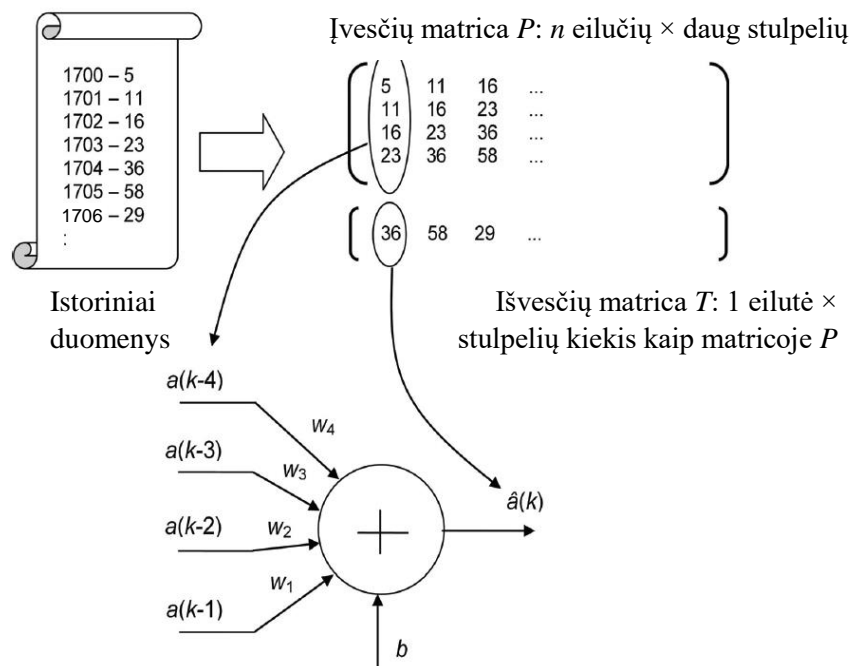


Laboratorinis darbas 2

Tikslai: susipažinti su prognozavimo uždavinio sprendimu panaudojant tiesinį dirbtinį neuroną, susipažinti neuroninio tinklo mokymosi, testavimo ir jų panaudojimo uždaviniais.

Darbo aprašas

Darbo metu bus panaudotas paprasčiausios struktūros dirbtinis neuroninis tinklas – vienetinis neuronas su tiesine aktyvavimo funkcija ($\text{purelin}(n)=\text{purelin}(Wp+b)=Wp+b$). Neuronu užduotimi bus laiko eilutės k -osios reikšmės $a(k)$ prognozavimas panaudojant n ankstesnes reikšmes $a(k-1)$, $a(k-2)$, ..., $a(k-n)$. Modelį, kurį realizuojame esant prielaidai, kad priklausomybė tarp prognozuojamos reikšmės ir prieš tai esančių eilės elementų gali būti aprašyta naudojant tiesinę funkciją, vadiname autoregresiniu tiesiniu modeliu n -tosios eilės.



1.1 pav. Tiesinio neurono schema kuri realizuoja tiesinį autoregresijos modelį eilės $n=4$.

Tiesinės autoregresijos modelio išraiška turi šią formą:

$$\hat{a}(k) = w_1 \cdot a(k-1) + w_2 \cdot a(k-2) + \dots + w_n \cdot a(k-n) + b \quad (1.1)$$

čia w_1, w_2, w_n ir b yra modelio parametrai ir $\hat{a}(k)$ pažymi sekančią eilės prognozuojamą reikšmę. Mūsų darbe autoregresinio modelio vaidmenį atliks dirbtinis neuronas, į kuriuo įėjimus padavinėsime prieš tai buvusias eilės reikšmes, modelio parametrų vaidmenį atliks šio neurono svoriniai koeficientai, o prognozuojama reikšmė pasirodys tik neurono išėjime (žr. 1.1 paveikslą). Išraišką 1.1 galime perrašyti:

$$a(k) = w_1 \cdot a(k-1) + w_2 \cdot a(k-2) + \dots + w_n \cdot a(k-n) + b + e(k) \quad (1.2)$$

čia $e(k) = a(k) - \hat{a}(k)$ yra prognozavimo klaida k -tajame laiko žingsnyje.

Turint istorinių duomenų rinkinį ieškosime optimalias autoregresinio modelio parametrų reikšmes. Tai reiškia, kad sieksime, kad $\hat{a}(k)$ prognozė, sugeneruota mūsų modelio pagalba, mažiausiai skirtųsi nuo

tikrosios $a(k)$ reikšmės, t.y. kad prognozės klaidų kvadratų suma $e(k)$ būtų mažiausia visam duomenų rinkiniui. Tuomet sukurtą modelį bus galima naudoti sekančioms reikšmėms prognozuoti.

