**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

INTELEKTIKOS PAGRINDAI (P176B101)

**2 laboratorinio darbo ataskaita.**

Atliko:

IFF-4/1 gr. studentas

Mangirdas Kazlauskas

Priėmė:

Doc. Germanas Budnikas

**KAUNAS 2017**

1. Užduotis „saulės dėmės“

## Programos kodas (1 dalis, iki 15 užduoties scenarijaus punkto)

clc, clear, close all;

load sunspot.txt;

% sunspot % patikrinti, ar užkrovė sunspot

% % clear sunspot

% Saulės dėmiu grafiko braižymas

figure(1);

plot(sunspot(:, 1), sunspot(:, 2), 'b-o');

grid on;

xlabel('Metai');

ylabel('Demiu skaicius');

title('Saules demiu aktyvumas 1700-1950 metais');

% P ir T matricų aprašymas

L = length(sunspot);

disp('P matrica (įvesties duomenys):')

P = [sunspot(1:L-2, 2)'; sunspot(2:L-1, 2)'] % Įvestys

disp('P matricos dydis:')

size(P)

disp('T matrica (išvesties duomenys):')

T = sunspot(3:L, 2)' % Išvestys

disp('T matricos dydis:')

size(T)

figure(2)

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

grid on;

xlabel('Saules demiu skaicius (n-2)-aisiais metais');

ylabel('Saules demiu skaicius (n-1)-taisiais metais');

zlabel('Saules demiu skaicius n-aisiais metais');

title('Saules demiu prognoziu, remiantis 2 ankstesniais metais, diagrama');

% Apmokymas

disp('Pu matrica (apmokymo duomenys):')

Pu = P(:, 1:200)

disp('Tu matrica (apmokymo rezultatai):')

Tu = T(:, 1:200)

disp('Pu matricos dydis:')

size(Pu)

disp('Tu matricos dydis:')

size(Tu)

net = newlind(Pu, Tu);

disp('Neurono svorio koeficientai:');

disp(net.IW{1});

disp('Neurono bias reikšmė:');

disp(net.b{1});

% svorio koeficientų reikšmių priskyrimas

w1 = net.IW{1}(1)

w2 = net.IW{1}(2)

b = net.b{1}

% modelio verifikacija (prognozuojamų reikšmių radimas) 1702-1901

Tsu = sim(net, Pu) % Neurono veikimo simuliacija

figure(3), hold on;

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

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

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title('Saules demiu prognozavimo kokybes patikrinimas 1702-1901 metams');

% modelio verifikacija 1702-2014

Ts = sim(net, P);

figure(4), hold on;

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

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

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title('Saules demiu prognozavimo kokybes patikrinimas 1702-2014 metams');

% Prognozės klaidų vektoriaus sukūrimas

disp('Prognozės klaidų vektorius')

e = T - Ts

figure(5);

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

grid on;

title('Prognozes klaidos grafikas 1702-2014 metams');

xlabel('Metai');

ylabel('Demiu skaiciaus skirtumas tarp tikrosios ir prognozuojamos reiksmiu');

% Prognozės klaidų histograma

figure(6);

hist(e);

title('Prognozes klaidu histograma');

xlabel('Prognozes klaidos reiksme');

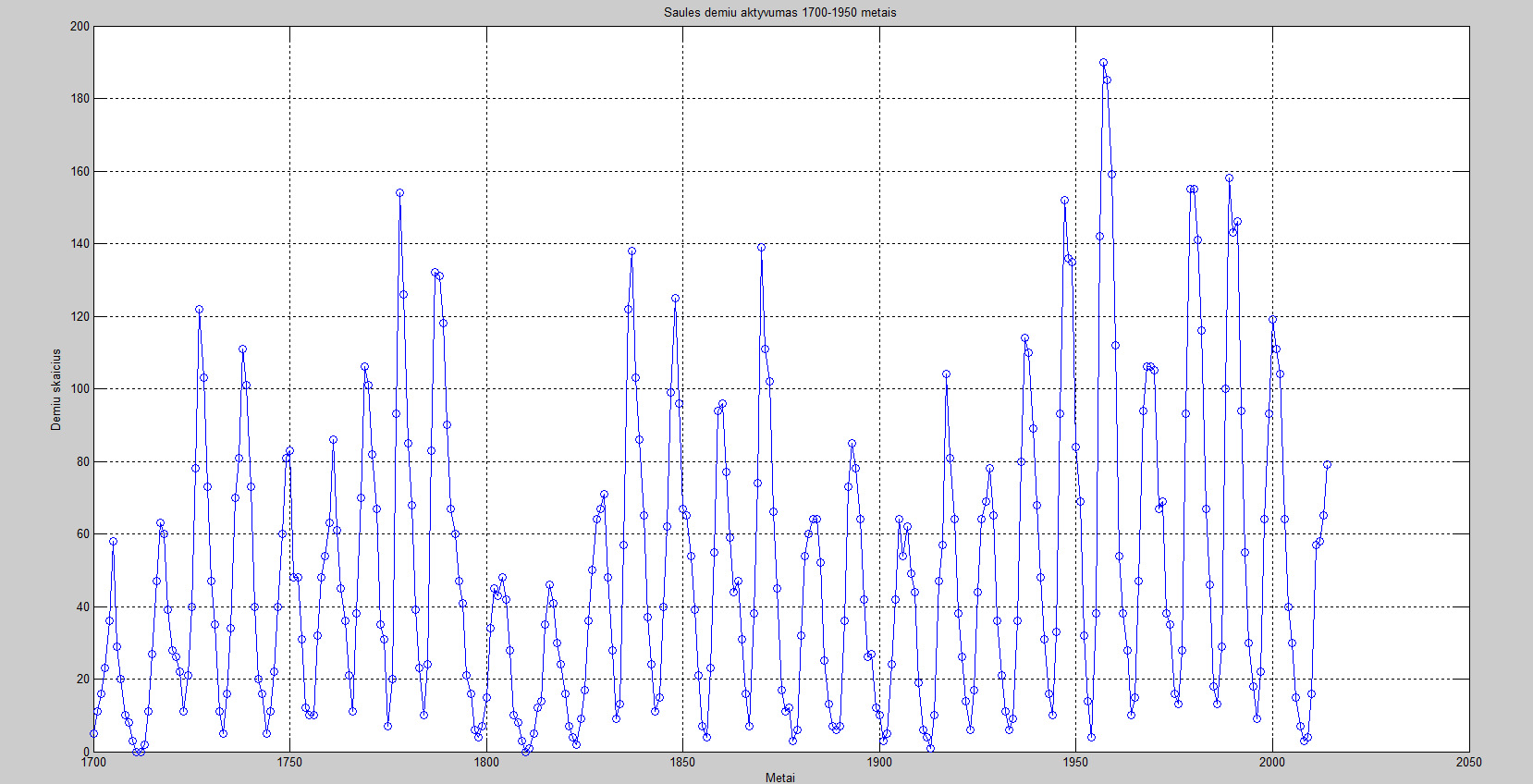
ylabel('Daznis');

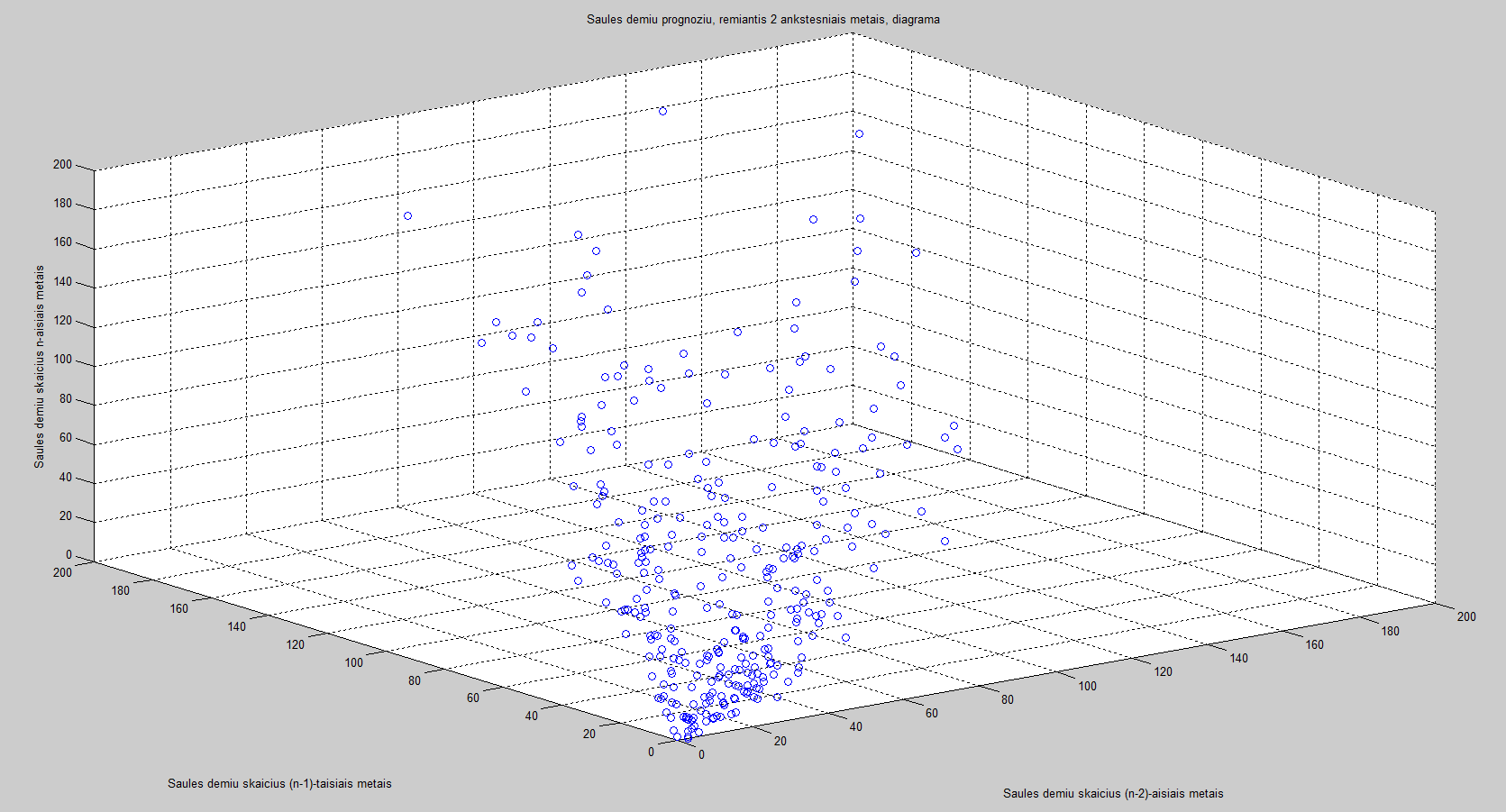
% Vidutinės kvadratinės prognozės klaidos reikšmės skaičiavimas

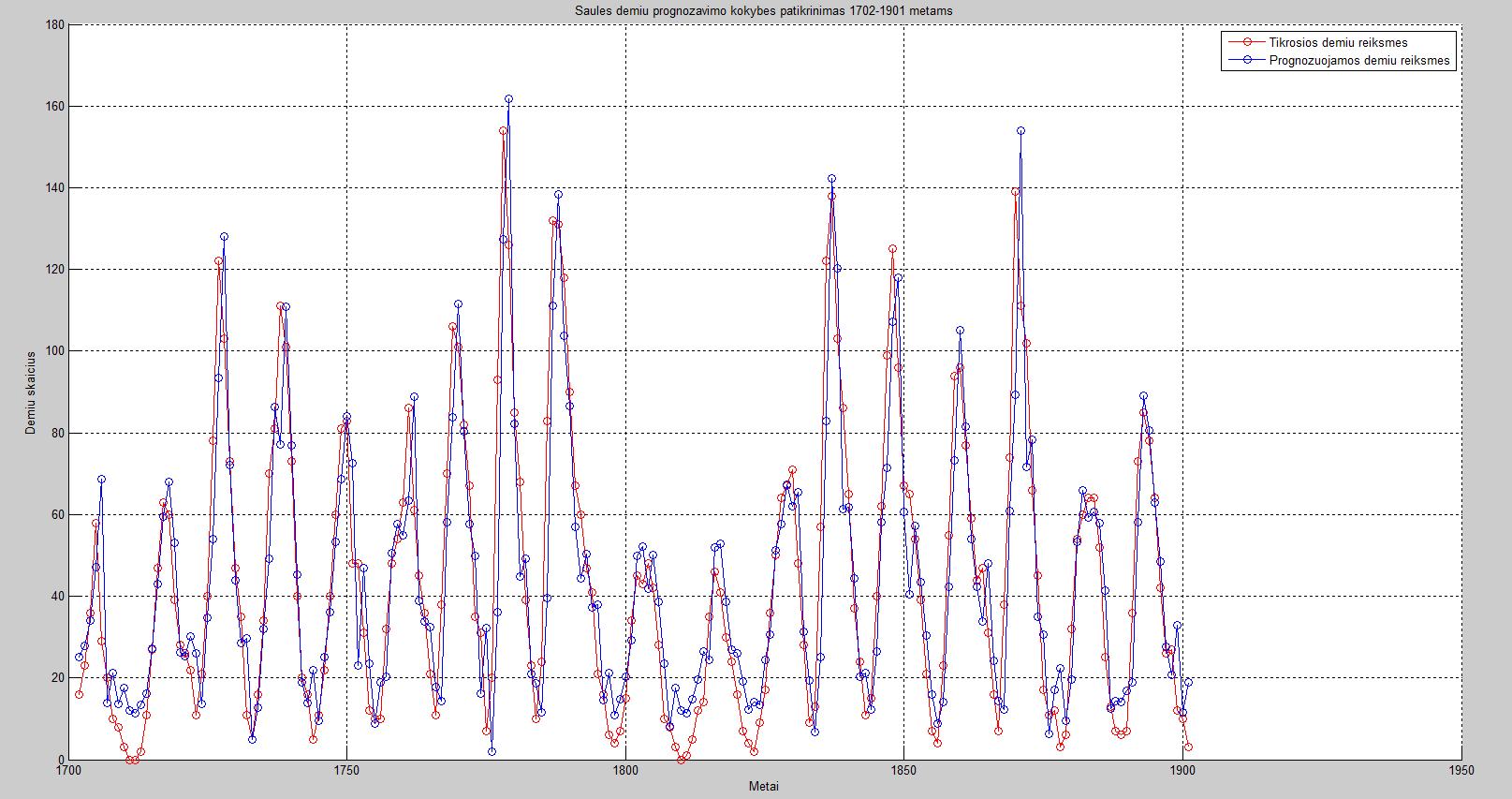
disp('Vidutinės kvadratinės prognozės klaidos reikšmė')

