

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM
VAZIRLIGI
MIRZO ULUG‘BEK NOMIDAGI O‘ZBEKISTON MILLIY
UNIVERSITETI**

Qo‘lyozma huquqida

UDK 519.95

Odilov Jahongir Qobil o‘g‘li

**“O‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion
modellarni yaratish”**

5A330101 – Axborot tizimlarining matematik va dasturiy ta’minoti

Magistr akademik darajasini olish uchun yozilgan

DISSERTATSIYA

Ilmiy rahbar:

dots. A.X.Ishmuhammedov

Toshkent – 2019 yil

MUNDARIJA

KIRISH	3
I BOB. KOHONEN NEYRON TO‘RLARI VA XARITALARI	7
1.1-§. Kohonen neyron to‘rini qurish printsiplari.....	7
1.2-§. Kohonen neyron to‘rini o‘qituvchisiz o‘rganish	15
1.3-§. Kohonen xaritalarini qurish prinsiplari	19
1.4-§. Kohonen xaritalar tuzish algoritmlari.....	22
1.5-§. O‘rganishning paketli algoritmlari	24
I Bob bo‘yicha xulosa	27
II BOB. KOHONEN NEYRON TO‘RIDA KLASTERIZATSIYA VA KLASTER TAHLIL	28
2.1-§. O‘z o‘zini tashkil qiluvchi Kohonen neyron to‘rida klasterizatsiya	28
2.2-§. Test masalalar klaster tahlilni o‘tkazish	29
II BOB bo‘yicha xulosa	42
III BOB. KOHONEN XARITASI USULIDA INFORMATSION MODELLARINI YARATISH VA TEST MASALALARI	43
3.1-§. O‘z-o‘zini tashkil qiluvchi Kohonen xaritasining informatsion modelini dasturiy taminoti.....	43
3.2-§. Dasturiy ta‘minotning modular tasnifi va tuzilish strukturasi.....	44
3.3-§. O‘z-o‘zini tashkil qiluvchi Kohonen xaritasining informatsion modelga yo‘riqnima	47
III BOB bo‘yicha xulosa	61
XULOSA	62
ADYOTLAR RO‘YXATI	63
ILOVA 1	65
ILOVA 2	100

KIRISH

Bugungi kunda fan va texnikaning jadal rivojlanishi natijasida barcha sohalarni tizimga tushirish va ularni avtomatlashtirish masalasi katta ahamiyatga ega bo'lmog'da.

Kadrlar tayyorlash milliy dasturi joriy qilingandan keyin o'qitish tizimida bir qator o'zgarishlar yuz berdi. Axborot texnologiyalarini rivojlanishi va uni joriy etish borasida qator qonunlar qabul qilindi. Bularga O'zbekiston Respublikasining "O'qitish to'g'risida"gi, "Kadrlar tayyorlash milliy dasturi", O'zbekiston Respublikasining "Axborotlashtirish to'g'risida"gi qonunlarini misol qilishimiz mumkin. Shuningdek qarorlarda ham oliy o'qitishga bo'lgan e'tiborni ko'rish mumkin.

Hozir kunda sun'iy intellektga asoslandan turli xil information modellar yaratilishning zarurligi va ularning amaliy ishlatilishga bo'lgan ehtiyoj tobora ortib borayotgani hech kimga sir emas. Bu kabi informatsion modellar jamiyat hayotidagi turli xilda katta va kichik muammolarni oson va tez echiliga salmoqli hissa qo'shadi.

Dissertatsiya ishi mavzusining asoslanishi va dolzarbligi. Bugungi kunga sun'iy intellekt sohasidagi ko'plab muammolarni echish klassifikatsiya masalasiga bog'liq bo'ladi. Shu sababli Kohonenning o'z-o'zini tashkil qiluvchi xaritalariga talab oshib bormog'da. Ushbu usulning boshqa neyron to'rlardan asosiy farqi, o'qituvchisiz o'rganish usulini qo'llanilgan ya'ni o'rganishda natija faqatgina kirish ma'lumotlariga bog'liq. Bundan tashqari bu usulda natijani vizual ko'rish mumkin bo'lgani uchun Kohonenning o'z-o'zini tashkil qiluvchi xaritalarini ko'plab sohalarda tadbiq qilinmog'da. Masalan, tibbiy sohada, geologiya sohada va boshqa sohalarda. "O'z-o'zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion modellarni yaratish" dolzarb muammolardan biri chunki, bir sohada qo'llanganda juda yaxshi natija bergan model boshqa bir sohada qo'llansa kerakli natijalarga erishib bo'lmasligi mumkin. Ushbu dissertatsiya ishida Kohonen xaritasi usuli asosida informatsion modellar yaratilgan.

Tadqiqot obyekti va predmeti. Tadqiqot obyekti o‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion model yaratish. Xususan tibbiyot muassasalari, geologiya sohasidagi, genetika va ta’lim tizimiga olishimiz mumkin. Tadqiqotning predmeti sifatida tadqiqotda ishlatilgan metodlarni va algoritmlarni olishimiz mumkin.

Dissertatsiya ishining maqsadi va vazifasi. Ushbu dissertatsiya ishining maqsadi o‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion modellarni yaratish. Yaratilgan modellarni bir nechta tanlanmalarga qo‘llagan holda eksprement hisoblovchi modellarni yaratish. Ilmiy ishning vazifasi esa tizimda kiritilgan xar bir tanlanma alomatlariga ko‘ra klasterlarda to‘g‘ri ajratish va natijani vizual xaritada akslantirish.

Dissertatsiy maqsadiga ershish uchun quyidagi vazifalar bjarilishi kerak:

- tadqiqot mavzusi bo‘yicha adabiyotlar tahlili o‘tkazish;
- dissertatsiya ishida masalani qo‘yilishi aniqlash;
- masalani echish uchun metod va usulni tanlash;
- klasterizatsiya masalasini echish;
- klaster tahlilni o‘tkazish;
- dasturiy taminotni yartish;
- bir nechta eksprement tanlanmalarni informatsoin modelda tadbiq qilib ko‘rish.

Mavzuning ilmiy yangiligi. Mavzuning ilmiy yangiligi shundan iboratki, turli alomatlar fazosida berilganlarga “O‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usuli” ni qo‘llashning yangi informatsion modellari ishlab chiqildi va oldin qo‘llanilmagan evirestik metodlardan foydalaniladi. Yaratilgan modelni bir nechta exprement o‘tkazildi va natijalar olindi. Yaratilgan gipotezalar to‘g‘riligi isbotlandi.

Tadqiqotning asosiy masalalari va farazlari. Klasterizatsiya masalasini, klaster tahlil qilish masalasi va natijani ikki o‘lchovli Kohonen xaritasida akslantirib natijani vizual ko‘rinishda hosil qilish.

Tadqiqot mavzusi bo'yicha adabiyotlar sharhi (tahlili). T.Kohonen ning "Self-Organizing Map" ilmiy asarida o'z-o'zini tashkil qiluvchi Kohonen xaritasi haqida umumiy tushunchalar berilgan. Стариков Алексей ning "Самоорганизующиеся карты Кохонена – математический аппарат" ilmiy asarida Kohonen xaritasi usulining matematik asoslari keltirilgan. Nils J. Nilson "The Quest for Artificial Intellegence" asarida esa sin'iy intellektning asosiy tushunchalari keltirilga.

Tadqiqotda qo'llanilgan metodikaning tavsifi. Ushbu tadqiqot ishida turli alomatlar fazosida berilganlarga. "O'z-o'zini tashkil qiluvchi Kohonen xaritasi usuli" ni qo'llashning yangi modellari ishlab chiqilgan. Tanlanmalarni klasterlarga ajratish uchun "G'olib barchasini oladi", "k-yaqin qo'shni" va "G'olib ko'prog'ini oladi" kabi bir qancha metodlardan foydalanilgan. Natijada har bir obyekt uchun individual xulosalar berish mumkin. Barcha tanlanmalarni alomatlariga ko'ra to'g'ri guruhlariga ajratilib har bir guruhning o'ziga hos xususiyatlari topilgan.

Tadqiqot natijalarining nazariy va amaliy ahamiyati. Ushbu tadqiqot ishida qo'llanilgan metodni obyektlararo turg'unlikni aniqlashda qo'llash mumkin. Shuningdek, ixtiyoriy miqdoriy alomatlardan iborat tanlanmalarni guruh to'g'ri ajratish va natijani vizual xarita ko'rishda muxim amaliy ahamiyatga ega.

Ish tuzilmasining tarkibi. Ushbu magistrlik ilmiy tadqiqot ishida o'z-o'zini tashkil qiluvchi Kohonen xaritasi usuli o'rganib chiqilgan va "O'z-o'zini tashkil qiluvchi Kohonen xaritasining informatsion modeli" yaratilib ba'zi eksprement tanlanmalarda tadbiqui keltirilgan. Ushbu magistrlik ilmiy tadqiqot ishi kirish, uchta bob, xulosa, foydalanilgan adabiyotlar ro'yxati va ilovadan iborat.

Tadqiqot ishning *birinchi bobida* Kohonen neyron to'rlari va Kohonen xaritalari tushunchasi, Kohonen neyron to'rini qurish printsiplari, Kohonen neyron to'rini o'qituvchisiz o'rganish, Kohonen xaritalarini qurish prinsiplari, Kohonen xaritalar o'rganish algoritmlari va algoritmni paketli o'rganish.

Bundan tashqari bu usulning boshqa usullardan kamchiligi va ustunliklarini ko'rib chiqilgan. Kiruvchi qiymatlar miqdoriy bo'lishi kerakligi, agar ba'zi alomatlari miqdoriy emas balki satr va boshqa ko'rinishlarda bo'lsa qaysi usullar yordamida miqdoriy ko'rinishga olib kelish keltirilgan. Agar miqdoriy qiymatlar katta qiymatlarda bo'lsa normallashtirish usulari keltirilgan.

Tadqiqot ishning *ikkinchi bobida* klasterlarga ajratish metod va usullari keltirilgan. Klasterlarga ajratib bo'lgandan so'ng natija qanchalik to'g'riligini tekshirish uchun klaster tahlil o'tkazish usullari ko'rib chiqildi. "O'z o'zini tashkil qiluvchi kohonen neyron to'rida klasterizatsiya usuli"ni dasturini tuzilda va bu datur "Talabalarning o'rtacha o'zlashtirishi" va "Iris guli" talanmalarida tadbiq qilindi. Talangan eksprement tanlanmalarda dastur yaxshi natija berdi.

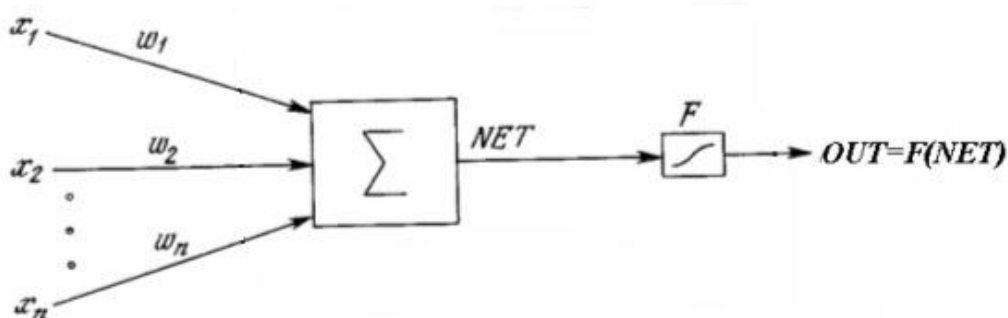
Tadqiqot ishining *uchinchi bobida* "O'z-o'zini tashkil qiluvchi Kohonen xaritasining informatsion modeli" ni dasturiy taminoti yatilgan. Bu informatsion modelni "Iris guli" talanmalarida tadbiq qilindi. Natijalar asosida Kohonen xaritasi yaratildi va vizuallashtirildi. Talangan eksprement tanlanmalarda model yaxshi natija berdi.

I BOB. KOHONEN NEYRON TO‘RLARI VA XARITALARI

1.1-§. Kohonen neyron to‘rini qurish printsiplari

Tirik mavjudotlar ichida eng oliysi bo‘lgan inson har qadamda anglash, o‘rganish va qaror qabul qilishga duch keladi. Neyroto‘rli yo‘nalish miya shunday murakkab masalalarni qay tarzda yechishi va bu prinsiplar avtomatik qurilmalarda qanday realizatsiya qilinishini tushunishga harakatdan paydo bo‘ldi. Hozircha sun‘iy neyron to‘rlari (artificial neural networks, ANN) tabiiy neyron to‘rlarning imkon doirasida soddalashgan eng oxirgi ko‘rinishidir. Inson va jonivorlarning nerv tizimi zamonaviy texnologiyalar yordamida yaratish mumkin bo‘lgan qurilmalardan ancha murakkabdir. Shunday bo‘lsada, nerv tizimlarining umumiy prinsiplaridan foydalanib ko‘pgina amaliy masalalarni muvaffaqiyatli yechishga erishilmoqda.

Sun‘iy neyron – biologik neyronning ba’zi funksiyalarini bajaruvchi ANN elementi bo‘lib, uning asosiy vazifasi kirish signallariga bog‘liq holda chiqish signalini shakllantirishdan iborat. Eng keng tarqalgan konfiguratsiyalarda kirish signallari moslashuvchi summatorlar orqali silliqiladi, keyin summatorning chiqish signali noxiziqli o‘zgartirgich (aktivatsion funksiyasi)ga kiradi va u yerda ham o‘zgartirilib chiqishga uzatiladi [3].



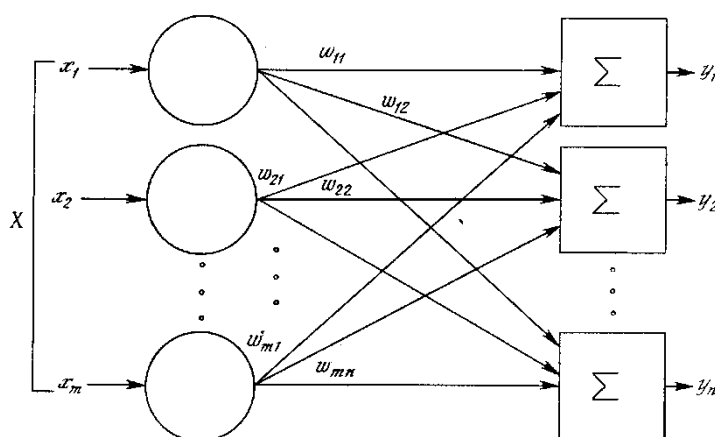
1-rasm. Sun‘iy neyron

Aktivlashtirish funksiyasi – sun‘iy neyronning chiqish signalini (OUT) hisoblovchi noxiziqli funksiya. Aktivatsionlar sifatida asosan quyidagi funksiyalar ishlatiladi [8]:

Neyroinformatika – biologik neyron to‘rlar ishlash prinsipi asosida hisoblash tizimlarini tadqiq qiladigan fan yo‘nalishi. Oldingi avlod hisoblash qurilmalaridan bunday tizimlarning asosiy farqi:

- hisoblashda yuqori parallellik;
- mavjud berilganlarni umumlashtirish qobiliyati;
- programmash o‘rnini o‘rgatish bilan almashtirish;
- shovqinga nisbatan yuqori turg‘unlik.

Neyron to‘rining arxitekturasi – neyron to‘rlari alohida elementlarining bog‘lanish va tashkillashtirish usuli.



2-rasm. Bir qatlamli neyron to‘ri.

Neyronlarning arxitektura farqi har xil aktivatsion funksiyadan foydalanishiga bog‘liq. Neyron to‘rlari arxitekturasiga ko‘ra 2 sinfga ajratish mumkin: to‘g‘ri tarqalgan to‘rlar va rekurrent to‘rlar.

1-jadval. To‘g‘ri tarqalgan va rekurrent to‘rlarning solishtirish.

	To‘g‘ri tarqalgan to‘rlar (teskari aloqasiz)	Rekurrent to‘rlar (teskari aloqali)
Afzalliklari	Oson realizatsiya. Kafolatlangan (matematik isbotlangan) yaqinlashuvchi hisoblash.	Neyron to‘rlari sonining kamligi (to‘g‘ri tarqalgan to‘rlarga nisbatan)
Kamchiliklari	Masala murakkabligi	Yaqinlashuvchi

	oshishi bilan neyronlar sonining tez o'sishi.	hisoblashni ta'minlovchi qo'shimcha shartlarning kerakliligi.
--	-----------------------------------------------	---------------------------------------------------------------

Neyroto'rlarni o'rgatish algoritmi – to'rni talab etilgandek ishlatish maqsadida arxitektura, sinaptik bog'lanishlar vazni va o'rgatuvchi tanlov bo'yicha neyron ostonalari sozlash protsedurasidir. O'rgatishning asosiy paradigmalari [3]:

“O'qituvchili o'rgatish” – o'rgatuvchi tanlovda kirish-chiqish juftliklari, ya'ni, har bir kirish uchun to'g'ri javoblar (to'rlarning chiqishi) ma'lum bo'ladi.

“Yordam bilan o'rgatish” – to'g'ri javoblar ma'lum emas, lekin, to'r chiqishi to'g'riligining kritik bahosi ma'lum.

“O'qituvchisiz o'rgatish” – o'rgatish tanlovi sifatida faqat kirish qiymatlaridan foydalaniladi.

“Aralash o'rgatish” – bir qism vaznlar “o'qituvchili o'rgatish” orqali, qolgan o'z-o'zini o'rgatish bilan topiladi.

Neyroto'rni o'rgatuvchi genetik algoritmi bu – neyroto'rni optimal arxitekturasini evolyusion yo'l bilan topuvchi algoritmdir. Bir nechta to'rlar tasodifiy arxitektura bilan yaratiladi. Har bir to'r genetik kodning xromosomasi sifatida qaralishi mumkin. Xromosomalar ustida chatishtirish (crossover), urchitish, mutatsiya amallari bo'lishi mumkin. Moslashish (fitness) funksiyasini hisoblashda berilgan qadamdagi eng optimal to'rlar arxitekturasi tanlanadi [7].

Obyekt va alomatlar. Berilgan x obyektning f alomati (feature) bu – obyektning qandaydir xususiyatini o'lchash natijasidir. Rasmiy tomondan alomat deb $f: X \rightarrow D_f$ akslantirishga aytiladi, bu yerda D_f – alomatning ruxsat etilgan qiymatlar to'plami. Xususiyl holda, ixtiyoriy $a: X \rightarrow Y$ algoritmi alomat sifatida ham qaralishi mumkin [10].

Tabiatan bog'lanishiga ko'ra D_f alomatlar to'plami bir nechta toifaga bo'linadi:

Agar $D_f = \{0, 1\}$ bo'lsa, f – binar alomat;

Agar D_f chekli to'plam bo'lsa, u holda f – nominal alomat;

Agar D_f chekli tartiblangan to'plam bo'lsa, u holda f – tartiblangan alomat;

Agar $D_f = \mathbb{R}$ bo'lsa, u holda f – miqdoriy alomat deyiladi.

Agar barcha alomatlar bir xil toifali bo'lsa, u holda boshlang'ich berilganlar bir toifali, aks holda har xil toifali deyiladi.

X^l tanlovdagi barcha obyektlarning $l \times n$ o'lchovli jadval ko'rinishda alomatlari tavsifi jamlanmasiga “obyekt-alomat” matritsa deyiladi. Obyekt-alomat matritsasi amaliy masalalarda boshlang'ich berilganlarni ifodalashni standart va eng ko'p tarqalgan ko'rinishi hisoblanadi.

Javoblar va masalalar toifalari

Precedent bo'yicha o'rgatish masalalari ruxsat etilgan javoblar to'plami Y ning tabiatiga ko'ra quyidagi toifalarga ajraladi:

Agar $Y = \{1, \dots, M\}$ bo'lsa, u holda bu masala - M ta kesishmaydigan sinfda klassifikatsiya masalasi deyiladi. Bu holatda barcha X obyektlar to'plami $K_y = \{x \in X: y^*(x) = y\}$ sinflarga ajratiladi va algoritm $a(x)$ “ x qaysi sinfga tegishli?” savoliga javob berishi shart. Ba'zi ilovalarda sinflarni obrazlar va masalani esa obrazlarni anglash (pattern recognition) masalasi ham deyiladi .

Agar $Y = \{0, 1\}^M$ bo'lsa, u holda masala - M ta kesishadigan sinfda klassifikatsiya masalasi deyiladi. Bunday holatda masala M ta mustaqil ikkita kesishmaydigan sinf klassifikatsiya masalasiga keltiriladi.

Agar $Y = \mathbb{R}$ bo'lsa, masala regressiya (regression estimation) masalasi deyiladi. Prognoz masalalari (forecasting) klassifikatsiya va regressiya masalalarining xususiy xoli hisoblanadi, ya'ni, $x \in X$ – x obyektning o'tmishdagi holatining tavsifi, $y \in Y$ – x obyektning kelajakdagi holatining ba'zi tavsifi bo'ladi.

Kohonen neyron to'rlari deganda o'z-o'zini tashkil qiluvchi neyron to'rlarni nazarda tutiladi. O'z-o'zini tashkil qiluvchi to'r ma'lum umumiy

xususiyatlarga ega kirish vektorlarining klasterlarini (guruhlarini) aniqlash imkonini beradi [1].

Klasterli tahlil – berilganlarni to‘plash, tanlov obyektlari haqidagi ma’lumotlarni saqlovchi va ularni bir jinsli guruhlariga nisbatan tartiblashni bajaruvchi ko‘p o‘lchamli statistik protseduradir [1].

Klasterizatsiya- bu tekshirilayotgan obyektlarni "o‘xshash" obyektlar klasterlari deb nomlangan guruhlariga ajratilishi. "Klaster" atamasi (Inglizcha Klaster - paket, guruh) sinf, takson, guruh atamalariga sinosim atamadir. Klasterizatsiya usuli klassifikatsiya usulidan tubdan farq qiladi. Klassifikatsiya usuli echimi predmetlarni har biriga predmet sinflardan biriga tayinlashdir. Klasterizatsiya vazifasida obyekt oldindan aniqlanmagan sinflardan biriga tayinlanish. Klasterlardagi obyektlarning klasterlarga ajratish bir vaqtning o‘zida klasterlar shakllanishi bilan amalga oshiriladi [**Ошибка! Источник ссылки не найден.**].

Klasterizatsiya, ma’lumotlarni yig‘ish vazifalarini bir qator hal qilishni osonlashtiradigan o‘xshash ma’lumotlarni to‘plash imkonini beradi [6].

- Ma’lumotlarni tahlil qilish, tahlil qilishni osonlashtirish. Olingan guruhlarini mazmunli tahlil qilish qonuniyatlarni aniqlash imkonini beradi. Masalan, siz uyali aloqa to‘ri mijozlarini aniqlashingiz mumkin, ular uchun yangi tarif rejasini taklif qilishingiz mumkin. Boshqa misollar - savdo to‘rining xaridor guruhlarini aniqlash, bozorni segmentlash. Klaster tarkibining tahlili turli xil guruhlarining obyektlariga tahlilning turli usullarini qo‘llash imkonini beradi;

- Prognozlash. Klasterlardan biriga yangi obyektini tayinlash orqali, uning xatti-harakati klaster obyektlarining xatti-harakati bilan o‘xshash bo‘lgani uchun, obyekt xatti-harakatlarini oldindan belgilashingiz mumkin;

- Anomaliyalarni aniqlash. Guruhlarni to‘g‘ri aniqlanishi anomaliyalarni aniqlashga yordam beradi. Odatda, bu klaster, yerda juda oz narsalar tushadi.

Har bir klasterini to‘g‘ri aniqlanishining ahamiyatini e’tiborga olish muhimdir. Har bir guruhga klaster xossalarini aks ettiruvchi nom berilishi kerak.

Buning uchun obyektlarni klasterga birlashtiradigan xususiyatlarni aniqlash kerak. Bu klaster obyektining xususiyatlarini statistik tahlil qilishni talab qiladi.

Kohonen to'rlari yordamida miqdoriy xususiyatlari bilan berilgan obyektlar klasterlari hisoblanadi [12].

Formal ravishda Klasterizatsiya usuli quyidagicha amalga oshirali. Har bir obyektning vektor x_j , $j=1,2,\dots,n$ atributlari (alomatlari) bilan ifodalanadigan obyektlar majmuasini $I = \{i_1, i_2, \dots, i_n\}$ berilgan: $x_j = \{x_{j1}, x_{j2}, \dots, x_{jm}\}$. C klaster to'plamini tuzish talab etiladi, C to'plami F to'plamini I to'plamga $F: I \rightarrow C$ akslantirishdan hosil bo'ladi. Klasterizatsiya to'plami quyidagicha tuziladi [10,11].

$$C = \{c_1, c_2, \dots, c_k, \dots, c_g\},$$

bu yerda c_k - to'plamdagi "o'xshash" obyektlarini o'z ichiga olgan klaster:

$$c_k = \{i_j, i_p \mid i_j \in I \text{ va } d(i_j, i_p) < \sigma\},$$

σ - bu bir obyektida obyektlarni kiritish uchun yaqinlik o'lchovini belgilovchi qiymat, $d(i_j, i_p)$ masofa deb ataladigan obyektlar orasidagi yaqinlik o'lchovidir.

Agar masofa $d(i_j, i_p)$ ma'lum bir qiymatdan kamroq bo'lsa, obyektlar yaqin deb hisoblanadi va bir xil klasterga joylashtiriladi. Aks holda, obyektlar bir-biridan farq qiladi va ular turli guruhlarga joylashtirilgan deb hisoblanadi.

Klasterizatsiya, vektor orasidagi masofani ishlatishga asoslangan. $d(x, y)$ quyidagi shartlar qondirilgan bo'lsa, x va y vektorlari orasidagi masofa deb nomlanadi [6].

1. $d(x, y) \geq 0$, barcha x va y lar uchun.
2. $d(x, y) = 0$, faqat va faqat $x=y$ bo'lganda.
3. $d(x, y) = d(y, x)$.
4. $d(x, y) \leq d(x, k) + d(k, y)$ - uchburchak tensizligi.

Kohonen to'rlarida odatda Yevklid masofa ishlatiladi [7].

$$d_E(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \|x - y\|$$

Yevklid masofa x va y vektorlari orasidagi masofa vektorlar orasidagi farqning evklid normasini, yoki x va y nuqtalarini birlashtiruvchi masofani o'zida aks ettiradi.

Yevklid masofasi Minkovskiy (H.Minkovskiy) masofasidan xususiy holidi [13].

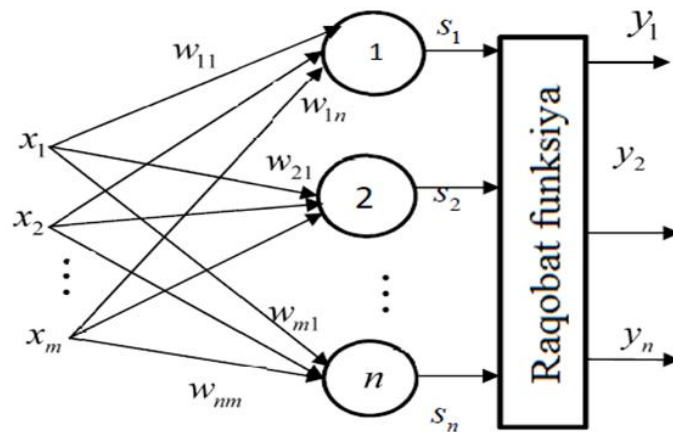
$$d_p(x, y) = \left(\sum_{i=1}^m |x_i - y_i|^p \right)^{1/p} = \|x - y\|_p,$$

Bu yerda $\|z\|_p = \left(\sum_{i=1}^m |z|^p \right)^{1/p}$ z ni p - normasi.

Agar 2- normasi bo'lsa –Yevklid masofasi bo'ladi.Yana bir alohida holat 1-norma, Manxetten masofa deb nomlanadi (shaharlardagi masofa).

$$d_1(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Kohonen neyron to'ring har bir qatlami WTA (*Winner Takes All* - g'olib hammasini oladi) tipidagi neyronlardan tashkil topgan to'rdir [14].



3-rasm. Kohonen neyron to'ring strukturasi

Neyron to'ring har bir neyroni m -o'lchovli $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ kirish vektorining barcha komponentlaridan iborat. Kirish vektori – klasterizatsiya obyektlarning birining ko'rinishi. Neyronlar soni to'ring ajratilishi kerak

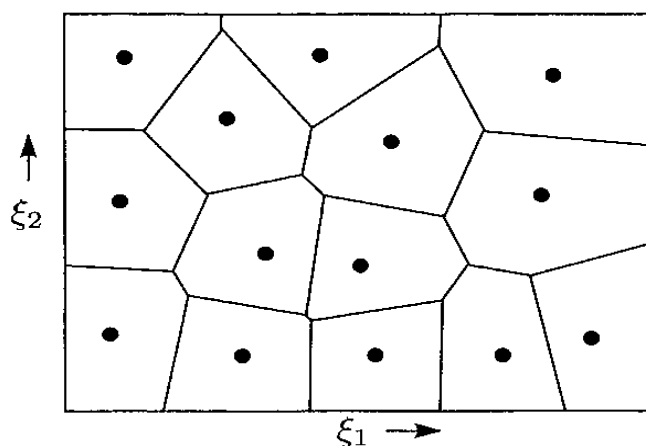
boʻlgan guruhlar soniga toʻgʻri keladi. Kohonen neyron toʻrining neyronlari qoʻshimchalar sifatida chiziqli ogʻirlikdagi belgilaydi.

$$s_j = b_j + \sum_{i=1}^m w_{ij} x_i$$

bu yerda j - neyronlarning raqami, i – kiritish raqami, s_j – moslashuvchan katakchanning chiqishi, w_{ij} - j -neyronning i - kirish ogʻirligi, b_j - chegara.

Har bir j - neyron vazn vektori $w_j = (w_{1j}, w_{2j}, \dots, w_{mj})$ bilan belgilanadi, bu yerda m - kirish vektorlarining komponentlari soni. Moslashuvchan qoʻshimchalar chiqqandan soʻng, signal "gʻolib hammasini oladi" qoidasiga koʻra ishlaydigan tanlov funktsiyasiga kiradi. Raqobat funktsiyasi maksimal chiqish qiymatiga ega boʻlgan moslashuvchan yigʻuvchining chiqishini topadi. k - bu kabi qabul qiluvchining raqami. Keyin toʻrning chiqishi $y_k = 1$ chiqish signali boʻlib, boshqa chiqish signallari esa nolga teng. Maksimal bir vaqtning oʻzida bir nechta chiqish signali birga teng boʻlsa, masalan, birinchisiga mos keladi.

Kohonen neyron toʻrining oʻrganish, vazn vektori tomonidan kiritilgan kirish vektorining oʻlchovi maʼnosida yaqinlik almashinishdagi xatolar yuzaga keltirishni kamaytiruvchi ogʻirliklar tanlovidir. Ushbu yondashuv vektorning kvantlashi deb ataladi va audio-video signallarni siqish muammolarida ishlatiladi. Vektorning kvantlash gʻoyasi kodli jadvalni tashkil qiluvchi kichik oʻlchovli qoʻllab-quvvatlash vektorlari yordamida koʻp oʻlchovli kirish vektorlarining ixcham koʻrinishini tashkil qiladi. Bunday holda Kohonen neyron toʻri kirish vektorlari gʻolib neyronlarning (klaster raqamlari) raqamlari bilan belgilangan boʻladi. Shunday qilib, kirish maydonining muayyan mintaqasidan kelgan barcha vektorlar ularning eng yaqin qoʻshnisi boʻlgan bir xil qoʻllab-quvvatlovchi vektor bilan almashtiriladi. Yevklid masofasidan foydalanilganda, kirish maydoni Voronoy (Вороной Г. Ф.). koʻpburchaklarga boʻlinadi. Voroniy ikki oʻlchovli koʻpburchagi 4-rasmda misol qilib keltirilgan [11].



4-rasm. Voronov ko'pburchagiga misol

Ko'p o'lchovli fazolarda Voronov ko'pburchagi gipersplanalar orqali hosil bo'ladi.

1.2-§. Kohonen neyron to'rini o'qituvchisiz o'rganish

Kohonen neyron to'rlarida o'qituvchisiz buyruq ishlatiladi. To'rni tuzish uchun raqobat mexanizmlari qo'llaniladi. To'rning kirishiga bir vektor x kiritilganda, vazn vektori kirish vektoridan kam farq qiladigan neyron oladi. G'olib neyron uchun, munosabat

$$d(x, w_j) = \min_{1 \leq i \leq n} d(x, w_i)$$

Bu yerda, n - neyronlarning soni, j - g'olib neyronning raqami, $d(x, w)$ - x va w vektorlari orasidagi masofa (tanlangan metrikaning ma'nosida). Ko'pincha, masofa o'lchovi sifatida, Yevkliddagi o'lchov olinadi.

$$d(x, w_j) = \|x - w_j\| = \sqrt{\sum_{j=1}^m (x_j - w_{ij})^2}$$

Boshqa o'lchovda masofalar ham qo'llaniladi.

Raqobatchan faollashtiruvchi funksiya katakchanning qiymatlarini tahlil qiladi va natijada minimal qiymatga ega bo'lgan "g'olib neyron" dan tashqari barcha neyronlar uchun 0 ga teng bo'lgan neyron chiqishi hosil qiladi.

Shunday qilib, chiqish vektorida 1 ga teng bitta element mavjud bo'lib, u neyronlarning g'olibiga to'g'ri keladi, qolganlari esa 0 ga teng. Faol

neyronlarning soni kirish vektorining eng yaqin bo'lgan guruhini (klaster) aniqlaydi.

Kohonen to'rida kirish qiymatlari kerakli (normal bo'lmasa ham) normallashtiriladi. Buni amalga oshirish uchun quyidagi formulalardan birini bosiladi:

$$x_{hi} = \frac{x_i}{\sum_{j=1}^m x_j^2}, \quad x_{hi} = \frac{x_i}{|x_i|}$$

bu yerda x_{hi} -kirish vektorining normallashtirilgan qismi. Kirish ma'lumotlarining normallashtirishi to'ring o'qitish tezligiga ijobiy ta'sir ko'rsatadi.

O'rganish jarayoni boshlanishidan oldin to'r hosil qilinadi, ya'ni vazn vektorlarining dastlabki instilizatsiya qilinadi. Eng oddiy holatda, vaznlarning tasodifiy qiymatlari berilgan. Kohonen to'rini o'rganish jarayoni qator bosqichlarni takroriy takrorlashdan iborat [2]:

1. Kirish ma'lumotlarini kirishga yuborish. Odatda bu kirish vektorlarining tasodifiy tanlovidir;
2. Har bir neyronning chiqishini topish;
3. "G'olib neyron"ni topish (uning vaznlari kirish vektorining tegishli tarkibiy qismlaridan farqli).
4. Kohonen qoidasiga ko'ra, "g'olib neyron" neyronlarning og'irliklarini o'zgartiriladi.

$$w_i^{(k+1)} = w_i^{(k)} + \eta_i^{(k)} \left[x - w_i^{(k)} \right]$$

bu yerda x - kirish vektori, k - o'qinishning siklini raqami, $\eta_j^{(k)}$ - i -neyronning k -siklidagi o'qitishning o'rganish tezligi koeffsienti.

