

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**



**YAPAY SİNİR AĞLARI İLE
PORTFÖY OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

MEHMET YAVUZ

BALIKESİR, HAZİRAN - 2012

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**



**YAPAY SİNİR AĞLARI İLE
PORTFÖY OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

MEHMET YAVUZ

BALIKESİR, HAZİRAN - 2012

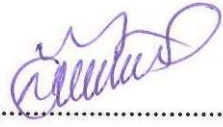
KABUL VE ONAY SAYFASI

Mehmet YAVUZ tarafından hazırlanan “**YAPAY SİNİR AĞLARI İLE PORTFÖY OPTİMİZASYONU**” adlı tez çalışmasının savunma sınavı 21.06.2012 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman
Yrd. Doç. Dr. Necati ÖZDEMİR



.....

Üye
Doç. Dr. Şakir SAKARYA



.....

Üye
Yrd. Doç. Dr. Fırat EVİRGEN



.....

Jüri üyeleri tarafından kabul edilmiş olan bu tez BAÜ Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Hilmi NAMLI

.....

ÖZET

**YAPAY SİNİR AĞLARI İLE PORTFÖY OPTİMİZASYONU
YÜKSEK LİSANS TEZİ
MEHMET YAVUZ
BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**

(TEZ DANIŞMANI: YRD. DOÇ. DR. NECATİ ÖZDEMİR)

BALIKESİR, HAZİRAN - 2012

Yapay zeka tekniklerinden biri olan yapay sinir ağları (YSA), finansal piyasalarda son yıllarda öngörü (tahmin) ve optimizasyon problemlerinde sıkça kullanılmaktadır. YSA, özellikle doğrusal olmayan sistemlerde, öngörüselsel açıdan istatistiksel tekniklere göre daha çok kolaylık sağlayan özelliklere sahiptir.

Bu tezde, İMKB-Ulusal Sınai Endeksinde yer alan 140 hisse senedinin 2010 yılına ait aylık ortalama getirileri kullanılarak risk-getiri tahmini ve portföy optimizasyonu amaçlanmıştır. Bu amaç için, belirtilen hisse senetleri ile aktif büyüklük, piyasa değeri, işlem hacmi ve özsermaye niceliklerine göre eşit ağırlıklı portföyler oluşturulmuş ve bu portföylerin risk-getirileri hesaplanmıştır. Bu değerler kullanılarak yapay sinir ağı eğitilmiş ve eğitilen bu ağı ile de test işlemi gerçekleştirilmiştir.

Test sonucunda getiri ve risk bazında en iyi sonuç özsermayeye göre oluşturulan portföylerde elde edilmiştir. Ayrıca YSA ile getiri tahmininin %1'in altında hata oranı ile gerçekleştiği, risk tahmininde ise hata miktarının binde 5'in altında olduğu gözlenmiştir.

Bununla beraber aktif büyüklüğü, piyasa değeri ve özsermayesi en yüksek olan hisse senetleriyle oluşturulan portföylerin getirileri diğer portföylere göre daha yüksek olmamasına rağmen risk seviyeleri diğer portföy risklerine nazaran minimum seviyededir. Fakat işlem hacmi en yüksek olan hisse senetleriyle oluşturulan portföyün getiri ve riskinin maksimum düzeyde olduğu gözlenmiştir.

Uygulamanın optimizasyon kısmında, bahsi geçen 140 şirketin risk ve getirileri kullanılarak eşit ağırlıklı 50 tane portföy oluşturulmuştur. Maksimum getiriye sahip portföyün getirisi olan %7.5916 değeri için YSA 0.0567 hata oranı ile %7.1590 değerini bulmuştur.

Ayrıca oluşturulan 50 portföy arasında minimum riske sahip olan portföyün riski (standart sapması) ise 0.0019 dur. Bu değer YSA'da 0.0005 hata farkıyla 0.0024 olarak tahmin edilmiştir.

ANAHTAR KELİMELER: yapay sinir ağları (YSA), risk-getiri tahmini, portföy optimizasyonu

ABSTRACT

**PORTFOLIO OPTIMIZATION WITH ARTIFICIAL NEURAL
NETWORKS
MSc THESIS
MEHMET YAVUZ
BALIKESIR UNIVERSITY INSTITUTE OF SCIENCE
MATHEMATICS**

(SUPERVISOR: ASSIST. PROF. DR. NECATİ ÖZDEMİR)

BALIKESİR, JUNE 2012

Artificial neural networks (ANN), which is one of the artificial intelligence techniques, is frequently used in financial markets forecasting (estimation) and optimization problems in recent years. Especially in non-linear systems, terms of predictive, ANN has properties which provide more convenience than statistical techniques.

In this thesis, using the monthly average returns for the year 2010 of 140 stocks contained in ISE-National Industrial Index, risk-return forecasting and portfolio optimization were aimed. For this purpose, using these stocks, equal-weighted portfolios were formed according to companies' active sizes, market capitalizations, trading volumes and equities. Meanwhile risks and returns of these portfolios were calculated. An artificial neural network was trained using the founded values and testing process was realized with this network was trained.

According to test results, the best results on the basis of return and risk were obtained in portfolios which generated from equity. In addition, the error rate of ANN's return prediction was realized approximately 1 percent, the amount of error of risk estimate was observed as less than 5 per thousand.

Besides, although returns of portfolios, which are generated from stocks whose active sizes, market values and equities are the highest, are not higher level than the other portfolios, their risk levels are minimum level compared to other portfolios' risk levels. However, it was observed that return and risk of portfolios, which are generated from stocks whose trading volume is the highest, are maximum level.

In optimization part of the application, using the return and risk of mentioned 140 companies were created equal-weighted 50 portfolios. ANN found that the return of portfolio which has the maximum return (7.5916%) as 7.1590% with rate of error 0.0567.

In addition, the risk (standart deviation) of the portfolio that has minimum risk in mentioned 50 portfolios is 0.0019. This value was estimated to be 0.0024 with 0.0005 amount of error.

KEYWORDS: artificial neural networks (ANN), risk-return prediction, portfolio optimization

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	viii
KISALTMALAR LİSTESİ	ix
ÖNSÖZ	x
1. GİRİŞ	1
2. YAPAY SİNİR AĞLARI	3
2.1 Yapay Sinir Ağı Nedir?	3
2.2 Yapay Sinir Ağlarının Genel Özellikleri	5
2.3 Yapay Sinir Ağlarının Tarihsel Gelişimi.....	8
2.3.1 1970 Öncesi Çalışmalar	9
2.3.2 1970 Sonrası Çalışmalar	10
2.4 Yapay Sinir Ağlarının Uygulama Alanları.....	11
2.4.1 Yapay Sinir Ağlarının Finans Alanındaki Uygulamaları	12
2.5 Yapay Sinir Ağlarının Avantaj ve Dezavantajları.....	14
2.5.1 Avantajları	14
2.5.2 Dezavantajları	14
3. YAPAY SİNİR AĞLARININ TEMEL BİLEŞENLERİ VE YAPISI...	16
3.1 Biyolojik Sinir Ağları	16
3.2 Yapay Sinir Hücreleri.....	18
3.2.1 Girdiler.....	22
3.2.2 Ağırlıklar.....	22
3.2.3 Toplama Fonksiyonu	23
3.2.4 Aktivasyon (Transfer) Fonksiyonu.....	24
3.2.4.1 Doğrusal (Lineer) Fonksiyon	26
3.2.4.2 Parçalı Doğrusal Fonksiyon	26
3.2.4.3 Basamak (Step) Fonksiyon	27
3.2.4.3.1 Tek Kutuplu (Unipolar) Basamak Fonksiyonu.....	27
3.2.4.3.2 Çift Kutuplu (Bipolar) Basamak Fonksiyonu.....	28
3.2.4.4 Sigmoid Fonksiyon	29
3.2.4.5 Tanjant Hiperbolik Fonksiyon	30
3.2.4.6 Sinüs Tipi Fonksiyon	30
3.2.5 Çıktı	31
3.2.6 Ölçekleme ve Sınırlama.....	31
3.3 Yapay Sinir Ağlarının Yapısı.....	32
3.3.1 Katmanlar.....	33
3.3.1.1 Girdi Katmanı (Input Layer)	33
3.3.1.2 Gizli Katman (Hidden Layer)	33
3.3.1.3 Çıktı Katmanı (Output Layer).....	34
3.3.2 Yapay Sinir Ağlarının Çalışma Prensipleri.....	34
3.3.3 İletim ve Bağlantı Çeşitleri	35
4. YAPAY SİNİR AĞLARININ SINIFLANDIRILMASI VE TEMEL ÖĞRENME KURALLARI	37

4.1	Yapay Sinir Ağlarının Sınıflandırılması.....	37
4.1.1	Topolojilerine Göre Sınıflandırılması.....	37
4.1.1.1	İleri Beslemeli (Feed-Forward) Yapay Sinir Ağları	38
4.1.1.2	Geri Beslemeli (Feed-Back) Yapay Sinir Ağları	39
4.1.2	Öğrenme Metotlarına Göre Sınıflandırılması	40
4.1.2.1	Danışmanlı (Supervised) Öğrenme	41
4.1.2.2	Danışmansız (Unsupervised) Öğrenme	42
4.1.2.3	Takviyeli (Reinforcement) Öğrenme	43
4.1.2.4	Karma Öğrenme	43
4.1.2.4.1	Çevrimiçi (On-line) Öğrenme.....	44
4.1.2.4.2	Çevrimdışı (Off-line) Öğrenme	44
4.1.3	Kullanım Amaçlarına Göre Sınıflandırılması.....	44
4.1.3.1	Tahmin-Öngörü Yapma	46
4.1.3.2	Fonksiyon Yaklaştırma	46
4.1.3.3	Desen (Örüntü) Sınıflandırma.....	47
4.1.3.4	Veri İlişkilendirme	48
4.1.3.5	Veri Kavramlaştırma / Kümeleme	49
4.1.3.6	Veri Filtreleme	50
4.1.3.7	Optimizasyon	50
4.1.3.8	Kontrol	51
4.2	Temel Öğrenme Kuralları	51
4.2.1	Hebb Kuralı.....	52
4.2.2	Hopfield Kuralı	52
4.2.3	Delta Kuralı.....	53
4.2.4	Eğimli İniş Kuralı	53
4.2.5	Kohonen Kuralı.....	57
4.3	Yapay Sinir Ağı Modelleri	58
4.3.1	Tek Katmanlı Yapay Sinir Ağları.....	58
4.3.1.1	Perceptron Modeli.....	60
4.3.1.2	ADALINE / MADALINE Modeli.....	61
4.3.2	Çok Katmanlı Algılayıcılar (Multi Layer Perceptron - MLP)....	63
4.3.2.1	ÇKA Ağı'nın Öğrenme Kuralı	64
4.3.3	LVQ (Öğrenme Vektör Nicelendirmesi) Ağı	72
4.3.4	Kohonen Ağı.....	75
4.3.5	ART (Adaptive Resonance Theory) Ağları	77
4.3.5.1	ART-1 Ağı	78
4.3.6	Hopfield Ağı	79
4.3.7	Jordan Ağı.....	80
4.3.8	Elman Ağı	81
5.	PORTFÖY OPTİMİZASYONU	83
5.1	Portföy Kuramı	84
5.2	Portföy İle İlgili Temel Kavramlar.....	84
5.2.1	Dönemlik Getiri	84
5.2.2	Beklenen Getiri	86
5.2.3	Sapma Ölçütleri	88
5.2.3.1	Ortalama Mutlak Sapma	88
5.2.3.2	Varyans ve Standart Sapma (Risk)	88
5.2.3.2.1	Risk Kaynakları	90
5.2.4	Kovaryans	93
5.2.5	Korelasyon Katsayısı	93

5.3	Portföy Teorileri	96
5.3.1	Geleneksel Portföy Teorisi	96
5.3.2	Modern Portföy Teorisi	97
5.3.2.1	Markowitz (Ortalama-Varyans) Modeli	98
5.4	Portföyün (Varlık Kombinasyonlarının) Getirisi ve Riski	100
5.4.1	Risk ve Beklenen Getiri İlişkisi	101
5.4.2	İki Varlıktan Oluşan Portföyün Beklenen Getirisi ve Riski	104
5.4.3	N Varlıktan Oluşan Portföyün Beklenen Getirisi ve Riski	108
5.5	Optimum Portföy	113
5.5.1	Etkin Sınır (Efficient Frontier)	114
6.	UYGULAMA: BİR YAPAY SİNİR AĞI MODELİ İLE RİSK-GETİRİ TAHMİNİ VE OPTİMUM PORTFÖYÜN ELDE EDİLMESİ	116
6.1	Risk-Getiri Tahmini	116
6.1.1	Uygulamanın Amacı	116
6.1.2	Veri Analizi	119
6.1.3	Risk-Getiri Tahmini İçin YSA Modelinin Geliştirilmesi	119
6.1.4	Test Sonuçları ve Tartışma	131
6.2	Optimum Portföyün Belirlenmesi	133
6.2.1	Uygulamanın Amacı	135
6.2.2	Portföy Optimizasyonu İçin YSA Modelinin Geliştirilmesi	136
6.2.2.1	Getiri Maksimizasyonu	137
6.2.2.1.1	Test Sonuçları ve Tartışma	139
6.2.2.2	Risk Minimizasyonu	139
6.2.2.2.1	Test Sonuçları ve Tartışma	141
7.	SONUÇ VE ÖNERİLER	143
8.	KAYNAKLAR	145
9.	EKLER	155
EK – A:	XOR Probleminin Çözümü	155
EK – B:	Çalışmada Kullanılan Hisse Senetlerinin Yıllık (2010 Yılına Ait) Ortalama Getiri ve Riskleri	162
EK – C:	Çalışmada Kullanılan Bir Portföyü Oluşturan Hisse Senetlerinin Aylık Ortalama Getirileri ve Varyans-Kovaryans Matrisi	164

ŞEKİL LİSTESİ

Sayfa

Şekil 3.1: Biyolojik sinir hücresinin yapısı (Freeman and Skapura, 1991, s. 10).....	17
Şekil 3.2: Biyolojik ve yapay sinir ağı	19
Şekil 3.3: Yapay sinir hücresinin yapısı (Tarassenko, 1998, s.7).....	20
Şekil 3.4: Dendrit gösterimi.....	21
Şekil 3.5: İşaret akış grafi gösterimi.....	21
Şekil 3.6: Blok diyagram gösterimi.....	21
Şekil 3.7: YSA yapısının detaylı gösterimi	22
Şekil 3.8: Doğrusal (lineer) fonksiyon.....	26
Şekil 3.9: Parçalı doğrusal (lineer) fonksiyon	27
Şekil 3.10: Tek kutuplu (unipolar) basamak fonksiyonu.....	28
Şekil 3.11: Çift kutuplu (bipolar) basamak fonksiyonu.....	29
Şekil 3.12: Sigmoid tipi fonksiyon	29
Şekil 3.13: Tanjant hiperbolik fonksiyon	30
Şekil 3.14: Sinüs fonksiyonu	31
Şekil 3.15: Çok katmanlı bir yapay sinir ağı modeli	32
Şekil 3.16: Bir yapay sinir ağının çalışma örneği.....	34
Şekil 4.1: Çok katmanlı ileri beslemeli ağ (Kaynar ve Taştan, 2009, s. 164). .	38
Şekil 4.2: Geri beslemeli ağ yapısı	40
Şekil 4.3: Danışmanlı öğrenme algoritmalarının işleyişi	42
Şekil 4.4: Danışmansız öğrenme algoritması (Elmas, 2003, s. 149).....	43
Şekil 4.5: Tahmin işleminin şematik gösterimi	46
Şekil 4.6: Fonksiyon yaklaştırma işleminin şematik gösterimi	47
Şekil 4.7: Desen sınıflandırma işleminin şematik gösterimi	48
Şekil 4.8: Veri ilişkilendirme işleminin şematik gösterimi	49
Şekil 4.9: Kümeleme işleminin şematik gösterimi	49
Şekil 4.10: Gerçek hayat problemlerine sistematik çözüm yaklaşımı.....	51
Şekil 4.11: Bir fonksiyon ve türevinin geometrik yorumu	54
Şekil 4.12: Hata değişimi ve ağırlık vektörlerinin geometrik yorumu	55
Şekil 4.13: İki girdi ve bir çıktıdan oluşan basit TKA modeli	59
Şekil 4.14: Ağırlıkların ve sınıf ayırıcı olan doğrunun geometrik yorumu	60
Şekil 4.15: Bir basit algılayıcı yapısı.....	61
Şekil 4.16: İki ADALINE ağından oluşan bir MADALINE ağı.....	62
Şekil 4.17: ÇKA ağ yapısı örneği	68
Şekil 4.18: Kohonen ağı (Frohlich, 1997)	76
Şekil 4.19: ART-1 ağının genel yapısı	78
Şekil 4.20: Hopfield ağı (Sağiroğlu vd. 2003, s. 62).....	80
Şekil 4.21: Jordan ağı (Kröse and Smagt, 1996, s. 48).....	81
Şekil 4.22: Elman ağı yapısı (Öztemel, 2012, s. 166)	82
Şekil 5.1: Toplam risk kaynakları.....	91
Şekil 5.2: Menkul kıymet sayısı ile risk arasındaki ilişki (Farrell, 1997, s. 30).....	92
Şekil 5.3: Pozitif tam korelasyon.....	94
Şekil 5.4: Negatif tam korelasyon	95
Şekil 5.5: Sıfır korelasyon	95

Şekil 5.6: Farklı varlık veya portföyler için getiri ve risk olanakları (Elton, Gruber, Brown and Goetzmann, 2009, s. 80).....	99
Şekil 5.7: Risk getiri değişimi (Piyasa Doğrusu).....	102
Şekil 5.8: Yatırımcı farksızlık eğrileri (Özçam, 1997, s. 13)	103
Şekil 5.9: Aralarında tam pozitif korelasyon olan iki menkul kıymetin risk-getiri grafiği	105
Şekil 5.10: Aralarında tam negatif korelasyon olan iki menkul kıymetin risk-getiri grafiği	106
Şekil 5.11: Aralarında sıfır korelasyon olan iki menkul kıymetin risk-getiri grafiği	107
Şekil 5.12: Aralarında çeşitli korelasyon olan iki menkul kıymetin risk-getiri grafiği	108
Şekil 5.13: Etkin sınır ve yatırım fırsat seti	114
Şekil 6.1: Risk ve getiri tahmini için oluşturulan MLP ağının topolojisi (Öztemel, 2012, s. 76; Karaoğlan, 2011, s. 517).....	120
Şekil 6.2: Aktif büyüklüğe göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	123
Şekil 6.3: Piyasa değerine göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	126
Şekil 6.4: İşlem hacmine göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	128
Şekil 6.5: Özsermayeye göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	131
Şekil 6.6: Portföy optimizasyonu için oluşturulan MLP ağının topolojisi	137
Şekil 6.7: Maksimum getiriye göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	138
Şekil 6.8: Minimum riske göre eğitimi tamamlanan YSA modelinin öğrenme grafiği	141

TABLO LİSTESİ

Sayfa

Tablo 3.1: İnsan beyni ile bilgisayar sistemlerinin karşılaştırılması (Crone, 2004, s. 8).....	18
Tablo 3.2: Biyolojik sınır sistemi ile YSA'nın yapısal benzerlikleri.....	19
Tablo 3.3: Toplama fonksiyonları ve özellikleri.....	24
Tablo 4.1: Kullanım amaçlarına göre YSA topolojileri (Anderson and McNeill, 1992, s. 31).....	45
Tablo 4.2: XOR problemi	63
Tablo 5.1: Ekonominin durumuna göre A şirketinin getirisi	87
Tablo 5.2: Bir şirketin varyans ve standart sapmasının hesaplanması.....	89
Tablo 5.3: Ağırlıklandırılmış varyans-kovaryans matrisi	110
Tablo 6.1: Araştırmada kullanılan hisse senetleri	117
Tablo 6.2: Portföylerin alt sektörlerdeki hisse adetleri	118
Tablo 6.3: Aktif büyüklüğe göre portföyler matrisi.....	121
Tablo 6.4: Piyasa değerine göre portföyler matrisi	124
Tablo 6.5: İşlem hacmine göre portföyler matrisi.....	126
Tablo 6.6: Özsermayeye göre portföyler matrisi	129
Tablo 6.7: Risk ve getiri tahminine ilişkin test sonuçları.....	132
Tablo 6.8: 140 hisse senediyle oluşturulmuş eşit ağırlıklı portföyler	133
Tablo 6.9: Getiri maksimizasyonuna ilişkin test sonuçları	139
Tablo 6.10: Risk minimizasyonuna ilişkin test sonuçları	142

KISALTMALAR LİSTESİ

YSA	: Yapay Sinir Ağları
İMKB	: İstanbul Menkul Kıymetler Borsası
GPT	: Geleneksel Portföy Teorisi
MPT	: Modern Portföy Teorisi
DARPA	: Defense Advanced Research Projects Agency
XOR	: Exclusive and Or
SOM	: Self-Organizing Model (Kendi Kendine Öğrenme Haritaları)
MLP	: Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
ÇKA	: Çok Katmanlı Algılayıcı
RBF	: Radial Basis Functions (Radyal Tabanlı Fonksiyonlar)
PNN	: Probabilistic Neural Networks (Probabilistik Ağlar)
ART	: Adaptive Rezonans Theory
GRNN	: General Regression Neural Networks (Genel Rekrasyon Ağları)
ADALINE	: Adaptive Linear Neurons
MADALINE	: Multi Adaptive Linear Neurons
LVQ	: Learning Vector Quantization (Öğrenme Vektör Nicelendirmesi)
GA	: Genetik Algoritmalar
TKA	: Tek Katmanlı Algılayıcı
EV	: Expected Value (Beklenen Değer)

ÖNSÖZ

Bu tez çalışması süresince çok değerli vakitlerini ayırıp; bilgi, tecrübe ve üstün görüşleriyle çalışmama yön veren, aynı zamanda büyük cesaret ve özgüven sağlayarak her zaman motive olmamda büyük rol sahibi olan kıymetli hocam ve danışmanım Sayın Yrd. Doç. Dr. Necati ÖZDEMİR'e;

Tez konusunun belirlenmesinde ve finans altyapısını oluşturmamda her zaman desteğini aldığım, çok yoğun zamanlarında bile bana ayıracak zaman bulabilen değerli hocam Sayın Doç. Dr. Şakir SAKARYA'ya;

Yapay sinir ağları ile ilgili üstün bilgi, tecrübe ve uygulamaları ile çalışmamda etkin rolü olan, çok yoğun zamanlarında dahi yardımını esirgemeyen ve yönlendirmeleri olmadan bu çalışmayı sonlandıramayacağımı düşündüğüm Sayın Yrd. Doç. Dr. Aslan Deniz KARAOĞLAN'a;

Yüksek lisans eğitimim boyunca bilgilerinden ve desteklerinden istifade ettiğim Balıkesir Üniversitesi Matematik Bölümü'nün değerli hocalarına, çalışmayı yapmamda gerekli ortamı sağlayan mesai arkadaşlarıma, isimlerini burada yazamadığım ve uzaktan yakından maddi-manevi desteğini aldığım herkese;

Özellikle, bu günlere gelmemi sağlayan; sevgi, ilgi, destek ve güvenlerini hiç esirgemeyip her zaman yanımda olan canım aileme en içten teşekkürlerimi sunarım...

1. GİRİŞ

Yapay Sinir Ağları (YSA) özellikle son yıllarda büyük gelişme gösteren bir çalışma disiplini. Teknolojinin çok hızlı bir şekilde ilerlediği günümüz dünyasında özellikle yapay zekâ ana başlığı altında yer alan pek çok yöntem ve teknik kendine uygulama alanı bulmuştur. Bu alanlardan en çok kullanılanlarından biri de şüphesiz finansdır. Bu çalışmanın amacı da yapay sinir ağlarını tanıtmak ve yapay sinir ağlarının finans alanında bir uygulamasını göstermektir. Çalışmada, oluşturulan bir yapay sinir ağı modeli ile İMKB Ulusal - Sınai Endeksi'nde işlem gören hisse senetlerinden çeşitli kriterlere göre portföyler oluşturulup risk ve getiri tahmininde bulunulmuştur. Aynı zamanda yatırımcının risk ve getiri tercihinine göre optimum portföy elde edilmeye çalışılmıştır.

Portföy optimizasyonunda temel faktörler risk ve getiridir. Yatırım gelecekte belli bir getiri elde etme beklentisiyle yapılır. Ancak, finansal piyasalarda fiyatlar çok sayıda ölçülebilen ve ölçülemeyen ekonomik, politik ve sosyal değişkenin etkileşimi ile meydana geldiğinden, finansal varlıkların fiyatlarını önceden tahmin etmek mümkün değildir. Dolayısıyla, finansal piyasalarda yatırım kararları önemli ölçüde belirsizlik içermekte ve yapılan yatırımlardan beklenen getiri için kayda değer bir risk söz konusu olmaktadır.

Risk ve getiri faktörleri göz önünde bulundurulduğunda portföy optimizasyonunda yatırımcıların kişisel tercihleri ön plana çıkmaktadır. Bir yatırım herhangi bir yatırımcı için optimum tercih olurken, diğeri için olmayabilir. Modelin kişiselleştirilebilmesi için istatistiksel analizler ve yapay sinir ağları gibi yöntemlerden yararlanılabilir.

Bu tez çalışması giriş bölümü hariç beş ana bölümden oluşmaktadır. Giriş bölümünde kısaca yapay sinir ağları yaklaşımı ve portföy optimizasyonu tanıtılmış, tez çalışmasının genel yapısı ve bölümlerinin içeriği hakkında bilgi verilmiştir.

Çalışmanın ilk bölümünde yapay sinir ağları genel olarak açıklanmış ve bu kapsamda yapay sinir ağlarının tanımı, tarihçesi, kullanım alanları ve özellikle finans alanında uygulanması, avantaj ve dezavantajlarından bahsedilmiştir.

İkinci bölümde, yapay sinir ağlarının çalışma prensibinin daha iyi anlaşılabilmesi için öncelikle biyolojik sinir yapısı açıklanmış ve buna bağlı olarak yapay sinir ağlarının biyolojik sinir yapısına benzerliği anlatılmıştır. Bu bölüm, yapay sinir ağlarının eğitilmesi ve test edilmesi açıklamalarıyla sonlandırılmıştır.

Çalışmanın üçüncü bölümünde yapay sinir ağlarının çeşitli özelliklerine göre sınıflandırılması ve temel öğrenme kuralları ele alınmıştır. Ayrıca bu bölümde literatürde en çok kullanılan ve belli sınıfların genel özelliklerini taşıyan yapay sinir ağı modelleri detaylı bir şekilde anlatılmıştır.

Dördüncü bölümde, finansal varlıkların ve bu varlıkların kombinasyonları olan portföyün getiri ve riskinin hesaplanması, Geleneksel Portföy Teorisi (GPT), Modern Portföy Teorisi (MPT) ve portföy optimizasyonu anlatılmıştır.

Çalışmanın beşinci bölümünde ise İMKB Ulusal Sınai Endeksi'nde yer alan hisse senetleri kullanılarak yapay sinir ağlarıyla risk ve getiri tahmininin yapıldığı, aynı zamanda optimum portföyün elde edildiği uygulama kısmı yer almaktadır.

2. YAPAY SINİR AĞLARI

Bu bölümde yapay sinir ağlarının tanımı, tarihçesi, kullanım alanları, özel olarak finans alanında kullanımı, avantaj ve dezavantajlarından bahsedilecektir.

2.1 Yapay Sinir Ağı Nedir?

Yapay sinir ağları (YSA) en genel çerçevede insan beynini taklit ederek insan gibi yorum yapabilmeye çalışan bir yapay zekâ yöntemidir. Diğer bir ifadeyle YSA normalde bir insanın düşünme, karmaşık problemleri çözebilmek için yorumlama ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmeye çalışır.

Peki *neural network* (sinir ağı) terimi nereden gelmektedir? Kişisel bilgisayarlarda, bir yerel alan *networku* (ağı) işleyen bir bütün halinde birkaç bilgisayarı birbirine bağlar. Beyin içerisinde de nöron adı verilen temel hesaplama birimleri aynı şekilde işleyen bir bütün halinde birbirleriyle bağlanmışlardır. Beyin çalışmalarında filizlenen teknolojiler, işleyen bir bütün halinde birbirleriyle bağlanmış temel nöron (neuron) mefhumunu kullanırlar. Böylelikle *neural network* (sinir ağı) terimi ortaya çıkmıştır (Klimasauskas, 1991, s. 45).

Yapay sinir ağlarının herkes tarafından kabul görmüş genel bir tanımı yoktur. Hatta bazı araştırmacılar YSA için genel bir tanım vermek yerine YSA türlerinin kendi içinde tanımlanması gerektiğini savunmaktadırlar. Bununla birlikte bütünlüğün sağlanması adına literatürde yer alan bazı tanımlar şu şekilde ifade edilebilir:

Yapay sinir ağları isminden de anlaşılacağı gibi beyni oluşturan sinir hücrelerini (nöron) matematiksel olarak taklit ederek akıllı bir sistem oluşturmaya çalışan bir yapay zekâ yöntemidir. YSA insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır. Başka bir deyişle YSA, biyolojik sinir ağlarını taklit eden bilgisayar

programlarıdır. Yapay sinir ağı bir programcının geleneksel yeteneklerini gerektirmeyen, kendi kendine öğrenme düzenekleridir. Bu ağlar öğrenmenin yanı sıra, ezberleme ve bilgiler arasında ilişkiler kurma yeteneğine sahiptirler (Elmas, 2003, s. 23).

Yapay sinir ağı kavramı beyin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri olarak ortaya çıkmış ve ilk çalışmalar beyin oluşturan biyolojik hücrelerin ya da literatürdeki ismi ile nöronların matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır (Efe ve Kaynak, 2000, s. 1).

Sağiroğlu, Beşdok ve Erler (2003)'e göre yapay sinir ağı, birçok basit işlemci elemandan oluşan yapılardır. Bu elemanlar farklı formda ifade edilebilen nümerik verileri taşıyan “bağlantılar” veya “ağırlıklar” ile birbirine bağlıdır.

Darpa Neural Network Study (1989) isimli çalışmada ise “YSA, birbirlerine paralel olarak çalışan birçok basit işlem elemanından oluşan ve fonksiyonu, ağı yapısı, bağlantı ağırlıkları ve elemanlarda gerçekleştirilen işlemler tarafından belirlenen bir sistemdir” tanımı verilmektedir.

Mcnelis (2005), lineer ve polinom yaklaşım metotları gibi yapay sinir ağını “birtakım girdi değişkenler kümesi $\{x_i\}$, $i = 1, \dots, k$ ile bu değişkenlere bağlı olarak bir ya da daha fazla çıktı değişkenler kümesi $\{y_i\}$, $i = 1, \dots, k$ arasındaki ilişki” olarak tanımlamaktadır. Üstelik bir yapay sinir ağı ile diğer yaklaşım metotları arasındaki fark ise sinir ağının, içinde, girdi değişkenlerinin sigmoid ya da lojistik fonksiyon olarak bilinen özel bir fonksiyonla dönüştürülmüş olduğu bir ya da daha fazla gizli katman kullanmasıdır (McNelis, 2005, s. 21).

Daha kapsamlı ve genel kabul gören bir tanım ise Haykin (1999) tarafından verilmiştir. Buna göre; “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. Bu işlemci iki farklı şekilde beyin ile benzerlik göstermektedir:

1. Bilgi, ağ tarafından öğrenme süreciyle çevreden elde edilir.

2. Elde edilen bilgileri biriktirmek için sinaptik ağırlıklar (synaptic weights) olarak da bilinen nöronlar arası bağlantı güçleri kullanılır”.

İnsan beyninin fonksiyonlarının benzetimini yapan ve biyolojik açıdan esinlenmiş bir paradigması olan YSA’lar, nöronların bildiğimiz çalışma prensibine dayanmaktadırlar. Çünkü YSA’lar beyinde bilgi işleme sorumlusu hücresel eleman olarak kabul edilirler. Bir YSA sistemi, birbiriyle etkileşim içindeki pek çok yapay sinir hücresinin paralel bağlı bir hiyerarşik organizasyonudur (Martin and Araque, 2006, s. 22). YSA modeli, sinir hücreleri arasında ilişki kuran, belli bir fonksiyonla bu ilişkilerden sonuçlar üreten ve bu sonuçları örneklerle karşılaştırarak optimize eden bir sistemdir (Fausett, 1994, s. 5). YSA, birbirine paralel olarak faaliyet gösteren ve doğrusal olmayan hesaplamalı elemanlardan oluşan, aynı zamanda biyolojik sinir ağlarını andıran modellerdir (Lippmann, 1987, s. 4).

2.2 Yapay Sinir Ağlarının Genel Özellikleri

Yapay sinir ağları; öğrenme, sınıflandırma, ilişkilendirme, genelleme, tahmin, özellik belirleme, optimizasyon gibi işlemlerin oluşmasını sağlarlar. Ağların kendi kendilerine öğrenebilme yetenekleri vardır. Görmedikleri örnekler hakkında bilgi üretebilirler. Yapay sinir ağları, uygulanan ağ modeline göre değişik karakteristik özellikler göstermelerine karşın birkaç temel ortak özelliğe sahiptirler. Bu özellikler şunlardır:

Doğrusal Olmama: YSA’ların en önemli özelliklerinden birisi gerçek hayattaki doğrusal olmayan problemlerin çözümünde kullanılabilir olmasıdır. Doğrusal olmayan yapıları dikkate alabilme özelliği bu çalışmanın içeriği açısından da önem taşımaktadır. Çünkü yapıları gereği finansal veriler de doğrusal olmayan yapıya sahiptirler. Ancak tahmin zorlukları nedeniyle analizler genellikle lineer yöntemlerle yapılmaktadır. Hâlbuki bu durum, muhtemel bir doğrusal olmayan yapı içerilmesi durumunda yanlış sonuçlara yol açabilmektedir; ya da en azından analizi yapılan sistemde açıklanamayan bileşenler oluşabilmektedir. Sonuç olarak, analiz konusunun içerdiği veri setinin doğrusal veya doğrusal olmayan yapı içeriyor olması, analiz sonuçlarını etkileyecek önemli bir faktördür.

Bu yüzden, doğrusal olmayan yapıları dikkate alabilmesi yapay sinir ağlarının önemli bir özelliğidir (Yurtoğlu, 2005, s. 35).

Paralel Çalışma: Günümüzde kullanılan bilgi işleme yöntemleri genellikle seri ve sıralı işlemlerden oluşmaktadır. Seri işlemcilerde herhangi bir birimin yavaş olması tüm işletim sistemini yavaşlattığından dolayı, YSA'ların sahip olduğu paralellik sayesinde yavaş bir birim, sistemin çalışması sırasında herhangi bir sorun oluşturmamaktadır (Bayır, 2006, s. 8). Ayrıca yapay sinir ağlarında tüm işlem elemanlarının eş zamanlı çalışmalarından dolayı elde edilen çıktılar diğer sistemlere göre daha hızlıdır.

Öğrenme: YSA'ların en önemli özelliklerinden biri de öğrenme sistemlerinin insan beyninin çalışma şekline benzemesidir. Bu özellikleri sayesinde YSA'lar, geleneksel yöntemlere göre çok karmaşık olan problemlere dahi çözüm üretebilmektedir. Bununla beraber YSA'lar, insanların basitçe yapabildiği ve geleneksel yöntemlerin uygulanamadığı problemlere de uygulanabilmektedir. Geleneksel yöntemlerden üstün olan bir diğer özelliği ise, yapay sinir ağlarının sahip olduğu sürekli öğrenmedir. YSA, kendilerine gösterilen yeni örnekleri öğrenebilmeleri ve yeni durumlara adapte olabilmeleri sayesinde sürekli olarak öğrenme potansiyeli içerisindedirler.

Genelleme: YSA'lar kendi kendine öğrenme yeteneği sayesinde eğitim aşamasında gösterilen örnekleri kullanarak daha önce hiç görmediği örneklerde genelleme yapabilme özelliğine sahiptir. Yani, eksik, hatalı ya da kayıp veriler için bile çözüm üretebilmektedir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş yapay sinir ağı modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı sergileyebilir (Vural, 2007, s. 31).

Hata Toleransı ve Esneklik: Yapay sinir ağları, geleneksel işlemcilerden farklı işlem yapmaktadırlar. Geleneksel işlemcilerde tek bir merkezi işlemci eleman her hareketi sırasıyla gerçekleştirmektedir. Seri bilgi-işlem yapan geleneksel bir sistemde herhangi bir birimin hatalı çalışması ya da fonksiyonunu tamamen yitirmiş olması tüm sistemin hatalı çalışmasına, bozulmasına veya yanlış çıktı elde etmesine sebep olacaktır. YSA modelleri, her biri büyük bir problemin

bir parçası ile ilgilenen çok sayıda basit işlemci elemanlardan oluşmaları ve bağlantı ağırlıklarının ayarlanabilmesi gibi özelliklerinden dolayı önemli derecede esnek bir yapıya sahiptirler. Bu esnek yapı sayesinde ağın bir kısmının zarar görmesi modelde sadece performans düşüklüğü oluşturmakta, problemin çözümünde ve çıktı elde edilmesinde büyük bir soruna yol açmamaktadır. Ayrıca bu durumda modelin işlevini tamamen yitirmesi de söz konusu değildir. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir. Ayrıca toplam işlem yükünü paylaşan işlemci elemanların birbirleri arasındaki yoğun bağlantı yapısı sinirsel hesaplamanın temel güç kaynağıdır. Bu yerel işlem yapısı sayesinde, YSA yöntemi en karmaşık problemlere bile uygulanabilmekte ve tatminkâr çözümler elde edilebilmektedir (Yurtoğlu, 2005, s. 35).

Eksik Verilerle Çalışma: Yapay sinir ağları, geleneksel sistemlerin aksine, eğitildikten sonra kendileri eksik bilgilerle çalışabilmekte ve gelen yeni örneklerde eksik bilgi olmasına rağmen çıktı üretebilmektedir. YSA'ların eksik bilgiler ile çalışması, performanslarının düşeceği anlamına gelmemektedir. Performansın düşmesi eksik olan bilginin önemine bağlıdır. Ayrıca hangi bilginin önemli olduğunu ağın kendisi eğitim sırasında öğrenmektedir. Kullanıcıların bu konuda bir fikri yoktur. Ağın performansı düşük olunca, kayıp ya da eksik olan bilginin önemli olduğu, performansta herhangi bir düşüş gözlenmediği takdirde ise eksik bilginin önemli olmadığı sonucuna varılır (Bayır, 2006, s. 10).

Çok Sayıda Değişken ve Parametre Kullanma: Yapay sinir ağları diğer istatistikî ve geleneksel yöntemlerin aksine, herhangi bir ekstra dönüşüme ihtiyaç olmadan çok sayıda değişken ve parametre ile çalışabilmektedir. Bu sayede mükemmel bir tahmin ve öngörü doğruluğu ile genel çözümler sunabilmektedir (Yurtoğlu, 2005, s. 36).

Uyarlanabilirlik: YSA ağırlıkları uygulanan probleme göre değiştirilir. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, işaret işleme, sistem tanımlama ve denetim gibi alanlarda etkin olarak kullanılır (Öztemel, 2012, s. 32). Yapay sinir ağları, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlayabilmektedir. Yani belirli bir problemi çözmek amacıyla

eđitilen bir yapay sinir ađı, problemdeki deđiřikliklere gre tekrar eđitilebilmekte, deđiřiklikler devamlı ise gerek zamanda da eđitime devam edebilmektedir. Bu zelliđi ile yapay sinir ađları, uyarlamalı (adaptive) rnek tanıma, sinyal iřleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılabilir (Sara, 2004, s. 14).

Bu zelliklerin dıřında yine YSA'ların ortak zellikleri arasında řunlar da sayılabilir (ztemel, 2012, s. 31-33):

- YSA makine đrenmesi gerekleřtirir.
- Programları alıřma síteli bilinen programlama yntemlerine benzememektedir.
- Bilgiyi saklarlar.
- rnekleri kullanarak đrenirler.
- Gvenle alıřtırılabilmeleri iin nce eđitilmeleri ve performanslarının test edilmesi gerekmektedir.
- Algılamaya ynelik olaylarda kullanılabilirler.
- řekil iliřkilendirme ve sınıflandırma yapabilirler.
- rnt tamamlama gerekleřtirebilirler.
- Kendi kendine organize etme ve đrenebilme yetenekleri vardır.
- Dereceli bozulma (Graceful degradation) gsterirler.
- Dađıtık belleđe sahiptirler. Veri dađıtılmıř birleřik hafıza yapısı kullanılır ve bilgi farklı formlara dnřtrlerek iřlenebilir.
- Sadece nmerik bilgi ve verilerle alıřabilmektedirler.

2.3 Yapay Sinir Ađlarının Tarihsel Geliřimi

İnsan beyninin biyolojik yapısı ve bu yapının bilgisayar ortamına aktarılması, kullanılması ve uygulanmasıyla ilgili alıřmalar yzlerce yıldır devam etmektedir. Ancak modern tekniđin geliřmesiyle dřnme ve uygulanabilirlik sreci daha da hızlanmaktadır. Yapay sinir ađlarının tarihesi, insanların nrobiyoloji konusuna ilgi duyması ve elde ettikleri bilgileri bilgisayar bilimine uygulamaları ile bařlamıřtır. YSA ile ilgili alıřmaları 1970 ncesi ve sonrası diye ikiye ayırmak gerekmektedir. nk 1970 yılında bu bilimin

tarihinde çok önemli bir dönüm noktası başlamış ve o zamana kadar olmaz denilen, başta XOR problemi olmak üzere birçok sorun çözülmüş ve yeni gelişmeler başlamıştır. Bu dönemde her şey bitti derken YSA yeniden doğmuştur (Öztemel, 2012, s. 37).

2.3.1 1970 Öncesi Çalışmalar

YSA'da 1970 çok kritik bir dönemdir. Bu döneme kadar yapay zekâ bilimi geliştikçe yapay zekâcılar kendilerini her geçen gün daha çok öne çıkartmak amacı ile yapay sinir ağlarının çalışmalarını çok yakından takip ediyor ve eleştirilerde bulunuyorlardı. Yapay sinir ağlarının tarihinde bir duraklama devrine neden olan olay ise yapay zekâ biliminin o devirde önde gelen isimlerinden Minsky ve Papert tarafından yazılan **Perceptron** (Algılayıcılar) isimli kitap oldu. Bu kitapta yazarlar özellikle yapay sinir ağlarına dayalı algılayıcıların bilimsel bir değerinin olmadığını ve doğrusal olmayan (nonlinear) problemlere çözüm üretemediğini iddia ettiler. Bu tezlerini kanıtlamak amacıyla da meşhur XOR problemini örnek gösterdiler. Bu örnek bu alanda çalışma yapan birçok kişiyi tatmin etti ve YSA ile ilgili çalışmalar bıçak gibi kesildi. Bu zamana kadar yapılan çalışmaların bazıları kronolojik olarak aşağıdaki gibi listelenebilir (Öztemel, 2012, s. 38).

1890: İnsan beyninin yapısı ve fonksiyonları ile ilgili ilk yayının yazılması (James, 1890),

1911: İnsan beyninin bileşenlerinin belirli bir düzenek ile sinir hücrelerinden (nöron) oluştuğu fikrinin benimsenmesi,

1943: Yapay sinir hücrelerine dayalı hesaplama teorisinin ortaya atılması ve eşik değerli mantıksal devrelerin (threshold logic device) geliştirilmesi,

1949: Biyolojik olarak mümkün olabilen öğrenme prosedürünün bilgisayarlar tarafından gerçekleştirilebilecek şekilde geliştirilmesi (Hebb, 1949),

1956-1962: ADALINE ve Widrow öğrenme algoritmasının geliştirilmesi, tek katmanlı algılayıcıların (single layer perceptron) geliştirilmesi,

1965: İlk makine öğrenmesi kitabının (Learning Machines) yayınlanması (Nilson, 1965),

1967-1969: Bazı gelişmiş öğrenme algoritmalarının geliştirilmesi,

1969: Tek katmanlı algılayıcıların problemleri çözme yeteneklerinin olmadığını gösterilmesi (Minsky and Papert, 1969),

1969: DARPA'nın yapay sinir ağlarını desteklemeyi durdurup diğer yapay zekâ çalışmalarına destek vermesi.

2.3.2 1970 Sonrası Çalışmalar

YSA ile ilgili çalışmaların 1969 yılında sekteye uğramasına ve gerekli finansal desteklerin kesilmesine rağmen bazı bilim adamları çalışmalarına devam ettiler. Özellikle Amari, Anderson, Cooper, Fukushima, Grossberg, Kohonen ve Hopfield gibi araştırmacıların çalışmaları 1980'li yıllarda meyvelerini vermeye başladı ve yapay sinir ağlarındaki çalışmalar sessizliğini bozdu. 1970 yılından sonra yapılan çalışmaların bazıları kronolojik olarak aşağıdaki gibi listelenebilir (Öztemel, 2012, s. 41):

1969-1972: Doğrusal ilişkilendiricilerin geliştirilmesi (Kohonen, 1972),

1972: Korelasyon matriks belleğinin geliştirilmesi,

1974: Geriye yayılım modelinin geliştirilmesi,

1978: ART modelinin geliştirilmesi,

1982: Kohonen öğrenmesi, SOM modeli, Hopfield ağları ve çok katmanlı algılayıcıların (MLP) geliştirilmesi (Kohonen, 1982),

1984: Boltzmann makinesinin geliştirilmesi,

1985: Çok katmanlı algılayıcıların (Genelleştirilmiş Delta Öğrenme Kuralı ile) geliştirilmesi

1988: RBF modelinin geliştirilmesi (Broomhead and Lowe, 1988),

1991: GRNN ve PNN modellerinin geliştirilmesi (Specht, 1988; Specht, 1991),

1991'den günümüze: Bugüne kadar YSA ile ilgili sayısız çalışma ve uygulamalar geliştirilmiştir.

Günümüzde yapay sinir ağları artık teorik ve laboratuvar çalışmaları olmaktan çıkmış ve günlük hayatta kullanılan sistemler oluşturmaya ve pratik olarak insanlara faydalı olmaya başlamışlardır (Öztemel, 2012, s. 40).

2.4 Yapay Sinir Ağlarının Uygulama Alanları

Doğrusal olmayan, çok değişkenli problem uzayına sahip, gürültülü, değişkenler arasında karmaşık etkileşimleri olan, kesik olmayan, eksik, kusurlu, hata olasılığı yüksek veriler ve problemlerin çözümü için özellikle matematiksel bir model ve algoritmanın bulunmaması gibi durumlarda, yaygın halde başarılı bir YSA uygulaması yapılabilmektedir. Bu amaçla geliştirilmiş ağlar genel olarak şu fonksiyonları yerine getirmektedir (Öztemel, 2012, s. 36):

- Probabilistik fonksiyon kestirimleri yaklaşımları,
- Sınıflandırma,
- Veri ilişkilendirme,
- Desen eşleştirme, tanıma,
- Kavramlaştırma / kümeleme,
- Zaman serileri analizleri,
- Veri sıkıştırma,
- Doğrusal olmayan sistem modelleme,
- Optimizasyon,
- Zeki ve doğrusal olmayan kontrol.

Yapay sinir ağları günlük hayatta karşılaşılan problemlerde oldukça geniş bir uygulama alanı bulmakta ve bu alanlar gittikçe artmaktadır. Bu alanların çok olmasında, şüphesiz YSA'nın eksik bilgilerle çalışabilme ve normal olmayan

verilere çözüm üretebilme yeteneđi büyük rol oynamaktadır. Günümüzde endüstri alanından finansa, otomasyon ve kontrolden tıp uygulamalarına, evimizdeki aletlerden cep telefonlarına kadar birçok alanda uygulanabilmektedir.

Verideki trend veya yapıyı (pattern) en iyi tanımlayan yöntem olmaları dolayısıyla, tahmin (prediction) ve öngörü işlemleri için çok uygundur. YSA'ların gerçek hayattaki yaygın uygulama alanlarına şu örnekler verilebilir (Yurtođlu, 2005, s. 9):

- Kalite – Kontrol
- Finansal Öngörü
- Ekonomik Öngörü
- Kredi Derecelendirme
- Konuşma ve Yapı Tanımlama
- İşlem Modelleme ve Yönetimi
- Laboratuvar Araştırmaları
- İflas Tahmini
- Petrol ve Gaz Arama

Bu uygulama alanları daha da çoğaltılabilir. Çünkü YSA'lar hemen hemen her alanda uygulanabilirliğe sahiptir. Ayrıca bütün bu alanlarda nasıl uygulandığını anlatmak da oldukça güçtür. Bu nedenle YSA'nın çalışmayla ilgili olan uygulamalarına yer verilecektir.

2.4.1 Yapay Sinir Ağlarının Finans Alanındaki Uygulamaları

Finansal piyasalarda bir deđişkenin ilerde alacağı deđeri önceden tespit etmek oldukça önemlidir. Yapay sinir ağları da bu deđerleri öngöründe bulunan ve finans problemleri çözümünde ele alınan yöntemlerin en çok kullanılanlarından ve en popüler olanlarından biridir. Bu problemlere örnek olarak; hisse senetlerinin performansının ölçülmesi ve seçimi, döviz kurlarının yönünün belirlenmesi, bir şirketin iflas edip etmeyeceğinin tahmin edilmesi, finansal krizlerin öngörülmesi, hileli (manipülatif) işlemlerin tespit edilmesi, menkul kıymet alım-satımı ve tahminler yapılması ile gelecekteki fiyatların tahmin edilmesi, endeks tahmini,

gayri menkul deęerleme, kredi danıřmanlıęı, kurumsal tahvil deęerlemesi, kredi kartı kullanım analizi, kurumsal finansal analiz, finansal sıkıntı teřhisi ve portföy optimizasyonu gibi konular sayılabilir (Akel ve Bayramoęlu, 2008, s. 83).

