

T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
EKONOMETRİ ANABİLİM DALI  
EKONOMETRİ PROGRAMI  
YÜKSEK LİSANS TEZİ

**YAPAY SİNİR AĞLARINDA DUYARLILIK  
ANALİZLERİ**

**Serkan ARAS**

Danışman

**Doç. Dr. İpek DEVECİ KOCAKOÇ**

2008

## Yemin Metni

Yüksek Lisans Tezi olarak sunduğum “**Yapay Sinir Ağlarında Duyarlılık Analizleri**” adlı çalışmanın, tarafımdan, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Tarih

.../.../...

Serkan ARAS

İmza

## YÜKSEK LİSANS TEZ SINAV TUTANAĞI

### Öğrencinin

**Adı ve Soyadı** : Serkan ARAS  
**Anabilim Dalı** : Yöneylem  
**Programı** : Ekonometri  
**Tez Konusu** : Yapay Sinir Ağlarında Duyarlılık Analizleri  
**Sınav Tarihi ve Saati** :

Yukarıda kimlik bilgileri belirtilen öğrenci Sosyal Bilimler Enstitüsü'nün ..... tarih ve ..... sayılı toplantısında oluşturulan jürimiz tarafından Lisansüstü Yönetmeliği'nin 18. maddesi gereğince yüksek lisans tez sınavına alınmıştır.

Adayın kişisel çalışmaya dayanan tezini ..... dakikalık süre içinde savunmasından sonra jüri üyelerince gerek tez konusu gerekse tezin dayanağı olan Anabilim dallarından sorulan sorulara verdiği cevaplar değerlendirilerek tezin,

BAŞARILI OLDUĞUNA	<input type="radio"/>	OY BİRLİĞİ	<input type="radio"/>
DÜZELTİLMESİNE	<input type="radio"/>	OY ÇOKLUĞU	<input type="radio"/>
REDDİNE	<input type="radio"/>		

ile karar verilmiştir.

Jüri teşkil edilmediği için sınav yapılamamıştır.  O\*\*\*  
Öğrenci sınava gelmemiştir.  O\*\*

- \* Bu halde adaya 3 ay süre verilir.  
\*\* Bu halde adayın kaydı silinir.  
\*\*\* Bu halde sınav için yeni bir tarih belirlenir.

Tez burs, ödül veya teşvik programlarına (Tüba, Fulbright vb.) aday olabilir.	<input type="radio"/>	Evet
Tez mevcut hali ile basılabilir.	<input type="radio"/>	
Tez gözden geçirildikten sonra basılabilir.	<input type="radio"/>	
Tezin basımı gerekliliği yoktur.	<input type="radio"/>	

### JÜRİ ÜYELERİ

### İMZA

.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....
.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....
.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....

## ÖZET

Yüksek Lisans Tezi

Yapay Sinir Ağlarında Duyarlılık Analizleri

Serkan ARAS

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Yöneylem Anabilim Dalı

Ekonometri Programı

Yapay Sinir Ağlarını (YSA) insan beyninin çalışma mantığından esinlenerek geliştirilmiş, örneklerden öğrenen ve öğrendiklerinden çıkarsamalar yapabilen bir istatistiksel metot olarak tanımlamak mümkündür. YSA, regresyon modelleri ve diğer ilgili istatistiksel tekniklerin geleneksel olarak kullanıldığı genel kapsamda tahminleme ve sınıflandırma alanlarında başarıyla kullanılmaktadır. İstatistik metotlara olan üstünlüğü gürültülü, doğrusal olmayan, eksik verilerin bulunduğu durumlarda eldeki girdi ve çıktı değişkenleri arasındaki ilişkiyi temel alarak, kabul edilebilir hata düzeyiyle sonuçlar oluşturabilmesidir. YSA, bu işlemi ağ içerisindeki hücrelerin birbiriyle iletişimini sağlayan bağlantı ağırlık değerlerini iteratif olarak değiştirerek yerine getirmektedir. Bağlantı ağırlıklarının nasıl değiştirileceğine karar verme işini öğrenme algoritması üstlenmektedir. Fakat klasik YSA yaklaşımlarında girdi ile çıktılar ilişkilendirilirken, hangi girdinin çıktı üzerinde ne yönde ve ne büyüklükte bir etkiye sahip olduğu belirlenmemektedir. Özellikle sosyal bilimler açısından ele alındığında amaç sadece basit bir tahminleme oluşturmak değil incelenen olayın açıklanabilmesidir.

Bu çalışmada girdi ve çıktı parametreleri arasındaki ilişki duyarlılık analiziyle açıklanmaya çalışılmıştır. Kullanılan duyarlılık analizi yöntemi, geri yayılım algoritmasının türetildiği zincir kuralından bir kez daha yararlanarak girdilerin çıktılar üzerindeki etkisinin belirlenmesine dayanmaktadır. Önerilen yöntemin doğruluğunun ve tutarlılığının gösterilmesi amacıyla incelenen problemin altında yatan fonksiyonun doğrusal olduğu varsayılmıştır. Böylelikle bu doğrusal fonksiyonun bilinen katsayı parametrelerinin tahmini sinir ağı ve klasik, bilindik çoklu doğrusal regresyon analiziyle karşılaştırmalı olarak yapılmıştır. Bununla da yetinilmemiş çoklu doğrusal regresyon analizinin bazı varsayımlarının sağlanmadığı durumlarda her iki yöntemden elde edilen sonuçların araştırılmasına gidilmiştir.

Çalışma sonucunda elde edilen bulgulara göre önerilen metot YSA 'nın işleyişi üzerine herhangi bir kısıt getirmemekte ve incelenen problemin altında yatan fonksiyonun modeli hakkında bir bilgiye ihtiyaç duymamaktadır.

**Doğrusal regresyon analizinin çeşitli varsayımlarının sağlanmadığı durumlarda regresyon analiziyle yapılan katsayı tahmininin kötüleştiği gözlemlenmiştir. Sinir ağlarında duyarlılık analizi adı altında tanıtılan metottan elde edilen sonuçlarda önemli herhangi bir farklılığa rastlanmamıştır. Böylece girdi ve çıktı değişkenleri arasındaki ilişkinin belirlenmesi sağlanmıştır. Bu bağlamda çalışmada klasik olarak YSA için söylenen kara kutu benzetmesinin ortadan kaldırılmasına yönelik bir adım atılmıştır.**

**Anahtar Kelimeler:** Sinir Ağları, Geri Yayılım, Duyarlılık Analizi, Levenberg-Marquardt, Conjugate Gradient

**ABSTRACT**  
**Master Thesis**  
**Sensitivity Analysis in Artificial Neural Networks**

**Serkan ARAS**  
**Dokuz Eylul University**  
**Institute of Social Sciences**  
**Department of Econometrics**  
**Programme of Econometrics**

**Artificial Neural Networks (ANN) that is developed with a high inspiration from the processing of the brain is a statistical methodology which can learn from samples and can make assessments and inferences from the generated knowledge. ANN has a successful usage in the areas of estimation and clustering in which regression analysis and related statistical techniques are traditionally used. ANN outperforms the traditional statistical methods in the existence of noised, nonlinear and incomplete data by relying on the relationships between the input and output variables in hand. ANN achieves this operation by iteratively altering the weight values providing the connection of the cells inside the network. The learning algorithm carries the duty of deciding how to change this connection weights. However during this association of the input and output variables, the direction and strength of the effects between the output and input variables cannot be defined by utilizing the classical approach. Especially in the aspect of social sciences, the goal is not only to maintain a simple estimation but also to understand and describe the studied phenomenon.**

**In this study the relationship between input and output parameters is attempted to be explained by using the sensitivity analysis. The proposed version of the sensitivity analysis is based on defining the effect of input variables on the output variables with a second time usage of the chain rule where the backpropagation algorithm derived. In order to demonstrate the accuracy and consistency of the proposed method the underlying function is assumed as linear. The estimation of the known parameters is done both by ANN and the classical multiple linear regression analysis and a comparison between them is put forward. In addition to this also a comparison of the proposed methodology is done with the conditions in which some assumptions of the regression analysis are not met.**

**According to the findings of the study the proposed method does not bring any restrictions on the processing of the ANN and it does not require any knowledge about the function underlying the studied problem. Whereas some assumptions of the multiple linear regression are lacking the performance of the**

**parameter estimation decreases. In all conditions the proposed method named “SA in ANN” provides better results regarding the classical approaches. By this way it is aimed to describe the relationship between the input and output variables. Thus it is a step forward taken to eliminate the general criticism and the common “black box” resemblance of ANN.**

**Keywords:** Neural Networks, Backpropagation, Sensitivity Analysis, Levenberg-Marquardt, Conjugate Gradient

## YAPAY SİNİR AĞLARIYLA DUYARLILIK ANALİZLERİ

YEMİN METNİ.....	ii
TUTANAK .....	iii
ÖZET.....	iv
ABSTRACT .....	vi
İÇİNDEKİLER .....	viii
KISALTMALAR .....	xi
TABLO LİSTESİ.....	xii
ŞEKİLLER LİSTESİ .....	xiii
EK LİSTESİ.....	xv
GİRİŞ .....	1

### BİRİNCİ BÖLÜM

#### YAPAY SİNİR AĞLARININ KAVRAMSAL OLARAK İNCELENMESİ

1.1. Yapay Sinir Ağları Nedir? .....	5
1.2. Yapay Sinir Ağlarının Kullanım Alanları .....	11
1.3. Yapay Sinir Ağlarıyla Çalışmanın Nedenleri .....	13
1.4. Tarihsel Gelişim .....	16
1.5. Sinir Ağlarının Faydaları .....	20
1.6. Gerçek ve Yapay Nöronlar.....	24
1.6.1. Gerçek Nöronlara Bir Bakış .....	25
1.6.2. Yapay Nöronlar .....	30
1.7. Gürültü ve Donanım Hatalarına Karşı Esneklik .....	33
1.8. İkili Olmayan Sinyal İletimi .....	36
1.9. Sosyal Bilimler İçerisinde Sinir Ağları Kullanımının Yayılmasında Karşılaşılan Engeller .....	39



## İKİNCİ BÖLÜM

### GERİ YAYILIM ALGORİTMASININ TÜRETİLMESİ

2.1. Nöron Modeli ve Ağ Mimarisi.....	43
2.1.1. Notasyon.....	43
2.1.2. Nöron Modeli .....	44
2.1.2.1. Tek Girdili Nöron.....	44
2.1.2.2. Transfer Fonksiyonları .....	45
2.1.2.3. Çok Girdili Nöron .....	47
2.1.3. Ağ Mimarisi .....	49
2.1.3.1. Tek Katmanlı Nöronlar .....	49
2.1.3.2. Çok Katmanlı Nöronlar.....	51
2.2. Widrow-Hoff Öğrenme Kuralı.....	53
2.2.1. Adaline Ağı .....	54
2.2.2. Ortalama Kare Hata .....	55
2.2.3. LMS Algoritması.....	58
2.3. Geri Yayılımın Oluşturulması.....	61
2.3.1. Giriş.....	61
2.3.2. Çok Katmanlı Ağın Yetenekleri.....	64
2.3.2.1. Desen Sınıflandırma.....	64
2.3.2.2. Fonksiyon Yaklaşımı .....	69
2.3.2.3. Özellik Belirleyici Olarak Gizli Hücreler .....	72
2.3.3. Geri Yayılım Algoritması.....	74
2.3.3.1. Performans İndeksi.....	75
2.3.3.2. Zincir Kuralı.....	76
2.3.3.3. Duyarlıkların Geri Yayılımı .....	78
2.3.3.4. Özet .....	81

## ÜÇÜNCÜ BÖLÜM

### UYGULAMADA GERİYAYILIM ALGORİTMASI VARYASYONLARI VE SİNİR AĞLARINDA DUYARLILIK ANALİZİ UYGULAMASI

3.1. Giriş.....	83
3.2. Temel Geriyayılım Algoritmasının Geliştirilmesi .....	83
3.2.1. Sezgisel Yaklaşımlar .....	85
3.2.1.1. Hızlandırıcılar.....	86
3.2.1.2. Değiştirilebilir Öğrenme Oranı .....	87
3.2.1.3. Dirençli Geriyayılım .....	88
3.2.2. Nümerik Optimizasyon Teknikleri.....	89
3.2.2.1. Eşlenik Gradient.....	92
3.2.2.2. Levenberg-Marquardt Algoritması .....	98
3.3. Sinir Ağlarında Duyarlılık Analizleri .....	107
3.3.1. Literatür .....	108
3.3.2. Girdi Duyarlılığı .....	111
3.3.3. Geri Yayılıma Dayalı Duyarlılık ve Regresyon Analizlerinin Karşılaştırılması.....	114
3.3.3.1. Durum1: Varsayımların Sağlanması.....	119
3.3.3.2. Durum2: Normallik Varsayımının Sağlanmaması.....	123
3.3.3.3. Durum3: Otokorelasyonun Bulunması Hali.....	125
3.3.3.4. Durum4: Bağımsız Değişkenler Arasında Çoklu Doğrusal Bağlantı Olması Hali.....	128
SONUÇ VE DEĞERLENDİRME .....	131
KAYNAKLAR.....	136
EKLER.....	143

## KISALTMA LİSTESİ

TLU	:	Threshold Logic Unit
YSA	:	Yapay Sinir Ağları
Adaline	:	Adaptive Linear Neuron
Madaline	:	Multiple Adaline
RBF	:	Radial Basis Functions
VLSI	:	Very Large Scale Integrated technology
mV	:	milivolt
PSP	:	Postsynaptic Potential
IPSP	:	Inhibitory Postsynaptic Potential
EPSP	:	Excitatory Postsynaptic Potential
LMS	:	Least Mean Square
EKKY	:	En Küçük Kareler Yöntemi
JB	:	Jarque-Bera
GEKKY	:	Genelleştirilmiş En Küçük Kareler Yöntemi

## TABLO LİSTESİ

Tablo 1- 1: (0,1) Ağırlık ve 0.5 Eşik Değeriyle TLU .....	33
Tablo 1- 2: (0.2, 0.8) Ağırlık ve 0.5 Eşik Değerleriyle TLU .....	34
Tablo 1- 3: Girdi Sinyalindeki Bozukluk ile TLU .....	35
Tablo 2- 1: $y_1, y_2$ Çıktıları .....	65
Tablo 2- 2: $y_1, y_2$ Kodları .....	66
Tablo 2- 3: 11 Girdili ve Tek Çıktılı Eğitim Deseni .....	73
Tablo 3- 1: Durum1 için Regresyon Sonuçları .....	120
Tablo 3- 2: Durum2 için YSA Duyarlılık Analizi Sonuçları .....	120
Tablo 3- 3: Durum2 için Regresyon Sonuçları .....	124
Tablo 3- 4: Durum2 için YSA Duyarlılık Analizi Sonuçları .....	125
Tablo 3- 5: Otokorelasyon Mevcutken Regresyon Sonuçları .....	126
Tablo 3- 6: GEKKY Göre Regresyon Sonuçları.....	127
Tablo 3- 7: Durum3 için YSA Duyarlılık Analizi Sonuçları .....	128
Tablo 3- 8: Durum4 için Regresyon Sonuçları .....	129
Tablo 3- 9: Yardımcı Regresyonların $F_i$ Değerleri .....	130
Tablo 3- 10: Durum4 için YSA Duyarlılık Analizi Sonuçları .....	130

## ŞEKİLLER LİSTESİ

Şekil 1-1:Basitleştirilmiş Bir Nöronun Temel Bileşenleri.....	5
Şekil 1-2: Basit Sinir Hücresi.....	6
Şekil 1-3: Sinir Ağının Basit Örneği.....	7
Şekil 1-4: YSA Genel Yapısı.....	10
Şekil 1-5: Biyolojik Nöron.....	26
Şekil 1-6: Kuvvet Gerilimlerinin Yayılımı.....	27
Şekil 1-7: Aktivasyon-Çıktı Eşik Değer İlişkisinin Grafiksəl Şekli.....	31
Şekil 1-8: TLU Hücresi.....	32
Şekil 1-9: Sinirsel Desenler.....	36
Şekil 1-10: Sigmoid Fonksiyonu.....	37
Şekil 1-11: Sigmoid 'in Parçalı Doğrusal Yaklaşımı.....	38
Şekil 1-12: Stokastik Bir Birimdeki Voltaj Akımı.....	39
Şekil 2-1: Tek Girdili Nöron.....	44
Şekil 2-2: Sınırlayıcı Transfer Fonksiyonu.....	45
Şekil 2-3: Doğrusal Transfer Fonksiyonu.....	46
Şekil 2-4: Log-Sigmoid Transfer Fonksiyonu.....	46
Şekil 2-5: Çok Girdili Nöron.....	47
Şekil 2-6: Kısaltılmış Notasyonla R Girdili Nöron.....	48
Şekil 2-7: S Nöronlu Katman.....	49
Şekil 2-8: Kısaltılmış Notasyonla S Nöronlu Katman.....	50
Şekil 2-9: Üç Katmanlı Ağ.....	51
Şekil 2-10: Kısaltılmış Notasyonla Üç Katmanlı Ağ.....	52
Şekil 2-11: Adaline Ağı.....	54
Şekil 2-12: A, B, C, D Sınıflarının Ayrışımı İçin Desen Uzayı.....	64
Şekil 2-13: Dört Sınıflı Sınıflandırma İçin İki Katmanlı Ağ.....	66
Şekil 2-14: Doğrusal Olmayan Ayrılabilir İki Sınıf.....	68
Şekil 2-15: Bağlılık ve Konvekslik.....	68
Şekil 2-16: Desen Uzayında Sonlu Bölge.....	69
Şekil 2-17: Tek Boyutlu Uzayda Fonksiyon Yaklaşımını Örneklendirmek İçin Basit Bir Ağ.....	70

Şekil 2-18: Şekil 2-17 ‘deki Ağ Örneğinin Fonksiyonelliği .....	71
Şekil 2-19: Ağırlıklar ve Özellikler.....	74
Şekil 3-1: Aralık Belirleme .....	96
Şekil 3-2: Aralık Büyüklüğünün Düşürülmesi.....	97
Şekil 3-3: Birim ve Katmanların Gösterimi .....	112
Şekil 3-4: Bir Sınır Ağı Olarak Gösterilen Regresyon Modeli.....	115
Şekil 3-5: Duyarlılık Değerinin Belirlenmesinde Kullanılan Ortalamaların Dağılımları .....	121
Şekil 3-6: Duyarlılık Değerinin Belirlenmesinde Kullanılan Medyanların Dağılımları .....	122
Şekil 3-7: Duyarlılık Değerinin Belirlenmesinde Kullanılan Trimmean (%15) Dağılımları .....	122
Şekil 3-8: Duyarlılık Değerlerinin Standart Sapma Dağılımları.....	123
Şekil 3-9: Regresyon Hatalarının Normal Dağılıp Dağılmadığının Testi.....	124

## EK LİSTESİ

EK1: Terimler Sözlüğü.....	143
EK2: Temel Geri Yayılım Algoritmasının Sayısal Bir Örneği .....	146
EK3: Eşlenik Gradient Algoritmasının Sayısal Bir Örneği.....	149
EK4: Durum2 İçin Duyarlılık Değerlerine İlişkin Grafikler.....	151
EK5: Durum3 İçin Duyarlılık Değerlerine İlişkin Grafikler.....	153
EK6: Durum4 İçin Duyarlılık Değerlerine İlişkin Grafikler.....	155
EK7: Duyarlılık Değerlerini ve Bunlara İlişkin Tanımlayıcı İstatistik Değerlerini Hesaplayan MatlabKodları.....	157
EK8: Yeni Ağırlık Değerlerine Bağlı Olarak Duyarlılıkları ve Tanımlayıcı İstatistik Değerleri Hesaplayan Matlab Kodları.....	157

## GİRİŞ

İnsan beyninin çalışma şeklinin tanımlanması yüzyıllardan beri insanlığın çözmeye çalıştığı ve günümüzde bile sınırlı bir bilginin elde edildiği bir alandır. Son yüzyılda yaşanan bilgi patlamasının altında yatan neden bilgisayarın keşfiyle teknik hesaplamalarda, veri depolanmasında, bilgi üretilmesinde ve paylaşımında hızlı bir ilerlemenin gerçekleşmiş olmasıdır. Bilgisayarın keşfi ise insan beyninin nasıl çalıştığının merak edilmesi ve bunun taklit edilmesine yönelik çalışmalar sonucunda ortaya çıkmıştır. Zamanla bilgisayar alanındaki gelişimin amacı insanların ondan elde edecekleri faydaya bağlanmıştır. Bilgisayarlar, insanların epeyce zaman harcadıkları rutin işleri ve hesaplamaları kısa sürede yerine getirmiştir. Böylelikle bilgisayar alanındaki gelişimin yönü temel amacından sapmış ve insan beyninin modellenmesine yönelik çalışmalarda da tatmin edici bir sonuç alınmadığından düşüş meydana gelmiştir. Az sayıdaki araştırmacı beyin modellenmesi çalışmalarını sürdürmüş ve yapay zeka başlığı altında önemli gelişmelerin ortaya çıkmasına neden olmuşlardır. Yapay sinir ağları, genetik algoritmalar, bulanık mantık ve uzman sistemler bu alanın temel konularını oluşturmaktadır.

Bilgisayarlar sayısal hesaplamalarda insan beyninin yeteneklerine göre oldukça hızlıdır. Bununla birlikte insan beyni bir bilgisayarda bulunması istenen çok sayıda özelliğe sahiptir. İnsan beyni geçmiş deneyimlerine dayanarak yargılarda bulunabilir, yorum yapabilir, daha önce hiç karşılaşmamış olduğu durumlar hakkında fikir yürütebilir. Bu gibi durumlar insan beynini modelleme ve anlama çalışmalarının temelini oluşturur. Beyin ve dokusunun incelenmesi sonucunda beyin hücreleri olan nöronların birbiriyle paralel şekilde çalışarak işlem yaptıkları görülmüştür. Dolayısıyla beyin nöronlar aracılığıyla bilgileri paralel olarak işlemektedir. Bu gerçekten hareketle araştırmacılar biyolojik nöronların yapay karşılıklarını oluşturarak bilenen gerçeklerle onları taklit etme yoluna gitmişlerdir. Yapay sinir ağları temelini bu taklitten alarak beyin çalışmasına benzer paralel hesaplama yeteneğine sahip yöntem, cihaz ve makinelerin geliştirilmesini içermektedir. Biyolojik bakış açısından bir sinir ağına olan gereksinim, gerçek ağların bilgi işlemede sahip olduklarına inanılan temel özelliklerinin elde edilmeye çalışılmasıdır. Fakat bir istatistikçi veya mühendis açısından gerçek ağların temel özelliklerinin elde



edilip edilmemesi önem taşımamaktadır. Bunlar için önemli olan, çözülmek istenen eldeki probleme daha uygun olabilecek alternatif bir paralel hesaplama şeklinin öneriliyor olmasıdır.

Sinir ağları karşılaşılan çok sayıdaki problemde çoğu standart yöntemden daha iyi sonuçlar verdiği gösterilmiş bir araç olma konumundadır. Özellikle eldeki verilerin gürültülü, eksik ve doğrusal olmaması sinir ağları kullanımının etkin olarak uygulandığı durumlardır. Ayrıca sinir ağları veriler hakkında herhangi bir varsayım yapmamaktadır. Örneğin istatistikteki regresyondan farklı olarak girdi değişkenleri arasında karşılıklı etkileşim bulunup bulunmadığına bakmaz. Sinir ağlarını ihmal edilen doğrusal olmama durumunun bir testi olarak da görmek mümkündür. İstatistik çözüm yöntemleriyle karşılaştırmalı olarak beraber kullanıldıkları literatüre geçen çok sayıda çalışma alanı bulunmaktadır. Sosyal bilimler açısından ele alındığında ise tahminleme, sınıflandırma ve kümeleme problemlerinde yoğun olarak kullanıldıkları görülmektedir. Burada çoklu doğrusal regresyon, lojistik regresyon, diskriminant analizi, faktör analizi, kümeleme analizi ve ilgili diğer istatistiksel tekniklere alternatif olma özelliğine sahiptir. Sinir ağları incelenen probleme analitik çözüm bulmanın zor veya var olmadığı durumlara çözümler önerebilmektedir.

İleri beslemeli bir ağın yetenekleri birkaç farklı yolla gösterilmektedir. Ağların ilk olarak uygulama buldukları ortam desen uzayıdır. İki katmana sahip bir ağın (biri gizli olmak üzere) desen uzayında istenilen türde bir ayrışımı (sınıflandırmayı) yerine getirebileceğini göstermek mümkün olabilmektedir. Ağların gösterilebilecek diğer bir yeteneği, sadece tek gizli katmanın kullanılmasıyla keyfi derecede herhangi bir fonksiyona yaklaşabileceğidir. Burada ağ girdilerin bir fonksiyonunu veya haritalanmasını yerine getirmektedir. Son olarak sinir ağları eğitim setindeki desenlerin sınıflandırılması veya tanımlanması için gerekli olan bilgiyi temsil eden özellikleri keşfedebilme yeteneğine sahiptirler.

İncelen probleme ait veri setinin bulunması sinir ağlarının çalışabilmesi için gerekli olan tek koşuldur. Sinir ağları örnekleri temel alarak kullanıcı tarafından kabul edilebilir hata seviyesiyle sonuçlar üretmektedir. Bu işlemi ağ içerisindeki hücrelerin birbiriyle iletişimini sağlayan bağlantı ağırlık değerlerini ardışık olarak

değiştirerek yerine getirmektedir. Ağırlık değerlerinin nasıl değiştirileceği sorusuna kullanılan öğrenme algoritması cevap vermektedir. Şimdiye kadar birçok öğrenme algoritması geliştirilmesine rağmen günümüzde en yaygın olarak kullanılan algoritma geri yayılım algoritmasıdır. Geri yayılım algoritması, ağı tahminleyeceği çıktı değerlerinin gerçek değerlerini önceden bilen bir denetimli/gözetmenli öğrenme türüdür. Geri yayılımında içersine dahil olduğu mevcut öğrenme algoritmaları girdi değerlerini inceleyerek çıktı değerlerini oluştururlar. Fakat girdi ve çıktı değişkenleri arasındaki ilişkinin yönü ve gücü hakkında bilgi vermezler. Hangi girdi değişkeni çıktı üzerinde en etkili sorusuna yanıt bulunamamaktadır. Özellikle sosyal bilimler açısından düşünüldüğünde amaç sadece basit bir tahminleme oluşturmak değil aynı zamanda incelenen olayın açıklanabilmesidir. Yapılacak çalışmayla da duyarlılık analizi adı altında girdinin çıktı üzerindeki etkisinin belirlenmesine çalışılacaktır. Duyarlılık analizi kapsamında önerilecek metod ileri beslemeli bir sinir ağının işleyişi üzerine herhangi bir kısıt getirmemekte ve matematiksel bir temele dayanmaktadır.

Bu çalışmada klasik olarak yapılan regresyon analiziyle sinir ağının bir tahminleme problemindeki hataları minimize etme yetenekleri karşılaştırılmayacaktır. Bunun yerine duyarlılık analizinden elde edilen sonuçların doğruluğunun kanıtlanması amacıyla ana kütle fonksiyonunun doğrusal bir modelden oluştuğu ve parametre değerlerinin bilindiği varsayılacaktır. Böylece bu anakütle parametrelerinin duyarlılık analiziyle tahminlenebileceği gösterilecektir. Modelimiz doğrusal olduğundan ilgilendiğimiz parametre değerleri her bir bağımsız değişkenin önündeki katsayılarıdır. Bilindiği üzere doğrusal modelde bağımsız değişkenin önündeki katsayı o bağımsız değişkenin bağımlı değişken üzerindeki etkisini göstermektedir. Sonuç olarak yapay sinir ağlarında önerilen duyarlılık analiziyle bu katsayılar ne kadar yaklaşılabildiği gösterilmeye çalışılacaktır.

Birinci bölüm içersinde, biyolojik nöronların işleyiş şekli ayrıntılı olarak verilmiş ve hangi özelliklerinin yapay nöronlarda nasıl taklit edilmeye çalışıldığı anlatılmıştır. İlk oluşturulan sinir ağı modelleri tanıtılıp ne gibi eleştirilere maruz kaldıklarına yer verilmiş ve günümüzde kullanılan ağların bu eleştirilerin üstesinden nasıl gelmiş oldukları anlatılmıştır. Sinir ağlarıyla çalışmanın nedenleri sıralanmış ve

kullanım alanları hakkında bilgi verilmiştir. Ayrıca sosyal bilimler içerisinde kullanılırken karşılaşılan sorunlara değinilmiştir.

İkinci bölümde ilk olarak sinir hücresinin ve ağının modellenmesi amacıyla oluşturulan notasyona ve şekilsel gösterimlere yer verilmiştir. Daha sonra geri yayılım algoritmasının temeli olan Widrow-Hoff öğrenme kuralı ayrıntılı olarak tanıtılmıştır. Son olarak Widrow-Hoff öğrenme kuralından geri yayılıma nasıl geçilmiş olduğu anlatılmış ve geri yayılımın ne gibi yeteneklere sahip olduğu gösterilerek temel algoritmanın tanımı yapılmıştır.

Üçüncü bölümde, temel geri yayılım algoritmasını pratik bir probleme uygularken karşılaşılan sorunlar anlatılarak algoritmayı geliştirmede yararlanılan tekniklerden ayrıntılı bir şekilde bahsedilmiştir. Daha sonra girdi ve çıktı değişkenlerini ilişkilendirmek amacıyla kullanılan duyarlılık analizi metodu anlatılmıştır. En sonunda bu metodun doğruluğunun gösterilmesine yönelik olarak varsayımların sağlanıp ve sağlanmadığı durumlarda çoklu doğrusal regresyonla karşılaştırılması yapılmış ve elde edilen sonuçlara yer verilmiştir.

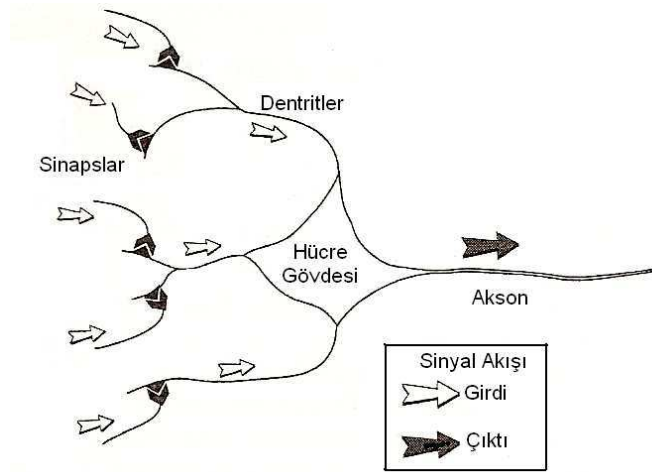
## 1. BÖLÜM

### YAPAY SINİR AĞLARININ KAVRAMSAL OLARAK İNCELENMESİ

#### 1.1. Yapay Sinir Ağları Nedir?

Yapay sinir ağlarının ne olduğunu anlamak için öncelikle basit bir tanım verilip daha sonra bu tanım içerisindeki önemli noktaların açıklanması üzerinde durulacaktır.

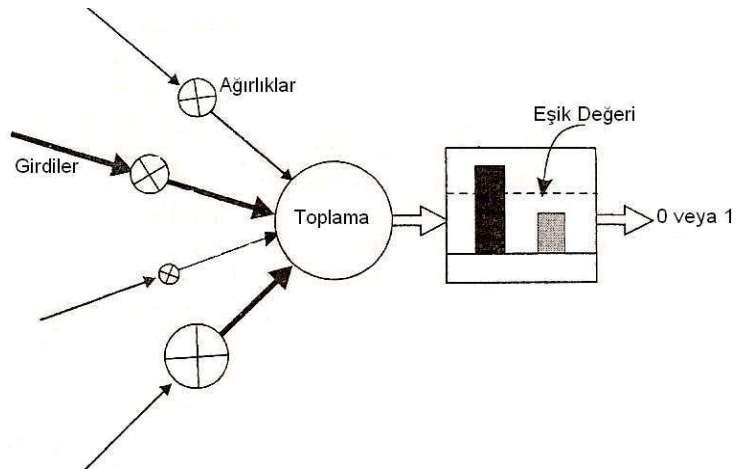
Sinir ağı, fonksiyonelliğini nöronlara dayandıran basit işlem elemanları olan birimlerin veya düğümlerin karşılıklı olarak bağlanarak bir araya gelmesidir. Ağın işlem yeteneği, “bir eğitim deseni setine uyarlanma” veya başka bir deyişle “öğrenme” süreciyle elde edilen karşılıklı birim bağlantısının gücünde veya ağırlığında saklanır (Gurney, 1997:1).



Şekil 1- 1:Basitleştirilmiş Bir Nöronun Temel Bileşenleri

İnsan beyni tahminen  $10^{11}$  (100 milyar) sinir hücresi veya nöronlardan oluşmuş ve bunun önemli derecede basitleştirilmiş bir örneği Şekil 1-1 de gösterilmiştir. Nöronlar hücre duvarı veya zarındaki voltaj tarafından beliren kısa süreli elektriksel sinyaller aracılığıyla iletişim kurarlar. Karşılıklı nöron bağlantıları, dentrit olarak adlandırılan hücrenin kolları üzerinde yerleşmiş elektrokimyasal

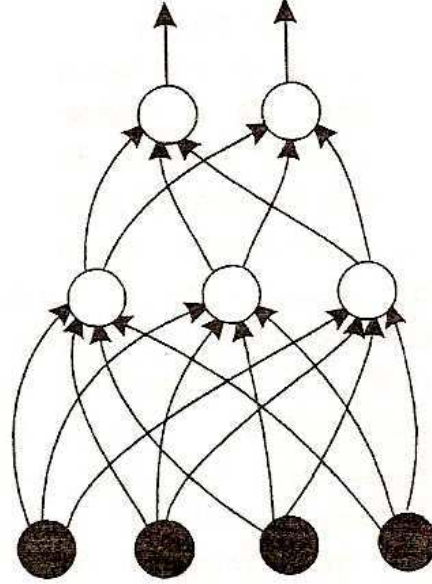
kavşaklar başka bir deyişle sinapslar tarafından yürütülmektedir. Her nöron genellikle diğer nöronlarla binlerce bağlantı kurmakta ve böylece hücre sürekli olarak gövdesine ulaşan çok sayıda girdi sinyali almaktadır. Hücre gövdesinde girdi sinyalleri bir şekilde toplanmakta veya birleştirilmektedir. Elde edilen bu değer belirli bir eşik değerini aşarsa hücre yanmakta veya cevap olarak bir voltaj üretimi oluşturmaktadır. Bu akson olarak bilinen bir lif yardımı aracılığıyla diğer nöronlara gönderilmektedir. Bir voltaj üretiminin üretilip üretilmeyeceğinin belirlenirken bazı girdi sinyalleri engelleyici etkiye neden olmakta ve yanmanın gerçekleşmesini engellemektedirler, diğerleri ise tetikleyici olmakta voltaj üretiminin oluşmasına katkıda bulunmaktadırlar.



**Şekil 1-2: Basit Sinir Hücresi**

Biyolojik nöronların yapay karşılıkları ön tanımımızda belirttiğimiz üzere düğümler veya birimlerdir ve tipik bir örneği Şekil 1-2 de gösterilmiştir. Sinapslar tek bir sayı veya ağırlık olarak modellenmiştir böylece her girdi hücre gövdesine karşılık gelen kısma gönderilmeden önce bir ağırlık değeriyle çarpılır. Burada ağırlıklandırılmış sinyaller bir düğümün veya birimin aktivasyon değerinin bulunması için basit aritmetik toplama işlemi aracılığıyla birleştirilir. Şekil 1-2 deki gibi gösterilen türdeki birime eşik değer birimi (Threshold Logic Unit (TLU)) denmektedir ve aktivasyonu bir eşik değeriyle karşılaştırılmakta, eğer aktivasyonu eşik değerini geçerse birim bir çıktı değeri üretmekte (geleneksel olarak 1) öteki durumda ise sıfır değerini çıktı olarak vermektedir. Şekilde sinyallerin büyüklüğü onlara karşılık gelen okların genişliği ile temsil edilmiştir ve ağırlıklar dairelerdeki

çarpım sembolü ile gösterilmiş, değerlerinin sembol büyüklüğü ile orantılı olduğu varsayılmış ayrıca yalnızca pozitif ağırlıklar kullanılmıştır. TLU modeli en basit ve tarihsel olarak da en eski yapay hücre modelidir.



**Şekil 1-3: Sinir Ağının Basit Örneği**

Bu çalışmada ağ terimi yapay nöronların herhangi bir sistemi hakkında bahsedilirken kullanılacaktır. Bu tek bir düğüm gibi basit bir yapıdan ağ içerisindeki her düğümün diğer bir düğüme bağlandığı ve içerisinde çok sayıda düğümün mevcut olduğu daha karmaşık ve büyük yapılara kadar değişebilmektedir. Şekil 1-3 de bir çeşit ağ türü gösterilmiştir. Her düğüm tek bir daire ile gösterilmiş ve tüm bağlantılardaki ağırlık değerleri açık bir şekilde gösterilmemiştir. Düğümler bir girdiden gelen her sinyalin bir çıktıya ulaşmadan önce iki düğümden geçtiği katmanlandırılmış bir yapı içerisinde organize edilmiştir. Bu ileri beslemeli yapı var olan yapılardan sadece birisidir ve genellikle bir girdi deseninin, çıktı katmanındaki değerler göz önünde tutularak bunlardan birine yerleştirilmesi amacıyla kullanılmaktadır. Örneğin eğer girdi bir el yazısı görüntüsündeki açık ve koyu desenlerin kodlanmasından oluşuyorsa çıktı katmanı, girdi karakterinin hangi harf sınıfına karşılık geldiğinin gösterilmesi amacıyla alfabedeki her harfe bir düğüm gelecek şekilde 29 düğüm içerebilir. Bu durum her sınıf için tek bir çıktı düğümünün

tahsis edilmesi ve her ne zaman bir sınıf deseni girdi olarak verilirse sadece tek bir çıktı düğümünün aktif olması gerekliliği ile yerine getirilir.

Hücresinin temel yapı elemanları ve bunların işlevleri hakkında daha ayrıntılı bilgi için öncelikle bu kısmın başında verilen tanım incelendiğinde deneyimden öğrenme vurgusuna dikkat etmek gerekir. Gerçek nöronlarda sinaptik bağlantıların gücü belirli koşullar altında değiştirilir böylece her nöronun davranışı değişebilir veya belli bir uyarıcı girdisine uyarlanabilir. Yapay nöronlarda bu durumun eşdeğeri ağırlık değerlerinin değiştirilmesidir. Bilgi işlem diliyle anlatılırsa ortada bilgisayar programları yoktur, ağırlık sahibi olduğu bilginin bir desen setindeki örnek uyarıcılara uyarlanma süreciyle değişen ağırlık değerlerinde saklandığı varsayılmaktadır. Ağa sunulan bir girdi desenine karşılık gelen cevabın bir hedef çıktı değeriyle karşılaştırıldığı Şekil 1-3 deki ağ çeşidiyle beraber kullanılan bir öğrenme türü gözetmenli öğrenmedir. El yazısı örneğini ele alalım, A harfinin girdi olarak ağa sunulması durumunda ağırlık çıktısı değeri A 'nın sınıf kod değeriyle karşılaştırılır. İki desen arasındaki farklılık ağırlıkların nasıl değiştirileceğini belirler. Ağırlıkların değiştirilmesine ilişkin bilgi öğrenme kuralını oluşturmaktadır. Arzu edilen ağırlık güncellemesi yerine getirildikten sonra diğer desen ağa sunulur, çıktı hedef ile karşılaştırılır ve yeni değişiklikler yapılır. Bu ardışık olaylar dizisi ağırlık davranışının her bir desene cevabı ona karşılık gelen hedef değerine yakın oluncaya kadar devamlı olarak çok kez tekrarlanır. Desen sunumlarının sırası, öğrenmenin durdurulma kriteri vb. ini içeren sürecin tamamı öğrenme algoritmasını oluşturmaktadır.

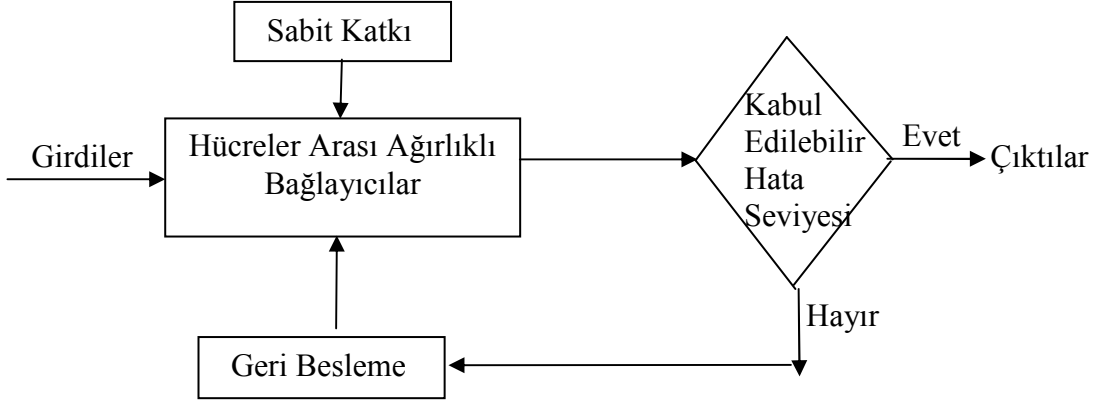
Öğrenme tamamlandıktan sonra merak edilen konu, ağa daha önceden gösterilmeyen bir desen sunulduğunda ağırlık vereceği cevaptır. Eğer ağ problem alanının altında yatan yapıyı öğrenmişse bu takdirde görülmemiş desenleri doğru olarak sınıflandıracak ve ağırlık genellemenin iyi olduğu söylenecektir. Ağ bu özelliğe sahip değilse eğitim setini ezberlemekten öteye gidemez ve pratik kullanımında oldukça yetersiz kalır. Bu yüzden iyi bir genelleme yeteneği sinir ağlarının önemli özelliklerinden birini oluşturmaktadır.

Bilgi işleme süreci olarak tanımlanabilen yapay sinir ağları (YSA), verilen girdilere karşı çıktılar üreten ayrıntılı bir kata kutu modeli olarak da tanımlanmaktadır. YSA' nın sahip olduğu özellikler klasik bilgi işleme yöntemlerinden farklı olmasını sağlamaktadır. Sahip olduğu bazı özellikler dikkate alındığında birçok yöneme göre daha iyi sonuçlar verdiği görülmektedir.

YSA bilgileri paralel olarak işlemektedir. Bilgilerin paralel olarak işlenmesi ağ içerisindeki her birimin birbirinden bağımsız olması demektir. Bu birbirinden bağımsız şekilde çalışan birimlerin herhangi birinde meydana gelen hata tüm sistemde belirgin bir hataya neden olmamaktadır. Birimlerin veya hücrelerin sistem içerisindeki ağırlıkları oranında bir etki ortaya çıkmaktadır. Bunun sonucu olarak bir birimdeki hata tüm sistemin hatalı sonuç vermesine veya sistemin durmasına neden olmamakta, genel sistem yerel hatalardan en az şekilde etkilenmektedir. Ayrıca paralel çalışan YSA modelinin karmaşık olmayan yapısı ve karışık fonksiyonlarla yapılan işlemler yerine basit işlemler içermesi birçok sorunun çözümünde tercih edilmesine neden olmuştur. YSA öğrenme sonucu elde edilen bilgilerin bağlantı ağırlıklarında saklanmasıyla bilginin uzun süre saklanmasına imkan vermektedir.

Bir YSA için ağdaki bağlantı (ağırlık) değerleri, eldeki verilere göre beklenen çıktıları girdilerden elde etmek amacıyla iteratif yöntemler kullanılarak belirlenir. Şekil 1-4 de bir YSA 'nın genel olarak nasıl çalıştığı gösterilmiştir. Ağın çıktıları beklenen çıktılar ile karşılaştırıldıktan sonra eğer kullanıcı tarafından kabul edilebilir hata sınırı içerisinde yaklaşıklık gösteriyorsa YSA 'nın eğitilmesine son verilir. Aksi bir durumda eğitime yani bağlantı değerlerinin değiştirilmesine devam edilir. Her eğitimin bir öncekini daha iyileştirecek biçimde yinelemeli olarak yapılması gerekmektedir.





**Şekil 1- 4: YSA Genel Yapısı**

*Kaynak: Şen, 2004, s.11*

YSA, mimari yapı ve bu mimari yapının çalışmasını sağlayan matematik fonksiyonlar olmak üzere iki kısma ayrılarak incelenebilir. Genel olarak giriş, saklı (ara) ve çıkış tabakalarındaki sinir hücreleri ile bunlar arasındaki bağlantılar ve sabit katkısı olan bir hücre mimari yapıyı oluşturmaktadır. Saklı hücreler içinde de işlemci (aktivasyon veya transfer fonksiyonu) denen bir iç işleyiş bulunmaktadır. YSA işleyişi, iç ve dış işleyişler diye iki matematik işlev olarak düşünülebilir. İç işleyiş saklı tabakadaki işlemciler vasıtasıyla sağlanır. Dış işleyiş ise ardışık katman hücreleri arasındaki bağlantı değerlerinin öncelikle rasgele küçük sayılara atanması sonrada çıktı tahmin hatasının geri besleme vb. işlemlerle en küçüklenerek yenilenmesi sonucunda sağlanır. YSA' nın öğrenme, eğitime, hatırlama ve sürekli yeni bilgiler edinme ile ağ bağlantılarının yenilenmesi görevini bu matematik işlemler yerine getirir.

YSA eldeki bilgileri zaman serileri ve istatistik veri işlem yöntemlerinde olduğu gibi sıra ile değil, tüm girdilerin birbirine paralel olarak önce basit bileşenlere ayrılması ve sonra da bunları birleştirerek istenen çıktıları yine paralel olarak elde etmeye yarar. YSA mimarisini gelen ışığı bileşenlerine ayırabilen bir prizmaya benzetebiliriz. Ağ mimarisi ve matematiği ile gelen verileri değişik basitlikte kısımlara ayırdıktan sonra bunları istenen çıktılarına uygun olacak biçimde yeniden birleştirmeyi yapar (Şen, 2004:12).

## 1.2. Yapay Sinir Ağlarının Kullanım Alanları

Kişinin amacına bağlı olarak sinir ağı çalışmaları farklı bakış açılarından ele alınabilmektedir. Aynı zamanda hangi sorunun cevaplanmaya çalışıldığının bilinmesi de bu konuda yardımcı olacaktır ve böylece sinir ağları denildiğinde ortaya çıkabilecek karışıklığın önüne geçilmiş olunacaktır.

Sinir ağları istatistiksel analiz ve veri modellemede sıklıkla kullanılmaktadır. Burada sinir ağları kümeleme analiz teknikleri veya standart doğrusal olmayan regresyonun bir alternatifi olarak görülmektedir (Choudhury, 2006:84; Tagliaferri, 2001:69; Malakooti, 2000:435). Sinir ağları, çok sayıdaki girdinin az sayıdaki çıktı sınıfına haritalanmasını içeren sınıflandırma problemleri için kullanılabilir (Curam, 1994:440; Ripley, 1994:409; Wang, 1995:555) . Benzer şekilde eksik veya bozulmuş girdilerin doğru sınıfının tanımlandığı tamamlama problemlerinin de üstesinden gelmektedir (Gupta, 1996:229; Wang, 2005:2583; Brouwer, 2003:23). Böylece sinir ağları faktör analizi ve diskriminant fonksiyon analizine bir alternatif olmaktadır. Sinir ağları özellikle yapısal eşitliklerle kolay bir şekilde açıklanamayan birbiriyle doğrusal olmayan tarzda ilişkili, çok sayıda girdinin mevcut olduğu durumlardaki tahminleme problemlerinde oldukça başarılıdır (Poli, 1994:117; Dunis, 2002:317; Chen, 2005:403). Tahminlemede, sinir ağları çok değişkenli genel doğrusal hipotez modellerine, yapısal eşitlik modellerine ve uzman sistemlere bir alternatiftir (Adams, 2005:861; Lam, 2004:567; Kurfess, 2000:7). Ayrıca sinir ağları, planlama gibi birçok parçadan oluşmuş faaliyetleri içeren kısıt doyurma problemlerini ve karmaşık talep kısıtlarını karşılamaya çalışan kaynak tahsis problemlerini çözebilmekte böylece doğrusal programlama ve dinamik sistem analizlerine bir alternatif olabilmektedir (Talavan, 2006:416; Iwataa, 2006:62; Jain, 1998:1249). Genelde ele alınan problem karmaşık ve yapısal olarak ortaya konamıyorsa, eksik veriler varsa, çok sayıda birbiriyle rekabet halde bulunan girdiler ve yapısal eşitliklerle formülasyonu önleyen karışık doğrusal olmayan yapıda ilişkili kısıtlar bulunuyorsa, sinir ağları tarafından üretilen yaklaşık çözümleri araştırmacının kabul etmesi koşuluyla, sinir ağları klasik istatistiksel yöntemlerden daha iyi sonuçlar ortaya koyabilir (Yoon, 1993:51; West, 1997:370; de Carvalho, 1998:717). Bazı örnekler görüntü, ses, metinsel karakter tanımlanması ve sağlık teşhisi, jeolojik

petrol arama, finansal pazar göstergesi tahmini gibi insan uzmanlık alanlarını içermektedir (Kimura, 2005:97; Courrieu, 2006:429; Samanta, 2006:175). İnsan uzmanlık alanları aynı zamanda klasik yapay zeka içersine de girmektedir böylece mühendisler ve bilgisayar bilimciler sinir ağlarını makine zekasında egemen olan geleneksel algoritmik tekniklere bir alternatif sağlayan paralel dağıtılmış hesaplamanın bir türü olarak görmektedirler (Zia, 2000:196; Qazi, 2006:595; Hakem, 2008:1572). Paralellik her hücrenin veya düğümün diğer hücre veya düğümlerle eşzamanlı ve bağımsız olarak işleme dahil olması gerçeğine dayanmaktadır. Ağ içersindeki bilgi, klasik bir bilgisayarda olduğu gibi birkaç hafıza konumuna yerleştirilmektense tüm ağırlık seti üzerine dağıtılmıştır. Bu alanın uygulamacıları biyolojiks gerçeklik ile kendilerini ilgili görmemekte ve dijital donanım problemlerinin çözümünün kolaylaşması veya belirli tekniklerin doğruluğu ve verimliliğine motive olmaktadır.

Nörobilimci ve psikologlar gerçek sinir dokusunun bilgi işleme için gerekli olduğuna inanılan özelliklerini özetleyerek geliştirilen, hayvan beyninin hesaplama modelleri olarak sinir ağları ile ilgilenirler (Sakumura, 2006:469; Segal, 2006:298; Galizia, 2008:81). Oluşturulan yapay nöronlar genellikle biyolojik nöronların oldukça basitleştirilmiş şeklidir ve çoğu nörobilimci beyin fonksiyonelliğini açıklamak için daha fazla ayrıntıya gerek duyulduğu konusunda ısrar ederek, bu zayıflatılmış modellerin nihai gücü hakkında kuşkuya düşmektedirler.

