

K. Kohonena tīkls

Kohonena tīkls (*Kohonen network*) pieder t.s. pašorganizējošo neironu tīklu modeļiem. Cits tā nosaukums ir Pašorganizējošā karte (*Self-Organizing Map, SOM*). Šo modeli 1982. gadā piedāvāja somu profesors Teivo Kohonens (*Teuvo Kohonen*).

K1. KOHONENA TĪKLA UZBŪVE UN DARBĪBAS PRINCIPI.

K1.1. PARAUGU KLĀSTERIZĀCIJA.

Paraugu klāsterizācija no klasifikācijas atšķiras ar to, ka iepriekš nav zināmas grupas, kurās paraugi būtu jāsklasificē. Paraugu sadalījums pa grupām tiek noteikts pašā klāsterizācijas procesā, zināms ir tikai grupu skaits, kurās klasificēt (neironu skaits Kohonena slānī). Taču nav teikts, ka visās grupās būtu jāieklasificē kaut cik līdzīgs skaits paraugu – sadalījums pa grupām var būt ļoti nevienmērīgs, vairākas grupas atstājot vispār bez paraugiem. Vienā grupā atrodošies paraugi ir savā ziņā līdzīgi viens otram, respektīvi tuvu viens otram pēc noteikta attāluma mēra. Par attāluma mēru starp paraugiem bieži lieto Eiklīda attālumu vai tā kvadrātu, kas ir paraugu atbilstošo elementu starpību kvadrātu summa.

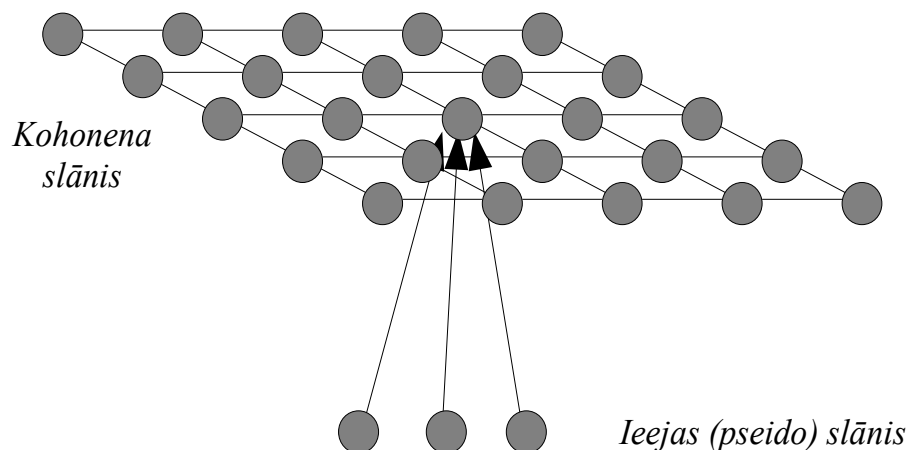
Klāsterizāciju parasti nevar tik tiešā veidā pielietot dažādu problēmu risināšanā kā klasifikāciju. Parasti ir nepieciešams papildus darbs – vairākkārtīga eksperimentēšana (dažādu sadalījumu klāsteros veidošana), datu analīze. Ja informācijas daudzums ir pietiekams, ērtāk ir izmantot klasifikāciju. Tomēr ne vienmēr pirms klasifikācijas veikšanas ir zināms, kādās tieši grupās būtu jāklasificē, un klāsterizācija varētu būt viens no veidiem, kā šādas grupas definēt.

Klāsterizāciju var veikt bankas darījumiem. Ja pēc klāsterizācijas kādā no klāsteriem par lielu daļu darījumu ir zināms, ka tie ir ar noziedzīgu raksturu, būtu īpaši jāpievērš uzmanība arī pārējiem darījumiem šajā klāsterī.

K1.2. KOHONENA TĪKLA GALVENĀS KONSTRUKTĪVĀS ĪPAŠĪBAS.

