

Öz-düzenlemeli Ağ Yapısı ile Farklı Yaklaşımların Sınanması Testing Different Approaches by Self Organizing Map

Faranak Nouri¹, N.Serap Şengör¹

¹ Elektronik ve Haberleşme Mühendisliği Bölümü
İstanbul Teknik Üniversitesi
fnouri@itu.edu.tr, sengorn@itu.edu.tr

Özet

Özdüzenlemeli ağ yapısı ile verilerin öbeklenmesi oldukça yaygın kullanılan bir yöntemdir. Bu çalışmada ilk olarak özdüzenlemeli ağ ile vektör kuantalama yapıları birlikte ele alınmıştır ve özdüzenlemeli ağ yapısında, vektör kuantalama yapısında olduğu gibi sınıf bilgisi kullanılmıştır. Ayrıca, son senelerde pekiştirmeli öğrenme ile birlikte özdüzenlemeli yapıların ele alındığı hibrit yaklaşımlardan esinlenilmiştir ve özdüzenlemeli ağ ve sınıf bilgisinin kullanıldığı özdüzenlemeli ağın eğitimi basit bir pekiştirmeli öğrenme yaklaşımı ile uyarlanarak yeni bir yaklaşım önerilmiştir. Özdüzenlemeli ağ yapısı ve önerilen yaklaşımlar MATLAB ortamında m-file olarak gerçekleştirilmiş ve basit bir nokta sınıflama problemine uygulanmıştır. Elde edilen benzetim sonuçları eğitim ve test kümelerindeki başarımları ele alınarak irdelenmiştir ve önerilen yöntemlerin geliştirilmesi için önerilerde bulunulmuştur.

Abstract

Clustering data with self organizing maps is a widely used method. In this work, self organizing map and learning vector quantization methods are considered for solving clustering problems. Firstly, self organizing map is modified to use class information like learning vector quantization method. Also, inspired by the hybrid methods proposed recently which combine reinforcement learning with self organizing map, a novel approach is proposed where a simple reinforcement learning algorithm is used both in self organizing map and self organizing map with class information. Self organizing map and all the proposed approaches are realized as m-files in MATLAB and applied to clustering two dimensional points set. The discussion on the simulation results are based on the results obtained for training and test sets and ideas are given to improve these results.

1. Giriş

Özellikle veri öbekleme ve verinin boyutunu indirgemedi yaygın olarak kullanılan özdüzenlemeli ağ (Self Organizing Map (SOM)) ve vektör kuantalamayı öğrenme (Learning Vector Quantization (LVQ)), T. Kohonen tarafından önerilmiştir [1-3]. SOM yapısının kazanan hepsini alır yaklaşımını kullanan diğer öbekleme algoritmalarından en

önemli farkı, kazanan hücrenin sadece kendisi değil bir veya iki boyutlu geometriye göre önceden tanımlanmış komşu hücrelerin de veriyi temsil edecek şekilde güncellenmesidir. LVQ yapısında ise komşuluk tanımlanmamıştır ancak sınıf bilgisinde öbekleme sürecinde kullanıldığından bu yapı genellikle eğitici öğrenme içeren bir yapı olarak ele alınmaktadır [1,2]. Yaygın kullanımına rağmen SOM yapısının iki sakıncası vardır: yakınsamadaki yavaşlığı ve yakın benzerlikteki verileri ayırt edememe özelliği. Bu çalışmada özellikle bu iki sakıncayı hedef alacak şekilde öncelikle SOM yapısında da sınıf bilgisi kullanılacak sonra basit bir pekiştirmeli öğrenme kuralı ile özdüzenlemeli ağ yapısının öğrenme kuralında değişiklik yapılacaktır. Önerilen yeni öğrenme kuralı ile hem SOM hem de sınıf bilgisinde kullanıldığı SOM yapıları MATLAB ortamında m-file olarak hazırlanan yazılım ile gerçekleştirilecek ve basit bir öbekleme problemine uygulanacaktır. Bu yayında asıl amaç yeni bir öğrenme yöntemi önerip nasıl uygulanacağını açıklamak olduğundan yöntemin sınındığı problem kolaylıkla üretilebilen ve eğitim amaçlı çokça kullanılan iki boyutlu noktaların öbeklenme problemidir [2].