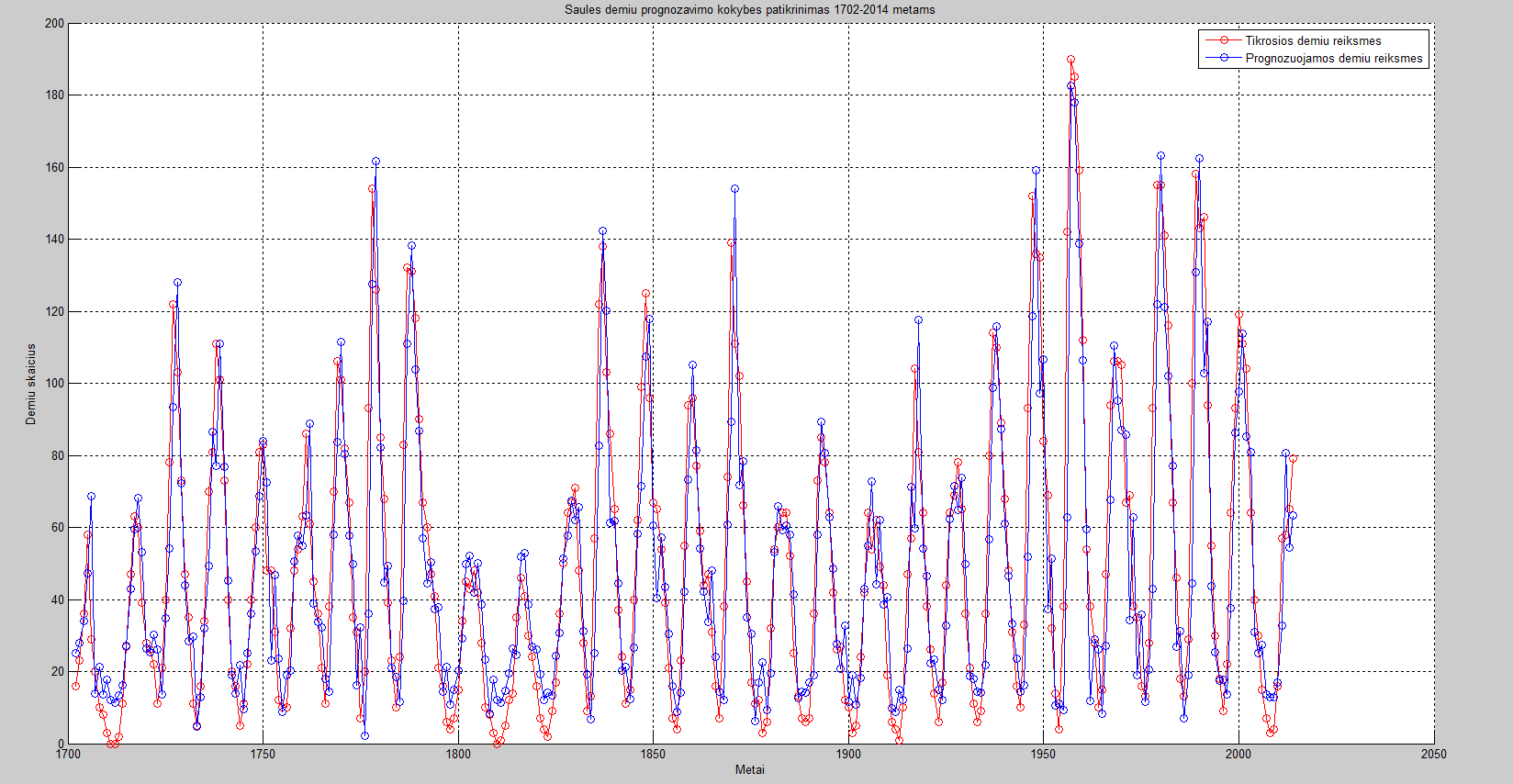
mse\_reiksme = mse(e)

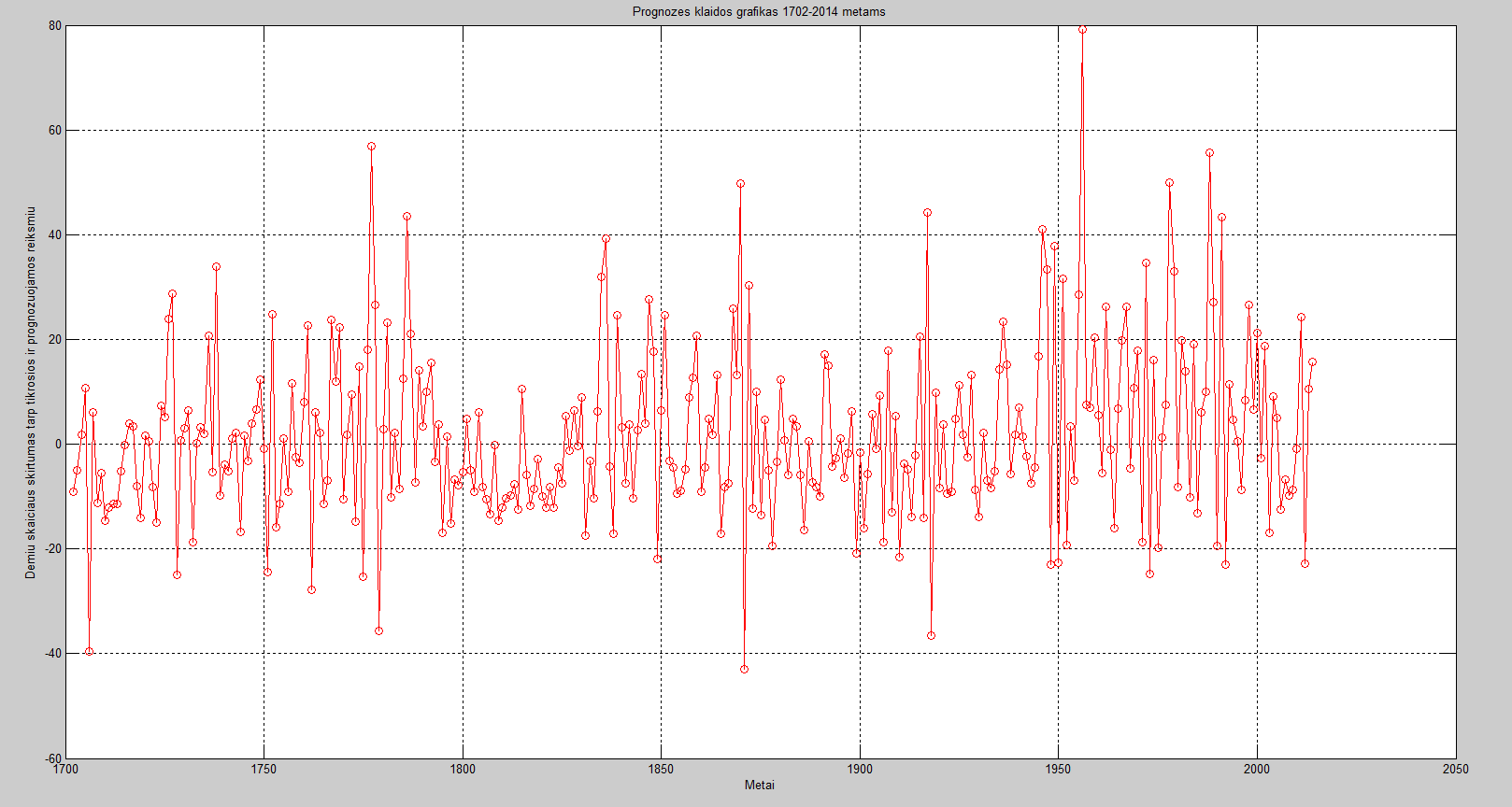
## PROGRAMOS VYKDYMAS (DIAGRAMOS)

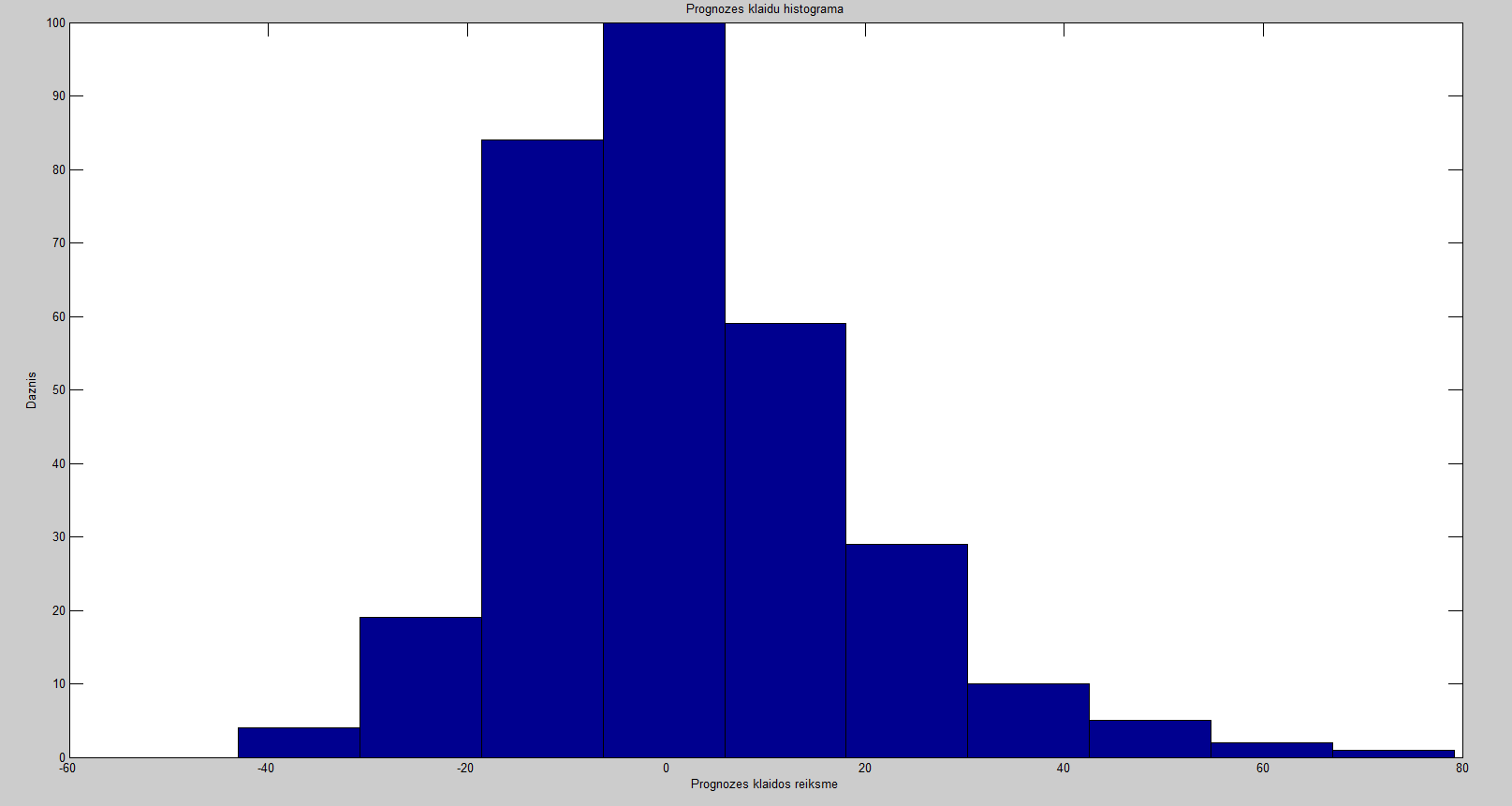












## PROGRAMOS KODAS (2 DALIS, IKI 21 UŽDUOTIES SCENARIJAUS PUNKTO)

clc, clear, close all;

load sunspot.txt;

