

# Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė

## Pavel Stefanovič

Vilniaus pedagoginio universiteto magistrantas  
Vilnius Pedagogical University, master student  
Studentų g. 39, Vilnius  
Tel. (8 5) 275 17 96, faks. (8 5) 275 06 09  
El. paštas: paliukas@gmail.com

## Olga Kurasova

Vilniaus pedagoginio universiteto docentė,  
daktarė  
Vilnius Pedagogical University, Assoc. Professor,  
PhD  
Studentų g. 39, Vilnius  
Tel. (8 5) 275 17 96, faks. (8 5) 275 06 09  
El. paštas: kurasova@ktl.mii.lt

*Straipsnyje nagrinėjamos ir lyginamos tarpusavyje trys saviorganizuojančių neuroninių tinklų (SOM) sistemos: NeNet, SOM-Toolbox ir Databionic ESOM. Pagrindinis šių sistemų tikslas yra suskirstyti duomenis į klasterius pagal jų panašumą, pateikti juos SOM žemėlapyje. Sistemos viena nuo kitos skiriasi duomenų pateikimu, mokymo taisyklėmis, vizualizavimo galimybėmis, todėl čia aptariami sistemų panašumai ir skirtumai. SOM žemėlapiams mokyti ir vizualizuoti naudojami irisų ir stiklo duomenys.*

## Įvadas

Šiuo metu įvairiose srityse plačiai taikomi dirbtiniai neuroniniai tinklai, kurie yra naudojami klasifikavimo, klasterizavimo, prognozavimo, vaizdų, garsų atpažinimo ir kitiems įvairiems uždaviniams spręsti. Dažniausia skirtingiems uždaviniams naudojami skirtingo tipo neuroniniai tinklai. Kad galėtume naudoti neuroninius tinklus praktiniams uždaviniams spręsti, būtinos sistemos, turinčios patrauklias grafines sąsajas, įvairių parametrų nustatymo galimybes ir pan.

Šiame straipsnyje analizuojami vieno tipo neuroniniai tinklai – tai saviorganizuojantys neuroniniai tinklai, dar vadinami žemėlapiais (angl. *self-organizing maps*, SOM), ir juos realizuojančios sistemos. Tarpusavyje lyginamos trys sistemos: *NeNet*, *SOM-Toolbox* ir *Databionic ESOM*. Pagrindinis saviorganizuojančių neuroninių tinklų tikslas yra duomenų klasterizavimas ir vizualizavimas. Vertinant sistemas būtina atsižvelgti į tai, kaip duomenys juose yra klasterizuojami ir pateikiami SOM žemėlapiuose. Sistemos skiriasi viena nuo kitos tiek grafine

sąsaja, tiek įvairiomis tinklo mokymo funkcijomis, duomenų pateikimu, rezultatų vizualizavimo būdu ir kt. Šio tyrimo tikslas – išryškinti analizuojamų sistemų pranašumus ir trūkumus.

## 1. Saviorganizuojantys neuroniniai tinklai

Sakykime, kad turime analizuojamų duomenų aibę  $X = \{X_1, X_2, \dots, X_m\}$ , kurią sudaro vektoriai  $X_1, X_2, \dots, X_m$ , charakterizuojantys tam tikrą objektų aibę, kuri apibūdinama bendrais parametrais  $x_1, x_2, \dots, x_n$ ; čia  $m$  – analizuojamų objektų (vektorių) skaičius,  $n$  – parametrų skaičius. Parametrų reikšmės  $x_{i1}, x_{i2}, \dots, x_{im}$  ir yra vektoriaus  $X_i, i \in \{1, \dots, m\}$  komponentės. Taigi,  $X_1, X_2, \dots, X_m$  yra  $n$ -mačiai vektoriai, kurie interpretuojami kaip taškai  $n$ -matėje erdvėje  $R^n$ ; čia  $n$  – erdvės matmenų skaičius. Iš tiesų turime analizuojamų duomenų aibės matricą  $X = \{X_1, X_2, \dots, X_m\} = \{x_{ij}, i = 1, \dots, m, j = 1, \dots, n\}$ , jos eilutės yra vektoriai  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ ,  $i = 1, \dots, m$ ; čia  $x_{ij}$  yra  $i$ -ojo vektoriaus  $j$ -oji komponentė.