Son olarak fizikçi ve matematikçiler doğrusal olmayan dinamik sistemler ve istatistiksel mekaniklere olan ilgiden ötürü sinir ağları çalışma alanı içersinde kendilerine yer bulurlar (Schindler, 2003:179; Lenze, 1999:303; He, 1998:131). Hatırlatmak gerekirse uygulamalı matematiğin işi bilimin diğer alanlarında önceden kullanılan araçları kullanarak yeni sistemin özelliklerini keşfetmek ve formülize etmektir. Örneğin özel bir tür ağ ile (Hopfield) manyetik sistemler arasında yüksek derecede bir bağlantı bulunmaktadır.

Bütün bu farklı gruptaki bilimciler farklı sorular sormaktadırlar: Nörobilimci hayvan beyninin nasıl çalıştığını bilmek ister, mühendisler ve bilgisayar bilimciler zeki makineler inşa etmek ister ve matematikçiler ise kompleks sistemler

olarak sinir ağlarının temel özelliklerini anlamak isterler. İş ve endüstri dünyasının çeşitli alanlarında bulunan ve belki daha büyük olan diğer bir grup insan ise sinir ağlarını çalışma alanlarında doğal olarak ortaya çıkan büyük ve anlaşılması zor veri setlerini analiz ve modellemek için kullanırlar.

### 1.3. Yapay Sinir Ağlarıyla Çalışmanın Nedenleri

YSA çalışmalarının temeli 1940 'lı yıllarda atılmış olsa da bunların etkin kullanımına ve sürekli gelişimine 1980 'li yıllarda başlanmıştır (Freeman ve Skapura, 1991:12). Yapay zeka başlığı altında toplanan YSA, bulanık mantık ve genetik algoritma yöntemleri araştırma, endüstri ve teknolojik gelişmelerde oldukça başarılı bir şekilde uygulanmaktadır. YSA uygulamalara kısa zamanda uyarlanabilen bir anlayış basitliğine ve işleyişine sahiptir. YSA uygulamalarının başarı ile yapılması için esaslarının ve çalışma işleyişlerinin çok iyi anlaşılması gereklidir. Böyle olmayan durumlarda yapılan çalışmaların çoğu sadece gerçek hayattan uzak birer uygulamayı geçmemektedir. Aşağıda her alanda kullanım yeri bulabilen sinir ağlarına bu çekiciliği kazandıran genel kullanım nedenleri verilmiştir.

- 1. Sınıflandırma:** Bilimsel çalışmalarda günlük sınıflandırma alışkanlığı devam eder ve incelenen olaylar kendi içinde örtüşmeyecek biçimde sınıflara ayrılarak yorumlanır. Örneğin yılda kişi başına geliri 30000\$' dan fazla olanlar zengin, 5000 ile 30000\$ arasında olanlar orta halli, 5000 ile 1000\$ arasında olanlar fakir ve yıllık geliri 1000\$ 'dan düşük olanlar ise açlık sınırında diye bir sınıflamayı YSA' nı eğiterek öğrettikten sonra bunlara verilen verilerle hangi sınıfın geçerli olacağını tespit etmesi talebinde bulunabiliriz.
- 2. Kümeleme:** Sınıflandırmada sınıf aralıkları önceden belirlenerek verilmesine karşın kümelemede böyle bir bilgi yoktur. Kümeleme birbirine benzer olan desenlerin (girdi verilerinin) aynı grubu oluşturması demektir. YSA o şekilde eğitilebilir ki kendisine verilen bir dizi desen (veri, vektör) ile önce bunları birbirinden farklı alt kümelere ayırabilir. Daha sonra kendisine verilen ve eğitim verileri arasında olmayan desenlerin hangi kümeye

gideceğine karar verebilir. Burada YSA 'nın çıktı tabakasındaki her hücre değişik kümeyi temsil edecek biçimde oluşturulmuştur. Kümeleme için benzerlikte genelde iki veri dizisi arasındaki Öklid mesafesine veya bu iki veriyi temsil eden vektörler arasındaki açığa ve dolayısıyla da istatistikteki korelasyon kavramına bakılmaktadır.

- 3. Vektör sayısallaştırılması:** Burada çok sayıda olan verilerin kendi aralarında az sayıda aynı özelliklere sahip olan verilere ayrılması işlemi önem kazanmaktadır. Klasik çalışmalarda bununla ilgili kabul veya pratik işlemler söz konusudur. Örneğin sonlu elemanlar ile diferansiyel denklemlerin modellenerek çözülmesinde her sonlu elemanın geçerli olduğu uzaydaki özelliklerinin bu uzaydaki tüm noktalarda aynı olduğu kabulü yapılmaktadır. Bunun sonucunda milyonlarca noktanın özelliği tek bir sayı ile temsil edildiğinden veri sayısı önemli miktarda azaltılır. Sayısallaştırma işlemi ile örneğin bir alan örtüşmeyen alt alanlara ayrılabilir. Avrupa ülkeleri haritasına bakılınca her ülkenin kendi hudutları ile sınırlanmış alanlarının olduğu görülür. İşte bunlar birer vektör sayısallaştırma ile alt alanlara ayrıştırma işlemidir. Bu tür işlemlerin insan aklı ile düşünüldüğüne benzer olarak YSA 'na yaptırılması mümkündür. Bunun için özel yöntem ve algoritmalar geliştirilmiştir.
- 4. Desen uygunluğu:** Burada bozuk, yıpranmış, eskimiş veya eksik olarak verilen bir desenin YSA 'nın hafızasında daha önceden depolanmış olan tam ve eksiksiz bir desenle karşılaştırılması sonucunda bozuk girdiye YSA düzgün desen çıktıları verebilir. İşte bu işleme desen araştırma ve karşılaştırma işlemi adı verilmektedir. Bu aynen gürültülü olan bir zaman serisinden örneğin Fourier çözümlenmeleri ile düzgün bir eğrinin çıkarılmasına benzetilebilir. Daireye çok yakın olan şekillerin idealleştirilerek daire kabul edilmesi bu tür bir insan düşüncesidir.
- 5. Fonksiyon yaklaşımı:** Matematik ifadesi bilinmeyen birçok şekil bulunmaktadır. Bu durumda istenilen şeklin basit ve düzgün olan başka şekillerin üst üste bindirilmesi ile yaklaşımı söz konusu olabilmektedir.

Verilen bir fonksiyona temel bazı şekillerle yaklaşmak YSA yoluyla mümkün olmaktadır.

6. **Tahmin yapmak:** Genel olarak gelecek, geçmişteki verilerin davranışlarının incelenmesi sonucunda olayın iç ve dış işleyişlerinin anlaşılmasından sonra tahmin edilebilir. Borsadaki salınımların, hava sıcaklığının, sellerin vb. tahminleri hep bu tür bir işlemdir. Mükemmel tahminlerin yapılabilmesi mümkün olmamakla birlikte yapay sinir ağları modelleri ile eldeki verilerden en iyi biçimde tahmin yapılması mümkündür. Aslında tahmin önceki adımda söylenen fonksiyon yaklaşımına benzemektedir.
7. **Kontrol sorunları:** Endüstride imal edilen birçok alet ve cihazın çalışması için matematik modellerle benzetilemeyecek kadar karmaşık iç ilişkiler vardır. Bu ilişkilerin insan tarafından modellenmesine otomatik kontrol veya otomasyon denir. Kontrol genel olarak verilen girdi değerlerinden istenen çıktı davranışının elde edilmesi olarak tanımlanabilir. Bunun için değişik çalışmalarda YSA modellemeleri başarı ile kullanılmaktadır. Zamanla olan değişiklikleri de içeren yinelemeli kontrollerin de yapılması mümkündür.
8. **En iyileme (Optimizasyon):** Birçok ticari ve bilimsel konularda incelenen olayın verilen kısıtlar altında hedefinin en büyüklenmesi (veya en küçüklenmesi) en iyileme işlemi olarak bilinir. En iyileme işlemi için önceki çalışmalarda klasik birçok yöntem geliştirilmiş olmasına karşılık bunun YSA modellemesi ile yapılması en azından sınırlayıcı matematik kabullerin bulunmaması açısından yararlıdır.
9. **Arama çalışmaları:** Özellikle yapay zeka uygulamalarında fazlaca kullanılan arama yöntemlerinin yapay sinir ağlarla başarılması da mümkündür. Bundan önce söylenen noktaların hemen hepsi birer arama işlemidir (Şen, 2004:12).

#### 1.4. Tarihsel Gelişim

Sinir ağlarında modern çağ McCulloch ve Pitts (1943) 'in öncü çalışmasıyla başlamıştır. McCulloch sinir sisteminde gerçekleşen bir olayın temsil edilmesi hakkında yaklaşık 20 yıl düşünerek geçirmiş psikiyatrist ve nöroanatomist eğitimi görmüş bir nörobiyolojistir. Pitts ise bir istatistik ve matematik dehası olup 1942 'de McCulloch ile bir araya gelmişlerdir ve beyindeki nöronların benzerlerinin kullanılmasıyla beyin fonksiyonelliğinin modellenmesinin matematiksel olarak temsil edilebileceğini öne sürmüşlerdir. 1930 'ların sonu 1940 'ların başında Chicago üniversitesindeki çalışmada McCulloch ve Pitts ikili değerler alan yapay nöronların bir ağıyla hesaplamaların yerine getirilebileceğini gösterdiler.

McCulloch-Pitts 'in biyolojik nöronlardan esinlendiği nokta, yapay nöron olarak tasarlanmış bir işlemci biriminin girdileri belirli bir eşik değerine ulaştığında o birimin bir sinyal gönderecek şekilde programlanabileceğidir. Biyolojik bir sistemde nöronları bağlayan dentritler programlanan bağlantılar aracılığıyla taklit edilebilir. Girdi ve nöronlar arasındaki bağlantılara bazılarının büyük sinyal geçişine izin verdiği ve bazılarının ise sadece güçsüz bir sinyale izin verdiği yani sinyalin iletim gücüne dayalı olarak farklılaşan biyolojik yapı olan sinapsleri taklit etmek amacıyla ağırlık değerleri atanabilir. Ayrıca biyolojik fenomen sinapslerin bazılarının dentritten alınan sinyallere ekleme yaparak tetikleme ve bazılarının ise alınan sinyali azaltarak engellemesi bağlantıların pozitif ve negatif olarak programlanmasıyla benzetilebilir.

Sinir ağlarında sonraki büyük gelişme savaş sonrası 1949 'da Donald O. Hebb 'in yazdığı Davranış Organizasyonu (Organization of Behavior) kitabıyla ortaya çıkmıştır. Hebb algılayıcı bir ağ içerisinde öğrenmenin nasıl gerçekleştiğini mantıksal olarak açıklamaya çalışan Hebb kuralı olarak da bilinen geriyayılım kuralının atası olan biyolojik bir modeli sunmuştur. Bu kural kısa dönemli sinirsel hafızanın, iki nöron eş zamanlı olarak tetiklendiğinde (excitatory) bunlar arasındaki bağlantının öğrenmeyi temsil edecek şekilde artırılarak düzenlenmesiyle taklit edilebileceği düşüncesinin önünü açmıştır. Sabitlenmiş ağırlık değerlerinin kullanıldığı McCulloch ve Pitts modelinden farklı olarak ilk kez resmi olarak

sinaptik deęişim dűşüncesini ortaya atmıştır. Hebb ayrıca uzun dönemli sinirsel hafızanın taklit edilebilmesi için sadece bağlantı aęırlıklarında düzenlemenin yeterli olmadığı aynı zamanda yapay sinir aęlarında yapısal olarak da deęişime gidilmesi gerektiğine inanmıştır. Hebb ‘in dűşüncesi pratik kullanım için çok basit olmasına rağmen gelecekteki tüm sinir aęı algoritmalarının üzerine inşa edildięi bir temel nitelięi taşımıştır.

Marvin Minsky, John McCarthy, Nathaniel Rochester ve Claude Shannon ‘ın organize ettięi 1956 ‘daki yapay zekaya yönelik ilk uluslararası konferansta sinir aęlarına olan ilgide artış olmuştur. Konferansın ana dűşüncesi her türlü öğrenme veya dięer zeka özellikleri hakkında temel varsayımların geliştirilmesi ve ilke olarak doğru bir şekilde tanımlanmasıyla bir makinenin bu özellikleri simülasyon edebilmesine dayanmaktaydı. IBM araştırmacısı Rochester, Hebb ‘in öne sürdüęü öğrenmeyi temel alan formüle edilmiş sinirsel teoriyi test etmek için bilgisayar simülasyonunun kullanıldığı ilk sinir aęını konferansta sunmuştur. Rochester ‘ın simülasyonu teorisinin gerçekte çalışabilmesi için tetiklemenin yanı sıra işin içerisine engellemenin (inhibitory) katılması gerektiğini göstermiştir. Aynı yıl içerisinde Uttley yaptığı çalışmayla ikili desen sınıflandırılmasının, desenler sunulurken alınan cevaba göre sinaptik bağlantıların deęiştirilmesiyle bir sinir aęı tarafından başarılabileceğini göstermiştir.

Minsky, 1954 yılında Princeton üniversitesinde sinir aęları konusunda doktora tezini yazmıştır. 1961 de yapay zeka alanında öncü bir çalışma olan “Yapay Zekaya Yönelik Adımlar” (“Steps Toward Artificial Intelligence”) yayımlamış ve bu çalışmanın büyük bölümünü sinir aęlarına ayırmıştır (Haykin, 1994:39).

McCulloch ve Pitts ‘in klasik çalışmasının yayımlanmasından yaklaşık 15 yıl sonra desen tanımlama problemine yeni bir yaklaşım Rosenblatt (1958) tarafından algılayıcı (perceptron) çalışmasında sunulmuştur. Rosenblatt çalışmasında öğrenme sürecinde rassal olarak seçilmiş nöronların kullanılması fikrini de içeren çeşitli türde sinirsel öğrenme algoritmaları geliştirmiştir. Rosenblatt sinirsel modelinin, basit şekildeki desenleri sınıflandırmayı öğrenebildiğini ve bilginin aęın her yerine dağıtılmış olmasından ötürü gürültülü veri ile karşılaştığında veya bazı bileşenlerinde



bozukluk meydana geldiğinde hala fonksiyonelliğini yerine getirebildiğini göstermiştir. Rosenblatt 'ın çalışmasındaki heyecan ve önem bir sinirsel bilgisayar sisteminin, örneğin istatistik paketlerinde olduğu gibi geleneksel baştan sona doğru algoritmik olarak kodlanmış komutlar yerine, örneklerden tümevarım aracılığıyla girdilerin sınıflandırılmasının öğrenebileceğini göstermiş olmasıdır. Bununla beraber algılayıcı yapısı itibariyle mutlaka bir deseni bir sınıfın içersine sokmaya çalışmaktadır yani hata değerini sıfır yapmaya çalışmaktadır; bu da kompleks desenlerin sınıflandırılmasında problemler çıkarmıştır. 1960 'da Widrow ve Hoff en küçük ortalama kare algoritmasını öne sürmüş ve bunu tek katmanlı sinir ağı olan Adaline (Adaptive Linear Neuron) üzerinde uygulamışlardır. Adaline ve algılayıcı arasındaki farklılık eğitim prosedürüne dayanmaktadır. Adaline hatayı en küçük düzeyde tutmaya çalışırken algılayıcı hatayı sıfır yapmaya çalışmakta aksi takdirde öğrenememektedir bu bakımdan Adaline hata durumunda da öğrenmeye devam etmekte ve algılayıcıya üstünlük sağlamaktadır. Widrow-Hoff öğrenme kuralı olarak da bilinen kural, şu anda en yaygın olarak kullanılan sinirsel mimari olan geri yayılım algoritmasının temelini oluşturmaktadır ve geri yayılım algoritması Widrow-Hoff öğrenmenin genelleştirilmiş şekli olmuştur. Widrow daha sonra Madaline (Multiple Adaline) olarak adlandırılan Adaline 'nın iki katmanlı versiyonunu oluşturmuştur. Widrow 1962 yılında sinir ağın gerçek bir dünya problemine ilk uygulaması olan meteoroloji tahminini gerçekleştirmiştir.

1960 'lardaki algılayıcının klasik periyodu süresince, sanki sinir ağları her şeyi yapabilecekmiş gibi görünüyordu. Fakat daha sonra Minsky ve Papert 'ın algılayıcının hesaplama yeteneğindeki temel sınırlamaları matematiksel olarak gösteren kitapları (1969) ortaya çıkmıştır. Kitaplarında algılayıcının yalnız girdi ve çıktı katmanlarına sahip olmasıyla yeterli karmaşıklaktan uzak olduğunu, bu tür modellerin sadece doğrusal olarak ayrılabilir problemleri çözme yeteneğinin olduğunu ve doğrusal olmayan şekilde ilişkilendirilmiş sınıfları ayırmada kullanılamayacağını öne sürerek algılayıcı modelinin yetersizliğini ortaya koymuşlardır. Minsky ve Papert tarafından algılayıcı uygulamalarına yapılan bu güçlü eleştiri sonrası 1970 'lerde sinir ağlarına yönelik araştırmalarda çok keskin bir düşüş dönemine girilmiştir. Psikoloji ve nörobilimciler dışında çoğu araştırmacı,

finansal olarak arařtırmalarına olan desteęinde kesilmesiyle beraber yaklaşık 10 yıl süren bu dönemde alıřma alanını terk etmişlerdir. Bununla beraber 1988 'de yeniden basılan kitaplarında Minsky ve Papert, algılayıcının sınırlarına yönelik alıřmalarının izleyen dönemdeki sinir aęlarına yönelik arařtırmalardaki düşüřte ok fazla etkisinin olmadığına, bu dönemin sinir aęlarının temel teorisine odaklanılarak daha iyi makinelerin ve eęitim algoritmalarının geliştirilmesiyle geçtiğini savunmuşlardır. Aslında o dönemde gözden kaçırılan gerek bu eleřtirinin doğrusal aktivasyon fonksiyonu kullanan iki katmanlı sinirsel sistemlere yönelik yapılmış olmasıdır

Sinir aęlarının önde gelen arařtırmacıları Minsky-Papert 'in eleřtirisine bir veya daha fazla gizli (hidden) katman içeren böylece karmařık problemlerin üstesinden gelen daha kompleks modellerin oluşturulmasıyla cevap vermişlerdir. Önceki algılayıcı modellerinden farklı olarak bu yeni modellerin içerisine doğrusal olmayan bir aktivasyon fonksiyonu konulmuřtur (Garson, 1997:5).

Bu durgunluk dönemi içersinde oldukça önemli bu alıřmanın amacı açısından ayrıntılarına girilmesinin pek önem taşımadığı alıřmalar ve yenilikler ortaya ıkmıştır. Sinir aęları literatürün ve alanının gelişmesine, farklı bakış açılarından genişlikler kazandırmış öncü arařtırmacılardan bazılarının ismini vermek gerekirse bunlar; James Anderson, Stephen Grossberg, von der Malsburg, Kohonen, Hinton ve John Hopfield 'dir. Özellikle Hopfield 'in alıřması yinelemeli aęlar (recurrent networks) tarafından yerine getirilen hesaplamaları formüle etmede fiziksel enerji fonksiyonları düşüncesinden yola ıkmış ve fizikçileri de sinir aęları alanına ekerek 1980 'lerin başında sinir aęlarına olan ilginin yeniden artmasında önemli rol oynamıştır.

Minsky-Papert 'in eleřtirisine yönelik olarak sinir aęları arařtırmacılarının doğrusal olmayan aktivasyon fonksiyonuna sahip ok katmanlı modeller yaratmasının ardından ortaya ıkan problem, bu ve daha karmařık aęlar için uygun bir eęitim algoritmasının geliştirilmesiydi. Werbos (1974), Harvard 'daki doktora tezinde geri yayılımın başlangıcı olan algoritmanın matematiğini geliřtirdi fakat bunu aędaki gizli katmanın aęlıklarını oluřtırmada kullanmamıştı. 1986 yılında

Rumelhart ve arkadaşları tarafından geri yayılım algoritmasının tanımlanmasıyla üne kavuşana kadar Werbos 'un çalışması yaklaşık 10 yıldan fazla bir süre zarfından bilim dünyası çevresi içinde bilinmeden varlığını sürdürmüştür. Geri yayılım öğrenme modeli hala genel kullanım içerisinde en çok başvurulan sinir ağı modeli olma özelliğini korumaktadır. Yaklaşık olarak aynı zaman içerisinde, iki farklı yerde ve birbirinden bağımsız olarak geri yayılım öğrenmesinin keşfedildiği görülmektedir (Garson, 1997:7).

1988 yılında Broomhead ve Lowe, radyal tabanlı fonksiyonları (Radial Basis Functions-RBF) kullanan, çok katmanlı algılayıcıya bir alternatif olan ileri beslemeli katmanlı ağ tasarımı için bir prosedür tanımladılar. RBF modelleri, sinir ağları tasarımının nümerik analiz alanıyla bağlama çabalarında önemli derecede yol kat etmiştir (Abdi, Valentin ve Edelman, 1999:61). 1989 yılıyla beraber sinir ağlarını modelleme hareketi 1970 'lerde ki gerilemeden tam anlamıyla kurtulmuştur. Hornik ve arkadaşları sinir ağları alanında kilometre taşı olan çalışmalarında, sinir ağlarının girdi ve çıktı bağlantıları arasındaki hem doğrusal hem de doğrusal olmayan ilişkiye yaklaşımda evrensel yaklaşımcı (Universal Approximators) olduğunu gösterdi. Kuan ve White (1994) ikili probit, ikili lojit ve doğrusal regresyon modellerinin sinir ağları modellerinin özel bir durumu olduğunu gösterdi.

Yapılan birçok yayım içerisinde 1980 'lerde sinir ağlarına olan ilginin yeniden yükselmesinden sorumlu tutulabilecek en önemli çalışmalar, 1982 'deki Hopfield 'in makalesi ve 1986 'daki iki ciltten oluşan Rumelhart ve McLelland tarafından yazılan kitaptır (Haykin, 1994:44). McCulloch ve Pitts 'den bu yana sinir ağları önemli bir yol kat etmiş ve nöroloji, psikoloji, matematik, fiziksel bilimler ve mühendisliği de içersine alan bir disiplinler arası bir alan haline geçmiş ve gelişimine de sürekli olarak devam etmektedir.

### **1.5. Sinir Ağlarının Faydaları**

Sinir ağları hesaplama gücünü, ilk olarak büyük bir paralellik içerisinde dağıtılmış olan yapısından ve ikinci olarak ise genelleme yapmasına imkân veren öğrenme yeteneğinden almaktadır. Genelleme sinir ağının eğitim (öğrenme)

esnasından hiç karşılaşmadığı girdiler için makul çıktılar üretmesine karşılık gelmektedir. Bu iki bilgi işleme yeteneği kontrol edilmesi güç olan karmaşık (büyük ölçekli) problemlerin çözülebilmesini sağlamaktadır. Sinir ağlarının kullanımı aşağıdaki işe yarayan özellik ve yeteneklerden yararlanması anlamına gelmektedir.

**1. Doğrusal Olmama.** Nöron temelde doğrusal olmayan bir yapıya sahiptir. Bir sinir ağı ise nöronların karşılıklı olarak bağlanmasından oluştuğu için doğası itibariyle doğrusal değildir. Ayrıca ağın her tarafına dağıtılmış olmasından ötürü özel bir çeşit doğrusal olmama durumu söz konusudur. Özellikle girdi setinin yaratılmasından sorumlu altta yatan fiziksel mekanizma doğası gereği doğrusal değilse bu durumda doğrusal olmama özelliği büyük bir öneme sahip olmaktadır. Sinir ağı modellerinin özellikle parametrik olmayan doğası, doğrusallık ve normallik varsayımının sağlandığından kuşku duyulan sosyal bilim verilerine uygunluk sağlamaktadır.

**2. Girdi ve Çıktıların Haritalanması.** Eğitim örnekleri olarak adlandırılan bir setin kullanılmasıyla sinir ağlarının sinaptik ağırlıklarının değiştirilmesine denetimli öğrenme denilmektedir. Her bir örnek tek bir girdi sinyali ve buna karşılık gelen istenilen cevaptan oluşmaktadır. Ağa girdi setinden rassal bir örnek sunulur ve ağın sinaptik ağırlıkları istenilen cevap ile uygun bir istatistiksel kritere göre girdi sinyalinden üretilen ağın gerçek cevapları arasındaki farklılığı minimize etmek amacıyla değiştirilir. Ağın eğitilmesine sinaptik ağırlıklarda önemli değişimlerin olmadığı bir durgunluk durumuna ulaşıncaya kadar eğitim setindeki çoğu örneğin tekrarlı bir şekilde kullanılmasıyla devam edilir. Daha önce kullanılmış eğitim örnekleri eğitim süresi boyunca tekrarlı olarak fakat farklı sırada olmak üzere kullanılabilir. Böylece ağ eldeki problem için girdi ve çıktıların bir haritasının oluşturarak örneklerden öğrenir. Böyle bir yaklaşım istatistiğin bir dalı olan model olmaksızın tahminleme ile ilgilenen parametrik olmayan istatistiksel yorumlama çalışmalarını akla getirmektedir. Örneğin fiziksel bir objeyi veya önceden kategorisi belirlenmiş bir olayı temsil eden bir girdi sinyalinin belirli bir sınıfa atanmasının gereklilik olduğu desen sınıflandırma problemini ele alalım. Bu problemi parametrik olmayan yaklaşım açısından değerlendirirsek

eldeki örnek setinin kullanılmasıyla desen sınıflandırma görevi için girdi sinyal uzayındaki keyfi karar sınırlarının tahminlenmesi bir gereklilik olmaktadır ve bu bir olasılıksal dağılım modeli istenmeksizin yapılmaktadır. Benzer bir bakış açısının parametrik olmayan istatistiksel yorumlama ve girdi-çıkı haritalanmasını yerine getiren bir sinir ağı arasındaki yakın bir benzerliği öneren denetimli öğrenme paradigmasında olduğu açıktır (Haykin, 1994:44).

- 3. Uyarlanabilirlik.** Sinir ağları sinaptik ağırlıklarını onu kuşatan çevredeki değişimlere göre uyarlayabilme yeteneğine sahiptir. Özellikle belirli bir ortam içerisinde işletilmek amacıyla eğitilen bir sinir ağı işletildiği ortamın koşullarındaki küçük değişikliklerin üstesinden gelmek için kolaylıkla yeniden eğitilebilir. Ayrıca durağan olmayan bir ortamda içerisinde işletiliyorsa sinir ağı zaman içerisinde sinaptik ağırlıklarını değiştirecek şekilde tasarlanabilir.
- 4. Delisel Cevap.** Desen sınıflandırılması kapsamında duruma bakarsak, sinir ağı sadece hangi desenin seçileceği hakkında bilgi sağlamamakta aynı zamanda verilen kararın güvenilirliği hakkında da bilgi sağlamaktadır. Bu bilgi gelecekte karşılaşılabilecek olan belirsiz desenlerin kabul edilmemesinde kullanılabilir. Böylece ağın sınıflandırma performansı geliştirilmiş olacaktır.
- 5. Bütünsel Bilgi.** Bilgi sinir ağının oldukça yapısal ve hareketsel durumu içerisinde temsil edilmektedir. Ağ içerisindeki her nöron gene ağ içerisindeki diğer tüm nöronların toplamsal faaliyeti tarafından bir etkiye maruz kalmaktadır. Sonuç olarak ise bütünsel bilgi sinir ağının doğası gereği ortaya çıkmakta ve böylece karşılıklı etkileşimleri kendiliğinden içerisinde barındırmaktadır.
- 6. Hata Toleransı.** Sinir ağı modellerinde hesaplama sürecinin çok sayıdaki işlem elemanının üzerine dağıtılmış olması sinir ağlarını hatalara karşı toleranslı yaparak gürültü, üst üste binmiş, yüksek derecede doğrusal ve

sürekli olmayan verilerle bile çalışmasına imkan vermektedir. Bir donanım içerisinde işletilen sinir ağının kötü çalışma koşullarının etkisiyle performansında düşüklüklerle karşılaşılabilir. Bu durum hata toleransı kavramının incelenmesine sebep olur. Oluşturulan sistemlerde hata toleransının yüksek olması istenmektedir böylece ortaya çıkabilecek sorunlarda sistem çalışma yeteneğini sürdürebilmektedir. Örneğin bir nöron veya onun bağlantı değerlerinde bir bozukluk meydana gelirse hafızaya alınan desenin hatırlanmasında kalite olarak bir zayıflık olur. Bununla beraber bilginin ağ içerisinde dağıtılmış olmasından ötürü bozukluk çok yüksek seviyede olmadığı sürece ağ genel olarak doğru cevaplar vermeye devam edecektir. Böylece karşılaşılan çok önemli olmayan bir sorun sistemin tamamıyla durmasına neden olmayacak belirli sınırlar altında sistem doğru bir şekilde işlevini sürdürecektir. Bu aynı zamanda sapan değerlerin çıktı üzerindeki etkisinin istatistiksel modellere göre daha az olmasını sağlamaktadır.

**7. Büyük Ölçekli Entegre Teknolojilerde Uygulanabilirlik.** Sinir ağının yüksek derecedeki paralellik yapısı onu belirli görevlerin hesaplanmasında oldukça hızlı yapmaktadır. Aynı özellik büyük ölçekli entegre teknolojileri (Very Large Scale Integrated (VLSI) technology)) kullanan uygulamalar içinde sinir ağlarının kullanımını uygun hale getirmektedir. Büyük ölçekli entegre teknolojinin başlıca özelliği hiyerarşik bir yöntemle oldukça karmaşık bir davranışının çözümlenmesine bir araç sağlamasıdır. Bu ise sinir ağlarının desen tanımlama, sinyal işleme ve kontrol gibi gerçek zamanlı uygulamalarda bir araç olarak kullanılması imkanını doğurmaktadır.

**8. Analiz ve Tasarımın Tek Düzeliği.** Sinir ağları bilgi işlemcilerinden kaynaklanan evrensellik özelliğine sahiptir. Bu bağlamda sinir ağları uygulamalarını içeren tüm alanlarda aynı notasyon kullanılır. Bu özellik kendini değişik şekillerde ortaya çıkarmaktadır:

a. Farklı biçimlerdeki nöronlar tüm sinir ağlarının genel bir parçasını temsil etmektedirler.

b. Bu genellilik özelliği farklı sinir ağları uygulamalarında teori ve öğrenme algoritmalarının paylaşılabirliğini mümkün kılmaktadır.

c. Birimlerin sürekli olarak entegre edilmesiyle değiştirilebilir ve düzenlenebilir ağlar inşa edilebilmektedir.

**9. Nörobiyolojik Benzerlik.** Sinir ağlarının tasarımı hata toleranslı paralel işleminin yalnız fiziksel olarak değil aynı zamanda hız ve güç özelliği ile de mümkün olduğunun yaşayan kanıtı olan beynin temel alınmasıyla oluşturulmuştur. Ancak belirtmelidir ki şu anki sinir ağı modelleri bu gücün sadece çok küçük bir oranını ele geçirebilme yeteneğine sahiptirler. Nörobiyologlar sinir ağlarına nörobiyolojik fenomenin yorumlanmasında bir araç olması gözüyle bakmaktadırlar. Öte yandan mühendisler ise nörobiyolojiye geleneksel tasarım tekniklerine göre oldukça karmaşık problemleri çözümede yeni düşünceler sağlanabileceği gözüyle bakmaktadırlar. Nörobiyolojik benzerlik şu bakımdan da oldukça yararlıdır. Nörobiyolojik yapının fiziksel olarak anlaşılması elektronik sanatını ve dolayısıyla da büyük ölçekli entegre teknolojileri etkileyebileceğine olan inanç ve umut nörobiyolojik benzerliğe yönelik çalışmalarını tetiklemektedir.

## **1.6. Gerçek ve Yapay Nöronlar**

Yapay sinir ağlarının temel inşa birimleri yapay hücrelerdir. Yapay hücreler, gerçek (biyolojik) nöronların bilgi işleme yeteneklerinin elde edilmesine yönelik bir girişimin sonucunda oluşturulmuştur. Bu nedenden ötürü öncelikli olarak gerçek nöronların işleyişinin ve yapısının kısa bir tanımının verilmesi benzer yapay nöronlara geçişte, oluşturulan benzerliğin boyutunun ve sınırlarının görülmesi açısından önem taşımaktadır. Gerçek nöronların tanıtılması konunun amacının dışına çıkılabileceği olasılığından ötürü belli bir giriş düzeyinde tutulacaktır.

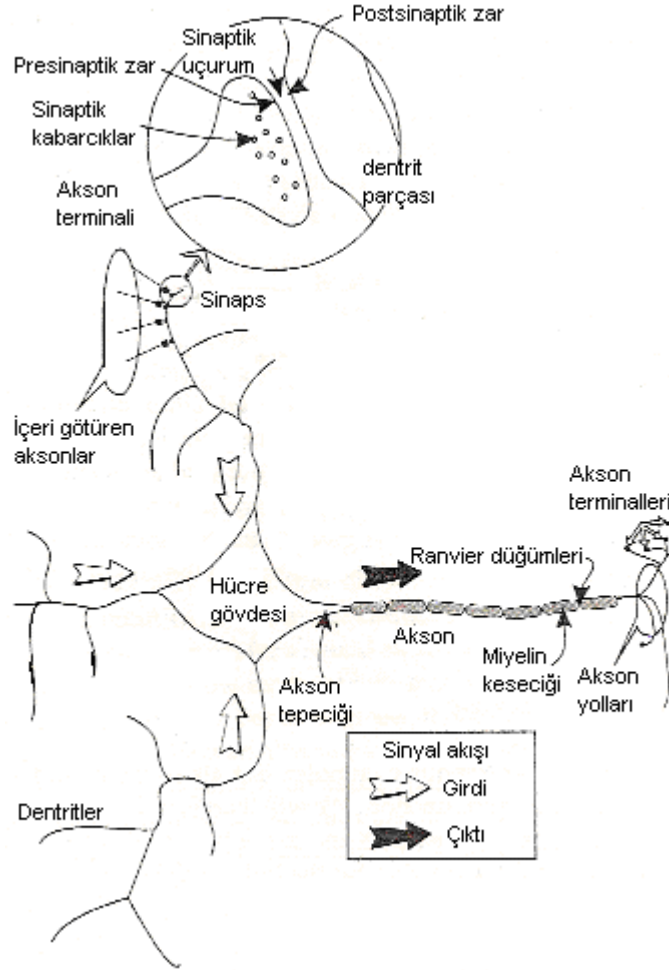
### 1.6.1. Gerçek Nöronlara Bir Bakış

Sinir hücreleri (nöron) yalnızca çok karmaşık değil aynı zamanda yapı ve fonksiyonlarının ayrıntılarında oldukça farklılık göstermektedirler. Bu yüzden sinir hücrelerinin çoğunluğunun sahip olduğu tipik özellikleri tanımlanıp ve hesaplama yeteneklerinin büyük bir bölümünü tasvir eden genel çalışma varsayımları yapılacaktır.

Tipik bir nöron Şekil 1-5 de verilmiştir. Hücre gövdesi neredeyse vücudun tüm hücrelerinde bulunan alt organelleri içerir (çekirdek, mitokondri, golgi aygıtı vs) fakat bunlar diyagramda gösterilmemiştir. Bunun yerine sinir hücresinin bir sinyal işleme aygıtı olarak fonksiyonelliğine izin verilerek diğer hücrelerden ne gibi farklılığının bulunduğu odaklanılacaktır. Bu yetenek büyük ölçüde hücre yüzeyini kaplayan hücre zarının elektrokimyasal süreçleri destekleyen özelliklerinden kaynaklanmaktadır. Şekilsel olarak baktığımızda ana farklılığın hücrenin gövdesinden çıkan lifler setine dayandığı görülür. Bu liflerden biri olan akson diğer hücrelere sinyal gönderilmesinden sorumludur ve böylece nöron çıktısı olarak görülebilir. Diğer tüm lifler öteki hücrelerden hücrenin gövdesine sinyaller taşıyan ve bir bakıma nöron girdileri gibi davranan dentritlerdir. Her nöron çok sayıda dentrite sahip olabilir fakat yalnızca bir aksone sahiptir. Dentritlerin yüksek derecede kola ayrılmış bir yapıya sahip oldukları görülürken aksonun diğer birçok nöronla iletişim kurulmasına olanak sağlayan eşit öneme sahip yol seti şeklinde kollara ayrıldığı görülür. Belirli bir nöronu ele aldığımızda, o nörona girdi sağlayan diğer nöronlara içeri götüren (afferent) ve bu nöronun diğer hücrelere yansması olarak görülen aksonal çıktısına ise dışarı götürülen (efferent) olarak isim verilmektedir. İçeri götüren aksonlar, sinaps olarak adlandırılan kavşaklarda dentritlerle iletişime geçmekte ve belirli bir nöronu sinirle donattıkları söylenmektedir. Aksonun uç noktası veya akson terminali olarak adlandırılan uç nokta dentrit yüzeyinin küçük bir parçasıyla (postsinaptik zar) birbirlerine çok yaklaşırlar. Burada presinaptik akson terminal zar ve onun postsinaptik karşılığı arasında sinaptik uçurum olarak adlandırılan bir boşluk vardır. Şekil 1-5 de kolaylık sağlaması açısından sadece birkaç sinaps gösterilmiştir halbuki tüm dentritlerin üzerine yerleşmiş ve muhtemelen



hücre gövdesinde de çok sayıda sinaps bulunmaktadır. Ayrıntılı sinapstik yapı şekil içerisinde ek bir şematik formda büyütülerek gösterilmiştir.

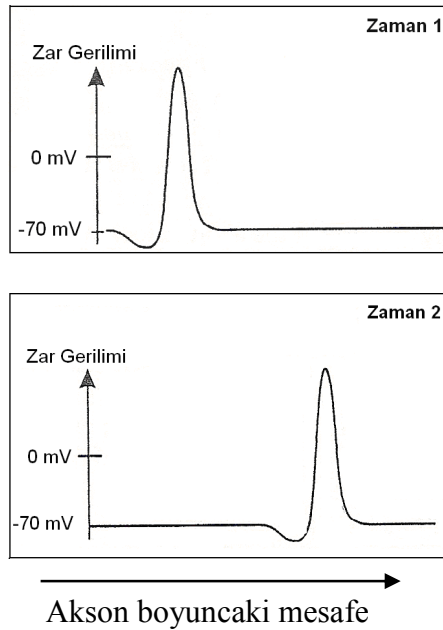


**Şekil 1- 5: Biyolojik Nöron**

*Kaynak: Gurney, 1997, s.8.*

Sinirsel yapının sinyal işlemeyi nasıl desteklediği ele alalım. Denge durumunda hücre zarı, negatif ve pozitif olarak yüklenmiş iyonlardaki elektriksel dengesizliğin iletilmesi görevini yerine getirir. İyonlar aşırı veya az yüklenmiş elektronlara sahip molekül veya atomlardır. Hücrenin içerisi ve zarı arasında gerilim farklılığı vardır. Hücrenin içerisi dışına göre yaklaşık olarak 70 mV (milivolt, bir voltun binde biri) negatif kutupsallaşmıştır. Hücre zarında bir volt ölçeği yardımıyla ölçüm yaptığımızı varsayarsak aygıt içerinin dışardan daha negatif olmasıyla 70 mV okuyacaktır. Buradaki temel nokta sinir zarının, zarın gerilimi dinamik olarak

değiştikçe elektriksel sinyal akışını desteklemesidir. Bu duruma bakmak için Şekil 1-6 da gösterildiği gibi bir akson boyunca sinyal yayılımı durumunu ele alalım. Akson boyunca yayılan sinyallerin veya kuvvet gerilimi de denebilir, tümü sivri uçlu nabza benzeyen aynı karakteristik şekle sahiptir. Şekil 1-6 daki her grafik o andaki tek bir kuvvet gerilimini gönderen bir akson parçası boyunca ki zar geriliminin anlık fotoğrafını göstermektedir. Şekil 1-6 'nın altındaki panel üstekine göre aynı durumun belirli bir zaman sonra aldığı şekli göstermektedir. Bu sürecin üretilmesinde çalışan iyonik mekanizmalar ilk olarak Hodgkin&Huxley tarafından ortaya çıkarılmıştır (Arbib, 2003:5). Süreç hücre zarı boyuncaki iyonik akımların her biri arasındaki karşılıklı etkileşime dayanmaktadır ve matematiksel tanımı oldukça karmaşıktır. Bu konunun ayrıntıları konumuzun dışında olmasına rağmen gerçek nöronların karmaşık, doğrusal olmayan dinamiklere sahip olduğunu görmekteyiz. Yapay nöronları kullanılarak modelleme çalışması yaptığımızda bu durumu pasif bir çıktı bağlantı kablosu olarak modelleyeceğiz. Çoğu nöron daha hızlı kuvvet gerilimi iletimine imkan veren ve miyelin olarak bilinen yağlı bir madde içerisinde muhafaza edilmiş olarak aksonlarına sahiptir. Aksonlar bu miyelin parçalarından oluşmuştur ve her miyelin arasında 1mm aralıklı küçük miyelinsiz bölümler (Ranvier düğümleri) bulunmaktadır, miyelinsiz bölümler bir telefon kablosu boyunca ki tekrarlatıcı istasyonlar gibi davranmaktadır.



**Şekil 1- 6: Kuvvet Gerilimlerinin Yayılımı**

Bu noktadan sonra artık içeriye götüren bir akson terminaline kuvvet geriliminin ulaşmasıyla başlayan ve bir tek nörona yönelik sinyallerin geçişini ele alabiliriz. Akson terminaldeki çok sayıdaki küçük kabarcığın içerisinde kimyasal bir madde veya nöroiletici bulunur. Bir kuvvet geriliminin alınmasıyla kabarcıklar prisinaptik zara göç eder ve sinaptik uçuruma nöroileticilerini bırakırlar. Daha sonra nöroileticiler postsinaptik zarda yakalayıcı bölgelerle kimyasal olarak bağlanırlar. Böylece sinapslardaki hücre zarının kutuplaşma durumunu değiştiren elektrokimyasal süreç başlatılmış olur. Postsinaptik gerilim (PSP) hücre zarının negatif durumdan 0 volt düzeyine doğru nötürleşmesine veya hücre zarının daha büyük bir negatif gerilime doğru kutuplaşmasına hizmet edebilir. İleride görebileceğimiz gibi sinirsel sinyal üretimi nötürleşme tarafından teşvik edilir böylece pozitif olan PSP ler tetikleyici iken (EPSP) hücre zarını negatif olarak kutupsallaştıran PSP ler ise engelleyicidir (IPSP). Kuvvet gerilimlerinin hepsi aynı karakteristik sinyal profiline ve aynı maksimum değere sahipken PSP ler sinapslerin elektriksel sinyal üretimde kimyasal ileticileri kullanma verimliliğine bağlı olarak sürekli bir aralıkta değerler alabilir. PSP sinapslerden yayılarak onu hücre gövdesine bağlayan dentrit boyunca yolculuk eder ve varış noktası olarak akson başlangıç kısmı ile hücre gövdesinin birleştiği akson tepeciğine ulaşır. Bununla eş zamanlı olarak aynı şekilde meydana gelen nöron üzerine dağılmış binlerce sinapstik olay gerçekleşir. Bunun sonucu sürekli bir şekilde akson tepeciğine ulaşan ve burada birlikte toplanarak nihai hücre zarı gerilimini ortaya çıkaran çok fazla sayıdaki PSP lerdir.

Akson tepeciğine gelen her PSP nin yok olmaya başlamadan önce milisaniyeler büyüklüğünde bir yaşam süresi vardır. Eğer iki PSP çok az bir zaman farklılığı ile akson tepeciğine ulaşırsa hala toplanma sürecine birlikte katılma şansına sahiptirler. Öte yandan uzun bir dentrit kolunun ucunda olmasından ötürü hücre gövdesinden uzakta gerçekleşen sinapstik bir olay ile hücre gövdesine yakın bir yerde gerçekleşen iki farklı sinapstik olayı ele alalım. Bir süre sonra uzakta bulunan sinapsdeki PSP akson tepeciğine ulaşmıştır fakat bu durumda hücre gövdesine daha yakın bir kaynaktan gelen PSP ortadan kaybolmuş olacaktır. Böylece her ne kadar PSP lerin ortaya çıkışı aynı zamana denk gelse de bunlar kuvvet geriliminin

yaratılmasında aynı sürecin içersine dahil olmamışlardır. Açık olarak görülen bir nöronun PSP lerini zaman ve yer kısıtının birlikte ele alınmasıyla topladığı veya birleştirdiğidir. Her ne kadar connectionist modeller aksonlara zamanla ilişkisi olmayan pasif bir bağlantı kablosu gözüyle baksa da çoğunluğu Rall (1957,1959) tarafından öncülük edilen önemli modelleme çabaları dentritler boyunca ki PSP lerin iletiminin ve bunların birbiri arasındaki karşılıklı etkileşimin tanımlanmasına yöneliktir (Gurney, 1997:10).

Akson tepesiğinde toplanan PSP hücre zarının gerilimini etkileyecek ve eğer bu belirli bir eşik değerini aşarsa (genellikle yaklaşık olarak -50mV civarında) bir kuvvet gerilimi oluşturulacaktır. Bu kuvvet gerilimi aksondan çıkan eşit öneme sahip yol setleriyle akson terminaline ulaşır ve komşu nöronlarda sinaptik olaylar sağanağının başlamasına neden olur. Bir kuvvet gerilimi üretildikten sonra bu kuvvet geriliminin üretiminde kullanılan iyonik metabolizmalar ortadan kaybolur ve kısa sürelik durgunluk dönemine girilir, durgunluk döneminde hücre zarı gerilimi hangi değeri alırsa alsın diğer bir kuvvet gerilimi oluşturma süreci başlatılmaz.

Yapay nöronları modellemek için gerekli olan gerçek nöronların fonksiyonelliği hakkında şimdiye kadar anlatılanları özetlememiz bu aşamada yararlı olacaktır. Ayrıca Ek 1 içersine gerçek nöronları anlatırken kullanılan terimlerin anlamlarını içeren küçük bir sözlük konmuştur.

- Nöronlar arasındaki sinyaller kuvvet gerilimleri aracılığıyla gönderilir. Kuvvet gerilimleri var ya da yok karakterini temsil edecek şekilde aynı profile sahiptirler yani kuvvet geriliminin yarısı gibi bir durum söz konusu değildir.
- Bir kuvvet gerilimi nöronsal girdiyi (sinapsi) uyardığında bunun etkisi bir PSP oluşumudur.
- PSP ler tetikleyici veya engelleyici olabilir.

- PSP ler akson tepeceğinde hücre zarı gerilimini oluşturmak amacıyla bir araya gelip toplanırlar.
- Eğer bu gerilim belirli bir eşik değerini aşarsa akson boyunca ilerleyecek olan kuvvet gerilimi başlatılır.

### 1.6.2. Yapay Nöronlar

Bu kısımda gerçek nöronlar hakkında bahsedilirken tanıtilen bileşenlerden bazılarının yapay nöronlar içerisinde nasıl modellenmiş olduğu anlatılacaktır. Böylece gerçek nöronlarla, onlara mümkün olduğunca benzetilmeyi temel alan yapay nöronlar arasında bir ilişki kurulabilecektir.

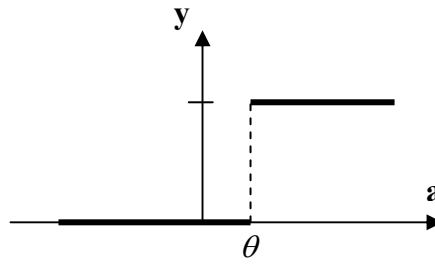
Kuvvet geriliminin “var yada yok” karakteri iki değerli bir sinyalin kullanılmasıyla modellenebilir. Böyle sinyallere genellikle binary (ikili) veya Boolean (doğru veya yanlış değerleri gösteren formal mantık) denilmektedir ve geleneksel olarak 0 ve 1 değerlerini almaktadır. Buradan hareketle eğer  $x_1, x_2, \dots, x_n$  gibi  $n$  tane girdi sinyalini alan bir hücreye sahipsek o zaman bu girdi sinyalleri sadece 0 veya 1 değerlerini alabilecektir. Daha önce söylenenler ile aynı doğrultuda hareket edildiğinde her sinapsin düzeltici etkisi basitçe girdi sinyalinin bir ağırlık değeriyle çarpılmasıyla açıklanır. Burada tetikleyici ve engelleyici gerilimler sırasıyla pozitif ve negatif değerlerin kullanılmasıyla modellenir. Böylece  $w_1, w_2, \dots, w_n$  gibi  $n$  tane ağırlık değeri  $w_1x_1, w_2x_2, \dots, w_nx_n$  gibi  $n$  tane çarpım değerine şekil verecektir. Artık her çarpım bir PSP benzerliğini taşımaktadır ve ağırlık işaretine bağlı olarak pozitif veya negatif olabilecektir. Daha sonra ise bu çarpımların akson tepeciğinde gerçekleşen olaya benzetilmek amacıyla benzer bir sürecin içerisinde birleştirilmesi gerekir. Bu basitçe  $a$  aktivasyonunun (akson tepeciğindeki hücre zarı gerilimine karşılık gelmektedir) üretilmesi için çarpım değerlerinin toplanmasıyla gerçekleştirilir.

$$a = w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (1.1)$$

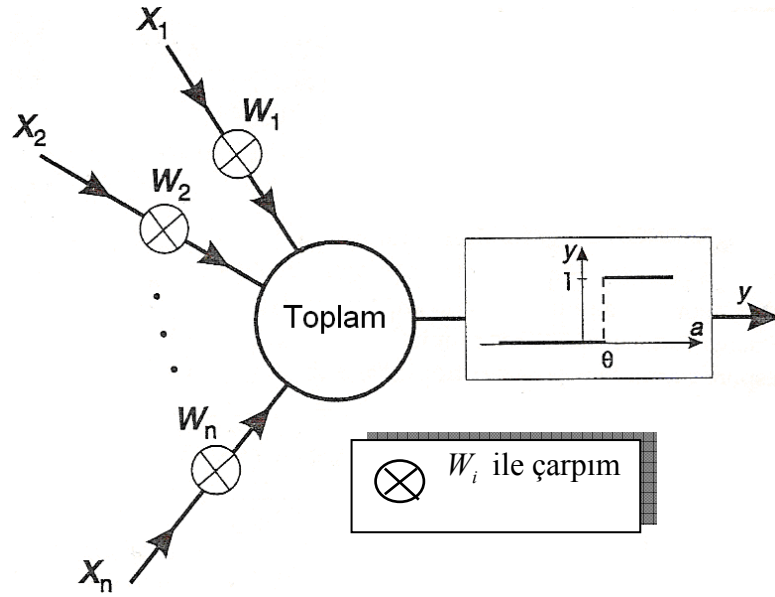
Örnek olarak beş girdili bir birimi şu ağırlık değerleri ve girdileriyle aldığımızı düşünelim; ağırlıklar  $w_1=0.5$ ,  $w_2=1$ ,  $w_3=-1$ ,  $w_4=-0.5$  ve  $w_5=1.2$  ve girdiler  $x_1=1$ ,  $x_2=1$ ,  $x_3=1$ ,  $x_4=0$ ,  $x_5=0$  olsun. (1.1) eşitliğinin kullanılmasıyla şu elde edilir:

$$\begin{aligned} a &= (0.5 \times 1) + (1 \times 1) + (-1 \times 1) + (-0.5 \times 0) + (1.2 \times 0) \\ &= 0.5 \end{aligned}$$

Kuvvet gerilimlerinin oluşturulma sürecinin benzetilebilmesi için bir  $\theta$  eşik değerine ihtiyaç vardır. Böylece eğer aktivasyon  $\theta$  değerini aşar veya ona eşit olursa bu durumda hücrenin çıktısı 1 (yani bir kuvvet gerilimi) ve eğer aktivasyon  $\theta$  değerinden daha küçük ise hücrenin çıktısı 0 olacaktır. Bu grafik olarak hücrenin çıktısının  $y$  sembolü ile gösterildiği Şekil 1-7 deki gibi gösterilebilir. Örneğimizde  $\theta = 0.2$  olduğunu varsayarsak  $a > 0.2$  olduğundan hücrenin çıktısı  $y$ , 1'e eşit olacaktır. Şuana kadar anlatılan hücre yapısı Şekil 1-8 de özetlenmiştir. Burada ağırlıklar daire içindeki çarpım işaretiyle gösterilmiştir. Ağırlıkların ve sinyallerin büyüklüğünün şekillerle orantılı olması varsayımı yapılmamıştır. Bu çeşit yapay nöron ilk olarak McCulloch ve Pitts (1943) tarafından önerilmiştir ve eşik değer mantık birimi (Threshold Logic Unit TLU) olarak bilinir.



