

**КОХОНЕН ЖЕЛІЛЕРІНІҢ ЕРЕКШЕЛІКТЕРІ**

**ОСОБЕННОСТИ СЕТИ КОХОНЕНА**

**FEATURES OF KOHONEN NETWORKS**

Қ.С. ИМАНБАЕВ, Г.Н. МАХСУТ  
K.S. IMANBAYEV, G.N. MAKHSUT

(Алматы технологиялық университеті)  
(Алматинский технологический университет)  
(Almaty Technological University)  
E-mail: gmakhsut@gmail.com

*Мақалада рекурренттік оқыту алгоритмі, композициялық WTA және WTM оқыту қағидаттарын қорыту болып табылатын, кластерлеу үшін қолжетімді және ықтималдық тәсілдерді біріктіретін кластерлік Кохонен желілерін оқыту ұсынылған. Кохонен желілерін оқыту үшін қолданылатын математикалық модельді пайдалануымен, МатЛаб бағдарламасының көмегімен Кохонен картасы құрылып оқытылған.*

*В статье предложен рекуррентный алгоритм обучения составной кластеризирующей сети Кохонена, являющейся обобщением WTA и WTM принципов обучения, а также объединяющий в себе возможностный и вероятностный подходы к кластеризации. С использованием математической модели обучения сети Кохонена, и с помощью программы МатЛаб построена и была обучена Карта Кохонена.*

*In this article proposes a recursive learning algorithm composite clustered Kohonen networks, which is a generalization of the WTA and WTM principles of learning, as well as combining possibilistic and probabilistic approaches to clustering. Using a mathematical model to training of Kohonen networks, and with the help of Matlab software was built and has been trained Kohonen map.*

**Негізгі сөздер:** нейрондық желілер, Кохонен желісі, математикалық модель, жеңімпаз-нейрон, кластерлеу.

**Ключевые слова:** нейронные сети, сеть Кохонена, математическая модель, нейрон-победитель, кластеризация.

**Keywords:** neural networks, Kohonen network, mathematical model, neuron-winner, clustering.

**Кіріспе**

Нейрон желілері – адам миының нейрон желілерін құру идеясына негізделген - математикалық модель. Нақты емес логика, адамның талқылауына үйреншікті «жоғары жылдамдық», «өте жоғары жылдамдық» сияқты нақты емес түсініктерді пайдаланады. Эволюциялық модельдеу кездей соқ өзгерістер мен биологиялық эволюцияны еске салатын, пайдалы қасиеттерді бекіту

механизмдерін пайдаланады. Тиімділеу әдісі орталықтанбаған өздігінен ұйымдастырушы жүйенің ұжымдық жүрісін сипаттайды. Осындай қысқаша сипаттамалар, есептеу интеллектісінің көптеген идеялары табиғаттан алынғанын көрсетеді. Сондықтан «есептеу интеллекті» ғылыми бағыты белгілі бір дәрежеде "Natural Computing" – «Табиғи есептеу» ғылыми бағытымен қиылысады, ол шешім қабылдаудың табиғи механизмдерінің

принциптерінен тұратын математикалық модельдерді біріктіреді [1].

Кохонен желілері (Kohonen T.) нейрон желілерне жатады [2]. Өздігінен ұымдас-тыру қасиеті бар желі белгілі бір ортақ қасиеттері бар кіріс векторларының кластер-лерін (топтарын) анықтауға мүмкіндік береді. Реттелмеген нейрондары бар (әдетте Кохонен қабаттары деп аталады) желілер және рет-телген нейрондары бар (Кохонен карталары деп аталады) желілер қолданылады. Кохонен карталары, картадағы жақын орналасқан де-ректтер кластеріне, оларға жақын орналасқан нейрондар сәйкес келетіндей, құрылымды көрсетеді [3, 4].

### ***Зерттеу нысандары мен әдістері***

Зерттеу нысандары ретінде кохонен желілері және кохонен картасы қолданылды. Бұл мақалада келесі зерттеу әдістері қол-данылған: кохонен желілерін оқыту үшін бә-секелік механизмі, кохонен картасын құру үшін математикалық модельдер.

### ***Нәтижелері және оларды талдау***

Желі кірісі  $\mathbf{X}$  векторын қолданса, салмақ векторы кіріс векторынан аз ғана дәрежеде ерекшеленетін нейрон жеңеді. Жеңімпаз-нейрон үшін келесі қатынас орындалады (1):

$$d(\mathbf{x}, \mathbf{w}_j) = \min_{1 \leq i \leq n} d(\mathbf{x}, \mathbf{w}_i), \quad (1)$$

мұндағы:  $n$  - нейрондар саны,  $j$  -

жеңімпаз-нейрон нөмірі,  $d(\mathbf{x}, \mathbf{w})$  -  $\mathbf{x}$  және  $\mathbf{w}$  векторлары арасындағы қашықтық (тандалған өлшемде).