5. Agar o'rganish tugatilmagan bo'lsa, 1-bosqichga o'ting. Xatolik funksiyasining kichik qiymatiga erishish mumkinligini tekshirish mumkin

$$E = \frac{1}{Q} \sum_{i=1}^Q \|x_i - w_{x_i}\|^2$$

bu yerda w_{x_i} – x_i kirish vektori g‘olib neyron og‘irlik vektori va Q – o‘rganish tanlanmasining o‘lchovi.

Shunday qilib, vazni vektorlari kirish vektoriga yaqin bo‘lgan neyron, yanada yaqinroq bo‘ladi. Natijada, bu neyron kirishga yaqin vektor yuborish orqali raqobatlashadi va sezilarli darajada farqli vektorni yo‘qotishni ta‘minlaydi. O‘rganishni bir necha marta takroriy ravishda amalga oshirilgandan so‘ng, vektor klasterga tegishli bo‘lganida 1 hosil qiluvchi neyron bo‘ladi va vektor klasterga tegishli bo‘lmaganida 0 hosil qiluvchi neyron bo‘ladi. Shunday qilib, to‘r kirish vektorlarini klassifikarsiyalash o‘rganadi.

Kohonen neyron to‘rini o‘rganish paytida "o‘lik" neyronlar deb ataladigan muammo tug‘iladi. Har qanday raqobatlashuvchi qatlamning cheklashlaridan birida ba‘zi neyronlarning ishlatilmay qolishi. Bu, kirish vektoridan sezilarli darajada ajralib chiqqan boshlang‘ich vaznli vektorlarga ega bo‘lgan neyronlar, o‘qishda qancha davom etishidan qat‘iy nazar, raqobatni g‘olib chiqmaydi. Natijada, bunday vektorlarni o‘qitish uchun ishlatilmaydi va tegishli neyronlar hech qachon g‘olib chiqmaydi. Bunday "mag‘lub neyron" larni "o‘lik" neyronlar deb ataladi, chunki ular foydali vazifani bajarmaydi. Shunday qilib, kirish ma‘lumotlari neyronlarning kichik soni bilan izohlanadi. Shuning uchun, biz barcha neyronlarni mag‘lub qilish uchun imkoniyat berishimiz kerak. Buning uchun o‘qitish algoritmi "o‘lik" neyronlarni o‘qitishda ishtirok etishini modifikatsiya qilish kerak [1,10].

Masalan o‘rganish algoritmi neyron tanlovi g‘olibni yo‘qotadigan tarzda o‘zgartiriladi. Neyronlarning faoliyatini hisobga olishning usullaridan biri – o‘rganish jarayonida har bir neyronning potentsialini hisoblash. Dastlab, neyronlarga $p_i(0) = \frac{1}{n}$ funktsiyasi beriladi, bu yerda n - neyronlarning (klasterlarning) soni, k -siklidagi o‘rganish salohiyati quyidagicha qoidalar bilan aniqlanadi:

$$p_i(k) = \begin{cases} p_i(k-1) + \frac{1}{n}, & i \neq j, \\ p_i(k-1) + p_{\min}, & i = j, \end{cases}$$

bu yerda j - g'olib vektor raqami.

Agar potensial $p_i(k)$ ning qiymati p_{\min} darajasidan past bo'lsa, u holda neyron "dam olish" dan olib tashlanadi. $p_{\min} = 0$ neyronlar kurashdan chiqarib tashlanmaydi. $p_{\min} = 1$ bo'lsa, neyronlar o'z navbatida g'alaba qozonadi, chunki har bir o'quv kursida faqat bittasi kurashishga tayyor. Amalda, $p_{\min} \approx 0.75$ bilan yaxshi natija olinadi.

Neural Network Toolboxda, "o'lik" neyronlarga qarshi kurashish uchun neyronlarning almashinuvidagi o'zgarish qo'llaniladi. O'lgan neyronlarning befarqligini hisobga olgan holda, mos keladigan o'zgartirish qoidasi *learncon* funksiyasi sifatida amalga oshiriladi va quyidagilardan iborat. O'rganish jarayonining boshida raqobatlashadigan qatlamning barcha neyronlariga bir xil faoliyat parametri berilgan: $c_0 = \frac{1}{N}$, bu yerda N - raqobatlashadigan qatlamning neyronlarning soni, klasterlar soniga teng. O'rganish jarayonida *learncon* funksiyasi bu parametрни faol neyronlar uchun qiymatlar kattalashishi va faol bo'lmaganlari uchun kamroq bo'ladi. Faoliyat parametrlari vektori uchun mos keladigan formulalar quyidagilar [6]:

$$c^{(k+1)} = (1 - r_i)c^{(k)} + r_i s^{(k)}$$

Bu yerda, r_i - tezligni sozlash parametrik, k - itoratsiya tartibi, $s^{(k)}$ - k -o'rganish jarayonida moslashuvchan qo'shimchalar chiqaradigan natijaviy vektori. Natijaviy vektorining tarkibiy qismlari quyidagi formula bo'yicha aniqlanadi.

$$b_i^{(k+1)} = e^{(1 - \ln c_i^{(k+1)})} - b_i^{(k)}$$

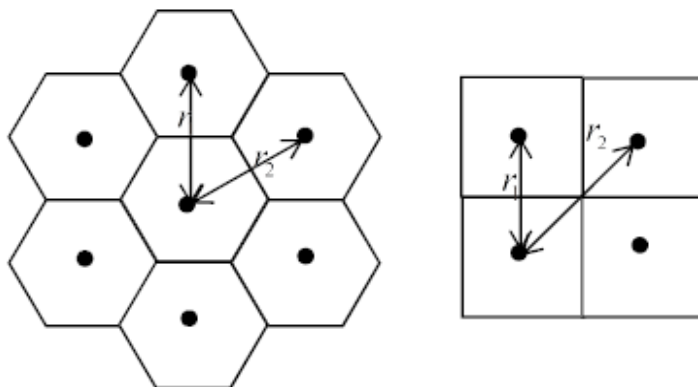
G'olib neyron uchun xatoliklar miqdori kamayadi va qolgan neyronlarning joylashuvi biroz ortib boradi. Sozlashning r_i tezligi parametri 0,001 ga teng. Faol bo'lmagan neyronlarning almashinuvi ortishi kirish qiymatlarining

kengayishini ta'minlaydi va nafaol neyronlar klasterlarini hosil qila boshlaydi. Natijada u yangi kirish vektorlarini jalb qilishni boshlaydi. Bu ikki afzallik beradi. Agar neyron raqobatni yutmasa, uning vazni vektori to'rt kirishiga kiradigan vektorlardan sezilarli darajada farq qiladi, unda uning o'rnini o'rganishda yetarli bo'ladi va u raqobatlashadi. Bunday holatda vazn vektori ma'lum kirish vektorlari guruhiga yaqinlasha boshlaydi. Neyron g'olib kela boshlagach, uning o'rnini bosishi kamayadi. Shunday qilib, "o'lik" neyronlarni faollashtirish vazifasi hal qilinmoqda. O'zgarishlarni sozlash bilan bog'liq bo'lgan ikkinchi afzallik, ular siz faoliyat parametrlari qiymatlarini moslashtirish va taxminan bir xil miqdordagi kirish vektorlarini jalb qilishiga imkon beradi. Shunday qilib, agar klasterlardan biri boshqasidan ko'ra ko'proq kirish vektorini tortadigan bo'lsa, to'ldirilgan maydon qo'shimcha neyronlarni jalb qiladi va kichikroq guruhlarga bo'linadi.

1.3-§. Kohonen xaritalarini qurish prinsiplari

Kohonen xaritalari (*o'z-o'zini tashkil qiluvchi xaritalar* yoki *SOM* — *self-organizing map*) ikki o'lchovli xaritada obyektlarning ko'p o'lchovli xususiyatlarini vizual ko'rsatish uchun mo'ljallangan. Kohonen xaritalari kichik o'lchovli (odatda, ikki o'lchovli) elementlarga yuqori o'lchovli kirish ma'lumotlarini xaritasini ishlab chiqadi. Kohonen xaritalari Kohonen neyron to'rlariga o'xshash. Farq shundaki, xaritada klaster markazlari bo'lgan neyronlar bir necha tuzilishlarga (odatda ikki o'lchovli to'rda) hosil qilinadi. Xaritani o'rganish jarayonida biz nafaqat neyronlar g'olibi, balki qo'shnilarining og'irliklarini ham o'zgartiramiz. Natijada, Kohonen neyron to'ridagi ba'zi metrikalarda yopilgan kirish vektorlari bir neyronga (klasterning markazi) tegishli bo'lib, Kohonen xaritasida ular atrofdagi turli yaqin neyronlarga murojaat qilishlari mumkin. Odatda, neyronlar to'rtburchaklar yoki olti burchakli neyron to'rlarga ega bo'lgan ikki o'lchovli panjada joylashgan. Neyron-qo'shnilar xaritadagi neyronlar orasidagi masofa bilan aniqlanadi (5-rasm). Markazlarida oltiburchak va to'rtburchaklar shaklidagi neyron to'rlar mavjud, ularning

markazlarida neyronlar mavjud. Oltiburchakli neyron to‘rlar xarita ustida obyektlar orasidagi dekart masofasini aniqroq ko‘rsatadi, chunki bu neyron to‘rlar uchun qo‘shni tugun markazlari orasidagi masofa bir xil bo‘ladi [9].



5-rasm. Oltiburchakli va to‘rtburchakli neyron to‘rlar

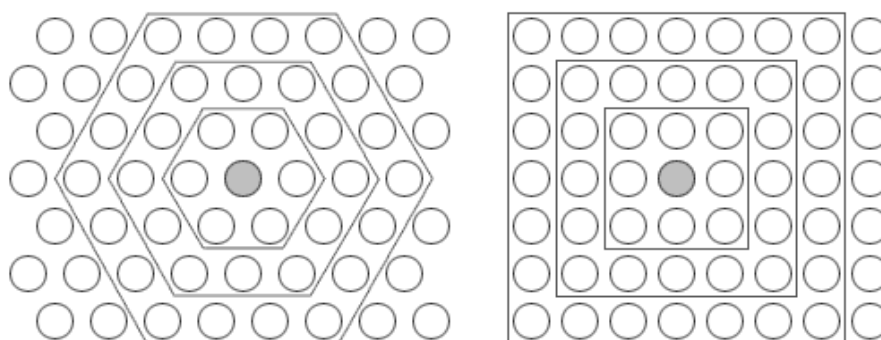
Har bir tugun Kohonen neyron to‘rining neyroniga to‘g‘ri keladi, ya‘ni, Kohonen xaritasida neyron soni xaritaning tugunlari soniga teng va Kohonen neyron to‘rining neyronlarning sonidan ko‘p, klasterlar soniga teng. Xaritaning tugunlar soni qiymatini kerakli parametriga bog‘liq va eksperimental ravishda bosiladi. O‘z-o‘zini tashkil qiluvchi xaritalar neyron to‘r algoritmlarining asosiy turlaridan biri hisoblanadi. Ushbu texnologiyaning boshqa neyron to‘rlardan asosiy farqi, o‘qituvchisiz o‘qitish usulini qo‘llanilgan, teskari tarqatish algoritmi bilan o‘rgatilgan, ya‘ni o‘rganishda natija faqatgina kirish ma‘lumotlarining tuzilishiga bog‘liq. Ushbu turdagi neyron to‘rlar ko‘pincha turli masalarni echish uchun, ma‘lumotlardagi bo‘shliqlarni tiklashdan ma‘lumotlarni tahlil qilish va qonunlarni topish uchun ishlatiladi, masalan, moliyaviy masalalarda [15].

O‘z-o‘zini o‘rganish xaritalarining ishlash algoritmi ko‘p o‘lchovli vektorlarni klaster taxlil variantlaridan biri hisoblanadi. Bunday algoritmlarning misol sifatida “k-eng yaqin qo‘shni” (k-means) algoritmini keltirish mumkin. SOM algoritmining muhim farqi shundaki, unda barcha neyronlar odatda ikki o‘lchovli to‘rni hosil qiladi. SOM da o‘rganish asosan “g‘olib hammasini oladi” qoidasiga ko‘ra amalga oshiriladi. Shu bilan birga, o‘rganish vaqtida

nafaqat neyronlarning g'oliblari, balki qo'shni bo'lganlar ham o'zgarmaydi, biroq u kichik darajada o'lchovli fazoga o'tkaziladi. Shu sababli, SOM o'lchamli fazodan nisbatan kichik o'lchamli fazoga o'tish usullaridan biri sifatida qaralishi mumkin. Ushbu algoritmni qo'llash natijasida, haqiqiy xarita fazosiga o'xshash vektorlar hosil qilinadi [15].

SOM neyronlarning tartiblangan strukturasi qo'llashni nazarda tutadi. Odatda bir va ikki o'lchamli to'rtburchak qo'llaniladi. Bu holda har bir neyron n-o'lchovli vektor ustuni $w = [w_1, w_2, \dots, w_n]^T$ bo'lib, bu yerda n haqiqiy fazo o'lchami (kirish vektorlarining o'lchami) bilan aniqlanadi. Bir va ikki o'lchovli to'rtburchaklardan foydalanish yuqori o'lchamdagi kengaytmali tuzilmalarni xaritada topishda muammolar yuzaga kelishi bilan bog'liq (bu yerda o'lchamni ikki o'lchovli o'lchamga tushirish, vizual ko'rish muammolarga bog'liq).

Odatda, neyronlar to'rtburchaklar yoki olti burchakli hujayralarga ega bo'lgan ikki o'lchovli to'rtburchak tugunlaridir. Bu holda, yuqorida aytib o'tilganidek, neyronlar ham bir-biri bilan ta'sir o'tkazishadi. Ushbu ta'sirning kattaligi xaritadagi neyronlar orasidagi masofa bilan belgilanadi. 6-rasmda oltiburchakli va to'rtburchakli to'rtburchaklar uchun neyronlar orasidagi masofa keltirilgan.



6-rasm. Xaritada olti burchakli (a) va to'rtburchak (b) to'rtburchaklar uchun neyronlar orasidagi masofa

Oltinchi burchakli to'rtburchakdagi neyronlar orasidagi Yevklid masofa to'rtburchakli to'rtburchakga qaraganda ko'proq ekanligini ko'rish oson. Bunday holda, xaritaning yakuniy umumlashtirish qobiliyati to'rtburchakdagi neyronlar soniga qarab aniqlanadi.

1.4-§. Kohonen xaritalar tuzish algoritmlari

Kohonenning o‘z-o‘zini tashkil qiluvchi xaritalarni o‘qitish algoritmi quyidagi bosqichlardan iborat [16]:

1. Har bir tugunning og‘irliklarini boshlang‘ich aniqlash;
2. Berilganlar to‘plamidan tasodifiy vektorlar dastlabki o‘qitish uchun bosiladi va to‘r hosil qilinadi;
3. To‘rning har bir tugunidagi kirish vektoriga o‘xshashligi hisoblanadi. G‘olib chiqqan tugun odatda bir qiymatni o‘zlashtiradi va g‘olib neyron deb ataladi.

$$K^2 = \sum_{i=0}^{i=n} (x_i - w_i)^2 \quad (1)$$

bu yerda x - joriy kirish vektori, w_i - i -tugundagi og‘irlik vektori va n - og‘irliklar soni.

4. G‘olib neyronning radiusi atrofi hisoblanadi. Bu katta ahamiyatga ega. Odatda to‘r radiusi sifatida g‘olib neyronning radiusi belgilanadi va har bir qadamda radius kamayib boradi.

$$\sigma(t) = \sigma_0 e^{(-t/\lambda)} \quad (2)$$

bu yerda t - joriy itoratsiya, λ - o‘zgarish vaqt $\lambda = \frac{N}{\sigma_0}$, σ_0 - xaritaning radiusi, N - takrorlanishlar soni.

5. Barcha tugunlarda g‘olib neyrodagi radius atrofi 4-qadamdagidek hisoblanadi, kirish vektori qiymatlari o‘zgaradi. Neyronlar g‘olib neyronning-tuguniga qanchalik yaqin bo‘lsa, unlarning vazni ham o‘zgaradi.

$$L(t) = L_0(t) e^{(-t/\lambda)},$$

$$w(t+1) = w(t) + \theta(t) L(t) (x(t) - w(t)) \quad (3)$$

bu yerda $L(t)$ – o‘qitish tezligi, $L_0(t)$ – boshlang‘ich o‘qitish tezligi, $\theta(t)$ - g‘olib neyrondan Gauss funktsiyasi bo‘yicha masofasi.

6. N-qatlamgacha 2-bosqichdan boshlab takrorlanadi.

Muntazam ravishda o‘rganish algoritmda har bir o‘rganish bosqichidan keyin og‘irliklar yangilanadi. Neyronlarning o‘rganish, Kohonen neyron to‘ridagi neyronlariga o‘xshaydi. Farq shundaki, g‘olib neyron atrofiga kiruvchi neyronlar yoki g‘olib neyronni o‘rganish radiusi qo‘shimcha ravishda o‘rgatiladi. Neyron xaritada g‘olib chiqqan neyron o‘rtasidagi masofa ma’lum bir qiymatdan pastroq bo‘lsa (neyronlarning og‘irliklari o‘zgaradi, lekin ularning xaritadagi joylashuvi o‘zgarmas), neyron g‘olib neyronning radius atrofiga tegishli. Bunday algoritm WTM tipidagi algoritmda deb ataladi (*Winner Takes Most*- g‘olib ko‘proq oladi). Klassik algoritmda g‘olib neyron va uning atrofidagi barcha neyronlarning vaznlari biroz o‘zgartirilgan Kohonen qoidasiga (2) muvofiq o‘rganishga tobe bo‘ladi. Radius atrofdan tashqaridagi neyronlarning og‘irliklari o‘zgarmaydi. Radius atrofning kattaligi va o‘rganish tezligi koeffitsienti - o‘rganish davrining ortishi bilan qiymati kamayadi [16].

Kohonen qoidalaridagi o‘zgarish shundaki, $\eta_i^{(k)}$ o‘rganish tezligi koeffitsienti ikki qismga bo‘lingan: $\eta_i(d, k)$ qo‘shnilik funktsiyasi va $a(k)$ o‘rganish tezligi funktsiyasi.

Qo‘shnilik funktsiyasi o‘zgarmas sifatida

$$\eta_i^{(k)} = \eta_i(d, k) \cdot a(k) \quad (4)$$

yoki Gaus funktsiyasi sifatida ishlatiladi.

$$\eta_i(d, k) = \begin{cases} const, & d_i \leq \sigma(k), \\ 0, & d_i > \sigma(k) \end{cases} \quad (5)$$

$$\eta_i(d, k) = e^{-\frac{d_i}{2\sigma(k)}}$$

Eng yaxshi natijaga Gauss funktsiyasidan foydalanilganda erishiladi. (4)

va (5) da $d_i = \|r_i - r_{c_j}\|$ i - neyron va c_j g‘olib neyron orasidagi masofa, r_i va r_{c_j} i -

kartasining to'ring tugunning va c_j g'olib bo'lgan neyron koordinatalari, $\|r_i - r_{c_j}\|$ kartasining to'ring i va c_j tugunlar orasidagi masofa. $\sigma(k)$ - funktsiya bu o'rganishning takrorlashi bilan kamayib boruvchi funktsiyasi. Eng odatda o'rganishning takrorlashi ishlatidi, chiziqli ravishda kamayadi.

Keling, $a(k)$ o'rganish tezligi funktsiyasini ko'rib chiqaylik. Bu funktsiya, shuningdek, o'rganishning takrorlashi kamayib boruvchi funktsiyani ifodalaydi. Ushbu funktsiyaning odatda ikkita turi ishlatiladi: chiziqli takrorlanuvchi va teskari proporsional

$$a(k) = \frac{A}{k + B}$$

bu yerda A va B lar o'zgarmaslar. Ushbu funktsiyani qo'llash o'rganishning barcha vektorlari o'rganishga sezilarli darajada ta'sir ko'rsatadi.

O'rganish ikki asosiy bosqichdan iborat: boshlang'iz bosqich vazn vektorlarini koeffitsientlarining aniqlash, bu neyron vektorlarini namunadagi misollarni taqsimlash o'rganish tezligi va o'rganish radiusining yetarlicha katta qiymatda bo'ladi. Oxirgi bosqichda - sozlash bosqichi, o'rganish tezligi parametrlarining dastlabki qiymatidan ancha kichik bo'lgan taqdirda, muvozanat aniqlanadi. To'r xatosi (3) yetarlicha kichik bo'lgunga qadar davom etadi.

1.5-§. O'rganishning paketli algoritmlari

Bundan tashqari algoritmni paketli o'rganish algoritmi mavjud. Ushbu algoritm orasidagi farq kiritishda barcha vektorlarning kiritilishi bo'lib, bu holda faqat og'irliklar yangilanadi. Ushbu algoritm normallashtirilgan kirish vektorlarini ishlatadi. Algoritm bir necha bosqichdan iborat [**Ошибка! Источник ссылки не найден.**]:

1. Barcha N ta vektorlar bir vaqtning o'zida kirishga yuboriladi. Keyin, har bir x_j kirish vektori va w_i vektorlar vazni Yevklid fazosida masofa bo'yicha hisoblanadi. G'olib neyronlar soni aniqlanadi:

$$c_j = \arg \min_i \left\{ \|w_i - x_j\| \right\},$$

2. Barcha vaznlar yangilanadi, chunki barcha vektorlar elementlari o'rtacha qiymatda berilgan:

$$w_i^{new} = \frac{\sum_{j=1}^N h_{c_j,i} x_j}{\sum_{j=1}^N h_{c_j,i}}$$

bu yerda $h_{c_j,i}$, i -qo'shnilik funksiya

$$h_{c_j,i} = \exp \left(\frac{\|r_i - r_{c_j}\|^2}{2\sigma^2(k)} \right)$$

Bu yerda r_i , r_{c_j} xaritasi to'ridagi i -neyron va c_j g'olib bo'lgan neyron koordinatalari bo'lib, $\|r_i - r_{c_j}\|$ - i va c_j qismlar orasidagi masofa, $\sigma(k)$ parametr o'tish sonini ortishi bilan kamayadi.

$$\sigma(k) = \sigma_0 (1 - k / k_{\max})$$

σ_0 - bosiladigan boshlang'ich qiymat, k_{\max} - maksimal o'tish soni. Xatolik funksiyasi (3) yetarli darajada kichik bo'lgunga qadar takrorlanadi.

Xaritani yaratishdan avval uning konfiguratsiyasi va neyronlar sonini aniqlang. Neyronlar soni xaritaning tafsilotini aniqlaydi, ammo neyronlar qancha ko'p bo'lsa, o'raganish shuncha uzoq davom etadi.

Ushbu ko'rsatish usuli yordamida, olingan xarita bir tort bo'lgi sifatida ifodalanishi mumkin. Ushbu qatlamning har bir qatlami asl ma'lumotlarning tarkibiy qismlaridan biri tomonidan yaratilgan rang. Olingan ranglar to'plami ma'lumotlar majmui komponentlari orasidagi mavjud naqshlarni tahlil qilish uchun ishlatilishi mumkin. Xaritani tashkil qilgandan so'ng biz ikki o'lchamli tasvir sifatida ko'rsatilishi mumkin bo'lgan bir qator tugunlarni olamiz. Bu holda xarita har bir tugunni to'rtburchak yoki to'rtburchak shaklidagi bo'linma

bilan bog'lash mumkin, ularning koordinatalari to'rdagi tegishli tugunning koordinatalari bilan belgilanadi. Vizualizatsiya uchun bu rasmning xujayralari rangini aniqlash uchungina qoladi. Buning uchun komponent qiymatlari ishlatiladi. Oddiy variant - bu kulrang o'lchamlardir. Bunday holda, komponentning minimal qiymatiga ega bo'lgan elementlar urilgan yoki hech qanday yozuvlar qayd qilinmagan xarita tugunlariga to'g'ri keladigan hujayralar qora rangda aks ettiriladi va bunday komponentning maksimal qiymatlari bilan yozuvlarni o'z ichiga olgan hujayralar oq hujayralarga mos keladi. Printsipial jihatdan bo'yoq uchun har qanday gradient palletidan foydalanishingiz mumkin.

Olingan ranglar birgalikda komponentlarning joylashishini, ular o'rtasidagi munosabatlarni va komponentlarning turli qiymatlarining nisbiy o'rnini ko'rsatadigan xarritani hosil qiladi.

I Bob bo'yicha xulosa

Tadqiqot ishning birinchi bobida Kohonen neyron to'rlari va Kohonen xaritalari tushunchasi, Kohonen neyron to'rini qurish printsiplari, Kohonen neyron to'rini o'qituvchisiz o'rganish, Kohonen xaritalarini qurish printsiplari, Kohonen xaritalar o'rganish algoritmlari va algoritmni paketli o'rganishlarni o'rgandim.

Bundan tashqari bu usulning boshqa usullardan kamchiligi va ustunliklarini o'rgandim. Kiruvchi qiymatlar miqdoriy va usulda Yevklid fazoda bo'lishi kerakligini, shuningdek alomatlarining qiymatlari normal-
lashtirish usulari o'rgandim.

II BOB. KOHONEN NEYRON TO‘RIDA KLASTERIZATSIYA VA KLASTER TAHLIL

2.1-§. O‘z o‘zini tashkil qiluvchi Kohonen neyron to‘rida klasterizatsiya

Kohonen neyron to‘rlari deganda o‘z-o‘zini tashkil qiluvchi neyron to‘rlarni nazarda tutadi. O‘z-o‘zini tashkil qiluvchi neyron to‘rlari ma’lum umumiy xususiyatlarga ega kirish vektorlarining klasterlarini (guruhlarini) aniqlash imkonini beradi.

Bizga har biri m ta alomatdan iborat $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$, $i = 1, 2, \dots, n$ vektorlar berilgan. Natijada kirish vektorlarini alomatlari bo‘yicha bir-biriga o‘xshash vektorlardan iborat klasterlarga ajratish kerak. Kohonen neyron to‘rida klasterlarga ajratish algoritmi quyidagicha [17].

1-bosqich. Barcha x_{ij} ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$) kirish vektorlari [0..1] oraliqda o‘tkazib (bu natijani tezroq topishga imkon beradi chunki [0..1] oraliqda amallar katta sonlarga nisbatan kamroq kompyuter vaqtini oladi) normalashtiriladi;

2-bosqich. O‘qitish uchun w_{ij} ($i = 1, 2, \dots, s$, $j = 1, 2, \dots, m$) (bu yerda t -klasterlar soni) vazn vektori koeffisientlari [0..1] oraliqdan tasodifiy tanlab olinadi;

3-bosqich. O‘qitish uchun tasodif kirish vektori bosiladi va $k=1$ bo‘ladi;

4-bosqich. Tanlangan kirish vektorining barcha neyronlarining har bir klastergacha bo‘lgan Yevklid masofasi hisoblanadi;

$$d_j = \rho(x, w_j) = \|x - w_j\| = \sqrt{\sum_{i=1}^m (x_i - w_{ij})^2} \quad (6)$$

bu yerda, x_i tanlangan vektorining i - neyron kirishi, w_{ij} - i -neyronning j -klasterga kirishining vazn koeffisienti, m neyronlar (alomatlar) soni, d_j - klastergacha bo‘lgan masofasi;

5-bosqich. d_j lar orasidan minimum masofagi neyron topiladi va g‘olib neyron hisoblanadi;

$$\min_{1 < j < m} d_j = \min_{1 < j < m} \|x - w_j\| = \min_{1 < j < m} \sqrt{\sum_{i=1}^m (x_i - w_{ij})^2} \quad (7)$$

6-bosqich. G'olib neyron topilgandan keyin j -klasterdagi vazn koeffisientlari o'zgartiriladi.

$$w_{ij}^{k+1} = w_{ij}^k + a(k)(x_i - w_{ij}^k) \quad (8)$$

bu yerda, w_{ij}^k - (k)- qatlamdagi vazn koeffisienti, x_i - kirish vektorini i -neyroni, $a(k)$ - o'qitish tezligi koeffisienti $a(k) = a_0 e^{-k/\lambda}$, w_{ij}^{k+1} - ($k+1$)- qatlamdagi vazn koeffisienti;

7-bosqich. Agar $k < N$ bo'lsa 4-bosqichga o'tiladi, aks holda 3-bosqichga o'tiladi;

8-bosqich. Agar kirish vektorlarini hammasi ko'rib chiqilgan bo'lsa algoritm tugaydi [17].

2.2-§. Test masalalar klaster tahlilni o'tkazish

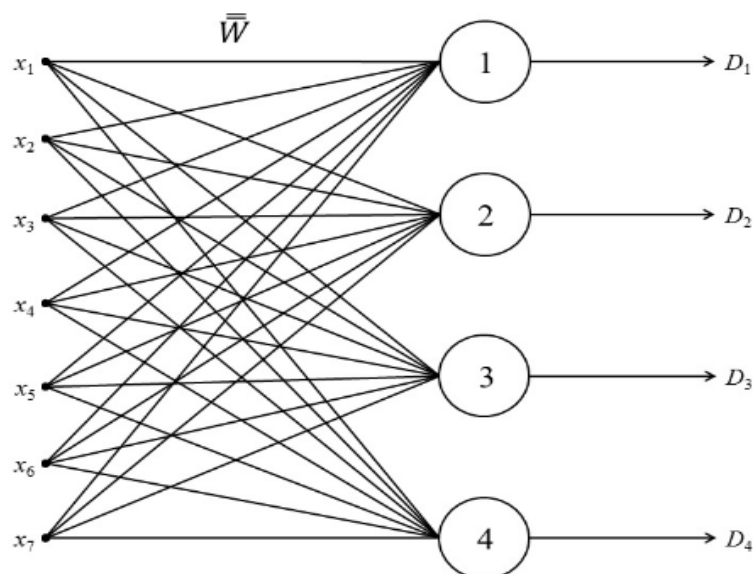
Klasterizatsiya masalasi bir nechta misollar ko'rib chiqamiz.

Bizga 20 ta talabanning imtihondan olgan ballari berilgan (2-jadval). Ularni klasterlarga ajratish kerak va hosil bo'lgan klasterlardagi alomatlarga ko'ra har bir klasterga kirgan talabanning alomatlariga qarab klasterga nom berish kerak.

Berilgan masalasda klasterlar soni 4 ta bo'lishi kerak. Masalni echish uchun $x_1 - x_7$ alomatlar yetarli bo'ladi x_8 alomat esa klaster tahlilida kerak bo'ladi. Neyron to'ring har bir qatlamda 7 ta kiruvchi neyron 4 ta chiqish bo'ladi (7-rasm).

2-jadval. Talabalarning o'rtacha o'zlashtirish tanlanmasi

№	Familiya	Jinsi	Qarzdorligi	Fanlarda olgan o'rtacha ballari					O'rtacha o'zlashtirish
				Mate matika	Informatika	Tarix	Fizika	Adabiyot	
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	Abbosov	E	Yo'q	60	79	60	72	63	1
2	Asqarov	E	Bor	60	61	30	5	17	0
3	Asqarova	A	Bor	60	61	30	66	58	0
4	Bahromov	E	Yo'q	85	78	72	70	85	1,25
5	Baxodirova	A	Yo'q	65	78	60	67	65	1
6	Baxtiyorova	A	Yo'q	60	78	77	81	60	1,25
7	Davronova	A	Yo'q	55	79	56	69	72	0
8	Dadjonov	E	Bor	55	56	50	56	60	0
9	Daliyev	E	Bor	55	60	21	64	50	0
10	Daminov	E	Bor	60	56	30	16	17	0
11	Elyorova	A	Yo'q	85	89	85	92	85	1,75
12	Eldorova	A	Yo'q	60	88	76	66	60	1,25
13	Elmurodov	E	Bor	55	64	0	9	50	0
14	Farhodova	A	Yo'q	80	83	62	72	72	1,25
15	Farhodov	E	Bor	55	10	3	8	50	0
16	Ilhomova	A	Yo'q	60	67	57	64	50	0
17	Ikromov	E	Yo'q	75	98	86	82	85	1,5
18	Lachinova	A	Yo'q	85	85	81	85	72	1,25
19	Mo'minov	E	Yo'q	80	56	50	69	50	0
20	Narmonov	E	Bor	55	60	30	8	60	0



7-rasm. Kohonen neyron to‘rida talabalarni klasterlarga ajratish strukturasi.

Keyin 1-bosqichga binoan barcha kirish vektorini [0..1] oraliqda quyidagicha normallashtiriladi.

- talaba jinsi: 0 – ayol, 1 – erkak;
- qarzdorligi: 0 – bor, 1 – yo‘q;
- fanlardan olgan ballarini: $x_i = x_i / 100$.

Natija 3-jadvalda ifodalangan.

3-jadval. Normallashtirilgan talabalarning o‘rtacha o‘zlashtirish tanlanmasi

№	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1,00	1,00	0,17	0,78	0,70	0,77	0,68
2	1,00	0,00	0,17	0,58	0,35	0,00	0,00
3	0,00	0,00	0,17	0,58	0,35	0,70	0,60
4	1,00	1,00	1,00	0,77	0,84	0,75	1,00
5	0,00	1,00	0,33	0,77	0,70	0,71	0,71
6	0,00	1,00	0,17	0,77	0,90	0,87	0,63
7	0,00	1,00	0,00	0,78	0,65	0,74	0,81
8	1,00	0,00	0,00	0,52	0,58	0,59	0,63

9	1,00	0,00	0,00	0,57	0,24	0,68	0,49
10	1,00	0,00	0,17	0,52	0,35	0,13	0,00
11	0,00	1,00	1,00	0,90	0,99	1,00	1,00
12	0,00	1,00	0,17	0,89	0,88	0,70	0,63
13	1,00	0,00	0,00	0,61	0,00	0,05	0,49
14	0,00	1,00	0,83	0,83	0,72	0,77	0,81
15	1,00	0,00	0,00	0,00	0,03	0,03	0,49
16	0,00	1,00	0,17	0,65	0,66	0,68	0,49
17	1,00	1,00	0,67	1,00	1,00	0,89	1,00
18	0,00	1,00	1,00	0,85	0,94	0,92	0,81
19	1,00	1,00	0,83	0,52	0,58	0,74	0,49
20	1,00	0,00	0,00	0,57	0,35	0,03	0,63

Boshlang‘ich 28 ta vazn koeffisientini [0..1] oraliqdan tasodifiy aniqlab olamiz (3-jadval).

3-jadval. Boshlang‘ich vazn koeffisientlari

№	Vazn koeffisienti w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,20	0,20	0,30	0,40	0,40	0,20	0,50
2	0,20	0,80	0,70	0,80	0,70	0,70	0,80
3	0,80	0,20	0,50	0,50	0,40	0,40	0,40
4	0,80	0,80	0,60	0,70	0,70	0,60	0,70

Takrorlanish soni N qanchalik katta bo‘lsa natija shunchalik aniq bo‘ladi, ammo N juda katta bo‘lib ketsa komyuter vaqtini ko‘p oladi va bu natijani olishni qiyinlashtiradi. Shuning uchun N yetarlicha kattalikda olishga ehtiyot bo‘lish kerak, biz $N=100$ deb olishimiz yetarli.

Tasodifiy kirish vektorini tanlab olamiz. Masalan № 10 olamiz. (6) ga asosan 4 ta klaster markazlarigacha bo‘lgan masofa aniqlanadi. d_j larni

qiymatlari 0,98, 1,65, 0,65 va 1,32 ga teng. (7) ga asosan g'olib neyron $\min_{1 < j < m} d_j = 0,65$ va g'olib 3- neyron . (8) ga asosan vazn koeffisientlari qiymatini o'zgartiramiz va k ni birga oshiramiz. Yangi o'zgartirgan vazn koeffisientlari: $w_{13}=0,86, w_{23}=0,14, w_{33}=0,40, w_{43}=0,51, w_{53}=0,39, w_{63}=0,32, w_{73}=0,28$.