Saviorganizuojantis neuroninis tinklas yra neuronų, paprastai išdėstytų dvimačio tinklelio, dar vadinamo žemėlapiu arba lentele, mazguose, masyvas  $M$  (Kohonen, 2001). Stačiakampės topologijos atveju  $M = \{M_{kl}, k = 1, \dots, r, l = 1, \dots, s\}$ ; čia  $r$  – tinklo eilučių skaičius,  $s$  – tinklo stulpelių skaičius,  $M_{kl}$  – vektorius, kurio matmenų skaičius  $n$  yra toks pat kaip mokymo duomenų vektorių. Mokymo pradžioje neuronų (vektorių)  $M_{kl}$  komponentių pradinės reikšmės dažniausiai nustatomos atsitiktinai. Neuroniniam tinklui daug kartų pateikiama skirtingų objektų, nusakomų  $n$ -mačiais vektoriais  $X_1, X_2, \dots, X_m$ . Kiekviename mokymo žingsnyje (iteracijoje) vienas mokymo aibės vektorius  $X_p \in \{X_1, X_2, \dots, X_m\}$  pateikiamas į tinklą. Vektorius  $X_p$  palyginamas su visais neuronais  $M_{kl}$ : dažniausiai skaičiuojamas Euklido atstumas  $\|X_p - M_{kl}\|$  tarp šio vektoriaus  $X_p$  ir kiekvieno neurono  $M_{kl}$ . Randama, iki kurio neurono  $M_c \in \{M_{kl}, k = 1, \dots, r, l = 1, \dots, s\}$  atstumas yra mažiausias; rastas neuronas  $M_c$  vadinamas *neuronu (vektoriumi) nugalėtoju*. Visų tinklo neuronų komponentės keičiamos naudojantis iteracine formule

$$M_{kl}(t+1) = M_{kl}(t) + h_{kl}^c(t)(X_p - M_{kl}(t)). \quad (1)$$

Šioje formulėje  $t$  yra iteracijos numeris,  $h_{kl}^c$  – kaimynystės funkcija. Baigus tinklo mokymą, mokymo ar kita, vadinamoji testavimo duomenų (vektorių) aibė pateikiama į tinklą, randamas kiekvieno vektoriaus vektorius nugalėtojas, jį atitinkančiame žemėlapiu langelyje užrašomas vektoriaus eilės numeris arba klasės, kuriai priklauso šis vektorius, pavadinimas, t. y. žemėlapis yra sužymimas.

Kai tinklas išmokytas, būtina įvertinti jo kokybę. Tam dažniausia naudojamos dvi paklaidos: kvantavimo ir topografinė. Kvantavimo paklaida parodo, kaip tiksliai jau išmokyto tinklo neuronai prisiderina prie mokymo aibės vektorių. Kvantavimo paklaida  $E_{QE}$  – tai vidutinis atstumas tarp duomenų vektorių  $X_p$  ir jų vektorių nugalėtojų  $M_{c(p)}$ :

$$E_{QE} = \frac{1}{m} \sum_{p=1}^m \|X_p - M_{c(p)}\|. \quad (2)$$

Topografinė paklaida parodo, kaip gerai SOM tinklas išlaiko analizuojamų duomenų topografiją, t. y. tarpusavio išsidėstymą. Topografinė paklaida  $E_{TE}$  skaičiuojama pagal šią formulę:

$$E_{TE} = \frac{1}{m} \sum_{p=1}^m u(X_p). \quad (3)$$

Jeigu SOM žemėlapyje vektoriaus  $X_p$  neuronas nugalėtojas yra šalia neurono, iki kurio atstumas nuo  $X_p$  yra mažiausias, neskaičiuojant iki neurono nugalėtojo, tai (3) formulėje  $u(X_p) = 0$ , priešingu atveju  $u(X_p) = 1$ .

## 2. Saviorganizuojančių neuroninių tinklų programinė įranga

Šiuo metu yra sukurta nemažai saviorganizuojančius neuroninius tinklus realizuojančių sistemų: *SOM-PAK*, *SOM-Toolbox*, sukurtos vadovaujant SOM tinklų pradininko T. Kohoneno, *NeNet*, *Viscovery SOMine*, *Matlab* sistemos yra neuroninių tinklų priemonių komplektas ir kt. Kelių sistemų lyginamoji analizė pateikta darbe (Dzemyda, Kurasova, 2002).