Finansal piyasalarda gnlk fiyat hareketleri dinamik ve dalgalı bir yapıya sahip olduęu için, YSA gibi bilgisayar temelli öęrenme algoritmaları finansal piyasaların yönn tahmin etmede oldukça uygun yöntemlerdir (Oh, Kim ve Kim, 2006, s. 96). YSA'ların ekonomi ve finans alanında kullanıldıęı yerlerin bařında tahmin etme yani geleceęi öngörme gelmektedir. Geniř bir uygulama sahasına sahip olan yapay sinir aęlarının en güçlü özellięi, verilerden yararlanarak bu veriler arasındaki iliřkileri öęrenebilme yeteneęidir. Bu nedenle, yapay sinir aęları kendilerinin dizayn edilmesinde kullanılan girdi verilerine baęımlıdır (Hagan, Demuth and Beale, 1996, s. 6).

Yapay sinir aęları, btn finansal piyasalarda, borsada, bono iřlemlerinde, uluslararası nakit para akıřında ve ticaret mallarında kullanılmaktadır. Birçok kullanıcı bu sistemleri para getiren sistemler (see green) olarak tanımlamakta ve gerçekten de Japonya finans piyasasında yapay sinir aęlarının çok bařarılı olduęu bildirilmektedir. Daiichi Kangyo Bankası'nın devlet tahvili iřlemleri ile ilgili raporuna göre, yapay sinir aęları bu tahvillerin isabet oranını %60'dan %75 deęerine yükseltmiřtir. Ayrıca, Daiwa Arařtırma Enstits, bir yapay sinir aęının Nikkei Endeksinden (Nikkei Average) %20 daha iyi sonuç verdięini belirtmiřtir (Anderson, McNeill, 1992, s. 65).

Btn bu uygulamalara ek olarak, YSA ařaęıdaki finansal alanlarda da uygulanabilmektedir:

- Makro ekonomik tahminler,
- Borsa benzetim çalıřmaları endekslerinin tahmini,
- Kredi kartı hilelerinin tespiti,
- Kredi kartı kurumlarında iflas tahminleri,
- Banka kredilerinin deęerlendirilmesi,
- Emlak kredilerin yönetilmesi,
- Döviz kuru tahminleri,
- Risk analizleri (Vural, 2007, s. 39).

2.5 Yapay Sinir Ağlarının Avantaj ve Dezavantajları

Yapay sinir ağları özellikle doğrusal olmayan (non-linear) sistemlerde geleneksel modellere nispeten bazı önemli avantajlara sahiptir. Bu avantajlara karşılık YSA da elbette kusursuz bir yapıya sahip değildir. Ağ yapısının belirlenmesinden eğitim sürecine, ağ davranışlarının açıklanmasından üretilen sonuçların yorumlanmasına kadar birçok bölümde önemli dezavantajlara sahiptir.

2.5.1 Avantajları

Yapay sinir ağları, hesaplama gücünü, öncelikle yoğun derecede paralel dağılmış yapısından almaktadır. Ayrıca bu gücü öğrenme ve dolayısıyla genelleştirme özelliklerinden de elde etmektedir. YSA'nın genelleştirmesi ise eğitim (öğrenme) süresince hiç görmediği girdiler için bile mantıklı çıktılar üretebilmesidir. Bu iki bilgi-işlem kapasitesi YSA'nın kontrol edilmesi zor olan ve karmaşık (büyük ölçekli) problemleri çözebilmesine olanak sağlar (Haykin, 1999, s. 2). Bu özelliklerinin yanı sıra YSA'nın diğer metotlarda bulunmayan güçlü yanlarını şu şekilde sıralamak mümkündür (Öztemel, 2012, s. 208):

- Kendi kendini organize edip ağ bağlantılarını değiştirerek öğrenme yeteneği,
- Kuralların bilinmediği kirli ve eksik verileri tolere etme yeteneği,
- Paralel olarak verileri işleyebilme yeteneği,
- Belirli örneklerden genelleme yapabilme yeteneği,
- Girdi değişkenleri arasındaki karmaşık ilişkileri bulma yeteneği,
- Çevredeki değişikliklere dinamik olarak uyarlanma yeteneği,
- Verilerin dağılımı ya da faktörler arası ilişkilerin şekli hakkında varsayımlar gerektirmemesi.

2.5.2 Dezavantajları

Yapay sinir ağlarının yukarıda anlatılan önemli avantajları hemen her konuda uygulama alanı bulmalarını sağlamıştır. Her ne kadar avantajları çok olsa

da bunun yanında göz önünde bulundurulması gereken dezavantajları da mevcuttur. Bunlar arasında en önemlisi, geniş veri seti kullanımını gerektirmesidir. Eğitim süresince YSA'nın eğitilebilmesi ve test edilebilmesine yetecek genişlikte veri setine ihtiyaç duyulmaktadır. Bununla beraber yeterli veri seti genişliği için belli bir kıstas bulunmamaktadır, yapılan uygulamaya göre değişmektedir.

Dezavantaj olarak nitelendirilebilecek diğer bir nokta ise basit olarak görülebilecek modelleme yapılarına rağmen bazı konularda uygulamanın zor ve karmaşık olabilmesidir. Bazı durumlarda, herhangi bir yakınsama sağlamak bile imkânsız hale gelebilmektedir, fakat bu durum da uygulama alanına bağlı olup çok karmaşık problem tiplerinde ortaya çıkmaktadır (Yurtoğlu, 2005, s. 37).

Doğru modelleme için genellikle deneme yanılma yapılması da önemli bir dezavantaj olarak görülebilir. Çünkü kullanılan modelin doğruluğunu ve elde edilen çözümün optimum çözüm olup olmadığını test etmek zordur. Model doğru kurulmuş olsa bile YSA, optimum çözüm garantisi vermez. Yalnızca kabul edilebilir çözüm sunar ya da bulunan çözümlerden en yakın olanı istenilen çözüm olarak sayılır. Ağın öğreneceği veri setinin ağa gösterilmesi de başlı başına bir problemdir. Yapay sinir ağları sayısal (nümerik) veri haricinde veri kabul etmemektedir. Sayısal olmayan verilerin de sayısal verilere dönüştürülmesi gerekir ki bu da tamamen kullanıcının ya da uygulayıcının becerisine bağlıdır. Sayısal olmayan verilerin ağa gösterilmesi sorunu hala tam olarak çözülememiştir. Günümüzde birçok olayın yapay sinir ağlarıyla hala çözülememiş olmasının en temel nedenlerinden biri de budur (Öztemel, 2012, s. 34).

Belki de en önemli sorun ağın davranışlarının açıklanamamasıdır. Bir çözüm üretildiği zaman neden ve nasıl bir çözüm üretildiği konusunda bilgi bulmak mümkün değildir (Öztemel, 2012, s. 35). Yani içeride ne olup bittiğini kimse tam olarak bilemez. Bu nedenle yapay sinir ağlarına Kara Kutu (Black Box) ismi verilmiştir. Bazı bilimciler ise Voodoo Mühendisliği (Voodoo Engineering) olarak ifade etmişlerdir (Anderson ve McNeill, 1992, s. 61).

3. YAPAY SINİR AĞLARININ TEMEL BİLEŞENLERİ VE YAPISI

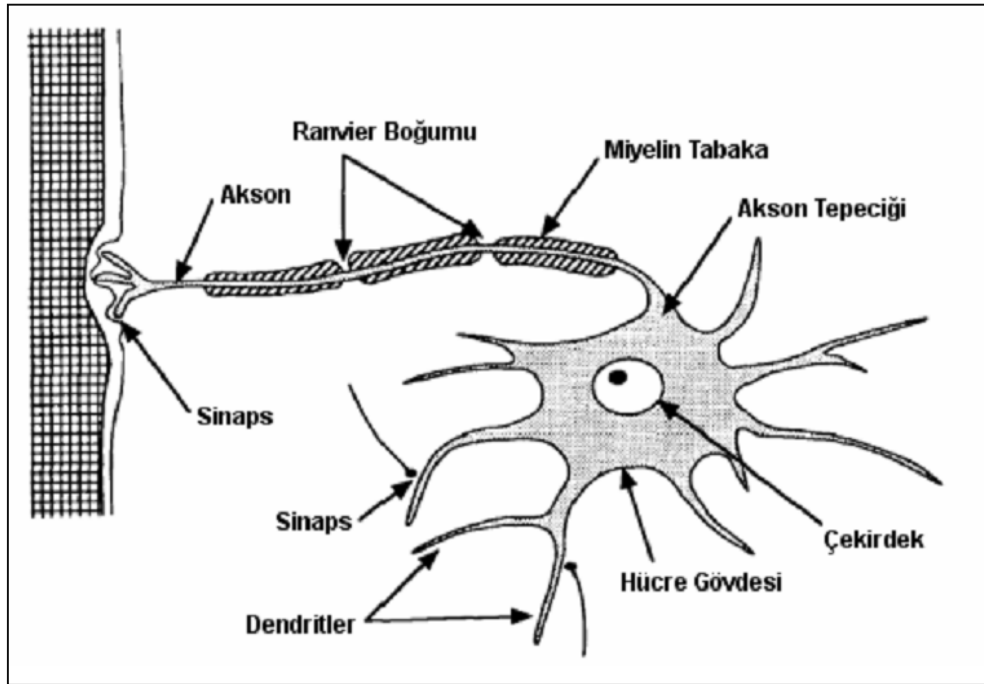
3.1 Biyolojik Sinir Ağları

Yapay sinir ağları beynin tüm yapısını olmasa da sinir hücrelerini modelleyerek çalıştığı için öncelikle biyolojik sinir hücrelerini tanımakta yarar vardır. Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin bulunduğu üç katmanlı bir sistem olarak açıklanmaktadır. Bu katmanlar; çevreden gelen girdileri elektriksel sinyallere dönüştürerek beyne ileten **Alıcı Sinirler** (Receptor), beynin ürettiği elektriksel sinyalleri çıktı olarak uygun tepkilere dönüştüren **Tepki Sinirleri** ve üçüncü olarak da bu alıcı sinirler ile tepki sinirleri arasında ileri ve geri besleme yaparak sinyallere uygun tepkiler üreten **Merkezi Sinir Ağı** olarak sıralanır (Saraç, 2004, s. 18).

İnsan beyninde bulunan sinir hücrelerine tıp literatüründe “nöron” adı verilmektedir. Nöron; sinir sisteminin temelini oluşturan çok özel bir hücredir. Nöronlar vücuda sinyal yaymada eşsizdirler. Bu sinyaller, hareket potansiyelleri ya da sinir uyarıları (impuls) olarak nitelendirilirler (Samuel, 2009). Nöronlar (sinir hücreleri) çevreden aldıkları hücresel bilgiyi işleme ve iletme yetenekleriyle donatılmış çok özel hücrelerdir.

Bir sinir hücresi; kendisine gelen uyarıyı yorumlayan **hücre gövdesi** (soma), bu gövdeyi çevreleyen, saç teline benzeyen ve biyolojik çevreden gelen uyarıları almakla görevli olan **dentrit**lerden (sinyal alıcıları - dendrite) ve hücre gövdesine bağlı olan, bu gövdeden aldığı bilgiyi diğer nöronlara ileten, kuyruk şeklindeki **akson**dan (axon) meydana gelmektedir. Temel olarak sinir hücrelerinde dentritler üzerinden girişler (sinaptik uçlara gelen sinirsel akımlar) alınmakta ve bu girişler hücre gövdesi tarafından genelde doğrusal olmayan bir şekilde işlenmektedir. Sinir hücresindeki sinyalleri taşıyan uzun bir sinirsel bağlantı halindeki akson ise, işlenen girişleri çıkışa aktarmaktadır. Aksonlar,

girişlerin iletirme hızını artıran bir çeşit yalıtım maddesi olan **miyelin tabaka** (myelin sheath) ile kaplıdır. Miyelin tabakanın üzerinde birkaç milimetrede bir yer alan ve girişleri periyodik olarak yeniden üretmeye yarayan **ranvier boğumu** (ranvier node) yer almaktadır. Akson - dendrit bağlantısı ise **sinaps** (synapse) olarak adlandırılmaktadır. Sinaps, sinir hücreleri arasındaki elektrokimyasal bağlantıyı sağlamaktadır (Baş, 2006, s. 18). Biyolojik bir sinir hücresinin temel yapısı Şekil 3.1’de gösterilmiştir.



Şekil 3.1: Biyolojik sinir hücresinin yapısı (Freeman and Skapura, 1991, s. 10).

Bir insanda ortalama 10^{11} sinir hücresi bulunduğu (Ndabahaliye, 2002) göz önüne alınırsa, çok büyük çoklukta sinapsın bulunduğu ortaya çıkar. Sinir hücrelerinin iletim hızları bilgisayarın iletim hızından daha yavaş olsa da sayısız sinapsın oluşturduğu bağlantılar, aynı anda birden fazla karar verilmesini ve beynin duyuşal verileri çok hızlı değerlendirmesini sağlamaktadır. Bu açıdan düşünüldüğünde insan beyninin karmaşık yapısı hala tam olarak çözülememiş ve birçok fonksiyon hala tam olarak açıklanamamıştır.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik bağlantıların ayarlanması ile oluşur. İnsanlar doğumlarından itibaren yaşayarak öğrenme sürecine girerler. Yaşadıkça beyin sürekli gelişmekte, yeni sinapslar oluşmakta, var olan sinapsların bağlantıları ve hücrelerin eşik değerleri ayarlanmaktadır. Öğrenme bu şekilde olmaktadır (Bayır, 2006, s. 17).

Tablo 3.1’de insan beyni ile bilgisayarların sahip oldukları özellikler sayısal değerlerle karşılaştırılmıştır:

Tablo 3.1: İnsan beyni ile bilgisayar sistemlerinin karşılaştırılması (Crone, 2004, s. 8).

Özellikler	İnsan Beyni	Bilgisayar
İşlemci Hızı	10^{-3} ms (0,25 MHz)	10^{-9} ms (2500 MHz)
Nöron / Transistör	10^{11} & 10^{33} bağlantı	10^9 (Yonga)
Ağırlık	1500 gr.	Kilogramlarca
Enerji Tüketimi	10^{-16} Joule	10^{-6} Joule
Hesaplama	100 Adım	Milyarlarca Adım

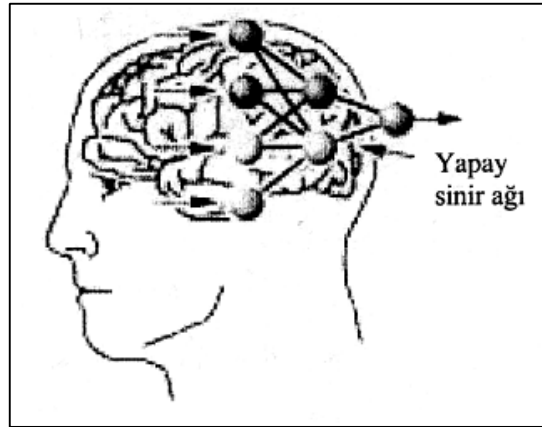
3.2 Yapay Sinir Hücreleri

Biyolojik sinir ağlarının sinir hücrelerinden oluşması gibi, yapay sinir ağları da yapay sinir hücrelerinden oluşmaktadır. Bir yapay sinir ağı birbiriyle bağlantılı birçok yapay sinir hücresinden meydana gelmektedir. Aynı zamanda yapay sinir hücreleri biyolojik sinir hücrelerinin basit bir modelidir. Bu yüzden aralarında yapısal olarak bazı benzerlikler vardır. Bu benzerlikler Tablo 3.2’de gösterilmiştir (Sağiroğlu, Beşdok ve Erler, 2003, s. 30):

Tablo 3.2: Biyolojik sinir sistemi ile YSA'nın yapısal benzerlikleri

Biyolojik Sinir Sistemi	Yapay Sinir Ağları
Sinir Sistemi	Sinirsel Hesaplama Sistemi
Nöron (Sinir Hücresi)	İşlemci Eleman (Yapay Sinir Hücresi)
Dendrit	Toplama Fonksiyonu
Hücre gövdesi (Soma)	Transfer (Aktivasyon) Fonksiyonu
Aksonlar	İşlemci Eleman Çıktısı
Sinapslar	İşlemci Elemanlar Arasındaki Bağlantı Ağırlıkları

Ayrıca biyolojik sinir ağlarının yapay sinir ağlarına benzerliği şematik olarak Şekil 3.2'de gösterilmiştir (Öztemel, 2012, s. 48):

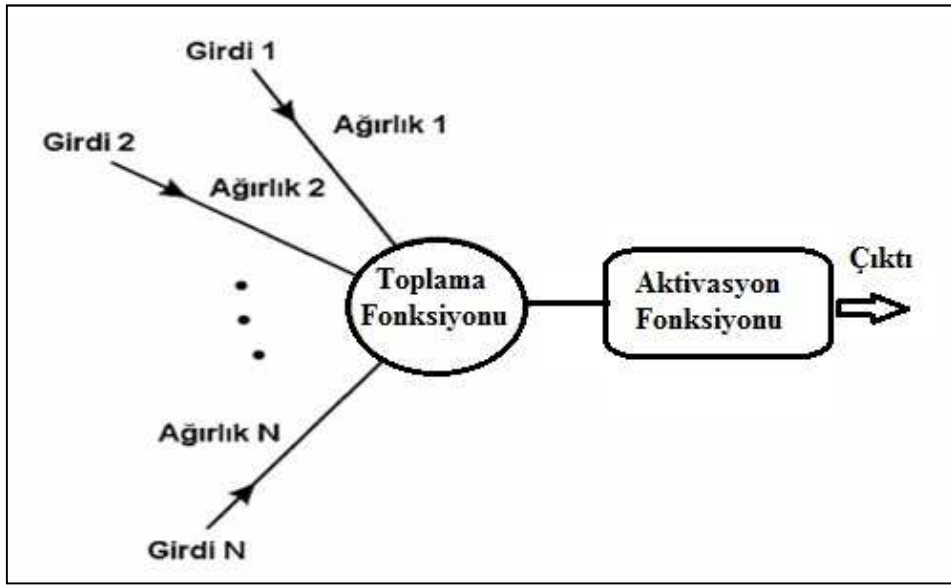


Şekil 3.2: Biyolojik ve yapay sinir ağı

Yapay sinir ağlarının içinde bulunan tüm sinir hücreleri (nöronlar), pek çok giriş sinyalini eş zamanlı olarak toplayabilen; ancak bağlantı ağırlık değerlerine, giriş sinyaline, eşik değerine ve aktivasyon fonksiyonuna bağlı olarak, yalnızca bir tek çıkış sinyali verebilen aygıtlardır (Hertz, Krogh and Palmer, 1991, s. 6). Bu çıktı yapay sinir ağının dışına verilen bir çıktı olabileceği gibi başka bir yapay sinir hücresine girdi olarak da verilebilir. Bir yapay sinir hücresi genel olarak beş temel bileşenden oluşmaktadır (Baş, 2006, s. 21).

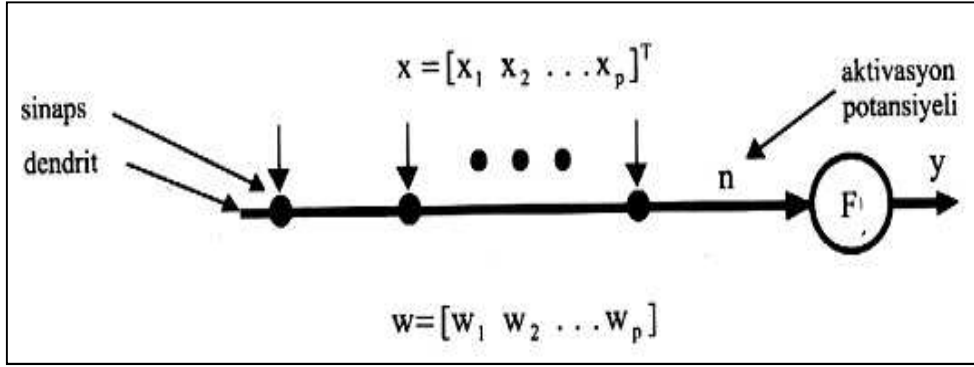
- Girdiler (Inputs)
- Ağırlıklar (Weights)
- Toplama Fonksiyonu
- Aktivasyon (Transfer) Fonksiyonu
- Çıktı (Output)

Aşağıda Şekil 3.3'te bir yapay sinir hücresinin temel bileşenleri şematik olarak gösterilmektedir.

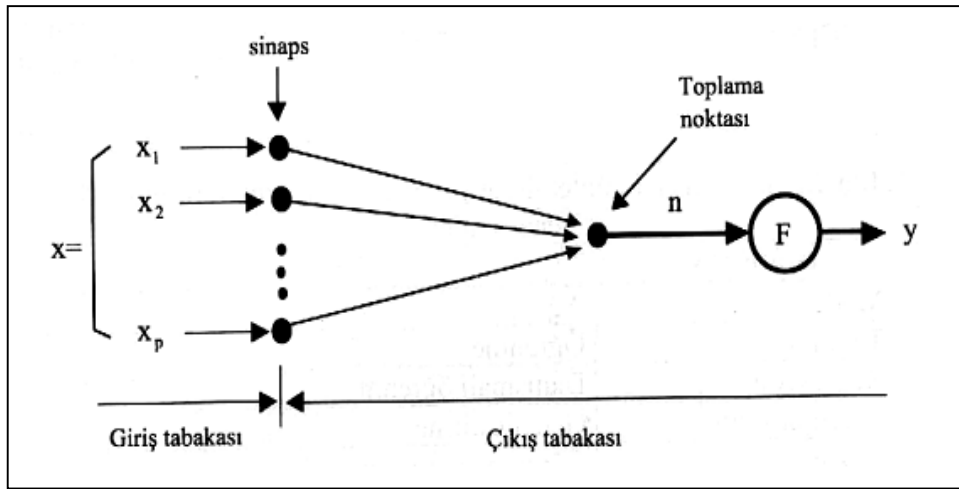


Şekil 3.3: Yapay sinir hücresinin yapısı (Tarassenko, 1998, s.7)

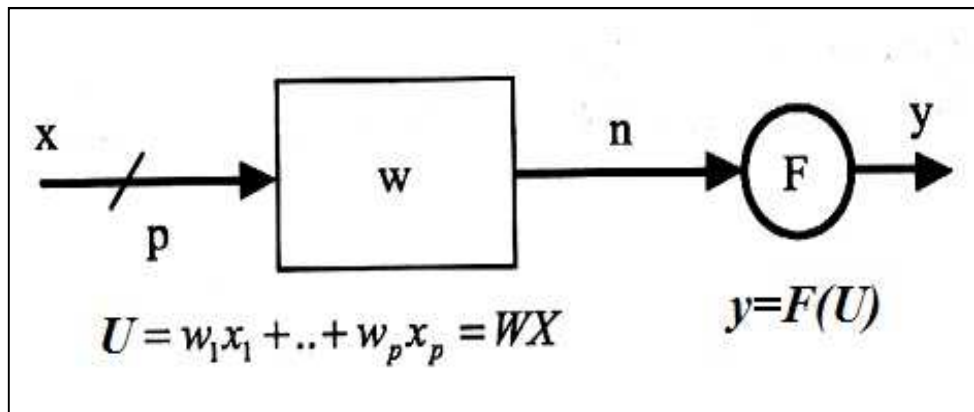
Yukarıda verilen şeklin yanı sıra bir yapay sinir hücresinin farklı şekillerde gösterimi de mevcuttur. Bunlar Şekil 3.4, 3.5, 3.6 ve 3.7 de gösterilmiştir (Sağiroğlu vd, 2003, s. 34):



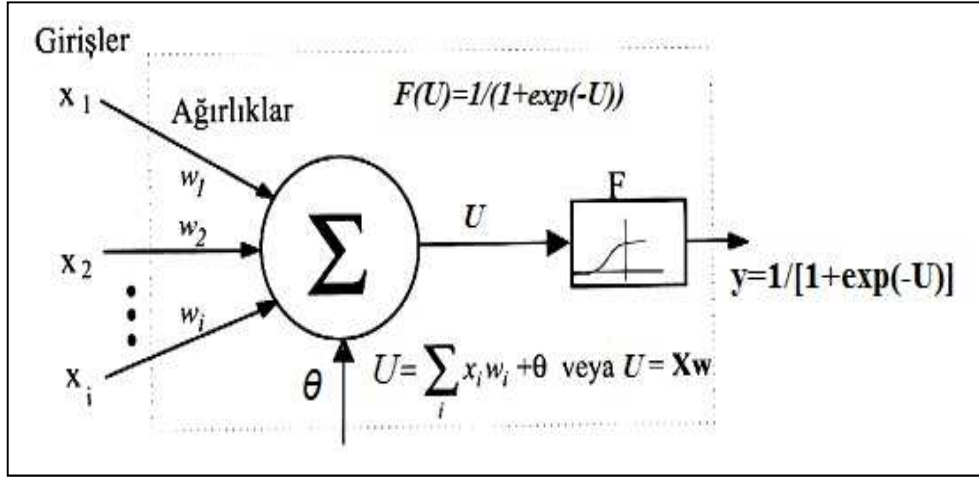
Şekil 3.4: Dendrit gösterimi



Şekil 3.5: İşaret akış grafi gösterimi



Şekil 3.6: Blok diyagram gösterimi



Şekil 3.7: YSA yapısının detaylı gösterimi

3.2.1 Girdiler

Bir YSA sisteminin girdileri (x_1, x_2, \dots, x_n) , yapay sinir hücresine gelen bilgilerdir. Bu bilgiler ya dış ortamdan direkt olarak yapay sinir ağına girer, ya da diğer bir sinir hücresinden gelir. Dış ortamlardan gelen bilgiler ağı eğitmek amacıyla kullanılan örnek verilerdir. Sistemin bağımsız değişkenleri olarak da nitelendirilebilir.

3.2.2 Ağırlıklar

Sistemde ağırlıklar (w_1, w_2, \dots, w_n) , yapay sinir hücresine giren her bir verinin hücre içinde hangi oranda ve hangi ağırlıkta değerlendirmeye alınacağını belirten katsayılarıdır. Ağırlıkların büyük ya da küçük olması önemli veya önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması o ağı için en önemli olay olabilir. Negatif değerler önemsiz demek değildir. Ağırlıkların sayısal değerlerinin pozitif veya negatif olması ağı etkisinin pozitif veya negatif olmasına işaret eder (Öztemel, 2012, s. 49). Bir yapay sinir ağına girdi olarak sunulan bilgiler, bağlantılar üzerindeki ağırlıklar vasıtasıyla hücreye girmekte ve ağırlıklar yapay sinirde girdi olarak kullanılacak değerlerin göreceli kuvvetini

(matematiksel katsayısını) göstermektedirler. YSA içinde girdilerin hücreler arasında iletimini sağlayan tüm bağlantıların farklı ağırlık değerleri bulunur. Böylelikle ağırlıklar her işlemci elemanın her girdisi üzerinde etki yapmış olur. Ağırlıklar değişken olabileceği gibi sabit değerler de olabilir (Baş, 2006, s. 22).

3.2.3 Toplama Fonksiyonu

Toplama fonksiyonu, hücreye gelen net girdiyi hesaplayan fonksiyondur ve n tane girişin her birinin kendi ağırlığıyla çarpımına eşit olan toplam;

$$U = \sum_{i=1}^n x_i w_i \quad (3.1)$$

şeklinde hesaplanır. 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 çeşidini bulmak için herhangi bir formül yoktur. Toplama fonksiyonu genellikle deneme yanılma yoluyla bulunmaktadır. Ayrıca bir yapay sinir ağındaki bütün işlemci elemanların aynı toplama fonksiyonuna sahip olması gibi bir zorunluluk da yoktur. Bazen aynı yapay sinir ağı içindeki işlemci elemanların bazıları aynı toplama fonksiyonunu, diğerleri ise farklı fonksiyonu kullanabilirler. Bu seçim tamamen tasarımcının kendi kararına bağlıdır (Öztemel, 2012, s. 49).

Ancak birçok uygulamada θ eşik değeri de bu toplamda göz önüne alınmıştır. Böylece toplama fonksiyonu;

$$U = \sum_{i=1}^n x_i w_i + \theta \quad \text{veya} \quad U = \sum_{i=1}^n x_i w_i - \theta \quad (3.2)$$

şeklini alır. θ eşik değerinin girişlerden bağımsız olduğu için bütün girişlerin sıfır olması durumunda çıkış değerlerinin sıfır değil de eşik değerine eşit olduğu görülür ki bu da, belirtilen şartlar altında nöron çıkışının sıfır olması zorunluluğunu ortadan kaldırır. Eşik değerinin kullanımı, toplama fonksiyonuna $+1$ ya da -1 değerine sahip sabit bir girişin θ ağırlığına sahip bir bağlantı ile eklendiği şeklinde yorumlanır (Öztemel, 2012, s. 60).

Ayrıca her model ve her uygulama için bu toplama fonksiyonunun kullanılması da şart değildir. Bununla birlikte bazı durumlarda toplama fonksiyonu için girişlerin değeri önemli olurken, bazen de sayısı önemli olabilmektedir. Bazı modeller, kullanılacak toplama fonksiyonunu kendileri belirler. Çoğu zaman daha karmaşık olan değişik toplama fonksiyonları kullanılır. Bu toplama fonksiyonları Tablo 3.3'te gösterilmiştir (Öztemel, 2012, s. 50).

Tablo 3.3: Toplama fonksiyonları ve özellikleri

Toplama Fonksiyonu	Açıklama
<p>Çarpım</p> $\text{Net Girdi } (U) = \prod_{i=1}^n x_i w_i$	<p>Net girdiyi hesaplamak için öncelikle girdi değeriyle o girdinin ağırlığı çarpılır. Daha sonra bu değerler de birbiriyle çarpılır.</p>
<p>Maksimum (En Çok)</p> $\text{Net Girdi } (U) = \max(x_i w_i), i = 1, \dots, n$	<p>Girdi değerleri kendi ağırlıklarıyla çarpılır ve en büyük değer (maksimum) net girdi olarak kabul edilir.</p>
<p>Minimum (En Az)</p> $\text{Net Girdi } (U) = \min(x_i w_i), i = 1, \dots, n$	<p>Girdi değerleri kendi ağırlıklarıyla çarpılır ve en küçük değer (minimum) net girdi olarak kabul edilir.</p>
<p>Çoğunluk</p> $\text{Net Girdi } (U) = \sum_{i=1}^n \text{sgn}(x_i w_i)$	<p>n adet girdi ve ağırlıklarıyla çarpımı hesaplanır. Negatif olanlar ile pozitif olanların sayısı bulunur. Bunlardan büyük olan net girdi olarak alınır.</p>
<p>Kümülatif Toplam</p> $\text{Net Girdi } (U) = \text{Net}(\text{eski}) + \sum_{i=1}^n x_i w_i$	<p>Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce elde edilen bilgilerle toplanarak hücrenin net girdisi bulunur.</p>

3.2.4 Aktivasyon (Transfer) Fonksiyonu

Aktivasyon fonksiyonunun seçimi yapay sinir ağları metodolojisinde önemli işlemlerden biridir ve bu fonksiyonun doğru seçilmesi ağın performansını önemli derecede etkiler. Literatürde bilinen ondan fazla aktivasyon fonksiyonu

mevcuttur. Genellikle bu fonksiyonların seçimi keyfi olarak yapılır. İlgilenilen problemin çeşidine ve ağ yapısına bağlı olarak değişebilir. Bununla beraber 1980'lerin sonunda ve 1990'ların başında transfer fonksiyonu olarak sigmoid fonksiyonu sıkça kullanılmıştır (Galushkin, 2007, s. 358).

Aktivasyon fonksiyonunun amacı, söz konusu nöronları belirlenmiş bir çıkış aralığı içinde sınırlamaktır. Girdilerin hesaplanan ağırlıklı toplamlarına bu aktivasyon fonksiyonu uygulanarak bu toplamlar çıktı değerine dönüştürülür. Çoğu durumda aktivasyon fonksiyonu net girdiyi -1 ile +1 arasına ya da 0 ile 1 arasına eşler. Bu tür aktivasyon fonksiyonları özellikle sınıflandırma işlevleri için kullanışlıdır. Herhangi bir reel değer üretmenin gerekli olduğu durumlarda ise son katmanda doğrusal (lineer) aktivasyon fonksiyonu kullanılabilir (Kamruzzaman, Sarker and Begg, 2006, s. 4).

Aktivasyon fonksiyonu genellikle doğrusal olmayan fonksiyondur. Doğrusal fonksiyonlarda çıktı ile girdiler orantılı olduğundan YSA'da bu fonksiyonlar fazla tercih edilmez. Aynı zamanda bu durum, ilk YSA denemelerinin başarısızlıkla neticelenmesinin temel sonucudur. Bilindiği gibi XOR probleminin çözülemeyip YSA çalışmalarının durması bu sebeptir. Bununla birlikte doğrusal olmayan fonksiyonların kullanılması YSA'nın çok karmaşık ve farklı problemlere uygulanmasını sağlamıştır.

Genellikle kullanılan transfer fonksiyonları doğrusal (lineer) fonksiyon, parçalı doğrusal fonksiyon, basamak (step) fonksiyonu, sigmoid fonksiyonu ve hiperbolik tanjant fonksiyonudur. Özellikle logistik (sigmoid) fonksiyonu ve hiperbolik tanjant fonksiyonu oldukça yaygındır. Çünkü bu fonksiyonlar temel diferansiyellenebilme özelliklerini sağlarlar. Buna ilaveten bu iki fonksiyon monotondur ve ara değerlerde değişim oranı çok büyük ve uç (extreme) değerlerde ise değişim oranı küçüktür. Bu özellikleri sayesinde, çıkış değer(ler)inin uç değerlerden biri olmasını mümkün kılarlar (Fyfe, 1996, s. 41).

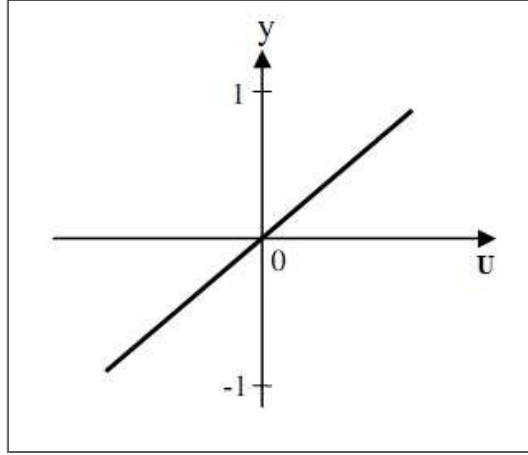
$$\begin{aligned} f(x) = \tanh(bx) &\Rightarrow f'(x) = b(1 - (f(x))^2) \\ f(x) = \frac{1}{1 + e^{-bx}} &\Rightarrow f'(x) = bf(x)(1 - f(x)) \end{aligned} \quad (3.3)$$

3.2.4.1 Doğrusal (Linear) Fonksiyon

Doğrusal fonksiyon, hücreye gelen girişleri olduğu gibi çıkışa verir. Doğrusal problemlerin çözümünde kullanılan bu fonksiyon

$$U = \sum_{i=1}^n x_i w_i + \theta \quad \text{veya} \quad U = \sum_{i=1}^n x_i w_i - \theta \quad (3.4)$$

olmak üzere $y = F(U) = AU$ şeklinde tanımlanmaktadır. Burada A sabit bir katsayıdır. Çoğunlukla ADALINE (Adaptive Linear Element) olarak bilinen doğrusal model, klasik işaret işleme ve regresyon analizinde kullanılır. Doğrusal fonksiyon grafiksel olarak şu şekilde gösterilebilir:

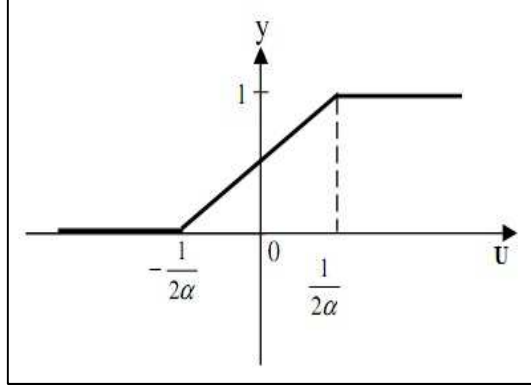


Şekil 3.8: Doğrusal (linear) fonksiyon

3.2.4.2 Parçalı Doğrusal Fonksiyon

Bu fonksiyon, küçük aktivasyon potansiyeli için α kazancı olan bir doğrusal toplayıcı (Adaline) olarak çalışır. Büyük aktivasyon potansiyeli için nöron doyuma ulaşır ve çıkış işareti 1 olur. Büyük kazançlar için, $\alpha \rightarrow \infty$ iken, parçalı doğrusal fonksiyon basamak fonksiyonu gibi davranır. Parçalı doğrusal fonksiyonun denklemi ve grafiği aşağıdaki gibidir:

$$y = F(U) = \begin{cases} 0 & U \leq -\frac{1}{2\alpha} \\ \alpha U + \frac{1}{2} & |U| < \frac{1}{2\alpha} \\ 1 & U \geq \frac{1}{2\alpha} \end{cases} \quad (3.5)$$



Şekil 3.9: Parçalı doğrusal (lineer) fonksiyon

3.2.4.3 Basamak (Step) Fonksiyon

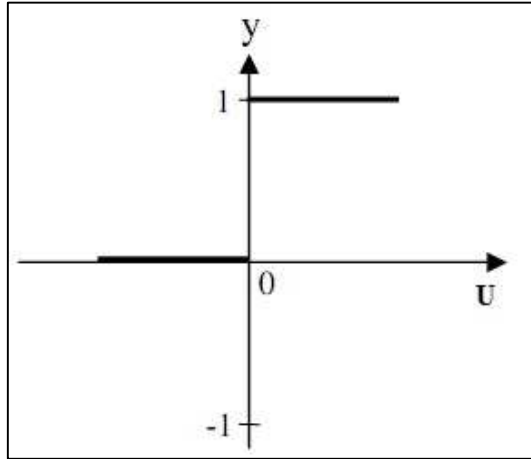
Basamak fonksiyonunda sisteme gelen net girdi değerinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre sistemin çıktı değeri 1 veya 0 olmaktadır. Perceptron (basit algılayıcı model) olarak bilinen yapay sinir hücresi aktivasyon fonksiyonu olarak bu fonksiyonu kullanır. Basamak fonksiyon tek veya çift kutuplu olabilir.

3.2.4.3.1 Tek Kutuplu (Unipolar) Basamak Fonksiyonu

Genel denklemi

$$y = F(U) = \begin{cases} 1 & U \geq 0 \\ 0 & U < 0 \end{cases} \quad (3.6)$$

şeklinde olan tek kutuplu basamak fonksiyonunun grafiği aşağıdaki gibidir:

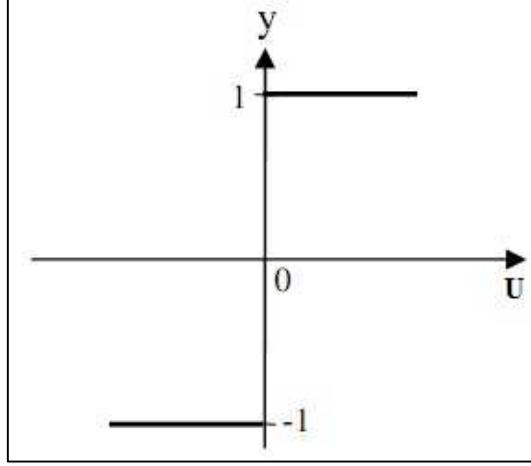


Şekil 3.10: Tek kutuplu (unipolar) basamak fonksiyonu

3.2.4.3.2 Çift Kutuplu (Bipolar) Basamak Fonksiyonu

Basamak fonksiyonlarının YSA'da kullanılan diğer bir çeşidi de çift kutuplu olanıdır. Bu fonksiyonun genel denklemi ve grafiği aşağıdaki gibidir:

$$y = F(U) = \begin{cases} +1 & U \geq 0 \\ -1 & U < 0 \end{cases} \quad (3.7)$$

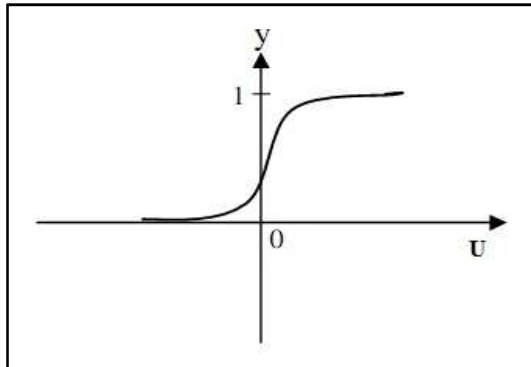


Şekil 3.11: Çift kutuplu (bipolar) basamak fonksiyonu

3.2.4.4 Sigmoid Fonksiyon

Uygulamalarda en çok kullanılan ve en popüler olan aktivasyon fonksiyonudur. Fonksiyonun formülü Denklem 3.8’de, şekli ise Şekil 3.12’de gösterilmiştir. Fonksiyonun en aktif bölgesi 0,2 ile 0,8 arasındadır. Tek kutuplu fonksiyon olarak da adlandırılmaktadır (Duman, 2006, s. 24; Aizenberg, 2011, s. 25).

$$y = F(U) = \frac{1}{1 + e^{-U}} = \frac{1}{2} [\tanh(U/2) + 1] \quad (3.8)$$



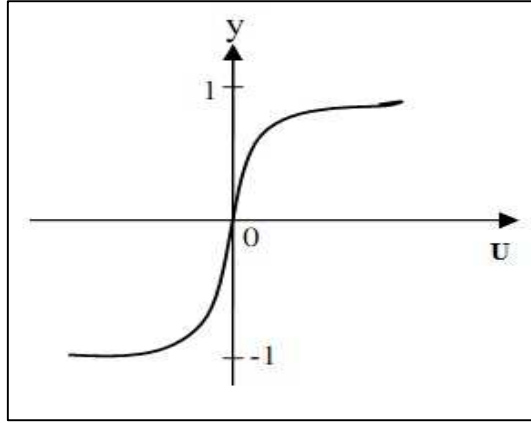
Şekil 3.12: Sigmoid tipi fonksiyon

3.2.4.5 Tanjant Hiperbolik Fonksiyon

YSA uygulamalarında en çok kullanılan aktivasyon fonksiyonlarından biri de Tanjant Hiperbolik fonksiyondur. Bu fonksiyon çift kutuplu fonksiyon olarak da bilinir. Giriş uzayının genişletilmesinde etkin bir şekilde kullanılan bir fonksiyondur. Fonksiyonun formülü Denklem 3.9'da, şekli ise Şekil 3.13' te verilmiştir:

$$y = \tanh(\beta U) = \frac{1 - e^{-2U}}{1 + e^{2U}} \quad (3.9)$$

Burada β parametresi fonksiyonun eğimini kontrol eder ve genellikle $\beta = 1$ alınır.

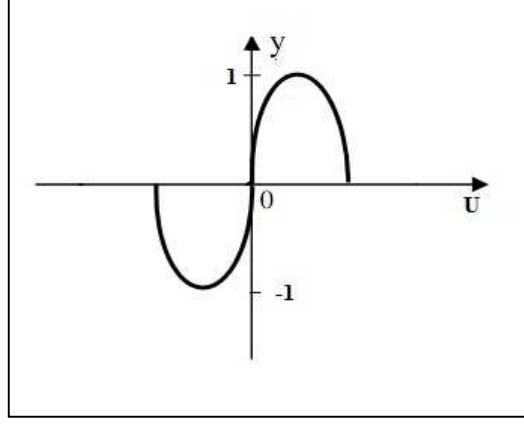


Şekil 3.13: Tanjant hiperbolik fonksiyon

3.2.4.6 Sinüs Tipi Fonksiyon

Öğrenilmesi düşünülen olaylar sinüs fonksiyonuna uygun dağılım gösteriyorsa böyle durumlarda aktivasyon fonksiyonu olarak sinüs fonksiyonu kullanılabilir. Fonksiyonun formülü Denklem 3.10'da ve şekli ise Şekil 3.14'de verilmiştir (Duman, 2006, s. 25):

$$y = F(U) = \sin U \quad (3.10)$$



Şekil 3.14: Sinüs fonksiyonu

3.2.5 Çıktı

Bir YSA'da çıkış işlevi, yani $y = F(U)$, kullanılan aktivasyon fonksiyonunun sonucunun diğer nöronlara ya da sistem dışına iletilmesidir. Aynı zamanda bir nöronun çıkışı kendisine veya kendisinden sonra gelen nöron(lar)a giriş olabilir. Bir işlem elemanının çok sayıda girdi değeri olabilirken sadece tek bir çıktı değeri bulunmaktadır.

Her bir düğümün çıkışı aktivasyon fonksiyonunun sonucuna eşittir. Fakat bazı ağ yapılarında komşu düğümler arasında yarışma oluşturmak için aktivasyon fonksiyonları yeniden düzenlenebilir. Bunun sonucunda yarışmacı düğümlerin girişleri, hangi düğümlerin öğrenme işlemine katılacağına karar verilmesine yardımcı olur (Öztemel, 2012, s. 51; Elmas, 2003, s. 35).

Bir yapay sinir ağının bu 5 temel elemanı dışında zaman zaman ihtiyaç duyulduğunda kullanılan bir elemanı daha vardır. Bu eleman ölçekleme ve sınırlama olarak adlandırılır.

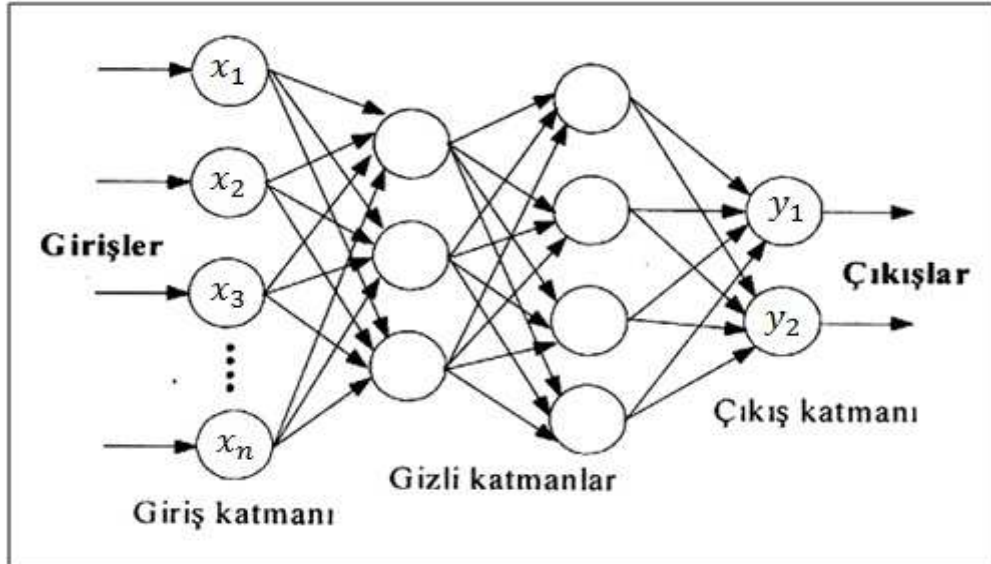
3.2.6 Ölçekleme ve Sınırlama

Etkinlik işlevinin sonuçları ölçek veya sınır işlemlerinden geçirilebilir. Ölçeklendirme basit bir şekilde bir ölçek etmeni ile aktivasyon değerinin

çarpılması sonucu elde edilir. Sınırlama ise, ölçeklenen bu sonuçların en az ve en çok sınırlarını aşmasına engel olmaktır (Elmas, 2003, s.35).

3.3 Yapay Sinir Ağlarının Yapısı

Birden fazla yapay sinir hücresi bir araya gelerek yapay sinir ağını oluşturur. Bu hücrelerin bir araya gelmesi rastgele olmaz. Ağdaki sinir hücreleri katmanlar içine yerleştirilir. Hücreler birbirleriyle bağlanarak; üç ya da daha fazla katman halinde ve her katman içinde paralel olarak bir araya gelerek ağ oluştururlar. Girdi katmanındaki süreç elemanları dış ortamdan bilgileri alarak ara katmanlara transfer etmekten sorumludur. Ara katmanlar kullanılarak girdi katmanından gelen bilgiler işlenip çıktı katmanına gönderilir. Bilgiler ara katmanda işlenir. Çıktı katmanı süreç elemanları, ara katmandan kendilerine gelen bilgileri işleyip ağın girdi katmanından sunulan girdi seti için üretilmesi gereken çıktıyı üretirler. Bu yapının daha iyi anlaşılması için Şekil 3.15’de çok katmanlı bir yapay sinir ağı şematik olarak verilmiştir (Öztemel, 2012, s. 52, Elmas, 2003, s. 45):



Şekil 3.15: Çok katmanlı bir yapay sinir ağı modeli

3.3.1 Katmanlar

Bir yapay sinir ağı sisteminde ağ yapısı oldukça basittir. Sistemde bulunan nöronların bir kısmı girişleri alırken bir kısmı da sistemin çıktılarını vermektedir. Giriş ve çıkış arasında kalan diğer katmanlar ise dış dünya ile direkt bağlantılı olmadığından gizli katman olarak nitelendirilirler. Nöronların toplandığı ve sistemi oluşturan bu katmanlar (tabakalar);

- Girdi katmanı,
- Gizli katman (lar),
- Çıktı katmanı

olmak üzere üç kısımda incelenir.

3.3.1.1 Girdi Katmanı (Input Layer)

Dış ortamdan alınan giriş bilgilerini gizli tabakalara transfer eden nöronlardan oluşmaktadır. Girdi katmanı, girdiler için bir tampon görevindedir ve nöronlar aldıkları giriş değerleri üzerinde herhangi bir işlem yapmaksızın bir sonraki tabakaya iletimini sağlarlar. Burada YSA'nın girdi değerleriyle işlem yapabilmesini sağlamak için girdi değerlerinin sadece nümerik ifadeler olması gerekmektedir (Tan, 1997, s. 27).

3.3.1.2 Gizli Katman (Hidden Layer)

Girdi katmanı ile çıktı katmanı arasında bulunan ve girdi tabakasından gelen bilgileri işleyerek çıktı tabakasına iletmeye işlevini yapan gizli katman(lar), özellikle karmaşık problemleri hesaplayabilmek için sistemde yer alması gereken kısımlardır. Ancak gizli katmanların sayısı arttıkça ağda genelleşme sorunu ortaya çıkar. Bu nedenle ağdaki gizli katman sayısı, sistem en iyi çıktıyı verecek şekilde seçilmelidir (Dere, 2009, s. 58). Gizli tabakanın oluşturduğu kısım tek tabakadan oluşabileceği gibi daha fazla tabakadan da oluşması mümkündür.

