

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**İLERİ YÖNLÜ YAPAY SİNİR AĞLARINDA KÜÇÜK  
DÜNYA AĞI YAKLAŞIMI VE UYGULAMALARI**

**DOKTORA TEZİ**

**OKAN ERKAYMAZ**

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**

**Tez Danışmanı : Prof. Dr. Nejat YUMUŞAK**

**Ortak Danışman : Prof. Dr. Mahmut ÖZER**

**Aralık 2012**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

İLERİ YÖNLÜ YAPAY SİNİR AĞLARINDA KÜÇÜK  
DÜNYA AĞI YAKLAŞIMI VE UYGULAMALARI

DOKTORA TEZİ

Okan ERKAYMAZ

Enstitü Anabilim Dalı : Bilgisayar ve Bilişim Mühendisliği

Bu tez 26/12/2012 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Prof. Dr.  
Nejat YUMUŞAK  
Jüri Başkanı



Doç. Dr.  
İbrahim ÇİL  
Üye



Doç. Dr.  
Resul KARA  
Üye



Doç. Dr.  
Ali ÖZTÜRK  
Üye



Yrd. Doç. Dr.  
Ali GÜLBAĞ  
Üye

## TEŐEKKÜR

Bu alıőmanın hazırlanmasında, desteklerini esirgemeyen danıőman hocam Prof. Dr. NEJAT YUMUŐAK' a teőekkürü bir bor bilirim. Akademik bakıő aısıyla tezimin her anında desteęini gördüğüm saygıdeęer hocam Prof. Dr. Mahmut ÖZER' e ilgi ve desteklerinden dolayı minnet ve őükranlarımı sunarım. Tezim süresince destekleri ile beni yalnız bırakmayan Yrd. Do. Dr. Muhammet UZUNTARLA ve Yrd. Do. Dr. Ergin YILMAZ' a dostlukları için teőekkür ederim.

Ek olarak tez süresince anlayıő ve hoőgörülerinden dolayı deęerli ailem, Annem Sultan ERKAYMAZ' a, Eőim Öğr. Gör. Hande ERKAYMAZ'a ve kızım İclal Sena ERKAYMAZ' a sonsuz teőekkürlerimi ve őükranlarımı sunarım.

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ .....	vii
ŞEKİLLER LİSTESİ.....	ix
TABLolar LİSTESİ .....	xv
ÖZET .....	xvi
SUMMARY .....	xvii
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
YAPAY SİNİR AĞLARI.....	8
2.1. YSA' nın Tarihsel Gelişimi.....	8
2.2. Biyolojik Sinir Ağı ve Nöron .....	9
2.3. YSA'da Kullanılan Kavramlar .....	10
2.3.1. İşlem birimi(Nöron) .....	11
2.3.2. Aktivasyon fonksiyonları.....	12
2.3.3. Katmanlar ve ağırlıklar .....	14
2.4. Yapay Sinir Ağlarının sınıflandırılması.....	15
2.4.1. Yapısal (Topolojik) sınıflandırılma.....	15
2.4.1.1. İleri yönlü ağ yapısı.....	15
2.4.1.2. Geri dönüşümlü ağ yapısı .....	16
2.4.2. Öğrenme algoritmasına göre sınıflandırma .....	17
2.4.2.1. Danışmanlı öğrenme .....	17

2.4.2.2. Danışmansız öğrenme .....	17
2.4.2.3. Destekleyici öğrenme.....	18
2.5. Uygulama Alanına Göre Sınıflandırma .....	19
2.5.1. Çevrim içi (Online) öğrenme.....	19
2.5.2. Çevrim dışı (Offline) öğrenme .....	19
2.6. YSA Öğrenme Kuralları .....	19
2.7. Algılayıcı Ağ Modeli .....	20
2.7.1. Algılayıcı model .....	21
2.7.2. Çok katmanlı algılayıcı model.....	22
2.8. Performans(Doğruluk) Ölçüm Kriterleri .....	24
2.9. Çapraz Doğrulama Metodu ve N Kat Çapraz Doğrulama Yöntemi. ..	26

### BÖLÜM 3.

KARMAŞIK AĞLAR VE ÇİZGE TEORİSİ .....	28
3.1. Karmaşık Ağlar .....	28
3.1.1. Küçük Dünya ağları .....	29
3.1.1.1. Watts-Strogatz modeli.....	30
3.1.1.2. Newman-Watts Küçük Dünya ağı modeli .....	33
3.1.1.3. Latora ve Marchiori metodu .....	34
3.1.1.4. Simard ileri yönlü Küçük Dünya ağ modeli .....	35
3.2. Çizge Teorsisi .....	37
3.2.1. Bağlantı matrisi.....	38
3.2.2. En kısa yolun bulunması .....	39
3.2.2.1. Floyd-Warshall algoritması .....	39

### BÖLÜM 4.

KARMAŞIK AĞ YAZILIMININ HAZIRLANMASI VE KULLANIMI .....	42
4.1. Karmaşık Ağ Yazılımının Hazırlanması .....	42
4.2. Karmaşık Ağ Yazılımının Kullanımı.....	43
4.2.1. Veri setinin normalizasyonu .....	43
4.2.2. Network topolojisinin oluşturulması.....	44

4.2.3. Küçük Dünya ağı tasarımı .....	45
4.2.3.1. $D_{Global}$ parametresinin hesaplanması .....	48
4.2.3.2. $D_{Lokal}$ parametresinin hesaplanması .....	49
4.2.3.3. $D_{global}$ ve $D_{lokal}$ parametreleri kullanılarak Küçük Dünya ağının tespiti .....	51
4.2.4. Küçük Dünya ağı YSA algoritması.....	52
4.2.4.1. Function netcreate(parametre: YBS): .....	54
4.2.4.2. Function initializes(): .....	57
4.2.4.3. Function designed_cross_data(): .....	58
4.2.4.4. Function cffw():.....	60
4.2.4.5. Function cbackpropagate(): .....	61
4.2.4.6. Function cffw_egitimout():.....	63
4.2.4.7. Function cffw_testout():.....	63
4.2.4.8. Function ctesthatahesapla(): .....	64
4.2.5. Ara katman sayısının hesaplanması .....	67

## BÖLÜM 5.

İLERİ YÖNLÜ YAPAY SİNİR AĞLARINDA KÜÇÜK DÜNYA AĞLARI UYGULAMALARI.....	68
5.1. Bağımsız Bağlantı Yenileme Metodu ile Oluşturulan Küçük Dünya Ağının Öğrenme Performansının Araştırılması .....	68
5.2. İleri Yönlü Watt-Strogatz Küçük Dünya Ağının Veri Seti Boyutuna Göre Öğrenme Performansının Araştırılması .....	73
5.3. İleri Yönlü Newman-Watts Küçük Dünya Ağlarının Araştırılması ...	80
5.3.1. İleri yönlü Newman-Watts Küçük Dünya ağı YBS aralığının araştırılması_.....	80
5.3.2. İleri yönlü Newman-Watts Küçük Dünya ağının veri seti boyutuna göre öğrenme performansının araştırılması .....	84
5.4. Karesel Olmayan İleri Yönlü Watt-Strogatz Küçük Dünya Ağ Mimarisinin Epilepsi Teşhisinde Kullanımı .....	88

5.5. Havalı Güneş Kolektörlerinin Termal Performansının Watts-Strogatz Küçük Dünya Ağları ile Tahmini .....	94
5.6. Yönlendirilmiş Yonga Levha (OSB) üretimde Eğilme Direnci Parametrelerinin Watt-Strogatz Küçük Dünya Ağları ile Tahmini .....	99
5.7. Şeker Hastalığı Teşhisinde Küçük Dünya Ağları Kullanılarak Bir Karar Destek Sisteminin Oluşturulması .....	106
<b>BÖLÜM 6.</b>	
<b>SONUÇ VE ÖNERİLER .....</b>	<b>112</b>
6.1. SONUÇLAR .....	112
6.2. ÖNERİLER .....	114
<b>KAYNAKLAR .....</b>	<b>116</b>
<b>ÖZGEÇMİŞ .....</b>	<b>124</b>

## SİMGELER VE KISALTMALAR LİSTESİ

YSA	Yapay sinir ağları
İYYSA	İleri yönlü yapay sinir ağları
SOM	Özdüzenleyici haritalar
ART	Uyarlamalı rezonans kuramı
ADALINE	Adaptive linear model
LVQ	Vektör kuantalamalı öğrenme
$X_i$	Ağ girişleri
K	Toplam fonksiyonu sonucu
$w_i$	Ağırlıklar
$\phi$	Eşik değeri
A	Öğrenme katsayısı
E	Karesel hata
M	Momentum katsayısı
$\delta$	Nöron çıkış hatası
$y_m$	Nöron çıkışı
$\Delta W$	Nöronlar arası ağırlıktaki değişim
MSE	Ortalama karesel hata
MAE	Ortalama mutlak hata
RMSE	Kök ortalama karesel hata



$R^2$	Belirlilik katsayısı
$D_{Global}$	Global bağlantı uzunluk katsayısı
$D_{lokal}$	Lokal bağlantı uzunluk katsayısı
MYBS	Maksimum olası yenilecek bağlantı sayısı
YBS	Yenilenen bağlantı sayısı
$k$	Ağın derecesi
$p$	Ağın bağlantı değiştirme olasılığı
$C$	Kümelenme katsayıları
$L$	Karakteristik yol uzunluğu
$d_{ij}$	$i$ düğümünden $j$ düğümüne geçilen bağlantı sayısı
$N$	Çizgede bulunan düğüm sayısını
$k_i$	$i$ . düğümün derecesi
$G_iBS$	$i$ . düğümün komşuları arasında var olan bağlantı sayısı
EEG	Elektroansefalografi
$T_o$	Kolektör giriş sıcaklığı
$T_i$	Kolektör çıkış sıcaklığı
$T_{sw}$	Depo suyu sıcaklığı
$T_s$	Kolektör yüzey sıcaklığı
$T_{amb}$	Ortam sıcaklığı
$\eta$	Kolektör ışıınım performansı
OSB	Yönlendirilmiş yonga levha
MOR	Eğilme direnci

## ŞEKİLLER LİSTESİ

Şekil 2.1.	Sinir hücresi yapısı .....	10
Şekil 2.2.	Yapay işlem birimi yapısı .....	11
Şekil 2.3.	İleri yönlü ağ topolojisi .....	16
Şekil 2.4	Geri dönüşümlü ağ topolojisi.....	17
Şekil 2.5.	Danışmanlı öğrenme blok diyagramı .....	17
Şekil 2.6.	Danışmansız öğrenme blok diyagramı .....	18
Şekil 2.7.	Destekleyici öğrenme blok diyagramı.....	18
Şekil 2.8.	Algılayıcı yapısal modeli .....	21
Şekil 2.9.	Eğitim sürecinde ağın ezberlemesi süreci .....	26
Şekil 3.1.	Watts-Strogatz tarafından önerilen düzenli, Küçük Dünya ve rastsal ağların topolojik yapısı .....	31
Şekil 3.2.	Watts-Strogatz Küçük Dünya ağı bağlantı yenileme işlemi .....	31
Şekil 3.3.	Bağlantı değiştirme olasılığı ile $C$ ve $L$ 'nin değişimi.....	33
Şekil 3.4.	Newman-Watts yeni bağlantı ekleme işlemi .....	34
Şekil 3.5.	Bağlantı değiştirebilme olasılığı ( $p$ ) ile $D_{global}$ ve $D_{lokal}$ 'in değişimi .....	35
Şekil 3.6.	İleri yönlü yapay sinir ağları için, düzenli, Küçük Dünya ve rastsal ağ topolojileri şeması .....	36
Şekil 3.7.	Tasarlanmış yönsüz çizge modeli .....	38
Şekil 3.8.	Bağlantı matrisi.....	38
Şekil 3.9.	Başlangıç mesafe matrisi.....	40
Şekil 3.10.	Floyd-Warshall algoritmasının sözde kodu .....	40
Şekil 3.11.	Mesafe matrisi.....	41
Şekil 4.1.	Tasarlanan kullanıcı arayüzü .....	42
Şekil 4.2.	Yazılımın normalizasyon fazı ve $c\#$ kodu .....	43
Şekil 4.3.	Ağ oluşturma paneli	44

Şekil 4.4.	Ağ oluşturma penceresi, a) gizli katman seçimi ekranı, b) gizli katman nöron sayısı belirleme ekranı .....	44
Şekil 4.5.	Küçük Dünya ağı tasarım ekranı .....	45
Şekil 4.6.	Ağ bağlantı uzunluk katsayılarının hesaplanması için kullanılan akış diyagramı .....	47
Şekil 4.7.	Dglobalhesap fonksiyonuna ait sözde kod bloğu .....	48
Şekil 4.8.	Dlocalhesap fonksiyonuna ait sözde kod bloğu.....	49
Şekil 4.9.	8-12-1 topolojisine sahip bir YSA için YBS ile Dglobal ve Dlokal değişimi.....	51
Şekil 4.10.	3-8-8-2 topolojisine sahip bir YSA için YBS ile Dglobal ve Dlokal değişimi.....	52
Şekil 4.11.	Öğrenim algoritması akış diyagramı .....	53
Şekil 4.12.	netcreate fonksiyonuna ait sözde kod bloğu .....	55
Şekil 4.13.	initializes fonksiyonuna ait sözde kod bloğu .....	57
Şekil 4.14.	designed_cross_data fonksiyonuna ait sözde kod bloğu .....	58
Şekil 4.15.	cffw fonksiyonuna ait sözde kod bloğu.....	61
Şekil 4.16.	cbackpropagate fonksiyonuna ait sözde kod bloğu .....	62
Şekil 4.17.	cffw_egitimout fonksiyonun oluşturduğu eğitimsonuc dizini ve içeriği .....	63
Şekil 4.18.	cffw_testout fonksiyonun oluşturduğu testsonuc dizini ve içeriği .....	64
Şekil 4.19.	ctesthatahesapla fonksiyonuna ait C# kod bloğu .....	65
Şekil 4.20.	Çapraz doğrulama bölümü. ....	66
Şekil 4.21.	Eğitim sonucu çıktı veri ızgarası görünümü. ....	67
Şekil 4.22.	Ara katman nöron sayısı hesaplama butonu. ....	67
Şekil 5.1.	5x5 topolojiye sahip ileri yönlü yapay sinir ağı için, düzenli, rastsal ve Küçük Dünya Ağları topolojik gösterimi. ....	69
Şekil 5.2.	5x5 İleri yönlü yapay sinir ağı için, bağımsız yenileme metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	69

Şekil 5.3.	5x5 nöron ağında, 5 örnekleli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi .....	71
Şekil 5.4.	5x5 nöron ağında, 20 örnekleli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi .....	72
Şekil 5.5.	5x5 nöron ağında, 32 örnekleli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi .....	72
Şekil 5.6.	5x5 İleri yönlü yapay sinir ağı için, Watt-Strogatz metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	74
Şekil 5.7.	5x5 nöron ağının, 5 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı .....	75
Şekil 5.8.	5x5 nöron ağının, 20 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı .....	75
Şekil 5.9.	5x5 nöron ağının, 40 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı .....	76
Şekil 5.10.	5x5 nöron ağının, 70 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı .....	76
Şekil 5.11.	5x5 nöron ağının, 100 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı .....	77
Şekil 5.12.	5 örnekleli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi .....	78
Şekil 5.13.	20 örnekleli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi .....	78
Şekil 5.14.	40 örnekleli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi .....	79
Şekil 5.15.	70 örnekleli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi .....	79
Şekil 5.16.	Şekil 5.16. 100 örnekleli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi .....	80

Şekil 5.17.	5x5 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	81
Şekil 5.18.	5x10 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	81
Şekil 5.19.	5x20 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	82
Şekil 5.20.	10x5 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	82
Şekil 5.21.	10x10 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	83
Şekil 5.22.	10x20 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	83
Şekil 5.23.	5x5 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	85
Şekil 5.24.	5x10 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	85
Şekil 5.25.	5x20 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	86
Şekil 5.26.	10x5 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	87
Şekil 5.27.	10x10 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	87
Şekil 5.28.	10x20 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi .....	88

Şekil 5.29.	(42-10-1) topolojili, Düzenli (a) ve Watts-Strogatz Küçük Dünya (b) ağı yapılarına ait topolojik gösterimi .....	89
Şekil 5.30.	42-20-1 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	90
Şekil 5.31.	3 Katmanlı YSA mimarisinde YBS ile eğitim hatasının (MSE) değişimi. (a) 42-10-1, (b) 42-20-1, (c) 42-42-1 ve (d) 42-63-1 .....	92
Şekil 5.33.	Havalı güneş kolektörlerinin termal performansının tahmini için kullanılan düzenli ağ topolojisi .....	94
Şekil 5.34.	8-12-1 İleri yönlü yapay sinir ağı için, Watts-Strogatz metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	95
Şekil 5.35.	8-12-1 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.....	96
Şekil 5.36.	Düzenli ve Küçük Dünya ağlarının termal performanslarının deneysel bulgularla karşılaştırılması.....	98
Şekil 5.37.	Yönlendirilebilir levha üretiminde eğilme direnci parametrelerinin tahmini için önerilen düzenli ağ topolojisi (3-8-8-2).....	100
Şekil 5.38.	3-8-8-2 İleri yönlü yapay sinir ağı için, Watts-Strogatz metodu kullanılarak elde edilen YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	101
Şekil 5.39.	3-8-82 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.....	102
Şekil 5.40.	Düzenli ve Küçük Dünya ağlarının eğilme direnci kestirim performansları (a) MOR (Paralel Eğilme Direnci), (b) MOR (Dikey Eğilme Direnci) .....	104
Şekil 5.41.	Karar destek sistemi düzenli ağ YSA topolojisi .....	107

Şekil 5.42.	17-10-10-1 İleri yönlü yapay sinir ağı için, YBS ile $D_{Global}$ ve $D_{Lokal}$ katsayılarının değişimi .....	108
Şekil 5.43.	17-10-10-1 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.....	109
Şekil 5.44.	Tüm veri seti için düzenli ve Küçük Dünya ağlarının korelasyon analizi (a) Düzenli ağ, (b) Küçük Dünya ağları .....	111

## TABLolar LİSTESİ

Tablo 2.1.	Aktivasyon fonksiyonları.....	14
Tablo 5.1.	İleri yönlü yapay sinir ağlarında farklı katman ve nöron sayıları için Küçük Dünya ağı YBS değer aralığı.....	70
Tablo 5.2.	Havalı güneş kolektörlerinin termal performansları için YBS ile test parametrelerinin değişimi.....	97
Tablo 5.3.	Bağımsız iki grup arası t-testi analiz sonuçları .....	99
Tablo 5.4.	Eğilme direnci tahmininde, YBS ile test parametrelerinin değişimi.....	103
Tablo 5.5.	Düzenli ve Küçük Dünya ağı içint-testi sonuçları. (a) MOR (Paralel Eğilme Direnci), (b) MOR (Dikey Eğilme Direnci) .....	105
Tablo 5.6.	Karar destek sisteminin kan testi girdi parametreleri.....	106
Tablo 5.7.	Diyabet hastalığı teşhisi için YBS tabanlı hata analiz sonuçları.....	110



## ÖZET

Anahtar kelimeler: İleri yönlü yapay sinir ağları, Küçük Dünya Ağları, Watts-Strogatz Küçük Dünya Ağı, Newman-Watts Küçük Dünya Ağı, Yapay Öğrenme.

Beyinde öğrenme süreci, biyolojik sinir ağlarının bilgiyi depo etmesi ve bu bilginin ağı oluşturan nöronlar arası iletilmesi aşamalarından meydana gelmektedir. Buradan hareketle biyolojik sinir ağlarının davranışını modelleyebilmek için çeşitli matematiksel yapay sinir ağı topolojileri ortaya konulmuştur. Literatürde en yaygın kullanılan ağ topolojisi farklı katmanlardan meydana gelen ileri yönlü yapay sinir ağı topolojisidir. Model bilginin giriş katmanından çıkış katmanına doğru ileri yönlü yapay nöronlar ile aktarılması ve nöronlar arası sinaptik bağlantı ağırlıklarının değiştirilmesi ilkesine göre çalışmaktadır.

Bu çalışmada, Küçük Dünya ağları modelinin ileri yönlü yapay sinir ağlarında uygulanması ve öğrenme performansının araştırılması hedeflenmiştir. Bu bağlamda, Watts-Strogatz, Newman-Watts ve Simard tarafından ortaya konulan bağlantı yenileme yöntemleri ile yeni ileri yönlü ağ topolojileri elde edilmiştir. Bu topolojilerin öğrenme süreci ve modelleme performanslarının testi için farklı alanlardan karmaşık problemler kullanılmıştır. Bu ağlardan elde edilen sonuçlar geleneksel ileri yönlü yapay sinir ağı sonuçları ile karşılaştırılmıştır. Küçük Dünya ağının oluşturulmasında, Global ve Lokal bağlantı uzunluk katsayıları kullanılmış ve Küçük Dünya ağı üretebilmek için gerekli yeni bağlantı sayısı aralığı tespit edilmiştir. Elde edilen bu bağlantı aralığı, denemelerle elde edilen ağın başarılı deneme sayısı dağılımı ile tutarlılık göstermektedir.

Watts-Strogatz ve Simard Küçük Dünya ağı modellerinin küçük ölçekli veri seti için daha iyi öğrenme performansı gösterdiği görülmüştür. Ancak veri seti büyüdükçe bu performansın düştüğü gözlemlenmiştir. Ayrıca, bu modellerin yeni bağlantı sayısı aralığının geniş ölçekli olduğu tespit edilmiştir. Newman-Watts Küçük Dünya ağında ise yeni bağlantı sayısı aralığının daha küçük ölçekli olduğu ve bu ağların öğrenme performansının veri seti büyüklüğünden bağımsız olduğu belirlenmiştir. Yapılan çalışmada literatürde ilk defa, Watts-Strogatz Küçük Dünya ağı modeli ile geleneksel ileri yönlü yapay sinir ağı modeli bağımsız örnekli t-testi kullanılarak istatistiksel olarak karşılaştırılmış ve Watts-Strogatz Küçük Dünya ağı modelinin geleneksel ileri yönlü yapay sinir ağı modelinden istatistikî olarak daha anlamlı ( $p < 0.01$ ) bir model olduğu ortaya konulmuştur.

Elde edilen sonuçlar ışığında yapay zekâ araştırmalarında ileri yönlü yapay sinir ağı modelinin geleneksel topolojisi yerine Küçük Dünya ağ topolojisinin kullanılabileceği gösterilmiştir.

# **SMALL WORLD NETWORKS APPROACH AND APPLICATIONS IN THE FEED FORWARD ARTIFICIAL NEURAL NETWORKS**

## **SUMMARY**

Key Words: Feedforward neural network, Small-World networks, Watts-Strogatz Small-World network, Newman-Watts Small-World network, Artificial Neuronal Learning

Learning process in the brain occurs two stages which are the storing of information in the biological neural networks and the transmission of the information between neurons consisting of network. Starting from this point, for modeling of behaviors of biological neural networks various mathematical neural network topologies have been proposed. In the literature, the most commonly used network topology is the feed forward artificial neural network topology that composed of different layers. The model has worked according to procedure that is the transmitting of information from input layer to output layer as feed forward manner and adjusting of the synaptic weights between neurons.

In this study, investigation of implementation of small-world network model in the feed forward artificial neural networks and learning performances have been aimed. In this context, the new feed forward network topologies are obtained with the rewiring methods proposed by Watts-Strogatz, Newman-Watts and Simard. For testing the learning process and the modeling performances of these topologies, complex problems from different field have been used. Obtained results from these networks have been compared with conventional feed forward artificial neural networks results. In generating of small-world networks the global and the local connectivity length parameters have been used and the new rewiring range required to obtain small-world networks are determined. The obtained rewiring range has showed consistency with the successful test distribution.

It was observed that Watts-Strogatz and Simard small-world network models exhibit better performance for small dataset. But it is seen that if the dataset grows this performance decreases. Besides, it is identified that the rewiring range of these models is large scale. In Newman-Watts small-world networks, it is determined that this rewiring range is smaller and learning performances of these networks are independent of the dataset. In the presented study, with the first time in the literature, Watts-Strogatz small-world model is statistically compared with the conventional feed forward artificial neural networks, and it is presented that Watts-Strogatz small-world model is statistically more significant ( $p < 0.01$ ) model than the conventional feed forward artificial neural network model.

In the light of obtained results, it was shown that the small-world network topology can be used instead of the conventional topology of feed forward artificial neural network in the researches of artificial intelligent.

## BÖLÜM 1. GİRİŞ

Yapay sinir ağı (YSA), beynin bilgiyi depo etme ve işleme yeteneklerini matematiksel olarak ortaya koyan bir sistemdir. Özellikle İnsanların bir problem karşısında deneyimlerinden faydalanarak sorun çözme becerisini modellemektedir. Bu nedenle insanlar gibi, YSA' da çok iyi bir genelleme yapabilme yeteneğine sahiptir. Bu kavramlar ışığında YSA günlük hayatımızda sağlık, finans, endüstriyel üretim, istatistik ve enerji sistemleri gibi pek çok alanda problem çözümünde kullanılmaktadır. Bu alanlarda genellikle istatistiksel kestirim, kontrol teorisi ve sınıflandırma amacıyla kullanılmaktadır [1].

YSA' biyolojik sinir hücresinin matematiksel modeli olan sinir hücrelerinin birbirleri ile bağlanmaları ile oluşan bir ağ sistemidir [2]. İnsanların doğumlarından itibaren öğrenme, sinir hücreleri arasındaki sinaptik bağlantıların ayarlanması ile gerçekleşmektedir. İnsan deneyimleri arttıkça sinir hücreleri arasında sinaptik bağlantılar yenilenmekte veya yeni bağlantılar eklenmektedir. Böylece insan deneyim ve gözlemleriyle öğrenme sürecini biyolojik olarak devam ettirmektedir. YSA ise, eğitim amaçlı kendine sunulan girdilere karşılık bir çıkış üretmektedir. Beklenen çıkış elde edilene kadar ağırlıklar sürekli değiştirilmektedir. Sonuç itibarıyla beklenen çıkış elde edildiğinde eğitim süreci tamamlanmaktadır. Yapay her bir sinir hücresi, transfer fonksiyonu denilen bir eşitlikten ibaret olan işlem elemanıdır. İşlem elemanı diğer sinir hücrelerinden gelen sinyalleri alıp birleştirip dönüştürerek bir sayısal sonuç üretmektedir. Böylelikle birbirine bağlı sinir hücreleri arasında bilgi depo edilmekte ve aktarılmaktadır.

Bilim adamları YSA' nın öğrenme becerisini fark ettikten sonra, YSA' ların öğrenme performanslarını artırmak amacıyla çeşitli çalışmalar yapmışlardır [3]. Bu çalışmaların odak noktası farklı topolojik yapılara sahip olan ağ modelleridir. Çalışmalar sonucunda İleri Yönlü Yapay sinir ağı (İYYSA), Geri Beslemeli Ağlar,

Elman Ağları, Hopfield Ağları, Özdüzenleyici Haritalar (SOM), Uyarlamalı Rezonans Kuramı (ART) gibi pek çok ağ modelleri ortaya konmuştur. Topolojik farklılıklar yanında ortaya atılan modellere ek olarak öğrenme algoritmaları da, ortaya konmuştur (Hebb, Hopfield, Delta, Kohonen).

Yapay zekâ çalışmalarının paralelinde, gelişen teknolojiyle birlikte, ağ kavramı hayatımızın önemli bir parçası olmuştur. Gerçek yaşam ağlarını tanımlanmasında karmaşık ağ yapıları geleceğe ışık tutmuştur. Özellikle, 1976 yılında Stanley Miligram tarafından yapılan deneyin önemi büyüktür. Deney, insanların tanımadıkları insanlarla arasında bilgi iletiminde yakın ilişkileri veya ortak yönleri üzerinden iletişim kurduklarını ve birbirini hiç tanımayan insanların ortalama 6 insan aracılığıyla iletişim kurduklarını ve dünyanın küçük olduğunu vurgulamıştır [4]. Watts ve Strogatz yaptıkları araştırmada dünyamızda var olan pek çok gerçek ağın ne düzenli, ne rastsal ağ yapısına uyduğunu ve her ikisinin de özelliklerini barındıran yeni bir ağ yapısına uygun olduğunu ortaya koymuşlardır. Bu ağ yapısında, Stanley Miligram deneyindeki gibi, nöronların bilgiyi uzak düğümlere akrabalık ilişkileri üzerinden kısa bağlantılar oluşturarak taşıdıkları gözlemlenmiştir [4]. Önerilen ağ sisteminde, her bir düğümün komşularıyla arasındaki ilişki aynı ya da belirli bir kurama bağlı ise bu ağ sistemi düzenli (regular lattice) kabul edilmektedir [5]. Ancak ağ içerisindeki her bir düğümün komşularıyla ilişkileri düzensiz ve rastgele ise bu tür ağlar rastsal ağlar olarak kabul edilmektedir [6]. Ancak gerçek dünya ağları, sezgisel olarak çok karmaşık olarak tanımlanmışlardır. Bu nedenle bu tür sistemler, rastsal ağ modelleri olarak kuruldu. Watts ve Strogatz (1998), bu ağ yapılarının karakteristiğini ifade edebilmek amacıyla 2 parametre ortaya koymuştur. Bunlar karakteristik yol uzunluğu ve kümelenme katsayısıdır [7]. Watts ve Strogatz (1998), düzenli ağların yüksek karakteristik yol uzunluğuna ve yüksek kümelenme katsayısına sahip olduğunu ve rastsal ağların ise küçük karakteristik yol uzunluğuna ve küçük kümelenme katsayısına sahip olduğunu vurgulamışlardır [7]. Bu bağlamda elde edilen yeni ağ yapısına Küçük Dünya Ağları denilmiştir.

Küçük Dünya Ağlarında, karakteristik yol uzunluğu küçük (rastsal ağ) ancak kümelenme katsayısı yüksek (düzenli ağ) olarak bulunmuştur [7-8]. Günümüzde Küçük Dünya Ağları, biyolojik sinir ağlarının karmaşık aktivite paternlerini nasıl

ürettiklerini anlamak için yaygın bir şekilde kullanılmaktadır [9-15]. Devam eden karmaşık ağ çalışmalarında, çok büyük ölçekli karmaşık ağların ölçeksiz olduğu ve ağı oluşturan düğümlerin bağlanabilirlik dağılımının güç yasası dağılımına uygun olduğu belirtilmiştir. [16]. Belirli gerçek dünya ağlarının ölçeksiz büyümesinden hareketle bu tür ağlara Ölçeksiz ağlar denilmiştir [17].

Yapay ve gerçek ağ kavramlarının tamamında, nöronlar arası bilginin iletimi ve saklanması ağ performansı olarak kabul edilmektedir. Bu bağlamda, topoloji biyolojik nöron ağlarında olduğu gibi yapay nöron ağlarında da önemli bir yere sahiptir. YSA'da yaygın kullanılan topoloji ileri yönlü yapay sinir ağlarıdır. Bu topoloji, yapay sinirlerin ileri yönlü olarak birbirlerine bağlandığı çok katmanlı yapıya sahiptir. İYYSA' da performansı geliştirmek amacıyla farklı öğrenme algoritmaları kullanılmıştır. Bu algoritmaların birbirlerine üstünlükleri, avantaj ve dezavantajları araştırılma konusu olmuştur [3]. Diğer taraftan, topolojik yapıda sinirler arası bağlantıların ve bağlantı ağırlıklarının değişiminin performansı nasıl etkilediği konusu sürekli araştırılmıştır.

Bu çalışmada amaç; gerçek yaşam problemlerinin çözümünde, İYYSA' da geleneksel topoloji yerine Küçük Dünya ağı yaklaşımının kullanılmasının ağın öğrenme performansına etkisinin araştırılmasıdır. Küçük dünya ağlarının temelinde yatan etken durum düğümler arasına kısa bağlantılar atılarak, düğümler arasında ki akrabalık ilişkilerinin kuvvetlendirilmesidir. Böylece bilginin iletimi daha hızlı ve doğru olacaktır. Bunu gerçekleştirmek için, literatürde yaygın kullanılan Watt-Strogatz, Newman-Watts ve Rastsal Küçük Dünya ağı modellerinin yeniden bağlantı oluşturma prosedürleri ileri yönlü yapay sinir ağı topolojine uygulanmıştır. İşlem sonucunda elde edilen ağ topolojilerinin karakteristik yol uzunlukları ve kümelenme katsayıları hesaplanarak küçük dünya ağları oluşturulmaya çalışılmıştır. Elde edilen ağlardan hareketle Küçük Dünya ağı özelliği gösteren ağ topolojisi aralığı tespit edilmeye çalışılmıştır. Düzenli ağ olarak isimlendirilen geleneksel ileri yönlü yapay sinir ağı topolojisi ile hesaplanan aralıktaki Küçük Dünya ağı arasında öğrenme performansı karşılaştırılmalı olarak ele alınmıştır.

Özetle tez çalışmasında:

1. Küçük Dünya ağları yaklaşımının ileri yönlü yapay sinir ağlarında kullanılabilirliği gösterilmiştir.
2. Watts-Strogatz Küçük Dünya ağı metodu ile ileri yönlü yapay sinir ağları elde edilmiştir.
3. Newman-Watts Küçük Dünya ağı metodu ile ileri yönlü yapay sinir ağları elde edilmiştir.
4. Farklı gerçek yaşam problemleri kullanılarak Küçük Dünya ağı öğrenme performansı araştırılmıştır.
5. Küçük Dünya Ağları yaklaşımının ileri yönlü yapay sinir ağlarına uygulanması için C# dili kullanılarak bir arayüz platformu oluşturulmuştur.

Küçük Dünya ağları ve yapay sinir ağları üzerine çalışmaları özetleyecek olursak;

Stanley Miligram tarafından 1967 yılında Birleşik devletlerde gerçekleştirilen deneyde, 300'e yakın insandan tanımadığı ve kendilerine çok uzak bir adreste ikamet eden bir insana mektup göndermesi istenmiştir. İnsanlar hedef adresteki insanı tanıyabilecek tahmini en yakın akraba ve arkadaşlarını seçerek mektubu hedef adrese en kısa zamanda ulaştırmaya çalışmıştır. Mektupların çoğu yerine ulaşmasa da, ulaşan mektuplardan hareketle mesafeler uzak bile olsa birbirini tanımayan iki insanın ortalama 6 insan üzerinden iletişim kurabileceği vurgulanmış ve aslında dünyanın sanıldığından büyük olmadığı sonucuna varılmıştır [4].

1998 yılında D. J. Watts ve S. Strogatz tarafından gerçekleştirilen çalışmada, insan eliyle oluşturulan doğal ağların özellikleri incelendiğinde, bazı ağ yapılarının ne bütünüyle düzenli ağ nede tamamen rastsal ağ olduğunu, ancak yeni ağ yapısının düzenli ve rastsal ağ yapısından özellikler barındırdığını matematiksel parametreler kullanarak ortaya koymuşlardır. Böylece yapısal özelliklerine göre bu yeni ağ yapısına Küçük Dünya ağı denilmiştir [7].

D. J. Watts, 1999 ve 2003 yıllarında, Küçük Dünya ağlarının özelliklerini, yüksek yerel kümeleme ve kısa global ayrıma sahip çizgeler kullanılarak incelemiş ve çizge

teorisinde günümüzde kullanılan Küçük Dünya ağı çizge modelini matematiksel olarak ortaya koymuştur [18-19].

M. E. J. Newman ve D. J. Watts 1999 yılında, büyüyen çapta sosyal ağlarda, düzenli ağ ile rastsal ağ arasında küçük dünya ağı davranışının varlığını araştırmış ve boyutu değişen ağ yapıları için yeni bir küçük dünya ağı modeli ortaya koymuştur [8].

2000 yılında M. Marcihori ve V. Latora, Ağ kavramında aslında önemli olan etken özelliğın, düğümler arası bağlantı uzunluğunu vurgulamış ve ilk defa Küçük dünya ağlarını tanımlayan Kümelenme ve Karakteristik yol uzunluğuna katsayılarının yerine Bağlantı uzunluk katsayılarının kullanılabilineceğini göstermiştir. Lokal ve global bağlantı uzunluk katsayısı küçük olan ağ çizgelerinin Küçük dünya ağı olduğunu söylemiştir [20].

Yine 2001 ve 2003 yıllarında M. Marcihori ve V. Latora, ağırlıklı ve ağırlıksız ağlarda Küçük dünya ağlarında kullanılan matematiksel modelin kullanılmayacağını vurgulayarak, bilginin ağ ortamında nasıl iletildiğinin ölçümü olan yerel ve global verim modelini önermişlerdir. Bu modelin Küçük dünya ağlarının ifadesinde daha net bilgiler içerdiğini göstermişlerdir [21-23].

2005 yılında ilk olarak D. Simard ve arkadaşları, geri yayılım öğrenme algoritması kullanan çok katmanlı ileri yapay sinir ağlarında, rastgele yeniden bağlantı oluşturarak Küçük Dünya ağının varlığını göstermiş ve Küçük Dünya ağının öğrenme performansı ve hızının düzenli ve rastsal ağlara göre daha iyi olduğunu göstermiştir [24].

S. Yang ve arkadaşları, 2006 yılında, geleneksel çok katmanlı yapay sinir ağlarında, gizli katman sayısı ve gizli katman nöron sayısının önemli bir sınırlandırıcı etken olduğunu vurgulayıp, küçük dünya ağı topolojisinin gizli katman kısıtlamalarını ortadan kaldırarak, geleneksel kurguya göre performansının daha iyi olduğunu karşılaştırmalı olarak göstermişlerdir [25].

Aynı zamanda 2006 yılında, D. S. Bassett ve E. Bullmore, görüntüleme teknolojileri (fMRI ve electroencephalography) kullanılarak, Makak maymunu ve Kediler üzerine yapılan canlı içi deneylerde beynin anatomik ve fonksiyonellik bakımından, küçük dünya ağı özelliği gösterdiğini göstermişler ve bu teoriyi beynin çizge haritasını çıkararak ve korelasyon matrisleri kullanımıyla ispatlamışlardır [26]. Farklı bir çalışmada, S. Yang ve ark., ilk çalışmalarından farklı olarak, geleneksel özdüzenleyici haritalar topolojisinde, nöron bağlantılarını Küçük Dünya ağı modeli kullanarak değiştirdiklerinde elde edilen yeni topolojinin, ağırlık dağılımı uygunluğu ve zaman karmaşıklığı bakımından daha iyi performans sergilediğini ve bu modelin 2 kısıtlayıcı faktöre sahip olduğunu göstermişlerdir [27].

2007 yılında, K. Fortney ve ark., beyin gibi karmaşık bir ağın Küçük Dünya ağı olabileceğini söyleyerek, ileri yönlü düzenli, rastsal ve Küçük Dünya yapay sinir ağı olmak üzere 3 ağ yapısı oluşturarak, bu ağları ortak bir veri seti ile eğitmişlerdir. Sonrasında, bir kaza senaryosu kurgulanıp 3 ağ yapısından da aynı sayıda rastgele bağlantılar koparılmıştır. Elde edilen bu ağ yapılarında öğrenme testi uygulandığında Küçük Dünya ağlarının kaza öncesi bilgileri diğer ağ topolojilerine göre daha iyi hatırladığı gösterilmiştir [28].

2008 yılında Li Shou-wei, YSA, Küçük Dünya ağları ve beyin ağlarının, hücre, bağlantı ve davranış bakımından özelliklerini araştırarak, bu üç ağın karşılaştırmalı olarak karmaşıklıklarını ve farklılıklarını ortaya koymuştur [29]. Aynı yılda T. Pallaver ve ark., beyin nöron organizasyonunun kohonen ağına benzerliğini ortaya koymuş ve bu ağın nöronal bağlantılarının Küçük Dünya ağı metoduna göre düzenlendiğinde öğrenme ve bilgi iletiminde ağ performansının arttığı gösterilmiştir [30].

2009 yılında, Fei Jiang ve ark., özdüzenleyici haritalarda network topolojisinin etkisini araştırmış ve nöron ağının komşusal ilişkilerinin karmaşık ağlara benzer değiştirildiğinde, oluşan ağın Küçük Dünya ağı özelliği gösterdiği ifade edip, öğrenme süreci performansının diğer ağlara oranla, aşağı yukarı %10 daha iyi performans sergilediğini ortaya koymuşlardır [31].



2011 yılına gelindiğinde Li Xiaohu ve ark. çok katmanlı ileri yönlü YSA kullanılarak, yeni bir küçük dünya ağı modeli üzerine çalışma yapmışlar ve fonksiyon yaklaşımı bakımından geleneksel YSA modeline göre Küçük Dünya ağlarının performansını karşılaştırmalı olarak ortaya koymuşlardır [32].

Tez çalışmasının diğer bölümleri şu şekilde düzenlenmiştir:

İkinci bölümde, YSA ve kullanılan YSA topolojisi hakkında temel bilgiler verilmiştir.

Üçüncü Bölümde, Karmaşık ağlar ve çizge teorisi konusu sunulmuştur.

Küçük Dünya Ağları ve özellikleri hakkında temel bilgiler verilmiştir.

Dördüncü bölümde, hazırlanan Küçük Dünya ağları arayüz programı tanıtılmıştır.

Beşinci bölümde, İleri yönlü yapay sinir ağlarında Küçük Dünya ağları yaklaşımının uygulanması ve farklı problemler üzerine performansının araştırıldığı uygulamalar ve sonuçları verilmiştir.

Altıncı bölümde ise tez çalışmasına ait sonuçlar verilmiştir.

## **BÖLÜM 2. YAPAY SINİR AĞLARI**

İnsan beyni karmaşık bir ağ yapısına sahiptir ve bu sebeple beyin karmaşıklığını modellemek bilim için önemli bir amaç olmuştur. İlk defa nöronun matematiksel olarak modellenmesinden bu güne YSA önemli bir gelişim kat etmiş ve bugün kimya, fizik, malzeme, elektrik, elektronik, bilgisayar, robotik gibi pek çok alanda önemle araştırılan ve kullanılan bir araç olmuştur. Model veriler kullanarak öğrenme sürecini gerçekleştirmektedir. Bugün gelinen noktada YSA farklı topolojik ya da sezgisel modelleri ile bilimin pek çok problemini çözer hale gelmiştir.

### **2.1. YSA' nın Tarihsel Gelişimi**

1943 yılında, McCulloch ve Pitts biyolojik sinir hücresinin basit matematiksel modelini oluşturmuştur [33]. Bu modelde sinir hücresi kendine gelen bilgiyi belirli bir eşik değerine göre anahtarlama yaparak ya iletmekte yada sönmlemektedir. 1949 'da Hebb, aynı anda ateşleyen iki sinir hücresinin arasındaki bağlantı ağırlığının artacağını vurgulayarak yeni bir öğrenme kuralı ortaya koymuş ve bu kural günümüzde kullanılan birçok öğrenme kuramının temelini oluşturmuştur [34]. Rosenblatt, 1959'a gelindiğinde, McCulloch ve Pitts'in geliştirdiği matematiksel modeli geliştirerek sinir hücresine perseptron adını vermiş ve bu model günümüzde kullanılmakta olan yapay sinir ağlarının temelini oluşturmuştur [35-36]. Perseptron modeli nörona gelen giriş sinyallerini girişteki ağırlıklarla çarpıp ve elde edilen sonuca göre -1 ile 1 arasında bir çıkış üretmekten bir modeldi. Yine aynı yıllarda Widrow ve Hoff ADALINE (Adaptive Linear Model) ve MADALINE (Multiple ADALINE) isimli çoklu adaptif doğrusal bir yapay nöron modeli geliştirmiştir [37]. Bu model kullanılarak ses ve karakter tanıma gibi hafıza davranışını modelleyen sistemler geliştirilmiştir [37-39]. Perseptron modelinden farklı olarak delta öğrenme kuralını kullanılmakta olan model, beklenen ve hesaplanan çıkış değerlerini birbirine yaklaştırmaya çalışılmaktadır [37]. Miksy ve Papert (1969) tek katmanlı perseptron

ağının XOR işlemini çözemediğini göstermiş ve YSA'nın ilgi çeken bir konu olmadığını vurgulamışlardır. Bu vurgudan sonra YSA çalışmaları durma noktasına gelmiştir [40-41].