Көп жағдайда қашықтық өлшемі ретінде евклидов өлшемі алынады (2):

$$d(\mathbf{x}, \mathbf{w}_i) = \|\mathbf{x} - \mathbf{w}_i\| = \sqrt{\sum_{j=1}^m (x_j - w_{ji})^2} \quad (2)$$

Бұдан басқа қашықтық өлшемдері де қолданылады.

Белсендірудің бәсекелік функциясы сумматор шамаларын талдайды және нейрон-дар шығысын құрайды. Онда «жеңімпаз-нейронның» шығысында ғана максимал шама болады, ал қалған барлық нейрондар үшін шығыс 0-ге тең болады. Осылайша, шығыс векторының бір ғана элементі болады, ол 1-ге тең және жеңімпаз-нейронға сәйкес келеді. Белсенді нейрон нөмірі, кіріс векторы ең жақын орналасқан топты (кластерді) анықтайды.

Кохонен желісінде кіріс шамаларын нормалау керек. Ол үшін келесі формулалардың бірін пайдалану керек (3):

$$x_{ni} = \frac{x_i}{\sqrt{\sum_{i=1}^m x_i^2}}, \quad x_{ni} = \frac{x_i}{|x_i|}, \quad (3)$$

мұнда  $x_{ni}$  - кіріс векторының норма-ланған құраушысы.

Кіріс деректерін нормалау, желіні оқы-ту жылдамдығына тиімді әсер етеді.

Оқыту үрдісінің алдында желі ини-циализацияланады, яғни салмақ векторының бастапқы шамалары беріледі. Қарапайым жағ-дайда салмақтарының кездейсоқ шамасы бері-леді. Кохонен желісін оқыту үрдісі бірнеше қа-дамның циклдік қайталануынан тұрады:

1. Кірісіне бастапқы берілгенді беру. Әдетте бұл кіріс векторларының бірін таңдау.
2. Әрбір нейронның шығысын табу.
3. «Жеңген» нейронды немесе жеңімпаз нейронды анықтау (салмағы кіріс векторының құраушысынан аз ғана ерекшеленетін нейрон).
4. Кохонен ережесі бойынша «жеңген» нейронның салмағын реттеу (4)

$$\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^{(k)} + \eta_i^{(k)} \left[ \mathbf{x} - \mathbf{w}_i^{(k)} \right], \quad (4)$$

мұнда:  $\mathbf{x}$  - кіріс векторы,  $k$  - оқыту циклінің нөмірі,  $\eta_i^{(k)}$  -  $k$ -ші оқыту цикліндегі  $i$ -ші нейронның оқыту жылдамдығының коэффициенті.

5. Егер оқыту аяқталса, 1 қадамға ауысу. Neural Network Toolbox MATLAB бағ-дарламасында оқыту циклінің саны беріледі. Және қателік функционалының аз шамасына жетуін тексеруге болады (5):

$$E = \frac{1}{Q} \sum_{i=1}^Q \|\mathbf{x}_i - \mathbf{w}_x\|^2 \quad (5)$$

мұнда:  $\mathbf{w}_x$  - кіріс векторы  $\mathbf{x}_i$  берілгендегі жеңімпаз-нейронның салмақ векторы,  $Q$  - оқыту таңдамасының өлшемі.

Салмақ векторы кіріс векторына жақын болған нейрон, кіріс векторына жақынырақ болу үшін жаңарып отырады. Нәтижесінде бұл нейрон, кірісіне жақын векторды берген-де, жеңеді және кіріс векторына алыс век-торды бергенде, жеңіледі. Оқыту вектор-ларын бірнеше рет берген соң, вектор клас-терге тиісті болған кезде 1 шамасын беретін нейрон пайда болады, ал кластерге тиісті болмаса, 0 береді. Осылайша желі кіріс век-торларын жіктеуді үйренеді.

Кохонен желісін оқытуда «өлі» нейрондар мәселесі туындайды. Кез келген бәсекелік қабаттың шектеулерінің бірі – кейбір нейрондардың үрдіске мүлдем қатыспауы. Бастапқы салмақтық векторлары бар, кіріс векторларынан алыс орналасқан нейрондар оқытуға қанша қатысса да, бәсекеде ешқашан жеңбейді. Нәтижесінде мұндай нейрондарды оқытуда қолданбайды. Ондай нейрондар «өлі» нейрондар деп аталады, себебі олар ешқандай пайдалы функция атқармайды. Соған байланысты кіріс деректері аздаған нейрондармен ғана түсіндіріледі. Сондықтан барлық нейрондардың жеңуіне мүмкіндік беру керек. Ол үшін оқыту алгоритмін «өлі» нейрондар оқытуға қатысатындай модификациялайды.

