



**T. C.**

**ULUDAĞ ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
EKONOMETRİ ANABİLİM DALI  
YÖNEYLEM BİLİM DALI**

# **YAPAY SİNİR AĞ MODELLERİ VE BİR TEKSTİL FİRMASINDA UYGULAMA**

**( DOKTORA TEZİ )**

**Vesile Sinem ARIKAN KARGI**

**BURSA- 2013**



**T. C.**

**ULUDAĞ ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
EKONOMETRİ ANABİLİM DALI  
YÖNEYLEM BİLİM DALI**

# **YAPAY SİNİR AĞ MODELLERİ VE BİR TEKSTİL FİRMASINDA UYGULAMA**

**( DOKTORA TEZİ )**

**Vesile Sinem ARIKAN KARGI**

**Danışman  
Prof. Dr. Ahmet ÖZTÜRK**

**BURSA- 2013**


TEZ ONAY SAYFASI

T. C.  
ULUDAĞ ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE


Ekonometri Anabilim Dalı, Yöneylem Bilim Dalı'nda 710817003 numaralı Vesile Sinem ARIKAN KARGI'nın hazırladığı "YAPAY SINIR AĞ MODELLERİ VE BİR TEKSTİL FİRMASINDA UYGULAMA" konulu Doktora Tezi Çalışması ile ilgili tez savunma sınavı, 10/05/2013 günü 13:00 – 15:00 saatleri arasında yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin Basarılı (başarılı/başarısız) olduğuna oybirliği (oybirliği/oy çokluğu) ile karar verilmiştir.

  
Üye (Tez Danışmanı ve  
Sınav Komisyonu Başkanı)  
Prof. Dr. Ahmet ÖZTÜRK  
Uludağ Üniversitesi

  
Üye  
Prof. Dr. Necmi GÜRSAKAL  
Uludağ Üniversitesi

  
Üye  
Doç. Dr. İrfan ERTUĞRUL  
Pamukkale Üniversitesi

Üye  
Prof. Dr. İbrahim DOĞAN  
Marmara Üniversitesi

  
Üye  
Prof. Dr. Ayşe OĞUZLAR  
Uludağ Üniversitesi

10/05/2013

## ÖZET

Yazar Adı ve Soyadı : Vesile Sinem ARIKAN KARGI  
Üniversite : Uludağ Üniversitesi  
Enstitü : Sosyal Bilimler Enstitüsü  
Anabilim Dalı : Ekonometri  
Bilim Dalı : Yöneylem  
Tezin Niteliği : Doktora Tezi  
Sayfa Sayısı : XII + 135  
Mezuniyet Tarihi : ... /... / 2013  
Tez Danışman(lar)ı : Prof. Dr. Ahmet ÖZTÜRK

### YAPAY SİNİR AĞ MODELLERİ VE BİR TEKSTİL FİRMASINDA UYGULAMA

Çalışmamızın temel amacı firmanın kumaş üretim sürecinde meydana gelen atkı hatalarını tahmin etmektir. Çalışmada tahmin için çoklu doğrusal regresyon(MLR) modeli ile yapay sinir ağ modellerinden çok katmanlı algılayıcı(MLP) modeli ve radyal tabanlı fonksiyon ağ (RBFN) model teknikleri uygulanmıştır. Diğer amacımız ise üç tekniğin tahmin sonuçlarına bakılarak hangi tekniğin atkı hata sayısını tahmin etmede daha başarılı olduğunu belirlemektir.

Çalışmamız teorik ve uygulama olmak üzere iki ana kısım içermektedir. İlk üç bölümde çoklu doğrusal regresyon modeli, yapay sinir ağ modelleri ve çok katmanlı algılayıcı model ile radyal tabanlı fonksiyon ağ modeli açıklanmıştır. İkinci kısım ise Bursa'da kumaş üreten bir tekstil firmasının kumaşlardaki atkı sayısında oluşan hatayı en aza indirmeyi sağlayacak çoklu doğrusal regresyon modeli, çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağ modellerinin uygulanmasına ilişkindir.

Çalışmada çoklu doğrusal regresyon model çözümü için SPSS 13 paket programı, çok katmanlı algılayıcı model ve radyal tabanlı fonksiyon ağ model çözümleri için de Matlab R2010b programı kullanılmıştır. Sonra modellerin tahmin sonuçları kıyaslanarak firma için en uygun model belirlenmiştir.

#### Anahtar Sözcükler:

Çoklu Doğrusal  
Regresyon Modeli

Yapay Sinir Ağları

Çok Katmanlı  
Algılayıcı Model

Radyal Tabanlı  
Fonksiyon Ağ  
Modeli

Kumaş Atkı Hatası

## ABSTRACT

Name and Surname : Vesile Sinem ARIKAN KARGI  
University : Uludağ Üniversitesi  
Institution : Sosyal Bilimler Enstitüsü  
Field : Ekonometri  
Branch : Yöneylem  
Degree Awarded : Doktora Tezi  
Page Number : XII + 135  
Degree Date : ... /... / 2013  
Supervisor (s) : Prof. Dr. Ahmet ÖZTÜRK

### ARTIFICIAL NEURAL NETWORK MODELS AND AN APPLICATION AT A TEXTILE FIRM

The main purpose of our study is to predict the weft defects which is occurred in the firm's fabric production process. In this study multiple linear regression(MLR) model and multilayer perceptron(MLP) model and radial basis function network(RBFN) model of artificial neural networks models are applied to forecast the weft defects. Our another aim by looking at the three tools forecasting results to predict the weft defects which Tecniques are more successful.

Our study is involved as a theoretical and practical two main divisions. In the first three section multiple linear regression model, artificial neural network model and multilayer perceptron model and radial basis function network model is explained. The second section is related to application of multiple linear regression model, multilayer perceptron model and radial basis function network model to minimize weft defects in fabric production of textile firm in Bursa.

In this study SPSS 13 software package program is used for multiple linear regression model solution and Matlab R2010b is used for multilayer perceptron model and radial basis function network model solutions. After by comparing the models estimation results we determined the optimal model for the firm.

#### Keywords:

Multiple Linear Regression Model      Artificial Neural Network Model      Multilayer Perceptron Model      Radial Basis Function Network Model  
Fabric's Weft Defect

## ÖNSÖZ

Çalışmamız yöneylem araştırmasının tahmin tekniklerinden olan çoklu doğrusal regresyon ve yapay sinir ağ modellerinin Bursa'daki bir tekstil firmasının kumaş üretimindeki atkı hatasını belirlemeye yöneliktir. Tez çalışmamız işletme yönetimine ve bu konuda çalışanlara yardımcı olacağı düşünülerek hazırlanmıştır.

Çalışmaya ilişkin doktora tezimin ortaya çıkarılmasında, her zaman yardım ve desteğini esirgemeyen tez danışmanın Sayın Prof. Dr. Ahmet ÖZTÜRK'e teşekkürü bir borç bilirim. Çalışmalarım sırasında beni teşvik eden, anlayış ve desteği ile yanımda olan eşim Sedat KARGI'ya ve bugünlere gelmemde emek ve desteklerini esirgemeyen sevgili annem Semra Mina ARIKAN 'a ve rahmetli babam Fethullah ARIKAN 'a sonsuz teşekkürlerimi sunarım. Ayrıca çalışmamızda kullandığımız verileri sağlayan firma yönetimine de teşekkür ederim.

**Vesile Sinem ARIKAN KARGI**

**Bursa**

**10.05.2013**

## İÇİNDEKİLER

TEZ ONAY SAYFASI .....	ii
ÖZET .....	iii
ABSTRACT .....	iv
ÖNSÖZ .....	v
İÇİNDEKİLER.....	vi
KISALTMALAR.....	ix
TABLolar LİSTESİ .....	x
ŞEKİLLER LİSTESİ .....	xi
GRAFİKLER LİSTESİ .....	xii
GİRİŞ.....	1

## BİRİNCİ BÖLÜM

### ÇOKLU DOĞRUSAL REGRESYON MODELİ

1. 1. Basit Doğrusal Regresyon Modeli.....	8
1. 2. Çoklu Doğrusal Regresyon Modeli .....	9
1. 2. 1. Parametrelerin Tahmini .....	10
1. 2. 2. Tahminlerin Standart Hatası .....	14
1. 2. 3. Çoklu Determinasyon(Belirlilik) Katsayısı .....	15
1. 2. 4. Regresyon Modelinin Genel Anlamlılığının(Uyum İyiliğinin) Sınanması .....	16
1. 2. 5. Regresyon Katsayılarının Önem Testi.....	18

## İKİNCİ BÖLÜM

### YAPAY SİNİR AĞLARININ KURAMSAL YAPISI

2. 1. Yapay Sinir Ağlarının Tanımı ve Özellikleri .....	19
---	----

2.2. Yapay Sinir Ağlarının Tarihçesi.....	22
2.3. Yapay Sinir Ağlarının Üstünlükleri ve Sakıncaları.....	25
2.4. Yapay Sinir Ağlarının Uygulama Alanları.....	26
2.5. Biyolojik Sinir Sistemleri.....	28
2.6. Yapay Sinir Hücresi.....	30
2.7. Yapay Sinir Ağının Yapısı.....	35
2.8. Yapay Sinir Ağlarının Çalışması.....	36
2.9. Yapay Sinir Ağlarında Eğitim, Öğrenme ve Test Etme.....	37
2.10. Tek Katmanlı Algılayıcılar.....	39
2.10.1. Basit Algılayıcı (Perceptron) Model.....	41
2.10.2. Adaline / Madaline Modeli.....	47
2.11. Yapay Sinir Ağlarının Sınıflandırılması.....	50
2.11.1. Yapay Sinir Ağların Mimari Yapılarına Göre Sınıflandırılması.....	51
2.11.2. Yapay Sinir Ağların Öğrenme Yapılarına Göre Sınıflandırılması.....	53
2.12. Temel Öğrenme Kuralları.....	55
2.13. Yapay Sinir Ağı Tasarımı.....	57

## ÜÇÜNCÜ BÖLÜM

### ÇOK KATMANLI ALGILAYICI MODEL VE RADYAL TABANLI FONKSİYON AĞ MODELİ

3.1. Çok Katmanlı Algılayıcı Model.....	65
3.1.1. Çok Katmanlı Algılayıcı Modelin Tanıtımı.....	65
3.1.2. Çok Katmanlı Algılayıcılarda Katman Yapısı.....	66
3.1.3. Geri Yayılım Algoritması.....	68
3.1.4. XOR Probleminin Çözümü.....	74
3.2. Radyal Tabanlı Fonksiyon Ağ Modeli.....	78
3.2.1. Radyal Tabanlı Fonksiyon Ağları.....	79
3.2.2. Radyal Tabanlı Fonksiyon Ağlarında Katman Yapısı.....	79
3.2.3. Öğrenme Stratejileri.....	82
3.3. RBF Ağları ile MLP Ağlarının Karşılaştırılması.....	84



## DÖRDÜNCÜ BÖLÜM

### BİR TEKSTİL İŞLETMESİNDE ÇOKLU DOĞRUSAL REGRESYON VE YAPAY SİNİR AĞ MODELLERİNİN UYGULANMASI

4.1. Firma ile İlgili Genel Bilgiler ve İş Akışı.....	86
4.2. Verilerin Toplanması.....	88
4.3. Çoklu Doğrusal Regresyon Modelin Firmaya Uygulanması .....	88
4.4. Yapay Sinir Ağ Modellerinin Firmaya Uygulanması.....	94
4.4.1. Çok Katmanlı Algılayıcı Model .....	95
4.4.2. Radyal Tabanlı Fonksiyon Ağ Modeli .....	103
<b>SONUÇ .....</b>	<b>114</b>
<b>KAYNAKLAR.....</b>	<b>117</b>
<b>EKLER .....</b>	<b>121</b>
<b>ÖZGEÇMİŞ .....</b>	<b>134</b>

## KISALTMALAR

<b>Kısaltma</b>	<b>Bibliyografik Bilgi</b>
MLR	Çoklu doğrusal regresyon
YSA	Yapay sinir ağları
MLP	Çok katmanlı algılayıcı
RBFN	Radyal tabanlı fonksiyon ağ
MSE	Hata kareler ortalaması
RMSE	Hata kareler ortalamasının karekökü
MAE	Ortalama mutlak hata
XOR	Doğrusal olmaya problem
ADALINE	Uyarlanan doğrusal eleman
MADALINE	Birden fazla uyarlanan doğrusal eleman
ART	Uyarlamalı rezonans teorisi
SOM	Özörgütleyici haritalar
LVQ	Doğrusal vektör parçalama
Perceptron	Basit algılayıcı model
a.g.e	Adı geçen eser

## TABLULAR LİSTESİ

	<b>Sayfa No</b>
Tablo 1.1 Çoklu doğrusal regresyonda varyans analizi.....	17
Tablo 2.1 Biyolojik sinir ağı ve yapay sinir ağının karşılaştırılması.....	30
Tablo 2.2 Kullanım amaçlarına göre yapay sinir topolojileri.....	58
Tablo 2.3 Öğrenme algoritmaları ve uygulandıkları alanlar.....	60
Tablo 3.1 XOR problemi için girdi ve çıktılar.....	74
Tablo 3.2 XOR problemini öğrendikten sonra ağın ürettiği çözümler ve hata oranları.....	78
Tablo 4.1 Katsayılar tablosu I.....	89
Tablo 4.2 Regresyon modelinin özet tablosu.....	90
Tablo 4.3 Varyans analizi tablosu.....	91
Tablo 4.4 Katsayılar tablosu II.....	91
Tablo 4.5 En uygun MLP modelini belirlemek için yapılan denemeler.....	96
Tablo 4.6 Giriş nöronları ile gizli katman nöronları arasındaki ağırlık değerleri.....	99
Tablo 4.7 Gizli katmanındaki nöronların eşik değerleri.....	99
Tablo 4.8 Gizli katmandaki nöronlar ile çıkış katmanındaki nöron arasındaki ağırlık değerleri.....	99
Tablo 4.9 Çıktı katmanındaki nöronun eşik değeri.....	100
Tablo 4.10 MLP ağının çıktıları ile gerçek veriler.....	100
Tablo 4.11 En uygun RBFN modelini belirlemek için yapılan denemeler.....	105
Tablo 4.12 Giriş nöronları ile gizli katman nöronları arasındaki ağırlık değerleri.....	106
Tablo 4.13 Gizli katmanındaki nöronların eşik değerleri.....	107
Tablo 4.14 Gizli katmandaki nöronlar ile çıkış katmanındaki nöron arasındaki ağırlık değerleri.....	109
Tablo 4.15 Çıktı katmanındaki nöronun eşik değeri.....	110
Tablo 4.16 RBF ağının çıktıları ile gerçek veriler.....	110

## ŞEKİLLER LİSTESİ

	Sayfa No
Şekil 1.1 En küçük karelerin geometrik yorumu.....	11
Şekil 2.1 Biyolojik sinir sisteminin blok gösterimi.....	29
Şekil 2.2 Biyolojik nöronun yapısı.....	29
Şekil 2.3 Yapay sinir hücresinin fonksiyonel yapısı.....	30
Şekil 2.4 En çok kullanılan transfer fonksiyonları.....	34
Şekil 2.5 Bir yapay sinir ağ örneği.....	36
Şekil 2.6 Bir yapay sinir ağı girdi-çıkı ilişkisi.....	37
Şekil 2.7 İki girdi ve bir çıkıtdan oluşan en basit tek katmanlı yapay sinir ağ modeli.....	39
Şekil 2.8 Ağırlıkların ve sınıf ayracı olan doğrunun geometrik gösterimi.....	40
Şekil 2.9 Bir basit algılayıcı yapısı.....	42
Şekil 2.10 VE probleminin doğrusal bölünebilirliği.....	44
Şekil 2.11 VEYA probleminin doğrusal bölünebilirliği.....	45
Şekil 2.12 VE DEĞİL probleminin doğrusal bölünebilirliği.....	46
Şekil 2.13 XOR probleminin doğrusal olarak bölünemezliği.....	47
Şekil 2.14 Bir ADALINE model yapısı.....	48
Şekil 2.15 İki ADALINE ağından oluşan MADALINE ağı.....	50
Şekil 2.16 Danışmanlı öğrenme yapısı.....	54
Şekil 2.17 Danışmansız öğrenme yapısı.....	54
Şekil 2.18 Takviyeli öğrenme yapısı.....	55
Şekil 2.19 Ağırlık uzayındaki düşme.....	62
Şekil 3.1 Çok katmanlı algılayıcı(MLP) yapay sinir ağının yapısı.....	66
Şekil 3.2 İterasyon sayısı ile ağ hatasının ilişkisi.....	73
Şekil 3.3 Radyal tabanlı fonksiyon(RBFN) yapay sinir ağının yapısı.....	80
Şekil 4.1 Firmanın iş akış şeması.....	87

## GRAFİKLER LİSTESİ

Sayfa No

Grafik 4.1 Eğitim, doğrulama ve test kümelerine ilişkin hata Performansları.....	98
Grafik 4.2 Eğitim verileri için gerçek ile tahmini atkı hata sayısı.....	102
Grafik 4.3 Test verileri için gerçek ile tahmini atkı hata sayısı.....	102
Grafik 4.4 Eğitim verileri için gerçek ile tahmini atkı hata sayısı.....	112
Grafik 4.5 Test verileri için gerçek ile tahmini atkı hata sayısı.....	112

## GİRİŞ

Talep yönlü küresel pazarlarda yaşanan şiddetli rekabet şirketleri en az hatalı ürün üretmeye zorlamaktadır. En az hatalı üretimi sağlayan iş yöntemleri olarak günümüzde yalın üretim, altı sigma, benchmarking, toplam kalite yönetimi ve tam zamanında envanter sistemi uygulanmaktadır. Küresel pazarların büyümesi ve yaşanan yoğun rekabet, yöneylem araştırmasına olan ihtiyacı bu iş yöntemlerinde olduğu gibi yönetimlerin rasyonel karar almasında önemli kılmaktadır.

Yöneylem araştırmasının kullandığı modellerden çok değişkenli regresyon ve yapay sinir ağları da işletmelerin üretiminde minimum hata düzeyini belirlemede yardımcı olabilmektedir. Günümüz problemleri karmaşık ve çözümü de bilimsel karar vermeyi gerektirir. Ayrıca karşılaşılan problemlerin çokluğu ve araştırmacıların problemlere farklı bakış açıları yapay sinir ağlarının iş hayatında uygulanmasının gerekli olduğunu ortaya çıkartmıştır. Dolayısıyla yapay sinir ağları 2000’li yıllarda iş hayatının çeşitli alanlarında uygulandığı görülmektedir.

Bursa’da ham kumaş dokuyan tekstil şirket yöneticileri ile yaptığımız görüşme sonucunda üretilen kumaşlarda oluşan atkı hata sayısında problem yaşadıklarını ve üretim öncesi bu hata sayısının en aza indirilebilmesini talep etmişlerdir. Bu problemin çözümü çalışmamızın temelini oluşturmuştur. İşletmenin problemi, sebep-sonuç ilişkisine dayalı bir tahmin problemidir. Sebep-sonuç ilişkisine dayanan tahmin problemlerinde kullanılan en yaygın tahmin aracı çoklu doğrusal regresyon modelidir. Ancak günlük yaşantıda karşılaşılan tahmin problemlerinin birçoğu doğrusal ilişki özelliği göstermeyebilir ve doğrusal olmayan ilişkinin tahmin edilebilmesi için veri yapısındaki fonksiyonel ilişkinin ön bilgisine ihtiyaç duyulur. Çoğu zaman bu ilişkiyi belirlemek güçtür. İşte bu güçlüğü üstesinden gelebilmek için yapay sinir ağ modellerine başvurulur. Çünkü sinir ağlarının doğrusal olmayan, gürültülü, eksik verilerin bulunması halinde tahmin yapabilme konusunda oldukça iyi sonuçlar verebilmektedir. Yapay sinir ağları iyi bir tahmin aracı

olmakla birlikte nedensellik hakkında bilgi vermez. Bu eksikliđi gidermek için de çoklu doğrusal regresyon modeli kullanılır.

Çalışmamızın temel amacı firmanın üretim sürecinde meydana gelen atkıdaki hata sayısını önceden belirleyebilmektir. Bunun için de çoklu doğrusal regresyon(MLR) modeli ile yapay sinir ađ modellerinden çok katmanlı algılayıcı(MLP) modeli ve radyal tabanlı fonksiyon ađ (RBFN) model tekniđinin uygulanmasının doğru olacađına karar verilmiştir. Diđer amacımız da söz ettiđimiz her üç tekniđin sonuçlarının tahmin etmedeki doğruluk oranına bakılarak hangi tekniđin atkı hata sayısını tahmin etmede daha başarılı olduđunu belirlemektir.

Teorik ve uygulama olarak iki ana kısımdan oluřan çalışmamızda, teorik kısım ilk üç bölümde uygulama kısmı ise dördüncü bölümde ele alınmıştır. řimdi kısaca bunları açıklayalım.

Çalışmamızın birinci bölümünde; basit doğrusal regresyon modeli, çoklu doğrusal regresyon modeli anlatılarak çoklu doğrusal regresyon modeline ilişkin parametre tahmini, tahminlerin standart hatası, çoklu determinasyon katsayısı, regresyon modelinin genel anlamlılıđının sınanması ve regresyon katsayısının önem testi açıklanmıştır.

İkinci bölümde ise yapay sinir ađ modelinin tanımı ve özellikleri, tarihçesi, bu ađların üstünlükleri ve sakıncaları, uygulama alanları anlatılmış, sonra biyolojik sinir sistemleri, yapay sinir hücresi, yapay sinir ađlarının yapısı, çalışması, eğitim öğrenme ve test edilmesi açıklanmıştır. Daha sonra tek katmanlı algılayıcılar hakkında bilgi verilmiştir. Yapay sinir ađları mimari ve öğrenme yapılarına göre sınıflandırılarak, temel öğrenme kuralları ve ađ tasarımı açıklanmıştır.

Üçüncü bölümünde ise yapay sinir ađ modellerinden çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ađlarına ilişkin bilgiler verilmiştir. Öncelikle çok katmanlı algılayıcı modeline ilişkin bilgiler verilmiş ve sonra çok katmanlı algılayıcıların katman yapısı anlatılarak geri yayılım algoritması açıklanarak yapay sinir ađlarının gelişmesinde katkısı olan XOR problemine ilişkin bir örnek problem çözülmüştür. Daha sonra radyal

tabanlı fonksiyon ağı anlatılarak bu ağların katman yapısı ve öğrenme stratejileri açıklanarak bu iki ağ yapısı karşılaştırılmıştır.

Çalışmamızın dördüncü bölümü, Bursa’da faaliyet gösteren bir tekstil firmasına ilişkin üretilen kumaşlardaki atkı sayısında oluşan hatayı en aza indirmeyi sağlayacak çoklu doğrusal regresyon modeli, çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağ modellerinin bu işleme uygulamasını içermektedir. Bunun için öncelikle firmaya ilişkin bilgiler toplanmıştır. Daha sonra modelimiz için firma ile yaptığımız görüşmeler sonucunda atkıda oluşan hatayı etkileyebilecek değişkenler belirlenerek bunlara ilişkin veriler toplanmıştır. Daha sonra ilk olarak çoklu doğrusal regresyon modeli ile tahmin işlemi yapılmış ve aynı zamanda atkıdaki hata sayısı üzerinde etkili olan değişkenler belirlenmiştir. Sonra da yapay sinir ağ modellerine ilişkin analizler yapılmıştır. İlk olarak çok katmanlı algılayıcı modeli ve daha sonra da radyal tabanlı fonksiyon ağ modeli kullanılarak modelin tahmini yapılmıştır. Sonra da ele aldığımız üç modele ilişkin performanslar kıyaslanarak firmanın kullanabileceği en uygun model belirlenmiştir.

Şimdi literatürde çoklu doğrusal regresyon ve yapay sinir ağ modellerinin tekstil alanında yapılmış olan çalışmalardan söz ederek çalışmamızın önemini ortaya koyalım.

Zhang ve diğerlerinin 2002 yılında yaptıkları “*Prediction on Tribological Properties of Short Fibre Composites using Artificial Neural Networks*” adlı çalışmalarında çok katmanlı ileri beslemeli ağ yapısını kullanarak kısa elyaf takviyeli polyamid 4.6 kompozitler için ölçülen veritabanına dayalı olan spesifik dayanma oranı ve sürtünme katsayısı tahmin edilmiştir<sup>1</sup>. 103 veriden oluşan modelin random olarak seçilen ve seksen sekizi eğitim için kullanılırken on beşi de test için kullanılmıştır. Burada spesifik dayanma oranı eğitim kümesindeki veriler için %99 tahmin yapılırken test kümesi verileri için %87 iyi tahmin ile sonuçlanmıştır. Gerçek veriler ile tahmin değerleri karşılaştırıldığında yapılan tahmin sonuçlarının gerçeğe dayalı olumlu sonuçlar verdiği görülmüştür. Dolayısıyla iyi eğitilmiş yapay sinir ağ belirli bir tribiyolojik uygulama ve

---

<sup>1</sup> Z.Zhang, K. Friedrich , K. Velten, **Prediction on Tribological Properties of Short Fibre Composites Using Artificial Neural Networks**, Wear Journal, Vol.252, 2002, p.668-675.



sistematik parametre çalışmaları için kompozit malzemelerin en iyi tasarım için çok yararlı olacağı bildirilmiştir.

Furferi ve Gelli'nin 2004'te "*Yarn Strength Prediction: A Practical Model Based on Artificial Neural Networks*" adlı çalışmalarında geri yayımlı yapay sinir ağ modeli ile iplik mukavemeti tahmin edilmeye çalışılmış ve daha sonra klasik yöntemlerden çoklu doğrusal regresyon modeli ile tahmin sonuçları kıyaslanmıştır<sup>2</sup>. Bu model iplik eğirme süreci başlamadan önce sıklıkla teknisyenler tarafından ölçülen fitil parametre dizilerinin iplik mukavemetinin güvenilir bir tahminini sağlamıştır. Geliştirilen model üç katmanlı olup 5-10-1 ağ yapısındadır. Girdi verileri; fitilin büküm ipliği, i'nci fitilden lifin ağırlıklı ortalama uzunluğu, i'nci fitilden lifin ağırlıklı ortalama inceliği, i'nci fitilden lifin ağırlıklı ortalama mukavemeti, çıktı da i'nci ipliğin mukavemetidir. 98 eğitim verisi ile eğitilmiş ve 50 yeni veri ile doğrulaması yapılmıştır. İplik mukavemeti tahmininde doğrulama kümesi kullanılarak bulunan ortalama hata değeri yüzde 4'ün altındadır ve 22 iterasyonda bu değere ulaşılmıştır.

Zeng ve diğerlerinin 2005'te "*Predicting the Tensile Properties of Air-Jet spun Yarns*" adlı çalışmalarında nümerik simülasyon ve yapay sinir ağ modeli yöntemlerini kullanarak hava ağızlı iplik eğirme makinesinde üretilen ipliklerin mukavemet özelliklerini tahmin etmeye yönelik bir çalışma yapmışlardır<sup>3</sup>. Çalışmada yapay sinir ağ modeli olarak çok katmanlı algılayıcı kullanılmıştır. 5-2-1 modeline sahip ağ yapısında giriş nöronları olarak; ilk püskürtme(nozzle) basıncı, ikinci püskürtme basıncı, dönüş hızı, arka makara kesme ile ilk püskürtme girişi arasındaki uzaklık ve ilk püskürtmede ağız deliği ile ağız girişi arasındaki uzaklık alınmıştır. Çıkış nöronu ise iplik mukavemetidir. Matlab 6.5 te yazılan programda öğrenme oranı 0,01 alınmış ve 4000 iterasyon sonucunda en iyi sonuçlara ulaşılmıştır. Çalışma sonucunda korelasyon katsayısı 0,98 dir. Bu da tahmin değerlerinin kabul edilir ölçüde iyi olduğunu göstermiştir.

---

<sup>2</sup> Rocco Furferi - Maurizio Gelli, "Yarn Strength Prediction: A Practical Model Based on Artificial Neural Networks", *Advances in Mechanical Engineering*, Vol. 2010, Article ID 640103, 2010, 11 pages.

<sup>3</sup> Y.Zeng-K.Wang-C.Yu, "Predicting the Tensile Properties of Air-Jet spun Yarns", *Textile Research Journal*, Vol. 74, No.8,2005, p. 689-694.

Beltran ve diğerlerinin 2006' da "*Predicting the pilling tendency of wool knits*" adlı çalışmalarında, lif, iplik ve kumaşın özelliklerinin boncuklanma eğilimi üzerine tahmine dayanmaktadır<sup>4</sup>. Çalışmada saf yünden yapılmış süprem ve ribana örme kumaşların lif, iplik ve kumaş özelliklerinin, boncuklanma(pilling) eğilimine olan etkisini belirlemek için bir yapay sinir ağı (YSA) modeli kurulmuştur. YSA modeli olarak çok katmanlı algılayıcı (MLP) kullanılmıştır. Model, 17 girdi nöronu (lif, top, iplik ve kumaş özelliklerinden) ile tek gizli katman ve bir çıkış nöronundan oluşmuştur. Çıkış nöronunda ve gizli katmanda sigmoid fonksiyonu kullanılmıştır. Geliştirilen bu model çoklu doğrusal regresyon modeli ile karşılaştırılmış ve çok katmanlı algılayıcı modelin performansının daha iyi olduğu sonucuna varılmıştır.

Oğulata ve diğerlerinin 2006' da "*The Prediction of Elongation and Recovery of Woven Bi-Stretch Fabric Using Artificial Neural Network and Linear Regression Models*" adlı çalışmalarında streç dokuma kumaşlar için geniş kullanıma sahip olan elastikiyet ve uzama özelliklerini, yapay sinir ağ ve çoklu doğrusal regresyon modeli kullanarak tahmin etmişlerdir<sup>5</sup>. Yapay sinir ağı modelinde yine çok katmanlı algılayıcı kullanılmıştır. Burada uzantı oranı, ölçü uzunluğu ve maksimum yük girdi katmanındaki nöronlar olarak belirlenmiş, tek gizli katman, atkı ve çözgü elastikiyeti ile atkı ve çözgü uzama olarak iki nörona sahip çıkış katmanı söz konusudur. Sonuçlar incelendiğinde her iki modelin tahmin gücü yüksek olmakla birlikte yapay sinir ağı ile yapılan tahmin regresyon modeline göre daha başarılı olmuştur.

Gharehaghaji ve diğerlerinin 2007' de "*Analysis of Two Methodologies for Predicting the Tensile Properties of Cotton-Covered Nylon Core Yarns*" adlı çalışmalarında çoklu doğrusal regresyon modelleri ile yapay sinir ağ modellerini kullanarak naylon özlü pamuk kaplamalı ipliklerin mukavemet özelliklerini

---

<sup>4</sup> Rafael Beltran-Lijing Wang-Xungai Wang, "Predicting the Pilling Tendency of Wool Knits", The Journal of the Textile Institute, Vol. 97, No.2, 2006, p.129-136.

<sup>5</sup>S. Noyan Oğulata ve diğerleri, "The Prediction of Elongation and Recovery of Woven Bi-Stretch Fabric Using Artificial Neural Network and Linear Regression Models," Fibres and Textiles in Eastern Europe, Vol.14, No.2, 2006,p.46-49.

incelemişlerdir<sup>6</sup>. Bu çalışmada da çok katmanlı algılayıcı kullanılmıştır. Geliştirilen modellerde test verilerinin tahmini korelasyon katsayısı ( R ) ve hata kareler ortalaması (MSE) ile doğrularak değerlendirilmiştir. Sonuçlar göstermiştir ki yapay sinir ağları çoklu doğrusal regresyon ile kıyaslandığında yapay sinir ağları daha iyi sonuç verdiği sonucuna varılmıştır. Tahmini yapılan bu iki modelin kopma mukavemeti ve kopma uzaması için hata kareler ortalaması sırasıyla 0,365 ve 0,119 dur.

Guruprasad ve Behera'nın 2010 yılında yaptıkları “ *Prediction of Bending of Woven Fabrics By Soft Computing*” adlı çalışmalarında pamuklu dokuma kumaşlarının eğilme özelliklerinin tahmin olasılıklarını yapay sinir ağları ve genetik algoritma yöntemleri ile araştırmışlardır<sup>7</sup>. İlk olarak ileri beslemeli sinir ağ modeli kurulmuştur. İkinci modelde ise hibrit bir öğrenme stratejisi benimsenmiştir. Her iki modelin tahmin performansları bakımından karşılaştırıldığında hibrit model yaklaşımı geri yayılım sinir ağ modeline göre daha iyi sonuç göstermiştir.

Bahadır ve diğerlerinin 2012’de yaptıkları “*An Artificial Neural Network Model for Prediction of Bursting Strength of Knitted Fabrics*” adlı çalışmalarında geri yayımlı yapay sinir ağ modelini kullanarak örme kumaşlarının patlama mukavemeti tahmin edilmiştir<sup>8</sup>. Çalışmanın girdi katmanında üç nöron vardır ve bunlar iplik numaraları, elastrometrik iplik numarası ve elastrometrik ipliğin bileşim oranıdır. Çıktı katmanında ise bir nöron vardır ve örme kumaşların patlama mukavemetidir Gizli katman ise bir tanedir ve yapılan denemeler sonucunda gizli katmandaki nöron sayısı dokuz nöron olduğunda en iyi ağ modeline ulaşılmıştır. Dolayısıyla en iyi model yapısı 3-9-1 dir. Örneklem kümesi 60 tane veriden oluşmaktadır ve bunun 48 tanesi eğitim kümesi 12 tanesi ise test kümesi için ayrılmıştır. Eğitim işlemi için 1000 iterasyon ve her biri beş kez çalıştırılmıştır. Öğrenme için transfer fonksiyonu tanjant hiperbolik fonksiyon kullanılmıştır. Eğitim işlemi 23 iterasyonda tamamlanmış ve hata kareler ortalaması (MSE) 0,0057 değerine

---

<sup>6</sup> Ali Akbar Gharehaghaji-Mohsen Shanbeh-Maziar Palhang, “Analysis of Two Methodologies for Predicting the Tensile Properties of Cotton-Covered Nylon Core Yarns”, Textile Research Journal, Vol. 77, 2007, p.565-571.

<sup>7</sup> R.Guruprasad - B.K. Behera, Prediction of Bending of Woven Fabrics By Soft Computing, 7<sup>th</sup> International Conference TEXSCI, Liberec (Çek Cumhuriyeti), 6-8 Eylül 2010.

<sup>8</sup> M. Çağatay Bahadır-Senem K. Bahadır- Fatma Kalaoğlu, “An Artificial Neural Network Model for Prediction of Bursting Strength of Knitted Fabrics”, International Conference on Machine Learning and Computer Science, Puket(Tayland), 11-12 Ağustos 2012.

ulaşılmıştır. Daha sonra test verileri ile ağın performans tahminine bakılmış ve bu korelasyon oranı 0,97 MSE değeri de 0,042 bulunmuştur. Çalışmanın sonucunda tek gizli katmanda dokuz nöronlu yapay sinir ağ modeli ile düşük hata oranında örme kumaşların patlama mukavemeti davranışının tahmin edilebileceğini ortaya koymuşlardır.

## BİRİNCİ BÖLÜM

### ÇOKLU DOĞRUSAL REGRESYON MODELİ

#### 1. 1. Basit Doğrusal Regresyon Modeli

Değişkenler arasındaki ilişkiyi belirlemek bilimin ilgilendiği konulardan biridir. Gerek günlük hayatta gerekse bilimsel araştırmalarda karşılaşılan sorunların çoğu, iki veya daha çok değişkenler arasında bir ilişkinin olup olmadığının belirlenmesi ile ilgilidir.

Regresyon terimi 19. yüzyılda İngiliz istatistikçi Francis Galton tarafından bir biyolojik inceleme için ortaya konulmuştur. Regresyon analizi, bağımlı değişken ile bağımsız değişken/değişkenler arasındaki ilişkiyi incelemeye yönelik bir yöntemdir<sup>9</sup>. Genellikle bağımsız değişkenler X, bağımlı değişkenler ise Y ile gösterilir.

*Regresyon modeli*; bağımsız değişkenlerin sayısına, fonksiyon özelliğine ve veri kaynağına göre gruplandırılır. Bağımsız değişkenlerin sayısına göre gruplandırıldığında basit regresyon analizi ve çoklu regresyon söz konusudur. Fonksiyon özelliğine göre; gruplandırıldığında doğrusal regresyon analizi ve doğrusal olmayan regresyon analizini içerir. Veri kaynağına göre gruplandırıldığında ise ana kütle verileri ile regresyon analizi; örneklem verileri ile regresyon analizi ve zaman serilerinde regresyon analizi söz konusudur.

Basit doğrusal regresyon modeli, y bağımlı değişkeninin tek bir bağımsız değişken x ile arasındaki ilişkinin doğrusal fonksiyonla ifade edilmesidir. Basit doğrusal regresyon modeli, aşağıdaki formül ile ifade edilir.

$$y = \beta_0 + \beta_1 x + u$$

Burada; x bağımsız(açıklayıcı) değişkeni, y bağımlı değişkeni ve u hata terimini(stokastik değişken) göstermektedir. u stokastik değişkeninin hangi değeri alacağı

---

<sup>9</sup> Kazım Özdamar, **Paket Programlar ile İstatistiksel Veri Analizi**, Kaan Kitabevi, Eskişehir, 2004.

önceden bilinmemektedir yani şansa bağlı olarak tesadüfî bir şekilde ortaya çıkan değişkendir. Hata terimi  $u$ , modele dahil edilemeyen değişkenleri içerir. Ayrıca model spesifikasyonunda yapılan hataların etkisi de hata terimine yansır.

Modelde yer alan parametreler  $\beta_0$ , doğrusal fonksiyon sabitidir ve bu parametreye kesme terimi denir.  $x$  sıfır olduğunda regresyon doğrusunun dikey eksen olan  $y$  ile kesiştiği noktayı gösterir.  $\beta_1$  ise doğrusal fonksiyonun eğimidir ve  $\beta_1$  parametresi regresyon analizinde bağımsız değişken  $x$ 'deki bir birimlik değişiminin bağımlı değişken  $y$ 'de ne kadar bir değişim yarattığını gösteren regresyon katsayısıdır.

## 1. 2. Çoklu Doğrusal Regresyon Modeli

Çoklu doğrusal regresyon bir bağımlı değişken ile iki ve daha fazla bağımsız değişken arasındaki doğrusal bağıntıyı inceleyen bir yöntemdir. Bağımlı değişken ( $y$ ) ile  $n$  adet bağımsız değişken( $x$ ) arasındaki ilişkiyi gösteren model genel olarak aşağıda verildiği şekilde ifade edilir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + u$$

Çoklu doğrusal regresyon modelinde  $\beta_0$  parametresi regresyon sabitidir. Diğer bir deyişle modelde yer alan bütün bağımsız değişkenlerin değerleri sabit tutulduğunda  $\beta_0$  bağımlı değişkenin ortalama değerini verir.  $\beta_1$  den  $\beta_n$  ye kadar olan  $x$  değişkenlerinin katsayıları ise kısmi regresyon katsayılarıdır. Örneğin  $\beta_n$  yorumlanırken bu parametreye ilişkin değişken  $x_n$  dışındaki diğer tüm bağımsız değişkenlerin etkilerinin sabit olduğu varsayılır ve bu varsayım altında  $x_n$ 'deki bir birim değişiminin bağımlı değişkenin ortalamasında ne kadar bir değişmeye yol açtığını gösterir. Buna marjinal etki katsayısı da denir.

$\beta_1.. \beta_n$  nin modelde yer alan işaretleri bağımsız değişken ile bağımlı değişken arasındaki bağıntının yönünü vermektedir. Buna karşın hangi bağımsız değişkenin etkisinin daha fazla olduğu konusunda  $\beta$ 'nin büyüklüğü yeterli değildir. Parametrelerin karşılaştırmalarda kullanılabilmesi için tüm bağımsız değişkenler aynı ölçü biriminde olmalıdır.

Çoklu Doğrusal Regresyon Model varsayımları aşağıda verilmiştir<sup>10</sup>.

- Model doğrusaldır.
- Hata payları birer tesadüfi değişken olup her  $x_i$  için  $E(u_i)=0$ 'dır.
- $u_i$ ' ler normal olarak dağılmaktadır.

$$u_i \sim N(0, \sigma_u^2)$$

- Homoskedastisite (sabit varyans) söz konusudur.

$$E(u_i^2) = \sigma_u^2$$

- Otokorelasyon (ardışık bağımlılık) bulunmamaktadır.

$$\text{cov}(u_i, u_j) = 0, \quad i \neq j$$

- $u_i$  hata payları bağımsız değişkenlerden bağımsızdır. Bir başka deyişle çoklu doğrusal bağlantı olmama durumudur.

$$\text{cov}(u_i, x_{ni}) = 0$$

Şimdi Çoklu Doğrusal Regresyon Analizin parametrelerinin nasıl tahmin edildiğini açıklamaya çalışalım.

### 1. 2. 1. Parametrelerin Tahmini

Çoklu doğrusal regresyon modelinde  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + u$ ,  $x_1, x_2$  ve  $y$  değişkenlerinin bütün değerleri bilindiğinde  $\beta_0, \beta_1, \beta_2$  parametrelerinin sayısal değerleri bulunabilir. Ancak tüm değerleri bilmek olanaksızdır. İşte bu nedenle, ana kütleyle ilişkin örnek gözlemler kullanılarak parametrelerin tahmini yapılır.

$x$  ile  $y$  arasındaki gerçek ilişki  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + u$  formülü ile

Gerçek regresyon doğrusu  $E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$  formülü ile

---

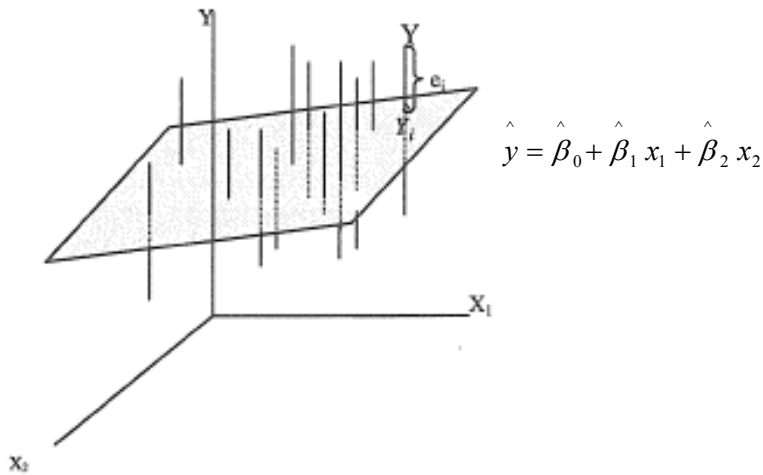
<sup>10</sup> Damodar N., Gujarati, **Temel Ekonometri**, çev. Ümit Şenesen-Gülay Günlük Şenesen, Literatür yayıncılık, İstanbul, 1995, s.59-68.

