

T.C.
İstanbul Üniversitesi
Sosyal Bilimler Enstitüsü
İşletme Anabilim Dalı
Sayısal Yöntemler Bilim Dalı

Doktora Tezi

Yapay Sinir Ağları ve Destek Vektör Makineleri
Yöntemlerinin Sınıflandırma Performanslarının
Karşılaştırılması: Borsa Endeks Yönünün Tahmini
Üzerine Bir Uygulama

Şenol EMİR
2502070268

Tez Danışmanı
Prof. Dr. Mehpere TİMOR

İstanbul, 2013



DOKTORA

TEZ ONAYI

ÖĞRENCİNİN

Adı ve Soyadı :ŞENOL EMİR Numarası :2502070268
Anabilim/Bilim Dalı :SAYISAL YÖNTEMLER Danışman :Prof.Dr.Mehpare Timor
Tez Savunma Tarihi :27.06.2013 Saati :13:30

Tez Başlığı :YAPAY SİNİR AĞLARI VE DESTEK VEKTÖR MAKİNELERİ
YÖNTEMLERİNİN SINIFLANDIRMA PERFORMANSLARININ KARŞILAŞTIRILMASI: BORSA
ENDEKS YÖNÜNÜN TAHMİNİ ÜZERİNE BİR UYGULAMA.

TEZ SAVUNMA SINAVI, Lisansüstü Öğretim Yönetmeliği'nin 50. Maddesi uyarınca yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin KABULÜ'NE OYBİRLİĞİ / OYÇOKLUĞUYLA karar verilmiştir.

JÜRİ ÜYESİ	İMZA	KANAATI (KABUL / RED / DÜZELTME)
PROF.DR.ÖNER ESEN		Kabul
PROF.DR.MEHPARE TİMOR		Kabul
PROF.DR.SEVİNÇ GÜLSEÇEN		Kabul
DOÇ.DR.DİLEK ALTAŞ	— (*)	—
DOÇ.DR.TUĞBA GÜRSOY		Kabul

YEDEK JÜRİ ÜYESİ	İMZA	KANAATI (KABUL / RED / DÜZELTME)
DOÇ.DR.SEDA TOLUN	—	—
DOÇ.DR.ZUHAL TANRIKULU		Kabul

(*) Doç. Dr. Dilek Altaş, M.Ü'ndeki Yaz Okulu derslerinin çekilmesi nedeniyle sınava katılmamıştır.

Yapay Sinir Ağları ve Destek Vektör Makineleri Yöntemlerinin Sınıflandırma Performanslarının Karşılaştırılması: Borsa Endeks Yönünün Tahmini Üzerine Bir Uygulama

Şenol EMİR

ÖZ

Bu tez çalışmasında, makine öğrenmesi yöntemlerinden olan Yapay Sinir Ağları ve Destek Vektör Makineleri incelenmiştir. Bu yöntemlerin İstanbul Menkul Kıymetler Borsası Ulusal 100 Endeksi'nin yön tahmininde kullanılabilirlikleri araştırılmıştır. Analizde 2005-2011 dönemine ait sırasıyla teknik göstergeler, borsaların endeks değişim oranları ve makroekonomik göstergelerden oluşan veri kümelerinin yön tahmininde gösterdikleri performanslar karşılaştırılmıştır. Sonuçlar incelendiğinde teknik göstergelerin diğer değişkenlere göre daha iyi performans gösterdiği belirlenmiştir. Daha sonra bu veri kümelerinin bir araya getirilmiş hali üzerinde öznitelik seçimi yapılarak yön tahmininin en az sayıdaki değişkenle en doğru şekilde yapılmasını sağlayacak veri kümesi oluşturulmuştur. Tüm analizlerde Yapay Sinir Ağları, Destek Vektör Makineleri'ne göre daha iyi sonuçlar vermiştir.

**Classification Performance Comparison of Artificial Neural Networks and
Support Vector Machines Methods: An Empirical Study on Predicting Stock
Market Index Movement Direction**

Şenol EMİR

ABSTRACT

In this thesis study, Artificial Neural Networks and Support Vector Machines which are machine learning methods were examined. Usability of these methods for the prediction of Istanbul Stock Exchange (ISE) National 100 Index movement direction was investigated. In analysis, performances of the 2005–2011 period data sets containing technical indicators, rate of change of stock market indices and macroeconomic indicators were compared. The results showed that technical variables give better performances than other variables. Later, a dataset which predicts the stock movement direction most accurately with minimum number of variables was formed by feature selection on combined data sets. Artificial Neural Networks gave better results than Support Vector Machines for all analyzes.

ÖNSÖZ

Bu tez çalışmasında son yıllarda hisse senedi piyasalarında tahmin yapmak için daha sık kullanılmaya başlanan Yapay Sinir Ağları ve Destek Vektör Makineleri yöntemlerinin İstanbul Menkul Kıymetler Borsası (İMKB) Ulusal 100 Endeksi'nin yönünün tahmininde gösterdikleri performanslar karşılaştırılmıştır. Ayrıca, endeks yönünün tahmin edilmesinde hangi tür değişkenlerin daha belirleyici olduğunun ortaya çıkarılması amaçlanmıştır.

Çalışmanın teorik bölümlerinde sırasıyla makine öğrenmesi ile ilgili temel kavramlar, Yapay Sinir Ağları, İstatistiksel Öğrenme Teorisi, Destek Vektör Makineleri konuları ele alınmıştır. Uygulamada ise İMKB-100 Endeksi'nin yön tahmininde kullanılmak üzere 2005-2011 dönemine ait teknik göstergeler, borsaların endeks değişim oranları ve makroekonomik göstergelerden oluşan veri kümeleri oluşturulmuştur. Bu veri kümeleri üzerinde Yapay Sinir Ağları ve Destek Vektör Makineleri yöntemleri çalıştırılarak her bir veri kümesinin yön tahmin performansları bulunmuştur. Elde edilen sonuçlar ışığında teknik göstergelerin yer aldığı veri kümesinin yön tahmininde diğerlerine göre daha faydalı olduğu sonucuna varılmıştır. Hem Yapay Sinir Ağları'nın hem de Destek Vektör Makineleri'nin yön tahmininde faydalı yöntemler olmakla birlikte Yapay Sinir Ağları'nın daha iyi sınıflandırma performansı gösterdiği görülmüştür.

Tez çalışmam boyunca desteğini esirgemeyen çok değerli danışmanım Sayın Prof. Dr. Mehpare Timor'a, teknik analiz göstergelerinin elde edilmesinde yardımcı olan ve uygulamanın finansal yönü hakkında fikirlerini paylaşan değerli çalışma arkadaşım Yard. Doç. Dr. Hasan Dinçer'e, beni manevi olarak destekleyerek tez yazım süresince karşılaştığım sıkıntıları aşmamda yardımcı olan aileme, sevgili eşime ve oğlum Emre Deniz'e teşekkür ederim.

Şenol Emir
İstanbul, Haziran 2013

İÇİNDEKİLER

ÖZ.....	i
ABSTRACT.....	ii
ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	xiii
ŞEKİLLER LİSTESİ.....	xvi
KISALTMALAR LİSTESİ.....	xviii
GİRİŞ.....	1
1. MAKİNE ÖĞRENMESİNE GENEL BİR BAKIŞ.....	4
1.1 Yapay Zekâ.....	4
1.1.1 Yapay Zekâ Yaklaşımları	5
1.1.1.1 İnsan Gibi Düşünme (Bilişsel Modelleme) Yaklaşımı	5
1.1.1.2 İnsan Gibi Davranma (Turing Testi) Yaklaşımı	5
1.1.1.3 Rasyonel Düşünme (Düşünce Kanunları) Yaklaşımı	6
1.1.1.4 Rasyonel Davranma (Zeki Ajan) Yaklaşımı	6
1.1.2 Yapay Zekâ'nın Tarihsel Gelişimi.....	7
1.2 Makine Öğrenmesi	9
1.2.1 Makine Öğrenmesi'nin Önemi	9
1.2.2 Makine Öğrenmesi'nin Amacı.....	10
1.2.3 Makine Öğrenmesi'ne İhtiyaç Duyulan Durumlar	12
1.2.3.1 Alan Uzmanlığının Olmadığı Durumlar	12
1.2.3.2 Uzmanlığın Açıklanamadığı Durumlar.....	12
1.2.3.3 Problemin Hızlı Değiştiği Durumlar	13

1.2.3.4 Özelleştirme Gerektiren Durumlar.....	13
1.2.4 Makine Öğrenmesi ve Veri Madenciliği	13
1.2.5 Makine Öğrenmesinin Diğer Bilimlerle İlişkisi	14
1.2.6 Öğrenme Probleminin Genel Çerçevesi	15
1.2.6.1 Üreteç	16
1.2.6.2 Sistem.....	16
1.2.6.3 Öğrenme Makinesi.....	16
1.2.7 Tümevarımsal (Inductive) Öğrenme.....	17
1.2.8 Temel Öğrenme Problemleri	21
1.2.8.1 Sınıflandırma.....	21
1.2.8.2 Regresyon.....	23
1.2.8.3 Yoğunluk Tahmini	23
1.2.8.4 Kümeleme	24
1.2.9 Öğrenme Yöntemleri	25
1.2.9.1 Denetimli (Supervised) Öğrenme	25
1.2.9.2 Denetimsiz (Unsupervised) Öğrenme	26
1.2.9.3 Yarı Denetimli (Semi Supervised) Öğrenme	26
1.2.9.4 Transdüktif (Transductive) Öğrenme.....	27
1.2.9.5 Pekiştirmeli (Reinforcement) Öğrenme	27
1.2.9.6 Aktif (Active) Öğrenme	28
1.2.10 Öğrenme Algoritmalarının Bileşenleri	28
1.2.10.1 Gösterim.....	29
1.2.10.2 Değerlendirme (Amaç) Fonksiyonu.....	30
1.2.10.3 Optimizasyon Tekniği.....	31
1.2.11 Model Seçimi.....	32

1.2.11.1 Üçlü Ödünleşim	35
1.2.11.1.1 Eğitim Verisi Miktarı - Genelleme Ödünleşimi.....	35
1.2.11.1.2 Kapasite - Genelleme Ödünleşimi	35
1.2.11.1.3 Yanlılık - Varyans (Bias – Variance) Ödünleşimi	36
1.2.11.2 “No Free Lunch” Teoremi	38
1.2.11.3 Boyut Belası	39
1.2.11.4 Teorik Garantiler	40
1.2.11.5 Occam’ın Usturası.....	40
1.2.11.6 Bağımlılık ve Nedensellik.....	41
1.2.12 Model Geçerleme.....	41
1.2.12.1 Dışarıda Tutma (Holdout).....	42
1.2.12.2 Çapraz Geçerleme (Cross Validation)	43
1.2.12.3 Tabakalı Çapraz Geçerleme (Stratified Cross Validation).....	43
1.2.12.4 Birini Dışarıda Bırakma (Leave One Out).....	43
1.2.12.5 Yerine Koyarak Rastgele Örnekleme (Bootstrap)	44
1.2.13 Model Karmaşıklığını Düzenleme.....	44
1.2.13.1 Düzenleştirme (Regularization).....	44
1.2.13.2 Erken Durdurma (Early Stopping)	45
1.2.13.3 Yapısal Risk Minimizasyonu (Structural Risk Minimization)	45
1.2.13.4 Bayesyen Çıkarsama (Bayesian Inference).....	45
1.2.13.5 Minimum Tanım Uzunluğu (Minimum Description Length).....	46
2. YAPAY SINIR AĞLARI.....	47
2.1 Yapay Sinir Ağları Hakkında Temel Bilgiler.....	47
2.1.1 Biyolojik Sinir Ağları	48
2.1.2 Biyolojik ve Yapay Sinir Ağları Karşılaştırması.....	51

2.1.3 Yapay Sinir Ağlarının Tarihsel Gelişimi	53
2.1.3.1 1970 Öncesi Çalışmalar	54
2.1.3.2 1970 Sonrası Çalışmalar	55
2.1.4 Yapay Sinir Ağlarının Kullanım Alanları.....	56
2.1.5 Yapay Sinir Ağlarının Güçlü Yönleri	57
2.1.5.1 Öğrenme.....	57
2.1.5.2 Genelleme	58
2.1.5.3 Doğrusal Olmama	58
2.1.5.4 Kendi Kendini Organize Etme	58
2.1.5.5 Paralel Çalışma.....	58
2.1.5.6 Sağlamlık ve Hata Toleransı	59
2.1.6 Yapay Sinir Ağlarının Zayıf Yönleri	59
2.1.7 Yapay Sinir Ağlarının Temel Elemanları	60
2.1.7.1 Katmanlar	60
2.1.7.1.1 Girdi Katmanı.....	61
2.1.7.1.2 Gizli Katman	61
2.1.7.1.3 Çıktı Katmanı	62
2.1.7.2 Ağırlıklar	63
2.1.7.3 Birleştirme Fonksiyonu.....	64
2.1.7.4 Etkinlik (Aktivasyon) Fonksiyonu	65
2.1.7.5 Hata Fonksiyonu	68
2.1.8 Yapay Sinir Ağı Mimarileri.....	69
2.1.8.1 İleri Yönelimli Ağlar.....	70
2.1.8.2 Geri Yönelimli Ağlar	71
2.1.9 Yapay Sinir Ağlarında Öğrenme	72

2.1.9.1 Toplu (Batch) Öğrenme	73
2.1.9.2 Çevrimiçi (Online) Öğrenme	73
2.1.9.3 Öğrenme Kuralları	74
2.1.9.3.1 Hata Düzeltme Kuralı	74
2.1.9.3.2 Boltzmann Öğrenme Kuralı	74
2.1.9.3.3 Hebb Öğrenme Kuralı	75
2.1.9.3.4 Yarışmacı Öğrenme Kuralı	75
2.1.9.3.5 Hafıza Tabanlı Öğrenme Kuralı	75
2.2 Tek Katmanlı Yapay Sinir Ağları	76
2.2.1 Basit Algılayıcı (Perceptron)	76
2.2.1.1 Basit Algılayıcı Eşlemesinin Geometrik Yorumu	77
2.2.1.2 Basit Algılayıcının Gösterim Yeteneği	78
2.2.1.3 Basit Algılayıcı Öğrenme Kuralı	79
2.2.1.4 Basit Algılayıcı Yakınsama Teoremi	80
2.2.2 ADALINE (Adaptive Linear Neuron)	81
2.2.2.1 Delta Kuralı	81
2.2.2.2 Eğimli İniş (Gradient Descent) Yöntemi	81
2.2.2.2.1 Eğimli İniş Yönteminin Elde Edilmesi	82
2.2.2.3 Stokastik Eğimli İniş Yöntemi	84
2.3 Çok Katmanlı Algılayıcılar (Multi Layer Perceptrons)	85
2.3.1 Çok Katmanlı Algılayıcıların Gösterim Yeteneği	85
2.3.2 Gizli Birimlerin Etkisi	86
2.3.3 Çok Katmanlı Ağlarda Öğrenme	88
2.3.4 Geri Yayılım Algoritması	89
2.3.4.1 Geri Yayılım Algoritmasının Elde Edilmesi	90

2.3.4.2 Geri Yayılım Algoritmasının Genelleştirilmiş Hali.....	94
2.3.4.3 Geri Yayılım Algoritmasında Dikkat Edilmesi Gereken Konular	95
2.3.4.3.1 Yerel Minimum.....	95
2.3.4.3.2 İlk Ağırlıklar.....	96
2.3.4.3.3 Öğrenme Katsayısı	96
2.3.4.3.4 Momentum Terimi	97
2.3.4.3.5 Aşırı Öğrenme	99
2.3.4.3.6 Durdurma Zamanı	100
2.3.4.3.7 Ağırlık Sönümlenme	101
2.4 Yapay Sinir Ağları Uygulama Adımları.....	102
2.4.1 Değişken Seçimi	102
2.4.2 Veri Toplama	103
2.4.3 Veri Ön İşleme	103
2.4.4 Ağ Parametrelerinin Belirlenmesi	104
2.4.5 Ağın Eğitilmesi	106
2.4.6 Ağın Uygulanması	106
3. DESTEK VEKTÖR MAKİNELERİ.....	108
3.1 İstatistiksel Öğrenme Teorisi.....	108
3.1.1 Ampirik Risk Minimizasyonu (ARM) İlkesi.....	111
3.1.2 ARM İlkesinin Tutarlılık Şartları	112
3.1.3 Kapasite Ölçütleri	114
3.1.3.1 Büyüme Fonksiyonu	114
3.1.3.2 VC Boyutu	115
3.1.4 Genelleme Sınırları	119
3.1.5 Yapısal Risk Minimizasyonu (YRM) İlkesi	121

3.2 Maksimum Marj Sınıflandırıcıları.....	124
3.2.1 Marj - Karmaşıklık İlişkisi.....	127
3.2.2 Maksimum Marj Sınıflandırıcıları Optimizasyon Problemi.....	130
3.2.3 Lagrange Çarpanları Yöntemi	134
3.2.4 Karush-Kuhn-Tucker (KKT) Koşulları	137
3.3 Destek Vektör Makineleri	137
3.3.1 Marj Tabanlı Kayıp Fonksiyonları	141
3.3.2 DVM Sınıflandırıcıları.....	143
3.3.2.1 Doğrusal Sert Marjli Sınıflandırıcı.....	143
3.3.2.2 Doğrusal Yumuşak Marjli Sınıflandırıcı.....	147
3.3.2.3 Doğrusal Olmayan DVM Sınıflandırıcıları.....	150
3.3.2.3.1 Kernel Fonksiyonları.....	152
3.3.2.3.2 Mercer Teoremi.....	155
3.3.2.3.3 Kernel Tipleri	156
3.3.2.4 Çoklu ve Dengesiz Sınıfların Sınıflandırılması	158
3.3.3 DVM Regresyonu	159
3.3.4 DVM Optimizasyon Yöntemleri	161
4. UYGULAMA.....	164
4.1 Uygulamanın Amacı ve Kapsamı.....	164
4.2 Hisse Senedi Piyasalarının Tahmin Edilebilirliği	166
4.2.1 Temel Analiz.....	168
4.2.2 Teknik Analiz	170
4.2.3 İstatistiksel Yöntemler	174
4.2.4 Makine Öğrenmesi Yöntemleri	176
4.2.4.1 Yapay Sinir Ağları	176

4.2.4.2 Destek Vektör Makineleri	177
4.3 İMKB Ulusal 100 Endeksinin Yön Tahmini.....	178
4.3.1 İMKB Hakkında Temel Bilgiler	178
4.3.2 İMKB Endeksleri ve Hesaplanmaları	182
4.3.3 Analizde Kullanılan Teknik Göstergeler	186
4.3.3.1 Hareketli Ortalamalar	186
4.3.3.2 MACD (Moving Average Convergence Divergence – Hareketli Ortalamalar Yakınlaşma - Uzaklaşma)	187
4.3.3.3 CCI (Commodity Channel Index - Mal Kanal Endeksi).....	188
4.3.3.4 RSI (Relative Strength Index - Göreceli Güç Endeksi)	188
4.3.3.5 Stokastik.....	188
4.3.3.6 William's %R.....	189
4.3.3.7 Momentum	189
4.3.3.8 MFI (Money Flow Index – Para Akışı Endeksi).....	190
4.3.3.9 ATR (Average True Range - Ortalama Doğru Aralık)	190
4.3.3.10 TRIX	191
4.3.3.11 MASS Endeksi	191
4.3.4 Analizde Kullanılan Borsa Endeks Verileri.....	192
4.3.5 Analizde Kullanılan Makroekonomik Göstergeler	193
4.3.6 Verilerin Toplanması ve Hazırlanması	194
4.3.7 Analizde Kullanılan Yazılım ve Özellikleri	195
4.3.8 Model Parametrelerinin Belirlenmesi	197
4.3.9 Sınıflandırıcıların Performans Değerlendirmesi.....	201
4.3.9.1 Teknik Göstergeler ile Endeks Yönü Tahmini.....	202
4.3.9.2 Borsa Endeks Verileriyle Endeks Yönü Tahmini	204

4.3.9.3 Makroekonomik Göstergeler ile Endeks Yönü Tahmini	207
4.3.9.4 Tüm Verilerin Kullanılmasıyla Endeks Yönü Tahmini	209
4.3.9.5 Öznitelik Seçimi Yaparak Endeks Yönü Tahmini	212
SONUÇ	217
KAYNAKÇA	220
ÖZGEÇMİŞ	237

TABLolar LİSTESİ

Tablo 1. Öğrenme Algorİtmaların Üç Temel Bileşeni	28
Tablo 2. Biyolojik Sinir Modeli İle Yapay Sinir Modeli Arasındaki Benzerlikler ...	50
Tablo 3. Bilgisayar Mimarisi ile Yapay Sinir Ağlarının Karşılaştırılması.....	53
Tablo 4. Yapay Sinir Ağları Uygulamalarına Örnekler	57
Tablo 5. Yapay Sinir Ağlarının Dezavantajları.....	60
Tablo 6. Birleştirme Fonksiyonları	65
Tablo 7. Yaygın Olarak Kullanılan Etkinlik Fonksiyonları	68
Tablo 8. İkili Girdi Sayısına Göre Doğrusal Ayrılabilir Fonksiyonların Oranı	79
Tablo 9. Katman Sayısına Göre Ağların Sınıflandırabildikleri Kümeler.....	87
Tablo 10. Örneklem Büyüklüğü ve Bellek Alanı İlişkisi.....	161
Tablo 11. Borsaların Piyasa Değeri (2011)	179
Tablo 12. Borsaya Kote Olan Şirket Sayısı (2011).....	180
Tablo 13. Hisse Senedi Hacmi (2011).....	181
Tablo 14. Teknik Göstergeler İle İlgili Temel İstatistikler	186
Tablo 15. Analizde Kullanılan Endeksler	192
Tablo 16. Modelde Yer Alan Borsalar İle İlgili Temel İstatistiksel Bilgiler.....	193
Tablo 17. Yıllara Göre İMKB 100 Endeksinin Arttığı ve Azaldığı Gün Sayıları... 194	
Tablo 18. Eğitim ve Test Kümelerinin Artış ve Azalış Yüzdeleri	195
Tablo 19. Hata Matrisi.....	201
Tablo 20. DVM Sınıflandırıcılarının Teknik Gösterge Verileri Üzerindeki Eğitim ve Test Performansları	202
Tablo 21. Teknik Göstergeler Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri.....	202
Tablo 22. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Teknik Göstergeler).....	203
Tablo 23. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Sınıflandırıcıları (Teknik Göstergeler)	203

Tablo 24. YSA Sınıflandırıcılarının Test Kümesi Performansları (Teknik Göstergeler).....	204
Tablo 25. DVM Sınıflandırıcılarının Borsa Endeks Verileri Üzerindeki Eğitim ve Test Performansları	205
Tablo 26. Borsa Endeks Verileri Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri.....	205
Tablo 27. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Borsa Endeks Verileri)	205
Tablo 28. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Borsa Endeks Verileri)	206
Tablo 29. YSA Modellerinin Test Kümesi Performansları (Borsa Endeks Verileri)	206
Tablo 30. Makroekonomik Veriler Üzerinde DVM Sınıflandırıcılarının Eğitim ve Test Performansları	207
Tablo 31. Makroekonomik Veriler Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri.....	207
Tablo 32. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Makroekonomik Göstergeler).....	208
Tablo 33. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Makroekonomik Veriler)	208
Tablo 34. YSA Sınıflandırıcılarının Test Kümesi Performansları (Makroekonomik Veriler)	209
Tablo 35. Tüm Veriler Üzerinde DVM Sınıflandırıcılarının Eğitim ve Test Performansları	210
Tablo 36. En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri (Tüm Veriler Üzerinde)	210
Tablo 37. Test Kümesi üzerinde Dairesel Tabanlı Kernelin Sınıflandırma Özeti (Tüm Veriler Üzerinde).....	210
Tablo 38. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Tüm Veriler)	211
Tablo 39. YSA Sınıflandırıcılarının Tüm Veriler Üzerindeki Test Performansları	211

Tablo 40. Öznitelik Seçiminde Kullanılan Ki-kare ve p Değerleri	213
Tablo 41. Öznitelik Seçimiyle Oluşturulan Küme Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcılarının Özellikleri	214
Tablo 42. Öznitelik Seçimi Yapılmış Veri Kümesi Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri	214
Tablo 43. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Öznitelik Seçimi Yapılmış Küme)	215
Tablo 44. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Öznitelik Seçimi İle Oluşturulan Küme)	215
Tablo 45. Öznitelik Seçimi İle Oluşturulan Veri Kümesi Üzerinde YSA Sınıflandırıcılarının Test Performansları	216

ŞEKİLLER LİSTESİ

Şekil 1. Öğrenme Sisteminin Bileşenleri	16
Şekil 2. Verilen Bir Eğitim Kümesi İçin Bulunan Üç Hipotez.....	20
Şekil 3. Yanlılık - Varyans İkilemi	38
Şekil 4. Biyolojik Sinir Hücresi	48
Şekil 5. Yapay Sinir Hücresi.....	49
Şekil 6. Yapay Sinir Ağı Katmanları	60
Şekil 7. Sigmoid Fonksiyonu	66
Şekil 8. Ağ Mimarileri	70
Şekil 9. İleri Yönelimli Çok Katmanlı Ağ.....	70
Şekil 10. Geri Yönelimli Ağ	71
Şekil 11. Yapay Sinir Ağlarında Denetimli Öğrenme	73
Şekil 12. Basit Algılayıcı (Perceptron)	77
Şekil 13. Kaydırma Teriminin Etkisi	78
Şekil 14. Eğimli İniş Yöntemi	82
Şekil 15. İki Boyutlu Girdi Uzayında Gizli Birimlerin Etkisi	87
Şekil 16. Çok Katmanlı Algılayıcılarda Fonksiyon ve Hata Sinyali.	88
Şekil 17. Küçük Momentum Değerinin Etkisi.....	97
Şekil 18. Büyük Momentum Değerinin Etkisi	98
Şekil 19. Aşırı Öğrenmenin Eğitim ve Geçerleme Hatası Üzerindeki Etkisi.	101
Şekil 20. Ampirik Risk Minimizasyonu (ARM) İlkesinin Tutarlılığı	113
Şekil 21. Doğrusal Sınıflandırıcıların VC Boyutu	117
Şekil 22. Büyüme Fonksiyonunun Davranışı	117
Şekil 23. Genelleme Sınırı	120
Şekil 24. Kapasitelerine Göre Sıralanmış Fonksiyon Kümelerinden Oluşan Yapı .	123
Şekil 25. Döndürme ve Öteleme İşlemleri.....	125
Şekil 26. Geniş Marjlı Ayırıcı Hiperdüzlem.....	126
Şekil 27. Destek Vektörleri.....	126
Şekil 28. Dar Marjlı Ayırıcı Hiperdüzlem	127

Şekil 29. Geniş Marjlı Ayırıcı Hiperdüzlem.....	128
Şekil 30. Silinen Gözlemler	129
Şekil 31. Destek Hiperdüzlemleri	131
Şekil 32. Marjın İzdüşüm Yoluyla Gösterilmesi	132
Şekil 33. Konveks Zarf	136
Şekil 34. İkili Sınıflandırma İçin Marj Tabanlı Kayıp Fonksiyonu.....	141
Şekil 35. DVM Regresyonu İçin Kayıp Fonksiyonu ve Aylak Değişkenler	143
Şekil 36. Yumuşak Marj Sınıflandırıcı	148
Şekil 37. Eşleme Fonksiyonu.....	151
Şekil 38. Regresyon İçin Duyarsız Bölge	159
Şekil 39. Temel Analiz Aşamaları	169
Şekil 40. STATISTICA Otomatik Ağ Arama Modülü.....	197
Şekil 41. Eğitim ve Geçerleme Kümesinin Büyüklüklerinin Belirlenmesi	198
Şekil 42. Ağ Parametrelerinin Belirlenmesi	198
Şekil 43. Ağda Kullanılacak Etkinlik Fonksiyonlarının Belirlenmesi	199
Şekil 44. Ağırlık Sönümlenme Parametrelerinin Belirlenmesi.....	199
Şekil 45. DVM Sınıflandırıcılarında Kullanılan Kernel Türleri	200
Şekil 46. Çapraz Geçerleme ve C Parametresinin Aranması.....	200
Şekil 47. STATISTICA Feature Selection Ekranı.....	212
Şekil 48. Öznitelik Seçimi Sonuçları	213

KISALTMALAR LİSTESİ

ARM	: Ampirik Risk Minimizasyonu
ÇKA	: Çok Katmanlı Ağ
DVM	: Destek Vektör Makineleri
HKT	: Hata Kareleri Toplamı
İMKB	: İstanbul Menkul Kıymetler Borsası
KKT	: Karush-Kuhn-Tucker
PAC	: Probably Approximately Correct
VC	: Vapnik-Chervonenkis
YRM	: Yapısal Risk Minimizasyonu
YSA	: Yapay Sinir Ağları

GİRİŞ

Hisse senedi piyasaları her gün çok sayıda işlemin yapıldığı büyük sistemlerdir. Böyle dinamik, doğrusal olmayan ve hayli karmaşık sistemlerde öngörü yapmak kolay değildir. Piyasalardaki fiyat davranışlarını tam olarak öngörebilen bir yöntem bulunamadığından, alandaki çalışmalar aktif olarak devam etmektedir. Klasik istatistiksel yöntemler bu konuda kısmen başarılı olabilmişlerdir. Son yıllarda ise örneklerden öğrenme yapabilen, genelleme yeteneğine sahip ve istatistiksel yöntemlere göre daha az varsayımları olan makine öğrenmesi yöntemlerinin kullanımı artmıştır. Tezde incelenen Yapay Sinir Ağları (YSA) ve Destek Vektör Makineleri (DVM) bu alanda yaygın olarak kullanılan makine öğrenmesi yöntemleridir.

YSA, biyolojik sinir ağlarındaki öğrenme sürecinden esinlenerek oluşturulmuş matematiksel modellerdir. Birbirine bağlı basit hesaplama birimlerinden oluşan yapay sinir ağlarında bilginin işlenmesi bu birimlerin arasındaki bağlantılar kullanılarak yapılır. YSA verideki örüntüleri keşfedebilen, gürültülü verilerle çalışabilen, doğrusal olmayan, esnek modellerdir. Bu özellikler, yöntemin borsa gibi dinamik sistemlerin modellenmesinde kullanılabilmesini sağlamaktadır. Yeterli sayıda doğrusal olmayan birime sahip bir yapay sinir ağının örneklerden öğrenme yoluyla herhangi karmaşık bir fonksiyonel ilişkiyi yüksek bir doğrulukla kestirebileceği teorik olarak ispatlanmıştır. Ampirik olarak da yöntemin sınıflandırma, regresyon, kümeleme gibi problem tiplerine uygulandığı ve başarılı sonuçların elde edildiği birçok çalışma mevcuttur.

DVM, İstatistiksel Öğrenme Teorisi'ne dayanan, sınıflandırma, regresyon problemlerine uygulanabilen ve gittikçe yaygınlaşan bir makine öğrenmesi yöntemidir. DVM, sınıfların doğrusal ayrılabilir olduğu durumdaki ikili sınıflandırma probleminde sonsuz sayıdaki doğrusal karar sınırından genelleme hatasını en aza indireni seçer. Optimal karar sınırını bulmak için oluşturulan optimizasyon problemi standart kuadratik programlama teknikleri ile

çözölebilmektedir. Sınıfların doğrusal olarak ayrılamadığı durumda ise optimizasyon probleminde çeşitli deęişiklikler yapılarak sonuca ulaşılmaktadır. DVM doğrusal olmayan karar sınırları oluşturmak için de kullanılabilir. Bunun için kernel fonksiyonları yoluyla girdiler daha yüksek boyutlu bir uzaya eşlenir ve doğrusal sınıflandırma problemi bu yüksek boyutlu uzayda çözölür. DVM; model karmaşıklığını ayarlayabilme, genelleme yeteneęi, kernel fonksiyonlarını kullanabilme gibi özellikleri dolayısıyla borsa öngörü uygulamalarında sıklıkla tercih edilen bir yöntem olmuştur.

YSA ve DVM yöntemlerinin borsa yönünün öngörölmesinde uygulanabilirliklerinin analiz edildięi bu tez çalışmasında dört bölüm bulunmaktadır.

İlk bölümde makine öğrenmesi ile ilgili temel kavramlara yer verilmiştir. Makine öğrenmesi yapay zekâ ana başlığı altında incelendięinden yapay zekânın tanımı, tarihçesi ve temel yapay zekâ yaklaşımlarından bahsedilmiştir. Makine öğrenmesinin tanımı, amacı, ihtiyaç duyulduęu durumlar incelenmiş ve öğrenme probleminin matematiksel bir çerçevesi oluşturulmuştur. Öğrenme yöntemleri ve öğrenme algoritmalarının bileşenleri ele alınmış ve makine öğrenmesi yöntemlerinde çok önemli yeri olan model seçimi, model geçerleme, model karmaşıklığını düzenleme konuları üzerinde durulmuştur.

YSA yöntemi ikinci bölümde incelenmiştir. Bu bölümde YSA'nın biyolojik sinir aęları ile benzerlikleri ve farklılıkları özetlemiştir. Yöntemin tarihsel gelişimi, kullanım alanları, güçlü ve zayıf yönleri ele alınmıştır. YSA mimarileri ve öğrenme kurallarına bölümde yer verilmiştir. Basit algılayıcı ve delta öğrenme kurallarından sonra çok katmanlı aęlarda geri yayılım algoritmasının çalışması incelenmiştir. Bölümün sonunda başarılı YSA modellerinin oluşturulabilmesi için dikkat edilmesi gereken hususlar ve takip edilmesi gereken adımlar ele alınmıştır.

Üçüncü bölümde DVM yönteminin teorisi bulunmaktadır. Bölümün başlangıcında DVM'nin teorik altyapısını oluşturan İstatistiksel Öğrenme Teorisi hakkında bilgiler yer almaktadır. Teori kapsamında Ampirik Risk Minimizasyonu, Vapnik–Chervonenkis boyutu, genelleme sınırları, Yapısal Risk Minimizasyonu gibi

önemli kavramlar açıklanmıştır. Daha sonra DVM yönteminin altyapısını oluşturan marj, ayırıcı hiperdüzlem, optimal ayırıcı hiperdüzlem, marj tabanlı kayıp fonksiyonları kavramları ve marj–karmaşıklık ilişkisi ele alınmıştır. Doğrusal sert marj ve doğrusal yumuşak marj sınıflandırıcılarının ardından doğrusal olmayan DVM sınıflandırıcıları açıklanmıştır. Ardından kernel fonksiyonlarının kullanım amacı, kernel tipleri ve DVM ile birlikte kullanımları yer almaktadır. Bölümün son kısmı DVM regresyonuna ve DVM optimizasyon yöntemlerine ayrılmıştır.

Dördüncü bölümde tezin uygulama alanı olan İstanbul Menkul Kıymetler Borsası hakkındaki temel bilgiler ve çeşitli istatistiksel veriler yer almaktadır. Hisse senedi piyasalarında öngörü yapmak için kullanılan yaklaşımlar ve değişkenlerle ilgili literatür taramaları sunulmuştur. Analizde kullanılan veri kümelerinde yer alan değişkenlerle ilgili açıklamalar ve bu veri kümeleri ile ilgili çeşitli istatistiksel bilgiler verilmiştir. Analizlerde oluşturulan sınıflandırıcıların performans karşılaştırmaları tablolar ile gösterilmiştir. Elde edilen sonuçlara göre İstanbul Menkul Kıymetler Borsası Ulusal 100 Endeksi'nin yön tahmininde hangi veri kümesinin daha etkili olduğu, YSA ve DVM yöntemlerinden hangisinin daha başarılı sonuçlar verdiği bulunmuştur.

BİRİNCİ BÖLÜM

1. MAKİNE ÖĞRENMESİNE GENEL BİR BAKIŞ

Makine öğrenmesi konusu yapay zekâ alanı altında incelendiğinden bu bölümde öncelikle yapay zekâ yaklaşımları, yapay zekâ'nın tarihsel gelişimi konularına yer verilmiştir. Daha sonra makine öğrenmesi ile ilgili temel kavramlar açıklanarak öğrenme problemi matematiksel bir bakış açısıyla incelenmiştir.

1.1 Yapay Zekâ

Günlük hayatta akıl ve zekâ kavramları karıştırılabilmektedir. Akıl, genetik yoldan gelen sevgi, korku, kıskançlık vb. doğal güdülerle birlikte çevresel ve sosyal şartlardan etkilenmektedir. Aklın herhangi bir yazılım tarafından taklit edilmesi mümkün değildir.

Anlama, öğrenme, analiz etme, ilk defa karşılaşılan veya ani olarak gelişen bir olaya uyum sağlayabilme zekâ ile gerçekleştirilir. Zekâ yazılımlarla taklit edilebilmektedir. Bu durumda zekâ, “*yapay zekâ*” olarak adlandırılır.

Yapay zekâ biliminin asıl örnek aldığı kaynak, insan zekâsıdır. Ancak; insan zekâsı bütünüyle keşfedilmiş ve çalışma sistemi anlaşılabilir değildir. Yapay zekâ konusundaki araştırmalarının önemli amaçlarından birisi de insan zekâsının çalışmasını anlamaktır. Böylece insan zekâsını taklit edebilecek sistemler geliştirilebilmek mümkün olacaktır.

Yapay zekâ'nın temel amacı, zekâ belirtisi sayılabilecek işleri yapabilecek bilgisayar sistemlerinin tasarımıdır. Yapay zekâ, bilgi edinme, algılama, görme, düşünme ve karar verme gibi yeteneklerle donatılmış bilgisayar sistemleri olarak da düşünülebilir (Dağ, 2006: 936).

1.1.1 Yapay Zekâ Yaklaşımları

Yapay zekâ konusunda yapılabilecek her tanım aşağıda belirtilen dört temel kategoriden birisine uyar (Russel & Norvig, 2009: 2-5).

- İnsan gibi düşünen sistemler
- İnsan gibi davranan sistemler
- Rasyonel düşünen sistemler
- Rasyonel davranan sistemler

Yapılan tüm yapay zekâ tanımlarının dayanak noktasını oluşturan bu tanım grupları aşağıda kısaca özetlenmiştir.

1.1.1.1 İnsan Gibi Düşünme (Bilişsel Modelleme) Yaklaşımı

İnsan gibi düşünen sistemlerden bahsedebilmek için öncelikle insanların nasıl düşündüğü hakkında net bir teorinin olması gerekir. Böyle bir teori olsaydı bunun bir bilgisayar programına çevrilmesi kolay olurdu. Ancak, insan düşüncesini tam anlamıyla açıklayan bir teori bulunmadığından, insan gibi düşünen sistemler şimdilik mevcut değildir. Eğer bir programın girdisi, çıktısı ve zamanlaması insan davranışına uyuyorsa bu programın insan gibi düşündüğü söylenebilir.

1.1.1.2 İnsan Gibi Davranma (Turing Testi) Yaklaşımı

Bu yaklaşım, 1950 yılında İngiliz matematikçi Alan M. Turing tarafından ortaya atılmıştır. Turing'in yaklaşımında zeki davranış, *“tüm bilişsel işleri bir müzakereciyi kandırabilecek ölçüde insana benzer şekilde yerine getirebilmek”* olarak tanımlanmıştır. Bunu ölçmek amacıyla Turing tarafından *“Turing Testi”* geliştirilmiştir.

Turing testinde bir hat üzerindeki müzakereci, hattın karşısında bulunan insana ve yapay zekâ yazılımı çalıştıran bilgisayara hangisinin insan hangisinin bilgisayar olduğunu anlayabilmek için çeşitli sorular sorar. Test sonunda müzakereci kimin bilgisayar kimin insan olduğunu ayırt edemez ise yapay zekâ yazılımı testi geçmiş sayılır ve zeki olduğu kabul edilir. Turing testini geçebilecek bir bilgisayarın doğal dil işleme (natural language processing) yeteneğine sahip olması şarttır. Buna

ek olarak anlık muhakeme gücüne, öğrenme yeteneğine ve bilgi depolama özelliğine sahip olması gerekir. Şu anda Turing Testini geçen herhangi bir yazılım bulunmamaktadır (Negnevitsky, 2004: 4). Hatta birçok yapay zekâ yazılımı bu teste gerek duyulmayacak ölçüde sınırlı ve basit düzeydedir.

1.1.1.3 Rasyonel Düşünme (Düşünce Kanunları) Yaklaşımı

Rasyonel düşünmede amaç doğru düşündürmektir. Doğru düşünecek bir yazılım geliştirebilmek için düşünme tarzının kodlanması gerekmektedir. Bunu ilk deneyen kişi Aristo'dur.

Aristo mantığına göre “*Sokrates insandır. İnsanlar ölümlüdür. Öyleyse Sokrates ölümlüdür*”. Bu şekildeki düşünme kurallarının düşünme işlemini yönettiği kabul edilmiştir. Mantık, doğru düşünme için gerekli kurallardan oluşur. Mantık bilimindeki gelişmeler sayesinde her türlü ilişki kesin ve net bir notasyon kullanılarak gösterilebilir hale gelmiştir. Örneğin, 1960'lı yıllarda mantıksal notasyonlar ile ifade edilen problemlere çözüm üretebilecek yapay zekâ yazılımları geliştirilmiştir. Böylece mantıksal akım başlamıştır. Bu akımı savunan bilim insanları mantık kurallarını kullanarak zeki sistemler oluşturmaya çalışırlar.

1.1.1.4 Rasyonel Davranma (Zeki Ajan) Yaklaşımı

Rasyonel davranmaktan kasıt amaca ulaşabilecek şekilde davranmaktır. Bu şekilde davranış sergileme yapay zekâ'nın çalışma alanlarından olan “*zeki ajanlar*” sayesinde gerçekleştirilir. Bir ajan, davranış sergileyen şeydir. Bir ajanı sıradan bir bilgisayar yazılımından ayıran farklar vardır. Örneğin, bir ajan çevreden bilgi alır, otonomdur, çevreye uyum sağlar ve bir başka ajanın görevini üstlenebilir.

Zeki ajan yaklaşımında, bir olayda doğru çıkarımda bulunmak ve elde edilen sonuca göre belirlenen şekilde davranmak amaçlanmaktadır. Bunun için zeki ajan oluşturmada rasyonel düşünme tanımından yola çıkılır. Yapay zekâ'ya mantıksal akımın gözüyle bakıldığında en önemli konu çıkarımların doğru olmasıdır. Doğru çıkarım yapmak zeki bir ajan olmanın önemli bir parçasıdır.

Yapay zekâ kavramının (Russel & Norvig, 2009) tarafından yapılan sınıflandırmaya esas teşkil eden tanımlamalarından bazıları şunlardır (Yıldız, 2009: 23):

- Zeki davranışları matematiksel süreçler halinde açıklama ve taklit etme alanındaki çalışmalar.
- Şu anda insanların bilgisayarlardan daha iyi oldukları alanlarda yaptıklarını, bilgisayarlara yaptırma çalışmaları.
- İnsanlar tarafından yapılırken zekâ kullanmayı gerektiren işlevleri yapacak bilgisayarlara yaratma sanatı.
- Gözlemcilerin zekâyı fark edebilecekleri şekilde davranan sistemler konusundaki çalışmalar.

1.1.2 Yapay Zekâ'nın Tarihsel Gelişimi

Yapay zekâ konusunda 20.yy ortalarına kadar kayda değer bir gelişme görülmemiştir. Aşağıda, bu alanda yapılan önemli çalışmalar tarih sırasına göre belirtilmiştir (Negnevitsky, 2004: 4-17).

1943 yılında Warren McCulloch ve Walter Pitts biyolojik sinir hücrelerini modellemişlerdir. Yapay sinir hücrelerinden oluşan bir ağın, her türlü fonksiyonu hesaplayabileceğini ve böyle bir ağın öğrenebileceğini savunmuşlardır. İlk genel amaçlı bilgisayar olan ENIAC'ı tasarlayan John Von Neuman, McCulloch ve Pitts'in konuyla ilgili 1943 yılında yayınladıkları makaleden esinlenmiştir. McCulloch ve Pitts'in çalışmaları yapay zekâ alanında "*bağlantıcılar (connectionists)*" akımını başlatmıştır.

Claude Elwood Shannon, 1950 yılında satranç oynayan bir bilgisayar programı üzerine bir makale yazmış bu makalede sezgisel yaklaşımın gereğini ortaya koymuştur.

Alan M. Turing tarafından 1950’de Turing Testi geliştirildi.

1956 yılında Dartmouth Koleji’nde düzenlenen bir toplantıda “yapay zekâ” kavramı ilk defa John McCarthy tarafından kullanılmıştır.

1956 yılında Allen Newell, Herbert Simon ve J.C. Shaw insanın problem çözme yeteneklerini taklit ederek bazı bilimsel ispatlar yapabilen “*Mantık Kuramcısı (Logic Theorist)*” adlı ilk yapay zekâ yazılımını sundular. Daha sonra Newell ve Simon, “*insan gibi düşünme*” yaklaşımına göre üretilmiş ilk program olan “*Genel Sorun Çözücü (General Problem Solver)*” programını geliştirmişlerdir.

John McCarthy 1958 yılında Massachusetts Institute of Technology çatısı altında bir yapay zekâ programlama dili olan *LISP* programlama dilini geliştirdi.

1960’lı ve 1970’li yıllar yapay zekâ konusundaki projelerin aşırı iyimser ve çok iddialı olmaları nedeniyle çalışmaların birçoğunun boşa gittiği bir dönem olmuştur. Geliştirilen yazılımların başarılarını göstermek için bir takım yapay problemler yaratılmıştır. Daha sonraları bu problemler, gerçek yaşamı hiçbir şekilde temsil etmeyen oyuncak problemler olarak suçlanmıştır. Yapay zekâ’nın yalnızca bu oyuncak problemlerde başarılı olabileceği ve gerçek yaşama ölçeklenemeyeceği ileri sürülmüştür.

1965 yılında Lotfi A. Zadeh yapay zekâ konusunda büyük bir adım olarak kabul edilen “*Bulanık Mantık (Fuzzy Logic)*” konusundaki ünlü makalesini yayınladı.

1965 yılında optimizasyon, öğrenme gibi problemlere biyolojik popülasyon genetiği kurallarına dayanarak yaklaşan “*Evrimsel Hesaplama (Evolutionary Computation)*” adlı yapay zeka teknolojisi Ingo Rechenberg ve Hans-Paul Schwefel tarafından geliştirilmiştir.

John Henry Holland’ın 1975 yılında doğal seleksiyon ve genetik konularındaki bilgileri problem çözme amacıyla bilgisayar ortamına aktarmasıyla başka bir yapay zekâ teknolojisi olan “*Genetik Algoritmalar (Genetic Algorithms)*” ortaya çıktı.

Evrimsel yaklaşımının bir başka uygulama biçimi olan “*Genetik Programlama (Genetic Proqraming)*” John Koza tarafından 1980’li yıllarda olgunlaştırılmıştır.

Bütün bu çalışmaların sonunda yapay zekâ araştırmacıları temel olarak iki gruba ayrıldılar. Bir grup araştırmacı insan gibi düşünen sistemler oluşturmak için çalışırken, diğer gruptakiler rasyonel düşünebilen sistemler oluşturmayı amaçlamaktadırlar.

1.2 Makine Öğrenmesi

İnsan zekâsının en güçlü yönlerinden birisi çevreye uyumdur. Çevredeki yeni koşullara uyum sağlama ve buna göre davranma, öğrenme yoluyla gerçekleştirilir. Öğrenme yeteneği insanoğlunu bilgisayarlara karşı üstün kılan özelliklerden birisidir. Bu nedenle yapay zekâ’nın en önemli alt alanlarından birisi de makine öğrenmesidir (Ertel, 2011: 3).

1.2.1 Makine Öğrenmesi’nin Önemi

İnsanın öğrenme yeteneğine yakın bir şekilde öğrenebilen yazılımlar henüz geliştirilememiştir. Fakat belli türdeki öğrenme görevleri için etkin olarak kullanılabilen yöntemler geliştirilmiştir. Öğrenmenin teorik bir altyapıya oturması sağlanmaya çalışılmıştır. Birçok öğrenme probleminde faydalı olan ticari yazılımlar geliştirilmektedir.

Depolama ve veri iletişim maliyetlerinin azalması çok büyük veri kümelerinin saklanabilmesine ve paylaşılabilmesine imkân sağlamıştır. Hesaplama hızındaki gelişmelere paralel olarak öğrenme algoritmalarını çok büyük veriler üzerinde uygulamak mümkün hale gelmiştir. Kredi başvuru bilgileri, finansal hareketler, tıbbi kayıtlar gibi bilgilerin tutulduğu çok büyük veri kümelerinden yararlı bilgileri çıkarmak için yaygın olarak makine öğrenmesi yöntemleri kullanılmaktadır (Mitchell, 1997: 1).

Daha önceleri, yapay zekâ’nın bir gün gerçek olabilmesi için yeni bir bakış açısına, yepyeni bir hesaplama modeline, ya da tamamen farklı yöntemlere gerek

olduđuna inanılırdı. Son yıllarda makine öğrenmesi yöntemlerinin çok çeşitli uygulamalarda göstermiş oldukları başarılarla bakıldığında belki de gerekenin bunlar değil, büyük miktarda örnek veri ve yeterli hesaplama gücü olduđu söylenebilir.

Örneđin, makine öğrenmesinin “dođal dil işleme” alanındaki uygulamaları sürekli artmaktadır. Bu alanın en etkileyici uygulamalarından birisi bilgisayarla çeviridir. Bunun için, daha önceleri elle tasarlanmış çeviri kuralları kullanılmıştır. Makine öğrenmesi algoritmalarını kullanan yazılımların gösterdikleri başarılar dikkate alındığında, bir yazılımın daha önceden oluşturulmuş kuralları kullanmak yerine çevrilmiş çok sayıdaki belgeyi inceleyip bir dildeki cümleleri başka bir dildeki cümlelere çeviren kuralları kendisinin örneklerden çıkarması en ümit verici yol olarak görünmektedir (Alpaydın, 2011: 12).

1.2.2 Makine Öğrenmesi'nin Amacı

Makine öğrenmesi'nin temeli *"Gözlemler sonucu elde edilen verileri bilgisayar yardımıyla işleyerek, bu gözlemlerin içinde yer alan örüntüler (patterns) keşfedilip tanımlanabilir mi?"* sorusuna dayanmaktadır (Hamel, 2009: 4).

İnsanlar çevrelerindeki farklı nesnelere farklı sınıflara ait olacak şekilde sınıflandırır. Örneđin, bir güvercin gördüklerinde kuş sınıfında, bir yunus gördüklerinde balık sınıfında yer alacak şekilde bir etiketleme kullanırlar. Makine öğrenmesindeki amaç da bunun gibi bir etiketleme sürecine en yakın modeli oluşturmaktır. Yani, bilgisayarın eldeki gözlemler ve bu gözlemlere karşılık gelen dođru etiketlerin yer aldığı veri kümesini işleyerek, görmediđi gözlemler için de dođru etiketleme yapabilecek süreci ortaya çıkarması amaçlanır.

Gözlenen verileri açıklayan bir süreç olduđuna inanılır. Bir sürecin, tüm ayrıntıları ile bilinmesi de rastgele olmadıđı bilinir (Bramer, 2007: 203). Sözelimi, insanlar alışveriş sepetlerini tamamen rastgele doldurmazlar. Bir ürünü aldıklarında genellikle bu ürünle ilgili başka ürünleri de alırlar. Dolayısıyla veride bir düzen olduđu varsayılır.

Verilerin altındaki süreç net olarak tanımlanamasa da faydalı olacak yaklaşık bir açıklaması oluşturulabilir. Bu yaklaşık açıklama verinin önemli bir bölümü için

yeterli olabilir. Süreci tam olarak tanımlamak mümkün olmasa da verideki önemli örüntülerin bulunabileceği düşünülür. Makine öğrenmesi yöntemleri kullanılarak bulunan modeller, sürecin anlaşılmasına yardım ettiği gibi öngörü yapmaya da imkân sağlar. En azından yakın geleceğin, verinin toplandığı geçmişten çok farklı olmadığı varsayılarak bu veriden oluşturulmuş kurallarla yapılacak öngörülerin de doğru olması beklenir. Bu, makine öğrenmesi'nin en temel varsayımlarından birisidir.

Bir yazılımına bir görev, bir performans ölçütü ve deneyimler girdi olarak verildiğinde, eğer performans deneyimler arttıkça iyileşiyorsa bu yazılımının öğrendiği söylenebilir (Mitchell, 1997: 2).

Örneğin, bir el yazısı tanıma probleminde görev; el yazısı ile yazılmış harfleri tanıma ve sınıflandırma, performans ölçütü; doğru sınıflandırılan harflerin yüzdesi, eğitim deneyimi; el yazısıyla yazılan harflerin ve bu harflerin sınıflarının yer aldığı veri kümesidir. Bir robotun araç sürmeyi öğrenmesi probleminde ise görev; görüntü algılayıcılarını kullanarak yolda araç sürme, performans ölçütü; bir hata oluşana kadar ortalama gidilen mesafe, eğitim deneyimi; gerçek bir sürücünün aracı sürerken kaydedilmiş direksiyon hareketleri ve yol görüntüleridir. Genel olarak öğrenmenin gerçekleşebilmesi için amaçlanan görev, geliştirilmesi gereken performans ölçütü ve deneyimin kaynağının belirlenmesi gerekir.

Deneyim, toplanan geçmiş verilere karşılık gelir. Veriler bir denetimci tarafından etiketlenmiş olarak veya çevre ile etkileşim yoluyla elde edilmiş farklı bir biçimde bulunabilir. Her iki durumda da verinin kalitesi, oluşturulacak modellerin başarısı için büyük öneme sahiptir (Mohri vd., 2012: 1).

Makine öğrenmesi yöntemlerinin uygulandığı projelerinin başarısını etkileyen en önemli faktör kullanılan özelliklerdir (attributes). Her birisi öğrenilmek istenen kavramla iyi korelasyona sahip birçok bağımsız özellik varsa öğrenme kolayca yapılabilir. Diğer taraftan, sınıf etiketleri özelliklerin çok karmaşık bir fonksiyonu ise öğrenme zorlaşır. Sıklıkla, ham veriler öğrenmeyi sağlayacak biçimde değildir. Fakat bu ham verilerden öğrenmeye daha uygun “öznitelikler (features)” oluşturulabilir.

Özniteliklerin belirlenmesi veya oluşturulması süreçleri makine öğrenmesi projelerinde zamanın çoğunun harcandığı işlemlerdir (Domingos, 2012: 82).

1.2.3 Makine Öğrenmesi'ne İhtiyaç Duyulan Durumlar

Makine öğrenmesi, *“bilgisayarların öğrenebilecek şekilde programlanabilmelerini sağlayacak yöntemler konusunda yapılan çalışmalardır”*. Bilgisayarlar farklı alanlardaki birçok görevde kullanılmaktadır. Bu görevlerin bir kısmı için gerekli yazılımların tasarlanması ve uygulanması programcılar için kolaydır. Fakat doğrudan programlama yapmanın çok zor veya imkânsız olduğu durumlar da bulunmaktadır. Temel olarak bu durumlar dört genel kategori altında toplanabilir.

1.2.3.1 Alan Uzmanlığının Olmadığı Durumlar

Algılayıcı bilgilerini inceleyerek arızaları oluşmadan öngörmek istenebilir. Fakat bazı durumlarda (örneğin, bir otomatik üretim tesisinde makineler yeni olduğunda) oluşturulacak yazılımda ihtiyaç duyulacak bilgileri verebilecek bir alan uzmanı bulunmaz. Bir makine öğrenmesi sistemi, kayıtlı verileri ve sonraki arızaları inceleyerek öngörü için kullanılacak kuralları öğrenebilir (Dietterich, 2003: 1).

1.2.3.2 Uzmanlığın Açıklanamadığı Durumlar

Konuşma tanıma, el yazısı tanıma ve doğal dil işleme gibi algılamaya dayalı değişik görevlerde insanlar uzman seviyesinde yeteneklere sahip olmalarına rağmen bu görevleri yerine getirirken hangi adımları takip ettiklerini detaylı olarak tanımlayamazlar. Fakat makine öğrenme algoritmaları bu görevlerde kullanılan girdiler ve girdilere karşılık gelen doğru çıktılarının yer aldığı örnekleri inceleyerek girdiler ile çıktıları eşleştirmeyi öğrenebilirler (Blum, 2012: 1).

Gerçek hayatta matematiksel bir modeli kurulamadığı için klasik bilgisayar programlama teknikleriyle çözülemeyen birçok problem olduğundan, güvenilir öğrenme sistemlerinin tasarlanması büyük öneme sahiptir. Örneğin, hali hazırda bolca örneği olmasına rağmen, el yazısı tanıma işini yerine getirebilen bir bilgisayar programının nasıl yazılacağı bilinmemektedir. İnsanların okumayı öğrendiği şekilde bilgisayarların herhangi bir harfi diğer harflerden ayırabilecek şekilde eğitilip

eğitilemeyeceği önemli bir sorudur. Aynı tür sorular DNA dizilimindeki genlerin bulunması, istenmeyen e-postaların filtrelenmesi veya nesnelerin tanınması konularında da sorulabilir. Bu problemlerin her birinin çözümü, günlük hayatın bazı yönlerini büyük ölçüde değiştirebilme potansiyeline sahiptir. Makine öğrenmesi algoritmaları bu tür problemlerin çözümüne önemli katkılar sunmaktadır (Cristianini & Shawe-Taylor, 2000: 1).

1.2.3.3 Problemin Hızlı Değiştiği Durumlar

Bazen problemler hızlı bir şekilde değişebilmektedir. Örneğin finansa, borsanın gelecekteki davranışı, tüketicilerin satın alma durumları veya kur oranları öngörülme istenmektedir. Fakat örnek verilen durumlar sıklıkla değiştiğinden, iyi öngörü yapabilen bir bilgisayar yazılımı oluşturulsa bile bu yazılımın sık sık yeniden yazılması gerekecektir. Diğer yandan, bir öğrenme sistemi kendini yeni duruma uyarlayarak programcıyı öğrenilmiş öngörü kuralları kümesini devamlı olarak değiştirmekten ve ayarlama yükünden kurtarabilir (Dietterich, 2003: 1).

1.2.3.4 Özelleştirme Gerektiren Durumlar

Her bir kullanıcı için ayrı ayrı özelleştirilmesi gereken uygulamalar olabilmektedir. İstenmeyen e-postaları filtreleyen bir bilgisayar programı ele alındığında her bir kullanıcının farklı filtrelere ihtiyaç duyacağı açıktır. Kullanıcıların kendi kurallarını yazmasını beklemek veya kurallarını güncel tutabilmeleri için her bir kullanıcıya yazılımcı tahsis etmek uygulanabilir bir yaklaşım değildir. Bir makine öğrenme sistemi ise kullanıcının hangi tür mesajları reddettiğini öğrenerek otomatik olarak kullanıcıya özgü filtreleme kuralları oluşturur (Flach, 2012: 2).

1.2.4 Makine Öğrenmesi ve Veri Madenciliği

“*Bilgi keşfi*”, elle incelenmesi çok zor olan büyük veri kümelerinden faydalı olabilecek bilgiyi ortaya çıkarma amacıyla yürütülen yarı otomatik bir süreçtir. Sürecin yarı otomatik olarak belirtilmesinin sebebi, bilgi keşfi sırasında bilgisayarların kullanılmasına karşılık bir uzmanın rehberliğinin de kesinlikle gerekli olmasından dolayıdır. Bilgi keşfi sonucunda elde edilen bilgiler “*model*” olarak adlandırılan açıklayıcı örüntüler şeklini almaktadır. Tüm modellerde istenen özellik,

modellerin gereksiz detayları atıp veri içindeki ana eğilimleri özetleyebilmesidir. Bir model çok büyük miktardaki veriyi temsil edebilir. Böylece, verinin içinde gizli olarak yer alan bilgiye erişimi sağlar (Hamel, 2009: 3).

“*Veri madenciliği*” sıklıkla bilgi keşfi süreci ile ilişkilendirilmektedir. Veri madenciliği veritabanlarındaki verilerden bilgi çıkarma amaçlı özel bir bilgi keşif süreci olduğundan bazen “*veritabanlarından bilgi keşfi*” olarak da adlandırılır (Nisbet, 2009: 18). Bilgi keşfi süreci, alan uzmanlığı, veri hazırlama, veri görselleştirme, model değerlendirme ve uygulama gibi birçok adım içermekle birlikte sürecin merkezinde makine öğrenmesi algoritmaları yer alır.

Veriden kural öğrenme, “*bilgi çıkarımı*” sağlar. Kural, veriyi açıklayan basit bir modeldir. Oluşturulan model incelenerek verinin altında yatan süreç hakkında bir açıklama elde edilebilir. Örneğin, kredi başvurusu yapan müşterileri düşük ve yüksek riskli müşteriler şeklinde ayıracak bir sınıflandırıcı oluşturulduğunda, düşük riskli müşterileri yüksek riskli müşterilerden ayıran özellikler hakkında bilgi sahibi olunur. Bu bilgi düşük riskli müşterileri hedefleyen kampanyalar tasarlamak için kullanılabilir.

Veriden kural çıkarımı ile veriyi özetleyen bir model elde edildiğinden öğrenme sürecinin bir tür sıkıştırma yaptığı söylenebilir (Kovalerchuk & Vityaev, 2000: 18). Modeli saklamak ve işlemek tüm veriyi saklamak ve işlemeye göre daha ucuzdur. Bu durum, toplamının bir kez öğrenildikten sonra olası tüm sayı çiftlerinin toplamalarını akılda tutmaya gerek kalmamasına benzetilebilir.

1.2.5 Makine Öğrenmesinin Diğer Bilimlerle İlişkisi

Öğrenme algoritmalarının başarısı kullanılan veriye bağlı olduğundan makine öğrenmesi doğal olarak istatistikle ilişkilidir. Genel olarak, makine öğrenmesi yöntemleri bilgisayar bilimlerindeki temel kavramları, istatistik, olasılık ve optimizasyon alanlarındaki fikirlerle birleştiren veri güdümlü yöntemlerdir (Mohri vd., 2012: 1).

Makine öğrenmesi model oluşturmada istatistik teorisinden faydalanır. Çünkü temeli, gözlenmiş bir örneklemden çıkarım yapmaktır. Bilgisayar bilimlerinin ise

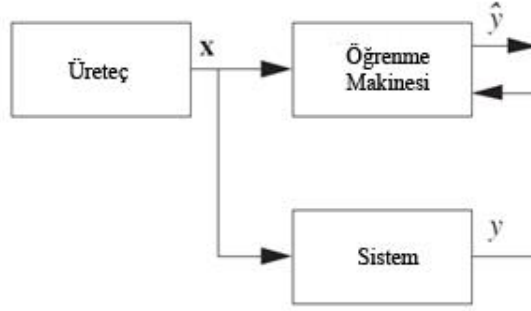
temelde iki rolü vardır. İlki, optimizasyon probleminin çözümü için etkin algoritmalara ihtiyaç duyulmasıdır. İkincisi, bir modelin bellekteki gösterimi ve bu modeli kullanarak çıkarım yapma işlemi de etkin olmalıdır. Bazı uygulamalarda öğrenme algoritmasının etkinliği yani bellek ve zaman karmaşıklığı öngörü başarısı kadar önemli olabilmektedir (Alpaydın, 2011: 3).

Öğrenme konusuyla ilgilenen psikoloji, bilişsel bilim, ya da nöroloji dallarının aksine makine öğrenmesinde esas amaç, insan ve hayvanların öğrenmelerinin altında yatan süreçleri anlamak değil, öğrenme süreci ile ilgili bu alanlardan elde edilen bilgileri kullanarak işe yarar yeni araçlar geliştirmektir.

Örneğin, ses tanıma probleminde ses verisi hakkında her bir bilim farklı sorular sorar. Makine öğrenmesi yaklaşımı sonuçları doğru veren ve verimli bir ses tanıma sisteminin oluşturulmasına odaklanırken, istatistik, psikoloji bilimi ile işbirliği yaparak ses tanıma sürecinin altındaki mekanizmalar hakkındaki hipotezleri test etmeyi amaçlar. Veri madenciliği yaklaşımı ise konuşmacıları yaş, cinsiyet veya eğitim seviyesine göre gruplandırmak için konuşma verisindeki örüntüleri arar (Dietterich, 2003: 2).

1.2.6 Öğrenme Probleminin Genel Çerçevesi

Genel olarak öğrenme senaryosu 3 temel bileşenden oluşur. Bu bileşenler; rassal girdi vektörlerini oluşturan üreteç, verilen bir girdi vektörü için çıktıyı oluşturan sistem ve sistemin kullandığı bilinmeyen bir girdi-çıkıtı ilişkisini gözlemlerden tahmin etmeye çalışan öğrenme makinesidir (Şekil 1). Tanımlanan bu çerçeve çok geneldir ve birçok pratik öğrenme problemini tanımlar (Cherkassky & Mulier, 2007: 21).



Şekil 1. Öğrenme Sisteminin Bileşenleri

1.2.6.1 Üreteç

Üreteç (örnekleme dağılımı) bilinmeyen sabit bir $P(x)$ olasılık dağılımından birbirinden bağımsız, d boyutlu $x \in \mathcal{R}^d$ rassal vektörlerini oluşturur. Sisteme sunulan veriler gözlemsel olduğu için modelleyici (öğrenme makinesi) sisteme gelen girdiler üzerinde herhangi bir kontrole sahip değildir.

1.2.6.2 Sistem

Sistem, bilinmeyen bir $P(y|x)$ koşullu olasılığına göre her bir x girdi vektörü için bir y çıktısı oluşturur. Bu tanım, spesifik bir durum olarak $y = t(x)$ deterministik sistemini ve $y = t(x) + \xi$ (ξ : sıfır ortalamalı gürültü) regresyon gösterimini de içermektedir. Gerçek sistemlerin tam rassal çıktılara sahip olması nadirdir. Sıklıkla sistemde olması gereken tüm değişkenler gözlenemediklerinden sistemde yer alamazlar. İstatistiksel olarak gözlenemeyen bu girdilerin sistem çıktıları üzerindeki etkisi rassal olarak karakterize edilir ve bir olasılık dağılımı ile temsil edilir (Kantardzic, 2011: 82).

1.2.6.3 Öğrenme Makinesi

En genel şekilde ele alınacak olursa öğrenme makinesi bir $f(x, \omega)$, $\omega \in \Omega$ (Ω , fonksiyonlar kümesini indeksleme amacıyla kullanılan parametreleri ifade eder) fonksiyonlar kümesini uygulayabilme kapasitesine sahiptir. Bu gösterimde, öğrenme makinesi tarafından uygulanacak fonksiyonlar kümesi öğrenme süreci başlamadan önce önsel (prior) olarak seçilen herhangi bir fonksiyonlar kümesi olabilir.