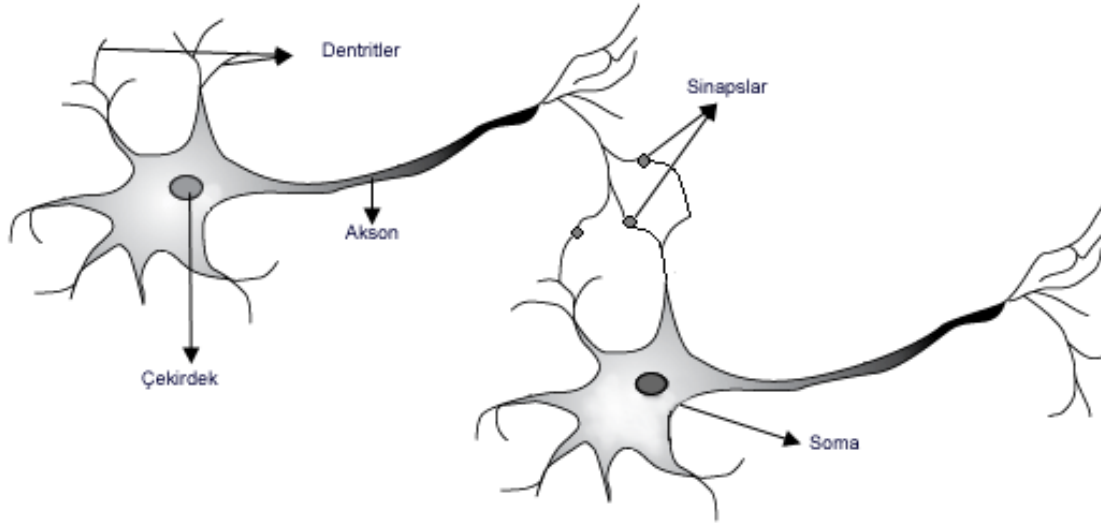
1982 yılında J.J. Hopfield, tek katmanlı geri beslemeli bir ağ modeli ortaya koymuştur. Bu modelde nöronlar binary eşik değerlerine sahiptir [42]. Ağ içerisinde nöronal bağlantıları bir enerji fonksiyonu oluşturmaktadır. Hopfield ağı olarak isimlendirilen model optimizasyon problemlerinin çözümde kullanılmıştır. Bu dönemde Hopfield ağlarıyla birlikte YSA çalışmaları hız kazanmış [42] ve 1986'da Grossberg, ART (Adaptive Resonance Theory) Uyarlanabilir Rezonans Teorisi ismini verdiği yeni bir YSA modeli sunmuştur [43]. ART insan beyninin nasıl hızlı öğrenebileceği ve öğrenilen olayları nasıl kararlı bir şekilde hatırlayabileceğiyle ilgili bir kuram olarak tanımlanmaktadır [44]. Aynı yıllarda, Kohonen, SOM (Self-Organizing Maps) özdüzenleyici haritalar isimli bir YSA modeli geliştirmiştir [45]. Model çok boyutlu girdileri kullanarak, yarışmacı öğrenme kuralıyla çıkış elde etmeye çalışmaktadır [45]. İlerleyen süreçte Rumelhart ve arkadaşları "Paralel Dağılımlı İşleme (Parallel Distributed Processing) isimli kitaplarında, ileri yönlü YSA'da yeni öğrenme kuralı olan hatanın geriye yayılımı algoritmasını (backpropagation algorithm) önermiştir [46]. Bu kural ile ileri yönlü YSA'daki öğrenme sorunlarının optimum çözülebildiği gösterilmiştir [46].

Günümüzde, Geri yayılım öğrenme algoritması pek çok YSA uygulamasında temel öğrenme algoritması olarak kullanılmaktadır. Bu nedenle hazırlanan tezde de öğrenme algoritması olarak kullanılmıştır.

## **2.2. Biyolojik Sinir Ağı ve Nöron**

Yapılan araştırmalarda insan sinir sisteminin yaklaşık 10 milyar sinir hücresinden oluştuğu ifade edilmiştir. Bir sinir hücresi ise, soma, akson ve dentrit denilen yapılardan oluşmaktadır. Soma, hücre çekirdeği, çekirdekçik ve stoplazmik sıvıdan oluşmaktadır. Dentritler lifli yapıları sayesinde diğer hücrelerinin dentritlerine bağlanarak ağ yapısını oluşturmakta ve diğer nöronlardan gelen elektriksel bilgiyi somaya iletmektedir. Akson ise somadan çıkan tek lifli yapıdır. Somadan gelen

elektriksel bilgiyi akson uçları yardımıyla diğer hücelere taşımakla görevlidir. Biyolojik hücre yapısı Şekil 2.1’ de gösterilmiştir.



Şekil 2.1. Sinir hücresi yapısı[46].

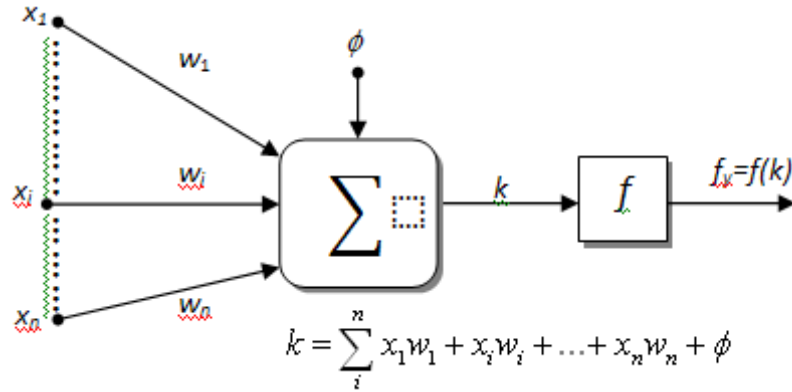
Sinir hücrelerinin birbirleri ile haberleşmesi hücreler arasındaki bağlantı noktalarında bulunan “sinaps” denilen boşluklar üzerinden yapılmaktadır. Dentritlerden alınan bilgi somada toplanıp yorumlanmaktadır. Belirli bir eşik değeri aşıldığında nöron ateşlenmekte ve bilgi aksiyon potansiyeli olarak akson üzerinden sinapslar ile diğer nöronlara aktarılmaktadır. Bu yapıda birbirine bağlanan biyolojik nöronlar karmaşık nöron ağlarını oluşturmaktadır. Sinir sistemi için öğrenme, nöronlar arasında oluşturulan sinaptik bağlantılarla gerçekleşmektedir. İnsanın doğuştan itibaren yaşamsal tecrübeleri ışığında nöronları arasında sinaptik bağlantıları ayarlanır veya yeni bağlantılar oluşturulur. Bu sayede öğrenme süreci gerçekleştirilmektedir.

### 2.3. YSA’da Kullanılan Kavramlar

İnsan beyninin basit matematiksel modeli olan YSA, insan beynine benzer bir davranışla, kendine sunulan örnekler üzerinden yani yaşamsal tecrübeler ile öğrenme sürecini tamamlayıp, yeni örnekler kendisine sunulduğunda genelleme yapabilmektedir. Topolojik olarak YSA çok sayıda birbirine bağlanmış olan birbirleriye paralel çalışabilen işlem elemanlarından(nöron) ve bu elemanlar arasında oluşturulan bağıntılardan oluşmaktadır. Topolojik duruma ek olarak, ağın yapı taşı olan nöronlar, bir öğrenme algoritması ile çalışma şekillerini belirlemektedirler.

### 2.3.1. İşlem birimi(Nöron)

İşlem birimi, dentritlere bezer şekilde kendine gelen bir yada daha fazla girdiyi toplayan, soma gibi, eşik değerine göre bilgi iletimine karar veren ve aksona benzer şekilde çıkış oluşturup diğer işlem birimlerine sunan yapıdır. İşlem birimi yapısı Şekil 2.2' de gösterilmiştir.



Şekil 2.2. Yapay işlem birimi yapısı

İşlem birimi dışarıdan yada başka bir işlem biriminden  $x_1, x_i, \dots, x_n$  şeklinde  $n$  adet giriş alabilmektedir. Bu girdiler hücreye sunulurken bağlantının ağırlıklarına bağlı olarak sırasıyla  $w_1, w_i, \dots, w_n$  gibi sunulmaktadır. Sunulan modelde ağırlık biyolojik sinir sistemindeki sinaptik bağlantılara karşılık gelmektedir. Somada olduğu gibi nöronun ateşlemesi için gerekli olan eşik değeri  $\phi$  ile gösterilmiştir. Toplam fonksiyonu bir işlem birimine giren net girişi hesaplamaktadır. Bu işlem her bir girişe karşılık gelen ağırlıkların çarpımının toplamı olarak tanımlanmaktadır. Son olarak bu toplama  $\phi$  eşik değeri eklenmektedir. Toplam fonksiyonu eşitlik 2.1'deki gibi ifade edilmektedir. :

$$k = \left( \sum_{i=1}^n x_i w_i \right) + \phi \quad (2.1)$$

İşlem birimi çıkışı, biyolojik nörona benzer şekilde, aksiyon potansiyelinin ateşleme oranına benzer bir aktif hale dönüşüm fonksiyonudur [41]. Bir yapay işlem biriminin net çıkışı eşitlik 2.2' de ifade edilmiştir.

$$f_y=f(k) \quad (2.2)$$

### 2.3.2. Aktivasyon Fonksiyonları

YSA ile bir problemin çözümünde, YSA çıkışının istenilen çıkışa yakınsanması için, çıkış katmanı veya ara katmanlarda bir aktivasyon fonksiyonu kullanılmaktadır. Aktivasyon fonksiyonu, bir işlem biriminin girdilere karşılık ürettiği, ölçeklendirilmiş çıkışı ifade etmektedir. Sinir ağı modellerinde, yapılacak işe göre işlem biriminde farklı aktivasyon fonksiyonları kullanılmaktadır. Temel olarak birim basamak, doğrusal, rampa, sigmoid ve hiperbolik tanjant olmak üzere 5 farklı çeşidi yaygın kullanılmaktadır. Bunların dışında yapılacak işe göre farklı aktivasyon fonksiyonları da kullanılmaktadır.

Birim basamak fonksiyonu, girişlerin ağırlıklı toplamları belirli bir eşik değerinin üstündeyse pozitif, eşik değerinin altındaysa negatif sonuçlar üreten aktivasyon fonksiyonudur. Genellikle 2 gruplu bir sınıflandırma işlemi söz konusu ise kullanılmaktadır. eşitlik 2.3' teki gibi tanımlanmaktadır.

Unipolar:

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2.3a)$$

Bipolar:

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (2.3b)$$

Doğrusal aktivasyon fonksiyonu, doğrusal problem çözümünde tercih edilen bir fonksiyondur ve hücre girdisini çıkışa değiştirmeden aktarmaktadır. Şekil 2.4' teki gibi tanımlanmaktadır. Burada n, sabit bir sayıdır.

$$f(x) = nx \quad (2.4)$$

Rampa fonksiyonu ise doymun doğrusal fonksiyon olarak bilinmektedir. Doğrusal fonksiyon çıkışı  $+l$  ile  $-l$  arasında sınırlandırılmak istenilirse kullanılacak aktivasyon fonksiyonudur ve Şekil 2.5' teki gibi tanımlanmaktadır. Bu sınırlandırma noktaları fonksiyonun doyum noktaları olarak ifade edilir.

$$f(x) = \begin{cases} +l, & x \geq l \\ x, & |x| < l \\ -l, & x \leq -l \end{cases} \quad (2.5)$$

Sigmoid fonksiyon, doğrusal olmayan ve sürekli türevi alınabilen bir fonksiyon olmasından dolayı YSA modellerinde en yaygın kullanılan fonksiyondur. 0-1 arasında çıktı üretmektedir. Şekil 2.5' te ifade edilmiştir.

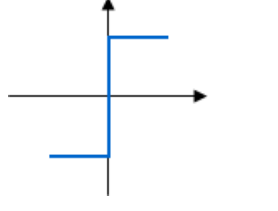
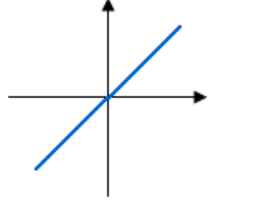
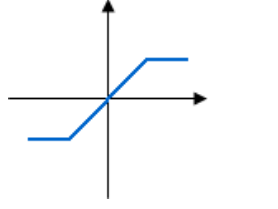
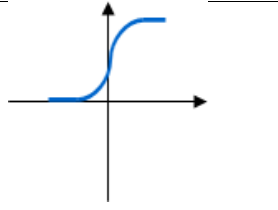
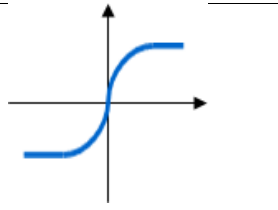
$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

Hiperbolik Tanjant fonksiyonu, sigmoid fonksiyondan farklı olarak negatif değerlerde üretebilen bir fonksiyondur.  $+\infty/-\infty$  uzayında aldığı girdileri  $[-1,+1]$  arasında çıkış değerlerine dönüştürmektedir. Bipolar hiperbolik tanjant fonksiyonu aşağıdaki gibi ifade edilmektedir.

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (2.6)$$

5 farklı aktivasyon fonksiyonuna ait grafiksel gösterim Tablo 1'de gösterilmiştir.

Tablo 2.1. Aktivasyon fonksiyonları

Birim Basamak	
Doğrusal	
Doygun Doğrusal	
Sigmoid	
Hiperbolik Tanjant	

### 2.3.3. Katmanlar ve ağırlıklar

Ağ kavramı YSA yapılarında işlem birimlerinin birbirleri arasında bağlantı kurmasıyla oluşmaktadır. Ancak bazı işlem birimleri problemin çözümüne göre gruplanmaktadır. Örneğin dış dünyadan girdileri algılayıcılar yardımıyla alan işlem birimleri bir paralel grup oluşturmaktadır. Bu grup yapısı giriş katmanı olarak adlandırılmaktadır. Yine dış dünyaya problem çıktısını aktaracak işlem birimlerinin oluşturduğu paralel gruba çıkış katmanı ismi verilmektedir. Bunun yanında, bazı problemlerde ara katman yada gizli katman'da denilen giriş katmanı ile çıkış katmanı arasında ilişkisel işlem performansını artıran bir işlem birimi grubuna ihtiyaç duyulmaktadır. Ara katman sayısı probleme bağlı olarak değişebilmektedir. Yine ara



katmandaki işlem birimi sayısı girdi ve çıktılardan bağımsız olarak değişebilmektedir.

Ağırlık kavramı  $w$ , YSA öğrenim sürecinin temel taşıdır. Ağırlıklar, işlem birimleri arasındaki ilişkinin kuvvetini belirlemekte ve öğrenme sürecinde minimum hatanın elde edilmesinde değişken yapılarıyla etkin rol oynamaktadır. YSA en iyi performansı elde etmek amacıyla bir öğrenme algoritmasına ihtiyaç duymaktadır. Bu öğrenme algoritması bir performans kriterine göre ağırlıkları optimum çıkış elde edilene kadar değiştirmektedir.

## **2.4. Yapay Sinir Ağlarının Sınıflandırılması**

YSA, yapısal olarak işlem birimi, katmanlar ve bağlantılardan oluşmaktadır. İşlem birimleri arasında veya katmanlar arasında bağlantıların değiştirilmesi topolojinin değişimi anlamına gelmektedir. Buradan hareketle YSA'lar yapılarına göre sınıflandırılabilir. Ayrıca YSA'lar yapısal değişikliğin yanında, öğrenme algoritmalarına göre de sınıflandırılmaktadırlar.

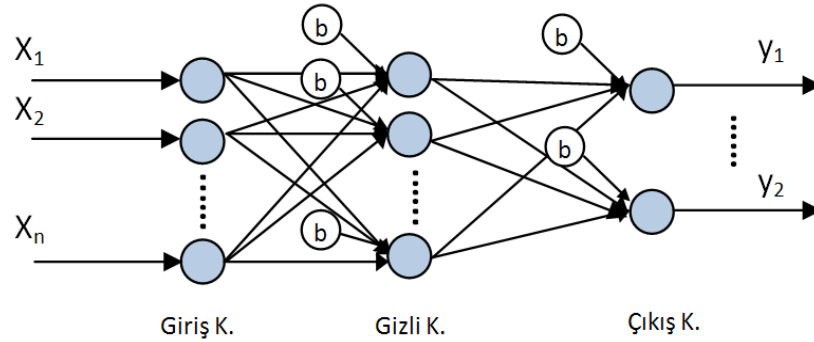
### **2.4.1. Yapısal (Topolojik) sınıflandırılma**

Yapısal sınıflandırma temel olarak İleri yönlü ve geri dönüşümlü YSA yapıları olarak ele alınmaktadır [40].

#### **2.4.1.1. İleri yönlü ağ yapısı**

Katmansal yerleştirilen işlem birimlerinin giriş katmanından başlayarak ara katmana ve ara katmandan çıkış katmanına doğru ileri yönlü olarak bağlandığı ağ yapısı olarak bilinmektedir. Burada bağlantısal bir geri dönüşüm söz konusu değildir. Bilgi giriş katmanına verilir oradan ara katmana iletilir ara katmanda işlenerek çıkış katmanına iletilir. Çıkış katmanında hesaplanan çıkış verisi, dış dünya ya aktarılır. Burada bir katmanın çıkışı diğer katmanın girdisi olmaktadır. İleri yönlü ağlara örnek

olarak, Çok katmanlı algılayıcı model, MADALINE ve LVQ ağları gösterilebilir. İleri yönlü ağ topolojisi örneği Şekil 2.3' te gösterilmiştir.



Şekil 2.3. İleri yönlü ağ topolojisi

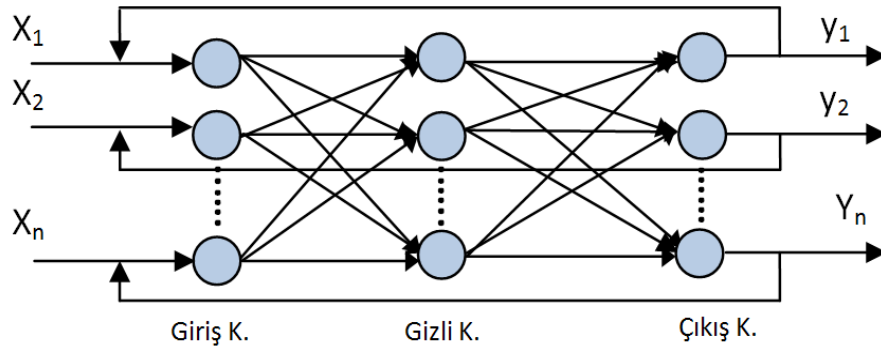
#### 2.4.1.2. Geri dönüşümlü ağ yapısı

Geri dönüşümlü ağlarda, işlem birimlerinin çıkışları belirli durumda geri gönderilerek girdi olarak verilmektedir. Bu geri dönüşüm yapısı, tam geri dönüşümlü veya kısmi geri dönüşümlü olarak 2 türlü gerçekleştirilebilir. Şekil 2.4' te geri dönüşümlü ağ örneği gösterilmiştir.

**Tam Geri Dönüşümlü Ağlar:** İleri yönlü bağlantılarının yanında, rastgele geri dönüşümlerde sahip ağlardır. Ağın bütün bağlantıları eğitilebilmektedir.

**Kısmi Geri Dönüşümlü Ağlar:** Ağın işlem birimlerine ek olarak birde içerik elemanları bulunmaktadır. Burada işlem birimleri ileri yönlü bağlanmıştır ve sadece içerik elemanları üzerinden geri dönüşüm yapılmaktadır. İleri yönlü bağlantılar eğitilebilirken, içerik elemanı bağlantıları eğitilememektedir. İçerik elemanları, ara katman işlem birimlerinin geçiş durumlarını daha sonra kullanılmak üzere hafızada tutmak zorundadır. Bu nedenle geri dönüşüm için kullanılırlar geçmişini hatırladıkları için dinamik bellek özelliğine sahiptirler.

Geri dönüşümlü ağlarda, ağ çıkışı hem önceki durumlara hem de o andaki duruma bağlı olarak hesaplanmaktadır. Geri dönüşüm işlemi ağın genelinde bir zaman gecikmesine sebep olmaktadır. Gerçek yaşamda zaman gecikmesinin önemli olduğu dinamik sistemlerde kullanılmaktadır. Hopfield, SOM, Elman ve Jordan ağları geri dönüşümlü ağlara örnektir [42,48-49].

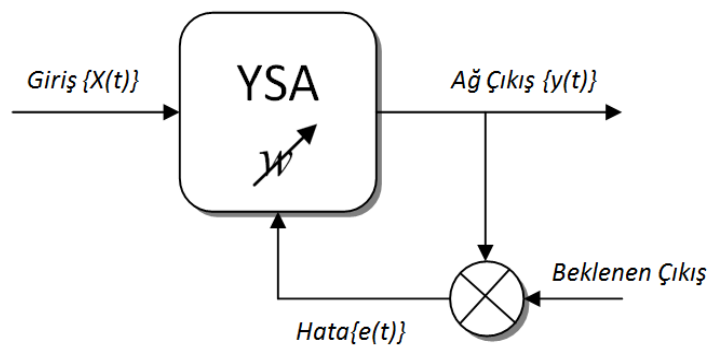


Şekil 2.4. Geri dönüşümlü ağ topolojisi

## 2.4.2. Öğrenme algoritmasına göre sınıflandırma

### 2.4.2.1. Danışmanlı öğrenme

Ağın eğitimi için girdi ve çıkışlar ağı sunulur. Ağ girişleri kullanarak çıkışları hesaplar. Elde edilen çıkış ile beklenen çıkış karşılaştırılarak ağ hatası elde edilir. Bu hata uygun görülen sınıra kadar ağırlıklara yansıtılarak, ağırlıklar güncellenir. Böylece ağın öğrenme süreci danışmanın belirlediği hat kriterine kadar devam eder. Bu öğrenme algoritması YSA uygulamalarında yaygın kullanılan öğrenme metodudur. Widrow ve Hoff [37] tarafından geliştirilen delta ve Rumelhart ve McClelland [46] geliştirilen geri yayılım algoritması danışmanlı öğrenme algoritmalarına en iyi örnektir. Danışmanlı öğrenmenin blok diyagramı Şekil 2.5' te gösterilmiştir.

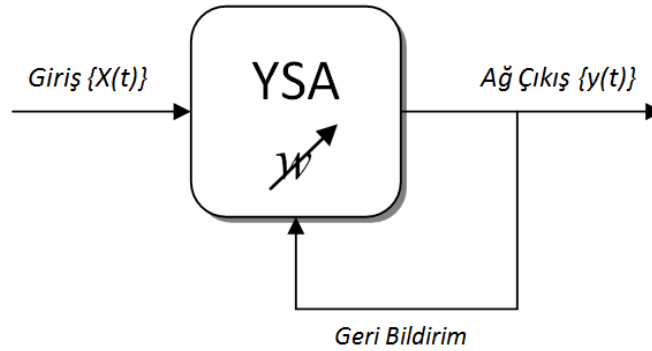


Şekil 2.5. Danışmanlı öğrenme blok diyagramı

### 2.4.2.2. Danışmansız öğrenme

Eğitim için sadece eğitim girdilerinin ağı sunulduğu ve beklenen çıkışların ağı sunulmadığı bir algoritmadır. Ağ, verilen girdilerin parametreleri arasındaki ilişkileri

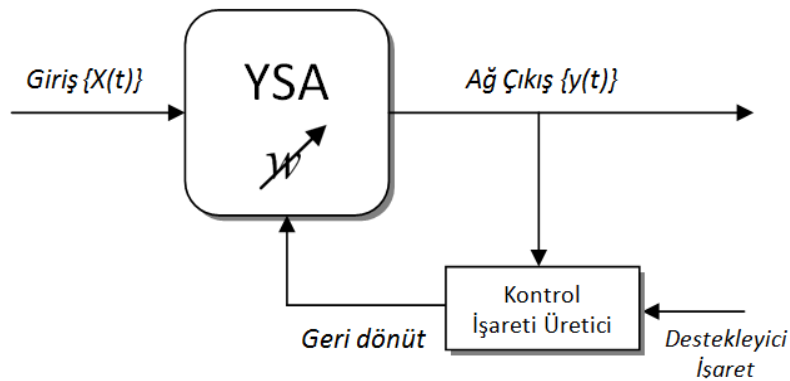
(gruplama) kullanarak öğrenme kurallarını kendisi oluşturmaktadır. Bu algoritma daha çok sınıflandırma problemlerinde kullanılmaktadır. Kohonen [45] tarafından geliştirilen SOM ve Carpenter ve Grossberg [50] tarafından geliştirilen Adaptif Rezonans Teorisi (ART) danışmansız öğrenme algoritmasını kullanmaktadır. Danışmansız öğrenme blok diyagramı Şekil 2.6' te gösterilmiştir.



Şekil 2.6. Danışmansız öğrenme blok diyagramı

### 2.4.2.3. Destekleyici öğrenme

Destekleyici öğrenme danışmanlı öğrenme algoritmasına benzemektedir. Ancak ağ çıkışından alınan bilgi beklenen çıkış ile karşılaştırılmaz, sadece ağ çıkışının doğru yada yanlış olduğunu geri dönüt olarak ağa bildiren bir danışman vardır. Danışmandan gelen bilgiye göre ağ eğitimi gerçekleştirir. Boltzman Makinesi [51], LVQ (*Learning Vector Quantizer*) Ağı[52-53] ve Genetik Algoritmalar destekleyici öğrenmeye örnek olarak verilebilir [54]. Bu özelliklerinden dolayı kontrol uygulamalarında kullanımı uygundur. Destekleyici öğrenme blok diyagramı Şekil 2.7' de gösterilmiştir.



Şekil 2.7. Destekleyici öğrenme blok diyagramı.

Üç temel öğrenme algoritmasının yanında hibrit yöntemlerde YSA modellerinde kullanılmaktadır. Bunlar; Kısmi danışmanlı, Olasılık tabanlı ve Radyal tabanlı öğrenme yöntemleri olarak örneklendirilebilir [3].

## 2.5. Uygulama Alanına Göre Sınıflandırma

### 2.5.1. Çevrim içi (Online) öğrenme

Bu öğrenme yapısında; YSA, gerçek zamanlı çıkış üretirken bir yandan da performansını artırmak için eğitimine devam etmektedir. Kohonen ve ART ağ modelleri bu öğrenme yapısına örnek olarak verilebilir [3,48].

### 2.5.2. Çevrim dışı (Offline) öğrenme

YSA, eğitim süreci ağı sunulan girdilerle tamamlanır ve sonra gerçek zamanlı bir uygulamada kullanılmaktadır. Kullanım sürecinde öğrenme olmamaktadır. Eğer öğrenilmesi gereken yeni bilgiler söz konusu ise ağ kullanımı iptal edilerek, tekrar eğitim süreci gerçekleşir. Delta öğrenme kuralı bu tip öğrenmeye örnek verilebilir [48].

## 2.6. YSA Öğrenme Kuralları

Yapay sinir ağlarında öğrenme kuralları, 1949 yılında Donald Hebb tarafından üretilen kuralın varyasyonları olarak karşımıza çıkmıştır. Yaygın bilinen öğrenme kuralları aşağıda açıklanmıştır.

**Hebb Kuralı:** Donald Hebb tarafından ortaya konan bu kural, birbirine bağlı iki nöron aynı anda ateşliyorsa aralarındaki bağlantının güçlendirilmesi esasına dayanmakta idi, diğer bir deyişle, aynı işarete sahip bağlı nöron çiftleri arasındaki bağlantı ağırlığının artırılması gerekliliğine dayanan öğrenme kuralıdır [56].

**Hopfield Kuralı:** Hebb kuralına benzemekle birlikte, farkı, aynı anda ateşleyen iki nöron çiftinin bağlantı ağırlığının artırılması veya azaltılması öğrenme algoritmasına

göre belirlenmektedir [57]. Öğrenme katsayısı [0-1] aralığında tanımlanan kullanıcı tanımlı sabit bir değerdir. Hopfield ağlarında kullanılmaktadır [58].

**Delta Kuralı:** Bu kural, Hebb kuralından geliştirilmiştir. Ağ çıkışı ile beklenen çıkış arasındaki fark(delta ) hesaplanır. Fark geriye doğru bir önceki katman ağırlıklarına yansıtılır, önceki katman ağırlıkları güncellenir. Böylece çıkış hatası(ortalama karesel hata) azaltılmaya çalışılmaktadır. Geriye yayılma işlemi ilk katmana kadar devam etmektedir. Delta kuralı, en küçük ortalama kareler (LMS) öğrenme kuralı olarak ta adlandırılmaktadır[37].

**Eğim Düşümü Kuralı:** Delta kuralından türetilen bir kural olmakla birlikte ek olarak hatanın geriye yayılımında öğrenme katsayısına ek olarak bir oransal sabitin eklendiği bir kuraldır. Burada, hata maliyet fonksiyonunun en küçük değeri aramaktadır. Böylece hatanın eğimi hesaplanarak aksi yönde küçük adımlar atılmakta ve hata küçültülmeye çalışılmaktadır [3] Böylece yerel küçük noktalara takılma denilen, ezberleme durumunun önüne geçilmeye çalışılmıştır.

**Kohonen Kuralı**[45], Özdüzenleyici haritalar (Self-Organizing Maps-SOM) ve Vektör Kuantalamalı Öğrenme (Learning Vector Quantization-LVQ) ağlarında kullanılan öğrenme kuralıdır. Temel davranışı n boyutlu bir girdi vektörünü bir boyutlu bir çıktı vektörüne dönüştürebilmektir. Öğrenme, kohonen katmanındaki işlem birimlerinin birbirleriyle yarışmasına dayanan kuramdır. Teorem “kazanan her şeyi alır” fikrini benimsemiştir. Girdi vektörü ile her bir nöronun çıktı(referans) vektörü arasındaki Öklid mesafesi ne kadar az ise o nöron kazanmaktadır. Kazanan ve komşu nöronların ağırlıkları değiştirilebilir [59]. Komşuluk sınırları eğitim süresinde değişmektedir. Bu öğrenme algoritması sınıflandırma problemlerinde çok yaygın kullanılmaktadır.

## 2.7. Algılayıcı Ağ Modeli

Çok katmanlı algılayıcı model [46], önerildiğinden bugüne en yaygın ağ modeli olarak kullanılmaktadır [60]. Hazırlanan tezde ÇKA sinir ağı modeli kullanılmıştır.

### 2.7.1. Algılayıcı model

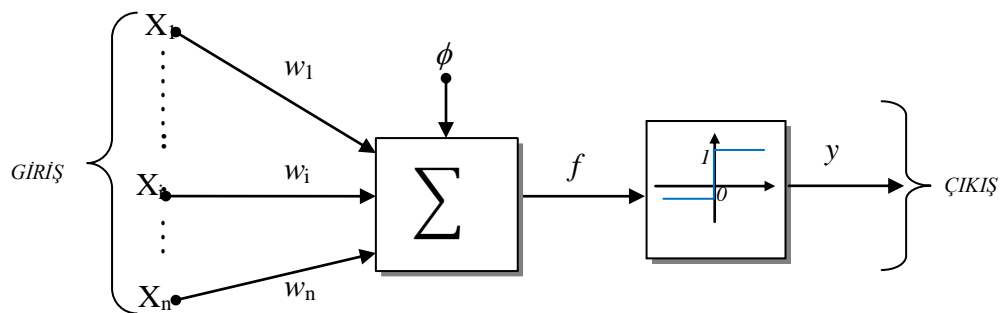
1950 yılında Rosenblatt tarafından Algılayıcı model, doğrusal ayrılabilen örüntüleri sınıflandırma amaçlı önerilmiştir [35]. Model bir eşik fonksiyonu ve ağırlıklı girdilerini içeren tek nöronlu bir yapıya sahiptir. Nöron  $x(x_1, x_2, \dots, x_n)$ , uzayında girdiler almaktadır. Nöron kendine gelen girdileri ve eşik değerini ağırlıklı toplam fonksiyonu kullanarak işlemektedir.  $f$ , ağırlıklı toplam fonksiyonudur. Bir algılayıcı model eşitlik 2.8' de gösterilmiştir.

$$f = \left( \sum_{i=1}^n x_i w_i \right) + \phi \quad (2.7)$$

$f$  fonksiyonu bir aktivasyon fonksiyonu kullanılarak aktif olacaktır. Bu bağlamda  $y$  çıkışı aşağıdaki gibi hesaplanmaktadır.

$$y = \begin{cases} 1, & \text{if } f \geq 0 \\ 0, & \text{if } f < 0 \end{cases} \quad (2.8)$$

Bir algılayıcı yapısal modeli Şekil 2.8' de gösterilmiştir.



Şekil 2.8. Algılayıcı yapısal modeli

Algılayıcı sinir hücresi modeli, sınıflandırma problemlerinde kullanılmaktadır. Algılayıcıda, girdiler ağırlık vektörü ile çarpılarak toplanır. Toplam ifadesi,  $\phi$ , eşik değerinden büyükse 1, değilse 0 çıkışı üretilmektedir. Veri kümesi algılayıcı eğitimi ve testi için rastgele ayrılarak algılayıcıya sunulmaktadır. Eğer işlem sonucu istenilen çıkışa sahip değilse, öğrenme sürecinde ağırlıklar aşağıdaki gibi ayarlanmaktadır.

$$W(i+1) = W(i) + \alpha (bc(i) - y(i)) x(i) \quad (2.9)$$

Burada  $y(i)$ , algılayıcı çıkışıdır. Öğrenme hızı(katsayısı)  $\alpha$ ,  $[0, 1]$  aralığında değişen bir sayıdır.  $i$ , indeksi ise rastgele bir sıradaki algılayıcıya uygulanan eğitim örneğini göstermektedir. Her bir örnek  $x(i)$ , ayrı ayrı sunulmaktadır. Böylece algılayıcı tüm eğitim örneklerinin ağırlıklarını güncelleyerek öğrenmektedir. Her bir örneğe karşılık algılayıcıdan beklenen değer  $bc(i)$  ile ifade edilmektedir. Eğer beklenen çıkış  $bc(i)$ , ile algılayıcı çıkışı  $y(i)$  farklı ise ağırlık vektörü eşitlik 2.9'da gösterildiği gibi belirli bir iterasyon sayısına kadar değiştirilmektedir. Eğer  $y(i) = bc(i)$  ise ağırlıklar değiştirilmez ve algılayıcı eğitimi durdurulmaktadır.

### 2.7.2. Çok katmanlı algılayıcı model

ÇKA, Tek nöronlu algılayıcı tasarımını kullanan nöron birimlerinin katmansal bağlantısı ile elde edilen bir topoloji olarak ortaya çıkmıştır. Kurulan bağlantılar ileri yönlü(beslemeli) oluşturulmaktadır. Her bir nöron türevi alınabilen bir aktivasyon fonksiyonu yardımıyla ateşlemektedir. Aktivasyon fonksiyonu olarak yaygın kullanılan doğrusal, sigmoid ve hiperbolik tanjant fonksiyonları kullanılmaktadır. Genel yapısında, bir giriş katmanı, bir çıkış katmanı ve bir yada daha fazla gizli (ara) katman içermektedirler. Ara karmanda işlem birimleri, çıkış katman nöronları, önceki katmanın bütün nöronlarından girdi alacak şekilde ileri doğru bağlanmaktadır. Bunun yanında, girdi katmanı kendine iletilen girdi vektörünü aynen ara katmana iletmektedir. Ara katman ve çıkış katmanındaki her bir nöron önceki katmandan gelen ağırlıklı girdileri ve eşik değerini bir toplam fonksiyonuna aktarırlar. Nöronun toplam fonksiyonu eşitlik (2.1)' de olduğu gibi ifade edilmektedir. Nöronlar, toplam fonksiyonu ile elde edilen bilgiyi aktivasyon fonksiyonundan geçirerek sonraki katmanda bulunan nörona aktarırlar. Çıkış katmanı nöronlarının çıkışları ağın çıkışı olarak elde edilir. Öğrenme sürecinde, ileriye doğru çıkış hesaplandıktan sonra, çıktı katmanındaki her bir nöron için öğrenme hatası hesaplanmaktadır. Her bir nöronun çıkış hatası eşitlik (2.10)' daki gibi hesaplanmaktadır.

$$e = (bc_i - y_i) \quad (2.10)$$



Orada,  $i$ , örnek sayısı,  $bc$ , beklenen çıkış ve  $y$ , ağ çıkışını göstermektedir. Bazı durumlarda çıkış hataları negatif değer almakta ve toplam çıkış hatası sıfır çıkabilmektedir. Bu sorunun önüne geçmek amacıyla ağ hatası, çıkış katmanı nöron hatalarının kareleri toplanarak elde edilen toplam karesel hatanın karekökü ile hesaplanmaktadır. Ağ hatası eşitlik (2.11)'de ifade edildiği gibi tanımlanmaktadır.

$$E = \frac{1}{2} \sum_{i=1}^{cn} e^2 \quad (2.11)$$

Orada,  $E$ , ortalama karesel hatayı,  $cn$ , çıkış katmanı nöron sayısını ve  $e$ , nöronların çıkış hatalarını göstermektedir. Eğer elde edilen toplam hata istenilen düzeyde değilse hata, önceki katmanlardaki nöronlara doğru geriye türevi alınarak dağıtılmalıdır.

Çıkış katmanı ile gizli katman arasındaki ağırlıkların değişimi;

$m$ . gizli katman nöronu ile  $k$ . çıkış katmanı nöronu arasındaki ağırlık  $W_{mk}$  ile gösterilirse,  $i+1$ , zamanında ağırlığın yeni değeri eşitlik (2.12)'deki gibi ifade edilmektedir.

$$W_{mk}(i+1) = W_{mk}(i) + \Delta W_{mk}(i) + m\Delta W_{mk}(i) \quad (2.12)$$

Orada,  $m$ , momentum katsayısını  $\Delta W_{mk}$ , herhangi bir zamandaki ağırlıktaki değişim miktarını göstermektedir. Ağırlıktaki değişim miktarı eşitlik (2.13)'teki gibi ifade edilmektedir.

$$\Delta W_{mk}(i) = \alpha y_m \delta_k(i) \quad (2.13)$$

Orada,  $\alpha$ , öğrenme katsayısı,  $y_m$ , gizli katmanın  $m$ . nöron çıkışını,  $\delta_k$ , çıkış katmanı  $k$ . nöronunun hatasını ifade etmektedir. Çıkış katmanı nöronlarında meydana gelen türevi alınan maliyet hatası eşitlik (2.14)'te gösterilmiştir.

$$\delta_k(i) = y_k(1 - y_k)e \quad (2.14)$$

Orada,  $y_k$ , çıkış katmanı  $k$ . Nöron çıkışını,  $e$ ,  $k$ . Nöron hatasını ifade etmektedir.

Ara katmanlar arası yada ara katman giriş katmanı arasındaki ağırlık değişimi eşitlik (2.15)' te gösterilmiştir.

$$W_{im}(i+1) = W_{im}(i) + \Delta W_{im}(i) + m\Delta W_{im}(i) \quad (2.15)$$

Orada,  $m$ , momentum katsayısını  $\Delta W_{im}$ , herhangi bir zamandaki ağırlıktaki değişim miktarını göstermektedir. Ağırlıktaki değişim miktarı eşitlik (2.16)'daki gibi ifade edilmektedir.

$$\Delta w_{im}(i) = \alpha x_i(i) \delta_m(i) \quad (2.16)$$

Orada,  $\alpha$ , öğrenme katsayısı,  $x_i$ , giriş katmanı  $i$ . nöron çıkışını,  $\delta_m$ , ara katman  $m$ . nöron hatasını ifade etmektedir. Ara katman nöronlarında meydana gelen türevi alınan maliyet hatası eşitlik (2.17)' de gösterilmiştir.

$$\delta_m(i) = y_m(i)(1 - y_m(i)) \sum_{k=1}^l \delta_k(i) w_{mk}(i) \quad (2.17)$$

Orada,  $y_m$ , ara katman  $m$ . nöron çıkışını,  $l$ , sonraki katman nöron sayısını,  $\delta_k$ , sonraki katman (çıkış katmanı veya ara katman) türevi alınan maliyet hatasını,  $w_{mk}$ , ara katman ve çıkış katman arası ağırlıkları ifade etmektedir.

## 2.8. Performans(Doğruluk) Ölçüm Kriterleri

Ağın test işlemi sonucunda literatürde yaygın kullanılan bazı performans kriterleri kullanılmıştır. Bunlar; Ortalama Karesel Hata (MSE), Ortalama Mutlak Hata (MAE), Kök Ortalama Karesel Hata (RMSE) ve Belirlilik Katsayısı ( $R^2$ )' dir. Burada MSE hem eğitim sürecinde hem de test sürecinde performans ölçütleri olarak

hesaplanmaktadır. Eğitimin sonlandırılması sürecinde MSE ölçütü kullanılmaktadır. Ortalama Karesel Hata eşitlik (2.18)' deki gibi ifade edilmiştir.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - yb_i)^2 \quad (2.18)$$

Orada,  $N$ , örnek sayısını,  $y_i$  ve  $yb_i$  ise sırasıyla, beklenen çıkışı ve hesaplanan ağ çıkışını ifade etmektedir.

Ağın test fazında kullanılan istatistiksel doğruluk ölçütleri, MAE, RMSE ve  $R^2$  eşitlik (2.19-2.21)'deki gibi ifade edilmiştir.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - yb_i| \quad (2.19)$$

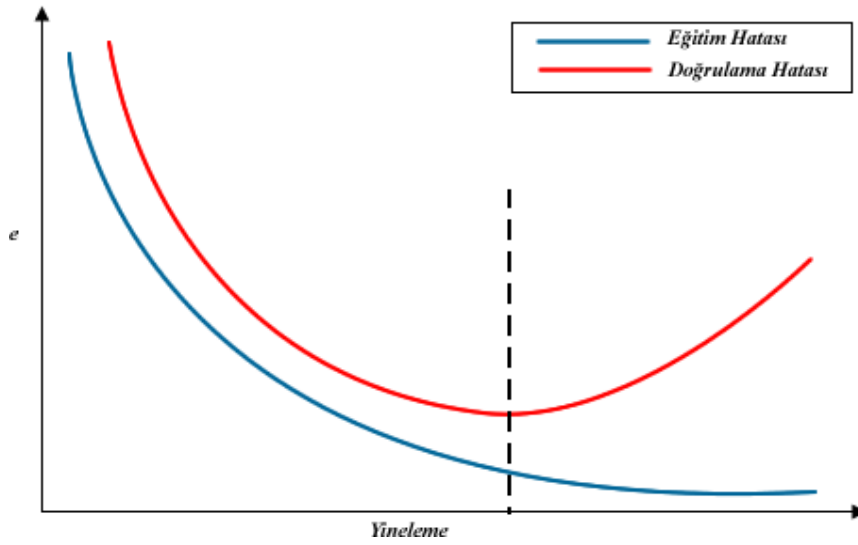
$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - yb_i)^2} \quad (2.20)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - yb_i)^2}{\sum_{i=1}^N yb_i^2} \quad (2.21)$$

İstatistiksel olarak, MAE, MSE VE RMSE değerlerinin 0'a yakın olması beklenmektedir. Bu durumda, modellenen YSA'nın doğru çıkışları ürettiği ifade edilmekte ve model doğruluğu kesin ifade edilmiş olmaktadır. Bir diğer önemli istatistiksel ölçüt ise Belirlilik katsayısıdır  $R^2$  ağın ürettiği çıktılar ile beklenen çıkışlar arasındaki ilişkiyi ölçmekte ve [0-1] arasında değerler almaktadır. Eğer 0'a yakın bir belirlilik katsayısı elde edilirse ağın çıktıları ile beklenen çıkışlar arasında ilişkinin zayıf ve yanlış sonuçlar elde edilmiş olduğu görülmektedir. 1'e yakın bir belirlilik katsayısı ise çıkışlar arasındaki ilişkinin büyüklüğünü istatistiksel olarak göstermekte ve ağın güvenilirliğini ifade etmektedir. Bu bağlamda ölçüm kriteri olarak  $R^2$ 'nin, 1'e yakın olması beklenmektedir.

## 2.9. Çapraz Doğrulama Metodu ve N Kat Çapraz Doğrulama Yöntemi

Yapay sinir ağlarının kullanıldığı problemlerde en büyük sorun eğitim ve test setinin belirlenmesidir. Aynı popülasyondan alınan veri örnekleriyle eğitilen bir ağ, test edildiğinde beklenen doğru çıkışı üretememekte ve model başarısız olmaktadır. Bu duruma literatürde ağın ezberlemesi denilmektedir. Şekil 2.9’ da bir ağın eğitim sürecinde ezberleme süreci gösterilmiştir.



Şekil 2.9. Eğitim sürecinde ağın ezberlemesi süreci

Ezberleme problemine çözüm olarak Çapraz Doğrulama metodu önerilmiştir [61-62]. Pek çok metod sunulsa da, en yaygın kullanılanı N Kat Çapraz doğrulama metodudur. Metod, tüm veri setinin eğitim sürecine dâhil edilmesini amaçlamakta ve böylece eğitim setinin seçimi problemini ortadan kaldırmaktadır. Veri seti N parçaya bölünmekte ve rastgele bir parça test (Validation) için, geriye kalan N-1 parça ise eğitim için kullanılmaktadır. Elde edilen veri setleriyle ağ eğitilir ve test hatası hesaplanmaktadır. Bu süreç N defa tekrar edildikten sonra hesaplanan test hatalarının ortalaması alınarak ağın test hatası belirlenmektedir [63-64].

Çalışmada, Çapraz doğrulama metodunun her bir parçası için test hatası olarak MAE, RMSE hata parametreleri kullanılmış ve ek olarak, istatistikî doğruluk parametresi  $R^2$  hesaplanmıştır. N tekrar sonrasında ağın toplam performansı ise bu parametrelerin ortalaması ile hesaplanmaktadır. Ağın ortalama test performansı eşitlik 2.22’de tanımlandığı gibidir.

$$[E_T, R_T^2, RMSE_T] = \frac{1}{k} \sum_{i=1}^k E_i, R_i^2, RMSE_i \quad (2.22)$$

Orada, k, kat sayısını ifade etmektedir. Diğer parametreler istatistiksel doğruluk ve hata değerleridir.

## BÖLÜM 3. KARMAŞIK AĞLAR VE ÇİZGE TEORİSİ

### 3.1. Karmaşık Ağlar

Karmaşık ağlar, önemli karakteristik özelliklere sahip olan ve gerçek ağlara benzer davranan ağ yapıları olarak tanımlanmaktadır. Bu nedenle, son yıllarda farklı alanlarda karmaşık ağ kavramı ilgi odağı olmuştur [65-69]. Bu alanlarda kullanılan ağlara örnek olarak, Elektrik Şebeke Ağları, Metabolik ve gen ağları [70], bilimsel akademisyen yayın ağı [71], biyolojik osilatör ağları [72-76], sensör ağları [77-80], otonom insansız araçların hareketi için oluşturulan ağ kümeleri [81], kimyasal bağlar sonucu oluşan, kimyasal ağlar, router ve pc'lerin birbirine kablo yada kablosuz bağlantısıyla oluşan internet ağı, insanlar arasındaki farklı sosyal ilişkiler sonucu oluşan sosyal ağlar [22], web sayfaları arasında kurulan köprüler ile oluşturulan www ağı sayılabilmektedir [82-83]. Gerçek dünya ağları, yapıları itibariyle 2 model ile açıklanmaktadır.

