

T.C.  
SAKARYA ÜNİVERSİTESİ  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 427 BULANIK MANTIK VE YAPAY  
SİNİR AĞLARINA GİRİŞ  
2. ÖDEV RAPORU**

**G221210045 – MEHMET BOSDANCI  
G221210071 – BİLAL AVCI**

**Dersi Veren : Doç. Dr. M. FATİH ADAK**

## BULANIK MANTIK VERİ SETİ İLE YSA MODELLEMESİ: İŞSİZLİK ORANI TAHMİNİ

### VERİ SETİNİN OLUŞTURULMASI VE ÖN İŞLEME

Veri Üretimi ödev gereksinimleri doğrultusunda, YSA'nın eğitimi için gerekli olan veri seti, ödev 1'de tasarlanan Bulanık Mantık modeli kullanılarak sentetik olarak üretilmiştir. Bunun için VeriSetiOlusturucu.java isimli bir sınıf kodlanmıştır.

Bu sınıf, aşağıdaki sınır değerleri arasında rastgele 4000 adet girdi seti üretmiştir:

- **Gelir:** 0 - 150 (bin TL)
- **Gider:** 0 - 150 (bin TL)
- **Nüfus:** 0 - 200 (Milyon Kişi)

Üretilen her bir rastgele girdi seti, jFuzzyLogic kütüphanesi aracılığıyla .fcl dosyasındaki kurallar sonucunda **İşsizlik Oranı (Çıktı)** hesaplanmıştır ve 4000 satırlık, virgülle ayrılmış (CSV formatında) bir dataset.txt dosyası elde edilmiştir.

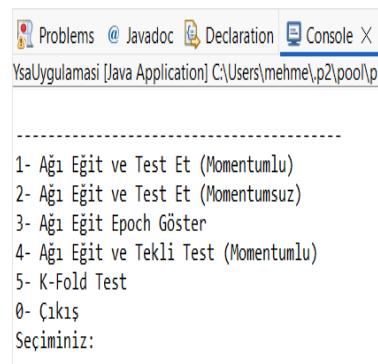
### Veri Normalizasyonu

Yapay sinir ağlarında sigmoid aktivasyon fonksiyonu kullanıldığı için, verilerin 0 ile 1 aralığında olması için ham veriler ağa verilmeden önce Min-Max normalizasyonuna tabi tutulmuştur. Kod içerisinde tanımlanan maksimum değerler (MAX\_GELIR = 150, MAX\_NUFUS = 200 vb.) kullanılarak her bir veri kendi maksimum değerine bölünmüştür ve [0, 1] aralığına indirgenmiştir.

### Eğitim ve Test Seti Ayrımı

4000 satırlık veri seti yüklenikten sonra Collections.shuffle() metodu ile rastgele karıştırılmıştır ve **%75 (3000 Veri)** ağıın eğitimi (Training) için, **%25 (1000 Veri)** ağıın performansının test edilmesi (Testing) için ayrılmıştır.

### Menü ve Veriseti



```
1 88.2301,33.7250,151.6746,6.5000,az
2 43.2864,52.6667,89.8325,9.2091,orta
3 105.5147,130.2476,131.1481,9.1667,orta
4 86.9264,43.6495,20.6458,6.5000,az
5 3.0445,133.8273,143.5498,20.2042,cokyuksek
6 30.4763,130.0037,162.8287,12.5000,yuksek
7 47.9215,146.8868,74.6388,12.5000,yuksek
8 142.2091,77.7914,142.5690,9.1685,orta
9 118.4434,109.1192,0.3258,6.5000,az
10 96.0941,144.1393,94.5064,9.1900,orta
11 91.6508,96.9895,195.5856,9.1667,orta
12 39.9769,18.7165,108.5539,9.1858,orta
13 49.1997,37.9959,113.3288,9.1738,orta
14 21.1593,98.0895,133.7648,20.1265,cokyuksek
15 141.0576,25.7481,49.1451,6.5000,az
16 24.2666,47.1999,197.2920,19.8657,cokyuksek
17 103.1954,110.1008,199.5550,9.1667,orta
18 135.4288,99.6787,187.2266,9.1667,orta
19 20.4791,147.9207,156.5267,20.1723,cokyuksek
20 118.6832,67.4389,10.0765,6.5000,az
21 63.0191,43.0927,192.5398,9.2094,orta
22 99.5016,3.3417,35.7850,6.5000,az
```

Oluşturulan veri setinde ilk 3 sütun girdiler, 4. sütun sayısal çıktı (Regresyon hedefi), 5. sütun ise okunabilirliği artırmak için eklenen bulanık küme etiketidir.