Tahmin edilen ilişki  $y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + e$  formülü ile

Tahmin edilen regresyon doğrusu  $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$  formülü ile ifade edilir.

Yukarıdaki modelde yer alan “^” işareti gerçek parametrenin tahmini değerini gösterirken, e ise u hata teriminin gerçek değerinin tahmin değerini ifade eder ve bu terime kalıntı terimi denilir.

Regresyon denkleminde yer alan katsayıları, en küçük kareler yönteminden yararlanarak gerçek y değerleri ile tahmin edilen y değerleri arasındaki farklar minimize edilerek bulunur.



**Şekil 1.1.** En küçük karelerin geometrik yorumu

Bir bağımsız değişken ile bir doğru oluşturan regresyon denklemi, iki bağımsız değişkenli üç boyutlu uzayda iki boyutlu bir düzlemin denklemini verir<sup>11</sup>.

Üç değişkenli bir modelde her noktanın üç koordinata sahip olduğu ve bir yüzey hesaplanacağı için denklem bir doğru denklemi olmayıp yukarıdaki şekilde gösterildiği

<sup>11</sup> Nevin Uzgören, “Uyum Analizinin Teorik Esasları ve Regresyon Analizi ile Benzerliğinin Grafikselleştirilmesi”, Dumlupınar Üniversitesi Sosyal Bilimler Dergisi, Sayı 18, 2007, s.13.



gibi en küçük kareler yüzeyidir. Burada gerçek y değerleri ile tahmin edilen y değerleri arasındaki uzaklıkların farklarının kareleri toplamı minimize edilir.<sup>12</sup>

En basit durum olan iki bağımsız değişken için örnek regresyon denklemi aşağıda verilmiştir<sup>13</sup>.

$$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + e$$

$$\min \sum e^2 = \sum \left( y - \hat{\beta}_0 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_2 \right)^2$$

Bu regresyon denkleminde ilk olarak örnekleme ilişkin gerçek gözlem değeri y ile bunun tahmini değeri  $\hat{y}$  arasındaki fark alınarak kalıntı terimi e'ye ulaşılır Daha sonra bu sapmaların karelerinin toplamı alınır. Regresyon modelini minimize eden optimal değerleri bulabilmek için modelde yer alan kalıntı teriminin her bir parametreye göre kısmi türevi alınır.

$$\frac{\delta \sum e^2}{\delta \hat{\beta}_0} = 0 \quad \frac{\delta \sum e^2}{\delta \hat{\beta}_1} = 0 \quad \frac{\delta \sum e^2}{\delta \hat{\beta}_2} = 0$$

Kısmi türevler alındıktan sonra aşağıdaki normal denklemler bulunur.

$$\sum y = n \hat{\beta}_0 + \hat{\beta}_1 \sum x_1 + \hat{\beta}_2 \sum x_2$$

$$\sum yx_1 = \hat{\beta}_0 \sum x_1 + \hat{\beta}_1 \sum x_1^2 + \hat{\beta}_2 \sum x_1 x_2$$

$$\sum yx_2 = \hat{\beta}_0 \sum x_2 + \hat{\beta}_1 \sum x_1 x_2 + \hat{\beta}_2 \sum x_2^2$$

Kısmi türevler sonucunda ulaşılan normal denklemdeki x ve y değerleri yerine her x ve y değerinin aritmetik ortalamasından farkları ( $x^* = x - \bar{x}$  ve  $y^* = y - \bar{y}$ ) yerleştirilir.

<sup>12</sup> Douglas Montgomery-Elizabeth Peck – Geoffrey Vining, **Introduction To Linear Regression Analysis**, Third Edition, A Wiley Interscience Publication, New York, 2001, s.81.

<sup>13</sup> Douglas Montgomery-Elizabeth Peck – Geoffrey Vining, a.g.e.,s.72.

Değişkenlerin kendi ortalamalarından farkları biçiminde ifade edildikleri aşağıdaki formüller, parametrelerin tahmin değerlerini bulmak için kullanılabilir.

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}_1 - \hat{\beta}_2 \bar{x}_2$$

$$\hat{\beta}_1 = \frac{(\sum y^* x_1^*)(\sum x_2^{*2}) - (\sum y^* x_2^*)(\sum x_1^* x_2^*)}{(\sum x_1^{*2})(\sum x_2^{*2}) - (\sum x_1^* x_2^*)^2}$$

$$\hat{\beta}_2 = \frac{(\sum y^* x_2^*)(\sum x_1^{*2}) - (\sum y^* x_1^*)(\sum x_1^* x_2^*)}{(\sum x_1^{*2})(\sum x_2^{*2}) - (\sum x_1^* x_2^*)^2}$$

Çoklu doğrusal regresyon modelinde açıklayıcı değişkenlerin sayısı arttıkça iterasyon yöntemiyle tahmincilerin elde edilmesi zorlaşır. Bu analizi basitleştirmek amacıyla matris yaklaşımına başvurulur.

Matris teriminde bu ilişki,

$$y = x\beta + u$$

biçiminde ifade edilir. Örneklem için gerçek ilişki ise

$$y = x\hat{\beta} + e$$

dir.

$$\sum e^2 = \sum (y - \hat{y})^2 \text{ terimi en küçükleyen regresyon katsayılarının tahminini}$$

belirler.

$$\sum e^2 = e' \cdot e = (y - x\hat{\beta})' (y - x\hat{\beta})$$

$\sum e^2 = yy' - 2y'x\hat{\beta} + \hat{\beta}'x'x\hat{\beta}$  eşitliğini  $\beta$  ya göre kısmi türevi alınıp sıfıra eşitlenir

$$\frac{\delta \sum e^2}{\delta \hat{\beta}} = 2x'y + 2x'x\hat{\beta} = 0$$

$$x'x\hat{\beta} = x'y$$

Buradan  $\beta$  tahmincisine ulaşabilmek için her iki taraf  $(x'x)$  in tersiyle çarpılır.

$$(x'x)^{-1}x'y = (x'x)^{-1}x'y$$

$$\hat{\beta} = (x'x)^{-1}x'y \text{ bulunur.}$$

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \cdot \\ \cdot \\ \hat{\beta}_n \end{bmatrix} = \begin{bmatrix} n & \sum x_1 & \sum x_2 & \cdot & \sum x_k \\ x_1 & \sum x_1^2 & \sum x_1x_2 & \cdot & \sum x_1x_k \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_k & \sum x_1x_k & \cdot & \cdot & \sum x_k^2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y \\ \sum x_1y \\ \cdot \\ \cdot \\ \sum x_ky \end{bmatrix}$$

### 1. 2. 2. Tahminlerin Standart Hatası

Tahminin standart hatası, regresyon yardımı ile yapılacak tahminin doğruluğunu gösterir. Tahminin standart hatası, gerçek  $y$  değerlerinin regresyon yüzeyi etrafında ne kadar uzağa dağıldıklarını gösterir. Hata sıfıra ne kadar yakınsa yapılan tahminler, o derece güvenilirdir. Tahminin standart hatasını bulmaya ilişkin formül aşağıda verilmiştir.

$$\sigma_e = \sqrt{\frac{\sum e^2}{n-k}} = \sqrt{\frac{\sum (y - \hat{y})^2}{n-k}}$$

Formülde yer alan n değeri örnekleme ilişkin veri sayısını, k değeri de modelde yer alan değişken sayısını gösterir. Bu formülü üç değişkenli bir model için aşağıda verildiği şekilde de yazabiliriz.

$$\sigma_e = \sqrt{\frac{\sum y^2 - \hat{\beta}_0 \sum y - \hat{\beta}_1 \sum x_1 y - \hat{\beta}_2 \sum x_2 y}{n-3}}$$

### 1. 2. 3. Çoklu Determinasyon(Belirlilik) Katsayısı

Çok değişkenli doğrusal regresyon modelinde,  $r^2$  modelin bağımsız değişkenlerinin bağımlı değişkendeki değişkenliği ne derece açıkladığının ölçüsüdür. İki değişkenli modelde determinasyon katsayısı  $r^2$  ile gösterilirken ikiden çok değişkenli modellerde ise  $R^2$  ile gösterilir.  $R^2$  de aslında  $r^2$  den türetilmiştir ve aşağıdaki gibi formüle edilir<sup>14</sup>.

$$\begin{array}{ccc} \text{(SST)} & \text{(SSE)} & \text{(SSR)} \\ \text{Toplam değişme} & = & \text{Açıklanan değişme} + \text{Açıklanamayan değişme(hata)} \end{array}$$

$$\sum (y - \bar{y})^2 = \sum (\hat{y} - \bar{y})^2 + \sum (y - \hat{y})^2$$

$$SST = SSE + SSR$$

$$1 = \frac{SSE}{SST} + \frac{SSR}{SST}$$

$$\frac{SSE}{SST} = 1 - \frac{SSR}{SST}$$

$$R^2 = 1 - \frac{SSR}{SST}$$

<sup>14</sup> Akif Bakar - Celal Aydın, **İstatistik**, Nobel yayınevi, Ankara, 2008, s.251.

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2} = 1 - \frac{\sum e^2}{\sum y^{*2}}$$

n değişkenli bir model için daha genel bir şekilde de aşağıdaki gibi genelleştirilebilir.

$$R^2 = \frac{\hat{\beta}_1 \sum yx_1 + \hat{\beta}_2 \sum yx_2 + \dots + \hat{\beta}_n \sum yx_n}{\sum y^2}$$

$R^2$  nin değeri 0 ile 1 arasındadır. Bu değer 0'a yaklaşması bağımsız değişkenlerin bağımlı değişkendeki değişmeyi açıklayamadığını gösterirken, 1'e yaklaşması ise bağımlı değişkendeki değişimin bağımsız değişkenler tarafından iyi açıklandığını gösterir<sup>15</sup>.

$R^2$  değeri modele yeni bir değişken eklendiğinde daima artar. Çünkü  $R^2$  ifadesinin payı artarken paydası aynı kalır. Bu sakıncayı ortadan kaldırmak için modele yeni değişkenler eklendiğinde serbestlik derecesi hesaba katılarak düzeltme yapılır. *Düzeltilmiş çoklu determinasyon* katsayısı aşağıda verilen formül ile hesaplanabilir.

$$\bar{R}^2 = 1 - (1 - R^2) \frac{n-1}{n-k} \quad \text{ya da} \quad \bar{R}^2 = 1 - \left[ \frac{\sum e^2 / (n-k)}{\sum y^2 / (n-1)} \right]$$

#### 1. 2. 4. Regresyon Modelinin Genel Anlamlılığının (Uyum İyiliğinin) Sınanması

Bağımsız değişkenlerin her biri ile bağımlı değişken arasındaki ilişkiyi sağlayan parametrelerin hepsi birlikte test edilmek istenirse F-testine başvurulur. F testi bağımsız değişkenlerin tamamının bağımlı değişken üzerindeki etkisini test etmek için kullanılır.

$H_0 : \beta_1 = \beta_2 = \beta_3 = \dots = \beta_n = 0$  (Sıfır hipotezi)

$H_1 : \text{En az bir tane } \beta_j \neq 0$  (Karşıt hipotez)

<sup>15</sup> Nuran Bayram, **Veri Analizi: Excel ve SPSS Uygulamalarıyla Birlikte**, Siyasal Kitabevi, Bursa, 2012, s.205.

**Tablo 1.1.** Çoklu Doğrusal Regresyonda Varyans Analizi

Değişim Kaynağı	Serbestlik Derecesi	Kareler Toplamı	Kareler Ortalaması	Test İstatistiği
Regresyon	k-1	$\sum (\hat{Y} - \bar{Y})^2 = SSR$	$MSR = \frac{SSR}{k-1}$	$F = \frac{MSR}{MSE}$
Hata	n-k	$\sum (Y - \hat{Y})^2 = SSE$	$MSE = \frac{SSE}{n-k}$	
Toplam	n-1	$\sum (Y - \bar{Y})^2 = SST$	$MST = \frac{SST}{n-1}$	

Belirlenen  $\alpha$  anlamlılık düzeyinde (genellikle  $\alpha=0,01$  veya  $\alpha=0,05$  alınır) k-1, n-k serbestlik derecesinde F tablo değeri hesaplanan F değeri ile karşılaştırılır. Eğer hesaplanan değer F tablo değerinden daha büyükse  $H_0$  hipotezi red edilir.  $H_0$  hipotezinin red edilmesi regresyonun anlamlı olduğunu yani bağımsız değişkenler X, bağımlı değişken olan Y yi açıklamakta anlamlı olduğunu ve parametrelerin hepsinin sıfır olmadığını gösterir. Eğer hesaplanan değer F tablo değerinden daha küçükse  $H_0$  hipotezi red edilemez. Bunun anlamı ise regresyonun anlamlı olmadığıdır. Bir başka deyişle bağımsız değişkenlerin bağımlı değişkeni açıklamakta yetersiz kalmasıdır.

$$F = \frac{\sum \hat{y}^2 / (k-1)}{\sum e^2 / (n-k)}$$

$R^2$  nin daha önce hesaplanmış olması halinde F testi  $R^2$  cinsinden yapılabilir. Aslında F oranı  $R^2$  nin anlamlılığının sınanmasıdır. Çünkü  $R^2$  istatistikî olarak anlamlı bulunmazsa Y ile X arasında doğrusal bir ilişki yok demektir. Yani parametreler sıfıra eşittir<sup>16</sup>.

$$F = \frac{\sum \hat{y}^2 / (k-1)}{\sum e^2 / (n-k)} = \frac{R^2 / (k-1)}{(1-R^2) / (n-k)}$$

<sup>16</sup> Anna Koutsoyiannis, **Ekonometri Kuramı**, çev. Ümit Şenesen ve Gülay, G. Şenesen, İstanbul Teknik Üniversitesi Matbaası, İstanbul, 1992, s.157.

### 1. 2. 5. Regresyon Katsayılarının Önem Testi

Parametrelerin tek tek istatistiksel olarak anlamlılığını ölçmek için t testi kullanılır. Buna ilişkin hipotez testleri aşağıda yer almaktadır. Burada anlamlılığı ölçülen parametre için sıfır hipotezinin red edilip karşıt hipotezin kabul edilmesi regresyon modelinde yer alan parametrenin anlamlı olduğunu gösterirken, sıfır hipotezinin red edilememesi yani kabul edilmesi ise modelde yer alan parametrenin anlamsız olduğunu ifade eder.

$$H_0 = \beta_i = 0 \text{ (Sıfır hipotezi)} \quad (i= 1,2\dots n)$$

$$H_1 = \beta_i \neq 0 \text{ (Karşıt hipotez)}$$

T istatistiği aşağıda verilen formül ile bulunur.

$$t = \frac{\beta_i}{\sigma_{\beta_i}}$$

Seçilen anlamlılık düzeyinde t dağılım tablosundan bulunan t değeri ile yukarıda verilen formül ile hesaplanan t değeri karşılaştırılır. Eğer  $|t_{hes}| > |t_{tab}|$  yani hesaplanan t değeri t tablo değerinden büyük ise sıfır hipotezi red edilir, aksi durumda ise kabul edilir.

Yapay sinir ağ modellerinde görüleceği üzere basit algılayıcı model, çok katmanlı algılayıcı model ve radyal tabanlı fonksiyon ağ modeli ile çoklu doğrusal regresyon model arasında sıkı bir ilişki vardır. Şimdi çoklu doğrusal regresyon modeli ile yakından ilişkili olan yapay sinir ağlarını açıklamaya çalışalım.

## İKİNCİ BÖLÜM

### YAPAY SİNİR AĞLARININ KURAMSAL YAPISI

#### 2. 1. Yapay Sinir Ağlarının Tanımı ve Özellikleri

Yapay Sinir Ağları (YSA), bilgisayar dünyasında insan beyninin ve sinir sisteminin davranışlarını taklit etme esası üzerine kurulmuş yeni bir bilgi işleme yaklaşımıdır. İnsan beyninin taklit edilmesi ile ortaya çıkan bu yapılar her ne kadar nanosaniyeler içerisinde işlemleri yerine getirebilme kapasiteli olsa da insan beyninin milisaniyeler düzeyindeki işlevsel hızına ulaşamamıştır.

Yapay sinir ağları, biyolojik sinir ağlarının karakteristiklerine benzer şekilde çalışan bir bilgi işleme sistemidir<sup>17</sup>. Bir başka tanıma göre ise bir sinir ağı, basit işlem birimlerinden oluşan, deneyimsel bilgileri biriktirmeye yönelik doğal bir eğilimi olan ve bunların kullanılmasını sağlayan yoğun bir şekilde paralel dağıtılmış bir işlemcidir<sup>18</sup>. YSA beyne iki yönüyle benzemektedir. Birincisi, bilgiyi öğrenerek elde eder. İkincisi ise nöronlar (sinir hücresi) arası ağırlıklar; bilgiyi saklamak için kullanılır<sup>19</sup>.

Yapay sinir ağları, biyolojik nöron yapısının aşağıdaki varsayımlara dayanan matematiksel modellerin geliştirilmesi sonucunda oluşturulmuştur.<sup>20</sup>

- Bilgi işleme, nöron adı verilen birçok basit elemanlarda gerçekleşir;
- Sinyaller, nöronlar arasındaki ilişkiyi sağlayan bağlantılarla iletilir;
- Her bir bağlantının bir ağırlık değeri vardır ve bu değer, gerçek nöronlarda olduğu gibi sinyal geçişini üretmektedir;

<sup>17</sup> Ercan Öztemel, **Yapay Sinir Ağları**, Papatya Yayıncılık, İstanbul, 2003, s.29.

<sup>18</sup> Simon Haykin, **Neural Networks: A Comprehensive Foundation**, New Jersey, 1999, s.1.

<sup>19</sup> Simon Haykin, a.g.e.,s.2

<sup>20</sup> Laurene Fausett, **Fundamentals of Neural Networks**, Prentice Hall International, London, 1994, s.3.



- Sinir ağı içindeki her bir nörona bir aktivasyon(transfer) fonksiyonu uygulanır (genelde bu doğrusal olmayan bir fonksiyondur) ve bu fonksiyonun giriş değeri sayesinde nöronun çıkış sinyali belirlenir.

Bir yapay sinir ağı;

- Nöronlar arasındaki bağlantının bir modeli yani mimarisi ile
- Bağlantılardaki ağırlıkların hesaplanması (bu hesaplama, eğitim veya öğrenme algoritması olarak da adlandırılır) yöntemi ile
- Transfer fonksiyonu ile tanımlanabilir.

Bir sinir ağı, milyarlarca sinir hücresinin bir araya gelmesi ile oluşmaktadır. Sinir hücreleri birbirleri ile bağlanarak fonksiyonlarını yerine getirirler. Beynimizde yaklaşık 10 milyar adet sinir ve bunların da 60 trilyondan fazla sayıda bağlantısı vardır. İnsan beyni çok hızlı çalışabilen mükemmel bir bilgisayar gibi görülebilir. Bir grup insan resmi içerisinde tanıdık bir resmi 100-200 ms (milisaniye) gibi kısa bir sürede tanıyabilir<sup>21</sup>.

Yapay sinir ağları, girdi-çıkış sinirlerinden(nöronlarından) oluşan bir sistemdir. Bu sistemde sinirler arasındaki bağlantıların ağırlıkları öğrenme veya eğitime yeteneği vardır. Bunun sonucu olarak ağ bir girdi grubuna etkili bir karşılık üretebilmektedir. Bu karşılık, örneğin sınıflandırma olabildiği gibi, bir zaman serisi ile gelecek tahmini şeklinde de olabilir.

YSA'ların sahip olduğu öğrenme yeteneği, kolaylıkla farklı problemlere uyarlanabilirliği, öğrenme işleminden sonra daha az bilgiye gereksinim duyması, genelleme yapabilme yeteneği, paralel yapılarından dolayı hızlı işlem yapabilmeleri ve zor matematiksel modelleri oldukça hızlı çözebilmeleri gibi sahip oldukları özellikleri nedeniyle birçok farklı alanda başarıyla uygulanmaktadır.

Yapay sinir ağlarının özellikleri uygulanan ağ modeline ve algoritmasına göre değişmektedir. Ancak literatürde yapay sinir ağlarının genel özellikleri aşağıdaki gibi özetlenebilir:

---

<sup>21</sup> Ercan Öztemel, a.g.e, s.45.

- **Doğrusal Olmama**

Yapay sinir ağlarının temel işlem elemanı olan hücre doğrusal olmadığı için bu hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Aynı katmanlar arasında zaman bağımlılığı da bulunmamaktadır. Bu tüm sistemin eş zamanlı çalışabilmesine olanak vermekte ve hızı arttırmaktadır. Bu özellik sayesinde yapay sinir ağları ile doğrusal olmayan karmaşık problemlerin çözülmesi olmaktadır<sup>22</sup>.

- **Örnekten Öğrenme**

Biyolojik sistemlerde öğrenme, nöronlar arasındaki bağlantıların ayarlanması ile olur. İnsanoğlu doğumlarından itibaren bir yaşayarak öğrenme süreci içerisinde. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe bağlantılar ayarlanır ve yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir.

Öğrenme, eğitime yoluyla örnekler kullanarak olur; başka bir deyişle, eğitime algoritmasının bu verileri kullanarak bağlantı ağırlıklarını bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur<sup>23</sup>. Bu şekilde yapay sinir ağları örnekleri kullanarak ilgili olay hakkında genelleme yapabilecek yeteneğe kavuşturulur<sup>24</sup>. Öğrenmede ağ, kullanılan öğrenme tekniğine göre danışmanlı öğrenme, danışmansız öğrenme gibi sistemin davranışını öğrenir. Örneğin, danışmanlı öğrenmede ağa rastgele seçilmiş bir örnek sunulur ve ağın bağlantı ağırlıkları, girdi değerlerinin işlenmesi sonucu ağın üreteceği çıktı ile önceden bilinen çıktı arasındaki fark en aza ininceye kadar güncellenir.

- **Paralellik**

YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağıtılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır. Oysa alışılmış bilgi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da hız ve güvenilirlik sorunlarını beraberinde getirmektedir. Seri bir işlem

---

<sup>22</sup> Patrick K.Simpson, **Artificial Neural Network**, Pergamon Press, New York, 1989,s.5.

<sup>23</sup> [tr.wikipedia.org/wiki/Yapay\\_sinir\\_ağları](http://tr.wikipedia.org/wiki/Yapay_sinir_ağları)

<sup>24</sup> Ercan Öztemel, a.g.e, s.31.

gerçekleşirken herhangi bir birimin yavaş oluşu tüm sistemi doğrudan yavaşlatırken, paralel bir sistemde yavaş bir birimin etkisi ise çok azdır<sup>25</sup>.

- **Genelleme Yeteneği**

Yapay sinir ağları, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir yapay sinir ağı, bozuk karakter girişleri olduğunda doğru karakterlerin elde edilmesini mümkün kılar<sup>26</sup>.

- **Uyarlanabilirlik**

Yapay sinir ağları, problemdeki değişikliklere göre yeniden eğitilebilmesi ve farklı koşullara uyarlanabilmesini sağlamaktadır. Bu özellik yapay sinir ağlarının örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılmasını sağlar<sup>27</sup>.

- **Hata Toleransı**

Yapay sinir ağlarındaki paralel yapı, ağın sahip olduğu bilginin tüm bağlantılara yayılmasını sağlar. Dolayısıyla bazı bağlantıların veya hücrelerin etkisiz hale gelmesi ağın doğru bilgi üretmesini önemli derecede etkilemez. Bununla birlikte ağların geleneksel yöntemlere göre hatayı tolere etme yetenekleri yüksek olur<sup>28</sup>. Yapay sinir ağlarının hataya toleranslı olma özelliği gürültülü(hatalı) ve eksik verilerden anlamlı düzeyde doğru çıktılar üretebilmesini sağlar. Oysa geleneksel bilgisayarlar genellikle eksiksiz verilere gereksinim duyar

## 2.2. Yapay Sinir Ağlarının Tarihçesi

Yapay sinir ağları kavramı beynin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya çıkmıştır.

İnsan beyнинin nasıl çalıştığı ile ilgili ilk bilgileri 1890 yılında William James vermiştir. 1940 yılından önce bazı bilim adamlarının (Pavlov, Helmholtz) yapay sinir ağı

---

<sup>25</sup> Patrick K Simpson, a.g.e, s.6.

<sup>26</sup> Halit Ergezer- Mehmet Dikmen- Erkan Özdemir, **Yapay Sinir Ağları ve Tanıma Sistemleri**, Pivolka 6, ELYAD DAL Araştırma Laboratuvarları, Başkent Üniversitesi, Ankara, 2003.

<sup>27</sup> Patrick K. Simpson, a.g.e, s.6.

<sup>28</sup> Patrick K. Simpson, a.g.e, s.6.

kavramı üzerinde çalıştıkları bilinmektedir. Ancak bu çalışmaların mühendislik değerinin olduğu söylenemez<sup>29</sup>.

1943 yılında sinir fizyologu olan Warren McCulloch ve matematikçi olan Walter Pitts, “Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap” adı ile yayımlanan bu çalışmada, bir sinir hücresinin ilk matematiksel modeli ele alınmıştır.<sup>30</sup>

1949’da Hebb “Davranış Organizasyonu” isimli kitabında, öğrenme ile ilgili temel teoriyi işlemiştir. Hebb tarafından geliştirilen kural Hebbian öğrenme kuralı olarak adlandırılmaktadır. Bu kural, sinir ağının bağlantı sayısı değiştirildiğinde öğrenmenin olabileceğini öngörmektedir<sup>31</sup>.

Otomatik olarak ağırlık ayarı yapabilen ilk nöro-bilgisayar Snarc Marvin Minsky ve Dean Edmonds tarafından 1951 yılında geliştirilmiştir<sup>32</sup>.

1954 yılında Farley ve Clark, rassal ağlar ile adaptif tepki üretme kavramını ortaya koymuşlardır. Bu kavram 1958 yılında Rosenblatt ve 1961 yılında Caianiello tarafından geliştirilmiştir. Özellikle Rosenblatt tarafından geliştirilen basit algılayıcı (perceptron) modeli yapay sinir ağları ile ilgili çalışmaların önce gelişmesine sonra duraklamasına, ardından çok katmanlı algılayıcıların ortaya çıkmasını sağlamıştır<sup>33</sup>. Basit algılayıcı model tek katmanlı, eğitilebilen ve tek çıkışa sahip olan bir yapay sinir ağıdır.

1959 yılında Bernard Widrow ve öğrencisi Marcian Hoff, ADALINE (Adaptive Linear Neuron) modelini ortaya atmışlardır. Yapay sinir ağlarının mühendislik uygulamalarında kullanılmaya başlanması için atılan ilk adımlardan biri olan bu model, 1970’lerin sonlarında ortaya çıkan ve ADALINE modelinin çok katmanlı hali olan MADALINE’nin temelini oluşturur. Basitçe sınıflandırma yapabilen adaline algılayıcı ile

---

<sup>29</sup> Ercan Öztemel, a.g.e, s.37.

<sup>30</sup> Mostafa, Y. S. Abu, **Neural Networks for Computing**, In Proceedings of the American Institute of Physics Meeting, New York, 1986,s.7.

<sup>31</sup> Çetin Elmas, **Yapay Zekâ**, Popüler Bilim, Sayı: 124,2004, s.32-37.

<sup>32</sup> Ercan Öztemel, a.g.e, s.37

<sup>33</sup> Ercan Öztemel, a.g.e, s.37.

aynı özelliklere sahip olup sadece öğrenme algoritması daha gelişmiş bir modeldir<sup>34</sup>. Madaline modeli ise, telefon hatlarında oluşan yankıları yok eden bir uyarlanabilir süzgeç olarak kullanılmıştır. Bu model, gerçek dünya sorunlarına uygulanmış ilk sinir ağıdır ve halâ kullanımda bulunmaktadır<sup>35</sup>.

Yapay zekâ biliminin önemli isimlerinden, Marvin Minsky ve Seymour Papert'ın 1969 yılında yayınladıkları Algılayıcılar (Perceptrons) isimli kitapla, yapay sinir ağları ile ilgili çalışmaları kısıtlamışlardır. Bu kitapta, algılayıcıların doğrusal olmayan problemlere çözüm üretemeyeceği sonucu ortaya atılmıştır. Bu tezlerini ileri sürerken yapay sinir ağlarının XOR problemini çözemediğini kanıtlamalarıyla bu alanda yapılan tüm araştırmalar duraklama dönemine girmiştir. Aynı yıl, Amerika Birleşik Devletlerinde araştırma geliştirme çalışmalarını yürüten bir organizasyon olan DARPA yapay sinir ağları ile ilgili çalışmaları desteklemeyi durdurmuştur<sup>36</sup>.

1982 yılında Hopfield tarafından yayınlanan “Yapay Sinir Ağları ve Gelişen Kolektif Hesapsal Yetenekli Fiziksel Sistemler” isimli makalede, yapay sinir ağlarının basit bir analog devre modelini sunmuş ve bu modeli kullanarak analog-ikil dönüştürücü tasarımı, Gezgin Satıcı gibi problemler için çözümler sunmuştur<sup>37</sup>. Böylece Hopfield'ın sinir ağı modellerini fiziksel olayları açıklamada kullanması ile duraklama dönemi sonlanmış ve yapılan bu çalışmalarla birlikte yapay sinir ağlarında yeni bir dönem başlamıştır.

1986 yılında Rumelhart ve arkadaşları tarafından çok katmanlı yapay sinir ağları için geriye yayılma algoritması geliştirilmiştir. Geriye yayılma algoritması yapay sinir ağlarında oldukça yaygın olarak kullanılan bir algoritmadır.

---

<sup>34</sup> Ercan Öztemel, a.g.e, s.38.

<sup>35</sup> Çetin Elmas, a.g.e, s.28.

<sup>36</sup> Ercan Öztemel, a.g.e, s.38.

<sup>37</sup> Haldun Akpınar, “Yapay Sinir Ağları Gelişimi ve Yapılarının incelenmesi”, İstanbul Üniversitesi İşletme Fakültesi Dergisi, Cilt:23, Sayı:1,1994, s.71.

1988’de Broomhead ve Lowe, özellikle filtreleme problemlerinde oldukça başarılı sonuçlar alabildikleri, Radyal Tabanlı Fonksiyonlar ( RBF) modelini geliştirdiler. Bu ağın çok katmanlı algılayıcılara alternatif olarak geliştirildiğini belirttiler. Daha sonra Specht bu ağların daha gelişmiş şekli olan Probabilistik Ağları (PNN)ve Regresyon Ağlarını (RNN) geliştirdi<sup>38</sup>.

1987 yılında ilk yapay sinir ağları sempozyumu yapılmış ve sonrasında uygulamalar artmaya başlamıştır. Günümüzde de bu konuda çalışan çok sayıda bilim adamı ve bu alanda yayınlanan birçok bilimsel yayın ile hızlı gelişimini devam ettirmektedir.

### 2.3. Yapay Sinir Ağlarının Üstünlükleri ve Sakıncaları

Yapay sinir ağları yönteminin bazı üstünlük ve sakıncaları bulunmaktadır. Bunlar<sup>39</sup>:

#### Üstünlükleri:

- Yapay sinir ağları eğitimleri sırasında kendilerine verilen örneklerden genellemeler çıkarırlar. Bu genellemeler ile kendilerine verilen örnekler hakkında bilgi üretirler.
- Yapay sinir ağları matematiksel bir modele ihtiyaç duymaz. Ayrıca yapay sinir ağı modelinde verilerin yapay sinir ağının eğitiminde kullanılması için gerekli bir varsayıma gerek yoktur.
- Yapay sinir ağları verilerden hareketle bilinmeyen ilişkileri akıllıca hemen ortaya çıkarabilir. Dolayısıyla ağların bu özelliği, uygulama açısından çok önemlidir.
- Geleneksel bilgisayar sistemleri, sistemde oluşacak hatalara karşı çok duyarlıdır. Sistemde meydana gelebilecek en ufak bir hata olduğunda sonuca ulaşamaz veya sonuçlarda büyük hataya yol açabilir. Ancak yapay sinir ağlarının bir veya birkaç nöronunun zarar görmesi durumunda sistemi geleneksel bilgi işlem teknolojilerinde olduğu kadar etkilemez. Çünkü hataya karşı toleranslıdır.
- Ağ, ağırlık katsayısı gibi kendi parametreleri değiştiğinde somut bir problemin çözümü için kendini uyarlama özelliğine sahiptir.

---

<sup>38</sup> Ercan Öztemel, a.g.e, s.40.

<sup>39</sup> Çetin Elmas, a.g.e, s. 27.

- Ağlar doğrusal değildir. Bu nedenle, yapay sinir ağ modeli doğrusal olmayan karmaşık problemlerin çözümünde kullanılır. Dolayısıyla doğrusal model çözüm tekniklerine göre üstünlük sağlar.

#### **Sakıncaları:**

- Probleme uygun bir ağ yapısının belirlenmesi zor bir süreçtir. Genellikle ağ yapısı belirlenirken deneme-yanılma yöntemi kullanılır. Bu işlemde bazen uzun zaman gerektirir. Eğer problem için uygun bir ağ yapısı belirlenemezse, çözümü olan bir problemin çözülememesi veya performansı düşük çözümlerin elde edilmesi ile karşılaşılır. Aynı zamanda böyle bir durumda bulunan çözümün en iyi çözüm olduğunun garantisini de vermez.
- Bazı ağlarda ağın parametre değerlerinin (örneğin öğrenme katsayısı, hücre sayısı v.b.) belirlenmesinde bir kural olmaması, çözümlere ulaşmada sorun yaratır. Bu değerler kullanıcının tecrübesine bağlıdır.
- Problemin ağ gösterimi çok önemli bir etkidir. Bilindiği üzere YSA sadece sayısal verilerle çalışır ve problemin sayısal değerlere çevrilmesi gerekmektedir.
- Seçilen ağın eğitiminin ne zaman bitirileceğine dair kesin kurallar yoktur. Eğitimde hata toleransının belli bir değer altına inmesi eğitimin tamamlanması için yeterli görülmektedir.

#### **2.4. Yapay Sinir Ağlarının Uygulama Alanları**

Yapay sinir ağları başlıca; sınıflandırma, modelleme ve tahmin uygulamaları olmak üzere, pek çok alanda kullanılmaktadır. Başarılı uygulamalar incelendiğinde, YSA'ların çok boyutlu, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek veriler olduğunda ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldığı görülmektedir.

Yapay sinir ağları; günlük hayatta kullanılan finansal konular olmak üzere mühendislik, tıp, işletme, iktisat gibi pek çok alanda uygulanabilmektedir. Evimizdeki aletlerden cep telefonlarına kadar günlük hayatımızda yapay sinir ağları uygulamaları görmek mümkündür. Bu uygulamalar çok çeşitli olup, en önemlileri aşağıda verilmiştir<sup>40</sup>.

---

<sup>40</sup> Tuğba, Saraç, Yapay Sinir Ağları, Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Bölümü Anabilim Dalı, Ankara,2004,s.11.

**Arıza Analizi ve Belirlenmesi:** Bir sistemin, cihazın ya da elemanın düzenli çalışma şeklini öğrenen bir yapay sinir ağı yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı vardır. Bu amaçla yapay sinir ağları, elektrik makinelerinin, uçakların ya da bileşenlerinin, entegre devrelerin v.b. arıza analizinde kullanılmaktadır.

**Finansal Uygulamalar:** Makro ekonomik tahminler, kredi kartı hilelerinin tespiti, kredi kartı kurumlarında iflas tahminleri, banka kredilerinin değerlendirilmesi, emlak kredilerinin yönetilmesi, döviz kuru tahminleri, risk analizleri gibi işlemlerde kullanılmaktadır.

**Tıp Uygulamaları:** EEG ve ECG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı, transplantasyon zamanlarının optimizasyonu ve hastanelerde giderlerin optimizasyonu gibi konularda uygulanmaktadır.

**Savunma Sanayi Uygulamaları:** Silahların otomasyonu ve hedef izleme, nesnelere/görüntüleri ayırma ve tanıma, yeni algılayıcı tasarımı ve gürültü önleme gibi alanlarda kullanılmaktadır.

**Haberleşme Uygulamaları:** Görüntü ve veri sıkıştırma, otomatik bilgi sunma servisleri, konuşmaların gerçek zamanda çevirisi gibi alanlarda uygulanmaktadır.

**Üretim Uygulamaları:** Üretim sistemlerinin optimizasyonu, ürün analizi ve tasarımı, ürünlerin kalite analizi ve kontrolü, planlama ve yönetim analizi gibi alanlarda kullanılmaktadır.

**Otomasyon ve Kontrol Uygulamaları:** Uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yol bulma ve gösterme, robot sistemlerin kontrolü, doğrusal olmayan sistem modelleme ve kontrolü, elektrikli sürücü sistemlerin kontrolü gibi alanlarda uygulanmaktadır.

Farklı alanlardaki uygulamalar incelendiğinde yapay sinir ağlarının genel olarak şu fonksiyonları gerçekleştirmek amacıyla kullanıldıkları görülmektedir<sup>41</sup>.

**Tahmin:** Bu amaçla kullanılan yapay sinir ağları, ağa sunulan bilgilerden yararlanarak karşılık gelen çıktı değerlerini tahmin edebilir.

---

<sup>41</sup> Ercan Öztemel, a.g.e, s.104.



**Sınıflandırma:** Bu amaçla kullanılan yapay sinir ağları, kendilerine verilen bilgileri kategorize etmek görevini üstlenmektedirler.

**Veri ilişkilendirme:** Bu amaçla eğitilen ağlar, ağa sunulan verilerin hatalı olup olmadıklarını belirler ve öğrendikleri bilgiler ile eksik bilgileri doldurur.

**Veri filtreleme:** Bu amaçla eğitilen ağlar, birçok veri arasından uygun verileri belirleme görevini yerine getirmektedir.

**Tanıma ve eşleştirme:** Eksik, karmaşık, belirsiz bilgileri işleyerek eşleştirme ve tanıma fonksiyonlarını gerçekleştirebilmektedir.

**Teşhis:** Bu amaçla geliştirilen ağlar, sistemlerin olumsuzluklarının ortaya konulması ve problemlerin belirlenmesi işlemini yerine getirir.

**Yorumlama:** Bir olay hakkında toplanan örneklerden elde edilen ve eğitim sonucu oluşturulan bilgiler kullanarak yeni olayların yorumlanması işlemleri yapay sinir ağları ile gerçekleştirilebilir.

## 2.5. Biyolojik Sinir Sistemleri

Yapay sinirlerin yapısını daha iyi anlayabilmek için insanın biyolojik sinir sistemini anlamak gerekir.

İnsan beyni, birbiri ile karmaşık ilişkiler içinde bulunan milyarlarca sinir hücresinden (nöron) oluşmaktadır. Bilim adamları insan beyninin çalışma şeklini, yapısını çözebilmek için geçmişten beri çeşitli çalışmalar yapmaktadır.

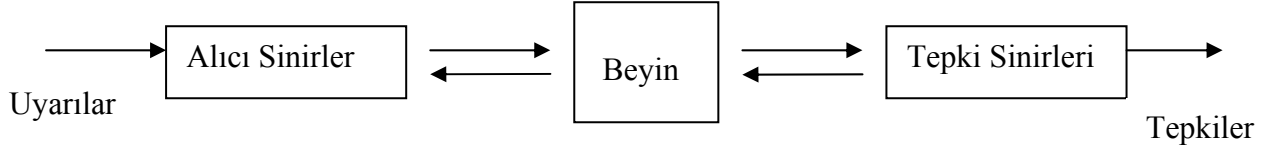
Yapay sinir ağları insan beyninin işleyişinden esinlenerek ortaya konulmuş modellerdir. İnsan beyni sinir hücrelerinin ve bu hücreler arasındaki bağlantıların işleyişiyle çalışmasını gerçekleştirir. İnsan beyninin üstünlüğü yüksek derecede paralel hesaplama yapabilen bir yapıya sahip olması ve bununla ortaya çıkan bilgiyi işleme yeteneğidir<sup>42</sup>.

Biyolojik sinir sistemi, merkezde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin bulunduğu üç katmanlı bir sistem olarak açıklanır. Bu sistem Şekil

---

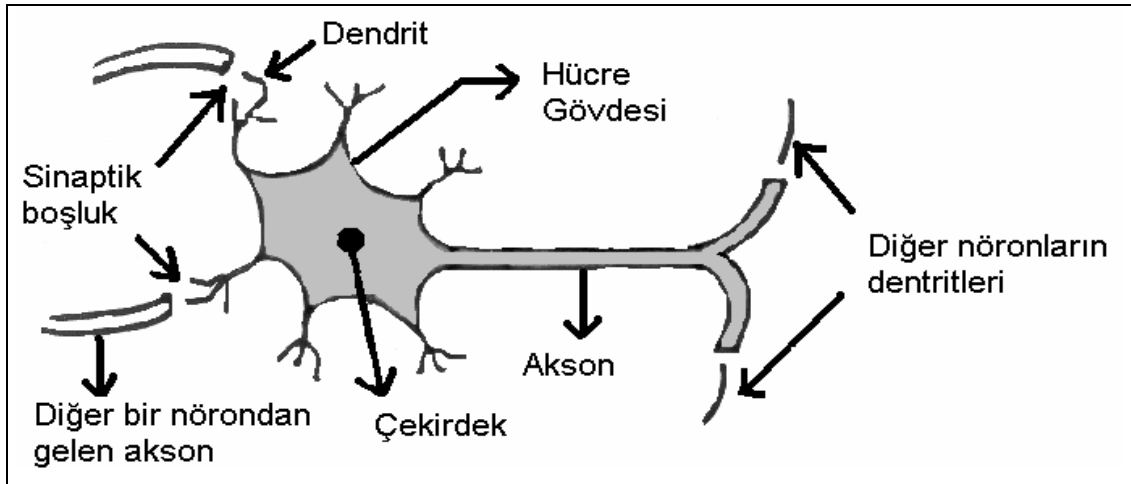
<sup>42</sup> Özer Arabacı, Makroekonomik Zaman Serisi Analizi ve Yapay Sinir Ağı Uygulamaları, Uludağ Üniversitesi, Bursa, 2007,s.48.

2.1' de gösterilmiştir<sup>43</sup>. Burada; çevreden gelen girdiler elektriksel sinyallere dönüştürerek beyine ileten alıcı sinirlerdir. Beynin ürettiği elektriksel sinyalleri çıktı olarak uygun tepkilere dönüştüren ise tepki sinirleridir. Alıcı ve tepki sinirleri arasında ileri ve geri besleme yaparak uygun tepkiler üreten merkezi sinir ağı(beyin)dır.