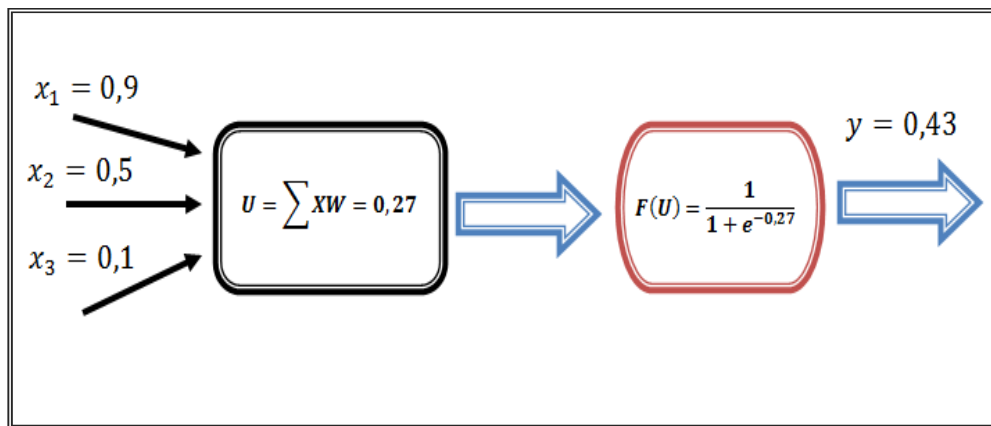
Tabakada yer alan çok sayıda nöronlar ağ yapısı içerisinde diğer nöronlarla bağlantılı bir yapıya sahiptir. Bu nedenle, gizli tabakada bulunması gereken nöron sayısı seçimi oldukça önemli olup dikkatli bir şekilde yapılmalıdır (Öztemel, 2012, s. 52).

3.3.1.3 Çıktı Katmanı (Output Layer)

Çıktı nöronu veya çıkış birimi ise, sistem dışına sinyal yollayan nöronudur. Bu katmandan elde edilen veri ele alınan problemin sonucudur. Örneğin, bir kredi uygulamasında sonuç evet ya da hayır olabilmektedir. YSA, sonuç olarak sayısal değerler atadığından dolayı böyle bir sistem için evet sonucu “1”, hayır sonucu ise “0” olarak değerlendirilir (Medsker, Turban ve Trippi, 1996, s. 8).

3.3.2 Yapay Sinir Ağlarının Çalışma Prensibi

Yapay sinir ağlarının çalışma prensibini basit bir örnekle anlatmak mümkündür. Aşağıdaki Şekil 3.16’da bir yapay sinir hücresinin çalışma şekli gösterilmiştir:



Şekil 3.16: Bir yapay sinir ağının çalışma örneği

Yapay sinir hücresine gelen net girdi bilindiği gibi ağırlıklar ile girişlerin çarpılması sonucu elde edilir. Verilen örnekte girişlerin sahip olduğu ağırlıklar $w_1 = 0,4$, $w_2 = -0,2$ ve $w_3 = 0,1$ olarak alınırsa hücreye giren net girdi;

$$\text{Net Girdi} = U = \sum XW = 0,9 * 0,4 + 0,5 * (-0,2) + 0,1 * 0,1 = 0,27 \quad (3.11)$$

şeklinde bulunur. Hücrenin Sigmoid tipli aktivasyon fonksiyonuna göre çıkışı ise şu şekildedir:

$$y = F(U) = \frac{1}{1 + e^{-0,27}} = 0,43 \quad (3.12)$$

Sonuç olarak verilen girdilere karşılık yukarıdaki işlemler sonucunda çıkış değeri bulunur.

3.3.3 İletim ve Bağlantı Çeşitleri

Bir sinir hücresinin çıkışının, diğer bir sinir hücresine giriş olarak bağlandığını ve bu bağlantının genellikle tek yönlü olmadığını, iki sinir arasında çift yönlü olduğunu söylemek mümkündür. Katmanda yer alan sinirlerin kendi aralarında bir bağlantı yapıları bulunmaktadır (Elmas, 2003, s. 46).

Katmanlar arasında kullanılan değişik türde bağlantılar vardır. Bu bağlantılara “**Katmanlar Arası Bağlantı**” denir. Bu bağlantı çeşitleri aşağıdaki gibidir (Küçükönder, 2011, s. 47):

- **Tam Bağlantılı:** Bir katmanda yer alan her bir sinirin kendisinden sonra gelen katmanda bütün sinirlerle bağlantılı olması durumudur (Elmas, 2003, s. 46).
- **Kısmi Bağlantılı:** Bir katmanda yer alan her bir sinirin kendisinden sonra gelen katmanda bütün sinirlerle bağlantılı olmaması durumudur.
- **İleri Besleme:** Bir katmandaki sinirlerin kendinden sonra gelen katmanda yer alan sinirlere çıkışlarını göndererek kendinden sonra gelen bu katmandaki sinirlerden giriş almadığı bağlantı türüdür.

- **Çift Yönlü:** Bir katmandaki sınırlar sonraki katmandaki sınırlara çıkışlarını gönderirler ve aynı zamanda sonraki katmandaki sınırların çıkışları da önceki katmandaki sınırlara ulaşır.

İleri beslemeli ve çift yönlü bağlantılar aynı zamanda tam veya kısmi bağlantılı olabilir.

- **Hiyerarşik:** Düşük bir katmanda yer alan sınırlar kendisinden bir sonraki seviyede yer alan katmanın sınırlarına iletilebilir.
- **Rezonans:** Çift yönlü bağlantılı katmanlar kesin durum oluşuncaya kadar girişleri ya da çıkışları bir katmandaki sınırdan diğer katmandaki sinire defalarca göndermeye devam ediyorsa bu bağlantılar rezonans bağlantılardır.

Karmaşık yapıdaki yapay sinir ağlarında aynı katman içindeki sınırlar kendi aralarında haberleşebilirler. Bu haberleşme **Sınırlar Arası Bağlantı** vasıtasıyla sağlanmaktadır. İki çeşit bağlantı türü vardır (Elmas, 2003, s. 46):

1. Tekrarlamalı

Aynı katman içindeki sınırlar tam veya kısmi bağlantılı olabileceği için, bu katman, giriş alındığında çıkışları diğer katmana göndermeden önce aynı katmanda yer alan diğer sınırlar arasında bağlantı oluşturarak bu bağlantılar ile sinir çıkışının diğerine defalarca aktarılması işlemi gerçekleşir.

2. Merkezde / Çevre Dışı (On Center / Off Surround):

Aynı katmandaki iki sinir arasında uyarıcı ve yasaklayıcı olmak üzere iki çeşit bağlantı vardır. Uyarıcı bağlantıda bir sinirin çıkışı bağlı bulunduğu sinirin faaliyet potansiyelini artırır, buna karşılık yasaklayıcı sinir ise faaliyet potansiyelini azaltır. Katmandaki sinirin kendisi ve komşuları arasında uyarıcı bağlantı kurulurken, diğer sınırlar arasında yasaklayıcı bağlantı kurulur. Bu durum bir yarışma ortamı gibi düşünülebilir. Birkaç işaret değişiminden sonra etkin çıkış değerli sınırlar kazanmış olur.

4. YAPAY SINİR AĞLARININ SINIFLANDIRILMASI VE TEMEL ÖĞRENME KURALLARI

Yapay sinir ağlarının kullanım alanlarının gittikçe çoğalması ve daha popüler hale gelmesi araştırmacıları yapay sinir ağlarını çeşitli kriterlere göre sınıflandırmaya götürmüştür. Nöronların bağlantı şekilleri, oluşturdukları katman sayısı ve katmanlar arasındaki veri iletim şekli ağın yapısını yani topolojisini belirler. Başka bir deyişle ağ oluşturmak için bir araya gelen nöronların bağlantı şekline **Ağın Topolojisi** denir (Fiesler, 1996, s. 2).

Yapay sinir ağları; topolojilerine, öğrenme metotlarına ve kullanım amaçlarına göre çeşitli sınıflara ayrılmaktadır. Bu bölümde YSA'nın belirtilen özelliklere göre sınıflandırılması, temel öğrenme kuralları ve YSA modelleri üzerinde durulacaktır.

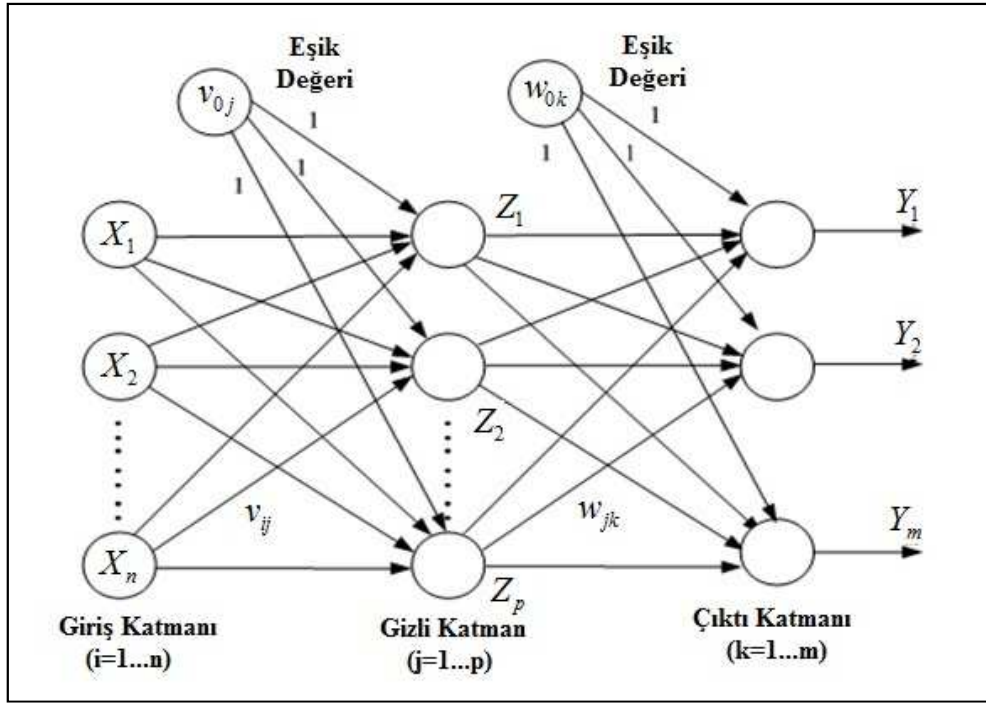
4.1 Yapay Sinir Ağlarının Sınıflandırılması

4.1.1 Topolojilerine Göre Sınıflandırılması

Bir ağın topolojisi (yapısı), ağın fonksiyonelliği ve performansı üzerinde önemli rol oynar. Ağın topolojisi kavramı ile bir ağın nöronlarının dizilişi ve bağlantılarının yapısı ifade edilir. Yapay sinir ağlarının topolojisi aynı zamanda işlemci elemanların birbirleriyle bağlantı şekillerine göre değişiklik göstermektedir. Problemlerin çözümü için ise, ağın topolojisine göre seçilen öğrenme algoritmaları yardımıyla ağın ağırlıkları belirlenir (Baş, 2006, s. 31). Topolojilerine göre YSA, İleri Beslemeli (Feed-Forward) ve Geri Beslemeli (Feed-Back) olmak üzere 2 kısımda incelenir.

4.1.1.1 İleri Beslemeli (Feed-Forward) Yapay Sinir Ağları

İlk yapay sinir ağları ile doğrusal olmayan ilişkiler öğrenilememekteydi. Bu sorunu çözmek için çok katmanlı ağlar geliştirilmiş ve 1986 yılında XOR probleminin çözümü Rumelhart ve arkadaşları tarafından yapılarak ileri beslemeli çok katmanlı bir ağ modeli geliştirilmiştir. Bu modele *hata yayma modeli* veya *geriye yayım modeli* (*backpropagation network*) de denmektedir. Şekil 4.1’de ileri beslemeli üç katmanlı (girdi katmanı, ara katman ve çıktı katmanı) bir yapay sinir ağı gösterilmektedir (Öztemel, 2012, s. 76).



Şekil 4.1: Çok katmanlı ileri beslemeli ağ (Kaynar ve Taştan, 2009, s. 164).

İleri beslemeli bir ağda işlemci elemanlar genellikle katmanlara ayrılmışlardır. Nöronlar arası iletişim giriş katmanından çıktı katmanına doğru tek yönlü bağlantılarla iletir. Nöronlar bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisindeki nöronlar birbirleriyle bağlantılı değildir. Bu yüzden ileri beslemeli ağlarda nöronlar arasındaki bağlantılar bir döngü

oluşturmamakta ve bu ağlar sisteme girilen verilere çok hızlı bir şekilde çıktı üretebilmektedir.

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 gizli katmandaki hücrelere iletir. Bilgi, gizli katmanda işlenip çıktı katmanına iletilerek ağın çıktısı oluşturulur. Bu yapı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli üç katmanlı yapay sinir ağlarının, gizli katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir (Saraç, 2004, s. 24).

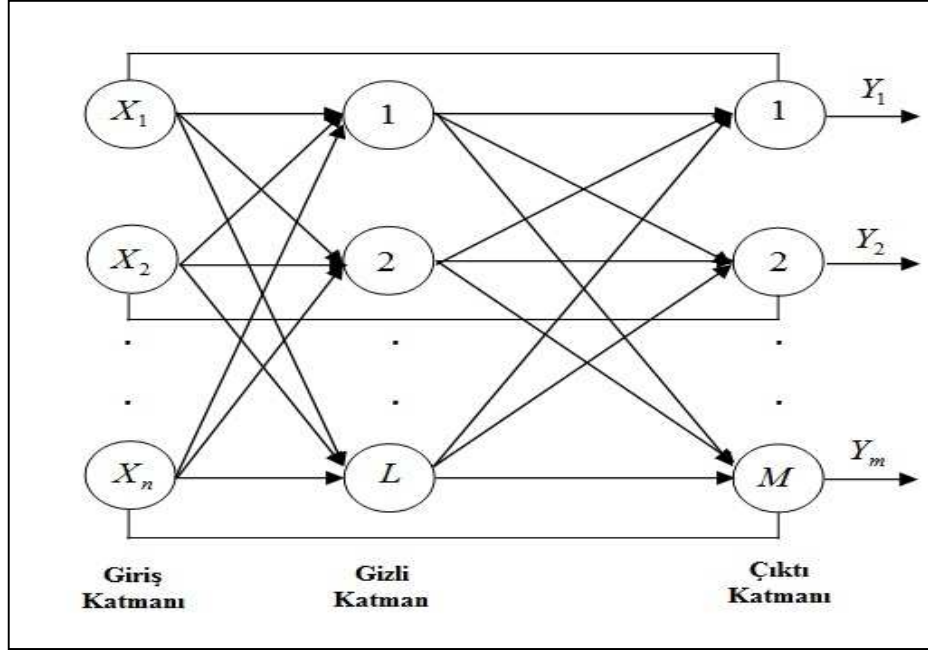
Bu tip yapay sinir ağlarının eğitiminde genellikle **Geri Yayılım Öğrenme Algoritması** kullanılmaktadır. Şekil tanıma, sinyal işleme ve sınıflandırma gibi problemlerin çözümünde genellikle bu topoloji uygulanmaktadır. İleri beslemeli ağlara örnek olarak **Çok Katmanlı Algılayıcılar** (Multi Layer Perceptron-MLP) ve **LVQ** (Learning Vector Quantization - Öğrenme Vektör Nicelendirmesi) ağları verilebilir (Bayır, 2006, s. 34).

4.1.1.2 Geri Beslemeli (Feed-Back) Yapay Sinir Ağları

Geri beslemeli ya da diğer adıyla yinelemeli (recurrent) yapay sinir ağları, ileri beslemeli ağların aksine dinamik bir yapıya sahiptir. Geri beslemeli ağlar, ara katmanlar veya çıktı katmanındaki işlemci elemanların giriş veya önceki ara katmanlardaki işlemci elemanlara geri beslendiği bir yapıya sahiptir. Böylece girdiler hem ileri hem de geri yönde aktarılmış olur. Ayrıca geri beslemeli yapay sinir ağları dinamik hafızaya sahiptir ve bir andaki çıktı hem o andaki hem de daha önceki girdileri yansıtır (Baş, 2006, s. 32).

Geri beslemeli yapay sinir ağlarında, en az bir işlemci elemanın çıktısı, kendisine ya da diğer işlemci elemanlara girdi olarak verilmekte ve genellikle besleme bir geciktirme elemanı (ara katman veya çıktı katmanındaki aktivasyon değerlerini bir sonraki iterasyona girdi olarak taşımakla görevli eleman) üzerinden yapılmaktadır. Geri besleme, bir katmandaki işlemci elemanlar arasında da

olabilmektedir. Bu yapısı sayesinde geri beslemeli yapay sinir ağı, nonlinear dinamik bir davranış sergilerler. Bu sayede, geri beslemenin yapılış şekline göre farklı yapı ve davranışta geri beslemeli ağı elde edilebilir (Saraç, 2004, s. 28).



Şekil 4.2: Geri beslemeli ağı yapısı

Şekil 4.2’de geri beslemeli ağların genel yapısı görülmektedir. Bu ağlarda iletim hızı ileri beslemeli ağlara nazaran daha düşüktür. Bunun sebebi, geri besleme döngülerinin ağın genelini belirli bir gecikmeye uğratmasıdır (Dede, 2008, s. 14).

4.1.2 Öğrenme Metotlarına Göre Sınıflandırılması

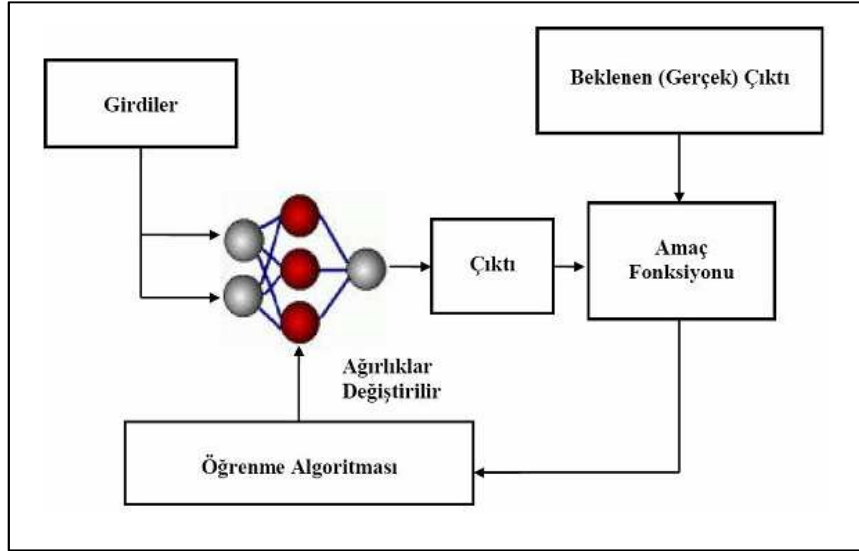
Yapay sinir ağları ile ilgili en büyük problem, yaklaşık ağırlıkları belirlemek ve bu ağırlıkların en iyi konfigürasyonunu elde etme süreci, yani eğitimidir (Radlak, 2008, s. 12). Yapay nöronlar da, biyolojik nöronlar gibi mevcut örnek kümesi üzerinde girdi ve çıktı arasındaki bağıntının ağırlıklarının değiştirilmesiyle eğitilirler. Sunulan girdi kümesi için, transfer fonksiyonu

tarafından sağlanan değerlere cevap olarak bağlantı ağırlıklarının tamamının veya bir kısmının istenen çıktı ile ağ çıktısı arasındaki farkın (hatanın) belirli bir değere düşünceye kadar eğitilmesidir. Bu amaçla günümüze kadar çeşitli öğrenme algoritmaları geliştirilmiştir. Bunlar temel olarak danışmanlı (supervised) öğrenme, danışmansız (unsupervised) öğrenme, takviyeli (reinforcement) öğrenme ve karma öğrenme olarak sınıflandırılabilir (Öcal, 2007, s. 39).

4.1.2.1 Danışmanlı (Supervised) Öğrenme

Bu tip öğrenmede, yapay sinir ağının eğitimi için örnek olarak girdi değerlerinden ve hedef çıktı değerlerinden oluşan bir veri seti verilir. İstenilen çıktı ile gerçek çıktı değeri arasındaki fark hata olarak ele alınır ve bu hatalar minimize edilmeye çalışılır. Bu yüzden de bağlantıların ağırlıkları en uygun çıkışı verecek şekilde değiştirilir. Bu sebeple danışmanlı öğrenme algoritmasının bir “öğretmene” veya “danışmana” ihtiyacı vardır.

Danışmanlı öğrenme algoritmalarında, genellikle hataların hesaplanması için Ortalama Mutlak Hata ve Hata Kareleri Ortalamasının Karekökü gibi performans ölçütleri kullanılır. Bahsi geçen hata sinyallerini minimize etme durumu bu ölçütler yardımıyla yapılır ve ağırlıklar bu minimizasyonu sağlamak amacıyla düzenlenir. Diğer bir ifadeyle, danışmanlı öğrenme yöntemiyle YSA, örnek girdiyi işleyerek kendi çıktısını üretir ve gerçek çıktı ile kıyaslar. Bu öğrenme yönteminde ağırlıkların düzenlenerek sisteme verilmesi yöntemiyle YSA'nın danışmana benzemesi amaçlanır (Haykin, 1999, s. 63). Bu sayede gerçek çıktı değerlerine en yakın çıktı değerleri YSA tarafından elde edilmiş olur. Şekil 4.3'te danışmanlı öğrenme algoritmasının işleyişi gösterilmektedir (Hamzaçebi, 2011, s. 23).



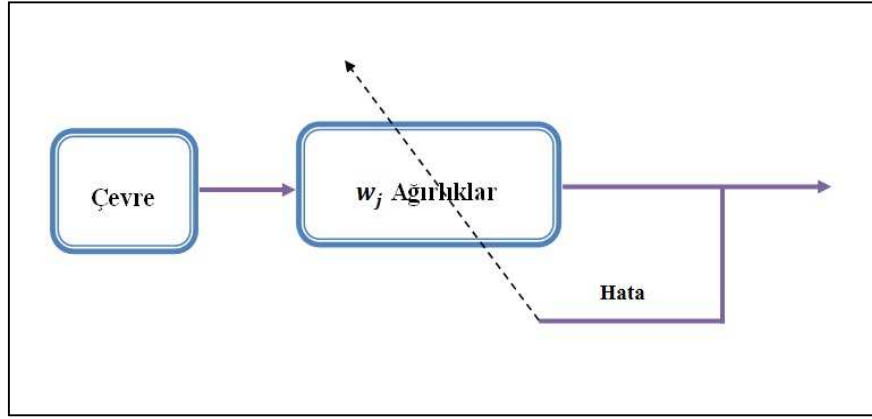
Şekil 4.3: Danışmanlı öğrenme algoritmalarının işleyişi

Giriş ve çıkış bilgilerinin ağı nasıl sunulacağı veya nasıl kodlanacağı hususu, bir ağı başarılı ve verimli bir şekilde yönlendirmek için önemli bir unsurdur. Yapay sinir ağları bilindiği gibi sadece giriş bilgileri ile çalışırlar. Bu yüzden ham bilgiler genellikle ölçeklendirilmeli ve ağı o şekilde sunulmalıdır (Elmas, 2003, s. 96).

Danışmanlı öğrenmede, öğrenme kuralları Perceptron, Delta, Genişletilmiş Delta ve Geri Yayılımlı (Back Propagation) Öğrenme Algoritması olmak üzere dört kısımda incelenir.

4.1.2.2 Danışmansız (Unsupervised) Öğrenme

Danışmansız öğrenmede, öğrenme süresince ağı sunulan bilgiler yalnızca girdi vektörlerinden oluşmakta ve hedef çıkışlar ağı sunulmamaktadır. Bu nedenle ağın üretmiş olduğu çıktıları kıyaslayarak kontrol edebileceği bir danışmanı (öğretmeni) bulunmamaktadır (Şen, 2004, s. 100). Danışmansız olarak eğitilebilen ağlar istenen ya da hedef çıkış olmadan giriş bilgilerinin özelliklerine göre ağırlık değerlerini ayarlar. Danışmansız öğrenmeye, Hebbian öğrenme, Grossberg öğrenme ve Kohonen'in özörgütlemeli harita ağı örnek olarak verilebilir. Şekil 4.4'te danışmansız öğrenme ana hatlarıyla gösterilmiştir.



Şekil 4.4: Danışmansız öğrenme algoritması (Elmas, 2003, s. 149)

Danışmansız öğrenme genellikle sınıflandırma problemlerinin çözümünde kullanılır. Öğrenme bittikten sonra çıktıların ne anlama geldiğini belirtmek için bir kullanıcıya ihtiyaç vardır. Bu yöntem **Kendi Kendine Öğrenme** (Self-Organization) ya da **Adaptasyon** olarak bilinir (Bayır, 2006, s. 27).

4.1.2.3 Takviyeli (Reinforcement) Öğrenme

Takviyeli öğrenme algoritmasında istenilen çıktı değerinin bilinmesine gerek yoktur. Takviyeli (reinforcement) öğrenme yöntemi öğretmenli (danışmanlı) öğrenme algoritmasına benzemekle beraber ağa hedeflenen çıktılar yerine, ağın çıktılarının ne ölçüde doğru olduğunu belirten bir netice veya derece bildirilir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı veya Genetik Algoritmalar (GA) takviyeli öğrenmeye örnek olarak verilebilir (Saraç, 2004, s. 30).

4.1.2.4 Karma Öğrenme

Karma öğrenme metodunda değişik uygulamalarda kullanılacak iki çeşit öğrenme örneği vardır. Bunlardan birincisi çevrimiçi, diğeri ise çevrimdışı ya da öbek öğrenme olarak adlandırılır. Bununla beraber her iki öğrenme tarzını

da birleřtirmek ve deęiřkenleri belli bir eęitim veri giriři yapıldıktan sonra gncelleřtirmek mmkndr.

4.1.2.4.1 evrimii (On-line) ęrenme

Bu kurala gre ęrenen sistemler, gerek zamanda alıřırken bir taraftan fonksiyonlarını yerine getirmekte, dięer taraftan da ęrenme iřlemine devam etmektedir. Eęer deęiřkenler her bir verinin sunulmasından sonra gncelleřtirilirse, evrimii ęrenme rneęi elde edilmiř olur. Bu ęrenme yntemi deęiřen karakteristięe sahip sistemler iin, evrimii deęiřken tanımlanması aısından nemlidir (ztemel, 2012, s. 26).

4.1.2.4.2 evrimdiři (Off-line) ęrenme

Bu kurala dayalı sistemler kullanıma alınmadan nce rnekler zerinde eęitilirler. Bu kuralı kullanan sistemler eęitildikten sonra gerek hayatta kullanıma alındıęında artık ęrenme iřlemi gerekleřmemektedir. Sistemin ęrenmesi gereken yeni bilgiler sz konusu olduęunda sistem kullanımdan ıkarılmakta ve evrimdiři olarak yeniden eęitilmektedir. Eęitim tamamlanınca sistem tekrar kullanıma alınmaktadır (ztemel, 2012, s. 26).

4.1.3 Kullanım Amalarına Gre Sınıflandırılması

Yapay sinir aęlarının kullanım ve uygulama alanları Blm 2.4'te incelenmiřti. Bu blmde ise YSA'nın kullanım amalarına gre hangi topolojilerin kullanıldıęı anlatılacaktır. Kullanım amalarına gre yapay sinir aęları 8 ęede incelenebilir. Tablo 4.1'de YSA'ların kullanım amaları ve uygulanan topolojiler gsterilmiřtir.

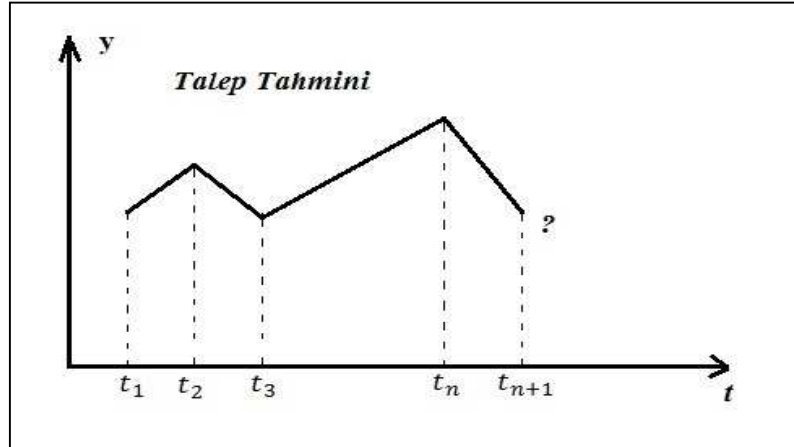
Tablo 4.1: Kullanım amaçlarına göre YSA topolojileri (Anderson and McNeill, 1992, s. 31)

Kullanım Amaçları	Uygulanılan Ağ Topolojileri
Tahmin - Öngörü	<ul style="list-style-type: none">▪ Geri Yayılım▪ Yönlendirilmiş Rassal Tarama (Directed Random Search)▪ Yüksek Dereceli Sinir Ağları▪ Geri Yayılım İçinde SOM▪ Radyal Tabanlı Fonksiyon (RBF)▪ Elman Ağı▪ Jordan Ağı
Fonksiyon Yaklaşırma	<ul style="list-style-type: none">▪ Geri Yayılım▪ RBF
Desen (Örüntü) Sınıflandırma	<ul style="list-style-type: none">▪ LVQ Ağı▪ ART Ağları▪ Olasılık Tabanlı Ağlar▪ Tek veya Çok Katmanlı Algılayıcı▪ Boltzmann Makinesi▪ RBF
Veri İlişkilendirme	<ul style="list-style-type: none">▪ Kohonen's SOM (Self Organizing Map)▪ Hopfield Ağları▪ Boltzmann Makinesi▪ Hamming Ağları▪ Çift Yönlü İlişkili Hafıza▪ Spatio-Temporal Desen Tanıma▪ ART
Kavramlaştırma / Kümeleme	<ul style="list-style-type: none">▪ LVQ▪ SOM▪ ART
Veri Filtreleme	<ul style="list-style-type: none">▪ Yeniden Dolaşım (Recirculation)
Optimizasyon	<ul style="list-style-type: none">▪ Geri Yayılım▪ Olasılık Tabanlı Sinir Ağları▪ RBF
Kontrol	<ul style="list-style-type: none">▪ Geri Yayılım▪ LVQ▪ RBF

Tablo 4.1, mevcut olan tüm ağ topolojilerini değil, en çok kullanılan tiplerini göstermektedir. Tablodan da anlaşılacağı üzere, bazı ağlar muhtelif birçok problemin çözümünde kullanılabilir.

4.1.3.1 Tahmin-Öngörü Yapma

Yapay sinir ağına girdi olarak sunulan elemanların tahmin ya da öngörü çıktısı elde etmek için kullanılmasıdır. Günümüzde YSA'nın en çok kullanılması amaçlarından biri olan tahmin ya da öngöründe bulunma, özellikle finansal çevrelerde çok sık tercih edilmektedir. Borsa endeksinin ve hisse senedi fiyatlarının tahminlemesi, döviz kuru tahmini, bir şirketin belirlenen kriterler dâhilinde iflas edip etmeyeceğini öngörmek ve finansal krizlerin tahmin edilmesi bu yöntemin kullanıldığı alanlardan bazılarıdır. Tahmin amaçlı kullanılan ağ topolojilerine örnek olarak **Geri Yayılım** (Back-Propagation), **Yönlendirilmiş Rassal Tarama** (Directed Random Search), **Yüksek Dereceli Sinir Ağları** (Higher Order Neural Networks) ve **Radyal Tabanlı Fonksiyon** (Radial Basis Function-RBF) verilebilir.

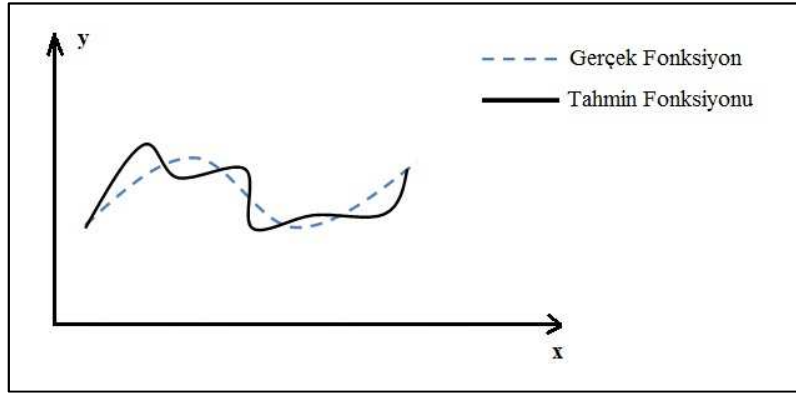


Şekil 4.5: Tahmin işleminin şematik gösterimi

4.1.3.2 Fonksiyon Yaklaşırma

Fonksiyon Yaklaşırma (Function Approximation) işlemi, eldeki birçok veri setini işleyerek veri çiftlerinin oluşturduğu bilinmeyen fonksiyonu tahmin etmeye çalışmaktır. Nonlineer modellerde kullanılan belirli fonksiyonel yapılar, veriyi üreten fonksiyonun genellikle yapay sinir ağlarının ima ettiğiinden farklı

olduğu ve bu nedenle yapay sinir ağlarının kullanılması için gerekli ekonometrik teorinin eksik tanımlı doğrusal olmayan modeller için uygun olduğunu göstermektedir. Bunların ön tanımlı yapıları dayanıksızken, YSA herhangi bir sürekli fonksiyona veya türevlerine yakınsama yeteneğine sahiptir. Bu nedenle **Evrensel Fonksiyon Yakınsayıcı Metot** (Universal Function Approximators) olarak tanımlanmaktadır (Beltratti, Margarita ve Terna, 1996, s. 75)

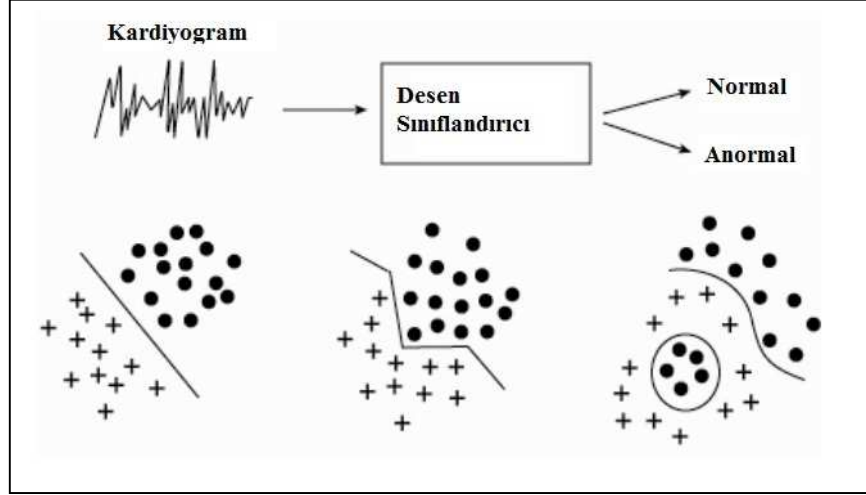


Şekil 4.6: Fonksiyon yaklaşırma işleminin şematik gösterimi

4.1.3.3 Desen (Örüntü) Sınıflandırma

YSA'nın kullanım amaçlarından bir diğeri de Desen / Örüntü Sınıflandırması (Pattern Classification and Recognition) yapmaktır. Ses dalgaları, resim, kamera görüntüsü, el yazısı ya da bazı görsel şekiller gibi desen içeren girdi değerlerini önceden belirlenmiş desenlerle tanımlamaya çalışan ağlardır. Bir video görüntüsünün ya da bir resmin kime / hangi varlığa ait olduğunu anlama, bir makinenin ürettiği ürünlerin hata grafiklerinden elde edilen desenleri inceleyerek makinenin hatalı çalışıp çalışmadığının kontrolü, laboratuvar ve tıbbi görüntüleme cihazlarının ürettiği verilerden yararlanarak hastalık teşhisinde bulunma ve yaklaşan bir nesnenin tespiti gibi durumlarda kullanılmaktadır. Bu amaçla kullanılan ağ topolojilerinden bazıları şunlardır: **LVQ** (Learning Vector Quantization), **ART** (Adaptive Resonance Theory), **Olasılık Tabanlı Sinir**

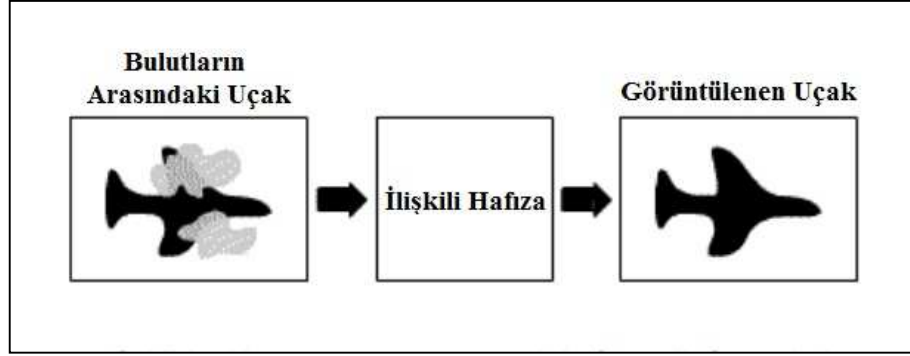
Ağları (Probabilistik Based Neural Networks), Geri **Yayılım**, **Boltzmann Makinesi**, **SOM** ve **RBF Ağları**.



Şekil 4.7: Desen sınıflandırma işleminin şematik gösterimi

4.1.3.4 Veri İlişkilendirme

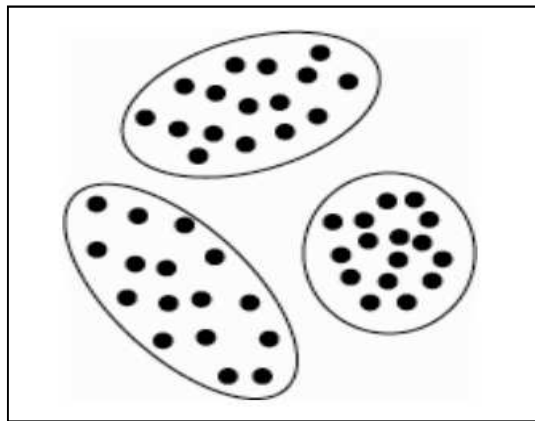
Günümüz bilgisayar teknolojilerinde bilginin içeriğinden ziyade bu bilginin saklandığı yerin adresi ile ilgilenilir. Eğer işlenecek olan bilginin bir kısmında hata varsa veya adresin hesaplanması sağlıklı bir şekilde yapılamazsa doğru bilgiye ulaşmak imkânsızlaşır. **İlişkili Hafızada** (Associative Memory) veya **İçerik Adresli Hafızada** (Content-Addressable Memory) veriye fiziksel adresi ile değil, içerik adresi ile ulaşılır. Hafızadaki içeriğe girdi bilgisi eksik de olsa, bilgi kısmen bozuk da olsa bu bilgi hafızada tamlaştıırılarak çağırılması sağlanır. Bulutların arkasında uçan bir nesnenin tanımlanmasında, farklı açılardan görselleştirilmiş bir kişinin tespitinde ve eksik parçaları bulunan desenlerin tamamlanarak bir şekil oluşturulmasında kullanılmaktadır. **Veri İlişkilendirme** (Data Association) amacıyla kullanılan ağların bazıları şunlardır: **SOM**, **Hopfield Ağları**, **Boltzmann Makinesi**, **Çift Yönlü İlişkili Hafıza** (Bidirectional Associative Memory) ve **ART Ağları**.



Şekil 4.8: Veri ilişkilendirme işleminin şematik gösterimi

4.1.3.5 Veri Kavramlaştırma / Kümeleme

Veri kavramlaştırma / Kümeleme işleminde eğitim için etiketlenmiş veya sınıflandırılmış veri setine ihtiyaç duyulmaz. Bununla beraber veri setindeki benzer desenleri inceleyerek benzerliklerine göre veri setini olabilecek en iyi şekilde kümelemeye çalışırlar. **Veri Madenciliği** ve **Veri Sıkıştırma** konularında sıkça kullanılır. **LVQ**, **Kohonen's SOM** ve **ART Ağları** bu amaçla kullanılan ağlardandır.



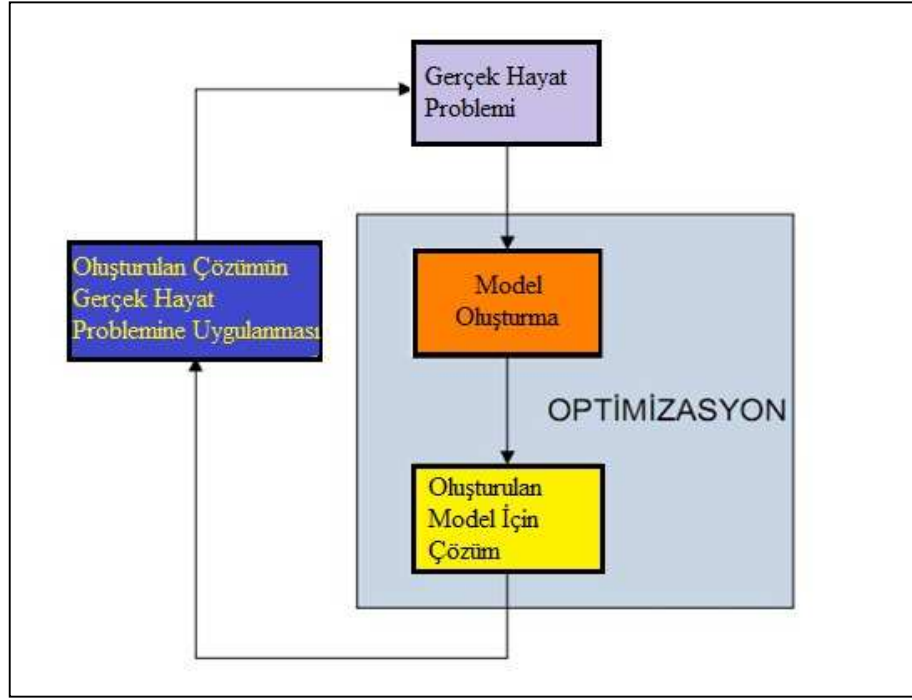
Şekil 4.9: Kümeleme işleminin şematik gösterimi

4.1.3.6 Veri Filtreleme

Yapay sinir ağlarının kullanıldığı en önemli noktalardan biri Veri Filtreleme (Data Filtering)'dir. Girdi sinyallerinin düzgünleştirilmesinde kullanılmaktadır. İlk ağ çeşitlerinden biri olan **MADALINE** ağı bu kategoride kullanılan bir ağıdır. Bununla beraber **Yeniden Dolaşım** (Recirculation) Ağları da bu amaçla kullanılmaktadır.

4.1.3.7 Optimizasyon

Matematik, istatistik, bilgisayar, mühendislik, tıp ve ekonomi gibi birçok bilim dalında kullanılan optimizasyon, en genel tanımıyla verilen kısıtlar altında en iyi çözümün bulunması problemidir. Optimizasyon problemlerinde amaç, verilen kısıtları göz önünde bulundurarak maksimizasyon veya minimizasyon yapmak, optimum sonuçları elde etmektir. Gerçek hayat problemlerinin çözümleri için bir karar verme sürecine ihtiyaç vardır. Bu süreçte düşünülen çözümlerin kısıtlı olması veya çözüm arayanın bakış açısının yeterince geniş olamaması nedeniyle önerilen veya gerçekleştirilen çözümler ancak uygun çözüm veya olası çözüm olarak nitelendirilebilir. **Geri Yayılım, Olasılık Tabanlı Sinir Ağları ve RBF Ağları** optimizasyon amaçlı kullanılan ağlardan bazılarıdır.



Şekil 4.10: Gerçek hayat problemlerine sistematik çözüm yaklaşımı

4.1.3.8 Kontrol

Dinamik kontrol gerektiren uygulamalar, kontrol girdisi ile çıktısı arasındaki ilişkiyi eş zamanlı olarak inceleyip gerekli tepkileri anında verecek şekilde tasarlanırlar. Referans modeline uygun olarak istenilen çıktıların üretilmesi için kontrol girdilerinin değiştirilmesi gerekir. Bu prensibi temel alarak çalışırlar. Otomobillerdeki hız kontrolü ve virajlarda savrulmayı önleyen sistemler bunlara örnek gösterilebilir. **LVQ, Geri Yayılım ve ART Ağları** bu amaçla kullanılan ağlardır (Bayır, 2006, s. 38-42; Jain, Mao and Mohiuddin, 1996, s. 31).

4.2 Temel Öğrenme Kuralları

Yapay sinir ağlarında mimari yapının oluşturulmasının yanı sıra, ağın öğrenmesinde / eğitilmesinde kullanılacak olan algoritma ve yöntemlerin de doğru olarak belirlenmesi oldukça önemlidir. Ağırlıkların değiştirilmesi bu yöntemlere göre yapılmaktadır. Bu algoritmalar, yapay sinir ağlarının topolojisine ve

karşılaşılan problemin türüne göre farklılık gösterse de birçoğunun temel olarak aldığı Hebb kuralı ve onun geliştirilmiş versiyonlarıdır. Bununla birlikte en çok kullanılan öğrenme kuralları Hebb, Hopfield, Delta, Dereceli Azaltma ve Kohonen Öğrenme Kurallarıdır.

4.2.1 Hebb Kuralı

Donald Hebb tarafından 1949 yılında geliştirilen ve temelinde biyolojik öğrenme yer alan en eski öğrenme kuralıdır. Diğer öğrenme kurallarının temelini de oluşturan Hebb Kuralı'na göre bir yapay sinir hücresi diğer bir hücreden bilgi alırsa ve her iki hücre de aktif ise (sayısal olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir (Anderson ve McNeill, 1992, s. 29). Diğer bir deyişle bu kural şu şekilde özetlenebilir: Bir hücre, kendisi aktif ise bağlı olduğu hücreyi aktif yapmaya, pasif ise pasif yapmaya çalışır (Öztemel, 2012, s. 26).

Hebb Kuralı'nın çalışma yöntemini bir örnekle izah etmek gerekirse; eğer A hücresi B hücreyi uyarmaya yetecek kadar yakın ise ve B hücrenin aktifleştirilmesinde sürekli olarak yer alıyorsa, A hücrenin etkinliği arttırılacak şekilde bir hücrenin ya da her ikisinin ağırlıklarında değişiklikler yapılır (Hebb, 1949, s. 50).

4.2.2 Hopfield Kuralı

Hopfield kuralı da genel olarak Hebb kuralına benzemektedir. Yapay sinir ağı elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi ya da zayıflatılması gerektiğini belirlemek amacıyla kullanılır. Eğer beklenen çıktı ve girdiler ikisi de aktif / pasif ise ağırlık değerlerini öğrenme katsayısı kadar kuvvetlendir / zayıflatılmaktadır. Yani, ağırlıkların kuvvetlendirilmesi ya da zayıflatılması öğrenme katsayısı yardımıyla gerçekleştirilir. Öğrenme katsayısı ise genel olarak 0-1 arasında olan ve kullanıcı tarafından atanan sabit ve pozitif bir değerdir.

4.2.3 Delta Kuralı

Bu kural Hebb kuralının biraz daha gelişmiş versiyonudur. Bu kuralın çalışma ilkesi, beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak amacıyla yapay sinir ağı elemanlarının bağlantı ağırlık değerlerinin sürekli değiştirilmesidir. Bu yüzden de ağı ürettiği çıktı ile üretilmesi gereken (beklenen) çıktı arasındaki hatanın karelerinin ortalamasını minimize etmek hedeflenmektedir. Eğer hata sıfır ise ağırlıklar değiştirilmeden eğitim işlemi sonlandırılır. Hata pozitif ise, kabul edilebilir hale getirilinceye kadar ağırlıklarda değişiklik yapılır.

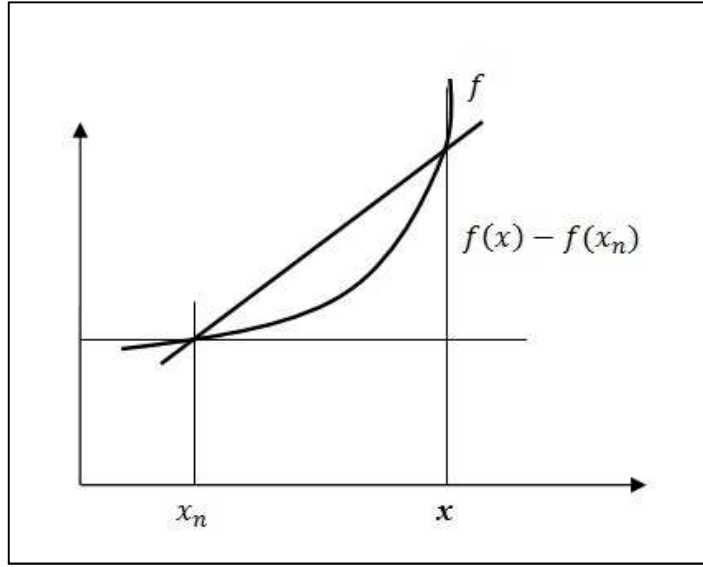
Delta kuralının çalışma sistemi şu şekildedir: Çıktı katmanındaki delta hatası transfer fonksiyonunun türevi ile dönüştürülerek (Transformation) bir önceki katmana iletilir. Bu iletim sonucunda bir önceki katmanın girdi bağlantılarının katsayısı değiştirilir. Yani hata her seferinde bir önceki katmana geri yayılır. Bu geri yayılım işlemi ilk katmana ulaşmaya kadar devam eder. Bu kuralı kullanan ağlara Geri Yayılım Ağları denir.

Delta kuralı kullanılırken dikkat edilmesi gereken en önemli konu, girdi setindeki verilerin rastgele dağılmış olmasıdır. Eğer eğitim setindeki veriler düzgün dağılmış ise istenilen doğruluğa ulaşmak zorlaşacak ve ağı öğrenme işlemini gerçekleştirmesine engel olacaktır (Bayır, 2006, s. 31).

