

**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLŞİM BİLİMLERİ FAKÜLTESİ**  
**BULANIK MANTIK VE YAPAY SİNİR AĞLARINA GİRİŞ**

**Yapay Sinir Ağları İle**  
**Mikrodalga Fırın Eğitimi**  
**2.ÖDEV**

**AD-SOYAD** : Zeynep İnan  
**NUMARA** : B201210004  
**DERS GRUBU** : 1.Öğretim  
**E-POSTA** : [zeynep.inan1@ogr.sakarya.edu.tr](mailto:zeynep.inan1@ogr.sakarya.edu.tr)

## 1. Giriş

Bu proje, bulanık mantık yöntemleriyle tasarlanan bir mikrodalga fırın kontrol sistemi veri setinin yapay sinir ağı ile eğitilmesini ve test edilmesini amaçlamaktadır.

Projenin ana hedefi, mikrodalga fırın kontrolü için optimize edilmiş bir model oluşturmak ve bu modelin gerçek dünya verileri üzerinde etkinliğini değerlendirmektir. Veri seti, bir önceki ödevde verilen random girdi değerleri yardımıyla üretilmiştir.

Bu çalışmanın katkısı, mikrodalga fırın kontrol sistemlerinin geliştirilmesinde yapay sinir ağı teknikleri kullanarak etkin ve akıllı bir model oluşturmaktır.

## 2.Proje Süreci

- Bulanık mantık yardımıyla geliştirilen bir önceki model yardımıyla 4000 satırlık bir veri seti oluşturulmuştur. Veri seti 4 girdi 1 çıktıdan oluşmaktadır.

```
String dosyaYolu = "C:\\Users\\zeyse\\Desktop\\Repos\\BulanikMantik\\BulanikOdevi\\dataSet.cs";
try {
    BufferedWriter writer = new BufferedWriter(new FileWriter(dosyaYolu));
    Oven oven = new Oven();
    int min = -10;
    int max = 70;
    double sicaklik;
    double agirlik;
    double guc;
    double yiyecekTur;
    double sure;
    Random random = new Random();

    for (int i=0; i<4000; i++) {
        sicaklik = random.nextInt(max - min + 1) + min;
        max=500;
        agirlik = random.nextInt(max+1);
        max=500;
        guc = random.nextInt(max+1);
        max=140;
        yiyecekTur = random.nextInt(max+1);
        sure = oven.getSure(sicaklik, agirlik, guc, yiyecekTur);

        String metin = sicaklik + " " + agirlik + " " + guc + " " + yiyecekTur + " " + sure;
        writer.write(metin);
        writer.newLine();
    }
    System.out.println("başarılı");
    writer.close();
}
```

- Üretilen veri seti bir dosya işlemlerinin gerçekleştiği class içerisinde eğitim ve test veri setlerine ayrılmıştır. Ayırma işlemi %75, %25 şeklinde random olarak gerçekleştirilmiştir.
- Bu veriler üzerinde min-max normalizasyonu uygulanarak DataSetler üretilmiştir.
- Momentumlu ve momentumsuz back propagation için iki ayrı sınıf tanımlanmıştır.
- İki yöntem için de DataSetlere eğitim ve test işlemleri uygulanarak hata oranları hesaplanmıştır.
- Momentumsuz back propagation için epoch hataları bulunup grafik çizdirilmiştir.
- Bu ağ yapıları parametreleri değiştirilerek 10 defa denenmiş analiz edilip en iyi ağ modeli projede kullanılmıştır.

## 3.Deney ve Sonuçlar:

### Örnek 1

```
araKatmanNeronSayisi: 20
momentum: 0,4
ogrenmeKatsayisi: 0,1
maxError: 0,0001
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.0034800443801863198
Momentumlu test hata: 0.007644213324511764
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.003929025682446146
Momentumsuz test hata: 0.008216000806590764
```

### Örnek 2

Nöron sayısı:25, momentum:0.1, öğrenme katsayısı: 0.3

```
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.002283953696703351
Momentumlu test hata: 0.005204773940146384
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.002933864792443315
Momentumsuz test hata: 0.006192798499002761
```

### Örnek 3

```
araKatmanNöronSayisi: 25
momentum: 0,4
öğrenmeKatsayisi: 0,1
maxError: 0,0001|
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.003147076518489761
Momentumlu test hata: 0.006353271720708866
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.0037133457821977027
Momentumsuz test hata: 0.007408937593493995
```