Şekil 2.1. Biyolojik Sinir Sisteminin Blok Gösterimi

Biyolojik sinir sisteminin temel yapı taşı nöron adı verilen sinir hücreleridir. Bir biyolojik nöronun üç tip bileşeni vardır. Bu bileşenler, *soma* (hücre gövdesi), *akson* ve *dentritlerden* oluşmaktadır. Biyolojik bir nöronun yapısı Şekil 2.2'de gösterilmiştir.<sup>44</sup> Şekilde görülen kısa ve dallanmış olan dentrit, giriş sinyallerini(bilgilerini) diğer nöronlardan alır. Alınan bu sinyaller, dentritler arasındaki sinaptik boşluklardan iletilen elektrik tepkileridir. Uzun ve tek olan akson ise çıkış sinyallerini diğer nöronlara taşır. Akson ile dentritin birleşim yerine sinaps adı verilir. Bunlar nöronlardan aldığı sinyalleri değerlendirir ve eşik değeri üzerinde bir giriş varsa bir sonraki hücreye iletirler.



Şekil 2.2. Biyolojik Nöronun Yapısı

<sup>43</sup> Simon Haykın, a.g.e, s.6.

<sup>44</sup> Laurene Fausett, a.g.e, s.6.

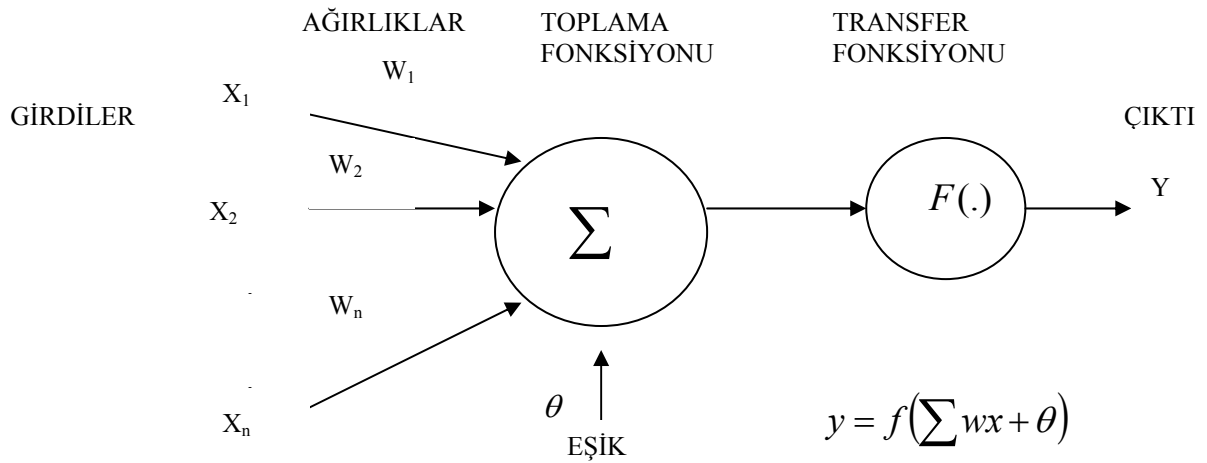
## 2.6. Yapay Sinir Hücresi

Biyolojik sinir ağlarının sinir hücrelerinden oluşması gibi, yapay sinir ağları da yapay sinir hücrelerinden meydana gelmektedir. Yapay sinir hücrelerine işlem elemanı ya da süreç elemanı da denilmektedir. Bir yapay sinir ağı, birbiriyle bağlantılı çok sayıda yapay sinir hücresinden meydana gelmektedir. Yapay sinir hücreleri biyolojik sinir hücrelerinin basit bir modelidir. Biyolojik sinir ile yapay sinirler arasındaki benzerlikler Tablo 2.1' deki gibi gösterilebilir<sup>45</sup>.

**Tablo 2.1.** Biyolojik Sinir Ağı ve Yapay Sinir Ağının Karşılaştırılması

Biyolojik Sinir Ağı	Yapay Sinir Ağı
Sinir Sistemi	Sinirsel Hesaplama Sistemi
Sinir Hücresi(Nöron)	İşlemci Eleman(Yapay Sinir Hücresi)
Sinaps	Sinirler arası Bağlantı Ağırlıkları
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Akson	İşlemci Eleman Çıktısı

Yapay sinir ağlarının içinde bulunan tüm sinir hücreleri bir veya birden fazla girdi alırlar ve tek bir çıktı verirler. Bu çıktı yapay sinir ağının dışına verilen bir çıktı olabileceği gibi başka bir yapay sinir hücresine girdi olarak da verilebilir.



**Şekil 2.3.** Yapay Sinir Hücresinin Fonksiyonel Yapısı

<sup>45</sup> Çetin Elmas, a.g.e.,s.35.

McCulloch-Pitts (1943) tarafından geliştirilen yapay sinir hücreleri (nöron), YSA'nın çalışmasının temelini oluşturan en küçük bilgi işleme birimidir. Bir yapay nöronun temel yapısı, genel haliyle, Şekil 2.3'te gösterilmiştir<sup>46</sup>. Şekildeki temel bileşenlerden biri olan girdiler  $x(i)$  ile ifade edilmekte ve  $i = 0,1,2,\dots,n$  girdi değerlerini göstermektedir. Bu girdi değerlerin her biri bir bağlantı ağırlığıyla çarpılır ve bu ağırlıklar  $w(i)$  ile gösterilir. En basit yapıda, bu çarpımlar toplanır. Bu toplama eşik değeri ( $\theta$ ) eklenerek bir transfer fonksiyonuna dönüştürülerek çıktı elde edilir. Uygulamada değişik toplama fonksiyonları ve transfer fonksiyonları kullanılmakta ve farklı ağ yapılarında uygulanmaktadır.

Bir yapay sinir hücresi genel olarak beş temel bileşenden oluşmaktadır. Bunlar;

- Girdiler
- Ağırlıklar
- Toplama fonksiyonu
- Transfer fonksiyonu
- Çıktı

Şimdi bu bileşenleri kısaca açıklayalım.

- **Girdiler:** Girdiler, yapay sinir hücresine dışarıdan alınan verilerdir. Bu veriler, ağırlık öğrenmesi istenen örnekler tarafından sağlandığı gibi başka bir sinir hücresinden veya yapay sinir hücresinin kendisinden de gelebilir. Girdiler( $x$ ),  $n$  elemanlı sütun vektörü olarak gösterilir.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

---

<sup>46</sup> Özgür Kişi-Turgay Partal, "Yapay Sinir Ağları ve Otopregresif Modellerle Rüzgâr Hızı Tahmini", **III. Atmosfer Bilimleri Sempozyumu Bildiri Kitabı**, İstanbul: 19-21 Mart 2003, s.168.

- **Ağırlıklar:** Ağırlıklar( $w$ ), gelen verilerin hücre üzerindeki etkisini belirleyen değerlerdir. Veriler(bilgiler), bağlantılar üzerindeki ağırlıklar üzerinden hücreye girer ve ağırlıklar yapay sinirde girdi olarak kullanılacak değerlerin göreceli kuvvetini gösterir. Ayrıca yapay sinir ağı içinde girdilerin hücreler arasında iletimini sağlayan tüm bağlantıların farklı ağırlık değerleri bulunur. Ağırlık değerlerinin büyük ya da küçük olması verilerin önemli veya önemsiz olduğu anlamına gelmez. Öte yandan verilerin artı veya eksi olması onların etkisinin pozitif veya negatif olduğunu gösterir.

Ağırlıklar,  $n$  elemanlı satır vektörü olarak düzenlenir.

$$W = [w_1 \dots w_n]$$

- **Toplama Fonksiyonu:** Toplama fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur. Net girdi, girişlerin ilişkin olduğu ağırlıkla çarpımlarının toplamıdır. Bunu aşağıdaki şekilde formüle edebiliriz.

$$Net\ Girdi = \sum_{i=1}^n w_i x_i$$

Yapay sinir ağının yapısına göre toplama fonksiyonu, maksimum, minimum, çarpım veya çeşitli normalizasyon işlemlerinden birisi olarak da ifade edilebilir. Bir problem için en uygun toplama fonksiyonu belirlemek için belirli bir formül yoktur. Dolayısıyla toplama fonksiyonu genellikle deneme yanılma yoluyla belirlenir. Ayrıca bir yapay sinir ağında bulunan süreç elemanlarının tamamı, aynı toplama fonksiyonlu olması gerekli değildir. Örneğin aynı yapay sinir ağı içindeki süreç elemanların bazıları, aynı toplama fonksiyonunu kullanırken, diğerleri ise başka fonksiyonları kullanabilir. Bu tamamen tasarımcının kendi öngörüsüne dayanarak aldığı karara bağlıdır<sup>47</sup>.

---

<sup>47</sup> Ercan Öztemel, a.g.e.,s.50.

- **Transfer(Aktivasyon)Fonksiyonu:** Transfer fonksiyonu, toplama fonksiyonundan gelen girdiyi işleyerek yapay sinir hücresinin çıkışı belirleyen fonksiyondur ve aşağıdaki şekilde gösterilebilir.

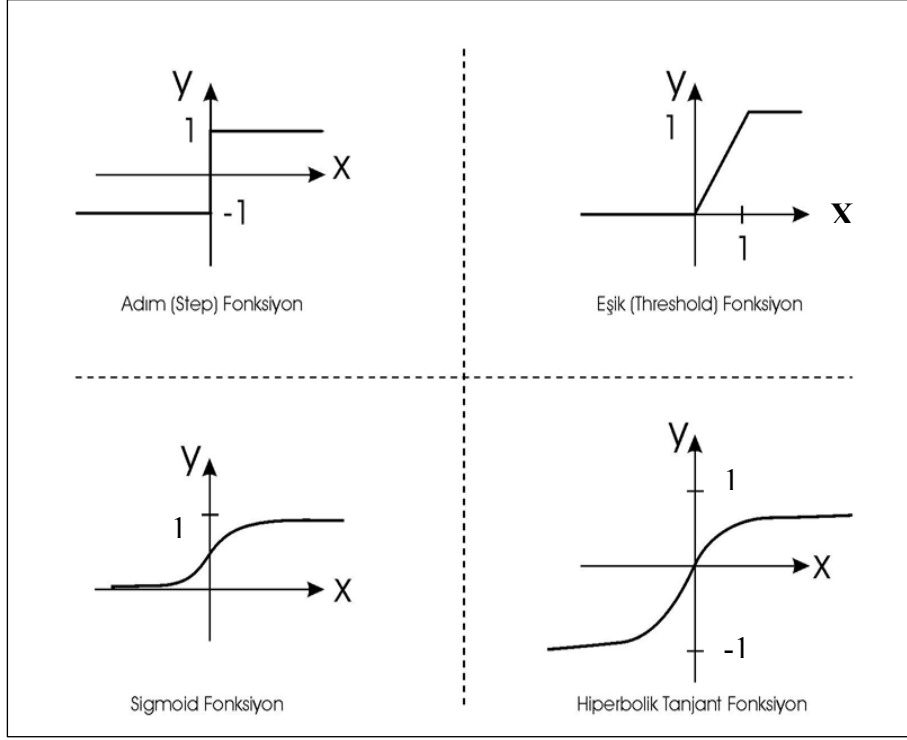
Bir çıktının hesaplanabilmesi için toplama fonksiyonunda olduğu gibi transfer fonksiyonunda da değişik formüller kullanılır. Uygun transfer fonksiyonunun seçimi, tasarımcının farklı fonksiyonları denemeleri sonucunda belirlenmektedir. Ayrıca fonksiyonun seçimi, yapay sinir ağı verilerine ve neyi öğrenmesinin istendiğine de bağlıdır. Çok katmanlı algılayıcı gibi bazı modeller transfer fonksiyonunun, türevi alınabilir bir fonksiyon olmasını şart koştuğu gibi bu fonksiyon genellikle doğrusal olmayan bir fonksiyondur. Doğrusal fonksiyonlarda, çıktı ile girdi arasındaki ilişki orantılı olduğundan ilk yapay sinir ağı denemelerinde doğrusal fonksiyonun başarısızlıkla sonuçlanmasının temel nedenidir<sup>48</sup>.

Transfer fonksiyonu, toplama fonksiyonundan gelen girdiyi dönüştürerek istenilen değerler arasında sınırlandırır. Bu sınırlandırmalar transfer (aktivasyon) fonksiyonunun türüne göre genellikle  $[0,1]$  veya  $[-1,1]$  değerleri arasındadır. Bu değerler transfer fonksiyonunun değeri olarak ya dış ortama ya da başka bir sinir hücresine girdi olarak iletilir. Yapay sinir hücre modellerinde yaygın olarak kullanılan transfer (aktivasyon) fonksiyonları; **doğrusal**, **adım**, **eşik**, **sigmoid**(logaritmik sigmoid) ve **hiperbolik tanjant**(tanjant sigmoid) fonksiyonlardır<sup>49</sup>. En çok kullanılan transfer fonksiyonlarının bazıları şekil 2.4 'de grafik şeklinde gösterilmiştir.

---

<sup>48</sup> Hasan Yurtoğlu, Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler için Türkiye Örneği, DPT Uzmanlık Tezi, Ankara, 2005, s.17.

<sup>49</sup> Mark Beale- Martin Hagan - Howard Demuth ,“Neural Network Toolbox for Use with MATLAB”, User’s Guide. The Math Works Inc.,2012, s.2-3.



**Şekil 2.4.** En Çok Kullanılan Transfer Fonksiyonları.

Şimdi bu fonksiyonları simgesel olarak ifade edelim.

Doğrusal Fonksiyon:  $F(x) = x$

Adım Fonksiyonu:  $y=F(x)= \begin{cases} 1 & x>0 \\ -1 & x\leq 0 \end{cases}$

Eşik Fonksiyonu:  $y=F(x)= \begin{cases} 1 & x>0 \\ 0 & x\leq 0 \end{cases}$

Sigmoid Fonksiyonu:  $y=F(x)=\frac{1}{1+e^{-x}}$

Hiperbolik Tanjant Fonksiyonu:  $y=F(x)=\frac{e^x - e^{-x}}{e^x + e^{-x}}$  şeklindedir.

- **Çıktı:** Transfer fonksiyonundan elde edilen değer yapay sinir hücresinin çıktısını oluşturur. Bu çıktı değeri, başka bir yapay sinir hücresine girdi olarak veya dış ortama gönderilebilir.

## 2.7. Yapay Sinir Ağının Yapısı

Yapay sinir hücreleri bir araya gelerek yapay sinir ağlarını oluşturur ve sinir hücrelerinin bir araya gelmesi de rastgele olmaz. Genel olarak hücreler üç katman(girdi katmanı, ara(gizli) katmanlar, çıktı katmanı) halinde ve her katman içinde paralel şekilde bir araya gelerek ağı oluştururlar<sup>50</sup>.

**Girdi katmanı:** Girdi katmanı, dışarıdan girdileri alan nöronları(sinirleri) içerir. Girdi katmanındaki nöronların girdi değerleri üzerinde herhangi bir işlem olmaz. Sadece söz konusu katman girdi değerlerini bir sonraki katmana iletir. Bu nedenle, girdi katmanı bazı araştırmacılar tarafından ağların katman sayısına dahil edilmez.

**Ara(Gizli) katmanlar:** Gizli katman; ağın temel işlevini gören katmandır. Gizli katmanda bulunan her bir işlem elemanı; giriş sinyalinin, giriş katmanındaki ağırlıklar ile çarpıldıktan sonra toplamını alır ve bu toplamı transfer fonksiyonundan geçirerek çıkan sonucu diğer (varsa) gizli katmana ya da çıkış katmanına gönderir.

Gizli katman sayısı ve katmandaki nöron sayısı ağ tasarımcısının kontrolünde olup probleme göre değişir. Gizli katmanda gereğinden daha az nöron kullanılması, ağın mevcut çıktıların tahmin doğruluğunu azaltır. Gerektiğinden daha çok sayıda nöron kullanılması da yeni girdi değerleri için yapılacak çıkış tahminlerinin doğruluk oranını azaltır.

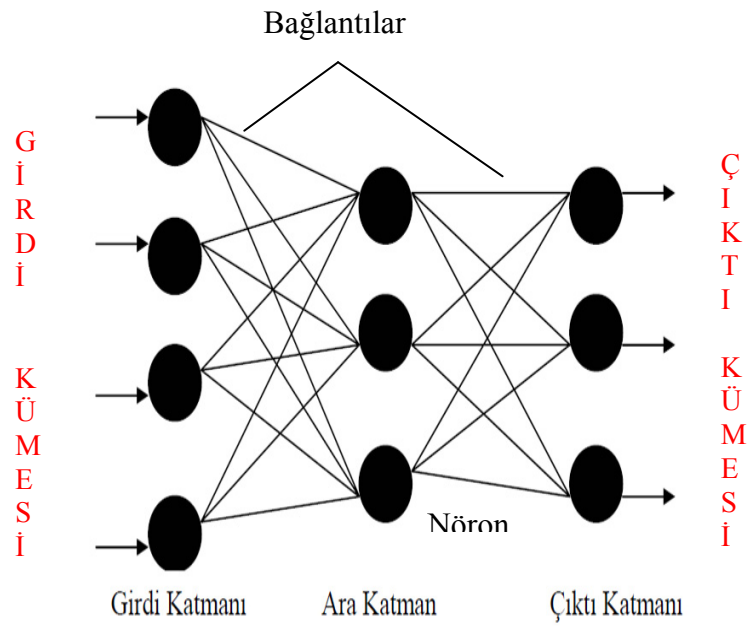
**Çıktı katmanı:** Çıktı(çıkış) katmanı; ağının en uç katmanıdır. Gizli katmandan aldığı veriyi ağın kullandığı fonksiyonla işleyerek ara katmandan gelen çıktıları dış ortama veya başka bir ağa ileten nöronları içeren katmandır. Çıkış katmanındaki nöron sayısı ağa sunulan her verinin çıkış sayısı kadardır. Bu katmandan elde edilen değerler yapay sinir ağının söz konusu problemin çıkış değerleridir.

---

<sup>50</sup> Ercan Öztemel, a.g.e.,s.52.



Girdi ve çıktı katmanları tek katmandan oluşur. İki katman arasında ise birden fazla ara katman bulunabilir. Bu ara katmanlar çok sayıda nöron içerir ve bu nöronlar ağ içindeki diğer katmanların nöronları ile bağlantılıdır. Çoğu ağ türünde, ara katmandaki bir nöron sadece bir önceki katmanın tüm nöronlarından sinyal alır. Nöron işlemini yaptıktan sonra ise çıktısını bir sonraki katmanın tüm nöronlarına gönderir. Aşağıda verilen Şekil 2.5'te üç katmanın her birinde bulunan nöronlar ve katmanlar arası ilişkiler bir yapay sinir ağı olarak gösterilmiştir.

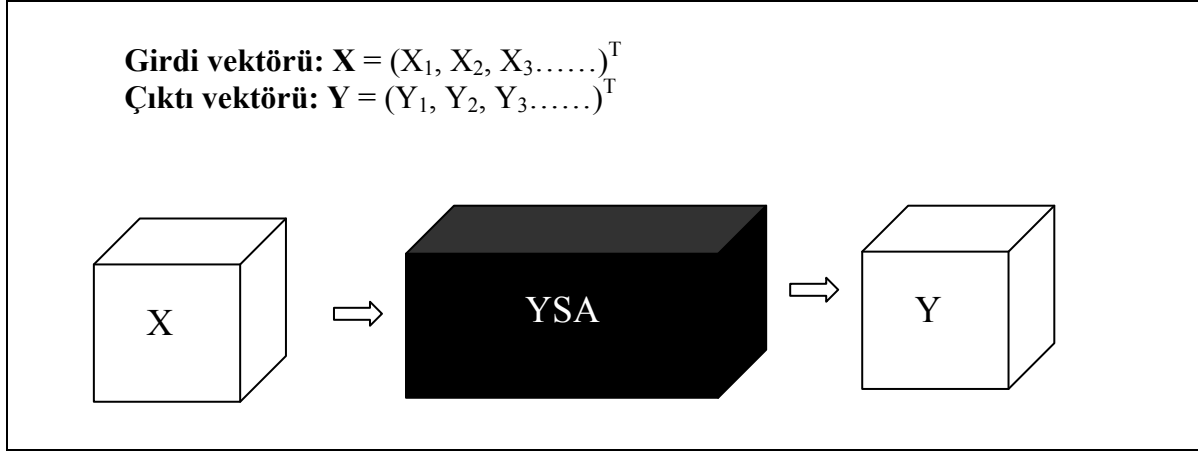


Şekil 2.5. Bir Yapay Sinir Ağ Örneği

## 2.8. Yapay Sinir Ağlarının Çalışması

Yapay sinir ağlarının genel çalışma ilkesi, bir girdi kümesini(setini) alarak onları çıktı kümesine çevirmek olarak açıklanabilir. Bunun için ağın kendisine gösterilen girdiler için doğru çıktıları üretecek hale gelmesi (yani eğitilmesi) gerekmektedir. Ağa sunulacak girdiler öncelikle bir vektör haline getirilerek ağa gönderilir ve ağ bu vektör için gerekli çıktı vektörünü üretir. Ağın parametre değerleri doğru çıktıyı üretecek şekilde düzenlenir. Bir yapay sinir ağı, herhangi bir girdi vektörünü nasıl dönüştürdüğü konusunda bilgi vermez. Mühendislik açısından bakıldığında yapay sinir ağları “kara kutu” gibi görülebilir. Kara kutu, dışarıdan bilgileri alır, dışarıya ürettiği çıktıları verir. İçeride ne olduğu ise bilinmemektedir. Diğer bir deyişle yapay sinir ağının sonuçları nasıl oluşturduğunu

açıklama yeteneği yoktur. Bu durum ağa olan güveni azaltsa da başarılı uygulamalar yapay sinir ağlarına olan ilgiyi sürekli artırmaktadır<sup>51</sup>



**Şekil 2.6.** Bir Yapay Sinir Ağı, girdi-çıkı ilişkisi

### 2.9. Yapay Sinir Ağlarında Eğitim, Öğrenme ve Test Etme

Yapay sinir ağında katmanlar arasındaki nöronların bağlantı ağırlık değerlerinin belirlenmesi işlemine “*ağın eğitilmesi*” denir. Başlangıçta rastgele atanan bu ağırlık değerleri, ağa verilen örneklerle değiştirilmektedir. Yapay sinir ağlar, kendilerine örnekler verildikçe bu ağırlık değerleri yenilenerek amaca ulaşılmaya çalışılır. Amaç ise, ağa verilen örnekler için doğru çıktıları üretecek ağırlık değerlerinin belirlenmesidir. Yapay sinir ağının eğitilmesinde kullanılan girdi ve çıktı çiftlerinden oluşan verilerin tümüne “*eğitim kümesi*” adı verilmektedir. Eğitim boyunca aynı eğitim kümesi defalarca ağa verilerek ağırlıkların en uygun düzeye gelmesi sağlanır. İterasyon (epoch) kelimesi, tüm eğitim kümesinin ağa verilmesi olarak ifade edilir. Eğer yapay sinir ağ verilen girdi çıktı çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır.

Yapay sinir ağında ağırlıkların doğru değerlere ulaşması, örneklerin temsil ettiği problem konusunda ağın genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleme, yapay sinir ağının eğitiminde kullanılmamış, ancak aynı evrenden gelen girdi-

<sup>51</sup> Ercan Öztemel, a.g.e.,s.54.

çıktı örneklerini doğru sınıflandırabilme yeteneği olarak tanımlanır. Ağın bu genelleştirme özelliğini taşıma işlemine “*ağın öğrenmesi*” denilir.

Yapay sinir ağlarında öğrenme iki aşamada gerçekleşir. Birinci aşamada ağa gösterilen örnek için ağın üreteceği çıktı belirlenir. Bu çıktı değerinin doğruluk derecesine göre, ikinci aşamada ağın bağlantılarının ağırlıkları değiştirilir. Ağın çıktısının belirlenmesi ve ağırlıkların değiştirilmesi öğrenme kuralına bağlı olarak farklı şekillerde olur. Yapay sinir ağında ağırlık değişimi yoksa öğrenme işlemi durmuş demektir.

Yapay sinir ağının eğitim sürecindeki hata düzeyi, test sürecindeki hata düzeyine göre bariz farklılıklar gösterdiğinde ezberleme sorunu ile karşılaşmış olunur. Bir bakıma genellemenin iyi gerçekleşmediğini ve girdi-çıkı eğrisinin düzgün olmadığını gösterir. Aşağıda belirteceğimiz gibi verilerin ezberlenmiş olması yapay sinir ağı için istenmeyen bir durumdur. Verileri ezberleyen ağa ilişkin hata miktarı, eğitim verilerinde azalmaya, test verilerinde ise artmaya yönelir. Dolayısıyla verileri ezberleyen ağ, gerçek hayattaki örüntüyü(şekli) iyi temsil etmediğinden kullanılamaz.

Bir yapay sinir ağının eğitiminin tamamlanmasının ardından, ağın öğrenip öğrenmediğini (performansını) ölçmek için denemeler yapılarak ağın test edilmesi gerekmektedir. Bir ağı test etmek için ağın eğitimi sırasında görmediği, yani veri kümesinden test amaçlı olarak ayrılan örnekler kullanılır ve bu örnekler “test kümesi” adını alır. Test işleminde ağın ağırlık değerleri değiştirilmemektedir. Örnekler ağa gösterilmekte ve ağ eğitimi sırasında belirlenen ağırlık değerlerini kullanarak daha önce görmediği bu örnekler için çıktılar üretmektedir. Elde edilen çıktılar doğruluk dereceleri, ağın öğrenmesi hakkındaki bilgiyi vermektedir. Sonuç ne kadar iyi olursa eğitimin performansı da o kadar iyi demektir<sup>52</sup>.

Sonuç olarak sinir ağı, ele alınan örneklerden öğrenir ve örnekleme verileri üzerindeki genelleştirme yeteneğini kullanır. Bir sinir ağının gücü, örnekleme verilerinden ne kadar iyi genelleştirme yapabildiğine bağlıdır. Öğrenme algoritması; istenen çıkış

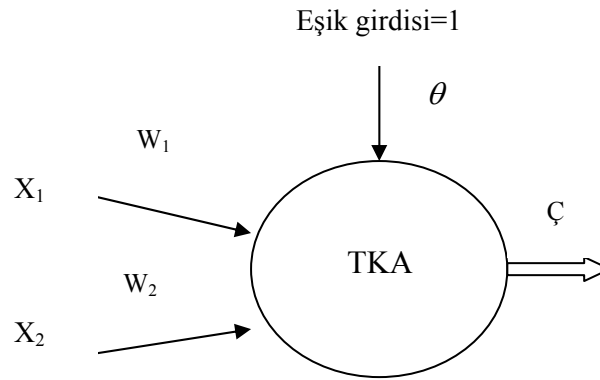
---

<sup>52</sup> Ercan Öztemel, a.g.e.,s.55.

vektörü ile gerçekleşen çıkış vektörü arasındaki farkı göz önüne alarak eğitim kümesi üzerindeki ortalama hatayı azaltacak şekilde, ağ içindeki ağırlıkları dengeler. Bu doğru olarak gerçekleştirildiğinde sinir ağı, yeni giriş verileri için istenen sonuçları doğru olarak tahmin eder.

## 2.10. Tek Katmanlı Algılayıcılar

Tek katmanlı yapay sinir ağları sadece girdi ve çıktı katmanlarından oluşur. Her ağın bir veya daha fazla girdisi ve tek bir çıktısı vardır. Çıktı üniteleri bütün girdi ünitelerine(X) bağlanmaktadır ve her bağlantının bir ağırlığı vardır. En basit tek katmanlı yapay sinir ağı, Şekil 2.7 de görüldüğü gibi iki girdili ve tek çıktılı ağıdır.



**Şekil 2.7** İki Girdi ve Bir Çıktıdan Oluşan En Basit Tek Katmanlı Yapay Sinir Ağı Modeli

Bu ağlarda süreç elemanlarının değerlerinin ve dolayısıyla ağın çıktısının sıfır olmasını önleyen ve değeri daima 1 olan bir de eşik değeri vardır. Ağın çıktısı, ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanıp transfer fonksiyonundan geçirilerek elde edilir. Bu işlem aşağıda gösterildiği şekilde formüle edilir.

$$y = f\left(\sum_{i=1}^n w_i x_i + \theta\right)$$

Burada;  $w_i$  ağırlıkları,  $x_i$  girdileri ve  $\theta$  ise eşik değerini göstermektedir. Tek katmanlı algılayıcıların çıktı fonksiyonu doğrusal bir fonksiyondur. Ağın çıktısı 1 veya -1 değerlerini alır ve bu değerler sınıfları temsil eder. Eğer ağın çıktısı 1 ise birinci bölgeye (sınıfa), -1 ise ikinci bölgeye (sınıfa) kabul edilmektedir.

$$F(x) = \begin{cases} 1 & \text{Eğer } y > 0 \text{ ise} \\ -1 & \text{Eğer } y < 0 \text{ ise} \end{cases}$$

İki sınıfı birbirinden ayırmak için kullanılan doğru, sınıf ayracı olarak da tanımlanır. Bu doğru aşağıdaki şekilde tanımlanır.

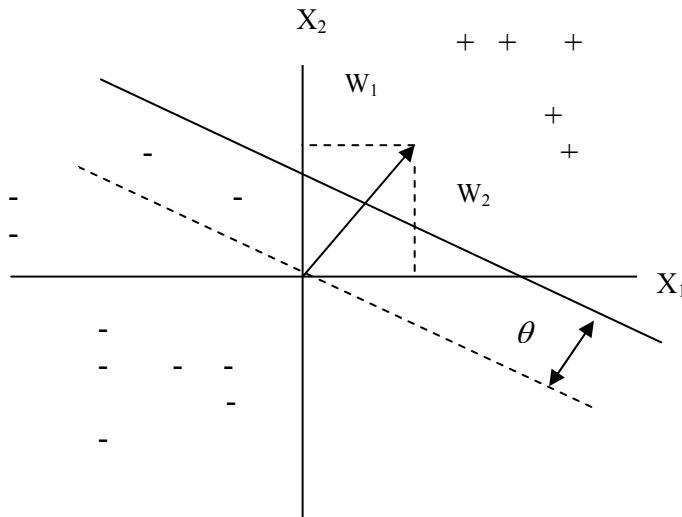
$$W_1 \cdot X_1 + W_2 \cdot X_2 + \theta = 0$$

Buradan;

$$X_1 = - (W_2/W_1) \cdot X_2 - \theta/W_1$$

$$X_2 = - (W_1/W_2) \cdot X_1 - \theta/W_2$$

olarak hesaplanır. Bu iki eşitlik kullanılarak sınıfları birbirinden ayıran doğru çizilir. Bu doğrunun geometrik gösterimi Şekil 2.8'deki gibidir<sup>53</sup>.



**Şekil 2.8** Ağırlıkların ve sınıf ayracı olan doğrunun geometrik gösterimi

<sup>53</sup> Ben Kröse - Patrick van der Smagt, **An Introduction to Neural Networks**, The University of Amsterdam, 1996, s.24.

Tek katmanlı yapay sinir ağlarında öğrenmeden istenen, sınıfları birbirinden ayıran en uygun doğrunun bulunmasıdır. Bunun için ağırlık değerlerinin değiştirilmesi gerekir. Ağırlıkların değiştirilmesi doğrunun eğiminin değiştirilmesidir. Yani t zaman biriminde ağırlık değerleri  $\Delta w$  kadar değiştirilir ise;

$$W_i(t+1) = W_i(t) + \Delta W_i(t) \quad \text{olur.}$$

Öğrenme sırasında bu değişim her iterasyonda gerçekleştirilerek doğrunun en uygun eğimi bulunmaya çalışılır. Ancak yapılan bu işlem yeterli olmayabilir. Bu nedenle eşik değerinin de değiştirilmesi gerekir. Bu işlem, doğrunun sınıflar arasında kaymasına yardımcı olmaktadır. Böylece transfer fonksiyonunun konumu belirlenmiş olur. Bu durumda t süresinde eşik değeri de;

$$\theta(t+1) = \theta(t) + \Delta \theta(t)$$

şeklinde değiştirilmektedir. Öğrenme sırasında ağırlıklarda olduğu gibi eşik değeri de her iterasyonda  $\Delta\theta$  kadar değiştirilmektedir.

Tek katmanlı yapay sinir ağlarında önemli iki modelden söz edilebilir. Bu modeller algılayıcı modeli ve ADALINE/MADALINE modelidir. Şimdi sırasıyla bunları açıklamaya çalışalım.

### 2.10.1. Basit Algılayıcı (Perceptron) Model

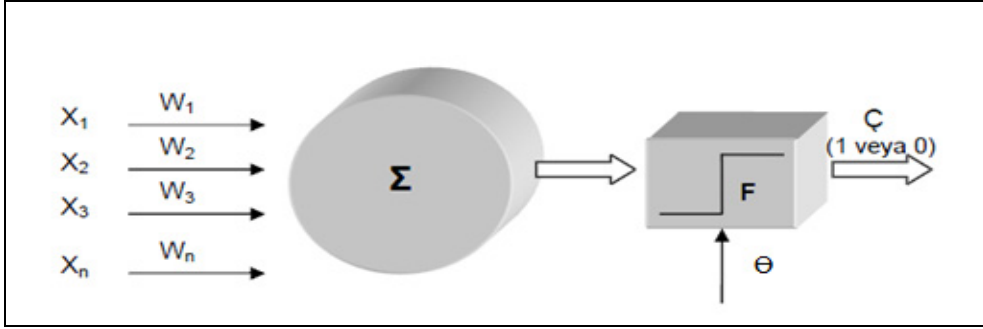
Basit algılayıcı model (perceptron), F. Rosenblatt tarafından 1958 yılında ortaya atılmış bir modeldir. Bu model örüntü(şekil) sınıflandırma amacı ile geliştirilmiştir.

Basit algılayıcı model, sunulan girdi vektörüne karşılık doğru çıktı vektörünün, hataların azaltılması yönünde ağırlıkların güncellenerek eğitimi ile elde edildiği tek katmanlı ve tek hücreden oluşan ağdır. Basit algılayıcılar özellikle örüntü tanımlama problemlerinde kullanıldığından dolayı bu ağlara algı anlamına gelen “perception” denilmektedir.<sup>54</sup>

---

<sup>54</sup> Leo Veelentruf, **Analysis and Applications of Artificial Neural Networks**, Prentice Hall, London, 1995, s.10.

Basit algılayıcı modeli, yapay sinir ağının öğrenilebilir niteliğini taşıyan ilk modelidir. Bu model basit algılayıcı tekrarlı öğrenme algoritmasıdır ve çözümün varlığı durumunda yakınsama niteliği taşır. Bu da basit algılayıcı modelinin en önemli özelliklerinden birisidir<sup>55</sup>. Aşağıda verilen Şekil 2.9 bir basit algılayıcı modelin yapısını göstermektedir<sup>56</sup>.



Şekil 2.9 Bir basit algılayıcı yapısı

Basit algılayıcı model, bir yapay sinir hücresinin birden fazla girdiyi alarak tek bir çıktı üretmesi temeline dayanır. Ağın çıktısı 1 veya 0 olmakta ve çıktı değerinin hesaplanması eşik değer fonksiyonu kullanılarak yapılır. Basit algılayıcı model, eğitilebilen tek bir işlemci elemandan oluşmaktadır. İşlemci elemana girdi değerleri ve bu değerlere karşılık gelen çıktı değerleri gösterilerek, ileriki kısımda açıkladığımız öğrenme kuralı olan Hebb öğrenme kuralına göre ağın çıktı değeri hesaplanır. Olması gereken çıktı değerine ulaşılan kadar ağırlıklar ve eşik değerleri değiştirilir.

Rosenblatt tarafından geliştirilmiş ve tek katmanlı algılayıcıların eğitiminde kullanılan algoritma aşağıda verilmiştir.

**Adım 1:** Yapay sinir ağına girdi değerleri ve ona karşılık gelen çıktı değerinin gösterilmesi. Birden fazla girdi değeri olabilmekte ve buna karşılık gelen çıktı değeri 1 ve 0 değerlerinden birisini alır.

**Adım 2:** Basit algılayıcı modeline gelen net girdi,

<sup>55</sup> Laurene Fausett, a.g.e.,s.59.

<sup>56</sup> Ercan Öztemel, a.g.e.,s.61.

$$\text{Net girdi} = \sum_{i=1}^m w_i x_i$$

formül ile hesaplanır.

**Adım 3:** Basit algılayıcı modelin çıktısının hesaplanması. Ayrıca net girdi, eşik değeri ile kıyaslanır ve eğer net girdi eşik değerinden büyükse çıktı 1, küçükse 0 değerini alır.

$$y = \begin{cases} 1 & \text{Eğer Net} > \theta \text{ ise} \\ 0 & \text{Eğer Net} \leq \theta \text{ ise} \end{cases}$$

Öte yandan gerçekleşen çıktı ile beklenen çıktı aynı değeri alırlarsa, ağırlıklarda herhangi bir değişiklik olmaz. Bu durum sınıf ayracının doğru pozisyonda olduğunu gösterir. Eğer farklı bir çıktı üretilirse bu durumda ağırlıklar değiştirilerek sınıf ayracının pozisyonu değiştirilmelidir. Böyle bir değişim söz konusu olduğunda ortaya aşağıda açıklayacağımız iki durum ortaya çıkar.

- Eğer ağırlık beklenen çıktı değeri 0, fakat gerçekleşen çıktı değeri 1 ise, ağırlık değerleri azaltılır. Bunun için gerekli olan aşağıdaki vektörü yazabiliriz.

$$W_n = W_0 - \lambda X$$

Burada  $\lambda$ , öğrenme katsayısını gösterir. Bu katsayı ağırlıkların değişim miktarlarını belirlemekte ve sabit bir değer olarak alınmaktadır.  $W_n$  yeni ağırlık vektörünü,  $W_0$  ise eski ağırlık vektörünü gösterir. Bu formül kullanılarak ağırlıklar girdi değerlerinin belirli bir oranında azaltılır.

- Eğer ağırlık beklenen çıktı değeri 1, fakat gerçekleşen çıktı değeri 0 ise, ağırlık değerleri artırılır. Bunun için kullanılan vektörel eşitlik ise;

$$W_n = W_0 + \lambda X \text{ dir.}$$



**Adım 4:** Yukarıda verilen ilk üç adım, bütün girdi kümesindeki örnekler için doğru sınıflandırmalar yapılınca kadar tekrarlanır.

### 2.10.1.1. Basit Algılayıcı Modelin Sınırları

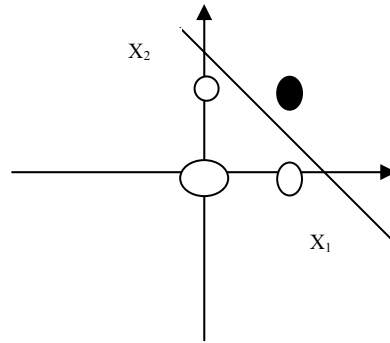
Basit algılayıcı modeli mantıksal fonksiyonları çözme amacıyla geliştirilmiştir. Ancak çoğu problemlerin çözümü için başarılı olamamıştır. Dolayısıyla 1980'li yılların ortalarında çok katmanlı algılayıcılar ve bunlar için öğrenme algoritması geliştirilmiştir. Basit algılayıcılar, doğrusal olmayan bir ilişkiyi ortaya koyan problemi çözememektedir. Doğrusal olmayan problemlere XOR problemi de denir. Bu konu çalışmamızın üçüncü bölümünde açıklanacaktır.

Basit algılayıcı model VE, VEYA, VE DEĞİL olarak isimlendirilen bazı mantıksal fonksiyon problemlerini çözerken XOR problemini çözemez<sup>57</sup>. Şimdi fonksiyonlar şeklinde ifade edilen problemleri kısaca açıklayalım.

VE fonksiyonunun matris yapısı aşağıda verilmiştir.

#### 1. VE Fonksiyonu

Girdi		Çıktı
X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	0
1	0	0
1	1	1



Şekil 2.10. VE Probleminin doğrusal bölünebilirliği

Şekil 2.10'da sıfır çıktısını veren girdi değerleri içi boş daire ile 1 çıktısını veren girdi değerleri ise içi dolu daire ile gösterilmiştir. Grafikten de görüldüğü gibi bu iki farklı sınıf doğrusal olarak bölünebilmektedir.

<sup>57</sup> Coşkun Hamzaçebi, **Yapay sinir Ağları**, Ekin basımevi, Bursa, 2011, s.34-36.

$$w_1 * 0 + w_2 * 0 + \theta < 0$$

$$w_1 * 0 + w_2 * 1 + \theta < 0$$

$$w_1 * 1 + w_2 * 0 + \theta < 0$$

$$w_1 * 1 + w_2 * 1 + \theta \geq 0 \text{ ise}$$

$$\theta < 0,$$

$$w_2 < -\theta,$$

$$w_1 < -b,$$

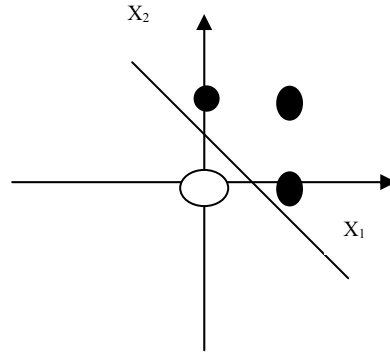
$$w_1 + w_2 \geq -\theta$$

Dolayısıyla doğrusal bir fonksiyon kullanan algılayıcı bu problemi çözebilir. Ayrıca yukarıdaki denklem sistemi bu problemin doğrusal bir çözümünün mümkün olduğunu göstermektedir.

VEYA fonksiyonunun matris yapısı aşağıda verilmiştir.

## 2.VEYA Fonksiyonu

Girdi		Çıktı
X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	1
1	0	1
1	1	1



Şekil 2.11. VEYA Probleminin doğrusal bölünebilirliği

$$w_1 * 0 + w_2 * 0 + \theta < 0$$

$$\theta < 0$$

$$w_1 * 0 + w_2 * 1 + \theta < 0$$

$$w_2 \geq -\theta$$

$$w_1 * 1 + w_2 * 0 + \theta < 0$$

$$w_1 \geq -b$$

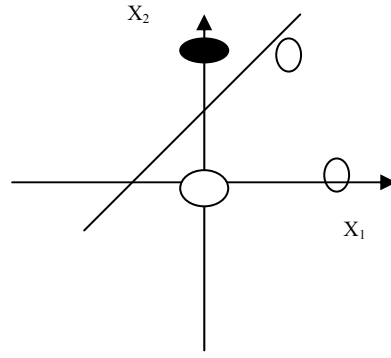
$$w_1 * 1 + w_2 * 1 + \theta \geq 0 \quad \text{ise} \quad w_1 + w_2 \geq -\theta$$

Şekil 2.11’de ve denklem sistemlerinde görüldüğü üzere VEYA problemi de doğrusal olarak bölünebilen bir problemdir. Algılayıcı bu problem için de uygun ağırlık değerlerini belirleyebilir.