Agar $k < N$ bo'lsa 4-bosqichga o'tamiz. $k=N$ bo'lguncha yuqridagi kabi amallarni qaytadan bajaraveramiz. $k=N$ bo'lsa 3-bosqichga o'tib yangi kirish olamiz $k=1$ deb olamiz. Bu siklni 20 ta kirish vektorini ko'rib chiqilmaguncha davom etadi. Eng oxiridagi vazn koeffisientlari 5-jadvalda.

5-jadval. Natijaviy vazn koeffisientlari

№	Vazn koeffisienti w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,06	0,06	0,21	0,52	0,36	0,55	0,57
2	0,00	1,00	0,50	0,80	0,80	0,80	0,73
3	1,00	0,00	0,04	0,48	0,26	0,22	0,42
4	1,00	0,99	0,69	0,77	0,79	0,78	0,81

Kohonen neyron to'ri tashkil qilishning eng so'ngi bosqichi bu klaster tahlilidir. Barcha to'rtta klasterga tegishli neyronlarni ko'rib chiqamiz. 1-klasterda ikkita № 3 va 16 tanlanmalar kirgan. 2-klasterga № 5,6,7,11,12,14,18 tanlanmalar kirgan. 3-klasterga № 2,8,9,10,13,15,19,20 tanlanmalar kirgan va to'rtinchi tanlanmaga № 1,4,17 tanlanmalar kirgan. So'ngi natijani topish uchun x_8 alomati kerak bo'ladi. Keyin har bir klasterga kirgan tanlanmalarni alomatlarini tahlil qilib klaster haqidagi ma'lumotlarni topamiz. (6-jadval).

6-jadval. Klater tahlil natijasi

№ klaster	Klaster o'Ichami	Jinsi	Qarz-dorligi	O'rta reyting balli	Talabalarning koeffisienti	Izoh
1	2	A	Bor	55	0,00	Biror fanlardan qarz-dorligi mavjud bo'lgan qiz bolalar
2	8	A	Yo'q	79	0,97	Barcha fanlardan qarz-dorligi bo'lmagan qiz bolalar
3	6	E	Bor	40	0,00	Biror fanlardan qarz-dorligi mavjud bo'lgan o'g'il bolalar
4	4	E	Yo'q	73	0,94	Barcha fanlardan qarz-dorligi bo'lmagan o'g'il bolalar

Hosil Kohonen neyron to'ri ixtiyoriy talaba natijalari kiritsan natijani chiqarib beradi. Nafaqat talabarni o'zlashtishi boshqa ko'plab masalarga yechim topishi mumkin. Faqatgi klasterlar soni to'g'ri topilishi tanlanmalarni to'g'ri berilsa yetarli.

Endi yana bitta test masalani ko'rib chiqamiz. Bizga Fisherning irisleri - 150 tanlanmadagi iris berilgan, ular uchta *iris setosa*, *iris virginica* va *iris versicolor* navlarining har birining 50 tanlanmasidan iborat (8-rasm). Tekshirish uchun tanlanmalarni qaysi turga mansub ekanligini kiritmaymiz qolgan 4 ta alomatini kiritamiz (7-jadval). Kirish vektoriga ko'ra programma algoritmi gullarni to'g'ri klasterlarga ajratishi va klaster tahlilni o'tkazadi.



a)



b)



c)

8-rasm. Iris gulining uch xil turi

Har bir tanlanmada iris gulining 4 ta alomatlari berilgan (santimetrda).

1. Gulkosasi uzunligi (*eng.* sepal length);
2. Gulkosasi kengligi (*eng.* sepal width);
3. Gulbargi uzunligi (*eng.* petal length);
4. Gulbargi kengligi (*eng.* petal width).

7-jadval. Iris gulining tanlanmasi

T/r	Gulkosasi uzunligi	Gulkosasi kengligi	Gulbargi uzunligi	Gulbargi kengligi	T/r	Gulkosasi uzunligi	Gulkosasi kengligi	Gulbargi uzunligi	Gulbargi kengligi
1	5,1	3,5	1,4	0,2	76	6,6	3	4,4	1,4
2	4,9	3	1,4	0,2	77	6,8	2,8	4,8	1,4
3	4,7	3,2	1,3	0,2	78	6,7	3	5	1,7
4	4,6	3,1	1,5	0,2	79	6	2,9	4,5	1,5
5	5	3,6	1,4	0,2	80	5,7	2,6	3,5	1
6	5,4	3,9	1,7	0,4	81	5,5	2,4	3,8	1,1
7	4,6	3,4	1,4	0,3	82	5,5	2,4	3,7	1
8	5	3,4	1,5	0,2	83	5,8	2,7	3,9	1,2
9	4,4	2,9	1,4	0,2	84	6	2,7	5,1	1,6
10	4,9	3,1	1,5	0,1	85	5,4	3	4,5	1,5
11	5,4	3,7	1,5	0,2	86	6	3,4	4,5	1,6
12	4,8	3,4	1,6	0,2	87	6,7	3,1	4,7	1,5
13	4,8	3	1,4	0,1	88	6,3	2,3	4,4	1,3
14	4,3	3	1,1	0,1	89	5,6	3	4,1	1,3
15	5,8	4	1,2	0,2	90	5,5	2,5	4	1,3
16	5,7	4,4	1,5	0,4	91	5,5	2,6	4,4	1,2
17	5,4	3,9	1,3	0,4	92	6,1	3	4,6	1,4

18	5,1	3,5	1,4	0,3	93	5,8	2,6	4	1,2
19	5,7	3,8	1,7	0,3	94	5	2,3	3,3	1
20	5,1	3,8	1,5	0,3	95	5,6	2,7	4,2	1,3
21	5,4	3,4	1,7	0,2	96	5,7	3	4,2	1,2
22	5,1	3,7	1,5	0,4	97	5,7	2,9	4,2	1,3
23	4,6	3,6	1	0,2	98	6,2	2,9	4,3	1,3
24	5,1	3,3	1,7	0,5	99	5,1	2,5	3	1,1
25	4,8	3,4	1,9	0,2	100	5,7	2,8	4,1	1,3
26	5	3	1,6	0,2	101	6,3	3,3	6	2,5
27	5	3,4	1,6	0,4	102	5,8	2,7	5,1	1,9
28	5,2	3,5	1,5	0,2	103	7,1	3	5,9	2,1
29	5,2	3,4	1,4	0,2	104	6,3	2,9	5,6	1,8
30	4,7	3,2	1,6	0,2	105	6,5	3	5,8	2,2
31	4,8	3,1	1,6	0,2	106	7,6	3	6,6	2,1
32	5,4	3,4	1,5	0,4	107	4,9	2,5	4,5	1,7
33	5,2	4,1	1,5	0,1	108	7,3	2,9	6,3	1,8
34	5,5	4,2	1,4	0,2	109	6,7	2,5	5,8	1,8
35	4,9	3,1	1,5	0,2	110	7,2	3,6	6,1	2,5
36	5	3,2	1,2	0,2	111	6,5	3,2	5,1	2
37	5,5	3,5	1,3	0,2	112	6,4	2,7	5,3	1,9
38	4,9	3,6	1,4	0,1	113	6,8	3	5,5	2,1
39	4,4	3	1,3	0,2	114	5,7	2,5	5	2
40	5,1	3,4	1,5	0,2	115	5,8	2,8	5,1	2,4
41	5	3,5	1,3	0,3	116	6,4	3,2	5,3	2,3
42	4,5	2,3	1,3	0,3	117	6,5	3	5,5	1,8
43	4,4	3,2	1,3	0,2	118	7,7	3,8	6,7	2,2
44	5	3,5	1,6	0,6	119	7,7	2,6	6,9	2,3
45	5,1	3,8	1,9	0,4	120	6	2,2	5	1,5
46	4,8	3	1,4	0,3	121	6,9	3,2	5,7	2,3
47	5,1	3,8	1,6	0,2	122	5,6	2,8	4,9	2
48	4,6	3,2	1,4	0,2	123	7,7	2,8	6,7	2
49	5,3	3,7	1,5	0,2	124	6,3	2,7	4,9	1,8
50	5	3,3	1,4	0,2	125	6,7	3,3	5,7	2,1
51	7	3,2	4,7	1,4	126	7,2	3,2	6	1,8
52	6,4	3,2	4,5	1,5	127	6,2	2,8	4,8	1,8
53	6,9	3,1	4,9	1,5	128	6,1	3	4,9	1,8
54	5,5	2,3	4	1,3	129	6,4	2,8	5,6	2,1
55	6,5	2,8	4,6	1,5	130	7,2	3	5,8	1,6
56	5,7	2,8	4,5	1,3	131	7,4	2,8	6,1	1,9
57	6,3	3,3	4,7	1,6	132	7,9	3,8	6,4	2
58	4,9	2,4	3,3	1	133	6,4	2,8	5,6	2,2
59	6,6	2,9	4,6	1,3	134	6,3	2,8	5,1	1,5
60	5,2	2,7	3,9	1,4	135	6,1	2,6	5,6	1,4

61	5	2	3,5	1	136	7,7	3	6,1	2,3
62	5,9	3	4,2	1,5	137	6,3	3,4	5,6	2,4
63	6	2,2	4	1	138	6,4	3,1	5,5	1,8
64	6,1	2,9	4,7	1,4	139	6	3	4,8	1,8
65	5,6	2,9	3,6	1,3	140	6,9	3,1	5,4	2,1
66	6,7	3,1	4,4	1,4	141	6,7	3,1	5,6	2,4
67	5,6	3	4,5	1,5	142	6,9	3,1	5,1	2,3
68	5,8	2,7	4,1	1	143	5,8	2,7	5,1	1,9
69	6,2	2,2	4,5	1,5	144	6,8	3,2	5,9	2,3
70	5,6	2,5	3,9	1,1	145	6,7	3,3	5,7	2,5
71	5,9	3,2	4,8	1,8	146	6,7	3	5,2	2,3
72	6,1	2,8	4	1,3	147	6,3	2,5	5	1,9
73	6,3	2,5	4,9	1,5	148	6,5	3	5,2	2
74	6,1	2,8	4,7	1,2	149	6,2	3,4	5,4	2,3
75	6,4	2,9	4,3	1,3	150	5,9	3	5,1	1,8

8-jadval. Iris gulining normallashtgan tanlanmasi

T/r	Gulkosasi uzunligi	Gulkosasi kengligi	Gulbargi uzunligi	Gulbargi kengligi	T/r	Gulkosasi uzunligi	Gulkosasi kengligi	Gulbargi uzunligi	Gulbargi kengligi
1	0,51	0,35	0,14	0,02	76	0,66	3	0,44	0,14
2	0,49	0,3	0,14	0,02	77	0,68	0,28	0,48	0,14
3	0,47	0,32	0,13	0,02	78	0,67	0,3	0,5	0,17
4	0,46	0,31	0,15	0,02	79	0,6	0,29	0,45	0,15
5	0,5	0,36	0,14	0,02	80	0,57	0,26	0,35	1
6	0,54	0,39	0,17	0,04	81	0,55	0,24	0,38	0,11
7	0,46	0,34	0,14	0,03	82	0,55	0,24	0,37	1
8	0,5	0,34	0,15	0,02	83	0,58	0,27	0,39	0,12
9	0,44	0,29	0,14	0,02	84	0,6	0,27	0,51	0,16
10	0,49	0,31	0,15	0,01	85	0,54	3	0,45	0,15
11	0,54	0,37	0,15	0,02	86	0,6	0,34	0,45	0,16
12	0,48	0,34	0,16	0,02	87	0,67	0,31	0,47	0,15
13	0,48	0,3	0,14	0,01	88	0,63	0,23	0,44	0,13
14	0,43	0,3	0,11	0,01	89	0,56	0,3	0,41	0,13
15	0,58	0,4	0,12	0,02	90	0,55	0,25	0,4	0,13
16	0,57	0,44	0,15	0,04	91	0,55	0,26	0,44	0,12
17	0,54	0,39	0,13	0,04	92	0,61	0,3	0,46	0,14
18	0,51	0,35	0,14	0,03	93	0,58	0,26	0,4	0,12
19	0,57	0,38	0,17	0,03	94	0,5	0,23	0,33	0,1
20	0,51	0,38	0,15	0,03	95	0,56	0,27	0,42	0,13

21	0,54	0,34	0,17	0,02	96	0,57	0,3	0,42	0,12
22	0,51	0,37	0,15	0,04	97	0,57	0,29	0,42	0,13
23	0,46	0,36	0,1	0,02	98	0,62	0,29	0,43	0,13
24	0,51	0,33	0,17	0,05	99	0,51	0,25	0,3	0,11
25	0,48	0,34	0,19	0,02	100	0,57	0,28	0,41	0,13
26	0,5	0,3	0,16	0,02	101	0,63	0,33	0,6	0,25
27	0,5	0,34	0,16	0,04	102	0,58	0,27	0,51	0,19
28	0,52	0,35	0,15	0,02	103	0,71	0,3	0,59	0,21
29	0,52	0,34	0,14	0,02	104	0,63	0,29	0,56	0,18
30	0,47	0,32	0,16	0,02	105	0,65	0,3	0,58	0,22
31	0,48	0,31	0,16	0,02	106	0,76	0,3	0,66	0,21
32	0,54	0,34	0,15	0,04	107	0,49	0,25	0,45	0,17
33	0,52	0,41	0,15	0,01	108	0,73	0,29	0,63	0,18
34	0,55	0,42	0,14	0,02	109	0,67	0,25	0,58	0,18
35	0,49	0,31	0,15	0,02	110	0,72	0,36	0,61	0,25
36	0,5	0,32	0,12	0,02	111	0,65	0,32	0,51	0,2
37	0,55	0,35	0,13	0,02	112	0,64	0,27	0,53	0,19
38	0,49	0,36	0,14	0,01	113	0,68	0,3	0,55	0,21
39	0,44	3	0,13	0,02	114	0,57	0,25	0,5	0,2
40	0,51	0,34	0,15	0,02	115	0,58	0,28	0,51	0,24
41	0,5	0,35	0,13	0,03	116	0,64	0,32	0,53	0,23
42	0,45	0,23	0,13	0,03	117	0,65	0,3	0,55	0,18
43	0,44	0,32	0,13	0,02	118	0,77	0,38	0,67	0,22
44	0,5	0,35	0,16	0,06	119	0,77	0,26	0,69	0,23
45	0,51	0,38	0,19	0,04	120	0,6	0,22	0,5	0,15
46	0,48	0,3	0,14	0,03	121	0,69	0,32	0,57	0,23
47	0,51	0,38	0,16	0,02	122	0,56	0,28	0,49	0,2
48	0,46	0,32	0,14	0,02	123	0,77	0,28	0,67	0,2
49	0,53	0,37	0,15	0,02	124	0,63	0,27	0,49	0,18
50	0,5	0,33	0,14	0,02	125	0,67	0,33	0,57	0,21
51	0,7	0,32	0,47	0,14	126	0,72	0,32	0,6	0,18
52	0,64	0,32	0,45	0,15	127	0,62	0,28	0,48	0,18
53	0,69	0,31	0,49	0,15	128	0,61	0,3	0,49	0,18
54	0,55	0,23	0,4	0,13	129	0,64	0,28	0,56	0,21
55	0,65	0,28	0,46	0,15	130	0,72	0,3	0,58	0,16
56	0,57	0,28	0,45	0,13	131	0,74	0,28	0,61	0,19
57	0,63	0,33	0,47	0,16	132	0,79	0,38	0,64	0,2
58	0,49	0,24	0,33	0,1	133	0,64	0,28	0,56	0,22
59	0,66	0,29	0,46	0,13	134	0,63	0,28	0,51	0,15
60	0,52	0,27	0,39	0,14	135	0,61	0,26	0,56	0,14
61	0,5	0,2	0,35	0,1	136	0,77	0,3	0,61	0,23
62	0,59	0,3	0,42	0,15	137	0,63	0,34	0,56	0,24
63	0,6	0,22	0,4	0,1	138	0,64	0,31	0,55	0,18

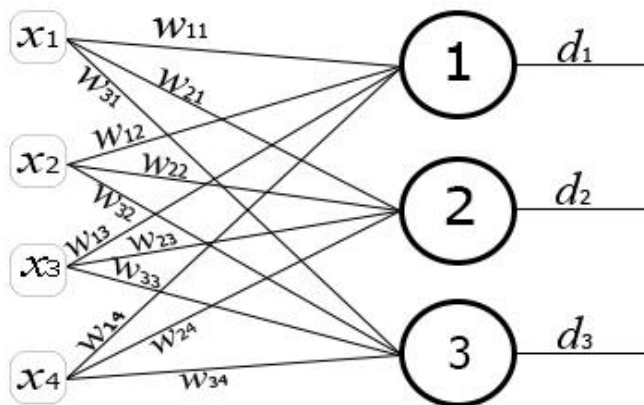
64	0,61	0,29	0,47	0,14	139	6	0,3	0,48	0,18
65	0,56	0,29	0,36	0,13	140	0,69	0,31	0,54	0,21
66	0,67	0,31	0,44	0,14	141	0,67	0,31	0,56	0,24
67	0,56	0,3	0,45	0,15	142	0,69	0,31	0,51	0,23
68	0,58	0,27	0,41	0,1	143	0,58	0,27	0,51	0,19
69	0,62	0,22	0,45	0,15	144	0,68	0,32	0,59	0,23
70	0,56	0,25	0,39	0,11	145	0,67	0,33	0,57	0,25
71	0,59	0,32	0,48	0,18	146	0,67	0,3	0,52	0,23
72	0,61	0,28	0,4	0,13	147	0,63	0,25	0,5	0,19
73	0,63	0,25	0,49	0,15	148	0,65	0,3	0,52	0,2
74	0,61	0,28	0,47	0,12	149	0,62	0,34	0,54	0,23
75	0,64	0,29	0,43	0,13	150	0,59	0,3	0,51	0,18

Berilgan masalada klasterlar soni 3 ta bo‘lishi kerak. Neyron to‘rining har bir qatlamda 4 ta kiruvchi neyron 3 ta chiqish bo‘ladi (9-rasm).

Keyin 1-bosqichga binoan barcha kirish vektorini [0..1] oraliqda quyidagicha normallashtiramiz.

$$- x_i = x_i / 10.$$

Natija 8-jadvalda ifodalangan.



9-rasm. Kohonen neyron to‘rida Iris gulini klasterlarga ajratish strukturasi

Boshlang‘ich vazn koeffitsientlarini [0..1] oraliqdan ixtiyoriy tanlab olamiz (9-jadval).

9-jadval. Boshlang'ich vazn koeffisientlari

№	Vazn koeffisienti w_{ij}			
	w_{1j}	w_{2j}	w_{3j}	w_{4j}
1	0,6	0,3	0,1	0,04
2	0,51	0,22	0,3	0,16
3	0,67	0,34	0,5	0,27

Takrorlanish soni N qanchalik katta bo'lsa natija shunchalik aniq bo'ladi, ammo N juda katta bo'lib ketsa komyuter vaqtini ko'p oladi va bu natijani olishni qiyinlashtiradi. Shuning uchun N yetarlicha kattalikda olishga ehtiyot bo'lish kerak, biz $N=100$ deb olishimiz yetarli.

Tasodifiy kirish vektorini tanlab olamiz. Masalan № 10 olamiz. (6) ga asosan 4 ta klaster markazlarigacha bo'lgan masofa aniqlanadi. d_j larni qiymatlari 0,88, 1,75, 0,05 va 1,92 ga teng. (7) ga asosan g'olib neyron $\min_{1 < j < m} d_j = 0,05$ va g'olib 3- neyron. (8) ga asosan vazn koeffisientlari qiymatini o'zgartiramiz va k ni birga oshiramiz. Yangi o'zgargan vazn koeffisientlari: $w_{13}=0,66$, $w_{23}=0,64$, $w_{33}=0,20$, $w_{43}=0,21$.

Agar $k < N$ bo'lsa 4-bosqichga o'tamiz. $k=N$ bo'lguncha yuqridagi kabi amallarni qaytadan bajaraveramiz. $k=N$ bo'lsa 3-bosqichga o'tib yangi kirish olamiz $k=1$ deb olamiz. Bu siklni 150 ta kirish vektorini ko'rib chiqilmaguncha davom etadi.

Eng oxiridagi vazn koeffisientlari 10-jadvalda.

10-jadval. Natijaviy vazn koeffisientlari

№	Vazn koeffisienti w_{ij}			
	w_{1j}	w_{2j}	w_{3j}	w_{4j}
1	0,03	0,05	0,14	0,84
2	0,051	0,84	0,03	0,76
3	0,97	0,44	0,58	0,07

Kohonen neyron to‘ri tashkil qilishning eng so‘ngi bosqichi bu klaster tahlilidir. Barcha uchta klasterga tegishli neyronlarni ko‘rib chiqamiz. 1-klasterda ikkita № 1-50 tanlanmalar kirgan, 2-klasterga № 51-100 tanlanmalar kirgan va 3-klasterga № 101-150 tanlanmalar kirgan. Keyin har bir klasterga kirgan tanlanmalarni alomatlarini tahlil qilib klaster haqidagi ma‘lumotlarni topamiz (11-jadval).

11-jadval. Klaster tahlil natijasi

№	Klaster	Klaster o‘lchami	O‘rtacha gulkosa uzunligi	O‘rtacha gulkosa kengligi	O‘rtacha gulbargi uzunligi	O‘rtacha gulbargi kengligi	Izoh
1	50	5,006	3,428	1,462	0,246	Gulkosasi kalta va keng, barglari kalta va ingichka	
2	50	5,936	2,77	4,26	1,326	Gulkosasi uzunligi o‘rtacha va ingichka, barglari o‘rtacha va o‘rtacha ingichka	
3	50	6,588	3,03	5,552	2,026	Gulkosasi uzun va keng, barglari uzun va keng	

Ko‘rib chiqilgan ikkita misolda ham Kohonen neyron to‘ri tanlanmalarni to‘g‘ri klasterlarga ajratdi va klaster tahlilini to‘g‘ri amalga oshirdi.

II BOB bo'yicha xulosa

Tadqiqot ishining ikkinchi bobida klasterlarga ajratish metod va usullari o'rgandim. Klasterlarga ajratib bo'lgandan so'ng natija qanchalik to'g'riligini tekshirish uchun klaster tahlil o'tkazish usullari ko'rib chiqdim. "O'z o'zini tashkil qiluvchi kohonen neyron to'rida klasterizatsiya usuli"ni "Talabalarning o'rtacha o'zlashtirishi" va "Iris guli" talanmalarida tadbiq qildim. Tanlangan eksprement tanlanmalarda dastur yaxshi natija oldim.

III BOB. KOHONEN XARITASI USULIDA INFORMATSION MODELLARINI YARATISH VA TEST MASALALARI

3.1-§. O‘z-o‘zini tashkil qiluvchi Kohonen xaritasining informatsion modelini dasturiy taminoti

O‘z-o‘zini tashkil qiluvchi Kohonen xaritalari (*SOM self-organizing map*) ikki o‘lchovli xaritada obyektlarning ko‘p o‘lchovli xususiyatlarini vizual ko‘rsatish uchun mo‘ljallangan. Kohonen xaritalari kichik o‘lchovli (odatda, ikki o‘lchovli) elementlarga yuqori o‘lchovli kirish ma‘lumotlarini xaritasini ishlab chiqaradi. Kohonen neyron to‘rlarida o‘qituvchisiz buyruq ishlatiladi. Kohonen xaritalari Kohonen neyron to‘rlariga o‘xshash. Farq shundaki, xaritada klaster markazlari bo‘lgan neyronlar bir necha tuzilishlarga (odatda ikki o‘lchovli to‘rda) hosil qilinadi. Xaritani o‘z-o‘zini tashkil qilish jarayonida biz nafaqat neyronlar g‘olibi, balki qo‘shnilarining og‘irliklarini ham o‘zgartiradi. Odatda, neyronlar to‘rtburchaklar yoki olti burchakli neyron to‘rlarga ega bo‘lgan ikki o‘lchovli panjada joylashgan. Neyron-qo‘shnilar xaritadagi neyronlar orasidagi masofa bilan aniqlanadi. Markazlarida oltiburchak va to‘rtburchaklar shaklidagi neyron to‘rlar mavjud, ularning markazlarida neyronlar mavjud. Oltiburchakli neyron to‘rlar xarita ustida obyektlar orasidagi dekart masofasini aniqroq ko‘rsatadi, chunki bu neyron to‘rlar uchun qo‘shni tugun markazlari orasidagi masofa bir xil bo‘ladi.

Progaramma taminotini yaratish uchun masalaga mohiyatiga mos berilganlar tuzilmasini tanlash muhim ahamiyatga ega. Chunki mos berilganlar tuzilmasini aniqlash programmani muvaffiaqiyatli yaratish garovidir. Kohonen neyron to‘rlari yordamida asosan miqdoriy alomatlar bilan berilgan obyektlardan foydalaniladi.

O‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion modellarni yaratish uchun birinchi navbatda tanlanmalarni alomatlarga ko‘ra klasterlarga ajratish kerak. Klasterga ajratayorganda tanlanmalar har bir klasterga kirayotganda ma‘lum alomatlari o‘xshashligiga farqlanadi. Tanlanmalarni klasterlarga ajratgandan so‘ng klaster tahlil o‘tkaziladi. Buning

sababi shundaki klasterlarni o'ziga hos xususiyatlarini topiladi. O'z-o'zini tashkil qiluvchi Kohonen xaritasi usulning boshqa usullarga nisbatan qulayligi, bu usulda katta (uch va undan yuqori) o'lchovli fazodagi tanlanmalarni nisbatan kichik o'lchovli fazoga keltiriladi. Natija klaster tahlilga mos ikki o'lchovli xaritada visual ko'rinishda bo'ladi. Bu usuldan foydalanilganda katta o'lchovli tanlanmalar olinganda ham visual ko'rish orqali natijani aniqligiga ishonch hosil qilish oson va tushinarli bo'ladi.

Tadqiqod ishida "O'z-o'zini tashkil qiluvchi Kohonen xaritasi usuli asosida informatsion modellarni yaratish" uchun "Borland C++ Builder6" programmalash muhitini tanladim. Chunki C++ tili masalani echish uchun barcha qulayliklar mavjud. Natijani grafik xaritasini tuzish juda qulaydir.

Dasturiy taminotni yaratish uchun C++ tilidagi ba'zi komponentalar kerak bo'ladi. Bular *Form*, *StringGrid*, *Edit*, *Label*, *Button*, *Image*, *OpenDialog* va boshqalar.

3.2-§. Dasturiy ta'minotning modular tasnifi va tuzilish strukturasi

Dasturiy ta'minoti bir-biri bilan bog'liq bo'lgan 6 ta moduldan iborat. Bular quyidagilar:

1. Qattiq diskdagi tanlanma yozilgan text faylni o'qish va o'qilgan ma'lumotlar bo'yicha natijalarni ekranga chiqaruvchi modul;
2. Boshlang'ich vazn koeffisientlarini aniqlovchi modul;
3. Klasterizatsiya qilib natijaviy vazv koeffisientlarini topuvchi modul;
4. Klaster tahlil qiluvchi modul;
5. Klasterlarni vizual ko'rsatuvchi modul;
6. Kohonen xaritasini hosil qilib vizuallashtiruvchi modul.

Fayldan o'qish moduli qattiq diskdagi tanlanma yozilgan text faylni o'qish va o'qilgan ma'lumotlar bo'yicha natijalarni ekranga jadval ko'rinishida chiqaradi.

Boshlang'ich vazn koeffisientlarini aniqlovchi modulda vazn koeffisientlari $[0..1]$ oraliqdan tasodifiy aniqlanadi.

Klasterizatsiya qilib natijaviy vazn koeffisientlarini topuvchi modulda boslang'ich vazn koeffisientlari asosida har bir neyron uchun "g'olib barchasini oladi" metodi orqali g'olib neyronlar topiladi va vazn koeffisientlari o'zgartiriladi. Barcha neyronlar g'oliblari topilgandan so'ng natijaviy vazn koeffisientlari topiladi va ekranga jadval ko'rinishida chiqariladi.

Klaster tahlil qiluvchi modulda klasterizatsiya jarayoni tugagandan so'ng har bir klasterga tegishli tanlanmalar soni aniqlanadi va barcha alomatlarni tahlil qilinadi va natijalarni faylga saqlanadi.

Klasterlarni vizual ko'rsatuvchi modulda klasterdagi tanlanmalarning alomatlarini o'zaro kombinatsiyasi va klasterizatsiya ikki o'lchovli koordinatalar sistemasida vizual aklantiriladi.

Kohonen xaritasini hosil qilib vizuallashtiruvchi modulda klasterdagi tanlanmalarning alomatlarini o'zaro kombinatsiyasi va klasterizatsiya ikki o'lchovli xaritada vizual aklantiriladi. Bunda har bir klaster uchun alohida rangni formadagi ColorBoxdan tashlash mumkin.

O'z-o'zini tashkil qiluvchi Kohonen xaritasini hosil qiluvchi class Kohonen_map sinfini yaratish uchun quyidagi asosiy o'zgaruvchilar ishlatiladi:

n- tanlanmalar soni;

m- alomatlar soni;

k- sinflar soni;

N- itoratsiya soni;

n1- elementlarga bo'linishlar soni;

X[1..n, 1..m]- kiruvchi tanlanmalardan tashkil topgan massiv;

w1[1..k, 1..m]- boslang'ich vazn koeffisientlari massivi;

w[1..k, 1..m]- natijaviy vazn koeffisientlari massivi;

K[1..n]- har bir tanlanmaning tegishli bo'lgan klaster raqamlaridan iborat bo'lgan massiv.

O'z-o'zini tashkil qiluvchi Kohonen xaritasini hosil qiluvchi class Kohonen_map sinfini yaratish uchun quyidagi asosiy funksiyalar ishlatiladi:

void set_X_vector(String fn)- ko'rsatilgan fayldan barcha tanlanmalarni o'qib olish funksiya;

void set_weight1()- boshlang'ich vazn koeffisientlarini [0..1] oraliqdan tasodifiy aniqlash funksiyasi;

void Set_Winner_neyron(int x_i)- x_i -g'olib neyronni topish funksiyasi.

void Klastrizatsiya()- klasterizatsiya qiluvchi funksiya;

void set_weight()- natijaviy vazn koeffisientlarini topuvchi funksiya;

void Klaster_tahlil()- klaster tahlilni amalga oshiruvchi funksiya;

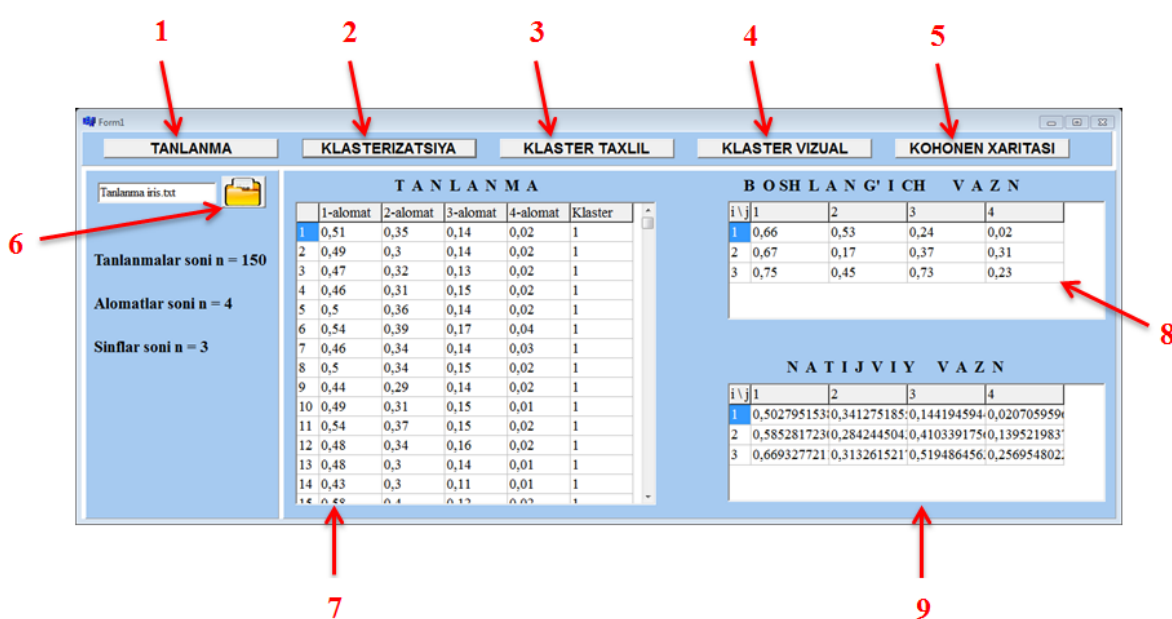
void Klaster_tahlil_file()- klaster tahlil natijalarini qattiq diskda matn fayli ko'rinishda saqlovchi funksiya;

void koordinata_oqi()- koordinata o'qlarini chizuvchi funksiya;

void Klaster_chiz()-klaterdagi tanlanmalarning alomatlarini o'zaro kambinatsiyasi va klaterizatsiya ikki o'lchovli koordinatalar sistemasida vizual aklantiriladigan funksiya;

void Kohonen_xaritasi()- klaterdagi tanlanmalarning alomatlarini o'zaro kambinatsiyasi va klaterizatsiya ikki o'lchovli xaritada vizual aklantiriladigan funksiya.

Dasturiy ta'minot tasnifi.



10-rasm. Dasturiy ta'minot tasnifi

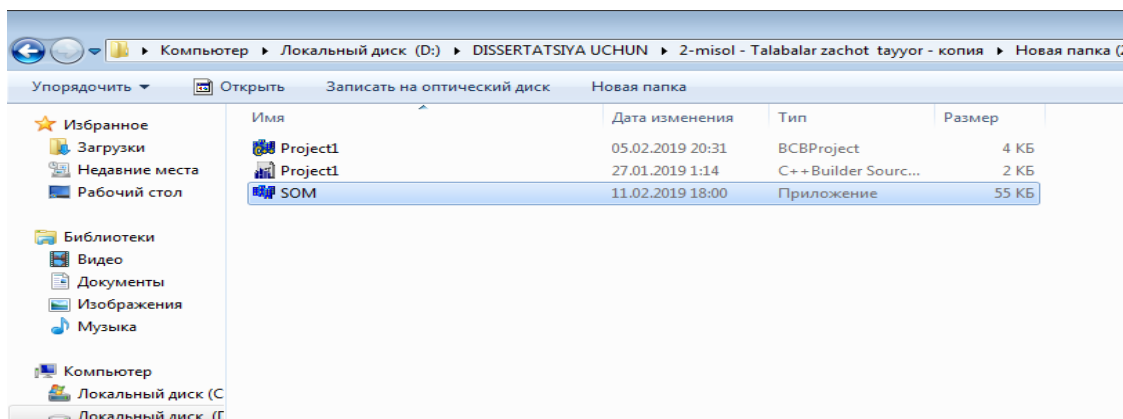
Bu yerda:

- 1- tanlanmalani o‘qish moduli;
- 2- klasterizatsiya qilish moduli;
- 3- klaster tahlil moduli;
- 4- klaterdagi tanlanmalarning alomatlarini o‘zaro kambinatsiyasi va klaterizatsiya ikki o‘lchovli koordinatalar sistemasida vizual aklantiriltiruvchi modul;
- 5- klaterdagi tanlanmalarning alomatlarini o‘zaro kambinatsiyasi va klaterizatsiya ikki o‘lchovli xaritada vizual aklantiriltiruvchi modul;
- 6- tanlanmani qattiq diskdan o‘qish tugmasi;
- 7- tanlanma keltirilgan jadval;
- 8- boshlang‘ich vazn koeffisientlari;
- 9- natijaviy vazn koeffisientlari.

