

**TIP II REGRESYON ANALİZİNE  
YAPAY SİNİR AĞI YAKLAŞIMI**

YÜKSEK LİSANS TEZİ

Berkalp TUNCA

Danışman

Doç. Dr. Sinan SARAÇLI

İSTATİSTİK ANABİLİM DALI

Haziran 2019

Bu tez çalışması 18.FEN.BİL.13 numaralı proje ile  
Afyon Kocatepe Üniversitesi BAPK tarafından desteklenmiştir.

**AFYON KOCATEPE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**TİP II REGRESYON ANALİZİNE YAPAY SİNİR AĞI YAKLAŞIMI**

**Berkalp TUNCA**

**Danışman**

**Doç. Dr. Sinan SARAÇLI**

**İSTATİSTİK ANABİLİM DALI**

**Haziran 2019**

## TEZ ONAY SAYFASI

Berkalp TUNCA tarafından hazırlanan “Tip II Regresyon Analizine Yapay Sinir Ağları Yaklaşımı” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 11/06/2019 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **İstatistik Anabilim Dalı’nda** **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** : Doç. Dr. Sinan SARAÇLI

**İmza**

**Başkan** : Doç. Dr. Çağdaş Hakan ALADAĞ  
Hacettepe Üniversitesi, Fen Fakültesi

**Üye** : Doç. Dr. Sinan SARAÇLI  
Afyon Kocatepe Üniversitesi, Fen-Edebiyat Fakültesi

**Üye** : Doç. Dr. İbrahim KILIÇ  
Afyon Kocatepe Üniversitesi, Veteriner Fakültesi

Afyon Kocatepe Üniversitesi  
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun  
...../...../..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

.....  
Prof. Dr. İbrahim EROL  
Enstitü Müdürü

**BİLİMSEL ETİK BİLDİRİM SAYFASI**  
**Afyon Kocatepe Üniversitesi**

**Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;**

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

11/06/2019



**İmza**

**Berkalp TUNCA**

**ÖZET**  
Yüksek Lisans Tezi

TİP II REGRESYONA YAPAY SİNİR AĞLARI YAKLAŞIMI

Berkalp TUNCA  
Afyon Kocatepe Üniversitesi  
Fen Bilimleri Enstitüsü  
İstatistik Anabilim Dalı

**Danışman:** Doç. Dr. Sinan SARAÇLI

Bu çalışmada, Tip II Regresyon tekniklerinden olan EKK Açıortay tekniğine ilk defa bu çalışma aracılığıyla Yapay Sinir Ağı (YSA) yaklaşımı yapılmıştır. Yeni oluşturulan bu YSA Açıortay tekniğinin performansının ölçülmesi amacıyla da EKK Açıortay tekniğiyle karşılaştırılmıştır. Öncelikle YSA ve Tip II Regresyon teknikleri konularında literatür bilgilerine yer verilerek, iki tekniğinde özelliklerinden bahsedilmiştir. Bu bilgiler doğrultusunda EKK tabanlı açıortay tekniği ile YSA tabanlı açıortay teknikleri arasında karşılaştırma yapılmıştır. Bu iki tekniğin karşılaştırılması amacıyla farklı dağılımlarda ve farklı örneklem hacimlerinde modellemeler yapılmıştır. Bu modellerin performanslarının karşılaştırılması amacıyla “Ortalama Mutlak Yüzde Hata” (MAPE) kriterinden yararlanılmıştır. Çalışma sonucunda YSA tabanlı Açıortay tekniği EKK tabanlı Açıortay tekniğine göre daha düşük hata ile daha iyi sonuç verdiği görülmüştür. Yapılan bu çalışma ile gelecek dönemde bu alanlarda çalışmak isteyen araştırmacılar için bir örnek temsilinde olduğu ön görülmektedir.

**2019, vii + 88 sayfa**

**Anahtar Kelimeler:** Yapay Sinir Ağı, Tip II Regresyon, Ölçüm Hata Modelleri, EKK tabanlı açıortay tekniği, YSA tabanlı açıortay tekniği

**ABSTRACT**  
M.Sc. Thesis

ARTIFICIAL NEURAL NETWORK APPROACH ON TYPE II REGRESSION  
ANALYSIS

Berkalp TUNCA

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Statistics

**Supervisor:** Assoc. Prof. Sinan SARAÇLI

In this study, Artificial Neural Network (ANN) approach was used for the first time to the OLS Bisector technique which is one of the Type II Regression techniques. In order to measure the performance of this newly created ANN Bisector technique, the OLS was compared with the Bisector technique. Firstly, the literature is given on ANN and Type II Regression techniques. In line with this information, the comparison between OLS-based bisector technique and ANN-based bisector techniques has been made. In order to compare these two techniques, different distributions and different sample volumes were modeled. “Mean Absolute Percentage Error” (MAPE) criterion was used to compare the performance of these models. As a result of the study, ANN-based bisector technique yielded better results with lower error than OLS-based bisector technique. With this study, it is foreseen that it represents an example for the researchers who want to work in these fields in the future.

**2019, vii + 88 pages**

**Keywords:** Artificial Neural Network, Type II Regression, Measurement Error Models, OLS based bisector technique, ANN based bisector technique

## TEŞEKKÜR

Bu tezin hazırlanması sürecinde, araştırma konusunun belirlenmesi, bilgi ve birikimini bana aktarması sayesinde böyle bir araştırmanın ortaya çıkması için büyük bir emeği olan ve her konuda sırtımı güvenle yaslayabileceğimi bildiğim tez danışmanım Sayın Doç. Dr. Sinan SARAÇLI'ya, tezimde ki analizlerin gerçekleşmesi ve doğru yerde yeterli bilgi verilmesi konusunda verdiği büyük emekler için Sayın Doç. Dr. Çağdaş Hakan ALADAĞ'a, akademik bir gelecek için beni cesaretlendirerek doğru adımlar atmamı sağlayan ve lisans döneminden itibaren bana olan akademik katkılarından ötürü Sayın Doç. Dr. İbrahim KILIÇ'a,

Hayatta bana her konuda inancını asla yitirmeyen, maddi ve manevi destekleriyle benim bu noktaya gelmemdeki en büyük aracı olan annem Hatice TUNCA'ya, babam Nejdett TUNCA'ya ve kardeşim Betül TUNCA'ya,

Manevi bakımdan büyük bir düşünüş yaşadığımda her konuda olduğu gibi bu tez konusunda da elimden geleni yapmam konusunda beni cesaretlendiren ve motivasyonumun artmasını sağlayan sevgili arkadaşım Ece ÖZGÖREN'e teşekkürlerimi sunarım.

Ayrıca bu tezin oluşmasına katkı sağlayan 18.FEN.BİL.13 proje numaralı BAPK'ne teşekkürlerimi sunarım.

Berkalp TUNCA  
AFYONKARAHİSAR, 2019

## İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER DİZİNİ.....	iv
SİMGELER ve KISALTMALAR DİZİNİ .....	v
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ.....	vii
1. GİRİŞ.....	1
2. LİTERATÜR BİLGİLERİ .....	3
2.1 Yapay Sinir Ağları .....	3
2.1.1 Biyolojik Sinir Hücresi ve Yapay Sinir Hücresi.....	3
2.1.2 Yapay Sinir Ağlarının Tarihsel Gelişimi .....	5
2.1.3 Yapay Sinir Ağlarındaki Temel Tanımlar .....	6
2.1.4 Literatürdeki Yapay Sinir Ağları Tanımları .....	8
2.1.5 Yapay Sinir Ağlarının Özellikleri.....	11
2.1.6 Yapay Sinir Ağlarının Temel Bileşenleri .....	13
2.1.6.1 Mimari Yapı .....	14
2.1.6.2 Aktivasyon Fonksiyonu .....	27
2.1.6.3 Yapay Sinir Ağlarında Öğrenme.....	30
2.1.7 Veri İşlenmesi .....	40
2.2 Tip I (Basit Doğrusal Regresyon) ve Tip II Regresyon Teknikleri.....	45
2.2.1 Tip I Regresyon Analizi.....	45
2.2.2 Tip II Regresyon Analizi .....	48
2.2.2.1 EKK-Açıortay Tekniği.....	48
2.2.2.2 Majör Eksen (MA) Ortogonal Regresyon Tekniği .....	49
2.2.2.3 İndirgenmiş Majör (RMA-Reduced Major Axis) Eksen Regresyon Tekniği .....	51
2.2.2.4 Deming Regresyon Tekniği .....	53
2.2.2.5 Passing-Bablok Regresyon Tekniği .....	57
2.2.2.6 York Regresyon Tekniği .....	60
3. MATERYAL ve METOT .....	64
4. BULGULAR .....	68
5. TARTIŞMA ve SONUÇ .....	76
6. KAYNAKLAR.....	79
ÖZGEÇMİŞ.....	88



## SİMGELER ve KISALTMALAR DİZİNİ

### Simgeler

---

$w$	Hücreler Arası Ağırlıklar
$b$	Yan
$\beta_0$	Regresyon Denkleminin Sabit Katsayısı
$\beta_1$	Regresyon Denkleminin Eğim Katsayısı
$\exp$	Üstel
$\alpha$	Alpha
$\beta$	Beta
$S_{ex}^2$	X Gözlemlerine Ait Hataların Varyansı
$S_{ey}^2$	Y Gözlemlerine Ait Hataların Varyansı
$x_i$	X Değişkeninin i. Gözlem Değeri
$y_i$	Y Değişkeninin i. Gözlem Değeri
$\pi$	Phi Sayısı
$\varepsilon$	Hata terimi

### Kısaltmalar

---

YSA	Yapay Sinir Ağı
EKK	En Küçük Kareler
LVQ	Doğrusal Vektör Nicelemesi
LM	Levenberg-Marquardt
ÇKA	Çok Katmanlı Algılayıcılar
ALC	Adaline Modeli
TH	Toplam Hata
EKK(X Y)	X Bağımlı Y bağımsızken Regresyon Modeli
EKK(Y X)	Y Bağımlı X Bağımsızken Regresyon Modeli
SS	Standart Sapma
n	Örneklem Hacmi
MER	Majör Eksen Regresyon
OR	Ortogonal Regresyon
RMA	İndirgenmiş Majör Eksen
HKT	Hata Kareler Toplamı
MAPE	Ortalama Mutlak Yüzde Hata-Mean Absolute Percentage Error

---

## ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1	Biyolojik Bir Sinir Hücresi .....	4
Şekil 2.2	Bir Yapay Sinir Ağı Hücresinin Şekilsel Gösterimi .....	4
Şekil 2.3	Bir Yapay Sinir Ağı Hücresi .....	13
Şekil 2.4	Örnek Bir İleri Beslemeli Yapay Sinir Ağ Mimarisi .....	14
Şekil 2.5	Bir Tek Katmanlı Algılayıcı Mimarisi .....	16
Şekil 2.6	İki Sınıfı Birbirinden Ayıran Doğrunun Grafiği .....	16
Şekil 2.7	Tekli Sınıflandırma Yapmak İçin Bir Perceptron Yapısı.....	18
Şekil 2.8	İki Adaline Ünitesinden Oluşmuş Madaline Ağı.....	20
Şekil 2.9	Çok Katmanlı Bir Algılayıcı Ağının Yapısı.....	23
Şekil 2.10	İleri Beslemeli Ağ Yapısı.....	25
Şekil 2.11	Geri Beslemeli Ağ Yapısı .....	27
Şekil 2.12	Purelin Fonksiyonu Çıktı Grafiği.....	28
Şekil 2.13	Sigmoid Fonksiyon Giriş-Çıkış Grafiği.....	28
Şekil 2.14	Tansig Fonksiyonun Girdi-Çıkış Grafiği.....	28
Şekil 2.15	Step (Adım) Fonksiyonu Çıktı Grafiği .....	29
Şekil 2.16	Eşik Değer Fonksiyonu Çıktı Grafiği .....	29
Şekil 2.17	Hiperbolik Tanjant Fonksiyonu Çıktı Grafiği.....	30
Şekil 2.18	Danışmanlı Öğrenme .....	33
Şekil 2.19	LVQ Ağ Mimarisi .....	35
Şekil 2.20	Ortogonal Regresyon Tekniğinde Minimize Edilmek İstenilen Hata Miktarının Grafiği .....	50
Şekil 2.21	İndirgenmiş Majör Ekseni Ortogonal Regresyon Tekniğinde Minimize Edilmek İstenen Hata Miktarı Grafiği .....	52
Şekil 2.22	Deming Regresyon Tekniğinin Bir Model Grafiği .....	54
Şekil 2.23	EKK Modelinde, X Hata İçermezken Y için Sabit Hataların Dağılımı.....	55
Şekil 2.24	Deming Regresyon Yönteminin Grafiği .....	55
Şekil 4.1	T(30), n=350 iken Hedef ve Tahminlerin Dağılım Grafiği .....	71
Şekil 4.2	T(30), n=150 iken Hedef ve Tahminlerin Dağılım Grafiği .....	72
Şekil 4.3	T(30), n=50 iken Hedef ve Tahminlerin Dağılım Grafiği .....	72
Şekil 4.4	T(10), n=350 iken Hedef ve Tahminlerin Dağılım Grafiği .....	73
Şekil 4.5	T(10), n=150 iken Hedef ve Tahminlerin Dağılım Grafiği .....	73
Şekil 4.6	T(10), n=50 iken Hedef ve Tahminlerin Dağılım Grafiği .....	74
Şekil 4.7	T(4), n=350 iken Hedef ve Tahminlerin Dağılım Grafiği .....	74
Şekil 4.8	T(4), n=50 iken Hedef ve Tahminlerin Dağılım Grafiği .....	75
Şekil 4.9	T(4), n=50 iken Hedef ve Tahminlerin Dağılım Grafiği .....	75

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
<b>Çizelge 2.1</b> Birleştirme Fonksiyonu Örnekleri .....	7
<b>Çizelge 2.2</b> Adaline Algoritması .....	19
<b>Çizelge 2.2</b> (Devam) Adaline Algoritması.....	20
<b>Çizelge 2.3</b> Geri Yayılım Algoritması (Genelleştirilmiş Delta Öğrenme Kuralı) .....	37
<b>Çizelge 2.4</b> Perceptron Algoritması .....	38
<b>Çizelge 2.4</b> (Devam) Perceptron Algoritması .....	39
<b>Çizelge 4.1</b> Örneklem Hacmi 350 iken, Farklı Dağılımlara İlişkin YSA-Açıortay ve EKK-Açıortay Tekniklerinin MAPE Değerleri.....	69
<b>Çizelge 4.2</b> Örneklem Hacmi 150 iken, Farklı Dağılımlara İlişkin YSA-Açıortay ve EKK-Açıortay Tekniklerinin MAPE Değerleri.....	69
<b>Çizelge 4.3</b> Örneklem Hacmi 50 iken, Farklı Dağılımlara İlişkin YSA-Açıortay ve EKK-Açıortay Tekniklerinin MAPE Değerleri.....	70

## 1. GİRİŞ

Yapay Sinir Ağları basit bir şekilde, biyolojik sinir hücresini (nöronunu) temel alarak bilgi işleme tekniğidir. Tıpkı insan beynindeki gibi sinir hücreleri arasında bağlar vardır. Bu bağların birbirlerine bağlanmasıyla Yapay Sinir Ağı oluşur. Günümüzde tıp, kalite kontrol, istatistik gibi birçok alanda kullanılabilir. Yapay Sinir Ağlarının en büyük özelliklerinden birisi öğrenbilme yeteneğidir. Bu öğrenbilme yeteneği sayesinde bir problem söz konusu olduğunda diğer alternatif tekniklere göre düşük hata ile daha iyi sonuçlar verdiği için araştırmacılar tarafından oldukça çok kullanılan bir teknik haline gelmiştir. Ayrıca literatür incelendiğinde diğer alternatif tekniklerle Yapay Sinir Ağı karşılaştırılması oldukça fazladır ve bu karşılaştırmalar sonucunda Yapay Sinir Ağının başarısı konusunda oldukça fazla kaynak bulunmaktadır. Bu karşılaştırmalara tartışma bölümünde bahsedilmiştir.

Tip II Regresyon tekniklerinin ortaya çıkmasının en büyük sebebi ölçüm hatalarından kaynaklanmaktadır. Klasik regresyon modelinde bağımsız değişkenden kaynaklanan hatalar modele katılmamaktadır. Fakat günlük hayatta bağımsız değişkenden kaynaklanan hatalarından olduğu bilinmektedir. Klasik regresyon modelinin bu dezavantajı yüzünden ortaya çıkan ölçümsel hata, Tip II Regresyon modeline ihtiyaç duyulmasını sağlamıştır. Tip II Regresyon teknikleri şunlardır: EKK Açıortay regresyon tekniği, Majör Eksen Ortogonal regresyon tekniği, İndirgenmiş Ortogonal Majör Eksen regresyon tekniği, Deming regresyon tekniği, Passing-Bablok regresyon tekniği ve York regresyon tekniği. Bu tekniklerin temel olarak aldığı ve en büyük ortak noktaları ise bağımsız değişkenden kaynaklı hataların da modele eklenmesidir.

Bu çalışmada ise, literatürde Tip II regresyon ve YSA karşılaştırmasının bulunmamasından dolayı Tip II Regresyon tekniği olan EKK tabanlı Açıortay regresyon tekniği ile YSA tabanlı Açıortay regresyon tekniği karşılaştırması yapılacak olup, sonuçların literatüre ve daha sonraki yapılacak çalışmalara da büyük bir ölçüde katkı sağlanacağı ön görülmektedir. Ayrıca çalışmada EKK Açıortay tekniğinin kullanılma sebebi, diğer Tip II Regresyon tekniklerine göre matematiksel işlem olarak daha kolay ve

uygulanabilir olmasıdır. Daha önceki çalışmalarda Yapay Sinir Ağları modelinin diğer modellere göre (Regresyon, Çoklu Regresyon, Box-Jenkins vd.) daha düşük hatayla daha iyi sonuçlar verildiği gözlemlenmiştir. Bu çalışmayla, Tip II Regresyonun ve Yapay Sinir Ağlarının tahminlemedeki zayıflıklarından ve üstünlüklerinden bahsedilerek, iki ayrı tekniğin de önemi vurgulanacaktır. Yapılan çalışmada, Tip II Regresyonun ve Yapay Sinir Ağlarının tahminlemedeki performansları ölçülmüş, hem de literatürdeki diğer karşılaştırmalarda olduğu gibi Yapay Sinir Ağları modelinin öngörü performansı ile Tip II Regresyon modelinin tahminleme performansı karşılaştırılmıştır.

## **2. LİTERATÜR BİLGİLERİ**

Bu bölümde öncelikle Yapay Sinir Ağları, sonrasında ise Tip II Regresyon teknikleriyle ilgili literatür bilgilerine yer verilmiştir.

### **2.1 Yapay Sinir Ağları**

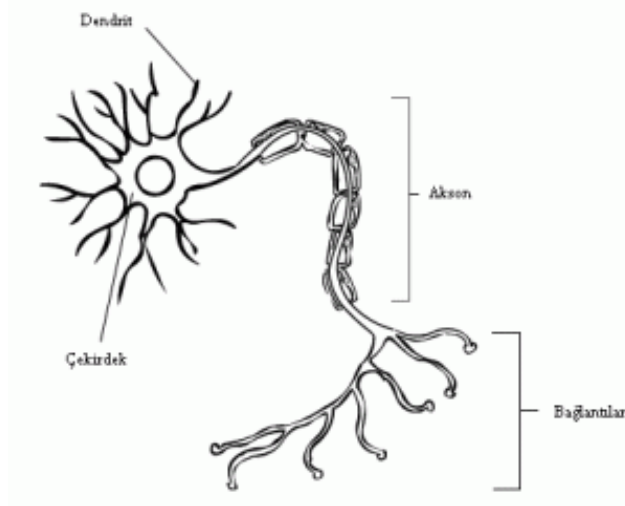
İnsan beyninin hücresinden yola çıkılarak ortaya çıkan Yapay Sinir Ağı modellerinin, yapısı, tarihsel gelişimi, literatürdeki Yapay Sinir Ağı tanımları, özellikleri, temel bileşenleri ve öğrenme türleri gibi bilgiler aşağıdaki bölümlerde verilmiştir.

#### **2.1.1 Biyolojik Sinir Hücresi ve Yapay Sinir Hücresi**

Biyolojik bir sinir hücresi dört bölümden oluşmaktadır;

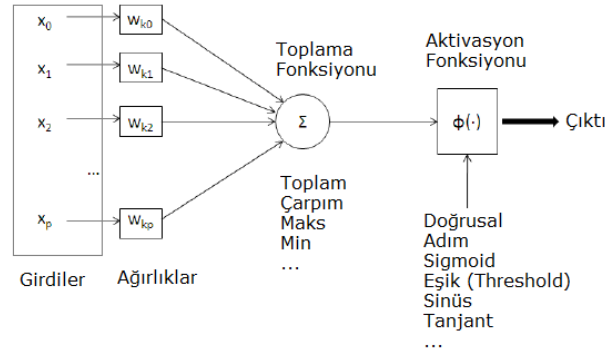
1. Çekirdek (Aktivasyon Fonksiyonu)
2. Akson (Çıktı)
3. Dendrit (Sinir uçları) (Birleştirme Fonksiyonu)
4. Sinaps (Bağlantılar) (Ağırlıklar)

Bir sinir hücresinde, çok sayıda dendrit bulunmaktadır. Bu dendritler kendilerine gelen sinyalleri çekirdeğe iletmekte görevlidirler. Dendritler aracılığıyla gelen bu sinyaller çekirdek yardımıyla bir araya toplanarak aksona iletilir. Çekirdekte toplanarak iletilen sinyaller akson aracılığıyla işlendikten sonra sinaplara gönderilir. Sinaplara iletilen sinyaller yeniden düzenlenir ve yeni üretilmiş sinyaller diğer sinir hücrelerine iletilir. Şekil 2.1’de bir biyolojik sinir hücresinin yapısı verilmiştir.



**Şekil 2.1** Biyolojik bir sinir hücresi

Bir yapay sinir ağı hücresi, biyolojik sinir ağı hücresinin simule edilmesiyle gerçekleştirilen hücrelerdir. Dışardan alınan girdiler, ağırlıklar aracılığıyla hücreye bağlanır. Birleştirme fonksiyonunda işlem (toplam, çarpım, maksimum, minimum) geçerek net girdi hesaplanır. Daha sonra net girdi aktivasyon fonksiyonundan geçerek net çıktı bulunmuş olur. Bu oluşturulan sinir ağı hücresi, çıktıların doğruluğuna göre en uygun ağırlık değerlerini buluncaya kadar işlemleri tekrar edecektir. Şekil 2.2’de bir yapay sinir ağı hücresinin şekilsel gösterimi verilmiştir.



**Şekil 2.2** Bir yapay sinir ağı hücresinin şekilsel gösterimi

YSA ile ilgili çalışmalar insan beynini modellemek ve nasıl çalıştığını anlamak amacıyla başlamıştır. İnsan beynini çalışma fonksiyonuna benzetme isteği YSA modellerine esin kaynağı olmuştur. Beyin, birbirine değişik etki seviyeleri ile bağlantılı binlerce hücreden oluşmaktadır. YSA modelleri de birbirlerine ağırlık bağlantılarıyla bağlanarak, temel

hücre modelleri oluşturmaktadır. YSA sonuç olarak bir bilgi işleme modeli olarak karşımıza çıkmaktadır (Canan 2006).

YSA tekniklerinin son yıllarda her alanda yoğun ilgi görmesinin sebebi, klasik yöntemlerle çözülemeyen problemler veya çözümü çok karmaşık olan problemler söz konusu olduğunda başarılı bir çözüm getirmesi ve klasik metotlara güçlü bir alternatif oluşturmasıdır.

### **2.1.2 Yapay Sinir Ağlarının Tarihsel Gelişimi**

YSA, tıpkı insan vücudundaki sinirler gibi genelleme yapabilme, öğrenebilme ve düşünebilme gibi becerilerin kazanılmasını için oluşturulan bir bilgi işleme sistemidir. İnsan gibi düşünebilme ve insan beyninin taklit edilebilmesi gibi istekler çok eski zamanlardan gelmektedir. Bu amaçla geliştirilen ilk teori denemeleri Plato ve Aristoteles gibi düşünülere kadar uzanmaktadır. Ayrıcı aynı amaçla yola çıkan Descartes'da 18. yüzyıl için bu konu hakkında önemli bir düşünür olmuştur (Kunt 2007).

YSA, hücrelerin birbirlerine farklı farklı şekillerde bağlanmasıyla oluşur. YSA, farklı algoritmalar aracılığıyla öğrenme aşamasına girerler ve daha sonra yapay hücrelerin aralarındaki bağlantıların ağırlıkları aracılığıyla bilgiyi saklayabilme ve genelleme yapabilme becerilerini elde ederler. Öğrenme aşamasında, YSA'nın yapısına göre birbirlerinden farklı birçok öğrenme algoritması kullanılmaktadır. McCulloch ve Pitts 1943 yılında yapmış oldukları bir çalışma ile yapay sinir ağının ilk matematiksel modellemesinin temelini atmışlardır. Farley ve Clark ise 1954 yılında yaptıkları çalışmayla dışardan gelen uyarılara tepki verebilen ve bu uyarılara karşı adapte olabilen bir model geliştirmişlerdir. Daha sonra 1960 yılında ilk defa sinirsel bir bilgisayar ortaya çıkmıştır. 1963 yılında ise basit modellerdeki eksiklikler fark edilmiştir ve 1970 ve 1980'lere kadar yapay sinir ağlarından başarılı sonuçlar alınabilmesi gecikmiştir. 1985 ise araştırmacılar için yapay sinir ağlarının tanındığı ve bu konudaki çalışmaların yoğun bir şekilde başladığı yıl olarak kabul görmüştür (Kunt 2007, Pankaj and Wah 1992).



Minsky and Papert'in yazdıkları Perceptron adlı kitaplarında, yapay sinir ağlarının ilgi çekici bir konu olmadığını söylemeleri, birçok araştırmacının bu konu hakkında çalışmak istememelerine sebep olmuştur. Buna rağmen Grossberg yapay sinir ağları konusunda çalışmalarına devam ederek, yapay sinir ağı modellerini geliştirmek amacıyla nörolojik verilerin kullanılması, hafıza ve algı için yapay sinir ağı modeli kullanılarak oluşturulan sistemlerin önerilmesi ve belirgin eşitliklerle bütünleşmiş olan sinaptik bir model için, bağdaştırıcı bir kaide üzerinde çalışmıştır (Öztemel 2012).

YSA alanında dikkat çeken başka bir gelişme ise 1982 yılında fizikçi Hopfield tarafından sunulmuştur. Hopfield, moleküler biyolojiden insan beynindeki hücrelerin işleyişi kuramına geçiş yapan bir model oluşturmuştur. Bu kurduğu modele kendi ismini vermiştir ve Hopfield ağı birçok alanda uygulanmıştır.

1987 yılında ise ilk defa YSA sempozyumu yapılmıştır ve bu sempozyumdan sonra yapay sinir ağları uygulamaları oldukça yaygınlaşmaya başlamıştır. Geçmişten günümüze yapay sinir ağları birçok bilim adamı ve araştırmacı grupları tarafından ilgi çekici bir konu olarak araştırılmaya devam edilmiştir (Öztemel 2012).

### **2.1.3 Yapay Sinir Ağlarındaki Temel Tanımlar**

**a) Girdiler;** Bir yapay sinir ağı hücresine dışardan gelen veya ağı diğer hücrelerinden gelen bilgilerdir. Yapay sinir ağının öğrenmesi istenilen bilgilerdir. Hiçbir işlem uygulanmadan direkt iletilir (Öztemel 2006).

**b) Ağırlıklar;** Basit bir şekilde bakılırsa, girdilerden gelen bilgi ağırlıklarla çarpılarak birleştirme fonksiyonuna gider. Ağırlıklar hücreye gelen bilginin hücre üzerindeki etkisini göstermektedir. Ağırlık değerleri negatif veya pozitif değerler olabilirler. Ağırlığın küçük veya büyük olmasının herhangi bir anlama gelmemektedir. Ağırlığın sıfır olması durumunda ise herhangi bir etkinin olmadığı anlamına gelmektedir. Ağırlıklar sabit ve değişken değerler olabilirler (Öztemel 2006).

c) **Birleştirme Fonksiyonu;** Bu fonksiyon, bir hücreye gelen net girdiyi hesaplar. Burada birçok farklı fonksiyonlar kullanılmaktadır. Örnek olarak Eşitlik 2.1’de ağırlıklı toplam fonksiyonu verilmiştir. Her gelen girdi değeri kendi ağırlığı ile çarpılarak toplanılır. Bu şekilde toplanan değerler yardımıyla ağa gelen net girdi bulunmuş olur (Öztemel 2006).

$$NET = \sum_i^n G_i A_i \quad (2.1)$$

Burada G girdileri, A ise ağırlıkları, n ise bir hücreye gelen toplam girdi sayısını göstermektedir. Yalnız yapay sinir ağlarında daima bu formülün kullanılması şart değildir. Uygulanan yapay sinir ağı modellerinden bazıları kullanılacak birleştirme fonksiyonunu belirleyebilmektedir. Geçmişte yapılan araştırmalara bakıldığında birleştirme fonksiyonunun değişik formüllerle kullanıldığı görülmektedir. Çizelge 2.1’de farklı birleştirme fonksiyonlarına örnekler verilmiştir. Burada araştırmacı kendi problemi için her bir birleştirme fonksiyonunu deneme yanılma yolunu izleyerek belirleyebilir (Öztemel 2006).