VE DEĞİL fonksiyonunun matris yapısı aşağıda verilmiştir.

### 3.VE DEĞİL Fonksiyonu

Girdi		Çıktı
X <sub>1</sub>	X <sub>2</sub>	Y
1	1	0
1	0	1
0	1	0
0	0	0



Şekil 2.12. VE DEĞİL Probleminin doğrusal bölünebilirliği

$$w_1 * 0 + w_2 * 0 + \theta < 0 \quad \theta < 0$$

$$w_1 * 0 + w_2 * 1 + \theta < 0 \quad w_2 < -\theta$$

$$w_1 * 1 + w_2 * 0 + \theta \geq 0 \quad w_1 \geq -\theta$$

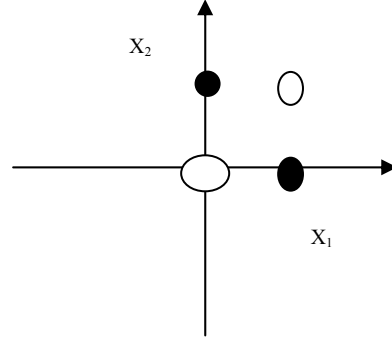
$$w_1 * 1 + w_2 * 1 + \theta \leq 0 \quad \text{ise} \quad w_1 + w_2 \geq -\theta$$

Şekil 2.12’de ve denklem sistemlerinde görüldüğü üzere VE DEĞİL problemi de doğrusal olarak bölünebilen bir problemdir.

XOR fonksiyonunun matris yapısı aşağıda verilmiştir.

#### 4. XOR Fonksiyonu

Girdi		Çıktı
X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	1
1	0	1
1	1	0



Şekil 2.13. XOR Probleminin doğrusal olarak bölünemezliği

$$w_1 * 0 + w_2 * 0 + \theta < 0 \quad \theta < 0$$

$$w_1 * 0 + w_2 * 1 + \theta \geq 0 \quad w_2 \geq -\theta$$

$$w_1 * 1 + w_2 * 0 + \theta \geq 0 \quad w_1 \geq -\theta$$

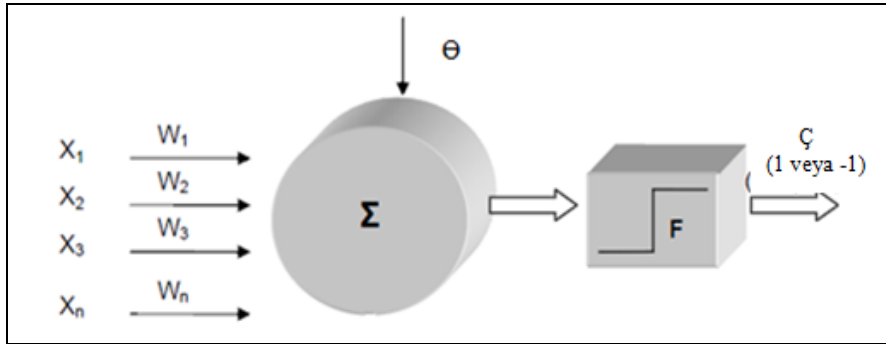
$$w_1 * 1 + w_2 * 1 + \theta \leq 0 \quad \text{ise} \quad w_1 + w_2 \leq -\theta$$

Şekil 2.13’de görüldüğü gibi bu durumda sınıflandırma yapabilmek için bir doğru fonksiyonunun oluşturulması işe yaramamaktadır. Ayrıca denklem sistemlerinden de anlaşılacağı üzere XOR probleminin doğrusal bir çözümü yoktur. Burada verilen ikinci ve üçüncü denklem eşitsizliklerin dördüncü eşitsizlikle uyumlu olmadığı açıkça görülmektedir.

#### 2.10.2. Adaline / Madaline Modeli

ADALINE, Widrow ve Hoff tarafından 1959 yılında geliştirilmiştir. Uyarlanan Doğrusal Eleman (Adaptive Linear Element)’in kısaltmasıdır. ADALINE genel olarak ADALINE model olarak adlandırılan tek bir işlemci elemandan oluşur. Öte yandan adından da anlaşılacağı gibi ADALINE, doğrusal sınıflandırma problemlerini çözmek

amacıyla geliştirilmiştir. Bu nedenle doğrusal transfer fonksiyonu kullanılmaktadır. ADALINE modelinin yapısı basit algılayıcı modeline benzemektedir. İki arasındaki temel fark kullandıkları öğrenme kurallarının farklılığından kaynaklanmaktadır. ADALINE modeli, en küçük kareler ortalaması yöntemine dayanmaktadır. Öğrenme kuralı, ortalamanın kare hatasını minimize etme ilkesine dayanır. Bu da ağı çıktısının beklenen çıktı değerine göre hatasını en aza indirecek şekilde ağı ağırlıklarının değiştirilmesidir. Aşağıda verilen Şekil 2.14 bir ADALINE model yapısını göstermektedir.



Şekil 2.14. Bir ADALINE model yapısı

Tek katmanlı ve doğrusal transfer fonksiyonuna sahip tek çıktılı bir yapay sinir ağı için çıktı değeri şu şekilde verilir.

$$y = \sum_j w_j x_j + \theta$$

ile elde edilir. Burada  $x_j$  girdi;  $w_j$  her girdinin ağırlık değeri;  $\theta$  ise ağ çıktısının 0'dan farklı bir değer olmasını sağlayan eşik değeridir.

Beklenen çıktı  $d$  olduğunda hata değeri  $E$ ,

$$E = d - y$$

dir. En küçük kareler yöntemine dayalı delta öğrenme kuralında, en küçük hata fonksiyonunu verecek ağırlıklar dereceli azaltma (*gradient descent*) yöntemi ile bulunur. Bu yöntemle göre, her bir ağırlık değerinin hataya bağlı kısmi türevlerinin negatif değerleri kullanılarak hata değeri düşürülür.

ADALINE ağının çıktı değerini üreten transfer fonksiyonu adım fonksiyonudur.  $y = \sum_j w_j x_j + \theta$  çıktı fonksiyonunun değeri 0'dan küçük ise çıktı -1; büyük ise +1 değerini alır. ADALINE modelinde herhangi bir anında hatayı azaltmak için kullanılan kural,

$$w_i(t) = w_i(t-1) + \lambda E x_i$$

eşitliği ile gösterilir. Burada  $w_i(t)$  ağırlığın t zamanındaki yeni değerini,  $w_i(t-1)$  ağırlığın değişmeden önceki değerini,  $\lambda$  öğrenme katsayısını, E beklenen değer ile çıktı arasındaki hatayı ve  $x_i$  de girdi değerini göstermektedir. Benzer şekilde  $\theta$  eşik değeri de yine zaman içerisinde değiştirilerek olması gereken eşik değeridir. Bunun matematiksel ifadesi de

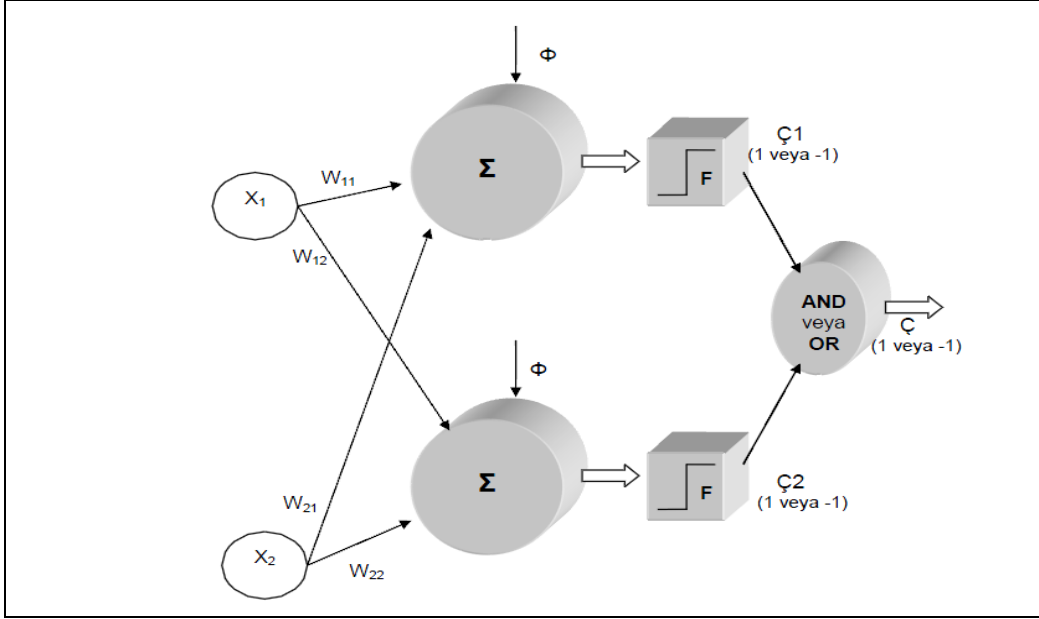
$$\theta_i(t) = \theta_i(t-1) + \lambda E$$

biçimindedir.

MADALINE ağları, birden fazla ADALINE modelinin bir araya gelerek oluşturdukları ağa verilen isimdir. MADALINE ağları genel olarak iki katmandan oluşmaktadır. Her katmanda değişik sayıda adaline modeli bulunmaktadır. Ağın çıktısı yine -1 ve +1 değerleri ile gösterilmektedir. MADALINE öğrenme kuralı adaline modelleri için kullanılan öğrenme kuralı ile aynıdır<sup>58</sup>. İki ADALINE modelinin birlikte kullanıldığı bir madaline ağı şekil 2.15 de verilmiştir.

---

<sup>58</sup> Laurene Fausett, a.g.e.,s.88.



**Şekil 2.15.** İki ADALINE ağından oluşan MADALINE ağı

Burada önemli konu şeklin son kısmında bulunan VE veya VEYA sonlandırıcısının ne anlama geldiğidir. Klasik mantık teorisinde olduğu gibi VE sonlandırıcısı olması durumunda bütün ADALINE ünitelerinin 1 değerini üretmesi sonucu MADALINE ağına çıktısının 1 olacağı, aksi halde -1 (veya 0) değerini alacaktır. VEYA sonlandırıcısı olması durumunda ise ADALINE modellerinin birisinin 1 üretmesi MADALINE ağına çıktısının 1 olması için yeterli olduğu yönündedir<sup>59</sup>.

## 2.11. Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları, hücrelerin birbirleri ile çeşitli şekillerde bağlanmalarından oluşur. Hücre çıkışları, ağırlıklar üzerinden diğer hücrelere ya da kendisine girdi olarak bağlanabilir. Ayrıca bu bağlantılarda gecikme birimi de kullanılabilir. Hücrelerin bağlantı şekillerine, öğrenme kurallarına ve transfer fonksiyonlarına göre çeşitli yapay sinir ağ yapıları geliştirilmiştir. Ağın yapısını her bir sinir hücresi arasındaki bağlantıların yapısı belirler. Bağlantıların istenilen hedefe ulaşmasını belirleyen ise öğrenme algoritmasıdır.

Yapay sinir ağları, mimari yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar. Şimdi bunları sırasıyla açıklayalım.

<sup>59</sup> Ercan Öztemel, a.g.e.,s.73.

### 2.11.1. Yapay Sinir Ağların Mimari Yapılarına Göre Sınıflandırılması

Yapay sinir ağları, mimari yapılarına göre ileri beslemeli (feedforward) ve geri beslemeli (feedback) ağlar olmak üzere iki şekilde sınıflandırılırlar.

- **İleri Beslemeli Yapay Sinir Ağları**

İleri beslemeli yapay sinir ağları katmanlar şeklinde düzenlenir. En basit şekliyle katmanlı ağın çıkış katmanına girişleri yansıtan bir giriş katmanı ve bunları işleyen bir çıkış katmanı yer alır ve bu ağlar ileri yönde işler. Bu tür ağlara tek katmanlı ağlar adı verilir. Buradaki tek katman çıkış katmanıdır. Giriş katmanında ise herhangi bir işlem yapılmadığından bu katman sayılmaz<sup>60</sup>.

1969 yılında Minsky ve Papert bir algılayıcının kısıtlamaları ve yeteneklerini ele aldıkları eserlerinde tek katmanlı algılayıcının iki durumlu XOR problemini çözemediğini belirtmişlerdir<sup>61</sup>. Gerçekten de tek katmanlı sınıflandırıcıların çözemediği pek çok türde problem vardır. Bu nedenle *çok katmanlı ileri beslemeli ağlar* ortaya çıkmıştır. Bu ağlarda giriş ve çıkış katmanları arasında bir veya daha fazla gizli(ara) katman bulunur ki bu katmandaki düğümler gizli nöron olarak adlandırılır.

İleri Beslemeli yapay sinir ağlarında nöronlar arasındaki iletişim, giriş katmanından çıktı katmanına doğru tek yönlü bağlantılarla iletilir. Nöronlar bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisindeki nöronlar birbiriyle bağlantılı değildir. Bu nedenle ileri beslemeli yapay sinir ağlarında, nöronlar arasındaki bağlantılar bir döngü oluşturmamakta ve bu ağlar girilen verilerle hızlı bir şekilde çıktı üretebilmektedirler.

İleri beslemeli yapay sinir ağlarında bir katmandaki tüm hücrelerin çıktıları bir sonraki katmandaki tüm hücrelere giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan ara katmandaki hücrelere iletir. Bilgi ara ve çıktı

---

<sup>60</sup> Simon Haykin, a.g.e.,s.43.

<sup>61</sup> Graupe, Daniel, **Principles of Artificial Neural Networks**, 2nd Edition. World Scientific Publishing Co. Pte. Ltd.:Singapore,2007, s.23.



katmanında işlenerek ağın çıktısı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir<sup>62</sup>.

İleri beslemeli yapay sinir ağları, öğrenme algoritması olarak geri yayılım öğrenme algoritmasını kullanır. Şekil tanıma, sinyal işleme, sınıflandırma gibi problemlerde genellikle bu ağ yapısı uygulanır.

Rosentblatt'ın basit algılayıcı modelini ortaya atmasından sonraki yıllarda çok katmanlı ileri beslemeli ağların birçok çeşidi önerilmiştir. Bunlardan bazıları ilerleyen bölümlerde anlatacağımız çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağlarıdır.

### • Geri Beslemeli Yapay Sinir Ağları

Geri Beslemeli veya yinelemeli yapay sinir ağları, ileri beslemeli ağların aksine dinamik yapılıdır. Geri beslemeli yapay sinir ağlarında, çıktı veya ara katmanlardaki nöronlar çıktılarını giriş veya önceki ara katmanlardaki nöronlara tekrar girdi olarak iletirler. Yani ileri beslemeli ağların aksine geri beslemeli ağlarda, giriş ve çıkış birimleri arasında döngü söz konusudur. Böylece bilgi hem ileri hem de geri yönde aktarılmış olmaktadır. Geri beslemeli yapay sinir ağları dinamik hafızalı olup bir andaki çıktı hem o andaki hem de önceki girdileri yansıtmaktadır. Geri beslemeli ağ mimarileri, genellikle danışmansız öğrenme kurallarının uygulandığı ağlarda kullanılmaktadır<sup>63</sup>.

Geri beslemeli yapay sinir ağlarında, en az bir işlemci elemanın çıktısı, kendisine ya da diğer işlemci elemanlara girdi olarak verilmekte ve genellikle geri besleme bir geciktirme elemanı (ara katman veya çıktı katmanındaki transfer değerlerini bir sonraki iterasyona girdi olarak taşımakla görevli eleman) üzerinden yapılmaktadır. Geri besleme, bir katmandaki işlemci elemanlar arasında olduğu gibi katmanlar arasındaki işlemci elemanlar arasında da olabilmektedir. Bu yapı nedeniyle geri beslemeli yapay sinir ağları, doğrusal olmayan dinamik bir davranış gösterirler. Bu sayede, geri beslemenin yapılış şekline göre farklı yapı ve davranışta geri beslemeli yapay sinir ağları elde edilebilir<sup>64</sup>. Geri beslemeli ağlara örnek olarak Hopfield, Elman ve Jordan ağları verilebilir.

---

<sup>62</sup> Tuğba, Saraç, a.g.e.,s.27.

<sup>63</sup> Çetin Elmas, a.g.e, s.63.

<sup>64</sup> Tuğba, Saraç, a.g.e.,s.28.

### 2.11.2. Yapay Sinir Ağların Öğrenme Yapılarına Göre Sınıflandırılması

Yapay sinir ağlarında öğrenme, başlangıçta rastgele olarak ağa atanan ağırlık değerlerini belirleme işlemidir. Öğrenme bir bakıma, ağa verilen örnekler yolu ile sağlanmaktadır. Ağ gördüğü her örnek için önce bilgi edinir ve bu işlem ağın doğru ağırlık değerlerine ulaşması, bir başka deyişle, örneklerin temsil ettiği olayla ilgili genelleme yapabilecek duruma gelmesi ile son bulur. Bu işlemin matematiksel karşılığı, birtakım yöntem, kural ve algoritmalar yardımıyla ağdaki işlemci elemanlar arasındaki bağlantı ağırlıklarının sürekli yenilenerek değiştirilmesidir. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Şimdi bu öğrenme kurallarını kısaca açıklamaya çalışalım.

#### • Danışmanlı Öğrenme

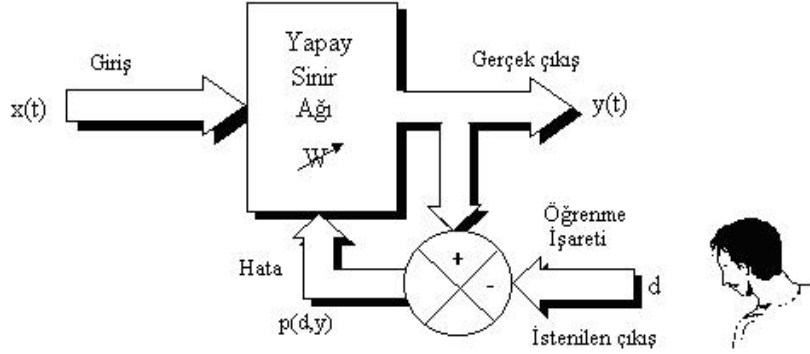
Danışmanlı öğrenme yöntemi, ileri beslemeli ağlarda daha sıklıkla kullanılan bir yöntemdir. Danışmanlı öğrenme yönteminde çalışma kümesi, verinin özellikleri ve gözlemlenebilir çıktılar hakkında bütün bilgileri içerir. Girdilerle çıktıların ilişkisi öğrenilerek modeller oluşturulur<sup>65</sup>.

Danışmanlı öğrenmede, ağa verilen girdi değerleri için çıktı değerleri verilir. İstenilen ile gerçek çıktı arasındaki farka (hataya) göre ağ, bağlantıların ağırlığını en uygun çıktının elde edilmesi için sonradan düzenleyebilir. Her iterasyonda, yapay sinir ağına örnek bir çıktı verilir ve bu çıktı ile ağın ürettiği çıktı karşılaştırılarak ağın hatası hesaplanır. Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplandıktan sonra, ağın yeni ağırlıkları bu hata miktarına göre düzenlenir. Hata miktarı hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her nörona düşen hata payı bulunur. Daha sonra her nöron kendine gelen ağırlıkları günceller. Bu hata minimum olana kadar ağ, nöronlar arasındaki ağırlıkları düzelterek iterasyona devam eder. Bu sebeple danışmanlı öğrenme algoritmasının bir öğretmene ya da danışmana ihtiyacı vardır. Şekil 2.16'da danışmanlı öğrenme yapısı gösterilmiştir<sup>66</sup>.

<sup>65</sup> Kate, A. Smith-Jatinder, N.D. Gupta, Neural Networks in Business: Techniques and Applications, IRM Press, United Kingdom, 2002, s.2.

<sup>66</sup> <http://cc.domaindx.com/hyuces/ysa/ogremealgrt.htm>.

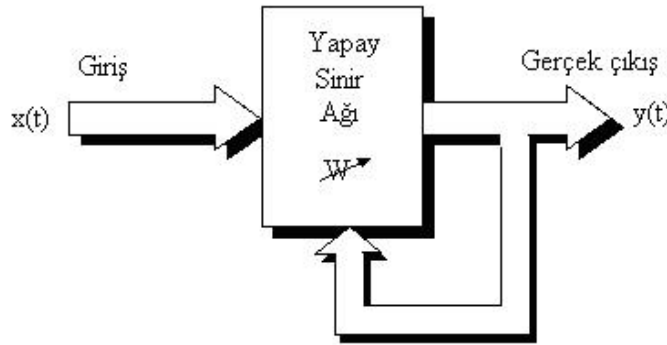
Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı ya da geri yayılım algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.



Şekil 2.16. Danışmanlı öğrenme yapısı

- **Danışmansız Öğrenme**

Danışmansız öğrenmede yapay sinir ağına sadece girdiler verilmekte, ulaşılması gereken beklenen çıktılar verilmemektedir. Girişte verilen örnek değerlere bakarak yapay sinir ağı, parametreler arasındaki ilişkileri kendi kendine öğrenir. Yani bu öğrenme algoritmalarında, istenilen çıktı değerinin bilinmesine gerek yoktur. Danışmansız öğrenme genellikle sınıflandırma problemlerinin çözümünde kullanılır. Aşağıda Şekil 2.17’de danışmansız öğrenme yapısı gösterilmiştir<sup>67</sup>.

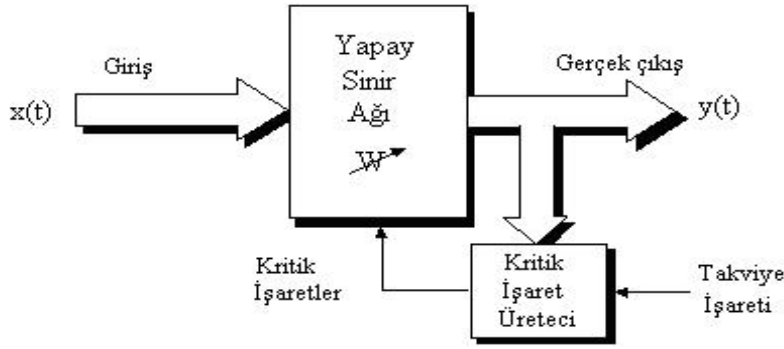


Şekil 2.17. Danışmansız öğrenme yapısı

<sup>67</sup> <http://cc.domainlx.com/hyuces/ysa/ogremealgrt.htm>.

- **Takviyeli Öğrenme**

Bu öğrenme kuralı danışmanlı öğrenmeye yakın bir yöntemdir. Takviyeli öğrenmede, danışmanlı öğrenmede olduğu gibi bir eğitici yardımcı olur. Farklı olarak sisteme sadece girdi değerleri verilir, beklenen çıktı değerleri verilmez. Ağın kendi çıktılarını oluşturması beklenir ve eğitici tarafından çıktı değerinin ne derece doğru olduğunu belirten bir sinyal, skor veya derece bildirilir. Ağ, eğiticiden aldığı sinyalle bağlantılarının ağırlık değerlerini değiştirerek öğrenme sürecini devam ettirir. Şekil 2.18’de takviyeli öğrenme yapısı gösterilmiştir<sup>68</sup>.



**Şekil 2.18.**Takviyeli öğrenme yapısı

## 2.12. Temel Öğrenme Kuralları

Yapay sinir ağları modellerinin eğitilmesinde kullanılan pek çok öğrenme algoritması bulunur. Bu algoritmalar; yapay sinir ağının şekline(mimarisine), karşılaşılan sorunun niteliğine göre farklılık gösterse de birçoğunun temel aldıkları öğrenme kuralı en eski öğrenme kuralı olan Hebb öğrenme kuralının bir varyasyonudur. Bu öğrenme algoritmalarının yüzden fazla çeşidi bulunmakla beraber temelde kullanılan; Hebb Kuralı, Hopfield Kuralı, Delta Kuralı ve Kohonen Kuralı’dır. Bunun dışında, araştırmacılar sürekli olarak yeni öğrenme kuralları geliştirmekte ve insanın öğrenmesine benzeyen çeşitli öğrenme kuralları geliştirmektedirler. Şimdi söz edilen kuralları kısaca açıklayalım.

<sup>68</sup> <http://cc.domainlx.com/hyuces/ysa/ogremealgrt.htm>.

- **Hebb Kuralı**

1949 yılında Kanadalı psikolog Donald Hebb tarafından biyolojik temele dayalı olarak geliştirilmiş olan Hebb algoritması en eski ve en ünlü öğrenme algoritmasıdır<sup>69</sup>. Hebb kuralı, diğer tüm öğrenme algoritmalarının temeli sayılır. Bu öğrenme algoritması basit bir mantığa dayanmaktadır. Buna göre, eğer bir nöronun girdi ve çıktısı aktif diğer bir ifade ile aynı işaretli ise, nöronlar arası ağırlıklar artar. Dolayısıyla nöronlar arasındaki ilişki güçlenir. Eğer çıktı aktif değil ve girdi aktif ise nöronlar arasında ağırlık azalır.

- **Hopfield Kuralı**

Hopfield Kuralı, zayıflatma ve kuvvetlendirme büyüklüğü dışında Hebb Kuralına benzer. Burada bağlantıların ne kadar kuvvetlendirilmesi veya zayıflatılması gerektiğinin belirlenmesi gerekir. Eğer girdi ve istenilen çıktının ikisi de aktifse, bağlantı ağırlığı öğrenme katsayısı kadar arttırılmakta, aksi durumda ise öğrenme katsayısı kadar azaltılmaktadır. Öğrenme katsayısı, genel olarak 0-1 arasında kullanıcı tarafından atanan sabit bir değerdir<sup>70</sup>.

- **Delta Kuralı**

Widrow ve Hoff tarafından geliştirilen bu kural Hebb Kuralının gelişmiş şeklidir. Delta Kuralına göre, yapay sinir hücresinin gerçek çıktısı ile beklenen çıktısı arasındaki farkı azaltmak için, yapay sinir ağlarının işlemci elemanları arasındaki bağlantı ağırlık değerleri sürekli olarak değiştirilmesi gerekmektedir. Bu kuralda, gerçek çıktı ile beklenen çıktı arasındaki farkların (hatanın) karelerinin ortalaması minimize edilmeye çalışılır.

Delta algoritması hataların karesinin en küçük olduğu noktayı bulurken dereceli azaltma yöntemini kullanmaktadır.<sup>71</sup> Dereceli azaltma yönteminde amaç, bağlantı ağırlıklarının hata kareleri toplamını minimize eden değerlerini belirlemektir. Bu yöntemde hata kareler toplamı her bir ağırlığa göre kısmi türevleri alınarak türevin negatif değerleri kullanılır. Böylece hatalar azaltılmış olur.

---

<sup>69</sup> Donald, O. Hebb, **The Organization of Behaviour**, New York,1949, s.44.

<sup>70</sup> John,J., Hopfield, “Neural Network and Physical Systems with Emergent Collective Computational Properties”, Proceedings of the National Academy of Science, Sayı 79, 1982, s.2554-2558.

<sup>71</sup> M.Caudill, Neural Network Primer Part 1, AI Expert, 1987, s.46-52

- **Kohonen Kuralı**

Teuvo Kohonen (1982) tarafından, biyolojik sistemlerdeki öğrenmeden esinlenilerek geliştirilen bir öğrenme kuralıdır. Bu kuralda işlemci elemanlar, ağırlıklarının ayarlanması için yarışmaktadır. Hebb kuralından farklı olarak bir seferde yalnız bir işlemci elemanın, yani yalnızca kazanan nöronun bağlantı ağırlıkları değiştirilmektedir. En uygun çıktılı işlemci elemanın kazandığı bu kuralda, kazanan işlemci eleman, kendisine komşu diğer işlemci elemanların ağırlıklarının değiştirilmesine de izin vermektedir. Sadece kazanan elemanın çıktı üretmesine ve komşu hücreleri ile birlikte ağırlıklarının değiştirilmesine izin verilir. Komşu sayısı eğitim süresince değişiklik gösterir. Kohonen kuralı hedef çıkışa gereksinim duymayıp danışmansız öğrenme ile eğitimi gerçekleştirilir<sup>72</sup>.

### 2.13. Yapay Sinir Ağı Tasarımı

Bir sinir ağı modelini oluşturmak için nöronların bağlanması sonucu oluşan topolojisi, işlemci elemanların kullandıkları toplama ve transfer fonksiyonları, kullanılacak öğrenme yöntemi, öğrenme kuralı ve algoritması belirlenmelidir. Eldeki veriye ve ağdan yapılmak istenen uygulamanın şekline göre model tasarlanır. Kurulan modelin başarısı modelin şeklinin doğru oluşturulması ile yakından ilgilidir. Bunun için YSA tasarımcısının, ağın yapısına ve işleyişine ilişkin aşağıdaki kararları vermesi gerekmektedir<sup>73</sup>.

- Ağ şeklinin seçimi ve yapısal özelliklerinin belirlenmesi (katman sayısı ve katmandaki işlemci eleman sayısı gibi),
- İşlemci elemanların kullandığı fonksiyonların karakteristik özelliklerinin belirlenmesi,
- Öğrenme algoritması ve parametrelerinin belirlenmesi,
- Eğitim ve test setinin(kümesinin) oluşturulması.

---

<sup>72</sup> Zekai Şen, **Yapay Sinir Ağları İlkeleri**, Su Vakfı Yayınları, İstanbul, 2004, s.98.

<sup>73</sup> Dildade Aşkın-İres İskender-Ali Mamızadeh, “Farklı Yapay Sinir Ağları Yöntemlerini Kullanarak Kuru Tip Transformatör Sargısının Termal Analizi”, **Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi**, Cilt:26, No: 4, 2011, s.905-913.

Bu kararların doğru verilmesi, yapay sinir ağının daha hızlı ve daha başarılı sonuçlar üretmesini sağlar. Şimdi yukarıda sıraladığımız maddeleri sırasıyla ele alalım.

- **Ağ Şeklinin Seçimi**

Yapay sinir ağlarının tasarımı sürecinde ağ yapısının seçilmesi, uygulama probleminde bağlı olarak seçilmelidir. Hangi problem için hangi ağın daha uygun olduğunun bilinmesi önemlidir. Kullanım amacı ve o alanda başarılı olan ağ modelleri aşağıdaki Tablo2.2 de verilmiştir<sup>74</sup>.

**Tablo2.2** Kullanım Amaçlarına Göre YSA Topolojileri

<b>Kullanım Amacı</b>	<b>Ağ Türü</b>	<b>Ağ Kullanımı</b>
Tahmin-Öngörü	<ul style="list-style-type: none"> <li>• Çok Katmanlı Algılayıcı(MLP)</li> <li>• Radyal Tabanlı Fonksiyon(RBF)</li> <li>• Elman Ağı</li> <li>• Jordan Ağı</li> </ul>	Ağın girdilerinden bir çıktı değerinin tahmin edilmesi
Fonksiyon Yaklaşırma	<ul style="list-style-type: none"> <li>• MLP</li> <li>• RBF</li> </ul>	Eldeki birçok veri çiftini işleyerek bunların oluşturduğu bilinmeyen fonksiyonu tahmin etmeye çalışması.
Desen(Örüntü) Sınıflandırma	<ul style="list-style-type: none"> <li>• LVQ</li> <li>• ART</li> <li>• Olasılık Tabanlı Sinir Ağları</li> <li>• Tek veya Çok Katmanlı Algılayıcı</li> <li>• RBF</li> <li>• Boltzmann Makinesi</li> </ul>	Girdilerin hangi sınıfa ait olduğunun belirlenmesi
Veri İlişkilendirme	<ul style="list-style-type: none"> <li>• Kohonen's SOM</li> <li>• Hopfield ağları</li> <li>• Boltzmann Makinesi</li> <li>• ART</li> </ul>	Girdilerin içindeki hatalı bilgilerin bulunması ve eksik bilgilerin tamamlanması
Kümeleme	<ul style="list-style-type: none"> <li>• LVQ</li> <li>• SOM</li> <li>• ART</li> </ul>	Ağa sunulan bilgileri sınıflandırır

<sup>74</sup> Dave Anderson- George McNeill, **Artificial Neural Networks Technology**, Rome Laboratory, A DACS State-of-the-Art Report, ELIN: A011, 1992, s.31.

Veri Filtreleme	<ul style="list-style-type: none"> <li>• Yeniden Dolaşım(Recirculation)</li> </ul>	Girdilerin içindeki hatalı Bilgilerin bulunması ve eksik bilgilerin tamamlanması
Optimizasyon	<ul style="list-style-type: none"> <li>• MLP</li> <li>• Olasılık Tabanlı Sinir Ağları</li> <li>• RBF</li> </ul>	Belirli kısıtlar altında bir fonksiyonun maksimum veya minimum noktalarının bulunması veya optimum çözümlerinin bulunmasıdır.

Yukarıda verilen Tablo2.2 hangi topolojinin hangi amaçlarla kullanıldığını göstermektedir. Tabloda tüm ağ topolojilerini değil, sadece çok kullanılan bazı topolojileri göstermektedir. Tabloda da görüldüğü gibi bazı ağlar, birden çok problem tipinin çözümünde kullanılmaktadır. Probleme uygun yapay sinir ağ topolojisinin seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına bağlıdır. Ağda kullanılacak öğrenme algoritması seçildiğinde, bu algoritmanın gerektirdiği şekilde zorunlu olarak seçilmiş olmaktadır. Örneğin, İleri beslemeli yapay sinir ağları, öğrenme algoritması olarak genellikle geri yayılım öğrenme algoritmasını kullanmakta ve bu nedenle bazen geri yayılım ağları olarak da adlandırılmaktadır.

- **Katman Sayısını Belirleme**

Yapay sinir ağ tasarımının bir diğer adımı ise, ağdaki katman sayısına karar vermektir. İşlemci elemanların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Katman sayısını belirlemenin en iyi yolu, birkaç deneme yapılarak ağın performansına bağlı olarak en uygun katman sayısına karar vermektir. Çoğu problem için iki veya üç katmanlı bir ağ istenilen sonuçları üretebilmektedir.

- **İşlemci Eleman(Nöron) Sayısını Belirleme**

Ağın yapısal özelliklerinden bir diğeri de her katmandaki işlemci eleman sayısının belirlenmesidir. Katmandaki işlemci elemanların sayısını belirlemenin en iyi yolu bir kaç deneme ile ağın performansına bağlı olarak en uygun işlemci sayısına karar vermektir. Bir yapay sinir ağında işlemci eleman sayısının olması gerekenden az sayıda olması



durumunda ađın öğrenme yeteneđi azalırken, geređinden çok sayıda işlemci elemanın olması durumunda ise genelleme yeteneđi azalmaktadır.

- **Fonksiyon Seçimi**

İşlemci elemanların karakteristik özelliklerinin belirlenmesi de yapay sinir ađının tasarımında alınacak önemli kararlardan birisidir. Toplama fonksiyonunun ve transfer fonksiyonunun seçimi büyük ölçüde verilerin özelliklerine ve ađdan öğrenilmesi istenen verinin türüne ve yapısına bađlıdır. Transfer fonksiyonları içinde en çok kullanılanlar, sigmoid, hiperbolik tanjant ve doğrusal fonksiyondur. Eđer ađın, bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid, ortalamadan sapmasını öğrenmesi isteniyorsa hiperbolik tanjant fonksiyonlarının kullanılması önerilir<sup>75</sup>.

- **Öğrenme Algoritmasının Seçimi**

Yapay sinir ađ yapısının seçiminden sonra uygulama başarısını belirleyen en önemli faktör öğrenme algoritmasıdır. Genellikle ađ yapısı öğrenme algoritmasının seçiminde belirleyicidir. Bu nedenle seçilen ađ yapısı üzerinde kullanılabilir öğrenme algoritmasının seçimi ađ yapısına bađlıdır. Yapay sinir ađının geliştirilmesinde kullanılacak çok sayıda öğrenme algoritması bulunmakla birlikte bunlar içinden bazı algoritmaların bazı tip uygulamalar için daha uygun olduđu bilinmektedir. Bu algoritmaların uygulama alanlarına göre sınıflandırılması Tablo 2.3 deki gibidir<sup>76</sup>.

**Tablo 2.3.** Öğrenme Algoritmaları ve Uygulandıkları Alanlar

Uygulama Tipi	Yapay Sinir Ađı
Öngörü Tanıma	<ul style="list-style-type: none"><li>• Geri yayılım</li><li>• Delta Bar Delta</li><li>• Geliştirilmiş Delta Bar Delta</li><li>• Yönlendirilmiş Rassal Tarama</li><li>• Geri Yayılım içinde Self Organizing Map</li></ul>
Sınıflandırma	<ul style="list-style-type: none"><li>• LVQ</li><li>• Olasılıklı Yapay Sinir Ađları</li></ul>

<sup>75</sup> Tuđba, Saraç, a.g.e.,s.35.

<sup>76</sup> Tuđba, Saraç, a.g.e.,s.33.

Veri İlişkilendirme	<ul style="list-style-type: none"> <li>• Hopfield</li> <li>• Boltzmann Makinesi</li> </ul>
Kümeleme	<ul style="list-style-type: none"> <li>• ART</li> <li>• SOM</li> </ul>

- **Öğrenme Katsayısı ve Momentumun Seçimi**

Öğrenme sürecinde öğrenme katsayısı veya oranı ( $\lambda$ ) ağırlıkların değişim miktarını belirler. Genellikle bu oran salınım (oscillation) neden olmayacak kadar büyük alınmaya çalışılır. Öğrenme oranı, ağırlıkların bir sonraki adımda hangi oranda değiştirileceğini belirler. Ağ üzerinde önemli bir etkisi bulunan öğrenme katsayısı uygulanan probleme göre değişir. Öğrenme katsayısı genellikle (0-1) aralığında belirlenir. Öğrenme katsayısının büyük seçilmesi salınım yol açar ve ağıın herhangi bir minimum değerine ulaşmasını zorlaştırır. Bu katsayının küçük bir değer seçilmesinde ise öğrenme süresi uzar ve ağıın yerel çözümlere takılmasına neden olur.

Yerel çözümü ise şu şekilde açıklayabiliriz. Bir problemin çözümü için en az hatayı veren ağırlık vektörünü pratikte her zaman belirlemek mümkün olmaz. Ağırlık vektörünü minimum kılan çözüm, ağıın sahip olabileceği en iyi çözümdür. Diğer yandan bu çözüme nasıl ulaşılacağı tam olarak bilinmemektedir. Ağ, eğitim sırasında söz konusu çözüme ulaşmaya çalışır. Ancak bazen ağ farklı bir çözümü izleyebilir ve performansını daha fazla iyileştirmek mümkün olmaz. İşte bu durumda çözüm vektörleri yerel çözümü verir. Kullanıcılar, ağların performansında  $\varepsilon$  (tolerans değeri) değerinde hatayı kabul ederler. Bu tolerans değerinin altındaki herhangi bir noktada, olay öğrenilmiş olduğu kabul edilir. Dolayısıyla bu nokta bir yerel çözüm sağlmasına rağmen kabul edilebilir bir çözümdür<sup>77</sup>.

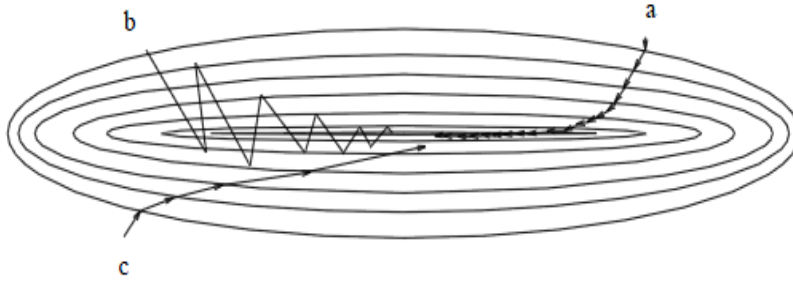
Momentum katsayısı, ağıın performansı üzerinde etkisi bulunan diğer bir parametredir. Bu katsayı da (0,1) gibi bir aralık içinde belirlenir. Öğrenme sürecini hızlandıran alternatif ağırlık ayarlama yöntemi olan momentum katsayısı, özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar bulması amacıyla kullanılır. Momentum katsayısı ile yapay sinir ağının öğrenme oranında belirli bir hızlanma elde

---

<sup>77</sup> Ercan Öztemel, a.g.e, s.83.

edilir. Bu deęerin çok küçük seilmesi yerel cözümlerden kurtulmayı güçleştirir. Deęerin çok büyük seilmesi de tek bir cözümüne ulaşılannda sorunlar yaratabilir.

Momentum katsayısının rolü şekil 2.19’da gösterilmiştir. Eęer momentum katsayısı kullanılmazsa aęın düşük öğrenme katsayısı ile yerel minimum cözümüne ulaşması uzun zaman alır. Öte yandan yüksek öğrenme katsayısı ile de salınımdan dolayı yerel minimuma hiçbir zaman ulaşamaz. Momentum katsayısı eklendiğinde ise yerel minimuma daha hızlı bir şekilde ulaşır<sup>78</sup>.



**Şekil 2.19.** Ağırlık Uzayındaki Düşme: a) düşük öğrenme katsayısı için, b) büyük öğrenme oranı için salınımlar, c) momentum terimi eklenmiş büyük öğrenme oranı

- **Eęitim ve Test Kümesinin Oluşturulması**

Yapay sinir aę modelini kurabilmek için verilmesi gereken kararlardan biri de hangi deęişkenlerin nasıl ve ne kadar veriyle kullanılacağıdır.

Eldeki tüm verilerin aęın eęitimi için kullanılması doğru bir yöntem değildir. Veri, eęitim, doğrulama ve test olmak üzere üç gruba ayrılır. Çoęu uygulamalarda, doğrulama eęitim kümesinin içinde yer alır. Oransal dağılım olarak da verilerin % 90’ı eęitim kümesi ve %10’u da test kümesi için ya da %80’i eęitim kümesi ve %20’si için test kümesi olarak bir dağılım kullanılabilir.

YSA’nın eęitimi ve testi için toplanan veri, sistemin düzgün çalışma uzayını kapsamalıdır. Örnek verilerin, çalışma uzayının sınırlarını belirlemesi ve YSA’nın yalnızca eęitildięi çalışma aralıęı için güvenilir sonuç verebildięi unutulmamalıdır. Genel özelliklerin net olarak belirlenmesi için örnek kümesinin geniş olması tercih edilir.

<sup>78</sup> Ben Kröse ve Patrick van der Smagt, a.g.e.,s.37.

Bir ağı doğru derecede eğitmek son derece önemlidir. Eğitim sürecinde çok fazla ya da çok az iterasyon kullanımından sakınılmalıdır. Gerekli iterasyon sayısı işlenecek kalıpların karmaşıklığına bağlıdır. Gerekenden az sayıda iterasyon, yetersiz bir eğitimle sonuçlanırken, gerekenden fazla iterasyon ise ağı eğitimi için gerekli süreyi uzatarak ağı verileri öğrenmek yerine ezberlemesine neden olur. Böylece ağı test kümesi üzerindeki performansı azalır<sup>79</sup>.