1.2.7 Tümevarımsal (Inductive) Öğrenme

“Veri uzayı”, ilgilenilen nesnelere olduğu kümedir. Bu küme, örneğin kredi başvurusu yapan müşterilerin bilgileri olabileceği gibi, borsa ile ilgili günlük veriler de olabilir. “Eğitim kümesi (training set)” veri uzayının gözlenebilen alt kümesidir. Veri uzayının tümü ile ilgili veriyi toplamak her zaman mümkün olmadığından, bir altkümesi alınır ve bu altkümenin veri uzayını temsil ettiği varsayılır.

Sürecin nasıl çalıştığı bilinmemekle birlikte, sürecin verilen girdilere karşılık oluşturduğu etiketler gözlenebilmektedir. Eğitim kümesi, gözlemlerden ve bu gözlemlere karşılık gelen etiketlerden oluşur. Eğitim kümesi veri uzayını ne kadar iyi temsil ediyorsa bulunan model de gerçek etiketleme sürecine o kadar yaklaşabilecektir (Hamel, 2009: 5).

Girdi-çıkı çiftleri girdileri çıktılarına eşleyen bir fonksiyonel ilişki taşırlar. Girdilerden çıktılarına bir “*dayanak fonksiyonu (underlying function)*” varsa bu fonksiyon “*hedef fonksiyonu*” olarak adlandırılır. Öğrenme algoritması tarafından öğrenilen hedef fonksiyonunun tahmini, öğrenme probleminin çözümüdür. Öğrenme probleminin bir sınıflandırma problemi olduğu durumlarda bu fonksiyon “*karar fonksiyonu*” olarak tanımlanır.

Çözüm, girdi uzayından çıktı uzayına eşleştirme yapan “*aday fonksiyonlar*” kümesi içerisinde seçilir. Genellikle doğru fonksiyonu öğrenmeye çalışmadan önce “*hipotez kümesi*” olarak da bilinen aday fonksiyonlar kümesi seçilir. Bu nedenle hipotez (fonksiyon) kümesinin seçimi, öğrenme stratejisinin önemli bir adımıdır. Eğitim kümesini girdi olarak alan ve fonksiyon kümesinden bir fonksiyon seçen “*öğrenme algoritması*” da öğrenme stratejisinin önemli bileşenlerinden birisidir (Cristianini & Shawe-Taylor, 2000: 1).

Model, gelecekle ilgili öngörüler yapmak için kullanılabilen *öngörücü (predictive)* ya da veriden bilgi çıkarmaya yönelik *açıklayıcı (descriptive)* bir model olabilir. Oluşturulan model, eğitim kümesinde yer almayan herhangi yeni bir gözlemin sınıfını tahmin etmek için kullanılabilir. Ayrıca, model incelenerek orijinal etiketleme süreci hakkında açıklayıcı bilgiler edinilebilir. Örneğin bir banka, kredi

başvurusu yapan müşterilerini sınıflandırmak için oluşturduğu böyle bir modele bakarak yeni başvuru yapan bir müşterinin kredisini ödeyip ödeyemeyeceğini tahmin edebilir veya kredisini ödeyemeyen müşterilerin ortak özelliklerinin neler olduğu gibi bilgileri edinebilir (Hamel, 2009: 8).

Örneklere öğrenme problemi *çıkarsama (inference)* kavramıyla ilişkilendirilebilir. Bütün öngörücü öğrenme süreçleri temel olarak:

- Verilen eğitim kümesinden sistemdeki bilinmeyen bağımlılıkları öğrenme veya tahmin etme.
- Tahmin edilen bağımlılıkları kullanarak sistemde yer alacak yeni girdiler için çıktılar öngörme.

aşamalarından oluşmaktadır (Kantardzic, 2011: 80).

Bu iki aşama sırasıyla; özelden (eğitim kümesi) genele (model) ulaşmak olarak tanımlanan tümevarım (induction) ve genel bir model ile birlikte verilen girdiden çıktıya ulaşmak olarak tarif edilen tümdengelim (deduction) isimli iki klasik çıkarsama tipine karşılık gelmektedir.

İstatistiksel model tahmini çerçevesinden bakıldığında öğrenme sürecinin amacı, bilinmeyen sistemin doğru *tanınması (identification)* iken, öngörücü öğrenmede sistem çıktısının doğru olarak *taklit (imitation)* edilmesi amaçlanır (Cherkassky & Mulier, 2007: 24).

Öğrenme makinesinin çözmesi gereken problem, uygulayabileceği fonksiyonlar kümesi içerisinde sistemin cevabına en iyi yaklaşan (approximating) fonksiyonu seçmektir. Öğrenme makinesi bu seçimi yapmak için sonlu sayıdaki (n) örneklerle sınırlandırılmıştır. Eğitim verilerinin, üreteç ve sistem tarafından $P(x, y) = P(x) P(y|x)$ ortak olasılık yoğunluk fonksiyonuna göre *bağımsız (independent)* ve *benzer (identical)* olarak oluşturulduğu varsayılır. Bu dağılımdan alınan sonlu örnekler (eğitim kümesi), (x_i, y_i) , $(i = 1, \dots, n)$ şeklinde gösterilir.

Öğrenme makinesinin bulduğu yaklaşmanın kalitesi $L(y, f(x, \omega))$ “*kayıp (loss) fonksiyonu*” ile ölçülür. Tanım gereği kayıp fonksiyonu negatif değerler almaz. Büyük pozitif değerler zayıf yaklaşmayı, sıfıra yakın küçük pozitif değerler ise iyi bir yaklaşmayı gösterir. Kaybın beklenen değeri “*risk fonksiyoneli*” (Denklem 1.1) olarak adlandırılır.

$$R(\omega) = \int L(y, f(x, \omega)) dP(x, y) dx dy \quad (1.1)$$

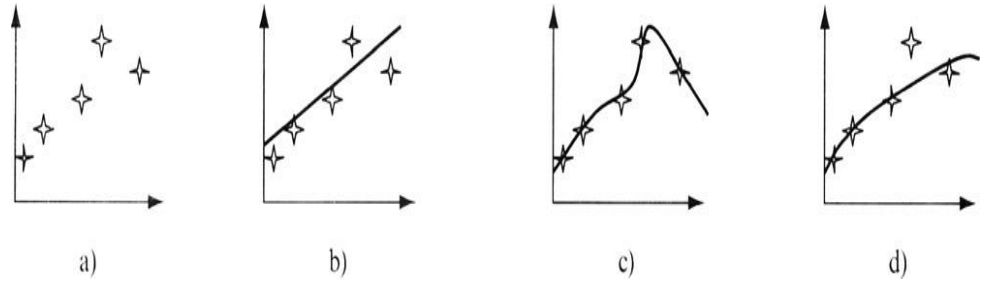
Seçilen bir yaklaşan fonksiyon için $R(\omega)$ risk değeri, sadece ω parametre kümesine bağlıdır. Öğrenme, sadece eğitim kümesini kullanarak ve $P(x, y)$ olasılık dağılımını bilmeden, öğrenme makinesi tarafından desteklenen fonksiyonlar kümesi üzerinde $R(\omega)$ risk fonksiyoneli minimize edecek $f(x, \omega_0)$ dayanak fonksiyonunu tahmin süreci olarak tanımlanır. Sınırlı sayıdaki eğitim verisi kullanılarak $f(x, \omega_0)$ fonksiyonu tam olarak bulunamayacağından, en uygun çözümünün bir tahmini olarak, eğitim kümesinden bir öğrenme yöntemi kullanılarak elde edilen, $f(x, \omega^*)$ yaklaşan fonksiyonu kullanılır (Kantardzic, 2011: 92).

Eğer $P(x, y)$ olasılık dağılımı bilinmiş olsaydı, herhangi bir öğrenme probleminin (Denklem 1.1) de gösterilen risk fonksiyoneli minimize ederek çözülebileceği açıktır. Bu olasılık dağılımı bilinmediğinden, yoğunluk tahmini en genel ve en zor öğrenme problemi tipidir. Sadece sınırlı verinin kullanılarak tek bir çözüm elde edebilmesi için öğrenme sürecine veriye ek olarak önsel bilginin de dahil edilmesi gerekir.

Önsel bilginin, yaklaşan fonksiyonlar kümesine yansıtıldığında öğrenme makinesinin eğitim verisini nasıl kullanacağı sorusu önemlidir. Bu soruya “*tümevarım ilkesi (inductive principle)*” olarak bilinen kavram yoluyla cevap bulunmuştur. Tümevarım ilkesi, gözlenebilen sınırlı eğitim kümesini kullanarak “*doğru bağımlılığın bir tahminini*” yaklaşan fonksiyonlar kümesi içinden elde etmek için genel bir çözüm yolu sunar (Cherkassky & Mulier, 2007: 25).

Makine öğrenmesi, sınırlı miktardaki bir veriyi eğitim kümesi olarak kullanarak tüm veri uzayında geçerli olan etiketleme sürecine yaklaşacak bir fonksiyon bulmaya çalıştığı için, tümevarımsal bir süreçtir. Yani eğitim kümesindeki gözlemlerden yola çıkarak tüm veri uzayına bir genelleme yapılmaktadır. Bu yöntem “tümevarımsal öğrenme” olarak bilinmektedir ve temeli eğitim kümesinin tüm veri uzayını temsil ettiği varsayımına dayanmaktadır. Buna göre tümevarımsal öğrenme hipotezi “Yeterince büyük bir eğitim kümesi üzerinde hedef fonksiyona iyi bir yaklaşma gösteren herhangi bir fonksiyon, gözlenmemiş (yeni) örnekler üzerinde de hedef fonksiyona iyi bir yaklaşma gösterecektir” biçiminde açıklanabilir (Hamel, 2009: 8).

Tüm öğrenme yöntemleri yaklaşan fonksiyon kümesinin seçimi yoluyla önsel bir bilgi kullanırlar. (Şekil 2) de tümevarımsal çıkarsamanın basit bir örneği görülmektedir. Burada amaç verilen eğitim noktalarına (a), en iyi uyacak fonksiyonu bulmaktır. Ek bir bilgi olmadan b, c, d ile gösterilen çözümlerden birini diğerine tercih etmek için bir ölçüt bulunmamaktadır.



Şekil 2. Verilen Bir Eğitim Kümesi İçin Bulunan Üç Hipotez

Çoğu kez, tutarlı olan birçok farklı hipotez bulmak mümkün olduğundan, tüm arama algoritmaları belli bir ölçüte göre çözüm uzayında arama yaparlar. Örneğin ölçüt, tüm eğitim gözlemlerine toplam mesafesi en küçük olan doğrusal yaklaşan bir fonksiyon olabilir. Bu önsel bilgi, arama uzayını (Şekil 2.b) deki gibi bir fonksiyon kümesiyle sınırlandıracaktır.

1.2.8 Temel Öğrenme Problemleri

Öğrenme algoritmaları günlük hayatta birçok probleme uygulanmaktadır. Bu uygulamalar çeşitli öğrenme problemlerine karşılık gelmektedirler. Genel olarak öğrenme problemi, sınıflandırma, regresyon, yoğunluk tahmini ve kümeleme olarak dört problem alt sınıfına ayrılabilir. Bu problemlerin her biri için kayıp fonksiyonu ve çıktı farklı olabilir. Bununla birlikte, yalnızca eğitim verisine bağlı olan risk fonksiyonelinin minimize edilmesi bu problemlerin tümünde ortak hedeftir (Cherkassky & Mulier, 2007: 25).

1.2.8.1 Sınıflandırma

Sınıf etiketleri farklı değerler alabilir. $\{0,1\}$ değerlerini alabileceği gibi $\{\text{doğru}, \text{yanlış}\}$ değerlerini de alabilir. Önemli olan kümenin biri sınıfa üye olma durumunu, diğeri sınıfa üye olmama durumunu gösteren iki farklı etikete sahip olmasıdır. Yukarıdaki tanım sadece ikili sınıflandırma problemi için gösterilmiştir. Fakat uygun sayıda farklı etiketler belirtilmek koşuluyla tanım çoklu sınıflandırma için de genişletilebilir (Hamel, 2009: 6).

İkili sınıflandırma probleminde sistemin çıktısı sadece sınıflara karşılık gelen $y = \{-1, +1\}$ veya $y = \{0,1\}$ gibi değerleri alabilir. Bu nedenle, $f(x, \omega)$, $\omega \in \Omega$ fonksiyonlar kümesi “gösterge (*indicator*)” fonksiyonlar kümesi haline gelir.

İkili sınıflandırma problemi için yaygın olarak kullanılan kayıp fonksiyonu sınıflandırma hatasını ölçer (Cherkassky & Mulier, 2007: 26).

$$L(y, f(x, \omega)) = \begin{cases} 0, & y = f(x, \omega) \text{ ise} \\ 1, & y \neq f(x, \omega) \text{ ise} \end{cases} \quad (1.2)$$

(1.2) kayıp fonksiyonu (Denklemler 1.1) de gösterilen $R(\omega)$ risk fonksiyonelinde kullanıldığında, risk fonksiyoneli yanlış sınıflandırma olasılığını ölçer. Bu durumda öğrenme, yalnızca eldeki eğitim verisini kullanarak, yanlış sınıflandırma olasılığını minimize eden $f(x, \omega_0)$ gösterge fonksiyonunun tahmini problemi haline gelir (Kantardzic, 2011: 92).

Sınıflandırma oranlarının ölçümünde her bir sınıflandırma kararının birbirinden bağımsız ve eşit önemde olduğu kabul edilir. Fakat bu varsayımlar eğer veride bazı bağımlılıklar varsa ihlal edilebilir.

Örneğin, eldeki erkek veya kız olarak etiketlenilmiş fotoğrafları inceleyerek yeni fotoğrafları erkek veya kız olarak sınıflandıracak bir şekil tanıma görevi ele alındığında eğitim kümesinde yer alan fotoğrafların alındığı yer de önemli hale gelmektedir. Fotoğrafların bir okuldaki büyük çoğunluğu kızlardan oluşan bir sınıftan (örneğin çocuk gelişimi sınıfı) veya büyük oranda erkeklerin yer aldığı bir sınıftan (örneğin motor sınıfı) alınması sınıflandırıcının performansını etkileyecektir. Eğer bir sınıflandırıcı, verinin içinde yığınlar (batch) olduğunu bilirse bir yığının bitip diğer yığının başladığı noktayı belirlemeye çalışarak daha yüksek doğruluk oranına ulaşmaya çalışır. Daha sonra bu yığınların içindeki her bir fotoğrafı tek bir sınıfa atayabilir. Bu tür bağımlılıklar sıklıkla meydana gelmektedir. Örneğin, bir doktor bulaşıcı hastalığa kapılan hastaların dalgalar halinde hastaneye geldiklerini bilir. Bu nedenle art arda gribe yakalanan birkaç hastayı gördükten sonra, büyük olasılıkla bir sonraki hastayı belirtileri önceki hastalar kadar açık olmasa bile grip hastası olarak sınıflandırmaya eğilim gösterebilir (Dietterich, 2003: 3).

Yanlış sınıflandırmadan kaynaklanan farklı maliyetlerin veya risklerin olduğu durumlarda sınıfların eşit önemde olması varsayımı ihlal edilebilir. Örneğin bazı laboratuvar ölçümlerine göre bir hastanın kanser olup olmadığına karar verecek bir sınıflandırıcı ele alındığında iki tip hata vardır. Bu hatalardan “yanlış pozitif (*false positive*)” sınıflandırıcının sağlam bir hastayı kanser hastası olarak sınıflandırdığında, “yanlış negatif (*false negative*)” ise gerçekte kanser olan bir hastayı sağlıklı olarak sınıflandırdığında oluşur (Vercellis, 2009: 231). Genel olarak yanlış negatif hatalar yanlış pozitif hatalardan daha maliyetlidir. Bu nedenle öğrenme algoritmasından daha az yanlış negatif hatalar yapan bir sınıflandırıcı tercih etmesi istenebilir.

1.2.8.2 Regresyon

Regresyon problemlerinde sistem çıktısı reel değerler alan rassal bir değişkendir. Bu nedenle regresyon, sınırlı sayıdaki gürültülü eğitim kümesinden reel değerli fonksiyonlar tahmin etme sürecidir. Çıktı, deterministik bir fonksiyonla sıfır ortalamalı rassal hatanın toplamı olarak

$$y = t(x) + \xi \quad (1.3)$$

şeklinde gösterilir (Wasserman, 2003: 335). (Denklem 1.3) de gösterilen deterministik fonksiyon

$$t(x) = \int yP(y|x)dy \quad (1.4)$$

koşullu olasılık çıktısının ortalamasıdır.

Öğrenme makinesi tarafından uygulanabilen $f(x, \omega)$, $\omega \in \Omega$ fonksiyonlar kümesi (Denklem 1.4) deki fonksiyonu içerebilir veya içermeyebilir. Regresyonda yaygın olarak

$$L(y, f(x, \omega)) = (y - f(x, \omega))^2 \quad (1.5)$$

olarak gösterilen hata kareli kayıp fonksiyonu kullanılır. Bu durumda öğrenme

$$R(\omega) = \int (y - f(x, \omega))^2 P(x, y) dx dy \quad (1.6)$$

risk fonksiyoneli minimize edecek $f(x, \omega_0)$ fonksiyonu bulma problemi haline gelir. Bu risk fonksiyoneli, öğrenme makinesinin sistem çıktılarını öngörme doğruluğunu ölçer.

1.2.8.3 Yoğunluk Tahmini

Yoğunluk tahmini için sistemin çıktısı kullanılmaz. Öğrenme makinesinin çıktısı yoğunluğu temsil ettiğinden $f(x, \omega)$, $\omega \in \Omega$ bir yoğunluklar kümesi haline gelir. Bu problem için “maksimum olabirlik (*maximum likelihood*)” kriteri kullanılır (MacKay, 2003: 300). Kayıp fonksiyonu olarak

$$L(f(x, \omega)) = -\ln f(x, \omega) \quad (1.7)$$

kullanılır. (1.7) kayıp fonksiyonu (1.1) denklemdeki risk fonksiyoneline yazıldığında, yoğunluk tahmini için kullanılan yaygın bir risk fonksiyoneli olan

$$R(\omega) = \int -\ln f(x, \omega)P(x)dx \quad (1.8)$$

elde edilir. Sadece x_1, \dots, x_n eğitim verilerini kullanarak (1.8) denkleminin minimize edilmesiyle $f(x, \omega_0)$ yoğunluk tahminine ulaşılır.

1.2.8.4 Kümeleme

Amaç, gelecek örneklerin tek bir nokta (küme merkezi) ile yaklaşılabilmesi için, girdi uzayındaki bilinmeyen dağılımın önceden belirlenen bir sayıdaki bölgeye (küme) bölünmesidir (Hsieh, 2009: 15). Bu problemde reel değerli $f(x, \omega)$, $\omega \in \Omega$ fonksiyon kümesi “vektör nicelendiricileri (vector quantizers)” olarak adlandırılır.

Bir vektör nicelendiricisi

$$x \xrightarrow{f(x, \omega)} c(x) \quad (1.9)$$

eşlemesini sağlar.

(1.9) denklemdeki $c(x)$, küme merkezinin koordinatlarını göstermektedir. Bu yolla, girdiler kesikli sayıdaki merkez noktalara eşlenmektedir. Vektör niceleyicisi, tamamen küme merkez koordinatları ve girdi uzayının bölünmesiyle (partitioning) tanımlanmaktadır. Bu öğrenme problemi tipinde, (\cdot) iç çarpım operatörünü göstermek üzere, genel olarak

$$L(f(x, \omega)) = (x - f(x, \omega)) \cdot (x - f(x, \omega)) \quad (1.10)$$

şeklindeki kayıp fonksiyonu kullanılır. Risk fonksiyoneli ise

$$R(\omega) = \int (x - f(x, \omega)) \cdot (x - f(x, \omega))P(x)dx \quad (1.11)$$

halini alır.

(1.11) denklemindeki risk fonksiyonelinin minimize edilmesi, gözlenen veriye dayanan optimal vektör niceleyicisini verir. Vektör niceleyicisi, $P(x)$ yoğunluğundan oluşacak olan gelecek gözlemlerin optimal olarak nicelenmesi için tasarlanır. Bu bağlamda vektör niceleme bir öğrenme problemidir ve verilen sonlu bir veri kümesinin sıkıştırılması amacından farklılıklar gösterir. Burada “*veri azaltma (data reduction)*” hedeflenmektedir (Cherkassky & Mulier, 2007: 29).

1.2.9 Öğrenme Yöntemleri

Girdi verisinden çıktının hesaplanması için belli bir yöntemin bulunmadığı veya hesaplamaların çok maliyetli olduğu durumlarda alternatif olarak girdi-çıkı arasındaki ilişki bilgisayar yardımıyla örneklerden yararlanılarak bulunmaya çalışılır. Bu, bir çocuğa spor arabaların özelliklerini tek tek belirtmeden birçok farklı araba içerisinden hangilerinin spor araba olduğunun gösterilmesi sonucu çocuğun artık spor arabaları ayırt edebilmesine benzemektedir (Cristianini & Shawe-Taylor, 2000: 1).

Bir tümevarım ilkesi veriyle ne yapılacağını, öğrenme algoritması ise tahminin nasıl elde edileceğini belirtir. Bu nedenle bir “*öğrenme algoritması*”, bir $f(x, \omega)$ fonksiyonlar kümesi içinden $f(x, \omega^*)$ tahminini seçmek için tümevarım ilkesinin bir uygulamasıdır. Bir tümevarım ilkesi için, makinenin uygulayabildiği farklı fonksiyon kümelerine karşılık gelen değişik öğrenme algoritmaları olabilir. Esas mesele eğitim verisini tanımlayabilecek doğru karmaşıklıkta aday modelleri seçebilmektir.

1.2.9.1 Denetimli (Supervised) Öğrenme

Sınıflandırma ve regresyon problemleri ile ilgili en yaygın öğrenme yöntemi denetimli öğrenmedir.

$$x = (x_i, y_i), \quad i = 1, \dots, n \quad (1.12)$$

(Denklem 1.12) de x ile gösterilen eğitim kümesindeki örnekler bağımsız ve benzer dağılmıştır. Yani, örneklerin küme içindeki sırası önemli değildir ve tüm

örnekler aynı $P(x)$ olasılık dağılımından örneklenmiştir. İndis değişkeni i , n gözlemden birini gösterir. x_i girdi ve y_i istenen çıktı değeridir. y_i ikili sınıf öğrenmede 0/1, $K > 2$ sınıflandırmada K boyutlu (yalnızca bir boyutun 1, diğerlerinin 0 olduğu) bir vektör, regresyonda ise reel bir değerdir.

Amaç, $f(x_i, \omega)$ modelini kullanarak y_i değerine yakın ve kullanışlı bir yaklaşma elde etmektir (Barber, 2012: 285).

1.2.9.2 Denetimsiz (Unsupervised) Öğrenme

Bu öğrenme yönteminde etiketli gözlemler kullanılmadığı için öğrenme sonucunu nicel olarak değerlendirmek biraz daha zordur. Kümeleme ve boyut azaltma bu öğrenme yöntemine örnek olarak gösterilebilir.

Girdi uzayının bir yapısı vardır. Bu yapıda bazı örneklere daha sık rastlanır. İstenilen, neyin sık neyin seyrek gerçekleştiğini görmektir. İstatistikte buna yoğunluk tahmini denir. Yoğunluk tahmini için kümeleme kullanılabilir. Bir şirket, geçmiş müşterilerinin hareketleri yanında onların demografik bilgilerini saklayarak müşterilerinin profilini yani ne tür müşterilerinin hangi yoğunlukta olduğunu görmek ister. Bu durumda, bir kümeleme algoritması birbirlerine benzeyen müşterileri aynı kümeye atar. Bu kümeler bulunduktan sonra şirket her bir küme için farklı hizmet ve ürünlere karar verebilir. Kümeleme uç örnekleri (outlier), yani genelden farklı müşterileri bulmak için de kullanılabilir. Böyle müşteriler henüz yeterince değerlendirilmeyen yeni bir alanı (niş) gösterebilir (Alpaydın, 2011: 8-9).

1.2.9.3 Yarı Denetimli (Semi Supervised) Öğrenme

Yarı denetimli öğrenmede öğrenme makinesi hem etiketli hem etiketsiz gözlemlerden oluşan eğitim kümesini alır ve bu kümede yer almayan yeni gözlemler için öngörüler yapar. Bu öğrenme yöntemi etiketsiz verilerin kolayca erişilebildiği fakat etiketleri elde etmenin maliyetli olduğu durumlarda kullanılır. Etiketsiz gözlemlerin de kullanılmasının sadece denetimli öğrenmenin kullanılmasına göre daha iyi sonuçlar vereceği düşünülür. Sınıflandırma, regresyon türündeki bazı problemler yarı denetimli öğrenme ile çözülmektedir (Mohri vd., 2012: 7).

1.2.9.4 Transdüktif (Transductive) Öğrenme

Tahmin edilen tek bir fonksiyonun tüm olası girdi değerleri için kullanılabilmesi düşünülebilir. Fakat pratikte karşılaşılan problemlerin birçoğunda sadece belirli girdi değerleri için çıktılarının oluşturulması beklenmektedir. Bu durumda, daha iyi bir yaklaşım global bir model oluşturmadan fonksiyonu sadece belirli gözlemlerin çıktılarını verecek şekilde tahmin etmek olabilir. Yerel tahminin global tahminden daha önemli olduğu bu yaklaşıma “*transdüktif (özelden özele) çıkarsama*” denilir. Bu yaklaşımın önemli bir uygulaması “*ilişki kurallarının (association rules)*” oluşturulmasıdır. Makine öğrenmesinin standart çerçevesi bu tip çıkarsamaya uygulanmamaktadır (Kantardzic, 2011: 88).

Transdüktif öğrenmede yarı denetimli öğrenmede olduğu gibi etiketli ve etiketsiz gözlemlerden oluşan eğitim kümesi öğrenme makinesine sunulur. Amaç, sadece belirli test gözlemleri için öngöründe bulunmaktır. Çeşitli uygulamaların ihtiyaçları bu öğrenme türü ile eşleşmektedir. Fakat yarı denetimli öğrenmede olduğu gibi bu öğrenme çerçevesinde hangi varsayımlar altında daha iyi performans elde edilebileceği sorusu halen tam olarak çözülebilmemiş değildir (Mohri vd., 2012: 7).

1.2.9.5 Pekiştirmeli (Reinforcement) Öğrenme

Bazı uygulamalarda sistemin çıktısı bir eylem dizisidir. Böyle bir durumda önemli olan tek bir eylem değil, bir politika yani, hedefe ulaşmak için gerçekleştirilen doğru eylem dizisidir. Ara bir durumda en iyi eylem tanımlanamaz, bir eylem ancak iyi bir politikanın parçasıysa iyidir. Böyle bir uygulamada makine öğrenmesi yazılımı, politikaların iyiliklerini değerlendirebilmeli ve geçmişteki iyi eylemlerden başarılı bir politika üretebilmelidir.

Pekiştirmeli öğrenmede eğitim ve test aşamaları birbirine karışmış durumdadır. Öğrenme makinesi bilgi toplamak için çevre ile etkileşim halindedir ve bazı durumlarda çevreyi etkiler. Her bir eylemden sonra hemen bir ödül alır. Öğrenme makinesinin hedefi bir eylemler dizisinden sonra aldığı ödülü maksimize etmektir. Bu tür öğrenmeye pekiştirmeli öğrenme denilir (Engelbrecht, 2007: 83).

1.2.9.6 Aktif (Active) Öğrenme

Aktif öğrenmede öğrenme makinesi bir uzmanı sorgulayarak yeni gözlemlerin etiketlerini öğrenmeye çalışır. Böylece etkileşimli olarak eğitim örneklerini toplar. Amaç, daha az sayıda etiketli gözlem kullanarak denetimli öğrenmeye yakın bir performans elde etmektir. Genellikle etiketleri elde etmenin maliyetli olduğu durumlarda kullanılır (Mohri vd., 2012: 8).

1.2.10 Öğrenme Algoritmalarının Bileşenleri

Makine öğrenmesi yöntemlerinin uygulanmasında karşılaşılan temel zorluklardan birisi çok sayıda algoritmadan hangisinin kullanılacağına belirlenmesidir. Literatürde çok sayıda makine öğrenmesi algoritması vardır. Bunlara ek olarak her yıl yeni birçok algoritma yayınlanmaktadır. En uygun algoritmayı seçebilmek için bu algoritmaların temelde 3 bileşenden oluştuğunu bilmek gerekir. (Tablo 1) bu üç bileşenin yaygın örneklerini göstermektedir. Bu bileşenler; gösterim (representation), değerlendirme fonksiyonu ve optimizasyon tekniği olarak sayılabilir (Domingos, 2012: 78-79).

Tablo 1. Öğrenme Algoritmalarının Üç Temel Bileşeni

Gösterim	Değerlendirme	Optimizasyon Tekniği
Gözlemler	Doğruluk/Hata oranı	Kombinatorik Optimizasyon
K-En Yakın Komşu	Hassaslık/Hatırlama	Açgözlü (Greedy) Arama
Destek Vektör Makineleri	Kareli Hata	Işın (Beam) Arama
Hiperdüzlemler	Olabilirlik	Dal Sınır Arama
Naive Bayes	Sonsal Olasılık	Sürekli Optimizasyon
Lojistik Regresyon	Bilgi Kazanımı	Kısıtsız Optimizasyon
Karar Ağaçları	K-L İraksaması	Eğimli İniş
Kural Kümeleri	Maliyet/Fayda	Eşlenik Eğimli İniş
Önerme Kuralları	Marj	Quasi-Newton Yönt.
Mantık Programları		Kısıtlı Optimizasyon
Yapay Sinir Ağları		Doğrusal Programlama
Grafiksel Modeller		Kuadratik Programlama
Bayesyen Ağlar		
Koşullu Rastgele Alanlar		

1.2.10.1 Gösterim

Bir sınıflandırıcının bilgisayarın anlayabileceği bir biçimsel dilde (formal language) gösterilebilmesi gerekir. Diğer taraftan, öğrenme makinesi için bir gösterimin belirlenmesi, öğrenme makinesinin uygulayabileceği muhtemel sınıflandırıcı kümesini seçmekle eşdeğerdir. Bu kümeye öğrenme makinesinin “*hipotez kümesi*” denilir. Eğer bir sınıflandırıcı bu hipotez kümesinde değil ise öğrenilemez.

Bir gösterimi seçmenin temel ölçütü eldeki önsel bilginin bu gösterimde kolayca ifade edilip edilemeyeceğidir. Örneğin gözlemleri birbirine benzer yapan etkenlerin neler olduğu hakkında önsel bilgi varsa “*gözlem tabanlı (instance based)*” yöntemlerin kullanılması iyi bir tercih olabilir. Buna benzer biçimde olasılıksal bağımlılıklar hakkında bir önsel bilgi olması durumunda “*grafiksel modellerin (graphical models)*” daha iyi sonuçlar vermesi beklenebilir.

$f(\cdot)$ ile model, x ile girdiler ve ω ile de model parametreleri gösterilmek üzere, bir $f(\cdot)$ modeli seçildiğinde bir fonksiyon kümesi tanımlanmış olur. Bu küme içinde her ω değeri farklı bir fonksiyona karşılık gelir. Modeli ve modelin varsayımlarını bir başka deyişle fonksiyon kümesini, makine öğrenme tasarımcısı uygulama hakkında önsel bilgilerini de göz önüne alarak belirler. Öğrenme algoritması, verilen bir fonksiyon kümesi içinden sabit fakat bilinmeyen olasılık dağılımından örneklenen eğitim kümesi üzerinde en uygun fonksiyonu seçmeye yani optimal ω parametrelerini bulmaya çalışır (Alpaydın, 2011: 31).

Hedef fonksiyonunun bir yaklaşması hesaplanmak istenildiğinden bu yaklaşan fonksiyonların (modellerin) nasıl temsil edileceğinin de belirtilmesi gerekir. Genel olarak iki tip model temsili kullanılmaktadır:

1. Şeffaf modeller
 - a. If-Then–Else kuralları

- b. Karar ağaçları
2. Şeffaf olmayan modeller
- a. Bir yapay sinir ağında bilgi işlem birimleri arasındaki bağlantıların ağırlıkları
 - b. Destek vektör makineleri yönteminde vektörlerin doğrusal birleşimleri

Şeffaf modeller, sonuçları kolayca yorumlanabilen modellerken şeffaf olmayan modellerin sonuçlarını yorumlamak daha zordur. Örneğin If-Then-Else yapısındaki bir kural incelendiğinde bu kural kolayca anlaşılabilirken, bir yapay sinir ağı modelinin oluşturduğu ağırlık değerleri incelendiğinde bu ağırlıkların ne anlama geldiği kolayca yorumlanamaz (Hamel, 2009: 10).

Şeffaf modeller, performans bakımından şeffaf olmayan modellere göre genellikle daha geridedirler. Bu modellerdeki sonuçların kolaylıkla yorumlanabilmesi şartı modelleme sürecini etkilediğinden şeffaf olmayan modeller kadar etkili sınıflandırma performansı gösteremezler.

1.2.10.2 Değerlendirme (Amaç) Fonksiyonu

Bir değerlendirme fonksiyonu iyi sınıflandırıcıları kötü sınıflandırıcılardan ayırmak için kullanılır. Amaç fonksiyonlarında genelde kayıp fonksiyonları kullanılır. Bir $L(\cdot)$ kayıp fonksiyonu, beklenen y_i çıktısıyla modelin $f(x_i, \omega)$ öngörüsü arasındaki fark nedeniyle oluşan maliyeti gösterir (Camastra & Vinciarelli; 2007: 96).

İkili sınıflandırma probleminde çıktılar 0/1 olduğu için $L(\cdot)$ beklenen çıktı ile modelin öngörüsünün eşit olup olmadığına bakar; regresyonda çıktı reel bir değerdir ve fark veya farkın karesi kullanılabilir. (Tablo 1) de yer alan değerlendirme

kriterlerinden uygun olan birisi seçilir. Seçilen bu kriterin probleme göre maksimize veya minimize edilmesi gerekir.

1.2.10.3 Optimizasyon Tekniđi

Sınıflandırıcılar arasında optimal sınıflandırıcıyı bulmak için bir arama yöntemine ihtiyaç duyulur. Optimizasyon tekniđinin seçimi öğrenme makinesinin etkinliđi için özel bir öneme sahiptir. Ayrıca, amaç fonksiyonunun birden fazla optimum deđer bulduđu durumlarda hangi sınıflandırıcının tercih edileceđini belirler.

Optimizasyon tekniđi (Denklem 1.13) de gösterildiđi gibi toplam hatayı (E) en aza indirecek ω^* parametrelerini bulmayı amaçlar:

$$\omega^* = \arg \min_{\omega} E(x, \omega) \quad (1.13)$$

Regresyon gibi problemlerde en iyi deđer analitik olarak bulunabilirken; daha karmaşık modeller ve kayıp fonksiyonları için eğim tabanlı yöntemler, benzetimli tavlama (simulated annealing), ya da genetik algoritmalar gibi daha karmaşık optimizasyon tekniklerinin kullanılması gerekir.

Özetle, öğrenmenin başarılı olabilmesi için şu koşulların sağlanması gerekir:

- $f(\cdot)$ modelinin çekildiđi fonksiyon kümesi, verinin altında yatan ve y_i çıktılarında gürültü eklenmiş olarak kendini gösteren dayanak fonksiyonunu içerecek kadar geniş yani yeterli kapasiteye sahip olmalıdır.
- Fonksiyon kümesi içinden dođru (ya da yeterince dođru) fonksiyonu bulmaya yetecek kadar eğitim verisi olmalıdır.
- Eğitim kümesi verildiđinde dođru fonksiyonu bulabilecek etkin bir optimizasyon tekniđi kullanılmalıdır (Alpaydın, 2011: 32).

Makine öğrenmesi algoritmaları gösterimlerindeki, değerlendirme fonksiyonlarındaki ya da kullandıkları optimizasyon tekniklerindeki farklılıklardan dolayı birbirinden ayrılırlar.

1.2.11 Model Seçimi

İlk makine öğrenmesi algoritmalarında alan uzmanları tarafından anlaşılabilen ve doğrulanabilen basit sembolik fonksiyonların gösterimlerinin öğrenilmesi amaçlanmıştır. Bu yaklaşımda öğrenme hedefi eğitim verisini doğru sınıflandırabilen bir fonksiyonu bulmak olduğundan, bu algoritmalarda sadece eğitim verisine uyan bir fonksiyon bulunmaya çalışılmıştır. Böyle bir fonksiyon tutarlı olarak kabul edilmiştir. Fakat doğrulanabilir tutarlı bir fonksiyon oluşturma hedefinde karşılaşılan iki temel problem vardır (Cristianini & Shawe-Taylor, 2000: 3).

İlk problem öğrenilmek istenilen fonksiyonun basit bir gösterime sahip olmaması nedeniyle kolayca doğrulanamamasıdır. Bu duruma örnek olarak bir DNA dizisindeki genlerin tespiti problemi gösterilebilir. DNA dizisinin bir kısım alt dizileri genlerden oluşurken bazıları genlerden oluşmamaktadır ve bunları kategorize etmenin basit bir yolu bilinmemektedir.

İkinci problem ise sıklıkla eğitim verilerinin gürültülü olmasıdır. Eğitim gözlemlerini doğru bir şekilde ilişkilendiren bir dayanak fonksiyonun varlığı garanti değildir. Gürültü, verideki istenmeyen aykırılıklardır. Gürültü yüzünden bir sınıfı öğrenmek zorlaşır. Basit bir fonksiyon sınıfıyla sıfır eğitim hatasına ulaşmak olanaksız olabilir. Gürültünün oluşması aşağıdaki nedenlerden kaynaklanabilir.

- Gözlemlerin kaydında yapılan hatalar.
- Gözlemlerin etiketlenmelerinde yapılan hatalar.
- Sınıf etiketini etkileyen ancak dikkate alınmayan veya gözlenemeyen başka saklı öznitelikler olabilir. Örneğin kredi denetimi problemi bu sınıfa girer çünkü ödeme zorluğuna düşme durumunu sisteme dahil edilemeyen başka

faktörler de etkilemiş olabilir. Göz ardı edilen bu özneliklerin etkisi rastgele bir bileşen olarak modellenip “gürültü” olarak nitelenir.

Gürültü olduğunda sınıfları birbirinden ayıran sınır basit olmayabilir ve onları ayırmak için daha yüksek kapasiteli bir fonksiyon kümesinden daha karmaşık bir yaklaşan fonksiyon kullanmak gerekir (Alpaydın, 2011: 23).

Verideki miktarına bağlı olarak gürültü, bilgi keşfi sürecini tehlikeye atan hayati bir problem haline alabilir. Gürültünün veri üzerindeki etkisi veri girişi sırasında öznelikler üzerine kısıtlar koyarak engellenebilir. Örneğin verilerin depolanmasını ve yönetimini sağlayan veritabanı yönetim yazılımları genellikle her bir öznelik için özelleştirilmiş kısıtlar oluşturma imkânı sunmaktadır (Cios, 2007: 42).

Makine öğrenmesi projelerinde kullanılması gereken veriler bahsedilen problemleri taşıyan özellikte olabilmektedir. Bu tür veriler üzerinde önerilen performans ölçütünü uygulamak zordur. Ortaya çıkan temel sorun, eğitim verisiyle tutarlı olan bir fonksiyon bulunsa bile bulunan bu fonksiyonun eğitim kümesinde yer almayan yeni gözlemleri doğru sınıflayamamasıdır. Bir sınıflandırıcının eğitim kümesinde yer almayan gözlemleri de doğru bir şekilde sınıflandırabilmesi yeteneğine “genelleme” denilir. Asıl amacın genelleme olarak değiştirilmesi, modelin bulduğu fonksiyonu dayanak fonksiyonunun tam doğru bir gösterimi olarak görme ihtiyacını ortadan kaldırır. Eğer fonksiyon eğitimde kullanılmayan veriler için de doğru çıktıyı verirse genelleme şartını sağlar (Cristianini & Shawe-Taylor, 2000: 3).

Model seçimi, olası fonksiyon kümeleri içinden en uygun olanını seçmektir. Bu seçimi yaparken, makine öğrenmesinin amacının eğitim kümesini kopya etmek değil bunun yerine istenenin yeni durumlar için doğru öngörü yapmanın (genellemenin) en önemli amaç olduğu unutulmamalıdır. Yani, eğitim kümesinde olmayan girdi örnekleri için de doğru çıktılar oluşturabilme asıl hedeftir.

Bir eğitim kümesi ve fonksiyon kümesi verildiğinde bu kümenin içinden eğitim verisine mükemmel şekilde uyan bir fonksiyon bulunabilir. Ama doğru fonksiyon kümesi seçilmemişse bu kümenin içinden hangi fonksiyon seçilirse seçilsin bu öğrenme makinesinin iyi genelleme yapabilmesi mümkün değildir.

Öğrenme sürecinde olası tüm hipotezlerle başlanılır. Eğitim gözlemleri görüldükçe bu gözlemlerle tutarlı olmayan hipotezler atılır. Eğer öğrenme kümesi anakütle yerine anakütlenin bir altkümesini içeriyorsa (genelde böyle olur) yani çıktılarının ne olması gerektiği olası durumların yalnızca küçük bir kısmı için biliniyorsa tek bir çözüm elde edilemez. Bu tür problemler “*kötü konumlanmış (ill posed)*” problem olarak adlandırılır (Cherkassky vd., 1999: 1076). Bu durumda yalnızca veri, tek bir çözüm elde etmek için yeterli değildir. Daha çok öğrenme örneği gördükçe veriyi oluşturan dayanak fonksiyon hakkında daha çok bilgi sahibi olunur ve daha çok sayıda yaklaşan fonksiyon tutarsız bulunup elenebilir ama yine de geriye birçok tutarlı yaklaşan fonksiyon kalabilir.

Olası tüm fonksiyonlar kümesi üzerinde herhangi bir kısıt konulmadığı takdirde, büyüklüğü ne olursa olsun hiçbir eğitim kümesi görülmeyen örnekleri sınıflandırmaya yetmeyeceğinden öğrenme imkânsız hale gelecektir. Fonksiyonların veriyi gördükten sonra seçilmesi de problemler doğurmaktadır. Öğrenme kötü konumlanmış bir problem olduğundan ve veri kendi başına çözüme ulaşmak için yeterli olmadığından, bu durumda eldeki veriyle tek bir çözüm elde edebilmek için bazı ek varsayımların yapılması gerekir. Öğrenmeyi gerçekleştirilebilir kılmak için yapılan bu tür varsayımlara “*model varsayımları*” veya “*öğrenme yanlılığı*” denir (Cristianini & Shawe-Taylor, 2000: 3).

Model varsayımlarının bir türü, bir yaklaşan fonksiyonlar kümesi seçerek olası yaklaşan fonksiyonları kısıtlamaktır. Örneğin regresyon probleminde doğrusal bir fonksiyon kullanmak bir model varsayımı olduğu gibi olası tüm doğrular içinden kareli hatayı en aza indireni seçmek de başka bir model varsayımdır.

Ama her fonksiyon kümesinin bir kapasitesi vardır ve ancak belirli fonksiyonları öğrenebilir. Öğrenilebilecek fonksiyon kümesi kapasitesi daha büyük

fonksiyonlar içeren bir fonksiyon kümesi kullanılarak genişletilebilir. Regresyon örneğindeki polinom derecesi artırıldığında kapasite artacaktır (Alpaydın, 2011: 29).

En iyi genelleme için veri altında yatan dayanak fonksiyonun kapasitesinin fonksiyon kümesinin kapasitesine eş olması gerekir. Destek vektör makineleri gibi öğrenme algoritmaları kapasite ayarlaması yapabilirler. Fakat naive bayes ve basit algılayıcı (perseptron) gibi öğrenme algoritmalarının sınıflandırıcı kapasitesini ayarlayabilme özellikleri yoktur. Bu algoritmalar görece olarak basit sınıflandırıcıları ele alırlar. Sonuç olarak, küçük eğitim kümelerinde oldukça iyi sonuçlar vermelerine rağmen, eğitim verisi arttıkça sınıflandırma performansları azalır. Bunun nedeni ise veriye uyumlarının düşük olmasıdır (Dietterich, 2003: 7).

1.2.11.1 Üçlü Ödünleşim

Tüm öğrenme algoritmaları üç faktör arasındaki “*ödünleşim (tradeoff)*” ile mücadele ederler. Bu faktörler öğrenilen sınıflandırıcının kapasitesi veya karmaşıklığı, eğitim kümesinin büyüklüğü ve yeni örnekler üzerindeki genelleme yeteneğidir. Spesifik olarak, yeni örnekler üzerindeki genelleme yeteneği eğitim verisi arttıkça artar. Öğrenilen sınıflandırıcının karmaşıklığı arttıkça genelleme yeteneği önce artar sonra azalır.

1.2.11.1.1 Eğitim Verisi Miktarı - Genelleme Ödünleşimi

Genelleme ile eğitim verisi miktarı arasındaki ilişki sezgisel olarak anlaşılabilir bir ilişkidir. Eğitim verisinin miktarı arttıkça öğrenme algoritması çözülmeye çalışılan problem hakkında daha çok kanıta sahip olacaktır. Uç durum olarak veri kümesi olası tüm örnekleri taşıyabilir. Böylece olası tüm örneklerin doğru etiketlerini bileceğinden kusursuz bir genelleme yapabilir (Dietterich, 2003: 5).

1.2.11.1.2 Kapasite - Genelleme Ödünleşimi

Genelleme ile sınıflandırıcı kapasitesi arasındaki ilişki ise çok açık değildir. Örneğin, sınıflandırma amacıyla kullanılan bir karar ağacının aşırı derecede büyümesine izin verildiği durumda, bu karar ağacının her sınıflandırıcıyı temsil edebileceği ispatlanabilir. Bu durumda karar ağacının “*düşük yanlılığa (bias)*” sahip olduğu söylenir. Aşırı büyümüş ağaç, istikrarsız bir durumun işaretçisi olan, her bir

yaprağı sadece birkaç eğitim gözlemine sahip olacak şekilde sonlanır. Eğitim gözlemlerinin ölçümünde veya etiketlenmesi sürecinde gürültü oluşmuşsa bu durumda sınıf etiketi de hatalı sonuç verecektir. Böyle sınıflandırıcılar “*yüksek varyanslı*” olarak adlandırılır. Nedeni ise eğitim gözlemlerindeki küçük bir değişikliğin sınıflandırma kararlarını değiştirmesidir. Örnekteki karar ağacı eğitim verilerini ezberlediğinden eğitim kümesi üzerinde mükemmel bir performans gösterse bile genellikle zayıf bir genelleme gösterecektir. Bu durumda sınıflandırıcının veriye “*aşırı uyumundan (overfitting)*” söz edilir (Engelbrecht, 2007: 97).

Diğer bir uç örnek olarak, karar ağacının tek bir düğüm ve iki yapraktan oluşması durumu ele alınırsa her bir yapraktaki sınıf etiketi çok sayıda eğitim gözlemine dayanacaktır. Bu durumda karar ağacı sınıflandırıcısının sonucunun değişmesi için eğitim verisinde büyük bir değişime gerek duyulacağından bu tür bir sınıflandırıcı “*düşük varyanslı*” sınıflandırıcı olarak adlandırılır. Eğer bir sınıflandırıcının problemi temsil etme şekli optimal sınıflandırıcıyı temsil etmesini engelliyorsa yüksek yanlılığa sahip sınıflandırıcı olarak adlandırılır. Böyle bir durumda modelin veriye “*eksik uyumundan (underfitting)*” söz edilir (Flach, 2012: 196).

Tüm öğrenme algoritmalarında, sınıflandırıcı karmaşıklığı ile veri karmaşıklığını eşleştirmeyi amaçlayan mekanizmalar bulunmaktadır. Örneğin karar ağaçları yönteminde aşırı uyum riskini azaltmak amacıyla ağacın fazla büyümesini engelleyen “*budama (pruning)*” prosedürleri vardır. Yapay sinir ağları, destek vektör makineleri gibi yöntemlerde ise modeldeki büyük ağırlıkları cezalandıran “*düzenleştirme (regularization)*” yöntemleri kullanılır. Bu tür yaklaşımlar, sınıflandırıcının karmaşıklığını sınırlandırmanın matematiksel eşdeğeridirler (Dietterich, 2003: 7).

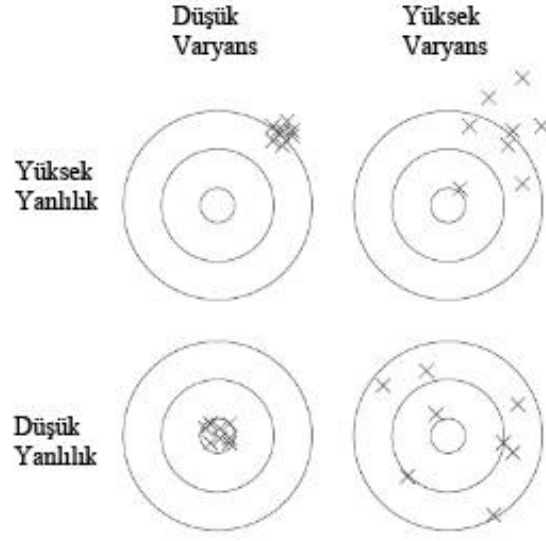
1.2.11.1.3 Yanlılık - Varyans (Bias – Variance) Ödünleşimi

Yanlılık ve varyans arasındaki ödünleşim şu şekilde görülebilir: Bir öğrenme algoritması birçok olası sınıflandırıcı arasında bir seçim yapmakla yüz yüze kalır.

Verinin çok az olduđu durumlarda olası sınıflandırıcıları birbirinden ayırabilecek bilgiye sahip değildir. Birçok sınıflandırıcı eğitim verisi üzerinde aynı doğruluk düzeyine sahip olacaktır. Öğrenme algoritması, içlerinden birisini rastgele seçecek olursa yüksek varyans oluşacaktır. Bu nedenle sınıflandırıcıları birbirinden ayırt edebilecek yeterli veriye sahip olana kadar sınıflandırıcıların karmaşıklıkları azaltılarak olası sınıflandırıcıların sayısı düşürülmelidir. Fakat bu yaklaşım varyansı düşürürken yanlılığın yükselmesine sebep olur.

Küçük veri kümelerinde eğitim yapılacaksa basit modellerin karmaşık modellere göre daha az oynaklık göstermesi beklenir yani basit modellerin varyansı düşüktür. Diğer taraftan, basit bir model daha çok varsayım yapar, daha az esnektir ve öğrenilecek sınıf gerçekte basit değilse yetersiz kalabilir. Basit bir modelin bunun için yanlılığı daha fazladır. En uygun modeli bulmak, yanlılık ve varyansın birlikte olabildiğince azaltılmasını gerektirir (Bishop, 2006: 149).

(Şekil 3) yanlılık–varyans arasındaki ödünleşimi hedef tahtası örneği üzerinde göstermektedir. En yüksek puan, okların büyük çoğunluğunun hedef tahtasının ortasında toplandığı durumda elde edilir. Bu örnekten yola çıkarak *“iyi bir sınıflandırıcının yanlılığı ve varyansı mümkün olduğunca düşük olmalıdır”* sonucuna varılabilir.



Şekil 3. Yanlılık-Varyans İkilemi

Aşırı öğrenmeden (varyans) kaçınırken az öğrenme (yanlılık) hatasına düşmek kolaydır. Her iki hatadan kaçınmak için mükemmel bir sınıflandırıcının öğrenilmesi gerekir. Şu anda her iki hatayı da aynı anda düşüren ve en iyi sonucu veren bir yöntem yoktur (Domingos, 2012: 81).

1.2.11.2 “No Free Lunch” Teoremi

Genellemenin asıl amaç olmasının temel bir sonucu da ne kadar çok veriye sahip olunursa olunsun sadece verinin yeterli olmayacağıdır. Örneğin bir milyon gözlem üzerinde 100 değişkene sahip olan bir boolean (ikili) fonksiyon öğrenilmeye çalışıldığı varsayıldığında, sınıfı bilinmeyen daha ($2^{100} - 10^6$) gözlem olacaktır. Bu gözlemlerin sınıflarının belirlenmeye çalışılması ek bilgi (varsayım) olmadan yazı tura atmaktan daha iyi sonuç vermeyecektir. Bu nedenle her öğrenme makinesi verilen bilginin ötesinde bilgiye veya varsayıma sahip olmalıdır (Murphy, 2012: 24).

Bu durum David Wolpert tarafından “No Free Lunch Teoremi” adıyla incelenmiştir. Bu teoreme göre; bir öğrenme makinesi öğrenilecek olası tüm fonksiyonlar üzerinde rastgele tahminden daha iyi olamaz. Bu durumda öğrenme mümkün değil gibi görünse de gerçek dünyada öğrenilmek istenilen fonksiyonlar olası tüm matematiksel fonksiyonlar kümesinden uniform (birörnek) çekilmediği için

bu durum bir problem oluşturmaz. Gerçekte, düzgünlük, benzer gözlemlerin benzer sınıflara sahip olması, sınırlı karmaşıklık gibi çok genel varsayımlar karşılaşılan problemlerin çözümünde işe yaramaktadır. Makine öğrenmesi'nin başarılı olma nedenlerinden birisi de budur (Domingos, 2012: 80).

1.2.11.3 Boyut Belası

Makine öğrenmesinde aşırı öğrenmeden sonra karşılaşılan en büyük sorun "*boyut belası (curse of dimensionality)*" problemidir. Boyut belası kavramı, ilk defa 1961 yılında Richard E. Bellman tarafından düşük boyutlarda iyi çalışan algoritmaların girdiler yüksek boyutlu olduğunda iyi çalışmamaya başlamasını ifade etmek için kullanılmıştır (Duda & Hart, 2000: 15).

Boyut arttıkça doğru genellemeyi sağlamak üssel olarak zorlaşmaktadır. Çünkü sabit büyüklükteki eğitim kümesi yüksek boyutlu girdi uzayını düşük bir oranda kapsayabilmektedir. Örneğin, özellik sayısının 100 ve eğitim kümesinin 1 trilyon gözleme sahip olduğu bir durumda bile girdi uzayı sadece 10^{-18} oranında kapsanabilmektedir. Bu durum makine öğrenmesi'ni hem gerekli hem de zor hale getiren temel unsurlardan birisidir.

Modelde daha fazla özelliğin yer almasının, en kötü ihtimalle öğrenilecek sınıf hakkında yeni bir bilgi sağlamayacakları düşünülerek, modelin performansına bir zarar vermeyeceği düşünülebilir. Fakat bazen yeni özelliklerin kullanılması boyut belası nedeniyle problemin çözümünü zorlaştırabilir (Domingos, 2012: 82).

Birçok uygulamada gözlemler üniform olarak dağılmazlar. Genellikle daha düşük boyutlardaki bölgelerde toplanırlar. Örneğin el yazısı rakamlarını tanıma problemi, her rakamın bir pikseli bir boyutla gösterildiğinden, yüksek boyutlu bir problem olsa bile bazı algoritmalar tarafından başarılı bir şekilde çözülebilmektedir. Bunun nedeni, bu örnekteki rakam görüntülerinin olası tüm görüntülerin uzayından çok küçük olmasıdır. Öğrenme makineleri bu daha düşük etkin boyutun avantajını kullanırlar. Fakat açık bir şekilde boyut azaltma tekniklerinin kullanılması gereken durumlar da olabilir (Harrington, 2012: 270).

1.2.11.4 Teorik Garantiler

Tümevarım ile tümdengelimden geleneksel olarak birbirinin zıttı olduğu düşünülür. Tümdengelimde öncüller doğru ise sonuçların doğruluğu garantidir. Tümevarım için ise böyle bir durumun geçerli olmadığı düşünülürdü. Makine öğrenmesi'ndeki önemli gelişmelerden birisi de tümevarım sonuçlarının da olasılıksal olarak garantilenebileceğinin fark edilmesi olmuştur.

Fakat bu teorik garanti sınırları doğru hipotezin nasıl seçileceği konusunda herhangi bir bilgi vermez. Bu sınırlar yoluyla sadece, hipotez uzayı doğru hipotezi içerdiğinde öğrenme makinesinin kötü bir sınıflandırıcı oluşturma olasılığının, eğitim kümesi büyüklüğü arttıkça azalacağı garantilenmektedir (Nasien vd., 2010: 762). Hipotez uzayının daraltılmasıyla daha sıkı teorik sınırlar elde edilebilmesine karşın, gerçek sınıflandırıcının hipotez uzayında olma şansı da azaltılmış olur.

Makine öğrenmesinde teorik garantilerin temel rolü, uygulamalar için bir ölçüt olmaktan çok, algoritmaları anlamak ve yeni algoritmalar geliştirmek için bir kaynak olmasıdır (Blum, 2007: 2).

1.2.11.5 Occam'ın Usturası

Makine öğrenmesi'nde Occam'ın usturası (Occam's razor) prensibi aynı eğitim hatasına sahip iki sınıflandırıcı verildiğinde, basit olan sınıflandırıcının daha düşük test hatası vereceği, dolayısıyla daha iyi genelleme yapabileceğini anlatmak için kullanılır (Lazar, 2010: 243). Girdilerde gürültü varsa ve gerçek sınıf zaten basit bir modelse, karmaşık bir modele göre öğrenme kümesinde biraz daha fazla hata yapsa da, daha küçük varyanslı ve gürültülü örneklerden daha az etkilenecek basit bir model daha iyi bir seçimdir. Eğitim hatalarının aşağı yukarı aynı olduğu durumlarda basit bir modelin karmaşık bir modelden daha iyi genelleme yaptığı söylenebilir. Occam'ın usturası ilkesi, basit açıklamaların daha inandırıcı olduğunu savunur ve gereksiz karmaşıklığın kesilip atılması gerektiğini belirtir.

Literatürde bu iddiayı destekleyen ispatların olduğu belirtilse de gerçekte birçok karşıt örnek de bulmak mümkündür. Karşıt örneklerden birisi "*birleştirilmiş*

(ensemble)” modellerdir. Birleştirilmiş bir modele eğitim hatası sıfıra düştükten sonra yeni sınıflandırıcılar eklendiğinde bile genelleme hatası düşmeye devam edebilmektedir. Diğer bir karşıt örnek de aşırı öğrenmeye düşmeden sonsuz sayıda parametreye sahip olabilen destek vektör makineleri yöntemidir. Bunun tersine tek parametreye sahip olup da keyfi büyüklükte ve keyfi olarak etiketlemiş gözlemleri ayırabilen fonksiyonlar da bulmak mümkündür. Bu nedenle önsezilerin tersine bir modelin parametre sayısı ile aşırı öğrenme eğilimi arasında zorunlu bir bağlantı yoktur (Domingos, 2012: 84).

1.2.11.6 Bağımlılık ve Nedensellik

Öğrenme süreciyle ilişkili olan iki kavram arasında açık bir ayırım yapmaya ihtiyaç vardır. Bunlar istatistiksel *bağımlılık (dependency)* ve *nedensellik (causality)* kavramlarıdır. Girdi kümesi ile çıktı kümesi arasındaki istatistiksel bağımlılık öğrenme yönteminin bulunduğu yaklaşan fonksiyon ile ifade edilir. Buradaki temel husus nedenselliğin tek başına, veri analizi sonucu öğrenilen bir model kullanılarak çıkarılamayacağıdır. Bunun yerine nedenselliğin öğrenme sürecinin dışında kanıtlarla kabul edilmesi veya gösterilmesi gerekir. Öğrenme süreci sonunda bağımlılıkları gösteren modeller oluşturulabilir fakat bu bağımlılıkların otomatik olarak nedensellik ilişkisinde bulduklarını yorumlamak yanlıştır. Sadece verinin toplandığı alanın uzmanları, keşfedilen bağımlılıklar konusunda daha fazla ve derin bir nedensellik yorumu önerebilirler (Kantardzic, 2011: 91).

1.2.12 Model Geçerleme

Bulunan bir yaklaşan fonksiyonun performansını ölçmenin en kestirme yolu bu fonksiyonu bir test kümesi üzerinde test etmektir. Fakat en iyi performansı gösteren modelin hangisi olduğunu öğrenmek için test kümesi üzerinde farklı öğrenme sistemlerinin karşılaştırılması test kümesi üzerinde eğitim yapmakla aynı anlama gelir. Bu durum eğitim kümesini genişletmesine ve genelleme performansında artışa yol açmasına rağmen aşırı öğrenme tehlikesi hala devam eder. Bunu engellemek için önerilen bir yaklaşım veri kümesini eğitim, geçerleme ve test kümesi olarak ayırmaktır (Hsieh, 2009: 137-38).

Eđitim kümesi üzerinde modeller oluşturulur. Geçerleme kümesi bu modellerin içinden en iyisini seçmek için kullanılırken; seçilen modelin yeni veriler üzerindeki performansı test kümesi üzerinde değerlendirilir (Linoff, 2004: 52).

Bir öğretmen ders anlatırken sınıfta çözdüğü örnek sorular eğitim kümesine, sınav soruları geçerleme kümesine ve daha sonra öğrencinin meslek hayatında karşılaştığı sorular da test kümesine benzetilebilir. Öğrenme sürecinde sürekli aynı eğitim ve geçerleme kümeleri kullanılamaz, çünkü bir kere kullanıldığında geçerleme kümesi eğitim kümesinin bir parçası olur. Bu durum, sınavlarında her yıl aynı soruları soran öğretmenin durumuna benzetilebilir. Kurnaz bir öğrenci konuları öğrenmeye çalışmayıp bu soruların cevaplarını ezberleyecektir. Öğrenme algoritmaları da dikkat edilmezse asıl hedef olan genelleme yerine verileri ezberleme yoluna gidebilirler (Alpaydın, 2011: 31).

Eđitim, geçerlilik ve test kümeleri için ayrılan veri miktarını etkileyen çeşitli durumlar vardır. Örneğin, geçerlilik kümesi için ayrılan verinin miktarı algoritmadaki serbest parametre sayısına bağlıdır. Eğer gözlem sayısı az ise eğitim kümesinin oranı daha büyük seçilir (Mohri vd., 2012: 4).

Verilerin bu şekilde kümelere ayrılması sonucunda eğitim kümesinin büyüklüğü azalır ve aşırı öğrenme olasılığı artar. Bu problemden kurtulmak için çeşitli geçerleme yöntemleri kullanılmaktadır (Nilsson, 1998: 81). Bunlar dışarıda tutma, çapraz geçerleme, tabakalı çapraz geçerleme, birini dışarıda bırakma, yerine koyarak rastgele örnekleme yöntemleridir.

1.2.12.1 Dışarıda Tutma (Holdout)

Bu yaklaşımda veri kümesi rastgele şekilde birbirinden bağımsız olarak eğitim ve test kümesi olarak ayrılır. Genellikle 2/3 oranında veri eğitim için, 1/3 oranında ise test için kullanılır. Model oluşturmak için eğitim kümesi kullanılırken, oluşturulan modelin doğruluğu test kümesinde tahmin edilir.

1.2.12.2 Çapraz Geçerleme (Cross Validation)

Çapraz geçerlemede eğitim kümesi eşit büyüklükteki birbiriyle kesişmeyen altkümelere ayrılır. Her bir altküme için eğitim geri kalan alt kümelerin birleşim kümesinde yapılır. Test performansı ise ilgili altküme üzerinde hesaplanır. Bir sınıflandırıcının yeni örnekler üzerindeki beklenen hata oranı bu altkümelerdeki hata oranlarının ortalaması alınarak bulunur.

Örneğin, k -katlı çapraz geçerlemede başlangıç verisi D_1, D_2, \dots, D_k şeklinde birbirinden bağımsız, aynı büyüklükteki k adet alt kümeye ayrılır. i . çevrimde D_i altkümesi test kümesi olarak ayrılır ve eğitim geri kalan alt kümeler kullanılarak yapılır. Her bir altküme aynı sayıda eğitim için ve bir defa da test için kullanılır (Clarke vd., 2009: 27).

1.2.12.3 Tabakalı Çapraz Geçerleme (Stratified Cross Validation)

Her bir kattaki gözlemlerin dağılımının başlangıçtaki verideki dağılımla aynı olması amaçlanır. Genellikle 10 kat tabakalı çapraz geçerleme yüksek doğruluk oranına ulaşmak için tercih edilir (Han vd., 2011: 371).

1.2.12.4 Birini Dışarıda Bırakma (Leave One Out)

Birini dışarıda bırakma geçerlemesi, k altküme sayısının, eğitim kümesindeki gözlem sayısına eşit olması özel durumu haricinde çapraz geçerleme ile aynıdır. Her bir altküme sadece bir gözlem içermektedir. Yani bir altküme test kümesi olarak kullanılırken bu kümede yer alan gözlemin doğru sınıflandırılıp sınıflandırılmadığına bakılır. Tüm altkümeler test kümesi olarak kullanıldıktan sonra oluşan toplam hata k sayısına bölünerek hata oranının bir tahmini bulunur (Nilsson, 1998: 82).

Birini dışarıda bırakma yöntemi neredeyse yansız olmasına rağmen yüksek varyansa sahip olduğundan güvensiz tahminler yapabilir. Ayrıca büyük veri kümelerinde kullanıldığında hesaplama maliyeti çok yüksektir. Model geçerleme için bu yöntemi kullanan modeller asimptotik olarak tutarsızdır. Çünkü gözlem sayısı sonsuza yaklaştığında en yüksek öngörü gücüne sahip modeli seçme olasılığı 1'e yakınsamaz (Olson & Delen, 2008: 147).

1.2.12.5 Yerine Koyarak Rastgele Örnekleme (Bootstrap)

Çok küçük veri kümelerinde kullanılır. Tek bir veri kümesinden birden çok veri kümesi oluşturulur. Veriden rastgele bir gözlem seçilir, kaydedilir ve bu gözlem geri konulur. Böylece aynı gözlem birden çok seçilebildiği gibi bazıları ise hiç seçilmeyebilir (Maloof, 2005: 33).

1.2.13 Model Karmaşıklığını Düzenleme

Optimal model karmaşıklığının (model complexity) belirlenmesi için kullanılan çeşitli yaklaşımlar bulunmaktadır. Bu yaklaşımlar önsel bilginin temsili, gerçek modelin yaklaşan fonksiyonlar kümesinde olmadığı durumlardaki uygulanabilirliği, önsel bilgiyi eğitim verisine dahil etme mekanizması ve öğrenme algoritmalarının bulunup bulunmamasına göre farklılıklar gösterirler. Bu yaklaşımların sınırlılıkları ve görece olarak birbirlerine üstünlükleri üzerinde belirgin bir karışıklık vardır. Bu durumun sebepleri arasında her birinin kendine özgü terminolojilerinin olması ve aralarında anlamlı karşılaştırmalar yapabilmeyenin mümkün olmamasıdır. Yaygın olarak kullanılan karmaşıklık düzenleme yaklaşımları düzenlileştirme, erken durdurma, yapısal risk minimizasyonu, bayesyen çıkarsama ve minimum tanım uzunluğu olarak sıralanabilir.