4.2.4 Eğimli İniş Kuralı

Gizli katman olmayan bir ağda hata yüzeyinde tek bir en iyi nokta bulunmaktadır. Gizli katmanın varlığı durumunda ise birden fazla iyi nokta olduğundan en iyi noktayı bulmak zorlaşmaktadır. Bu durumda en iyi değer elde edilinceye kadar eğimli iniş yöntemi ile ağırlıklar değiştirilir.

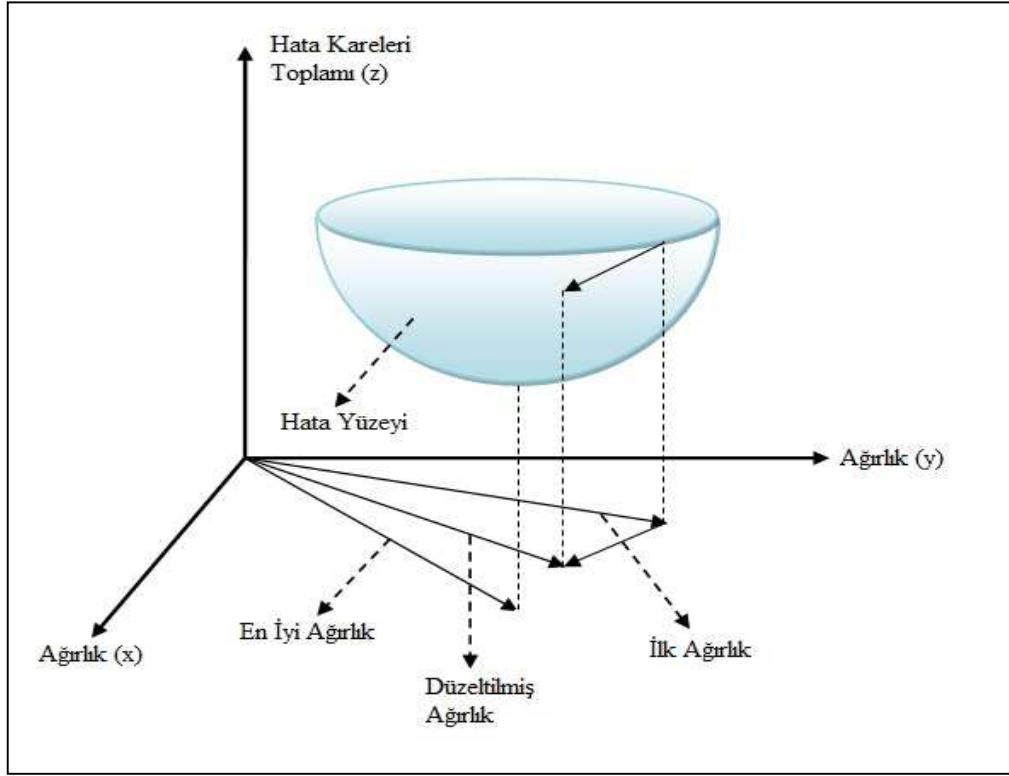
Eğimli iniş yönteminde, ağı çıkış değerinin her bir ağırlık değerine göre kısmi türevi alınır. Bu yöntemde elde edilen değer bir önceki değerden her zaman daha küçük olacaktır.



Şekil 4.11: Bir fonksiyon ve türevinin geometrik yorumu

Hata değişimi ile ağırlık vektörü arasında değişim eğrisi Şekil 4.11’de gösterilmiştir.

Hata düzelterek öğrenme işleminde her döngü bu paraboldeki iniş gerçekleştirmektedir. Eğer gizli katman var ise yerel minimum değere iniş gerçekleşir. Bunu görebilmek için ağırlıkların mümkün olan tüm değerleri, hataların karelerine karşılık gelecek şekilde xyz -koordinat düzleminde çizilir. Bu çizim bize hata yüzeyini verir. Bu yüzey Şekil 4.12’de görüldüğü gibi bir tasa benzemektedir. Tasın en alt kısmı hataların kareleri toplamının en küçüklerine karşılık gelmektedir. Ağı eğitme esnasında amaç, ağırlıklar kümesinin en iyisini yani tasın en alt kısmını bulmaktır. Geri yayılım algoritması, o andaki ağırlıkların yerine, hata yüzeyinin eğimini hesaplayarak amacına ulaşır. Daha sonra da bu ağırlıkları tasın alt kısmına doğru artımsal olarak değiştirir. İşte bu artımsal olarak tasın üst kısmından alt kısmına doğru ilerleme işlemine **eğimli iniş** denir.



Şekil 4.12: Hata değişimi ve ağırlık vektörlerinin geometrik yorumu

n adet giriş ve m adet çıkış düğümü olan tek katmanlı bir ağda her bir i -inci düğümün çıkışı şu şekilde ifade edilebilir:

$$y_i = a\check{g}_i = (w_i, x) = w_{i1}x_1 + \dots + w_{in}x_n, \quad i = 1, \dots, m \quad (4.1)$$

Eğitim kümesi ise

$$\{(x^1, y^1), \dots, (x^K, y^K)\} \quad (4.2)$$

şeklinde ifade edilir. K eğitim çifti sayısını göstermektedir. Bununla beraber tek katmanlı ileri beslemeli bir ağda toplam ağ hatası

$$E = \frac{1}{2} \sum_{k=1}^K \sum_{i=1}^m (v_i^k - z_i^k)^2 \quad (4.3)$$

olarak tanımlanır. Burada i çıkış düğümünün sırasını, k eğitim çifti sırasını, v_i^k da i -inci çıkış düğümünün k -inci istenen eğitim çıkışını gösterir. z_i^k ise i -inci çıkış

değerinin k -inci eğitim çıkışına göre hesaplanan değerini gösterir ve şu şekilde ifade edilir:

$$y_i(x^k) = (w_i, x^k) \quad (4.4)$$

E sistemin toplam hata değerini göstermektedir. Burada ağırlıkların değişimi için eğimli iniş yöntemi kullanıldığında her bir ağırlık sistem hatasının kendisine göre kısmi türevi alınarak güncelleştirilir. Genel olarak bu durum şu eşitlikle ifade edilir:

$$w_{ij} = w_{ij} + \eta \frac{\partial E_k}{\partial w_{ij}} \quad (4.5)$$

Burada η öğrenme sabiti olup, $\eta > 0$ olarak alınır. Ağırlık değerine göre kısmi türev şu şekilde alınır:

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial a\check{g}_i} \frac{\partial a\check{g}_i}{\partial w_{ij}} = -(v_i - z_i)x_i \quad (4.6)$$

Burada

$$a\check{g}_i = w_{i1}x_1 + \dots + w_{in}x_n, \quad i = 1, \dots, m \quad (4.7)$$

olarak gösterilir. Böylece yeni ağırlık değeri

$$w_{ij} = w_{ij} + \eta (v_i - z_i)x_i, \quad i = 1, \dots, n \quad (4.8)$$

eşitliği ile hesaplanır. Sistemin hata değeri δ_i^k ile gösterilirse

$$\delta_i^k = \frac{\partial E_k}{\partial a\check{g}_i} = (v_i - z_i) \quad (4.9)$$

olur. Ağırlık değeri yeniden yazılırsa,

$$w_{ij} = w_{ij} + \eta \delta_i^k x_i \quad (4.10)$$

şeklinde kısaca yazılabilir. Bu öğrenme kuralında şu adımlar izlenmektedir:

Adım 1: $\eta > 0$ olmak üzere η değeri ve $E_{max} > 0$ olarak E_{max} değeri belirlenir.

Adım 2: Tüm ağırlıklara rastgele küçük değerler atanır. $k = 1$ ve başlangıç hata değeri $E = 0$ alınır.

Adım 3: Sistem çıkışı hesaplanır.

$$y_i(x) = (w_i, x) = w_i^T x, \quad i = 1, \dots, m \quad (4.11)$$

Adım 4: Ağırlıklar güncellenir.

$$w_{ij} = w_{ij} + \eta(v_i - z_i)x_j, \quad i = 1, \dots, n \quad (4.12)$$

Adım 5: Toplam hata değeriyle, şuna kadarki hata değerlerinin kümülatif toplamı bulunur.

$$E = E + \frac{1}{2}(v_i - z_i)^2, \quad i = 1, \dots, n \quad (4.13)$$

Adım 6: Eğer $k < K$ ise $k = k + 1$ yapılır ve Adım 3'e gidilir.

Adım 7: Bu işlemler sonucunda bir döngülük eğitim işlemi yapılmış olur. $E < E_{max}$ ise ağırlıklar sabitlenir ve eğitim bitirilir. $E > E_{max}$ ise $E = 0$ yapılır ve yeniden eğitime başlamak için Adım 3'e gidilir (Elmas, 2003, s.107-111).

4.2.5 Kohonen Kuralı

Teuvo Kohonen (1982) tarafından geliştirilen bu öğrenme kuralında, biyolojik nöron sistemlerinde gerçekleşen öğrenmeden esinlenilmiştir. **Yarışmacı Öğrenme Kuralı (Competitive)** da denilen bu kuralda işlemci elemanlar, ağırlıklarının ayarlanması için yarışmaktadırlar. Kazanan işlemci elemanın bağlantı ağırlıkları güncellenmektedir. En uygun çıktıya sahip işlemci elemanın kazandığı kuralda işlemci eleman, kendisine komşu olan işlemci elemanların ağırlıklarının değiştirilmesine de izin vermektedir. Bu özelliğinden dolayı bu kural "kazanan tamamını alır" kuralı olarak da bilinir.

4.3 Yapay Sinir Ağı Modelleri

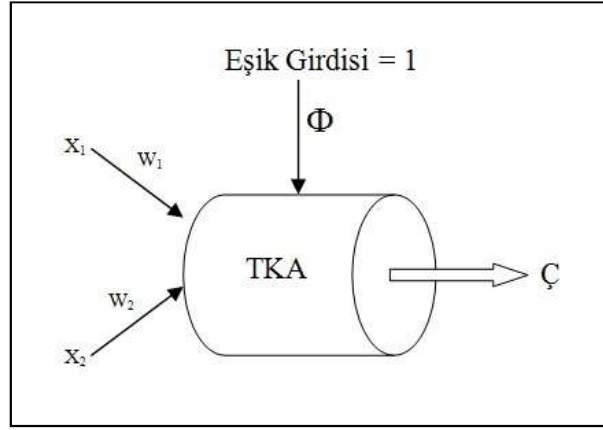
Bu bölümde ilk yapay sinir ağı çalışmaları ve geliştirilen ilk modeller açıklanacaktır. Bu kapsamda tek katmanlı algılayıcılar, Perseptron ve Adaline / Madaline modellerinin yapısı ve öğrenme kurallarına değinilecektir.

4.3.1 Tek Katmanlı Yapay Sinir Ağları

Yapay sinir ağları ile ilgili çalışmalar Tek Katmanlı Algılayıcılar (TKA - Single Layer Perceptron) ile başlamıştır. Bu algılayıcıların en önemli özelliği problem uzayını bir doğru veya düzlem ile sınıflara ayırmalarıdır. Problemin girdileri ağırlıklar ile çarpılıp toplandıktan sonra elde edilen bir eşik değerinden büyük veya küçük olmasına göre girdinin sınıfı belirlenir. Sınıflar 1 veya -1 rakamları (bazen 1 ve 0 rakamları) ile gösterilir. Öğrenme sırasında hem ağın ağırlıkları hem de eşik değer ünitesinin ağırlık değeri değiştirilir. Eşik değer ünitesinin girdisi sabit olup 1'dir (Öztemel, 2012, s. 59).

Tek katmanlı algılayıcıların en önemli problemi doğrusal olmayan olayları öğrenememeleridir. O nedenle TKA geliştirilmiş ve yeni modeller oluşturulmuştur. Tek katmanlı yapay sinir ağları sadece girdi ve çıktı katmanlarından oluşur. Her ağın bir veya daha fazla girdisi ve çıktısı (\mathcal{C}) vardır. Çıktı üniteleri bütün girdi ünitelerine (X) bağlanmaktadır. Her bağlantının bir ağırlığı vardır (W). Bu ağlarda proses elemanlarının değerlerinin ve dolayısıyla ağın çıktısının sıfır olmasını önleyen bir de eşik değeri (Φ) vardır. Eşik değerinin girdisi daima 1'dir.

En basit şekliyle aşağıda iki girdi ve bir çıktıdan oluşan en basit TKA modeli verilmiştir:



Şekil 4.13: İki girdi ve bir çıktıdan oluşan basit TKA modeli

Ağın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu bulunur. Bu girdi değeri bir aktivasyon fonksiyonundan geçirilerek ağın çıktısı hesaplanır. Bu çıktıyı şu şekilde formülize etmek mümkündür:

$$Ç = f \left(\sum_{i=1}^m w_i x_i + \Phi \right) \quad (4.14)$$

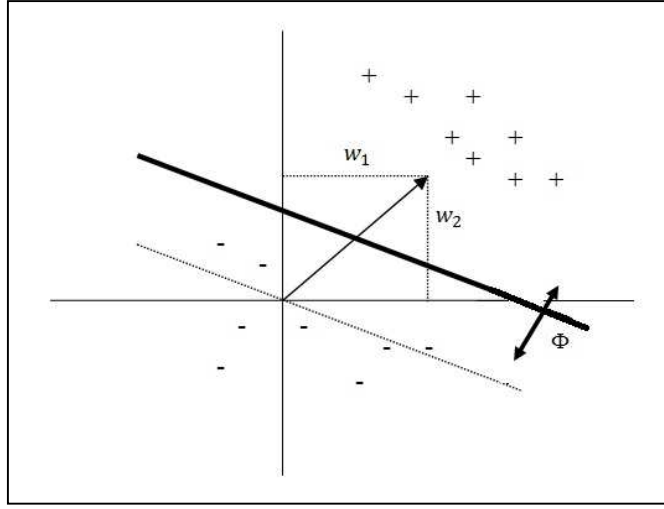
Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal fonksiyondur. Yani ağa gösterilen örnekler iki sınıf arasında paylaştırılarak iki sınıfı birbirinden ayıran doğru bulunmaya çalışılır. Onun için eşik değer fonksiyonu kullanılmaktadır. Burada ağın çıktısı 1 veya -1 değerlerini almaktadır. 1 ve -1 sınıfları temsil etmektedir. Eğer ağın çıktısı 1 ise birinci sınıfta, -1 ise ikinci sınıfta kabul edilmektedir (Bazı araştırmacılar bu sınıfları 1 veya 0 olarak da göstermektedir – mantıkta bir değişme yoktur).

$$f(g) = \begin{cases} +1 & \text{eğer } Ç > 0 \text{ ise} \\ -1 & \text{aksi halde} \end{cases} \quad (4.15)$$

Bu formül incelendiğinde ağa gelen toplam girdinin pozitif olması durumunda ağa sunulan örnek 1. sınıfta, negatif olması durumunda ise -1 sınıfına ait demektir. Sıfır olması durumu ise tasarımcının kabulüne kalmıştır (yukarıdaki formülde -1 sınıfına konulmuştur). Dikkat edilirse iki sınıfı ayıran bir doğrudur. Sınıf ayırıcı da denilen bu doğru şu şekilde tanımlanmaktadır:

$$x_1 w_1 + x_2 w_2 + \Phi = 0 \quad (4.16)$$

ve buradan $x_1 = -(w_2 / w_1)x_2 - \Phi / w_1$ ve benzer şekilde $x_2 = -(w_1 / w_2)x_1 - \Phi / w_2$ olarak hesaplanır. Bu iki formülden hareketle sınıf ayırıcı doğrusu çizilebilir. Ağırlıkların ve sınıf ayırıcı olan doğrunun geometrik gösterimi aşağıdaki gibidir (Öztemel, 2012, s. 60):



Şekil 4.14: Ağırlıkların ve sınıf ayırıcı olan doğrunun geometrik yorumu

Bilinen en önemli tek katmanlı algılayıcılar şunlardır:

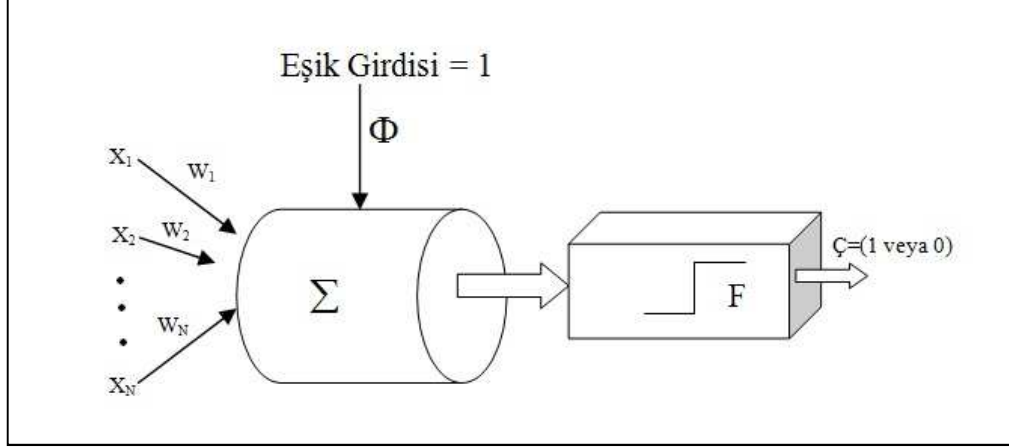
- Basit Tek Katmanlı Algılayıcılar (Perceptronlar)
- Adaline / Madaline Üniteleri

4.3.1.1 Perceptron Modeli

Bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanan perceptron, Rosenblatt tarafından 1959 yılında bulunmuştur. Rosenblatt (1961) *Principles of Neurodynamics* isimli çalışmasında basitleştirilmiş ağ olarak tanımladığı perceptronların, biyolojik sistemlerin ve daha geniş ağların bir parçasına karşılık geldiğini belirtmiştir. Bu sebeple sonuçları direkt

uygulanabilmektedir. Ağın çıktısı bir veya sıfırdan oluşan mantıksal (boolean) değerdir. Çıktının değerinin hesaplanmasında eşik değer fonksiyonu kullanılır.

Perseptron'un yapısı Şekil 4.15'te gösterildiği gibidir:



Şekil 4.15: Bir basit algılayıcı yapısı

4.3.1.2 ADALINE / MADALINE Modeli

ADALINE Modeli, Widrow ve Hoff tarafından 1959 yılında geliştirilmiş olup Adaptif Doğrusal Eleman (ADaptif LINear Element) ağının kısaltılmış şeklidir. Genel olarak ADALINE bir proses elemanından (Adaline ünitesi) oluşan bir ağıdır.

Bu ağ en küçük ortalamaların karesi (least mean square) yöntemine dayanmaktadır. Öğrenme kuralına Delta Kuralı da denmektedir. Öğrenme kuralı, ağın çıktısının beklenen çıktı değerine göre hatasını minimize edecek şekilde ağın ağırlıklarının değiştirilmesi prensibine dayanır. ADALINE'in yapısı tek katmanlı algılayıcıya benzemektedir, aradaki fark ise öğrenme kuralındadır.

ADALINE ağının çıktı değerini üreten aktivasyon fonksiyonu adım fonksiyonudur. Çıktı fonksiyonunun değeri 0'dan küçük ise çıktı -1, büyük ise +1 değerini alır. ADALINE modelinde herhangi bir t anında hatayı azaltmak için kullanılan kural, $w_i(t) = w_i(t - 1) + \eta E x_i$ eşitliği ile gösterilir. Burada $w_i(t)$

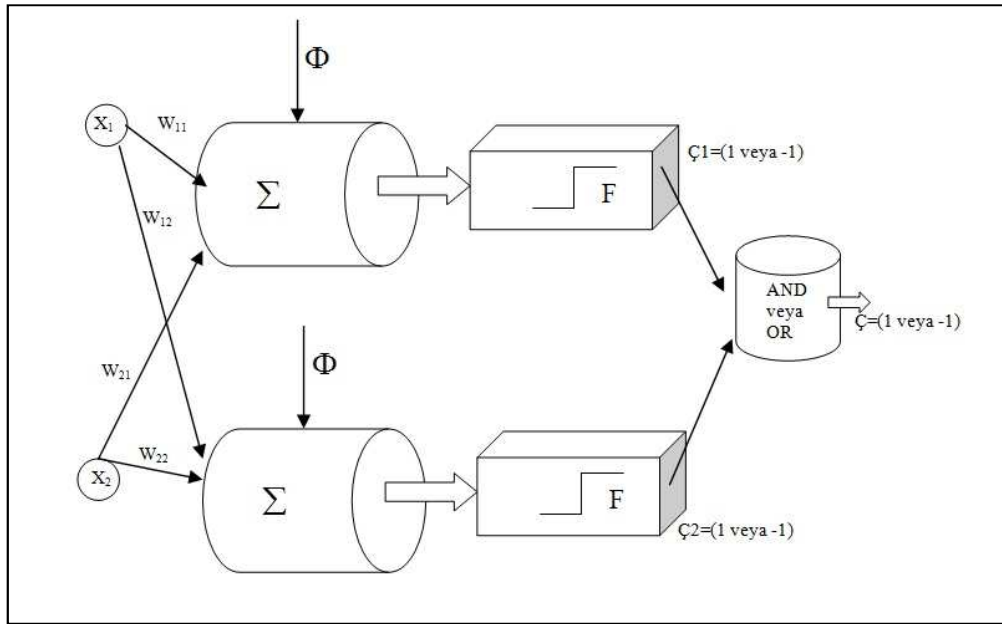
ağırlığın t zamanındaki yeni değerini, $w_i(t-1)$ ağırlığın değişmeden önceki değerini, η öğrenme katsayısını, E beklenen değer ile çıktı arasındaki hatayı ve x_i de girdi değerini göstermektedir.

Benzer şekilde Φ eşik değeri de yine zaman içerisinde değiştirilerek olması gereken eşik değeri hesaplanmaktadır. Bunun matematiksel ifadesi ise,

$$\Phi_i(t) = \Phi_i(t-1) + \eta E \quad (4.17)$$

şeklindedir (Baş, 2006, s. 46).

MADALINE ağları ise birden fazla ADALINE ünitesinin bir araya gelerek oluşturdukları ağa verilen isimdir. MADALINE ağları genel olarak iki katmandan oluşur. Her katmanda değişik sayıda ADALINE ünitesi bulunmaktadır. Ağın çıktısı da yine 1 ve -1 değerleri ile gösterilmektedir. Her biri bir sınıfı temsil etmektedir. Aşağıdaki şekilde iki ADALINE ünitesinden oluşan bir MADALINE gösterilmiştir (Baş, 2006, s. 46):



Şekil 4.16: İki ADALINE ağından oluşan bir MADALINE ağı

4.3.2 Çok Katmanlı Algılayıcılar (Multi Layer Perceptron - MLP)

Bir yapay sinir ağından çözmesi istenilen problem doğrusal değilse, ilk başta tasarlanmış yapay sinir ağları ile bunu modellemek mümkün değildir. Bir önceki bölümde anlatılan yapay sinir ağlarının ilk modellerinin en temel özellikleri doğrusal olan olayları çözebilme yeteneklerine sahip olmalarıdır. Bu ağlar ile doğrusal olmayan ilişkiler öğrenilememektedir. Bu tür problemler için daha gelişmiş bir ağ türüne ihtiyaç duyulmuştur. Çok katmanlı algılayıcılar (MLP) da bu amaçla geliştirilen bir ağdır. Peki, problemlerin doğrusal olup olmaması ne demektir? Bu konuyu iyi anlamak için ünlü XOR problemine bakmak gerekir. Bu problemin özelliği doğrusal olmayan ilişkileri göstermesidir. Yani çıktıların arasında bir doğru veya doğrular çizerek onları iki veya daha fazla sınıfa ayırmak mümkün değildir. Bu problem Tablo 4.2’de gösterildiği gibidir. Perceptron, ADALINE ve MADALINE ile bu problemi çözmek mümkün olmamıştır (Öztemel, 2012, s. 75). XOR probleminin çözümü Ek-A’da verilmiştir.

Tablo 4.2: XOR problemi

Girdi_1	Girdi_2	Çıktı
0	0	0
0	1	1
1	0	1
1	1	0

Birçok öğrenme algoritmasının bu ağı eğitmede kullanılabilir olması bu modelin yaygın kullanıldığına işarettir. Bir MLP modeli, bir giriş, bir veya daha fazla ara katman ve bir de çıkış katmanından oluşur. Her bir katmanda da bir veya daha fazla sayıda işlem elemanı bulunur. Bir katmandaki bütün işlem elemanları bir üst katmandaki işlem elemanlarının tamamına bağlıdır. Bilgi akışı sürekli ileri doğru olup geri besleme yoktur. Bu yüzden bu tip ağlar ileri beslemeli sinir ağı modeli diye anılmaktadır. Aynı zamanda MLP ağlarında giriş katmanında herhangi bir bilgi işleme yapılmaz. Giriş ve çıkış katmanındaki işlemci elemanı sayısı tamamen uygulanan problemin tipine bağlıdır. Buna ilaveten ara katman sayısı ve ara katmanlardaki işlemci elemanı sayısı ise deneme-yanılma yolu ile bulunur (Sağiroğlu vd, 2004, s.54-55).

4.3.2.1 ÇKA Ağının Öğrenme Kuralı

ÇKA ağları öğretmenli öğrenme stratejisine göre çalışırlar. Yani, bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi gereken (beklenen) çıktılar gösterilir. Ağın görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir. ÇKA ağının öğrenme kuralı en küçük kareler yöntemine dayalı Delta öğrenme kuralının genelleştirilmiş halidir. O nedenle öğrenme kuralına **Genelleştirilmiş Delta Kuralı** da denmektedir. Ağın öğrenebilmesi için eğitim seti adı verilen ve örneklerden oluşan bir sete ihtiyaç vardır. Bu set içinde her örnek için ağın hem girdiler hem de o girdiler için ağın üretmesi gereken çıktılar belirlenmiştir. Genelleştirilmiş Delta Kuralı iki safhadan oluşur (Öztemel, 2012, s.77):

1. Safha - İleri Doğru Hesaplama: Ağın çıktısını hesaplama safhasıdır.
2. Safha - Geriye Doğru Hesaplama: Ağırlıkları değiştirme safhasıdır.

1. Safha:

Bu adımda bilgi işleme, eğitim setindeki bir örneğin girdi katmanından (I_1, I_2, \dots) ağa gösterilmesi ile başlar. Daha önce belirtildiği gibi, girdi katmanında herhangi bir bilgi işleme olmaz. Gelen girdiler hiçbir değişiklik olmadan ara katmana gönderilir. Yani girdi katmanındaki t . proses elemanının çıktısı y_t şu şekilde belirlenir:

I_t : Girdi katmanındaki t . proses elemanına gelen girdi.

$y_t = I_t$: t . proses elemanının çıktısı.

Ara katmandaki her proses elemanı girdi katmanındaki bütün proses elemanlarından gelen bilgileri bağlantı ağırlıkları (W) ile çarpılmış olarak alır. Önce ara katmandaki proses elemanlarına gelen net girdi (u_j) şu formül kullanılarak hesaplanır:

$$u_j = \sum_{t=1}^N w_{tj} y_t \quad (4.18)$$

u_j : j . ara katmandaki proses elemanlarına gelen net girdi.

w_{ij} : t . girdi katmanı proses elemanını, j . ara katman proses elemanına bağlayan bağlantının ağırlık değeri.

j . ara katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonundan (genellikle sigmoid fonksiyondan) geçirilmesiyle hesaplanır. Burada önemli olan türevi alınabilir bir fonksiyon kullanmaktır. Çünkü geriye doğru hesaplamada burada kullanılan fonksiyonun türevi alınacaktır. Sigmoid fonksiyon kullanılması halinde çıktı değeri:

$$y_j = \frac{1}{1 + e^{-(u_j + b_j)}} \quad (4.19)$$

şeklinde olacaktır. Burada b_j , ara katmanda bulunan j . elemana bağlanan eşik değer elemanının ağırlığını göstermektedir. Bu eşik değer ünitesinin çıktısı sabit olup 1'e eşittir. Ağırlık değeri ise sigmoid fonksiyonun oryantasyonunu belirlemek üzere konulmuştur. Eğitim esnasında ağ bu değeri kendisi belirlemektedir.

Ara katmanın bütün proses elemanları ve çıktı katmanının proses elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve aktivasyon fonksiyonundan geçirilmesi sonucu belirlenir. Çıktı katmanından çıkan değerler, yani ağın çıktıları bulununca ağın ileri hesaplama işlemi tamamlanmış olur.

2. Safha:

Ağa sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir. O nedenle geriye doğru hesaplamada bu hata ağın ağırlık değerlerine dağıtılarak bir sonraki iterasyonda hatanın azaltılması sağlanır. Çıktı katmanındaki k . proses elemanı için oluşan hata e_k ise:

$$e_k = E[y_k] - y_k \quad (4.20)$$

$E[y_k]$: çıktı katmanındaki k . proses elemanının beklenen çıktı değeri

Bu bir proses elemanı için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı E ile gösterirsek, E 'yi hesaplamak için bütün hataların toplanması gerekir. Bazı hata değerleri pozitif, bazıları ise negatif olacağından, pozitif ve negatiflerin birbirini götürmesini ve toplam hatanın sıfır olmasını engellemek için ağırlıkların kareleri hesaplanarak sonrasında düzeltme işlemi amacıyla karekökü alınır. ÇKA ağının eğitilmesindeki amaç bu hatayı minimize etmektir. Toplam hata:

$$E = \frac{1}{2} \sum_k e_k^2 \quad (4.21)$$

Toplam hatayı minimize etmek için bu hatanın kendisine neden olan proses elemanlarına dağıtılması gerekmektedir. Bu ise proses elemanlarının ağırlıklarının değiştirmek demektir. Ağın ağırlıklarını değiştirmek için iki durum söz konusudur (Öztemel, 2012, s.79):

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

k . çıktı katmanındaki bir proses elemanına ait hata e_k ise ve k . çıktı ünitesinin ağırlıklara dağıtılacak hata olan (δ_k), hata değeri ile çıktının türevinin çarpılması yoluyla hesaplanır:

$$\delta_k = f'(u)e_k \quad (4.22)$$

Örnek olarak, eğer δ sigmoid fonksiyon için hesaplanırsa, k . proses elemanı:

$$\delta_k = y_k(1 - y_k)e_k \quad (4.23)$$

$y_k(1 - y_k)$: sigmoid fonksiyonun türevi

Eğer i . ara katmanla, k . çıktı katmanı arasındaki ağırlıkların değişim miktarı hesaplanıyorsa δ :

$$\delta_i = f'(u) \sum_k \delta_k w_{ik} \quad (4.24)$$

Fonksiyon sigmoid fonksiyonsa:

$$\delta_i = f'(u) \sum_k \delta_k w_{ik} = y_i(1 - y_i) \sum_k \delta_k w_{ik} \quad (4.25)$$

Herhangi bir t anında ya da iterasyonunda ağırlıklardaki değişim miktarı:

$$\Delta w_{ik}(t) = \eta \delta_k y_i + \alpha \Delta w_{ik}(t-1) \quad (4.26)$$

$\Delta w_{ik}(t)$: Herhangi bir t anında ara katmandaki i . proses elemanı ile k . çıktı katmanındaki proses elemanı arasındaki ağırlıklarda gerçekleşecek değişim miktarıdır.

η : öğrenme oranı

α : momentum katsayısı (momentum katsayısı yapay sinir ağının yerel bir optimum noktaya takılmasını engeller, bir önceki değişimin ($t-1$ anı) belli bir miktarı α katsayısı yardımıyla yeni değişime eklenir.

Değişim miktarlarını hesapladıktan sonra ağırlıkların t . iterasyondaki yeni değerleri

$$w_{ik}(t) = w_{ik}(t-1) + \Delta w_{ik}(t) \quad (4.27)$$

Diğer katmanlar içinde ağırlıklar benzer şekilde hesaplanır. Eşik değerlerin ağırlıklarının da benzer biçimde değiştirilmesi gerekmektedir. Bu ünitelerin çıktı değerleri sabit ve 1'e eşittir. Dolayısıyla çıktı katmanındaki eşik değer(ler)in ağırlığının değişimi şu şekilde yapılır:

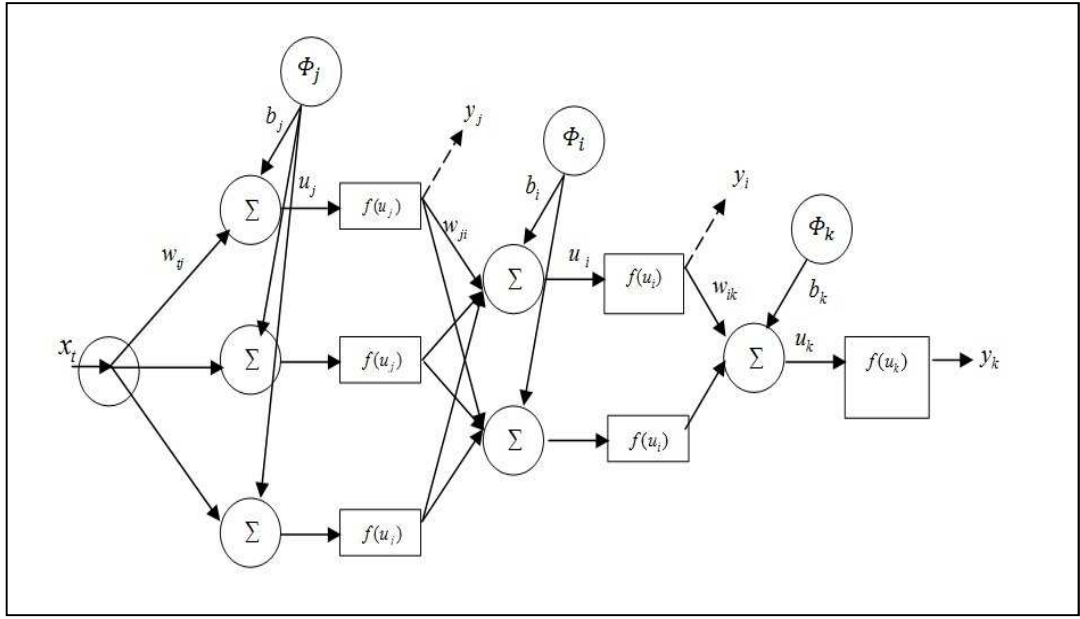
$$\Delta b_k(t) = \eta \delta_k + \alpha \Delta b_k(t-1) \quad (4.28)$$

Değişim miktarı hesaplandıktan sonra t . iterasyondaki yeni ağırlık değerleri:

$$b_k(t) = b_k(t-1) + \Delta b_k(t) \quad (4.29)$$

$b_k(t)$: eşik değer ünitesi ile k . çıktı katmanını proses elemanı arasındaki ağırlığın yeni değeri

Diğer ağırlıklar da benzer şekilde hesaplanır. Aşağıdaki şekilde geri yayımlı bir ağ için 1 girdi, 2 ara katman (gizli katman) ve 1 çıktı katmanından oluşan ve sırasıyla her katmanda 1, 3, 2, 1 proses elemanı (nöron) bulunan basitleştirilmiş örnek bir ağ yapısı verilmiş ve hesaplamalar bu yapı üzerinden örnek olması amacıyla formüle edilmiştir:



Şekil 4.17: ÇKA ağ yapısı örneği

Notasyon aşağıdaki gibidir:

b_i, b_j, b_k : Eşik değer üniteleri için ağırlık değerleri

w_{ik}, w_{ji}, w_{tj} : Nöronlar için ağırlık değerleri

y_i, y_j, y_k : Aktivasyon fonksiyonlarının çıktıları

u_i, u_j, u_k : Nöronlara gelen toplam girdi

Hesaplamalar aşağıdaki gibidir:

$$E = \frac{1}{2} \sum_k e_k^2 \quad (4.30)$$

$$e_k = E[y_k] - y_k \quad (4.31)$$

$\Phi_q = 1$ için,

Feedforward 1 (İleri besleme 1):

$$y_j = f(u_j) \text{ ve } u_j = \left(\sum_t w_{jt} x_t \right) + \Phi_j b_j \quad (4.32)$$

Feedforward 2 (İleri besleme 2):

$$y_i = f(u_i) \text{ ve } u_i = \left(\sum_j w_{ji} y_j \right) + \Phi_i b_i \quad (4.33)$$

Feedforward 3 (İleri besleme 3):

$$y_k = f(u_k) \text{ ve } u_k = \left(\sum_i w_{ik} y_i \right) + \Phi_k b_k \quad (4.34)$$

Sigmoid fonksiyon için

$$f(u) = \frac{1}{1+e^{-u}} \text{ ve } f(u)' = f(u)(1-f(u)) \quad (4.35)$$

Back Propagation 1 (Geri yayılım 1):

$$\frac{\partial E}{\partial w_{ik}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial w_{ik}} = e_k (-1) f'(u_k) y_i \quad (4.36)$$

$$\delta = f(u)(1-f(u))e \quad (4.37)$$

$$\delta_k = e_k f'(u_k) \quad (4.38)$$

$$\frac{\partial E}{\partial w_{ik}} = e_k (-1) f'(u_k) y_i = -\delta_k y_i \quad (4.39)$$

$$\Delta w_{ik} = -\eta \frac{\partial E}{\partial w_{ik}} = +\eta \delta_k y_i \quad (4.40)$$

Eğer yerel optimuma takılmayı önlemek amacıyla momentum katsayısı kullanılırsa:

$$\Delta w_{ik}(t) = \eta \delta_k y_i + \alpha \Delta w_{ik}(t-1) \quad (4.41)$$

$$\frac{\partial E}{\partial b_k} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial w_{ik}} = e_k (-1) f'(u_k) (+1) \quad (4.42)$$

$$\frac{\partial E}{\partial b_k} = -e_k f'(u_k) = -\delta_k \quad (4.43)$$

$$\Delta b_k = -\eta \frac{\partial E}{\partial b_k} = +\eta \delta_k \quad (4.44)$$

Eğer yerel optimuma takılmayı önlemek amacıyla momentum katsayısı kullanılırsa:

$$\Delta b_k(t) = \eta \delta_k + \alpha \Delta b_k(t-1) \quad (4.45)$$

$$w_{ik}(\text{yeni}) = w_{ik}(\text{eski}) + \Delta w_{ik} \quad (4.46)$$

$$b_k(\text{yeni}) = b_k(\text{old}) + \Delta b_k \quad (4.47)$$

Back Propagation 2 (Geri yayılım 2):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial y_i} \frac{\partial y_i}{\partial u_i} \frac{\partial u_i}{\partial w_{ji}} = e_k (-1) f'(u_k) w_{ik} f'(u_i) y_i \quad (4.48)$$

$$\frac{\partial E}{\partial w_{ji}} = - \left[f'(u_i) \sum_k \delta_k w_{ik} \right] y_j \quad (4.49)$$

$$\delta_i = f'(u_i) \sum_k \delta_k w_{ik} \quad (4.50)$$

$$\frac{\partial E}{\partial w_{ji}} = -\delta_i y_j \quad (4.51)$$

$$\Delta w_{ji} = \eta \delta_i y_j \quad (4.52)$$

$$\frac{\partial E}{\partial b_i} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial y_i} \frac{\partial y_i}{\partial u_i} \frac{\partial u_i}{\partial b_i} = e_k (-1) f'(u_k) w_{ik} f'(u_i) (+1) \quad (4.53)$$

$$\Delta b_i = \eta \frac{\partial E}{\partial b_i} \quad (4.54)$$

$$w_{ji}(\text{yeni}) = w_{ji}(\text{eski}) + \Delta w_{ji} \quad (4.55)$$

$$b_i(\text{yeni}) = b_i(\text{eski}) + \Delta b_i \quad (4.56)$$

Back Propagation 3 (Geri yayılım 3):

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial y_i} \frac{\partial y_i}{\partial u_i} \frac{\partial u_i}{\partial y_j} \frac{\partial y_j}{\partial u_j} \frac{\partial u_j}{\partial w_{ij}} = e_k (-1) f'(u_k) w_{ik} f'(u_i) w_{ji} f'(u_j) x_t \quad (4.57)$$

$$\frac{\partial E}{\partial w_{ij}} = - \left[f'(u_j) \sum_i \delta_i w_{ji} \right] x_t \text{ ve } \delta_j = \left[f'(u_j) \sum_i \delta_i w_{ji} \right] \quad (4.58)$$

$$\Delta w_{ij}(t) = \eta \delta_j x_t \quad (4.59)$$

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial y_i} \frac{\partial y_i}{\partial u_i} \frac{\partial u_i}{\partial y_j} \frac{\partial y_j}{\partial u_j} \frac{\partial u_j}{\partial b_j} = e_k (-1) f'(u_k) w_{ik} f'(u_i) w_{ji} f'(u_j) (+1) \quad (4.60)$$

$$\frac{\partial E}{\partial b_j} = -e_k f'(u_k) w_{ik} f'(u_i) w_{ji} f'(u_j) = -\delta_j \quad (4.61)$$

$$\Delta b_j = \eta \delta_j \quad (4.62)$$

$$w_{ij}(yeni) = w_{ij}(eski) + \Delta w_{ij} \quad (4.63)$$

$$b_j(yeni) = b_j(eski) + \Delta b_j \quad (4.64)$$

Süreç Adımları:

- 1) Rastgele değerleri (random values) $w_{ij}, b_j, w_{ji}, b_i, w_{ik}, b_k$ ağırlıklarına ata. ($\eta > 0$ için)
- 2) y_j, y_i, y_k değerlerini (4.32, 4.33 ve 4.34) formüllerini kullanarak hesapla.
- 3) 4.30 ve 4.31 formüllerini kullanarak hatayı hesapla.
- 4) $\delta_k, \delta_i, \delta_j, \Delta w_{ik}, \Delta w_{ji}, \Delta w_{ij}, \Delta b_k, \Delta b_i, \Delta b_j$ değerlerini hesapla.
- 5) 4.46, 4.47, 4.55, 4.56, 4.63 ve 4.64 formüllerini kullanarak ağırlıkları güncelleştir.

4.3.3 LVQ (Öğrenme Vektör Nicelendirmesi) Ağı

1984 yılında Kohonen tarafından geliştirilen LVQ (Learning Vector Quantization) ağının temel felsefesi n boyutlu bir vektörü daha düşük boyutlu vektörlerle temsil etmektir. LVQ için öğrenme; girdi vektörünün hangi çıkış vektörü ile eşleşeceğinin bulunmasıdır. Çıktı katmanındaki vektör setine referans vektörü adı verilir. Amaç; giriş vektörü ile referans vektörlerini eşleştirmektir.

Çok katmanlı perseptronlar gibi LVQ ağları da 3 katmandan oluşmaktadır. İlk katman olan girdi katmanında bilgi işleme olmamaktadır. Dış ortamdan gelen bilgiler bu katmandan ağa giriş yaparlar. İkinci katman, Kohonen katmanı da denilen ara katmandır. Bu katmanda girdi setine en yakın ağırlık vektörü belirlenir. Bu katmandaki her eleman bir referans vektörünü gösterir. Girdi vektörü, girdi katmanı ile Kohonen katmanı arasındaki ağırlıkların oluşturduğu referans vektörlerine uydurulmaktadır. Üçüncü katman olan çıktı katmanında ise girdi vektörünün ait olduğu sınıf belirlenir (Baş, 2006, s.59).

LVQ ağırları genel olarak sınıflandırma ve görüntü işleme problemlerinde kullanılmaktadır. Çıktı vektörlerinden sadece bir tanesi 1 değerini, diğerleri ise 0 değerini alırlar. 1 değerini alan referans vektörü girdi vektörünün sınıfını verir. Girdiler ile çıktılar arasındaki yakınlık Euclidian yöntemi ile bulunur. *Kazanan herşeyi alır (winner takes all)* stratejisine göre çalışır ve bütün ağırlıklar değil sadece kazanan referans vektörünün ağırlıkları güncellenir (Öztemel, 2012, s. 115).

LVQ ağırlarının eğitilmesindeki amaç her iterasyonda girdi vektörüne en yakın referans vektörünü bulmaktır. Referans vektörleri daha önce de belirtildiği gibi Kohonen katmanındaki işlemci elemanları girdi katmanındaki işlemci elemanlara bağlayan ağırlık değerleridir. Öğrenme esnasında sadece referans vektörlerinin ağırlık değerleri değiştirilmektedir. Bu işlem Kohonen öğrenme kuralı kullanılarak gerçekleştirilmektedir. Bilindiği gibi Kohonen öğrenme kuralı, Kohonen tabakasındaki işlemci elemanların birbirleriyle yarışması ilkesine dayanmaktadır. Yarışma kriteri, girdi vektörü ile ağırlık vektörleri (referans vektörleri) arasındaki öklid (euclid) mesafesine bağlıdır. Girdi vektörü X ile referans vektörü W arasındaki mesafe d ile gösterilirse, i . işlemci elemanın mesafesi:

$$d_i = \|W_i - X\| = \sqrt{\sum_j (w_{ij} - x_j)^2} \quad (4.65)$$

ile hesaplanır. Burada w_{ij} , ağırlık vektörünün j . x_j ise girdi vektörünün j . değerini ifade eder. Girdi vektörü ile referans vektörlerinin hepsinin aralarındaki mesafe tek tek hesaplandıktan sonra hangi işlemci elemanın referans vektörü girdi vektörüne en yakın ise yarışmayı o kazanır. Öğrenme işlemi esnasında, sadece girdi katmanını bu işlemci elemana bağlayan ağırlık değerleri değiştirilir. Kazanan işlemci eleman için iki durum söz konusudur:

İlk durumda kazanan işlemci eleman doğru sınıfın bir üyesidir. Bu durumda ilgili ağırlıklar girdi vektörüne biraz daha yaklaştırılmaktadır. Bu, aynı örnek ağırlık tekrar gösterildiğinde yine aynı işlemci elemanın kazanması için yapılır. Bu durumda ağırlıkların değiştirilmesi,

$$W_n = W_0 + \eta(X - W_0) \quad (4.66)$$

denklemini kullanılarak gerçekleştirilir. Burada η öğrenme katsayısıdır. Öğrenme katsayısı zaman içinde 0 değerini alacak şekilde monoton olarak azalır. Bunun nedeni, girdi vektörünün referans vektörüne çok yaklaştığında durması ve aksi yönde tekrar uzaklaşmamasıdır. Aksi takdirde ters yönde tekrar uzaklaşma meydana gelecektir.

Kazanan işlemci elemanın ikinci durumu ise, bu işlemci elemanın yanlış sınıfta olmasıdır. Bu durumda ağırlık vektörü girdi vektöründen uzaklaşmaktadır. Bunun amacı ise, bir daha aynı örnek geldiğinde aynı işlemci elemanın kazanmamasıdır. Bu durumda ağırlıklar,

$$W_n = W_0 - \eta(X - W_0) \quad (4.67)$$

kullanılarak değiştirilir. Öğrenme katsayısının zaman içinde azalması burada da geçerli olmaktadır. Kohonen katmanı ile çıktı katmanı arasındaki ağırlıklar (α) eğitim sırasında değiştirilmemektedir. Kohonen katmanındaki her işlemci elemanın çıktısı y^k ise,

$$y_i^k = \begin{cases} 1 & \text{Eğer } i. \text{ işlemci eleman yarışta kazanırsa} \\ 0 & \text{Aksi halde} \end{cases} \quad (4.68)$$

Kohonen katmanındaki işlemci elemanların çıktıları, bu işlemci elemanları çıktı katmanına bağlayan ağırlık değerleri ile çarpılarak ağırlık çıktısı hesaplanmaktadır. Yani,

$$y_i = \sum_j y_j^k \alpha_{ki} \quad (4.69)$$

olmaktadır. Bu Kohonen katmanında yarışmayı kazanan işlemci elemana bağlı olan çıktı elemanın değerinin 1, diğerlerinin değerinin 0 olması anlamına gelmektedir. Ağırlık çıktıları belirlendikten sonra çıktının doğru sınıflandırılıp sınıflandırılmadığı sorgulanır. Bu sorunun cevabına göre Kohonen katmanındaki

yarışmayı kazanan işlemci elemanı girdi katmanına bağlayan ağırlıklar değiştirilmektedir.

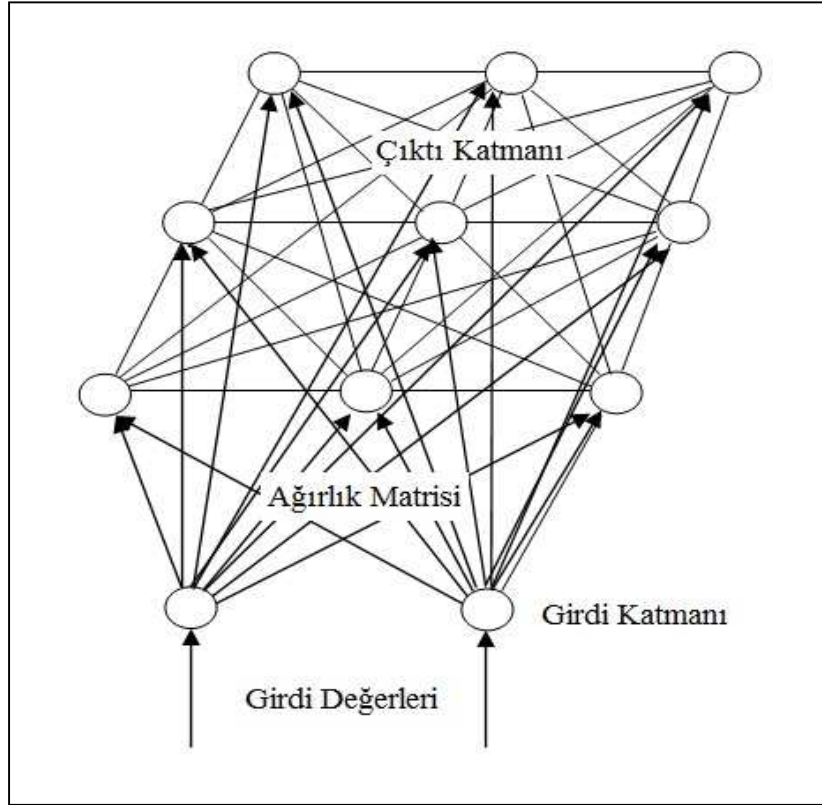
Eğitim setindeki tüm örnekler doğru sınıflandırılınca öğrenme gerçekleştirilmiş olur. Anlatılan bu model standart bir LVQ modelidir (Baş, 2006, s. 60-62).