Birinci model: Bu model, gerçek ağlarının düğümleri arasında oluşturduğu kısa ilişkiyel bağlantılar kullanılarak oluşturulmuştur. Modelde nöronlar arası kısa bağlantılar oluşturularak bilginin en hızlı nöronlar arasında taşınması kuramını benimsemektedir. Bu durum “Küçük Dünya” teoremiyle açıklanmıştır ve bu model Küçük Dünya ağı olarak isimlendirilmiştir. Bu modele örnek olarak sunulan İnternet ağı incelendiğinde, ağ içerisinde paketler gönderilmeden önce link uzunlukları hesaplanarak en kısa yollar seçilerek gönderildiği gözlemlenmektedir [84].

İkinci model: bu modelde, ağın büyümesi ağ derecesine ( $k$ ) bağlı olarak değişmektedir.  $k$ , bir güç kanununu takip ederek değişiyorsa veya  $k$ 'nın tersi oranıyla azalıyorsa bu ağ yapısı “ölçeksiz ağ” olarak tanımlanmaktadır [66,85-86]. Hazırlanan çalışmada karmaşık ağ modellerinden, Küçük Dünya ağları seçilmiştir.

### 3.1.1. Küçük Dünya ağları

Nöronal sistem, reseptörleri yardımıyla, dış dünyadan bilgileri alır ve nöron ağı içerisinde kodlayarak depo etmektedir. Bu işlemin gerçekleşmesinde öğrenme sürecinde sinaptik bağlantıların değiştirilmesi veya nöronlar arası yeni bağlantıların oluşturması önemli bir süreç olarak karşımıza çıkmaktadır. Bu bağlamda, görülüyor ki bir nöron ağının davranışında ağın topolojik yapısı önemli bir etkidir. Bu nedenle, son zamanlarda yapılan çalışmalar, ağ topolojilerinin ifade edilmesi, özelliklerinin belirlenmesi ve yeni ağ topolojilerinin bilim dünyasına girmesine neden olmuştur [12-15]. Elde edilen bulgular ışında gerçek dünya ağlarının yapısal özelliklerine göre; Small-World(Küçük Dünya) yada Scale-Free(Ölçeksiz) ağ yapısına sahip olduğu vurgulanmıştır [87].

Kavramsal olarak ağ yapıları incelendiğinde, ağ benzer birimsel(nöronal) özellikler içeriyorsa düzenli ağ yada karmaşık nöronal özelliklerinden dolayı rastsal ağ olarak tanımlanmaktadır. Son zamanlarda yapılan araştırmalarda, bazı ağların düzenli ile rastsal ağlar arasında olduğu kabul edilmiş ve onların ne tamamen düzenli ağ, nede tamamen düzensiz ağ özelliği gösterdiği gözlemlenmiştir. Elde edilen yeni ağ topolojisinin yapısal olarak “Küçük-Dünya” teoremine uygun olduğu ortaya konulmuştur. “Küçük-Dünya” teoremi incelendiğinde teoremi açıklamak için ortak bir kurgudan bahsedilmektedir. Bu kurgu bir rastsal parti kurgusudur.

Kurgu: Ahmet parti verecektir ve partiye davet etmek amaçlı bulunduğu şehrin sokaklarını dolaşmaktadır. Rastgele karşısına çıkan Soner ile tanışır. Ahmet, Soner ile alışkanlıkları, hobileri, özellikleri üzerine konuşurken, cümle içerisinde Soner’in futbol oynamayı çok sevdiği ve genellikle en yakın arkadaşı Korhan ile futbol oynadıklarından bahsetmektedir. Korhan ismini duyan Ahmet yıllar önce birlikte aynı ilkokula gittiği arkadaşı Korhan’ı hatırlayacaklardır. Korhan İlkokulu bitirdikten sonra o mahalleden taşınmıştır. Daha sonra Ahmet Soner ile Korhan hakkında konuştuğunda, Korhan’ın yıllardır görmediği arkadaşı olduğunu fark ettiğinde bunun nasıl olabileceğini Soner ile düşünmüşler ancak bir cevap bulamamışlardır. Sonuç olarak, her ikisi de “Dünya Küçük mü?” sorusu ile durumu sorgulamışlardır.

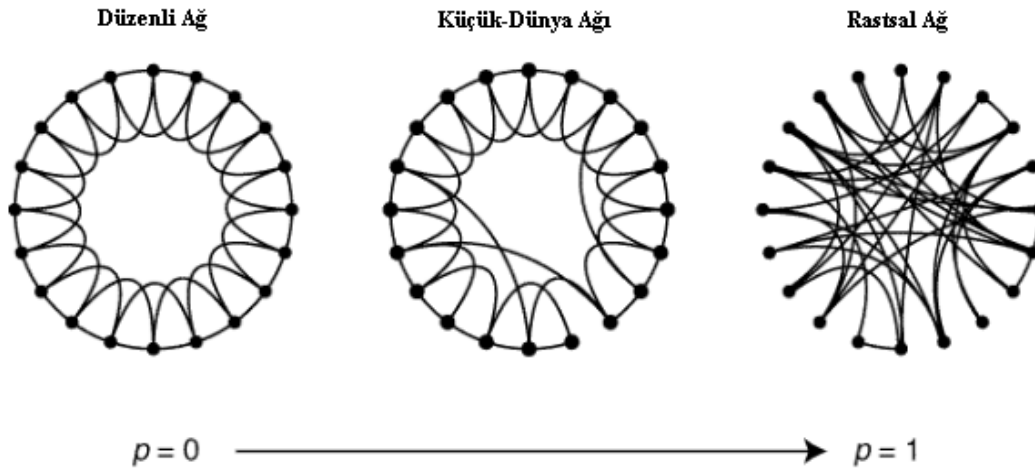
Bu kurgunun temeli 1967 yılında sosyolog Stanley Milgram tarafından atılmış ve “Küçük-Dünya” teoremi ilk defa ortaya konulmuştur. Yapılan deneyde, Nebraska eyaletinin Omaha ve Kansas eyaletinin Wichita şehirlerinden, Massachusetts eyaletinin Boston şehrinde bir adrese, seçilen adres hakkında herhangi bilgisi bulunmayan insanlar seçilerek 296 mektubun ulaştırılması istenmektedir. Mektuplar insanlar arasında aktararak iletilmektedir. İletilen mektuplar incelendiğinde, mektupların hepsinin yaklaşık 6 farklı insan üzerinden hedef adrese ulaştığı gözlemlenmiş ve “dünya küçüktür” denilmiştir [4,88]. Daha sonraları John Guare bu teoremi “Altı Dereceli Ayrım” şeklinde isimlendirmiştir [88-90]. Yapılan deneysel çalışma bir insanın dünyadaki başka bir insanla, komşuluk ilişkileri üzerinden ortalama altı farklı kişi üzerinden bağlantı kurabileceğini göstermiştir. Bu davranış özelliği karmaşık ağların açıklanmasında önemli bir problemi çözmüştür. Böylece bu tür ağ yapılarına Küçük Dünya Ağları denilmiştir. Zaman içerisinde, çok farklı Küçük Dünya Ağları modelleri önerilmiştir. Ancak bunlardan Watts-Strogatz ve Newman-Watts Küçük Dünya ağ modelleri en yaygın bilinen yapılarıdır.

### 3.1.1.1. Watts-Strogatz modeli

1998 yılında Watts ve Strogatz tarafından yapılan çalışmada Küçük Dünya kuramının karmaşık ağların tanımlanmasında önemli bir yapı taşı olduğu vurgulanmış ve bazı karmaşık dünya ağlarının modellenmesi için dinamik sistem ağları kullanılmıştır. Kullanılan dinamik sistem ağının, düzenli ve rastsal ağlardan yapısal farklılıkları olduğu gözlemlenmiştir. Bu tür ağların düzenli ağlar gibi yüksek kümelenmeye sahip ve rastsal ağlar gibi nöronları arasında kısa yollara sahip olduğu gözlemlenmiştir. Küçük Dünya ağı karakteristiği düzenli ve rastsal ağ kavramı arasında konumlandırılmıştır [7,90]. Yüksek kümelenme ve düşük kısa yol uzunluğuna sahip topolojik model olarak dairesel çizgelerin uygun olduğu vurgulanmıştır [91]. Bu nedenle dairesel yönsüz çizge kullanılmıştır. Çizgede,  $N$  adet nöron kullanılmış, her nöron için ise  $k$  bağlantı sayısı tanımlanmıştır. Bunlara ek olarak ağın bağlantı değiştirme olasılığı olarak tanımlanan  $p$  parametresi kullanılmıştır. Ağın toplam bağlantı sayısı,  $kN/2$  formülü ile elde edilmiş ve yenilenecek bağlantı sayısı  $kNp/2$  ifadesi ile tanımlanmıştır. Burada,  $p=0$  iken ağ düzenli,  $p=1$  iken ağ düzensiz ağ olarak kabul edilmiştir.  $0 < p < 1$  durumunda ise ağın

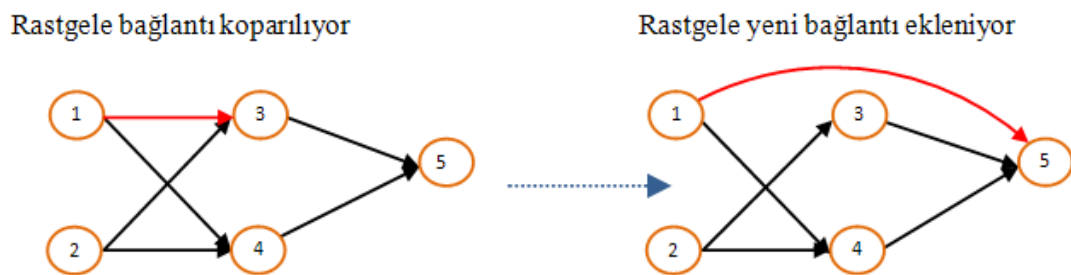


Küçük Dünya ağı olduğu vurgulanmıştır. Çalışmada, 3 farklı ağ modeli oluşturulmuştur. 20 nöronlu ve nöron derecesi 4 olan dairesel latis kullanılarak elde edilen sırasıyla, düzenli, Küçük Dünya ve rastsal ağların topolojik yapısı Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Watts-Strogatz tarafından önerilen düzenli, Küçük Dünya ve rastsal ağların topolojik yapısı [7].

Düzenli ağdan, rastsal ağın oluşturulması işleminde toplam bağlantı sayısı sabit tutulmuş ve yeni bağlantı eklemede rastgele seçilen nöron çiftlerinin sadece son noktası değiştirilmiştir. Watts-Strogatz modelinde kullanılan bağlantı yenileme kurgusu Şekil 3.2’de gösterilmiştir.



Şekil 3.2. Watts-Strogatz Küçük Dünya ağı bağlantı yenileme işlemi [92].

Ağları sınıflandırmak amacıyla, nöronlar arasındaki ayrımı ifade eden, karakteristik yol uzunluğu(L) ve bir nöronun komşularının küme oluşturma durumunu tanımlayan, C, kümelenme katsayıları kullanılmıştır. Buna göre düzenli ağın L ve C parametreleri

büyük, rastsal ağın  $C$  ve  $L$  parametreleri küçük ve Küçük Dünya ağlarının ise  $C$  parametresi büyük iken  $L$  parametresinin küçüktür [7].

Karakteristik En Kısa Yol Uzunluğu,  $L$ , bir  $G$  çizgesinin düğümleri arasındaki mesafenin ölçülmesiyle hesaplanmaktadır ve denklem 3.1'deki gibi tanımlanmaktadır.

$$L = \frac{1}{N(N-1)} \sum_{i \neq j \in N} d_{ij} \quad (3.1)$$

Eşitlikte,  $N$  çizgede bulunan düğüm sayısını,  $d_{ij}$  ise  $i$  düğümünden  $j$  düğümüne gitmek amacıyla geçilen bağlantı sayısını ifade etmektedir.

Kümelenme katsayısı ise,  $G$  çizgesinde bulunan düğümlerin kümelenme katsayılarının aritmetik ortalamasıdır. Her bir düğümün kümelenme katsayısı bir düğümün en yakın komşuları arasındaki var olan bağlantı sayısının, komşular arasında olası maksimum bağlantı sayısına oranı olarak tanımlanır ve aşağıdaki gibi ifade edilmektedir.

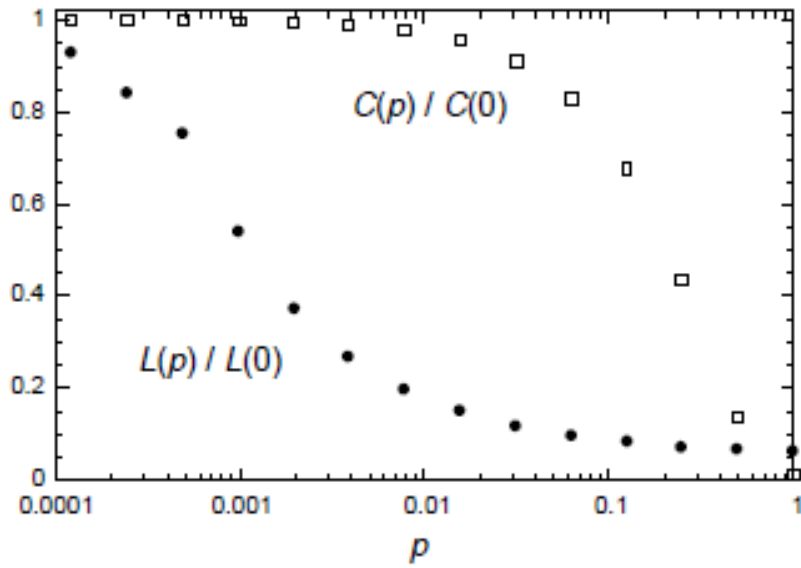
$$C_i = \frac{G_i BS}{k_i(k_i - 1) / 2} \quad (3.2)$$

$$C = \frac{1}{N} \sum_{i \in N} C_i \quad (3.3)$$

Eşitlikte,  $k_i$  düğümün derecesi olarak tanımlanır ve bir düğümün direkt komşu sayısına eşittir.  $G_i BS$  ise  $i$ . düğümün komşuları arasında var olan bağlantı sayısıdır.

Ağ bağlantılarının değiştirilmesi süreci, düzenli ağ yapısının oluşturulması ile başlamaktadır. Elde edilen düzenli ağın ( $p=0$ ) bağlantıları rastsal ağ ( $p=1$ ) elde edilene kadar  $p$  olasılık değerinin 0-1 arasında değişimi ile yenilenmektedir. Bu süreçte her

$p$ , olasılığı için yeniden bağlantıları ayarlanan dairesel çizgenin  $C$  ve  $L$  parametreleri hesaplanarak Şekil 3.3' te gösterilmiştir.



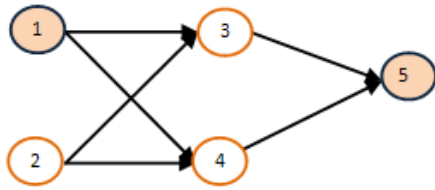
Şekil 3.3. Bağlantı değiştirme olasılığı ile  $C$  ve  $L$ 'nin değişimi [7].

Watts ve Strogatz tarafından önerilen Küçük Dünya ağı modeli, karmaşık dünya ağlarının ifade edilmesinde öncülük etmiştir. Bu bağlamda, bilimsel araştırmalarda yeni karmaşık ağ modellerinin önerilmesine olanak sağlamıştır.

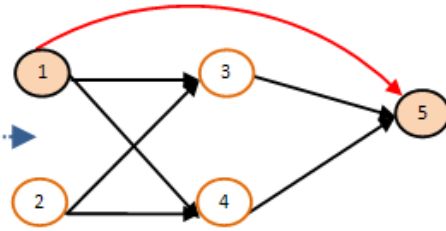
### 3.1.1.2. Newman-Watts Küçük Dünya ağı modeli

1999 yılında Newman ve Watts, Watts ve Strogatz modelini kullanarak ağın analizini kolaylaştıracak bir kablolama teorisi ortaya koymuştur. Önerilen yapıda rastgele bağlantı kesilmemekte ve onun yerine rastgele seçilmiş bağlı olmayan nöronlar arasında bağlantılar eklenmektedir. Böylece Watts-Strogatz modelinde var olan bağlı olmayan nöron çiftlerinin arasındaki yol uzunluğunun sonsuz olma probleminin önüne geçilmektedir. Newman-Watts Küçük Dünya ağlarında kullanılan bağlantı ekleme işlemi Şekil 3.4' te gösterilmiştir [8,93].

Rastgele bağı olmayan nöron çiftleri seçiliyor



Yeni bağlantı ekleniyor



Şekil 3.4. Newman-Watts yeni bağlantı ekleme işlemi

### 3.1.1.3. Latora ve Marchiori metodu

2000 yıllarında V. Latora, ve M. Marchiori, yaşantımızın her anında karmaşık ağlarla karşı karşıya olduğumuzu belirtmiş ve bu ağların çoğunun Küçük Dünya ağı özelliği gösterdiğini ifade etmiştir[20]. Yapılan araştırmada, ağ yapısında bağlanmamış nöron çiftleri bulunan ayırık ağlar için Watt-Strogatz tarafından önerilen Küçük Dünya Ağı matematiksel modelinin, bağı olmayan nöron çiftlerinin arasındaki mesafenin sonsuz olmasından dolayı kullanılamayacağı ifade edilmiş ve onların yerine yeni bir model önerilmiştir. Önerilen modelde ağın global ve lokal özelliklerini tanımlayan C(Kümelenme Katsayısı) ve L(Karakteristik yol uzunluğu) parametrelerinin yerine benzer  $D_{lokal}$  ve  $D_{global}$  bağlantı uzunluk katsayıları tanımlanmıştır. Bu bağlamda, L'nin,  $D_{global}$ 'e benzer ve C'nin ise  $1/D_{lokal}$ 'e benzer davranış göstermekte olduğu vurgulanmıştır [20-24].

Bir ağın global bağlantı uzunluk katsayısı aşağıdaki gibi ifade edilmektedir [20-23,93]:

$$D_{Global} = \frac{N(N-1)}{\sum_{i \neq j \in N} d_{ij}} \quad (3.4)$$

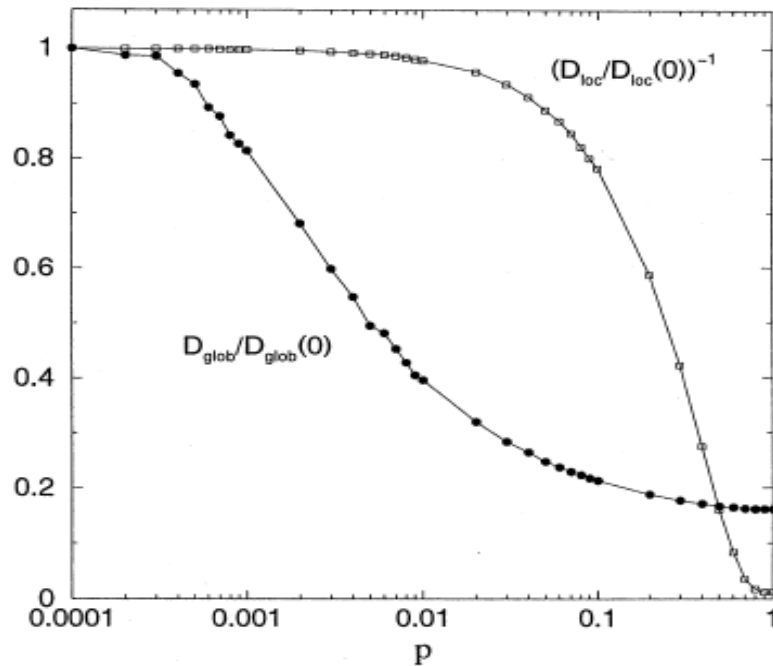
Eşitlikte,  $N$  ağda bulunan düğüm(nöron) sayısını,  $d_{ij}$   $i$ . ve  $j$ . Düğümleri arasındaki mesafeyi göstermektedir.

Çizgenin lokal bağlantı uzunluk katsayısı ise aşağıdaki gibi ifade edilmektedir [20-23]:

$$D_{Lokal} = \frac{1}{N} \sum_{i \in N} D(G_i) \quad (3.5)$$

$$D(G_i) = \frac{N_i(N_i - 1)}{\sum_{k \neq l \in N} d_{kl}} \quad (3.6)$$

Burada  $N_i$ , düğümün bağlı olduğu komşu sayısını,  $d_{kl}$  ise seçilen düğüm çıkarıldığında kalan alt çizgenin düğümleri arasındaki en kısa yol uzunluğunu ifade etmektedir [20-23].  $D_{Global}$  ve  $D_{Lokal}$  parametre değerlerinin her ikisinin de küçük olduğu aralıkta Küçük Dünya ağ yapısı elde edilmektedir [22-24]. Bağlantı değiştirebilme olasılığıyla Global ve lokal bağlantı uzunluk katsayıları değişimi Şekil 3.5' te gösterilmiştir.

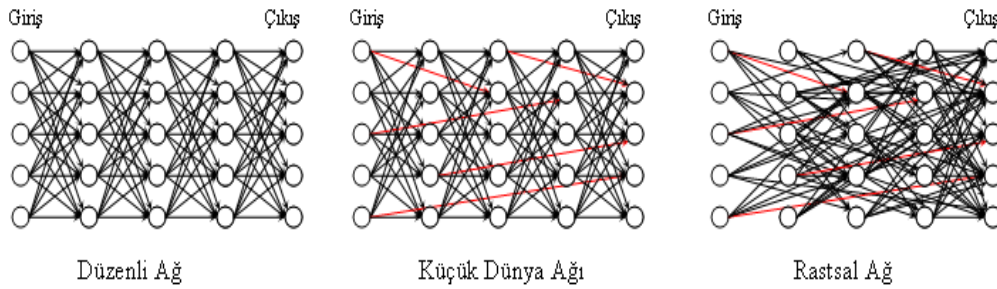


Şekil 3.5. Bağlantı değiştirebilme olasılığı(p) ile  $D_{global}$  ve  $D_{lokal}$ 'in değişimi [20].

#### 3.1.1.4. Simard ileri yönlü Küçük Dünya ağ modeli

2005 yılında D. Simard ve arkadaşları ilk defa ileri yönlü çok katmanlı bir yapay sinir ağı kullanmış, kullanılan ağda Watts-Strogatz modelinden farklı bağlantı

yenileme işlemi kullanarak ileri yönlü yapay Küçük Dünya ağını elde etmiştir[23]. Geleneksel yapay sinir ağı topolojisi, düzenli ağ olarak tanımlanıp, bağlantı yenileme işlemi yardımıyla bağlantılar rastgele yenilenmiştir. Olası tüm bağlantuların yenilenmesiyle elde edilen ağ yapısına ise rastsal ağ denilmiştir. Geleneksel çok katmanlı bir yapay sinir ağı kullanılarak, bağlantı yenileme işlemi ile Küçük Dünya ağı ve rastsal ağ topolojilerinin elde ediliş şeması Şekil 3.6'da gösterilmiştir.



Şekil 3.6. İleri yönlü yapay sinir ağları için, düzenli, Küçük Dünya ve rastsal ağ topolojileri şeması [93].

Bağlantıların yenilenmesi sürecinde, Watts-strogatz modelinde olduğu gibi düzenli ve rastsal ağ arasında bir yerde Küçük Dünya ağı varlığı gösterilmiştir. Model, geleneksel karesel bir yapay sinir ağı topolojisini kullanmaktadır. Bu topolojinin toplam bağlantı sayısı katmandaki nöron sayısı ve katman sayısı kullanılarak aşağıdaki gibi hesaplanmıştır.

$$K = N_{nöron}^2 * (N_{katman} - 1) \quad (3.7)$$

p olasılığı ve katmandaki nöron sayısı kullanılarak, YBS değeri aşağıdaki gibi elde edilmiştir.

$$YBS = N * p \quad (3.8)$$

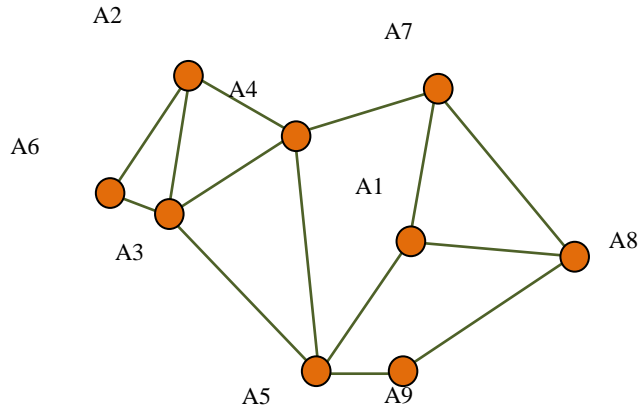
Modelde ağlar elde edilirken ağın toplam bağlantı sayısı değiştirilmemektedir. Bağlantı yenileme işlemi; rastgele seçilen 2 nöronun bağlantısının koparılmasıyla başlamakta ve yine daha önce aralarında bağlantı bulunmayan iki nöronun seçilerek aralarında yeni bağlantının eklenmesi ile devam etmektedir. Bu işlem ağ içerisinde

bağlı olmayan nöron çifti kalmayana kadar sürmektedir. Her bir bağlantı değiştirildiğinde eşitlik 3.4-3.6'da ifade edilen lokal ve global bağlantı uzunluk katsayıları hesaplanmaktadır. Ve bu iki katsayının küçük olduğu topolojiler Küçük Dünya ağı kabul edilmiştir. Elde edilen Küçük Dünya ağının öğrenme ve test performansları perseptron öğrenme algoritması ile test edilmektedir.

### 3.2. Çizge Teorsisi

Çizge teorisi matematiksel olarak karmaşık ağ yapılarının mimari modelini çıkarılması ve yerleşim problemlerinin çözümü için kullanılan bir bilim dalıdır. İlk olarak 1736'da Leonhard Euler [94] tarafından temeli atılmış bir kuram olarak günümüze ulaşmıştır. Teoremden, karmaşık ağ, G çizgesi olarak tanımlanmakta, ağın birimleri ise köşe (düğüm) olarak ifade edilmektedir. Ağın içerdiği birimler arası yollar kenar (bağlantı) şeklinde tanımlanmaktadır [95-99]. Bir G çizgesi, A adet düğüm içermekte ve bu düğümler arasında K, adet kenar bulundurmaktadır. Her bir  $k \in K$  olmak koşuluyla, iki x ve y köşesi ilişkilendirilebilmektedir. Eğer köşelerde bir yön bilgisi yoksa  $k=(x,y)$  veya  $k=(y,x)$  yazılabilir. Köşeleri arasında ifade edildiği gibi yön bilgisi olmayan ve her iki yönde de ilişki kurulabilen çizgelere yönsüz çizge denilmektedir.

Çizge içerisinde köşeler arasında yön bilgisi varsa ve çizgenin kenarları sıralı çiftler olarak tanımlanmış ise çizgeler yönlü çizge olarak ifade edilmektedir. Köşeler arası oluşan kenarların maliyetinin bulunduğu çizgeler ağırlıklı çizge olarak tanımlanmaktadır [100]. Gerçek bulgulardan hazırlanan yönsüz çizgenin ifadesi Şekil 3.7' de gösterilmiştir.



Şekil 3.7. Tasarlanmış yönsüz çizge modeli.

### 3.2.1. Bağlantı matrisi

Bir  $G$ , çizgesinin köşeleri arasındaki bağlantıları ifade eden matris olarak tanımlanmaktadır. Yönsüz ve yönlü çizgeler için bu matris, direkt bağlı köşe çiftleri arasındaki bağlantı değerinin 1, bağlı olmayan düğüm çiftleri arasındaki bağlantı değerinin ise 0 alınması ile elde edilmektedir. Ağırlıklı çizgede ise matris, düğümler arasında tanımlanmış maliyet değerleri kullanılarak oluşturulmaktadır [95-99]. Şekil 3.7' de tanımlanan çizgenin bağlantı matrisi Şekil 3.8' de verilmiştir.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0	0	0	0	1	0	1	1	0
A2	0	0	1	1	0	1	0	0	0
A3	0	1	0	1	1	1	0	0	0
A4	0	1	1	0	1	0	1	0	0
A5	1	0	1	1	0	0	0	0	1
A6	0	1	1	0	0	0	0	0	0
A7	1	0	0	1	0	0	0	1	0
A8	1	0	0	0	0	0	1	0	1
A9	0	0	0	0	1	0	0	1	0

Şekil 3.8. Bağlantı matrisi.



### 3.2.2. En kısa yolun bulunması

İlişkisel düğümlerin oluşturduğu dünya ağlarında, en kısa yol problemi en önemli sorun olarak görülmekte ve bu bağlamda pek çok çözüm algoritmaları sunulmaktadır. Bu algoritmaların ortak noktaları düğümler arasında maliyet bakımında en kısa yolun hesaplanmasıdır. Bu çalışmada, Floyd-Warshall en kısa yol algoritması kullanılmıştır

#### 3.2.2.1. Floyd-Warshall algoritması

Daha çok yönlü ve maliyetli çizgelerin düğümleri arasındaki en kısa yolu bulmak için kullanılan basit bir algoritmadır. Algoritmada, çizge içerisindeki her bir (x,y) düğüm çifti için en kısa yol bulunmaktadır. Hesaplama sürecinde, iki düğüm arasındaki mesafe hesaplanırken, 3. bir nöron üzerinden de en kısa yol hesaplanmakta ve hesaplanan yollardan en kısa olanı alınmaktadır. Kısa yolu bulma çevrimi uzun sürse bile her zaman kesin sonuç vermektedir. Eşitlik 3.9'da olduğu gibi kısa yolu sorgulamaktadır [101-102].

$$G[x, y] = \text{Math.Min}(G[x, y], G[x, k] + G[k, y]); \quad (3.9)$$

Bağlantı matrisinden hareketle mesafe matrisi elde edilir burada, aralarında bağlantı olan çiftler 1 değeri, olmayan çiftler  $\infty$  değeriyle ifade edilmektedir. Şekil 3.7' de gösterilen çizge için bağlantı matrisinden elde edilen başlangıç mesafe matrisi Şekil 3.9' da gösterilmiştir.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	$\infty$	$\infty$	$\infty$	$\infty$	<b>1</b>	$\infty$	<b>1</b>	<b>1</b>	$\infty$
A2	$\infty$	$\infty$	<b>1</b>	<b>1</b>	$\infty$	<b>1</b>	$\infty$	$\infty$	$\infty$
A3	$\infty$	<b>1</b>	$\infty$	<b>1</b>	<b>1</b>	<b>1</b>	$\infty$	$\infty$	$\infty$
A4	$\infty$	<b>1</b>	<b>1</b>	$\infty$	<b>1</b>	$\infty$	<b>1</b>	$\infty$	$\infty$
A5	<b>1</b>	$\infty$	<b>1</b>	<b>1</b>	$\infty$	$\infty$	$\infty$	$\infty$	<b>1</b>
A6	$\infty$	<b>1</b>	<b>1</b>	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
A7	<b>1</b>	$\infty$	$\infty$	<b>1</b>	$\infty$	$\infty$	$\infty$	<b>1</b>	$\infty$
A8	<b>1</b>	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	<b>1</b>	$\infty$	<b>1</b>
A9	$\infty$	$\infty$	$\infty$	$\infty$	<b>1</b>	$\infty$	$\infty$	<b>1</b>	$\infty$

Şekil 3.9. Başlangıç mesafe matrisi.

Algoritmada, mesafe matrisi kullanılarak, her x düğümünden y düğümüne gitmek için yeni bir k düğümü üzerinden gidilecek diğer bir yol var ise ve bu yeni yolun maliyeti daha az ise, x'den y'e en kısa yol yeni yol seçilmektedir. Floyd-Warshall algoritmasının sözde kodu Şekil 3.10' de verilmiştir.

```

for k=1 to nrons
  for x=1 to nrons
    for y=1 to nrons
      G[x, y] = Min(G[x, y] , G[x, k] + G[k, y]);

```

Şekil 3.10. Floyd-Warshall algoritmasının sözde kodu [102].

Şekil 3.7' de tanımlanan çizgenin Floyd-Warshall algoritması uygulandıktan sonra elde edilen mesafe matrisi Şekil 3.11' de gösterilmiştir.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	2	3	2	2	1	3	1	1	2
A2	3	2	1	1	2	1	2	3	3
A3	2	1	2	1	1	1	2	3	2
A4	2	1	1	2	1	2	1	2	2
A5	1	2	1	1	2	2	2	2	1
A6	3	1	1	2	2	2	3	4	3
A7	1	2	2	1	2	3	2	1	2
A8	1	3	3	2	2	4	1	2	1
A9	2	3	2	2	1	3	2	1	2

Şekil 3.11. Mesafe matrisi.

Mesafe matrisiyle elde edilen düğümler (nöronlar) arası mesafeler eşitlik 3.4-3.6 kullanılarak global ve lokal bağlantı uzunluk katsayıları hesaplanmaktadır. Hesaplanan katsayılardan hareketle katsayıların her ikisinin de küçük olduğu aralık Küçük Dünya ağları aralığı olarak belirlenmektedir. Bu aralıkta bağlantısı yenilen nöron ağları Küçük Dünya ağları özelliği göstermektedir.

## BÖLÜM 4. KARMAŞIK AĞ YAZILIMININ HAZIRLANMASI VE KULLANIMI

### 4.1. Karmaşık Ağ Yazılımının Hazırlanması

Bu bölümde Küçük Dünya ağları deneylerinin gerçekleştirildiği yazılımının tanıtımı yapılacaktır. Yazılımın gelişiminde ilk adım, Küçük Dünya ağı oluşumu için gerekli olan matematiksel ve topolojik algoritmaların kurulmasıdır. Yazılımın geliştirilmesinde ikinci adım, dinamik parametreler ile ayarlanabilen bir yapay sinir ağı yazılım ortamı hazırlanmasıdır. Her iki adım için gerekli parametreler ve model sonuçları grafik arayüz üzerinde tanımlanmıştır. Bu temel yaklaşımla hazırlanan yazılım platformunun görsel arayüzü Şekil 4.1’ de gösterilmiştir.

The screenshot displays the SWANN Gui v0.1 interface, which is divided into several sections for configuring and running a neural network simulation.

- Network Parameters:** Includes Training Parameters (Number of Samples: 80, Dataset Normalization: Max 1, Min 0), Initial Parameters (Iteration Number: 1000, Lr: 0.5, mu: 0.9), and Activation Functions (Uni-Polar Sigmoid).
- Complex Network Parameters:** SW Network Types (Watt-Strogatz), Rewiring Number (0), and Network Connectivity.
- Processes:** Cross-Validation (checked), Fold Number (10), CV Training Error (0.0003), and CV Test Error (0.039983).
- Errors:** Training Error (mse) (0.0003000014) and Test Error (mae) (0.0421113487).
- Network Building:** A button labeled 'Network' and a display showing '6480'.
- Dataset:** A 'Load Dataset' button, checkboxes for 'Input Data' and 'Output Data', and 'Sample Number: 80'.
- Dataset List:** A table with columns 'Inputs' and 'Outputs' showing a list of input and output values for 20 samples.
- Test Results:** A table with columns 'Education Outputs' and 'Test Outputs' showing experimental and network output values for 10 samples.

Inputs	Outputs
1	10
1	11
1	12
1	13
1	14
1	15
1	16
1	17
2	10
2	11
2	12

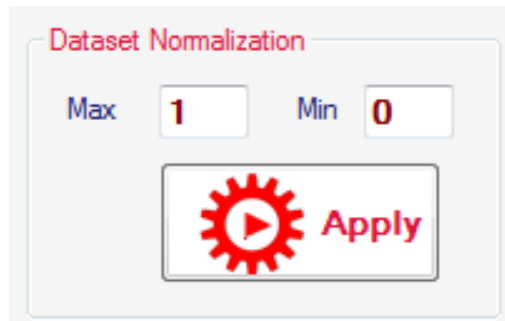
Education Outputs	Test Outputs
0.442025501	0.39682349058991
0.254974521	0.194276977835175
0.124256016	0.100723367329816
0.129632479	0.100998873068363
0.08897136	0.098848306791317
0.096738131	0.0886379200962091
0.18570767	0.137885751331568
0.221375661	0.156715210375178
0.449941301	0.398895755365248
0.227655822	0.178109813085745

Şekil 4.1. Tasarlanan kullanıcı arayüzü.

## 4.2. Karmaşık Ağ Yazılımının Kullanımı

### 4.2.1. Veri setinin normalizasyonu

Hazırlanan yazılım ortamında, veri seti ağa sunulmadan önce belirli bir aralıkta normalize edilmektedir. Bu işlem için yaygın kullanılan normalizasyon algoritmalarından biri olan min-max algoritması tanımlanmıştır. Kullanıcı başlangıç aşamasında verilerin normalizasyonu için minimum ve maksimum aralığını seçmektedir. Yazılımın normalizasyon fazı Şekil 4.2’de gösterilmiştir.



```

int maxd, mind;
double[,] normalize(double[,] v1, double mi, double ma)
{
    maxd = Convert.ToInt16(inpmax.Text);
    mind = Convert.ToInt16(inpmin.Text);
    double[,] dd = new double[v1.GetLength(0), v1.GetLength(1)];

    for (i = 0; i < v1.GetLength(0); i++)
    {
        for (j = 0; j < v1.GetLength(1); j++)
        {
            dd[i, j] = ((v1[i, j] - mi) * maxd) / (ma - mi) +mind;
        }
    }
    return dd;
}

double revnormalize(double v1, double mi, double ma)
{
    double dd;

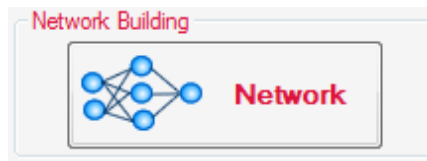
    dd = ((v1 - mind) / maxd) * (ma - mi) + mi;
    return dd;
}

```

Şekil 4.2. Yazılımın normalizasyon fazı ve c# kodu.

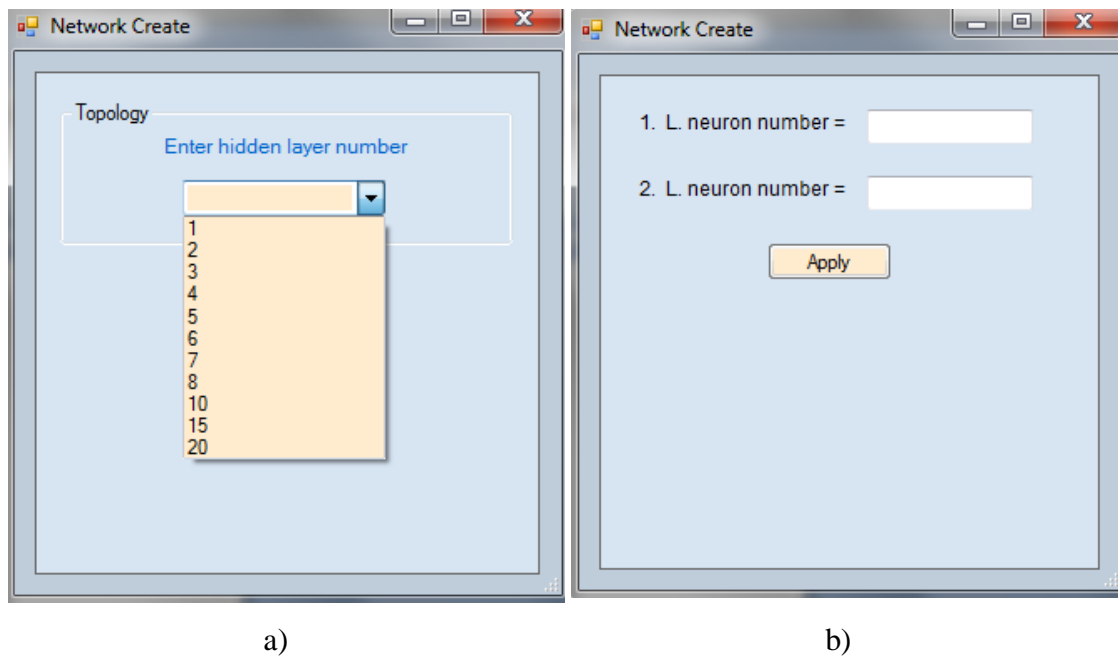
#### 4.2.2. Network topolojisinin oluşturulması

Karmaşık ağ yapısının tasarlanmasında geleneksel ileri yönlü yapay sinir ağları topolojisi temel alınmıştır. Hazırlanan yazılımda ilk adım olarak temel yapay sinir ağı topolojisi kullanıcı tarafından seçilmelidir. Bunu gerçekleştirmek için, ana program penceresinden network butonu yardımıyla ağ oluşturma penceresi açılmaktadır. Network butonu Şekil 4.3'te gösterilmiştir.



Şekil 4.3. Ağ oluşturma paneli.

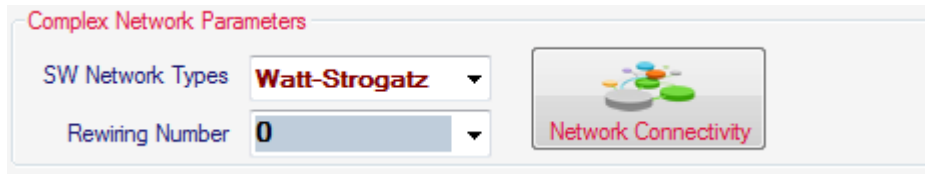
Ağ oluşturma penceresinde kullanıcılar, gizli katman sayısını seçmekte sonrasında ise seçilen gizli katmanlar için nöron sayısını belirleyebilmektedir. Apply butonu ile temel ağ topolojisi tasarımını gerçekleştirmektedir. Network oluşturma penceresi Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Ağ oluşturma penceresi, a) gizli katman seçimi ekranı, b) gizli katman nöron sayısı belirleme ekranı.

### 4.2.3. Küçük Dünya ağı tasarımı

Programın bu bölümünde, Küçük Dünya ağlarının model tipi belirlenmekte ve yenilenecek bağlantı sayısı girilebilmektedir. Network Connectivity butonu kullanılarak, Küçük Dünya parametreleri tanımlanmış olan ağın MYBS (Maksimum olası yenilecek bağlantı sayısı) değeri hesaplanmaktadır. MYBS değeri elde edildikten sonra düzenli ağdan (YBS=0) başlayarak, ağa ya yeni bağlantı eklenir yada, ağın var olan bağlantıları birer birer değiştirilerek yeni ağ topolojileri elde edilmektedir. Her yeni topoloji için ağın bağlantı uzunluk katsayıları ( $D_{Global}$  ve  $D_{Lokal}$ ) hesaplanmaktadır. Hesaplanan katsayı değerleri program kök dizininde oluşturulan globloc dizini içerisinde ayrı ayrı dosyalarda tutulmaktadır. Maksimum yenilenecek bağlantı sayısına ulaşıldığında, hesaplama işlemi tamamlanmaktadır. Şekil 4.5'te Küçük Dünya ağı tasarım bölümü gösterilmiştir.



Şekil 4.5. Küçük Dünya ağı tasarımı ekranı.

Yazılımda, karesel bir ağ seçildiğinde, MYBS denklem 3.7 kullanılarak hesaplanmaktadır. Bunun yanında karesel olmayan geleneksel bir ileri yönlü yapay sinir ağı seçilmiş ise MYBS denklem 3.7 ile hesaplanamamaktadır. Bu nedenle değişken katmanlı ve katmanlarında farklı sayıda nörona sahip olan ileri yönlü ağ topolojileri için iki farklı denklem önerilmiştir. Birinci denklemde giriş katmanı ile çıkış katmanı nöronları arasında yeni kısa bağlantı eklenmemiş ve ona göre hesap yapılmıştır. Önerilen denklem aşağıda verilmiştir.