1. Līdzīgi kā vienslāņa perceptronom – ir viens ieejas slānis (pseido slānis) un viens īstais slānis (Kohonena slānis).
2. Visi ieejas slāņa neironi ir savienoti ar visiem Kohonena (izejas) slāņa neironiem. Ja neironu tīklā ir m ieejas un n izejas, tad ieejas slānī ir m neironi un attiecīgi katram izejas slāņa neironam ir m svāri, kā arī izejas slānī ir n neironi.

3. Kohonena slānī ir svarīgs neironu savstarpējais izvietojums. Starp jebkuriem diviem neironiem ir noteikts attālums (kaimiņa līmenis), kam ir nozīme apmācības procesā.
4. Gan ieejas un izejas vērtības, gan svari ir robežās $[0; +1]$.



Attēls K1. Kohonena tīkla grafs ar 3 ieejām un 25 neironiem Kohonena slānī. Saites parādītas tikai uz vienu no 25 Kohonena slāņa neironiem. Neironu tīklu izvietojums Kohonena slānī – 2-dimensiju, kvadrātisks.

K1.3. KOHONENA TĪKLA DARBĪBAS PRINCIPI.

1. Apmācība notiek iteratīvi. Katrā tīkla apmācības solī uz tīkla ieejām tiek uzstādītas ieejas vērtības, visos neironos tiek veikta darbināšana, un daļai neironu (bieži tikai vienam) tiek veikta svaru korekcija.
2. Apmācības procesu beidz, ja veikts pietiekoši liels apmācības soļu skaits vai arī tīkla nosacītā kļūda kļūst stabila.
3. Atšķirībā no perceptrona – apmācības paraugiem nav dotas vēlamās vērtības.
4. Paraugu klasificēšana notiek vienā solī un salīdzinoši ātri.

K2. APMĀCĪTA KOHONENA TĪKLA DARBINĀŠANA.

Tīkla inicializācija ar ievadvērtībām.

$$O_1 \leftarrow X$$

kur $X[m]$ – paraugs,

$O_1[m]$ – ieejas slāņa neironu izejas.

Parauga atpazīšana notiek vienā solī.

Katram izejas slāņa neironam notiek aprēķini – formulas K1, K2 un K3.

Summēšanas funkcija.

Summēšanas funkcija nosaka attālumu starp ievadvērtību vektoru un neirona svaru vērtību vektoru. Formulā K1 tiek izmantots t.s. Eiklīda attāluma kvadrāts, bet var izmantot arī citus attāluma mērus.

$$NET_j = \|W_j - O_j\|^2 = \sum_{i=1}^m (w_{ji} - o_i)^2 \quad (K1)$$

*kur NET_j – j-tā neirona summēšanas vērtība,
 w_{ji} – j-tā neirona i-tais svars,
 o_i – ieejas slāņa i-tā neirona izejas vērtība.*

Par summēšanas funkciju var izmantot arī Eiklīda attālumu tiešā veidā:

$$NET_j = \|W_j - O_j\| = \sqrt{\sum_{i=1}^m (w_{ji} - o_i)^2} \quad (K1a)$$

Aktivizācijas funkcija.

Neirona izejas vērtība ir atkarīga no citiem neironiem slānī – tikai neironu-uzvarētājs izejā iegūst nenulles vērtību.

Formulā K2 parādīta uzvarētāja (iegūstot tā kārtas numuru) noskaidrošana, bet formulā K3 – aktivizācijas funkcija, ņemot vērā noskaidroto uzvarētāju.

$$j_{win} = index(\min(\{NET_1, NET_2, \dots, NET_n\})) \quad (K2)$$

*kur NET_j – j-tā neirona summēšanas funkcija,
 j_{win} – uzvarētāja neirona kārtas numurs.*

$$o_j = \begin{cases} 1; & j = j_{win} \\ 0; & j \neq j_{win} \end{cases} \quad (K3)$$

kur o_j – j-tā neirona izeja.

Rezultāts.

Pēc tīkla darbināšanas (uz paraugu X) atbildi 1 dod tikai viens neirons. Tas arī norāda, kurai grupai ticis pieklasificēts dotais paraugs.