**Şekil 1- 7: Aktivasyon-Çıktı Eşik Değer İlişkisinin Grafikselsel Şekli**



**Şekil 1- 8: TLU Hücresi**

TLU ‘nun fonksiyonelliğini grafiksel bir şekildense sembolik bir form olarak gösterilmesi daha uygundur. Eşitlik (1.1) ile aktivasyon hesabı için elimizde bir form bulunmakla beraber bunu eşitlik (1.2) gibi daha derli toplu bir notasyonda ifade edebiliriz.

$$a = \sum_{i=1}^n w_i x_i \quad (1.2)$$

y çıktısını elde etmede kullanılan eşik değeri ilişkisi aşağıdaki gibi yazılabilir:

$$y = \begin{cases} 1 & , a \geq \theta \\ 0 & , a < \theta \end{cases} \quad (1.3)$$

TLU ‘nun zamanla ilgili hiçbir şey içermediğine dikkat etmek gerekir. Gerçek nöronlar girdilerine zaman ve buldukları konumu da dikkate alarak cevap verirken TLU ‘de hücre girdilerine anında cevap verilmektedir. Dentritler, ağırlıklar ve toplama işlemi arasındaki pasif bağlantı linkleri olarak gösterilmiş olup kuvvet gerilimi üretimi eşik değeri fonksiyonu aracılığıyla yapılmaktadır.

## 1.7. Gürültü ve Donanım Hatalarına Karşı Esneklik

Yapay sinir ağlarının biyolojik sinirsel sistemlerle ortak olarak paylaştığı önemli bir karakteristik hata toleransıdır (Fausett, 1993:6). Biyolojik sinirsel sistemler hata toleransına iki açıdan sahiptirler. Birincisi, daha önceden gördüğümüz herhangi bir girdi sinyalinden bir dereceye kadar farklı olan girdi sinyallerini fark edebilme yeteneğine sahibizdir. Bunun bir örneği aradan geçen epeyce uzun bir zaman diliminden sonra gördüğümüz arkadaşımızı tanıyabilmemiz veya daha önce hiç görmediğimiz resimdeki bir insanı fark edebilmemizdir. İkincisi, sinir sisteminin kendisinde meydana gelen hasarların tolere edilebilmesidir. İnsanlar yaklaşık olarak 100 milyar nöronla birlikte doğmaktadırlar. Bunların çoğunluğu beyinde bulunmakta ve hücreler öldükleri zaman yerlerine yenileri konmamaktadır. Sürekli olarak nöron kaybına rağmen öğrenmeye devam etmekteyiz. Hatta sarsıcı derecedeki sinirsel kayıp durumlarında bile bazen diğer nöronlar zarar gören hücrelerin fonksiyonlarını üstlenmek için eğitilebilmektedir. Benzer şekilde yapay sinir ağları da ağda oluşabilecek küçük boyutlu hasarlara karşı duyarsız olacak ve ağdaki ciddi hasar durumlarında (veri ve bazı bağlantıların kaybı gibi) yeniden eğitilebilecek şekilde tasarlanabilir.

Basit bir sinir hücresi modeli ile bile sinir ağlarının genel iki özelliği ortaya konulabilir. Örneğin elimizde ağırlıkları (0,1) ve eşik değeri 0.5 olan iki girdili TLU bulunsun. Mümkün olabilecek tüm girdi setine vereceği cevaplar Tablo 1.1 de gösterilmiştir.

**Tablo 1- 1: (0,1) Ağırlık ve 0.5 Eşik Değeriyle TLU**

$x_1$	$x_2$	Aktivasyon	Çıktı
0	0	0	0
0	1	1	1
1	0	0	0
1	1	1	1



Donanımımızın TLU fonksiyonelliğini hatalı olarak yerine getirdiğini varsayalım bunun sonucunda ağırlıklar gerçek değerlerinde tutulmamakta ve (0.2, 0.8) olarak yanlış kodlanmaktadır. Bu ağırlık değerleriyle uygulanan TLU 'in fonksiyonelliği Tablo 1.2 de verilmiştir. Dikkat edilmesi gereken durum, her ne kadar aktivasyon değeri değişmiş de olsa çıktı değerinin orijinal TLU 'da olduğu gibi kalacağıdır. Bu durumun ortaya çıkmasının nedeni aktivasyondaki değişimlerin eşik değerini aşmadığı sürece çıktıda herhangi bir değişikliğin olmamasıdır. Böylece eşik değer fonksiyonu aktivasyonun  $\theta$  'nın hemen altında veya  $\theta$  'dan çok düşük olup olmadığına bakmaksızın 0 değerini çıktı olarak verecektir. Benzer şey aktivasyonun  $\theta$  'yı ne kadar aştığı konusunda da söylenebilir bu durumda TLU her zaman 1 değerini çıktı olarak sunacaktır.

**Tablo 1- 2: (0.2, 0.8) Ağırlık ve 0.5 Eşik Değerleriyle TLU**

$x_1$	$x_2$	Aktivasyon	Çıktı
0	0	0	0
0	1	0.8	1
1	0	0.2	0
1	1	1	1

Bu davranış şekli doğrusal olmayan sistemlerin karakteristiğini göstermektedir. Doğrusal bir sistemde, çıktı girdi ile orantılı olarak ilişkilidir; girdideki küçük veya büyük değişimler her zaman ona karşılık gelen çıktıda küçük veya büyük değişimlere yol açacaktır. Diğer taraftan doğrusal olmayan ilişki böyle bir orantısız kısıtlamaya uymaz. Çıktıdaki değişim büyüklüğünün girdidekini yansıtması gerekli değildir. Bu bakımdan TLU örneğinde aktivasyon 0 dan 0.2 ye değişmiş olmasına (0.2 lik bir farklılık söz konusu) rağmen bu çıktı üzerinde herhangi bir farklılık yaratmamıştır. Eğer aktivasyon 0.49 dan 0.51 e değişmiş olsaydı (yine 0.2 lik bir farklılık söz konusu) çıktı aniden 0 değerinden 1 değerine geçecekti.

Tüm bunlardan çıkarılacak sonuç TLU nun donanım hatalarına karşı dirençli olduğudur. Donanımımızda meydana gelebilecek ufak bir sorunda, TLU onun doğrusal olmayan fonksiyonelliğinin bir sonucu olarak hala görevini kusursuzca yerine getiriyor olacaktır.

**Tablo 1- 3: Girdi Sinyalindeki Bozukluk ile TLU**

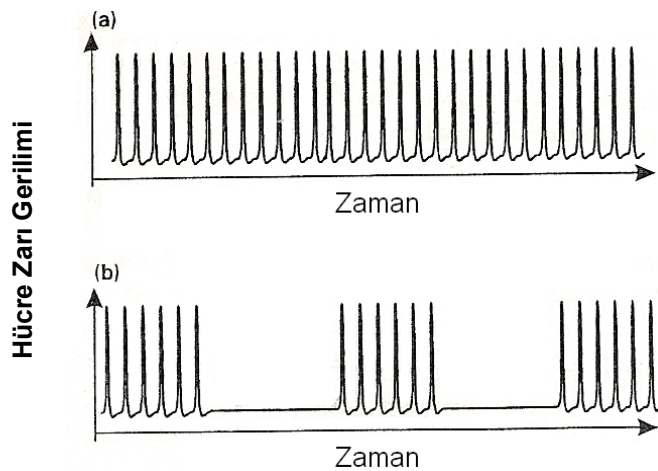
$x_1$	$x_2$	Aktivasyon	Çıktı
0.2	0.2	0.2	0
0.2	0.8	0.8	1
0.8	0.2	0.2	0
0.8	0.8	0.8	1

Şimdi ise ağırlıklardaki değişiklik yerine girdi sinyallerinde oluşabilecek gürültü veya kısmi güç kaybı gibi bazı nedenlere bağlı olarak bir bozukluk olduğunu ve sinyallerin değiştiğini varsayalım. Bu amaca yönelik olarak önceden 1 olan sinyalimiz 0.8 ve 0 olan sinyalimiz ise 0.2 ile ifade edilmiş ve son TLU fonksiyonunun aldığı değerler Tablo 1.3 de gösterilmiştir. Burada TLU fonksiyonu aynıdır ve eşik değerin yerine getirdiği doğrusal olmamayı içeren aynı mantık uygulanmaktadır. Sonuç gürültü veya bozulan sinyal girdilerinin varlığı durumuna karşı TLU nun dirençli olduğudur. Buradaki durum hem ağırlıkların hem de girdilerin aynı anda bozulması için de incelenebilir. Eğer ağırlık ve sinyallerdeki değişim miktarı çok fazla olursa TLU yanlış cevaplar vermeye başlayacaktır. Büyük bir ağı ele aldığımız zaman donanım ve/veya sinyallerdeki bozuklukların derecesi arttıkça yanlış sonuçlar veren TLU birimlerinin sayısı da artacaktır. Bu süreç olumlu bozulma olarak (graceful degradation) adlandırılmaktadır ve bir devredeki sinyal gücündeki kabın veya bir bileşenindeki değişimin tüm makine hatasına sebep olabileceği geleneksel bilgisayarlarla karşılaştırıldığında üstünlüğü gözler önüne çıkmaktadır.

## 1.8. İkili Olmayan Sinyal İletimi

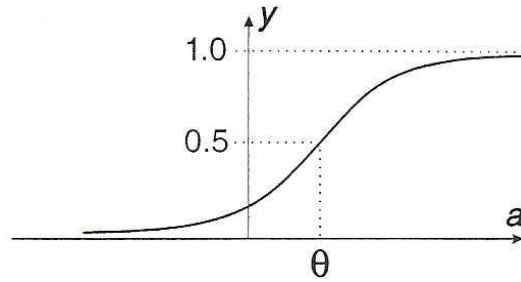
Şimdiye kadar ele aldığımız sinyaller sadece iki değer alıyordu. Bu durum gerçek nöronlar söz konusu olduğunda bir kuvvet gerilimi oluşmasıyla voltaj üretilmesi ve buna bağlı olarak da geriliminin artmasına veya hücre zarının geriliminde bir artışın olmamasına karşılık gelmektedir. TLU açısından ise sırasıyla kullanışlı olan 1 ve 0 değerleriyle temsil edilmektedir. Ancak gerçek nöronların basitçe tek bir voltaj üretiminin varlığı ve yokluğu yerine kuvvet gerilimlerinin oluşma desenine bağlı olarak sinyal değerlerini kodladıklarına inanılmaktadır. Şekil 1-9 da gözlemlenen çok sayıda karakteristik desenin iki genel örneği gösterilmiştir.

Şeklin (a) parçası kuvvet gerilimi voltajlarının sürekli akışını gösteriyorken (b) parçası ardışık bir voltaj üretimi daha sonra ise bunu bir durgunluk döneminin takip ettiği bir serinin kendini belirsiz olarak tekrarladığı bir deseni göstermektedir. (a) daki gibi sürekli bir akış saniye başına kuvvet gerilimin voltaj oluşturma frekansı ile karakterize edilebilir ve bunun aslında nöron tarafından gönderilen sinyal kodu olduğu varsayılabilir. Bu yaklaşım ikna edici bir şekilde Hartline tarafından özel bir yengeç türü üzerinde, görsel uyarıcının yoğunluğuyla gönderilen sinyal oranının arttığının gösterilmesiyle açıklanmıştır (Arbib, 2003:96). Her ne kadar çok sayıda kod bulunsa da çoğu örnekte frekans kodların kullanıldığı görülmektedir.



Şekil 1- 9: Sinirsel Desenler

Sinirsel voltaj üretme sıklığına  $f$  dersek bu takdirde  $f$  'in alttan 0 değeriyle ve üstten durgunluk dönemi süresi tarafından yönetilen bir maksimum  $f_{\max}$  değeriyle sınırlandırıldığını söyleyebiliriz. Şu anda yapay nöronlarda  $f$  'i kodlayabileceğimiz iki yol bulunmaktadır. İlk olarak basitçe, sinyal gösterimlerini 0 ve 1 değerlerinden sürekli bir aralığa genişletebiliriz. Böyle bir durum sinyal bozukluğunun etkisinin incelenmesinde açıklanmış ve TLU 'nun girdi kısmı içerisinde 0 ve 1 den farklı sinyal değerleri için içersine katılmıştı. Ancak çıktı kısmında adım fonksiyonunun (step function) kullanılması sinyallerin ikili değerler almasına sebep olmaktadır. Böylece ağ içerisinde TLU 'lar bağlı ise sürekli olarak derecelendirilmiş sinyallerin ortaya çıkmasına imkân kalmamaktadır. Bu durumun üstesinden adım fonksiyonunun yumuşatılıp onun yerine sürekli bir ezici/düzeltici/sıkıştırıcı fonksiyon (squashing function) kullanılmasıyla gelinir. Artık  $y$  çıktısı  $a$  aktivasyonuna daha yumuşak bir şekilde bağlı olur. Bunun için kullanılan uygun bir form Şekil 1-10 'da gösterilen Sigmoid fonksiyonudur.



**Şekil 1- 10: Sigmoid Fonksiyonu**

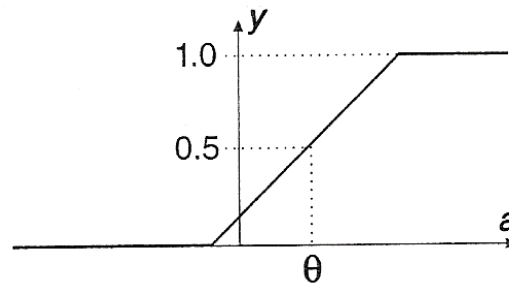
Sigmoid fonksiyonunda  $a$  büyük pozitif değerler aldıkça  $y$  'de 1 değerine yaklaşacak ama asla gerçekte bu değeri almayacaktır. Benzer şekilde  $a$  büyük negatif değerler alırken Sigmoid 0 değerine yaklaşacaktır. Eğer  $y$  değerini gerçek voltaj üretme oranı olarak yorumlamak istersek, basitçe Sigmoid fonksiyonunun değerini  $f_{\max}$  ile çarpabiliriz. Sigmoid fonksiyonu  $y$  eksenindeki 0.5 değeri civarında simetriktir ve buna karşılık gelen  $\theta$  ile gösterilen aktivasyon değerini eşik değeri gibi yorumlayabiliriz. Sigmoid fonksiyonu geleneksel olarak yunan alfabesindeki küçük sigma  $\sigma$  simgelenmiştir ve matematiksel ifadesi aşağıdaki gibidir.

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-(a-\theta)/\rho}} \quad (1.4)$$

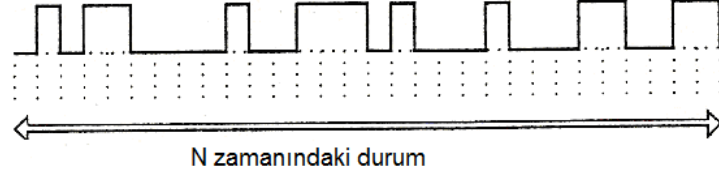
Formülasyonda  $e \approx 2.7183$  sonsuz ondalık değerine sahip bir matematiksel sabittir.  $\rho$  değeri büyük değerler aldıkça eğriyi yassı yaparken küçük değerlerinde ise eğrideki yükselişin daha dik olmasını sağlayarak fonksiyonun şeklini belirlemektedir. Bazen bu parametre atlanmakta bu durumda değerinin 1 olarak atandığı anlaşılmaktadır.  $\rho$  değerini sürekli olarak küçük değerlerde tutmamız TLU ‘da kullanılan adım fonksiyonuna daha çok benzemesine ve hatta özel bir durumu gibi görülmesine neden olacaktır. Böylece  $\theta$  değerine eşik değer denmesi TLU ‘daki aynı parametrenin rolünü almasından ötürü daha mantıklı hale gelecektir.

Sigmoid çıktı ilişkisini kullanan yapay nöronlar veya birimlere yarı doğrusal denilmektedir. Aktivasyon hala (1.2) eşitliğiyle elde edilmekte fakat çıktı değeri (1.4) tarafından verilmektedir. Yarı doğrusal terimi Şekil 1-11 de gösterildiği gibi sürekli bir parçalı doğrusal fonksiyon aracılığıyla Sigmoid fonksiyonuna yaklaşabiliyor olmamız gerçeğinden gelmektedir. İlgili bölgenin önemli bir kısmında, çıktı değerleri aktivasyon fonksiyonun orta boyuttaki değerleriyle eğimi sıfır olmayan doğrusal bir ilişki içerisindedir.

Sürekli sinyal değerlerinin kullanılmasına bir alternatif olarak, gerçek nöronun öykülenerek bir sinyali Şekil 1-12 de gösterildiği gibi bir voltaj akımındaki 1 ‘in ortaya çıkma frekansı olarak kodlayabiliriz.



**Şekil 1- 11: Sigmoid 'in Parçalı Doğrusal Yaklaşımı**



**Şekil 1- 12: Stokastik Bir Birimdeki Voltaj Akımı**

Zaman kesikli kısımlara ayrılmakta ve her kısım bir 0 (voltaj yok) veya bir 1 (voltaj var) olarak sınıflandırılmaktadır. Birim çıktısı önceden olduğu gibi şekil almakta yalnız Sigmoid fonksiyonunun değeri direkt olarak gönderilmeden önce bunu bir voltajın veya 1 'in ortaya çıkma olasılığı olarak yorumlarız. Bu şekilde olasılık kuralları tarafından yürütülen bir sürece stokastik ve bu birimlere de stokastik yarı doğrusal birimler denilmektedir. Artık sıra aldıkları sinyalleri girdileri olarak yorumlamaları beklenen birimlerin bunları nasıl birleştirip ileteceğine gelir. Birimlerin N zamanlık bölüm içerisinde aldıkları sinyalleri birleştirmeleri gerekmektedir. İçeri götüren hücrenin y olasılığı ile voltaj ürettiğini varsayalım. Bu zaman zarfı içerisinde voltaj üretim sayısının beklenen değeri  $yN$  'dir ve genellikle gerçekte üretilen  $N_1$  bu değere eşit olmayacaktır. Sinyalleri alan bir hücrenin birleştirme zamanı süresince ki 1 'leri üretme oranının en iyi tahmini  $N_1/N$  'dir. Bu durum madeni bir parayla yazı-tura atma deneyine benzemektedir. Bir paranın atılmasında yazı gelme olasılığı 0.5 'tir fakat para arka arkaya atılırken yazıların sayısı  $N_h$  toplam deneme sayısının yarısına eşit olmak zorunda değildir. Bununla birlikte toplam deneme sayısı N artarken  $N_h/N$  oranı sonunda 0.5 değerine yaklaşacaktır.

## **1.9. Sosyal Bilimler İçerisinde Sinir Ağları Kullanımının Yayılmasında**

### **Karşılaşılan Engeller**

Sinir ağları analizi son yıllarda tahminlemenin önemli olduğu problem uygulamalarında sıklıkla kullanılan bir metodoloji olmuş ve örneğin hisse senetlerinin finansal analizi gibi bir alan içerisinde karşılaşılmaya başlanmış bir durum haline gelmiştir. Buna bakılarak sosyal bilimlerde kullanılan geleneksel yaklaşımların yerine hızlı bir şekilde geçebileceği düşünülebilir fakat gerçekte bu böyle olmamaktadır. Bunun böyle olmamasının nedenlerine ve aynı zamanda sinir

ağları araştırmacısının karşılaştığı zorluklara bakmak gerekmektedir. Bu nedenlerden biri yaklaşımın yeniliğinden kaynaklanmaktadır. Rumelhart ve McClelland çalışmaları 1986 yılında yayımlanmış ve sosyal bilimcilerin kullanabileceği sinir ağları yazılımları ancak 1990 'lı yılların başlarında olanaklı olmuştur. Yöntemin yeniliğinin haricinde sosyal bilimlerde sinir ağları analizinin yayılmasına engel olan dört farklı neden bulunmaktadır.

İlk olarak, sinirsel modeller tahminleme alanında kendilerine yer bulsa da nedensel analizde yer bulamamaktadır. Sinirsel ağların sonuçlarına nasıl ulaştığını anlamak oldukça zor olabilmektedir. Şimdiye kadar tasarlanan sistemler, alternatif tekniklerin mesela uzman sistemlerde olduğu gibi sistemin sonuçlarına nasıl vardığını açıklayabilme yeteneğine sahip değildir. Daha da fazlası sinir ağları, uzman sistemlerin bir uzman sistem olarak var olan avantajlarını ele geçirecek şekilde tasarlanamazlar. Özellikle gizli katmanların işin içersinde olması, sinirsel analiz algoritmalarının nedensel bir yorumu ağ ağırlıklarına atamasının zor olduğu sonuçlar üretmektedir. Sinirsel modelleri kullanan birkaç tane nedensel analiz yaklaşımı bulunsa da açık bir şekilde sosyal bilimcilerin ana ilgisi basit bir tahminleme değil açıklama olduğu ortadadır. Sinirsel modellerin kullanımının şu anda olduğundan daha yaygın olmamasının ana nedeni bu olmuştur. Bu bağlamda Andrew Hunter (1997) sinir ağları ile şu gözlemini aktarmıştır (Garson, 1997:16):

Bir makinenin gerçekte tam anlamıyla nasıl çalıştığını anlamamanın olanaklı olamayabileceği kabul edilmektedir. Gerekli fonksiyonları yerine getirmeyi öğrenebilecek ağların inşa edilebilmesiyle, bizim aslında tam olarak onların nasıl işlediğini anlamaya ihtiyacımız yoktur. Klasik eğitilmiş bir bilimci bakış açısından, bir şeylerin anlaşılmadan oluşturulması neredeyse dinsel/toplumsal değerlere aykırı görüş ile eşdeğer anlam taşımaktadır.

İkinci neden ise sinir ağı modellerinin çok sayıda ve karmaşık olmasıdır. Geri yayılım modeli en çok kullanılan teknik olmasına rağmen sinir ağları analizi sadece bir teknikten oluşmamaktadır. Oldukça fazla sayıda sinirsel modeller bulunmaktadır. Bir insan tüm hayatını alternatif ağlarla deney yaparak, onları optimize etmeye çalışarak ve farklı parametrelerin etkilerini araştırarak geçirebilir.

Sinirsel modelleme bir sanat dalı ve sosyal bilimci ise işi hiç sonlanmayan bir sanatçı veya topluma sunduğu analiz sonuçlarının optimal olduğundan asla emin olamayan bir zanaatçı gibi görülebilir. Sinirsel modellerin çekiciliğinden etkilenerak işin içine dahil olan sosyal bilimci genellikle sonradan onun karmaşıklığı tarafından geri püskürtülmektedir.

Üçüncüsü, uygulama düzeyinde çoğu sosyal bilimci için araştırma metodolojisini oldukça sınırlayarak tanımlayan SAS, SPSS ve diğer standart istatistiksel paketlerin içersinde sinirsel analizi barındırmaları biraz yavaş olmuştur. Fakat son yıllarda sinir ağlarına olan ilginin önemli derecede artmış olmasının da etkisiyle bu durum değişmeye başlamıştır. Şu anda SPSS ve SAS içinde de sinirsel model oluşturulabilmektedir.

Dördüncüsü, belki diğerlerine göre sosyal bilimlerde sinirsel modellemenin yayılması önünde daha az öneme sahip bir engel olarak görülebilecek neden, sosyal bilime uzak alanlarda ortaya çıkan metotlar olması gerçeğidir. Sinir ağları literatürünü okumaya başlayan sosyal bilimci oldukça farklı bir teknik dil ile karşılaşmaktadır.

1. Gözlemlere veya olaylara “desen” denilmektedir.
2. Değişkenlere “özellik” denilmektedir.
3. Bağımsız değişkenlere “girdiler” denilmektedir.
4. Bağımlı değişkenlere “hedefler” veya “çıktılar” denilmektedir.
5. Artıklara “hatalar” denilmektedir.
6. Tahminlemeye eğitim, öğrenme veya kendini ayarlama (self organization) denilmektedir.
7. Geçerliliğe (Validation) ise “genelleştirme” denilmektedir.



Ayrıca arařtırmacı regresyon ve diskriminant analizi ile gözetmenli öğrenme arasında, ana bileşenler analizi ile gözetmensiz öğrenme arasında ve rekabetçi öğrenme ile sınıflandırma analizi arasında bir bağlantı kurulmalıdır. Bu ve diğer terminolojik problemler sinir ağıları analizinde yeni olan sosyal bilimci için zorluklar çıkarmaktadır.

## 2. BÖLÜM

### GERİ YAYILIM ALGORİTMASININ TÜRETİLMESİ

#### 2.1. Nöron Modeli ve Ağ Mimarisi

Birinci bölümde basitleştirilmiş biyolojik nöronun ve sinir ağlarının genel yapısı ele alındıktan sonra bu bölümde nöronların matematiksel modeli tanıtılacak ve bu yapay nöronların ağ mimarilerini oluşturmak amacıyla nasıl bir araya geldikleri anlatılacaktır.

##### 2.1.1. Notasyon

YSA 'nın yeni bir yöntem olmasından ötürü standart matematiksel notasyon ve mimari gösterimde henüz ortak bir noktaya varılamamıştır. Buna ek olarak sinir ağları üzerinde yapılan çalışma ve kitaplar mühendislik, fizik, psikoloji ve matematik gibi farklı alanlarda ortaya çıkmakta ve yazarlar kendi uzmanlık alanlarına özgü kelimeleri kullanma eğilimi göstermektedirler. Dolayısıyla YSA alanı içerisinde yazılan kitap ve makalelerde ortak bir dil ve notasyon kullanılmadığından okuma zorluğu ortaya çıkmakta ve kullanılan kavramlar onları gerçekte olduğundan daha karmaşık görünmesini sağlamaktadır. Bu durum ise bilinen kavramların yeniden ortaya konulmasından da öte önemli yeni düşüncelerin yayılmasını engellemektedir.

Bu çalışma içerisinde, uygulamanın gerçekleştirileceği Matlab programı ile farklılık olmaması açısından oradaki notasyonun paralelinde gidilmiştir. Notasyon ile ilgili ek açıklamalar ihtiyaç duyuldukça yapılacaktır. Matematiksel eşitlikler, şekiller ve metin içerisinde aşağıdaki notasyon kullanılacaktır:

Skalerler –  $a, b, c$  gibi küçük italik harfler

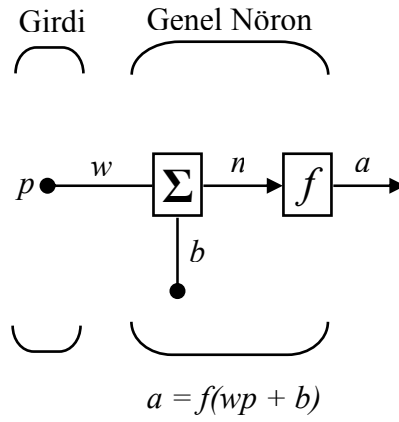
Vektörler –  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  gibi italik olmayan kalın küçük harfler

Matrisler –  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  gibi italik olmayan kalın büyük harfler

## 2.1.2. Nöron Modeli

### 2.1.2.1. Tek Girdili Nöron

Şekil 2-1 de gösterilen tek girdili nöron modelinde skaler  $p$  girdisi skaler  $w$  ağırlığıyla çarpılarak toplama işlemine gönderilecek  $wp$  'yi oluştururlar. Toplamcıya gönderilecek bir diğer girdi sapma (bias) değeriyle çarpılacak olan 1 dir. Genellikle net girdi olarak adlandırılan toplamcı çıktısı  $n$ , nöronun skaler çıktısı  $a$  'yı üretecek olan aktivasyon (transfer) fonksiyonu içersine gönderilir.



**Şekil 2- 1: Tek Girdili Nöron**

Nöronun çıktısı şu şekilde hesaplanır:

$$a = f(wp + b) \quad (2.1)$$

Çıktı değeri bir sonraki kısımda ele alınan seçilmiş olan transfer fonksiyonuna bağlıdır. Transfer fonksiyonunun net girdisi  $n$ , ağırlıklandırılmış girdi  $wp$  ve sapma değeri  $b$  nin toplamına eşittir.

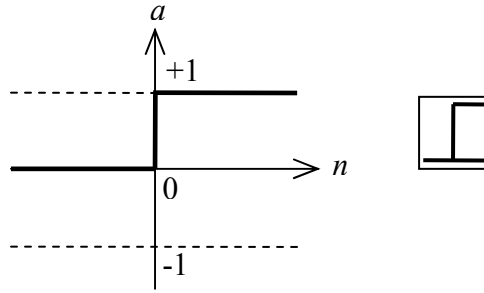
Nöronun ayarlanabilir skaler parametreleri  $w$  ve  $b$  dir. Sinir ağlarının ana düşüncesi, ağın istenilen veya ilgilenilen davranışı sergilemesi amacıyla bu parametrelerin değiştirilmesine dayanmaktadır. Böylece ağırlık ve sapma parametrelerinin ayarlanmasıyla ağ belirli bir görevi yerine getirmesi için eğitilebilir.

### 2.1.2.2. Transfer Fonksiyonları

Transfer fonksiyonu  $n$  'in doğrusal veya doğrusal olmayan bir fonksiyonu olabilmektedir. Nöronun çözmeye çalıştığı problemin bazı özelliklerinin karşılanması amacıyla belirli bir transfer fonksiyonu seçilebilmektedir. Çok sayıda transfer fonksiyonu bulunmaktadır, bunlardan en çok kullanılan üç tanesi aşağıda verilmiştir.

#### a) Sınırlayıcı Transfer Fonksiyonu

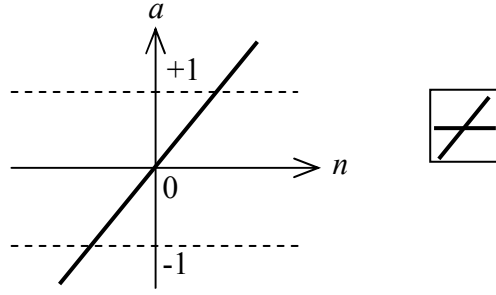
Şekil 2-2 'de gösterilen sınırlayıcı transfer fonksiyonu eğer  $n$  net girdi değeri 0 'dan küçük ise nöronun çıktı değerini 0 ve eğer net girdi değeri 0 'a eşit veya daha büyükse nöronun çıktı değerini 1 olarak sınırlayacaktır. Genellikle girdileri iki farklı kategoriye ayıran nöronların yaratılmasında kullanılmaktadır. Şeklin yanındaki ikonun içerisinde Sınırlayıcı transfer fonksiyonunun görüntüsü bulunmaktadır. Bu ağ diyagramlarındaki  $f$  'in yerine yerleştirilecek böylece hangi transfer fonksiyonunun kullanılmakta olduğu anlaşılacaktır.



Şekil 2- 2: Sınırlayıcı Transfer Fonksiyonu

#### b) Doğrusal Transfer Fonksiyonu

Aşağıdaki şekilde gösterilen doğrusal transfer fonksiyonunun çıktısı  $a$  onun girdisine  $n$  eşittir. Doğrusal yaklaşımlar için kullanılan nöronlarda bu çeşit fonksiyon kullanılmaktadır.

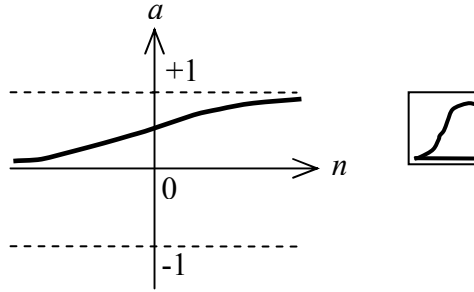


**Şekil 2- 3: Doğrusal Transfer Fonksiyonu**

### c) Sigmoid Transfer Fonksiyonu

Tanıtılacak olan diğer bir transfer fonksiyonu eksi ve artı sonsuz arasındaki herhangi bir girdi değerini alabilen ve çıktısını aşağıdaki ifadeye göre 0 ve 1 aralığı içerisine sıkıştıran Sigmoid fonksiyonudur.

$$a = \frac{1}{1 + e^{-n}} \quad (2.2)$$

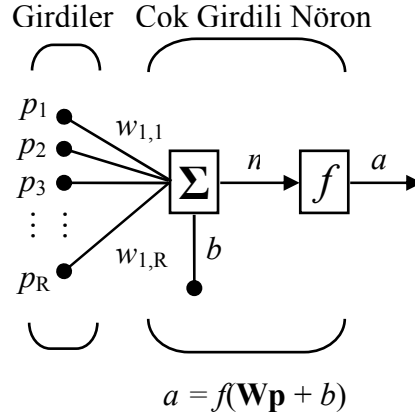


**Şekil 2- 4: Log-Sigmoid Transfer Fonksiyonu**

Sigmoid transfer fonksiyonu türevi alınabilir olmasından ötürü genellikle geri yayılım algoritması kullanılarak eğitilen çok katmanlı ağlarda kullanılmaktadır.

### 2.1.2.3. Çok Girdili Nöron

Genellikle bir nöronun birden çok girdisi vardır. Şekil 2-5 de bir nöron R tane girdi elemanı ile gösterilmiştir.  $p_1, p_2, \dots, p_R$  tekil girdileri  $\mathbf{W}$  ağırlık matrisinin bunlara karşılık gelen her bir  $w_{1,1}, w_{1,2}, \dots, w_{1,R}$  ağırlık elemanı ile ağırlıklandırılmıştır.



Şekil 2- 5: Çok Girdili Nöron

Bir  $b$  sapma değerine sahip nöron  $n$  net girdisinin elde edilmesi amacıyla ağırlıklandırılmış girdilerle beraber toplanır:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b \quad (2.3)$$

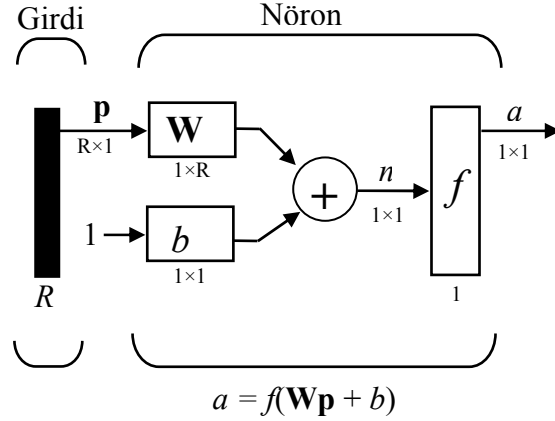
Bu ifade matris formatında yazılırsa:

$$n = \mathbf{W}\mathbf{p} + b \quad (2.4)$$

Burada tek nöron olmasından ötürü  $\mathbf{W}$  matrisi sadece tek bir satıra sahiptir.

Ağırlık matrisinin elemanlarının indisleri belirli bir amaç doğrultusunda atanmıştır. İlk indeks ağırlığın varış noktası olan nöronu gösterir. İkinci indeks o nöronu besleyen sinyalin kaynağını göstermektedir. Örneğin  $w_{1,2}$  indisi ikinci kaynaktan ilk nörona olan bağlantıyı göstermektedir. Bu kural birden fazla nöron olması durumunda oldukça yararlı olmaktadır.

Yukarıda gösterilen tek bir nöronun şeklinde birçok ayrıntı bulunmaktadır. Birden çok katmanı ve birden çok hücresi bulunan bir ağı düşündüğümüzde ana düşüncenin kaybolmasına yol açacak çok fazla detay ortaya çıkacaktır. Bunu gidermek için aşağıdaki gibi kısaltılmış notasyon kullanılır.



**Şekil 2- 6: Kısaltılmış Notasyonla R Girdili Nöron**

Şekil 2-6 da siyah dikey çubukla  $\mathbf{p}$  girdi vektörü gösterilmiştir.  $\mathbf{p}$  sembolünün altında girdi vektörünün boyutu  $R \times 1$  olarak verilmiştir. Bu girdiler tek nöron olmasından ötürü tek bir satıra ve  $R$  tane sütuna sahip  $\mathbf{W}$  ağırlık matrisine gitmektedir. Sabit 1 değeri nörona bir girdi olarak girmekte ve  $b$  skaler sapma değeriyle çarpılmaktadır.  $f$  transfer fonksiyonunun net girdisi  $n$ ,  $\mathbf{W}\mathbf{p}$  çarpımı ve sapma  $b$ ' nin toplamından oluşmaktadır. Burada bir nörona sahip olduğundan dolayı ağın çıktı değeri  $a$ , bir skalerdir. Birden fazla nörona sahip olunması durumunda ağ çıktısı bir vektör olacaktır. Her ne zaman bu kısaltılmış ağ notasyonu kullanılırsa matrislerin boyutları matris değişkenlerinin altında gösterilir. Böylece mimarinin anlaşılması kolaylaşacak ve bunlarla ilişkilendirilmiş matris matematiğinin takip edilmesinde zorluk yaşanmayacaktır.

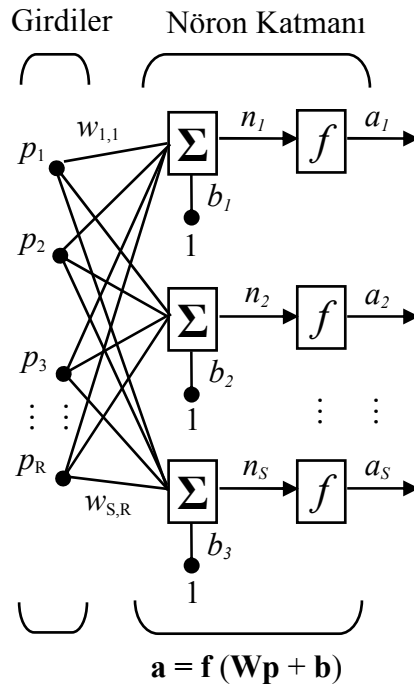
Ağın girdi sayısı ele alınan problemin özellikleri tarafından belirlenmektedir. Örneğin uçuş koşullarını tahmin eden bir ağ tasarlanmak isteniyorsa, girdiler; hava sıcaklığı, rüzgarın hızı ve nem oranından oluşacak ve bu şekilde ağın üç girdisi olacaktır.

### 2.1.3. Ağ Mimarisi

Genellikle çok sayıda girdiye sahip bir nöronla çalışmak yetersiz kalacaktır. İki veya daha fazla sayıda nöron paralel çalışacak şekilde bir katman içerisinde birleştirilebilir ve birden çok katmanı olan ağlar da oluşturulabilir. İlk olarak tek katman ele alınacaktır.

#### 2.1.3.1. Tek Katmanlı Nöronlar

$R$  tane girdi elemanı ve  $S$  nörondan oluşan tek katmanlı bir ağ Şekil 2-7 'de gösterilmiştir. Burada  $R$  tane girdi her bir nörona bağlıdır ve artık ağırlık matrisi  $S$  satıra sahiptir.



Şekil 2- 7:  $S$  Nöronlu Katman

Bir katmanı oluşturan elemanlar sırasıyla ağırlık matrisi, toplayıcılar,  $\mathbf{b}$  sapma vektörü, transfer fonksiyon kutuları ve  $\mathbf{a}$  çıktı vektörüdür. Bazı yazarlar girdileri ayrı bir katman olarak görmektedir, fakat burada böyle bir ayrıma gidilmeyecektir. Genellikle bir katmandaki girdilerin sayısı o katmandaki hücrelerin sayısından farklı



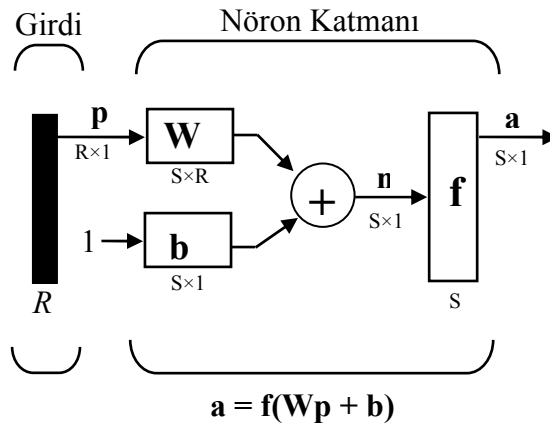
olmaktadır ( $R \neq S$ ). Bir katman içerisindeki tüm nöronların aynı transfer fonksiyonuna sahip olması gibi bir zorunluluk bulunmamaktadır. Farklı transfer fonksiyonlarına sahip nöronlar bir ağ içerisinde kullanılabilir.

Girdi vektörünün elemanları ağ içerisine  $\mathbf{W}$  ağırlık matrisi aracılığıyla girer:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad (2.5)$$

Daha önce bahsedildiği gibi  $\mathbf{W}$  matrisinin elemanlarını gösteren satır indisi ağırlığın varış noktası olan nöronu gösterirken sütun indisi o nöronu besleyen sinyalin kaynağını göstermektedir. Böylece  $w_{3,2}$  üçüncü nörona ikinci kaynaktan olan bağlantıyı temsil etmektedir.

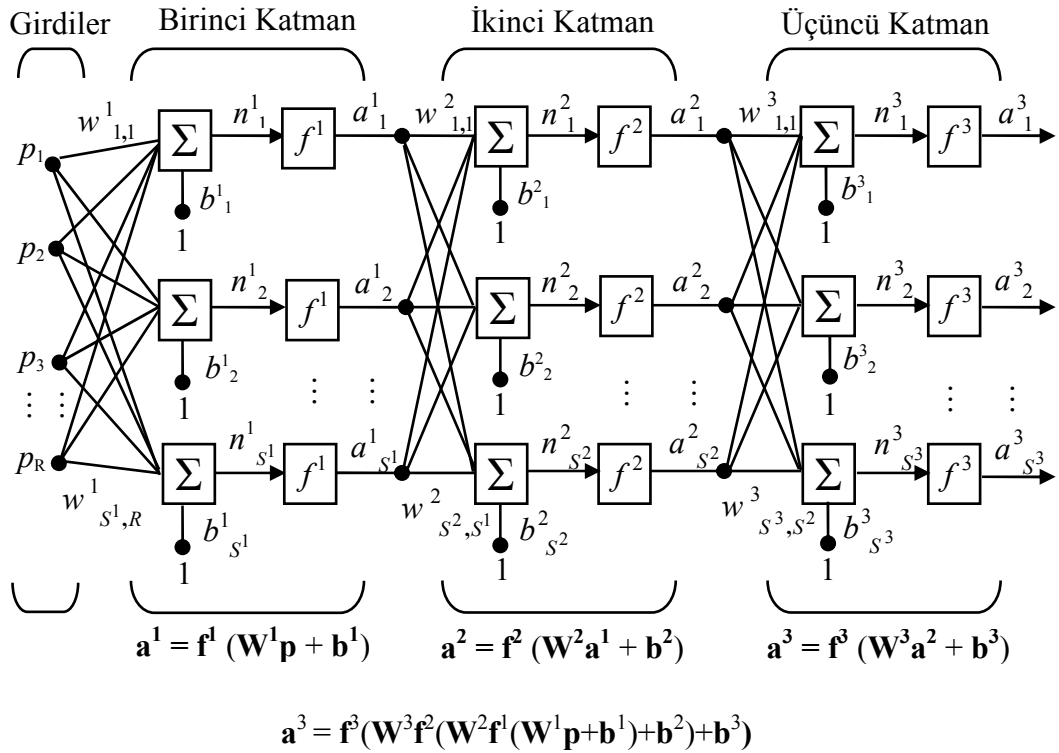
Aşağıdaki şekil  $S$  nöronlu  $R$  girdili ve tek katmanlı bir ağın kısaltılmış notasyonla çizilmiş halini göstermektedir.



**Şekil 2- 8: Kısaltılmış Notasyonla S Nöronlu Katman**

### 2.1.3.2. Çok Katmanlı Nöronlar

Bir ağ birden çok katmana sahip olabilir. Her katmanın kendine ait ağırlık matrisi, sapma vektörü, net girdi vektörü ve çıktı vektörü bulunmaktadır. Bu ağırlık matrislerini, çıktı vektörlerini vb. katmanlar bazında ayırt etmek için ilgili değişkenin ismine katman numarasını gösteren bir üst simge eklenir. Aşağıdaki şekilde üç katmanlı ağ içerisinde bu katman notasyonunun kullanımı görülmektedir.

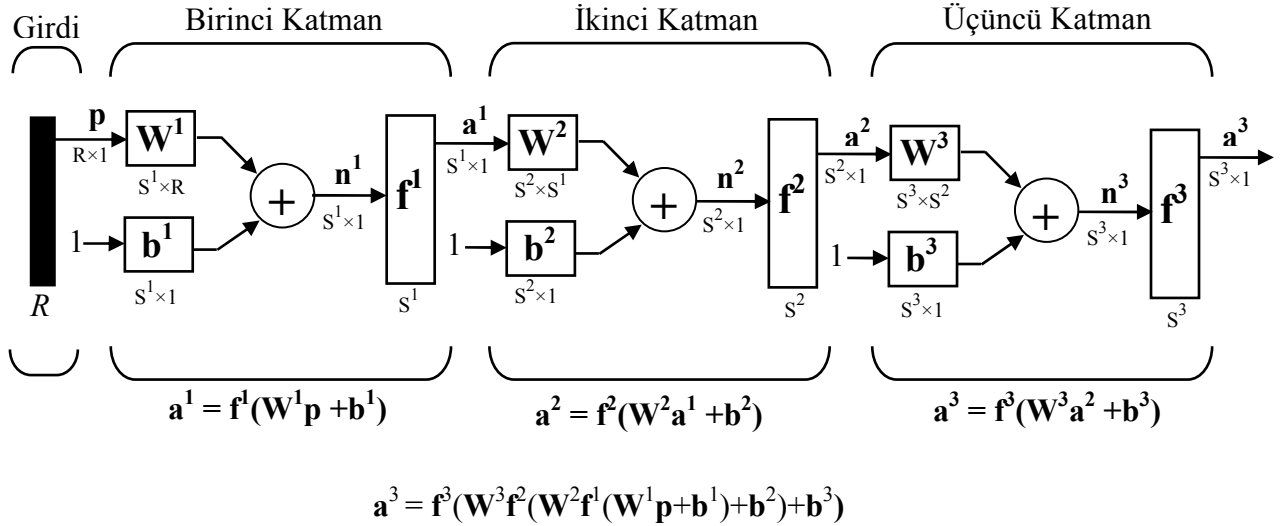


Şekil 2- 9: Üç Katmanlı Ağ

Görüldüğü gibi  $R$  tane girdi, ilk katmanda  $S^1$  nöron, ikinci katmanda  $S^2$  nöron bulunmaktadır. Farklı katmanlar farklı sayıda nöronlara sahip olabilir. Bir veya iki katmanın çıktıları yine bir veya iki katmanın girdileri olabilmektedir. Böylece ikinci katmana  $R=S^l$  girdili,  $S=S^2$  nöronlu ve  $S^2 \times S^l$  boyutlu  $W^2$  ağırlık matrisine sahip tek katmanlı bir ağ gözüyle bakılabilir. İkinci katmana girdi  $\mathbf{a}^1$  ve çıktı  $\mathbf{a}^2$  dir.

Çıktı değeri ağın çıktısı olan katman çıktı katmanı olarak adlandırılır. Diğer tüm katmanlara ise gizli katman denilmektedir. Yukarıdaki gibi üç katmanlı bir ağ

bir tane çıktı katmanına (üçüncü katman) ve iki tane gizli katmana sahiptir (birinci ve ikinci katman). Şekil 2-10 da yukarıda tartışılan ağın kısaltılmış notasyonla çizilmiş hali gösterilmektedir.



**Şekil 2- 10: Kısaltılmış Notasyonla Üç Katmanlı Ağ**

Çok katmanlı ağlar tek katmanlı ağlara göre oldukça güçlüdür. Örneğin birinci katmanında Sigmoid ve ikinci katmanında doğrusal fonksiyona sahip iki katmanlı bir ağ çoğu fonksiyona keyfi derecede yaklaşım için eğitilebilir. Tek katmanlı ağlar bunu yapamamaktadır (Hagan, Demuth ve Beale, 1996:2-12).

Bir ağ kurulurken oluşturulacak parametrelerin sayısının belirlenmesi biraz kafa karıştırıcı gibi görülebilir. Bu sorunu ele almak için ilk olarak ağın girdilerinin ve çıktılarının sayısının incelenen problemin özellikleri tarafından belirlendiğini hatırlayalım. Böylece girdi olarak kullanılacak dört tane dış değişken varsa ağında dört girdisi olacaktır. Benzer şekilde ağın yedi çıktısı varsa çıktı katmanında yedi nöron olması zorunluluğu vardır. Ayrıca çıktı sinyalinin istenilen karakteristiği çıktı katmanındaki transfer fonksiyonunun seçiminde yardımcı olacaktır. Eğer çıktı -1 veya 1 lerden oluşacaksa, simetrik sınırlayıcı transfer fonksiyonu kullanılmalıdır. Bu şekilde tek katmanlı bir ağın mimarisi, girdi ve çıktılarının kesin bir sayısı ve belirli

çıktı sinyali karakteristiğini kapsayacak şekilde neredeyse tümüyle ele alınan problemin özellikleri tarafından belirlenmektedir.

İki katmandan daha fazlasına sahipsek durum biraz farklılık göstermektedir. Bu durumda söz konusu problem, gizli katmanlarda kaç tane nöronun gerekliliği olduğunu direkt olarak bize söylemeyecektir. Gizli katmanda optimal kaç nöronun bulunmasına ihtiyaç duyulduğu konusunda kesin belirlenmiş bir kural yoktur. Gizli katmandaki nöron sayısını belirlemek için yapılan, deneme yanılmalar sonucunda en iyi uyum sağlayan ağ tasarımının bulunmasıdır.