3.3-§. O‘z-o‘zini tashkil qiluvchi Kohonen xaritasining informatsion modelga yo‘riqima

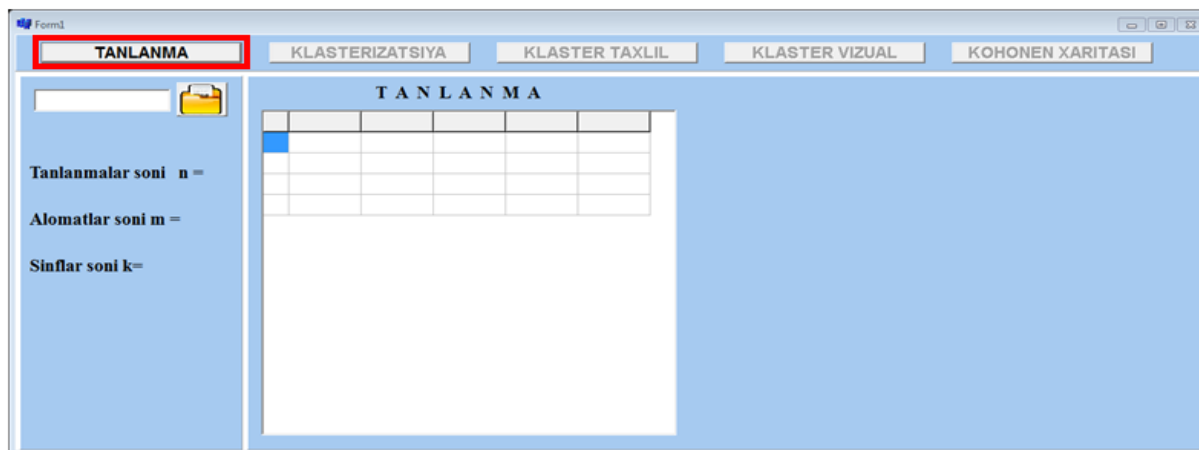
“O‘z-o‘zini tashkil qiluvchi Kohonen xaritasi usulining informatsion modeli” ishlashini Iris gulining tanlanma misolida ko‘rib chiqamiz.

Birinchi bo‘lib dastur ishda tushiriladi. Buning uchun “SOM” nomli dasturning *exe* ilovasini bosiladi va sichqonchanning o‘ng tugmasi ikki marta bosiladi (11-rasm) .

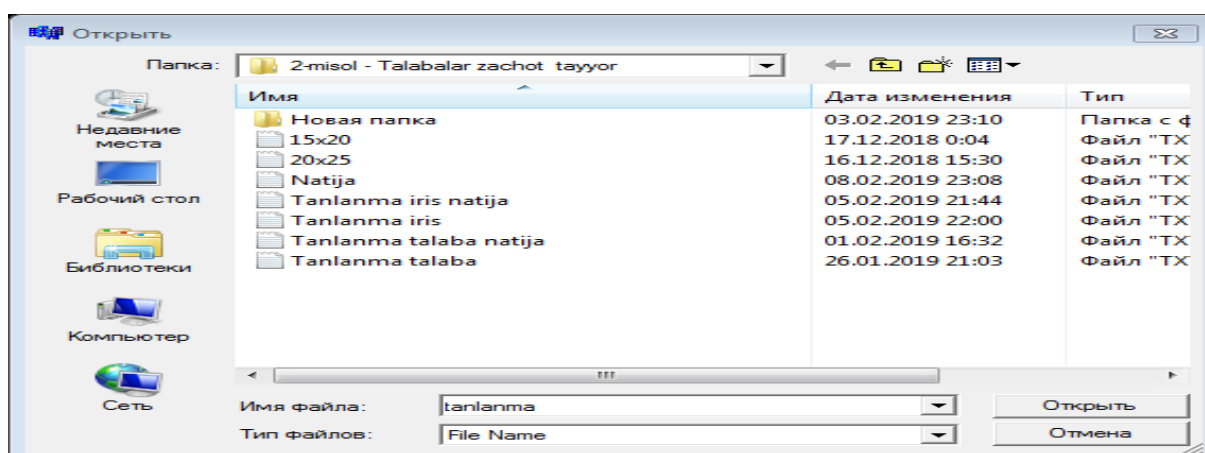


11-rasm. “SOM” dasturini ishga tushirish

Dastur ishga tushgandan keyin “TANLANMA” tugmasini bosiladi (12-rasm). Natijada kerakli tanlanmani tanlash uchun muloqot oynasi ochiladi (13-rasm). Bu yerdan Iris gulining tanlanmalari joylashgan “Tanlanma iris.txt” matn fayli bosiladi. Natijada tanlanma ekranda StingGridda ko‘rsatiladi (14-rasm).



12-rasm. “SOM” dasturining dastlabki ko‘rinishi

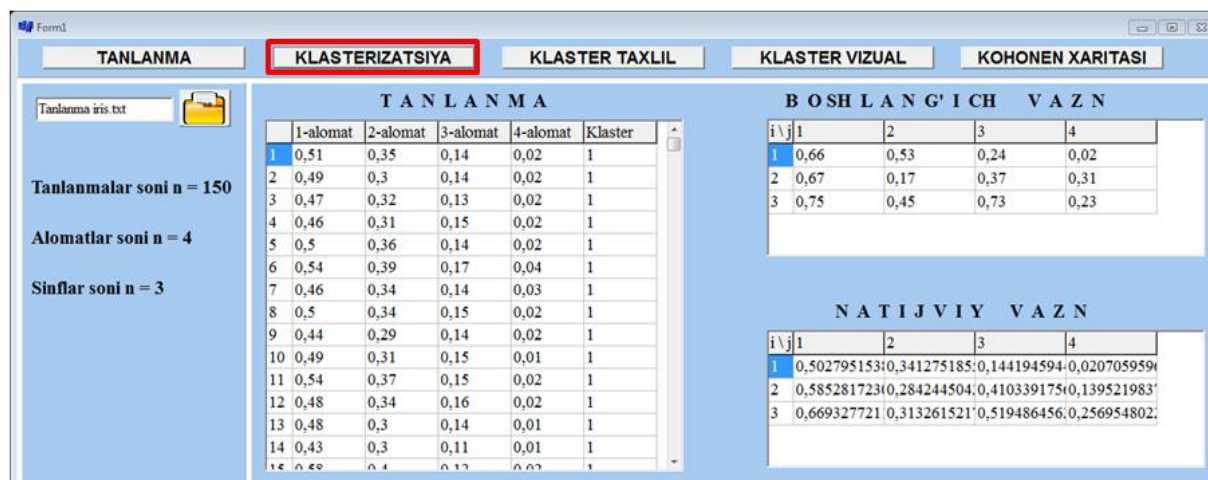


13-rasm. Tanlanmani tanlash uchun ochiladigan muloqot oynasi



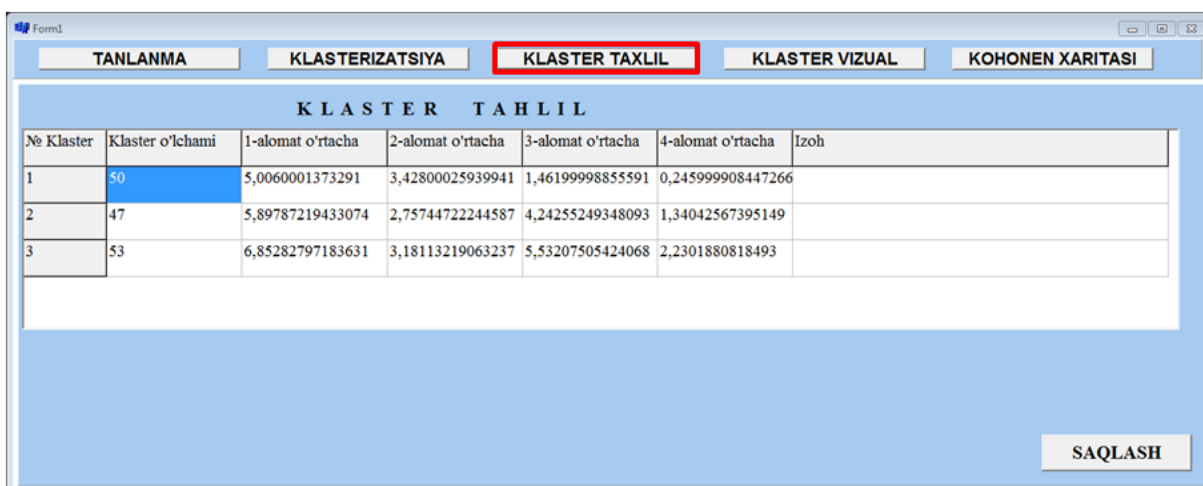
14-rasm. Tanlanmani dasturga yuklangan holati

Boshlang'ich vazn koeffisientlarini [0..1] oraliqda taqriban aniqlab. Boshlag'ich vazn koeffisientlari asosida kasterizatsiya qilish uchun "KLASTERIZATSIYA" tugmasi bosiladi. Natijada tanlanma klasterizatsiya qilinadi va har bir tanlanmaning klaster raqami tanlanma alohida ustunda chiqariladi. Oxirgi natijaviy vazn koeffisientlari StringGirdga chiqariladi.



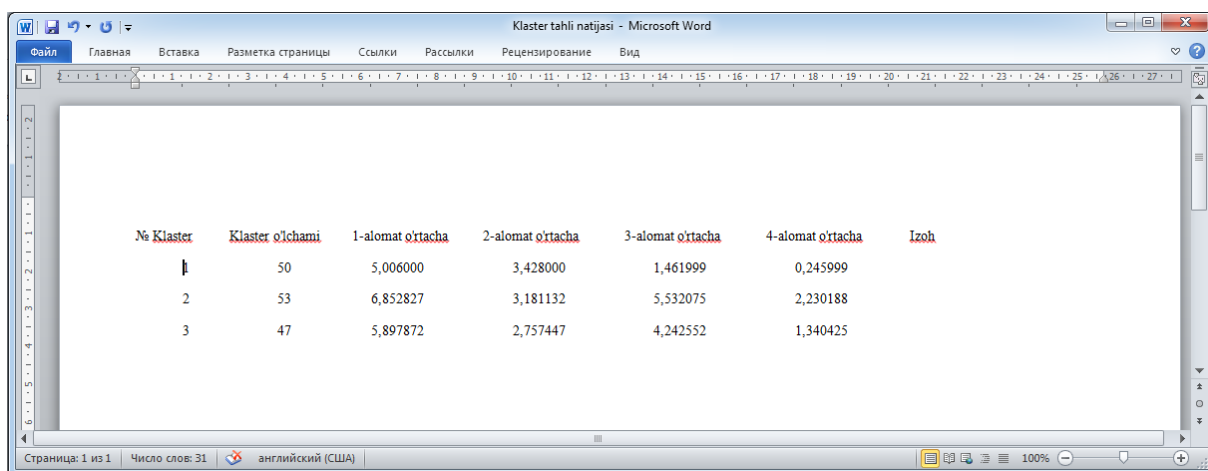
15-rasm. Tanlanmani klasterizatsiyalash natijasi

Tanlanmani klasterlarga ajratib bo'lgandan so'ng. Klaster tahlilini o'tkazishimiz mumkin. Buning uchun "KLASTER TAHLIL" tugmasi bosiladi. Natija 16-rasmda keltirilgan.



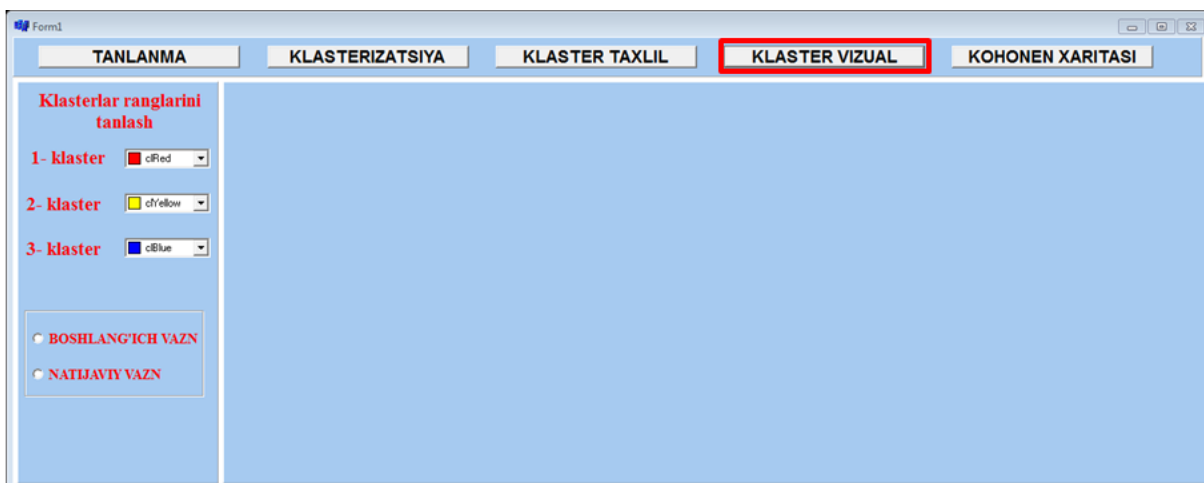
16-rasm. Klaster tahlil natijasi.

Jadvaldagi “Izoh” ustuniga har bir klasterning alohida xususiyatlarini yozib qo‘yishimiz va “SAQLASH” tugmasini bosish orqali barcha ma’lumotlarni faylga saqlab qo‘yishimiz mumkin. 17-rasmda klaster tahlil natijalari saqlangan fayl keltirilgan.



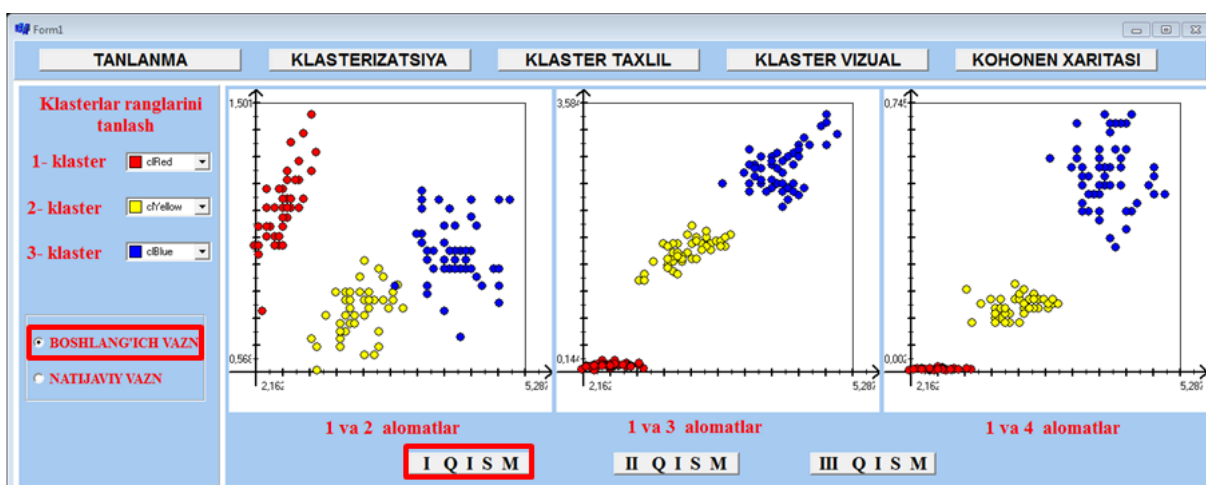
17-rasm. Klaster tahlil natijasi saqlangan fayl

Klasterizatsiya natijalarini vizual ko‘rish uchun “KLASTER VIZUAL” tugmasi bosiladi (18-rasm).



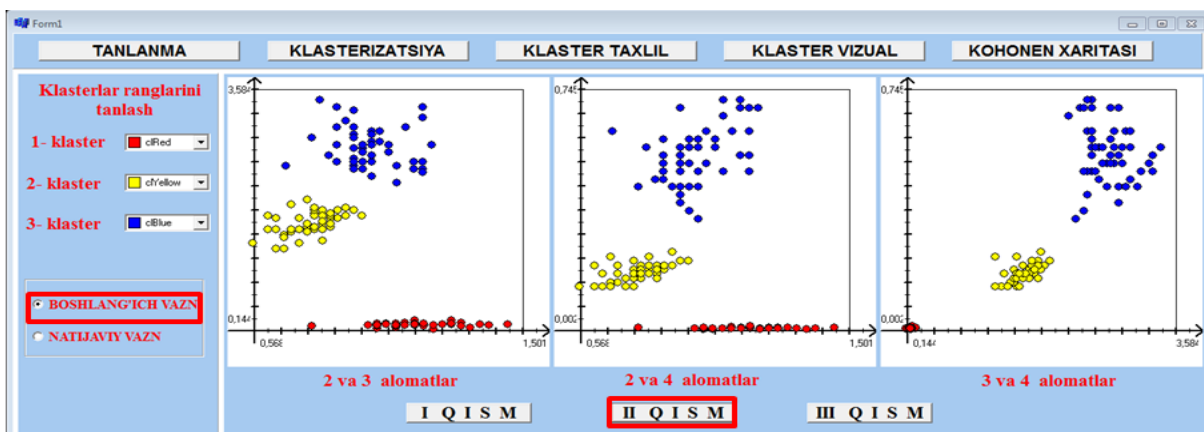
18-rasm. “KLASTER VIZUAL” oynasini daslabki ko‘rinishi

“KLASTER VIZUAL” oynasidan birinchi bo‘lib har bir klasterni rangini ColorBox dan tanlanadi, keyin RadioGroup dan “BOSHLANG‘ICH VAZN” yoki “NATIJAVIY VAZN” lardan biri tanlanadi. Agar “BOSHLANG‘ICH VAZN” tanlansa berilgan tanlanmani boshlang‘ich vazn koeffisienlariga ko‘ra klasterlar vizual ko‘rsatiladi (19-rasm).

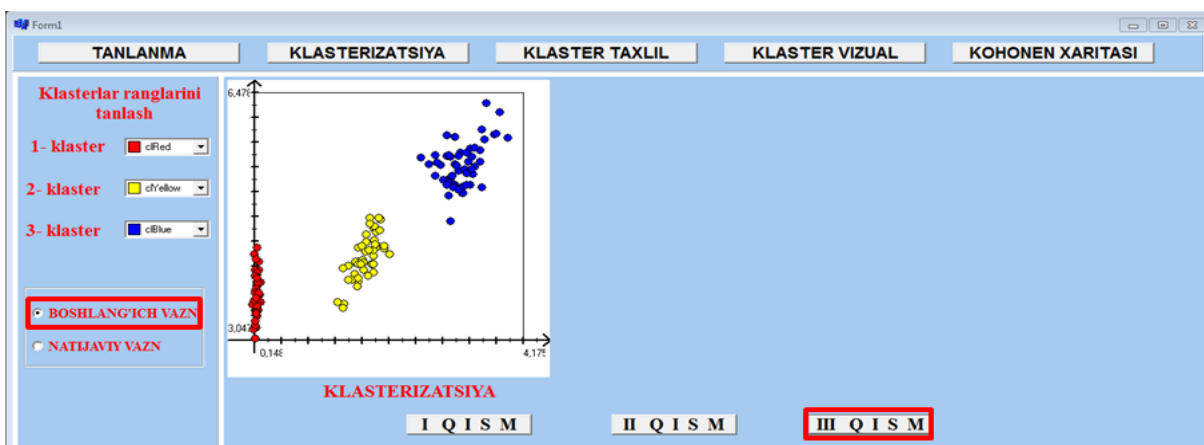


19-rasm. “BOSHLANG‘ICH VAZN” koeffisienlari asosida klasterdagi tanlanmalarni alomatlarini vizual ko‘rinishi

Boshqa alomatlarga ko‘ra klasterlarni vizual ko‘rish uchun “II QISM” (20-rasm) va “III QISM” (21-rasm) tugamlarini bosiladi. Natijalar quyidagicha ko‘rinishdad bo‘ladi.

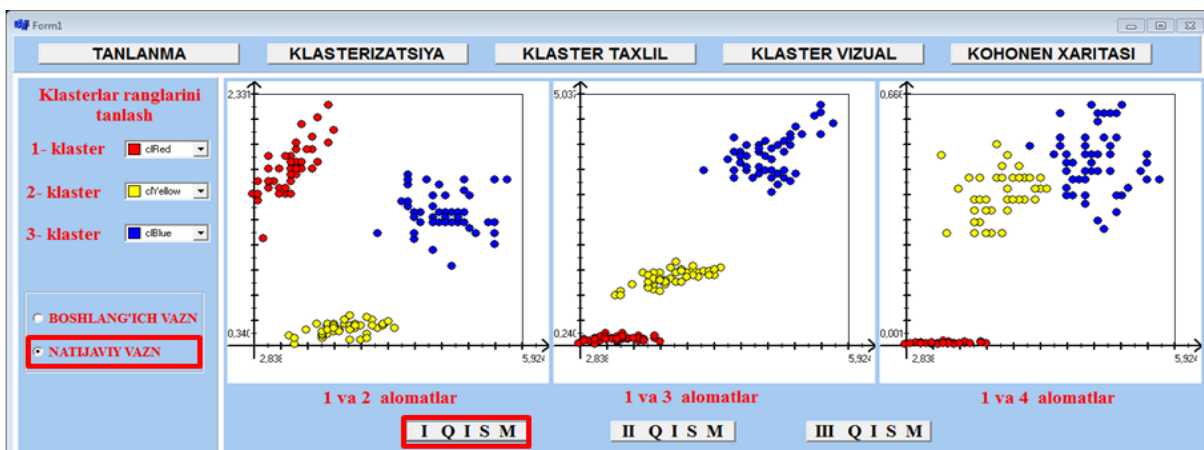


20-rasm. “BOSHLANG‘ICH VAZN” koeffisientlari asosida (2;3), (2;4) va (3;4) alomatlarning vizual ko‘rinishi



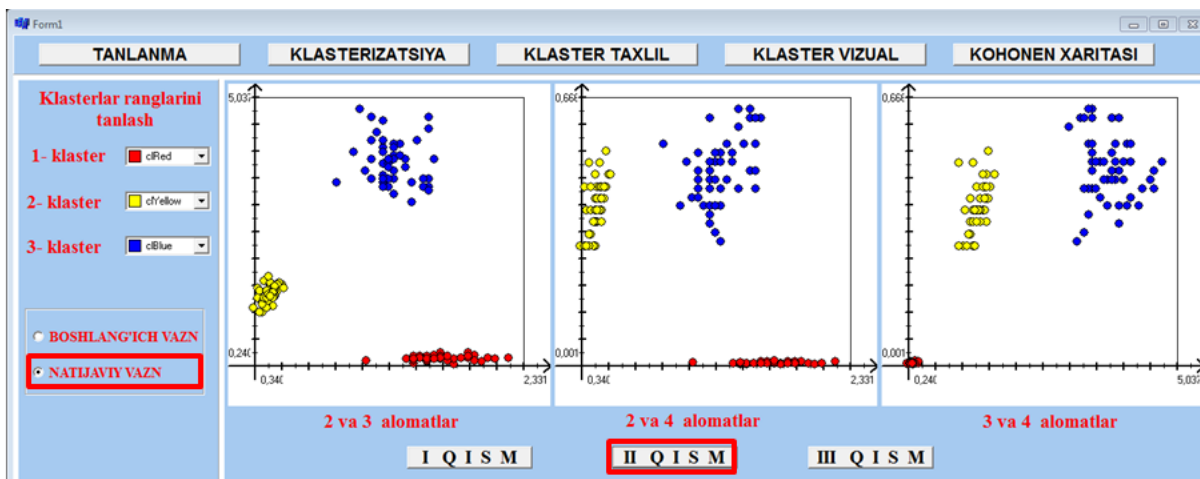
21-rasm. “BOSHLANG‘ICH VAZN” koeffisientlari asosida klasterizatsiyaning vizual ko‘rinishi

“KLASTER VIZUAL” oynasidagi RadioGroup dan “NATIJAVIY VAZN” tanlansa berilgan tanlanmani natijaviy vazn koeffisienlariga asosida klasterlar vizuallashtiriladi (22-rasm).

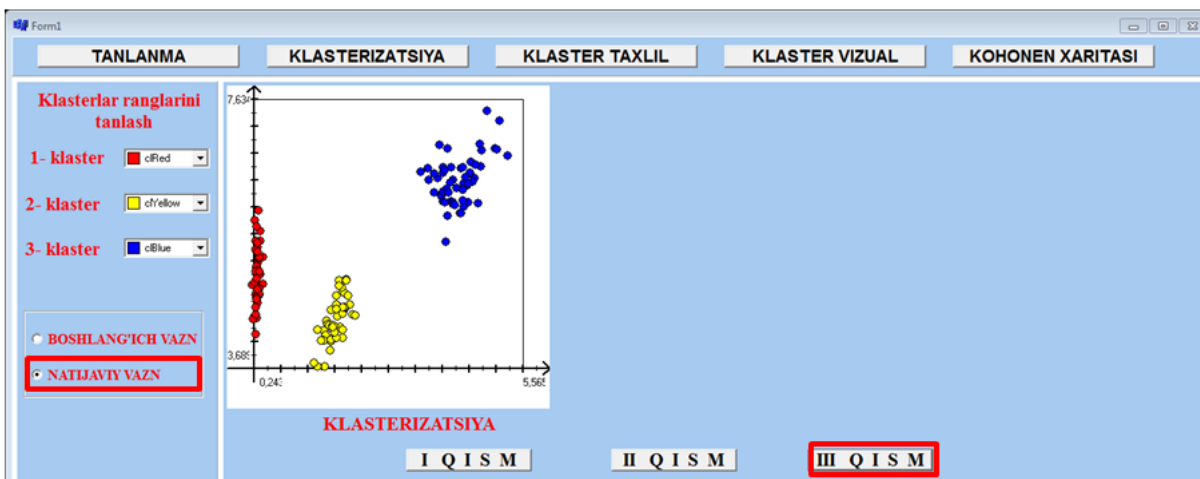


22-rasm. “NATIJAVIY VAZN” ko‘ffisientlari asosida klasterdagi tanlanmalarni alomatlarini vizual ko‘rinishi

Boshqa alomatlariga ko‘ra klasterlarni vizual ko‘rish uchun “II QISM” (23-rasm) va “III QISM” (24-rasm) tugamlarini bosiladi. Natijalar quyidagicha ko‘rinishdad bo‘ladi.



23-rasm. “NATIJAVIY VAZN” ko‘ffisientlari asosida (2;3), (2;4) va (3;4) alomatlarning vizual ko‘rinishi

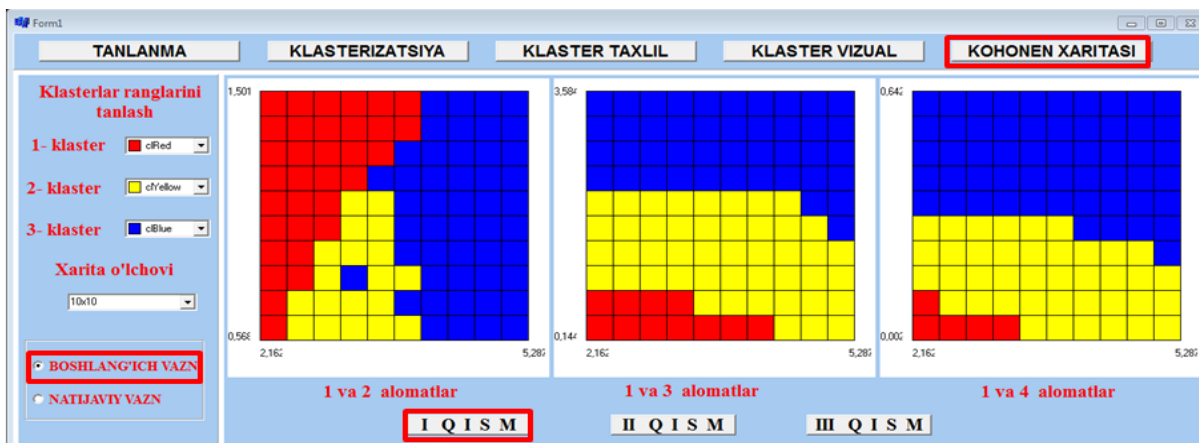


24-rasm. “NATIJAVIY VAZN” ko‘ffisientlari asosida klasterizatsiyaning vizual ko‘rinishi

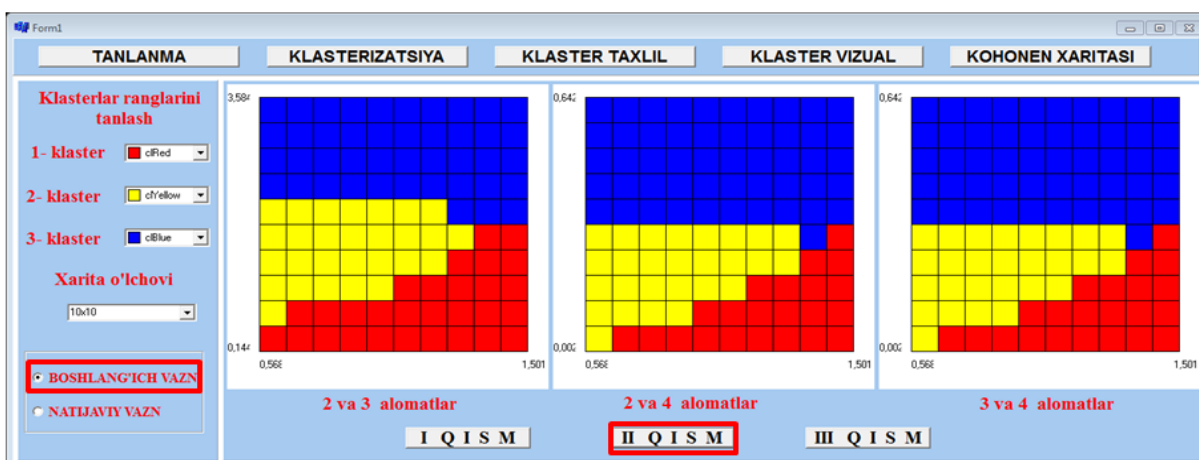
Ushbu dissertatsiya ishida “O‘z-o‘zini tashkil qiluvchi Kohonen xaritasi” to‘rburchakli elementlardan foydalanib tuzilgan. Buning uchun Kohonen xaritasi to‘rburchaklarga bo‘linadi. Vizualashtirishda bo‘lingan to‘rburchaklar qanchalik ko‘p bo‘lsa natijani ham shunchalik aniq ko‘rish mumkin.

Yaratilgan dasturiy ta'minotida Kohonen xaritasini 10x10, 30x30 va 100x100 bo'laklarga bo'lish mumkin. Bundan tashqari dasturda har bir klasterni rangini ComboBoxdan tanlash orqali o'zgartirish mumkin.

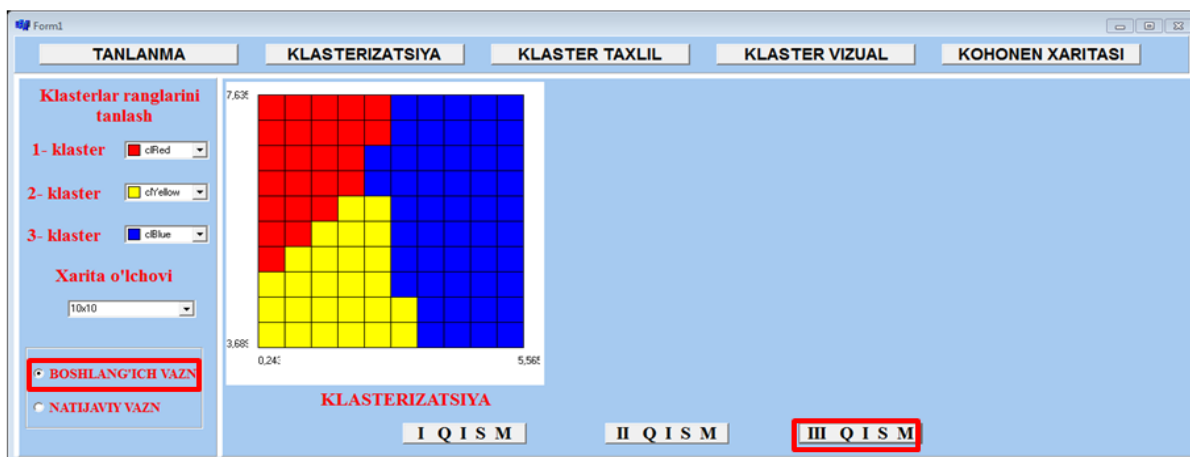
Kohonen xaritasini vizuallashtirishni "Iris gulining tanlanmalari" da tadbiq qilish orqali dastur foydalanish yo'riqnomasi keltirilgan.



