KLASTERIZĀCIJAS METODES IZMANTOŠANA RBF NEIRONU TĪKLOS
APPLICATION OF CLUSTERING METHOD IN THE RBF NEURAL NETWORKS

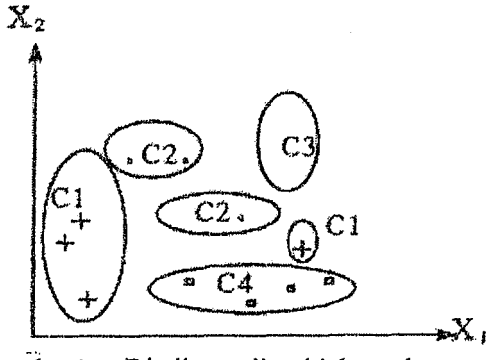
Pēteris Grabusts, Rēzeknes Augstskola,
Atbrīvošanas aleja 90, Rēzekne LV–4600, Latvija, tālr.: 4623798, e–pasts: peter@ru.lv

Abstract. *This paper describes one of classification algorithms, cluster analysis, that plays a significant role in the implementation of learning algorithm as applied to RBF–type artificial neural networks. The mathematical description of the K–means clustering algorithm is given and its implementation is demonstrated by experiment.*

Keywords: *RBF neural network, clustering, K–means*

1. Klasteranalīze un tās mērķi

Termins "klasteranalīze" radies 1939.gadā. Tas faktiski ietver sevī dažādu klasifikācijas algoritmu kompleksu. Dažādās pētniecības jomās aktuāls ir jautājums, kā organizēt novērojamos datus pārskatāmās struktūrās. Pastāv viedoklis [3], ka atšķirībā no daudzām citām statistiskām procedūrām vairumā gadījumu klasteranalīzes metodes tiek izmantotas tad, kad nav kaut kādu hipotēžu attiecībā par klasēm, bet vēl aizvien noris datu vākšanas stadija. Klasteranalīzes metodes ļauj sadalīt pētāmos objektus "līdzīgu" objektu grupās, ko sauc par *klasteriem*. Klasterizācijas būtība ir attēlota 1.zīmējumā.



1. zīm. Divdimensiju objektu telpas sadalījums klasteros

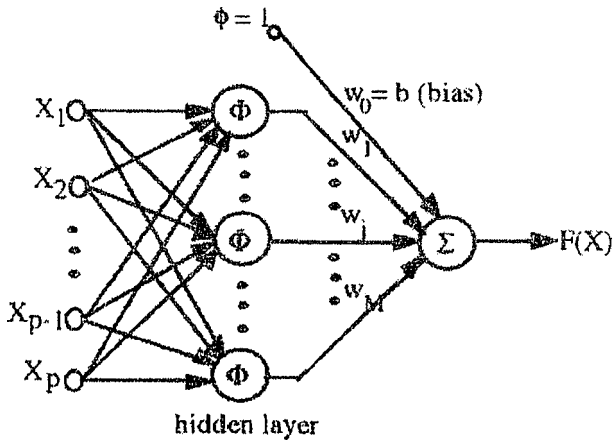
Klasteranalīzes process formāli sastāv no šādiem posmiem:
- analīzei nepieciešamo datu savākšana;
- klašu datu (klasteru) raksturojošo lielumu un robežu noteikšana;
- datu grupēšana klasteros;
- klašu hierarhijas noteikšana un rezultātu analīze.

2. Klasterizācija – RBF neironu tīkla apmācības pirmais etaps

Klasterizācijas algoritms tiek pielietots neironu tīklu ar radiālo aktivācijas funkciju apmācībā (saīsināti un tālāk – *RBF tīkls* no angļu valodas *Radial Basis Function Neural Network)* [1].

ISBN 9984 - 585 - 36 - 0

RBF neironu tīklus pārsvarā izmanto tēlu klasifikācijas un funkciju aproksimācijas uzdevumos. To izmantošana pēdējā laikā ieguvusi lielu popularitāti sakarā ar iespēju apstrādāt izplūdušos IF–THEN likumus. Tipiska RBF tīkla arhitektūra parādīta 2.zīmējumā.