Yakın zamanda önerilen kimi çalışmalarda pekiştirmeli öğrenme ile özdüzenlemeli yapılar bir arada kullanılmaktadır [4-6]. Bunlardan [4]'de önerilen eğitim kuralında, SOM yapısı pekiştirmeli öğrenmede yeri olan aracı için ilk eğitim verilerini az sayıdaki eğitim kümesinden oluşturmada kullanılmıştır. Böylece eğitim sürecinin hızlanması sağlanmıştır. [5]'de önerilen metod da ise kamera görüntülerinden oluşmuş eğitim kümesi verilerinin sınıflanması SOM ve pekiştirmeli öğrenme içiçe kullanılarak sağlanmıştır. [6] ise sinirbilim ile ilişkili olarak primatların görme sistemini SOM yapısını bazı ampirik kurallar ile pekiştirmeli öğrenme çerçevesinde uyarlayarak modellemeyi amaçlamaktadır.

Bu bildiri kapsamında yapılan çalışma, SOM yapısını gerçekleyip, komşuluk bilgisinden faydalanarak öbekleme problemini çözmek, ayrıca yukarıda değinilen yayınlardan esinlenerek pekiştirmeli öğrenme yapısının en azından ödül mekanizmasını kullanarak, elde edilen SOM yapısına ilişkin yeni bir öğrenme yaklaşımı önermeyi kapsamaktadır. İkinci bölümde SOM ve bu çalışma kapsamında bölüm 2.1.2, 2.2.1 ve 2.2.2'de ilk defa önerilen yaklaşımlara ilişkin algoritmalar verilecek, üçüncü bölümde MATLAB m-file olarak oluşturulan bu yapılar iki boyutlu noktaları öbekleme

problemine uygulanacak ve elde edilen benzetim sonuçları tartışılacaktır. Son bölümde ise bu tartışmalar ışığında ileriye dönük çalışmalar için önerilerde bulunulacaktır.

2. Özdüzenlemeli Ağ Yapısı ve Farklı Yaklaşımlar

2.1. Özdüzenlemeli Ağ

SOM eğiticişiz öğrenme ile giriş verilerini geometrik bir yapıda yerleştirilmiş sonlu sayıdaki birimler ile ifade edebilen bir yapay sinir ağı yapısıdır. Geometrik yapıda düzenlenmiş birimlerin komşuluk ilişkisi bu birimleri tanımlayan ağırlık vektörlerinin güncellenmesi sırasında gözönüne alındığından yakın komşuluktaki birimler benzer verileri temsil etmektedir. Böylece verilerin oluşturduğu giriş uzayının topolojik özellikleri bir veya iki boyutlu geometrik yapıya aktarılmış olur. Bu topolojik özelliğin korunmasına yol açan öğrenme kuralı nedeniyle SOM öbekleme probleminde kullanılan diğer eğiticişiz öğrenme yapılarından farklılık gösterir [1,2].

2.1.1. Özdüzenlemeli ağı algoritması

Eğitim kümesindeki verilerin ağa tekrar ve tekrar sunulması ile nöronların ağırlıkları uyarlanarak verilerin oluşturduğu öbekleri temsil edecek şekilde değişir. Komşu nöronların ağırlıkları benzer şekilde değiştiğinden verilerin bulunduğu giriş uzayının özellikleri topolojik olarak korunarak ayırık nöron uzayına dönüştürülmüş olur.

- **İlk değerlerin atanması:** Ağırlık vektörleri $w_j \in R^m$, $j = 1, \dots, l$ rastgele atanır. Böylece verilerin öbek merkezlerine karşı düşecek l tane nöronun ilk değerleri belirlenmiş olur.