K3. KOHONENA TĪKLA APMĀCĪBA.

K3.1. APMĀCĪBAS PROCESA VISPĀRĪGS APRAKSTS.

Dots neironu tīkls $\{O_1[m], W_2[n][m], O_2[n]\}$, kurā ir m ieejas un n izejas. (Piemērs, kad $m=3$ un $n=25$, parādīts attēlā K1.)

Tāpat kā vienslāņa perceptronam, dots testa piemēru kopums P , kurā ir piemēri skaitā r . Katrs testa piemērs tiek identificēts kā P_s , kur $s=1..r$. Katra ieejas parauga P_s garums ir m .

Apmācības process ir tāds pats kā vienslāņa perceptronam un notiek šādi:

1. Visu svaru inicializācija ar gadījuma vērtībām intervālā $[0, +x]$, kur $0 < x < 1$.
2. Apmācības procesa veikšana pēc kārtas uz visiem apmācības piemēriem, atkārtojot to tik ilgi, kamēr tiek sasniegts konverģences nosacījums – ir veikts pietiekoši liels apmācības soļu skaits vai tīkla nosacītā kļūda kļūst stabila (pārāk daudz nemainās no epohas uz epohu) – formulas K8, K8a.

Katrā apmācības procesa solī tiek veiktas šādas darbības:

1. Tīkla inicializācija ar parauga P_s vērtībām.
2. Tīkla darbināšana, lai noskaidrotu neironu – uzvarētāju (formulas K1, K2).
3. Svaru korekcija neironam – uzvarētājam un daļai neironu uzvarētāja apkārtnē (K4, K5).

K3.2. APMĀCĪBAS PROCESA VIENA SOĻA APRAKSTS.

Svara korekcijas aprēķināšana neironam.

Svaru koriģēšana ir svaru izmaiņa ieejas parauga virzienā.

$$\Delta w_{ji} = \eta (o_i - w_{ji}) D(d(j, j_{win})) \quad (K4)$$

kur Δw_{ji} – svara izmaiņa,

η (eta) – apmācības koeficients $[0..1]$ (learning rate),

o_i – i -tā ieeja (parauga i -tais elements),

w_{ji} – j -tā neirona i -tais svars,

$d(\cdot, \cdot)$ – divu neironu nosacītais savstarpējais attālums, kam jābūt

definētam starp katriem 2 neironiem Kohonena slānī (piemēram, K6),

$D(\cdot)$ – kaimiņa funkcija (kaimiņa faktors) – piemēram, formula K7.

Svara korekcija.

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \quad (\text{K5})$$

kur $w_{ji}(t+1)$ – jaunā svara vērtība,

$w_{ji}(t)$ – vecā svara vērtība,

Δw_{ji} – svara izmaiņa,

t – soļa numurs (1..N).

Attālums starp neironiem.

Lai būtu iespējams noteikt attālumu starp neironiem, ir jābūt definētai slāņa topoloģijai. Piemērs ir redzams attēlā 2. Šajā piemērā neironi izkārtoti rindās pa 4, un attālums starp neironiem ($d(\cdot, \cdot)$) tiek definēts kā šķautņu skaits, kas nepieciešams, lai aizietu no viena neirona līdz otram.

$$d(i, k) = |(i-1)/n_0 - (k-1)/n_0| + |(i-1) \bmod n_0 - (k-1) \bmod n_0| \quad (\text{K6})$$

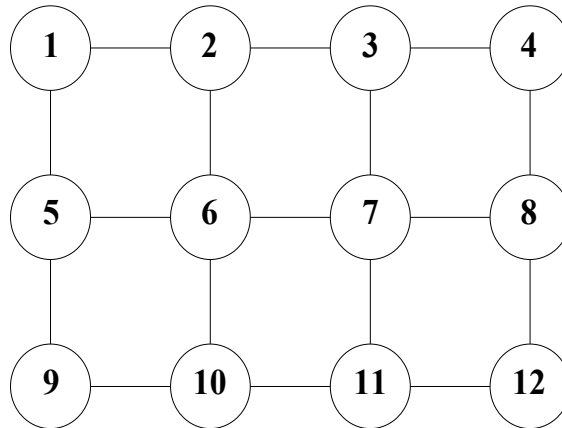
kur $|\cdot|$ – absolūtā vērtība (modulis),

i, k – neironu kārtas numuri,

n_0 – neironu skaits vienā rindā,

$/$ – veselu skaitļu dalīšana (atmetot atlikumu),

$\bmod$ – veselu skaitļu dalījuma atlikums.