## Seçenek 1

```
Problems @ Javadoc Declaration Console ×
YsaUygulamasi [Java Application] C:\Users\mehme\p2\pool\pl

-----
1- Ağrı Eğit ve Test Et (Momentumlu)
2- Ağrı Eğit ve Test Et (Momentumsuz)
3- Ağrı Eğit Epoch Göster
4- Ağrı Eğit ve Tekli Test (Momentumlu)
5- K-Fold Test
0- Çıkış
Seçiminiz: 1
Veri seti yükleniyor (Momentumlu)...
Ağ eğitiliyor...
Eğitim Tamamlandı!
Eğitim Hatası (MSE): 0.0015801098202664278
Test Hatası (MSE): 0.0016494236913485853
-----
```

Eğitim işleminde **MomentumBackpropagation** algoritması kullanılmıştır. Öğrenme katsayısı (Learning Rate) 0.2 ve Momentum değeri 0.7 olarak ayarlanmıştır. Eğitim sonucunda MSE (Ortalama Karesel Hata) değerleri hem eğitim hem de test seti için oldukça düşük (binde 1-2 seviyelerinde) çıkarak başarılı bir öğrenme gerçekleştiği gözlemlenmiştir.

## Seçenek 2

```
Problems @ Javadoc Declaration Console ×
YsaUygulamasi [Java Application] C:\Users\mehme\p2\pool\pl

-----
1- Ağrı Eğit ve Test Et (Momentumlu)
2- Ağrı Eğit ve Test Et (Momentumsuz)
3- Ağrı Eğit Epoch Göster
4- Ağrı Eğit ve Tekli Test (Momentumlu)
5- K-Fold Test
0- Çıkış
Seçiminiz: 2
Veri seti yükleniyor (Momentumsuz)...
Ağ eğitiliyor (Momentumsuz)...
Eğitim Tamamlandı!
Eğitim Hatası (MSE): 0.002113051342604723
Test Hatası (MSE): 0.0020750667022837238
-----
```

Eğitim algoritması olarak standart **BackPropagation** (Momentumsuz) tercih edilmiştir. Momentumun yokluğu, ağın yerel minimumlardan kaçmasını zorlaştırdığı için hata oranlarının (MSE), momentumlu eğitime kıyasla bir miktar daha yüksek çıktıgı görülmüür. Bu durum, momentum parametresinin öğrenme üzerindeki olumlu etkisini doğrulamaktadır.