### 2.1. Sistema *NeNet*

Sistema *NeNet* yra pritaikyta *Windows* operacinei sistemai. Internete platinama tik demonstracinė versija, todėl ji turi apribojimų (Hassinen ir kt., 1999). Galima sukurti žemėlapi, kurio ir eilučių, ir stulpelių skaičius neviršija 10, nagrinėti vektorius, kurių matmenų skaičius neviršija 6 ir vektorių skaičius neviršija 2000. Programa turi savo grafinę sąsają, kurioje galima lengvai mokyti ir vizualizuoti saviorganizuojančius neuroninius tinklus (SOM). Žemėlapis kuriamas trimis etapais: inicijavimo (*init*), mokymo (*train*) ir testavimo (*test*). Žemėlapiu inicijavimui skirtoje kortelėje galima nurodyti SOM eilučių ir stulpelių skaičių (*SOM Dimension*), pasirinkti šešiakampio ar stačiakampio formos žemėlapi (*Topology*), nurodyti, kuriuo būdu (burbuliuko ar Gauso metodu) bus skaičiuojama kaimynystės funkcija (*Neighbourhood Function*), pasirinkti inicijavimo

tipą – linijinį arba atsitiktinį (*Initialiazition Type*), atsitiktinių skaičių generatoriaus inicijavimo skaitmenį (*Random Seed*) ir įkelti mokymo duomenų failą (*Initialization & Training Data File*). Tinklo mokymui skirtoje kortelėje galima keisti įvairius mokymo parametrus: funkcijos tipą (*Function Type*), mokymo greitį (*Learning Rate*), kaimynystės spindulį (*Neighbourhood Radius*) ir mokymo trukmę (*Training Length*). Prieš testavimą nurodomas testuojamų duomenų failas (*Testing Data File*). Sistemai *NeNet* duomenys turi būti pateikti tekstiniam failui. Skaičius pirmoje eilutėje nurodo vektorių matmenų skaičių, po juo surašomi analizuojamus duomenis atitinkantys vektoriai. Paskutinis stulpelis skirtas vektorių pavadinimams nurodyti.

## 2.2. Sistema *SOM-Toolbox*

Sistema *SOM-Toolbox* yra sukurta *Matlab* programa, todėl ją galima naudoti tik šioje programoje (Vesanto ir kt., 2005). *SOM-Toolbox* turi daugiau kaip 200 įvairių funkcijų, jos suskirstytos į rinkinius pagal veikimo pobūdį. Pagrindiniai funkcijų rinkiniai, skirti SOM žemėlapiui sukurti, yra šie: struktūros ir failai (*Struct Conversion and File I/O*), duomenų apdorojimas (*Data Preprocessing*), inicijavimas ir apmokymo funkcijos (*Initialization and Training Function*) ir vizualizavimas (*Visualization*). Toliau pateikiamos funkcijos, kuriomis galima inicijuoti, išmokyti ir pavaizduoti paprasčiausią SOM žemėlapi:

**sd=som\_read\_data('filename')** – duomenų nuskaitymas iš failo, 'filename' – failo pavadinimas, sd – grąžinamas funkcijos parametras – duomenų struktūra.

**sd=som\_normalize(sd,'var')** – duomenų normavimas, sd – pradiniai duomenys, nurodžius 'var' duomenys normuojami taip, kad dispersija būtų lygi 1.

**sm=som\_make(sd)** – žemėlapio sukūrimas ir mokymas, sd – perduoti pradiniai duomenys, sm – grąžinamas funkcijos parametras – išmokytas žemėlapis.

**sm=som\_autolabel(sm,sd,'vote')** – žemėlapio elementų žymėjimas, sd – perduoti pradiniai duomenys, sm – žemėlapis, kurį norime sužymėti; nurodžius 'vote' bus rodoma tik dominuojanti klasė viename žemėlapi langelyje.

**map=som\_show(sm)** – žemėlapio vizualizavimas, sm – vizualizuojamas žemėlapis.

**som\_show\_add('label',sm)** – sužymėto žemėlapi vizualizavimo funkcija, sm – žemėlapis, kurį sužymime ir vizualizuojame.