1.2.13.1 Düzenlileştirme (Regularization)

Çözümleri kısıtlamak için, eğitim hatasına bir ceza terimi eklenir. Bu ceza terimi, olası her bir $f(x, \omega)$ fonksiyon kümesi ile ilişkilendirilen negatif olmayan bir fonksiyondur. Cezanın eğitim hatasına göre önemini gösteren bir düzenlileştirme parametresi belirlenir. Ceza terimi eğitim verisinden bağımsızdır. Bu çerçevede önsel bilgi ceza terimi şeklinde modele dahil edilir ve böyle bir bilginin etkisi düzenlileştirme parametresi yoluyla kontrol edilir. Yaygın olarak kullanılan birçok yaklaşan fonksiyon kümesi için karmaşıklığı ölçen ceza fonksiyonelleri geliştirmek mümkündür. Optimal (en küçük öngörü riskini sağlayan) düzenlileştirme parametresi değeri genellikle yeniden örnekleme (resampling) metotları kullanılarak seçilir. Bu yaklaşımda optimal model tahmini, veriye uyum ve önsel bilgi (ceza terimi) arasındaki ödünleşimin bir sonucu olarak bulunur (Hsieh, 2009: 135).

1.2.13.2 Erken Durdurma (Early Stopping)

Yapay sinir ağırları'nda hata minimize edilirken aşırı uyumdan kaçınmak için erken durdurma kullanılır (Demuth vd., 1992: 162). Erken durdurma işlemi dolaylı yoldan bir cezalandırma biçimi olarak görülebilir. Eğimli iniş (gradient descent) yöntemiyle bulunan çözümler, parametre uzayında başlangıç noktasından itibaren eğimli iniş adımlarına göre cezalandırılır. Bu tür bir cezalandırma tekniği rastgele seçilen başlangıç durumlarına, eğitim verisine ve kullanılan optimizasyon tekniğine aşırı derecede bağımlıdır. Ek olarak erken durdurma kuralları yoluyla yapılan cezalandırmanın kontrolü ve yorumlanması zordur (Cherkassky & Mulier, 2007: 46).

1.2.13.3 Yapısal Risk Minimizasyonu (Structural Risk Minimization)

Öğrenme makinesinin uygulayabildiği yaklaşan fonksiyonlardan karmaşıklıklarına göre sıralanmış iç içe bir yapı oluşturulur. Örneğin, polinomsal yaklaşan fonksiyonlar sınıfında bir yapının elemanları verilen derecedeki polinomlardır. m dereceli polinomlar $m + 1$ dereceli polinomların alt kümesi olduğundan iç içe yapı oluşturma şartı sağlanır. Öğrenmenin amacı, bir yapının optimal elemanını seçmek ve verilen eğitim kümesini kullanarak bu elemanın parametrelerini tahmin etmektir. Karmaşıklık ölçütü olarak, polinomlar gibi parametrelerinde doğrusal olan fonksiyonlar için serbest parametre sayısı, parametrelerinde doğrusal olmayan fonksiyonlar için ise “*VC boyutu*” kullanılır (Shin vd., 2005: 131). Beklenen riskin üst sınırını minimize etmek amaçlanır. Bu yaklaşımın uygulanabilmesi için veriler görülmeden önce hipotez uzayının tanımlanması gerekir.

1.2.13.4 Bayesyen Çıkarsama (Bayesian Inference)

Bayesyen çıkarsama, sınırlı eğitim verisinden tek bir öngörücü model oluşturabilmek için yaklaşan fonksiyonlar hakkında ek bilgi kullanır. Bu bilgi, yaklaşan fonksiyonlar kümesinden herhangi bir fonksiyonun bilinmeyen gerçek dayanak fonksiyonu olma olasılığını belirten önsel olasılık dağılımı şeklindedir. Bulunacak model daha çok önsellerin iyi seçilmesine bağlı olduğundan öğrenme makinesinin tasarımına öznellik katmaktadır (Cherkassky & Mulier, 2007: 47).

1.2.13.5 Minimum Tanım Uzunluđu (Minimum Description Length)

Occam'ın usturası ilkesinin bilinen en iyi uygulaması, rastgelelik kavramının bilgi teorisi tabanlı analizine dayanan minimum tanım uzunluđu yaklaşımdır. Bahsi geçen diđer yaklaşımların aksine bu yaklaşımda modeller eğitim verisini kodlayan (encoding) kodlar olarak ele alınır. Herhangi bir veri kümesinin dođru bir şekilde kodlanabileceđi ve kod uzunluđunun da verinin dođal bir özelliđi olduđu fikrine dayanır. Buna göre en kısa tanım uzunluđuna sahip model, bilinmeyen bađımlılıđın dođru temsilini, dolayısıyla da minimum beklenen riski sađlar (Forster, 2000: 215).

İKİNCİ BÖLÜM

2. YAPAY SİNİR AĞLARI

Bu bölümde yapay sinir ağlarının yapısı, temel elemanları, mimarileri açıklanmıştır. Tek katmanlı algılayıcı, delta ve geri yayılım öğrenme algoritmaları incelenerek öğrenmenin nasıl gerçekleştiği incelenmiştir. Bölümün sonunda yapay sinir ağı uygulamalarında dikkat edilmesi gereken durumlar hakkında açıklamalar yer almaktadır.

2.1 Yapay Sinir Ağları Hakkında Temel Bilgiler

Sınıflandırma, regresyon, kümeleme vb. problem tipleri için birçok yöntem geliştirilmiştir. Bu yöntemlerin bazıları varsayımların sağlandığı durumlarda başarılı sonuçlar vermekle birlikte, tasarlandıkları eğitim kümesinin haricindeki gözlemler için iyi performans gösterecek kadar esnek değildirler. Yapay Sinir Ağları (YSA) klasik yöntemlere alternatif bir yaklaşım olarak kullanılmaktadır (Jain vd., 1996: 1).

Son yıllarda YSA, pek çok uygulama alanında ilgi çekmektedir. YSA'ların en fazla örüntü tanıma (sınıflandırma) uygulamalarında kullanıldığı görülmektedir. Özellikle bilgisayarların birim zamanda yapabildiği işlem sayısının fazlasıyla artmış olması YSA'ların daha etkin kullanılabilmesini sağlamıştır.

1980'li yılların ortalarından başlayarak fizik, istatistik, psikoloji, nöroloji, bilişsel bilim, dilbilim, bilgisayar bilimleri, elektronik ve robotik gibi çok farklı alanlarda YSA yaklaşımına olan ilgi artmıştır. Yapay sinir ağlarının belki de en büyük katkısı farklı alanları, özellikle istatistik ve mühendisliği, bir araya getirebilmesidir. Bugün makine öğrenmesi denilen alan bu birliktelikten ortaya çıkmıştır (Alpaydın, 2011: 231).

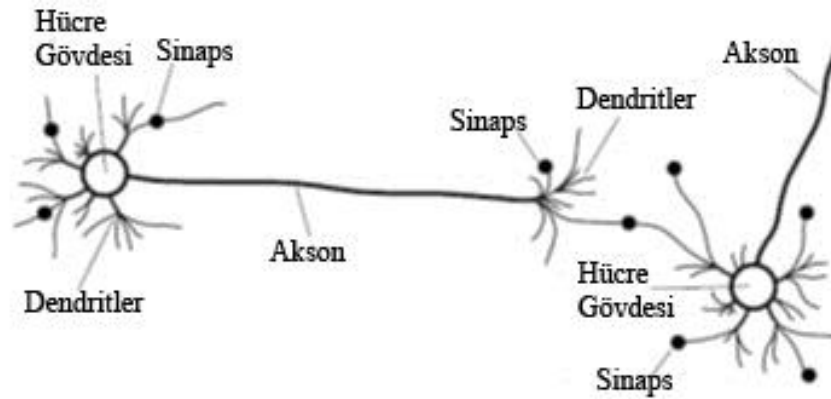
YSA, beynin çalışma ilkelerinin bilgisayarlar üzerinde taklit edilmesi fikrinden ortaya çıkmıştır. Alandaki ilk çalışmalar beyni oluşturan "*biyolojik sinir*

hücrelerinin (nöron)” matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır. Bu çalışmaların ortaya çıkardığı bulgular, her bir nöronun komşu nöronlardan bazı bilgiler aldığı ve bu bilgilerin biyolojik nöron dinamiğinin öngördüğü biçimde bir çıktıya dönüştürüldüğü şeklinde idi (Efe & Kaynak, 2000: 1).

Biyolojik sinir sisteminden esinlenilerek ortaya atılmış olan YSA modelinin en önemli özelliği sunduğu bilgi işleme şeklidir. Yapay sinir ağını, “*herhangi bir problemi çözmek için birbirleri ile etkileşim içerisinde olan çok sayıdaki bilgi işlem biriminin birlikte uyumlu şekilde çalışmasından oluşan bir bütün*” olarak algılamak mümkündür.

2.1.1 Biyolojik Sinir Ağları

Beynin bilgiyi işleme süreci halen tam olarak bilinmemekle birlikte, bu süreci açıklamaya çalışan çeşitli teoriler bulunmaktadır. Buna göre, insan beyinde her bir nöron kendisine komşu olan nöronlardan dendritler vasıtasıyla sinyalleri toplar ve binlerce dala bölünen aksonlar üzerinden diğer nöronlara elektrik akımı gönderir (Şekil 4). Her bir dalın sonunda sinaps olarak adlandırılan bir yapı aksondaki canlılığı elektriksel etkilere dönüştürür ve böylece bağlı nöronlara canlılığı “*harekete geçirici (excitatory)*” veya “*tutucu (inhibitory)*” etkiler gönderir (Gorunescu, 2011: 192-193). Her bir nöron diğer nöronlara sinyal gönderilip gönderilmeyeceğine karar vermek gibi çok basit bir hesaplamayı yapan bir işlemci gibi görülebilir.

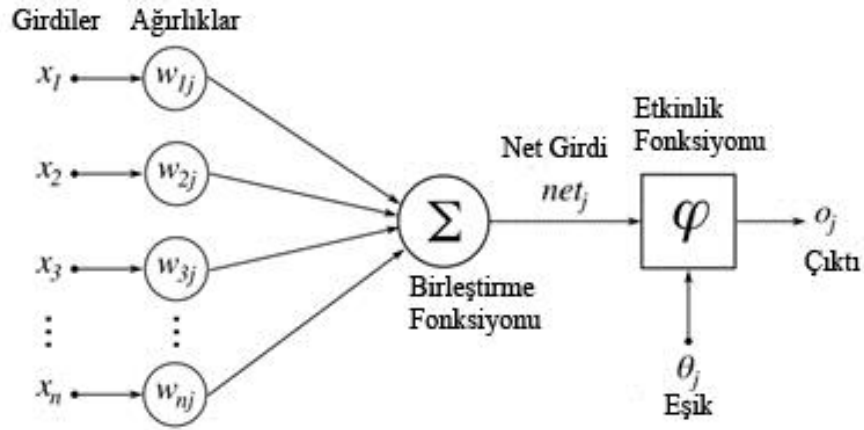


Şekil 4. Biyolojik Sinir Hücresi

Bir nöron, tutucu girdiden daha fazla harekete geçirici girdi aldığı anda aksonu üzerinden elektriksel bir canlılık gönderir. Öğrenme, sinapsların etkilerinin değiştirilmesiyle başka bir ifadeyle bir nöronun başka bir nöron üzerindeki etkisinin değiştirilmesiyle olur (Cura, 2008: 20).

Beyindeki karmaşık öğrenme sisteminin birbiriyle sıkı bir şekilde bağlantılı nöron kümeleri içerdiğinin fark edilmesi YSA için ilham kaynağı olmuştur. Yapısal olarak bir nöronun basit olmasına karşılık birbirine yoğun bir şekilde bağlı olan nöronlar karmaşık öğrenme görevlerini yerine getirebilirler. Örneğin insan beyni yaklaşık olarak 10^{11} nöron içermekte, her bir nöron diğer nöronlarla ortalama 10.000 bağlantı yaptığından, nöronlar arasında yaklaşık 10^{15} bağlantı bulunmaktadır. Bu nedenle beyin, 10^{11} işlemciden oluşan devasa büyüklükteki paralel bir bilgisayara benzetilebilir (Larose, 2005: 128).

(Şekil 5) yaygın olarak kullanılan yapay sinir modelini göstermektedir.



Şekil 5. Yapay Sinir Hücresi

Genel olarak bir YSA, ağırlıklandırılmış bağlantılarla birbirine bağlı çok sayıda basit bilgi işlem biriminden meydana gelmektedir. Benzetme yapılacak olursa, bilgi işlem birimleri beyindeki nöronlara benzetilebilir. Her bir bilgi işlem birimi birçok başka birimden girdileri alır ve bir çıktı oluşturur. Çıktı ağıdaki diğer birimlere girdi olacak şekilde dağıtılır (Reed & Marks, 1999: 1).

(Tablo 2) de biyolojik sinir modelinin yapay sinir modelindeki karşılıkları görülmektedir.

Tablo 2. Biyolojik Sinir Modeli İle Yapay Sinir Modeli Arasındaki Benzerlikler

Biyolojik Sinir Modeli	Yapay Sinir Modeli
Sinapslar	Bağlantı Ağırlıkları
Aksonlar	Çıktılar
Dendritler	Girdiler
Hücre Gövdesi	Birleştirme Fonksiyonu

Bilgi işlem birimleri bir araya gelerek yapay sinir ağlarını oluştururlar. Bu birimlerin bir araya gelmesi rastgele olmaz. Genel olarak girdi, gizli ve çıktı katmanı olmak üzere üç katman halinde ve her katman içinde paralel şekilde bir araya gelerek ağı oluştururlar.

Beyinde bir nörondan diğer nörona gönderilen bilgi çok küçük boyuttadır. Bu, beyinde kritik bilgilerin doğrudan taşınmadığını, nöronlar arasındaki ara bağlantılarda tutulduğunu ve dağıtıldığını göstermektedir. YSA'nın bir başka ismi olan "*bağlantıcı (connectionist) model*" ismi bu özelliğinden gelmektedir (Mehrotra vd., 1997: 1).

"*Esneklik (plasticity)*", sinir hücreleri arasındaki sinaptik bağlantıların değiştirilmesi veya yeni bağlantıların oluşturulmasıdır (Negnevitsky, 2004: 166). Beyindeki öğrenmede temel kavram bu esnekliktir. Beyinde sinapsların ağırlıklarının nasıl değiştirildiği tüm yönleriyle henüz bilinmemektedir.

Yapay sinir ağları ile biyolojik sinir ağları arasındaki etkileşim kısaca aşağıdaki gibi özetlenebilir (Jain vd., 1996: 6).

- YSA yaklaşımı biyolojik sinir ağlarından esinlenilerek geliştirilmiştir.
- Birbirine yoğun bir şekilde bağlı basit bilgi işleme birimlerine sahip bir ağ, çok güçlü bir hesaplama yeteneğine sahiptir.
- YSA'nın performans değerlendirmesi için biyolojik sinir ağları bir ölçüt olarak kullanılır.
- Biyolojik sinir ağları, zeki makineler oluşturabileceği düşüncesinin bir çeşit kanıtı olarak görülmektedir.

YSA, biyolojik sinir sistemlerinden ilham alınarak tasarlanmışsa da beyinde yer alan birçok karmaşık süreci modelleyemez. Bu nedenle YSA modellerinde yer alan özelliklerin birçoğu biyolojik sinir ağlarıyla birebir uyumlu değildir.

2.1.2 Biyolojik ve Yapay Sinir Ağları Karşılaştırması

Hesaplama yönünden bakıldığında, beyin tam olarak bir algoritmadan beklenildiği gibi çalışır. Gürültülü ve eksik verilerle çalışabilir ve çok yüksek boyutlu verilerden (örneğin görüntüler) hızlı bir şekilde ve genellikle doğru olan çıktılar oluşturabilir. Zamanla yaşlanmayla birlikte bazı sinir hücreleri yok olsa da performansı aniden azalmaz, bu yönüyle de sağlamdır (Marsland, 2009: 11).

Beyinde, yapılan işlem başına saniyede 10^{-16} Joule (J) enerji harcanırken, bir bilgisayarda bu değer yaklaşık olarak 10^{-6} (J) olmaktadır. Bu veriler, beynin enerji kullanımında, bilgisayara göre çok daha etkin olduğunu göstermektedir (Haykin, 1999: 28).

İnsan beyninin çalışma mekanizmasını taklit etmeye çalışan sistemler, günümüz teknolojisinin ürettiği birim işlem zamanı nanosaniyeler mertebesinde olan elektronik devreler ile gerçekleştirilebilirler de, insan beyninin birim işlem zamanı

milisaniyeler mertebesindeki nöronlarının birlikte çalışmalarıyla elde edilen işlevsellikten halen çok uzaktadırlar.

Deneyle, bir insanın daha önceden bildiği bir nesneyi algılamasının yaklaşık 10^{-1} saniye zaman aldığı göstermiştir. Beyindeki sinirlerin ortalama geçiş süresinin (switching time) 10^{-3} sn olduğu göz önünde tutulduğunda bu 100 adımlık bir paralel işlem adımına karşılık gelir. Klasik bilgisayar mimarisinde ise pratikte 100 seri adımda hiçbir şey yapılamaz (Kriesel, 2007: 5).

İnsan beyni sayısal işlemleri yapabilmek için belli bir zamana ihtiyaç duymasına karşın, idrak etmeye yönelik olayları çok kısa sürede yapabilir. Örneğin, yolda giden bir şoför yolun kayganlık derecesi, aradaki mesafe gibi değerleri hesaplayamasa da önceki tecrübelerine dayanarak tehlikeyi önceden sezip hızını azaltabilir. Bilgisayarlar ise çok karmaşık sayısal işlemleri çok hızlı yapabilmelerine rağmen idrak etme ve deneyimlerle kazanılmış bilgileri kullanabilme konusunda insana göre çok yetersizdirler. İnsan beynini üstün kılan temel özellik, sinirsel algılayıcılar vasıtası ile kazanılmış ve sınıflandırılmış bilgileri kullanabilmesidir (Elmas, 2007: 21).

Biyolojik bilgisayar olan beyin, modern bilgisayarlarda kullanılan Von Neumann mimarisinden tamamen farklı bir mimari kullanmaktadır (Jain vd., 1996: 9). Tablo 3 de bu farklılıklar özetlenmiştir.

Tablo 3. Bilgisayar Mimarisi ile Yapay Sinir Ağlarının Karşılaştırılması

	Bilgisayar	Yapay Sinir Ağları
İşlemci	Karmaşık Yüksek hızlı Bir veya daha fazla	Basit Düşük hızlı Çok sayıda
Hafıza	İşlemciden ayrı Yerleştirilmiş İçerik adreslenemez	İşlemciyle bütünleşik Dağıtık İçerik adreslenebilir
Hesaplama	Merkezi Seri Yazılımlar	Dağıtık Paralel Öğrenme
Güvenilirlik	Çok hassas	Hataya dirençli
Uzmanlık	Sayısal ve sembolik işlemler	Algısal problemler
İşletim Ortamı	İyi tanımlanmış İyi kısıtlanmış	Zayıf tanımlanmış Kısıtlanmamış

Bilgisayarlarda işlemci etkin, bellek edilgen iken beyinde işlemci ve belleğin ağ üzerinde dağıtılmış olduğu düşünülür. İşlemler ağdaki birimler tarafından yapılırken, bellek bağlantılardadır (Alpaydın, 2011: 197).

Yapay sinir ağları, karar hızı açısından insan beyni ile yarışabilecek aşamayı henüz kat edememiş olmalarına rağmen, karmaşık eşleştirmeleri hassas bir biçimde gerçekleştirebilmeleri ve yapısal sağlamlığa sahip olmaları nedeniyle gün geçtikçe uygulama alanları genişlemektedir (Efe & Kaynak, 2000: 1).

2.1.3 Yapay Sinir Ağlarının Tarihsel Gelişimi

Tarihsel olarak, YSA üzerinde çalışan iki farklı araştırmacı grubu bulunmaktadır. Bir grup biyolojik öğrenme sürecini modellemek için YSA çalışırken, diğer grupta yer alan araştırmacılar biyolojik sürece uygun olup olmadığına bakmaksızın yüksek derecede etkin makine öğrenmesi algoritmaları oluşturmak için çalışmalar yapmaktadırlar.

Yapay sinir ağlarının tarihçesi programlanabilir bilgisayarların tarihi ile hemen hemen aynı zamanda, yani 1940'ların ilk yıllarıyla başlar. Yapılan ilk çalışmalar nörobiyoloji konusuna ilgi duyan araştırmacıların elde ettikleri sonuçları bilgisayar bilimine uygulamaları ile başlamıştır. Çalışmalar, 1970 öncesi ve sonrası olmak üzere iki dönemde incelenebilir (Öztemel, 2006: 37-41).

2.1.3.1 1970 Öncesi Çalışmalar

1943: Warren McCulloch ve Walter Pitts, ilk sinir ağları modelini oluşturduklar. Basit ağların bile hemen hemen her mantıksal ve aritmetik fonksiyonu hesaplayabileceklerini gösterdiler.

1949: Donald O. Hebb tarafından günümüzde birçok öğrenme kuralının temelinde yer alan "*Hebb kuralı*" ortaya atıldı.

1950: Nöropsikolog Karl Lashley, beyindeki bilginin depolanmasının dağıtık (distributed) bir şekilde yapıldığı tezini öne sürmüştür.

1956: Beynin çalışmasının nasıl taklit edilebileceğine dair ilk tartışmalar Dartmouth Üniversitesi'ndeki toplantıda yapılmıştır.

1957 - 1958: Frank Rosenblatt ve Charles Wightman, "*Mark I Perceptron*" adındaki ilk sinirsel bilgisayarı (neurocomputer) geliştirdiler. Bu bilgisayar rakamları tanıyabilme yeteneğine sahipti.

1959: Frank Rosenblatt "*Perceptron Yakınsama Teoremini*" ispatladı.

1960: Bernard Widrow ve Marcian E. Hoff, ticari kullanıma sahip ilk sinir ağı olan "*ADALINE (ADaptive LInear NEuron)*" adındaki öğrenme sistemini tanıttılar.

1961: Karl Steinbruch sinirsel "*çağrışımlı bellek (associative memory)*" ile ilgili ilk çalışmaları yaptı.

1965: İlk makine öğrenmesi kitabı, Nils Nilson tarafından "*Learning Machines*" adıyla yayınlandı. Bu kitapta, sinir ağları araştırmaları hakkında genel bilgiler ve o zamana kadar olan gelişmeler yer almaktadır.

1969: Marvin Minsky ve Seymour Papert, perceptron modelinin birçok temel problemi (örneğin XOR problemi) çözme yeteneğinin olmadığını gösteren matematiksel ispatlarını yayınladılar. Bu gelişme, yapay sinir ağlarının popülerliğinin azalmasına ve araştırma fonlarının kesilmesine sebep oldu. Bu tarihten itibaren alandaki araştırmalar eski hızını kaybetti ve çok az sayıda araştırmacı çalışmalarına devam etti (Peretto, 1992: 10).

2.1.3.2 1970 Sonrası Çalışmalar

1972: Teuvo Kohonen ve James A. Anderson, birbirlerinden bağımsız olarak bir çağrışımlı bellek modeli olan “*doğrusal ilişkilendirici (linear associator)*” modelini buldular.

1973: Christoph von der Malsburg doğrusal olmayan (nonlinear) ve biyolojik modele daha yakın olan bir sinir modelini oluşturdu.

1974: Paul Verbos doktora tezi için “*hatanın geri yayılımı (error back propagation)*” adını verdiği bir öğrenme prosedürü geliştirdi.

1976: Stephen Grossberg değişik sinir modellerini matematiksel olarak analiz eden çok sayıda çalışma yaptı. Bir yapay sinir ağının daha önce öğrendiği ilişkilere zarar vermeden öğrenmeye devam edebilmesi problemi üzerine çalıştı.

1980: Stephen Grossberg ve Gail Carpenter tarafından “*Adaptif Rezonans Teorisi (ART)*” geliştirildi.

1982: Teuvo Kohonen, “*Kendini Düzenleyen Haritalar (Self Organizing Maps - SOM)*” konusundaki çalışmasını yayınladı.

1982: John Hopfield tarafından “*Hopfield Ağları*” geliştirildi.

1983: Miyake Fukushima ve Ito Fukushima el yazısı karakterlerini tanıyabilen “*Neocognitron*” modelini tanıttılar.

1984: Geoffrey E. Hinton ve Terry Sejnowski tarafından “*Boltzmann Makinesi*” geliştirilmiştir.

1985: John Hopfield, gezgin satıcı probleminin Hopfield ağıları kullanılarak kabul edilebilir çözümlerinin bulunabileceği konusundaki makalesini yayınladı.

1986: David E. Rumelhart ve James McClelland tarafından delta kuralının genelleştirilmiş hali olarak “geri yayılım algoritması” geliştirildi ve yayımlandı.

1988: David Broomhead ve David Lowe çok katmanlı algılayıcılara alternatif olarak “Dairesel Tabanlı Fonksiyon (Radial Basis Functions - RBF)” ağılarını geliştirdiler.

1988: Donald Specht tarafından “Olasılıksal Ağ (Probabilistic Neural Networks)” modeli geliştirildi.

1991: Donald Specht “Genel Regresyon Ağları (General Regression Neural Networks)” modelini geliştirdi.

1991’den günümüze ise sayısız çalışma ve uygulama yapılmıştır. Bunların listesini burada vermek nerede ise imkânsızdır.

2.1.4 Yapay Sinir Ağlarının Kullanım Alanları

YSA uygulamaları hayatın hemen hemen her alanında görülmektedir. Genel olarak sınıflandırma, tahmin, tanıma, kontrol, teşhis problemlerinde başarıyla kullanılmaktadır. Değişik alanlardaki YSA uygulama türlerinden bazıları Tablo 4 de gösterilmiştir (Tarassenko, 1998: 52).

Tablo 4. Yapay Sinir Ağları Uygulamalarına Örnekler

	Hata Teşhisi	Durum İzleme	Tahmin	Sinyal/Görüntü Analizi	Örüntü Tanıma	Endüstriyel Denetleme	Sahtekârlık Tespiti	Süreç Kontrol
Parça İmalat	✓	✓				✓		
Süreç İmalat		✓					✓	✓
Perakende			✓		✓		✓	
Finans - Sigorta			✓		✓		✓	
Mühendislik	✓	✓		✓				✓
Üretim Kontrol	✓		✓					✓
Hizmet			✓		✓			
Maliye - Hazine			✓				✓	
Satış - Pazarlama			✓		✓			

Daha spesifik olarak, yapay sinir ağlarının finansal alanda kullanımına iflas tahmini, kredi kartı uygulamaları, ipoteğe dayalı krediler, tahvil derecelendirmesi, endeks tahmini, banka iflaslarının tahmini, hisse senedi seçimi, opsiyon fiyatlaması, sermaye piyasası analizleri, ekonomik gösterge tahmini, yatırım fonu seçimi, nakit akımlarının tahmini, döviz kuru tahmini, arbitraj fiyatlama modeli uygulamaları örnek olarak verilebilir (Diler, 2003: 73).

2.1.5 Yapay Sinir Ağlarının Güçlü Yönleri

YSA'nın belli başlı avantajları aşağıdaki gibi sıralanabilir.

2.1.5.1 Öğrenme

Yapay sinir ağları, güçlü bir öğrenme yeteneğine sahiptir. Ağ parametrelerini değiştirerek çalıştıkları ortama uyum sağlayabilirler. Bu yeteneklerini mümkün kılan

ise güçlü öğrenme algoritmalarıdır. Öğrenme, yapay sinir ağlarının en göze çarpan özelliğidir.

Girdi örnekleri ve bunlara karşılık beklenen çıktılar verildiğinde, her bir eğitim girdisi için doğru çıktıyı oluşturacak şekilde, ağırlıkları değiştiren algoritmalar mevcuttur. Bu konuda geri yayılım (backpropagation) algoritması en yaygın olarak kullanılan algoritmadır. Pratikte eğitimin her zaman başarılı olamaması gibi zorluklar olsa da, teorik olarak bir veri kümesine uyacak bir ağ her zaman bulunabilir (Reed & Marks, 1999: 4).

2.1.5.2 Genelleme

Doğru bir eğitimin sonucunda ağ sadece eğitim gözlemlerini öğrenmekle kalmaz, aynı zamanda eğitim kümesinde yer almayan gözlemler verildiğinde doğru çıktıyı oluşturacak şekilde girdiler ile çıktılar arasındaki ilişkiyi öğrenir. Kısaca, iyi eğitilmiş bir YSA iyi bir genelleme gösterebilir (Du & Swamy, 2006: 16).

2.1.5.3 Doğrusal Olmama

Bir YSA birimi, tipik olarak doğrusal olmama özelliğine sahiptir. YSA “*evrensel yaklaşma (universal approximating)*” yeteneğini kullanarak problemleri çözmeyi amaçlar. Problem hakkında detaylı bilgi sahibi olunması veya problemin yapısının analiz edilmesi gibi gereksinimler ortadan kalkmaktadır (Warren, 1994: 5).

2.1.5.4 Kendi Kendini Organize Etme

Bazı yapay sinir ağları (örneğin kendini düzenleyen haritalar ve yarışmacı öğrenme tabanlı ağlar) kendi kendini organize etme özelliğine sahiptirler. Bu ağların eğitimi denetimsiz öğrenme algoritmaları kullanılarak başarılr (Zurada, 1992: 58).

2.1.5.5 Paralel Çalışma

Yapay sinir ağları, paralel bilgi işleme yapısına sahiptirler. Paralel yapı, YSA'nın yüksek derecede paralel çalışan yazılım ve donanım uygulamalarının gerçekleştirilebilmesine izin verir. Paralel yapının bir diğer yararı da çok hızlı hesaplama yapabilmeyi sağlamasıdır. Bir birimin t zamanındaki çıktısı, sadece $t - 1$ zamanındaki girdilere bağlıdır. Aynı katmandaki birimler birbirlerinden bağımsız

olduklarından, bu birimler üzerinde aynı anda işlem yapılabilir (Mehrotra vd., 1997: 39). Katmanlar birbirlerine seri olarak bağlandıklarından L katmanlı ileri yönelimli bir ağ, girdideki değişikliğe L adım sonra cevap verebilir. 2000 birimli iki katmanlı bir ağ çıktıyı, bu birimlerin seri olarak işlendiğinde gerekecek olan 2000 adım yerine sadece 2 adımda bulacaktır.

2.1.5.6 Sağlık ve Hata Toleransı

YSA, sağlık ve hatalara karşı toleranslı olma özelliklerine sahiptir. Bir YSA bulanık, eksik, gürültülü ve olasılığa dayalı veri ile işlem yapabilir. Bilginin ağ yapısında dağılmış bir şekilde bulunmasından dolayı, dağıtık bir bilgi işleme sistemidir. Her bir birimin çıktısı oluşturulurken birçok girdi, sistemin nihai çıktısını oluşturmak için ise birçok birim kullanıldığından sistem küçük bozulmalara karşı çok hassas değildir.

Dağıtık yapıdan dolayı birkaç birimin kullanım dışı kalması sistem performansında azalmaya sebep olsa da bütünüyle bir çökmeye yol açmaz (Munakata, 2008: 35). Herhangi bir bozulmada ağ, bağlantı ağırlıklarını yeniden ayarlayarak hemen performansı arttırabilir. Böylece kendi kendini tamir eder ve hatalara karşı tolerans yeteneği kazanır.

2.1.6 Yapay Sinir Ağlarının Zayıf Yönleri

Yapay sinir ağlarının yukarıda belirtilen avantajlarının yanında bazı dezavantajları da vardır. (Vellido vd., 1999: 62) tarafından yapılan literatür taramasında, 1992-1998 yılları arasında YSA kullanılarak yapılan finansal uygulamalar incelemiş ve araştırmacıların YSA'nın dezavantajları konusundaki görüşleri Tablo 5 de özetlenmiştir.

Tablo 5. Yapay Sinir Ağlarının Dezavantajları

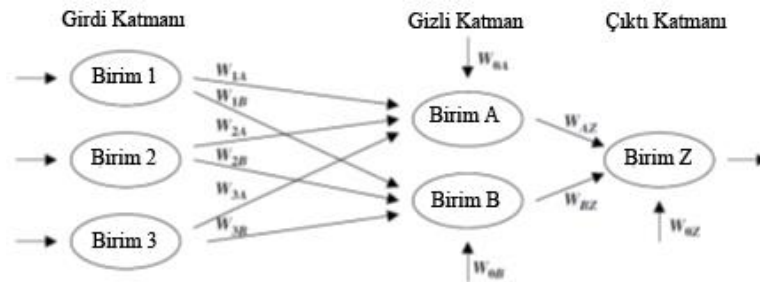
Belirtilen YSA Dezavantajı	Yayın Adedi
Ağın davranışını açıklayacak teorik altyapı yok (Kara kutu).	28
Optimal ağ mimarisini ve parametrelerini seçmek için teorik altyapı yok (Deneme yanılma süreci).	21
Öğrenme süresi çok uzun olabilmektedir.	11
Aşırı öğrenmeye düşerek genelleme yapamayabilir.	10
Öğrenme algoritmalarının seçimi için açık tanımlanmış kurallar yok.	8
Verilerin miktarına ve kalitesine aşırı bağımlı olabilmektedir.	6
Eğitim sırasında yerel minimuma takılma riski bulunmaktadır.	5
Halen gelişmekte olduğundan henüz güvenilir ve güçlü bir yöntem değildir.	3
Klasik istatistikteki güven aralığı ve hipotez testi gibi özellikleri taşımamaktadır.	2

2.1.7 Yapay Sinir Ağlarının Temel Elemanları

Yapay sinir ağlarında yer alan temel elemanlar aşağıda açıklanmıştır.

2.1.7.1 Katmanlar

Ağların birçoğu girdi, gizli ve çıktı katmanı olmak üzere üç katmandan oluşur (Şekil 6). Gizli katman sayısı daha fazla olabilir fakat birçok uygulama için tek bir gizli katman yeterli olmaktadır. İki birim arasındaki her bir bağlantının w ile gösterilen bir ağırlığı vardır (Larose, 2005: 132).



Şekil 6. Yapay Sinir Ağı Katmanları

Girdi birimlerinin sayısı veri kümesindeki özelliklerin tiplerine ve sayısına bağlıdır. Gizli katman sayısı ve her bir gizli katmandaki birim sayısı ise kullanıcı tarafından belirlenebilir. Çözülecek probleminin özelliğine bağlı olarak çıktı katmanında birden fazla birim olabilir.

2.1.7.1.1 Girdi Katmanı

Girdi katmanı varsayımsal bir katmandır. Özellik değerlerini veri kümesinden alır ve bu değerleri herhangi bir işleme tutmadan gizli katmana iletir. Girdi katmanda yer alan birimlerin üzerinde hiçbir hesaplama yapılmadığından bu katmandaki birimlerin hataları da olmaz. Dolayısıyla, gizli ve çıktı katmandaki gibi detaylı bir birim yapısına sahip değildir.

2.1.7.1.2 Gizli Katman

Gizli katmandaki birimler, doğrusal olmama davranışları dolayısıyla, YSA'nın davranışındaki doğrusal olmama durumunun kaynağını teşkil ederler.

Daha fazla gizli katman biriminin ağı karmaşık örüntüleri belirlemedeki yeteneğini ve esnekliğini arttıracak düşünülduğünden, bu katmanda çok fazla düğüm kullanılmak istenebilir. Diğer taraftan, gereğinden büyük bir gizli katman eğitim kümesinin ezberlenerek genelleme yeteneğinin düşmesine yani aşırı öğrenmeye yol açmaktadır. Eğer aşırı öğrenme oluşmuşsa, gizli katmandaki birim sayısı azaltılabilir veya tam tersi eğer eğitim sonuçları istenilen seviyede değilse gizli katmandaki birim sayısı arttırılabilir (Mehrotra vd., 1997: 85).

Gizli katman birimleri, n boyutlu girdi uzayından gizli birimlerin tanımlandığı H boyutlu uzaya doğrusal olmayan bir eşleme gerçekleştirirler (Alpaydın, 2011: 208).

Ağa her yeni gizli katman eklendiğinde, eğitim sürecinde hata fonksiyonunun takılabileceği yerel minimum noktaları eklenir. Bu nedenle, önce tek bir gizli katmanlı ağ oluşturulur. Ancak, bu ağ yetersiz ise o zaman yeni bir gizli katman

eklemek gerekir. Uygulamalarda, ihtiyacı karşılamaya yetecek en az sayıda gizli katman kullanılmaya çalışılır (Kriesel, 2007: 93).

Gizli katman(lar)daki birim sayısının optimallik anlamında doğru sayısını veren herhangi bir analitik yöntem geliştirilememiştir. Dolayısıyla, gizli katman sayısındaki ve bu katmanların birim sayılarındaki belirsizlikleri aşabilmenin tek yolu deneme yanılma yöntemidir (Munakata, 2008: 10).

“Doğrusal ayrılabilir (*linearly separable*)” problemler için gizli katman kullanılmasına gerek yoktur. Çünkü böyle problemlerde çok katmanlı modellerin tek katmanlı modellere bir üstünlüğü yoktur (Kecman, 2001: 244).

2.1.7.1.3 Çıktı Katmanı

İkili sınıflandırma problemleri için çıktı katmanında sınıfları ayıracak bir “eşik (*threshold*)” değerinin önsel olarak atandığı tek bir birim kullanılması yaygındır. Tek bir çıktı birimi, birden fazla sınıfın olduğu problemlerde de kullanılabilir. Aynı şekilde bu sınıfları birbirinden ayırt edebilmek için her bir sınıf için eşik değerlerinin belirlenmesi gerekir. Eşik değerlerinin doğru bir şekilde belirlenmesi için alan uzmanlarının değerlendirmeleri gereklidir.

Eğer olası sınıf sayısı fazla değilse bayrak (*flag*) değişkenleri de kullanılabilir. K sınıftan oluşan değişkenler $K - 1$ adet bayrak değişkenine çevrilebilirler.

Birçok ağda, bilgi işlem birimleri aşağıdaki şekilde çıktı üretirler.

$$y = f\left(\sum_i w_i x_i\right) \quad (2.1)$$

(Denklem 2.1) de x_i , diğer birimlerin çıktılarını veya dışarıdan gelen girdi değerlerini, w_i bağlantı ağırlıklarını, $f(\cdot)$ ise doğrusal olmayan bir fonksiyonu gösterir. Yani her bir birim kendisine gelen girdilerin ağırlıklandırılmış toplamını

hesaplar ve skaler bir çıktı oluşturmak amacıyla doğrusal olmayan f fonksiyonuna gönderir. Genellikle f , sınırlı, azalmayan, doğrusal olmayan bir fonksiyondur (Reed & Marks, 1999: 1).

2.1.7.2 Ağırlıklar

YSA modellerinde bilginin gösterimi ağırlıklar ile yapılır. Bilginin gösterim şekli ağı tasarımı ve dolayısıyla performansını etkileyen önemli bir faktördür. Bu amaçla dört temel kural kullanır (Haykin, 1999: 48-50).

Kural 1: Benzer sınıflardan gelen benzer girdiler, ağda benzer gösterimler oluşturmalı yani aynı kategoride yer alacak şekilde sınıflandırılmalıdırlar. Girdilerin benzerliklerini ölçmek için Euclid uzaklığı, iç çarpım veya Mahalanobis uzaklığı gibi benzerlik ölçüleri kullanılır.

Kural 2: Aynı kategoriler halinde sınıflandırılacak girdilerin ağdaki gösterimleri arasında büyük farklılıklar olmalıdır. Bu kural ilk kuralın tam tersidir.

Kural 3: Bir öznelik önemliyse bu özneliğin temsilinde yer alan birim sayısı daha fazla olmalıdır. Birim sayısının fazla olması karar vermede yüksek kesinlik ve hata durumlarına karşı tolerans sağlar.

Kural 4: Önsel bilgiler, ağ tasarımı sırasında kullanılmalıdır. Böylece, bu bilgiler öğrenilmek zorunda kalınmayacağı için daha basit bir ağ oluşturulabilir.

Önsel bilgiyi ağ tasarımında kullanmanın kesin tanımlanmış kuralları olmamakla birlikte “*ağırlık paylaşımı (weight sharing)*” gibi iyi sonuçlar verdiği bilinen yaklaşımlar kullanılmaktadır. Ağırlık paylaşımında, ağırlıkların seçimi kısıtlanarak ağdaki serbest parametre sayısı belirgin bir şekilde düşürülebilir (Bishop, 1995: 325).

2.1.7.3 Birleştirme Fonksiyonu

Birleştirme fonksiyonu her bir birim için ilgili birimin girdi değerleri ile ağırlıklarının doğrusal birleşiminden *net girdi* olarak adlandırılan bir değer üretir. En yaygın olarak kullanılan birleştirme fonksiyonu, her gelen girdinin kendi ağırlığı ile çarpılarak toplandığı ağırlıklı toplamdır. Ağırlıklı toplam kullanılarak herhangi bir j düğümü için net girdi (Denklem 2.2) deki gibi bulunur.

$$net\ girdi_j = \sum_i w_{ij}x_{ij} = w_{0j}x_{0j} + w_{1j}x_{1j} + \dots + w_{lj}x_{lj} \quad (2.2)$$

x_{ij} , j düğümüne gelen i . girdiyi göstermektedir. w_{ij} , toplam $l + 1$ adet girdinin olduğu j düğümü ile i . girdi arasındaki ağırlığı göstermektedir. x_1, x_2, \dots, x_l birime gelen girdileri temsil eder. $x_{0j} = 1$ olarak gösterilir yani her zaman sabit olarak 1 değerini almaktadır. Bu yüzden, her bir gizli veya çıktı katman birimi değeri 1 olan ekstra bir girdiye sahiptir (Rumelhart vd., 1986: 535).

Birleştirme fonksiyonu olarak değişik seçenekler bulunmaktadır (Tablo 6). Bir problem için en uygun birleştirme fonksiyonunu belirlemek için bulunmuş bir formül yoktur. Genellikle deneme yanılma yöntemiyle birleştirme fonksiyonu belirlenmektedir (Öztemel, 2006: 49-50).

Tablo 6. Birleştirme Fonksiyonları

Birleştirme Fonksiyonu	Net Girdi
Ağırlıklı toplam	$net\ girdi = \sum_i x_i w_i$
Çarpım	$net\ girdi = \prod_i x_i w_i$
Maksimum	$net\ girdi = maks(x_i w_i)$
Minimum	$net\ girdi = min(x_i w_i)$
Çoğunluk	$net\ girdi = \sum_i sgn(x_i w_i)$
Kümülatif toplam	$net\ girdi = net(eski) + \sum_i x_i w_i$

2.1.7.4 Etkinlik (Aktivasyon) Fonksiyonu

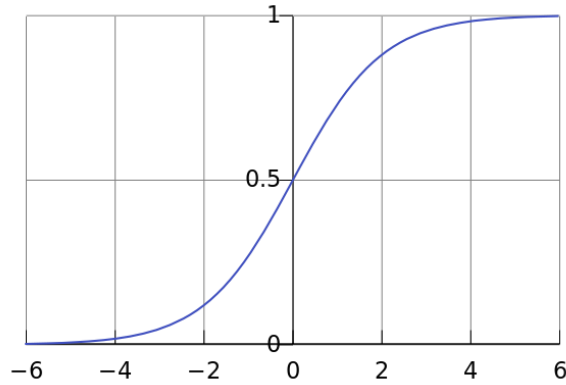
YSA'da birimlerin davranışını belirleyen önemli faktörlerden birisi, kullanılan etkinlik fonksiyonlarıdır. Biyolojik nöronlarda, eğer bir nöronun ağırlıklandırılmış girdilerinin toplamı belli bir eşik değerini geçerse, bu nöronun diğer nöronlara elektriksel sinyal gönderdiği bilinmektedir. Ayrıca biyolojik nöronlarda çıktı, girdideki değişimlere göre doğrusal olarak değişmez yani doğrusal olmayan bir davranış gösterir. Bu davranışa benzer bir davranışı yapay nöronlarda elde etmek için doğrusal olmayan etkinlik fonksiyonları kullanılır.

En yaygın kullanılan etkinlik fonksiyonu 0 ile 1 arasında çıktılar oluşturan sigmoid fonksiyonudur (Denklem 2.3). Bulunan net girdi değeri sigmoid fonksiyonuna girdi olarak sunulur. Fonksiyon, bu girdiye göre ilgili birimin çıktısını oluşturur. Bu çıktı değeri aynı şekilde çıktı katmanına ağırlıklandırılmış olarak gönderilir (Zurada, 1992: 34).

$$f(x) = \frac{1}{1 + e^{(-ax)}} \quad (2.3)$$

(2.3) denklemindeki a , eğim parametresidir. Bu parametre değiştirilerek farklı eğimlere sahip sigmoid fonksiyonları elde edilebilir. Orijinde eğim $a/4$ tür. Eğim sonsuza yaklaştığında sigmoid fonksiyonu eşik fonksiyonuna dönüşür.

Sigmoid fonksiyonu, girdiye bağlı olarak doğrusala yakın, eğrisel ve sabite yakın davranışlarını birleştiren bir fonksiyondur. Merkeze yakın yerlerde doğrusala yakın, merkezden uzaklaşmaya başladıkça eğrisel, uç değerlere yaklaşıncaya neredeyse sabit davranış gösterir (Şekil 7). Bu nedenle bulunduğu konuma bağlı olarak girdi değerindeki küçük artışlar fonksiyonun değerinde farklı büyüklükte artışlar meydana getirir. Merkeze yakın noktadaki küçük artışlar fonksiyonda küçük artışlar meydana getirirken, uç noktalarda girdi değerinin artması fonksiyonun değerinde çok küçük artışlar oluşturur (Larose, 2005: 134).



Şekil 7. Sigmoid Fonksiyonu

Etkinlik fonksiyonunun negatif değerleri de alması istenirse sigmoid fonksiyonu yerine hiperbolik tanjant (\tanh) fonksiyonu (Denklemler 2.4) kullanılır.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.4)$$

Etkinlik fonksiyonunun parametrik olması ağırlıkların kazandırdığı esnekliğe ek olarak değişik eşleştirmelerin yapılabilmesini mümkün kılan bir esneklik sağlar. Bu amaçla hiperbolik tanjant etkinlik fonksiyonu, parametrik olarak yeniden düzenlenebilir (Denklem 2.5). Parametrelerin değiştirilmesi ile etkinlik fonksiyonunun şekli değiştirilebilir (Efe & Kaynak, 2000: 8).

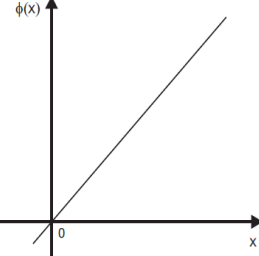
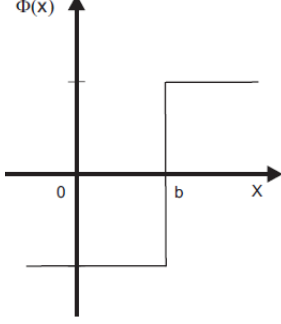
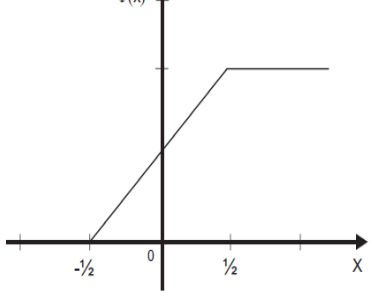
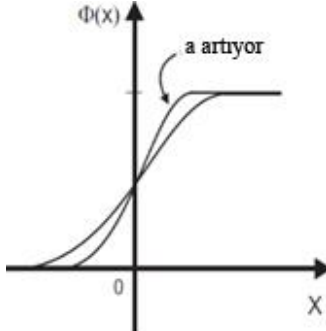
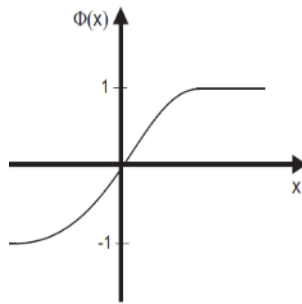
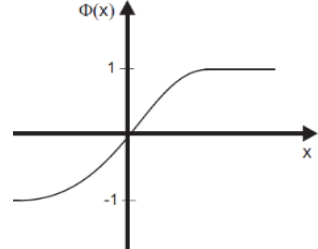
$$f(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}} \quad (2.5)$$

Gizli birimlerde doğrusal olmayan bir etkinlik fonksiyonu yerine doğrusal fonksiyon kullanılırsa gizli katman hiçbir işe yaramaz. Çünkü doğrusal birleşimin doğrusal birleşimi yine doğrusaldır.

Fonksiyon yaklaşma problemleri için gizli katman birimlerinde hiperbolik tanjant, çıktı birimlerinde ise çıktı katmanında sınırlandırılmış aralıkta değerler istenmediği için doğrusal etkinlik fonksiyonu tercih edilir (Kriesel, 2007: 94).

(Tablo 7) yaygın kullanılan etkinlik fonksiyonlarını göstermektedir.

Tablo 7. Yaygın Olarak Kullanılan Etkinlik Fonksiyonları

 <p>(a) Doğrusal fonksiyon $y = \varphi(x) = x$</p>	 <p>(b) Basamak-eşik fonksiyonu $y = \varphi(x) = \begin{cases} 1 & x \geq b \\ -1 & x < b \end{cases}$</p>	 <p>(c) Parçalı doğrusal fonksiyon $y = \varphi(x) = \begin{cases} 1 & x \geq 1/2 \\ z & -1/2 \leq x < 1/2 \\ 0 & x \leq -1/2 \end{cases}$</p>
 <p>(d) Lojistik (sigmoid) fonksiyonu $y = \varphi_1(x) = \frac{1}{1 + e^{-ax}}$</p>	 <p>(e) Simetrik sigmoid $y = \varphi_2(x) = 2\varphi_1(x) - 1 = \frac{1 - e^{-ax}}{1 + e^{-ax}}$</p>	 <p>(f) Hiperbolik tanjant $y = \varphi_3(x) = \tan g\left(\frac{x}{2}\right) = \varphi_2(x)$</p>

2.1.7.5 Hata Fonksiyonu

Hata fonksiyonu olarak birçok YSA modelinde “*Hata Kareleri Toplamı (HKT)*” kullanılır (Denklem 2.6). HKT, eğitim kümesindeki tüm gözlemlerin tüm çıktığı birimlerindeki tahmin hatalarının karelerinin toplamıdır (Zurada, 1992: 176).

$$HKT = \sum_{\text{girdiler}} \sum_{\substack{\text{çıktı} \\ \text{birimleri}}} \left(\begin{matrix} \text{beklenen} \\ \text{çıktı} \end{matrix} - \begin{matrix} \text{gerçekleşen} \\ \text{çıktı} \end{matrix} \right)^2 \quad (2.6)$$

Uygulamalarda hata kareleri toplamına ek olarak standart olarak kullanılan birkaç hata fonksiyonu vardır. “*Ortalama Hata Kareleri (OHKT)*”, HKT değerini

eđitimde kullanılan gözlem sayısı ve ađın çıktı sayısını kullanarak normalize etmektedir. Ađ çıktısının bir olasılık tahmini vermesinin istenildiđi durumlarda hata fonksiyonu olarak “*çapraz entropi (cross entropy)*” fonksiyonu kullanılır (Mitchell, 1997: 118).

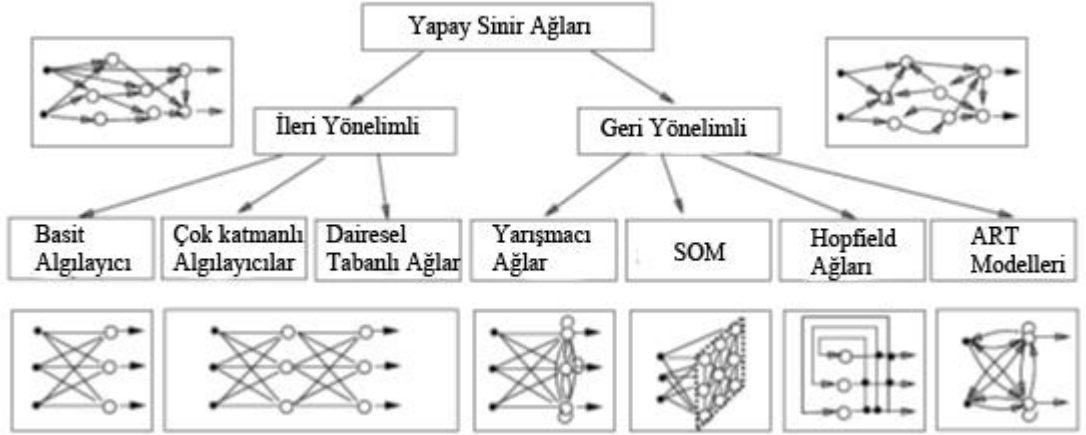
Standart hata fonksiyonlarının kolaylıkla türevlenebilir olmaları, hataların önceki hataların eğilimlerinden ve büyüklüklerinden etkilenmemeleri ve girdiden bağımsız olarak eşit hataların maliyetlerinin eşit olması bu fonksiyonların avantajları arasında sayılabilir. Bu sayede modellerin analiz edilmesi daha kolay hale gelmiştir.

Bazı uygulamalarda ise girdi değerlerine veya başka faktörlere bađlı olarak birbirine yakın değerlerdeki hataların farklı maliyet değerleri oluşturdukları fonksiyonlar daha yararlı olmaktadır. Hata fonksiyonlarının seçimi uygulamaya bađımlı olmakla birlikte, uygulamaların büyük bir kısmında standart hata fonksiyonları kullanılmaktadır (Reed & Marks, 1999: 10).

2.1.8 Yapay Sinir Ađı Mimarileri

Farklı bađlantı türleri, ađın farklı davranışlar göstermesine neden olur. Yapay sinir ađlarında ileri yönelimli ve geri yönelimli olmak üzere iki temel mimari söz konusudur. İleri yönelimli ađlar, örüntü tanıma, fonksiyon kestirme, tahmin gibi uygulama alanlarında kullanılırken, geri yönelimli ađlar ise daha çok optimizasyon ve kümeleme problemlerinin çözümünde kullanılmaktadır (Cura, 2008: 19).

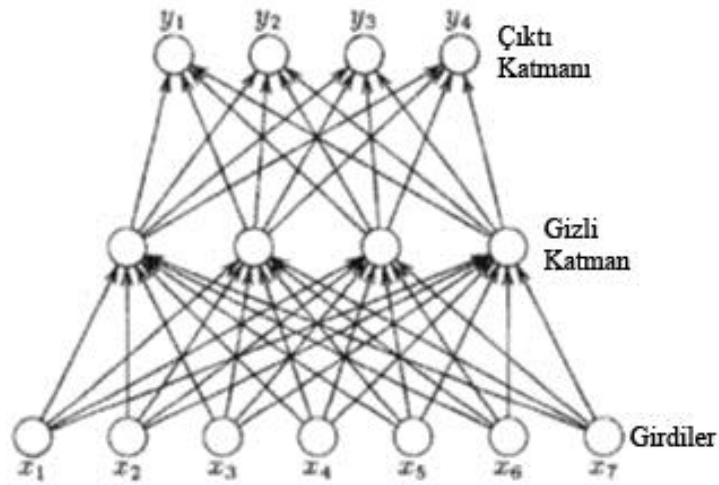
Şekil 8 ileri ve geri yönelimli mimariye sahip temel YSA modellerini göstermektedir.



Şekil 8. Ağ Mimarileri

2.1.8.1 İleri Yönelimli Ağlar

(Şekil 9) da gösterilen ağ, çıktılardan girdilere geriye doğru herhangi bir çevrim (loop) içermediğinden “ileri yönelimli (*feedforward*)” bir yapıdadır. Ağ, önceki sistem durumlarından bağımsız şekilde sadece eldeki güncel girdilere bağlı olarak girdiler ile çıktılar arasında statik eşleştirme yapmaktadır (Reed & Marks, 1999: 2). Bir girdiye karşılık oluşturulan çıktı, ağın önceki durumundan bağımsız olduğundan ileri yönelimli ağlar genellikle belleksiz ağlar olarak da bilinirler (Jain vd., 1996: 17).

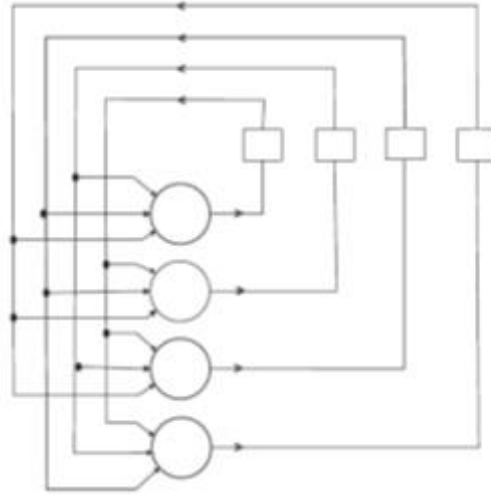


Şekil 9. İleri Yönelimli Çok Katmanlı Ağ

(Şekil 9) da gösterilen ağ, aynı zamanda “tam bağlantılı (*fully connected*)” bir ağıdır. Yani L katmanındaki her bir birim $L - 1$ katmanındaki her birimden girdi alır ve $L + 1$ katmanındaki birimlerin tamamına çıktısını gönderir. Katmanlardaki birimlerin birbirleriyle bağlantıları yoktur.

2.1.8.2 Geri Yönelimli Ağlar

“Geri yönelimli (*feedback - recurrent*)” ağlar ise dinamik sistemlerdir (Şekil 10). Yeni bir girdi ağı sunulduğunda karşılık gelen çıktı hesaplanır. Geri besleme bağlantılarından dolayı her bir birime gelen girdilerin değiştirilmesiyle ağ yeni bir durumu girer. Bu süreç yakınsama sağlanıncaya kadar tekrarlanır (Dreyfus, 2010: 8).



Şekil 10. Geri Yönelimli Ağ

Geri yönelimli bağlantılar, kısa süreli bir bellek gibi davranır ve ağın yakın geçmişte gördüğü veriyi hatırlamasını sağlar. En sık, çok katmanlı bir ağı kısıtlı sayıda geriye bağlantıların eklendiği yapılar kullanılır. Böylece; çok katmanlı ağın tahmin yeteneği ile geri yönelimli bağlantıların zamansal gösterim yeteneği birleştirilmiş olur. Fakat belli bir uygulama için hangi geri yönelimli ağ yapısının en iyi olduğu konusunda fazla teorik bir çalışma yapılmamıştır (Alpaydın, 2011; 229).

Farklı ağ tiplerinde farklı matematiksel araçlar kullanılır. İleri yönelimli ağlarda genellikle hesaplamalar için türev kullanılırken, dinamik sistemlerde

diferansiyel denklemler kullanılır. Geri dönüşlü ağların eğitimi ve analizi daha zordur (Jain vd., 1996: 17).

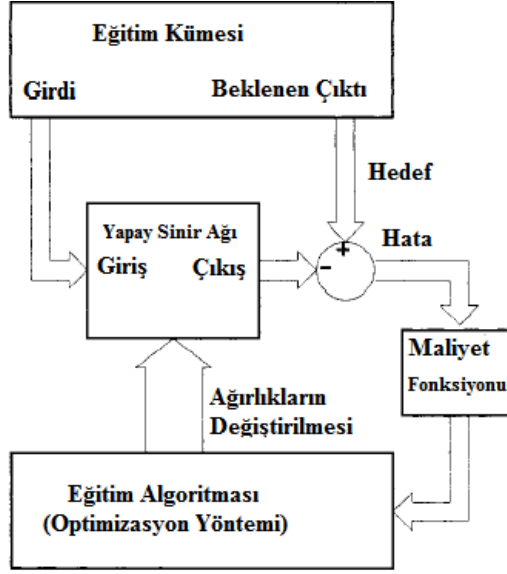
2.1.9 Yapay Sinir Ağlarında Öğrenme

Teorik olarak yapay sinir ağlarında öğrenme:

- Yeni bağlantılar oluşturma
- Var olan bağlantıları silme
- Bağlantı ağırlıklarını değiştirme
- Birimlerin eşik değerlerini değiştirme
- Etkinlik fonksiyonunu değiştirme
- Yeni birimler oluşturma
- Var olan birimleri veya bağlantıları kaldırma

yoluyla olur. Bu yollardan en yaygın olarak kullanılanı ağırlıkları değiştirmektir (Kriesel, 2007: 51).

Yapay sinir ağlarında öğrenme, ağın verilen bir girdiye karşılık beklenen çıktıyı oluşturacak şekilde ağırlıkların ayarlaması sürecidir. Girdiler ile çıktılar arasındaki fonksiyonel ilişki bilinmediğinden birçok fonksiyonu uygulayabilecek kadar esnek bir ağ ile başlanır ve bu ağ üzerinde eldeki veriye uyacak şekilde ayarlamalar yapılır. Bu, genellikle bir optimizasyon işlemi eşliğinde devam eden tekrarlamalı (iteratif) bir süreçtir (Şekil 11). Ağ mükemmel bir şekilde eğitilmediği sürece, gerçekleşen ile beklenen sonuçlar arasında farklar olacaktır. Amaç, verilen bir veri kümesi üzerinde toplam hatayı en aza indirecek ağı oluşturmaktır (Reed & Marks, 1999: 7).



Şekil 11. Yapay Sinir Ağlarında Denetimli Öğrenme

2.1.9.1 Toplu (Batch) Öğrenme

Eğitim kümesi üzerinde bir tam geçiş yapıp bütün örnekler için bir kez güncelleme yapmaya “*dönem (epoch)*” denir (Munakata, 2008: 12). Bu durumda, ağıdaki ağırlıklar gözlemlerin tamamı ağa sunulmadıkça değiştirilmez. Genel olarak eğitim kümesinde az sayıda gözlem olduğunda kullanılmaktadır. Toplu öğrenmede ağırlıklar, girdilerin çoğunluğunun istediği yönde değiştirilir.

2.1.9.2 Çevrimiçi (Online) Öğrenme

Her bir gözlem ağa sunulduktan sonra ağıın parametreleri biraz güncellenerek zaman içinde veriye uyarlanır. Çevrimiçi öğrenmede hata fonksiyonu tüm örnekler üzerinde değil, tekil gözlemler üzerinde tanımlanır. Bu yaklaşımın YSA uygulamalarında kullanılma sebepleri şöyle sıralanabilir (Alpaydın, 2011: 203).

- Eğitim kümesinin ya da en iyi parametre değerlerini ararken oluşan ara sonuçların saklanması için ek bir belleğe ihtiyaç yoktur.
- Öğrenilmek istenilen görev, yapısı gereği zamanla değişebilir. Bu nedenle önceden bir eğitim kümesi oluşturma anlamlı olmayabilir.

Eđitim kümesi çok sayıda birbirinin aynısı gözlem içerdiğinde çevrimiçi öğrenme toplu öğrenmeye göre daha avantajlıdır. Çevrimiçi öğrenmenin programlanması daha kolay olduđu için daha yaygın olarak kullanılır. Ayrıca yerel minimuma yakalanma olasılığı daha az olduğundan daha iyi sonuçlara ulaşılabilir (Marsland, 2009: 60).

2.1.9.3 Öğrenme Kuralları

Bir öğrenme probleminin çözümü için kullanılan iyi tanımlanmış kurallar kümesine “*öğrenme algoritması*” denilir. YSA tasarımı için tek bir öğrenme algoritması yoktur. Avantajları ve dezavantajları olan farklı birçok öğrenme algoritması vardır (Zurada, 1992: 26). Temelde öğrenme algoritmaları, ağırlıkları deđiştirme biçimlerine göre farklılaşırlar.

Farklı ağ mimarileri farklı öğrenme algoritmaları gerektirir. Öğrenme yöntemi (denetimli, denetimsiz, vb.) ağa hangi bilginin verildiğini belirtirken, “*öğrenme kuralı*” ağırlıkların ne şekilde güncellendiğini gösterir.

2.1.9.3.1 Hata Düzeltme Kuralı

Ağın çıktısı ile beklenen çıktı arasındaki fark (hata) adım adım azaltılmaya çalışılır. Bunun için amaç fonksiyonu kullanılır. Ağırlık deđişimi, sistem durağan hale gelinceye kadar devam eder (Gorunescu, 2011: 209).

2.1.9.3.2 Boltzmann Öğrenme Kuralı

Boltzmann öğrenme kuralı, kökleri istatistiksel mekanik alanındaki fikirlere dayanan stokastik bir öğrenme kuralıdır. Bu kurala dayalı olarak tasarlanan yapay sinir ağına “*Boltzmann makinesi*” denir. Bir Boltzmann makinesi, geri yönelimli bir yapıdadır. Birimler aktif veya pasif haldedirler (Haykin, 1999: 83). Beklenen çıktı ile gerçekleşen çıktı arasındaki farkın yerine, farklı çalışma durumlarında iki birimin çıktılarının korelasyonları arasındaki fark kullanılır. Bu yönüyle Boltzmann öğrenme kuralı, hata düzeltme kuralının özel bir şekli olarak deđerlendirilebilir (Jain vd., 1996: 23).

2.1.9.3.3 Hebb Öğrenme Kuralı

Bu kurala göre, bir sinapsın iki tarafındaki sinir hücreleri aynı anda aktif hale gelirse bu sinir hücreleri arasındaki bağlantının gücü artar, aynı anda aktif hale gelmezlerse bağlantının gücü azalır veya bu bağlantı ortadan kalkar (Peretto, 1992: 5). Bir sinapsın her iki tarafındaki sinyallerin arasındaki korelasyon, sinaptik ağırlık değişiminden sorumludur. Dolayısıyla Hebb öğrenme kuralında korelasyon, öğrenmenin temelini oluşturur. Bu öğrenme kuralında karşılaşılan problemlerden birisi, öğrenme ilerledikçe bağlantı ağırlıklarının bir üst sınır olmadan büyümesidir.

2.1.9.3.4 Yarışmacı Öğrenme Kuralı

Yarışmacı öğrenmede ağırlık çıktı birimleri kendi aralarında aktif olmak için yarışır. Bu öğrenme türünde aynı anda sadece tek bir çıktı birimi aktif olabilir. Bu birime “*kazanan birim*” denir. Bir çıktı biriminin kazanan birim olabilmesi için, ağırlık çıktı birimleri içinde en büyük net girdiye sahip olması gerekir. Yarışma kazanan çıkış birimin değeri 1, diğer tüm çıkış birimlerinin değeri 0 olur (Yegnanarayana, 2005: 201-202).

2.1.9.3.5 Hafıza Tabanlı Öğrenme Kuralı

Eğitim gözlemlerinin tamamı veya birçoğu, doğru girdi çıktı çiftleri halinde büyük bir hafızada tutulur. Bir test gözleminin sınıflandırılması istendiğinde algoritma bu gözlemin yerel komşuluğunda yer alan eğitim gözlemini analiz ederek bir sonuç üretir.

Tüm hafıza tabanlı öğrenme algoritmaları iki temel bileşen içerir (Haykin, 1999: 76).

1. Test gözleminin yerel komşuluğunu tanımlamak için kullanılacak ölçüt.
2. Test gözleminin yerel komşuluğundaki eğitim gözlemlerine uygulanılacak öğrenme kuralı.

