

İstanbul Teknik Üniversitesi
Elektrik ve Elektronik Fakültesi
Elektronik ve Haberleşme Mühendisliği



Yapay Sinir Ağları

Ödev-3

Dersi Veren Öğretim Üyesi
Prof. Dr. Neslihan Serap ŞENGÖR

Öğrenci Grubu 3

Yunus Örük

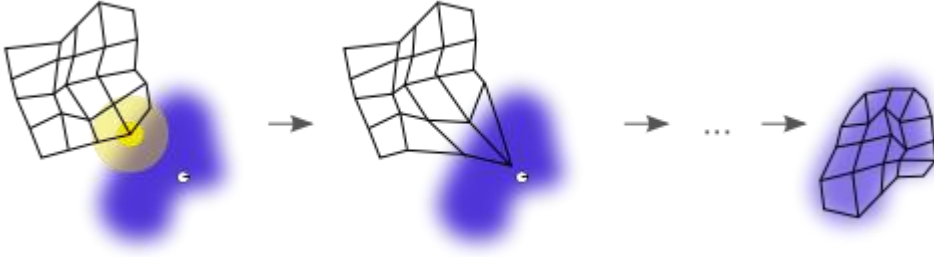
040170083

Yusuf Demirel

040180033

KOHONEN ÖZ DÜZENLEMELİ AĞI (SOM)

Kohonen öz düzenlemeli ağı daha yüksek boyutlu bir veri kümesinin düşük boyutlu bir temsiliyi üretmek için kullanılan denetimsiz bir makine öğrenimi tekniğidir. Verilerin topolojik yapısını korumaktadır. Örneğin, n gözlemde ölçülen p değişkenli bir veri seti, değişkenler için benzer değerlere sahip gözlem kümeleri olarak temsil edilebilir. Bu kümeler daha sonra iki boyutlu bir "harita" olarak görselleştirilebilir, öyle ki yakın kümelerdeki gözlemler, uzak kümelerdeki gözlemlerden daha benzer değerlere sahiptir. Bu, yüksek boyutlu verilerin görselleştirilmesini ve analiz edilmesini kolaylaştırabilir. SOM bir tür yapay sinir ağıdır, ancak diğer yapay sinir ağları tarafından kullanılan hata düzeltme öğrenimi yerine rekabetçi öğrenme kullanılarak eğitilir. SOM, 1980'lerde Finli profesör Teuvo Kohonen tarafından tanıtıldı ve bu nedenle bazen Kohonen haritası veya Kohonen ağı olarak adlandırılır. Kohonen haritası veya ağı, 1970'lerdeki nöral sistemlerin biyolojik modelleri ve 1950'lerde Alan Turing'e dayanan morfogenez modelleri üzerinde hesaplama açısından uygun bir soyutlamadır.