**Çizelge 2.1** Birleştirme fonksiyonu örnekleri

<b>Net Giriş</b>	<b>Açıklama</b>
<p><i>Çarpım</i></p> $Net\ Girdi = \prod_i G_i A_i$	Ağırlık değeri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak net girdi hesaplanır
<p><i>Maksimum</i></p> $Net\ Girdi = Max(G_i A_i), i = 1 \dots N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
<p><i>Minimum</i></p> $Net\ Girdi = Min(G_i A_i), i = 1 \dots N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
<p><i>Çoğunluk</i></p> $Net\ Girdi = \sum_i sgn(G_i A_i)$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
<p><i>Kümülatif Toplam</i></p> $Net\ Girdi = Net(eski) + \sum_i sgn(G_i A_i)$	Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur.

*d) Çıktı;* Aktivasyon fonksiyonunda hesaplanan değer hücrenin çıktı değeridir (Aktivasyon fonksiyonu hakkında ayrıntılı bilgi aşağıda verilmiştir). Bu çıktı değeri istenirse yapay sinir ağının çıktısı olarak da kullanılabilir, istenirse de ağın içerisinde tekrar kullanılabilir. Her yapay sinir ağ hücresinin birçok girdisi olmasına karşın her hücrenin tek bir çıktısı olmaktadır. Bu çıktı, istenilen sayıda hücreye bağlanabilir (Öztemel 2012). Ancak bu durum mimari yapıya göre değişebilir.

#### **2.1.4 Literatürdeki Yapay Sinir Ağları Tanımları**

YSA modelleri, insan beyninin düşünebilme, öğrenebilme ve problem çözebilme yetilerinden yararlanılarak oluşturulduğu için, araştırmacılar tarafından oldukça ilgi çekici bulunmuştur. Araştırmacılar, beynin bu yetilerini kısmen yerine getirebilen modelleri bilgisayar ortamına geçirerek çok sayıda YSA modeli kurmuşlardır (Öztemel 2006, Çelik 2008).

Yurtoğlu (2005) “YSA, en kısa ve basit şekilde, bir örnekler kümesi yardımıyla parametrelerin uyarlanabilmesini sağlayacak bir matematiksel formül için yazılan bilgisayar programı olarak tanımlanabilir. Bu tanım, YSA’yı en basit şekilde ve teknik detaya girilmeksizin ifade etmektedir.” şeklinde yapay sinir ağlarını tanımlamıştır.

Haykin (1999), bir YSA, deneysel bilginin depolanması ve kullanım için uygun bir biçime getirilmesi için doğal bir eğilime sahip olan basit işlem hücrelerinden oluşan, büyük ölçüde paralel olarak dağıtılmış bir işlemcidir ve insan beynine iki açıdan benzemektedir:

1. Bilgi ağı, çevre tarafından bir öğrenme süreci yoluyla edinilir.
2. Sinaptik ağırlıklar olarak bilinen sinirler arasındaki bağlantılar, sahip olunan bilgiyi depolayabilmek amacıyla kullanılmaktadır.

Ayrıca Haykin (1999), “Öğrenme sürecini gerçekleştirmek için kullanılan prosedür, bir öğrenme algoritması olarak adlandırılır; bunun fonksiyonu, istenen tasarım hedefine

ulaşmak için ađın sinaptik ađırlıklarını dzenli bir şekilde modifiye etmektir.” diyerek yapay sinir ađlarını en genel bir şekilde tanımlamıştır.

Yapay sinir ađları, geđmişten gelen geliştirmeler aracılıyla, hayatın herhangi bir alanında karşılaşılan problemlere uygun çözümler getirebilmektedir. Bu açıdan araştırmacılar için oldukça yoğun ilgi görmüştür. Yapay sinir ađlarının bazı kullanım alanları aşıđıda verilmiştir (Yurtođlu 2005):

- Kalite Kontrol
- Tıp
- Savunma Sanayi
- Haberleşme
- Finansal ve Ekonomik Öngörüler
- İstatistik
- Kredi Derecelendirme
- Otomasyon
- Konuşma ve Yapı Tanımlama
- İşlem Modelleme ve Yönetimi
- Laboratuvar Araştırmaları
- İflas Tahmini
- Petrol ve Gaz Arama gibi alanlarda YSA kullanıldığı görölmektedir.

YSA’lar, tanımlanmamış test verileri hakkında karar verirken genelleme yapabildikleri için iyi birer yapı tanımlayıcısı ve sınıflandırıcıdırlar. Fonksiyonel tahmin ve sistem modelleme gibi fiziksel işlemin anlaşılamadığı veya aşırı karmaşık olduđu problemler yanında konuşma, karakter ve sinyal tanımlama gibi çeşitli sınıflandırma problemleri için çözüm yolları sağlamaktadırlar. Ayrıca, kontrol problemlerinde de uygulama sahası bulmaktadırlar (Yurtođlu 2005). Bu alanlarda uygulanan YSA uygulamaları incelendiğinde genel olarak aşıđıdaki işlemler için YSA kullanıldığı görölmektedir (Ataseven 2007):

- **Tahmin:** Tahmin amacıyla yapay sinir ađları, ađa sunulmuş olan bilgiden yararlanarak karşılık gelen çıktı deđerini tahmin ederler. Hava tahmini, borsada hisselerin deđerlerinin, ileriki satışların, üretim ihtiyacının, pazar performansının, enerji ihtiyacının, at yarışlarının tahmini, kanser riskini belirleme, döviz düzeylerinin tahmini vb. örnekler verilebilir.
- **Sınıflandırma:** Sınıflandırma amacıyla kullanılan yapay sinir ađları kendilerine sunulmuş olan bilgileri gruplama görevini üstlenirler. Müşteri-Pazar profilleri, şekil tanıma, tıbbi teşhis, risk ve borçlanma deđerlendirmeleri vb. örnekler verilebilir.
- **Veri ilişkilendirme:** Veri ilişkilendirme amacıyla eğitilen yapay sinir ađları, ađa sunulmuş olan girdilerde eksik ve hatalı bir veri olması durumunda kullanılır. Ayrıca ađın öğrendiđi bilgiler sayesinde eksik bilgi olması durumunda bu bilgileri kendiliğinden tamamlar. Örneđin; eksik bir resmin tamamlanması.
- **Veri filtreleme:** Veri filtreleme amacıyla eğitilen yapay sinir ađları, oldukça fazla veri söz konusu olduğunda, bu veriler arasından uygun olabilen verilerin belirlenmesinde kullanılır. Giriş verilerindeki uygun olmayan verilerin ayırt edilmesini sağlar.
- **Tanıma ve eşleştirme:** Tanıma ve eşleştirme amacıyla oluşturulan yapay sinir ađları; farklı şekillerin veya örüntülerin tanımlanması, karmaşık yada eksin olan bilgilerin tamamlanarak eşleştirilmesinde kullanılır. Kalite kontrol alanında kullanılan şemalardaki şekilleri tanıyabilen ađlar, örnek olarak verilebilir.
- **Teşhis:** Teşhis amacıyla oluşturulan yapay sinir ađları, herhangi bir sistemdeki olumsuzlukların ve problemlerin teşhis edilmesinde kullanılır. Örneđin; makinelerin hatalarının teşhis edilmesi. Ayrıca tıp alanında da bu tür sistemler yaygın olarak geliştirilmektedir.

Ayrıca son yıllarda YSA, zaman serileri analizlerinde de oldukça fazla kullanılmaktadır. Ekonomik zaman serileri (örneđin, döviz kurları, borsa fiyatları, deprem büyüklükleri) genellikle hem doğrusal hem de doğrusal olmayan verileri içeren karmaşık yapılardan oluşmaktadır. Döviz kurlarında ki deđişkenlik ve belirsizlik sıklıkla görülmektedir. Bu nedenle, başarılı bir tahmin için ve tüm dünyadaki ekonomik faktörler için ölçüm hatalarını en aza indirmek çok daha önemli hale gelmektedir. Klasik yöntemlerin

eksikliği bu süreci daha da zorlaştırmaktadır. Ancak, YSA hem doğrusal hem de doğrusal olmayan zaman serileri için öğrenme yeteneğine sahiptir. Bu yüzden zaman serileri için YSA, iyi bir alternatif ve başarılı bir öngörü aracı olarak kullanılmaktadır (Günay vd. 2007).

### **2.1.5 Yapay Sinir Ağlarının Özellikleri**

YSA yöntemi klasik yöntemler gibi çeşitli model varsayımları gerektirmez (Gutierrez *et al.* 2008, Aladağ 2009). YSA, girdi ve çıktı değişkenleri arasındaki herhangi bir ön bilgiye ve varsayıma ihtiyaç duymadan gerekli modellemeyi sağlayabilmektedir. Bu sebeple YSA, öngörü aracı olarak diğer yöntemlere göre daha fazla avantaja sahiptir ve daha başarılı sonuçlar ortaya koyabilmektedir (Ataseven 2007).

YSA'nın günümüzde ürettiği iyi sonuçları ve uygulanabilirliği konusunda ki kolaylığı gibi avantajlarıyla matematik, istatistik, fizik, mühendislik ve bilgisayar bilimleri gibi alanlarda başarılı bir şekilde kullanılmaktadır (Günay vd. 2007, Aladağ 2009).

YSA'nın özelliklerinden biri de esnek bir yapıda olmasıdır. YSA, her biri problemin farklı bir bölümüyle ilgilenen birçok hücreden oluşmaktadır ve bağlantı ağırlıklarının değiştirilebilir olması bu esnek yapıya olanak sağlamaktadır. Bu esneklik sayesinde, kurulan bir yapay sinir ağı modelinin bir bölümünün yok olması durumunda ağı performansının düşmesine rağmen modelin hala çalışmasına olanak sağlamaktadır (Karahana 2011).

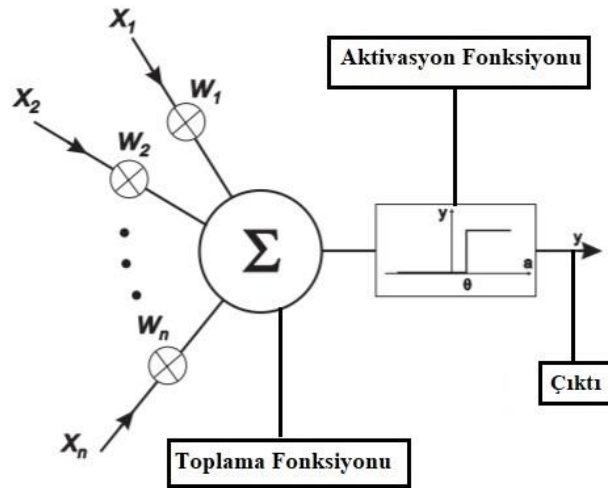
YSA'nın hesaplama ve bilgi işleme gücünü, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme sürecinde karşılaşılmayan veri setleri için de YSA'nın uygun tepkileri üretmesi olarak tanımlanır. Bu üstün özellikleri, YSA'nın karmaşık problemleri çözebilme yeteneğini gösterir. Günümüzde birçok bilim alanında YSA, aşağıdaki özellikleri nedeniyle etkin olmuş ve uygulama yeri bulmuştur (Ergezen vd. 2003):

- **Öğrenme;** YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA'nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemden aldığı verileri kullanarak problemi öğrenmelidir. Aksungur (2009)'a göre, "Yapay sinir ağlarının öğrenme özelliği, araştırmacıların dikkatini çeken en önemli özelliklerden birisidir. Çünkü herhangi bir olay hakkında girdi ve çıktılar arasındaki ilişkiyi, doğrusal olsun veya olmasın, elde bulunan mevcut örneklerden öğrenerek daha önce hiç görülmemiş olayları, önceki örneklerden çağrışım yaparak ilgili olaya çözümler üretebilme özelliği yapay sinir ağlarındaki zeki davranışın da temelini teşkil eder." diyerek yapay sinir ağının en ilgi çekici özelliklerinden birisi olan öğrenme özelliğinin önemini vurgulamıştır.
- **Genelleme;** Yapay sinir ağı, herhangi öğrenmiş olduğu problemden sonra eğitim sürecinde olmayan farklı test verileri içinde çözüm üretebilmektedir. Örneğin; karakter tanımlama yapılabilmesi amacıyla eğitilen bir ağ, bozulmuş veya parazitli karakterlerin girişinde de doğru olan karakteri verebilmektedir.
- **Uyarlanabilirlik;** Herhangi bir problemi çözmek amacıyla eğitilen bir ağ, problemde oluşan değişikliklere karşı ağırlıklarını düzenleyerek tekrar eğitilebilir ve bu değişimler sürekli ise değişim sırasında da eğitime devam edebilir. Sinyal işleme, sistem tanımlama gibi problemler söz konusu olduğunda kullanılabilir.
- **Hata Toleransı;** Yapay sinir ağı, oldukça fazla sayıda yapay hücrenin farklı şekillerde birbirlerine bağlanmasıyla paralel olarak dağılan bir yapıdadır ve bütün bilgi ağın bağlantılarının üzerinde dağılmış bir durumdadır. Bu sebeple, yapay bazı hücrelerin veya bazı bağlantıların eğitilmiş bir yapay sinir ağında kullanılmaması veya etkisiz bir duruma gelmesi, bu kurulan ağın probleme karşı doğru bir çözüm üretilmesini büyük bir oranda etkilememektedir ve bu sayede YSA, diğer klasik tekniklere göre oldukça yüksek hata toleransı yapabilme özelliğine sahiptir.

## 2.1.6 Yapay Sinir Ağlarının Temel Bileşenleri

Tek bir hücre, bazı basit bilgi işlem işlevlerini yerine getirebilmesine rağmen, sinirsel hesaplamaların gücü, bir ağdaki hücrelerin bağlanmasından kaynaklanır. Yapay sinir ağlarının varsayılan zekâsı bir tartışma meselesidir. Yapay sinir ağları nadiren birkaç yüz veya birkaç binden fazla işlem elemanına sahipken, insan beyninin yaklaşık 100 milyar hücresi vardır. İnsan beyni oldukça karmaşıktır ve bilimsel işlevlerinin çoğu ise hala bilinmemektedir. YSA, çok miktarda veriyi işleyebilir ve bazen şaşırtıcı derecede doğru olan tahminler yapabilir. Bu, onları her zamanki “insan” anlamında zeki yapmaz, bu yüzden bilgisayar zekâsı terimi bu sistemleri tanımlamanın daha iyi bir yolu olabilir. Şimdiye kadar tasarlanan birçok hücresel ağ tipi vardır, ancak hepsi hücrelerinin aktivasyon fonksiyonları, öğrenme kuralı ve bağlantı formülü ile tanımlanabilir (Agatonovic-Kustrin and Beresford 2000).

Şekil 2.3’de bir yapay sinir hücresine örnek verilmiştir. Görüldüğü gibi yapay sinir hücresinin 5 elemanı vardır. Girdiler ( $X_1, X_2, \dots, X_n$ ), ağırlıklar ( $W_1, W_2, \dots, W_n$ ),  $\Sigma$  (toplama-birleştirme (aggregation) fonksiyonu) fonksiyonu, aktivasyon fonksiyonu ve çıktıdan oluşmaktadır.



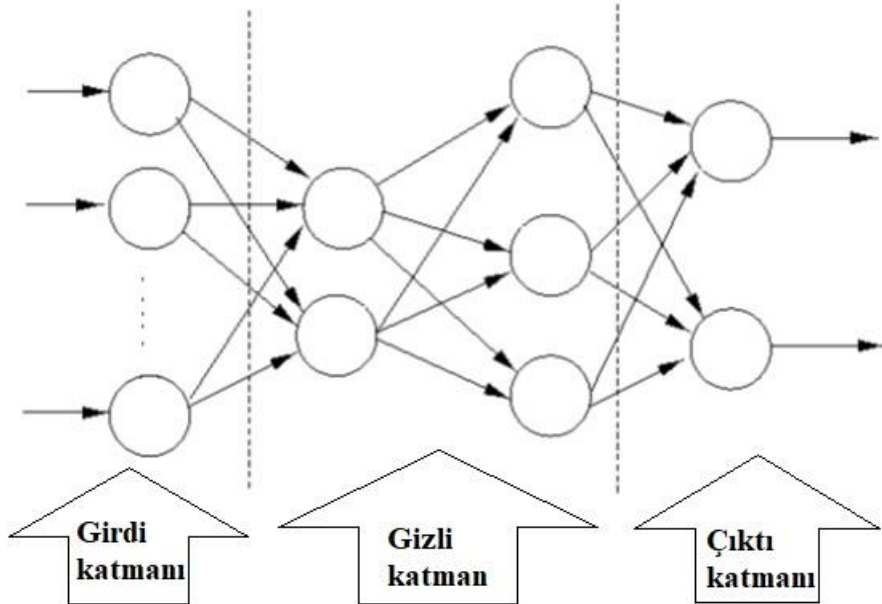
Şekil 2.3 Bir yapay sinir ağı hücresi (Gurney 1996)



### 2.1.6.1 Mimari Yapı

Yapay sinir ağı, yapay sinir hücrelerinin birbirleriyle bağlanmalarından oluşur. Önceki bölümde de bahsettiğimiz gibi çıktılar ağırlıklar yardımıyla diğer hücrelere ya da kendisine girdi olarak bağlanabilmektedir. Hücreler birbirlerine birçok şekilde bağlanabilir. Farklı öğrenme kurallarına göre ve aktivasyon fonksiyonlarına göre literatürde çeşitli yapay sinir ağı modelleri geliştirilmiştir.

En basit bir şekilde bir yapay sinir ağı modelinde tek katman ve tek sinir bulunmaktadır. Bu modellerde birden çok girdi ve tek bir çıktı bulunmaktadır ve çıktı değeri 1 ve 0 olmaktadır. Bu modeller çoğunlukla nesnelere iki sınıfa ayırmak için kullanılırlar. Sadece girdi katmanı ve çıktı katmanı olan ağı karmaşık problemleri çözebilmek konusunda yetersizlerdir. Bu tarz karmaşık problemler söz konusu olduğunda yapay sinir ağına en az bir gizli katman bulundurulması gereklidir (Elmas 2010). Bu nedenle, yapay sinir ağına, birbirleriyle bağlantılı sinirlerin yer aldığı girdi katmanı, gizli katman ve çıktı katmanı olmak üzere üç temel katman bulunmaktadır (Budak ve Erpolat 2012). Şekil 2.4’de yapay sinir ağına basit bir yapı verilmiştir.



Şekil 2.4 Örnek bir ileri beslemeli yapay sinir ağı mimarisi

En yaygın kullanılan YSA'larının (çok katmanlı Perseptron) mimarisinde, giriş katmanı, saklı katman ve çıkış katmanı yer almaktadır. Giriş ve çıkış katmanlarındaki hücre sayıları modellenmesi istenen veri kümesince belirlenir. Giriş katmanındaki hücre sayısı veri kümesinin özellik sayısı kadardır. Çıkış katmanı ise eğer veri kümesi bir sınıflandırma problemiye sınıf sayısı kadar, regresyon problemiye çıkış sayısı kadar hücre içerir. Ayrıca, giriş ve saklı katman, orijinden gerekli kaymaları ayarlamak için sapma hücreleri de içerir (Amasyalı 2015).

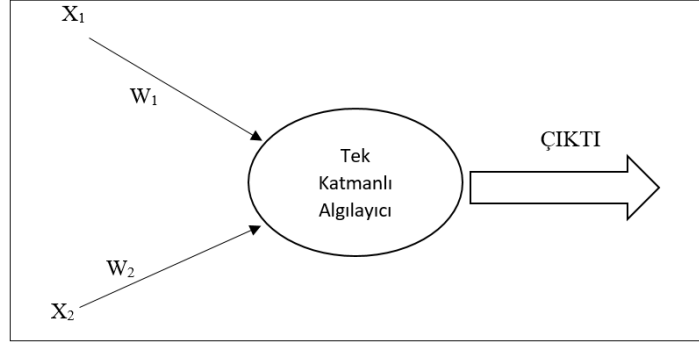
Bir YSA modelinde ilk katman olarak giriş katmanı vardır ve bu katman istatistikte bağımsız değişkenlere karşılık gelmektedir. Son katmanı ise çıktı katmanıdır ve istatistikte bu katman bağımlı değişkenlere karşılık gelmektedir. Yapay sinir ağı modelindeki diğer katman ise gizli katmandır. Bu katmanın işlevi; gizli katmanından gelen bilgileri sinyaller aracılığıyla çıkış katmanına iletmektir. Gizli katman, giriş katmanı ve çıkış katmanı arasında bulunur (Budak ve Erpolat 2012).

Yapay Sinir Ağları yapılarına göre iki sınıfta incelenebilir (Hamzaçebi 2011):

- i. *Otoasosyatif ağlar*, girdi hücrelerinin ayrıca çıktı hücreleri olarak görev aldığı ağlardır. Örnek olarak Hopfield ağları bu türdendir.
- ii. *Heteroasosyatif ağlar*, girdi ve çıktılarının farklı olduğu ağlardır. Algılayıcı, çok katmanlı algılayıcı ve Kohonen ağı bu türdendir.

#### ***i) Tek Katmanlı Algılayıcılar***

Tek katmanlı algılayıcıların yapısında gizli katman bulunmamaktadır. Girdi ve çıktı katmanından oluşurlar. Tek katmanlı algılayıcıyla kurulan herhangi bir ağın bir veya birden çok girdiye ve çıktıya sahip olabilir. Girdi birimleri (X), ağırlıklar (W) aracılığıyla Çıktı birimlerine (Ç) bağlanır. Bu tür ağlarda daima 1 olan bir eşik değeri bulunmaktadır ve görevi süreçteki elemanlarının değerinin sıfır olmasını engellemektir. (Adıyaman 2007). Şekil 2.5'de tek katmanlı algılayıcının mimarisi verilmiştir.



Şekil 2.5 Bir Tek Katmanlı Algılayıcı Mimarisi (Öztemel 2012)

Bu ağlarda çıktı fonksiyonu doğrusal bir fonksiyondur. Bu doğrusal fonksiyon, ağa girmiş olan girdi birimlerini iki sınıfa ayırmaya çalışan doğruyu bulmaya yaramaktadır. Eşik değer fonksiyonu da tam bu noktada gereklidir. Bu iki sınıfa ayırmaya çalışan doğrunun denklemi şu şekildedir:

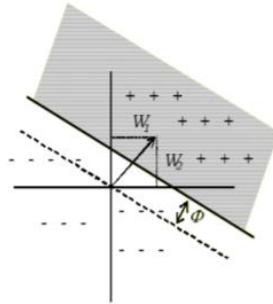
$$W_1 \cdot X_1 + W_2 \cdot X_2 + \Phi = 0 \quad (2.2)$$

Buradan (Arı ve Erşen 2017):

$$X_1 = \frac{W_2}{W_1} X_2 - \frac{\Phi}{W_1} \quad (2.3)$$

$$X_2 = \frac{W_1}{W_2} X_1 - \frac{\Phi}{W_2} \quad (2.4)$$

şeklinde elde edilir. Eşitlik 2.3 ve 2.4 kullanılarak Şekil 2.6'da ki gibi iki sınıfı ayıran doğru çizilebilir.



Şekil 2.6 İki sınıfı birbirinden ayıran doğrunun grafiği

Bu ağırlarda öğrenme: ayırıcı doğru iki sınıfı da en uygun şekilde ayırması sonucu gerçekleşir ve bu ayırma işleminin gerçekleşebilmesi için ağırlıkların değiştirilmesi gerekmektedir. Eşitlik 2.5’de görüldüğü gibi t zamanda  $\Delta W$  kadar bir ağırlık değişimi söz konusu olur ve ağırlık değeri değişmiş olur (Adıyaman 2007):

$$W_i(t + 1) = W_i(t) + \Delta W_i(t) \quad (2.5)$$

olur.

Burada her ağırlık değişimi, doğrunun eğiminin değişmesi anlamına gelmektedir ve bu da eşik değerin değişmesi anlamına da gelmektedir. Eşik değeri t zamanda  $\Delta \Phi$  kadar bir değişim gösterir ve değiştirilmiş yeni eşik değer Eşitlik 2.6’dan hesaplanır:

$$\Phi_i(t + 1) = \Phi_i(t) + \Delta \Phi_i(t) \quad (2.6)$$

olur.

#### ***a) Perceptron (Basit Algılayıcı Modeli)***

Perceptron ilk olarak 1958 yılında Rosenblatt’ın örüntü sınıflandırma amacıyla yaptığı bir çalışma ile geliştirilmiştir (Hopfield and Tank 1985). Tarihsel olarak muhtemelen önerilen en erken yapay hücre olan Perceptron, aynı zamanda neredeyse tüm YSA’ların temel yapı taşı olarak sayılmaktadır. Eşitlik 2.7’de görüldüğü gibi biyolojik hücrenin çok kaba fakat basit bir modelidir (Graupe 2007).

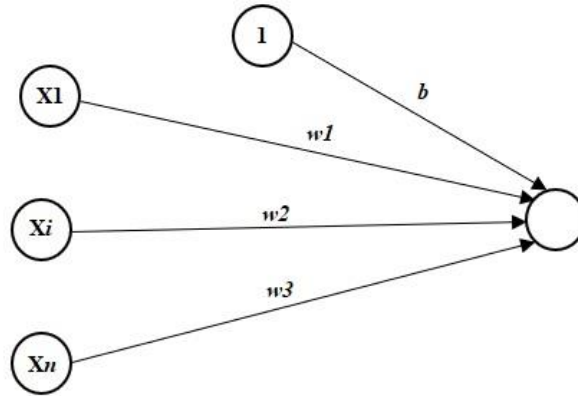
$$Z = \sum_i w_i x_i \quad (2.7)$$

$$y = f_N(z) \quad (2.8)$$

Perceptron, belki de ilk sinir ağlarının herhangi birinin en geniş kapsamlı etkisine sahiptir. Perceptron öğrenme kuralı, Hebb kuralından daha güçlü bir öğrenme kuralıdır. Uygun varsayımlar altında, yinelemeli öğrenme prosedürünün, doğru ağırlıklara, yani, ağların, her bir eğitim girdi modeli için doğru çıktı değerini üretmesine izin veren ağırlıklara

yaklaştığı kanıtlanabilir. Şaşırtıcı olmayan bir şekilde, gerekli varsayımlardan biri, böyle ağırlıkların mevcut olmasıdır. Özel bir basit algılayıcı, duysal ve birleştirme birimleri için ikili aktivasyonlar ve yanıt birimi için +1, 0 veya -1 aktivasyonunu kullanırlar. Duysal birimler, rastgele atanan +1, 0 veya -1 değerlerine sahip sabit ağırlıklar ile bağlantılar aracılığıyla birleştirici birimlere bağlanırlar (Fautsett 1994).

Orijinal basit algılayıcıdaki ilişkili birimlerden gelen çıktı ikili bir vektördür; bu vektör, takip eden bölümlerde çıkış birimine giriş sinyali olarak kabul edilir. Perceptron öğrenme kuralı yakınsama teoreminin kanıtına göre, çıktı birimine yardımcı birimlerin ayarlanabilmesi için, Şekil 2.7’de gösterilen ağıın tek katmanlı kısmıyla sınırlandırılır. Böylece, yardımcı üniteler, giriş birimleri gibi işlev görür ve mimari, şekillerde verildiği gibidir (Fautsett 1994).



Şekil 2.7. Tekli sınıflandırma yapmak için bir Perceptron yapısı

### ***b) Adaline Modeli (ALC)***

1950’li yılların sonlarına doğru Rosenblatt’ın basit algılayıcı modeli geliştirdiği zamanlarda, Bernard Widrow da yapay sinir ağları konusunda çalışmalarına başlamıştır. Widrow yanına bir yüksek lisans öğrencisi olan Marcian Hoff ile beraber 1960 yılında ADALINE ağı aracılığıyla “En Küçük Kareler” adlı algoritmayı geliştirmişlerdir. Bu geliştirilen model aslında Perceptron modeli ile aynıdır, fakat Perceptron eşik değer fonksiyonu kullanırken Adaline modeli doğrusal bir fonksiyon kullanmaktadır. Aynı Perceptron modelindeki gibi Adaline modeli de iki sınıfı birbirinden doğrusal bir şekilde

ayırmaya çalışmaktadır. Eğer bir problem doğrusal olarak ayrılamiyor ise bu iki model de etkinliğini yitirmektedir (Arı ve Erşen 2017).