Test aşamasında, eğitim sürecinde ağı hiç görmediği ve ağı genelleştirebilme yeteneğini test edebilen bir örnek kümesi üzerinde, daha önce eğitilmiş olan ağı performansı değerlendirilir.

- **Normalizasyon**

Yapay sinir ağlarının en belirgin özelliklerinden olan doğrusal olmama özelliğini anlamlı kılan yaklaşım, verilerin normalizasyon işlemidir. Veriler ağa sunulmadan önce normalizasyon işlemine tabi tutulurlar. Bu işlem verideki aşırı salınımları engellemek ve sistem performansını arttırmak için kullanılır. Çünkü normalizasyon işlemi giriş verileri transfer edilirken fonksiyonun aktif olan bölgesinden aktarılmasını sağlar. Genellikle verilerin [0,1] veya [-1,1] aralıklarından birine ölçeklendirilmesi önerilmektedir.

Veri kümesinin [0,1] arasında bir ölçeklendirmeye tabi tutulabilmesi için o kümenin  $X_{\min}$   $X_{\max}$  aralığı bulunur ve aşağıdaki formüle göre ölçeklendirme yapılır.

$$X_{yeni} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

- **Performans Fonksiyonunun Seçilmesi**

Performans fonksiyonu YSA'nın öğrenme performansını ölçmede kullanılan bir ölçüttür. İleri beslemeli ağlarda genellikle hata kareleri ortalaması (Mean Square Error-MSE) performans fonksiyonu olarak kullanılır. MSE fonksiyonu istenen çıktı değerleri ile ağı ürettiği değerler arasındaki farkın kareleri toplamının ortalaması olup aşağıdaki formülle belirlenir:

---

<sup>79</sup> C. R. Krishnaswamy- Erika W. Gilbert- Mary M. Pashley, **Neural Network Applications in Finance: A Practical Introduction**, Financial Practice and Education, 2000, s.78.

$$MSE = \frac{1}{n} \sum_{i=1}^n \left( y_i - \hat{y}_i \right)^2$$

Burada;

$y_i$ : İstenen(gerçek) çıktı değerlerini

$\hat{y}_i$  : Ağın ürettiği çıktı değerlerini

$n$ : Veri sayısını gösterir.

Hatanın sıfıra yaklaşması, istenen çıktı değerine yakın çıktı değerinin elde edilmiş olduğunu gösterir.

Kullanılan diğer performans fonksiyonlarından bazıları hata kareleri ortalamasının karekökü (Root Mean Square Error-RMSE) ve ortalama mutlak hata (Mean Absolute Error-MAE) dır. Bunların denklem olarak gösterimi ise

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( y_i - \hat{y}_i \right)^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n \left| y_i - \hat{y}_i \right|$$

dır.

Çalışmamızın bu bölümünde yapay sinir ağının teorik yapısını ortaya koyduk. Uygulamasını yaptığımız işletme probleminin tahmini için kullandığımız ağ türü çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağıdır. Şimdi bu ağ modellerini açıklamaya çalışalım.

## ÜÇÜNCÜ BÖLÜM

### ÇOK KATMANLI ALGILAYICI MODEL VE RADYAL TABANLI FONKSİYON AĞ MODELİ

Bu bölümde uygulamada kullanacağımız çok katmanlı algılayıcı model ve radyal tabanlı fonksiyon ağ modeli ele alınacaktır.

#### 3.1. Çok Katmanlı Algılayıcı Model

Bu kısımda çok katmanlı algılayıcı modelin tanımı, katman yapısı, geriye yayılım algoritması ve XOR probleminin çözümü açıklanacaktır.

##### 3.1.1. Çok Katmanlı Algılayıcı Modelin Tanıtımı

Giriş ve çıkış katmanları arasında en az bir katmanın kullanıldığı yapay sinir ağını oluşturan çok katmanlı algılayıcı (MLP), tek katmanlı algılayıcıların aksine doğrusal olmayan problemlere çözüm üretmeleri nedeniyle günümüzde geniş kullanım alanları bulan en gözde yapay sinir ağ türüdür.

Basit algılayıcı modellerindeki sorun daha önce değindiğimiz gibi, yalnızca doğrusal problemlerinin çözümünü sağlamasıdır. Basit algılayıcı modeller doğrusal olmayan problemlerin (XOR) çözümünü gerçekleştiremediğinden durgunluğa giren yapay sinir ağ çalışmaları, 1969 yılında Minsky ve Papert'ın iki katmanlı ileri beslemeli ağ yapısı ile ilgili çalışmalarıyla basit algılayıcı modelin sınırlılığını gideren modeli ortaya koymuşlardır. Ancak geliştirdikleri çözüm hala giriş katmanından ara (gizli) katmana olan ağırlıkların nasıl belirleneceğine ilişkin bir çözüm olmamıştır. Burada eksik olan gizli katmanların ağırlıklarını belirlemek için kullanılabilecek öğrenme kuralının olmamasıdır. Rumelhart, Hinton ve Williams 1986 yılında yapay sinir ağlarına uyguladıkları geri yayılım algoritması ile XOR problemini çözüme ulaştırmışlardır. Böylece yapay sinir ağ çalışmaları tekrar hız kazanmıştır.

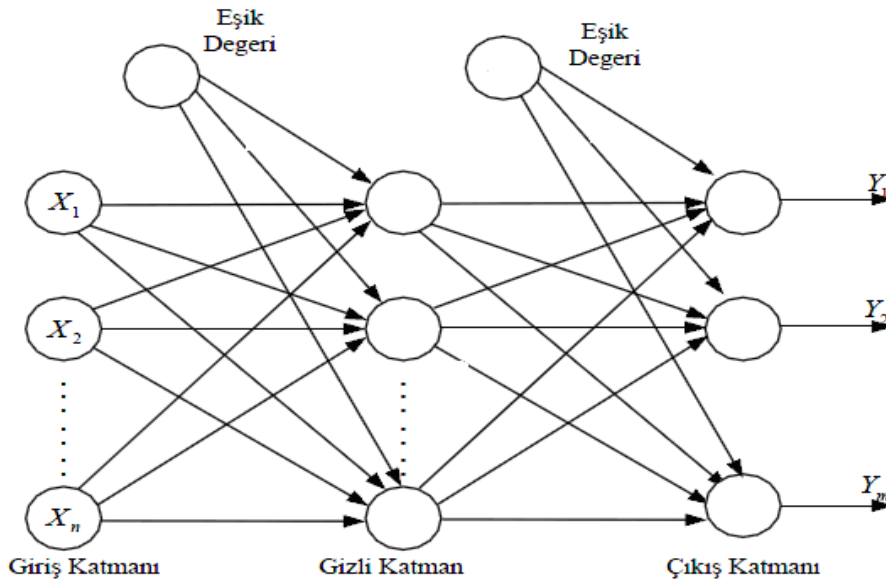
Bu çözümün altındaki ana düşünce; ara katmanlardaki birim hatalarının, çıktı katmanındaki birim hatalarının geriye yayılması ile belirlenmesidir. Bu nedenle yöntem *Geriye Yayılım Öğrenme Algoritması* (Back-Propagation Learning Algorithm) olarak da adlandırılmaktadır.

Çok katmanlı algılayıcı, *Genelleştirilmiş Delta Kuralı* adı verilen öğrenme kuralını kullanır ve bu öğrenme kuralına *Geriye Yayılım Algoritması* da denir. Genelleştirilmiş delta kuralı, en küçük kareler yöntemine dayalı bir öğrenme kuralıdır.

Çok katmanlı algılayıcı ağı danışmanlı öğrenme stratejisini kullanır. Yani, bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi beklenen çıktılar gösterilir.

### 3.1.2. Çok Katmanlı Algılayıcılarda Katman Yapısı

Çok katmanlı algılayıcılar giriş ve çıkış katmanları arasında en az bir ara katmanın kullanıldığı yapay sinir ağ sistemleridir<sup>80</sup>.



Şekil 3.1. Çok Katmanlı Algılayıcı(MLP) Yapay Sinir Ağının Yapısı

<sup>80</sup> Richard, P.,Lippmann, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 1987, s.15.

*Giriş katmanı*, dış ortamdan gelen bilgileri alarak ara katmana gönderir. Bu katmanda herhangi bir veri işleme olmaz. Çok katmanlı algılayıcılara birden fazla girdi gelebilir, fakat her işlemci elemanın bir tek girdisi ve bir tek çıktısı olur. Bu çıktı da bir sonraki katmandaki tüm işlemci elemanlara gönderilir. Bir anlamda bir katmandaki bir işlemci eleman, bir sonraki katmandaki tüm işlemci elemanlara bağlanır. Çok katmanlı algılayıcılarda bütün katmanlardaki işlemci elemanlar yalnızca bir sonraki işlemci elemanlarla bağlantılıdır ve dolayısıyla kendi içlerinde bağlantıları bulunmaz.

Çok katmanlı algılayıcı ağ yapısında veri akışı ileri doğru olur. Bu nedenle çok katmanlı algılayıcılar ileri beslemeli yapay sinir ağları sınıfına girer. Girdi katmanları, ağların katman sayısına dahil edilmez. Girdi katmanı, iki ara katman ve çıktı katmanından oluştuğunda ağ yapısına üç katmanlı algılayıcı denir.

*Ara katman*, girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Gizli katman olarak da adlandırılan ara katmanlar, çok katmanlı algılayıcıda birden fazla sayıda olabilir. Ayrıca her ara katmanda birden fazla işlemci eleman bulunabilir. Ara katman sayısı ile ara katmanlardaki işlemci eleman sayısı, deneme-yanılma yoluyla bulunur. Ara katmanda da her işlemci eleman bir sonraki katmandaki tüm işlemci elemanlarla bağlantılıdır. *Çıkış katmanı*, ara katmandan gelen bilgileri işleyerek dış ortama gönderir.

Geri yayılım ağlarında katman sayısı ve her katmandaki işlemci eleman sayısı, ağın performansı yönünden önemli bir karardır. Ancak bu sayıların neler olacağına ilişkin net seçim kriterleri bulunmamaktadır. Bunun yerine uygulamalar sonucunda ortaya çıkmış ve araştırmacılar tarafından kabul görmüş bazı kurallar bulunmaktadır. Bir ağı oluşturmak için aşağıda belirtilen kurallar uygulandıktan sonra, eğitime işlemine başlanır. Şimdi bu kuralları kısaca özetleyelim;<sup>81</sup>

**Kural 1:** Girdi ve çıktı verileri arasındaki ilişkinin karmaşıklık derecesi arttıkça, işlemci eleman sayısı da arttırılmalıdır.

---

<sup>81</sup> Dave Anderson ve George McNeill, a.g.e.,s.33.



**Kural 2:** Eđer modellenen problem birçok aşamaya ayrılabiliyorsa, ilave ara katmanlar gerekebilir. Eđer problem aşamalara ayrılamıyorsa, ilave katmanlar kolayca ezberlenebilir ve bu durum problemin gerçek genel çözümlü deęildir.

**Kural 3:** Yapay sinir aęının eęitilmesinde kullanılan örnek kümesinin genişlięi, ara katmanlardaki işlemci elemanların sayısı için bir üst limit kriteri oluşturur. Bu üst limiti belirlemek için önce eęitim kümesindeki girdi-çıkıtı çiftlerinin sayısı bulunmalı ve bulunan bu sayı aędaki toplam girdi ve çıkıtı işlemci elemanların sayısına bölünmelidir. Bölüm sonucunda elde edilen katsayı ölçeklendirme katsayısı olarak kullanılır. Bu katsayı beş ile on arasında bir deęerdir. Daha büyük ölçekleme katsayısı daha hatalı (gürültülü) veri için kullanılır. Aşırı gürültü içeren veri kümeleri için bu katsayı yirmi ile elli arasında deęerler alabilir. Hatasız (temiz) veri girişı olduğunda ise bu katsayı iki seviyesine kadar düşebilir. Ayrıca, genelleme yeteneęinin kaybolmasıyla aę kullanıřsız olacaęından dolayı bir ara katmandaki işlemci eleman sayısı çok fazla olamamalıdır.

### 3.1.3. Geri Yayılım Algoritması

Geri yayılım algoritması veya bir dięer adıyla Genelleştirilmiş Delta Algoritması uygulamada en çok kullanılan öğrenme algoritmasıdır. Geri yayılım algoritması verilen bir X giriş vektörü için beklenen çıkıřın bilindięi danışmanlı öğrenme yönteminin kullanıldıęı aęlar için uygundur. Bu öğrenme algoritması, sürekli, türevi alınabilir transfer fonksiyonlu nöronlardan meydana gelen ileri beslemeli ve çok katmanlı bir aę mimarisinde uygulanabilmektedir.

Geri yayılım algoritmasındaki hatalar, aędaki aęırlıkların bir fonksiyonu olarak görülür ve hataların kareleri ortalaması delta kuralında olduğü gibi, dereceli azaltma yöntemi kullanılarak minimize edilmeye çalışılır. Bu algoritma, hataları çıkıřtan girişe geriye doğru azaltmaya çalıştıęından geri yayılım adını almıřtır.

Geri yayılım algoritmasında azaltma işlemi nöronlar arasındaki baęlantı deęerlerinin bir başka deyişle aęırlıkların ayarlanmasıdır.

Bir yapay sinir ağının geri yayılım yoluyla eğitime işlemi, ileri doğru hesaplama ve geriye doğru hesaplama adı verilen iki aşamada gerçekleşir. Şimdi bunları kısaca açıklayalım.<sup>82</sup>

**İleri doğru hesaplama**, ağın çıktısını hesaplama aşamasıdır. Bilgi işleme, eğitim kümesindeki bir verinin girdi katmanından ağa gösterilmesi ile başlar. Daha önce de değinildiği gibi, girdi katmanında herhangi bir bilgi işleme gerçekleşmemektedir. Gelen girdiler hiç bir değişiklik yapılmadan ara katmana gönderilir. Bu durum  $y_k^i = x_k$  eşitliği ile gösterilir.

Ara katmandaki her işlemci eleman, girdi katmanındaki bütün işlemci elemanlardan gelen bilgileri bağlantı ağırlıklarıyla ağırlıklandırarak alır. Önce ara katmandaki işlemci elemanlara gelen net girdi;

$$S_j^a = \sum_{k=1}^n w_{kj} y_k^i \text{ ile hesaplanır.}$$

Burada;

$w_{kj}$  : k. girdi katmanı elemanını, j ara katman elemanına bağlayan bağlantının ağırlık değerini;

$y_k^i$  : girdi katmanındaki k. işlemci elemanın çıktısını göstermektedir.

j. ara katman elemanının çıktısı ise bu net girdinin transfer fonksiyonundan geçirilmesiyle hesaplanır.

Transfer fonksiyonu olarak genellikle sigmoid veya hiperbolik tanjant fonksiyon kullanılmakla birlikte, bu bir zorunluluk değildir. Önemli olan türevi alınabilir bir fonksiyon kullanmaktır. Geriye doğru hesaplamada fonksiyonun türevinin alınması gerektiğinden seçilen fonksiyonun, türevi alınabilir bir fonksiyon olmasına dikkat edilmelidir. Sigmoid fonksiyon kullanılması durumunda çıktı;

---

<sup>82</sup> Ercan Öztemel, a.g.e, s.78-79.

$$y_j^a = \frac{1}{1 + e^{-(s_j^a + \beta_j^a)}}$$

şeklindedir. Burada  $\beta_j$  ara katmanda bulunan j. elemana bağlanan eşik değer elemanın ağırlığını ve  $S_j$  de net girdiyi ifade eder. Bu eşğin ağırlık değeri sigmoid fonksiyonun yönünü belirler. Eğitim esnasında ağ bu değeri kendisi belirler.

Ara ve çıktı katmanların işlemci elemanlarının çıktıları, aynı şekilde kendilerine gelen net girdinin hesaplanması ve sigmoid fonksiyonundan geçirilmesi sonucu belirlenmektedirler. Çıktı katmanından çıkan değerler, yani çıktıları, bulununca ağın ileri hesaplama aşaması tamamlanmış olur.

**Geriye doğru hesaplama** aşamasında, ağa sunulan girdiler için üretilen çıktılar beklenen çıktı değerleriyle karşılaştırılır. Aradaki fark hata olarak kabul edilir. Amaç bu hatanın azaltılmasıdır. Geriye doğru hesaplamada söz konusu hata, ağın ağırlık değerlerine dağıtılır ve bir sonraki iterasyonda hatanın azaltılması sağlanır. Çıktı katmanındaki m. nöronda oluşan hata;

$$E_m = B_m - y_m$$

Burada;

$E_m$  : m. nöronda oluşan hata,

$B_m$  : m. nöronda sistemin ürettiği çıktı,

$y_m$  : m. nöronda beklenen çıktıyı gösterir.

$E_m$  değeri sadece bir tek nöronda oluşan hatadır. Çıktı katmanında oluşan toplam hata ise aşağıda verilen formül ile bulunur.

$$TH = \frac{1}{n} \sum_1^n (B_m - y_m)^2$$

Toplam hatayı minimize etmek için işlemci elemanlar hataları önceki katmanlara gönderir. Hata önceki katmanlara gönderilirken transfer fonksiyonun türevi ile bir transfer işlemi uygulanır. Hatanın iletilmesi, katman katman geriye doğru gidilerek ve süreçte genelleştirilmiş delta kuralı kullanılarak bağlantı ağırlıkları belirlenir. Ağırlıkların belirlenirken iki durum söz konusudur.

- Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi
- Ara katmanlar arası veya ara katman ile girdi katmanı arasındaki ağırlıkların değiştirilmesidir.

Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi şu şekilde açıklanabilir. Ara katmandaki  $j$ . işlemci elemanı çıktı katmanındaki  $m$ . işlemci elemana bağlayan bağlantının ağırlığındaki değişim miktarı  $\Delta w^a$  olmak üzere; herhangi bir  $t$  zamanında ( $t$ . iterasyonda) ağırlığın değişim miktarı,

$$\Delta w_{jm}^a(t) = \lambda \delta_m + \alpha \Delta w_{jm}^a(t-1)$$

formülü ile hesaplanır.

Burada;

$\lambda$  : öğrenme oranını,

$\alpha$  : momentum katsayısını,

$\delta_m$  :  $m$ . çıktı ünitesinin hatasını göstermektedir.

$\lambda$  öğrenme oranı, ağırlıkların bir sonraki düzeltmede hangi oranda değiştirileceğini belirlemektedir. Momentum katsayısı  $\alpha$ , MLP ağının öğrenmesi esnasında yerel bir optimum noktada kalmaması için ağırlık değişim değerinin, belirli bir oranda bir sonraki değişime eklenmesini sağlar. Eşitlikteki  $\delta_m$  ise  $m$ . çıktı ünitesinin hatasını göstermekte ve şu şekilde hesaplanmaktadır.

$$\delta_m = f'(s)E_m$$

Buradaki  $f'(s)$  transfer fonksiyonun türevidir. Sigmoid fonksiyonunun kullanılması durumunda

$$\delta_m = y_m(1 - y_m)E_m$$

olacaktır. Değişim miktarı hesaplandıktan sonra ağırlıkların  $t$ . iterasyondaki yeni değerleri ise

$$w_{jm}^a(t) = w_{jm}^a(t-1) + \Delta w_{jm}^a(t)$$

olur. Benzer şekilde eşik değeri elemanının da ağırlıklarını değiştirmek gerekir. Onun için önce değişim miktarı hesaplanmalıdır. Eğer çıktı katmanında bulunan işlemci elemanların eşik değeri ağırlıkları  $\beta^0$  ile gösterilirse, bu elemanın çıktısının sabit ve 1 olması nedeni ile değişim miktarı,

$$\Delta \beta_m^0(t) = \lambda \delta_m + \alpha \Delta \beta_m^0(t-1)$$

olacaktır. Eşik değeri  $t$ 'nci iterasyondaki ağırlık yeni değeri ise,

$$\beta_m^0(t) = \beta_m^0(t-1) + \Delta \beta_m^0(t)$$

olur.

Geriye doğru hesaplama aşamasındaki ikinci durum, ara katmanlar arası veya ara katman ile girdi katmanı arasındaki ağırlıkların değiştirilmesidir. Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesinde her ağırlık için sadece çıktı katmanındaki bir işlemci elemanın hatası dikkate alınırken, girdi katmanı ile ara katman arasındaki ağırlıkların değiştirilmesinde çıktı katmanındaki bütün işlemci elemanların hatasından payını alması gerekmektedir. Bu ağırlıklardaki değişim (mesela girdi katmanı ile ara katman arasındaki ağırlıkların değişimi)  $\Delta w^i$  ile gösterilirse değişim miktarı,

$$\Delta w_{kj}^i(t) = \lambda \delta_j^a y_j^i + \alpha \Delta w_{kj}^i(t-1)$$

olacaktır. Buradaki hata terimi  $\delta^a$  ise şu şekilde hesaplanır.

$$\delta_j^a = f'(s) \sum_m \delta_m w_{jm}^a$$

Transfer fonksiyonu olarak sigmoid fonksiyonu kullanılırsa hata değeri,

$$\delta_j^a = y_j^a(1 - y_j^a) \sum_m \delta_m w_{jm}^a$$

formülü ile hesaplanır. Değişim miktarı hesaplandıktan sonra ağırlıkların yeni değerleri ise,

$$w_{kj}^i(t) = w_{kj}^i(t-1) + \Delta w_{kj}^i(t)$$

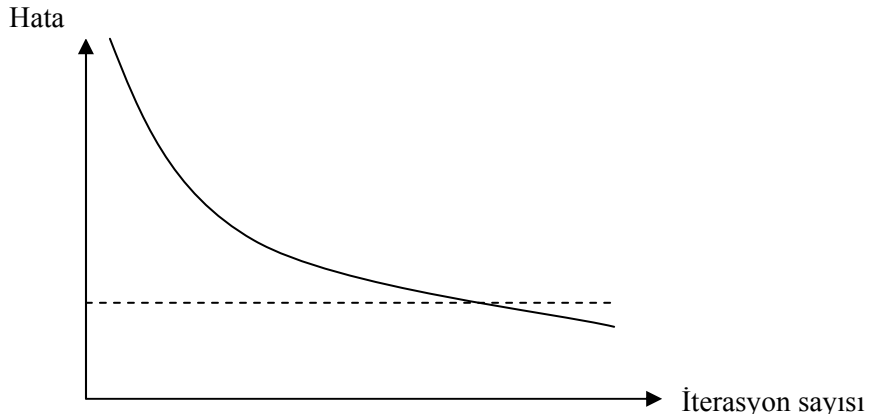
formülü ile belirlenir. Ara katman eşik değer ağırlıkları  $\beta^a$  ile gösterilirse değişim miktarı;

$$\Delta \beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1)$$

olacaktır. Ağırlıkların yeni değerleri ise t'nci iterasyonda aşağıdaki şekilde belirlenir.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta \beta_j^a(t)$$

Yapılan bu işlemler sonunda ağırlıklarının hepsi değiştirilmiş olacaktır. Bir iterasyon hem ileri hem de geriye hesaplamaları yapılarak tamamlanır. Daha sonra yeni bir örnek verilerek ikinci iterasyona başlanır. Burada her bir iterasyonda ağ hatasının azalması beklenir. Şekil 3.2'de bu durum gösterilmektedir. Bir anlamda iterasyon sayısı arttıkça ağ hatası azalır.



Şekil 3.2. İterasyon sayısı ile ağ hatasının ilişkisi

Belirli bir iterasyon sayısından sonra hata sayısı daha fazla azalmaz. Bu durum, ağırlıkların öğrenmesinin durduğunu ve daha iyi bir sonuca ulaşamayacağı anlamına gelir. Söz konusu işlemler gerçek çıktı ile beklenen çıktı arasındaki fark en aza indirilene kadar sürdürülür. Öğrenmenin sonlanması bir durdurma ölçütü ile gerçekleştirilir. Bu ölçüt genelde yönetimin kabul edebildiği bir hata düzeyine inmesiyle gerçekleşir.

### 3.1.4. XOR Probleminin Çözümü

XOR problemi, yapay sinir ağları için çok önemli bir yere sahiptir. Bunun nedeni ise basit algılayıcı modellerin doğrusal olmayan problemlerin (XOR) çözümünü gerçekleştirememesidir.

Aşağıda görüldüğü üzere söz konusu problemin çözümü beş adımda gerçekleşir. Şimdi XOR probleminin geriyayılım ağı ile nasıl çözüldüğünü bir örnek problem ile açıklamaya çalışalım.<sup>83</sup>

**Tablo 3.1 :** XOR Problemi için Girdi ve Çıktılar

	Girdi 1	Girdi 2	Çıktı
Örnek 1	0	0	0
Örnek 2	0	1	1
Örnek 3	1	0	1
Örnek 4	1	1	0

**1.Adım:** Örneklerin toplanması:

XOR problemi için 4 örnek ele alınmıştır ve her örnek için girdiler ve beklenen çıktı tablo 3.1’de görülmektedir.

**2.Adım:** Ağın topolojik yapısının belirlenmesi:

XOR probleminde iki girdi ve bir de çıktı olduğundan oluşturulacak olan MLP ağının girdi katmanında 2 adet işlemci eleman ve çıktı katmanında ise 1 adet işlemci eleman olacaktır. 1 ara katman ve 2 tane de ara katman işlemci elemanı ile problemin çözüleceği varsayılmaktadır. Ayrıca ara katman için 1 adet ve çıktı katmanı için de 1 adet eşik değeri elemanı vardır.

**3.Adım:** Öğrenme parametrelerinin belirlenmesi:

---

<sup>83</sup> Ercan Öztemel, a.g.e.,s.85-90.

Oluşturulan ağ için transfer fonksiyonu olarak sigmoid fonksiyon kullanıldığı, öğrenme oranı ( $\lambda$ ) ve momentum katsayısı ( $\alpha$ ) olarak aşağıda verilen değerler ile belirlendiği varsayalım,

$$\lambda = 0,5$$

$$\alpha = 0,8$$

**4.Adım:** Ağırlıkların başlangıç değerlerinin atanması:

Oluşturulan ağ için ağırlık vektörleri ve başlangıç değerleri de aşağıdaki şekilde belirlenmiş olsun. Girdi katmanı ile ara katman arasındaki ağırlıklar  $w^i$  matrisi ile gösterildiğine göre

$$W^i = \begin{bmatrix} 0,129952 & 0,570345 \\ -0,923123 & 0,328932 \end{bmatrix}$$

dir. Çıktı katmanı ile ara katman arasındaki ağırlıklar ise  $w^a$  ile gösterilsin

$$W^a = [0,164732 \quad 0,752621]$$

olarak verilsin. Eşik değer ağırlıkları da aşağıdaki şekilde verildiğini düşünelim.

$$\beta^a = [0,341332 \quad -0,115223]$$

$$\beta^0 = [-0,993423]$$

**5.Adım:** Örneklerin ağa verilmesi ve ileri doğru hesaplama işlemi:

Tablo 3.1 de görüldüğü üzere birinci örnek,  $x_1 = 0$ ,  $x_2 = 0$  ve  $d = 0$  olarak belirlenmiştir. Bu örneğin ağa verilmesi sonucunda, ara katman işlemci elemanlarının net girdileri (eşik değer işlemci elemanının ağırlık değerleri eklenmiş olarak) şu şekilde belirlenir.

$$s_1 = (0 * 0,129952) + (0 * -0,923123) + (1 * 0,341232) = 0,341232$$

$$s_2 = (0 * 0,570345) + (0 * -0,328932) + (1 * -0,115223) = -0,115223$$

Ara katman işlemci elemanlarının çıktı değerleri ise



$$y_1 = \frac{1}{1 + e^{-0,341232}} = 0,584490$$

$$y_2 = \frac{1}{1 + e^{0,115223}} = 0,471226$$

dır. Çıktı katmanındaki işlemci elemanın net girdisi de

$$s = (1 * -0,993423) + (0,584490 * 0,164732) + (0,471226 * -0,752621) = -0,542484$$

dır. Bu değer ile ağıın çıktısı da

$$y = \frac{1}{1 + e^{0,542484}} = 0,367610$$

olur. Beklenen çıktı 0 olduğuna göre ağıın hatası:  $E = 0 - 0,367610 = -0,367610$  olur. Bu hatanın geriye doğru yayılması sonucu ara katman ile çıktı katmanı arasındaki ağırlıkların değişim miktarları şu şekilde belirlenir.

$$\delta_1 = y_1(1 - y_1).E_1$$

$$\delta_1 = 0,367610 * (1 - 0,367610) * (-0,367610)$$

$$\delta_1 = -0,085459$$

$$\Delta w_{11}^a(t) = 0,5 * -0,085459 * 0,584490 + 0,8 * 0 = -0,024875$$

$$\Delta w_{21}^a(t) = -0,020135$$

$$\Delta \beta_1^0(t) = -0,042730$$

Ağırlıklardaki bu değişim miktarları ile ara katman ve çıktı katmanı arasındaki ağırlıklar yeniden belirlenebilir.

$$w_{11}^a(t) = A_{11}^a(t-1) + \Delta A_{11}^a(t)$$

$$w_{11}^a(t) = 0,16732 - 0,024975 = 0,139757$$

$$w_{21}^a(t) = 0,752621 - 0,020135 = 0,732486$$

$$\beta_1^0(t) = -0,993423 - 0,042730 = -1,036153$$

Benzer şekilde, girdi katmanı ile ara katman arasındaki ağırlıkların değişim miktarları ve yeni ağırlık değerleri de belirlenebilir. Ara katmandaki hata oranları ve değişim miktarları şu şekilde bulunur.

$$\delta_1^a(t) = y_1(1 - y_1)\delta_1 w_{11}^a(t-1)$$

$$\delta_1^a = 0,584490 * (1 - 0,584490) * (0,164732) * (-0,085459)$$

$$\delta_1^a = -0,034190$$

$$\delta_2^a = -0,160263$$

$$\Delta w_{11}^i(t) = 0,5 * -0,034190 * 0 + 0,8 * 0 = 0$$

$$\Delta w_{12}^i(t) = 0,5 * -0,034190 * 0 + 0,8 * 0 = 0$$

$$\Delta w_{21}^i(t) = 0$$

$$\Delta w_{22}^i(t) = 0$$

$$\beta_1^a(t) = 0,5 * 1 * -0,034190 = -0,017095$$

$$\beta_2^a(t) = 0,5 * 1 * -0,160263 = -0,080132$$

Bu değerler kullanılarak ağırlıklar değiştirilir. Ağırlıklardaki değişim miktarı 0 olduğunda ağırlık değerlerinde herhangi bir değişiklik olmayacak ancak eşik değeri ağırlıklarında değişiklik olacaktır.

$$\beta_1^a(t) = 0,341232 - 0,017095 = 0,3242038$$

$$\beta_2^a(t) = 0,115223 - 0,081325 = -0,0350905$$

Birinci iterasyon bittikten sonra ikinci iterasyon başlar. Bu kez söz konusu tabloda ikinci örnek ağı verilerek  $y_1 = 0$   $y_2 = 1$  ve  $d = 1$  olur. Yukarıdaki işlemler aynı şekilde tekrar

edilir. Bu iterasyonlar bütün çıktılar olumlu yanıt verinceye kadar devam eder. Ağ öğrendikten sonraki ağırlık değerleri ise aşağıdaki gibidir.

$$w^i = \begin{bmatrix} -6,072185 & -4,894898 \\ 2,454509 & 7,283063 \end{bmatrix}$$

$$w^a = [9,484580 \quad -4,473972]$$

$$\beta^a = [-6,062263 \quad -4,893081]$$

$$\beta^0 = [-9,792470]$$

Bu ağırlıklar ile girdiler ağa tekrar verildiğinde aşağıda verilen Tablo 3.2 deki sonuçlar elde edilir. Tablodaki sonuçları değerlendirdiğimizde, ağın problemi çok düşük hatalar ile çözebilecek şekilde öğrendiğini gösterir.

**Tablo 3.2 : XOR Problemini Öğrendikten Sonra ağın Ürettiği Çözümler ve Hata Oranları**

Girdi 1	Girdi 2	Beklenen Çıktı	Ağın Çıktısı	Hata
0	0	0	0,017622	-0,017
1	0	1	0,981504	0,018
0	1	1	0,981491	0,018
1	1	0	0,022782	-0,020

Çalışmamızın üçüncü bölümünün ilk dört kısmında çok katmanlı algılayıcı modelinin yapısı, nasıl çalıştığı ve buna ilişkin bir örnek problem ele alınarak açıklanmaya çalışılmıştır.

### 3.2. Radyal Tabanlı Fonksiyon Ağ Modeli

Bu kısımda radyal tabanlı fonksiyon ağ modelinin teorik yapısı ele alınacaktır.

### 3.2.1. Radyal Tabanlı Fonksiyon Ağları

Radyal tabanlı fonksiyon ağları (RBFN) ilk olarak çok değişkenli interpolasyon<sup>84</sup> problemlerinin çözümünde uygulanmıştır. Bu konu üzerindeki ilk çalışma Powell (1985) ve daha sonraki çalışma Light (1992) tarafından yürütülmüştür. Şimdi RBF nümerik analiz araştırmalarının en temel alanlarından biri durumundadır. Broomhead ve Lowe (1988) ise RBF'yi sinir ağları tasarımı konusunda kullanan ilk kişilerdir<sup>85</sup>.

RBF ağları MLP' ye oranla daha kısa zamanda eğitilmesi ve yerel minimumlara takılmaksızın en iyi çözüme yaklaşma özelliğini bulundurur. Bundan dolayı RBF ağları tahmin, eğri uydurma ve fonksiyon yaklaştırma gibi problemlerin çözümünü içeren uygulamalarda, MLP' ye alternatif bir sinir ağı olarak kullanılmaya başlanmıştır<sup>86</sup>.

RBF, öğrenme sürecinde danışmansız ve danışmanlı öğrenme olmak üzere karma bir öğrenme süreci kullanır. Danışmansız öğrenme ile gerçekleştirilen ilk aşama, girdi verilerine gizli tabakada radyal tabanlı fonksiyonu uygulama sürecidir. Danışmanlı öğrenme ile gerçekleştirilen ikinci aşama ise, gizli tabakadan elde edilen çıktılara doğrusal dönüşüm uygulanarak ağ çıktıları ile gözlenen çıktılar arasındaki hatayı en küçük yapan ağırlıkların belirlenmesi sürecidir.

### 3.2.2. Radyal Tabanlı Fonksiyon Ağlarında Katman Yapısı

RBF ağları bir giriş katmanı, transfer fonksiyonu olarak ağı ismini veren radyal fonksiyonların kullanıldığı tek bir gizli katman ve çıkış katmanı olmak üzere 3 katmanlı yapıdan oluşan ileri beslemeli bir ağıdır<sup>87</sup>. Bu ağın girişleri doğrusal değilken çıkışı ise doğrusaldır.

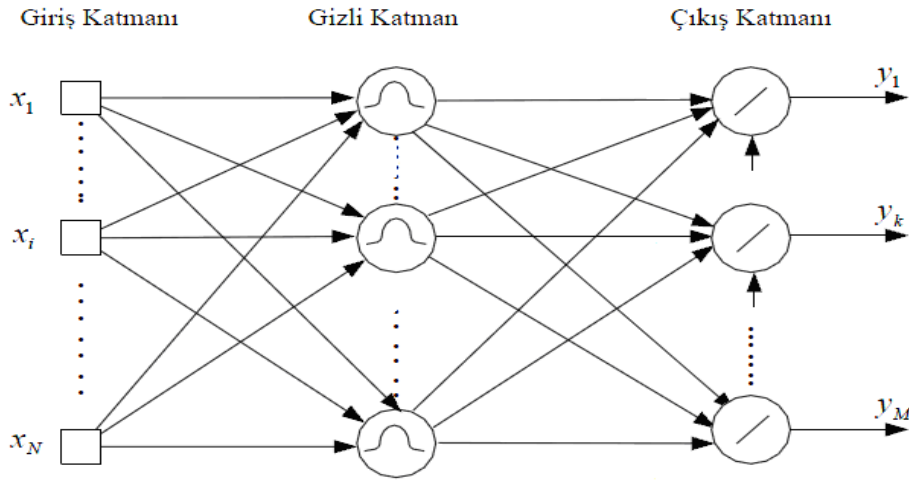
---

<sup>84</sup> İnterpolasyon: Varolan (bilinen) değer noktalarından yola çıkarak bu noktalar arasında, farklı bir yerde ve değeri bilinmeyen bir noktadaki olası değeri bulmaya/tahmin etmeye yarayan yöntemlerin tümüne verilen genel isimdir.

<sup>85</sup> Simon Haykin, a.g.e.,s.256.

<sup>86</sup> Oğuz Kaynar-Serkan Taştan-Ferhan Demirkoparan, "Ham Petrol Fiyatlarının Yapay Sinir Ağları ile Tahmini", Ege Akademik Bakış, 2010, s.565.

<sup>87</sup> Kwang Baek Kim-Cheol Ki Kim, "Performance Improvement of RBF Network Using ART2 Algorithm and Fuzzy Logic System", Australia Conference on Artificial Intelligence, 2004, s854.



**Şekil 3.3.** Radyal Tabanlı Fonksiyon (RBF) Yapay Sinir Ağının Yapısı

Girdi(giriş) katmanı, kaynak düğümlerden oluşur ve ağın çevre ile olan bağlantısını sağlar. Ağda bulunan tek gizli katman olan ikinci katman, girdi alanından gizli alana doğrusal olmayan dönüştürme yapar. Giriş katmanından gizli katmana dönüşüm, radyal tabanlı transfer fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür ve çoğu uygulamada gizli alan yüksek boyutludur. Çıktı katmanı ise doğrusal olup girdi katmanına uygulanan transfer sinyali olan ağa yanıt veren katmandır. Gizli katmandan çıkış katmanına uyarlamalı ve doğrusal bir dönüşüm gerçekleştirilir. Böylece ağın, girişe uygulanan transfer sinyaline yanıtını sağlayan çıkış katmanı ise doğrusallık özelliğini taşır<sup>88</sup>.

Doğrusal olmayan dönüşümün ardından doğrusal bir dönüşüm uygulanmasının mantığının matematiksel açıklaması ilk olarak Cover (1965) tarafından yayınlanan makalesinde görülmüştür. Bu makaleye göre, yüksek boyutlu bir uzayda yer alan bir örüntü- sınıflandırma probleminin doğrusal olarak ayrılabilirliği, düşük boyutlu uzaydakine göre daha olasıdır. Bu yüzden bir RBF ağındaki gizli uzayın boyutu sıklıkla yüksek yapılıdır. Diğer bir önemli nokta da gizli alan boyutunun ağın düzgün bir girdi-çıkış eşleştirmesine yaklaşması doğrudan ağ kapasitesiyle ilgili olma gerçeğidir. Gizli alanın boyutu ne kadar yüksek olursa yaklaşmanın doğruluğu da o kadar iyi olur<sup>89</sup>.

<sup>88</sup> Simon Haykın, a.g.e.,s.256

<sup>89</sup> Simon Haykın, a.g.e.,s.256.

RBF ağının çalışma ilkesi, giriş verisine bağlı olarak gizli katmanda, *uygun genişlik ve merkez değerli* RBF'leri belirlemek ve sonra da çıkış katmanında, bu fonksiyonların ürettiği çıktılarının uygun ağırlık değerleriyle doğrusal birleşimleri oluşturarak giriş-çıkış arasındaki ilişkiyi belirlemesidir<sup>90</sup>.

Giriş katmanındaki veriler, MLP' den farklı olarak ağırlık değerleri ile çarpılmadan doğrudan gizli katmana iletilirler. Gizli katmanda yer alan radyal fonksiyonların merkezlerini gösteren  $u_j$  referans vektörleri ile giriş vektörü arasındaki uzaklık değerine bağlı olarak gizli katman çıkışında aşağıdaki formülde gösterildiği şekliyle ( $\varphi_j$ ) bir değer üretilir. Birçok uzaklık ölçüsü tanımlanmasına rağmen uzaklık ölçüsü olarak genelde iki nokta arasındaki doğrusal uzaklığı ölçen *öklid uzaklığı* kullanılır.

RBFN modellerinde transfer fonksiyonu olarak birçok fonksiyon türü kullanılmaktadır. Bunlar; Doğrusal, Kübik, Gauss, Multi-Kuadratik, Ters Multi-Kuadratik fonksiyonlardır<sup>91</sup>. Gizli katmanda kullanılmak üzere çok sayıda radyal tabanlı fonksiyon önerilmesine karşın uygulamalarda en çok *Gauss fonksiyonu* tercih edilmektedir. Bu fonksiyonun gösterimi aşağıda verilmiştir.

$$\varphi_j = \exp\left(-\frac{\|x_i - u_j\|^2}{2\sigma_j^2}\right)$$

Burada;

$x_i$  : Ağa sunulan girdi vektörünü,

$u_j$  : Merkez(referans) vektörünü,

$\|\cdot\|$  uzaklık fonksiyonu,

$\sigma_j$  : Gauss fonksiyonunun genişlik değerini göstermektedir.

<sup>90</sup> Oğuz Kaynar-Serkan Taştan-Ferhan Demirkoparan,a.g.e.,s.565.

<sup>91</sup> Umut Okkan ve H. Yıldırım Dalkılıç, "Radyal Tabanlı Yapay Sinir Ağları ile Kemer Barajı Aylık Akımlarının Modellenmesi", İMO Teknik Dergi, 2012, s.5959.

Gizli katman çıkışında elde edilen  $\varphi_j$  değerleri daha sonra ağırlık katsayıları ile çarpılıp toplanarak aşağıdaki gibi ağırlık çıkışı belirlenir.

$$y_k = \sum_{j=1}^L w_{kj} \varphi_j + b_k$$

Burada;

L: Gizli katmandaki düğümlerin (fonksiyonların) sayısını,

$y_k$ : Çıkış katmanındaki j. düğümün k. giriş için çıkışı,

$w_{kj}$ : k. RBF birimi ile j. çıkış düğümü arasındaki ağırlığı,

$b_k$  : k. düğümdeki eşik (bias) terimini göstermektedir.

### 3.2.3. Öğrenme Stratejileri

Radyal tabanlı fonksiyon ağları tarafından gerçekleştirilen öğrenme sürecinde şu şekilde bir görselleştirme olabilir: Ağırlık çıkış birimleri ile ilişkili doğrusal ağırlıklar, gizli birimlerin doğrusal olmayan transfer fonksiyonları ile karşılaştırıldıklarında, başka bir zaman ölçeğinde gelişme eğilimini gösterir. Bu nedenle gizli katmanların transfer fonksiyonları, doğrusal olmayan optimizasyon stratejisine bağlı olarak daha yavaş ilerlerken, çıkış katmanların ağırlıkları, doğrusal optimizasyon stratejilerine bağlı olarak seri bir şekilde kendilerini ayarlarlar. Buradaki önemli nokta, radyal tabanlı fonksiyon ağlarının farklı katmanları farklı görevleri gerçekleştirir. Bu da bir ağırlık gizli katmanları çıkış katmanlarının optimizasyonunu farklı tekniklerle, hatta belki de farklı zaman ölçekleri kullanılarak gerçekleştirilmeyi mantıklı hale getirir<sup>92</sup>.