## Seçenek 3

```
Problems @ Javadoc Declaration Console ×
YsaUygulamasi [Java Application] C:\Users\mehme\p2\pool\plugins\

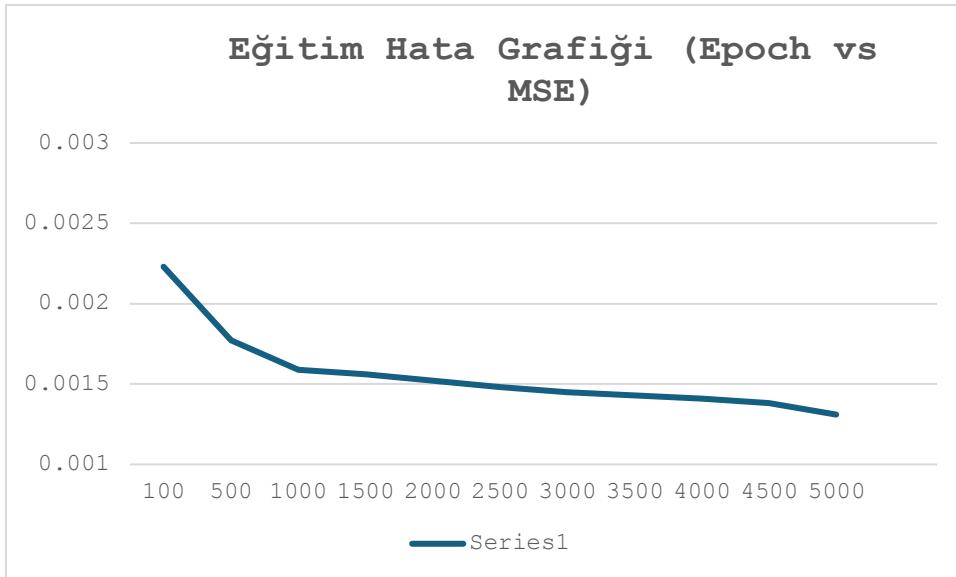
1- Ağrı Eğit ve Test Et (Momentumlu)
2- Ağrı Eğit ve Test Et (Momentumsuz)
3- Ağrı Eğit Epoch Göster
4- Ağrı Eğit ve Tekli Test (Momentumlu)
5- K-Fold Test
0- Çıkış
Seçiminiz: 3
Veri seti yükleniyor (Momentumlu)...
Ağ eğitiliyor...
Epoch: 100 | Eğitim Hatası: 0.002233312931699075
Epoch: 200 | Eğitim Hatası: 0.0020064274248310725
Epoch: 300 | Eğitim Hatası: 0.0018975540899449216
Epoch: 400 | Eğitim Hatası: 0.0018357052095073242
Epoch: 500 | Eğitim Hatası: 0.0017732782958304616
Epoch: 600 | Eğitim Hatası: 0.0016919257664408383
Epoch: 700 | Eğitim Hatası: 0.0016414618712035816
Epoch: 800 | Eğitim Hatası: 0.0016183634302154796
Epoch: 900 | Eğitim Hatası: 0.0016064317869127925
Epoch: 1000 | Eğitim Hatası: 0.0015981493954620883
Epoch: 1100 | Eğitim Hatası: 0.0015909431215255759
Epoch: 1200 | Eğitim Hatası: 0.001584060000002243
Epoch: 1300 | Eğitim Hatası: 0.0015772682443295472
Epoch: 1400 | Eğitim Hatası: 0.0015704992551505213
Epoch: 1500 | Eğitim Hatası: 0.0015637281555697048
Epoch: 1600 | Eğitim Hatası: 0.001556986456229235
Epoch: 1700 | Eğitim Hatası: 0.0015499513131013937
Epoch: 1800 | Eğitim Hatası: 0.0015427387681032976
Epoch: 1900 | Eğitim Hatası: 0.0015351609562219968
Epoch: 2000 | Eğitim Hatası: 0.0015271607225024215
Epoch: 2100 | Eğitim Hatası: 0.0015187823681315348
Epoch: 2200 | Eğitim Hatası: 0.0015101997943673673
Epoch: 2300 | Eğitim Hatası: 0.0015016802313940756
Epoch: 2400 | Eğitim Hatası: 0.0014934818920189737
Epoch: 2500 | Eğitim Hatası: 0.0014857686305181019
Epoch: 2600 | Eğitim Hatası: 0.0014786044029134293
Epoch: 2700 | Eğitim Hatası: 0.0014719961987721894
```

Epoch:	Eğitim Hatası:
2700	0.0014719961987721894
2800	0.0014659306172452266
2900	0.0014603851630140916
3000	0.0014553244228643803
3100	0.0014506965110110226
3200	0.0014464359109753
3300	0.0014424702177581153
3400	0.0014387265279832623
3500	0.001435135261894417
3600	0.0014316312823697685
3700	0.001428153044474364
3800	0.0014246402889599905
3900	0.001421029825652385
4000	0.0014172474844584497
4100	0.0014131926213636844
4200	0.001408710208128008
4300	0.0014035465836408086
4400	0.0013972960102410224
4500	0.0013893766859825105
4600	0.0013791054175784888
4700	0.0013659068086243206
4800	0.0013494827189849156
4900	0.001330251988258704
5000	0.0013116541016387736

Eğitim Tamamlandı!