Attēls K2. Kohonena slānis ar neironu numerāciju 2-dimensiju izkārtojumā. Neironu skaits $n=12$. Neironu skaits vienā rindā $n_0=4$. Slāņa topoloģija atbilst formulai K6.

Vienas dimensijas gadījumā attālumu starp neironiem aprēķina šādi.

$$d(i, k) = |i - k| \quad (\text{K6a})$$

kur $|\cdot|$ – absolūtā vērtība (modulis),

i, k – neironu kārtas numuri.

Kaimiņa funkcija (kaimiņa faktors).

Kaimiņa funkcija nosaka pakāpi, kādā attālums no uzvarētāja neirona ietekmē dotā neirona apmācību (attēls K3). Vienkāršākā kaimiņa funkcija D_0 ir sliekšņveida funkcija un nosaka, ka neirons – uzvarētājs apmācās par 100%, bet pārējie neapmācās nemaz (formula K7).

$$D_0(z) = \begin{cases} 1; & z=0 \\ 0; & z>0 \end{cases} \quad (\text{K7})$$

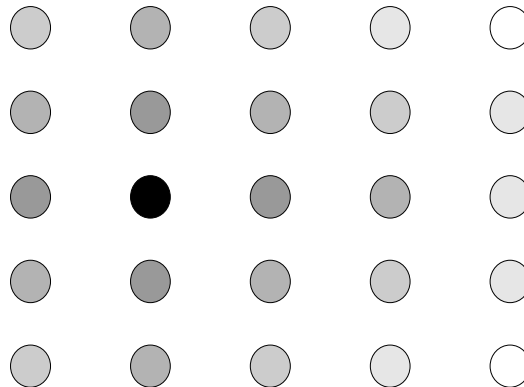
kur z – attālums starp neironiem.

Salīdzinoši sarežģītāka ir Gausa funkcija. Tā dod iespēju apmācīties arī citiem neironiem – noteiktu pakāpi mazākā mērā nekā uzvarētājam (formula K7a).

$$D_{Gauss}(z) = e^{-\frac{z^2}{2\sigma^2}} \quad (\text{K7a})$$

kur z – attālums starp neironiem,

σ (sigma) – līknes slīpuma (liekuma) faktors.



Attēls K3. Kaimiņa funkcijas efekts, ja neirons – uzvarētājs atrodas 3. rindas 2. pozīcijā.

Apmācības procesa konverģences nosacījums (pārbauda katras epohas beigās)

Apmācības procesu beidz, kad ir veikts noteikts skaits epohu (UMAX):

$$u > UMAX \quad (\text{K8})$$

kur u – kārtējās epohas numurs,

UMAX – maksimālais pieļaujamais epohu skaits.

K3.3. APMĀCĪBAS PROCESA SPECIFISKAS LIETAS.

Apmācības procesa divas fāzes.

Kohonena tīkla apmācības procesu varētu iedalīt 2 fāzēs [Haykin, 1999]:

1. Pašorganizēšanās (kārtšanas) fāze;
2. Konverģences fāze.

Pašorganizēšanās fāzes laikā notiek topoloģiskā sakārtošana un neironu svaru vektori nonāk aptuveni vajadzīgajās vietās. Pašorganizēšanās fāze varētu prasīt apmēram 1000 vai vairāk iterāciju (epohu).