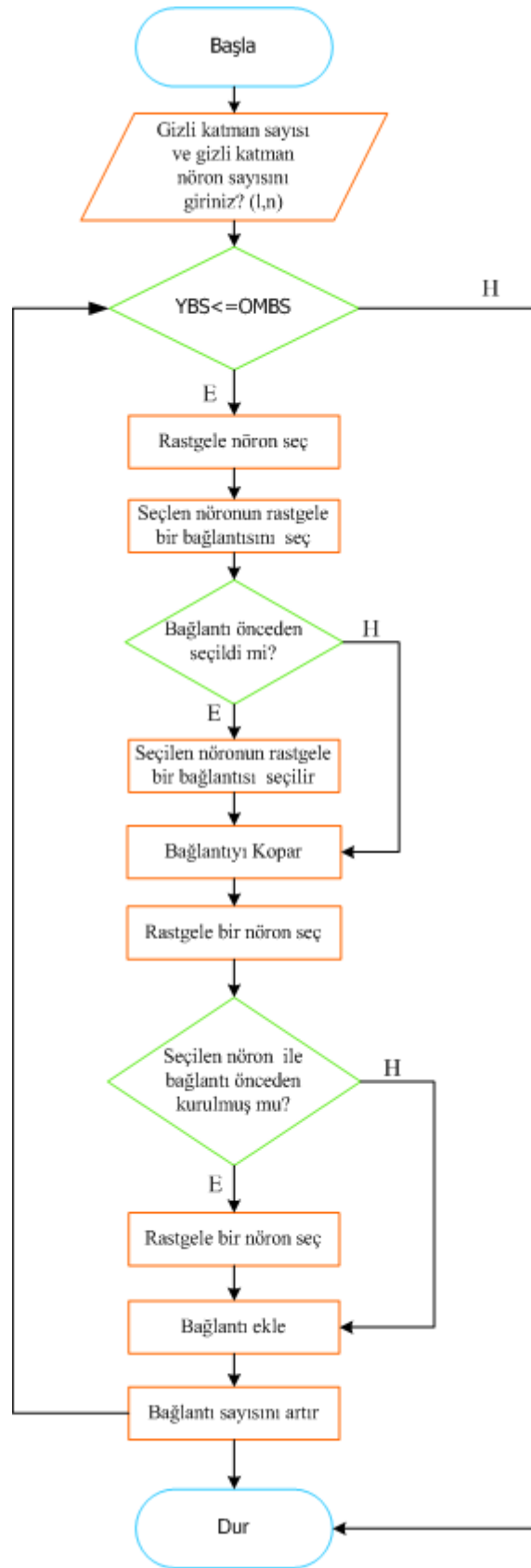
$$MYBS = \left\{ \begin{array}{l} n_1 * n_l, \quad gks = 1 \\ \left[ \sum_{i=1}^{l-2} \sum_{j=i+2}^l (n_i * n_j) \right] - (n_1 * n_l), \quad gks > 1 \end{array} \right\} \quad (4.1)$$

İkinci denklemde, bağlantı yenileme sürecinde giriş katmanı ile çıkış katmanı arasında yeni kısa bağlantı eklenmiştir. Bu nedenle MYBS aşağıdaki gibi hesaplanmaktadır.

$$\text{MYBS} = \left\{ \begin{array}{l} n_1 * n_l, \quad gks = 1 \\ \left[ \sum_{i=1}^{l-2} \sum_{j=i+2}^l (n_i * n_j) \right], \quad gks > 1 \end{array} \right\} \quad (4.2)$$

Denklemlerde (4.1-4.2),  $l$  seçilen ağırlık katman sayısını,  $n_i$   $i$ . katman nöron sayısını,  $n_j$   $j$ . katman nöron sayısını,  $gks$  gizli katman sayısını ve MYBS olası maksimum yenilenecek bağlantı sayısını ifade etmektedir. Her yenilenecek bağlantı için ağırlık bağlantı uzunluk katsayıları hesaplanmaktadır. bağlantı uzunluk katsayıları hesaplanması süreci için uygulanacak algoritmik prosedüre ait akış diyagramı Şekil 4.6' da verilmiştir.





Şekil 4.6. Ağ bağlantı uzunluk katsayılarının hesaplanması için kullanılan akış diyagramı.

#### 4.2.3.1. $D_{Global}$ parametresinin hesaplanması

Programlama sürecinde, Küçük Dünya ağı tasarımının en önemli bölümü global ve lokal bağlantı uzunlu katsayılarının hesaplanması prosedürüdür. Global bağlantı uzunluğu, iki nöron arasındaki bağlantı uzunluğunu tanımlamakta ve nöronlar arası bilgi iletimin durumunu ifade etmektedir. Büyük bağlantı uzunluğu daha yavaş iletim, kısa bağlantı uzunluğu daha hızlı iletimi betimlemektedir. Bu bağlamda ağın global bağlantı uzunluğu hesabı ağdaki bütün nöronların bağlantı uzunluğu ortalamasının tersi ile hesaplanmaktadır. Programda kullanılan  $D_{globhesap}$  fonksiyonuna ait sözde kod Şekil 4.7' de verilmiştir.

```

Function Dglobhesap()
Integer nrons değişkeni tanımla ve değerinin ağın norun sayısına eşitle;
Double dglobal, eglobal değişkenlerini tanımla;
2 boyutlu double tipli dismat, G ve A dizi değişkenlerini nrons boyutlu
tanımla ;
    For i := 0 to nrons-1
        For j := 0 to nrons-1
            if i = j then
                G[i][j] := 0;
            Endif
            G[I][j] := infinity;
        EndFor
    EndFor
    For each link c in con
G[link c kaynak nöron.numara][ link c hedef nöron.numara] := 1;
A[link c kaynak nöron.numara][ link c hedef nöron.numara] := 1;
    EndFor
        For k := 0 to nrons-1
            For i := 0 to nrons-1
                For j := 0 to nrons-1
                    G[i][j] = Min(G[i][j], G[i][k]+ G[k][j]);
                EndFor
            EndFor
        EndFor
        Değeri 0 olan Integer T değişkeni tanımla
    For i := 0 to nrons-1
        For j := 0 to nrons-1
            if G[i][j] = 0 Then
                continue;
            else
                T += 1/ G[i][j];
            Endif
        EndFor
    EndFor
    eglobal = T / ((nrons - 1) * nrons);
    dglobal = 1 / eglobal;
    Return dglobal;
EndDglobhesap

```

Şekil 4.7.  $D_{globhesap}$  fonksiyonuna ait sözde kod bloğu.

#### 4.2.3.2. $D_{\text{lokal}}$ parametresinin hesaplanması

Küçük Dünya ağları için bir diğer önemli katsayı ise lokal bağlantı katsayısıdır.  $D_{\text{lokal}}$ , bir ağda nöronlar arası ilişki kurabilme olasılığını tanımlar ve gerçek dünya ağlarındaki akrabalık veya arkadaşlık ilişkilerinin kurulumuna benzemektedir. Tanımadığınız bir insanı arkadaşınız tanıyor olabilir ve onun üzerinde tanımadıklarınızla ilişki kurabilirsiniz. Bu duruma benzer olarak nöronlar diğer nöronlarla bağlı oldukları nöronlar üzerinden bağ kurarlar. Her bir nöronun lokal bağlantı uzunluk katsayısı, ilgili nöron çıkarıldığında, geriye kalan alt ağda, komşular arasındaki bağlantı sayısının, komşular arasında olası bağlantı sayısına oranının tersi ile hesaplanmaktadır. Ağın lokal bağlantı uzunluk katsayısı ise tüm nöronların lokal bağlantı uzunluklarının ortalaması olarak hesaplanmaktadır. Programda her bir yenilenen bağlantı için lokal bağlantı uzunluğu hesaplanmaktadır. Bu süreç  $D_{\text{lochesap}}$  fonksiyonu ile yapılmaktadır. Bu fonksiyonun sözde kodu Şekil 4.8'de gösterilmiştir.

```

Function Dlochesap()
    Integer nrons değişkeni tanımla ve değerinin ağın nöron
    sayısına eşitle
    Double T,TE,Dlocal ve Elocal değişkenlerini tanımla
    2 boyutlu double tipli cimat,G ve A dizi değişkenlerini
    nrons boyutlu tanımla

    For i := 0 to nrons-1
        For j := 0 to nrons-1
            if i = j then
                G[i][j] := 0;
            Endif
            G[i][j] := infinity;

        EndFor
    EndFor

    For each link c in con
        G[link c kaynak nöron.numara][ link c hedef nöron.numara] := 1;
        A[link c kaynak nöron.numara][ link c hedef nöron.numara] := 1;
    EndFor

```

Şekil 4.8.  $D_{\text{lochesap}}$  fonksiyonuna ait sözde kod bloğu.

```

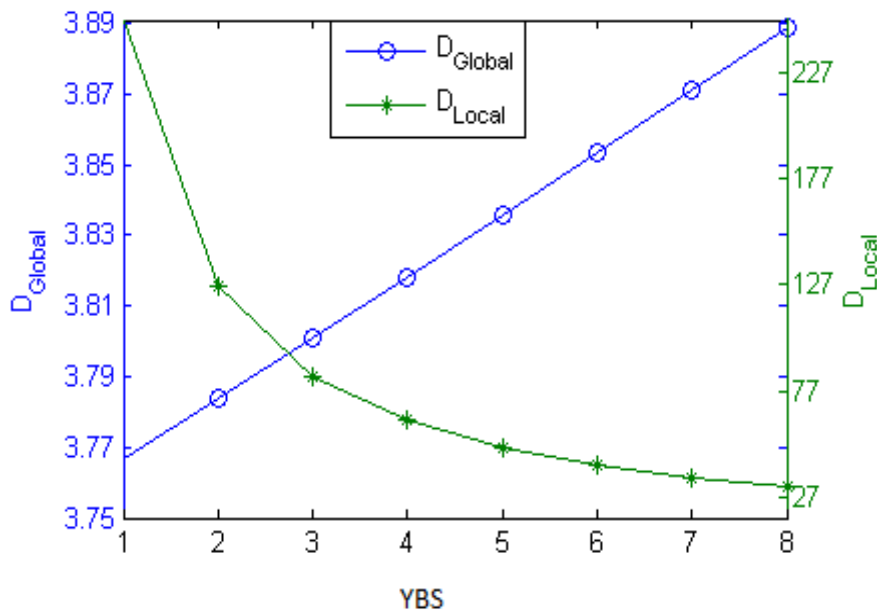
        For k := 0 to nrons-1
            For i := 0 to nrons-1
                For j := 0 to nrons-1
                    G[i][j] = Min(G[i][j], G[i][k]+ G[k][j]);
                EndFor
            EndFor
        EndFor
    EndFor
List tipli bir edd deęişkeni tanımla
Double tipli kki deęişkeni tanımla
TE deęişkenine 0 deęerini at
For k := 0 to nrons-1
    kki = 0;
    For i := 0 to nrons-1
        if G[k][i] = 1 then
            kki deęişkenini 1 artır
            edd listesine i deęişkenini ekle
        Endif
    EndFor //i
    T deęişkenine 0 deęerini at
    For i := 0 to kki-1
        For j := 0 to kki-1
            if G[edd[i]][edd[j]] <> 0 then
                T += 1 / G[edd[i]][edd[j]]
            Endif
        EndFor //j
    EndFor //i
    edd Listesini temizle
    if kki > 1 then
        EG[k] = T / (kki * (kki - 1));
    else
        continue;
    Endif
    TE += EG[k];
EndFor //k
Elocal = TE / nrons;
Dlocal = 1 / Elocal;
Return Dlocal;
EndDlocalhesap

```

Şekil 4.8. (Devam).

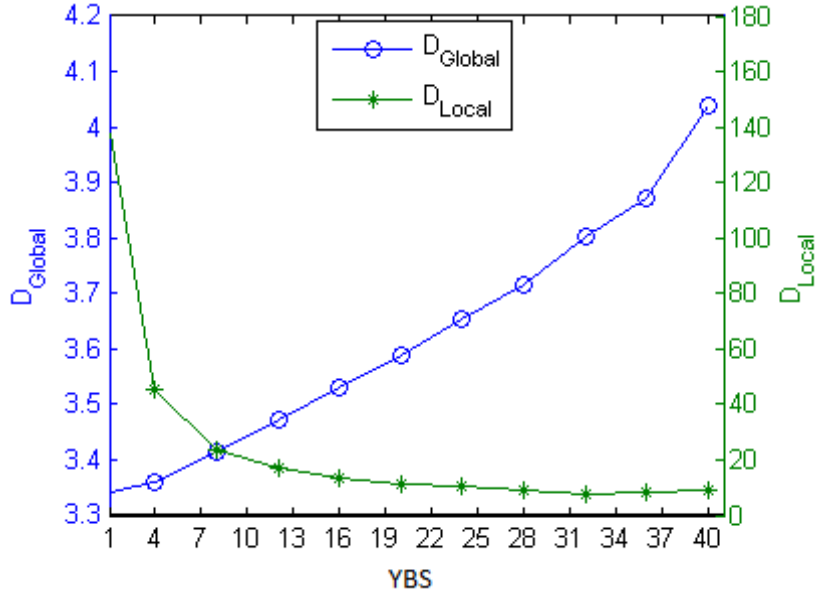
#### 4.2.3.3. $D_{global}$ ve $D_{lokal}$ parametreleri kullanılarak Küçük Dünya ağının tespiti

$D_{global}$  ve  $D_{lokal}$  ağ bağlantı uzunluklarının hesaplanması işlemiyle hazırlanan program 2 adet dosya oluşturmaktadır. Bu dosyalar MATLAB yazılımı kullanılarak aynı grafikte işlenmiş ve her iki bağlantı uzunluk parametresinin de küçük olduğu aralık tespit edilmeye çalışılmıştır [20-23,24]. Denemeler sonucunda her iki uzunluk eğrisinin kesiştiği nokta için bulunan YBS değeri, Küçük Dünya Ağlarını elde edebilmek için minimum ihtiyaç duyulan yenilenecek bağlantı sayısını ifade etmektedir. Burada kesirli sonuçlar aşağıya yuvarlanmıştır. Böylece Küçük Dünya ağları aralığı tespit edilmiş olacaktır. Örnek olarak yazılımdan elde edilen 8-12-1 topolojisine sahip bir YSA için, Küçük Dünya ağları aralığı tespiti aşamasında elde edilen  $D_{global}$  ve  $D_{lokal}$  ağ bağlantı uzunluklarının değişimi Şekil 4.9' da gösterilmiştir.



Şekil 4.9. 8-12-1 topolojisine sahip bir YSA için YBS ile  $D_{global}$  ve  $D_{lokal}$  değişimi.

Şekil 4.9'de görüldüğü gibi Küçük Dünya ağları için gerekli olan minimum YBS değeri 2-8 arasında elde edilmiştir. Bir başka çalışmada 3-8-8-2 topolojili 4 katmanlı bir YSA kullanılmış ve bu ağ için YBS ile elde edilen  $D_{global}$  ve  $D_{lokal}$  değişim grafiği Şekil 4.10' de gösterilmiştir.

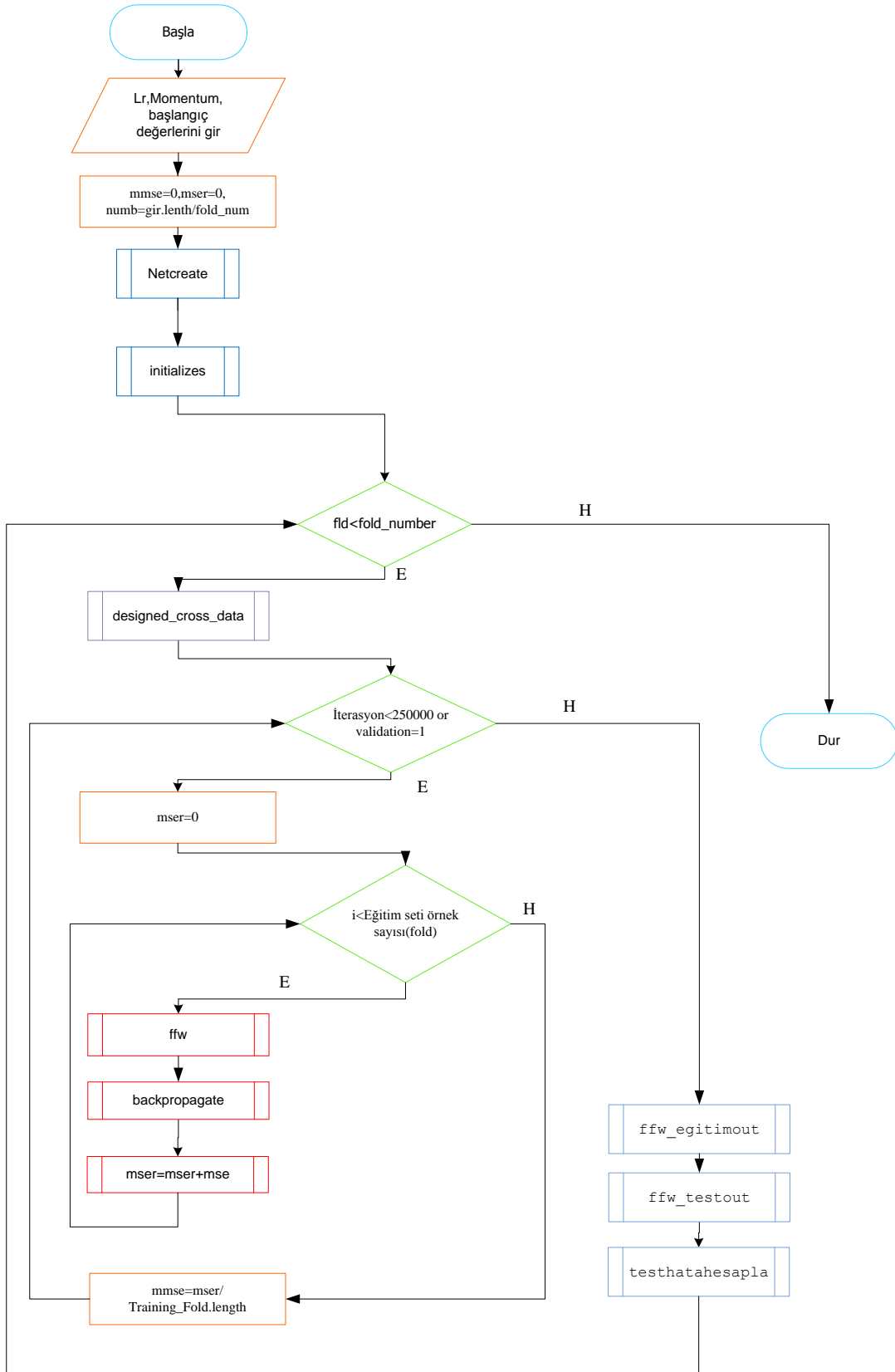


Şekil 4.10. 3-8-8-2 topolojisine sahip bir YSA için YBS ile  $D_{Global}$  ve  $D_{Local}$  değişimi.

Şekilde görüldüğü gibi, Küçük Dünya ağları için gerekli olan minimum YBS değerinin 8-40 arasında olduğu gözlemlenmiştir.

#### 4.2.4. Küçük Dünya ağı YSA algoritması

Küçük Dünya ağı elde edebilmek için gerekli olan YBS değeri tespit edildikten sonra yapay sinir ağlarında çok yaygın kullanılan geri yayılım algoritması hazırlanmış ve farklı YBS değerine sahip ağ topolojilerinin yapay öğrenim performansı ölçülmeye çalışılmaktadır. Yazılım, seçilen topolojileri kullanarak 60 deneme yapmakta ve en başarılı öğrenim denemesini kabul etmektedir. Deneme sayısı kullanıcı tanımlı değiştirilebilmektedir. Hazırlanan algoritmanın öğrenim sürecini tanımlayan kod parçasına ait akış diyagramı Şekil 4.11' de gösterilmiştir.



Şekil 4.11. Öğrenim algoritması akış diyagramı.

Hazırlanan algoritma; `netcreate()`, `initializes()`, `designed_cross_data()`, `cffw()`, `cbackpropagate()`, `cffw_egitimout()`, `cffw_testout()` ve `ctesthatahesapla()` isimli fonksiyonlar kullanılarak oluşturulmuştur.

#### **4.2.4.1. Function netcreate(parametre: YBS):**

Program öncelikle topolojinin oluşturulması üzerine kurulmuştur. Bu bağlamda eğitim sürecinin ilk basamağında ağ topolojisinin oluşturulmaktadır. Tanımlanan fonksiyonda dışarıdan alınan YBS değerine ve seçilen Küçük Dünya ağı modeline göre yeni ağ topolojileri oluşturulmaktadır. Watt-Strogatz Küçük Dünya ağı modeli seçilmiş ise; rastgele bağlı olan 2 nöron seçilir ve aradaki bağlantı koparılır, sonra ilk nöron sabit bırakılıp, ilk nöronun daha önce bağlanmadığı bir nöron rastgele seçilerek koparılan bağlantının yerine yeni bağlantı eklenmektedir. Eğer seçilen nöronla daha önceden bağlantı kurulmuş ise başka bir nöron seçilir. Böylece ağın bağlantı sayısı değiştirilmemektedir. Bu sürece bağlantı yenileme denilmektedir. Newman Watts Küçük Dünya ağı modeli seçilmiş ise; rastgele bağlı olmayan 2 nöron seçilerek yeni bağlantı eklenmektedir. Burada ağın bağlantı sayısı artmaktadır. Bu kurgu bağlı olmayan nöron çifti var olduğu müddetçe devam etmektedir. Bu sürece bağlantı ekleme denilmektedir. Ağ topolojisinin oluşturulmasına ait algoritmanın sözde kod bloğu Şekil 4.12' de gösterilmiştir.



```

Function netcreate(integer rw)
    ycon listesini temizle;
    kbag listesini temizle;
    integer number deęişkenini sıfıra eşitle;
    For i := 0 to katman sayısı-2
    For j := 0 to i. katman nöron sayısı-1;
    i. katman j. nöronunun numarasını number deęişkenine eşitle;
        Vertices listesine (i,j)nöronunu ekle;
        For b := 0 to (i+1). Katmanın nöron sayısı;
        (i,j) nöronunun outdent listesine, (i+1,b) nöronunu ekle;
        (i,j) nöronunu (i+1,b) nöronuna bağla ve con listesine ekle ;
        EndFor
            number deęişkenini 1 artır.
        EndFor
    EndFor
EndFor

    For i := katman sayısı-1 to katman sayısı-1
        For j := 0 to i. katman nöron sayısı-1;
            (i,j) nöronunun numarasını number deęişkenine eşitle;
            Vertices listesine (i,j)nöronunu ekle;
            number deęişkenini 1 artır.
        EndFor
    EndFor

    Integer nsayı deęişkenini number deęişkenine eşitle

    For i := katman sayısı - 1 to 1
    For j := 0 to i. katman nöron sayısı-1
        For b := 0 to (i-1). katman nöron sayısı-1;
        (i,j)nöronunun indent listesine, (i-1,b) nöronunu ekle;
        EndFor
        EndFor
    EndFor

    Dosyan isimli bir dosya deęişkeni olustur.

    bag class tipinde nor deęişkenini oluşturun;
    integer kl, kn, kb, yl, yn, ind, indek isimli deęişkenleri oluşturun ve 0'a
    eşitle;

```

Şekil 4.12. netcreate fonksiyonuna ait sözde kod bloęu.

```

    For x := 0 to rw //YBS sayısı
      Restgele (0,Layer-2) aralığında sayı üret kl'e eşitle;
    2 ms gecikme uygula;
    Restgele (0,noron) aralığında sayı üret kn'e eşitle;
      2 ms gecikme uygula;
      Restgele (0,outdent) aralığında sayı üret kb'e eşitle;
    2 ms gecikme uygula;
    Restgele (kl+2,Layer) aralığında sayı üret yl'e eşitle;
      2 ms gecikme uygula;
      Restgele (0,noron) aralığında sayı üret kn'e eşitle;
      Restgele (0,yl.noron) aralığında sayı üret yn'e eşitle;

while searchconnect(kl, kn, yl, yn) eşit 1 or (kl,kn).outdent[kb].durum
eşit 1
      Restgele (0,Layer-2) aralığında sayı üret kl'e eşitle;
      2 ms gecikme uygula;
    Restgele (0,noron) aralığında sayı üret kn'e eşitle;
      2 ms gecikme uygula;
      Restgele (0,outdent) aralığında sayı üret kb'e eşitle;
    2 ms gecikme uygula;
    Restgele (kl+2,Layer) aralığında sayı üret yl'e eşitle;
      2 ms gecikme uygula;
    Restgele (0,noron) aralığında sayı üret kn'e eşitle;
      Restgele (0,yl.noron) aralığında sayı üret yn'e eşitle;

EndWhile
nor değişkenini (kl,kn). nöron kb. output dentritinin nöronunu ata ;
kcon listesine (kl,kn). nöron ile nor içerisindeki nöron arasındaki
bağlantıyı ekle;
(kl,kn). nöron ile nor nöronu arasındaki bağlantıyı sil ;

nor nöronunun indent listesinden (kl,kn). nöronu sil;
nor nöronunun oid değişkenini 1 azalt;
nor nöronunun dentw liste boyutunu, nor.oid sayısı kadar ayarla;
nor nöronunun prew liste boyutunu, nor.oid sayısı kadar ayarla;
(kl,kn). nöronun kb. outdent layer değerini yl'ye eşitle;
(kl,kn). nöronun kb. outdent nöron değerini yn'ye eşitle;
(kl,kn). nöronun kb. outdent durum değerini 1'e eşitle

(yl,yn). nöronunun inden listesine (kl,kn). nöronunu ekle;
(yl,yn). nöronunun oid değerini 1 artır;

```

Şekil 4.12. (Devam).

```

(yl,yn). nöronunun dentw liste boyutunu (yl,yn). nöron oid değeri kadar
ayarla;
(yl,yn). nöronunun prew liste boyutunu (yl,yn). nöron oid değeri kadar
ayarla;
    con listesine (kl,kn,yl,yn) bağlantısını ekle;
    ycon listesine (kl,kn,yl,yn) bağlantısını ekle;
    kbag listesine (kl,kn).kb bağlantısını ekle;
dosyan dosya değişkenine (kl,kn). nöron ile (yl,yn). nöron arasındaki
bağı yaz;
        EndFor
        dosyan dosya değişkenini kapat;
Endnetcreate

```

Şekil 4.12. (Devam).

#### 4.2.4.2. Function initializes():

Öğrenme sürecinin başlangıcında ağ topolojisinin başlangıç ağırlıklarının belirlenmesi süreci initializes fonksiyonu ile tanımlanmıştır. Başlangıç ağırlıkları, uygulamada standart olarak [-1 1] aralığında rastgele atanmaktadır. Ağırlıkların ayarlandığı initializes() fonksiyonuna ait sözde kod bloğu Şekil 4.13' te gösterilmiştir.

```

Function initializes()
For i := 1 to katman sayısı-1
    For j := 0 to i. nöron sayısı-1
        (i,j). nöron biası r.NextDouble() * 2) - 1 değerine eşittir;
        (i,j). nöron önceki bias 0.0 değerine eşittir;
        (i,j). nöron bias delta değeri 0.0 değerine eşittir;
        For x := 0 to (i,j). nöron input dentrit sayısı-1;
            20 ms gecikme uygula;
        (i,j). nöronun x. ağırlığı (r.NextDouble()*2 - 1 değerine eşittir;
        (i,j). nöronun x. önceki ağırlığı 0'a eşittir;
        (i,j). nöronun x. ağırlığın delta değeri 0'a eşittir;

        EndFor

    EndFor

EndFor
Endinitializes;

```

Şekil 4.13. initializes fonksiyonuna ait sözde kod bloğu.

#### 4.2.4.3. Function designed\_cross\_data():

YSA yazılımında Çapraz Doğrulama işleminin gerçekleşmesi için veri setinin dışarıdan girilen kat sayısına göre parçalanması gerekmektedir. Çalışmada N Kat Çapraz Doğrulama algoritması hazırlanmıştır. Bu algoritmada eğitim setini N parçaya bölünmektedir. Bölünen parçalardan, 1 parça test için geriye kalan N-1 parça ise eğitim için kullanılmaktadır. Bu süreç N defa tekrar etmektedir. Böylece veri setinin tamamı eğitim ve test sürecine dahil olmaktadır. Böylece eğitilen ağ için ezberleme olayının önüne geçilmiştir[62]. Yazılım başlangıç koşulunda kat sayısı 10 olarak atanmakta kullanıcı isterse kat sayısını değiştirebilmektedir. Veri setinin parçalanması sürecini gerçekleştiren designed\_cross\_data() fonksiyonuna ait sözde kod Şekil 4.14' te gösterilmiştir.

```
Function designed_cross_data(integer f, integer fnum)

integer tipli numb değişkeni tanımla;
numb değişkenine (int)gir.GetLength(0) / fnum işlem sonucunu ata;
double tipli, 2 boyutlu ftestc, ftestg, nftestc, nftestg değişkenlerini
[numb, cik.GetLength(1)] boyutlu olarak tanımla;

double tipli, 2 boyutlu ftrainc, ftraing, nftrainc, nftraing
değişkenlerini [(gir.GetLength(0) - numb), gir.GetLength(1)] boyutlu
olarak tanımla;

if f==0 then

    For i := 1 to numb -1
        For j := 0 to gir.GetLength(1)-1
            ftestg[i, j] dizisini gir[i, j] dizisine eşitle;
            nftestg[i, j] dizisini ngir[i, j] dizisine eşitle;
            if j < cik.GetLength(1) then
                ftestc[i, j] dizisini cik[i, j] dizisine eşitle;
                nftestc[i, j] dizisini ncik[i, j] dizisine eşitle;
            Endif
        EndFor
    EndFor
EndFor
```

Şekil 4.14. designed\_cross\_data fonksiyonuna ait sözde kod bloğu.

```

For i := numb to gir.GetLength(0 -1
  For j := 0 to gir.GetLength(1)-1
    ftraing[i - numb, j] dizisini gir[i, j] dizisine eşitle;
    nftraing[i - numb, j] dizisini ngir[i, j] dizisine eşitle;
    if j < cik.GetLength(1) then
      ftrainc[i - numb, j] dizisini cik[i, j] dizisine eşitle;
      nftrainc[i - numb, j] dizisini ncik[i, j] dizisine eşitle;
    Endif
  EndFor
EndFor
Endif
if f > 0 and f < fnum - 1 then
  For i := 1 to f * numb-1
    For j := 0 to gir.GetLength(1)-1
      ftraing[i, j] dizisini gir[i, j] dizisine eşitle;
      if j < cik.GetLength(1) then
        ftrainc[i, j] dizisini cik[i, j] dizisine eşitle;
        nftrainc[i, j] dizisini ncik[i, j] dizisine eşitle;
      Endif
      nftraing[i, j] dizisini ngir[i, j] dizisine eşitle;
    EndFor
  EndFor
  integer tipli kalan değişkeni tanımla ve i değerine eşitle;
  For i := kalan to ((kalan) + numb-1)
    For j := 0 to gir.GetLength(1)-1
      ftestg[i - kalan, j] dizisini gir[i, j] dizisine eşitle;
      if j < cik.GetLength(1) then
        ftestc[i - kalan, j] dizisini cik[i, j] dizisine eşitle;
        nftestc[i - kalan, j] dizisini ncik[i, j] dizisine eşitle;
      Endif
      nftestg[i - kalan, j] dizisini ngir[i, j] dizisine eşitle;
    EndFor
  EndFor
  kalan değişkenini i değerine eşitle;
  For i := kalan to gir.GetLength(0)-1
    For j := 0 to gir.GetLength(1)-1
      ftraing[i - numb, j] dizisini gir[i, j] dizisine eşitle;
      if j < cik.GetLength(1) then
        ftrainc[i - numb, j] dizisini cik[i, j] dizisine eşitle;
        nftrainc[i - numb, j] dizisini ncik[i, j] dizisine eşitle;
      Endif
    Endif
  Endif

```

Şekil 4.14. (Devam).

```

        nftraing[i - numb, j] dizisini ngir[i, j] dizisine eşitle;
        EndFor
    EndFor
Endif
if f == fnum - 1 then

    For i := 1 to (gir.GetLength(0) - numb) -1
        For j := 0 to gir.GetLength(1)-1
            ftraing[i, j] dizisini gir[i, j] dizisine eşitle;
            if j < cik.GetLength(1) then
                ftrainc[i, j] dizisini cik[i, j] dizisine eşitle;
                nftrainc[i, j] dizisini ncik[i, j] dizisine eşitle;
            Endif
            nftraing[i, j] dizisini ngir[i, j] dizisine eşitle;
        EndFor
    EndFor
    kalan değişkenini i değerine eşitle;
    For i := kalan to gir.GetLength(0)-1
        For j := 0 to gir.GetLength(1)-1
            ftestg[i - kalan, j] dizisini gir[i, j] dizisine eşitle;
            if j < cik.GetLength(1) then
                ftestc[i - kalan, j] dizisini cik[i, j] dizisine eşitle;
                nftestc[i - kalan, j] dizisini ncik[i, j] dizisine eşitle;
            Endif
            nftestg[i - kalan, j] dizisini ngir[i, j] dizisine eşitle;
        EndFor
    EndFor
Endif
Enddesigned_cross_data

```

Şekil 4.14. (Devam).

#### 4.2.4.4. Function cffw():

Ağın eğitimi sürecinde verilen giriş örneklerine karşılık çıkışın hesaplanması gerekmektedir. Bu nedenle çıkış hazırlanan kod içerisinde ileri yönlü olarak giriş katmanından çıkış katmanına doğru denklem 2.1 kullanılarak katmanlar arasında hesaplanmaktadır. Fonksiyon dışarıdan örnek numarası ve ilgili çapraz doğrulama bölümü için çıkış dizisini almaktadır. Fonksiyon sonucunda elde edilen çıkış değerleri ağın öğrenme hatasının hesaplanması için kullanılmaktadır. İleri yönlü

hesaplama işlemini gerçekleyen cffw fonksiyonuna ait sözde kod bloğu Şekil 4.15' te gösterilmiştir.

```

Function cffw(integer v,double[, ] ndata)
    double ssum değişkeni tanımla;
    For n := 0 to ilk katman nöron sayısı-1
        (0,n). nöron output değerini ndata[v, n] değerine eşitle;
    EndFor
    For l := 1 to ağ katman sayısı-1
        For n := 0 to l. katman nöron sayısı-1
            Ssum değişkenini 0'a eşitle;
            (l,n). nöron çıkışını 0'a eşitle;
            For k := 0 (i,j). nöron input dentrit sayısı-1
                ssum değerine [(l,n). nöron k. input dentritinin nöron çıkışı * (l,n).
                nöron k. ağırlık] işlem sonucu ekle ;
            EndFor
            ssum değerine (l,n). nöron biasını ekle
            activation(ssum,sec) fonksiyonunu çağır ve sonucu (l,n). nöron output
            değerine ata;
        EndFor
    EndFor
Endcffw

```

Şekil 4.15. cffw fonksiyonuna ait sözde kod bloğu.

#### 4.2.4.5. Function cbackpropagate():

Çıkış hatasının hesaplanması işlemi sonrası hesaplanan hata kabul edilebilir hatadan daha büyükse ve bir önceki hatadan küçükse hesaplanan hata geriye doğru aktarılarak azaltılmakta ve ağırlıklar güncellenmektedir. Bu süreçte denklemler 2.12-17 kullanılarak geri yayılım algoritması geliştirilmiştir. Bu bağlamda elde edilen geriye yayılım algoritmasının sözde kod bloğu Şekil 4.16' da gösterilmiştir.

```

Function cbackpropagate(integer r, double[,] ncikis)
    Double tipli sum deęişkeni tanımla;
    For j := 0 to son katman nöron sayısı-1
    Son katman j. noron hata deęerini [ncikis[r, j] - Son katman j. nöron
output] ifadesine eşitle;
    Son katman j. nöron delta deęerini [deriv(Son katman j. nöron output )
* Son katman j. noron hata] ifadesine eşitle ;
    EndFor
    For i := katman sayısı-2 to 1
        For j := 0 to i. katman nöron sayısı-1
            sum deęişkenini 0'a eşitle;
            for k := 0 to (i,j). nöron outdent saysısı -1
                (i,j). nöron k. outdent deltası ile (i,j). nöron k. outdent detw
ağırlık deęerini .arp sum deęişkenine ekle;
            EndFor
            (i,j). nöron deltasını, (i,j). nöron output ile sum deęikeninin
çarpımına eşitle;
            EndFor
        EndFor
    EndFor
    For i :=1 to katman sayıs-1
        For j := 0 to i. katman nöron sayısı-1
            For k := 0 to (i,j). nöron indent saysısı -1
                momentum deęerini (i,j). nöron k. prew ağırlığı ile çarp, (i,j). nöron k.
detw ağırlığına ekle;
            EndFor
            momentum deęerini (i,j). nöron prewbias deęeri ile çarp, (i,j). nöronun
biasına ekle;
            EndFor
        EndFor
        For i :=1 to katman sayıs-1
            For j := 0 to i. katman nöron sayısı-1
                For k := 0 to (i,j). nöron indent saysısı -1
                    (i,j). nöron k. prew deęerini, [Lr* (i,j). nöron delta * (i,j). nörona
baęlı k. indet nöronu output deęeri] ifadesine eşitle;
                    (i,j). nöron k. detw deęerine (i,j). nöron k. prew deęerini ekle;
                EndFor
                (i,j). nöron prewbias deęerini, Lr * (i,j). nöron delta ifadesine eşitle;
                (i,j). nöron bias deęerine (i,j). nöron prewbias deęerini ekle;
            EndFor
        EndFor
    Endcbackpropagate

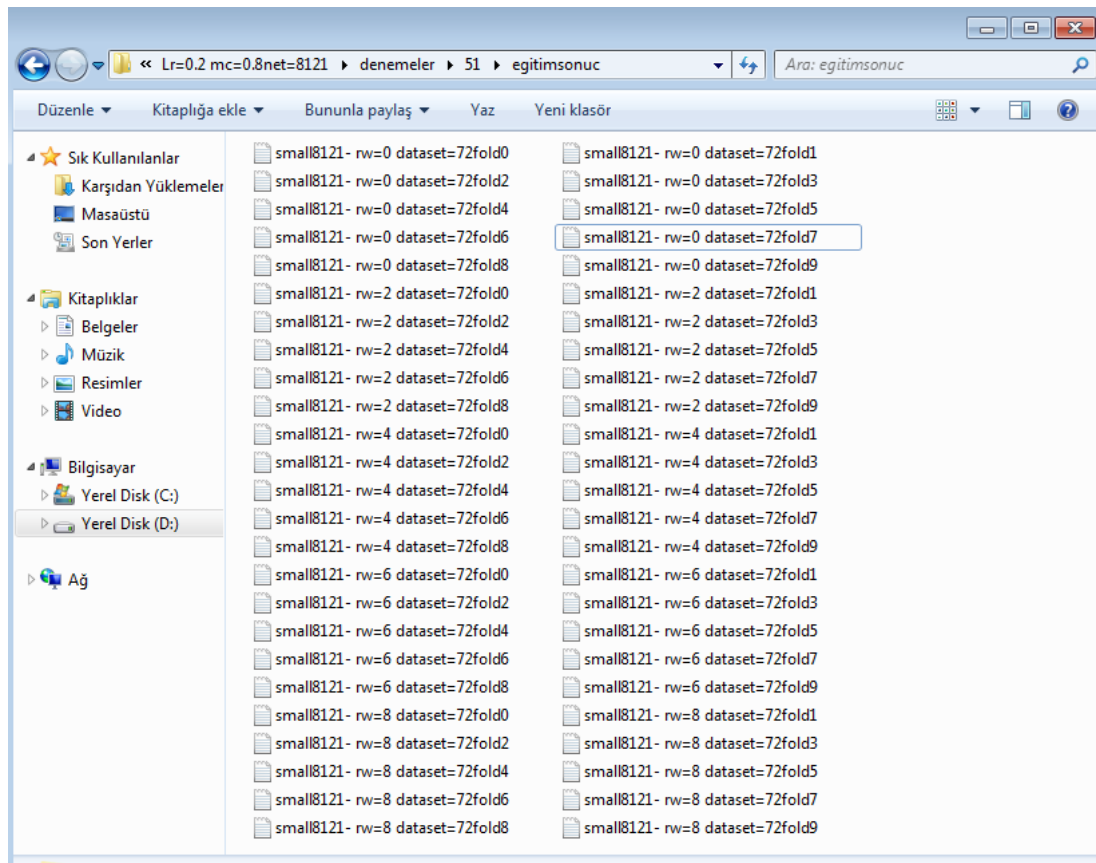
```

Şekil 4.16. cbackpropagate fonksiyonuna ait sözde kod bloęu.



#### 4.2.4.6. Function `cffw_egitimout()`:

Eğitim sürecini tamamlayan ağ topolojisinin örnek girdilere karşılık hesapladığı çıktı değerlerinin elde edilmesi amacıyla `cffw_egitimout` fonksiyonu hazırlanmıştır. Bu fonksiyon dışarıdan 3 değişken almakta girdi dizisi, çıktı dizisi ve örneğe ait çapraz doğrulama kat numarası almaktadır. Fonksiyon çalıştırıldığında, YBS değeri ile elde edilen ağın seçilmiş olan doğrulama parçası için eğitim hatasını hesaplamakta ve hesaplanan hata `egitimsonuc` dizisinde ağ parametreleri ile oluşturulan bir txt uzantılı dosya oluşturup içerisine beklenen ve hesaplanan ağ çıktılarını yazmaktadır. Şekil 4.17' de bir deneme için elde edilen `egitimsonuc` dizisinin içeriği gösterilmektedir.

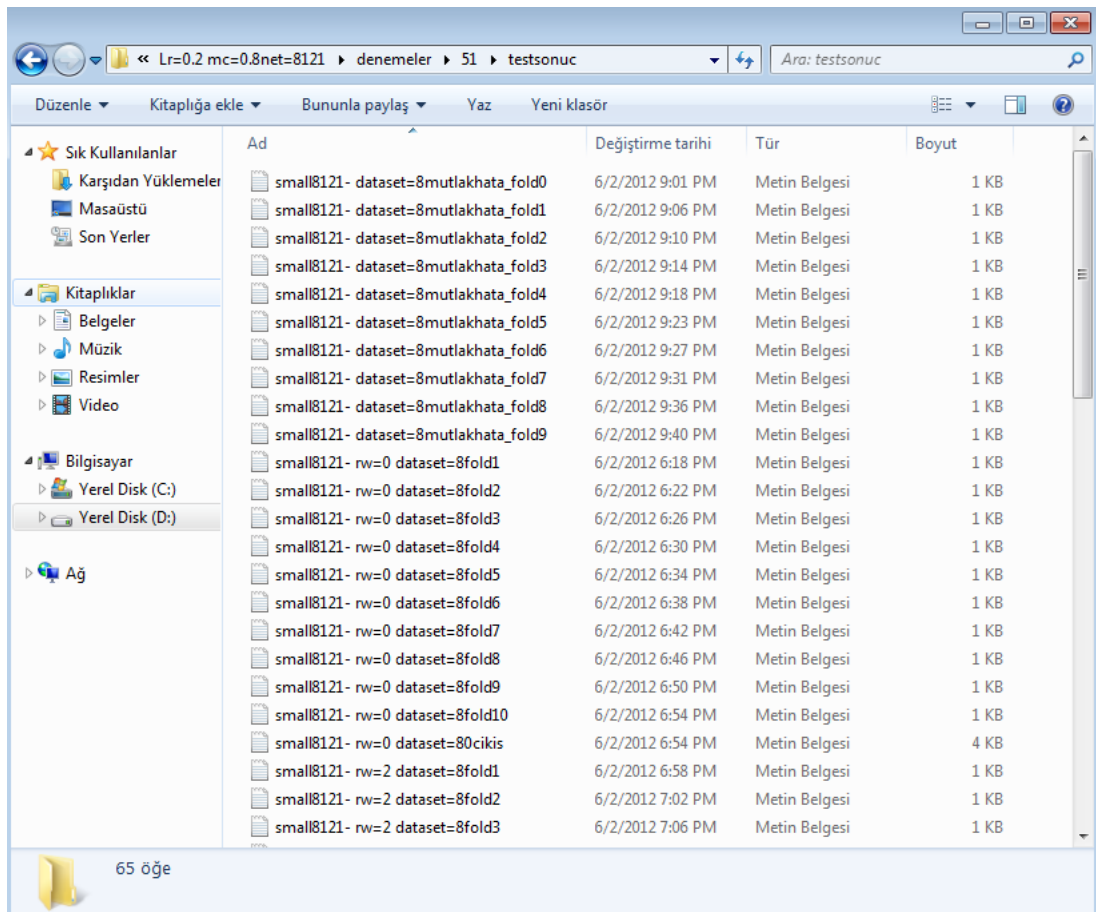


Şekil 4.17. `cffw_egitimout` fonksiyonun oluşturduğu `egitimsonuc` dizini ve içeriği.

#### 4.2.4.7. Function `cffw_testout()`:

Ağın eğitiminin doğruluğunun fiziksel kanıtı test çıktılarının beklenen değerlerle belirli ölçülerde örtüşmesidir. Bu bağlamda hazırlanan `cffw_testout` fonksiyon algoritmasında Çapraz Doğrulama fazının tamamlanmasıyla ağın girdi setine

karşılık, hesaplanan çıkış değerleri alınmaktadır. Elde edilen çıkış değeri her bir deneme için oluşturulan testsonuc dizini içerisine kaydedilmektedir. Burada her bir kat test verisi için çıkış değerleri alınabildiği gibi tüm veri setine karşılık elde edilen çıkış değeri de alınabilmektedir. Fonksiyon kat ismi ve topoloji bilgilerini kullanarak dosyalar oluşturularak beklenen ve hesaplanan çıkışları içerisine yazmaktadır. Sonuç olarak, tüm veri setine karşılık elde edilen dosyadaki çıktı değerleri MATLAB 2008a programında grafik çıktılarına dönüştürülmektedir. Şekil 4.18’ de bir denemeye ait testsonuc dizini ve içeriği gösterilmiştir.



Şekil 4.18. cffw\_testout fonksiyonun oluşturduğu testsonuc dizini ve içeriği.