- **Komşuluğun belirlenmesi:** l tane nöron istenilen bir veya iki boyutlu geometride yerleştirilerek komşuluk düzeni belirlenir ve her nöron arasındaki fiziksel uzaklık d_{ij}^{mesafe} hesaplanır.

$$d_{ij}^{mesafe} = \|d_i - d_j\|, j = 1, \dots, l, i = 1, \dots, l \quad (1)$$

- **Verilerin sunulması:** Eğitim kümesinden bir veri $x_k \in R^m$, $k = 1, \dots, p$ seçilir

- **Kazanan nöronun belirlenmesi:** Eğitim kümesinden seçilen k . veriye l tane nöron içinden en benzerinin belirlenmesi için $d_{ij}^{benzerlik}$ hesaplanır,

$$d_{ki}^{benzerlik} = \left(\sum_{j=1}^m (x_{kj} - w_{ij})^2 \right)^{1/2}, i = 1, \dots, l \quad (2)$$

ve k . veri için kazanan nöronun indisi

$$indis^{kazanan}(k) = \min_i \{d_{ki}^{benzerlik}\}, i = 1, \dots, l \quad (3)$$

- **Kazanan nöron ve komşularının ağırlığının güncellenmesi:** Kazanan nöronun ağırlığı en fazla olmak üzere belirlenen komşuluk fonksiyonu

$\Theta(d_{ij}^{mesafe}, n)$ 'e göre kazanan nöron ve komşularının ağırlığı denklem (4)'deki gibi güncellenir.

$$w(n+1) = w(n) + \Theta(d_{ij}^{mesafe}, n) \mu(n)(x(n) - w(n)) \quad (4)$$

Burada öğrenme hızına karşılık gelen $\mu(n)$ denklem (5)'deki gibi ifade edilir, ve ardışıl güncellemeler boyunca üstel olarak azalır.

$$\mu(n) = \mu_o \exp\left(-\frac{n}{\lambda}\right) \quad (5)$$

Ardışıl güncellemeye bağlı bir başka terimde komşuluk fonksiyonudur ve ardışıl güncellemeler boyunca daralmaktadır.

- **Sonlanma:** Eğitim kümesindeki aynı veriler için aynı nöronlar kazanmaya başlayınca öbekleme oluşmuştur, ancak öbek merkezlerine karşılık gelen nöronların o öbekteki verileri temsil etme özelliklerinin etkinleşmesi için bu aşamadan sonra eğitim sürecine o aşamaya kadar atılan adım sayısının yaklaşık dört katı kadar bir süre devam edilir.

2.1.2. Özdüzenlemeli ağda sınıf bilgisinin kullanılması

Yine T. Kohonen tarafından önerilen LVQ yapısı, SOM'dan farklı olarak eğiticişiz öğrenmedir ve verilerin öbeklenmesi aşamasında sınıf bilgisini de kullanmaktadır. Ancak bu yapıda komşuluk ilişkisi gözönüne alınmayıp sadece kazanan nöronun ağırlığı güncellenmektedir. Bu güncellenme sırasında kazanan nörona atanan sınıf etiketi ile verinin sınıf etiketi aynı ise nöronun ağırlığı artırılır, değilse ağırlığı azaltılır. Örneğin üç sınıf ile etiketlenmiş bir veri kümesinde 3., 5., 7. nöronlar 1. sınıf etiketini, 1., 2., 4. nöronlar 2. sınıf etiketini, 6. ve 8. nöronlar 3. sınıf etiketini taşıyabilir.

Bu çalışmada bu sınıf bilgisi SOM yapısında da ele alınacaktır ve yapılan bu değişiklik ile elde edilen yapı eğiticişiz özdüzenlemeli yapı (E-SOM) olarak isimlendirilecektir. Bu önerilen yeni yapıya ilişkin algoritma adımları 2.1.1 bölümünde SOM için verilen ile aşağıda nasıl değiştikleri belirtilen "ilk değerlerin belirlenmesi" ve "kazanan nöron ve komşularının ağırlığının güncellenmesi" adımları dışında aynı kalacaktır.