2.2 Tek Katmanlı Yapay Sinir Ağları

Matematiksel modellerin amacı, tüm gereksiz detayları ortadan kaldırarak üzerinde çalışılan konuyu doğru bir şekilde temsil edecek temel özellikleri ortaya çıkarmaktır. McCulloch ve Pitts 1943 yılında bir sinir hücresinin matematiksel modelini ortaya koyarak YSA alanındaki ilk çalışmayı yapmışlardır.

McCulloch - Pitts yapay nöronu bir “*ikili eşik (binary threshold)*” aygıtıdır. Çok basit bir çalışma yöntemine sahip olmasına karşın daha karmaşık modellerin temelini oluşturduğu için tarihsel bir öneme sahiptir (Rojas, 1996: 32).

2.2.1 Basit Algılayıcı (Perceptron)

İlk yapay sinir ağı 1958 yılında Rosenblatt tarafından ortaya atılmıştır ve “*perceptron*” olarak isimlendirilmiştir. Basit algılayıcı modeli, öğrenme yeteneğine sahiptir. Şekil tanıma, görsel algılama problemlerini çözmek için geliştirilmiştir. İlk ortaya atıldığı zaman, öğrenme çözülememiş bir görev olduğundan gözlemleri kullanarak ağırlıkların otomatik değiştirilebilmesi fikri çok heyecan yaratmıştır. (Cura, 2008: 19-20).

Basit algılayıcı McCulloch-Pitts birimlerinin bir araya gelmesiyle oluşur. Birimler birbirinden tamamen bağımsızdır yani her bir birime bağlı ağırlıklar birbirinden ayrıdır. Paylaşılan tek şey girdilerdir. Her bir birim ağa sunulan tüm girdileri işler (Negnevitsky, 2004: 213).

x_1, \dots, x_n girdileri verildiğinde basit algılayıcı tarafından hesaplanan $o(x_1, \dots, x_n)$ çıktısı

$$o(x_1, \dots, x_n) = \begin{cases} 1 & w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0 \text{ ise} \\ -1 & \text{değilse} \end{cases} \quad (2.7)$$

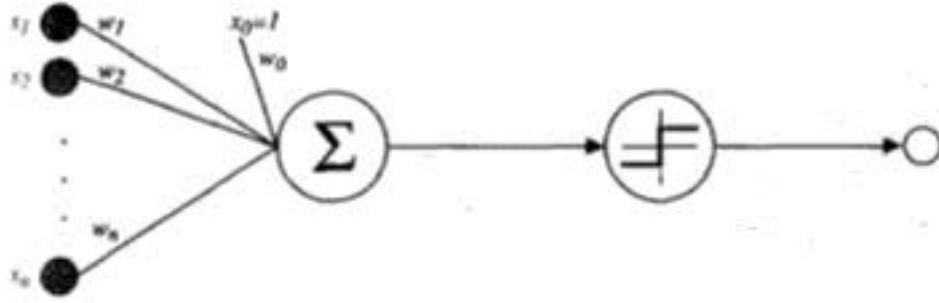
şeklinde gösterilebilir (Şekil 12). w_i , bir x_i girdisinin basit algılayıcının çıktısına olan katkısını belirtir. $(-w_0)$ ise basit algılayıcının 1 değerini oluşturabilmesi için

girdilerin ağırlıklı birleşiminin $(w_1x_1 + w_2x_2 + \dots + w_nx_n)$ aşması gereken eşik değerini göstermektedir.

Gösterimi basitleştirmek için, basit algılayıcıya her zaman 1 değerini alan $x_0 = 1$ ek girdisi eklenir. Bu girdiye “*kaydırma (offset)*” terimi denilir. Böylece (2.7) denklemindeki eşitsizlik daha kısaca $\sum_{i=0}^n w_i x_i > 0$ olarak yazılabilir (Şekil 12). İç (nokta) çarpımı kullanarak $(w \cdot x) > 0$ biçiminde de aynı ifadeyi göstermek mümkündür. Bu durumda basit algılayıcı karar fonksiyonu *sgn* işaret fonksiyonunu göstermek üzere

$$o(x) = \text{sgn}(w \cdot x) \quad (2.8)$$

şeklindedir.



Şekil 12. Basit Algılayıcı (Perceptron)

Basit algılayıcıda öğrenme w_0, \dots, w_n ağırlık değerlerinin belirlenmesidir. Bu nedenle, basit algılayıcı öğrenmede $H = \{w \mid w \in \mathcal{R}^{(n+1)}\}$ aday hipotezler kümesi, olası tüm reel değerli ağırlık vektörlerinin kümesidir (Mitchell, 1997: 86).

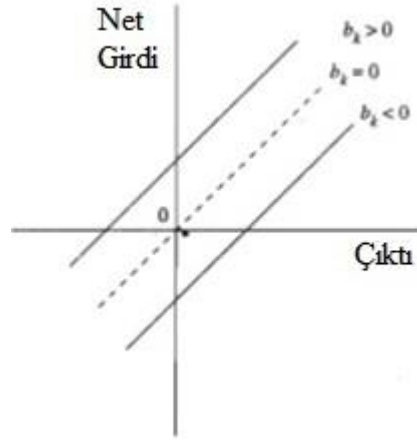
2.2.1.1 Basit Algılayıcı Eşleşiminin Geometrik Yorumu

Tek girdili ve tek çıktılı basit algılayıcı bir doğru gerçektir. Birden çok girdi varsa, n boyutlu girdi uzayında hiperdüzlem şeklinde bir karar yüzeyi bulunur. Hiperdüzlemin bir tarafında yer alan gözlemler için +1 çıktısı oluştururken diğer

yanında kalan gözlemler için -1 çıktısı oluşturulur. Bu karar hiperdüzleminin denklemi $w \cdot x = 0$ biçiminde gösterilebilir (Yeung vd., 2009: 7).

Ağırlık vektörünün (w) uzunluğunun sınıflandırma üzerinde bir etkisi yoktur. Fakat bu vektörün yönü önemlidir. Ağırlık vektörü her zaman ayırma (diskriminant) doğrusuna diktir ve yönü pozitif sınıfın yönündedir. Sınıflama etiketlerinin değiştirilmesi karar sınırını veya ağırlık vektörünün konumunu etkilemez. Fakat ağırlık vektörü her zaman pozitif etiketli sınıfı gösterdiğinden, ağırlık vektörünün yönü değişir (Kecman, 2001: 203).

Öğrenme tamamlanınca doğrusal ayırma fonksiyonu orijinden kayar. Bu kaymanın miktarını kaydırma terimi (b_k) belirler (Şekil 13). Kaydırma terimi olmadan sadece orijinden geçen hiperdüzlemler modellenebilir.



Şekil 13. Kaydırma Teriminin Etkisi

2.2.1.2 Basit Algılayıcının Gösterim Yeteneği

AND ve OR gibi mantık fonksiyonları doğrusal ayrılabilir olduklarından basit algılayıcıyla gösterilebilirler. Fakat XOR gibi bazı mantıksal fonksiyonlarda bu mümkün değildir. Nedeni ise, XOR probleminin doğrusal ayrılamayan bir problem olmasıdır.

(Tablo 8), belirtilen sayıda ikili girdiye sahip mantıksal fonksiyonların toplam sayısını ve bu fonksiyonların doğrusal ayrılabilenlerinin oranlarını göstermektedir (Kriesel, 2007: 83). Görüldüğü gibi, doğrusal ayrılabilir problemlerin oranı girdi sayısı arttıkça hızla azalmaktadır. Bu da tek katmanlı algılayıcıların fonksiyonelliğini kısıtlayan çok önemli bir faktördür.

Tablo 8. İkili Girdi Sayısına Göre Doğrusal Ayrılabilir Fonksiyonların Oranı

İkili Girdi Sayısı	Mantıksal Fonksiyonların Sayısı	Doğrusal Ayrılabilir Fonksiyonların Sayısı	Oran
1	4	4	% 100
2	16	14	% 87.5
3	256	104	% 40.6
4	65.536	1.772	% 2.7
5	$4.3 \cdot 10^9$	94.572	% 0.002
6	$1.8 \cdot 10^{19}$	5.028.134	\approx % 0

Tek bir birimden oluşan basit algılayıcı sadece ikili sınıflandırma yapabilir. Birden fazla çıktı birimi kullanılırsa iki kategoriden fazla olduğunda da sınıflandırma yapılabilir. Yine, algılayıcının doğru çalışabilmesi için bu sınıfların doğrusal ayrılabilir olması gerekir (Haykin, 1999: 139).

2.2.1.3 Basit Algılayıcı Öğrenme Kuralı

Birçok öğrenme işleminde genellikle (0,1) arasında bir değer alan bir “*öğrenme katsayısı (learning rate)*” kullanılır. Basit algılayıcıda güncellemenin büyüklüğü η ile gösterilen bu öğrenme katsayısına bağlıdır. Eğer bu değer çok büyük olursa, son gözlemlerin etkisi daha fazla olur ve kısa soluklu bir öğrenme gerçekleşir. Bu katsayı küçük olduğunda, yakınsama için daha fazla sayıda güncelleme adımı gerekir (Alpaydın, 2011: 205).

Ağırlık güncellemeleri, t : ilgili girdi için beklenen (hedef) çıktı, o : basit algılayıcının bu girdi için ürettiği çıktı olmak üzere

$$w_i \leftarrow w_i + \Delta w_i$$
$$\Delta w_i = \eta(t - o)x_i \quad (2.9)$$

biçiminde yapılır. Buna göre, eğer gerçekleşen çıktı beklenen çıktıya eşitse güncelleme yapılmaz. Aradaki fark ne kadar büyük olursa güncellenenin büyüklüğü de o kadar büyük olur. Eğer gerçekleşen çıktı beklenen çıktıdan küçükse girdi pozitif olduğunda güncelleme pozitif, negatif olduğunda negatiftir. Böylece gerçekleşen çıktının değeri artırılır ve beklenen çıktı ile farkı azaltılır. Eğer gerçekleşen çıktı beklenen çıktıdan büyükse girdi pozitif olduğunda güncelleme negatif, negatif olduğunda pozitiftir. Böylece gerçekleşen çıktının değeri azaltılır ve yine beklenen çıktıya yaklaşır. Güncelleme büyüklüğü girdiye de bağlıdır. Girdi sifıra yakınsa gerçek çıktıya etkisi küçük olduğundan sorumluluğu da azdır ve ağırlığındaki güncelleme de küçüktür. Girdi büyüdükçe ağırlığı daha büyük miktarda güncellenir.

Basit algılayıcıda, öğrenme sırasında tanımlanmış bir hata fonksiyonu yoktur. Bu nedenle, basit algılayıcı optimize edilmemiş bir çözüm sunar. Öğrenme sırasında herhangi bir hata fonksiyonu minimize edilmeye çalışılmadığından tüm gözlemleri doğru olarak ayıran ilk ağırlık vektörü bulunduğu, bu vektör üzerinde herhangi bir değişiklik yapılmadan, çözüm olarak sunulur. Yani, basit algılayıcıda doğrusal olarak ayrılabilen sınıfları ayırabilecek herhangi bir doğru bulunmaya çalışılır (Kecman, 2001: 203).

2.2.1.4 Basit Algılayıcı Yakınsama Teoremi

Doğrusal ayrılabilir bir eğitim kümesi verildiğinde tüm eğitim gözlemlerini doğru olarak ayırabilecek bir ağırlık vektörüne sonlu sayıdaki bir adım sonucunda yakınsama sağlandığı “*Basit Algılayıcı Yakınsama Teoremi*” ile ispatlamıştır. Eğer eğitim kümesi doğrusal ayrılabilir değil ise bu yakınsama garanti değildir (Mitchell, 1997: 86). Bu teoreme göre, yakınsama için gerekli maksimum adım sayısı, öğrenme

katsayısına, başlangıç ağırlıklarına ve eğitim gözlemlerinin ağı sunulma sırasına bağlıdır. Teorem, karşılıklı olarak doğrusal olarak ayrılabilen istenilen sayıda sınıf için geçerlidir (Alavala, 2008: 131).

2.2.2 ADALINE (Adaptive Linear Neuron)

Hem basit algılayıcının hem de ADALINE modelinin kökenleri doğrusallığa dayanır. Basit algılayıcı, örneklerin doğrusal olduğu durumlar için en basit fakat güçlü bir sınıflandırıcıdır. ADALINE ise girdi ve çıktılar arasında ilişki gerçekte doğrusal ise iyi bir regresyon çözümdür. Aynı zamanda karar yüzeyi doğrusal olduğunda sınıflandırma için de kullanılabilir.

ADALINE modelinde çıktı katmanındaki birimlerde doğrusal etkinlik fonksiyonu kullanılır. Bu fonksiyon türevi alınabilen en basit fonksiyon olduğu için ağırlıklara bağlı olarak bir hata fonksiyonu oluşturulabilir (Kecman, 2001: 193). Problem doğrusal ayrılabilir ise ADALINE, her zaman HKT değerini en küçük yapan çözümleri bulacaktır. Fakat bu çözüm, uç gözlemlerin etkilerini azaltacak özellikte değildir. Bu, normal dağılmayan veriyle karşılaşıldığında görülen bir eksikliklerdir.

2.2.2.1 Delta Kuralı

Veriler doğrusal ayrılabilir değil ise basit algılayıcı eğitim kuralı yakınsayamaz. ADALINE modelinde kullanılan delta öğrenme kuralı bu zorluğu aşmak için tasarlanmıştır. Eğer eğitim kümesi doğrusal ayrılabilir değilse delta kuralı en iyi yaklaşıma doğru yakınsar. Delta kuralı, “*En Küçük Kareler (Least Mean Squares - LMS)*” veya “*Widrow-Hoff öğrenme kuralı*” olarak da bilinir (Yeung vd., 2009: 8).

2.2.2.2 Eğimli İniş (Gradient Descent) Yöntemi

Delta kuralının arkasındaki temel fikir, ağırlık vektörlerinden oluşan hipotez uzayında arama yaparken eğimli iniş yönteminin kullanılmasıdır. Delta eğitim kuralının çalışmasını gösterebilmek için eşsiz basit algılayıcının eğitildiği varsayılacaktır. Bu doğrusal birimde çıktı, $o(x) = w \cdot x$ şeklinde bulunur. Girdiler

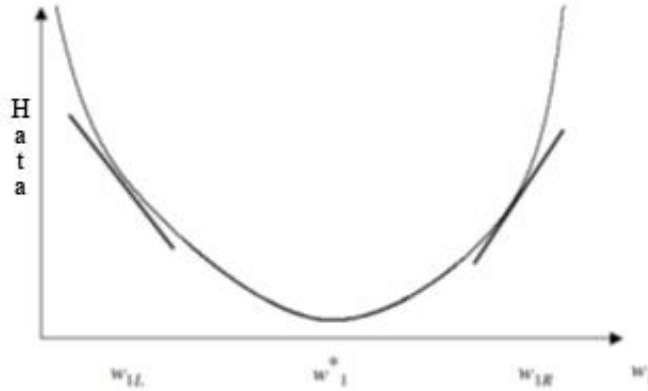
eđitim boyunca sabit olduklarından, hata sadece ađırlıkların bir fonksiyonu olarak $E(w)$ biçiminde gösterilecektir. (Denklem 2.10) da hata kareleri toplamını kullanan hata fonksiyonu görölmektedir.

$$E(w) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad (2.10)$$

(2.10) denkleminde D eđitim kümesini, t_d bu eđitim kümesindeki d . girdinin beklenen çıktısını, o_d ise bu girdi için doğrusal birimin oluşturduğu çıktıyı gösterir.

2.2.2.2.1 Eđimli İniş Yönteminin Elde Edilmesi

Tek bir ađırlığın olduđu varsayılan durumda hatanın bu ađırlığa göre grafiđi (Şekil 14) de gösterildiđi gibi olsun. Bu durumda amaç, hatayı en küçük yapacak w_1^* ile gösterilen ađırlığı bulmaktır.



Şekil 14. Eđimli İniş Yöntemi

Geçerli ađırlık deđerinden optimal w_1^* deđerine doğru ilerlenmesini sağlayacak bir kuralın (Denklem 2.11) kullanılması gerekir.

$$w_{yeni} = w_{geçerli} + \Delta w_{geçerli} \quad (2.11)$$

(Denklem 2.11) $\Delta w_{geçerli}$ geçerli konumda yapılan ağırlık değişimini göstermektedir.

Güncel ağırlık değeri $w_{geçerli}$, w_L yakınında olduğunda, w_1^* optimal değerine yaklaşabilmek için güncel ağırlığın artırılması, w_R yakınlarıdaysa değerinin azaltılması gerekir. $\partial E / \partial w_1$ türevi hata eğrisinin w_1 noktasındaki eğimini göstermektedir. w_L 'ye yakın olan w_1 değerleri için eğim negatif; w_R 'ye yakın değerler için ise eğim pozitifdir. Bu nedenle $w_{geçerli}$ için ağırlık değişimi ($\Delta w_{geçerli}$) E ' nin $w_{geçerli}$ noktasındaki türevinin negatif işareti yönünde yapılır (Larose, 2005: 135).

Genelleme yapılacak olursa, hata yüzeyinde en dik iniş (steepest descent) yönü, hatanın (E) her bir ağırlığa (w) göre kısmi türevi alınarak hesaplanır. Bu yönlü türeve E 'nin w 'ye göre "gradyeni (eğimi)" denilir ve

$$\nabla E(w) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

biçiminde gösterilir. Gradyen hatada en yüksek artışın olduğu yönünü belirttiğinden negatif en büyük azalmanın yönünü verir. Buradan yola çıkılarak eğimli iniş eğitim kuralı

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w = -\eta \nabla E(w)$$

biçiminde yazılabilir. Güncelleme miktarı daha açık bir şekilde her bir ağırlık bileşeni cinsinden

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \quad (2.12)$$

halinde gösterilebilir.

(2.11) denklemine göre ağırlıkları güncelleyecek pratik bir algoritma oluşturulabilmesi için her bir adımda gradyeni hesaplayacak bir yol bulunması gerekir. (2.12) denklemindeki hatanın ağırlıklara göre gradyeni şu adımlar sonucunda bulunur.

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\
&= \frac{1}{2} \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - w \cdot x_d) \\
\frac{\partial E}{\partial w_i} &= \sum_{d \in D} (t_d - o_d)(-x_{id}) \quad (2.13)
\end{aligned}$$

x_{id} , d . gözleminin tek bir x_i girdi değerini göstermektedir. (2.13) denklemi (2.12) denkleminde gösterilen güncelleme kuralında yerine konulduğunda eğimli iniş ağırlık güncelleme kuralı (Denklem 2.14) elde edilir.

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d)x_{id} \quad (2.14)$$

Eğimli iniş yönteminin uygulanmasında karşılaşılan temel zorluklar, minimuma yakınsamasının çok yavaş olabilmesi ve hata yüzeyinde çok sayıda yerel minimum nokta varsa yöntemin bir global minimum noktasına ulaşmasının garanti edilememesi olarak sayılabilir (Alavala, 2008: 148).

2.2.2.3 Stokastik Eğimli İniş Yöntemi

Standart eğimli iniş yönteminde karşılaşılan problemleri azaltmak amacıyla bu yöntemin değiştirilmiş hali olan “*stokastik eğimli iniş yöntemi*” kullanılır. Yeterince küçük öğrenme katsayısı kullanıldığında stokastik eğimli iniş, standart eğimli iniş

yöntemine yaklaşmaktadır. Ağırlık güncellemesi (Denklem 2.15) de gösterilen kurala göre yapılır.

$$w_i \leftarrow w_i + \eta(t - o)x_i \quad (2.15)$$

Stokastik eğimli iniş kuralında hata, her bir d gözlemi için hesaplanır (Denklem 2.16).

$$E_d(w) = \frac{1}{2}(t_d - o_d)^2 \quad (2.16)$$

İki yöntemin arasındaki farklar şu şekilde belirtilebilir.

- Standart eğimli iniş yönteminde ağırlıklar güncellenmeden önce tüm gözlemlerin hataları toplanırken stokastik eğimli iniş yönteminde ağırlıklar her bir eğitim gözlemin hatasına göre güncellenir.
- Standart eğimli iniş yönteminde ağırlık güncelleme adımı başına daha çok hesaplama gerektirir.
- Çoklu yerel minimum noktaların olduğu durumlarda, stokastik eğimli iniş tek bir gradyen yerine farklı gradyenler kullandığı için bu yerel minimumlara takılmaktan kurtulabilir.

2.3 Çok Katmanlı Algılayıcılar (Multi Layer Perceptrons)

Girdiler ve çıktılar arasındaki ilişki doğrusal olmayan yapıda ise tek katmanlı algılayıcılar ile öğrenme gerçekleştirmek mümkün değildir. Bu tür ilişkilerin öğrenilebilmesi için “Çok Katmanlı Algılayıcı (ÇKA)” gibi daha gelişmiş modeller kullanılır.







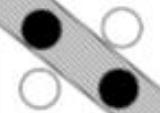




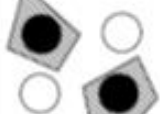



2.3.1 Çok Katmanlı Algılayıcıların Gösterim Yeteneği

Aşağıda belirtilen üç genel fonksiyon sınıfının ileri yönelimli çok katmanlı ağlar ile temsil edilebileceği bilinmektedir (Mitchell, 1997: 105).

1. **Boolean (mantıksal) Fonksiyonlar:** Tüm mantıksal fonksiyonlar iki katmanlı bir ağ ile tam olarak gösterilebilir. En kötü durumda, gizli katmandaki birimlerin sayısı ağdaki girdi sayılarına göre üssel olarak artmaktadır.
2. **Sürekli Fonksiyonlar:** Her sınırlı ve sürekli fonksiyon, küçük bir keyfi hata payıyla, gizli katmanında sigmoid birimleri ve çıktı katmanında doğrusal birimleri kullanan iki katmanlı bir ağ kullanılarak yaklaşık olarak (Cybenko, 1989; Hornik vd., 1989). Gizli katmandaki birimlerin sayısı yaklaşık olarak istenen fonksiyona bağlıdır.
3. **Keyfi Fonksiyonlar:** (Cybenko, 1988) tarafından ispatlanan teoreme göre herhangi bir fonksiyon, üç katmanlı bir ağ kullanılarak belli bir kesinlikle yaklaşık olarak. Çıktı katmanı doğrusal, iki gizli katman ise sigmoid birimlerden oluşurken genel olarak her bir katmandaki birim sayısı bilinmemektedir.

2.3.2 Gizli Birimlerin Etkisi

Çok katmanlı ağların önemli özelliklerinden birisi faydalı olabilecek öznelikleri gizli katmanlarında otomatik olarak keşfedilebilmeleridir (Venugopal & Baets, 1994: 33). Sadece ağ tasarımcısının önceden belirlediği özellikleri kullanmaya zorlanan öğrenme yöntemlerinin aksine bu özellik, açıkça belirtilmemiş özneliklerin keşfedilebilmesi için önemli ölçüde esneklik sağlar. Ağda daha fazla katman kullanıldıkça daha karmaşık karar yüzeyleri elde edilebilir (Şekil 15).

Yapı	Karar Sınırları	XOR Problemi	Sınıfların Kesişim Bölgeleri	Genel Alan Şekilleri	En Genel Alan Şekilleri
 Tek katman	Hiperdüzlem tarafından sınırlanmış yarım düzlem				
 İki katman	Keyfi (Karmaşıklık gizli birim sayısı ile sınırlı)				
 Üç katman	Keyfi (Karmaşıklık gizli birim sayısı ile sınırlı)				

Şekil 15. İki Boyutlu Girdi Uzayında Gizli Birimlerin Etkisi (Jain vd., 1996: 32)

(Tablo 9), eğitilebilir ağırlık katmanlarının sayısına göre ağların sınıflandırabildikleri kümeleri göstermektedir (Kriesel, 2007: 86).

Tablo 9. Katman Sayısına Göre Ağların Sınıflandırabildikleri Kümeler

Eğitilebilir Katman Sayısı	Sınıflandırılabilen Kümeler
1	Hiperdüzlem
2	Konveks Çokgen
3	Tüm Kümeler
4	Tüm Kümeler (Herhangi bir avantaj yok)

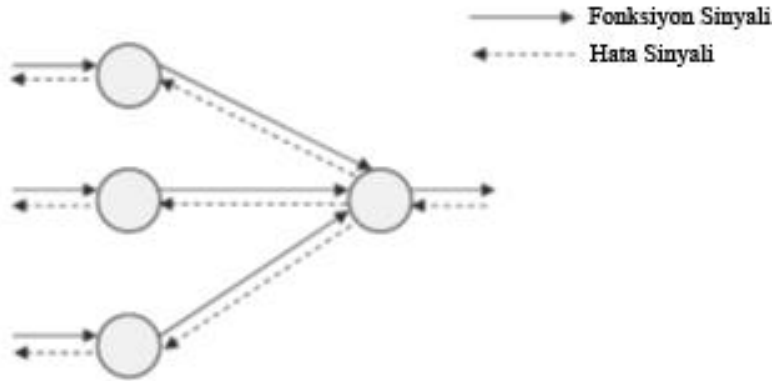
Gizli katmandaki doğrusal olmayan etkinlik fonksiyonları, ÇKA'nın "evrensel yaklaşan (*universal approximator*)" olmasını sağlar. Bu nedenle etkinlik fonksiyonlarının doğrusal olmaması gösterim problemini, türevlenebilir olmaları da öğrenme problemini çözer (Samarasinghe, 2006: 70).

n adet girdi birimi, H adet gizli katman birimi, K adet çıktı birimi olan bir çok katmanlı algılayıcının birinci katmanında $H * (n + 1)$, ikinci katmanında $K * (H + 1)$ sayıda ağırlık vardır. Bu yüzden hem zaman hem de bellek karmaşıklığı $\mathcal{O}(H * (K + n))$ olur. Dönem sayısı e ile gösterilirse, öğrenme zaman karmaşıklığı $\mathcal{O}(e * H * (K + n))$ dir. Bir uygulamada n ve K değerleri önceden bellidir. Karmaşıklığı ayarlamak için kullanılacak tek parametre gizli birim sayısıdır (Alpaydın, 2011: 217).

2.3.3 Çok Katmanlı Ağlarda Öğrenme

Çok katmanlı algılayıcılarda iki tür sinyal vardır (Şekil 16).

1. **Fonksiyon sinyali:** Fonksiyon sinyali, girdi katmanından girer ağda ilerler ve çıktı katmanından çıktı sinyali olarak çıkar. Girdilerin ve ağırlıkların bir fonksiyonu olarak hesaplandığı için bu isim verilmiştir.
2. **Hata sinyali:** Çıktı birimlerinde oluşturulur ve ağda geriye doğru yayılır. Çıktı birimlerinde bir hata fonksiyonu kullanılarak hesaplandığı için bu isim verilmiştir (Haykin, 1999: 181-182).



Şekil 16. Çok Katmanlı Algılayıcılarda Fonksiyon ve Hata Sinyali.

Gizli katman birimleri için beklenen bir değer yoktur. Bir gizli birimin hata sinyali, bu gizli birime bağlı tüm çıktı birimlerindeki hata sinyalleri kullanılarak yinelemeli (rekürsif) olarak belirlenir.

2.3.4 Geri Yayılım Algoritması

Geri yayılım algoritması, çok katmanlı ağlarda ağırlıkları öğrenmek için kullanılır. Hesaplama karmaşıklığı ağdaki ağırlık sayısına göre doğrusal olarak değiştiği için hesaplama yönünden etkindir. Bu algoritmanın geliştirilmesiyle ileri yönelimli çok katmanlı ağlar en yaygın kullanılan mimari haline getirmiştir (Yegnanarayana, 2005: 123).

Geri yayılım algoritmasında eğimli iniş yönteminden faydalanılır. Tek bir çıktı birimi yerine çoklu çıktı birimleri göz önüne alındığında tüm çıktı birimlerindeki hatalar toplamı (Denklem 2.17) de gösterildiği gibi bulunur.

$$E(w) = \sum_{d \in D} \sum_{\substack{k \in \text{Çıktı} \\ \text{birimleri}}} (t_{kd} - o_{kd})^2 \quad (2.17)$$

Çok katmanlı algılayıcılarda birçok yerel minimum olabilmektedir. Eğimli iniş, sadece yerel bir minimum noktaya yakınsamayı garantiler. Global minimum hatayı verecek noktaya yakınsamayı garantilemez. Bu engele rağmen birçok gerçek uygulamada geri yayılım algoritmasının üstün sonuçlar verdiği görülmektedir (Rumelhart vd., 1986: 535).

Algoritma, başlangıçta belirlenen gizli birim sayısına göre bir ağ oluşturur. Tüm ağ ağırlıklarına rastgele küçük değerler atanır. Ağ yapısı sabitlendikten sonra, tüm eğitim gözlemleri sırasıyla ağa sunulur.

Ağdaki her bir birime bir indis atanmıştır. Kullanılan notasyonlar şu şekildedir:

x_{ji} : i biriminden j birimine gelen girdi.

w_{ji} : i biriminden j birimine doğru olan ağırlık.

δ_j : j birimiyle ilişkilendirilen hata terimi.

Geri yayılım algoritmasında, delta öğrenme kuralında olduğu gibi, her bir ağırlık, öğrenme katsayısı (η), ağırlığın uygulandığı girdi (x_{ji}) ve birimin çıktısında oluşan hata ile orantılı olarak güncellenir. Fark ise; delta kuralında hatayı gösteren ($t - o$) teriminin daha karmaşık bir hata terimi olan δ_j ile değiştirilmiş olmasıdır (Mitchell, 1997: 99).

Her bir k çıktı birimi için hesaplanan hata terimi δ_k ile gösterilir. Ağda etkinlik fonksiyonu olarak sigmoid fonksiyonu kullanıldığı varsayıldığında, çıktı katmanındaki birimlerin hata terimleri, delta öğrenme kuralındaki ($t_k - o_k$) hata değerinin sigmoid fonksiyonunun türevi olan $o_k(1 - o_k)$ terimi ile çarpılmış halidir.

Gizli katman birimlerinin hata terimleri (δ_h) de benzer biçimdedir. Fakat eğitim örnekleri sadece çıktı katmanları için beklenen değerleri (t_k) verdikleri için, gizli birimlerin hatalarını doğrudan gösterecek bir beklenen değer yoktur (Yeung vd., 2009: 10). Bunun yerine h gizli katman birimleri için δ_h hata terimleri, her bir gizli katman biriminin bağlı olduğu çıktı birimlerinin hata terimlerinin (δ_k) aradaki ağırlıklar (w_{kh}) ile ağırlıklandırılmış toplamları kullanılarak hesaplanır. Bu ağırlık, h gizli biriminin k çıktı biriminde oluşan hatadaki sorumluluğunu gösterir.

Kısaca; her bir k çıktı birimi için hata terimi (δ_k) hesaplanır.

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k) \quad (2.18)$$

Daha sonra, her bir gizli katman birimi (h) için hata terimi (δ_h) hesaplanır.

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in \text{Çıktı}} w_{kh} \delta_k \quad (2.19)$$

2.3.4.1 Geri Yayılım Algoritmasının Elde Edilmesi

Bir ağırlık güncelleme yönteminin, güncel ağırlığın değiştirileceği yönü ve bu yönde ne kadar mesafe alınacağını belirlemesi gerekir. Geri yayılım algoritmasında

yön eğimin (gradyen) negatifi alınarak bulunurken, mesafe öğrenme katsayısı kullanılarak belirlenir.

Stokastik eğimli iniş yönteminde her bir d eğitim gözlemi için w_{ji} ağırlıkları

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \quad (2.20)$$

kuralına göre güncellenir. E_d , herhangi bir d gözleminin tüm çıktı birimleri üzerinde oluşturduğu toplam hata olup (Denklem 2.21) de gösterildiği gibi yazılabilir.

$$E_d(w) = \frac{1}{2} \sum_{\substack{k \in \text{Çıktı} \\ \text{birimleri}}} (t_k - o_k)^2 \quad (2.21)$$

Geri yayılım algoritmasında stokastik eğimli iniş yönteminin çalışmasını takip edebilmek için kullanılacak notasyonun tamamı aşağıda gösterilmiştir.

- x_{ji} = j birime gelen i . girdi.
- w_{ji} = j birimi ile i birimi arasındaki ağırlık.
- δ_j : j birimiyle ilişkilendirilen hata terimi.
- $net_j = \sum_i w_{ji}x_{ji}$ (j birime gelen net girdi).
- o_j = j birimin çıktısı.
- t_j = j birimin beklenen çıktısı.
- σ = sigmoid fonksiyonu.
- Çıktılar = ağın son (çıkış) katmanındaki birimlerin kümesi.
- $akış(j)$ = girdi olarak j biriminin çıktısını kullanan birimlerin kümesi.

(2.20) denkleminde görülen ağırlık güncelleme kuralının uygulanabilmesi için $\frac{\partial E_d}{\partial w_{ji}}$ teriminin hesaplanması gerekir. w_{ji} ağırlığı, ağın geri kalan kısmını sadece

net_j yoluyla etkileyebilir. Bu nedenle, zincir kuralı kullanılarak bu terim (Denklem 2.22) gibi yazılabilir.

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} x_{ji} \quad (2.22)$$

(2.22) denklemindeki $\frac{\partial E_d}{\partial net_j}$ için uygun bir ifadenin bulunması gerekir. Bunun için j 'nin çıktı birimi veya gizli birim olduğu iki durumun ele alınması gerekir.

Çıktı birimler için güncelleme kuralı: w_{ji} , ağıın geri kalanını sadece net_j yoluyla etkilediği gibi, net_j de ağı sadece o_j yoluyla etkileyebilir. Dolayısıyla zincir kuralı kullanılarak

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j} \quad (2.23)$$

bulunur. (2.23) denkleminde ilk terim olan $\frac{\partial E_d}{\partial o_j}$ ele alınırsa

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in \text{Çıktılar}} (t_k - o_k)^2 \quad (2.24)$$

elde edilir. $k = j$ olduğu durumlar haricinde $\frac{\partial}{\partial o_j} (t_k - o_k)^2$ değerleri tüm k çıktı birimleri için 0 değerini alacağından, çıktı birimleri üzerindeki toplama işlemi kaldırılır ve $k = j$ ataması yapılır. Böylece

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} (t_j - o_j)^2$$

$$\begin{aligned}
&= \frac{1}{2} 2(t_j - o_j) \frac{\partial(t_j - o_j)}{\partial o_j} \\
&= -(t_j - o_j) \quad (2.25)
\end{aligned}$$

eşitliği elde edilir. Bu aşamada (2.23) denklemindeki ikinci terimin daha açık bir ifadesinin bulunması gerekir. Gizli katmanda etkinlik fonksiyonu olarak sigmoid fonksiyonu kullanıldığı varsayılmaktadır. Bu durumda, $o_j = \sigma(net_j)$ olduğundan $\frac{\partial o_j}{\partial net_j}$ terimi sigmoid fonksiyonunun türevidir. Kısaca

$$\begin{aligned}
\frac{\partial o_j}{\partial net_j} &= \frac{\partial \sigma(net_j)}{\partial net_j} \\
&= o_j(1 - o_j) \quad (2.26)
\end{aligned}$$

elde edilir. (2.25) ve (2.26) ifadeleri (2.23) de yerine yazılırsa sonuç olarak

$$\frac{\partial E_d}{\partial net_j} = -(t_j - o_j)o_j(1 - o_j) \quad (2.27)$$

bulunur. (2.27) denkleminde gösterilen bu sonuç (2.20) ve (2.22) denklemleri ile birleştirildiğinde, çıktı birimleri için stokastik eğimli iniş kuralı (Denklem 2.28) elde edilmiş olur.

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \eta(t_j - o_j)o_j(1 - o_j)x_{ji} \quad (2.28)$$

Gizli birimler için eğitim kuralı: j biriminin gizli katmanda yer aldığı durumlarda, w_{ji} ağırlıklarının bulunması için, bu ağırlıkların çıktıyı ve dolayısıyla E_d hatasını etkileyebileceği dolaylı yolların dikkate alınması gerekir. Bu sebeple j biriminin akış yönündeki tüm birimlerin (doğrudan girdilerinin içinde j biriminin çıktısı da olanların) kümesinin gösterilmesi gerekir. Bu küme $akış(j)$ olarak gösterilecektir. net_j , ağ çıktısını dolayısıyla E_d hatasını sadece $akış(j)$ yoluyla

etkileyebilir. Bu açıklamalar matematiksel olarak yazıldığında (Denklem 2.29) elde edilir.

$$\begin{aligned}
\frac{\partial E_d}{\partial net_j} &= \sum_{k \in akış(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} \\
&= \sum_{k \in akış(j)} -\delta_k \frac{\partial net_k}{\partial net_j} \\
&= \sum_{k \in akış(j)} -\delta_k \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \\
&= \sum_{k \in akış(j)} -\delta_k w_{kj} \frac{\partial o_j}{\partial net_j} \\
&= \sum_{k \in akış(j)} -\delta_k w_{kj} o_j (1 - o_j) \quad (2.29)
\end{aligned}$$

δ_j , $-\frac{\partial E_d}{\partial net_j}$ değerini göstermek üzere (2.29) denklemi yeniden düzenlenirse

$$\delta_j = o_j (1 - o_j) \sum_{k \in akış(j)} \delta_k w_{kj}$$

$$\Delta w_{ji} = \eta \delta_j x_{ji} \quad (2.30)$$

gizli katman birimleri için stokastik eğimli iniş kuralı (Denklem 2.30) elde edilmiş olur.

2.3.4.2 Geri Yayılım Algoritmasının Genelleştirilmiş Hali

Geri yayılım algoritması sadece tek bir gizli katmana sahip ağlarda değil, keyfi sayıda gizli katmana sahip olan tüm ileri yönelimli ağlarda kullanılabilir. Ağırlık güncelleme kuralı aynı kalmakla birlikte algoritmada yapılması gereken değişiklik, hata terimlerinin hesaplanması ile ilgili olan kısımlardır. L katmanındaki herhangi bir r biriminin hata terimi (δ_r), kendisinden sonra gelen $L + 1$ katmanındaki hata terimleri (δ_s) kullanılarak

$$\delta_r = o_r(1 - o_r) \sum_{\substack{s \in L+1 \\ \text{katmanı}}} w_{sr} \delta_s \quad (2.31)$$

biçiminde hesaplanır.

2.3.4.3 Geri Yayılım Algoritmasında Dikkat Edilmesi Gereken Konular

Ağırlık güncellemeleri ağırlık vektörünün bulunduğu noktayı başka bir noktaya kaydırır. Eğer ağırlıklara göre alınan kısmi türevler çok küçük genlikte ve ağırlık vektörü optimal noktaya çok uzakta ise öğrenme işlemi çok uzun zaman alabilir (Rojas, 2006: 192). Öğrenme süresi önemli bir parametre olduğundan eğimli iniş yerine “Eşlenik Eğilim (Conjugate Gradient)”, “Newton-Raphson” gibi ikinci dereceden türev almaya dayalı optimizasyon yöntemleri de kullanılabilir.

Geri yayılım, en yaygın ÇKA öğrenme algoritması olmakla birlikte daha özelleşmiş görevler için farklı algoritmalar da geliştirilmiştir. Örneğin, geri yönelimli ağlarda yönlü çevrimler bulunduğundan bu tür ağların eğitimlerinde ağın hem yapısını hem de ağırlıklarını değiştiren daha farklı algoritmalar kullanılır (Mitchell, 1997: 123).

Geri yayılım algoritmasının kullanımı sırasında dikkat edilmesi gereken konular aşağıda sıralanmıştır.

2.3.4.3.1 Yerel Minimum

YSA tarafından temsil edilen karmaşık hata yüzeyleri üzerinde yerel minimuma takılmanın hangi durumlarda sorun oluşturacağını öngörebilecek kesin bir yöntem henüz bilinmemektedir. Bu problemin üstesinden gelmek için yaygın olarak “sezgisel (heuristics)” teknikler kullanılır (Mitchell, 1997: 104). Sezgisel tekniklerin her biri, yerel minimumdan kurtulmak için çeşitli öneriler sunar. Bu öneriler sayesinde global minimuma ulaşma ihtimali artar. Benzetilmiş tavlama ve genetik algoritmalar en sık başvurulan sezgisel yöntemler olmakla birlikte, parçacık sürü

optimizasyonu, tabu arama ve karınca kolonileri optimizasyonu teknikleri de ağıdaki ağırlıkların belirlenmesinde kullanılmaktadır (Cura, 2008: 28).

2.3.4.3.2 İlk Ağırlıklar

Global minimumun bilinmemesinin sebebi hata yüzeyinin tamamının bilinmemesidir. Sadece, bulunulan konunun komşuluğunda hesaplamalar yapılabilir. Hangi minimum noktada durulacağı başlangıç konumuna bağlıdır. Global minimuma yakın bir noktadan aramaya başlanırsa, bu global minimum noktasına ulaşma olasılığı yüksektir. Eğer yerel bir minimum noktasının yakınından aramaya başlanmış ise bu kez yerel minimuma ulaşıldığında arama biter. Global minimuma ulaşmak için başlangıç konumları farklı olan ağların eğitilmesi yaygındır.

Öğrenme sürecinde çıktı katmanından uzaklaştıkça geri yayılım algoritması yavaşlar. Bu nedenle, girdi katmanına yakın katmanlarda daha büyük ağırlıklar vermek gerekebilir (Kriesel, 2007: 92).

2.3.4.3.3 Öğrenme Katsayısı

Eğer algoritma başlangıcında öğrenme katsayısı küçük seçilmişse ağı yakınsaması uzun sürecektir. Bu sorunun çözümü için büyük öğrenme katsayısı kullanılması ise optimal noktanın atlanmasına sebep olabilir (Samarasinghe, 2006: 131).

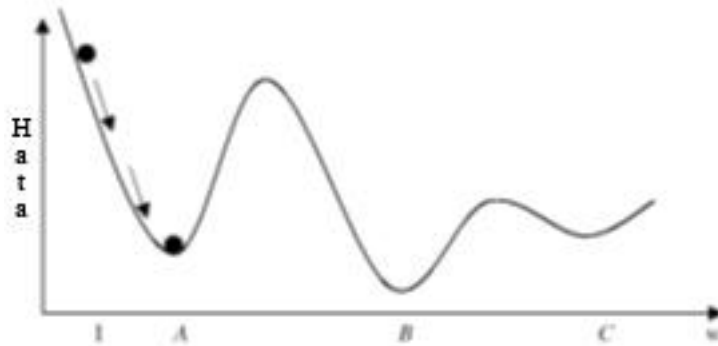
Çözümlerden birisi, öğrenme katsayısının eğitim ilerledikçe değerinin değişmesine izin vermektir. Eğitimin başında, öğrenme katsayısına optimal çözümün yakınına hızlıca yaklaşması amacıyla biraz büyük bir değer verilir. Bu yaklaşıma göre ağırlıklarda büyük değişimler sadece eğitimin başında rastgele olarak atanan ağırlıklar üzerinde yapılmalıdır. Daha sonra, ağı yakınsamaya başladıkça öğrenme katsayısı yavaş yavaş küçültülerek minimum noktayı aşması engellenmiş olur. Eğitim ilerledikçe ağırlıklar üzerinde hala büyük değişiklikler yapılıyorsa bir sorun var demektir (Marsland, 2009: 62).

2.3.4.3.4 Momentum Terimi

Geri yayılım algoritmasının başka önemli bir sorunu ani sıçramalardır. Ağırlık uzayında oluşan yüzey, gürültüden veya başka faktörlerden dolayı küçük genlikli iniş çıkışlar içerebilir. Bu iniş çıkışlar, kısmi türevlerin anlık değerlerinin hesaplanması nedeniyle parametre vektörünün optimal noktaya çok yakın olduğu durumlarda dahi türev değerlerinin yüksek genliklere ulaşmasına neden olabilirler. Bu tür ani sıçramalar, güncelleme kuralında “*momentum terimi*” olarak bilinen bir terimin kullanılması ile önlenir (Efe & Kaynak, 2000: 20).

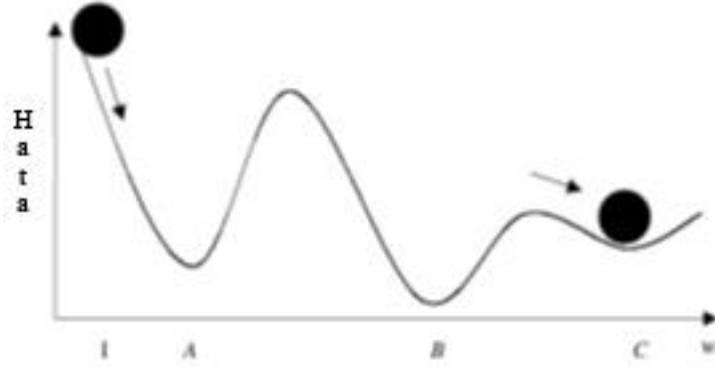
Momentum terimi, ağırlık güncellemelerinin aynı yönde olmasını destekleyerek optimal nokta yakınında meydana gelebilecek salınımları azaltır. Algoritmanın ilk aşamalarında ağırlıkların optimal yakınlıklarına ulaşma oranlarını artırarak da algoritmaya yardımcı olur.

Bu terimin etkisi (Şekil 17) ve (Şekil 18) de görülmektedir. Her iki şekilde de ağırlıklar I noktasında başlatılmıştır. Global minimum B noktasında, yerel minimumlar ise A ve C noktalarındadır. (Şekil 17) de, momentum terimi (α) için küçük bir değer (küçük bir top ile temsil edilen) verildiği varsayılmaktadır. Eğer bu top aşağı doğru yuvarlanırsa, ilk tepe noktasını aşamayıp çukur noktada takılabilir. Yani, küçük momentum değeri global minimum olan B noktası yerine yerel minimum olan A noktasını çözüm olarak verecektir (Larose, 2005: 141).



Şekil 17. Küçük Momentum Değerinin Etkisi

(Şekil 18) de daha büyük bir momentum terimi (büyük bir top şeklinde gösterilen) kullanılmaktadır. Eğer bu büyük top yuvarlanırsa hızlanarak B noktasındaki global minimumu aşarak C noktasındaki yerel minimuma ulaşacaktır.



Şekil 18. Büyük Momentum Değerinin Etkisi.

Geri yayılım algoritması, momentum teriminin (Denklem 2.32) gösterildiği gibi eklenmesiyle daha güçlü hale gelmiştir (Rojas, 1996: 187).

$$\Delta w_{yeni} = -\eta \frac{\partial E}{\partial w} + \alpha \Delta w_{önceki} \quad (2.32)$$

Yüksek α değerleri, güncel ağırlığın değişiminin (Δw_{yeni}) önceki ağırlık değişimi yönünde olmasına etki edecektir. Geri yayılım algoritmasına momentum teriminin eklenmesiyle, ağırlık değişiminin önceki tüm ağırlık değişimlerinin üssel ortalaması (Denklem 2.33) haline geldiği gösterilmiştir.

$$\Delta w_{yeni} = -\eta \sum_{p=0}^{\infty} \alpha^p \frac{\partial E}{\partial w_{yeni-p}} \quad (2.33)$$

(2.33) denklemi incelendiğinde, daha yakın güncellemelerin daha büyük etkileri olduğu görülmektedir. Büyük α değerleri, algoritmanın güncelleme geçmişindeki daha çok terimi hatırlamasına izin verir. Küçük α değerleri ise önceki

güncellemelerin etkilerini azaltır. En iyi sonuca ulaşmak için farklı α değerlerini denemek gerekmektedir. Momentum katsayısı genelde 0.5 ve 1.0 arası bir değer alır.

Momentum terimi öğrenme sırasında ağırlık vektörünün izlediği yolu düzleştirdiği için yararlıdır. Fakat önceki güncelleme değerlerini saklamak için ek bellek gerektirmektedir (Alpaydın, 2011: 217).

2.3.4.3.5 Aşırı Öğrenme

Eğitim sonucunda düşük hata elde edilmesi ağırlık her zaman sağlıklı çalıştığı anlamına gelmez. Hatta sifıra yakın toplam hata, aşırı öğrenme göstergesi olduğundan istenmeyen bir durumdur. Çünkü böyle durumlarda YSA eğitim kümesine aşırı bağımlı hale gelmiş olmaktadır (Cura, 2008: 29). Eğitim çok uzun tutulursa fonksiyonel ilişki ile birlikte gürültü de öğrenilir. Böylece oluşan model gereğinden fazla karmaşık olacağı için genelleme yeteneği düşük olacaktır.

Aşırı öğrenmenin ilk tekrarlarında değil de sonraki tekrarlarında oluşmasının sebebi şöyle açıklanabilir. Başlangıçta rastgele küçük değerler verildiğinden, tüm ağırlıklar birbirine yakındır. Bu nedenle, sadece çok düzgün karar yüzeyleri oluşturulabilir. Eğitimin ilerlemesiyle bazı ağırlıklar eğitim kümesinin hatasını azaltmak için artmaya başlar. Çok sayıda tekrar yapıldığında ise, algoritma genellikle eğitim verisindeki gürültüyü veya belirli bir eğitim gözleminin belirli bir karakteristiğini modelleyerek olması gerekenden daha karmaşık karar yüzeyleri oluşturur (Mitchell, 1997: 111).

Çok katmanlı ağlarda genelleme üç faktörden etkilenir (Haykin, 1999: 228).

- Eğitim kümesinin büyüklüğü ve problemi ne kadar iyi temsil ettiği
- Ağın mimarisi
- Problemin karmaşıklığı

Problemin karmaşıklığı için yapılacak bir şey yoktur. Fakat ilk iki faktör için:

- Ağ mimarisi sabit tutularak iyi genelleme için ihtiyaç duyulan eğitim kümesi büyüklüğü.
- Eğitim kümesi büyüklüğü sabit tutularak iyi genelleme sağlayacak en iyi mimari.

bulunmaya çalışılır.

Gizli birim sayısı gereğinden fazla olduğunda genelleme kötüleşir. Yanlılık-varyans ikilemi başka istatistiksel tahminleyiciler gibi çok katmanlı algılayıcılar için de geçerlidir (Alpaydın, 2011: 218). Basit ağların daha iyi genelleme yaptığı bir ipucu bilgisidir. Böyle bir ipucu, karmaşıklığı cezalandıran bir terimi hataya ekleyerek tanımlanabilir. Basit ağların karmaşık ağlardan her zaman daha iyi olduğu söylenemez. Fakat ikisi de aynı eğitim hatasına sahip iki ağ olduğunda, basit yani daha az parametresi olanın, geçerleme kümesinde daha iyi genelleme yapma olasılığının daha yüksek olduğu söylenebilir.

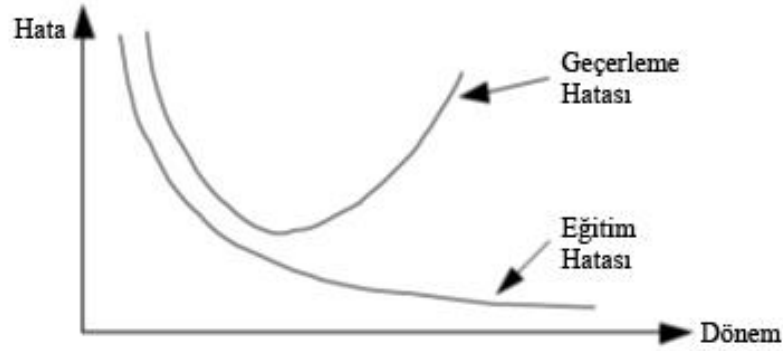
2.3.4.3.6 Durdurma Zamanı

Algoritmanın sonlandırma kriterine ulaşıncaya kadar eğitim kümesinin birçok kez ağda işlenmesi gerekebilir. Eğitim süresi göz önüne alındığında, eğitim kümesinin ağda en fazla kaç defa işleneceği veya algoritmanın maksimum çalışma süresi gibi sonlandırma koşulları kullanılabilir.

Başka bir seçenek ise hatada azalma olmuyorsa öğrenmeyi durdurmaktır. Eğer dönem başına hata değişim oranı yeterince küçük ise geri yayılım algoritmasının yakınsadığı varsayılır.

Öğrenmenin ne zaman durdurulacağı çapraz geçerlemeyle de belirlenebilir. Daha önceden eğitilen ağ, genelleme yeteneğini belirlemek için geçerleme kümesinde çalıştırılır. Ağ, birkaç dönem daha çalıştırılarak aynı işlem tekrarlanır.

Geçerleme kümesindeki hata değeri belirli bir düzeyden sonra tekrar artmaya başlar. Bu noktadan itibaren ağ, veriyi oluşturan fonksiyon hakkında öğrenmeyi bırakıp verideki gürültüyü öğrenmeye başlamış demektir (Şekil 19). Tam bu noktada eğitimin durdurulması gerekir. Buna “*erken durdurma (early stopping)*” denilir (Samarasinghe, 2006: 197-198).



Şekil 19. Aşırı Öğrenmenin Eğitim ve Geçerleme Hatası Üzerindeki Etkisi.

2.3.4.3.7 Ağırlık Sönümlenme

“*Ağırlık sönümlenme (Weight decay)*” yaklaşımında her bir tekrarda ağırlıklar belli bir oranda azaltılır. Bu, hata terimine ağırlıkların toplamına karşılık gelen bir ceza terimi eklemek anlamına gelir. Temel amaç, ağırlıkları küçük değerlerde tutarak gereğinden karmaşık karar yüzeylerinin oluşması engellenmektedir (Bishop, 1995: 338-339). Bu yaklaşımın kullanılmasıyla küçük ağırlık değerlerinden dolayı hata fonksiyonu daha az değişkenlik gösterdiğinden daha kolay ve kontrollü bir eğitime imkân verir.

Bir birim veya bağlantının gereksiz olup olmadığını anlamının yolu, ilgili birim veya bağlantı modelde yer alacak ve almayacak şekilde eğitim yaparak geçerleme hatasında fark olup olmadığına bakmaktır. Ama olası tüm birim-bağlantı bileşenleri için bunu yapmak çok maliyetlidir (Alpaydın, 2011: 222). Ağırlık sönümlenme işlemini yeterli bir süre çalıştırdıktan sonra, eğer bir birime bağlı ağırlık sıfır veya sıfıra yakınsa, bu birimi ve bağlı ağırlıkları kaldırılarak ağırlık

ezberlemesinin önüne geçilebilir. Buna “*budama (pruning)*” denir (Du & Swamy, 2006: 74).

2.4 Yapay Sinir Ağları Uygulama Adımları

Finansal öngörülerde kullanılacak bir YSA tasarımında genellikle aşağıda belirtilen adımlar kullanılmaktadır.

2.4.1 Değişken Seçimi

YSA tasarımının başarılı bir şekilde yapılabilmesi için çözülmek istenen problemin çok iyi anlaşılması gerekir. Tahmin edilmek istenen değişkeni etkileyecek girdi değişkenlerini belirlemek kritik bir öneme sahiptir. Bu değişkenleri belirlemek için konu ile ilgili teorik altyapıdan faydalanılmalıdır (Dreyfus, 2010: 26).

Örneğin, piyasa fiyatları tahmin edilirken bir veya birkaç piyasadan gelen teknik ve temel ekonomik girdilerden hangilerinin kullanılacağına karar verilmelidir. Teknik girdiler, bağımlı değişkenin gecikmeli değerleri veya bu gecikmeli değerlerden hesaplanan “*göstergeler*” olarak tanımlanırlar. Temel girdiler ise, bağımlı değişkeni etkilediği düşünülen ekonomik değişkenlerdir. En basit YSA, değişkenlerin gecikmeli değerlerini veya ilk farklarını girdi olarak kullanır. Daha yaygın bir yaklaşım ise, tahmin edilmek istenen piyasanın geçmiş fiyatlarına dayanan teknik göstergeleri kullanmaktır (Kaastra & Boyd, 1996: 219).

Yurtiçi ve uluslararası piyasalar arasında yakın bir ilişki olduğundan, tahmin performansını arttırmak için birbiriyle ilişkili piyasaların verilerinin de kullanılması tavsiye edilmektedir (Choudhry & Garg, 2008: 316).

Pratikte, eğitim kümesindeki gözlem sayısının en az ağın serbest parametreleri ile kabul edilebilen hata yüzdesinin çarpımı kadar olmasına dikkat edilir. Örneğin kabul edilebilecek hata oranı %10 ise gözlem sayısı, en az ağdaki ağırlıkların 10 katı olmalıdır (Haykin, 1999: 230).

Karmaşık fonksiyonlar öğrenilmek istendiğinde, öğrenmenin iyi olabilmesi için daha yoğun eğitim gözlemlerine ihtiyaç duyulur (Flood & Kartam, 1994: 142). Fakat boyut belasından dolayı, yüksek boyutlarda yoğun örnekler bulmak daha zordur.

2.4.2 Veri Toplama

Toplanan verilerde hata veya eksiklik olup olmadığının dikkatli bir şekilde kontrol edilmesi gerekir. Eksik veriler için farklı yöntemler uygulanmaktadır. Eksik veriye sahip olan tüm gözlemler eğitim kümesinden çıkarılabilir veya hareketli ortalamaları kullanılabilir. Girdi birimlerinden birini, eksik veri içeren gözlemleri belirtmek için ayırmak da kullanılan yöntemlerden biridir (Bishop, 1995: 301-302).

Teknik veya temel değişkenler girdi olarak kullanıldığında, zaman serisi boyunca bu değişkenlerin hesaplama yöntemlerinin değişmediğinden emin olmak gerekir.

2.4.3 Veri Önleme

Veri önleme, girdi ve çıktı değişkenlerinin gürültüyü azaltmak, önemli ilişkileri belirlemek, eğilimleri yakalamak, değişkenlerin dağılımını düzgünleştirmek amacıyla analiz edilmesi ve gerekli dönüştürmelerin yapılmasıdır (Du & Swamy, 2006: 48).

Ağ tasarımında verinin gösterimi kritik bir öneme sahiptir. Girdi ve çıktı değişkenleri nadiren elde edildikleri biçimleriyle ağa sunulurlar. En azından toplanan ham verinin etkinlik fonksiyonunun alt ve üst sınırları (genellikle $[0,1]$ veya $[-1,1]$) arasında ölçeklendirilmesi gerekir. Bu amaçla doğrusal ve ortalama/standart sapma ölçeklemeleri sıklıkla kullanılır (Bigus, 1996: 50).

Hem istatistiksel yöntemlerde hem de YSA tahmin modellerinde en sık olarak kullanılan veri dönüştürme yöntemi, değişkenlerin ilk farklarını veya doğal logaritmalarını almaktır. İlk farklar kullanılarak, veriden doğrusal eğilimin kaldırılması sağlanır. Logaritmik dönüşüm, hem küçük hem de büyük değerler alan

ve genelde sağı yatık olan dağılımlar üzerinde faydalıdır. Hem girdi hem çıktı verisinin basit veya üssel hareketli ortalamalar ile düzgünleştirilmesi de sıklıkla kullanılan dönüştürme yöntemidir (Witten & Eibe, 2005: 311).

Girdi değişkenlerinin oranları da yaygın olarak kullanılan veri dönüştürme biçimlerindedir. Oranlar, hem önemli ilişkileri belirtir hem de bağımsız değişkenler için daha az girdi birimi kullanılmasını sağlar.

Teknik analiz, yön hareketleri ve oynaklık (volatilite) hakkında birçok gösterge sunar. Değişken sayısını düşürmek ve periyodik olarak ağıın tekrar eğitilerek değişen piyasa koşullarına uyum sağlayabilmesi için, farklı teknik göstergelerin birlikte kullanılması faydalıdır (Choudhry & Garg, 2008: 317).

Test kümesinin büyüklüğü genellikle eğitim kümesinin %10'u ile %30'u arasında olur. Eğitilen ağıın performansı, geçерleme kümesi üzerinde kontrol edilir. Geçerlilik kümesinin büyüklüğü belirlenirken, eğitilmiş ağıı değerlendirmeye yetecek büyüklükte olması ile geriye eğitim ve test kümelerine yetecek kadar veri kalmasına dikkat edilmelidir.

Test kümesi, rasgele veya eğitim kümesindeki gözlemleri takip edecek şekilde oluşturulabilir. Test kümesi küçük olduğunda, sadece tek bir eğilimi gösteren gözlemlere sahip olabilir. Test kümesini rastgele oluşturmak, modelin performansını değerlendirirken sadece bu eğilime göre değerlendirilmesinin önüne geçmeyi sağlar. Test kümesini diğıer şekilde oluşturmanın avantajı ise, eski verilerden daha önemli olan yakın verileri içermesidir.

2.4.4 Ağ Parametrelerinin Belirlenmesi

Bir YSA oluşturmanın sonsuz sayıda yolu vardır. Girdi birimlerinin sayısı, belirlenecek ağ parametreleri içerisinde en kolay olanıdır. Çünkü bağımsız değişkenler üzerinde önışleme işlemleri yapıldıktan sonra, her bir bağımsız değişken bir girdi birimi ile temsil edilir.

YSA tasarlanırken bir veya en fazla iki gizli katmanlı bir modelle başlanması önerilir. Eğer iki gizli katmanlı bir ağ, farklı rastgele başlangıç değerleri verilerek ve farklı gizli birim sayılarıyla denendikten sonra, tatmin edici sonuçlar vermiyorsa üçüncü bir gizli katman eklemek yerine girdilerin sayılarında değişiklik yapılması gerekir. Hem teoride hem de finansal tahmin uygulamalarında iki gizli katmandan fazla gizli katman kullanmanın modelin performansına herhangi bir katkısının olmadığı belirtilmektedir

Önemli bir ağ parametresi olmasına rağmen, en uygun gizli birim sayısını veren bir formül yoktur. Genellikle deneme yanılma yoluyla gizli birimlerin sayısı belirlenmekle birlikte, literatürde pratik olarak kullanılacak öneriler bulunmaktadır. Bu önerilere göre gizli katman birim sayısı (Kaastra & Boyd, 1996: 225):

- Girdi birimi sayısı ile çıktı birim sayısının çarpımlarının karekökü kadar olmalıdır.
- Girdi birimlerinin %75 i kadar olmalıdır.
- Girdi birim sayısının 1.5-3 katı olmalıdır.
- Ağın test kümesi üzerindeki performansı bozulmaya başlayıncaya kadar gizli birim sayısı iki katına çıkarılmalıdır.

Farklı sayıda gizli katman birimine sahip ağları test ederken, diğer tüm parametrelerin sabit tutulması önemlidir. Aksi halde, diğer parametreler değiştirildiğinde farklı hata yüzeyleri elde edileceğinden, en uygun gizli katman birim sayısını belirleme işlemi daha karmaşık hale gelir.

En iyi ağ büyüklüğü bulunurken en sık kullanılan yöntem, farklı ağ modellerini öğrenme kümesinde eğitip geçерleme kümesinde en iyi olanını

seçmektir. Diğer bir yöntem ise böyle bir yapısal uyarlamayı öğrenme algoritmasına dahil etmektir. Bu, iki farklı biçimde yapılabilir (Alpaydın, 2011: 222):

- **Azalımlı yaklaşım:** Büyük bir ağla başlanır. Sonra birim ya da bağlantılar budanır.
- **Artımlı yaklaşım:** Küçük bir ağla başlanır. Performansı arttıracak biçimde yeni birim veya bağlantılar eklenir.

Etkinlik fonksiyonu olarak simetrik olmayan bir fonksiyon yerine simetrik olan bir fonksiyon kullanmak eğitim için gerekli adım sayısını azalttığından geri yayılım algoritmasının daha hızlı çalışmasını sağlar. Bunun için genellikle simetrik olmayan standart sigmoid fonksiyonunun yerine simetrik olan hiperbolik tanjant fonksiyonu kullanılır (Haykin, 1999: 201).

2.4.5 Ağın Eğitilmesi

Eğer hata fonksiyonu, performansta devamlı bir iyileşme göstermeden hızlı bir şekilde değişiyorsa, bu öğrenme katsayısının yüksek olduğunun bir göstergesidir. Hata fonksiyonu yavaş bir şekilde değişiyor fakat bu değişim performansta çok düşük bir iyileşmeye veya hiçbir iyileşmeye neden olmuyorsa, bu sefer öğrenme katsayısının düşük olduğu anlaşılır. Küçük öğrenme katsayısı öğrenme süresinin uzamasının sebeplerinden birisidir. Her iki durumda da; eğitim sırasında öğrenme katsayısının değiştirilmesi veya tüm ağırlıklara baştan rastgele ağırlıklar atayıp her yeni eğitimde öğrenme katsayısı değiştirilerek, en uygun öğrenme katsayısının bulunması gerekir. Salınımına sebep olmadan öğrenme katsayısını arttırmanın ve böylece öğrenmeyi hızlandırmanın yolu, öğrenme kuralında bir momentum terimi kullanmaktır.

2.4.6 Ağın Uygulanması

YSA'nın bir avantajı, periyodik olarak ağın tekrar eğitilmesiyle değişen ortam şartlarına uyum sağlayabilmesidir. Uygulamaya konulduktan sonra eğer periyodik olarak ağ tekrar eğitilmez ise performansında zamanla bir azalma görülür.

Fakat tekrar eğitim yapılırsa da modelde yer alan bağımsız değişkenler zaman içinde önemlerini yitirmeye başlarsa performansın aynı kalmasının bir garantisi yoktur. İyi bir model, eğitim sıklığına karşı sağlam olan ve eğitimin daha sık yapılmasıyla performansı gelişme gösteren modeldir (Bigus, 1996: 113).

ÜÇÜNCÜ BÖLÜM

3. DESTEK VEKTÖR MAKİNELERİ

Bu bölümde, “*Destek Vektör Makineleri (DVM)*” yöntemi ve İstatistiksel Öğrenme Teorisi’ne ait temel kavramlar matematiksel bir bakış açısıyla incelenmiştir.

Vladimir Vapnik ve arkadaşları tarafından önerilen DVM; sınıflandırma, regresyon ve yoğunluk tahmini gibi pek çok yaygın problemin çözümünde kullanılan güçlü bir yöntemdir (Press vd., 2007: 883).

DVM ile ilgili ilk makale 1964 yılında yayınlanmasına rağmen yöntem 1990’lı yıllara kadar çok ciddiye alınmamıştır. Bunun sebebi, o yıllarda makine öğrenmesi topluluğunda DVM’nin güçlü teorisine rağmen pratikte uygulanabilir bir yöntem olmadığına dair inanıştır. Yöntem, ancak rakam tanıma, bilgisayarla görme ve metin sınıflandırma vb. görevlerde diğer yöntemlere göre daha iyi sonuçlar verdiği görüldükten sonra ciddiye alınmaya başlanmıştır (Huang vd., 2006: 12).

DVM yönteminin temeli İstatistiksel Öğrenme Teorisi’ne dayanmaktadır. İstatistiksel Öğrenme Teorisi önceleri sadece verilen bir veri kümesine uyan fonksiyon tahmini probleminin teorik analizi için kullanılmıştır. Daha sonra bu teoriye dayanan DVM yöntemin önerilmesiyle teorik analizlerin yanında, yeni öğrenme algoritmalarının oluşturulmasında da kullanılabileceği görülmüştür (Vapnik, 1999: 988).