#### 4.2.4.8. Function ctesthatahesapla():

Burada ağın doğruluk testi yapılmaktadır. Testi gerçekleştirmek amacıyla istatistiksel doğruluk(performans) kriterleri kullanılmıştır. Bu performans kriterleri bölüm 2.7’de ifade edilmiştir. Kullanılan doğruluk kriterleri, Ortalama Karesel Hata(MSE),

Ortalama Mutlak Hata(MAE), Kök Ortalama Karesel Hata(RMSE) ve Belirlilik Katsayısı( $R^2$ ) algoritmik olarak bu bölümde ifade edilmiştir. Algoritma her bir çapraz doğrulama parçacığı için bu kriterleri hesaplamakta ve çıktıları önceden oluşturulan testsonuc dizinine kayıt etmektedir. Çapraz Doğrulama sürecinde her kat parçacığı için elde edilen hata kriterlerinin ortalaması alınarak ağın ortalama performans kriterleri ele edilmektedir. ctesthatahesapla fonksiyonuna ait algoritmanın C# kodu Şekil 4.19’ da gösterilmiştir.

```

void ctesthatahesapla(double[,] ncikis, double[,] cikis, int f)
{
    int say = 0;
    double errt = 0, ermse = 0, abdiff = 0, diff = 0, mret = 0, kare = 0,
    sse = 0, rkare = 0, ara = 0, rmse = 0, mre = 0

    for (int nf = 0; nf < ncikis.GetLength(0); nf++)
    {
        for (int n = 0; n < networks.layers[L- 1].noron.Length; n++)
        {
            abdiff = Math.Abs(cikis[nf, n] - double.Parse(container[nf]));
            diff += Math.Pow(cikis[nf, n] - double.Parse(container[nf]), 2);
            errt += Math.Abs(cikis[nf, n] - double.Parse(container[nf]));
            mret += abdiff / cikis[nf, n];
            kare += Math.Pow(double.Parse(container[nf]), 2);
            if (Math.Round(double.Parse(container[nf]), 0) != ncikis[nf, n])
                say++;
        }
    }
    sse = diff;
    rkare = 1 - (sse / kare);
    ara = sse / cikis.GetLength(0);
    rmse = Math.Pow(ara, 0.5) * 100
    mre = 100 * mret / cikis.GetLength(0)
    ermse = diff / 2;

    abs.Text=Convert.ToString(errt/(ncikis.GetLength(0)*
    networks.layers[networks.layers.Length - 1].noron.Length));
    kattest+=(errt/(ncikis.GetLength(0)*networks.layers[networks. layers.Length
    - 1].noron.Length));
}

```

Şekil 4.19. ctesthatahesapla fonksiyonuna ait C# kod bloğu.

```

string dizad = dosyaadiana + "\\testsonuc\\";
FileStream dosyas1 = new FileStream(dizad + "\\" + "small" + dosyaadi + "-
dataset=" + ncikis.GetLength(0).ToString() + "mutlakhata_kat" +
f.ToString() + ".txt", FileMode.Append, FileAccess.Write);

    StreamWriter yazz1 = new StreamWriter(dosyas1);
    yazz1.Write(rewir.ToString() + "          ");
    yazz1.Write(rkare.ToString() + "         ");//rkare
    yazz1.Write(abs.Text + "         ");//mae
    yazz1.Write(ermse.ToString() + "         ");//mse
    yazz1.Write(rmse.ToString() + "         ");//rmse
    yazz1.Flush();
    dosyas1.Close();

}

    FileStream dosyas1 = new FileStream(dizad + "\\" + "small" +
dosyaadi + "- dataset=" + ncikis.GetLength(0).ToString() +
"mutlakhata_kat" + f.ToString() + ".txt", FileMode.Append,
FileAccess.Write);

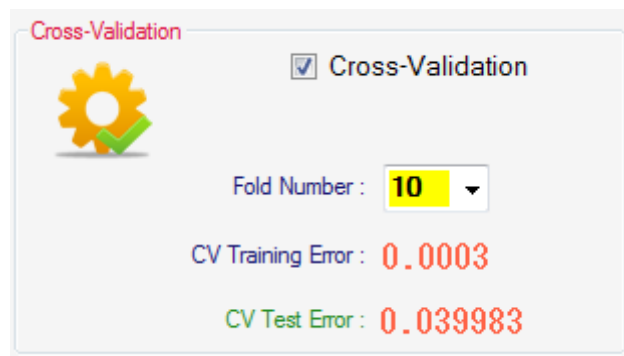
    StreamWriter yazz1 = new StreamWriter(dosyas1);
    yazz1.Write(rewir.ToString() + "          ");
    yazz1.Write(rkare.ToString() + "         ");//rkare
    yazz1.Write(abs.Text + "         ");//mae
    yazz1.Write(ermse.ToString() + "         ");//mse
    yazz1.Write(rmse.ToString() + "         ");//rmse
    yazz1.Flush();
    dosyas1.Close();

}

```

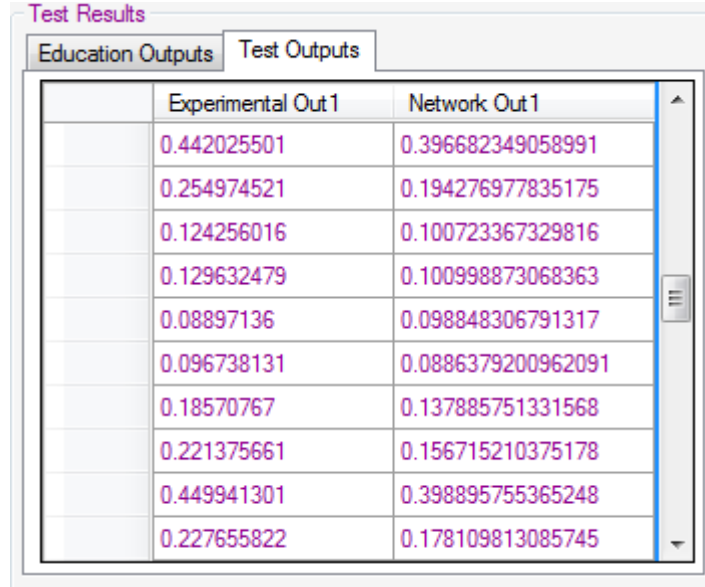
Şekil 4.19. (Devam).

Çapraz Doğrulama süreci sonrası, hazırlanan yazılımda ağın ortalama mutlak hatası(MAE) label içerisinde gösterilmiştir. Yazılımın Çapraz Doğrulama bölümü Şekil 4.20' de gösterilmiştir.



Şekil 4.20. Çapraz doğrulama bölümü.

Ayrıca tüm modelleme süresi sonucunda ağ çıktıları ve beklenen çıktılar Test Results bölümünde kullanıcıya veri ızgarası içinde sunulmaktadır. Eğitim sonrası ağ çıktı veri ızgarası Şekil 4.21’ de gösterilmiştir.



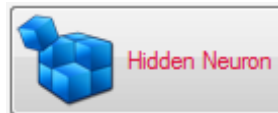
The screenshot shows a window titled "Test Results" with two tabs: "Education Outputs" and "Test Outputs". The "Test Outputs" tab is active, displaying a table with two columns: "Experimental Out 1" and "Network Out 1". The table contains 10 rows of numerical data.

	Experimental Out 1	Network Out 1
	0.442025501	0.396682349058991
	0.254974521	0.194276977835175
	0.124256016	0.100723367329816
	0.129632479	0.100998873068363
	0.08897136	0.098848306791317
	0.096738131	0.0886379200962091
	0.18570767	0.137885751331568
	0.221375661	0.156715210375178
	0.449941301	0.398895755365248
	0.227655822	0.178109813085745

Şekil 4.21. Eğitim sonucu çıktı veri ızgarası görünümü.

#### 4.2.5. Ara katman sayısının hesaplanması

YSA uygulamalarında en önemli problemlerden biriside optimum ara katman nöron sayısının bulunmasıdır. Bu bağlamda hazırlanan yazılımda, 3 katmanlı bir YSA uygulamasında yaygın kullanılan ağ mimarisi temel alınmıştır. Bu bölüm bir ara katmanlı ağ topolojisinin nöron sayısını algoritmik olarak 10 deneme yaparak hesaplamaktadır. Bu deneme sayısı uygulama hassasiyetine göre değiştirilebilmektedir. Ara katman nöron sayısını bulmak amacıyla program içerisinde kullanılan buton Şekil 4.22’de gösterilmiştir.

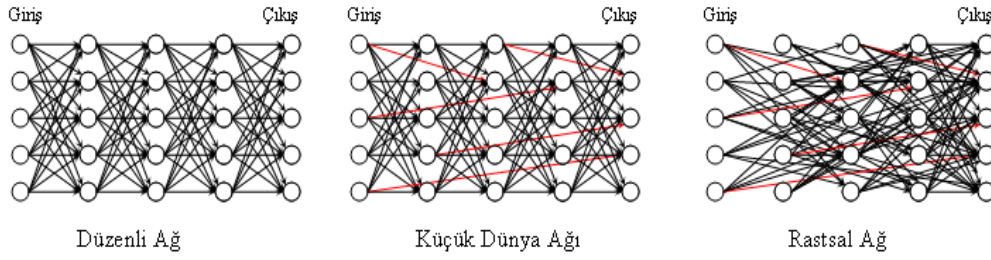


Şekil 4.22. Ara katman nöron sayısı hesaplama butonu.

## **BÖLÜM 5. İLERİ YÖNLÜ YAPAY SINIR AĞLARINDA KÜÇÜK DÜNYA AĞI UYGULAMALARI**

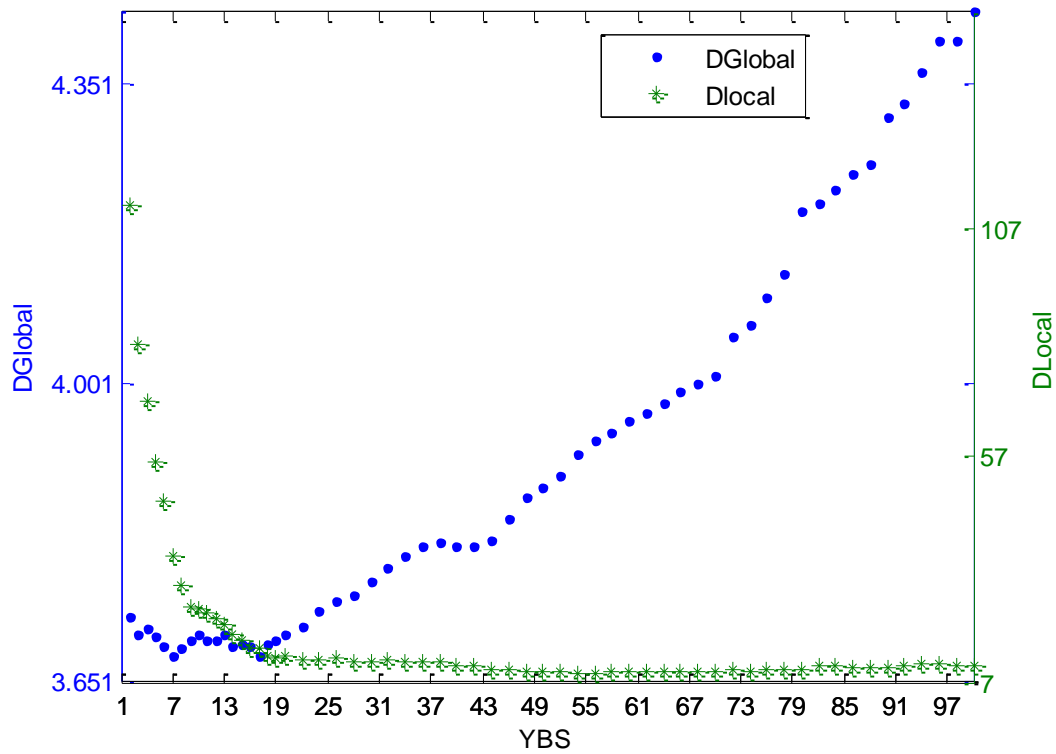
### **5.1. Bağımsız Bağlantı Yenileme Metodu ile Oluşturulan Küçük Dünya Ağının Öğrenme Performansının Araştırılması**

Bu bölümde, bağımsız bağlantı yenileme metodu kullanılarak ileri yönlü küçük dünya ağlarının öğrenme performansı araştırılmıştır. Eğitim seti olarak, 5 girişli ve 5 çıkışlı, rastgele elde edilen, dijital verilerden oluşan bir veri seti kullanılmıştır. Ağ topolojisi olarak karesel ağ topolojileri tercih edilmiştir. Bunlar; 5x5, 5x7, 10x5,10x10,15x8 ve 15x15 olarak belirlenmiştir. İleri yönlü yapay sinir ağları topolojisi düzenli ağ olarak kabul edilmiş, rastgele seçilen iki nöronun bağlantısı kopartılarak yine rastgele seçilen ve bağlantısı olmayan iki nöron arasına bağlantı eklenmiştir. Böylece ağın bağlantı sayısı sabit tutulmuştur. Uygulanan bu metoda bağımsız bağlantı yenileme metodu ismi verilmiştir. Düzenli ağın tüm bağlantılarının değiştirilmesiyle elde edilen topoloji rastsal ağ olarak kabul edilmiştir. Her bağlantı yenilendiğinde 3. Bölüm’de ifade edilen global ve lokal bağlantı uzunluk katsayıları hesaplanmıştır. Her iki katsayının da küçük olduğu YBS sayısına sahip topoloji, Küçük Dünya ağı kabul edilmiştir. Ayrıca, bu katsayılar yardımıyla çizilen grafik incelendiğinde her iki katsayının küçük olduğu değer aralıklarından Küçük Dünya Ağı YSB aralığı tespit edilmiştir. Topolojileri oluşturmak amacıyla, katman sayısı 5-15 arasında değişen bir yapay sinir ağı kullanılmıştır. Örnek olarak, yapılan çalışmada, 5 katmanlı ve her katmanda 5 nöron bulunan yapay sinir ağı modeli için sırasıyla, düzenli, Küçük Dünya ve rastsal ağ topolojileri Şekil 5.1.’de gösterilmiştir.



Şekil 5.1. 5x5 topolojiye sahip ileri yönlü yapay sinir ağı için, düzenli, rastsal ve Küçük Dünya Ağları topolojik gösterimi.

5 katmanlı ve her katmanında 5 nöron bulunan (5x5) topolojiye ait, bağlantı yenileme işlemi sonrasında elde edilen global ve lokal bağlantı uzunluk katsayılarının YBS ile değişimi Şekil 5.2' de gösterilmiştir.



Şekil 5.2. 5x5 İleri yönlü yapay sinir ağı için, bağımsız yenileme metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Local}$  katsayılarının değişimi.

Şekil 5.2' de görüldüğü gibi, aynı katman nöronları arasında bağlantı olmaması nedeniyle, lokal parametre  $\infty$  hesaplanmaktadır. Bu nedenle YBS=0 için global ve lokal parametre Şekilde ifade edilmemiştir. Her iki parametrenin de kesişim noktası küçük kabul edildiğinde, 5x5 ileri yönlü yapay sinir ağında Küçük Dünya ağı için

gerekli minimum YBS sayısı  $\approx 18$  olarak elde edilmiştir. Şekilden bu topoloji için Küçük Dünya ağı aralığı 13-25 arasında elde edilmiştir.

Ayrıca, farklı katman ve nöron sayısına sahip ileri yönlü yapay sinir ağları için gerekli minimum YBS değer aralığı elde edilmiş ve Tablo 1’de gösterilmiştir.

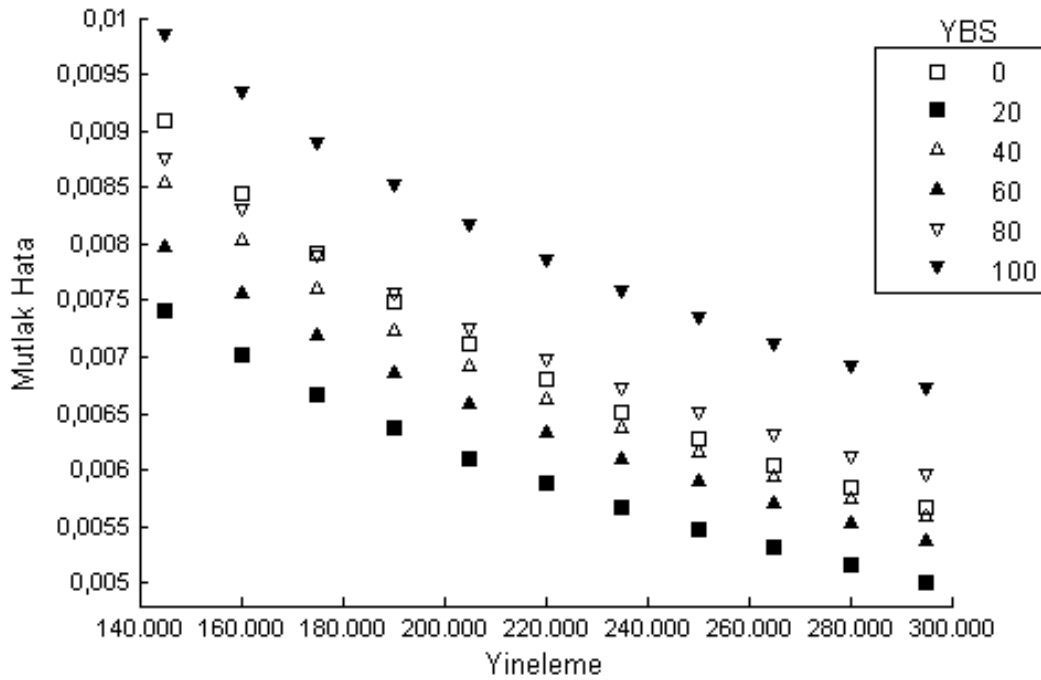
İkinci adımda, 5 katmanlı ve her katmanında 5 nöron bulunan ileri yönlü yapay sinir ağlarında düzenli, düzensiz ve Küçük Dünya ağ yapılarının öğrenme hatasına etkileri 5 farklı veri kümesi için elde edilmiş ve Şekil 5.3’ de gösterilmiştir. Öğrenme hatası olarak yapay sinir ağları uygulamalarında yaygın olarak kullanılan mutlak hata kullanılmıştır.

Tablo 5.1. İleri yönlü yapay sinir ağlarında farklı katman ve nöron sayıları için Küçük Dünya ağı YBS değer aralığı

Nöron Sayısı	Katman Sayısı	Toplam Bağlantı Sayısı	YBS
5	5	100	18±5
5	7	150	28±6
10	5	400	48±6
10	10	900	340±60
15	8	1575	320±80
15	15	3150	1310±250

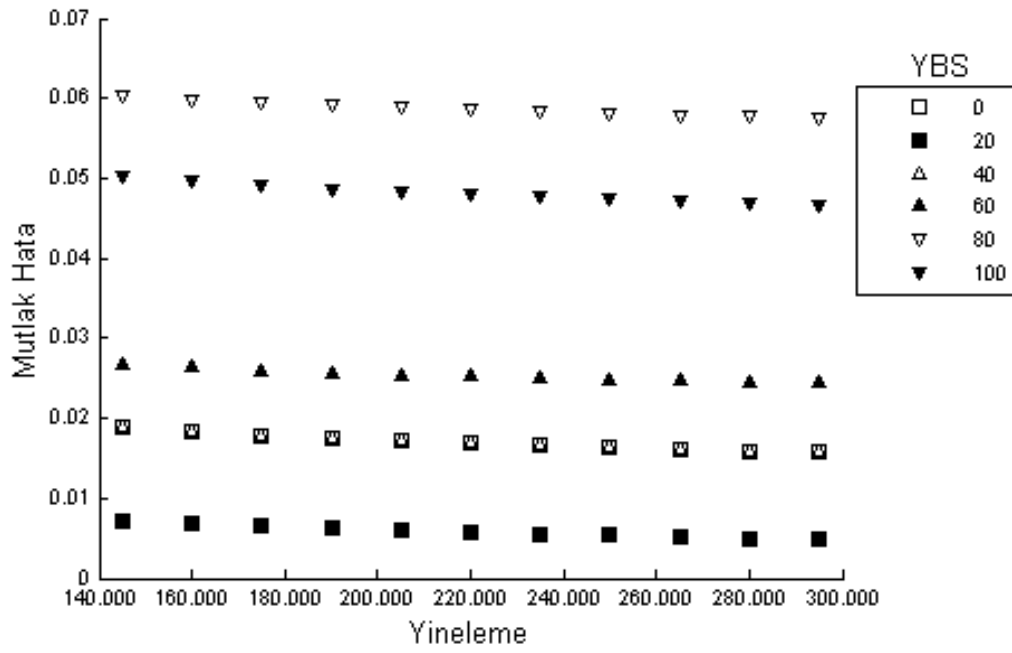
Tablo 5.1’de 5x5 yapay sinir ağı için Küçük Dünya ağ yapısı YBS=18±5 aralığı için elde edilmektedir. Bu nedenle Şekil 5.3’ de YBS=20, Küçük Dünya ağına karşılık gelmektedir. Küçük Dünya ağı için mutlak hata minimum (0.005) olmaktadır. YBS’ nin bu değerinin altında (düzenli ağ, YBS=0) ve üzerinde (düzensiz ağ) ise mutlak hata büyümektedir.



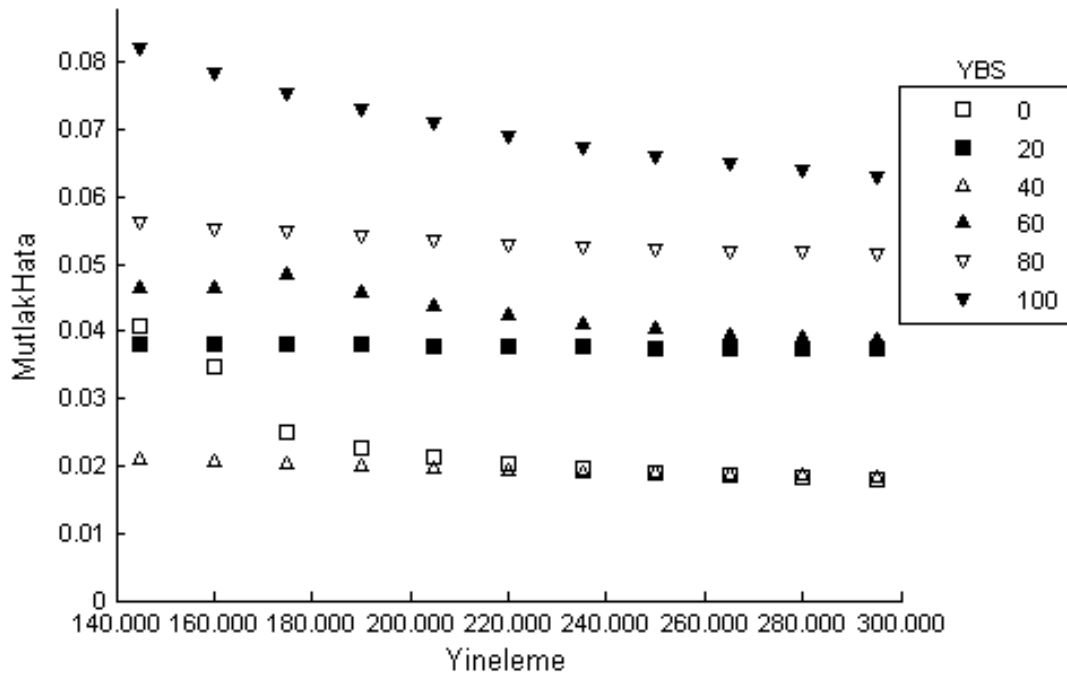


Şekil 5.3. 5x5 nöron ağında, 5 örnekli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi.

Son olarak, aynı ağ için veri küme sayısı artırılmış ve düzenli, düzensiz ve Küçük Dünya ağ yapılarının öğrenme hatasına etkileri 20 ve 32 örnekli veri kümeleri için elde edilmiş ve sonuçlar sırasıyla Şekil 5.4 ve Şekil 5.5’ te gösterilmiştir. Veri kümesi sayısı 20 olduğunda, veri kümesi 5 için elde edildiği gibi Küçük Dünya ağ yapısı için mutlak hata minimum (0.0065) olmakta; YBS’nin Küçük Dünya ağ yapısına karşılık gelen değerinin (YBS=20) altında (düzenli ağ, YBS=0) ve üzerinde (düzensiz ağ) ise mutlak hata büyümektedir. Ayrıca, ağa verilen veri kümesi sayısı arttığında öğrenme hatasının minimum değerinde bir miktar artış gözlenmiştir. Veri kümesi sayısı 32 olduğunda ise, Küçük Dünya ağ yapısının daha düşük veri küme sayıları için elde edilen en iyi öğrenme performansına sahip olma özelliğinin ortadan kalktığı görülmektedir. Bu durumda en iyi öğrenme performansı YBS=40 değeri (düzensiz ağ) için elde edilmektedir. YBS>0 için mutlak hata tekrar büyümektedir.



Şekil 5.4. 5x5 nöron ağında, 20 örnekleli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi.

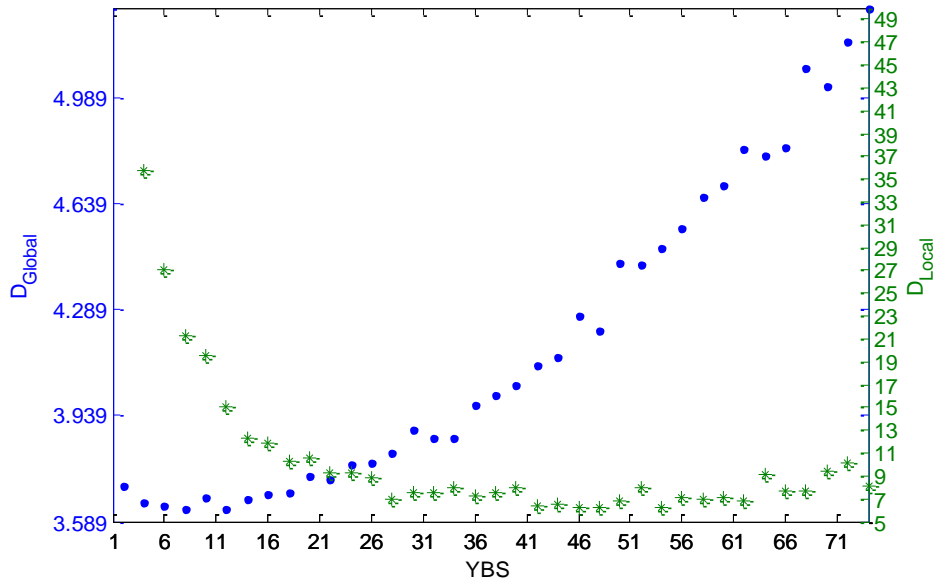


Şekil 5.5. 5x5 nöron ağında, 32 örnekleli veri kümesi için farklı ağ topolojilerinin öğrenme hatasına etkisi.

Sonuç olarak, bu çalışmada 5x5 ileri yönlü yapay sinir ağlarında farklı veri küme sayıları için düzenli, düzensiz ve Küçük Dünya ağ yapılarının öğrenme hatasına etkileri incelenmiş ve düşük veri küme sayıları için Küçük Dünya ağ yapısının en iyi öğrenme performansına sahip olduğu gösterilmiştir. Ancak, veri küme sayısı arttıkça Küçük Dünya ağ yapısının öğrenme performansını iyileştirme etkisinin ortadan kalktığı görülmektedir. Farklı katman ve nöron içeren ileri yönlü sinir ağlarında farklı veri küme sayıları için düzenli, düzensiz ve Küçük Dünya ağ yapılarının öğrenme hatasına etkilerini belirlemeye yönelik çalışmalar devam etmektedir.

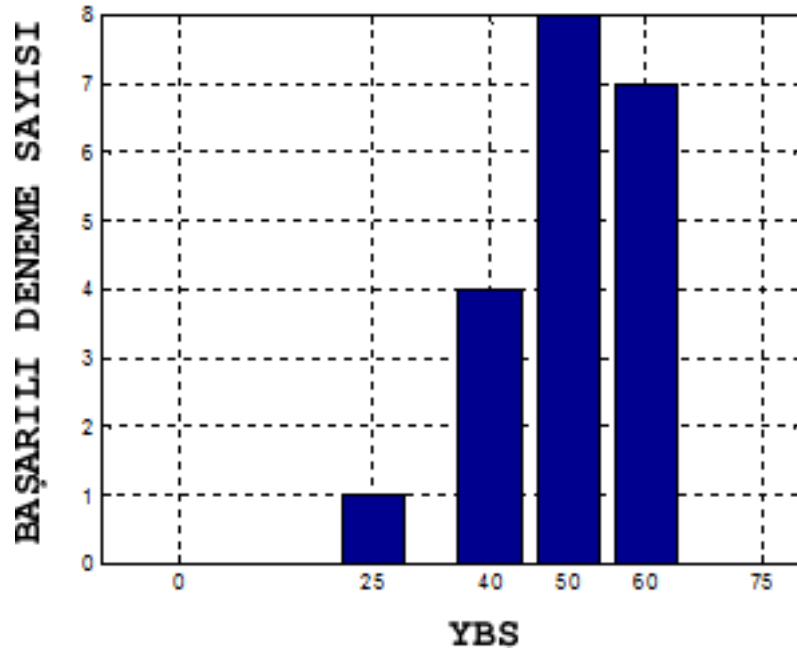
## **5.2. İleri Yönlü Watt-Strogatz Küçük Dünya Ağının Veri Seti Boyutuna Göre Öğrenme Performansının Araştırılması**

Bu çalışmada, 5 katmanlı ve her katmanında 5 nöronu bulunan geleneksel YSA topolojisi temel alınarak elde edilen Watts-Strogatz Küçük Dünya ağı öğrenme performansının veri seti büyüklüğüne göre değişimi araştırılmıştır. Bu bağlamda, 100 örneklilik, 5 girişli ve 5 çıkışlı bir dijital veri seti rastgele hazırlanmıştır. Sonraki süreçte bu topolojiye ait Watts-Strogatz Küçük Dünya ağ aralığı tespit edilmiştir. Aralık tespiti için Watts-Strogatz Bağlantı yenileme metodu kullanılmıştır. Bu metoda, bağlantı sayısı sabit tutularak bir nöronun var olan bağlantısı koparılarak, nöronun daha önce bağlı olmadığı yeni bir nörona bağlantı eklenmektedir. Bundan dolayı, 100 bağlantı sayısına sahip 5x5 ağ topolojisinin için 75 farklı yeni bağlantı kurulabilmektedir. Bu bağlantı metoduna ait yapısal davranış bölüm 3.1.1.1’de belirtilmiştir. Her bir yenilenen bağlantı için global ve lokal parametreler hesaplanmış ve Şekil 5.6’da gösterilmiştir.

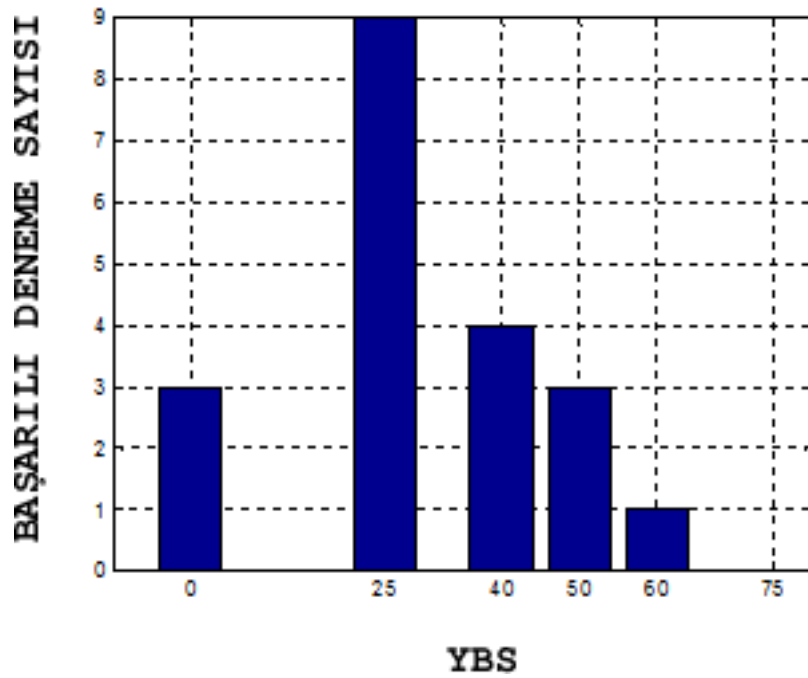


Şekil 5.6. 5x5 İleri yönlü yapay sinir ağı için, Watt-Strogatz metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Local}$  katsayılarının değişimi.

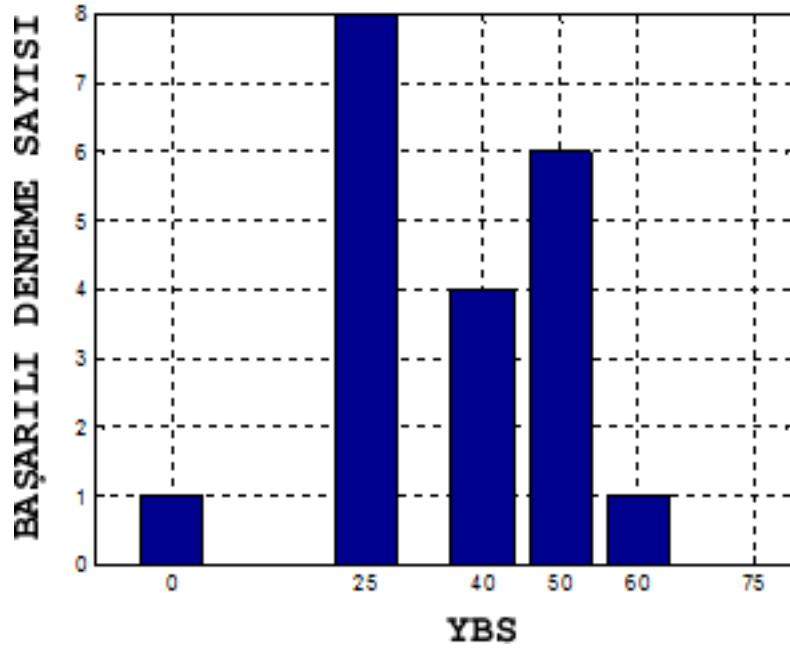
Şekil 5.6' da görüldüğü gibi, global ve lokal parametre değerleri YBS=20 değerine yakın noktada kesişmektedir.  $D_{Local}$ , YBS=25 seviyesinde doyuma ulaşarak en küçük değerini ulaşmakta ve  $D_{Global}$ , ise küçük değişimlerle sürekli artış göstermektedir. Watts-Strogatz Küçük Dünya ağlarının elde edilebilmesi için gerekli olan minimum YBS değeri 20 seviyelerindedir. Aralık araştırmasını pekiştirmek amacıyla, hazırlanan veri setinin farklı sayılı örnekleri için YBS sayısı 0-75 arası değişen 5x5 YSA topolojileri eğitime tabi tutulmuşlardır. 100000 iterasyon üzerinden eğitim simülasyonları gerçekleştirilmiş ve her bir topoloji için 20 farklı deneme yapılmıştır. Her deneme sonucunda minimum MSE değeri baz alınarak, topolojilerin eğitim performansı gözlemlenmiştir. Değişken örnekli veri seti kullanarak gerçekleştirilen 20 deneme sonucu olarak elde edilen minimum MSE değerine sahip başarılı deneme sayısının YBS ile dağılımı, sırasıyla 5, 20, 40, 70 ve 100 veri seti için Şekil 5.7-11' de gösterilmiştir.



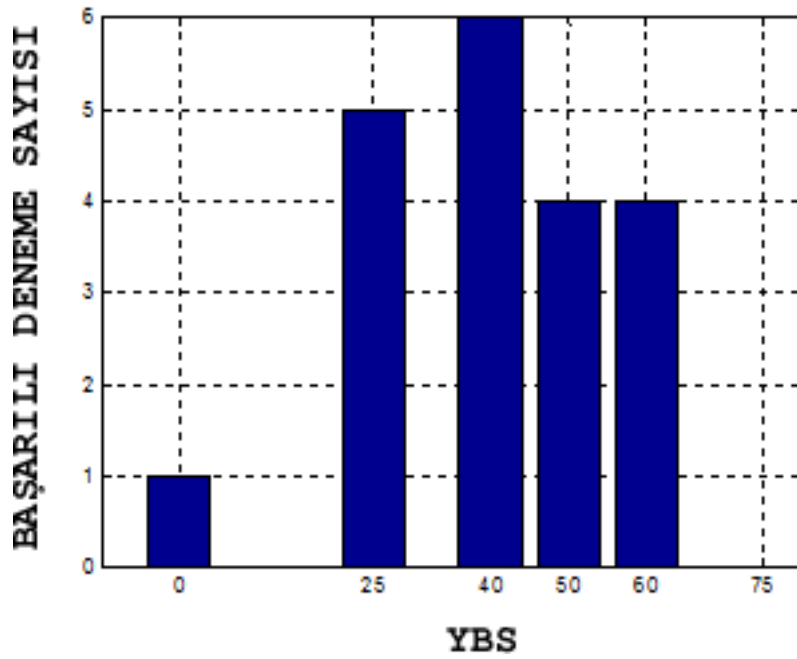
Şekil 5.7. 5x5 nöron ağının, 5 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı.



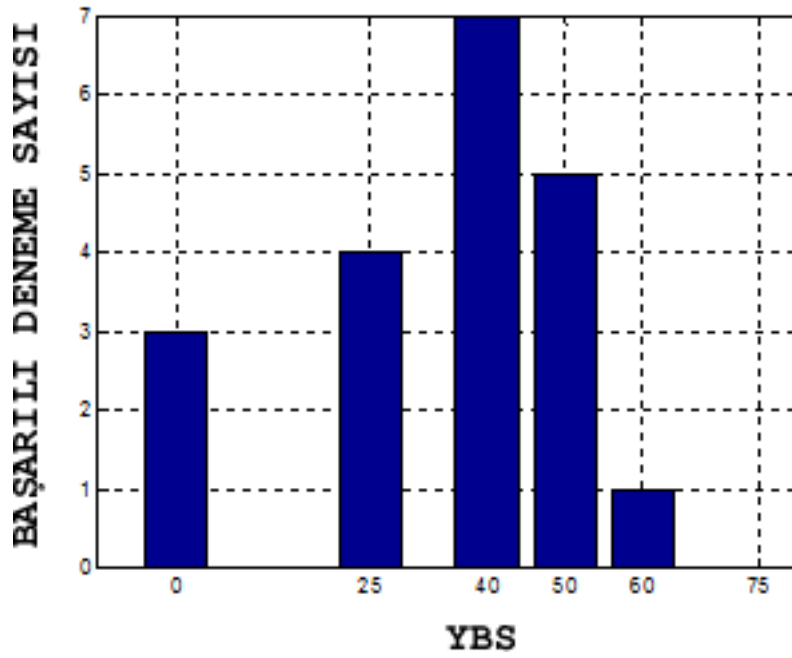
Şekil 5.8. 5x5 nöron ağının, 20 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı



Şekil 5.9. 5x5 nöron ağının, 40 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı.

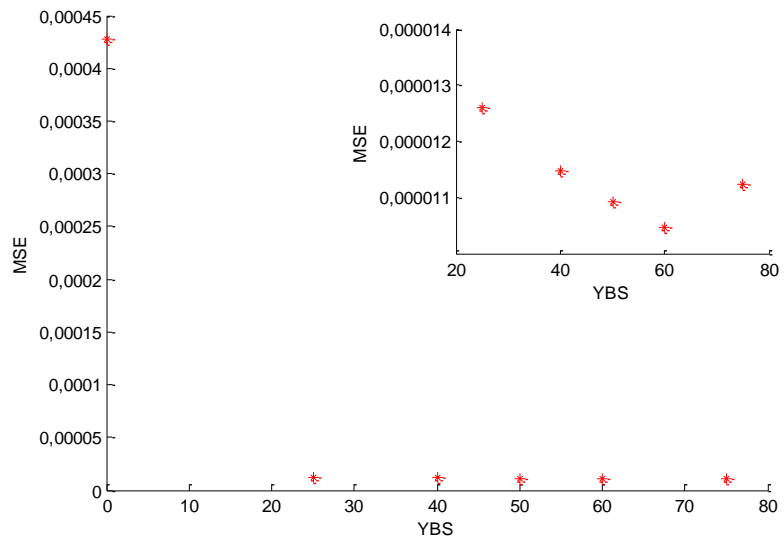


Şekil 5.10. 5x5 nöron ağının, 70 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı.

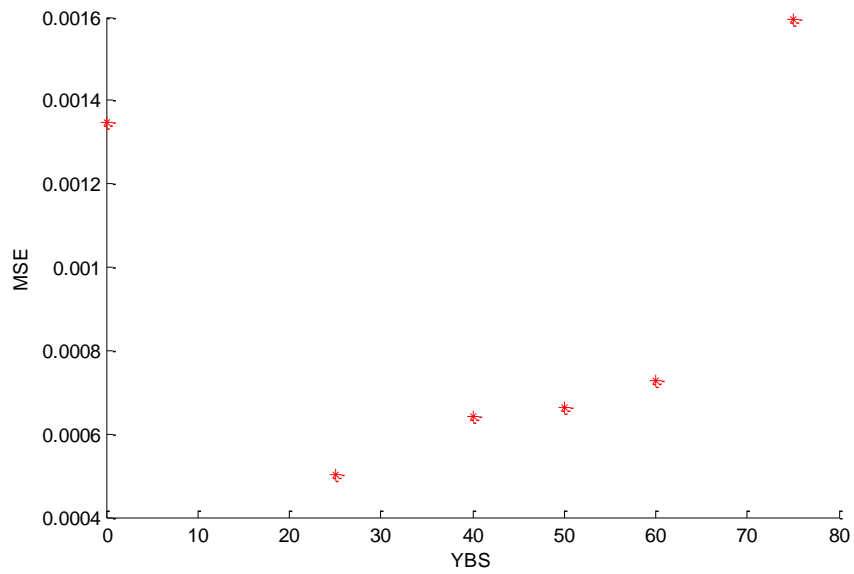


Şekil 5.11. 5x5 nöron ağının, 100 veri seti eğitiminden elde edilen başarılı deneme sayısının YBS ile dağılımı.

Şekil 5.7-11' de görüldüğü üzere Watts-Strogatz Küçük Dünya ağının davranışının elde edilmesi için derekli YBS aralığı 25-60 arasında olduğu gözlemlenmiştir. Ayrıca veri seti örnek sayısı büyüdükçe düzenli ağ performansı artmaktadır. Bulunan bu aralık  $D_{Global}$  ve  $D_{Lokal}$  parametreleriyle elde edilen aralıkla örtüşmektedir. Eğitim sürecinde, her bir veri seti ve network topolojisi için, 20 deneme ile elde edilen eğitim hatalarının ortalamaları alınmış ve Şekil 5.12-16' de gösterilmiştir.

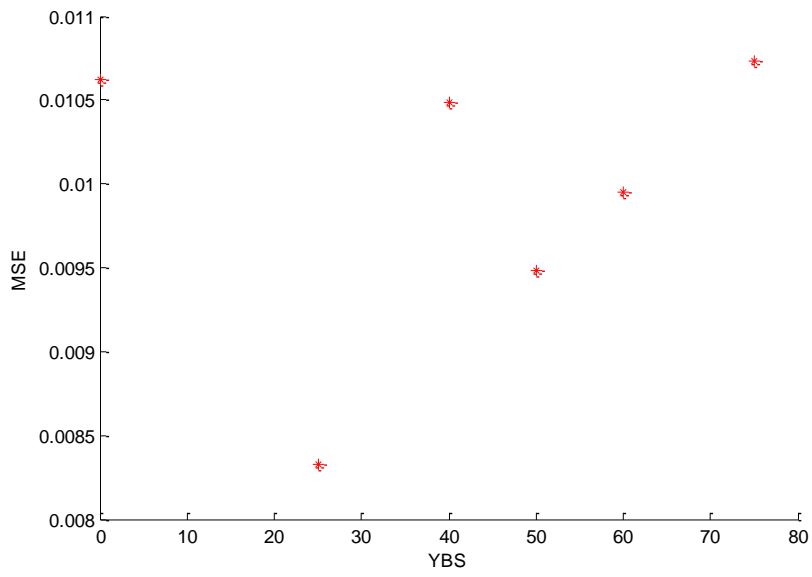


Şekil 5.12. 5 örnekli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi.

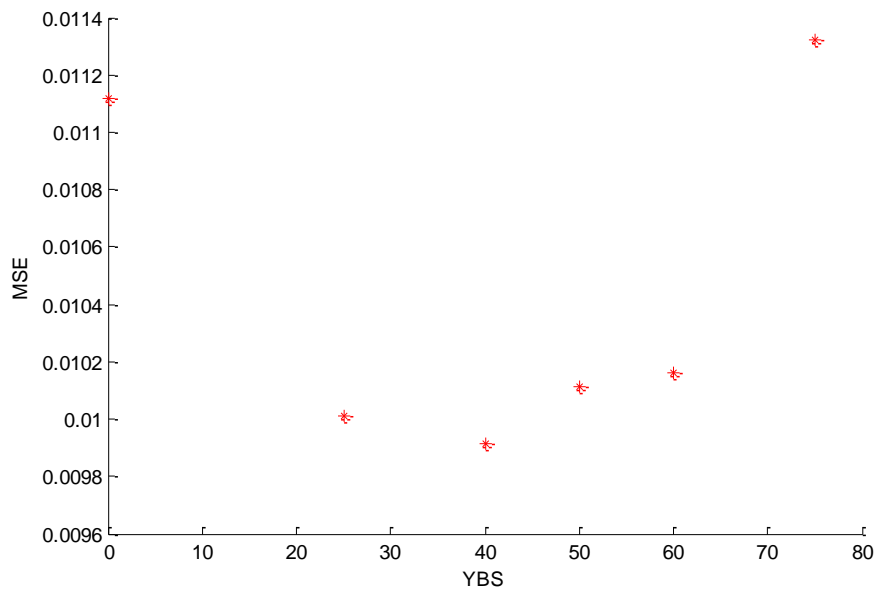


Şekil 5.13. 20 örnekli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi.