Duomenys sistemai *SOM-Toolbox* pateikiami kaip ir sistemai *NeNet* tekstiniam failui. Vektorių matmenų skaičius gali būti nurodomas tekstinio failo pirmoje eilutėje arba vektorių nuskaitymo funkcijos argumente `som_read_data(filename, dimension)`. Duomenų failo paskutinis stulpelis skirtas vektorių pavadinimams.

## 2.3. Sistema *Databionic ESOM*

Sistema *Databionic ESOM* yra neseniai sukurta sistema, skirta didelėms duomenų aibėms klasterizuoti ir vizualizuoti (Ultsch, Moerchen, 2005). Kaip ir sistema *NeNet*, ji turi grafinę sąsają. *Databionic ESOM* yra nemokama sistema, todėl jokių apribojimų neturi. Norint sukurti SOM žemėlapi iš pradžių reikia paruošti du failus: mokymo failą ir vektorių klasių pavadinimų failą. Šiuos failus galima sukurti paprasčiausia užrašinės programa, mokymo duomenų failą išsaugoti prievardžiu *\*.lrm*, o klasių pavadinimų failą prievardžiu *\*.cls*. Pirmoje mokymo failo eilutėje rašomas vektorių skaičius, antroje – stulpelių skaičius plius vienas stulpelis indeksams, trečioje – skaičius 9 (jis yra unikalus, nekeičiamas) ir pagal tai, kiek duomenų stulpelių (parametru), rašome 1 arba 0 (skaičius 1 reiškia, jog stulpelis nuskaitytas, skaičius 0 – jog stulpelis bus ignoruojamas), ketvirtoje – stulpelių pavadinimai. Vektorių pavadinimų failo pirmoje eilutėje rašomas vektorių skaičius, žemiau pateikiami visi vektoriai. Pirmas stulpelis kaip ir mokymo failo skirtas eilučių indeksams. Turint abu failus, vizualizavimas vyksta dviem etapais: mokymas ir žemėlapi sužymėjimas. Pirma etape įkeliamas mokymo failas (*File/Load \*.lrm*) ir žemėlapis yra mokomas (*Training*). Prieš mokymą galima keisti įvairius parametrus: žemėlapi dydį, žemėlapi formą, kaimynystės funkciją, skaičiavimo greitį ir daugelį kitų. Antrame etape įkeliamas vektorių pavadinimų failas (*File/Load \*.cls*). Kad klasės būtų sužymėtos gautame žemėlapyje, reikia kortelėje *Classes* uždėti varnelę prie *Draw letters*.

### 3. Sistemų palyginimo rezultatai

#### 3.1. Tyrimo duomenys

Analizuoti du duomenų rinkiniai: irisų ir stiklo duomenys (Asuncion, Newman, 2007). Irisų duomenys – tai trijų veislių gėlės: *Iris Setosa*, *Iris Versicolor* ir *Iris Virginica*. Yra išmatuoti keturi gėlių parametrai:  $x_1$  – taurėlapių ilgiai (*sepal length*),  $x_2$  – taurėlapių pločiai (*sepal width*),  $x_3$  – vainiklapių ilgiai (*petal length*),  $x_4$  – vainiklapių pločiai (*petal width*). Sudaryti 4-mačiai vektoriai  $X_1, X_2, \dots, X_{150}$ ; čia  $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$ ,  $i = 1, \dots, 150$ .

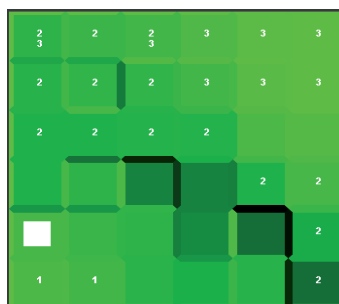
Vektoriai  $X_1, X_2, \dots, X_{50}$  atitinka I klasės irisus (*Iris Setosa*), vektoriai  $X_{51}, X_{52}, \dots, X_{100}$  – II klasės (*Iris Versicolor*) ir vektoriai  $X_{101}, X_{102}, \dots, X_{150}$  – III klasės (*Iris Virginica*).