Šiame darbe bus prognozuojamas saulės dėmių aktyvumas, išreikštas saulėje stebimų dėmių kiekiu tam tikrais kalendoriniais metais (žr. 1.2 paveikslą). Šis aktyvumas turi 11 metų cikliškumą.

Panaudojant istorinius duomenis, kurios aprašo saulės dėmių aktyvumą, pirmiausia aprašysime įvesties duomenis, kurie paduodami į neurono įėjimus, o taip pat išvesties duomenis („mokytojo atsakymus“ – žr. 1.1 paveikslą). Taip pat duomenų rinkinį padalinsime į mokymosi ir testavimo dalis.



1.2 pav. Paveikslas. Saulės dėmės.

Toliau atliksime modelio svorio koeficientų optimalių reikšmių parinkimo procedūrą. Tiesinio neurono atveju tai galima atlikti dviem būdais: tiesiogiai apskaičiuojant panaudojant lygčių sistemą arba panaudojant iteracinį metodą, t.y. neurono mokymasis panaudojant mokymąsi su mokytoju. Apskaičiavus svorio koeficientus, bus atlikta modelio verifikacija, kurios tikslas patikrinti prognozavimo kokybę.

Paruošiamieji veiksmai

1. Susipažinti su paketo *Matlab* bibliotekos *Neural Network Toolbox* demonstraciniais įrankiais, iliustruojančiais vieno neurono (tiesinio ir netiesinio) veikimą. Tam reikia aktyvuoti atitinkamą įrankį, įrašant jo pavadinimą Matlab komandinėje eilutėje:

nnd2n1 – neurono su vienu įėjimu veikimo iliustracija;
nnd2n2 – neurono su dviem įėjimais veikimo iliustracija;
demolin1 – optimalių svorio koeficientų reikšmių apskaičiavimas tiesiniam neuronui su dviem įėjimais (tiesioginis metodas – apskaičiuojant svorius per vieną žingsnį). Įrankio scenarijų galima redaguoti `edit demolin1`. Susipažinkite su juo turiniu.
demolin2 – optimalių svorio koeficientų reikšmių apskaičiavimas tiesiniam neuronui su dviem įėjimais (iteracinis metodas – taip vadinamas neurono apmokymas).

2. Susipažinti su bibliotekos *Neural Network Toolbox* funkcijomis, kurias naudosime darbe. Tam *Matalb* paketo komandinėje eilutėje įrašykite `help` ir funkcijos pavadinimą:

- | | |
|----------------------|---|
| <code>newlind</code> | – tiesinių neuronų sluoksnio sukūrimas (taip pat vieno neurono) taip pat svorio koeficientų reikšmių apskaičiavimas duotiems duomenų rinkiniams panaudojant tiesioginį metodą – sprendžiant lygčių sistemą; |
| <code>newlin</code> | – tiesinių neuronų sluoksnio sukūrimas (taip pat vieno neurono) taip pat svorio koeficientų reikšmių inicializavimas atsitiktinėmis reikšmėmis. |
| <code>train</code> | – vienkrypčio neuroninio tinklo (tai pat vienro tiesinio neurono) apmokymas, t.y. – svorio koeficientų apskaičiavimas iteraciniu metodu. |
| <code>sim</code> | – bet kokio neuroninio tinklo (taip pat tiesinio neurono) veikimo imitacija. Kad gauti informaciją apie šią funkciją – <code>help network/sim</code> . |

Darbo eiga

- Atsisiųsti sunspot.txt ir įrašyti jį į darbinę Matlab katalogą. Failę pateikti duomenys apie saulės dėmių aktyvumą nuo 1700 iki 1950 metų.
- Užkrauti failo turinį į Matlab darbinę atmintį: load sunspot.txt
- Patikrinti ar užkrauta atitinkama matrica – pirmas stulpelis atitinka metus, antras – saulės dienų aktyvumą. Pašalinti kintamąjį sunspot-clear sunspot .
- Pirma užduotis, kurią turi realizuoti mūsų programa – nubrėžti saulės dėmių aktyvumo už 1700-1950 metus grafiką. Grafikas turi būti pilnai aprašytas – pateikti ašių ir grafiko pavadinimus
Scenarijų redaktoriaus lange (*File/New/M-File*) patalpinti komandas

```
clear  
close all
```

Susipažinti su grafiku braižymu įrankiu (help plot). Pavyzdžiui, funkcijos plot iškvietimas gali būti toks:

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

Papildykite grafiką ašių bei diagramos pavadinimais(xlabel, ylabel, title).
- Priimkime, kad autoregresinio modelio eilė bus lygi 2 ($n=2$). T.y priimame, kad sekančių metų dėmių prognozė yra įmanoma turint tik dviejų ankstesnių metų dėmių skaičių. Tuomet neuronas turės tik du įėjimus. Papildykite scenarijų, aprašant matricas P ir T, kuriose atitinkamai pateikiami (mokymosi) įvesties duomenys o taip pat išvesties duomenys (žr. pav.1.1).

```
L = length(sunspot);           % duomenų kiekis  
P = [sunspot(1:L-2,2)' ;       % įvesties duomenų  
     sunspot(2:L-1,2)'];      % matrica  
T = sunspot(3:L,2)';          % išvesties duomenų  
                                % vektorius
```

Išanalizuoti matricų P ir T turinį ir matmenis (size).
- Susipažinti su funkcijų parametrais: plot3, grid, zlabel. Nubrėžti trimatę diagramą naujame grafiniame lange (figure 2) joje vaizduojant įvesties ir išvesties duomenis P ir T atitinkamai. Papildyti scenarijų. Pvz. braižymo funkcijai plot3 iškviesti galima naudoti

`plot3(P(1,:), P(2,:), T, 'bo')` . Įvykdyti scenarijų, išanalizuoti gautą grafiką. Pridėti ašių ir grafiko pavadinimus. Kokia yra neurono svorio koeficientų w_1 , w_2 , b optimalių reikšmių parinkimo grafinė interpretacija? *Pastaba*: diagramą galima sukurti panaudojant ikoną *Rotate 3D*.

7. Išskirkime iš įvesties P ir išvesties T duomenų rinkinių fragmentus, turinčius po 200 duomenų – taip vadinamą apmokymo duomenų rinkinį. Remiantis šiuo rinkiniu apskaičiuosime optimalias neurono svorio koeficientų reikšmes (autoregresinio modelio parametrus). Likę duomenys bus panaudoti modeliui verifikuoti. Tuomet, panaudojant jau esamas P ir T matricas, apibrėžkime dvi naujas – P_u ir T_u . Pavyzdžiui: P_u matricos apibrėžimas gali būti tokiau $P_u = P(:, 1:200)$; Įvykdyti scenarijų, įsitikinti, kad buvo sukurtos atitinkamos matricos, patikrinti jų turinį ir dydį.
8. Sukurti aptartos struktūros dirbtinį neuroną ir apskaičiuoti jo svorio koeficientų reikšmes tiesioginiu metodu (funkcija `newlind` – žr. `demolin1` pavyzdį). Šiuo tikslu panaudoti apmokymo duomenų matricas P_u ir T_u . Kintamąjį, atitinkantį sukurtą tinklą, suteikti vardą `net`. Atitinkamą komandą pridėti į scenarijų.

9. Pavaizduoti gautas neurono svorio koeficientų reikšmes

```
disp('neurono svorio koeficientai: ' )
disp( net.IW{1} )
disp( net.b{1} )
```

Atitinkamų svorio koeficientų reikšmes priskirti pagalbiniais kintamiesiems

```
w1 = net.IW{1}(1)
w2 = net.IW{1}(2)
b = net.b{1}
```

10. Sekančiame žingsnyje atliksime modelio verifikaciją – t.y. patikrinsime prognozavimo kokybę atliekant neurono veikimo imitaciją. Pradžioje tai atliksime su apmokymo duomenų rinkiniu, kuris buvo panaudotas svorio koeficientams apskaičiuoti.

Tarkime, turime suprognozuoti salės dėmių aktyvumą 1702–1901 metais. Tuo tikslu padavinėsime į neurono įėjimus aktyvumo rinkinius tokiais metais: 1700 ir 1701, 1701 ir 1702, ..., 1899 ir 1900. Tai galima atlikti automatizuotu būdu paduodant įvesties duomenis, esančius matricoje P_u . Pasėkoje gausime rezultatų vektorių T_{su} , t.y. prognozuojamas aktyvumo reikšmes 1702–1901 metams:

```
Tsu = sim(net, Pu)
```

Kadangi turime tikrąsias dėmių aktyvumo reikšmes nagrinėjamu laiko periodu (T_u), jas galime patikrinti su prognozuojamomis reikšmėmis (T_{su}). Tam galime naudoti kelis grafikus viename grafiniame lange (`hold on`). Gautame grafike vaizdavimai turi būti paženklinėti skirtingomis spalvomis bei turėti legendą (`legend`).

Atitinkinamomis komandomis papildyti scenarijų.

11. Užduotis analogiška pateiktai 10-me punkte. Pakartoti prognozių sudarymą visiems turimiems duomenims (1702–2014 metams). Nuraižyti prognozuojamų T_s (neurono išvesčių) ir tikrų T reikšmių lyginamąjį grafiką. *Pastaba*: sudarant prognozę modelio svorio koeficientai neturi būti

perskaičiuojami. Norima patikrinti kaip neuronas įvykdys prognozę panaudojant testavimo duomenis (naudojama funkcija `sim`).