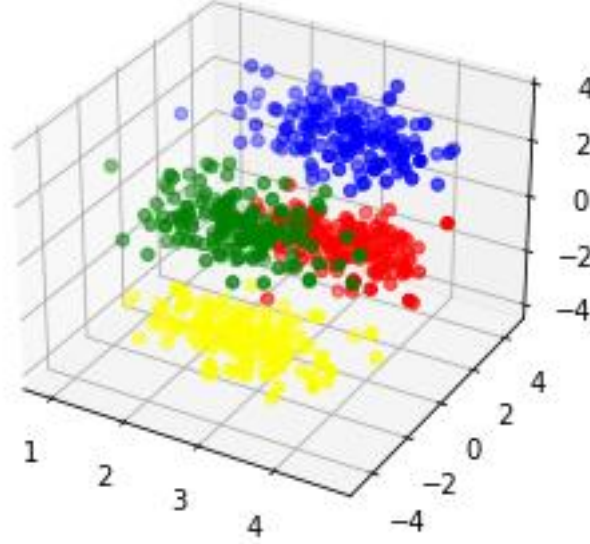


Kendi kendini organize eden haritada öğrenmenin amacı, ağın farklı bölümlerinin belirli girdi modellerine benzer şekilde yanıt vermesini sağlamaktır. Bu kısmen görsel, işitsel veya diğer duyuşal bilgilerin insan beynindeki serebral korteksin ayrı bölümlerinde nasıl işlendiğiyle temsil edilir. Nöronların ağırlıkları ya küçük rastgele değerlere başlatılır ya da en büyük iki ana bileşen özvektörü tarafından yayılan alt uzaydan eşit olarak örneklenir. İkinci alternatifle öğrenme çok daha hızlıdır çünkü başlangıç ağırlıkları zaten SOM ağırlıklarının iyi bir tahminini verir. Ağ, eşleme sırasında beklenen vektör türlerini mümkün olduğunca yakın temsil eden çok sayıda örnek vektörle beslenmelidir. Örnekler genellikle yinelemeler halinde birkaç kez uygulanır. Eğitim, rekabetçi öğrenmeyi kullanır. Ağa bir eğitim örneği beslendiğinde, tüm ağırlık vektörlerine Öklid mesafesi hesaplanır. Ağırlık vektörü girdiye en çok benzeyen nörona en iyi eşleşen birim denir. SOM ızgarasında en iyi eşleşen birim ve ona yakın nöronların ağırlıkları giriş vektörüne göre ayarlanır. Değişimin büyüklüğü zamanla ve en iyi eşleşen birime olan mesafe ile azalır.

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s)),$$

Soru 1

Öncelikle 4 farklı sınıfa ait, 3 boyutlu farklı ortalama ve standart sapmalara sahip 4*150 adet nokta oluşturuldu. Her bir sınıf için farklı bir renk seçildi ve 3 boyutlu düzlem üzerinde bu noktalar çizdirildi.



Şekil 1 Veri kümesinin 3 boyutta çizdirilmesi

Kazanan nöronları ve ağırlıkları daha iyi yorumlayabilmek adına her bir sınıfa bir değer atandı (Eğitimde kullanılmamak şartıyla). Sarılara 0, kırmızılara 1 gibi.

2 boyutlu olacak şekilde nöronların belirlenmesi kararlaştırıldı ve bu x ve y eksenindeki nöron sayılarını seçme işi parametrik olarak yazıldı. Eğer bir ekseninde 1 nöron diğerinde farklı sayıda bir nöron seçilirse tek boyutlu nöron kümesi de elde edilebilme imkânı da tanındı. Genel olarak ise 5'e 5'lik bir nöron haritası seçilmiştir. Ardından eğitime geçilmeden önce her bir nöronun diğerlerine olan uzaklıklarını gösterecek olan tablo çizdirildi.

```
number_neurons_x = 5      #X eksenindeki nöron sayısının belirlenmesi  
number_neurons_y = 5      #Y eksenindeki nöron sayısının belirlenmesi
```

Şekil 2 Nöron sayısının eksenlere göre seçilmesi

Bizim verilerimiz genel olarak -3 ve 3 arasında oldukları için nöronların ilk ağırlıkları da rastgele olarak -3, 3 arasında seçildi.