% sunspot % patikrinti, ar užkrovė sunspot

% % clear sunspot

% Saulės dėmiu grafiko braižymas

figure(1);

plot(sunspot(:, 1), sunspot(:, 2), 'b-o');

grid on;

xlabel('Metai');

ylabel('Demiu skaicius');

title('Saules demiu aktyvumas 1700-1950 metais');

% P ir T matricų aprašymas

L = length(sunspot);

disp('P matrica (įvesties duomenys):')

P = [sunspot(1:L-2, 2)'; sunspot(2:L-1, 2)'] % Įvestys

disp('P matricos dydis:')

size(P)

disp('T matrica (išvesties duomenys):')

T = sunspot(3:L, 2)' % Išvestys

disp('T matricos dydis:')

size(T)

figure(2)

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

grid on;

xlabel('Saules demiu skaicius (n-2)-aisiais metais');

ylabel('Saules demiu skaicius (n-1)-taisiais metais');

zlabel('Saules demiu skaicius n-aisiais metais');

title('Saules demiu prognoziu, remiantis 2 ankstesniais metais, diagrama');

% Apmokymas

disp('Pu matrica (apmokymo duomenys):')

Pu = P(:, 1:200)

disp('Tu matrica (apmokymo rezultatai):')

Tu = T(:, 1:200)

disp('Pu matricos dydis:')

size(Pu)

disp('Tu matricos dydis:')

size(Tu)

% -----------------------------------------------------------------------

% ANTROS DALIES PAKEITIMAI

% -----------------------------------------------------------------------

% net = newlind(Pu, Tu);

fprintf('Maksimali leistina mokymosi proceso greičio koeficiento lr reikšmė: %d\n', maxlinlr(Pu));

net = newlin(Pu, Tu, 0, maxlinlr(Pu));

net.trainParam.goal = 200;

net.trainParam.epochs = 1000;

net = train(net, Pu, Tu);

disp('Neurono svorio koeficientai:');

disp(net.IW{1});

disp('Neurono bias reikšmė:');

disp(net.b{1});

% svorio koeficientų reikšmių priskyrimas

w1 = net.IW{1}(1)

w2 = net.IW{1}(2)

b = net.b{1}

% modelio verifikacija (prognozuojamų reikšmių radimas) 1702-1901

Tsu = net(Pu) % Neurono veikimo simuliacija

figure(3), hold on;

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

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

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title('Saules demiu prognozavimo kokybes patikrinimas 1702-1901 metams');

% modelio verifikacija 1702-2014

Ts = net(P);

figure(4), hold on;

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

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

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title('Saules demiu prognozavimo kokybes patikrinimas 1702-2014 metams');

% Prognozės klaidų vektoriaus sukūrimas

disp('Prognozės klaidų vektorius')

e = T - Ts

figure(5);

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

grid on;

title('Prognozes klaidos grafikas 1702-2014 metams');

xlabel('Metai');

ylabel('Demiu skaiciaus skirtumas tarp tikrosios ir prognozuojamos reiksmiu');

% Prognozės klaidų histograma

figure(6);

hist(e);

title('Prognozes klaidu histograma');

xlabel('Prognozes klaidos reiksme');

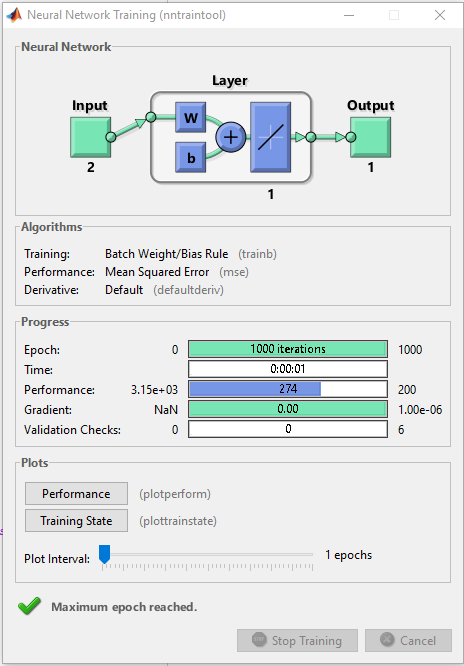
ylabel('Daznis');

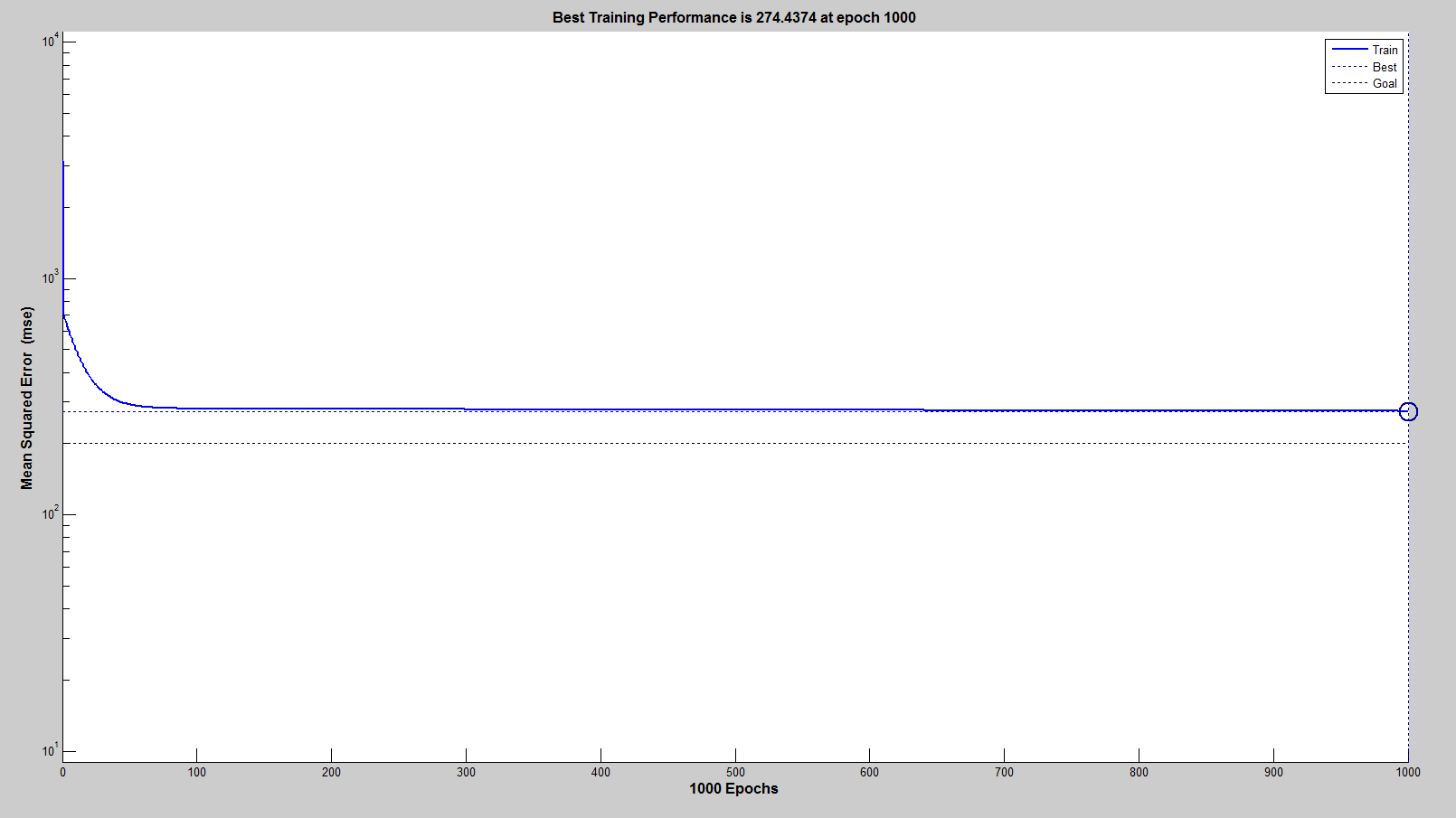
% Vidutinės kvadratinės prognozės klaidos reikšmės skaičiavimas

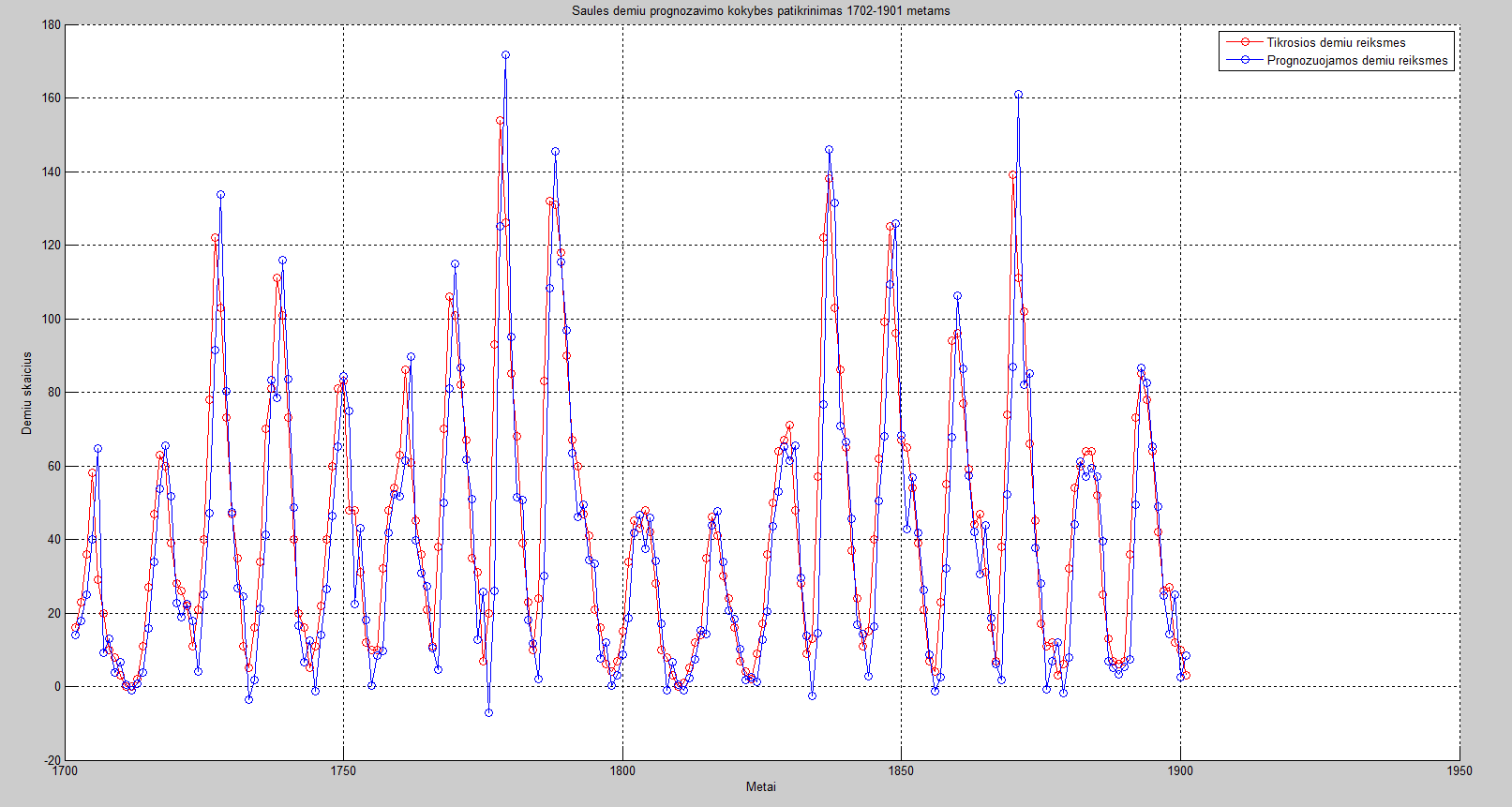
disp('Vidutinės kvadratinės prognozės klaidos reikšmė')