3.1 İstatistiksel Öğrenme Teorisi

İstatistiksel Öğrenme Teorisi, makine öğrenmesinin matematiksel altyapısını oluşturmaya çalışır. Teorinin cevaplamaya çalıştığı temel sorular şu şekilde sıralanabilir (Von Luxburg & Schölkopf, 2008: 2):

- Genel olarak hangi öğrenme görevleri bilgisayarlar tarafından yerine getirilebilir?
- Makine öğrenmesinin başarılı olabilmesi için hangi varsayımlar yapılmalıdır?
- Bir öğrenme algoritmasının başarılı olabilmesi için hangi temel özelliklere ihtiyacı vardır?
- Öğrenme algoritmasının sonuçları üzerinde hangi performans garantileri verilebilir?

İstatistiksel Öğrenme Teorisi, “*Vapnik-Chervonenkis (VC) Teorisi*” olarak da bilinir. Sınırlı sayıda gözlem kullanarak esnek tahminler yapmak için şu anda var olan en iyi teori olarak kabul edilmektedir. Öğrenme için önemli sonuçları olan matematiksel ispatları içerir. Öğrenme probleminin genel bir çerçevesini oluşturmak için gerekli tüm kavramlar bu teoride tanımlanmıştır.

Yapay sinir ağlarında model seçimi, belirlenen geçiş kümeleri üzerinde yapılırken, VC teorisinin temel hedefi belli bir küme üzerindeki ampirik hata yerine genelleme hatasını karakterize etmektir. Bunun için genelleme üzerinde olasılıksal sınırlar oluşturulur (Suykens vd., 2002: 44).

VC teorisi, sınırları belirlerken (Valiant, 1984) tarafından önerilen “*Olası Yaklaşık Öğrenme (Probably Approximately Correct – PAC)*” modelini genelleştirir. PAC öğrenmesi, ikili sınıflandırma sistemlerinde öğrenme ve genellemeyi incelemek için olasılıksal bir çerçeve oluşturur. Öğrenme algoritmasına 2 kontrol parametresi sağlanır. Bunlar;

1. **Hata parametresi** ($\epsilon \in (0,1]$): Yaklaşmada izin verilen hatayı belirtir.

2. **Güven parametresi** ($\delta \in (0,1]$): İyi bir yaklaşmanın olabilirliğini kontrol eder (Kearns & Vezirani, 1994: 10).

PAC öğrenmede, bilinmeyen bir olasılık dağılımından çekilmiş gözlemler verildiğinde, istenen $\delta \leq 1/2$ ve $\varepsilon > 0$ değerleri için, hatanın en az $1 - \delta$ olasılıkla en çok ε olması için gereken minimum örnek sayısı (n) bulunmak istenir. Buna göre, bir fonksiyon (hipotez) kullanıldığında verilen bir gözlemin yanlış sınıflandırma olasılığı, en az $1 - \delta$ güven seviyesinde, en çok ε olacaktır. δ değeri azaltılarak güven seviyesi istenilen ölçüde arttırılabilir, ya da ε değeri azaltılarak istenilen ölçüde az hata yapılması sağlanabilir. Gereken örnek sayısı, $1/\varepsilon$ ve $1/\delta$ terimlerinin yavaş artan, sırasıyla doğrusal ve logaritmik bir fonksiyonudur (Alpaydın, 2011: 21).

VC teorisi ilk olarak “*Ampirik Risk Minimizasyonu (ARM)*” tümevarım ilkesini teorik olarak incelemek için geliştirilmiştir. İyi genelleme için gereken şartlar belirlenmiş ve bu şartların düzgün yakınsama ile yakından ilgili olduğu gösterilmiştir. Bulunan sonuçlar, model karmaşıklığı ile eldeki bilgi (sınırlı sayıda gözlemden oluşan eğitim kümesi) arasındaki ödünleşimin (tradeoff) niceliksel olarak ifade edilmesini sağlamıştır.

VC teorisi temel olarak dört kısımdan oluşur (Vapnik, 1998: 49-50).

1. Ampirik Risk Minimizasyonu (ARM) ilkesinin tutarlılık koşullarının belirlenmesi.
2. Tutarlılık koşullarında öğrenme makinesinin genelleme sınırlarının bulunması.
3. Genelleme sınırlarına bağlı olarak sınırlı gözlemlerden tümevarımsal çıkarım yapmak için gerekli ilkelerin bulunması.
4. Bulunan tümevarım ilkelerini uygulayan algoritmaların oluşturulması.

3.1.1 Ampirik Risk Minimizasyonu (ARM) İlkesi

Bir z uzayında $z = (x, y)$, girdi-çıkıı çiftlerini gösterisin. Öğrenme probleminde, bilinmeyen fakat benzer bir $P(z)$ olasılık yoğunluk fonksiyonuna göre oluşturulmuş birbirinden bağımsız n adet gözleminden oluşan $Z_n = \{z_1, z_2, \dots, z_n\}$ eğitim kümesi ve $Q(z, \omega)$, $\omega \in \Omega$ kayıp fonksiyonları verilir. Öğrenmenin amacı, (Denklem 3.1) de gösterilen risk fonksiyoneli minimize eden $Q(z, \omega_0)$ fonksiyonunu bulmaktır (Cherkassky & Mulier, 2007: 102).

$$R(\omega) = \int Q(z, \omega)P(z)dz \quad (3.1)$$

(3.1) denkleminde $Q(z, \omega) = L(y, f(x, \omega))$ belli bir öğrenme problemine özgü kayıp fonksiyonları kümesini göstermektedir.

Risk fonksiyoneli, bilinmeyen $P(z)$ olasılık dağılımına bağılı olduğundan hesaplanamaz. ARM tümevarım ilkesinde, hesaplanamayan risk fonksiyonelinin yerine hesaplanabilen ampirik risk fonksiyoneli (Denklem 3.2) kullanılır. Ampirik riski minimize eden $Q(z, \omega^*)$ kayıp fonksiyonu bulunmaya çalışılır.

$$R_{amp}(\omega) = \sum_{i=1}^n Q(z_i, \omega) \quad (3.2)$$

Sınıflandırma problemi ele alınacak olursa, ampirik risk eğitim verisinin yanlış sınıflandırılma (eğitim hatası) olasılığı, beklenen (gerçek) risk ise bilinmeyen $P(z)$ olasılık dağılımı üzerindeki yanlış sınıflandırma olasılığıdır.

ARM, çok genel bir ilkedir. Regresyon tahmininde kullanılan en küçük kareler (least squares) veya yoğunluk tahmininde kullanılan en büyük olabilirlik (maximum likelihood) gibi klasik yöntemler, bu ilkenin belirli kayıp fonksiyonları için gerçekleştirilmiş halleridir (Cucker & Zhou, 2007: 50). Bu nedenle, ARM ilkesi ile ilgili herhangi bir teori, klasik yöntemlerde de geçerlidir.

$Q(z, \omega) = L(y, f(x, \omega))$ olduğundan öğrenme probleminin kayıp fonksiyonları veya yaklaşan fonksiyonlar cinsinden gösterilmesi arasında fark

yoktur. Fakat ampirik risk fonksiyonelinin, tutarlılık ve yakınsama koşullarını açıklamak için $Q(z, \omega)$ cinsinden gösterilmesi daha uygundur (Cherkassky & Mulier, 2007: 102).

3.1.2 ARM İlkesinin Tutarlılık Şartları

Herhangi bir tümevarım ilkesinin sağlaması gereken temel özelliklerden birisi, asimptotik tutarlılıktır. Bu özellik, ampirik riski minimize eden fonksiyonun değerinin, eğer gerçek risk doğrudan minimize edilebilseydi elde edilecek değere asimptotik olarak (eğitim gözlem sayısı arttıkça) yakınsamasını ifade eder (Schölkopf & Smola, 2001: 131).

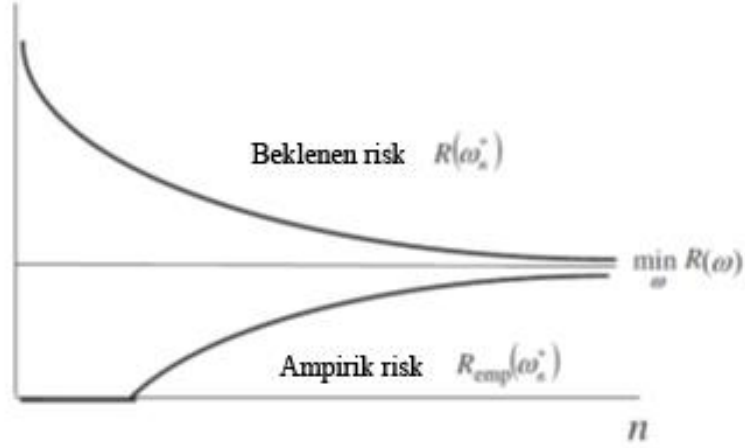
Bu noktada, akla “*amaç küçük (sınırlı) örneklem için algoritmalar oluşturmak ise neden bir tutarlılık teorisine ihtiyaç duyulur*” sorusu gelmektedir. Bu sorunun cevabı VC teorisinin ARM ilkesinin yakınsaması için sadece yeter şartların değil gerek şartların da neler olduğunu belirtmesidir (Vapnik, 2000: 35).

$R_{amp}(\omega_n^*)$, n gözlemlili Z_n eğitim kümesi üzerindeki ampirik riski minimize eden $Q(z, \omega_n^*)$ kayıp fonksiyonu tarafından sağlanan ampirik riskin değerini, $R(\omega_n^*)$ ise aynı $Q(z, \omega_n^*)$ kayıp fonksiyonu için bilinmeyen gerçek riski gösterebilir. Z_n eğitim kümesinin rassal özelliğinden dolayı $R_{amp}(\omega_n^*)$ ve $R(\omega_n^*)$ değerleri de rassal diziler oluştururlar. Sezgisel olarak eğitim kümesi büyüklüğü (n) arttıkça bu rassal dizilerin aynı limit değerine yakınsamaları beklenir. Daha açık şekilde belirtilirse; ARM ilkesi, eğer $R_{amp}(\omega_n^*)$ ve $R(\omega_n^*)$ rassal dizileri, n sonsuza yaklaştıkça olasılıksal olarak aynı $R(\omega_0) = \min_{\omega} R(\omega)$ değerine yakınsıyorlarsa (Denklem 3.3) tutarlıdır denilir.

$$\begin{aligned} R(\omega_n^*) &\rightarrow R(\omega_0), \quad n \rightarrow \infty \text{ için} \\ R_{emp}(\omega_n^*) &\rightarrow R(\omega_0), \quad n \rightarrow \infty \text{ için} \end{aligned} \quad (3.3)$$

(Şekil 20) ARM ilkesinin tutarlılığı için hem gerçek riskin hem de ampirik riskin aynı minimum risk değerine yakınsaması gerektiğini göstermektedir. Verilen bir eğitim kümesi üzerinde bir öğrenme algoritması, her zaman ampirik riski minimize edebilecek fonksiyonu seçebilse bile bu fonksiyon gerçek riski minimize

edecek fonksiyon olmayabilir. Bu nedenle; ampirik riskin gerçek riskten küçük yani $R_{emp}(\omega_n^*) < R(\omega_n^*)$ olması beklenir. Diğer bir ifade ile bir eğitim kümesi üzerinde ARM ilkesi ile bulunan bir fonksiyon daima gerçek riski minimize eden fonksiyonun yanlı tahminidir (Cherkassky & Mulier, 2007: 103).



Şekil 20. Ampirik Risk Minimizasyonu (ARM) İlkesinin Tutarlılığı

Sınırlandırılmış kayıp fonksiyonları için, ARM ilkesi sadece ve sadece ampirik risk gerçek riske (Denklem 3.4) de gösterildiği gibi

$$\lim_{n \rightarrow \infty} P \left[\sup_{\omega} (R(\omega) - R_{amp}(\omega)) > \varepsilon \right] = 0, \quad \forall \varepsilon > 0 \quad (3.4)$$

biçiminde düzgün yakınsar ise tutarlıdır denilir. (Denklem 3.4) tutarlılık için gerek ve yeter şartları belirttiği için VC teorisinin “*anahtar teoremi*” olarak bilinmektedir. Teoremden ampirik hatanın küçük olması değil, sadece gerçek hatanın iyi bir tahmini olması gerektiği belirtilmektedir (Von Luxburg & Schölkopf, 2008: 6).

Düzgün yakınsama, tutarlılık koşullarının olası tüm yaklaşan fonksiyonlar için geçerli olmasını yani tutarlılığın fonksiyonlar kümesindeki tek bir fonksiyona bağlı olmamasını gerektirir. Bu gereksinime göre fonksiyonlar kümesinden en iyi fonksiyon çıkarılsa bile ARM ilkesi tutarlı olmaya devam etmelidir. Kısaca, anahtar teoreme göre tutarlılık ampirik risk ile gerçek risk arasındaki en büyük farka göre

yani “*en kötü duruma (worst case)*” göre belirlenir (Herbrich, 2001: 121). Ortalama durum analizine göre tutarlı bir öğrenme teorisi geliştirmek mümkün değildir.

Geniş fonksiyon kümeleri için $P[\sup_{\omega}(R(\omega) - R_{amp}(\omega)) > \varepsilon]$ olasılığı daha büyük olacağı için böyle fonksiyon kümelerinde tutarlılığı sağlamak küçük fonksiyon kümelerine göre daha zordur.

Anahtar teorem, teorik olarak ilgi çekici olmasına rağmen pratik kullanıma çok elverişli değildir. Çünkü bir öğrenme makinesi kullanılmak istenildiğinde her defasında bu yakınsama özelliğinin kontrol edilmesi gerekir. Dolayısıyla, pratik kullanıma daha uygun olan, öğrenme makinelerinin yani fonksiyon kümelerinin riskin düzgün yakınsamasını garanti eden özelliklerinin neler olduğu konusu önem kazanır (Schölkopf & Smola, 2001: 134).

3.1.3 Kapasite Ölçütleri

Sınıflandırma problemi için, sınırlı sayıda gözleme sahip bir eğitim kümesi üzerinde yapılacak öğrenmede en iyi genelleme performansı, bu eğitim kümesi üzerinde elde edilen doğru sınıflandırma oranı ile öğrenme makinesinin herhangi bir eğitim kümesini hatasız olarak öğrenme yeteneğini gösteren kapasite arasında doğru denge sağlanırsa başarılabilir. Çok yüksek kapasiteye sahip olan bir öğrenme makinesi, kendisine yeni bir ağaç gösterildiğinde ağacın yaprak sayısının daha önceden gördüğü ağaçların yaprak sayısından farklı olduğu için bunun bir ağaç olmadığını söyleyen kişiye benzetilebilir. Düşük kapasiteli bir öğrenme makinesi ise eğer gösterilen bir nesne yeşilse otomatik olarak bunun bir ağaç olduğunu kabul eden kişi üzerinden açıklanabilir. Her iki durumda da asıl amaç olan genelleme sağlanamaz. Bu dengenin nasıl kurulacağı teorik olarak VC teorisinde çok başarılı bir şekilde açıklanmıştır (Burges, 1998: 122).

3.1.3.1 Büyüme Fonksiyonu

İkili sınıflandırma problemlerinde kullanılan gösterge (indicator) fonksiyonları ele alındığında, her bir gösterge fonksiyonu eğitim kümesini iki parçaya (sınıfa) bölmeler. Bu bölünmelerin her birine “*dikotomi (dichotomy)*” denilir.

“*Büyüme (Growth) fonksiyonu*”, bir eğitim kümesi üzerinde bir fonksiyonlar kümesinden seçilen gösterge fonksiyonlarını kullanılarak elde edilen maksimum dikotomi sayısını gösterir. Maksimum dikotomi sayısının tüm örneklemelerden değil, sadece bir örneklemden elde edilmesi yeterli sayılır. n gözlemden oluşan bir kümenin olası tüm farklı ikili bölünmelerinin sayısı en çok 2^n olduğundan $G(n)$ büyüme fonksiyonu için üst sınır, $n \ln 2$ dir (Kecman, 2001: 144-145).

Büyüme fonksiyonu, VC teorisinde ARM ilkesinin tutarlılık ve yakınsama şartlarının belirlenmesinde rol oynar. Herhangi bir $n > n_0$ için, c pozitif bir sabit olmak üzere

$$P\left((R(\omega) - R(\omega^*)) < \varepsilon\right) = e^{-cn\varepsilon^2} \quad (3.5)$$

sınırı sağlanıyorsa asimptotik yakınsama oranı hızlıdır denilir (Cherkassky & Mulier, 2007: 106).

VC teorisine göre, dağılımdan bağımsız olarak ARM ilkesinin tutarlılığı ve hızlı yakınsaması için hem gerek hem yeter şart

$$\lim_{n \rightarrow \infty} \frac{G(n)}{n} = 0 \quad (3.6)$$

olmasıdır. Büyüme fonksiyonu bir olasılık dağılımına bağlı olmadığı için (3.6) denkleminde belirtilen koşul da dağılımdan bağımsızdır. Bu koşul aynı zamanda hızlı yakınsamayı da garanti eder (Schölkopf & Smola, 2001: 140).

3.1.3.2 VC Boyutu

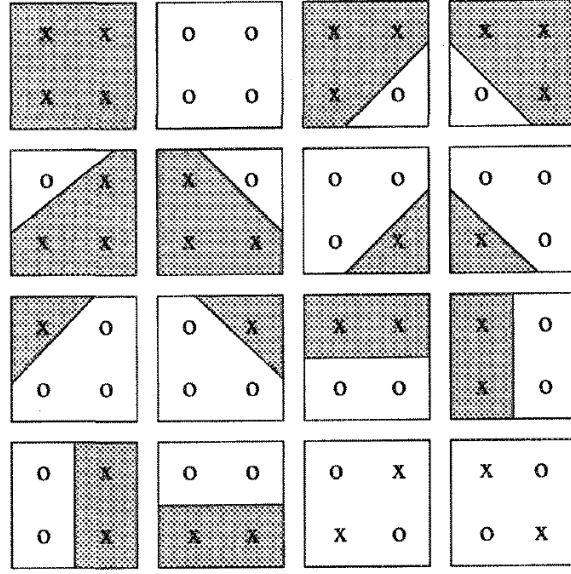
“*VC boyutu (VC dimension)*”, sınıflandırıcıların kapasitesinin bir ölçütüdür. Örneğin, ikili bir sınıflandırıcının sınıflar arasındaki sınırı ne kadar iyi modelleyebileceğini ölçer. VC boyutu büyük olduğunda, sınıflandırıcının kapasitesi daha yüksektir ve eğitim gözlemlerini sınıflarına göre daha iyi ayırt edebilmektedir. Genel olarak bir sınıflandırıcının kapasite arttıkça, eğitim kümesindeki gözlemleri daha iyi ayırabilmektedir (Hamel, 2009: 172).

Eğer n gözlemden oluşan eğitim kümesi bir fonksiyonlar kümesindeki fonksiyonlar ile 2^n farklı şekilde ayrılabilirse bu eğitim kümesinin ilgili fonksiyonlar kümesi tarafından “*ayrılabilirdiği (shatter)*” söylenir (Bousquet vd., 2004: 189).

VC boyutu, tüm olası farklı etiketlemeleri bir gösterge fonksiyonlar kümesi tarafından hatasız bir şekilde ayrılabilen en büyük gözlem sayısıdır (Kulkarni & Harman, 2011: 550). Yani bir fonksiyonlar kümesi eğer h adet gözlemi ayırabiliyor fakat $h + 1$ gözlemi ayıramıyorsa bu fonksiyon kümesinin VC boyutu h olur. Bu tanım sadece bir tane h elemanlı kümenin bulunmasını yeterli sayar. h elemanlı her kümenin ayrılabilme zorunluluğunu aramaz.

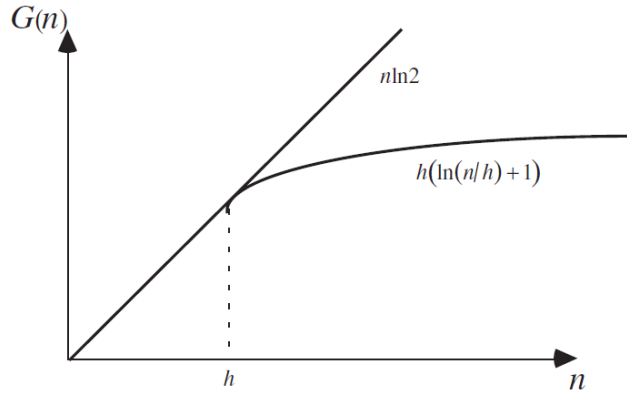
Teorem: \mathcal{R}^n uzayında m adet noktadan oluşan bir küme ele alınsın. Bu noktalardan herhangi biri orijin olarak seçilsin. Bu m nokta, ancak ve ancak geri kalan gözlemlerin vektörleri kendi aralarında doğrusal bağımsız iseler yönlü hiperdüzlemler tarafından ayrılabilirler. Bu teoreme dayanarak, \mathcal{R}^n uzayındaki hiperdüzlemler kümesinin VC boyutu $n + 1$ olarak belirlenebilir. Çünkü \mathcal{R}^n uzayında birisi orijin olarak seçildiğinde geri kalan n tanesi kendi aralarında daima doğrusal bağımsız olan $n + 1$ nokta seçilebilir (Burges, 1998: 125).

Örneğin 2 boyutlu uzayda, 4 nokta en çok $2^4 = 16$ farklı şekilde etiketlenebilir (Şekil 21). En sondaki iki durum doğrusal olarak ayrılamayan XOR problemini göstermektedir. Bu durumda 2 boyutlu uzayda, bir doğru ile en çok 3 nokta ayrılabilir. Bu sonuçtan yola çıkarak bu uzayda doğrusal sınıflandırıcıların VC boyutunun 3 olduğu açıkça görülmektedir. Doğrusal olmayan sınıflandırıcılar için ise VC boyutu daha büyük olabilir.



Şekil 21. Doğrusal Sınıflandırıcıların VC Boyutu (Suykens vd., 2002: 47)

Vapnik ve Chervonenkis tarafından büyüme fonksiyonunun ya doğrusal ya da gözlem sayısının (n) logaritmik bir fonksiyonuyla sınırlandırıldığı gösterilmiştir (Şekil 22).



Şekil 22. Büyüme Fonksiyonunun Davranışı

Büyümenin yavaşlamaya başladığı, yani $n = h$ olduğu noktaya “VC boyutu” denilir ve h ile gösterilir. Eğer VC boyutu sonlu ise, büyüme fonksiyonu yeterince

büyük eğitim kümeleri için doğrusal olarak büyümez. (Denklem 3.7) de gösterildiği gibi sınırlandırılmıştır (Kecman, 2001:145).

$$G(n) \leq h \left(\ln \frac{n}{h} + 1 \right) \quad (3.7)$$

VC boyutu (h) değerinin sonlu olması ARM ilkesine dayalı öğrenmenin dağılımdan bağımsız olarak tutarlılığı ve hızlı yakınsaması için gerek ve yeter şartları sağlar (Brunato & Battiti, 2005: 829).

VC boyutu sonsuz olduğunda ise büyüme fonksiyonu doğrusaldır ve daima $n \ln 2$ değerini alır. Bu durumda, n gözleme sahip eğitim kümesi bir fonksiyonlar kümesi ile olası tüm (2^n) farklı şekilde ayrılabilse de (3.6) denkleminde yola çıkarak geçerli bir genellemenin mümkün olmadığı görülebilir

Sınırlı örneklerden öğrenme (genelleme), sadece gözlem sayısı (n), sonlu VC boyutunu (h) aştığında gerçekleşir. Bu sonuçlara göre, öğrenme olasılık dağılımı bilinmeden de mümkündür.

Sınıflandırma problemleri için, gösterge kayıp fonksiyonunun VC boyutu, yaklaşan fonksiyonlar kümesinin VC boyutuna eşittir. Regresyon problemlerinde ise h_f yaklaşan fonksiyonlar kümesinin VC boyutunu, h hata kareli kayıp fonksiyonlarının VC boyutunu göstermek üzere Vapnik tarafından bir c sabiti için $h_f \leq h \leq ch_f$ eşitsizliğinin geçerli olduğu gösterilmiştir. Yine Vapnik tarafından uygulamalarda $h \approx h_f$ olacak şekilde kullanılabilceği belirtilmiştir. Kısaca sınıflandırma ve regresyon problemlerinde yaklaşan fonksiyonlar ile kayıp fonksiyonlarının VC boyutları birbirlerinin yerine kullanılabilir (Cherkassky & Mulier, 2007: 111).

VC boyutunun modelin parametre sayısı ile bir ilişkisi yoktur. Prensip, çok sayıda parametreye sahip olup küçük VC boyutu olan modeller bulunabilir. Bunun tersi de doğrudur. Örneğin, $f(x, \omega) = \text{sgn}(\sin(\omega x))$, $\omega \in \mathcal{R}$ tek parametrelili gösterge fonksiyonu ele alınsın. Amaç, seçilen herhangi bir n değeri için n adet gözlemin ayrılabilmesi olsun. $x_i = 10^{-i}$, $i = 1, 2, \dots, n$ seçildiğinde, $f(x, \omega)$

fonksiyonlar kümesi tarafından bu noktaların ikili olarak sınıflandırılabilmesi için ω parametresinin $\omega = \pi \left(1 + \sum_{i=1}^n \frac{(1-y_i)10^{-i}}{2} \right)$ kuralına göre seçilmesi yeterlidir. Bu özel durumda sadece tek bir parametreye sahip olan bir fonksiyonlar kümesinin VC boyutu sonsuzdur (Haykin, 1999: 119).

VC boyutundan başka VC entropi, annealed VC entropi gibi kapasite ölçütleri olsa da bunlar dağılıma bağlı olduklarından uygulanmaları daha zordur. VC boyutu uygulanabilirlik açısından içlerinden en uygun olan kapasite ölçütüdür (Schölkopf & Smola, 2001: 139).

3.1.4 Genelleme Sınırları

Tüm olası fonksiyon sınıfları içinden ampirik riski sıfıra düşürecek bir fonksiyon bulmak daima mümkün olduğundan, sadece ampirik riski minimize etmeyi yeterli görmek aşırı iyimser bir yaklaşımdır. Bu durum teorik açıdan bir açmaza neden olmaktadır. Olasılık dağılımı bilinmediğinden veri uzayındaki optimal modeli bulmak için beklenen risk kullanılamaz. Ampirik risk ise fazla iyimser olduğu için optimal modeli bulmada herhangi bir faydası yoktur. Bu açmazdan kurtulmak amacıyla, Vapnik bir modelin genelleme hatasını belirlemek için VC boyutuna dayanan “VC güveni (VC confidence)” kavramını geliştirmiştir.

ARM ilkesini uygulayan bir öğrenme makinesinin genelleme yeteneği için (3.8) denkleminde gösterilen “genelleme sınırı” ampirik riski minimize eden $Q(z, \omega_n^*)$ kayıp fonksiyonu da dahil olmak üzere tüm kayıp fonksiyonları (sınırlı ve negatif olmayan) için en az $1 - \delta$ olasılıkla geçerlidir.

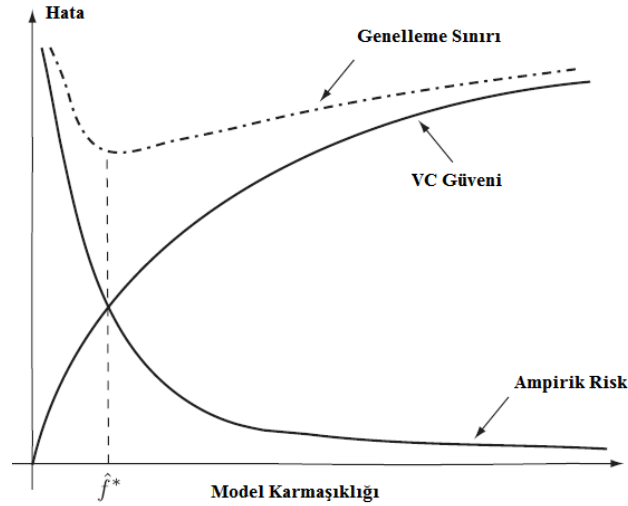
$$R(\omega) \leq R_{amp}(\omega) + \psi(n, h, \delta) \quad (3.8)$$

(Denklem 3.8) de ψ terimi “VC güveni” olarak isimlendirilir. VC güveni, ampirik riskle beraber herhangi bir modelin beklenen riski için bir üst sınır belirler. Bu terim daha açık bir şekilde (Denklem 3.9) daki gibi gösterilebilir (Clarke vd., 2009: 270).

$$\psi(n, h, \delta) = \sqrt{\frac{h \left(\log \left(\frac{2n}{h} \right) + 1 \right) - \log \left(\frac{\delta}{4} \right)}{n}} \quad (3.9)$$

(3.8) eşitsizliğinin sol tarafını hesaplamak mümkün değildir. Eğer h değeri biliniyorsa eşitsizliğin sağ tarafı hesaplanabilir. Bu nedenle farklı öğrenme makineleri yani $f(x, \omega)$ fonksiyonlar kümesi verildiğinde ve sabit, yeterince küçük δ değeri için, eşitsizliğin sağ tarafını minimize eden makineyi seçmekle gerçek risk üzerinde en küçük üst sınırı veren makine seçilmiş olur. Bu, belirli bir görev için öğrenme makinesi seçmenin ilkesel bir yolunu gösterir (Burges, 1998: 124).

(3.9) denklemi incelendiğinde, VC güveninin doğrudan VC boyutu (h) ile orantılı olduğu görülür. Yüksek bir VC boyutu, modelin karmaşıklığının artmasına, bu da genelleme hatasının yüksek olmasına sebep olur. Gereğinden fazla karmaşık modeller eğitim verilerini aşırı öğrenmeye eğilim gösterirler ve bu nedenle iyi genelleme yapamazlar. Ampirik risk ile VC güveninin toplamı bu iki eğrinin oluşturduğu bir zarfı temsil etmektedirler. Bu zarf genelleme sınırını gösterir (Şekil 23). Bu sınırın minimum olduğu noktada optimal modele ulaşılır (Hamel, 2009: 178).



Şekil 23. Genelleme Sınırı

VC güveni eğitim kümesinin büyüklüğü (n) ile ters orantılıdır. Bundan dolayı eğitim kümesi büyüdükçe veri uzayı hakkında daha çok şey bilineceği için daha düşük genelleme hatası oluşur.

Elde edilen sınırların uygulanabilir olması için $(1 - \delta)$ ile belirtilen güven seviyesinin gözlemlerin bir fonksiyonu olarak seçilmesi daha uygundur. Böylece; gözlem sayısı fazla olduğunda güven seviyesi yüksek, az olduğunda ise düşük olarak belirlenir. Vapnik, güven seviyesi için (Denklem 3.10) da gösterilen kuralı önermiştir (Cherkassky & Mulier, 2007: 116).

$$\delta = \min\left(\frac{4}{\sqrt{n}}, 1\right) \quad (3.10)$$

VC güveni, gözlem sayısının VC boyutuna oranına (n/h) sıkı bir şekilde bağlı olduğundan, bu oran göz önüne alınarak

- $(n/h \leq 20)$ ise eğitim kümesinin küçük
- $(n/h > 20)$ ise eğitim kümesinin büyük

olduğu kabul edilir (Kecman, 2001: 138).

VC boyutu gözlemleri oluşturan olasılık dağılımına bağlı olmadığı için elde edilen genelleme sınırları tüm olasılık dağılımlarına uygulanabilir. Öte yandan, eldeki dağılımın belirli özelliklerini dikkate almadığından daha gevşek sınırlar oluşturması bir dezavantajdır (Von Luxburg & Schölkopf, 2008: 1).

VC boyutunun (h) sonlu bir değere sahip olması, ARM ilkesinin tutarlılığı ve risk fonksiyonelinin hızlı yakınsamasına ek olarak tümevarımsal öğrenme sürecinin iyi bir genelleme yeteneğine sahip olması için gerek ve yeter şartları sağlamaktadır.

3.1.5 Yapısal Risk Minimizasyonu (YRM) İlkesi

YRM tümevarım ilkesi, sınırlı gözlemler için optimal model karmaşıklığını seçmek amacıyla geliştirilmiş bir mekanizmadır. YRM ilkesi ilk olarak sınıflandırma problemleri için geliştirilmiş ve uygulanmış olsa da risk fonksiyonelinin minimize edilmesi gereken herhangi bir öğrenme probleminde uygulanabilir.

Fonksiyon kümeleri sonsuz büyüklükte olduğu için ek bir bilgi olmadan optimal fonksiyonu bulmak amacıyla bu fonksiyon kümelerinin elamanlarının

tamamını teker teker incelemek pratik bir yaklaşım değildir. Bu nedenle Vapnik, kapasite ile eğitim verisini eşleştirip en iyi genelleme performansını yakalamak için, fonksiyon sınıflarının VC boyutlarını optimal modelin bulunmasında bir rehber olarak kullanmayı önermiş ve bu yönetime “*Yapısal Risk Minimizasyonu (Structural Risk Minimization)*” adını vermiştir (Evgeniou vd., 2002: 425).

ARM ilkesine dayanan parametrik yöntemler parametre sayılarının belli olduğu, kapasitesi sabit bir yaklaşan fonksiyonlar kümesi kullanırlar ve dayanak (hedef) fonksiyonun bu kümede olduğunu varsayarlar. Bu yöntemler ancak bu varsayım doğru ise işe yaramaktadır. Ayrıca, ARM ilkesi sadece büyük eğitim kümelerinde iyi sonuçlar verir. Böyle bir durumda VC güveni küçük olacağı için ampirik risk ile gerçek risk birbirine yakındır (Nasien vd., 2010: 762). Yani küçük ampirik risk, küçük gerçek riski garanti eder.

Küçük eğitim kümeleri için genelleme sınırının sağ tarafındaki her iki terim de minimize edilmelidir. Sağ taraftaki ilk terim olan ampirik risk, fonksiyonlar kümesinden seçilen belli bir fonksiyona bağlıken; ikinci terim fonksiyonlar kümesinin VC boyutuna bağlıdır. Gerçek riski minimize etmek için VC boyutu bir kontrol değişkeni olarak kullanılmalıdır.

Genelleme sınırını minimize etmek için kullanılabilir iki YRM stratejisi vardır:

1. Model karmaşıklığını (VC boyutunu) sabit tutup ampirik riski minimize etmek.
2. Ampirik riski sabit (küçük) tutup VC boyutunu minimize etmek.

Birçok istatistiksel yöntem ve yapay sinir ağları ilk stratejiyi uygularken, destek vektör makineleri ise ikinci stratejiyi uygular (Wang & Zong, 2003: 56).

YRM ilkesine göre sınırlı veri ile bir öğrenme problemini çözmek için yaklaşan fonksiyonlar kümesi üzerinde önsel bir yapı tanımının yapılması gerekir.

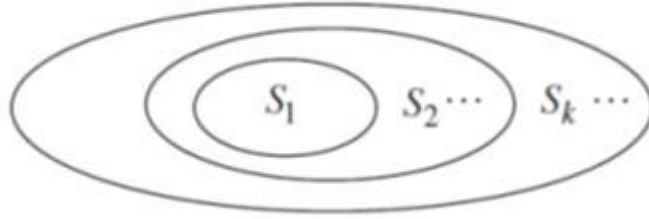
Bunun için, kayıp fonksiyonları kümelerinden birbirinin altkümüsi olacak şekilde bir yapı (Denklem 3.11) oluşturulur.

$$S_1 \subset S_2 \subset \dots \subset S_k \subset \dots \quad (3.11)$$

Her bir altkümenin (S_k), VC boyutu sonludur ve h_k ile gösterilir (Şekil 24).

Tanım gereği, oluşturulan yapı altkümelerin kapasitelerine (VC boyutuna) göre sıralı olmasını sağlar (Li vd., 2009: 437).

$$h_1 \leq h_2 \leq \dots \leq h_k \leq \dots$$



Şekil 24. Kapasitelerine Göre Sıralanmış Fonksiyon Kümelerinden Oluşan Yapı

Bu yapı belirlendikten sonra optimal model seçimi:

- Yapının optimal kapasiteye sahip elemanını seçmek
- Bu elemandan modeli tahmin etmek

adımlarını içerir. YRM ilkesinin uygulanabilmesi için

- Yapıdaki her bir S_k alt kümesinin VC boyutu hesaplanır.
- Her bir alt küme için ampirik risk minimize edilir (Cherkassky & Mulier, 2007: 123).

YRM ilkesinde belli bir yaklaşan fonksiyon tipi (polinomsal, ileri yönelimli yapay sinir ağı, dairesel tabanlı fonksiyonlar vb.) belirtilmez. Böyle bir seçim İstatistiksel Öğrenme Teorisi'nin dışındadır ve önsel bir bilgi ile veya modeli oluşturan araştırmacının deneyimleriyle belirlenir (Evgeniou, 2000: 11).

YRM ilkesi destek vektör makineleri yönteminin temelinde yer alan “*maksimum marj sınıflandırıcı (maximum margin classifier)*” fikrini de şekillendirmiştir. Bu tür sınıflandırıcılarda, iyi genelleme için en büyük marja sahip modellerin bulunması amaçlanmaktadır (Cox & Adhami, 2002: 273). Bu durumda marj kavramı ile kapasite kavramı (VC boyut) arasındaki ilişki önem kazanır.

3.2 Maksimum Marj Sınıflandırıcıları

Maksimum marja dayanan sınıflandırıcı modellerinde temel amaç, iki sınıfın birbirine en yakın olduğu konumdaki sınıf sınırlarına eşit uzaklıkta olan bir karar yüzeyi bulmaktır. Ayrıca, sınıf sınırları arasındaki mesafe maksimize edilmeye çalışılır. Bu yaklaşımı kullanılan modellere “*maksimum marj sınıflandırıcıları*” denilir (Steinwart & Christmann, 2008: 14).

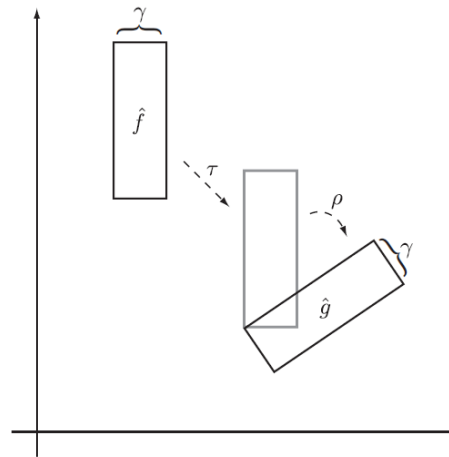
Eğitim kümesi veri uzayının sadece bir yaklaşmasını temsil etmektedir. Karar yüzeyi sınıf sınırlarına eşit uzaklıktaki bir konuma yerleştirildiğinde, eğitim kümesinde yer almayan gözlemlerin doğru sınıflandırma olasılığı artar. Karar yüzeyinden sınıf sınırlarına olan mesafeyi maksimize etmek ise bu olasılığı daha da arttırır (Ramon & Christodoulou, 2006: 5-7).

Eğer bir hiperdüzlem, doğrusal karar yüzeyine paralel ve ilgili sınıfın tüm gözlemleri hiperdüzlemin altında veya üstündeyseniz “*destek hiperdüzlemi (support hyperplane)*” olarak isimlendirilir (Bennet & Campbell, 2000: 2). Destek hiperdüzlemi, karar yüzeyinin bir kopyasının ilgili sınıfın sınırına dokunduğu noktaya taşınmış hali olarak düşünülebilir. İkili sınıflandırma problemlerinde tipik olarak iki destek hiperdüzlemi vardır. Bunlardan birisi pozitif olarak etiketlenen sınıf yönünde, diğeri negatif olarak etiketlenen sınıf yönünde taşınmıştır.

“*Optimal ayırıcı hiperdüzlemi (optimal seperating hyperplane)*” oluşturabilmek için kullanılan en önemli kavramlardan birisi marj kavramıdır. İkili sınıflandırma probleminde iki destek hiperdüzlemi arasındaki mesafeye “*marj (margin)*” denilir (Clarke vd., 2009: 263). Marj kavramı destek vektör makineleri formülasyonunu anlamamanın ilk önemli adımıdır.

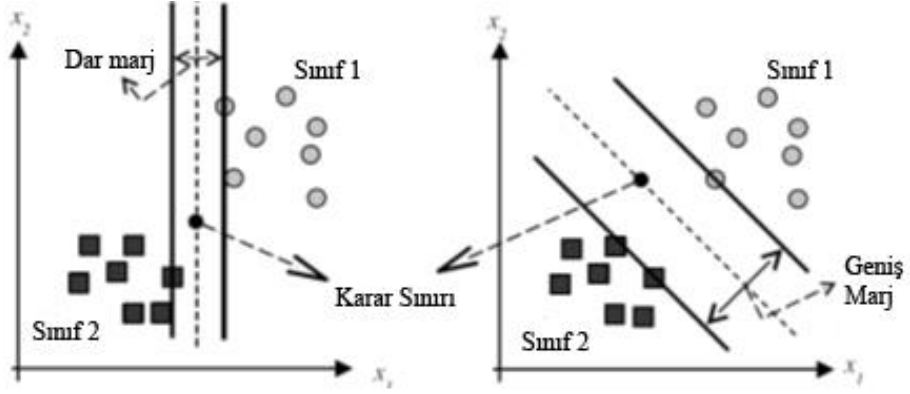
Destek hiperdüzlemi ve marj kavramları göz önünde bulundurularak “*optimal ayırıcı hiperdüzlem*” kriterleri söyle ifade edilebilir: İkili sınıflandırma problemi için bulunan hiperdüzlem, eğer iki destek hiperdüzlemine eşit mesafede ve marj değerini maksimize ediyorsa optimal ayırıcı hiperdüzlemdir (Abe, 2010: 16). Optimal ayırıcı hiperdüzlem en büyük marja sahip ayırıcı hiperdüzlem olduğu için tektir.

Destek hiperdüzlemleri üzerinde yapılacak herhangi bir döndürme (rotation) veya öteleme (translation) işlemi marjın değişmesine neden olur. Marjı γ büyüklüğünde olan ayırıcı hiperdüzlemler sınıfı $F[\gamma]$ ile gösterilsin. $F[\gamma]$ sınıfının döndürme ve öteleme işlemlerine göre kapalı olduğu yani olası tüm γ büyüklüğündeki marjları içerdiği kabul edilsin. Buna göre bütün $f \in F[\gamma]$ hiperdüzlemleri üzerinde tüm döndürme (ρ) ve öteleme (τ) işlemleri için $\rho(f) \in F[\gamma]$, $\tau(f) \in F[\gamma]$ özellikleri sağlanmaktadır. (Şekil 25) de görüldüğü gibi herhangi iki $f, g \in F[\gamma]$ hiperdüzlemi için $\rho(\tau(f)) = g$ olacak şekilde bir ρ döndürmesi ve τ ötelemesi bulmak mümkündür (Hamel, 2009: 172).



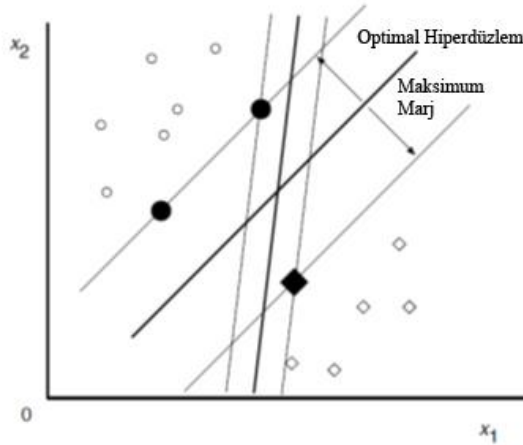
Şekil 25. Döndürme ve Öteleme İşlemleri

(Şekil 26) verilen bir eğitim kümesi için bulunabilecek birçok ayırıcı hiperdüzlemlerden ikisini göstermektedir. Sağda, istenen geniş marjlı hiperdüzlem, solda ise daha az geçerli olan dar marjlı hiperdüzlem görünmektedir (Huang vd., 2006: 22).



Şekil 26. Geniş Marjlı Ayırıcı Hiperdüzlem

(Şekil 27) de marjın, her bir sınıftaki içi dolu olarak gösterilen noktalar tarafından belirlendiği görülmektedir. Destek hiperdüzleminin tam üzerinde olan bu noktalara "*destek vektörleri*" denilir (Abe, 2010: 18). Eğer bir hiperdüzlem, ilgili sınıfın bir destek vektörünü geçecek olursa artık bir destek hiperdüzlemi olarak görülmez. Çünkü böyle bir durumda sınıf gözlemleri destek hiperdüzleminin her iki yanında da yer almaktadır.

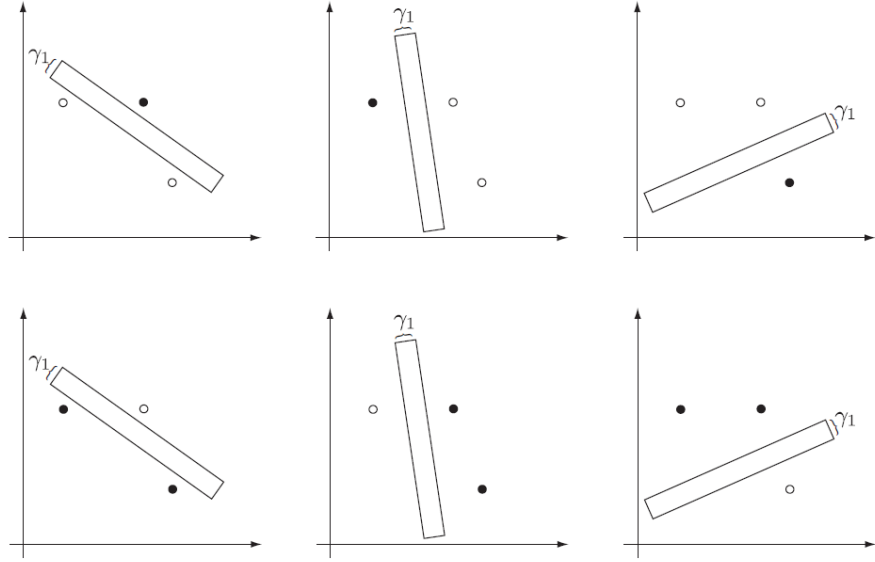


Şekil 27. Destek Vektörleri

3.2.1 Marj - Karmaşıklık İlişkisi

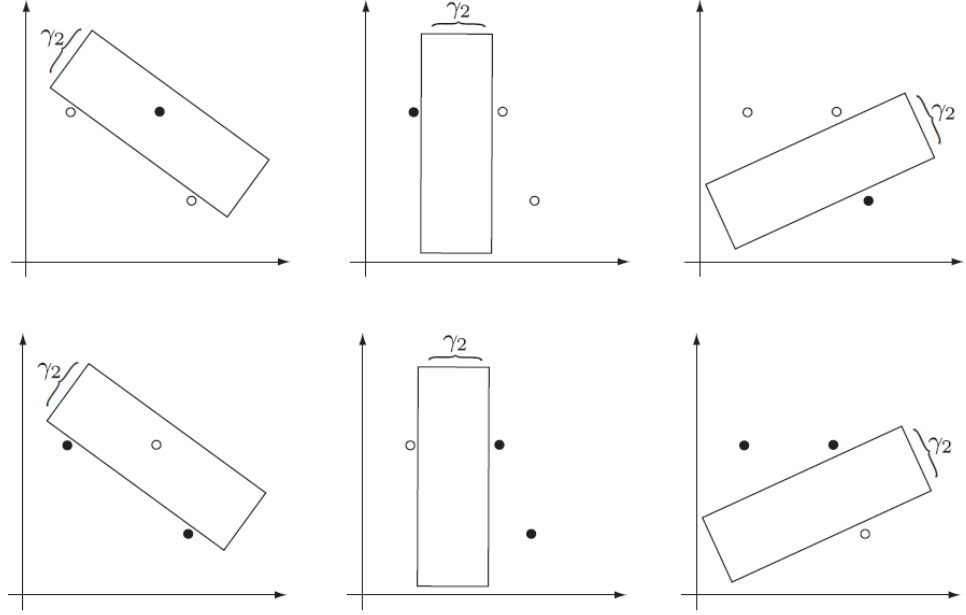
Maksimum marj sınıflandırıcılarının gücü, İstatistiksel Öğrenme Teorisi tarafından belirlenen genelleme sınırlarının açık şekilde optimize edilmesinden gelmektedir (Cristianini & Schölkopf, 2002: 35). İstatistiksel Öğrenme Teorisi'nin sonuçları kullanılarak maksimum marja sahip hiperdüzlemin optimal ayırıcı hiperdüzlem olduğu gösterilebilir.

VC boyutu kavramı, marj esas alınarak tanımlanabilir. Bir veri kümesi üzerinde tanımlanmış olan $F[\gamma]$ hiperdüzlemler kümesinin VC boyutu, veri kümesinin $F[\gamma]$ tarafından ayrılabilen (shatter) en büyük sonlu alt kümesinin eleman sayısıdır. (Şekil 28) VC boyutu (h_1) 3 olan γ_1 marjlı ayırıcı hiperdüzlemleri göstermektedir. Veri kümesi içindeki etiketlemeler içi boş ve dolu toplar şeklinde temsil edilmiştir. Tüm gözlemlerin aynı etikete sahip olduğu durumlar ise ayırarak herhangi bir şey olmadığından gösterilmemiştir.



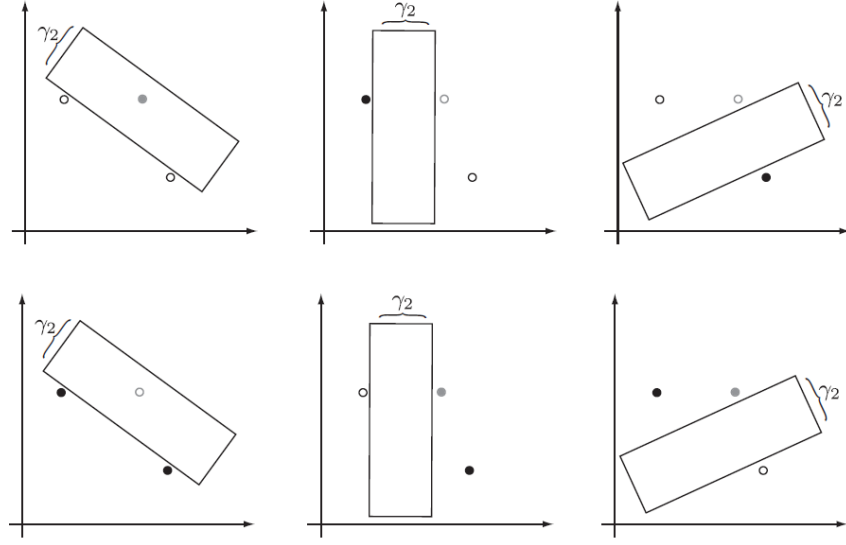
Şekil 28. Dar Marjlı Ayırıcı Hiperdüzlem

Aynı veri kümesi üzerinde bu defa $\gamma_2 > \gamma_1$ olan ikinci bir $F[\gamma_2]$ hiperdüzlemler sınıfı tanımlandığında (Şekil 29) da görüldüğü gibi tüm gözlemler doğru şekilde ayırlanamamaktadırlar.



Şekil 29. Geniş Marjlı Ayırıcı Hiperdüzlem

Eğitim kümesindeki gözlemlerden birisi (gri top) kaldırılırsa hiperdüzlem sınıfı eğitim kümesinin bu alt kümesini ayırabilir (Şekil 30). $F[\gamma_2]$ hiperdüzlemler sınıfı, eğitim kümesinin iki elemanlı altkümesini ayırabildiği için VC boyutu (h_2) 2 dir (Hamel, 2009: 174).



Şekil 30. Silinen Gözlemler

Büyük VC boyutları, yüksek karmaşıklığa sahip model sınıflarını temsil etmektedir. Bunun tersi de doğrudur. Yukarıda incelenen durumda $h_1 > h_2$ olduğundan, $F[\gamma_1]$ modelleri $F[\gamma_2]$ modellerinden daha karmaşıktır. Bu durum $F[\gamma_1] \supset F[\gamma_2]$ olarak gösterilmektedir. $\gamma_1 < \gamma_2$ olması, dar marjlı modellerin geniş marjlı modellerden daha karmaşık olduğunu gösterir. Karmaşık modeller, daha az karmaşık modellere göre veriye daha iyi uyarlar. $F[\gamma_1] \supset F[\gamma_2]$ olması kapasitesi daha düşük bir model ile modellenebilen bir problemin daha yüksek kapasiteli bir modelle de çözülebileceğini ifade etmektedir.

Vapnik, analitik olarak γ -marjlı hiperdüzlemlerin VC boyutu için

$$h \leq \min\left(\frac{r^2}{\gamma^2}, d\right) + 1$$

sınırının geçerli olduğunu ispatlamıştır (Belousov vd., 2002: 16). r , tüm eğitim gözlemlerini içeren en küçük hiperkürenin yarıçapıdır. Buna göre göre hiperdüzlemin karmaşıklığını (VC boyutunu) boyuttan (d) bağımsız olarak doğrudan kontrol etmek mümkündür. Bu sınır incelendiğinde minimum

karmaşıklığa, dolayısıyla en yüksek genelleme yeteneğine sahip olan ayırıcı hiperdüzlemin en geniş marja sahip olan hiperdüzlem olduğu görülmektedir.

3.2.2 Maksimum Marj Sınıflandırıcıları Optimizasyon Problemi

Optimal ayırıcı hiperdüzlemi bulma problemi, uygun (feasible) çözümlerin farklı destek hiperdüzlemler olduğu bir optimizasyon problemidir. Amaç fonksiyonu her bir uygun çözümün marjını hesaplar. Maksimum marja ulaşılacak istendiğinden amaç fonksiyonu da maksimize edilmeye çalışılır. Kısıtlar ise kendi sınıf sınırlarını geçmelerine izin verilmeyen destek hiperdüzlemlerinin konumlarıdır.

Bu durumda optimizasyon probleminin amaç fonksiyonu

$$m^* = \text{maks } \phi(w, b)$$

şeklinde yazılabilir. $\phi(w, b)$ fonksiyonu, $w \cdot x = b$ ayırıcı hiperdüzleminin marjını hesaplar. Maksimum marj (m^*), $w^* \cdot x = b^*$ optimal ayırıcı hiperdüzlemin marjını göstermektedir.

Analitik geometri kullanarak amaç fonksiyonu optimizasyon için daha uygun bir biçimde yazılabilir. Bunun için “izdüşüm (projection)” kavramı kullanılır. a ve b , \mathcal{R}^n uzayında iki vektör ve bunların arasındaki açı θ olmak üzere a vektörünün b vektörü yönündeki izdüşümü p_a ile gösterilir ve

$$p_a = \|a\| \cos \theta = \frac{a \cdot b}{\|b\|}$$

şeklinde hesaplanır (Hamel, 2009: 78).

$w^* \cdot x = b^*$ optimal ayırıcı hiperdüzlemi gösterdiğinden bu hiperdüzlemden eşit uzaklıkta olan destek hiperdüzlemleri

$$w^* \cdot x = b^* + k$$

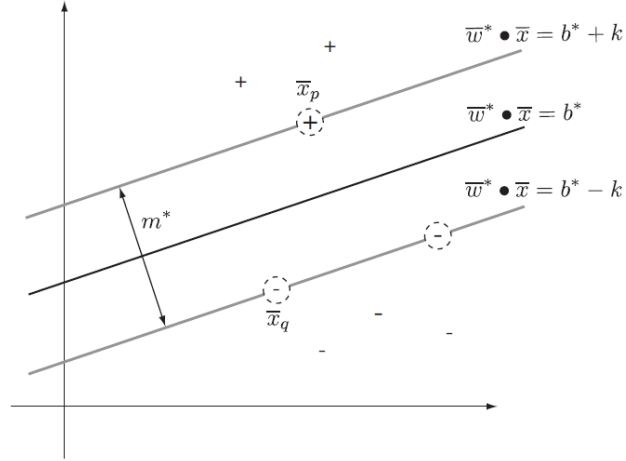
$$w^* \cdot x = b^* - k$$

şeklinde yazılabilir. x_p , +1 sınıfının; x_q ise -1 sınıfının bir destek vektörü olduğu varsayıldığında

$$w^* \cdot x_p = b^* + k$$

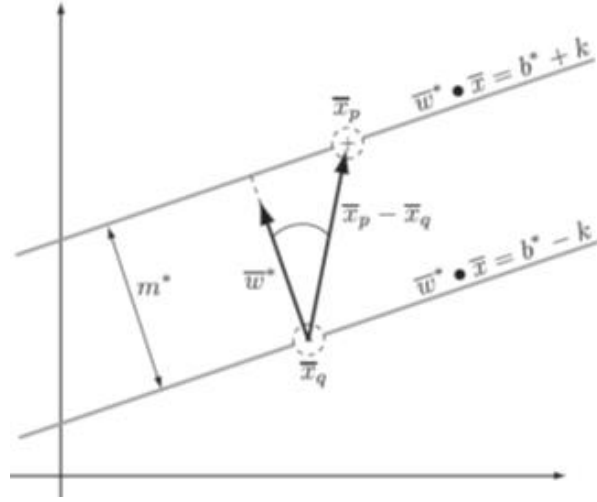
$$w^* \cdot x_q = b^* - k$$

eşitlikleri sağlanmalıdır (Şekil 31).



Şekil 31. Destek Hiperdüzlemleri

Marj geometrik olarak $x_p - x_q$ farkının w^* yönündeki izdüşümü kullanarak hesaplanır. Bu durumda marj, iki destek vektörü farkının optimal ayırıcı hiperdüzlemin normal vektörü yönündeki izdüşümü olarak da tarif edilebilir (Şekil 32).



Şekil 32. Marjın İzdüşüm Yoluyla Gösterilmesi

Aradaki açı θ olmak üzere yukarıdaki ifadeler cebirsel olarak yazılırsa

$$\begin{aligned}
 m^* &= \|x_p - x_q\| \cos \theta = \frac{w^*(x_p - x_q)}{\|w^*\|} = \frac{w^* \cdot x_p - w^* \cdot x_q}{\|w^*\|} \\
 &= \frac{(b^* + k) - (b^* - k)}{\|w^*\|} \\
 &= \frac{2k}{\|w^*\|}
 \end{aligned}$$

bulunur. Bu sonucu göre amaç fonksiyonu

$$m^* = \text{maks} \frac{2k}{\|w\|}$$

olur. Herhangi bir maksimizasyon problemi aynı zamanda minimizasyon problemi olarak da ifade edilebileceği için aynı amaç fonksiyonu $m^* = \min \frac{\|w\|}{2k}$ olarak da yazılabilir. Kare alma işlemi monoton artan olduğundan bir değeri minimize etmek ile bu değerin karesini minimize etmek aynı anlama gelmektedir. Ayrıca optimizasyon sonucu, sabit bir değerle çarpma işleminden de bağımsızdır (Boyd &

Vandenberghe, 2004: 130-131). Bu bilgilerden yola çıkarak amaç fonksiyonu yeniden düzenlenirse

$$\min \frac{\|w\|}{2k} = \min \frac{\|w\|^2}{2k} = \frac{1}{2k} w \cdot w = \min \frac{1}{2} w \cdot w$$

sonucu elde edilir. Bulunan bu sonuca göre amaç fonksiyonu

$$\phi(w, b) = \frac{1}{2} w \cdot w$$

şeklinde yazılır. Görüldüğü gibi bu fonksiyonda b terimi yer almamaktadır. b teriminin kısıtların oluşturulmasında etkisi vardır.

Optimizasyon sırasında destek hiperdüzlemlerinin ait oldukları sınıf sınırını aşmalarına izin verilmez. Yani optimizasyon süresince destek hiperdüzlemleri olarak kalmaları gerekmektedir. Bu durumda

$$w^* \cdot x_i \geq b^* + k \quad \forall y_i \in +1 \text{ Sınıfı}$$

$$w^* \cdot x_i \leq b^* - k \quad \forall y_i \in -1 \text{ Sınıfı}$$

eşitsizliklerinin sağlanması gerekir. Bu kısıtlar olası tüm destek hiperdüzlemleri için geçerlidir. Genelleme yapmak için $k = 1$ değeri alındığında

$$w^* \cdot x_i \geq 1 + b^* \quad \forall y_i \in +1 \text{ Sınıfı}$$

$$w^* \cdot (-x_i) \geq 1 - b^* \quad \forall y_i \in -1 \text{ Sınıfı}$$

şeklindeki “kanonik biçim (canonical form)” elde edilir (Mavroforakis & Theodoridis, 2006: 672). Bu eşitsizlik kısıtları

$$w \cdot (y_i x_i) \geq 1 + y_i b$$

olarak tek bir kısıt olarak gösterilebilir. Bu gösterimde tüm gözlemlerin kısıt oluşturmada yer aldığı görülür. Kısıtlar sadece uygun çözümleri tanımlarlar.

Özetle; doğrusal ayrılabilen ve iki sınıflı bir eğitim kümesi verildiğinde $w^* \cdot x = b^*$ optimal ayırıcı hiperdüzlemi

Amaç fonksiyonu

$$\min \phi(w, b) = \min \frac{1}{2} w \cdot w$$

Kısıtı

$$w \cdot (y_i x_i) \geq 1 + y_i b$$

olan optimizasyon problemi çözüldüğünde bulunabilir (Hamel, 2009: 82).

Maksimum marj amaç fonksiyonu w parametrelerindeki küçük değişimlere karşı aşırı hassas değildir. Küçük parametre değişimleri durumunda yanlış sınıflandırma olasılığı maksimum marj için minimize edilmiştir (Boser vd., 1992: 4).

Amaç fonksiyonunda $w \cdot w$ değerinin minimize edilmesi yapay sinir ağlarındaki ağırlık sönümüne yakından benzemektedir (Suykens vd., 2002: 30).

Amaç fonksiyonunun konveks olması global minimumunun bulunabileceğini göstermektedir. Başka bir ifade ile uygun çözümler kümesi verildiğinde amaç fonksiyonunun en küçük değerini almasını sağlayacak, örneğin kuadratik programlama gibi, bir yöntem kullanılabilir. Bu maksimum marj sınıflandırıcılarının yapay sinir ağları gibi çözümü başlangıç durumlarına veya kontrol edilmesi zor olan parametrelere bağlı olan algoritmalara karşı önemli bir üstünlüğüdür.

3.2.3 Lagrange Çarpanları Yöntemi

Optimizasyon probleminin dualini almak probleme yeni bakış açıları sağlar. Bu yeni bakış açıları optimizasyon probleminin çözümü için yeni teknikler sağlayabildiği gibi, destek vektör makineleri gibi yeni algoritmaların oluşturulmasına da yol açabilmektedir. Bir optimizasyon probleminin dualini oluşturmanın uygun bir yolu Lagrange çarpanları yöntemidir.

Başlangıçtaki optimizasyon probleminde tüm $x \in \mathcal{R}^n$ ve $i = 1, \dots, n$ için amaç fonksiyonu

$$\min_x \phi(x)$$

Kısıtlar

$$g_i(x) \geq 0$$

şeklindedir. Bu tür bir gösterime “*primal problem*” denilir. ϕ fonksiyonu konveks ve g_i kısıtları da doğrusal ise primal problem “*Lagrange çarpanları*” yöntemiyle çözülebilir (Boyd & Vandenberghe, 2004: 223).

Lagrange optimizasyon problemi

$$\max_{\alpha} \min_x L(\alpha, x) = \max_{\alpha} \min_x \left(\phi(x) - \sum_{i=1}^n \alpha_i g_i(x) \right)$$

$$\alpha_i \geq 0$$

şeklinde oluşturulur. Yeni $L(\alpha, x)$ amaç fonksiyonu, orijinal ϕ amaç fonksiyonu ile g_i kısıtlarının doğrusal birleşiminden oluşmaktadır ve “*Lagrange fonksiyonu*” olarak bilinir. $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ parametreleri Lagrange çarpanlarını göstermektedir (Campbell & Ying, 2011: 70).

Her bir g_i kısıtı için bir α_i Lagrange çarpanı bulunmaktadır. x *primal değişken*, α *dual değişken* olarak isimlendirilir. Bu yeni optimizasyon probleminde birbirinin tersi amaçlara sahip iki optimizasyon operatörü iç içe bulunmaktadır. Her bir operatör diğer operatör tarafından optimize edilmesi için kısmi değerlendirilmiş bir fonksiyon geriye çevirir. Örneğin optimal x değeri olarak x^* bulunmuş ise problem

$$\max_{\alpha} L(\alpha, x^*) = \max_{\alpha} \left(\phi(x^*) - \sum_{i=1}^n \alpha_i g_i(x^*) \right)$$

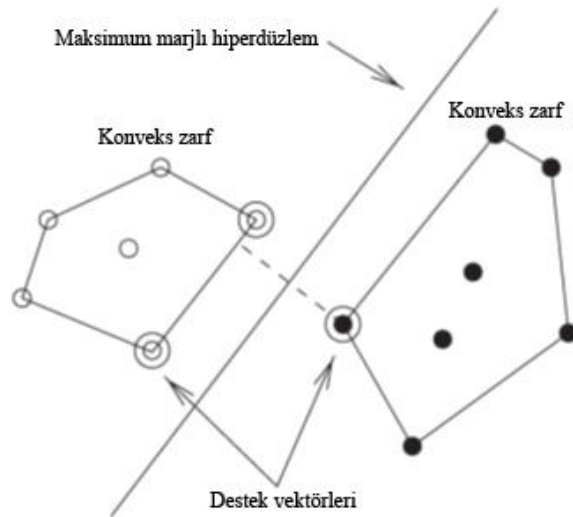
şeklinde bir maksimizasyon problemi halini alır. Tersini durumda optimal α değeri olarak α^* bulunmuş ise bu sefer de

$$\min_x L(\alpha^*, x) = \min_x \left(\phi(x) - \sum_{i=1}^n \alpha_i^* g_i(x) \right)$$

şeklinde yazılabilen bir minimizasyon problemi haline gelir (Hamel, 2009: 93).

Çözüm $L(\alpha, x)$ değerini α dual değişkenine göre maksimize, x primal değişkenine göre minimize eden noktadır. Bu da çözümün $L(\alpha, x)$ fonksiyonunun bir “*semer (eyer) noktası*” olacağı anlamına gelir (Gunn, 1998: 8). Amaç fonksiyonu konveks ve kısıtların da doğrusal olduğu varsayıldığı için tek bir semer noktası vardır. Dolayısıyla semer noktada Lagrange fonksiyonunun primal ve dual değişkenlere göre kısmi türevi sıfır olacaktır.

Dual gösterimin geometrik yorumu şöyle yapılabilir: Her bir sınıfın, tüm gözlemleri içeren en küçük konveks küme anlamına gelen, “*konveks zarfı (convex hull)*” belirlenir (Şekil 33). Optimal ayırıcı hiperdüzlem, sınıfların konveks zarfları arasındaki en kısa mesafeyi birleştiren bağlantının ortasından geçen hiperdüzlemdir (Bennet & Campbell, 2000: 1).