RBF ağının tasarımında radyal tabanlı fonksiyonların merkez vektörünü belirlemek için farklı öğrenme stratejileri izlenebilir. Bu yaklaşımlardan bazıları şu şekildedir<sup>93</sup>.

---

<sup>92</sup> Simon Haykin, a.g.e.,s.298.

<sup>93</sup> Simon Haykin, a.g.e.,s.299-302.

### 3.2.3.1. Rastgele Seçilmiş Sabit Merkezler

En basit yaklaşım, gizli birimlerin transfer fonksiyonlarının sabit radyal tabanlı fonksiyonlar tarafından tanımlandığını varsaymasıdır. Merkezlerin konumları eğitim kümesi içinde rastgele seçilebilir. Eğitim kümesindeki verilerin eldeki problemi temsil edecek bir şekilde dağılmış olması durumunda, bu yöntemi uygulamak mantıklı bir yaklaşım olarak kabul edilir. Radyal tabanlı fonksiyonlar için, standart sapma merkezlerin dağılımına göre sabit, *izotropik* bir Gauss fonksiyonu kullanılabilir. Bu durumda,  $t_i$  noktasında merkezlenmiş normalize bir RBF şu şekilde tanımlanabilir.

$$G(\|x - t_i\|^2) = \exp\left(-\frac{m_1}{d_{\max}^2} \|x - t_i\|^2\right), \quad i=1,2,\dots,m_1$$

Burada  $m_1$  merkezlerin sayısı ve  $d$  seçilen merkezler arasındaki maksimum uzaklıktır. Aslında, bütün Gauss RBF'lerinin standart sapması (veya genişliği) aşağıdaki gibi sabit alınabilir:

$$\sigma = \frac{d_{\max}}{\sqrt{2m_1}}$$

Standart sapmanın bu şekilde seçilmesi bize, Gauss fonksiyonunun ne aşırı geniş ne de aşırı dar olacağını garanti eder. Her iki uç nokta da istenmeyen durumlardır. Bu yaklaşımda öğrenilmesi gereken tek parametre, ağırlık çıkış katmanındaki doğrusal ağırlıklarıdır. Bunun basit bir yöntemi, *sanki tersi* (pseudoinverse) yöntemidir.

### 3.2.3.2. Merkezlerin Kendinden Uyarlamalı Seçimi

Rastgele seçilmiş sabit merkezler yönteminin temel problemi, tatmin edici düzeyde performans için büyük bir eğitim kümesi gerektirmesidir. Bunun üstesinden gelmenin bir yolu da iki aşamadan oluşan karma(hibrit) bir öğrenme sürecini kullanmaktır. Şimdi bu iki aşamalı süreci kısaca açıklayalım.

**a. Danışmansız öğrenme süreci aşaması.** Bu aşamada gizli katmandaki radyal tabanlı fonksiyonların merkezleri için uygun tahmin merkezlerinin belirlenmesi amaçlanır.



Bunun için veri kümesi mümkün olduğu kadar homojen alt gruplara ayıran k-ortalamalar ve Kohonen'in öz örgütleyici haritaları (Self Organizing Maps-SOM) gibi kümeleme algoritmaları kullanılır.

**b. Danışmanlı öğrenme süreci aşaması.** Bu aşamada ise çıkış katmanının doğrusal ağırlıklarını hesaplamak için kullanılan danışmanlı(denetimli) öğrenme işleminde, en küçük ortalama kare (Least Mean Square-LMS) yönteminin geliştirilmiş hali olan hata düzeltimi kullanılabilir.

### 3.2.3.3. Merkezlerin Denetimli Seçimi

Üçüncü yaklaşımda, radyal tabanlı fonksiyonların merkezleri ve ağı diğer bağımsız parametreleri danışmanlı öğrenme sürecine sokulur. Başka bir deyişle RBF ağı onun en genel biçimini ele alır. Böyle bir işlem için dereceli azaltma yöntemini uygulayan ve LMS algoritmasının geliştirilmiş hali olan hata düzeltimi (Error-correction) kullanılır.

### 3.3. RBF Ağları ile MLP Ağlarının Karşılaştırılması

Radyal tabanlı fonksiyon ağları ve çok katmanlı algılayıcılar; doğrusal olmayan katmanlı ileri beslemeli ağ örneklerindedir. Her ikisi de evrensel yaklaşıtııcılarıdır. Bu yüzden; belirlenmiş MLP'yi tam olarak taklit edebilecek kapasitede bir RBF ağının her zaman olduğunu, ya da tam tersini, bilmek şaşırtıcı değildir. Bununla birlikte, bu iki ağ birçok önemli açıdan birbirlerinden ayrılırlar. Bunları şu şekilde sıralayabiliriz<sup>94</sup>.

- Bir RBF ağı (en temel formunda) tek bir gizli katmanlı olabilirken, bir MLP ağı bir veya daha fazla gizli katmanlı olabilir.
- Tipik olarak bir MLP'nin gizli veya çıktı katmanında konumlanmış hesaplama düğümleri ortak bir sinirsel modeli paylaşır. Öte yandan, RBF ağının gizli

---

<sup>94</sup> Simon Haykin, a.g.e.,s.293.

katmanındaki hesaplama düğümleri biraz farklı olduğu gibi ağıın çıktı katmanındakilerden daha farklı bir yapıdadır.

- Bir RBF ağıının gizli katmanı doğrusal olmaz iken çıktı katmanı ise doğrusaldır. Öte yandan, bir MLP'nin birer örüntü sınıflandırıcısı olarak kullanılan gizli ve çıktı katmanlarının tümü genellikle doğrusal değildir.
- Bir RBF ağıındaki her gizli birimdeki transfer fonksiyonu, giriş vektörü ile o birimin merkezi arasındaki öklit uzaklığını belirler. Bu arada, bir MLP'deki her gizli birimin transfer fonksiyonu da giriş vektörünün iç çarpımını ve o birimin sinaptik ağırlık vektörünü hesaplar.
- MLP ağıları doğrusal olmayan giriş-çıkış tasarımlarına global yaklaşımlar oluşturur. Öte yandan üstel olarak azalan, yerel doğrusal dışılığı (Örn. Gauss fonksiyonları) kullanan RBF ağıları doğrusal olmayan giriş- çıkış tasarımına yerel yaklaşımlar oluşturur.

Bunların sonucunda doğrusal olmayan bir girdi-çıkış eşleme yaklaştırmasında aynı derecede doğruluk için MLP'nin RBF ağıından daha az sayıda parametreye sahip olması gerektiği anlamına gelir. RBF ağıı çıktı katmanının doğrusal özellikleri, çok katmanlı geriye yayılma algoritmasından daha çok Rosenblatt'ın geriye yayılma algoritmasıyla daha yakından ilişkili olduğunu gösterir. Bununla birlikte; RBF ağıı, girdi uzayının rastgele doğrusal olmayan dönüşümlerini uygulama kapasitesi, geriye yayılma algoritmasından farklıdır.

## DÖRDÜNCÜ BÖLÜM

### BİR TEKSTİL İŞLETMESİNDE ÇOKLU DOĞRUSAL REGRESYON VE YAPAY SİNİR AĞ MODELLERİNİN UYGULANMASI

#### 4.1. Firma ile İlgili Genel Bilgiler ve İş Akışı

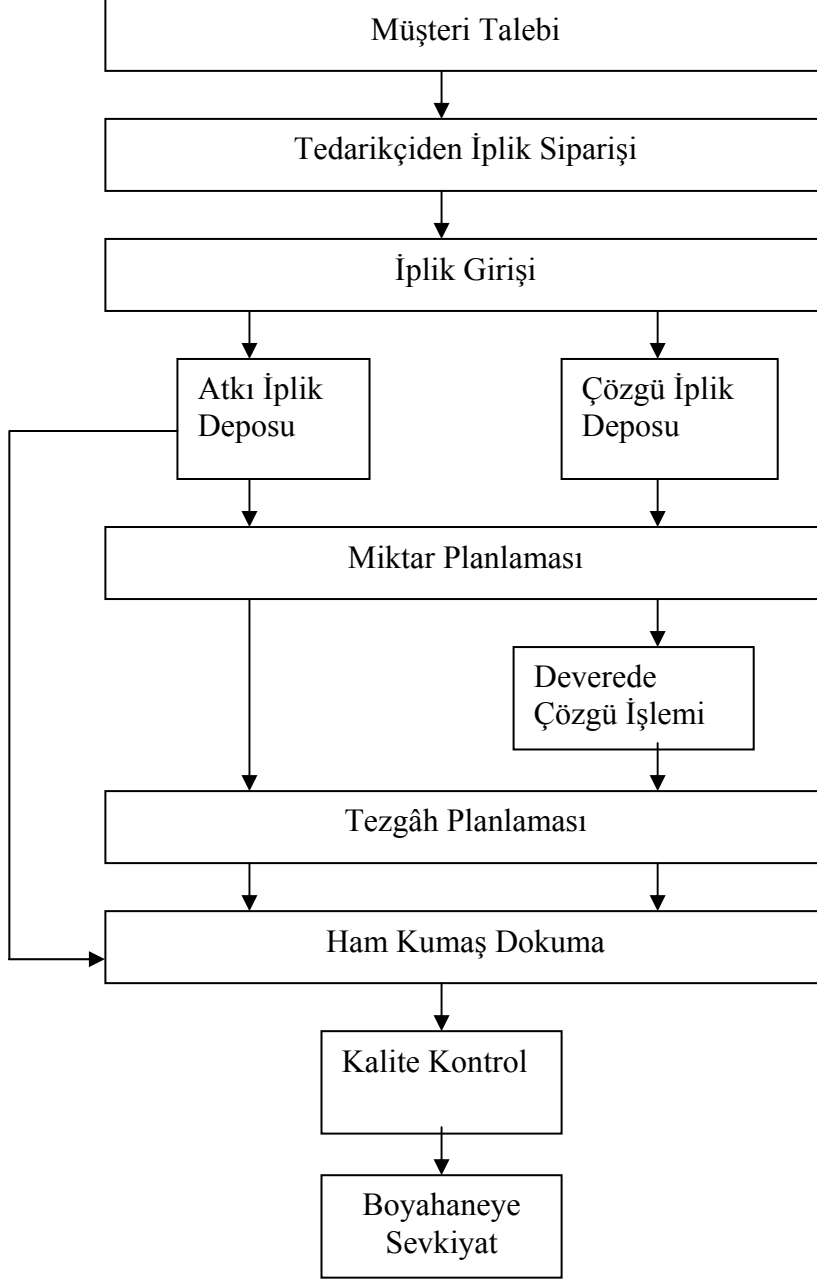
Uygulamasını yaptığımız tekstil firması 1987 yılında Bursa’da kurulmuştur. Tekstil firması ham kumaş dokuma ve dokunan kumaşların boyama işlemini yapan iki fabrikadan oluşmaktadır. Dokuma fabrikasında, müşteri siparişlerine göre pantolonların iç kısmında kullanılan iç ceplik ve iç astarı, kemer astarı gömlek ve pantolonluk ham kumaş üretilmektedir. Boyama fabrikasında üretilen ham kumaşlara boyama ve terbiye işlemleri yapılarak müşterilerin talepleri karşılanmaktadır. Bu firmanın günlük kumaş üretim kapasitesi 70.000 metre olup dünya standartlarında üretim yapmaktadır. Ürünlerin %50’si yurtiçi pazarlarında satılırken %50’si de yurtdışına ihraç edilmektedir.

Bu şirketin misyonu, hazır giyim ve konfeksiyon sektöründe müşteri taleplerine en iyi şekilde cevap vermek, memnuniyetlerini en üst düzeyde sağlamak, yurtiçi ve yurtdışındaki hedef pazarlarda en iyi tedarikçi olmaktır.

Şimdi firmanın üretim sürecini kısaca açıklayalım:

Müşterinin kumaş taleplerine göre tedarikçilere çözü ve atkı ipliği sipariş edilir. Sonra tedarikçiden gelen iplikler atkı ve çözü deposuna gelir. Depodaki çözü iplikleri müşteri talebine göre kaç metre çözüleceği ve kaç kilo atkı ipliği gideceği planlama kısmı bölümünce belirlenir. Depodaki çözü ipliklerine deverede çözü işlemi yapılır. Çözü işlemi bittikten sonra deverede çözülen iplikler ve depodaki atkı iplikleri birlikte ele alınarak hangi tezgâha bağlanacağına ilişkin planlama yapılır. Daha sonra planlanan bu iş emirlerine göre tezgâhlarda dokunur. Dokunan ham kumaşın kalite kontrolü yapılarak boyahane fabrikasına sevk edilir.

Yukarıda anlattığımız bu iş akışının şema olarak gösterimi aşağıdaki Şekil 4.1’de verilmiştir.



Şekil 4.1. Firmanın İş Akış Şeması

## 4.2. Verilerin Toplanması

Uygulama çalışması söz konusu işletmenin Temmuz 2010 ile Nisan 2012 yılları arasındaki dokuma kumaş toplarında meydana gelen atkı hata sayısı ve bu hataya neden olduğunu belirlediğimiz değişkenleri içermektedir.

Atkıda meydana gelen ve hataya neden olabilecek değişkenlerin neler olabileceğini belirlerken öncelikle firmanın kalite kontrol müdürü, planlama müdürü ve firma yetkilisi ile yaptığımız beyin fırtınası ile belirlenmiştir.

Atkıdaki hata sayısını ve hataya neden olan değişkenleri belirlemek için üretilen farklı parti ve top numaraları gereklidir. Hataya neden olan değişkenler(girdiler) kumaşın metresi, makine devir hızı, kumaş eni, çözgü sıklığı, salon randıman, iplik mukavemeti, atkı iplik no, çözgü iplik no, atkı sıklık, atkı iplik cinsi, çözgü iplik cinsi, tezgâh tipi ve örgü tipinden oluşmaktadır. Tutarlı bir tahmin için parti ve top numaralarına göre farklı tarihlerde üretilen kumaşlarla ilgili veriler toplanmıştır. Her parti ve top numarasına karşılık atkı hata sayısını gösteren çıktı verileri yine işletme kayıtlarından elde edilmiştir.

Çalışmamızda atkı iplik cinsi, çözgü iplik cinsi, örgü tipi ve tezgâh tipi kategorik veriler olup numaralandırılarak sınıflandırılmıştır. Dokuma üretiminde kullanılan atkı iplik cinsi; pamuk, polypamuk, pamuk polyester ve polyesterdir. Çözgü iplik cinsi polypamuk ve polyester olmak üzere iki çeşittir. Örgü tipinden dokuz çeşit bulunmaktadır ve bunlar; 1/1 bez, 2/1 dimi, 2/1 herringbone, 2/2 dimi, 2'li 1/1 bez, 2'li 2/2 dimi, 2'li 2/2 oxford, 3/1 twill ve armürdür. Dokuma kumaşı üreten tezgâh tipleri ise CTP, Somet Alpha, Somet Excel ve Sulzerdir. Söz konusu tezgâhlarda üretilen parti no ve top numarasına karşılık gelen 320 adet veri Excel tablosuna kaydedilerek Ek 1'de verilmiştir.

## 4.3. Çoklu Doğrusal Regresyon Modelin Firmaya Uygulanması

Teorik kısımda da belirttiğimiz gibi çoklu doğrusal regresyon(MLR) model yapısını gösteren denklem yapısı aşağıda verilmiştir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_n x_n + u$$

Formülde daha önce de belirttiğimiz gibi x'ler bağımsız değişkenleri, y ise bağımlı değişkeni gösterir. Buna göre modelimizde yer alan bağımsız değişkenler kumaşın metresi, makine devir hızı, kumaşın eni, çözgü sıklık, salon randımanı, iplik mukavemeti, atkı no, çözgü no, atkı sıklık, atkı cinsi, çözgü cinsi, tezgâh tipi ve örgü tipidir. Bağımlı değişken ise atkıdaki hata sayısıdır. Modelimizde bağımsız değişkenin her bir değeri için bağımlı değişken değerleri normal dağılıma sahiptir.

Modelimizde yer alan 13 bağımsız değişken arasında çoklu doğrusal bir ilişkinin var olup olmadığını öğrenmek için bu değişkenlerin tolerans ve VIF değerlerinin belirlenmesi gerekir. Kullandığımız SPSS 13 paket programı yardımı ile bu değişkenlerin tolerans ve VIF değerleri bulunarak Tablo 4.1'de verilmiştir. VIF değeri eşik değerden küçük ise, söz konusu bağımsız değişken ile diğer bağımsız değişkenler arasında çoklu doğrusal bağlantı probleminin olmadığı söylenir. Literatürde bu değer 10 olarak kabul edilir. Tablo 4.1' de görüldüğü üzere modelimizdeki VIF değerlerinin tümü 10'dan küçük olduğundan dolayı çoklu doğrusal bağlantı problemi yoktur. Bir başka deyişle bağımsız değişkenler arasında tam bir ilişki yoktur.

**Tablo 4.1.** Katsayılar Tablosu-1

Model		Tolerans	VIF
(Sabit )			
Kumaşın metresi	(X <sub>1</sub> )	0,717	1,394
Makina devir hızı	(X <sub>2</sub> )	0,595	1,671
Kumaşın eni	(X <sub>3</sub> )	0,555	1,801
Çözgü sıklık	(X <sub>4</sub> )	0,971	1,029
Salon Randımanı	(X <sub>5</sub> )	0,863	1,159
İplik Mukavemeti	(X <sub>6</sub> )	0,720	1,388
Atkı no	(X <sub>7</sub> )	0,969	1,032

Çözü no	(X <sub>8</sub> )	0,968	1,033
Atkı sıklık	(X <sub>9</sub> )	0,869	1,151
Atkı cinsi	(X <sub>10</sub> )	0,928	1,077
Çözü cinsi	(X <sub>11</sub> )	0,968	1,033
Tezgâh tipi	(X <sub>12</sub> )	0,940	1,064
Örgü tipi	(X <sub>13</sub> )	0,934	1,070

Ek 1'deki 320x14 boyutlu tabloda yer alan bağımsız değişkenlerin bağımlı değişkeni ne kadar açıkladığı veya aralarındaki ilişkiyi belirlemek için SPSS 13 paket program kullanılmış ve program çıktısı özet olarak aşağıda Tablo 4.2 de verilmiştir.

**Tablo 4.2.** Regresyon Modelin Özet Tablosu

Model	R	R <sup>2</sup>	Düzeltilmiş R <sup>2</sup>	Tahminin Standart Hatası
1	0,850	0,722	0,711	3,616

Modelin özeti tablosunda yer alan R<sup>2</sup> değeri bize bağımlı değişken olan atkı hatasının yüzde kaçlık kısmının bağımsız değişkenler tarafından açıklandığını göstermektedir. Determinasyon katsayısı R<sup>2</sup> = 0,722 ve düzeltilmiş determinasyon katsayısı R<sup>2</sup> = 0,711 gibi yüksek bir değer olarak karşımıza çıkar. Bu da atkı hata sayısı ile hata oluşumuna etki eden değişkenler arasındaki ilişkisinin güçlü olduğunu gösterir. Dolayısıyla düzeltilmiş R<sup>2</sup>, serbestlik derecesini dikkate aldığından dolayı modelin açıklama gücünün yorumlanmasında tercih edilir. Buna göre atkı hatasındaki değişimin %71,1'inin modele dahil ettiğimiz kumaşın metresi, makine devir hızı, kumaşın eni, çözgü sıklığı, salon randımanı, iplik mukavemeti, atkı no, çözgü no, atkı sıklığı, atkı cinsi, çözgü cinsi, tezgah tipi ve örgü tipi değişkenleri tarafından açıklanmaktadır.

Modelin anlamlılığını belirlemek için yine SPSS 13 paket program kullanılarak aşağıda varyans analizini gösteren Tablo verilmiştir.

**Tablo 4.3.** Varyans Analizi Tablosu

<b>Değişkenlik Kaynağı</b>	<b>Kareler Toplamı</b>	<b>Serbestlik Derecesi</b>	<b>Kareler Ortalaması</b>	<b>F</b>	<b>Sig.</b>
Regresyon	10417,559	13	801,351	61,278	0,000
Hata	4001,626	306	13,077		
Toplam	14419,186	319			

Varyans analizi tablosu, modelimizin bir bütün olarak anlamlı olup olmadığını test etmemizi sağlar. Tablodaki 61,278 F değeri ve buna karşı anlamlılık düzeyini gösteren sig. değeri 0,000 dır. Biliyoruz ki teoride anlamlılık düzeyini gösteren sig. değeri 0,05'ten küçük olduğunda anlamlıdır. Dolayısıyla 0,000 değeri modelimizin bir bütün olarak anlamlı olduğunu göstermektedir.

Modelimizde yer alan değişkenlerin katsayı değerleri ve bu katsayıların anlamlı olup olmadığı yine SPSS 13 paket program kullanılarak aşağıdaki Tablo 4.4 de verilmiştir.

**Tablo 4.4.** Katsayılar Tablosu-2

<b>Model</b>		<b>Standartlaştırılmamış Katsayılar</b>		<b>Standartlaştırılmış Katsayılar</b>	<b>t</b>	<b>Sig.</b>
		<b>B</b>	<b>Standart hata</b>	<b>Beta</b>		
(Sabit )		-71,526	8,064		-8,870	0,000
Kumaşın metresi	(X <sub>1</sub> )	-0,008	0,002	-0,134	-3,770	0,000
Makina devir hızı	(X <sub>2</sub> )	0,13	0,005	0,114	2,926	0,004
Kumaşın eni	(X <sub>3</sub> )	0,521	0,024	0,888	21,975	0,080



Çözümlü sıklık	(X <sub>4</sub> )	0,210	0,103	0,062	2,038	0,060
Salon Randımanı	(X <sub>5</sub> )	-24,007	8,281	-0,094	-2,899	0,004
İplik Mukavemeti	(X <sub>6</sub> )	-0,019	0,009	-0,071	-1,991	0,047
Atkı iplik no	(X <sub>7</sub> )	-0,189	0,056	-0,104	-3,387	0,001
Çözümlü iplik no	(X <sub>8</sub> )	-0,005	0,006	-0,027	-0,893	0,372
Atkı sıklık	(X <sub>9</sub> )	-0,095	0,037	-0,082	-2,545	0,311
Atkı iplik cinsi	(X <sub>10</sub> )	-0,732	0,210	-0,109	-3,482	0,001
Çözümlü iplik cinsi	(X <sub>11</sub> )	-0,237	0,450	-0,016	-0,528	0,598
Tezgâh tipi	(X <sub>12</sub> )	-0,438	0,208	-0,065	-2,105	0,036
Örgü tipi	(X <sub>13</sub> )	-0,375	0,123	-0,095	-3,046	0,003

Katsayılar tablosu ile her bir bağımsız değişkenin etkisi değerlendirilebilir. İlk sütundaki “B” değerleri önceki bölümde değinildiği gibi kısmi regresyon katsayılarını göstermektedir. Bir anlamda bu katsayılar çoklu doğrusal regresyon modeli oluşturulurken kullanılan bağımsız değişkenlerin katsayılarıdır.

Katsayıların kurduğumuz modelin anlamlı olup olmadığını belirlemek için t istatistik değerleri bulunmuş ve bu değerler katsayılar tablosunun t istatistiği sütununda verilmiştir. Sonra t istatistiği ile her değişkenin regresyon modeline katkı sağlayıp sağlamadığı yine tablodaki sig. değerlerine bakılarak karar verilir. Buna göre 0,05 anlamlılık düzeyi kullanıldığında, kumaşın eni, çözgü sıklık, çözgü iplik no, atkı sıklık ve çözgü iplik cinsi dışındaki bağımsız değişkenler 0,05 anlamlılık düzeyinden küçüktür. Buna göre analize giren 13 bağımsız değişkenden 8’i anlamlıdır. Dolayısıyla uygulama yaptığımız işletmenin çoklu doğrusal regresyon modeli aşağıdaki şekilde olmalıdır.

$$\hat{Y} = -71,526 - 0,008x_1 + 0,13x_2 - 24,366x_5 - 0,019x_6 - 0,189x_7 - 0,732x_{10} - 0,438x_{12} - 0,375x_{13}$$

Modelimizde on üç bağımsız değişkenden hangilerinin atkı sayısındaki hata oluşumu üzerinde etkili olduğu belirlenmeye çalışılmıştır. Regresyon analizi sonucunda sekiz değişken modele girmiştir. Söz konusu değişkenler; kumaşın metresi( $X_1$ ), makine devir hızı( $X_2$ ), salon randımanı( $X_5$ ), iplik mukavemeti( $X_6$ ), atkı iplik no( $X_7$ ), atkı iplik cinsi( $X_{10}$ ), tezgâh tipi( $X_{12}$ ), örgü tipi( $X_{13}$ ) değişkenleridir. Dolayısıyla bu 8 değişken atkı hata sayısının oluşmasında etkilidir. Diğer beş değişken yani kumaşın eni, çözgü sıklık, çözgü iplik no, atkı sıklık ve çözgü iplik cinsi hata sayısının belirlenmesi üzerinde etkisi yoktur.

Şimdi bu 8 değişkenin atkı hata sayısı üzerindeki etkisini açıklayalım.

Üretilen kumaşın metresindeki 1 santimetrelik artış ortalama atkı sayısındaki hatanın 0,008 azalmasına neden olur. Makine devir sayısındaki 1 dakikalık artış (normal hızından fazla) ortalama atkı sayısındaki hatanın 0,13 artmasına neden olur. Salon randımanındaki yüzde 1'lik artış ortalama atkı sayısındaki hatanın 24,007 azalmasına neden olur. İplik mukavemetindeki 1 rkm'lik artış ortalama atkı sayısındaki hatanın 0,019 azaltır. Atkı iplik no'sundaki 1 denyelik artış ortalama atkı sayısındaki hatanın 0,189 azalmasına neden olacaktır.

Modelimizde atkı iplik cinsi, tezgâh tipi ve örgü tipi kategorik verilerdir. Şimdi bu verileri yorumlayalım.

Poly pamuk atkı ipliği pamuk atkı ipliğine göre 0,732 adet daha az atkı sayısında hataya neden olurken, pamuk polyester atkı ipliği pamuk atkı ipliğine göre 1,464 adet daha az atkı sayısında hataya neden olur. Polyester atkı ipliği pamuk atkı ipliğine göre 2,196 adet daha az atkı sayısında hataya neden olur.

Tezgâh türlerinin atkı hata sayısına etkisi için şunları söyleyebiliriz.

Tezgâh tipi 2 (Sulzer) tezgâh tipi 1'e (CTP) göre 0,438 adet daha az atkı sayısında hataya neden olurken, tezgâh tipi 3 (Somet Excell) tezgâh tipi 1'e (CTP) göre 0,876 adet daha az atkı sayısında hataya neden olur ve tezgâh tipi 4 (Somet Alpha) tezgâh tipi 1'e (CTP) göre 1,314 adet daha az atkı sayısında hataya neden olur.

Üretimde örgü tipi 1 (1/1 bez) yerine örgü tipi 2 (2/1 dimi) iplik kullanıldığında kumaşlardaki atkı hata sayısında 0,375 adet bir azalma olur. Örgü tipi 3 (2/1 herringbone) ise örgü tipi 1'e (1/1 bez) göre 0,750 adet daha az atkı sayısında hataya neden olmaktadır. Örgü tipi 4 (2/2 dimi) örgü tipi 1'e (1/1 bez) göre 1,125 adet daha az atkı sayısında hataya neden olur. Örgü tipi 5(2'li 1/1 bez) örgü tipi 1'e (1/1 bez) göre 1,5 adet daha az atkı sayısında hataya neden olur. Örgü tipi 6 (2'li 2/2 dimi) örgü tipi 1'e (1/1 bez) göre 1,875 adet daha az atkı sayısında hataya neden olur. Örgü tipi 7 (2'li 2/2 oxford) örgü tipi 1'e(1/1 bez) göre 2,25 adet daha az atkı sayısında hataya neden olmaktadır. Örgü tipi 8(3/1 twill) örgü tipi 1'e (1/1 bez) göre 2,625 adet daha az atkı sayısında hataya neden olur. Örgü tipi 9 (armür) örgü tipi 1'e (1/1 bez) göre 3 adet daha az atkı sayısında hataya neden olduğu görülmektedir.

Regresyon modelimizin tüm verileri işleme alınarak hata değerlerini hesapladık. Hesaplama sonucunda elde ettiğimiz MSE (hata kareler ortalaması), RMSE (hata kareler ortalamasının karekökü) ve MAE(ortalama mutlak hata) hata değer sonuçları; MSE değeri için 13.52, RMSE değeri için 3.68, ve MAE değeri için ise 2.72 dir.

Çoklu doğrusal regresyon modeli ile yapmış olduğumuz tahmin işleminden sonra şimdi de yapay sinir ağ modellerinin atkı hata sayısı tahmininde nasıl kullanılabileceğini açıklamaya çalışalım.

#### **4.4. Yapay Sinir Ağ Modellerinin Firmaya Uygulanması**

Bu kısımda atkı hata sayıları tahmini için yapay sinir ağ modellerinden çok katmanlı algılayıcı (MLP) model ve radyal tabanlı fonksiyon ağ (RBFN) modellerine ilişkin yaptığımız çalışmalar ve model sonuçlarıyla ilgili bilgiler verilecektir.

Veri toplama kısmında açıkladığımız gibi 320 tane veriden oluşan modelimizin girdi verileri; kumaşın metresi, makine devir hızı, kumaşın eni, çözgü sıklık, salon randımanı, iplik mukavemeti, atkı no, çözgü no, atkı sıklık, atkı cinsi, çözgü cinsi, tezgâh tipi ve örgü tipidir. Bu girdi verilerine karşılık çıktı da atkıdaki hata sayısıdır. Bu değişkenler arasındaki ilişkiyi tahmin etmek amacıyla öncelikle yapay sinir ağ

modellerinden çok katmanlı algılayıcı model ve daha sonra da radyal tabanlı fonksiyon ağ model uygulaması yapılmıştır. Çok katmanlı algılayıcı model ve radyal tabanlı ağ modelin çözümünde Matlab R2010b programından faydalanılmıştır.

#### 4.4.1. Çok Katmanlı Algılayıcı Model

İlk olarak Excel sayfasında yer alan ve 320 adet veriden oluşan veriler, Excel programından Matlab programına aktarılmıştır. Daha sonra da verilerin %80 'i eğitim verisi ve %20'si de test verisi olarak bölünmüştür. Eğitim verilerinin de %25'i doğrulama verisi olarak bölünmüş ve böylece verilerin %60'ı eğitim(192 adet), %20'si doğrulama(64 adet) ve %20'si de test(64 adet) verisi olarak üç parçaya bölünmüştür. Bu bölme işlemi rasgele yapılmıştır.

Veriler eğitim, doğrulama ve test verileri olarak belirlenmiştir. Sonra verilerin eğitimi için çok katmanlı algılayıcı model kurulmuştur. Bu modelin girdi katmanındaki 13 nöron yukarıda söz ettiğimiz değişkenleri içerirken çıkış katmanındaki 1 nöron da atkıdaki hata sayısını göstermektedir. Ancak ağın gizli katmanı ve bu katmandaki nöron sayısı belli değildir. Bunlar *deneme yanılma yoluyla* belirlenmiştir.

Modelimizde gizli katman sayısı ise bir tane alınmıştır. Çünkü birden fazla gizli katmanın yer almasının pratikte bir yarar sağlamadığı gibi öğrenmeyi de yavaşlattığı ortaya konulmuştur<sup>95</sup>. Gizli katmanda kaç tane gizli nöron olacağını belirleyebilmek için bu katmana 1'den 50'ye kadar nöron sayısı verilmiş ve her bir model 10'ar kez denenerek bir anlamda programı 500 kere çalıştırarak en iyi model belirlenmeye çalışılmıştır.

Modelde giriş ile gizli katman arasında kullanılan transfer fonksiyonu *hiperbolik tanjant sigmoid* (tansig) dir. Gizli katman ile çıkış katmanı arasında kullanılan transfer fonksiyonu ise *doğrusal* (pureline) fonksiyondur. Modelde kullanılan transfer fonksiyonu, sigmoid tanjant türünde olduğundan veri kümeleri ağa verilmeden önce her bir giriş ve

---

<sup>95</sup> Timothy, Masters, Practical Neural Network Recipes in C++, Academic Press, Toronto, 1993, s.174.

çıkış değişkenlerinin maksimum ve minimum değerleri bulunarak -1 ile +1 arasında normalize edilmiştir.

Çalışmamızda MLP modeli, yapay sinir ağ mimarisi olarak ileri beslemeli sinir ağı modelidir. Öğrenme tipi olarak ise denetimli öğrenmedir. En uygun modelin bulunması aşamasında, verilerin eğitimi için hatayı geriye yayan algoritmalarından *Levenberg-Marquardt* (LM) algoritması kullanılmıştır. Programda eğitim yapılırken maksimum *iterasyon*(epoch) sayısı 1000 dir ve zaman sınırlaması yoktur. Performans kriteri *hata kareleri ortalaması*(MSE) dir ve amaç bu değerini sıfır olmasıdır. *Maksimum doğrulama kontrolü* 6'dır. Matlab kendisi bu değeri belirler. Bunun anlamı doğrulama kümesi üst üste altı kez kötüye giderse eğitimi durdur anlamına gelir. Çünkü sistemin ezberlemeye başladığını gösterir. *Öğrenme katsayısını* ise program başlangıçta 0,001 olarak alır. Bu katsayı değeri eğitim sürecinde kötüleşme olursa kendisi müdahale ederek artırıp veya azaltarak bu oranı değiştirir. Başka bir deyişle kullanılan eğitim algoritması (LM) eğitim aşamasındaki öğrenme katsayısını kendisi ayarlar.

Parametrelere uygun model kurulduktan ve eğittikten sonra modelin sonuçları test edilerek en uygun model belirlenir. En iyi modeli belirlemede hata performansı olarak MSE, RMSE ve MAE ölçütleri kullanılmıştır. Bu değerlerden hata kareleri ortalamasının(MSE) en küçük olduğu değer ile determinasyon katsayısının ( $R^2$ ) en büyük olduğu satırı veren model en iyi çözümü veren ağ modelidir.

Modeli oluşturmak için yapılan denemeler aşağıda Tablo 4.5'de verilmiştir.

**Tablo 4.5.** En uygun MLP modelini belirlemek için yapılan denemeler

Gizli Katman Nöron Sayısı	R	$R^2$	MSE	RMSE	MAE
1	0,89	0,79	10,84	3,29	2,50
2	0,91	0,82	10,61	3,26	2,47
3	0,91	0,83	8,98	3,00	2,25
4	0,90	0,81	9,45	3,07	2,32
5	0,95	0,90	5,49	2,34	1,89
6	0,96	0,93	4,21	2,05	1,63
7	0,95	0,90	5,69	2,39	1,90

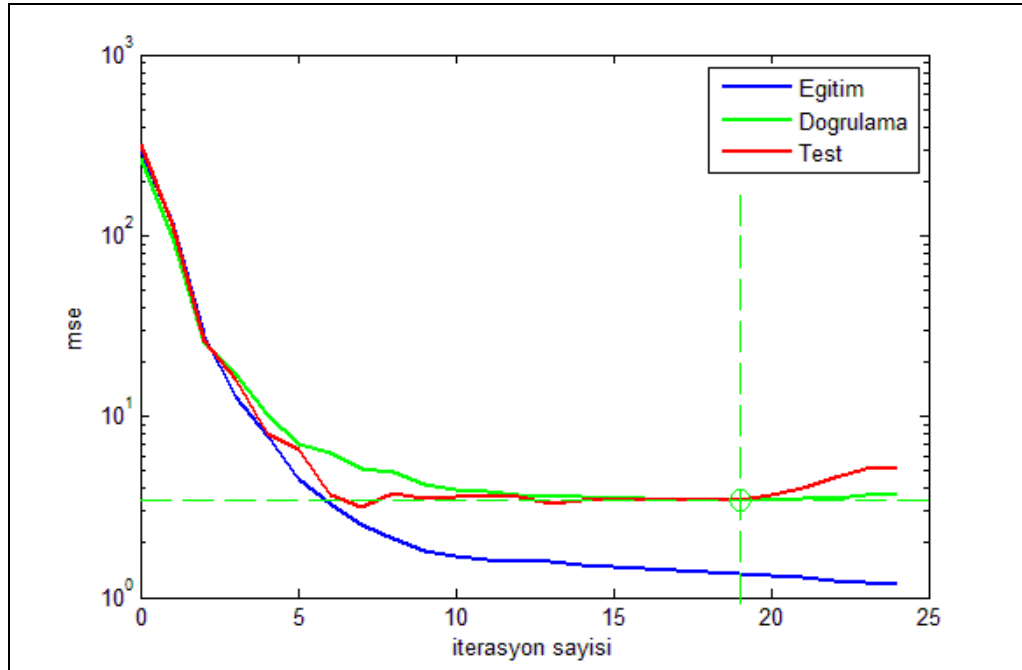
8	0,96	0,92	5,91	2,43	1,98
9	0,95	0,90	5,76	2,40	1,96
10	0,93	0,87	7,02	2,65	2,09
11	0,91	0,84	8,34	2,89	2,46
12	0,94	0,88	6,53	2,56	2,04
13	0,95	0,90	6,21	2,49	1,85
14	0,95	0,90	5,86	2,42	1,85
15	0,90	0,81	10,90	3,30	2,71
16	0,92	0,85	8,00	2,83	2,40
17	0,94	0,88	7,05	2,66	2,21
18	0,93	0,87	7,17	2,68	2,27
19	0,91	0,83	8,72	2,95	2,50
20	0,94	0,88	6,53	2,56	2,07
21	0,95	0,90	6,15	2,48	2,12
22	0,94	0,89	6,52	2,55	2,08
23	0,94	0,88	7,37	2,72	2,03
24	0,91	0,83	11,53	3,40	2,56
25	0,91	0,83	8,69	2,95	2,42
26	0,93	0,87	7,31	2,70	2,08
27	0,91	0,83	9,31	3,05	2,55
28	0,88	0,78	11,64	3,41	2,97
29	0,91	0,83	11,84	3,44	2,93
30	0,87	0,75	14,79	3,85	3,17
31	0,86	0,73	15,34	3,92	3,33
32	0,90	0,81	12,26	3,50	2,84
33	0,89	0,79	12,12	3,48	3,05
34	0,82	0,67	18,70	4,32	3,70
35	0,87	0,76	15,57	3,95	3,17
36	0,86	0,74	14,93	3,86	3,15
37	0,86	0,74	15,48	3,93	3,27
38	0,84	0,71	19,88	4,46	3,75
39	0,84	0,70	17,96	4,24	3,25
40	0,81	0,66	20,28	4,50	3,71
41	0,82	0,67	19,19	4,38	3,33
42	0,77	0,60	21,88	4,68	3,73
43	0,85	0,73	16,99	4,12	3,38
44	0,88	0,78	15,65	3,96	3,28
45	0,88	0,77	16,36	4,04	3,17
46	0,82	0,67	21,03	4,59	3,60
47	0,84	0,71	22,76	4,77	4,02
48	0,85	0,72	20,64	4,54	4,07
49	0,84	0,70	21,96	4,69	3,74
50	0,81	0,65	23,06	4,80	4,00

Yukarıdaki Tablo 4.5 de görüldüğü üzere gizli katmandaki nöron sayısını belirlemek için yaptığımız denemeler sonucunda en iyi modeli veren ağ yapısı için gizli katmandaki nöron sayısının 6 olması gerektiği sonucuna varılmıştır. Çünkü Tablodan da

görüldüğü üzere en küçük ortalama hata karesinin (MSE) değerini ve en büyük  $R^2$  değerini 6 adet gizli nöron içeren bir yapıda sağlandığı gözlenmektedir. Bu nedenle geliştirilecek yapay sinir ağının gizli katmanında bulunacak nöron sayısı 6 olarak seçilmiştir. Bu ağ yapısında MSE değeri 4,21 iken  $R^2$  değeri ise 0,93'tür. Dolayısıyla en uygun model 13-6-1 ağ yapılı modeldir.

Eğitim sonucunda her iterasyondaki eğitim, doğrulama ve test kümelerine ilişkin hata değerlerinin ne şekilde değiştiğini aşağıdaki Grafik 4.1 ile gözlemleyebiliriz. Grafikte görüldüğü üzere ağın eğitimi 19. iterasyonda(epoch) optimum sonuca ulaşmıştır.

**Grafik 4.1.** Eğitim, doğrulama ve test kümelerine ilişkin hata performansları



Yapay sinir ağı eğitimini tamamlanmasıyla ağ öğrenme işlemini tamamlar. Ağın öğrenmesindeki amaç, ağıdaki giriş nöronları ile ara katmandaki nöronların ve ara katmandaki nöronlar ile çıkış nöronunun ağırlık değerlerinin belirlenmesidir. Matlab programı ile ara katmandaki altı tane nöron ve çıkış katmanında bulunan bir nöronun eşik değeri de belirlenmiştir. Şimdi ağın öğrenme işleminin tamamlanmasıyla söz ettiğimiz ağırlık değerleri Tablo 4.6 ve 4.8 de, eşik değerleri ise Tablo 4.7 ve 4.9 da verilmiştir.

**Tablo 4.6.** Giriş nöronları ile gizli katman nöronları arasındaki ağırlık değerleri

	1	2	3	4	5	6
1	-1,05188	-0,03315	-0,72641	-0,25652	0,000875	0,455241
2	-0,30537	-1,55634	1,338325	0,892647	-3,28382	3,468618
3	0,743233	1,388676	-0,15201	-0,0583	-2,94845	-1,18493
4	-1,17038	-0,17237	-0,10022	0,055761	-0,01139	-0,50551
5	1,122748	0,093075	0,141432	0,414692	-0,01824	-0,31523
6	-0,53579	-0,15042	0,026291	-0,49916	-0,01657	0,052102
7	-0,31939	-0,90722	0,573461	-1,14893	0,090786	0,384887
8	-0,37582	-0,13303	-1,91202	0,508712	0,018881	0,281167
9	-0,64818	-0,38325	0,607804	0,087299	-0,00616	-0,54368
10	-1,11506	-0,01908	-0,24243	0,650451	-0,00563	-0,38202
11	-0,88564	-0,78805	-1,03688	-0,19648	0,006525	-0,39699
12	-0,63438	0,077311	0,582185	0,318522	-0,02899	0,060333
13	1,292661	-0,09734	0,939653	1,331989	-0,10019	-0,02967

**Tablo 4.7.** Gizli katmandaki nöronların eşik değerleri

Gizli katmandaki nöronlar	Eşik değerleri
1	2,191710538
2	-1,286336288
3	0,673465065
4	1,120004761
5	0,406787713
6	1,480036409

**Tablo 4.8.** Gizli katmandaki nöronlar ile çıkış katmanındaki nöron arasındaki ağırlık değerleri

	1	2	3	4	5	6
1	0,122988091	0,274687	-0,06853	-0,11371	-0,58052	-0,35876



**Tablo 4.9.** Çıktı katmandaki nöronun eşik değeri

Çıkış katmandaki nöron	Eşik değeri
1	0,519213

Ağ eğitiminin tamamlanmasıyla ağın öğrenip öğrenmediğini bir başka ifadeyle performansını ölçmek için denemeler yapılarak ağın testi yapılmıştır. Ağı test etmek için ağın eğitimi sırasında görmediği, yani test etmek amacıyla ayırmış olduğumuz test kümesindeki veriler kullanılmıştır. Bu veriler ağa gösterilir ve ağ eğitimi sırasında, ağın belirlemiş olduğu ağırlık ve eşik değerlerini kullanarak daha önce görmediği bu veriler için çıktılar üretir.