Widrow-Hoff kuralı da denilen bu En Küçük Kareler algoritması, perceptronun öğrenme kuralından daha güçlüdür. Bu algoritma, kareli ortalamanın hatasını minimize etmeye çalıştığından, eğitim kalıplarını sınır çizgisinden uzak tutmaya çalışmaktadır. Fakat Perceptron sınır çizgisine yakınlığı yüzünden parazite (gürültü) duyarlı olabilmektedir (Arı ve Erşen 2017).

B. Widrow'un (1960) Adaline'ı, bipolar Perceptronun temel yapısına sahiptir. Yukarıdaki Eşitlik 2.7'yi ve bir tür En Küçük Kareler (LS) ağırlık eğitimi içerir. Burada girdi/katman ilişkilerine uyulmaktadır ve Eşitlik 2.9'da ki gibi hesaplanır (Graupe 2007):

$$z = w_0 + \sum_{i=1}^n w_i x_i \quad (2.9)$$

burada sapmalı bir terim olduğu ve bölümün eğitim prosedürüne tabi olduğu durumlarda “ALC'nin LMS eğitimi” veya “ALC'nin en dik iniş eğitimi” yöntemleri kullanılmaktadır. Eşitlik 2.8'in doğrusal olmayan elemanı burada, Adaline çıktısını  $y$  olarak veren basit bir eşik elemanıdır ve Eşitlik 2.10 aracılığıyla hesaplanır:

$$y = \text{sign}(z) \quad (2.10)$$

$$z = \sum_i w_i x_i \quad (2.11)$$

bunu elde ederiz

### Çizelge 2.2 Adaline Algoritması (Fautsett 1994)

---

#### *Adaline Algoritması*

---

**Adım 0.** Ağırlıkların başlangıç değeri atanır. Öğrenme oranı  $\alpha$  ayarlanır.

---

**Adım 1.** Durdurma koşulu yanlış olduğunda 2-6 adımları tekrar yapılır.

---

**Adım 2.** Her eğitim çifti için s:t, 2-5 adımları yapılır.

---

**Çizelge 2.2** (Devam) Adaline Algoritması (Fautsett 1994)

**Adım 3.** Giriş birimlerinin aktivasyonları ayarlanır,  $i=1, \dots, n$ :

$$x_i = s_i.$$

**Adım 4.** Çıkış ünitesinin hesaplama yanıtı:

$$y_{in} = b + \sum_i x_i w_i.$$

**Adım 5.** Sapma ve ağırlıklar güncellenir,  $i=1, \dots, n$ :

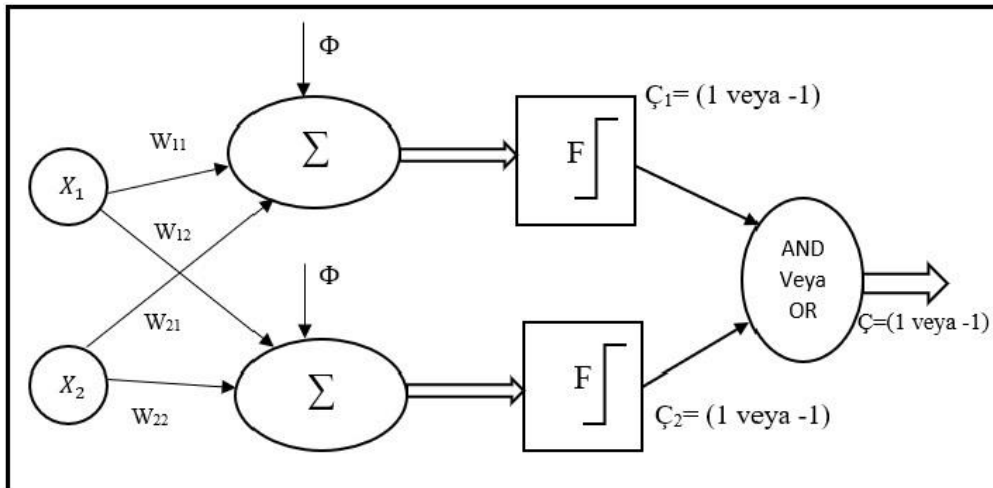
$$b(\text{yeni}) = b(\text{eski}) + \alpha(t - y_{in})$$

$$w_i(\text{yeni}) = w_i(\text{eski}) + \alpha(t - y_{in})x_i,$$

**Adım 6.** Test durdurma koşulu: adım 2'de en büyük ağırlık değişimi belirli bir tolerans değerinden daha küçük ise durdurulur, aksi halde algoritma çalışmaya devam eder.

**c) Madaline Modeli**

Widrow ve öğrencisi Hoff Adaline modelini geliştirmelerinden sonra MADALINE modelini de geliştirmişlerdir. Madaline modelinde birden fazla adaptif eleman bulunmaktadır. Madaline ağları birden fazla Adaline ünitelerinin birleşmesiyle oluşturulan ağa verilen isimdir. Madaline ağları genellikle iki katmandan oluşmaktadır. Her katmanda değişik sayıda Adaline ünitesi bulunmaktadır. Ağın çıktısı da 1 ve -1 değerleri ile gösterilmektedir. Her biri bir sınıfı temsil etmektedir. Şekil 2.8'de iki adaline ünitesinden oluşturulmuş bir Madaline ağı gösterilmektedir (Öztemel 2012).



**Şekil 2.8** İki Adaline ünitesinden oluşmuş Madaline ağı

Madaline eğitimi, Adaline eğitiminden farklıdır. Çünkü iç katmanların kısmi olarak istenen çıktılarının bulunamaması veya mevcut olması mümkün değildir. Böylece iç tabakalara gizli tabakalar denir. Tıpkı insan merkezi sinir sisteminde (CNS) olduğu gibi, bu öğrenmeye katılan bireysel hücrelerin sonuçları hakkında bilinçli olmasa da öğrenim bilgisi alınabilir. Ayrıca bu modelde hücrelerin iç tabakaları mevcuttur (Graupe 2007).

## ***ii) Çok Katmanlı Algılayıcılar***

Minsky ve Papert (1969) bir basit algılayıcı (Perseptron) modelinin XOR problemini çözemediğini göstermekle beraber, YSA'nın doğrusal olmayan problemlere bir çözüm getiremediğini iddia ederek araştırmacıların yaptığı araştırmaların durmasına neden olmuşlardır. Çünkü gerçek hayata bakıldığında günlük olayların hemen hemen hepsi doğrusal olmayan nitelikler taşımaktadır. Fakat buna rağmen birkaç araştırmacı çalışmalarına devam ederek XOR problemini çözmüşlerdir ve bu sayede araştırmacıların YSA konusunda tekrar çalışmalar yapmaya başlamasına sebep olmuşlardır. XOR probleminin çözülmesi amacıyla yapılan çalışmalar sonucunda Çok Katmanlı Algılayıcı Modeli (ÇKA) geliştirilmiştir. Rumelhart vd. (1986) tarafından geliştirilen bu modele *geriye yayılım modeli* veya *hata yayma modeli* (back-propagation) denilmektedir. ÇKA modeli sayesinde yapay sinir ağlarına olan ilgi hızlı bir şekilde artmıştır ve yapay sinir ağları konusunda yeni bir dönemin başlamasına neden olmuştur (Öztemel 2006).

Bazı zor ve çeşitli sorunları çözmek için hata geri yayılma algoritması olarak bilinen oldukça popüler bu algoritma ile denetimli bir şekilde eğiterek çok katmanlı algılayıcılar başarıyla uygulanmıştır. Bu algoritma, hata düzeltme öğrenme kuralını temel alır (Haykin 1999).

Temel olarak, hata geri yayılımının öğrenilmesi ağın farklı katmanlarından iki geçişten oluşur: ileri geçiş ve geri geçiş. İleri geçişte, ağın algısal hücrelerine bir aktivite modeli (giriş vektörü) uygulanır ve etkisi ağ katmanı boyunca katmandan yayılır. Son olarak, ağın gerçek tepkisi olarak bir dizi çıktı üretilir. Geriye doğru geçiş sırasında, diğer yandan, sinaptik ağırlıkların tümü bir hata düzeltme kuralına göre ayarlanır. Spesifik olarak, ağın gerçek tepkisi, bir hata sinyalini önceden oluşturmak için istenen (hedef) bir



yanıttan çıkarılır. Bu hata sinyali daha sonra ağ üzerinden sinaptik bağlantıların yönüne doğru yayılır, bu nedenle "hata geri yayılımı" adı verilir. Sinaptik ağırlıklar, ağın gerçek tepkisini istatistiksel anlamda istenen cevaba yaklaştırmak için ayarlanır. Hata geri yayılım algoritması, literatürde geri yayılma algoritması veya basitçe geri yayılma olarak da ifade edilir. Algoritma ile gerçekleştirilen öğrenme sürecine geri yayılma öğrenmesi denir. Çok katmanlı bir algılayıcının üç ayırt edici özelliği vardır (Haykin 1999):

1. Ağdaki her bir hücre modeli, doğrusal olmayan bir aktivasyon işlevi içerir. Burada vurgulanacak önemli nokta, doğrusal olmayanlığın Roseblatt'ın algılayıcısında kullanılan zor sınırlamanın aksine, pürüzsüz (yani her yerde farklılaşabilir) olmasıdır. Bu gerekliliği yerine getiren yaygın bir şekilde kullanılan doğrusal olmayanlık formu, lojistik işlevi tarafından tanımlanan bir sigmoidal olmayan doğrusallıktır ve Eşitlik 2.12'den hesaplanır:

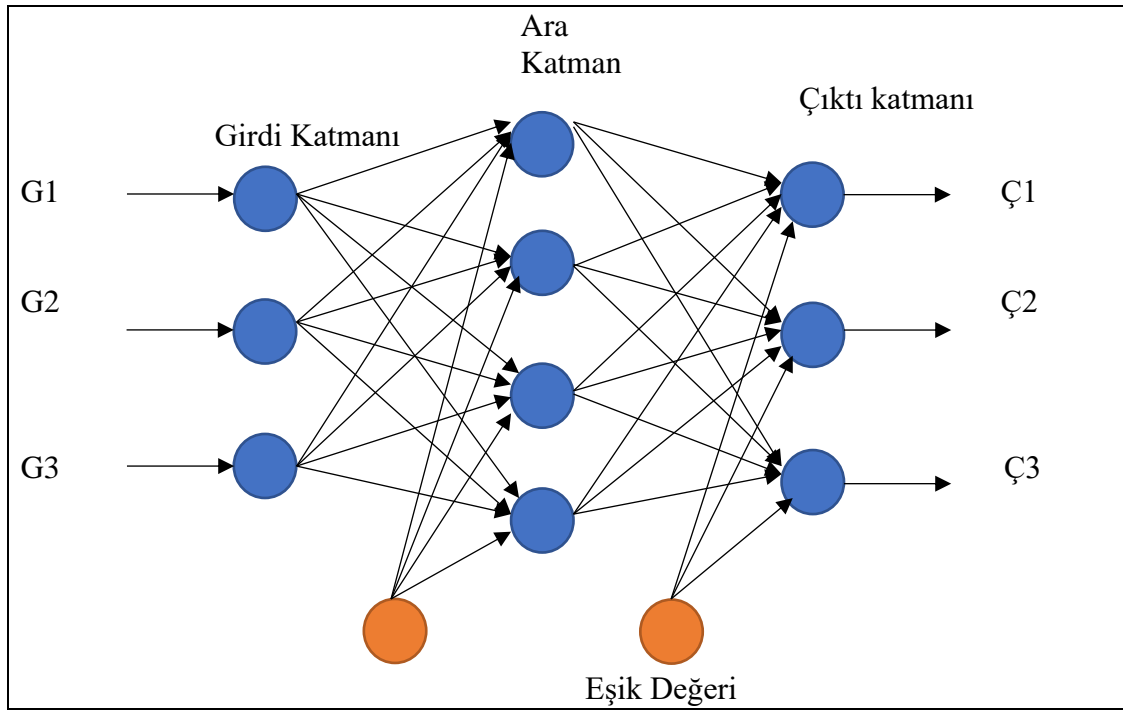
$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (2.12)$$

Burada  $V_j$ ,  $j$  hücresinin uyarılmış yerel alanıdır (yani, tüm sinaptik girdilerin ve toplam önyargının ağırlıklı toplamı) ve  $y_j$  ise hücresin çıktısıdır. Doğrusal olmayanların varlığı önemlidir, çünkü aksi takdirde ağın giriş-çıkış ilişkisi, tek katmanlı bir algılayıcıninkine indirgenebilir. Ayrıca, lojistik fonksiyonun kullanımı biyolojik olarak motive edilir, çünkü gerçek hücrelerin dayanıklılık kısmını hesaba katmaya çalışır.

2. Ağ, ağ girişinin veya çıkışının bir parçası olmayan bir veya daha fazla gizli hücre katmanları içerir. Bu gizli hücreler, giriş desenlerinden (vektörler) aşamalı olarak daha anlamlı özellikler çıkararak ağın karmaşık görevleri öğrenmesini sağlar.
3. Ağ, ağ sinapsları tarafından belirlenen, yüksek derecede bir bağlantı sergilemektedir. Ağın bağlantısındaki bir değişiklik, sinaptik bağlantıların sayısında veya ağırlıklarında bir değişiklik gerektirmektedir.

ÇKA'nın eğitim sürecinde girdi birimlerinin karşılığı olarak üretilmesi beklenen

çıktıların gösterildiği modeldir. Bu ağın amacı, ağa girmiş olan girdi birimlerine göre çıktıyı bulabilmektir. Ağın mimari yapısı ise Şekil 2.9'da gösterildiği gibi üç katmandan oluşmaktadır. Ağ giren bilgiler ağırlıklar ile çarpılarak gizli katmana iletilir. Burada gizli katman bir ya da birden fazla olabilmektedir ve bu katmanda ağ giren bilgiler işlenerek çıktı katmanına iletilir. Her bir girdiye karşılık bir çıktı değeri hesaplanır.



Şekil 2.9 Çok Katmanlı Bir Algılayıcı Ağının Yapısı

Çok katmanlı ağlarında girdi katmanında herhangi bir bilgi işleme yapılmamaktadır. Buradaki işlem görecekleman sayısı doğrudan girdi biriminin sayısına bağlıdır. Gizli katmanın sayısı ve bu katmandaki işlem göreceklemanların sayısı ise deneme-yanılma yolu ile bulunabilmektedir. Bu ağlarda bir danışmalı öğrenme kuralı aracılığıyla ağa bir veri seti gösterilerek, bu veri setine nasıl bir sonuç üretmesi gerektiği ifade edilir. Bu veri seti girdi katmanından ağa girer, gizli katmanda işlem görür ve çıkış katmanından da çıktı olarak sonuç elde edilmiş olur (Doğan 2010).

Çağdaş YSA'da iki temel mimari vardır. Bunlar ileri beslemeli ve geri beslemeli ağlardır. Geri beslemeli ağlar, optimizasyon problemleri için çözüm aramalarında kullanılır. İleri

beslemeli ağırlar ise ise değişkenler arasındaki ilişkileri tanımlamak için yaygın olarak kullanılır (Kocak 2015). İlk olarak ileri doğru hesaplama yaparken girdi katmanında bulunan eleman sayısı  $k$  olsun ve girdi katmanında bulunan süreç elemanlarının çıktı değeri Eşitlik 2.13 ile hesaplanır (Alpaydın 2011, Öztemel 2012, Gör 2016):

$$\zeta_k^i = G \quad (2.13)$$

Gizli katmandaki süreç elemanlarının  $NET$  girdi değerlerinin bulunması,  $A_{kj}$   $k$ . girdi elemanının  $j$ . gizli katman elemanına bağlanmış olan ağırlık değeri olsun ve Eşitlik 2.14 aracılığıyla hesaplanır:

$$NET_j^a = \sum_{k=1}^n A_{kj} \zeta_k^i \quad (2.14)$$

Gizli katmandaki girdilerin  $NET$  değerleriyle aktivasyon fonksiyonu kullanılarak gizli katman elemanı  $j$  için çıktı hesaplanır. Aktivasyon fonksiyonunun seçilmesi konusunda herhangi bir formül bulunmamakla birlikte, bu konuda ki çalışmalara bakıldığında genelde sigmoid fonksiyonun tercih edildiği görülmüştür. Burada sigmoid fonksiyon kullanılması durumunda gizli katmanın elemanı olan  $j$  için bir çıktı değeri Eşitlik 2.15 aracılığıyla hesaplanır:

$$\zeta_j^a = \frac{1}{1 + e^{-(Net_j^a + \beta_j^a)}} \quad (2.15)$$

Eşitlik 2.31'de  $\beta_j^a$  olarak hesaplanan değer,  $j$ . elemanla ilişkili olan eşik değer ögesinin ağırlığını göstermektedir. Buradaki eşik değer ögesinin çıktısı 1'dir ve sabit bir değerdir.

Geriye doğru hesaplama yapılırken ise ilk olarak hata değeri hesaplanır. Beklenen değerler ( $E_1, E_2, E_3 \dots$ ) ve çıktı değerleri ( $O_1, O_2, O_3 \dots$ ) olduğu kabul edilirse, hata değerleri ( $H_1, H_2, H_3 \dots$ ) Eşitlik 2.16 aracılığıyla hesaplanır:

$$H_m = E_m - O_m \quad (2.16)$$

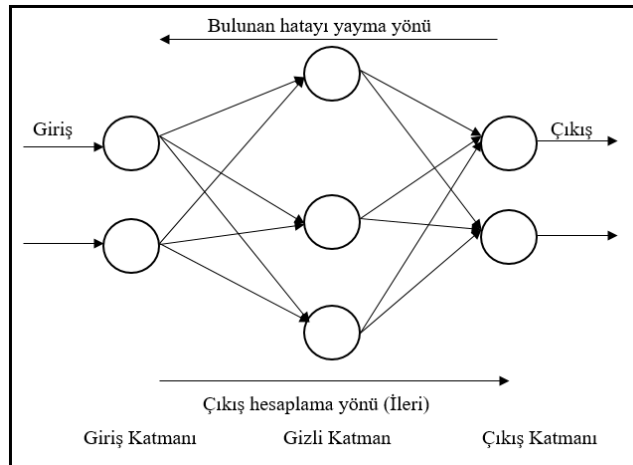
olarak hesaplanır ve ağıın toplam hatası ( $TH$ ) ise Eşitlik 2.17 yardımıyla hesaplanır:

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (2.17)$$

olarak hesaplanır. Çok katmanlı bir algılayıcının eğitimi gerçekleştirilirken her istatistiksel model gibi hatanın minimum değere indirilmesi amaçlanır. Hesaplamadan elde edilen hata değeri ağıdaki ağırlık değerlerine paylaştırılır. Sonrasında gizli katman-çıkış katmanı, gizli katman-gizli katman ve gizli katman-giriş katmanı aralarındaki ağırlıklar güncellenir. Bu güncelleme ile bütün ağırlık değerleri değiştirilmiş olur. Bu şekilde ilk iterasyon bir örnek için gerçekleştirilmiş olur. Daha sonra ikinci iterasyonun gerçekleşmesi için başka bir örnek ağı girer ve ilk iterasyondaki işlemler bu örnek için de tekrarlanır (Alpaydın 2011, Öztemel 2012 akt.: Gör 2016).

#### a) İleri Beslemeli Ağlar (*Feed-Forward Nets*)

İleri beslemeli yapay sinir ağlarında hücreler girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. Yapay sinir ağının girişine gelen bilgiler bir değişime uğratılmadan orta noktaya diğer bir deyişle gizli katmandaki hücrelere iletilir. Daha sonra sırasıyla çıkış katmanından işlenerek geçer ve dış ortama aktarılır. Şekil 2.10'da ileri beslemeli ağ yapısının mimarisine örnek verilmiştir (Öztürk ve Şahin 2018).



Şekil 2.10 İleri Beslemeli Ağ Yapısı

Genel tanım olarak ileri beslemeli bir sinir ağının, hücreleri hesaplama birimleri olan ve yönlendirilmiş kenarları hücreden hücreye sayısal bilgi ileten bir hesaplama grafiği olduğu söylenebilir. Her hesaplama birimi, girişinin tek bir ilkel işlevini değerlendirebilmektedir. Aslında ağ, bir girdiyi bir çıktı vektörüne (model olarak adlandırılır) dönüştüren bir fonksiyon bileşimleri zincirini temsil etmektedir. Ağ, girişten çıkış alanına ağ işlevi olarak adlandırılan bileşik bir fonksiyonun özel bir uygulamasıdır. Öğrenme problemi, ağırlıkların en uygun kombinasyonunu bulmaktan oluşur, böylece ağ fonksiyonu  $\mathcal{F}$  verilen bir  $f$  fonksiyonuna mümkün olduğunca yaklaşır (Rojas 1996).

Rojas (1996)'ya göre "Herhangi bir sayıda gizli birimden oluşan ve istenen ileri beslemeli bağlantı düzeninin sergileyebilen  $n$  giriş ve  $m$  çıkış birimleri bulunan ileri beslemeli bir ağ düşünülün. Ayrıca  $\{(x_1, t_1), \dots, (x_p, t_p)\}$  şeklindeki bir eğitim seti verilsin, giriş ve çıkış kalıpları olarak adlandırılan, sıralı  $n$  ve  $m$  boyutlu  $\mathcal{F}$  vektör çiftlerinden oluşur. Ağın her hücresindeki ilkel işlevlerin sürekli ve farklı olmasına izin versin ve kenarların ağırlıkları rastgele seçilen gerçek sayılar olsun. Eğitim setinden giriş verisi  $x_i$  bu ağa sunulduğunda, genel olarak hedef  $t_i$ 'den farklı bir  $o_i$  çıkışı üretir. Burada olması istenilen  $o_i$  ve  $t_i$ 'yi  $i=1, \dots, p$  için aynı yapmaktır. Ağın hata işlevini en aza indirmek istenirse Eşitlik 2.18'de ki gibi hesaplanır:

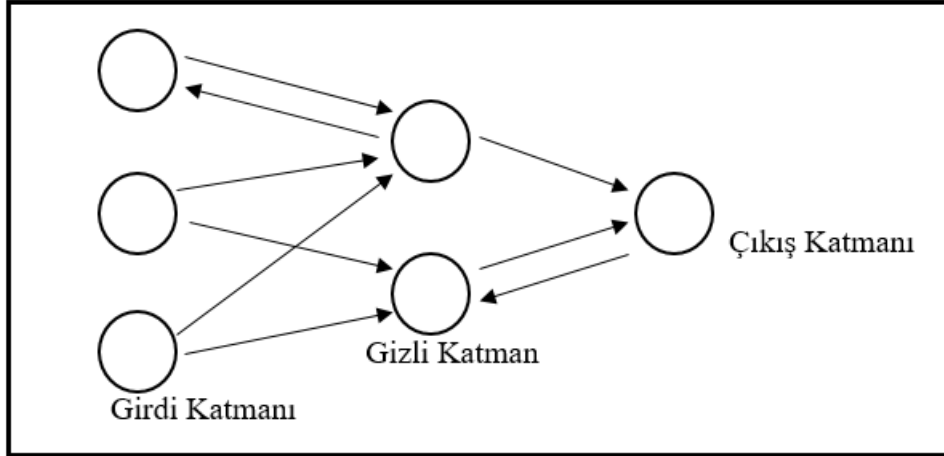
$$E = \frac{1}{2} \sum_{i=1}^p \|o_i - t_i\|^2 \quad (2.18)$$

olur ve eğitim seti için bu işlevi en aza indirdikten sonra, ağa yeni bilinmeyen giriş kalıpları sunulur ve enterpolasyon yapması beklenir. Böylece ağ, yeni bir giriş vektörünün öğrenilen kalıplara benzer olup olmadığını algılamalı ve benzer bir çıktı üretmelidir.

### ***b) Geri Beslemeli Ağlar***

Geri beslemeli bir yapay sinir ağı modelinde, çoğunlukla geri besleme geciktirme ögesi sayesinde bir (veya daha fazla) hücrenin çıktısının kendine veya başka hücrelere girdi olarak gösterilmesiyle gerçekleşir. Bu geri besleme sistemi bir katmanın elemanı olan

yapar hücreler arasında da olabilir veya katmanların arasındaki yapay hücreler arasında da olabilmektedir ve bu işleyişinden dolayı dinamik bir hareket göstermektedir. Yani, hücreler arasındaki bağlantı şekillerine göre farklı ağ yapıları elde edilebilmektedir. Şekilde 2.11’de geri beslemeli bir ağ yapısı verilmiştir (İnt.Kay.2).

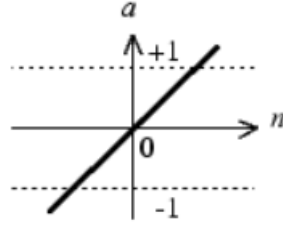


Şekil 2.11 Geri Beslemeli Ağ Yapısı

### 2.1.6.2 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonundan iletilen net girdi işlenerek bu girdiye karşılık üretilecek çıktı belirlenir. Bu fonksiyon eşik fonksiyonu olarak da adlandırılır. Aktivasyon fonksiyonunda olduğu gibi bu fonksiyonda da çıktı hesaplaması için farklı formüller vardır. Fakat bazı modeller aktivasyon fonksiyonunun türevlenebilir olması şartı koşmaktadır. Fonksiyonların hiçbirisinin birbirlerine göre üstünlükleri yoktur. Herhangi bir problem için en uygun fonksiyon ancak denemeler sonucu öğrenilebilir. Uygun fonksiyonu bulmak için herhangi bir formül yoktur. Bu konuda ki çalışmalara bakıldığında, günümüzde araştırmacıların genelde “Sigmoid fonksiyon” kullandıkları görülmektedir. Bazı aktivasyon fonksiyonları aşağıda verilmiştir (Öztemel 2006) (Aktivasyon fonksiyonlarının  $[0, 1]$  ve  $[0, -1]$  aralığında olanlarına tek kutuplu,  $[-1, +1]$  aralığında olanlara ise çift kutuplu denilmektedir):

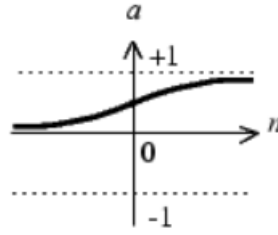
**Purelin (Lineer) Fonksiyon:** Bu fonksiyonun çıktısının değişim aralığının  $[-1, 1]$  aralığında ve doğrusal bir değişim göstermektedir. Bu fonksiyonun grafiği Şekil 2.12’de ve formülü ise Eşitlik 2.19’da verilmiştir.



Şekil 2.12 Purelin Fonksiyonu çıktı grafiği

$$a = \text{Purelin}(n) = n \quad (2.19)$$

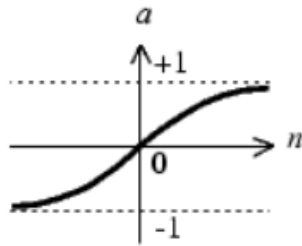
**Logsig (Sigmoid) Fonksiyon:** Bu fonksiyonun çıktısının değişim aralığı [0 1] aralığındadır. Bu fonksiyonun grafiği Şekil 2.13’de ve formülü ise Eşitlik 2.20’de verilmiştir.



Şekil 2.13 Sigmoid fonksiyon giriş-çıkış grafiği

$$a = \frac{1}{1 + e^{-n}} \quad (2.20)$$

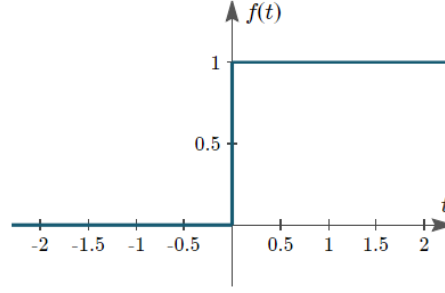
**Tansig Fonksiyon:** Bu fonksiyonun çıktısının değişim aralığı [-1 1] aralığında ve doğrusal olmayan bir değişim göstermektedir. Bu fonksiyonun grafiği Şekil 2.14’de ve formülü ise Eşitlik 2.21’de verilmiştir.