Konverģences fāze nodrošina t.s. īpašību kartes (*feature map*) noprecizēšanu, lai nodrošinātu ieejas paraugu telpas pareizu statistisko kvantizāciju. Konverģences fāze ilgst vismaz $500n$ iterāciju, kur n ir neironu skaits tīklā.

Apmācības koeficienta un līknes slīpuma koeficienta atkarība no laika.

Apmācības procesā atbilstoši formulām K4 un K7a piedalās 2 konstantes – apmācības koeficients η (eta) un kaimiņa funkcijas līknes slīpuma koeficients σ (sigma).

Šie koeficienti apmācības procesa gaitā var mainīties – laika gaitā pakāpeniski samazinoties gan apmācības koeficientam η (palēninās apmācība), gan līknes slīpuma koeficientam σ (līkne paliek stāvāka).

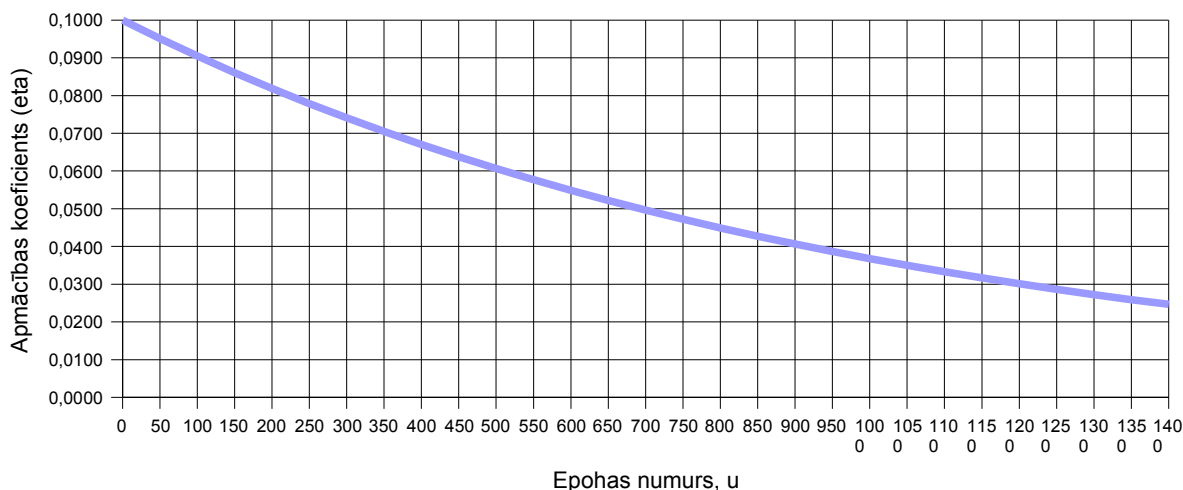
Apmācības koeficienta izmaiņa laikā.

Apmācības koeficienta izmaiņu laikā raksturo šāda formula:

$$\eta(u) = \eta_0 e^{\left(-\frac{u}{\tau_\eta}\right)} \quad (\text{K11})$$

kur η_0 – apmācības koeficienta sākuma vērtība,
 τ_η – apmācības koeficienta izmaiņas ātrums,
 u – iterācijas (epohas) numurs 0, 1, ...

Apmācības process būtu jāsāk ar apmācības koeficientu, kas ir apmēram 0.1. Tam būtu lēnām jādilst, bet pašorganizēšanās fāzes laikā jāpaliek virs 0.01. Šādiem nosacījumiem atbilst sākuma vērtību izvēle $\eta_0 = 0.1$ un $\tau_\eta = 1000$ (attēls K4).



Attēls K4. Apmācības koeficienta η izmaiņa pašorganizēšanās fāzē.

Konverģences fāzē apmācības koeficientam jābūt mazam un jāpaliek ar kārtu 0.01.

Līknes slīpuma koeficienta izmaiņa laikā.

Līknes slīpuma koeficienta izmaiņai jānodrošina, lai kaimiņa funkcija laika gaitā paliek stāvāka, līdz ar to apmācāmo neironu apgabalam koncentrējoties ap neironu – uzvarētāju.