4.3.4 Kohonen Ağı

Kohonen ağı, 1972 yılında Kohonen tarafından beynin Neocortex tabakasında yaygın olan duyusal haritalardan yola çıkılarak geliştirilen ve özellikle sınıflandırma problemlerinin çözümünde oldukça başarılı olan bir ağıdır. Bir giriş ve bir de çıkış olmak üzere iki katmandan oluşan bu ağ, SOM (Self-Organization-Feature Map - Özörgütlemeli Özellik Haritası) ağı olarak da bilinmektedir. Kohonen Ağları, eğitimi sırasında herhangi bir danışmana ihtiyaç duymadığından ve kestirim yapılacak bir çıktı (bağlı) değişken bulunmadığından dolayı denetimsiz öğrenim gerçekleştiren bir sinir ağı türüdür (Oğuzlar, 2006, s. 62). Bu ağlar, girdi (bağımsız) değişken kümesindeki örüntülerin açığa çıkarılması amacıyla kullanılmaktadır. Bu ağın çıktısında, gözlemler gruplandırılmış olarak elde edilmektedir. Bir grup veya kümenin içinde yer alan gözlemlerin birbirine benzer olduğu, farklı gruplarda yer alan gözlemlerin ise birbirine benzer olmadığı söylenebilir. Kohonen ağları bir girdi ve iki boyutlu bir Kohonen tabakasından oluşmaktadır (Kiang, 2001, s. 163).

Çıktı katmanındaki işlemci elemanlar genel olarak iki boyutlu aralıklar halinde düzenlenir. Bununla birlikte çıkış katmanındaki her işlemci eleman, bütün giriş işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları verilen çıkış işlemci elemanı ile ilgili olan referans vektörünün elemanlarını oluşturur (Saraç, 2004, s. 54).

Yarışmacı öğrenme yöntemini kullanan Kohonen ağında kazanan işlemci elemanlar 1, diğer elemanlar ise 0 değerini alır. Eğitimi sırasında hem yarışmayı kazanan işlemci elemanın hem de komşusu olan işlemci elemanların ağırlıkları güncellenir.



Şekil 4.18: Kohonen ağı (Frohlich, 1997)

Kohonen ağı'nın eğitiminde, herhangi bir t zamanında örnek setinden herhangi bir örnek ağı gösterilir. Girdi vektörü X ve ağırlık vektörü W normalize edilmiş olmalıdır. Çıktı elemanlarından kazanan işlemci elemanı bulabilmek için iki yöntem vardır. Bunlardan birincisinde, her elemanın çıktısı, ağırlıklarla girdilerin çarpımının toplamı ile bulunur. Bu toplamın matematiksel ifadesi ise,

$$y_i = \sum_i w_{i0} x_i \quad (4.70)$$

şeklindedir. Buradan elde edilen çıktı değerlerinden en yüksek değere sahip olan işlemci eleman yarışmayı kazanır. Bu işlemci elemanın k . eleman olması durumunda,

$$y_k = 1 \text{ ve } y_i = 0, \quad i = 1, 2, \dots \text{ ve } i \neq k \quad (4.71)$$

bulunur. İkinci yöntemde ise, Öklid mesafesi (d) kullanılarak elde edilen, girdi vektörüne en yakın ağırlık vektörüne sahip işlemci eleman kazanan elemandır. İki vektör arasındaki mesafe,

$$d_j = \|X - W_j\| \quad (4.72)$$

ile hesaplanır. Her çıktı elemanı için bu mesafeler hesaplanmakta ve en küçük mesafe değerine sahip işlemci eleman kazanan eleman olarak belirlenmektedir. Kazanan işlemci elemanın belirlenmesinin ardından bu elemanın ve komşularının ağırlıkları,

$$W(t+1) = W(t) + \eta g(i, k) [X(t) - W(t)] \quad (4.73)$$

formülü kullanılarak değiştirilir. Burada η , eğitim esnasında değeri zamanla küçültülen öğrenme katsayısını göstermektedir. $g(i, k)$, komşuluk fonksiyonu olup i ve k elemanlarının komşuluklarını belirlemektedir. $i=k$ olması durumunda $g(i, k)=1$ olur. Bu fonksiyon zaman içerisinde azalan bir fonksiyondur. Genel olarak $g(i, k)$,

$$g(i, k) = \left[\exp\left(-\frac{\|d_i - d_k\|^2}{2\sigma^2}\right) \right] \quad (4.74)$$

şeklinde ifade edilir. Formülde, d_i ve d_k i . ve k . elemanların pozisyonunu gösteren vektörler ve σ ise komşuluk alanının genişliğini gösteren ölçüttür. Bu genişlik zaman içerisinde azalmaktadır (Baş, 2006, s. 65-66).

4.3.5 ART (Adaptive Resonance Theory) Ağları

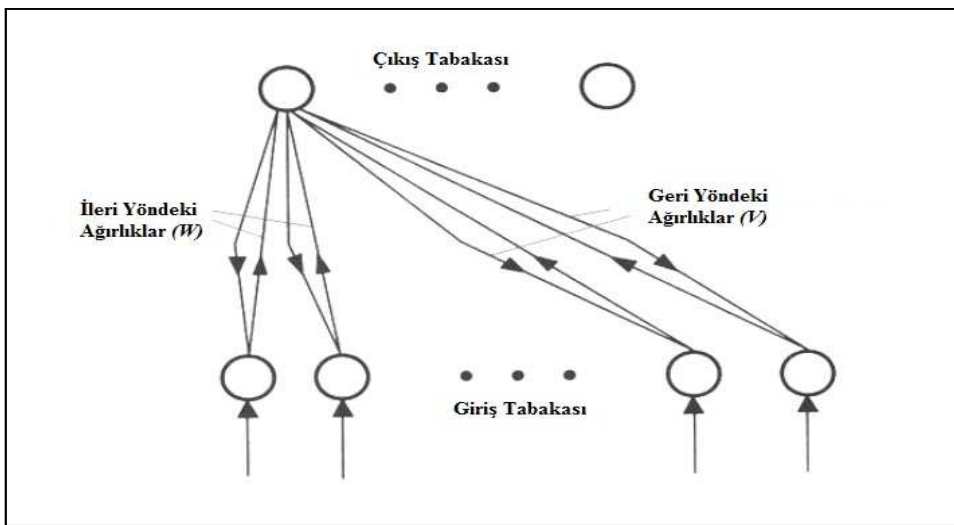
Adaptive Resonance Theory (Uyarlanır Rezonans Teorisi) ağı, 1970'lerin ortalarında Stephen Grossberg tarafından geliştirilmiştir. Uyarlanır rezonansa dayanan giriş verilerinin kategorilerini bu ağ oluşturmaktadır. Bu topoloji biyolojik olarak akla uygundur ve danışmansız öğrenme işlevini kullanmaktadır.

Bu sistem belirgin giriş veri setini analiz eder ve muhtemel özellikleri kontrol eder veya giriş vektörü içerisindeki karakteristikleri sınıflandırır (Elmas, 2003, s. 86).

ART, adını öğrenme ve hatırlama arasındaki karşılıklı etkileşimi gerçekleştiren yöntemden yani rezonanstaki almıştır. ART yapay sinir ağlarında işlemci elemanların çıktılarını katmanlar arasında sürekli olarak ileri-geri hareket etmektedir. Bu esnada eğer örnek girdi, belirlenmiş bir sınıfa uyuyorsa ağ kararlı hale gelmekte ve ART ağı rezonanstadır denilmektedir (Gülseçen, 1993, s. 75).

4.3.5.1 ART-1 Ağı

ART-1 ağı, sadece ikili (binary) girdiler ile çalışan en basit ART ağı modelidir. Bu ağıda, karşılaştırma (girdi) ve tanıma (çıkıtı) katmanları yer almaktadır. Karşılaştırma katmanındaki bütün işlemci elemanlar, tanıma katmanındaki her işlemci elemanla bağlantı halindedir. Bu bağlantılar sürekli değerlerden oluşan LTM bağlantılarıdır. Bu bağlantıların en önemli özelliği ileri doğru olmalarıdır. Bununla birlikte tanıma katmanından karşılaştırma katmanına, geriye doğru ikili değerlerden meydana gelen bağlantılar bulunmaktadır. ART-1 ağının genel şeması Şekil 4.19'da gösterilmiştir (Baş, 2006, s. 68).



Şekil 4.19: ART-1 ağının genel yapısı

4.3.6 Hopfield Ağı

John Hopfield tarafından 1982 yılında geliştirilmiş olan Hopfield ağı, en çok optimizasyon problemleri için kullanılmaktadır. Tek katmanlı ve geri dönüşümlü olan bu ağ, genellikle ikili (0 veya 1) ve bipolar (+1 veya -1) girişler kabul etmektedir. ART ağında olduğu gibi tek katmandaki işlemci elemanlarının her biri bir diğerine bağlanmıştır. Ayrıca her hücrenin çıkışı, bir ağırlıkla işleme tabi tutularak diğer hücrelere giriş değeri olarak gönderilmektedir. Ağın bağlantı değerleri bir enerji fonksiyonu olarak saklanır. Hopfield ağının eğitilmesi sadece bir adım alır ve w_{ij} ağırlıkları aşağıdaki gibi hesaplanır:

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{c=1}^p x_i^c x_j^c, & i \neq j \\ 0, & i = j \end{cases} \quad (4.75)$$

Burada;

w_{ij} : i işlemci elemanından j işlemci elemanına olan bağlantının ağırlığını,

x_i^c : c sınıfı için eğitime giriş deseninin i . elemanını,

p : sınıf sayısını,

N : işlemci eleman sayısını göstermektedir.

Denklem (4.75)'de $w_{ij} = w_{ji}$ ve $w_{ii} = 0$ durumları ağın kararlılığını garantiler.

Bilinmeyen bir bilgi ağa girildiğinde, ağın çıkış verileri bilinmeyen bir desen elemanlarına eşitlenir, yani;

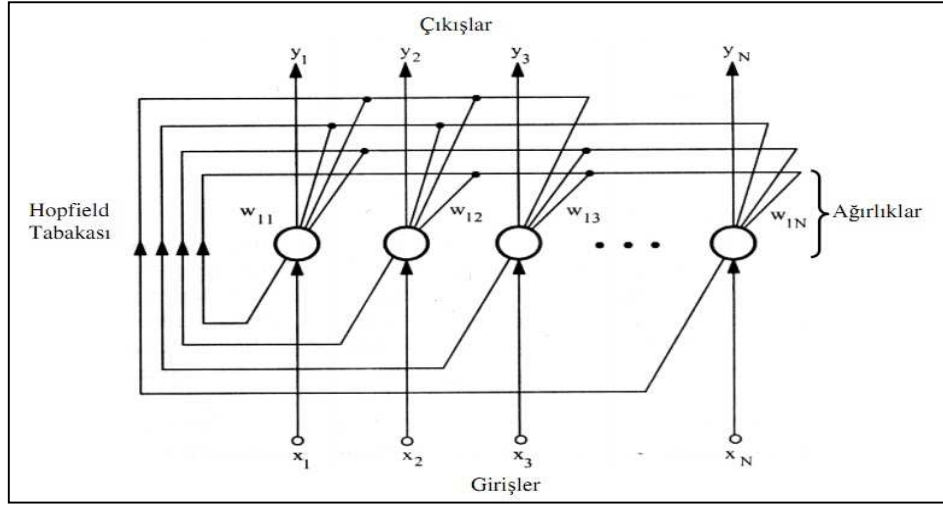
$$y_i(0) = x_i, \quad 1 \leq i \leq N \quad (4.76)$$

bulunur. Bu başlangıç değerleri ile başlayarak Hopfield ağı bir sonraki denklemi kullanarak minimum enerji durumuna geçmek için döngüye girer.

$$y_i(k+1) = f \left[\sum_{j=1}^N w_{ij} y_j(k) \right], \quad 1 \leq i \leq N \quad (4.77)$$

Burada f aşağıdaki gibi tanımlanan keskin sınırlama fonksiyonudur (Duman, 2006, s. 38-39).

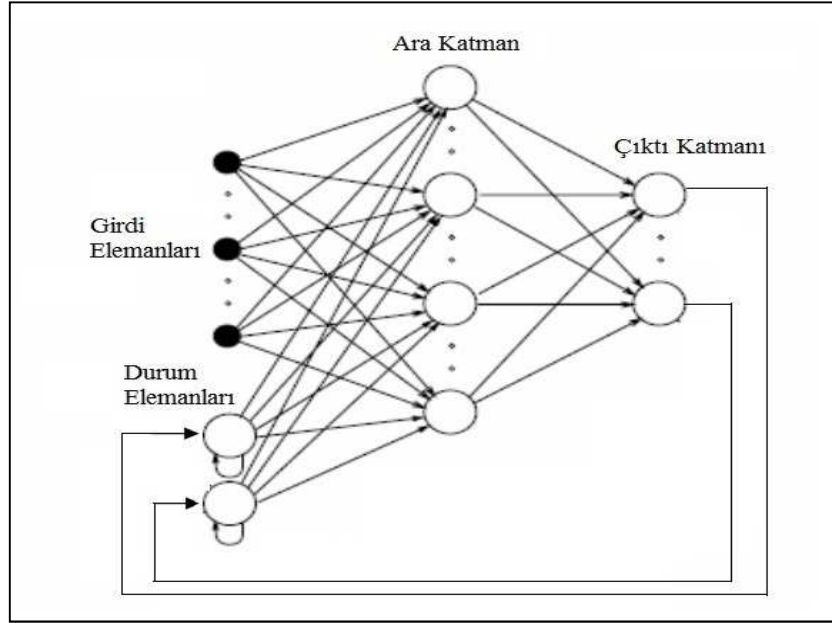
$$f(x) = \begin{cases} -1 & x < 0 \\ +1 & x > 0 \end{cases} \quad (4.78)$$



Şekil 4.20: Hopfield ağı (Sağıroğlu vd. 2003, s. 62)

4.3.7 Jordan Ağı

Jordan ağı, çok katmanlı geri beslemeli bir yapay sinir ağıdır. Çok katmanlı algılayıcılara benzer bir yapıda olan Jordan ağlarında, girdi, çıktı ve ara proseslere ek olarak durum elemanları (state units) adı verilen özel işlemci elemanlar da bulunmaktadır. Durum elemanları, çıktı tabakasından aldıkları aktivasyon değerlerini bir sonraki iterasyona girdi olarak taşımakla görevlidirler. Diğer katmanlardaki işlemci elemanlar ise çok katmanlı algılayıcılara (ÇKA) benzer bir şekilde çalışırlar (Baş, 2006, s. 80).



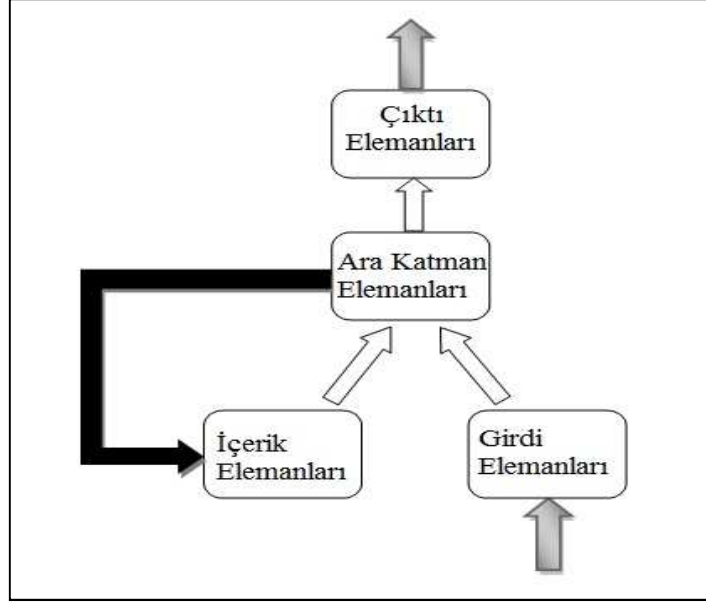
Şekil 4.21: Jordan ağı (Kröse and Smagt, 1996, s. 48)

4.3.8 Elman Ağı

Elman tarafından 1990 yılında geliştirilen bu ağı; girdi katmanı, çıktı katmanı, ara katmanlar ve bunlara ilaveten içerik katmanından (context layer) oluşmaktadır. İçerik elemanları, ara katmandan aldıkları aktivasyon değerlerini bir sonraki iterasyona girdi olarak taşırlar. Elman ağları ile Jordan ağları birbirine çok benzer. Bu benzerlikle birlikte temel farklılıklar da mevcuttur. Bu farklılıklardan ilki geri besleme yaptıkları aktivasyon değerlerini çıktı katmanından değil, ara katmandan almalarıdır. Diğer bir farklılık ise, içerik elemanlarının kendilerine bağlantılarının bulunmamasıdır. Elman ağlarında ara elemanlar ve içerik elemanları arasındaki bağlantı ağırlıkları sabit bir değerdir (Baş, 2006, s. 81).

Elman ağının yapısı Şekil 4. 22' de gösterilmiştir. Elman ağının yukarıda anlatılan proses elemanları da şekilde görülmektedir. Bunlardan girdi ve çıktı elemanları dış dünya ile etkileşim içindedir. Girdi elemanları dış dünyadan aldıkları bilgileri ara katmanlara iletirler. ÇKA' da olduğu gibi, Elman ağında da girdi elemanlarının bilgi işleme özellikleri yoktur. Çıktı elemanları ise ağın çıktısını dış dünyaya iletirler. Çıktı ünitelerinin bilgi işleme fonksiyonları

doğrusaldır. Sadece kendilerine gelen bilgileri toplarlar. Ara katman elemanları ise hem doğrusal hem de doğrusal olmayan aktivasyon fonksiyonlarına sahip olabilirler. İçerik elemanları ara katman elemanlarının önceki aktivite değerlerini hatırlamak için kullanılmaktadır. Bu elemanların özelliği bir adım gecikmeyi (one step time delay) içeriyor olmalarıdır (Öztemel, 2012, s. 166).



Şekil 4.22: Elman ağı yapısı (Öztemel, 2012, s. 166)

5. PORTFÖY OPTİMİZASYONU

Günümüzde finansal piyasalar ülke sınırlarını aşarak evrensel bir yapıya bürünmüş ve yatırım yaparak elindeki kaynağı en iyi şekilde değerlendirmek isteyen milyonlarca insanın beslediği canlı bir organizma haline gelmiştir. İşin daha da önemli kısmı; bu piyasaların albenisi ve çekiciliği insanlara çok cazip gelmiş ve giderek daha da cazip hale gelmektedir. Çünkü rasyonel kararlar doğrultusunda yatırım yaparak çok büyük getiriler elde eden yatırımcıların sayısının giderek artması, yeni yatırımcılara büyük bir örnek teşkil etmektedir. Bununla birlikte piyasada yatırım yapan yatırımcı kadar, piyasada yatırım yapılabilecek yatırım enstrümanının sayısı da çok fazladır. Ayrıca, her günün akabinde o günkü pazar koşullarına göre yatırım enstrümanlarının fiyatları da değişmektedir.

Finansal varlıklar bilindiği gibi yapısı gereği çok değişkenlidir. Bazılarının değeri artarken bazılarının değeri düşebilir ya da aynı yönde hareket edebilirler. Bazıları çok yüksek fiyat değişkenliği gösterirken bazıları küçük fiyat aralıklarıyla hareket edebilir. Varsayalım ki mevcut varlığı iki farklı sektöre yatırma imkânımız olsun: dondurma şirketi ve bir de şemsiye şirketi. Bütün sermayenin şemsiye şirketine yatırıldığını düşündüğümüzde, eğer o yıl yağmurlu geçerse büyük kazanç elde edilir, fakat güneşli bir yıl olursa büyük oranda zarar etme durumu söz konusudur. Bununla birlikte her iki sektöre birlikte ve aynı oranda yatırım yapılırsa, belki çok büyük kazanç elde edilmemiş olacaktır fakat çok büyük oranda da zarar elde edilmeyecektir. Böylelikle getiriler ve riskler (zararlar) azaltılmış olacaktır (Roudier, 2007, s. 5).

Bu açıklamalar ışığında, milyonlarca kişinin, binlerce yatırım enstrümanı arasından, her gün yeniden oluşan fiyatlar doğrultusunda en iyi yatırımı yapma çabası içerisinde olduğu açıktır. Burada sözü edilen “en iyi yatırımı yapma çabası” daha genel bir ifadeyle, eldeki kaynakların ulaşılmak istenen amaçlar doğrultusunda yönlendirilmesi için gerçekleştirilen finansal planlamalar bütünüdür. Yani “portföy optimizasyonu” işlemidir.

5.1 Portföy Kuramı

Portföy (Portfolio) ile ilgili değişik kaynaklarda farklı tanımlamaları görmek mümkündür. Kelime anlamı olarak portföy, para çantası ya da cüzdan demektir. Menkul kıymetler açısından portföy ise, ağırlıklı olarak hisse senedi, tahvil gibi çeşitli menkul değerler ve türev ürünlerden meydana gelen, belirli bir kişi ya da kurumun elinde bulunan finansal nitelikteki varlıklardır. Bilindiği gibi, menkul kıymetlere yatırım, belli amaçları yerine getirmek için yapılmaktadır. Portföy, her ne kadar belli menkul değerlerden meydana gelse de, bu değerler arasında bir korelasyon olduğundan, portföy, kendine özgü ölçülebilir nitelikleri olan bir varlıktır. Bu nedenle portföyü, içerdiği menkul kıymetlerin basit bir toplamı olarak düşünmek yanlış olur (Korkmaz ve Ceylan, 2006, s. 471).

Bir portföyü hisse senedi ve tahvil gibi temel menkul kıymetler dışındaki yatırım araçlarıyla da oluşturmak mümkündür. Bu tür portföyler oluşturulurken, yatırım araçları arasında karşılaştırma yapılır. Yatırım sürecinde hangi tür varlıkların daha verimli olacağını, istatistikî ve matematiksel yöntemlerle hesaplayarak tahmin etmek mümkündür.

5.2 Portföy İle İlgili Temel Kavramlar

Portföy optimizasyonu modellerine geçmeden önce bu modellerde kullanılacak olan temel kavramlar üzerinde durmak yerinde olacaktır. Bu kavramlar dönemlik getiri, beklenen getiri, sapma ölçütleri, kovaryans, korelasyon katsayısı ve finansal varlıklardan muhtelif adetlerde alınarak oluşturulan portföylerin beklenen getiri-varyans ölçümleridir.

5.2.1 Dönemlik Getiri

Dönemlik getiri, herhangi bir yatırım enstrümanının belli bir zaman dilimi içerisindeki toplam getirisini ifade etmektedir. Getiri oranı r olmak üzere dönemlik getiri,

$$r = \frac{P_t - P_{t-1} + D_t}{P_{t-1}} \quad (5.1)$$

şeklinde hesaplanmaktadır (Dobbins, Witt and Fielding, 1994, s. 5). Burada;

P_t : Hisse senedinin dönem sonu fiyatı

P_{t-1} : Hisse senedinin dönem başı fiyatı

D_t : Temettü ödemesi (dividend payment)

Farklı dönemlerdeki getirileri karşılaştırmak için getiriler genellikle yıllık baza indirgenir. Getirileri yıllık bazda ifade etmenin çeşitli yolları vardır. Getiriler; basit, bileşik veya sürekli bileşik getiri hesaplamaları ile yıllık baza indirgenir (Ulucan, 2004, s. 7).

Basit getiri hesaplaması, menkul kıymeti elde bulundurma dönemi boyunca her gün aynı getirinin elde edildiğini varsayar.

$$r_{basit} = \frac{1}{t} \left[\frac{P_t - P_{t-1} + D_t}{P_{t-1}} \right] \quad (5.2)$$

Burada t , elde bulundurma döneminin yıl birimine göre uzunluğunu ifade etmektedir.

Bileşik getiri hesaplaması, menkul kıymeti elde bulundurma dönemi sonunda elde edilen getiri ve anaparanın tekrar yatırıma dönüştürülerek yıllık bazda büyüdüğünü varsayar. Bileşik getiri hesaplaması ise şu şekilde yapılır:

$$r_{bileşik} = N \left[\frac{P_t - P_{t-1} + D_t}{P_{t-1}} \right]^{\frac{1}{tN}} \quad (5.3)$$

Burada N , bir yıl içinde ele alınan dönem sayısını göstermektedir.

Sürekli bileşik getiri hesaplama yöntemi ise elde bulundurma döneminin sonsuz sayıda zaman dilimlerine bölünerek, her bir dilimde getirisinin

hesaplanması ve anapara ile birlikte bir sonraki zaman dilimine aktarılması esasına göre çalışır. Matematiksel ifadesi ise şöyledir (Ulucan, 2004, s. 8):

$$r_{\text{sürekli}} = \frac{1}{t} \ln \left[\frac{P_t + D_t}{P_{t-1}} \right] \quad (5.4)$$

Bu formülleri gelecekle ilgili kararlarda kullanmamız mümkün değildir. Gelecekle ilgili kararlarımız beklentilerimizin gerçekleşme olasılıklarına bağlı olmakta, diğer bir ifade ile beklenen getiri ve riske dayanmaktadır.

5.2.2 Beklenen Getiri

Yatırım kararları geleceğe yönelik olduğundan yalnızca dönemlik getiri çok fazla önem arz etmemektedir. Bunun yerine beklenen değer / getiri (expected value) kavramını kullanmak gerekmektedir. Bir yatırımın **Beklenen Değeri**, muhtemel getirilerinin olasılık dağılımının beklenen değeridir. Yani, çeşitli durumlardaki beklenen getirilerin ağırlıklı ortalamasıdır. Matematiksel olarak şu şekilde gösterilmektedir (Brandimarte, 2006, s. 40):

$$E[R] = \mu = \sum_{j=1}^N r_j p_j \quad (5.5)$$

$E[R] = \mu$: Beklenen getiri

r_j : Her bir durumun beklenen getirisi

p_j : Her bir durumun olasılık değeri

Bu formülden yararlanarak; ekonominin çeşitli durumlarına göre getirisi ve bu getirilere ait olasılıkları Tablo 5.1'de verilen bir şirkete yatırım yapan yatırımcının beklenen getirisini şu şekilde hesaplayabiliriz:

Tablo 5.1: Ekonominin durumuna göre A şirketinin getirisi

Ekonomi	Olasılık	Getiri
Durgun	0.17	-0.02
Hafif Durgun	0.24	0.00
Normal	0.36	0.02
İyi	0.13	0.03
Mükemmel	0.10	0.05

Bu verilere göre yatırımcının beklenen getirisi;

$$\begin{aligned} E[R] &= \mu = \sum_{j=1}^5 r_j p_j \\ &= (0.17)(-0.02) + (0.24)(0.00) + (0.36)(0.02) + (0.13)(0.03) + (0.10)(0.05) \\ &= 0.0127 \end{aligned}$$

olarak bulunur.

Beklenen getirinin iki önemli özelliği vardır (Ulucan, 2004, s. 8). Bunlardan birincisi; iki getirinin toplamının beklenen değerinin, bu iki getirinin ayrı ayrı beklenen getiri değerleri toplamına eşit olmasıdır. Yani,

$$E[R_1 + R_2] = \sum_{j=1}^N (r_{1j} p_{1j} + r_{2j} p_{2j}) = \sum_{j=1}^N r_{1j} p_{1j} + \sum_{j=1}^N r_{2j} p_{2j} = E[R_1] + E[R_2] \quad (5.6)$$

İkinci özellik ise; herhangi bir getirinin bir sabitle çarpımının beklenen getirisinin, getirinin beklenen değerinin sabitle çarpımına eşit olduğudur. Bu özellikten,

$$E[k.R] = \sum_{j=1}^N (k.r_j p_j) = k \cdot \sum_{j=1}^N r_j p_j = k.E[R] \quad (5.7)$$

yazılabilir (Ulucan, 2004, s. 9).

5.2.3 Sapma Ölçütleri

Bir yatırımın sadece beklenen getirisine değil, aynı zamanda elde edilen getirilerin ortalamadan ne kadar farklı olduğuna bakmak gerekmektedir. Bu farklılık, basitçe her bir getirinin ortalamadan farkına bakılarak alınır. Ancak bu hesabı yapmanın en büyük sakıncası, bu farklardan elde edilecek negatif ve pozitif değerlerin birbirini götürmesi ile anlamsız bir sonuca ulaşmaktır. İşte bu sorunu çözenin iki yolu vardır. Bu yollar ortalama mutlak sapma ve varyans - standart sapma ölçümlerinin kullanılmasıdır.

5.2.3.1 Ortalama Mutlak Sapma

Ortalama mutlak sapma yönteminde getirilerin ortalamadan ne kadar farklı olduğu hesaplanırken, işaretlere bakılmadan yalnız mutlak değerlerin hesap edilmesidir. Bu yöntem analitik hesaplamalar için çok tercih edilen bir yöntem değildir. Aşağıdaki gibi hesaplanır (Ulucan, 2004, s. 9):

$$AAD = \sum_{j=1}^N p_j |r_j - \mu| \quad (5.8)$$

μ : Beklenen getiri

r_j : Her bir durumun beklenen getirisi

p_j : Her bir durumun olasılık değeri

5.2.3.2 Varyans ve Standart Sapma (Risk)

Getirilerin ortalamadan ne kadar farklı olduğunu ölçmede kullanılan hesaplamalardan biri de varyanstır. Varyans, getiriler ile beklenen getirilerin farklarının kareleri toplamı alınarak hesaplanan bir risk ölçütüdür. Portföy optimizasyonu modellerinde risk ölçütü olarak genellikle varyanstan yararlanır. Matematiksel olarak şu şekilde ifade edilmektedir (Karan, 2004, s. 139):

$$Var = \sigma^2 = \sum_{j=1}^N p_j (r_j - \mu)^2 \quad (5.9)$$

Varyans kare ile ifade edildiğinden elde edilen sonucu açıklamak biraz zordur. Bu nedenle riski, varyansın karekökü olan standart sapma ile açıklamak daha anlamlıdır.

$$Standart Sapma = \sigma = \sqrt{\sum_{j=1}^N p_j (r_j - \mu)^2} \quad (5.10)$$

Tablo 5.1'de verilen şirketin getirilerinin varyans ve standart sapması şu şekilde hesaplanır:

Tablo 5.2: Bir şirketin varyans ve standart sapmasının hesaplanması

Getiri (r_j)	Beklenen Getiri (\bar{r})	$r_j - \bar{r}$	$(r_j - \bar{r})^2$	Olasılık (p_j)	$p_j (r_j - \bar{r})^2$
-0.02	0.013	-0.033	0.001089	0.17	0.0001851
0.00	0.013	-0.013	0.000169	0.24	0.0000406
0.02	0.013	0.007	0.000049	0.36	0.0000176
0.03	0.013	0.017	0.000289	0.13	0.0000376
0.05	0.013	0.037	0.001369	0.10	0.0001369
					$\sigma^2 = 0.000418$
					$\sigma = 0.020440$

Bir varlığın getirilerinin sabit bir değerle toplanmasıyla elde edilen getiri serisinin varyansı, o varlığın varyansına eşittir. Bununla beraber, bir varlığın getirilerinin bir sabit değerle çarpılmasıyla elde edilen getiri serisinin varyansı, varlığın varyansı ile sabitin karesinin çarpımına eşittir. Yani,

$$\begin{aligned} Var(k + R) &= Var(R) \\ Var(k.R) &= k^2.Var(R) \end{aligned} \quad (5.11)$$

eşitlikleri yazılabilir.

5.2.3.2.1 Risk Kaynakları

Gerek menkul kıymet yatırımcısı olsun gerek olmasın her insan yaşadığı süreç içerisinde, pazardan alacağı meyve gibi basit bir durumdan hayatı boyunca devam edeceği meslek dalının seçimi gibi karmaşık durumlara varıncaya kadar belirli bir amaca yönelik olarak tercih yapmak zorunda kalmaktadır. İnsanlar bu tercihlerini yaparken ileriye yönelik istedikleri bir sonucu elde etme amacı taşırlar. Sahip olunan alternatifler içerisinde birine karar vermek çoğu zaman bir riski ya da belirsizliği beraberinde getirmektedir. Bir finansal varlığa veya daha geniş anlamda finansal varlıkların belli ağırlıklarla oluşturdukları bir portföye ilişkin toplam riski oluşturan unsurlar iki grup altında incelenir. Toplam riskin bileşenleri, sistematik risk ve sistematik olmayan risktir. Toplam riski matematiksel olarak,

$$\sigma_i^2 = \beta_i^2 \sigma_m^2 + \sigma_e^2 \quad (5.12)$$

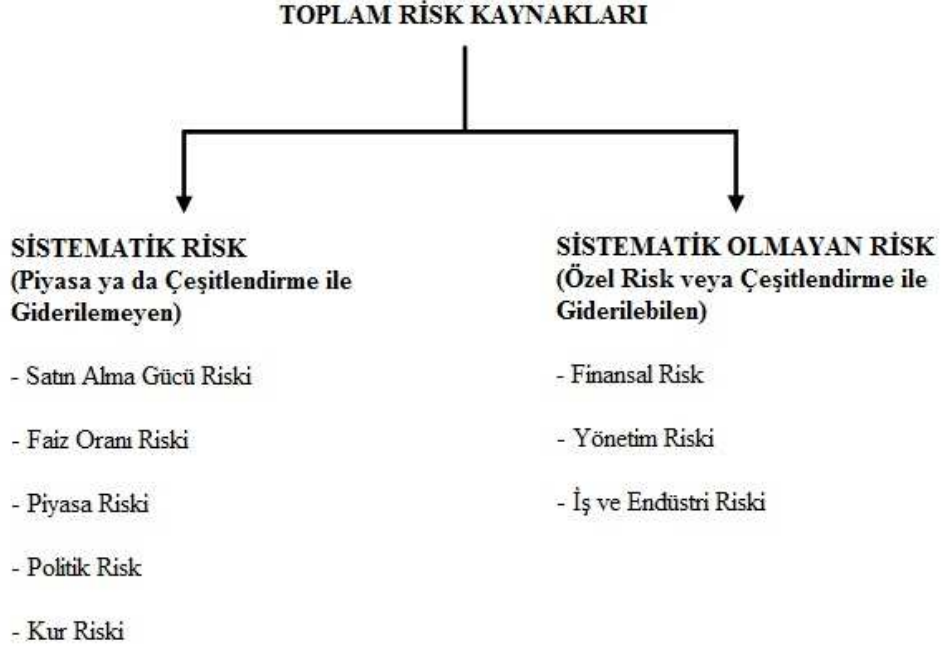
şeklinde gösterebiliriz (Dobbins vd, 1994, s. 142). Burada;

σ_i^2 : Yatırım yapılan menkul değerın toplam riskini,

β_i : Menkul kıymetin sistematik riske karşı duyarlılığını yani betasını,

σ_m : Sistematik riski (Pazar riski),

σ_e : Menkul kıymetin kendine özgü olan ve sistematik olmayan riskini ifade etmektedir.



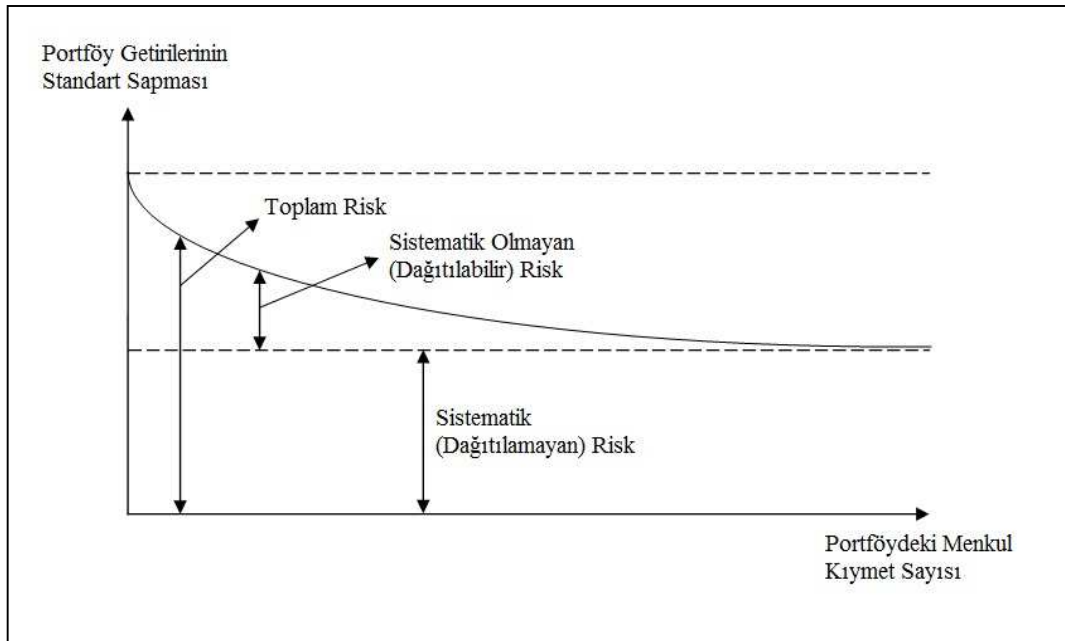
Şekil 5.1: Toplam risk kaynakları

Toplam risk içerisindeki sistematik risk (pazar riski), pazarın taşıdığı risklerin oluşturduğu risk olup pazardaki finansal varlıkların getirilerini muhtelif oranlarda fakat genel olarak aynı yönde etkileyen risktir. Bir ülkede meydana gelen ekonomik, politik ve sosyal nedenlerle pazarda oluşan olumlu veya olumsuz yöndeki değişimlerin sonucunda finansal varlıklar üzerindeki etkisi ortaya çıktığı ve tüm finansal varlıkların getirilerini aynı yönde etkilediği için yatırımcının yaptığı çeşitlendirme ile giderilemeyen risk türüdür (Sevinç, 2007, s.7).

Yatırımcının toplam riski içerisindeki diğer bir risk çeşidi ise sistematik olmayan risktir. Sistematik olmayan risk, bir şirket veya sektöre özgü olan risktir. İşçi grevi, yönetim hataları, keşifler, reklam kampanyaları, tüketici tercihlerindeki değişimler, şirket gelirinde sistematik olmayan değişimlere yol açabilir. Sistematik olmayan faktörler, diğer endüstriler ve genel olarak menkul kıymetler piyasasını etkileyen faktörlerden bağımsızdır (Teziş, 1987, s. 32). Sistematik olmayan risk, çok iyi çeşitlendirilmiş bir portföyle ortadan kaldırılabilecek bir risk türüdür (Bekçioğlu, 1984, s. 59). Sistematik riskin kontrol edilmesi imkânsızken,

sistemik olmayan riskin kaynaklarında yapılan deęişmelerle ve yönlendirmelerle azaltılması veya yok edilmesi mümkündür.

Şekil 5.2’de de görüldüğü gibi portföy içindeki menkul kıymet sayısı arttırıldıkça, diđer bir deyişle çeşitlendirme yapılarak toplam risk belli bir seviyeye kadar düşürülebilir. Burada toplam riskin düşmesinin sebebi de sistemik olmayan riskin çeşitlendirme ile azaltılmasıdır. Daha önce belirtildiği gibi sistemik riskin çeşitlendirme ile azaltılamamasının sebebi ise, pazardaki tüm menkul kıymetleri aynı yönde etkilemesidir. Dolayısıyla portföye ne kadar menkul kıymet alınır alınmaz, bu varlıklar pazardaki deęişmelerden hep birlikte ve genellikle aynı yönde etkileneceği için portföydeki sistemik risk ortadan kaldırılamayacaktır.



Şekil 5.2: Menkul kıymet sayısı ile risk arasındaki ilişki (Farrell, 1997, s. 30)

5.2.4 Kovaryans

Kovaryans, iki tesadüfî deęişkenin zaman içinde görelî hareketlilięinin aynı andaki uyumunu (ilişkisini) gösteren istatistikî bir ölçüttür. Örneęin A ve B gibi iki varlığın getirileri arasındaki kovaryans deęeri şu şekilde hesaplanır:

$$Cov(R_A, R_B) = \sum_{i=1}^N (r_{Ai} - \mu_A)(r_{Bi} - \mu_B) p_i \quad (5.13)$$

Eęer varlıkların ortalamalarından sapmaları aynı zaman dilimlerinde aynı yönde olursa, varlıklar arasında pozitif bir kovaryans mevcuttur. Öte yandan, varlıkların ortalamalarından sapmaları aynı zaman dilimlerinde farklı yönlerde ise, varlıklar arasındaki kovaryans negatif bir deęer alacaktır. Varlıkların ortalamalarından sapma deęerleri arasında anlamlı bir ilişki yoksa, o zaman da kovaryans deęeri sifıra yaklaşacaktır.

İki varlığın getirilerinin toplamlarının varyansı, varlıkların ayrı ayrı varyansları ve aralarındaki kovaryansın iki katının toplamına eşittir (Focardi and Fabozzi, 2004, s. 514).

$$Var(R_1 + R_2) = Var(R_1) + Var(R_2) + 2.Cov(R_1, R_2) \quad (5.14)$$

Ancak kovaryans hesaplaması ile elde edilen deęeri, negatif ya da pozitif bir ilişki olup olmadığının belirlenmesi dışında yorumlamak güçtür. Çünkü elde edilen deęerin büyüklüğünü açıklamak mümkün deęildir. Bu nedenle daha anlamlı bir ölçüt olan **korelasyon katsayısını** kullanabiliriz.

5.2.5 Korelasyon Katsayısı

Kovaryans deęerlerinin $(-1, +1)$ aralığında salınacak şekilde standardize edilmiş hali korelasyon katsayısı olarak adlandırılır. Kovaryansa ek olarak ilişkinin derecesini yani gücünü belirlemekte olup, kovaryans deęerini kullanarak korelasyon katsayısını şu şekilde hesaplayabiliriz (Capinski and Zastawniak, 2003, s. 99):

$$\delta_{A,B} = \frac{Cov(R_A, R_B)}{\sigma_A \sigma_B} \quad (5.15)$$

Burada;

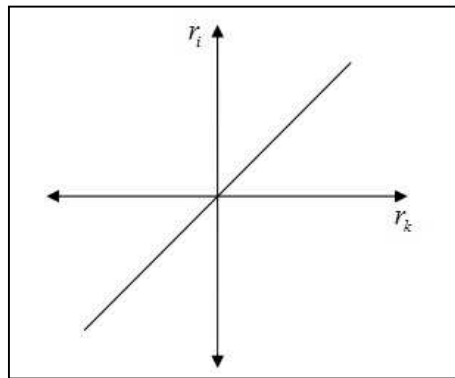
σ_A : A deęişkeninin riski

σ_B : B deęişkeninin riski

δ_{AB} : A ve B deęişkenleri arasındaki korelasyon katsayısı

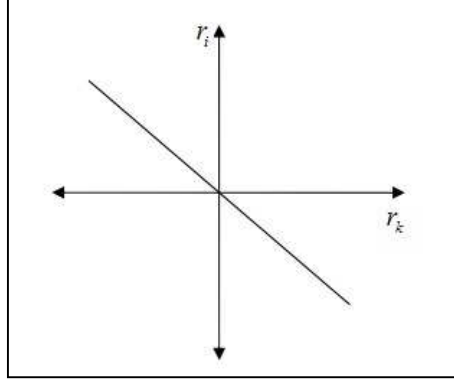
$Cov(R_A, R_B)$: A menkul deęeri ile B menkul deęerinin getirileri arasındaki kovaryans deęeri.

Korelasyon katsayısı -1 ile $+1$ arasında deęer almaktadır. İki tesadüfi deęişken arasında sıfırdan büyük korelasyon katsayısı için pozitif yönlü bir ilişki ve $+1$ olduğunda da pozitif (aynı) yönde güçlü bir ilişki olduğu anlaşılır. Sıfırdan küçük korelasyon katsayısı için iki menkul kıymet arasında negatif (zıt) yönlü bir ilişki ve -1 olduğunda da ters yönlü güçlü bir ilişkinin varlığından söz edilir. Korelasyon katsayısının sıfır (0) olması durumunda menkul kıymetler arasında herhangi bir ilişkinin söz konusu olmadığı anlamı ortaya çıkar.



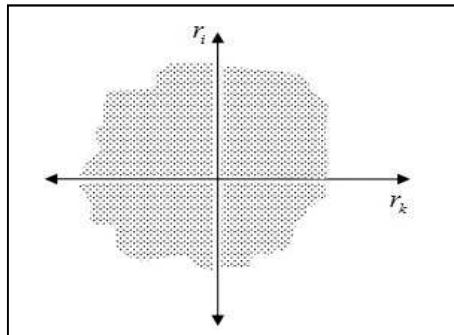
Şekil 5.3: Pozitif tam korelasyon

Şekil 5.3’de aralarında tam pozitif korelasyon olan iki menkul kıymetin getirilerinin hareket yönü gösterilmektedir. Buna göre, iki menkul kıymet arasındaki korelasyon katsayısı (+1) ise “bu iki menkul kıymetin getirileri aynı yönde ve aynı oranda hareket ediyordur” denilir.



Şekil 5.4: Negatif tam korelasyon

Şekil 5.4’de aralarında tam negatif (ters yönlü) korelasyon olan iki menkul kıymetin getirilerinin hareket yönü gösterilmektedir. İki menkul kıymetin arasındaki korelasyon katsayısı (-1) ise getirilerinin ters yönlü ve aynı oranda hareket ettiği, dolayısıyla birisinin getirisi düştüğünde diğerinin getirisinin aynı oranda artacağı söylenir.



Şekil 5.5: Sıfır korelasyon

Şekil 5.5’de ise aralarında negatif veya pozitif yönlü bir ilişki olmayan korelasyon katsayısı sıfır olan iki menkul kıymetin getirilerinin karşılıklı olarak hareketi gösterilmiştir. Menkul kıymet getirileri arasındaki hareketin yönü bakımından aynı yönlü veya ters yönlü anlamlı bir ilişki olmadığı söylenir (Sevinç, 2007, s. 35-36).

5.3 Portföy Teorileri

Portföy teorilerinde finansal varlıklardan oluşan portföyün beklenen getirisini artırmanın yanında toplam riski düşürmek amacıyla çeşitlendirme yoluna gidilmiştir. Bu amaçla geliştirilen iki temel teori vardır. Bunlardan birincisi; 1950’li yıllara kadar varlıklar arasındaki ilişkiyi gözetmeden tesadüfî olarak seçilen menkul kıymetlerin portföye alınması temeline dayanan ve basit çeşitlendirme düşüncesiyle hareket ederek oluşturulan Geleneksel Portföy Teorisi (GPT)’dir. Diğeri ise, 1950’lerden sonra ortaya çıkan ve varlıklar arasındaki ilişkileri de göz önüne alarak portföy oluşturma düşüncesine dayanan Modern Portföy Teorisi (MPT)’dir.

5.3.1 Geleneksel Portföy Teorisi

Geleneksel Portföy Teorisi, portföy içindeki varlıkların arasındaki ilişkiye bakmaksızın varlık sayısının arttırılması (yalın çeşitlendirme) ile toplam riskin azaltılabileceği görüşüne dayanır. Birden fazla varlığa yatırım yapılması görüşüne dayandığı için “bütün yumurtaların aynı sepete konulmaması” şeklinde yorumlanabilir. Geleneksel portföy yaklaşımına göre, portföy yönetimi bir bilim değil, sanattır. Bu yaklaşımın amacı, yatırımcının sağlayacağı faydayı maksimize etmektir. Öte yandan, çeşitli portföy getirilerine göre ortaya çıkabilecek riskler de hesaplanmalıdır. Portföy oluşturma asıl amacı, riskin dağıtılmasıdır. Portföyü oluşturan menkul kıymetlerin getirileri aynı yönde hareket etmeyeceğinden, portföyün riski tek bir menkul kıymetin riskinden küçük olacaktır. İşte geleneksel portföy teorisi, bu prensipten hareketle, portföy içerisindeki menkul kıymet sayısının arttırılması ilkesine dayanır (Korkmaz ve Ceylan, 2006, s. 505).

Çeşitlendirme ile riskin azaltılabileceği yönündeki çalışmaların en önemlilerinden biri J. L. Evans ve S. H. Archer tarafından 1968 yılında yapılan çalışmadır. 1958-67 yılları arasını kapsayan bu çalışmada, varlık sayısı 10 ile 15 arasında olduğunda portföyün toplam riski içindeki sistematik olmayan (çeşitlendirme ile giderilebilen) riskin, çeşitlendirme etkisiyle hemen hemen giderilebileceği, dolayısıyla portföyün toplam riskinin sistematik risk (çeşitlendirme ile giderilemeyen) seviyesine indirilebileceği ortaya çıkarılmıştır. Gökçe (2001) tarafından İMKB-30 Endeksi üzerine 2000 yılının ilk altı ayını içerecek şekilde yapılan diğer bir çalışmada ise iyi çeşitlendirilmiş bir portföyün 6 ile 16 arasında menkul kıymet içermesi gerektiği sonucuna varılmıştır. Al Suqaiier ve Al Ziyud (2011) Amman Stock Exchange –ASE (Amman Borsası)’nda 2005-2010 yılları arasında işlem gören 100 şirketin haftalık kapanış fiyatlarını ele alarak yaptıkları çalışmada 15-16 adet hisse içeren portföylerde çeşitlendirme yoluyla kâr elde edildiğini göstermişlerdir. Ayrıca İskenderoğlu ve Karadeniz (2011), İMKB-30 hisselerinin 02.01.2009 – 31.12.2009 tarihleri arasındaki günlük kapanış fiyatlarını ele alarak yaptıkları çalışmada, optimal çeşitlendirmede İMKB-30 ve İMKB-100’den daha düşük riskli bir portföy oluşturabilmek için 5-10 arası hisse senedine yatırımın yeterli olduğunu göstermişlerdir.

5.3.2 Modern Portföy Teorisi

Menkul kıymetler arasındaki ilişki göz ardı edilerek portföy oluşturma felsefesine dayanan Geleneksel Portföy Teorisinde, düşük getirili menkul kıymetlerin de portföye alınmış olması olasılığı artacağı için, bu yöntemde her ne kadar risk azaltılsa da elde edilen getirinin de az olması sorunu ortaya çıkmaktadır. Bu sorun karşısında Modern Portföy Teorisinin (MPT) kurucusu olarak kabul edilen ve ilk önce 1952 yılında makale olarak yayınlayıp daha sonra kitap haline getirdiği (1959) çalışması ile bu teorinin temelini atan Harry Markowitz, etkin varlıkların kombinasyonlarının belirlenmesi olarak açıklanan ve bu varlıklar arasındaki getiri ilişkisine dayalı bir portföy seçim modelinden bahsetmiştir. Markowitz’e göre sadece çeşitlendirme ile riski azaltmak mümkün değildir.