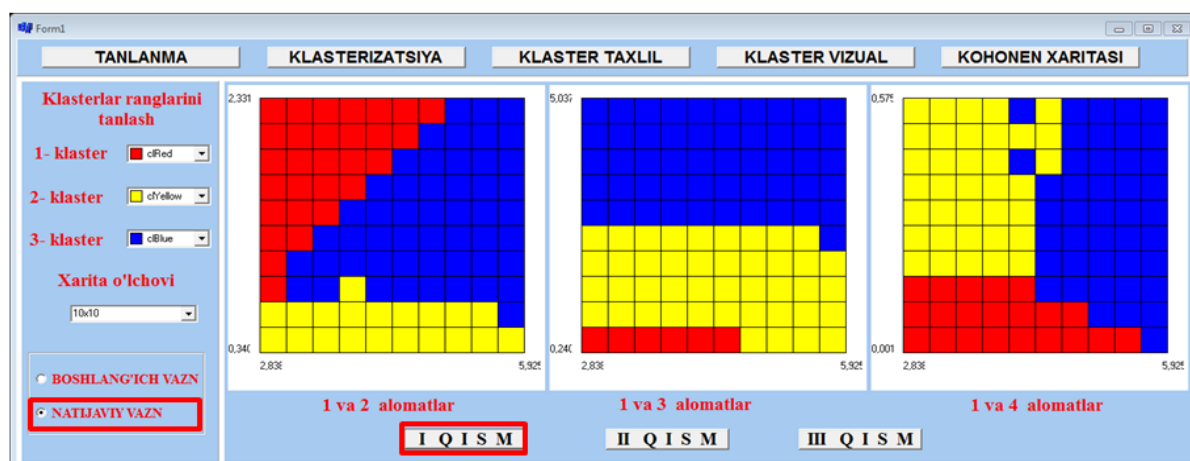
25-rasm. "BOSHLANG'ICH VAZN" koeffisienlari asosida klasterdagi (1;2), (1;3) va (1;4) alomatlarning Kohonen xaritasi



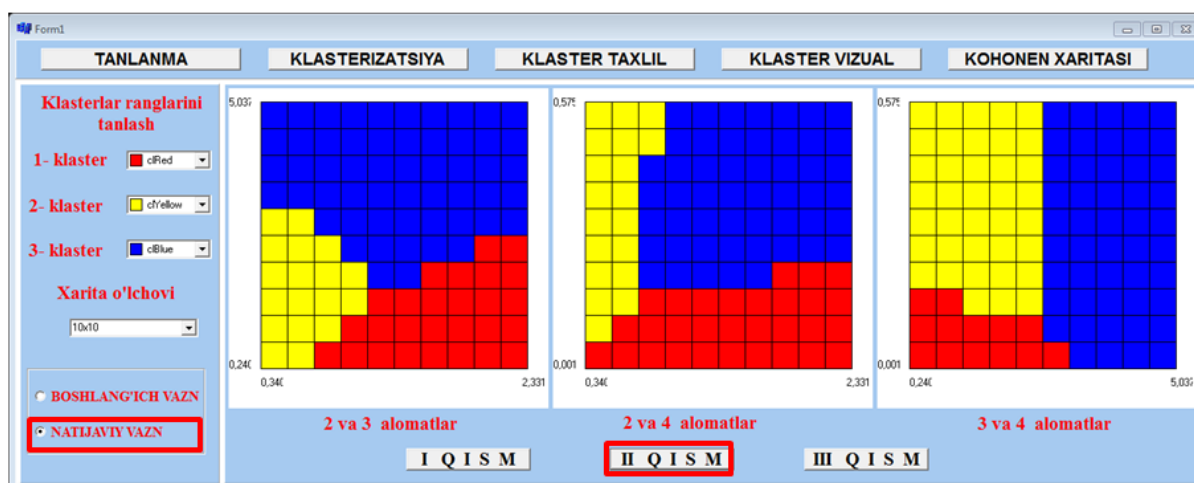
26-rasm. "BOSHLANG'ICH VAZN" koeffisienlari asosida klasterdagi (2;3), (2;4) va (3;4) alomatlarning Kohonen xaritasi



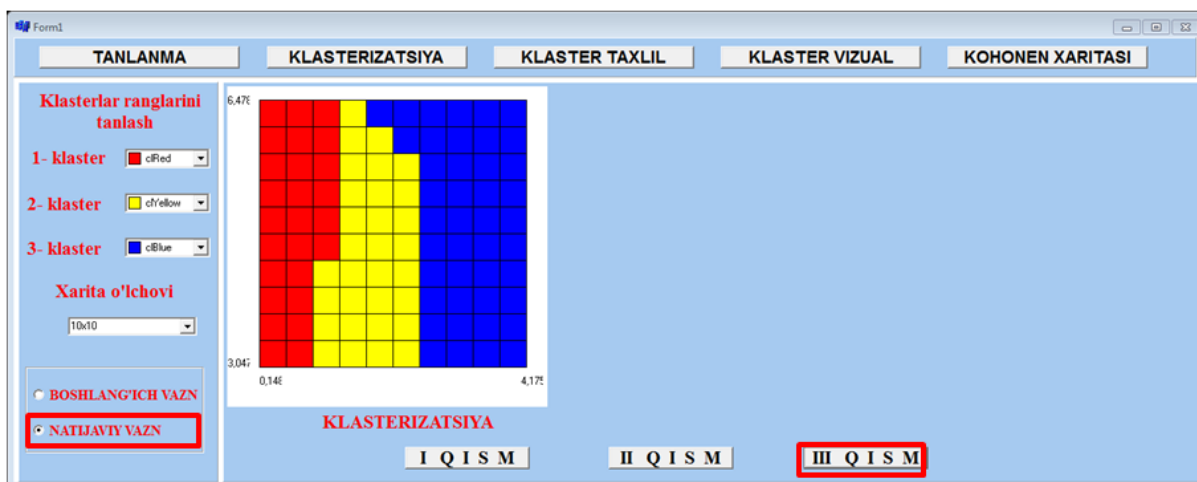
27-rasm. “BOSHLANG‘ICH VAZN” koeffisienlari asosida klasterizatsiyani Kohonen xaritasi



28-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterdagi (1;2), (1;3) va (1;4) alomatlarning Kohonen xaritasi

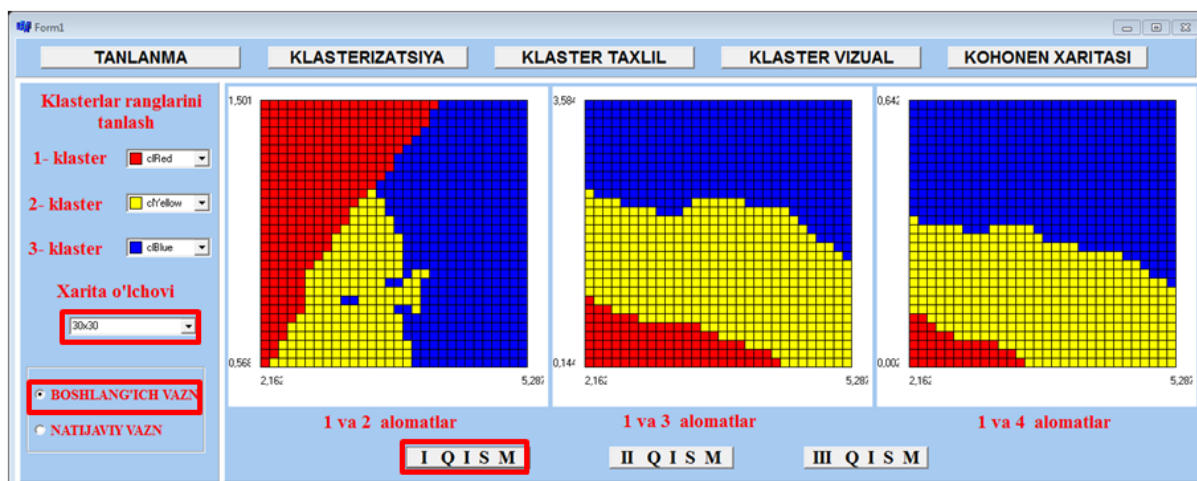


29-rasm. “NATIJAVIY VAZN” koeffitsienlari asosida klasterdagi (2;3), (2;4) va (3;4) alomatlarning Kohonen xaritasi

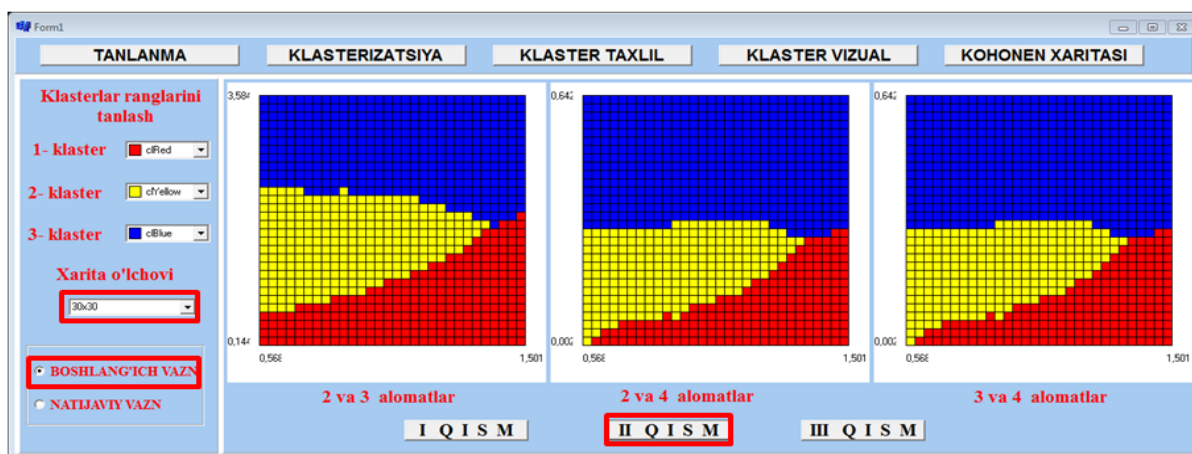


30-rasm. “NATIJAVIY VAZN” koeffitsienlari asosida klasterizatsiyani Kohonen xaritasi

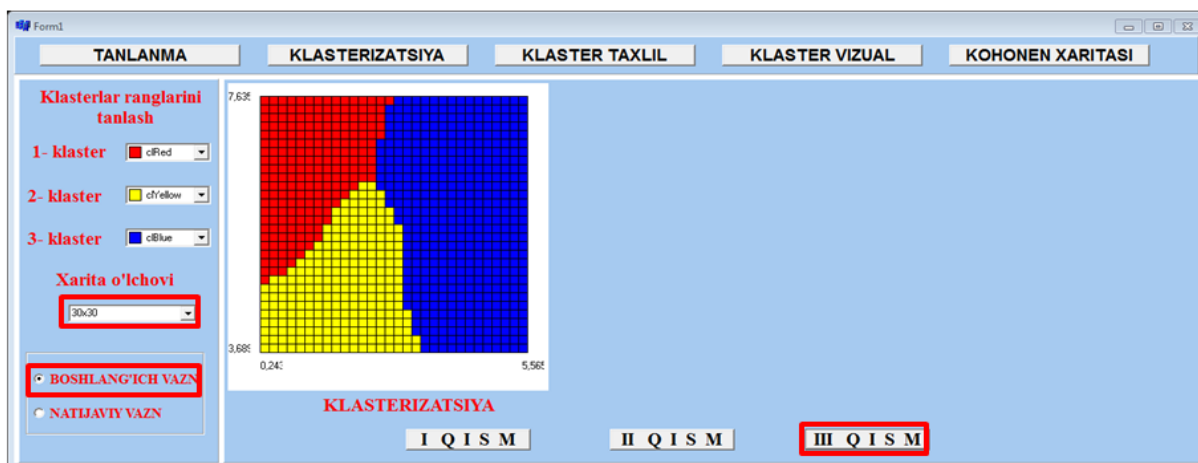
Yuqorida keltirilgan rasmlarda natijani aniq tasavvur qilish qiyin bo‘ladi. Shuning uchun xaritani eni va bo‘yini 30 tadan bo‘lib chiqamiz. Natijada xaritani aniq tasavvur qilish biroz yaxshiroq bo‘ladi. Buni 31- 36- rasmlarda ko‘rish mumkin.



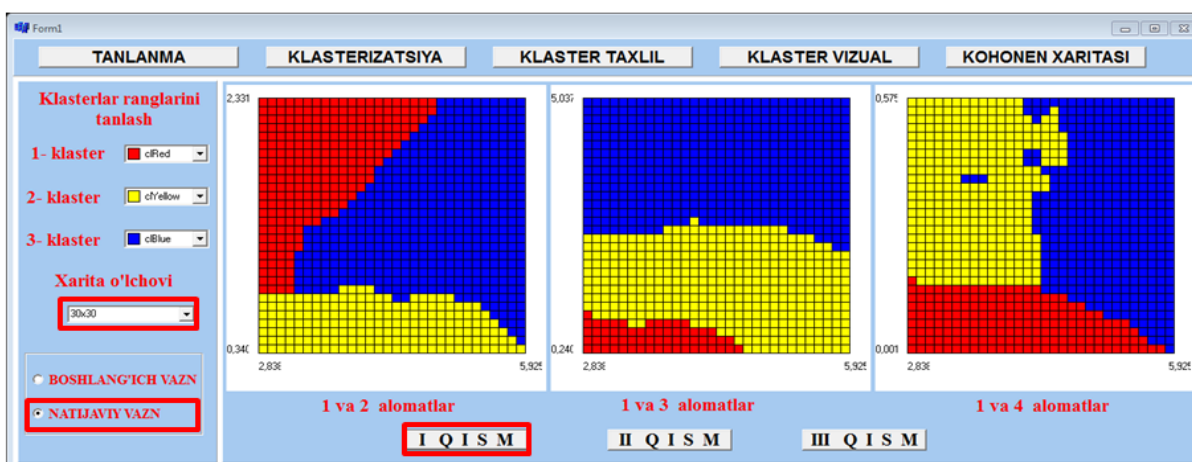
31-rasm. “BOSHLANG‘ICH VAZN” koeffitsienlari asosida klasterdagi (1;2), (1;4) va (1;4) alomatlarning Kohonen xaritasi



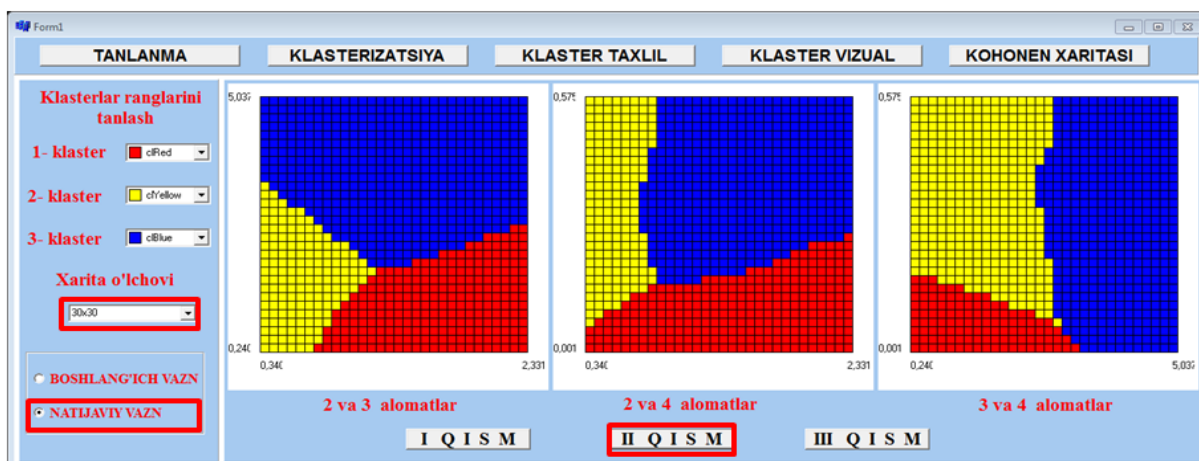
32-rasm. “BOSHLANG‘ICH VAZN” koeffisienlari asosida klasterdagi (2;3), (2;4) va (3;4) alomatlarning Kohonen xaritasi



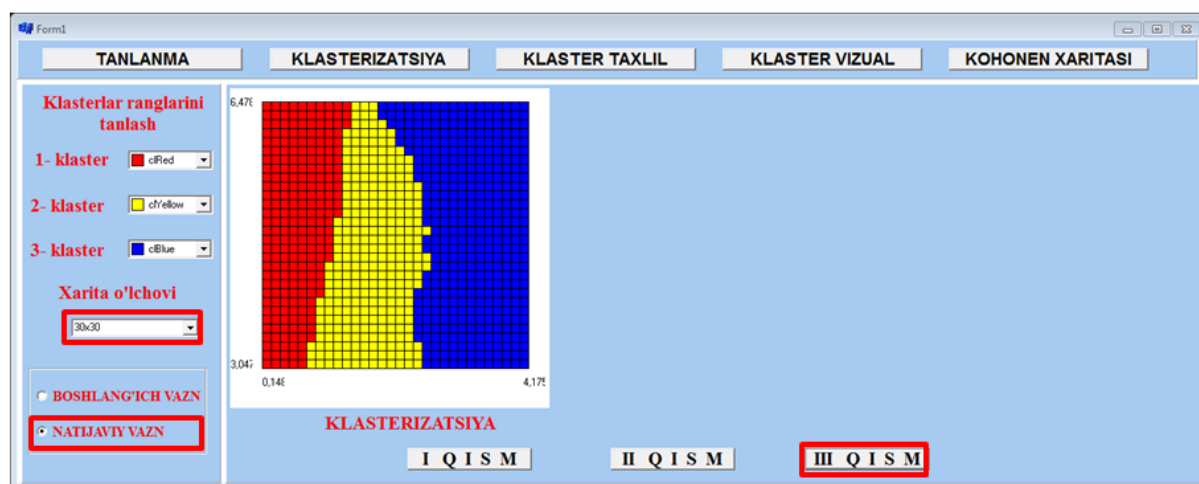
33-rasm. “BOSHLANG‘ICH VAZN” koeffisienlari asosida klasterizatsiyani Kohonen xaritasi



34-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterdagi (1;2), (1;4) va (1;4) alomatlarning Kohonen xaritasi

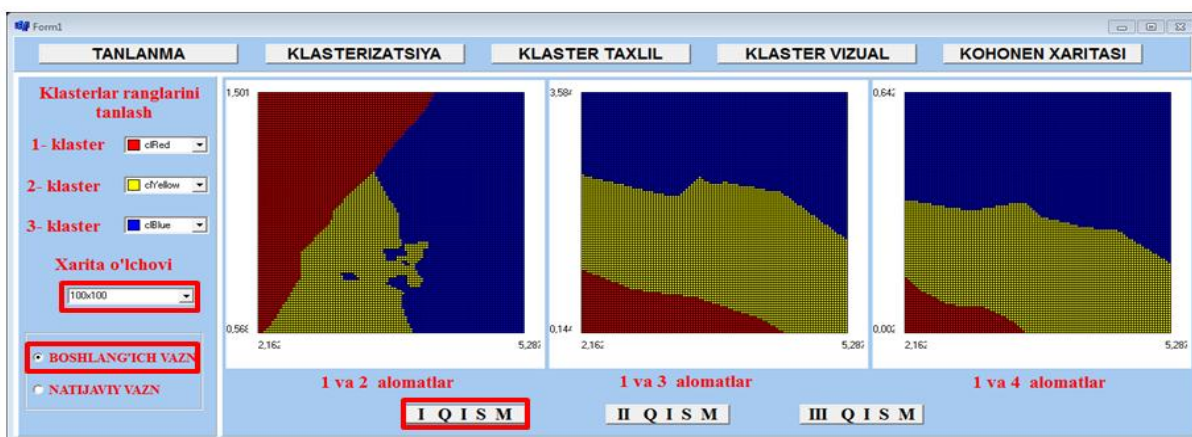


35-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterdagi (1;2), (2;4) va (3;4) alomatlarning Kohonen xaritasi

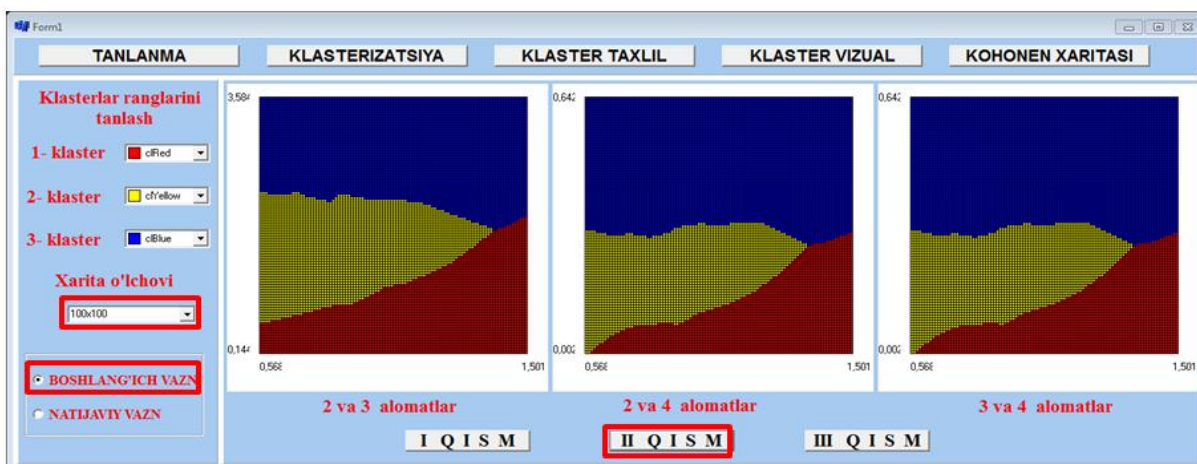


36-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterizatsiyani Kohonen xaritasi

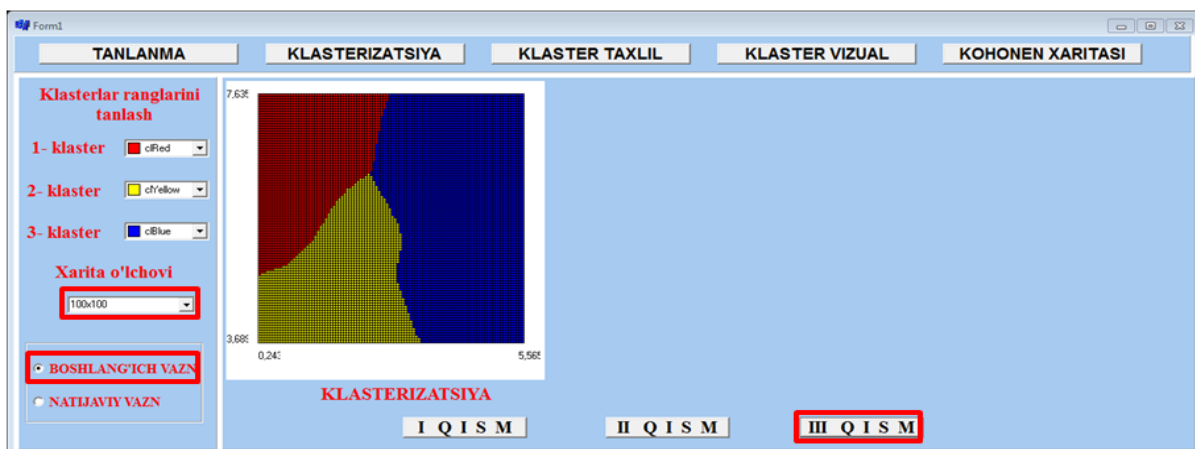
Yuqorida keltirilgan rasmlarda natijani aniq tasavvur qilish qiyin bo‘ladi. Shuning uchun xaritani eni va bo‘yini 100 tadan bo‘lib chiqamiz. Natijada xaritani aniq tasavvur qilish biroz yaxshiroq bo‘ladi. Buni 37- 42- rasmlarda ko‘rish mumkin.



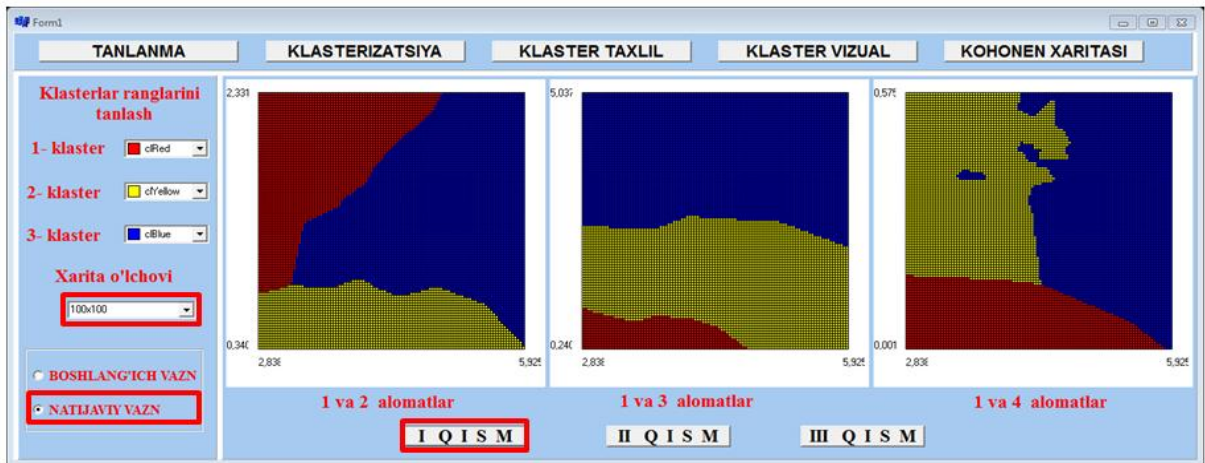
37-rasm. “BOSHLANG‘ICH VAZN” koefitsienlari asosida klasterdagi alomatlarning Kohonen xaritasi



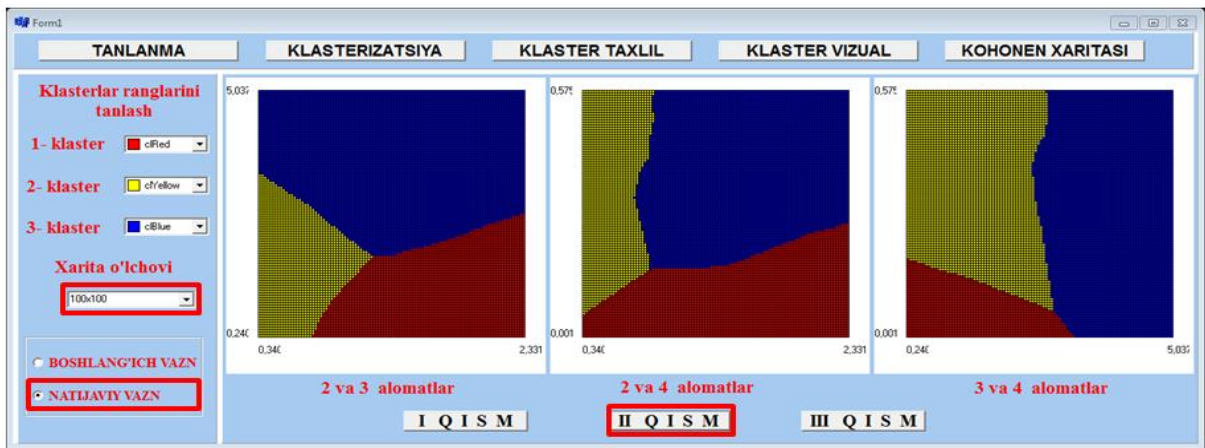
38-rasm. “BOSHLANG‘ICH VAZN” koefitsienlari asosida klasterdagi (1;2), (2;4) va (3;4) alomatlarning Kohonen xaritasi



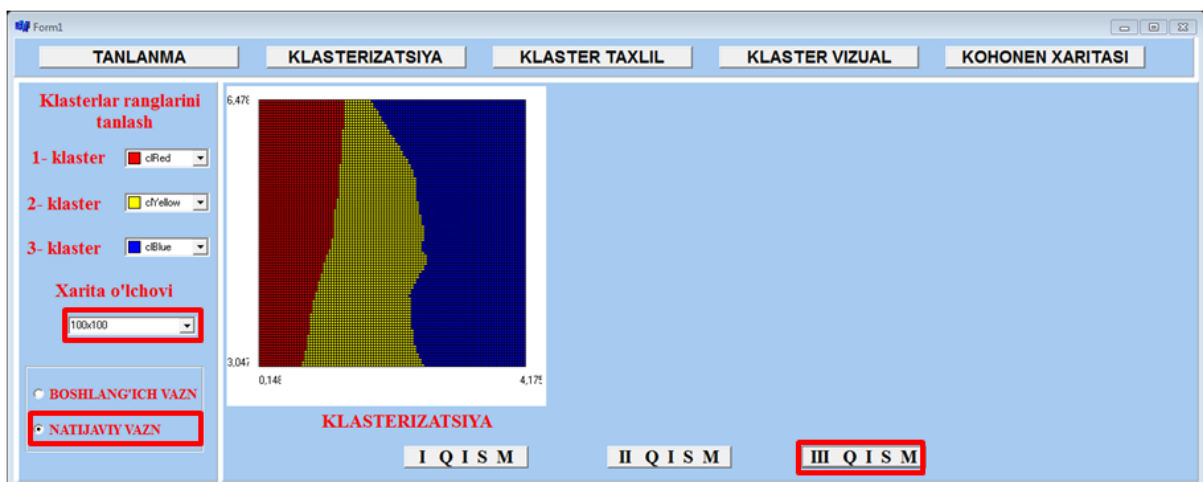
39-rasm. “BOSHLANG‘ICH VAZN” koefitsienlari asosida klasterizatsiyani Kohonen xaritasi



40-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterdagi (1;2), (2;4) va (3;4) alomatlarning Kohonen xaritasi



41-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterdagi (1;2), (2;4) va (3;4) alomatlarning Kohonen xaritasi



42-rasm. “NATIJAVIY VAZN” koeffisienlari asosida klasterizatsiyani Kohonen xaritasi

III BOB bo'yicha xulosa

Bu bobda informatsion modelning “O‘z-o‘zini tashkil qiluvchi Kohonen xaritari” usulini eng asosiy qismi yani Kohonen xaritasi hosil qilish bo‘limini yaratdim. Modelning bu bo‘limida Kohonen xaritasini hosil qilish xaritani to‘rtburchak elementlarga bo‘lib oldim. Kohonen xaritasini 100 ta, 900 va 10000 elementlarga bo‘lib olib hosil qildim. Kohonen xaritasini 10000 ta to‘rtburchak elementlarga bo‘lganimda natija 100 ta va 900 elementga bo‘lganimga nisbatan aniqroq ko‘rindi.

XULOSA

Ushbu dissertatsiya ishi birinchi bobida Kohonen neyron to'rlari va Kohonen xaritalari tushunchasi, Kohonen neyron to'rini qurish printsiplari, Kohonen neyron to'rini o'qituvchisiz o'qitish, Kohonen xaritalarini qurish printsiplari, Kohonen xaritalar o'rganish algoritmlari va algoritmni paketli o'qitish usularini o'rgandim.

Bundan tashqari bu usulning boshqa usullardan kamchiligi va ustunliklarini o'rgandim. Kiruvchi qiymatlar miqdoriy va usulda Yevklid fazoda bo'lishi kerakligini, shuningdek alomatlarining qiymatlari normal-lashtirish usulari o'rgandim.

Dissertatsiya ishimning ikkinchi bobida klasterlarga ajratish metod va usullari o'rgandim. Klasterlarga ajratib bo'lgandan so'ng natija qanchalik to'g'riligini tekshirish uchun klaster tahlil o'tkazish usullari ko'rib chiqdim. "O'z o'zini tashkil qiluvchi kohonen neyron to'rida klasterizatsiya usuli"ni dasturini tuzilda va bu dastur "Talabalarning o'rtacha o'zlashtirishi" va "Iris guli" talanmalarida tadbiq qildim. Tanlangan test tanlanmalarda dastur yaxshi natija oldim.

Dissertatsiya ishimning ikkinchi bobda informatsion modelning "O'z-o'zini tashkil qiluvchi Kohonen xaritari" usulini eng asosiy qismi yani Kohonen xaritasi hosil qilish bo'limini yaratdim. Modelning bu bo'limida Kohonen xaritasini hosil qilish xaritani to'rburchak elementlarga bo'lib oldim. Kohonen xaritasini 100 ta, 900 va 10000 elementlarga bo'lib olib hosil qildim. Kohonen xaritasini 10000 ta to'rtburchak elementlarga bo'lganimda natija 100 ta va 900 elementga bo'lganimga nisbatan aniqroq ko'rindi.

ADIVOTLAR RO‘YXATI

1. T.Honkela, S.K.Lagus, T.Kohonen. Exploration of Full-Text Databases with Self-Organizing Map // Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 56-62
2. Teuvo Kohonen. Self-Organizing Map. Springer, Berlin, Heidelberg. 2001.
3. Nils J. Nilson “The Quest for Artificial Intellegence”. Washington 2008.
4. Terry Norton “Learning C++ by Developing Games with Unity 3D”. New York 2009.
5. John Sharp “Microsoft Visual C++ 2012 Step by Step”. Washington 2014.
6. Стариков Алексей Самоорганизующиеся карты Кохонена – математический аппарат [Электрон адабиёт]/ Стариков Алексей – Режим доступа: <https://basegroup.ru/community/articles/som>, свободный.
7. Золин А.Г., Силаева А.Ю. Применение нейронных сетей в медицине. Актуальные проблемы науки, экономики и образования XXI века : материалы II Международной научно-практической конференции, 5 марта – 26 сентября 2012 года. Е. Н. Шереметьева. – Самара: Самарский институт (фил.) РГТЭУ, 2012. – 392 с.— с. 264-271. — <http://www.creativeconomy.ru/articles/25938/>.
8. Лекции по организatsии интеллектуальных вычислений [Электронный ресурс] – Режим доступа: <http://www.studmed.ru/docs/document>.
9. Горбаченко В.И. Сети и карты Кохонена [Электрон адабиёт] - Режим доступа: http://gorbachenko.self-organization.ru/articles/Self-organizing_map.pdf.
10. Заенцев И.В. Нейронные сети: основные модели [Электрон адабиёт] – Режим доступа: <http://nncourse.chat.ru/course.pdf>.
11. Shyam M. Guthikonda Kohonen Self-Organizing Maps [Электрон адабиёт]. Shyam M. Guthikonda – 2011г – Режим доступа: <http://www.shy.am/wp-content/uploads/2009/01/kohonen-self-organizing-maps-shyam-guthikonda.pdf>, свободный.

12. Markus Varsta Temporal Kohonen Map and the Recurrent Self-Organizing Map: Analytical and Experimental Comparison [Электрон адабиёт]/ Markus Varsta , Jukka Heikkonen , Jouko Lampinen , Joseè Del R. Millaèn – Режим доступа: <http://web.cs.dal.ca/~tt/CSCI650805/papers/tsom.pdf>.
13. Markus Varsta Context Learning with Self-Organizing Maps [Электрон адабиёт] Markus Varsta , Jukka Heikkonen , Jouko Lampinen , Joseè Del R. Millaèn – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi>.
14. С. Aaron and Y. Tadjeddine. Description of the group dynamic of funds' managers using Kohonen's s map. European Symposium on Applied Neural Network (ESANN04), 2004.
15. Ж.Қ.Одилов, А.Х.Ишмухаммедов. “Коҳоненнинг ўз-ўзини ташкил этувчи хариталари - математик модели”. Навои давлат Педагогика институти “Информатика ва ахборот коммуникация технологиялари таълимини модернизатсиялаш истиқболлари” номли илмий конференция , 25.05.2018 йил, 119-121-бетлар.
16. Ж.Қ.Одилов, А.Х.Ишмухаммедов. “Коҳоненнинг ўз-ўзини ташкил этувчи хариталарини - ўқитувчисиз ўргатиш алгоритми ”. Мирзо Улуғбек номидаги Ўзбекистон Миллий университети “Амалий математика ва информацион технологияларнинг долзарб муаммолари – Ал-Хоразмий 2018 ” халқаро анжумани, 13-15-Сентябр 2018-йил, 216-бет.
17. Ж.Қ.Одилов, А.Х.Ишмухаммедов. “Ўз-ўзини ташкил қилувчи коҳонен нейрон тармоғида кластеризатсия”. Муҳаммад Ал-Хоразмий номидаги Тошкент Ахборот технологиялари университети “Иқтисодиётнинг тармоқларини инноватсион ривожлантиришда ахборот-коммуникация технологияларининг аҳамияти” номли илмий конференция 14-15-март 2019-йил, 12-14-бетлар.