Şekil 2.14 Tansig fonksiyonun girdi-çıkış grafiği

$$a = \frac{2}{1 + e^{(-2n)}} - 1 \quad (2.21)$$

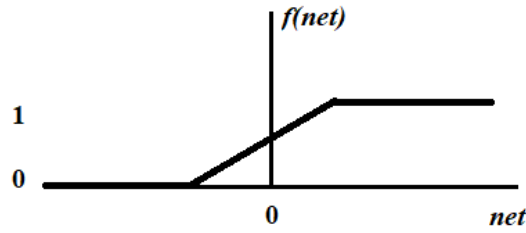
**Step (Adım) Fonksiyonu:** Bu fonksiyon, net girdinin belirli bir eşit değer altında veya üstünde olmasına göre çıktı 0 veya 1 değerini alır. Bu fonksiyonun grafiği Şekil 2.15’de ve formülü ise Eşitlik 2.22’de verilmiştir.



Şekil 2.15 Step (Adım) fonksiyonu çıktı grafiği

$$F(Net) = \begin{cases} 1, & \text{if } Net > \text{Eşik Değer} \\ 0, & \text{if } Net \leq \text{Eşik Değer} \end{cases} \quad (2.22)$$

**Eşik Değer Fonksiyonu:** Bu fonksiyon, gelen girdilerin NET değerlerinin 0’dan ya da 1’den küçük-büyük olması durumuna göre bir değer alır. Bu fonksiyon 0-1 arasında değerler alabilir. Bu fonksiyonun grafiği Şekil 2.16’da ve formülü ise 2.23’de verilmiştir.

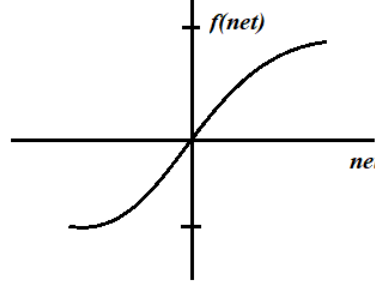


Şekil 2.16 Eşik değer fonksiyonu çıktı grafiği

$$f(net) = \begin{cases} 0, & \text{if } Net \leq 0 \\ Net, & \text{if } 0 < Net < 1 \\ 1, & \text{if } Net \geq 1 \end{cases} \quad (2.23)$$



**Hiperbolik Tanjant Fonksiyonu:** Gelen net girdi değerlerinin tanjant fonksiyonundan geçmesiyle hesaplanır. Bu fonksiyonun grafiği Şekil 2.17’de ve formülü ise Eşitlik 2.24’de verilmiştir.



Şekil 2.17 Hiperbolik tanjant fonksiyonu çıktı grafiği

$$f(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \quad (2.24)$$

### 2.1.6.3 Yapay Sinir Ağlarında Öğrenme

Hem insan hem de hayvan davranışlarının bir örneği olarak öğrenme, psikologlar, fizyologlar ve diğerler araştırmacılar tarafından uzun bir deney ve çalışma konusu olmuştur; Bununla birlikte, deneysel ve gözlemsel sonuçlara dayanan bilimsel öğrenme çalışması, temel olarak son 80 yılda psikologlar tarafından gerçekleştirilmiştir. Öğrenme ve öğrenme teorilerinin psikolojideki önemi, insanın çok çeşitli davranışlarını tanımlama üzerindeki etkisinden kaynaklanmaktadır. Buradan yola çıkarak, öğrenme sistemleri, kontrol sistemlerinin ve örüntü tanıma sistemlerinin sentezine formüle edilmiş ve uygulanmış ve bu sistemlerin öğrenmenin niteliklerine sahip olduğu söylenebilmektedir (Mendel and McLaren 1970).

Öğrenme becerisini elde etmenin anahtarı, ağ yapısındaki bağlantıların dağılış biçiminin değiştirilmesinde yatmaktadır (Görz and Nebel 2003, Pozan 2006). İnsan sinir sisteminin çalışmasını taklit eden YSA, eldeki probleme ilişkin verileri kullanarak veri yapısında saklı ilişkileri ortaya çıkarmaya çalışır. Bu işlem, ağın öğrenmesi olarak adlandırılır (Hamzaçebi 2011). YSA’ların eğitimi, içerdikleri hücreler arası ağırlık değerlerinin, eğitim kümesi girişlerini, çıkışlarına dönüştürecek şekilde ayarlanmasıdır (Amasyalı 2015). Bir ağ belirli bir uygulama için yapılandırıldıktan sonra, bu ağ eğitime hazırdır.

Bu sürece başlamak için başlangıç ağırlıkları rastgele seçilir. Sonra, eğitime veya öğrenme başlar (Anderson and McNeill 1992).

Formülleştirecek olursak, herhangi bir ( $w$ ) ağırlığı için Eşitlik 2.25 ile hesaplanır:

$$w_{yeni} = w_{eski} + \Delta w \quad (2.25)$$

Bu denklem yapay sinir ağının öğrenme aşamasının matematiksel bir biçimde nasıl gerçekleştiğini göstermektedir. Eşitlik 2.25'deki  $\Delta w$  değeri ağırlık değerlerinin değişimindeki miktarı vermektedir. Buradaki  $\Delta w$ 'nin belirlenmesi amacıyla geliştirilmiş algoritmalar vardır ve bu algoritmalara öğrenme algoritmaları denilmektedir. Hangi ağırlık kümesinin en iyisini temsil ettiğini bulmaya yönelik geliştirilen birçok öğrenme algoritması sunulmuştur (Hamzaçebi 2011). Bu algoritmalar kabaca iki sınıfta incelenebilir; danışmanlı-denetimli (supervised) ve danışmansız-denetimsiz (unsupervised) öğrenmedir (Ataseven 2013). Denetimli eğitim, ağın istenen performansı elde etmek için, ağın performansını istenen çıktılarını girdi ile sağlayarak, istenen çıktıya sahip olma mekanizmasını içerir. Denetimsiz eğitim, ağın dışarıdan yardım almadan girdileri anlamlı olması gerektiğidir. Ağların büyük bir çoğunluğu denetimli eğitimden yararlanmaktadır. Denetimsiz eğitim, girdilerde bazı başlangıç karakterizasyonlarını gerçekleştirmek için kullanılır (Anderson and McNeill 1992).

#### ***a) Danışmanlı-Denetimli-Öğretmenli Öğrenme***

Yapay sinir ağlarında gerçek çıktı istenen çıktıyla kıyaslanır. Rasgele değişen ağırlıklar ağ tarafından öyle ayarlanır ki, bir sonraki döngüde gerçek çıkış ile istenen çıkış arasında daha yakın karşılaştırma üretebilsin. Öğrenme yöntemi, ağ üzerinde yayılmış olan bağlantıların anlık hatalarını en aza indirmeye çalışır. Bu hata azaltma işlemi, kabul edilebilir doğruluğa ulaşana kadar ağırlıklar devamlı olarak güncellenir (Elmas 2003).

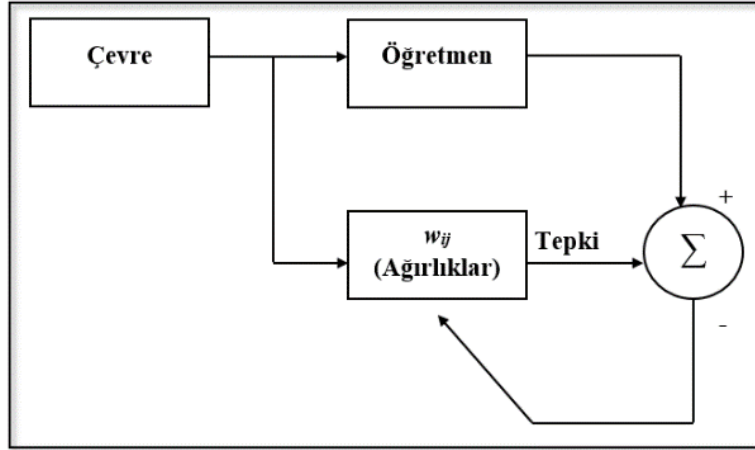
Denetimli öğrenme, bir yapay sinir ağının parametrelerini eğitim verilerinden ayarlayan bir makine öğrenme tekniğidir. YSA'da öğrenme, çıktı değerini gördükten sonra herhangi bir geçerli girdi değeri için parametrelerinin değerini belirlemektir. Ağdaki eğitim verileri, girdi değerlerinden ve çıktı değerlerinden oluşmaktadır. Denetimli öğrenim, her biri güçlü ve zayıf yönleri olan çok çeşitli sınıflandırıcılara sahip olduğu sınıflandırma

olarak da ifade edilebilir. Belirli bir problem için uygun bir sınıflandırıcı (Çok katmanlı algılayıcı, Destek Vektör Makineleri, k-en yakın komşu algoritması, Gaussian karışım modeli, Gauss, Bayes, karar ağacı, radyal temel fonksiyon sınıflandırıcıları, ...) seçmek ağın performansını doğrudan etkilediği için uygun sınıflandırıcıya karar vermek oldukça önemlidir (Krenker *et al.* 2011).

Belirli bir denetimli öğrenme probleminin çözülebilmesi için bazı aşamaların gerçekleştirilmesi gerekmektedir (Krenker *et al.* 2011):

1. İlk adımda eğitim örneklerinin türü belirlenmelidir.
2. İkinci adımda, belirli bir problemi tatmin edici bir şekilde tanımlayan bir eğitim veri seti oluşturulması gereklidir.
3. Üçüncü adımda, seçilen yapay sinir ağına anlaşılabilir bir şekilde toplanmış eğitim verisini tanımlanması gereklidir.
4. Dördüncü aşamada ise, öğrenme yapılır ve öğrendikten sonra, öğrenilmiş yapay sinir ağının performansını test (doğrulama) veri seti ile test edilir. Test veri seti, öğrenme sırasında yapay sinir ağına tanıtılmamış verilerden oluşur.

Danışmanlı öğrenmede giriş ve çıkış çiftlerinden oluşan eğitim bilgileri vardır. Ağ giriş bilgisine göre ürettiği çıkış değerini, istenen değerle karşılaştırarak ağırlıkların değişmesinde kullanılacak bilgiyi elde eder. Girilen değerle istenen değer arasındaki fark hata değeri olarak önceden belirlenen değerlerden küçük oluncaya kadar eğitime devam edilir. Hata değeri istenen değer altına düştüğünde tüm ağırlıklar sabitlenerek eğitim işlemi sonlandırılır. Eğitim işlemi sırasında her bir eğitim verisi için oluşan hata değerine göre ağırlıkların değiştirilmesine “örüntü kipi” öğrenme, tüm eğitim kümesi için hataların toplanarak toplam hata değerine göre ağırlıkların değiştirilmesine ise “küme tipi” öğrenme denilmektedir. Danışmanlı öğrenme Şekil 18’de gösterilmiştir (Elmas 2003).



Şekil 2.18 Danışmanlı öğrenme

Verilen girişe karşılık istenilen bir çıkış üretilmediyse, ağırlık çıkış değerinin hatasının en küçük olacak şekilde ağırlıklarının değiştirilmesi sağlanır.

#### ***b) Danışmansız-Denetimsiz-Öğretmensiz Öğrenme***

Denetimli öğrenmede amaç, girdiyle doğru değeri bir gözetmen tarafından verilen çıktı arasında bir eşleme öğrenmektir. Denetimsiz öğrenmede böyle bir gözetmen yoktur ve elimizde sadece girdi verisi vardır; amaç, girdideki düzenlilikleri bulmaktır. Girdi uzayının bir yapısı vardır, bazı örneklerde daha sık rastlanır; istediğimiz neyin sık, neyin seyrek gerçekleştiğini görmektir. İstatistikte buna “dağılım kestirimi” denir (Alpaydın 2011).

Bu tür öğrenme yöntemlerinde, sistemin öğrenmesine yardımcı olan herhangi bir öğretmen yoktur. Sisteme sadece girdi değerleri gösterilir. Örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendisine öğrenmesi beklenir. Bu, daha çok sınıflandırma problemleri için kullanılmaktadır. Yalnız sistemin öğrenmesi bittikten sonra çıktıların ne anlama geldiğini gösteren etiketlendirmenin kullanıcı tarafından yapılması gerekmektedir (Öztemel 2012). Bu öğrenme yönteminde en çok kullanılan yöntemler, Adaptif Rezonans Teorisi (ART), Hopfield Ağı, Kohonen Ağı'dır.

Denetimsiz öğrenmede, ağırlıklar ve sapmalar sadece ağ girişlerine yanıt olarak değiştirilir. Mevcut bir hedef çıkışları yoktur. İlk bakışta bu pratik olmayabilir. Ne

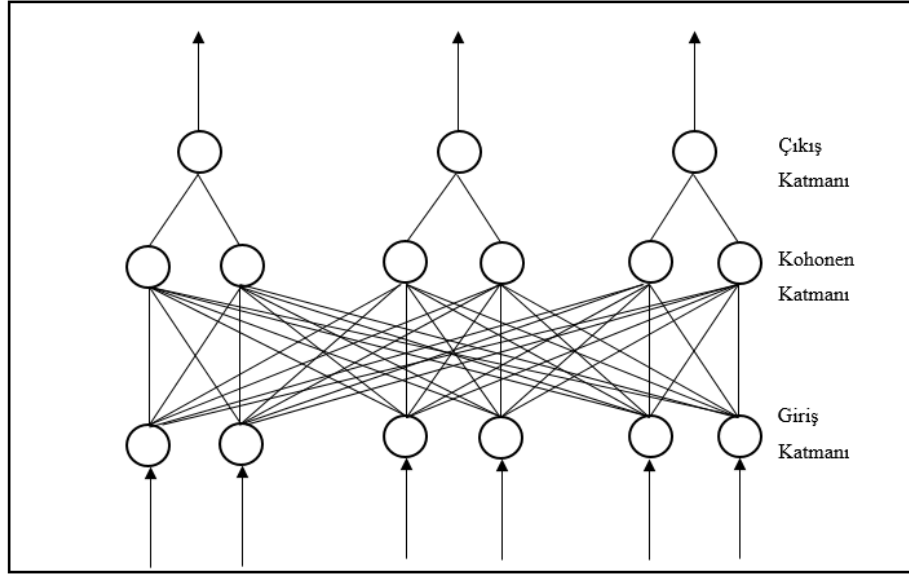
yapması gerektiği bilinmiyorsa bir ağı eğitmek mümkün olmayabilir. Bu algoritmaların çoğu, bir çeşit küme işlemini gerçekleştirir. Giriş değerlerini sınırlı sayıda sınıfa ayırmayı öğrenirler. Bu, özellikleri sayesinde vektör niceleme gibi uygulamalarda da yararlıdırlar (İnt.Kay.1).

### c) *Destekleyici-Takviyeli Öğrenme*

Öğretmenli öğrenme sisteminde olduğu gibi ağı bir öğretmen yardımcı olur. Ancak burada öğretmen, olması gereken çıktıları sisteme tanıtmaz. Bir girdi vektörüne karşılık ağın ürettiği çıktı için olumlu ya da olumsuz sinyal üretilerek ağ ağırlıklarının düzenlenmesi sağlanır (Hamzaçebi 2011). Her bir ağ girdisi için doğru çıktı ile sağlanmak yerine, algoritmaya yalnızca bir not verilir. Derece (veya skor), bazı giriş dizileri üzerindeki ağ performansının bir ölçüsüdür. Bu öğrenme türü şu anda denetimli öğrenmeden çok daha az yaygındır (İnt.Kay.1).

Yılmaz (2017) ise takviyeli öğrenmeyi şöyle açıklamıştır: “Takviyeli öğrenme sistemlerinin danışmalı ve danışmansız öğrenme yöntemlerine benzer yanları bulunmaktadır. Sisteme danışmansız öğrenme yöntemlerinde olduğu gibi probleme ait sadece giriş bilgileri verilmektedir. Fakat sistem için bir danışman bulunarak sistem çıkışının doğru ve yanlış olduğunu göstermektedir. Bu yönden de çıkış bilgileri verilmesi de çıkışın doğru hesaplanması için danışman tarafından takviye yapılmış olmaktadır. Dolayısıyla danışman öğrenen sisteme yardımcı olmaktadır.”

Takviyeli öğrenme yöntemini kullanan ağlardan birisi LVQ (Linear Vector Quantization – Doğrusal Vektör Nicelemesi)’dur. LVQ ağlarının temelinde n boyutlu bir vektörü parçalayarak vektörler setine uyarlamak yatmaktadır. LVQ ağları daha çok sınıflandırma problemleri için uygulanmaktadır. Ağın ürettiği çıkışlardan sadece biri 1 değerini alırken, diğerleri 0 değerini almaktadır.1 üretilen çıkış, giriş verisinin hangi vektör sınıfına ait olduğunu işaret etmektedir. LVQ ağları 3 katmandan oluşmaktadır. Şekil 1’de LVQ ağı örneği verilmiştir (Yılmaz 2017).



**Şekil 2.19** LVQ Ağ Mimarisi

*Giriş katmanında* herhangi bir bilgi işleme olmamaktadır. Sisteme başka bir ağdan veya dış dünyadan verinin sunulması bu katman üzerinden olmaktadır. *Kohonen katmanı* ise bir ara katman gibi işlev görmektedir. Bu katmandaki her eleman referans vektörünü temsil etmektedir. Bu sayede giriş katmanından gelen verilere en yakın ağırlık vektörü burada belirlenir. *Çıkış katmanında* ise girişin sınıflandırması yapılmaktadır (Yılmaz 2017).

Yapay Sinir Ağının parametrelerinin değerlerinin değişmesi belirli bir kurallara göre gerçekleşmektedir. Bu kurallara öğrenme kuralları (öğrenme algoritmaları) denir. Farklı birçok yöntemle göre değişik öğrenme kuralları vardır. Bilinen bazı öğrenme kuralları ve tanımları aşağıda verilmiştir.

#### **a) Hebb Öğrenme Kuralı**

Hebb öğrenme kuralı hakkında detaylı bilgi için Donald Hebb'in 1949 yılında yazmış olduğu "The Organization of Behaviour" adlı kitabından yararlanılabilir. Bu kural basit bir şekilde özetlenecek olursa; sinyal alan yapay hücre ile sinyal veren yapay hücre arasındaki bağlantı ağırlıklarının ikisi de aynı işarete (matematiksel olarak) sahipse, bu

iki yapay hücre arasındaki bağ güçlendirilmedi, fakat zıt bir durum söz konusuysa bu bağın gücü azaltılmalıdır (Çelik 2008).

Donald Hebb tarafından geliştirilen bu kural bilinen en eski öğrenme kuralıdır. Hebb kuralı diğer öğrenme kurallarının temelini oluşturarak, öğrenme kurallarının en başında gelmektedir. Kısaca Hebb kuralına göre; hücrenin çıktısı ile öğrenme sinyali birbirlerine eşittir. Eğer bir yapay hücre aktif ise, bu hücrenin bağlandığı başka bir hücreyi de aktif yapar. Aynı şekilde hücre eğer pasif ise bağlandığı hücreyi de pasif yapma amacındadır.

### ***b) Hopfield Öğrenme Kuralı***

Hebb kuralına benzer. YSA elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi veya zayıflatılması gerektiği belirlenir. Eğer beklenen çıkış ve girişlerin her ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerlerini kuvvetlendir/zayıflat denmektedir. Yani ağırlıkların kuvvetlendirilmesi veya zayıflatılması öğrenme katsayısı yardımıyla gerçekleştirilmektedir. Öğrenme katsayısı genellikle 0-1 arasında kullanıcı tarafından atanan sabit ve pozitif bir değerdir (Efendigil 2008).

### ***c) Delta (Geri Yayılım) Öğrenme Kuralı***

Hebb kuralının gelişmiş bir şekli olarak Widrow ve Hoff tarafından delta öğrenme kuralı geliştirilmiştir. Öğrenme kurallarından en çok kullanılan kurallardan biridir ve yapay hücrenin gerçek çıktısıyla beklenen çıktı arasındaki farkı düşürmek için ağırlık elemanları arasındaki bağlantıların ağırlık değerlerini sürekli değiştirmesi prensibine dayanır. Bu kural, gerçek çıktı ile beklenen çıktı arasındaki hatanın karesinin en aza indirilmesini amaçlamaktadır. Bu nedenle bu kurala “En Küçük Kareler Kuralı” da denilmektedir. Bunlara ek olarak, geri yayılım ve Widrow-Hoff Kuralı olarak da adlandırılmaktadır (Saygılı 2008).

En yaygın kullanılan öğrenme algoritmalarından birisi, geri yayılım (back propagation) algoritmasıdır. Denetimli öğrenme yöntemlerinden birisi olan geri yayılım algoritması; giriş hedefi örneklerinin bilinen bir veri kümesine uygulanır. Tahmin ve sınıflandırma

uygulamalarında kullanılmıştır. Geri yayılım algoritması eldeki veri ile ağın çıktısı arasındaki farka dayalı olarak ağırlıkların güncellenmesini gerçekleştirir. Katmanlar, giriş katmanından gizli katmanlara ve oradan da çıkış katmanına sırasıyla bağlanır. Burada katmanlar arasındaki bağlantıları ağırlıkları içerir ve her bir katman bir ya da daha fazla hücre içermektedir (Altan 2008, Hameed 2017). Geri yayılım öğrenme algoritmasının adımları, Çizelge 2.3’de verilmiştir.

**Çizelge 2.3** Geri yayılım algoritması (Genelleştirilmiş delta öğrenme kuralı) (Fullér 1995)

<i>Geriyayılım Algoritması</i>	
Eğitim veri seti	$\{(x^1, y^1), \dots, (x^K, y^K)\}$
Burada $x^k = (x_1^k, \dots, x_n^k)$ ve $y^k \in \mathbb{R}, k=1, \dots, K$ .	
<b>Adım 1.</b> $\eta > 0, E_{\max} > 0$ seçilir,	
<b>Adım 2.</b> Ağırlıklar $w$ , küçük rastgele değerlerde $k = 1$ ile başlatılır ve çalışma hatası $E$ , 0 olarak ayarlanır,	
<b>Adım 3.</b> Eğitim burada başlar. $X^k$ girişi sunulur, $x = x^k, y = y^k$ ve $O$ çıkışı hesaplanır;	
	$O = \frac{1}{1 + \exp(-W^T o)}$
Burada $o_l$ , gizli katmanın çıkış vektörüdür;	
	$o_l = \frac{1}{1 + \exp(-w_l^T x)}$
<b>Adım 4.</b> Çıktı biriminin ağırlıklarının güncellenir;	
	$W = W + \eta \delta o$
Burada $\delta = (y - O)O(1 - O)$ .	
<b>Adım 5.</b> Gizli birimlerin ağırlıkları güncellenir;	
	$w_l = w_l + \eta \delta W_l o_l (1 - o_l) x, l = 1, \dots, L$
<b>Adım 6.</b> Kümülatif aşamanın hatası, mevcut hatayı $E$ 'ye ekleyerek hesaplanır;	
	$E = E + \frac{1}{2} (y - O)^2$
<b>Adım 7.</b> Eğer $k < K, k = k + 1$ ise ve 2. adıma geri dönerek eğitime devam edilir, aksi takdirde 8. adıma gidilir.	
<b>Adım 8.</b> Eğitim döngüsü tamamlanır. $E < E_{\max}$ için eğitim sonlanır. $E > E_{\max}$ ise, $E = 0, k = 1$ ve Adım 3'e geri dönerek yeni bir eğitim döngüsüne başlanır.	



#### d) Kohonen Öğrenme Kuralı

Kohonen tarafından 1982 yılında geliştirilmiş bu öğrenme kuralında, işlemci elemanlar, öğrenme yapabilmek için birbirleriyle yarışmaktadırlar. Bu yarış sonucunda kazanan işlemci elemanı bağlantı ağırlıklarını değiştirmektedir. En uygun çıktıyı veren işlemci elemanının kendisine komşu işlemci elemanlarının ağırlıklarının değiştirilmesi de olanak sağlamaktadır (Saygılı 2008).

Bu kurala göre ağırlık elemanları ağırlıklarını değiştirmek için birbirleri ile yarışır. En büyük çıktıyı üreten hücre, kazanan hücre olmakta ve bağlantı ağırlıkları değiştirilmektedir. Bu, o hücrenin yakınındaki hücrelere karşı daha kuvvetli hale gelmesi demektir (Efendigil 2008).

#### e) Perceptron Öğrenme Kuralı

Yukarıdaki Perceptron ağırlığının amacı her bir girdi modelini ait veya ait olmayan bir sınıf olarak sınıflandırmaktır. Ait olma, çıkış birimi tarafından +1 değeri verilerek belirtilir; ait olmayan bir cevap ise -1 ile gösterilir. Ağ, iteratif teknikler ile daha önce açıklanan ve Çizelge 2.4’de verilen algoritma ile bu sınıflandırmayı gerçekleştirmek için eğitilmiş olur (Fautsett 1994).

**Çizelge 2.4** Perceptron Algoritması (Fautsett 1994)

---

##### *Perceptron Algoritması*

---

**Adım 1.** Ağırlıklar ve sapma ayarlanır. Öğrenme oranı  $\alpha$  ayarlanır ( $0 < \alpha < 1$ ).

---

**Adım 2.** Durdurma koşulu yanlış olduğunda 3-7 adımları tekrar yapılır.

---

**Adım 3.** Her eğitim çifti için s:t, 5-6 adımları yapılır.

---

**Adım 4.** Giriş birimlerinin aktivasyonları ayarlanır:

$$x_i = s_i.$$

**Adım 5.** Çıkış ünitesinin hesaplama yanıtı:

$$y_{in} = b + \sum_i x_i w_i ;$$
$$y = \begin{cases} 1, & \text{eğer } y_{in} > \theta \\ 0, & \text{eğer } -\theta \leq y_{in} \leq \theta \text{ hesaplanır.} \\ -1, & \text{eğer } y_{in} < -\theta \end{cases}$$

---

## Çizelge 2.4 (Devam) Perceptron Algoritması (Fautsett 1994)

**Adım 6.** Bu örnek için bir hata oluşursa ağırlıklar ve sapma aşağıdaki formüllere göre güncellenir.

Eğer  $y = t$ ,

$$w_i(\text{yeni}) = w_i(\text{eski}) + \alpha t x_i,$$

$$b(\text{yeni}) = b(\text{eski}) + \alpha t.$$

Eğer  $y \neq t$  ise,

$$w_i(\text{yeni}) = w_i(\text{eski}),$$

$$b(\text{yeni}) = b(\text{eski}).$$

**Adım 7.** Test durdurma koşulu: adım 2'de ağırlık değişmemişse algoritma durur; fakat ağırlık değişmişse doğru sınıflandırma yapılmıncaya kadar algoritma tekrarlanır.

(Burada  $\theta$  eşik değeridir.)

### f) Levenberg-Marquardt Öğrenme Kuralı

Dik iniş ve Newton (Newton metodu ile ilgili detaylı bilgiye ulaşmak için Hernández'in 1992'de yazdığı "Newton-Raphson's Method and Convexity" makalesine bakılabilir.) algoritmalarından türetilen LM algoritması güncellemesi Eşitlik 2.26'da verilmiştir (Çavuşoğlu vd. 2012).

$$\Delta w = (J^T J + \mu I)^{-1} J^T e \quad (2.26)$$

$w$  ağırlık vektörü,  $I$  birim matrisi,  $\mu$  kombinasyon katsayısıdır.  $J$   $(P \times M) \times N$  boyutunda Jacobian matrisini,  $e$   $(P \times M) \times 1$  boyutunda hata vektörünü göstermektedir.  $P$  eğitim örnek sayısını,  $M$  çıkış sayısını ve  $N$  ağırlık sayısını göstermektedir.