- **İlk değerlerin atanması:** Ağırlık vektörleri $w_j \in R^m$, $j = 1, \dots, l$ rastgele atanır. Böylece verilerin öbek merkezlerine karşı düşecek l tane nöronun ilk değerleri belirlenmiş olur. Bu l tane nöron veri kümesindeki sınıf sayısı r ise r farklı etiket ile etiketlenir.
- **Kazanan nöron ve komşularının ağırlığının güncellenmesi:** Kazanan nöronun ağırlığı en fazla olmak üzere belirlenen komşuluk fonksiyonu $\Theta(d_{ij}^{mesafe}, n)$ 'e göre kazanan nöron ve komşularının ağırlığı denklem (6)'deki gibi güncellenir.

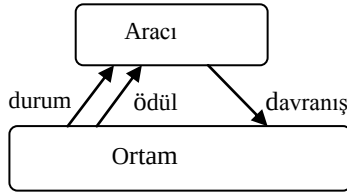
$$w(n+1) = w(n) + \Theta(d_{ij}^{mesafe}, n) \delta^{simf}(n) \mu(n)(x(n) - w(n)) \quad (6)$$

Burada sınıf bilgisini öğrenme kuralını etkilemesini sağlayan terim $\delta^{sınıf}(n)$ denklem (7) ile verilmiştir

$$\delta^{sınıf}(n) = \begin{cases} 1 & \text{n. verinin etiketi ile kazanan} \\ & \text{nöronun etiketi ay n ise} \\ -1 & \text{n. verinin etiketi ile kazanan} \\ & \text{nöronun etiketi ay n ise} \end{cases} \quad (7)$$

2.2. Pekiştirmeli Öğrenme

SOM yapısı gibi Pekiştirmeli Öğrenme (PE) yaklaşımı da eğitimcisi öğrenme yaklaşımıdır ve yapının öğrenmesi aracının davranışlarının ortamda değerlendirilip ödülendirilmesine dayanmaktadır. Şekil 1'de verilen yapıda da gözüktüğü gibi PE yapısında aracının davranışlarını değerlendiren bir ortam ve davranışları ile ortamın durumunu değiştiren bir aracı bulunmaktadır. Aracının amacı en uygun durumlara yol açacak davranışları ortam ile etkileşerek belirlemektir.



Şekil 2: Genel bir pekiştirmeli öğrenme yapısı

Bu çalışmada PE yaklaşımında eğitim aşamasında kullanılan ödül kavramından esinlenerek ve SOM yapısında her veri için belirlenen kazanan nöronun aracının ortamı değiştiren davranışı olarak düşünülerek bu davranış 2.2.1 bölümünde belirtildiği gibi değerlendirilip bu değerlendirmeye göre kazanan nöronun ağırlığının değiştirilip değiştirilmeyeceğine karar verilecektir.

2.2.1. Pekiştirmeli öğrenmenin uyarlandığı özdenlemeli ağırlık algoritması

Bölüm 2.1.1'de verilen SOM algoritmasında sadece güncelleme aşamasında değişiklik yapılacaktır. Bu aşamaya gelince k. veri için kazanan nöronun benzerliğine dikkat edilecek ve bu benzerlik k. veri için eğitimin bir önceki iterasyonunda kazanan nöronun benzerliğinden daha az ise yani denklem 2 ile hesaplanan değer daha büyük ise kazanan nöronun ağırlığı güncellenmeyecek yok değilse güncellenecektir. Dolayısıyla SOM yapısında hangi nöronun seçildiği değerlendirilip seçilen nöron daha önceki iterasyon adımıyla elde edilen sonuçtan daha iyi değilse ağırlık güncellenmeyecek, daha iyi ise ödül olarak ağırlığı güncellenecektir. Dolayısıyla sadece kazanan nöronun ve komşularının ağırlığının güncellenmesi aşamasında değişiklik olacaktır.