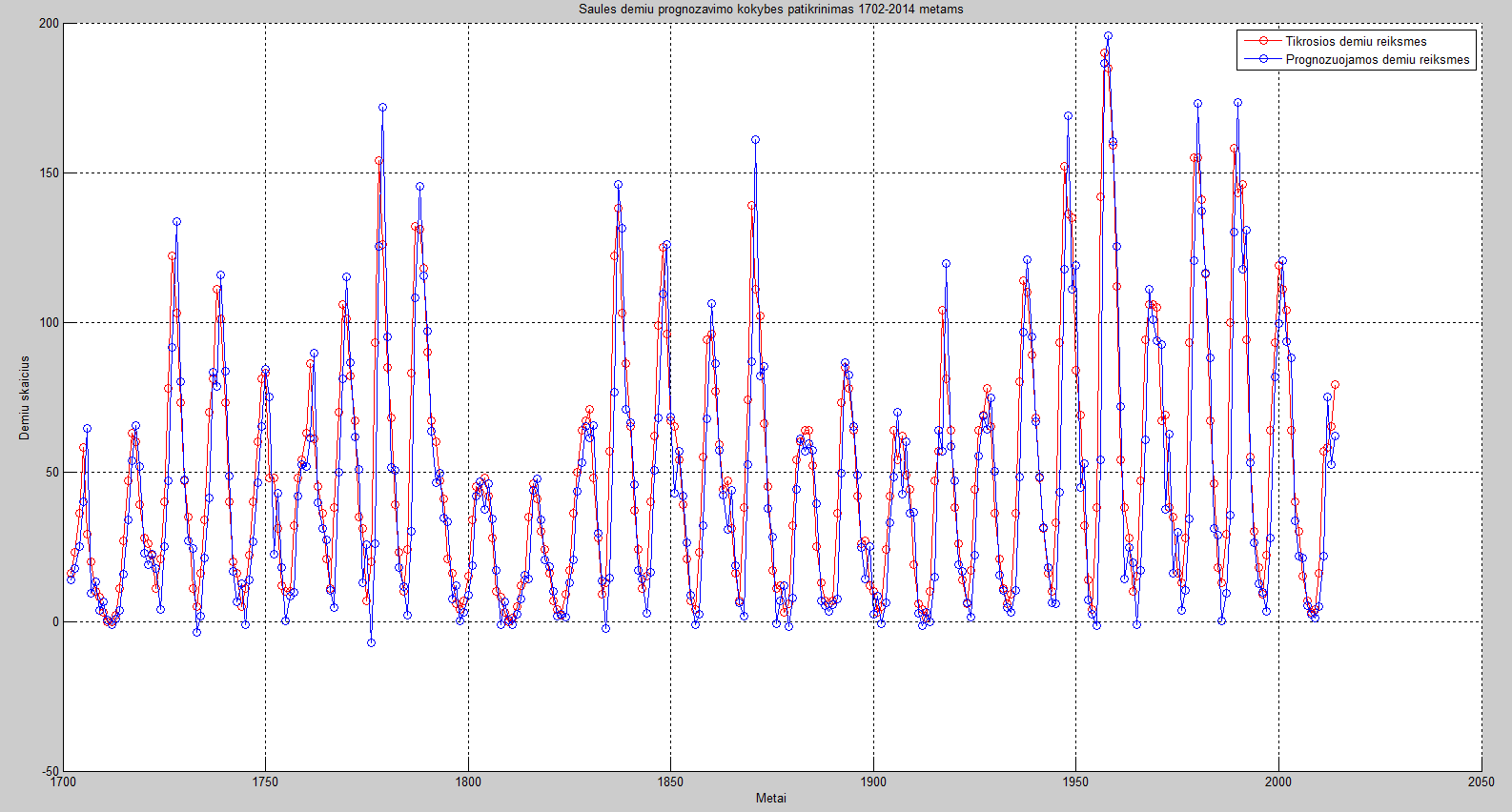
mse\_reiksme = mse(e)

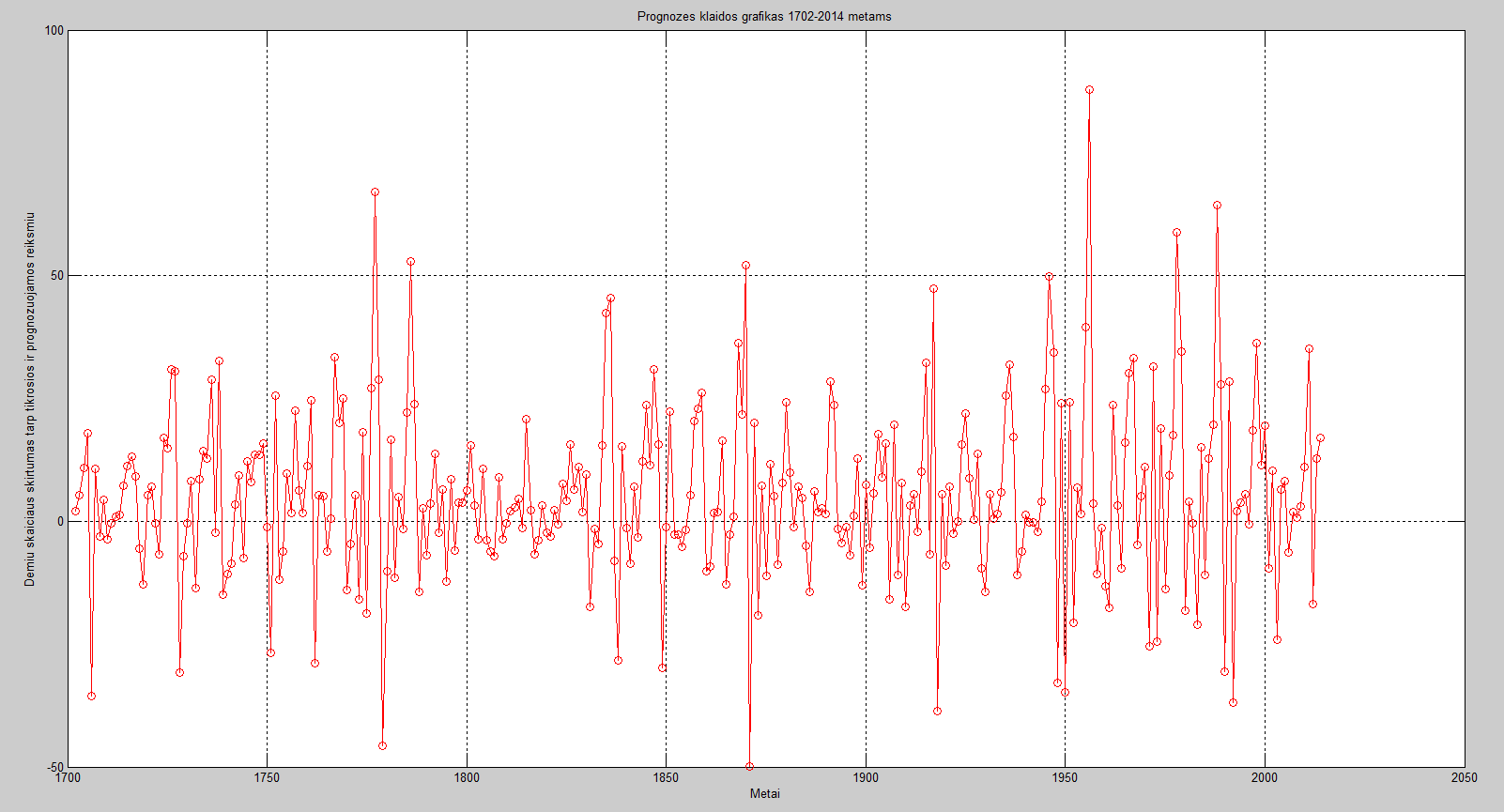
## PROGRAMOS VYKDYMAS (DIAGRAMOS, ATSAKYMAI Į KLAUSIMUS)

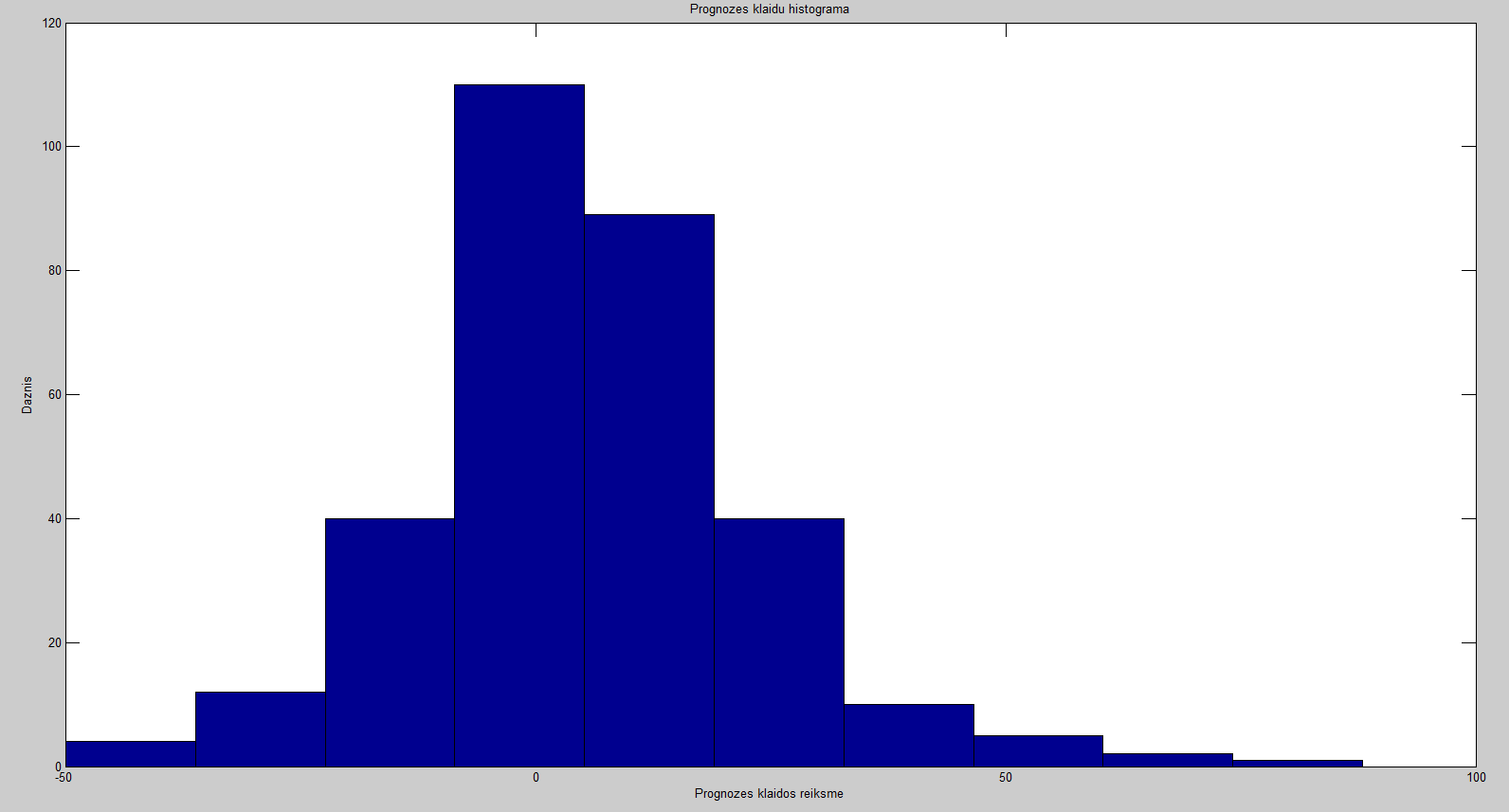










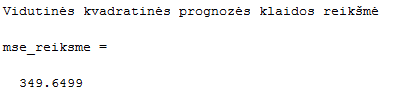


**Atsakymai:**

**Kokios yra naujos neurono svorių koeficientų reikšmės?**



**Kokia yra vidutinė klaidų kvadratų reikšmė?**



## PROGRAMOS KODAS (21 UŽDUOTIES SCENARIJAUS PUNKTAS)

clc, clear, close all;

load sunspot.txt;

% sunspot % patikrinti, ar užkrovė sunspot

% % clear sunspot

n = 6 % Ankstesnių reikšmių kiekis, kuriomis remiasi tolimesnės prognozės

% Saulės dėmių grafiko braižymas

figure(1);

plot(sunspot(:, 1), sunspot(:, 2), 'b-o');

grid on;

xlabel('Metai');

ylabel('Demiu skaicius');

title('Saules demiu aktyvumas 1700-1950 metais');

% % P ir T matricų aprašymas

L = length(sunspot);

P = []; % Įvestys

for i = 1:n

P = [P; sunspot(i:L-(n+1)+i, 2)'];

end

disp('P matrica (įvesties duomenys):'), P

disp('P matricos dydis:')

size(P)

disp('T matrica (išvesties duomenys):')

T = sunspot((n+1):L, 2)' % Išvestys

disp('T matricos dydis:')

size(T)

% figure(2)

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

% grid on;

% xlabel('Saules demiu skaicius (n-2)-aisiais metais');

% ylabel('Saules demiu skaicius (n-1)-taisiais metais');

% zlabel('Saules demiu skaicius n-aisiais metais');

% title('Saules demiu prognoziu, remiantis 2 ankstesniais metais, diagrama');

% Apmokymas

disp('Pu matrica (apmokymo duomenys):')

Pu = P(:, 1:200)

disp('Tu matrica (apmokymo rezultatai):')

Tu = T(:, 1:200)

disp('Pu matricos dydis:')

size(Pu)

disp('Tu matricos dydis:')

size(Tu)

net = newlind(Pu, Tu);

disp('Neurono svorio koeficientai:');

disp(net.IW{1});

disp('Neurono bias reikšmė:');

disp(net.b{1});

% % svorio koeficientų reikšmių priskyrimas

% w1 = net.IW{1}(1)

% w2 = net.IW{1}(2)

% b = net.b{1}

% modelio verifikacija (prognozuojamų reikšmių radimas)

Tsu = net(Pu) % Neurono veikimo simuliacija

figure(3), hold on;

plot(sunspot((n+1):200+n, 1), Tu, 'r-o');

plot(sunspot((n+1):200+n, 1), Tsu, 'b-o');

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title(sprintf('Saules demiu prognozavimo kokybes patikrinimas %d-%d metams', 1700+n, 1900+n-1));

% modelio verifikacija 1702-2014

Ts = net(P);

figure(4), hold on;

plot(sunspot(n+1:315, 1), T, 'r-o');

plot(sunspot(n+1:315, 1), Ts, 'b-o');

xlabel('Metai');

ylabel('Demiu skaicius');

grid on;

legend('Tikrosios demiu reiksmes', ...