Levenberg-Marquardt algoritması parametre güncelleme işlemlerini, tüm giriş örnek değerleri için oluşturduğu hata vektörünü ve Jacobian matrisini kullanarak yapmaktadır. Jacobian matrisinin elde edilmesi Eşitlik 2.27'de gösterilmektedir (İnt.Kay.4):

$$\begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \dots & \frac{\partial e_{11}}{\partial w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{Pn}}{\partial w_1} & \dots & \frac{\partial e_{Pn}}{\partial w_N} \end{bmatrix} \quad (2.27)$$

Ayrıca Jacobian matrisinin hata vektörü Eşitlik 2.28 ile elde edilmektedir.

$$e = \begin{bmatrix} e_{11} \\ \dots \\ e_{1n} \\ \dots \\ e_{p1} \\ \dots \\ e_{pn} \end{bmatrix} \quad (2.28)$$

Güncelleme denkleminde  $\mu$  ayarlanabilir bir parametredir. Eğer bu parametre çok büyükse yöntem dik iniş metodu gibi çok küçükse Newton metodu gibi davranmaktadır. Bu parametre için uyarlamalı bir yapı Eşitlik 2.29'da verilmiştir (Çavuşoğlu vd. 2012):

$$\mu(t) = f(x) = \begin{cases} \mu(t)k, & E(t) > E(t - 1) \\ \mu(t)/k, & E(t) \leq E(t - 1) \end{cases} \quad (2.29)$$

### 2.1.7 Veri İşlenmesi

Ağ girişleri ve hedefleri üzerinde ön işlem adımları uygulayarak sinir ağı eğitimi daha verimli hale getirilebilir. Verinin işlenmesi için kullanılan fonksiyonlar, ağ kullanımı için girdileri daha iyi bir forma dönüştürür. Ham girdiler için normalleştirme sürecinin, verilerin eğitime uygun olması için hazırlanmasında büyük etkisi vardır. Bu normalleşme olmadan, sinir ağlarını eğitmek çok yavaş olmaktadır. Birçok veri normalleştirme türü vardır. Bir özellik için sinir ağı içindeki yanlılığı en aza indirmek amacıyla her giriş özelliği için verileri aynı değer aralığında ölçeklendirmek için kullanılabilir. Veri normalleştirilmesi aynı ölçekte her bir özellik için öğrenme sürecini başlatarak öğrenme süresini hızlandırabilir. Girdilerin genellikle farklı ölçeklerde olduğu modelleme uygulaması için özellikle yararlıdır. Farklı teknikler, maksimum kural, minimum kural, toplam kural, ürün kuralı ve benzeri gibi farklı kuralları kullanabilir. Tekniklerden bazıları aşağıda verilmiştir (Yüksek vd. 2007, Doğan vd. 2007, Jayalakshmi and Santhakumaran 2011, Yavuz ve Deveci 2012):

- 1) *İstatistiksel veya Z-Score Normalizasyonu*; Bu teknik, her bir giriş özelliği vektörünü normalleştirmek için bir dizi eğitim verisi boyunca her özellik için ortalama ve standart sapmayı kullanır. Her özellik için ortalama ve standart sapma hesaplanır. Aşağıdaki formülle veriler normalleştirilerek veriler arasındaki

uzaklıklar ortadan kaldırılır, eğitim bittikten sonra ise normalleştirilen veriler tekrar ilk haline dönüştürülür. Dönüşüm Eşitlik 2.30'da verilmiştir:

$$x' = \frac{x_i - \mu_i}{\sigma_i} \quad (2.30)$$

*Burada;*

$x'$  = Normalizasyonu gerçekleşmiş veri,

$x_i$  = Girdi değeri,

$\mu_i$  = Girdi setinin ortalaması,

$\sigma_i$  = Girdi setinin standart sapması,

*ifade etmektedirler.*

Bu, her bir özelliğin sıfır ortalama ve birim varyansa sahip olduğu yerde veri üretir. Normalleştirme tekniği, ilk önce veri setindeki tüm özellik vektörlerine uygulanır; yeni bir eğitim seti yarattıktan sonra eğitim başlar. Bir dizi eğitim verisi üzerinde her özellik için ortalamalar ve standart sapmalar hesaplandıktan sonra, son sistem tasarımında ağırlık olarak kullanılmaları gerekir. Aksi takdirde, sinir ağının performansı önemli ölçüde değişecektir, çünkü normalize edilmemiş verilerden farklı bir veri gösterimi üzerine eğitilmiştir. Bu istatistik kuralının avantajı, aykırı değerlerin veri üzerindeki etkilerini azaltmasıdır.

- 2) **Min-Max Normalizasyonu;** Bu yöntem, özellikleri veya çıktıları bir değer aralığından yeni bir değer aralığına ölçeklendirir. Daha sık olarak, özellikler 0 ile 1 ya da -1 ile 1 arasında olacak şekilde yeniden ölçeklendirilir. Yeniden ölçeklendirme, genellikle Eşitlik 2.31'de ki gibi doğrusal bir yorumlama formülü kullanılarak yapılır:

$$x' = (x_{max} - x_{min}) \frac{(x_i - x_{min})}{(x_{max} - x_{min})} + x_{min} \quad (2.31)$$

*Burada;*

$x'$  = Normalizasyonu gerçekleşmiş veri,

$x_i$  = Girdi verisinin  $i$ . değeri,

$x_{min}$  = Girdi verisi içerisindeki en küçük sayı,  
 $x_{max}$  = Girdi verisi içerisindeki en büyük sayı,

ifade etmektedirler.

Burada  $(x_{max} - x_{min}) = 0$ . Bir özellik için  $(x_{max} - x_{min}) = 0$  olduğunda, verilerdeki o özellik için sabit bir değer gösterir. Verilerde sabit bir değere sahip bir özellik değeri bulunduğu, sınır ağına herhangi bir bilgi sağlamadığından kaldırılmalıdır. Min-max normalizasyon uygulandığında, her özellik yeni değerler aralığında olmalı ve aynı değerler kalmalıdır. Min-max normalizasyon, verilerdeki tüm ilişkilerin tam olarak korunması avantajına sahiptir.

- 3) **Medyan Normalizasyonu;** Medyan yöntemi her örneği, örnekteki tüm girdiler için ham girdilerin medyanı ile normalleştirir. İki hibridize edilmiş örnek arasındaki oranın hesaplanması gerektiğinde, kullanılması normal bir işlemdir. Medyan, aşırı sapmaların büyüklüğünden etkilenmez. Dağıtım yaparken daha yararlı olabilir. Eşitlik 2.32 ile hesaplanır:

$$x' = \frac{x_i}{median(a_i)} \quad (2.32)$$

Burada;

$x'$  = Normalizasyonu gerçekleşmiş veri,

$x_i$  = Girdi değeri,

$a_i$  = Girdi verisinin medyanı,

ifade etmektedir.

- 4) **Sigmoid Normalizasyon;** Sigmoid normalizasyon fonksiyonu, örnekleri 0 ve 1 veya -1 ila +1 aralığında ölçeklendirmek için kullanılır. Çeşitli doğrusal olmayan sigmoid fonksiyon tipleri mevcuttur. Bunun dışında, tan sigmoid fonksiyonu normalizasyon sürecini hızlandırmak için iyi bir seçimdir. Genelde görüntülü verilerde bu normalizasyon yöntemi kullanılmaktadır. Eşitlik 2.33 ile hesaplanabilir:

$$x' = \frac{e^{xi} - e^{-xi}}{e^{xi} + e^{-xi}} \quad (2.33)$$

Burada;

$x'$  = Normalizasyonu gerçekleşmiş veri,

$x_i$  = Girdi değeri,

$e$  = Doğal logaritma değerini,

ifade etmektedir.

- 5) **Ortalama ve Standart Sapma Normalizasyonu;** Ağ girişlerini ve hedeflerini ölçeklendirmek için bir başka yaklaşım, eğitim setinin ortalamasını ve standart sapmasını normalleştirmektir. Bu fonksiyon girişleri ve hedefleri normalleştirir, böylece sıfır ortalama ve birlik standart sapmasına sahip olurlar. Normalleştirilmiş girdiler ve hedefler iade edildiğinde, sıfır ortalama ve birlik standart sapması olacaktır. Bu yöntem eğitim seti verilerini önceden işlemek için kullanıldığında, eğitilmiş ağ yeni girdilerle birlikte kullanılır ve yeni girdiler orijinal veri seti kullanılarak eğitim seti için hesaplanan araçlar ve standart sapmalarla ön işlem yapılabilir. Eşitlik 2.34 ile hesaplanabilir:

$$y' = (x_i - \bar{x}) \times \frac{SS_y}{SS_x} + \bar{y} \quad (2.34)$$

- 6) **İstatistiksel Sütun Normalizasyonu;** İstatistiksel sütun normalleştirme yöntemi, her örneği bir sütun normalizasyon değeri ile normalleştirir. Sütunları bir uzunluğa normalleştirerek her bir sütunun normalizasyonu hesaplanır. Normalize sütun özelliğini bölerek ve küçük bir önyargı değeriyle çarparak her örnek için hesaplanır. Eşitlik 2.35 ile hesaplanabilir:

$$x' = \frac{xi - n(c_a)}{n(c_a)} \times 0,1 \quad (2.35)$$

- 7) **D\_Min\_Max Normalizasyonu;** Verilerin 0,1-0,9 arasında normalize edilmesi istenilirse bu model kullanılmaktadır. Bu sayede normalize edilen veriler boyutsuz bir hale getirilmiş olur. Eşitlik 2.36 ile hesaplanabilir:

$$x' = 0,8 x \frac{x_i - x_{min}}{x_{max} - x_{min}} + 0,1 \quad (2.36)$$

*Burada;*

$x'$  = Normalizasyonu gerekleřmiř veri,

$x_i$  = Girdi deęeri,

$x_{min}$  = Girdi verisindeki en kk sayı,

$x_{max}$  = Girdi verisindeki en byk sayı,

*ifade etmektedir.*

## 2.2 Tip I (Basit Doğrusal Regresyon) Ve Tip II Regresyon Teknikleri

Regresyon teknikleri, sadece bağımlı değişkenin ölçümünde meydana gelebilecek hatanın dikkate alınması ya da ele alınan tüm değişkenlerin ölçümünde meydana gelebilecek hataların dikkate alınması bakımından incelendiğinde, Tip I ve Tip II regresyon teknikleri olarak ikiye ayrılmaktadır (Saraçlı 2008). Bu bölümde Tip I ve Tip II Regresyon teknikleri hakkında bilgiler verilerek, Tip II Regresyon tekniğinin nasıl ve ne amaçla ortaya çıktığından bahsedilmiştir.

### 2.2.1 Tip I Regresyon Analizi

Regresyon analizi, birçok alanda verin analizi için başvurulan önemli bir istatistiksel teknik olmakla beraber, değişkenler arasındaki ilişkiyi açıklamak için kullanılan en yaygın yöntemlerden biridir. Ural ve Kılıç (2018)'da regresyonu basitçe şu şekilde açıklamıştır: “Bağımlı bir değişken ile bağımlı değişken üzerinde etkisi olduğu varsayılan bağımsız değişkenler arasındaki ilişkinin matematiksel bir model ile açıklanmasıdır.”

En basit halde doğrusal regresyon modeli Eşitlik 2.37'deki gibidir (Kalaycı 2016):

$$y = \beta_0 + \beta_1 x + e \quad (2.37)$$

*Burada;*

$\beta_0$  : Doğrunun  $y$ -eksenini kestiği nokta (sabit katsayı),

$\beta_1$  : Doğrunun eğimi (eğim katsayısı),

$e$  : Şansa bağlı hata terimidir.

Burada  $\beta_0$  ve  $\beta_1$  değerleri hesaplanan anakütle parametreleridir. Ancak yine de dikkate alınmayan bağımsız değişkenler olabileceğinden, verilerin tesadüfi değişimlerini gösteren hata terimi  $e$  modele eklenmiştir. Pratikte  $\beta_0$  ve  $\beta_1$  değerleri bilinmiyorsa, anakütleden bir örnek alınarak anakütle parametreleri hakkında istenilen bilgiler üretilir. Bu noktada tahmini değerler olarak  $b_0$  ve  $b_1$  Eşitlik 2.38'de ki gibi kullanılır (Kalaycı 2016).



$$\hat{y} = b_0 + b_1x \quad (2.38)$$

Burada;  $\hat{y}$  :  $y$ 'nin tahmin değeridir.

**Regresyon Parametrelerinin En Küçük Kareler Tahmin Edicileri;**  $x_i, i = 1, \dots, n$  girdi değişkenlerine karşılık gelen  $Y_i$  yanıtlarının gözlemlendiği ve basit doğrusal regresyon modelinde  $\alpha$  ve  $\beta$ 'yi tahmin etmek için kullanıldığını varsayalım.  $\alpha$  ve  $\beta$  tahmin edicilerini belirlemek için izleyen sonuçlara varılır: Eğer  $\alpha$ 'nın tahmin edicisi  $A$ ,  $\beta$ 'nin tahmin edicisi  $B$  olursa, o zaman  $x_i$  girdi değişkenine karşılık gelen yanıtın tahmin edicisi  $A+Bx_i$  olur. Gerçek yanıt  $Y_i$ , kareli fark  $(Y_i - A - Bx_i)^2$  ve böylece  $A$  ve  $B$ ,  $\alpha$  ve  $\beta$ 'nin tahmin edicileri olduğundan, gerçek yanıt değerleri ve tahmin edilen yanıtlar arasındaki kareli farkların toplamı –bu değere SS denilsin– Eşitlik 2.39'da ki gibi hesaplanır (Ross 2009, Çelebioğlu ve Kasap 2012):

$$SS = \sum_{i=1}^n (Y_i - A - Bx_i)^2 \quad (2.39)$$

ile verilir. En küçük kareler yöntemi,  $SS$ 'yi minimize eden  $A$  ve  $B$  değerlerini  $\alpha$  ve  $\beta$ 'nin tahmin edicileri olarak seçer. Bu tahmin edicileri belirlemek için  $SS$ 'nin Eşitlik 2.40'dan  $A$  ve Eşitlik 2.41'den  $B$ 'ye göre türevi alınır:

$$\frac{\partial SS}{\partial A} = -2 \sum_{i=1}^n (Y_i - A - Bx_i) \quad (2.40)$$

$$\frac{\partial SS}{\partial B} = -2 \sum_{i=1}^n (Y_i - A - Bx_i)x_i \quad (2.41)$$

Bu kısmi türevler 0'a eşitlenerek minimize eden  $A$  ve  $B$  değerleri için Eşitlik 2.42 ve Eşitlik 2.43 elde edilir:

$$\sum_{i=1}^n Y_i = nA + B \sum_{i=1}^n x_i \quad (2.42)$$

$$\sum_{i=1}^n x_i Y_i = A \sum_{i=1}^n x_i + B \sum_{i=1}^n x_i^2 \quad (2.43)$$

Bu denklemler normal denklemler olarak bilinir. Eğer:

$$\bar{Y} = \sum_i \frac{Y_i}{n}, \quad \bar{x} = \sum_i \frac{x_i}{n} \quad (2.44)$$

olarak alınırsa:

$$A = \bar{Y} - B \bar{x} \quad (2.45)$$

olarak ilk normal denklem yazılabilir. İkinci normal denklemde  $A$  değerini yerine koyularak:

$$\sum_{i=1}^n x_i Y_i = (\bar{Y} - B \bar{x}) n \bar{x} + B \sum_{i=1}^n x_i^2 \quad (2.46)$$

veya

$$B \left( \sum_i x_i^2 - n \bar{x}^2 \right) = \sum_i x_i Y_i - n \bar{x} \bar{Y} \quad (2.47)$$

ya da

$$B = \frac{\sum_i x_i Y_i - n \bar{x} \bar{Y}}{\sum_i x_i^2 - n \bar{x}^2} \quad (2.48)$$

elde edilir ve böylece aşağıdaki denklemler aracılığıyla parametrelerin tahmin işlemleri Eşitlik 2.49 ile gerçekleştirilir:

$$B = \frac{(\sum_{i=1}^n x_i Y_i - \bar{x} \sum_{i=1}^n Y_i)}{\sum_{i=1}^n x_i^2 - n \bar{x}^2} \quad (2.49)$$

$$A = \bar{Y} - B \bar{x} \quad (2.50)$$

aynı zaman da  $A+Bx$  düz doğrusuna, tahmin edilmiş regresyon doğrusu denir.

## 2.2.2 Tip II Regresyon Analizi

Bilindiği gibi Tip I regresyon ile hesaplanabilecek bir problem veya problemler söz konusu olduğunda, X bağımsız değişkeninin ölçümünün hatasız olduğu varsayımı altında çözüme ulaşma yoluna gidilir. Bu tür problemlerde hatanın Y değişkenine ait olduğu varsayılır ve minimize edilmek istenilen hata, gözlem değerleri olan Y ile tahmin edilmek istenilen regresyon doğrusu üzerinde yer alan ve tahmin edilen Y değerleri arasındaki dik uzaklığın karesidir (Saraçlı 2008).

Saraçlı (2008) yapmış olduğu çalışmada ise Tip II regresyonu şu şekilde açıklamıştır: “Tip II regresyon problemlerinde, hem X hem de Y değişkenlerine ait gözlem değerlerinin hatalı olarak ölçüldüğü varsayılır. Tip II regresyon tekniklerinde genel olarak minimize edilmek istenilen hata, gerçek değerlerden  $\varepsilon$  ve  $\delta$  büyüklüklerinde, çeşitli ölçüm hatası sebepleri ile yanlış ölçülmüş olan  $X_i$  ve  $Y_i$  gözlem değerlerinin, tahmin edilmek istenilen regresyon doğrusu üzerinde yer alan tahmini değerlerine olan dik ya da belirli bir açı ile olan uzaklıklarının karesidir.” Sözü geçen ölçüm hataları konusunda, Fuller’in 1987 yılında yazdığı “Ölçüm Hatalı Modeller” isimli kitabından detaylı bilgilere ulaşılabilir.

Tip II Regresyon Teknikleri hakkında bilgiler aşağıda verilmiştir.

### 2.2.2.1 EKK-Açıortay Tekniği

Daha önceki bölümde anlatıldığı gibi EKK tekniğiyle ilgili; gözlem noktalarının doğruya olan uzaklıklarının üzerindeki hataların tespit edildiği belirtilmiştir. EKK-Açıortay tekniğinde ise EKK doğrusunun açıortayı alınarak işlemler gerçekleştirilmektedir (Baydili 2015).

Tip II regresyon tekniklerinden biri olan En Küçük Kareler (EKK) Açığortay tekniği, gözlem noktalarının tahmin edilen regresyon doğrusuna olan uzaklığını, EKK(Y|X) doğrusu ile EKK(X|Y) regresyon doğrusunun açığortayını dikkate alarak minimize etmeye çalışır. EKK-Açığortay doğrusu konusunda literatüre bakıldığında, bu doğrunun eksikliği hakkında herhangi bir çalışmaya rastlanmamıştır (Isobe *et al.* 1990). Özetlenecek olursa, Saylor *et al.* (2006)'da yaptığı çalışmada, “EKK-Açığortay regresyon tekniği sayesinde, EKK(Y|X) ve EKK(X|Y) doğrularını matematiksel olarak ikiye bölen çizgiyi tanımlar.” diyerek bu teknik hakkındaki tanımını belirtmiştir.

Bu teknik yardımıyla hesaplanacak eğim katsayısı ( $\hat{\beta}_{AO}$ ), sabit katsayı ( $\hat{\beta}_0$ ), varyans hesabı ve varyans hesabı sırasındaki kovaryans terimlerinin hesaplanması Eşitlik 2.51-2.54'de ki eşitliklerle hesaplanır:

$$\hat{\beta}_{AO} = (\hat{\beta}_1 + \hat{\beta}_2)^{-1} \left[ \hat{\beta}_1 \hat{\beta}_2 - 1 + \sqrt{(1 + \hat{\beta}_1^2)(1 + \hat{\beta}_2^2)} \right] \quad (2.51)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_{AO} * \bar{x} \quad (2.52)$$

$$\begin{aligned} Var(\hat{\beta}_{AO}) = & \frac{\hat{\beta}_3^2}{(\hat{\beta}_1 + \hat{\beta}_2)^2 (1 + \hat{\beta}_1^2)(1 + \hat{\beta}_2^2)} \left[ (1 + \hat{\beta}_1^2)^2 Var(\hat{\beta}_1) \right. \\ & + 2(1 + \hat{\beta}_1^2)(1 + \hat{\beta}_2^2) Cov(\hat{\beta}_1, \hat{\beta}_2) \\ & \left. + (1 + \hat{\beta}_2^2) Var(\hat{\beta}_2) \right] \end{aligned} \quad (2.53)$$

$$\begin{aligned} Cov(\hat{\beta}_1, \hat{\beta}_2) = & (\hat{\beta}_1 S_{xx}^2)^{-1} \left\{ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) [y_i - \bar{y} - \hat{\beta}_1(x_i \right. \\ & \left. - \bar{x})] y_i - \bar{y} - \hat{\beta}_2(x_i - \bar{x}) \right\} \end{aligned} \quad (2.54)$$

### 2.2.2.2 Majör Eksen (MA) Ortogonal Regresyon Tekniği

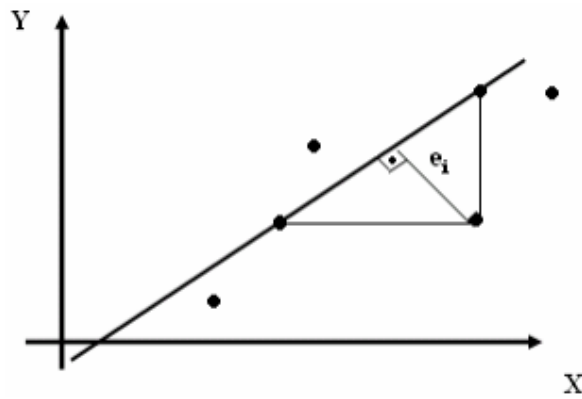
Ortogonal Regresyon (OR) yöntemi 1879'de ilk olarak Adcock (1878) tarafından bulunmuştur ve 1901 yılında da Pearson tarafından önerilmiştir (Isobe *et al.* 1990, Pearson 1901, Saraçlı 2008, Keleş ve Altun 2016). Ayrıca literatür incelendiğinde, OR yönteminin çoğu zaman birbirlerinden bağımsız bir şekilde birçok kez yeniden keşfedildiği görülmüştür (Ding *et al.* 2013, Keleş ve Altun 2016).

Majör Eksen Regresyonu (MAR), hem X hem de Y'de doğal deęişkenlik problemini ele almak için yaygın bir yöntemdir. MAR teknięi, sıradan EKK regresyonundan farklı olarak bir kayıp fonksiyonunu en aza indirgeyen bir yöntemdir. EKK, öngörülen deęerler hakkındaki gözlemlenen karelerin sapma toplamını ((gözlemlenen-tahmin edilen)<sup>2</sup>) azaltır. MAR teknięinin amacı ise, her nokta ve regresyon doğrusu arasındaki dikey mesafedeki karelerin toplamını en aza indirmektir (İnt.Kay.7). Aynı zamanda EKK teknięinin aksine, MAR teknięi deęişkenin hangisinin baęımlı, hangisinin baęımsız olduęuyla ilgilenmemekle birlikte, EKK(X|Y) doğrusu ve EKK(Y|X) doğrusu arasında yer almaktadır (Amman and Ness 1988, Saraçlı and Çelik 2011).

Wu ve Yu (2018) yaptıęı bir çalışmada, “OR tüm veri noktalarından regresyon çizgisine kadar olan kareli dik mesafelerin toplamını en aza indirir ve eşit hata varyanslarını göz önünde bulundurur.” diyerek OR konusundaki görüşlerini sunmuştur. Bu teknięin Hata Kareler Toplamı (HKT) Eşitlik 2.55 aracılığıyla hesaplanmaktadır:

$$HKT = \sum_{i=1}^N [(x_i - X_i)^2 + (y_i - Y_i)^2] \quad (2.55)$$

OR regresyon teknięinde minimize edilmek istenilen hata miktarının grafięi Şekil 2.20’de verilmiştir.



**Şekil 2.20** Ortogonal Regresyon teknięinde minimize edilmek istenilen hata miktarının grafięi (Saraçlı 2008)

Bu teknik yardımıyla hesaplanması istenilen regresyon doğrusu için eğim katsayısı ve varyans hesabı Eşitlik 2.56 ve Eşitlik 2.57 ile hesaplanır:

$$\hat{\beta}_{MA} = \frac{1}{2} \left[ (\hat{\beta}_2 - \hat{\beta}_1^{-1}) + \text{Sign}(S_{xy}) \sqrt{4 + (\hat{\beta}_2 - \hat{\beta}_1^{-1})^2} \right] \quad (2.56)$$

$$\text{Var}(\hat{\beta}_{MA}) = \frac{\hat{\beta}_{MA}^2}{4\hat{\beta}_1^2 + (\hat{\beta}_1\hat{\beta}_2 - 1)^2} [\hat{\beta}_1^{-2}\text{Var}(\hat{\beta}_1) + 2\text{Cov}(\hat{\beta}_1, \hat{\beta}_2) + \hat{\beta}_1^2\text{Var}(\hat{\beta}_2)] \quad (2.57)$$

Fuller (1987) bu tekniğin eğim katsayısı formülünü daha anlaşılır bir şekilde düzenleyerek Eşitlik 2.58’de ki formülü yazmıştır (Carroll and Ruppert 1996):

$$\hat{\beta}_{MA} = \frac{\sigma_y^2 - \sigma_x^2 + \sqrt{(\sigma_y^2 - \sigma_x^2)^2 + 4\sigma_{xy}^2}}{2\sigma_{xy}} \quad (2.58)$$

### 2.2.2.3 İndirgenmiş Majör (RMA-Reduced Major Axis) Eksen Regresyon Tekniği

İndirgenmiş majör eksen, Ortogonal Regresyonun ölçek bağımlılığını hafifletmek amacıyla, 1940 yılında Strömberg, 1950 yılında da Kermack ve Haldane tarafından birbirlerinden bağımsız bir şekilde öne sürülmüştür (Isobe *et al.* 1990, Saraçlı 2008).

Literatürde istatistikçiler tarafından “Uyarlanmış Majör Eksen” olarak da adlandırılmış bu teknik, EKK Açığortay tekniğinin eğim katsayılarını hesaplamakta kullanılan, EKK(Y|X) ve EKK(X|Y) teknikleri aracılığıyla hesaplanan eğim katsayılarının geometrik ortalamasının bulunması temeline dayanmaktadır (Isobe *et al.* 1990, Saraçlı 2008).

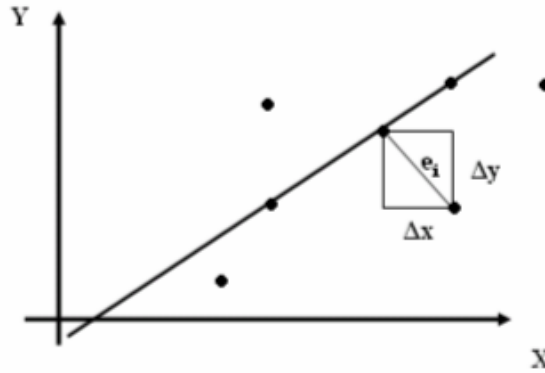
RMA Regresyon tekniği, her ne kadar Ortogonal Regresyon tekniğinin ölçek bağımlılığını hafifletmek amacıyla geliştirilmiş olsa da, Isobe vd. (1990) yaptığı çalışmada, bu tekniğinde bazı istenmeyen özelliklere sahip olduğunu açıkça belirtmiştir.

RMA, alanların toplamını minimuma indirir (böylece, sonuç kısmındaki veri noktalarının hem dikey hem de yatay mesafelerini kullanır), en küçük kareler toplamı dikey mesafelerdir. Standart en küçük kareler regresyonu ile ilgili konulardan biri, en küçük kareler regresyon denkleminin ( $y=a+bx$ ) değerinin normal bir denklem olarak ele alınamaması ve y'nin y'yi öngören bir denklem elde etmek için geri çözülememesidir. En küçük kareler tekniğinde, x ve y değişkenleri değiştirdiğinde, elde edilen regresyon denklemi ( $x=(y-a)/b$ )'ye eşdeğer değildir. Ek olarak, bunu en küçük kareler tekniğiyle yapmak, değişkenler birbirleriyle değiştirildiklerinde öngörülenin altında ve üstünde olan paradokslarla sonuçlanır. Bir RMA denklemiyle, bu basit cebirsel ifade verimlilikle gerçekleştirilebilir, çünkü RMA'nın bir araya getirilen değişkenlerle elde edeceği denklem ile eşleşecektir. Sonuç olarak, elde edilen RMA regresyonu ( $x=(y-a)/b$ )'ye eşdeğeri olacaktır (Harper 2014).

RMA regresyon tekniğinde minimize edilmek istenilen hata (HKT) Eşitlik 2.59'da ki gibi hesaplanır:

$$HKT = \sum \{(x_i - X_i) (y_i - Y_i)\} \quad (2.59)$$

İndirgenmiş majör eksen Ortogonal regresyon tekniğinde minimize edilmek istenilen hatanın grafiği Şekil 2.21'de verilmiştir.