Līknes slīpuma koeficienta izmaiņu laikā raksturo šāda formula:

$$\sigma(u) = \sigma_0 e^{\left(-\frac{u}{\tau_\sigma}\right)} \quad (\text{K12})$$

kur σ_0 – līknes slīpuma koeficienta sākuma vērtība,

τ_σ – līknes slīpums koeficienta izmaiņas ātrums,

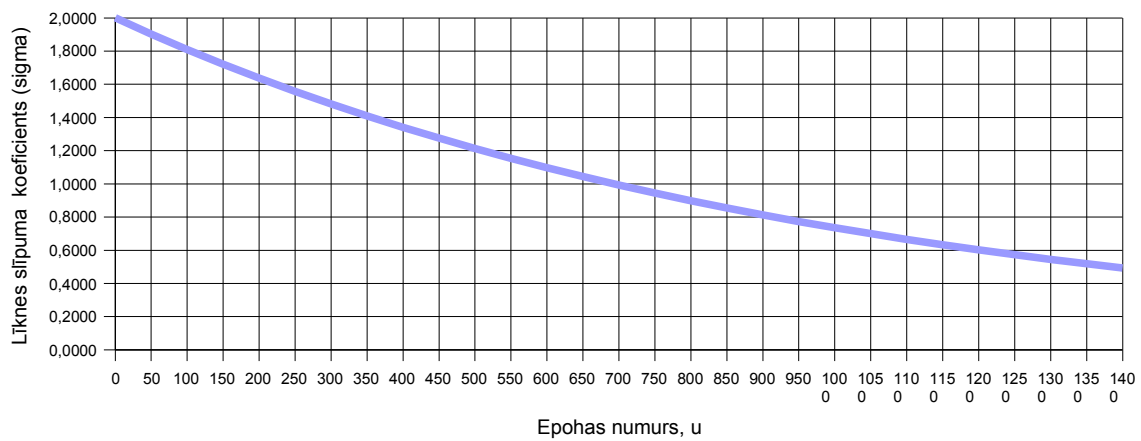
u – iterācijas (epohas) numurs 0, 1, ...

Divdimensiju Kohonena slānim par līknes slīpuma koeficienta sākuma vērtību der

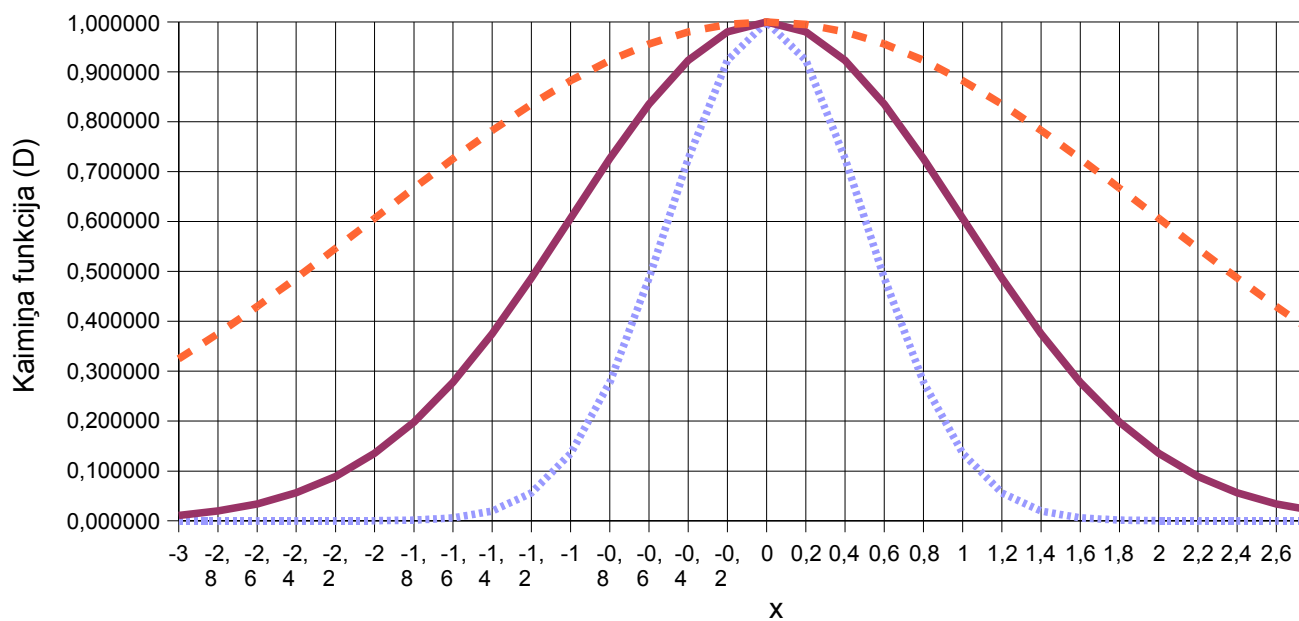
Kohonena slāņa rādiuss, t.i. apmēram $\sigma_0 = \frac{\sqrt{n}-1}{2}$, kur n – neironu skaits slānī, bet par