'Prognozuojamos demiu reiksmes');

title(sprintf('Saules demiu prognozavimo kokybes patikrinimas %d-2014 metams', 1700+n));

% Prognozės klaidų vektoriaus sukūrimas

disp('Prognozės klaidų vektorius')

e = T - Ts

figure(5);

plot(sunspot(n+1:315), e, 'r-o');

grid on;

title(sprintf('Prognozes klaidos grafikas %d-2014 metams', 1700+n));

xlabel('Metai');

ylabel('Demiu skaiciaus skirtumas tarp tikrosios ir prognozuojamos reiksmiu');

% Prognozės klaidų histograma

figure(6);

hist(e);

title('Prognozes klaidu histograma');

xlabel('Prognozes klaidos reiksme');

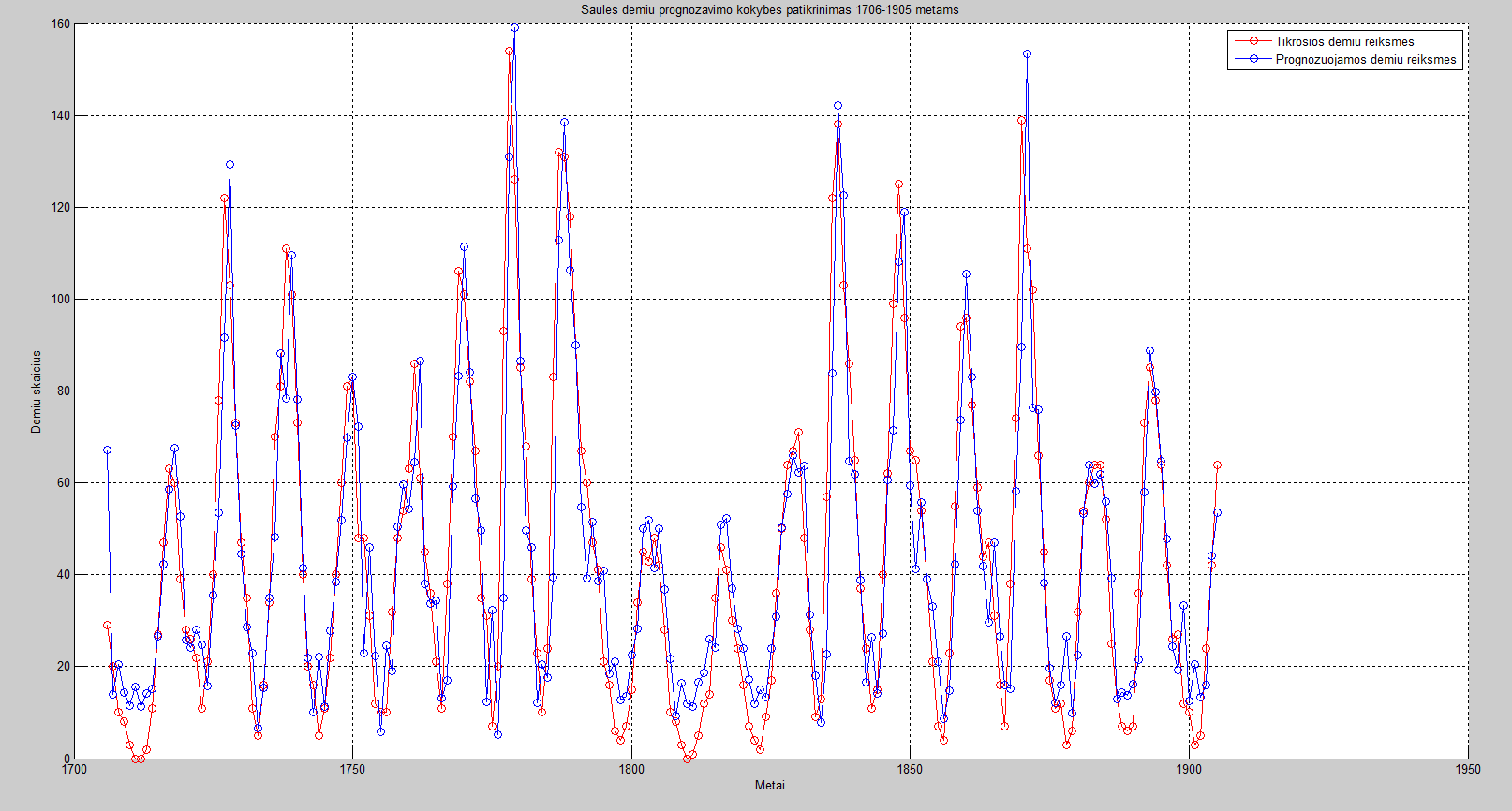
ylabel('Daznis');

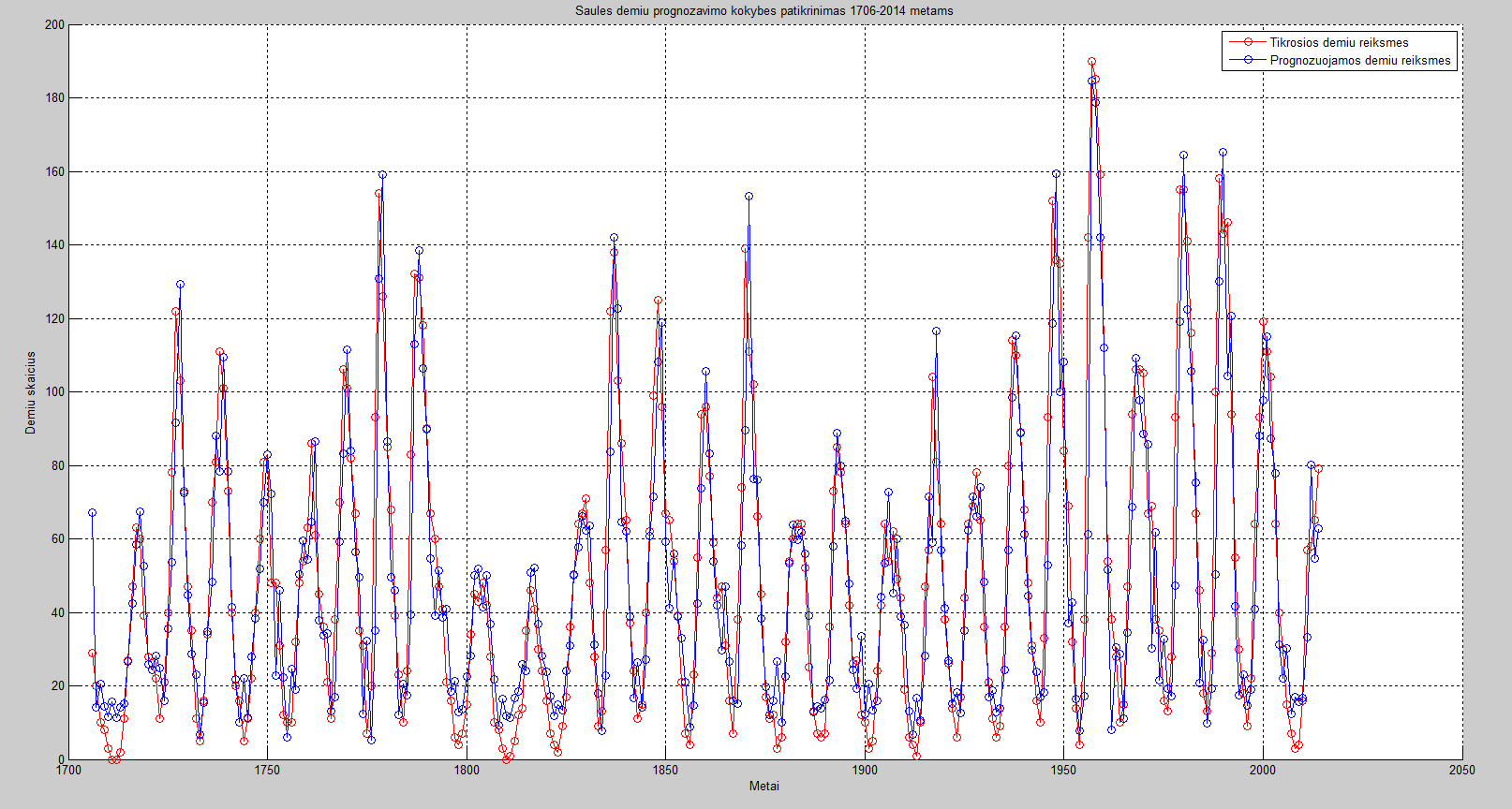
% Vidutinės kvadratinės prognozės klaidos reikšmės skaičiavimas

disp('Vidutinės kvadratinės prognozės klaidos reikšmė')

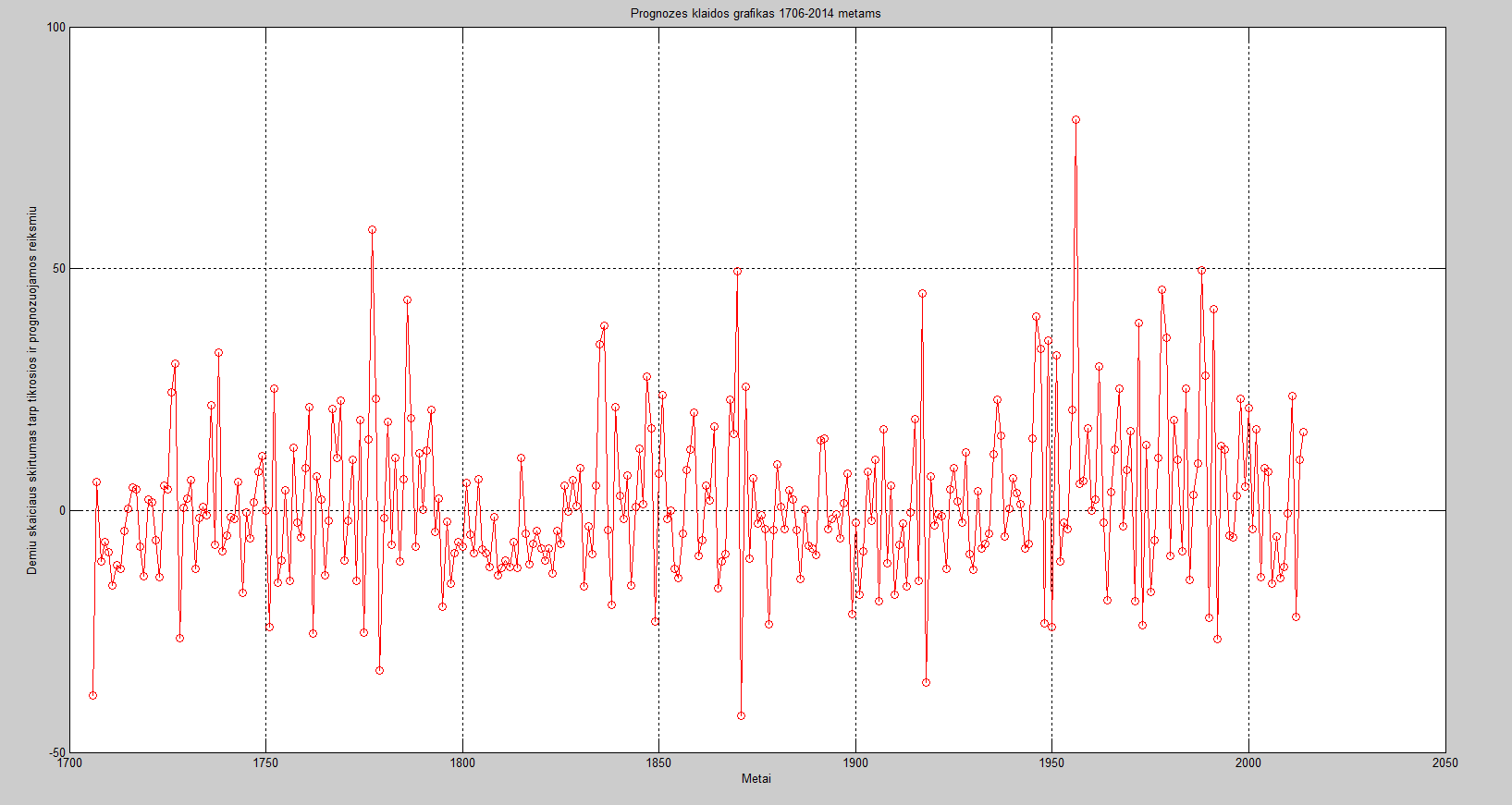
mse\_reiksme = mse(e)