3 boyutlu 600 adet nokta içeren veri kümesi rastgele olarak karıştırıldı ve ardından %80 e %20 olacak şekilde ayrıldı. Yani 480 nokta eğitim kümesine, 120 nokta test kümesine yerleştirildi. Sigma ve learning rate değerleri seçildi.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	1	2	3	4	1	1.41421	2.23607	3.16228	4.12311	2	2.23607	2.82843	3.60555	4.47214	3	3.16228	3.60555	4.24264	5	4	4.12311	4.47214	5	5.65685
1	1	0	1	2	3	1.41421	1	1.41421	2.23607	3.16228	2.23607	2	2.23607	2.82843	3.60555	3.16228	3	3.16228	3.60555	4.24264	4.12311	4	4.12311	4.47214	5
2	2	1	0	1	2	2.23607	1.41421	1	1.41421	2.23607	2.82843	2.23607	2	2.23607	2.82843	3.60555	3.16228	3	3.16228	3.60555	4.47214	4.12311	4	4.12311	4.47214
3	3	2	1	0	1	3.16228	2.23607	1.41421	1	1.41421	3.60555	2.82843	2.23607	2	2.23607	4.24264	3.60555	3.16228	3	3.16228	5	4.47214	4.12311	4	4.12311
4	4	3	2	1	0	4.12311	3.16228	2.23607	1.41421	1	4.47214	3.60555	2.82843	2.23607	2	5	4.24264	3.60555	3.16228	3	5.65685	5	4.47214	4.12311	4
5	1	1.41421	2.23607	3.16228	4.12311	0	1	2	3	4	1	1.41421	2.23607	3.16228	4.12311	2	2.23607	2.82843	3.60555	4.47214	3	3.16228	3.60555	4.24264	5
6	1.41421	1	1.41421	2.23607	3.16228	1	0	1	2	3	1.41421	1	1.41421	2.23607	3.16228	2.23607	2	2.23607	2.82843	3.60555	3.16228	3	3.16228	3.60555	4.24264
7	2.23607	1.41421	1	1.41421	2.23607	2	1	0	1	2	2.23607	1.41421	1	1.41421	2.23607	2.82843	2.23607	2	2.23607	2.82843	3.60555	3.16228	3	3.16228	3.60555
8	3.16228	2.23607	1.41421	1	1.41421	3	2	1	0	1	3.16228	2.23607	1.41421	1	1.41421	3.60555	2.82843	2.23607	2	2.23607	4.24264	3.60555	3.16228	3	3.16228
9	4.12311	3.16228	2.23607	1.41421	1	4	3	2	1	0	4.12311	3.16228	2.23607	1.41421	1	4.47214	3.60555	2.82843	2.23607	2	5	4.24264	3.60555	3.16228	3
10	2	2.23607	2.82843	3.60555	4.47214	1	1.41421	2.23607	3.16228	4.12311	0	1	2	3	4	1	1.41421	2.23607	3.16228	4.12311	2	2.23607	2.82843	3.60555	4.47214
11	2.23607	2	2.23607	2.82843	3.60555	1.41421	1	1.41421	2.23607	3.16228	1	0	1	2	3	1.41421	1	1.41421	2.23607	3.16228	2.23607	2	2.23607	2.82843	3.60555
12	2.82843	2.23607	2	2.23607	2.82843	2.23607	1.41421	1	1.41421	2.23607	2	1	0	1	2	2.23607	1.41421	1	1.41421	2.23607	2.82843	2.23607	2	2.23607	2.82843
13	3.60555	2.82843	2.23607	2	2.23607	3.16228	2.23607	1.41421	1	1.41421	3	2	1	0	1	3.16228	2.23607	1.41421	1	1.41421	3.60555	2.82843	2.23607	2	2.23607
14	4.47214	3.60555	2.82843	2.23607	2	4.12311	3.16228	2.23607	1.41421	1	4	3	2	1	0	4.12311	3.16228	2.23607	1.41421	1	4.47214	3.60555	2.82843	2.23607	2
15	3	3.16228	3.60555	4.24264	5	2	2.23607	2.82843	3.60555	4.47214	1	1.41421	2.23607	3.16228	4.12311	0	1	2	3	4	1	1.41421	2.23607	3.16228	4.12311
16	3.16228	3	3.16228	3.60555	4.24264	2.23607	2	2.23607	2.82843	3.60555	1.41421	1	1.41421	2.23607	3.16228	1	0	1	2	3	1.41421	1	1.41421	2.23607	3.16228
17	3.60555	3.16228	3	3.16228	3.60555	2.82843	2.23607	2	2.23607	2.82843	2.23607	1.41421	1	1.41421	2.23607	2	1	0	1	2	2.23607	1.41421	1	1.41421	2.23607
18	4.24264	3.60555	3.16228	3	3.16228	3.60555	2.82843	2.23607	2	2.23607	3.16228	2.23607	1.41421	1	1.41421	3	2	1	0	1	3.16228	2.23607	1.41421	1	1.41421
19	5	4.24264	3.60555	3.16228	3	4.47214	3.60555	2.82843	2.23607	2	4.12311	3.16228	2.23607	1.41421	1	4	3	2	1	0	4.12311	3.16228	2.23607	1.41421	1
20	4	4.12311	4.47214	5	5.65685	3	3.16228	3.60555	4.24264	5	2	2.23607	2.82843	3.60555	4.47214	1	1.41421	2.23607	3.16228	4.12311	0	1	2	3	4
21	4.12311	4	4.12311	4.47214	5	3.16228	3	3.16228	3.60555	4.24264	2.23607	2	2.23607	2.82843	3.60555	1.41421	1	1.41421	2.23607	3.16228	1	0	1	2	3
22	4.47214	4.12311	4	4.12311	4.47214	3.60555	3.16228	3	3.16228	3.60555	2.82843	2.23607	2	2.23607	2.82843	2.23607	1.41421	1	1.41421	2.23607	2	1	0	1	2