otras konstantes vērtību der $\tau_\sigma = \frac{1000}{\log(\sigma_0)}$. Izvēloties šādu līknes slīpuma izmaiņu,

apmācības procesa beigās kaimiņa funkcijai būtu jāsaturs tikai daži paši tuvākie kaimiņi vai pat neviens kaimiņš.



Attēls K5. Līknes slīpuma koeficienta σ izmaiņa apmācības procesa laikā ($\sigma_0=2$, $\tau_\sigma=1000$).



Attēls K6. Kaimiņa funkcijas vērtības atkarība no līknes slīpuma koeficienta σ .

K4. KOHONENA TĪKLA PIEMĒRS.

K4.1. UZDEVUMA NOSTĀDNE UN TĪKLA KONSTRUEŠANA.

Uzdevums.

Izveidot Kohonena tīklu ar piecām ieejām un divām izejām un apmācīt dotajiem paraugiem.

Apmācības paraugi.

Doti 3 apmācības paraugi garumā 5.

$$P_1 = \{1, 1, 0, 1, 0\};$$

$$P_2 = \{1, 1, 1, 0, 0\};$$

$$P_3 = \{0, 0, 1, 1, 1\};$$

Veidojamā Kohonena tīkla uzbūve.

Tīklam ir piecas ieejas un divas izejas ($m=5$, $n=2$), tātad tas sastāv no diviem neironiem izejas slānī. Piemērā uzskatāmības dēļ ieejas vērtības apzīmēsim ar $x_1, \dots, x_5$, bet izejas vērtības ar y_1 un y_2 .

K4.2. APMĀCĪBAS PROCESS.

Pieņemsim, ka tīkla svaru vērtības $\{w_{11}, \dots, w_{15}, w_{21}, \dots, w_{25}\}$ inicializētas ar sākuma vērtībām $\{0.03, 0.25, 0.2, 0.48, 0.01, 0.29, 0.21, 0.41, 0.42, 0.26\}$.

Attālums starp neironiem tiek definēts formulā K6a.

Apmācības koeficients $\eta=0.25$.

Maksimālā pieļaujamā kļūda $\epsilon=0.2$.

Maksimālais pieļaujamais epohu skaits $UMAX = 100$.

Algoritms.

Neironu tīkls tiek inicializēts ar sākuma vērtībām.