Katman sayısının belirlenmesi ise ağ oluşturmadaki başka bir probleme işaret etmektedir. Çoğu uygulamalı sinir ağı çalışmaları sadece iki veya üç katmana sahiptir. Dört veya daha fazla katmanlı ağlar nadiren kullanılmaktadır.

Son olarak sapma kullanımı hakkında birkaç şey söylemek gerekirse, istenirse sapmalı ve sapmasız nöronlar oluşturulabilir. Sapmalar ağa fazladan bir değişkene sahip olma imkanı vermekte ve böylece sapmalara sahip bir ağın sapmalara sahip olmayan bir ağdan daha güçlü olması beklenmektedir. Bazı örnekler sapma değeri kullanılmadan yapılmaktadır. Bazen de üç veya daha fazla değişkenin gösterimi zor olduğundan bilerek değişken sayısının düşürülmesi amacıyla sapmalar için içersine dahil edilmez.

## **2.2. Widrow-Hoff Öğrenme Kuralı**

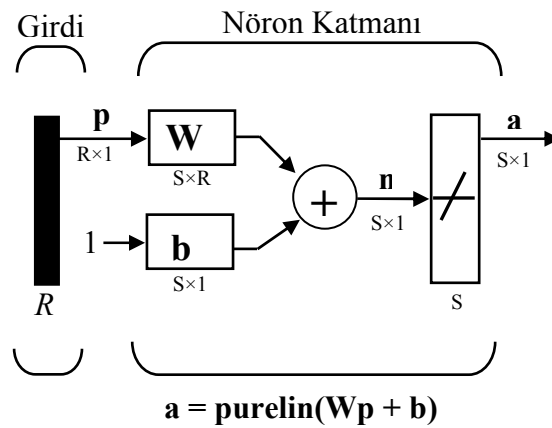
Oluşturulan ilk sinir ağı modeli algılayıcıda (perceptron) eğitim setindeki veriler doğrusal olarak ayrılabilme koşuluyla tam olarak doğru sınıflandırıcaya kadar eğitime devam edilmektedir. Hataya izin verilmemekte, bu ise yöntemin geniş bir uygulama alanı bulamamasına sebep olmaktadır. Algılayıcının bu dezavantajını ortadan kaldıracak yöntem Widrow-Hoff öğrenme algoritmasıdır. (Abdi, Valentin ve Edelman, 1999:9). Aynı zamanda Widrow-Hoff öğrenme günümüzde en çok kullanılan algoritma olan geri yayılım algoritmasının da temelini oluşturmaktadır. Bu sebepten ötürü öncelikle Widrow-Hoff öğrenme algoritması tanıtılacak ve buradan hareketle geri yayılım algoritmasına geçiş yapılacaktır.

Bernard Widrow, Frank Rosenblatt'ın algılayıcı öğrenme kuralını geliştirmesiyle yaklaşık aynı dönemde 1950 'lerin sonlarında sinir ağları konusunda çalışmaya başlamıştır. 1960 'da Widrow ve onun yüksek lisans öğrencisi Marcian Hoff, Adaline (Adaptive Linear Neuron) ağını ve en küçük ortalama kare (LMS- Least Mean Square) algoritması olarak adlandırdıkları öğrenme kuralını tanıtmışlardır.

Adaline, transfer fonksiyonunda sınırlayıcının değil de doğrusal fonksiyonun kullanılması dışında algılayıcıya çok benzemektedir. Adaline ve algılayıcının her ikisi de yalnızca doğrusal olarak ayrılabilen problemleri çözebilmektedir. Bununla beraber LMS algoritması, algılayıcı öğrenme kuralından daha güçlüdür. Algılayıcı eğitim setindeki desenlerin doğru olarak sınıflandırılmasını sağlarken gürültülere karşı oldukça hassas olabilmektedir. LMS algoritması ortalama karesel hatayı minimize ettiğinden gürültüye karşı daha duyarlıdır ve algılayıcıya göre özellikle dijital sinyal işleme alanı gibi çok daha fazla kullanım yeri bulmuştur.

### 2.2.1. Adaline Ağı

Adaline ağı Şekil 2-11 de gösterilmiştir. Farklılık doğrusal bir transfer fonksiyonuna sahip olmasından kaynaklanmaktadır.



Şekil 2- 11: Adaline Ağı

Ağın çıktısı şöyle olacaktır:

$$\mathbf{a} = \text{purelin}(\mathbf{W}\mathbf{p} + \mathbf{b}) = \mathbf{W}\mathbf{p} + \mathbf{b} \quad (2.6)$$

Ağın çıktı vektörünün  $i$  inci elemanını şu şekilde yazılabilir:

$$a_i = \text{purelin}(n_i) = \text{purelin}({}_i\mathbf{w}^T \mathbf{p} + b_i) = {}_i\mathbf{w}^T \mathbf{p} + b_i \quad (2.7)$$

Burada  ${}_i\mathbf{w}$ ,  $\mathbf{W}$  matrisinin  $i$  inci satırının elemanlarından oluşturulmuştur.

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix} \quad (2.8)$$

### 2.2.2. Ortalama Karesel Hata

Adaline ağının genel şeklini gördükten sonra artık LMS algoritmasının geliştirilmesine başlanabilir. LMS algoritması, ağın doğru şekilde davranmasını sağlayacak örnekler setinin verildiği bir gözetmenli veya denetmenli, danışmanlı eğitim türüdür.

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \quad (2.9)$$

Burada  $\mathbf{p}_Q$ , ağa sunulan girdi ve  $\mathbf{t}_Q$  ise ona karşılık gelen hedef çıktı değeridir. Her girdi ağa uygulandıkça ağın çıktı değeri hedef değer ile karşılaştırılmaktadır.

LMS algoritması ağın çıktı değerleri ve hedef değerleri arasındaki farklılık olan hataların karelerinin ortalamasını en küçük yapacak şekilde ağın ağırlık ve sapma değerlerini ayarlayacaktır. Bu kısımda performans indeksi üzerinde durulacak ve ilk olarak tek nöron durumu ele alınacaktır. Performans indeksi ağın performansının sayısal bir ölçüsüdür. Ağın performansı iyiyken performans indeksi küçük ve ağın performansı kötüyken ise performans indeksi büyük bir değerdir. Amaç performans indeksini düşürmek amacıyla parametre uzayında araştırma (ağın

ağırlık ve sapma değerlerinin ayarlanması) yapmaktır. Bu sürece bir optimizasyon gözüyle bakmak mümkündür.

Basitlik sağlanması açısından sapmalarda dahil olmak üzere ayarlamaya çalıştığımız tüm parametreleri tek bir vektör içersine yerleştirelim.

$$\mathbf{x} = \begin{pmatrix} {}_1\mathbf{w} \\ b \end{pmatrix} \quad (2.10)$$

Benzer şekilde, sapmanın girdi değerini 1 olarak girdi vektörünün bir bileşeni olarak görebiliriz.

$$\mathbf{z} = \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} \quad (2.11)$$

Artık genellikle (2.12) deki gibi ifade edilen ağın çıktısı 2.13 daki gibi ifade edilebilir.

$$a = {}_1\mathbf{w}^T \mathbf{p} + b \quad (2.12)$$

$$a = \mathbf{x}^T \mathbf{z} \quad (2.13)$$

Bu bize ortalama karesel hata ifadesini daha kullanışlı şekilde yazma imkanını sağlamaktadır.

$$F(\mathbf{x}) = E [ e^2 ] = E [ (t - a)^2 ] = E [ (t - \mathbf{x}^T \mathbf{z})^2 ] \quad (2.14)$$

Burada beklenen değer tüm girdi/hedef setleri üzerinden alınmıştır. Bu ifadeyi aşağıdaki şekilde genişletebiliriz.

$$F(\mathbf{x}) = E [ t^2 - 2t\mathbf{x}^T \mathbf{z} + \mathbf{x}^T \mathbf{z} \mathbf{z}^T \mathbf{x} ] \quad (2.15)$$

$$= E [ t^2 ] - 2\mathbf{x}^T E [ t\mathbf{z} ] + \mathbf{x}^T E [ \mathbf{z} \mathbf{z}^T ] \mathbf{x}$$

(2.15) deki ifade ařağıdaki řekilde yazılabilir.

$$F(\mathbf{x}) = c - 2\mathbf{x}^T \mathbf{h} + \mathbf{x}^T \mathbf{R} \mathbf{x} \quad (2.16)$$

Burada

$$c = E [t^2], \quad \mathbf{h} = E [t\mathbf{z}] \quad \text{ve} \quad \mathbf{R} = E [\mathbf{z}\mathbf{z}^T] \quad \text{'dir.}$$

$\mathbf{h}$  vektörü, girdi vektörü ve onunla ilişkilendirilmiş hedef arasındaki çapraz-korelasyonu (cross-correlation) verirken  $\mathbf{R}$  ise girdi korelasyon matrisidir. Bu matrisin köşegen elemanları girdi vektörünün elemanlarının ortalama karesel değerlerine eşittir.

(2.16) eşitliğini ařağıda (2.17) eşitliğıyle verilen kuadratik fonksiyonun genel formu ile karşılařtıralım.

$$F(\mathbf{x}) = c - 2\mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (2.17)$$

Sonuç, Adaline ağı için ortalama hata karenin performans indeksi bir kuadratik fonksiyondur.

Burada

$$\mathbf{d} = -2\mathbf{h} \quad \text{ve} \quad \mathbf{A} = 2\mathbf{R} \quad \text{'dir.}$$

Bu çok önemli bir sonuçtur çünkü bilindiğı üzere kuadratik fonksiyonun özellikleri temel olarak  $\mathbf{A}$  Hessian matrisine bağıdır (Strang, 1988:335). Örneğın Hessian matrisinin tüm özdeğerleri pozitif ise bu takdirde fonksiyon tek bir global minimum değerine sahip olacaktır.

Tüm korelasyon matrislerinin pozitif tanımlı veya pozitif yarı tanımlı olduğı gösterilebilir (Ertař, 2007). Bunun anlamı hiçbir zaman negatif özdeğerlerinin olmayacağıdır. Elimizdeki Hessian matrisi korelasyon matrisinin iki katı olduğundan



iki olasılık bulunmaktadır. Eğer korelasyon matrisi sadece pozitif özdeğerlere sahipse performans indeksi tek bir global minimuma sahiptir. Eğer korelasyon matrisinin bazı özdeğerleri sıfırsa  $\mathbf{d} = -2\mathbf{h}$  vektörüne bağlı olarak performans indeksi ya güçsüz minimuma (weak minimum) sahip olacak yada performans indeksinin minimum değeri olmayacaktır.

Şimdi performans indeksinin denge noktasına bakılırsa kuadratik fonksiyonun gradientinin aşağıdaki gibi olduğu görülür.

$$\nabla F(\mathbf{x}) = \nabla (c + \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}) = \mathbf{d} + \mathbf{A} \mathbf{x} = -2\mathbf{h} + 2\mathbf{R} \mathbf{x} \quad (2.18)$$

$F(\mathbf{x})$  'in denge noktası gradientin 0 a eşitlenmesiyle bulunabilir.

$$-2\mathbf{h} + 2\mathbf{R} \mathbf{x} = 0 \quad (2.19)$$

Böylece korelasyon matrisi pozitif tanımlıysa güçlü minimuma sahip tek bir denge noktası olacaktır.

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{h} \quad (2.20)$$

Dikkat edilirse tek bir çözümün varlığı sadece korelasyon matrisine bağlıdır. Böylece girdi vektörünün karakteristiği tek bir çözümün olup olmadığının belirleyicisidir.

### 2.2.3. LMS Algoritması

Performans indeksini analiz ettikten sonra bir sonraki adım minimum noktaya ulaşmak için bir algoritma geliştirilmesidir. Eğer istatistiksel  $\mathbf{h}$  ve  $\mathbf{R}$  değerleri hesaplayabilirsek minimum noktayı direkt olarak eşitlik (2.20) 'den hesaplayabiliriz.  $\mathbf{R}$  'nin tersini hesaplamak istemiyorsak eşitlik (2.18) 'de gradienti verilen eğim iniş (steepest descent) algoritmasını kullanabiliriz. Bununla birlikte genellikle  $\mathbf{h}$  ve  $\mathbf{R}$  'nin hesaplanması ya istenmez yada mümkün değildir. Bu nedenden ötürü tahminlenen gradientleri kullanan yaklaşık eğim iniş algoritması kullanılır.

Anahtar nokta  $F(\mathbf{x})$  ortalama hata karelerin  $\hat{F}(\mathbf{x})$  tarafından tahmin edilebilir olmasıdır.

$$\hat{F}(\mathbf{x}) = (t(k) - a(k))^2 = e^2(k) \quad (2.21)$$

Burada karesel hatanın beklenen değeri,  $k$  inci iterasyondaki karesel hata ile yer değiştirmiştir. Daha sonra her iterasyonda aşağıdaki gibi gradientin tahminine erişilir.

$$\hat{\nabla}F(\mathbf{x}) = \nabla e^2(k) \quad (2.22)$$

$\nabla e^2(k)$  'nin ilk  $R$  elemanı ağırlıklara göre türevlerinden oluşuyorken  $(R+1)$  inci eleman artığa göre türevdir. Böylece

$$\left[ \nabla e^2(k) \right]_j = \frac{\partial e^2(k)}{\partial w_{1,j}} = 2e(k) \frac{\partial e(k)}{\partial w_{1,j}} \quad \text{için } j = 1, 2, \dots, R \quad (2.23)$$

Ve

$$\left[ \nabla e^2(k) \right]_{R+1} = \frac{\partial e^2(k)}{\partial b} = 2e(k) \frac{\partial e(k)}{\partial b} \quad (2.24)$$

Şimdi bu eşitliklerin sonundaki kısmi türev terimlerini ele alalım. İlk olarak  $e(k)$  'nin  $w_{1,j}$  ağırlığına göre kısmi türevine bakalım.

$$\begin{aligned} \frac{\partial e(k)}{\partial w_{1,j}} &= \frac{\partial [t(k) - a(k)]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}} [t(k) - (\mathbf{w}^T \mathbf{p}(k) + b)] \\ &= \frac{\partial}{\partial w_{1,j}} \left[ t(k) - \left( \sum_{i=1}^R w_{1,i} p_i(k) + b \right) \right] \end{aligned} \quad (2.25)$$

Burada  $p_i(k)$ , k inci iterasyonda girdi vektörünün i inci elemanıdır. Şu şekilde basitleştirebiliriz.

$$\frac{\partial e(k)}{\partial w_{i,j}} = -p_j(k) \quad (2.26)$$

Benzer şekilde gradientin son elemanını da yazalım.

$$\frac{\partial e(k)}{\partial b} = -1 \quad (2.27)$$

Dikkat edilirse  $p_j(k)$  ve 1,  $\mathbf{z}$  girdi vektörünün elemanlarıdır böylece k inci iterasyondaki karesel hatanın gradienti şu şekilde yazılabilir.

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{z}(k) \quad (2.28)$$

Artık eşitlik (2.21) 'de olduğu gibi k inci iterasyondaki tek bir hata ile ortalama karesel hataya yaklaşmanın yararı görülmektedir. Bu yaklaşık gradienti hesaplamak için tek ihtiyacımız olan girdi ile hatanın çarpılmasıdır.

Bu  $\nabla F(\mathbf{x})$  'e yaklaşım şimdi eğim iniş algoritması içerisinde kullanılabilir. Sabit öğrenme oranıyla eğim iniş algoritması aşağıdaki şekilde ifade edilir:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} \quad (2.29)$$

$\nabla F(\mathbf{x})$  değerini eşitlik (2.28) deki  $\hat{\nabla} F(\mathbf{x})$  değeriyle yer değiştirirsek

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k) \mathbf{z}(k) \quad (2.30)$$

Veya

$${}_1\mathbf{w}(k+1) = {}_1\mathbf{w}(k) + 2\alpha e(k) \mathbf{p}(k) \quad (2.31)$$

Ve

$$b(k+1) = b(k) + 2\alpha e(k) \quad (2.32)$$

Bu son iki eşitlik en küçük ortalama karesel hata (LMS) algoritmasını oluşturmaktadır. Bu algoritmaya aynı zamanda Delta Kuralı veya Widrow-Hoff öğrenme algoritması da denilmektedir (Demuth, Beale ve Hagan, 2006:4-13).

Önceki sonucu, çoklu çıktının yani çoklu nöronun olduğu durumu ele almak için düzenleyebiliriz. Ağırlık matrisinin i inci satırını güncellemek istiyorsak

$${}_i\mathbf{w}(k+1) = {}_i\mathbf{w}(k) + 2\alpha e_i(k) \mathbf{p}(k) \quad (2.33)$$

$e_i(k)$ , k inci iterasyondaki hatanın i inci elemanıdır. Sapmaların i inci elemanını güncellemek istiyorsak

$$b_i(k+1) = b_i(k) + 2\alpha e_i(k) \quad (2.34)$$

LSM algoritması matris notasyonuna uygun şekilde yazılabilir:

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} + 2\alpha \mathbf{e}(k) \mathbf{p}^T(k) \quad (2.35)$$

Ve

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha \mathbf{e}(k) \quad (2.36)$$

Artık  $\mathbf{e}$  hatası ve  $\mathbf{b}$  sapması vektör durumundadır.

## 2.3. Geri Yayılımın Oluşturulması

### 2.3.1. Giriş

Çok katmanlı ağları eğitmek için kullanılabilen, LMS algoritmasının genelleştirilmiş haline geri yayılım denmektedir. LMS öğrenme kuralında olduğu gibi geri yayılım, performans indeksinin ortalama karesel hata olduğu bir yaklaşık eğim iniş algoritmasıdır (approximate steepest descent). Geri yayılımın diğer bir

tanımı, gerçek ve ağıın çıktı değerleri arasındaki global hatayı minimize etmek amacıyla çok boyutlu ağırlık uzayında optimum noktayı arayan bir optimizasyon algoritması olmasıdır (Lu, AbouRizk ve Hermann, 2001:299). LMS algoritması ve geri yayılım arasındaki farklılık sadece türevlerin hesaplanma şeklidir. Tek katmanlı doğrusal ağda hata ağıın ağırlıklarının açık bir doğrusal fonksiyonu olmakta ve ağırlıklara göre türevler kolaylıkla hesaplanabilmektedir. Doğrusal olmayan transfer fonksiyonuna sahip çok katmanlı ağlarda, ağıın ağırlıkları ve hata arasındaki ilişki çok daha karmaşıktır. Türevlerin hesaplanması amacıyla matematikteki zincir kuralının kullanılmasına ihtiyaç duyulmaktadır.

Önceki sinir ağları modellere ilişkin ana problem onların doğrusal problemlerle ilgili olup sınırlandırılmasıydı. Araştırmacılar (McCulloch & Pitts, 1943) çok öncelerden bu kısıtlamanın üzerinden çıktı katmanı ve girdi katmanı arasına bir veya daha fazla gizli katman eklenilerek gelinebileceğini biliyorlardı. Gözden kaçırılan şey algılayıcı gibi bir öğrenme kuralının gizli katmanların hücrelerinin ağırlıklarını ayarlamak için kullanılabileceğiydi. Hata geri yayılımının da yaptığı şey budur. Hata geri yayılımının var olmaması 1960 ların sonlarında sinir ağlarına olan ilginin azalmasının muhtemel nedenlerinden biriydi ve varlığı 1980 lerde sinir ağlarına olan ilginin yeniden canlanmasına katkıda bulunmuştur (Hagan, Demuth ve Beale, 1996:11-2).

Çok katmanlı ağları eğitmek amacıyla ilk algoritma tanımı 1974 'de Paul Werbos 'un tezinde verilmiştir. Tez sinir ağlarının özel bir durum olarak ele alındığı genel ağlar bağlamında algoritma sunmuş ve sinir ağları camiasında yaygınlık kazanamamıştır. 1980 lerin ortalarında geri yayılım algoritması yeniden keşfedilmiş ve geniş olarak tanınmıştır (Haykin, 1994:37). Birbirlerinden bağımsız olarak David Rumelhart, Geoffrey Hinton ve Ronald Williams (1986), David Parker (1985), ve Yann Le Cun (1985) tarafından yeniden keşfedilmiştir. Algoritmaya psikolog David Rumelhart ve James McClelland öncülüğünde oluşturulan Paralel Distributed Processing (1986) kitabında yer verilmesiyle popülerlik kazanmıştır. Bu kitabın yayınlanması sinir ağları alanında çok sayıda araştırmanın yapılmasını teşvik etmiştir. Geri yayılım algoritmasıyla eğitilen çok katmanlı ağlar şu anda en çok

kullanılan sinir ağı türü olmaktadır. Desen çiftlerini ilişkilendirmek için herhangi bir ileri beslemeli ağda ağırlık değerlerinin ayarlanması işini yürütmektedir.

Geri yayılım ağları doğrusal olmayan birimlerden oluşmuş çok katmanlı ağlardır (aslında sadece gizli katmanın doğrusal olmaması yeterlidir). Doğrusal bir birim gibi, doğrusal olmayan birim aldığı tüm ağırlıklandırılmış aktivasyonları toplayarak kendi aktivasyon düzeyini hesaplar. Bununla beraber doğrusal birimden farklı olarak, doğrusal olmayan bir birim aktivasyonunu doğrusal olmayan transfer fonksiyonu aracılığıyla bir cevaba dönüştürür. Geri yayılım ağlarının amacı girdi-çıkı desenleri çiftleri arasındaki doğrusal olmayan haritalamayı öğrenmektir. Bu ağlar desen sınıflandırıcı olarak veya daha genellikle doğrusal olmayan problemleri çözmek için kullanılabilirler.

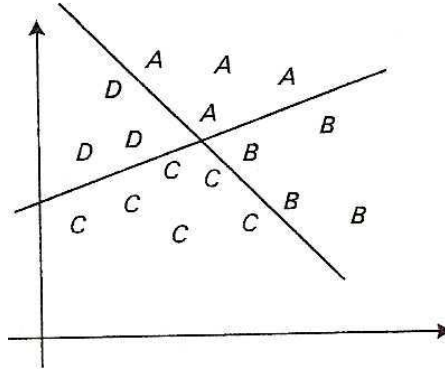
Bir geri yayılım ağı bazı yönlerden algılayıcıya benzemektedir bazı yönlerden ise farklılıklar göstermektedir. Algılayıcı gibi, geri yayılım ağı gözetmenli/denetimli öğrenmeyi kullanır: Bir çıktı biriminin cevabı ve beklenen cevap arasındaki farklılık ağ tarafından yapılan hatadır. Bu bağlantı ağırlıklarının düzeltilmesine temel teşkil etmektedir. Çıktı katmanının hücreleri bu hatayı bağlantı ağırlıklarını düzeltmede direkt olarak kullanır. Gizli katman hücreleri hata ile doğrudan bağlantılı durumda değildir. Onlar hatalarını tahmin etme gereği duyarlar. Bunu yapmak için de hata geri yayılımını kullanırlar. Ağ tarafından yapılan hata miktarı ilk olarak doğrusal olmayan transfer fonksiyonunun değişim oranı (eğim veya türev) orantısında bir hata sinyali dönüştürülürler. Bu hata sinyali daha sonra bağlantı ağırlıkları aracılığıyla gizli katman birimlerine geriye gönderilir (geri yayılır). Gizli birimler bağlı olduğu çıktı birimlerinden aldığı hata sinyallerinin ağırlıklandırılmış bir toplamı olarak kendi hatalarını tahmin ederler. Bir çıktı birimin hatası büyükse veya gizli birim ve çıktı birimi arasındaki ağırlık büyükse bu takdirde gizli birim için hata sinyali de büyük olacaktır. Gizli birimin hata sinyalleri tahminlendikten sonra tüm bağlantı ağırlıklarını, onların hata sinyali orantısında güncelleştirilir. Bu kısımda öncelikle çok katmanlı ağların yetenekleri üzerinde bilgi verilmeye çalışılacak ve daha sonra geri yayılım algoritması sunulacaktır.

## 2.3.2. Çok Katmanlı Ağıın Yetenekleri

### 2.3.2.1. Desen Sınıflandırma

Geri yayılım ağıının yeteneklerini göstermek üzere öncelikle doğrusal olmayan bir sınıflandırmanın çok katmanlı ve hücreli algılayıcı ile nasıl yerine getirilebileceği gösterilecektir. Bu sınıflandırmanın yapılabilmesi için gereken varsayımlar irdelenecek daha sonra bu varsayımlar olmaksızın yine doğrusal olmayan bir sınıflandırmanın geri yayılım ağı tarafından oluşturulması incelenecektir. Böylelikle doğrusal olmayan desen uzayında geri yayılım ağıının yaptığı görevin kaba bir resmi elde edilmiş olunacaktır.

Öncelikle Şekil 2-12 'deki gibi desen uzayında iki düzlemle ayrılabilir A, B, C, D gibi dört sınıfın olduğunu varsayalım. Her ne kadar desenler birçok girdiye sahip olabilse de buradaki diyagram çok boyutlu uzayın şematik bir gösterimidir. Eksenlere isim verilmemesinin nedeni belirli bir vektör bileşenine karşılık gelmemesindedir. Sadece desen uzayında vektörel bazda düşündüğümüzü varsayıyoruz. Bu sınıfların ayrıştırabilmesi için tek katmanlı bir ağıın kullanılmaya çalışılması boşuna olacaktır. Örneğin A sınıfı diğer sınıflardan doğrusal olarak ayrılamaz. Problemin çözümü için doğrusal olarak ayrıştırma mümkün olmamasına rağmen desen uzayı doğrusal olarak ayrılabilir alt bölgelere bölünerek ve bu bölgelerin belirli kombinasyonlarla üst üste bindirilmesinin araştırılmasıyla çözüme ulaşmak mümkündür.



Şekil 2- 12: A, B, C, D Sınıflarının Ayrışımı İçin Desen Uzayı

Desen uzayının bölünmesi sürecinin ilk aşaması, ilk katmanın TLU birimlerinden oluşturularak ve sonraki katmanda bu birimlerden elde edilen çıktıların kombinasyonları değerlendirilerek başarılabilir. Bu stratejiyi örnek üzerinde ayrıntılı bir şekilde açıklayalım. İlk olarak Şekil 2-12 de görüldüğü gibi tek bir sınıfı (örneğin A sınıfı) diğer sınıflardan doğrusal olarak ayırmak mümkün olmasa da A ve B den oluşan daha yüksek dereceli bir sınıfı C ve D nin oluşturduğu bir sınıftan doğrusal olarak ayırmak mümkündür. Bu sınıflar hakkında söz etmeyi kolaylaştırmak amacıyla A ve B nin tüm desenlerinden oluşan sınıfa AB denilirse, aynı şey C ve D yi içeren CD içinde söz konusudur. Desen uzayı incelendiğinde AB ve CD ‘nin doğrusal olarak ayrılabilir olduğu gibi AD ve BC de doğrusal olarak ayrılabilir. Dolayısıyla Tablo 2-1 de gösterilen bu ikililikleri yerine getirecek  $y_1$  ve  $y_2$  çıktılarına sahip  $U_1$  ve  $U_2$  birimlerini eğitebiliriz.

**Tablo 2- 1:  $y_1, y_2$  Çıktıları**

U <sub>1</sub> birimi		U <sub>2</sub> birimi	
Sınıf	$y_1$	Sınıf	$y_2$
AB	1	AD	1
CD	0	BC	0

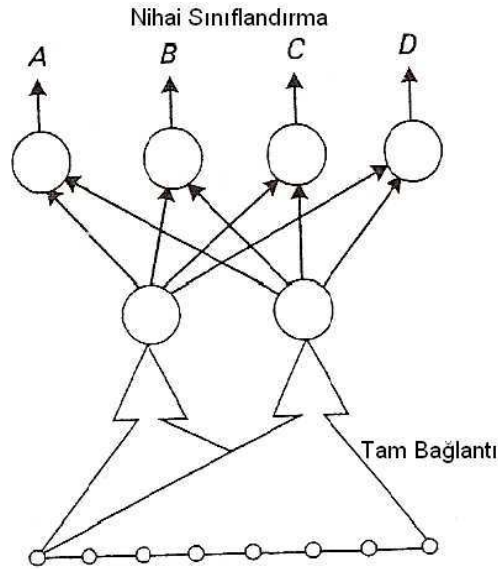
Şimdi varsayalım ki A sınıfına ait bir üye  $U_1$  ve  $U_2$  ‘nin her birine girdi olsun. Tablodan anlaşılacağı üzere bu durumda  $y_1 = y_2 = 1$  çıktıları ortaya çıkacaktır. Tersî durumda bilinmeyen bir  $x$  vektörü girdi ve çıktıların her ikisi de 1 olsun. Böylece  $U_1$  ele alındığında çıktı AB ve  $U_2$  biriminde ise çıktı değeri AD olacaktır. Bunun sağlanabileceği tek koşul girdinin A olması durumudur. Buradan çıkarılacak sonuç  $x$  girdi vektörü ancak ve ancak A ise  $y_1 = 1$  ve  $y_2 = 1$  olduğudur. Diğer üç olasılık da bu şekilde değerlendirilirse her bir sınıfın  $y_1$  ve  $y_2$  terimlerine göre tek bir özel kodu Tablo 2-2 ‘de gösterildiği gibi elde edilir.



**Tablo 2- 2:  $y_1, y_2$  Kodları**

$y_1$	$y_2$	Sınıf
0	0	C
0	1	D
1	0	B
1	1	A

Bu kodlar Şekil 2-13 'da gösterildiği gibi  $U_1$  ve  $U_2$  birimlerinin her ikisine de bağlı, dört tane iki girdili TLU 'ların bir kümesiyle çözülebilir. A sınıfına ait bir elemanı belirtmek amacıyla (1,1) girdisi için "1" çıktısına ve diğer tüm girdiler içinde "0" çıktısına sahip iki girdili bir TLU oluşturulur. B sınıfı elemanında ise TLU sadece (1,0) girdisi sunulduğunda "1" çıktısını üretmelidir. Aynı mantık C ve D içinde devam edecektir. Artık bu girdi-çıkıtı ilişkisi doğrusal olarak ayrılabilir. Dikkat edilirse dört TLU çıktı biriminden sadece birisi herhangi bir zamanda "1" çıktısını gönderecektir böylece sınıflandırma tek bir anlama gelecek şekilde oluşturulmuş olacaktır.



**Şekil 2- 13: Dört Sınıflı Sınıflandırma İçin İki Katmanlı Ağ**

İki önemli noktanın açıklanması gerekmektedir. Birincisi, çıktı birimleri eğitilmemiştir. Desen uzaylarının araştırılmasıyla her biri için uygun ağırlık değerlerinin atanması yapılmıştır. İkincisi, eğer AC veya DB gruplandırmaları kullanılmaya çalışılırsa o zaman hata yapılmış olunacaktı. Çünkü bunların her biri ikili değerlerle doğrusal olarak ayrılabilme özelliğine sahip değildir. Böylece iki birimin eğitilmesi amacıyla iki küçük bilginin gerekli olduğu anlaşılmaktadır.

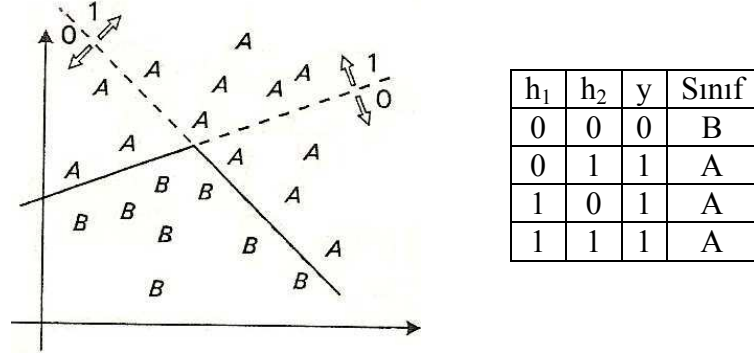
a) Dört sınıfın iki hiper düzlem tarafından ayrılabilir olması

b) AB, CD ‘den doğrusal olarak ayrılabilir ve AD, BC ‘den doğrusal olarak ayrılabilir olması

Bir problemi başarıyla çözebilmek için o problem hakkında ne kadar az bilgi toplanılmasına ihtiyaç duyuluyorsa o kadar yararlı bir çözüm yöntemi geliştirilmiş demektir. Bu amaca yönelik olarak desen uzayı hakkında ön bilgiye ihtiyaç duymayan farklı bir yaklaşım temelli yeni bir eğitim algoritmasının (geri yayılım) kullanılması gerekmektedir.

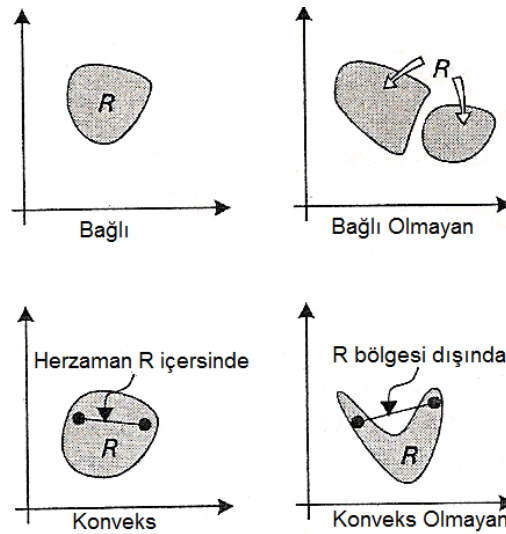
Artık benzer bir olay, geri yayılım algoritmasının doğrusal olmayan ayrılabilir bir desen uzayını sınıflandırmadaki gücünün gösterilmesi için kullanılabilir. Şekil 2-14 ‘ün sağında gösterilen desen uzayındaki durumu ele alalım. A ve B sınıfları tek bir hiper düzlem ile ayrılamazlar. Genellikle keyfi olarak şekil almış bir karar yüzeyine ihtiyaç vardır. Buradaki duruma iki düzlem parçasıyla yaklaşılabilir. Bu parçaların uzatılmasıyla desen etiketlerinin farklı olması haricinde durumun Şekil 2-12 ‘deki benzer olduğu görülebilir. Ağın çözümü yine  $h_1$ ,  $h_2$  (her düzlem için bir tane) gizli hücrelerinden oluşacak fakat A sınıfı için “1”, B sınıfı için “0” sinyali gönderecek tek bir çıktı hücresinden meydana gelecektir. Aslında çıktı birimi hiçbir zaman bu değerleri üretemeyecektir ama Sigmoid fonksiyonunun yaklaşık uç değerleri üretebilmesini sağlayan yeterli büyüklükteki ağırlıklara sahip bulunduğu varsayılmaktadır. Ayrıca gizli hücrelerinde doyum noktalarına yakın değerler ürettiği varsayılmıştır. Gerekli y çıktı birimi fonksiyonu şeklin sağ tarafında gösterilmiştir. Desen uzayının iki girdisi olmasından ötürü iki

boyutlu uzay ile temsil edilebilir. (0,0) girdisi dışında diğer tüm girdiler “1” değerini üretecektir ve böylece doğrusal olarak ayrılabilirlik imkanı ortaya çıkacaktır.



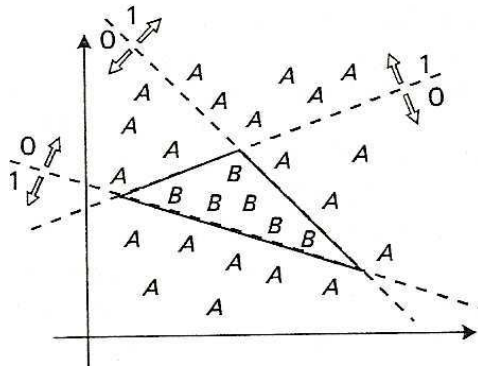
**Şekil 2- 14: Doğrusal Olmayan Ayrılabilir İki Sınıf**

Daha fazla sayıda gizli hücrenin kullanılmasıyla desen uzayında daha kompleks bölgeler oluşturmak mümkündür. Bunu daha ayrıntılı incelemek için birkaç kavramın tanıtılması yararlı olacaktır. Desen uzayında bir R bölgesi için, eğer herhangi bir  $P_1, P_2$  noktalarını tümüyle R bölgesi içerisinde birbirine bağlayan bir yol mevcutsa bu R bölgesine bağlı (connected) denmektedir. Daha resmi olmayan bir ifadeyle Şekil 2-15 ‘in üst kısmında gösterildiği gibi R tek bir ayrık olmayan bölgeden oluşmaktadır. Bağlı bir desen uzayı içerisinde tüm  $P_1, P_2$  noktaları tamamıyla R içerisinde bir doğruyla Şekil 2-15 ‘in alt kısmında gösterildiği gibi bağlanıyorsa bu R bölgesinin konveks olduğu söylenir.



**Şekil 2- 15: Bağlılık ve Konvekslik**

Bu noktadan sonra desen uzayının R bölgesinde veya onun bütünleyicisi  $\bar{R}$  bölgesinde bulunup bulunmadığının haberini veren tek bir çıktı birimine sahip bir ağı ele alalım. Şekil 2-14 'deki durum B sınıfını içeren alanın R olduğu bağlı ve konveks olma durumunu göstermektedir. Örnekte R 'nin desen uzayında sonsuza doğru uzamakta olması şematik olarak bakıldığında diyagramın alt kısmının sınırsız olmasından anlaşılmaktadır. Bununla birlikte üçüncü bir gizli birimin kullanılmasıyla problemi Şekil 2-16 'da gösterilen şekilde ele almak mümkündür. R bölgesi hala konveks ve bağlıdır fakat şimdi sınırlı hale gelmiştir. Daha fazla gizli birimi bu şekilde ekleyerek bölgenin çevresine kenarlıklar eklemeye devam edilebilir. Tek bir gizli katmanı kullanarak bu şekilde ne kadar ilerlenebileceği konusunda çeşitli farklı görüşler olmasına rağmen genellikle herhangi bir R bölgesi tek bir gizli katmanla oluşturulabilmektedir (Gurney, 1997:75).



**Şekil 2- 16: Desen Uzayında Sonlu Bölge**

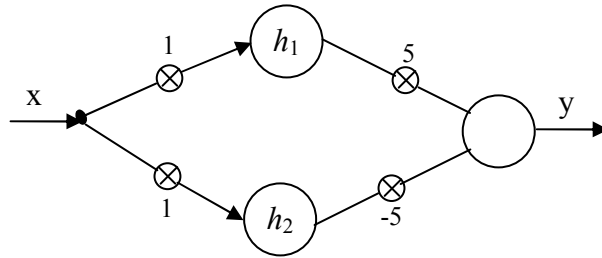
Sonuç olarak gizli hücreleri kullanarak desen uzayında kompleks bölgeler yaratma imkanı vardır. Ayrıca belirtmelidir ki eşik değer fonksiyonu yerine Sigmoid fonksiyonunun çıktı biriminde kullanılması, karar yüzeylerinin sert çizgilerle (uzun doğrular) oluşturulması yerine daha esnek bir başka deyimle daha eğimi fazla karar yüzeylerinin ortaya çıkmasına neden olacaktır.

### 2.3.2.2. Fonksiyon Yaklaşımı

Şimdiye kadar ağı işleyiş şekli desen uzayının geometrik kavramlarıyla açıklanmaya çalışıldı. Alternatif bir açıklama ağı, girdilerinin matematiksel bir fonksiyonunu yerine getirdiği düşüncesidir. Özellikle girdi ve çıktı sinyallerinin

sürekli değerler alması (0, 1 Boolean değerleriyle sınırlandırılmaması) bu duruma uygundur. Örneğin bir hisse senedi fiyatının  $k$  önceki değerleri  $p_{n-1}, p_{n-2}, \dots, p_{n-k}$  temel alınarak  $n$  zamanındaki  $p_n$  değeri tahmin edilmek istenebilir. Böylelikle bu değerler arasındaki eğer varsa fonksiyonel ilişkiyi keşfedecek bir ağı eğitilmesi istenmektedir. Başka bir deyişle sürecin altında yatan  $p_n = p_n(p_{n-1}, p_{n-2}, \dots, p_{n-k})$  fonksiyonu bulunmaya çalışılmaktadır.

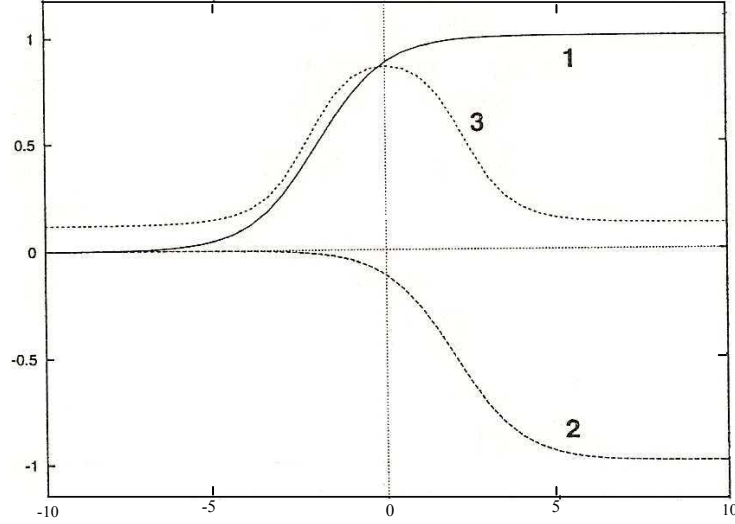
Bu şekilde düşünme tarzının basit bir örneği ağınnın şekli Şekil 2-17 'de gösterilerek verilmiştir. Örnekte tek bir  $x$  girdisi ve sırasıyla  $w_1=1, \theta_1=2, w_2=1, \theta_2=-2$  ağırlık ve eşik değerlerine sahip iki  $h_1, h_2$  gizli hücreleri vardır. Bunlar 5, -5 ağırlıklarına ve 2 eşik değerine sahip tek çıktı hücrelerine girdi olmaktadır. Ağıdaki fonksiyonların işleyiş tarzı Şekil 2-17 'de gösterilmiştir. Yatay eksen ağ hücrelerinin girdileri ve dikey eksen ise çıktılarıdır. Birinci eğri  $h_1$  'in çıktısı  $y_1$  'i, ikinci eğri  $h_2$  'in negatif çıktısı  $y_2$  'i göstermektedir. Çıktı hücrelerinin  $a$  aktivasyon değeri, bu iki eğriyle gösterilen iki miktarın toplamının bir ölçeklendirilmiş kopyası olan  $5(y_1 - y_2)$  'dir. Çıktı hücrelerinin Sigmoid fonksiyonu aktivasyon değerini (0,1) aralığına sıkıştırmak için çalışacaktır fakat  $y$  çıktı değeri (üçüncü eğri)  $a$  ile aynı temel şekle sahip olacaktır.



**Şekil 2- 17: Tek Boyutlu Uzayda Fonksiyon Yaklaşımını Örneklendirmek İçin Basit Bir Ağ**

İlk olarak şuna dikkat etmek gerekir  $x$  'in çok büyük negatif ve pozitif değerleri için  $a$  aktivasyonu neredeyse sabitleşecek ve sifıra eşit olacaktır. Bu aynı zamanda  $y$  'nin de yaklaşık olarak sabit olması anlamına gelecektir.  $x$  büyük negatif değerlerden arttırıldığında ilk gizli hücrenin  $y_1$  çıktısı artışı geçecek  $y_2$  sifıra yakın

değerlerde kalacaktır. Bunun sonucunda çıktı hücresinin  $a$  değeri (dolayısıyla  $y$ ) de artışa geçecektir.  $x$  hala arttırılmaya devam edilirse  $h_2$  için içersine dahil olacak ve  $a$  üzerine negatif katkısını yapacaktır. Böylece  $a$  'da bir düşüş başlayacak ve en sonunda  $y_1$  ve  $y_2$  yaklaşık eşit olacak, birbirlerinin  $a$  üzerine katkılarını götürüp  $a$  'nın neredeyse sıfır olmasını sağlayacaklardır.



**Şekil 2- 18: Şekil 2-17 'deki Ağ Örneğinin Fonksiyonelliği**

Eşik değerlerinin birbirine göre aralarındaki farklılığın arttırılması çıktının daha geniş bir tepeye sahip olmasına neden olurken çıktı hücresinin ağırlık değerlerinin arttırılması çıktının şeklini daha dikleştirecektir. Ayrıca çıktı hücresinin ağırlıklarının büyüklük olarak farklılaştırılması simetrikliği de bozacaktır. Ağın çıktısı  $y$  'i uygun ağ parametrelerinin seçimiyle, tek tepeli (single-humped) veya tek modlu (unimodal) fonksiyonların geniş sınıfından herhangi bir tanesini yerine getiren  $x$  girdisinin bir fonksiyonu olarak düşünebiliriz. Diğer bir  $h_3$  gizli hücresinin için içine katılması ikinci bir tepenin görünür olmasını sağlayacak ve aynı mantıkla  $h_4$ ,  $h_5$  öteki gizli hücrelerin oluşturulmasına devam edilebilecektir. Bu şekilde ağların eğitilerek  $y = y(x)$  karmaşık fonksiyonel ilişkiye sahip problemleri çözebilmeleri mümkün olmaktadır.

Çoğu durumda, ağ çok sayıda  $x_1, x_2, \dots, x_n$  girdiye ve  $y_1, y_2, \dots, y_m$  çıktıya sahip olacaktır. Bu durumda  $y_j = y_j(x_1, x_2, \dots, x_n)$  şekline sahip  $m$  fonksiyonun yerine getirildiği düşünülebilir.  $n = 2$  iken bu ilişki görsel olarak yüzey grafiği ile gösterilir fakat  $n > 3$  iken gerçek görüntüleme imkansız olmaktadır. Yapılabilecek en iyi şey girdilerin ikisi ( $p$  ve  $q$ ) dışında tümünün sabit tutulması ve  $x_p, x_q$  'nun bir fonksiyonu olarak  $y_j$  'deki değişimin incelenmesidir.

Belirli bir gizli katman sayısı için karar yüzeyinin nasıl kompleks olacağına bilinmesinin geometrik bağlamda önemli olması gibi, fonksiyonel anlamda ise her bir durumda hangi çeşit fonksiyonun kullanılacağı sorusu önem kazanmaktadır. Hornik ve arkadaşları 1989 'da tek bir gizli katmanı kullanarak herhangi bir sürekli fonksiyona istenilen düzeyde yaklaşmanın mümkün olabildiğini göstermişlerdir. Bu tek bir gizli katmanın herhangi bir karar yüzeyini kodlamak için yeterli olduğunu söyleyen desen uzayı analizi ile uyumludur (Gurney, 1997:78).

### 2.3.2.3. Özellik Belirleyici Olarak Gizli Hücreler

Tablo 2-3 'de gösterilen dördü 11 girdi ve tek çıktılı eğitim desenlerini ele alalım. Bu eğitim desenindeki özelliklerin neler olduğu konunun ilgi odağını oluşturmaktadır. Özelliğinden kasıt desen uzayındaki desenleri birbirinden ayırmada desene ait tüm bileşenlerinin incelenmesine gerek kalmadan istenilen bilginin sağlanmasıdır. İlk bakışta  $x_3$  ve  $x_6$  'dan  $x_9$  'a kadarki vektör bileşenlerinin her zaman "1" değerini alma özelliğine sahip olduğu görülmektedir. Bu bileşenlere değerleri değişmediğinden ve her bir vektör sınıfı hakkında bilgi vermediğinden "arka plan bileşenleri" (background components) denilebilir. Özellik tanımını daha iyi yansıtabilecek bileşenler, her değer (0 veya 1)  $y$  çıktısıyla bir şekilde ilişkili olduğu  $x_1$  veya  $x_{11}$  gibi bileşenlerdir. Örneğin  $x_1$  ele alınırsa her zaman  $y = x_1$  olmasından ötürü  $x_1$  girdisi çıktıyı tamamıyla belirleyen oldukça bilgi verici bir özelliktir. Daha karmaşık problemlerde belirli bir desen sınıflandırması hakkında emin olmak için birkaç özellikteki bilgileri birleştirme gerekliliği duyulabilir.

**Tablo 2- 3:** 11 Girdili ve Tek Çıktılı Eğitim Deseni

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$y$
1	1	1	1	1	1	1	1	1	0	0	1
1	1	1	0	0	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	0
0	0	1	1	0	1	1	1	1	1	1	0

Bu örnekteki vektörleri eğitmek için Sigmoid fonksiyonuna sahip bir hücre kullanılırsa 1. girdi bileşenine ait pozitif büyük bir ağırlık değerinin olması beklenir böylece hücre aktivasyonunun pozitif değerler alması sağlanacaktır. Öte yandan  $x_{11}$  çıktıyla zıt değerler olarak bir ilişki kurmaktadır. Bu ilişki 11. girdi bileşeni  $x_{11} = 1$  olduğunda, çıktıyı sıfır olması yönünde zorlayacak negatif bir ağırlık değerinin geliştirilmesiyle ele geçirilir.

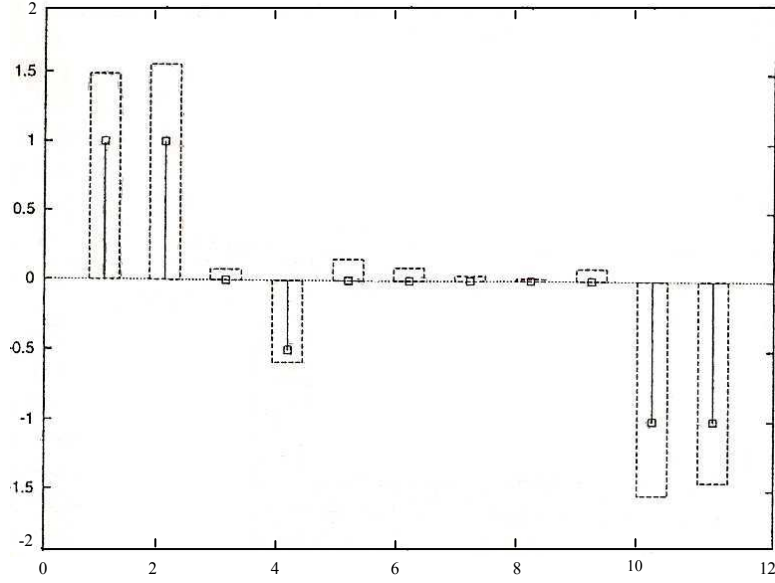
Sıra bu düşüncenin matematiksel olarak formülize edilmesine gelmiştir. Tablo 2-3 'deki her "0" değerini yeni bir  $\bar{x}_i$  bileşeni ve  $\bar{y}$  çıktısı elde etmek için "-1" değeriyle değiştirildiğini varsayalım. Böylece her  $p$  deseni ve her  $\bar{x}_i^p$  bileşeni için şu form  $\bar{x}_i^p \bar{y}^p$  ortaya çıkacaktır. Bu girdi-çıkıtı korelasyonunun bir ölçüsünü vermektedir. Girdi ve çıkıtı aynı ise ifade +1 ve farklı ise -1 değerini alacaktır. Dikkat edilirse orijinal Boolean notasyonu (0 ve 1) kullanılırsa, bu çarpım değeri  $x_i$  veya  $y$ 'den herhangi biri sıfır olmasından ötürü bu ikisinin eşit olup olmadığına bakmaksızın sıfır değerine eşit olacaktır. Şimdi  $i$  inci bileşendeki  $c_i$  ortalama korelasyonu tanımlayalım.

$$c_i = \frac{1}{4} \sum_p \bar{x}_i^p \bar{y}^p \quad (2.37)$$

Örneğin  $x_1$  gibi bir özellik için eşitlik (2.37) +1 değerini verecektir. Arka plan bileşenlerinde girdi-çıkıtı benzerlikleri (+1 ler) sayısı kadar farklılıkları (-1 ler)



olduğundan  $c_i$  değeri 0 olacaktır.  $x_{11}$  gibi bir bileşen çıktıyla anti korelasyonludur ve  $c_{11} = -1$  'dir. Bu en başta tanımlanan özellik tanımıyla oldukça uyum sağlamaktadır.