Şimdi test kümesinde yer alan gerçek veriler ile MLP ağının ürettiği tahmini çıktılara ilişkin değerler aşağıdaki Tablo 4.10'da verilmiştir. Bu tahmin değerleri modelimizi %93 ile başarılı tahmin ettiği Tablo 4.5'de göstermiştik. İşte bu %93 tahmin ile işletmenin gerçek verileri ile modelimizin çıktı tahmin değerlerin kıyaslanması için Tablo 4.10'da verilmiştir.

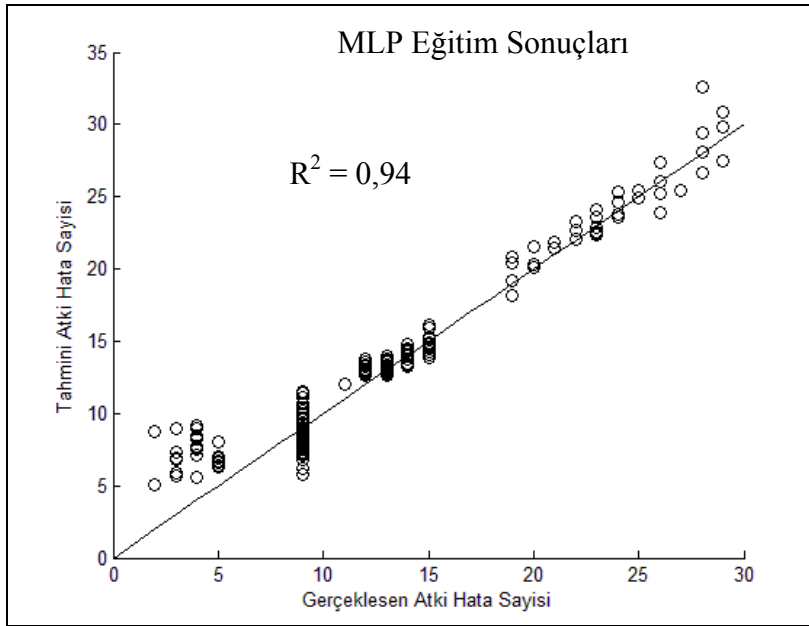
**Tablo 4.10.** MLP ağının çıktıları ile gerçek veriler

Test Kümesi	
MLP ağının çıktısı	Gerçek veri( Atkıdaki hata sayısı)
7,11	9
14,37	14
21,13	20
8,93	9
15,23	15
22,14	23
13,64	12
8,45	9
14,32	15
25,01	26
14,25	14
23,41	21
29,42	28
24,72	23
8,03	9
9,20	4
13,47	13
9,85	13
7,81	9
13,11	13

16,23	15
13,72	12
14,32	15
23,79	21
13,22	12
8,38	5
9,04	9
9,57	9
7,13	9
7,24	11
8,40	9
11,75	13
7,93	9
16,91	14
12,79	13
21,83	21
23,81	24
7,60	9
32,06	28
23,83	24
12,58	13
23,62	21
7,17	9
20,44	21
7,49	9
9,56	10
8,22	9
21,51	21
9,14	9
6,62	5
7,47	9
11,12	4
22,84	24
8,41	9
13,84	14
27,41	26
6,67	5
23,85	24
13,55	13
11,45	13
14,20	14
8,30	9
8,39	9
18,89	19

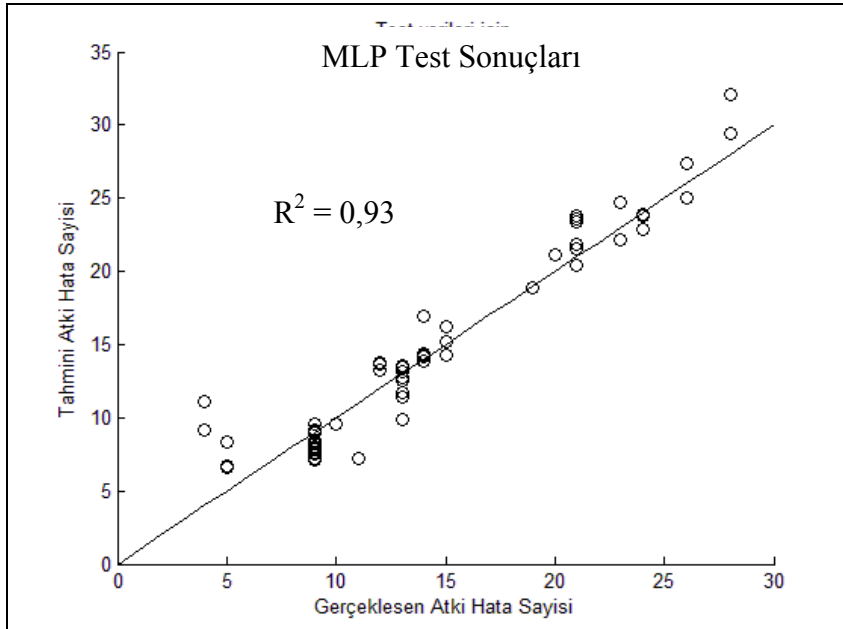
Eđitim kümesi için ayırdığımız 192 adet verinin gerçek çıktısı (atkıdaki hata sayısı) ile MLP ağının çıktısı (tahmini atkı hata sayısı) arasındaki ilişki Grafik 4.2'de gösterilmiştir.

**Grafik 4.2.** Eğitim verileri için gerçek ile tahmini atkı hata sayısı



Test kümesi için ayırdığımız 64 adet verinin gerçek çıktısı (atkıdaki hata sayısı) ile MLP ağının çıktısı (tahmini atkı hata sayısı) arasındaki ilişki Grafik 4.3’de gösterilmiştir.

**Grafik 4.3.** Test verileri için gerçek ile tahmini atkı hata sayısı



Atkı hata sayısının tahmini için geliştirdiğimiz modelde bulduğumuz determinasyon katsayısı(  $R^2$  ) değeri, yapılan tahminin doğruluğuna karar vermek için kullanılır. Çok katmanlı algılayıcı modelimizde sırasıyla eğitim verilerinde  $R^2=0,94$  test verilerinde ise  $R^2=0,93$  olarak bulunmuştur. Determinasyon katsayısı değerinin yüksek olması yaptığımız tahmininin başarılı olduğunu ve tahminin doğruluğunu göstermektedir.

#### 4.4.2. Radyal Tabanlı Fonksiyon Ağ Modeli

Bu model için de ilk olarak Excel sayfasında yer alan çalışmamıza ilişkin 320 adet veri Excel programından Matlab programına aktarılmıştır. Daha sonra da verilerin %60 'ı eğitim verisi, %20'si doğrulama verisi ve %20'si de test verisi olarak ayrılmıştır. Bu bölme işlemi de rasgele yapılmıştır.

Veriler belirlendikten sonra eğitim sırasında kullanacağımız bir ağ modelinin oluşturulması gerekir. Bu modelin giriş katmanında on üç nöron, çıkışta bir nöron vardır. Bu ağ yapısında daha önce de değindiğimiz gibi gizli katman sayısı her zaman birdir. Bu ağ yapısında da belirlenmesi gereken genişlik(spread parameter) ve merkez değer parametreleridir. Söz konusu parametreler çalışmamızda *deneme yanılma yoluyla* bulunmuştur.

Genişlik parametresi sekiz olarak belirlenmiştir. Bu parametre değerleri 1, 10, 100, 1000, 10000, 100000, 1000000, 10000000 dir. Gizli katmandaki nöron sayısını belirlemek için de 50' den başlayarak ve 50 arttırarak 250'ye kadar denemeler yapılmıştır. Dolayısıyla çalışmamızda nöron sayısı 50, 100, 150, 200 ve 250 olarak alınmıştır. 250 değerini almamızın temel nedeni ise eğitim ve doğrulama verileri toplamının 256 olmasıdır. Katman sayısı belirli olan ağ modeli için belirlediğimiz bu parametrelerden sonra program çalıştırılmıştır. Programda belirlenen her bir genişlik değerine karşılık 50'den 250'ye kadar olan nöronlar da tek tek denenerek Matlab programı ile en iyi model ve buna ilişkin parametreler belirlenmiştir.

Modelde giriş ile gizli katman arasında kullanılan transfer fonksiyonu radyal tabanlı olan *gauss fonksiyonudur*. Gizli katman ile çıkış katmanı arasında kullanılan transfer fonksiyonu ise *doğrusal* (pureline) fonksiyondur.

Çalışmamızda kullanılan RBFN modeli yapay sinir ağ mimarisi olarak ileri beslemeli sinir ağ modelidir. Öğrenme tipi olarak ise aslında danışmanlı öğrenmeyi kullanmakla birlikte başlangıçta merkez vektörünün belirlenmesi aşamasında danışmansız öğrenmeyi kullandığından dolayı karma bir öğrenme süreci kullandığı söylenebilir. Performans kriteri, MLP’de olduğu gibi *hata kare ortalaması*(MSE) değeri olup amaç ise bu değerini sıfıra yaklaşmasıdır.

Parametrelere uygun model oluşturulduktan ve eğitildikten sonra modelin sonuçları test edilerek en uygun model belirlenir. Çalışmamızda hata performansı olarak MSE, RMSE ve MAE ölçütleri kullanılmıştır. Bu değerlerden hata kare ortalamasının(MSE) en küçük olduğu değer ile determinasyon katsayısının ( $R^2$ ) en büyük olduğu satırı veren model en iyi çözümü veren ağ modeli olur.

Aşağıdaki Tablo 4.11 en uygun RBFN modelini belirlemek için yapılan deneme sonuçlarını gösterir.

**Tablo 4.11.** En uygun RBFN modelini belirlemek için yapılan denemeler

Uygun Genişlik	Gizli Katmandaki Nöron Sayısı	R	R <sup>2</sup>	MSE	RMSE	MAE
S=1	50	0,42	0,18	58,11	7,62	6,03
	100	0,42	0,18	73,99	8,60	6,90
	150	0,39	0,15	91,41	9,56	7,87
	200	0,34	0,12	134,89	11,61	10,03
	250	0,39	0,15	66,32	8,14	7,45
S=10	50	0,58	0,33	47,79	6,91	5,40
	100	0,55	0,31	60,39	7,77	5,86
	150	0,49	0,24	80,34	8,96	6,66
	200	0,41	0,17	98,11	9,90	7,26
	250	0,76	0,58	23,11	4,81	3,63
S=100	50	0,95	0,90	5,31	2,30	1,79
	100	0,95	0,90	5,54	2,35	1,66
	150	0,86	0,74	16,47	4,06	2,57
	200	0,71	0,50	46,53	6,82	4,05
	250	0,74	0,55	32,95	5,74	4,12
S=1000	50	0,92	0,84	8,86	2,98	2,42
	100	0,89	0,80	11,20	3,35	2,36
	150	0,87	0,75	15,79	3,97	2,72
	200	0,68	0,47	64,88	8,05	5,76
	250	0,35	0,12	744,19	27,28	14,08
S=10000	50	0,90	0,81	10,35	3,22	2,62
	100	0,89	0,80	10,65	3,26	2,44
	150	0,88	0,77	12,68	3,56	2,74
	200	0,88	0,77	12,77	3,57	2,68
	250	0,88	0,78	12,00	3,46	2,58
S=100000	50	0,94	0,88	6,47	2,54	2,25
	100	0,94	0,89	6,06	2,46	2,18
	150	0,94	0,88	6,57	2,56	2,32
	200	0,94	0,89	6,26	2,50	2,24
	250	0,94	0,88	6,32	2,51	2,24
S=1000000	50	0,85	0,72	14,04	3,75	2,92
	100	0,85	0,72	14,04	3,75	2,92
	150	0,85	0,73	14,00	3,74	2,92
	200	0,85	0,72	14,04	3,75	2,92
	250	0,85	0,72	14,04	3,75	2,92
S=10000000	50	0,85	0,72	14,04	3,75	2,91
	100	0,85	0,72	14,03	3,75	2,91
	150	0,85	0,72	14,05	3,75	2,92
	200	0,85	0,72	14,04	3,75	2,91
	250	0,85	0,72	14,04	3,75	2,92

Yapılan denemeler sonucunda en uygun modelin 50-100 ağ yapılı model olduğu belirlenmiştir. Çünkü yukarıdaki Tablo 4.11’de görüldüğü üzere en küçük MSE değeri ve

en büyük  $R^2$  değerine bu ağ yapısında ulaşılmıştır. Elde edilen MSE değeri 5,31 iken  $R^2$  değeri ise 0,90'dır.

Sinir ağının eğitimi tamamlanınca ağ öğrenme işlemini sonlandırır. Ağın öğrenmesindeki amaç, ağdaki merkez değerlerin ve ağırlık katsayılarının bulunmasıdır. Burada yine ara katmandaki 50 tane nöron ve çıkış katmanında bulunan bir nöronun eşik değeri de belirlenmiştir. Şimdi ağın öğrenme işleminin tamamlanmasıyla belirlenen merkez değerleri, ağırlık katsayıları ve eşik değerleri aşağıdaki tablolarda gösterilmiştir.

Giriş nöronları ile gizli katman nöronları arasındaki merkez değerleri Tablo 4.12'de verilmiştir.

**Tablo 4.12.** Giriş nöronları ile gizli katman nöronları arasındaki merkez değerleri

	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13
u1	271	180	144	26	0,88	94,4	30	75	54	2	1	3	7
u2	468	103	164	30	0,91	57,2	30	30	54	3	1	3	3
u3	165	261	133	26	0,91	13,8	30	75	54	1	1	1	2
u4	278	105	174	26	0,91	40,2	30	150	40	1	1	3	5
u5	227	266	157	24	0,88	72,8	30	75	54	1	2	1	3
u6	268	155	136	24	0,84	21,6	30	150	40	4	1	3	7
u7	256	193	137	24	0,85	13,4	40	150	54	3	1	1	3
u8	91	251	136	24	0,91	11,1	30	75	54	1	1	1	3
u9	278	104	174	26	0,91	40,2	30	30	54	1	1	3	3
u10	468	249	164	24	0,87	57,2	30	150	54	3	2	1	2
u11	462	164	130	24	0,88	32,5	30	75	54	2	2	1	3
u12	296	259	159	26	0,87	66,6	30	30	54	1	2	1	1
u13	290	150	139	24	0,9	17,7	30	30	54	2	1	1	3
u14	458	104	171	30	0,9	43,8	30	75	54	3	1	4	9
u15	402	185	144	30	0,86	92,9	30	150	54	3	1	3	3
u16	137	257	134	26	0,89	12,6	20	75	54	3	3	4	3
u17	271	181	143	26	0,9	95,5	30	75	54	2	1	3	3
u18	252	247	137	24	0,9	10,1	30	150	40	2	1	1	3
u19	227	107	157	24	0,91	72,8	30	150	54	1	2	1	3
u20	206	269	130	24	0,9	16,9	30	75	54	4	1	3	4
u21	137	257	134	26	0,87	12,6	30	75	54	1	2	2	3
u22	281	254	135	24	0,87	11,9	30	75	54	1	1	2	1

u23	125	278	127	24	0,88	21,4	30	75	54	3	1	1	3
u24	324	183	143	24	0,9	95,8	30	150	54	1	1	1	6
u25	191	271	129	26	0,89	17,9	20	75	54	1	3	1	1
u26	458	154	171	26	0,86	44	30	75	54	2	1	3	3
u27	444	103	152	30	0,9	91,1	30	150	40	1	1	1	3
u28	435	109	146	30	0,91	12	30	75	40	2	1	1	3
u29	264	162	132	26	0,84	29,6	30	30	54	2	2	1	4
u30	325	167	129	26	0,88	36,3	30	75	54	2	1	1	6
u31	375	185	142	24	0,85	10,4	30	75	54	3	1	1	3
u32	457	181	143	24	0,88	99,5	20	75	54	3	2	3	3
u33	470	154	137	30	0,84	20,7	30	75	54	2	1	1	6
u34	296	100	159	26	0,91	66,6	20	75	54	3	1	3	1
u35	88	258	134	26	0,91	13	30	75	54	2	2	1	3
u36	448	247	156	30	0,9	76,2	30	75	54	1	1	1	3
u37	455	190	139	24	0,9	12,3	30	75	54	2	1	1	2
u38	255	265	131	30	0,89	15,5	30	30	54	1	1	2	4
u39	167	182	146	24	0,91	83,6	40	75	40	1	2	3	3
u40	220	156	136	26	0,86	22,6	30	150	54	1	2	1	3
u41	347	109	145	24	0,88	12,6	30	30	54	1	1	3	2
u42	157	249	137	26	0,87	10,6	30	75	54	1	1	1	4
u43	121	165	132	26	0,88	30,1	20	75	54	2	2	3	3
u44	138	191	139	30	0,88	12,2	30	75	54	4	1	1	3
u45	173	246	138	24	0,9	99,7	30	75	40	1	1	1	3
u46	162	166	129	24	0,82	36,3	30	75	54	3	1	3	3
u47	281	254	135	24	0,87	11,9	30	75	54	3	1	1	3
u48	138	269	130	24	0,89	16,7	30	75	54	1	1	1	1
u49	136	159	134	30	0,85	25,5	30	75	54	4	1	1	3
u50	227	105	150	24	0,83	10,1	30	75	54	1	1	2	3

Gizli katmandaki nöronların eşik değerleri aşağıdaki Tablo 4.13’de verilmiştir.

**Tablo 4.13.** Gizli katmandaki nöronların eşik değerleri

Gizli katmandaki nöronlar	Eşik değeri
1	0,008326
2	0,008326
3	0,008326
4	0,008326
5	0,008326
6	0,008326



7	0,008326
8	0,008326
9	0,008326
10	0,008326
11	0,008326
12	0,008326
13	0,008326
14	0,008326
15	0,008326
16	0,008326
17	0,008326
18	0,008326
19	0,008326
20	0,008326
21	0,008326
22	0,008326
23	0,008326
24	0,008326
25	0,008326
26	0,008326
27	0,008326
28	0,008326
29	0,008326
30	0,008326
31	0,008326
32	0,008326
33	0,008326
34	0,008326
35	0,008326
36	0,008326
37	0,008326
38	0,008326
39	0,008326
40	0,008326
41	0,008326
42	0,008326
43	0,008326
44	0,008326
45	0,008326
46	0,008326
47	0,008326
48	0,008326
49	0,008326
50	0,008326

Gizli katmandaki nöronlar ile çıkış katmanındaki nöronun ağırlık değerlerini veren bilgisayar çıktısı aşağıdaki Tablo 4.14’de gösterilmiştir.

**Tablo 4.14.** Gizli katmandaki nöronlar ile çıkış katmanındaki nöronların ağırlık değerleri

Gizli katmandaki nöronlar	Ağırlık değerleri
c1	1199,132
c2	27,44452
c3	1759,784
c4	39,54302
c5	41,61456
c6	-11,585
c7	-81,9789
c8	1515,908
c9	67,52629
c10	32,81669
c11	152,4108
c12	12,56312
c13	-130,46
c14	-79,3003
c15	-56,3839
c16	105,7111
c17	-1232,66
c18	59,74389
c19	-1,51045
c20	-469,319
c21	386,7032
c22	2461,744
c23	1189,051
c24	35,81417
c25	-108,089
c26	161,2495
c27	40,88006
c28	11,90111
c29	57,27134
c30	35,12385
c31	-21,6654
c32	-106,293
c33	-111,34
c34	-31,8373
c35	-1352,58
c36	52,58375
c37	-118,183
c38	22,73717
c39	-115,098
c40	26,22306
c41	28,73236
c42	-260,134
c43	-31,3204
c44	-540,155
c45	53,10135

c46	157,6324
c47	-2394,91
c48	-2605,79
c49	234,3321
c50	-19,1381

**Tablo 4.15.** Çıktı katmanındaki nöronun eşik değeri

Çıktı katmanındaki nöron	Eşik değeri
1	1,380864

Ağın eğitiminin tamamlanmasıyla ağın öğrenip öğrenmediğini bir başka ifadeyle performansını ölçmek için denemeler yapılarak ağın testi yapılır. Ağı test etmek için ağın eğitimi sırasında görmediği, yani test etmek amacıyla ayırmış olduğumuz test kümesindeki veriler kullanılır. Bu veriler ağa gösterilir ve ağ eğitimi sırasında belirlemiş olduğu ağırlık ve eşik değerlerini kullanarak daha önce görmediği bu veriler için çıktılar üretir.

Kullandığımız Matlab R2010b programı ile Tablo 4.11’de elde edilen  $R^2 = 0,90$  değeri yani %90 doğrulukla test kümesindeki verileri tahmin ettiğimizde şirketin gerçek verileri ile ağın ürettiği değerler aşağıdaki Tablo 4.16’da verilmiştir.

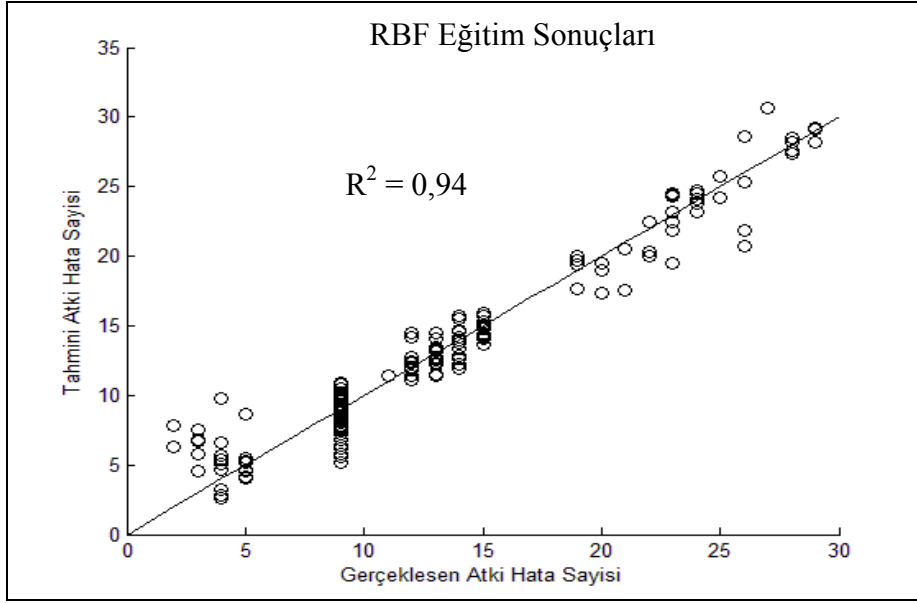
**Tablo 4.16.** RBF ağının çıktıları ile gerçek veriler

Test Kümesi	
RBF ağının çıktısı	Gerçek veri( Atkıdaki hata sayısı)
9,651740496	9
14,99339952	14
16,53567787	20
9,997203971	9
14,62255424	15
21,24177738	23
12,11019118	12
8,071248673	9
14,35176973	15
28,89026222	26
12,85388045	14
16,56608074	21
28,19339638	28
25,57973008	23
9,501106022	9
4,350525542	4
13,48357268	13

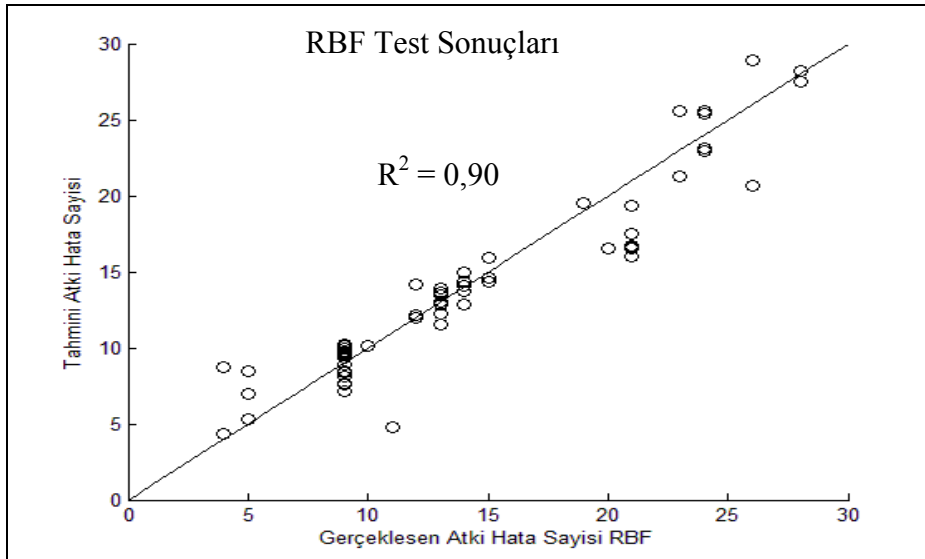
12,23295846	13
7,572356112	9
12,98280059	13
15,92122225	15
14,12776671	12
14,33109999	15
16,69881856	21
11,97268415	12
5,328680147	5
10,09516941	9
9,630523083	9
8,364383063	9
4,755267607	11
10,14818329	9
13,86218111	13
9,383613461	9
13,71106515	14
11,50850228	13
16,64160852	21
23,13641795	24
7,693368424	9
27,484608	28
25,36873876	24
12,87291914	13
17,49501835	21
9,325812435	9
19,3610148	21
8,477380412	9
10,09345436	10
9,809998093	9
16,00697268	21
10,17438267	9
8,455793421	5
7,179977445	9
8,768028932	4
25,54202604	24
10,25665318	9
14,11308636	14
20,68332868	26
6,946097296	5
22,92245386	24
13,59704677	13
13,43188849	13
14,35531373	14
8,888854321	9
8,366055645	9
19,5353755	19

Eđitim ve test kumesine iliřkin gerek ıktı (atkıdaki hata sayısı) ile RBF ađının ıktısı(tahmini atkı hata sayısı) arasındaki iliřkiyi gosteren grafikler ařađıda verilmiřtir.

**Grafik 4.4.** Eđitim verileri iin gerek ile tahmini atkı hata sayısı



**Grafik 4.5.** Test verileri iin gerek ile tahmini atkı hata sayısı



řirketin kumař üretiminde atkı hata sayısı tahmini iin geliřtirdiđimiz modelde yapay sinir ađ modelinden olan radyal tabanlı fonksiyon ađ modelinde sırasıyla eđitim

verilerinde  $R^2=0,94$  test verilerinde ise  $R^2=0,90$  olarak bulunmuştur. Determinasyon katsayı( $R^2$ ) değerinin yüksek olması yapılan tahminin doğruluğunu göstermektedir.

Modelimizde daha önce sözünü ettiğimiz girdi değişkenlerini ele alarak bunlara karşı çıktı değeri alan atkı hatasının % 90 doğrulukla tahmin edebildiği sonucuna ulaşılmıştır. Daha sonra firmadan aldığımız ve test kümesi olarak ayırdığımız verilerle, ağın %90 tahmin ile ürettiği çıktılar kıyaslanmıştır. Bu kıyaslama Tablo 4.16'da verilmiştir.

Özetle MLP ve RBF ağ modellerin tahmin sonuçları işletme yönetiminin müşteri siparişleri için kumaş üretiminde girdi değişkenlerinin bazılarını değiştirerek üreteceği kumaşlarda ne kadar atkı hatasının olacağını belirlemede yardımcı olur. Böylece de kumaş üretiminde oluşacak atkı hata sayısı azalarak işletmenin kumaş üretim verimliliği ve kalitesi artarak müşterisine değer sağlayan ürünler sunar. Bir bakıma çalışmamızın bu bölümünde ele aldığımız yapay sinir ağ modelleri işletme yönetimine planlama aşamasında hangi top kumaşlarda ne kadar atkı hatası olabileceğinin tahminini verir.

Sonuç olarak ele aldığımız üç modele ilişkin determinasyon( $R^2$ ) katsayısı ve hata kareleri ortalaması (MSE) değerlerini incelediğimizde; çoklu doğrusal regresyon(MLR) modeline ilişkin  $R^2= 0.71$ ,  $MSE=13.52$ , çok katmanlı algılayıcı (MLP) modeli için  $R^2= 0.93$ ,  $MSE= 4.21$  ve radyal tabanlı fonksiyon ağ (RBFN) modeline ilişkin  $R^2= 0.90$ ,  $MSE= 5.31$  sonuçlarına ulaşılmıştır. Bu sonuçlardan da anlaşılıyor ki şirket için en iyi model MLP modelidir. Çünkü MLP modeli ile yapılacak tahminler %93 oranında başarılı olacaktır.

## SONUÇ

Tahmin, gelecekte neyin nasıl olacağını önceden öngörülmesi olup kesinliği olmayan bir süreçtir. Neredeyse tüm yönetsel kararlar ileriye yönelik tahminlere dayanır. Dolayısıyla firmalar için gelecekteki belirsiz durumların tahmini verilecek kararın güvenilirliği açısından önemlidir. Dolayısıyla uygulamasını yaptığımız işletme probleminin çözümünde bir tahmin aracı olarak MLR, MLP ve RBFN modelleri kullanılmıştır.

Şimdi çalışmamızda kullanılan model uygulamalarının sonuçlarına değinelim. İlk modelimiz olan çoklu doğrusal regresyon modelini uygulamadan önce firmadan elde edilen verilerin normal dağılıma uygun olup olmadığı SPSS 13 paket program yardımıyla incelenmiştir.

Ayrıca modelimizde yer alan on üç değişken arasında çoklu doğrusal bir ilişkinin var olup olmadığı incelenmiştir. Katsayılar Tablosundan elde ettiğimiz VIF değerleri 10 dan küçük olduğundan bağımsız değişkenler arasında çoklu doğrusal bağlantı problemi olmadığı sonucuna varılmıştır.

Veri dağılımının normal olduğu ve bağımsız değişkenler arasında çoklu doğrusal bağlantı problemi olmadığı sonucuna varıldıktan sonra çoklu doğrusal regresyon modeli işletmenin atkı hata problemine uygulanmıştır. İşletmenin atkı hatasının 0,71'lik kısmı bağımsız değişkenler olan kumaşın metresi, makine devir hızı, kumaşın eni, çözgü sıklığı, salon randımanı, iplik mukavemeti, atkı no, çözgü no, atkı sıklığı, atkı cinsi, çözgü cinsi, tezgâh tipi ve örgü tipi tarafından açıklanmıştır. Varyans analizi tablosundan elde ettiğimiz F tablo değeri ile modelin bir bütün olarak anlamlı olduğu sonucuna varılmıştır. Daha sonra modele ilişkin tahmin yapılarak bağımsız değişkenlerin katsayı değerleri bulunmuş ve t testi yapılmıştır. Bu test ile her değişkenin bireysel olarak atkıdaki hata sayısı üzerinde etkili olup olmadığına bakılmıştır. Buna göre kumaşın metresi, makine devir hızı, salon randımanı, iplik mukavemeti, atkı iplik no, atkı iplik cinsi, tezgâh tipi ve örgü tipi değişkenlerinin atkıdaki hata sayısı üzerinde etkili olduğu sonucuna varılmıştır.

Modelin atkıdaki hata üzerine etkili olan deęişkenlere ilişkin elde edilen sonuçlara baktığımızda; üretilen kumaşın metresindeki 1 santimetrelik artış ortalama atkı sayısı hatasında 0,008 azalmasına neden olurken makine devir sayısındaki 1 dakikalık artış hızı (normal hızından fazla) ortalama atkı sayısındaki hatanın 0,13 artmasına neden olur. Salon randımanındaki yüzde 1'lik artış ortalama atkı sayısındaki hatanın 24,007 azalmasına neden olur. İplik mukavemetindeki 1 rkm'lık artış beklenen atkı sayısı hatasını 0,019 azaltmaktadır. Atkı iplik no'sundaki 1 denyelik artış beklenen atkı sayısındaki hatanın 0,189 azalmasına neden olmaktadır. Öte yandan, polypamuk atkı iplięi pamuk atkı iplięine göre 0,732 adet daha az atkı sayısında hataya neden olurken, pamuk polyester atkı iplięi pamuk atkı iplięine göre 1,464 adet daha az atkı sayısında hataya neden olur. Polyester atkı iplięi ise pamuk atkı iplięine göre 2,196 adet daha az atkı hata sayısına neden olur.

Çok deęişkenli regresyon modelimizin kumaş dokumada kullanılan tezgâhlar hakkında elde ettiğimiz sonuçları şöyle açıklayabiliriz. Kumaş dokunurken tezgâh tipi 2 (Sulzer) tezgâh tipi 1'e (CTP) göre 0,438 adet daha az atkı hatasına neden olurken, tezgâh tipi 3 (Somet Excell) tezgâh tipi 1'e (CTP) göre 0,876 adet daha az hatalı atkı sayısına neden olmaktadır. Tezgâh tipi 4 (Somet Alpha) tezgâh tipi 1'e (CTP) göre 1,314 adet daha az atkı sayısında hataya neden olur. Bu da yönetime hangi tezgâhların daha az hatalı kumaş üretimi yapacağına ilişkin kararlarında yardımcı olabilir.

Örgü tipi dokumada kullanılan iplik cinslerine göre modelimizin verdiği sonuçlardan bazıları ise; üretimde örgü tipi 1 (1/1 bez) yerine örgü tipi 2 (2/1 dimi) iplik kullanıldığında kumaşlardaki atkı hata sayısında 0,375 adet bir azalma olur. Örgü tipi 3 (2/1 herringbone) ise örgü tipi 1'e (1/1 bez) göre 0,750 adet daha az atkı sayısında hataya neden olmaktadır. Örgü tipi 4 (2/2 dimi) örgü tipi 1'e (1/1 bez) göre 1,125 adet daha az atkı sayısında hataya neden olur. Buna ilişkin bilgiler çalışmamızın dördüncü bölümünde ayrıntılı olarak verilmiştir.

İşletmeden aldığımız verilere göre ikinci modelimiz olan çok katmanlı algılayıcı (MLP) modeli kurulmuş ve model eğitildikten sonra sonuçları test edilerek en uygun model belirlenmiştir. En iyi modeli belirlemede hata performansı olarak MSE, RMSE ve MAE ölçütleri kullanılmıştır. Yaptığımız denemeler sonucunda en küçük MSE değeri 4,21 ve en yüksek R<sup>2</sup> değeri 0,93 olan gizli katmandaki nöron sayısı 6 olan ağ modeli en uygun modeldir. Dolayısıyla bu değerlere göre en uygun modeli veren ağ yapısı olarak 13-6-1 ağ



yapılı model seçilmiştir. Bu da bize modelimizin 0,93 doğruluk ile işletmenin başarılı tahmin yapabileceğini gösterir.

Çalışmamızın üçüncü modeli olan radyal tabanlı fonksiyon ağ(RBFN) modeli de MLP modeli gibi yapay sinir ağ mimarisi olarak ileri beslemeli sinir ağ modeli yapısındadır. Her iki yapay sinir ağ modeli için en uygun modelin belirlenmesinde Matlab R2010b programı kullanılmıştır. Bu program ile yapılan denemeler sonucunda en uygun modelin 50-100 ağ yapılı model olduğu belirlenmiştir. Bu modeli seçmemizdeki etken yapmış olduğumuz denemeler sonucunda ulaştığımız en küçük MSE değeri olarak 5,31 ve en büyük  $R^2$  değerinin ise 0,90 olmasıdır.

Çalışmamızda yer alan bu üç modelin tahmin performanslarını kıyasladığımızda çoklu doğrusal regresyon(MLR) modeli için  $R^2= 0.71$ ,  $MSE=13.52$ , çok katmanlı algılayıcı(MLP) modeli için  $R^2= 0.93$ ,  $MSE= 4.21$  ve radyal tabanlı fonksiyon ağ (RBFN) modeli için de  $R^2= 0.90$ ,  $MSE= 5.31$  dir. Bu sonuçlardan da görülüyor ki şirket için en iyi model MLP modelidir. Eğer şirket kumaş üretim planlamasında MLP modelini kullanırsa tahminlerinde %93 oranında başarılı olacaktır.

Çalışmamızda görüldüğü üzere yapay sinir ağ modelleri bir araç olarak kullanıldığında firmalar müşteri siparişini üretmeden önce ürün özelliklerini ve hata sayılarını tahmin edebilir. Bu sayede alınan bir takım önlemler ile yönetim doğru kararlar alarak kaynakların doğru yere kanalize edilmesini ve üretim için gerekli olan hammaddelerin önceden tedarikini belirleyebilir. Böylece firma hem üretimden kaynaklanan hata maliyetlerini hem de stok maliyetlerini düşürebilir.

Sonuç olarak çalışmamız teorik ve uygulamalı çalışmalar yapan kişilere yardımcı olabileceği gibi uygulamasını yaptığımız firmanın kumaş planlama aşamasında hangi top kumaşlarda ne kadar atkı hatası olabileceğini girdi değişkenlerinin bazılarını değiştirerek üretilecek kumaşlarda ne kadar atkı hatasının olacağını belirlemede yardımcı olabilir. Bu da işletmelerin atkı sayısından kaynaklanan kumaş hataları giderilerek kumaşın üretim verimliliğinin ve kalitesinin artmasına neden olur. Dolayısıyla da firma müşterilerine hatasız kumaşlar sunarak müşteri tatminini artırır.

## KAYNAKLAR

- ABU-MOSTAFA, Y.S, Neural Networks for Computing, In Proceedings of the American Institute of Physics Meeting, New York, 1986.
- AKPINAR, Haldun, “Yapay Sinir Ağları Gelişimi ve Yapılarının incelenmesi”, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, Cilt:23, Sayı:1,1994.
- ANDERSON, Dave-MCNEIL, George, Artificial Neural Networks Technology, Rome Laboratory, A DACS State-of-the-Art Report, ELIN: A011, 1992.
- ARABACI, Özer, Makroekonomik Zaman Serisi Analizi ve Yapay Sinir Ağı Uygulamaları, Uludağ Üniversitesi, Bursa, 2007.
- AŞKIN, Dildade-İSKENDER, İres-MAMIZADEH, Ali, “Farklı Yapay Sinir Ağları Yöntemlerini Kullanarak Kuru Tip Transformator Sargısının Termal Analizi”, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, Cilt:26, No: 4, 2011.
- BAKAR, Akif – AYDIN, Celal, İstatistik, Nobel yayınevi, Ankara, 2008.
- BAHADIR, M. Çağatay–BAHADIR, Senem K.- KALAOĞLU, Fatma, “An Artificial Neural Network Model for Prediction of Bursting Strength of Knitted Fabrics”, International Conference on Machine Learning and Computer Science, Puket(Tayland), 11-12 Ağustos 2012.
- BAYRAM, Nuran, Veri Analizi: Excel ve SPSS Uygulamalarıyla Birlikte, Siyasal Kitabevi, Bursa, 2012.
- BEALE, Mark-HAGAN, Martin-DEMUTH, Howard ,“Neural Network Toolbox for Use with MATLAB”, User’s Guide. The Math Works Inc., 2012.
- BELTRAN, Rafael –WANG, Lijing-WANG Xungai, “Predicting the Pilling Tendency of Wool Knits”, *The Journal of the Textile Institute*, Vol. 97, No.2, 2006.
- CAUDILL, M. “Neural Network Primer Part 1”, AI Expert, 1987.
- DANIEL, Graupe, Principles of Artificial Neural Networks, 2nd Edition. World Scientific Publishing Co. Pte. Ltd., Singapore, 2007.
- EFE, M. Önder-KAYNAK, Okyay, Yapay Sinir Ağları ve Uygulamaları, Boğaziçi Üniversitesi, 2000.
- ELMAS, Çetin, Yapay Zekâ, Popüler Bilim, Sayı: 124, 2004.

- ERGEZER, Halit-DİKMEN, Mehmet- ÖZDEMİR, Erkan, Yapay Sinir Ağları ve Tanıma Sistemleri, Pivolka 6, ELYAD DAL Araştırma Laboratuvarları, Başkent Üniversitesi, Ankara, 2003.
- FAUSETT, Laurene, Fundamentals of Neural Networks, Prentice Hall International, London, 1994.
- FURFERI, Rocco-GELLI, Maurizio, “Yarn Strength Prediction: A Practical Model Based on Artificial Neural Networks”, *Advances in Mechanical Engineering*, Vol. 2010, Article ID 640103, 2010.
- GHAREHAGHAJI, Ali Akbar-SHANBEH, Mohsen- PALHANG, Maziar, “Analysis of Two Methodologies for Predicting the Tensile Properties of Cotton-Covered Nylon Core Yarns”, *Textile Research Journal*, Vol. 77, 2007, s.565-571.
- GUJARATI, Damodar N., Temel Ekonometri, çev. Ümit Şenesen-Gülay Günlük Şenesen, Literatür yayıncılık, İstanbul, 1995.
- GURUPRASAD, R.-BEHERA,B.K.,“Prediction of Bending of Woven Fabrics By Soft Computing”, 7<sup>th</sup> International Conference TEXSCI, Liberec (Çek Cumhuriyeti), 6-8 Eylül 2010.
- HAMZAÇEBİ, Coşkun, Yapay sinir Ağları, Ekin basımevi, Bursa, 2011.
- HAYKIN, Simon, Neural Networks: A Comprehensive Foundation, New Jersey, 1999,
- HEBB, Donald, O, The Organization of Behaviour. New York, 1949.
- HOPFIELD, John,J., “Neural Network and Physical Systems with Emergent Collective Computational Properties”, Proceedings of the National Academy of Science, Sayı 79, 1982.
- KAYNAR, Oğuz-TAŞTAN, Serkan-DEMİRKOPARAN, Ferhan, “Ham Petrol Fiyatlarının Yapay Sinir Ağları ile Tahmini”, *Ege Akademik Bakış*, 2010.
- KIM, Kwang Baek –KIM, Cheol Ki, “Performance Improvement of RBF Network Using ART2 Algorithm and Fuzzy Logic System” , Australia Conference on Artificial Intelligence, 2004.
- KİŞİ, Özgür-PARTAL, Turgay, “Yapay Sinir Ağları ve Otoregresif Modellerle Rüzgâr Hızı Tahmini”, *III. Atmosfer Bilimleri Sempozyumu Bildiri Kitabı*, İstanbul: 19-21 Mart 2003.
- KOUTSOYİANNİS, Anna, Ekonometri Kuramı, çev. Ümit Şenesen ve Gülay, G. Şenesen, İstanbul Teknik Üniversitesi Matbaası, İstanbul, 1992.
- KRISHNASWAMY, C. R.- GILBERT, Erika W.- PASHLEY, Mary M., Neural Network Applications in Finance: A Practical Introduction, Financial Practice and Education, 2000.