Modern Portföy Teorisi (Portföy Seçimi) normatif bir teoridir. Pozitif teoriler gerçek yatırımcı davranışlarını göz önüne alırken, normatif teoriler, yatırımcıların nasıl davranması gerektiğini belirtir. Bu yönüyle, normatif bir teori olarak MPT, optimum portföy oluşturmak amacıyla yatırımcının takip etmesi gereken standart veya normal davranışı açıklamaktadır (Fabozzi, Gupta and Markowitz, 2002, s. 7; Genel, 2004, s. 21).

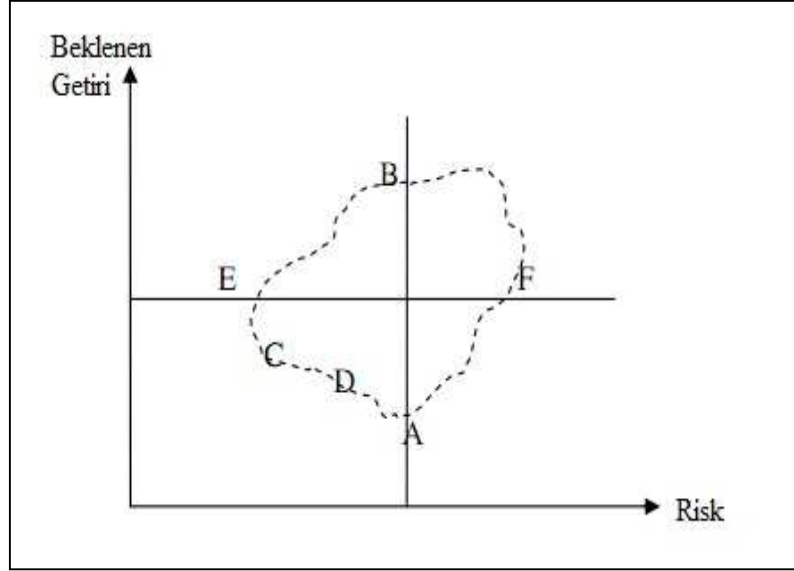
Bu modele göre varlıklar arasındaki getiri korelasyonu (korelasyon katsayıları) incelenerek tam pozitif ilişki içinde olmayan menkul değerlerin yani, aralarındaki korelasyon katsayısı 1'den küçük olan ($\delta_{A,B} < 1$) varlıkların portföye dahil edilmesiyle portföyün getirisini düşürmeden riskin azaltılabileceği gösterilmiştir (Markowitz, 1952, s. 89).

5.3.2.1 Markowitz (Ortalama-Varyans) Modeli

Modern Portföy Kuramının kurucusu olan Harry Markowitz'in yaklaşımının temel çerçevesi bir yatırımcının bugün sahip olduğu belirli bir tutarı çeşitli menkul değerlere yatırarak bir dönem elinde tutması oluşturmaktadır. Bu yaklaşım, yatırımcının muhtemel portföylerden seçeceği menkul değerlerden meydana gelen bir portföye dayanmaktadır. Finans literatüründe Portföy Seçim Problemi olarak da bilinmektedir.

Bu bölümde Markowitz'in ortaya koyduğu portföy seçimi problemi incelenecektir. Markowitz, belirli bir parayı menkul değerlere yatıran yatırımcının dönem sonunda elde edebileceği meblağı bilmesinin mümkün olmadığını söyleyerek, yatırımcının hisse senedinin geçmişteki performansından yararlanarak bazı tahminlerde bulunabileceğini belirtmiştir. Bu aşamada yatırımcının dayanabileceği iki değişken söz konusudur. Bunlardan birincisi, menkul değer beklenen getirisi, diğeri ise menkul değer riskidir. Yatırımcılar beklenen getirilerini yükseltmek isterken, elde edecekleri getiri ile ilgili belirsizlikleri (risk) azaltmak isterler. Bununla birlikte aynı getiri düzeyinde daha az riski tercih ederler. Daha basit bir ifade ile bir taraftan getirilerini maksimize etmek isterken diğeri taraftan da risklerini minimize etmek isterler (Karan, 2004, s.135).

Bu iki yaklaşıma göre aynı risk düzeyinde B portföyünü veya varlığını, A varlığı veya portföyüne; aynı getiri düzeyinde ise E portföyünü veya varlığını, F varlığı veya portföyüne tercih edecektir (Bkz. Şekil 5.6).



Şekil 5.6: Farklı varlık veya portföyler için getiri ve risk olanakları (Elton, Gruber, Brown and Goetzmann, 2009, s. 80).

Markowitz'in MPT'yi ortaya koyarken kullandığı birtakım varsayımlar vardır. Bu portföy seçim teorisi şu temel varsayımlara dayanmaktadır (Karan, 2004, s.136):

- Yatırımların getirileri yatırımların çıktısı olarak ifade edilebilir.
- Yatırımcının risk tahmini, varlıkların ya da portföyün getirilerinin varyansı ile orantılıdır.
- Yatırımcılar kararlarını verirken sadece beklenen getiri ve getirinin varyansını model parametreleri olarak kullanmaya razıdırlar.
- Yatırımcı riskten kaçma eğilimi göstermektedir. Herhangi bir beklenen getiri düzeyinde, ulaşabileceği minimum riski, herhangi bir risk düzeyinde de ulaşabileceği maksimum getiriyi seçecektir.
- Yatırımcılar aynı ve tek dönemlik yatırım ufkuна sahip olmakla birlikte, yatırımların beklenen getirilerini olasılık dağılımları olarak ifade ederler.

5.4 Portföyün (Varlık Kombinasyonlarının) Getirisi ve Riski

Bölüm 5.2.2.'de bir varlığın beklenen getirisinin tüm senaryolardaki getirilerin ağırlıklı ortalamasına eşit olduğu açıklanmıştı. Bir portföyün beklenen getirisi için de benzer tanımlı vermek mümkündür. Bir portföyün beklenen getirisi, portföyü oluşturan menkul kıymetlerin beklenen getirilerinin ortalaması olarak verilir.

Portföyü oluşturan menkul değerlerin beklenen getirilerinin portföy içindeki ağırlıkları toplamı 1'e eşittir. Yani n adet menkul değerden oluşan bir portföyü göz önüne aldığımızda i . menkul değer portföydeki ağırlığı w_i olarak alınır,

$$w_1 + w_2 + \dots + w_n = \sum_{i=1}^n w_i = 1 \quad (5.16)$$

elde edilir. O halde R_p , portföy getirisi rassal değişkenini göstermek üzere, portföy getirisinin beklenen değeri:

$$E[R_p] = \mu_p = \sum_{i=1}^n w_i \cdot E[R_i] = \sum_{i=1}^n w_i \cdot \mu_i \quad (5.17)$$

olarak bulunur. Burada;

n : Portföydeki menkul kıymet sayısını,

$E[R_i] = \mu_i$: i . menkul kıymetin beklenen değerini,

w_i : i . menkul değer portföydeki ağırlığını göstermektedir.

Modern Portföy Teorisine göre portföyün riski varyansı ya da standart sapması ile ölçülür. Portföyün riski, beklenen getiride olduğu gibi portföyü oluşturan menkul değerlerin ağırlıklı ortalaması değildir. Portföyün riski, portföyü oluşturan menkul değerlerin standart sapmalarının (risk) ağırlıklı ortalamalarından daha küçük bir değerdir. Menkul değerlerin ayrı ayrı riskleri birbirlerini ortadan kaldırarak portföy riskini azaltmaktadır. Portföy riskinin de hesaplanmasında her bir menkul kıymetin ortalama risklerinin toplamından yola çıkılırsa, menkul

kıymetlerin kendi aralarındaki ilişki ihmal edilmiş olacaktır. Bu nedenle her bir menkul kıymet kombinasyonunun getirileri arasındaki kovaryans değeri hesaba katılmalıdır (Karan, 2004, s. 143):

$$Var(R_p) = \sigma_p^2 = E\left[\left\{w^T (R_p - \mu_p)\right\}^2\right] = \sum_{i=1}^n w_i^2 \sigma_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij}, \quad (i \neq j) \quad (5.18)$$

veya

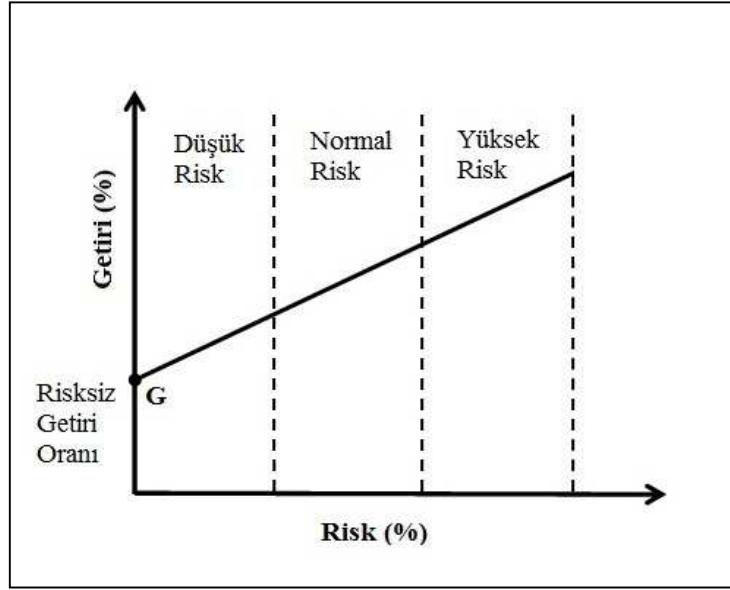
$$Var(R_p) = \sigma_p^2 = \sum_{i=1}^n w_i^2 \sigma_i^2 \quad (5.19)$$

$$\sigma_{ij} = E\left[\{R_i - \mu_i\}\{R_j - \mu_j\}\right]$$

5.4.1 Risk ve Beklenen Getiri İlişkisi

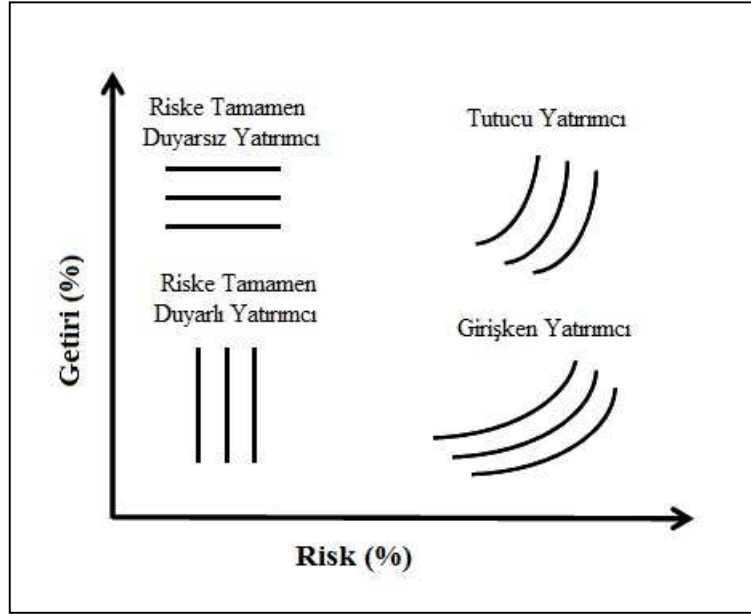
Risksiz bir yatırımda beklenen getirinin elde edilmesi konusunda bir güvence söz konusudur. Yatırımcılar riskli yatırım araçlarına yatırım yaptıklarında ise beklenen getirinin elde edilememesi ya da yapılan yatırımın kaybedilmesi olasıdır. Beklenen getiride belirsizlik, yani yatırımın riski arttıkça, yatırımcıların talep edecekleri getiri oranı da artacaktır. Yatırımcılar daha fazla risk üstlenmeyi ancak daha fazla getiri beklentisi ile kabul edecektir. Risk ve getiri arasında bu genel ilişkiyi gösteren piyasa doğrusu Şekil 5.7’de gösterilmiştir.

Şekilde G noktası risksiz getiri oranını, piyasa doğrusunun eğimi ise risk birimi başına talep edilen ek getiriyi göstermektedir. Bu doğru, risk ile getiri arasındaki genel ilişkiyi açıklamaktadır. Ancak, tüm yatırımcıların riske karşı aynı duyarlılığı göstermesini söyleyemeyiz. Bazı yatırımcılar kazandıkları paranın değeri yüksek olduğu için riskten kaçınma eğiliminde olurlar. Bazıları ise daha yüksek getiri beklentisiyle risk almaktan hoşlanırlar (Genel, 2004, s. 10).



Şekil 5.7: Risk getiri değişimi (Piyasa Doğrusu)

Yatırımcıların risk ve getiri tercihleri arasındaki ilişki farklılık (kayıtsızlık) eğrileri ile gösterilmektedir. Farklılık eğrileri bir yatırımcı için farklı beklenen getiri ve riske sahip olmakla beraber aynı düzeyde fayda sağlayan yatırımları birleştiren eğrilerdir. Bu eğriler yatırımcıların aldıkları risk karşısındaki getiri beklentilerini göstermektedir. Şekil 5.8’de gösterildiği gibi, ikisi uç durumları temsil etmek üzere, risk-getiri değişimi karşısında dört yatırımcı tipi gösterilmektedir.



Şekil 5.8: Yatırımcı farksızlık eğrileri (Özçam, 1997, s. 13)

Riske Tamamen Duyarlı Yatırımcı: Bu tür yatırımcılar, yatırım kararlarını sadece risk düzeyine göre belirlemektedir. Dolayısıyla da, belirli bir risk düzeyi, getiri ne olursa olsun aynı faydayı sağlamaktadır. Bu durum, getiri eksenine paralel farksızlık eğrisiyle gösterilmektedir.

Riske Tamamen Duyarsız Yatırımcı: Farksızlık eğrisi risk eksenine paralel olan böyle bir yatırımcı için yatırım kararları beklenen getiri düzeyine göre belirlenmektedir. Belirli bir getiri düzeyi, riskten bağımsız olarak, aynı faydayı sağladığı için, bu getirinin hangi risk düzeyinde sağlandığı önemli değildir.

Riske tamamen duyarlı ya da tamamen getiriye odaklanmış (riske duyarsız) yatırımcı tipleri gerçekçi değildir. Gerçek hayatta yatırımcıların risk-getiri tercihleri bu iki uç noktanın arasında bir yerdedir.

Tutucu Yatırımcı: tutucu bir yatırımcı açısından yatırımın riski beklenen getirisine göre daha önemlidir. Bunun sonucu olarak, bu tip bir yatırımcı, küçük bir risk artışını ancak daha büyük bir getiri artışı karşılığında kabullenecektir.

Girişken Yatırımcı: Tutucu yatırımcının aksine olası bir getiri artışı için daha büyük miktarlarda risk üstlenme eğiliminde olan yatırımcı tipidir. Bu tür bir yatırımcı için elde edilecek getiri riskten daha önemlidir (Özçam, 1997, s. 13-14).

Gerçekçi olmayan yatırımcı tipleri dikkate alınmazsa, yatırımcıların aynı fayda düzeyini koruyabilmeleri için risk-getiri değişimini gözetmeleri gerekir. Normal bir yatırımcı daha fazla bir riski ancak daha fazla bir getiri beklentisi ile kabul edecektir. Bununla beraber ne kadar riski kabul edeceği yatırımcı için önemli bir karardır. Herkesin risk kabul etme derecesi farklıdır ve kişinin risk toleransı sabit olmayıp hayat boyunca değişme eğilimindedir (Genel, 2004, s. 12).

5.4.2 İki Varlıktan Oluşan Portföyün Beklenen Getirisi ve Riski

Herhangi iki menkul kıymetten oluşan portföyün bir portföyün beklenen getirisi, bilindiği gibi varlıkların beklenen getirileri ile portföy içerisindeki ağırlıkları çarpımlarının toplamıdır.

$$E[R_p] = \mu_p = \sum_{i=1}^2 w_i \cdot E[R_i] = \sum_{i=1}^2 w_i \cdot \mu_i = w_1 \mu_1 + w_2 \mu_2 \quad (5.20)$$

İki varlıktan oluşan bir portföyün varyansı da aşağıdaki gibi hesaplanır:

$$\begin{aligned} \sigma_p^2 &= Var(R_p) = Var(w_1 R_1 + w_2 R_2) = E\left[\left((w_1 R_1 + w_2 R_2) - E[w_1 R_1 + w_2 R_2]\right)^2\right] \\ &= E\left[\left(w_1 (R_1 - \mu_1) + w_2 (R_2 - \mu_2)\right)^2\right] \\ &= E\left[w_1^2 (R_1 - \mu_1)^2 + w_2^2 (R_2 - \mu_2)^2 + 2w_1 w_2 (R_1 - \mu_1)(R_2 - \mu_2)\right] \\ &= w_1^2 E\left[(R_1 - \mu_1)^2\right] + w_2^2 E\left[(R_2 - \mu_2)^2\right] + 2w_1 w_2 E\left[(R_1 - \mu_1)(R_2 - \mu_2)\right] \\ &= w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2w_1 w_2 \sigma_{12} \\ &= w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2w_1 w_2 Cov(R_1, R_2) \end{aligned} \quad (5.21)$$

Her iki tarafın karekökü alındığında portföy standart sapması

$$\sigma_p = \sqrt{w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2w_1 w_2 \sigma_{12}} \quad (5.22)$$

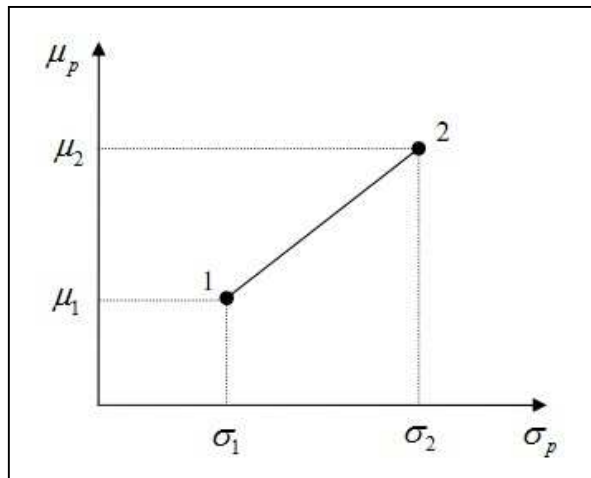
şeklinde bulunur.

Şimdi portföy riskini bu iki menkul kıymet arasındaki olası korelasyon durumlarına göre hesaplayalım. Varsayalım ki bu iki değer arasında tam pozitif korelasyon ($\delta_{1,2} = +1$) olsun. $\sigma_{12} = Cov(R_1, R_2) = \sigma_1 \sigma_2 \delta_{1,2}$ olduğundan $\sigma_{12} = Cov(R_1, R_2) = \sigma_1 \sigma_2 \delta_{1,2} = \sigma_1 \sigma_2$ elde edilir. Bu değeri (5.22) de yerine yazarsak

$$\begin{aligned}\sigma_p &= \sqrt{w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2w_1 w_2 \sigma_1 \sigma_2} \\ &= \sqrt{(w_1 \sigma_1 + w_2 \sigma_2)^2} \\ &= w_1 \sigma_1 + w_2 \sigma_2\end{aligned}\quad (5.23)$$

bulunur. Görüldüğü gibi, iki varlık arasındaki korelasyon katsayısının +1 olması durumunda portföy riski iki menkul kıymetin risklerinin ağırlıklı ortalamasının toplamı olarak bulunmuştur ki, bu da portföy etkisinin olmadığını gösterir. Bu durumda menkul kıymet getirileri aynı yönde ve aynı oranda hareket ettiği için çeşitlendirmenin faydasından yararlanılamamakta ve portföy riski sınırlandırılmamaktadır.

Portföyü oluşturan menkul değerlerin getirileri arasındaki ilişkinin tam pozitif yönlü olması durumunda portföy riski Şekil 5.9'da görüldüğü gibi iki menkul kıymetin riskleri arasında bir yerde olacaktır.

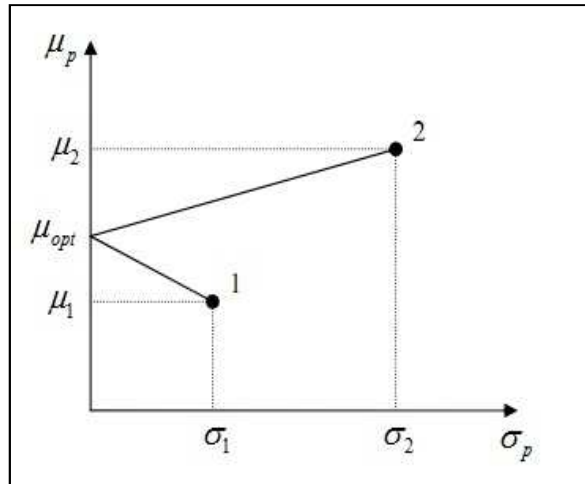


Şekil 5.9: Aralarında tam pozitif korelasyon olan iki menkul kıymetin risk-getiri grafiği

İki menkul kıymet arasında tam negatif korelasyon ($\delta_{1,2} = -1$) olması durumunda portföy riski; $\sigma_{12} = Cov(R_1, R_2) = \sigma_1\sigma_2\delta_{1,2} = -\sigma_1\sigma_2$ eşitliğinden dolayı

$$\begin{aligned}\sigma_p &= \sqrt{w_1^2\sigma_1^2 + w_2^2\sigma_2^2 - 2w_1w_2\sigma_1\sigma_2} \\ &= \sqrt{(w_1\sigma_1 - w_2\sigma_2)^2} \\ &= w_1\sigma_1 - w_2\sigma_2\end{aligned}\quad (5.24)$$

olarak bulunur. Görüldüğü gibi, aralarında tam negatif (ters) yönlü korelasyon olan iki menkul değerden oluşan portföyün riski minimuma indirilmiş olacaktır. Aralarında tam negatif (ters) yönlü korelasyon olan iki menkul değer farklı ağırlık oranlarında bileşimi ile oluşan portföylerin getiri-risk değerlerindeki değişim Şekil 5.10'da olduğu gibidir. Buna ilaveten, belirlenen uygun ağırlık oranlarında bir portföy oluşturulması halinde portföyün riski, matematiksel olarak sifıra indirilebilmektedir.

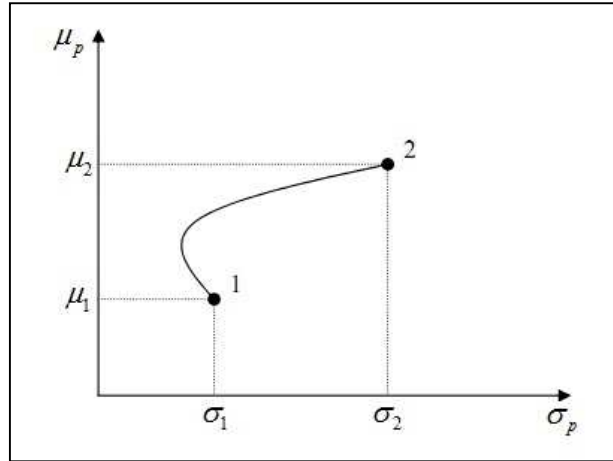


Şekil 5.10: Aralarında tam negatif korelasyon olan iki menkul kıymetin risk-getiri grafiği

İki menkul kıymetin getirileri arasındaki korelasyon katsayısının sıfır ($\delta_{1,2} = 0$) olduğu durumda portföy riski; $\sigma_{12} = Cov(R_1, R_2) = \sigma_1\sigma_2\delta_{1,2} = 0$ eşitliğinden dolayı

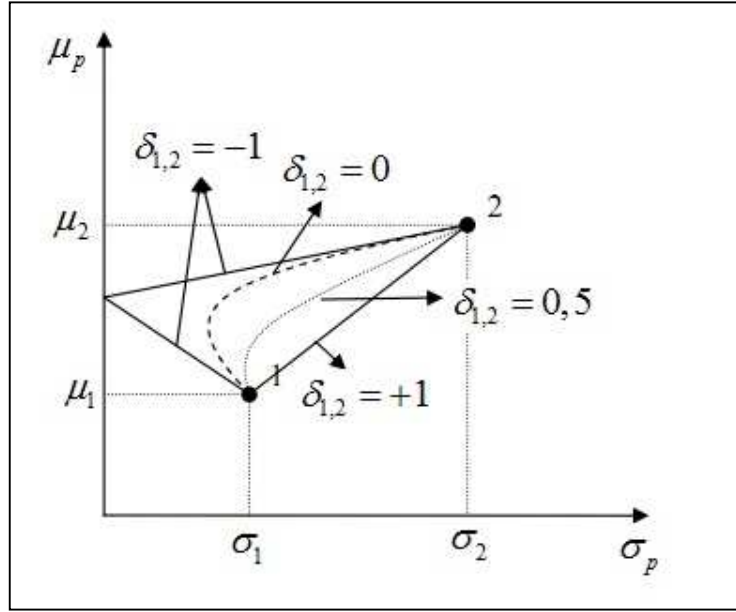
$$\begin{aligned}\sigma_p &= \sqrt{w_1^2\sigma_1^2 + w_2^2\sigma_2^2 - 2w_1w_2\sigma_1\sigma_2} \\ &= \sqrt{w_1\sigma_1 + w_2\sigma_2}\end{aligned}\quad (5.25)$$

elde edilir. Korelasyon katsayısı sıfır olduğunda portföyün riski, Şekil 5.11’de olduğu gibi menkul kıymetlerin farklı ağırlık oranlarıyla birleşiminde oluşacak farklı portföylerin elde edilen riskleri ve bu riskleri karşılığında beklenen getirileri, tam pozitif korelasyon ve tam negatif korelasyon durumunda oluşan portföy riskleri ve beklenen getirilerinin gösterildiği çizgiler arasında, birinci menkul kıymet ile ikinci menkul kıymet arasında yer alan çizgi üzerinde oluşacaktır. Yine portföyün riski belli bir ağırlık bileşiminin olduğu noktaya kadar azalmakla birlikte belli bir ağırlık bileşiminden sonra yakaladığı minimum risk seviyesinden uzaklaşacaktır.



Şekil 5.11: Aralarında sıfır korelasyon olan iki menkul kıymetin risk-getiri grafiği

Farklı beklenen getirilere ve standart sapmalara sahip menkul kıymetlerin değişik ağırlık bileşimleriyle oluşturulan portföyün beklenen getirisi ve riski, menkul kıymetler arasındaki korelasyon katsayısının durumuna göre değişik varyasyonlarda olabilir. Bu durumlar genel olarak şu şekilde gösterilebilir (Dobbins vd, 1994, s. 31):



Şekil 5.12: Aralarında çeşitli korelasyon olan iki menkul kıymetin risk-getiri grafiği

5.4.3 N Varlıktan Oluşan Portföyün Beklenen Getirisi ve Riski

Oluşturulan portföyün ikiden fazla finansal varlık içermesi durumunda ise beklenen getiri, varlıkların beklenen getirileri ile portföy içerisindeki ağırlıklarının çarpımları toplamına eşittir. N adet menkul değer söz konusu olduğu bir portföyün getiri ve riskinin hesaplanması için N adet getiri hesabı, N adet standart sapma, $N(N-1)/2$ adet kovaryans değeri olmak üzere toplam $(N^2 + 3N)/2$ adet veri hesaplamak gerekmektedir. Buna göre N adet menkul değerden oluşan bir portföyün beklenen getirisi şu şekilde hesaplanır:

$$E[R_p] = \mu_p = E\left[\sum_{i=1}^N w_i \cdot R_i\right] = \sum_{i=1}^N E[w_i \cdot R_i] = \sum_{i=1}^N w_i E[R_i] = \sum_{i=1}^N w_i \mu_i \quad (5.26)$$

N adet menkul değerden oluşan bir portföyün riski;

$$\begin{aligned} \sigma_p^2 = \text{Var}(R_p) &= \sum_{i=1}^N w_i^2 \sigma_i^2 + \sum_{i=1}^N \sum_{\substack{k=1 \\ k \neq i}}^N w_i w_k \text{Cov}(R_i, R_k) \\ &= \sum_{i=1}^N w_i^2 \sigma_i^2 + 2 \sum_{i=1}^{N-1} \sum_{k=i+1}^N w_i w_k \text{Cov}(R_i, R_k) \end{aligned} \quad (5.27)$$

ile bulunmaktadır (Dobbins vd, 1994, s. 25). Bu formülü daha geniş formda yazarsak;

$$\begin{aligned}\sigma_p^2 = Var(R_p) = & w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + w_3^2\sigma_3^2 + \dots + w_N^2\sigma_N^2 \\ & + 2w_1w_2Cov(R_1, R_2) + 2w_1w_3Cov(R_1, R_3) + 2w_2w_3Cov(R_2, R_3) + \dots \\ & + 2w_{N-1}w_NCov(R_{N-1}, R_N)\end{aligned}\quad (5.28)$$

elde edilir. Bu formül aynı zamanda

$$\sigma_p^2 = Var(R_p) = \sum_{i=1}^N \sum_{k=1}^N w_i w_k Cov(R_i, R_k) \quad (5.29)$$

veya

$$\sigma_p^2 = Var(R_p) = \sum_{i=1}^N \sum_{k=1}^N w_i w_k \sigma_i \sigma_k \delta_{i,k} \quad (5.30)$$

şeklinde de ifade edilebilir.

N sayıda menkul kıymetten meydana gelen portföyün varyansının veya standart sapmasının bulunmasında kullanılan formülün hesaplanabilmesi için her bir menkul kıymetin yer aldığı N -tane satır ve N -tane sütundan oluşan bir matris oluşturulur ve her bir hücrenin içine varyans-kovaryans değerlerini hesaplayan formül yazılıp daha sonra bu değerlerin toplamı alındığında portföyün varyansı elde edilir. Buna göre ağırlıklandırılmış varyans-kovaryans matrisi Tablo 5.3'te verildiği gibidir.

Tablo 5.3: Ağırlıklandırılmış varyans-kovaryans matrisi

	1	2	3	N
1	$w_1 w_1 Cov(R_1, R_1)$ veya $w_1 w_1 \sigma_1 \sigma_1 \delta_{1,1}$	$w_1 w_2 Cov(R_1, R_2)$ veya $w_1 w_2 \sigma_1 \sigma_2 \delta_{1,2}$	$w_1 w_3 Cov(R_1, R_3)$ veya $w_1 w_3 \sigma_1 \sigma_3 \delta_{1,3}$	$w_1 w_N Cov(R_1, R_N)$ veya $w_1 w_N \sigma_1 \sigma_N \delta_{1,N}$
2	$w_2 w_1 Cov(R_2, R_1)$ veya $w_2 w_1 \sigma_2 \sigma_1 \delta_{2,1}$	⋮
3	$w_3 w_1 Cov(R_3, R_1)$ veya $w_3 w_1 \sigma_3 \sigma_1 \delta_{3,1}$	⋮
⋮	⋮	⋮
⋮	⋮	⋮
N	$w_N w_1 Cov(R_N, R_1)$ veya $w_N w_1 \sigma_N \sigma_1 \delta_{N,1}$	$w_N w_N Cov(R_N, R_N)$ veya $w_N w_N \sigma_N \sigma_N \delta_{N,N}$

Portföy riskinin hesaplanması matrislerin çarpımı yardımıyla da ifade edilebilir. Buna göre portföyün varyansı;

$$\sigma_p^2 = [w_1 \ w_2 \ \dots \ w_N] \begin{bmatrix} Cov(R_1, R_1) & Cov(R_1, R_2) & \dots & Cov(R_1, R_N) \\ Cov(R_2, R_1) & Cov(R_2, R_2) & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ Cov(R_N, R_1) & \cdot & \dots & Cov(R_N, R_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ w_N \end{bmatrix}$$

şeklinde de gösterilebilir. Özel olarak iki menkul değerden oluşan bir portföy için kovaryans matrisi şu şekilde yazılabilir ve Σ ile sembolize edilebilir (Fermanian and Scaillet, 2005, s. 937).

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\delta_{1,2} \\ \sigma_2\sigma_1\delta_{2,1} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \quad (5.31)$$

Buna göre matris şeklinde ifade edilen N adet menkul kıymetten oluşan portföy varyansında yer alan kovaryans matrisini açarak yazarsak, kovaryans matrisi yerine;

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \cdot & \dots & \cdot \\ 0 & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & \cdot & \cdot & \dots & \sigma_N \end{bmatrix} \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \dots & \delta_{1,N} \\ \delta_{2,1} & \delta_{2,2} & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \delta_{2,N} & \cdot & \dots & \delta_{N,N} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \cdot & \dots & \cdot \\ 0 & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & \cdot & \cdot & \dots & \sigma_N \end{bmatrix} \quad (5.32)$$

matrisleri yazılabilir. Bu matrisleri kullanarak portföy varyansı;

$$\sigma_p^2 = w \Sigma w' \quad (5.33)$$

şeklinde de gösterilebilir. (5.33) eşitliğinde w' ifadesi ağırlık matrisinin transpozisini göstermektedir.

Burada elde edilen portföy riskinin çeşitleme yoluyla azaltılmasını matematiksel olarak görmek de mümkündür. Bir portföyün varyansını ifade eden (5.27) eşitliğini göz önünde bulunduralım. Portföydeki tüm varlıkların bağımsız olduğunu ve bu sebeple menkul kıymetler arasındaki korelasyon katsayısının sıfır olduğunu varsayalım. Bu durumda (5.27) eşitliği

$$\sigma_p^2 = Var(R_p) = \sum_{i=1}^N w_i^2 \sigma_i^2 \quad (5.34)$$

halini alacaktır. Bir varsayım daha yaparak, portföy içerisinde yer alan menkul değerlerin portföyde eşit ağırlığa sahip olduğunu düşünelim. Bu durumda n sayıda menkul değer yer aldığı portföyde w_i terimi yerine $1/n$ yazabiliriz. Böylece (5.34) eşitliği

$$\sigma_p^2 = Var(R_p) = \sum_{i=1}^N \left(\frac{1}{n}\right)^2 \sigma_i^2 \quad (5.35)$$

veya

$$\sigma_p^2 = Var(R_p) = \frac{1}{n} \sum_{i=1}^N \left(\frac{\sigma_i}{n}\right)^2 \quad (5.36)$$

haline gelecektir. Son eşitlikte görüldüğü gibi n arttıkça portföy varyansı azalacaktır. Diğer bir ifadeyle portföyde bulunan menkul kıymet sayısının artması ile portföyün riski azalacaktır.

Ancak daha gerçekçi davranmak gerekirse, portföydeki menkul kıymetlerin birbirinden tamamıyla bağımsız olduğunu öne süremeyiz. Çünkü biliyoruz ki, ekonomik dalgalanmalar neticesinde genel olarak menkul kıymetlerin getirileri hep beraber belirli bir oranda yükselmekte veya düşebilmektedir. Bu durum bize menkul kıymetler arasında belirli oranlarda korelasyon olabileceğini göstermektedir. Menkul kıymetler arasında korelasyon var iken çeşitlemenin etkisini görebilmek amacı ile yine eşit oranda ve n sayıda menkul değerden oluşan bir portföye yatırım yapıldığını varsayalım. (5.27) ifadesinde yine w_i yerine $1/n$ yazarsak;

$$\sigma_p^2 = Var(R_p) = \sum_{i=1}^N \left(\frac{1}{n}\right)^2 \sigma_i^2 + \sum_{i=1}^N \sum_{\substack{k=1 \\ k \neq i}}^N \left(\frac{1}{n}\right) \left(\frac{1}{n}\right) Cov(R_i, R_k) \quad (5.37)$$

elde ederiz. Bu ifadeyi şu şekilde yazmak da mümkündür:

$$\sigma_p^2 = Var(R_p) = \frac{1}{n} \sum_{i=1}^N \left(\frac{\sigma_i}{n}\right)^2 + \left(\frac{n-1}{n}\right) \sum_{i=1}^N \sum_{\substack{k=1 \\ k \neq i}}^N \left(\frac{Cov(R_i, R_k)}{n(n-1)}\right) \quad (5.38)$$

Bu eşitlikte her iki terim de " n " sayıdaki menkul değere bölünerek bir ortalama değeri vermektedirler. Eşitliğin birinci terimi (5.36) ifadesiyle aynıdır. İkinci terimde i ve j sayılarının birbirine eşit olmaması nedeniyle $n(n-1)$ adet kovaryans vardır. Eşitliğin ikinci bölümü kovaryans toplamına bölünerek ortalama kovaryans elde edilecek ve sonuç olarak eşitlik,

$$\sigma_p^2 = \frac{1}{n} \overline{\sigma_i^2} + \left(\frac{n-1}{n} \right) \overline{Cov(R_i, R_k)} \quad (5.39)$$

haline gelecektir. Bu ifade ise bize, bireysel menkul kıymetlerin varyansının portföy varyansı üzerindeki etkisinin portföydeki menkul kıymet sayısı (n) arttıkça azaldığını, buna karşılık menkul kıymet sayısı arttıkça kovaryans teriminin etkisinin ortalamaya yaklaştığını göstermektedir. Böylelikle portföy oluşturarak bireysel menkul değerlerin risklerinin çeşitlendirme yolu ile azaltılabileceği, diğer taraftan menkul değerlerin kovaryansından kaynaklanan riski çeşitleme yöntemiyle azaltabilmenin mümkün olmadığı ortaya konulmaktadır (Karan, 2004, s. 155-156).

5.5 Optimum Portföy

Uygun (feasible) bir portföy, belirlenen kısıtları sağlayandır. Ortalama-varyans *etkin* (efficient) portföy ise bu uygun portföylerden, belirlenen (veya daha düşük) varyans değeri için maksimum beklenen getiriyi sağlayan ya da belirlenen (veya daha büyük) beklenen getiri için minimum varyansı elde eden portföydür. Beklenen değer ve varyans, etkin ortalama-varyans (EV) kombinasyonu diye isimlendirilen etkin portföy tarafından oluşturulur. İşte tüm etkin ortalama-varyans (EV) kombinasyonlarının oluşturduğu kümeye de “*etkin sınır*” (efficient frontier) denir ve portföy yöneticisinin amacı “etkin sınır üzerindeki noktaları belirlemektir” (Jakobs, Levy and Markowitz, 2005, s. 586).

N sayıda menkul kıymet bulunan bir portföyde, menkul kıymetlere değişik ağırlıklar verilerek sınırsız sayıda portföy oluşturulabilir. Bu nedenle yatırımcı, ya beklenen bir getiri oranı düzeyinde varyansı (ya da riski) minimize edecek bir portföyü seçecek ya da belirli bir varyans değerinde getirisi en yüksek olan portföyü seçecektir. Yani yatırımcılar portföy oluştururken hem beklenen getiriyi hem de riski göz önüne alır. Portföydeki her bir menkul kıymetin payı ise matematiksel olarak kuadratik modelleme ile hesaplanır (Gökgöz, 2006, s. 64).

mümkündür. Şekilde yine X_1 ve X_2 portföyleri aynı getiriye sağlarken, X_2 portföyünün risk durumu X_1 portföyüne göre daha fazladır. Dolayısıyla X_2 portföyü etkin bir portföy değildir. Böylece bu üç portföyden optimum olanı X_1 portföyüdür. Özetle, üst sınır üzerinde olan portföylere etkin portföy, bu sınıra da etkinlik sınırı denilmektedir (Gökgöz, 2006, s. 64-65).

6. UYGULAMA: BİR YAPAY SİNİR AĞI MODELİ İLE RİSK-GETİRİ TAHMİNİ VE OPTİMUM PORTFÖYÜN ELDE EDİLMESİ

Yapay sinir ağlarının finans alanına uygulamasının yapıldığı bu bölümde iki farklı uygulamaya yer verilmiştir. Bunlardan birincisi; portföylerin risk ve getirilerinin tahmin edilmesi, diğeri ise optimum (maksimum getiri ve minimum riske sahip) portföylerin oluşturulmasıdır.

6.1 Risk-Getiri Tahmini

Son yıllarda finans alanında yapıla gelen bilimsel çalışmalardan büyük bir kısmı menkul kıymetlerin veya menkul kıymetlerden oluşan çeşitli portföylerin gelecekteki fiyatlarının, dolayısıyla da getirilerinin tahmini üzerinde yoğunlaşmıştır (Tektaş ve Karataş, 2004, s. 344). Çalışmanın bu bölümünde de çalışmanın amacına yönelik olarak ele alınan verilerin bir yapay sinir ağı modeli geliştirilerek risk ve getiri tahmini yapılmış ve sonuçlar değerlendirilmiştir.

6.1.1 Uygulamanın Amacı

Yapay sinir ağları ile risk ve getiri tahmininin yapıldığı bu bölümde İMKB Ulusal-Sınai Endeksinde yer alan 140 şirketin (bkz. Tablo 6.1) 2010 yılı içerisindeki aylık kapanış (2. Seans) fiyatları ele alınarak getiri ve riskleri hesaplanmış, bu şirketlerle çeşitli kriterlere göre eşit ağırlıklı portföyler oluşturulmuştur. Bu kriterler; hisse senetlerine konu olan şirketlerin aktif büyüklükleri, piyasa değerleri, işlem hacimleri ve özsermayeleridir.

Çalışmanın uygulama kısmının bu bölümünde amaç, geliştirilen bir yapay sinir ağı modeliyle portföy risk ve getirisini tahmin etmek ve portföy oluşturmada belirtilen dört özellikten hangisi kullanılarak daha optimum (maksimum getirili - minimum riskli) portföylerin oluştuğunu görebilmektir.

Tablo 6.1: Araştırmada kullanılan hisse senetleri

Hisse Kodu	Hisse Simgesi	Hisse Kodu	Hisse Simgesi	Hisse Kodu	Hisse Simgesi	Hisse Kodu	Hisse Simgesi	Hisse Kodu	Hisse Simgesi
1	ALYAG	29	KRTEK	57	AKSA	85	BUCIM	113	KRDMA
2	AEFES	30	KORDS	58	ALKIM	86	CMBTN	114	KRDMB
3	BANVT	31	LUKSK	59	AYGAZ	87	CMENT	115	KRDMD
4	COLLA	32	MNDRS	60	BAGFS	88	CIMSA	116	SARKY
5	ERSU	33	MTEKS	61	BRISA	89	DENCM	117	ALCAR
6	FRIGO	34	SKTAS	62	DEVA	90	ECYAP	118	ASUZU
7	KENT	35	YATAS	63	DYOBY	91	EGSER	119	ARCLK
8	KERTV	36	YUNSA	64	ECILC	92	GOLTS	120	BFREN
9	KNFRT	37	ALKA	65	EGGUB	93	HZNDR	121	BSHEV
10	KRSTL	38	BAKAB	66	GOODY	94	IZOCM	122	DITAS
11	MERKO	39	DENTA	67	GUBRF	95	KONYA	123	EGEEN
12	PENGD	40	DOBUR	68	HEKTS	96	KUTPO	124	EMKEL
13	PETUN	41	DGZTE	69	MRSHL	97	MRDIN	125	EMNIS
14	PINSU	42	DURDO	70	PETKM	98	NUHCM	126	FMIZP
15	PNSUT	43	HURGZ	71	PTOFS	99	TRKCM	127	FROTO
16	SKPLC	44	IHGZT	72	PIMAS	100	USAK	128	GEREL
17	TATKS	45	IPEKE	73	SODA	101	UNYEC	129	IHEVA
18	TBORG	46	KAPLM	74	TRCAS	102	BRSAN	130	KARSN
19	TUKAS	47	KARTN	75	TUPRS	103	BURCE	131	KLMSN
20	ULKER	48	KOZAA	76	ADANA	104	BURVA	132	MUTLU
21	AKALT	49	TIRE	77	ADBGR	105	COMDO	133	OTKAR
22	ATEKS	50	OLMKS	78	ADNAC	106	CELHA	134	PARSN
23	ALTIN	51	VKING	79	AFYON	107	CEMTS	135	TUDDF
24	ARSAN	52	GENTS	80	AKCNS	108	DMSAS	136	TOASO
25	BOSSA	53	KLBM0	81	ANACM	109	ERBOS	137	PRKAB
26	DERIM	54	ADEL	82	BTCIM	110	EREGL	138	TTRAK
27	DESA	55	SERVE	83	BSOKE	111	FENIS	139	VESTL
28	IDAS	56	SASA	84	BOLUC	112	IZMDC	140	VESBE

Tablo 6.1’de yer alan hisse senetleri kullanılarak oluşturulan eşit ağırlıklı portföylerin İMKB Ulusal-Sınai Endeksi içerisindeki alt sektörlerde bulunan hisse senedi adetleri Tablo 6.2’ de gösterilmiştir. Burada kriter olarak alt sektörlerin imalat sektörü içindeki ağırlıkları baz alınmıştır. Örneğin, Tablo 6.2’de 1. portföy oluşturulurken her alt sektördeki en iyi 2 hisse senedi alınmıştır. 2. portföy oluşturulurken de her alt sektörün en iyi 3. ve 4. hisse senetleri (2 adet) alınmıştır. Benzer şekilde 6. portföy oluşturulurken 15 tane hisse senedi bulunan Metal Ana Sanayii’den en iyi 10. hisse senedi (1 adet), 26 adet hissesi bulunan Taş ve Toprağa Dayalı Sanayii’den en iyi 12., 13. ve 14. hisse senetleri (3 adet) alınmıştır. Diğer alt sektörlerden de tabloda belirtilen adette hisse senedine yer

verilmiştir. Ele alınacak olan tüm kriterlere göre oluşturulan portföyler (40 adet) bu minval üzere seçilmiştir.

Tablo 6.2: Portföylerin alt sektörlerdeki hisse adetleri

2010 Yılında İmalat Sanayiide İşlem Gören Şirketlerin Alt Sektörlere Göre Dağılım Oranları				Oluşturulan Portföylerin Alt Sektör İçindeki Hisse Senedi Adedi									
Sıra No	İmalat Sanayii Alt Sektörler	Hisse Senedi Adedi	Sektördeki Ağırlığı (%)	1. Portföy (Adet)	2. Portföy (Adet)	3. Portföy (Adet)	4. Portföy (Adet)	5. Portföy (Adet)	6. Portföy (Adet)	7. Portföy (Adet)	8. Portföy (Adet)	9. Portföy (Adet)	10. Portföy (Adet)
1	Gıda, İçki ve Tütün	20	14.3	2	2	2	2	2	2	2	2	2	2
2	Dokuma, Giyim Eşyası ve Deri	16	11.4	2	2	2	2	2	2	1	1	1	1
3	Orman ve Kağıt Ürünleri, Basım, Yayın ve Kırtasiye	19	13.6	2	2	2	2	2	2	2	2	2	1
4	Kimya, Petrol, Kauçuk ve Plastik Ürünler	20	14.3	2	2	2	2	2	2	2	2	2	2
5	Taş ve Toprağa Dayalı Sanayi	26	18.6	2	2	2	2	3	3	3	3	3	3
6	Metal Ana Sanayi	15	10.7	2	2	2	2	1	1	1	1	1	2
7	Metal Eşya, Makine ve Gereç Yapım	24	17.1	2	2	2	2	2	2	3	3	3	3
Toplam (Adet)		140	100	14	14	14	14	14	14	14	14	14	14

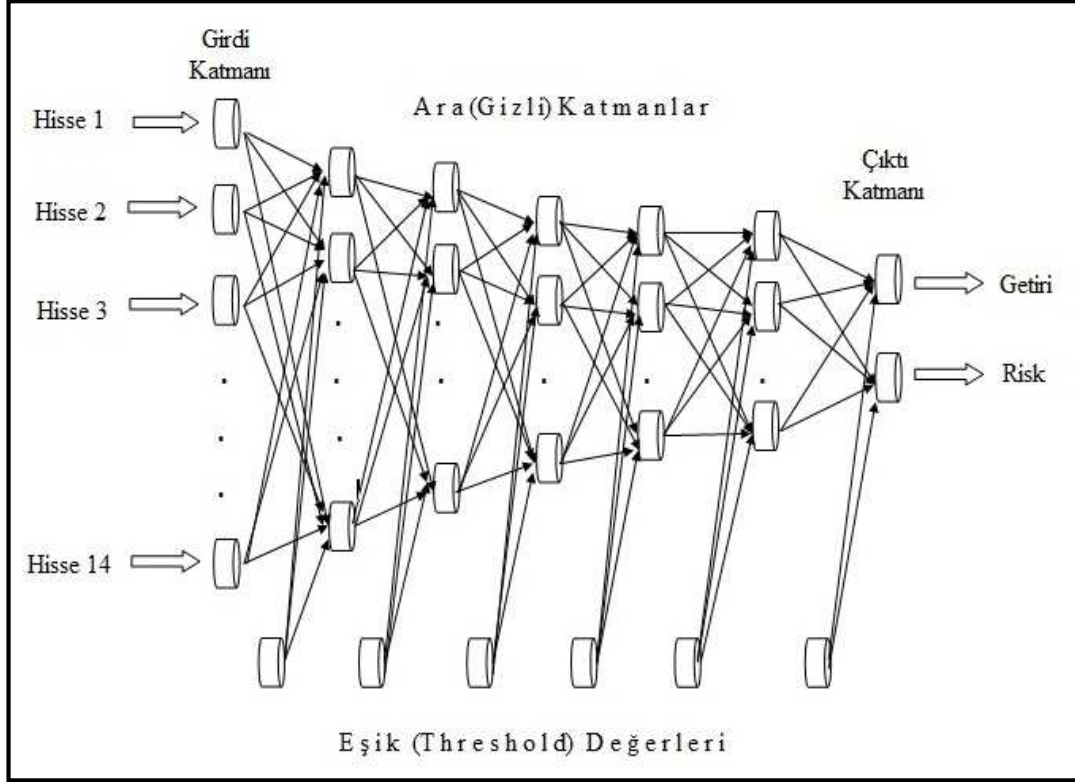
6.1.2 Veri Analizi

Çalışmada kullanılan hisse senetlerine ait aylık getiriler kullanılarak ve önceki bölümlerde anlatılan yöntemler göz önünde bulundurularak her bir hisse senedinin yıllık (2010 yılına ait) ortalama getirisi ve varyansı- standart sapması (riski) hesaplanmıştır. Hisse senetlerine ait veriler İstanbul Menkul Kıymetler Borsası'nın (İMKB) resmi internet sitesinden elde edilmiştir. Hisse senetlerine ait risk - getiri tablosu ve örnek olarak bir portföye ait varyans - kovaryans matrisi sırasıyla Ek-B ve Ek-C'de verilmiştir.