Şekil 33. Konveks Zarf

Dual problemi çözmek, iki destek hiperdüzlemi arasındaki maksimum marjı bulmak ile eşdeğerdir. Eğitim gözlemlerinin sadece küçük bir bölümü (destek vektörleri) problemin hem primalinin hem dualinin çözümünü belirler. Primal problemde, marjın tam üstünde olan dolayısıyla marjı belirleyen gözlemler destek vektörleridir. Dual problemde ise, aynı destek vektörleri, her sınıfın konveks zarfı içinde birbirine en yakın noktaları belirlerler (Cherkassky & Mulier, 2007: 422).

3.2.4 Karush-Kuhn-Tucker (KKT) Koşulları

Belirli koşullar altında Lagrange dualinin çözümü aynı zamanda primalin de çözümü olmaktadır. α^* ve x^* değerlerinin Lagrange dualinin çözümü olduğu kabul edildiği durum

$$\max_{\alpha} \min_x L(\alpha, x) = L(\alpha^*, x^*) = \phi(x^*) - \sum_{i=1}^n \alpha_i^* g_i(x^*)$$

olarak ifade edilebilir. x^* değerinin primalin de çözümü olabilmesi için

$$\frac{\partial L}{\partial x}(\alpha^*, x^*) = 0 \quad (x^* \text{ değerinin semer noktası üzerinde olmasını garantiler})$$

$$g_i(x^*) \geq 0 \quad (\text{Primal kısıt})$$

$$\alpha_i^* \geq 0 \quad (\text{Dual kısıt})$$

$$\alpha_i^* g_i(x^*) = 0 \quad (\text{Tamamlayıcı gevşeklik koşulu})$$

koşullarının sağlanması gerekir (Campbell & Ying, 2011: 71-42). Bu koşullar kümesi “*Karush–Kuhn–Tucker (KKT) koşulları*” olarak bilinir. “*Tamamlayıcı gevşeklik (complementary slackness)*” koşulu nedeniyle Lagrange fonksiyonundaki $\sum_{i=1}^n \alpha_i^* g_i(x^*)$ teriminin değeri 0 olur. Bu durumda Lagrange fonksiyonunun çözümü $L(\alpha^*, x^*) = \phi(x^*)$ olacağından dual çözüm ile primal çözüm aynı sonucu verir.

3.3 Destek Vektör Makineleri

Destek Vektör Makineleri (DVM), marj, dualite, kernel gösterimi gibi farklı temel kavramları bir araya getiren bir yöntemdir. Bu kavramlar farklı amaçlarla

kullanılsalar da aslında uzun süredir bilinmektedirler. Fakat İstatistiksel Öğrenme Teorisi'nin sağladığı sağlam temellerden yoksun olduklarından uygulanabilir öğrenme algoritmalarının oluşturulmasını sağlayamamışlardır (Cherkassky & Mulier, 2007: 408). Örneğin kernellerin kullanılması fikri 1960'lı yıllarda ortaya atılmıştır. Marj kavramı doğrusal ayrılabilir verilerin sınıflandırılabilmesi için yine 1960'lı yıllarda geliştirilmiştir. Fakat DVM yönteminde kernellerin kullanımı ve doğrusal ayrılmayan verilerin sınıflandırılabilmesi için aradan yaklaşık 30 yıl geçmesi gerekmiştir (Boser vd., 1992; Cortes & Vapnik, 1995). Bundan sonra DVM yöntemi diğer öğrenme problem tiplerine uyarlanmış ve birbirinden farklı birçok uygulamada başarılı bir şekilde uygulanmıştır.

DVM yaklaşımı ile istatistiksel yöntemler arasında varsayımlar açısından çok önemli farklar bulunmaktadır. Klasik istatistiksel çıkarıma aşağıdaki üç temel varsayıma dayanmaktadır (Huang vd., 2006: 12-13).

1. Veri, parametrelerinde doğrusal olan bir fonksiyonlar sınıfı ile modellenabilir.
2. Birçok problemde veri normal dağılmaktadır.
3. İkinci varsayımdan dolayı parametre tahmini için maksimum olabilirlik metodu kullanılır. Uygulamalarda genelde hata karelerinin toplamı olarak belirlenen amaç fonksiyonu minimize edilmeye çalışılır.

Fakat bu varsayımlar günümüzdeki birçok problem için uygun değildir. Bunun nedenleri şöyle özetlenebilir:

1. Problemler artık yüksek boyutludur. Eğer hedef fonksiyonu doğrusal değilse, girdi uzayının boyutunun (bağımsız değişken sayısının) artması model parametrelerinin üssel olarak artmasına neden olur (Boyut belası).

2. Veriyi oluşturan olasılık dağılımı normal dağılımdan farklı olabilmektedir. Etkin bir öğrenme algoritması oluşturabilmek için bu farklılığın dikkate alınması gerekir.
3. İlk iki nedenden dolayı yüksek boyutlu ve normal dağılmayan verileri modelleyebilecek yeni bir tümevarım ilkesi kullanılmalıdır.

Bu nedenlerden dolayı DVM, YRM tümevarım ilkesini kullanan, yüksek boyutlu veri kümeleri dikkate alınarak geliştirilmiş parametrik olmayan bir yöntemdir.

Cover teoremine göre doğrusal olarak ayrılamayan gözlemlerden oluşan bir girdi uzayı gerekli koşullar sağlandığında yüksek olasılıkla gözlemlerin doğrusal ayrılacağı daha yüksek boyutlu bir öznitelik uzayına dönüştürülebilir (Haykin, 1999: 352). Uygulamalarda da girdileri öğrenme problemi hakkındaki bir önsel varsayımına göre yeni bir değişkenler kümesine (özniteliklere) “eşlemek (*mapping*)” yaygındır. Öğrenme algoritması orijinal girdiler yerine bu öznitelikleri kullanır. Bu nedenle gözlemlerin doğrusal yolla kolayca ayrılacakları yüksek boyutlu uzaya (öznitelik uzayı) eşlenmeleri önemlidir. Öznitelik uzayında oluşturulan doğrusal karar fonksiyonları girdi uzayında doğrusal olmayan karar fonksiyonlarına karşılık gelir (Cristianini & Schölkopf, 2002: 34). Bunun için eşlemenin doğrusal olmaması ve öznitelik uzayının boyutunun yeterince büyük olması gerekir. DVM yönteminde girdi uzayı doğrusal olmayan bir dönüştürme ile daha yüksek boyutlu öznitelik uzayına eşlenir. DVM de öznitelik sayısına bir kısıtlama getirilmediğinden öznitelik uzayı çok büyük hatta sonsuz olabilir (Pontil & Verri, 1998: 955).

Birçok öğrenme yaklaşımı doğrusal olmayan modeller oluşturur. Doğrusal olmayan modellerin yaklaşan fonksiyonu daha iyi temsil etme potansiyeli bulunmaktadır. Fakat yapay sinir ağları gibi doğrusal olmayan modeller oluşturan yöntemlerin, global çözümü veren bir optimizasyon yöntemine ve bir karmaşıklık ölçütüne sahip olmamalarından kaynaklanan çeşitli problemleri vardır. DVM

yönteminde ise doğrusal modellerin karmaşıklığı doğru bir şekilde belirlenebildiğinden ve ampirik risk için global minimumu veren optimizasyon yöntemleri bulunduğundan yüksek boyutta gözlemleri ayırmak için doğrusal fonksiyonlar kullanılmaktadır (Moguerza & Muñoz, 2006: 324).

Destek vektör makineleri, maksimum marj sınıflandırıcılarının dual biçimi olarak görülebilir. Dual biçim, yüksek boyutlu uzaylarda problemin daha kolay çözülmesine yardımcı olur (Hamel, 2009: 91). DVM de model parametrelerinin belirlenmesi için kuadratik optimizasyon probleminin çözülmesi gerekir. Fakat yüksek boyutlu öznitelik uzayında çok sayıda parametre olması optimizasyon problemini zorlaştırır. Bunun için çözümün öznitelik uzayının boyutuna göre değil eğitim gözlemlerinin sayısına göre ölçeklendiği dual gösterimden faydalanılır (Cherkassky & Mulier, 2007: 407).

Maksimum marj sınıflandırıcıların aylak (slack) değişken kavramı kullanılarak eğitim kümesi üzerinde hata yapmalarına izin verilir. Bir sınıflandırıcının hata yapmasına izin vermek anlamsız gelebilir. Fakat gerçek uygulamalardaki verilerin birçoğunun gürültü içerdiği göz önüne alınmalıdır. Gürültü, bir sınıflandırma probleminde sınıflar arasındaki sınırın çok karmaşık olmasına sebep olabilmektedir. Hata yapmasına izin verilmeyen bir sınıflandırıcı da çok karmaşık karar yüzeyleri oluşturur (Mammone vd., 2009: 285). Karar yüzeylerinin çok karmaşık olmasına zorlayan gözlemlerin gürültü olduğunu kabul edip bu noktaları göz ardı ederek çok daha basit karar fonksiyonları oluşturmak mümkündür

Yanlış sınıflandırmaya izin verilen maksimum marj sınıflandırıcılara “*yumuşak (soft) marjlı sınıflandırıcılar*” denilir. Sınıflandırma hatasının yapılmasına izin verilmeyen sınıflandırıcılar ise “*sert (hard) marjlı sınıflandırıcılar*” olarak bilinmektedir (Steinwart & Christmann, 2008: 14-15).

Doğrusal sınıflandırıcıların kolaylıkla doğrusal olmayan sınıflandırıcılara genellenebilmesi DVM yöntemin uygulama alanlarını çok genişletmiştir. Bu genellenmenin merkezinde “*kernel fonksiyonları*” yer alır. Kernel fonksiyonu,

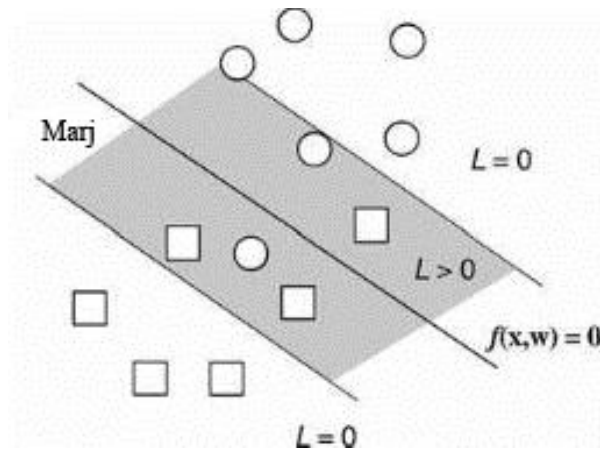
uygulamaya özgü gözlemler arasındaki benzerlik ölçütüdür ve kendi benzerlik tanımına göre öznelik uzayını oluşturur (Alpaydın, 2009: 310).

Dual gösterimin çözümüne karşılık gelen yaklaşan fonksiyon, tipik fonksiyon gösterimi yerine kernel gösterimiyle verilir. Kernel gösterimindeki çözüm, destek vektörlerinin ağırlıklı toplamları olarak yazılır (Mammone vd., 2009: 286). Bu durumda destek vektörleri eğitim kümesinin öğrenme probleminin çözümüne karşılık gelen alt kümesi olarak da tanımlanabilir

DVM yönteminde karmaşıklık kontrolü marj tabanlı kayıp fonksiyonları kullanılarak yapılır. Ampirik sonuçlar, gözlemlerin sınırlı olduğu durumlarda marj tabanlı yöntemlerin klasik yöntemlere göre daha sağlam (robust) olduğunu göstermektedir.

3.3.1 Marj Tabanlı Kayıp Fonksiyonları

Marj tabanlı kayıp fonksiyonları gözlemleri sıfır kayıplı gözlemler ve sıfırdan büyük kayıplı gözlemler olmak üzere iki parçaya ayırırlar. İkili sınıflandırma problemi ele alındığında karar sınırı ($f(x, w) = 0$), girdi uzayını pozitif ($f(x, w) > 0$) ve negatif ($f(x, w) < 0$) bölgelere ayırır. Model tarafından doğru sınıflandırılan ve karar sınırından belli bir mesafede bulunan gözlemlere sıfır kayıp; yanlış sınıflandırılan veya karar sınırına yakın olan gözlemlere pozitif kayıp atanır (Şekil 34).



Şekil 34. İkili Sınıflandırma İçin Marj Tabanlı Kayıp Fonksiyonu

İyi bir karar sınırının:

- Marj içindeki gözlemlerin toplam ampirik kaybını minimize etmek.
- Marjı maksimize etmek.

amaçları arasındaki optimal dengeyi sağlaması gerekir. Açıktır ki, bu iki amaç birbiriyle çelişir. Çünkü daha büyük marj daha büyük ampirik kayba sebep olur. Bu nedenle iyi genelleme için en uygun marj büyüklüğünün seçilmesi çok önemlidir.

Sınıflandırma problemleri için marj büyüklüğü (γ) ile parametrelendirilmiş DVM kayıp fonksiyonu (Denklem 3.12) de gösterilmiştir.

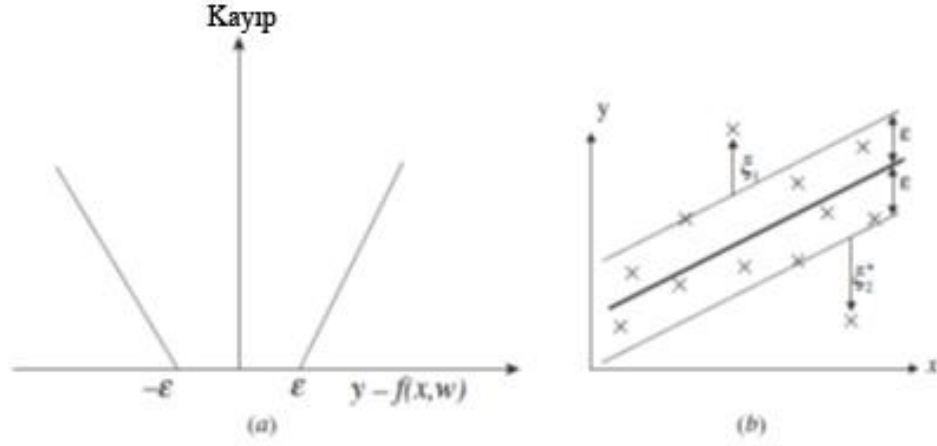
$$L_{\gamma}(y, f(x, \omega)) = \text{maks} (\gamma - \gamma f(x, \omega), 0) \quad (3.12)$$

Regresyonda tahmin edilen model reel değerli bir fonksiyon olduğundan kayıp, öngörülen $f(x, \omega)$ çıktısı ile y gerçek çıktısı arasındaki fark olarak ölçülür. Sınıflandırma problemine benzer şekilde model veriyi iyi açıkladığında değeri küçük, yanlış tahminde bulunduğu ise büyük bir değer verecek bir kayıp fonksiyonu kullanılır.

Vapnik tarafından regresyon için “ ε -duyarsız (ε -insensitive)” kayıp fonksiyonu (Denklem 3.13)

$$L_{\varepsilon}(y, f(x, \omega)) = \text{maks} (|y - f(x, \omega)| - \varepsilon, 0) \quad (3.13)$$

şeklinde tanımlanmıştır. ε parametresi küçük ve büyük sapmalar arasındaki ayrımı kontrol eder. (Şekil 35) DVM regresyonu için kayıp fonksiyonunu (a) ve aylak değişkenleri (b) göstermektedir (Cherkassky & Mulier, 2007: 412).



Şekil 35. DVM Regresyonu İçin Kayıp Fonksiyonu ve Aylak Değişkenler

3.3.2 DVM Sınıflandırıcıları

DVM, Vapnik'in öne sürdüğü “*asıl problemden önce ara bir basamak olarak daha karmaşık bir problem çözülmemelidir*” ilkesine dayanır. Örneğin sınıflandırmada asıl problem karar sınırını bulmak olduğu için, S sınıfları göstermek üzere, $P(x|S_i)$ sınıf yoğunluklarının veya $P(S_i|x)$ sonsal olasılıkların tahmininin yapılmasına gerek yoktur. Yapılması gereken sadece $P(S_i|x) = P(S_j|x)$ olduğu karar sınırının bulunmasıdır (Alpaydın, 2009: 309).

DVM' nin sınıflandırma problemine kavramsal yaklaşımı, idealize edilmiş bir başlangıç noktasından başlayıp genellemeler yaparak daha kolay anlaşılabilir. Bu başlangıç noktası, verinin “*doğrusal ayrılabilir (linearly seperable)*” olduğunun varsayıldığı özel durumdur.

3.3.2.1 Doğrusal Sert Marjlı Sınıflandırıcı

Sınıflandırma problemlerinde ayırıcı hiperdüzlem, eğitim kümesini hatasız bir şekilde ayırabilen doğrusal bir fonksiyondur. $(x_1, y_1), \dots, (x_n, y_n)$, $x \in \mathcal{R}^d$, $y \in \{-1, +1\}$ şeklinde gösterilen eğitim kümesi, uygun w ve b değerleri için

$$D(x) = (w \cdot x) + b \quad (3.14)$$

denklemiyle ifade edilen karar fonksiyonu ile ayrılabilir (Cherkassky & Mulier, 2007: 418).

Optimal ayırıcı hiperdüzlemi bulmak için

$$\frac{1}{2} (w \cdot w) \quad (3.15)$$

amaç fonksiyonunu minimize eden w, b değerlerinin

$$y_i[(w \cdot x_i) + b] \geq 1 \quad (3.16)$$

kısıtları altında bulunması gerekir. Gösterilen primal problemin çözümü $d + 1$ parametre içerir. Orta büyüklükteki boyutlar (d) için bu problem kuadratik programlama ile çözülebilir. Çok yüksek boyutlu uzaylarda ise uygulanabilir değildir. Bu nedenle primal problem, karmaşıklığı boyuta değil gözlem sayısına bağlı olan duale dönüştürülür. Çok yüksek uzaylarda bile dual problem standart kuadratik optimizasyon teknikleri ile kolayca çözülebilir. Optimal çözüme daima polinom zamanda ulaşır. Dual problemin zaman karmaşıklığı üst sınırı $\mathcal{O}(n^3)$, bellek karmaşıklığı üst sınırı ise $\mathcal{O}(n^2)$ dir (Alpaydın, 2009: 313).

Primalden duale iki adımda geçilebilir. İlk adımda Lagrange çarpanları kullanılarak kısıtsız optimizasyon problemi (Denklem 3.17) deki gibi oluşturulur.

$$L(w, b, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^n \alpha_i \{ y_i[(w \cdot x_i) + b] - 1 \} \quad (3.17)$$

α_i , ile gösterilen Lagrange çarpanlarının büyük bir büyük bir kısmının değeri sıfırdır ($\alpha_i = 0$). Bu gözlemler, $y_i[(w \cdot x_i) + b] > 1$ kısıtını sağlarlar yani ayırıcı hiperdüzlemden yeterince uzak mesafede bulunan ve ayırıcı hiperdüzlemin konumuna herhangi bir etkisi olmayan gözlemleri temsil ederler. Bunlar karar sınırını belirleyecek herhangi bir bilgi taşımadıklarından kaldırılırsalar bile yine aynı çözüm elde edilir.

İkinci adım ise (3.17) denklemindeki w ve b parametrelerini sadece α_i cinsinden gösterebilmek için (KKT) koşullarının kullanılmasıdır. Bu işlemden sonra

(3.17) denklemi sadece Lagrange çarpanlarına göre maksimizasyonu gerektiren dual probleme dönüşür (Abe, 2010: 18-19). KKT koşullarına göre çözümü gösteren w^*, b^*, α^* parametrelerinin

$$\frac{\partial L(w^*, b^*, \alpha^*)}{\partial b} = 0$$

$$\frac{\partial L(w^*, b^*, \alpha^*)}{\partial w} = 0$$

koşullarını sağlaması gerekir. Bu kısmı türev denklemlerinin çözülmesi sonucu optimal hiperdüzlemin şu özellikleri bulunur.

1. α_i^* katsayıları

$$\sum_{i=1}^n \alpha_i^* y_i = 0 \quad \alpha_i^* \geq 0 \quad (3.18)$$

kısıtını sağlamalıdır.

2. w^* vektörü, dolayısıyla optimal ayırıcı hiperdüzlem, eğitim kümesindeki gözlemlerin doğrusal birleşimidir. Yani:

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i \quad \alpha_i^* \geq 0 \quad (3.19)$$

Tamamlayıcı aylıklık koşuluna göre

$$\alpha_i^* [y_i (w^* \cdot x_i + b^*) - 1] = 0$$

eşitliği de sağlanmalıdır. (3.19) denkleminin (3.14) denkleminde yerine yazılmasıyla karar fonksiyonu $\alpha_1^*, \dots, \alpha_n^*$ ve b^* cinsinden elde edilir.

$$D(x) = \sum_{i=1}^n \alpha_i^* y_i (x \cdot x_i) + b^* \quad (3.20)$$

b^* parametresi destek vektörlerinin özel durumları kullanılarak bulunur. Destek vektörleri tam marj sınırının üstünde bulduklarından herhangi bir (x_s, y_s)

destek vektörü için $y_s[(w^* \cdot x_s) + b^*] = 1$ koşulu sağlanır. Bu koşul (3.20) denkleminde yerine konulursa

$$b^* = y_s - \sum_{i=1}^n \alpha_i^* y_i (x_i \cdot x_s)$$

elde edilir.

Karar fonksiyonundaki α_i^* değerlerini bulmak için dualin oluşturulması ve çözülmesi gerekir. Bunun (3.18) ve (3.19) de bulunan sonuçlar (3.17) deki Lagrange fonksiyonunda yerine konulur. Böylece Lagrange fonksiyonu daha açık olarak

$$L(w, b, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^n \alpha_i y_i x_i \cdot w - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i$$

biçiminde yazılabilir. (3.18) de gösterilen koşul ($\sum_{i=1}^n \alpha_i^* y_i = 0$) nedeniyle eşitliğin sağ tarafındaki üçüncü terim sıfır olur. (3.19) da bulunan sonucun Lagrange fonksiyonunda yerine yazılmasıyla

$$L(\alpha) = -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^n \alpha_i \quad (3.21)$$

elde edilir. (3.21) denkleminde gösterilen $L(\alpha)$ ifadesi, dual amaç fonksiyonudur ve $\alpha_1, \dots, \alpha_n$ parametrelerine göre maksimize edilmesi gerekir. Kısaca dual optimizasyon problemi:

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.22)$$

fonksiyoneli

$$\sum_{i=1}^n y_i \alpha_i = 0 \quad \alpha_i \geq 0 \quad (3.23)$$

kısıtları altında maksimize eden α_i parametrelerini bulmaktır

Dualin çözümünde karşılık gelen α_i^* değerleri sıfırdan büyük olan gözlemler, destek vektörleridir. Destek vektörleri karar sınırına en yakın gözlemler oldukları için sınıflandırılmaları en zor olan ve optimal ayırıcı hiperdüzlemin konumunu belirleyen gözlemlerdir (Cherkassky & Mulier, 2007: 419). Destek vektörleri sınıflandırma problemi ile ilgili tüm bilgiyi taşır. Eğer destek vektörleri haricindeki tüm gözlemler kaldırılrsa dahi yine aynı çözüm elde edilir. Bu özellik “çözümün seyrekliği (*sparsity*)” olarak bilinir ve hem uygulamada hem de algoritmanın analizinde önemli sonuçları vardır (Cristianini & Schölkopf, 2002: 40).

Vapnik tarafından,

$$E_n[\text{hata oranı}] \leq \frac{E_n[\text{Destek vektörlerinin sayısı}]}{n} \quad (3.24)$$

sınırının sağladığı ispatlanmıştır. (Denklem 3.24) de E_n operatörü n elemanlı eğitim kümeleri üzerindeki hata beklentisini gösterir (Cortes & Vapnik, 1995: 275). Bu sınır, problem boyutundan (d) bağımsızdır. Bu sınıra dayanarak daha az sayıdaki destek vektörü ile oluşturulabilen ayırıcı hiperdüzlemin daha iyi bir genelleme sağlayabileceği kabul edilir.

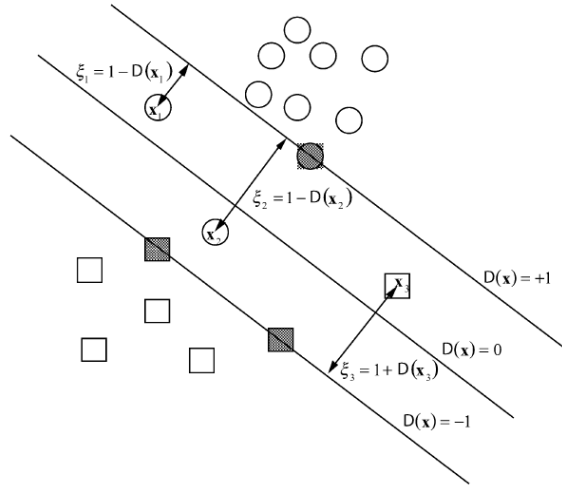
3.3.2.2 Doğrusal Yumuşak Marjlı Sınıflandırıcı

Doğrusal ayrılabilen gözlemler için oluşturulan kuadratik optimizasyon gösterimi, gözlemlerin doğrusal ayrılabilir olmadığı durumlarda kısıtlar sağlanmadığı için kullanılamaz. Bu durumda, yanlış sınıflandırılan her bir gözleme karşılık gelen α_i Lagrange çarpanları sonsuz değerini almaya eğilim gösterirler. Bu tür gözlemler, doğru sınıflandırma yapabilmek için α_i değerini arttırarak karar sınırı üzerinde güçlü bir etki oluşturmaya çalışırlar. α_i değerleri maksimum sınıra ulaştığında etkilerini artık arttıramaz ve karşılık gelen gözlemler yanlış sınıflandırılmaya devam eder. Böyle bir durumda, algoritma hemen hemen tüm gözlemleri destek vektörü olarak ele alır (Huang vd., 2006: 32). Bu nedenle, yöntemin bazı gözlemlerin yanlış sınıflandırmasına izin verecek şekilde değiştirilmesi gerekir.

Marja dayanan yöntemlerde iki tür sapma gözlenir. Birincisi, gözlem ayırıcı hiperdüzlemin yanlış tarafındadır ve yanlış sınıflandırılır. İkincisi, gözlem ayırıcı

hiperdüzlemin doğru tarafındadır ama marj içindedir (Alpaydın, 2009: 315). Yumuşak marj yaklaşımında marj içinde kalan gözlemlerin ayırıcı hiperdüzlemin yanlış tarafında olmaları göz ardı edilir. Yani sert marj sınıflandırıcılardaki kısıtlar gevşetilir.

Doğrusal ayrılamayan sapmaların sayısı yerine daha kolayca çözüme ulaştıran saplamaların toplamı minimize edilmeye çalışılır. Gözlemler için ampirik risk, $\xi_i = \max(1 - y_i f(x_i, \omega), 0)$ şeklinde tanımlanan “aylak (slack)” değişkenler kullanılarak bulunur (Cherkassky & Mulier, 2007: 425). Bu değişkenlerin toplamı marj sınırından toplam sapmayı verir (Şekil 36). Sadece yanlış sınıflandırılan gözlemler değil, daha iyi genelleme sağlanması için doğru sınıflandırılan fakat marj içinde bulunan gözlemler de cezalandırılır.



Şekil 36. Yumuşak Marj Sınıflandırıcı

Yumuşak marjlı optimal ayırıcı hiperdüzlem

$$\frac{1}{2}(w \cdot w) + \frac{C}{n} \sum_{i=1}^n \xi_i$$

fonksiyonunun

$$y_i[(w \cdot x_i) + b] \geq 1 - \xi_i$$

kısıtları altında minimize edilmesiyle bulunur. Amaç fonksiyonunda yer alan C parametresi, karmaşıklık ile doğrusal ayrılamayan gözlemlerin oranı arasındaki ödünleşimi kontrol eder ve model tasarlayıcısı tarafından belirlenir (Cortes & Vapnik, 1995: 286). C değerinin ∞ olması yanlış sınıflandırma yapılmasına izin verilmemesi anlamına geldiğinden doğrusal ayrılabilir olmayan gözlemler için uygun değildir. Bu durumlarda, problem sadece $C < \infty$ değerleri için uygun çözümü verir (Huang vd., 2006: 32).

Problemin duali:

Amaç fonksiyonu

$$\text{maks } L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

Kısıtlar

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C/n$$

şeklinde gösterilir. Doğrusal ayrılabilen durum ile aradaki tek fark, Lagrange çarpanlarına C/n üst limitinin eklenmesidir (Mavroforakis & Theodoridis, 2006: 673). Tüm ξ_i değerlerinin 0 olduğu durumda yumuşak marj problemi sert marj problemine dönüşür.

Yumuşak marj dualinin olası sonuçları (Abe, 2010: 24).

1. $\alpha_i = 0, \xi_i = 0, x_i$ gözlemi doğru sınıflandırılmıştır.
2. $0 < \alpha_i < C$ ise tamamlayıcı aylıklık koşuluna göre $y_i[w^T x_i + b] - 1 + \xi_i = 0$ ve $\xi_i = 0$ olmalıdır. Bu nedenle, $y_i[w^T x_i + b] = 1$ olacağından x_i destek

vektörüdür. $0 \leq \alpha_i \leq C$ eşitsizliğini sağlayan destek vektörlerine “*serbest destek vektörleri*” denilir.

3. $\alpha_i = C$, $y_i[w^T x_i + b] - 1 + \xi_i = 0$, $\xi_i \geq 0$ ise x_i destek vektörüdür. $\alpha_i = C$ eşitliğini sağlayan destek vektörleri “*sınırlandırılmış destek vektörü*” olarak adlandırılır. Marjın yanlış tarafında bulunurlar. Eğer $0 \leq \xi_i < 1$ ise x_i doğru sınıflandırılır; $\xi_i \geq 1$ ise yanlış sınıflandırılırlar.

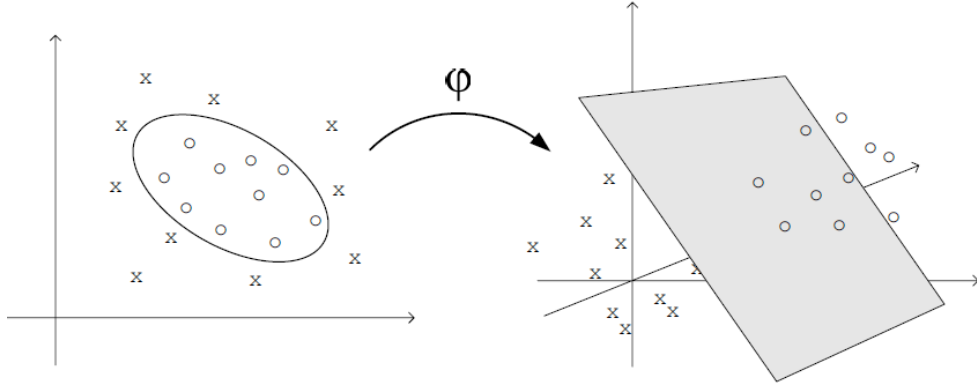
Karar fonksiyonu doğrusal ayrılabilir durumdakiyle aynıdır.

$$D(x) = \sum_{i=1}^n \alpha_i^* y_i (x \cdot x_i) + b^* \quad (3.25)$$

3.3.2.3 Doğrusal Olmayan DVM Sınıflandırıcıları

Yumuşak marj yaklaşımında kısıtlar gevşetilerek belli ölçüde hatalı sınıflandırmaya izin verilir. Fakat problem büyük ölçüde doğrusal ayrılabilir değil ve yanlış sınıflandırma hatası çok yüksek ise yumuşak marj işe yaramaz. Gerçek uygulamaların çoğunda doğrusal olmayan karar fonksiyonları oluşturmak gerekmektedir. Doğrusal olmayan karar fonksiyonları oluşturulabilirse doğrusal ayrılamayan eğitim kümeleri sınıflandırılabilir (Huang vd., 2006: 36-37).

Doğrusal ayrılamayan problemleri doğrusal ayrılabilir hale getirmek için gözlemler “*eşleme (mapping) fonksiyonları*” kullanılarak yüksek boyutlu öznitelik uzayına eşlenir (Wu vd., 2010: 1166). (Şekil 37) de görüldüğü gibi temel fikir, n boyutlu bir girdi uzayında doğrusal olmayan bir ayırıcı hiperdüzlemin daha yüksek boyutlu (N) öznitelik uzayında doğrusal bir hiperdüzleme eşlenebileceğidir. Ne yazık ki en uygun eşleme fonksiyonunu veren analitik bir yöntem yoktur. Bu nedenle eşleme fonksiyonları deneme yanılma yoluyla bulunmaya çalışılır.



Şekil 37. Eşleme Fonksiyonu

Eğitim kümesindeki her bir gözlem φ eşleme fonksiyonu kullanılarak

$$x = (x_1, x_2, \dots, x_n) \xrightarrow{\varphi} \varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_N(x))$$

biçiminde daha yüksek boyutlu (N) uzayda bir noktaya eşlenir. Eşleme sonucunda, öznitelik uzayında girdi uzayındaki gözlemlerin görüntüleri olan

$$\{ (\varphi(x_1), y_1), (\varphi(x_2), y_2), \dots, (\varphi(x_n), y_n) \}$$

kümesi elde edilir.

Örneğin 2 boyutlu uzaydan 5 boyutlu bir uzaya eşleme yapan φ

$$(x_1, x_2) \xrightarrow{\varphi} (x_1^2, x_1x_2, x_2^2, x_1, x_2)$$

fonksiyonu kullanılırsa öznitelik uzayında doğrusal olarak oluşturulan karar fonksiyonu, girdi uzayında tüm doğrusal veya kuadratik fonksiyon sınıflarını (doğrular, elipsler, hiperboller) içerecek kadar geniş hale gelir (Press vd., 2007: 890).

Bir girdi uzayını daha yüksek boyutlu öznitelik uzayına eşlerken dikkate alınması gereken iki temel husus aşağıda belirtilmiştir.

1. Seçilen $\varphi(x)$ eşleme fonksiyonu geniş bir hiperdüzlemler sınıfı oluşturabilmelidir.

2. Öznitelik sayısı (öznitelik uzayının boyutu) çok fazla olduğunda hesaplamaları yapmak çok zorlaşabilir.

n boyutlu girdi uzayında ikinci dereceden polinom olan bir karar fonksiyonu oluşturabilmek için öznitelik uzayının boyutu $\frac{n(n+3)}{2}$ olmaktadır. Örneğin 256 boyutlu girdi uzayında böyle bir karar fonksiyonu için 33.152 boyutlu bir öznitelik uzayı kullanmak gerekir. Bu büyüklükteki bir uzayda hesaplama maliyeti çok yüksektir. Aynı girdi uzayında 2.derece yerine daha yüksek bir dereceden karar fonksiyonu oluşturulmak istenirse çok daha yüksek boyutlu öznitelik uzaylarında çalışılması gerekir (Huang vd., 2006: 40).

Sert marjlı ve yumuşak marjlı sınıflandırıcılar için elde edilen karar fonksiyonu (Denklem 3.20, Denklem 3.25) öznitelik uzayı için tekrar yazılırsa

$$D(\varphi(x)) = \sum_{i=1}^N \alpha_i^* y_i (\varphi(x) \cdot \varphi(x_i)) + b^* \quad (3.26)$$

elde edilir. (Denklem 3.26) incelendiğinde karar fonksiyonunda gözlemlere ihtiyaç duyulan tek işlemin $\varphi(x) \cdot \varphi(x_i)$ iç çarpım işlemi olduğu görülür. Bu, karar fonksiyonunun doğrusal olmayan biçime dönüştürülmesini sağlayan temel özelliktir (Cristianini & Schölkopf, 2002: 39).

3.3.2.3.1 Kernel Fonksiyonları

Çok yüksek boyutlu uzaylarda iç çarpım yapmak hesaplama açısından çok zor olduğundan kernel fonksiyonları kullanılır.

Bir kernel

$$K(x_i, x_j) = \varphi^T(x_i) \cdot \varphi(x_j) \quad (3.27)$$

eşitliğini sağlayan fonksiyondur. (Denklem 3.27) incelendiğinde $K(x_i, x_j)$ kernel fonksiyonunun girdi uzayında işlem yaptığı görülür. Buradan yola çıkarak kernel

fonksiyonu kullanmanın esas avantajının $\varphi(x)$ eşlemesinin yapılma zorunluluğunun kalmaması olarak açıklanabilir. Öznitelik uzayında yapılması gereken $\varphi^T(x_i) \cdot \varphi(x_j)$ iç çarpımı kernel kullanılarak girdi uzayındaki eğitim gözlemleri üzerinde doğrudan yapılır. Bir kernel kullanarak sonsuz boyutlu bir öznitelik uzayında işlem yapacak bir DVM oluşturmak mümkündür.

“*Kernel hilesi (kernel trick)*”, $\varphi(x)$ eşlemesinin ne olduğunu bilme zorunluluğunun olmamasıdır. İhtiyaç duyulan tek şey herhangi bir eşlemeden oluşturulabilecek K_{ij} kernelinin hesaplanmasıdır (Abe, 2010: 26).

Uygun şartlar altında, öznitelik uzayındaki bir iç çarpım girdi uzayında eşdeğer bir kernele sahiptir. Örnek olarak iki boyutlu $x = (x_1, x_2)$ gözlemlerini altı boyutlu bir öznitelik uzayına

$$\varphi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

biçiminde eşleyen fonksiyon ele alınsın. Bu eşleme fonksiyonuna karşılık gelen kernel fonksiyonunun

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = (1 + x_i \cdot x_j)^2$$

olduğu gösterilebilir (Moguerza & Muñoz, 2006: 325). Bu örnek üç boyutlu $x = (x_1, x_2, x_3)$ gözlemler için

$$\varphi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

şeklindeki eşleme fonksiyonu kullanılarak genişletilebilir. Bu eşlemeye karşılık gelen kernel yine ikinci dereceden polinom kerneldir. Benzer yolla, İki boyutlu $x = (x_1, x_2)$ gözlemleri için

$$\varphi(x) = (1, \sqrt{3}x_1, \sqrt{3}x_2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1x_2, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, x_1^3, x_2^3)$$

eşleme fonksiyonuna eşdeğer kernelin üçüncü dereceden

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = (1 + x_i \cdot x_j)^3$$

polinom kerneli olduğu gösterilebilir (Osuna vd., 1997: 12). Sonsuz boyutlu öznitelik uzayına yapılan eşlemeye $x \in [1, \pi]$ olmak üzere

$$\varphi(x) = \left(\sin(x), \frac{1}{\sqrt{2}} \sin(2x), \frac{1}{\sqrt{3}} \sin(3x), \frac{1}{\sqrt{4}} \sin(4x), \dots \right)$$

fonksiyonu örnek olarak verilebilir. Bu sonsuz seri açılımına karşılık gelen kernel

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = \frac{1}{2} \log \left| \sin \left(\frac{x_i + x_j}{2} \right) / \sin \left(\frac{x_i - x_j}{2} \right) \right|$$

şeklinde gösterilen basit bir ifadeye sahiptir (Ivanciuc, 2007: 329).

Özetle, doğrusal ayrılamayan gözlemlerin sınıflandırılması için karar fonksiyonu

$$D(x) = \sum_{i=1}^n \alpha_i^* y_i K(x_i, x) + b \quad (3.28)$$

biçiminde gösterilebilir ve $\alpha_i^*, i = 1, \dots, n$ katsayıları, $(x_i, y_i), i = 1, \dots, n$ eğitim kümesi, bir iç çarpım kerneli (K), düzenleme parametresi (C) verildiğinde aşağıdaki kuadratik optimizasyon probleminin çözülmesiyle bulunur.

Amaç fonksiyonu

$$\text{maks } L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Kısıtlar

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C/n$$

b parametresi, (x_s, y_s) destek vektörlerinden birini göstermek üzere

$$b^* = y_s - \sum_{i=1}^n \alpha_i^* y_i K(x_i \cdot x_s)$$

denklemleri kullanılarak bulunur (Cherkassky & Mulier, 2007: 431).

3.3.2.3.2 Mercer Teoremi

Mercer Teoremi bir kernelin herhangi bir uzayda iç çarpım kerneli olup olmayacağını, dolayısıyla kullanıma uygun olup olmadığını gösterir. Fakat $\varphi(x)$ eşleme fonksiyonlarının nasıl oluşturulacağı konusunda herhangi bir bilgi vermez.

$K(x, x')$, $a \leq x \leq b$ ve $a \leq x' \leq b$ kapalı aralıklarında tanımlı, sürekli ve simetrik bir kernel olsun. Bu kernel

$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x) \varphi_i(x'), \quad \lambda_i > 0, \quad \forall i \text{ için}$$

şeklinde açılabilir. Bu açılımın geçerli olması ve düzgün yakınsayabilmesi için gerek ve yeter şart

$$\int_a^b \varphi^2(x) dx < \infty$$

koşulunu sağlayan tüm $\varphi(\cdot)$ için

$$\iint_a^b K(x, x') \varphi(x) \varphi(x') dx dx' \geq 0$$

olmasıdır. φ_i fonksiyonlarına açılımın özfonksiyonları (eigenfunction), λ_i değerlerine ise özdeğerleri (eigenvalue) denilir. Tüm özdeğerlerin pozitif olması $K(x, x')$ kernelinin pozitif tanımlı olması demektir (Haykin, 1999: 353).

Kernel değerlerinin tüm gözlemler üzerinde hesaplanmasıyla Gram matrisi elde edilir.

$$G = K(x_i, x_j) = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) \dots & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) \dots & K(x_2, x_n) \\ \vdots & \vdots & \vdots \\ K(x_n, x_1) & K(x_n, x_2) \dots & K(x_n, x_n) \end{bmatrix}$$

Bu matris simetriktir. Eđer Gram matrisi “pozitif tanımlı (*positive definit*)” ise bu kernel de pozitif tanımlıdır. Simetrik, pozitif tanımlı bir matris, herhangi bir uzayda kernel matrisi yani iç çarpım matrisi olarak kabul edilir (Huang vd., 2006: 42).

3.3.2.3.3 Kernel Tipleri

Kerneller bir benzerlik ölçütü olarak değeriendirilirler. Uygulama açısından x ve y birbirine benzer gözlemler ise $K(x, y)$ büyük değeri alır. Bundan yola çıkarak uygun tanımlanmış kerneller oluşturularak, uygulamayla ilgili önsel bilgiler modelde kullanılabilir (Cortes & Vapnik, 1995: 285).

Birçok öğrenme algoritmasında gözlemler vektörler halinde gösterilir. Bu gözlemler üzerinde benzerlik ölçütü olarak iç çarpım veya Euclid uzaklığı kullanılır. Bir kernel fonksiyonu ise bunun ötesine geçmeye izin verir. Örneğin G_1 ve G_2 iki çizge (graph) olduğunda $K(G_1, G_2)$ değeri, bu iki çizge açıkça vektör şeklinde gösterilmeden de hesaplanabilir (Alpaydın, 2009: 309).

Kernel fonksiyonları, metin belgeleri, resimler veya DNA zincirleri gibi farklı türdeki gözlemleri öğrenme algoritmasının çalıştırılabileceği uzaya eşleyebilirler. Bu kernel yaklaşımının getirdiği önemli yeniliklerden birisidir (Cristianini & Schölkopf, 2002: 40).

Maksimum marj yöntemi, kullanılan kernelden bağımsızdır. Kernel seçimi maksimum marj belirlendikten sonra alan bilgisine veya başka tekniklere göre yapılır. Alan uzmanları gözlemler arasındaki etkin benzerlik ölçütlerini bilirler. Bu nedenle problem alanlarına özgü birçok kernel geliştirilmektedir.

Yaygın olarak kullanılan kernel tipleri şunlardır (Ivanciuc, 2007: 329-332):

Doğrusal Kernel

x ve y değerlerinin iç çarpımı doğrusal kerneli tanımlar.

$$K(x_i, x_j) = x_i \cdot x_j$$

Eđitim kümesinin dođrusal olmama durumunu test etmek ve dođrusal olmayan kernellerin kullanılması sonucu sınıflandırma performansındaki artışı takip etmek için kullanılır.

Polinom Kernel

Polinom kernel, dođrusal olmayan ilişkileri modellemek için basit ve etkin bir yol sağlar.

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^d$$

d parametresi kernelin derecesini göstermektedir. Derece arttıkça aşırı uyum riski artar.

Dairesel Tabanlı Kernel

Yaygın olarak kullanılan dairesel tabanlı kernel

$$K(x_i, x_j) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

biçiminde gösterilir. σ parametresi ayırıcı hiperdüzlemin şeklini kontrol eder.

Sigmoid Kernel

Yapay sinir ağlarında en sık kullanılan etkinlik fonksiyonu olan hiperbolik tanjant (\tanh) fonksiyonuna karşılık gelen kernel

$$K(x_i, x_j) = \tanh(ax_i \cdot x_j + b)$$

biçiminde gösterilebilir.

Kerneller birleştirilerek yeni kerneller oluşturmak mümkündür. $K_1(x, y)$ ve $K_2(x, y)$ geçerli kerneller ve ϑ bir sabit olmak üzere

$$K(x, y) = \begin{cases} \vartheta K_1(x, y) \\ K_1(x, y) + K_2(x, y) \\ K_1(x, y) \cdot K_2(x, y) \end{cases}$$

kernelleri de geçerlidir (Campbell & Ying, 2011: 46).

3.3.2.4 Çoklu ve Dengesiz Sınıfların Sınıflandırılması

DVM yöntemi çoklu sınıflara, gelecek gözlemlerin eğitim gözlemlerinden daha farklı dağıldığı durumlara (dengesiz dağılım) ve yanlış sınıflandırma maliyetlerinin eşit olmadığı durumlara uyarlanabilir. Bu durumlarla pratik uygulamalarda sıklıkla karşılaşmaktadır.

Çoklu sınıflar: DVM yöntemini sınıf sayısının ikiden fazla olduğu problemlere uygularken yaygın olarak kullanılan dönüştürme işlemi, problemi ikili sınıflandırma alt problemleri halinde ayırmaktır (Fradkin & Muchnik, 2006: 6). J sınıflı bir problem için J adet ayrı ayrı ikili sınıflandırma problemi oluşturulur. Her bir problemde, eğitim gözlemleri, $j = 1, \dots, J$ olmak üzere, j sınıfında olanlar ve j sınıfında olmayanlar biçiminde ayrılır.

Dengesiz dağılımlar ve eşit olmayan maliyetler: DVM amaç fonksiyonu, dengesiz dağılımları ve farklı maliyetleri dikkate alacak şekilde değiştirilebilir. Her bir sınıf için farklı önsel olasılıkları ve yanlış sınıflandırma maliyetlerini hesaba katmak için iki farklı ceza parametresi (C) kullanılır. Böylece amaç fonksiyonu

$$\frac{1}{2} w \cdot w + C^+ \sum_{i \in +1 \text{ sınıfı}} \xi_i + C^- \sum_{j \in -1 \text{ sınıfı}} \xi_j$$

haline gelir. C^+ ve C^- ceza parametrelerini, p^+ ve p^- gelecek gözlemler için önsel sınıf olasılıklarını, p_s^+ ve p_s^- ise eğitim gözlemleri için önsel sınıf olasılıklarını göstermek üzere

$$C^+ = \text{Maliyet}(\text{yanlış negatif})p^+p_s^-$$

$$C^- = \text{Maliyet}(\text{yanlış pozitif})p^-p_s^+$$

şeklinde hesaplanır. Dual fonksiyon C parametresine bağlı olmadığı için değiştirilmez. Tek değişiklik kısıtların (Denklem 3.29) de görüldüğü gibi değiştirilmesidir (Cherkassky & Mulier, 2007: 431).

$$\begin{cases} 0 \leq \alpha_i \leq C^+, & i \in +sınıflı \\ 0 \leq \alpha_j \leq C^-, & j \in -sınıflı \end{cases} \quad (3.29)$$

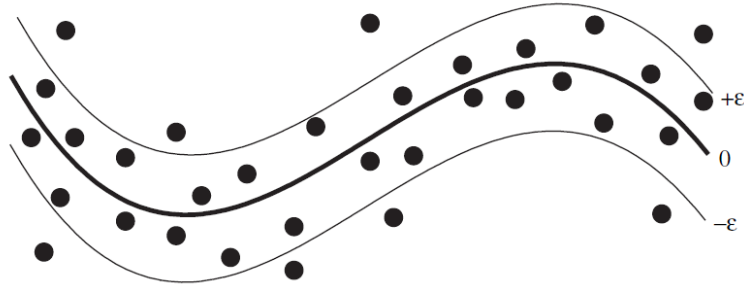
3.3.3 DVM Regresyonu

Regresyon modelinin ampirik riski, n gözlemden oluşan Z_n eğitim kümesi ve ε -duyarsız kayıp fonksiyonu için

$$R_{amp}(\omega, Z_n) = \frac{1}{n} \sum_{i=1}^n L_{\varepsilon}(y_i, f(x_i, \omega)) \quad (3.30)$$

şeklinde yazılır.

Belirlenen ε değerine kadar hatalar tolere edilir (Şekil 38). Bu değerden büyük hataların etkisi karesel değil doğrusaldır. Bu nedenle DVM regresyonu sağlamdır (Alpaydın, 2009: 329).



Şekil 38. Regresyon için Duyarsız Bölge

Sınırlı sayıdaki gözlemler için, marj tabanlı kapasite kontrolü yoluyla (ε -duyarsız bölgenin genişliği ayarlanarak) ampirik riskin (Denklem 3.30) minimize edilmesi iyi modeller oluşturulabilmesini sağlar. Regresyon probleminde, bir eğitim kümesi için ε parametresini ayarlamak, sınıflandırmada marjın ayarlanmasına benzer (Ivanciuc, 2007: 297).

ε -duyarsız kayıp, eğitim gözlemlerinin ε -duyarsız bölgeden ne kadar saptıklarını gösteren negatif olmayan aylak değişkenler (ξ_i ve ξ_i^*) kullanılarak tanımlanır. Bu durumda DVM regresyon probleminde

Amaç fonksiyonu

$$\min \frac{1}{2}(w \cdot w) + \frac{C}{n} \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (3.31)$$

Kısıtlar

$$\begin{cases} y_i - (w \cdot x_i) - b \leq \varepsilon + \xi_i \\ (w \cdot x_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, n \end{cases} \quad (3.32)$$

biçimindedir (Cherkassky & Mulier, 2007: 441). Kısıtları doğrusal olan bir kuadratik optimizasyon problem olduğundan dual biçime Lagrange fonksiyonu ve KKT koşulları kullanılarak dönüştürülebilir.

ε değeri ve C düzenleme parametresi verildiğinde doğrusal DVM için α_i ve β_i katsayıları

$$L(\alpha_i, \beta_i) = -\varepsilon \sum_{i=1}^n \alpha_i + \beta_i + \sum_{i=1}^n y_i (\alpha_i - \beta_i) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \beta_i)(\alpha_j - \beta_j)(x_i \cdot x_j)$$

kuadratik fonksiyonunu

$$\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i, \quad 0 \leq \alpha_i \leq C/n, \quad 0 \leq \beta_i \leq C/n$$

kısıtları altında maksimize eden optimizasyon probleminin çözülmesiyle bulunur. Bu problemin çözümünden elde edilen α_i^* ve β_i^* değerleri DVM regresyon fonksiyonunu (Denklemler 3.32-3.33) verir (Suykens vd., 2002: 54).

$$f(x) = \sum_{i=1}^n (\alpha_i^* - \beta_i^*)(x_i \cdot x) + b \quad (3.33)$$

(3.33) denklemindeki destek vektör regresyonu gösteriminde eğitim gözlemlerinin sadece küçük bir bölümü sıfırdan farklı katsayılara sahiptir. ε -duyarsız bölgenin tam üstünde veya dışında kalan gözlemler destek vektörleridir.

Kaydırma terimi (b), (x_s, y_s) herhangi bir destek vektörünü göstermek üzere;

$$b^* = y_s - \sum_{i=1}^n (\alpha_i^* - \beta_i^*) (x_i \cdot x_s)$$

denklemleriyle bulunur.

Doğrusal regresyon gösteriminin doğrusal olmayan destek vektör regresyonuna dönüştürülmesi kernel hilesi kullanılarak başarılabilir. Bu durumda regresyon fonksiyonu $0 \leq \alpha_i^* \leq C/n$ ve $0 \leq \beta_i^* \leq C/n$ olmak üzere

$$f(x) = \sum_{i=1}^n (\alpha_i^* - \beta_i^*) K(x_i, x) + b \quad (3.34)$$

biçiminde yazılabilir (Gunn, 1998: 35). Doğrusal olmayan DVM regresyon tahmininin kalitesi, uygun ε, C parametrelerinin ve kernelin seçimine bağlıdır.

3.3.4 DVM Optimizasyon Yöntemleri

(Tablo 10), örneklem büyüklüğü ile K kernel matrisini saklamak için gereken bellek alanı arasındaki ilişkiyi göstermektedir (Steinwart & Christmann, 2008: 422).

Tablo 10. Örneklem Büyüklüğü ve Bellek Alanı İlişkisi

Örneklem Büyüklüğü (n)	Depolama Alanı $4n(n + 1)$
100	40 KB
1000	4 MB
10.000	400 MB
100.000	40 GB
1.000.000	4 TB

Buna göre 100.000 gözlemden oluşan bir eğitim kümesinin kernel matrisini saklamak için 40 GB bellek gerekmektedir. Bu büyüklükteki hesaplamaların bugünkü standart bilgisayarlarda yapılması mümkün değildir. Bu nedenle, büyük boyutlu veri kümeleri üzerinde optimizasyon yapabilmek için yeni yöntemler getirilmiştir (Huang vd., 2006: 57).

DVM için kullanılan optimizasyon yöntemlerinden birisi olan “*Chunking*” nihai modele sadece destek vektörlerinin katkı verdiği gözlemi üzerine dayalıdır. Bu nedenle, *Chunking* yönteminde verinin keyfi bir alt kümesi DVM çözümünü oluşturmak için kullanılır. Sonra sadece destek vektörleri kümede bırakılarak geri kalan gözlemler kümeden atılır. Alt kümeyi tamamlamak için ek gözlemler dahil edilerek yeni bir çözüm belirlenir. Bu işlem, her bir gözlem için KKT koşulları sağlanıncaya kadar tekrar ettirilir. *Chunking* yaklaşımı destek vektörleri için kernel matrisi ana bellekte saklanabildiği sürece çalışır (Shawe - Taylor & Sun, 2011: 3613). Eğer bu mümkün değilse Decomposition gibi alternatif yöntemler gerekir.

“*Decomposition*” yaklaşımında gözlemler ve karşılık gelen parametreler “*çalışma kümesi (working set)*” adındaki sabit büyüklükteki birkaç kümeye ayrılır. Diğer parametreler sabitlenerek her bir çalışma kümesi üzerinde optimizasyon işlemi yapılır (Osuna vd., 1997: 24). Yaygın olarak kullanılan DVM yazılımlarından SVMLight ve SVMTorch decomposition stratejisini kullanır.

“*SMO - Sequential Minimal Optimization*” algoritması (Platt, 1999) ise iki gözlemden oluşan çalışma kümeleri kullandığından decomposition yönteminin bir uç şekli olarak görülebilir. SMO iyi ölçeklenebilir bir algoritmadır ve *Chunking* yöntemine göre bellek ihtiyacı daha düşüktür. En yaygın olarak kullanılan DVM yazılımlarından birisi olan LIBSVM, SMO yönteminin farklı bir biçimini kullanır (Fan vd., 2005).

Öğrenme problemini değiştirerek optimizasyon problemini basitleştirmek mümkündür. Bu basitleştirme işlemi hata fonksiyonunu veya cezalandırma terimini

değiştirerek kısıt sayısını azaltmakla olur. Örneğin “*LSVM*” (Mangasarian & Musicant, 2001). yöntemi doğrusal eşitsizlik sistemlerinin çözümüne bağlı olan bir formülasyon kullanır. *LSVM*, doğrusal karar sınırları için milyarlarca gözlemden oluşan eğitim kümeleri üzerinde optimizasyon problemlerini çok kısa sürede çözebilmektedir (Bennet & Campbell, 2000: 6). Yöntemin eksi yönü ise öğrenme problemi üzerinde yapılan değişikliklerin genelleme performansını nasıl etkilediğinin halen tam olarak bilinmemesidir.

DÖRDÜNCÜ BÖLÜM

4. UYGULAMA

Bu bölümde, Yapay Sinir Ağları (YSA) ve Destek Vektör Makineleri (DVM) yöntemleri İstanbul Menkul Kıymetler Borsası (İMKB) Ulusal 100 endeksinin yön tahmininde kullanılmıştır. Bu amaçla sırasıyla teknik göstergeler, borsaların endeks değişim oranları, makroekonomik göstergeler, bahsedilen ilk üç veri kümesinin tamamı ve bu küme üzerinde öznitelik seçimi (feature selection) yapıldıktan sonra elde edilen verilerin yer aldığı 5 ayrı veri kümesi oluşturulmuştur. Bu veri kümelerinden hangisinin endeks yön tahmininde daha kullanışlı olduğu belirlenmeye çalışılmıştır. Ayrıca, YSA ve DVM yöntemlerinin bu sınıflandırma probleminde nasıl bir performans gösterdikleri incelenmiştir.

4.1 Uygulamanın Amacı ve Kapsamı

Hisse senedi getirilerinin tahmin edilebilirliği konusu, fiyat davranışlarını tam olarak belirleyebilecek bir yöntem bulunmadığından dolayı, finans çevrelerinde en çok araştırılan konu olarak karşımıza çıkmaktadır (Kutlu & Badur, 2009: 27).

Finansal piyasaların küreselleşmesiyle gelişmekte olan piyasalara artan bir ilgi vardır. Borsalar genellikle gelişmiş ve gelişmekte olan borsalar biçiminde iki gruba ayrılmaktadır. Gelişmekte olan ülkelerde uluslararası ticaretin hızla gelişmesi, serbestleşme çabaları, gelişmekte olan ülkelerin finansal piyasalarının hızlı gelişimi ve uluslararası portföy yöneticilerinin çeşitlendirme yapma gereksinimleri dikkatleri gelişmekte olan ülke piyasaları üzerine yoğunlaştırmıştır. Uluslararası Finans Kurumuna (IFC) göre, 1980'li yıllardan itibaren gelişmekte olan ülkelere giden portföy yatırımları artarak devam etmiştir. Yabancı portföy yatırımları, gelişmekte olan ülkelerin hisse senedi piyasalarının büyümesine önemli katkılar sağlamıştır (Sayılğan & Süslü, 2011: 74).

Hisse senetlerinin işlem gördüğü borsalar, dinamik, doğrusal olmayan, kaotik yapıda olduklarından bu piyasalarda tahmin yapmak zordur. Buna ek olarak borsalar üzerinde siyasi olaylar, şirketlerin politikaları, genel ekonomik durum, yatırımcı beklentileri, kurumsal yatırımcıların seçimleri, diğer borsaların hareketleri ve yatırımcıların psikolojileri etkili olmaktadır (Kara vd., 2011: 5311). Yüksek belirsizlik ve oynaklık nedeni ile hisse senedi yatırımları diğer tüm yatırım alanlarından daha fazla risk taşır.

Hisse senedi yatırımlarında, yapılan yatırımın performansının değerlendirilmesi büyük önem taşır. Performansın ölçülmesinde ise en basit ve aynı zamanda en temel araç olarak piyasa endekslerinden yararlanılmaktadır. Endeksin hareketine göre, yapılan yatırımın başarılı olup olmadığına karar verilmektedir. Bu nedenle sermaye piyasalarında, yatırımcılar için ister hisse senetlerinin işlem gördüğü borsa tarafından isterse bağımsız bir kuruluş tarafından düzenlensin piyasa endekslerine şiddetle ihtiyaç duyulmaktadır. Özellikle başka ülkelerde yatırım faaliyetlerinde bulunan uluslararası yatırımcılar için bu endekslerin önemi daha da fazladır (Dağlı, 2000: 204).

Borsa endeks değerinin ve yönünün tahmini finansal tahmin uygulamalarının en zorlarından birisi olarak görülmektedir. Endeks değerini çeşitli yaklaşımlarla tahmin eden çalışmalar olmasına rağmen bu çalışmaların birçoğu gelişmiş ülke borsaları üzerinedir. Türkiye gibi gelişmekte olan ülke borsalarının endeks yönünün tahmini konusunda ise yapılan çalışma sayısı çok sınırlıdır.

İMKB oynaklığı yüksek bir borsa olarak bilinmektedir. Yüksek getiri olasılığı olduğundan birçok yerli ve yabancı yatırımcının ilgisini çekmektedir. Ayrıca işlem gören şirket sayısı ve toplam işlem hacmi gün geçtikçe artmaktadır. İMKB-100 endeks hareket yönünün doğru tahmin edilmesinin yatırımcılara yatırım stratejilerini oluşturmalarında faydalı olacağı düşünülmektedir. Yatırımcılar potansiyel risklere ve spekülörlere karşı önlem alabilirler ve borsa endeks yönü tahminine göre yatırım yaparak kâr elde etme fırsatı yakalayabilirler.

Borsa endeks yönünün tahmini artma ve düşme sınıflarının olduğu ikili sınıflandırma problemidir. Sınıflandırma problemlerinde amaç mevcut gözlemlerden öğrenme yoluyla oluşturulan bir fonksiyon yardımıyla sınıfları birbirinden ayıracak ve test gözlemleri üzerinde de iyi sonuçlar verecek (genelleme yapabilecek) bir sınıflandırıcı oluşturmaktır.

Tezin uygulama kısmında kullanılacak olan YSA ve DVM sınıflandırma problemlerinde yaygın olarak kullanılan makine öğrenmesi yöntemleridir. Tezin konusu olan İMKB-100 endeks yönünün tahmin edilmesinde bu yöntemler kullanılarak hangisinin daha iyi sonuçlar verdiği araştırılmıştır.

Çalışmanın katkısı, İMKB-100 endeks yönünün YSA ve DVM yöntemleri ile tahmin edilebilir olduğunu göstermesi ve endeksin en iyi hangi veri kümesi ile tahmin edilebileceğini ortaya çıkarmasıdır.

4.2 Hisse Senedi Piyasalarının Tahmin Edilebilirliği

Uygun bilginin finansal varlıkların fiyatına yansiyabildiği bir piyasayı tanımlamak için kullanılan “*etkinlik (efficiency)*” kavramı finans biliminin merkezinde yer almaktadır. (Fama, 1970)’ya göre piyasanın etkinlik ölçütü; menkul kıymet fiyatlarının, piyasaya ilişkin bilgiler yoluyla tahmin edilebilirlik derecesidir. Yani bir piyasa, mevcut bilgiler fiyatlara yansıdığı ölçüde etkin olmaktadır. Ekonomik analizde kabul edilen en önemli varsayımlardan birisi olan “*Etkin Piyasalar Hipotezi*” piyasa etkinliğini üç şekilde derecelendirmekte ve test etmektedir. Bunlar; *Zayıf Form*, *Yarı Güçlü Form* ve *Güçlü Form* piyasa etkinliğidir (Atan vd., 2009, 34-36).

- Zayıf formda etkinlik bir finansal varlığın geçmişine ait tüm bilgilerin finansal varlığın fiyatına yansımış olduğu haldir. Bu suretle bir finansal varlığın geçmişine ait fiyat hareketlerini inceleyerek finansal varlığın gelecekteki fiyatını belirlemek mümkün değildir.

- Yarı güçlü formda etkin piyasalar, finansal varlık ile ilgili olarak sadece geçmişteki bilgiler değil, aynı zamanda kamuya verilmiş mevcut tüm bilgilerin finansal varlıkların fiyatlarına yansıdığı piyasalardır.
- Güçlü formda etkin piyasa hipotezi finansal varlık fiyatlarına, kamuya açıklanan, açıklanmayan ve özel tüm bilgilerin yansıdığını ileri sürer. Bu nedenle hiçbir yatırımcı normalin üstünde bir getiri elde etmesi sağlayacak özel bir bilgiye sahip değildir. Bilgiler bütün yatırımcılar için aynı anda ve kolaylıkla elde edilebilir özelliğe sahiptir.

Ancak gerçek hayatta bilgi akışı kolay olmamakta, ekonomi ile ilgili tüm bilgilere ilgililer kolayca ulaşamamaktadır. Bu durumda ekonomik verilere ulaşanlar ile ulaşamayan kişiler arasında önemli farklılıklar oluşmakta, haksız rekabet koşulları ortaya çıkmaktadır. Bu noktada, bilgi yönetiminin önemi ortaya çıkmaktadır. Daha tecrübeli, bilgili yatırımcılar ve bilgi teknolojilerindeki ilerlemeler borsayı daha etkin kılmaktadır.

Fama (1970)' ya göre bütün bilgilerin fiyatlara yansıdığı etkin bir piyasada, fiyatta herhangi bir değişme oluyorsa, bunun tek nedeni beklenmeyen yeni bir bilginin piyasaya ulaşmasıdır. Böyle bir bilgi karşısında menkul kıymetin fiyatı aşağı ve yukarı doğru hareket halinde olacaktır. Aşağı ve yukarı hareket, piyasaya gelen fiyat lehine önemsiz olayların sayıca çokluğu ve önemli aleyhteki olayların sayıca azlığıyla dengelenecektir (Demirelli vd., 2010: 54).

(Aygören, 2008) tarafından yapılan çalışmada İMKB fiyat davranışı incelenmiş ve Etkin Piyasa Hipotezi'ni destekleyici bir sonuç elde edilememiştir.

(Eken & Adalı, 2008) yaptıkları çalışmada İMKB'nin klasik Etkin Piyasa Hipotezi' ne göre zayıf formda etkin olup olmadığını araştırmışlardır. Araştırma sonuçları uzun dönemde geçmiş fiyat bilgileri ile bugünün fiyat bilgileri arasında istatistiksel olarak anlamlı pozitif ilişkinin bulunduğunu göstermekle beraber kısa

dönemlerde, özellikle günümüze yaklaştıkça, ilişkinin azaldığı ya da kaybolduğu yönündedir. Bu durum belirli dönemlerde belirli hisse senedi ve endeks değerleri için piyasanın zayıf formda etkin olduğu şeklinde yorumlanmıştır.

(İçke & Aytürk, 2011) gelişmiş ve gelişen hisse senedi piyasalarında bazı dönemlerde gözlenen Fiyat–Kazanç Oranı Etkisi’ nin Nisan 2001–Mart 2009 döneminde İMKB’de geçerliliğini araştırmışlardır. Analiz sonucunda, fiyat–kazanç oranına dayanan yatırım stratejisinin istatistiksel olarak anlamlı normalüstü getiri sağlamadığı ve belirtilen dönemde İMKB’de Fiyat–Kazanç Oranı Etkisi’ nin geçerli olmadığı tespit edilmiştir. Çalışmada bunun İMKB’nin zayıf formda etkin bir piyasa olduğuna ilişkin bir kanıt niteliği taşıdığı belirtilmiştir.