- LIPPMANN, Richard, P., An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 1987.
- MASTERS, Timothy, Practical Neural Network Recipes in C++, Academic Press, Toronto, 1993.
- MONTGOMERY, Douglas-PECK, Elizabeth-VINING, Geoffrey, Introduction To Linear Regression Analysis, Third Edition, A Wiley Interscience Publication, New York, 2001.
- OĞULATA, S. Noyan ve diğerleri, “The Prediction of Elongation and Recovery of Woven Bi-Stretch Fabric Using Artificial Neural Network and Linear Regression Models”, *Fibres and Textiles in Eastern Europe*, Vol.14, No.2, 2006.
- OKKAN, Umut –DALKILIÇ, H. Yıldırım, “Radyal Tabanlı Yapay Sinir Ağları ile Kemer Barajı Aylık Akımlarının Modellenmesi”, *İMO Teknik Dergi*, 2012.
- ÖZDAMAR, Kazım, Paket Programlar ile İstatistiksel Veri Analizi, Kaan Kitabevi, Eskişehir, 2004
- ÖZTEMEL, Ercan, Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul, 2003.
- ROSENBLATT, Frank, “The Perceptron: A Probabilistic Model for Information Storage and Organization In The Brain”, *Psychol. Rev.*, Sayı 65, 1958.
- SARAÇ, Tuğba, Yapay Sinir Ağları, Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Bölümü Anabilim Dalı, Ankara, 2004.
- SIMPSON, Patrick K., Artificial Neural Network, Pergamon Press, New York, 1989.
- SMITH, Kate, A.- GRUPTA, Jatinder, N.D., Neural Networks in Business: Tecniques and Applications , IRM Press, United Kingdom, 2002.
- ŞEN, Zekai, Yapay Sinir Ağları İlkeleri, Su Vakfı Yayınları, İstanbul, 2004.
- UZGÖREN, Nevin, “Uyum Analizinin Teorik Esasları ve Regresyon Analizi ile Benzerliğinin Grafiksel Boyutta Karşılaştırılması”, *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, Sayı 18, 2007.
- VEELENTURF, Leo, Analysis and Applications of Artificial Neural Networks, Prentice Hall, London, 1995.
- YURTOĞLU, Hasan, Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler için Türkiye Örneği, DPT Uzmanlık Tezi, Ankara, 2005.
- ZENG, Y.-WANG, K.-YU, C., “Predicting the Tensile Properties of Air-Jet spun Yarns”, *Textile Research Journal*, Vol. 74, No.8,2005.

ZHANG, Z. FRIEDRICH K. , VELTEN, K., “Prediction on Tribological Properties of Short Fibre Composites Using Artificial Neural Networks”, *Wear Journal*, Vol.252, 2002.

ZURADA, Jacek M., Introduction to Artificial Neural Systems, West Publishing Company, 1992

[tr.wikipedia.org/wiki/Yapay\\_sinir\\_ağları](http://tr.wikipedia.org/wiki/Yapay_sinir_ağları)

<http://cc.domaindx.com/hyuces/ysa/ogremealgrt.htm>.

## EKLER

### EK-1. Firma Verileri

	Kumaşın metresi	Makine devir hızı	Kumaşın eni	Çözümlü sıklık	Salon Randımanı	İplik Mukavemeti	Atkı no	Çözümlü no	Atkı sıklık	Atkı cinsi	Çözümlü cinsi	Tezgâh tipi	Örgü tipi	Atkı hatası
1	191	271	130	24	0,88	17,70	30	75	54	Pamuk	Pamuk	CTP	1/1 bez	12
2	191	271	129	26	0,89	17,90	20	75	54	Pamuk	Pamuk	CTP	1/1 bez	12
3	137	257	134	26	0,89	12,60	20	75	54	Pamuk Polyester	Polyester	Somet Alpha	2/1 herringbone	13
4	157	249	137	24	0,89	10,60	20	75	40	Pamuk Polyester	Polyester	Somet Excel	2/1 herringbone	15
5	268	247	133	24	0,88	13,30	20	75	40	Pamuk Polyester	Pamuk	Somet Excel	1/1 bez	13
6	268	247	133	26	0,88	13,40	20	30	40	Pamuk	Pamuk	Somet Excel	2/1 herringbone	13
7	307	247	137	24	0,90	10,00	20	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	15
8	307	247	137	24	0,87	10,00	20	75	54	Pamuk Polyester	Pamuk	Somet Excel	1/1 bez	15
9	233	266	131	26	0,87	15,80	30	30	40	Pamuk Polyester	Pamuk	Somet Alpha	2/1 herringbone	13
10	233	266	131	26	0,87	15,80	30	75	54	Pamuk	Polyester	Somet Excel	2/1 herringbone	13
11	146	254	135	24	0,90	11,80	30	30	54	Polyester	Pamuk	CTP	2/1 dimi	14
12	146	254	135	24	0,89	11,80	30	30	40	Pamuk	Pamuk	Somet Excel	2/1 herringbone	14
13	146	254	135	24	0,90	11,80	30	150	54	Polyester	Pamuk	CTP	2/1 herringbone	14
14	146	254	135	24	0,89	11,90	30	75	40	Pamuk	Pamuk	CTP	2'li 1/1 bez	14
15	173	246	138	24	0,89	99,40	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	1/1 bez	15
16	173	246	138	24	0,90	99,70	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	15
17	183	273	129	30	0,90	18,70	30	75	54	Polypamuk	Pamuk	CTP	2/2 dimi	12
18	165	261	129	26	0,90	13,80	30	75	40	Polypamuk	Pamuk	Sulzer	2/1 herringbone	13
19	165	261	133	24	0,90	13,90	30	30	54	Pamuk	Pamuk	Somet Alpha	2/1 dimi	13
20	88	258	134	30	0,88	13,00	30	30	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	13
21	91	251	136	24	0,90	11,10	30	30	40	Pamuk Polyester	Pamuk	Somet Alpha	2/1 herringbone	14

22	92	246	138	30	0,90	98,50	30	75	54	Polyester	Pamuk	Somet Excel	1/1 bez	15
23	125	278	127	24	0,88	21,40	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	11
24	335	254	135	30	0,90	11,60	30	30	40	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	14
25	335	253	135	30	0,90	11,70	30	150	54	Pamuk	Pamuk	Sulzer	2'li 2/2 dimi	14
26	252	251	136	24	0,90	11,00	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 dimi	14
27	242	252	136	26	0,89	11,10	30	150	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	14
28	252	247	137	24	0,89	10,10	30	75	40	Pamuk	Pamuk	Somet Excel	2'li 2/2 dimi	15
29	252	247	137	24	0,89	10,10	30	75	40	Pamuk	Pamuk	Sulzer	2/1 herringbone	15
30	255	265	131	30	0,89	15,50	30	30	54	Pamuk	Pamuk	Sulzer	2/2 dimi	13
31	255	266	131	24	0,87	15,60	30	75	54	Pamuk Polyester	Pamuk	Sulzer	2/1 herringbone	12
32	281	254	135	24	0,87	11,90	20	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	14
33	281	254	135	24	0,87	11,90	20	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	14
34	269	266	131	24	0,87	15,70	30	75	54	Polyester	Pamuk	CTP	1/1 bez	12
35	269	266	131	24	0,92	15,90	30	30	54	Pamuk	Pamuk	CTP	2/2 dimi	13
36	120	265	131	24	0,92	15,50	30	150	54	Polypamuk	Polyester	CTP	2/1 herringbone	13
37	120	265	131	26	0,92	15,50	30	30	40	Polypamuk	Pamuk	CTP	2/1 herringbone	13
38	138	268	130	24	0,92	16,70	30	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	12
39	138	269	130	26	0,92	16,70	30	150	54	Pamuk	Polyester	CTP	2/1 herringbone	12
40	110	267	131	24	0,92	16,20	30	75	40	Polyester	Pamuk	CTP	2/1 herringbone	12
41	110	267	131	26	0,90	16,30	30	30	54	Pamuk	Polyester	CTP	2/1 herringbone	12
42	206	269	130	24	0,90	16,90	30	75	54	Polyester	Pamuk	Somet Excel	2/2 dimi	12
43	206	269	130	24	0,90	17,00	30	75	54	Pamuk	Pamuk	CTP	2'li 2/2 oxford	12
44	196	256	134	30	0,90	12,30	30	150	54	Pamuk	Pamuk	CTP	2/2 dimi	14
45	196	256	134	24	0,90	12,40	20	30	54	Pamuk	Pamuk	CTP	3/1 twill	14
46	212	266	131	24	0,90	15,90	20	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	13
47	212	266	131	30	0,90	15,90	20	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	13
48	198	249	137	30	0,90	10,60	20	75	54	Pamuk Polyester	Pamuk	CTP	3/1 twill	15
49	198	249	137	26	0,90	10,60	20	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	15
50	136	276	128	26	0,88	20,10	30	75	40	Polypamuk	Polyester	Somet Excel	1/1 bez	12
51	122	266	131	30	0,91	15,80	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	13
52	227	107	151	24	0,91	99,90	30	75	54	Pamuk	Pamuk	Somet Alpha	2/2 dimi	21
53	227	105	150	24	0,91	10,10	30	30	54	Pamuk Polyester	Polyester	Sulzer	2/1 herringbone	21
54	227	107	157	24	0,91	72,80	30	150	54	Pamuk	Polyester	CTP	2/1 herringbone	23

55	299	100	155	30	0,91	79,90	30	75	40	Pamuk Polyester	Pamuk	Somet Excel	2/2 dimi	23
56	299	100	155	30	0,91	80,20	30	150	40	Pamuk	Pamuk	CTP	2/1 herringbone	23
57	468	103	164	30	0,91	57,20	30	30	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	26
58	468	103	163	30	0,90	57,90	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	25
59	458	104	171	30	0,90	43,80	30	75	54	Pamuk Polyester	Pamuk	Somet Alpha	armür	28
60	458	105	171	30	0,91	44,00	30	30	54	Polypamuk	Pamuk	CTP	2/1 herringbone	28
61	435	109	146	30	0,91	12,00	30	75	40	Polypamuk	Pamuk	CTP	2/1 herringbone	20
62	435	109	146	26	0,91	12,10	30	30	40	Pamuk	Pamuk	CTP	2/1 herringbone	19
63	444	103	152	24	0,91	90,20	30	75	54	Pamuk Polyester	Pamuk	Sulzer	2/1 herringbone	22
64	444	103	152	26	0,91	91,10	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	22
65	448	103	156	24	0,93	75,60	30	30	54	Pamuk	Polyester	Sulzer	2/1 herringbone	23
66	448	100	156	24	0,93	76,20	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	23
67	194	100	161	26	0,93	62,20	30	30	54	Pamuk	Polyester	Somet Alpha	2/1 herringbone	24
68	194	100	161	24	0,93	62,30	30	75	54	Pamuk	Pamuk	CTP	2'li 1/1 bez	24
69	224	110	145	24	0,91	12,80	30	75	54	Pamuk	Pamuk	CTP	2'li 1/1 bez	19
70	224	110	145	30	0,91	12,90	30	75	54	Polypamuk	Pamuk	Somet Excel	1/1 bez	19
71	351	108	147	24	0,91	11,70	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	20
72	358	104	151	24	0,91	96,00	30	75	54	Polypamuk	Pamuk	Somet Alpha	2/1 herringbone	21
73	358	105	151	30	0,91	97,20	20	30	54	Pamuk	Pamuk	Somet Alpha	armür	21
74	296	100	160	26	0,91	66,00	20	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	24
75	296	100	159	26	0,91	66,60	20	75	54	Pamuk Polyester	Pamuk	Somet Excel	1/1 bez	24
76	332	105	166	30	0,91	52,40	30	75	54	Pamuk	Pamuk	Somet Excel	2/2 dimi	26
77	332	105	166	26	0,91	52,60	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	26
78	332	104	172	24	0,91	42,30	30	30	40	Polypamuk	Pamuk	Somet Excel	2/2 dimi	28
79	278	105	174	26	0,91	40,20	30	150	40	Pamuk	Pamuk	Somet Excel	2'li 1/1 bez	29
80	278	104	174	26	0,91	40,20	30	30	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	29
81	335	105	150	26	0,91	10,20	30	75	54	Pamuk	Pamuk	Somet Excel	2/2 dimi	21
82	347	109	145	24	0,88	12,60	30	30	54	Pamuk	Pamuk	Somet Excel	2/1 dimi	20
83	341	109	158	24	0,88	70,50	30	30	54	Polypamuk	Pamuk	CTP	2/2 dimi	24
84	456	108	158	24	0,88	70,80	30	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	24
85	459	108	147	26	0,89	11,70	30	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	20
86	214	110	145	26	0,89	12,80	30	150	54	Pamuk	Pamuk	CTP	2'li 2/2 dimi	19
87	395	89	169	26	0,89	46,40	30	75	54	Polyester	Pamuk	Somet Excel	1/1 bez	27



88	324	182	143	24	0,90	95,30	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	4
89	324	183	143	24	0,90	95,80	30	150	54	Pamuk	Pamuk	CTP	2'li 2/2 dimi	4
90	443	194	137	24	0,90	14,30	30	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	5
91	455	190	139	24	0,90	12,30	30	75	54	Polypamuk	Pamuk	CTP	2/1 dimi	5
92	370	194	137	30	0,90	14,10	30	150	54	Pamuk	Polyester	CTP	2/1 dimi	5
93	370	195	137	24	0,88	14,30	30	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	5
94	138	196	136	30	0,88	14,70	30	30	40	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	5
95	138	191	139	30	0,88	12,20	30	75	54	Polyester	Pamuk	CTP	2/1 herringbone	5
96	173	184	142	24	0,88	10,10	30	150	40	Pamuk	Polyester	CTP	2/1 herringbone	4
97	173	185	142	30	0,88	10,20	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	4
98	264	198	136	24	0,88	15,10	30	75	54	Pamuk	Polyester	Somet Excel	2/1 herringbone	5
99	264	195	136	26	0,88	15,10	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	5
100	271	180	144	26	0,88	94,40	30	75	54	Polypamuk	Pamuk	Somet Excel	2'li 2/2 oxford	4
101	271	181	143	26	0,90	95,50	30	75	54	Polypamuk	Pamuk	Somet Excel	2/1 herringbone	4
102	434	188	139	26	0,90	12,20	30	75	54	Pamuk	Pamuk	CTP	2'li 2/2 oxford	5
103	434	189	139	24	0,88	12,40	30	30	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	5
104	337	196	140	24	0,88	12,40	30	30	54	Polyester	Polyester	Somet Excel	1/1 bez	5
105	337	190	138	24	0,88	12,80	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	5
106	457	181	143	24	0,88	99,50	20	75	54	Pamuk Polyester	Polyester	Somet Excel	2/1 herringbone	4
107	457	183	142	24	0,88	10,10	20	75	54	Pamuk	Pamuk	CTP	2'li 2/2 oxford	4
108	468	177	145	26	0,88	85,50	20	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	4
109	376	191	137	24	0,91	13,90	30	30	54	Pamuk Polyester	Polyester	CTP	2/2 dimi	5
110	376	193	137	24	0,91	14,10	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	5
111	165	193	139	24	0,91	12,70	30	75	54	Pamuk Polyester	Polyester	Somet Alpha	armür	5
112	165	190	138	24	0,91	12,70	30	30	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	5
113	167	182	146	24	0,91	83,60	40	75	40	Pamuk	Polyester	Somet Excel	2/1 herringbone	4
114	167	176	146	24	0,91	83,20	40	75	54	Polyester	Pamuk	CTP	3/1 twill	4
115	349	177	148	24	0,86	76,30	40	30	40	Pamuk	Pamuk	CTP	2/1 herringbone	4
116	349	174	147	24	0,86	76,40	30	75	54	Polypamuk	Pamuk	Somet Excel	3/1 twill	4
117	395	192	142	24	0,86	11,00	30	150	54	Polypamuk	Pamuk	Somet Excel	2/1 herringbone	5
118	395	186	140	24	0,86	11,30	30	150	54	Polypamuk	Pamuk	CTP	2/1 herringbone	4
119	402	185	144	30	0,86	92,90	30	150	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	4
120	402	180	144	24	0,86	93,30	30	75	54	Pamuk Polyester	Pamuk	Somet Alpha	2/1 herringbone	4

121	409	178	147	24	0,86	79,10	30	75	54	Pamuk	Polyester	Somet Alpha	2/1 herringbone	10
122	409	175	147	26	0,86	79,10	30	75	54	Polyester	Pamuk	CTP	2/1 herringbone	4
123	256	193	137	24	0,85	13,40	40	150	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	5
124	256	194	137	24	0,85	13,50	40	150	54	Polypamuk	Polyester	CTP	2/2 dimi	5
125	320	190	139	24	0,85	12,10	40	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	5
126	320	191	139	24	0,85	12,10	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	5
127	375	185	142	24	0,85	10,40	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	4
128	375	186	142	24	0,86	10,50	30	75	54	Polyester	Pamuk	CTP	2/1 herringbone	11
129	277	163	131	24	0,86	30,70	30	75	54	Polypamuk	Pamuk	Somet Excel	2/1 herringbone	9
130	310	156	136	24	0,86	22,40	30	75	54	Pamuk Polyester	Polyester	Somet Alpha	2/1 dimi	3
131	315	156	136	24	0,86	22,60	30	75	54	Pamuk Polyester	Pamuk	CTP	2/2 dimi	3
132	389	153	138	24	0,86	19,30	30	75	40	Pamuk	Pamuk	Somet Alpha	armür	9
133	391	154	137	24	0,87	20,60	30	150	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
134	351	165	130	24	0,90	33,00	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	9
135	351	165	130	24	0,90	33,80	30	150	54	Pamuk	Pamuk	CTP	armür	3
136	349	166	129	24	0,88	36,20	40	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
137	349	167	128	24	0,87	37,30	30	75	54	Pamuk	Polyester	CTP	armür	3
138	470	154	137	26	0,88	20,10	30	150	54	Polypamuk	Polyester	CTP	2/1 herringbone	9
139	470	154	137	24	0,89	20,70	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
140	270	164	130	24	0,89	32,80	40	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
141	270	165	130	24	0,90	33,20	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
142	290	150	140	24	0,92	17,50	30	75	54	Polypamuk	Pamuk	Sulzer	2/1 herringbone	9
143	290	150	139	24	0,90	17,70	30	30	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
144	232	164	130	26	0,88	32,30	30	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
145	232	164	130	26	0,87	32,60	30	30	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	9
146	136	159	134	24	0,87	25,30	30	75	54	Pamuk Polyester	Pamuk	Somet Alpha	2/1 dimi	9
147	136	159	134	24	0,88	25,50	30	150	54	Polypamuk	Pamuk	Somet Excel	2/1 herringbone	3
148	121	165	132	24	0,90	30,10	30	30	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
149	121	163	131	24	0,87	30,70	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	9
150	113	163	131	24	0,87	31,20	30	150	54	Pamuk Polyester	Polyester	CTP	1/1 bez	9
151	438	163	132	24	0,90	30,20	30	150	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
152	438	164	131	24	0,92	31,50	30	75	54	Polypamuk	Pamuk	Sulzer	1/1 bez	9
153	459	165	130	24	0,86	33,20	30	75	54	Polypamuk	Pamuk	CTP	2/2 dimi	9

154	459	166	130	24	0,86	34,80	20	150	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
155	460	157	135	24	0,85	23,60	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	9
156	465	153	137	24	0,85	20,00	20	30	54	Pamuk Polyester	Pamuk	CTP	2/1 dimi	9
157	393	153	138	24	0,85	19,50	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
158	393	153	137	24	0,85	20,10	30	75	54	Pamuk	Pamuk	Somet Excel	2/1 dimi	9
159	386	159	134	24	0,83	25,40	30	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
160	386	159	133	24	0,83	26,30	30	30	54	Polypamuk	Polyester	CTP	2/1 herringbone	9
161	365	163	132	24	0,83	29,90	30	75	54	Pamuk	Polyester	Somet Excel	2/1 herringbone	9
162	365	163	131	24	0,83	31,00	30	75	54	Pamuk	Pamuk	CTP	2/2 dimi	9
163	254	162	132	30	0,83	28,60	30	75	40	Pamuk	Pamuk	Somet Alpha	2/1 herringbone	9
164	254	162	132	24	0,83	28,90	30	30	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
165	163	160	133	24	0,82	27,30	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
166	163	161	133	24	0,82	27,40	30	150	54	Pamuk	Pamuk	CTP	2/1 dimi	9
167	150	154	137	24	0,82	20,50	30	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
168	150	154	137	24	0,82	20,50	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
169	408	161	132	24	0,82	28,30	30	75	54	Polypamuk	Pamuk	Somet Excel	2/1 dimi	9
170	408	161	132	24	0,88	28,50	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
171	412	161	132	26	0,85	28,10	30	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
172	412	161	132	26	0,85	28,30	20	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
173	220	156	136	26	0,86	22,60	30	150	54	Pamuk	Polyester	CTP	2/1 herringbone	9
174	163	157	135	24	0,86	23,10	30	30	54	Polyester	Pamuk	Somet Excel	2/1 dimi	9
175	160	158	134	24	0,86	24,90	30	150	54	Pamuk	Polyester	CTP	2/1 herringbone	9
176	264	162	132	24	0,84	29,50	30	75	54	Pamuk	Pamuk	CTP	2/1 dimi	9
177	264	162	132	26	0,84	29,60	30	30	54	Polypamuk	Polyester	CTP	2/2 dimi	9
178	233	160	133	26	0,84	26,90	30	150	54	Polypamuk	Polyester	CTP	2/1 herringbone	9
179	233	160	133	24	0,84	26,90	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	9
180	411	161	133	24	0,84	27,50	30	30	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	9
181	411	161	132	24	0,84	27,90	30	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	9
182	403	156	136	24	0,84	22,30	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	9
183	403	156	136	24	0,84	22,60	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
184	461	154	137	24	0,84	20,30	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
185	461	154	137	24	0,88	20,50	30	75	54	Pamuk	Pamuk	CTP	2/2 dimi	9
186	467	158	134	24	0,88	24,80	20	75	54	Pamuk Polyester	Pamuk	CTP	2/2 dimi	9

187	467	159	134	24	0,88	25,00	40	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
188	462	164	130	24	0,88	32,20	30	150	54	Pamuk	Pamuk	CTP	2'li 1/1 bez	9
189	462	164	130	24	0,88	32,50	30	75	54	Polypamuk	Polyester	CTP	2/1 herringbone	9
190	322	163	131	24	0,88	30,40	30	30	54	Polyester	Polyester	CTP	2'li 1/1 bez	3
191	322	163	131	24	0,88	30,60	20	75	54	Pamuk	Pamuk	CTP	2'li 2/2 dimi	3
192	325	167	129	26	0,88	36,30	30	75	54	Polypamuk	Pamuk	CTP	2'li 2/2 dimi	3
193	325	156	136	26	0,86	22,10	30	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	3
194	325	156	136	26	0,85	22,20	30	75	54	Pamuk	Pamuk	CTP	2/1 dimi	9
195	334	150	139	24	0,85	17,60	30	75	54	Polyester	Pamuk	CTP	2/1 herringbone	2
196	334	150	139	24	0,85	17,70	30	30	54	Pamuk	Polyester	Somet Excel	2/1 herringbone	9
197	347	161	132	26	0,82	27,90	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	2
198	347	161	132	24	0,82	28,00	30	150	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	9
199	162	166	129	24	0,82	36,30	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	9
200	160	163	131	26	0,84	30,50	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
201	165	152	138	24	0,84	18,80	30	75	54	Polyester	Polyester	CTP	2/1 herringbone	9
202	268	155	136	24	0,84	21,50	30	75	40	Polyester	Polyester	CTP	2/1 herringbone	9
203	268	155	136	24	0,84	21,60	30	150	40	Polyester	Pamuk	Somet Excel	2'li 2/2 oxford	9
204	268	155	136	26	0,84	21,60	20	75	54	Polypamuk	Pamuk	Somet Excel	2'li 2/2 oxford	9
205	268	155	136	26	0,84	21,60	20	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
206	268	259	133	26	0,84	13,30	30	150	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	13
207	146	254	135	26	0,91	11,90	30	75	40	Pamuk	Pamuk	CTP	3/1 twill	14
208	173	246	138	24	0,91	99,40	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	15
209	173	246	138	26	0,91	99,70	30	150	54	Polyester	Pamuk	CTP	2/1 herringbone	15
210	183	273	129	26	0,91	18,80	30	150	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	12
211	165	261	133	26	0,91	13,80	30	75	54	Pamuk	Pamuk	CTP	2/1 dimi	13
212	165	261	133	24	0,91	13,90	30	150	40	Pamuk Polyester	Pamuk	Somet Alpha	2/1 herringbone	13
213	88	258	134	26	0,91	13,00	30	75	54	Polypamuk	Polyester	CTP	2/1 herringbone	13
214	91	251	136	24	0,91	11,10	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	14
215	92	246	138	24	0,90	98,50	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	15
216	125	278	127	30	0,85	21,40	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	11
217	335	254	135	24	0,85	11,60	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	14
218	335	253	135	24	0,85	11,70	30	150	40	Polypamuk	Pamuk	CTP	2/1 herringbone	14
219	252	251	136	24	0,85	11,00	30	75	40	Polypamuk	Pamuk	CTP	2/1 herringbone	14

220	242	252	136	24	0,90	11,10	30	75	40	Pamuk	Polyester	Sulzer	2/1 herringbone	14
221	252	247	137	26	0,90	10,10	30	75	40	Polypamuk	Pamuk	CTP	2/1 herringbone	15
222	252	247	137	24	0,90	10,10	30	150	40	Polypamuk	Pamuk	CTP	2/1 herringbone	15
223	255	265	131	24	0,90	15,50	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	13
224	255	266	131	24	0,87	15,60	30	150	54	Pamuk	Pamuk	Sulzer	1/1 bez	12
225	281	254	135	24	0,87	11,90	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	14
226	281	254	135	24	0,87	11,90	30	75	54	Pamuk	Pamuk	Sulzer	1/1 bez	14
227	269	266	131	26	0,87	15,70	30	75	40	Pamuk	Pamuk	CTP	1/1 bez	12
228	269	266	131	26	0,87	15,90	30	30	54	Pamuk	Polyester	Somet Alpha	2/2 dimi	13
229	120	265	131	26	0,88	15,50	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	13
230	120	265	131	26	0,88	15,50	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	13
231	138	268	130	24	0,88	16,70	30	30	54	Pamuk	Pamuk	CTP	1/1 bez	12
232	138	269	130	24	0,89	16,70	30	75	54	Pamuk	Pamuk	CTP	1/1 bez	12
233	110	267	131	30	0,86	16,20	30	30	54	Pamuk	Polyester	CTP	2/1 herringbone	12
234	110	267	131	24	0,86	16,30	30	30	40	Pamuk	Pamuk	CTP	2/2 dimi	12
235	206	269	130	24	0,86	16,90	30	75	54	Pamuk	Pamuk	CTP	1/1 bez	12
236	206	267	130	26	0,90	17,00	30	30	40	Pamuk	Pamuk	CTP	2/1 herringbone	12
237	196	269	134	26	0,90	12,30	30	75	54	Pamuk	Pamuk	CTP	2'li 2/2 oxford	14
238	196	256	134	24	0,90	12,40	30	30	40	Pamuk	Polyester	CTP	3/1 twill	14
239	212	266	131	24	0,90	15,90	30	75	54	Pamuk	Polyester	Somet Excel	2/1 herringbone	13
240	444	103	152	24	0,90	90,20	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	22
241	444	103	152	30	0,90	91,10	30	150	40	Pamuk	Pamuk	CTP	2/1 herringbone	22
242	448	100	156	24	0,90	75,60	30	75	54	Pamuk	Polyester	Somet Excel	2/2 dimi	23
243	448	247	156	30	0,90	76,20	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	23
244	194	100	161	24	0,87	62,20	30	75	40	Pamuk	Polyester	CTP	2/2 dimi	24
245	194	256	161	26	0,87	62,30	30	75	54	Pamuk	Polyester	Somet Excel	1/1 bez	24
246	224	110	145	26	0,87	12,80	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	19
247	224	110	145	24	0,87	12,90	30	30	40	Pamuk	Pamuk	CTP	2/1 herringbone	19
248	351	257	147	24	0,87	11,70	30	75	54	Pamuk	Pamuk	Somet Excel	1/1 bez	20
249	358	259	151	24	0,87	96,00	30	30	54	Pamuk	Pamuk	CTP	2/1 herringbone	21
250	358	257	151	26	0,87	97,20	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	21
251	296	257	160	26	0,87	66,00	30	75	54	Pamuk	Polyester	CTP	1/1 bez	24
252	296	259	159	26	0,87	66,60	30	30	54	Pamuk	Polyester	CTP	1/1 bez	24

253	332	154	166	24	0,87	52,40	30	75	54	Pamuk	Pamuk	CTP	2'li 1/1 bez	26
254	332	154	166	24	0,89	52,60	30	75	40	Pamuk	Pamuk	CTP	armür	26
255	332	105	172	26	0,89	42,30	30	30	54	Pamuk	Pamuk	Somet Alpha	2/1 dimi	28
256	278	105	174	24	0,89	40,20	30	150	54	Pamuk	Pamuk	CTP	armür	29
257	278	257	174	24	0,87	40,20	30	75	40	Pamuk	Pamuk	Sulzer	2/1 dimi	29
258	137	257	134	26	0,87	12,60	30	75	54	Pamuk	Polyester	Sulzer	2/1 herringbone	13
259	157	249	137	26	0,87	10,60	30	75	54	Pamuk	Pamuk	CTP	2/2 dimi	15
260	268	259	133	24	0,87	13,30	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	13
261	268	259	133	24	0,87	13,40	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	13
262	307	247	137	30	0,85	10,00	30	75	40	Pamuk	Pamuk	CTP	2'li 1/1 bez	15
263	307	247	137	26	0,85	10,00	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	15
264	233	266	131	26	0,83	15,80	30	75	40	Pamuk	Pamuk	CTP	2/1 dimi	13
265	198	249	137	26	0,83	10,60	30	150	40	Pamuk	Pamuk	CTP	2/1 herringbone	15
266	198	249	137	26	0,83	10,60	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	15
267	136	276	128	30	0,83	20,10	30	75	54	Pamuk	Pamuk	Sulzer	armür	12
268	122	266	131	24	0,83	15,80	30	75	40	Pamuk	Pamuk	Sulzer	2/1 herringbone	13
269	227	249	151	24	0,83	99,90	30	75	40	Pamuk	Pamuk	Sulzer	2/1 herringbone	21
270	227	105	150	24	0,83	10,10	30	75	54	Pamuk	Pamuk	Sulzer	2/1 herringbone	21
271	227	266	157	24	0,88	72,80	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	23
272	299	247	155	24	0,88	79,90	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	23
273	299	107	155	24	0,88	80,20	30	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	23
274	468	249	164	24	0,87	57,20	30	150	54	Pamuk Polyester	Polyester	CTP	2/1 dimi	26
275	468	154	163	26	0,87	57,90	30	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	25
276	458	154	171	24	0,86	43,80	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	28
277	458	154	171	26	0,86	44,00	30	75	54	Polypamuk	Pamuk	Somet Excel	2/1 herringbone	28
278	349	166	129	26	0,86	36,10	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
279	349	167	128	24	0,86	37,30	30	75	54	Pamuk Polyester	Pamuk	Somet Excel	2'li 1/1 bez	9
280	470	249	137	30	0,86	20,10	30	75	54	Polypamuk	Polyester	Somet Excel	2/1 herringbone	9
281	470	154	137	30	0,86	20,70	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	9
282	270	164	130	24	0,87	32,80	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
283	270	165	130	24	0,87	33,20	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
284	290	150	140	24	0,87	17,50	30	75	54	Polyester	Polyester	CTP	2/1 herringbone	9
285	290	150	139	30	0,88	17,70	30	75	54	Pamuk	Polyester	Somet Excel	2/1 dimi	9

286	232	164	130	24	0,88	32,30	40	75	54	Polyester	Pamuk	Somet Excel	2/1 herringbone	9
287	232	164	130	24	0,88	32,60	30	75	54	Pamuk Polyester	Pamuk	CTP	2/1 dimi	9
288	136	159	134	24	0,88	25,30	20	75	54	Pamuk	Pamuk	CTP	1/1 bez	9
289	136	159	134	30	0,85	25,50	30	75	54	Polyester	Pamuk	CTP	2/1 herringbone	9
290	121	165	132	26	0,88	30,10	20	75	54	Polypamuk	Polyester	Somet Excel	2/1 herringbone	9
291	351	165	130	26	0,88	33,80	30	75	54	Pamuk Polyester	Polyester	CTP	2/1 herringbone	9
292	349	166	129	30	0,86	36,10	30	75	54	Pamuk	Polyester	CTP	2/1 herringbone	9
293	349	167	128	24	0,84	37,30	20	150	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
294	470	154	137	24	0,84	20,10	20	75	54	Pamuk Polyester	Pamuk	Somet Excel	2/1 herringbone	9
295	470	154	137	30	0,84	20,70	30	75	54	Polypamuk	Pamuk	CTP	2'li 2/2 dimi	9
296	270	164	130	24	0,84	32,80	20	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
297	270	165	130	24	0,84	33,20	20	75	54	Pamuk Polyester	Pamuk	CTP	2/1 herringbone	9
298	290	150	140	26	0,86	17,50	40	75	54	Pamuk	Pamuk	Somet Excel	2/1 herringbone	9
299	290	150	139	24	0,87	17,70	30	150	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9
300	232	164	130	26	0,87	32,30	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	9
301	296	257	160	26	0,87	66,00	30	75	54	Pamuk	Polyester	CTP	1/1 bez	24
302	296	259	159	26	0,87	66,60	30	30	54	Pamuk	Polyester	CTP	1/1 bez	24
303	332	154	166	24	0,87	52,40	30	75	54	Pamuk	Pamuk	CTP	2'li 1/1 bez	26
304	332	154	166	24	0,89	52,60	30	75	40	Pamuk	Pamuk	CTP	armür	26
305	332	105	172	26	0,89	42,30	30	30	54	Pamuk	Pamuk	Somet Alpha	2/1 dimi	28
306	278	105	174	24	0,89	40,20	30	75	54	Pamuk	Pamuk	CTP	armür	29
307	278	257	174	24	0,87	40,20	30	75	40	Pamuk	Pamuk	Sulzer	2/1 dimi	29
308	137	257	134	26	0,87	12,60	30	75	54	Pamuk	Polyester	Sulzer	2/1 herringbone	13
309	157	249	137	26	0,87	10,60	30	75	54	Pamuk	Pamuk	CTP	2/2 dimi	15
310	268	259	133	24	0,87	13,30	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	13
311	268	259	133	24	0,87	13,40	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	13
312	307	247	137	30	0,85	10,00	30	75	40	Pamuk	Pamuk	CTP	2'li 1/1 bez	15
313	307	247	137	26	0,85	10,00	30	75	40	Pamuk	Polyester	CTP	2/1 herringbone	15
314	233	266	131	26	0,83	15,80	30	75	40	Pamuk	Pamuk	CTP	2/1 dimi	13
315	198	249	137	26	0,83	10,60	30	75	40	Pamuk	Pamuk	CTP	2/1 herringbone	15
316	198	249	137	26	0,83	10,60	30	75	54	Pamuk	Pamuk	CTP	2/1 herringbone	15
317	136	276	128	30	0,83	20,10	30	75	54	Pamuk	Pamuk	Sulzer	armür	12
318	122	266	131	24	0,83	15,80	30	75	40	Pamuk	Pamuk	Sulzer	2/1 herringbone	13

<b>319</b>	470	154	137	30	0,84	20,70	30	75	54	Polypamuk	Pamuk	CTP	2'li 2/2 dimi	9
<b>320</b>	270	164	130	24	0,84	32,80	20	75	54	Polypamuk	Pamuk	CTP	2/1 herringbone	9



## EK-2. Çok Katmanlı Algılayıcı Modelde Kullanılan Kodlar

```
clear all
clc
data=xlsread('veri.xls',1,'a2:n321');
G=data(:,1:13);
C=data(:,14);
I=randperm(size(data,1));
testsayi=fix(0.2*size(data,1));
valsayi=fix(0.2*size(data,1));
egitimsayi=size(data,1)-testsayi-valsayi;
testI=I(1:testsayi);
valI=I(testsayi+1:testsayi+valsayi);
egitI=I(testsayi+valsayi+1:end);
save data G C testI valI egitI I

function [yhat enet]=mlpdene(G,C,egitI,valI,testI,gizli)

net = newfit(G',C',gizli);
net.divideFcn='divideind';
net.divideparam.trainInd=egitI;
net.divideparam.valInd=valI;
net.divideparam.testInd=testI;
[enet]=train(net,G',C');
yhat=sim(enet,G')';
clear all
clc
load data
df=size(testI,2)
for i=1:50
    i
    clear r mse rmse mea enet;
    for j=1:10
        [yhat enet{j}]=mlpdene(G,C,egitI,valI,testI,i);
        r(j)=corr(C(testI),yhat(testI));
        mse(j)=sum((C(testI)-yhat(testI)).^2)/df;
        rmse(j)=sqrt(mse(j));
        mae(j)= sum(abs(C(testI)-yhat(testI)))/df;
    end
    [a b]=min(mse);
    sonuc(i,:)=[r(b) mse(b) rmse(b) mae(b)];
    NET{i}=enet{b};
end
end
```

### EK-3. Radyal Tabanlı Fonksiyon Ağ Modelinde Kullanılan Kodlar

```
function [yhat net]=rbfdene(G,C,egitI,valI,testI,gizli,sp)

x=[G(egitI,:);G(valI,:)];
y=[C(egitI);C(valI)];
net = newrb(x',y',0,sp,gizli);
yhat=sim(net,G')';
clear all
clc
load data
df=size(testI,2)
sp=[1 10 100 1000 10000 100000 1000000 10000000];
sonuc=[];NET=[];
for i=1:8
    for j=50:50:250
        [yhat
enet{j/50}]=rbfdene(G,C,egitI,valI,testI,j,sp(i));
        r(j/50)=corr(C(testI),yhat(testI));
        mse(j/50)=sum((C(testI)-yhat(testI)).^2)/df;
        rmse(j/50)=sqrt(mse(j/50));
        mae(j/50)= sum(abs(C(testI)-yhat(testI)))/df;
        b=j/50;
        sonuc=[sonuc;r(b) mse(b) rmse(b) mae(b)];
        NET=[NET;enet{b}];
    end
end
```

## ÖZGEÇMİŞ

<b>Adı, Soyadı</b>	Vesile Sinem ARIKAN KARGI		
<b>Doğum Yeri ve Yılı</b>	Diyarbakır – 26.01.1981		
<b>Bildiği Yabancı Diller</b>	İngilizce, Almanca		
<b>ve Düzeyi</b>	İyi düzey, az		
<b>Eğitim Durumu</b>	<b>Başlama - Bitirme Yılı</b>		<b>Kurum Adı</b>
<b>Lise</b>	1992	1999	Bursa Anadolu Lisesi
<b>Lisans</b>	2000	2004	Uludağ Üniversitesi Ekonometri Bölümü
<b>Yüksek Lisans</b>	2006	2008	Uludağ Üniversitesi Ekonometri Bölümü Yöneylem Bilim Dalı
<b>Doktora</b>	2008	2013	Uludağ Üniversitesi Ekonometri Bölümü Yöneylem Bilim Dalı
<b>Çalıştığı Kurum (lar)</b>	<b>Başlama - Ayrılma Yılı</b>		<b>Çalışılan Kurumun Adı</b>
<b>1.</b>	2004	2005	Bemsa A.Ş. Kalite Bölümü Proses Sorumlusu
<b>2.</b>	2006	Devam ediyor	Uludağ Üniversitesi Yöneylem Bilim Dalı Araştırma Görevlisi
<b>Yayımlar:</b>	<b>MAKALELER</b> <b>1.</b> Vesile Sinem Arıkan, Mustafa Umut Öztürk, Ahmet Öztürk, Gülay Kasap, "Öncelikli Hedef Programlamanın Bir Tekstil İşletmesine Uygulanması", Review of Social, Economic and Business Studies, 2008. <b>2.</b> Ahmet Öztürk, Vesile Sinem Arıkan, Mustafa Umut Öztürk, "Süreç İyileştirme Yöntemleri ve Yöneylem Araştırması", Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi, Erzurum, 2011. <b>3.</b> Ahmet Öztürk, Şenol Erdoğan, Vesile Sinem Arıkan, "Analitik Hiyerarşi Yöntemi Kullanılarak Tedarikçilerin Değerlendirilmesi: Bir Tekstil Firmasında Uygulama", Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, İzmir, 2011.		

	<p><b>4.</b>Vesile Sinem Arıkan Kargı, Ahmet Öztürk,"Subcontractor Selection Using Analytic Hierarchy Process", Business and Economics Research Journal, Vol.3, Number3, 2012.</p> <p><b>KONGRE FAALİYETLERİ</b></p> <p><b>1.</b>Ahmet Öztürk, Şenol Erdoğan, Vesile Sinem Arıkan, Analitik Hiyerarşi Süreci (AHS) Kullanılarak Tedarikçilerin Değerlendirilmesi: Bir Tekstil Firmasında Uygulama, IX. Ulusal Ekonometri ve İstatistik Sempozyumu, 2008.</p> <p><b>2.</b>Ahmet Öztürk, Vesile Sinem Arıkan, "Lean Production For Profitable Growth And Its Application At Yeşim Textile Company", Zmiany W Strategiach Zarzadzania Organizacjami, Bialystok, 2009.</p> <p><b>3.</b>Ahmet Öztürk, Vesile Sinem Arıkan, Mustafa Umut Öztürk, Süreç İyileştirme Yöntemleri ve Yöneylem Araştırması, X. Ulusal Ekonometri ve İstatistik Sempozyumu 2009.</p> <p><b>4.</b>Vesile Sinem Arıkan Kargı, Ahmet Öztürk, Analitik Hiyerarşi Süreci ile Fasoncu Seçimi , XII. Uluslararası Ekonometri, Yöneylem Araştırması ve İstatistik Sempozyumu, Denizli, 2011.</p>
<b>Diğer:</b>	
<b>İletişim (e-posta):</b>	vesa@uludag.edu.tr
	<p><b>Tarih</b> 10.05.2013</p> <p><b>İmza</b></p> <p><b>Adı Soyadı</b> Vesile Sinem ARIKAN KARGI</p>