Мысалы, жеңімпаз-нейрон белсенділігін жоғалтатындай оқыту алгоритмін модификациялайды [5]. Нейрондардың белсенділігін ескерудің бір әдісі – оқыту үрдісіндегі әрбір нейронның  $p_i$  потенциалын есептеу. Бастапқыда нейрондарға  $p_i(0) = \frac{1}{n}$  потенциалы меншіктеледі, мұндағы  $n$  — нейрондар саны (кластерлер). Оқытудың  $k$  -ші циклінде потенциал келесі ережемен анықталады (6):

$$p_i(k) = \begin{cases} p_i(k-1) + \frac{1}{n}, & i \neq j, \\ p_i(k-1) - p_{\min}, & i = j, \end{cases} \quad (6)$$

мұнда:  $j$  - жеңімпаз-нейрон нөмірі.

Егер  $p_i(k)$  потенциал шамасы  $p_{\min}$  деңгейінен төмен болса, нейрон қарастырылмайды – «демалады».  $p_{\min} = 0$  кезінде нейрондар бәсекеге қатысады.  $p_{\min} = 1$  кезінде нейрондар тізім бойынша жеңіп отырады, себебі оқытудың әрбір цикліне олардың бірі ғана бәсекеге дайын болады. Тәжірибеде  $p_{\min} \approx 0.75$  кезінде жақсы нәтижеге жетуге болады.

Neural Network Toolbox бағдарламасында «өлі» нейрондармен күрес үшін нейрондардың ығысуын өзгерту қолданылады [8]. Өлі нейрондардың сезімсіздігін ескеретін сәйкес ереже learncon функциясы түрінде орындалған. Реттеу үрдісінің басында бәсекеге түсетін қабаттың барлық нейрондарына бірдей белсенділік параметрі беріледі  $c_0 = \frac{1}{N}$ , мұнда  $N$  — кластер санына тең бәсекеге түсетін нейрондар саны. Реттеу

кезінде learncon функциясы, белсенді нейрондар үшін осы параметр жоғары, ал белсенді емес нейрондар үшін төмен шамада болатын реттейді. Белсенділік параметрінің векторына сәйкес келетін формула (7):

$$\mathbf{c}^{(k+1)} = (1 - r_l) \mathbf{c}^{(k)} + r_l \mathbf{s}^{(k)} \quad (7)$$

мұнда:  $r_l$  - реттеу жылдамдығының параметрі;  $k$  - оқыту циклінің нөмірі;  $\mathbf{s}^{(k)}$  -  $k$ -ші оқыту цикліндегі бейімделген сумматорлар шығысының векторы.

Ығысу векторының құраушысы келесі формуламен есептеледі (8):

$$b_i^{(k+1)} = e^{(1 - \ln c_i^{(k+1)})} - b_i^{(k)} \quad (8)$$

«Жеңімпаз-нейрон» үшін ығысу азаяды, ал қалған нейрондар үшін артады. Реттеу жылдамдығының параметрі  $r_l = 0,001$ . Белсенді емес нейрондар үшін ығысудың артуы кіріс шамалар диапазонын кеңейтеді, және белсенді емес нейрон кластер құра бастайды. Соңында ол жаңа кіріс векторларын тартады. Егер салмақ векторы желі кірісіне түсетін векторлардан ерекше болатын нейрондар жеңімпаз болмаса, онда оның ығысуы оқыту барысында үлкен шамаға жетеді және ол бәсекеге қабілетті болады. Сонымен қатар оның салмақ векторы кіріс векторларының шамасына жете бастайды. Нейрон жеңе бастағанда, оның ығысуы азая бастайды. Осылай «өлі» нейрондар мәселесі шешіледі. Ығысумен байланысты екінші артықшылық – олар кіріс векторының шамамен бірдей санын тартады. Егер кластерлердің бірі басқасы мен салыстырғанда көп кіріс векторын тартса, оның толтырылған облысы нейрондардың қосымша санын тартып, өлшемі кіші кластер құрады.

Осыған байланысты Кохонен желісін MATLAB бағдарламасының көмегімен құрамыз және оқытамыз. Ол үшін келесі түрдегі competlayer функциясы қолданылады: competlayer (numClasses, kohonenLR, conscience LR), мұнда numClasses – жіктелетін кластерлер саны (5-тен), kohonenLR – Кохонен салмағын оқыту жылдамдығы (үнсіздік кезінде 0,01), conscience LR – реттеу жылдамдығының параметрі (үнсіздік кезінде 0,001).