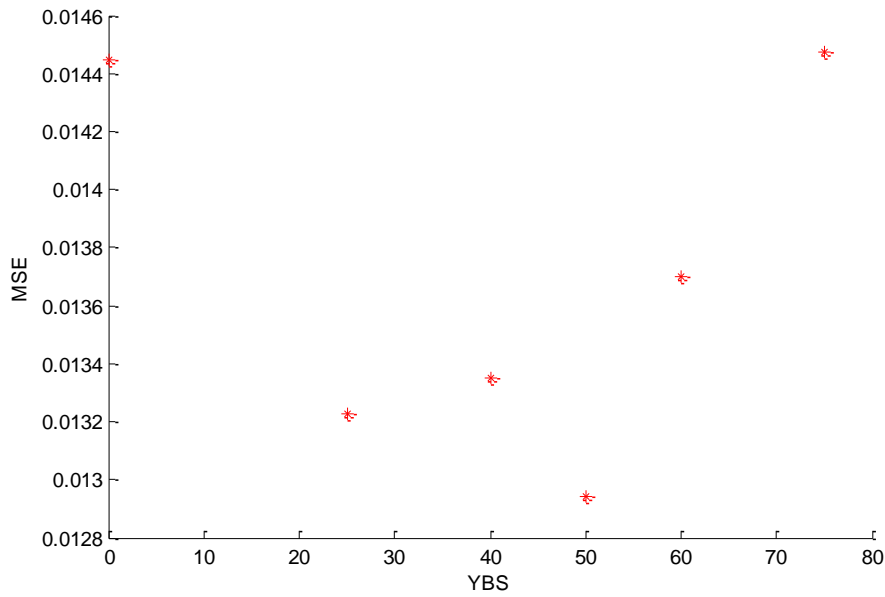




Şekil 5.14. 40 örneklı veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile deęişimi.



Şekil 5.15. 70 örneklı veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile deęişimi.



Şekil 5.16. 100 örnekli veri seti ile yapılan 20 deneme için ortalama eğitim hatasının YBS ile değişimi.

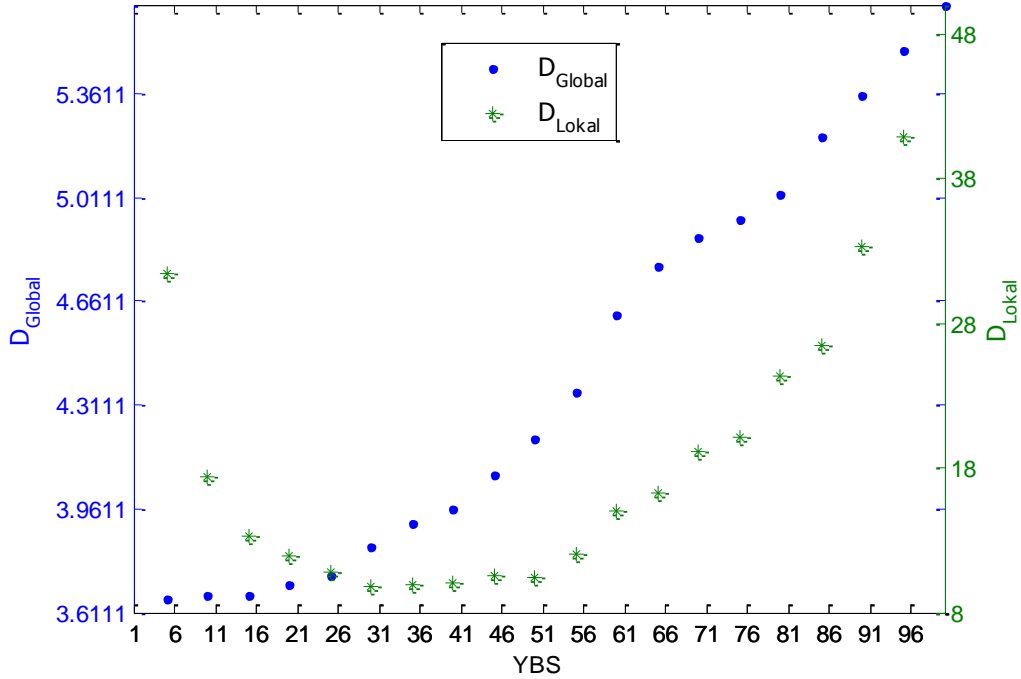
YBS ile ortalama eğitim hatası değişim Şekillerinde görüldüğü üzere, en düşük öğrenme hatasının elde edildiği YBS aralığı 25-60 'dır. Sonuç olarak Watts-Strogatz Küçük Dünya ağlarının eğitim performansının düzenli ağa göre daha iyi olduğu gözlemlenmiştir.

### 5.3. İleri Yönlü Newman-Watts Küçük Dünya Ağlarının Araştırılması

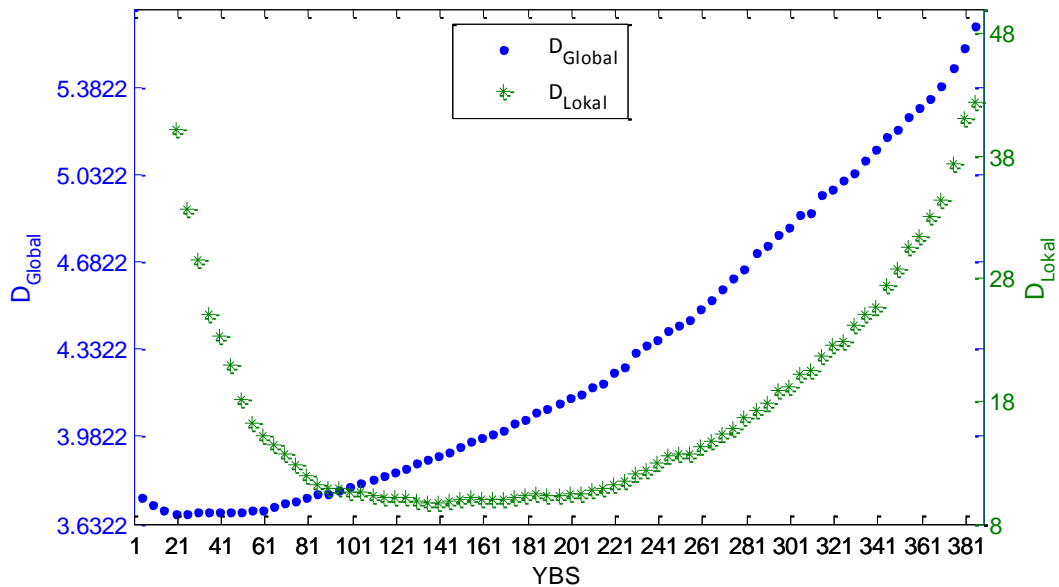
#### 5.3.1. İleri yönlü Newman-Watts Küçük Dünya ağı YBS aralığının araştırılması

Bu çalışmada Newman-Watts bağlantı yenileme algoritması ile elde edilen Küçük Dünya ağının, ağ boyutuna göre YBS aralığı tespit edilmeye çalışılmıştır. Bu bağlamda, Bölüm 3.1.1.2' de ifade edildiği gibi temel alınan YSA topolojisine yeni bağlantılar eklenmiş ve ağın bağlantı sayısı artırılmıştır. Bu işlem ağ içerisinde katmanlar arasında bağlı olmayan nöron kalmayana kadar devam etmektedir. Her bir bağlantı eklendiğinde  $D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmaktadır. Böylece farklı topolojilere sahip Küçük Dünya ağları için gerekli minimum YBS değeri elde edilmeye çalışılmıştır. Ağ topolojileri olarak , 5 katmanlı ve 10 katmanlı

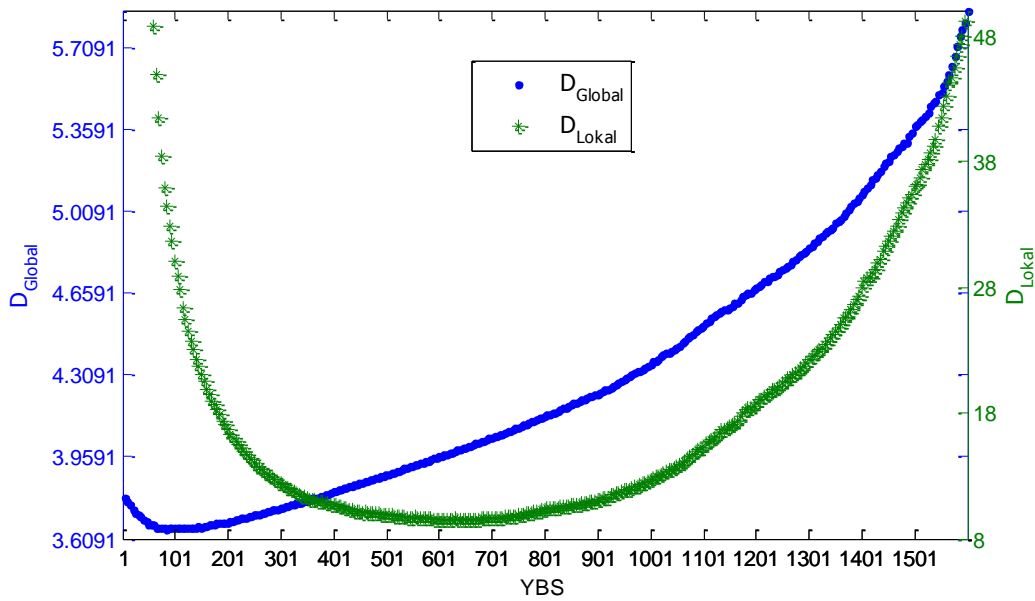
ileri yönlü YSA ağları kullanılmıştır. 5 katmanlı ağlarda, sırasıyla 5,10 ve 20 nöron kullanılmış, 10 katmanlı ağlarda ise sırasıyla 5,10 ve 20 nöron kullanılmıştır. 5 katmanlı ağlar için elde edilen  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının YBS ile değişimi Şekil 5.17-19’ da verilmiştir.



Şekil 5.17. 5x5 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

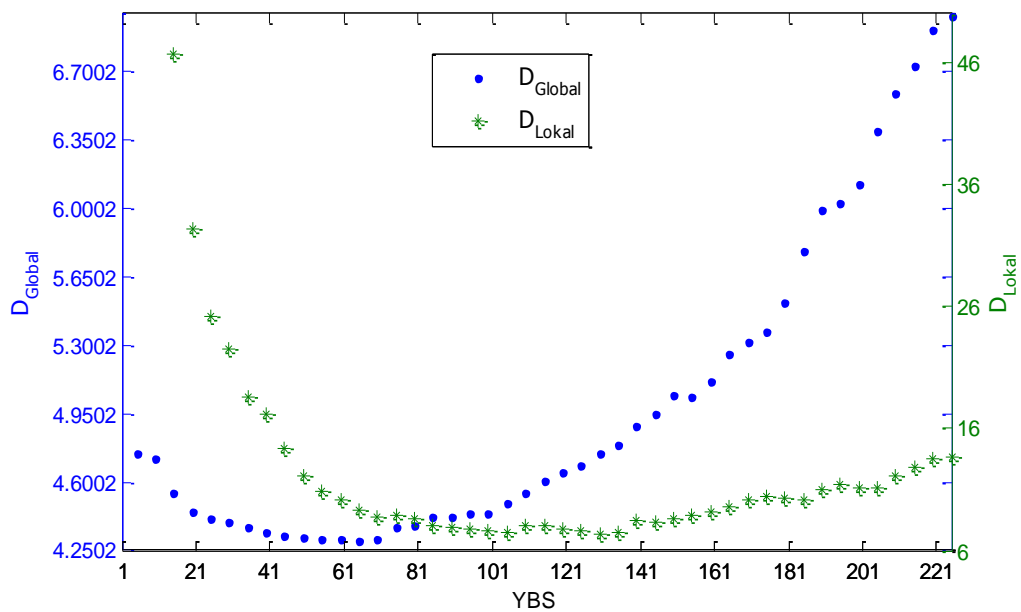


Şekil 5.18. 5x10 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

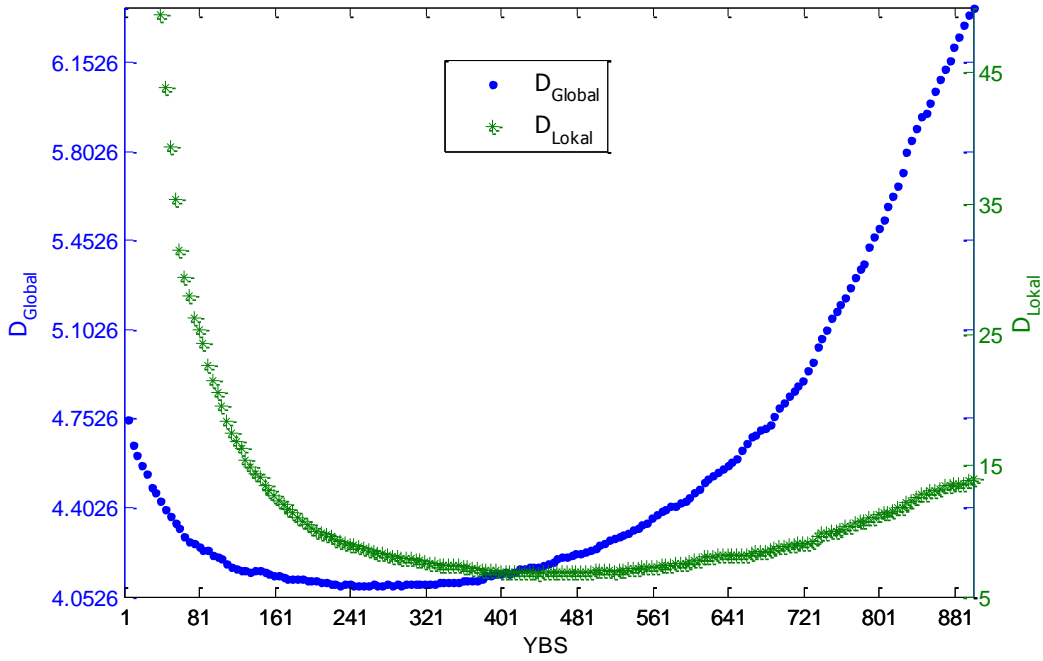


Şekil 5.19. 5x20 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

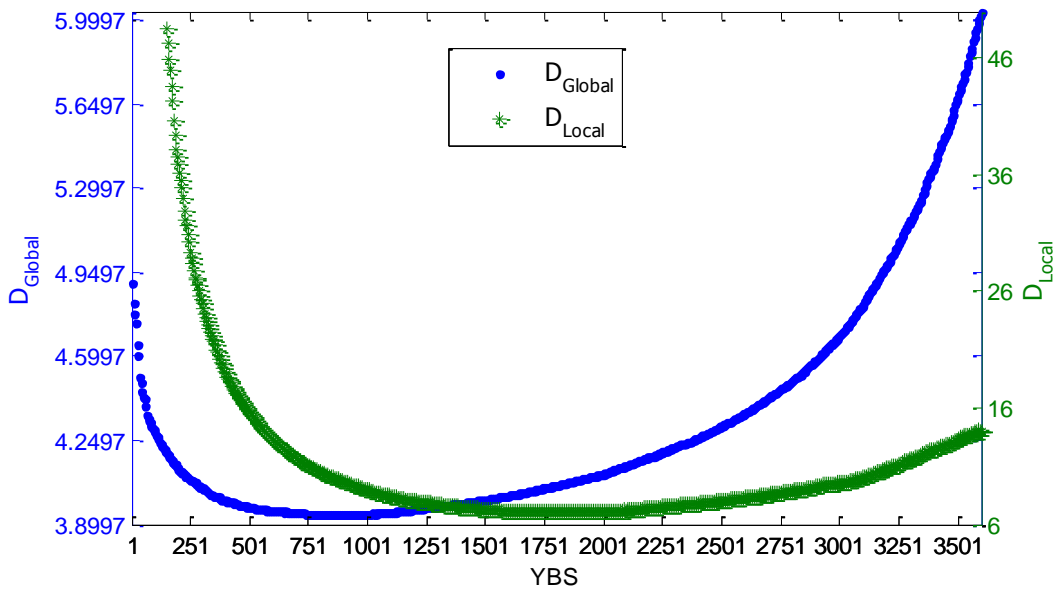
Şekil 5.17-19' da görüldüğü üzere; 5x5 topoloji için minimum  $YBS \cong 25$ , 5x10 topoloji için minimum  $YBS \cong 90$  ve 5x20 topoloji için minimum  $YBS \cong 375$ , olarak bulunmuştur. 10 katmanlı ağlar için elde edilen  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının YBS ile değiş şekilleri Şekil 5.20-22' de verilmiştir.



Şekil 5.20. 10x5 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.



Şekil 5.21. 10x10 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.



Şekil 5.22. 10x20 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

Şekil 5.20-22'da görüldüğü üzere; 10x5 topoloji için minimum  $YBS \cong 86$ , 10x10 topoloji için minimum  $YBS \cong 410$  ve 10x20 topoloji için minimum  $YBS \cong 1350$  olarak gözlemlenmiştir. Sonuç olarak 5 katmanlı ağlar için gerekli olan minimum YBS:

$$\text{YBS sayısı} \cong 1/4 * \text{Toplam Bağlantı Sayısı} \quad (5.1)$$

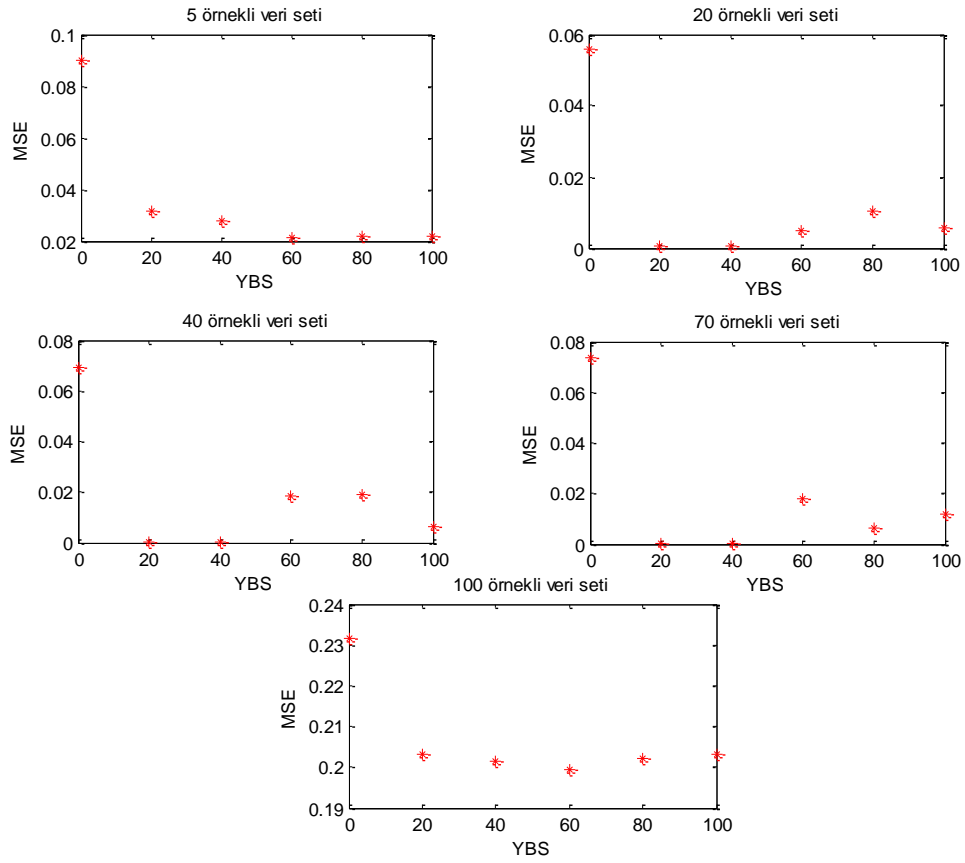
olarak tanımlanmıştır. 10 katmanlı ağlar için ise, YBS sayısı:

$$\text{YBS} \cong 1/2 * \text{Toplam Bağlantı Sayısı} \quad (5.2)$$

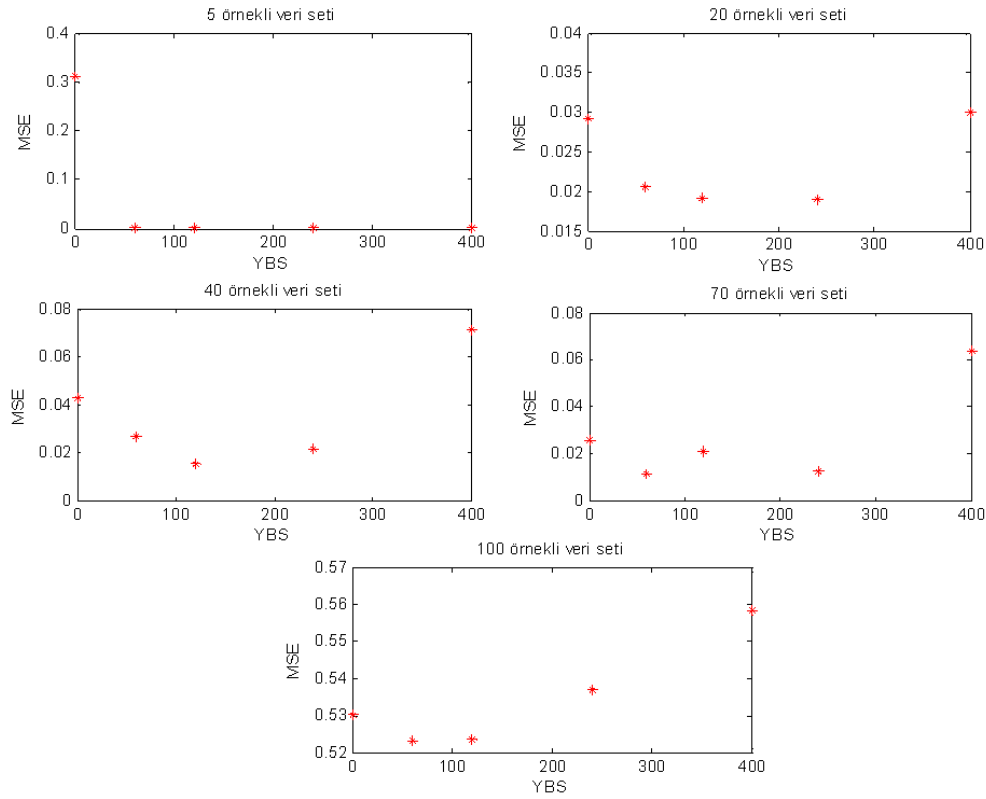
olarak ifade edilmiştir.

### **5.3.2. İleri yönlü Newman-Watts Küçük Dünya ağının veri seti boyutuna göre öğrenme performansının araştırılması**

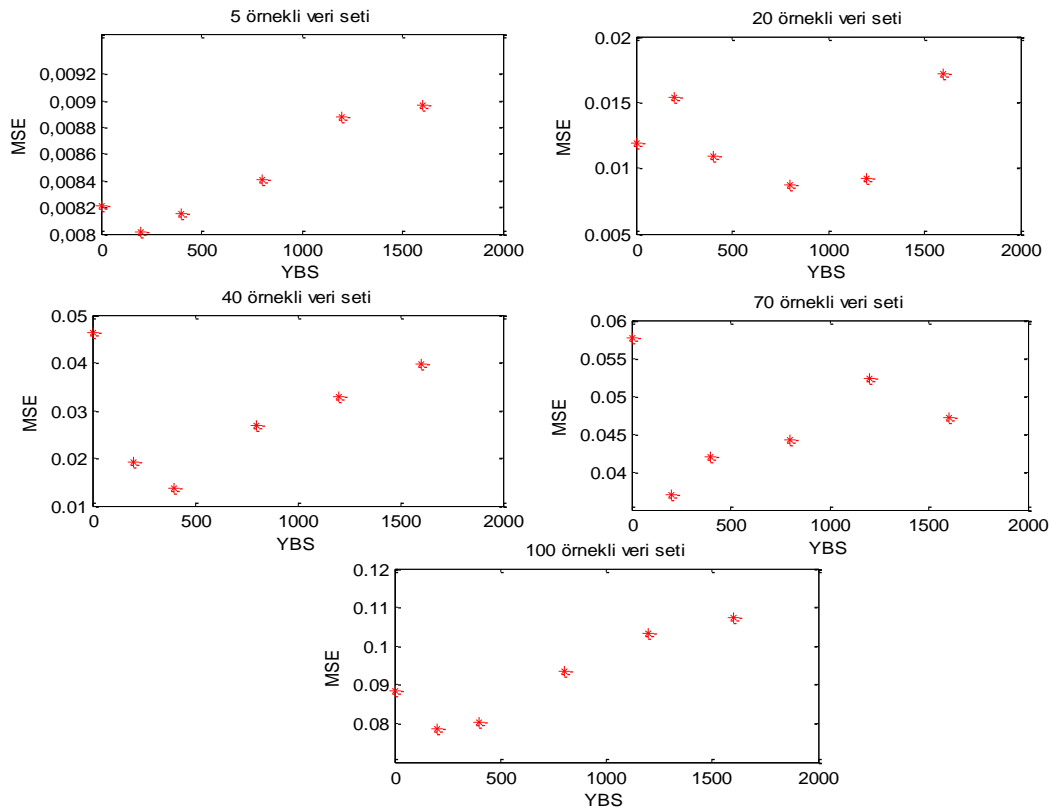
Bu çalışmada, 5 ile 10 katmanlı ve her katmanında 5, 10 ile 20 nöronu bulunan geleneksel YSA topolojisi kullanılmıştır. Newman-Watts bağlantı yenileme metodu kullanılarak Küçük Dünya ağının farklı büyüklükte veri seti ile öğrenme performansı araştırılmıştır. Bu nedenle sırasıyla 5, 20, 40, 70 ve 100 veri setiyle her bir topoloji eğitilmiş ve elde edilen öğrenme performansı MSE istatistik hatasıyla ölçülmeye çalışılmıştır. 5 katmanlı topolojilerin eğitim sürecinden elde edilen hataların ortalaması alınarak Şekil 5.23-25 'te gösterilmiştir.



Şekil 5.23. 5x5 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi.



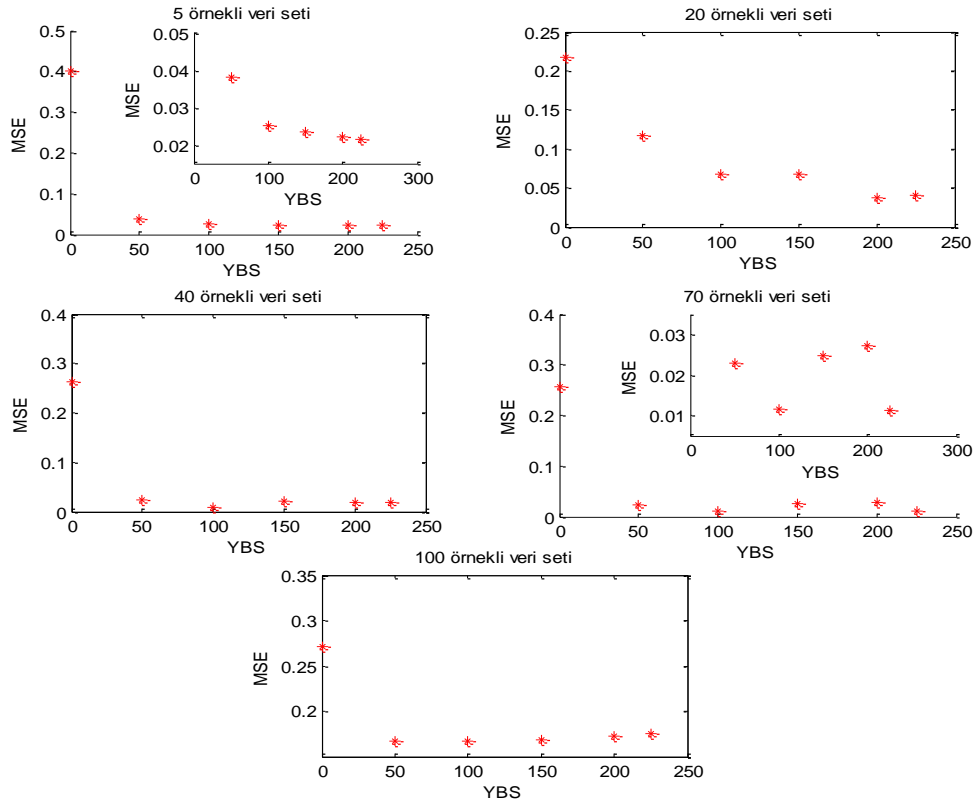
Şekil 5.24. 5x10 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi.



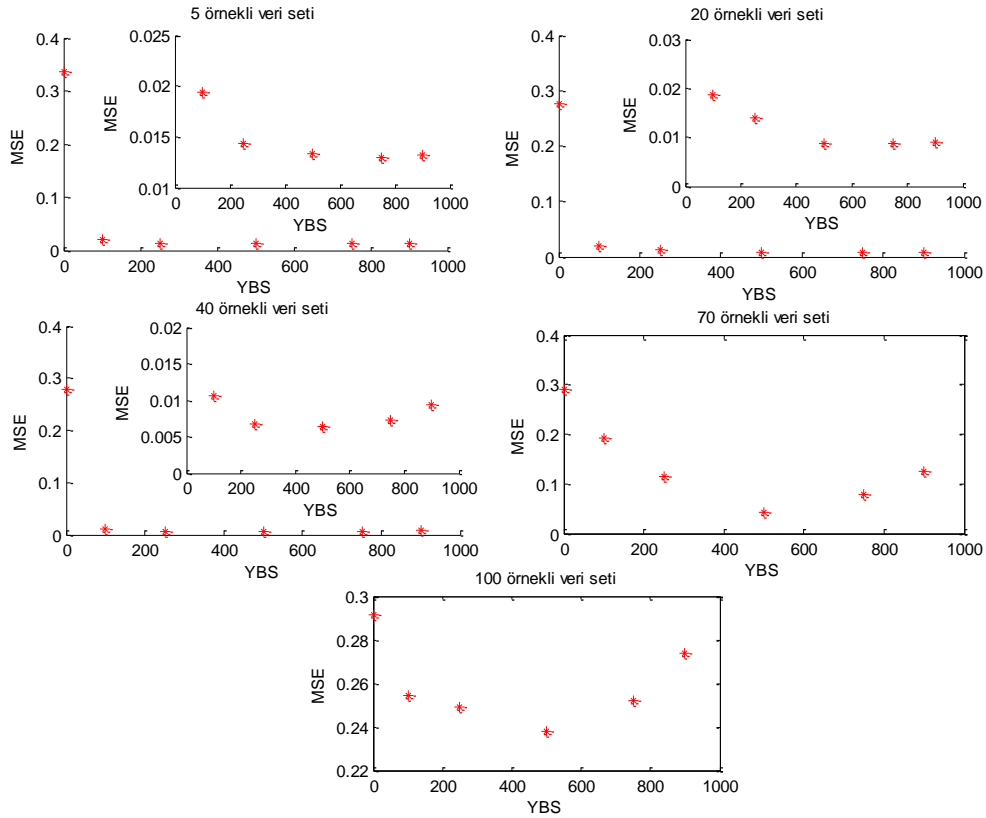
Şekil 5.25. 5x20 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi

5 katmanlı topoloji deneylerinden elde edilen sonuçlara göre Newman-Watts Küçük Dünya ağlarının öğrenme performansı düzenli ağa oranla daha iyidir. Veri seti arttıkça düzeli ağa oranla Küçük Dünya ağları performansını yitirmektedir. Katmanlardaki nöron sayısı kademi olarak artırıldığında ise düzenli ağ ile Küçük Dünya ağlarının öğrenme hataları düşmekte ve Küçük Dünya davranış aralığı daralmaktadır. 10 katmanlı topolojilerin eğitim sürecinden elde edilen hataların ortalaması alınarak Şekil 5.26-28' te gösterilmiştir.

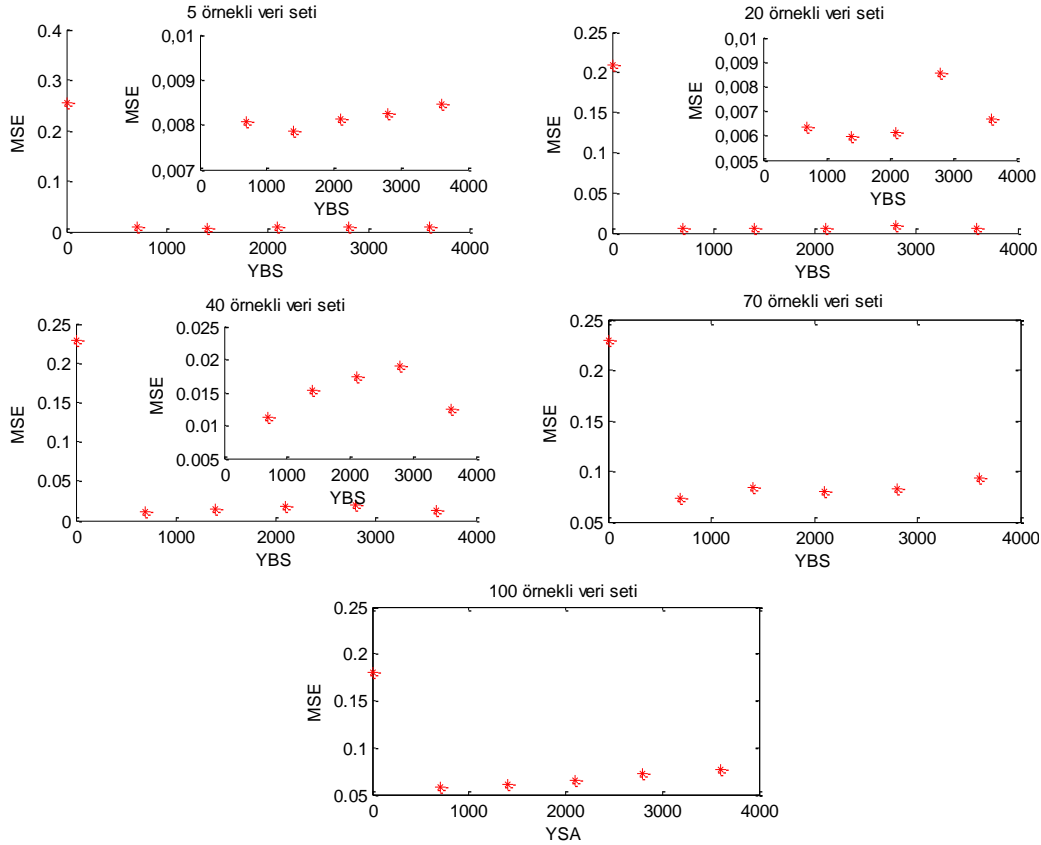




Şekil 5.26. 10x5 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi.



Şekil 5.27. 10x10 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi.



Şekil 5.28. 10x20 nöron ağının, farklı büyüklükteki veri seti için eğitim hatasının YBS ile değişimi.

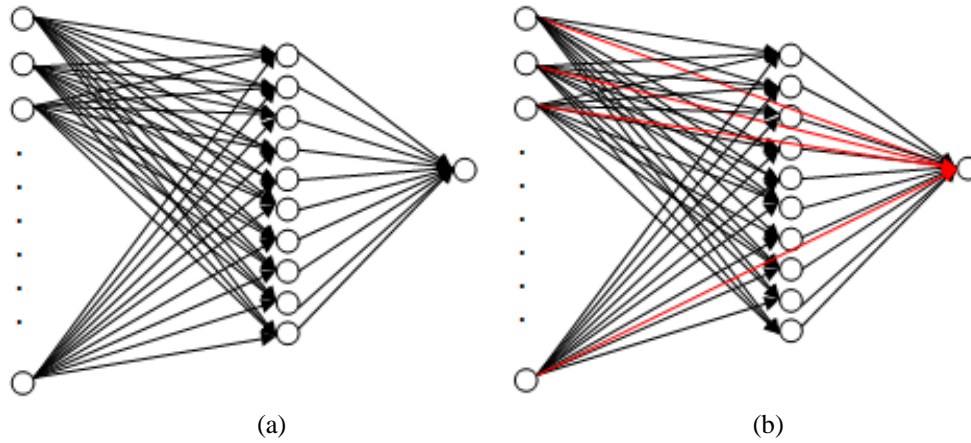
10 katmanlı mimariye, katman sayısı artırıldığında görülmüştür ki, 5 katmanlı mimariye oranla öğrenme hatası genel olarak artmıştır. Ancak düzenli ağ ile Küçük Dünya ağı arasındaki öğrenme performansı değişimi belirgin şekilde ortaya çıkmış ve Küçük Dünya ağlarının öğrenme performansının daha iyi olduğu net bir şekilde görülmüştür. Bunun yanında, katman sayısı sabit tutulup nöron sayısı artırıldığında, Küçük Dünya ağı ile düzenli ağ arasında fark gözlemlenmemiştir.

#### 5.4. Karesel Olmayan İleri Yönlü Watt-Strogatz Küçük Dünya Mimarisinin Epilepsi Teşhisinde Kullanımı

Bu çalışmada, karesel olmayan bir ağ topolojisinde Watts-Strogatz Küçük Dünya ağı performansı araştırılmıştır. Uygulamayı gerçekleştirmek amacıyla sentetik veri seti yerine, sağlıklı ve epilepsi hastalarından alınan 100 EEG örneği bölümü içeren ve 5 sınıfa ayrılmış bir veri seti kullanılmıştır. Her EEG bölümü, 23.6 saniye süreyle 173.6 Hz örnekleme frekansı ile kaydedilmiştir. [92]. EEG kayıtlarından göz ve kas

hareketlerinden kaynaklı istenmeyen durumlar temizlenmiştir. A ve B sınıflarında yüzeysel EEG ölçümü alınmış, C, D ve E sınıflarında ise içsel EEG ölçümü yapılmıştır. A,B sınıfları ölçümler, 5 sağlıklı bireyden, gözler açık ve kapalı şekilde alınmış, C ve D sınıfları 5 epilepsi hastasının epileptik ve epileptik olmayan beyin yarım kürelerinden epileptik atak geçirmediği zamanda alınmıştır. Son olarak E sınıfı ise 5 epilepsi hastasının atak geçirdikleri zamanda ölçülmüştür [103]. Çalışmada, veri seti A, B sağlıklı ve C, D, E epilepsi hastası olarak 2 sınıf halinde kullanılmıştır.

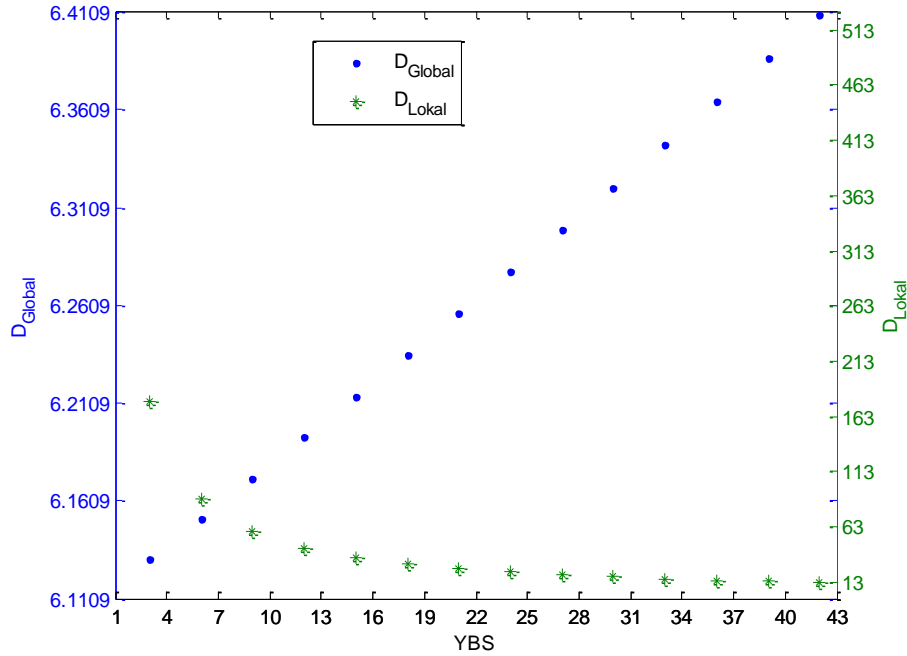
Küçük Dünya ağı elde edebilmek amacıyla, karesel olmayan, geleneksel 3 katmanlı 42 girişli ve 1 çıkışlı bir YSA topolojisi seçilmiştir. 20 deneme yapılarak optimum gizli katman sayısı 10 olarak tayin edilmiştir. Temel olarak kullanılan 3 katmanlı YSA topolojisi (Düzenli ağ) ve Küçük Dünya ağlarını ifade eden topolojik gösterim Şekil 5.29’ da gösterilmiştir.



Şekil 5.29. (42-10-1) topolojili, Düzenli (a) ve Watts-Strogatz Küçük Dünya (b) ağ yapılarına ait topolojik gösterimi.

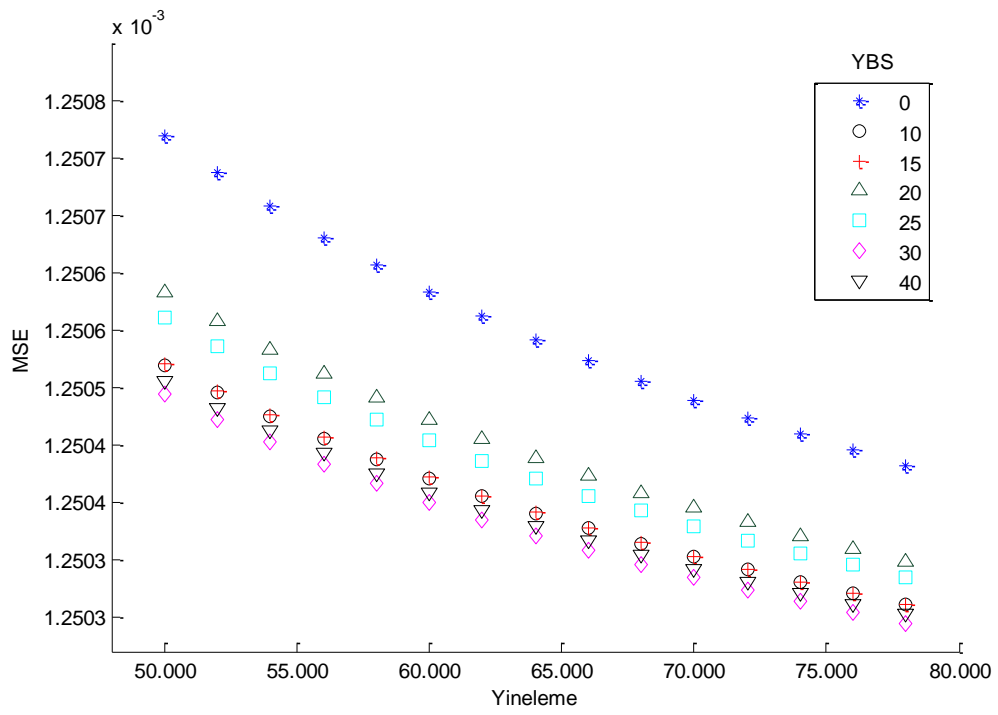
Geleneksel 3 ağ topolojisinden Küçük Dünya ağlarını elde edilmesi sürecinde, seçilen 42-10-1 nöronlu topolojide birbirleriyle bağlı olmayan nöron sayısı 42 olarak hesaplanmış ve bağlantı yenileme süreci 42 nörona kadar devam etmektedir. Her bir bağlantı yenileme sürecinde  $D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmıştır. YBS değişimi ile elde edilen katsayı değişkenlerine ait  $D_{Global}$  ve  $D_{Lokal}$  değerleri Şekil 5.30’da gösterilmiştir. Orada görüldüğü gibi  $D_{Global}$  YBS değeri arttıkça artmaktadır.  $D_{Lokal}$  ise azalarak doyuma ulaşmaktadır. Her iki parametrenin

küçük olduğu yerler Küçük Dünya Ağı topolojisi kabul edilmiştir. Bu bağlamda, Küçük Dünya ağı YBS aralığı yaklaşık 5-30 olarak gözlemlenmiştir (Şekil 5.30).