12. Sukurti prognozės klaidos vektorių `e` (žr. išraišką 1.2). Naujame grafiniame lange nubraižyti prognozės klaidos grafiką (`plot`). Aprašyti jo ašis ir suteikti pavadinimą.

13. Nubraižyti prognozės klaidų histogramą (`hist`).

14. Remiantis (1.3) apskaičiuoti vidutinės kvadratinės prognozės klaidos reikšmę (ang. *Mean-Square-Error, MSE*):

$$MSE = \frac{1}{N} \sum_{k=1}^N (a(k) - \hat{a}(k))^2 = \frac{1}{N} \sum_{k=1}^N e(k)^2 \quad (1.3)$$

Tam tikslui galima naudoti funkciją `mse`.

Sekančiuose punktuose sukurtą scenarijaus tekstą norėsima taip modifikuoti, kad neurono svorio koeficientai būtų skaičiuojami iteraciniu metodu – atliekant neurono apmokymo procedūrą.

15. Scenarijų išsaugoti nauju vardu. Užkomentuoti programos kodą, kuriame nurodomas tiesioginis apmokymo būdas (8 punktas).

16. Panaudojant funkciją `newlin` (žr. panaudojimo iliustraciją: `help newlin`) sukurti tiesinį neuroną. Apibrėžti funkcijos parametrus (įvesties užlaikymas – 0 ir mokymosi greitis (`lr`) – tarp 0 ir 1).

17. Apibrėžti siekiamą mokymosi klaidos reikšmę (ang. *error goal*) ir maksimalų mokymosi žingsnių kiekį (ang. *epochs*), pvz.

```
net.trainParam.goal = 100;  
net.trainParam.epochs = 1000;
```

18. Panaudoti funkciją `train` tinklo apmokymui, pvz.:

```
net = train(net, Pu, Tu);
```

 Apskaičiuoti tinklo prognozavimo tikslumą MSE pagalba.

19. Išsaugoti ir įvykdyti scenarijų. Atsakyti raštu į klausimus:

- Kokios yra naujos neurono svorių koeficientų reikšmės?
- Kokia yra vidutinė klaidų kvadratų reikšmė?

20. Procedūrą pakartoti kitoms 17 punkto parametrų reikšmėms. Ištirti jų reikšmės įtaką į mokymosi proceso eigą ir prognozavimo kokybę. Kokia yra maksimali leistina mokymosi proceso greičio koeficiento `lr` reikšmė, kuri užtikrina proceso konvergenciją? Susipažinti su funkcijos `maxlinlr` paskirtimi.

21. Darbą atlikome priimant pradžioje pasiūlytą mūsų modelio struktūrą – sekančios reikšmės prognozavimas atliekamas remiantis dviejų ankstesniųjų metų duomenimis (t.y. modelio eilė $n=2$).

Scenarijų pakoreguoti tokiu būdu, kad prognozė remtųsi didesniu nei anksčiau duomenų kiekiu (pvz. $n=6$, $n=12$). Tuo tikslu reikės atitinkamai modifikuoti matricių P ir T apibrėžimus. Iširti (grafiškai ir pakomentuojant raštu) modelio struktūros keitimo įtaką į prognozavimo kokybę.

Darbo ataskaitoje pateikti: scenarijaus programos kodą su komentarais, sukurtus grafikus, klausimų atsakymus bei atliktų tyrimų rezultatus su pakomentavimu.

Papildoma užduotis

- i. Susipažinti su aprašu <https://se.mathworks.com/help/nnet/gs/fit-data-with-a-neural-network.html#9-43356>
- ii. Realizuoti užduotį iš pateikto sąrašo:
 1. Namų kainų prognozavimas (<https://www.mathworks.com/help/nnet/examples/house-price-estimation.html>)
 2. Krabų klasifikavimas (<https://www.mathworks.com/help/nnet/examples/crab-classification.html>)
 3. Vyno klasifikavimas (<https://www.mathworks.com/help/nnet/examples/wine-classification.html>)
 4. Vėžio aptikimas (<https://www.mathworks.com/help/nnet/examples/cancer-detection.html>)
 5. Simbolių atpažinimas (<https://www.mathworks.com/help/nnet/examples/character-recognition.html>)

Užduoties numeris apskaičiuojamas pagal formulę

<eilės numeris užsiėmimo registracijos sąraše> Mod 4+1

Papildomos užduoties ataskaitoje pateikti pagrindinių užduoties etapų vykdymo žingsnių iliustracijas.