Eğitim Hatası (MSE): 0.0025928834537294544  
Test Hatası (MSE): 0.002711325503058206

Ağın öğrenme sürecini (konverjans) anlık olarak takip edebilmek için bu modül geliştirilmiştir. Eğitim sırasında her 100 Epoch'ta bir, ağın o anki toplam hata değeri konsola yazdırılmıştır. Ekran çıktısında görüldüğü üzere, Epoch sayısı arttıkça hata değeri istikrarlı bir şekilde azalmaktadır.



#### Seçenek 4

```

Problems @ Javadoc Declaration Console X
YsaUygulaması [Java Application] C:\Users\mehme\p2\pool\plugins\org.ec
-----
1- Ağrı Eğit ve Test Et (Momentumlu)
2- Ağrı Eğit ve Test Et (Momentumsuz)
3- Ağrı Eğit Epoch Göster
4- Ağrı Eğit ve Tekli Test (Momentumlu)
5- K-Fold Test
0- Çıkış
Seçiminiz: 4
Tüm veri seti ile ağ eğitiliyor (Lütfen bekleyiniz)...
Eğitim bitti. Şimdi test zamanı!
Gelir Giriniz (0-150): 100
Gider Giriniz (0-150): 70
Nüfus Giriniz (0-200): 80
|-----
Tahmin Edilen İşsizlik Oranı: %8,85
|-----
```

Bu seçenekte veri setinin tamamı kullanılarak eğitilmiş ve ardından kullanıcıdan manuel veri girişi beklenmiştir. Örnek olarak Ödev 1'de kullanılan girdiler (**Gelir: 100, Gider: 70, Nüfus: 80**) sisteme girilmiştir. Ağın bu girdiler için ürettiği tahmin **%8,85** olmuştur. Bu değer, Bulanık Mantık modelinin ürettiği sonuca (%9,19) oldukça yakındır ve YSA modelinin bulanık mantık kurallarını başarıyla öğrendiğini göstermektedir.

#### Seçenek 5

```

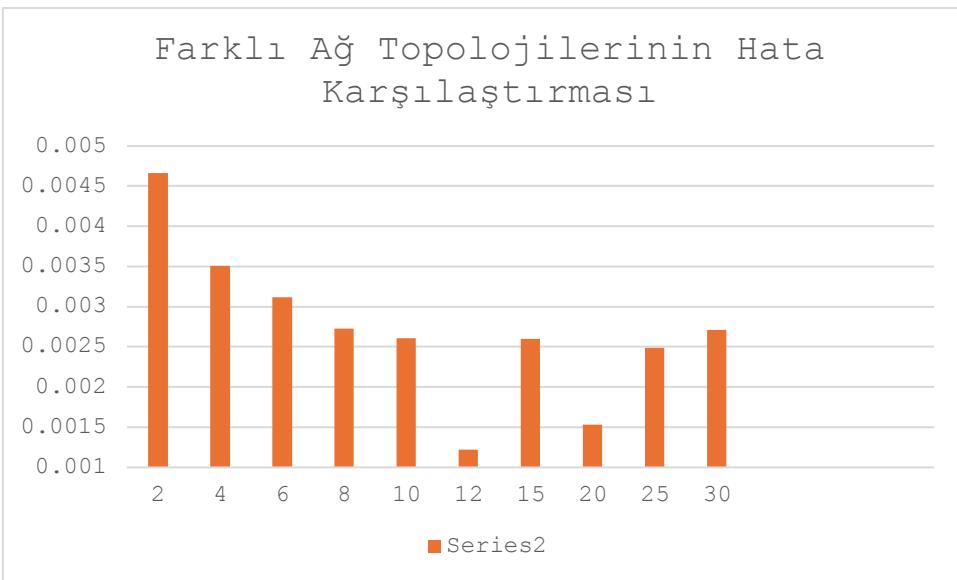
Problems @ Javadoc Declaration Console X
YsaUygulaması [Java Application] C:\Users\mehme\p2\pool\plugins\org.ec
-----
1- Ağrı Eğit ve Test Et (Momentumlu)
2- Ağrı Eğit ve Test Et (Momentumsuz)
3- Ağrı Eğit Epoch Göster
4- Ağrı Eğit ve Tekli Test (Momentumlu)
5- K-Fold Test
0- Çıkış
Seçiminiz: 5
K Değerini Giriniz : 10
Veri seti yükleniyor ve karıştırılıyor...
|-----
10-Fold Cross Validation Başlıyor...
Fold 1 -> Eğitim MSE: 0,00224 | Test MSE: 0,00190
Fold 2 -> Eğitim MSE: 0,00289 | Test MSE: 0,00274
Fold 3 -> Eğitim MSE: 0,00273 | Test MSE: 0,00288
Fold 4 -> Eğitim MSE: 0,00273 | Test MSE: 0,00293
Fold 5 -> Eğitim MSE: 0,00276 | Test MSE: 0,00299
Fold 6 -> Eğitim MSE: 0,00235 | Test MSE: 0,00250
Fold 7 -> Eğitim MSE: 0,00237 | Test MSE: 0,00287
Fold 8 -> Eğitim MSE: 0,00247 | Test MSE: 0,00222
Fold 9 -> Eğitim MSE: 0,00227 | Test MSE: 0,00227
Fold 10 -> Eğitim MSE: 0,00293 | Test MSE: 0,00294
|----- SONUCLAR -----
Ortalama Eğitim Hatası: 0.0025726431650675077
Ortalama Test Hatası: 0.0026214459618498502
|-----
```