Төрт екі элементті вектордан массив береміз, оны екі классқа бөлу керек:

```
>> p=[0.1 0.8 0.1 0.9; 0.2 0.9 0.1 0.8]
p = 0.1000 0.8000 0.1000 0.9000
    0.2000 0.9000 0.1000 0.8000
```

Мысал бойынша екі вектор (0,0) нүктесіне жақын және екі вектор – (1,1) нүктесіне жақын.

Екі элементті кіріс векторларын талдау үшін екі нейрондары бар Кохонен қабатын құрамыз:

```
>> net = competlayer (2);
```

Енді, өздігінен ұйымдастырушы нейрондық желі құрылған соң оны 500 оқыту циклінде оқытамыз. Ол үшін train функциясы қолданылады:

```
>> net.trainParam.epochs = 500;
```

Оқытудың қандай функциясы net атаулы құрылған желіде қолданылатынын білу үшін келесі команданы орындау керек:

```
>> net.trainFcn
```

```
ans = trainru
```

Кохонен ережесі бойынша реттеу learnk функциясымен беріледі (үнсіздік кезінде беріледі). Тексеру үшін келесі команданы орындаймыз:

```
>> net.inputWeights{1,1}.learnFcn
```

```
ans = learnk
```

Ығысу белсенділігінің параметрін реттеу үшін learncon функциясы қолданылады. Бұл функцияны пайдалану үшін келесі сәйкес қасиеттерді орнату керек:

```
>>net.inputWeights{1}.learnFcn='learncon'
```

Желіні оқытумыз:

```
>> net1=train(net, p);
```

Оқытудан кейін желіні модельдейміз:

```
>> a = net1(p)
```

```
a = 1 0 1 0
```

```
0 1 0 1
```

Алынған нәтиже бойынша, бірінші кіріс векторы нөмірі 1 болатын кластерге жатады, екінші вектор – 2 кластерге жатады, ал үшінші вектор – 1 кластерге, төртінші вектор – 2 кластерге жатады. Нәтижені vec2ind функциясының көмегімен көрсеткен ыңғайлы, ол әрбір вектордағы бірлік элемент векторын шығарады:

```
>> ac=vec2ind(a)
```

```
ac = 1 2 1 2
```

Желі кіріс векторларын екі кластерге жіктеуді үйренді: координата басына жақын орналасқан векторлар бірінші кластерлерге жатады; ал (1, 1) нүктесіне жақын орналасқан векторлар – екінші кластерге жатады.

Желінің салмағы мен ығысуы келесіге тең:

```
>> wts1=net1.IW{1,1}
```

```
wts1 = 0.1000 0.1500  
0.8500 0.8500
```

```
>> biases1 = net1.b{1}
```

```
biases1 = 5.4367
```

5.4365

Салмақ пен ығысу шамалары көрсетілген шамадан өзгеше болуы мүмкін, себебі кіріс векторлар кездейсоқ ретте оқыту үшін таңдалады. Бірінші нейронның салмақ векторы (ол салмақ матрицасының бірінші жолы) (0, 0) векторына жақын, ал екінші нейронның салмақ векторы (1, 1) жақын.

Келесі векторды беру арқылы желіні тексереміз:

```
>> p1=[0.2; 0.3]
```

```
p1 = 0.2000
```

```
0.3000
```

Желіні модельдеу арқылы келесі нәтижені аламыз:

```
>> ac1=vec2ind(net(p1))
```

ac1 = желі векторды бірінші кластерге дұрыс жіктеді.

### **Қорытынды**

Нейрон желілерін қолданудың негізгі артықшылығы әр түрлі бейресми міндеттерді шешу мүмкіндігі болып табылады. Тіпті бұл ретте әр түрлі жағдайларды модельдеуге, желі кірісіне әр түрлі деректерді енгізе отырып, желі арқылы берілген нәтижелерді бағалау.

Нейрон желілерін қолдану барысында белгіленген елеулі кемшіліктері: желі нәтижесін алу процесін түсіну күрделілігі. Осы сұрақтың шешімі үшін алғашқы қадамы нейрон желілеріне негізделген жаңа технология дамыту болып табылады.

Тәжірибелік деректер кестесін пайдалана отырып, осы мәселені шешу үшін нақты алгоритмді алуға болады.

### **ӘДЕБИЕТТЕР ТІЗІМІ**

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - М.: Мир, 1992.- 240 с.
2. Хайкин С. Нейронные сети: полный курс. - М.: Вильямс, 2006. - 1104 с.
3. Осовский С. Нейронные сети для обработки информации. - М.: Финансы и статистика, 2002. - 244 с.
4. Кохонен Т. Самоорганизующиеся карты. - М.: БИНОМ. Лаборатория знаний, 2008. - 655 с.
5. Элементарное введение в технологию нейронных сетей с примерами программ / Р. Тадеусевич, Б. Боровик, Т. Гончаж, Б. Леппер.- М: Горячая линия-Телеком, 2011. - С. 408.