Katrā solī notiek šādas darbības (1-5):

1. Kārtējā parauga vērtību uzstādīšana uz tīkla ieejām.
2. Summēšanas funkcijas aprēķināšana katram neironam j (formula K1):
$$NET_j = (x_1 - w_{j1})^2 + (x_2 - w_{j2})^2 + (x_3 - w_{j3})^2 + (x_4 - w_{j4})^2 + (x_5 - w_{j5})^2.$$
3. Izejas vērtības aprēķināšana katram neironam (formula K3).
4. Svaru korekcijas aprēķināšana uzvarētājam neironam nākošajam solim $t+1$ no izrēķinātajām vērtībām solī t un svaru korekcija (formulas K4, K5, K7):
$$w_{win,i}(t+1) = w_{win,i}(t) + 0.25(x_i - w_{win,i}(t)).$$
5. Ja epohas beigas, tad aprēķināt epohas kopējo nosacīto kļūdu E (formula K10a) un ja kļūda ir kļuvusi mazāka par maksimāli pieļaujamo ($\epsilon=0.2$) vai arī soļa numurs t

pārsniedz maksimāli pieļaujamo (formula K8a), tad beigt, citādi turpināt ar nākošo soli, atgriežoties punktā 1.

Tabula K1. Algoritma demonstrācija (1 epoha = 3 soli).

Pa-raugi	1. neirons							2. neirons							E _t	E _u	Δ E _u
	w ₁₁	w ₁₂	w ₁₃	w ₁₄	w ₁₅	NET ₁	y ₁	w ₂₁	w ₂₂	w ₂₃	w ₂₄	w ₂₅	NET ₂	y ₂			
11010	0,03	0,25	0,2	0,48	0,01	1,81	0	0,29	0,21	0,41	0,42	0,26	1,7	1	1,7		
11100	0,03	0,25	0,2	0,48	0,01	2,37	0	0,47	0,41	0,31	0,56	0,19	1,46	1	1,46		
00111	0,03	0,25	0,2	0,48	0,01	1,95	1	0,6	0,56	0,48	0,42	0,14	2	0	1,95	5,11	
11010	0,02	0,19	0,4	0,61	0,26	1,99	0	0,6	0,56	0,48	0,42	0,14	0,94	1	0,94		
11100	0,02	0,19	0,4	0,61	0,26	2,41	0	0,7	0,67	0,36	0,57	0,11	0,94	1	0,94		
00111	0,02	0,19	0,4	0,61	0,26	1,1	1	0,78	0,75	0,52	0,43	0,08	2,57	0	1,1	2,98	2,13
11010	0,02	0,14	0,55	0,71	0,44	2,29	0	0,78	0,75	0,52	0,43	0,08	0,72	1	0,72		
11100	0,02	0,14	0,55	0,71	0,44	2,61	0	0,83	0,81	0,39	0,57	0,06	0,76	1	0,76		
00111	0,02	0,14	0,55	0,71	0,44	0,62	1	0,87	0,86	0,54	0,43	0,05	2,95	0	0,62	2,1	0,88
11010	0,01	0,11	0,66	0,78	0,58	2,6	0	0,87	0,86	0,54	0,43	0,05	0,66	1	0,66		
11100	0,01	0,11	0,66	0,78	0,58	2,84	0	0,91	0,89	0,41	0,57	0,03	0,7	1	0,7		
00111	0,01	0,11	0,66	0,78	0,58	0,35	1	0,93	0,92	0,56	0,43	0,03	3,19	0	0,35	1,71	0,39
11010	0,01	0,08	0,75	0,84	0,69	2,89	0	0,93	0,92	0,56	0,43	0,03	0,65	1	0,65		
11100	0,01	0,08	0,75	0,84	0,69	3,06	0	0,95	0,94	0,42	0,57	0,02	0,67	1	0,67		
00111	0,01	0,08	0,75	0,84	0,69	0,19	1	0,96	0,96	0,56	0,43	0,01	3,32	0	0,19	1,52	0,19

Pēc neironu svaru vērtībām apmācības beigās redzams, ka pirmais neirons apmācījies uz paraugu P₃, bet otrais neirons – uz paraugiem P₁ un P₂.

K5. KOHONENA TĪKLA NOVĒRTĒJUMS UN ĪPAŠĪBAS.

Kohonena algoritms zināmā mērā ir līdzīgs procesiem, kas notiek smadzenēs. Kohonena tīkls paredzēts ieejas signālu vektoru sadalīšanai grupās.

Kohonena tīkls ir noturīgs pret traucējumiem (ieejas signālu kropļojumiem).