- **Kazanan nöron ve komşularının ağırlığının güncellenmesi:** Kazanan nöronun ağırlığı en fazla olmak üzere belirlenen komşuluk fonksiyonu $\Theta(d_{ij}^{mesafe}, n)$ 'e göre kazanan nöron ve komşularının ağırlığı denklem (8)'deki gibi güncellenir.

$$w(n+1) = w(n) +$$

$$\Theta(d_{ij}^{mesafe}, n) \delta^{ödül}(n) \mu(n)(x(n) - w(n)) \quad (8)$$

Burada $\delta^{ödül}(n)$ denklem (9)'da tanımlandığı gibi alınır.

$$\delta^{ödül}(n) = \begin{cases} 1 & d_{ki}^{benzerlik}(n) \geq d_{ki}^{benzerlik}(n-1) \\ 0 & d_{ki}^{benzerlik}(n) < d_{ki}^{benzerlik}(n-1) \end{cases} \quad (9)$$

2.2.2. Pekiştirmeli öğrenmenin uyarlandığı ve sınıf bilgisinin kullanıldığı özdenlemeli ağırlık algoritması

Bu yapıda ise bu çalışmada ilk defa önerilen bölüm 2.1.2 ve bölüm 2.2.1'deki iki yaklaşım bir arada değerlendirilmiş ve kazanan nöronun ve komşularının ağırlığının güncellenmesi aşağıda açıklandığı gibi ele alınmıştır.

- **Kazanan nöron ve komşularının ağırlığının güncellenmesi:** Kazanan nöronun ağırlığı en fazla olmak üzere belirlenen komşuluk fonksiyonu $\Theta(d_{ij}^{mesafe}, n)$ 'e göre kazanan nöron ve komşularının ağırlığı denklem (10)'deki gibi güncellenir.

$$w(n+1) = w(n) +$$

$$\Theta(d_{ij}^{mesafe}, n) \delta^*(n) \mu(n)(x(n) - w(n)) \quad (10)$$

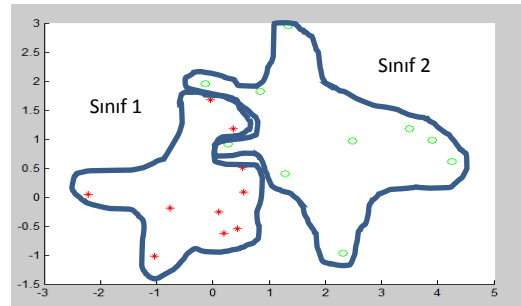
$$\delta^*(n) = \begin{cases} 1 & d_{ki}^{benzerlik}(n) \geq d_{ki}^{benzerlik}(n-1), \delta^{sınıf}(n) = 1 \\ -1 & d_{ki}^{benzerlik}(n) \geq d_{ki}^{benzerlik}(n-1), \delta^{sınıf}(n) = -1 \\ 0 & d_{ki}^{benzerlik}(n) < d_{ki}^{benzerlik}(n-1) \end{cases}$$

3. Benzetim Sonuçları

Bölüm 2.1.1, 2.1.2, 2.2.1 ve 2.2.2'de sırası ile algoritmaları verilen SOM, E-SOM, PE-SOM, ve PE-E-SOM yapılarını sınamak için MATLAB ortamında m-file olarak hazırlanan ve parametre değerleri Çizelge 1'de verilen yazılımlar ile bir örneği Şekil 1'de verilen noktaları öbekleme problemine uygulanmıştır.

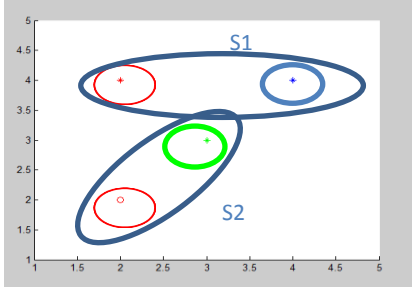
Çizelge 1: Kullanılan Parametre değerleri

Parametreler	l_0	σ_0	λ	m	l	k
Parametre değerleri	1	3	2.5	2	4,6,8,10	Eğitim:20 Test:6



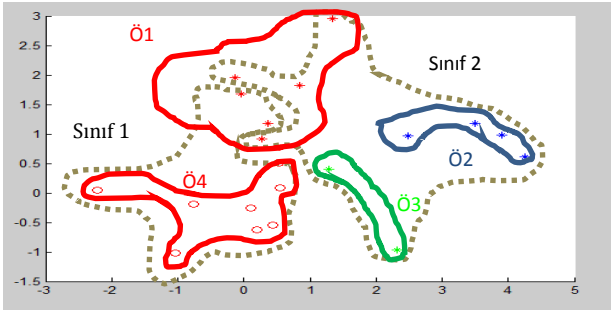
Şekil 1: İki sınıfa ait noktaların bulunduğu bir eğitim kümesi