2.zīm. Neironu tīkla ar radiālo aktivācijas funkciju arhitektūra

Vispārīgā gadījumā tīklam ir N ieejas x_i (i=1,2,...,N) un viena izeja. Ieejas signāls tiek padots slēptajā slānī, kuru veido neironi ar radiālajām funkcijām Φ^h_j , kur j=1,2,...,M ir slēptā slāņa neironu skaits. Svaru koeficienti w, kuri saista ieejas slāņa neironus ar slēptā slāņa neironiem, faktiski ir radiālo funkciju centra parametra w^h_j vērtības (i=i,...,N; j=1,...,M).
Radiālās funkcijas ir radiāli simetriskas funkcijas telpā $\Re^n \rightarrow \Re$, vispārīgi uzdotas šādi:

$$\Phi(x) = \Phi(\| x - c \|), \qquad x,c \in \Re^n \tag{1}$$

Visbiežāk izmantojamā radiālā funkcija RBF tīklos ir Gausa jeb potenciāla funkcija

$$\Phi(x) = e^{-\frac{\|x_i - c_i\|}{\sigma^2}}, \text{ kur} \tag{2}$$

x_i – n–dimensiju ieejas vektora x komponentes ;
c_i – radiālās funkcijas centrs (RBF tīklos bieži uzdod kā w_i);
σ – standarta novirze.

Slēptā slāņa neironu skaits tiek noteikts apmācības gaitā. Parasti katrs slēptā slāņa neirons atbilst konkrētai objektu klasei. Uzskatāmības labad var teikt, ka slēptā slāņa neironi izskaitļo Eiklīda distanci starp ieejas objektiem un radiālās funkcijas centru. Slēptā slāņa neironu izejas vērtības ir ieejas signāli izejas slāņa neironam. Tīkla izeja ir svērta izejas neirona ieejas signālu summa.

Neironu tīkla ar radiālo aktivācijas funkciju apmācība notiek divos etapos.

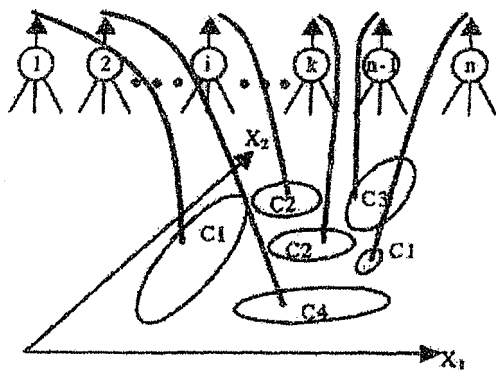
1. apmācības etaps – klasterizācija, pēc kuras tiek noteikti radiālās aktivācijas funkcijas forma un parametri, to skaitā klasteru centri;
2. apmācības etaps – apmācība ar skolotāju (LMS algoritma pielietošana).

Šis apmācības algoritms ir izvēlēts tāpēc, ka tā ir pamatmetode šāda tipa neironu tīklu apmācībā.

ISBN 9984 - 585 - 36 - 0

3. K–Means algoritms

Apmācība slēptajā slānī tiek veikta ar nekontrolējamu apmācības algoritmu palīdzību (*unsupervised learning*), kurus parasti sauc par *klasterizācijas* algoritmiem. Lai apmācītu RBF tīklu, tiek pielietots klasterizācijas algoritms ar nosaukumu "K–iekšgrupas vidējais" (*K–Means Clustering Algorithm*). Klasterizācijas mērķis ir ieejas vektorus sadalīt klasēs (klasteros) un noteikt to centrus. Centru vērtības turpmāk tiek izmantotas radiālo aktivācijas funkciju skaitļošanai RBF tīkla slēptajā slānī. Klasteru centru attēlošanas būtība uz slēptā slāņa neironiem parādīta 3. zīmējumā.



3. zīm. Klasteru attēlošana uz RBF neironiem

Algoritms "K iekšgrupas vidējais" minimizē *kvalitātes rādītāju,* kurš noteikts kā visu punktu, kas pieder klastera apgabalam, attālumu līdz klastera centra kvadrātu summa. Šī procedūra ieguva tādu nosaukumu, jo pamatojas uz klasteru grupas iekšienē vidējo attālumu aprēķināšanas līdz klastera centram.

Ievedam dažas definīcijas.