Stiklo duomenys buvo surinkti mokslininkų, kurie norėjo padėti atpažinti kriminalistų rastas stiklo šukes. Šių duomenų rinkinį sudaro 9-mačiai vektoriai, tačiau dėl sistemos *NeNet* demonstracinės versijos ribotumo buvo analizuoti 6-mačiai vektoriai, atsisakyta trijų paskutinių parametrų. Matuoti šie parametrai:  $x_1$  – lūžimo indeksas (*Refractive Index*),  $x_2$  – natrio (*Sodium*),  $x_3$  – magnis (*Magnesium*), – aliuminis (*Aluminum*),  $x_5$  – silikonas (*Silicon*),  $x_6$  – kalis (*Potassium*). Taigi, turime 6-mačius vektorius  $X_1, X_2, \dots, X_{119}$ ; čia  $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$ ,  $i = 1, \dots, 119$ .

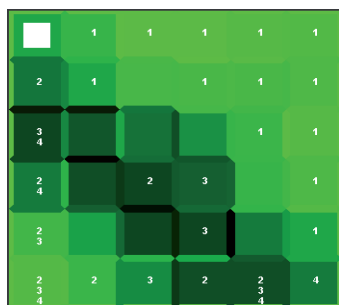
Pastatų langų (*Building windows*) stiklai sudaro I klasę, juos atitinka vektoriai  $X_1, X_2, \dots, X_{70}$ , taros stiklai (*Containers*) sudaro II klasę (vektoriai  $X_{71}, X_{72}, \dots, X_{83}$ ), stalo įrankiai (*Tableware*) sudaro III klasę (vektoriai  $X_{84}, X_{85}, \dots, X_{92}$ ) ir žibintai (*Headlamps*) sudaro IV klasę (vektoriai  $X_{93}, X_{94}, \dots, X_{119}$ ).

#### 3.2. Gautų žemėlapių analizė

Sistemoje *NeNet* vektoriai yra išdėstomi  $r \times s$  žemėlapyje (mūsų atveju  $r = s = 6$ ). Atspalviai atitinka U-matricos reikšmes (Ultsch, Siemon, 1989). Tamsesnė spalva tarp žemėlapio langelių reiškia, kad tuose langeliuose esantys vektoriai yra tolimesni nei tie, tarp kurių briaunos spalva yra šviesesnė. Irisų atveju aiškiai matyti, jog



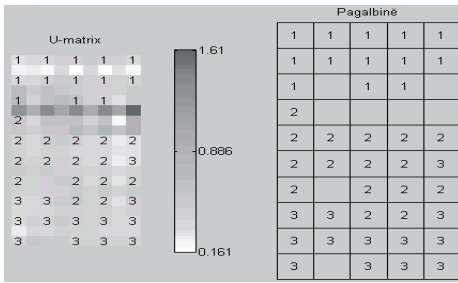
1 pav. Irisų duomenys 6×6 SOM žemėlapyje, gautame sistema *NeNet*



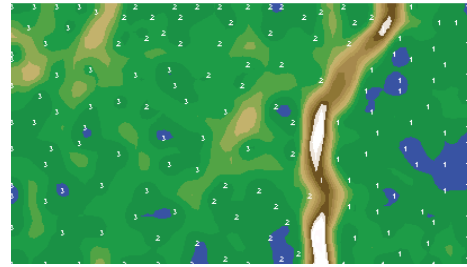
2 pav. Stiklų duomenys 6×6 SOM žemėlapyje, gautame sistema *NeNet*

viename kampe dominuoja I klasės vektoriai, o kitoje pusėje – II ir III klasės vektoriai (1 pav.). Žemėlapyje matome po vieną arba du skaičius viename langelyje, nors iš tiesų į tą patį langelį gali patekti daug tos pačios klasės vektorių, bet sistema vaizduoja tik po vieną tos klasės narį. Stiklo duomenys, ne taip kaip irisų, išsidėsto padidričiau, bet vėl matyti, jog I klasė (pastatų langai) atsiskiria nuo kitų trijų klasių (2 pav.).

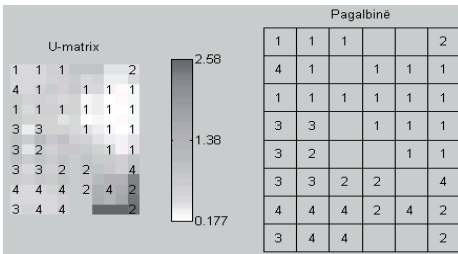
*SOM-Toolbox* sistemoje, priešingai nei yra *NeNet* sistemoje, žemėlapio elementus skiria ne siaura briauna, bet tarpiniai langeliai, todėl nors irisų atveju pasirinktas 8x6 žemėlapis, o stiklo – 10x5, tačiau matomi 15x11 ir 19x9 žemėlapiai (3 ir 4 pav.). Pagalbinėse lentelėse matomi tik tinklo elementai be tarpinių langelių. 3 paveiksle vėl matomos išsiskiriančios dvi irisų grupės – tai I klasė ir II kartu su III klase. 4 paveiksle taip pat matyti, jog stiklo II, III, IV klasių vektoriai yra tarpusavyje susimaišę, o I klasės vektoriai visi greta. Šioje sistemoje sunkiau pastebėti skirtingas stiklo duomenų klases negu *NeNet* sistemoje.