Şekil 3 Oluşturulan nöronların birbirine olan uzaklıklarını içeren matris

Eğitimde iç çarpım yöntemi kullanıldı. Her bir data ağı girdiğinde bütün nöronların ağırlıklarıyla çarpıldı. Bu iç çarpım sonuçları bir matris içinde tutuldu. Bu iç çarpımlardan maksimum olan yani verilen dataya en benzeyen nöron kazanan olarak belirlendi. Her bir veri için kazanan nöronun index' i daha sonrasında yorumlayabilmek adına matris içinde tutuldu. Nöronların ağırlıkları kazanan nörona olan uzaklıklarına göre kazanan en yakın olan en fazla güncellenecek, en uzak olan en az güncellenecek şekilde yenilendi. İterasyon sayısı ise 400 seçildi.

Nöron sayısı 5'e 5 toplam 25 nöron, sigma 0.5 ve learning rate 0.5 olan bir eğitimin sonucu incelenecektir.

```
#Merkez noktaları ve sigmaları belirlenen 4 adet küme
A = np.random.multivariate_normal([3, 3, 2], 0.4*np.eye(3), 150)
B = np.random.multivariate_normal([3, 3, -2], 0.4*np.eye(3), 150)
C = np.random.multivariate_normal([3, -3, 2], 0.4*np.eye(3), 150)
D = np.random.multivariate_normal([3, -3, -2], 0.4*np.eye(3), 150)
```

Şekil 4 Eğitime giren verilerin merkez noktaları ve sigma değerleri (sigmalar 0.4)

	0	1	2
0	2.26741	-2.70309	-2.02674
1	2.68036	-2.19995	-1.86815
2	2.9912	1.59192	-2.0612
3	3.40723	1.89926	-2.3043
4	2.57052	3.1854	-2.42814
5	2.2461	-2.87436	-2.4369
6	2.82173	-2.61025	-1.77454
7	2.93958	1.4605	-1.79469
8	2.90939	2.54257	-1.75461
9	2.48547	2.65601	-1.52199
10	2.87735	-2.96269	-0.763692
11	2.9221	-2.37292	-0.54328
12	2.91475	0.987953	-0.55664
13	2.82948	2.73718	0.133243
14	2.66273	2.8966	0.626707
15	3.17755	-2.43431	1.67828
16	2.99914	-2.46096	1.45113
17	2.92635	0.472275	1.43631
18	2.48346	2.92991	1.79566
19	2.58758	2.93643	1.56092
20	2.68386	-2.47064	1.74867
21	2.83832	-2.33831	1.47486
22	3.0492	0.168192	1.74454

23	3.34165	3.02408	1.97572
24	2.84989	2.74554	1.49554

Şekil 5 Eğitim sonucu çıkan nöronlara ait ağırlık değerleri

Ağırlık sonuçlarından da anlaşılacağı üzere her bir sınıfı belli bir ağırlık grubu temsil ediyor.