Hisse senedi fiyatları ile genel ekonomik durum arasındaki ilişki uzun yıllardan beri çeşitli ekonomi ve finans uzmanlarının araştırmalarına konu olmuştur. Bazı araştırmacılar da olaya Etkin Piyasa Hipotezi açısından yaklaşarak hisse senedi fiyatlarının gelecekle ilgili bütün beklentileri yansıttığını ve bu yüzden de geçmiş ekonomik verilerle gelecekteki fiyat değişimlerinin tahmin edilmesinin mümkün olmayacağı görüşünü savunmuşlardır. Diğer bir ifade ile bu araştırmacılar etkin bir piyasada hiçbir yatırımcının geçmiş fiyat hareketlerini analiz ederek ortalama piyasa getirisi üzerinde bir getiri elde edemeyeceğini ileri sürerler. Bazı araştırmacılar ise ekonomik göstergelerin yardımı ile hisse senedi fiyatlarında meydana gelebilecek bir artış veya azalışın önceden tahmin edilebileceğini savunmuşlardır.

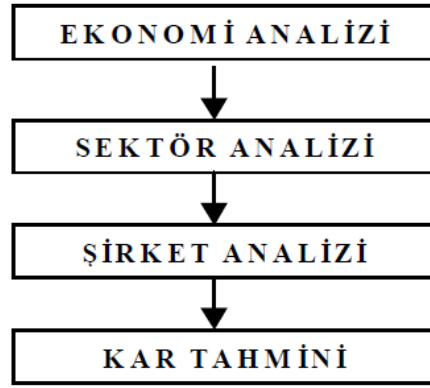
Bu bölümde hisse senedi piyasalarında tahmin yapmak için kullanılan çeşitli yöntemler kısaca açıklanarak İMKB ve diğer borsalar üzerinde yapılmış çalışmalara ait literatür taraması sunulacaktır.

4.2.1 Temel Analiz

Hisse senedi yatırımlarının performansını etkileyen unsurları dikkate alarak hisse senedinin gerçek değerini bulmaya ve bulunan bu değeri piyasa fiyatları ile karşılaştırarak alım - satım kararını vermeye yarayan analiz yöntemi "*Temel Analiz*"

olarak adlandırılır. Temel analize göre, hisse senedinin piyasa fiyatı gerçek değerinin altında olduğunda alış, üstünde olduğunda ise satış kararı verilir.

Temel analiz ekonomi, endüstri ve firma analizi olmak üzere üç aşamalı bir analizdir (Şekil 39). Ekonominin farklı dönemlerde gösterdiği eğilimlerin belirlenmesi, çeşitli sektörlerin bu gelişmelerden nasıl etkilenecekleri ve buna bağlı olarak işletmelerin gelecekteki nakit akımlarının tahmini yapılır.



Şekil 39. Temel Analiz Aşamaları (TSPAKB, 2011: 10)

Ekonomik gelişmelere karşı sermaye piyasası bazı dönemlerde aşırı tepki verebilmektedir. Makroekonomik değişkenlerle hisse senetleri bazen pozitif yönde bazen de negatif yönde hareket edebilirler. Makroekonomik olarak meydana gelen değişme ve gelişmeler, bir ekonomide faaliyet gösteren bütün işletmeleri etkilemektedir. Dolayısıyla bu faktörler, hisse senedi fiyatlarının topluca artma veya düşme eğilimine girmesine neden olurlar. Bununla birlikte, her bir firmanın performansı ile makroekonomik gelişmeler arasındaki ilişkinin yönü ve derecesi, teknoloji ve pazar şartları gibi faktörlerden ötürü az ya da çok farklılaşabilmektedir (Zügül & Şahin, 2009: 2-3).

Makroekonomik göstergeler ile İMKB endeksi arasındaki ilişkilerin incelendiği çalışmalar aşağıda sıralanmıştır.

(Türsoy vd., 2008) tarafından yapılan çalışmada 13 makroekonomik değişkenin hisselerin getirileri üzerindeki etkileri ölçülmüştür. Kullanılan temel ekonomik göstergeler: *para arzı(M2)*, *sanayi üretimi*, *ham petrol fiyatları*, *tüketici fiyat endeksi*, *ithalat*, *ihracat*, *altın fiyatları*, *döviz kuru*, *faiz oranları*, *yurtiçi milli hâsıla*, *yabancı rezerv*, *işsizlik oranı*, *pazar baskı endeksi* değişkenleridir.

(Erdem vd., 2005), makroekonomik değişkenlerin İMKB endeksi üzerindeki etkisini incelemiştir. Çalışmaya göre incelenen makroekonomik değişkenlerin içinde sadece *enflasyon*, *faiz oranı*, *döviz kuru* İMKB getirisini etkilemektedir.

(Zügül & Şahin, 2009) tarafından yapılan çalışmada Ocak 2004 - Aralık 2008 dönemi aylık verileri kullanılarak, İMKB-100 Endeksi ile *dolar döviz kuru*, *M1 para arzı*, *faiz oranı* ve *tüketici fiyat endeksi* makroekonomik değişkenleri arasında bir ilişki olup olmadığını incelenmiştir. Elde edilen sonuçlara göre M1 para arzı, döviz kuru ve faiz ile hisse senedi getiri endeksi arasında negatif yönlü bir ilişki olduğu, buna karşılık enflasyon oranıyla İMKB-100 Endeksi arasında pozitif yönlü bir ilişki olduğu görülmüştür.

4.2.2 Teknik Analiz

Teknik analiz, geçmiş dönemde gerçekleşen piyasa hareketlerinin incelenerek geleceğe yönelik olarak fiyat tahminlerinin yapıldığı analiz yöntemidir. Teknik analizin mantığına göre geçmişte meydana gelen bir hareket gelecekte tekrar meydana gelecektir.

Teknik göstergelerle ilgili temel kavramlar şu şekilde özetlenebilir (Çetinyokuş & Gökçen, 2002: 48-49).

- **Trend:** Trend kısaca aşağı ya da yukarı yönü temsil eder. Tepe ve dip noktalarıyla trendin yönü belirlenebilir. Yükselen bir trendde her yeni tepe ve dip bir öncekinden daha yukarıda oluşur. Düşen trend için ise bunun tam tersi geçerlidir. Altı aydan uzun oluşan trend, uzun dönemli

trend, 1 aydan 6 aya kadar olan trend orta, 1 aydan az sürene ise kısa dönemli trend adı verilir.

- **Destek - Direnç:** Destek geçmişte oluşmuş bir dip, direnç ise geçmişte oluşmuş bir tepedir. Bu noktalarda oluşan hareketler çok önemlidir. Destek noktası aşağı doğru kırıldığında yeni bir aşağı trend başlar. Aşağı kırılmış olan bir destek noktası ise ileride bir direnç noktası haline gelir. Fiyat destek noktasından geri dönerse aşağı trend sona erer. Direnç noktalarında da eğer fiyatlar bu seviyeyi aşarsa yukarı trend devam eder. Eğer fiyat o noktadan dönerse yukarı trend başarısız olmuştur ve fiyatlar geri döner. Direnç noktalarındaki işlem hacimleri de önemlidir. Yüksek işlem hacmiyle direncin aşılması sağlıklı bir yükselişi de beraberinde getirir. Bir destek ve direnç noktası ne kadar çok denenmişse o kadar önemli hale gelir.
- **Trend Çizgileri:** Yukarı trend çizgisi, yükseliş süresince oluşan dip noktaların birleştirilmesiyle çizilir. Ancak, trend çizgisinin geçerliliği için fiyatların trend çizgisini birkaç kez denemesi ve tepki vermesi gerekir. Bir trend çizgisi ne kadar çok denenirse o kadar güçlü hale gelir. Güçlü bir trend çizgisinin kırılmasıyla yeni trend başlar.

Fiyat trendlerini önceden görmeye yönelik çalışmalar, teknik analizin en önemli bölümünü teşkil eder. Orta ve uzun dönemli trendlerin teşhisi bugün doğruya çok yakın olarak önceden görülebilmektedir. Fakat vade süresi kısaldıkça yapılan tahminlerde hata oranı artmaktadır. Kısa vadeli teknik göstergelerin yardımıyla yön tahmininde destekleyici bilgiler aranmaktadır. Teknik göstergelerle hisse senedinin gitmekte olduğu yön görülebilir ve gücü ölçülebilir.

Teknik göstergeler osilatörler ve indikatörler şeklinde ikiye ayrılır (Kaynak: <http://www.witforex.com>):

1. **Osilatörler:** Alış - satış taleplerinin fiyat hareketi ile ilişkisini belirlemeye çalışırlar. Özellikle belli bir trendin olmadığı ve yatay yönde ilerleyen bir piyasada iyi sonuçlar verir. Belli bir sıfır çizgisinin üstünde ve altında hareket eden osilatörlerde bu çizginin üstüne çıkarken alış, altına inerken satış yapılması gerektiği mantığı geçerlidir.
2. **İndikatörler:** Özellikle trend yapan piyasada daha çok kullanılan indikatörler, fiyatların geçmişte yaptığı hareketlerin modellenmesi ile geleceğe ilişkin tahmin yapmak üzere kullanılan göstergelerdir.

Piyasanın değişik özellikler gösterdiği durumlarda bir gösterge türü iyi sonuç vermekteyken, diğeri işe yarayamayabilir. Osilatörler veya indikatörler tek başlarına mükemmel olmadıklarından farklı türdeki göstergelerden birkaç tanesini birlikte kullanmak gerekir.

Verdikleri bilgi açısından ise teknik göstergeler dörde ayrılır:

1. **Momentum Göstergeleri:** Belirli bir zaman içinde fiyatların ne kadar hızlı değiştiğini ölçmeye yarar. Momentum göstergelerinden bazıları şunlardır: *Commodity Channel Index, Moving Average, Convergence-Divergence, Mass Index, Momentum, Relative Strength Index, Stochastic Oscillator, TRIX, William's %R.*
2. **Trend Göstergeleri:** Bu türdeki göstergeler mevcut trendin nereye kadar devam edeceğini ve trend dönüşlerini anlamanın yanı sıra trend ile ilgili çeşitli bilgileri elde etmeye çalışırlar. *Moving Average, Convergence-Divergence, Parabolic SAR, R Squared, Standart Error* göstergeleri örnek olarak verilebilir.
3. **Oynaklık Göstergeleri:** Fiyatlardaki iniş çıkışların ne derece agresif olduğu, büyük oranlarda hareket ettiği ile ilgili göstergelerdir. Bu gruptaki göstergelerden bazıları şunlardır: *Avarage True Range, Bollinger Bands,*

Commodity Selection Index, Moving Avarage, Chaikin's Volatility.

4. **Güç Göstergeleri:** Piyasanın gücünü ölçmeye yarayan göstergelerdir. Güç göstergelerine örnek olarak *Accumulation / Distribution, Demand Index, Chaikin Money Flow, Money Flow Index, Price Volume Trend, Volume Rate of Change* göstergeleri gösterilebilir.

Yukarıda yapılan genel değerlendirmelerin ışığı altında Temel ve Teknik Analiz yöntemleri arasındaki farklılıklar şu şekilde sıralanabilir (TSPAKB, 2011: 9):

- Temel analiz teknik analiz ile karşılaştırıldığında daha zor bir analiz tekniğidir.
- Temel analizin yapılabilmesi için iyi bir ekonomi ve finans bilgisine ihtiyaç duyulmaktadır.
- Temel analiz sonucunda elde edilen sonuçların yatırımcılar tarafından kullanılabilmesi, bu kişilerin ekonomi bilgi düzeylerine bağlı olacaktır.
- Teknik analizin temel mantığı “*tarih tekerrürden ibarettir*” düşüncesidir. Ancak etkin piyasalar açısından bu pek de doğru bir mantık değildir. Dolayısı ile teknik analiz etkin piyasa hipotezi ile zaman zaman çelişmektedir.
- Temel analize getirilen en önemli eleştirilerden birisi, hesaplama konusunda tam bir birlikteliğin olmamasıdır

Borsa endeks tahmininde teknik göstergeleri kullanan çalışmalar ile ilgili literatür taraması aşağıda sunulmuştur.

(Kim & Han, 2000) tarafından Kore Borsası (KOSPI) endeksinin tahmini için yapılan çalışmada *Stochastic %K, Stochastic %D, Stochastic slow %D, Momentum,*

ROC, William's %R, A/D, Disparity 5 günlük, Disparity 10 günlük, OSCP, CCI, RSI teknik göstergeleri kullanılmıştır.

(Kara vd., 2011) yaptıkları çalışmada İMKB endeks yönünün tahmini için 10 teknik gösterge kullanılmıştır. Bu teknik göstergeler şunlardır: *Basit 10 günlük hareketli ortalama, Ağırlıklandırılmış 10 günlük hareketli ortalama, Momentum, Stochastic K%, Stochastic D%, RSI, MACD, William's R%, A/D Oscillator, CCI.*

(Diler, 2003) İMKB-100 endeksinin günlük olarak yönünü tahmin etmek amacıyla oluşturduğu modelde *Basit hareketli ortalama, Ağırlıklandırılmış hareketli ortalama, Momentum, Stokastik, RSI, MACD* teknik göstergelerini kullanmıştır.

(Dunis vd., 2012) tarafından yapılan çalışmada Madrid borsasının haftalık yön değişimini tahmin etmek için *MACD ve RSI* teknik göstergeleri kullanılmıştır.

4.2.3 İstatistiksel Yöntemler

Sermaye piyasalarında tahmin yapmak için istatistiksel yöntemleri kullanan çalışmalar aşağıda sunulmuştur.

(Gençtürk, 2009) yaptığı çalışmada makroekonomik verileri kullanarak *çoklu doğrusal regresyon analizi* metoduyla krizlerin yaşandığı dönem ile krizlerin yaşanmadığı dönemde, hisse senedi fiyatları ile makroekonomik faktörler arasındaki ilişkiyi belirlemeye çalışmıştır. Araştırma sonucunda, kriz dönemlerinde İMKB endeksini etkileyen makroekonomik değişkenler *tüketici fiyat endeksi ile para arzı* olarak çıkmıştır. Krizlerin yaşanmadığı dönemde; *sanayi üretim endeksi, tüketici fiyat endeksi, altın, dolar, para arzı ve hazine bonusu faiz oranı* ile endeks arasında anlamlı ilişki çıkmıştır. Bu dönemde altın, tüketici fiyat endeksi ve para arzı ile endeks arasında pozitif yönlü, sanayi üretim endeksi, dolar ve hazine bonusu faiz oranları ile endeks arasında negatif yönlü ilişki bulunmuştur.

(Altay & Satman, 2005) İstanbul Menkul Kıymetler Borsasında endeks tahmini için *doğrusal regresyon* yöntemini kullanmıştır. Araştırmada 02.01.1997-

28.02.2005 tarih aralığında *İMKB-100* ve *İMKB-30* endekslerine ait kapanış, en yüksek ve en düşük değerleri günlük haftalık ve aylık periyotlar halinde kullanılmıştır.

(Doğan & Yalçın, 2007) çalışmalarında döviz kuru değişimlerinin İMKB endeksi üzerindeki etkisini araştırmışlardır. Çalışmanın sonucunda *kur değişimleri* ile İMKB endeksi arasında yakın bir ilişki olduğu görülmüştür. Analiz aracı olarak *Vector Auto Regression (VAR)* modeli kullanılmıştır.

(Eryiğit, 2009) tarafından yapılan çalışmada İMKB endeks değerleri incelenerek dolar bazında petrol fiyatlarının borsa endeksinin değişiminde etkisi görülmeye çalışılmıştır. Modeli oluşturmak için *En Küçük Kareler Regresyon* yöntemi kullanılmıştır. Analizde *Türk Lirası cinsinden petrol fiyatı, dolar cinsinden petrol fiyatları ve TL Amerikan doları döviz kuru girdi* değişkenleri olarak kullanılmıştır.

İstatistiksel yöntemleri kullanarak İMKB ile diğer borsalar arasındaki karşılıklı bağımlılık derecesini araştıran çeşitli çalışmalar bulunmaktadır. Bu çalışmalar aşağıda özetlenmiştir.

(Vuran, 2010) tarafından yapılan çalışmada İMKB-100 endeksi ile gelişmiş ve gelişmekte olan bazı ülkelerinin borsa endeksleri arasındaki uzun vadeli ilişki 2006-2009 dönemine ait günlük veriler kullanarak *Johansen Eşbütünleşim (Cointegration) Analizi* ile test edilmiştir. Çalışmanın sonucunda belirtilen dönem için İMKB-100 endeksinin *FTSE-100, DAX, BOVESPA, Merval ve IPC* endeksleri ile uzun vadede ilişkili olduğu sonucuna ulaşılmıştır.

(Ozun, 2007) çalışmasında gelişmiş ülke borsalarının Brezilya ve Türkiye gibi gelişmekte olan ülke borsaları üzerindeki etkisini incelemiştir. *BOVESPA, İMKB-100, NIKKEI-225, FTSE-100, DAX, CAC-40, S&P-500, NASDAQ* borsalarının günlük kapanış değerleri *E-GARCH* metodu kullanılarak analiz

edilmiştir. Sonuç olarak İMKB endeksinin gelişmiş borsalardan zayıf pozitif yönde etkilendiği belirtilmiştir.

(Korkmaz vd., 2011) yaptıkları çalışmadaki *Granger nedensellik testi* sonucuna göre *S&P-500* endeksinin İMKB-100 endeksini etkilediği görülmüştür.

4.2.4 Makine Öğrenmesi Yöntemleri

İstatistiksel yöntemler gözlemler üzerinde çok ciddi varsayımlarda bulunurlar. Bu yöntemlere göre daha az varsayım içeren ve daha esnek olan YSA ve DVM gibi makine öğrenmesi yöntemlerinin sermaye piyasalarının tahmininde kullanıldığı çalışmalar gün geçtikçe artmaktadır.

Bu bölümde, makine öğrenmesi yöntemlerinden olan YSA ve DVM yöntemlerinin hisse senedi piyasalarındaki uygulamaları ile ilgili literatür taraması sunulmaktadır.

4.2.4.1 Yapay Sinir Ağları

Türkiye'deki YSA çalışmaları genelde finansal başarısızlık ve iflasların tahmini için kullanılmıştır. Yurtdışında borsa endeksi tahmini konusunda çalışmalar olduğu halde Türkiye'de bu tip çalışmaların eksikliği görülmektedir.

(Çinko & Avcı, 2007) yaptıkları çalışmada İMKB-100 endeksinin tahmini için YSA ve doğrusal regresyon kullanmışlar ve sonuçları karşılaştırmışlardır. Modelin değişkenleri olarak *endeksin ve işlem hacminin gecikmeli değerleri ile hareketli ortalamalarını* kullanmışlardır. Çalışmada YSA' nın regresyon modellerine göre genelde daha iyi sonuçlar verdiği görülmüştür.

(Akcan & Kartal, 2011) İMKB sigorta sektörü endeksini oluşturan yedi adet şirketin hisse senedi fiyatlarını YSA ile tahmin etmeye çalışmışlardır. Modelde *İMKB-100 endeksi günlük kapanış değeri, tüketici fiyat endeksi, dolar efektif satış kuru, günlük cumhuriyet altını fiyatı* makroekonomik değişkenleri kullanılmıştır.

(Kutlu & Badur, 2009) yaptıkları çalışmada İMKB endeks yönünün tahmini için YSA kullanmışlardır. Çalışmada kullanılan değişkenler şunlardır: *Önceki günün endeks değeri, önceki günün ABD Doları değeri, önceki günün gecelik faiz değeri, haftanın günlerini belirten 5 kukla değişken, önceki günün Fransa, Almanya, İngiltere, Brezilya, Japonya, NASDAQ, DOW JONES ve S&P-500 borsa endeks değerleri.*

(Yıldız vd., 2008) tarafından yapılan çalışmada İMKB-100 endeksinin yön tahmini için YSA kullanılmıştır. Analiz için *İMKB en yüksek, en düşük, kapanış değerleri ile ABD doları kuru* kullanılmıştır.

(Boyacıoğlu & Avcı, 2010), İMKB endeksini doğru bir şekilde tahmin edebilmek için ANFIS (Adaptive Neuro Fuzzy Inference System) yöntemini kullanmışlardır. Bağımsız değişken olarak 6 makroekonomik (*cumhuriyet altını satış fiyatı, ABD Dolar kuru, faiz oranları, tüketici fiyat endeksi, sanayi üretim endeksi, hazine faiz oranları*) ve 3 endeks (*DOW JONES, DAX ve BOVESPA*) değeri kullanılmıştır.

4.2.4.2 Destek Vektör Makineleri

DVM yönteminin İMKB verilerine uygulandığı uygulamalar çok sınırlıdır. Aşağıda dünyada ve Türkiye’de DVM kullanılarak yapılan borsa yön tahmini uygulamaları yer almaktadır.

(Kim, 2003) tarafından yapılan çalışmada Kore Borsası (KOSPI) endeks yönünün günlük verilerle tahmini için DVM kullanılmıştır. DVM performansı YSA ile karşılaştırılmıştır. Analizde 12 adet teknik gösterge kullanılmıştır. Bu göstergeler *%K, %D, Slow %D, Momentum, ROC, William’s %R, A/D Oscillator, Disparity 5 günlük, Disparity 10 günlük, OSCP, CCI, RSI* teknik göstergeleridir. 2347 gözlem eğitim için (%80), 581 gözlem test için kullanılmıştır (%20). Analiz sonucuna göre DVM (%57,83), YSA (%54,73) başarı göstermiştir.

(Huang vd., 2005), NIKKEI-225 endeksinin haftalık deęişim yönünü tahmin etmek için DVM yöntemini kullanmışlardır. NIKKEI borsasının deęişim yönünü tahmin etmek için Japonya'nın en fazla ihracat yaptığı ülke olan ABD'nin genel ekonomik durumunu gösteren *S&P-500 endeksi ve Japon Yeni Amerikan Doları kuru* modelde girdi deęişkenler olarak kullanılmıştır.

(Özdemir vd., 2011) tarafından yapılan çalışmada İMKB-100 endeksinin yönünü tahmin etmek için Şubat 1997'den Aralık 2010'a kadar olan dönemi kapsayan aylık veriler üzerinde DVM yöntemi uygulanmıştır. Analizde, 8'i makroekonomik gösterge, 3'ü dięer borsaların endeks deęerleri olmak üzere toplam 11 deęişken kullanılmıştır.

(Kara vd., 2011) çalışmalarında 02.01.1997-31.12.2007 dönemine ait verilerle İMKB 100 endeksinin yön tahminini yapmak amacıyla DVM yöntemini kullanmışlardır.

4.3 İMKB Ulusal 100 Endeksinin Yön Tahmini

Bu bölümde, İMKB ve endeksler hakkında temel bilgiler verildikten sonra literatür taraması sonucu İMKB'yi etkilediđi saptanan deęişkenlerden günlük olarak erişilebilenlerden veri kümeleri oluşturularak bu veri kümeleri üzerinde YSA ve DVM yöntemlerinin performansları karşılaştırılacaktır.

4.3.1 İMKB Hakkında Temel Bilgiler

İMKB'deki şirketlerin piyasa deęeri 2011 yılında bir önceki yıla göre %34 gerilemiş, 201 milyar \$'a düşmüştür. Dünya borsaları arasında İMKB'nin konumuna bakıldığında, piyasa deęerine göre 2009 yılında 25. sıraya kadar çıkan İMKB'nin 2010 yılında 30, 2011 yılında ise 32. sıraya gerilediđi görülmektedir (Tablo 11).

Tablo 11. Borsaların Piyasa Değeri (2011)

Sıra	Borsa	Ülke	Piyasa Değeri (Milyar \$)	Piyasa Değeri Payı (%)	Piyasa Değeri / GSYH (%)
1	New York Borsası	ABD	11.796	%24,9	%78
2	NASDAQ	ABD	3.845	%8,1	%26
3	Tokyo Borsası	Japonya	3.325	%7	%57
4	Londra Borsası	İngiltere, İtalya	3.266	%6,9	%69
5	EuroNext	Hollanda, Belçika Fransa, Portekiz	2.447	%5,2	%55
6	Şangay Borsası	Çin	2.357	%5,0	%34
7	Hong Kong Borsası	Hong Kong	2.258	%4,8	%914
8	TMX Grubu	Kanada	1.912	%4,0	%109
9	BM & BOVESPA	Brezilya	1.229	%2,6	%49
10	Avustralya Borsası	Avustralya	1.198	%2,5	%79
32	İMKB	Türkiye	201	%0,4	%26

Kaynak: TSPAKB “Türkiye Sermaye Piyasası 2011” Raporu

Borsaların piyasa değerinin, bu borsaların yer aldığı ülkelerdeki milli gelire oranı hisse senedi piyasasının ülke ekonomisindeki yerini göstermektedir. İMKB'nin piyasa değeri, Türkiye milli gelirinin dörtte birine denk gelmektedir.

2011 sonu itibariyle dünya borsalarında kote şirket sayısı 46.204' tür. Toplam kote şirket sayısında İMKB, 263 adet şirket ile 34. sırada yer almaktadır (Tablo 12). Mevcut durumda sadece 1 yabancı şirket İMKB'ye kotedir. Kote şirket sayısı içine Ulusal Pazardaki şirketler ile gayrimenkul ve girişim sermayesi yatırım ortaklıkları alınmıştır.

Tablo 12. Borsaya Kote Olan Şirket Sayısı (2011)

Sıra	Borsa	Şirket Sayısı	Toplamdaki Payı
1	Bombay Borsası	5.112	% 11,1
2	TMX Grubu	3.945	% 8,5
3	BME İspanya Borsası	3.276	% 7,1
4	Londra Borsası Grubu	2.886	% 6,2
5	NASDAQ	2.680	% 5,8
6	New York Borsası	2.308	% 5,0
7	Tokyo Borsası	2.291	% 5,0
8	Avustralya Borsası	2.079	% 4,5
9	Güney Kore Borsası	1.816	% 3,9
10	Ulusal Hindistan Borsası	1.640	% 3,5
34	İMKB	263	% 0,6

Kaynak: TSPAKB “Türkiye Sermaye Piyasası 2011” Raporu

2011 yılı sonu itibariyle hisse senedi hacmi bakımından İMKB, 20. sırada yer almaktadır. Hisse senedi işlem hacminin toplam piyasa değerine oranını, piyasanın likiditesine ilişkin bir gösterge olarak kullanmak mümkündür. Hisse senedi devir hızını ifade eden bu oranın yüksek olması likit bir piyasaya işaret ederken, aynı zamanda bu borsadaki yatırımcıların portföylerini daha kısa vadeli olarak değerlendirdiklerini göstermektedir. %210 devir hızıyla İMKB 3. sırada yer almaktadır (Tablo 13).

Tablo 13. Hisse Senedi Hacmi (2011)

Sıra	Borsa	Hisse Senedi Hacmi (Milyar \$)	Hisse Senedi Hacmi Payı	Hacim / Piyasa Değeri
1	New York Borsası	18.027	%28,6	%153
2	NASDAQ	12.724	%20,2	%331
3	Tokyo Borsası	3.972	%6,3	%119
4	Şangay Borsası	3.658	%5,8	%155
5	Şenzhen Borsası	2.838	%4,5	%269
6	Londra Borsası Grubu	2.837	%4,5	%87
7	EuroNext	2.134	%3,4	%87
8	Güney Kore Borsası	2.029	%3,2	%204
9	Deutsche Börse	1.758	%2,8	%148
10	TMX Grubu	1.542	%2,4	%81
20	İMKB	424	%0,7	%210

Kaynak: TSPAKB “Türkiye Sermaye Piyasası 2011” Raporu

İMKB 2011 yılında gelişmekte olan ülke borsaları arasında hisse senetleri işlem hacmi açısından 7. sıradadır. 2011 yılı sonu itibarıyla yabancı yatırımcıların sahip olduğu hisse senetlerinin halka açık piyasa değeri içerisindeki payı yüzde 62,1 olan İMKB, piyasa değeri açısından gelişmekte olan ülke borsaları arasında 15. sırada yer almaktadır.

2011 yılı sonunda bir önceki yıla göre ortalama günlük işlem hacmi yüzde 8 artarak 2,75 milyar TL’ye (1,7 milyar \$), 326 bin olan ortalama günlük sözleşme sayısı yüzde 24 artarak 405 bine yükselmiştir. Ortalama günlük işlem miktarı ise, 2011 yılında 798,4 milyon adet seviyesinde gerçekleşmiştir. İMKB-100 endeksi 2011 yılı sonunda 51.266,62 puan (ABD doları bazında 1.580,45 puan) olmuştur (Kaynak: İMKB 2011 Faaliyet Raporu).

4.3.2 İMKB Endeksleri ve Hesaplanmaları

Bir menkul kıymetler borsasının davranışını, hisse senetlerinin fiyat ve işlem hacmi performansını en iyi yansıtan araç borsa endeksleridir. Bu nedenle, menkul kıymetler borsasının yapısı, hisse senetlerinin davranışları ve borsa endeksleri hakkında yapılan çalışmalara önem verilmektedir. Bir ülkenin menkul kıymetler borsası hem genel hem de sektörel ekonomiyi ilgilendiren iç ve dış faktörlerden etkilenir. Borsa endeksleri yatırımcıların değişik ekonomik olaylara verdiği tepkiler sonucunda hisse senedi fiyatları ve işlem hacimlerinde meydana gelen değişimler yoluyla artış ya da azalış gösterirler (Yüksel & Gülerüz, 2010: 98).

Piyasa endekslerinin belli başlı kullanım alanları şu şekilde sıralanabilir (Dağlı, 2000: 191):

1. Endeksler, hisse senedi fiyatlarında zaman içinde meydana gelen toplu değişimi ortaya koyarlar.
2. Kurumsal ve bireysel yatırımcılara, sahip oldukları portföylerin belli bir dönem boyunca gösterdiği performansı, piyasanın genel performansıyla karşılaştırma imkânı sağlarlar.
3. Endeksler, hisse senedi piyasasındaki gelişmeler ile genel ekonomik göstergelerdeki değişimler arasında karşılaştırma yapma imkânı sağlarlar.
4. Sermaye piyasası analizlerinde endekslerden yararlanılır.
5. Endekslerden yararlanarak, endekse dayalı yeni yatırım araçları geliştirilebilir.
6. Alternatif yatırım araçlarının getirilerinin karşılaştırılmasında endeksler yatırımcılara faydalı bilgiler sağlar

İMKB de hesaplanan endeksler aşağıda sıralanmıştır (Kaynak: Borsa İstanbul Eğitim Klavuzları).

- **İMKB-100 Endeksi:** İMKB Hisse Senetleri Piyasası için temel endeks olarak kullanılmaktadır. 1986 yılında 40 şirketin hisse senedi ile başlayan ve zamanla sayısı 100 şirketin hisse senedi ile sınırlanan Bileşik Endeks' in devamı niteliğindedir. Ulusal Pazar'da işlem gören şirketlerle, Kurumsal Ürünler Pazarı'nda işlem gören gayrimenkul yatırım ortaklıkları ve girişim sermayesi yatırım ortaklıkları arasından seçilen 100 hisse senedinden oluşmakta olup, İMKB-30 ve İMKB-50 endekslerine dahil hisse senetlerini de kapsar.
- **İMKB-50 Endeksi:** Ulusal Pazar'da işlem gören şirketlerle, Kurumsal Ürünler Pazarı'nda işlem gören gayrimenkul yatırım ortaklıkları ve girişim sermayesi yatırım ortaklıkları arasından seçilen 50 hisse senedinden oluşmakta olup, İMKB-30 endeksine dahil hisse senetlerini de kapsar.
- **İMKB-30 Endeksi:** Ulusal Pazar'da işlem gören şirketlerle, Kurumsal Ürünler Pazarı'nda işlem gören gayrimenkul yatırım ortaklıkları ve girişim sermayesi yatırım ortaklıkları arasından seçilen 30 hisseden oluşur.
- **İMKB-10 Banka Endeksi:** Ulusal Pazar'da işlem gören bankalar arasından seçilen 10 hisse senedinden oluşur.
- **İMKB 100 - 30 Endeksi:** İMKB 100 Endeksine dahil olup, İMKB 30 endeksinde yer almayan 70 hisse senedinden oluşur.
- **İMKB Kurumsal Yönetim Endeksi:** İMKB pazarlarında işlem gören ve belirlenmiş asgari kurumsal yönetim derecelendirme notuna sahip şirketlerin hisse senetlerinden oluşur.

- **İMKB Tüm Endeksi:** Menkul Kıymet Yatırım Ortaklıkları hariç olmak üzere, İMKB pazarlarında işlem gören şirketlerin hisse senetlerinden oluşur.
- **İMKB Tüm - 100 Endeksi:** İMKB Tüm endeksine dahil olup, İMKB-100 endeksinde yer almayan hisse senetlerinden oluşur.
- **Sektör Endeksleri ve Alt Sektör Endeksleri:** Menkul Kıymet Yatırım Ortaklıkları hariç olmak üzere, İMKB pazarlarında işlem gören şirketlerin hisse senetlerinden oluşur.
- **İMKB Ulusal Endeksi:** Ulusal Pazar'da işlem gören şirketlerin hisse senetlerinden oluşur.
- **İMKB İkinci Ulusal Endeksi:** İkinci Ulusal Pazar'da işlem gören şirketlerin hisse senetlerinden oluşur.
- **İMKB Menkul Kıymet Yatırım Ortaklıkları Endeksi:** İMKB pazarlarında işlem gören menkul kıymet yatırım ortaklıklarının hisse senetlerinden oluşur.
- **İMKB Şehir Endeksleri:** İMKB pazarlarında işlem gören ve ana üretim/hizmet faaliyetlerinin gerçekleştiği ya da şirket merkezinin bulunduğu şehre göre gruplandırılmış hisse senetlerinden oluşur. Holdingler hariç mali sektörde faaliyet gösteren şirketler ile perakende ticaret sektöründe faaliyet gösteren şirketler kapsam dışındadır.

Endekslerin hesaplanmasında tescil edilmiş en son fiyatlar kullanılır. Endeksler, kapsamlarında bulunan hisse senetlerinin fiili dolaşımdaki

paylarının piyasa deęerleri ile aęırlıklı olarak hesaplanır. Endekslerin hesaplanma yöntemi ařaęıdaki gibidir.

$$E_t = \frac{\sum_{i=1}^n F_{it} * N_{it} * H_{it}}{B_{it}}$$

E_t : Endeksin t zamandaki deęeri

n : Endekse dahil olan hisse (řirket) sayısı

F_{it} : i . hisse senedinin t zamandaki fiyatı

N_{it} : i . hisse senedinin t zamandaki toplam sayısı

H_{it} : i . hisse senedinin t zamandaki fiili dolařım pay oranı (hisse senedinin halka aęıklık oranı)

B_{it} : Endeksin t zamandaki bölen deęeri

İMKB hisse senedi endekslerinin seans sonu kapanıř deęerleri, ABD Doları ve Avro bazında da hesaplanarak ilan edilir. Endekslerin yabancı para birimi cinsinden hesaplanmasında ařaęıdaki formül kullanılır.

$$EY_t = \frac{(E_t/K_t)}{(E_b/K_b)} * EY_b$$

EY_t : Endeksin yabancı para cinsinden t zamandaki deęeri

E_t : t günü endeks kapanıř deęeri

K_t : t günü döviz kuru

E_b : Endeksin bařlangıç günündeki kapanıř deęeri

K_b : Endeksin bařlangıç günündeki döviz kuru

EY_b : Endeksin yabancı para birimi cinsinden bařlangıç deęeri

4.3.3 Analizde Kullanılan Teknik Göstergeler

İMKB-100 endeksinin yön tahmini için kullanılacak olan 1.veri kümesi MASS, ATR, MO (Momentum), CCI, MACD, TRIX, MOV (Hareketli Ortalama), MFI, RSI, STOCH (Stokastik), WILLR (William's %R) değişken isimleriyle gösterilen 11 adet teknik göstergeden oluşmaktadır. (Tablo 14) de bu teknik göstergeler ile ilgili temel istatistiksel bilgileri bulunmaktadır.

Tablo 14. Teknik Göstergeler İle İlgili Temel İstatistikler

Teknik Gösterge	Gözlem Sayısı	Ortalama	Minimum	Maksimum	Standart Sapma
MASS	1747	25,01	22,19	29,04	1,14
ATR	1747	1055,30	421,15	2232,72	327,58
MO	1747	100,75	70,77	119,70	6,53
CCI	1747	15,51	-309,54	333,47	110,60
MACD	1747	114,91	-2893,93	2014,85	890,32
TRIX	1747	0,05	-1,43	0,91	0,35
MOV	1747	44499,06	23800,36	69595,72	12583,81
MFI	1747	56,54	6,53	99,61	16,30
RSI	1747	53,45	13,21	88,37	12,93
STOCH	1747	56,83	3,04	99,72	26,77
WILLR	1747	-41,45	-100,00	0,00	30,51

Eksik teknik gösterge gözlemi bulunmamaktadır. Tüm teknik göstergeler incelenen dönemde İMKB'nin açık olduğu gün sayısı (1747) kadardır. Standart sapması en büyük olan teknik göstergeler sırasıyla MOV, MACD, ATR ve CCI olarak belirlenmiştir. Analizde kullanılan teknik göstergelerin hesaplanmaları kısaca özetlenmiştir (Perşembe, 2010; Erdinç, 2004).

4.3.3.1 Hareketli Ortalamalar

Basit Hareketli Ortalama

$$BHO_t = F_t + F_{t-1} + F_{t-2} + \dots + F_{t-n} / n$$

şeklinde hesaplanır.

BHO_t : Şimdiki basit hareketli ortalama

F : Fiyat

t : Şimdiki tarih

n : Ortalama alınan süre

Üssel hareketli ortalama ise son günleri ön plana çıkarmanın yanı sıra, basit hareketli ortalamanın aksine mevcut tüm veri noktalarını hesaba katar. Hesaplanması şu şekilde yapılır.

$$\ddot{U}HO_t = \ddot{U}HO_{t-1} + (DF * (F_t - \ddot{U}HO_{t-1}))$$

$\ddot{U}HO_t$: Üssel hareketli ortalama

DF : Düzeltme faktörü($2/(n + 1)$)

Hareketli ortalamaların temel olarak üç ana işlevi bulunur:

1. Piyasanın ne yöne doğru trend yaptığını belirlemek.
2. Oynaklığı daha düz bir çizgiye indirgeyip fiyat zikzaklarının çıkardığı gürültüyü azaltarak, ana trendleri ortaya çıkarmak (düzeltme işlevi).
3. Fiyat ve tepe diplerini belirlemek.

Hareketli ortalamalar trend takip edici göstergelerdir. Piyasanın ardından gecikmeli (ardışık) olarak gelirler. Dolayısıyla yaptıkları iş, fiyatların nereye gideceğini göstermek değil fiyatlara tepki göstermektir. Hareketli ortalamaların bu gecikmeli olma özelliği “düzeltme” işlevini yerine getirmelerini sağlar.

4.3.3.2 MACD (Moving Average Convergence Divergence – Hareketli Ortalamalar Yakınlaşma - Uzaklaşma)

İki hareketli ortalama arasındaki uyum veya uyumsuzluğun tespit edilmesinde kullanılır. MACD indikatörü hesaplanış bakımından basittir. Öncelikle fiyatların bir uzun vadeli (26 günlük) bir de kısa vadeli (12 günlük) üssel hareketli ortalaması bulunur. Bunlar bulunduktan sonra kısa vadeli hareketli ortalama uzun vadeli hareketli ortalama çıkarılır. Daha sonra, dönüş hareketleri ve uyumsuzluğun

tespitinde yararlanılan sinyal eğrisi, MACD değerinin 9 günlük üssel hareketli ortalaması alınarak hesaplanır.

$$MACD = \ddot{U}HO(12) - \ddot{U}HO(26)$$
$$Sinyal = \ddot{U}HO(MACD, 9)$$

4.3.3.3 CCI (Commodity Channel Index - Mal Kanal Endeksi)

CCI trend takip edici bir göstergedir ve *EY*: En yüksek, *ED*: En düşük değeri göstermek üzere şöyle hesaplanır:

1. *Günlük Ortalama Fiyat (GOF) = (EY + ED + Kapanış)/3* bulunur.
2. GOF'un 20 periyotluk Basit Hareketli Ortalaması (BHO) alınır.
3. *Ortalama Sapma (OS) = GOF - BHO* biçiminde hesaplanır.
4. Son 20 periyodun (BHO' nun süresi 20 olduğu için) OS'larının toplamı (TOS) alınır.
5. $CCI = (TOS * 0,015)/(BHO - GOF)$ şeklinde hesaplanır.

4.3.3.4 RSI (Relative Strength Index - Göreceli Güç Endeksi)

Aşırı alım / aşırı satım göstergeleri arasında en yaygın olarak kullanılan göstergelerden birisidir. Çok çeşitli şekillerde kullanılabilen bu basit ama basit olduğu kadar güvenilir gösterge 1978 yılında geliştirilmiştir.

$$RSI = 100 - (100 / (1 + RS))$$

$$RS = \frac{\text{Son 14 periyodun YUKARI kapanışlarının ortalaması}}{\text{Son 14 periyodun AŞAĞIYA kapanışlarının ortalaması}}$$

4.3.3.5 Stokastik

Stokastik bir aşırı alım / aşırı satış göstergesidir. Bir yukarı trendde kapanışların günlük işlem aralığının üst taraflarında, bir aşağı trendde de alt taraflarında yoğunlaşacağı varsayımı üzerine inşa edilmiştir. Stokastik formülü %K denilen değeri hesaplar.

$$\text{Ham } \%K_t = ((\text{Kapanıř}_t - ED_n) / (EY_n - ED_n)) * 100$$

Ham $\%K_t$: En son periyodun *Ham* $\%K$ 'sı

Kapanıř $_t$: En son periyodun kapanıřı

EY $_n$ ve *ED* $_n$: Son *n* periyodun en yksek ve en dřđ

4.3.3.6 William's %R

$\%R$, RSI ve Stokastik gibi gstergelerle aynı ailedendir (bir ařırı alım / ařırı satım gstergesi). Aslında Stokastik gstergesinin bir varyasyonudur. $\%R$, iřlem aralıđı (seilen periyodun, rneđin bir gn, en ykseđi ile en dřđ arasındaki mesafedir) demek olan *Range* kelimesinin kısaltılmıřıdır. Aynı Stokastik gstergenin arkasında yatan varsayımda olduđu gibi, kapanıř fiyatının seilen periyodun iřlem aralıđının en ykseklerine yakın olması, piyasanın alıřta olduđunun ve řiřmeye bařladıđının resmidir. Eđer kapanıř fiyatı seilen iřlem aralıđının alt taraflarındaysa, piyasanın ařırı satılmıř olduđu varsayılır. Bu varsayımlar zerine inřa edilen $\%R$ forml řyledir:

$$\%R_t = ((EY_n - \text{Kapanıř}_t) / (EY_n - ED_n)) * 100$$

$\%R_t$: En son $\%R$

Kapanıř $_n$: En son kapanıř

n: Seilen sre

4.3.3.7 Momentum

Momentum gstergesinin en nemli zelliđi hem trend yapan hem de yatay piyasalarda kullanılıyor olmasıdır. Momentum ok basit bir formlle hesaplanır:

$$\text{Momentum} = F_s - F_{s-n}$$

F_s : En son fiyat

F_{s-n} : *n* periyot nceki fiyat

Momentum, en son fiyatı (örneğin bu günkü kapanışı) n periyot önceki fiyatla (örneğin 10 gün önceki kapanışla) karşılaştırır. Eğer, bu en son fiyat değeri, n periyot önceki fiyattan yukarıysa pozitif, aşağıysa negatif bir değer çıkar. Başka bir şekilde ifade edecek olursak, momentum çizgisi sıfırın etrafında dalgalanan, bazen pozitif bazen de negatif değer taşıyan bir çizgidir.

4.3.3.8 MFI (Money Flow Index – Para Akışı Endeksi)

MFI bir işlem hacmi göstergesidir. Günün ortalama fiyatı baz alınarak hesaplanır.

$$\text{Ortalama Fiyat} = (EY + ED + Kapanış) / 3$$

Bu değer işlem hacmiyle çarpılarak Para Akışı bulunur.

$$\text{Para Akışı} = \text{Ortalama Fiyat} * \text{İşlem Hacmi}$$

Eğer en son periyodun ortalama fiyatı bir önceki periyodun ortalama fiyatından yukarıdaysa para girişi, altındaysa para çıkışı olmuştur. Para Girişi, seçilen süredeki (örneğin son 14 gün) para girişlerinin toplamı, Para Çıkışı para çıkışlarının toplamıdır. Bu iki değer kullanılarak Para Oranı bulunur.

$$\text{Para Oranı} = \text{Para Girişi} / \text{Para Çıkışı}$$

Bu durumda Para Akış Endeksi

$$\text{Para Akış Endeksi} = 100 - (100 / (1 + \text{Para Oranı}))$$

biçiminde hesaplanır.

4.3.3.9 ATR (Average True Range - Ortalama Doğru Aralık)

Fiyatlardaki dalgalanmanın boyutlarını ölçen bir göstergedir. Değerinin artıyor olması fiyatlardaki değişkenliğin (hareketliliğin) arttığını, aksi durum ise fiyatlardaki dalgalanmanın düştüğünü gösterir. Dolayısıyla fiyatlar aniden düşmeye

başladığında veya aniden yükselmeye başladığında bu gösterge artar. Eğer fiyatlar belirli bir değer etrafında dalgalanıyorsa düşer. Hesaplanışı şu şekildedir:

$$TR_t = Maks(|EY_t - ED_t|, |EY_t - Kapanış_{t-1}|, |ED_t - Kapanış_{t-1}|)$$

$$ATR_0 = TR_0$$

$$ATR_t = \frac{(n - 1)ATR_{t-1} + TR_t}{n}$$

TR_t : t anındaki Doğru Aralık (True Range)

4.3.3.10 TRIX

TRIX, bir momentum göstergesidir ve fiyatlardaki küçük iniş çıkışları elimine etmek için, belirlenmiş süreler için kapanış fiyatlarının üç kere arka arkaya üssel hareketli ortalaması alınarak hesaplanır. Bu sayede grafiği çizildiğinde fiyatların trendi çok daha net görülebilir ve yorumlanabilir. TRIX göstergesi

$$TRIX = \frac{\ddot{ÜHO3}_t - \ddot{ÜHO3}_{t-1}}{\ddot{ÜHO3}_{t-1}}$$

formülüyle hesaplanır.

4.3.3.11 MASS Endeksi

Trend değişimini haber vermeyi amaçlayan bir göstergedir. Bu endeks hesaplanırken en düşük ve en yüksek fiyatlar arasındaki mesafenin artıp azaldığına bakılır. Eğer en yüksek ve en düşük fiyatlar arasındaki mesafe genişlerse MASS endeksi artar, daralırsa azalır. Hesaplanışı şu şekildedir:

$$MASS\ Endeksi = \sum_1^{27} \left[\frac{9\text{ günlük } \ddot{ÜHO}(EY - ED)}{9\text{ Günlük } \ddot{ÜHO}(9\text{ günlük } \ddot{ÜHO}(EY - ED))} \right]$$

4.3.4 Analizde Kullanılan Borsa Endeks Verileri

Endeks yönünün tahmin edilmesinde kullanılacak 2. veri kümesinde borsa endeks değerleri yer almaktadır. Analizde endeks verileri kullanılan borsalar (Tablo 15) de listelenmiştir.

Tablo 15. Analizde Kullanılan Endeksler

Kısaltma	Endeks	Ülke
İMKB-100	İMKB-100 Endeksi	Türkiye
BOVESPA	São Paulo Stock Exchange	Brezilya
DAX	Deutscher Aktien Index	Almanya
DJIA	Dow Jones Industrial Average	ABD
FTSE-100	Financial Times Stock Exchange 100 Index	İngiltere
NIKKEI-225	Nihon Keizai Shimbun Index 225 Index	Japonya
NYSE	New York Stock Exchange	ABD
SP-500	Standart & Poors 500 Index	ABD
HSI	Hang Seng Index	Hong Kong

İMKB-100 endeksinin yön tahmini için 9 borsanın 1 ve 2 gün gecikmeli olmak üzere toplam 18 adet yüzde değişim oranı kullanılmıştır. Microsoft Excel yazılımı kullanılarak borsaların açık oldukları bir önceki güne göre % değişim oranları $((Kapaniş_t - Kapaniş_{t-1}) / Kapaniş_{t-1}) * 100$ formülü kullanılarak bulunmuştur. Analizde İMKB-100 endeksinin yönü tahmin edilmeye çalışıldığı için İMKB'nin açık olduğu günler temel alınmıştır. İMKB'nin açık olduğu bir günde bağımsız değişken olarak kullanılan herhangi bir borsa kapalıysa bu borsanın değişim oranının bir önceki açık olan günün değişim oranıyla aynı olduğu kabul edilmiştir.

Analizde kullanılan borsaların yüzdelerik deęişimleri ile ilgili bilgiler (Tablo 16) da gösterilmiştir. Yüzdelerik deęişim aralığının en büyük olduęu endeksler NIKKEI-225 ve BOVESPA endeksleridir. Yüzdelerik deęişimlerin standart sapması en büyük olan borsalar gelişmekte olan borsalar kategorisinde yer alan İMKB ve BOVESPA borsalarıdır.

Tablo 16. Modelde Yer Alan Borsalar İle İlgili Temel İstatistiksel Bilgiler

Borsa	Gözlem Sayısı	Ortalama	Minimum	Maksimum	Std. Sapma
İMKB-100	1747	0,069415	-8,6194	12,89317	1,895772
BOVESPA	1747	0,057807	-11,3931	14,65600	1,891846
DAX	1747	0,018190	-7,1639	11,40195	1,472825
DJIA	1747	0,006554	-7,8733	11,08033	1,290940
FTSE-100	1747	0,013546	-8,8483	9,83867	1,339373
NIKKEI-225	1747	-0,003889	-11,4064	14,15030	1,650103
NYSE	1747	0,006233	-9,7260	12,21624	1,497724
SP-500	1747	0,003273	-9,0350	11,58004	1,412971
HSI	1747	0,021503	-8,6538	12,81508	1,718419

4.3.5 Analizde Kullanılan Makroekonomik Göstergeler

Literatürde sermaye piyasalarının tahmininde çok deęişik makroekonomik deęişkenlerin kullanıldığı görülmektedir. Bu deęişkenlerin birçoęu haftalık, aylık veya üç aylık periyotlarla yayınlanan veriler olduğundan günlük yön tahminin yapıldığı analizde yer almamışlardır.

3.veri kümesinde makroekonomik değişkenler yer almaktadır. Bu değişkenler şunlardır: Avro, Sterlin, Dolar, Yen kurlarının 1 günlük gecikmeli değerleri ve dolar cinsinden altının ons başına 1 günlük gecikmeli değeri.

4.3.6 Verilerin Toplanması ve Hazırlanması

İMKB kapanış verileri “<http://www.imkb.gov.tr>” adresinden alınmıştır. Diğer borsaların kapanış verileri ise “<http://finance.yahoo.com>” sayfasından temin edilmiştir. Makroekonomik veriler için (“<http://evds.tcmb.gov.tr>”) adresindeki Merkez Bankası Elektronik Veri Dağıtım Sistemi’nden faydalanılmıştır. Teknik analiz göstergelerini elde edebilmek için MetaStock yazılımı kullanılmıştır.

Günlük değişim yönü “-1” ve “1” olarak etiketlenmiştir. Eğer İMKB-100 endeksi t gününde $t - 1$ gününden daha yüksek değerde kapanmış ise “1” olarak; daha düşük değerde kapanmış ise “-1” olarak gösterilmiştir.

Analizde kullanılan veriler 03.01.2005-30.12.2011 döneme aittir. Toplam gözlem sayısı 1747 işlem gününden oluşmuştur. Bu dönemde İMKB-100 endeksinin yükseldiği gün sayısı 921 iken endeksin düşme yönünde olduğu 826 gün vardır. Bu durumda artan yöndeki günler tüm işlem günlerinin %52,8’ ini, geri kalan %47,2’lik kısım ise düşme yöndeki günleri oluşturur (Tablo 17). Bu değerler birbirine yakın olduklarından artış ve düşüş sınıflarının dengeli olduğu söylenebilir.

Tablo 17. Yıllara Göre İMKB 100 Endeksinin Arttığı ve Azaldığı Gün Sayıları

	Yıllar							Toplam
	2005	2006	2007	2008	2009	2010	2011	
Yükselme	145	131	126	104	147	140	128	921
%	57,8	52,4	50,6	41,8	58,8	56,7	51	52,8
Düşme	106	119	123	145	103	107	123	826
%	42,2	47,6	49,4	58,2	41,2	43,3	49	47,2
Toplam	251	250	249	249	250	247	251	1747

Yıllara göre borsanın açık olduğu gün sayıları birbirine yakındır. İncelenen dönemde işlem yapılan gün sayısı 247 ile 251 arasında değişmektedir. Yıllara göre artış ve düşüş yüzdelerine bakıldığında bu oranların da birbirine yakın olduğu görülmektedir. Yıl içinde artışın olduğu günlerin yüzdesi, %41,8 (2008 yılı) ile %58,8 (2009 yılı) arasında değişmektedir. Endeksin arttığı günlerin yüzdesi düştüğü günlerin yüzdesinden 2008 yılı haricinde daha yüksektir.

Analizde 2005-2009 dönemi (5 yıl) eğitim kümesi, 2010-2011 dönemi (2 yıl) test kümesi olarak kullanılacaktır. Bu dönemlere ilişkin artış ve düşüş bilgileri (Tablo 18) de belirtilmiştir. Buna göre eğitim kümesi 1249 gözlemden, test kümesi ise 498 gözlemden oluşmaktadır. Artışın olduğu günlerin yüzdeleri incelendiğinde eğitim ve test kümelerindeki oranların (sırasıyla %52,2 ve %53,9) birbirine yakın oldukları görülmektedir. Buna göre eğitim ve test kümelerindeki sınıf dağılımlarının birbirine benzer olduğu söylenebilir.

Tablo 18. Eğitim ve Test Kümelerinin Artış ve Azalış Yüzdeleri

	Dönem	
	2005-2009 (Eğitim)	2010-2011 (Test)
Yükselme	653	268
%	52,2	53,9
Düşme	596	230
%	47,8	46,1
Toplam	1249	498

4.3.7 Analizde Kullanılan Yazılım ve Özellikleri

YSA ve DVM yöntemlerinin elde edilen veriler üzerinde çalıştırılması için STATISTICA 10 ("<http://www.statsoft.com>") tercih edilmiştir. Bu yazılım YSA uygulamalarında çok fazla zaman gerektiren farklı ağ parametrelerinin denenmesi

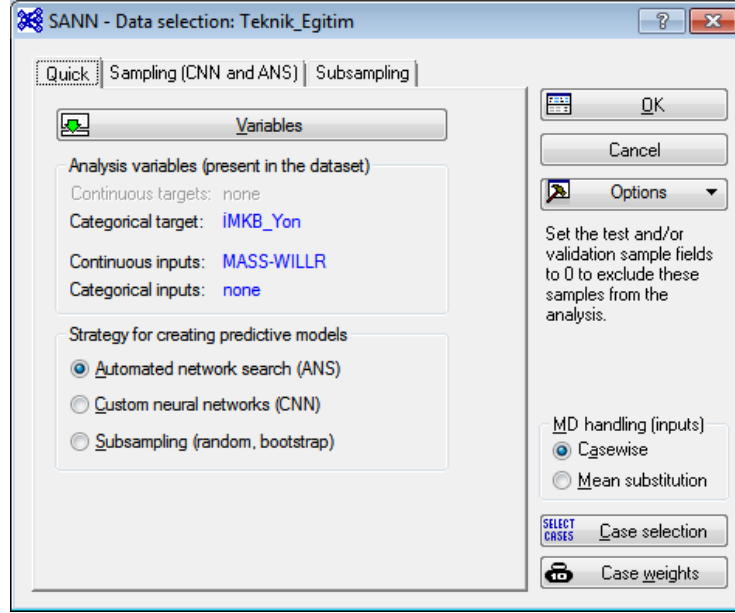
işlemini otomatik hale getiren “*SANN (STATISTICA Automated Neural Networks)*” adında bir modüle sahiptir. Bu modülün özellikleri şu şekilde sıralanabilir.

1. Sınıflandırma, regresyon ve zaman serileri problemleri için veri seçimi, nominal değer kodlama, ölçekleme, normalizasyon ve eksik veri yerine koymayı içeren ön işleme (pre-processing) işlemleri bütünleşik olarak yapılabilmektedir.
2. Çok fazla zaman alan deneme yanılma yoluyla ağ parametrelerinin değiştirilmesi yerine belirtilen parametreleri veya belirli bir aralıktaki parametreleri otomatik olarak deneyerek en uygun ağın bulunmasını sağlar.
3. Eğimli İniş (Gradient Descent) haricinde ikinci dereceden türeveye dayanan Conjugate Gradient Descent (Eşlenik Eğimli İniş) ve BFGS (Broyden – Fletcher – Goldfarb – Shanno) gibi farklı optimizasyon algoritmaları kullanılabilir.
4. Birleştirilmiş ağların (ensemble networks) kullanımını destekler.
5. Kapsamlı grafiksel ve istatistiksel sonuçlar oluşturabilir.

STATISTICA 10 yazılımının destek vektör makineleri modülünde sınıflandırma için *C-SVM (Type 1)* ve *nu-SVM (Type 2)*, regresyon için ise *epsilon SVM (Type 1)* ve *nu-SVM (Type 2)* modellerini desteklemektedir. Linear polynomial, RBF ve sigmoid olmak üzere dört farklı kernel kullanılabilir. Bu kernellere ait derece, gamma parametreleri değiştirilebilmektedir. İstenilen katta (fold) çapraz geçiş yapılılabilmektedir. Ayrıca *C* ceza parametresi, *nu* ve *epsilon* değerleri belli bir aralıkta aranabilmektedir (Grid Search). Eğitimleri durdurmak için belirli bir çevrim sayısı veya bir hata miktarı belirtmek mümkündür.

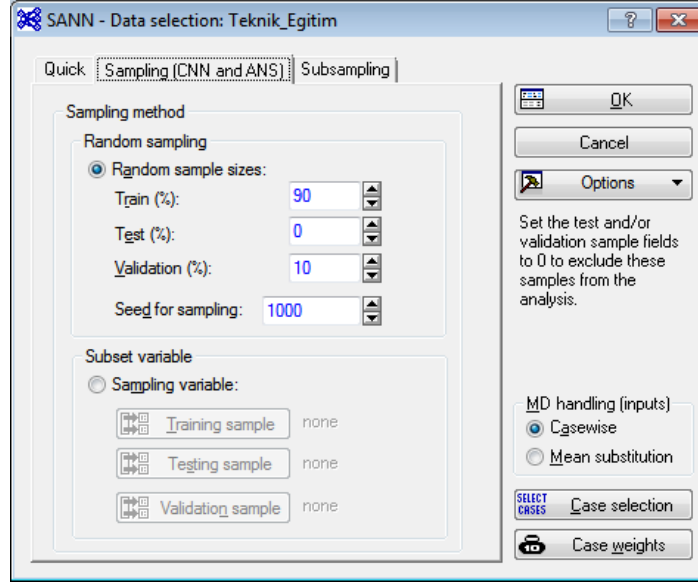
4.3.8 Model Parametrelerinin Belirlenmesi

Yapay sinir ağı modellerinin oluşturulmasında ANS (Automated Network Search - Otomatik Ağ Arama) stratejisi kullanılmıştır (Şekil 40). Böylece belirtilen parametrelere göre arama yapılarak en uygun ağ yapısının bulunması amaçlanmıştır.



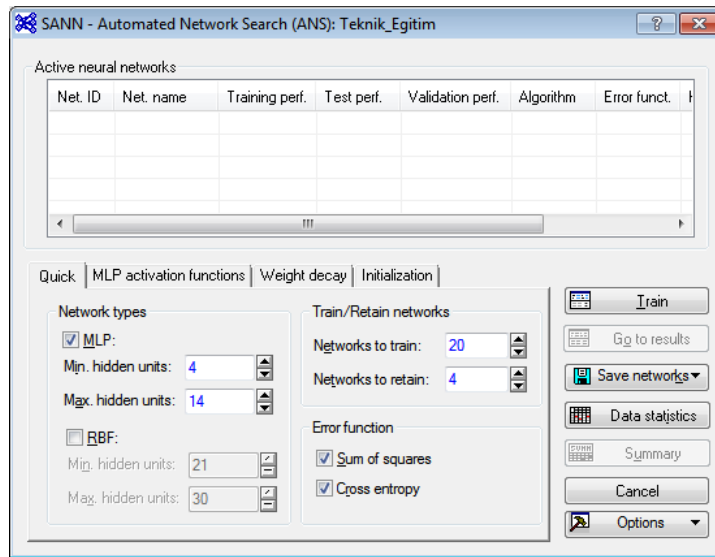
Şekil 40. STATISTICA Otomatik Ağ Arama Modülü

Eğitim için ayrılan kümenin %90'ı eğitim, %10'u geçerleme için kullanılmıştır (Şekil 41).



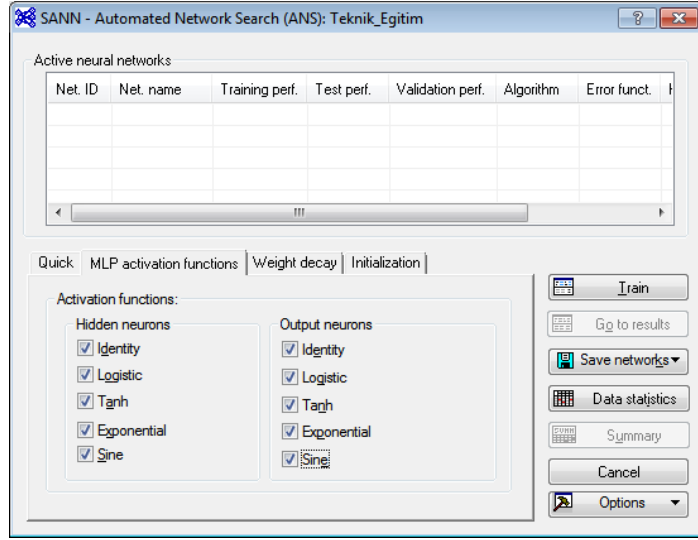
Şekil 41. Eğitim ve Geçerleme Kümesinin Büyüklüklerinin Belirlenmesi

Gizli birim sayıları girdi sayısına bağlı olarak belli bir aralıkta aranabilmektedir. Hata fonksiyonu olarak hata kareleri toplamı (HKT) ve entropi hata fonksiyonları kullanılmıştır. YSA sınıflandırıcıları oluşturulurken her bir veri kümesi üzerinde eğitilen 20 ağın içerisinde en iyi sonuçları veren 4 tanesinin eğitim ve geçerleme kümesindeki performans sonuçlarının raporlanması istenmiştir (Şekil 42).



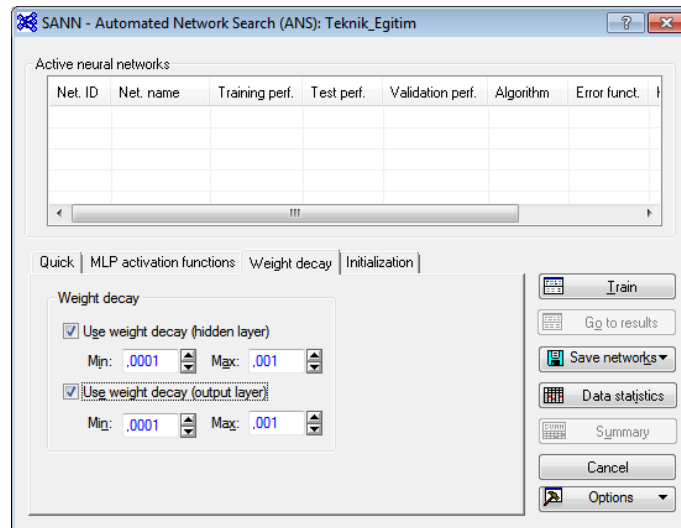
Şekil 42. Ağ Parametrelerinin Belirlenmesi

Gizli katmanda ve çıktı katmanında kullanılacak etkinlik fonksiyonu olarak (Şekil 43) de görülen tüm etkinlik fonksiyonlarının yazılım tarafından denenmesi sağlanacaktır.



Şekil 43. Ağda Kullanılacak Etkinlik Fonksiyonlarının Belirlenmesi

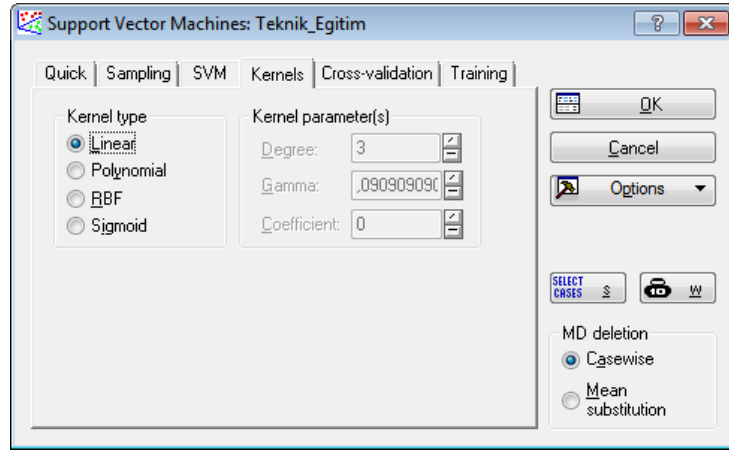
Ağın gereğinden fazla karmaşık olmasını engellemek için girdi ve çıktı katmanlarında ağırlık sönümlenmesi (weight decay) varsayılan değerler arasında kullanılmıştır (Şekil 44).



Şekil 44. Ağırlık Sönümlenme Parametrelerinin Belirlenmesi

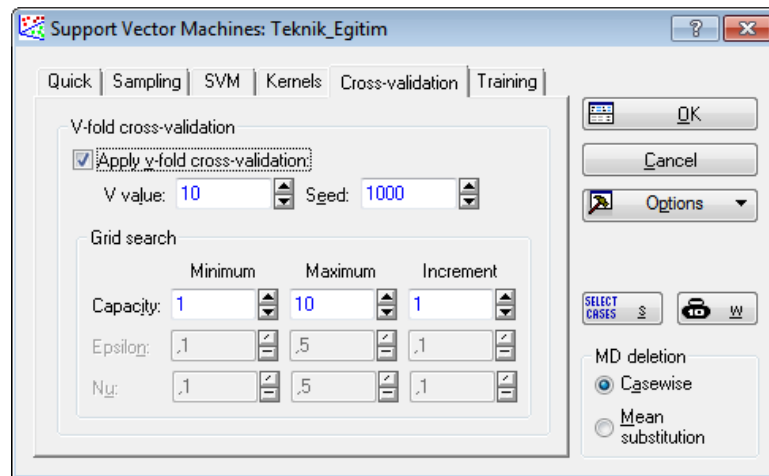
Eđitim sonucunda en iyi sonuları veren ađların parametreleri bir dosyaya kaydedilir. Daha sonra bu dosya test kumesi üzerinde aynı ađların alıřtırabilmesi iin yklenir.

DVM modelleri oluřturulurken linear (dođrusal), polynomial (polinom), RBF (daireysel tabanlı) ve sigmoid kernelleri sırasıyla denenmiřtir (řekil 45).