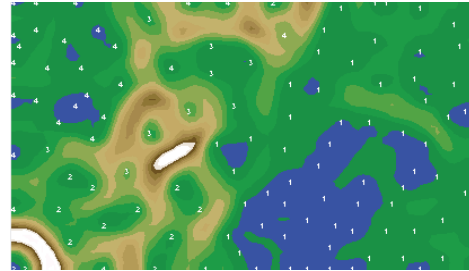
3 pav. Irisų duomenys 10×5 SOM žemėlapyje, gautame sistema SOM-Toolbox



5 pav. Irisų duomenys SOM žemėlapyje, gautame sistema Databionic ESOM



4 pav. Irisų duomenys 8×6 SOM žemėlapyje, gautame sistema SOM-Toolbox



6 pav. Stiklo duomenys SOM žemėlapyje, gautame sistema Databionic ESOM

Databionic ESOM sistema skirta dideliems SOM tinklams kurti, todėl žemėlapio dydį reikia nustatyti pakankamai didelį (50×82). Kaip ir pirmiau, matome dvi atskiras irisų sritis: I klasė ir II su III klase kartu (5 pav.). Šioje sistemoje „grioviai“ parodo skiriamąją klasių ribą. 6 paveiksle I klasės stiklo duomenys atsiskiria nuo kitų klasių.

Iš anksto žinant duomenų klases, apie duomenų klasterizavimo SOM žemėlapyje kokybę galima spręsti iš gautų vaizdų. Tačiau, kai kasės nežinomos, būtina vertinti skaitines paklaidas. Taigi, išmokius SOM žemėlapi, pagal (2) ir (3) formules apskaičiuojamos kvantavimo ir topografinės paklaidos. Jos pateikiamos 1 lentelėje. Matome, kad paklaidos ir irisų, ir stiklo duomenims yra mažesnės naudojant SOM-Toolbox sistemą, t. y. žemėlapio elementai (neuronai) labiau priderinti prie analizuojamų duomenų. Databionic ESOM sistemoje nėra galimybės matyti kvantavimo ir topografinių paklaidų. Aišku, jas galima suskaičiuoti žinant neuronų nugalėtojų reikšmes, tačiau sistema pateikia neuronų nugalėtojų koordinatas SOM žemėlapyje atskirai nuo pačių neuronų nugalėtojų reikšmių, todėl reikia nemažai papildomo darbo, kad nustatytume, į kurį SOM tinklo langelį pakliu-

1 lentelė. Kvantavimo ir topografinės paklaidos

Sistema	Irisų duomenys		Stiklo duomenys	
	kvantavimo	topografinė	kvantavimo	topografinė
NeNet	0,5538	0,3667	0,1990	0,1579
SOM-Toolbox	0,320	0,093	0,118	0,05

vo tam tikras analizuojamas vektorius. Šiame tyrime tai nėra atlikta.

Žinoma, kad SOM tinklo kokybė priklauso nuo neuronų pradinių reikšmių parinkimo, mokymo taisyklės, žemėlapio dydžio ir kitų parametų, todėl norint spręsti apie tinklo kokybę neužtenka vertinti paklaidas, gautas vienu eksperimentu, o būtina eksperimentų atlikti daugiau ir vertinti vidutines paklaidas. Kadangi kiekviena sistema gaunami šiek tiek kitokie SOM vaizdai, todėl tikslinga naudoti kelias sistemas, vertinti jomis gautus vaizdus ir interpretuoti rezultatus. Keli gauti vaizdai leidžia daryti išsamesnes išvadas apie analizuojamus duomenis. Susisteminti nagrinėtų sistemų pranašumai ir trūkumai pateikiami 2 lentelėje.