Bu uygulama için nöronların nasıl bir düzende sıralandığına ilişkin bir örnek 4 nöron için Şekil 2’de verilmiştir. Eğitim sonunda her bir nöronun hangi sınıfa ilişkin verileri içerdiği de Şekil 3’de verilmiştir. Şekil 3’den takip edilebileceği gibi 4 nöron ele alındığından iki sınıfa ait veriler bu dört nöronda öbeklenmiştir. 1. Öbeği temsil eden nöronun 2 boyutlu geometride bulunduğu koordinat (4,2), 2., 3. ve 4. Öbeği temsil eden nöronların bulunduğu koordinatlarda sırası ile (4,4), (3,3), (2,2)’dir. Görüldüğü gibi aynı sınıfa karşı gelen öbekler yakın komşulardır.

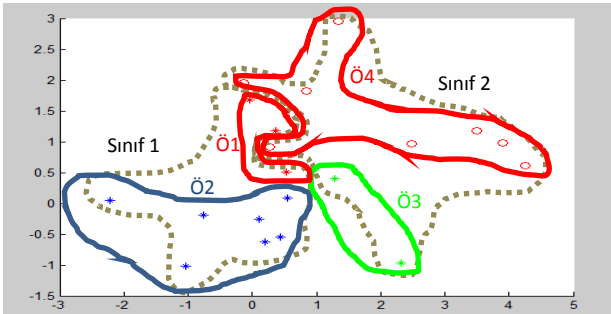


Şekil 2: Dört nöron için komşuluk tanımının yapıldığı 2-boyutlu geometri.

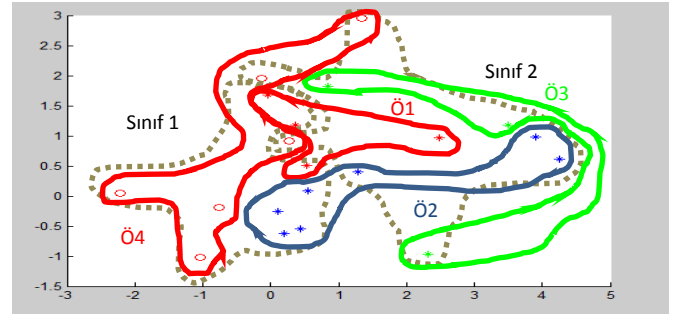
Şekil 1’de verilen veriler için ele alınan dört yöntemde uygulandığında eğitim sonunda her bir öbeğe ilişkin veriler ve sınıf bilgisi Şekil 3-6’da verilmiştir. Bu şekillerden de takip edilebileceği gibi E-SOM yapısı ve PE-E-SOM yapıları ile oluşan öbekler sınıfları hatasız belirlerken, SOM ve PE-SOM yapıları ile sırası ile 2 ve 7 hatalı veri vardır. Ancak yine SOM ile PE-SOM’a ilişkin sonuçlara dikkatli bakılırsa her ne kadar PE-SOM’un genel performansı kötü ise de SOM’un ayıramadığı birbirine çok yakın ama ayrı sınıflara ait verileri farklı öbeklere yerleştirmiştir.



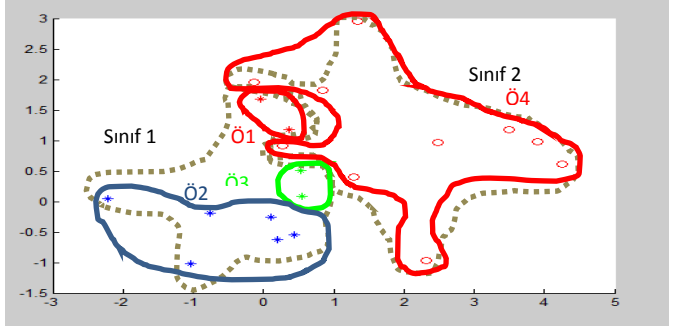
Şekil 3: Şekil 1’de verilen eğitim kümesi için SOM yapısına ilişkin sonuçlar