ILOVA 1

Dasturiy ta'minot matni

```
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <math.h>
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
bool b1=true,b2=false,b3=false,b=true;
bool k1=true,k2=false,k3=false,kkk=false;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
class Kohonen_map
{
TPoint nuqta [5];
double **x,**w,**w1,d,t,**x1,**kls;
float **MK,**xy;
int n,m,k,i,j,klaster [4],*K,n1,m1,k1,nn,mm,**MN;
bool wt;

FILE *f,*f1,*f2,*f3,*f4;
public:
//funksiyalar
int getn(){return n;};
int getm(){return m;};
int getk(){return k;};
void set_X_vector(String fn); //Barcha tanlanmalarni fayldan o'qib olish
void set_weight(); //Waznlari [0..1] oraliqdan tasodifiy tanlash
double Evklid_distanse(int x_i, int w_j);
double Min_d_j(int x_i, int &d_j);
void Set_Winner_neyron(int x_i);
double nyu(double t){return 0.1*exp(-t/0.4);}
void Klastrizatsiya();
void set_weight1();
void Klaster_tahlil();
void Klaster_tahlil_file();
```

```

void koordinata_oqi(TImage *Imagen,float ulchov,int x, int y,TLabel *Ex1,
TLabel *Ex2,TLabel *Ey1, TLabel *Ey2);
void Klaster_chiz(TImage *Imagen,int xi,int yi,float ulchov,int wn,TLabel
*Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2);
void Klaster_chiz1(TImage *Imagen,int x1,int y1,int x2,int y2,float ulchov,int
wn,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2);
void Kohonen_xaritasi(TImage *Image,int x,int y,int ulchov,int kn,int
File_Name,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2,int xx,int
yy);
void Kohonen_xaritasi1(TImage *Imagen,int x1,int x2,int y1,int y2,float
ulchov,int kn,int File_Name,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel
*Ey2,int xx,int yy);

//bajarish
};
    Kohonen_map KS;
    String FileName;
////////////////////////////////////
void Kohonen_map::Kohonen_xaritasi1(TImage *Imagen,int x1,int x2,int
y1,int y2,float ulchov,int kn,int File_Name,TLabel *Ex1, TLabel *Ex2,TLabel
*Ey1, TLabel *Ey2,int xx,int yy)
{   int g=ulchov;
TImage *img=Imagen;
img->Visible=true;
img->Left=xx; img->Top=yy;
Ex1->Left=xx+g/10+g/50; Ex1->Top=yy+g+g/10-g/50;
Ex2->Left=xx+g/10+g; Ex2->Top=yy+g+g/10-g/50;
Ex1->Caption="";Ex2->Caption="";Ey1->Caption="";Ey2->Caption="";
Ey1->Left=xx; Ey1->Top=yy+g-g/50;
Ey2->Left=xx; Ey2->Top=yy+g/20-g/50;
Ex1->Width=25; Ex1->Height=15;
img->Canvas->Brush->Color=c1White;
img->Canvas->Pen->Color=c1White;
img->Canvas->Rectangle(0,0,g+g/5,g+g/5);
if(File_Name==1)f3=fopen("10.txt","rt");
else if(File_Name==2)f3=fopen("30.txt","rt");
else if(File_Name==3)f3=fopen("100.txt","rt");
fscanf(f3,"%i %i",&n1,&m1); //n-tugunlar soni m-to'rtburchaklar soni

MN=new int* [m1]; MK=new float* [n1];
for(i=0;i<m1;i++)MN [i]=new int [4];
for(i=0;i<n1;i++)MK [i]=new float [2];
for(i=0;i<m1;i++)
    fscanf(f3,"%i %i %i %i",&MN [i] [0],&MN [i] [1],&MN [i] [2],&MN [i] [3]);

```

```

for(i=0;i<n1;i++)
  fscanf(f3,"%f %f",&MK [i] [0],&MK [i] [1]);
float **M,**N;
M=new float* [m1];
for(i=0;i<m1;i++)M [i]=new float [2];

for(i=0;i<m1;i++)
{
M [i] [0]=((MK [MN [i] [0]-1] [0]+24)+(MK [MN [i] [1]-1] [0]+24))/2;
M [i] [1]=((MK [MN [i] [0]-1] [1])+(MK [MN [i] [2]-1] [1]))/2;
}
Ey1->Top=Ey1->Top+10;

img->Canvas->Pen->Color=clBlack;

fclose(f3);
f4=fopen("Tanlanma iris aniq.txt","rt");

  int u;
  fscanf(f4,"%i",&u);
  fscanf(f4,"%i",&u);
  fscanf(f4,"%i",&u);
  xy=new float* [n];  N=new float* [n];
  for(i=0;i<n;i++){xy [i]=new float [m]; N [i]=new float [2];}

  for(i=0;i<n;i++){
  for(j=0;j<m;j++){
  fscanf(f4,"%f",&xy [i] [j]);
  }
  fscanf(f4,"%i",&K [i]);
  }

  fclose(f4);
  float xmax,xmin,ymax,ymin,xmk,ymk;  i=0;
img->Canvas->Pen->Width=1;

///_*_***_*_*_*_*_*_*

if(kn==2)
{
xmax=xy [0] [x1]*w [K [i]-1] [x1]+xy [0] [x2]*w [K [i]-1] [x2]; xmin=xy [0]
[x1]*w [K [i]-1] [x1]+xy [0] [x2]*w [K [i]-1] [x2];
ymax=xy [0] [y1]*w [K [i]-1] [y1]+xy [0] [y2]*w [K [i]-1] [y2]; ymin=xy [0]
[y1]*w [K [i]-1] [y1]+xy [0] [y2]*w [K [i]-1] [y2];
for(i=0;i<n;i++)

```

```

{
  if(xmax<xy [i] [x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1] [x2])xmax=xy [i]
[x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1] [x2];
  if(xmin>xy [i] [x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1] [x2])xmin=xy [i]
[x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1] [x2];
  if(ymax<xy [i] [y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2])ymax=xy [i]
[y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2];
  if(ymin>xy [i] [y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2])ymin=xy [i]
[y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2];
}
  xmk=g/(xmax-xmin) ;
  ymk=g/(ymax-ymin);

```

```

Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true; Ex2->Visible=true;
Ey1->Visible=true; Ey2->Visible=true;

```

```

for(i=0;i<n;i++)
{ if(K [i]!=1){
  N [i] [0]=g/10+(xy [i] [x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1] [x2]-
xmin)*xmk;
  N [i] [1]=g-(xy [i] [y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2]-
ymin)*ymk+30;
}
else
{
  N [i] [0]=g/10+(xy [i] [x1]*w [K [i]-1] [x1]+xy [i] [x2]*w [K [i]-1]
[x2]+xmin)*xmk;
  N [i] [1]=g-(xy [i] [y1]*w [K [i]-1] [y1]+xy [i] [y2]*w [K [i]-1] [y2]-(ymax-
xmax))*ymk+30;
}
}

```

```

float d,mind;
int p,dh;

```

```

i=0;
for(i=0;i<m1;i++)

```

```

{ mind=sqrt(pow(N [0] [0]-M [i] [0],2)+pow(N [0] [1]-M [i] [1],2));
  dh=K [0];
  for(j=0;j<n;j++)
  {
  d=sqrt(pow(N [j] [0]-M [i] [0],2)+pow(N [j] [1]-M [i] [1],2));
  if(mind>d)
  {
  mind=d;
  dh=K [j];
  }
  }

```

```

nuqta [0].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [0].y=MK [MN [i] [0]-1] [1]-
g/20;
nuqta [1].x=MK [MN [i] [1]-1] [0]+g/10; nuqta [1].y=MK [MN [i] [1]-1] [1]-
g/20;
nuqta [2].x=MK [MN [i] [3]-1] [0]+g/10; nuqta [2].y=MK [MN [i] [3]-1] [1]-
g/20;
nuqta [3].x=MK [MN [i] [2]-1] [0]+g/10; nuqta [3].y=MK [MN [i] [2]-1] [1]-
g/20;
nuqta [4].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [4].y=MK [MN [i] [0]-1] [1]-
g/20;

```

```

img->Canvas->Pen->Width=0.5;
img->Canvas->Pen->Color=clBlack;

```

```

if(dh==1){img->Canvas->Brush->Color=Form1->ColorBox5->Selected;}
//Image1->Canvas->Pen->Color=clRed;

```

```

if(dh==2){img->Canvas->Brush->Color=Form1->ColorBox6->Selected;} //
Image1->Canvas->Pen->Color=clYellow;

```

```

if(dh==3){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
if(dh==4){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
img->Canvas->Polygon(nuqta,4);
}
}

```

```

///_.*_**_.*_.*_.*_.*_

```

```

if(kn==1)
{
xmax=xy [0] [x1]*w1 [K [i]-1] [x1]+xy [0] [x2]*w1 [K [i]-1] [x2]; xmin=xy [0]
[x1]*w1 [K [i]-1] [x1]+xy [0] [x2]*w1 [K [i]-1] [x2];

```

```

ymax=xy [0] [y1]*w1 [K [i]-1] [y1]+xy [0] [y2]*w1 [K [i]-1] [y2]; ymin=xy [0]
[y1]*w1 [K [i]-1] [y1]+xy [0] [y2]*w1 [K [i]-1] [y2];
for(i=0;i<n;i++)
{
if(xmax<xy [i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1] [x2])xmax=xy
[i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1] [x2];
if(xmin>xy [i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1] [x2])xmin=xy
[i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1] [x2];
if(ymax<xy [i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2])ymax=xy
[i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2];
if(ymin>xy [i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2])ymin=xy
[i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);

```

```

Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true; Ex2->Visible=true;
Ey1->Visible=true; Ey2->Visible=true;

```

```

for(i=0;i<n;i++)
{
if(K [i]!=1){
N [i] [0]=g/10+(xy [i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1] [x2]-
xmin)*xmk;
N [i] [1]=g-(xy [i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2]-
ymin)*ymk+30;
}
else
{
N [i] [0]=g/10+(xy [i] [x1]*w1 [K [i]-1] [x1]+xy [i] [x2]*w1 [K [i]-1]
[x2]+xmin)*xmk;
N [i] [1]=g-(xy [i] [y1]*w1 [K [i]-1] [y1]+xy [i] [y2]*w1 [K [i]-1] [y2]-
(ymax-xmax))*ymk+30;
}
}
}

```

```
float d,mind;
```

```

int p,dh;

i=0;
for(i=0;i<m1;i++)
{ mind=sqrt(pow(N [0] [0]-M [i] [0],2)+pow(N [0] [1]-M [i] [1],2));
  dh=K [0];
  for(j=0;j<n;j++)
  {
d=sqrt(pow(N [j] [0]-M [i] [0],2)+pow(N [j] [1]-M [i] [1],2));
if(mind>d)
  {
mind=d;
dh=K [j];
  }
}

nuqta [0].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [0].y=MK [MN [i] [0]-1] [1]-
g/20;
nuqta [1].x=MK [MN [i] [1]-1] [0]+g/10; nuqta [1].y=MK [MN [i] [1]-1] [1]-
g/20;
nuqta [2].x=MK [MN [i] [3]-1] [0]+g/10; nuqta [2].y=MK [MN [i] [3]-1] [1]-
g/20;
nuqta [3].x=MK [MN [i] [2]-1] [0]+g/10; nuqta [3].y=MK [MN [i] [2]-1] [1]-
g/20;
nuqta [4].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [4].y=MK [MN [i] [0]-1] [1]-
g/20;

img->Canvas->Pen->Width=0.5;
img->Canvas->Pen->Color=clBlack;

if(dh==1){img->Canvas->Brush->Color=Form1->ColorBox5->Selected;}
//Image1->Canvas->Pen->Color=clRed;

if(dh==2){img->Canvas->Brush->Color=Form1->ColorBox6->Selected;} //
Image1->Canvas->Pen->Color=clYellow;

if(dh==3){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
if(dh==4){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
img->Canvas->Polygon(nuqta,4);
}
}

}
////////////////////////////////////

```

```

void Kohonen_map::Kohonen_xaritasi(TImage *Image,int x1,int y1,int
ulchov,int kn,int File_Name,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel
*Ey2,int xx,int yy)
{ int g=ulchov;
TImage *img=Image;
img->Visible=true;
img->Left=xx; img->Top=yy;
Ex1->Left=xx+g/10+g/50; Ex1->Top=yy+g+g/10-g/50;
Ex2->Left=xx+g/10+g; Ex2->Top=yy+g+g/10-g/50;
Ex1->Caption="";Ex2->Caption="";Ey1->Caption="";Ey2->Caption="";
Ey1->Left=xx; Ey1->Top=yy+g-g/50;
Ey2->Left=xx; Ey2->Top=yy+g/20-g/50;
Ex1->Width=25; Ex1->Height=15;
img->Canvas->Brush->Color=clWhite;
img->Canvas->Pen->Color=clWhite;
img->Canvas->Rectangle(0,0,g+g/5,g+g/5);
if(File_Name==1)f3=fopen("10.txt","rt");
else if(File_Name==2)f3=fopen("30.txt","rt");
else if(File_Name==3)f3=fopen("100.txt","rt");
fscanf(f3,"%i %i",&n1,&m1); //n-tugunlar soni m-to'rtburchaklar soni

MN=new int* [m1]; MK=new float* [n1];
for(i=0;i<m1;i++)MN [i]=new int [4];
for(i=0;i<n1;i++)MK [i]=new float [2];
for(i=0;i<m1;i++)
fscanf(f3,"%i %i %i %i",&MN [i] [0],&MN [i] [1],&MN [i] [2],&MN [i] [3]);

for(i=0;i<n1;i++)
fscanf(f3,"%f %f",&MK [i] [0],&MK [i] [1]);
float **M,**N;
M=new float* [m1];
for(i=0;i<m1;i++)M [i]=new float [2];

for(i=0;i<m1;i++)
{
M [i] [0]=((MK [MN [i] [0]-1] [0]+24)+(MK [MN [i] [1]-1] [0]+24))/2;
M [i] [1]=((MK [MN [i] [0]-1] [1])+(MK [MN [i] [2]-1] [1]))/2;
}
Ey1->Top=Ey1->Top+10;

img->Canvas->Pen->Color=clBlack;

fclose(f3);
f4=fopen("Tanlanma iris aniq.txt","rt");

```



```

    int u;
    fscanf(f4,"%i",&u);
    fscanf(f4,"%i",&u);
    fscanf(f4,"%i",&u);
    xy=new float* [n];  N=new float* [n];
    for(i=0;i<n;i++){xy [i]=new float [m]; N [i]=new float [2];}

    for(i=0;i<n;i++){
    for(j=0;j<m;j++){
    fscanf(f4,"%f",&xy [i] [j]);
    }
    fscanf(f4,"%i",&K [i]);
    }
    fclose(f4);

```

```

float xmax,xmin,ymax,ymin,xmk,ymk;  i=0;
img->Canvas->Pen->Width=1;

```

```

///_*_*_*_*_*_*_*_*_*_*

```

```

if(kn==1)
{
xmax=xy [0] [x1]*w [K [i]-1] [x1]; xmin=xy [0] [x1]*w [K [i]-1] [x1];
ymax=xy [0] [y1]*w [K [i]-1] [y1]; ymin=xy [0] [y1]*w [K [i]-1] [y1];
for(i=0;i<n;i++)
{
if(xmax<xy [i] [x1]*w [K [i]-1] [x1])xmax=xy [i] [x1]*w [K [i]-1] [x1];
if(xmin>xy [i] [x1]*w [K [i]-1] [x1])xmin=xy [i] [x1]*w [K [i]-1] [x1];
if(ymax<xy [i] [y1]*w [K [i]-1] [y1])ymax=xy [i] [y1]*w [K [i]-1] [y1];
if(ymin>xy [i] [y1]*w [K [i]-1] [y1])ymin=xy [i] [y1]*w [K [i]-1] [y1];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);

```

```

Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true;  Ex2->Visible=true;
Ey1->Visible=true;  Ey2->Visible=true;

```

```

for(i=0;i<n;i++)
{
N [i] [0]=g/10+(xy [i] [x1]*w [K [i]-1] [x1]-xmin)*xmk;
N [i] [1]=g-(xy [i] [y1]*w [K [i]-1] [y1]-ymin)*ymk+g/10;
}

```

```

float d,mind;
int p,dh;

```

```

i=0;
for(i=0;i<m1;i++)
{ mind=sqrt(pow(N [0] [0]-M [i] [0],2)+pow(N [0] [1]-M [i] [1],2));
dh=K [0];
for(j=0;j<n;j++)
{
d=sqrt(pow(N [j] [0]-M [i] [0],2)+pow(N [j] [1]-M [i] [1],2));
if(mind>d)
{
mind=d;
dh=K [j];
}
}
}

```

```

nuqta [0].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [0].y=MK [MN [i] [0]-1] [1]-
g/20;
nuqta [1].x=MK [MN [i] [1]-1] [0]+g/10; nuqta [1].y=MK [MN [i] [1]-1] [1]-
g/20;
nuqta [2].x=MK [MN [i] [3]-1] [0]+g/10; nuqta [2].y=MK [MN [i] [3]-1] [1]-
g/20;
nuqta [3].x=MK [MN [i] [2]-1] [0]+g/10; nuqta [3].y=MK [MN [i] [2]-1] [1]-
g/20;
nuqta [4].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [4].y=MK [MN [i] [0]-1] [1]-
g/20;

```

```

img->Canvas->Pen->Width=0.5;
img->Canvas->Pen->Color=clBlack;

```

```

if(dh==1){img->Canvas->Brush->Color=Form1->ColorBox5->Selected;}
//Image1->Canvas->Pen->Color=clRed;

```

```

if(dh==2){img->Canvas->Brush->Color=Form1->ColorBox6->Selected;} //
Image1->Canvas->Pen->Color=clYellow;

```

```

if(dh==3){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
if(dh==4){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
img->Canvas->Polygon(nuqta,4);
}
}

```

```

///_*_*_*_*_*_*_*_*_*_*

```

```

if(kn==2)
{
xmax=xy [0] [x1]*w1 [K [i]-1] [x1]; xmin=xy [0] [x1]*w1 [K [i]-1] [x1];
ymax=xy [0] [y1]*w1 [K [i]-1] [y1]; ymin=xy [0] [y1]*w1 [K [i]-1] [y1];
for(i=0;i<n;i++)
{
if(xmax<xy [i] [x1]*w1 [K [i]-1] [x1])xmax=xy [i] [x1]*w1 [K [i]-1] [x1];
if(xmin>xy [i] [x1]*w1 [K [i]-1] [x1])xmin=xy [i] [x1]*w1 [K [i]-1] [x1];
if(ymax<xy [i] [y1]*w1 [K [i]-1] [y1])ymax=xy [i] [y1]*w1 [K [i]-1] [y1];
if(ymin>xy [i] [y1]*w1 [K [i]-1] [y1])ymin=xy [i] [y1]*w1 [K [i]-1] [y1];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);

```

```

Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true; Ex2->Visible=true;
Ey1->Visible=true; Ey2->Visible=true;

```

```

for(i=0;i<n;i++)
{
N [i] [0]=g/10+(xy [i] [x1]*w1 [K [i]-1] [x1]-xmin)*xmk;
N [i] [1]=g-(xy [i] [y1]*w1 [K [i]-1] [y1]-ymin)*ymk+g/10;
}

```

```

float d,mind;
int p,dh;

```

```

i=0;

```

```

for(i=0;i<m1;i++)
{ mind=sqrt(pow(N [0] [0]-M [i] [0],2)+pow(N [0] [1]-M [i] [1],2));
  dh=K [0];
  for(j=0;j<n;j++)
  {
d=sqrt(pow(N [j] [0]-M [i] [0],2)+pow(N [j] [1]-M [i] [1],2));
if(mind>d)
  {
mind=d;
dh=K [j];
  }
}

nuqta [0].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [0].y=MK [MN [i] [0]-1] [1]-
g/20;
nuqta [1].x=MK [MN [i] [1]-1] [0]+g/10; nuqta [1].y=MK [MN [i] [1]-1] [1]-
g/20;
nuqta [2].x=MK [MN [i] [3]-1] [0]+g/10; nuqta [2].y=MK [MN [i] [3]-1] [1]-
g/20;
nuqta [3].x=MK [MN [i] [2]-1] [0]+g/10; nuqta [3].y=MK [MN [i] [2]-1] [1]-
g/20;
nuqta [4].x=MK [MN [i] [0]-1] [0]+g/10; nuqta [4].y=MK [MN [i] [0]-1] [1]-
g/20;

img->Canvas->Pen->Width=0.5;
img->Canvas->Pen->Color=clBlack;

if(dh==1){img->Canvas->Brush->Color=Form1->ColorBox5->Selected;}
//Image1->Canvas->Pen->Color=clRed;

if(dh==2){img->Canvas->Brush->Color=Form1->ColorBox6->Selected;} //
Image1->Canvas->Pen->Color=clYellow;

if(dh==3){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
if(dh==4){img->Canvas->Brush->Color=Form1->ColorBox7->Selected;}
img->Canvas->Polygon(nuqta,4);
}
}

}
//-----

//*****
void Kohonen_map::koordinata_oqi(TImage *Imagen,float ulchov,int x, int
y,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2)

```

```

{ float g=ulchov;
  TImage *img=Imagen;
  img->Visible=true;
  img->Left=x; img->Top=y;
  Ex1->Left=x+g/10+g/50; Ex1->Top=y+g+g/10-g/50;
  Ex2->Left=x+g/10+g; Ex2->Top=y+g+g/10-g/50;
  Ex1->Caption="";Ex2->Caption="";Ey1->Caption="";Ey2->Caption="";
  Ey1->Left=x; Ey1->Top=y+g-g/50;
  Ey2->Left=x; Ey2->Top=y+g/20-g/50;
  Ex1->Width=25; Ex1->Height=15;
  //img->Height=g+g/5; img->Width=g+g/5;
  img->Canvas->Pen->Color=clWhite;
  img->Canvas->Brush->Color=clWhite;
  img->Canvas->Rectangle(0,0,g+g/5,g+g/5);
  img->Canvas->Pen->Color=clBlack;
  img->Canvas->Pen->Width=2;
  img->Canvas->MoveTo(g/10,0);
  img->Canvas->LineTo(g/10,g+g/10);
  img->Canvas->MoveTo(0,g+g/20);
  img->Canvas->LineTo(g+g/5,g+g/20);
  img->Canvas->Pen->Width=2;
  for(int i=g+g/20;i>0;i=i-g/10)
  { //Oy katta chiziq
    img->Canvas->MoveTo(g/10-g/75,i);
    img->Canvas->LineTo(g/10+g/75,i) ;
  }
  for(int i=g/10;i<g+g/5;i=i+g/10)
  { //Ox katta chiziq
    img->Canvas->MoveTo(i,g+g/20-g/75);
    img->Canvas->LineTo(i,g+g/20+g/75) ;
  }
  img->Canvas->Pen->Width=0.5;
  img->Canvas->MoveTo(g/10,g/20);
  img->Canvas->LineTo(g+g/10,g/20);
  img->Canvas->LineTo(g+g/10,g+g/20);
  for(int i=g;i>0;i=i-g/10)
  { //Oy kichik chiziq
    img->Canvas->MoveTo(g/10-g/100,i);
    img->Canvas->LineTo(g/10+g/100,i) ;
  }
  for(int i=g/10+g/20;i<g+g/5;i=i+g/10)
  { //Ox kichik chiziq
    img->Canvas->MoveTo(i,g+g/20-g/100);
    img->Canvas->LineTo(i,g+g/20+g/100) ;
  }
}

```

```

img->Canvas->Pen->Width=2;
    img->Canvas->MoveTo(g/10,0);
    img->Canvas->LineTo(g/10-g/40,g/40) ;
    img->Canvas->MoveTo(g/10,0);
    img->Canvas->LineTo(g/10+g/40,g/40) ;

    img->Canvas->MoveTo(g+g/5-g/40,g+g/40);
    img->Canvas->LineTo(g+g/5,g+g/20) ;
    img->Canvas->MoveTo(g+g/5-g/40,g+g/20+g/40);
    img->Canvas->LineTo(g+g/5,g+g/20) ;
}

//*****
void Kohonen_map::set_weight()
{
    w=new double* [k];
    w1=new double* [k];
    kls=new double* [k];
    for(i=0;i<k;i++){w [i]=new double [m]; w1 [i]=new double [m];kls [i]=new
double [m];}
    Randomize();
    for(i=0;i<k;i++)
        for(j=0;j<m;j++){w [i] [j]=1+random(89); w [i] [j]/=100; Randomize();
w1 [i] [j]=w [i] [j];}

    Form1->StringGrid2->ColCount=m+1; // StringGrid2 ni ustunlar sonini
aniqlash
    Form1->StringGrid2->RowCount=k+1;

    Form1->StringGrid3->ColCount=m+1; // StringGrid3 ni ustunlar sonini
aniqlash
    Form1->StringGrid3->RowCount=k+1;

    Form1->StringGrid4->ColCount=m+3; // klaster tahlilni ustunlar sonini
aniqlash
    Form1->StringGrid4->RowCount=k+1;
    Form1->StringGrid3->Visible=true;
    Form1->StringGrid2->Visible=true;
    Form1->StringGrid2->Cells [0] [0]="i \ j";
    Form1->StringGrid3->Cells [0] [0]="i \ j";
    for(i=0;i<k;i++)
    {
        Form1->StringGrid2->Cells [0] [i+1]=i+1;
        Form1->StringGrid3->Cells [0] [i+1]=i+1;
    }
}

```

```

        for(i=0;i<m;i++)
        {
            Form1->StringGrid2->Cells [i+1] [0]=i+1;
            Form1->StringGrid3->Cells [i+1] [0]=i+1;
        }
    for(i=0;i<k;i++)
        for(j=0;j<m;j++){Form1->StringGrid2->Cells [j+1] [i+1]=w1 [i] [j];
            Form1->StringGrid3->Cells [j+1] [i+1]=w [i] [j];}
    wt=true;

}
//-----

void Kohonen_map::set_weight1()
{
    fclose(f);
    fclose(f2);
    f=fopen("Boshlang'ich vazn.txt","wt");
    f1=fopen("Natija vazn.txt","wt");
    fprintf(f1,"%i %i\n",k,m);
    fprintf(f,"%i %i\n",k,m);
    for(i=0;i<k;i++)
    {
        for(j=0;j<m;j++)
        {
            fprintf(f,"%f ",w1 [i] [j]);
            fprintf(f1,"%f ",w [i] [j]);
        }
        fprintf(f1,"\n");
        fprintf(f,"\n");
    }
    fclose(f);
    fclose(f1);
}
//-----

//Evkilid masofa ||x-w [j]||
double Kohonen_map::Evklid_distanse(int x_i, int w_j)
{
    double d_j=0;
    for(i=0;i<m;i++){d_j+=pow((x [x_i] [i]-w [w_j] [i]),2);}
    d_j=sqrt(d_j); //Form1->StringGrid2->Cells [w_j+1] [0]=d_j;
    return d_j;
}
//-----

```

```

void Kohonen_map::Set_Winner_neyron(int x_i)
{
    int winner;
    double min;   t=3 ;
    for(int h=0;h<60;h++){ min=Min_d_j(x_i,winner);
    for(j=0;j<m;j++)
    {
        w [winner] [j]=w [winner] [j]+nyu(t)*(x [x_i] [j]-w [winner] [j]);

    } t=t-0.05;}

}
//-----
void Kohonen_map::Klastrizatsiya()
{
    bool y=false;   int s=0;
        while(!y){
            for(i=0;i<k;i++){y=false; s=0;
            for(j=0;j<m;j++)
                if(w [i] [j]==w1 [i] [j])s++;
                if(s==m)
                {
                    y=false;
                    set_weight();
                    for(int g=0;g<n;g++)
                        Set_Winner_neyron(g);
                }
                else y=true;
            }
        }

        for(i=0;i<k;i++)
            for(j=0;j<m;j++)Form1->StringGrid3->Cells [j+1] [i+1]=w [i] [j];
        int p;
        for(int i=0;i<n;i++)
        { Min_d_j(i,p);
          K [i]=p+1;
          Form1->StringGrid1->Cells [m+1] [i+1]=K [i];

        }

}
//-----
void Kohonen_map::Klaster_tahlil()
{ float *s,*kk;

```



```

kk=new float [k]; s=new float [k];
Form1->StringGrid4->Cells [0] [0]="1 Klaster";
Form1->StringGrid4->Cells [1] [0]="Klaster o'lchami";
Form1->StringGrid4->Cells [6] [0]="Izoh";
for(i=0;i<k;i++){Form1->StringGrid4->Cells [0] [i+1]=i+1; kk [i]=0;}
for(i=0;i<m;i++)Form1->StringGrid4->Cells [i+2] [0]=IntToStr(i+1)+"-
alomat o'rtacha";

```

```

for(i=0;i<n;i++)
{
if(K [i]==1)kk [0]++;
if(K [i]==2)kk [1]++;
if(K [i]==3)kk [2]++;
}
for(j=0;j<m;j++)
{
for(i=0;i<k;i++)s [i]=0;
for(i=0;i<n;i++)
{
if(K [i]==1)s [0]+=x [i] [j];
if(K [i]==2)s [1]+=x [i] [j];
if(K [i]==3)s [2]+=x [i] [j];
}
for(i=0;i<k;i++)
{
Form1->StringGrid4->Cells [1] [i+1]=kk [i];
Form1->StringGrid4->Cells [j+2] [i+1]=10*s [i]/kk [i];
} }
}
//-----

```

```

void Kohonen_map::Klaster_tahlil_file()
{
f2=fopen("Klaster tahli natijasi.txt", "wt");
for(i=0;i<k+1;i++)
{fprintf(f2, "\n");
for(j=0;j<m+3;j++)
if(Form1->StringGrid4->Cells [j] [i].Length() != 0)
{
if(i != 0)
fprintf(f2, "\t%.8s\t", Form1->StringGrid4->Cells [j] [i]);
if(i == 0)
fprintf(f2, "%s\t", Form1->StringGrid4->Cells [j] [i]);
} }
fclose(f2);

```

```

}
//-----
void Kohonen_map::Klaster_chiz(TImage *Imagen,int xi,int yi,float g,int
wn,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2)
{ TImage *img=Imagen;
//g - ulcham;

g=g-g/20; img->Canvas->Pen->Color=clBlack;
float xmax,xmin,ymax,ymin,xmk,ymk; i=0;
img->Canvas->Pen->Width=1;
if(wn==1)
{
xmax=x [0] [xi]*w [K [i]-1] [xi]; xmin=x [0] [xi]*w [K [i]-1] [xi];
ymax=x [0] [yi]*w [K [i]-1] [yi]; ymin=x [0] [yi]*w [K [i]-1] [yi];
for(i=0;i<n;i++)
{
if(xmax<x [i] [xi]*w [K [i]-1] [xi])xmax=x [i] [xi]*w [K [i]-1] [xi];
if(xmin>x [i] [xi]*w [K [i]-1] [xi])xmin=x [i] [xi]*w [K [i]-1] [xi];
if(ymax<x [i] [yi]*w [K [i]-1] [yi])ymax=x [i] [yi]*w [K [i]-1] [yi];
if(ymin>x [i] [yi]*w [K [i]-1] [yi])ymin=x [i] [yi]*w [K [i]-1] [yi];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);

for(i=0;i<n;i++){
if(K [i]==1){img->Canvas->Brush->Color=Form1->ColorBox1->Selected; }

if(K [i]==2){img->Canvas->Brush->Color=Form1->ColorBox2->Selected;}
if(K [i]==3){img->Canvas->Brush->Color=Form1->ColorBox3->Selected; }
if(K [i]==4){img->Canvas->Brush->Color=Form1->ColorBox4->Selected; }
img->Canvas->Ellipse(g/10+(x [i] [xi]*w [K [i]-1] [xi]-xmin)*xmk-5,g-(x [i]
[yi]*w [K [i]-1] [yi]-ymin)*ymk+g/10-5,g/10+(x [i] [xi]*w [K [i]-1] [xi]-
xmin)*xmk+5,g-(x [i] [yi]*w [K [i]-1] [yi]-ymin)*ymk+5+g/10);
}
}
if(wn==2)
{
xmax=x [0] [xi]*w1 [K [i]-1] [xi]; xmin=x [0] [xi]*w1 [K [i]-1] [xi];
ymax=x [0] [yi]*w1 [K [i]-1] [yi]; ymin=x [0] [yi]*w1 [K [i]-1] [yi];
for(i=0;i<n;i++)
{
if(xmax<x [i] [xi]*w1 [K [i]-1] [xi])xmax=x [i] [xi]*w1 [K [i]-1] [xi];
if(xmin>x [i] [xi]*w1 [K [i]-1] [xi])xmin=x [i] [xi]*w1 [K [i]-1] [xi];
if(ymax<x [i] [yi]*w1 [K [i]-1] [yi])ymax=x [i] [yi]*w1 [K [i]-1] [yi];

```

```

if(ymin>x [i] [yi]*w1 [K [i]-1] [yi])ymin=x [i] [yi]*w1 [K [i]-1] [yi];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);
for(i=0;i<n;i++){
if(K [i]==1){img->Canvas->Brush->Color=Form1->ColorBox1->Selected; }

if(K [i]==2){img->Canvas->Brush->Color=Form1->ColorBox2->Selected;}
if(K [i]==3){img->Canvas->Brush->Color=Form1->ColorBox3->Selected; }
if(K [i]==4){img->Canvas->Brush->Color=Form1->ColorBox4->Selected; }
img->Canvas->Ellipse(g/10+(x [i] [xi]*w1 [K [i]-1] [xi]-xmin)*xmk-5,g-(x [i]
[yi]*w1 [K [i]-1] [yi]-ymin)*ymk+g/10-5,g/10+(x [i] [xi]*w1 [K [i]-1] [xi]-
xmin)*xmk+5,g-(x [i] [yi]*w1 [K [i]-1] [yi]-ymin)*ymk+5+g/10);
}
}
Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true; Ex2->Visible=true;
Ey1->Visible=true; Ey2->Visible=true;

};
void Kohonen_map::Klaster_chiz1(TImage *Imagen,int x1,int x2,int y1,int
y2,float g,int wn,TLabel *Ex1, TLabel *Ex2,TLabel *Ey1, TLabel *Ey2)
{ TImage *img=Imagen;
//g - ulcham;

g=g-g/20; img->Canvas->Pen->Color=clBlack;
float xmax,xmin,ymax,ymin,xmk,ymk; i=0;
img->Canvas->Pen->Width=1;
if(wn==1)
{
xmax=x [0] [x1]*w [K [i]-1] [x1]+x [0] [x2]*w [K [i]-1] [x2];
xmin=x [0] [x1]*w [K [i]-1] [x1]+x [0] [x2]*w [K [i]-1] [x2];
ymax=x [0] [y1]*w [K [i]-1] [y1]+x [0] [y2]*w [K [i]-1] [y2];
ymin=x [0] [y1]*w [K [i]-1] [y1]+x [0] [y2]*w [K [i]-1] [y2];
for(i=0;i<n;i++)
{
if(xmax<x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1] [x2])
xmax=x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1] [x2];