## 2 lentelė. Analizuojamų sistemų pranašumai ir trūkumai

Sistema	Pranašumai	Trūkumai
<i>NeNet</i>	Lengvai suprantama pradedančiajam vartotojui; turi patrauklią grafinę sąsają	Yra tik bandomoji versija, todėl turi apribojimų; jau senokai sistema nėra atnaujinama
<i>SOM-Toolbox</i>	Turi daugiau kaip 200 funkcijų; įvairūs vizualizavimo būdai	Veikia tik <i>Matlab</i> sistemoje; naujesnėse <i>Matlab</i> versijose kai kurios funkcijos neveikia; reikia turėti žinių apie <i>Matlab</i> sistemą
<i>Databionic ESOM</i>	Lengvai ir aiškiai atvaizduoja didelius duomenų rinkinius; galima keisti daug mokymo nustatymų	Sudėtingai atskirai parengiami mokymo ir vektorių pavadinimų failai; nėra galimybės matyti kvantavimo ir topografinių paklaidų

### Išvados

Norint gauti daugiau informacijos apie duomenų klasterius, susidariusius SOM žemėlapyje, būtina naudoti kelias sistemas, kuriose realizuotos skirtingos tinklo mokymo taisyklės ir įvairūs žemėlapio vizualizavimo būdai. Šiame tyrime naudotos trys sistemos: *NeNet*, *SOM-Toolbox*, *Databionic ESOM*. Visos sistemos duomenis klasterizavo panašiai, išskirdamos į tuos pačius

klasterius, tik vienoje jie labiau išryškėja, kitoje prasčiau. Patogiausia ir lengviausia sukurti SOM žemėlapi *NeNet* sistema, kuri tinka pradedantiems tyrinėti SOM, bet kur kas daugiau galimybių turi ir profesionalesnė yra *SOM-Toolbox* sistema. Neseniai sukurta *Databionic ESOM* sistema patogi norint vizualizuoti didelius duomenų rinkinius, kuriuos galima atskirti skirtingais būdais, pritaikyti pagal vartotojo poreikius. Patraukliausią grafinę sąsają turi *Databionic ESOM* sistema.

### LITERATŪRA

ASUNCION, A.; NEWMAN D. J. (2007). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>.

DZEMYDA, G.; KURASOVA, O. (2002). Comparative Analysis of the Graphical Result Presentation in the SOM Software. *Informatika*, vol. 13(3), p. 275–286.

HASSINEN, P.; ELOMAA, J.; RÖNKKÖ, J.; HALME, J.; HODJU, P. (1999). *Neural Networks Tool – Nenet* [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html>>.

KOHONEN, T. (2001). *Self-organizing Maps*. 3rd ed. Springer series in information sciences. Berlin: Springer-Verlag. 506 p. ISBN 3540679219.

ULTSCH, A.; SIEMON, H. (1989). *Exploratory Data Analysis: Using Kohonen Networks on Transputers*. Technical Report 329, Univ. of Dortmund, Dortmund, Germany.

ULTSCH, A.; MOERCHEN, F. (2005). *ESOM-Maps: Tools for Clustering, Visualization, and Classification with Emergent SOM*. Technical Report Dept. of Mathematics and Computer Science, University of Marburg, Germany, no. 46 [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://databionic-esom.sourceforge.net>>.

VESANTO, J.; HIMBERG, J.; ALHONIEMI E.; PARHANKANGAS, J. (2005). *SOM Toolbox for Matlab 5* [interaktyvus] [žiūrėta 2009 m. gegužės 20 d.]. Prieiga per internetą: <<http://www.cis.hut.fi/projects/somtoolbox/about.shtml>>.

### COMPARATIVE ANALYSIS OF SELF-ORGANIZING MAP SYSTEMS

Pavel Stefanovič, Olga Kurasova

#### Summary

In the article, we compare three systems of self-organizing maps: *NeNet*, *SOM-Toolbox* and *Databionic ESOM*. The main target of the usage of the systems is data clustering and their graphical presentation on the self-organizing map (SOM). The self-organizing maps are one of types of artificial neural networks. The SOM

systems are different one from other in their interfaces, the data pre-processing, learning rules, visualization manners, etc. Similarities and differences of the systems have been highlighted here. The experiments have been carried out with two data sets: iris and glass. Quantization and topographic errors of SOMs have been estimated, too.