6.1.3 Risk-Getiri Tahmini İçin YSA Modelinin Geliştirilmesi

Çalışmada ileri beslemeli ve çok katmanlı bir yapay sinir ağı (MLP) modeli kullanılmıştır. Tablo 6.1'de yer alan 140 hisse senedi kullanılarak her bir portföyde 14 hisse senedi olacak şekilde 10 tane portföy oluşturulmuştur. Aktif büyüklük, piyasa değeri, işlem hacmi ve özsermaye kriterlerinden her birine göre 10 adet portföy oluşturulduğu için uygulamanın ilk kısmı (risk –getiri tahmini) toplam 40 portföy üzerinde yapılmıştır. Her gruptan ilk 9 portföy ile, geliştirilen yapay sinir ağı eğitilmiş ve 1 portföy de eğitilen ağın performansını ve tahminleme yeteneğini test etmek amacıyla kullanılmıştır.

İleri beslemeli ağın topolojisi 1 girdi katmanı, 5 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 14 (14 adet bağımsız girdi değişkenine karşılık olarak) 12, 10, 8, 6, 4 ve 2 (2 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, diğer katmanlarda ise tanjant sigmoid ve logistik sigmoid fonksiyonları kullanılmıştır. Bu çalışmanın risk-getiri tahmini aşamasında kullanılan yapay sinir ağının topolojisi Şekil 6.1'de görüldüğü gibidir.



Şekil 6.1: Risk ve getiri tahmini için oluşturulan MLP ağının topolojisi (Öztemel, 2012, s. 76; Karaođlan, 2011, s. 517).

Tablo 6.3'te aktif büyüklüklerine göre belirlenmiş eşit ağırlıklı portföylerin kodları ve bu portföylerin risk-getiri değerleri gösterilmiştir.

Ağın topolojisi belirlendikten sonra Tablo 6.3'de verilen girdi ve çıktılar, bütün değerler 1 den küçük olacak şekilde belirlenen bir katsayıya bölünerek kodlanmıştır. Eğitim algoritması olarak “**gradient descent**” algoritması kullanılmıştır. Bilindiği gibi YSA eğitimleri deneme yanılma metoduna dayanmaktadır. En uygun eğitim parametrelerinin belirlenebilmesi amacı ile deneme yanılma yöntemi kullanılarak ağ eğitilmiştir. Bu yolla eğitim parametrelerinden **öğrenme oranı** (lr) ve **momentum katsayısı** (mc)'nın hangi kombinasyonunun verilen ağ topolojisi için ortalama hata kareyi (mean square error- mse) minimize ettiği araştırılmıştır. Yapay sinir ağının eğitimi MATLAB 7.5.0'da gerçekleştirilmiştir.

Tablo 6.3: Aktif büyüklüğe göre portföyler matrisi

Aktif Büyüklüğe Göre Portföyler Matrisi																
Portföyler	Bağımsız Değişkenler (Girdiler)														Bağımlı Değişkenler (Çıktılar)	
	Hisse 1	Hisse 2	Hisse 3	Hisse 4	Hisse 5	Hisse 6	Hisse 7	Hisse 8	Hisse 9	Hisse 10	Hisse 11	Hisse 12	Hisse 13	Hisse 14	Getiri (%)	Risk
Portföy 1	20	3	30	23	43	45	71	70	99	81	110	102	127	121	4.1250	0.0037
Portföy 2	7	17	25	32	48	41	67	73	80	88	116	112	138	133	4.4833	0.0048
Portföy 3	15	13	34	22	49	50	61	62	98	82	113	114	135	130	4.4921	0.0059
Portföy 4	18	8	35	21	44	39	74	66	90	92	115	105	118	129	4.1850	0.0051
Portföy 5	16	19	36	24	47	37	63	60	95	85	101	107	134	137	4.8657	0.0090
Portföy 6	12	14	29	27	52	38	65	58	97	76	77	111	117	132	2.7271	0.0048
Portföy 7	10	9	28	51	54	69	72	78	84	83	108	131	123	120	4.9550	0.0072
Portföy 8	11	1	26	40	46	68	75	91	94	96	109	128	124	125	3.8254	0.0076
Portföy 9	6	5	31	42	53	59	64	100	79	89	106	122	126	119	8.9421	0.0415
Portföy 10	2	4	33	55	57	56	86	93	87	103	104	136	139	140	4.4850	0.0074

Aktif büyüklüğüne göre oluşturulan portföylerde risk ve getiri tahminine ilişkin yapay sinir ağı kodlaması aşağıdaki gibidir:

```

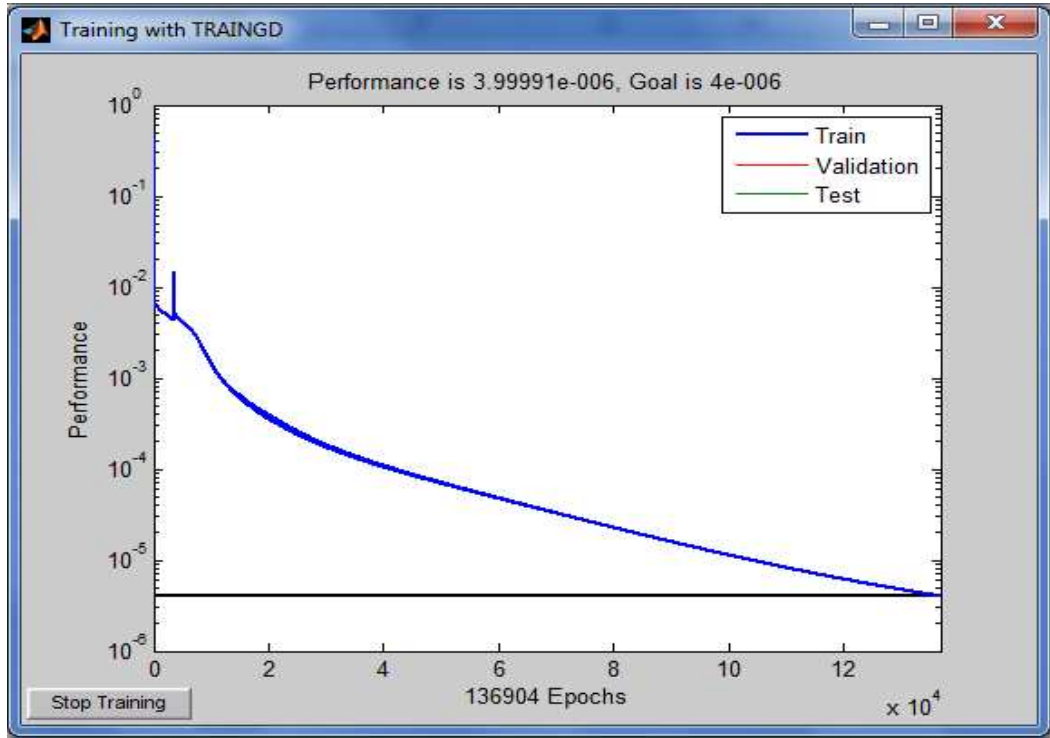
P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premnmx(P,T);
[S0,Q]=size(P);
S1=12; %1. Ara katman nöron sayısı
S2=10; %2. Ara katman nöron sayısı
S3=8; %3. Ara katman nöron sayısı
S4=6; %4. Ara katman nöron sayısı
S5=4; %5. Ara katman nöron sayısı
S6=2; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6],{'purelin',
'tansig','tansig','tansig','tansig','logsig','tansig','tansig'
},'traingd');
net.trainParam.epochs=200000;
net.trainParam.show=1000;
net.trainParam.goal=0.000004;
net.trainParam.mc=0.7; % Momentum coefficient
net.trainParam.lr=0.05; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save aktifbuyukluk net

```

Ağın eğitilmesinde öğrenme ve momentum katsayılarının belirlenmesi de ağın performansını yakından ilgilendirmektedir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağın dolaşması ve osilasyon yaşaması söz konusudur. Küçük değerler seçilmesi ise öğrenme zamanını artırmaktadır. Bu yüzden öğrenme katsayısı için genellikle 0.2 - 0.4 arasındaki değerler kullanılır. Fakat bu tamamen ilgili probleme bağlıdır (Öztemel, 2012, s. 99). Nitekim bu çalışmada ön denemeler sonucunda **öğrenme oranı (*lr*)**'nin 0.009 – 0.04 aralığında olmasına karar verilmiş ve ağ en iyi performansını 0.05 dolaylarında iken göstermiştir.

Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler. Momentum katsayısı, bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesi olarak görülmektedir. Bu katsayı, özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar vermesini sağlamak amacıyla önerilmiştir. Bu değer küçük olması yerel çözümlerden kurtulmayı

zorlaştırabilir. Çok büyük değerler ise tek bir çözüme ulaşmada sorunlar yaşatabilir. Tecrübeler bu değerın 0.6 – 0.8 arasında seçilmesinin uygun olacağını göstermektedir (Öztemel, 2012, s. 99). Bu çalışmada **momentum katsayısı** (*mc*)'nin 0.2 – 0.9 arasında olmasına karar verilmiş ve ağ en iyi performansını 0.7 dolaylarında iken sergilemiştir. Aktif büyüklük değişkenini göz önünde bulundurup yukarıdaki YSA kodu kullanılarak eğitilen ağın eğitim grafiği de aşağıdaki gibidir:



Şekil 6.2: Aktif büyüklüğe göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

Şekil 6.2' de görüldüğü gibi eğitim 1 ila 0,1 arasında bir hata ile başlayıp, yaklaşık ilk 100 iterasyonda 0,01 seviyelerine kadar inmiş ve toplamda 136904 iterasyonda hedef değer olan 0,000004 hata oranının altına inmiştir. Eğitim sonucunun istenilen performans düzeyini karşılması üzerine ağın test edilmesine geçilmiştir.

Piyasa değeri göz önünde bulundurularak oluşturulan portföy kombinasyonları ise Tablo 6.4'te verilmiştir.

Tablo 6.4: Piyasa değerine göre portföyler matrisi

Piyasa Değerine Göre Portföyler Matrisi																
Portföyler	Bağımsız Değişkenler (Girdiler)														Bağımlı Değişkenler (Çıktılar)	
	Hisse 1	Hisse 2	Hisse 3	Hisse 4	Hisse 5	Hisse 6	Hisse 7	Hisse 8	Hisse 9	Hisse 10	Hisse 11	Hisse 12	Hisse 13	Hisse 14	Getiri (%)	Risk
Portföy 1	2	4	30	23	43	48	75	71	99	98	110	112	121	119	3.3021	0.0020
Portföy 2	7	20	25	34	47	45	59	70	80	88	102	115	127	136	5.7554	0.0081
Portföy 3	15	17	24	32	49	41	67	64	95	81	105	113	138	139	4.2864	0.0074
Portföy 4	3	13	22	29	44	50	61	74	97	94	116	114	140	135	5.1193	0.0048
Portföy 5	18	8	36	27	54	39	57	73	87	79	101	107	120	133	9.1271	0.0400
Portföy 6	1	6	28	55	68	72	86	93	100	106	104	122	124	128	2.8732	0.0058
Portföy 7	16	10	31	46	38	69	66	76	90	77	103	126	118	132	5.6150	0.0072
Portföy 8	14	9	26	51	53	65	56	84	91	78	109	117	137	134	4.4321	0.0095
Portföy 9	11	5	33	40	42	58	63	83	89	96	108	131	123	125	4.3129	0.0075
Portföy 10	19	12	21	35	52	37	62	60	82	92	85	111	130	129	2.2621	0.0046

Ağın topolojisi belirlendikten sonra Tablo 6.4'te verilen girdi ve çıktılar, bütün değerler 1 den küçük olacak şekilde belirlenen bir katsayıya bölünerek kodlanmıştır.

Piyasa değerine göre oluşturulan portföylerde risk ve getiri tahminine ilişkin yapay sinir ağı kodlaması aşağıdaki gibidir:

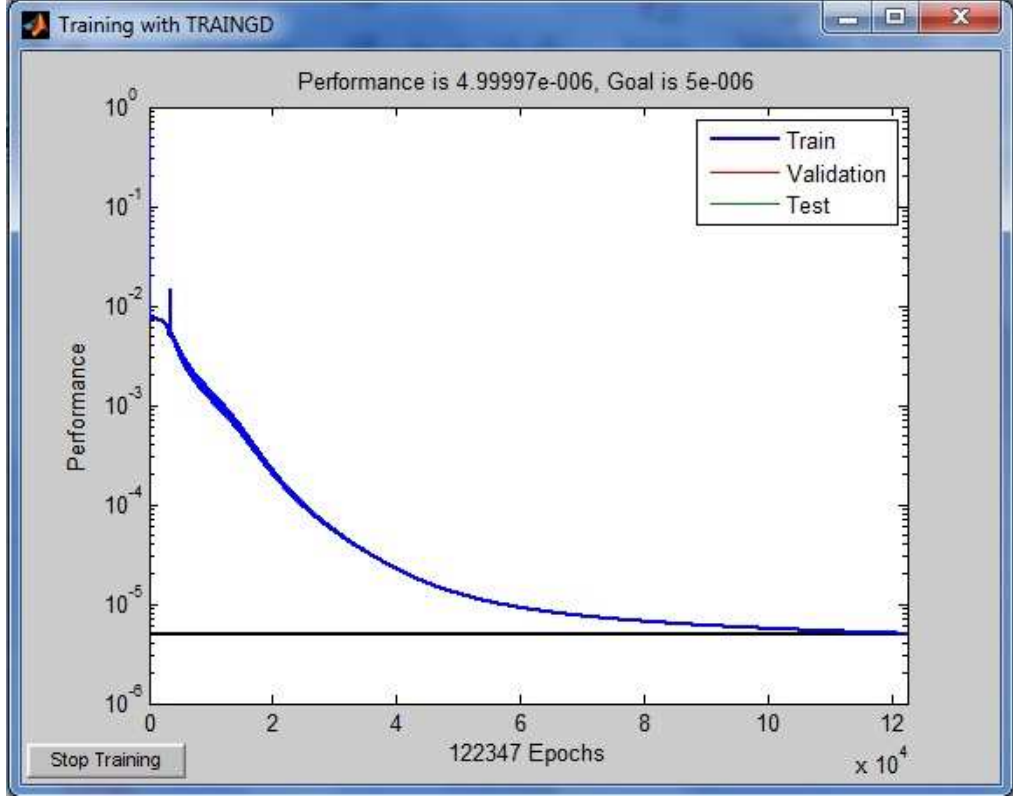
```

P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premnmx(P,T);
[S0,Q]=size(P);
S1=14; %1. Ara katman nöron sayısı
S2=12; %2. Ara katman nöron sayısı
S3=10; %3. Ara katman nöron sayısı
S4=8; %4. Ara katman nöron sayısı
S5=5; %5. Ara katman nöron sayısı
S6=3; %6. Ara katman nöron sayısı
S7=2; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6,S7],{'purelin','tansig','tansig','tansig','tansig','tansig','tansig','tansig'},'traingd');
net.trainParam.epochs=200000;
net.trainParam.show=1000;
net.trainParam.goal=0.000005;
net.trainParam.mc=0.6; % Momentum coefficient
net.trainParam.lr=0.05; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save piyasadegeri net

```

Piyasa değeri kriterine göre elde edilen portföyleri eğitmek için kullanılan ileri beslemeli YSA modelinde diğer ağlara nispeten farklı bir kodlama kullanılmıştır. Bunun sebebi, ağın bu kodlama ile daha iyi eğitilmesidir. Burada ağın topolojisi 1 girdi katmanı, 6 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 14 (14 adet bağımsız girdi değişkenine karşılık olarak) 14, 12, 10, 8, 5, 3 ve 2 (2 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, diğer katmanlarda ise tanjant sigmoid fonksiyonu kullanılmıştır. Ayrıca bu algoritmada **Öğrenme oranında (*lr*)** değişiklik yapılmazken **momentum katsayısı (*mc*)** 0.6 olarak belirlenmiştir.

Piyasa değeri baz alınarak oluşturulan portföylerin, verilen YSA kodu ile eğitilmesi sonucu oluşan eğitim grafiği Şekil 6.3'te görüldüğü gibidir.



Şekil 6.3: Piyasa değerine göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

İşlem hacmi göz önünde bulundurularak oluşturulan portföy kombinasyonları da Tablo 6.5'te verilmiştir.

Tablo 6.5: İşlem hacmine göre portföyler matrisi

İşlem Hacmine Göre Portföyler Matrisi																
Portföyler	Bağımsız Değişkenler (Girdiler)														Bağımlı Değişkenler (Çıktılar)	
	Hisse 1	Hisse 2	Hisse 3	Hisse 4	Hisse 5	Hisse 6	Hisse 7	Hisse 8	Hisse 9	Hisse 10	Hisse 11	Hisse 12	Hisse 13	Hisse 14	Getiri (%)	Risk
Portföy 1	20	7	32	24	48	45	70	75	79	95	110	115	129	120	10.4769	0.0479
Portföy 2	2	3	29	28	49	43	60	64	91	78	113	114	119	130	2.8921	0.0041
Portföy 3	1	8	33	35	47	44	67	61	92	100	112	104	139	136	6.1754	0.0075

Portföy 4	16	12	31	26	38	51	57	59	99	86	111	103	121	127	5.2871	0.0086
Portföy 5	10	19	27	21	41	54	65	71	80	88	90	102	126	138	3.9214	0.0048
Portföy 6	4	9	36	30	42	39	56	62	96	81	89	109	135	124	5.3964	0.0054
Portföy 7	15	5	25	40	50	69	68	94	83	97	107	132	134	123	4.9057	0.0067
Portföy 8	13	6	22	55	58	72	85	82	77	106	108	117	122	125	2.1950	0.0042
Portföy 9	18	14	34	53	52	73	66	87	84	93	116	137	128	118	3.2036	0.0058
Portföy 10	17	11	23	37	46	63	74	76	98	101	105	140	131	133	2.8171	0.0045

Ağın topolojisi belirlendikten sonra Tablo 6.5'te verilen girdi ve çıktılar, bütün değerler 1 den küçük olacak şekilde belirlenen bir katsayıya bölünerek kodlanmıştır.

İşlem hacmi dikkate alınarak oluşturulan portföylerde risk ve getiri tahminine ilişkin yapay sinir ağı kodlaması aşağıdaki gibidir:

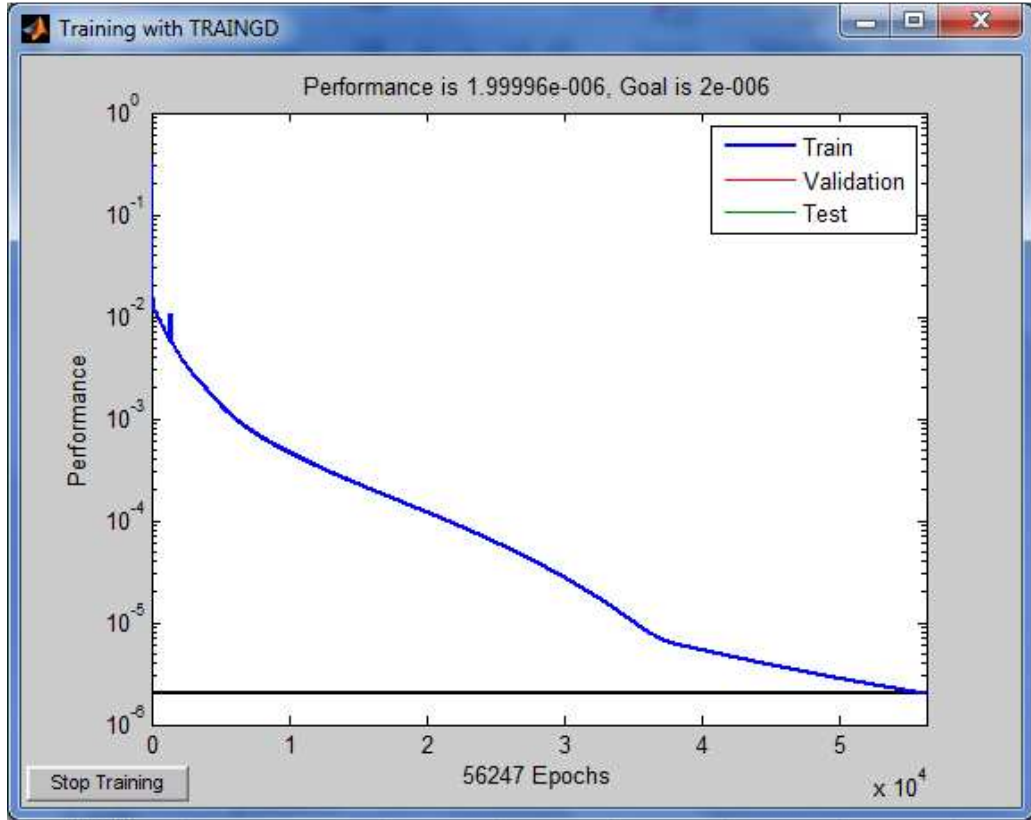
```

P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premnmx(P,T);
[S0,Q]=size(P);
S1=12; %1. Ara katman nöron sayısı
S2=10; %2. Ara katman nöron sayısı
S3=8; %3. Ara katman nöron sayısı
S4=6; %4. Ara katman nöron sayısı
S5=4; %5. Ara katman nöron sayısı
S6=2; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6],{'purelin',
'tansig','tansig','tansig','logsig','tansig','tansig',
'},'traingd');
net.trainParam.epochs=200000;
net.trainParam.show=1000;
net.trainParam.goal=0.000002;
net.trainParam.mc=0.7; % Momentum coefficient
net.trainParam.lr=0.05; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save islemhacmi net

```

İşlem hacmi kriterine göre elde edilen portföyleri eğitmek için kullanılan ileri beslemeli YSA modelinde de yine ağın topolojisi 1 girdi katmanı, 5 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 14 (14 adet bağımsız girdi değişkenine karşılık olarak) 12, 10, 8, 6, 4 ve 2 (2 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, diğer katmanlarda ise tanjant sigmoid ve logistik sigmoid fonksiyonları kullanılmıştır. Daha önceki eğitimlerde kullanılan **Öğrenme oranı (lr)** ve **momentum katsayısı (mc)** değerleri bu eğitimde de iyi performans gösterdiğinden dolayı önceki değerler bu eğitimde de korunmuştur.

İşlem hacmine göre oluşturulan portföyler kullanılarak, verilen YSA kodu ile eğitilmesi sonucu oluşan eğitim grafiği Şekil 6.4’de görüldüğü gibidir.



Şekil 6.4: İşlem hacmine göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

Şekil 6.4' de görüldüğü gibi eğitim 1 ile 0,1 arasında bir hata ile başlayıp, yaklaşık ilk 100 iterasyonda 0,01 seviyelerine kadar inmiş ve toplamda 56247 iterasyonda hedef değer olan 0,000002 hata oranının altına inmiştir. Eğitim sonucunun istenilen performans düzeyini karşılaması üzerine ağın test edilmesine geçilmiştir.

Son olarak şirketlerin özsermaye büyüklüğüne göre belirlenen portföy bileşenleri ve onların risk-getiri değerleri Tablo 6.6'da verilmiştir.

Tablo 6.6: Özsermayeye göre portföyler matrisi

Özsermayeye Göre Portföyler Matrisi																
Portföyler	Bağımsız Değişkenler (Girdiler)														Bağımlı Değişkenler (Çıktılar)	
	Hisse 1	Hisse 2	Hisse 3	Hisse 4	Hisse 5	Hisse 6	Hisse 7	Hisse 8	Hisse 9	Hisse 10	Hisse 11	Hisse 12	Hisse 13	Hisse 14	Getiri (%)	Risk
Portföy 1	2	4	30	25	43	48	75	71	99	88	110	115	119	127	2.4650	0.0032
Portföy 2	20	15	23	32	45	41	64	59	80	98	102	112	136	139	3.2100	0.0054
Portföy 3	7	13	22	34	44	50	70	57	81	87	116	105	121	140	6.4547	0.0068
Portföy 4	3	17	21	29	47	49	73	74	82	95	113	107	138	117	4.2564	0.0054
Portföy 5	18	14	24	27	52	37	67	61	78	92	101	114	129	134	4.5600	0.0085
Portföy 6	19	10	35	36	54	39	62	66	97	85	83	108	118	133	4.1957	0.0043
Portföy 7	12	16	28	38	40	56	60	84	76	77	111	130	137	132	2.6871	0.0075
Portföy 8	6	9	33	55	63	72	89	86	93	103	104	128	125	120	5.4686	0.0071
Portföy 9	11	1	31	53	42	65	68	90	79	100	106	126	122	124	8.3304	0.0430
Portföy 10	9	5	26	46	51	58	69	91	94	96	109	135	131	123	5.4579	0.0096

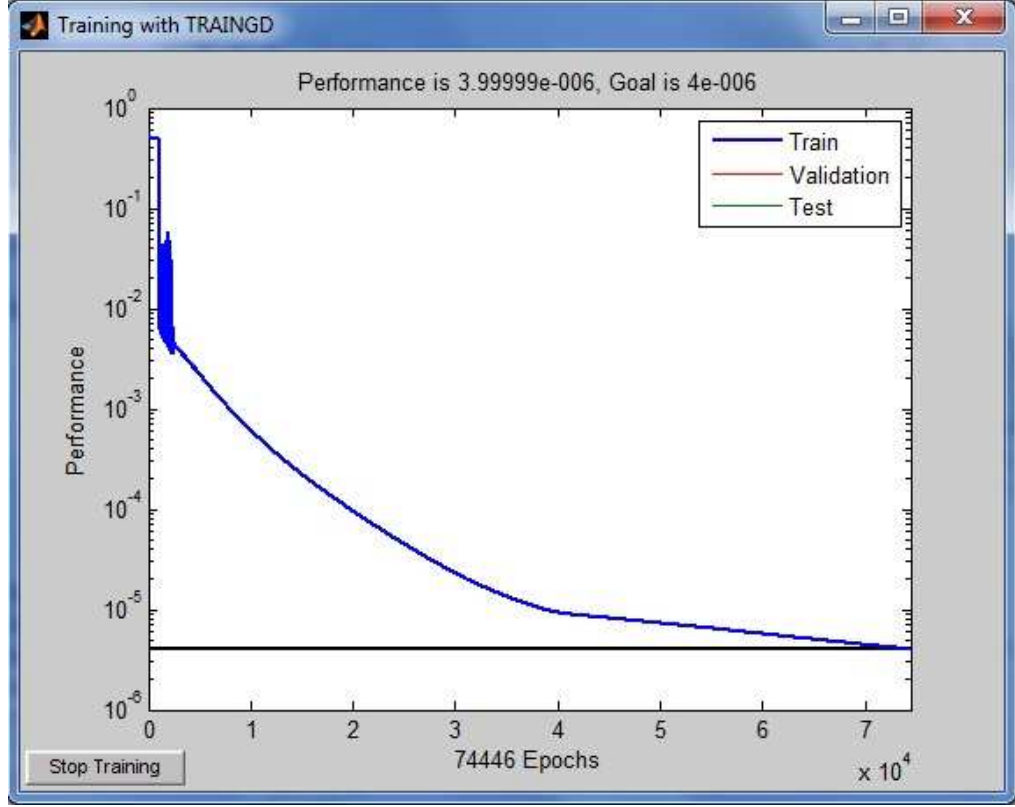
Ağın topolojisini belirledikten sonra yukarıdaki Tablo 6.6'da verilen girdi ve çıktı ürünleri, bütün değerler 1 den küçük olacak şekilde belirlenen bir katsayıya bölünerek kodlanmıştır.

Özsermaye yapısı dikkate alınarak oluşturulan portföylerde risk ve getiri tahminine ilişkin yapay sinir ağı kodlaması ise aşağıdaki gibidir:

```
P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premnmx(P,T);
[S0,Q]=size(P);
S1=12; %1. Ara katman nöron sayısı
S2=10; %2. Ara katman nöron sayısı
S3=8; %3. Ara katman nöron sayısı
S4=6; %4. Ara katman nöron sayısı
S5=4; %5. Ara katman nöron sayısı
S6=2; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6],{'purelin',
'tansig','tansig','tansig','logsig','tansig','tansig'},
'traingd');
net.trainParam.epochs=200000;
net.trainParam.show=1000;
net.trainParam.goal=0.000004;
net.trainParam.mc=0.7; % Momentum coefficient
net.trainParam.lr=0.05; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save ozsermaye net
```

Yukarıdaki YSA kodunda görüldüğü gibi, önceki kriterlere göre eğitim yapılan YSA kodlarıyla yaklaşık aynı yapıya sahiptir. Kullanılan katman sayısı, aktivasyon fonksiyonu türleri, öğrenme oranı ve momentum katsayıları aynıdır.

Burada verilen YSA eğitim algoritması ile eğitilen ağın eğitim sonucunda gösterdiği performans grafiği de aşağıdaki gibidir:



Şekil 6.5: Özsermayeye göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

Şekil 6.5’ de görüldüğü gibi eğitim 1 ila 0,1 arasında bir hata ile başlayıp, yaklaşık ilk 2000 iterasyonda 0,005 seviyelerine kadar inmiş ve toplamda 74446 iterasyonda hedef değer olan 0,000004 hata oranının altına inmiştir. Eğitim sonucunun istenilen performans düzeyini karşılaması üzerine ağın test edilmesine geçilmiştir.

6.1.4 Test Sonuçları ve Tartışma

Hisse senetlerinin aktif büyüklüğü, piyasa değeri, işlem hacmi ve özsermaye büyüklüğü kriterlerine göre oluşturulmuş portföylerden ilk 9 tanesi ile daha önce de belirtildiği üzere ağın eğitimi yapılmıştı. Eğitilmiş olan YSA Tablo 6.3, 6.4, 6.5 ve 6.6’da verilmiş olan 14 bağımsız değişkenin (14 tane çeşitlendirilmiş portföy) risk ve getirisi kullanılarak ilgili dönem verileri için test edilip ağa hiç gösterilmeyen ve belirtilen dört kriterden her birinin 10. portföyü ile

test aşaması gerçekleştirilmiştir. Test sonuçları aşağıdaki Tablo 6.7’de olduğu gibidir.

Tablo 6.7: Risk ve getiri tahminine ilişkin test sonuçları

YSA Testi İçin Portföy Kriterleri	(A) Getiri (%)	(B) YSA ile Tahmin Edilen Getiri (%)	(C) =(A-B)/A Hata Oranı	(D) Risk (Stand. Sapma)	(E) YSA ile Tahmin Edilen Risk (Stand. Sapma)	(F) =(D-E) Hata Miktarı
Aktif Büyüklüğe Göre Test Verisi	4.4850	4.5285	-0.0097	0.0074	0.0039	-0.0035
Piyasa Değerine Göre Test Verisi	2.2621	2.2695	-0.0033	0.0046	0.0033	-0.0013
İşlem Hacmine Göre Test Verisi	2.8171	2.8290	-0.0042	0.0045	0.0034	0.0011
Özsermayeye Göre Test Verisi	5.4579	5.4420	0.0029	0.0096	0.0097	-0.0001

Tabloda görüldüğü gibi eğitim sonuçları getiri ve risk bazında en iyi sonucu özsermayeye göre oluşturulan portföylerde vermiştir. Bunun sebebinin, özsermayeye göre oluşturulan portföylerle ağırlık daha iyi eğitilmiş olmasından kaynaklandığı söylenebilir. Ayrıca YSA ile getiri tahmininin %1’in altında hata oranı ile gerçekleştiği, risk tahmininde ise hata miktarının binde 5’in altında olduğu gözlenmektedir.

Buna ilaveten; Tablo 6.3’te aktif büyüklüğü en yüksek olan hisse senetleriyle oluşturulan portföyün getirisi diğer portföylere göre daha yüksek getiriye sahip olmamasına rağmen risk seviyesi diğer portföy risklerine nazaran minimum seviyededir. Benzer şekilde Tablo 6.4’te yer alan ve piyasa değeri en yüksek olan hisse senetleriyle oluşturulan portföyün getirisi maksimum olmamasına rağmen, risk seviyesi diğer portföylere göre en düşük seviyededir.

Tablo 6.5'te görüldüğü üzere, işlem hacmi en yüksek olan hisse senetleriyle oluşturulan portföyün getiri ve riski maksimum düzeydedir. Bununla birlikte Tablo 6.6'da yer alan özsermayesi en yüksek olan hisse senetlerinin oluşturduğu portföyün getirisi maksimum değerde bulunmazken, riskinin minimum düzeyde yer aldığı gözlenmektedir.

6.2 Optimum Portföyün Belirlenmesi

Optimizasyon yöntemlerinde amaç belirtilen kısıtlar altında en iyi çözümün elde edilmesidir. Bu çalışmada da Tablo 6.1'de yer alan 140 hisse senedi arasından aktif büyüklük, piyasa değeri, işlem hacmi, özsermaye ya da herhangi başka bir kriter göz önüne alınmaksızın rastgele olarak (random) seçilen 10'ar hisse senediyle eşit ağırlıklı portföyler oluşturulmuştur. Bu yöntemle oluşturulan eşit ağırlıklı 50 portföyün risk ve getirileri hesaplanmış ve bu değerler Tablo 6.8'de verilmiştir.

Tablo 6.8: 140 hisse senedi ile oluşturulmuş eşit ağırlıklı portföyler

140 Hisse Senedi Kullanılarak Rastgele Oluşturulmuş Eşit Ağırlıklı Portföyler												
Portföyler	Bağımsız Değişkenler (Girdiler)										Bağımlı Değişkenler (Çıktılar)	
	Hisse 1	Hisse 2	Hisse 3	Hisse 4	Hisse 5	Hisse 6	Hisse 7	Hisse 8	Hisse 9	Hisse 10	Getiri (%)	Risk
Portföy 1	20	3	30	23	43	45	71	70	99	81	4.1030	0.0040
Portföy 2	7	17	25	32	48	41	67	73	80	88	4.2496	0.0065
Portföy 3	15	13	34	22	49	50	61	62	98	82	4.2190	0.0050
Portföy 4	18	8	35	21	44	39	74	66	90	92	4.9790	0.0054
Portföy 5	16	19	36	24	47	37	63	60	95	85	5.4530	0.0092
Portföy 6	12	14	29	27	52	38	65	58	97	76	2.7320	0.0050

Portföy 7	10	9	28	51	54	69	72	78	84	83	4.0500	0.0057
Portföy 8	11	1	26	40	46	68	75	91	94	96	4.4355	0.0072
Portföy 9	6	5	31	42	53	59	64	100	79	89	4.8380	0.0055
Portföy 10	2	4	33	55	57	56	86	93	87	103	5.0300	0.0091
Portföy 11	2	4	30	23	43	48	75	71	99	98	3.1790	0.0019
Portföy 12	7	20	25	34	47	45	59	70	80	88	6.4076	0.0110
Portföy 13	15	17	24	32	49	41	67	64	95	81	4.5280	0.0078
Portföy 14	3	13	22	29	44	50	61	74	97	94	5.0870	0.0040
Portföy 15	18	8	36	27	54	39	57	73	87	79	4.3950	0.0054
Portföy 16	19	12	21	35	52	37	62	60	82	92	2.7260	0.0046
Portföy 17	16	10	31	46	38	69	66	76	90	77	4.5800	0.0061
Portföy 18	14	9	26	51	53	65	56	84	91	78	4.4670	0.0093
Portföy 19	11	5	33	40	42	58	63	83	89	96	4.2210	0.0065
Portföy 20	1	6	28	55	68	72	86	93	100	106	3.0245	0.0061
Portföy 21	20	7	32	24	48	45	70	75	79	95	6.3016	0.0093
Portföy 22	2	3	29	28	49	43	60	64	91	78	2.5560	0.0035
Portföy 23	1	8	33	35	47	44	67	61	92	100	7.2825	0.0088
Portföy 24	16	12	31	26	38	51	57	59	99	86	4.3750	0.0093
Portföy 25	10	19	27	21	41	54	65	71	80	88	3.5200	0.0042
Portföy 26	4	9	36	30	42	39	56	62	96	81	4.1800	0.0040
Portföy 27	15	5	25	40	50	69	68	94	83	97	4.4690	0.0047
Portföy 28	17	11	23	37	46	63	74	76	98	101	3.0170	0.0040
Portföy 29	18	14	34	53	52	73	66	87	84	93	3.4200	0.0056
Portföy 30	13	6	22	55	58	72	85	82	77	106	1.8570	0.0043
Portföy 31	2	4	30	25	43	48	75	71	99	88	2.4060	0.0025

Portföy 32	20	15	23	32	45	41	64	59	80	98	3.0700	0.0044
Portföy 33	7	13	22	34	44	50	70	57	81	87	7.5916	0.0089
Portföy 34	3	17	21	29	47	49	73	74	82	95	3.9020	0.0061
Portföy 35	18	14	24	27	52	37	67	61	78	92	4.8320	0.0082
Portföy 36	19	10	35	36	54	39	62	66	97	85	4.2900	0.0038
Portföy 37	12	16	28	38	40	56	60	84	76	77	2.8470	0.0080
Portföy 38	9	5	26	46	51	58	69	91	94	96	4.8800	0.0080
Portföy 39	11	1	31	53	42	65	68	90	79	100	4.0705	0.0053
Portföy 40	6	9	33	55	63	72	89	86	93	103	6.0930	0.0083
Portföy 41	81	110	102	127	121	7	17	25	32	48	4.9046	0.0074
Portföy 42	88	116	112	138	133	15	13	34	22	49	4.9880	0.0044
Portföy 43	82	113	114	135	130	18	8	35	21	44	4.8840	0.0071
Portföy 44	92	115	105	118	129	16	19	36	24	47	4.6270	0.0104
Portföy 45	85	101	107	134	137	12	14	29	27	52	2.7380	0.0066
Portföy 46	76	77	111	117	132	10	9	28	51	54	3.2840	0.0057
Portföy 47	83	108	131	123	120	11	1	26	40	46	5.5245	0.0099
Portföy 48	96	109	128	124	125	6	5	31	42	53	2.8650	0.0074
Portföy 49	64	100	122	126	119	2	4	33	55	57	2.9680	0.0036
Portföy 50	2	4	127	136	15	17	140	135	18	8	4.6370	0.0054

6.2.1 Uygulamanın Amacı

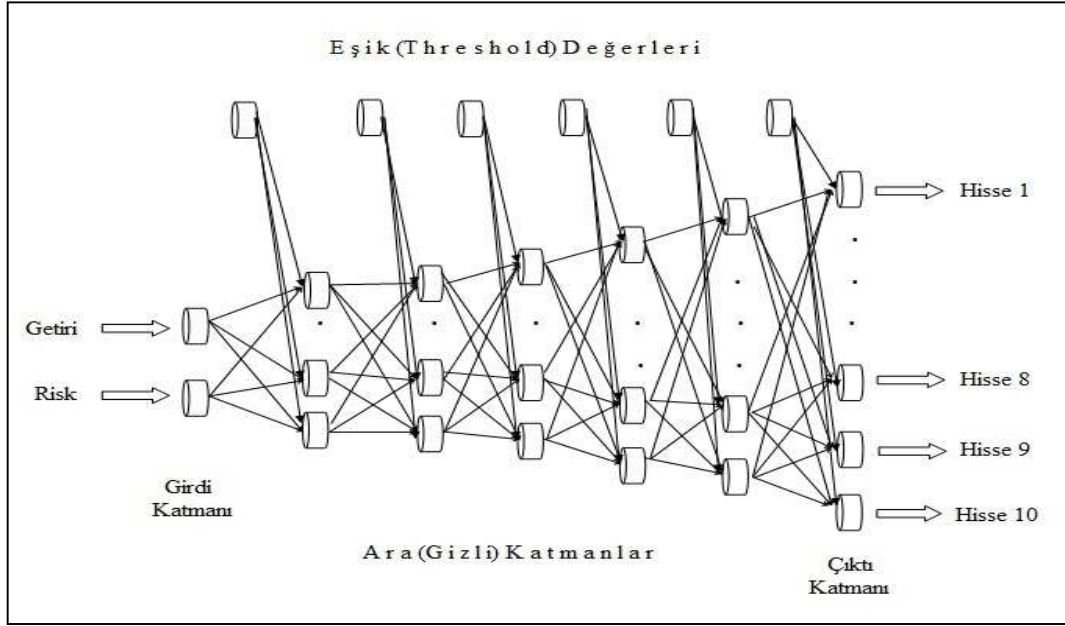
Çalışmanın uygulama kısmının bu bölümünde amaç, geliştirilen bir yapay sinir ağı modeli kullanılarak optimum portföyün elde edilmesidir. Bunun için öncelikle bir yapay sinir ağı modeli geliştirilmiş ve geliştirilen bu ağ Tablo 6.1’de

verilen 140 hisse senedi içerisinde 10'ar tanesinin rastgele (random) alınmasıyla oluşturulan portföylerin risk ve getirileri kullanılarak eğitilmiştir. Eğitilen bu ağ yardımıyla optimum portföy (minimum risk veya maksimum getiri) elde edilmeye çalışılmıştır. Ayrıca daha önce de belirtildiği üzere, bu portföyleri seçerken Bölüm 6.1'de olduğunun tersine herhangi bir kriter göz önünde bulundurulmamıştır.

6.2.2 Portföy Optimizasyonu İçin YSA Modelinin Geliştirilmesi

Çalışmanın optimizasyon kısmında da risk-getiri tahmininde olduğu gibi ileri beslemeli ve çok katmanlı bir yapay sinir ağı (MLP) modeli kullanılmıştır. Tablo 6.1'de yer alan 140 hisse senedi kullanılarak her bir portföyde 10 hisse senedi olacak şekilde Tablo 6.8'de gösterildiği gibi 50 tane eşit ağırlıklı portföy oluşturulmuştur. Bu 50 portföyden 49 tanesi ağı eğitmek için kullanılmış ve en yüksek getiriye sahip portföy ve en düşük riske sahip portföy de eğitilen ağın performansını ve optimizasyon yeteneğini test etmek amacıyla kullanılmıştır.

Kullanılan ileri beslemeli ağın topolojisi 1 girdi katmanı, 5 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 2 (2 adet bağımsız girdi değişkenine karşılık olarak) 4, 6, 8, 10 ve 10 (10 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, diğer katmanlarda ise tanjant sigmoid ve logistik sigmoid fonksiyonları kullanılmıştır. Bu çalışmanın optimizasyon aşamasında kullanılan yapay sinir ağının topolojisi Şekil 6.6'da görüldüğü gibidir.



Şekil 6.6: Portföy optimizasyonu için oluşturulan MLP ağının topolojisi

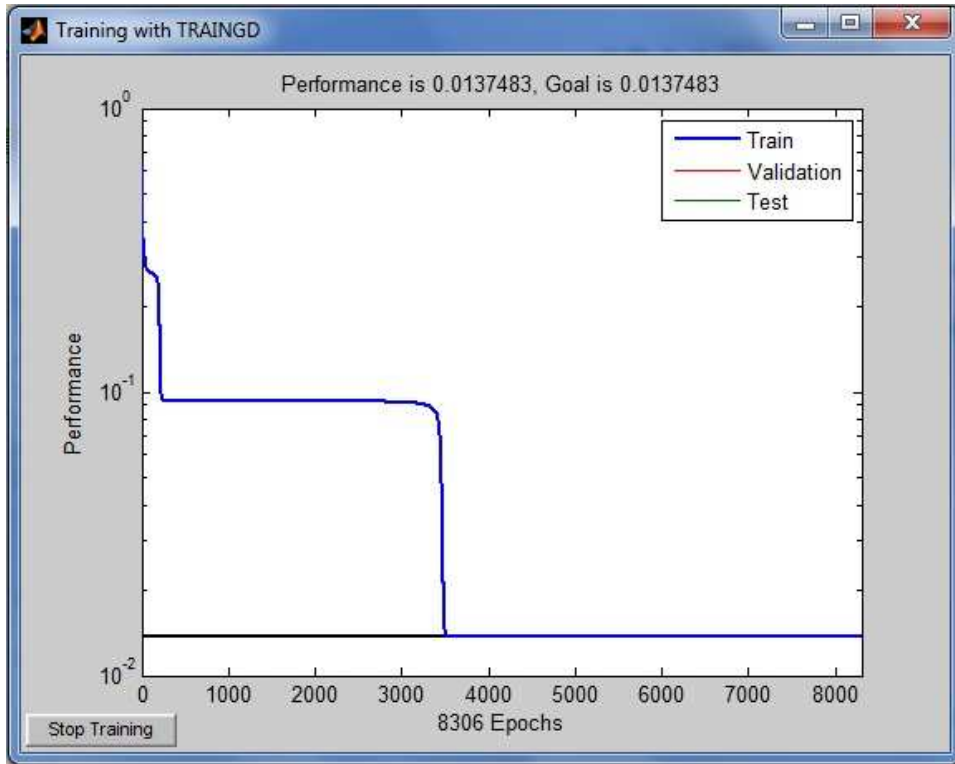
6.2.2.1 Getiri Maksimizasyonu

Oluşturulan portföylerden maksimum getiriye elde etmek için kullanılan yapay sinir ağı kodlaması aşağıdaki gibidir:

```
P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premmx(P,T);
[S0,Q]=size(P);
S1=2; %1. Ara katman nöron sayısı
S2=4; %2. Ara katman nöron sayısı
S3=6; %3. Ara katman nöron sayısı
S4=8; %4. Ara katman nöron sayısı
S5=10; %5. Ara katman nöron sayısı
S6=10; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6],{'purelin','log
sig','logsig','logsig','tansig','tansig','tansig'},'traing
d');
net.trainParam.epochs=50000; net.trainParam.show=1000;
net.trainParam.goal=0.0137;
net.trainParam.mc=0.8; % Momentum coefficient
net.trainParam.lr=0.053; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save portopt_egitimsonucugetiri net
```

Tablo 6.8’de verilen 50 portföyden maksimum getiriye sahip olan 33 numaralı portföyü elde etmeye yönelik ağ eğitimi için kullanılan ileri beslemeli YSA modelinde de yine ağın topolojisi 1 girdi katmanı, 5 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 2 (2 adet bağımsız girdi değişkenine karşılık olarak) 2, 4, 6, 8, 10 ve 10 (10 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, sonraki 3 katmanda logistik sigmoid ve son 3 katmanda ise tanjant sigmoid fonksiyonları kullanılmıştır. Ayrıca ağ eğitiminde **öğrenme oranı (lr)** olarak 0,053 değeri ve **momentum katsayısı (mc)** olarak da 0,8 değeri kullanılmıştır.

Adı geçen portföylerin risk - getirileri kullanılıp verilen YSA kodu ile eğitilmesi sonucu oluşan eğitim grafiği Şekil 6.7’de görüldüğü gibidir.



Şekil 6.7: Maksimum getiriye göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

Şekil 6.7’ de görüldüğü gibi eğitim 1 ile 0,1 arasında bir hata ile başlayıp, yaklaşık ilk 3500 iterasyonda 0,02 seviyelerine kadar inmiş ve toplamda 8306

iterasyonda hedef deęer olan 0,01375 hata oranının altına inmiřtir. Eęitim sonucunun istenilen performans dzeyini karřılaması zerine aęın test edilmesine geilmiřtir.

6.2.2.1.1 Test Sonuları ve Tartıřma

řekil 6.7’de grlen eęitilmiř aę modeli kullanılarak test ařaması gerekleřtirilmiřtir. Bu ařamada maksimum getiriye sahip portfy elde etmek amalanmıřtır. Bu amala hedef getiri olan %7.5916 deęerine ulařmaya ynelik test sonuları ařaęıdaki tabloda verildięi gibidir.

Tablo 6.9: Getiri maksimizasyonuna iliřkin test sonuları

(A) Optimum Portfyn Getirisi (%)	YSA ile Elde Edilen Portfyde Yer Alan Hisseler	(B) YSA ile Tahmin Edilen Portfyn Getirisi (%)	(C) =(A-B)/A Hata Oranı
7.5916	24, 27, 44, 47, 54, 59, 63, 68, 78, 81	7.1590	0.0567

Tablo 6.9’da grldęi gibi YSA kullanılarak %7.5916 getiri oranına yakın bir getiri deęeri (%7.1590) elde edilmiřtir. Ayrıca bu getiriye saęlayan portfyler ve hata oranı da tabloda belirtilmiřtir.

6.2.2.2 Risk Minimizasyonu

Tablo 6.8’de verilen 50 portfy ierisinden minimum riske sahip portfyn elde edilmesine ynelik alıřmada kullanılan yapay sinir aęı kodlaması ařaęıdaki gibidir.