**Şekil 2-19: Ağırlıklar ve Özellikler**

Şekil 2-19 'da  $c_i$  korelasyon katsayıları  $i$  'lere göre işaretlenmiş ve ucunda küçük kare sembolü olan doğrularla gösterilmiştir. Aynı zamanda delta kuralı kullanılarak tek bir hücrenin desen setine dayanarak eğitilmesi sonucunda elde edilen ağırlık değerleri, bu doğruları içine alan dikdörtgenlerle çizilmiştir. Görüldüğü gibi ağırlıklar ve korelasyon katsayıları (özellikler) arasında çok yakın bir eşleşme vardır böylece hücrenin eğitim setindeki özellikleri tespit etmeyi öğrendiği söylenebilir.

Özellikle  $x_4$  bileşenini ele aldığımızda üç desen için çıktının zıt değerini almakta ve biri için aynı değeri almaktadır. Bu anti korelasyonlu bir özelliktir fakat  $x_{11}$  bileşenindeki kadar bir güce sahip değildir.

### 2.3.3. Geri Yayılım Algoritması

Önceden tanıtılan çok katmalı ağ için kullanılan kısaltılmış notasyonla geri yayılım algoritmasının geliştirilmesi kolaylaşacaktır. Kısaltılmış notasyonla üç katmanlı ağ Şekil 2-10 'da verilmişti.

Çok katmanlı ağlarda bir katmanın çıktısı onu takip eden katmanın girdisi olmaktadır. Bu süreci tanımlayan eşitlik aşağıdaki gibi ifade edilebilir:

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1} (\mathbf{W}^{m+1} \mathbf{a}^m + \mathbf{b}^{m+1}) \quad \text{için } m = 0, 1, \dots, M-1 \quad (2.38)$$

Burada  $M$ , ağdaki katmanların sayısıdır. İlk katmandaki nöronlar dış girdileri alır.

$$\mathbf{a}^0 = \mathbf{p} \quad (2.39)$$

Eşitlik (2.38), (2.39) için başlangıç noktasını sağlamaktadır. Son katmandaki nöronların çıktısı ağın çıktısı olarak görülmektedir.

$$\mathbf{a} = \mathbf{a}^M \quad (2.40)$$

### 2.3.3.1. Performans İndeksi

Çok katmanlı ağlar için geri yayılım algoritması, LMS algoritmasının geliştirilmiş şeklidir. Her iki algoritmada hata kareler ortalaması performans indeksini kullanmaktadır. Algoritmaya doğru ağ davranışının aşağıdaki gibi bir örnek setinin verilmesiyle başlanır.

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \quad (2.41)$$

Burada  $\mathbf{p}_Q$ , ağa sunulan girdi ve  $\mathbf{t}_Q$  ise ona karşılık gelen hedef çıktı değeridir. Her girdi ağa uygulandıkça ağın çıktı değeri hedef değer ile karşılaştırılmaktadır. Algoritma, hata karesinin ortalamasını minimize etmek amacıyla ağın parametrelerini ayarlamalıdır.

$$F(\mathbf{x}) = E [ e^2 ] = E [ (t - a)^2 ] \quad (2.42)$$

Burada  $\mathbf{x}$  ağın ağırlık ve sapmalarının bir vektörüdür. Bunu ağın çoklu çıktıya sahip olması durumuna genellebiliriz.

$$F(\mathbf{x}) = E [ \mathbf{e}^T \mathbf{e} ] = E [ (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a}) ] \quad (2.43)$$

LMS algoritmasında olduğu gibi (2.44) eşitliğiyle hata karesinin ortalamasına yaklaşacağız.

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k) \quad (2.44)$$

Burada hata karesinin beklenen değeri,  $k$  inci iterasyondaki hata karesi ile yer değiştirmiştir.

Yaklaşık hata kare ortalaması için eğim iniş algoritmasını şöyledir.

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \quad (2.45)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m} \quad (2.46)$$

Burada  $\alpha$  öğrenme oranıdır.

Şimdiye kadar LMS algoritması ile benzer bir gelişim gösteren algoritmanın zor parçası olan kısmi türevlerin hesaplanması kısmına gelinmiştir.

### 2.3.3.2. Zincir Kuralı

Tek katmanlı doğrusal ağda, eşitlik (2.35) ve (2.36) de kullanılarak kısmi türevlerin hesaplanması gösterildi. Çok katmanlı ağ durumunda, hata gizli katmandaki ağırlıkların açık bir fonksiyonu değildir bu yüzden buradaki kısmi türevler kolay bir şekilde hesaplanamamaktadır.

Hatanın gizli katmandaki ağırlıkların dolaylı bir fonksiyonu olmasından ötürü türevleri hesaplamak için matematikteki zincir kuralından yararlanılır. Zincir kuralını hatırlatmak gerekirse farz edelim ki elimizdeki  $f$  fonksiyonu sadece  $n$  değişkenine

bağlı olsun ve biz üçüncü bir değişken olan  $w$  ye göre  $f$  fonksiyonunun türevini almak isteyelim. Bu durumda zincir kuralı:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw} \quad (2.47)$$

Bu kavram eşitlik (2.45) ve (2.46) 'daki türevleri bulmak için kullanılacaktır.

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (2.48)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \quad (2.49)$$

Bu eşitliklerdeki ikinci terim,  $m$  katmanına net girdi o katmandaki ağırlık ve sapmaların doğrudan bir fonksiyonu olduğu için kolayca hesaplanabilir.

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \quad (2.50)$$

Böylece

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \quad \frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (2.51)$$

Şimdi eşitlik (2.52) tanımını da yapılırsa

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m} \quad (2.52)$$

Eşitlik (2.52) 'nin anlamı,  $m$  inci katmandaki net girdinin  $i$  inci elamanındaki değişimlere  $\hat{F}$  'in duyarlılığıdır. Öyleyse eşitlik (2.48) ve (2.49) 'u aşağıdaki şekilde basitleştirebiliriz.

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \quad (2.53)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \quad (2.54)$$

Artık yaklaşık eğim iniş algoritması aşağıdaki gibi ifade edilebilir.

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad (2.55)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (2.56)$$

Matris formatında yazılırsa,

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (2.57)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (2.58)$$

Burada

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix} \quad (2.60)$$

### 2.3.3.3. Duyarlılıkların Geri Yayılımı

Bu noktadan sonra geriye zincir kuralının diğer bir uygulamasını gerektiren  $\mathbf{s}^m$  duyarlılıklarının hesaplanması kalmıştır. Geri yayılım ismini ortaya çıkaran bu süreçtir. Çünkü  $m$  inci katmandaki duyarlılığı  $m+1$  inci katmandaki duyarlılıktan hesaplayan yinelemeli bir ilişkiyi tanımlamaktadır.

Duyarlılıklar için yinelemeli ilişkiyi türetmede aşağıdaki Jacobian matrisinden yararlanılacaktır.

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix} \quad (2.61)$$

Bu matrisin bir ifadesinin bulunulması amacıyla matrisin  $i, j$  inci elemanları ele alınırsa:

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial \left( \sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} \dot{f}^m(n_j^m) \end{aligned} \quad (2.62)$$

Burada

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \quad (2.63)$$

Böylece Jacobian matrisi şu şekilde ifade edilebilir

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m) \quad (2.64)$$

Burada

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \quad (2.65)$$

Şimdi matris formunda zincir kuralı kullanılarak duyarlılıklar için yinelemeli ilişki yazılabilir.

$$\begin{aligned} \mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left( \frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} \end{aligned} \quad (2.66)$$

Geri yayılım algoritmasının adının nereden türediği artık görülmektedir. Duyarlılıklar ağıın son katmanından geriye doğru ilk katmanına yayılmaktadır.

$$\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \dots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1 \quad (2.67)$$

Geri yayılım algoritması, LMS algoritmasında kullanılan aynı yaklaşık eğim iniş tekniğini kullanmaktadır. Tek zorluk gradientin hesaplanması için öncelikle duyarlılıkların geriye doğru yayılması zorunluluğudur. Bu noktada da zincir kuralı etkili bir şekilde kullanılmaktadır.

Geri yayılım algoritmasının tamamlanması için tek bir adım geriye kalmıştır. Eşitlik (2.66) daki yinelemeli ilişki için ihtiyaç duyulan şey,  $\mathbf{s}^M$ , başlangıç noktasıdır. Bu son katmandan elde edilir.

$$s_i^M = \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})}{\partial n_i^M} = \frac{\partial \sum_{j=1}^{S^M} (t_j - a_j)^2}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M} \quad (2.68)$$

Burada

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = \dot{f}^M(n_i^M) \quad (2.69)$$

Şu şekilde ifade edilebilir:

$$s_i^M = -2(t_i - a_i)\dot{f}^M(n_i^M) \quad (2.70)$$

Matris notasyonu ile yazarsak,

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \quad (2.71)$$

#### 2.3.3.4. Özet

Geri yayılım algoritmasını özetlemek gerekirse, ilk adım girdilerin ağ boyunca ileri yayılmasıdır.

$$\mathbf{a}^0 = \mathbf{p} \quad (2.72)$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \quad \text{için } m = 0, 1, \dots, M-1 \quad (2.73)$$

$$\mathbf{a} = \mathbf{a}^M \quad (2.74)$$

Sonraki adım, duyarlılıkların ağ boyunca geriye doğru yayılmasıdır.

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \quad (2.75)$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} \quad \text{için } m = M-1, \dots, 2, 1. \quad (2.76)$$

Son olarak, ağırlık ve sapmalar yaklaşık eğim iniş kuralı kullanılarak güncellenir.



$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (2.78)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (2.79)$$

Böylece geri yayılım algoritmasında kullanılan ağırlık değişimlerinin formülasyonu tanımlanmış oldu. Burada tanımlanan eğim iniş kuralına dayalı geri yayılım algoritmasına temel geri yayılım algoritması denmektedir. Temel geri yayılım algoritması uygulamalarda hatayı belirli bir seviyenin altına çeken ağırlılık değerlerinin elde ederken oldukça yavaş kalmaktadır. Dolayısıyla 3. Bölümde tanıtılacak olan temel geri yayılım algoritmasını hızlandıran algoritmaların arayışlarına başlanmıştır.

### 3. BÖLÜM

## UYGULAMADA GERİYAYILIM ALGORİTMASI VARYASYONLARI VE SİNİR AĞLARINDA DUYARLILIK ANALİZİ UYGULAMASI

### 3.1. Giriş

Önceki bölümde teorik alt yapısıyla tanıtılan temel geriyayılım algoritması yapay sinir ağları araştırmasında önemli bir devrim niteliği taşımaktadır. Bunun yanı sıra çoğu pratik uygulamalarda temel algoritma oldukça yavaş kalmaktadır. Temel algoritma pratik bir probleme uygulandığında eğitim aşamasının tamamlanması için uzun bir süre bilgisayar çalışmasına ihtiyaç duyulur. Bu bölümde uygulamalarda sıkça kullanılan, algoritmayı daha pratik hale getiren ve önemli derecede hızlandıran temel algoritmanın varyasyonları tanıtılıp ve bir uygulama üzerinde çalışabilirliği örneklendirilecektir.

Bir önceki bölümde sunulan ve bu bölüm içerisinde sunulacak tüm algoritmalar türevlerin ağın son katmanından ilk katmanına doğru işlendiği geriyayılım prosedürünü kullanmaktadırlar. Bu nedenden ötürü tümüne birden geriyayılım algoritmaları denebilmektedir. Algoritmalar arasındaki farklılık ağırlıkları güncellemek için kullanılan nihai türevlerden kaynaklanmaktadır. Temel geriyayılım algoritmasının bir yaklaşık eğim iniş algoritmasıdır (approximate steepest descent). Bazı durumlarda geriyayılım olarak bahsedilen algoritmanın aslında eğim iniş algoritması olduğu görülmektedir. Bu karışıklığın önüne geçmek amacıyla temel geriyayılım algoritmasına eğim iniş geriyayılımı da denilebilecektir.

### 3.2. Temel Geriyayılım Algoritmasının Geliştirilmesi

Temel geriyayılım algoritmasına nazaran çok daha hızlı algoritmalar oluşturma çalışmaları iki kategoriye ayrılmaktadır. İlk kategori standart geriyayılım algoritması performansının analizi üzerine dayalı sezgisel (heuristic) tekniklerin geliştirilmesini içermektedir. Bu sezgisel teknikler değiştirilebilir öğrenme oranı

(variable learning rate), hızlandırıcı (momentum) kullanılması ve dirençli geriyayılım (resilient backpropagation) gibi düşünce alt yapılarına dayandırılmıştır.

Hızlı algoritmaların ikinci kategorisi ise standart nümerik optimizasyon tekniklerini kullanarak sinir ağları eğitiminin gerçekleştirilmesini amaçlamaktadır. Karesel hatayı minimize etmek amacıyla ileri beslemeli sinir ağının eğitilmesi basitçe bir nümerik optimizasyon problemidir. Nümerik optimizasyonun geçmişe dayanan önemli bir araştırma alanı olması hızlı eğitim algoritmalarının aranmasında var olan nümerik optimizasyon tekniklerinden yararlanılması düşüncesini doğurmaktadır. Bu bölümde çok katmanlı ağların eğitilmesine başarılı bir şekilde uygulanabilen Eşlenik (Conjugate) Gradient, Quasi-Newton ve Levenberg-Marquardt nümerik optimizasyon teknikleri verilecektir.

Öğrenme oranı çok büyük olmadığı sürece LMS algoritması, ortalama karesel hatayı minimize eden bir çözüme ulaşmayı garanti etmektedir. Tek katmanlı doğrusal ağ için ortalama karesel hatanın kuadratik bir fonksiyon olması bu sonucu doğurmaktadır. Çünkü kuadratik fonksiyonun tek bir denge noktası bulunmakta ve Hessian matrisi sabit bir değer almaktadır. Böylece herhangi bir doğrultuda fonksiyonun eğimi değişmemekte ve fonksiyon konturları eliptik bir şekil olmaktadır.

Temel geriyayılım algoritması LMS algoritmasının genelleştirilmiş şekli olup aynen LMS algoritması gibi ortalama karesel hatayı minimize eden yaklaşık eğitim algoritmasıdır. Tek katmanlı doğrusal bir ağ kullanılması durumunda bu iki algoritma birbirine eşdeğer olmaktadır. Çok katmanlı ağlar söz konusu iken temel geriyayılım algoritmasının karakteristikleri oldukça farklılaşmaktadır. Bu farklılık tek katmanlı doğrusal ağın ve çok katmanlı doğrusal olmayan ağın ortalama karesel hata performans yüzeyinden kaynaklanmaktadır. Tek katmanlı doğrusal ağın performans yüzeyi tek bir minimum noktaya ve sabit eğime sahipken, çok katmanlı ağın performans yüzeyi çok sayıda yerel minimum noktasına ve parametre uzayının farklı bölgelerinde oldukça farklılaşan bir eğime sahiptir. Bu nedenden ötürü eğitim algoritması için uygun bir öğrenme oranını seçmek zor olmaktadır. Performans yüzeyinin bazı bölgeleri oldukça yassı olup öğrenme oranının büyük seçilmesine izin

verirken eğimin yüksek olduğu diğer bölgeler küçük bir öğrenme oranının seçilmesini gerektirmektedir.

Ağ içersinde kullanılan Sigmoid transfer fonksiyonuyla performans yüzeyinde yassı bölgelerin oluşması kaçınılmazdır. Bunun nedeni Sigmoid fonksiyonunun büyük girdi değerlerine önemli derecede yassı olmasıdır. Yüzeyin bazı kısımlarının eğiminin dik ve bazı kısımlarının ise yassı olması bu yüzeyde standart eğim iniş algoritmasının kullanılmasında sorun doğurmaktadır. Örneğin eğim iniş algoritması eğimin sifira yaklaştığı fakat gerçekte yerel minimum olmayan bir noktada algoritmayı durdurabilecektir. Bu durumun üstesinden farklı başlangıç değerleriyle algoritmanın çalıştırılıp her seferinde yaklaşık aynı minimum noktaya ulaşıldığından emin olunmasıyla gelinmektedir. Ayrıca genellikle ağın başlangıç sapma ve ağırlık değerlerinin rassal küçük sayılara atandığını belirtmek gerekir. Başlangıç sapma ve ağırlık değerlerinin rassal seçimi algoritmanın performansını etkileyecektir. Farklı algoritmaların performansları karşılaştırılmak isteniyorsa her algoritma birkaç farklı başlangıç ve sapma setinin kullanılmasıyla test edilmelidir.

Algoritma performans yüzeyinin yassı bölgelerinden geçerken hızlı bir geçişin sağlanması amacıyla öğrenme oranının arttırılması istenir. Bununla birlikte öğrenme oranının çok fazla arttırılması ise denge noktasına ulaşamamasına neden olabilecektir. Özellikle eğimin arttığı bölgelerde minimum noktayı içeren vadiye ulaşmak için öğrenme oranı küçültülecektir (Gurney, 1997:69). Bu durum yassı yüzeylerde öğrenme oranının arttırılabileceğini ve eğim arttıkça ise oranın düşürülebileceğini göstermektedir. Asıl sorun algoritmanın yassı bir yüzeyde olduğunu nasıl bileceği sorusudur. Bu soru algoritmanın geliştirilmesini sağlayan izleyen kısımda verilecek bazı prosedürlerle karşılanmaya çalışılacaktır.

### **3.2.1. Sezgisel Yaklaşımlar**

Temel geriyayılım algoritmasının veya başka bir deyişle eğim iniş geriyayılım algoritmasını, yukarda bir kısmı bahsedilen ve takip eden konular içersinde bahsedilecek nedenlerden ötürü geliştirilmesine yönelik olarak ilk etapta

sezgisel yaklaşımlar ele alınmıştır. Bu kısım içerisinde üç çeşit sezgisel metot verilmiştir.

### 3.2.1.1. Hızlandırıcı

İlk olarak hızlandırıcı teriminin kullanılmasını ele alalım. Hızlandırıcı kullanan temel geriyayılım algoritmasının daha hızlı bir şekilde sonuca ulaşması mümkün olacaktır. Hızlandırıcı ağırlık sadece yerel eğimi baz alarak ağırlıkların değişimine karar vermemesini, işin içine hata (performans) yüzeyindeki son trendleri de katabilmesini sağlamaktadır. Hızlandırıcıyla ağırlık hata yüzeyindeki küçük özellikleri ihmal edebilmektedir. Böylece ağırlık dar bir yerel minimumda sıkışıp kalmasının önüne geçilmiş olmaktadır.

Algoritmaya hızlandırıcı eklendiğinde ağırlık değişimi, temel geriyayılım kuralınca önerilen yeni değişim değeri ve bir önceki iterasyondaki ağırlık değişim değerinin oransal toplamıyla bulunur. Bir önceki iterasyondaki ağırlık değişiminin veya başka bir deyişle son ağırlık değişiminin etkisinin büyüklüğüne 0 ve 1 arasında herhangi bir değer alan  $\gamma$  hızlandırıcı katsayısına göre karar verilir. Hızlandırıcı katsayısı 0 ise bunun anlamı ağırlık değişiminin yalnızca eğimlere dayandırıldığıdır. Hızlandırıcı katsayısı 1 ise eğim ihmal edilmiş ve ağırlık değişimi son ağırlık değişimine eşitlenmiş demektir.

Parametre değerlerinin temel geriyayılım algoritmasına göre yenilenmesinin eşitlik (2.78) ve (2.79) ile yapılmış olduğunun hatırlatılmasından sonra hızlandırıcı katsayısının kullanıldığı eşitlik (3.1) ve (3.2) aşağıda verilmiştir.

$$\Delta \mathbf{W}^m(k) = \gamma \Delta \mathbf{W}^m(k-1) - (1-\gamma) \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (3.1)$$

$$\Delta \mathbf{b}^m(k) = \gamma \Delta \mathbf{b}^m(k-1) - (1-\gamma) \alpha \mathbf{s}^m \quad (3.2)$$

Burada  $\gamma$  hızlandırıcı katsayısı aşağıdaki eşitsizliği sağlamalıdır.

$$0 \leq \gamma \leq 1 \quad (3.3)$$

### 3.2.1.2. Deęiřtirilebilir Öğrenme Oranı

Deęiřtirilebilir öğrenme oranındaki mantık yassı yüzeylerde öğrenme oranının arttırılarak minimum noktaya varmanın hızlandırılması ve eğimin arttığı yüzeylerde ise öğrenme oranının düşürülerek minimum noktaya varmanın garanti altına alınmasıdır. Bilindięi üzere tek katmanlı doğrusal aęın ortalama karesel hata performans yüzeyi her zaman kuadratik fonksiyon olmakta ve maksimum öğrenme oranının belirlenmesinde bir sorun bulunmamaktadır (Hagan, Demuth ve Beale, 1996:9-6). Çok katmanlı aęın performans yüzeyi kuadratik fonksiyon olmadığından parametre uzayının farklı bölgelerinde yüzeyin řekli oldukça farklılaşabilmektedir. Durum böyleyken öğrenme oranı yüzeyde bulunulan yere göre ayarlanarak minimum noktaya varma hızlandırılabilir. Asıl sorun oranın ne kadar ve ne zaman deęiřtirileceęinin belirlenmesidir. Eğitimden önce optimal öğrenme oranının belirlenmesi pratik deęildir. Algoritma performans yüzeyinde hareket ederken eğitim süreci boyunca optimal öğrenme oranı deęişmektedir. Optimal öğrenme oranının belirlenmesine yönelik olarak farklı yaklaşımlar bulunmaktadır. Burada ayarlanacak parametre sayısının en az olduęu prosedür verilmiřtir.

1. Aęırlıkların güncellenmesinden sonra karesel hata önceden belirlenmiř  $\zeta$  yüzde deęerinden (genellikle yüzde beř) daha fazla artıyorsa bu durumda aęırlıkların güncellenmesi iptal edilir, öğrenme oranı  $0 < \rho < 1$  çarpımıyla çarpılır ve eęer  $\gamma$  hızlandırıcı katsayısı kullanılıyorsa sıfıra eřitlenir.
2. Aęırlıkların güncellenmesinden sonra karesel hata azalıyorsa aęırlıkların güncellenmesi kabul edilir ve öğrenme oranı  $\eta > 1$  çarpımıyla çarpılır. Eęer  $\gamma$  önceden sıfıra eřitlenmiřse orijinal deęerine geri döndürülür.
3. Karesel hata  $\zeta$  deęerinden daha az olarak artıyorsa aęırlık güncellemesi kabul edilir fakat öğrenme oranı deęiřtirilemez. Eęer  $\gamma$  önceden sıfıra eřitlenmiřse orijinal deęerine geri döndürülür.

### 3.2.1.3. Dirençli Geriyayılım

Çok katmanlı ağlar gizli katmanlarında genellikle Sigmoid transfer fonksiyonlarını kullanır. Bu fonksiyonlara sonsuz aralıktaki bir girdiyi sonlu aralıktaki bir çıktıya dönüştürmesinden ötürü sıkıştırıcı fonksiyonlarda denmektedir. Sigmoid fonksiyonları girdi değerleri gittikçe büyürken eğimlerinin sıfıra yaklaşması gerçeğiyle karakterize edilmiştir. Bu durum Sigmoid fonksiyonlarını kullanan çok katmanlı bir ağın eğim iniş metoduyla eğitilmesinde soruna neden olmaktadır. Çünkü türevler çok küçük değerler alabilmekte böylece ağırlık ve sapma değerleri optimal değerlerinden uzak olsa bile çok küçük değişimlere maruz kalabilmektedir.

Dirençli geriyayılım eğitim algoritmasının amacı kısmi türev büyüklüklerindeki bu olumsuz etkilerin ortadan kaldırılmasıdır. Türevlerin büyüklüğünün ağırlıkların güncellenmesinde etkisinin olmaması istenmektedir. Bu amaçla yalnızca türevlerin işareti ağırlık güncellemesinin yönünün belirlenmesinde kullanılmaktadır. Ağırlık değişiminin büyüklüğü ayrı bir güncelleme değeri tarafından belirlenmektedir. Performans fonksiyonunun söz konusu ağırlığa göre türevinin ardışık iki iterasyonda aynı işarete sahip olmasıyla ağırlığın güncelleme değeri bir artış çarpanı kadar arttırılmaktadır. Söz konusu ağırlığa göre türev son iki iterasyonda işaret değiştiriyorsa bu durumda güncelleme değeri bir azalış çarpanı kadar azaltılmaktadır. Türev sıfıra eşitse güncelleme değeri aynı bırakılmaktadır. Her ne zaman ağırlık değişimleri farklı yönlerde gerçekleşiyorsa değişim azaltılır. Ağırlık değişimleri birkaç iterasyon boyunca aynı doğrultuda değişmeye devam ediyorsa bu takdirde ağırlık değişimlerinin büyüklüğü arttırılır.

Sezgisel yaklaşımlar kullanılarak temel geriyayılım algoritmasının modifiye edilmesi genellikle daha hızlı bir şekilde sonuca varılmasını sağlamaktadır. Bununla birlikte bu metotların birkaç dezavantajı vardır. Birincisi temel geriyayılım algoritması için ayarlanması gereken tek parametre öğrenme oranı iken modifikasyonlar  $\zeta$ ,  $\rho$  ve  $\gamma$  gibi birkaç parametre değerinin ayarlanmasını gerektirmektedir. Daha karmaşık sezgisel yaklaşımlarda ise ayarlanması gereken beş, altı parametreye sahip olunabilmektedir. Genellikle algoritmanın performansı bu parametrelerdeki değişimlere oldukça duyarlıdır. Parametre değerlerinin seçimi ayrı

bir problemi doğurmaktadır. Bu modifikasyonlara yönelik ikinci dezavantaj temel algoritmayla sonunda bir çözüm bulunabilecek bazı problemlerde bir sonuca ulaşılamayabilinecek olunmasıdır. Genellikle bu dezavantajlar daha kompleks algoritmaların kullanılması durumunda ortaya çıkmaktadır.

### 3.2.2. Nümerik Optimizasyon Teknikleri

Temel geriyealım algoritmasının geliştirilmesine yönelik bazı sezgisel yaklaşımların incelenmesinden sonra bu bölümde standart nümerik optimizasyon tekniklerine dayanan metotlar ele alınacaktır. Bu optimizasyon algoritmalarının temel ilkeleri 17yy' da Kepler, Fermat, Newton ve Leibniz gibi matematikçi ve bilim adamları tarafından kurulmuştur. Temel geriyealım algoritması tekrar belirtmek gerekirse bir nümerik optimizasyon tekniği olan eğim iniş metodunu kullanmaktadır. Bu kısımda sinir ağları eğitimine iyi bir şekilde uyarlanabilen, genel kapsamda diğer iki nümerik optimizasyon tekniği incelenecektir. Bunlar Eşlenik Gradient ve Levenberg-Marquardt algoritmalarıdır.

Ele alınan algoritmaların amacı bir  $F(\mathbf{x})$  performans indeksini optimize etmektir. Optimize etmekten kasıt  $F(\mathbf{x})$ ' i minimize eden  $\mathbf{x}$  değerinin bulunmasıdır. İncelenen tüm optimizasyon algoritmaları iteratiftir. Bir  $\mathbf{x}_0$  başlangıç değerinden başlar ve (3.4) eşitliğine göre aşama aşama tahmin değerinin güncellenmesine dayanır.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (3.4)$$

Veya

$$\Delta \mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k) = \alpha_k \mathbf{p}_k \quad (3.5)$$

Burada  $\mathbf{p}_k$  arama yönünü (search direction) gösterirken pozitif skaler  $\alpha_k$  ise adım büyüklüğünü (length of step) belirleyen öğrenme oranıdır.



Algoritmalar  $\mathbf{p}_k$  arama yönünün seçimiyle birbirlerinden farklılaşmaktadır. Ayrıca  $\alpha_k$  öğrenme oranının seçimine ilişkin farklı yollarda bulunmaktadır. Bunlar, sadece kuadratik fonksiyonlar için maksimum öğrenme oranını belirleyen denge öğrenme oranı (stable learning rate) ve performans indeksini her iterasyonda  $\alpha_k$  'ya göre minimize yapan doğru boyunca minimizasyondur (minimizing along a line). Aşağıda minimize edilecek  $F(\mathbf{x})$  fonksiyonunun kuadratik olduğu varsayıp,  $\alpha_k$  'nın seçiminde kullanılan denge öğrenme oranı ve doğru boyunca minimizasyonun sırasıyla formülasyonları verilmiştir.

$$\alpha < \frac{2}{\lambda_{mak}} \quad (3.6)$$

Burada  $\lambda_{mak}$  ,  $F(\mathbf{x})$  performans indeksinin Hessian matrisi  $\mathbf{A}$  'nın en büyük özdeğeridir.

Ve

$$\alpha_k = \frac{\nabla F(\mathbf{x})^T \Big|_{\mathbf{x}=\mathbf{x}_k} \mathbf{p}_k}{\mathbf{p}_k^T \nabla^2 F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k} \mathbf{p}_k} = -\frac{\mathbf{g}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A}_k \mathbf{p}_k} \quad (3.7)$$

Burada  $\mathbf{A}_k = \nabla^2 F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}$  ve  $\mathbf{g}_k^T = \nabla F(\mathbf{x})^T \Big|_{\mathbf{x}=\mathbf{x}_k}$  ' dir.

Aşağıdaki kısımda tanıtılacak olan Levenberg-Marquardt algoritması Newton metodundan türetilmiştir. Bu bakımdan öncelikle Newton metodundan burada kısaca bahsetmenin faydalı olacağı düşünülmüştür. Eğim iniş algoritması  $F(\mathbf{x})$  fonksiyonunun birinci dereceden Taylor serisi ile açılımına dayandırılmıştır. Newton metodu ise bu fonksiyonun ikinci dereceden Taylor serisi ile açılmasına dayandırılmıştır.

$$F(\mathbf{x}_{k+1}) = F(\mathbf{x}_k + \Delta \mathbf{x}_k) \approx F(\mathbf{x}_k) + \mathbf{g}_k^T \Delta \mathbf{x}_k + \frac{1}{2} \Delta \mathbf{x}_k^T \mathbf{A}_k \Delta \mathbf{x}_k \quad (3.8)$$

Newton metodundaki temel mantık,  $F(\mathbf{x})$  fonksiyonuna olan bu kuadratik yaklaşımın denge noktasını bulmaktır.  $\Delta \mathbf{x}_k$  ' ya göre bu kuadratik fonksiyonun gradienti alınıp sifira eşitlenirse şu elde edilir:

$$\mathbf{g}_k + \mathbf{A}_k \Delta \mathbf{x}_k = 0 \quad (3.9)$$

$\Delta \mathbf{x}_k$  yalnız bırakılırsa:

$$\Delta \mathbf{x}_k = -\mathbf{A}_k^{-1} \mathbf{g}_k \quad (3.10)$$

Dolayısıyla Newton metodunun tanımı şu şekilde yapılır.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k \quad (3.11)$$

Newton metodu her zaman kuadratik bir fonksiyonun minimum noktasını tek bir adımda bulacaktır. Bunun nedeni metodun kuadratik fonksiyon temelli olmasıdır. Eğer  $F(\mathbf{x})$  fonksiyonu kuadratik değilse minimum noktaya genellikle tek bir adımda yaklaşamayacaktır. Bu durumda aslında hiçbir zaman minimum noktaya yaklaşılacağından emin olunmaz, bu fonksiyona ve başlangıç değerine bağlı olacaktır (Nguyen ve Widrow, 1990:21-26). Bununla birlikte Newton metodu çoğu uygulamalarda hızlı bir şekilde yakınsama sağlamaktadır. Çünkü incelenen analitik fonksiyonun minimum noktasına yakın komşulukta doğru bir şekilde yaklaşılabilir. Ayrıca Newton metodu kuadratik fonksiyona yaklaşımı temel aldığı için yerel ve global minimum arasındaki farklılığı ayırt edememektedir. Bu bakımdan başlangıç noktası metodun doğru sonuca ulaşmasında önemli bir rol oynamaktadır.

Newton metodu eğim inişe göre daha hızlı yakınsama sağlamasına rağmen Newton metodunun davranışı oldukça karmaşık olabilmektedir. Algoritmanın denge noktasına ulaşmaması da söz konusu olabilmektedir. Halbuki eğim iniş metodunda, öğrenme oranı çok büyük tutulmaz veya her aşamada doğrusal minimizasyon uygulanırsa yakınsama garanti edilmektedir. Newton metoduyla ilgili diğer bir

problem ise Hessian matrisinin ve onun tersinin her seferinde hesaplanmasının zorluğudur.

### 3.2.2.1. Eşlenik Gradient

Newton metodu sonlu sayıdaki iterasyonla kuadratik bir fonksiyonu minimize etme özelliğine sahiptir. Fakat bu ikinci türevlerin hesaplanması ve saklanması gerektirmektedir. Parametre sayısı fazlayken ikinci türevlerin hesaplanması pratik olmamaktadır. Özellikle büyük bir ağdaki her bir ağırlık değeri düşünüldüğünde bu daha bir önem kazanmaktadır. Bu sebeple sadece birinci türevlerin hesaplanmasını gerekli kılan fakat aynı zamanda Newton metodunun özelliğine sahip bir metodun geliştirilmek istenmesi eşlenik gradient algoritmasının doğuş nedenidir.

$\mathbf{p}_k$  vektörlerinden oluşan bir küme aşağıdaki koşulu sağlamasıyla pozitif tanımlı Hessian matrisine göre karşılıklı olarak eşlenikdir (Pedregal, 2004:119).

$$\mathbf{p}_k^T \mathbf{A} \mathbf{p}_j = 0 \quad k \neq j \quad (3.12)$$

$\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$  eşlenik yön setinin herhangi bir doğrusal dizisinin oluşturulmasıyla n parametrelili kuadratik fonksiyonun gerçek minimumuna en fazla n aramayla ulaşılacağı gösterilebilir (Gill, Murray ve Wright, 1981:149). Asıl sorun Hessian matrisini kullanmadan eşlenik arama yön setinin nasıl oluşturulacağıdır. Kuadratik fonksiyonun birinci ve ikinci türevinin (3.13) ve (3.14) 'deki gibi olduğu bilinmektedir.

$$\nabla F(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{d} \quad (3.13)$$

$$\nabla^2 F(\mathbf{x}) = \mathbf{A} \quad (3.14)$$

Bu iki eşitliğin birleştirilmesiyle k+1 inci iterasyonda gradient deki değişim bulunur.

$$\Delta \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k = (\mathbf{A}\mathbf{x}_{k+1} + \mathbf{d}) - (\mathbf{A}\mathbf{x}_k + \mathbf{d}) = \mathbf{A}\Delta \mathbf{x}_k \quad (3.15)$$

(3.5)' den şu bilinmektedir.

$$\Delta \mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k) = \alpha_k \mathbf{p}_k \quad (3.16)$$

Artık eşlenik koşulunda şöyle bir düzenleme yapılabilir.

$$\alpha_k \mathbf{p}_k^T \mathbf{A} \mathbf{p}_j = \Delta \mathbf{x}_k^T \mathbf{A} \mathbf{p}_j = \Delta \mathbf{g}_k^T \mathbf{p}_j = 0 \quad k \neq j \quad (3.17)$$

Bu noktadan sonra Hessian matrisinin bilinmesi gerekliliği ortadan kalkmıştır. Eşlenik olma koşulu algoritmanın ardışık iterasyonlarındaki gradient değişimleriyle sağlanmaktadır. Arama yönü, gradientteki değişime ortogonale eşlenik olacaktır.

İlk arama yönü  $\mathbf{p}_0$  keyfidir ve  $\mathbf{p}_1, \Delta \mathbf{g}_0$  ' a ortogonal herhangi bir vektör olabilir. Böylece sonsuz bir eşlenik vektör set sayısı bulunmaktadır. Genellikle aramaya eğim iniş yönünde başlanır.

$$\mathbf{p}_0 = -\mathbf{g}_0 \quad (3.18)$$

Daha sonra her iterasyonda  $\{\Delta \mathbf{g}_0, \Delta \mathbf{g}_1, \dots, \Delta \mathbf{g}_{k-1}\}$  ' e ortogonal bir  $\mathbf{p}_k$  ' nın oluşturulmasına ihtiyaç duyulur. Bu noktada aşağıda basitleştirilmiş şekli verilen Gram-Schmidt ortogonalleştirmeye benzer bir prosedür kullanılır (Hagan, Demuth ve Beale, 1996:9-17).

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1} \quad (3.19)$$

$\beta_k$  skaleri kuadratik fonksiyonlar için aynı sonucu veren farklı birkaç metoda göre seçilebilmektedir.

Hestenes ve Steifel' e göre

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\Delta \mathbf{g}_{k-1}^T \mathbf{p}_{k-1}} \quad (3.20)$$

Fletcher ve Reeves' e göre

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (3.21)$$

Son olarak Polak ve Ribiere' e göre

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (3.22)$$

Özetlemek gerekirse eşlenik gradient metodu aşağıdaki adımlardan oluşmaktadır.

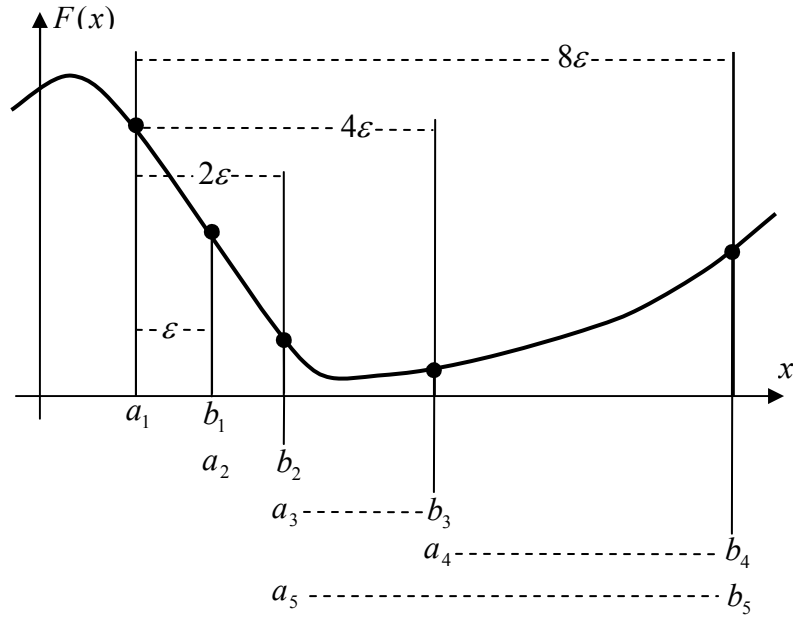
1.  $\mathbf{p}_0$  ilk arama yönünü (3.18) 'da olduğu gibi gradientin negatifi olarak seçilir.
2. Arama yönü boyunca fonksiyonu minimize edecek  $\alpha_k$  öğrenme oranı kuadratik fonksiyonlar için (3.7) 'e göre bulunarak, eşitlik (3.16) 'a göre bir sonraki adım belirlenir.
3.  $\beta_k$  'yı hesaplamak için (3.20), (3.21) veya (3.22) kullanılarak bir sonraki arama yönü eşitlik (3.19) 'a göre bulunur.
4. Algoritmanın yakınsaması tamamlanmazsa Adım2 ye dönülerek aynı süreç tekrarlanır.

Yukarıda adımları verilen eşlenik gradient algoritmasının direkt olarak sinir ağları eğitimine uygulanabilmesi için performans indeksi formülasyonunun kuadratik

olması gerekmektedir. Halbuki çok katmanlı ağlarda kullanılan performans indeksi kuadratik değildir. Bu algoritmayı iki yönde etkilemektedir. Birincisi Adım2 de kuadratik fonksiyonlar için tanımlanmış (3.7) doğru boyunca minimizasyon formülü kullanılmayacaktır. İkinci olarak ise algoritmanın sonlu sayıdaki adımla ulaşacağı gerçek minimuma ulaşamayabilecektir. Bu yüzden belirli bir iterasyon sayısından sonra arama yönünün yeniden belirlenmesi ihtiyacı ortaya çıkacaktır.

İlk olarak doğru boyunca minimizasyon formülünün kuadratik olmayan fonksiyonlar için geliştirilmesini ele alalım. Belirlenen bir doğrultuda bir fonksiyonun minimum noktasına yerleşecek genel bir prosedüre ihtiyaç vardır. Bu prosedür aralık belirlenmesi (interval location) ve aralık düşürülmesi (interval reduction) adımlarından oluşmaktadır. Aralık belirlenmesi adımının amacı yerel bir minimumu içeren başlangıç aralığının bulunmasıdır. Aralık düşürmenin ise başlangıç aralığının büyüklüğünü minimum noktayı içeren istenen aralık boyutuna kadar düşürmektir.

Aralık belirleme adımını yerine getirmek için fonksiyon karşılaştırma metodu kullanılacaktır. Bu prosedüre Şekil 3-1 de gösterilmiştir. Şekilde  $a_1$  ile gösterilen bir başlangıç noktasında performans indeksinin değerlendirilmesiyle başlanır. Bu nokta ağırlık o andaki sapma ve ağırlık değerlerine karşılık gelmektedir. Başka bir deyişle  $F(\mathbf{x}_0)$  değerlendirilir. Sonraki adım şekilde  $b_1$  ile gösterilen ikinci bir noktada fonksiyonu değerlendirmektir. İkinci nokta başlangıç noktasından,  $\mathbf{p}_0$  ilk arama yönü boyunca  $\varepsilon$  kadar bir uzaklıkta bulunmaktadır. Başka bir deyişle  $F(\mathbf{x}_0 + \varepsilon \mathbf{p}_0)$  değerlendirilir.

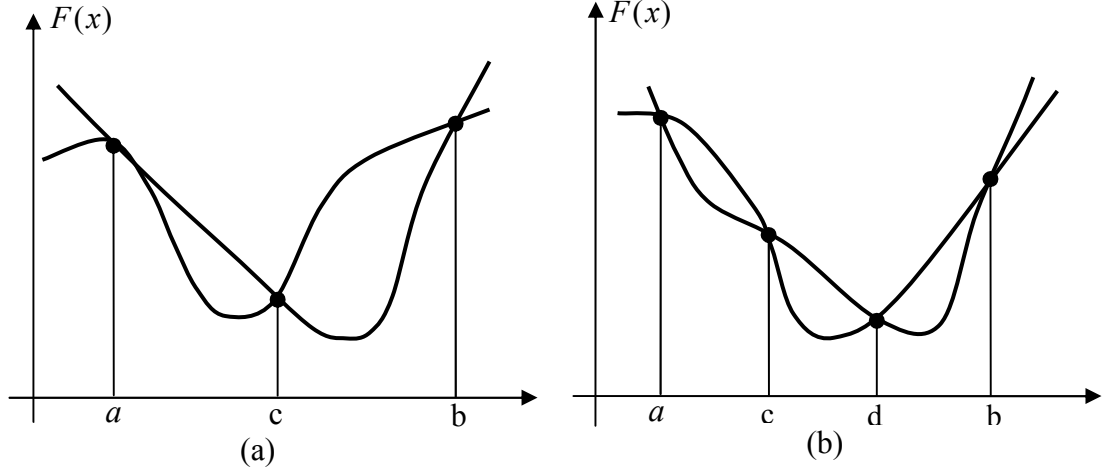


**Şekil 3- 1: Aralık Belirleme**

Noktalar arasındaki uzaklık ardışık olarak iki katına çıkarılarak, performans indeksinin yeni  $b_1$  noktalarını da değerlendirilmesine devam edilir. Bu süreç ardışık iki değerlendirme arasında fonksiyon değeri artınca durdurulur. Şekil 3-1 de bu durum  $b_3$ ' den  $b_4$ ' e noktalarıyla gösterilmiştir. Artık minimumun  $a_5$  ve  $b_5$  noktaları arasında sınırlandırıldığı bilinmektedir. Minimum nokta  $[a_4, b_4]$  veya  $[a_3, b_3]$  aralığında olabileceğinden artık daha fazla daraltılamaz. Bu iki olasılık Şekil 3-2 'nin (a) kısmında gösterilmiştir.

Minimum noktayı içeren aralığın belirlenmesiyle sonraki adım aralık düşürülmesidir. Buna aralık belirleme adımında seçilen  $[a_5, b_5]$  aralığının içindeki noktalarda fonksiyonun değerlendirilmesiyle başlanır. Şekil 3-2 'den görüldüğü gibi aralık boyutunu düşürmek amacıyla en az iki iç noktada fonksiyonun değerlendirilmesine ihtiyaç duyulur. Fonksiyon değerlendirilmesinin bir iç noktada yapılması Şekil 3-2 (a) görüldüğü gibi minimum noktanın belirlenmesinde hiçbir bilgi sağlamamaktadır. Bununla beraber fonksiyon Şekil 3-2 (b) 'deki gibi c ve d noktalarında değerlendirilirse aralığın boyutu düşürülebilir. Eğer  $F(c) > F(d)$ , Şekil 3-2 (b), ise minimum  $[c, b]$  aralığındadır. Tersi  $F(c) < F(d)$  ise minimum  $[a, d]$

aralığındadır. Burada başlangıç aralığı içerisinde tek bir minimum olduğu varsayımı yapılmıştır.



**Şekil 3- 2: Aralık Büyüklüğünün Düşürülmesi**

Aralık büyüklüğünün düşürülmesi için yukarıda tanımlanan metottan sonra şimdi sıra c ve d iç noktalarının yerlerinin belirlenmesine gelmiştir. Bunun yapılabilmesi için birkaç yol bulunmaktadır. Burada gerekli fonksiyon değerlendirilmeleri sayısını düşürmek için tasarlanan Altın Bölme (Golden Section) arama metodu ele alınmıştır. Her iterasyonda bir tane yeni fonksiyon değerlendirmesi yeterli olmaktadır. Örneğin Şekil 3-2 (b) 'de a noktası çıkarılır ve c noktası dış nokta olur. Daha sonra yeni bir c noktası orijinal c ve d noktaları arasında yerleştirilir. Bu şekilde yeni noktanın oluşturulmasıyla aralığın büyüklüğü mümkün olduğunca çabuk şekilde azaltılmaya çalışılır.

Aşağıda Altın Bölme arama algoritması bilgisayar kodu şeklinde orijinalliği bozulmadan verilmiştir (Hagan, Demuth ve Beale, 1996:12-17).

$$\tau = 0.618$$

$$\text{Set } c_1 = a_1 + (1 - \tau)(b_1 - a_1), F_c = F(c_1)$$

$$d_1 = b_1 + (1 - \tau)(b_1 - a_1), F_d = F(d_1)$$



For  $k = 1, 2, \dots$  repeat

    If  $F_c < F_d$  then

$$\begin{aligned} \text{Set } & a_{k+1} = a_k; b_{k+1} = d_k; d_{k+1} = c_k \\ & c_{k+1} = a_{k+1} + (1 - \tau) (b_{k+1} - a_{k+1}) \\ & F_d = F_c; F_c = F(c_{k+1}) \end{aligned}$$

    else

$$\begin{aligned} \text{Set } & a_{k+1} = c_k; b_{k+1} = b_k; c_{k+1} = d_k \\ & d_{k+1} = b_{k+1} - (1 - \tau) (b_{k+1} - a_{k+1}) \\ & F_c = F_d; F_d = F(d_{k+1}) \end{aligned}$$

    end

end until  $b_{k+1} - a_{k+1} < tol$

Burada  $tol$  kullanıcı tarafından belirlenen doğruluk tolerans ayarıdır.

Artık eşlenik gradient algoritmasını sinir ağı eğitimine uygulamadan önce son bir ayarlamamanın yapılması kalmıştır. Kuadratik fonksiyonlarda  $n$  optimize edilecek parametre sayısı iken algoritma en fazla  $n$  iterasyonda minimum noktaya ulaşacaktır. Çok katmanlı ağlar için performans indeksi kuadratik olmadığından doğal olarak algoritma minimum noktaya  $n$  iterasyonda ulaşamayacaktır. Bununla birlikte eşlenik gradient algoritması geliştirilirken  $n$  iterasyonluk bir döngü tamamlandıktan sonra arama yönünün ne seçileceği konusunda bir şey söylenmemiştir. Geleneksel olarak  $n$  iterasyon sonunda arama yönü eğim iniş yönünde (gradientin negatifi) seçilmektedir.

### 3.2.2.2. Levenberg-Marquardt Algoritması

Newton metodundan türetilen Levenberg-Marquardt algoritması doğrusal olmayan fonksiyonların karesel toplamlarını minimize etmek amacıyla tasarlanmıştır. Buradan hareketle performans indeksi ortalama karesel hata olan sinir ağları eğitiminde uyarlanabileceği fikri doğmuştur.

Daha önceden bilindiği üzere  $F(\mathbf{x})$  performans indeksini optimize eden Newton metodu şu şekildedir:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k \quad (3.23)$$

Burada  $\mathbf{A}_k = \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$  ve  $\mathbf{g}_k^T = \nabla F(\mathbf{x})^T|_{\mathbf{x}=\mathbf{x}_k}$  'dir.

$F(\mathbf{x})$  'in karesel fonksiyonların bir toplamı olduğunu varsayılırsa:

$$F(\mathbf{x}) = \sum_{i=1}^N v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x}) \mathbf{v}(\mathbf{x}) \quad (3.24)$$

Bu takdirde gradientin j inci elemanı şöyle olur:

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j} \quad (3.25)$$

Böylece gradient matris formunda yazılabilir:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \quad (3.26)$$

Burada  $\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix}$  Jacobian matrisidir.

Sonraki adım Hessian matrisinin bulunmasıdır. Hessian matrisinin  $k, j$  inci elemanı şöyle olur:

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(\mathbf{x})}{\partial x_k} \frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x}) \frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \right\} \quad (3.27)$$

Hessian matrisi böylece matris formunda ifade edilebilir.

$$\nabla^2 \mathbf{F}(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \quad (3.28)$$

Burada  $\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x})\nabla^2 v_i(\mathbf{x})$  'dir.