## PROGRAMOS VYKDYMAS (DIAGRAMOS, ATSAKYMAI Į KLAUSIMUS)

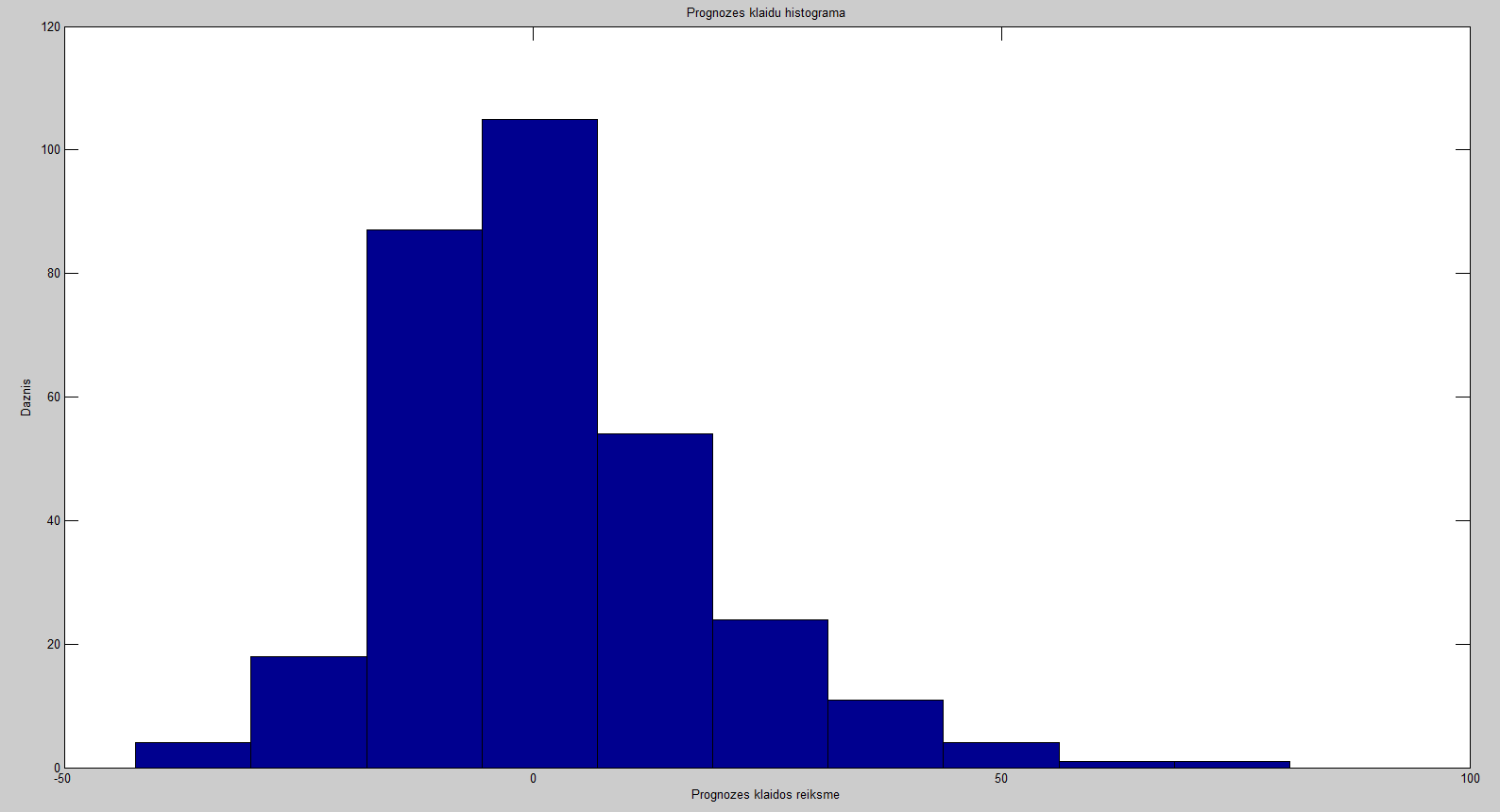




Iš pirmųjų dviejų grafikų galima įžvelgti, jog prognozuojamos reikšmės, kai remiamasi ankstesnių 6 duomenų rinkiniu, yra labai arti tikrųjų reikšmių, prognoze yra gana tiksli.



Šis grafikas rodo skirtumą tarp tikrosios ir prognozuojamų reikšmių tam tikru laikotarpiu, o kadangi labiausiai grafikas svyruoja (daugiausiai grafiko taškų yra išsidėstę) apie 0 reikšmę, tai galima spręsti, kad daugeliu atveju prognozės paklaidos nėra labai didelės.



Histograma papildo ankstesnį prognozės klaidų grafiką, parodydama, kad daugiausia klaidų reikšmių svyruoja apie 0 (daugiau nei 100 kartų). Taip pat iš histogramos matyti, kad prognozavimo tikslumas yra geresnis, kai remiamasi 6 ankstesnių duomenų kiekiu, nes kai buvo remiamasi 2 ankstesniais duomenimis (žiūrėti to atvejo diagramą), prognozės klaidų histogramos stulpelis ties 0 reikšme neviršijo 100, todėl galime daryti išvadą, kad prognozuojama tiksliau, kai remiamasi daugiau ankstesnių duomenų.

# 2 laboratorinio darbo papildoma užduotis “Krabų klasifikavimas”

## UŽDUOTIES REALIZACIJA (PROGRAMOS KODAS IR DIAGRAMOS)

[x,t] = crab\_dataset;

size(x)

size(t)

% Reikalinga, kad išvengtumėm atsitiktinių rezultatų vykdymo metu

setdemorandstream(491218382)

% Sukuriamas 2 sluoksnių tinklas su 10 neuronų paslėptam sluoksny

net = patternnet(10);

view(net)

% Vykdomas apmokymas (kol tinklas tobulėja, remiantis validacijos rinkiniu)

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

nntraintool

% Reikalingas peržiūrėti, kaip tinklas tobulėjo apmokymo metu

plotperform(tr)

% testavimas

testX = x(:,tr.testInd);

testT = t(:,tr.testInd);

testY = net(testX);

testIndices = vec2ind(testY)

% Braižo, kiek klasifikacijų atlikta sėkmingai

plotconfusion(testT,testY)

% Bendras teisingai ir neteisingai suklasifikuotų reikšmių procentas

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

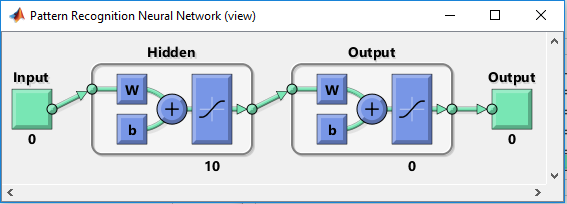
fprintf('Teisingai suklasifikuota: %f%%\n', 100\*(1-c));

fprintf('Neteisingai suklasifikuota: %f%%\n', 100\*c);

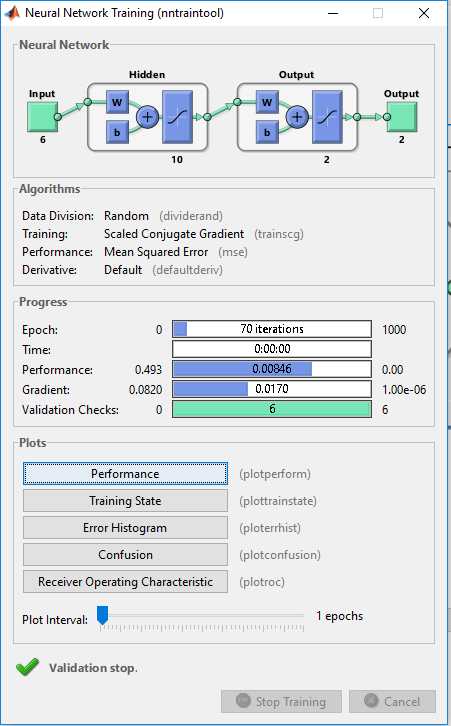
% Kitas teisingumo grafikas

plotroc(testT,testY)

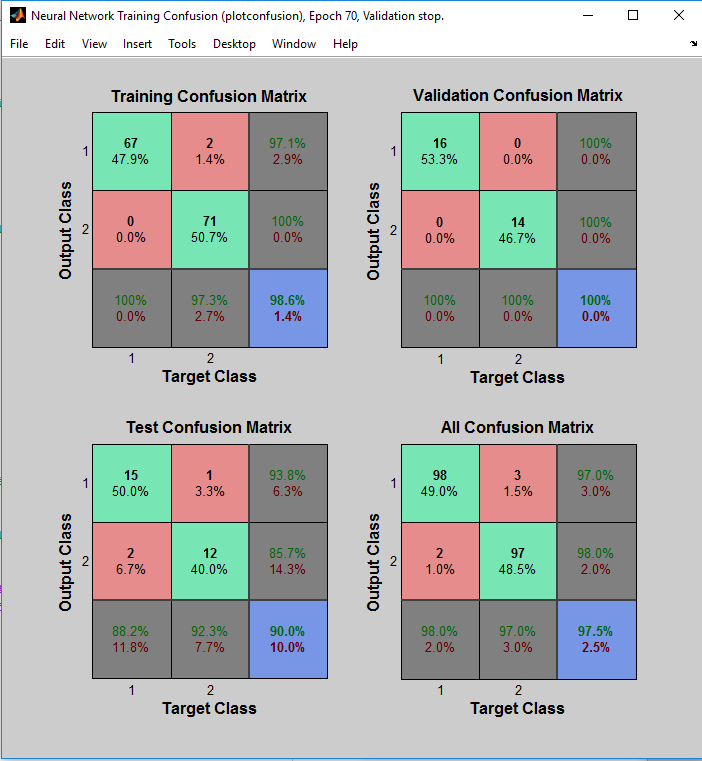
Uždavinio sprendimui naudojamo neuroninio tinklo schema:



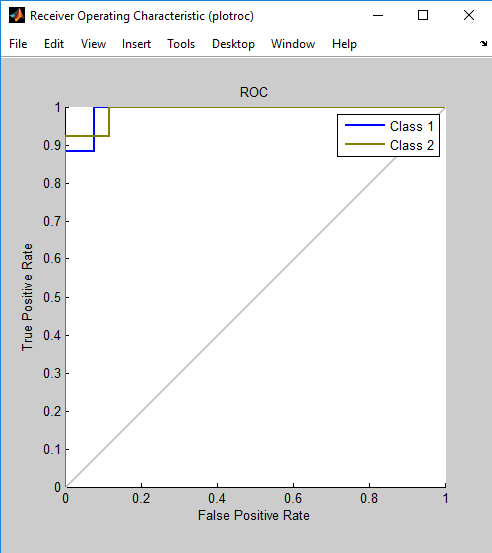
Neuroninio tinklo apmokymas:



Diagrama, vaizduojanti sėkmingus (žali kvadratėliai) ir nesėkmingus (raudoni kvadratėliai) suskirstymus į klases.

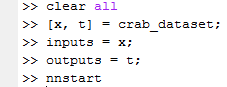


Diagrama, vaizduojanti, kaip siejasi teigiamos ir neigiamos reikšmės, kai rezultatai svyruoja tarp 0 ir 1.