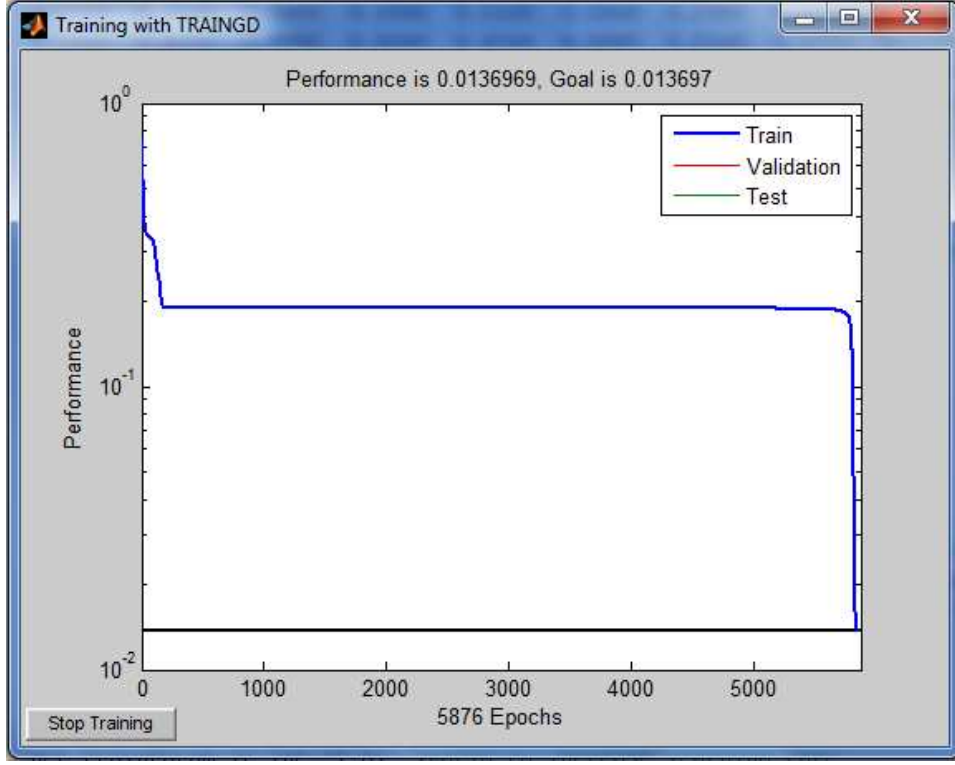

```

P0=[Girdi Matrisi];
Tpre=[Çıktı Matrisi];
P=P0'; % Transpozunu al.
T=Tpre'; % Transpozunu al.
[Pn,minP,maxP,tn,minT,maxT]=premnmx(P,T);
[S0,Q]=size(P);
S1=2; %1. Ara katman nöron sayısı
S2=4; %2. Ara katman nöron sayısı
S3=6; %3. Ara katman nöron sayısı
S4=6; %4. Ara katman nöron sayısı
S5=8; %5. Ara katman nöron sayısı
S6=10; %6. Ara katman nöron sayısı
S7=10; % Çıktı katmanı nöron sayısı
net=newff(minmax(P),[S0,S1,S2,S3,S4,S5,S6,S7],{'purelin',
'logsig','logsig','logsig','logsig','logsig','tansig','tansig','tansig'},
'traingd');
net.trainParam.epochs=100000;
net.trainParam.show=1000;
net.trainParam.goal=0.0136963;
net.trainParam.mc=0.8; % Momentum coefficient
net.trainParam.lr=0.053; % Learning rate
net.trainParam.lr_inc=1.01; % Ratio to increase lr
[net,tr]=train(net,P,T);
save portopt_egitimsonucurisk net

```

Tablo 6.8’de verilen 50 portföyden minimum riske sahip olan 11 numaralı portföyü elde etmeye yönelik ağ eğitimi için kullanılan ileri beslemeli YSA modelinde de yine ağın topolojisi 1 girdi katmanı, 6 ara katman ve 1 de çıktı katmanından oluşmuştur. Her katmanda sırasıyla 2 (2 adet bağımsız girdi değişkenine karşılık olarak) 2, 4, 6, 6, 8, 10 ve 10 (10 adet bağımlı çıktı değerine karşılık olarak) nöron bulunmaktadır. Girdi katmanındaki nöronlarda aktivasyon fonksiyonu olarak pürelin, sonraki 4 katmanda logistik sigmoid ve son 3 katmanda ise tanjant sigmoid fonksiyonları kullanılmıştır. Ayrıca ağ eğitiminde **öğrenme oranı (*lr*)** olarak 0,053 değeri ve **momentum katsayısı (*mc*)** olarak da 0,8 değeri kullanılmıştır.

Adı geçen portföylerin risk - getirileri kullanılıp verilen YSA kodu ile eğitilmesi sonucu oluşan eğitim grafiği Şekil 6.8’de görüldüğü gibidir.



Şekil 6.8: Minimum riske göre eğitimi tamamlanan YSA modelinin öğrenme grafiği

Şekil 6.8’ de görüldüğü gibi eğitim 1 ile 0,1 arasında bir hata ile başlayıp, yaklaşık ilk 500 iterasyonda 0,5 seviyelerine kadar inmiş ve toplamda 5876 iterasyonda hedef değer olan 0,013697 hata oranının altına inmiştir. Eğitim sonucunun istenilen performans düzeyini karşılaması üzerine ağın test edilmesine geçilmiştir.

6.2.2.2.1 Test Sonuçları ve Tartışma

Şekil 6.8’de görülen eğitilmiş ağ modeli kullanılarak test aşaması gerçekleştirilmiştir. Bu aşamada minimum riske sahip portföyü elde etmek amaçlanmıştır. Bu amaçla hedef risk değeri olan 0,0019 değerine ulaşmaya yönelik test sonuçları aşağıdaki tabloda verildiği gibidir.

Tablo 6.10: Risk minimizasyonuna ilişkin test sonuçları

(A) Optimum Portföyün Riski	YSA ile Elde Edilen Portföyde Yer Alan Hisseler	(B) YSA ile Tahmin Edilen Portföyün Riski	(C) =(A-B) Hata Miktarı
0.0019	4, 23, 30, 48, 54, 57, 66, 71, 75, 98	0.0024	-0.0005

Tablo 6.10’da görüldüğü gibi YSA kullanılarak 0.0019 risk değerine yakın bir değer (0.0024) elde edilmiştir. Ayrıca bu risk değerine sahip portföyler ve hata miktarı da tabloda gösterilmiştir.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, geliştirilen bir yapay sinir ağı modeliyle risk-getiri tahmini ve portföy optimizasyonu işlemi gerçekleştirilmiştir. Çalışmanın giriş bölümü hariç ilk 3 bölümünde yapay sinir ağları teferruatlı bir şekilde açıklanmış, 5. Bölümde portföy optimizasyonu ele alınmış ve son bölümde de uygulamaya yer verilmiştir.

Çalışmanın uygulama kısmı risk-getiri tahmini ve portföy optimizasyonu olmak üzere iki bölümden oluşmaktadır.

Uygulamanın 1. kısmında geliştirilen bir yapay sinir ağı modeliyle, İMKB-Ulusal Sınai Endeksinde yer alan 140 şirket arasından aktif büyüklük, piyasa değeri, işlem hacmi ve özsermaye kriterlerine göre oluşturulan portföyler eğitilmiş ve bu eğitilen ağ ile risk-getiri tahmini yapılmıştır. Eğitimden sonra getiri ve risk bazında en iyi tahmin özsermayeye göre oluşturulan portföylerde elde edilmiştir. Bunun sebebinin, özsermayeye göre oluşturulan portföylerle ağın daha iyi eğitilmiş olmasından kaynaklandığı söylenebilir. Ayrıca YSA ile getiri tahminin %1'in altında hata oranı ile gerçekleştiği, risk tahmininde ise hata miktarının binde 5'in altında olduğu gözlenmektedir.

Bununla beraber aktif büyüklüğü, piyasa değeri ve özsermayesi en yüksek olan hisse senetleriyle oluşturulan portföylerin getirileri diğer portföylere göre daha yüksek olmamasına rağmen risk seviyeleri diğer portföy risklerine nazaran minimum seviyededir. Fakat işlem hacmi en yüksek olan hisse senetleriyle oluşturulan portföyün getiri ve riskinin maksimum düzeyde olduğu gözlenmiştir.

Uygulamanın 2. kısmında, bahsi geçen 140 şirketin risk ve getirileri kullanılarak eşit ağırlıklı 50 tane portföy oluşturulmuştur. Bu portföylerden 49 tanesi geliştirilen bir yapay sinir ağını eğitmek için, son 1 tanesi de (maksimum getiri ve minimum riskli) eğitilen ağ test etmek için kullanılmıştır. Maksimum getiriye sahip portföyün getirisi olan %7.5916 değeri için YSA 0.0567 hata oranı ile %7.1590 değerini bulmuştur. Bu getiriye sahip olan portföyü oluşturan hisse

senetlerinin kodları ve simgeleri ise 24 (ARSAN), 27 (DESA), 44 (IHGZT), 47 (KARTN), 54 (ADEL), 59 (AYGAZ), 63 (DYOBY), 68 (HEKTS), 78 (ADNAC) ve 81 (ANACM) dir.

Ayrıca oluşturulan 50 portföy arasında minimum riske sahip olan portföyün riski de 0.0019 dur. Bu değer YSA'da 0.0005 hata farkıyla 0.0024 olarak tahmin edilmiştir. Bu risk seviyesine sahip portföyü oluşturan hisse senetleri ise 4 (CCOLA), 23 (ALTIN), 30 (KORDS), 48 (KOZAA), 54 (ADEL), 57 (AKSA), 66 (GOODY), 71 (PTOFS), 75 (TUPRS) ve 98 (NUHCM) kodlu hisselerdir.

Daha önceki bölümlerde de belirtildiği gibi, yapay sinir ağlarının finans alanındaki uygulamaları oldukça fazladır. Bundan sonra yapılacak çalışmalarda sistemin uygulanabilirliği test edildikten sonra daha iyi sonuçlar elde edilebilmesi göz önünde bulundurulmalıdır. İleriki çalışmalarda risk ve getiri tahmininde sonuçların daha az hata oranı ile elde edilebilmesi için YSA'yı test etmek amacıyla daha geniş bir zaman dilimi (örneğin; 2 ya da 3 yıllık veriler) kullanılabilir. Ayrıca bu zaman dilimleri daha sık aralıklara (örneğin; günlük ya da haftalık değerlere) bölünerek ele alınabilir. Böylelikle ağ eğitimi daha iyi gerçekleştirilmiş olacak ve tahmin hatası azaltılabilecektir.

Maksimum getirili ve minimum riskli portföyü elde etmek için de risk getiri tahmininde olduğu gibi daha geniş bir zaman dilimi kullanılabilir. Ayrıca elde edilen portföyü oluşturan 10 adet hisse senedi (çıktı değerleri) yerine daha az (örneğin; 5-8 arası) çıktı değeri kullanılabilir. Çünkü 2 girdi değerine karşılık ağın 10 adet çıktı değeri üretmesi zorlaşmakta ve ağın eğitim süreci uzamaktadır. Bununla beraber girdi değerleri olarak risk ve getiri dışında endeks değeri, faiz oranı ve şirketlere ait farklı veriler de kullanılabilir. Böylelikle hem ağ eğitmek kolaylaşacak, hem eğitim daha az sürede gerçekleştirilecek ve hem de hata oranı düşürülebilecektir.

Finansal piyasalarda getiri beklentisi içerisinde olan bireysel ve kurumsal yatırımcılar açısından YSA gibi tahmin modelleri oldukça önem arz etmektedir. Bu nedenle de çalışmaların bu boyutta ele alınıp değerlendirilmesi daha uygun sonuçlar verebilecektir.

8. KAYNAKLAR

Aizenberg, I. (2011). *Complex-Valued Neural Networks with Multi-Valued Neurons*. USA: Springer-Verlag Berlin Heidelberg.

Akel, V. ve Bayramođlu, M. F. (2008). Kriz Dönemlerinde Yapay Sinir Ağları ile Finansal Öngörüde Bulunma: İMKB 100 Endeksi Örneđi. *International Symposium on International Capital Flows and Emerging Markets*, Balıkesir, 79-87.

Al Suqaier, F. S. and Al Ziyud, H. A. (2011). The Effect of Diversification on Achieving Optimal Portfolio. *European Journal of Economics, Finance and Administrative Sciences (EJEFAS)*, 32, 55-63.

Anderson, D. and McNeill, G. (1992). *Artificial Neural Networks Technology*. Rome Laboratory, ELIN:A011, A DACS State-of-the-Art Report: New York.

Baş, N. (2006). Yapay Sinir Ağları Yaklaşımı ve Bir Uygulama. Yüksek Lisans Tezi, *Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü*, İstatistik Anabilim Dalı, İstanbul.

Bayır, F. (2006). Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama. Yüksek Lisans Tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Anabilim Dalı, İstanbul.

Bekçiođlu, S. (1984) Hisse Senetlerinin Riskliliđi: Bazı Türk Firmalarına Ait Hisse Senetleri Üzerinde Bir Deneme. *İstanbul Üniversitesi İşletme Fakültesi Muhasebe Enstitüsü Dergisi*, 10(37), s. 59.

Beltratti, A., Margarita, S. and Terna, P. (1996). *Neural Networks For Economic and Financial Modelling*. London: International Thomson Computer Press.

Brandimarte, P. (2006). *Numerical Methods in Finance and Economics: A MATLAB - Based Introduction*. New Jersey: John Wiley & Sons, Inc.

Broomhead, D. S. and Lowe, D. (1988). *Radial Basis Functions, Multi Variable Functional Interpolation and Adaptive Networks*. Royal Signals and Radar Establishment Memorandum, 4148.

Capinski, M. and Zastawniak, T. (2003). *Mathematics for Finance: An Introduction to Financial Engineering*. USA: Springer Undergraduate Mathematics Series.

Crone, S. F. (2004). *Business Forecasting with Artificial Neural Networks*. Presentation of IBF Tutorial, Boston: Institute of Business Forecasting.

Dede, G. (2008). Yapay Sinir Ağları İle Konuşma Tanıma. Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Elektronik Mühendisliği Anabilim Dalı, Ankara.

Dere, A. (2009). Yapay Sinir Ağları Yöntemi İle Sıvılaştırma Analizi ve Adapazarı İçin Örnek Bir Uygulama. Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Jeofizik Mühendisliği, Sakarya.

Dobbins, R., Witt, S. F. and Fielding, J. (1994). *Portfolio Theory and Investment Management*. USA: Blackwell Publishers.

Duman, N. (2006). Yapay Sinir Ağları ve Bir Uygulama. Yüksek Lisans Tezi, *Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Anabilim Dalı, Sivas.

Efe, Ö. ve Kaynak, O. (2000). *Yapay Sinir Ağları ve Uygulamaları*. İstanbul: Boğaziçi Üniversitesi Yayınları.

Elmas, Ç. (2003). *Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)*. Ankara: Seçkin Yayıncılık.

Elton, E. J., Gruber, M. J., Brown, S. J. and Goetzmann, W. N. (2009). *Modern Portfolio Theory and Investment Analysis*. USA: John Wiley and Sons.

Evans, J. L. and Archer, S. H. (1968). Diversification and the Reduction of Dispersion: An Empirical Analysis. *The Journal of Finance*, 23(5), 761-767.

Fabozzi, F. J., Gupta, F. and Markowitz, H. M. (2002). The Legacy of Modern Portfolio Theory. *The Journal of Investing*, 11(3), 7-22.

Farrell, J. L. (1997). *Portfolio Management: Theory and Application*. Singapore: The McGraw-Hill Companies Inc.

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Englewood Cliffs, NJ: Prentice-Hall International.

Fermanian, J. D. and Scailet, O. (2005). Sensitivity Analysis of VaR and Expected Shortfall for Portfolios Under Netting Agreements. *Journal of Banking and Finance*, 29, s. 937.

Fiesler, E. (1996). Neural Network Topologies. (eds: E. Fiesler and R. Beale), *Handbook of Neural Computation*. New York: Oxford University Press and IOP Publishing.

Focardi, S. M. and Fabozzi, F. J. (2004). *The Mathematics of Financial Modeling and Investment Management*. New Jersey: John Wiley & Sons, Inc.

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

Fröhlich, J. (1997). Neural Net Overview / Kohonen Feature Map [online]. (23 Nisan 2012), <http://fbim.fh-regensburg.de/~saj39122/jfroehl/diplom/e-1.html>.

Fyfe, C. (1996). *Artificial Neural Networks*. The University of Paisley: Lecture Notes.

Galushkin, A. I. (2007). *Neural Networks Theory*. New York: Springer-Verlag Berlin Heidelberg.

Genel, H. (2004). Genetik Algoritmalarla Portföy Optimizasyonu. Yüksek Lisans Tezi, *Ankara Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Anabilim Dalı, Ankara.

Gökçe, G. A. (2001). Risk, Çeşitlendirme ve İMKB-30 Endeksinde İyi Çeşitlendirilmiş Bir Portföyün Büyüklüğünün Hesaplanması. Yayınlanmamış Yüksek Lisans Tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, Muhasebe, Finansman ve Sayısal Yöntemler Anabilim Dalı, İstanbul.

Gökgöz, E. (2006). *Riske Maruz Değer (VaR) ve Portföy Optimizasyonu*. Ankara: Sermaye Piyasası Kurulu.

Gülseçen, S. (1993). Yapay Sinir Ağları, İşletme Alanında Uygulanması ve Bir Örnek Çalışma. Basılmamış Doktora Tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Fakültesi Sayısal Yöntemler Anabilim Dalı, İstanbul.

Hagan, M. T., Demuth, H. B. and Beale, M. (1996). *Neural Network Design*. USA: PWS Publishing Company.

Hamzaçebi, C. (2011). *Yapay Sinir Ağları: Tahmin Amaçlı Kullanımı, Matlab ve Neurosolutions Uygulamalı*. Bursa: Ekin Yayınevi.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. New Jersey: Prentice-Hall.

Hebb, D. (1949). *The Organization of Behavior*. New York: Wiley.

Hertz, J. A., Krogh, A. S. and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley Publishing Company.

İskenderoğlu, Ö. ve Karadeniz, E. (2011). Optimum Portföyün Seçimi: İMKB-30 Üzerinde bir Uygulama. *C.Ü. İktisadi ve İdari Bilimler Dergisi*, 12(2), 235-257.

İstanbul Menkul Kıymetler Borsası (İMKB). Hisse Senetleri Piyasası / Gelişen İşletmeler Piyasası / Serbest İşlem Platformu Verileri. [online]. (24.01.2012). <http://www.imkb.gov.tr/Data/StocksData.aspx>

Jain, A. K., Mao, J. and Mohiuddin, K. M. (1996). Artificial Neural Network: A Tutorial. *IEEE Computer*, 31-44.

Jakobs, B. I., Levy, K. N. and Markowitz, H. M. (2005). Portfolio Optimization with Factors, Scenarios and Realistic Short Positions. *Operations Research*, 53(4), 586-599.

James, W. (1890). *Psychology (Briefer Course)*. NewYork: Holt, XVI, 253-279.

Kamruzzaman, J., Begg, R. K. and Sarker R. A. (2006). *Artificial Neural Networks In Finance and Manufacturing*. USA: Idea Group Publishing.

Karan, M. B. (2004). *Yatırım Analizi ve Portföy Yönetimi*. Ankara: Gazi Kitabevi.

Karaoglan, A. D. (2011). An Integrated Neural Network Structure For Recognizing Autocorrelated and Trending Processes, *Mathematical & Computational Applications*, 16 (2), 514-523.

Kaynar, O. ve Taştan, S. (2009). Zaman Serisi Analizinde MLP Yapay Sinir Ağları ve ARIMA Modelinin Karşılaştırılması. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 33, 161-172.

Kiang, M. Y. (2001). Extending the Kohonen Self-Organizing Map Networks for Clustering Analysis. *Computational Statistics & Data Analysis*, 38, 161-180.

Klimasauskas, C. C. (1996). Applying Neural Network. (eds: R. R. Trippi and E. Turban), *Neural Network in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance*. USA: Irwin Professional Publishing Co, 45, 69.

Kohonen, T. (1982). Self-organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43, 59-69.

Korkmaz, T. ve Ceylan, A. (2006). *Sermaye Piyasası ve Menkul Değer Analizi*. Bursa: Ekin Kitabevi.

Kröse, B. and Smagt, P. V. D. (1996). *An Introduction to Neural Networks*. Netherlands: University of Amsterdam.

Küçükönder, H. (2011). Yapay Sinir Ağları ve Tarımda Bir Uygulama. Doktora Tezi, *Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü*, Zootekni Anabilim Dalı, Kahramanmaraş.

Lippmann, R. P. (1987). An Introduction to Computing with Neural Nets. *ASSP Magazine, IEEE*, 4(2), 2-22.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91.

Martin, E. D. and Araque, A. (2006). Astrocytes and the Biological Neural Networks. (eds: J. R. Rabunal and J. Dorado), *Artificial Neural Networks in Real-Life Applications*, USA: Idea Group Publishing, 22-45.

McNelis, P. D. (2005). *Neural Network in Finance: Gaining Predictive Edge in The Market*, USA: Elsevier Akademik Press.

Medsker, L., Turban, E. and Trippi, R. R. (1996). Neural Network Fundamentals for Financial Analysts. (eds: R. R. Trippi and E. Turban), *Neural Network in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance*. USA: Irwin Professional Publishing Co, 1-24.

Minsky, M. and Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Ndabahaliye, A. (2002). Number of Neuron In a Human Brain [online]. (1 Nisan 2012), <http://hypertextbook.com/facts/2002/AniciaNdabahaliye2.shtml>

Nilson, N. J. (1965). *Learning Machines*. Columbus: McGraw-Hill.

Oğuzlar, A. (2004). *Veri Madenciliğine Giriş*. Bursa: Ekin Kitabevi.

Oh, K. J., Kim, T. Y. and Kim, C. (2006). An Early Warning System for Detection of Financial Crisis Using Financial Market Volatility. *Expert Systems*, 23(2), 83-98.

Öcal, O. (2007). Yapay Sinir Ağları Algoritması Kullanılarak Akarsu Havzalarında Yağış-Akış-Katı Madde İlişkisinin Belirlenmesi. Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, İnşaat Mühendisliği Anabilim Dalı, Denizli.

Özçam, M. (1997). *Varlık Fiyatlama Aracılığıyla Dinamik Portföy Yönetimi*. Ankara: Sermaye Piyasası Kurulu.

Öztemel, E. (2012). *Yapay Sinir Ağları*. İstanbul: Papatya Yayıncılık.

Radlak, M. (2008). Artificial Intelligence with Modern Portfolio Theory. MSc Thesis, *The University of Birmingham*, Business School, England.

Rosenblatt, F. (1959). Two Theorems of Statistical Separability in the Perceptron. *Proceedings of a Symposium on the Mechanization of Thought Processes*. Her Majesty's Stationary Office, London, 421-456.

Rosenblatt, F. (1961). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. New York: Cornell Aeronautical Laboratory.

Roudier, F. (2007). Portfolio Optimization and Genetic Algorithms. MSc Thesis. *Swiss Federal Institute of Technology*, Zurich.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, vol. 323, 533-536.

Sağıroğlu, Ş., Beşdok, E. ve Erler, M. (2004). *Mühendislikte Yapay Zeka Uygulamaları I: Yapay Sinir Ağları*. Kayseri: Ufuk Kitap Kırtasiye.

Samuel, L. (2009). The Nervous System / What is a Neuron? [online]. (20 Nisan 2012), <http://www.interactive-biology.com/44/what-is-a-neuron>

Saraç, T. (2004). *Yapay Sinir Ağları*. Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Anabilim Dalı, Ankara.

Sevinç, E. (2007). İMKB-30 Endeksinde Yer Alan Menkul Kıymetlerden Ortalama-Varyans Modeline Göre Optimal Portföy Oluşturulması ve Riske Maruz Değer Yaklaşımıyla Portföy Riskinin Hesaplanması. Yüksek Lisans Tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Anabilim Dalı, İstanbul.

Specht, D. F. (1988). Probabilistic neural Networks for Classification, Mapping, or Associative Memory. San Diego: *IEEE Conference on Neural Networks*, I, 525-532.

Specht, D. F. (1991). A General Regression Neural Network. *IEEE Transactions on Neural Networks*, 2(6), 568-576.

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

Tan, C. N. W. (1997). An Artificial Neural Networks Primer with Financial Applications Examples in Financial Distress Predictions and Foreign Exchange Hybrid Trading System. PhD Thesis, *Bond University*, Australia.

Tarassenko, L. (1998). *A Guide to Neural Computing Applications*. London: Arnold.

Tektaş, A. ve Karataş, A. (2004). Yapay Sinir Ağları ve Finans Alanına Uygulanması: Hisse Senedi Fiyat Tahminlemesi. *İktisadi ve İdari Bilimler Dergisi*, 18 (3-4), 337-349.

Teziş, F. (1987). Hisse Senetlerinde Risk Türlerinin Ölçülmesi. *Para ve Sermaye Piyasası Dergisi*, 9(99), s. 32.

U.S. Defense Advanced Research Projects Agency (DARPA). (1989). DARPA Neural Network Study. *Lincoln Laboratory Technical Report, ESD-TR-88-311*, Massachusetts Institute of Technology, Lexington, Massachusetts.

Ulucan, A. (2004). *Portföy Optimizasyonu - Kuadratik Programlama Tabanlı Modelleme*. Ankara: Siyasal Kitabevi.

Vural, B. B. (2007). Yapay Sinir Ağları İle Finansal Tahmin. Yüksek Lisans Tezi, *Ankara Üniversitesi Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı*, Ankara.

Yurtoğlu, H. (2005). Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği. Uzmanlık Tezi, *Ekonomik Modeller ve Stratejik Araştırmalar Genel Müdürlüğü*, Ankara.

EKLER

9. EKLER

EK – A: XOR Probleminin Çözümü

1. adım: Örneklerin Toplanması:

XOR problemi için 4 örnek vardır. Bunlar 1 ve 0 değerlerinden oluşmaktadır. Her örnek için girdiler ve beklenen çıktı şöyledir:

XOR	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

2. adım: Ağın Topolojik Yapısının Belirlenmesi:

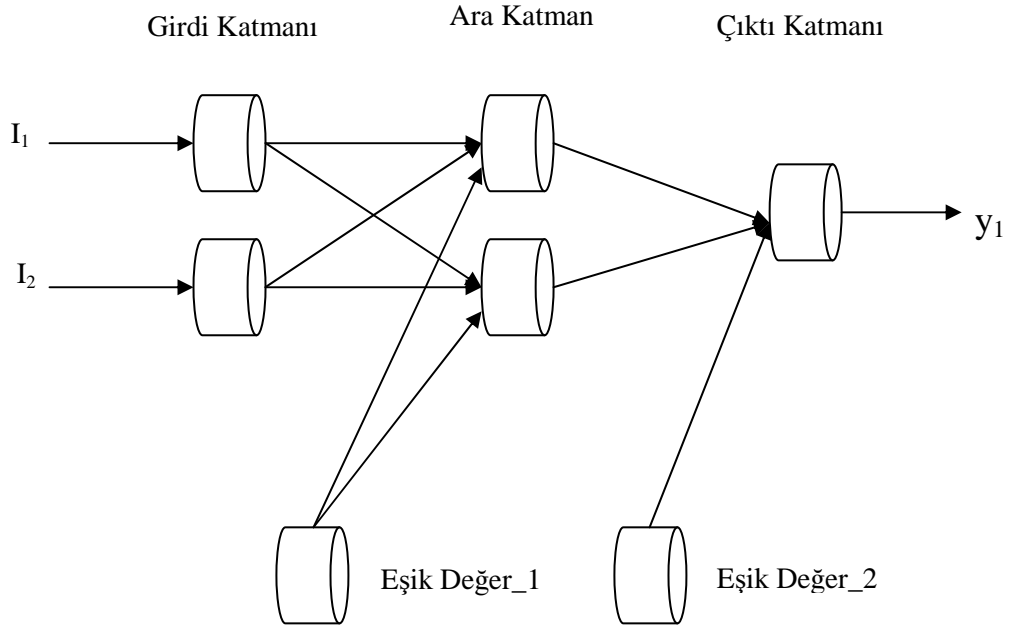
XOR probleminde 2 girdi ve 1 çıktı olduğundan oluşturulacak olan MLP ağının da 2 girdi ünitesi ve 1 çıktı ünitesi olacaktır. 1 ara katman ve 2 tane de ara katman proses elemanının bu problemi çözebileceğini varsayalım.

3. adım: Öğrenme Parametrelerinin Belirlenmesi:

Oluşturulan ağ için aktivasyon fonksiyonu olarak sigmoid fonksiyonunun kullanıldığını, öğrenme (η) ve momentum (α) katsayısının ise şu şekilde belirlendiği varsayalım:

$$\eta = 0.5$$

$$\alpha = 0.8$$



4. adım: Ağırlıkların Başlangıç Değerlerinin Atanması:

Oluşturulan ağ için ağırlık vektörleri ve başlangıç değerleri de şu şekilde belirlenmiş olsun:

Girdi katmanı ile ara katman arasındaki ağırlıklar W^i ile gösterilsin:

$$W^i = \begin{bmatrix} 0.129952 & 0.570345 \\ -0.923123 & 0.328932 \end{bmatrix}$$

Çıktı katmanı ile ara katman arasındaki ağırlıklar ise W^a ile gösterilsin:

$$W^a = \begin{bmatrix} 0.164732 & 0.752621 \end{bmatrix}$$

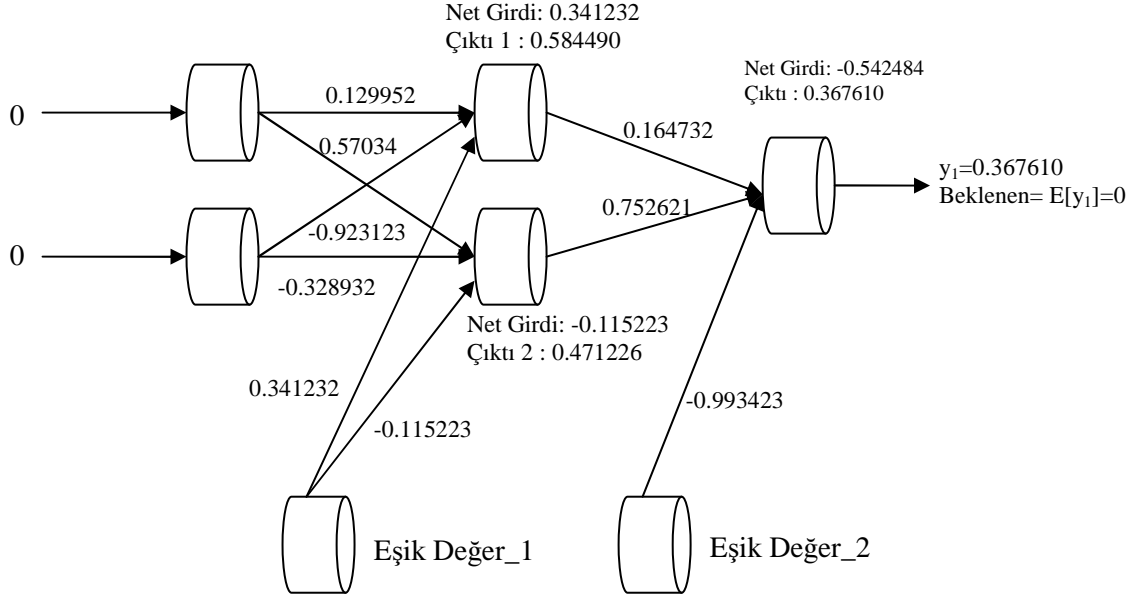
Eşik değer ağırlıkları şöyle olsun:

$$b^a = \begin{bmatrix} 0.341332 & -0.115223 \end{bmatrix}$$

$$b^y = \begin{bmatrix} -0.993423 \end{bmatrix}$$

5. adım: Örneklerin Ağa Gösterilmesi ve İleri Doğru Hesaplama:

Birinci örnek $I_1 = 0, I_2 = 0$ ve $b = 0$ olarak belirlenmiştir. Bu örnek ağa gösterilirse ileri doğru hesaplama şu şekilde olacaktır:



Ara katman ünitelerinin NET girdileri (eşik değer ünitesinin ağırlık değerleri eklenmiş olarak) şu şekilde hesaplanır:

$$\text{Net1} = (0 * 0.129952) + (0 * -0.923123) + (1 * 0.341232) = 0.341232$$

$$\text{Net2} = (0 * 0.570345) + (0 * -0.328932) + (1 * -0.115223) = -0.115223$$

Ara katman ünitelerinin çıktıları ise şöyle hesaplanır:

$$C_1 = \frac{1}{1 + e^{-0.341232}} = 0.584490$$

$$C_2 = \frac{1}{1 + e^{0.115223}} = 0.471226$$

Çıktı katmanındaki proses elemanlarının NET girdisi hesaplanırsa:

$$\text{Net} = (1 * -0.993423) + (0.584490 * 0.164732) + (0.471226 * 0.752621) = -0.542484$$

ve ağıın çıktısı

$$y_1 = \frac{1}{1 + e^{0.542484}} = 0.367610$$

Beklenen çıktısı = 0 olduğundan,

$$\text{Bu durumda Hata} = e = 0 - 0.367610 = -0.367610$$

Bu hatanın geriye doğru yayılması sonucu ara katman ile çıktı katmanı arasındaki ağırlıkların değişim miktarları şu şekilde hesaplanır:

$$\delta_1^{y_1} = y_1(1 - y_1)e$$

$$\delta_1^{y_1} = 0.367610 * (1 - 0.367610) * (-0.367610)$$

$$\delta_1^{y_1} = -0.085459$$

Ağırlık değişimine ilişkin önceden verilmiş olan formül hatırlanırsa:

$$\Delta w_{ay}(t) = \lambda \delta_1^{y_1} y_a + \alpha \Delta w_{ay}(t-1)$$

y_a : ara katmandaki proses elemanının çıktısı

$\Delta w_{ay}(t)$: ara katmandaki proses elemanı ile çıktı katmanındaki proses elemanı arasındaki bağlantının ağırlığındaki değişim miktarı

$\Delta w_{ay}(t-1)$: Bir önceki iterasyonda yapılmış olan değişim miktarı

Ara katman 1. proses elemanı ile çıktı katmanı 1. proses elemanı arasındaki ağırlıklarda yapılacak değişim miktarı:

$$\Delta w_{11}^a(t) = 0.5 * -0.085459 * 0.584490 + 0.8 * 0 = -0.024875$$

Ara katman 2. proses elemanı ile çıktı katmanı 1. proses elemanı arasındaki ağırlıklarda yapılacak değişim miktarı:

$$\Delta w_{21}^a(t) = 0.5 * -0.085459 * 0.471226 + 0.8 * 0 = -0.020135$$

Çıktı katmanı proses elemanı eşik değer ağırlıklarındaki değişim ise

$$\Delta b^{y_1}(t) = \lambda \delta_1^{y_1} + \alpha \Delta b^{y_1}(t-1)$$

b^{y_1} : Çıktı katmanı 1. proses elemanına ait bias değerinin bağlantı ağırlığındaki değişim miktarı formülünden

$$\Delta b_1^{y_1}(t) = 0.5 * -0.085459 + 0.8 * 0 = -0.042730$$

Ağırlıkların bu değişim miktarları ile ara katman ve çıktı katmanı arasındaki ağırlıklar yeniden hesaplanabilir:

$$w_{11}^a(t) = w_{11}^a(t-1) + \Delta w_{11}^a(t) = 0.164732 - 0.024975 = 0.139757$$

$$w_{21}^a(t) = w_{21}^a(t-1) + \Delta w_{21}^a(t) = 0.752621 - 0.020135 = 0.732486$$

$$b_1^{y_1}(t) = b_1^{y_1}(t-1) + \Delta b_1^{y_1}(t) = -0.993423 - 0.042730 = -1.036153$$

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

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

$$\delta_1^a(t) = 0.584490 * (1 - 0.584490) * (0.164732) * (-0.085459) = -0.034190$$

$$\delta_2^a(t) = -0.160263$$

Ağırlık değişimine ilişkin önceden verilmiş olan formül hatırlanırsa: $\Delta w_{ia}^i(t) = \lambda \delta_1^a y_i + \alpha \Delta w_{ia}^i(t-1)$ (i: girdi katmanı, a: ara katman, y: çıktı katmanı indisleri)

$$y_i = \text{girdi} = 0$$

$$\Delta w_{11}^i(t) = 0.5 * -0.034190 * 0 + 0.8 * 0 = 0$$

$$\Delta w_{12}^i(t) = 0.5 * -0.034190 * 0 + 0.8 * 0 = 0$$

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

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

$$\Delta b_1^i(t) = \lambda \delta_1^a + \alpha \Delta b_1^i(t-1)$$

$$\Delta b_1^i(t) = 0.5 * 1 * -0.034190 = -0.017095$$

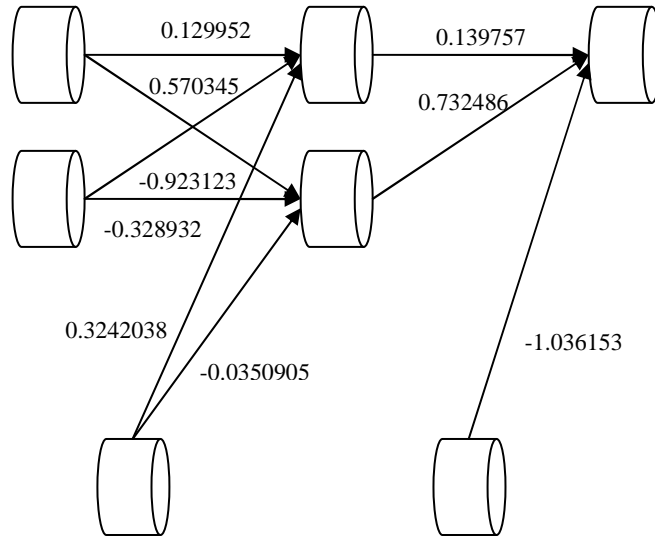
$$\Delta b_2^i(t) = 0.5 * 1 * -0.160263 = -0.080132$$

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

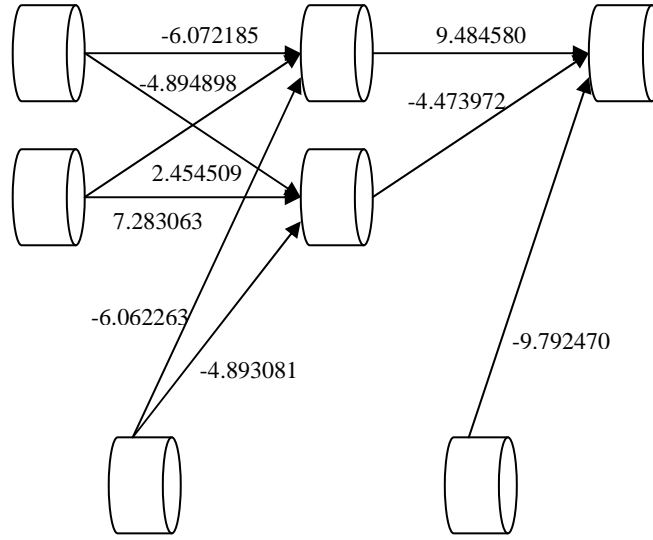
$$b_1^i(t) = 0.341232 - 0.017095 = 0.3242038$$

$$b_2^i(t) = 0.115223 - 0.081325 = -0.0350905$$

Aşağıdaki şekilde yeni belirlenen ağırlık değerleri gösterilmektedir:



Birinci iterasyon bittikten sonra ikinci iterasyon başlayacaktır. Bu kez ikinci örnek ağa gösterilir. $I_1=0$, $I_2=1$ ve Beklenen Çıktı=1. Yukarıdaki işlemler aynı şekilde tekrar edilir. Bu iterasyonlar bütün çıktılar doğru cevap verinceye kadar devam etmelidir. Aşağıdaki şekil bu şekilde eğitilmiş ağı (XOR problemi öğrenildikten sonra) en son ağırlık değerlerini göstermektedir:



Bu ağırlıklar ile girdiler ağı tekrar gösterildiğinde aşağıdaki sonuçlar elde edilir. Bu sonuçlar ağın problemi çok düşük hatalar ile çözebilecek şekilde öğrendiğini göstermektedir:

ÇÖZÜM XOR	Girdi 1	Girdi 2	Beklenen Çıktı	Ağın Çıktısı	Hata
Örnek 1	0	0	0	0.017622	-0.017
Örnek 2	1	0	1	0.981504	0.018
Örnek 3	0	1	1	0.981491	0.018
Örnek 4	1	1	0	0.022782	-0.020

Kaynak: (Öztemel, 2012, s. 85-90)

EK – B: Çalışmada Kullanılan Hisse Senetlerinin Yıllık (2010 Yılına Ait) Ortalama Getiri ve Riskleri

Hisse Senedi Kodu	Hisse Senedi Simgesi	Yıllık Ortalama Getiri (2010)	Getirilerin Standart Sapması (Risk)	Hisse Senedi Kodu	Hisse Senedi Simgesi	Yıllık Ortalama Getiri (2010)	Getirilerin Standart Sapması (Risk)	Hisse Senedi Kodu	Hisse Senedi Simgesi	Yıllık Ortalama Getiri (2010)	Getirilerin Standart Sapması (Risk)
1	ALYAG	6.765	21.49	48	KOZAA	0.19	9.66	95	KONYA	13.43	25.25
2	AEFES	3.19	7.33	49	TIRE	-1.31	7.29	96	KUTPO	0.37	7.47
3	BANVT	3.56	11.37	50	OLMKS	4.23	11.11	97	MRDIN	2.04	4.61
4	CCOLA	3.14	9.39	51	VKING	1.6	12.84	98	NUHCM	0.97	6.93
5	ERSU	5.75	21.03	52	GENTS	5.2	9.46	99	TRKCM	5.28	8.68
6	FRIGO	-0.91	9.56	53	KLBMO	4.58	13.37	100	USAK	4.97	9.32
7	KENT	22.766	59.41	54	ADEL	7.73	9.2	101	UNYEC	3.42	10.63
8	KERVT	7.28	13.47	55	SERVE	0.17	8.77	102	BRSAN	4.94	18.67
9	KNFRT	4.7	13.75	56	SASA	7.27	20.29	103	BURCE	19.92	38.24
10	KRSTL	7.15	18.81	57	AKSA	4.04	12.86	104	BURVA	4.33	22.25
11	MERKO	-0.27	8.88	58	ALKIM	1.73	7.31	105	COMDO	1.49	7.61
12	PENGD	1.69	9.61	59	AYGAZ	3.93	7.8	106	CELHA	3.19	10.87
13	PETUN	4.83	7.98	60	BAGFS	5.33	9.68	107	CEMTS	3.16	11.72
14	PINSU	0.46	8.27	61	BRISA	7.79	14.38	108	DMSAS	5.82	15.39
15	PNSUT	6.91	7.85	62	DEVA	0.87	7.1	109	ERBOS	6.32	15.13
16	SKPLC	2.72	17.57	63	DYOBY	2.88	11.3	110	EREGL	1.36	7.85
17	TATKS	3.32	9.44	64	ECILC	0.52	7.75	111	FENIS	-0.07	10.63
18	TBORG	1.26	11.4	65	EGGUB	5.01	12.57	112	IZMDC	4.02	9.16
19	TUKAS	3.68	11.39	66	GOODY	6.22	12.31	113	KRDMA	4.46	14.06
20	ULKER	4.37	10.71	67	GUBRF	7.33	12.42	114	KRDMB	5.27	15.98
21	AKALT	-0.61	8.45	68	HEKTS	5.48	6.29	115	KRDMD	1.87	9.16
22	ATEKS	4.59	15.2	69	MRSHL	10.14	15.35	116	SARKY	3.95	7.59
23	ALTIN	8.84	15.72	70	PETKM	4.65	12.35	117	ALCAR	4.05	10.12
24	ARSAN	8.74	28.13	71	PTOFS	2.72	7.93	118	ASUZU	3.51	13.24
25	BOSSA	-1.23	8.36	72	PIMAS	0.62	7.66	119	ARCLK	2.93	8.18
26	DERIM	6.95	19.45	73	SODA	3.84	5.55	120	BFREN	10.46	17.77
27	DESA	5.31	12.09	74	TRCAS	2.94	9.03	121	BSHEV	6.13	6.13
28	IDAS	-0.67	9.49	75	TUPRS	3.25	10.48	122	DITAS	2.53	7.5
29	KRTEK	0.85	17.44	76	ADANA	1.47	4.12	123	EGEEN	11.09	20.98
30	KORDS	3.41	7.23	77	ADBGR	2.04	6.9	124	EMKEL	2.04	14.44
31	LUKSK	5.61	12.05	78	ADNAC	5.31	13.13	125	EMNIS	-0.24	6.92
32	MNDRS	2.07	11.16	79	AFYON	66.81	247.94	126	FMIZP	4.54	10.76
33	MTEKS	3.65	18.13	80	AKCNS	1.84	9.85	127	FROTO	4.29	9.94
34	SKTAS	11.28	22.63	81	ANACM	5.21	9.12	128	GEREL	1.08	14.92
35	YATAS	2.15	10.9	82	BTCIM	2.03	7.73	129	IHEVA	1.93	10.01
36	YUNSA	5.47	10.35	83	BSOKE	3.11	8.57	130	KARSN	2.27	8.88
37	ALKA	1.03	9.53	84	BOLUC	0.81	8.78	131	KLMSN	1.5	8.72
38	BAKAB	3.56	20.44	85	BUCIM	0.28	6.38	132	MUTLU	4.84	13.3
39	DENTA	7.31	18.32	86	CMBTN	8.37	19.17	133	OTKAR	3.4	8.31
40	DOBUR	4.25	15.86	87	CMEN	-1.71	10.03	134	PARSN	4.9	16.25
41	DGZTE	-0.94	9.75	88	CIMSA	3.31	9.48	135	TUDDF	8.7	22.88
42	DURDO	4.05	14.65	89	DENCM	16.69	9.16	136	TOASO	5.4	11.47

43	HURGZ	0.8	10.29	90	ECYAP	1.32	7.45	137	PRKAB	2.11	8.58
44	IHGZT	16.03	25.25	91	EGSER	7.98	14.67	138	TTRAK	8.9	10.22
45	IPEKE	2.19	12.93	92	GOLTS	5.89	15.81	139	VESTL	-0.12	8.88
46	KAPLM	5.57	11.29	93	HZNDR	2.26	12.36	140	VESBE	2.88	13.99
47	KARTN	10.97	21.2	94	IZOCM	4.01	10.15				

EK – C: Çalışmada Kullanılan Bir Portföyü Oluşturan Hisse Senetlerinin Aylık Ortalama Getirileri ve Varyans-Kovaryans Matrisi

Aylar	MERKO	ALYAG	DERIM	DOBUR	KAPLM	HEKTS	TUPRS	EGSER	IZOCM	KUTPO	ERBOS	GEREL	EMKEL	EMNIS
1	4.84	9.91	16.24	7.12	13.78	5.41	5.75	15.46	29.66	4.94	27.32	-9.43	12.63	0.00
2	-4.62	-6.72	-16.17	0.34	5.83	2.07	-5.19	17.58	-0.75	-1.22	-10.87	-7.56	-31.65	-1.82
3	0.78	-2.46	-3.69	4.26	-4.76	5.84	-1.28	6.45	-1.85	9.33	0.88	7.50	6.11	11.01
4	4.88	-1.61	19.02	18.49	8.62	2.24	13.87	7.64	0.93	6.64	2.70	0.63	14.91	3.81
5	-8.89	0.00	-16.33	-4.80	3.57	2.29	0.00	15.20	-1.83	-4.95	5.71	1.92	-10.94	-0.94
6	4.65	-8.15	36.87	8.70	-5.88	3.15	18.10	4.17	0.93	0.45	1.94	14.71	23.08	-3.64
7	7.50	62.65	14.74	-2.13	6.25	6.72	0.87	-1.64	8.00	3.76	5.64	0.00	0.00	6.80
8	-15.49	-16.16	-26.07	-14.78	3.70	9.29	-13.53	-17.01	-6.54	-18.08	-17.22	-23.60	-7.14	-15.57
9	0.71	30.26	32.70	39.56	-9.24	20.00	2.59	2.08	-7.76	-7.80	0.81	11.95	6.67	1.67
10	15.57	18.75	17.78	21.62	31.49	7.95	23.21	28.57	0.74	7.28	38.98	11.97	7.14	6.19
11	-15.28	-16.88	-7.53	-20.60	-5.73	-8.33	-10.40	-15.79	10.19	-1.60	-1.67	-24.06	-10.91	-9.60
12	2.13	11.59	15.87	-6.80	19.25	9.09	5.04	33.00	16.38	5.65	21.62	28.97	14.58	-0.79

Var-Cov.	MERKO	ALYAG	DERIM	DOBUR	KAPLM	HEKTS	TUPRS	EGSER	IZOCM	KUTPO	ERBOS	GEREL	EMKEL	EMNIS
MERKO	0.0079	0.0114	0.0129	0.0092	0.0053	0.0020	0.0080	0.0080	0.0019	0.0047	0.0097	0.0087	0.0077	0.0047
ALYAG	0.0114	0.0462	0.0197	0.0132	0.0054	0.0069	0.0057	0.0062	0.0031	0.0043	0.0123	0.0120	0.0062	0.0085
DERIM	0.0129	0.0197	0.0378	0.0214	0.0014	0.0046	0.0155	0.0081	0.0040	0.0063	0.0144	0.0186	0.0224	0.0058
DOBUR	0.0092	0.0132	0.0214	0.0252	0.0010	0.0064	0.0105	0.0075	-0.0042	0.0026	0.0076	0.0118	0.0098	0.0060
KAPLM	0.0053	0.0054	0.0014	0.0010	0.0127	0.0006	0.0056	0.0115	0.0043	0.0034	0.0128	0.0043	0.0021	0.0017
HEKTS	0.0020	0.0069	0.0046	0.0064	0.0006	0.0040	0.0011	0.0019	-0.0017	-0.0010	0.0013	0.0045	0.0028	0.0011
TUPRS	0.0080	0.0057	0.0155	0.0105	0.0056	0.0011	0.0110	0.0091	0.0010	0.0046	0.0109	0.0103	0.0103	0.0038
EGSER	0.0080	0.0062	0.0081	0.0075	0.0115	0.0019	0.0091	0.0215	0.0043	0.0064	0.0157	0.0155	0.0043	0.0052
IZOCM	0.0019	0.0031	0.0040	-0.0042	0.0043	-0.0017	0.0010	0.0043	0.0103	0.0035	0.0085	-0.0002	0.0039	0.0003
KUTPO	0.0047	0.0043	0.0063	0.0026	0.0034	-0.0010	0.0046	0.0064	0.0035	0.0056	0.0071	0.0054	0.0045	0.0040
ERBOS	0.0097	0.0123	0.0144	0.0076	0.0128	0.0013	0.0109	0.0157	0.0085	0.0071	0.0229	0.0112	0.0109	0.0049
GEREL	0.0087	0.0120	0.0186	0.0118	0.0043	0.0045	0.0103	0.0155	-0.0002	0.0054	0.0112	0.0222	0.0124	0.0060
EMKEL	0.0077	0.0062	0.0224	0.0098	0.0021	0.0028	0.0103	0.0043	0.0039	0.0045	0.0109	0.0124	0.0208	0.0031
EMNIS	0.0047	0.0085	0.0058	0.0060	0.0017	0.0011	0.0038	0.0052	0.0003	0.0040	0.0049	0.0060	0.0031	0.0048