**Şekil 2.21** İndirgenmiş Majör Ekseni Ortogonal regresyon tekniğinde minimize edilmek istenilen hata miktarı grafiği (Saraçlı 2008)

RMA eğim katsayısı ve bu katsayıya ilişkin varyansın hesaplaması Eşitlik 2.60 ve Eşitlik 2.61’de ki formüller yardımıyla bulunur (Li 2012):

$$\begin{aligned}\hat{\beta}_{RMA} &= \text{sign}(S_{XY}) \sqrt{\hat{\beta}_{EKK(Y|X)} \hat{\beta}_{EKK(X|Y)}} & (2.60) \\ &= \text{sign}(S_{XY}) \sqrt{(S_{XY}/S_{XX})(S_{YY}/S_{XY})} \\ &= \text{sign}(S_{XY}) \sqrt{S_{YY}/S_{XX}} = \text{Sign}(S_{XY})(\hat{\beta}_1 \hat{\beta}_2)^{1/2}\end{aligned}$$

$$\text{Var}(\hat{\beta}_{RMA}) = \frac{1}{4} \left[ \frac{\hat{\beta}_1}{\hat{\beta}_2} \text{Var}(\hat{\beta}_1) + 2\text{Cov}(\hat{\beta}_1, \hat{\beta}_2) + \frac{\hat{\beta}_1}{\hat{\beta}_2} \text{Var}(\hat{\beta}_2) \right] \quad (2.61)$$

#### 2.2.2.4 Deming Regresyon Tekniği

Birçok alanda çalışmaları bulunan ve kendisini kaliteyle ilgili olan teorisiyle dünyaya tanıtmış olan Prof. Dr. William Edwards Deming, 1943 yılında yaptığı bir çalışmayla diğer regresyon türlerinin aksine, hem X hem de Y değişkenlerinin hatayla ölçüldüğü iki boyutlu verilere düz bir çizgi sığdırmak için kullanılan bir teknik olduğunu belirtmiştir. Bu, yalnızca Y bağımlı değişkeninin hatayla ölçüldüğü basit doğrusal regresyondan farklıdır. Deming regresyonu, iki ölçüm metodu arasındaki sistematik farklılıkları araştırmak için genellikle klinik kimyadaki metot karşılaştırma çalışmaları için kullanılmaktadır (Saraçlı vd. 2009, Baydili 2015, İnt.Kay.5).

Deming regresyon tekniğinde minimize edilmek istenilen hata (HKT) ve Deming tekniğiyle regresyon doğrusunun kestirilebilmesi için  $\lambda$  değeri Eşitlik 2.62 ve Eşitlik 2.63’deki formüller aracılığıyla hesaplanır (Saraçlı vd. 2009):

$$HKT = \sum \{(x_i - X_i)^2 + \lambda(y_i - Y_i)^2\} \quad (2.62)$$

$$\lambda = \frac{S_{ex}^2}{S_{ey}^2} \quad (2.63)$$

Burada hesaplanan  $S_{ex}^2$  ve  $S_{ey}^2$ , sırasıyla X ve Y gözlem değerlerine ait hataların varyanslarıdır. Ayrıca bu hesaplanan  $\lambda$  değeri, kareli sapmaların toplamalarını doğru üzerinde minimize eder ve aradaki açığı belirlemeyi sağlar (Saraçlı vd. 2009).

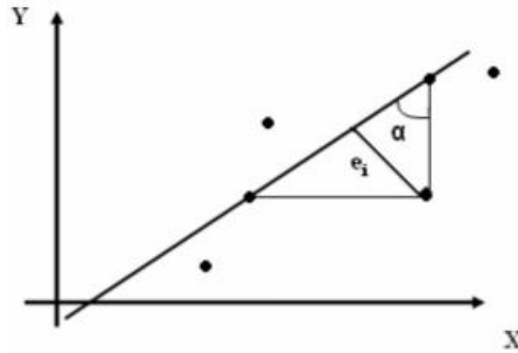


Deming regresyon, ölçüm hata oranının ( $\lambda = V(\varepsilon_i)/V(\delta_i)$ ) sabit olduğunu varsayar. Bu işlem, kullanıcının bilinen bir hata oranını girmesini ya da her konudan çoklu ölçümler yapmasını, böylece hata oranının verilerden hesaplanmasını gerektirir.  $\lambda = 1$  olduğunda ise, Deming regresyon Ortogonal Regresyon ile aynı sonucu verir (İnt.Kay.5).

Hem basit (ağırlıksız) hem de ağırlıklandırılmış Deming regresyon yöntemleri de mevcuttur. Regresyon katsayıları ve öngörülen değerler, Linnet'in (1990)'da yaptığı bir çalışmadaki verilen formüller kullanılarak hesaplanır. Regresyon katsayılarının standart hataları ve öngörülen değerler, Jacknife bırakma yöntemi kullanılarak hesaplanır (İnt.Kay.5).

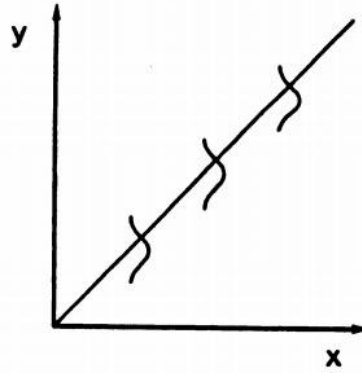
Deming'in formülüne göre eğimi hesaplamak için, bir kişinin Gauss hata ölçümleri alması gerekir. Ayrıca bu ölçümler alınırken x ve y değerleri boyunca, x ve y değerlerinin sabit olması gerekmektedir (Mandel 1964, Wakkers *et al.* 1975, Cornbleet and Gochman 1979). Ayrıca Deming regresyon yöntemi, X ve Y gözlem değerlerinin hatalarının Gaussian (Normal) dağıldığını varsayar (Akkoca 2012).

Şekil 2.22'de bir Deming Regresyon modelinin grafiği sunulmuştur.

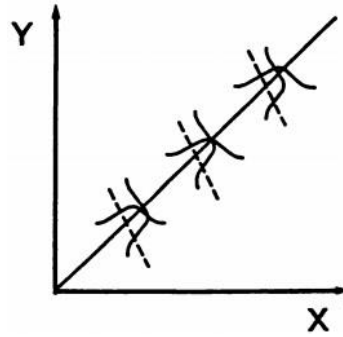


Şekil 2.22 Deming Regresyon tekniğinin bir model grafiği (Saraçlı 2008)

EKK ve Deming yöntemleri için analitik standart sapmaların oranının belirlediği açıyla, kareye olan mesafelerin minimuma indirilmesi ile tahmin edilen regresyon çizgisinin grafikleri Şekil 2.23 ve Şekil 2.24'de verilmiştir (Linnet 1993).



Şekil 2.23 EKK modelinde, X hata içermezken y için sabit hataların dağılımı



Şekil 2.24 Deming regresyon yönteminin grafiği

Birçok bilim adamının üzerinde çalıştığı Deming tekniğinde, tahmin edilmek istenen regresyon denkleminin eğim katsayısı ( $\hat{\beta}_1$ ) ve sabit katsayı ( $\hat{\beta}_0$ ), Eşitlik 2.67 ve Eşitlik 2.68 aracılığıyla hesaplanır. Bu hesaplama da kullanılan q, p ve u ifadeleri ise Eşitlik 2.64-2.66'da belirtildiği gibi hesaplanmaktadır.

$$u = \sum (x_i - \bar{x})^2 \quad (2.64)$$

$$q = \sum (y_i - \bar{y})^2 \quad (2.65)$$

$$p = \sum (x_i - \bar{x})(y_i - \bar{y}) \quad (2.66)$$

$$\hat{\beta}_1 = \frac{(\lambda q - u) + \sqrt{(u - \lambda q)^2 + 4\lambda p^2}}{2\lambda p} \quad (2.67)$$

$$\hat{\beta}_0 = \bar{y} - b\bar{x} \quad (2.68)$$

Daha önce de bahsettiğimiz Deming regresyon katsayılarının standart hataları ve öngörülen değerler, Jacknife bırakma yöntemi kullanılarak hesaplanır. Jacknife yöntemi Eşitlik 2.69-2.72’de ki formüller yardımıyla hesaplanır (Linnet 1990).

$$b_i = kb - (k - 1)b^i \quad (i = 1, \dots, k) \quad (2.69)$$

$$\tilde{b} = \sum_{i=1}^k \frac{b_i}{k} \quad (2.70)$$

$$V_j(b) = \sum_{i=1}^k \frac{(b_i - \tilde{b})^2}{(k - 1)} \quad (2.71)$$

$$SE(b) = \sqrt{V_j(b)/k} \quad (2.72)$$

Parvin (1984)’de yaptığı bir çalışmayla, EKK eğim katsayısı ( $b_{LS}$ ) ile Deming Regresyonun eğim katsayısını doğrusal fonksiyonel eğim katsayısı ( $b_{LF}$ ) olduğunu kabul ederek yeniden formüle etmiştir ve bu sayede doğrudan karşılaştırılabilme imkânı sağlamıştır. Cebirsel ifade ve yeniden düzenleme işleminden sonra, doğrusal fonksiyonel eğim Eşitlik 2.73’de ki gibi yazılabilir:

$$b_{LF} = 1/2 \left[ \frac{b_{LS}}{r^2} - \frac{\lambda}{b_{LS}} + \left[ \left( \frac{b_{LS}}{r^2} - \frac{\lambda}{b_{LS}} \right)^2 \right]^{1/2} \right] \quad (2.73)$$

Burada, belirli bir  $r$  ve  $\lambda$  değeri için  $b_{LF}$  ve  $b_{LS}$  arasındaki kesin ilişkiyi tanımlamaktadır.

Parvin Eşitlik 2.73’de düzenlediği bu denklemin matematiksel değerlendirmesini aşağıdaki gibi yapmıştır:

- Deming regresyonun eğim katsayısının doğrusal fonksiyonel eğim katsayısı her zaman EKK eğim katsayısından büyüktür

- $\lambda$  değeri büyüdükçe, Deming doğrusal fonksiyonel eğim katsayısı EKK eğim katsayısına yaklaşır
- $\lambda$  değeri küçüldükçe, Deming doğrusal fonksiyonel eğim katsayısı,  $b_{LS}/r^2$  değerine yaklaşır, tıpkı eksenleri ters çevrilmiş olan en küçük kareler eğim tahmini gibi
- $\lambda$  azaldıkça, deming doğrusal fonksiyonel eğim katsayısı ve EKK eğim katsayısı arasındaki fark artar
- Deming doğrusal fonksiyonel eğim katsayısı ve EKK eğim katsayısı arasındaki fark,  $\lambda = 1$  olduğunda  $\lambda$  değerine oldukça duyarlıdır.

### 2.2.2.5 Passing-Bablok Regresyon Tekniği

Passing ve Bablok (1983)'de klasik doğrusal regresyon varsayımlarının üstesinden gelen iki ölçüm yönteminin karşılaştırılmasını sağlayan bir regresyon metodu geliştirmiştir (İnt.Kay.8). En Küçük Kareler tekniğinin varsayımlarını hatırlamak gerekirse (Alma ve Vupa 2008, Arı ve Önder 2013, İnt.Kay.8):

- Regresyon denkleminde Y'yi açıklayan X bağımsız değişken(ler)i hata içermez,
- Hata terimlerinin dağılımı normaldir,
- Hataların beklenen değerini 0'dır,
- Hataların varyansı homojen ve  $\sigma^2$ 'ye eşittir,
- Ölçüm hatasının varyansı sabittir,
- Hata terimleriyle X bağımsız değişken(ler)i arasında herhangi bir ilişki yoktur,
- Ayrıca model aykırı değerden etkilendiği için, model aykırı değer içermez.

Passing ve Bablok, bu varsayımların üstesinden gelen bir yöntem önerdi: iki değişkenin, dağılımları hakkında varsayımda bulunmaya gerek kalmadan rastgele bir role (ölçüm hatasını ve ortada ölçülen ögenin dağılımını temsil eden) ve aynı dağılıma sahip olduğunu varsaydılar. Bu öneri aşağıdaki Eşitlik 2.74 ve Eşitlik 2.75'de ki gibi tanımlanır (İnt.Kay.8):

$$y_i = a + bx_i + \xi_i \quad (2.74)$$

$$x_i = A + By_i + \eta_i \quad (2.75)$$

Burada  $\xi$  ve  $\eta$ 'in aynı dağılımı izlemektedir. Passing-Bablok yöntemi a ve b katsayılarının (A ve B'yi  $B=1/b$  ve  $A=-a/b$  kullanarak hesapladığı gibi) ve bu değerler etrafındaki güven aralığının hesaplanmasına izin verir. Bu değerlerin incelenmesi, yöntemlerin karşılaştırılmasına yardımcı olur. Çok yakınlarsa, b 1'e ve a 0'a yakındır (İnt.Kay.8).

Passing-Bablok regresyonu aşağıdaki varsayımları gerektirir (Saraçlı 2008, İnt.Kay.6):

- i. X ve Y değişkenleri yüksek ve pozitif korelasyonludur (sadece yöntem karşılaştırma problemleri için).
- ii. X ve Y arasındaki ilişki doğrusaldır.
- iii. X ve Y'nin dağılımları (varyansları dâhil) hakkında özel bir varsayım yapılmaz.
- iv. Bu teknikte klasik regresyondaki gibi hataların dağılımı normal dağılıma uymasının gerekliliği gibi bir şart yoktur.

Bu varsayımlar altında genellikle yöntem karşılaştırması için kullanılan Passing-Bablok regresyonu, hem X hem de Y değişkenlerinin hatayla ölçüldüğü iki boyutlu verilere düz bir çizgi yerleştirmek için sağlam ve parametrik olmayan bir yöntemdir. Genellikle yöntem karşılaştırma araştırmalarında aynı ölçümleri vermesi gereken iki cihaz söz konusu olduğunda ve bunlar karşılaştırmak istendiğinde kullanışlı bir yöntemdir. Passing-Bablok tekniğinde, doğrusal bir regresyon çizgisi tahmin ederek ve kesişmenin sıfır ve eğimin bir olup olmadığını test ederek gerçekleştirilir. Passing-Bablok regresyon tekniği doğrusal denklemin kesişimi  $\beta_0$ , eğimi ise  $\beta_1$ 'dir (İnt.Kay.6).

Eğimin yani  $\beta_1$ 'in tahmini, 0/0 veya -1 eğimle sonuçlanan çiftler hariç, tüm olası veri noktası çiftlerinden oluşturulabilecek tüm eğimlerin medyanı olarak hesaplanır. Bu eğimlerin bağımsız olmayışından kaynaklı tahmin yanlılığını düzeltmek için, medyan -1'den küçük eğim sayısı olan bir faktör (K) ile kaydırılır. Bu yaklaşık olarak tarafsız bir tahmin edici oluşturur. Karşılaşmanın  $\beta_0$  tahmini,  $(Y_i - \beta_1 X_i)$ 'nin medyanıdır. Bunlara ek olarak, kesişme iki yöntem arasındaki sistematik önyargı (fark) olarak yorumlanır. Eğim, iki yöntem arasındaki orantılı önyargı miktarını ölçer (İnt.Kay.6).

Passing-Bablok regresyon tekniğinde, regresyon modelini oluşturmak için hesaplanan eğim katsayısı Eşitlik 2.76 yardımıyla hesaplanır (Saraçlı 2008, Passing and Bablok 1983):

$$S_{ij} = \frac{y_i - y_j}{x_i - x_j} \quad 1 \leq i \leq j \leq n \text{ için} \quad (2.76)$$

$S_{ij}$ 'nin yapısal ilişkisini genel bir ifade olarak yazılacak olursa:

$$S_{ij} = \frac{y_i^* - y_j^* + \eta_i - \eta_j}{x_i^* - x_j^* + \xi_i - \xi_j} \quad (2.77)$$

$y_i^* = a + \beta_i^*$  ve  $d_{ij} = (x_i^* - x_j^*)$ 'den Eşitlik 2.78'i elde ederiz:

$$S_{ij} = \frac{\beta d_{ij} + (\eta_i - \eta_j)}{d_{ij} + (\xi_i - \xi_j)} = \beta \frac{d_{ij} + \frac{\eta_i - \eta_j}{\beta}}{d_{ij} + (\xi_i - \xi_j)} = \beta \frac{(d_{ij} + z_{ij})}{(d_{ij} + z_{ij})} \quad (2.78)$$

Burada  $z_{ij}$  ve  $z'_{ij}$ 'nin bağımsız ve aynı dağıtımdan olduğu yerlerdir. K,  $S_{ij} < -1$  olan  $S_{ij}$  değer sayıdır ve buna göre:

$$\beta_1 = \begin{cases} S_{\left(\frac{N+1}{2}+K\right)}, & N\text{'nin çift olduğu durumda} \\ \frac{1}{2} \left( S_{\left(\frac{N}{2}+K\right)} S_{\left(\frac{N}{2}+1+K\right)} \right), & N\text{'in tek olduğu durumda} \end{cases} \quad (2.79)$$

olur.

Passing-Bablok regresyon tekniğinde, regresyon modelini oluşturmak için hesaplanan sabit katsayı Eşitlik 2.80 yardımıyla hesaplanır (Passing and Bablok 1983):

$$\beta_0 = \text{med}(y_i - \beta_1 x_i) \quad (2.80)$$

### 2.2.2.6 York Regresyon Tekniđi

York vd. (2004) regresyon tekniđini Eşitlik 2.81’de ki şekilde açıklamıştır (Saylor *et al.* 2006):

$$b = \frac{\sum_{i=1}^N W_i(Y_i - \bar{Y})}{\sum_{i=1}^N W_i(X_i - \bar{X})} \quad (2.81)$$

buradan,

$$W_i = \frac{w(X_i)w(Y_i)}{w(X_i) + b^2w(Y_i) - 2br_i\sqrt{w(X_i)w(Y_i)}} \quad (2.82)$$

$$\beta_i = W_i \left[ \frac{X_i - \bar{X}}{w(Y_i)} + \frac{b(Y_i - \bar{Y})}{w(X_i)} - \frac{(b(X_i - \bar{X}) + (Y_i - \bar{Y}))r_i}{\sqrt{w(X_i)w(Y_i)}} \right] \quad (2.83)$$

ve

$$\bar{X} = \frac{\sum_{i=1}^N W_i X_i}{\sum_{i=1}^N W_i} \quad (2.84)$$

$$\bar{Y} = \frac{\sum_{i=1}^N W_i Y_i}{\sum_{i=1}^N W_i} \quad (2.85)$$

$W_i$  ve  $\beta_i$ ,  $b$ ’nin işlevleri olduğundan, Eşitlik 2.81 yinelenerek çözümlenmelidir. Her bir veri noktası için  $w(X_i)$  ve  $w(Y_i)$  ağırlıkları ve hata korelasyonları  $r_i$  göz önüne alındığında,  $b$  için bir ilk tahmin seçilir (Muhtemelen EKK(X|Y) veya Ortogonal teknikten). İterasyon  $b$ ’nin ardışık değerleri önceden tanımlanmış bir tolerans gösterene kadar aşağıdaki adımlar gerçekleştirilir (Saylor *et al.* 2006):

- $b$ ,  $w(X_i)$ ,  $w(Y_i)$  ve  $r_i$  kullanılarak, Eşitlik 2.82 yardımıyla her veri noktası için  $W_i$  hesaplanır.
- Gözlem noktalarını  $(X_i, Y_i)$  ve  $W_i$ ’yi kullanarak, Eşitlik 2.84 ve Eşitlik 2.85 yardımıyla  $\bar{X}$  ve  $\bar{Y}$  hesaplanır.
- Her veri noktası için Eşitlik 2.82’den  $\beta_i$  hesaplanır.
- $b$  için Eşitlik 2.81’den yeni tahmin hesaplanır.

- Kesişme noktası  $a$ , daha sonra  $a = \bar{Y} - b\bar{X}$  'den bulunur.

York regresyon tekniğinin uygulanması oldukça basittir ve deneyimlere göre yakınsaklık için 10'dan fazla iterasyon (genellikle daha az) gerektirmektedir (Saylor *et al.* 2006).

Legendre ve Legendre (2012)' ye göre, yaptıkları simülasyon çalışmalarının sonuçlarını göz önüne aldıklarında, rasgele ve hatalı ölçülen değişkenler arasındaki fonksiyonel doğrusal ilişkilerin parametrelerini tahmin etmek zorunda olan araştırmacılara aşağıdaki önerileri sunmuşlardır (İnt.Kay.3).

- 1) Yanıt değişkeni  $y$  üzerindeki rasgele değişimin büyüklüğü (hata değişkeni: Örneklem varyansının aksine,  $x$  veya  $y$ 'deki hata varyansı verilerden tahmin edilemez. Bir tahmin ancak değişkenlerin ölçülme şeklinden haberdar edilebilir.),  $x$  açıklayıcı değişkenindeki değerden çok daha büyükse (üçten daha fazla), aşağıdaki gibi devam edilir,
- 2) Bir dağılım diyagramına bakarak veya resmi bir anlamlılık testi uygulayarak verilerin yaklaşık olarak iki değişkenli olup olmadığını kontrol edilir. Değilse, onları iki değişkenli normal yapmak için dönüşümler denenir. Doğru bir şekilde iki değişkenli normal olması veya yapılabilmesi için, 3. ve 4. önerileri dikkate alınır. Eğer değilse 5. Öneriye bakılır.
- 3) İki değişkenli normal veriler için, her iki değişken de aynı fiziksel birimlerde (orijinal olarak aynı birimlerde ölçülen dönüştürülmemiş değişkenler) ifade edilirse veya makul ölçüde olabilirse boyutsuz (örneğin, log-dönüştürülmüş değişkenler) ise ana eksen (MA) regresyonunu kullanılır. Değişkenlerin hata değişikliklerinin yaklaşık olarak eşit olduğu varsayılır.

Jolicoeur (1990)'a göre, sonuçların dikkatle yorumlanması şartıyla, hata değişikliklerinin oranı hakkında hiçbir bilgi bulunmadığında ve bunun 1'den farklı olacağına inanmak için bir neden olmadığında, MA kullanılabilir. MA tarafsız eğim tahminleri ve doğru güven aralıkları üretir diyerek konu hakkındaki görüşünü belirtmiştir.



- 4) İki deęişkenli normal veriler için, eęer deęişkenler aynı fiziksel birimlerde ifade edilmedięinden veya iki eksen üzerindeki hata deęişikliklerinden farklı olduęu için MA kullanılamıyorsa, eęer makul bir şekilde olabilirse, fonksiyonel doęrusal iliřkinin parametrelerini tahmin etmek için iki alternatif yöntem kullanılabilir. Her eksenindeki hata varyansının karřılık gelen deęişkenin varyansı ile orantılı olduęunu, yani ( $y$ 'nin hata varyansı/ $y$ 'nin varyansı) ( $x$ 'in hata varyansı/ $x$ 'in varyansı) olduęu varsayılmıřtır (McArdle 1988).
- a) Menzilli ana eksen regresyonu (RMA) kullanılabilir. RMA'dan önce, nesnelerin bir daęılım řemasını kullanarak aykırı deęerlerin varlıęını kontrol etmelidir.
- b) Standart ana eksen regresyonu (SMA) kullanılabilir. İlk önce bir iliřkinin hipotezinin desteklenip desteklenmedięini belirlemek için ilk önce korelasyon katsayısının ( $r$ ) önemini test edilmediir. Bu kořul karřılanmadıęında hiębir SMA regresyon denklemi hesaplanmamalıdır. SMA eęim tahminleri anlamlılık için test edilemedięinden, bu ideal olmayan bir çözüml olarak kalmaya devam etmektedir. Güven aralıkları da dikkatli kullanılmalıdır: simülasyonlar, eęim  $\pm 1$ 'den ayrıldıęından, SMA eęim tahmininin gittikçe önyargılı olduęunu ve güven aralıęının geręek deęeri daha az ve daha sık ięerdięini görölmüřtür. Eęim  $\pm 1$ 'in yakınında olduęunda,  $n$  çok küçükse veya korelasyon zayıfsa güven aralıęı çok dardır.
- 5) Daęılım iki deęişkenli normal deęilse ve veriler bu kořulu saęlamak için dönüřtürölemiyorsa (örneęin daęıtım iki veya daha fazla moda sahipse), bir regresyon çizgisinin eęiminin geręekten arasındaki iki deęişken arasındaki fonksiyonel iliřkiyi tanımlamak için yeterli bir model olup olmadıęına bakılmalıdır. Daęılım iki deęişkenli normal olmadıęından, öncelikle iki deęişkenli normal daęılımın ilk temel bileřenini tanımlayan MA, SMA veya RMA gibi modeller uygulamak için çok az sebep olduęu görölmektedir. Dolayısıyla, iliřki doęrusal ise, regresyon çizgisinin parametrelerini tahmin etmek için OLS önerilir. Eęimin önemi permütasyonla test edilmelidir, çünkü parametrik

testlerin dağıtım varsayımları tatmin edici değildir. Düz bir çizgi uygun bir model değilse, polinom veya doğrusal olmayan regresyon düşünülmelidir.

- 6) Çalışmanın amacı, fonksiyonel bir ilişkinin parametrelerini tahmin etmek değil, sadece verilen  $x$ 'ler için  $y$  değerlerini tahmin etmek olduğunda, her durumda OLS kullanılmadır. OLS,  $y$ 'deki kare artıklarını en aza indiren tek yöntemdir. OLS regresyon çizgisinin kendisi anlamsızdır. Bununla birlikte, standart hata ve güven bantlarını kullanmamalıdır; ancak,  $x$ 'in hatasız olduğu bilinmediği sürece (Sokal and Rohlf 1995); Bu uyarı OLS için hesaplanan % 95 güven aralığı için geçerlidir.
- 7) Gözlemler, modelin kalitesini değerlendirmek için istatistiksel veya deterministik bir modelin (örneğin simülasyon modeli) öngörülleri ile karşılaştırılabilir. Model, hata ile ölçülen rasgele değişkenler içeriyorsa, gözlemler ve model tahminleri aynı birimlerde olması gerektiğinden karşılaştırma için MA kullanılır.

Model verilere iyi uyuyorsa, eğimin 1 olması ve 0'ın kesişmesi beklenir. 1'den büyük ölçüde farklı olan bir eğim, gözlemlenen değerlerle orantılı olarak gözlenen ve simüle edilmiş değerler arasında bir fark olduğunu gösterir. Göreceli ölçek değişkenleri için, 0'dan önemli ölçüde farklılaşan bir kesişme, gözlemler ve simülasyonlar arasında sistematik bir farkın varlığını göstermektedir (Mesplé *et al.* 1996).

- 8) Tüm yöntemlerde,  $n$  küçük olduğunda güven aralıkları büyüktür;  $n$  yaklaşık 60'a yükseldikçe küçülür ve daha yavaş bir şekilde değişir. Tip II regresyonu ideal olarak 60 veya daha fazla gözlem içeren veri setlerine uygulanmalıdır.

### 3. MATERYAL ve METOT

Bu çalışmada, YSA ve Tip II Regresyon teknikleri konularındaki literatür bilgilerinin yer verilmesi ve simülasyon aracılığıyla üretilen veriler üzerinde bu iki tekniğinde uygulanarak hangi tekniğin daha iyi sonuçlar vereceği amaçlanmıştır. Bu amaçla MATLAB programı yardımıyla rasgele olarak üretilen 5000 birimlik gözlemin, serbestlik dereceleri 4, 10 ve 30 olan t dağılımlarında ( $T \sim 4$ ,  $T \sim 10$ ,  $T \sim 30$ ) ve farklı örneklem büyüklüklerinde ( $n=50$ ,  $n=150$ ,  $n=350$ ) analiz edilmiştir. Verilerin 4, 10 ve 30 serbestlik derecelerinin t dağılımından çekilmesinin sebebi, dağılımın basıklığının değişmesiyle hata miktarındaki değişim miktarının incelenecek olmasıdır.

Çalışmada Tip II Regresyon tekniklerinden olan EKK-Açıortay tekniğiyle, parametreleri YSA kullanılarak hesaplanan YSA tabanlı açıortay (YSA-Açıortay) tekniği karşılaştırılması süreci aşağıdaki basamaklarla gerçekleştirilmiştir.

Öncelikle simülasyon aracılığıyla üretilen veriler üzerinde, YSA eğitiminin performansının ve verimliliğinin artırılması amacıyla bölüm iki de anlatılan verilerin işlenmesi yöntemleri kullanılmıştır. “Min-Max Normalizasyonu” yönteminin verilerdeki tüm ilişkilerin tam olarak korunması avantajı göz önünde bulundurularak verilerin normalizasyonu için bu yöntem uygun görülmüştür ve Eşitlik 3.1 kullanılarak verilerin normalizasyonu gerçekleştirilmiştir.

$$x' = (x_{max} - x_{min}) \times \frac{(x_i - x_{min})}{(x_{max} - x_{min})} + x_{min} \quad (3.1)$$

Daha sonra verilerin %70'i eğitim (training) verisi %30'u ise test (validation) verisi olarak ayrılmıştır. Veri seti eğitim için ayrılırken sırasıyla, %90, %80 ve %70 şeklinde ayrılarak deneme-yanılma yoluna gidilmiştir. En düşük hata %70'de alındığından veriler %70 eğitim %30 test olarak ayrılmıştır.