A sınıfını yani 3,3,2 değerlerini en iyi olarak 18,19,23,24 nöronlarının temsil ettiğini,

B sınıfını yani 3,3,-2 değerlerini en iyi olarak 3,4,8,9 nöronlarının temsil ettiğini,

C sınıfını yani 3,-3,2 değerlerini en iyi olarak 15,16,20,21 nöronlarının temsil ettiğini,

D sınıfını yani 3,-3,-2 değerlerini en iyi olarak 0,1,5,6 nöronlarının temsil ettiğini anlayabiliyoruz.

Tam olarak temsil etmeyen diğer ağırlıklar ise yakınlarındaki diğer nöronlara benzedikleri anlaşıyor.

Test kümesi için gelen sonuçları değerlendirelim.

	0		0
0	23	0	0
1	15	1	2
2	23	2	0
3	5	3	3
4	4	4	1
5	23	5	0
6	23	6	0
7	23	7	0
8	5	8	3
9	15	9	2
10	10	10	3
11	15	11	2
12	5	12	3
13	23	13	0
14	15	14	2
15	23	15	0
16	4	16	1
17	3	17	1
18	23	18	0
19	5	19	3
20	4	20	1
21	10	21	3
22	15	22	2

Test kümesindeki ilk 22 elemanın verdiği sonuçlardan da anlaşılacağı üzere 0. Sınıftaki verileri 23. nöron,

1.sınıftakileri 3. Ve 4. nöron (benzer ağırlıklara sahip nöronlar),

2.sınıftakileri 15. nöron

3.sınıftakileri 5. ve 10. nöron (benzer ağırlıklar) kazanmaktadır.

Bu kıyaslamalardan 1. Sorudaki ağın doğru bir şekilde çalıştığı anlaşılmaktadır.

Şekil 7 Kazanan nöronlar

Şekil 6 Verilerin sınıfları

Bunlara ek olarak sigma değeri az iken kazanan nöronların komşu nöronları kendi değerlerini çok güncelleyemiyorlar. Sadece belirli nöronlar bir sınıfı temsil edebiliyor. Komşu nöronların ağırlıkları bir anlam ifade etmemeye başlıyor. Her sınıfı yaklaşık olarak bir nöron temsil ediyor. Yani kapsayıcı bir “Balıkçı ağı” oluşturulamıyor.

Sigma değeri fazla ise bir epok içinde kazanan nöron ve çevresi çok artan bir şekilde güncelleniyor. Ağa farklı sınıfta bir veri verildiğinde ise yine kazanan nöron ve çevresi çok artan bir şekilde güncelleniyor. Yani ağırlıklar bir aşağı bir yukarı çok hızlı gitmeye başlıyor ve ağ yeterli olarak ayırım yapamıyor. Sonuç olarak 2 sınıf iyi temsil edilebilirken diğer 2 sınıf temsil edilse bile tam iyi bir sonuç alınamıyor.

Learning rate az bir değerde iken eğitim her halükârda gerçekleşiyor ancak iyi bir eğitim için gerekli olan epok sayısı daha fazla olabiliyor.

Learning rate belli bir değerden fazla değil ise eğitim daha hızlı gerçekleşiyor ancak çok fazla arttırılırsa sigmanın çok olduğu durumdaki gibi ağırlıklar bir aşağı bir yukarı çok hızlı şekilde gidiyor ve eğitim neredeyse gerçekleşmiyor.