řekil 45. DVM Sınıflandırıcılarında Kullanılan Kernel Trleri

Ařırı đrenmeyi engellemek iin 10 kat apraz geerleme yapılmıřtır. En uygun C parametresini bulabilmek amacıyla C parametresi 1-10 aralıđında aranmıřtır (řekil 46).



řekil 46. apraz Geerleme ve C Parametresinin Aranması

4.3.9 Sınıflandırıcıların Performans Değerlendirmesi

Sınıflandırıcıların performanslarını karşılaştırmak için “*hata matrisi (confusion matrix)*” kullanılmıştır. Hata matrisi sınıflandırıcının farklı sınıfları ne şekilde tanıdığını analiz etmek için kullanılan bir araçtır (Han vd., 2011: 365).

P: Pozitif gözlem (ilgilenilen esas sınıfa ait gözlemler)

N: Negatif gözlem

DP: Doğru pozitif

DN: Doğru negatif

YP: Yanlış pozitif

YN: Yanlış negatif

sayılarını göstermek üzere hata matrisi (Tablo 19) da gösterildiği gibi oluşturulur.

Tablo 19. Hata Matrisi

		Tahmin		
		<i>Evet</i>	<i>Hayır</i>	Toplam
Gerçek sınıf	<i>Evet</i>	<i>DP</i>	<i>YN</i>	<i>P</i>
	<i>Hayır</i>	<i>YP</i>	<i>DN</i>	<i>N</i>
	Toplam	<i>P'</i>	<i>N'</i>	<i>P + N</i>

DP ve *DN*, sınıflandırıcının ne zaman doğru sınıflandırma yaptığını gösterirken, *YP* ve *YN* yanlış yapılan sınıflandırmaları gösterir. Bir sınıflandırıcının iyi bir *doğruluk (accuracy)* değerine sahip olabilmesi için hata matrisinde köşegenlerin dışındaki değerlerin sıfıra çok yakın olması gerekir. Bir sınıflandırıcının doğruluk değeri doğru sınıflandırılan gözlemlerin oranı olarak hesaplanır.

$$\text{Doğruluk} = \frac{DP + DN}{P + N}$$

Doğruluk, sınıflandırıcının toplam tanıma oranı olarak da bilinir Hata oranı veya yanlış sınıflandırma oranı ($1 - \text{Doğruluk}$) şeklinde hesaplanır.

$$\text{Hata oranı} = \frac{YP + YN}{P + N}$$

4.3.9.1 Teknik Göstergeler ile Endeks Yönü Tahmini

Teknik göstergelerin yer aldığı 1.veri kümesi üzerinde sırasıyla doğrusal polinom, dairesel ve sigmoid kernelleri kullanan DVM sınıflandırıcıları oluşturulmuştur. Test kümesi üzerinde bu sınıflandırıcıların doğru sınıflandırma oranları (Tablo 20) de verilmiştir.

Tablo 20. DVM Sınıflandırıcılarının Teknik Gösterge Verileri Üzerindeki Eğitim ve Test Performansları

Kernel Tipi	Eğitim Performansı (%)	Test Performansı (%)
Doğrusal	71,2	69,3
Polinom	67,7	64,4
Dairesel	73	73,7
Sigmoid	62,3	63,6

Teknik göstergeler veri kümesi üzerinde en iyi performansı gösteren DVM sınıflandırıcısı %73,7 doğru sınıflandırma oranıyla dairesel kerneli kullanan olmuştur. Doğrusal kernel, dairesel kernele yakın eğitim (%71,2) performansı göstermekle birlikte test performansı (%69,3) daha düşüktür. En iyi performansı gösteren DVM sınıflandırıcısı ile ilgili detaylar (Tablo 21) de gösterilmiştir

Tablo 21. Teknik Göstergeler Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri

Özellikler	Değerler
Kernel Tipi	Dairesel Tabanlı
Destek Vektör Sayısı	863 (826 sınırlı)
C (Capacity)	10
Gamma	0,091

Dairesel kernelin test kümesi üzerindeki sınıflandırma özeti (Tablo 22) de verilmiştir. Buna göre endekste artışın olduğu günler %78,7 gibi iyi bir oranda doğru tahmin edilebilmektedir. Endekste düşüşün olduğu günlerin doğru tahmin edilme oranı ise %67,9 olmuştur.

Tablo 22. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Teknik Göstergeler)

Yön	Toplam	Doğru	Yanlış	Doğru(%)	Yanlış (%)
-1	230	156	74	67,9	32,1
1	268	211	57	78,7	21,3

1.veri kümesi üzerinde YSA modelleri oluşturulurken gizli birim sayısı 4-14 arasında aranmıştır. En iyi sonucu veren 4 ağın sonuçları (Tablo 23) de özetlenmiştir.

Tablo 23. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Sınıflandırıcıları (Teknik Göstergeler)

Sıra	Ağ Mimarisi	Eğitim perfor.	Geçerleme perfor.	Hata Fonk.	Gizli Katman Etkinlik Fonk.	Çıktı Katman Etkinlik Fonk.
1	11-4-2	79,8	77,4	Entropi	Tanh	Softmax
2	11-9-2	79,8	79	Entropi	Tanh	Softmax
3	11-12-2	84	80,6	Entropi	Tanh	Softmax
4	11-11-2	86	83	Entropi	Tanh	Softmax

Oluşturulan ağlardan en yüksek eğitim (%86) ve geçerleme (%83) veren, 11-11-2 mimarisine sahip ağ olmuştur. Tüm ağların hata fonksiyonları ile etkinlik fonksiyonları aynıdır. Elde edilen bu ağların test kümesi üzerinde çalıştırılmasıyla bulunan test performansları (Tablo 24) de gösterilmiştir.

Tablo 24. YSA Sınıflandırıcılarının Test Kümesi Performansları (Teknik Göstergeler)

Mimari	Sonuç	Yön(-1)	Yön(+1)	Tümü
11-4-2	Doğru (%)	70,9	87	79,5
	Yanlış (%)	29,1	13	20,5
11-9-2	Doğru (%)	67,4	79,1	73,7
	Yanlış (%)	32,6	20,9	26,3
11-12-2	Doğru (%)	75,7	83,6	80
	Yanlış (%)	24,3	16,4	20
11-11-2	Doğru (%)	70,9	86,1	79,1
	Yanlış (%)	29,1	13,9	20,9

Buna göre test kümesi üzerinden en yüksek performansı (%80) gösteren ağ 11-12-2 ağıdır. YSA modellerinin DVM modelleri gibi artış yönünü daha iyi tahmin ettikleri görülmektedir. Örneğin 11-4-2 ağı artış yönünü %87 gibi gayet iyi bir oranda doğru tahmin edebilmiştir. Diğer ağlar da artış yönünü tatmin edici oranlarda tahmin edebilmektedir. Azalma yönünü %75,7 başarı oranı ile en iyi tahmin eden ağ ise aynı zamanda ortalama olarak en iyi performansı gösteren 11-12-2 ağıdır.

Bu sonuçlar göz önüne alındığında, teknik göstergeler kullanılarak İMKB-100 endeksinin yön tahmininin, %80 doğru sınıflandırma oranına sahip YSA sınıflandırıcıları ile %73,7 performans gösteren DVM sınıflandırıcılarına göre daha iyi yapılabildiği söylenebilir.

4.3.9.2 Borsa Endeks Verileriyle Endeks Yönü Tahmini

Borsa endeks değerlerinin yer aldığı 2.veri kümesi üzerinde DVM modellerinin test kümesindeki sonuçları (Tablo 25) de sunulmuştur. En yüksek test performansını (%56,8) gösteren DVM sınıflandırıcıları dairesel ve sigmoid kernelleri kullanan sınıflandırıcılar olmuştur. Eğitim kümesinde en yüksek (%62,1) performansı doğrusal kernel göstermesine rağmen test performansında ikinci sırada kalmıştır.

Tablo 25. DVM Sınıflandırıcılarının Borsa Endeks Verileri Üzerindeki Eğitim ve Test Performansları

Kernel Tipi	Eğitim Performansı (%)	Test Performansı (%)
Doğrusal	62,1	55,8
Polinom	52,3	53,8
Dairesel	60,4	56,8
Sigmoid	59,5	56,8

Dairesel tabanlı kernel kullanan DVM sınıflandırıcısının detayları (Tablo 26) da gösterilmiştir.

Tablo 26. Borsa Endeks Verileri Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri

Özellikler	Değerler
Kernel Tipi	Dairesel Tabanlı
Destek Vektör Sayısı	1148 (1140 sınırlı)
C (Capacity)	7
Gamma	0,056

Bu sınıflandırıcının yönlere göre sınıflandırma performansını (Tablo 27) yansıtmaktadır. Buna göre sınıflandırıcı artış yönünün tahmininde iyi bir performans (%82,8) gösterirken düşme yönünün tahmininde başarısızdır.

Tablo 27. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Borsa Endeks Verileri)

Yön	Toplam	Doğru	Yanlış	Doğru(%)	Yanlış (%)
-1	230	61	169	26,5	73,5
1	268	222	46	82,8	17,2

2.veri kümesi üzerinde en iyi eğitim ve geçerleme performansını gösteren YSA modelleri (Tablo 28) de yer almaktadır. Modellerin performansları birbirine yakın olmakla birlikte hem eğitim (%62) hem geçerleme performansı (%59,8) en yüksek olan ağ 18-11-2 olmuştur.

Tablo 28. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Borsa Endeks Verileri)

Sıra	Ağ Mimarisi	Eğitim perfor.	Geçerleme perfor.	Hata Fonk.	Gizli Katman Etkinlik Fonk.	Çıktı Katman Etkinlik Fonk.
1	18-17-2	60,5	57,8	Entropi	Lojistik	Lojistik
2	18-9-2	58,8	59,2	Entropi	Sine	Sine
3	18-5-2	61	59,2	HKT	Üssel	Üssel
4	18-11-2	62	59,8	Entropi	Tanh	Tanh

Bu dört modelin test kümesi üzerinde çalıştırılmasıyla elde edilen sonuçlar ise (Tablo 29) da gösterilmiştir. Test kümesi üzerinde en iyi performansı (%57,2), eğitim kümesinde en iyi sonucu veren 18-11-2 modeli vermiştir. Modellerin tamamı artış yönünü daha iyi tahmin edebilmektedirler. 18-9-2 mimarisine sahip model artış yönünü %73,1 oranında tahmin edebilirken düşme yönünü tahminde düşük bir performans (%34,8) göstermiştir.

Tablo 29. YSA Modellerinin Test Kümesi Performansları (Borsa Endeks Verileri)

Mimari	Sonuç	Yön (-1)	Yön (+1)	Tümü
18-17-2	Doğru (%)	41,3	68,7	56
	Yanlış (%)	58,7	31,3	44
18-9-2	Doğru (%)	34,8	73,1	55,4
	Yanlış (%)	65,2	26,9	44,6
18-5-2	Doğru (%)	39,6	70,9	56,4
	Yanlış (%)	60,4	29,1	43,6
18-11-2	Doğru (%)	41,3	70,9	57,2
	Yanlış (%)	58,7	29,1	42,8

Bu sonuçlara göre İMKB-100 endeksinin tahmini için diğer borsaların gecikmeli değişim oranları kullanıldığında DVM ve YSA sınıflandırıcıları birbirine yakın sonuçlar vermektedir (sırasıyla %56,8 ve %57,2). Her iki değişim yönü dikkate alındığında YSA modellerinin DVM modellerine göre çok hafif bir üstünlüğünün olduğu görülmektedir.

4.3.9.3 Makroekonomik Göstergeler ile Endeks Yönü Tahmini

3. veri kümesinde makroekonomik değişkenler bulunmaktadır. Bu veri kümesi üzerinde DVM sınıflandırıcılarının performansı (Tablo 30) da özetlenmiştir. Buna göre en yüksek eğitim (%55,4) ve test performansını (%53,8) yine dairesel tabanlı kernel göstermiştir. Doğrusal ve sigmoid kernelin test performansları ise %50'nin altında kalmıştır.

Tablo 30. Makroekonomik Veriler Üzerinde DVM Sınıflandırıcılarının Eğitim ve Test Performansları

Kernel Tipi	Eğitim Performansı (%)	Test Performansı (%)
Doğrusal	55,2	48,4
Polinom	52,3	53,8
Dairesel	55,4	53,8
Sigmoid	54,8	46,2

Dairesel tabanlı kernel kullanan DVM sınıflandırıcısının detayları (Tablo 31) de gösterilmiştir.

Tablo 31. Makroekonomik Veriler Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri

Özellikler	Değerler
Kernel Tipi	Dairesel Tabanlı
Destek Vektör Sayısı	1094 (1063 sınırlı)
C (Capacity)	8
Gamma	0,2

Bu sınıflandırıcının yönlere göre sınıflandırma performansını (Tablo 32) yansıtmaktadır. Tablo incelendiğinde, DVM sınıflandırıcısının tüm işlem günlerini artış yönünde tahmin ettiği görülmektedir.

Tablo 32. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Makroekonomik Göstergeler)

Yön	Toplam	Doğru	Yanlış	Doğru(%)	Yanlış (%)
-1	230	0	230	0	100
1	268	268	0	100	0

(Tablo 33) 3.veri kümesi üzerinde en yüksek eğitim ve geçerleme performansını veren dört YSA modelini göstermektedir. En iyi geçerleme sonucunu (%58,9) veren ağ, 5-9-2 mimarisine sahip olan ağdır. Ağların eğitim ve geçerleme performansları birbirine çok yakın değerlerde oluşmuştur.

Tablo 33. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Makroekonomik Veriler)

Sıra	Ağ Mimarisi	Eğitim perfor. (%)	Geçerleme perfor. (%)	Hata Fonk.	Gizli Katman Etkinlik Fonk.	Çıktı Katman Etkinlik Fonk.
1	5-3-2	55,5	57,2	HKT	Birim	Üssel
2	5-7-2	55,3	58,1	HKT	Üssel	Üssel
3	5-9-2	56,3	58,9	Entropi	Üssel	Softmax
4	5-4-2	56,4	58,1	Entropi	Üssel	Softmax

Elde edilen bu ağların test kümesi üzerindeki performansları (Tablo 34) de yer almaktadır. Buna göre makroekonomik veriler üzerinde elde edilen en iyi performansı %54,4 ile 5-4-2 ağı sağlamıştır. Bu ağın artış yönünü doğru tahmini %70,5 iken, azalma yönünü tahmin etmede %35,7 gibi düşük bir başarı gösterebilmiştir. 5-7-2 ağı ise artış yönünü %97 oranında doğru tahmin yapmakla birlikte azalma yönünü sadece %3 oranında doğru tahmin edebilmiştir.

Tablo 34. YSA Sınıflandırıcılarının Test Kümesi Performansları (Makroekonomik Veriler)

Mimari	Sonuç	Yön (-1)	Yön (+1)	Tümü
5-3-2	Doğru (%)	46,1	57,8	52,4
	Yanlış (%)	53,9	42,2	47,6
5-7-2	Doğru (%)	3	97	53,6
	Yanlış (%)	97	3	46,4
5-9-2	Doğru (%)	32,6	64,8	49,6
	Yanlış (%)	67,4	35,8	50,4
5-4-2	Doğru (%)	35,7	70,5	54,4
	Yanlış (%)	64,3	29,5	45,6

Sonuçlara göre, makroekonomik değişkenlerin bulunduğu veri kümesiyle yön tahmininde DVM ve YSA modelleri birbirlerine çok yakın sonuçlar (sırasıyla %53,8 ve %54,4) vermiştir. Her iki yöntem de artış yönlerini daha yüksek oranda tahmin etmiştir. DVM sınıflandırıcıları incelendiğinde dairesel tabanlı kernelin diğerlerine göre daha iyi bir performans gösterdiği saptanmıştır. Elde edilen sonuçlar borsa endekslerinin gecikmeli değişim oranlarının ve makroekonomik göstergelerin İMKB-100 endeksinin yön tahmininde teknik gösterge verilerine göre belirgin bir şekilde daha kötü sonuçlar verdiğini göstermektedir.

4.3.9.4 Tüm Verilerin Kullanılmasıyla Endeks Yönü Tahmini

4. veri kümesinde 11 teknik gösterge, 18 borsa endeks değeri ve 5 makroekonomik gösterge olmak üzere 34 bağımsız değişkenin tamamı bir araya getirilmiştir.

Oluşturulan bu veri kümesi üzerinde DVM sınıflandırıcılarının verdiği sonuçlar (Tablo 35) de belirtilmiştir.

Tablo 35. Tüm Veriler Üzerinde DVM Sınıflandırıcılarının Eğitim ve Test Performansları

Kernel Tipi	Eğitim Performansı (%)	Test Performansı (%)
Doğrusal	71,5	71,7
Polinom	64	65
Dairesel	73,4	71,9
Sigmoid	70	70,5

Doğrusal, daireysel ve sigmoid kernel kullanan sınıflandırıcılar birbirlerine yakın test sonuçları vermişlerdir. İçlerinden en iyi test performansını %71,9 ile daireysel tabanlı kernel kullanan DVM sınıflandırıcısı göstermiştir. Bu sınıflandırıcının model detayları (Tablo 36) da yer almaktadır.

Tablo 36. En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri (Tüm Veriler Üzerinde)

Özellikler	Değerler
Kernel Tipi	Dairesel Tabanlı
Destek Vektör Sayısı	877 (847 sınırlı)
C (Capacity)	10
Gamma	0,029

Dairesel tabanlı kerneli kullanan DVM sınıflandırıcısının test kümesi üzerindeki sınıflandırma özeti (Tablo 37) de verilmiştir. Buna göre endekste yükselmenin olduğu günler %78 oranında doğru tahmin edilebilmektedir. Endekste düşüşün olduğu günlerin doğru tahmin edilme oranı ise %64,8 olmuştur.

Tablo 37. Test Kümesi üzerinde Dairesel Tabanlı Kernelin Sınıflandırma Özeti (Tüm Veriler Üzerinde)

Yön	Toplam	Doğru	Yanlış	Doğru(%)	Yanlış (%)
-1	230	149	81	64,8	35,2
1	268	209	59	78	22

(Tablo 38) eğitim kümesi üzerinde en iyi performansı gösteren ağların bilgilerini vermektedir. Buna göre, ağların eğitim performansları DVM sınıflandırıcılarının eğitim performanslarına göre belirgin bir şekilde yüksektir. En iyi eğitim performansını %91 ile 34-8-2 ağı göstermiştir.

Tablo 38. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Tüm Veriler)

Sıra	Ağ Mimarisi	Eğitim perfor. (%)	Geçerleme perfor. (%)	Hata Fonk.	Gizli Katman Etkinlik Fonk.	Çıktı Katman Etkinlik Fonk.
1	34-24-2	84	82,3	HKT	Tanh	Tanh
2	34-8-2	89,7	87,1	HKT	Tanh	Tanh
3	34-8-2	91	84,7	HKT	Lojistik	Birim
4	34-18-2	88,7	86,3	Entropi	Tanh	Softmax

Bu ağların test kümesi üzerindeki detaylı sınıflandırma sonuçları (Tablo 39) de sunulmuştur. Buna göre en yüksek test performansını %73,7 ile 34-8-2 ağı göstermiştir. Tüm YSA modellerinin düşme yönünü yükselme yönüne göre daha iyi tahmin ettikleri görülmektedir.

Tablo 39. YSA Sınıflandırıcılarının Tüm Veriler Üzerindeki Test Performansları

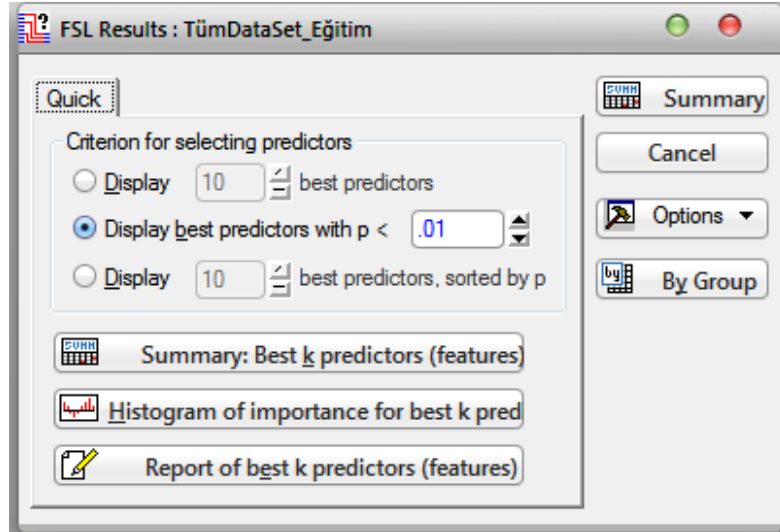
Mimari	Sonuç	Yön (-1)	Yön (+1)	Tümü
34-24-2	Doğru (%)	81,3	58,6	69,1
	Yanlış (%)	18,7	41,4	30,9
34-8-2	Doğru (%)	95,2	49,6	70,7
	Yanlış (%)	4,8	50,4	29,3
34-8-2	Doğru (%)	88,3	61,2	73,7
	Yanlış (%)	11,7	38,8	26,3
34-18-2	Doğru (%)	90	49,3	68,1
	Yanlış (%)	10	50,7	31,9

Buraya kadar elde edilen sonuçlar özetlenirse, ilk 3 veri kümesi üzerindeki en yüksek test performansı hem DVM yöntemiyle hem de YSA yöntemiyle teknik

göstergeleleri kullanan veri kümesi üzerinde elde edilmiştir. DVM sınıflandırıcılarının teknik göstergeler üzerinde %73,7 olan doğru sınıflandırma başarısı 4. veri kümesi üzerinde %71,9'a düşmüştür. YSA sınıflandırıcıların %80 olan doğru sınıflandırma başarısı ise %73,7'ye düşmüştür. Buradan yüksek boyut probleminin (boyut belası) YSA sınıflandırıcılarını DVM sınıflandırıcılarına göre daha çok etkilediği sonucuna varılabilir.

4.3.9.5 Öznitelik Seçimi Yaparak Endeks Yönü Tahmini

34 değişkenin yer aldığı eğitim kümesi üzerinde öznitelik seçimi yapılmıştır. Bu işlem için STATISTICA yazılımının Feature Selection modülünden yararlanılmıştır. Yazılım, öznitelik seçimi için Ki-Kare (Chi-Square) sonuçlarını kullanmaktadır. p değeri .01 den küçük olan bağımsız değişkenlerin seçilmesi istenmiştir (Şekil 47).



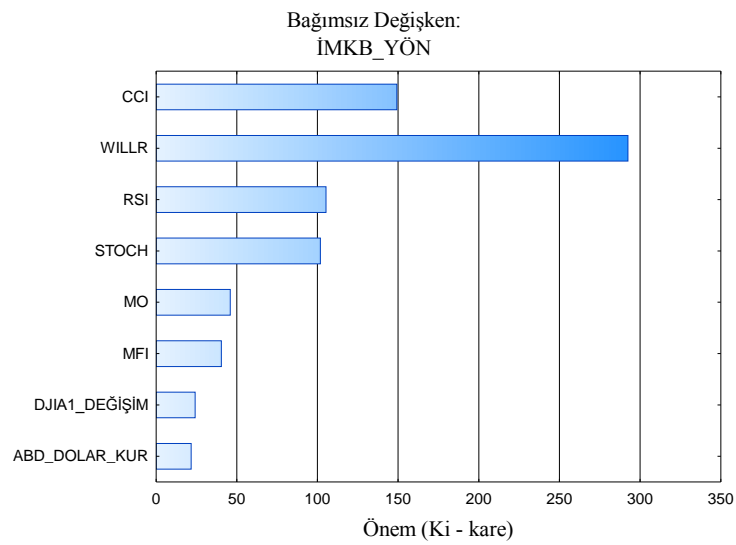
Şekil 47. STATISTICA Feature Selection Ekranı

Buna göre bağımsız değişken olan İMKB-100 endeksinin yönünü en iyi tahmin edeceği düşünülen değişkenlerin Ki - Kare ve p değerleri (Tablo 40) da gösterilmiştir. Öznitelik seçimi sonucunda 34 olan değişken sayısı 8'e düşürülerek boyut azaltma yapılmıştır.

Tablo 40. Öznitelik Seçiminde Kullanılan Ki-kare ve p Değerleri

Değişken	Ki - Kare	p-değeri
CCI	149,2221	0,000000
WILLR	292,4849	0,000000
RSI	105,2608	0,000000
STOCH	101,7980	0,000000
MO	45,9494	0,000001
MFI	40,4241	0,000006
DJIA1_DEĞİŞİM	24,1802	0,001059
ABD_DOLAR_KUR	21,7172	0,005468

Beklenildiği gibi en çok değişken, oluşturulan veri kümeleri içinden en iyi sonuçları veren teknik göstergelerden seçilmiştir. 11 teknik göstergeden 6 tanesi (CCI, WILLR, RSI, STOCH, MO, MFI) seçilmiştir. 18 gecikmeli borsa endeks değişkeninden ise sadece DOW JONES endeksinin 1 gecikmeli değişim oranı (DJIA1_DEĞİŞİM), 5 makroekonomik değişkenden ise sadece ABD Dolar kuru (ABD_DOLAR_KUR) seçilmiştir. Bu sonuçlarla birlikte, İMKB-100 endeksinin yönünün tahmininde teknik göstergelerin diğer borsa endekslerine ve makroekonomik göstergelere göre daha etkili olduğu açıkça görülmektedir. Seçilen değişkenlerin önemleri (Şekil 48) de görülmektedir. En etkili teknik gösterge olarak William's %R göstergesi seçilmiştir.



Şekil 48. Öznitelik Seçimi Sonuçları

Öznitelik seçimi yapıldıktan sonra elde edilen veri kümesi üzerinde oluşturulan DVM sınıflandırıcılarının performansları (Tablo 41) de gösterilmektedir. Buna göre en yüksek doğru sınıflandırmayı yine dairesel tabanlı kernel kullanan DVM sınıflandırıcısı yapmıştır. Bu sınıflandırıcının doğru sınıflandırma oranı %68,5 olmuştur. DVM sınıflandırıcılarının eğitim ve test performanslarının birbirine çok yakın olduğu görülmektedir. Bu da DVM sınıflandırıcılarının genelleme yeteneğinin yüksek olduğunun bir göstergesidir.

Tablo 41. Öznitelik Seçimiyle Oluşturulan Küme Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcılarının Özellikleri

Kernel Tipi	Eğitim Performansı (%)	Test Performansı (%)
Doğrusal	64,3	64,5
Polinom	66,5	65
Dairesel	69	68,5
Sigmoid	63,3	66,3

Sınıflandırıcının model detayları (Tablo 42) de yer almaktadır.

Tablo 42. Öznitelik Seçimi Yapılmış Veri Kümesi Üzerinde En İyi Performansı Gösteren DVM Sınıflandırıcısının Özellikleri

Özellikler	Değerler
Kernel Tipi	Radyal Tabanlı
Destek Vektör Sayısı	878 (854 sınırlı)
C (Capacity)	8
Gamma	0,125

Dairesel tabanlı kerneli kullanan DVM sınıflandırıcının test kümesi üzerindeki sınıflandırma özeti (Tablo 43) de yer almaktadır. Buna göre endekste yükselmenin olduğu günler %77,2 oranında doğru tahmin edilebilmektedir. Endekste düşmenin olduğu günlerin doğru tahmin edilme oranı ise %58,3 olmuştur.

Tablo 43. Test Kümesi üzerinde Dairesel Kernelin Sınıflandırma Özeti (Öznitelik Seçimi Yapılmış Küme)

Yön	Toplam	Doğru	Yanlış	Doğru(%)	Yanlış (%)
-1	230	134	96	58,3	41,7
1	268	207	61	77,2	22,8

5.veri kümesi üzerinde elde edilen en iyi dört ağ (Tablo 44) de sıralanmıştır. Buna göre en yüksek eğitim performansı %81,1 olmuştur.

Tablo 44. Eğitim Kümesi Üzerinde En İyi Sonuçları Veren YSA Modelleri (Öznitelik Seçimi İle Oluşturulan Küme)

Sıra	Ağ Mimarisi	Eğitim perfor. (%)	Geçerleme perfor. (%)	Hata Fonk.	Gizli Katman Etkinlik Fonk.	Çıktı Katman Etkinlik Fonk.
1	8-11-2	80,2	75,8	Entropi	Tanh	Softmax
2	8-9-2	75,5	75	Entropi	Üssel	Softmax
3	8-9-2	81,1	75	Entropi	Tanh	Softmax
4	8-9-2	80,4	75	Entropi	Üssel	Softmax

Bu ağların test kümesi üzerindeki performansları (Tablo 45) de yer almaktadır. Tablo incelendiğinde ağların birbirine çok yakın sonuçlar verdiği görülmektedir. En yüksek sınıflandırma başarısına %73,5 oranı ile 8-11-2 ağı ulaşmıştır.

Tablo 45. Öznitelik Seçimi İle Oluşturulan Veri Kümesi Üzerinde YSA Sınıflandırıcılarının Test Performansları

Mimari	Sonuç	Yön (-1)	Yön (+1)	Tümü
8-11-2	Doğru (%)	64,3	81,3	73,5
	Yanlış (%)	35,6	18,7	26,5
8-9-2	Doğru (%)	57,4	82,8	71,1
	Yanlış (%)	42,6	17,2	28,9
8-9-2	Doğru (%)	68,7	77,2	73,3
	Yanlış (%)	31,3	22,8	26,7
8-9-2	Doğru (%)	64,8	79,5	72,7
	Yanlış (%)	35,2	20,5	27,3

Boyutu azaltılmış veri kümesi üzerinde DVM sınıflandırıcılarının performansı (%68,5) tüm değişkenlerin kullanıldığı veri kümesindeki performansa göre (%71,9) biraz azalmıştır. YSA sınıflandırıcılarında ise tüm değişkenlerin kullanıldığı veri kümesindeki performans (%73,7) ile boyut azaltılmış veri kümesi üzerindeki performans (%73,5) birbirine çok yakındır. Buradan yola çıkarak öznitelik seçimi sonucunda elde edilen daha az sayıdaki (8 adet) değişkenle, tüm değişkenlerin (34 adet) kullanılmasıyla elde edilen performansa yakın bir performans elde edilebileceği görülmektedir.

SONUÇ

Son yıllarda, hisse senedi piyasalarında öngörü yapmak için klasik istatistiksel yöntemlere göre daha az varsayım gerektiren makine öğrenmesi yöntemlerinin kullanımının arttığı görülmektedir. Bu yöntemler örneklerden öğrenme yapabilme ve genelleme yeteneğine sahiptirler. Tezde makine öğrenmesi yöntemlerinden olan Yapay Sinir Ağları ve Destek Vektör Makineleri teorik olarak incelendikten sonra bu yöntemlerin İstanbul Menkul Kıymetler Borsası Ulusal 100 Endeksi'nin yön tahmininde gösterdikleri sınıflandırma performansları incelenmiştir.

Yapay Sinir Ağları (YSA), biyolojik sinir ağlarının çalışması taklit edilerek oluşturulmuş bir yöntemdir. Verideki örüntüleri keşfedebilme, gürültülü verilerle çalışabilme özelliklerine sahip, doğrusal olmayan, esnek modellerdir. Bu özellikleri sayesinde dinamik sistemlerin modellenmesinde yaygın olarak kullanılmaktadırlar. Yeterli sayıda doğrusal olmayan birime sahip bir yapay sinir ağı, örneklerden öğrenme yoluyla herhangi karmaşık bir fonksiyonel ilişkiyi yüksek bir doğrulukla kestirebilmektedir. Yöntemin sınıflandırma, regresyon, kümeleme gibi farklı problem tiplerinde başarıyla uygulandığı birçok çalışma bulunmaktadır.

Destek Vektör Makineleri (DVM) yönteminin temeli İstatistiksel Öğrenme Teorisi'ne dayanmaktadır. Model karmaşıklığını ayarlayabilme, genelleme yeteneği, kernel fonksiyonlarını kullanabilme gibi özellikleri sayesinde borsa öngörü uygulamalarında son yıllarda sıklıkla tercih edilen bir yöntem olmuştur. DVM, sınıfların doğrusal ayrılabilir olduğu durumdaki ikili sınıflandırma probleminde en geniş marja sahip sınıflandırıcıyı oluşturmaya çalışır. Bunun için çözülmesi gereken optimizasyon problemi standart kuadratik programlama teknikleri ile çözülür. DVM doğrusal olmayan karar sınırları oluşturmak için kernel fonksiyonlarını kullanır. Bu fonksiyonlar yardımıyla girdiler daha yüksek boyutlu bir uzaya eşlenir ve doğrusal sınıflandırma problemi bu yüksek boyutlu uzayda çözülür.

Tezin uygulama alanı olan İstanbul Menkul Kıymetler Borsası (İMKB) hakkındaki temel bilgiler ve çeşitli istatistiksel verilere yer verilmiştir. Hisse senedi piyasalarında öngörü yapmak için kullanılan yaklaşımlar ve değişkenlerle ilgili literatür taramaları sunulmuştur. Analizde kullanılan veri kümelerinde yer alan değişkenlerle ilgili açıklamalar ve bu veri kümeleri ile ilgili çeşitli istatistiksel bilgiler aktarılmıştır. Analizlerde oluşturulan sınıflandırıcıların performans karşılaştırmaları tablolar ile gösterilmiştir. İstanbul Menkul Kıymetler Borsası Ulusal 100 Endeksi'nin yön tahmininde hangi veri kümesinin daha etkili olduğu, Yapay Sinir Ağları ve Destek Vektör Makineleri yöntemlerinden hangisinin daha başarılı sonuçlar verdiği araştırılmıştır.

Bu amaçla teknik göstergeler, borsaların endeks değişim oranları, makroekonomik göstergeler, ilk üç veri kümesinin tamamı ve öznitelik seçimi yapılmış veri kümesi olmak üzere 5 farklı veri kümesi oluşturularak endeks yönünün en iyi hangi veri kümesiyle tahmin edilebileceği araştırılmıştır. 2005–2009 dönemi günlük verileri eğitim için, 2010–2011 dönemi günlük verileri test için kullanılmıştır. Eğitim kümesi üzerinde YSA ve DVM yöntemleri kullanılarak eğitim yapıldıktan sonra elde edilen sınıflandırıcılar test kümesi üzerinde çalıştırılarak performansları karşılaştırılmıştır.

Bulunan sonuçlara göre endeksin geçmiş değerleri esas alınarak hesaplanan teknik göstergelerin İMKB-100 endeksinin yönünü belirlemede diğer borsa endekslerine ve makroekonomik değişkenlere göre çok daha faydalı olduğu görülmüştür. 11 Teknik gösterge kullanılarak elde edilen en iyi test performansı %80 iken, bu oran 18 borsa endeksinin kullanıldığı veri kümesi üzerinde %57,2 makroekonomik göstergelerin (5 adet) yer aldığı veri kümesi üzerinde ise %54,4 olabilmektedir. Tüm veri kümelerinin birleştirilmesiyle elde edilen 34 değişkene sahip veri kümesi üzerinde elde edilen en yüksek test performansı %73,7 olmuştur. Bu veri kümesi üzerinde öznitelik seçimi yapılmasıyla endeks yönünü en az bağımsız değişkenle en yüksek oranda tahmin edebilecek veri kümesi oluşturulmuştur. Böylece 34 olan toplam değişken sayısı 8'e düşürülmüştür. Bu veri kümesinde 6

teknik gösterge, 1 borsa endeksi, 1 makroekonomik gösterge bulunmaktadır. Boyut azaltılmış veri kümesindeki en iyi sonuç %73,5 olarak elde edilmiştir.

İncelenen veri kümeleri üzerinde YSA ve DVM yöntemlerinden en yüksek performansı her zaman YSA göstermiştir. Buna göre İMKB gibi gelişmekte olan bir borsada endeks yönünü tahmin etmek için bu yöntemlerin faydalı araçlar olduğu görülmekle birlikte YSA yönteminin DVM yöntemine göre daha az tahmin hatası oluşturduğu sonucuna varılmıştır.

Bulunan sonuçları daha da geliştirmek için ilerideki çalışmalarda bulanık mantık ile YSA veya DVM yöntemlerinin bir arada kullanıldığı hibrit yöntemler kullanılabilir. Arama yapmak için genetik algoritmaların kullanılması da başka bir seçenektir.

Sınıflandırıcıların performansları iki yolla arttırılabilir. Birincisi parametrelerin daha hassas bir şekilde ve daha geniş bir aralıkta ayarlanmasıdır. İkinci yol ise çalışmadaki veri kümesinde yer alan değişkenlere ek olarak konuyla ilgili olabilecek diğer değişkenlerin eklenmesidir.

KAYNAKÇA

- Abe, Shigeo, **Support Vector Machines for Pattern Classification**, London, Springer, 2010.
- Akcan, Ahmet, Kartal, Cem: “İMKB Sigorta Endeksini Oluşturan Şirketlerin Hisse Senedi Fiyatlarının Yapay Sinir Ağları İle Tahmini”, **Muhasebe ve Finansman Dergisi**, Temmuz 2011, s.27–40.
- Alavala, Chennakesava R., **Fuzzy Logic and Neural Networks: Basic Concepts and Applications**, New Age International Pvt Ltd Publishers, 2008.
- Alpaydın, Ethem, **Introduction to Machine Learning**, 2.bs., ABD, The MIT Press, 2009.
- Alpaydın, Ethem, **Yapay Öğrenme**, İstanbul, Boğaziçi Üniversitesi Yayınevi, 2011.
- Altay, Erdinç, Satman, M. Hakan: “Stock Market Forecasting: Artificial Neural Network and Linear Regression Comparison in an Emerging Market”, **Journal of Financial Management and Analysis**, C:18, No: 2, 2005, s.18–33.
- Atan, Sibel Duman, Özdemir, Zeynel Abidin, Atan, Murat: “Hisse Senedi Piyasasında Zayıf Formda Etkinlik: İMKB Üzerine Ampirik Bir Çalışma”, **Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, C:24, No:2, 2009, s.33-48.
- Aygören, Hakan: “İstanbul Menkul Kıymetler Borsasının (İMKB) Fractal Analizi”, **Dokuz Eylül Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi**, C:23, No:1, 2008, s.125–134.
- Barber, David, **Bayesian Reasoning and Machine Learning**, UK, Cambridge University Press, 2012.

- Belousov, A. I., Verzakov, S. A., Frese, J. Von: “A Flexible Classification Approach with Optimal Generalisation Performance: Support Vector Machines”, **Chemometrics and Intelligent Laboratory Systems**, C:64, No:1, 2002, s.15–25.
- Bennett, Kristin P., Campbell, Colin: “Support Vector Machines: Hype or Hallelujah?”, **ACM SIGKDD Explorations Newsletter**, C:2, No:2, 2000, s.1–13.
- Bigus, Joseph P., **Data Mining with Neural Networks: Solving Business Problems from Application Development to Decision Support**, ABD, McGraw-Hill Inc., 1996.
- Bishop, Christopher M., **Neural Networks for Pattern Recognition**, Oxford, Clarendon Press, 1995.
- Bishop, Christopher M., **Pattern Recognition and Machine Learning**, Singapur, Springer, 2006.
- Blum, Avrim: “**Machine Learning Theory**”, (Çevrimiçi) <http://www.cs.cmu.edu/~avrim/Talks/mlt.pdf>, 2 Aralık 2012.
- Borsa İstanbul: “**Borsa İstanbul Eğitim Klavuzları**”, (Çevrimiçi) <http://borsaistanbul.com/data/egitimklavuzlari/KLVZ17ENDEKS.PDF>, 12 Mayıs 2013.
- Borsa İstanbul: “**İMKB Faaliyet Raporu 2011**”, (Çevrimiçi), http://borsaistanbul.com/kurumsal/yayinlar/faaliyet_raporlari/fr2011.zip, 2 Mayıs 2013.
- Boser, Bernhard E., Guyon, Isabelle M., Vapnik, Vladimir N.: “A Training Algorithm for Optimal Margin Classifiers”, **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**, 1992, s.144–152.

- Bousquet, Olivier, Boucheron, Stéphane, Lugosi, Gábor: “Introduction to Statistical Learning Theory”, **Advanced Lectures on Machine Learning**, 2004, s.169–207.
- Boyacioglu, Melek Acar, Avci, Derya: “An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the Prediction of Stock Market Return: The Case of the Istanbul Stock Exchange”, **Expert Systems with Applications**, C:37, No:12, 2010, s.7908–7912.
- Boyd, Stephen, Vandenberghe, Lieven, **Convex Optimization**, UK, Cambridge University Press, 2004.
- Bramer, Max, **Principles of Data Mining**, London, Springer, 2007.
- Brunato, Mauro, Battiti, Roberto: “Statistical Learning Theory for Location Fingerprinting in Wireless LANs”, **Computer Networks**, C:47, No:6, 2005, s.825–845.
- Burges, Christopher J. C.: “A Tutorial on Support Vector Machines for Pattern Recognition”, **Data Mining and Knowledge Discovery**, C:2, No:2, 1998, s.121–167.
- Camastra, Francesco, Vinciarelli, Alessandro, **Machine Learning for Audio, Image and Video Analysis: Theory and Applications**, London, Springer, 2007.
- Campbell, Colin, Ying, Yiming, **Learning with Support Vector Machines**, Morgan & Claypool Publishers, 2011.
- Cherkassky, Vladimir, Mulier, Filip, **Learning from Data: Concepts, Theory, and Methods**. 2.bs., ABD, Wiley-IEEE Press, 2007.
- Cherkassky, Vladimir, Shao, Xuhui, Mulier, Filip M., Vapnik, Vladimir N.: “Model Complexity Control for Regression Using VC Generalization Bounds”, **IEEE Transactions On Neural Networks**, C:10, No:5, 1999, s.1075–1089.

- Choudhry, Rohit, Garg, Kumkum: "A Hybrid Machine Learning System for Stock Market Forecasting", **Proceedings of World Academy of Science, Engineering and Technology**, C:29, May 2008, s.315–318.
- Cios, Krzysztof J., Pedrycz, Witold, Swiniarski, Roman W., Kurgan, Lukasz Andrzej, **Data Mining: A Knowledge Discovery Approach**, Springer, 2007.
- Clarke, Bertrand, Fokoue, Ernest, Hao, Zhang, Helen, **Principles and Theory for Data Mining and Machine Learning**, Springer, 2009.
- Cortes, Corinna, Vapnik Vladimir: "Support-vector Networks", **Machine Learning**, C:20, No:3,1995, s.273–297.
- Cox, P. G., Adhami, R.: "Multi-class Support Vector Machine Classifier Applied to Hyper-spectral Data", **Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory**, 2002, s.271–274.
- Cristianini, Nello, Schölkopf, Bernhard: "Support Vector Machines and Kernel Methods: The New Generation of Learning Machines", **AI Magazine**, C:23, No:3, 2002, s.31-41.
- Cristianini, Nello, Shawe-Taylor, John, **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods**, UK, Cambridge University Press, 2000.
- Cucker, Felipe, Zhou, Ding Xuan, **Learning Theory: An Approximation Theory Viewpoint**, New York, Cambridge University Press, 2007.
- Cura, Tunçhan, **Modern Sezgisel Teknikler ve Uygulamaları**, İstanbul, Papatya Yayıncılık, 2008.
- Cybenko, G.: "Continuous Valued Neural Networks With Two Hidden Layer Are Sufficient", **Technical Report**, Tufts University, Medford, 1988.

- Cybenko., G.: "Approximations by Superpositions of Sigmoidal Functions", **Mathematics of Control, Signals and Systems**, C:2, No:4, 1989, s.303-314.
- Çetinyokuş, Tahsin, Gökçen, Hadi: "Borsada Göstergelerle Teknik Analiz İçin Bir Karar Destek Sistemi", **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, C:17, No:1, 2002, s.43–58.
- Çinko, Murat, Avcı, Emin: "A Comparision of Neural Network and Linear Regression Forecasts of The ISE-100 Index", **Öneri**, C:7, No:28, Haziran 2007, s.301–307.
- Dağ, Funda: "Yapay Zekâ: Temel Kavramlar", **Türkiye Bilişim Ansiklopedisi**, Ed. Ören, T., Çölkesen, R., İstanbul, Papatya Yayıncılık, 2006, s.931-939.
- Dağlı, Hüseyin: "Hisse Senedi Piyasa Endeksleri Ve Türkiye", **Balıkesir Üniversitesi Sosyal Bilimler Enstitüsü Dergisi**, C:3, No:4, 2000, s.189–206.
- Demireli, Erhan, Akkaya, Göktuğ Cenk, İbaş, Elif: "Finansal Piyasa Etkinliği: S&P 500 Üzerine Bir Uygulama", **C.Ü. İktisadi Ve İdari Bilimler Dergisi**, C:11, No:2, 2010, s.53–67.
- Demuth, Howard, **Neural Network Toolbox for Use with MATLAB: User's Guide**, MathWorks Inc., 1992.
- Dietterich, Thomas. G.: "Machine Learning", **Nature Encyclopedia of Cognitive Science**, London, Macmillan, 2003.
- Diler, Ali İhsan: "İMKB Ulusal 100 Endeksinin Yönünün Yapay Sinir Ağları Hata Geriye Yayma Yöntemi ile Tahmin Edilmesi", **İMKB Dergisi**, C:7, No:25-26, 2003, s.65-81.
- Dogan, Nukhet, Yalcin, Yeliz: "The Effects of the Exchange Rate Movements on the Istanbul Stock Exchange", **Applied Financial Economics Letters**, C:3, No:1, January 2007: s.39–46.

- Domingos, Pedro: “A Few Useful Things to Know About Machine Learning”, **Communications of the ACM**, C:55, No:10, 2012, s.78–87.
- Dreyfus, Gérard, **Neural Networks: Methodology and Applications**, Berlin, Springer, 2010.
- Du, K.L, Swamy, M.N.S., **Neural Networks in a Softcomputing Framework**, Springer, 2006.
- Duda, Richard O., Hart, Peter E., Stork, David G., **Pattern Classification**, 2.bs., Wiley-Blackwell, 2000.
- Dunis, Christian, Rosillo, L., Rafael, Fuente, David de la, Pino, Raúl: “Forecasting IBEX-35 Moves Using SupportVectorMachines”, **Neural Computing and Applications**, C:23, No:1, 2012, s. 229-236.
- Efe, M. Önder, Kaynak, Okyay, **Yapay Sinir Ağları Ve Uygulamaları**, İstanbul, Boğaziçi Üniversitesi Yayınları, 2000.
- Eken, Hasan, Adalı, Sait: “Piyasa Etkinliği ve İMKB: Zayıf Formda Etkinliğe İlişkin Ekonometrik Bir Analiz”, **MUFAD**, No:37, 2008, s.74-87.
- Elmas, Çetin, **Yapay Zekâ Uygulamaları**, İstanbul, Seçkin Yayıncılık, 2007.
- Engelbrecht, Andries P., **Computational Intelligence: An Introduction**, 2.bs., Wiley, 2007.
- Erdem, Cumhuri, Arslan, Cem Kaan, Erdem, Meziyet Sema: “Effects of Macroeconomic Variables on Istanbul Stock Exchange Indexes”, **Applied Financial Economics**, C:15, No:14, October 2005, s.987–994.
- Erdinç, Yaşar, **Yatırımcı ve Teknik Analiz Sorgulanıyor**, Ankara, Siyasal Kitabevi, 2004.
- Ertel, Wolfgang, **Introduction to Artificial Intelligence**, London, Springer, 2011.

- Eryiğit, Mehmet: “Effects of Oil Price Changes on the Sector Indices of Istanbul Stock Exchange”, **International Research Journal of Finance and Economics**, No:25, 2009, s.209–216.
- Evgeniou, Theodoros, Poggio, Tomaso, Pontil, Massimiliano, Verri, Alessandro: “Regularization and Statistical Learning Theory for Data Analysis”, **Computational Statistics & Data Analysis**, C:38, No:4, February 2002, s.421–432.
- Evgeniou, Theodoros, Pontil, Massimiliano, Poggio, Tomaso: “Statistical Learning Theory: A Primer”, **International Journal of Computer Vision**, C:38, No:1, June 2000, s.9–13.
- Fama, E. F.: “Efficient Capital Markets: A Review of Theory and Empirical Works”, **Journal of Finance**, C:25, No:2, 1970, s.383–417.
- Fan, Rong-En, Chen, Pai-Hsuen, Lin, Chih-Jen: “Working Set Selection Using Second Order Information for Training Support Vector Machines”, **The Journal of Machine Learning Research**, C:6, 2005, s.1889–1918.
- Flach, Peter, **Machine Learning: The Art and Science of Algorithms That Make Sense of Data**, Cambridge University Press, 2012.
- Flood, I., Kartam, N.: “Neural Networks in Civil Engineering. I: Principles and Understanding”, **Journal of Computing in Civil Engineering**, C:8, No:2, 1994, s.131–148.
- Forster, Malcolm R.: “Key Concepts in Model Selection: Performance and Generalizability”, **Journal of Mathematical Psychology**, C:44, No:1, 2000, s.205-231.
- Fradkin, Dmitriy, Muchnik, Ilya: “Support Vector Machines for Classification” **Discrete Methods in Epidemiology**, C:70, 2006, s.13–20.

- Gençtürk, Mehmet: “Finansal Kriz Dönemlerinde Makroekonomik Faktörlerin Hisse Senedi Fiyatlarına Etkisi”, **Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, C:14, No:1, 2009, s:127-136.
- Gorunescu, Florin, **Data Mining: Concepts, Models and Techniques**, Berlin, Springer, 2011.
- Göstergeler, “**Teknik Göstergeler**”, (Çevrimiçi)
<http://www.witforex.com/index.php/gostergeler/teknik-gostergeler>, 10 Mart 2013.
- Gunn, Steve R.: “Support Vector Machines for Classification and Regression”, **Technical Report**, University of Southampton, May 1998.
- Hamel, Lutz H., **Knowledge Discovery with Support Vector Machines**, New Jersey, Wiley-Interscience, 2009.
- Han, Jiawei, Kamber, Micheline, Pei, Jian, **Data Mining: Concepts and Techniques**, 3.bs., ABD, Morgan Kaufmann, 2011.
- Harrington, Peter, **Machine Learning in Action**, ABD, Manning Publications, 2012.
- Haykin, Simon, **Neural Networks: A Comprehensive Foundation**, 2.bs., Singapur, Prentice Hall, 1999.
- Herbrich, Ralf, **Learning Kernel Classifiers: Theory and Algorithms**, Massachusetts, The MIT Press, 2001.
- Hornik, Kurt, Stinchcombe, Maxwell B., White, Halbert: “Multilayer Feedforward Networks are Universal Approximators”, **Neural Networks**, C:2, No:5, 1989, s.359-366.
- Hsieh, William W., **Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels**, New York, Cambridge University Press, 2009.

- Huang, Te-Ming, Kecman, Vojislav, Kopriva, Ivica, **Kernel Based Algorithms for Mining Huge Data Sets: Supervised, Semi-supervised, and Unsupervised Learning**, Springer, 2006.
- Huang, Wei, Nakamori, Yoshiteru, Wang, Shou-Yang: “Forecasting Stock Market Movement Direction with Support Vector Machine”, **Computers & Operations Research**, C:32, No:10, October 2005, s.2513–2522.
- Ivanciuc, Ovidiu: “Applications of Support Vector Machines in Chemistry”, **Reviews in Computational Chemistry**, Ed. Boyd, Donald B., Lipkowitz, Kenny B., Cundari, Thomas R., C:26, New Jersey, John Wiley & Sons, 2007.
- İçke, Başak Turan, Aytürk, Yusuf: “Fiyat-Kazanç Oranı Etkisinin Değer Yatırım Stratejileri Kapsamında Analizi: İMKB İçin Ampirik Bir Uygulama”, **Öneri**, C:9, No:35, Ocak 2011, s.103-115.
- Jain, A. K., Mao, Jianchang, Mohiuddin, K. M.: “Artificial Neural Networks: a Tutorial”, **Computer**, C:29, No:3, March 1996, s.31–44.
- Kaastra, Iebling, Boyd, Milton: “Designing a Neural Network for Forecasting Financial and Economic Time Series”, **Neurocomputing**, C:10, No:3, April 1996, s.215–236.
- Kantardzic, Mehmed, **Data Mining: Concepts, Models, Methods, and Algorithms**, 2.bs., New Jersey, Wiley-Blackwell, 2011.
- Kara, Yakup, Boyacioglu, Melek Acar, Baykan, Ömer Kaan: “Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange”, **Expert Systems with Applications**, C:38, No:5, 2011, s.5311–5319.
- Kearns, Michael J., Vazirani, Umesh, **An Introduction to Computational Learning Theory**, Cambridge, The MIT Press, 1994.

- Kecman, Vojislav, **Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models**, Cambridge, A Bradford Book, 2001.
- Kim, Kyoung-jae, Han, Ingoo: “Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index”, **Expert Systems with Applications**, C:19, No:2, 2000, s.125–132.
- Kim, Kyoung-jae: “Financial Time Series Forecasting Using Support Vector Machines”, **Neurocomputing**, C:55, No:1–2, September 2003, s.307–319.
- Korkmaz, Turhan, Çevik, Emrah I., Birkan, Elif, Özataç, Nesrin: “Causality in Mean and Variance Between ISE 100 and S&P 500: Turkcell Case”, **African Journal of Business Management**, C:5, No:5, 2011, s.1673–1683.
- Kovalerchuk, Boris, Vityaev, Evgenii, **Data Mining in Finance: Advances in Relational and Hybrid Methods**, Springer, 2000.
- Kriesel, David, **A Brief Introduction to Neural Networks**, available at <http://www.dkriesel.com>, 2007.
- Kulkarni, Sanjeev R., Harman, Gilbert: “Statistical Learning Theory: a Tutorial”, **Wiley Interdisciplinary Reviews: Computational Statistics**, C:3, No:6, November 2011, s.543–556.
- Kutlu, Birgül, Badur, Bertan: “Yapay Sinir Ağları Ile Borsa Endeksi Tahmini”, **Yönetim**, C:20, No:63, 2009, s.25–40.
- Larose, Daniel T., **Discovering Knowledge in Data: An Introduction to Data Mining**, Wiley-Interscience, 2005.
- Lazar, Nicole: “Ockham’s Razor”, **Wiley Interdisciplinary Reviews: Computational Statistics**, C:2, No:2, 2010, s.243–246.

- Li, Zhao, Yongchun, Wu, Xuexia, Wu: “Research of Information Security Risk Management Based on Statistical Learning Theory”, **Proceedings of the 2009 International Forum on Computer Science-Technology and Applications**, C:3, 2009, s.436–438.
- Linoff, Gordon S., Berry, Michael J., **Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management**, 2.bs., John Wiley & Sons, 2004.
- MacKay, David J. C., **Information Theory, Inference and Learning Algorithms**, Cambridge University Press, 2003.
- Maloof, Marcus A., **Machine Learning and Data Mining for Computer Security: Methods and Applications**, Springer, 2005.
- Mammone, Alessia, Turchi, Marco, Cristianini, Nello: “Support Vector Machines”, **Wiley Interdisciplinary Reviews: Computational Statistics**, C:1, No:3, 2009, s.283–289.
- Mangasarian, Olvi L., Musicant, David R.: “Lagrangian Support Vector Machines”, **The Journal of Machine Learning Research**, C:1, 2001, s.161–177.
- Marsland, Stephen, **Machine Learning: An Algorithmic Perspective**, Chapman and Hall/CRC, 2009.
- Mavroforakis, Michael E., Theodoridis, Sergios: “A Geometric Approach to Support Vector Machine (SVM) Classification”, **IEEE Transactions on Neural Networks**, C:17, No:3, May 2006, s.671–682.
- Mehrotra, Kishan, Mohan, Chilukuri K., Ranka, Sanjay, **Elements of Artificial Neural Networks. Complex Adaptive Systems Series**, The MIT Press, 1997.
- Mitchell, Tom, **Machine Learning**, ABD, McGraw-Hill, 1997.

- Moguerza, Javier M., Muñoz, Alberto: “Support Vector Machines with Applications”, **Statistical Science**, C:21, No:3, August 2006, s.322–336.
- Mohri, Mehryar, Rostamizadeh, Afshin, Talwalkar, Ameet, **Foundations of Machine Learning**, ABD, The MIT Press, 2012.
- Munakata, Toshinori, **Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More**, 2.bs., London, Springer, 2008.
- Murphy, Kevin P., **Machine Learning: a Probabilistic Perspective**, Cambridge, The MIT Press, 2012.
- Nasien, Dewi, Yuhaniz, Siti S., Haron, Habibollah: “Statistical Learning Theory and Support Vector Machines”, **Proceedings of the 2010 Second International Conference on Computer Research and Development**, 2010, s.760–764.
- Negnevitsky, Michael, **Artificial Intelligence: A Guide to Intelligent Systems**, 2.bs., Essex, Addison-Wesley, 2004.
- Nello, Cristianini, Schölkopf, Bernhard: “Support Vector Machines and Kernel Methods: The New Generation of Learning Machines”, **AI Magazine**, C:23, No:3, 2002, s.31-41.
- Nilsson, Nills J., **Introduction to Machine Learning: An Early Draft of A Proposed Textbook**, Stanford University, 1998.
- Nisbet, Robert, Elder IV, John, Miner, Gary, **Handbook of Statistical Analysis and Data Mining Applications**, Kanada, Academic Press, 2009.
- Olson, David L., Delen, Dursun, **Advanced Data Mining Techniques**, Berlin, Springer, 2008.
- Osuna, Edgar, Freund, Robert, Girosi, Federico: “Support Vector Machines: Training and Applications”, **Technical Report**, Massachusetts Institute of Technology, ABD, 1997.

- Ozun, Alper: “Are The Reactions of Emerging Equity Markets to The Volatility in Advanced Markets Similar?: Comparative Evidence From Brazil and Turkey”, **International Research Journal of Finance and Economics**, C: 9, 2007, s.220–230.
- Özdemir, A. Kerem, Tolun, Seda, Demirci, Ebru: “Endeks Getirisi Yönünün İkili Sınıflandırma Yöntemiyle Tahmin Edilmesi: IMKB-100 Endeks Örneği”, **Niğde Üniversitesi İİBF Dergisi**, C:4, No:2, 2011, s.45–59.
- Öztemel, Ercan, **Yapay Sinir Ağları**, 2.bs., İstanbul, Papatya Yayıncılık, 2006.
- Peretto, Pierre, **An Introduction to the Modeling of Neural Networks**, Cambridge University Press, 1992.
- Perşembe, Ali, **Teknik Analiz mi Dedin? Hadi Canım Sen De! Üçüncü Kitap**, 5.bs., İstanbul, Scala Yayıncılık, 2010.
- Platt, J.: “Fast Training of Support Vector Machines Using Sequential Minimal Optimization”, **Advances in Kernel Methods Support Vector Learning**, (Ed.) Schölkopf, B., Burges, C. J., Smola, A. J., Cambridge MA, The MIT Press, 1999, s.185–208.
- Pontil, Massimiliano, Verri, Alessandro: “Properties of Support Vector Machines”, **Neural Computation**, C:10, No:4,1998, s.955–974.
- Press, William H., Teukolsky, Saul A., Vetterling, William T., Flannery, Brian P., **Numerical Recipes, The Art of Scientific Computing**, 3.bs., Cambridge University Press, 2007.
- Ramon, Manel Martinez, Christodoulou, Christos, **Support Vector Machines for Antenna Array Processing and Electromagnetics**, ABD, Morgan & Claypool Publishers, 2006.
- Reed, Russell D., Marks, Robert J., **Neural Smithing: Supervised Learning**, The MIT Press, 1999.

- Rojas, Raul, **Neural Networks: A Systematic Introduction**, Springer, 1996.
- Rumelhart, David E., Hinton, Geoffrey E., Williams, Ronald J.: “Learning Representations by Back-propagating Errors”, **Nature**, C:323, No:6088, 1986: s.533–536.
- Russell, Stuart, Norvig, Peter, **Artificial Intelligence: A Modern Approach**, 3.bs., New Jersey, Prentice Hall, 2009.
- Samarasinghe, Sandhya, **Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition**, Auerbach Publications, 2006.
- Sayılğan, Güven, Süslü, Cemil: “Makroekonomik Faktörlerin Hisse Senedi Getirilerine Etkisi: Türkiye Ve Gelişmekte Olan Piyasalar Üzerine Bir İnceleme”, **BDDK Bankacılık Ve Finansal Piyasalar Dergisi**, C:5, No:1, 2011: s.73–96.
- Schölkopf, Bernhard, Smola, Alexander J., **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**, Cambridge MA, The MIT Press, 2001.
- Shawe-Taylor, John, Sun, Shiliang: “A Review of Optimization Methodologies in Support Vector Machines”, **Neurocomputing**, C:74, No:17, 2011, s.3609–3618.
- Shin, Kyung-Shik, Lee, Taik Soo, Kim, Hyun-jung: “An Application of Support Vector Machines in Bankruptcy Prediction Model”, **Expert Systems with Applications**, C:28, No:1, 2005, s.127–135.
- Songur, Hilmi: “Küresel Krizin İstanbul Borsası Üzerindeki Etkisi”, **Maliye Finans Yazıları**, C:22, No:82, 2009, s.93–108.
- Steinwart, Ingo, Christmann, Andreas, **Support Vector Machines**, Springer, 2008.

- Suykens, Johan A. K., Gestel, Tony Van, De Brabanter, Jos, De Moor, Bart, Vandewalle, Joos, **Least Squares Support Vector Machines**, Singapur, World Scientific Publishing Company, 2002.
- Tarassenko, Lionel, **Guide to Neural Computing Applications**, Butterworth-Heinemann, 1998.
- TSPAKB: “**Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi**”, (Çevrimiçi)https://www.tspakb.org.tr/tr/Portals/0/ETM_KILAVUZLAR/ETM_kilavuzlar_ileri_analiz_yontemleri_EKIM_2011.pdf, 2011, 3 Nisan 2013.
- TSPAKB: “**Türkiye Sermaye Piyasası 2011 Raporu**”, İstanbul, Mayıs 2012.
- Türsoy, Turgut, Nil, Günsel, Rjoub, Husam: “Macroeconomic Factors, the APT and the Istanbul Stock Market”, **International Research Journal of Finance and Economics**, No:22, 2008, s.49–57.
- Valiant, Leslie G.: “A Theory of the Learnable”, **Communications of the ACM**, C: 27, No:11, 1984, s.1134–1142.
- Vapnik, Vladimir N., **Statistical Learning Theory**, New York, Wiley-Interscience, 1998.
- Vapnik, Vladimir N., **The Nature of Statistical Learning Theory**, New York, Springer, 2000.
- Vapnik, Vladimir N.: “An Overview of Statistical Learning Theory”, **IEEE Transactions On Neural Networks**, C:10, No:5, 1999: s.988–999.
- Vellido, Alfredo, Lisboa, Paulo J. G., Vaughan, J.: “Neural Networks in Business: a Survey of Applications (1992–1998)”, **Expert Systems with Applications**, C:17, No:1, 1999, s.51–70.

- Venugopal, Venu, Baets, W.: “Neural Networks and Statistical Techniques in Marketing Research: a Conceptual Comparison”, **Marketing Intelligence & Planning**, C:12, No:7, 1994, s.30–38.
- Vercellis, Carlo, **Business Intelligence: Data Mining and Optimization for Decision Making**, Wiley, 2009.
- Von Luxburg, Ulrike, Schölkopf, Bernhard: “Statistical Learning Theory: Models, Concepts, and Results”, **arXiv Preprint arXiv:0810.4752**, 2008.
- Vuran, Bengü: “The Determination of Long-run Relationship Between ISE 100 and International Equity Indices Using Cointegration Analysis”, **Istanbul University Journal of the School of Business Administration**, C:39, No:1, 2010, s.154–168.
- Wang, Xiangying, Zhong, Yixin: “Statistical Learning Theory and State of the Art in SVM”, **Proceedings of the Second IEEE International Conference on Cognitive Informatics**, 2003, s.55–59.
- Warren, Warle S.: “Neural Networks and Statistical Models”, **Proceedings of the Nineteenth Annual SAS Users Group International Conference**, Texas, SAS Publishing, 1994.
- Wasserman, Larry, **All of Statistics: A Concise Course in Statistical Inference**, Springer, 2003.
- Witten, Ian H., Eibe, Frank, **Data Mining: Practical Machine Learning Tools and Techniques**, 2.bs., ABD, Morgan Kaufmann, 2005.
- Wu, Chih-Hung, Ken, Yun, Huang, Tao: “Patent Classification System Using a New Hybrid Genetic Algorithm Support Vector Machine”, **Applied Soft Computing**, C:10, No:4, September 2010: s.1164–1177.

- Yayar, Rüştü, Karaca, S. Serdar: “Endeks Tahmini Ve İstanbul Menkul Kıymetler Borsası (İMKB) Sınai Endeksi Üzerine Bir Uygulama”, **MÖDAV**, No:12, 2011, s.163–188.
- Yegnanarayana, B., **Artificial Neural Networks**, New Delhi, Prentice-Hall of India, 2005.
- Yeung, Daniel S., Cloete, Ian, Shi, Daming, Ng, Wing W. Y., **Sensitivity Analysis for Neural Networks**, Berlin, Springer, 2009.
- Yıldız, Birol, **Finansal Analizde Yapay Zekâ**, Ankara, Detay Yayıncılık, 2009.
- Yıldız, Birol, Yalama, Abdullah, Coşkun, Metin: “Forecasting the Istanbul Stock Exchange National 100 Index Using an Artificial Neural Network”, **World Academy of Science, Engineering and Technology**, No:46, 2008, s.36–39.
- Yüksel, Ebru, Güteryüz, Güldal: “İstanbul Menkul Kıymetler Borsası’nda Eşhareketlilik Ve Asimetrik Ayarlama”, **Maliye Finans Yazıları**, C:24, No:88, Temmuz 2010, s.97–113.
- Zurada, Jacek M., **Introduction to Artificial Neural Systems**, West Group, 1992.
- Zügül, Muhittin, Şahin, Cumhuri: “İMKB 100 Endeksi İle Bazı Makroekonomik Değişkenler Arasındaki İlişkiyi İncelemeye Yönelik Bir Uygulama”, **Akademik Bakış**, No:16, Nisan 2009, s.1–16.

ÖZGEÇMİŞ

Şenol Emir 1980 yılında Bursa' da doğmuştur. Liseyi Mustafakemalpaşa Anadolu Öğretmen Lisesi'nde tamamladı. 2002 yılında burslu olarak okuduğu Beykent Üniversitesi Matematik-Bilgisayar bölümünü bitirdi. Yüksek lisans derecesini İstanbul Üniversitesi Enformatik Bölümü'nden 2006 yılında almıştır. 2007 yılında İstanbul Üniversitesi Sayısal Yöntemler Anabilim Dalı'nda doktora bilimsel hazırlık programıyla başladı. 2003 yılından bu yana Beykent Üniversitesi Meslek Yüksekokulu Bilgisayar Programcılığı' nda öğretim görevlisi ve program sorumlusu olarak çalışmaktadır. Akademik ilgi konuları makine öğrenmesi, optimizasyon, yöneylem araştırması, istatistik ve hesaplama algoritmalarıdır. Yazar, evli ve bir çocuk babasıdır. İyi derecede İngilizce bilmektedir.