Normalizasyon sonrasında veriler girdi olarak doğrudan ağa girmiştir. Bu girdiler ağırlıklarla çarpılarak, ikinci bölümde anlatılan birleştirme fonksiyonlarından olan ağırlıklı toplam fonksiyonu Eşitlik 3.2'deki formül aracılığıyla net girdi hesaplanmıştır.

$$NET = \sum_i^n G_i A_i \quad (3.2)$$

Daha sonra bu hesaplanan çıktılar aktivasyon fonksiyonuna aktarılmıştır. Literatürde birçok aktivasyon fonksiyonu bulunmaktadır. Bu aktivasyon fonksiyonları hakkında ikinci bölümde bilgi verilmiştir. Aktivasyon fonksiyonlarının birbirlerinden herhangi bir üstünlüğü yoktur ve hangi problem için hangi aktivasyon fonksiyonunun en uygun fonksiyon olduğunu belirlemek için herhangi bir yöntem bulunmamaktadır. En uygun fonksiyonu belirlemek adına deneme yanılma yoluna gidilerek, “Sigmoid Fonksiyon, Purelin Fonksiyon, Step Fonksiyon ve Tansig Fonksiyon” denenmiştir. Denemeler sonucu veri yapısına en uygun çıktıyı veren fonksiyon “Tansig Fonksiyon” olarak belirlenmiştir.

Ağ mimarisi için yine deneme yanılma yoluna gidilerek, ikinci gizli katmanda birden yüze kadar denemeler yapılmıştır. Bu denemeler sonucu birinci gizli katmanda bir hücre, ikinci gizli katmanda ise on hücre kullanılmıştır. Öğrenme oranı farklı veri seti ve büyüklüklerde farklı sonuçlar çıkardığı ve en iyi öğrenme oranı konusunda kesin bir bilgi olmadığından sırasıyla 0,01, 0,05, 0,001, 0,005 olarak deneme yanılma yoluna gidilmiştir. En iyi sonuç 0,005 değerinde hesaplanmıştır, dolayısıyla öğrenme oranı 0,005 olarak belirlenmiştir.

YSA'nın yapısı için çok katmanlı algılayıcılardan yararlanılmıştır. Çok katmanlı algılayıcılardan ileri beslemeli ağlarından yararlanılmıştır. Ağ eğitimi için Levenberg-Marquardt algoritmasından yararlanılmıştır. Levenberg-Marquardt algoritması hakkında bilgi önceki bölümlerde verilmiştir.

YSA-AçıOrtay (YSAAO) tekniğinin kullanılabilmesi için öncelikle girdi değişkeni bağımlı değişken hedef değişkeni bağımsız değişken olarak alınmıştır. Sonrasında ise girdi değişkeni bağımsız değişken hedef değişkeni bağımlı değişken olarak alınmıştır. Bu sayede YSA(Y|X) (Y bağımlı değişken, X bağımsız değişken) denklemi ve YSA(X|Y) (X bağımlı değişken, Y bağımsız değişken) denklemleri hesaplanmıştır. Bu denklemler MATLAB programı aracılığıyla hesaplanırken, kurulan sinir ağı sonucunda gözlem

değerleriyle bir dönüştürme uygulanarak, YSA'nın eğrisel yapısı doğrusal bir yapıya dönüştürülmüştür. Bu sayede o doğrusal yapının regresyon denklemi elde edilmiştir. YSA(Y|X) ve YSA(X|Y) denklemlerinin eğim katsayıları sırasıyla  $a$  ve  $b$  olarak alınmış olup, aşağıdaki formüllerle YSA-Açıortay denkleminin eğim katsayısı Eşitlik 3.3'le, sabit katsayı Eşitlik 3.4'le ve model aracılığıyla tahmin edilen  $y_i$  değerleri ise Eşitlik 3.5'le hesaplanmıştır.

$$\hat{\beta}_{AO} = (a + b)^{-1} \left[ ab - 1 + \sqrt{(1 + a^2)(1 + b^2)} \right] \quad (3.3)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_{AO} * \bar{x} \quad (3.4)$$

$$y_{iYSAAO} = \hat{\beta}_0 + \hat{\beta}_{AO} * x_i \quad (3.5)$$

Diğer taraftan EKK-Açıortay denklemini elde etmek amacıyla EKK(Y|X) (Y bağımlı değişken, X bağımsız değişken) ve EKK(X|Y) (X bağımlı değişken, Y bağımsız değişken) denklemlerinin eğim katsayısı Eşitlik 3.6'daki formülle ve sabit katsayı değerleri ise Eşitlik 3.7'deki formülle hesaplanmıştır:

$$\hat{\beta}_1 = \frac{n \sum y_i x_i - \sum y_i \sum x_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (3.6)$$

$$\hat{\beta}_0 = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum y_i x_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (3.7)$$

EKK(Y|X) ve EKK(X|Y) denklemlerinin eğim katsayıları sırasıyla  $a$  ve  $b$  olarak alınmıştır. Daha sonrasında Eşitlik 3.3 ve Eşitlik 3.4'deki formüllerle EKK-Açıortay (EKKAO) denkleminin eğim katsayısı ve sabit katsayısı hesaplanmıştır.

Kurulan bu iki modele ilişkin performanslarının ölçülmesi ve karşılaştırılması amacıyla "Ortalama Mutlak Yüzde Hata" (MAPE-Mean Absolute Percent Error) kriteri Eşitlik 3.8'deki formül kullanılarak hesaplanmıştır.

$$MAPE = \frac{1}{n} \sum_{1}^n |(\hat{y} - y)/y| \quad (3.8)$$

MAPE kriterine ait hesaplanan deęerler dördüncü bölümde verilmiştir.

#### 4. BULGULAR

EKK-Açıortay ve YSA-Açıortay teknikleriyle oluşturulan, örneklem hacmi 350 olduğunda ki regresyon denklemleri Eşitlik 4.1-4.3'de verilmiştir:

$$\begin{aligned} \text{T(4)} \quad EKK_{Açıortay} &= 0,0652 + 0,5724 * x_i \\ YSA_{Açıortay} &= 0,0045 + 0,4264 * x_i \end{aligned} \quad (4.1)$$

$$\begin{aligned} \text{T(10)} \quad EKK_{Açıortay} &= 0,0638 + 0,4661 * x_i \\ YSA_{Açıortay} &= -0,0183 + 0,2696 * x_i \end{aligned} \quad (4.2)$$

$$\begin{aligned} \text{T(30)} \quad EKK_{Açıortay} &= -0,0448 + 0,4362 * x_i \\ YSA_{Açıortay} &= -0,0668 + 0,2350 * x_i \end{aligned} \quad (4.3)$$

EKK-Açıortay ve YSA-Açıortay teknikleriyle oluşturulan, örneklem hacmi 150 olduğunda ki regresyon denklemleri Eşitlik 4.4-4.6'da verilmiştir:

$$\begin{aligned} \text{T(4)} \quad EKK_{Açıortay} &= -0,0388 + 0,4878 * x_i \\ YSA_{Açıortay} &= -0,0818 + 0,4949 * x_i \end{aligned} \quad (4.4)$$

$$\begin{aligned} \text{T(10)} \quad EKK_{Açıortay} &= -0,0172 + 0,4354 * x_i \\ YSA_{Açıortay} &= -0,0474 + 0,3640 * x_i \end{aligned} \quad (4.5)$$

$$\begin{aligned} \text{T(30)} \quad EKK_{Açıortay} &= -0,0932 + 0,4128 * x_i \\ YSA_{Açıortay} &= 0,0088 + 0,1337 * x_i \end{aligned} \quad (4.6)$$

EKK-Açıortay ve YSA-Açıortay teknikleriyle oluşturulan, örneklem hacmi 50 olduğunda ki regresyon denklemleri Eşitlik 4.7-4.9'da verilmiştir:

$$\begin{aligned} \text{T(4)} \quad EKK_{Açıortay} &= 0,0258 + 0,5047 * x_i \\ YSA_{Açıortay} &= 0,1039 + 0,8560 * x_i \end{aligned} \quad (4.7)$$

$$\begin{aligned} \text{T(10)} \quad EKK_{Açıortay} &= 0,0636 + 0,4675 * x_i \\ YSA_{Açıortay} &= -0,0366 + 0,2087 * x_i \end{aligned} \quad (4.8)$$

$$\begin{aligned} \text{T(30)} \quad EKK_{Açıortay} &= -0,0968 + 0,4524 * x_i \\ YSA_{Açıortay} &= -0,0582 + 0,1600 * x_i \end{aligned} \quad (4.9)$$

EKK-Açıortay ve YSA-Açıortay tekniklerinin karşılaştırılması amacıyla farklı veri setleri ve örneklem hacimlerinde hesaplanan MAPE kriterine ilişkin bulgular Çizelge 4.1-4.3’de verilmiştir.

**Çizelge 4.1** Örneklem hacmi 350 iken, farklı dağılımlara ilişkin YSA-Açıortay ve EKK-Açıortay tekniklerinin MAPE değerleri

	<b>YSA-Açıortay</b>	<b>EKK-Açıortay</b>
<b>Dağılım Biçimi</b>	<b>MAPE</b>	<b>MAPE</b>
<b>T(30)</b>	<b>1,4420</b>	1,7724
<b>T(10)</b>	<b>1,7270</b>	2,4608
<b>T(4)</b>	<b>4,8993</b>	7,6225

Çizelge 4.1’e göre, örneklem hacminin n=350 olduğu durumda ve farklı dağılım türleri açısından YSA-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri incelendiğinde; T(30) için MAPE=1,4420, T(10) için MAPE=1,7210 ve T(4) için ise MAPE=4,8993 olduğu görülmektedir. EKK-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri ise; T(30) için MAPE=1,7724, T(10) için MAPE=2,4608 ve T(4) için ise MAPE=7,6225 olduğu görülmektedir. Burada, veri setinin normale göre basıklığı azaldıkça YSA-Açıortay ve EKK-Açıortay tekniklerinin ikisi için de MAPE değerlerinin artış gösterdiği görülmektedir. Çizelge 4.1 özetlenecek olursa; örneklem hacmi 350 iken verilen dağılım türleri için hesaplanan MAPE kriterine göre YSA-Açıortay tekniğinin EKK-Açıortay tekniğine göre daha düşük hatayla daha iyi sonuçlar verdiği görülmektedir.

**Çizelge 4.2** Örneklem hacmi 150 iken, farklı dağılımlara ilişkin YSA-Açıortay ve EKK-Açıortay tekniklerinin MAPE değerleri

	<b>YSA-Açıortay</b>	<b>EKK-Açıortay</b>
<b>Dağılım Biçimi</b>	<b>MAPE</b>	<b>MAPE</b>
<b>T(30)</b>	<b>1,1604</b>	1,5958
<b>T(10)</b>	<b>1,2788</b>	1,3685
<b>T(4)</b>	<b>1,3787</b>	1,3793

Çizelge 4.2’ye göre, örneklem hacminin n=150 olduğu durumda ve farklı dağılım türleri açısından YSA-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri



incelendiğinde; T(30) için MAPE=1,1604, T(10) için MAPE=1,2788 ve T(4) için MAPE=1,3787 olduğu görülmektedir. EKK-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri ise; T(30) için MAPE=1,5958, T(10) için MAPE=1,3685 ve T(4) için ise MAPE=1,3793 olduğu görülmektedir. Dağılım türü T(4) olduğunda her ne kadar MAPE değerleri her iki teknik içinde birbirlerine yakın olarak da hesaplansa YSA-Açıortay tekniği az bir farkla EKK-Açıortay tekniğine göre daha düşük çıkmıştır. Örneklem hacmi 350 olduğu durumda olduğu gibi yine dağılımın normale göre basıklığı azaldıkça MAPE değerleri YSA-Açıortay tekniğinde daha arttığı görülmektedir. Fakat bu yargıyı EKK-Açıortay tekniğinde de söylemek örneklem hacminin 150 olduğu durumda mümkün değildir. Çizelge 4.2 özetlenecek olursa; örneklem hacmi 150 iken verilen dağılım türleri için hesaplanan MAPE kriterine göre YSA-Açıortay tekniğinin EKK-Açıortay tekniğine göre daha düşük hatayla daha iyi sonuçlar verdiği görülmektedir.

**Çizelge 4.3** Örneklem hacmi 50 iken, farklı dağılımlara ilişkin YSA-Açıortay ve EKK-Açıortay tekniklerinin MAPE değerleri

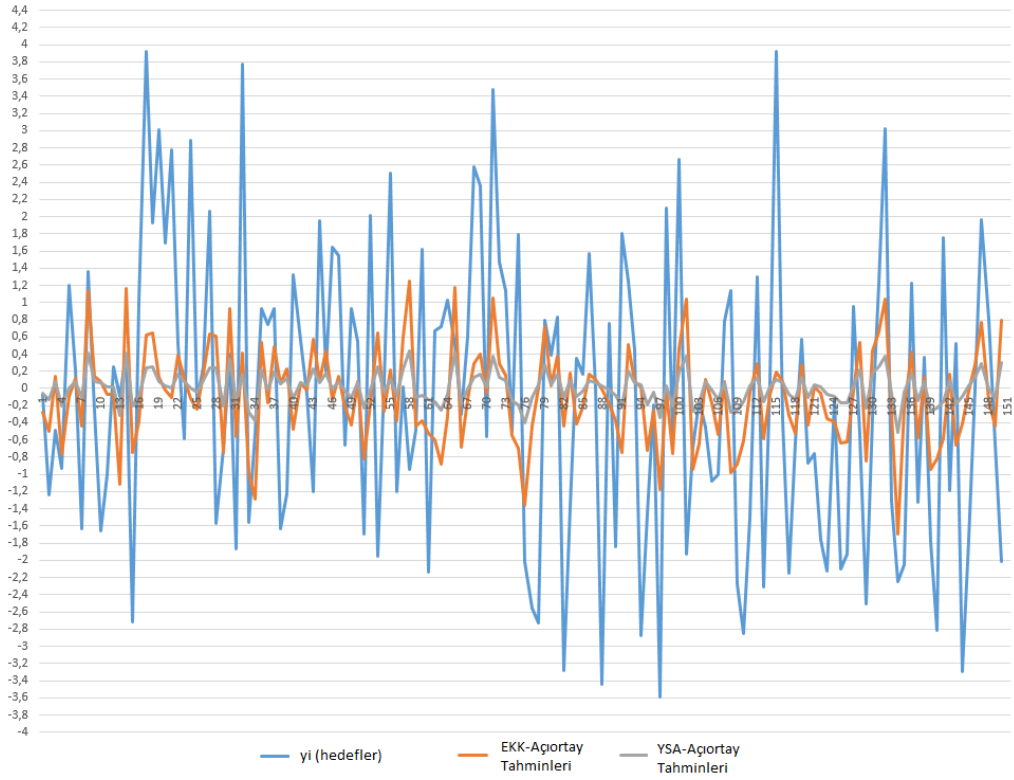
Dağılım Biçimi	YSA-Açıortay	EKK-Açıortay
	MAPE	MAPE
T(30)	<b>2,8443</b>	13,6761
T(10)	<b>2,6477</b>	2,7415
T(4)	2,0263	<b>1,3061</b>

Çizelge 4.3'e göre, örneklem hacminin n=50 olduğu durumda ve farklı dağılım türleri açısından YSA-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri incelendiğinde; T(30) için MAPE=2,8443, T(10) için MAPE=2,6477 ve T(4) için ise MAPE=2,0263 olduğu görülmektedir. EKK-Açıortay tekniği aracılığıyla kurulan modele ilişkin MAPE değerleri ise; T(30) için MAPE=13,6761, T(10) için MAPE=2,7415 ve T(4) için ise MAPE=1,3061 olduğu görülmektedir. Burada, veri setinin normale göre basıklığı azaldıkça YSA-Açıortay ve EKK-Açıortay tekniklerinin ikisi için de MAPE değerlerinin azaldığı görülmüştür. Çizelge 4.3 özetlenecek olursa; örneklem hacmi 150 iken verilen dağılım türleri için hesaplanan MAPE kriterine göre YSA-Açıortay tekniğinin EKK-Açıortay tekniğine göre daha düşük hatayla daha iyi sonuçlar verdiği görülmektedir. Fakat burada T(4) dağılımı için MAPE değerleri karşılaştırıldığında EKK-Açıortay tekniğinin daha düşük hata ile daha iyi sonuç verdiği görülmektedir.

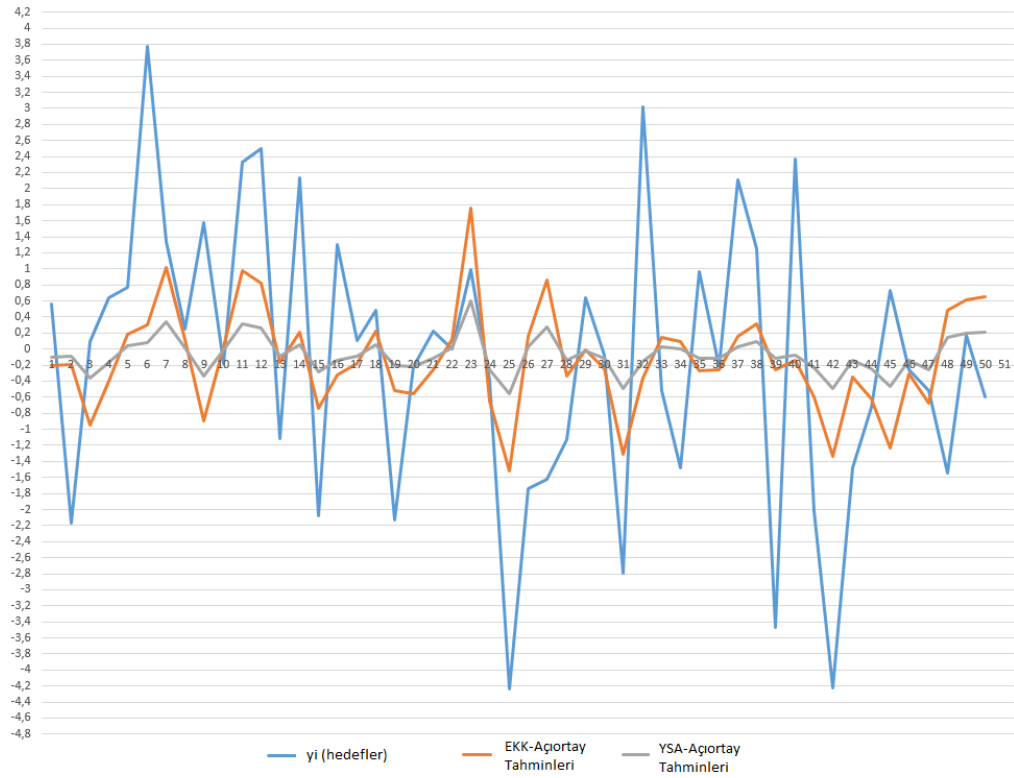
Yukarıdaki bulgulara göre, farklı dağılımlardan gelen veriler ile farklı örneklem hacimlerinde her iki teknik içinde de hesaplanan tahmin değerlerine ilişkin grafikler Şekil 4.1-4.9’da verilmiştir.



Şekil 4.1 T(30), n=350 iken hedef ve tahminlerin dağılım grafiği



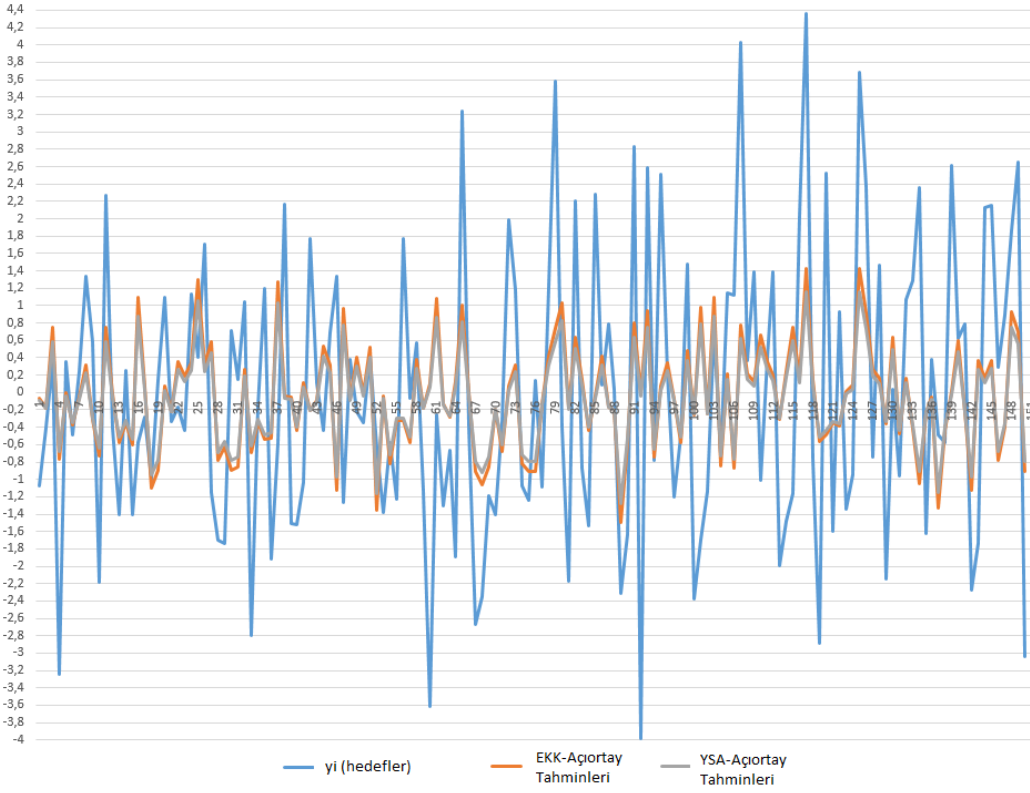
Şekil 4.2 T(30), n=150 iken hedef ve tahminlerin dağılım grafiği



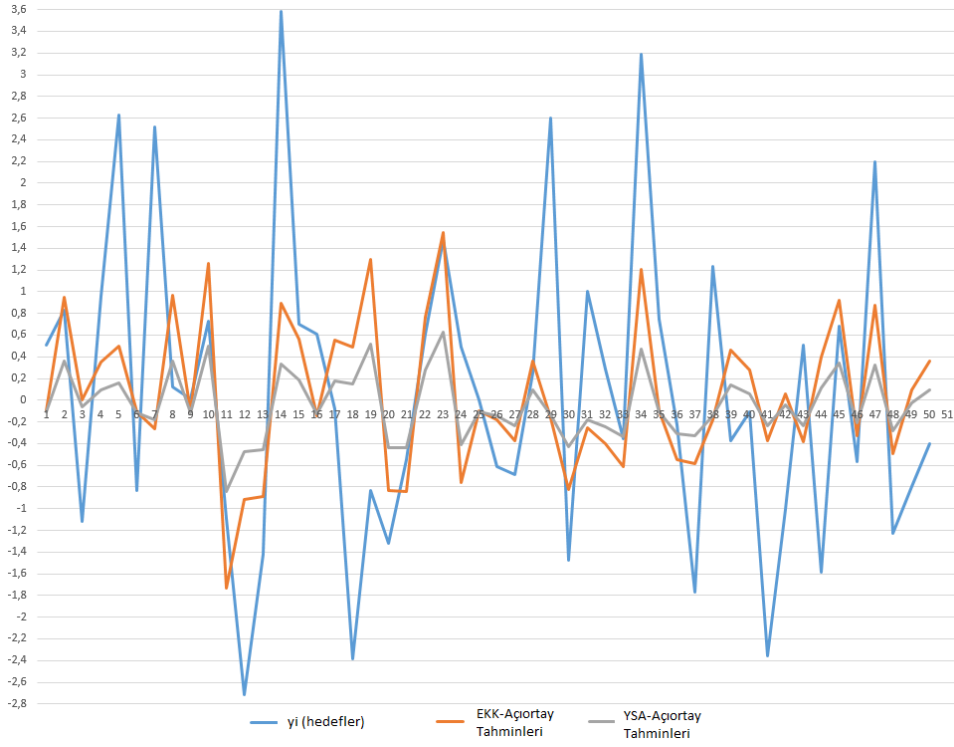
Şekil 4.3 T(30), n=50 iken hedef ve tahminlerin dağılım grafiği



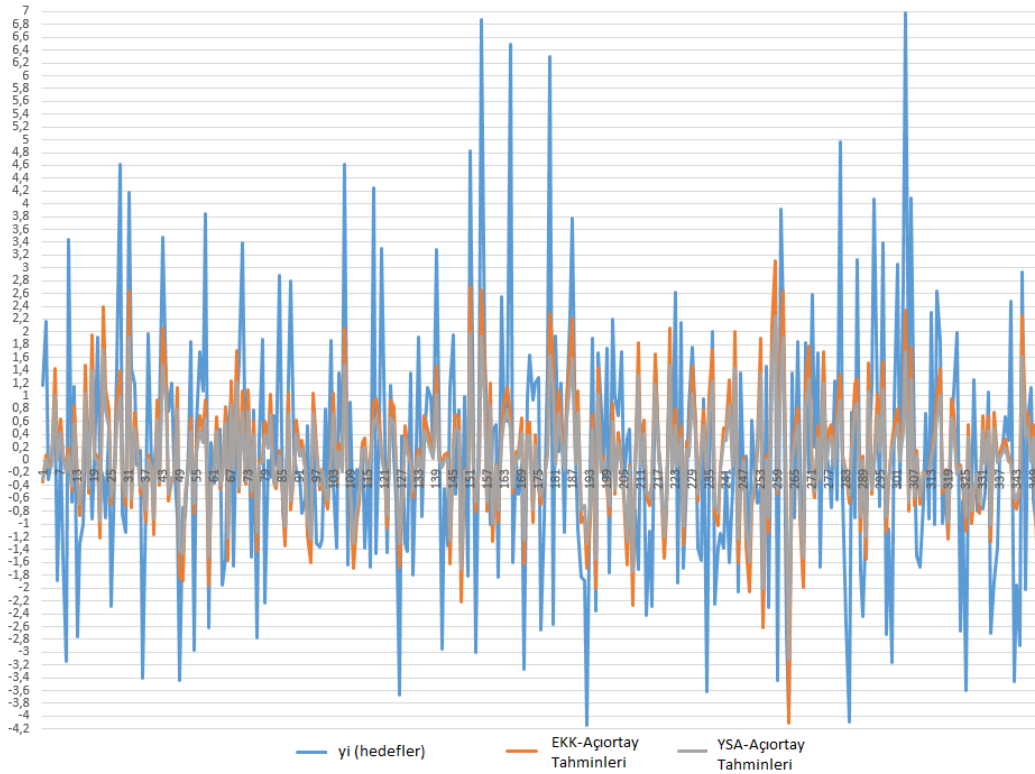
Şekil 4.4 T(10), n=350 iken hedef ve tahminlerin dağılım grafiği



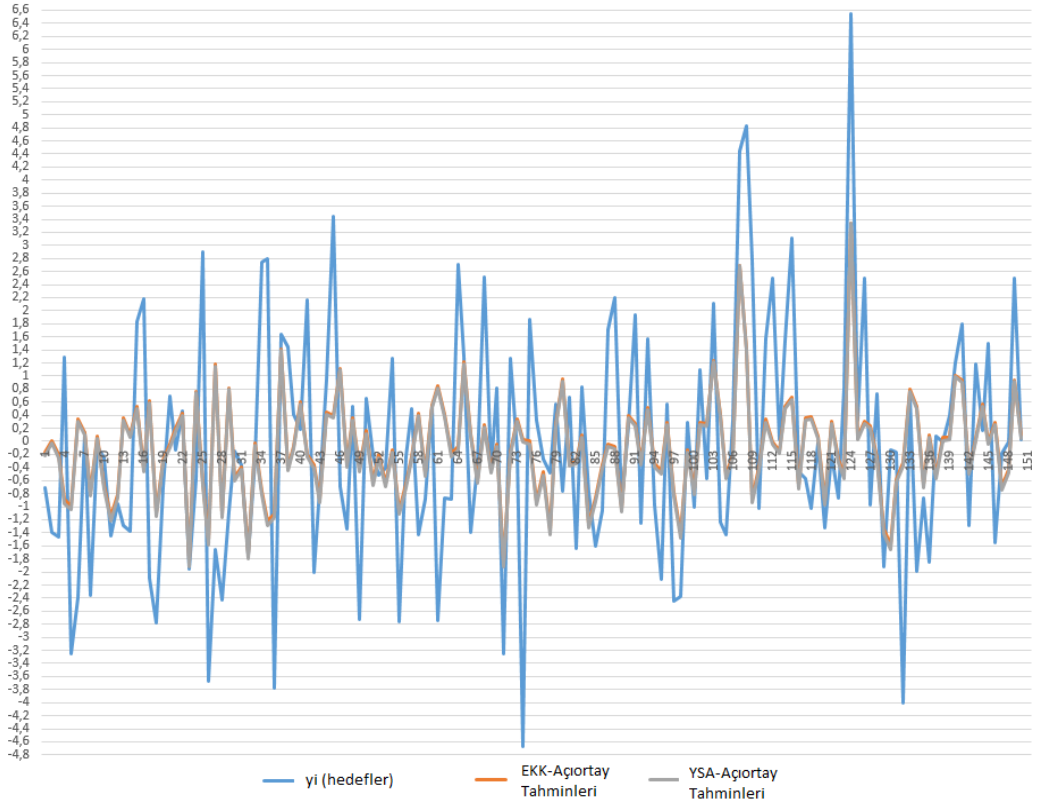
Şekil 4.5 T(10), n=150 iken hedef ve tahminlerin dağılım grafiği



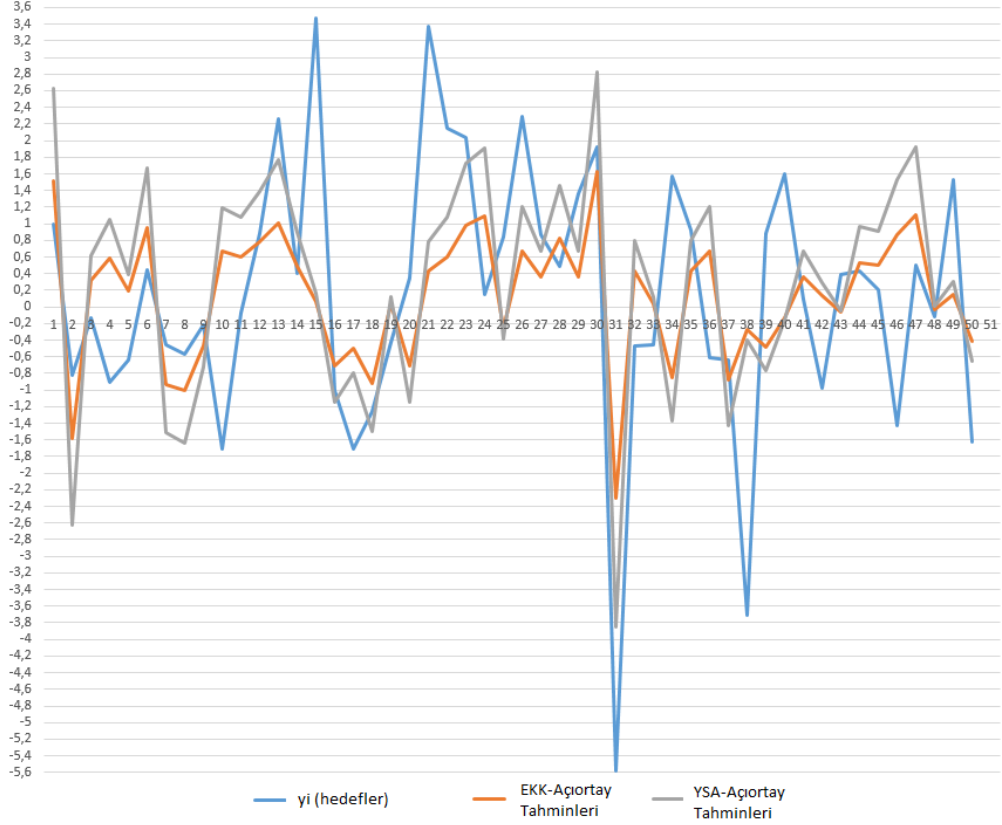
Şekil 4.6 T(10), n=50 iken hedef ve tahminlerin dağılım grafiği



Şekil 4.7 T(4), n=350 iken hedef ve tahminlerin dağılım grafiği



Şekil 4.8 T(4), n=50 iken hedef ve tahminlerin dağılım grafiği



Şekil 4.9 T(4), n=50 iken hedef ve tahminlerin dağılım grafiği

## 5. TARTIŞMA ve SONUÇ

Herhangi bir problem için bir tahminleme yapılacaksa ve varsayımlarının uygun koşulları sağlaması halinde ilk akla gelen modelleme klasik regresyon modellemesidir. Fakat bu problem için ölçümsel bir hatanın olması durumunda Tip II Regresyon tekniklerinden yararlanmanın çok daha doğru bir karar olacaktır. Çünkü Tip I Regresyon teknikleri, gerçekte var olan bağımsız değişkenlerden kaynaklanan hataların modele katılmaması Tip I Regresyon için bir dezavantajdır. Bu dezavantajdan kurulmak için Tip II Regresyon tekniklerinden yararlanmak oldukça doğru bir karardır.