Ja dots vektors $X=\{x_1,x_2,....x_n\}$, *tad*

vidējā vērtība tiek rēķināta pēc formulas $$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i \qquad (3)$$

novirze $$\sigma_i^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2 \qquad (4)$$

Pirms klasterizācijas algoritma pielietošanas nepieciešams normalizēt ieejas objektus. Normalizēšanas rezultātā vidējā vērtība kļūst = ar 0, bet novirze = 1.

$$x_{i\_norm}=\frac{x_i-\bar{x}}{\sigma} \qquad (5)$$

$$\overline{x_{norm}}=\frac{1}{n}\sum_{i=1}^{n}x_{i\_norm} \qquad (6)$$

$$\sigma_{norm}^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_{i\_norm}-\overline{x_{norm}})^2 \qquad (7)$$

Eiklīda distanci starp diviem telpas X un Y punktiem aprēķina šādi:

$$d=\sqrt{(x_1-y_1)^2+.....(x_n-y_n)^2} \qquad (8)$$

ISBN 9984 - 585 - 36 - 0

Algoritms *"K–Means"* tiek izpildīts vairākos soļos.

1. solis. Inicializē klasteru centrus $\mathbf{w}_j$ (j – nepieciešamo klasteru skaits uzdevuma risināšanai).

2.solis. Grupē visus apmācības izlases punktus ap tuvākā klastera centru, t.i., katru punktu x_i saista ar klasteru j^*, kuram

$$\| x_i - w_{j^*} \| = \min_j \| x_i - w_j \| \qquad (9)$$

3.solis. Izskaitļo jaunos klasteru centrus, t.i., visiem w_j izskaitļo

$$w_j = \frac{1}{m_j} \sum_{x_i \in klasteram\ j} x_i \ , \qquad (10)$$

kur m_j – klasteram j piederošo punktu skaits.

4. solis. Atkārtot 2.soli tik ilgi, kamēr iterāciju laikā nemainās klasteru centru vērtības.

Algoritma darbības rezultātā tiek noteikti galīgie klasteru centri w_j , ievērojot nosacījumu, ka attālumu kvadrātu summai starp visiem punktiem, kas pieder grupai j, un klastera centram ir jābūt minimālai.

Par algoritma "K – iekšgrupas vidējais" priekšrocībām var uzskatīt popularitāti, lielu efektivitāti un procedūras vienkāršību. Bet gadījumā, kad objektu izvietojums ir neviendabīgs, algoritms var arī nesasniegt labus rezultātus. Tad tam ir jāmaina parametri (klasteru centru skaits) un atkal jāmēģina atkārtot algoritma darbību. Par trūkumu tiek uzskatīts tas, ka algoritms nav universāls.

Būtisks jautājums "K–Means" algoritma realizēšanā ir klasteru skaita un sākotnējo centru noteikšana [2]. Vienkāršākajos uzdevumos mēs pieņemam, ka *apriori* ir zināms klasteru skaits. Par sākotnējām m klasteru centru vērtībām tiek piedāvāts ņemt apmācošās kopas pirmos m objektus [1].

Pēc slēptā slāņa apmācības pabeigšanas ir jābūt iegūtiem aktivācijas funkciju parametriem. Tie ir klasteru centri $\mathbf{w}_j^h$ un klasteru standarta novirze σ_j^2 (j ir klasteru skaits). Lielums σ_j^2 ir jānoteic pēc formulas

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x \in \Theta_j} (x - w_j^h)^T (x - w_j^h), \text{ kur} \qquad (11)$$

kur Θ_j – objektu skaits apmācības izlasē, kas grupējas ap klastera centru w_j^h;

M_j – objektu skaits Θ_j ;

T – transponētās matricas vai vektora apzīmējums.

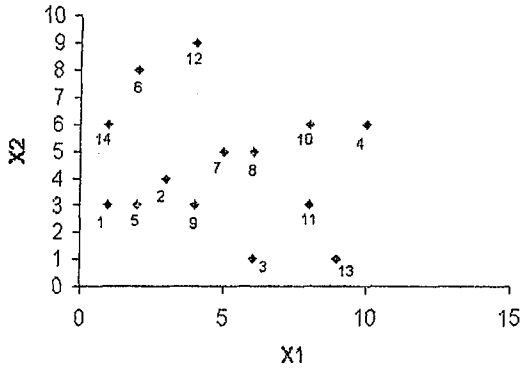
4. Klasterizācijas metodes pielietojuma piemērs

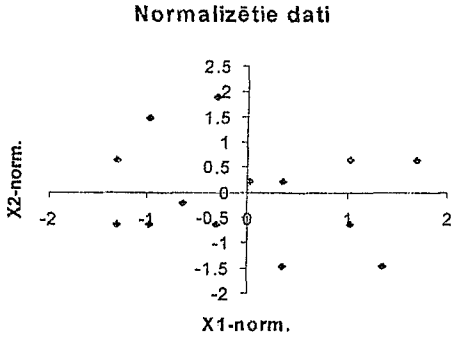
Lai nodemonstrētu klasterizācijas algoritma darbību, pieņemam, ka mums ir 14 ieejas vektori, kuri sadalīti divos klasteros. Ar "K–means" klasterizācijas algoritma palīdzību nepieciešams noteikt katram klasteram piederošos punktus un klasteru centrus.