$\mathbf{S}(\mathbf{x})$  'in küçük olduğu varsayırsa Hessian matrisine şöyle yaklaşılabilir:

$$\nabla^2 \mathbf{F}(\mathbf{x}) \equiv 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \quad (3.29)$$

Artık eşitlik (3.29) ve (3.26) 'yı eşitlik (3.23) içersine yerleştirilirse Gauss-Newton metodu elde edilir.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - [2\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} 2\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \end{aligned} \quad (3.30)$$

Dikkat edilirse Gauss-Newton' un standart Newton metoduna olan üstünlüğü ikinci türevlerin hesaplanmasına gerek duymamasıdır. Gauss-Newton metoduyla ilgili bir problem  $\mathbf{H} = \mathbf{J}^T\mathbf{J}$  matrisinin tersinin bulunamaması durumudur. Bunun üstesinden yaklaşık Hessian matrisine aşağıdaki değişiklikle gelinir.

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I} \quad (3.31)$$

Bu matrisin tersinin nasıl oluşturulabileceğini görmek amacıyla  $\mathbf{H}$  matrisinin özdeğer ve özvektörlerinin sırasıyla  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$  ve  $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$  olduğu varsayalım. Böylece (3.32) eşitliği yazılabilir.

$$\mathbf{G}\mathbf{z}_i = [\mathbf{H} + \mu\mathbf{I}]\mathbf{z}_i = \mathbf{H}\mathbf{z}_i + \mu\mathbf{z}_i = \lambda_i\mathbf{z}_i + \mu\mathbf{z}_i = (\lambda_i + \mu)\mathbf{z}_i \quad (3.32)$$

Şimdi  $\mathbf{G}$  matrisinin özvektörleri  $\mathbf{H}$  matrisininkiyile aynıdır ve  $\mathbf{G}$ ' nin özdeğerleri ise  $(\lambda_i + \mu)$ ' dir.  $\mathbf{G}$  matrisi tüm  $i$  değerleri için  $(\lambda_i + \mu)$  oluncaya kadar  $\mu$ ' nin artırılmasıyla pozitif tanımlı yapılabılır böylece tersi de alınabilecektir.

Bu durum Levenberg-Marquardt algoritmasının ortaya çıkmasına neden olacaktır (Gill, Murray ve Wright, 1981:136).

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \quad (3.33)$$

Veya

$$\Delta\mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \quad (3.34)$$

Bu algoritma oldukça yararlı bir özelliğe sahiptir,  $\mu_k$  artarken küçük öğrenme oranına sahip eğim iniş algoritmasına yaklaşılmaktadır.  $\mu_k$  değeri büyük değerler aldığıında aşağıdaki formülasyon kullanılabilir.

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k}\nabla F(\mathbf{x}) \quad (3.35)$$

$\mu_k$  değeri sıfıra yaklaştıkça algoritma Gauss-Newton halini almaktadır. Algoritmaya  $\mu_k$  'nın küçük bir değere ( $\mu_k = 0.01$ ) atanmasıyla başlanılır. Eğer bir adım sonra daha küçük bir  $F(\mathbf{x})$  değerine ulaşılmazsa aynı adım  $\mu_k$  'nın  $\mathcal{G} > 1$  ( $\mathcal{G} = 10$ ) katsayısıyla çarpılmasıyla tekrarlanır. Bu işlemin tekrarlı bir şekilde yapılmasıyla en sonunda eğim iniş yönünde küçük bir adımın atılmasına sebep olunacaktır ve böylece  $F(\mathbf{x})$  değerinde düşüş meydana gelecektir.  $F(\mathbf{x})$  için küçük değer elde edilmesi adımından sonra  $\mu_k$  değeri  $\mathcal{G}$  çarpanıyla bölünecektir. Böylece algoritma daha hızlı şekilde sonuca ulaşmayı sağlayan Gauss-Newton metoduna yaklaşacaktır. Sonuç olarak algoritma Newton metodunun hızı ve eğim

inişin minimum noktayı garanti etme özelliği arasında bir denge noktası sağlayarak etkin bir yöntem olma özelliğine sahiptir.

Artık sıra Levenberg-Marquardt algoritmasının çok katmanlı ağı eğitimi sorununa nasıl uygulanabileceğinin araştırılmasına gelmiştir. Çok katmanlı ağı eğitimi için performans indeksi ortalama karesel hatadır. Ortalama karesel hata eğitim setindeki  $Q$  tane hedefin karesel hatalarının toplamından bulunur.

$$\begin{aligned}
 F(\mathbf{x}) &= \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) \\
 &= \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2
 \end{aligned} \tag{3.36}$$

burada  $e_{j,q}$ ,  $q$  inci girdi-hedef çifti için  $j$  inci bileşenin hatasıdır.

Eşitlik (3.36) Levenberg-Marquardt algoritması için tasarlanan eşitlik (3.24) performans indeksinin benzeridir. Dolayısıyla algoritmanın ağı eğitime uyarlanmasının kolay bir şekilde olacağı düşünülse de ayrıntılarda dikkat edilmesi gereken noktalar vardır.

Levenberg-Marquardt algoritmasındaki önemli nokta Jacobian matrisinin hesaplanmasıdır. Temel geriyayılım prosedüründe karesel hataların ağı sapma ve ağırlık değerlerine göre türevleri hesaplanırdı. Jacobian matrisini oluşturmak için ise karesel hataların türevlerinin yerine hataların türevlerinin hesaplanması gerekmektedir. Jacobian 'ın hesaplama prosedürüne geçmeden önce aşağıda sırasıyla Jacobian matrisi tanımlanırken gösterilen hata vektörü ve parametre vektörü verilmiştir.

$$\mathbf{v}^T = [v_1 \ v_2 \ \dots \ v_N] = [e_{1,1} \ e_{2,1} \ \dots \ e_{S^M,1} \ e_{1,2} \ \dots \ e_{S^M,Q}] \tag{3.37}$$

$$\mathbf{x}^T = [x_1 \ x_2 \ \dots \ x_n] = [w_{1,1}^1 \ w_{1,2}^1 \ \dots \ w_{S^1,R}^1 \ b_1^1 \ \dots \ b_{S^1}^1 \ w_{1,1}^2 \ \dots \ b_{S^M}^M] \quad (3.38)$$

Burada  $N \approx Q \times S^M$  ve  $n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1)$ .

Bu elemanların (3.35) içersine yerleştirilmesiyle çok katmanlı ağıın eğitimi için Jacobian matrisi şu şekilde yazılabilir:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \end{bmatrix} \quad (3.39)$$

Jacobian matrisindeki terimler geri yayılım algoritmasında küçük bir değişikliğin yapılmasıyla hesaplanabilmektedir. Temel geri yayılım ile aşağıdaki gibi terimler hesaplanıyordu.

$$\frac{\partial \hat{F}(\mathbf{x})}{\partial x_l} = \frac{\partial \mathbf{e}_q^T \mathbf{e}_q}{\partial x_l} \quad (3.40)$$

Levenberg-Marquardt algoritması için gerekli olan Jacobian matrisinin her bir elemanını elde etmek için aşağıdaki gibi terimlerin hesaplanmasına ihtiyaç duyulmaktadır.

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l} \quad (3.41)$$

Geriyayılımın oluşturulmasında daha önce (2.48) olarak verilen ve aşağıda tekrar verilen eşitlikten yararlanılmaktaydı.

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (3.42)$$

Eşitliğin sağ tarafındaki ilk terim duyarlılık olarak tanımlanmıştı:

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m} \quad (3.43)$$

Geriyayılım sürecinde duyarlılıklar son katmandan ilk katmana doğru yinelemeli bir ilişki ile hesaplanmaktaydı. Eğer yeni bir Marquardt duyarlılığı tanımlanırsa aynı mantıkla Jacobian matrisi için gerekli olan terimler hesaplanabilir:

$$\bar{s}_{i,h}^m \equiv \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (3.44)$$

Burada eşitlik (3.37) 'den  $h = (q - 1)S^M + k$  'dır.

Artık Jacobian matrisinin elemanlarını aşağıdaki (3.45) ve (3.46) eşitlikleriyle hesaplanabilir.

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \bar{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \bar{s}_{i,h}^m \times a_{j,q}^{m-1} \quad (3.45)$$

Veya  $x_l$  bir sapma ise

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial b_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \bar{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \bar{s}_{i,h}^m \quad (3.46)$$

Marquardt duyarlılıkları standart duyarlılıkların hesaplandığı eşitlik (2.66) deki gibi hesaplanabilmektedir, tek farklılık eşitlik (2.71) ile hesaplanan son katmandan kaynaklanmaktadır. Marquardt duyarlılıkları son katmanda şuna eşittir:

$$\begin{aligned} \bar{S}_{i,h}^M &\equiv \frac{\partial v_h}{\partial n_{i,q}^M} = \frac{\partial e_{k,q}}{\partial n_{i,q}^M} = \frac{\partial (t_{k,q} - a_{k,q}^M)}{\partial n_{i,q}^M} = -\frac{\partial a_{k,q}^M}{\partial n_{i,q}^M} \\ &= \begin{cases} -\dot{f}^M(n_{i,q}^M) & \text{for } i = k \\ 0 & \text{for } i \neq k \end{cases} \end{aligned} \quad (3.47)$$

Böylece  $\mathbf{p}_q$  girdisi ağa uygulandığında ona karşılık gelen  $\mathbf{a}_q^M$  ağ çıktısı hesaplanarak Levenberg-Marquardt geriyayılımı eşitlik (3.48) ile başlatılır.

$$\bar{\mathbf{S}}_q^M = -\dot{\mathbf{F}}^M(\mathbf{n}_q^M) \quad (3.48)$$

Burada  $\dot{\mathbf{F}}^M(\mathbf{n}^M)$  eşitlik (2.65) ile tanımlanmıştır.  $\bar{\mathbf{S}}_q^M$  matrisinin her bir sütunu Jacobian matrisinin bir satırını oluşturmak amacıyla eşitlik (2.66) 'yı kullanarak ağ içerisinde geri yayılmalıdır. Sütunlar eşitlik (3.49) 'yı kullanarak da hep beraber geri yayılabilirler.

$$\bar{\mathbf{S}}_q^m = -\dot{\mathbf{F}}^m(\mathbf{n}_q^m)(\mathbf{W}^{m+1})^T \bar{\mathbf{S}}_q^{m+1} \quad (3.49)$$

Her katman için toplam Marquardt duyarlılıkları matrisi her girdi için hesaplanan matrislerin birleştirilmesiyle elde edilir.

$$\bar{\mathbf{S}}^m = \left[ \bar{\mathbf{S}}_1^m \mid \bar{\mathbf{S}}_2^m \mid \dots \mid \bar{\mathbf{S}}_Q^m \right] \quad (3.50)$$

Her bir girdi ağa sunuldukça  $S^M$  duyarlılık vektörü geri yayılacaktır. Bunun nedeni hata kareler toplamının türevi yerine her bir hatanın türevlerinin hesaplanmasıdır. Ağ sunulan her girdi için ağın çıktı birimlerinin her birine karşılık gelecek şekilde  $S^M$  tane hata oluşacaktır. Bu hatalar Jacobian matrisinin her bir



satırına karşılık gelecektir. Duyarlıkların geri yayılmasından sonra eşitlik (3.45) ve eşitlik (3.46) kullanılarak Jacobian matrisi hesaplanır.

Levenberg-Marquardt geriyayılım algoritmasının iterasyonları aşağıdaki gibi özetlenebilir (Hagan, Demuth ve Beale, 1996:12-25).

1. Tüm girdiler ağa sunulur, eşitlik (2.72) ve eşitlik (2.73) kullanılarak bunlara karşılık gelen ağ çıktıları ve  $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$  hataları hesaplanır. Eşitlik (3.36) kullanılarak  $F(\mathbf{x})$  hata kareler toplamı hesaplanır.
2. Jacobian matrisinin, eşitlik (3.39), hesaplanmasına gelinmiştir. Öncelikle Levenberg-Marquardt geriyayılımı eşitlik (3.48) ile başlatıldıktan sonra eşitlik (3.49) ile duyarlılıkların hesaplanmasına geçilir. Her bir girdi vektörü için hesaplanan duyarlılık matrisleri eşitlik (3.50) 'in kullanılmasıyla birleştirilerek toplam Marquardt duyarlılığı elde edilir. Böylece eşitlik (3.45) ve (3.46) ile Jacobian matrisinin elemanları hesaplanır.
3.  $\Delta \mathbf{x}_k$  'yı elde etmek amacıyla eşitlik (3.34) çözülür.
4.  $\mathbf{x}_k + \Delta \mathbf{x}_k$  kullanılarak hata kareler toplamı yeniden hesaplanır. Eğer bu yeni kareler toplamı adım1 de hesaplanandan daha küçükse o zaman  $\mu$ ,  $\mathcal{G}$  'ya bölünür ve  $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$  olarak alınır. Eğer kareler toplamı azalmasa bu takdirde  $\mu$ ,  $\mathcal{G}$  ile çarpılır ve adım3 'e geri dönülür.

Algoritma, gradientin normunun önceden belirlenen bir değerden küçük olması veya kareler toplamının belirli hata hedefinin aşağısına düşürülmesi durumunda minimum noktaya ulaşacağını varsayar. Algoritma şuana kadar tanıtılan metotlara göre daha az iterasyonla minimum noktaya ulaşmasına rağmen matris tersi içerdiğinden, diğer metotlara göre her iterasyonda daha çok hesaplama yapılmasını gerektirmektedir. Orta büyüklükteki ağ parametre sayısı için Levenberg-Marquardt

geriyayılım algoritmasının en hızlı sinir ağı eğitim algoritması olduğu gözlemlenmiştir (Hagan ve Menhaj, 1994:991).

Levenberg-Marquardt algoritmasının dezavantajı  $\mathbf{J}^T\mathbf{J}$  yaklaşık Hessian matrisinin her seferinde saklanması gereğidir. Bu matrisin boyutu  $n$  ağıdaki parametre sayısı iken  $n \times n$  'dir. Daha önce tanıtilen metotlar sadece  $n$  boyutlu vektör olan gradientin hesaplanmasına ihtiyaç duyuyordu. Dolayısıyla parametre sayısı çok büyük olduğunda Levenberg-Marquardt algoritmasını kullanmak pratik olmaktan çıkmaktadır. Buradaki çok büyük kelimesi kullanılan bilgisayarın hafızasına bağlı olmaktadır fakat genellikle birkaç yüz parametre üst limittir.

### 3.3. Sinir Ağlarında Duyarlılık Analizleri

Temel geri yayılım algoritmasının varyasyonları yüksek derecede doğrusal olmayan girdi-çıkı eşleşmesi yapabilme, gizli desenleri ve tahminleyici bilgiyi öğrendiği örneklerden gözlemleyerek çıkarabilme yeteneğine sahiptirler. Bununla birlikte bu algoritmalar nedensellik bilgisi hakkında bilgi vermezler. Çünkü tahminleme veya sınıflandırma süreci tahminlenen çıktı değişkeni üzerinde etkili olan değişken veya her bir girdi parametresinin etkisi hakkında bilgi vermeksizin önceden analiz edilen örnekler ve yeni örnekler arasındaki bilinmeyen bir korelasyona dayanır (Lu, AbouRizk ve Hermann, 2001:299).

Yapılan sinir ağları uygulamalarında modelin değerlendirilmesi öğrenme sürecinde sinir ağından saklanan bağımsız bir test veri seti üzerinde nihai ağın doğruluk ölçümüne bağlı tutulmuştur. Parametrelerdeki değişikliğe karşı modelin duyarlılığı genellikle nihai ağın çeşitli girdi senaryolarına verdiği cevabın incelenmesiyle araştırılmıştır.

Sinir ağı modelleri, model çıktılarının veya cevapların nasıl elde edilmiş olduğu ve çıktı üzerinde girdi parametrelerinin etkisi hakkında bilgi vermeyerek bir kara kutu fonksiyonelliği göstermektedirler. Yorum yapabilme yeteneği eksikliği sinir ağı modelleri için bir engel oluşturmakta ve sinir ağlarının gerçek dünya problemlerinde sahip olduğu potansiyeli tam olarak gösterememesine neden

olmaktadır. Sinir ağlarının paralel olarak dağıtılmış veri işleme özelliğine sahip doğrusal olmayan yüksek derecedeki fonksiyonel eşitlikleri, bir sinir ağı modelinde hangi parametrenin ne çeşit bir davranışa yol açtığı belirlenmesini zorlaştırmaktadır. Bu analiz için var olan matematiksel metotlar oldukça karmaşık olup buldukları teorik varsayımlardan ötürü tatmin edicilik sağlamamaktadırlar.

Literatürde bu konuya yönelik olarak yapılan çalışmaları kabaca iki kategoriye ayırma imkanı vardır. Birincisi mantık süzgecinden geçirilmiş fakat matematiksel altyapısı çok sağlam olmayan sezgisel yöntemlerdir. Sezgisel yaklaşımları anlamak ve kullanmak kolay olmasına rağmen sonuçların her zaman tutarlı ve doğru olacağı konusunda bir kesinlik yoktur. İkinci kategoriye ise temelleri yoğun istatistiksel analizlere dayandırılan teknikler oluşturmaktadır. Bu kategorideki teknikler veriler üzerinde çok sayıdaki varsayımlar ile yöntemin kullanılabilirliğini ve özgürlüğünü kısıtlamaktadırlar. Böylece yöntemin en önemli özelliklerinden biri olan varsayım gerektirmemesinden feda edilmiş olmaktadır.

Bu çalışmanın temel amacı da bu noktada ortaya çıkmaktadır. Çalışma YSA'nın işlerliği hakkında varsayımlar bulunmaksızın fakat matematiksel temeli olan bir yöntem geliştirilmesine odaklanmıştır. Tanıtılacak olan yöntem eğitim algoritması olarak geri yayılım algoritmasının kullanıldığını varsayarak geliştirilmiştir. Bilindiği üzere geri yayılım algoritması çıktı sonuçlarına göre bağlantı ağırlıklarını değiştirmek amacıyla matematikteki zincir kuralının ardışık olarak kullanılmasından türetilmiştir. Burada zincir kuralından bir kez daha yararlanılmış yalnız bu sefer çıktının girdilere göre değişiminin incelenmesinde kullanılmıştır.

### **3.3.1. Literatür**

Sinir ağları literatüründe sıklıkla karşılaşılmamasına rağmen duyarlılık analizi adı altında yapılan girdi ve çıktı değerlerini ilişkilendirme çalışmaları genel kapsamda ikiye ayrılmaktadır. Bunlar sezgisel ve çeşitli varsayımlara dayanan yoğun istatistiksel metotlardır. Daha önce belirtildiği üzere iki tür metodun da kullanım aşamasında zorlukları bulunmaktadır. Sezgisel yaklaşımların uygulamasında basitlik mevcut olmasına rağmen elde edilen sonucun kesinliğinden emin olunamazken

istatistiksel yöntemlere dayandırılan metotlar ise kullanım alanını kısıtlayan varsayımlara sahiptir. Aşağıda bu konuda yapılmış birkaç çalışmadan kısaca bahsedildikten sonra bu çalışmada temel alınan makale hakkında bilgi verilip bu makaleye duyarlılık analizinin geliştirilmesi kapsamında ne gibi önerilerin getirilmiş olduğuna sonuç kısmında yer verilmiştir.

Genellikle yapılan duyarlılık analizi çalışmalarında girdi vektöründe yapılan değişimlere çıktının nasıl değişerek sonuçlar verdiği araştırılmıştır. Ayrıca yine duyarlılık analizi adı altında optimal ağ tasarımını oluşturmak amacıyla etkisi az olan ağırlık değerleri çıktı hatasının sonuçları etkilenmeyecek şekilde ağdan atılmaya çalışılmıştır. Burada çıkarılan ağırlık değerlerine çıktının verdiği cevaplar inceleme konusu olmuştur. Başka bir deyişle bu çalışmalardaki amaç çıktı hatası sonuçlarını etkilemeden en az düzeyde karmaşık ağ tasarımının oluşturulmasıdır. Duyarlılık analizi başlığında yapılan bir başka çalışma türü girdi vektöründeki hata ve bozulmalara karşı ağın yinede makul cevaplar üretebilme sınırının testidir. Ancak bu çalışmadaki amaç sözü edilen duyarlılık çalışmaları ile aynı kategoride değildir. İlgilenilen konu hangi girdinin çıktı üzerinde ne kadar etkisinin belirlenmesine yöneliktir.

Li ve arkadaşları (1999) yapay sinir ağının çıktısını açıklamadaki yetersizliğinin kullanıcı tarafından yöntemin kabul edilebilirliğini olumsuz yönde etkilemesine neden olduğunu fark etmişlerdir. Bunu gidermek amacıyla bir karar destek aracı olarak inşaat sektöründeki fiyat artışlarının nasıl ve ne kadar gerçekleşeceğinin tahminini sezgisel bir metoda dayalı açıklamaya çalışmışlardır. Ancak yazarların da belirttiği üzere geliştirilen metot konu uzmanının bilgisine ihtiyaç duyduğundan tam anlamıyla açıklayıcı bir bilgi sağlamayı garanti etmemektedir (Li, Shen ve Love, 1999:185).

Sinha ve McKim (2000) yapı firmalarının organizasyonel etkinliklerini ölçmek amacıyla bir sinir ağı modeli oluşturmuşlardır. Girdi uzayının boyutunun azaltılması ve çıktı değişkeni üzerindeki etkili faktörlerin belirlenmesine yönelik olarak yapay sinir ağı modeline ana bileşenler analizi, adımsal (stepwise) regresyon ve korelasyon analizi gibi istatistiksel analiz metotları uygulanmıştır. Teorik

bağımlılıklarından ötürü bu gibi istatistiksel metotlar gerçek problemlerin çözümlenmesine uygulanırken dikkatli bir çalışmanın yürütülmesi gerekmektedir. En küçük kareler regresyonunun varsayımlarının gerçekleşmemesi regresyon ve korelasyon analizinin yanlış kullanılabilmesine neden olabilmektedir. Ayrıca çıktı değişkeni üzerinde etkili olan faktörlerin belirlenmesinde ana bileşenler kullanılması değişkenler arasında sadece doğrusal ilişkinin bulunduğunu varsayacak ve doğrusal olmayan ilişkiye sahip olanları analiz kapsamına dahil etmeyecektir (Sinha ve McKim, 2000:9). Görüldüğü gibi yapay sinir ağlarında girdi ve çıktı değerlerinin ilişkilendirilmesi için kullanılan istatistiksel metotlar yöntemin var olan özelliklerini köreltmektedir.

Yao ve arkadaşları (1998) Singapur pazarındaki renkli televizyonların satış performansını etkileyen önemli değişkenleri bulmak için sinir ağlarını pazarlama karar destek sistemi içersine entegre etmeye çalışmışlardır. Singapur' daki renkli tv satışlarına ait tarihsel veriler toplanarak ağın eğitimi tamamlanmış ve ağın girdi değişkenlerinin sayısının azaltılmasına yönelik olarak iki sezgisel metotla girdi biriminin ağırlıklarının analizi yapılmıştır. Ayrıca çıktı üzerinde en etkili olan değişken araştırılmıştır (Yao vd., 1998:523).

Girdiler ve çıktılar arasındaki ilişkinin gücünü karakterize etmek için eğitimi tamamlanmış nihai bir ağın ağırlıkları da incelenebilmektedir. Knowles (1997) tek gizli katmanlı ve eğitimi tamamlanmış bir sinir ağı modelinde bir girdi biriminden her gizli birime olan ağırlıkların mutlak değerini alarak toplamıştır (Knowles, 1997:122). Bir girdi biriminin toplam ağırlık değeri o girdi biriminden ağın gizli katmanına olan bağlantının yoğunluğunu gösterecektir. Böylece ne kadar büyük bir toplam değeri elde edilirse girdi parametresi o kadar önemli sayılacaktır. Her ne kadar bu sezgisel yaklaşımın uygulanması ve anlaması kolay olsa da gizli ve çıktı katmanı arasındaki bağlantılar dikkate alınmadığından sonuçların tutarsız ve yanlış çıkabilme şansı bulunmaktadır.

Literatürdeki incelemeler sonucunda bu çalışmada temel olarak alınan ve üzerine çeşitli önerilerin getirildiği çalışma Lu vd. (2001) tarafından yapılmıştır. Lu vd. geri yayılım algoritmasına dayanan ve ne istatistiksel metotlardaki gibi

varsayımlarla kısıtlamalar getirmeyen ne de sezgisel metotlardaki gibi tutarsız sonuçlar vermeyen bir duyarlılık analizi metodu geliştirmişlerdir. Sonraki kısımda ayrıntılarıyla verilecek metodu küçük bir veri seti üzerinde regresyonla kıyaslamış ve daha sonra gerçek bir uygulamada işlerliğini örneklendirmişlerdir (Lu, AbouRizk ve Hermann, 2001:306).

### 3.3.2. Girdi Duyarlılığı

Geri yayılım algoritmasına dayalı duyarlılık analizinin oluşturulması amacıyla öncelikle daha önceden tanımlanan notasyona birkaç ekleme yapılması gerekmektedir. Bunlar aşağıda sıralanarak verilmiştir:

- $p$  alt indisi ağın önceki katmanındaki bir birime karşılık gelmektedir.
- $c$  alt indisi ağın o anki katmanındaki bir birime karşılık gelirken  $C$  ise o anki katmanda bulunan toplam birim sayısını göstermektedir.
- $n$  alt indisi ağın bir sonraki katmanındaki bir birimi göstermektedir.

Eğer o anki birim girdi birimi ise bu takdirde çıktı sinyali şöyle olacaktır:

$$a_c = n_c \quad (3.51)$$

$$n_c = a_p w_{cp} + \sum_{i \neq p} a_i w_{ci} \quad (3.52)$$

$$\frac{\partial n_c}{\partial a_p} = w_{cp} \quad (3.53)$$

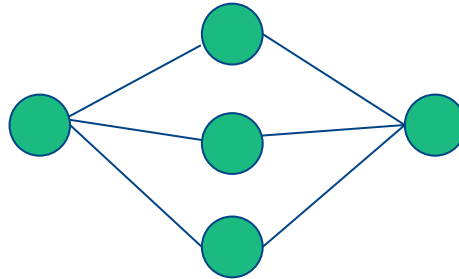
Dikkat edilirse eşitlik (3.52) önceki katmandaki bir birimin diğerlerinden ayrıştırılması haricinde bir hücreye gelen net girdiyi tanımlamaktadır. Bir birimin net girdisi ve çıktısı arasındaki ilişkiyi doğrusal olmayan transfer fonksiyonu

belirlemektedir. Transfer fonksiyonu olarak Sigmoid kullanıldığında aşağıdaki eşitlik (3.54) 'ün var olduğu bilinmektedir.

$$\frac{\partial a_c}{\partial n_c} = a_c(1 - a_c) \quad (3.54)$$

Şekil 3-3 'de gösterildiği gibi üç katmanlı bir ağda  $p$  birimi girdi katmanındaki girdi birimine,  $c$  birimi orta katmandaki gizli birime,  $n$  birimi çıktı katmanındaki çıktı birimine karşılık gelmektedir. Duyarlılık analiziyle incelenen nokta  $n$  birimindeki çıktı sinyalinin ( $a_n$ )  $p$  birimindeki girdi sinyali ( $n_p$ ) üzerine değişiminin incelenmesi üzerine matematikteki zincir kuralından yararlanılmasıdır. Analiz gizli katmandaki birimlerde Sigmoid fonksiyonunun ve çıktı katmanında ise doğrusal fonksiyonun kullanıldığı varsayımıyla geliştirilmiştir.

$$\frac{\partial a_n}{\partial n_p} = \frac{\partial a_n}{\partial a_p} = \frac{\partial a_n}{\partial a_c} \cdot \frac{\partial a_c}{\partial a_p} = \left( \frac{da_n}{dn_n} \cdot \frac{\partial n_n}{\partial a_c} \right) \cdot \left( \frac{da_c}{dn_c} \cdot \frac{\partial n_c}{\partial a_p} \right) \quad (3.55)$$



**Şekil 3- 3: Birim ve Katmanların Gösterimi**

Eşitlik (3.53) 'den  $\frac{\partial n_c}{\partial a_p} = w_{cp}$  ve  $\frac{\partial n_n}{\partial a_c} = w_{nc}$

Eşitlik (3.54) 'den  $\frac{da_c}{dn_c} = a_c(1 - a_c)$  ve  $\frac{da_n}{dn_n} = 1$  olduğu bilinmektedir.

Böylece bunlar eşitlik (3.55) içersine yazılırsa ortaya şu denklem çıkar:

$$\frac{\partial a_n}{\partial n_p} = w_{nc} a_c (1 - a_c) w_{cp} \quad (3.56)$$

Gizli katmanda birden fazla hücre bulunması durumunda (C tane hücre için) girdi duyarlılığının genel formu şu şekilde yazılabilir:

$$\frac{\partial a_n}{\partial n_p} = \sum_{i=1}^C w_{cip} w_{nci} a_{ci} (1 - a_{ci}) \quad (3.57)$$

Artık duyarlılık analizinde kullanılacak formülasyonun tanımlanmasından sonra yöntemin uygulanabilirliğini kolaylaştıracak sırasıyla adımlar halinde verilen bir tanım verilmiştir.

1. Eldeki eğitim setiyle ağın eğitilmesi tamamlanır.
2. Nihai ağın **IW** (girdi ve gizli katman arasındaki ağırlık seti) ve **LW** (gizli ve çıktı katman arasındaki ağırlık seti) ağırlık değerleri elde edilir.
3. Araştırılacak olan her girdi bileşeni için  $[[\mathbf{IW}][\mathbf{P}]]$  (buradaki p eğitimde kullanılacak tüm girdilerin oluşturduğu bir matristir) matris çarpımından elde edilen her bir değer için lojistik fonksiyonundan geçirilmesiyle  $\mathbf{a}_c$  vektörü ve buradan da  $(1 - \mathbf{a}_c)$  vektörü elde edilir.
4. Duyarlılık formülasyonu kullanılarak her bir çıktının girdiye göre duyarlılığı bulunur ve eldeki veri setine dayanarak simülasyonlar sonucunda duyarlılık frekans aralıkları belirlenir.



### 3.3.3. Geri Yayılıma Dayalı Duyarlılık ve Regresyon Analizlerinin Karşılaştırılması

Bu bölüm içerisinde regresyon analiziyle sinir ağı modelleri karşılaştırılacaktır. Bunun nedeni çoğu araştırmacının regresyon modelinin kullanımına yönelik bilgiye sahip olmasıdır. Böylece sinir ağlarının regresyon analizine göre açıklaması verilerek sinir ağları konusunda daha derin bir anlayış kazanılmaya çalışılmıştır.

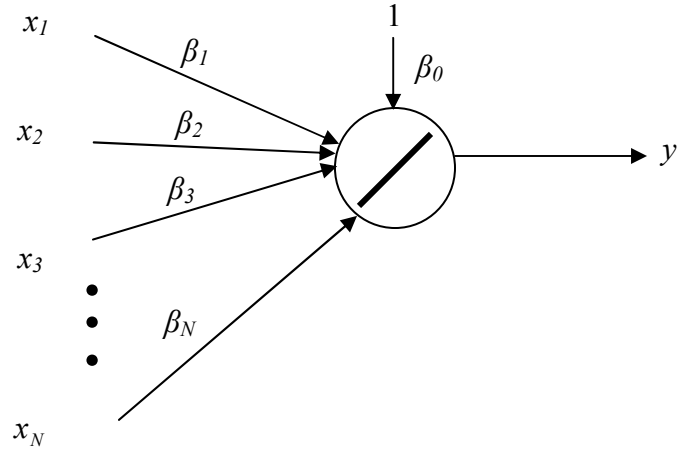
Bilinen çıktı değerleriyle eşleştirilmiş girdi desen seti verilmişken amaç girdi ve çıktılar arasındaki fonksiyonel ilişkiyi tahminlemek için denetimli öğrenmeyi kullanarak ağı eğitmektir. Bu durum bağımsız değişkenler (girdiler) ve bağımlı değişkenler (çıktılar) veri setinin elimizde bulunduğu ve bu ikisi arasındaki ilişkiyi bulmak istediğimiz regresyon problemine benzerlik göstermektedir. Regresyon modeli şu şekilde ifade edilir:

$$y_p = \beta_0 + \sum_{i=1}^N \beta_i x_{pi} + \varepsilon_p \quad (3.58)$$

Burada  $\varepsilon_p \sim N(0, \sigma^2)$  'dır. Regresyon probleminin amacı hata kareler toplamını minimize eden  $\beta_i$  katsayılarını bulmaktır.

$$E = \sum_{p=1}^n \left( y_p - \sum_{i=0}^N \beta_i x_{pi} \right)^2 \quad (3.59)$$

Katsayıları bulmak için bağımsız değişkenleri ve bunlara karşılık gelen bilinen bağımlı değişkenleri içeren bir veri setinin bulunması gereklidir. Bu sinir ağlarında denetimli öğrenmedeki eğitim setine benzemektedir.



**Şekil 3- 4: Bir Sinir Ağı Olarak Gösterilen Regresyon Modeli.**

Şekil 3-4 de gösterilen ağ doğrusal transfer fonksiyonuna sahip tek katmanlı ileri beslemeli sinir ağıdır. Bu ağ türü çoklu doğrusal regresyon modeliyle eşdeğerdir. Bağımsız değişkenler ağın girdilerine ve cevap değişkeni  $y$  ise ağın çıktısına karşılık gelmektedir. Transfer fonksiyonu doğrusal olarak alınmıştır ve  $\beta$  katsayıları sinir ağlarındaki ağırlıklara karşılık gelmektedir. Sinir ağlarında ağırlıkları bulmak amacıyla geri yayılım ve eşitlik (3.59) 'a benzer bir performans fonksiyonu kullanılmaktadır. İki yaklaşım arasındaki farklılık çoklu doğrusal regresyonun katsayıları bulmada kapalı bir form çözümünden hareket ederken sinir ağları ise iteratif bir süreci kullanmaktadır (Warner ve Misra, 1996:290).

Regresyon modellerinde fonksiyonel bir form veriye uydurulmaya çalışılır. Çoklu doğrusal regresyon durumunda çıktının bağımsız değişkenlerin doğrusal bir kombinasyonu ile ilişkili olduğu varsayılmaktadır. Eğer bu varsayılan model doğru değilse tahminlemede hatalar ortaya çıkacaktır. Alternatif yaklaşım herhangi bir fonksiyonel ilişkinin varsayılmaması ve verilere fonksiyonel formu tanımlamalarına izin verilmesidir. Sinir ağlarının gücünün temelini de bu oluşturmaktadır. Daha önce belirtildiği üzere Sigmoid transfer fonksiyonuna sahip iki katmanlı ileri beslemeli bir ağla herhangi bir sürekli fonksiyona istenilen doğruluk derecesinde yaklaşılabilmektedir. Bu yüzden bağımlı ve bağımsız değişkenler arasındaki fonksiyonel ilişki hakkında hiçbir bilgiye sahip değilken sinir ağları kullanılabilir.

oldukça yararlı bir araç olma niteliği taşımaktadır. Ancak fonksiyonel ilişki hakkında bir bilgi bulunuyorsa regresyon modelinin kullanılması daha doğrudur.

Fonksiyonel bir form varsayımının avantajı farklı hipotez testlerinin yapılabilmesine olanak sağlamasıdır. Örneğin regresyon, istatistiksel önem düzeyinde bireysel katsayıları test ederek fonksiyonel ilişkileri araştırma imkanı vermektedir. Ayrıca oluşturulan iki farklı regresyon modelinden hangisinin daha iyi olduğunun belirlenmesi için test yapılabilmektedir. Bir sinir ağı modeli Sigmoid fonksiyonlarının toplamı içersine gömülmüş olduğundan hiçbir zaman fonksiyonel ilişki hakkında açıklamada bulunamaz. Sinir ağlarıyla ilgili bir diğer problem gizli hücre sayısı, öğrenme oranı, başlangıç ağırlık değeri ve eğitimin ne zaman durdurulacağına kararını gibi parametre değerlerinin seçimidir. Bu seçimde bir kesinlik bulunmamakta, sürekli olarak farklı parametre değerlerinin kullanıldığı ve değerlendirildiği deneye dayalı bir süreç şeklinde olmaktadır. Dolayısıyla bu sürecin tamamlanması regresyonla kıyaslandığında oldukça fazla zamana mal olmaktadır.

Genel olarak YSA' nın regresyon analizine üstünlüklerini şu şekilde sıralamak mümkündür:

- ❖ YSA, EKK (En Küçük Kareler yöntemi) 'dan farklı olarak veriler üzerinde herhangi bir kısıt öne sürmemektedir.
- ❖ YSA tek çıktılı regresyon analizinden farklı olarak birden çok girdi ve çıktı ilişkilendirmesini desteklemektedir.
- ❖ Regresyon analizine kıyasla veriler üzerindeki gereksinimleri oldukça rahatlatır. Hata toleransı ve gözlenen verilerin az sayıda olması durumunda doğrusal olmayan ilişkilendirme yeteneğine sahiptir.

Duyarlılık analizi kapsamında YSA' nın regresyon analiziyle karşılaştırılması istenirse eşitlik (3.57) 'nin incelenmesiyle  $\frac{\partial a_n}{\partial n_p}$  değerinin şu gibi faktörlere bağımlı olduğu görülür (Lu, AbouRizk ve Hermann, 2001:303):

1. Ağın iç yapısına başka bir deyişle gizli hücre ve gizli katman sayısına.
2. Ağın ağırlık değerlerine. Bundan kastedilen şey girdi ve çıktı katmanı arasındaki birimlerin birbiriyle olan bağlantısının yoğunluğudur. Aslında bu ağın eğitiminin sonucunda ortaya çıkar dolayısıyla eğitimde kullanılan veri setine bağlıdır.
3. Ağın girdi birimlerine o anda yüklenen girdi değerlerine. Eşitlik (2.2), ve (2.4) 'den  $a_c$  değerlerinin ağın ağırlık değerlerinin ve girdi katmanındaki o anki girdi değerlerinin fonksiyonu olduğu açıkça görülmektedir.

Dikkat edilmesi gereken şey ağın eğitimi veri setine bağlı olarak tamamlandıktan sonra ağın yapısı, ağırlıklar ve eğitimde kullanılan veri seti faktörlerinin sabitlenmiş olacaktır. Böylece çıktı değişkenine göre girdinin duyarlılığı tamamıyla üçüncü faktör olan o anki girdi değerleriyle belirlenecektir. Eğer o anki girdi değerlerine, girdi uzayındaki bir girdi noktasının koordinat değerleri gözyle bakılırsa bu girdi uzayının boyutu girdi parametrelerinin sayısına eşit olacaktır. Eğitimi tamamlanmış ağ için şu eşitlik tanımlanabilecektir:

$$\frac{\partial a_n}{\partial n_p} = F(\text{GirdiNoktası}) \quad (3.60)$$

Burada  $F$  bir fonksiyonu göstermektedir. Dolayısıyla doğrusal regresyon analiziyle karşılaştırıldığında tek bir duyarlılık değeri elde edilmeyecektir. Aksine her bir girdi noktası için duyarlılık değeri elde edilmesi imkanı ortaya çıkacaktır. Böylece incelenen veriyi dikkate alan daha esnek sonuçların elde edilmesi sağlanacaktır.

Bu çalışmada öncelikle incelenen problemin altında yatan fonksiyonun bilindiği varsayılacaktır. Sinir ağlarında duyarlılık analiziyle bu fonksiyonun bağımsız değişkenlerinin bağımlı değişkenler üzerinde ne yönde ve ne kadarlık bir etkiye sahip oldukları araştırılacaktır. Ayrıca çoklu regresyon analiziyle elde edilen sonuçların karşılaştırılması yapılacaktır. Karşılaştırma için 4 farklı senaryo üretilmiş

ve bunlara sırasıyla Durum1, Durum2, Durum3 ve Durum4 isimleri verilmiştir. Yapılan karşılaştırma klasik regresyon analiziyle sinir ağının bir tahminleme problemindeki hataları minimize etme yeteneklerini incelemeyecektir. Bunun yerine önerilen metodun doğruluğunun ve tutarlılığının gösterilmesi amacıyla çoğu araştırmacı tarafından bilinen, klasik bir araç olan regresyon analizinden yararlanılacaktır.

Regresyon analizi ve sinir ağlarıyla yaklaşılmaya çalışılan fonksiyonun doğrusal olduğu ve parametre değerlerinin bilindiği varsayılmıştır. Böylece bu anakütle parametrelerinin duyarlılık ve regresyon analiziyle tahminlenmesine gidilecektir. Ele alınan fonksiyonun doğrusal olmasından ötürü ilgilenilen parametre değerleri her bir bağımsız değişkenin önündeki katsayılarıdır. Bilindiği üzere doğrusal modelde bağımsız değişkenin önündeki katsayı o bağımsız değişkenin bağımlı değişken üzerindeki etkisini göstermektedir. Dolayısıyla duyarlılık analiziyle bu katsayı değerlerine ne ölçüde yaklaşılabildiği gösterilmeye çalışılacaktır.

Duyarlılık analizini yerine getirmek için Matlab ortamında önceden programın kendisi tarafından bilinen YSA fonksiyonlarından ve ek olarak yine Matlab 'de yazılan kodlardan faydalanılmıştır. Yazılan kodlar Ek7 ve Ek8 'de verilmiştir. İlk olarak Matlab ortamında 0 ve 1 arasında Uniform dağılıma sahip 4 girdi değişkeninden 350 'şer tane türetilmiştir. Daha sonra eşitlik (3.61) 'deki anakütle parametrelerine sahip olduğu varsayılan doğrusal modelden bağımlı değişken değerleri elde edilmiştir.

$$Y = 3X_1 + 8X_2 - 5X_3 + 12X_4 + \varepsilon \quad (3.61)$$

Burada  $\varepsilon$ , 0 ve 1 arasında normal dağılıma sahip gürültü değeridir.

Bağımlı ve bağımsız değişkenler arasındaki fonksiyonel ilişki bilinmediği zaman sinir ağlarının kullanımı daha değerli olabilmektedir. Teori ve önceki bilgiler altta yatan ilişki hakkında bilgi veriyorsa regresyonun daha iyi sonuçlar vereceği yapılan birçok çalışmadan bilinmektedir. Durum1 içersinde doğrusal regresyonun tüm varsayımları sağlanacak şekilde veriler düzenlenip regresyon ve sinir ağlarında

duyarlılık analizi yapılmıştır. Durum2 'de anakütle hata terimi normal dağılmayacak şekilde oluşturularak varsayım ihlali halinde iki yöntemin sonuçları incelenmiştir. Durum3 içersinde ana kütle hata teriminin ardışık değerleri arasında ilişki oluşturulmuş ve literatürde bilindik bir yöntemle otokorelasyon ortadan kaldırılmaya çalışıldıktan sonra regresyon ve sinir ağı sonuçları elde edilmiştir. Son olarak Durum4 'de  $X_1$  ve  $X_2$  değişkenleri arasında doğrusal bağlantı oluşturularak varsayım ihlali yapılmış ve sonuçlar verilmiştir. Dikkat edilirse oluşturulan doğrusal modelin sabit terim içermediği varsayılmıştır. Buna bağlı olarak kurulan sabit terimsiz regresyon modelinde  $r^2$  belirlilik katsayısı uygun bir ölçü olmaktan çıkmaktadır. Başka bir deyişle  $r^2$  'nin yorumu anlamlı olmayacaktır.

Öncelikle tüm durumlarda regresyon analizi uygulanmıştır. Daha sonra aynı değişkenler YSA modeline sokulmuştur. Tüm durumlarda kullanılacak YSA modelinin eğitim sürecinde belirlenmesi gereken parametre değerlerine bilgisayar ortamında yürütülen deneme ve yanımlarla karar vermeye çalışılmıştır ve hepsinde ortak değerler kullanılmıştır. . YSA modelinde gizli katmandaki hücre sayısı deneme yanımlarla 9 olarak belirlenmiştir. Tek gizli katman ve gizli katmanın transfer fonksiyonu ise Sigmoid olarak alınmıştır. Çıktı katmanında doğrusal transfer fonksiyonu kullanılmıştır. Eğitim algoritması olarak Levenberg-Marquardt algoritmasının kullanılmasının en performansı yüksek sonuçlar verdiği görülmüştür. 250 iterasyonun sonuç vermede yeterli olduğuna ve her bir nokta için de 300 duyarlılık değerinin hesaplanmasına karar verilmiştir. Ayrıca asıl amacın duyarlılık analizi olmasına rağmen bağımlı değişkene yönelik her iki yöntemin de yaptıkları tahmin değerleri merak edilmiş ve her durum için ortalama mutlak sapma değerleri verilmiştir.

### **3.3.3.1. Durum1: Varsayımların Sağlanması**

Doğrusal regresyon modeliyle katsayıların En Küçük Kareler Yöntemi (EKKY) ile tahmini elde edilip bu tahmin değerlerinin gerçek anakütle parametrelerine yakınlığı test edilip ve  $Y$  tahminlerin gerçek  $Y$  değerlerine yakınlığı araştırılır. Ancak bu tahmin test süreçlerinde kullanılan EKKY formülleri bazı varsayımların sağlandığının kabul edilmesiyle kurulmuştur. Dolayısıyla model

sonuçları elde edilmeden önce varsayımların sağlanıp sağlanmadığına bakmak gerekir. Gerçek hayattaki uygulamalara bakıldığında tüm varsayımları aynı anda sağlamanın özel bir durum olduğu görülmektedir. Bu çalışmada EKKY ‘nin bazı varsayımlarının sağlanmadığı veri gruplarında katsayıları elde etmek amacıyla önce klasik yöntemle sonuç alınmaya çalışılacak ve daha sonra ise aynı veri grubu üzerinde herhangi bir varsayım gerektirmeyen YSA yöntemiyle katsayı değerleri tahminlenmeye çalışılacaktır. Bağımlı değişkenin tahmin değerlerinin ortalama mutlak sapması regresyon analizinde 0,737 iken sinir ağlarında 0,669 olarak bulunmuştur.

**Tablo 3- 1: Durum1 için Regresyon Sonuçları**

Dependent Variable: Y  
Method: Least Squares  
Sample: 1 350  
Included observations: 350

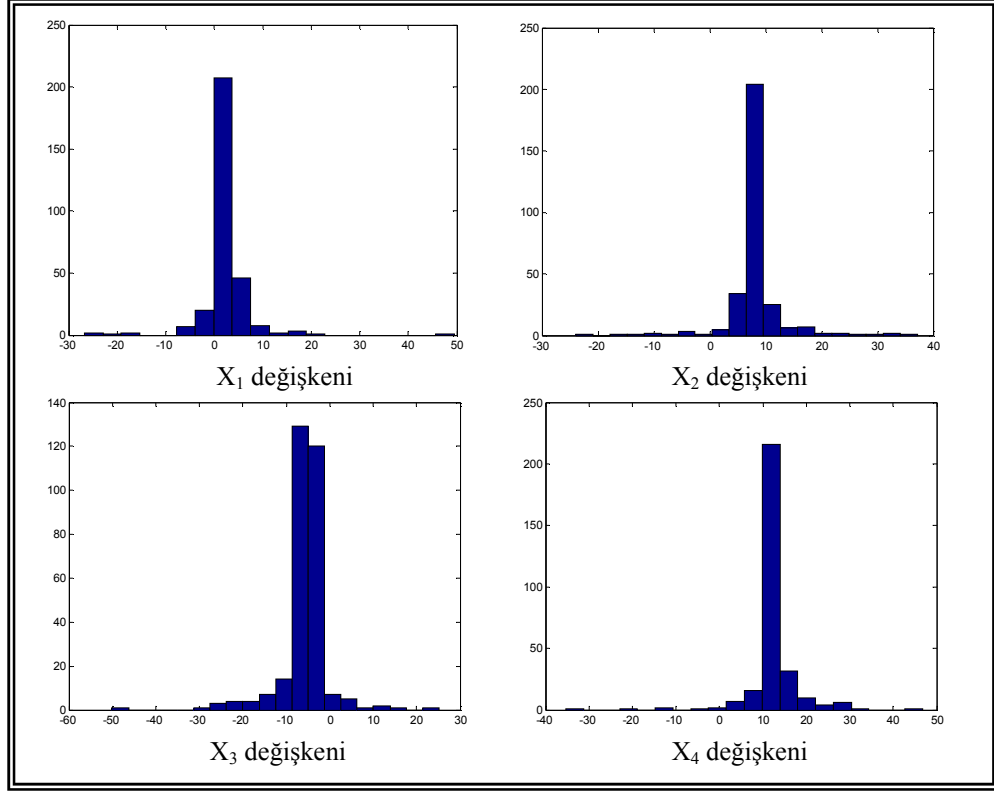
Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	3.140113	0.155453	20.19972	0.0000
X2	7.936600	0.153460	51.71774	0.0000
X3	-4.995555	0.149858	-33.33527	0.0000
X4	11.85116	0.155170	76.37536	0.0000
R-squared	0.959430	Mean dependent var	8.990796	
Adjusted R-squared	0.959078	S.D. dependent var	4.592272	
S.E. of regression	0.928976	Akaike info criterion	2.701895	
Sum squared resid	298.5969	Schwarz criterion	2.745986	
Log likelihood	-468.8317	F-statistic	2727.493	
Durbin-Watson stat	1.947337	Prob(F-statistic)	0.000000	

**Tablo 3- 2: Durum2 için YSA Duyarlılık Analizi Sonuçları**

DEĞİŞKENLER	X1	X2	X3	X4
<b>Ortalama</b>	2.587285	8.104469	-5.7513	12.08995
<b>Medyan</b>	2.911233	7.692137	-5.30363	11.84647
<b>Trimmean(%15)</b>	2.758202	7.937514	-5.61241	12.04806
<b>Std Sapma</b>	8.095199	8.279625	8.951567	9.006931

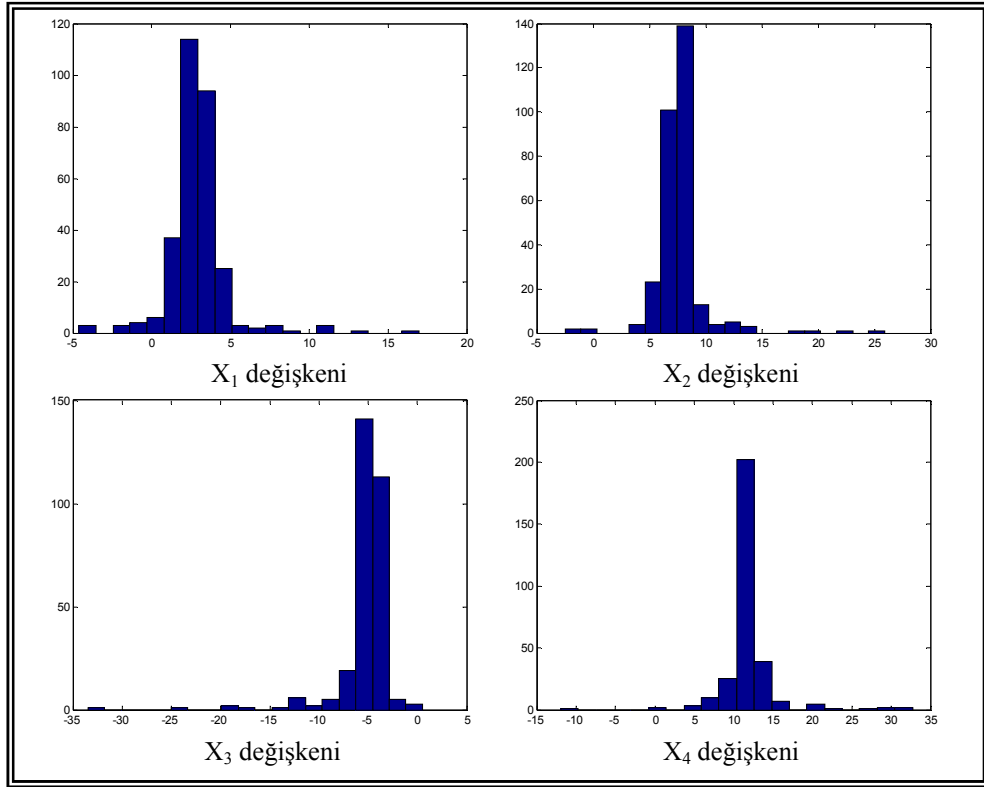
Varsayımların sağlanması durumunda regresyon analizinden elde edilen katsayı tahminlerinin sinir ağlarında duyarlılık analiziyle elde edilen duyarlılık dağılımının her bir tanımlayıcı istatistik değerine göre daha iyi sonuçlar vermiştir. Zaten bu beklenen bir sonuçtur. Çünkü çoklu doğrusal regresyonun başarıyla

uygulanabilmesi için tüm koşullar mevcuttur. Ancak amaç önerilen metodun verdiği sonuçları incelemek olduğundan duyarlılık analizinden elde edilen sonuçlar en iyi olmasa da istenilen katsayı değerlerine yakınlık göstermiştir.