Bu çalışmada, Tip II Regresyon tekniğinin bağımsız değişkendeki hataları ele almasının avantajı ile Yapay Sinir Ağı'nın tahminlemedeki üstünlüğünün birleştirilmesi ile YSA-Açıortay tekniği ortaya konulmuştur. Bu teknik ile farklı dağılım ve farklı örneklem hacimlerinde YSA-Açıortay ve EKK-Açıortay teknikleri karşılaştırılmıştır.

YSA-Açıortay tekniği ile EKK-Açıortay tekniğinin karşılaştırılması amacıyla farklı dağılımlarda ve farklı örneklem hacimlerinde MAPE kriterinden yararlanılarak analizler gerçekleştirilmiştir. Bu amaçla hesaplanan MAPE kriterine göre Yapay Sinir Ağı temelli kurulan YSA-Açıortay modelinin tahminlemedeki performansı, EKK-Açıortay tekniğine göre daha düşük hata ile daha iyi sonuçlar vermiştir.

Literatürdeki diğer Yapay Sinir Ağları'nın karşılaştırmalı çalışmalarına bakıldığında ise;

- Tolon ve Tosunoğlu (2008)'de yaptıkları bir çalışmalarında, tüketici tatmini verilerinin analizini Yapay Sinir Ağı modeli ve Klasik Regresyon modeli kullanarak iki modelin de tahmin gücünü karşılaştırmıştır. Bu karşılaştırma sonucunda Yapay Sinir Ağı modelinin daha düşük hata ile daha iyi sonuç verdiğini söylemişlerdir.
- Takma vd. (2012)'de yaptıkları bir çalışmalarında, laktasyon süt verimlerine uyum yeteneklerinin karşılaştırılması konusunda yaptıkları Çoklu Doğrusal Regresyon ve Yapay Sinir Ağı modellerinin karşılaştırmasını

gerçekleştirmişlerdir. Bu araştırma ile MSE, MAPE, MAD, RMSE ve  $R^2$  kriterlerine göre bu iki modelin karşılaştırılmasında, Yapay Sinir Ağı modelinin Çoklu Doğrusal Regresyon modeline göre daha düşük hata ile daha iyi sonuçlar verdiğini belirtmişlerdir.

- Budak ve Erpolat (2012)'de kredi riski tahmini için yaptıkları çalışmayla, Lojistik Regresyon ve Yapay Sinir Ağı'nın karşılaştırmasını uygulamaya dökmüşlerdir. Bu çalışmayla, Yapay Sinir Ağı'nın tahminlemedeki gücünün Lojistik Regresyon'a göre daha üstün olduğunu söylemişlerdir. Ayrıca Yapay Sinir Ağı modelinin Lojistik regresyona göre daha iyi bir doğru sınıflandırma oranına sahip olduğu görülmüştür.
- Kaynar ve Taştan (2009)'da yaptıkları çalışmayla, bir zaman serisi analizi için Yapay Sinir Ağı modeli ile ARIMA modelinin karşılaştırmasını ele almışlardır. Bu karşılaştırma sonucunda elde edilen bulgulara göre finansal veriler için Yapay Sinir Ağı modelinin başarılı bir model olarak kullanılabileceğini söylemişlerdir.
- Aynı zamanda Tosun (2007)'de üniversite öğrencilerinin başarılarını etkileyen faktörlerin ele alındığı bir çalışmada, Yapay Sinir Ağı ve Karar Ağaçları metodunu karşılaştırmıştır. Bu çalışma sonucu Yapay Sinir Ağı, Karar Ağaçları metoduna göre daha yüksek bir başarı oranı ile daha iyi sonuçlar vermiştir.
- Aygören vd. (2012)'de yaptıkları bir çalışma ile klasik zaman serileri için ARMA (p, q), Nümerik Arama Modelleri için Newton Tekniği ve Yapay Sinir Ağı modeli için ise Geri Yayılım algoritmasından yararlanarak bir karşılaştırma gerçekleştirmişlerdir. Bu karşılaştırma sonucunda, klasik zaman serisi modeli ve nümerik arama modeli yapay sinir ağı modeline göre daha düşük bir performans sergileyerek, yine tahminleme problemlerindeki başarısını göstermiştir.

Bu literatürdeki YSA karşılaştırmalarında olduğu gibi bu çalışmada da, YSA tabanlı Açığortay tekniği, EKK tabanlı Açığortay tekniğine göre daha iyi sonuçlar verdiği görülmüştür. Aynı zamanda bu YSA-Açığortay tekniğinin bağımsız değişkenlerdeki



hatayı da ele aldığı için klasik regresyon yöntemlerine göre daha başarılı bir tahminleme yapabileceği öngörülmektedir.

Bu çalışma için, Yapay Sinir Ağı'nın optimizasyonu konusunda herhangi bir çalışma yapılmamıştır. Ağdaki gizli değişkenin hücre sayısı, aktivasyon fonksiyonu gibi parametreler denemeler sonucundaki elde edilen en iyi sonuçlar hesaplanmıştır. Başka bir çalışmayla, farklı ağ parametreleri alınarak daha iyi sonuçlarında elde edilebileceği ön görülmektedir. Bu çalışmayla Yapay Sinir Ağı tekniğinin diğer tahminleme yöntemlerine göre daha iyi sonuçlar vererek, oldukça başarılı bir alternatif teknik olarak kullanılabilirliği ortaya konulmuştur.

## 6. KAYNAKLAR

- Adcock, R.J. (1878). A Problem in Least Squares. *Annals of Mathematics*, **5(2)**: 53-54.
- Adıyaman, F. (2007). Talep Tahmininde Yapay Sinir Ağlarının Kullanılması. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Agatonovic-Kustrin, S. and Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, **22**: 717-727.
- Akkoca, G. (2012). Klinik Araştırmalarda Sürekli Sonuçlu Ölçüm Tekniklerinin Uyumunun İncelenmesinde Kullanılan İstatistiksel Yöntemler. Yüksek Lisans Tezi, Ankara Üniversitesi, Sağlık Bilimleri Enstitüsü, Ankara.
- Aksungur, S. (2009). Üç Serbestlik Dereceli Bir Robotun, Yapay Sinir Ağları ve Genetik Algoritma Kullanılarak Engelli Ortamda Çarpışmasız Yörünge Planlaması. Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- Aladağ, Ç.H. (2009). Yapay Sinir Ağlarının Mimari Seçimi için Tabu Arama Algoritması. Doktora Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü. Ankara.
- Alma, G.Ö. ve Vupa, Ö. (2008). Regresyon Analizinde Kullanılan En Küçük Kareler ve En Küçük Medyan Kareler Yöntemlerinin Karşılaştırması. *SDÜ Fen Edebiyat Fakültesi Fen Dergisi (E-Dergi)*, **3(2)**: 219-229.
- Alpaydın, E. (2011). Yapay Öğrenme. Boğaziçi Üniversitesi Yayınevi, İstanbul.
- Altan, Ş. (2008). Döviz Kuru Öngörü Performansı için Alternatif Bir Yaklaşım: Yapay Sinir Ağı. *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **10(2)**: 141-160.
- Amasyalı, M.F. (2015). Yapay Sinir Ağlarında Parçalı Eğitim. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, **8(1)**: 25-33.
- Amman, L. and Ness, J.V. (1988). A Routine for Converting Regression Algorithms in to Corresponding Orthogonal Regression Algorithms. *ACM Transactions on Mathematical Software*, **14(1)**: 76-87.

- Anderson, D. and McNeill, G. (1992). Artificial Neural Networks Technology. Kaman Sciences Corporation, Utica, New York.
- Arı, A. ve Erşen, M.B. (2017). Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. *Acta INFOLOGICA Dergisi*, **1(2)**: 55-73
- Arı, A. ve Önder, H. (2013). Farklı Veri Yapılarında Kullanılabilecek Regresyon Yöntemleri, *Anadolu Tarım Bilim Dergisi*, **28(3)**: 168-174.
- Ataseven, B. (2007). Satış Öngörü Modellemesi Tekniği Olarak Yapay Sinir Ağlarının Kullanımı: “PETKİM’de Uygulanması”, Yüksek Lisans Tezi, Celal Bayar Üniversitesi, Sosyal Bilimler Enstitüsü, Manisa.
- Ataseven, B. (2013). Yapay Sinir Ağları ile Öngörü Modellemesi. *Öneri Dergisi*, **10(39)**: 101-115.
- Aygören, H., Sarıtaş, H. ve Moralı, T. (2012). İMKB 100 Endeksinin Yapay Sinir Ağları ve Newton Nümerik Arama Modelleri ile Tahmini. *Uluslararası Alanya İşletme Fakültesi Dergisi*, **4(1)**: 73-88.
- Baydili, K.N. (2015). Deming Regresyonu ve Türetilmiş Verilerde Uygulanması. Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- Budak, H. ve Erpolat, S. (2012). Kredi Riski Tahmininde Yapay Sinir Ağları ve Lojistik Regresyon Analizi Karşılaştırılması. *AJIT-e: Online Academic Journal of Information Technology*, **3(9)**: 23-30.
- Canan, S. (2006). Yapay Sinir Ağları ile GPS Destekli Navigasyon Sistemi. Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- Carroll, R.J. and Ruppert, D. (1996). The Use and Misuse of Orthogonal Regression in Linear Errors in Variables Models. *The American Statistician*, **50(1)**: 1-6.
- Cornbleet, P.J. and Gochman, N. (1979). Incorrect Least-Squares Regression Coefficients in Method-Comparison Analysis. *Clin. Chem.*, **25(3)**: 432-438.
- Çavuşlu, M.A., Becerikli, Y. ve Karakuzu, C. (2012). Levenberg-Marquardt Algoritması ile YSA Eğitiminin Donanımsal Gerçekleşmesi. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, **5(1)**.

- Çelik, B. (2008). Yapay Sinir Ağları Metodolojisi ile Zaman Serisi Analizi: Teori ve Uygulama. Yüksek Lisans Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- Deming, W.E. (1943). Statistical Adjustment of Data. *Annals Math. Stat.*, **21**: 488-506.
- Ding, G., Chu, B., Jin, Y., and Zhu, C. (2013). Comparison of orthogonal regression and least squares in measurement error modeling for prediction of material property, Nanotechnology and Material Engineering Research. *Advanced Materials Research*, **661**: 166-170.
- Doğan, E., Işık, S. ve Sandalcı, M. (2007). Günlük Buharlaşmanın Yapay Sinir Ağları Kullanarak Tahmin Edilmesi. *İMO Teknik Dergisi*, **271**: 4119-4131.
- Doğan, G. (2010). Yapay Sinir Ağları Kullanılarak Türkiye'deki Özel Bir Sigorta Şirketinde Portföy Değerlendirmesi. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Efendigil, T. (2008). Müşteri Odaklı Sistemler İçin Yapay Sinir Ağı ve Bulanık Çıkarım Tabanlı Bir Karar Destek Sistemi Yaklaşımı. Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Elmas, Ç. (2003). Yapay Sinir Ağları. Seçkin Yayıncılık, Ankara.
- Elmas, Ç. (2010). Yapay Zekâ Uygulamaları, Seçkin Yayıncılık, Ankara.
- Ergezen, H. Dikmen, M. ve Özdemir, E. (2003). Yapay Sinir Ağları ve Tanıma Sistemleri. *PILVOLKA*, **6**: 14-17.
- Fautsett, L. (1994). Fundamentals of Neural Networks: Architectures, Algorithms and Applications. Pearson Education, Delhi, India.
- Fullér, R. (1995). Neural Fuzzy Systems. Åbo Academy University, Åbo.
- Fuller, W.A. (1987). Measurement Error Models. John Wiley & Sons, New York.
- Gör, İ. (2016). Çok Katmanlı Algılayıcı Yapay Sinir Ağı ile Lineer Diferansiyel Denklem Sisteminin Çözümü. 18. Bilişim Konferansı, Adnan Menderes Üniversitesi, Aydın, 3-5 Şubat.
- Görtz, G. and Nebel, B. (2003). Künstliche Intelligenz. Fischer Taschenbuch, 1. Edition, Frankfurt, Germany.

- Graupe, D. (2007). Principle of Artificial Neural Networks, 2nd Edition. World Scientific, USA.
- Gurney, K. (1996). Computers and Symbols versus Nets and Neurons, UCL Draft Papers, sf: 4.
- Gutierrez, R.S., Solis, A.O. and Mukhopadhyay S. (2008). Lumpy demand forecasting using neural networks, *International Journal of Production Economics*, **111**: 409-420.
- Günay, S., Eğrioglu, E. ve Aladağ, Ç.H. (2007). Tek değişkenli zaman serileri analizine giriş. Hacettepe Üniversitesi Yayınları, Ankara: 230s.
- Hameed, A.A. (2017). Robust Adaptive Learning Approach of Artificial Neural Networks. Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü. Konya.
- Hamzaçebi, C. (2011). Yapay Sinir Ağları. Ekin Yayıncılık. Bursa.
- Harper, W.V. (2014). Reduced Major Axis Regression: Teaching Alternatives to Least Squares. 9th International Conference on the Teaching of Statistics, Flagstaff, Arizona, USA, 13-18 July,
- Haykin, S. (1999). Neural Networks A Comprehensive Foundation. Prentice Hall International, Inc, Canada.
- Hernández, M.A. (1992). Newton-Raphson's Method and Convexity. *Zb. Rad. Prirod-Mat. Fax. Ser. Mat.*, **22(1)**: 159-166.
- Hopfield, J.J. and Tank, D.W. (1985). Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics*, **52**: 141-152.
- Isobe, T., Feigelson, E.D., Akritas, M.G. and Babu, G.J. (1990). Linear Regression in Astronomy I. *The Astrophysical Journal*, **364**: 104-113.
- Jayalakshmi, T. and Santhakumaran, A. (2011). Statistical Normalization and Back Propagation for Classification. *International Journal of Computer Theory and Engineering*, **3(1)**: 1793-8201.
- Jolicoeur, P. (1990). Bivariate Allometry: Interval Estimation of the Slopes of the Ordinary and Standardized Normal Major Axes and Structural Relationship, *Journal of Theoretical Biology*, **144**: 275-285.

- Kalaycı, E. (2016). SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri. Küçüksille, E., (Ed.), Basit Doğrusal Regresyon, Asil Yayıncılık. Ankara.
- Karahan, M. (2011). İstatistiksel Tahmin Yöntemleri: Yapay Sinir Ağları Metodu İle Ürün Talep Tahmini Uygulaması. Doktora Tezi, Selçuk Üniversitesi, Sosyal Bilimler Enstitüsü, Konya.
- Kaynar, O. ve Taştan, S. (2009). Zaman Serisi Analizinde MLP Yapay Sinir Ağları ve Arıma Modelinin Karşılaştırılması. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **33**: 161-172.
- Keleş, T. ve Altun, M. (2016). Comparison of Classical Linear Regression and Orthogonal Regression with Respect to the Sum of Squared Perpendicular Distances. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, **7(2)**: 296-308.
- Krenker, A., Beşter, J. and Kos, A. (2011). Introduction to the Artificial Neural Networks. In: Suzuki, K. (Eds.), *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, InTech, USA. 1-18.
- Kocak, C. (2015). A New High Order Fuzzy ARMA Time Series Forecasting Method by Using Neural Networks to Define Fuzzy Relations. 14 pages. In: Egrioglu, E., Khashei, M., Aladag, C.H., Turksen, B.I. and Yolcu, U. (Eds.), *Advanced Time Series Forecasting Methods, Mathematical Problems in Engineering*, Vol. 2015.
- Kunt, F. (2007). Hava Kirliliğinin Yapay Sinir Ağları Yöntemiyle Modellenmesi Ve Tahmini. Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- Legendre, P and Legendre, L. (2012). *Numerical Ecology*. Elsevier, 3. English edition, Amsterdam, UK.
- Li, J. (2012). Multivariate Generalization of Reduced Major Axis Regression. PhD Thesis, Arizona State University, Arizona.
- Linnet, K. (1990). Estimation of the Linear Relationship between the Measurements of Two Methods with Proportional Errors. *Statistics in Medicine*, **9(12)**: 1463-1473.
- Linnet, K. (1993). Evaluation of Regression Procedures for Methods Comparison Studies. *Clin. Chem.*, **39(3)**: 424-432.

- Mandel, J. (1964). The Statistical Analysis of Experimental Data. John Wiley and Sons, New York, 290-231.
- McArdle, B.H. (1988). The Structural Relationship: Regression in Biology. *Canadian Journal of Zoology*, **66(11)**: 2329-2339.
- Mendel, J.M. and McLaren, R. W. (1970). Reinforcement Learning Control and Pattern Recognition Systems. *Mathematics in Science and Engineering*, **66**: 287-318.
- Mesplé, F., Troussellier, M., Casellas, C. and Legendre, P. (1996). Evaluation of Simple Statistical Criteria to Qualify a Simulation. *Ecological Modelling*, **88**: 9-18.
- Minsky, M. and Papert, S. (1969). Perceptrons. The MIT Press.
- Öztemel, E. (2006). Yapay Sinir Ağları (2. Baskı). Papatya Yayınları. İstanbul.
- Öztemel, E. (2012). Yapay Sinir Ağları. Papatya Yayıncılık. İstanbul.
- Öztürk, K. ve Şahin, M.E. (2018). Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. *Takvim-i Vekayi*, **6(2)**: 25-36.
- Pankaj, M., and Wah, B.W. (1992). Artificial Neural Networks: Concepts and Theory. IEEE Computer Society Press, Washington, USA.
- Parvin, C.A. (1984). A Direct Comparison of Two Slope-Estimation Techniques Used in Method Comparison Studies. *Clin. Chem.*, **30(5)**: 751-754.
- Passing, H. and Bablok, W. (1983). A New Biometrical Procedure for Testing the Equality of Measurements from Two Different Analytical Methods. Application of linear regression procedures for method comparison studies in Clinical Chemistry, Part I. *Journal of Clinical Chemistry & Clinical Biochemistry*, **21**: 709-720.
- Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, **2**: 559-572.
- Pozan, Ö. (2006). Yapay Zeka. İnkılap Kitapevi. İstanbul.
- Rojas, R. (1996). Neural Networks – A Systematic Introduction. Springer-Verlag, Berlin.
- Ross, S.M. (2009). Introduction to Probability and Statistics for Engineers and Scientists. Çev. Çelebioğlu, S. ve Kasap, R. (2012). Olasılık ve İstatistiğe Giriş. Bal, H., (Ed.) Regresyon. Nobel Yayıncılık. Ankara.

- Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986). Learning Representations by Back-propagation Errors. *Nature*, **323**: 533-536.
- Saraçlı, S. (2008). Ölçüm Hatalı Modellerde Doğrusal Regresyon Tekniklerinin Karşılaştırılması -Monte-Carlo Simülasyon Çalışması-. Doktora Tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü. Eskişehir.
- Saraçlı, S. and Çelik, E.H. (2011). Performance of OLS-Bisector Regression in Method Comparison Studies. *World Applied Sciences Journal*, **12(10)**: 1860-1865.
- Saraçlı, S., Doğan, İ. ve Doğan, N. (2009). Medikal Metot Karşılaştırma Çalışmalarında Deming Regresyon Tekniği. *Türkiye Klinikleri Biyoistatistik Dergisi*, **1(1)**: 9-15.
- Saygılı, Y.S. (2008). İstatistiksel Yöntemlerle Yapay Sinir Ağları Uygulamalarının Karşılaştırılması: Millî Savunma Bakanlığı Bütçesinin Öngörülmesi. Yüksek Lisans Tezi, Kara Harp Okulu, Savunma Bilimleri Enstitüsü, Ankara.
- Saylor, R.D., Edgerton, E.S. and Hartsell, B.E. (2006). Linear Regression Techniques for Use in the EC Tracer Method of Secondary Organic Aerosol Estimation. *Atmospheric Environment*, **40**: 7546-7556.
- Sokal, R.R. and Rohlf, F.J. (1995). Biometry: The Principles and Practice of Statistics in Biological Research. New York W.H Freeman. New York.
- Takma, Ç., Atıl, H. ve Aksakal, V. (2012). Çoklu Doğrusal Regresyon ve Yapay Sinir Ağı Modellerinin Laktasyon Süt Verimlerine Uyum Yeteneklerinin Karşılaştırılması. *Kafkas Üniversitesi Vet. Fakültesi Dergisi*, **18(6)**: 941-944.
- Tolon, M. ve Tosunoğlu, N.G. (2008). Tüketici Tatmini Verilerinin Analizi: Yapay Sinir Ağları ve Regresyon Analizi Karşılaştırması. *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **10(2)**: 247-259.
- Tosun, S. (2007). Sınıflandırmada Yapay Sinir Ağları ve Karar Ağaçları Karşılaştırması: Öğrenci Başarıları Üzerine Bir Uygulama. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü. İstanbul.
- Ural, A. ve Kılıç, İ. (2018). Bilimsel Araştırma Süreci ve SPSS ile Veri Analizi (5. Baskı). Detay Yayıncılık. Ankara.



- Wakkers, P.J., Hellendoorn, H.B., Op de Weegh, G.J. and Heersping, W. (1975). Applications of Statistics in Clinical Chemistry. A Critical Evaluation of Regression Lines. *Clin Chim Acta*, **64(2)**: 173-184.
- Wu, C. and Yu, J.Z. (2018). Evaluation of Linear Regression Techniques for Atmospheric Applications: The Importance of Appropriate Weighting. *Atmospheric Measurement Techniques*, **11**: 1233-1250.
- Yavuz, S. ve Deveci, M. (2012). İstatistiksel Normalizasyon Tekniklerinin Yapay Sinir Ağın Performansına Etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **40**: 167-187.
- Yılmaz, A. (2017). Yapay Zekâ. Kodlab Yayınları. İstanbul.
- York, D., Evensen, N.M., Martinez, M.L. and Delgado, J.D.B. (2004). Unified Equations for the Slope, Intercept, and Standard Errors of the Best Straight Line. *American Journal of Physics*, **72**: 367-375.
- Yurtoğlu, H. (2005). Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği. Devlet Planlama Teşkilatı-Uzmanlık Tezleri, Yayın No: Dpt: 2683.
- Yüksek, A.G., Bircan, H., Zontul, M. ve Kaynar, O. (2007). Sivas İlinde Yapay Sinir Ağları ile Hava Kalitesi Modelinin Oluşturulması Üzerine Bir Uygulama, *C. Ü. İktisadi ve İdari Bilimler Dergisi*, **8(1)**: 97-112.

## İnternet Kaynakları

- 1) [http://hagan.ecen.ceat.okstate.edu/4\\_Perceptron.pdf](http://hagan.ecen.ceat.okstate.edu/4_Perceptron.pdf), 04.12.2018
- 2) <http://www.alicavuslu.gen.tr/2014/07/20/levenberg-marquardt-algoritmasi>,  
03.04.2019
- 3) <https://cran.r-project.org/web/packages/lmodel2/vignettes/mod2user.pdf>, 21.02.2019
- 4) <https://docplayer.biz.tr/24509964-Yapay-sinir-aglari-artificial-neural-networks-doc-dr-ersan-kabalci-e-s-n-e-k-h-e-s-a-p-l-a-m-a-y-o-n-t-e-m-l-e-r-i-ii.html>, 05.01.2019
- 5) [https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Deming\\_Regression.pdf](https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Deming_Regression.pdf), 15.05.2019
- 6) [https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Passing-Bablok\\_Regression\\_for\\_Method\\_Comparison.pdf](https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Passing-Bablok_Regression_for_Method_Comparison.pdf), 27.04.2019
- 7) <http://documentation.statsoft.com/STATISTICAHelp.aspx?path=NonlinearEstimation/NonlinearEstimation/Overview/MajorAxisRegression>, 27.04.2019
- 8) <https://www.xlstat.com/en/solutions/features/passing-and-bablok-regression>,  
27.04.2019

## ÖZGEÇMİŞ

Adı Soyadı : Berkalp TUNCA  
Doğum Yeri ve Tarihi : Altındağ / 14.03.1994  
Yabancı Dili : İngilizce  
İletişim (Telefon/E-posta) : berkalp0606@gmail.com

### Eğitim Durumu (Kurum ve Yıl)

Lise : 2008-2012 GAZİ ÇİFTLİĞİ ANADOLU LİSESİ  
(SAYISAL) (GAZİ MAHALLESİ/ANKARA)  
Lisans : 2013-2017 AFYON KOCATEPE ÜNİVERSİTESİ  
(İSTATİSTİK BÖLÜMÜ)  
Yüksek Lisans : 2017- AFYON KOCATEPE ÜNİVERSİTESİ  
(UYGULAMALI İSTATİSTİK ANABİLİM DALI)

### Çalıştığı Kurum/Kurumlar ve Yıl

### Yayımları (SCI ve diğer)

**Tunca B.**, Saraçlı S., *Artificial Neural Network approach on Type II Regression Analysis*, 11<sup>th</sup> International Statistics Days Conference, 3 - 7 October 2018 Muğla, Turkey.

### Diğer Konular