ISBN 9984 - 585 - 36 - 0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X1	1	3	6	10	2	2	5	6	4	8	8	4	9	1
X2	3	4	1	6	3	8	5	5	3	6	3	9	1	6

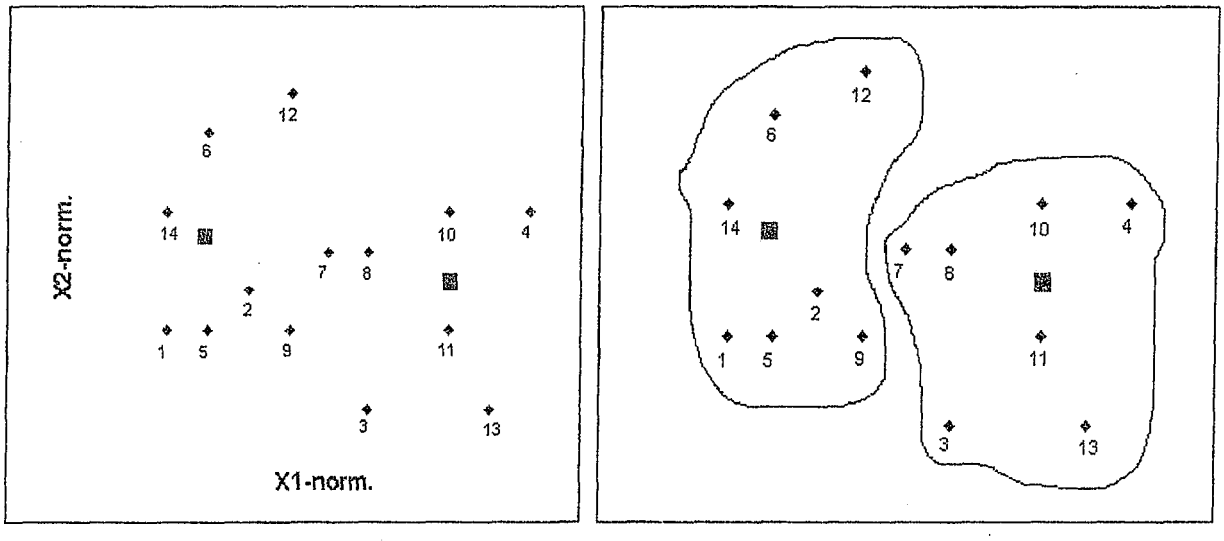
Katram ieejas vektoram (jeb punktam) ir divas komponentes: x_1 un x_2. Punktu sadalījums 2–D plaknē parādīts 4.a) zīmējumā. Pēc pamatdatu normalizācijas – izmantojot formulas (3) – (7) – iegūstam normalizēto punktu sadalījumu, parādītu 4.b) zīmējumā.





4.zīm. a) Ieejas punktu sadalījums b) Normalizētie dati

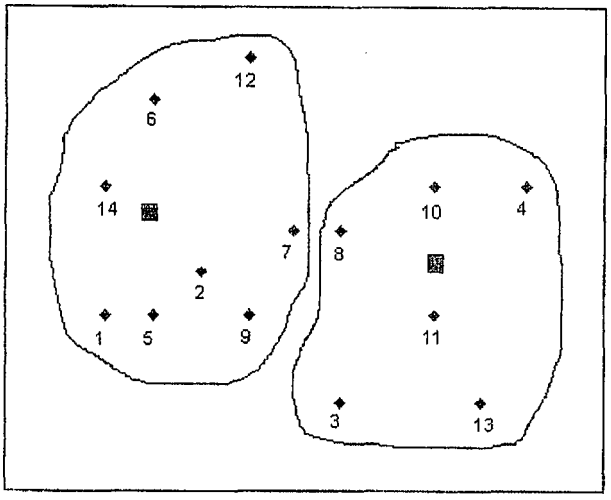
Lai varētu sākt pielietot klasterizācijas algoritmu, ir jānosaka klasteru skaits un to sākotnējie centri. Šajā uzdevumā pieņemam, ka ieejas punkti sadalīti divās klasēs, tātad izmantosim divus klasterus. Zemāk redzamajā 5.a) zīmējumā ir parādīts punktu sadalījums un uzskatāmības labad novāktas koordinātu asis. Orientējoši uzdodam sākotnējos klasteru centrus ar šādām koordinātēm: $\mathbf{w_1=(x_1,x_2)=(-1;0.5)}$ **un** $\mathbf{w_2=(1;0)}$. Zīmējumā tie parādīti kā kvadrātiņi. Sākam pielietot K–Means algoritmu. Zīmējumā 5.b) iezīmēti klasteriem piederošie punkti pēc **pirmās iterācijas.**