```

```

if(xmin>x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1] [x2])
    xmin=x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1] [x2];

if(ymax<x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2])
    ymax=x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2];
if(ymin>x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2])
    ymin=x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);

for(i=0;i<n;i++){
if(K [i]==1){img->Canvas->Brush->Color=Form1->ColorBox1->Selected; }

if(K [i]==2){img->Canvas->Brush->Color=Form1->ColorBox2->Selected;}
if(K [i]==3){img->Canvas->Brush->Color=Form1->ColorBox3->Selected; }
if(K [i]==4){img->Canvas->Brush->Color=Form1->ColorBox4->Selected; }
img->Canvas->Ellipse(g/10+(x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1]
[x2]-xmin)*xmk-5,g-(x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2]-
ymin)*ymk+g/10-5,g/10+(x [i] [x1]*w [K [i]-1] [x1]+x [i] [x2]*w [K [i]-1]
[x2]-xmin)*xmk+5,g-(x [i] [y1]*w [K [i]-1] [y1]+x [i] [y2]*w [K [i]-1] [y2]-
ymin)*ymk+5+g/10);
    }
}
if(wn==2)
{
xmax=x [0] [x1]*w1 [K [i]-1] [x1]+x [0] [x2]*w1 [K [i]-1] [x2];
xmin=x [0] [x1]*w1 [K [i]-1] [x1]+x [0] [x2]*w1 [K [i]-1] [x2];
ymax=x [0] [y1]*w1 [K [i]-1] [y1]+x [0] [y2]*w1 [K [i]-1] [y2];
ymin=x [0] [y1]*w1 [K [i]-1] [y1]+x [0] [y2]*w1 [K [i]-1] [y2];
for(i=0;i<n;i++)
{
if(xmax<x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K [i]-1] [x2])
    xmax=x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K [i]-1] [x2];
if(xmin>x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K [i]-1] [x2])
    xmin=x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K [i]-1] [x2];

if(ymax<x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1] [y2])
    ymax=x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1] [y2];
if(ymin>x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1] [y2])
    ymin=x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1] [y2];
}
xmk=g/(xmax-xmin) ;
ymk=g/(ymax-ymin);
}

```

```

for(i=0;i<n;i++){
if(K [i]==1){img->Canvas->Brush->Color=Form1->ColorBox1->Selected; }

if(K [i]==2){img->Canvas->Brush->Color=Form1->ColorBox2->Selected;}
if(K [i]==3){img->Canvas->Brush->Color=Form1->ColorBox3->Selected; }
if(K [i]==4){img->Canvas->Brush->Color=Form1->ColorBox4->Selected; }
img->Canvas->Ellipse(g/10+(x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K [i]-
1] [x2]-xmin)*xmk-5,g-(x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1]
[y2]-ymin)*ymk+g/10-5,g/10+(x [i] [x1]*w1 [K [i]-1] [x1]+x [i] [x2]*w1 [K
[i]-1] [x2]-xmin)*xmk+5,g-(x [i] [y1]*w1 [K [i]-1] [y1]+x [i] [y2]*w1 [K [i]-1]
[y2]-ymin)*ymk+5+g/10);
}
}
Ex1->Caption=FloatToStr(xmin*10);
Ex2->Caption=FloatToStr(xmax*10);
Ey1->Caption=FloatToStr(ymin*10);
Ey2->Caption=FloatToStr(ymax*10);
Ex1->Height=15; Ex1->Width=25;
Ex2->Height=15; Ex2->Width=25;
Ey1->Height=15; Ey1->Width=25;
Ey2->Height=15; Ey2->Width=25;
Ex1->Visible=true; Ex2->Visible=true;
Ey1->Visible=true; Ey2->Visible=true;

};
//-----
double Kohonen_map::Min_d_j(int x_i,int &d_j)
{
double min_d_j=1000,*dj; dj=new double [k];
for(j=0;j<k;j++)
{
dj [j]=Evklid_distanse(x_i,j);
if(min_d_j>dj [j]){ min_d_j=dj [j]; d_j=j; }
}
return min_d_j;
}
//-----
void Kohonen_map::set_X_vector(String fn)
{
FileName=fn;
wt=false;
f=fopen(fn.c_str(),"rt"); //tanlanma.txt fayli ochildi
f1=fopen("NATIJA.txt","wt");

```

```

//n tanlanmalar soni
fscanf(f, "%i ", &n); Form1->LN->Caption="Tanlanmalar soni n =
"+IntToStr(n);
//m har bir tanlanmadagi alomatlar soni
fscanf(f, "%i ", &m); Form1->LM->Caption="Alomatlar soni n =
"+IntToStr(m);
//sinflar soni
fscanf(f, "%i ", &k); Form1->LK->Caption="Sinflar soni n = "+IntToStr(k);

```

```

x=new double* [n],K=new int [n]; String a;
char b [100]; String k1 ;
for(int i=0;i<n;i++)x [i]=new double [m];
double z;

```

```

Form1->StringGrid1->ColCount=m+2; // StringGrid1 ni ustunlar sonini
aniqlash
Form1->StringGrid1->RowCount=n+1; // StringGrid1 ni satrlar sonini aniqlash

```

```

//Form2->StringGrid1->ColCount=m+2; // StringGrid1 ni ustunlar sonini
aniqlash
//Form2->StringGrid1->RowCount=n+1; // StringGrid1 ni satrlar sonini
aniqlash

```

```

Form1->StringGrid1->Refresh();
Form1->StringGrid1->Refresh();
for(int i=0;i<n+1;i++)
{
Form1->StringGrid1->Cells [m+1] [i]="";
//Form2->StringGrid1->Cells [m+1] [i]="";
}
for(int i=0;i<n;i++)
{ Form1->StringGrid1->Cells [0] [i+1]=i+1;
// Form2->StringGrid1->Cells [0] [i+1]=i+1;
//i tanlanma faylda o'qib olinyapti
fgets(b,100,f); a=b;
for(int j=0;j<m;j++)
{ Form1->StringGrid1->Cells [j+1] [0]=IntToStr(j+1)+"-alomat";

if(a.Pos(" ")>0)
{
k1=a.SubString(0,a.Pos(" ")-1);
a=a.SubString(a.Pos(" ")+1,a.Length());
x [i] [j]=k1.ToDouble();
Form1->StringGrid1->Cells [j+1] [i+1]=x [i] [j];
}
}
}

```

```

    }
    else if(a.Pos(" ")==0)
    {
        k1=a.SubString(1,a.Length()-1); x [i] [j]=k1.ToDouble();
        Form1->StringGrid1->Cells [j+1] [i+1]=x [i] [j];
    }
}   Form1->StringGrid1->Cells [m+1] [0]="Klaster";

}   for(i=0;i<n;i++)K [i]=0;
fclose(f);
}

void __fastcall TForm1::BTanlanma_openClick(TObject *Sender)
{
//   if(OpenDialog1->Execute())
//FileName=OpenDialog1->FileName;

    for(int i=0;i<KS.getn()+1;i++)
    for(int j=0;j<KS.getm()+2;j++)
    {
        Form1->StringGrid1->Cells [j] [i]="";
    }
    /*for(int i=0;i<KS.getk()+1;i++)
    for(int j=0;j<KS.getm()+1;j++){Form1->StringGrid2->Cells [j] [i]="";
    Form1->StringGrid3->Cells [j] [i]=""; }
    Button2->Enabled=true; */
    FileName="Tanlanma iris.txt";
    // FileName="Tanlanma iris aniq.txt";
    ETanlanma->Text=FileName;

    KS.set_X_vector(FileName);
    Form1->Button2->Enabled=true;
    Form1->Button3->Enabled=false;
    Form1->Button4->Enabled=false;
    Form1->Button5->Enabled=false;

}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
//Klastrizatsiya tugamasi
KS.set_weight();
for(int i=0;i<KS.getn();i++)

```

```

KS.Set_Winner_neyron(i); KS.Klastrizatsiya();
Form1->PKlaster_tahlik->Visible=false;
//Form1->PKlaster_vizual1->Visible=true;
//Form1->PKlaster_vizual2->Visible=true;
Form1->PKohonen_xritasi->Visible=false;
//Form1->PTanlanma_open->Visible=false;
//Form1->PTanlanma_1->Visible=false;
Form1->Button2->Enabled=true;
Form1->Button3->Enabled=true;
Form1->Button4->Enabled=true;
Form1->Button5->Enabled=true;
StringGrid2->Visible=true;
StringGrid3->Visible=true;
Label2->Visible=true;
Label3->Visible=true;

}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
KS.Klaster_tahlil();

Form1->PKlaster_tahlik->Left=5;
Form1->PKlaster_tahlik->Top=50;
Form1->PKlaster_tahlik->Height=450;
Form1->PKlaster_tahlik->Width=1330;
Form1->PKlaster_tahlik->Visible=true;
Form1->PKlaster_vizual1->Visible=false;
Form1->PKlaster_vizual2->Visible=false;
Form1->PKohonen_xritasi->Visible=false;
Form1->PTanlanma_open->Visible=false;
Form1->PTanlanma_1->Visible=false;
}
//-----

void __fastcall TForm1::Button7Click(TObject *Sender)
{
KS.Klaster_tahlil_file();
}
//-----

```



```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
Form1->PTanlanma_open->Visible=true;
Form1->PTanlanma_1->Visible=true;
Form1->PTanlanma_open->Left=5;
Form1->PTanlanma_open->Top=50;
Form1->PTanlanma_open->Width=250;
Form1->PTanlanma_open->Height=450;
Form1->PTanlanma_1->Left=260;
Form1->PTanlanma_1->Top=50;
Form1->PTanlanma_1->Height=450;
Form1->PTanlanma_1->Width=1075;

Form1->PKlaster_tahlik->Visible=false;
Form1->PKlaster_vizual1->Visible=false;
Form1->PKlaster_vizual2->Visible=false;
Form1->PKohonon_xritasi->Visible=false;
Form1->PTanlanma_open->Visible=true;
Form1->PTanlanma_1->Visible=true;
Form1->Button2->Enabled=false;
Form1->Button3->Enabled=false;
Form1->Button4->Enabled=false;
Form1->Button5->Enabled=false;

}
//-----

void __fastcall TForm1::Button4Click(TObject *Sender)
{
RadioGroup1->ItemIndex=-1;
Form1->PKlaster_tahlik->Visible=false;
Form1->PKlaster_vizual1->Left=5;
Form1->PKlaster_vizual1->Top=50;
Form1->PKlaster_vizual1->Width=225;
Form1->PKlaster_vizual1->Height=450;
Form1->PKlaster_vizual1->Visible=true;

```

```

Form1->PKlaster_vizual2->Left=235;
Form1->PKlaster_vizual2->Top=50;
Form1->PKlaster_vizual2->Width=1100;
Form1->PKlaster_vizual2->Height=450;
Form1->PKlaster_vizual2->Visible=true;
Image1->Visible=false; Image3->Visible=false; Image2->Visible=false;
Label10->Visible=false; Label11->Visible=false; Label12->Visible=false;
Label13->Visible=false; Label14->Visible=false; Label15->Visible=false;
Label16->Visible=false; Label17->Visible=false; Label18->Visible=false;
Label19->Visible=false; Label20->Visible=false; Label21->Visible=false;
Label22->Visible=false; Label23->Visible=false; Label24->Visible=false;
Button6->Visible=false; Button9->Visible=false; Button10-
>Visible=false;
b=false; kkk=true ;
b=true; kkk=false;
Form1->PKlaster_tahlik->Visible=false;
Form1->PKlaster_vizual1->Visible=true;
Form1->PKlaster_vizual2->Visible=true;
Form1->PKohonen_xritasi->Visible=false;
Form1->PTanlanma_open->Visible=false;
Form1->PTanlanma_1->Visible=false;
}
//-----

```

```

void __fastcall TForm1::Button6Click(TObject *Sender)
{
if(b==true)
{
b1=true; b2=false; b3=false;
k1=false; k2=false; k3=false;
if(RadioGroup1->ItemIndex==0)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,0,1,300,1,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,0,2,300,1,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,0,3,300,1,Label18,Label19,Label20,Label21);
Label22->Caption="1 va 2 alomatlar ";
Label23->Caption="1 va 3 alomatlar ";
Label24->Caption="1 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
}
else if(RadioGroup1->ItemIndex==1)
{

```

```

KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,0,1,300,2,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,0,2,300,2,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,0,3,300,2,Label18,Label19,Label20,Label21);

}
}
if(kkk==true)
{ k1=true; b2=false; b3=false;
  b1=false; k2=false; k3=false;
int fn=Form1->ComboBox1->ItemIndex+1;

  if(RadioGroup2->ItemIndex==0)
  {

KS.Kohonen_xaritasi(Image1,0,1,300,1,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,0,2,300,1,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,0,3,300,1,fn,Label18,Label19,Label20,Label21,7
35,5);
}
if(RadioGroup2->ItemIndex==1)
{

KS.Kohonen_xaritasi(Image1,0,1,300,2,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,0,2,300,2,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,0,3,300,2,fn,Label18,Label19,Label20,Label21,7
35,5);
}
}
Label22->Caption="1 va 2 alomatlar ";
Label23->Caption="1 va 3 alomatlar ";
Label24->Caption="1 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
}
//-----

```

```

void __fastcall TForm1::FormActivate(TObject *Sender)
{
Form1->Left=0;
Form1->Top=50;
Form1->PTanlanma_open->Visible=true;
Form1->PTanlanma_1->Visible=true;
Form1->PTanlanma_open->Left=5;
Form1->PTanlanma_open->Top=50;
Form1->PTanlanma_open->Width=250;
Form1->PTanlanma_open->Height=450;
Form1->PTanlanma_1->Left=260;
Form1->PTanlanma_1->Top=50;
Form1->PTanlanma_1->Height=450;
Form1->PTanlanma_1->Width=1075;

Form1->PKlaster_tahlik->Visible=false;
Form1->PKlaster_vizual1->Visible=false;
Form1->PKlaster_vizual2->Visible=false;
Form1->PKohonen_xritasi->Visible=false;
Form1->PTanlanma_open->Visible=true;
Form1->PTanlanma_1->Visible=true;
Form1->Button2->Enabled=false;
Form1->Button3->Enabled=false;
Form1->Button4->Enabled=false;
Form1->Button5->Enabled=false;
}
//-----

void __fastcall TForm1::Button9Click(TObject *Sender)
{
Label22->Caption="2 va 3 alomatlar ";
Label23->Caption="2 va 4 alomatlar ";
Label24->Caption="3 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
if(b=true){
b2=true; b1=false; b3=false;
if(RadioGroup1->ItemIndex==0)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,1,2,300,1,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,1,3,300,1,Label14,Label15,Label16,Label17);
}
}
}

```

```

KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,2,3,300,1,Label18,Label19,Label20,Label21);

}
else if(RadioGroup1->ItemIndex==1)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,1,2,300,2,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,1,3,300,2,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,2,3,300,2,Label18,Label19,Label20,Label21);

}}

if(kkk==true)
{ k2=true; b2=false; b3=false;
  b1=false; k1=false; k3=false;
int fn=Form1->ComboBox1->ItemIndex+1;

  if(RadioGroup2->ItemIndex==0)
  {

KS.Kohonen_xaritasi(Image1,1,2,300,1,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,1,3,300,1,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,1,3,300,1,fn,Label18,Label19,Label20,Label21,7
35,5);
}
if(RadioGroup2->ItemIndex==1)
{

KS.Kohonen_xaritasi(Image1,1,2,300,2,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,1,3,300,2,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,2,3,300,2,fn,Label18,Label19,Label20,Label21,7
35,5);
}
}
}

```

```

}
//-----

void __fastcall TForm1::RadioGroup1Click(TObject *Sender)
{
//+++++
Image1->Visible=true;
Image2->Visible=true;
Image3->Visible=true;
if(b1==true)
{
if(RadioGroup1->ItemIndex==0)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,0,1,300,1,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,0,2,300,1,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,0,3,300,1,Label18,Label19,Label20,Label21);
Label22->Caption="1 va 2 alomatlar ";
Label23->Caption="1 va 3 alomatlar ";
Label24->Caption="1 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
}
else if(RadioGroup1->ItemIndex==1)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,0,1,300,2,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,0,2,300,2,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,0,3,300,2,Label18,Label19,Label20,Label21);
Label22->Caption="1 va 2 alomatlar ";
Label23->Caption="1 va 3 alomatlar ";
Label24->Caption="1 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
} }
//+++++
else if(b2==true)
{
if(RadioGroup1->ItemIndex==0)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,1,2,300,1,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);

```

```

KS.Klaster_chiz(Image2,1,3,300,1,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,2,3,300,1,Label18,Label19,Label20,Label21);
Label22->Caption="2 va 3 alomatlar ";
Label23->Caption="2 va 4 alomatlar ";
Label24->Caption="3 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
}
else if(RadioGroup1->ItemIndex==1)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz(Image1,1,2,300,2,Label10,Label11,Label12,Label13);
KS.koordinata_oqi(Image2,300,370,5,Label14,Label15,Label16,Label17);
KS.Klaster_chiz(Image2,1,3,300,2,Label14,Label15,Label16,Label17);
KS.koordinata_oqi(Image3,300,735,5,Label18,Label19,Label20,Label21);
KS.Klaster_chiz(Image3,2,3,300,2,Label18,Label19,Label20,Label21);
Label22->Caption="2 va 3 alomatlar ";
Label23->Caption="2 va 4 alomatlar ";
Label24->Caption="3 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
}
}
//+++++
else if(b3==true)
{
Label23->Visible=false; Label24->Visible=false;
Label14->Visible=false; Label18->Visible=false;
Label15->Visible=false; Label19->Visible=false;
Label16->Visible=false; Label20->Visible=false;
Label17->Visible=false; Label21->Visible=false;
Image2->Visible=false; Image3->Visible=false;
Label22->Caption="KLAUSTERIZATSIYA";
Label22->Visible=true; Label23->Visible=false; Label24->Visible=false;
if(RadioGroup1->ItemIndex==0)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz1(Image1,3,2,1,0,300,1,Label10,Label11,Label12,Label13);
}
else if(RadioGroup1->ItemIndex==1)
{
KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
KS.Klaster_chiz1(Image1,3,2,1,0,300,2,Label10,Label11,Label12,Label13);
}
}
}
Button6->Visible=true;

```

```
Button9->Visible=true;
Button10->Visible=true;
```

```
}
//-----
```

```
void __fastcall TForm1::Button10Click(TObject *Sender)
{
    Label23->Visible=false; Label24->Visible=false;
    Label14->Visible=false; Label18->Visible=false;
    Label15->Visible=false; Label19->Visible=false;
    Label16->Visible=false; Label20->Visible=false;
    Label17->Visible=false; Label21->Visible=false;
    Image2->Visible=false; Image3->Visible=false;
    Image1->Visible=false;
    Label22->Caption="KLAsterIZATSIYA";
    if(b==true)
    {
        Image1->Visible=true;

        b3=true; b2=false; b1=false;
        if(RadioGroup1->ItemIndex==0)
        {
            KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
            KS.Klaster_chiz1(Image1,3,2,1,0,300,1,Label10,Label11,Label12,Label13);
        }
        else if(RadioGroup1->ItemIndex==1)
        {
            KS.koordinata_oqi(Image1,300,5,5,Label10,Label11,Label12,Label13);
            KS.Klaster_chiz1(Image1,3,2,1,0,300,2,Label10,Label11,Label12,Label13);
        }
    }

    if(kkk==true)
    {
        k3=true; b2=false; b3=false;
        b1=false; k1=false; k2=false;
        int fn=Form1->ComboBox1->ItemIndex+1;

        if(RadioGroup2-
>ItemIndex==0)KS.Kohonen_xaritasi1(Image1,3,2,1,0,300,1,fn,Label10,Label1
1,Label12,Label13,5,5);
```



```

    if(RadioGroup2-
>ItemIndex==1)KS.Kohonen_xaritasi1(Image1,3,2,1,0,300,2,fn,Label10,Label1
1,Label12,Label13,5,5);

}
}
//-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{
Form1->PKohonen_xritasi->Visible=true;
Form1->PKlaster_tahlik->Visible=false;
Form1->PKlaster_vizual1->Visible=false;
Form1->PKlaster_vizual2->Visible=true;
Form1->PTanlanma_open->Visible=false;
Form1->PTanlanma_1->Visible=false;
Image1->Visible=false; Image3->Visible=false; Image2->Visible=false;
Label10->Visible=false; Label11->Visible=false; Label12->Visible=false;
Label13->Visible=false; Label14->Visible=false; Label15->Visible=false;
Label16->Visible=false; Label17->Visible=false; Label18->Visible=false;
Label19->Visible=false; Label20->Visible=false; Label21->Visible=false;
Label22->Visible=false; Label23->Visible=false; Label24->Visible=false;
Button6->Visible=false; Button9->Visible=false; Button10-
>Visible=false;
b=false; kkk=true ; k1=true; k2=false; k3=false;
RadioGroup2->ItemIndex=-1;
Form1->PKohonen_xritasi->Left=5;
Form1->PKohonen_xritasi->Top=50;
Form1->PKohonen_xritasi->Width=225;
Form1->PKohonen_xritasi->Height=450;
Form1->PKohonen_xritasi->Visible=true;
Form1->PKlaster_vizual2->Left=235;
Form1->PKlaster_vizual2->Top=50;
Form1->PKlaster_vizual2->Width=1100;
Form1->PKlaster_vizual2->Height=450;
Form1->PKlaster_vizual2->Visible=true;
}
//-----

void __fastcall TForm1::RadioGroup2Click(TObject *Sender)
{ int fn=Form1->ComboBox1->ItemIndex+1;

if(k1==true)
{
Button6->Visible=true;

```

```

Button9->Visible=true;
Button10->Visible=true;
Label22->Caption="1 va 2 alomatlar ";
Label23->Caption="1 va 3 alomatlar ";
Label24->Caption="1 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
  if(RadioGroup2->ItemIndex==0)
  {

KS.Kohonen_xaritasi(Image1,0,1,300,1,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,0,2,300,1,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,0,3,300,1,fn,Label18,Label19,Label20,Label21,7
35,5);
  }
  if(RadioGroup2->ItemIndex==1)
  {

KS.Kohonen_xaritasi(Image1,0,1,300,2,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,0,2,300,2,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,0,3,300,2,fn,Label18,Label19,Label20,Label21,7
35,5);
  }
  }

else if(k2==true)
{ Button6->Visible=true;
Button9->Visible=true;
Button10->Visible=true;
Label22->Caption="2 va 3 alomatlar ";
Label23->Caption="2 va 4 alomatlar ";
Label24->Caption="3 va 4 alomatlar ";
Label22->Visible=true; Label23->Visible=true; Label24->Visible=true;
  if(RadioGroup2->ItemIndex==0)
  {

KS.Kohonen_xaritasi(Image1,1,2,300,1,fn,Label10,Label11,Label12,Label13,5,
5);

```

```

KS.Kohonen_xaritasi(Image2,1,3,300,1,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,1,3,300,1,fn,Label18,Label19,Label20,Label21,7
35,5);
}
if(RadioGroup2->ItemIndex==1)
{

KS.Kohonen_xaritasi(Image1,1,2,300,2,fn,Label10,Label11,Label12,Label13,5,
5);

KS.Kohonen_xaritasi(Image2,1,3,300,2,fn,Label14,Label15,Label16,Label17,3
70,5);

KS.Kohonen_xaritasi(Image3,2,3,300,2,fn,Label18,Label19,Label20,Label21,7
35,5);
}
}

else if(k3==true)
{ Button6->Visible=true;
Button9->Visible=true;
Button10->Visible=true;
Label22->Caption="KLAUSTERIZATSIYA ";

Label22->Visible=true;
{

if(RadioGroup2-
>ItemIndex==0)KS.Kohonen_xaritasi1(Image1,3,2,1,0,300,1,fn,Label10,Label1
1,Label12,Label13,5,5);

if(RadioGroup2-
>ItemIndex==1)KS.Kohonen_xaritasi1(Image1,3,2,1,0,300,2,fn,Label10,Label1
1,Label12,Label13,5,5);

}
}
}

```

ILOVA 2

Tezislar



**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA‘LIM
VAZIRLIGI**

NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

**INFORMATIKA VA AXBOROT KOMMUNIKATSIYA
TEXNOLOGIYALARI TA‘LIMINI MODERNIZASİYALASH
ISTIQBOLLARI**

Respublika ilmiy-amaliy konferensiyasi materiallari

T O ` P L A M I

(2018-yil 25-may)

NAVOIY – 2018

1

MUNDARIJA

1.	SO'Z BOSHI	3
2.	AXBOROT TEXNOLOGIYALARINI JORIY QILISH VOSITALARI Ma'murov T.T., NavDPI	5
3.	KULON TAJRIBASINI TUSHUNTIRISHDA KOMPYUTER TEXNOLOGIYALARIDAN FOYDALANISH Abdullayev J. M., NavDPI	6
4.	ТАЪЛИМ ТИЗИМИДА МАСОФАЛИН ЎҚИТИШ УСЛУБИ ВА АФЗАЛЛИКЛАРИ Абдурашмонов Д.Э., ГулДУ	7
5.	MUSTAQIL TA'LIMDA ZAMONAVIY DASTURLASH TILLARINING INNOVATSION TAQBIQI Jo'raqulov T.T., Do'stov S.T., Adizova Z.M., NavDPI	9
6.	ZAMONAVIY AXBOROT TEXNOLOGIYALARIDAN FOYDALANISHNING TA'LIM SAMARADORLIGINI OSHIRISHDAGI O'RNI Boboyorova N.S., SQXI	11
7.	TABIIY FANLARNI O'QITISHDA AXBOROT TEXNOLOGIYALARIDAN FOYDALANISH MASALALARI Boboyorova N.S., SQXI	12
8.	ФОРМИРОВАНИЕ ИНФОРМАЦИОННО – КОММУНИКА- ТИВНЫХ КОМПЕТЕНЦИЙ НА УРОКАХ ХИМИИ Нематова Л., учительница школы №20 Каракульского района	13
9.	ELEKTRON AXBOROT RESURSLARINI KOMPYUTER TARMOG'IDA TAQDIM ETISH USULLARI Qilichev X., Jo'raqulov D.A., NavDPI	15
10.	KIMYO FANINI O'QITISHDA TA'LIM VOSITALARIDAN SAMARALI FOYDALANISH Hotamova M.S., NavDPI	16
11.	BOSHLANG'ICH TA'LIMDA AXBOROT TEXNOLOGIYALARI Isroilova R.S., NavDPI	17
12.	ИСПОЛЬЗОВАНИЕ ИКТ В ОБУЧЕНИИ АНГЛИЙСКОМУ ЯЗЫКУ Эргашова Д.Р., учительница школы №17, г.Наван	19
13.	AMALIY MASHG'ULOT JARAYONIDA KOMPYUTERLARDAN FOYDALANISH QOIDALARI Mahmanazarov E.E., Ne'matova R.H., NDKI akademik litsey	20
14.	ELEKTRON O'QUV-USLUBIY MAJMUALAR HAMDA TA'LIM RESRUSLARIGA BO'LGAN ASOSIY TALABLAR Mahmanazarov E.E., Jumanazarov J.M., NDKI akademik litsey	22
15.	BOSHLANG'ICH SINIF ONA TILIDARSLARINI TASHKIL ETISHDA TA'LIM METODLARINING SAMARADORLIGI Jumayeva M., NavDPI	24
16.	TA'LIM MUASSASASALARIDA AXBOROT TEXNOLOGIYALARDAN FOYDALANISHNING AFZALLIKLARI Kanatbayeva T.S., Murodova G.R., NavDPI	25
17.	TALABALAR BILIMINI BAHOLASHDA NOOSTANDART TESTLARDAN FOYDALANISH Kanatbayeva T.S., Muhammadriyeva M., NavDPI	26

79.	KOHONENNING O'Z-O'ZINI TASHKIL ETUVCHI XARITALARI MATEMATIK MODELI Ishmuhammedov A.X., Odilov J.Q., O'zMU	118
80.	TA'LIM JARAYONIDA ISPRING DASTURI VA UNING IMKONIYATLARIDAN FOYDALANISH Sidiqov M., Ortiqova N., FarDU	120
81.	KASB-HUNAR KOLLEJLARIDA INFORMATIKA VA AXBOROT TEXNOLOGIYALARI FANINI O'QITISH JARAENINI TASHKIL ETISH HAMDA UNING SAMARADORLIGINI OSHIRISH USULLARI Юлдашев Ў.А.,ГулДУ	122
82.	O'QUV JARAENIDA ELEKTRON O'QUV-USLUBII MAJMUALARINING O'RNI VA AHAMIYATI Юлдашев Ў.А.,ГулДУ	123
83.	YOSHLAR TARBIYASIDA AXBOROT MADANIYATINI SHAKLLANTIRISH Rasulov R.O., NavDPI	125
84.	UMUMTA'LIM MAKTABLARINING O'QUV JARAYONIDA MULTIMEDIA TEXNOLOGIYALARIDAN FOYDALANISH Xodjayeva Z., Inoyatova M., Karmana tuman 17-umumiy o'rta ta'lim maktabi	127
85.	MATEMATIKA DARSLARIDA MATHCAD DASTURIDAN FOYDALANISH USULLARI Jumayev S.S., Sobirov J.A., NavDPI	128
86.	TA'LIM JARAYONIDA AUTOPLAY DASTURI VA UNING IMKONIYATLARIDAN FOYDALANISH Jo'rakulov T.T., NavDPI, Rajabova D.A., Karmana tuman 17-umumiy o'rta ta'lim maktabi	130
87.	ARXITEKTURA SOHASI MUTAXASSISLARINI TAYYORLASHDA KOMPYUTER GRAFIKASINING AHAMIYATI Rasimov O.D., Sa'dullayeva F.S., NavDPI	131
88.	MATEMATIK MATNLARNI TAYYORLASH TEXNOLOGIYALARI Xidirova M., Bozorova D., NDKI akademik litsey	132
89.	UMUMTA'LIM MAKTABLARI DARS JARAYONIDA AXBOROT TEXNOLOGIYALARIDAN FOYDALANISH Sindarova M.Sh., Ulug'murodova L.D., JDPLJK	134
90.	IKKI ARGUMENTLI FUNKSIYA XUSUSIY HOSILASINING AMALIY MASALALAR YECHISHGA TATBIQI (ENG KICHIK KVADRATLAR METODI) Jumayeva N.F., NavDPI	136
91.	TA'LIM-TARBIYA JARAYONIDA KOMPYUTER TEXNOLOGIYALARI IMKONIYATLARIDAN FOYDALANISH Botirova S.R., NavDPI	138
92.	KASB-HUNAR TA'LIMIDA KIMYO FANINI AXBOROT TEXNOLOGIYALARI ASOSIDA O'QITISH MASALALARI Botirova S.R., NavDPI	140
93.	KIMYO TA'LIMIDA KOMPYUTERLI O'QITISH TEXNOLOGIYALARIDAN FOYDALANISHNING AMALIY AHAMIYATI Suyarova H.H., Qurbonova K., NavDPI	141

AKTni tadbiq qilish ta'limni differensiallash va individuallashtirish g'oyasini amalga oshirish imkonini beradi. AKT asosida ishlab chiqilga zamonaviy o'quv qo'llanmalar interfaollikka ega (o'quvchi bilan muloqot qilish qobiliyati).

O'quv jarayonida axborot kommunikatsiya texnologiyalarini tadbiq qilishni ko'raylik. Axborot texnologiyalaridan birinchi navbatda quyidagilar uchun foydalaniladi:

- o'quv jarayonini tashkillashtirish;
- o'quv qo'llanmalar tayyorlash;
- yangi o'quv materiallarini o'rganish (mustaqil taqdimot yaratish yoki tayyor dasturlardan foydalanish);
- bilimlarni kompyuterli nazorat qilish;
- Internet tarmog'idan axborotlarni olish va ular bilan ishlash;
- ota-ona, o'qituvchi va o'quvchini o'zaro bog'lash uchun sayt bilan ishlash va yaratish.

Ananaviy o'qitish metodlarida kompyuter texnologiyalaridan samarali foydalanish juda muhim. Bunda albatta o'quvchining sog'lig'ini himoya qilish shartlarini hisobga olish lozim.

Shuni ta'kidlab o'tish lozimki, axborot-kommunikatsiya texnologiyalaridan foydalanishda qo'shimcha tayyorgarlik ko'rish uchun sarflanadigan vaqt so'zsiz ortadi. Ammo, asta sekinlik bilan metodik baza shakllanadi, bu esa keyingi tayyorgarliklarni sezilarli darajada yengillashtiradi.

Zamonaviy texnologiyalar va kommunikatsiyalar o'quv-tarbiya jarayonini tashkillashtirishni o'zgartirish imkonini beradi: ta'lim oluvchini axborot muhiti bilan to'la band qilish, ta'lim sifatini oshirish, axborotni qabul qilish jarayoni va bilimlarni o'zlashtirishni motivatsiyalash. Yangi axborot texnologiyalari turli faoliyat sohaslarida, shu bilan birga ta'lim tizimida ham, boshqaruv va tashkillashtirishda kompyuterli va telekommunikatsion muhitni yaratadi. Axborot texnologiyalarining ta'lim dasturlariga integratsiyasi barcha darajalarda: maktab, oliy ta'lim va oliy ta'limdan keyingi ta'limda amalga oshirilmoqda.

Adabiyotlar:

1. Вильямс, Р., Маклин К. Компьютер в школе - М.: Просвещение, 2008.
2. Ганичева, Е.М. Повышение качества подготовки школьников с применением информационных технологий / Е.М. Ганичева. - М.: 2007.
3. Иванов, Д. Компетентности и компетентностный подход в современном образовании. -М.: Чистые пруды, 2007. (Библиотека «Первого сентября», серия «Воспитание. Образование. Педагогика.» Вып.6(12)).

KOHONENNING O'Z-O'ZINI TASHKIL ETUVCHI XARITALARI MATEMATIK MODEL

Ishmuhannomedov A.X., Odilov J.Q., O'zMU

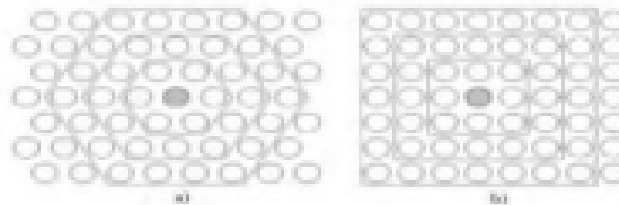
O'z-o'zini tashkil etuvchi xaritalar neyron tarmoq algoritmlarining asosiy turlaridan biri hisoblanadi. Ushbu texnologiyaning boshqa neyron tarmoqlardan asosiy farqi, o'qituvchisiz o'qitish usulini qo'llanilgan, teskari tarqatish algoritmi bilan o'rgatilgan, ya'ni o'rgatishda natija faqatgina kirish ma'lumotlariga bog'liq. Ushbu turdagi neyron tarmoqlar asosan turli masalarni yechish uchun: ma'lumotlardagi bo'shliqlarni tiklashdan, ma'lumotlarni tahlil qilish va qonunlarni topish uchun ishlatiladi, masalan, sun'iy intellekt masalalarida. Ushbu ishda biz asosi tushunchalarni va o'z-o'zini tashkil etadigan xaritalardan foydalanishning ayrim jihatlarni ko'rib chiqamiz.

O'z-o'zini tashkil qiluvchi xaritalari (Self Organizing Maps – SOM) ning ishlash algoritmi ko'p o'lehovli vektorlarni klaster taxlid variantlaridan biri hisoblanadi, ya'ni o'ziga o'xshash obyektlarni yig'adi. Bunday algoritmlarga misol sifatida "k-eng yaqin qo'shni" (k-means) algoritmini keltirish mumkin. SOM algoritmining muhim farqi shundaki, unda barcha neyronlar odatda ikki o'lehovli to'mni hosil qiladi. SOM da o'rgatish asosan "g'olib hammasini oladi"

qoidasiga ko'ra amalga oshiriladi. Shu bilan birga, o'rgatish vaqtida nafaqat neyronlarning g'oliblari, balki qo'shni bo'lganlar ham o'zgarib bormaydi, biroq u kichik darajada o'lgovli fazoga o'tkaziladi. Shu sababli, SOM ko'p o'lgovli fazodan nisbatan kichik o'lgovli fazoga o'tish usullaridan biri sifatida qaralishi mumkin. Ushbu algoritmnı qo'llash natijasida, haqiqiy xarita fazosiga o'xshash vektorlar hosil qilinadi.