Şekil 4: Şekil 1’de verilen eğitim kümesi için E-SOM yapısına ilişkin sonuçlar



Şekil 5: Şekil 1’de verilen eğitim kümesi için PE-SOM yapısına ilişkin sonuçlar



Şekil 6: Şekil 1’de verilen eğitim kümesi için PE-E-SOM yapısına ilişkin sonuçlar

Her dört yöntemi de karşılaştırmak amacıyla istatistiksel olarak anlamlı olması için 30’ar kere farklı şekilde üretilen eğitim kümeleri için tüm yöntemlere ilişkin hazırlanan m-file’lar çalıştırılmış ve performanslarını karşılaştırmak için ortalamaları verilmiştir. Çizelge 3-6’da 30’ar kere gelişigüzel üretilen eğitim ve test kümeleri aynı tutularak sonuçlar verilmiş, Çizelge 7-10’da ise her yöntem için birbirlerinden farklı olarak tamamen rastgele üretilen eğitim ve test kümeleri için sonuçlar verilmiştir.

Çizelge 2: 30 kere gelişigüzel üretilmiş eğitim kümeleri için öbeklerin ortalama oluşma adım sayısı

	SOM	E-SOM	PE-SOM	PE-E-SOM
Ortalama adım sayısı	5.43±0.9	4.6±1.0	3.63±0.8	3.76±0.8

Çizelge 2’den de izlenebileceği gibi PE ile uyarlanmış yöntemlerde öbeklerin oluşması hızlanmıştır ancak hata sayısı diğer iki yönteme göre fazladır. Öbekler oluştuğundan sonra öbek merkezlerinin öbekleri daha iyi temsil etmesi için devam edilen eğitim aşaması 100 adım kadar devam edilse bile öbeklere giren verilerin kazanan nörona uzaklıklarının ortalamasında fark binde bir mertebesinde kalmaktadır.

Çizelge 3: 30 kere gelişigüzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için SOM yapısına ilişkin sonuçlar. Aynı veri kümeleri Çizelge 4-6’da da kullanılmıştır.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	3.43±1.5	3.83±1.4	1.44±0.4
6nöron	2.80±1.3	3.00±1.8	1.02±0.3
8nöron	2.40±1.4	2.63±1.3	0.30±1.4
10nöron	2.30±1.4	2.60±1.4	0.50±0.2

Önerilen üç yöntemden en iyi sonuç veren E-SOM yapısı olmuştur. Ancak ödül mekanizması daha iyi kurgulandığında PE-SOM yapısında birbirine çok benzer verileri farklı öbeklere yerleştirebilme yeteneğinden faydalanabilmenin mümkün olabileceği görülmektedir. Yöntemlerin kararlı davrandığı tamamen farklı veriler ile aynı verilerin kullanıldığı durumlarda sonuçların çok değişmemesinden izlenebilir.

Çizelge 4: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için E-SOM yapısına ilişkin sonuçlar. Veri kümeleri Çizelge 3,5,6 ile aynıdır.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	0.33±0.8	1.50±1.2	3.33±1.3
6nöron	0.40±0.6	2.00±1.3	3.34±1.3
8nöron	0.26±0.8	1.80±1.3	2.81±1.1
10nöron	0.43±0.7	1.36±1.1	1.90±1.4

Çizelge 5: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için PE-SOM yapısına ilişkin sonuçlar. Veri kümeleri Çizelge 3,4,6 ile aynıdır.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	6.80±1.6	5.46±0.9	5.20±2.7
6nöron	6.23±1.5	5.43±1.1	4.08±2.1
8nöron	5.10±1.2	4.36±1.6	2.31±1.1
10nöron	4.50±1.9	4.30±1.6	2.02±1.1