### Örnek 4

```
araKatmanNöronSayisi: 30
momentum: 0,4
öğrenmeKatsayisi: 0,1
maxError: 0,0001
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.003899552767182335
Momentumlu test hata: 0.00797295944392527
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.00367926737011427
Momentumsuz test hata: 0.007508392421469671
```

### Örnek 5

```
araKatmanNöronSayisi: 25
momentum: 0,1
öğrenmeKatsayisi: 0,1
maxError: 0,0001
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.003690893191516055
Momentumlu test hata: 0.007056686456255981
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.003721047639444479
Momentumsuz test hata: 0.007093606629102303
```

### Örnek 6

```
araKatmanNöronSayisi: 25
momentum: 0,8
öğrenmeKatsayisi: 0,1
maxError: 0,0001
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.0038410510651241286
Momentumlu test hata: 0.007271484883448922
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.0038801763194164403
Momentumsuz test hata: 0.007088508318893707
```

### Örnek 7

```
araKatmanNeronSayisi: 25
momentum: 0,1
ogrenmeKatsayisi: 0,5
maxError: 0,0001
epoch: 1000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.0023652601330581805
Momentumlu test hata: 0.0055729049072744995
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.001999509820127273
Momentumsuz test hata: 0.004517347133887354
```

## Örnek 8

Nöron sayısı:15, momentum:0.4, öğrenme katsayısı: 0.3

---

```
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.00467263258428159
Momentumlu test hata: 0.00946685997599312
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.005249534782236106
Momentumsuz test hata: 0.010433532307797982
Epoch hatalar
```

## Örnek 9

```
araKatmanNeronSayisi: 25
momentum: 0,1
ogrenmeKatsayisi: 0,5
maxError: 0,0001
epoch: 100
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.004520067207391356
Momentumlu test hata: 0.009303695346864226
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.004483089061214492
Momentumsuz test hata: 0.009197754171343843
```

## Örnek 10

```
araKatmanNeronSayisi: 25
momentum: 0,1
ogrenmeKatsayisi: 0,5
maxError: 0,0001
epoch: 2000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.001948378246037996
Momentumlu test hata: 0.003620991479412888
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.0019761318242590203
Momentumsuz test hata: 0.003547452565696451
```

## Örnek 11

```
araKatmanNeronSayisi: 25
momentum: 0,1
ogrenmeKatsayisi: 0,5
maxError: 0,0001
epoch: 3000
Momentumlu başladı
Momentumlu eğitim tamamlandı
Momentumlu eğitimdeki hata: 0.0018274206455827595
Momentumlu test hata: 0.0038363632132731914
Momentumsuz başladı
momentumsuz eğitim tamamlandı
Momentumsuz eğitimdeki hata: 0.0016819983628224217
Momentumsuz test hata: 0.003471680506551402
```

## 4.Sonuç

1,2,3,4 örneklerinden yola çıkılarak ara katman nöron sayısı 15 olarak seçilmiştir. Nöron sayısı düştükçe, 30'a çıktığında hata oranları artmaktadır. 15-25 arası neredeyse aynı değerleri vermektedir. Bu yüzden 15 değerine karar verilmiştir.

3,5,6 örneklerinden yola çıkılarak momentum 0,4 olarak seçilmiştir. Momentum değerini arttırdıkça hata oranları büyük oranda değiştiği için 0,4 değeri kullanılmıştır.

5,7,8 örneklerinden yola çıkarak hata oranlarında düşüşten dolayı öğrenme katsayısı parametresi 0,3 olarak seçilmiştir. Öğrenme katsayısı çok düştüğünde geç öğreniyor arttığında ise hızlı unutabileceği için 0,3 olarak seçilmiştir.

7,8,9,10,11 örneklerine bakılarak epoch sayısı 1000 olarak seçilmiştir. Epoch sayısını düşürdüğümüz zaman ağ yeterince öğrenemediği için hata oranı artmaktadır. 1000-3000 aralığında ise hata oranlarında kayda değer bir değişim gözlenmediği için algoritmayı yavaşlatmamak amacıyla 1000 epoch olmasına karar verilmiştir.

Bu değerlerden yola çıkılarak örnek 8 seçilmiştir.

Momentumsuz algoritmanın örnek 8 için epoch hata oranları;