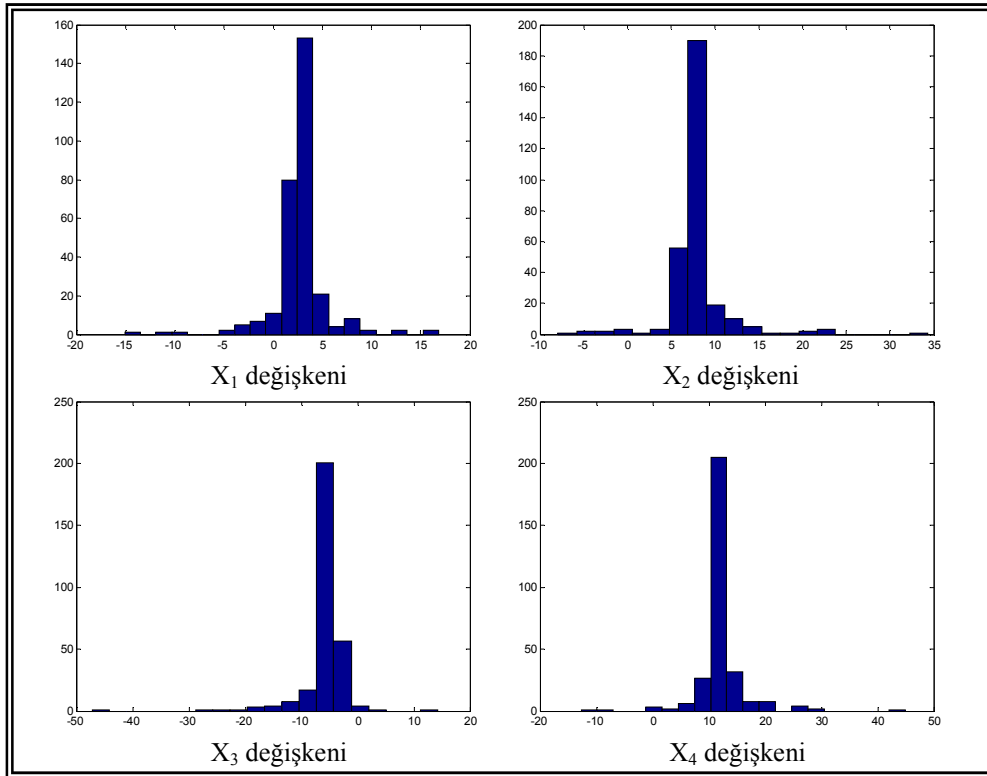


**Şekil 3- 5: Duyarlılık Değerinin Belirlenmesinde Kullanılan Ortalamaların Dağılımları**

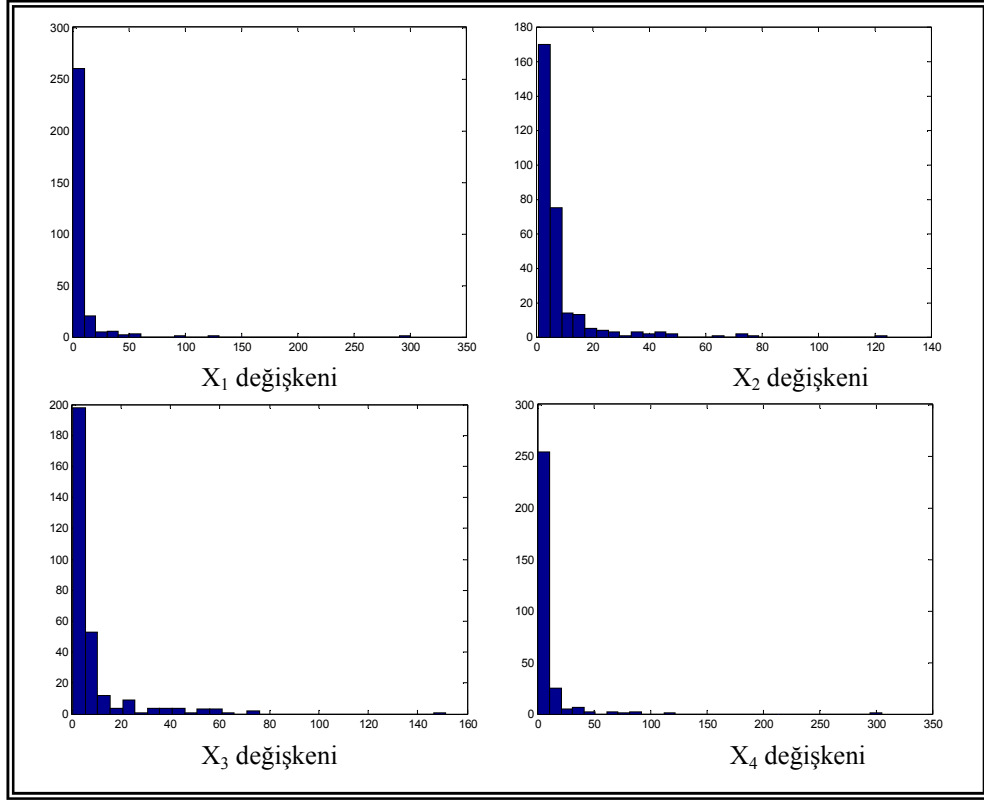




**Şekil 3- 6: Duyarlılık Değerinin Belirlenmesinde Kullanılan Medyanların Dağılımları**



**Şekil 3- 7: Duyarlılık Değerinin Belirlenmesinde Kullanılan Trimmean (%15) Dağılımları**



**Şekil 3- 8: Duyarlılık Değerlerinin Standart Sapma Dağılımları**

### 3.3.3.2. Durum2: Normallik Varsayımının Sağlanmaması

İstatistikte tüm testler ancak normal dağılmış bir anakütleden çekilen örneklere uygulanabilmektedir. Eğer hata terimi  $u$  normal dağılıyorsa EKK  $\hat{b}_1$  ve  $\hat{b}_2$  'nin tahminçileri de normal dağılır. Çünkü normal dağılmış değişkenleri olan bir doğrusal fonksiyonun kendisinde normal dağılımlıdır. Böylece  $b$  tahminleri konusunda gerektiğinde bir test uygulamak (t, F testi gibi) mümkün olabilir. Hata teriminin normal dağılıp dağılmadığını öğrenmek için çeşitli testlerden yararlanılmaktadır. Bu analiz kapsamında büyük örnekler için normallik varsayımını test eden Jarque-Bera (JB) testinden faydalanılmıştır. JB testinde sıfır hipotezi hata teriminin normal dağılıma sahip olmasıdır. JB test istatistiği  $\chi^2$  tablo değerinden küçükse hata terimlerinin normal dağıldığı hipotezi kabul edilir. Şekil 3-9 'da JB değerinin oldukça büyük çıktığı görülmüştür. Dolayısıyla buradan hata teriminin normal dağılmadığı anlaşılmaktadır. Bağımlı değişkenin tahmin değerlerinin

ortalama mutlak sapması regresyon analizinde 0,628 iken sinir ağlarında 0,589 olarak bulunmuştur.

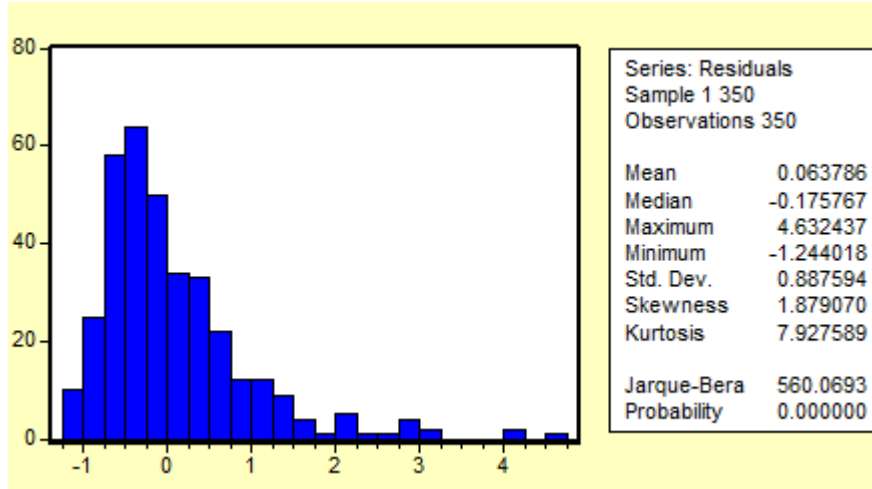
**Tablo 3- 3: Durum2 için Regresyon Sonuçları**

Dependent Variable: Y  
Method: Least Squares  
Sample: 1 350  
Included observations: 350

Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	3.592629	0.149557	24.02184	0.0000
X2	8.124354	0.147639	55.02852	0.0000
X3	-4.659657	0.144174	-32.31976	0.0000
X4	12.44832	0.149284	83.38672	0.0000

R-squared	0.962147	Mean dependent var	9.861152
Adjusted R-squared	0.961819	S.D. dependent var	4.573916
S.E. of regression	0.893739	Akaike info criterion	2.624556
Sum squared resid	276.3741	Schwarz criterion	2.668647
Log likelihood	-455.2974	F-statistic	2931.574
Durbin-Watson stat	2.047786	Prob(F-statistic)	0.000000



**Şekil 3- 9: Regresyon Hatalarının Normal Dağılıp Dağılmadığının Testi**

Yukarıdaki regresyon analizi sonuçları verildikten Tablo 3-4 'de duyarlılık analizi sonucunda elde edilen dağılımın çeşitli tanımlayıcı istatistik değerleri verilmiştir. Ayrıca Ek4 'de duyarlılık dağılımları bulunmaktadır. Görüldüğü gibi normal dağılımın sağlanmadığı durumda YSA ile elde edilen sonuç bu durumda az da olsa regresyon analizine göre daha iyi sonuçlar vermiştir.

**Tablo 3- 4: Durum2 için YSA Duyarlılık Analizi Sonuçları**

DEĞİŞKENLER	X1	X2	X3	X4
Ortalama	2.986602	7.693805	-4.79897	12.02662
Medyan	3.098458	7.454971	-4.59712	11.61217
Trimmean(%15)	3.026915	7.67735	-4.67694	11.81906
Std Sapma	7.100617	7.91141	7.983851	6.76955

### 3.3.3.3. Durum3: Otokorelasyonun Bulunması Hali

Doğrusal regresyon modelinin varsayımlarından biri anakütle hata teriminin ardışık değerleri arasında ilişki (otokorelasyon) olmamasıdır. Otokorelasyon modele dahil edilmeyen bağımsız değişkenler, modelin fonksiyonel şeklinin yanlış alınması, bağımlı değişkenin ölçme hatası taşınması gibi sebeplere dayanır (Akkaya ve Pazarlıoğlu, 2000:445). Aşağıdaki eşitlikte anakütle hata terimi  $u_t$  ve onun bir önceki dönemde aldığı değer  $u_{t-1}$  arasındaki birinci dereceden otokorelasyon durumu gösterilmektedir. Buradaki  $p$  birinci dereceden otokorelasyon katsayıdır.

$$u_t = pu_{t-1} + v_t \quad , \quad -1 < p < 1 \quad (3.62)$$

Otokorelasyon tespiti için grafik yöntem, sıra testi ki-kare testi, Durbin-Watson testi, Wallis testi gibi yöntemler vardır. Ancak bu otokorelasyon tespit yöntemlerinin hepsi birinci dereceden otokorelasyon olması varsayımı altında çalışırlar. Otokorelasyon katsayısı  $p$  'nin birden fazla olması durumunda bu testler geçerliliğini yitirir. Aşağıdaki eşitlikte verildiği gibi otokorelasyonun bulunduğu durumlarda Breusch-Godfrey otokorelasyon testi kullanılır.

$$u_t = p_1u_{t-1} + p_2u_{t-2} + v_t$$

$$u_t = p_1u_{t-1} + p_2u_{t-2} + p_3u_{t-3} + v_t$$

..... (3.63)

$$u_t = p_1 u_{t-1} + p_2 u_{t-2} + \dots + p_s u_{t-s} + v_t$$

**Tablo 3- 5: Otokorelasyon Mevcutken Regresyon Sonuçları**

Dependent Variable: Y  
Method: Least Squares  
Sample: 1 350  
Included observations: 350

Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	5.036890	0.280803	17.93747	0.0000
X2	10.48313	0.277202	37.81767	0.0000
X3	-3.032001	0.270696	-11.20078	0.0000
X4	14.07435	0.280291	50.21334	0.0000
R-squared	0.869329	Mean dependent var	13.71972	
Adjusted R-squared	0.868196	S.D. dependent var	4.622122	
S.E. of regression	1.678054	Akaike info criterion	3.884510	
Sum squared resid	974.2895	Schwarz criterion	3.928600	
Log likelihood	-675.7892	F-statistic	767.2891	
Durbin-Watson stat	1.216953	Prob(F-statistic)	0.000000	

Breusch-Godfrey otokorelasyon testini uygulamak için öncelikle yukarıda kurulan regresyondan hata terimleri elde edilir. Daha sonra otokorelasyonun derecesine göre şu yardımcı regresyon denklemi kurulur:

$$e_t = b_1 + b_2 X_t + p_1 e_{t-1} + p_2 e_{t-2} + \dots + p_s e_{t-s} + v_t \quad (3.64)$$

Bu regresyondan elde edilen  $R^2$  değeri hesaplanarak  $nR^2$  değeri  $\alpha$  önem düzeyinde ve  $s$  serbestlik derecesindeki  $\chi_{tab}^2$  değerinden büyükse otokorelasyon bulunmaktadır sıfır hipotezi kabul edilir.

Eşitlik (3.64) 'e göre 2.dereceden otokorelasyon baz alınarak yardımcı regresyon denklemi kurulmuş olup  $R^2=0.841$  olarak bulunmuştur. Buradan  $\chi_{hes}^2 = (346)(0.841) = 290.986 > \chi_{tab}^2 = 5.991$  olduğundan otokorelasyonun varlığı tespit edilmiştir.

Otokorelasyon bulunması halinde EKK tahmincileri minimum varyansa sahip değildirler fakat doğrusal sapmasızlık ve tutarlılık özelliklerini sürdürürler. Buna

bağlı olarak  $t$  ve  $F$  testi sonuçları güvenilmez olur.  $t$  hesap ve  $F$  hesap değerleri gerçekten daha küçük olacağından parametreler anlamlı olduğu halde anlamsız olması sonucu çıkabilecektir. Ayrıca  $R^2$  olduğundan daha büyük bir değere sahip olacak ve katsayılar için güven aralıkları genişleyecektir. Bu nedenlerden ötürü otokorelasyonlu bir modelde otokorelasyonu giderme yolları aranır. Uygulamada  $p$  otokorelasyon katsayısının bilinip bilinmemesine bağlı olarak yöntemler geliştirilmiştir. Genelde anakütle otokorelasyon parametresi  $p$  önceden bilinmez. Otokorelasyonu ortadan kaldırmak için genelleştirilmiş fark denklemleri kurularak bunun üzerinde EKK yönteminin uygulanması yapılır (Gujarati, 2005:426). Bu çalışmada  $p$  'nin bilinmediği varsayılarak iki aşamalı Cochrane-Orcutt yöntemine göre otokorelasyonun ortadan kaldırılmasına çalışılmıştır. İki aşamalı Cochrane-Orcutt yöntemine göre genelleştirilmiş fark denkleminin sonuçları aşağıda verilmiştir.

**Tablo 3- 6: GEKKY Göre Regresyon Sonuçları**

Dependent Variable: Y-YP  
Method: Least Squares  
Sample(adjusted): 2 350  
Included observations: 349 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1-X1P	4.846489	0.236565	20.48690	0.0000
X2-X2P	9.939580	0.230797	43.06636	0.0000
X3-X3P	-3.230315	0.225089	-14.35128	0.0000
X4-X4P	13.79905	0.241641	57.10553	0.0000
R-squared	0.908805	Mean dependent var	8.462253	
Adjusted R-squared	0.908012	S.D. dependent var	4.877595	
S.E. of regression	1.479352	Akaike info criterion	3.632481	
Sum squared resid	755.0268	Schwarz criterion	3.676665	
Log likelihood	-629.8679	F-statistic	1146.032	
Durbin-Watson stat	1.758433	Prob(F-statistic)	0.000000	

Tablo 3-6 'da GEKKY (Genelleştirilmiş En Küçük Kareler Yöntemi) 'e göre regresyon sonuçları görülmektedir. Tablo 3-5 'de oluşturulan ilk regresyon modeliyle karşılaştırıldığında katsayı parametrelerine daha yaklaştığı görülmektedir. Ancak duyarlılık analiziyle elde edilen katsayı tahminleri çok daha iyi sonuçlar vermiştir. Ayrıca bağımlı değişkenin tahmin değerlerinin ortalama mutlak sapması regresyon analizinde 1,304 iken sınır ağlarında 0,598 olarak bulunmuştur.

**Tablo 3- 7: Durum3 için YSA Duyarluluk Analizi Sonuqları**

DEĐİŐKENLER	X1	X2	X3	X4
Ortalama	3.456514	8.14574	-4.38653	11.01428
Medyan	2.358912	7.995207	-4.71467	11.17933
Trimmean(%15)	2.361869	7.968164	-4.50982	11.17974
Std Sapma	16.01145	14.09183	13.09271	12.36186

#### **3.3.3.4. Durum4: Bađımsız DeđiŐkenler Arasında Çoklu Dođrusal Bađlantı Olması Hali**

Çok deđiŐkenli bir regresyon modelinde bađımsız deđiŐkenler arasındaki korelasyon katsayısının 1 'e eŐit veya yakın olmasına yani deđiŐkenler arasında bir iliŐkinin mevcut olması durumuna çoklu dođrusal bađlantı denir. Çoklu dođrusal bađlantı halinde regresyon katsayılarının standart hata deđerleri büyüyecek ve buna bađlı olarak katsayılar tam ve dođru tahmin edilemeyecektir. Çoklu dođrusal bađlantının varlıđı durumunda  $R^2$  olduđundan büyük çıkacak  $F$  testi ile katsayılar topluca anlamlı iken  $t$  testi sonucunda katsayılar birer birer anlamsız veya az anlamlı olacaktır. Ayrıca bađımsız deđiŐkenler arasında ikiŐerli kuvvetli bir iliŐki ortaya çıkabilecektir.

Regresyon kurulmadan önce veriler oluŐturulurken  $x_1$  ve  $x_2$  deđiŐkenleri arasında bir dođrusal bađlantı kurulmuŐtur. Daha sonra ise bu verilere regresyon uygulanmıŐtır ve sonucu aŐađıda verilmiŐtir.

**Tablo 3- 8: Durum4 için Regresyon Sonuçları**

Dependent Variable: Y  
Method: Least Squares  
Sample: 1 350  
Included observations: 350

Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	1.935862	0.259341	7.464544	0.0000
X2	10.53184	0.476198	22.11651	0.0000
X3	-4.811909	0.051675	-93.11785	0.0000
X4	12.28314	0.050874	241.4402	0.0000
R-squared	0.994240	Mean dependent var	7.923981	
Adjusted R-squared	0.994190	S.D. dependent var	4.140879	
S.E. of regression	0.315636	Akaike info criterion	0.542907	
Sum squared resid	34.47058	Schwarz criterion	0.586998	
Log likelihood	-91.00878	F-statistic	19907.09	
Durbin-Watson stat	2.037810	Prob(F-statistic)	0.000000	

Çoklu doğrusal bağlantının belirtilerinden biri yüksek  $R^2$  ve  $F$  testi sonucunda katsayılar topluca anlamlı iken  $t$  testi ile katsayıların anlamsız veya az anlamlı çıkması durumudur. Model çıktısı incelendiğinde sanki çoklu doğrusal bağlantı yokmuş gibi görünmektedir. Ön bilgi olarak  $x_1$  ve  $x_2$  arasında doğrusal bir bağlantı olduğunun bilinmesine rağmen çoklu doğrusal bağlantıyı tespit etmede kullanılan yardımcı regresyon kriteriyle bu ilişki aşağıda gösterildiği gibi ortaya çıkarılmıştır.

Yardımcı regresyon kriterinde hangi  $x$  değişkeninin diğer  $x$  değişkenleriyle çoklu doğrusal bağlantılı olduğunu bulabilmek amacıyla her  $x$  değişkeniyle geri kalan  $x$  değişkenleri arasında doğrusal regresyon kurulur. Buradan her bir belirlilik katsayısı hesaplanarak aşağıdaki eşitlikte sırasıyla yerine konur. Eğer hesaplanan  $F_i$  değeri  $F_{tab}$  değerinden büyükse  $x_i$  ile diğer  $x$  'ler arasında çoklu doğrusal bağlantının olduğuna karar verilir.

$$F_i = \frac{R_{xi.x1x2...xk}^2 / (k - 2)}{(1 - R_{xi.x1x2...xk}^2) / (n - k + 1)} \quad (3.65)$$

$k$ , modeldeki değişken sayısı;  $n$ , örnek hacmi;  $R_{xi.x1x2...xk}^2 = x_i$  ile kalan  $x$  değişkenleri arasındaki belirlilik katsayısıdır.  $F_{tab}=2.65$  iken aşağıdaki tabloda sırasıyla  $F_i$  değerleri verilmiştir.



**Tablo 3- 9: Yardımcı Regresyonların  $F_i$  Değerleri**

	$X_1$	$X_2$	$X_3$	$X_4$
$F_i$	2632.529	2633.295	0.723	3.372

Tablodan görüldüğü gibi özellikle  $x_1$  ve  $x_2$  değişkenleri için bu sınama olumlu sonuç vermekte ve çoklu doğrusal bağlantının var olduğu anlaşılmaktadır.

Çoklu doğrusal bağlantı ile EKK yönteminin varsayımlarından biri ihlal edilmiş olunur. Çoklu doğrusal bağlantıyı tek bir ve hatasız şekilde ortadan kaldırma yolu yoktur. Yapılabilecek en kolay şey aralarında kuvvetli ilişki olan bağımsız değişkenlerden birini modelden çıkarmaktır. Ancak bu yapılıncaya da modelde olması gereken bir değişken çıkarılmış olunabileceğinden spesifikasyon hatası yapma riski ortaya çıkar. Bu durumda ise katsayılar gerçek değerlerinin altında veya üstünde tahmin edilir. Eğer model kurulmasındaki amaç tahmin ise çoklu doğrusal bağlantının bulunması çok önemli değilken parametrelerin güvenilir tahminleri elde edilmek isteniyorsa çoklu doğrusal bağlantı önemli bir problem olma niteliği taşımaktadır.

**Tablo 3- 10: Durum4 için YSA Duyarlılık Analizi Sonuçları**

DEĞİŞKENLER	X1	X2	X3	X4
<b>Ortalama</b>	3.438006	9.944231	-6.13554	13.62294
<b>Medyan</b>	3.070281	8.51093	-5.2991	12.34843
<b>Trimmean(%15)</b>	3.190997	9.593052	-5.46196	12.64573
<b>Std Sapma</b>	9.635371	15.02185	7.307223	9.727016

Çoklu doğrusal bağlantı mevcutken regresyon ve YSA duyarlılık analizi sonuçlarını karşılaştırdığımızda YSA 'nın bağımsız değişkenler arasında bağlantı olup olmamasına bakmamış ve tahminlerini sağlıklı bir şekilde vermiştir. Görüldüğü YSA herhangi bir varsayıma bağlı kalmaksızın sonuçlara ulaşmaktadır. Ayrıca bağımlı değişkenin tahmin değerlerinin ortalama mutlak sapması regresyon analizinde 0,263 iken sinir ağlarında 0,221 olarak bulunmuştur.

## SONUÇ VE DEĞERLENDİRME

Yapılan çalışmada, ele alınan problemin altında yatan fonksiyonel form hakkında herhangi bir varsayımda bulunmaksızın o fonksiyona sinir ağlarıyla yaklaşılabildiği gösterilmeye çalışılmıştır. Bu amaca yönelik olarak gerçekte incelenen problemin doğrusal fonksiyon kaynaklı olduğu varsayılmış ve bu doğrusal fonksiyondaki katsayı parametrelerinin duyarlılık analizi adı altında sinir ağlarıyla tahminlenmesine gidilmiştir. İncelenen fonksiyonun tahmin edilmeye çalışılan parametleri önceden bilinmektedir. Dolayısıyla bu parametre değerlerine ne kadar yaklaşılabildiğinin ölçüsü aynı zamanda yönteminde başarısının ölçüsünü göstermektedir. Bağımlı değişkeninin değerlerini belirleyen ve en baştan sanki bilinmediği kabul edilen fonksiyonun, doğrusal olarak seçilmiş olmasının iki nedeni vardır. Birincisi doğrusal bir fonksiyonun parametre değerlerinin neredeyse tüm araştırmacılar tarafından bilinen klasik çoklu regresyon analiziyle tahmin edilebilme olanağına sahip olmasıdır. Böylece sinir ağlarında duyarlılık analiziyle girdi (bağımsız) ve çıktı (bağımlı) değişkenlerini ilişkilendirmede kullanılan yeni yöntemin kullanılışlığının ve yararının daha iyi anlaşılabilinmesinde bilindik yöntemden faydalanılmış olunacaktır. İkinci neden, önceden bilinmeyen fonksiyonel ilişkinin aslında basit ve klasik yöntemlerle çözülebileceği bir durumda sanki karmaşık problemmiş gibi düşünülmesi ve sinir ağının nasıl sonuçlar üreteceğinin gözlenmesine imkan vermiş olmasıdır. Gerçek hayattaki problemlerde çoğu zaman elimizde sadece ilgili probleme ait veriler bulunurken fonksiyonel ilişki bilinmemektedir. Literatürde yapılan çalışmalardan da görüleceği gibi böyle bir durumda sinir ağlarıyla klasik tekniklerin performansı karşılaştırılmaktadır. Ancak hangi yöntemin üstün olduğu konusunda karar vericiye kesin bir bilgi sağlanmamaktadır. Doğru olan eldeki problemi en iyi yansıtan modelin seçilmesidir. Bu çalışmanın öncelikli ilgi odağı klasik bir yöntemle sinir ağının performansının karşılaştırılmasından öte, YSA modeliyle girdi ve çıktı değişkenleri arasındaki ilişkinin yönünün ve gücünün belirlenmesidir. Sinir ağlarının doğrusal olmayan, gürültülü, eksik verilerin bulunması halinde tahminleme ve sınıflandırma alanlarında oldukça iyi sonuçlar verdiği bilinmektedir. Ancak iyi bir tahminleme aracı olmasına rağmen nedensellik hakkında bir bilgi vermemektedir. Bir başka deyişle bağımlı ve

bağımsız değişkenleri ilişkilendirememektedir. Çalışmanın amacı da bu noktada ortaya çıkmış ve çıktı değişkeninin girdi değişkeninden etkilenme gücünün yani duyarlılığının tespit edilmesine çalışılmıştır.

Çalışma kapsamında ilk olarak YSA genel olarak tanıtılmış ve neden, ne zaman, nerelerde ve nasıl kullanıldığı hakkında bilgi verilmiştir. İkinci bölümde yapılan çalışmada kullanılan öğrenme şekli olan geri yayılım algoritmasının ayrıntılı bir şekilde matematiksel olarak türetilmesine yer verilmiştir. Üçüncü bölümde ise ikinci bölüm içerisinde yer verilen temel geri yayılım algoritmasının uygulamada yaşadığı problemler hakkında bilgi verilmiş olup bunun üstesinden gelmek için ne tür yöntemlerin ortaya konduğu anlatılmıştır. Ayrıca üçüncü bölüm içerisinde çalışmanın odak noktası olan girdinin çıktı üzerindeki etkisini ölçmeye yarayan duyarlılık analizi hakkında açıklamalara gidilmiştir. Lu vd. (2001) tarafından yapılan sinir ağlarıyla duyarlılık analizi çalışması temel alınmıştır. Lu vd. yürüttüğü çalışma ileri beslemeli bir geri yayılım ağının oluşturulduğunu, temel geri yayılım veya eğim iniş algoritmasının kullanıldığını ve gizli ve çıktı katmanında transfer fonksiyonu olarak Sigmoid fonksiyonunun alındığını varsayarak oluşturulmuştur. Halbuki literatürde yapılan birçok çalışmada gizli katmanda Sigmoid ve çıktı katmanında doğrusal transfer fonksiyonuna sahip bir ağla herhangi bir sürekli fonksiyona istenilen düzeyde yaklaşılabileceği gösterilmiştir. Sigmoid fonksiyonu sinir ağına doğrusal olmama özelliğini kazandıran birimdir. Böyle olmasına rağmen Sigmoid fonksiyonunun özelliği gereği uç gözlemlere karşı tepkisizdir. Fonksiyona giren net girdi değerleri belirli bir eşik değerini aştıkları takdirde fonksiyon bu gibi değerlerin arasında ayırım yapmamakta yaklaşık aynı değerleri çıktı olarak vermektedir. Tüm bunlar dikkate alındığında bu çalışmada sadece gizli birimler içersine Sigmoid fonksiyonu yerleştirilmiş, çıktı katmanı içindeki birimlere ise doğrusal transfer fonksiyonu konmuş ve duyarlılık formülasyonu buna göre düzenlenmiştir. Lu vd. temel geri yayılım veya eğim iniş algoritmasını kullanarak ağın eğitimini tamamlamışlardır. Eğim iniş algoritması performans fonksiyonunu minimize eden ağırlık değerlerine ulaşmada ağı eğitirken çok yavaş bir işleyiş göstermektedir. Bu çalışmada temel geri yayılım algoritmasını önemli derecede hızlandıran ve uygulamalarda sıklıkla kullanılan temel algoritmanın varyasyonları üzerinde

duyarlılık analizi çalışması yürütülmüştür. Ayrıca uygulamada kolaylık sağlaması bakımından önerilen duyarlılık analizinde kullanıcılara rehberlik sağlayacak yapılması gereken adımların sırasıyla tanımlandığı bir tanım verilmiştir. Lu vd. duyarlılık formülasyonunu kullanmadan önce hatanın belirli bir seviyenin altına düşmesiyle ağın eğitimini durdurarak ağın ağırlık değerlerini sabitlemektedirler. Sadece bu ağırlık değerlerini alarak veri setinde bulunan her bir nokta için tek bir duyarlılık değeri hesaplamaktadırlar. Hatta genelleştirme yeteneğinin artması için veri setine dayanarak veri sayısının çoğaltılmasına gitmektedirler. Daha sonra bu noktalardan elde edilen duyarlılık değerlerinin frekans sınıflarını oluşturarak histogramdaki dağılıma bakarak duyarlılıkla ilgili yorum yapmaktadırlar. Bilgisayar ortamında yürütülen çalışmalarda istenilen ağ tasarımı ve hata hangi sınırın altındayken eğitim dursun kararından sonra elde edilen ağırlık değerlerinin aynı işlemin tekrar yapılmasıyla elde edilen ağırlık değerlerinden farklı olduğu görülmüştür. Bunun nedeni öğrenme oranı, hızlandırıcı katsayısı, başlangıç ağırlık değerlerinin atanması ve performans yüzeyinin şekli gibi birçok bilinmeyen faktörün işin içersine girip farklı ağırlık değerlerinin elde edilmesini sağlamasıdır. Dolayısıyla tek bir ağırlık setine bağlı kalarak her bir nokta için tek bir duyarlılık değeri bulunması yerine ağın ağırlıklarını elde etme sürecinin tekrarlanması daha iyi sonuçlar verdiği görülmüştür. Böylece artık ağ sabitlenip veri türetilmesi yoluna gidilmeyecektir. Verilerin türetilmesi eldeki verilerin dağıldığı frekanslardan yola çıktığı için aslında aynı noktaların tekrar edilmesinden başka bir şey değildir. Ayrıca gerçek hayatta yeterince veri noktasının bulunabilmesi de ayrı bir problemdir. Bunun sonucu olarak bu uygulamada veri sayısı sabitlenmiş ve ağırlık değerlerinin türetilmesi yoluna gidilmiştir. Her bir ağırlık seti için eldeki her bir noktanın kullanılmasıyla elde edilen duyarlılık dağılımını temsil eden bir tanımlayıcı istatistik değeri bulunmuştur. Artık ağırlık setlerinin sayısı gözlem sayısı olduğundan her bir gözlemden elde edilen tanımlayıcı istatistik değerlerinin dağılımına bakılmış ve bu dağılımdan tekrar bir tanımlayıcı istatistik elde etmeye çalışılmıştır. Çalışmada kullanılan tanımlayıcı istatistikler; ortalama, medyan ve kırılmış ortalama değerleridir. Nihai olarak elde edilen tanımlayıcı istatistik değeri duyarlılık değerinin nokta tahmini olarak kullanılmıştır. Özellikle verilerdeki gürültü arttıkça yazarlar tarafından önerilen ağın sabitlenerek veri türetilmesine dayanan duyarlılık değeri

hesaplamalarında sapmalar çıkmasının gözlenmesi sonucunda biraz önce anlatılan yöntemin kullanılması yoluna gidilmiştir.

Yöntemin kullanılmasını ve yararını test etmek amacıyla 4 farklı senaryo türetilerek, burada bilindik bir yöntem olan çoklu doğrusal regresyon analiziyle kıyaslanmasına çalışılmıştır. Altta yatan fonksiyonun doğrusal model olmasından ötürü ilgilenilen ve duyarlılık analiziyle tahminlenecek parametre değerleri her bir bağımsız değişkenin önündeki katsayılarıdır. Matlab ortamında 0 ve 1 arasında Uniform dağılıma sahip 4 girdi değişkeninden 350 'şer tane türetilmiştir. Ayrıca duyarlılık analizi kapsamında kullanılacak yöntemi standart olarak uygulayan bir bilgisayar programı olmamasından ötürü Matlab ortamında kod yazımına gidilmiştir. Birinci senaryo olarak, incelenen problemin altında yattığı varsayılan doğrusal fonksiyona 0 ve 1 arasında normal dağılan gürültü değerleri eklenerek bağımlı değişkenin değerleri bulunmuştur. Burada oluşturulan veriler çoklu doğrusal regresyon analizinin tüm varsayımlarını sağlayacak şekilde düzenlenmiştir. Öncelikle regresyon analiziyle katsayıların tahmini gerçekleştirilmiş ve daha sonra aynı değerler sinir ağlarıyla tahminlenmiştir. Eldeki veri grubuna en uygun yöntem olan regresyon analizi gerçek parametre değerlerine sinir ağlarına nazaran daha yakın sonuçlar üretmiştir. Bu zaten gerçekleşmesi beklenen bir durumdur. Ancak amaç zaten en baştan saf bir karşılaştırma yapmaktan ziyade sinir ağlarında duyarlılık analizi adı altında önerilen metodun doğruluğunun gösterilmesidir. Bu kapsamda sinir ağlarıyla elde edilen sonuçlar yöntemin doğru bir şekilde çalıştığının göstergesi olmuştur. İkinci senaryo olarak, çoklu regresyon varsayımlarından biri olan hata teriminin normal dağılması varsayımı ihlal edilecek şekilde veriler düzenlenmiştir. Sırasıyla bu verilere regresyon ve sinir ağı uygulanmıştır. Elde edilen sonuç küçük farkla da olsa sinir ağının katsayıları tahmin etmede regresyona göre daha iyi sonuç verdiğidir. Üçüncü senaryoda, hata teriminin ardışık değerleri arasında ilişki kurularak veriler düzenlenmiştir. Burada verilerde otokorelasyon olduğunun tespiti yapılarak otokorelasyonun kaldırılmasına yönelik kullanılan yöntem sonrası katsayı tahminleri yapılmıştır. Her ne kadar sinir ağlarıyla yapılan tahminde önceki senaryolara nazaran daha fazla bir bozulma gerçekleşse de regresyon analizine kıyasla çok daha iyi sonuçlar vermiştir. Sinir ağların tahmininde yaşanan

bozulmanın kaynağı hata teriminin etkisiyle ağa verilen bağımlı değişkenin asıl değerinden uzaklaşmasıdır. Başka bir deyişle yanlış bağımlı değişken değerleri verilmekte ve ağda buna dayanarak sonuçlarını elde etmektedir. Sinir ağı tahminindeki bozulmanın nedeni herhangi bir varsayımın sağlanmamasıyla alakalı değildir. Dördüncü ve son senaryo olarak iki değişken arasında doğrusal bir ilişki olacak şekilde verilerin türetilmesi yoluna gidilmiştir. Çoklu doğrusal bağlantı bulunması halinde regresyon analiziyle özellikle doğrusal bağlantılı değişkenlerin katsayı tahmininde çok büyük derecede bozulma meydana gelmiştir. Buna rağmen sinir ağı sonuçlarında kayda değer büyük bir sorun ortaya çıkmamıştır. Değişkenler arasındaki ilişkileri dikkate almadan sonuçları vermeye devam etmiştir. Ayrıca tüm senaryolar için geçerli olan bir bulgudan bahsetmek gerekirse sinir ağı her senaryoda regresyon analizine göre bağımlı değişkenin tahmin hatalarını daha küçük seviyede tutmuştur. Sonuç olarak sinir ağlarında duyarlılık analiziyle uygulanan metodun herhangi bir varsayım gerektirmeden çalışabilirliği gösterilmiştir. Yöntemin tek ihtiyacı olan şey, doğru bir bağımlı ve bağımsız değişkenlerin oluşturduğu veri setinin bulunmasıdır. Elbette metodun işlevliliğinin geliştirilmesine yönelik yapılması gereken çalışmalar bulunmaktadır. Bunlar aşağıda özetlenmiş bulunmaktadır.

- ❖ Sigmoid yerine farklı transfer fonksiyonları ile analizin hassasiyetinin artırılmasına çalışılabilir.
- ❖ Ağın eğitiminde gerçekleştirilecek gelişme daha iyi duyarlılık sonuçlarının elde edilmesine sebep olacaktır.
- ❖ Doğrusal olmayan modellerde önerilen metodun incelemesi yapılabilir.
- ❖ Veri sayısının az olması veya varsayımların bir kısmının karşılanmadığı durumlarda klasik EKK ile arasındaki farklar incelenebilir.
- ❖ 0 ve 1 arasında normalize edilmiş veriler yerine orijinal verilerin kullanıldığı varsayımı altında analizinin geliştirilmesi yapılabilir.

## KAYNAKLAR

Abdi, H., Valentin, D. ve Edelman, B. (1999). *Neural Networks* (Sage University Papers Series on Quantitative Application in the Social Sciences, series no.07-124). Thousand Oaks, CA: Sage.

Adams, J. A. ve Wert, Y. (2005). Logistic and Neural Network Models for Predicting a Hospital Admission. *Journal of Applied Statistics*. 32(8): 861–869.

Akkaya, Ş. ve Pazarlıođlu, M. V. (2000). *Ekonometri I*. İzmir: Berk Masa Üstü Yayıncılık.

Arbib, M.A. (2003). *The Handbook of Brain Theory and Neural Networks*. London: MIT Press.

Brouwer, R.K ve Pedrycz, W. (2003). Training a feed-forward network with incomplete data due to missing input variables. *Applied Soft Computing* 3: 23–36.

Chen, A.S. ve Leung, M. T. (2005). Performance Evaluation of Neural Network Architectures: The Case of Predicting Foreign Exchange Correlations. *Journal of Forecasting*. 24: 403–420.

Cheng, B. ve Titterington, D. M. (1994). Neural Networks: A Review from a Statistical Perspective. *Statistical Science*. 9(1): 2-54.

Choudhury, A., Nair, P. B. ve Keane, A. J. (2006). Constructing a speculative kernel machine for pattern classification. *Neural Networks*. 19: 84–89.

Courrieu, P. (2006). Density codes and shape spaces. *Neural Networks*. 19: 429–445.

Curram, S. P. ve Mingers, J. (1994). Neural Networks, Decision Tree Induction and Discriminant Analysis: An Empirical Comparison. *The Journal of the Operational Research Society*. 45(4): 440-450.

de Carvalho, M. C. M., Dougherty, M.S., Fowkes, A. S. ve Wardman, M. R. (1998). Forecasting Travel Demand: A Comparison of Logit and Artificial Neural Network Methods. *The Journal of the Operational Research Society*. 49(7): 717-722.

Demuth, H., Beale, M. ve Hagan, M. (2006). *Neural Network Toolbox User's Guide*. The Matworks, Inc.

Dunis, C. L. ve Huang, X. (2002). Forecasting and Trading Currency Volatility: An Application of Recurrent Neural Regression and Model Combination. *Journal of Forecasting*. 21: 317–354.

Elmas, Ç. (2003). *Yapay Sinir Ağları*. Ankara: Seçkin.

Ertuş, K. (2007). *Multivariate Statistical Models*. Unpublished Lecture Notes, Institute of Social Sciences, Dokuz Eylül University, İzmir.

Fausett, L. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. N.J.:Prentice-Hall, Englewood Cliffs.

Freeman, J. A. ve Skapura, D. M. (1991). *Neural Networks: Algorithms, Applications and Programming Techniques*. California: Addison-Wesley Publishing Company.

Galizia, C.G. ve Szyszka, P. (2008). Olfactory coding in the insect brain: molecular receptive ranges, spatial and temporal coding. *Entomologia Experimentalis et Applicata*. 128: 81–92.

Garson, G. D. (1997). *Neural Networks: An Introductory Guide for Social Scientists*. London: Sage.



Gill, P. E., Murray, W. ve Wright, M. H. (1981). *Practical Optimization*. San Diego: Academic Press.

Gujarati, D.N. (2005). *Temel Ekonometri*. İstanbul: Literatür Yayıncılık.

Gupta, A. ve Lam, M. S. (1996). Estimating Missing Values Using Neural Networks. *The Journal of the Operational Research Society*. 47(2): 229-238.

Gurney, K. (1997). *An Introduction to Neural Networks*. London: UCL.

Hagan, M. T. ve Menhaj, M. B. (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*. 5(6): 989-993.

Hagan, M. T., Demuth, H. B. ve Beale, M. (1996). *Neural Network Design*. Boston: PWS.

Hakeem, M.A., Kamil, M. ve Arman, I. (2008). Prediction of temperature profiles using artificial neural networks in a vertical thermosiphon reboiler. *Applied Thermal Engineering* 28:1572–1579.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. New York: Macmillan.

He, S. ve Unbehauen, R. (1998). Approximate Feedback Linearisation Using Multilayer Neural Networks. *Neural Processing Letters*. 8: 131–144.

Iwataa, K., Ikedab, K. ve Sakai, H. (2006). The asymptotic equipartition property in reinforcement learning and its relation to return maximization. *Neural Networks* 19: 62–75.

Jain, A.S ve Meeran, S. (1998). Job-Shop Scheduling Using Neural Networks. *International Journal of Production Research*. 36(5): 1249-1272.

Kimura, Y., Wakahara, T. ve Tomono, A. (2005). Combination of Statistical and Neural Classifiers for a High-Accuracy Recognition of Large Character Sets. *Systems and Computers in Japan*. 36(9): 97-107.

Knowles, R. P. (1997). Predicting Construction Labour Productivity Using Neural Networks. MSc thesis. University of Alberta, Edmonton, Canada.

Kocakoç Deveci, İ. (2007). *Matlab ve İstatistiksel Veri Analizi*. Ankara: Nobel.

Kurfess, F. J. (2000). Neural Networks and Structured Knowledge: Rule Extraction and Applications. *Applied Intelligence*. 12: 7–13.

Lam, M. (2004). Neural network techniques for financial performance prediction:integrating fundamental and technical analysis. *Decision Support Systems*. 37: 567– 581.

Lenze, B. (1999). Mathematics and Neural Networks–A Glance at some Basic Connections. *Acta Applicandae Mathematicae*. 55: 303–311.

Li, H., Shen, L. Y. ve Love, P. E. D. (1999). ANN-Based Mark-Up Estimation System With Self-Explanatory Capacities. *Journal of Construction Engineering and Management*. 125(3): 185-189.

Lu, M., AbouRizk, S. M. ve Hermann, U. H. (2001). Sensitivity Analysis of Neural Networks in Spool Fabrication Productivity Studies. *Journal of Computing in Civil Engineering*. 15(4): 299-308.

Malakooti, B. ve Raman, V. (2000). Clustering and selection of multiple criteria alternatives using unsupervised and supervised neural networks. *Journal of Intelligent Manufacturing*. 11: 435-451.

Mukherjee, A. ve Deshpande, J. M. (1995). Modeling Intial Design Process Using Artificial Neural Networks. *Journal of Computing in Civil Engineering*.9(3):194-200.

Nguyen, D. ve Widrow, B. (1990). Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights. Proceedings of the IJCNN. vol3: 21-26.

Öztemel, E. (2003). Yapay Sinir Ağları. İstanbul: Papatya.

Pedregal, P. (2004). Introduction to Optimization. New York: Springer.

Poli, I. ve Jones, R. D. (1994). A Neural Net Model for Prediction. Journal of the American Statistical Association. 89(425): 117-121.

Qazi, M. ve Linshu, H. (2006). Nearly-orthogonal sampling and neural network metamodel driven conceptual design of multistage space launch vehicle. Computer-Aided Design 38:595–607.

Ripley, B. D. (1994). Neural Networks and Related Methods for Classification. Journal of the Royal Statistical Society. 56(3): 409-456.

Sakumura, Y. ve Ishii, S. (2006). Stochastic resonance with differential code in feedforward network with intra-layer random connections. Neural Networks. 19: 469–476.

Samanta, B., Bandopadhyay, S. ve Ganguli, R. (2006). Comparative Evaluation of Neural Network Learning Algorithms for Ore Grade Estimation. Mathematical Geology. 38(2): 175–197.

Schindler, K. H. ve Sanguineti, M. (2003). Bounds on the Complexity of Neural-Network Models and Comparison with Linear Methods. International Journal of Adaptive Control and Signal Processing. 17:179–194.

Segal, M.E., Goodman, P.H., Goldstein, R., Hauck, W., Whyte, J., Graham, J.W., Polansky, M.ve Hammond, F.M. (2006). The Accuracy of Artificial Neural

Networks in Predicting Long-term Outcome After Traumatic Brain Injury. *Journal of Head Trauma Rehabilitation*. 21(4): 298–314.

Sinha, S. K. ve McKim, R. A. (2000). Artificial Neural Network for Measuring Organizational Effectiveness. *Journal of Computing in Civil Engineering*. 14(1):9-14.

Strang, G. (1988). *Linear Algebra and Its Applications*, New York: Academic Press.

Şen, Z. (2004). *Yapay Sinir Ağları İlkeleri*. İstanbul: Su Vafkı.

Tagliaferri, R., Eleuteri, A. ve Barone, F. (2001). Fuzzy min-max neural networks: from classification to regression. *Soft Computing*. 5: 69-76.

Talavan, P. M. ve Yanez, J. (2006). The generalized quadratic knapsack problem: A neuronal network approach. *Neural Networks*. 19: 416–428.

Wang, S. (1995). The Unpredictability of Standard Back Propagation Neural Networks in Classification Applications. *Management Science*. 41(3): 555-559.

Wang, S. (2005). Classification with incomplete survey data: a Hopfield neural network approach. *Computers & Operations Research*. 32: 2583–2594.

Warner, B. ve Mısra, M. (1996). Understanding Neural Networks as Statistical Tools. *The American Statistician*. 50(4): 284-293.

West, P. M., Brockett, P. L. ve Golden, L. L. (1997) .A Comparative Analysis of Neural Networks and Statistical Methods for Predicting Consumer Choice. *Marketing Science*. 16(4): 370-391.

Yao, J., Teng, N., Poh, H. L. ve Tan, C. L. (1998). Forecasting and Analysis of Marketing Data Using Neural Networks. *Journal of Information Science and Engineering*. 14(4): 523-545.

Yoon, Y., Swales, G. J. R. ve Margavio, T. M. (1993). A Comparison of Discriminant Analysis Versus Artificial Neural Networks. *The Journal of the Operational Research Society*. 44(1): 51-60.

Zhang, P. G. (2004). *Neural Networks in Business Forecasting*. London: Idea Group Publishing.

Zia, Ö. (2000). Application of intelligent control in the design of a test flight maneuver autopilot: An example of cooperation between NASA and a university. *Computer Applications in Engineering Education*. 8(3-4): 196-2.

# EKLER

## EK1: Terimler Sözlüğü

İtalik olan terimlerin yine sözlük içerisinde anlamlarına yer verilmiştir (Gurney, 1997:12).

**Kuvvet gerilimi (Action potential):** Bir sinir hücresinin aktif çıktısını oluşturan basamaklıp voltajlardır. *Akson* aracılığı ile diğer nöronlara yayılırlar.

**İçeri götüren (Afferent):** Belirli bir sinir hücresini ele aldığımızda o sinir hücresini uyaran (*sinirle donatan*) aksona verilen isimdir.

**Akson (Axon):** *Hücre gövdesinden çıkan liftir ve kuvvet gerilimlerinin diğer hücrelere iletilmesini sağlar.*

**Akson tepeciği (Axon hillock):** *Akson ve hücre gövdesinin birleştiği yerdir. Burada hücre zarı gerilimi belirli bir eşik değerini aşarsa kuvvet gerilimleri oluşturulur.*

**Akson terminali (Axon terminal):** Bir *sinapsin presinaptik* bileşeni olan akson terminalinde bir akson *eşit öneme sahip yol seti* kollarına ayrılır.

**Kimyasal bağlanma (Chemical binding):** *Nöroileticilerin bir yakalayıcı bölge ile kimyasal olarak birleşerek PSP lerin başlatıldığı süreçtir.*

**Eşit öneme sahip yol seti (Collateral):** Akson, diğer birçok nöronla veya bir nöronla çok sayıda iletişim kurulmasına olanak verecek şekilde eşit öneme sahip yol seti şeklinde kollara ayrılır.