Çizelge 6: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için PE-E-SOM yapısına ilişkin sonuçlar. Aynı veri kümeleri Çizelge 3,4-5'de de kullanılmıştır.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	1.90±1.4	2.90±0.8	8.63±4.8
6nöron	2.53±0.9	3.37±0.8	5.77±2.4
8nöron	2.20±1.3	3.33±1.1	4.00±1.6
10nöron	1.03±1.3	2.36±1.8	3.76±1.8

Çizelge 7: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için SOM yapısına ilişkin sonuçlar.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	3.03±1.3	4.40±1.3	1.55±0.3
6nöron	2.07±1.4	3.00±1.4	1.07±0.4
8nöron	1.83±1.3	3.27±1.3	3.87±2.7
10nöron	2.06±1.0	2.83±1.6	0.44±0.1

Çizelge 8: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için E- SOM yapısına ilişkin sonuçlar.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	0.23±0.6	1.50±1.0	3.08±0.9
6nöron	0.50±0.6	2.10±1.3	3.50±1.0
8nöron	0.33±0.7	1.87±1.2	2.62±1.4
10nöron	0.73±1.0	2.03±1.3	2.03±1.3

Çizelge 9: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için PE- SOM yapısına ilişkin sonuçlar.

Nöron sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	7.00±1.3	5.43±1.00	5.50±4.4
6nöron	6.23±1.5	5.23±1.2	4.20±2.5
8nöron	4.70±1.4	4.20±1.5	2.05±1.1
10nöron	4.23±1.4	3.83±1.4	2.00±1.1

Çizelge 10: 30 kere geliş güzel üretilmiş 20 verilik eğitim ve 6 verilik test kümesi için PE-E- SOM yapısına ilişkin sonuçlar.

Nöron Sayısı	Eğitim kümesinde hatalı veri sayısı	Test kümesinde hatalı veri sayısı	Ortalama benzerlik
4nöron	2.00±1.6	3.06±0.8	7.31±2.9
6nöron	1.93±1.1	3.50±0.8	5.48±1.9
8nöron	1.60±1.3	3.20±1.3	4.80±2.2
10nöron	0.76±0.8	3.23±1.4	3.72±1.7

Nöron sayısının artması sonuçları belli bir değere kadar olumlu etkilerken 10 nöron için elde edilen sonuçlar her zaman diğer sonuçlardan daha iyi olmamıştır. Dolayısıyla nöron sayısının çok fazla artması sonuçları her zaman olumlu etkilememektedir.

4. Sonuçlar

Bu çalışmada öncelikle SOM yapısına ilişkin algoritma verilmiş ve sonra bu yapıyı geliştirmek amacıyla bölüm 2.1.2, 2.2.1 ve 2.2.2'de yeni yapılar önerilmiştir. Bu yapılar basit ama sınama problemi olarak çok kullanılan iki boyutlu noktaların öbeklenmesi problemini çözmek için MATLAB ortamında hazırlanan m-file 'lar kullanılmış ve sonuçlar tartışılmıştır.

5. Kaynaklar

- [1] Kohonen, T. "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, Cilt no.43, sf. 59-69, 1982.
- [2] Haykin, S., *Neural Networks and Learning Machines*, 3rd Edition, Pearson International Edition, New Jersey, 2009.
- [3] Somervuo, P., Kohonen T. "Self-organizing Maps and Learning Vector Quantization for Feature Sequences", *Jour. Neural Processing Letters*, Cilt no. 10/2, sf. 151-159, 1999.
- [4] Tateyama, T., Kawata, S., Oguchi, T. "A teaching method using a self-organizing map for reinforcement learning", *Artif. Life Robotics*, Cilt no.7, sf.193-197, 2004.
- [5] Dozono, H., Fujiwara, R., ve Takahashi, T., "Visual Reinforcement Learning Algorithm using Self Organizing Map and Its Simulation in OpenGL Environment", *WSEAS Transactions on Information Science & Applications*, Cilt no. 5/5, sf. 685-694, 2008.
- [6] Brohan, K., Gurney, K., Dudek, P. "Using Reinforcement Learning to Guide the Development of Self-organised Feature Maps for Visual Orienting", *International Conference on Artificial Neural Networks*, 2010, sf. 180-189.