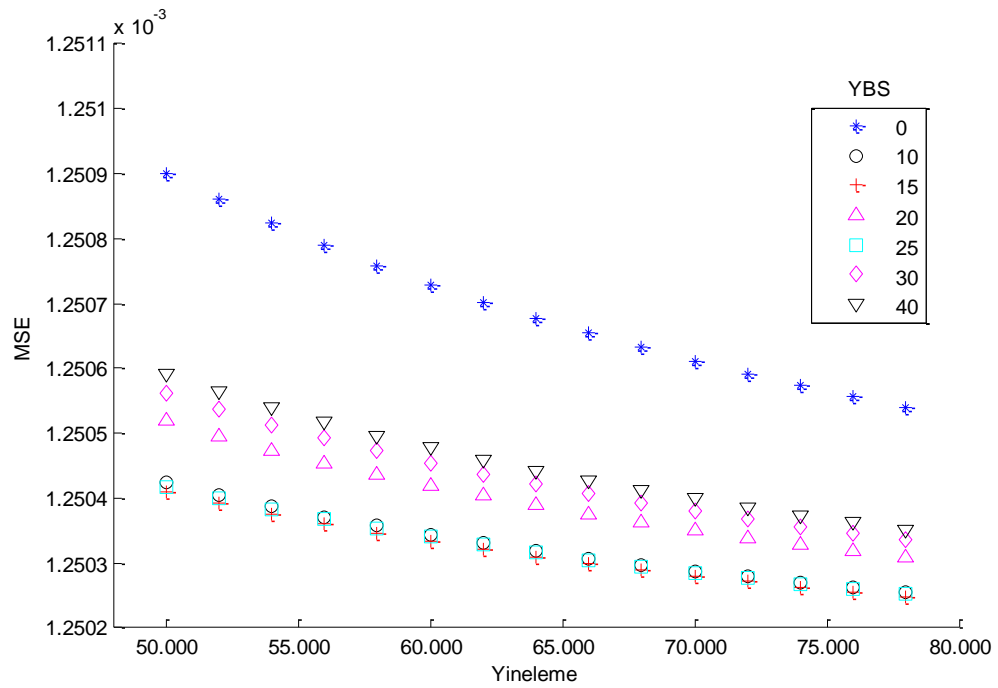


Şekil 5.30. 42-20-1 İleri yönlü yapay sinir ağı için, Newman-Watts metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

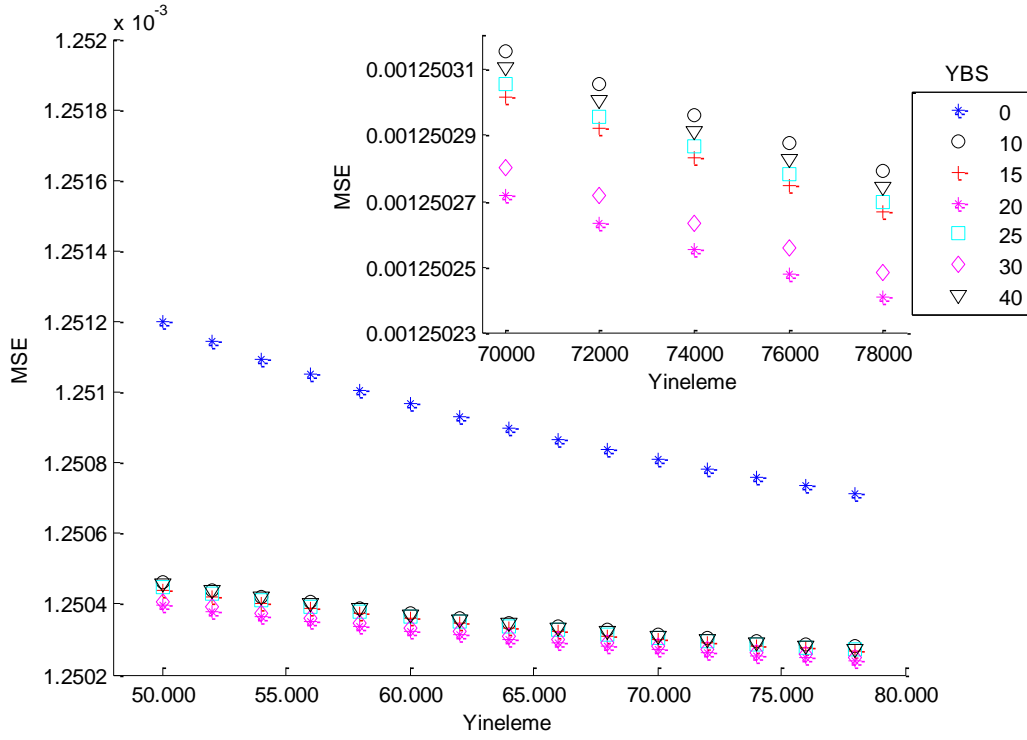
Küçük Dünya ağı aralığı bulunduğundan sonra eğitim performansını görmek amacıyla, baz alınan 3 katmanlı 42 girişli 1 çıkışlı mimarimizde, farklı gizli katman nöron sayısının eğitim performansı araştırılmıştır. Bu bağlamda, gizli katman için, sırasıyla 10, 20, 42 ve 63 nöron kullanılmıştır. Düzenli ve Küçük Dünya ağlarının performanslarını gözlemek amacıyla seçilen topolojide, sırasıyla 0, 10, 15, 20, 25, 30, 40 YBS değeri kullanılarak farklı ağ topolojileri elde edilmiştir. Topolojilerin eğitimi için 80000 yinelemeyle ağların eğitim hataları ortalama karesel hata (MSE) hesaplanmıştır. Farklı gizli katman nöron sayısına sahip ağların eğitim hatalarının yineleme ile değişimi Şekil 5.31’de gösterilmiştir.



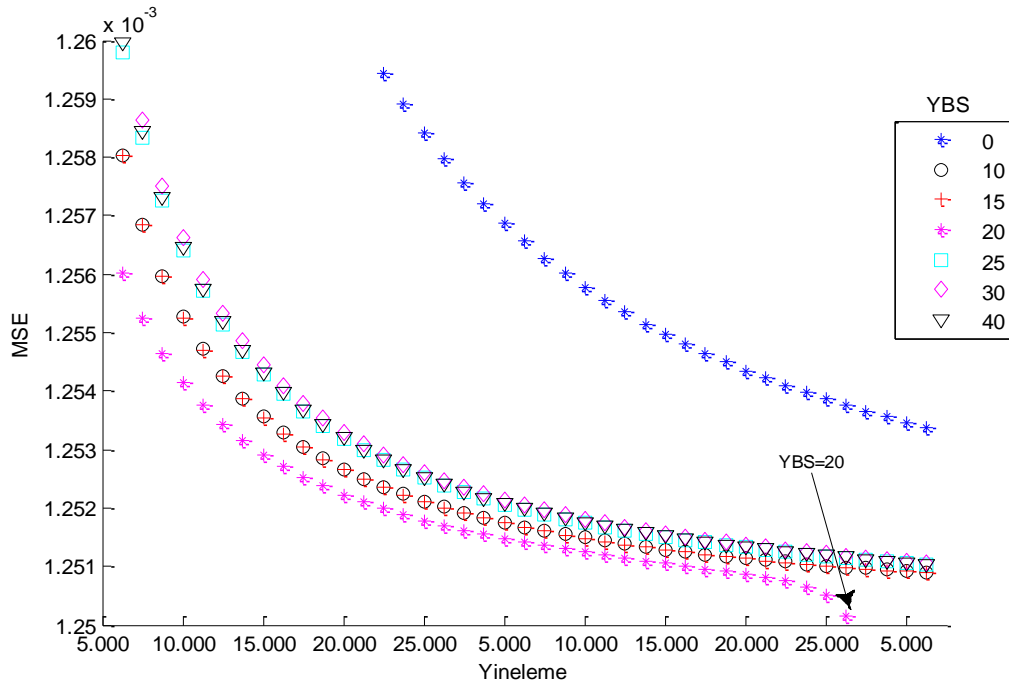
(a)



(b)



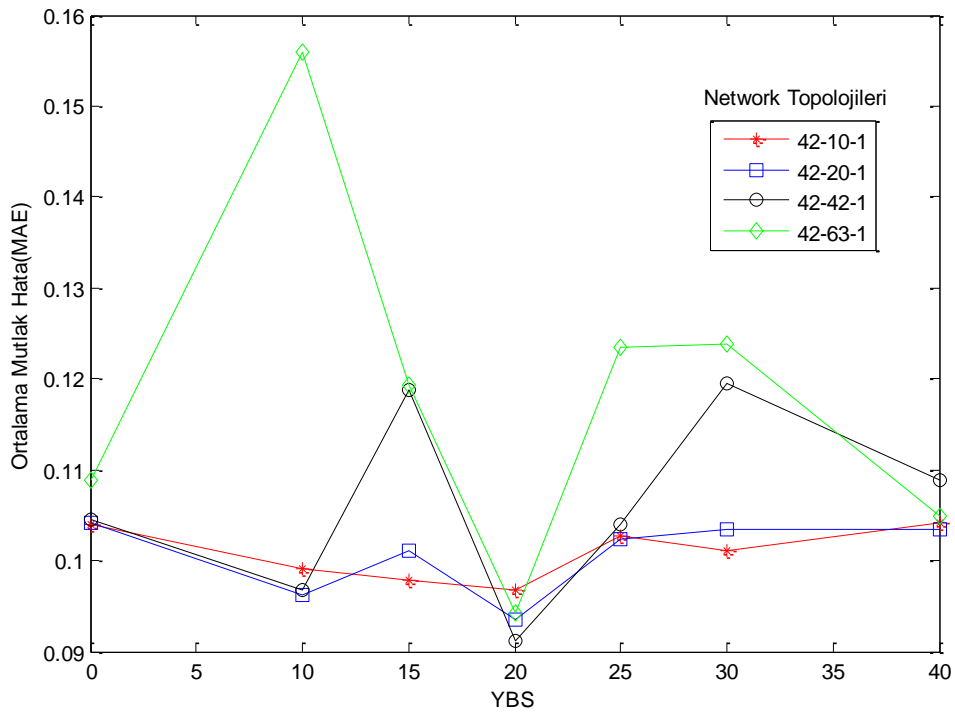
(c)



(d)

Şekil 5.31. 3 Katmanlı YSA mimarisinde YBS ile eğitim hatasının (MSE) değişimi. (a) 42-10-1, (b) 42-20-1, (c) 42-42-1 ve (d) 42-63-1.

Şekil 5.31’ de görüldüğü gibi, bağlantı sayıları değiştirilmiş ağların eğitim performansları düzenli ağa (YBS=0) göre daha düşük hesaplanmıştır. aynı zamanda eğitim sürecinin de daha kısa sürdüğü gözlemlenmiştir. Buradan hareketle Watt-Strogatz Küçük Dünya ağlarının eğitim epilepsi teşhisinde eğitim sürecini kısalttığı görülmüştür. Gizli katman nöron sayısının artırılmasının Küçük Dünya ağları performansını etkilemediği gözlemlenmiştir [104]. Son olarak, tüm ağların sınıflandırma hataları hesaplanmış ve YBS ile değişimi Şekil 5.32’ de gösterilmiştir.

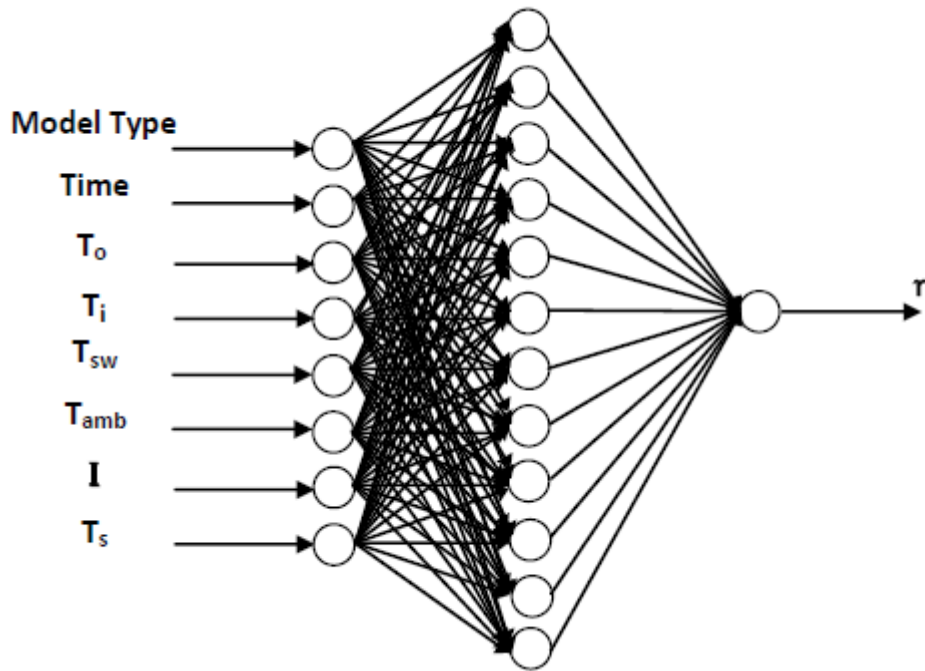


Şekil 5.32. 3 katmanlı, Farklı gizli katman nöron sayılı topolojilerin çıkış hatasının YBS ile değişimi.

Şekil 5.32’ de görüldüğü gibi, YBS=20 için ağın çıkış hatasının en düşük olduğu gözlemlenmiştir. Buradan hareketle bulunan YBS değeri Şekil 5.30’da elde edilen aralıkla örtüşmektedir. Sonuç olarak gerçek veri setinde ve değişken nöron sayılı YSA topolojilerinde Watts-Strogatz Küçük Dünya ağlarının performansının eğitim zamanını kısalttığı ve sınıflandırma hatasının en küçük elde edildiği gözlemlenmiştir.

### 5.5. Havalı Güneş Kolektörlerinin Termal Performansının Watts-Strogatz Küçük Dünya Ağları ile Tahmini

Düz ve çapraz plaka yüzeyine sahip iki farklı havalı güneş kolektörünün termal performanslarının deneysel olarak araştırılması E. Gedik tarafından hazırlanmıştır [105]. Önerilen havalı güneş kolektörlerinin deneysel bulguları M. Caner ve arkadaşları 3 katmanlı ve her katmanında sırasıyla 8, 20 ve 1 nöron bulunan bir düzenli YSA modeli kullanılarak termal modellerin termal performansları tahmin edilmeye çalışılmıştır [106]. Bu çalışmada ise, Watts-Strogatz Küçük Dünya ağı kullanarak 2 modelin termal performansı tahmin edilmiş ve önceden önerilen çalışmalarla karşılaştırılmıştır. İlk olarak Küçük Dünya ağlarını elde etmek için, 3 katmanlı ve her katmanda sırasıyla 8, 12, 1 nöron bulunan bir düzenli ağ topolojisi kullanılmıştır. Seçilen temel düzenli ağ yapısı Şekil 5.33’ te gösterilmiştir.



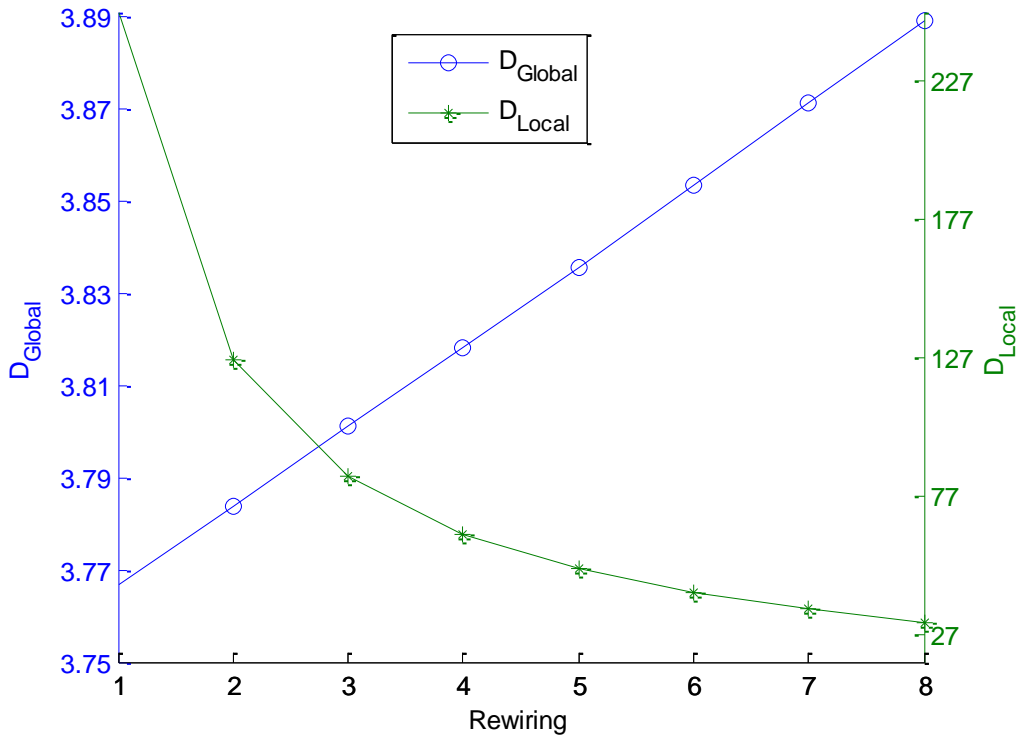
Şekil 5.33. Havalı güneş kolektörlerinin termal performansının tahmini için kullanılan düzenli ağ topolojisi.

Güneş kolektörlerinin 8 parametresi M. Caner ve arkadaşlarının yaptığı çalışmaya benzer şekilde giriş olarak kullanılmıştır. Şekil 5.33’te görüldüğü gibi; Giriş parametreleri olarak,  $T_i$ ,  $T_o$  sırasıyla giriş ve çıkış sıcaklıklarını,  $T_{sw}$ ; depo suyu



sıcaklığını,  $T_{amb}$ ; ortam sıcaklığını,  $T_s$ , Kolektör yüzey sıcaklığını,  $I$ ; güneş ışınım yoğunluğunu,  $Time$ , ölçüm zamanını ve model type ise kolektör model tipini ifade etmektedir. Çıkış parametresi ise Kolektör performansını ifade etmektedir,  $\eta$ .

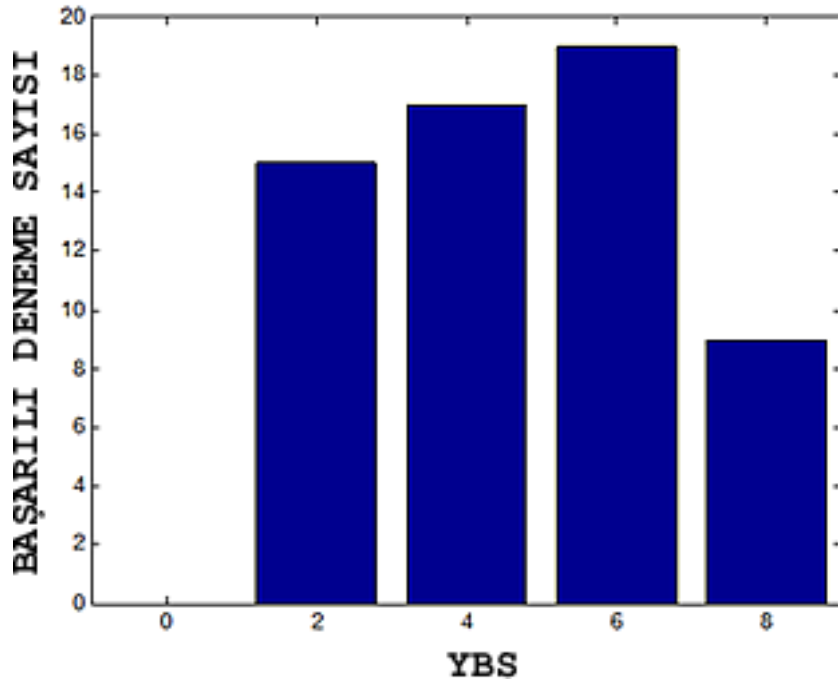
2. adım olarak seçilen 3 katmanlı mimariye Watt-Strogatz bağlantı yenileme metodu uygulanarak Küçük Dünya ağları YBS aralığı tespit edilmeye çalışılmıştır. Seçilen ağ topolojisinin (8-12-1) bağlı olamayan 8 farklı bağlantısı bulunduğundan YBS süreci 8 YBS değerine kadar devam etmiştir. Her bir bağlantı yenileme sürecinde  $D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmıştır. YBS değişimi ile elde edilen katsayı değişkenlerine ait  $D_{Global}$  ve  $D_{Lokal}$  değerleri Şekil 5.34'te gösterilmiştir.



Şekil 5.34. 8-12-1 İleri yönlü yapay sinir ağı için, Watts-Strogatz metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

Şekil 5.34' te görüldüğü gibi, YBS ile  $D_{Lokal}$  azalırken,  $D_{Global}$  artan bir davranış sergilemiştir. Bu bağlamda, her iki parametrenin küçük olduğu aralık araştırılmış ve  $D_{Global}$  değerinin artışı küçük oranda gerçekleşmiştir. 2 parametrenin kesişim noktası baz alındığında Küçük Dünya aralığı 2-8 arasında tespit edilmiştir.

3. adımda, eğitimin güvenilirliğini artırmak amacıyla her YBS ile elde edilen ağ topolojisi için, 10 kat Çapraz Doğrulama yöntemi kullanılmıştır. Bu bağlamda, 80 örnekli veri seti 0-1 arasında ölçeklendirilmiştir. Hazırlanan veri seti 10 parçaya bölünmüş ve rastgele bir parça test fazı için ayrılmış geriye kalan 9 parça eğitim için kullanılmıştır. Bu süreç bütün parçalar test ve eğitim için kullanılana kadar 10 kez tekrarlanmıştır. Eğitim sürecinde, eğitimi durdurmak amacıyla minimum eğitim hatası(MSE) 0.000001 olarak alınmıştır. Test fazında, (MAE, MSE) istatistik parametreleriyle test hatası hesaplanmıştır. buna ek olarak istatistik doğrulama için ( $R^2$ , RMSE) parametreleri hesaplanmıştır. her kat için elde edilen test parametrelerinin ortalaması alınarak ortalama ağın test hatası hesaplanmıştır. Her bir topoloji için 60 deneme yapılmış ve YBS ile bu denemelerde elde edilen başarılı deneme sayısının dağılımı Şekil 5.35' te gösterilmiştir.



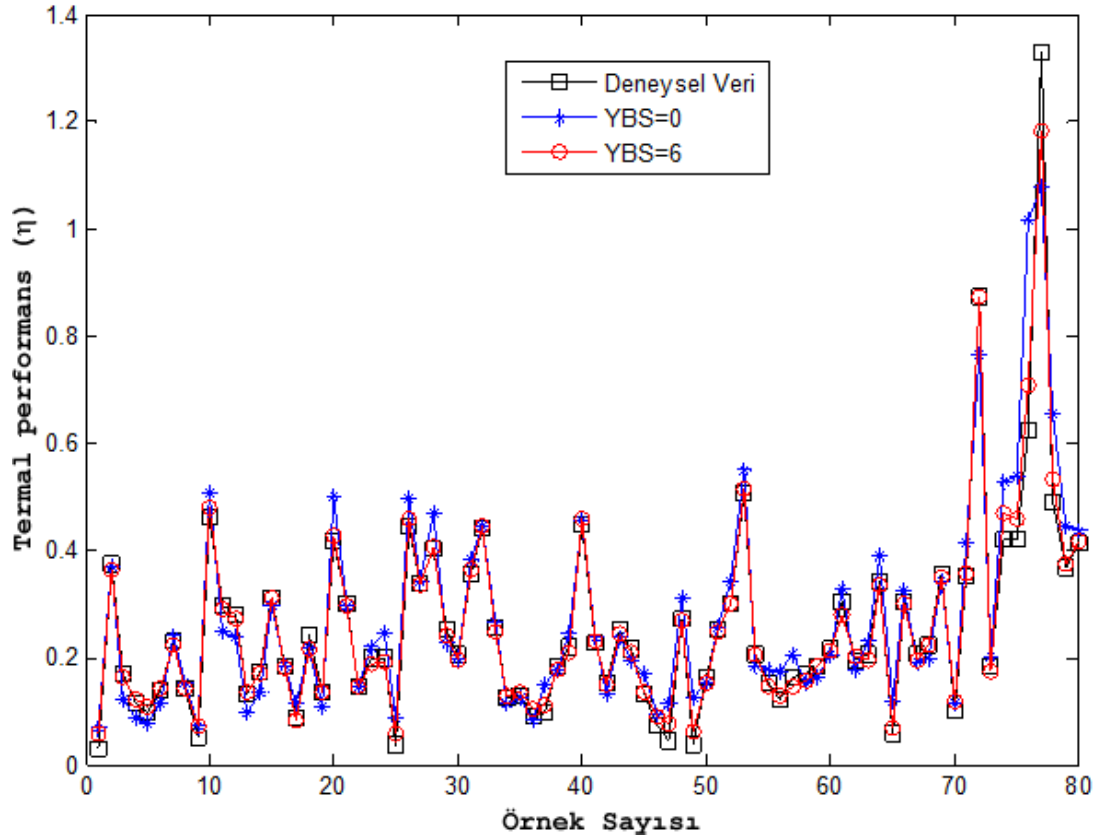
Şekil 5.35. 8-12-1 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.

Şekil 5.35' te görüldüğü gibi Küçük Dünya ağları için YBS aralığı 2-8 arasında gözlemlenmiştir. Sonuçlar Şekil 5.34'te elde edilen aralıkla da örtüşmektedir. 60 deneme içerisinde en küçük test hatasına sahip 51. deneme ele alınmıştır. 51. Denemeye ait test hatalarını ve istatistik doğruluk parametreleri Tablo 5.2' de gösterilmiştir.

Tablo 5.2. Havalı güneş kolektörlerinin termal performansları için YBS ile test parametrelerinin değişimi.

<b>YBS</b>	<b>R<sup>2</sup></b>	<b>MAE</b>	<b>MSE</b>	<b>RMSE (%)</b>
<b>0</b>	0.959341	0.049025	0.021994	5.922908
<b>2</b>	0.989063	0.02177	0.004267	2.911202
<b>4</b>	0.98624	0.025331	0.005196	3.183115
<b>6</b>	0.994048	0.017879	0.003085	2.297599
<b>8</b>	0.989019	0.022255	0.00381	2.86277

Tablo 5.2' de görüldüğü gibi en düşük test hata ve doğruluk parametreleri YBS=6 değeri için tespit edilmiştir. M. Caner ve arkadaşlarının [106] hesaplamış oldukları hata değerlerinden daha düşük hata değerleri daha az nöron sayısı ile elde edilmiştir. 51. Denemeden elde edilen düzenli ağ ve Küçük Dünya ağlarının termal performansının deneysel bulgularla karşılaştırılması Şekil 5.36' da gösterilmiştir.



Şekil 5.36. Düzenli ve Küçük Dünya ağlarının termal performanslarının deneysel bulgularla karşılaştırılması.

Şekil 5.36’ da görüldüğü gibi, Küçük Dünya ağın performansının, düzenliği ağa göre daha iyi olduğu gözlemlenmiştir. Bu bulgunun istatistik olarak anlamlı olup olmadığının araştırılması için, literatürde yaygın kullanılan bağımsız iki grup arası farkların t-testi kullanılmıştır [107]. Bu test iki grup ortalamaları karşılaştırılarak, aradaki farkın rastlantısal mı, yoksa istatistiksel olarak anlamlı mı olduğuna karar vermektedir. T-testi t ve p (iki yönlü anlamlılık) değerleri üzerinden doğruluk analizi yapmaktadır. Bu bağlamda bu iki parametre hesaplanmıştır. Analiz SPSS programı kullanılarak gerçekleştirilmiştir. Analizde İki yönlü anlamlılık seviyesi  $\alpha=0.01$  olarak alınmıştır. Elde edilen analiz sonuçları tablo 5.3’ de gösterilmiştir.

Tablo 5.3. Bağımsız iki grup arası t-testi analiz sonuçları.

	<b>Düzenli Ağ</b>	<b>Küçük Dünya Ağları</b>
<b>Mean</b>	23.1387	8.5608
<b>Std. Deviation</b>	37.6303	16.4435
<b>Std. Error Mean</b>	4.2072	1.8384
<b>N</b>	80	80
<b>Df</b>	108.11	
<b>T Value</b>	3.1751	
<b>P-Value</b>	0.0020	

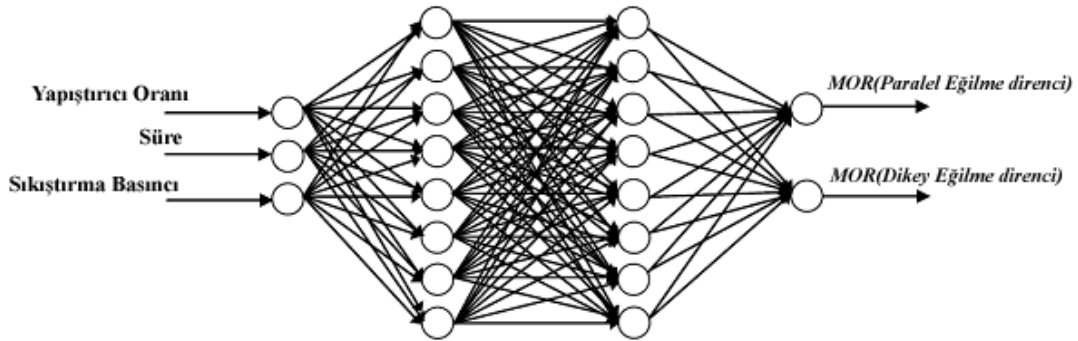
Tablo 5.3' te görüldüğü gibi analizden elde edilen iki yönlü anlamlılık değeri (p) 0.0020 olarak bulunmuştur.  $P < 0.01$  olduğu için  $H_0$  hipotezi  $\alpha = 0.01$  anlamlılık seviyesi için reddedilmiştir. Buna ek olarak 2 grup arasındaki farkın istatistiksel açıdan anlamlı olduğu görülmüştür. Sonuç olarak, Küçük Dünya ağı termal performansının düzenli ağdan daha iyi olduğu ve aradaki bu farkın rastlantısal olmadığı istatistiksel açıdan da vurgulanmıştır [108].

### **5.6. Yönlendirilmiş Yonga Levha (OSB) üretimde Eğilme Direnci parametrelerinin Watt-Strogatz Küçük Dünya Ağları ile tahmini**

Günümüzde, yönlendirilmiş levha üretiminde pek çok ağaç materyal kullanılmıştır. Bunlardan birisi ise Sarıçam (*Pinus sylvestris* L.) ağacıdır. 2008 yılında çalışmada, F. Yapıcı [109] Sarıçam ağacını yönlendirilmiş levha üretiminde kullanmış ve üretilen levhaların kopma modülü parametrelerini deneysel olarak elde etmiştir. Yapılan deneysel levha boyutları, 80mm uzunlukta, 20mm genişliğinde ve 0.7mm eninde seçilmiştir. Materyal üretiminde, yapışkan malzeme katılmadan önce, ahşap lifleri %3 nem oranına ulaşana kadar kurutulmuş, sonra 47% liquid phenol- formaldehyde resin içeren cilasız yapıştırıcı kurutulmuş ahşap liflerinin ağırlıklarına göre sırasıyla %3, 4.5 ve 6 oranlarında uygulamıştır. Sonraki aşamada, yapışkan sürülen materyallere 3, 5 ve 7 dakika süre ile 35, 40 ve 45 kg/cm<sup>2</sup> sıkıştırma basıncı uygulanmıştır. Levha taslaklarının oluşturulurken, orta ve dış tabakada kullanılan strandlerin oranı %40-60 oranında ve levha yoğunlukları 0.70 g/cm<sup>3</sup> olarak

uygulanmış ve 27 OSB paneli 56 x 56 x 1.2 cm ebatlarında üretilmiştir. Her bir panel 1'den 27'ye kadar numaralandırılmıştır.  $182 \pm 3^{\circ}\text{C}$ ' de paneller preslenmiştir [109].

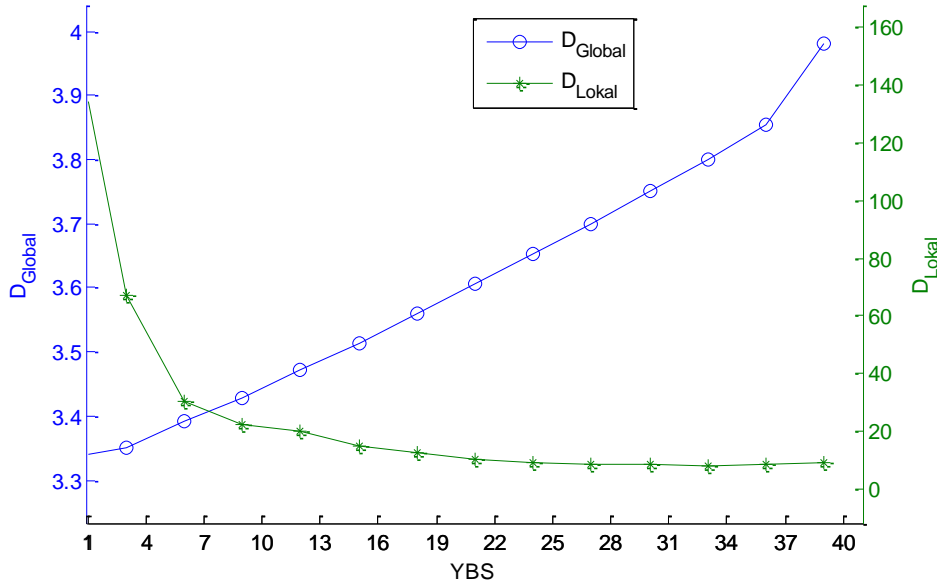
Presleme işleminden sonra, paneller stabil ağırlığa ulaşana kadar pres saçları arasında soğuya kadar bekletilmiştir. Tamamen soğuyan levhalar, sıcaklığı  $20 \pm 2^{\circ}\text{C}$  ve bağıl nemi  $65 \pm 5\%$  olan bir iklimlendirme dolabında 3 hafta boyunca iklimlendirilmişlerdir [110]. Her bir levha için 10 ayı eğilme direnci (MOR) deneyleri TS-EN 310 (1999)'da belirlenen yöntemlerle yapılmıştır [111]. Eğilme direnci (MOR) değerleri levha boyuna paralel yönde ve levha boyuna dik yönde ayrı ayrı elde edilmiştir [109]. F.yapıcı tarafından gerçekleştirilen bu deneysel çalışmada elde edilen veri seti benzer koşullarda Küçük Dünya ağları ile tahmin edilmeye çalışılmıştır. Bu bağlamda, 4 katmanlı bir düzenli YSA topoloji denemeler ışığında tespit edilmiştir. Kullanılan YSA topolojisi Şekil 5.37' de gösterilmiştir. Topoloji 3 girişli, 2 çıkışlı ve 2 ara katmandan oluşmakta ve ara katmanlarında 8 nöron bulunmaktadır. Ağ girdi olarak, Süre, Yapıştırıcı oranı ve Sıkıştırma basın. Değerlerini kullanarak, Paralel ve Dikey eğilme direnci (MOR) parametrelerini çıkış olarak hesaplamaktadır.



Şekil 5.37. Yönlendirilebilir levha üretiminde eğilme direnci parametrelerinin tahmini için önerilen düzenli ağ topolojisi(3-8-8-2).

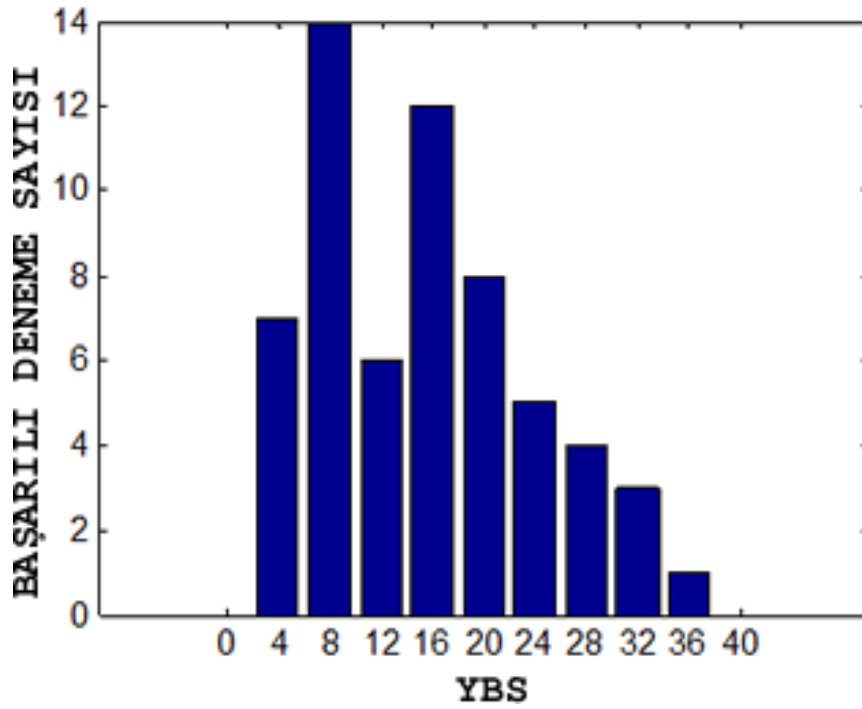
Deney sonunda 270 örnekli deneysel veri setinden, her bir panel ortalamaları alınarak 27 örnekli ortalama eğilme direnci parametreleri hesaplanıp YSA veri seti oluşturulmuştur. Düzenli YSA topolojisi belirlendikten sonra, ağın olası maksimum YBS değeri, giriş katmanı ile çıkış katmanı nöronları arası olası bağlantılar yok sayılarak hesaplanmış ve  $YBS_{Max}=40$  olarak bulunmuştur. Watts-Strogatz bağlantı yenileme metodu kullanılarak yeni topoloji elde edilmiş ve her bir işlem için  $D_{Global}$

ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmış ve süreç maksimum YBS değerine kadar devam etmiştir. Elde edilen  $D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları Şekil 5.38' de gösterilmiştir.



Şekil 5.38. 3-8-8-2 İleri yönlü yapay sinir ağı için, Watts-Strogatz metodu kullanılarak elde edilen YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

Şekil 5.38' de görüldüğü her iki eğrinin minimum değerlerinden Küçük Dünya ağlarının davranışının 8 ile 40 arasında gerçekleşeceği gözlemlenmiştir. Bu adımdan sonra ağların eğitimi aşamasına geçilmiştir. Burada, 10 kat Çapraz Doğrulama yöntemi kullanılarak ağın ezberleme sürecinin önüne geçilmiştir. Çapraz Doğrulama sürecinde, 27 örnekli veri seti [0-1] arasında ölçeklenmiş ve 10 parçaya bölünmüştür. Rastgele seçilen parça test için geriye kalan 9 parça eğitim için seçilmiştir. Sonra eğitim süreci başlatılmış ve minimum hata= 0.000001 olana kadar devam etmiştir. Seçilen test parçası ile ağ test edilmiş ve sırasıyla MSE, MAE, RMSE hata parametreleri ve  $R^2$  istatistiki doğruluk parametresi hesaplanmıştır. bu süreç 10 kez tekrar edilmiştir. Sonuç olarak hesaplanan test parametrelerinin ortalaması alınarak ağın genel test sonuçları elde edilmiştir. Her bir ağ topolojisi için Çapraz Doğrulama fazı 60 kez tekrar edilmiştir. 60 tekrar sonrası gözlemlenen YBS topolojisi için başarılı deneme sayısının dağılımı Şekil 5.39' da gösterilmiştir.



Şekil 5.39. 3-8-82 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.

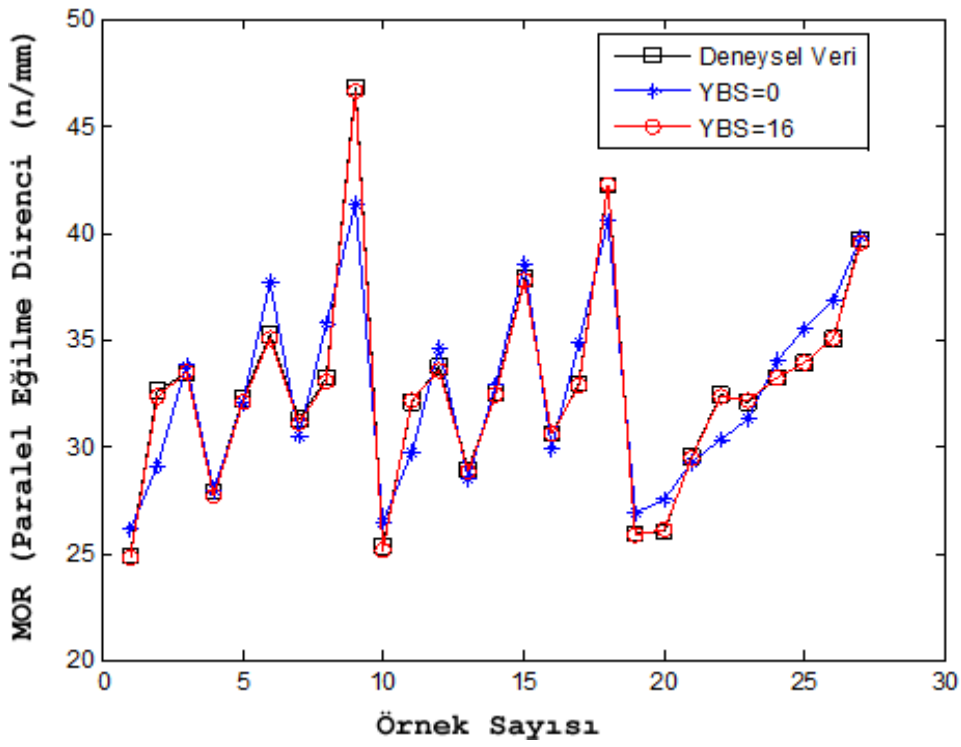
60 denemeden gözlemlenen başarılı deneme sayısı dağılımına bakıldığında, YBS aralığının 8-40 arasında yüksek ortalamaya sahip olduğu görülmüştür. Şekil 5.38’de önerilen aralıkla örtüşmektedir. Ağların eğitimi sonucunda 60 denemeden en düşük test hatasına sahip olan 37. deneme seçilmiştir. Seçilen denemenin test parametreleri tablo 5.4’ te gösterilmiştir. Tabloya bakıldığında en iyi test parametresi YBS=16 için elde edilmiştir.



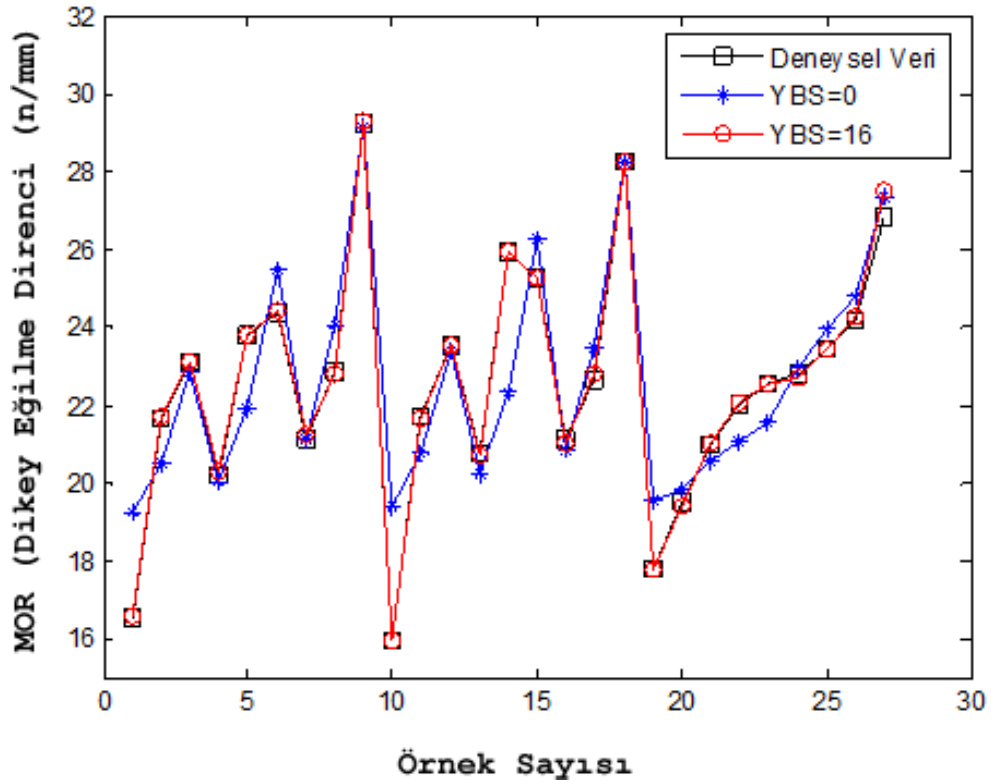
Tablo 5.4. Eğilme direnci tahmininde, YBS ile test parametrelerinin değişimi.

YBS	R <sup>2</sup> -Output 1	R <sup>2</sup> - Output2	MAE	MSE	RMSE (%)
0	0.981913	0.94994	0.022299	0.004031	5.127057
4	0.997913	0.982364	0.012101	0.001127	2.801207
8	0.998752	0.980487	0.009118	0.000658	2.098895
12	0.995861	0.977478	0.010825	0.001001	2.382575
16	0.998496	0.995746	0.006923	0.000364	1.583459
20	0.996022	0.991652	0.010226	0.000781	2.226508
24	0.996045	0.976741	0.011605	0.001094	2.582601
28	0.992838	0.984314	0.013337	0.001312	2.979785
32	0.995462	0.977063	0.014287	0.001307	3.100186
36	0.737677	0.97934	0.025947	0.012859	5.760594
40	0.986455	0.974657	0.016255	0.002324	3.730499

YBS=16 ağı için hesaplan dikey ve paralel eğilme direnci çıkışları Şekil 5.40' da gösterilmiştir. Şekil 5.40' da görüldüğü gibi, Küçük Dünya ağları performansının düzenli ağa göre daha iyi olduğu gözlemlenmiştir.



(a)



(b)

Şekil 5.40. Düzenli ve Küçük Dünya ağlarının eğilme direnci kestirim performansları (a) MOR (Paralel Eğilme Direnci), (b) MOR (Dikey Eğilme Direnci).