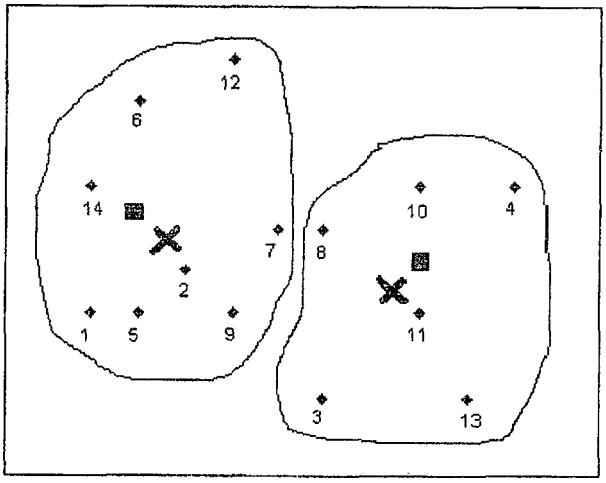
5. zīm. a) Punktu sadalījums ar patvaļīgi izvēlētu centru b) Klasteru sadalījums pēc 1. iterācijas

ISBN 9984 - 585 - 36 - 0

Atkal pēc formulas (3) izrēķinām katram klasteram vidējās vērtības, t.i., izrēķinām jaunos klasteru centrus: w_{1-new}=(−0.83359 ; 0.27036) un w_{2-new}=(0.833588 ; −0.27036). Tā kā tie atšķiras no mūsu patvaļīgi izvēlētā sākotnējā klasteru centra, tad turpinām pielietot klasterizācijas algoritmu. Rezultāti parādīti 6. zīm.



6.zīm. a) Klasteru sadalījums pēc 2.iterācijas



b) Klasteru sadalījums pēc 3.iterācijas

Kā redzams, pēc 2.iterācijas 7−ais punkts pamainījis savu klases piederību. Jaunie klasteru centri tagad būs šādi: w_{1-new}=(−0.72641 ; 0.26285) un w_{2-new}=(0.96855 ; −0.35047). Tā kā tie atšķiras no pirmajā iterācijā iegūto klasteru centriem, tad turpinām pielietot klasterizācijas algoritmu.

Trešajā iterācijā punkti savu piederību klasteriem nav mainījuši, t.i., otrajā iterācijā izskaitļotie klasteru centri paliek nemainīgi. Līdz ar to var secināt, ka klasterizācijas algoritma pielietojuma rezultātā ir noteikti klasteru centri un tiem atbilstošie punkti no apmācāmās kopas (punkti ir *klasterizēti*). Zīmējumā jaunie klasteru centri ieskicēti ar krustiņu.

5. Nobeigums

Pēc apmācības slēptajā slānī un radiālo funkciju centru noteikšanas notiek apmācība neironu tīkla izejas slānī, pielietojot kontrolējamas apmācības metodes vai tā saucamo *"apmācību ar skolotāju"*. Šim mērķim tiek izmantots *mazākās vidējās kvadrātiskās kļūdas algoritms (Least Means Square algorithm*). Pēc otrā apmācības etapa pabeigšanas RBF tīklu var uzskatīt par apmācītu un sagatavotu pētīšanas eksperimentiem.

Par klasterizācijas algoritmu priekšrocībām var uzskatīt popularitāti, lielu efektivitāti un procedūras vienkāršību. Jāatzīmē, ka klasterizācijas skaitliskie rezultāti var būt diskutējami [3] un parasti to rezultāti rūpīgi jāanalizē.

Literatūra

1. Hush D.R., Horne B.G. Progress in Supervised Neural Networks. What's new since Lippmann?, IEEE Signal Processing Magazine, January, 1993, vol.10,No 1.
2. Статистические методы для ЭВМ. – Москва: Наука, 1986.
3. Панкова Л.А., Трахтенгерц Э.А. Субьективность в интелектуальном анализе данных. – Москва: Препринт/Институт проблем управления, 1999.

ISBN 9984 - 585 - 36 - 0