SOM neyronlarning tartiblangan strukturasi qo'llashni nazarda tutadi. Odatda bir va ikki o'lgovli to'ra qo'llaniladi. Bu holda har bir neyron n -o'lgovli vektor ustuni $w = [w_1, w_2, \dots, w_n]^T$ bo'lib, bu erda n haqiqiy fazo o'lgovi (kirish vektorlarining o'lgovi) bilan aniqlanadi. Bir va ikki o'lgovli to'rlardan foydalanish yuqori o'lgovdagi kengaytmali tuzilmalarnı xaritada topishda muammolar yuzaga kelishi bilan bog'liq (bu yerda o'lgovni ikki o'lgovli o'lgovga tushirish, vizual ko'rish muammolarga bog'liq).

Odatda, neyronlar to'rtburchaklar yoki oltı burchakli hujayralarga ega bo'lgan ikki o'lgovli to'ra tugunlaridir. Bu holda, yuqorida aytib o'tilganidek, neyronlar ham bir-biri bilan ta'sir o'tkazadi. Ushbu ta'sirning kattaligi xaritadagi neyronlar orasidagi masofa bilan belgilanadi. 1-rasmda oltı burchakli va to'rtburchakli to'rlar uchun neyronlar orasidagi masofa keltirilgan.



1-rasm. Oltı burchakli va to'rtburchakli to'rlar.

Xaritada oltı burchakli (a) va to'rtburchakli (b) to'rlar uchun neyronlar orasidagi masofa. Oltı burchakli to'rdagi neyronlar orasidagi masofa to'rt burchakli to'rga qaraganda ko'proq ekanligini ko'rish oson.

Bunday holda, xaritaning yakuniy umumlashtirish qobiliyati to'rdagi neyronlar soniga qarab aniqlanadi.

Xaritani dastlabki ishga tushirish. SOM algoritmini qo'llashda to'ra konfiguratsiyasi (to'rtburchak yoki oltı burchakli), shuningdek tarmoqdagi neyronlarning soni ham oldindan belgilanadi. Ko'p hollarda xaritada maksimal mumkin bo'lgan neyron sonini qo'llashni tavsiya qiladi. Shu bilan birga, dastlabki o'qitish olingan xaritaning yordamida umumlashtirish qobiliyatiga katta ta'sir ko'rsatadi. Xaritadagi tugunlar soni o'qitish masalasida amallar sonidan ko'p bo'lgan hollarda, algoritmnı muvaffaqiyatli ishlatilishi o'rgatishning boshlang'ich radiusining to'g'ri tanlashiga bog'liq. Biroq, xarita o'lgovi n minglab neyron bo'lsa, xaritani o'qitish uchun zarur bo'lgan vaqt odatda amaliy muammolarnı hal qilish uchun juda katta bo'ladi, shuning uchun tugunlar sonini tanlashda maqbul kelishuvga erishish kerak.

O'qitish neyronlarnı ifodalovchi vektorli o'zgarishlar ketma-ketligidan iborat. O'qitish jarayonining har bir bosqichida, unga eng ko'p o'xshashlik, ya'ni neyron vektori koeffitsientlarining vektorlardan biri tasodifiy tanlanma to'plamidan tanlanadi. Bunday holatda neyron tanlovi g'olib neyronni tanlanadi va u vektorga o'xshaydi. Ushbu muammoning o'xshashligi odatda Evklid fazoda hisoblangan vektorlar orasidagi masofani bildiradi. Shunday qilib, agar neyron g'olibini $\|x - w_i\| = \min_j \{\|x - w_j\|\}$ formula bilan aniqlab olamiz.

G'olib neyron topilganidan so'ng, neyron tugunlarining vaznlari o'zgaradi. Bunda, g'olib-neyronni ko'rsatadigan vektor va to'rdagi qo'shnilarini tasvirlaydigan vektorlar kirish vektori yo'nalishida harakatlanadi. Bu ikki o'lgovli vektor uchun 2-rasmda keltirilgan.



2- rasm. G' olib chiqqan neyron va uning qo'shnilarining vazni o'zgartirish.

Kirish vektorining koordinatalari x bilan belgilanadi, modifikatsiyadan keyin xarita tugunlarining koordinatalari kulrangda ko'rsatiladi. Modifikatsiyadan keyin to'rt ko'rinishi kesilgan chiziqlar bilan ko'rsatilgan.

Bunday holatda quyidagi vazn koeffitsientlarini o'zgartirish uchun ishlatiladi: $w_i(t+1) = w_i(t) + h_i(t) * [x(t) - w(t)]$, bu yerda t (alohida vaqtda) bosqich raqamini bildiradi. $x(t)$ vektor iteratsiya t bosqichda o'qitish namunasidan tasodifiy tanlangan. $h(t)$ funksiyasi qo'shni neyron funksiyasi deb ataladi. Bu funksiya, g'olib neyron va to'rtidagi qo'shni neyronlar orasidagi masofani bildiradi. Masofa quyidagi ifoda yordamida

topiladi. $h(d, t) = \begin{cases} const, d \leq \sigma(t) \\ 0, d > \sigma(t) \end{cases}$ bu yerda $const$ o'zgarmas.

Faraz qiling $\sigma(t)$ funksiya o'qitish tezligini ifodalasin. Bu funksiya shuningdek, vaqt bilan kamayib boruvchi funksiyadir. Bu funksiyaning ikkita variantini ko'pincha ishlatiladi: chiziqli va vaqtga nisbatan teskari proporsional shaklidagi $\sigma(t) = \frac{A}{t+B}$, bu erda A va B o'zgarmaslar.

Ushbu funksiyani qo'llash o'qitishning natijaga erishishiga katta hissa qo'shadi.

O'rgatish ikki asosiy bosqichdan iborat: Dastlabki bosqichda o'qitish tezligi va o'qitish radiusining yetarlicha kattaligi tanlanadi, bu esa neyron vektorlarini namunadagi misollarni taqsimlash bo'yicha tartibga solish imkonini beradi va keyinchalik o'qitish tezligi parametrlarining qiymatlari dastlabki qiymatlarini ancha past bo'lganida vaznlarning aniq sozlanishi amalga oshiriladi. Chiziqli boshlash ishlatilganda dastlabki sozlash bosqichidagi qo'pol xatolarga yo'l qo'yilmaydi.

Algoritmi qo'llash. SOM algoritmi ikkita asosiy yo'nalishni – ya'ni vek-torning kvantlash va proektsiyalarini birlashtirganligi uchun ushbu algoritmning asosiy qonuniyatini topish mumkin. Bu usul tanlanma ma'lumotlarida qonuniylikni topish va tahlil qilish uchun ishlatilishi mumkin. Bunday holatda, neyronlar xaritaga joylashtirilganidan so'ng, olingan xarita ko'rsatilishi mumkin. Olingan xaritani ko'rsatishning turli usullarini ko'rib chiqing.

To'rtning barcha tugunlardagi neyron qiymatlari va vaznlari topilganidan keyin ikki o'lehevli xaritani hosil qilamiz. Bu xaritani vizual ko'rishni hosil qilish uchun neyron qiymatlari va vaznlari ushbu tugunning chizilgan rangini aniqlash uchun ishlatiladi. Odatda, kulrang ishlatiladi va masofa qanchalik uzoq bo'lsa, tugun yanada quyuglashadi. Bu foydalanish bilan ular va qo'shnilar o'rtasidagi eng katta masofa bo'lgan tugunlar qora rangda va yaqin atrofdagi tugunlarga - oq rangga mos keladi.

TA'LIM JARAYONIDA ISPRING DASTURI VA UNING IMKONIYATLARIDAN FOYDALANISH

Sidiqov M., Ortiqova N., FarDU

Zamonaviy texnologiyalarning rivojlanish holati birinchi navbatda jamiyatning intellektual salohiyatiga, ya'ni ta'lim sohasining rivojlanishiga bog'liq. Ta'lim mazmuni va sifat masalalari ustuvor yo'nalish sifatida qaralmoqda. Dunyoning rivojlangan va rivojlanayotgan



NATIONAL UNIVERSITY OF UZBEKISTAN
KHOREZM MAMUN ACADEMY
URGENCH STATE UNIVERSITY

To the 100th anniversary of the National University of
Uzbekistan named after Mirzo Ulug'bek

"АМАЛИЙ МАТЕМАТИКА ВА ИНФОРМАЦИОН
ТЕХНОЛОГИЯЛАРНИНГ ДОЛЗАРЬ МУАММОЛАРИ –
АЛ-ХОРАЗМИЙ 2018" ХАЛҚАРО АНЖУМАН
ТЕЗИСЛАРИ ТЎПЛАМИ

ABSTRACTS
OF THE VI INTERNATIONAL SCIENTIFIC CONFERENCE
"MODERN PROBLEMS OF THE APPLIED MATHEMATICS AND
INFORMATION TECHNOLOGY - AL-KHOREZMIY 2018"

ТЕЗИСЫ
МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "АКТУАЛЬНЫЕ
ПРОБЛЕМЫ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ – АЛЬ-ХОРЕЗМИ 2018"

NUU, TASHKENT, SEPTEMBER 13 – 15, 2018.

<http://www.al-khorezmiy.nuu.uz>



MINISTRY OF HIGHER AND SECONDARY SPECIALIZED
EDUCATION OF THE REPUBLIC OF UZBEKISTAN
NATIONAL UNIVERSITY OF UZBEKISTAN
KHOREZM MAMUN ACADEMY
URGENCH STATE UNIVERSITY

"AMALIY MATEMATIKA VA INFORMATSION
TEXNOLOGIYALARNING DOLZARB MUAMMOLARI –
AL-KHORAZMIY 2018" XALQARO ANJUMAN
TEZISLARI TO‘PLAMI
2018 yil, 13-15 Sentabr

A B S T R A C T S
OF THE INTERNATIONAL SCIENTIFIC CONFERENCE
"MODERN PROBLEMS OF APPLIED MATHEMATICS AND
INFORMATION TECHNOLOGIES – AL-KHOREZMIY 2018"
13-15 September, 2018

Т Е З И С Ы
МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ
"АКТУАЛЬНЫЕ ПРОБЛЕМЫ ПРИКЛАДНОЙ
МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ – АЛЬ-ХОРЕЗМИ 2018"
13-15 сентября, 2018

Dedicated to 27th anniversary of Independence of the Republic of Uzbekistan and
to 100th anniversary of the National University of Uzbekistan

SECTION №6. INFORMATION TECHNOLOGY IN EDUCATION.

Butaev U.X. <i>Indicators of political system stability assessment</i>	205
Rakhmanov K.S. <i>The role of information systems on development, retraining, and developing government rules</i>	205
Temirbekov N.M., Zhizhimov O.L., Baigereyev D.R., Omirzhanova B.B., Temirbekov A.N., Orazgaliyeva A.M., Iskakova A.M. <i>Creation of an integrated distributed library information system to support scientific research of Academgorodok</i>	206
Tulaganov Z.Sh. <i>Geodetic adjustments for Autocad</i>	207
Begmatova N.K., Musurmonova M. <i>Opportunities of usage of multimedia tools in educational process by preschool education institutions</i>	207
Ashurova D. <i>A new paradigm of education</i>	208
Tillaev A.I. <i>Creation and use of the pedagogical software with the use of information technologies</i>	209
Khodiev Sh.I., Kobilov S. <i>Regarding design languages and specifications</i>	209
Artikov M.T., Artikova M.A. <i>Role and importance of technologies of virtual reality in the field of education</i>	210
Makharov T.A., Makharov Q.T. <i>Innovative technologies in education</i>	210
Abdullaeva N.I. <i>Interactive software as a tool to improve the teaching effectiveness of discrete mathematic</i>	211
Dilshodbekov Sh. <i>Training of engineering disciplines with using computer graphics</i>	212

ADDITIONAL ABSTRACTS.

Djalilov A., Shamsiev A. <i>On the sequence of displacements for mapping of circle with one shell</i>	213
Djalilov A., Abdusakimov S., Adilov B. <i>On the time of the Poincare's return</i>	214
Xashimov A.R. <i>A general boundary-value problem for a nonstationary third-order equation of composite type</i>	215
Ishmuhammedov A.X., Odilov J.Q. <i>To study charts of factory which organized by itself without teacher</i>	216
Karimjanov A., Kadirova M.A. <i>Limiting equation and stabilization of movements of non-stationary system of equations</i>	216
Irgashev B.Yu. <i>Boundary problem for the fourth order of equation with discharge coefficient</i> ..	217
Baltaeva I.I., Urazalieva Sh.G. <i>On the integration of the loaded KdV equation with a self-consistent source that corresponds to moving eigenvalues</i>	218
Dalabaev U. <i>Algorithm of construction of the approximate solution of differential equations by the method of moved nodes</i>	219
Jarov V.K. <i>Information-pedagogical environment of higher and transitional schools of modern Uzbekistan</i>	219
Kalimbetov B.T., Temirbekov M.A. <i>Degree border layer for a singularly perturbed differential system</i>	220
Saxaev Sh., Saxaev A.Sh. <i>Estimations of one problem of electrodynamics arising in magnetic hydrodynamics in the space $W_p^{2,1}(Q_T)$, $p > 1$</i>	221
Saxaev Sh. <i>Estimations of solutions of one problem of free heat convection</i>	221
Umirxonov M.T. <i>Two-phase hyperbolic problem of heat conduction</i>	222
Ergashev T.G. <i>Solution of the Holmgren problem for one elliptic equations with several singular coefficients by the Green's function method</i>	223
Matyakubov A.S., Raupov D.R. <i>On the qualitative properties of blow-up solutions for a quasilinear system of equations of parabolic type</i>	223
Urunbaev E. <i>Algebraic method for solving logical equations</i>	224

TO STUDY CHARTS OF FACTORY WHICH ORGANIZED BY ITSELF WITHOUT TEACHER
 KOXONANING O'Z-O'ZINI TASHKIL ETUVCHI XARITALARINI O'QITUVCHISIZ O'RGATISH ALGORITMLARI

A.X. Ishmuhammedov (UzMU dots), J.Q. Odilov (UzMU magistrant).
 e-mail: Jahongir_Odilov_93@mail.ru

Koxonning o'z-o'zini tashkil etuvchi xaritalarini o'qituvchisiz o'rgatish algoritmi quyidagi bosqichlaridan iborat:
 1. Har bir tugunning og'irliklarini boshlang'ich aniqlash.
 2. Berilganlar to'plamidan tasodifiy vektor dastlabki o'rgatish uchun tanlanadi va qatlam hosil qilinadi.
 3. Koxonen neyron to'ring har bir tugunidagi kirish vektoriga o'xshashligi hisoblanadi. G'olib chiqqan tugun odatda bir qiymatini o'zlashtiradi va g'olib neyron deb ataladi.

$$K = \sum_{i=0}^{i=n} (I_i - W_i)^2$$

bu yerda K -natijaviy Koxonen neyron to'ri, I -joriy kirish vektori, W -tugundagi og'irlik vektori va n -og'irliklar soni.

4. G'olib chiqqan neyronning radiusi atrofi hisoblanadi. Bu katta ahamiyatga ega. Odatda neyron to'r radiusi sifatida g'olib neyronning radiusi belgilanadi va har bir qadamda radius kamayib boradi.

$$\sigma(t) = \sigma_0 e^{(-t/\lambda)}, \lambda = \frac{N}{\sigma_0}$$

bu yerda t -joriy itoratsiya, λ -o'zgarimas vaqt, σ_0 -xaritaning radiusi, N -takrorlanishlar soni.

5. Barcha tugunlarda g'olib neyronlardagi radius atrofi 4-bosqichdagidek hisoblanadi, kirish vektori qiymatlari o'zgaradi.

$$W(t+1) = W(t) + \theta(t)L(t)(I(t) - W(t))$$

$$L(t) = L_0(t)e^{(-t/\lambda)}$$

bu yerda $W(t)$ -vektor vazni, $L(t)$ -o'rgatish tezligi, $L_0(t)$ -boshlang'ich o'rgatish tezligi, $\theta(t)$ -g'olib neyrondan Gauss funksiyasi bo'yicha masofasi. Neyronlar g'olib neyronning tuguniga yaqin bo'lsa ularning vazni o'zgaradi.

6. N -qatlamgacha 2-bosqichdan boshlab takrorlanadi.

Koxonen neyron to'ring barcha tugunlardagi neyron qiymatlari va vaznlari topilganidan keyin ikki o'lchovii xarita hosil qilinadi. Bu xaritaning vizual ko'rinishini hosil qilish uchun kirish neyronlari qiymati va vaznlari ushbu tugunning rangini aniqlash uchun ishlatiladi. Odatda kulrang ishlatiladi va masofa qanchalik uzoq bo'lsa, tugun yanada quyuyqlashdi. Qo'shnilar orasidagi eng katta masofa bo'lgan tugunlar qora rangda va yaqin atrofdagi tugunlar oq rangga mos keladi.

Foydalanilgan adabiyotlar

1. Кохонос Т. Самоорганизующиеся карты // М.: БИНОМ. Лаборатория знаний, 2008. 654-655 б.

LIMITING EQUATION AND STABILIZATION OF MOVEMENTS OF NON-STATIONARY SYSTEM OF EQUATIONS
 ПРЕДЕЛЬНЫЕ УРАВНЕНИЯ И СТАБИЛИЗАЦИЯ ДВИЖЕНИЙ НЕСТАЦИОНАРНЫХ УПРАВЛЯЕМЫХ СИСТЕМ

А. Каримжанов, М.А. Кадырова
 Андижанский государственный университет имени З.М. Бабура, Андижан
 e-mail: abdulaziz-karimjanov@rambler.ru

Задачи о стабилизации и об оптимальной стабилизации сформулированы Н.Н. Красовским и решены им в случае, когда правые части уравнений не зависят явно от времени [1].

Оптимальная стабилизация по части переменных рассматривается в работе В.В. Румянцева [2].

Общая идея применения предельных уравнений для анализа качественных свойств движений, развитых в работах [3-5], применяется для исследования задач оптимальной стабилизации механических систем.

В данной работе исследуется задача об оптимальной стабилизации невозмущенного движения механической системы в классе управляющих воздействий, явно зависящих от времени. Устанавливаются достаточные условия оптимальной стабилизации управляемых систем с нейтральной (без управления) частью.

Полученные в данной работе теоремы модифицируют результаты Н.Н. Красовского [1] и В.В. Румянцева [2] тем, что ослабляют условия, налагаемые на производную оптимальной функции Ляпунова.

ЛИТЕРАТУРА

1. Красовский Н.Н. В кн.: И.Г. Малкина Теория устойчивости движения. М.: Наука, Дополнение 4, с. 475-514.
 2. Румянцев В.В. Об оптимальной стабилизации управляемых систем // Прикл. математика и механика. 1970, 34, вып. 3, с. 440-456.
 3. Каримжанов А. Предельные системы в задаче об устойчивости неавтономных систем // Прикл. механика. - 1985. - 21, с 5. - С. 110-117.
 4. Martynyuk A.A., Karimzhanov A. Limiting equations and stability of non-stationary motions // J.Math.Anal. and Appl. 1988. 133(1), p. 101-108.

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АХБОРОТ ТЕХНОЛОГИЯЛАРИ ВА
КОММУНИКАЦИЯЛАРИНИ РИВОЖЛАНТИРИШ ВАЗИРЛИГИ
МУҲАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**

**ЎЗБЕКИСТОН РАДИОТЕХНИКА, ЭЛЕКТРОНИКА ВА АЛОҚА
ИЛМИЙ-ТЕХНИКА ЖАМИЯТИ**

**ИҚТИСОДИЁТНИНГ ТАРМОҚЛАРИНИ ИННОВАЦИОН
РИВОЖЛАНИШИДА АХБОРОТ-КОММУНИКАЦИЯ
ТЕХНОЛОГИЯЛАРИНИНГ
АҲАМИЯТИ**

**Республика илмий-техник анжуманининг
МАЪРУЗАЛАР ТЎПЛАМИ
2-қисм**



**2 - часть
СБОРНИК ДОКЛАДОВ
Республиканской научно-технической конференции**

**РОЛЬ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ
ТЕХНОЛОГИЙ В ИННОВАЦИОННОМ РАЗВИТИИ
ОТРАСЛЕЙ ЭКОНОМИКИ**

14-15 март 2019 йил

ТОШКЕНТ 2019

МУНДАРИЖА

3-ШУЪБА. ДАСТУРНИЙ МАҚСУЛОТЛАРНИ ЯРАТИШ ВА УНИНГ ИСТИҚБОЛЛАРИ	
<i>Sapayev S.H.O., Rajabov O.T.</i> CHEKLI ELEMENTLAR USULI YORDAMIDA UCH O'LCHOVLI SOHANING DISKRET MODEL VA GRAFIK DASTURIY TA'MINOT	4
<i>Bashimov N.R., Rajabov O.T.</i> BIR, IKKI HAMDA UCH O'LCHOVLI SOHANING DISKRET MODELINI OBYEKTGA YO'NALTIRILGAN PROGRAMMALASH(OYP) TEXNOLOGIYASI ASOSIDA QURISH	6
<i>Karimov J.J.</i> MATNLI MA'LUMOTLARGA NOANIQ IZLASH ALGORITMLARI YORDAMIDA DASTLABKI QAYTA ISHLOV BERISH	8
<i>Matvarepov S.R.</i> TASHQI IQTISODIY FAOLIYATDA ELEKTRON TIZIMLARDAN FOYDALANISH AFZALLIKLARI	10
<i>Odilov J.Q., Ishmuhammedov A.X.</i> O'Z O'ZINI TASHKIL QILUVCHI KOHONEN NEYRON TARMOG'IDA KLASTERIZATSIYA	12
<i>Bashimov N.R.</i> ANSYS DASTURI YORDAMIDA QUVUR BOSIMNING OSESIMMETRIK TAHLILI	14
<i>Abdurazzoqov F.B., Yuldashev Yu.Sh., Nurimov P.B.</i> NUTQ SIGNALLARGA RAQAMLI BERISH NAZARIYASI VA TEXNOLOGIYASI	16
<i>Mamatov H.S., Samijonov A.H., Zoirov J.U.</i> ПРОБЛЕМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ УЗБЕКСКОЙ РЕЧИ	18
<i>Samijonov A.H., Yuldashev Yu.Sh., Abduraimov J.F.</i> ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА РЕЧЕВЫХ СИГНАЛОВ	20
<i>Mamatov H.S., Karaxatov H.A., Almatov H.M.</i> ВЫБОР ИНФОРМАТИВНЫХ ПРИЗНАКОВ ДЛЯ ОЦЕНКИ ЗАБОЛЕВАНИЯ	22
<i>Bekmurotov T.F., Soliyeva B.T.</i> МЕТОД ПОСТРОЕНИЯ БАЗЫ НЕЧЕТКИХ ПРАВИЛ С ИСПОЛЬЗОВАНИЕМ КЛАСТЕРИЗАЦИИ	23
<i>Muxamediyeva D.T.</i> ИСПОЛЬЗОВАНИЕ Z-ЧИСЕЛ В СИСТЕМЕ НЕЧЕТКОГО ВЫВОДА	26
<i>Muxamediyeva D.T., Oxunbadaev U.</i> ИСПОЛЬЗОВАНИЕ ИММУННОГО АЛГОРИТМА ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА	28
<i>Zsaimberdiyev H.A., Xasanov U.V., Xudolberdiyev A.</i> АЛГОРИТМЫ ПОСТРОЕНИЯ НЕЧЕТКИХ ЛОГИЧЕСКИХ МОДЕЛЕЙ СЛАБОФОРМАЛИЗУЕМЫХ ПРОЦЕССОВ	30
<i>Sayfiyev J., Zsaimberdiyev H.A.</i> УТОЧНЕНИЕ ОЦЕНОК НЕЧЕТКИХ ПАРАМЕТРОВ В НЕЛИНЕЙНЫХ ПО ПАРАМЕТРАМ МОДЕЛЯХ	32
<i>Muxamediyeva D.K.</i> МЕТОД НЕЛИНЕЙНОГО РАСЩЕПЛЕНИЯ ДЛЯ ПОСТРОЕНИЯ АВТОМОДЕЛЬНЫХ РЕШЕНИЙ В РЕЖИМЕ ОБОСТРЕНИЯ	35
<i>Muxamediyeva D.K.</i> АВТОМОДЕЛЬНЫЕ РЕШЕНИЯ НЕЛИНЕЙНОГО УРАВНЕНИЯ РЕАКЦИИ ДИФФУЗИИ С ДВОЙНОЙ НЕЛИНЕЙНОСТЬЮ И С ПЕРЕМЕННОЙ ПЛОТНОСТЬЮ	37
<i>Niyozmatova H.A.</i> СОЗДАНИЕ СИСТЕМЫ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА МОДЕЛИРУЮЩЕЙ ЗАВИСИМОСТЬЮ СЛАБО ФОРМАЛИЗУЕМЫХ ПРОЦЕССОВ	40

yagona elektron axborot tizimi bazasida qayd qilishlari mumkin. Custom Management dasturining E-Kontrakt dasturidan farqi shundaki, E-Kontrakt dasturida mahsulotga buyurtma beriladi, lekin soliq turlari va ularning miqdori hisobga olinmagan va foydalanuvchilar E-IMZO orqali ro'yxatdan o'tishlari shart. Custom Management dasturida esa foydalanuvchidan dasturga kirish uchun E-IMZO talab qilinmaydi, faqat o'zining login va paroli orqali dasturga kirish imkoniyatiga ega.

Foydalanilgan adabiyotlar ro'yxati

1. Karimov I.A. «O'zbekiston buyuk kelajak sari» Toshkent. «O'zbekiston», 1998 y.
2. O'zbekiston Respublikasi Soliq Kodeksi. Toshkent, 2000 y.
3. Jo'raev A.S., Kuzieva N.R. "Markaziy Osiyo davlatlari soliq tizimi" – o'quv qo'llanma. Toshkent – 2005

O'Z O'ZINI TASHKIL QILUVCHI KOHONEN NEYRON TARMOG'IDA KLASTERIZATSIYA

J.Q. Odilov (magistrant, M. Ulugbek nomidagi O'zMU)

A.X. Ishmuhammedov (dotsent, M. Ulugbek nomidagi O'zMU)

Kohonen neyron tarmoqlari deganda o'z-o'zini tashkil etuvchi neyron tarmoqlarni nazarda tutiladi. O'z-o'zini tashkil qiluvchi neyron tarmoqlari ma'lum umumiy xususiyatlarga ega kirish vektorlarining klasterlarini (guruhlarini) aniqlash imkonini beradi.

Klasterizatsiya- bu tekshirilayotgan obyektlarni "o'xshash" obyektlar klasterlar deb nomlangan guruhlariga ajratilishi. "Klaster" atamasi (Inglizcha Klaster - paket, guruh) sinf, guruh atamalariga sinonim atamadir. Klasterizatsiya usuli klassifikatsiya usulidan tubdan farq qiladi. Klassifikatsiya usuli yechimi predmetlarning har biriga predmet sinflardan biriga tayinlashdir. Klasterizatsiya vazifasida obyekt oldindan aniqlanmagan sinflardan biriga tayinlanish. Klasterlardagi obyektlarning klaster-larga ajratish bir vaqtning o'zida klasterlar shakllanishi bilan amalga oshiriladi.

Klasterizatsiya, ma'lumotlarni yig'ish vazifalarini bir qator hal qilishni osonlashtiradigan o'xshash ma'lumotlarni to'plash imkonini beradi.

- Ma'lumotlarni tahlil qilish, tahlil qilishni osonlashtirish. Olingan guruhlarni mazmunli tahlil qilish qonuniylikni aniqlash imkonini beradi. Masalan, siz uyali aloqa tarmog'i mijozlarini aniqlashingiz mumkin, ular uchun yangi tarif rejasini taklif qilishingiz mumkin. Boshqa misollar - savdo tarmog'ining xaridor guruhlarini aniqlash, bozorni segmentlash. Klaster tarkibining tahlili turli xil guruhlarning obyektlariga tahlilning turli usullarini qo'llash imkonini beradi.

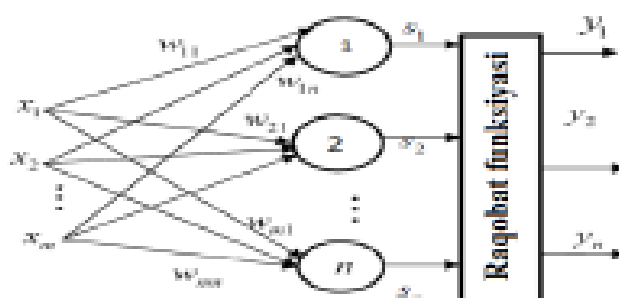
- Prognozlash. Klasterlardan biriga yangi obyektni tayinlash orqali, uning xatti-harakati klaster obyektlarining xatti-harakati bilan o'xshash bo'lgani uchun, obyekt xatti-harakatlarini oldindan belgilashingiz mumkin.

- Anomaliyalarni aniqlash. Guruhlarni to'g'ri aniqlanishi anomaliyalarni aniqlashga yordam beradi. Odatda, bu klaster, yerda juda oz narsalar tushadi.

Har biri m ta alomatdan iborat $x_i = (x_{i1}, x_{i2}, \dots, x_{im}), i = 1, 2, \dots, n$ vektorlar berilgan. Natijada kirish vektorlarini alomatlari bo'yicha bir-biriga o'xshash vektorlardan iborat klasterlarga ajratish kerak (1-rasm). Kohonen neyron tarmog'ida klasterlarga ajratish algoritmi quyidagicha.

1-bosqich. Barcha $x_j (j = 1, 2, \dots, n, j = 1, 2, \dots, m)$ kirish vektorlari $[0:1]$ oraliqda o'tkazib (bu natijani tezroq topishga imkon beradi chunki $[0:1]$ oraliqda amallar katta sonlarga nisbatan kamroq kompyuter vaqtini oladi) normallashtiriladi.

2-bosqich. O'qitish uchun $w_j (j = 1, 2, \dots, s, j = 1, 2, \dots, m)$ (bu yerda t -klasterlar soni) vazn vektori koeffisientlari $[0:1]$ oraliqdan tasodifiy tanlab olinadi



1-rasm. Kohonen neyron tarmog'ining strukturasi.

3-bosqich. O'qitish uchun tasodif kirish vektori tanlanadi va $k=1$ bo'ladi.

4-bosqich. Tanlangan kirish vektorining barcha neyronlarining har bir klastergacha bo'lgan Yevklid masofasi hisoblanadi.

$$d_j = \rho(x, w_j) = \|x - w_j\| = \sqrt{\sum_{i=1}^m (x_i - w_{ji})^2} \quad (1)$$

bu yerda, x_i - tanlangan vektorining i - neyron kirishi, w_{ji} - i - neyronning j - klasterga kirishining vazn koeffisienti, m - neyronlar soni, d_j - klastergacha bo'lgan masofasi

5-bosqich. d_j lar orasidan eng kichik masofagi neyron topiladi va u g ' olib neyron hisoblanadi.

$$\min_{k \in \text{son}} d_j = \min_{k \in \text{son}} \|x - w_j\| = \min_{k \in \text{son}} \sqrt{\sum_{i=1}^m (x_i - w_{ji})^2} \quad (2)$$

6-bosqich. G ' olib neyron topilgandan keyin j -klasterdagi vazn koeffisientlari o'zgartiriladi.

$$w_j^{k+1} = w_j^k + a(k)(x_i - w_j^k) \quad (3),$$

$$a(k) = a_0 e^{-k/\alpha}$$

bu yerda, w_j^k - (k) - qatlamdagi vazn koeffisienti, x_i - kirish vektorini i -neyroni, $a(k)$ -o'qitish tezligi koeffisienti, w_j^{k+1} - $(k+1)$ - qatlamdagi vazn koeffisienti .

7-bosqich. Agar $k < N$ bo'lsa 4-bosqichga o'tiladi, aks holda 3-bosqichga o'tiladi.

8-bosqich. Agar kirish vektorlarini hammasi ko'rib chiqilgan bo'lsa algoritm tugaydi.

Hosil bo'lgan Kohonen neyron tarmog'idan foydalanib o'z o'zini tashkil qiluvchi Kohonen xartiasini tuziladi. Kohonen xartiasi natijani visual ko'rish uchun juda qulay usuldir.

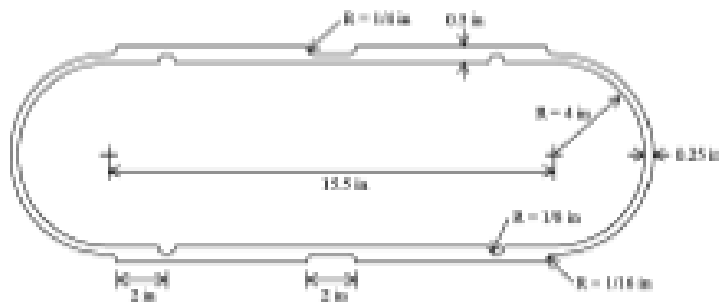
Foydalanilgan adabiyotlar

1. Кохонен Т. Самоорганизующиеся карты//М.: БИНОМ. Лаборатория знаний, 2008.
2. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика //М. : Мир, 1992.

ANSYS DASTURI YORDAMIDA QUVUR BOSIMNING OSESIMMETRIK TAHLILI

N.R.Beshimov (magistrant, M.Ulugbek nomidagi O'zMU)

Quyida ko'rsatilgan quvur chugundan ya'ni quyma temirdan tayyorlangan ($E=14,5 \text{ M Pa}, \nu=0,21$) va ichki bosim $p=1700 \text{ Pa}$ ga teng. ANSYS dasturida uzunlik o'lchov birligi uchun dyuym ya'ni inglizchasiga "in" o'lchov birligi bilan ishlash qulay shuning uchun uzunlik o'lchovi uchun in qo'llaniladi. Silindrsimon idishning ichki diametri 8 in gacha va uning o'lchamlarini 1-rasmda ko'rishimiz mumkin.



1-rasm Jism modeli va o'lchamlari

Ushbu masalada ANSYS dasturi yordamida jism ichki bosim ta'sirida devorlarning kengayishini baholaymiz va bosim ta'sirining visual korinishini hosil qilamiz. Jism tuzilishi bitta markaziy o'qqa nisbatan simmetrik bo'lganligi uchun uni va chekli elementlari 8 ta tugunli to'rtburchak elementlardan iborat osesimmetrik ko'rinishda yechamiz. Silindr jism markaziy o'qqa nisbatan simmetrik bo'lgani uchun uning to'rtidan bir qismi uchun masala yechimini hosil qilamiz.

ANSYS dasturida tahlil: ANSYS dasturini ishga tushiramiz va unda quyidagi boshlang'ich amallarni bajaramiz.

*Utility Menu->File->Change Jobname...->*kiritamiz "osesimmetrik_masala"->OK

*Utility Menu->File->Change Title...->*kiritamiz "Silindr ichki bosimining osesimmetrik tahlili"->OK

*ANSYS Main Menu->Preference->Preferences for GUI Filtering->*tanlaymiz "Structural" va "h-method"->OK

Jismining geometrik modelini yaratish uchun *Preprocessor* bo'limiga kiramiz:

Element turi va xususiyatlarini aniqlaymiz.

*ANSYS Main Menu->Preprocessor->Element Type->Add/Edit/Delete->*tanlaymiz *Solid Quad 8 node 82 (PLANE82)->OK->Click Options...->*K3 uchun "Axisymmetric" ni tanlaymiz. ->OK->Close.