Şekil 5.40' ta görüldüğü gibi, Küçük Dünya ağı, düzenliği ağa göre daha iyi olduğu gözlemlenmiştir. Bu bulgunun istatistik olarak anlamlı olup olmadığının araştırılması için, literatürde yaygın kullanılan bağımsız iki grup arası farkların t-testi kullanılmıştır[107]. Düzenli ve Küçük Dünya ağları çıktıları kullanılarak iki metot arasında istatistiki açıdan fark olup olmadığı Tablo 5.4' te olduğu gibi yönlü anlamlılık seviyesi  $\alpha=0.01$  alınarak test edilmiş ve SPSS programı çıktıları Tablo 5.5' te gösterilmiştir.

Tablo 5.5. Düzenli ve Küçük Dünya ağı içint-testi sonuçları. (a) MOR (Paralel Eğilme Direnci), (b) MOR (Dikey Eğilme Direnci).

	<b>Düzenli Ağ</b>	<b>Küçük Dünya Ağları</b>
<b>Mean</b>	4.08059795	0.320180225
<b>Std. Deviation</b>	3.295172256	0.229358048
<b>Std. Error Mean</b>	0.634156196	0.044139977
<b>N</b>	27	27
<b>Df</b>	26.25	
<b>T Value</b>	5.9155	
<b>P-Value</b>	0.000003	

(a)

	<b>Düzenli Ağ</b>	<b>Küçük Dünya Ağları</b>
<b>Mean</b>	4.435303658	0.303957035
<b>Std. Deviation</b>	4.782524812	0.447958736
<b>Std. Error Mean</b>	0.920397329	0.086209699
<b>N</b>	27	27
<b>df</b>	26.46	
<b>T Value</b>	4.4691	
<b>P-Value</b>	0.000132	

(b)

Tablo 5.5' te görüldüğü gibi iki yönlü anlamlılık derecesi (p) her iki çıkış için sırasıyla 0.000003 ve 0.000132 olarak hesaplanmış ve bu değer anlamlılık seviyesinden küçüktür. Böylece H0 hipotezi reddedilmekte ve iki grup arasında istatistiki açıdan anlamlı bir fark olduğu gözlemlenmiştir. Buradan hareketle Küçük

Dünya ağlarının gerçek dünya problemlerinin çözümünde düzenli ağa göre başarısı istatistikî olarak da ortaya konulmuştur [108].

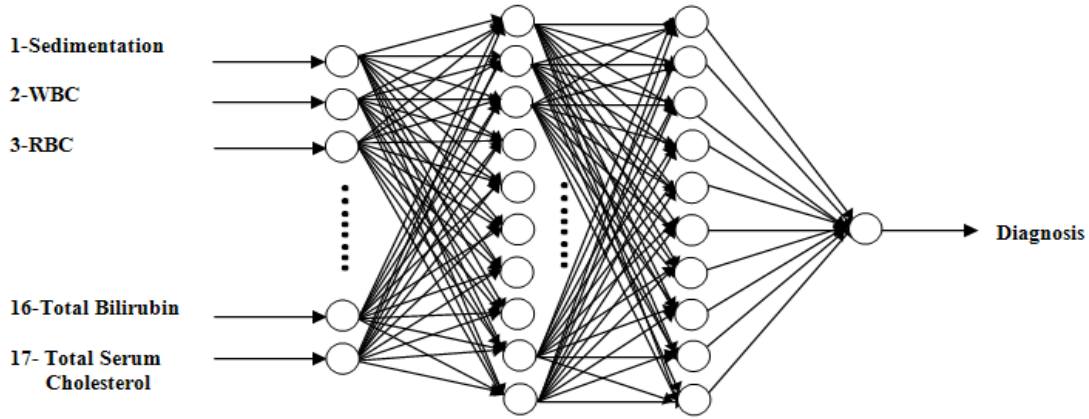
### 5.7. Şeker Hastalığı Teşhisinde Küçük Dünya Ağları Kullanılarak Bir Karar Destek Sistemi Oluşturulması

Klinik ortamlarda doktora yardım amacıyla, bilgisayar destekli akıllı karar destek sistemleri, günümüzde klinik teşhislerinde oldukça sık görünmeye başlanmıştır. Bu bağlamda oluşturulan karar destek sistemlerinde kullanılan ileri yönlü düzenli YSA topolojisi yerine Küçük Dünya ağları topolojisi kullanılmış ve sonuçları araştırılmıştır. Çalışmada diyabet hastalığı teşhisi için bir karar destek sistemi hazırlanmış ve kullanılacak veri seti, Türkiye’de bir devlet hastanesi Kulak, Burun ve Boğaz kliniğinden alınan 54 adet denekten alınan kan testi sonuçları ile hazırlanmıştır. Elde edilen kan testlerinden 17 adet deney parametresi diyabet hastalığı teşhisinde, girdi olarak seçilmiştir. Karar destek sisteminde kullanılan girdi parametreleri tablo 5.6’ da gösterilmiştir.

Tablo 5.6. Karar destek sisteminin kan testi girdi parametreleri.

Referans Özellikler	Referans Aralığı
Sedimentation Rate	(1-20)
White Blood Cell Count	(4,0-11,0)
Red Blood Cell Count	(4,00-6,20)
Hemoglobin	(11,0-18,0)
Hematocrit	(35,0-55,0)
Fasting Blood Glucose	(70-110)
Urea	(10-50)
Triglyceride Levels	(30-150)
High Density Lipoprotein	(35-75)
Very Low-Density Lipoprotein	(7,0-32,2)
Low-Density Lipoprotein	(0-130)
Mean Platelet Volume	(6-10)
Aspartate Aminotransferase	(10-50)
Alanine Aminotransferase	(5-45)
Direct Bilirubin	(0,00-0,30)
Total Bilirubin	(0,0-1,0)
Total Serum Cholesterol	(120-200)

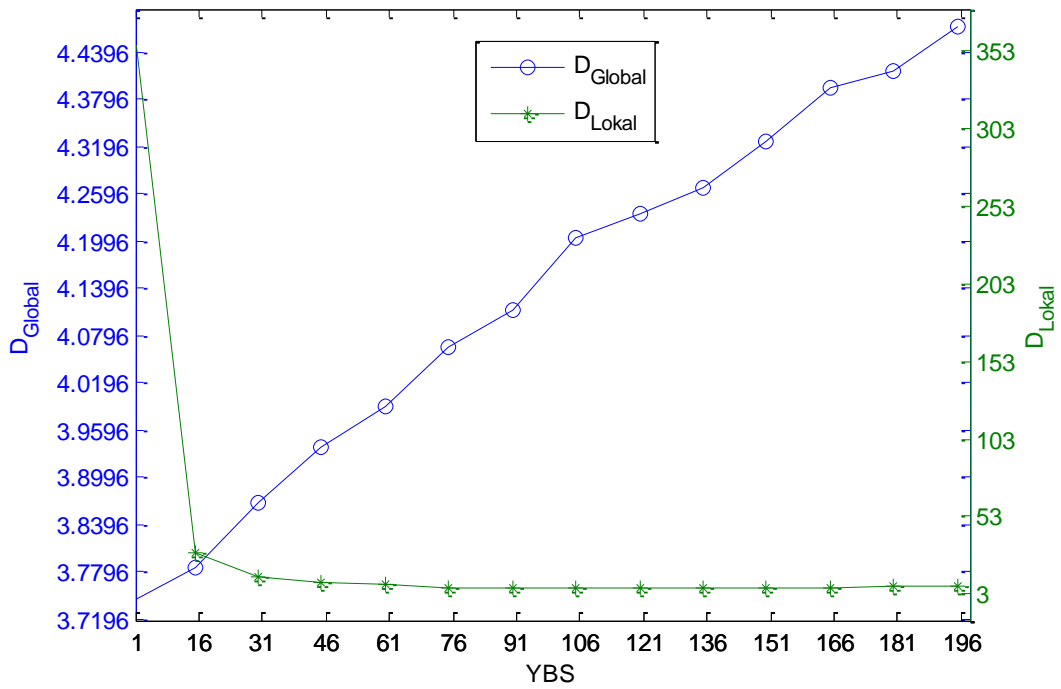
Bu girdi değerleri kullanılarak klinik ortamda doktor kararları dikkate alınarak, sistem çıkışı, hastanın diyabet hastası (1) veya olmadığına (0) karar vermektedir. 17 girişli ve 1 çıkışlı sistemde düzenli ağ yapısı için, 4 katmanlı ve ara katmanlarında 10 nöron bulunan bir YSA topolojisi belirlenmiştir. Temel alınan düzenli YSA topolojisi Şekil 5.41’ de gösterilmiştir.



Şekil 5.41. Karar destek sistemi düzenli ağ YSA topolojisi.

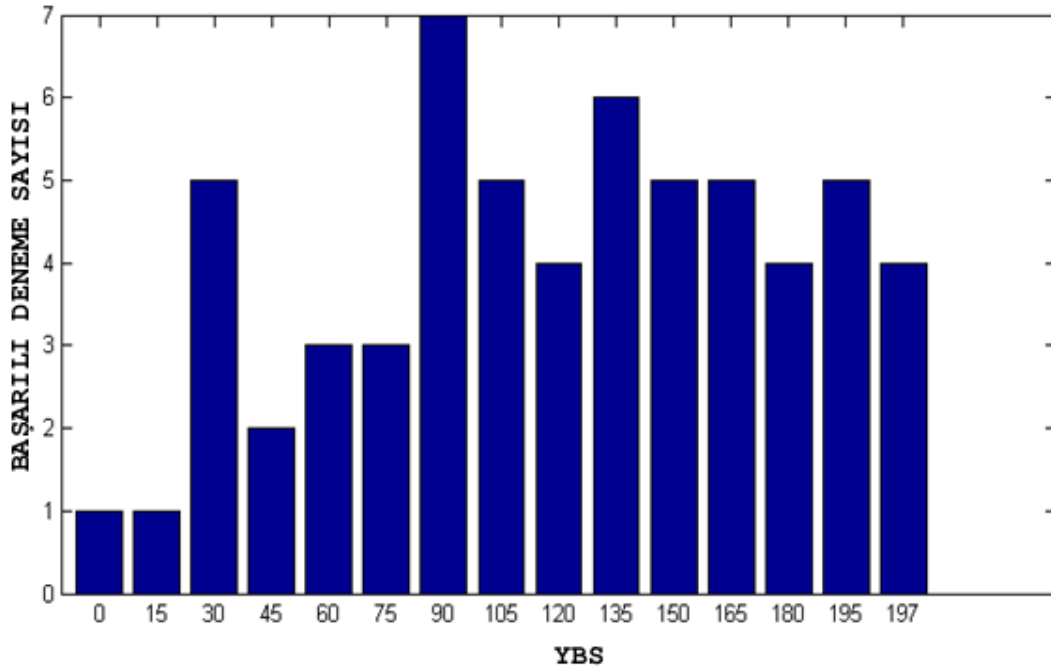
Küçük Dünya ağları topolojisini elde etmek amacıyla 4 katmanlı topolojinin maksimum YBS değeri 197 olarak hesaplanmıştır. buradan hareketle bağlantılar rastsal olarak Watts-Strogatz bağlantı yineleme metodu kullanılarak yenilenmiştir. Her yenilen bağlantıdan sonra  $D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmıştır. YBS değişimi ile elde edilen parametrelere ait  $D_{Global}$  ve  $D_{Lokal}$  değerleri Şekil 5.42’ de gösterilmiştir.

$D_{Global}$  ve  $D_{Lokal}$  bağlantı uzunluk katsayıları hesaplanmıştır. YBS değişimi ile elde edilen katsayı değişkenlerine ait  $D_{Global}$  ve  $D_{Lokal}$  değerleri Şekil 5.34’ te gösterilmiştir.



Şekil 5.42. 17-10-10-1 İleri yönlü yapay sinir ağı için, YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişimi.

Şekil 5.42' de görüldüğü gibi YBS artarken  $D_{Global}$  artmakta ve  $D_{Lokal}$  azalmaktadır. Bu doğrultuda kesişim noktasının  $YBS \approx 17$  seviyelerinde her iki değerinde en küçük oldukları durum gözlemlenmiştir. Küçük Dünya ağlarının olası aralığı 17-197 olarak öngörülmüştür. Aralık tespiti sonrasında, düzenli ağ ile belirli YBS değerlerine sahip topolojilerin eğitilmesi süreci gerçekleştirilmiştir. Burada 54 örnekle veri seti kullanılarak 3-Kat Çapraz Doğrulama metodu kullanılmıştır. veri seti 3 parçaya bölünmüş ve rastgele bir parça test, geriye kalan 2 parça eğitim ve validation için kullanılmıştır. Bu süreç 3 kez tekrar edilmiş ve bütün veri seti hem eğitim hemde test fazında kullanılmıştır. böylece ezberleme probleminin önüne geçilmiş ve ağın eğitiminin doğruluğu elde edilmiştir. Eğitim evresinde her bir kat için MAE, MSE, RMSE,  $R^2$  doğruluk ve istatistiksel parametreleri hesaplanmıştır. son olarak hesaplanan hata ve istatistiksel doğrulama parametrelerinin ortalaması eşitlik 2.22' de ifade edildiği gibi hesaplanmıştır. Her bir ağ topolojisi için bu süreç 60 defa tekrar edilmiş ve her bir YBS topolojisi için başarılı deneme sayısının dağılımı Şekil 5.43' te gösterilmiştir.



Şekil 5.43. 17-10-10-1 nöron ağının, 60 deneme için, başarılı deneme sayısının YBS ile dağılımı.

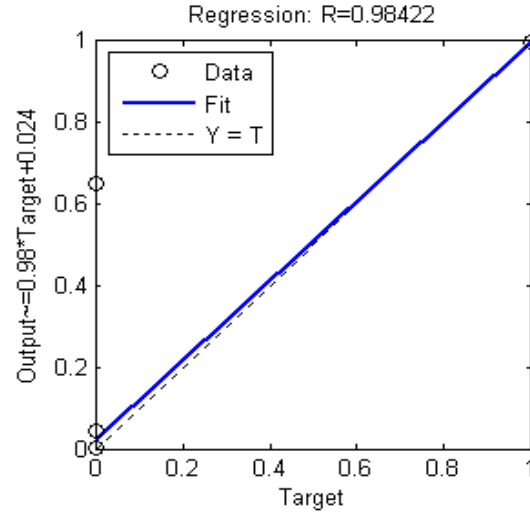
Şekil 5.43' te görüldüğü gibi, YBS aralığının 17-197 arasında yüksek ortalamaya sahip olduğu görülmüştür. Bu Şekil 5.42' de önerilen aralıkla örtüşmektedir. Ağların eğitimi sonucunda 60 denemeden en düşük test hatasına sahip olan 25. deneme seçilmiştir. Seçilen denemenin test parametreleri tablo 5.7' de gösterilmiştir. Tabloya bakıldığında YBS=135 değeri için en iyi ağ performansı elde edilmiş ve bu Küçük Dünya ağının sınıflandırma performansı %100 olarak elde edilmiştir.

Tablo 5.7. Diyabet hastalığı teşhisi için YBS tabanlı hata analiz sonuçları

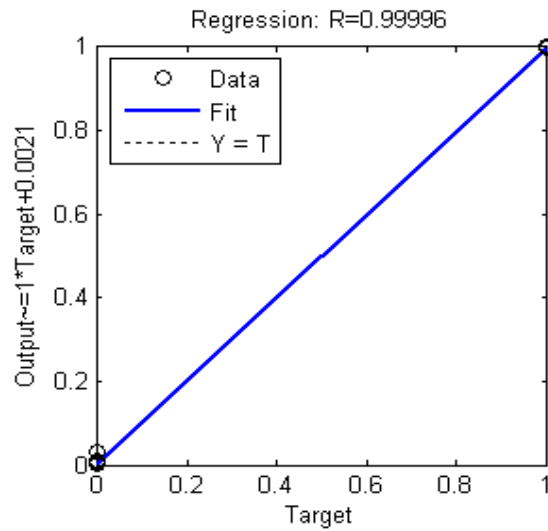
<b>YBS</b>	<b>R<sup>2</sup></b>	<b>MAE</b>	<b>MSE</b>	<b>RMSE(%)</b>
<b>0</b>	0.88754	0.063085	0.43627	21.56413
<b>5</b>	0.81071	0.099507	0.68705	27.49494
<b>30</b>	0.77633	0.094931	0.7593	28.33638
<b>45</b>	0.8932	0.072252	0.44765	20.3199
<b>60</b>	0.78262	0.109184	0.83569	29.89853
<b>75</b>	0.81252	0.095185	0.72828	24.47509
<b>90</b>	0.96306	0.021832	0.16576	8.485121
<b>105</b>	0.87541	0.070502	0.46492	22.34224
<b>120</b>	0.93651	0.048899	0.28896	15.60248
<b>135</b>	0.98803	0.017366	0.0595	5.512432
<b>150</b>	0.92405	0.042553	0.33595	16.878
<b>165</b>	0.81009	0.111459	0.75274	28.17579
<b>180</b>	0.89682	0.060073	0.45393	14.78806
<b>195</b>	0.79138	0.107068	0.73395	26.4395
<b>197</b>	0.71783	0.140057	1.08049	32.59111

Yapılan analizlere ek olarak düzenli ağ ile Küçük Dünya ağlarının karşılaştırmalı korelasyon analizi yapılmış ve deneysel veriler ile ağ çıktıları arasındaki korelasyon incelenmiştir. Böylece tüm veri seti için ağın başarımlı güvenilirliği elde edilmeye çalışılmıştır. Bu doğrultuda elde edilen ağların korelasyon katsayısı(R) hesaplanmıştır. Korelasyon katsayısının 1'e yakın olması, hesaplanan çıkış ile deneysel veriler arasındaki yüksek orandaki doğruluğu ifade etmektedir. 0'a yakın olması ise düşük korelasyonu gösterip doğru genelleme yapamayan bir ağı işaret etmektedir. Elde edilen karşılaştırmalı sonuç Şekil 5.44' te gösterilmiştir.





(a)



(b)

Şekil 5.44. Tüm veri seti için düzenli ve Küçük Dünya ağlarının korelasyon analizi (a) Düzenli ağ, (b) Küçük Dünya ağları.

Toplamda, Küçük Dünya ağlarının düzenli ağdan daha iyi sınıflandırma yaptığı gözlemlenmiştir. Buna ek olarak, Küçük Dünya ağları çıkışının düzenli ağ çıkışına göre daha yüksek korelasyona ( $R=0.999$ ) ve daha düşük çıkış hata değerlerine sahip olduğu gözlemlenmiştir. Bunlar ( $MAE=0.017366$ ), ( $MSE=0.059504$ ) ve ( $RMSE \cong \%5$ ) şeklinde hesaplanmıştır. Bu çalışmada, literatürde olmayan bir tıbbi karar destek sistemi ortaya konulmuştur. Böylelikle bundan sonraki çalışmalarda Küçük Dünya ağlarının etkili bir YSA metodu olarak, tıbbi sınıflandırma sisteminde yüksek verim amaçlı kullanılabileceği vurgulanmıştır.

## BÖLÜM 6. SONUÇ VE ÖNERİLER

### 6.1. Sonuçlar

Bu çalışmada, yapay sinir ağı uygulamalarının temel sorunlardan birisi optimum ağ topolojisinin seçimidir. Bu soruna kesin bir çözüm üretmek amacıyla, günlük yaşamımızda yeni bir kavram olan Küçük Dünya ağları modeli kullanılmıştır. Bu bağlamda bağlantı yenileyerek ağ davranışını kuvvetlendiren Küçük Dünya ağlarının farklı bağlantı metotları ile İYYSA modelleri hazırlanmıştır. Farklı alanlardan seçilen gerçek yaşam problemlerinin çözümünde ise hazırlanan Küçük Dünya YSA modeli etkin bir model olarak önerilmiştir.

İYYSA performansının artırılması pek çok nedene bağlı olmakla birlikte, son yıllarda en önemli faktörün topolojik etkenler olduğu ortaya konulmuştur[1]. Bu gerekçeden hareketle, daha önce farklı topolojili YSA modelleri önerilmiştir. Buna paralel olarak Küçük Dünya ağları kavramı da farklı bir ağ topolojisi olarak karmaşık ağ çalışmalarında ön plana çıkmıştır. Çalışmada, Karesel ve Değişken boyutlu ileri yönlü YSA mimarisi temel alınarak Küçük Dünya ağları; Watts-Strogatz, Neman – Watts ve Rastasal bağlantı yenileme yöntemleri ile elde edilmeye çalışılmıştır. Bu bağlamda, literatürden farklı olarak, geleneksel değişken boyutlu YSA topolojisi için Küçük Dünya ağları, bağlantı uzunluk katsayılarıyla ( $D_{Global}$  ve  $D_{Lokal}$ ) elde edilmeye çalışılmıştır. Kullanılan Küçük Dünya ağı metoduna göre ağın maksimum YBS değeri bir formül önerilerek hesaplanmıştır. Seçilen metoda göre ağın bağlantıları ya koparılıp değiştirilmiş yada koparılmadan rastgele yeni bağlantıların eklenmesiyle değiştirilmiştir. Her bağlantı değişiminde,  $D_{Global}$  ve  $D_{Lokal}$  katsayıları c# programlama dili yardımıyla hazırlanan arayüzde hesaplanmıştır. Süreç maksimum YBS değerine kadar devam etmiştir. YBS ile  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının değişim grafikleri çizilmiştir. Bu grafikler yardımıyla Küçük Dünya ağı YBS aralığı tespit edilmiştir. Watts-Strogatz ve Rastasal Küçük Dünya ağı modelleri için YBS

aralığının,  $D_{Global}$  ve  $D_{Lokal}$  eğrilerinin kesişim noktasından başlayarak maksimum YBS'ye kadar devam ettiği gözlemlenmiştir. Newman-Watts Küçük Dünya ağı modeli YBS aralığı ise, yaklaşık  $D_{Global}$  ve  $D_{Lokal}$  eğrilerinin kesişim bölgesinden başlayarak, belirli bir  $D_{Global}$  değerine kadar devam ettiği görülmüştür. Bu modeller karşılaştırıldığında görülmüştür ki; Newman-Watts Küçük Dünya ağı modeli YBS aralığının, Watts-Strogatz ve rastsal Küçük dünya ağı modellerinin YBS aralığına oranla daha küçük olduğu gözlemlenmiştir.

Çalışmada Küçük Dünya ağları ve düzenli YSA ağı arasındaki performans bilgisi, elde edilen topolojilere standart geri yayılım ve momentum geri yayılım algoritması uygulanarak elde edilmeye çalışılmıştır. Bu bağlamda her iki modelinde farklı gerçek yaşam problemlerinin çözümünde öğrenim performansları karşılaştırmalı olarak ortaya konulmuştur. Her bir topoloji için öğrenme süreci 20-60 arasında değişen tekrarlar yapılarak elde edilmiş ve tekrarlar sonucunda en iyi performansa sahip topoloji seçilmiştir. Buradan hareketle denemeler sonrası, her denemenin başarılı topolojisi tespit edilerek, topolojilerin YBS ile başarılı deneme sayıları dağılımı elde edilmiştir. Bu dağılımla,  $D_{Global}$  ve  $D_{Lokal}$  katsayıları değişim grafiği ile tespit edilen Küçük Dünya ağ aralığı karşılaştırılarak doğruluk analizi yapılmıştır. Tüm uygulamalar için bu aralığın tutarlı olduğu yapılan denemeler sonucunda da ortaya konmuştur.

Öğrenim sürecine bakıldığında, Watts-strogatz ve rastsal bağlantı yenileme yöntemiyle elde edilen Küçük Dünya ağlarının öğrenim performansı düzenli ağa göre daha iyi olduğu gözlemlenmiştir. Ancak veri seti örnek sayısı artırıldığında, Küçük Dünya ağlarının performansının azaldığı gözlemlenmiştir. Newman-Watts Küçük Dünya ağının performansının geleneksel düzenli ağa göre yine daha iyi olduğu gözlemlenmiş ve bu ağ topolojisinin performansının veri setinden bağımsız olduğu gözlemlenmiştir.

Algoritmik modelleme için, Visual studio .NET platformunun Visual C# programlama dili yardımıyla bir görsel arayüz programı tasarlanmıştır. Uygulamalar bu program yardımıyla gerçekleştirilmiştir. Tez süreci içerisinde tasarlanan arayüz geliştirilerek kullanıcı dostu haline getirilmiştir.

Bu çalışmada, önceki çalışmalardan farklı olarak Küçük Dünya ağları öğrenme sürecinde N kat Çapraz Doğrulama yöntemi kullanılarak ağ davranışlarının veri setinden bağımsız olarak incelenmesi sağlanmış ve ağın ezberleme sürecinin önüne geçilmiştir. Arayüz programı, istenilen değerde çapraz doğrulama uygulaması gerçekleştirilecek şekilde tasarlanmıştır. Seçilen uygulamalarda ise 10 ve 3 kat Çapraz Doğrulama yöntemi kullanılmıştır. Bu yöntemle Küçük Dünya ağlarının öğrenmedeki topolojik etkisi net bir şekilde ortaya konulmuştur.

Düzenli ağ ile Küçük Dünya ağlarının öğrenme sürecindeki farkın anlamlı olup olmadığının tespiti için, istatistiksel t-testi yapılmıştır. İstatistiksel t-testi için her iki ağ topolojisinin ortalama test hataları baz alınarak, önem seviyeleri hesaplanmıştır. Elde edilen önem seviyeleri göstermiştir ki, iki ağ topolojisi arasındaki fark istatistiksel açıdan anlamlıdır. Buradan hareketle, Küçük Dünya ağları topolojisinin istatistiksel olarak da düzenli ağ modeline göre daha iyi bir model olduğu literatürde ilk defa ortaya konulmuştur. Böylece ağların öğrenim süresince elde ettiği bulguların rastlantısal olmadığı gösterilmiştir.

## 6.2. Öneriler

Bu tezde Küçük Dünya ağları topolojisinin ileri yönlü YSA performansını geliştiren daha iyi bir model olduğu vurgulanmış ve Küçük Dünya ağı modelinin geleneksel İYYSA ağlarının yerine kullanılabilirliği ortaya konulmuştur. Buna bağlı olarak, bundan sonra farklı YSA topolojileri içinde Küçük Dünya modeli uygulanarak performans araştırmaları yapılabilir veya yeni topolojik çalışmalarla yeni bir YSA modeli tasarlanabilir. Buna ek olarak, Küçük Dünya ağları hızlı sonuca ulaşmak istenilen YSA çalışmalarında, geleneksel ağ yerine kullanılabilir.

Çalışmada, belirli bilimsel alanlardan seçilen problemler, Küçük Dünya ağları ile çözüme kavuşturulmuştur. İstenildiğinde daha farklı bilimsel alanlarda da Küçük Dünya ağları modeli uygulanarak yüksek başarımlar elde edilebilir. Çalışmada kullanılan momentum geri yayılım algoritması yerine Levenberg–Marquardt geri

yayımla algoritması kullanılarak Küçük Dünya ağlarının öğrenim performansı daha da geliştirilebilir.

Küçük Dünya ağlarının aralık tespitinde kullanılan  $D_{Global}$  ve  $D_{Lokal}$  katsayılarının yerine farklı bir Küçük Dünya ağı yaklaşımı kullanılarak, Küçük Dünya ağları elde edilebilir. Yada istenilen alana uygulanabilirliği araştırılabilir. Sonuç olarak çalışmanın kapsamı bakımından Küçük Dünya ağlarının yapay zeka alanında etkileri araştırılmıştır. Bundan farklı olarak, özellikle nöronal ağların kullanıldığı bilim dallarında da Küçük Dünya ağları araştırılabilir ve geliştirilebilir niteliktedir.

## KAYNAKLAR

- [1] ROJAS, R., Neural networks – a systematic introduction, Springer: Berlin, 1996.
- [2] <http://www.cs.utsa.edu/~bylander/cs6243/neural-networks.pdf>, Mayıs, 2012.
- [3] HAYKIN, S., Neural networks: A comprehensive foundation, Upper Saddle River, NJ: Prentice Hall, 1994.
- [4] MILGRAM, S., The small world problem, Psychology Today 1967; pp. 60-67.
- [5] WANG X.F., CHEN G., Complex networks: small-world, scale-free and beyond, Circuits and Systems Magazine, IEEE 2003; 3(1):6-20.
- [6] ERDOS, P., RENYI, A., On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci., 5, 17-60, 1959.
- [7] WATTS, D., STROGATZ, S., Collective dynamics of small world networks, Nature 1998; pp. 440-442.
- [8] NEWMAN M.E.J., WATTS D.J., Scaling and percolation in the small-world network model, Phys. Rev. E 1999; 60:7332-7342.
- [9] THIPIREDDY, S. B., Models for evolution and joining of small world networks, M. Sc. Thesis, Jawaharlal Nehru Technological University, 2007.
- [10] BARABÁSI, A.L., ALBERT, R., JEONG, H., Mean-field theory for scale-free random networks, Physica A 1999; 272:173–187.
- [11] GANESH, A., XUE, F., On the connectivity and diameter of small-world Networks, Adv. Appl. Probab. 2007;39:853–863.
- [12] SPORNS O., CHIALVO, D.R., KAISER, M., HILGETAG, C.C., Organization, development and function of complex brain networks, Trends in Cognitive Sciences 2004; 8:418-425.
- [13] BASSETT, D.S., BULLMORE, E., Small-world brain networks, Neuroscientist 2006; 12:512–523.

- [14] ÖZER, M., PERC, M., UZUNTARLA, M., Controlling the spontaneous spiking regularity via channel blocking on newman-watts networks of hodgkin-huxley neurons, *EuroPhysics Letters (EPL)* 2009; 86(4):40008.
- [15] ÖZER, M., PERC, M., UZUNTARLA, M., Stochastic resonance on newman-watts networks of hodgkin-huxley neurons with local periodic driving, *Physics Letters A* 2009; 373:964-968,
- [16] ÖZER, M., UZUNTARLA, M., KAYIKCIOGLU, T., GRAHAM, L.J., Collective temporal coherence for subthreshold signal encoding on a stochastic small-world hodgkin-huxley neuronal network, *Physics Letters A* 2008; 372(43):6498-6503.
- [17] ÖZER, M., UZUNTARLA, M., Effects of the network structure and coupling strength on the noise-induced response delay of a neuronal network, *Physics Letters A* 2008; 372(25):4603-4609.
- [18] WATTS, D.J., Networks, dynamics, and the small-world phenomenon, *American Journal of Sociology* 1999; 105(2):493–527.
- [19] WATTS, D.J., Six degrees: the science of a connected age, W. W. Norton & Company, 2003.
- [20] MARCHIORI, M., LATORA, V., Harmony in the small world, *Physica A* 2000; 285:539-546.
- [21] LATORA, V., MARCHIORI, M., Efficient behavior of small-world networks, *Phys. Rev. Letters* 2001; 87(19):198701.
- [22] LATORA, V., MARCHIORI, M., Is the boston subway a small-world network?, *Physica A*, 2002; 314:109–113.
- [23] LATORA, V., MARCHIORI, M., Economic small-world behavior in weighted networks, *Eur. Phys. J. B.* 2003; 32:249-263.
- [24] SIMARD, D., NADEAU, L., KROGER, H., Fastest learning in small-world neural networks, *Physics Letters A* 2005; 336:8-15.
- [25] YANG, S., LUO, S., LI, J., Building multi-layer small world neural network, *Lecture Notes in Computer Science Series*, Springer:Berlin, 2006; 3971: 695-700.
- [26] BASSETT, D.S., BULLMORE, E.T., Small-world brain networks, *Neuroscientist* 2006; 12:512–523.
- [27] YANG, S., LUO, S., LI, J., An extended model on self-organizing map, *ICONIP 1* 2006; 4232:987-994.

- [28] FORTNEY, K., PAHLE, J., DELGADO, J., OBERNOSTERER, G., SHAH, V., WOJNOWICZ M., QAISAR, N., Effects of simulated brain damage on small-world neural networks, Computational Cognitive Neuroscience Conference, San Diego, CA, 2007.
- [29] LI SHOU-WEI, Analysis of contrasting neural network with small-world network, International Seminar on Future Information Technology and Management Engineering 2008; 2(3):1098-1113.
- [30] PALLAVER, T., KROGER, H., PARIZEAU, M., Biological principles in self-organization of young brain - viewed from kohonen model, AIP Conf. Proc. 2008; 1018:207-214.
- [31] JIANG, F., BERRY, H., SCHOENAUER, M., The impact of network topology on self-organizing maps, GEC Summit 2009;247-254.
- [32] XIAOHU, L., XIAOLING, L., JINHUA, Z., YULIN, Z., MAOLIN, L., A new multilayer feed forward small-world neural network with its performances on function approximation, Proceedings of IEEE International Computer Science and Automation Engineering (CSAE) 2011; 3:353-357.
- [33] MC CULLOCH, W.S., PITTS, W.H., A logical calculus of the ideas immanent in neural nets, Bull. Math. Biophys 1943; 5:115-133.
- [34] HEBB, D.O., The organization of behavior, John Wiley Sons, New York-USA, 1949.
- [35] ROSENBLATT, F., Principles of neuradynamics, Spartan Books, New York-USA, 1959.
- [36] ROSENBLATT, F., Principles of neuradynamics, perceptrons and the theory of brain,mechanisms, Spartan Books, Washington DC-USA, 1962.
- [37] WIDROW, B., HOFF, M., Adaptive switching circuits, IRE Wescon Convention Record 1960; 4:96-104.
- [38] ANDERSON, C., Learning problem solving with multilayer connectionist systems, Ph.D. Thesis, University of Massachusetts, 1986.
- [39] KOHONEN, T., Correlation matrix memories, IEEE Transactions on Computers 1972; 21:353-359.
- [40] MINSKY, M.L., PAPERT, S.A., Perceptron, expended edition, Cambridge, MIT Pres., London-UK, 1969.
- [41] GÜVEN A., Göze ait elektrofizyolojik sinyaller kullanılarak yapay sinir ağları destekli bazı göz hastalıklarının teşhisi, Doktora Tezi, Erciyes Üniversitesi, Kayseri, 2005.



- [42] HOPFIELD, J.J., Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. 1982; 79:2554-2558.
- [43] GROSSBERG, S., The adaptive self-organization of serial order in behavior: Speech, language, and motor control, In E.C. Schwab and H.C. Nusbaum (Eds.), Pattern recognition by humans and machines, Vol. 1: Speech perception, Academic Press, 187-294,1986.
- [44] WESTMACOTT, J., An artificial neural network for robust shape recognition in real time, M. Sc. Thesis, The University of South Africa, 2000.
- [45] KOHONEN, T., Self-organization and associative memory, Springer Verlag, Berlin, 1984.
- [46] RUMELHART, D.E., MC CLELLAND, J.L., Parallel distributed processing: explorations in the microstructure of cognition, MIT Press, Cambridge,1986.
- [47] <http://www.neuralpower.com/images/tecnology/figure1.gif>, Mayıs, 2012.
- [48] ÖZTEMEL, E., Yapay Sinir Ağları, Papatya Yayıncılık, 2003.
- [49] KRÖSE, B., VAN DER SMAGT, P., An introduction to neural networks, Eight Ed., The University of Amsterdam, 1996.
- [50] CARPENTER, G.A., GROSSBERG, S., The art of adaptive pattern recognition by a self organizing neural network, Computers 1988; 1:77-88.
- [51] HINTON, G.E. AND SEJNOWSKI, T.J., Optimal perceptual inference, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition 1983; pp. 448-453.
- [52] KOHONEN, T., Learning vector quantization, The Handbook of brain Theory and Neural Networks, MIT Press, Cambridge, 1995.
- [53] MASTERS, T., Practical neural network recipes in c++, San Diego: Academic Press., 1993.
- [54] BAYIR, F., Yapay sinir ağları ve tahmin modellemesi üzerine bir uygulama, Y. Lisans Tezi, İstanbul Üniversitesi, 2006.
- [55] YAVUZ, (ÇEKİRDEK) S., Performans artırmaya yönelik paralel mimarilerin yapay sinir ağları yaklaşımı ile değerlendirilmesi, Doktora Tezi, Yıldız Teknik Üniversitesi, 2006.
- [56] HEBB, D.O. The organization of behavior, New York: Wiley & Sons, 1949.

- [57] ÖĞÜTÜCÜ, O., Yapay sinir ağları ile sistem tanıma, Y. Lisans Tezi, İstanbul Teknik Üniversitesi, 2006.
- [58] HOPFIELD, J.J., Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the USA* 1982; 79(8);2554–2558.
- [59] DEDE, G.,Yapay sinir ağları ile konuşma tanıma, Y. Lisans Tezi, Ankara Üniversitesi, 2008.
- [60] BİLGİN, S., Kalp hızı değişkenliğinin dalgacık dönüşümü ve yapay sinir ağları kullanılarak analizi, Doktora Tezi, Sakarya Üniversitesi, 2008.
- [61] GEISSER, S.,The Predictive sample reuse method with applications, *Journal of the American Statistical Association* 1975; 70(350):320-328.
- [62] STONE, M., Cross-validation choice and assessment of statistical predictions (with Discussion), *Journal of the Royal Statistical Society* 1974; 36:111–147.
- [63] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., *The Elements of statistical learning*, New York: Springer-Verlag, 2001.
- [64] CLARKE, B., FOKOUE, E., ZHANG, H. H., *Principles and theory for data mining and machine learning*, Springer Series in Statistics, 2009.
- [65] NEWMAN, M.E.J., The structure and function of complex networks, *SIAM Review* 2003; 45:167-256.
- [66] ALBERT, R., BARABASI, A.L., Statistical mechanics of complex networks, *Reviews of Modern Physics* 2002; 74:47-97.
- [67] DOROGOVTSSEV, S.N., MENDES, J.F.F., Evolution of networks, *Advances in Physics* 2002; 51:1079-1187.
- [68] BOLLOBÁS, B., Mathematical results on scale-free random graphs, In *Handbook of Graphs and Networks* (ed. by S. Bornholdt, H. G. Schuster), Wiley/VCH,2003; pp. 1–34.
- [69] KLEINBERG, J., Complex networks and decentralized search algorithms, *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.
- [70] KAUFFMAN, S.A., Metabolic stability and epigenesis in randomly constructed genetic nets, *Journal of Theoretical Biology* 1969; 22(3):437–467.
- [71] NEWMAN, M. E. J., Scietific collaboration networks. I. network construction and fundamental results, *Physical Review E* 2001; 4(1):016131/1–016131/8.

- [72] KOPELL, N., ERMENTROUT, G.B., Symmetry and phaselocking in chains of weakly coupled oscillators, *Communications in Pure and Appl. Math.* 1986; 39:623–660.
- [73] KURAMOTO, Y., *Chemical oscillators, waves, and turbulence*, Springer:Berlin, 1984.
- [74] MIROLLO, R.E., STROGATZ, S.H., Synchronization of pulse-coupled biological oscillators, *SIAM J. Appl. Math.* 1990; 50:1645–1662.
- [75] STROGATZ, S.H., From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators, *Physica D* 2000; 143:1–20.
- [76] WINFREE, T., Biological rhythms and the behavior of populations of coupled oscillators, *J. Theoretical Biology* 1967; 16:15–42.
- [77] CORTES, J., MARTINEZ, S., T. KARATAS, T., F. BULLO, F., Coverage control for mobile sensing networks. *The IEEE Trans. on Robotics and Automation* 2004; 20(2):243–255.
- [78] ESTRIN, D., GOVINDAN, R., HEIDEMANN, J., KUMAR, S., Next century challenges: scalable coordination in sensor networks. *Proc. of Mobile Computing and Networking* 1999; pp. 263–270.
- [79] OGREN, P., FIORELLI, E., LEONARD, N.E., Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment, *IEEE Trans. on Automatic Control*, August, 2004.
- [80] SPANOS, D., OLFATI-SABER, R., MURRAY, R.M., Dynamic Consensus on mobile networks, the IFAC World Congress, 2005.
- [81] OLFATI-SABER, R., Flocking for multi-agent dynamic systems: Algorithms and theory, *IEEE Trans. Automatic Control* 2006; 51(3):401–420.
- [82] KLEINBERG, J., LAWRENCE, S., The structure of the web, *Science* 2001; 294:1849–1850.
- [83] DEAN, A.W., *Fighting networks: the defining challenge of irregular warfare*, Ph. D. Thesis, Naval Postgraduate School, 2011.
- [84] <http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf>-random-complex network lecture.pdf, Haziran, 2012.
- [85] BARABASI, A.L., ALBERT, R., Emergence of scaling in random networks, *Science* 1999; 286:509–512.

- [86] HUANG, R., Interplay between structural randomness, composite disorder, and electrical response: resonances and transient delays in complex impedance networks, M. Sc. Thesis, Rensselaer Polytechnic Institute, 2009.
- [87] AGUIRRE, C., CORBACHO, F.J., HUERTA, R., A realistic substrate for small-world networks modeling, DEXA Workshop2001;649-653.
- [88] GUARE, JOHN, Six Degrees of Separation: A Play (First edition ed.), New York: Random House, 1990.
- [89] <http://orion.math.iastate.edu/rymartin/talks/SixDegrees/6dMillersville10.pdf>, Mayıs, 2012.
- [90] ALFATI-SABER, R., Ultrafast consensus in small-world networks, Proc. Amer. Control Conf., 2005; pp. 2371-2378.
- [91] WATTS D.J., Small worlds: the dynamic of networks between order and randomness, Princeton University Press, 1999.
- [92] ERKAYMAZ, O., ÖZER, M., YUMUSAK, N., Effect of small-world network topology on learning in feed forward neural network, Symposium on Innovations in Intelligent Systems and Applications, 132, 2010.
- [93] NEWMAN, M.E.J., WATTS, D.J., Renormalization group analysis of the small-world network model, Physics Letters A 1999; 263(4-6):341-346.
- [94] BENTHIEN, G. W., Leonhard euler: his life and his faith, 2008, <http://gbenthien.net/Euler/Euler.pdf>.
- [95] [http://acikders.hacettepe.edu.tr/dersler/ders\\_belge/MTK361/cizge.pdf](http://acikders.hacettepe.edu.tr/dersler/ders_belge/MTK361/cizge.pdf), Mayıs, 2012.
- [96] BIGGS, N.; LLOYD, E.; WILSON, R., Graph theory, Oxford University Press, 1736–1936, 1986.
- [97] GIBBONS, ALAN, Algorithmic graph theory, Cambridge University Press, 1985.
- [98] TRUDEAU, R. J., Introduction to graph theory, Dover, NY, 1993.
- [99] BONDY, J.A., MURTY, U.S.R., Graph theory, Springer, 2008.
- [100] <http://algs4.cs.princeton.edu/41undirected>, Mayıs, 2012.
- [101] Floyd, R. W., Algorithm 97: shortest path, Communications of the ACM 1962;5(6):345.
- [102] <http://www.thexpot.net/2008/01/floyd-warshall-algoritmasi>, Haziran, 2012.

- [103] ORHAN, U., HEKİM, M., ÖZER, M., Diagnosis of epilepsy from eeg by using artificial neural network probability distribution based clustering, Symposium on Innovations in Intelligent SysTems and Applications, Kayseri, Turkey, 2010.
- [104] ERKAYMAZ, O., ÖZER, M., YUMUSAK, N., Performance analysis of a feed-forward artificial neural network with small-world topology, *Procedia Technology* 2012; 1:291-296.
- [105] GEDİK, E., Düz ve zigzaglı emici plaka yüzeyine sahip farklı iki hava akışkanlı güneş kollektörü tasarımlarının deneysel olarak incelenmesi ve performanslarının kıyaslanması, Y. Lisans Tezi, Karaelmas Üniversitesi, 2007.
- [106] CANER, M., GEDİK, E., KECEBAS, A., Investigation on thermal performance calculation of two type solar air collectors using artificial neural network, *Expert Systems with Applications* 2011; 38(3):1668-1674.
- [107] WELCH, B.L., The generalization of student's problem when several different population variances are involved, *Biometrika* 1947; 34.
- [108] ERKAYMAZ, O., ÖZER, M., YUMUSAK, N., Impact of Small-World topology on the performance of a feed-forward artificial neural network based on two different real-life problems , *Turkish Journal of Electrical Engineering & Computer Sciences*, 2012. (In press).
- [109] YAPICI, F., The effect of some production factors on the properties of osb made from scotch pine (pinus sylvestris l.) wood, Doktora Tezi, Zonguldak Karaelmas Üniversitesi, 2008.
- [110] TS 642/ISO 554, Standard atmospheres and /or testing; Specifications, 1997.
- [111] TS EN 310, Wood-Based panels-Determination of modulus of elasticity and of bending strength, TSE, Ankara, 1999.

## ÖZGEÇMİŞ

Okan ERKAYMAZ, 03.03.1980 de Sungurlu'da doğru ilk, orta ve lise eğitimini Sungurlu'da tamamladıktan sonra 1996 yılında Karadeniz Teknik Üniversitesi Biyoloji öğretmenliğine girdi. Burada bir yıl okuduktan sonra, Gazi Üniversitesi Bilgisayar Sistemleri Öğretmenliğinden mezun oldu. Aynı yıl Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Ana Bilim Dalında Yüksek Lisans Eğitimine başladı. "Geleneksel seçim sistemine bir alternatif olarak e-seçim uygulaması" adlı yüksek lisans tezi ile 2004 yılında mezun oldu. 2005 yılında Zonguldak Karaelmas Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Ana Bilim Dalı' nda ikinci kez yüksek lisans eğitimine başladı. Nöron Dinamiklerinin Neuron Yazılımı İle Modellenmesi adlı yüksek lisans tezi ile 2007 yılında mezun oldu 2007 yılından beri Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği Ana Bilim Dalı' nda doktora çalışmalarını sürdürmektedir. Halen, Karabük Üniversitesi Elektronik ve Bilgisayar Eğitimi Bölümünde Öğretim Görevlisi olarak çalışmaktadır.