Soru 2

İris verileri txt dosyasının içinden okundu. Yöntem olarak iç çarpım metodu kullandığımız için verilerimizin daha uygun olması adına her bir sütunu kendi içinde -1 ve 1 arasında normalize ettik. Örnek vermek gerekirse

5.1, 3.5, 1.4, 0.2

7, 3.2, 4.7, 1.4

Verilerinde 7 4 5 2 ağırlığına ve 5, 3, 1, 0.1 ağırlığına sahip iki nöron gelirse ikisinde de kazanan 7 4 5 2 ağırlığına sahip nöron olacaktır ve doğru bir sınıflandırma olmayacaktır. Bundan dolayı her bir sütunu kendi içinde normalize ettik. Son sütun için düşünecek olursak 2.5 değeri 1’e 0.2 değeri -1’e yuvarlanır ve iç çarpım metoduyla daha sağlıklı bir sınıflandırma elde edilebilir.

Normalize olan değerler üzerinden her bir sınıfın 4 adet belirleyici özelliğinin ortalamaları bulundu ve bu değerler eğitim sonucunda elde edilecek olan ağırlıklarla kıyaslanmak üzere bir matris içinde kaydedildi.

	0	1	2	3	4
0	0	-0.607778	0.181667	-0.842712	-0.88
1	1	-0.0911111	-0.358333	0.105085	0.0216667
2	2	0.271111	-0.188333	0.543051	0.605

Şekil 8 Sınıfların 4 adet özelliklerinin ortalamaları

Ardından sigma 0.5 ve learning rate 0.5, nöron sayıları ise 5'e 5 olmak üzere 25 adet seçildi. Eğitim sonuçları incelenecektir.

	0	1	2	3
0	-0.278106	-0.239308	-0.0481664	-0.146673
1	-0.395638	-0.187712	-0.329426	-0.393966
2	-0.338647	-0.33781	-0.0391918	-0.128309
3	-0.394853	-0.382662	-0.121408	-0.186358
4	-0.522657	-0.0814032	-0.55818	-0.621797
5	-0.014322	-0.249823	0.205579	0.126833
6	-0.0122116	-0.175463	0.0795403	0.0590452
7	-0.299216	-0.127523	-0.250682	-0.284294
8	-0.499541	-0.099953	-0.511636	-0.531433
9	-0.690401	0.0723452	-0.829362	-0.882567
10	0.263588	-0.27973	0.413591	0.36588
11	0.346475	-0.268146	0.352921	0.239415
12	0.118529	-0.174584	0.252969	0.2475
13	-0.362595	-0.0111968	-0.400504	-0.425918
14	-0.534246	0.0636007	-0.66535	-0.706077
15	0.273679	-0.425402	0.366896	0.249886
16	0.213033	-0.251052	0.424931	0.387096
17	0.140769	-0.301752	0.457175	0.328943
18	0.0653089	-0.168758	0.255285	0.253198
19	-0.0602349	-0.133385	0.0294754	-0.00585612
20	0.278574	-0.214172	0.437075	0.404292
21	0.559947	0.188723	0.617128	0.599803
22	0.197096	-0.139217	0.451975	0.478131

Ağırlık sonuçları ile ortalama matrisini kıyaslayarak inceleyecek olursak

0 sınıfını yani -0.6, 0.18, -0.84, -0.88 değerlerini en iyi olarak 9. ve 14. nöronların temsil ettiğini,

1 sınıfını yani -0.09, -0.35, 0.1, 0.021 değerlerini 5. Ve 6. Nöronları temsil edebileceğini,

2 sınıfını yani 0.27, 0.18, 0.54, 0.605 değerlerini en iyi olarak 21. nöronun temsil ettiğini,

Tam olarak temsil etmeyen diğer ağırlıklar ise yakınlarındaki diğer nöronlara benzedikleri anlaşıyor.