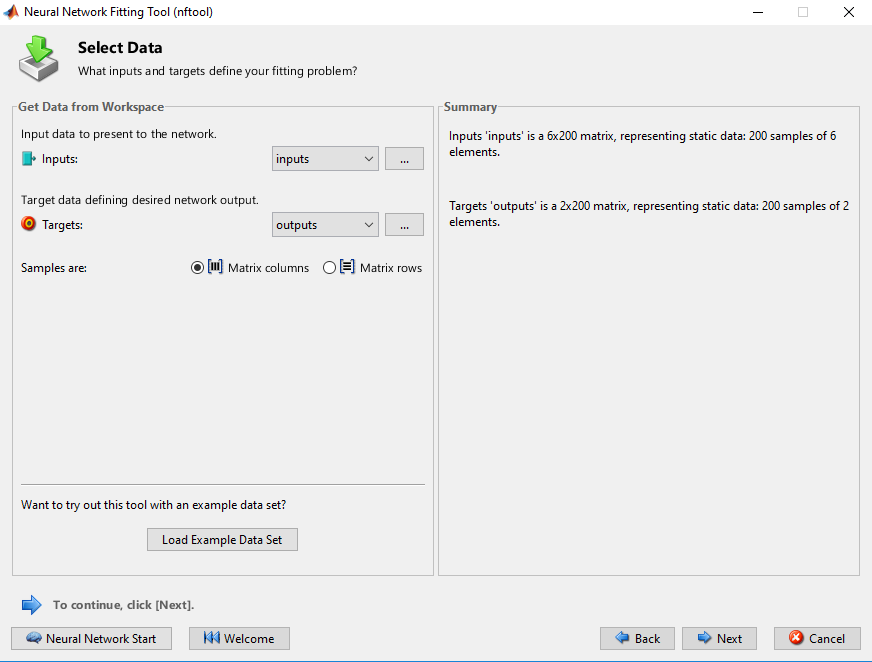


## UŽDUOTIES REALIZACIJA NAUDOJANT GRAFINĘ SĄSAJĄ

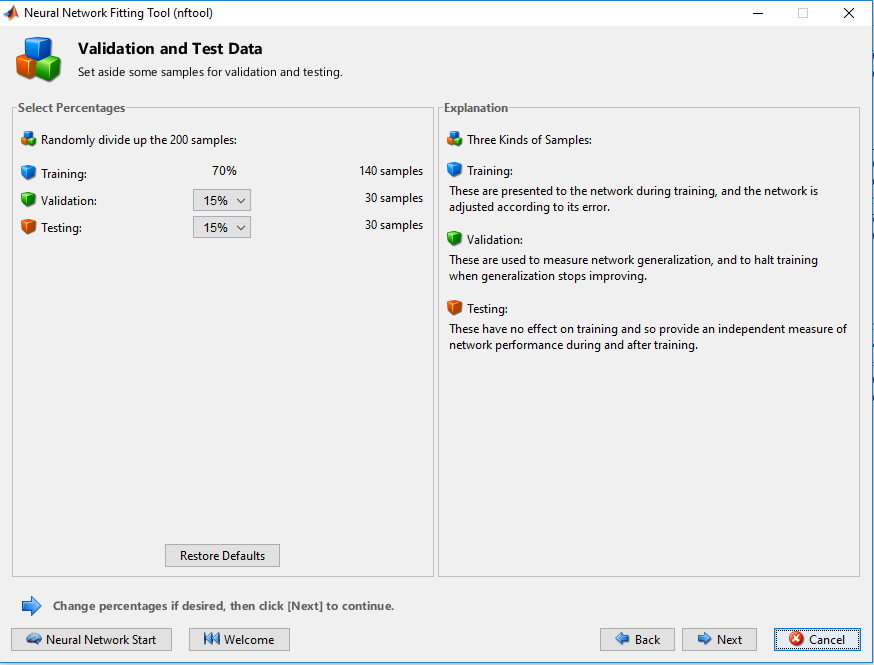
Pirmiausia išvalomi visi kintamieji bei iš duomenų rinkinio „crab\_dataset“ išskiriamos įvesties ir išvesties duomenų aibės:



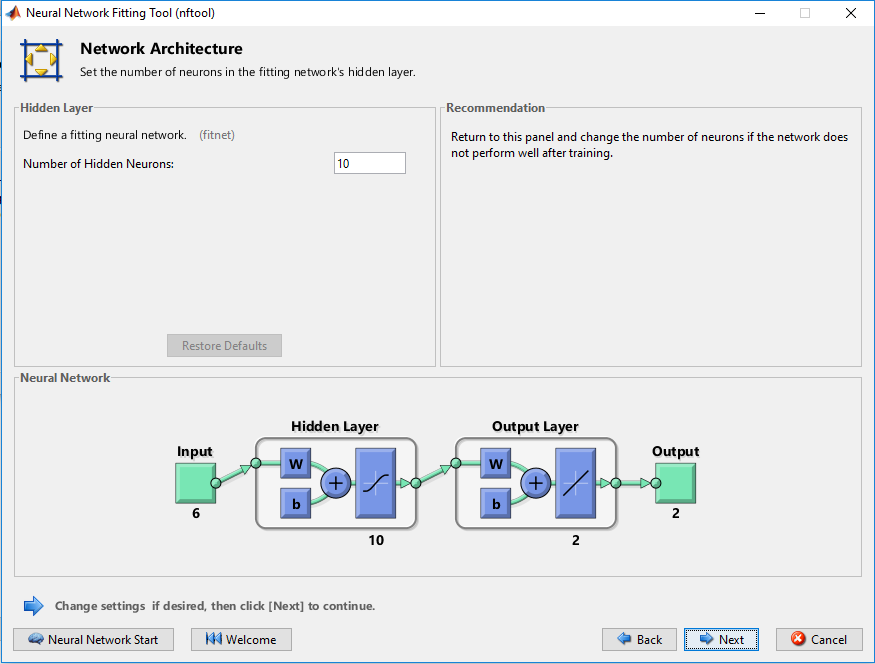
Įvedus komandą *nnstart* ir pasirinkus įrankį *Fitting tool,* atsiveria langas, kuriame reikia įvestims ir išvestims priskirti duomenų rinkinius. Įvesties ir išvesties reikšmėms atitinkamai priskyriau rinkinius, kuriuos buvau sudaręs kiek anksčiau:



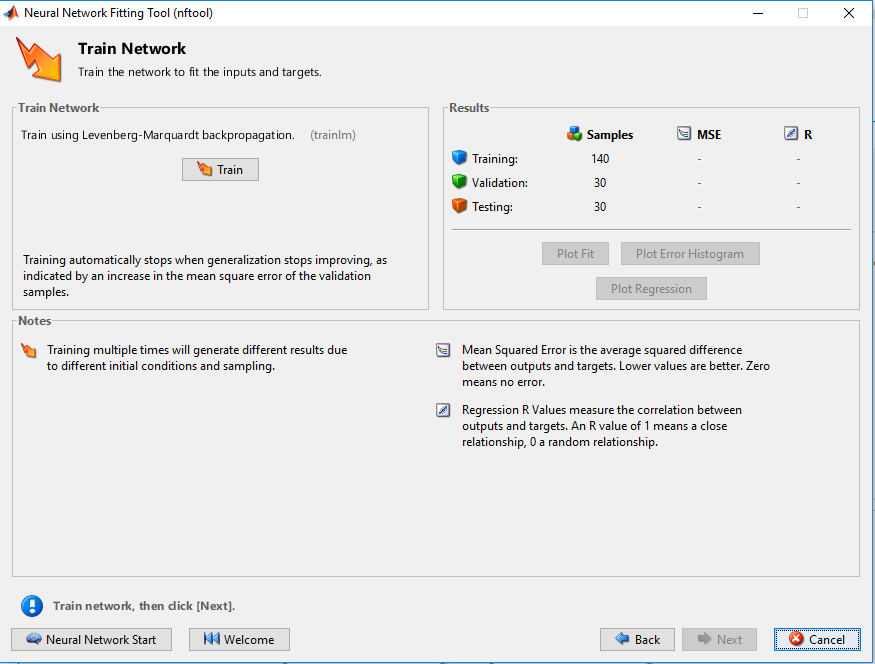
Paspaudžiama next, ir kitame lange pasirenkama duomenų rinkinio aibės procentinė dalis, kuri bus naudojama neurono apmokymui, validacijai bei testavimui (paliktos pagal nutylėjimą nustatytos reikšmės, atitinkamai 70%, 15% ir 15%):



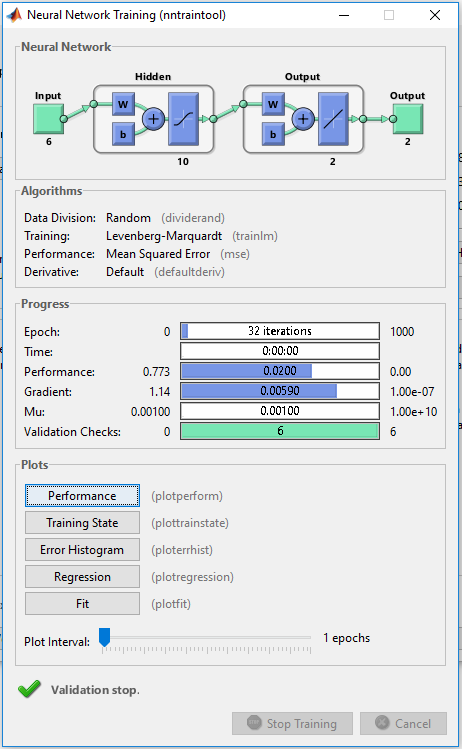
Spaudžiame next ir įvedame, kiek paslėptame sluoksnyje bus neuronų. Taip pat palikau pagal nutylėjimą 10 neuronų.



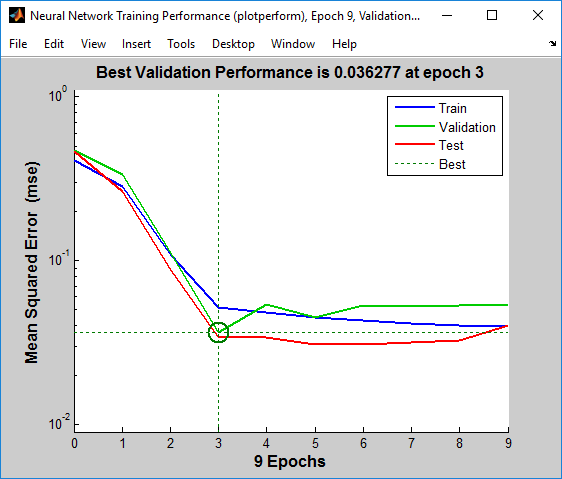
Tada pasirenkamas neuroninio tinklo apmokymas.



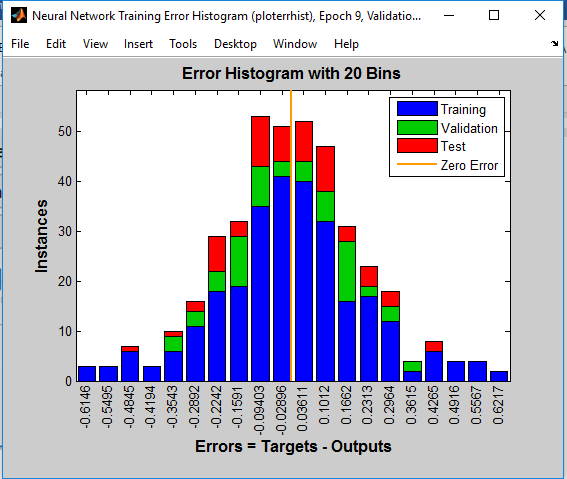
Pateikiamas neuroninio tinklo testavimo langas:



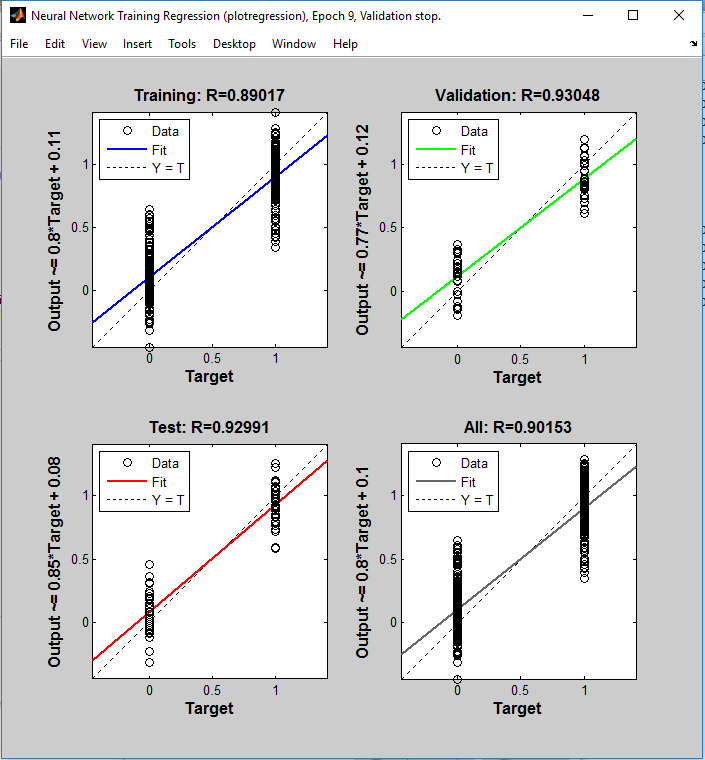
Pasirenkama performance diagrama:



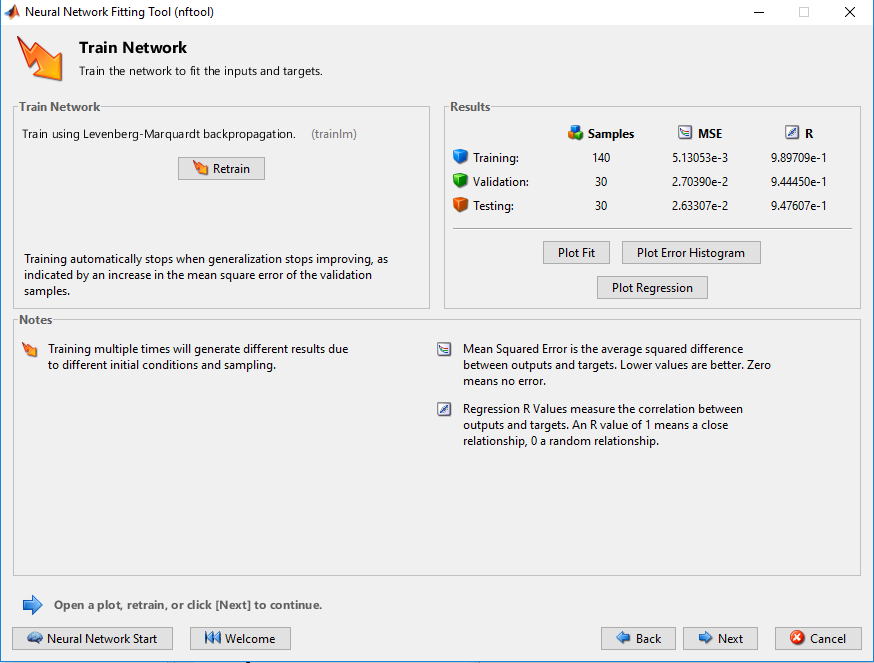
Taip pat pasirenkama klaidų historgrama:



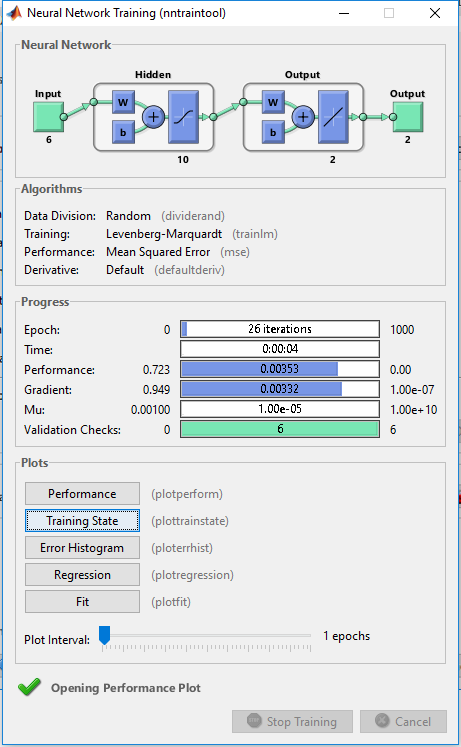
Regresijos diagrama:



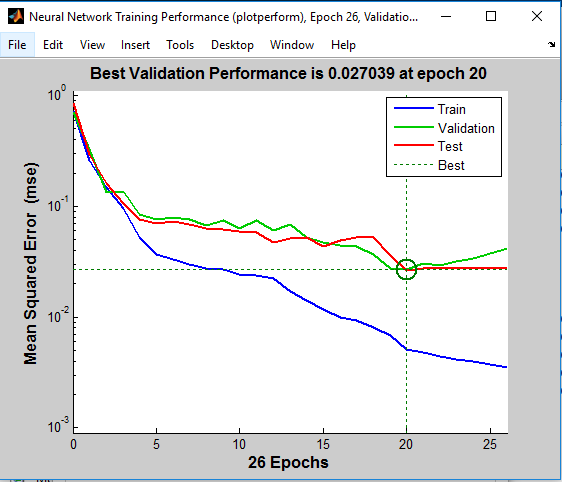
Toliau buvo pasirinkta apmokyti neuroninį tinklą dar kartą (spaudžiama *Retrain*):



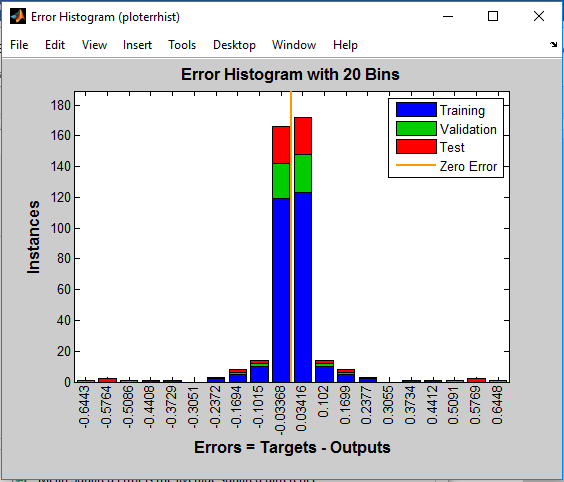
Gaunamas naujas testavimo langas su skirtingais testavimo rezultatais:



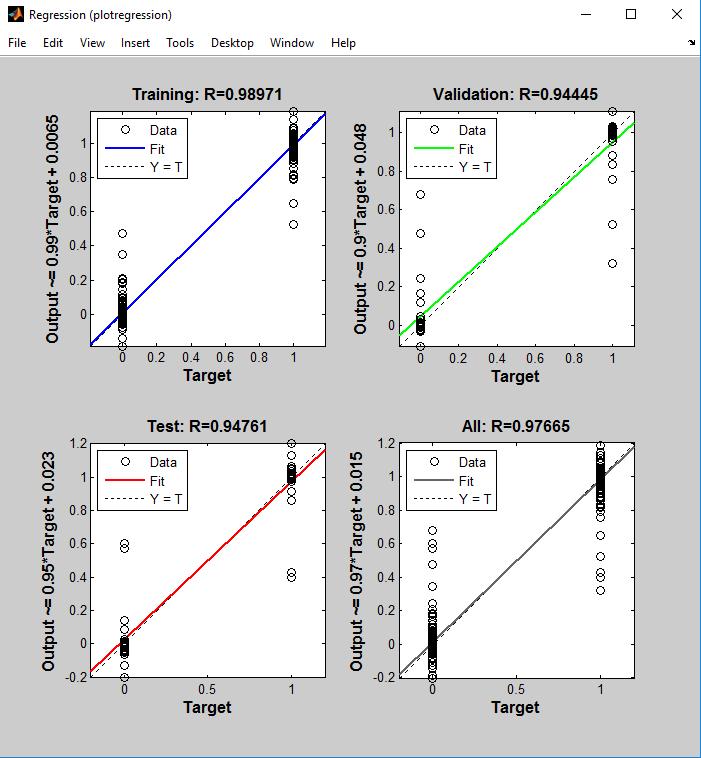
Performance diagrama:

l

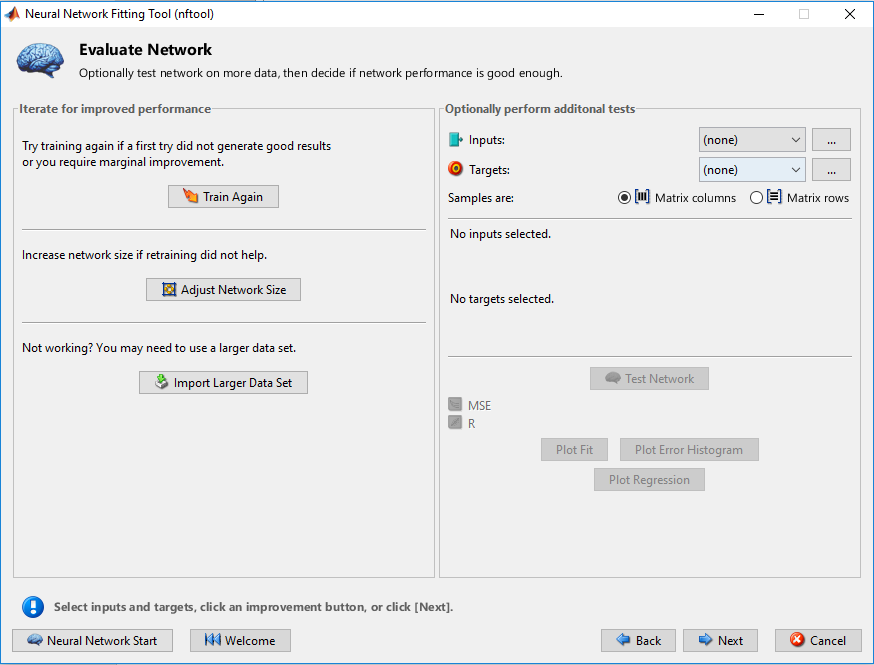
Klaidų histragrama:



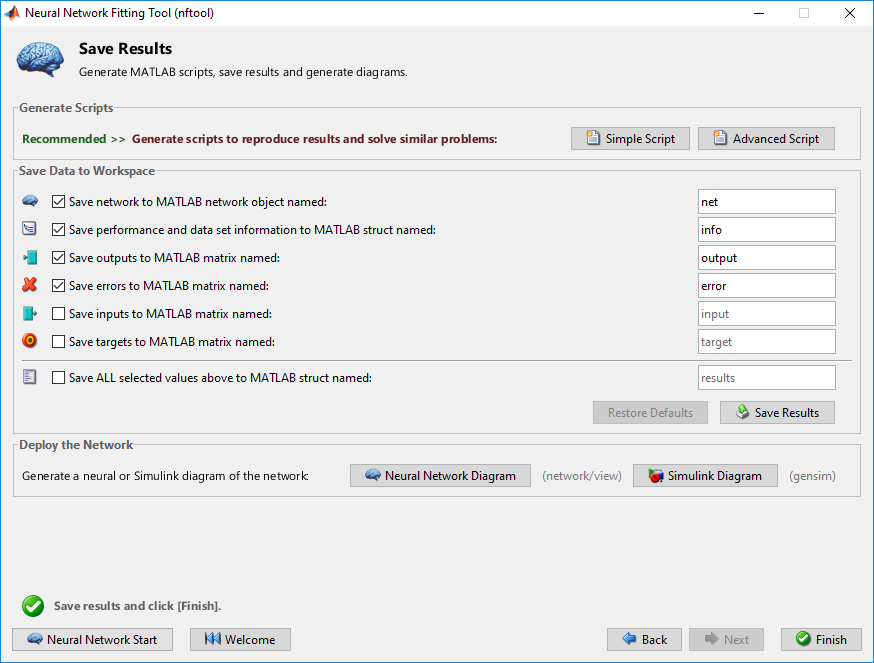
Regresijos diagrama:



Iš diagramų matome, kad rezultatai yra geresni, nei prieš pakartotinį apmokymą.



Po to, jei norima, galima išsaugoti sugeneruotą užduoties sprendimo kodą:



% Solve an Input-Output Fitting problem with a Neural Network

% Script generated by NFTOOL

% Created Wed Mar 15 09:32:16 EET 2017

%

% This script assumes these variables are defined:

%

% inputs - input data.

% outputs - target data.

inputs = inputs;

targets = outputs;

% Create a Fitting Network

hiddenLayerSize = 10;

net = fitnet(hiddenLayerSize);

% Setup Division of Data for Training, Validation, Testing

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

% Train the Network

[net,tr] = train(net,inputs,targets);

% Test the Network

outputs = net(inputs);

errors = gsubtract(targets,outputs);

performance = perform(net,targets,outputs)

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, plotfit(net,inputs,targets)

%figure, plotregression(targets,outputs)

%figure, ploterrhist(errors)

# Išvados

Laboratorinio darbo metu buvo susipažinta su dirbtinio neuroninio tinklo sąvoka, buvo susipažinta su prognozavimo uždaviniu, kurio sprendimas buvo realizuotas dirbtinių neuronų pagalba, panaudojant jų apmokymą remiantis ankstesnių bandymų duomenimis bei tikrosiomis reikšmėmis (mokymas su mokytoju). Taip pat papildomos užduoties metu buvo susipažinta su neuroninių tinklų panaudojimo uždaviniais, konkrečiu taikymu.