**Dentrit (Dendrite):** Sinir hücresinin dallanan liflerinden biridir ve *PSP* ler aracılığıyla girdi bilgilerinin hücreye iletilmesini sağlar.

**Nötürleşme (Depolarization):** Sinir hücrenin *hücre zarı gerilimi* negatif dinlenme değerinde veya bir başka deyişle denge değerinde konumlanmaktadır. Bu negatif değerin azaltılması nötürleşmeye yol açar. *Akson tepciğindeki* yeterli nötürleşme bir kuvvet geriliminin ortaya çıkmasına neden olacaktır.

**Dışarı götüren (Efferent):** Aksonun *eşit öneme sahip yollara* ayrılmasıyla diğer nöronlarla iletişimi sağlayan nörona verilen isimdir.

**EPSP:** Tetikleyici Postsinaptik Gerilim (Excitatory Postsynaptic Potential). Hücre zarını *nötürleştiren* PSP ye verilen isimdir.

**Kutupsallaşma (Hyperpolarization):** Sinir hücrenin *hücre zarı gerilimi* negatif dinlenme değerinde veya bir başka deyişle denge değerinde konumlanmaktadır. Bu negatif değerin daha da negatifleştirilmesi bir kutupsallaşmaya yol açar ve hücre zarını *nötürleştiren* EPSP lerin oluşturulmasını engeller.

**Sinirle donatma (Innervate):** B nöronuna sinyal gönderen A nöronu için B nöronunun sinirle donatılması denilmektedir.

**IPSP:** Engelleyici Postsinaptik Gerilim (Inhibitory Postsynaptic Potential). Hücre zarını daha da *kutupsallaştıran* PSP ye verilen isimdir.

**Zar gerilimi (Membrane potential):** Sinir zarının herhangi bir noktasındaki voltaj değeridir.

**Nöroiletici (Neurotransmitter):** *Sinaptik uçurum* boyunca yayılarak sinaptik faaliyete aracılık eden kimyasal madde.

**Organel (Organelle):** Metabolizma içerisinde yer alan alt hücresel bileşenlerdir.

**Postsinaptik zar (Postsynaptic membrane):** *Dentrit* üzerine yerleşmiş bir *sinapsin* parçasıdır ve dentritik zar ile birlikte *yakalayıcı bölgelerden* oluşur.

**Gerilim farklılığı (Potential difference):** Hücre zarı boyunca ki gerilim farklılığı.

**Presinaptik zar (Presynaptic membrane):** *Akson terminali* üzerine yerleştirilmiş herhangi bir *sinapsin* parçasıdır.

**PSP:** Postsinaptik Gerilim (Postsynaptic Potential). *Sinapsdaki* faaliyet sonucu ortaya çıkan *hücre zarı gerilimindeki* değişimdir.

**Yakalayıcı bölge (Receptor sites):** *Nöroiletici* moleküllerin bağlandığı *postsinaptik zar*daki bölgelerdir. Bu kimyasal bağlanma bir PSP nin oluşturulması sürecini başlatır.

**Durgunluk dönemi (Refractory period):** İki *kuvvet gerilimi* oluşturma süreci arasındaki en kısa zamandır.

**Sinaps (Synapse):** Nöronlar arasındaki sinyal ve fiziksel iletişimi sağlayan bölgedir. Bir *sinapsin akson terminalinde kuvvet geriliminin* alınmasıyla *nöroileticiler sinaptik uçuruma* bırakılır ve *postsinaptik zara* doğru harekete geçerler. Orada yakalayıcı bölgeler ile kimyasal olarak bağlanarak postsinaptik gerilim (PSP) üretimi başlatılır.

**Sinaptik uçurum (Synaptic cleft):** Sinaptik faaliyet süresince kimyasal *nöroileticilerin* yayıldığı postsinaptik zar ve presinaptik zar arasındaki boşluktur.

**Kabarcık (Vesicles):** *Akson terminalinde* bulunan içerisinde *nöroileticileri* taşıyan küresel konteynirlardır. Akson terminaline bir kuvvet geriliminin ulaşmasıyla kabarcıklar içerdikleri nöroileticileri *sinaptik uçuruma* bırakırlar.



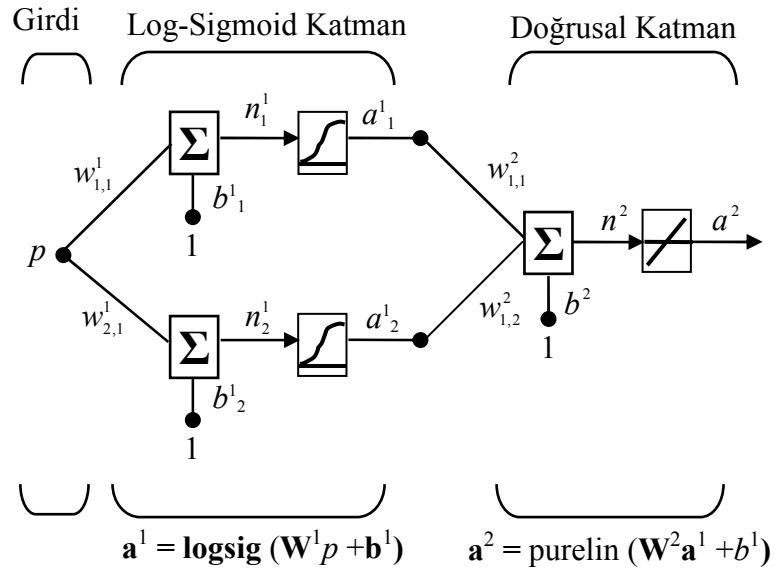
## EK2: Temel Geri Yayılım Algoritmasının Sayısal Bir Örneği

Geri yayılım algoritmasının işleyişini tarif etmek amacıyla bir ağ seçilmiş ve belirli bir problem üzerine uygulamasına yer verilmiştir (Hagan, Demuth ve Beale, 1996:11-14). Kolaylık sağlama açısından Şekil 1-1 'de oluşturulan ağın yapısı gösterilmiştir.

Ağın çözeceği bir problemi tanımlamak amacıyla aşağıdaki fonksiyona yaklaşmak istediğimizi varsayalım.

$$g(p) = 1 + \sin\left(\frac{\pi}{4} p\right) \quad \text{için } -2 \leq p \leq 2$$

Eğitim setini elde etmek için  $p$  'nin birkaç değerinde bu fonksiyonun alacağı değerler belirlenir.



Şekil 1-1: Fonksiyon Yaklaşım Ağı Örneği

Geri yayılım algoritmasına başlamadan önce ağın ağırlık ve artık değerleri için başlangıç değerlerinin seçilmesine ihtiyaç vardır. Bunlar genellikle küçük rassal değerler olarak seçilmektedir. Değerlerin şöyle olduğunu varsayalım.

$$\mathbf{W}^1(0) = \begin{bmatrix} -0,27 \\ -0,41 \end{bmatrix}, \mathbf{b}^1(0) = \begin{bmatrix} -0,48 \\ -0,13 \end{bmatrix}, \mathbf{W}^2(0) = [0,09 \quad -0,17], \mathbf{b}^2(0) = [0,48]$$

Şimdi algoritmayı başlatmak için başlangıç girdiyi  $p=1$  olarak seçtiğimizi varsayalım.

$$a^0 = p = 1$$

İlk katmanın çıktısı,

$$\begin{aligned} \mathbf{a}^1 &= \mathbf{f}^1(\mathbf{W}^1 \mathbf{a}^0 + \mathbf{b}^1) = \mathbf{logsig}\left(\begin{bmatrix} -0,27 \\ -0,41 \end{bmatrix} [1] + \begin{bmatrix} -0,48 \\ -0,13 \end{bmatrix}\right) = \mathbf{logsig}\left(\begin{bmatrix} -0,75 \\ -0,54 \end{bmatrix}\right) \\ &= \begin{bmatrix} \frac{1}{1 + e^{0,75}} \\ \frac{1}{1 + e^{0,54}} \end{bmatrix} = \begin{bmatrix} 0,321 \\ 0,368 \end{bmatrix} \end{aligned}$$

İkinci katmanın çıktısı,

$$a^2 = f^2(\mathbf{W}^2 \mathbf{a}^1 + \mathbf{b}^2) = \mathit{purelin}([0,09 \quad -0,17] \begin{bmatrix} 0,321 \\ 0,368 \end{bmatrix} + [0,48]) = [0,446]$$

Hata şöyle olacaktır:

$$e = t - a = \left\{1 + \sin\left(\frac{\pi}{4} p\right)\right\} - a^2 = \left\{1 + \sin\left(\frac{\pi}{4} 1\right)\right\} - 0,446 = 1,261$$

Algoritmanın bir sonraki adımı duyarlıkların geri yayılmasıdır. Geri yayılmaya başlamadan önce hatırlanacağı gibi,  $\hat{f}^1(n)$  ve  $\hat{f}^2(n)$ , transfer fonksiyonlarının türevlerine ihtiyaç vardır. İlk katman için;

$$\hat{f}^1(n) = \frac{d}{dn} \left( \frac{1}{1 + e^{-n}} \right) = \frac{e^{-n}}{(1 + e^{-n})^2} = \left(1 - \frac{1}{1 + e^{-n}}\right) \left(\frac{1}{1 + e^{-n}}\right) = (1 - a^1)(a^1)$$

İkinci katman için;

$$\dot{f}^2(n) = \frac{d}{dn}(n) = 1$$

Artık geri yayılımı uygulayabiliriz. Başlangıç noktası eşitlik (2.75) 'in kullanılmasıyla ikinci katmandan hesaplanır.

$$\mathbf{s}^2 = -2\dot{\mathbf{F}}^2(\mathbf{n}^2)(\mathbf{t} - \mathbf{a}) = -2[\dot{f}^2(n^2)](1,261) = -2[1](1,261) = -2,522$$

İlk katmanın duyarlılığı, eşitlik (2.76) 'ı kullanılarak ikinci katmandaki duyarlılığın geri yayılmasıyla hesaplanır.

$$\begin{aligned} \mathbf{s}^1 &= \dot{\mathbf{F}}^1(\mathbf{n}^1)(\mathbf{W}^2)^T \mathbf{s}^2 = \begin{bmatrix} (1-a_1^1)(a_1) & 0 \\ 0 & (1-a_2^1)(a_2^1) \end{bmatrix} \begin{bmatrix} 0,09 \\ -0,17 \end{bmatrix} [-2,522] \\ &= \begin{bmatrix} (1-0,321)(0,321) & 0 \\ 0 & (1-0,368)(0,368) \end{bmatrix} \begin{bmatrix} 0,09 \\ -0,17 \end{bmatrix} [-2,522] \\ &= \begin{bmatrix} 0,218 & 0 \\ 0 & 0,233 \end{bmatrix} \begin{bmatrix} -0,227 \\ 0,429 \end{bmatrix} = \begin{bmatrix} -0,0495 \\ 0,0997 \end{bmatrix} \end{aligned}$$

Algoritmanın son aşaması ağırlıkların güncellenmesidir. Kolaylık sağlaması açısından öğrenme oranını  $\alpha = 0.1$  olarak alınmıştır. Eşitlik 2.78 ve 2.79 aracılığıyla,

$$\begin{aligned} \mathbf{W}^2(1) &= \mathbf{W}^2(0) - \alpha \mathbf{s}^2 (\mathbf{a}^1)^T = [0,09 \quad -0,17] - 0,1[-2,522][0,321 \quad 0,368] \\ &= [0,171 \quad -0,0772] \end{aligned}$$

$$\mathbf{b}^2(1) = \mathbf{b}^2(0) - \alpha \mathbf{s}^2 = [0,48] - 0,1[-2,522] = [0,732]$$

$$\mathbf{W}^1(1) = \mathbf{W}^1(0) - \alpha \mathbf{s}^1 (\mathbf{a}^0)^T = \begin{bmatrix} -0,27 \\ -0,41 \end{bmatrix} - 0,1 \begin{bmatrix} -0,0495 \\ 0,0997 \end{bmatrix} [1] = \begin{bmatrix} -0,265 \\ -0,420 \end{bmatrix}$$

$$\mathbf{b}^1(1) = \mathbf{b}^1(0) - \alpha \mathbf{s}^1 = \begin{bmatrix} -0,48 \\ -0,13 \end{bmatrix} - 0,1 \begin{bmatrix} -0,0495 \\ 0,0997 \end{bmatrix} = \begin{bmatrix} -0,475 \\ -0,140 \end{bmatrix}$$

Bu geri yayılım algoritmasının ilk iterasyonunu oluşturmaktadır. Bir sonraki iterasyona geçmeden önce diğer bir  $p$  girdisi seçilir ve iterasyona devam edilir. Ağın cevabı ve hedef fonksiyon arasındaki fark kabul edilebilir bir düzeye ulaşıncaya kadar iterasyonlara devam edilir.

### EK3: Eşlenik Gradient Algoritmasının Sayısal Bir Örneği

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \mathbf{x} \text{ fonksiyonu } \mathbf{x}_0 = \begin{bmatrix} 0,8 \\ -0,25 \end{bmatrix} \text{ başlangıç değeriyle verilmiştir.}$$

Eşlenik Gradient algoritması kullanılarak bir iterasyon ilerletilmiştir. Fonksiyon karşılaştırılması ile aralık belirlenmesine ve Golden Section araması ile aralık düşürülmesine karar verilmiştir (Hagan, Demuth ve Beale, 1996:12-40). İlk olarak fonksiyonun gradienti hesaplanırsa:

$$\nabla F(\mathbf{x}) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 2x_2 \end{bmatrix}$$

Eğim inişte de olduğu gibi eşlenik gradient algoritması için ilk arama yönü gradientin negatiftir.

$$\mathbf{p}_0 = -\mathbf{g}_0 = -\nabla F(\mathbf{x})^T \Big|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} -1,35 \\ -0,3 \end{bmatrix}$$

İlk iterasyon için  $F(\mathbf{x})$  fonksiyonunun doğru boyunca minimizasyonuna ihtiyaç vardır:

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{p}_0 = \begin{bmatrix} 0,8 \\ -0,25 \end{bmatrix} + \alpha_0 \begin{bmatrix} -1,35 \\ -0,3 \end{bmatrix}$$

Sıra aralık belirlenmesine gelmiştir. Başlangıç adım büyüklüğünün  $\varepsilon = 0.075$  olduğu varsayılmıştır. Aralık belirlenmesi aşağıdaki gibi olacaktır.

$$F(\mathbf{a}_1) = F\left(\begin{bmatrix} 0,8 \\ -0,25 \end{bmatrix}\right) = 0.5025$$

$$b_1 = \varepsilon = 0.075, F(b_1) = F\left(\left[\begin{array}{c} 0.8 \\ -0.25 \end{array}\right] + 0.075 \left[\begin{array}{c} -1.35 \\ -0.3 \end{array}\right]\right) = 0.3721$$

$$b_2 = 2\varepsilon = 0.15, F(b_2) = F\left(\left[\begin{array}{c} 0.8 \\ -0.25 \end{array}\right] + 0.15 \left[\begin{array}{c} -1.35 \\ -0.3 \end{array}\right]\right) = 0.2678$$

$$b_3 = 4\varepsilon = 0.3, F(b_3) = F\left(\left[\begin{array}{c} 0.8 \\ -0.25 \end{array}\right] + 0.3 \left[\begin{array}{c} -1.35 \\ -0.3 \end{array}\right]\right) = 0.1373$$

$$b_4 = 8\varepsilon = 0.6, F(b_4) = F\left(\left[\begin{array}{c} 0.8 \\ -0.25 \end{array}\right] + 0.6 \left[\begin{array}{c} -1.35 \\ -0.3 \end{array}\right]\right) = 0.1893$$

Ardışık iki değerlendirme arasında fonksiyon arttığından minimumun  $[0.15, 0.6]$  aralığında olması gerektiği bilinmektedir. Bir sonraki adım Golden Section araması kullanılarak aralık düşürülmesidir.

$$c_1 = a_1 + (1 - \tau)(b_1 - a_1) = 0.15 + (0.382)(0.6 - 0.15) = 0.3219$$

$$d_1 = b_1 - (1 - \tau)(b_1 - a_1) = 0.6 - (0.382)(0.6 - 0.15) = 0.4281$$

$$F_a = 0.2678, F_b = 0.1893, F_c = 0.1270, F_d = 0.1085$$

$F_c > F_d$  olmasından dolayı

$$a_2 = c_1 = 0.3219, b_2 = b_1 = 0.6, c_2 = d_1 = 0.4281$$

$$d_2 = b_2 - (1 - \tau)(b_2 - a_2) = 0.6 - (0.382)(0.6 - 0.3219) = 0.4938$$

$$F_a = F_c = 0.1270, F_b = F_d = 0.1085, F_d = F(d_2) = 0.1232$$

Bu sefer  $F_c < F_d$  olmasından ötürü

$$a_3 = a_2 = 0.3219, b_3 = d_2 = 0.4938, d_3 = c_2 = 0.4281$$

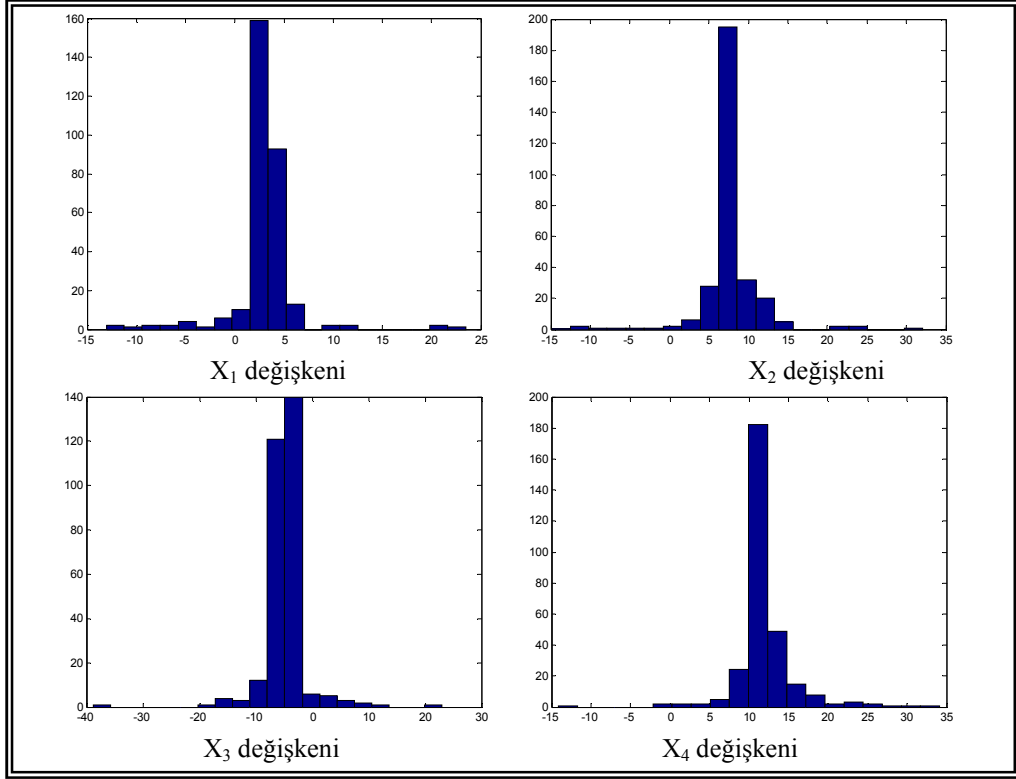
$$c_3 = a_3 + (1 - \tau)(b_3 - a_3) = 0.3219 + (0.382)(0.4938 - 0.3219) = 0.3876$$

$$F_b = F_d = 0.1232, F_a = F_c = 0.1085, F_c = F(c_3) = 0.1094$$

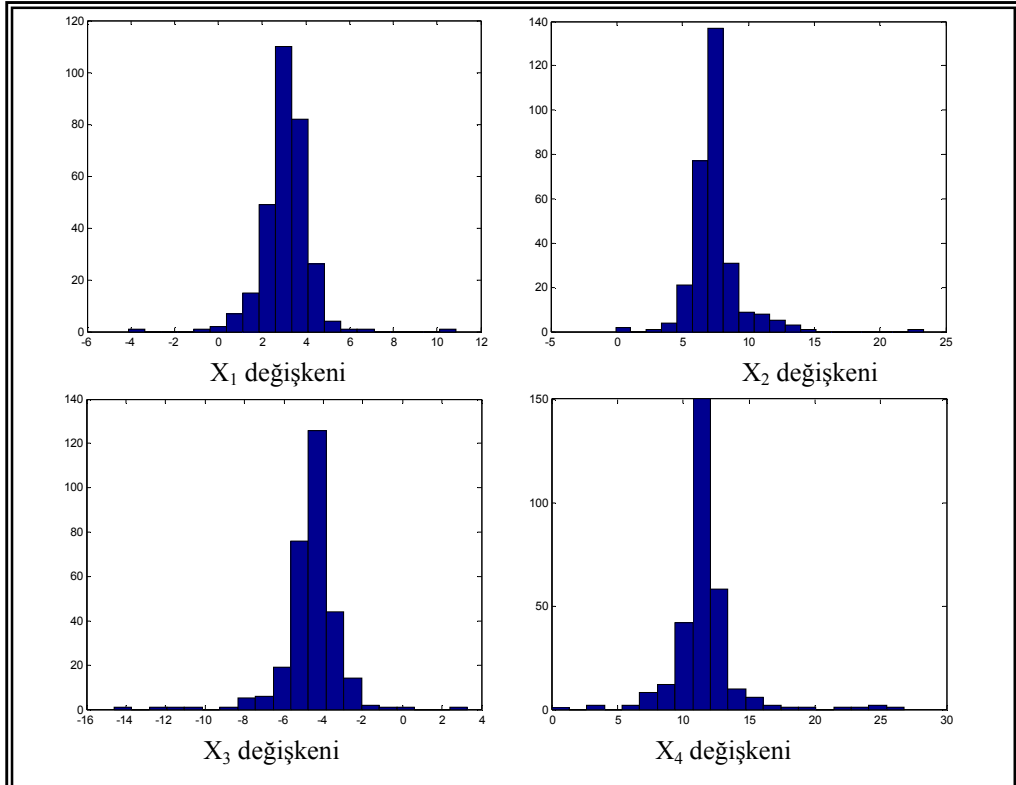
Algoritmaya  $b_{k+1} - a_{k+1} < tol$  oluncaya kadar bu şekilde devam edilir.

## EK4: Durum2 İin Duyarlılık Deęerlerine İlişkin Grafikler

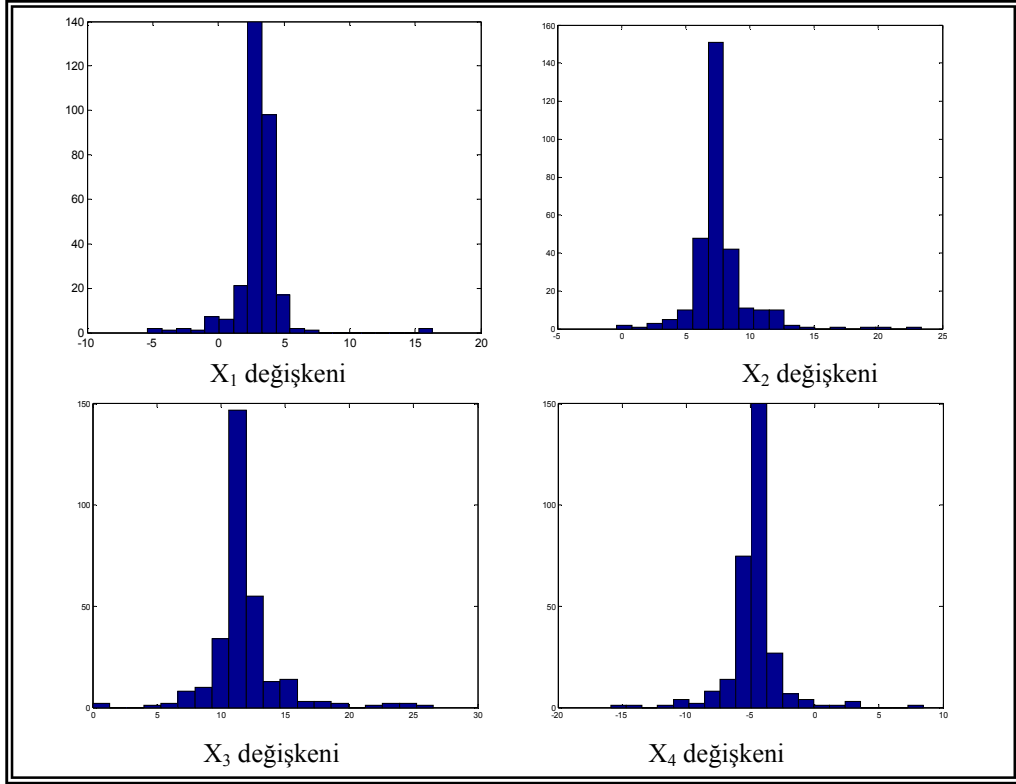
### a) Duyarlılık deęerinin belirlenmesinde kullanılan ortalamaların daęılımları



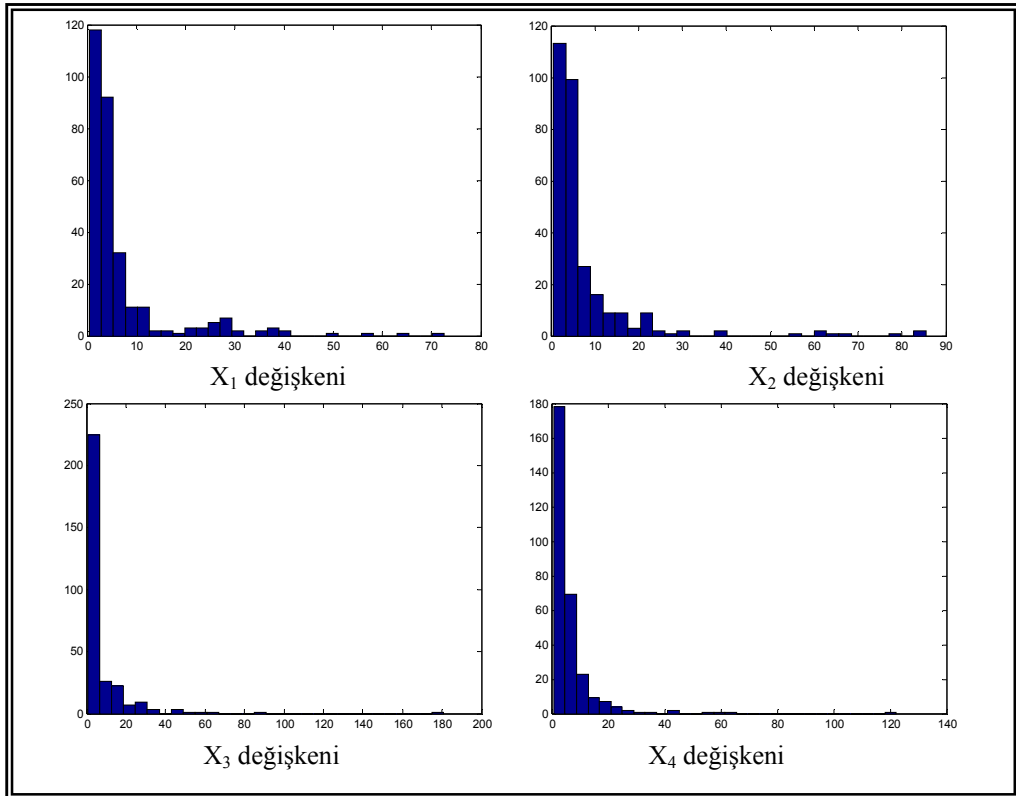
### b) Duyarlılık deęerinin belirlenmesinde kullanılan medyanların daęılımları



**c) Duyarlılık değerin belirlenmesinde kullanılan trimmean(%15) dağılımları**

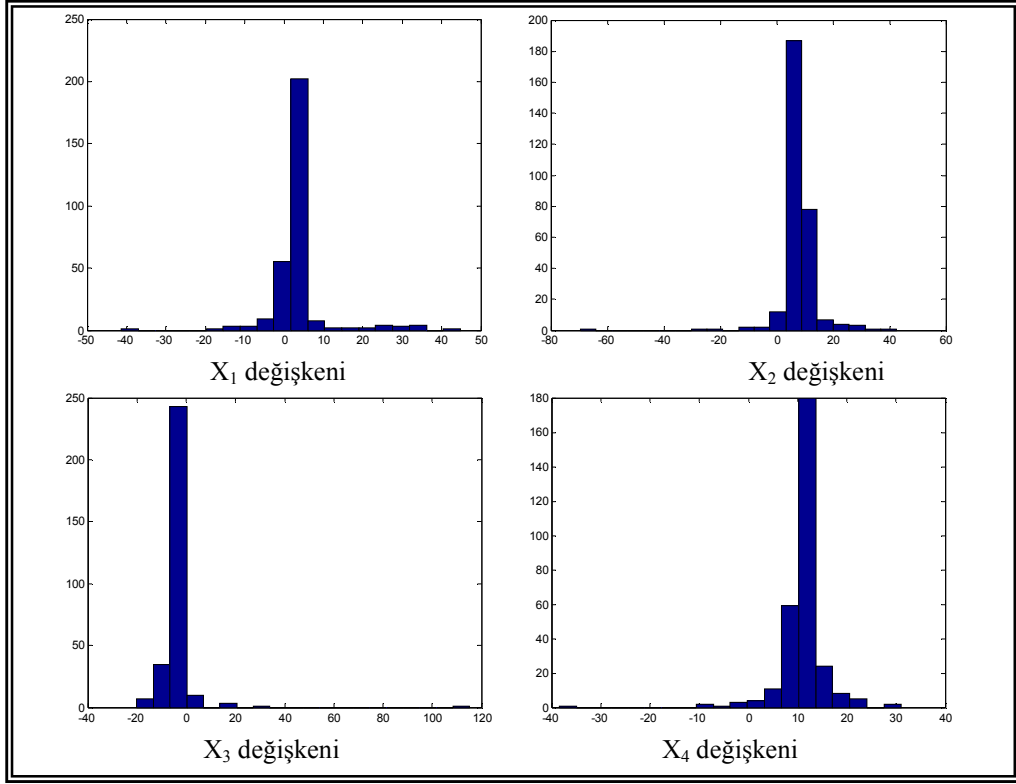


**d) Duyarlılık değlerinin standart sapma dağılımları**

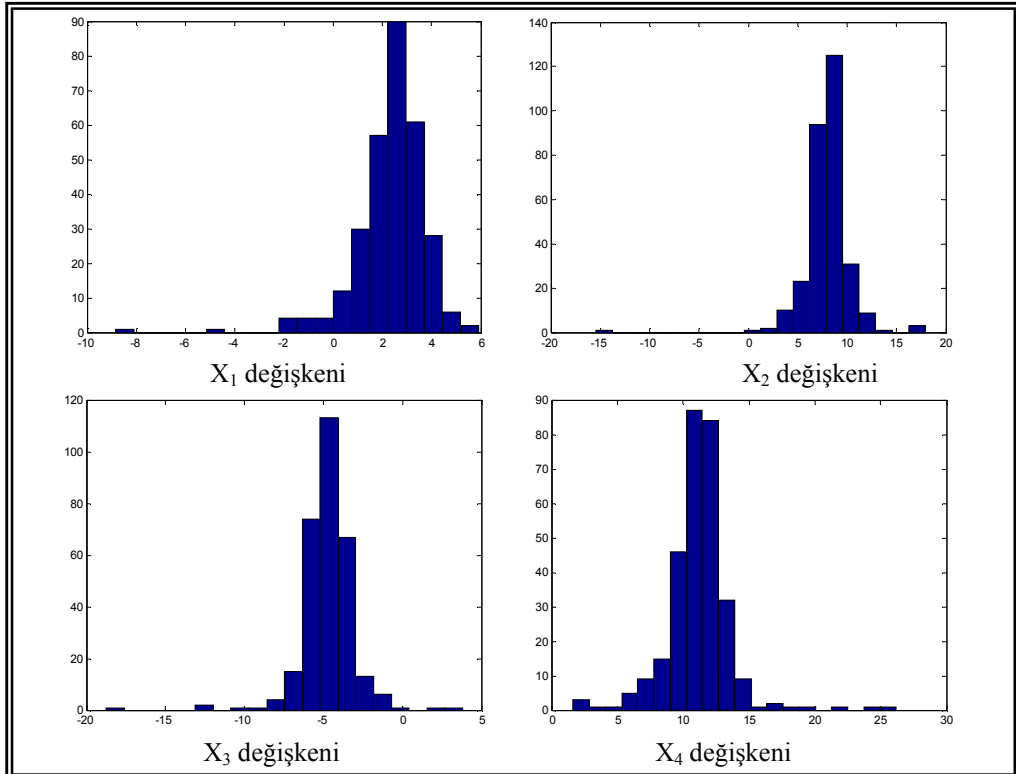


## EK5: Durum3 İin Duyarlılık Deęerlerine İliřkin Grafikler

### a) Duyarlılık deęerinin belirlenmesinde kullanılan ortalamaların daęılımları

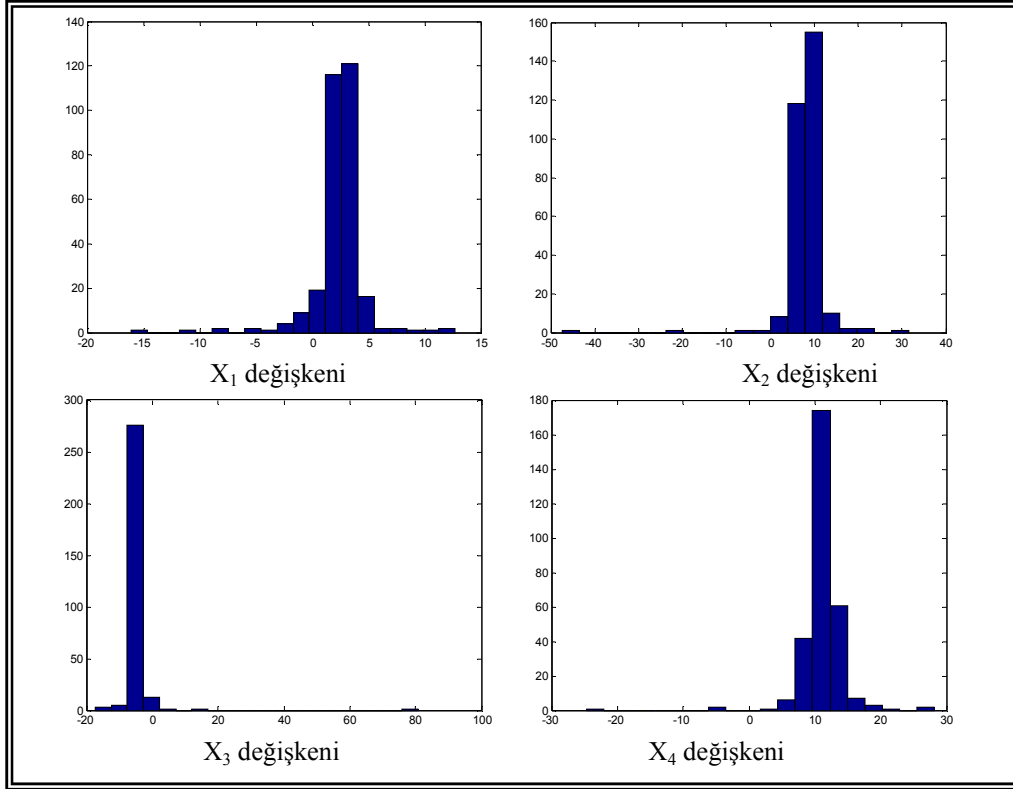


### b) Duyarlılık deęerinin belirlenmesinde kullanılan medyanların daęılımları

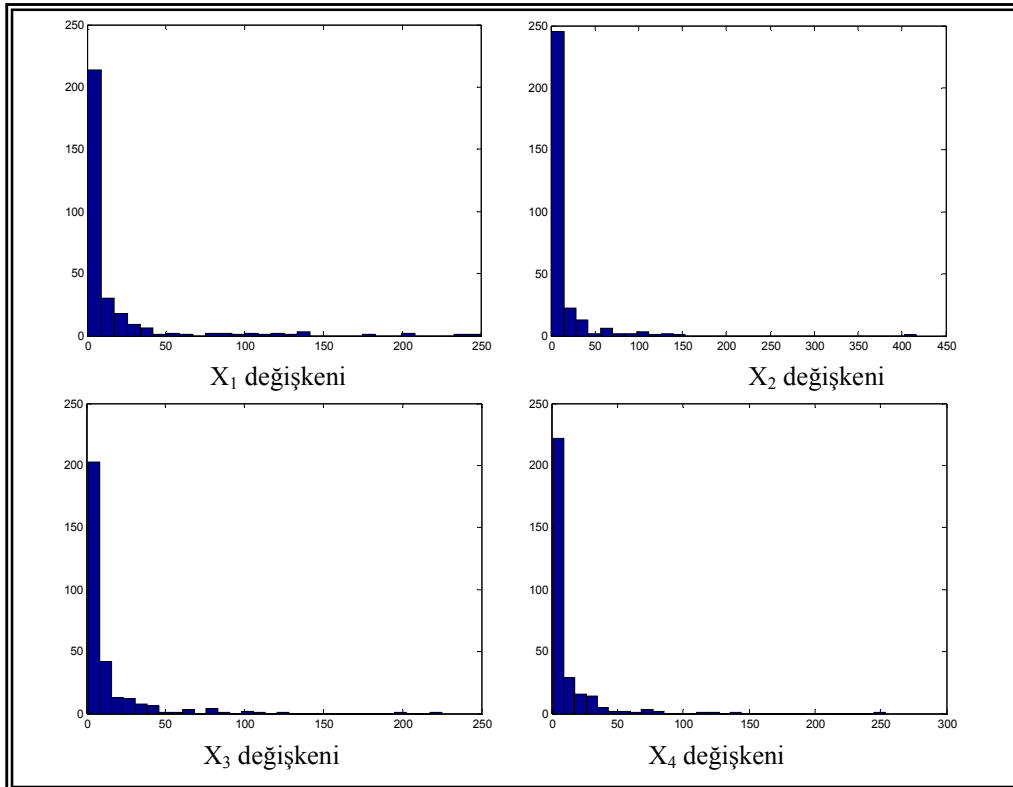




**c) Duyarlılık deęerinin belirlenmesinde kullanılan trimmean(%15) daęılımları**

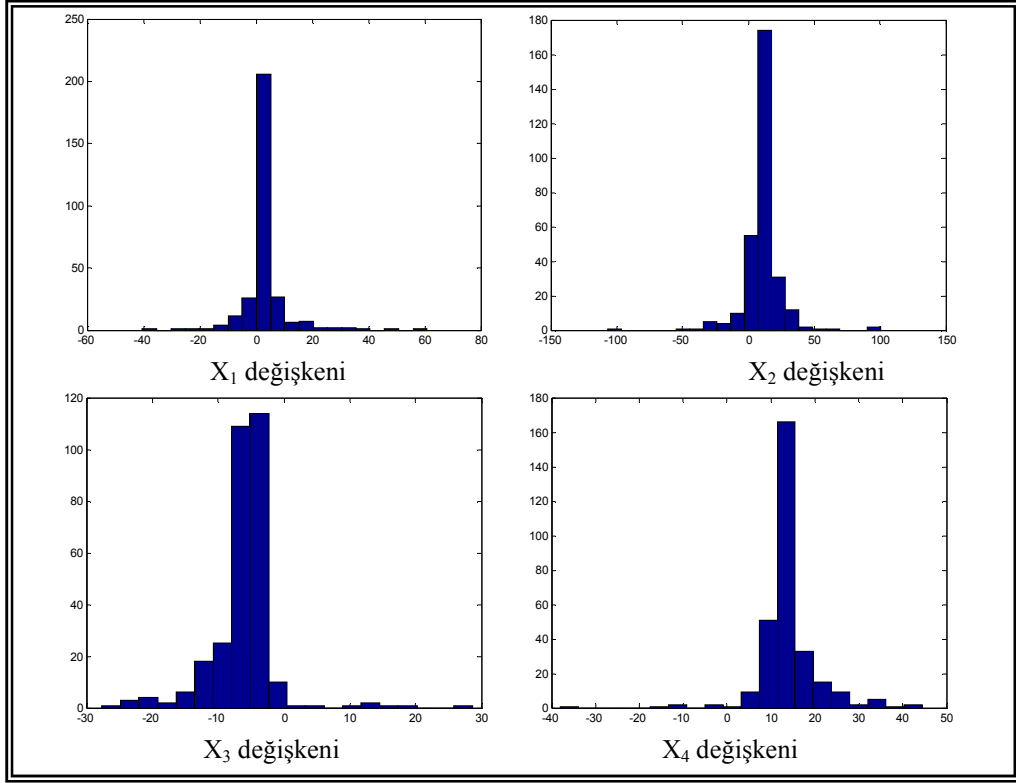


**d) Duyarlılık deęerlerinin standart sapma daęılımları**

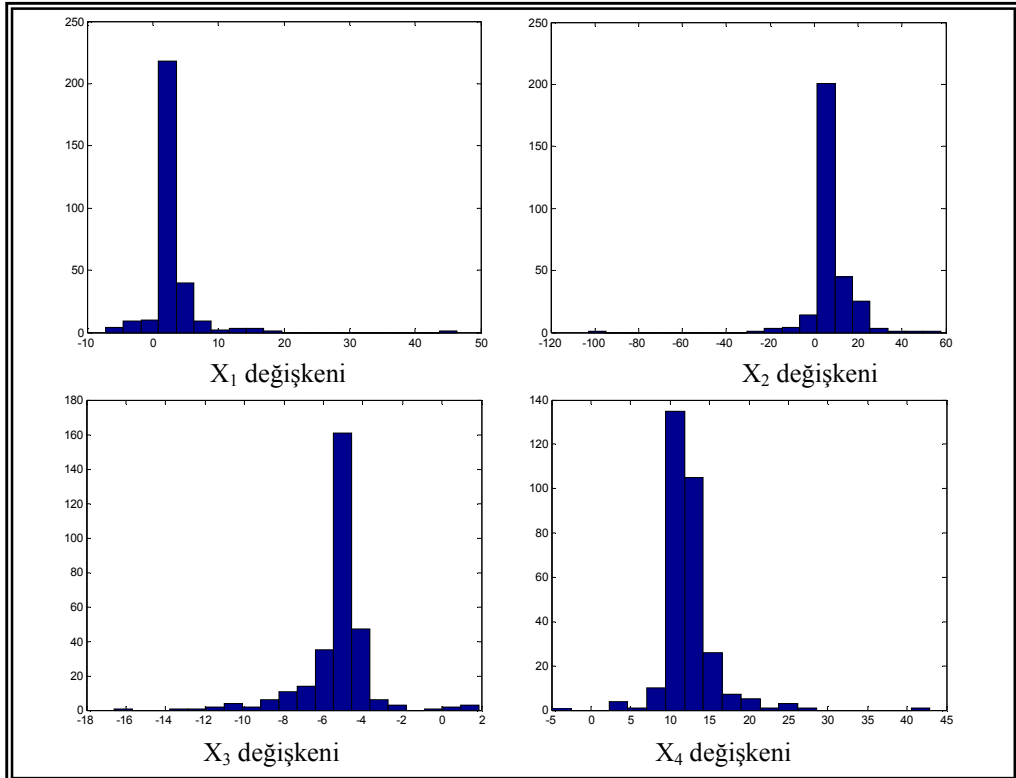


## EK6: Durum4 İin Duyarlılık Deęerlerine İlişkin Grafikler

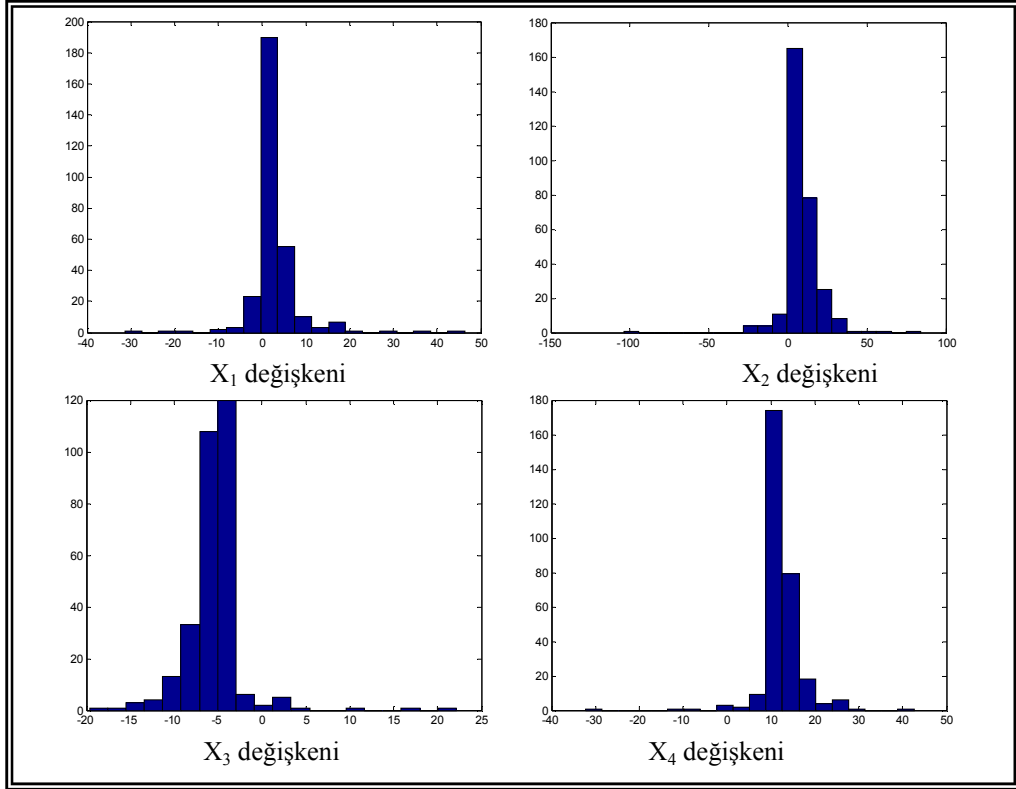
### a) Duyarlılık deęerinin belirlenmesinde kullanılan ortalamaların daęılımları



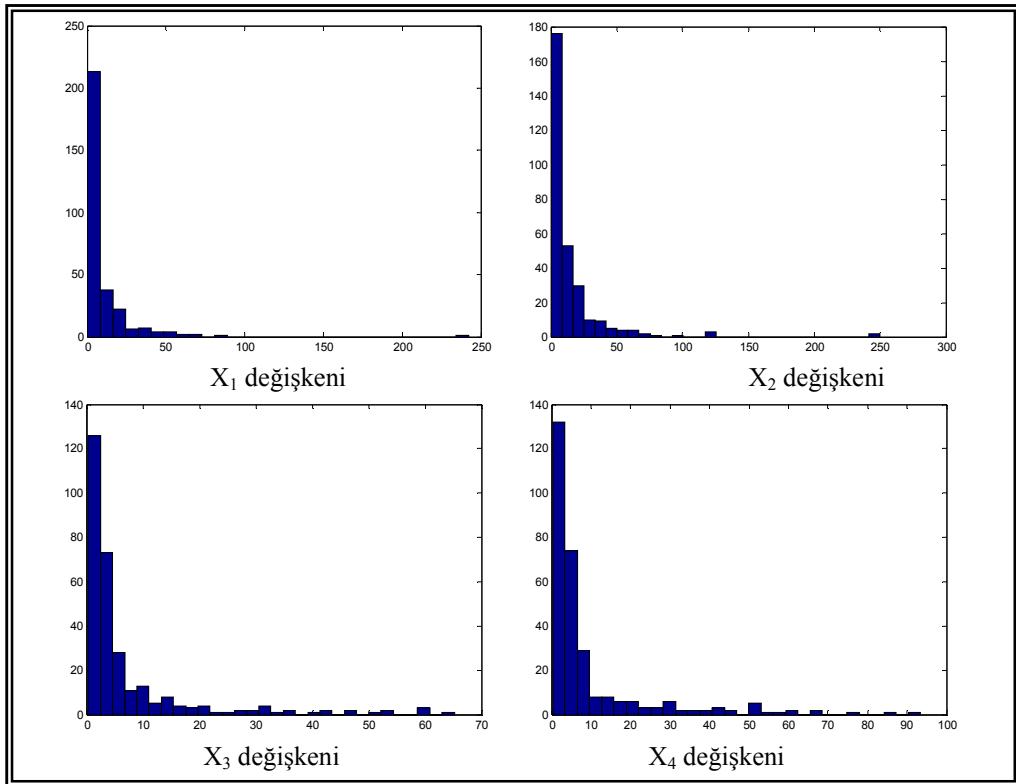
### b) Duyarlılık deęerinin belirlenmesinde kullanılan medyanların daęılımları



**c) Duyarlılık değerin belirlenmesinde kullanılan trimmean(%15) dağılımları**



**d) Duyarlılık değlerinin standart sapma dağılımları**



## EK7: Duyarlılık Değerlerini ve Bunlara İlişkin Tanımlayıcı İstatistik Değerlerini Hesaplayan Matlab Kodları

```
function [stdsapma, ort,med,tmean]=duyarlilik(A,B,C,D)
echo off all
[k2,k]=size(C);
[m2,m]=size(B);
[n2,n]=size(A);

for i=1:k
for j=1:n
for l=1:m
duy{i,j,l}=sum(A(:,j).*B(:,l).*C(:,i).*D(:,i));
end
end
end

for j=1:n
for l=1:m
for i=1:30
dd=cell2mat(duy(:,j,l));
stdsapma(j,l)=std(dd);
ort(j,l)=mean(dd);
med(j,l)=median(dd);
tmean(j,l)=trimmean(dd,15);
%hist(dd)
%pause
end
%hist(dd,20)
%pause
end
end
end
```

## EK8: Yeni Ağırlık Değerlerine Bağlı Olarak Duyarlılıkları ve Tanımlayıcı İstatistik Değerleri Hesaplayan Matlab Kodları

```
function [ortalama, medyan, standart, trimmean, ortort, medort,
tmeanort, medmed]=simduy(p,t,epoch,b)
echo off all
for i=1:b
i
net=newff(minmax(p),[9 1],{'logsig' 'purelin'},'trainlm');
net.trainParam.epochs=epoch; net.trainParam.goal=1e-5;
net.trainParam.show=250;
[net,tr]=train(net,p,t);
iw=net.IW{1,1};
s1=iw*p;
lwt=net.LW{2,1}';
Nc=logsig(s1);
bireksiN1=ones(9,350)-Nc;
[stdsapma,ort,med,tmean]=duyarlilik(lwt,iw,Nc,bireksiN1)
ortalama(i,:)=ort;
medyan(i,:)=med;
standart(i,:)=stdsapma;
trimmean(i,:)=tmean;
end
```

```
ortort=mean(ortalama);  
medort=mean(medyan);  
medmed=median(medyan);  
tmeanort=mean(trimmean);
```