23	0.0778291	-0.214283	0.321143	0.244966
24	0.125077	-0.27165	0.2727	0.179754

Şekil 9 Ağırlık sonuçları

	0		0
0	1	0	3
1	0	1	9
2	1	2	3
3	1	3	21
4	1	4	3
5	1	5	3
6	2	6	15
7	1	7	21
8	0	8	9
9	2	9	21
10	1	10	15
11	0	11	9
12	0	12	9
13	2	13	3
14	1	14	9
15	1	15	3
16	1	16	17
17	2	17	21
18	1	18	21
19	2	19	21
20	0	20	9
21	2	21	17
22	2	22	21

Şekil 11 Sınıflar

Şekil 10 Kazanan nöronlar

Test kümesindeki ilk 22 elemanın verdiği sonuçlar incelenecek olursa sınıflandırmanın iyi bir şekilde yapıldığı anlaşılabılır. Genel olarak

2. sınıfı 21. Nöronun (bazen 15 ve 17 gibi benzer nöronların)

1. sınıfı 3. Nöronun (diğer kümelere benzer olan veriler 2. Sınıf veya 0. Sınıf olarak algılanmış. Genel olarak sınıflandırma hataları bu 1. Sınıftan kaynaklanıyor)

0. sınıfı 9. Nöronun temsil ettiği görülüyor.

Soru 3

Bu sorunun çözümünde minisom kütüphanesi kullanılmıştır.

Birinci soruda kullanmış olduğumuz veriyi train_test_split yardımıyla ayırdık ve som kütüphanesi yardımıyla eğittik.

	0	1	2
0	3.36307	-3.35572	1.9596
1	2.98905	-2.91466	2.3497
2	2.83341	-0.491862	2.43477
3	2.94884	2.46837	2.22443
4	3.11892	3.08187	2.50461

Her sınıfı bir ağırlık grubu temsil etmektedir.

0 nöronu B sınıfını 3,-3,2 değerlerini en iyi olarak temsil etmektedir.

1 nöronu B sınıfını 3,-3,2 değerlerini en iyi olarak temsil etmektedir.

2 nöronu B sınıfını 3,-3,2 değerlerini en iyi olarak temsil etmektedir.

3 nöronu A sınıfını 3,3,2 değerlerini en iyi olarak temsil etmektedir.

4 nöronu A sınıfını 3,3,2 değerlerini en iyi olarak temsil etmektedir.

4.83264	5.21303	5.7694	7.28247	7.90795
4.96587	4.47308	5.38835	7.11772	8.01468
3.41556	3.07349	4.22206	6.22947	6.7611
1.2159	1.50198	3.75778	5.72218	6.15071
1.05128	0.688183	3.73862	5.65551	5.54211

Aktivasyon haritası her bir nöronun verilere olan uzaklığını göstermektedir.

	0	1	2		0
0	1.97784	3.37985	3.03545	0	0
1	3.33642	2.68512	2.38992	1	0
2	3.02347	-3.89338	-2.45626	2	3
3	3.51668	4.38098	1.9407	3	0
4	3.78463	-3.31931	-2.88209	4	3
5	3.76247	-3.96123	-2.20708	5	3
6	3.063	-4.09703	-2.16116	6	3
7	3.16743	-4.10979	1.67055	7	2
8	3.86139	-2.70542	1.87816	8	2
9	3.39763	2.70046	1.5038	9	0
10	3.48258	-2.96507	1.00686	10	2
11	2.4125	2.9644	-2.08101	11	1
12	3.37385	-2.73544	-2.14913	12	3
13	2.9838	1.59847	-1.32896	13	1
14	2.80997	-3.25836	2.63954	14	2
15	3.10061	-2.99265	0.848258	15	2

Test kümesi sonuçları incelendiğinde;

A sınıfını 3,3,2 değerlerini en iyi olarak 0 nöronu temsil etmektedir.

B sınıfını 3,3,-2 değerlerini en iyi olarak 1 nöronu temsil etmektedir.

C sınıfını 3,-3,2 değerlerini en iyi olarak 2 nöronu temsil etmektedir.

D sınıfını 3,-3,-2 değerlerini en iyi olarak 3 nöronu temsil etmektedir.

Sonuçlar 1.1 sorudaki sonuçlarımız ile örtüşmektedir. Ağımızın başarılı bir şekilde çalıştığını bu şekilde anlamaktayız.

	precision	recall	f1-score	support
0.0	0.95	0.97	0.96	38
1.0	0.97	0.95	0.96	38
2.0	0.95	1.00	0.97	37
3.0	1.00	0.95	0.97	37
accuracy			0.97	150
macro avg	0.97	0.97	0.97	150
weighted avg	0.97	0.97	0.97	150

Sınıflandırma raporu 0.97 kesinlik ile ağımızın çalıştığını göstermektedir. Nöronlarımızın hepsi 0.95 üstü bir hassasiyet oranı ile çalışmaktadır.

İris veri kümesini bu ağ ile öbeklediğimizde;

	0	1	2	3
0	7.43271	2.76523	6.52604	2.1265
1	6.83904	3.08826	5.90043	2.21501
2	6.4905	3.25535	5.70064	2.31453
3	5.72058	3.64686	2.80124	0.914983
4	5.48141	3.85331	1.43862	0.239095

Her sınıfı bir ağırlık grubu temsil etmektedir.

0 nöronu İris-virginica türünü en iyi olarak temsil etmektedir.

1 nöronu İris-virginica türünü en iyi olarak temsil etmektedir.

2 nöronu İris-virginica türünü en iyi olarak temsil etmektedir.

3 nöronu İris-versicolor türünü en iyi olarak temsil etmektedir.

4 nöronu İris-setosa türünü en iyi olarak temsil etmektedir.

	0	1	2	3	4
0	1.977	1.39993	1.39665	2.47237	3.93633
1	1.10325	0.879733	0.989312	3.41635	3.98713
2	0.886981	0.69092	1.12094	2.85486	4.18472
3	0.466502	0.758815	1.13921	1.79353	3.64464
4	0.372798	0.838912	1.26205	1.55293	1.98667

Aktivasyon haritası her bir nöronun verilere olan uzaklığını göstermektedir.s

	0	1	2	3		0
0	5.5	3.5	1.3	0.2	0	Iris-setosa
1	5.6	2.5	3.9	1.1	1	Iris-versicolor
2	5.4	3.7	1.5	0.2	2	Iris-setosa
3	6.7	3.3	5.7	2.5	3	Iris-virginica
4	4.6	3.6	1	0.2	4	Iris-setosa
5	5	3.4	1.5	0.2	5	Iris-setosa
6	5	3	1.6	0.2	6	Iris-setosa
7	6.5	2.8	4.6	1.5	7	Iris-versicolor
8	5.4	3	4.5	1.5	8	Iris-versicolor
9	5.1	2.5	3	1.1	9	Iris-versicolor
10	6.9	3.1	5.1	2.3	10	Iris-virginica
11	6.4	2.8	5.6	2.2	11	Iris-virginica
12	5	3.5	1.3	0.3	12	Iris-setosa
13	6.7	3.1	4.7	1.5	13	Iris-versicolor
14	6.4	3.2	4.5	1.5	14	Iris-versicolor
15	6.3	2.5	4.9	1.5	15	Iris-versicolor

Test kümesi sonuçları incelendiğinde;

Verilerin her tür için başarılı bir şekilde öbeklendiğini görmekteyiz.

Sonuçlar 1.2 sorudaki sonuçlarımız ile örtüşmektedir. Ağımızın başarılı bir şekilde çalıştığını bu şekilde anlamaktayız.

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	13
Iris-versicolor	1.00	0.92	0.96	12
Iris-virginica	0.93	1.00	0.96	13
accuracy			0.97	38
macro avg	0.98	0.97	0.97	38
weighted avg	0.98	0.97	0.97	38

Sınıflandırma raporu 0.97 kesinlik ile ağımızın çalıştığını göstermektedir. Iris-setosa ve Iris-versicolor 1.0 hassasiyet ile çalışmasına rağmen Iris-virginica 0.93 hassasiyet ile çalışmaktadır.