Veri seti 10 eşit parçağa bölünmüştür, her adımada 9 parça eğitim, 1 parça test için kullanılmıştır. Sonuç tablosunda görüldüğü üzere, 10 farklı katlammanın hata oranları birbirine çok yakındır (Standart sapması düşüktür).

#### 10 Adet Ağda Deneme

Ağın giriş katmanında 3 nöron (Gelir, Gider, Nüfus) ve çıkış katmanında 1 nöron (İşsizlik Oranı) sabit tutulmuştur. Ancak, ağa öğrenme kapasitesini doğrudan etkileyen **ara katman (gizli katman) nöron sayısı** değiştirilerek 10 farklı topoloji test edilmiştir.

Tüm denemelerde öğrenme katsayısı (Learning Rate) **0.2**, Momentum değeri **0.7** ve maksimum iterasyon sayısı **5000** olarak sabit tutulmuş; sadece gizli katmandaki nöron sayısı değiştirilmiştir.



Elde edilen **Test Hatası (MSE)** sonuçları yukarıdaki ekran görüntülerinde/tabloda sunulmuştur. Bu sonuçlara göre:

- Düşük Nöron Sayısı (Underfitting):** Ara katmanda 2 veya 4 nöron kullanıldığında, ağıın kapasitesi veri setindeki karmaşık ilişkileri öğrenmekte yetersiz kalmış ve hata oranları yüksek çıkmıştır (Örn: 2 nöron için MSE ~0.0047).
- Optimum Nokta:** Nöron sayısı artırıldıkça hata oranı düşüş eğilimi göstermiş ve **12 Gizli Nöron** kullanıldığında **0.00122** ile en düşük test hatasına (Minimum MSE) ulaşılmıştır.
- Yüksek Nöron Sayısı:** 12 nörondan sonra (15, 25, 30 vb.) hata oranında belirgin bir düşüş gözlenmemiştir, aksine bazı durumlarda hata tekrar yükselme eğilimi göstermiştir. Ayrıca gereksiz nöron artışı, ağıın işlem yükünü artırmakta ve aşırı öğrenme (overfitting) riski doğurmaktadır.

Yapılan 10 farklı deneme sonucunda, en düşük hata oranını veren ve en kararlı yapıyı sergileyen **3-12-1 (3 Girdi, 12 Gizli, 1 Çıktı)** topolojisi, projenin nihai modeli olarak seçilmiştir.