraintool

%%

% To see how the network's performance improved during training, either

% click the "Performance" button in the training tool, or call PLOTPERFORM.

%

% Performance is measured in terms of mean squared error, and is shown in a

% log scale. It rapidly decreased as the network was trained.

%

% Performance is shown for each of the training, validation and test sets.

plotperform(tr)

%% Testing the Classifier

% The trained neural network can now be tested with the testing samples.

% This will give us a sense of how well the network will do when applied to

% data from the real world.

%

% The network outputs will be in the range 0 to 1, so we can use \*vec2ind\*

% function to get the class indices as the position of the highest element

% in each output vector.

testX = x(:,tr.testInd);

testT = t(:,tr.testInd);

testY = net(testX);

testIndices = vec2ind(testY)

%%

% One measure of how well the neural network has fit the data is the

% confusion plot. Here the confusion matrix is plotted across all samples.

%

% The confusion matrix shows the percentages of correct and incorrect

% classifications. Correct classifications are the green squares on the

% matrices diagonal. Incorrect classifications form the red squares.

%

% If the network has learned to classify properly, the percentages in the

% red squares should be very small, indicating few misclassifications.

%

% If this is not the case then further training, or training a network with

% more hidden neurons, would be advisable.

plotconfusion(testT,testY)

%%

% Here are the overall percentages of correct and incorrect classification.

[c,cm] = confusion(testT,testY)

fprintf('Percentage Correct Classification : %f%%\n', 100\*(1-c));

fprintf('Percentage Incorrect Classification : %f%%\n', 100\*c);

%%

% Another measure of how well the neural network has fit data is the

% receiver operating characteristic plot. This shows how the false

% positive and true positive rates relate as the thresholding of outputs is

% varied from 0 to 1.

%

% The farther left and up the line is, the fewer false positives need to be

% accepted in order to get a high true positive rate. The best classifiers

% will have a line going from the bottom left corner, to the top left

% corner, to the top right corner, or close to that.

plotroc(testT,testY)

%%

% This example illustrated using a neural network to classify crabs.

%

% Explore other examples and the documentation for more insight into neural

% networks and their applications.

%%

% Copyright 2012 The MathWorks, Inc.

Grafikai:

