

**T.C.
İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANA BİLİM DALI
ÜRETİM YÖNETİMİ BİLİM DALI**

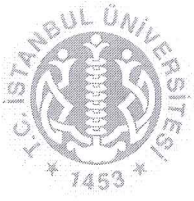
DOKTORA TEZİ

**ARALIKLI TALEP YAPISINA SAHİP
ÜRÜNLERİN TALEP TAHMİNİNDE MAKİNE
ÖĞRENME YÖNTEMLERİNİN UYGULANMASI**

**DERYA SAATÇIOĞLU
2502100307**

**TEZ DANIŞMANI
PROF. DR. NECDET ÖZÇAKAR**

İSTANBUL, 2016



T.C.
İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



DOKTORA
TEZ ONAYI

ÖĞRENCİNİN;

Adı ve Soyadı : DERYA SAATÇIOĞLU Numarası : 25021003070
Anabilim Dalı /
Anasanat Dalı / Programı : ÜRETİM Danışmanı : PROF.DR NECDET ÖZÇAKAR
Tez Savunma Tarihi : 15.02.2016 Saati : 13:00
Tez Başlığı : ARALIK TALEP YAPISINA SAHİP ÜRÜNLERİN TALEP TAHMİNİNDE MAKİNE
ÖĞRENME YÖNTEMLERİNİN UYGULAMASI

TEZ SAVUNMA SINAVI, İÜ Lisansüstü Eğitim-Öğretim Yönetmeliği'nin 50. Maddesi uyarınca yapılmış,
sorulara alınan cevaplar sonunda adayın tezinin KABULÜNE OYBİRLİĞİ / OYÇOKLUĞUYLA karar verilmiştir.

JÜRİ ÜYESİ	İMZA	KANAATİ (KABUL / RED / DÜZELTME)
1-PROF.DR NECDET ÖZÇAKAR		KABUL
2- PROF.DR ALP BARAY		KABUL
3- PROF.DR YAMAN ÖZTEK		KABUL
4- DOÇ.DR ALİ GÖRENER		KABUL
5- YRD.DOÇ.DR BİRGÜL KÜÇÜKÇİRPİN		KABUL

YEDEK JÜRİ ÜYESİ	İMZA	KANAATİ (KABUL / RED / DÜZELTME)
1- PROF.DR SELİM ZAIM		
2-DOÇ.DR ÖZLEM AKÇAY KASAPOĞLU		

ÖZ

ARALIKLI TALEP YAPISINA SAHİP ÜRÜNLERİN TALEP TAHMİNİNDE MAKİNE ÖĞRENME YÖNTEMLERİNİN UYGULANMASI

DERYA SAATÇIOĞLU

Talep tahmini ve doğruluğunun bir işletmenin başarısına ve müşteri memnuniyetine doğrudan etkisi bulunmaktadır. Düzgün talep yapısına sahip ürünlerin talep tahmini ve planlamasında birçok yöntem başarılı sonuçlar verirken çoğu zaman diliminde sıfır talep gören aralıklı talep yapısına sahip ürünlerin talep tahmininde başarılı olamamaktadır. Talep büyüklüğünün ve şeklinin değişkenliği bu ürünler için talep tahmini ve planlamanın yapılmasını zorlaştırmaktadır. Bu tez çalışmasında, aralıklı talep yapısına sahip ürünlerin talep tahminini için basit üstel düzeltme, Croston yöntemi ve modifikasyonları ile makine öğrenme yöntemlerinden olan Destek Vektör Makineleri ve aynı zamanda bir yapay sinir ağı modeli olan Çok Katmanlı Algılayıcılar ve Öz Örgütlemeli Harita Ağları incelenmiştir. Bu yöntemler e-ticaret sektöründe faaliyet gösteren bir işletmenin satış yaptığı bir kategorideki aralıklı talep yapısına sahip ürünlerinin talep tahmini için kullanılmıştır. Daha sonra her bir yöntemin performansı uygun ölçütler kullanılarak karşılaştırılmıştır. Sonuçlar incelendiğinde en iyi performansın bir yapay sinir ağı modeli olan Öz Örgütlemeli Harita Ağlarının ve ona çok yakın değerlere sahip olan Çok Katmanlı Algılayıcılar yönteminin sağladığı görülmektedir.

Anahtar Sözcükler: *Aralıklı Talep, Makine Öğrenmesi, Yapay Sinir Ağları*

ABSTRACT

IMPLEMENTING MACHINE LEARNING METHODS IN FORECASTING INTERMITTENT DEMAND

DERYA SAATÇIOĞLU

Forecasting and accuracy of demand has a direct effect in the success of a business and customer satisfaction. Whereas many methods show successful results in forecasting and planning demand of products with smooth demand, they fail with products that have many time periods with zero demands. Variability of capacity and pattern of demand causes forecasting and planning of it to become difficult. In this dissertation study, for forecasting intermittent demand, simple exponential smoothing, Croston Method and modifications, Support Vector Machines; one of the machine learning methods, Multilayer Perceptron and Self Organizing Maps, which is also an artificial neural network have been analyzed. These methods have been used for forecasting intermittent demand in one of the categories of a business that operates in e-trade sector. Afterwards, performance of each method has been compared using appropriate accuracy measures. When the results are examined, it can be seen that the best performance is been delivered by Self Organising Maps which is an artificial neural network model and Multilayer Perceptrons which has a very close value to Self Organising Maps.

Key Words: *Intermittent Demand, Machine Learning, Artificial Neural Networks*

ÖNSÖZ

Bu tez çalışmasında Basit Üstel Düzeltme, Croston yöntemi ve modifikasyonları, Destek Vektör Makineleri ve bir yapay sinir ağı modeli olan Çok Katmanlı Algılayıcılar ve Öz Örgütlemeli Harita Ağları yöntemlerinin aralıklı talep tahmini için bir e-ticaret sitesindeki verilerle uygulanabilirlikleri araştırılmış ve gösterdikleri performanslar karşılaştırılmıştır.

Tez çalışmasının teorik kısmında sırasıyla aralıklı talep tahmini, makine öğrenmesi, Destek Vektör Makineleri ve Yapay Sinir Ağları konularına yer verilmiştir. Uygulama kısmında ise e-ticaret sektöründe faaliyet gösteren bir işletmenin satış yaptığı bir kategorideki aralıklı talep yapısına sahip ürünleri için Basit Üstel Düzeltme, Croston yöntemi ve modifikasyonları, Destek Vektör Makineleri, Çok Katmanlı Algılayıcılar ve Öz Örgütlemeli Harita Ağları yöntemleri ile talep tahmini yapılmış ve performansları ölçülmüştür. Yapılan karşılaştırılma sonucunda en iyi performansı sağlayan ilk iki yöntemin Öz Örgütlemeli Harita Ağları ve Çok Katmanlı Algılayıcılar yöntemlerinin olduğu görülmüştür.

Doktora eğitimim ve tez çalışmam boyunca göstermiş olduğu her türlü destek ve yardımdan dolayı çok değerli danışmanım Prof. Dr. Necdet Özçakar, Prof. Dr. Alp Baray, Yrd. Doç Dr. Birgül Küçük Çırpın ve Yrd. Doç Dr. Tuncay Özcan'a teşekkürlerimi sunarım.

Ayrıca çalışmamda kullandığım yöntemlerle ilgili kodların yazılma süresinde göstermiş oldukları destekten ötürü Dr. Etem Deniz ve Velican Küpeli'ye teşekkür ederim.

Son olarak bu zorlu süreçte varlıklarını her zaman yanımda hissettiren sevgili aileme, eşim Gürhan Saatçioğlu'na ve oğullarım Kaan ve Ozan'a çok teşekkür ediyorum.

Derya Saatçioğlu
İstanbul, Şubat 2016

İÇİNDEKİLER

ÖZ	iii
ABSTRACT	iv
ÖNSÖZ.....	v
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ.....	xii
KISALTMALAR LİSTESİ.....	xiv
GİRİŞ.....	1

BİRİNCİ BÖLÜM

ARALIKLI TALEP YAPISINA SAHİP ÜRÜNLERİN TALEP TAHMİNİ

1.1 Talep Tahmini	3
1.1.1 Tahmin Sistemindeki Yedi Adım	4
1.1.2 Tahminin Karakteristik Özellikleri	4
1.1.3 İyi Bir Tahminin Unsurları	5
1.1.4 Tahmin Yöntemleri	5
1.1.4.1 Kalitatif Yöntemler	5
1.1.4.2 Kantitatif Yöntemleri	7
1.2 Talep Şekillerinin Sınıflandırılması	7
1.3 Aralıklı Talebin Tanımı.....	11
1.4 Aralıklı Talep Tahmininde Kullanılan Yöntemler	12
1.4.1 Basit Üstel Düzeltme Yöntemi (SES)	12
1.4.2 Hareketli Ortalamalar Yöntemi	13
1.4.3 Croston Yöntemi	14
1.4.4 Croston Yönteminin Modifikasyonları ve Karşılaştırmaları	16
1.4.5 Aralıklı Talep Tahmininde Kullanılan Diğer Yöntem ve Algoritmalar	20
1.5 Tahmin Doğruluğunun Ölçümü	21

1.5.1	Ortalama Hataların Karesi (MSE)	22
1.5.2	Ortalama Mutlak Sapma (MAD).....	22
1.5.3	Ortalama Mutlak Yüzde Hata (MAPE).....	23
1.5.4	Ortalama Hata Kareleri Karekökü (RMSE)	23
1.5.5	Simetrik Ortalama Mutlak Yüzde Hata (sMAPE).....	24
1.5.6	Ortalama Mutlak Sapmanın Geometrik Ortalaması (GMAMAD).....	24
1.5.7	En İyi Yüzde Oranı (PB)	25
1.5.8	Ortalama Mutlak Ölçekli Hata (MASE).....	25

İKİNCİ BÖLÜM

MAKİNE ÖĞRENMESİ

2.1	Makine Öğrenmesinin Tanımı.....	27
2.2	Makine Öğrenmesinin Amacı ve Önemi.....	29
2.3	Makine Öğrenmesi Uygulama Alanları.....	31
2.4	Makine Öğrenme Problemleri.....	32
2.4.1	Sınıflandırma	32
2.4.2	Regresyon	34
2.4.3	Sıralama.....	35
2.4.4	Kümeleme.....	37
2.4.5	Boyut İndirgeme	40
2.5	Makine Öğrenme Tipleri.....	41
2.5.1	Denetimli Öğrenme	42
2.5.2	Denetimsiz Öğrenme	43
2.5.3	Yarı Denetimli Öğrenme	45
2.5.4	Transdüktif Öğrenme.....	46
2.5.5	Takviyeli Öğrenme.....	48
2.5.6	Aktif Öğrenme.....	49
2.5.7	Çevrimiçi Öğrenme	50

ÜÇÜNCÜ BÖLÜM

DESTEK VEKTÖR MAKİNELERİ

3.1	Destek Vektör Makinesi Tanımı.....	51
------------	---	-----------

3.2 İstatiksel Öğrenme Teorisi (Vapnik - Chervonenkis Teorisi).....	52
3.2.1 Deneysel Risk Minimizasyonu.....	54
3.2.2 VC Boyutu.....	55
3.2.3 Yapısal Risk Minimizasyonu	58
3.3 Sınıflandırma İçin Destek Vektör Makineleri.....	59
3.3.1 İki Sınıflı Problemler için Destek Vektör Makineleri	60
3.3.1.1 En İyi Ayıran Hiper Düzlem	60
3.3.1.2 Doğrusal Ayrılabilir Sert Kenarlı Destek Vektör Makineleri.....	61
3.3.1.3 Doğrusal Ayrılmaz Yumuşak Kenarlı Destek Vektör Makineleri.....	64
3.3.1.4 Kernel Yöntemleri.....	67
3.3.1.4.1 Çekirdek (Kernel) Hilesi ve Mercer Koşulu	68
3.3.1.4.2 Kernel Fonksiyon Çeşitleri	70
3.3.2 Çok Sınıflı Problemler İçin Destek Vektör Makineleri.....	71
3.4 Regresyon için Destek Vektör Makineleri.....	73
3.5 Karesel Programlama Problemlerinin Çözümünde Kullanılan	
Algoritmik Yaklaşımlar.....	76
3.5.1 Chunking ve Osuna Algoritması	77
3.5.2 Ardışık Minimal Optimizasyon Yöntemi (SMO).....	78
3.6 Destek Vektör Makineleri Avantaj ve Dezavantajları	81

DÖRDÜNCÜ BÖLÜM

YAPAY SİNİR AĞLARI

4.1 Yapay Sinir Ağları Tanımı ve Kullanım Alanları	84
4.2 Yapay Sinir Ağlarının Tarihi Gelişimi	85
4.3 Yapay Sinir Ağlarının Yapısı ve Ana Öğeleri	88
4.3.1 Biyolojik Sinir Hücresi.....	88
4.3.2 Yapay Sinir Hücresi	89
4.3.3 Yapay Sinir Ağlarının Yapısı	92
4.4 Yapay Sinir Ağlarında Temel Öğrenme Kuralları.....	93
4.5 Yapay Sinir Ağı Mimarilerinin Sınıflandırılması.....	95
4.6 Tek Katmanlı Algılayıcılar.....	96
4.6.1 Perceptron.....	97

4.6.2	XOR Problemi	99
4.6.3	ADALINE/MADALINE Modelleri	100
4.7	Çok Katmanlı Algılayıcılar	101
4.7.1	Geriye Yayılım Algoritması	103
4.7.2	İleri ve Geri Hesaplama	104
4.8	Yapay Sinir Ağlarında Dikkat Edilmesi Gereken Konular	106
4.8.1	Veri Tabanı Büyüklüğü ve Bölümlenmesi	106
4.8.2	Girdi ve Çıktıların Gösterimi	108
4.8.3	Ağdaki İlk Ağırlıkların Belirlenmesi	109
4.8.4	Öğrenme ve Momentum Katsayısı	109
4.8.5	Yakınsama ve Yerel Minimum	110
4.8.6	Aktivasyon İşlevi	110
4.8.7	Gizli Katmanların Büyüklüğü	111
4.8.8	Ağın Eğitilmesi	111
4.8.9	Verilerin Ölçeklendirilmesi	112
4.8.10	Genelleme, Aşırı Eğitim ve Eğitimi Durdurma Zamanı	113
4.9	Özörgütlemeli Harita Ağları (Self Organizing Maps)	115
4.9.1	Özörgütlemeli Harita Ağlarının Yapısı	115
4.9.2	Özörgütlemeli Harita Ağlarının Eğitilmesi	117
4.9.3	Özörgütlemeli Harita Ağlarının Algoritması	118

BEŞİNCİ BÖLÜM

UYGULAMA

5.1	Uygulamanın Amacı ve Kapsamı	123
5.2	Veri Setinin Hazırlanması	123
5.3	Aralıklı Talebin Sınıflandırılması	124
5.4	Veri Analizi ve Bulgular	125
5.4.1	Basit Üstel Düzeltme Yöntemi	126
5.4.2	Croston Yöntemi	128
5.4.3	Croston Yönteminin Modifikasyonları	131
5.4.3.1	Syntetos & Boylan Modifikasyonu	131
5.4.3.2	Leven & Segerstedt Modifikasyonu	133

5.4.4 Destek Vektör Makinesi Yöntemi	136
5.4.5 Çok Katmanlı Algılayıcılar Yöntemi	139
5.4.6 SOM Ağları Yöntemi	142
5.5 Aralıklı Talep Yapısına Sahip Ürünlerin Talep Tahmini İçin Kullanılan Yöntemlerin Karşılaştırılması.....	151
SONUÇ VE ÖNERİLER.....	157
KAYNAKÇA	160
ÖZGEMİŞ.....	170

TABLolar LİSESİ

Tablo 1	Varyans bölümlleme fikrine dayalı olarak talebin sınıflandırılması.....	9
Tablo 2	Standart ölçülere göre en iyi tahmin yöntemi	19
Tablo 3	SES yöntemine göre gerçekleşen ve tahmin edilen talep değerleri	128
Tablo 4	Croston yöntemine göre gerçekleşen ve tahmin edilen talep değerleri.	130
Tablo 5	Croston S&B yöntemine göre gerçekleşen ve tahmin edilen talep değerleri.....	132
Tablo 6	Croston L&S yöntemine göre gerçekleşen ve tahmin edilen talep değerleri.....	134
Tablo 7	SES, Croston ve modifikasyonları için optimum alfa değerleri	135
Tablo 8	DVM yöntemine için kullanılan parametreler	137
Tablo 9	DVM yöntemine göre gerçekleşen ve tahmin edilen talep değerleri....	138
Tablo 10	ÇKA yöntemi için kullanılan parametreler	140
Tablo 11	ÇKA yöntemine göre gerçekleşen ve tahmin edilen talep değerleri...	141
Tablo 12	Som_randinit fonksiyonu için kullanılan parametreler.....	146
Tablo 13	Som_seqtrain fonksiyonu için kullanılan parametreler.....	146
Tablo 14	SOM ağlarında kullanılan parametreler	149
Tablo 15	SOM ağları yöntemine göre gerçekleşen ve tahmin edilen talep değerleri.....	150
Tablo 16	MSE ölçütüne göre yöntemlerin karşılaştırılması.....	151
Tablo 17	RMSE ölçütüne göre yöntemlerin karşılaştırılması	152
Tablo 18	MASE ölçütüne göre yöntemlerin karşılaştırılması.....	153
Tablo 19	MAD ölçütüne göre yöntemlerin karşılaştırılması.....	154
Tablo 20	sMAPE ölçütüne göre yöntemlerin karşılaştırılması	155
Tablo 21	Tüm ölçütlere göre yöntemlerin karşılaştırılması	156

ŞEKİLLER LİSESİ

Şekil 1	Talep şekillerinin sınıflandırılması	10
Şekil 2	Makine öğrenmesi ve sınıflandırma	29
Şekil 3	Sınıflandırma probleminde karar sınıfları	34
Şekil 4	K-ortalamar yöntemi örnek sonuçları	39
Şekil 5	Denetimli öğrenme	43
Şekil 6	Denetimsiz öğrenme	44
Şekil 7	Yarı denetimli öğrenme	46
Şekil 8	Takviyeli öğrenme	49
Şekil 9	Üç noktanın bir düzlem ile parçalanması	55
Şekil 10	Doğrusal fonksiyon kümesinin VC boyutu	56
Şekil 11	Sinüs dalgasının VC boyutu	57
Şekil 12	İki sınıfı ayıran hiper düzlemler	60
Şekil 13	En iyi ayıran hiper düzlem.....	62
Şekil 14	İki boyutlu uzayda doğrusal ayrılama durumu	65
Şekil 15	Kernel fonksiyonu	68
Şekil 16	Bire tüm yönteminde sınıflandırılmayan bölgeler	73
Şekil 17	ϵ –duyarsız kayıp fonksiyonu	75
Şekil 18	Chunking, Osuna Algoritması ve ardışık minimal optimizasyon.....	81
Şekil 19	Biyolojik sinir hücresi	89
Şekil 20	Yapay sinir hücresi	90
Şekil 21	Farklı aktivasyon işlevleri.....	91
Şekil 22	Bir yapay sinir ağı örneği	92
Şekil 23	Kara kutu benzetimi.....	93
Şekil 24	Bir yapay sinir ağının ileri ve geri beslemeli topolojisi.....	96
Şekil 25	Perceptron modeli.....	97
Şekil 26	Perceptron modeliyle yapılan sınıflandırma.....	98
Şekil 27	XOR probleminin girdi ve çıktıları.....	100

Şekil 28	ADALINE ünitesinin yapısı	101
Şekil 29	Çok katmanlı algılayıcıların yapısı	102
Şekil 30	Eğitim, test ve geçerlik alt kümeleri	107
Şekil 31	Eğitimi durdurma zamanı	114
Şekil 32	Özörgütlemeli harita ağlarında topoloji örnekleri	116
Şekil 33	Özörgütlemeli harita ağlarının yapısı	117
Şekil 34	Topoloji fonksiyonları	119
Şekil 35	Bir nöronun topolojik komşuluk örnekleri	120
Şekil 36	SOM ağları haritalama örneği 1	121
Şekil 37	SOM ağları haritalama örneği 2	121
Şekil 38	SOM ağları haritalama örneği 3	122
Şekil 39	Aralıklı talebe sahip bir ürünün talep zaman grafiği	125
Şekil 40	Periyodik ilerleme mekanizması	126
Şekil 41	SES yöntemine göre bir ürünün talep zaman grafiği.....	127
Şekil 42	Croston yöntemine göre bir ürünün talep zaman grafiği.....	130
Şekil 43	Croston S&B yöntemine göre bir ürünün talep zaman grafiği.....	132
Şekil 44	Croston L&S yöntemine göre bir ürünün talep zaman grafiği	134
Şekil 45	DVM yöntemine göre bir ürünün talep zaman grafiği	138
Şekil 46	ÇKA modeli.....	140
Şekil 47	ÇKA yöntemine göre bir ürünün talep zaman grafiği	141
Şekil 48	Bir ürün için SOM ağları haritalama örneği (başlangıç ağırlıkları)	147
Şekil 49	17 döngüden sonra bir ürün için SOM ağları haritalama örneği	147
Şekil 50	34 döngüden sonra bir ürün için SOM ağları haritalama örneği	148
Şekil 51	50 döngüden sonra bir ürün için SOM ağları haritalama örneği	148
Şekil 52	SOM ağları yöntemine göre bir ürünün talep zaman grafiği.....	150
Şekil 53	MSE ölçütüne göre yöntemlerin karşılaştırılma grafiği	151
Şekil 54	RMSE ölçütüne göre yöntemlerin karşılaştırılma grafiği.....	152
Şekil 55	MASE ölçütüne göre yöntemlerin karşılaştırılma grafiği	153
Şekil 56	MAD ölçütüne göre yöntemlerin karşılaştırılma grafiği	154
Şekil 57	sMAPE ölçütüne göre yöntemlerin karşılaştırılma grafiği.....	155
Şekil 58	Tüm ölçütlere göre yöntemlerin karşılaştırılma grafiği.....	156

KISALTMALAR LİSTESİ

ART	: Adaftif Rezonans Teorisi
CFE	: Kümülatif Tahmin Hatası
ÇKA	: Çok Katmanlı Algılayıcılar
DVM	: Destek Vektör Makineleri
GMAMAD	: Ortalama Mutlak Sapmanın Geometrik Ortalaması
KKT	: Karush – Kuhn – Tucker
L&S	: Leven ve Segerstedt
MAD	: Ortalama Mutlak Sapma
MAPE	: Ortalama Mutlak Yüzde Hata
MASE	: Ortalama Mutlak Ölçekli Hata
MSE	: Ortalama Hataların Karesi
PB	: En İyi Yüzde Oranı
RMSE	: Ortalama Hata Kareleri Karekökü
RBF	: Dairesel Tabanlı / Gaussian Fonksiyonu
S&B	: Syntetos ve Boylan
SES	: Basit Üstel Düzeltme
sMAPE	: Simetrik Ortalama Mutlak Yüzde Hata
SMO	: Ardışık Minimal Optimizasyon
SOM	: Öz Örgütlemeli Haritalar
VC	: Vapnik – Chervonenkis
VQTAM	: Vektör Nicemlenmiş Geçici Çağrışımsal Bellek

GİRİŞ

Rekabetin yoğun olarak yaşandığı günümüz Pazar koşullarında talep tahmini ve planlama işletmeler açısından büyük önem taşımaktadır. Talep tahmini ve planlamanın doğruluğu bir işletmenin başarısını doğrudan etkilemektedir. Doğru talep tahmini ile müşteri memnuniyeti arttırılmakta, stok fazlalığı azaltılmakta, üretim ve satın alma faaliyetleri daha etkili hale gelmekte, taşıma faaliyetleri daha iyi organize edilmekte ve bu sayede maliyetler de azaltılmaktadır.

Düzenli bir talep yapısına sahip olan ürünlerin talep tahmininde klasik talep tahmini yöntemleri başarılı sonuçlar vermektedir. Fakat rastgele gerçekleşen, çoğu zaman periyodunda sıfır talep gören aralıklı talep yapısına sahip ürünlerin talep tahmininde bu yöntemler başarısız olmaktadır. Talep büyüklüğünün ve şeklinin değişkenliği, bu ürünlerin talep tahmin ve planlamasının yapılmasını zorlaştırmaktadır. Bu sebeple bu çalışmada aralıklı talep yapısına sahip ürünlerin talep tahmini ve planlaması için klasik yöntemlerin dışında daha gelişmiş makine öğrenme yöntemleri uygulanıp, her bir yöntemin performansı ölçülerek karşılaştırma yapılmıştır.

Basit üstel düzeltme (SES), Croston yöntemi ve modifikasyonları, destek vektör makineleri (DVM) ve bir yapay sinir ağı modeli olan çok katmanlı algılayıcılar (ÇKA) ve öz örgütleme harita (SOM) ağlarının aralıklı talep tahmini için uygulanabilirliklerinin araştırıldığı bu tez çalışması beş bölümden oluşmaktadır.

Birinci bölümde talep tahmininden bahsedilerek aralıklı talebin tanımı yapılmış ve talep şekillerinin nasıl sınıflandırıldığına üzerinde durulmuştur. Daha sonra aralıklı talep tahmininde kullanılan yöntemler incelenip tahmin doğruluğunun saptanabilmesi için gerekli ölçütler anlatılmıştır.

İkinci bölümde ise makine öğrenmesinin tanımı, amacı, önemi ve uygulama alanlarından bahsedilerek makine öğrenme problemleri ve tipleri üzerinde durulmuştur.

İstatiksel öğrenme teorisini temel alan DVM ise üçüncü bölümde incelenmiştir. Bu bölümde öncelikle istatiksel öğrenme teorisi, Vapnik–Chervonenkis (VC) boyutu, deneysel risk minimizasyonu ve yapısal risk minimizasyonu gibi kavramlar açıklanmıştır. Daha sonra sınıflandırma ve regresyon için DVM incelenmiş olup DVM'nin avantaj ve dezavantajlarından bahsedilmiştir.

Dördüncü bölümde yapay sinir ağlarının tanımı, kullanım alanları, tarihsel gelişimi, yapısı ve ana öğeleri üzerinde durularak yapay sinir ağı modelleri anlatılmıştır. Bölümün sonunda ise yapay sinir ağlarında dikkat edilmesi gereken konular incelenerek SOM ağları hakkında bilgi verilmiştir.

Son bölümde ise tezin uygulama kısmı yer almaktadır. Uygulama kısmında e-ticaret sektöründe faaliyet gösteren bir işletmenin aralıklı talep yapısına sahip ürünlerinin talep tahmininde; SES, Croston yöntemi ve modifikasyonları, DVM ve bir yapay sinir ağı modeli olan ÇKA ve SOM ağları kullanılarak talep tahmini yapılmıştır. Daha sonra bu yöntemlerin tahmin hataları karşılaştırılarak performansları ölçülmüştür. Elde edilen sonuçlar doğrultusunda aralıklı talep yapısına sahip ürünlerinin talep tahmininde hangi yöntemin daha başarılı olduğu saptanmıştır.

BİRİNCİ BÖLÜM

ARALIKLI TALEP YAPISINA SAHİP ÜRÜNLERİN TALEP TAHMİNİ

Bu bölümde öncelikle talep tahmininden bahsedilerek aralıklı talebin tanımı yapılacak ve talep şekillerinin nasıl sınıflandırıldığına üstünde durulacaktır. Daha sonra aralıklı talep tahmininde kullanılan yöntemler incelenip tahmin doğruluğunun saptanabilmesi için gerekli ölçütler anlatılacaktır.

1.1 Talep Tahmini

Tahminler, gelecekteki talep hakkında bilgi edinilmesi açısından bilgi sağladıkları için operasyon yönetimindeki karar süreçlerinin temel girdisini oluşturmaktadır. Operasyon yönetiminin öncelikli amacı arz ve talep arasındaki dengenin sağlanmasıdır. Talebin karşılanmasında kapasite ve arzın belirlenmesi için talep tahmininin yapılması önemli bir gerekliliktir. Operasyon bölümü, personel ve ekipman kararlarını verebilmek ve bütçeyi hazırlayabilmek için ne kadar kapasiteye ihtiyaç duyulduğunu bilmek ister. Aynı şekilde satın alma bölümü, kendi planlarını yapabilmek için tedarikçilerden ve tedarik zinciri ortaklarından hangi miktarlarda sipariş vereceğinin bilgisine ihtiyaç duyar (Stevenson, 2009: 72).

Üretim ve tesis kapasite planlama ve koordinasyonu, doğru tahminlere gereksinim duyar. Doğru tahmin ve sonuçlar işletmeye kapasite ya da stokla ilgili pahalı değişikliklerle tepki vermek yerine, proaktif bir şekilde davranarak kaynakların verimli bir şekilde kullanılmasını sağlar (Bowersox & Closs, 1996: 223).

Tahminler bir organizasyon içerisinde operasyondaki karar ve faaliyetleri doğrudan etkilediği gibi; muhasebe, finans, insan kaynakları, pazarlama ve yönetim bilgi sistemleri bölümlerindeki karar ve faaliyetleri de doğrudan etkilemektedir. Bir bölüm için alınan kararlar diğer bölümleri de ilgilendirmekte, bu sebeple etkilenen bölümler için genel bir tahmin politikası izlenmelidir (Stevenson, 2009: 72).

Tahmin problemleri farklı boyutlara göre sınıflandırılabilir. En yaygın olarak yapılan sınıflandırma zaman dilimine göre sınıflandırılmasıdır. Tahmin problemleri zaman dilimine göre; kısa dönemli, orta dönemli ve uzun dönemli tahminler olarak ayrılmaktadır. Kısa dönemli tahminler, günlük ve haftalık planları içermekte olup envanter yönetimi ve üretim planları için gereklidir. Orta dönemli tahminler, haftalık ve aylık planlar şeklinde olup ürün gruplarının satış paternlerini, çalışan gereksinim ve bulunabilirliklerini ve kaynak gereksinimlerini içermektedir. Uzun dönemli tahminler ise, aylık ve yıllık planlar olup bir işletmenin üretim stratejisinin tamamını etkilemektedir. Örnek vermek gerekirse, bir işletme eğer ileriye dönük taleplerde yüksek oranda bir artış olacağını hesaplıyorsa mevcut tesislerin yeni teknolojiler kullanılarak yeni tesislere dönüştürmesi gerekir. Bu karar kapasite ihtiyaçlarının belirlenmesinde verilmiş uzun dönemli bir planlamadır (Nahmias, 2013: 55).

1.1.1 Tahmin Sistemindeki Yedi Adım

Tahmin sistemi aşağıdaki yedi adımı izlemektedir. Bu yedi adım bir tahmin sisteminin başlatma, tasarım ve uygulamanın sistematik bir yolla ifade edilmesidir (Heizer & Render, 2014: 142-143).

- 1) Tahminin kullanım alanının belirlenmesi
- 2) Tahmin yapılacak ürünlerin seçilmesi
- 3) Tahmin yapılacak zaman diliminin belirlenmesi
- 4) Tahmin model ya da modellerinin seçilmesi
- 5) Tahmin yapabilmek için gerekli olan verinin toplanması
- 6) Tahminin yapılması
- 7) Tahmin sonuçlarının sağlanması ve geçerliliğinin denetlenmesi

1.1.2 Tahminin Karakteristik Özellikleri

Tahminler genellikle her zaman yanlıştır. İyi bir tahmin bile beklenen belli bir hata payı içermektedir. Toplanmış birimlerle tahmin yapmak, birbirinden ayrı tek tek birimlerin tahmininden daha doğru sonuçlar verir. Uzak zaman dilimi için yapılan

tahminler daha fazla hatalı sonuç alınmasına sebep olur. Tahmin yöntemi uygulanırken bilinen bilgiler dışarıda tutulmamalı, her bilgi dahil edilmelidir (Nahmias, 2013: 56).

1.1.3 İyi Bir Tahminin Unsurları

Doğru bir tahminin yapılabilmesi için aşağıdaki şartların mutlaka yerine getirilmesi gerekir (Stevenson, 2009: 74).

- Tahmin zamanında yapılmalı, tahmini içeren bilgiye cevap verecek belli bir zaman dilimini kapsamalıdır.
- Tahmin kesin, hatasız olmalıdır.
- Tahmin güvenilir olmalıdır.
- Tahmin anlamlı birimlerle ifade edilebilir olmalıdır.
- Tahmin yazılı olmalıdır.
- Tahmin yönteminin anlaşılması ve kullanılması kolay olmalıdır.
- Tahmin düşük maliyetli olmalıdır.

1.1.4 Tahmin Yöntemleri

Tahmin problemlerinin çözümü için iki genel yaklaşım bulunmaktadır. Bunlar; kantitatif ve kalitatif yaklaşımlardır. Kantitatif tahminler, matematiksel modellerin bir çeşidi olup talebi tahmin edebilmek için geçmiş verileri ve ilgili değişkenleri esas almaktadır. Subjektif ya da kalitatif tahminler ise, bir tahmin yapılırken karar vericinin sezgilerini, duygularını ve kişisel tecrübeleri gibi faktörleri de sürece dahil eder. Her iki yaklaşım da ayrı ayrı kullanıldığı gibi her iki yaklaşımın birlikte kullanılması da mümkündür ve genellikle en verimli sonuçları verir (Heizer & Render, 2014: 143).

Bir yöntemin uygulanabilirliğini anlayabilmek için; tahmin doğruluğu, tahmin zaman dilimi, tahminin değeri, verinin bulunabilirliği, verinin şekli ve tahmini yapacak kişinin deneyimleri gibi kriterlerin değerlendirilmesi gerekir. Her bir tahmin yöntemi hem kantitatif hem de kalitatif olarak bu altı kritere göre değerlendirilmelidir (Bowersox vd., 2002: 264).

1.1.4.1 Kalitatif Yöntemler

Kalitatif yöntemler çalışanların deneyimlerini ve anketleri esas alan, bununla birlikte basit matematik araçları da kullanarak farklı tahmin yöntemleriyle de kombine edilebilen yöntemlerdir. Kalitatif yöntemler genellikle geçmiş verilerin yetersiz olması durumunda uzun ve orta dönemli tahminlerde kullanılırlar. Yeni bir ürün ya da hizmetin piyasaya sürülmesi, bir ürünün paketinin değişmesi ya da gelecek talep şeklinin politik ya da teknolojik nedenlerden dolayı etkilenmesi durumları örnek verilebilir (Ghiani vd., 2004: 28).

En yaygın kullanılan kalitatif yöntemler aşağıda yer almaktadır (Heizer & Render, 2014: 143-144).

Yönetici Düşünce Kurulu: Üst düzey uzmanlar ve yöneticilerin oluşturduğu bir grubun görüşleri doğrultusunda, istatistiksel modellerin de dahil edilmesiyle talep tahmini gerçekleştirilir.

Delphi Yöntemi: Delhi yönteminde; karar vericiler, personel ve yanıtlayıcılar olmak üzere üç tip katılımcı vardır. Karar vericiler, genellikle asıl tahmini yapacak 5 ya da 10 kişilik bir uzman grubundan oluşmaktadır. Çalışan personel ise anket ve anket sonuçlarını hazırlayarak, dağıtarak, toplayarak ve özetleyerek karar vericilere yardımcı olurlar. Yanıtlayıcılar, farklı yerleşim bölgelerinde bulunan ve kararlarına değer verilen insan gruplarından oluşur. Bu grup, tahmin yapmadan önce karar vericilere önemli verilerin teminini sağlarlar.

Satış Gücü Karması: Bu yöntemde her satışçı kendi satış bölgesinde yapılacak satışları tahmin eder. Bu tahminler daha sonra gözden geçirilerek gerçeğe uygun olup olmadığı değerlendirilir. Toplam talebe ulaşabilmek için her bölgenin talebi birleştirilir.

Müşteri Anketleri: Bu yöntemde mevcut ve potansiyel müşterilerden gelecek satın alma planlarıyla ilgili bilgiler istenir. Müşterilerden alınacak bu bilgiler sadece talep tahmininin yapılmasına değil aynı zamanda yeni ürünler için ürün tasarım ve planlamanın da geliştirilmesine yardımcı olur.

1.1.4.2 Kantitatif Yöntemler

Kantitatif yöntemler, geçmişe ait yeterli talep bilgisinin olduğu durumlarda kullanılan yöntemlerdir. Kantitatif yöntemler; nedensel yöntemler, zaman serileri yöntemleri ve yapay zeka yöntemleri olmak üzere üç grupta incelemektedir (Ghiani vd., 2004: 29).

Nedensel Yöntemler: Regresyon, ekonometrik modeller, girdi-çıktı modelleri, yaşam çevrimi analizleri ve bilgisayar simülasyon modellerini içeren bu yaklaşımda gelecek talebin, değişkenlerin mevcut ya da geçmiş verilerine bağlı olduğu hipotezi esas alınmaktadır.

Zaman Serileri Yöntemleri: Zaman serileri yöntemleri, geçmişe ait net ve değişmez satış verilerinin ve trendin olduğu durumlarda kullanılan istatistiksel yöntemlerdir (Bowersox vd., 2002: 265). Geleceğe yönelik tahmin yapılırken geçmişe ait verilerin değerlendirilmesinde, geçmişe ait zaman serilerinin davranışları ayrı değerlendirilmelidir. Zaman serilerini şekillendiren bileşenler; trend, mevsimsellik, çevrim ve rastlantısallıktır. Box-Jenkins yöntemi, hareketli ortalamalar, basit üstel düzeltme ve trend analizi zaman serileri yöntemlerindedir.

Yapay Zeka Yöntemleri: İnsan beyninin özelliği olan öğrenme vasıtasıyla yeni bilgiler üretebilme becerisini otomatik bilgisayarlarla gerçekleştiren yöntemlerdir. Örnek olarak genetik algoritmalar, destek vektör makineleri ve yapay sinir ağları gösterilebilir. (Öztemel, 2012: 29)

1.2 Talep Şekillerinin Sınıflandırılması

Belirli bir ürün için kullanılan talep tahmin yönteminin doğruluğu, o ürüne ait geçmiş talep verilerinin özelliklerinin bilinmesine bağlıdır. Bu sebeple talep zaman serileri, en iyi talep tahmin yönteminin uygulanması için birkaç farklı sınıfa ayrılırlar (Kostenko & Hyndman, 2006: 1256).

İlk olarak Williams (1984) talep şekillerinin sınıflandırılmasıyla ilgili olarak varyans bölümlene fikrine dayalı bir metod geliştirmiştir. Bu metotta tedarik zamanı süresince talebin varyansı, onun rastgele parçalarına ayrılmaktadır. Bu sınıflandırmanın amacı,

çıkan sınıflandırma sonuçlarına göre en uygun talep tahmin ve stok kontrol yöntemlerinin belirlenmesidir (Syntetos vd., 2005: 496).

Talebin varyansının ayrıldığı rastgele parçalar aşağıdaki şekildedir. Bu parçalar birbirinden bağımsız ve ratsgele olarak dağıldığı düşünülmektedir (Babiloni vd., 2010: 116).

1. Ardışık birimler halinde gelen siparişlerin sayısı, ortalama (n) ve varyans $var(n)$ ile
2. Siparişlerin büyüklüğü, ortalama (x) ve varyans $var(x)$ ile
3. Tedarik süresi, ortalama (L) ve varyans $var(L)$ ile

Williams tedarik süresince gerçekleşen talebin varyansının (DDLT), 1.1 numaralı denklemdenki şekilde bölündüğünü öne sürmektedir. 1.2 numaralı denklemden ise değişkenlik / varyasyon (CV) katsayısı yer almaktadır (Babiloni vd., 2010: 117).

$$var(DDLT) = x^2 L var(n) + n L var(x) + n^2 x^2 var(L) \quad (1.1)$$

$$CV^2_{DDLT} = \frac{CV^2_n}{L} + \frac{CV^2_x}{nL} + CV^2_L \quad (1.2)$$

Williams (1984) tarafından farklı sınıftaki ürünlerin stok kontrol politikalarına ve talep tahmin yöntemlerine göre talep şekillerinin sınıflandırılması üzerine yapılan çalışma Eaves ve Kingsman (2004) tarafından revize edilerek geliştirilmiştir (Kostenko & Hyndman, 2006: 1256).

Tablo 1’de hem Williams hem de Eaves ve Kingsman tarafından yapılan sınıflandırma yer almaktadır.

Bir diğer yaklaşım ise Syntetos ve diğerleri tarafından (2005) tarafından geliştirilmiştir. Yapmış oldukları sınıflandırmada belli varsayımlar altında basit üstel düzeltme ve üstel ağırlıklandırılmalı hareketli ortalama tahmin yöntemlerinin hata karelerinin ortalamalarını (MSE) esas almışlardır. Croston (1972) yöntemi ve Syntetos & Boylan (1999) tarafından geliştirilen ve yakınlaştırma yöntemi adı verilen yöntemlerle karşılaştırma yapmışlardır.

Tablo 1 - Varyans Bölümleme Fikrine Dayalı Olarak Talebin Sınıflandırılması

Kaynak: Babiloni, E., Cardos, M., Albarracin, J. M. ve Palmer, M. E., “Demand Categorisation, Forecasting And Inventory Control For Intermittent Demand”, **South African Journal of Industrial Engineering**, C: 21, No: 2, 2010, p.117

$CV^2_{DDL T}$			Talebin Sınıflandırılması	
$\frac{CV^2_n}{L}$	$\frac{CV^2_x}{nL}$	CV^2_L	Williams (1984)	Eaves ve Kingsman (2004)
Düşük	Düşük		Düzensiz	Düzensiz
Düşük	Yüksek		Düzensiz	Düzensiz olmayan
Yüksek	Düşük		Yavaş hareket eden	Yavaş hareket eden
Yüksek	Yüksek	Düşük	Aralıklı	Orta aralıklı
Yüksek	Yüksek	Yüksek	Aralıklı	Yüksek aralıklı

Syntetos ve diğerleri (2005) talep şekillerini; değişken talep, düzensiz talep, düzensiz olmayan talep ve aralıklı talep olmak üzere dört kategoride sınıflandırmışlardır. Bu sınıflandırma iki parametre esas alınarak yapılmıştır. Bu parametreler; p (talebin sıklığı) ve CV^2 (talep büyüklüğünün değişkenlik / varyasyon katsayısının karesi) parametreleridir (Kostenko & Hyndman, 2006: 1256).

Yapılan bu sınıflandırmanın tanımları aşağıda yer almaktadır (Ghobbar & Friend, 2002: 227) :

Değişken Talep: Büyük ölçüde tutarsız gerçekleşen, tutarsızlığın birim zamanda gerçekleşen talepten daha çok talep büyüklüğüne bağlı olan talep.

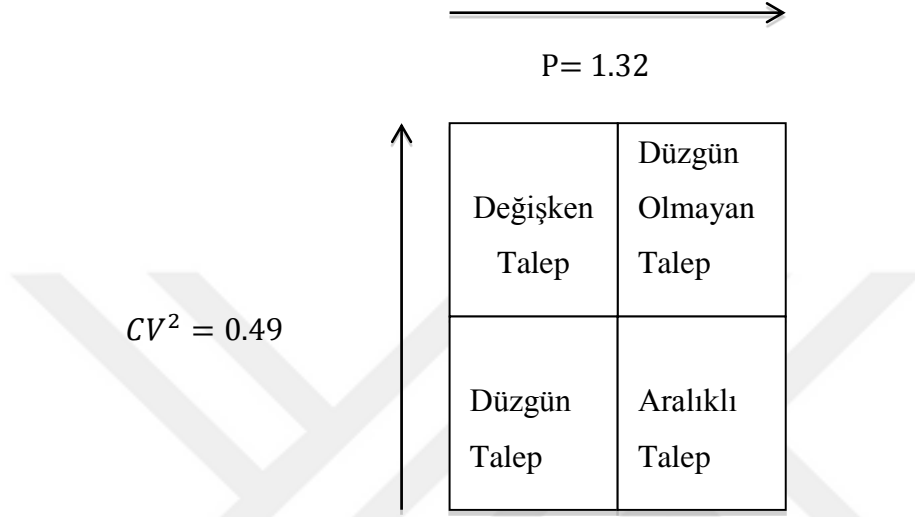
Düzensiz Talep: Rastgele gerçekleşen ve belli zaman aralıklarında talebin sıfır olduğu taleptir. Talep küçük miktarlarda gerçekleşir.

Düzensiz Olmayan Talep: Rastgele gerçekleşen ve çoğu zaman diliminde talep miktarının sıfır olduğu taleptir. Talep gerçekleştiğinde ise yüksek oranda değişkenlik söz konusudur.

Aralıklı Talep: Rastgele gerçekleşen ve çoğu zaman diliminde talep miktarının sıfır olduğu taleptir. Talep gerçekleştiğinde ise yüksek oranda değişkenlik söz konusu değildir.

Şekil 1 - Talep Şekillerinin Sınıflandırılması

Kaynak: Syntetos, A. A. Boylan, J. E. ve Croston, J. D., "On The Categorization Of Demand Patterns", *Journal of the Operational Research Society*, C: 56, No: 5, 2005, p. 500



Syntetos ve diğerleri tarafından (2005) tarafından geliştirilen sınıflandırmada eşik değerler $p = 1.32$ ve $CV^2 = 0.49$ olarak belirlenmiştir.

$P \leq x, CV^2 \leq y$ durumu, talebi çok aralıklı ve değişken olmayan stok birimleri için ölçülmüştür. (örnek olarak hızlı hareket eden parçalar)

$P > x, CV^2 \leq y$ durumu, yavaş hareket eden ya da sürekli/genellikle aralıklı talep yapısına sahip ve talep büyüklüklerinde yüksek değişikliklerin olmadığı stok birimleri için ölçülmüştür.

$P > x, CV^2 > y$ durumu, talebi düzenli olmayan stok birimleri için ölçülmüştür.

$P \leq x, CV^2 > y$ durumu, talebi değişken fakat çok aralıklı olmayan stok birimleri için ölçülmüştür.

Tüm bu durumlar için x eşik değeri $p = 1.32$ olarak belirlenen talepler arasındaki sürenin ortalamasını, y ise eşik değeri $CV^2 = 0.49$ olarak belirlenen talep büyüklüğünün değişkenlik / varyasyon katsayısının karesini ifade etmektedir (Ghobbar & Friend, 2003: 2105).

1.3 Aralıklı Talebin Tanımı

Aralıklı talep literatürde tahmin edilmesi zor olan talep senaryolarından biri olarak görülmektedir (Varghese & Rossetti, 2008: 864). Yavaş hareket eden özelliklerinden dolayı aralıklı talebe sahip ürünler tükenme riskiyle karşı karşıyadırlar. Bu durum bir organizasyonun operasyonları üzerinde önemli bir etkiye neden olmaktadır (Kourentzes, 2013: 198).

Croston (1972) aralıklı talep zaman serilerinin klasik zaman serilerinden farklılık gösteren ve birden çok zaman diliminde talebin sıfır olduğu durumdur şeklinde tanımlamıştır.

Johnston & Boylan (1996) sıfırdan farklı olarak gerçekleşen talebin zaman aralıklarının ortalamasının, envanter gözden geçirme periyodundan 1.25 kat daha büyük olması durumunda talebin aralıklı olduğunu ortaya koymuştur.

Silver ve diğerleri (1998) aralıklı talebi, seyrek olarak gerçekleşen ardışık işlemlerin arasındaki ortalama zaman birim zaman periyodunun sonraki tahmin güncelleme aralığından ciddi oranda fazladır şeklinde tanımlamıştır.

Syntetos & Boylan (2001) aralıklı talep için, rastgele gerçekleşen ve çoğu zaman diliminde talebin sıfır olduğu taleptir şeklinde tanımlamıştır. Talep gerçekleştiğinde ise talep miktarı arada sırada bir birimden fazladır. Aralıklı talep üretim ve tedarik süreçlerinde talep tahmini ve stok kontrolle ilgili olarak önemli problemler yaratmaktadır. Talep büyüklüğünün değişkenliği kadar talep şeklinin değişkenliği de aralıklı talebin tahmin edilmesini zorlaştırmaktadır.

Smart (2003) bir talep serisinin en az %30'unda talebin sıfır olmasını, US donanması stoktan sorumlu yöneticiler ise talep serisinin %60-70 oranına eşit ya da daha az olarak talebin sıfırdan farklı olmasını aralıklı talep olarak tanımlamaktadırlar.

Johnston ve diğerleri (2003) aralıklı talebe sahip ürünlerini toplam stok değerinin %60'ını oluşturduğunu ve genellikle havacılık, otomotiv, askerlik ve bilgi işlem sektöründe rastlandığını öne sürmüşlerdir.

Willemain ve diğeri (2004) ağır makineler, yedek parçalar, uçak servis parçaları elektronik ve denizcilik yedek parçalarıyla aralıklı taleple ilgili olarak çalışmalar yapmış yine Syntetos & Boylan (2005) otomotiv yedek parçalardaki talep aralıklılığı üzerine çalışmıştır. Ghobbar & Friend (2002) talebi aralıklı gerçekleşen yüksek değerli uçak bakım parçaları ile ilgili çalışmıştır.

1.4 Aralıklı Talep Tahmininde Kullanılan Yöntemler

Aralıklı talebe sahip ürünlerin talep tahmininde pratikte genellikle basit üstel düzeltme ve hareketli ortalama gibi tahmin yöntemleri kullanılmaktadır. Fakat bu basit klasik talep tahmin yöntemleri, çoğu zaman diliminde talebin sıfır olmasından dolayı aralıklı talep senaryolarında elverişsiz kalmaktadır (Varghese & Rossetti, 2008,: 865).

Bununla birlikte, aralıklı talebe sahip ürünlerin talep tahmininde kullanılan herkesçe kabul edilen bir diğer yöntem de Croston (1972) tarafından geliştirilmiş ve literatürde Croston yöntemi olarak yer almaktadır.

Syntetos & Boylan (2001) ve daha sonra Leven & Segerstedt (2004) Croston metodunda bir takım düzeltmeler yaparak Croston metodunun farklı varyasyonlarını ortaya koymuşlardır. Aralıklı talebe sahip ürünlerin talep tahmininde destek vektör makinesi ve yapay sinir ağları gibi yeni yöntemlerin de uygulandığı çalışmalar literatürde yer almaktadır.

1.4.1 Basit Üstel Düzeltme Yöntemi (SES)

Basit üstel düzeltme yöntemi kısa dönemli talep tahmininde en yaygın olarak kullanılan istatistiksel yöntemlerden biridir. Bu yöntem uygulaması ve anlaşılması kolay olan, çok yönlü ağırlıklandırılmış ortalama yöntemidir (Silver vd., 1998: 89).

Her yeni tahmin, bir önceki tahmin ile tahminin ve gerçekleşen değer arasındaki farkın yüzdesinin toplamına eşittir. Aşağıdaki şekilde göstermek mümkündür (Stevenson, 2009: 83-84) :

Bir sonraki tahmin = Bir önceki tahmin + α (Gerçekleşen – Bir önceki tahmin) (1.3)

1.3 numaralı denklemde yer alan (Gerçekleşen – Bir önceki tahmin) bölümü tahmin hatasını temsil etmektedir. α değeri ise düzeltme katsayısıdır ve 0 ile 1 arasında bir değer almaktadır. Düzeltme katsayısı geçmiş verilerin etkisini düzene sokmaktadır. Düşük bir değere sahip olan düzeltme katsayısı geçmiş verilerin üzerinde durulmasını sağlayarak daha çok sabit talep için uygundur, fakat sistematik değişikliklere karşı hızlı tepki verilememesine neden olacaktır. Yüksek bir değere sahip olan düzeltme katsayısı ise en son yapılan gözlemlerin üzerinde durulmasını sağlayarak talep değiştiğinde hızlı tepki verilmesi gereken durumlar için uygundur (Wallström & Segerstedt, 2010: 626).

Özetlenecek olursa:

$$F_t = F_{t-1} + \alpha (D_{t-1} - F_{t-1}) \quad (1.4)$$

F_t = t dönemi için tahmin

F_{t-1} = bir önceki dönem için tahmin (t-1 dönemi için)

α = düzeltme faktörü

D_{t-1} = bir önceki dönem için gerçekleşen talep

Yüksek hacme sahip ürünlerin talep tahmininde basit üstel düzeltme yöntemi başarılı sonuçlar verirken düşük hacme sahip, aralıklı talebin söz konusu olduğu durumlarda bu yöntem başarılı sonuçlar vermemektedir (Snyder vd., 2012: 486).

1.4.2 Hareketli Ortalamalar Yöntemi

Basit ve yaygın olarak kullanılan yöntemlerden biri de hareketli ortalamalar yöntemidir. Bu yöntemde n kadar dönemin basit aritmetik ortalaması alınarak hesaplama yapılır (Nahmias, 2013: 64). Hareketli ortalamalar yöntemiyle tahmin yapılırken geçmişte gerçekleşen gerçek değerler kullanılır ve aşağıdaki denklemle hesaplama yapılır (Stevenson, 2009: 81).

$$F_t = MA_n = \left(\frac{1}{n}\right) \sum_{i=1}^n A_{t-i} = \frac{A_{t-n} + \dots + A_{t-2} + A_{t-1}}{n} \quad (1.5)$$

F_t = t dönemi için tahmin

MA_n = n dönemin hareketli ortalaması

A_{t-1} = t-1 dönemi için gerçekleşen talep

n = hareketli ortalama kullanılan dönem sayısı

Basit üstel düzeltme yöntemi de hareketli ortalamalar yöntemi de aralıklı talebin doğasında yer alan talep büyüklüğü ve talep olasılığı gibi iki faktörü göz ardı ettiğinden dolayı bu talep türünün tahmininde yeterli olmamaktadır (Teunter vd., 2011: 606).

1.4.3 Croston Yöntemi

Aralıklı talep yapısına sahip ürünlerin talep tahmini için kullanılan klasik yöntem Croston (1972) tarafından geliştirilmiş ve Rao (1973) tarafından cebirsel bir yöntemle düzeltilmiştir. Croston yöntemi iki farklı üstel düzeltme tahminini kullanmaktadır. Bunlardan biri talep büyüklüğü diğeri ise talebin sıfır olmadığı durumlar arasındaki zaman aralıklarıdır. Croston talebin normal dağıldığını, talepler arasında gerçekleşen sürenin ise Bernoulli dağılımına uygun olarak dağıldığını varsaymaktadır ve iki süreç de birbirinden bağımsızdır.

Tahminler sadece talep meydana geldiğinde güncellenir. Bundan dolayı Croston yöntemi talebin her dönemde gerçekleşmesi durumunda klasik üstel düzeltme yöntemiyle aynıdır. Croston yönteminde varyans basit üstel düzeltme yöntemindeki varyanstan daha düşüktür (Babiloni vd., 2010: 120).

Croston tarafından aralıklı talep yapısına sahip ürünler için geliştirilen yöntemin algoritması aşağıdaki şekildedir (Croston, 1972: 302) :

$$y_{t=x_t} (\bar{Z}_{\eta-1} + e_{\eta}) \quad (1.6)$$

t = zaman aralıklarının gözden geçirilmesi

η = talebin sıfır olmadığı dönemlerin sayıları

$x_{t=}$ Bernoulli dağılımına göre talebin gerçekleşme olasılığı ($1/p$)

$y_{t=}$ t dönemi için tahmin

$Z_{\eta=}$ ortalama (μ) ve standart sapma (σ) ile talep miktarının sıfır olmayan gözlemleri

$e_{\eta=}$ tahmincinin hatası

p = gelişler arasındaki sürenin ortalaması

q = talepler arasındaki süre (p ortalamayla geometrik dağılıma uygun dağılır.)

Bernoulli dağılımına göre:

$$x_{t=} \begin{cases} 1, & \text{olasılık } (1/p) \\ 0, & \text{olasılık } (1 - 1/p) \end{cases} \quad (1.7)$$

$$E[(x_{i-1/p})(x_j - 1/p)] = 0 \quad \text{her } i, j, \quad i \neq j \quad (1.8)$$

Z_n sürecin talep miktarının sıfır olmayan gözlemleridir ve üstel ağırlıklı hareketli ortalama (EWMA) ile dağıldığı varsayılır.

$$\bar{Z}_{\eta=} \bar{Z}_{\eta-1}(\lambda) + e_{\eta} \quad (1.9)$$

$$\bar{Z}_{\eta-1}(\lambda) = \lambda \sum_{j=1}^{\infty} (1 - \lambda)^{j-1} z_{\eta-j} \quad (1.10)$$

x_t ve z_{η} birbirlerinden bağımsızdır ve λ geçmiş verilere göre tahmin edilir. e_{η} bağımsızdır ve sıfır ortalamayla normal dağıldığı varsayılır. Sürecin güncellenmiş hali 1.11 ve 1.12 numaralı denklemlerde yer almaktadır.

$$q = q + 1, \quad y_t = 0 \quad (1.11)$$

$$\begin{aligned}
e_\eta &= y_t - \bar{z}_{\eta-1} \\
\bar{z}_\eta &= \bar{z}_\eta - \alpha e_n \\
m_\eta &= (1 - \alpha) m_{\eta-1} + \alpha |e_\eta| \\
R_t &= \bar{z}_\eta + km_\eta \\
\bar{p}_\eta &= \bar{p}_{\eta-1} (1 - \alpha) + \alpha q \\
\bar{y}_t &= \bar{z}_\eta / \bar{p}_\eta \\
q &= 1,
\end{aligned}
\left. \vphantom{\begin{aligned} e_\eta \\ \bar{z}_\eta \\ m_\eta \\ R_t \\ \bar{p}_\eta \\ \bar{y}_t \\ q \end{aligned}} \right\} y_t \neq 0 \tag{1.12}$$

Croston yönteminin iki önemli dezavantajı vardır. Birincisi yöntemin pozitif eğilimli olması, diğeri ise talep miktarının sıfır olmasından sonra kendini güncelleyememesidir. Talebin sıfır olduğu zaman periyodlarından sonra tahminler geçersiz kalmakta ve stok tükenme riskiyle karşılaşılan durumlarda elverişli olmamaktadır. Stok tükenmesi özellikle yavaş hareket eden ve aralıklı talebe sahip olan ürünlerin stok yönetimi için önemli bir konudur. Yapılan çalışmalarda stokta uzun süredir bulunan ürünlerin büyük bir kısmında yine uzun bir süre hiç talep olmadığı görülmüştür (Teunter vd., 2011: 606).

1.4.4 Croston Yönteminin Modifikasyonları ve Karşılaştırmaları

Croston (1972) hem talep büyüklüğünü hem de talebin meydana gelme sıklığını göz önünde bulundurarak aralıklı talep yapısına sahip ürünler için bir yöntem geliştirmiştir. Croston yönteminin böyle bir tahmin süreci için teorik üstünlüğü olmasına rağmen diğer basit talep tahmin yöntemleriyle kıyaslandığında gözlemsel kanıtlar, bu yöntemin biraz daha iyi sonuçlar verdiğini fakat bazı gözlemlerde başarısız kaldığını göstermektedir.

Willemain ve diğerleri (1994) Croston yöntemi ve basit üstel düzeltme yöntemini karşılaştırmak için hem yapay verileri hem de gerçek talep verileri kullanarak bir çalışma yapmıştır. Yaptıkları bu çalışmada Croston yönteminin basit üstel düzeltme yöntemine göre daha iyi sonuçlar verdiğini ispatlamışlardır (Wallström & Segerstedt, 2010: 627).

Syntetos ve Boylan (2001) Croston yöntemini geliştirebilmek amacıyla yöntem üzerinde bir çalışma yapmış ve talebin beklenen tahminindeki matematiksel türevinde bir hata bulmuşlardır (Syntetos & Boylan, 2001: 457-458).

Croston yönteminde yer alan yanılıgyı düzeltmek amacıyla Syntetos ve Boylan (2005) Croston yöntemine 1.13 numaralı denklemi yönteme eklemiştirler. Bu yöntem “Yakınlık yöntemi” olarak da bilinmektedir. 1.13 numaralı denklemde yer alan α , \bar{p}_η için düzgünleştirme parametresidir (Babiloni vd., 2010: 122).

$$\bar{y}_t = \left(1 - \frac{\alpha}{2}\right) \frac{\bar{z}_\eta}{\bar{p}_\eta} \quad (1.13)$$

Ghobbar ve Friend (2003) aralıklı talebe sahip uçak yedek parçaları için 13 talep tahmin yöntemini karşılaştırmış ve ağırlıklı hareketli ortalama, Holt üstel düzeltme ve Croston yöntemlerinin diğer yöntemlere göre daha iyi sonuç verdiğini ispatlamışlardır. Benzer bir çalışma da Regattieri ve diğerleri (2005) tarafından yapılmıştır. 20 talep tahmin yöntemini karşılaştırmışlar ve aynı sonuca ulaşmışlardır.

Eaves ve Kingsman (2004) Kraliyet hava kuvvetleri vaka çalışmasında yedek parçalar üzerine bir çalışma yapmış ve basit üstel düzeltme, Croston (1972), Syntetos ve Boylan (2005) tarafından geliştirilen yöntemleri karşılaştırarak Syntetos ve Boylan (2005) tarafından geliştirilen yöntemin daha üstün olduğunu ispatlamıştır.

Leven ve Segerstedt (2004) de Croston yöntemi üzerinde bir düzeltme yaparak her bir dönemdeki ortalama talep için 1.14 numaralı denklemi yönteme eklemiştirler (Babiloni vd., 2010: 123).

$$\bar{y}_t = \alpha \left(\frac{\bar{z}_\eta}{\bar{p}_\eta}\right) + (1 - \alpha) \bar{y}_{t-1} \quad (1.14)$$

Shenstone ve Hyndman (2005) Croston yönteminin temelinde stokastik model olan, tamamen formüle edilmemiş, özel ve geçici olarak oluşturulmuş bir yöntem olduğunu vurgulamışlardır. Aralıklı talebin yapısının özelliklerinden dolayı Croston yönteminin tutarsız olduğunu fakat pratik işe yararlılığından dolayı deneysel olarak diğer klasik yöntemlere göre daha iyi sonuçlar verdiğini belirtmişlerdir.

Shale ve diğerkleri (2006) sipariřlerin poisson dađılımına gre gelmesi durumunda uygun modifikasyonun 1.15 numaralı denklemdede yer aldıđı řekilde olması gerektiđini ne srmüşlerdir (Kourentzes, 2013: 199). Talebin poisson dađılımına gre gelmesi durumunda talepler arasındaki süre negatif üstel dađılıma uygun řekilde dađılacaktır (Babiloni vd., 2010: 123).

$$\bar{y}_t = \left(1 - \frac{\alpha}{2-\alpha}\right) \frac{\bar{z}_\eta}{\bar{p}_\eta} \quad (1.15)$$

Boylan ve Syntetos (2007) , Leven ve Segerstedt (2004) tarafından geliştirilen yöntemin Croston (1972) yöntemine gre özellikle yüksek derecede aralıklı talebin olduđu durumlarda daha kötü sonuç verdiđini ispatlamışlardır (Kourentzes, 2013: 198).

Altay ve diğerkleri (2008) aralıklı talep yapısına sahip ürünlerin talep tahmini için Croston yönteminin son modifikasyonuyla Wright (1986) tarafından deđiřtirilmiş Holt üstel düzeltme yöntemlerini karşılařtırmıştır. İşletmenin önceliđinin stok seviyelerini düşürmek olması durumunda deđiřikliğe uğratılmış Croston yönteminin, müşteri memnuniyetini daha öncelikli tutması durumunda ise deđiřikliğe uğratılmış Holt yönteminin daha üstün olduđunu belirtmişlerdir.

Teunter ve Sani (2009); Croston yöntemi, Syntetos tarafından geliştirilen fakat yayınlanmayan yöntem, Leven & Segerstedt (2004) ve Syntetos v& Boylan (2005) tarafından geliştirilen yöntemleri karşılařtırarak bir alıřma yapmıştır. Leven ve Segerstedt (2004) tarafından geliştirilen yöntemin diđer tahmin yöntemlerine gre aşırı tahminde bulunarak kötü sonuçlar verdiđini ispatlamıştır. En az sapmanın yařandıđı yöntem ise Syntetos tarafından geliştirilen yöntemdir. Fakat bu yöntemdeki eksiklik, varyans başarısının Syntetos ve Boylan (2005) tarafından geliştirilen yöntem kadar başarılı olmamasıdır (Wallström & Segerstedt, 2010: 627).

Wallström ve Segerstedt (2010); basit üstel düzeltme yöntemi, Croston yöntemi, Leven & Segerstedt (2004) ve Syntetos & Boylan (2005) tarafından geliştirilen yöntemleri karşılařtırarak yaptıđı alıřmada Leven ve Segerstedt (2004) tarafından geliştirilen yöntemin en kötü sonuçları verdiđini ve en az uygun olan yöntem olduđunu

ispatlamışlardır. Standart ölçütlere göre yapılan karşılaştırma sonuçları Tablo 2’de yer almaktadır (Wallström & Segerstedt, 2010: 635).

Tablo 2 - Standart Ölçülere Göre En İyi Tahmin Yöntemi

Kaynak: Wallström, P. ve Segerstedt, A., “Evaluation Of Forecasting Error Measurements And Techniques For Intermittent Demand”, **Int. J. Production Economics**, C: 128, 2010, p.635

Ölçüt	En İyi Yöntem
MSE (ortalama hataların karesi)	Croston esaslı S&B yöntemi / SES
MAD (ortalama mutlak sapma)	Croston esaslı S&B yöntemi
sMAPE (simetrik ortalama mutlak yüzde hata)	Croston esaslı L&S yöntemi
CFE (Kümülatif tahmin hatası)	SES

Teunter ve diğerleri (2011) Croston yönteminin \bar{z}_η ve \bar{p}_η için sınırlı güncelleme olması sebebiyle stok tükenme sorununda uygun olmadığını ispatlamışlardır ve Croston yöntemine benzer bir yöntem geliştirmişlerdir. Bu yöntem doğrudan aralıklı zaman serilerini modellememektedir.

Yöntemde iki vektör göz önünde bulundurulmuştur: talebin gerçekleşme olasılığı (d_η) ve talep büyüklüğüdür (z_η). Talebin η zamanda gerçekleşme olasılığı 0 ya da 1’dir ve basit üstel düzeltme yöntemi kullanılarak modellendirilmiştir ve sadece pozitif talep gerçekleştiğinde güncelleme gerçekleşmektedir. Her düzgünleştirme modeli farklı parametreler kullanmaktadır (Kourentzes, 2014: 182).

Teunter ve diğerleri (2011) talebin gerçekleşme olasılığının düzgünleştirme parametresinin talebin büyüklüğünün düzgünleştirme parametresinden küçük olması gerektiğini savunmaktadırlar. Bu yöntem 1.16 numaralı denklemde yer almaktadır.

$$\bar{y}_t = \bar{d}_\eta \bar{z}_\eta \quad (1.16)$$

Kourentzes (2014) aralıklı talep tahmin modellerinin optimizasyonu ve seçimi üzerine bir çalışma yapmış ve model seçimi üzerine açık bir öneri sunamamıştır. Literatürde yer alan yeni yöntemler Croston yöntemi ve türevlerinden daha iyi sonuç vermekte olduğunu, güçlü ve geçerli bir model seçimi metodolojisinin gerekli olduğunu ortaya koymuştur.

1.4.5 Aralıklı Talep Tahmininde Kullanılan Diğer Yöntem ve Algoritmalar

Willemain ve diğerleri (2004) yedek parçalar üzerine bir çalışma yaparak sabit tedarik süresi üzerine aralıklı talebin kümülatif dağılımının tahmini için patentli bir algoritma geliştirmişlerdir. Geliştirdikleri yöntemde Efron (1979) tarafından ortaya çıkan özyükleme (bootstrap) yöntemini aralıklı talep verilerinin tahmini için ve olasılık tümlev dönüşümünü aralıklı talep verilerine uygun tahmin doğruluğu ölçütünü bulmak için uyarlamışlardır. Willemain ve diğerleri (2004) tarafından geliştirilen yöntem parametrik olmayan bir özyükleme (bootstrap) yaklaşımıdır.

Özyükleme (bootstrap) yöntemi; kısıtlı varsayımların bulunduğu, sık kullanılan istatistiksel yöntemlerin uygulanmasının elverişsiz olduğu durumlarda kullanılan, karmaşık matematik formüllerin yer almadığı ve uygulanması kolay olan bir yöntemdir (Takma & Atıl, 2006: 64).

Yaptıkları çalışmada talep miktarlarının sıfır olduğu ya da sıfırdan farklı olduğu değerler arasında pozitif korelasyon olduğunu varsaymışlardır. Geliştirdikleri bu yöntem basit üstel düzeltme ve Croston yöntemlerine göre, özellikle geçmiş verilerin daha az olduğu durumlarda daha başarılı sonuçlar vermiştir (Willemain vd., 2004: 376).

Synder (2002) Croston yönteminin uyarlamalarını talep tahmini ve stok kontrolüyle entegre edilmiş parametrik bir özyükleme (bootstrap) yaklaşımıyla karşılaştırmıştır.

Hua ve diğerleri (2007) de yedek parça talebinin tahmini üzerine talep miktarının sıfır olduğu ve talep miktarının sıfır olmadığı durumlar arasındaki ilişkiyi dikkate alarak özyükleme (bootstrap) yöntemi ve regresyon analizini birlikte kullanarak petrokimya

sektörü için benzer bir çalışma yapmışlardır. Kendi yöntemlerini Croston yöntemi, basit üstel düzeltme ve Willemain (2004) tarafından özyükleme (bootstrap) yöntemleriyle karşılaştırmışlardır. Karşılaştırma sonunda, talep miktarının sıfır olduğu durumlardaki ortalama hatalarının istatistiksel sonuçlarına bakıldığında kendi geliştirdikleri yöntemin daha başarılı sonuçlar verdiğini görmüşlerdir.

Bao ve diğerleri (2004) aralıklı talep yapısına sahip ürünlerin tahmini için destek vektör makinesini ve regresyon analizini kullanarak bir çalışma yapmışlardır.

Hua ve Zhang (2006) aralıklı talep yapısına sahip yedek parçalar için destek vektör makinesi ve lojistik regresyonunu kullanarak hibrid bir yaklaşım geliştirmişlerdir ve diğer yöntemlere göre daha başarılı olduğunu ispatlamışlardır.

Gutierrez ve diğerleri (2008) yapay sinir ağlarını esas alarak basit üstel düzeltme, Croston (1972), Syntetos ve Boylan (2005) tarafından geliştirilen yöntemlerden daha iyi sonuç veren bir tahmin yöntemi geliştirmişlerdir. Yapay sinir ağlarını kullanarak benzer bir çalışma da Kourentzes (2013) tarafından yapılmıştır.

Nikolopoulos ve diğerleri (2010) ile Babai ve diğerleri (2012) ise bu alanda henüz yaygın olarak kullanılmayan küme temelli yöntemleri kullanarak çalışmalar yapmışlardır.

1.5 Tahmin Doğruluğunun Ölçümü

Tahminlerin doğruluğu ve kontrol edilmesi talep tahmini süreci için oldukça önemlidir. Bu sebeple tahmini yapacak kişiler tahmin hatalarının olabildiğince minimum seviyede olmasını isterler. Gerçek dünya değişkenlerinin karmaşık tabiatı gereği gelecek değerleri düzgün bir şekilde tahmin etmek neredeyse imkansızdır. Tüm değişkenler dikkatte alınsa bile rastgele gerçekleşen değişiklikler söz konusu olabilir.

Tahmin hataları talep tahmini sürecindeki kararları iki şekilde etkiler. Bunlardan biri çeşitli talep tahmin yöntemleri arasında yapılacak seçim, diğeri ise kullanılan talep tahmin yönteminin başarı ya da başarısızlığının ölçülmesidir (Stevenson, 2009: 75).

Tahmin hatası, belirli bir zaman dilimi için tahmin edilen değer ile gerçekleşen değer arasındaki farktır. Tahmin doğruluğunun saptanabilmesi için bir takım ölçütler kullanılmaktadır (Nahmias, 2013,:61).

1.5.1 Ortalama Hataların Karesi (MSE)

Tahmin hatalarının belirlenmesinde en yaygın olarak kullanılan ölçüt Ortalama Hataların Karesi (MSE)' dir. Ortalama Hataların Karesi (MSE) aynı veri setine uygulanan farklı yöntemleri karşılaştırmak için oldukça kullanışlıdır fakat farklı ölçeklere sahip veri setlerinin karşılaştırılmasında kullanılması tavsiye edilmez (Hyndman & Koehler, 2006: 679). Ortalama Hataların Karesi (MSE) aşağıdaki şekilde hesaplanmaktadır (Nahmias, 2013: 61).

t dönemi için tahmin hatası e_t , t dönemi için tahmin değeri F_t ve t dönemi için gerçekleşen değer D_t olması durumunda tahmin hatasının denklemi 1.17'de yer almaktadır.

$$e_t = D_t - F_t \quad (1.17)$$

n dönemde gerçekleşen tahmin hataları e_1, e_2, \dots, e_n olması durumunda ortalama hataların karesi (MSE) :

$$MSE = (1/n) \sum_{i=1}^n e_i^2 \quad (1.18)$$

Temelini üstel düzeltme oluşturan tüm talep tahmin yöntemlerinde Ortalama Hataların Karesi yaygın olarak kullanılan optimizasyon ölçütlerindedir (Kourentzes, 2014: 183).

1.5.2 Ortalama Mutlak Sapma (MAD)

Tahmin hatalarının belirlenmesinde yaygın olarak kullanılan bir diğer ölçüt ise Ortalama Mutlak Sapma (MAD)' dir. MAD hesaplaması en kolay fakat hataları doğrusal olarak değerlendiren bir ölçüttür. Ortalama Hataların Karesi (MSE) ile

karşılaştırıldığında MAD tüm hataları tam olarak düz bir şekilde ölçerken, MSE hataları onların karelerine göre ölçer (Stevenson, 2009: 76).

Ortalama Mutlak Sapma (MAD) aşağıdaki şekilde hesaplanmaktadır (Nahmias, 2013: 61).

$$MAD = (1/n) \sum_{i=1}^n |e_i| \quad (1.19)$$

1.5.3 Ortalama Mutlak Yüzde Hata (MAPE)

Değişkenliğin bir diğer sezgisel ölçütü Ortalama Mutlak Yüzde Hata (MAPE)'dir. Yüzde olarak ifade edildiği için genellikle talep değerlerinin büyüklüğünden etkilenmez. Bununla birlikte, talep değerleri çok düşükse ve çoğu zaman periyodu sıfır değer alıyorsa hataların ölçülmesinde uygun bir ölçüt değildir (Silver vd., 1998: 110).

Aralıklı talep yapısına sahip ürünlerin talep tahmininde, bu tip zaman serileri çoğu dönemde sıfır değere sahip oldukları için serilerle bağıntılı ölçek bağımsız ölçütlerden biri olan Ortalama Mutlak Yüzde Hata (MAPE)'nin hesaplanması imkansızdır (Kourentzes, 2013: 202).

Ortalama Mutlak Yüzde Hata (MAPE) aşağıdaki şekilde hesaplanmaktadır (Nahmias, 2013: 61).

$$MAPE = \left[(1/n) \sum_{i=1}^n |e_i/D_i| \right] \times 100 \quad (1.20)$$

Gutierrez ve diğerleri (2008) düzensiz, aralıklı talebe sahip ürünlerin talep tahmininde çoğu zaman diliminde sıfır değer olmasından dolayı tahmin doğruluğunun ölçümünde geleneksel MAPE ölçütünün hatalı sonuçlar verdiğini gözlemlemişlerdir. Geleneksel MAPE ölçütünü kullanmak yerine alternatif olarak tanımlanan 1.21 numaralı denklemde yer alan MAPE ölçütünü kullanmışlardır.

$$MAPE = \frac{\sum_{i=1}^n |e_i|}{\sum_{i=1}^n |D_i|} \quad (1.21)$$

1.5.4 Ortalama Hata Kareleri Karekökü (RMSE)

Ortalama Hata Kareleri Karekökü (RMSE) basit bir şekilde ifade edilecek olursa Ortalama Hataların Karesinin (MSE) karekökünün alınmasıdır. Tahmin hatalarının büyüklüğünü tanımladığı için diğer doğruluk ölçütlerine göre birçok uygulayıcı tarafından tercih edilmektedir.

Ortalama Hata Kareleri Karekökü (RMSE) ise aşağıdaki şekilde hesaplanır (Hyndman & Koehler, 2006: 682).

$$RMSE = \sqrt{(1/n) \sum_{i=1}^n e_i^2} \quad (1.22)$$

1.5.5 Simetrik Ortalama Mutlak Yüzde Hata (sMAPE)

Ortalama Mutlak Yüzde Hata (MAPE) gibi zaman serileri ilgili doğruluk ölçütleri çoğu zaman diliminde talep değeri sıfır olan aralıklı talep tahmininde kullanılamamaktadır. Bu durumda Simetrik Ortalama Mutlak Yüzde Hata (sMAPE) hesaplanır (Syntetos & Boylan, 2005: 307).

Böyle bir durumda sMAPE ölçütünün kullanılmasının iki nedeni vardır. Birincisi talep değeri sıfıra yakın ve tahmin değeri taleple karşılaştırıldığında daha büyük ise yüzdesel hatanın büyümeyecek olmasıdır. İkinci neden ise hatanın simetrisidir. Talep değerinin tahmin değerinden daha büyük olması ya da aksi durumu önemli değildir, hata yine aynı kalacaktır (Wallström & Segerstedt, 2010: 628).

Simetrik Ortalama Mutlak Yüzde Hata (sMAPE) aşağıdaki şekilde hesaplanmaktadır.

$$sMAPE = (1/n) \sum_{i=1}^n \frac{|D_i - F_i|}{(D_i + F_i)/2} \times 100 \quad (1.23)$$

1.5.6 Ortalama Mutlak Sapmanın Geometrik Ortalaması (GMAMAD)

Zaman serileri ilgili doğruluk ölçütlerinin çoğu zaman diliminde talep değeri sıfır olan aralıklı talep tahmininde kullanılamaması sorununun üstesinden gelebilmek için kullanılan bir hata ölçütü de Ortalama Mutlak Sapmanın Geometrik Ortalaması (GMAMAD)'dır.

Ortalama Mutlak Sapmanın Geometrik Ortalaması aşağıdaki şekilde hesaplanmaktadır (Şahin vd., 2013: 10).

$$GMAMAD = \left(\prod_{i=1}^n (1/n) \sum_{i=1}^n |e_i| \right)^{1/n} \quad (1.24)$$

1.5.7 En İyi Yüzde Oranı (PB)

Syntetos ve Boylan (2005) aralıklı talep yapısına sahip ürünlerin talep tahmininde tahmin doğruluğunu ölçmek için En İyi Yüzde Oranı (PB) ölçütünü kullanmışlardır. En İyi Yüzde Oranı (PB) bir tahmin yöntemin diğer tahmin yöntemlerinden daha iyi olduğu zaman periyotlarının yüzdesini ifade etmektedir. PB ölçütü hesaplaması kolay, basit yorumlanabilir bir ölçüttür. Özellikle aralıklı talep verilerinin olduğu durumlarda anlamlı olmaktadır çünkü tüm seriler ve her bir seri içindeki tüm veri periyotları sonuç üretir.

Örnek olarak m tahmin yönteminin En İyi Yüzde Oranı (PB) matematiksel olarak aşağıdaki şekilde ifade edilir (Gutierrez vd., 2008: 414).

$$PB_m = \frac{\sum_{i=1}^n B_{m,i}}{n} \times 100 \quad (1.25)$$

i zaman dilimi için, $|D_{m,i} - F_i|$ değerinin diğer bütün k yöntemlerinin $|D_{k,i} - F_i|$ değerinden daha düşük olması durumunda $B_{m,i} = 1$ olarak kabul edilir, aksi halde $B_{m,i} = 0$ kabul edilir.

1.5.8 Ortalama Mutlak Ölçekli Hata (MASE)

Hyndman ve Koehler (2006) aralıklı talebe sahip ürünlerin talep tahmininde tahmin doğruluğunun ölçülebilmesi için Ortalama Mutlak Ölçekli Hata (MASE) olarak bilinen yeni bir ölçüt önermişlerdir. Farklı ölçeklere sahip, sıfıra yakın ya da negatif verilerin bulunduğu durumlarda Ortalama Mutlak Ölçekli Hatanın (MASE) tahmin doğruluğunun ölçülmesinde en iyi ölçüt olduğunu öne sürmüşlerdir.

Ortalama Mutlak Ölçekli Hata (MASE) aşağıdaki şekilde hesaplanmaktadır (Hyndman & Koehler, 2006: 685).

$$MASE = ortalama (|q_t|) \quad (1.26)$$

$$q_t = e_t / \frac{1}{n-1} \sum_{i=2}^n |D_i - D_{i-1}| \quad (1.27)$$

Bu hata ölçütü zaman serileriyle ilgili diğer ölçütlerde hesaplamadan doğan sıfır ve sonsuz hata olasılığını minimize etmektedir (Kourentzes, 2013: 202).

İKİNCİ BÖLÜM

MAKİNE ÖĞRENMESİ

Bu bölümde öncelikle makine öğrenmesinin tanımı yapılarak amacı, önemi ve uygulama alanlarından bahsedilecektir. Daha sonra makine öğrenme problemleri ve tipleri incelenecektir.

2.1 Makine Öğrenmesinin Tanımı

Makine öğrenmesi, yapay zekanın bir alt dalıdır. Dünyanın en anlaşılması zor makinesi olarak görülen insan beyni sayısal işlemleri belirli bir süre içinde yaparken kavramayla ilgili işlemleri çok kısa sürede yerine getirir. İnsan beyni bir olayla karşılaştığında olayla ilgili durumu çok hızlı bir şekilde sayısal olarak değerlendiremezse bile geçmişteki deneyimlerine dayanarak olayla ilgili hızlı bir tepki üretir ve çözüme ulaşabilir. Bilgisayarlar ise sayısal işlemleri çok hızlı sürede yapmalarına rağmen tecrübe ile edinilmiş bilgilerin kullanımı konusunda başarısız kalmaktadırlar. Bu durum akıllara acaba bilgisayarları kullanarak benzer bir zeka oluşturulabilir mi sorusunu getirmektedir (Elmas, 2010: 21).

Makine öğrenmesi, bilgisayarın geçmiş veri ve tecrübelerden faydalanarak hesaplama işlemi yaparak sistemleri dizayn etmesidir. Sistemler öğrenir ve tecrübeyle gelişir ve önceki öğrenmeyi temel alarak zamanla yeni problemlere karşı yeni çıktılar oluşturabilir (Bell, 2014: 2).

Makine öğrenmesi kavramı ilk olarak 1950 yılında Alan Turing tarafından yazılan “Bilgisayar Mekanizması ve Zeka” adlı makalesinde bilgisayarlar düşünür mü sorusuna dayanmaktadır. Makalesinde taklit oyunu adı verilen, bir insan, yine bir insan olan bir yargılayıcı ve bir makine olmak üzere üç katılımcıdan oluşan bir deneyi ortaya koymuştur. Yargılayıcı bir program vasıtasıyla insan ve makineyle iletişim içinde bulunmaktadır. Bu iletişimi sırasından insandan ve makineden almış olduğu cevaplar doğrultusunda hangisinin insan hangisinin makine olduğuna kara vermek durumundadır. Eğer yargılayıcı aldığı cevapların insandan mı yoksa makineden mi geldiğini sürekli olarak ayırt edemezse oyunu makine kazanacaktır. Bu test

günümüzde Loebner Ödülleri çerçevesinde her yıl gerçekleştirilen yapay zeka yarışmasında uygulanmaktadır (Bell, 2014: 1) .

Arthur Samuel (1959) makine öğrenmesini bilgisayarların açıkça programlanmadan öğrenme yeteneği kazanmasını sağlayan bir çalışma alanıdır şeklinde tanımlamaktadır (Bell, 2014: 2).

Tom M. Mitchell her bir bilgisayar programının bir takım görevler vasıtasıyla kendi performansını geliştireceğine değinmiştir. Bir bilgisayar programının P performans ölçütü tarafından ölçülen T görevlerinin performansının E tecrübesiyle geliştirilebilmesi durumunda, birtakım T görevleri ve P performans ölçütü doğrultusunda E tecrübesi ile öğrenebilmesinin söz konusu olduğunu ileri sürmektedir. Görev T, Performans Ölçütü P ve Tecrübe (Yapılan öğrenme) E bir öğrenme problemindeki üç önemli olan üç özelliğidir (Mitchell, 1997: 2-3).

Örnek olarak bir el yazısı tanınmasının öğrenilmesi problemini düşünebiliriz. Böyle bir problemde:

Görev T: Görüntüler içinden el yazısıyla yazılmış kelimelerin sınıflandırılması ve tanınması.

Performans Ölçütü P: Doğru sınıflandırılan kelimelerin yüzdesi.

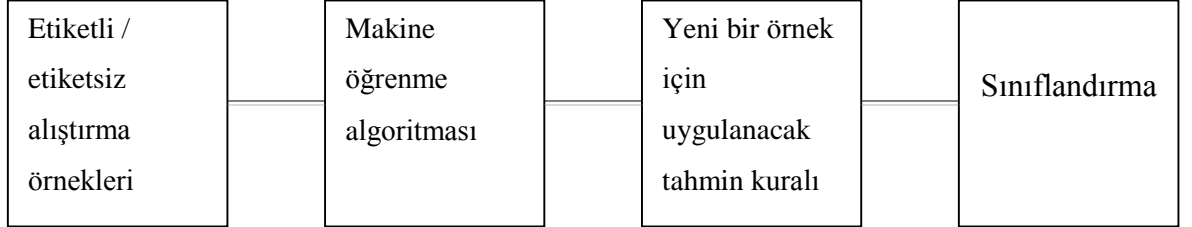
Tecrübe (Yapılan öğrenme) E: Sınıflandırılması verilen el yazısıyla yazılmış kelimelerin veri tabanı.

Özetleyecek olursak makine öğrenmesinde, bir bilgisayar geçmiş tecrübelerin yol gösterici olması durumunda kendi performansını arttırmak için bir dizi görevi işleyebilecek şekilde programlanmaktadır.

Makine öğrenmesini genel konsepti Şekil 2’de yer almaktadır.

Şekil 2 - Makine Öğrenmesi ve Sınıflandırma

Kaynak: Kulkarni, P., **Reinforcement And Systemic Machine Learning For Decision Making**,
New Jersey, USA, John Willey & Sons Inc, 2012, p.7



2.2 Makine Öğrenmesinin Amacı ve Önemi

Bilgisayar tarafından bir problemin çözümünün gerçekleştirilebilmesi için izlenmesi gereken işlemler dizisinin bilinmesi ve sistemde tanımlanmış olması gerekir. Fakat bazı problemlerin çözümünde bilinen bir komutlar dizisi olmayabilir.

Örnek olarak, e-posta kutumuza gelen e-postaların istenmeyenlerinin silinmesini gösterebiliriz. E-posta kutumuza gelen e-posta girdiyi oluştururken, bu e-postayı silip silmememiz için gelen evet / hayır sorusu ise çıktıyı oluşturmaktadır. Bu örnekte girdiyi göz önünde bulundurarak çıktıyı oluşturacak bir algoritma bulunmamaktadır. Sebebi ise bu e-postaların hangisinin silinmesi, hangisinin silinmemesi gerektiğinin zamana ve kişiye göre değişebilir olmasıdır. Aynı zamanda e-posta kutumuz çok fazla sayıda e-postanın olduğu veri setini de içerebilir. Böyle bir problemin çözümünde bilgisayarın geçmiş örnekleri göz önünde bulundurup hangi e-postanın silinip silinmemesi gerektiğini bu örneklerden öğrenmesini yani kendisinin bir algoritma üretmesi amaçlanır (Alpaydın, 2012: 1).

Makine öğrenmesinin ana amacı uygulanabilir değeri olan öğrenen algoritmalar üretmektir. Makine öğrenmesi, gerçek hayattaki karar verme ihtiyaçlarını karşılamak için kullanılan bilgisayar algoritma ve modellerinin gelişimini ve iyileştirilmesini hedefler ve birçok uygulamada bu hedefine başarıyla ulaşmaktadır. Makine öğrenmesinde sonuç çıkarımları veriye dayalı olarak gerçekleşir. Verinin kaynağı sınırlı olabilir ya da çoğunlukla kullanışlı verinin tanımlanmasında zorluklarla karşılaşılabilir. Büyük miktarda verinin bulunduğu durumlar ve bu verilerin arasında

önemli ilişki / korelasyonların olması da söz konusu olabilir. Makine öğrenmesi veri madenciliği uygulamalarının konusu olan veriler arasındaki bu ilişkiyi ortaya çıkartmaktadır. Makine öğrenmesinin amacı, gerçek hayat problemlerini çözmek için kullanılacak akıllı sistemlerin kurulmasını kolaylaştırmaktır (Kulkarni, 2012: 8).

Makine öğrenmesinde ilk olarak “ Neden makineler öğrenmeli? Niçin makineler ilk seferde istenildiği gibi performans göstermesi için tasarlanmıyorlar? ” sorularını aklı gelmektedir. Makine öğrenmesinin önemli olmasının birçok sebebi bulunmaktadır. Makine öğrenmesinin başarısı bize hayvanların ve insanların nasıl öğrendiğini anlamamızı sağlamakla birlikte makine öğrenmesinin önemli olmasının mühendislik sebepleri de vardır. Bu sebepler aşağıda yer almaktadır (Nilsson, 1998: 2-3).

- Girdi ve çıktı çiftlerini belirleyebilmek mümkün olabilir fakat girdi ve istenen çıktılar arasında kesin bir ilişki kurmak mümkün olmayabilir. Bu durum dışındaki bazı görevler iyi bir şekilde tanımlanamayabilir. Büyük sayıdaki örnek girdilere uygun doğru çıktılar üretilebilmesi için makinelerin kendi iç yapılarını oluşturabilmeleri istenir.
- Büyük veri gruplarının arasında gizlenmiş olan önemli ilişki ve korelasyonlar olması muhtemeldir. Makine öğrenme yöntemleri bu ilişkilerin ortaya çıkmasında kullanılabilir.
- İnsanlar çoğunlukla makineleri kullandıkları ortamda istenilen kadar iyi şekilde üretememektedirler. Makinenin üretilmesi sırasında çalışma ortamının genel özellikleri kesin olarak bilinmemektedir. Makine öğrenme yöntemleri mevcut makine tasarımının işin yapılış aşamasındaki gelişiminde kullanılabilir.
- Belli görevler hakkında mevcut bilginin miktarı insanlar tarafından kodlamak için çok büyük olabilmektedir. Makineler bu bilgiyi aşamalı olarak insanların kayda almasından daha çok kaydederler.
- Ortam zamanla değişmektedir. Makineler değişen ortama adapte olabilmekte ve yeniden tasarım ihtiyacını azaltmaktadırlar.
- Görevlerle ilgili yeni bilgi insanlar tarafından keşfedilmektedir. Kelime dağarcığı değişmektedir. Dünyada yeni olaylarla ilgili sürekli bir akış söz

konusudur. Sistemleri sürekli olarak yeni bilgiye uyumlu hale getirmek için yeniden tasarlamak elverişsiz olmaktadır. Fakat makine öğrenme yöntemleri bu yenilikleri kolayca takip edebilir.

2.3 Makine Öğrenmesi Uygulama Alanları

Makine öğrenmesi bilgisayar bilimi, mühendislik, istatistik ve daha birçok disiplinin içinde yer almaktadır. Politikadan yerbilimlerine kadar pek çok alanda başarıyla kullanılmaktadır ve birçok problemin çözümünde bir araçtır (Harrington, 2012: 6).

Öğrenen algoritmaların kullanıldığı uygulamalara örnek verecek olursak (Mohri vd. 2012: 1-2):

- Metin ve belge sınıflandırması, örnek olarak çok sayıda kişiye giden istenmeyen e-postaların tespit edilmesi,
- Doğal dil işleme, morfolojik analizler, istatistiksel ayrıştırma, metin parçası etiketleme, varlık ismi tanımlama,
- Konuşma tanıma, konuşma sentezleyici, konuşmacı doğrulama,
- Optik karakter tanıma
- Sayısal biyoloji uygulamaları, örnek olarak protein yapı tahmini,
- Bilgisayarlı görme, örnek olarak görüntü ve yüz tanıma
- Sahtekarlıkların saptanması (telefon ve kredi kartı), şebeke ihlali
- Oyunlar, örnek olarak satranç ve tavlâ
- Yardımcısız araç kontrolü (navigasyon)
- Medikal tanı
- Tavsiye sistemleri, arama motorları, bilgi çıkarım sistemleri

İstatistik, zeka modelleri, uyumsal kontrol teorisi, psikolojik modeller, evrimsel modeller ve yapay zeka gibi disiplinler makine öğrenmesine katkıda bulunmaktadır (Nilsson, 1998: 3-4).

2.4 Makine Öğrenme Problemleri

Öğrenen algoritmalar her geçen gün yeni uygulamalarda uygulanmaktadır ve bu uygulamalar birçok öğrenme problemini çözüme ulaştırmaktadır. Başlıca öğrenme problemleri aşağıdaki şekildedir (Mohri vd., 2012: 2).

- Sınıflandırma
- Regresyon
- Sıralama
- Kümeleme
- Boyut İndirgeme

2.4.1 Sınıflandırma

Sınıflandırma problemleri girdi vektörleri ve bu vektörlerin her sınıftan alınan örneklerle yapılan öğrenmelerle N kadar sınıftan hangisine ait olduğunun karar verilmesini içermektedir. Sınıflandırma problemindeki en önemli nokta her bir farklı örneğin tamamen bir sınıfa ait olması ve sınıfların olası tüm çıktı uzayından yola çıkmasıdır. Bu iki kısıtın gerçeğe uygun olması bir zorunluluk değildir, bazen örnekler kısmi olarak iki farklı sınıfa da ait olabilmektedir. Bunun dışında olası her girdiyi kategorize edemeyeceğimiz durumlar da söz konudur. Örnek verilecek olursa, para ile çalışan satış otomatındaki tüm farklı bozuk paraları tanımak için makine öğrenmesinin kullanıldığını düşünelim. Sınıflandırıcı o ülkeye ait para birimindeki bozuk paraları tanıyarak sınıflandırma yaparken farklı bir ülkeye ait bozuk para makineye atıldığında durum ne olacaktır? Eğer bozuk paraların görünüşleri birbirine yakınsa makine bu bozuk parayı o ülkenin bozuk parası olarak tanıyabilir. Ya da bu bozuk parada bir aykırılık olduğunu tespit edecektir. Bu örnekten de anlaşılacağı üzere tüm girdileri her zaman tam olarak sınıflandırmak mümkün olamamaktadır (Marsland, 2009: 9-10).

Makine öğrenmesinde makine geçmişe ait verileri değerlendirir ve öğrenen bir model oluşturur. Yeni bir girdiyle karşılaştığında oluşturduğu bu modele göre girdiyi ya kabul eder ya da reddeder. Geçmişe ait verilerden bir kural oluşturarak çıktılar elde eder.

Sınıflandırma problemlerine başka örnekler verecek olursak (Alpaydın, 2012: 5-6):

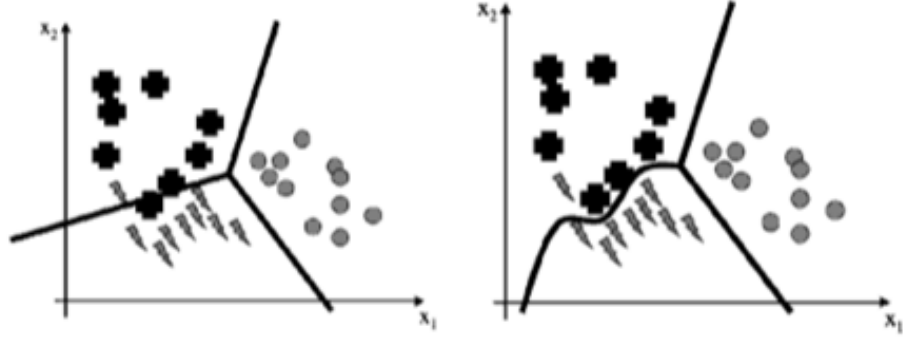
- Dokümanların politika, iş, spor, hava vb. şekilde kategorilere atanması,
- Resimlerin görüntü, porte, hayvan vb. şekilde kategorilere atanması,
- Medikal tanıda, hasta hakkında bilinen yaş, cinsiyet, geçmiş ve şimdiki bulgularına göre hastalıkların teşhis edilmesi,
- Optik karakter tanımda, harflerin görüntülerinden tanımak istediğimiz sınıf kadar karakterden hangisine ait olduğunun tespit edilmesi,
- Yüz tanımda, yüzlerin hangi insanlara ait olduğuna dair kimlik tespitinin yapılması,
- Doğal dil işlemede, makinenin bir dilde yer alan bir cümleyi başka bir dile çevirmesi,
- Konuşma tanımda, farklı seslere göre söylenmesi mümkün olan kelimelerin sınıflandırılması,
- Biyometride, insanların fiziksel özelliklerinden ya da davranışlarından yola çıkarak kimliklerinin teşhis edilmesi.

Dokümanların politika, iş, spor, hava vb. şekilde kategorilere atanması örneğinde kategorilerin küçük olmasıyla birlikte, yüz tanıma ve konuşma tanıma gibi örneklerde sınıflandırılacak kategorilerin sayıları çok geniş olabilmektedir (Mohri vd., 2012: 2).

Sınıflandırmada makinenin bir takım ölçütleri göz önünde bulundurması gerekir. Bozuk para sınıflandırması örneğinden gidecek olursak, bu ölçütler çap, ağırlık ve şekil gibi girdi vektörünü oluşturacak özellikler olabilir. Şekli kodlayabilmek için 1 = daire, 2 = altıgen gibi özelliklere sayılarla numaralandırma yapılır. Diğer özellikler de bu şekilde kodlanır. Eğer makine atomik absorpsiyon spektrometrisi mevcut ise maddenin yoğunluğu tahmin edilebilir ya da kamera varsa fotoğrafı çekilerek görüntü sınıflandırması yapılabilir. Burada önemli olan hangi girdilerin sağlanması gerektiğidir. Bozuk paranın sınıflandırması için daire, altıgen olarak kodlaması yapılan bir girdi eğer o ülkedeki bozuk paraların hepsi daire ise bu girdi kullanışsız bir girdi olacaktır. Gereğinden fazla girdinin bulunması sınıflandırıcı için vakit kaybına sebep olacaktır. İstenilen özelliklere sahip güvenilir sınıfların ayrıştırılmasında karar sınırlarının belirlenmesi gerekir (Marsland, 2009: 10).

Şekil 3 - Sınıflandırma Probleminde Karar Sınırları

Kaynak: Marsland, S., **Machine Learning An Algorithmic Perspective**, USA, A Chapman & Hall Book CRC Press, 2009, p. 11.



Şekil 3’de üç farklı sınıfa ve iki farklı karar sınırına sahip iki boyutlu girdiler gösterilmektedir. Sağdaki şekilde karar sınırları doğrusal olmayan bir şekilde kıvrılmaktadır. Soldaki şekilde ise karar sınırları düz çizgiler şeklindedir bu yüzden basittir fakat sağdaki şekilde kadar sınıfları iyi kategorize edememektedir.

2.4.2 Regresyon

Regresyon problemlerinde çıktılar gerçek bir değer olarak tahmin edilir. Regresyon problemlerine stok değerlerinin ve ekonomik değişkenlerin değişiminin tahmini örnek olarak gösterilebilir. Böyle bir problemde yanlış bir tahminin cezası gerçek ve tahmin değerleri arasındaki farkın büyüklüğüne bağlıdır, sınıflandırma probleminin aksine farklı kategoriler arasında bir yakınlık olduğu fikri bulunmamaktadır (Mohri vd., 2012: 2).

Regresyon problemine örnek olarak ikinci el otomobil fiyatlarını belirleyen bir bilgisayar programını verebiliriz. Bu programın girdisi otomobil fiyatlarıyla yakından ilişkili olabilecek marka, model, yıl, kilometre, renk, kasa tipi, motor hacmi vb. özelliklerdir. Çıktı ise bir sayı olarak ifade edilen otomobilin fiyatı olacaktır (Alpaydın, 2012: 7).

Otomobilin yukarıda belirtilen özellikleri x , otomobilin fiyatını tahmin edecek model ise y olarak tanımlanırsa, tek bir özellik göz önünde bulundurularak oluşturulacak doğrusal model aşağıda yer almaktadır (Orhunbilge, 2002: 16).

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (2.1)$$

β_1 , regresyon katsayısını ifade etmektedir ve x 'de yaşanacak bir birimlik değişikliğin y 'de oluşturacağı değişimi anlatır. ε , hatayı yani gerçek değer ile modelin tahmin değeri arasındaki farkı ifade eder. x değerinin sıfır olması durumunda y 'nin alacağı değer ise β_0 olarak ifade edilir. Bir regresyon probleminde amaç tahmin değeri ile gerçekleşen değer arasındaki farkı en küçük yapan modelin tanımlanmasıdır. β_0 ve β_1 parametreleri de bu amaca yönelik olarak aşağıdaki şekilde hesaplanır (Orhunbilge, 2002: 21).

$$\beta_1 = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}} \quad (2.2)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (2.3)$$

Doğrusal modelin kısıtlı kalması durumunda girdiyi daha yüksek bir dereceden yazarak çıktıyı 2.4 numaralı denklemdeki gibi hesaplamak mümkündür. Bu derecelendirme arttıkça kullanılan model daha kolay uyum sağlayacak ve hatanın azalmasını sağlayacaktır (Alpaydın, 2012: 28).

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon \quad (2.4)$$

Regresyon ve sınıflandırma arasındaki fark regresyonda tahmin edilen değer sürekli olması, sınıflandırmada ise kesikli olmasıdır. Regresyon istatistiğinin en kullanışlı araçlarından biridir. Hata karelerini en küçükmek regresyon denklemlerindeki girdi özelliklerinin ağırlıklarını en iyi şekilde bulmak için kullanılır (Harrington, 2012: 177).

2.4.3 Sıralama

Sıralama probleminin öğrenimi, örnek veri setlerini inceleyip bu verilerin önemine göre sıralayarak bir model oluşturulmasıdır. Sıralama öğrenimi yeni ve özellikle son

yıllarda makine öğrenmesinde çok popüler olan bir konu haline gelmiştir (Cao vd., 2007: 139).

Sıralama probleminin öğrenimi; arama motorlarının tasarımı, bilgi çıkarım platformları, film tavsiye sistemleri gibi birçok modern uygulamada kullanılmaktadır. Veri setinin çok büyük olmasından dolayı bir sınıflandırıcı tarafından tüm parçaların konuyla ilgili olarak etiketlenmesi uygulaması zor ya da imkansız olmaktadır. Arama motorunu kullanan bir kullanıcı sorgusuna karşılık tüm dokümanlara bakmak yerine en üstteki on dokümana bakmayı tercih edecektir. Bir başka örnek vermek gerekirse, bir kredi kartı işletmesinde sahtekarlıkların saptanmasıyla ilgili olarak çalışan bir departman çalışanı sahtekarlıkla ilgili potansiyel binlerce sınıflandırılmış işlemi araştırmak yerine sadece en fazla şüpheli görülen bir düzine işlemi araştırmayı tercih edecektir (Mohri vd., 2012: 209).

Sıralama öğreniminin en çok kullanıldığı uygulamalar; belgeye yeniden erişim, uzman arama, tanım arama, arama motorları tasarımı, soruların cevaplandırılması, anahtar cümle çıkarımı, doküman özetleme ve makine tercümesidir (Nonmember, 2011: 1).

Belgeye yeniden erişim uygulamasından örnek verecek olursak, sistem öncelikli olarak dokümanları biriktirir. Yapılan sorguya göre biriktirdiği dokümanlar arasından sorguda geçen kelimelere göre dokümanlara yeniden erişim sağlar ve sıralama fonksiyonu bu dokümanların her birine bir skor verir. Bu skorlara göre dokümanlar azalan bir şekilde sıralanır. Öğrenme ise şu şekilde gerçekleşir: her bir sorgu doküman listesinin kusursuz sıralanmasıyla yakından ilişkilidir, sıralama fonksiyonu eğitim verisini kullanarak oluşturulur ve böyle bir model sıralama listelerini tam olarak tahmin edebilir. Sıralama modelinin amacı, dokümanlarla sorgu arasındaki ilişkiye göre sıralı bir doküman listesi oluşturmaktır (Cao vd., 2007: 139).

Sıralama öğreniminde kullanılan yöntemlerin iki önemli özelliği bulunmaktadır (Liu, 2011: 17) :

1. **Özellik esaslılığı:** Araştırma altındaki tüm dokümanlar, dokümanlarla sorgu arasındaki ilişkiyi yansıtan özellik vektörleri tarafından temsil

edilmektedir. Vektör $x = \emptyset (qd)$ şeklinde gösterecek olursak, sorgu q , ilişkili doküman d , \emptyset ise özellik çıkarıcı olarak ifade edilmektedir.

2. **Ayrt edilen eğitim:** Öğrenme süreci, ayırt edilen öğrenmenin dört parçası tarafından tanımlanmaktadır. Bir sıralama öğrenimi yöntemi 1) kendi girdi uzayına, 2) çıktı uzayına, 3) hipotez uzayına ve 4) kayıp fonksiyonuna sahiptir. Ayırt edilen eğitim, eğitim verisini temel alan otomatik bir öğrenme sürecidir. Bu özellik, gerçek arama motorları için oldukça talep edilen bir özelliktir, çünkü bu arama motorları her gün çok sayıda kullanıcı geribildirim almaktadır. Bu geribildirimlerden otomatik öğrenim yapılması ve sıralama mekanizmasının anlık olarak geliştirilmesi oldukça önemlidir.

Yukarıda bahsedilen bu iki özelliğe bağlı olarak ticari arama motorlarının kullanımında sıralama öğrenimi yaygın bir şekilde kullanılmaktadır. Bununla birlikte sıralama öğrenimi akademik araştırma topluluklarında da ilgi toplamaktadır.

Sıralama probleminin öğreniminde en yaygın olarak kullanılan yaklaşımlar ise; nokta bazlı, çift bazlı ve liste bazlı yaklaşımlardır (Nonmember, 2011: 5).

2.4.4 Kümeleme

Kümeleme, özellikle büyük veri setlerinin analizinde örneklerin birbirleri arasındaki yakınlıklarını bulup homojen bölümlenimin yapılmasıdır. Verilerin ve örneklerin benzerlikleri esas alınarak gruplanması olarak da tanımlamak mümkündür (Mohri vd., 2012: 2).

Amerika'da 2000-2004 başkanlık seçimlerinin sonuçları birbirine çok yakın çıkmıştır. Adaylardan biri halk oylarının %50,7'sini alırken diğer aday %47,9 oy almıştır. Oy verenlerin belirli bir yüzdesi eğer taraf değiştirseydi seçim sonuçları çok farklı olabilirdi. Taraf değiştirebilecek oy verenler çok küçük bir grup olmasına karşın başa baş giden bu yarışta çok büyük bir değişikliğe sebep olacak ve sonucu doğrudan etkileyecektir. Taraf değiştirebilme ihtimali olan bu grup kısıtlı bütçeyle nasıl bulunabilir sorusu kümeleme problemine çok güzel bir örnektir. Böyle bir problemde öncelikli olarak oy verenlerle ilgili, onlar için neyin önemli olduğu ve oy vermelerinde

neyin etkili olduđu hakkında ipucu verecek her türlü bilgi toplanır. Daha sonra bu bilgiler kümeleme algoritmalarına yerleştirilir. Her bir küme için söz konusu olan oy verenlerin ilgisini çekecek bir mesaj hazırlanır. Seçim kampanyası başlar ve yöntemin çalışıp çalışmadığı ölçülür (Harrington, 2012: 207).

Kümeleme uygulamaları gerçek hayatta birçok alanda uygulanmaktadır. Kümeleme çoğunlukla görüntü analizlerinde, görüntü segmentasyonunda, yazılım değişim sistemlerinde ve sosyal ağ analizlerinde kullanılmaktadır. Bilgisayar bilimleri alanının dışında kümeleme algoritmaları, biyolojik sınıflandırma, gen analizi ve suç analizinde de kullanılmaktadır (Wali, 2014: 195).

Kümeleme problemlerinde hiyerarşik ve hiyerarşik olmayan yöntemler uygulanmaktadır. Hiyerarşik kümelemede, kümeleme için birçok adım kullanılmakta ve veriler tek bir adımda belirli kümelere bölünmemektedir. Böyle bir durumda aşamalı kümeleme söz konusudur. Tek bir kümeyle başlanır ve nesnelere yavaşça uygun kümelere bölünür. Hiyerarşik kümeleme bir dizi agromatif (birleştirici) yöntemler içermektedir, n kadar nesnenin anlamlı gruplar şeklinde toplanmasıyla başlar ve ayırıcı yöntemler ile n kadar nesne ardışık şekilde uygun gruplara ayrılır. Hiyerarşik yöntemler genellikle aykırı değerlere karşı hassastır ve küme sayıları optimal bir şekilde belirlenir. En çok kullanılan K-Ortalamlar yöntemi, hiyerarşik olmayan yöntemlerden biridir (Kulkarni, 2012: 192).

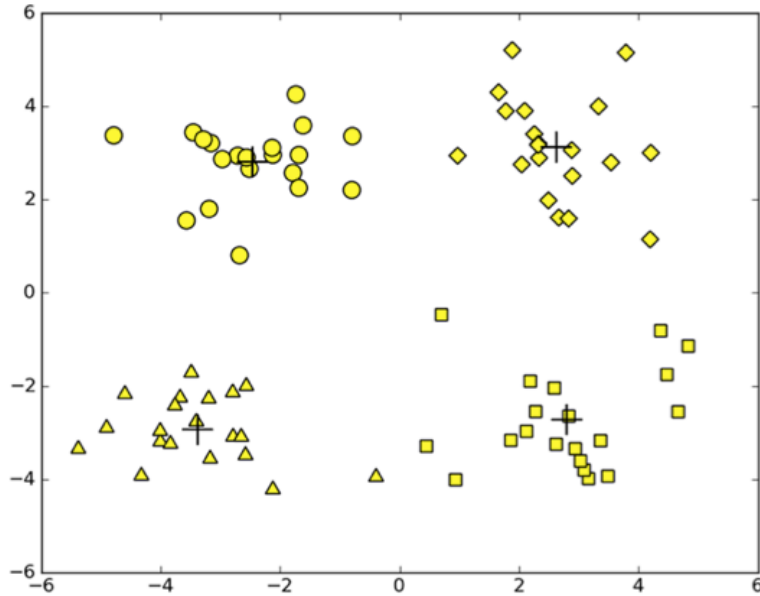
K-Ortalamlar yöntemi, bir veri setinde k kadar küme bulan bir algoritmayı içermektedir. Kümelerin sayısı k , kullanıcı tarafından belirlenir. Her bir küme ağırlık merkezi olarak bilinen bir nokta tarafından tanımlanır. Ağırlık merkezi kümedeki tüm noktaların merkezindedir. Öncelikle k ağırlık merkezleri rastgele olarak bir noktaya atanır. Daha sonra her bir nokta kendisine en yakın olan ağırlık merkezinin bulunduğu kümeye atanır. Son olarak ağırlık merkezleri, kümedeki tüm noktaların değerinin ortalaması alınarak güncellenir. Bu adımlar gerçekleştirilirken en yakın ağırlık merkezini bulabilmek için herhangi bir uzaklık ölçüsü kullanılır. Veri setindeki K-Ortalamlar'ın performansı kullanılan uzaklık ölçüsü tarafından belirlenir (Harrington, 2012: 209).

K-Ortalamlar yönteminin uygulama alanına örnek olarak nicemlemeyi gösterebiliriz. Nicemleme, çok büyük değerlerin bulunduğu bir kümenin daha küçük değerlere eşlenmesi sürecidir ve görüntü işlemede sıklıkla kullanılır. Örnek vermek gerekirse, 16 milyon renk içeren ve her noktası 24 bitle gösterilen bir görüntümüzü her nokta için 8 bit kullanan bir bilgisayar ekranında görüntülemeye çalıştığımızı düşünelim. Ekran her nokta için 8 bit kullandığından dolayı en fazla 256 renk görüntülenecektir. Bu sebeple 16 milyon rengin 256 rene indirilmesi gerekir. Tabii bu işlem yapılırken görüntünün en az bozulmaya uğraması göz önünden bulundurulması gerekir. Böyle bir örnekte K-Ortalamlar yöntemi kullanılarak renk nicemlemesi gerçekleştirilir (Alpaydın, 2012: 119).

Şekil 4’de K-Ortalamlar yöntemiyle yapılmış kümeleme sonuçları yer almaktadır. Örnekte üç iterasyon sonunda oluşan sonuçlar yer almaktadır. Benzer şekillerden oluşmuş kümeler belirlenmiştir ve kümelerin ortasındaki artı işaretleri ise ağırlık merkezlerini ifade etmektedir.

Şekil 4 - K-Ortalamlar Yöntemi Örnek Sonuçları

Kaynak: Harrington, P., *Machine Learning In Action*, NY, USA, Manning Publications Co, 2012, p.212.



2.4.5 Boyut İndirgeme

Boyut indirgeme, çok fazla verinin bulunduğu çok boyutlu bir uzaydan verinin etkin özelliklerini daha düşük boyutlu bir uzaya indirmektedir (Mohri vd., 2012: 281).

Boyut indirgemeye örnek verecek olursak, bir monitörden içinde bir topun da bulunduğu bir spor müsabakası izlendiğini düşünelim. Monitör milyonlarca pikseli içermektedir ve top binlerce pikselden oluşmaktadır. İnsan beyni ne yapacağını bilmektedir ve topun hareketini düşünmeden doğal olarak takip edecektir. Böyle bir sahnenin arkasında aslında monitördeki milyonlarca piksel, topun hareketini gösteren üç boyutlu bir görüntüye dönüşmektedir. Özetle milyonlarca boyutlu veri üç boyuta indirgenmektedir. Makine öğrenme algoritmaları uygulanmaya başlamadan önce genellikle uygun özelliklerle belirlenmek zorundadır. Daha az boyutlu veriyle çalışmak her zaman daha kolay olmaktadır (Harrington, 2012: 269).

Boyut indirgeme yöntemlerinin anahtar unsurları aşağıda yer almaktadır (Mohri vd., 2012: 281).

- **Sayılaştırma:** Başlangıç verisini ön işleme adımıyla daha sonraki işlemleri hızlandırmak adına sıkıştırır.
- **Görselleştirme:** Girdileri iki ya da üç boyutlu uzaya indirgeyip haritalayarak araştırmayla ilgili olarak görselleştirir.
- **Özellik çıkarımı:** Daha küçük ve daha etkin bir özellik grubu oluşturur.

En iyi bilinen ve en çok kullanılan boyut indirgeme yöntemi temel bileşen analizi yöntemidir. Temel bileşen analizinde örnek verideki girdi değişkenleri doğrusal ilişkisiz değişkenlere dönüştürülmektedir. Bu dönüştürülen özellikler örnek verinin temel bileşenleri olarak adlandırılmaktadır. Temel bileşen analizi bir grup girdi değerlerinin temel bileşenlerini hesaplamak için kovaryans matrisini ve tekil değer ayrışımını kullanır. Temel bileşen analizi algoritması n girdi vektöründeki m kadar boyutu indirmek için sırasıyla aşağıdaki adımları izler (Wali, 2014: 223).

1. X girdi vektöründen kovaryans matrisi Σ hesaplanır.

$$\text{kovaryans matrisi } \Sigma = \frac{1}{N} X^T X \quad (2.5)$$

2. Kovaryans matrisi Σ üzerinde tekil deęer ayrışımı (SVD) uygulanarak U , S ve V matrisleri hesaplanır. U matrisi $m \times m$, S matrisi $m \times n$ ve V matrisi $n \times n$ büyüklüğündedir. Matris M , örnek verideki m kadar boyutlu n kadar girdi vektörünü temsil etmektedir. S matrisi konvansiyonel bir matristir ve M matrisinin tekil deęeri olarak ifade edilir. U ve V matrisleri ise M matrisinin sol ve saę tekil vektörleridir. Temel bileşen analizinde, S matrisi indirgeme bileşeni ve U matrisi örnek verinin rotasyon bileşeni olarak adlandırılmaktadır.

$$M_{m \times m} = U_{m \times m} S_{m \times n} V_{n \times n} \quad (2.6)$$

3. $m \times m$ büyüklüğündeki U matrisinden, Σ matrisinin indirgenmiş sol tekil vektörü ya da indirgenmiş rotasyon matrisi adı verilen $U_{\text{indirgenmiş}}$ matrisini oluşturmak için ilk k kolonları seçilir.

4. Z ile ifade edilen k boyutlu vektörler hesaplanır.

$$Z = U_{\text{indirgenmiş}}^T \times X \quad (2.7)$$

Z matrisinin büyüklüğü k boyutlu n girdi vektörünü temsil eden $k \times n$ 'dir. k boyut sayısının daha düşük deęerli olması veride daha yüksek deęerli varyans kaybına neden olmaktadır. Bundan dolayı, k deęeri varyansın sadece küçük bir kısmı kaybolacak şekilde seçilmelidir.

Matris Z 'den ve $U_{\text{indirgenmiş}}$ sol tekil vektöründen orjinal X girdi vektörü aşağıdaki şekilde yeniden oluşturulur.

$$X = U_{\text{indirgenmiş}} \times Z \quad (2.8)$$

2.5 Makine Öğrenme Tipleri

Makine öğrenme algoritmaları problemlerin çözümünde amaçlanan sonuca göre farklı şekillerde sınıflandırılmaktadırlar. Sıklıkla kullanılan makine öğrenme tiplerini sıralayacak olursak (Mohri vd., 2012: 7-8):

- Denetimli Öğrenme
- Denetimsiz Öğrenme
- Yarı Denetimli Öğrenme
- Transdüktif Öğrenme
- Takviyeli Öğrenme
- Aktif Öğrenme
- Çevrimiçi Öğrenme

2.5.1 Denetimli Öğrenme

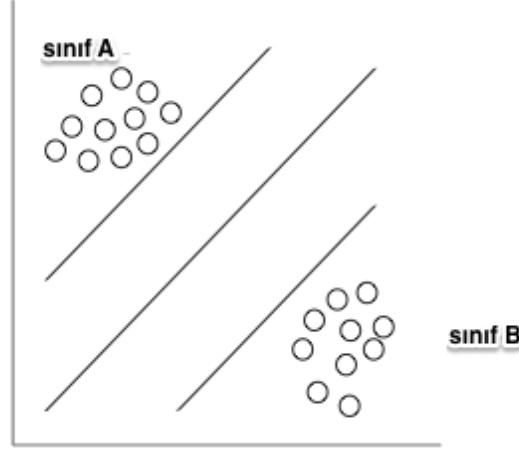
Denetimli öğrenmede, öğrenciler sınıflandırılmış bir dizi örneği eğitim verisi olarak kullanırlar ve görünmeyen tüm noktalar için tahminde bulunurlar. Bu öğrenme tipi özellikle sınıflandırma, regresyon ve sıralama problemleriyle yakından ilişkilidir (Mohri vd., 2012: 7).

Denetimli öğrenme, sınıflandırılmış verilerden bir model oluşturan bir makine öğrenme görevini tanımlamak için kullanılır. Burada bahsedilen sınıflandırılmış, örnek veriler gözlemlenen değerleri ifade etmektedir. Başlangıç verisi, denetimli makine öğrenme yöntemleri tarafından yeni bir model yaratmak için kullanılır ve modelin eğitim verisi adını alır (Wali, 2014: 41).

Sınıflandırma problemi için dokümanların sınıflandırılmasında denetimli öğrenmeye örnek verecek olursak, öğrenci mevcut dokümanları ve onların sınıflarını esas alarak öğrenmeyi gerçekleştirmektedir. Yazılım sınıfları özelleştirmek için girdi dokümanlarını haritalayacaktır, yani her sınıfa (doküman tipi) bir nesne (doküman) atayacaktır. Denetimli öğrenmenin görevi, bir dizi sınıflandırılmış eğitim verisinden bir sınıflandırıcı oluşturmaktır. Tipik bir sınıflandırma örneği Şekil 5’de gösterilmektedir. Şekil 5, öğrenmeden sonra sınıf A ve sınıf B olmak üzere iki sınıfa ayrılmış bir hiper düzlemi göstermektedir. Burada noktalar dokümanları ifade etmektedir. Öğrenme, dokümanlar arasına ayırıcı bir çizgi ya da hiper düzlem hesaplamıştır. Bilinmeyen bir doküman tipinin pozisyonu bu ayırıcıya göre belirlenecektir (Kulkarni, 2012: 2).

Şekil 5 - Denetimli Öğrenme

Kaynak: Kulkarni, P., **Reinforcement And Systemic Machine Learning For Decision Making**, New Jersey, USA, John Willey & Sons Inc, 2012, p. 2.



Denetimli öğrenmede kullanılan en yaygın algoritmalar ise; K-en yakın komşu algoritması, saf bayes sınıflandırıcısı, destek vektör makineleri, karar ağaçları, doğrusal sınıflandırma, yerel ağırlıklı doğrusal regresyon, Lasso ve Ridge regresyonudur (Harrington, 2012: 10).

2.5.2 Denetimsiz Öğrenme

Denetimsiz öğrenmede, öğrenciler sadece sınıflandırılmamış eğitim verisini kullanırlar ve görünmeyen tüm noktalar için tahminde bulunurlar. Bu öğrenme tipinde genel olarak sınıflandırılmış bir örnek mevcut değildir, bu yüzden öğrencinin performansını sayısal olarak değerlendirmek zor olmaktadır. Kümeleme ve boyut indirgeme denetimsiz öğrenme problemlerine birer örnektir (Mohri vd., 2012: 7).

Denetimsiz öğrenmede, algoritma birlikte kategorize edilebilecek girdiler arasındaki benzerlikleri tanımlamaya çalışmaktadır. Denetimsiz öğrenmeye istatistiksel yaklaşım ise yoğunluk tahmini olarak bilinmektedir (Marsland, 2009: 6).

Düzgün bir şekilde sınıflandırılmış verilerin bulunmadığı durumlarda denetimli öğrenme yapılabilmesi mümkün değildir. Böyle durumlarda denetimsiz öğrenmeye ihtiyaç duyulur ve benzerliklerle farklılıklar denetimsiz öğrenmeyle matematiksel

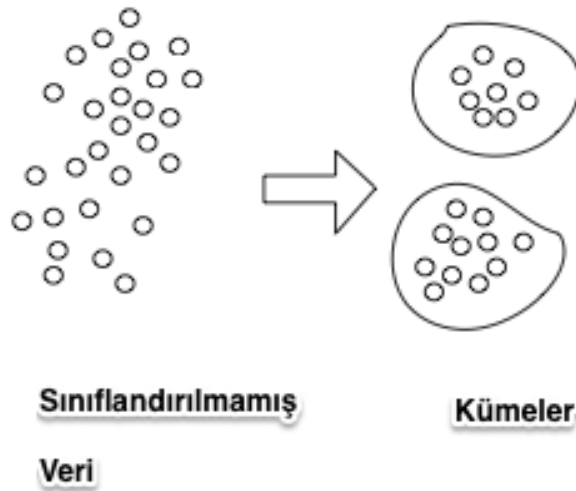
olarak temsil edilir. Birçok nesnenin bulunduğu bir ortamda, sıklıkla bu nesnelere anlamaya çalışırız ve benzerliklerini göz önüne getirmeye çalışırız.

Benzerlikler üzerinden bir örnek verecek olursak, bir çocuk bir kuşu diğer hayvanlardan kolayca ayırabilir. Ayırma işlemini yaparken bir özelliğini ya da benzerliğini göz önünde bulundurur, kuşların kanatları olması gibi. Başlangıç aşamasındaki kriter bu nesnelere en görünür yönleridir. Birçok denetimsiz öğrenme algoritması benzerlik esaslı haritalamalarda benzer hiyerarşik düzenlemeleri oluşturmaktadır. Hiyerarşik kümelemenin görevi, nesnelere bir hiyerarşiye göre küçük nesne grupları şeklinde düzenlemektir. Hiyerarşik olmayan kümeleme ise veriyi parçalanmış kümelere bölmeyi amaçlar (Kulkarni, 2012: 3).

Şekil 6'da öğrencinin dağınık noktalarda, öğrenmeden sonra iki farklı küme oluşturduğu görülmektedir. Benzer özellik ve yakınlıklara göre noktalar gruplanmış ve kümeler oluşturulmuştur.

Şekil 6 Denetimsiz Öğrenme

Kaynak: Kulkarni, P., **Reinforcement And Systemic Machine Learning For Decision Making**, New Jersey, USA, John Willey & Sons Inc, 2012, p. 4.



Denetimli öğrenmede, bir denetleyici tarafından sunulan çıktı ile girdi değerleri arasında bir eşleştirme ile öğrenme yaparak doğru değeri bulmak söz konusudur. Denetimsiz öğrenmede ise bir denetleyici yoktur ve eldeki girdi verileriyle uygunluk bulunmaya çalışılmaktadır. Girdi verilerinde bir takım örneklere daha sık rastlanırken bir takım örneklere daha az rastlanır. Böyle bir durumda hangi örneklerin daha sık, hangi örneklerin ise daha nadir gerçekleştiğini öğrenmek isteriz. Örnek olarak müşteri ilişkileri yönetimi için bir şirketin müşterilerine özel farklı hizmet ve ürün sunabilmek amacıyla, müşterilerini demografik özelliklerine göre farklı sınıflara ayırması gösterilebilir (Alpaydın, 2012: 8).

Denetimsiz öğrenmede kullanılan en yaygın algoritmalar ise; K-Ortalamalar, yoğunluk temelli mekânsal kümeleme (DBSCAN), Parzen penceresi ve beklenti maksimizasyonu algoritmalarıdır (Harrington, 2012: 10).

2.5.3 Yarı Denetimli Öğrenme

Yarı denetimli öğrenmede, öğrenciler hem sınıflandırılmamış hem de sınıflandırılmamış eğitim verisini kullanırlar ve görünmeyen tüm noktalar için tahminde bulunurlar. Yarı denetimli öğrenme genelde sınıflandırılmış verilerin elde edilmesinin maliyetli, sınıflandırılmamış verilere ise kolay ulaşımın sağlandığı durumlarda tercih edilir. Sınıflandırma, regresyon ve sıralama gibi problemlerde yarı denetimli öğrenme yöntemleri kullanılmaktadır (Mohri vd., 2012: 7).

Yarı denetimli öğrenme, denetimli öğrenme ve denetimsiz öğrenmenin ortasında yer almaktadır. Bu öğrenme tipinde sınıflandırılmamış veriye ek olarak algoritma denetimli bilgiyle oluşturulmaktadır. Veri seti $X = (x_i)_{i \in [n]}$ iki parçaya ayrılmaktadır: $Y_l := (y_1, \dots, y_l)$ etiketli $X_l := (x_1, \dots, x_l)$ noktaları ve etiketi bilinmeyen $X_u := (x_{l+1}, \dots, x_{l+u})$ noktaları (Chapelle vd., 2006: 2).

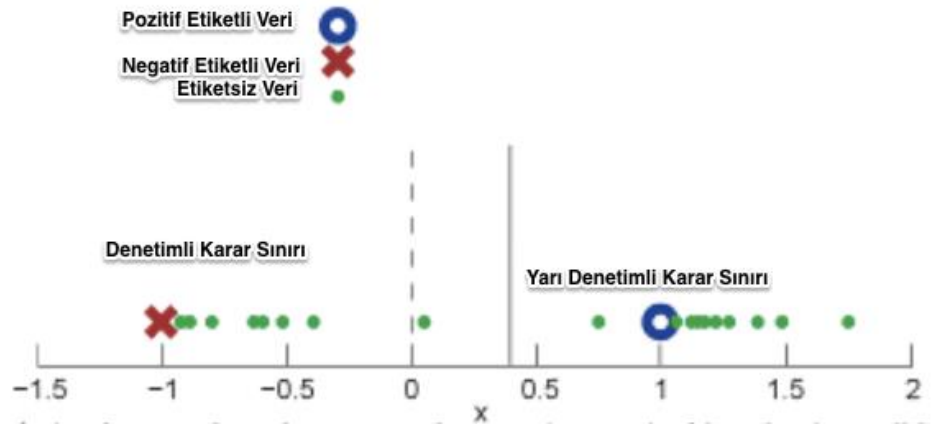
Yarı denetimli öğrenme, hem etiketli hem de etiketsiz verileri kullandığı için denetimli öğrenmeden daha iyi performans sağlamaktadır. Yarı denetimli öğrenmeye bir örnek verilecek olursak, Şekil 7'de tek boyutlu özelliğe sahip $x \in \mathbb{R}$ örnekler yer

almaktadır. Pozitif ve negatif olmak üzere iki sınıf mevcuttur (Zhu & Goldberg, 2009: 11).

- Denetimli öğrenmede $(x_1, y_1) = (-1, -)$ ve $(x_2, y_2) = (1, +)$ olmak üzere iki tane etiketli eğitim verisi bulunmaktadır ve Şekil 7’de çarpı ve yuvarlak simgeleriyle gösterilmektedir. Tahmin için karar sınırı açıkça bellidir ve $x = 0$ ’dır: $x < 0$ olan tüm örnekler $y = -$ olarak sınıflandırılmakta, $x \geq 0$ olan tüm örnekler içinse $y = +$ olarak sınıflandırılmaktadır.
- Etiketli verilere ek olarak Şekil 7’de küçük noktalar olarak gösterilmiş etiketsiz veriler de yer almaktadır. Bu etiketsiz verilerin doğru sınıfları tam olarak bilinmemektedir ve bu verilerin iki sınıfa ataması gerçekleştirilecektir. Her bir sınıftaki tüm örneklerin normal dağılıma göre dağıldığı varsayımı altında, etiketlenmemiş verilerden bilgi sağlanmaktadır. Hem etiketlenmiş hem de etiketlenmemiş verileri kullanan yarı denetimli öğrenme iki grup arasındaki karar sınırını $x \cong 0,4$ olarak tahmin etmektedir. Etiketsiz verilerin dağılımı aynı etiketli bölgelerin tanımlanmasına yardımcı olmuş ve daha az etiketli veriyle gerçek etiketler belirlenmiştir.

Şekil 7 - Yarı Denetimli Öğrenme

Kaynak: Zhu, X. ve Goldberg, A. B., **Introduction To Semi-Supervised Learning**, Ed. by. R. J. Brachman, ve T. G. Dietterich, NY, US, Morgan&Claypool Publishers. 2009, p.12.



2.5.4 Transdüktif Öğrenme

Yarı denetimli öğrenmede olduğu gibi, öğrenciler hem sınıflandırılmamış test verileri hem de sınıflandırılmamış eğitim verisini kullanırlar. Bununla birlikte, transdüktif öğrenmenin amacı sadece bu belirli sınıflandırılmamış test verileri için tahminde bulunarak sonuç çıkartmaktır ve birçok modern uygulamanın ihtiyaçlarını karşılamaktadır. Fakat yarı denetimli öğrenmede olduğu gibi hangi varsayımlar altında daha iyi performans gösterip göstermediği tam anlamıyla çözülememiş ve hala tartışma konusudur (Mohri vd., 2012: 7).

Transdüktif öğrenme, eğitim örneği $\{(x_i, y_i)\}_{i=1}^l$, $\{x_j\}_{j=l+1}^{l+m}$ için $f: x^{l+m} \rightarrow y^{l+m}$ fonksiyonunu oluşturur ve oluşturulan bu f fonksiyonun etiketlenmemiş $\{x_j\}_{j=l+1}^{l+m}$ verisi için iyi bir öngörüle bulunması beklenir. Unutulmamalıdır ki f fonksiyonu sadece eğitim örneği için tanımlıdır, eğitim örneğinin dışında kalan durumlar için öngörüle bulunmaz (Zhu & Goldberg, 2009: 12).

Transdüktif öğrenme için bilgiye erişim probleminden bir örnek verebiliriz. Kullanıcı öncelikli olarak bir arama motoruna sorgusunu girer, daha sonra sorgusuna karşılık gelen bu dokümanları ilgili olup olmamalarına göre etiketler. Bu dokümanlar bir sınıflandırma probleminin eğitim örneklerini oluşturur. Amaç veri tabanında kalan diğer dokümanlar için sorguyla ilişki durumuna göre doğru sınıflandırmalar yapılması için bir kural oluşturmaktır. Problem bu şekilde açıkça denetimli öğrenme problemi olarak düşünülebilir. Fakat transdüktif öğrenme en az iki yönüyle diğer öğrenme yöntemlerinden farklılaşmaktadır (Chapelle vd., 2006: 105) :

1. Öğrenme algoritması genel bir kural oluşturmak yerine sadece veri tabanında yer alan dokümanlar için yani sınırlı sayıda test örneği için doğru bir öngörüle bulunmaya çalışır.
2. Test örnekleri deneyden önce bilinir ve eğitim sırasında öğrenme algoritması tarafından gözlemlenebilir. Bu durum öğrenme algoritmasına test örneklerinin merkezinde yer alan her türlü bilgiden faydalanmasını sağlar.

Transdüktif öğrenme, yarı denetimli öğrenmenin özel bir türüdür ve test verisindeki etiketlenmemiş örnekleri kullanmasını sağlar.

Transdüktif öğrenme ve tümevarımsal öğrenmeyi birbirinden ayırmak için ilginç bir örnek verilebilir. Tümevarımsal öğrenme, sınıfta yapılan bir sınava benzer. Öğrenciler sınava girmeden önce soruları bilmediklerinden dolayı olası çıkabilecek tüm sorulara hazırlanmak zorundadırlar. Transdüktif öğrenme ise tümevarımsal öğrenmenin aksine eve götürülmüş ve evde yapılmış sınav gibidir. Öğrenciler sınav sorularını bilirler ve bu sınav sorularına hazırlanırlar, bu soruların haricindeki soruları hazırlamalarına gerek yoktur (Zhu & Goldberg, 2009: 12).

2.5.5 Takviyeli Öğrenme

Takviyeli öğrenmede, eğitim ve test aşamaları birbirine karışmış durumdadır. Öğrenci bilgiyi toplamak için çevresiyle aktif bir şekilde etkileşim içinde bulunur ve kimi zaman çevreyi etkiler. Öğrenci eylemlerinin sonucunda hemen ödül alır ve amacı ödülünü yükseltmek için çevresiyle çok daha fazla etkileşime girmek ve eylemde bulunmaktır (Mohri vd., 2012: 8).

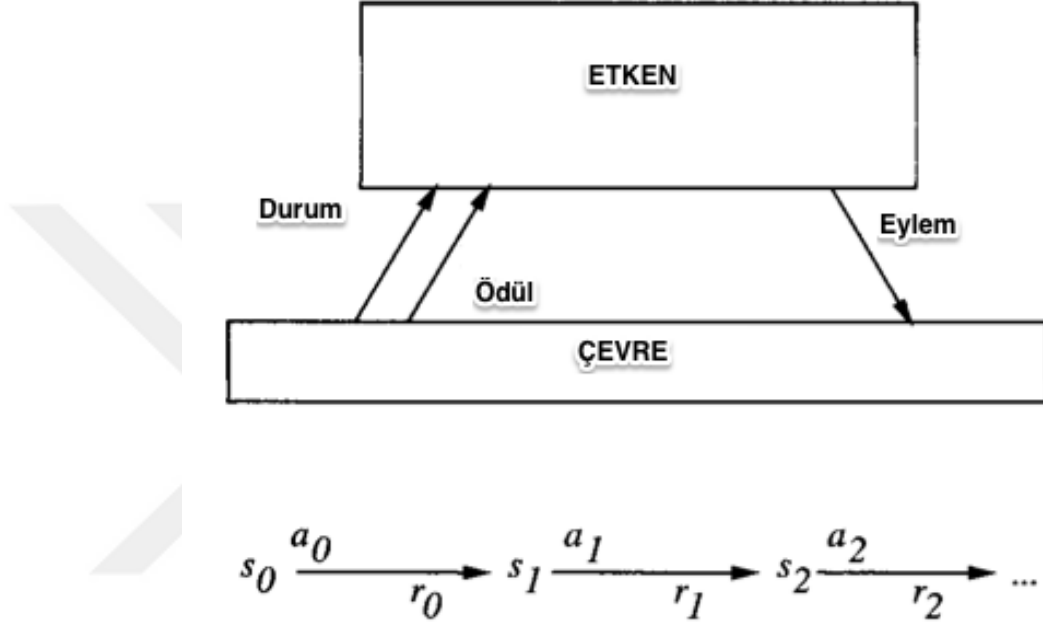
Takviyeli öğrenmeye satranç oyununu öğrenmeye çalışan bir yazılımı örnek verebiliriz. Birçok oyun boyunca denetim yaparak her durum için hangi hamlenin yapılmasını gerektiğini söyleyecek bir denetleyici oldukça maliyetli olacaktır. Bu sebeple denetimli öğrenme algoritması kullanmak tercih edilmez. Bununla birlikte satranç oyununda bir hamlenin iyi bir hamle olması onu takip eden diğer hamlelerle ilişkilidir. Tek bir hamle bir şey ifade etmeyeceği gibi arka arkaya yapılan doğru hamleler oyunun kazanılmasını sağlayacaktır. Bu örnekte elde edeceğimiz tek geribildirim oyun sonunda elde edilen galibiyet ya da mağlubiyettir (Alpaydın, 2012: 383).

Şekil 8’de etken diye tanımlanan öğrenci, olası bir dizi S durumunun bulunduğu bir çevreyle etkileşim içindedir. Bu eylem dizisi içindeki herhangi bir eylemi gerçekleştirebilir. Etken s_t durumu için gerçekleştirdiği a_t eylemi için r_t ödülüne

sahip olur. Etkenin görevi $\pi : S \rightarrow A$ yönetim politikasını öğrenmek ve en fazla ödül sağlayacak eylemleri gerçekleştirmektir (Mitchell, 1997: 368).

Şekil 8 - Takviyeli Öğrenme

Kaynak: Mitchell, T. M., **Machine Learning**, NY, USA, MacGraw Hill Education, 1997, p.368.



2.5.6 Aktif Öğrenme

Aktif öğrenmede, öğrenci yeni noktaların etiketlerini öğrenmek için bir uzmanı sorgulayarak uyarlamalı ve etkileşimli bir şekilde eğitim verilerini toplar. Aktif öğrenmenin amacı, denetimli öğrenmeyle karşılaştırıldığında, daha az etiketli örnek kullanarak daha iyi bir performansa ulaşmaktır. Sayısal biyoloji uygulamaları gibi genellikle etiketli verilerin sağlanmasının zor ve maliyetli olduğu durumlarda uygulanır (Mohri vd., 2012: 8).

Öğrenme senaryoları öğrencilerin oynadıkları rollere göre değişiklik göstermektedir. Bu senaryolarda aktif ve pasif öğrenciler birbirlerinden kolaylıkla ayırt edilebilirler. Aktif öğrenciler eğitim sırasında çevreleriyle etkileşim içinde, soru sorarak deneyi gerçekleştirirken pasif öğrenciler ise çevreyle hiçbir etkileşimde bulunmadan sadece ortamın sağladığı bilgileri gözlemlerler.

Çok sayıda kişiye gelen istenmeyen e-postaları filtreleme için yapılan bir öğrenmeyi gerçekleştiren öğrenci genellikle pasif öğrencidir. Kullanıcıların kendilerine gelen e-postaları işaretlemesini bekler. Aktif öğrenmede ise öğrenci, hangi e-postanın istenmeyen bir e-posta olduğunu öğrenmeyi geliştirmek için, kendi tarafından seçilen belirli e-postaları kullanıcının etiketlemesini ister (Shalev-Schwartz & Ben-David, 2014: 5).

2.5.7 Çevrimiçi Öğrenme

Çevrimiçi öğrenmede, diğer öğrenme tiplerinin aksine çevrimiçi senaryosu çoklu dizileri içermektedir. Eğitim ve test aşamaları birbirine karışmış durumdadır. Öğrenci her bir dizide etiketlenmemiş eğitim verisi kullanır, doğru etiketlere ulaşabilmek için öngöründe bulunur ve bir kayba neden olur. Çevrimiçi öğrenmenin amacı tüm dizilerdeki kümülatif kaybı en aza indirmektir (Mohri vd., 2012: 7).

Çevrimiçi öğrenmede öğrenci öğrenme süreci boyunca gerçek zamanlı olarak karşılık verme durumundadır ve öğrenci büyük miktarda veriyi işleme şansı elde ettikten sonra uzmanlık kazanacaktır. Örnek olarak günlük kararlar vermek zorunda olan bir borsacıyı gösterebiliriz. Borsacı günlük kararlarını o zamana kadar biriktirmiş olduğu tecrübeler ışığında vermektedir. Süreç içerisinde kimi zaman maliyetli hatalar yapmış fakat zamanla uzmanlık kazanmıştır. Çevrimiçi öğrenmede öğrenci, birçok öğrenme yönteminin aksine çıktı sonuçlarına ulaşmadan büyük miktarda eğitim verisini kullanmaktadır (Shalev-Schwartz & Ben-David, 2014: 5).

ÜÇÜNCÜ BÖLÜM

DESTEK VEKTÖR MAKİNELERİ

Bu bölümde istatistiksel öğrenme teorisinden bahsedilerek sınıflandırma ve regresyon için kullanılan destek vektör makineleri incelenecektir.

3.1 Destek Vektör Makinesi Tanımı

Yeni bir öğrenme yöntemi olan Destek Vektör Makinesi (DVM) özellikle sınıflandırma ve regresyon problemlerinin çözümünde kullanılmaktadır. Destek Vektör Makinesi ilk olarak 1995 yılında Vapnik tarafından kullanılmıştır.

DVM istatistik öğrenim teorisi üzerine bir çeşit sınıflandırıcının öğrenim yöntemidir. Bu algoritma doğrusal sınıflandırıcıdan türemektedir ve iki çeşitli sınıflandırıcı problemini çözebilmekte, doğrusal olmayan çalışma alanlarına da uygulanmaktadır. Başka bir deyişle örnek kümeleri sınıflandırmak için en iyi hiper düzlemin bulunmasına olanak sağlamaktadır. Destek Vektör Makinesi deneysel risk minimizasyonu yönteminden ziyade istatistiksel öğrenme teorisindeki yapısal risk minimizasyonu prensibini esas almaktadır (Wu, 2009: 7604).

DVM makine öğrenmesinin en popüler denetimli öğrenme algoritmalarından biridir. Bu algoritma özellikle geniş ve yüksek boyutlu veri setlerinde diğer makine öğrenme algoritmalarına göre daha iyi sınıflandırma performansı sağlamaktadır (Marsland, 2009: 119).

DVM, çözümün aralıklı olmasında ve kernel fonksiyonlarının kullanımı için karar fonksiyonlarının kapasite kontrolü tarafından tanımlanan özel öğrenme algoritmalarındandır. Diğer sinir ağlarında uygulanan deneysel hatayı minimize etmektense genelleme hatasının üst sınırını minimize etmeyi amaçlayan yapısal risk minimizasyonu prensibine dayanmaktadır. Bu sayede makine öğrenme problemlerinin çözümünde geleneksel sinir ağlarından daha iyi genelleme performansı sağlamaktadır (Hua & Zhang, 2006: 1037).

Destek vektör makineleri bir çok alanda başarıyla uygulanmaktadır. Bu uygulamalara örnek verecek olursak (Byun & Lee, 2002: 213) :

- Yüz tanıma ve doğrulama
- Nesne tanıma
- El yazısı ve karakter tanıma
- Metin ve belge sınıflandırması
- Konuşma/konuşmacı tanıma ve doğrulama
- Bilgi çıkarım sistemleri
- Tahmin

3.2 İstatiksel Öğrenme Teorisi (Vapnik - Chervonenkis Teorisi)

Vladimir Vapnik ve Alexey Chervonenkis tarafından geliştirildiği için Vapnik – Chervonenkis teorisi olarak da bilinen istatiksel öğrenme teorisi denetimli öğrenme problemlerine matematiksel bir çerçeve sağlamaktadır.

Denetimli öğrenmede, makine bir dizi girdi-çıkıtlı çiftleri için verilen görevi yerine getirmek için programlanmak yerine eğitilmektedir. Burada bahsedilen eğitimin anlamı girdiler ve çıktılar arasındaki en iyi ilişkiyi ifade eden bir fonksiyonun seçilmesidir. “Seçilen fonksiyon ne kadar iyi genelleme yapmaktadır ya da daha önce keşfedilmemiş girdiler için çıktıları ne kadar iyi tahmin etmektedir?” sorusu istatiksel öğrenme teorisinin asıl sorusudur (Evgeniou vd., 2000: 421).

İstatiksel öğrenme teorisi 1960’lı yılların sonunda ortaya çıkmış ve 1990’lara kadar sadece toplanmış verilerden fonksiyon tahmini probleminin teorik analizi olarak hayata geçmiştir. 1990’lı yılların ortasında geliştirilmiş teori esas alınarak DVM adı verilen yeni öğrenme algoritma tipleri önerilmiştir. Bu durum, istatiksel öğrenme teorisini sadece teorik analiz için bir araç olmaktan çıkarmış aynı zamanda çok boyutlu fonksiyonların tahmini için kullanışlı algoritmalar üreten bir araç haline getirmiştir. (Vapnik, 1999: 988) İstatiksel öğrenme teorisi sinir ağları, kontrol sistemleri gibi çeşitli uygulamalarda kullanılan çok etkili bir araçtır. Bu teorisinin ana amacı, Monte

Carlo simülasyonu ile hesaplanmış deneysel ortalamasının yakınsaklık özelliklerini genişletmektir (Chamanbaz vd., 2014: 1617).

İstatistiksel öğrenme teorisi, deneysel risk minimizasyonu prensibini düzenlemek üzerine kurgulanmıştır. Bu teoriye göre deneysel kayıpları minimize eden yöntemlerin genelleme sınırları uzay boyutuna doğrudan bağlı değil, geçerli bir dizi fonksiyonun kapasite faktörlerine (VC entropi, büyüme fonksiyonu, VC boyutu) bağlıdır (Vapnik, 2006: 415).

İstatistiksel öğrenme teorisinin ana amacı, bir problemin sonuçlandırılması için bilgi kazanımı, tahminde bulunma, karar verme ile ya da bir veri setinden modeller kurarak probleme istatistiksel bir çerçeve sağlamaktır (Bousquet vd., 2004: 169).

Denetimli öğrenmede, X diye adlandırdığımız girdilerimiz ve Y diye adlandırdığımız çıktılarımız bulunmaktadır. İkili sınıflandırmada bu verileri $\{-1, +1\}$ şeklinde etiketleyerek tanımlayabiliriz. Her bir nesne bu iki sınıftan birine aittir. Girdi ve çıktılar arasındaki ilişkiyi bulan fonksiyon ise $f : X \rightarrow Y$ sınıflandırıcısıdır (Luxburg & Schölkopf, 2011: 653).

3.1 numaralı denklemde yer alan ve örnekleri, bilinmeyen bir $P(x,y)$ dağılım fonksiyonunu temel alan bir veri setimiz olduğunu düşünelim.

$$D = \{(x_1, y_1), \dots, (x_l, y_l)\} \in X \times Y \quad (3.1)$$

Bu öğrenme problemi beklenen kaybı minimize edilecek şekilde aşağıdaki şekilde biçimlendirilebilir. 3.2 numaralı denklemde yer alan $\mathbb{L}(\cdot)$ ifadesi kayıp fonksiyonunu ifade etmektedir. Kayıp fonksiyonu karar algoritmasının ya da sınıflandırıcının performansının ölçülmesinden bir araç olarak kullanılmaktadır. Bu fonksiyon her bir sınıflandırıcının hareketinin başka bir ifadeyle gerçekleşen değer ile tahmin değeri arasındaki farkın maliyetini ölçer (Camastra & Vinciarelli, 2010: 160).

$$R[f] = \int_{X \times Y} \mathbb{L}(y, f(x)) dP(x, y) \quad (3.2)$$

Örnekteki öğrenme problemini pratikte çözmek mümkün değildir. Beklenen riski tanımlayan olasılık dağılımı $P(x,y)$ belirsiz olduğu için 3.2 numaralı denklemde yer

alan integral hesaplama yaparak tahminde bulunamaz. Bu eksikliği ortadan kaldırmak için sahip olunan sınırlı eğitim verisinden öğrenmeyi gerçekleştirecek bir tümevarım ilkesine ihtiyaç duyulur. Vapnik (1998) tarafından geliştirilen istatistiksel öğrenme teorisi, deneysel risk minimizasyonu adı verilen tümevarım ilkesi üzerine temellendirilmiştir (Evgeniou vd., 2000, 424).

3.2.1 Deneysel Risk Minimizasyonu

Deneysel risk minimizasyonu ilkesi oldukça yaygındır. Regresyon tahminindeki en küçük kareler yöntemi ya da yoğunluk tahmini problemindeki maksimum olabilirlik yöntemi gibi belirli bir öğrenme probleminin çözümünde kullanılan klasik yöntemler deneysel risk minimizasyonu ilkesini esas almaktadır (Vapnik, 1999: 989).

Deneysel risk minimizasyonu ilkesi, deneysel riski minimize eden $f(\cdot)$ fonksiyonunun seçimine dayanmaktadır ve 3.3 numaralı denklemde yer almaktadır. Deneysel risk minimizasyonu ilkesi teorik olarak güvenilir ve tutarlıdır (Camastra & Vinciarelli, 2010: 160) .

$$R_{emp}[f] = \frac{1}{l} \sum_{i=1}^l \mathbb{L}(y_i, f(x_i)) \quad (3.3)$$

İki sınıflı bir öğrenme probleminin öğreniminde beklenen test hatası ya da beklenen risk $R[f]$, deneysel risk ise $R_{emp}[f]$ olarak ifade edilmektedir. Deneysel risk $R_{emp}[f]$, beklenen riskten $R[f]$ iki şekilde farklılaşmaktadır (Haykin, 1999: 92).

1. Bilinmeyen dağılım fonksiyonuna bağlı değildir.
2. f ağırlık vektörü göz önünde bulundurularak teoride minimize edilebilir.

Deneysel risk minimizasyonu ilkesine göre, veri setindeki eleman sayısının sonsuza yaklaşması $l \rightarrow \infty$ durumunda, olasılık dahilinde $R_{emp}[f] \rightarrow R[f]$ yakınsarsa minimum $R_{emp}[f] \rightarrow$ minimum $R[f]$ yakınsamaktadır. Bu koşulun olmaması durumunda tutarsızlık söz konusudur (Brunato & Battiti 2005, 829).

İstatiksel öğrenme teorisi her bir fonksiyon için beklenen risk ile deneysel risk arasındaki uzaklıkların olasılı sınırlarının belirlenmesini sağlar. Sınırlar fonksiyon uzayındaki l örnek sayısı ve h kapasiteyi kapsamaktadır. Teorinin en yaygın olarak kullanılan kapasite ölçütü VC boyutu (Vapnik-Chervonenkis, 1971) ya da VC boyutunun ölçek hassas versiyonlarıdır (Evgeniou vd., 2002: 424).

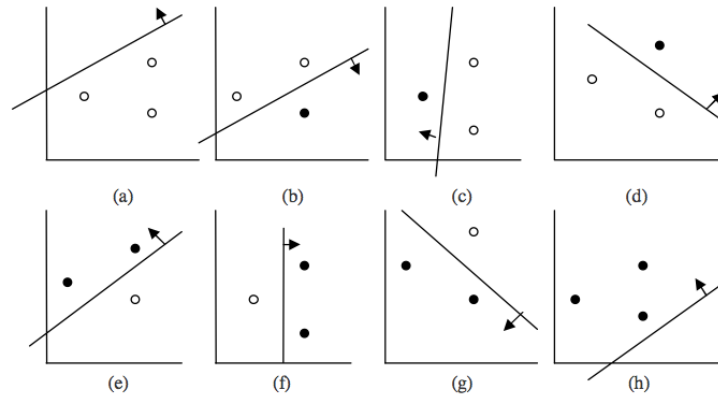
3.2.2 VC Boyutu

İstatiksel öğrenme teorisi, sınıflandırıcının karmaşıklık seviyesini VC boyutu (Vapnik-Chervonenkis dimension) adı verilen bir kapasite ölçütü ile belirlenmesini sağlar (Brunato & Battiti, 2005: 829).

l nokta içeren $C = \{i(x, \alpha)\}$ fonksiyonlar kümesinde $i(\cdot)$ değerinin $\{-1, +1\}$ olmak üzere iki değer alacağını varsayalım ve α değerinin parametre vektörü olduğunu düşünelim (Camastra & Vinciarelli, 2010: 156). l nokta 2^l kadar farklı şekilde $\{-1, +1\}$ değerleri alarak etiketlenecektir. \mathcal{H} hipotez uzayında $\{-1\}$ değerli noktaları $\{+1\}$ değerli noktalardan ayırabilen $h \in \mathcal{H}$ bulunması durumunda \mathcal{H} hipotez uzayının l noktayı parçaladığı söylenir. \mathcal{H} hipotez uzayı ile hatasız parçalanabilen en büyük örnek sayısına VC boyutu denir (Alpaydın, 2012: 20).

Şekil 9 - Üç Noktanın Bir Düzlem İle Parçalanması

Kaynak: Basak, D., Pal, S. ve Patranabis, D. C., "Support Vector Regression", **Neural Information Processing - Letters and Reviews**, C: 11, No: 10, 2007, p. 204.

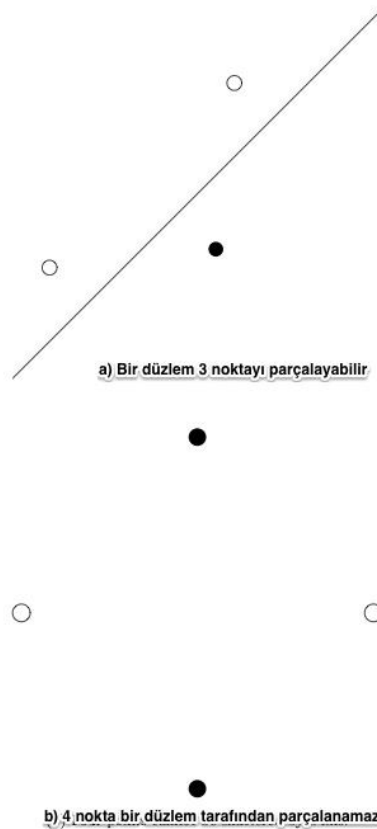


Şekil 9’da iki boyutlu bir uzayda yer alan 3 noktanın, 2^3 yani 8 farklı şekilde nasıl etiketlendiği gösterilmektedir (Basak vd., 2007: 203).

VC boyutu genellikle h harfi ile gösterilir ve eğitim kümesindeki l örnek sayını geçemez. Şekil 10’da iki boyutta (\mathbb{R}^2) üç noktanın bir düzlem tarafından parçalanabilirken, dört noktanın bir düzlem tarafından parçalanamadığı görülmektedir. İki boyutlu bir uzayda doğrusal fonksiyon kümesinin VC boyutu üçtür. Bu örneği genelleştirecek olursak, n boyuttaki bir hiper düzlemin VC boyutu $n + 1$ olmaktadır (Camastra & Vinciarelli, 2010: 157). Başka bir ifadeyle n boyutlu hiper düzlem kümesi en fazla $n + 1$ sayıda örneği parçalayabilir ve VC boyutu $n + 1$ olabilir (Abe, 2005: 74).

Şekil 10 - Doğrusal Fonksiyon Kümesinin VC Boyutu

Kaynak: Camastra, F. ve Vinciarelli, A., **Machine Learning For Audio, Image And Video Analysis**, London, Springer Science+Business Media, 2010, p.158.

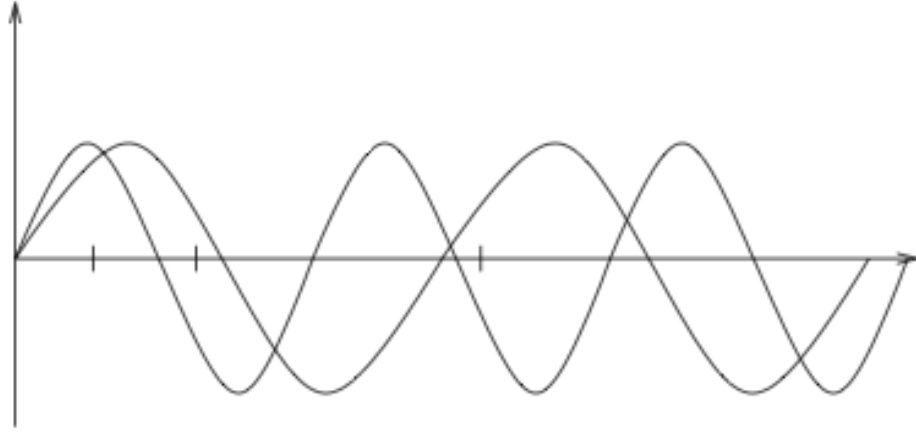


VC boyutu, fonksiyon kümesinin parametre sayısına bağlı değildir. Şekil 11’de, 3.4 numaralı denklemle ifade edilen sadece tek parametrelilik bir fonksiyon kümesi yer almaktadır. Bu fonksiyon kümesi sonsuz VC boyutuna sahiptir. Noktalar kümesi, $\sin(\alpha x)$ fonksiyon kümesi tarafından uygun bir α değeri seçilerek parçalanabilmektedir (Bousquet vd., 2004: 189).

$$\{sgn(\sin(\alpha x)) : t \in \mathbb{R}\} \quad (3.4)$$

Şekil 11 - Sinüs Dalgasının VC Boyutu

Kaynak: Bousquet, O., Boucheron, S. ve Lugosi, G., “Introduction To Statistical Learning Theory”, **Advanced Lectures on Machine Learning**, C: 3176, 2004, p. 190.



Vapnik ve Chervonenkis beklenen riskten deneysel riskin sapması üzerine sınırlar belirlemişlerdir. $1 - \eta$ olasılığını sağlayan bir sınır 3.5 numaralı denklemde yer almaktadır. Θ ifadesi bilinmeyen verinin güven aralığını (sınıflandırma tahmin hatası) ifade etmektedir (Abe, 2005: 75). Daha küçük bir beklenen risk elde etmek amacıyla logaritmik faktörler ihmal edilirse, hem deneysel risk ile hipotez uzayıyla VC boyutu arasındaki h/l oranının hem de örnek sayısının küçük olması zorunluluğu söz konusudur (Brunato & Battiti, 2005: 829).

$$R[f] \leq R_{emp}[f] + \Theta \quad (3.5)$$

Test verisinin genelleme hatası, eğitim verisinin hatasıyla tahmin hatasının toplamına eşittir. İkili bir sınıflandırma probleminde, $1 - \eta$ ($\eta > 0$) olasılığında tahmin hatası 3.6 numaralı denklemde yer almaktadır (Camastra & Vinciarelli, 2010: 157).

$$E_{est} = \sqrt{\frac{1}{l} \left(h \left(\ln \frac{2l}{h} + 1 \right) + \ln \frac{4}{\eta} \right)} \quad (3.6)$$

Veri göstergesinin sınırlı olduğu durumlarda h ile ifade edilen VC boyutunun uygun değerinin seçilmesi, iyi bir genelleme performansının sağlanması için oldukça önemlidir (Brunato & Battiti, 2005: 829).

3.2.3 Yapısal Risk Minimizasyonu

Yapısal risk minimizasyonu, Vapnik tarafından önerilen VC boyutunu temel alan model seçim kriteridir (Camastra & Vinciarelli, 2010: 159). Bir denetimli öğrenme probleminin çözümünde başa çıkılması gereken sorun, makine kapasitesiyle problemdeki mevcut eğitim verisini eşleştirerek en iyi genelleme performansını sağlamaktır. Yapısal risk minimizasyonu, tümevarımsal bir yöntem olarak, bu amaca ulaşabilmek için öğrenme makinesinin VC boyutunu bir kontrol değişkeni olarak kullanır (Haykin, 1999: 101).

Başka bir ifadeyle yapısal risk minimizasyonu ilkesi 3.5 numaralı denklemde yer alan eşitsizliğin sağ tarafını minimize ederek sınıflandırıcının genelleme yeteneğini kontrol etmeyi ($R[f]$ beklenen riski minimize etmek) amaçlamaktadır. Bunu gerçekleştirmek için, sınıflandırıcının VC boyutu ile deneysel risk arasında ödünleşimin gerçekleşmesi gerekir (Fumera & Roli, 2002: 74).

Yapısal risk minimizasyonu yöntemi, hipotez uzaylarının iç içe geçmiş bir dizisini $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots \subset \mathcal{H}_n \subset \dots$ belirlemektedir. Her bir hipotez uzayı \mathcal{H}_n , sonlu bir kapasiteye h_n sahiptir ve önceki dizilerden daha büyüktür $h_1 \leq h_2, \dots, \leq h_n$. Örnek vermek gerekirse \mathcal{H}_n , n dereceli bir polinom dizisi ya da daha karmaşık, doğrusal olmayan bir parametreleştirme olabilir (Evgeniou vd., 2002: 425).

Yapısal risk minimizasyonu yöntemin diğer adımları aşağıdaki şekildedir. (Haykin, 1999: 101)

- Her bir sınıflandırıcı için deneysel risk (eđitim hatası) minimize edilir.
- En küçük güvenceli riske (genelleme hatasının sınırı) sahip olan sınıflandırıcı tanımlanır. Bu belirlenmiş sınıflandırıcı, birbiriyle rekabet içinde olan eđitim hatası ile güven aralığı arasındaki en iyi uzlaşmayı sağlar.

Logaritmik faktörler göz ardı edilecek olursa problem aşağıdaki şekilde çözülmektedir. (Brunato & Battiti, 2005: 829)

$$\min_{\mathcal{H}_n} \left(R_{emp}[f] + \sqrt{\frac{h(n)}{l}} \right) \quad (3.7)$$

Yapısal risk minimizasyonu yöntemi, genelleme hatasını minimize eden modelin seçilmesini sağlamaktadır. Fakat genelleme hatasının sınırı sıklıkla çok gevşektir ve her zaman fonksiyon kümesinin VC boyutunu hesaplamak mümkün olmamaktadır. Buna karşılık, VC boyutu için sadece bir üst sınır hesaplanabilmektedir. Bu nedenden dolayı, yapısal risk minimizasyonu bir model seçim kriteri olarak genellikle kesin olmayan, gevşek bir kriterdir (Camastra & Vinciarelli, 2010: 159).

3.3 Sınıflandırma İçin Destek Vektör Makineleri

Sınıflandırma, destek vektör makinelerinin genel uygulama alanlarından biridir. Sınıflandırma probleminde nesnelere sınıf adı verilen belirli kategorilere göre sınıflandırılmaktadır. Sınıflandırıcının kullandığı girdiler her bir sınıfı en iyi şekilde temsil eden özellikleri ifade etmektedir.

Sınıflandırma problemlerinin çözümünde parametrik ve parametrik olmayan yaklaşımlar olmak üzere iki türlü yaklaşım söz konusudur. Sinir ađları, bulanık sistemler ve destek vektör makineleri parametrik olmayan sınıflandırıcılardır. Eđitim sırasında girdi-çıkı eşlerini kullanarak, sınıflandırıcılar bir girdi verisini bir sınıfa atayan karar fonksiyonlarını elde ederler (Abe, 2005: 3).

Bir sınıflandırma probleminin destek vektör makineleriyle çözümünde diđer yöntemlere göre daha iyi genelleme performansı sağladığı, optimizasyon asıllı

olduğundan dolayı hesaplamadaki karmaşıklık ve kullanım açısından daha başarılı sonuçlar verdiği görülmüştür (Ayhan & Erdoğan, 2014: 176). Destek vektör makineleri iki sınıflı ve çok sınıflı problemlerde kullanılmaktadır.

3.3.1 İki Sınıflı Problemler için Destek Vektör Makineleri

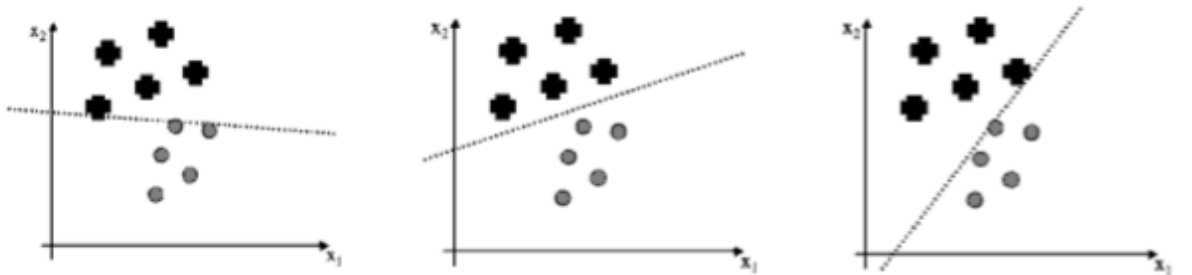
Bu bölümde öncelikle iki sınıflı sınıflandırma problemi ve iki sınıflı bir sınıflandırma problemi için en iyi ayıran hiper düzlemin nasıl bulunacağı anlatılacaktır. Daha sonra girdi uzayında doğrusal ayrılabilir sert kenarlı ve doğrusal ayrılabilir yumuşak kenarlı destek vektör makineleri ile Kernel yöntemlerine yer verilecektir.

3.3.1.1 En İyi Ayıran Hiper Düzlem

Destek vektör makinesinin asıl problemi iki sınıflı bir sınıflandırma problemi için en iyi ayıran hiper düzlemin bulunmasıdır. Destek vektör makinesi iki sınıf arasındaki en geniş kenar boşluğunu bularak (sınıfların sınırları arasındaki uzaklıkları ölçerek), kenar boşluğundaki bir yere en iyi ayıran hiper düzlemi yerleştirir (Navarrete & Solar, 2002: 30).

Şekil 12 - İki Sınıflı Ayıran Hiper Düzlemler

Kaynak: Marsland, S., *Machine Learning An Algorithmic Perspective*, USA, A Chapman & Hall Book CRC Press, 2009, p. 120.



Şekil 12’de basit bir sınıflandırma probleminde iki sınıfın üç farklı şekilde nasıl ayrıldığı gösterilmektedir. Her bir hiper düzlem iki sınıfı birbirinden ayırmış ve kısmen doğru ayırım yapılmıştır. Hangi ayırımın daha iyi olduğu sorusuna ilk bakışta vereceğimiz cevap ortadaki olur.

Şeklin sağında ve solunda yer alan ayırımlarda hiper düzlemler örnekleme yer alan noktalara çok yakın olduğu için bu noktaların hiper düzlemin yanlış tarafında olma ihtimali çok yüksektir. Fakat ortadaki ayırımda böyle bir problem yoktur. Böyle bir problemde en iyi ayıran hiperdüzlemin hesaplanması gerekir (Marsland, 2009: 121).

3.3.1.2 Doğrusal Ayrılabilir Sert Kenarlı Destek Vektör Makineleri

Eğitim setinde yer alan verilerin doğrusal olarak ayrıldığı iki sınıflı bir sınıflandırma problemini göz önünde bulunduralım ve bu iki sınıfın $\{-1, +1\}$ değerleri olarak etiketleneceğini düşünelim. $\{(x_i, y_i)\}_{i=1}^N$ örnekleminde x_i girdiyi, y_i ise çıktıyı temsil etmektedir. $x_i \in C_1$ olduğunda $y_i = +1$, $x_i \in C_2$ olduğunda $y_i = -1$ değerini aldığını düşünelim. Bu iki sınıf arasındaki en geniş kenar boşluklarını bularak en iyi hiper düzlem aşağıdaki şekilde ifade edilir (Haykin, 1999: 320).

$$w^T x + b = 0 \quad w, x \in \mathbb{R}^n, b \in \mathbb{R} \quad (3.8)$$

x girdi vektörünü, w ayarlanabilir ağırlık vektörünü ve b eşik değeri simgeler ve aşağıdaki koşullar sağlanmalıdır (Vapnik, 1999: 996).

$$w^T x_i + b \geq 1 \quad y_i = +1 \text{ için} \quad (3.9)$$

$$w^T x_i + b \leq -1 \quad y_i = -1 \text{ için} \quad (3.10)$$

Yukarıdaki koşullar 3.11 numaralı denklemde tek bir satırda yer almaktadır. En iyi ayıran hiper düzlem belirlenirken kenar boşluğunu mümkün olduğunca büyütmek için vektör boyunu $\|w\|$ mümkün olduğunca küçültmek ($\min \frac{1}{2} \|w\|^2$) istenir.

$$y_i [(w^T x_i) + b] \geq 1 \quad i = 1, \dots, l \quad (3.11)$$

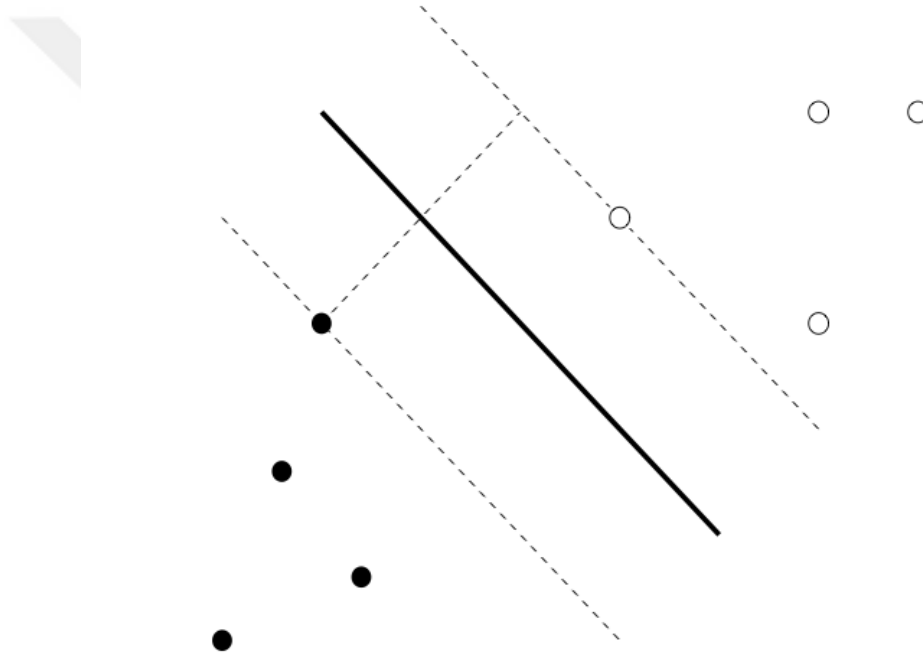
Şekil 13'de iki farklı sınıf siyah ve beyaz noktalar olarak gösterilmiştir. Siyah ve beyaz noktanın üzerinden geçen kesik çizgiler destek vektörleridir. İki destek vektörü arasındaki uzaklığa ise kenar boşluğu denilmektedir. Kenar boşluğunun mümkün olduğunca büyük olması istenir. En iyi ayıran hiper düzlem, kenar boşluğunu en büyük yapan hiper düzlemdir. En iyi ayıran hiper düzlem kalın çizgiyle gösterilmiştir ve iki

sınıf arasındaki en kısa çizgiyi dik keserek iki sınıfı ortadan ayırmaktadır (Alpaydın, 2012: 263) (Camastra & Vinciarelli, 2010: 217).

Daha iyi bir genelleme performansı sağlayabilmek için noktaların, hiper düzlemin yanında değil belli bir mesafede kalması istenir (Alpaydın, 2012: 263).

Şekil 13 - En İyi Ayıran Hiper Düzlem

Kaynak: Camastra, F. ve Vinciarelli, A., **Machine Learning For Audio, Image And Video Analysis**, London, Springer Science+Business Media, 2010, p.217.



Böyle bir problem doğrusal eşitsizlik kısıtları olan kareli, dış bükey bir optimizasyon problemidir. Kısıtları olan bu minimizasyon problemi, dual optimizasyon problemi olarak sonuçlanan Lagrange çarpanları α_i kullanılarak çözülmektedir (Hastie vd., 2009: 420). Lagrange dual probleminin çözümünde ise Karush-Kuhn-Tucker koşulları kullanılır.

α_i Lagrange çarpanlarının kullanıldığı, kısıtlanmamış şekilde yeniden yazılan optimizasyon problemi aşağıdaki yer almaktadır (Alpaydın, 2012: 264).

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^1 \alpha_i \{[(x_i w^T) + b]y_i - 1\} \quad (3.12)$$

Lagrange fonksiyonu w, b değerlerine göre minimum ve $\alpha_i \geq 0$ değişkenlerine göre maksimum olmalıdır. Eyer noktasında, w_0, b_0 çözümleri ve α_0 aşağıdaki koşulları sağlamalıdır ve eyer noktası problemin çözümüne ulaştırmaktadır (Vapnik, 1999: 996).

$$\frac{\partial L(w_0, b_0, \alpha^0)}{\partial b} = 0 \quad (3.13)$$

$$\frac{\partial L(w_0, b_0, \alpha^0)}{\partial w} = 0 \quad (3.14)$$

Bu eşitlikleri açık bir şekilde yazacak olursak:

- En iyi hiper düzlem için α_i^0 katsayıları aşağıdaki kısıtları sağlamalıdır.

$$\sum_{i=1}^l \alpha_i^0 y_i = 0, \quad \alpha_i^0 \geq 0 \quad i = 1, \dots, l \quad (3.15)$$

- En iyi hiper düzlemin parametreleri eğitim seti vektörlerinin doğrusal kombinasyonudur.

$$w_0 = \sum_{i=1}^l \alpha_i^0 x_i y_i, \quad \alpha_i^0 \geq 0 \quad i = 1, \dots, l \quad (3.16)$$

- Çözüm aşağıdaki koşulları sağlamalıdır. 3.17 numaraları denklemden verilen eşitsizlik kısıtları ve onların Lagrange çarpanları arasındaki ilişki “*Karush-Kuhn-Tucker tamamlayıcılık koşulları*” ya da “*Karush-Kuhn-Tucker (KKT) koşulları*” olarak bilinir (Abe, 2005: 18).

$$\alpha_i^0 \{[(x_i w_0) - b_0] y_i - 1\} = 0 \quad i = 1, \dots, l \quad (3.17)$$

Primal problem dış bükey maliyet fonksiyonu ve doğrusal kısıtlarla ilgilenmektedir. Bu şekilde kısıtlandırılmış bir optimizasyon probleminde dual problem adı verilen başka bir problem kurmak mümkündür. Dual problem optimal çözümün sağlanması için Lagrange çarpanları kullanılarak primal problemle aynı optimal sonucu sağlamaktadır (Haykin, 1999: 324)

Yukarıda bahsedilen Karush-Kuhn-Tucker koşulları denklem 3.12’de yerine koyulacak olursa 3.18 numaralı denklem elde edilir (Alpaydın, 2012: 264).

$$L(\alpha) = -\frac{1}{2} \sum_I \sum_J \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{i=1}^l \alpha_i \quad (3.18)$$

Bu fonksiyon maksimize edilirken aşağıdaki kısıtlar göz önünde bulundurulmalıdır.

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad ve \quad \alpha_i \geq 0, \quad i = 1, \dots, l \quad (3.19)$$

3.12 numaralı denklemden, $\alpha_i = 0$ ya da $\alpha_i \neq 0$ ve $y_i(w^T x_i + b) = 1$ koşullarını sağlamalıdır. $\alpha_i \neq 0$ olan x_i eğitim verisi “*destek vektörleri*” olarak adlandırılır (Abe, 2005: 19).

Formüle edilen destek vektör makinesi sert kenarlı destek vektör makinesi olarak tanımlanır.

3.3.1.3 Doğrusal Ayrılmaz Yumuşak Kenarlı Destek Vektör Makineleri

Eğitimin setinde yer alan verilerin doğrusal olarak ayrılabilmesi durumunda iki sınıfı hatasız olarak ayırabilen hiper düzlem bulunmamaktadır. Böyle bir durumda minimum hata yapan hiper düzlem bulunmak istenir (Alpaydın, 2012: 266).

Gerçek dünya problemlerinde gürültünün varlığından dolayı bazı yanlış etiketlenmiş örnekler olabilir ve sınıflar kısmi olarak üst üste çıkışabilir. Bu sebeple bazı örneklerin bu kuralı bozma ihtimaline izin vermek gerekir. Bunu sağlayabilmek için aşağıda gösterilen değişkenler tanımlanır. Örnek x_i ’nin 3.11 numaralı denklemi ihlal etmesi durumunda ξ_i artık değişkeni kesinlikle pozitifdir aksi halde negatiftir (Camastra & Vinciarelli, 2010: 221).

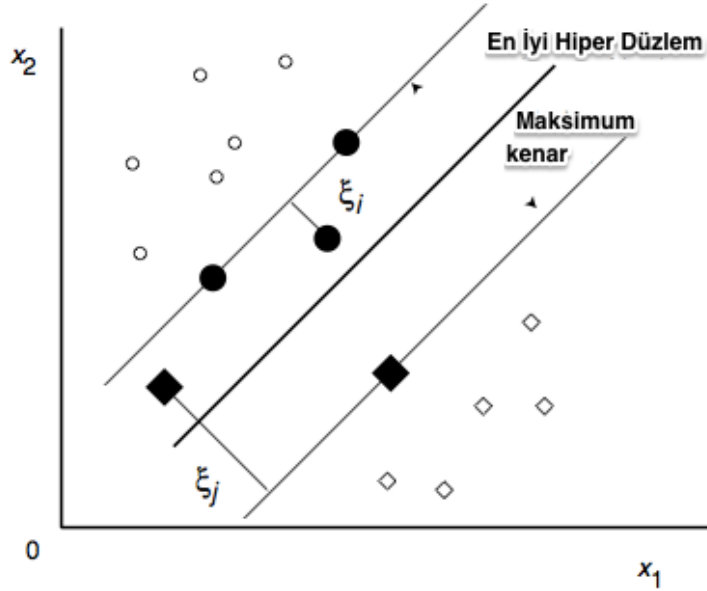
$$\xi_i \geq 0 \quad i = 1, \dots, l \quad (3.20)$$

Artık deęişkenleri kullanarak kısıtları gevşetebiliriz. Artık deęişkenleri 3.11 numaralı denkleme eklediğimizde aşağıdaki şekilde yazabiliriz.

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad (3.21)$$

Şekil 14 - İki Boyutlu Uzayda Doğrusal Ayrılama Durumu

Kaynak: Abe, S., **Support Vector Machines For Pattern Classification**, London, Springer Science + Business Media, 2005, p.22.



Eđitim verisi x_i için eđer $0 \leq \xi_i \leq 1$ ise maksimum kenara sahip deęildir fakat hala doęru sınıflandırılmaktadır. Fakat $\xi_i \geq 1$ ise veri en iyi hiper düzlem tarafından yanlış sınıflandırılmaktadır. En iyi hiper düzlemi elde edebilmek için maksimum kenara sahip olmayan eđitim verisi sayısı minimum olmalı, $Q(w) = \sum_{i=1}^l \xi_i$ minimize edilmelidir. Böyle bir tümleşik problemin çözümlü zordur. Bunun yerine 3.21 numaralı denklemden yer alan kısıtları baęlı olarak aşağıdaki amaç fonksiyonu minimize edilmelidir (Abe, 2005: 22).

$$L(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (3.22)$$

3.22 numaralı denklemdeki $C \geq 0$ parametresi düzgünleştirme sabiti olarak ifade edilmektedir ve hata sayısı ve hiper düzlemin arasındaki ödünleşimin yönetilmesine olanak sağlamaktadır. Başka bir ifadeyle C , kenarın maksimizasyonu ile sınıflandırma hatasının minimizasyonu arasındaki ödünleşimi belirleyen kenar parametresidir. C parametresinin büyük bir değer alması hatalara atanan cezanın daha büyük olmasına sebep olmaktadır (Camastra & Vinciarelli, 2010: 221).

Yeni kısıtlar 3.12 numaralı denkleme eklenip yeniden yazıldığında aşağıdaki denklem elde edilir (Mohri vd., 2012: 73). β_i değeri, ξ_i değerlerinin negatif olmamasını sağlayan yeni Lagrange parametrelerini ifade etmektedir (Alpaydın, 2012: 266).

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i \{[(x_i w^T) + b]y_i - 1 + \xi_i\} - \sum_{i=1}^l \beta_i \xi_i \quad (3.23)$$

Optimal çözüm için Karush-Kuhn-Tucker koşulları sağlanmalıdır (Abe, 2005: 23).

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = \sum_{i=1}^l \alpha_i y_i = 0 \quad (3.24)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = w - \sum_{i=1}^l \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i \quad (3.25)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi} = C - \alpha_i - \beta_i = 0 \quad (3.26)$$

$$\alpha_i (y_i (w^T x_i + b) \geq 1 - \xi_i) = 0 \quad i = 1, \dots, l \quad (3.27)$$

$$\beta_i \xi_i = 0 \quad i = 1, \dots, l \quad (3.28)$$

$$\alpha_i \geq 0, \quad \beta_i \geq 0, \quad \xi_i \geq 0 \quad i = 1, \dots, l \quad (3.29)$$

Yukarıda yer alan eşitlikler Lagrange fonksiyonunda yerine koyulduklarında 3.30 numaralı denklemde yer alan dual problem elde edilecektir (Haykin, 1999: 328).

$$\max_{\alpha} -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{i=1}^l \alpha_i \quad (3.30)$$

Kısıtlar:

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad \text{ve} \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \quad (3.31)$$

3.30'da yer alan problem çözüldüğünde sınırın doğru tarafında ve yeterli uzaklıkta bulunan örnekler için $\alpha_i = 0$ değerini alır. $\alpha_i \geq 0$ olan örnekler destek vektörleridir ve 3.25 numaralı denklemdeki gibi w ağırlık vektörünü ifade eder. $\alpha_i < C$ olan örnekler kenarın üzerinde yer alır, $\xi_i = 0$ olur ve b değerini hesaplamak için kullanılır. $\alpha_i = 0$ olan örnekler ise kenar boşluğunun içinde kalarak yanlış sınıflandırılan örneklerdir (Alpaydın, 2012: 268).

Doğrusal ayrılabilir durumdan tek farkı α_i Lagrange çarpanlarındaki üst sınır C 'dir. α_i , C 'yi geçemez. Sınıflandırma fonksiyonu ise doğrusal ayrılabilir sert kenarlı destek vektör makinelerinin sınıflandırma fonksiyonuyla aynıdır (Mohri vd., 2012: 74).

$$f(x) = \text{sign} \left(\sum_{i=1}^l y_i \alpha_i \langle x, x_i \rangle + b \right) \quad (3.32)$$

3.3.1.4 Kernel Yöntemleri

Bir problemde yer alan veriler doğrusal değilse, en iyi hiper düzlemin doğrusal bir model ile belirlenememesi durumunda veri seti Kerneller ya da Kernel fonksiyonları adı verilen fonksiyonları temel alan Kernel yöntemleri ile yüksek boyutlu bir özellik uzayında doğrusal olarak modellenip verilerin ayrılabilmesi sağlanır (Mohri vd., 2012: 89).

n boyutlu bir girdi uzayında d dereceden bir polinom oluşturabilmek için, en iyi hiper düzlemin oluşturulduğu özellik uzayının $O(n^d)$ boyutlu olması gerekir. Örnek vermek gerekirse, 4. ya da 5. dereceden bir polinomu 200 boyutlu bir uzayda oluşturabilmek için milyar boyutlu bir özellik uzayında hiper düzlemler oluşturmak gereklidir. Kernel

ile girdi uzayında, özellik uzayındaki en iyi hiper düzleme eş değer sonuçları oluşturmak mümkündür (Vapnik, 1999: 997).

Şekil 15’de doğrusal olmayan bir karar sınırı belirlemek için X girdi uzayından, doğrusal ayrılmanın mümkün olduğu daha yüksek boyutlu Z özellik uzayına doğrusal olmayan eşleştirme Φ yapıldığı görülmektedir (Schölkopf & Smola, 2002: 29).

Şekilde iki boyutlu (x_1, x_2) bir girdi uzayı ve üç boyutlu (Z_1, Z_2, Z_3) bir özellik uzayı bulunmaktadır. İki boyutlu girdi uzayını üç boyutlu özellik uzayına eşleştirme yapan fonksiyon aşağıda yer almaktadır.

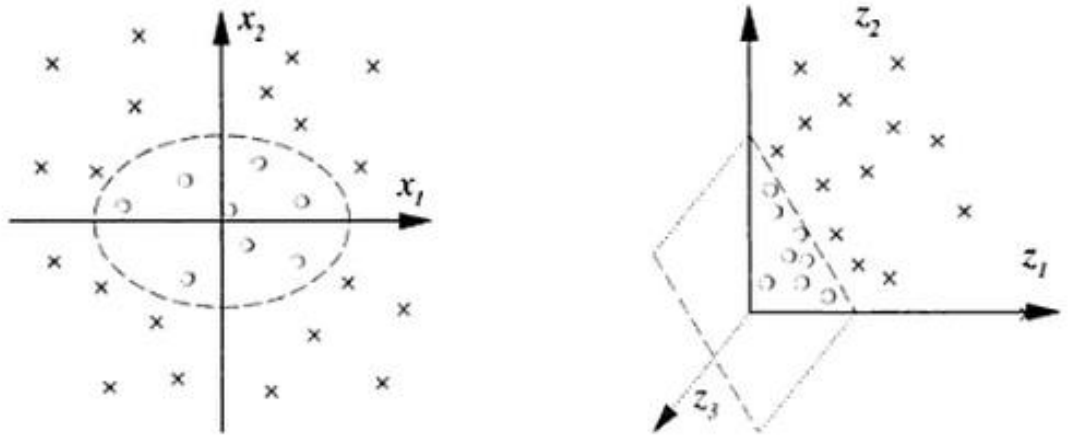
$$\Phi_2(x) = (Z_1, Z_2, Z_3) = x_1^2, x_2^2, \sqrt{2}x_1x_2 \quad (3.33)$$

Girdi uzayının özellik uzayına eşleştirildiğinde oluşan iç çarpım ise aşağıdaki şekildedir.

$$\Phi(x) \cdot \Phi(x') = x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1x_2 x_1'x_2' + x_1^2 = (x, x')^2 \quad (3.34)$$

Şekil 15 - Kernel Fonksiyonu

Kaynak: Schölkopf, B. ve Smola, A. J., **Learning With Kernels: Support Vector Machines, Regularization, Optimization, And Beyond**, Cambridge, The MIT Press, 2002, p.29.



3.3.1.4.1 Çekirdek (Kernel) Hilesi ve Mercer Koşulu

Destek vektör makinelerinde, en iyi hiper düzlem genelleme yeteneğini maksimize etmek için belirlenir. Eğer eğitim verisi doğrusal olarak ayrılamıyorsa, hiper düzlem

en iyi şekilde belirlense bile elde edilen sınıflandırıcı yüksek genelleme yeteneğine sahip olamaz. Bu sebeple, doğrusal ayrılabilirliği arttırmak için orijinal girdi uzayı, özellik uzayı adı verilen daha yüksek boyutlu nokta çarpımı uzayına eşleştirilir (Abe, 2005: 25).

Doğrusal olmayan sınıflandırıcılar x girdi verisini daha yüksek boyutlu $\Phi(x)$ vektörüne eşleştirme yaparak dönüştürülmüş uzayda doğrusal sınıflandırmayı kullanır ve sınıflandırma fonksiyonu aşağıda yer almaktadır (Brunato & Battiti, 2005: 831).

$$f(x) = \text{sign} \left(\sum_{i=1}^l y_i \alpha_i \Phi(x) \cdot \Phi(x_i) + b \right) \quad (3.35)$$

Kernel fonksiyonu $K(x, x_i) \equiv \Phi^T(x) \cdot \Phi(x_i)$ ile birlikte sınıflandırma fonksiyonu, denklem 3.36'daki gibi ve kareli optimizasyon problemi ise 3.37'deki gibi olacaktır (Haykin, 1999: 331).

$$f(x) = \text{sign} \left(\sum_{i=1}^l y_i \alpha_i K(x, x_i) + b \right) \quad (3.36)$$

$$\max_{\alpha} -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i x_j) + \sum_{i=1}^l \alpha_i \quad (3.37)$$

Kısıtlar:

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad \text{ve} \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \quad (3.38)$$

Kernel fonksiyonu kullanılarak aynı şekilde en iyi ayıran hiper düzlem de aşağıdaki şekilde tanımlanabilir (Alpaydın, 2012: 271).

$$\sum_{i=1}^l \alpha_i y_i K(x, x_i) \quad (3.39)$$

Özetlemek gerekirse, çok boyutlu nitelik uzayında olması gereken $\Phi^T(x) \cdot \Phi(x_i)$ iç çarpımı, girdi uzayında yer alan örnekler üzerinde $K(x, x_i)$ Kernel fonksiyonu

kullanılarak yapılmaktadır. Başka bir ifadeyle, x, x_i girdileri Z nitelik uzayına eşleştirme yapıldıktan sonra iç çarpımı orada yapmak yerine, Kernel fonksiyonu doğrudan girdi uzayında uygulanmaktadır. Bu yaklaşım *Kernel hilesi* olarak tanımlanır.

Kernel fonksiyonları *Mercer koşulunu* sağlamalıdır yani pozitif tanımlı simetrik ve sürekli fonksiyonlar olmalıdırlar. Mercer koşulunu açıklayacak olursak: $X \subset \mathbb{R}^N$ ve $K: X \times X \rightarrow \mathbb{R}$ sürekli ve simetrik bir fonksiyon olsun. Eğer herhangi bir integrallenebilir fonksiyonun karesi $c (c \in L_2(x))$ 3.40'da yer alan koşulu sağlarsa K , 3.41'de yer alan formun düzgünce yakınsak yayılımını kabul eder. Bu koşul DVM gibi algoritmalar için optimizasyon probleminin dış bükeyliğini garanti etmek için önemlidir ve bu sayede yakınsaklık garanti edilir (Mohri vd., 2012: 91).

$$\iint_{X \times X} c(x)c(x') K(x, x') dx dx' \geq 0 \quad (3.40)$$

$$K(x, x') = \sum_{n=0}^{\infty} a_n \Phi_n(x) \Phi_n(x'), \quad a_n > 0 \quad (3.41)$$

3.3.1.4.2 Kernel Fonksiyon Çeşitleri

Destek vektör makinelerinde genellikle kullanılan Kernel fonksiyon çeşitleri aşağıda yer almaktadır (Marsland, 2009: 127).

1. Doğrusal Kernel
2. Çokterimli (Polinom) Kernel
3. Dairesel Tabanlı (Gaussian) Kernel
4. Sigmoid Kernel

Doğrusal Kernel: Eğer bir sınıflandırma problemi girdi uzayında doğrusal ayrılabiliriyorsa, girdi uzayını yüksek boyutlu bir uzaya eşleştirme yapmak zorunda kalmayız. Böyle bir durumda doğrusal Kernel kullanılır (Abe, 2005: 27).

$$K(x, x') = x^T x' \quad (3.42)$$

Çokterimli (Polinom) Kernel: Çokterimli Kernel d derece için aşağıdaki şekilde yazılmaktadır.

$$K(x, x') = (x^T x' + 1)^d \quad (3.43)$$

$d = 1$ olduğunda Kernel doğrusaldır. d değerinin önceden saptanması gereklidir. Örnek vermek gerekirse $d = 2$ ve $m = 2$ değerleri için 3.44'de ifade edilen Kernel, 3.45'de yer alan taban fonksiyonunun iç çarpımına eşittir (Alpaydın 2012, 272).

$$\begin{aligned} K(x, x') &= (x^T x' + 1)^2 \\ &= (x_1 x_1' + x_2 x_2' + 1)^2 \\ &= 1 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_2 x_1' x_2' + x_1^2 x_1'^2 + x_2^2 x_2'^2 \end{aligned} \quad (3.44)$$

$$\Phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2 x_2^2]^T \quad (3.45)$$

Dairesel Tabanlı (Gaussian) Kernel: Gaussian Kernel ya da daireysel tabanlı Kernel aşağıdaki şekilde yazılmaktadır (Mohri vd., 2012: 94).

$$K(x, x') = \exp\left(-\frac{\|x' - x\|^2}{2\sigma^2}\right) \quad (3.46)$$

Uygulamalarda en çok kullanılan Kernel çeşitlerinden biri olan Gaussian Kerneller, $n \geq 0$ dereceden çokterimli Kernellerin pozitif doğrusal kombinasyonlarıdır.

Sigmoid Kernel: $a, b > 0$ sabitleri için Sigmoid Kernel aşağıdaki şekilde tanımlanmaktadır.

$$K(x, x') = \tanh(a(x^T x') + b) \quad (3.47)$$

Destek vektör makinelerinde kullanılan Sigmoid Kerneli basit sinir ağlarının temelindeki öğrenme algoritmalarıyla yakından ilgili bir algoritmayı yönlendirmektedir.

3.3.2 Çok Sınıflı Problemler İçin Destek Vektör Makineleri

Destek vektör makineleri genel olarak iki sınıflı problemler için formüle edilmiştir. Fakat gerçek dünya problemlerinde sınıflar ikiden fazla $n > 2$ olabilir. Çok sınıflı problemler için kullanılan yöntemler aşağıda yer almaktadır (Camastra & Vinciarelli, 2010: 228).

1. Bire Tüm (One versus All) Yöntemi
2. Bire Bir (One versus One) Yöntemi
3. Diğer Yöntemler

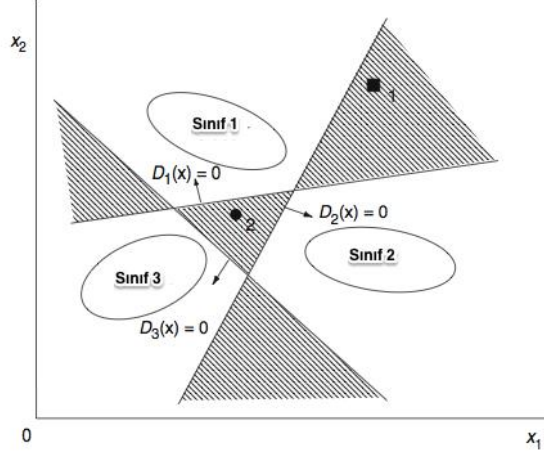
Bire Tüm (One versus All) Yöntemi: Bire tüm destek vektör makinelerinde n sınıflı problem n tane iki sınıflı probleme dönüştürülür ve her i . iki sınıflı problem için, sınıf i öteki sınıflardan ayrılır. Fakat bu formülasyon ile eğer kesikli karar fonksiyonları kullanılırsa sınıflandırılmayan bölgeler meydana gelir. Eğer kesikli karar fonksiyonları yerine sürekli karar fonksiyonları kullanılırsa sınıflandırılmayan bölgeler çözüme ulaşır (Abe, 2005: 83). Şekil 16'da sınıflandırılmayan bölgeler gösterilmektedir

Bire tüm yöntemi, j . sınıflandırıcının öğrenimi sırasında tüm eğitim verilerini $y_i = j$ ise +1 olarak ve $y_i \neq j$ ise -1 olarak etiketleyerek n destek vektör makinesi sınıflandırıcısı D_j 'nin öğrenimini içermektedir ve karar fonksiyonu aşağıdaki şekildedir (Camastra & Vinciarelli, 2010: 228).

$$f(x) = \arg \max_{j \in Y} D_j(x) \quad (3.48)$$

Şekil 16 - Bire Tüm Yönteminde Sınıflandırılmayan Bölgeler

Kaynak: Abe, S., **Support Vector Machines For Pattern Classification**, London, Springer Science + Business Media, 2005, p.85.



Bire Bir (One versus One) Yöntemi: Bire bir yöntemi, her bir ayrı sınıf çifti için $(j, j') \in Y^2$, $j \neq j'$, ikili sınıflandırıcı j ve j' sınıfları arasında $f_{jj'}: X \rightarrow \{-1, 1\}$ olarak ayıran bir öğrenimi içermektedir. n sınıflı problem, tüm ikili sınıfları içerecek şekilde $n(n-1)/2$ tane iki sınıflı probleme dönüştürülmektedir. Her $(j, j') \in Y^2$ için $f_{jj'}$, j ve j' olarak etiketlenmiş noktaları içeren alt örnekler üzerinde ikili sınıflandırma algoritmasının öğrenimiyle elde edilmektedir. +1 değerli örnekler sınıf j' ve -1 değerli örnekler sınıf j olarak eğitilir ve karar fonksiyonu aşağıdaki şekildedir (Mohri vd., 2012: 199).

$$f(x) = \arg \max_{j'=Y} |\{j: f_{jj'}(x) = 1\}| \quad (3.49)$$

Diğer Yöntemler: Bire tüm ve bire bir yöntemlerinin dışında çok sınıflı problemler için birleştirilmiş birçok yöntem de yer almaktadır. Yönlendirilmiş devirsiz grafik ve tenis turnuvası yöntemi örnek olarak gösterebiliriz. Yönlendirilmiş devirsiz grafik destek vektör makinesi, ardarda gelen ikili sınıflandırmaların yönlendirilmiş devirsiz grafiklerin meydana gelmesini içermektedir. Bu şekilde bir sınıf hiyerarşisi oluşturulmaktadır. Tenis turnuvası yöntemi ise her düğümün ikili sınıflandırıcı destek vektör makinesi olduğu yerde ikili bir karar ağacı oluşturur (Camastra & Vinciarelli, 2010: 229).

3.4 Regresyon için Destek Vektör Makineleri

Sınıflandırmanın yanında regresyon, destek vektör makinelerinin genel uygulama alanlarından biridir. Sınıflandırmada çıktı y sadece iki değer ($y \in \{\pm 1\}$) alırken, regresyonda çıktı gerçek bir değeri ($y \in \mathbb{R}$) ifade etmektedir. Regresyonun temelinde yatan fikir, gerçek değerlerin belirli bir uzaklığı içerisindeki hataları göz ardı eden kayıp fonksiyonunun belirlenmesidir (Camastra & Vinciarelli, 2010: 229).

x_i girdi vektörü, y_i çıktı değeri ve n toplam örneklem sayısı olmak üzere $\{(x_1, y_1), \dots, (x_l, y_l)\} \in \mathbb{R}^n \times \mathbb{R}$ eğitim setimiz olduğunu düşünelim. x_i ve $f(x_i)$ arasındaki ilişkiyi tanımlayan regresyon modeli w ağırlık vektörü, b eşik seviyesi olmak üzere 3.50 numaralı eşitlikteki gibi gösterilmektedir. Buna ek olarak w ve b katsayıları ise 3.51 numaralı denklemde gösterilen doğrusal optimizasyon problemi tarafından tahmin edilmektedir (Wang vd., 2014: 1440) (Yu vd., 2013: 1057).

$$f(x_i) = w^T x_i + b \quad (3.50)$$

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (3.51)$$

Kısıtlar:

$$((w^T x_i) + b) - y_i \leq \varepsilon + \xi_i \quad i = 1, \dots, l \quad (3.52)$$

$$y_i - ((w^T x_i) + b) \leq \varepsilon + \xi_i^* \quad i = 1, \dots, l \quad (3.53)$$

$$\xi_i \geq 0 \quad i = 1, \dots, l \quad (3.54)$$

$$\xi_i^* \geq 0 \quad i = 1, \dots, l \quad (3.55)$$

ξ_i hatalı örnekler için tanımlanan artık değişkeni, C ceza katsayısını, ε ise duyarsız kayıp fonksiyonunu ifade etmektedir. Hatalar artı ve eksi olmak üzere iki yönlü olabileceği için artık değişkenler ξ_i ve ξ_i^* olarak tanımlanmaktadır ve kısıtlar da bu doğrultuda yazılmaktadır (Moura vd., 2011: 1529) (Alpaydın, 2012: 279).

Destek vektör makinesinde eğitim hatası, ε duyarsız kayıp fonksiyonunun ortalaması tarafından aşağıdaki şekilde belirlenir.

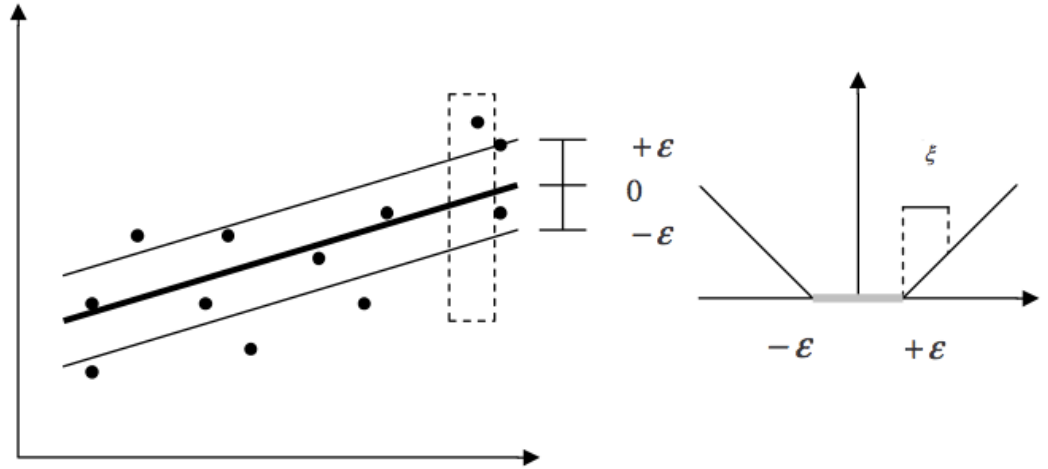
$$e(x, y, f) = \max(0, |y - f(x)| - \varepsilon) \quad (3.56)$$

Eğer tahmin değeri $f(x)$, ve gözlemlenen değer y arasındaki fark ε duyarsız kayıp fonksiyonundan küçük ise 3.56 numaralı denklem kullanılır. Aksi halde 3.56 numaralı denklem, $(f(x) - y)$ ile ε arasındaki mutlak farka eşittir (Moura vd., 2011: 1529).

ε - Duyarsız Kayıp Fonksiyonu Şekil 17’de gösterilmektedir.

Şekil 17 - ε - Duyarsız Kayıp Fonksiyonu

Kaynak: Basak, D., Pal, S. ve Patranabis, D. C., “Support Vector Regression”, **Neural Information Processing - Letters and Reviews**, C: 11, No: 10, 2007, p. 206.



3.51 numaralı denklemde gösterilen problem kareli dış bükey bir optimizasyon problemidir. Lagrange fonksiyonu ve KKT koşullarının uygulanmasıyla dual forma aşağıdaki şekilde dönüştürülür (Camastra & Vinciarelli, 2010: 253).

$$\max_{\alpha, \alpha^*} -\frac{1}{2} \sum_i \sum_j (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) (x_i^T x_j) - \varepsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \quad (3.57)$$

Kısıtlar:

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \quad 0 \leq \alpha_i \leq C \quad \text{ve} \quad 0 \leq \alpha_i^* \leq C \quad i = 1, \dots, l \quad (3.58)$$

Yukarıda yer alan dual probleminin aşağıda yer alan regresyon fonksiyonunu elde etmemizi sağlar (Levis & Papageorgiou, 2005: 1012).

$$f(x) = w^T x + b = \sum_i (\alpha_i - \alpha_i^*) x_i^T x + b \quad (3.59)$$

Mercer koşullarını sağlayan Kernel fonksiyonları kullanılarak doğrusal olmayan destek vektör regresyon fonksiyonu ise aşağıdaki şekilde yazılabilir (Yu vd., 2013:1058). En çok kullanılan Kernel fonksiyonları Gaussian ve çokterimli Kernel fonksiyonlarıdır.

$$\max_{\alpha, \alpha^*} -\frac{1}{2} \sum_i \sum_j (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i x_j) - \varepsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \quad (3.60)$$

Kısıtlar:

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \quad 0 \leq \alpha_i \leq C \quad \text{ve} \quad 0 \leq \alpha_i^* \leq C \quad i = 1, \dots, l \quad (3.61)$$

$$f(x) = w^T x + b = \sum_i (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (3.62)$$

Regresyon için Schölkopf vd. (2000) tarafından benzer bir $v - \text{DVM}$ tanımlaması yapılmıştır. Bu tanımlamada ε yerine destek vektör oranını belirleyen v değeri sabitlenmekte ve ek olarak yine bir C parametresi bulunmaktadır (Alpaydın, 2012: 281).

3.5 Karesel Programlama Problemlerinin Çözümünde Kullanılan Algoritmik Yaklaşımlar

Doğrusal ve kareli programlama problemlerinin çözümünde kullanılan pek çok yöntem bulunmaktadır. Doğrusal programlama *Simpleks* yöntem ile

çözölebilmektedir. Birçok popüler matematiksel yazılım paketlerinde doğrusal programlama paketleri yer almaktadır.

Kareli programlama problemleri kısıtların doğrusal olduđu, amaç fonksiyonunun karesel olduđu doğrusal olmayan problemlerdir. Karesel programlama problemlerinin çözümü için de *Eşlenik Gradyan* ve *Primal-Dual İç Nokta* algoritmalarını içeren birçok uygulanabilir teknik bulunmaktadır (Camastra & Vinciarelli, 2010: 223).

Fakat bu teknikler büyük boyutlu verilerin bulunduđu problemler için uygun değildir. Bunun sebepleri ise aşağıda yer almaktadır (Byun & Lee, 2002: 218).

- Kernel matrisini $K(x_i x_j)$ hesaplamak ve depolamak için yüksek boyutta bellek kapasitesine ihtiyaç duyarlar.
- Bu teknikler için pahalı matris operasyonları gerekmektedir.
- Kendi DVM sınıflandırıcısını geliştirmek isteyen uygulayıcılar için bu algoritmaların kodlamasını yapmak oldukça zordur.

Eğer örnek sayısı oldukça büyükse hesaplama yapabilmek için genellikle *Ayrıştırma yöntemleri* yardımcı olmaktadır. Ayrıştırma yöntemleri çözüm aşamasındaki zorluğu ortadan kaldırmak için optimizasyon probleminden daha küçük, kontrol edilebilir alt problemler oluşturmakta ve bu problemleri iterasyonlarla çözüme ulaştırmaktadır. Sayısal bir problemin çözümü için bu teknik genellikle *Chunking* ile ilişkilendirilmektedir (Steinwart & Christmann, 2008: 422).

Destek vektör makineleri için karesel programlama problemlerinde yaygın olarak kullanılan üç yöntem *Chunking*, *Osuna algoritması* ve *Ardışık Minimal Optimizasyon (Sequential minimal optimization - SMO)* yöntemidir.

3.5.1 Chunking ve Osuna Algoritması

Destek vektör makineleri karesel programlama problemlerinin çözümü için Vapnik (1982), *Chunking* adı verilen bir yöntem geliştirmiştir.

Bu yöntemde büyük karesel programlama problemleri, temel hedefi sıfır olmayan Lagrange çarpanlarını belirlemek ve sıfır Lagrange çarpanlarının tümünden kurtulmak

olan daha küçük karesel programlama problemlerine bölünür. Bu adımdan sonra chunking algoritması, problemi aşağıda örnekleri verilen adımlarla çözer (Platt, 1999: 188).

- Önceki adımdan sıfır olmayan her Lagrange çarpanı ve KKT koşullarını sağlamayan en kötü M örnekleri bulunur.
- Eğer bir adımda KKT koşullarını sağlamayan daha az M örneği varsa, bu koşulları sağlamayan örneklerin hepsi eklenir.
- Her karesel programlama alt problemi, bir önceki alt problemin sonuçlarıyla başlatılır.
- Son adımda, tüm sıfır olmayan Lagrange çarpanları tanımlanır ve böylece son adım büyük karesel programlama problemini çözer.

Chunking yönteminde kullanılan indirgenmiş matris yüksek boyutta bellek kapasitesine ihtiyaç duyduğu ve belleğe sığmadığı için hala büyük ölçekli problemlerin üstesinden gelememektedir (Camastra & Vinciarelli, 2010: 223).

Osuna (1997) da karesel programlama problemlerinin çözümü için bir teorem geliştirmiştir ve bu teorem Osuna algoritması olarak bilinmektedir.

Bu teoremden, büyük karesel programlama problemi daha küçük karesel programlama problemlerine bölünür. En az bir örnek KKT koşullarını sağlamadığı sürece, bir önceki alt problemin örneklerine eklenir. Her adım amaç fonksiyonunun maliyetini azaltır ve tüm kısıtlara uygun muhtemel noktayı hesaplar. Bu nedenle, daima koşulları ihlal eden en az bir örneğin eklendiği ardışık karesel programlama alt problemleri sonuca yakınsamayı garanti eder (Byun & Lee, 2002: 218).

3.5.2 Ardışık Minimal Optimizasyon Yöntemi (SMO)

Ardışık Minimal Optimizasyon (Sequential minimal optimization - SMO) yöntemi, DVM karesel programlama problemini ekstra matris depolamasına ihtiyaç duymadan, sayısal karesel programlama optimizasyon adımları kullanmadan hızlıca çözen bir algoritmadır (Mohri vd., 2012: 85).

Diğer yöntemlerin tersine SMO, her adımda en küçük olabilir optimizasyon problemini çözmeyi seçer. Standart DVM karesel programlama problemi için, en küçük olabilir optimizasyon problemi iki Lagrange çarpanını içermektedir. Çünkü Lagrange çarpanları doğrusal eşitlik kısıtlarına uymalıdır. Her adımda SMO, iki Lagrange çarpanını ortaklaşa optimize etmek için seçer, bu çarpanlar için en iyi değerleri bulur ve DVM'yi yeni değerleri yansıtacak şekilde güncelleştirir (Platt, 1999: 190). SMO, ekstra bellek kapasitesine ihtiyaç duymadığı için çok büyük DVM karesel programlama problemlerini kişisel bilgisayarları kullanarak dahi çözüme kolayca ulaştırabilmektedir (Byun & Lee, 2002: 219).

$$\max_{\alpha} -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{i=1}^l \alpha_i \quad (3.63)$$

Kısıtlar:

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad \text{ve} \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \quad (3.64)$$

3.64 numaralı eşitlikte kısıtları verilen 3.63 numaralı dual problemin çözümünde SMO yöntemini uygulayarak bir örnek verecek olursak, α_1 ve α_2 kullanılarak problem aşağıdaki şekilde küçültülecektir. ($Y = y_1 \sum_{i=3}^l y_i \alpha_i$, $s = y_1 y_2 \in \{-1, +1\}$, $K_{ij} = (x_i, x_j)$, $v_i = \sum_{j=3}^l \alpha_j y_j K_{ij}$ $i = 1, 2$ için) (Mohri vd., 2012: 85)

$$\max_{\alpha_1, \alpha_2} \underbrace{\alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - s K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2}_{\Psi_1(\alpha_1, \alpha_2)} \quad (3.65)$$

$$\Psi_1(\alpha_1, \alpha_2)$$

Kısıtlar:

$$0 \leq \alpha_1, \alpha_2 \leq C \quad \text{ve} \quad \alpha_1 + s \alpha_2 = Y \quad (3.66)$$

Sadece α_2 'ye bağlı yeni bir amaç fonksiyonu Ψ_2 elde etmek için, doğrusal kısıt $\alpha_1 = Y - s\alpha_2$ 'yi Ψ_1 'de yerine koyarsak α_2 , Ψ_2 amaç fonksiyonunu minimize edecektir. ($\eta = K_{11} + K_{22} - 2K_{12}$)

$$\alpha_2 = \frac{s(K_{11} - K_{12})Y + y_2(v_1 - v_2) - s + 1}{\eta} \quad (3.67)$$

$$(v_1 - v_2) = f(x_1) - f(x_2) + \alpha_2^* y_2 \eta - s y_2 Y (K_{11} - K_{12}) \quad (3.68)$$

$f(x_1) = \sum_{i=1}^l \alpha_i^* y_i(x_i x) + b^*$ ve α_i^* değerleri optimizasyondan önceki Lagrange çarpanlarını ifade etmektedir (Camastra & Vinciarelli, 2010: 224).

$$\alpha_2 = \alpha_2^* + y_2 \frac{(y_2 - f(x_2)) - (y_1 - f(x_1))}{\eta} \quad (3.69)$$

α_2 'de alt ve üst sınırlar $s = +1$ için $L = \max\{0, Y - C\}$ ve $H = \min\{C, Y\}$ olarak tanımlanır. Benzer şekilde, $s = -1$ için $L = \max\{0, -Y\}$ ve $H = \min\{C, C - Y\}$ olarak tanımlanır. SMO için güncelleme kuralı α_2 değerinin kesilmesini (*Clipping*) içerir.

$$\alpha_2^{clip} = \begin{cases} \alpha_2 & \text{eğer } L < \alpha_2 < H \text{ ise} \\ L & \text{eğer } \alpha_2 \leq L \text{ ise} \\ H & \text{eğer } \alpha_2 \geq H \text{ ise} \end{cases} \quad (3.70)$$

Yeni α_1 değeri ise aşağıdaki şekilde bulunur.

$$\alpha_1 = \alpha_1^* + s(\alpha_2^* - \alpha_2^{clip}) \quad (3.71)$$

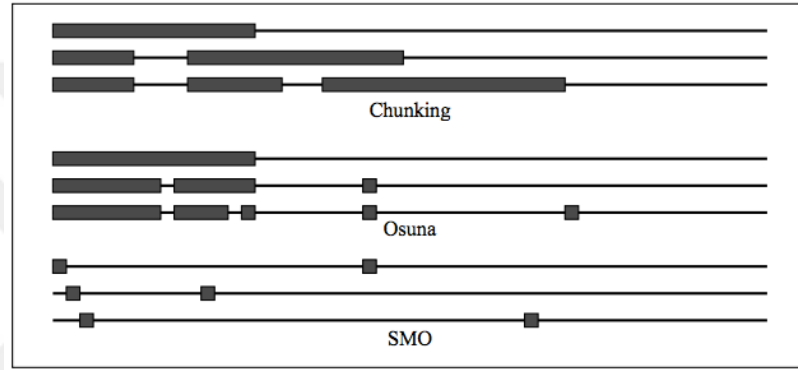
Destek vektör makineleri karesel programlama problemlerinin çözümünde kullanılan Chunking, Osuna algoritması ve SMO yöntemlerini Şekil 18 yardımıyla karşılaştırabiliriz. Şekil 18'de her yöntem için üç adım resimle gösterilmiştir. Her adımdaki yatay ince çizgiler eğitim setini, koyu renk kutular ise o adımda optimize edilen Lagrange çarpanlarını ifade etmektedir (Platt, 1999: 189).

Chunking yönteminde sıfır olan Lagrange çarpanları her adımda çıkartılırken, sabit sayıda belirlenmiş örnek eğitim setine eklenir. Bu nedenle her adımdaki eğitilen örnek sayısı büyüme eğilimi gösterir. Osuna algoritmasında her adımda sabit sayıda

belirlenmiş örnek optimize edilir: her adımda aynı sayıda örnek eklenir ve çıkartılır. SMO yönteminde ise her adımda sadece iki örnek analitik olarak optimize edilir, böylece her adım çok hızlıdır.

Şekil 18 - Chunking, Osuna Algoritması ve Ardışık Minimal Optimizasyon (SMO) Yöntemi

Kaynak: Platt, J. C., “Sequential Minimal Optimization: A Fast Algorithm For Training Support Vector Machines”, **Advances in Kernel Methods: Support Vector Learning**, 1999, p.189.



3.6 Destek Vektör Makineleri Avantaj ve Dezavantajları

Destek vektör makineleri son yıllarda oldukça yaygın olarak kullanılan yöntemlerden biridir. Deneysel eğitim hatasını minimize eden geleneksel yöntemlerin aksine destek vektör makineleri, veri ile ayıran hiper düzlem arasındaki kenar boşluğunu maksimize ederek genelleme hatasının üst sınırını minimize etmeyi amaçlamaktadır.

Destek vektör makinesi, küçük eğitim örneklerinin bulunduğu koşullarda çok yüksek boyutlu uzayda bile iyi genelleme yeteneğine sahip olmasından dolayı yapay sinir ağları gibi diğer geleneksel yöntemlerden daha üstündür (Byun & Lee, 2002: 213). Destek vektör makinelerini çok katmanlı yapay sinir ağlarıyla karşılaştırarak avantaj ve dezavantajlarından bahsedecek olursak:

Avantajları:

Genelleme yeteneğinin maksimizasyonu: Çok katmanlı bir yapay sinir ağı öğreniminde, çıktılar ile arzu edilen eğitim çıktıları arasındaki hata karelerinin toplamı minimize edilmektedir. Böylece, başlangıç ağırlıkları değişikçe sınıf sınırları da

değişmektedir. Bu durum genelleme yeteneği için de söz konusudur. Özellikle eğitim verisi sınırlı ve doğrusal ayrılabilir olduğunda, genelleme yeteneği kötüleşmektedir. Fakat destek vektör makinesinin kenar payını maksimize etmeye çalışmasından dolayı genelleme yeteneği bu koşullarda bile çok fazla kötüleşmemektedir (Abe, 2005: 39).

Tek optimum çözümün elde edilmesi: Destek vektör makinelerinde optimizasyon problemi dış bükey olduğundan dolayı tek optimum çözüm elde edilmektedir. Fakat yapay sinir ağlarında yerel minimum noktalarına bağlı olarak çoklu çözümler ortaya çıkmaktadır ve bu sebeple farklı örnekler üzerinde güçlü durmamaktadır (Auria & Rouslan, 2008: 7).

Aykırı değerlere karşı dayanıklılık: Yapay sinir ağı sınıflandırıcıları hata karelerinin toplamını kullandıkları için aykırı değerlere karşı dayanıklı değildir. Destek vektör makinelerinde ise kenar payı parametresi C , yanlış sınıflandırma hatasını kontrol etmekte ve aykırı değerleri baskı altında tutmaktadır (Abe, 2005: 39).

Dezavantajları:

Uzun eğitim süresi ve bellek büyüklüğü ihtiyacı: Bir destek vektör makinesinin eğitimi, değişken sayısının eğitim veri sayısına eşit olan ilgili dual problemin çözümünü içermektedir. Destek vektör makinesinin uygulamadaki en büyük problemi yüksek algoritmik karmaşıklığı, dual problemin çözümü için büyük bellek gereksinimi ve uzun eğitim süresidir (Suyken vd., 2003: 392).

Çok sınıflı problemlerde uygulama karmaşıklığı: Yapay sinir ağı sınıflandırıcılarının aksine, destek vektör makineleri doğrudan karar fonksiyonlarını kullanmaktadır. Bu nedenle çok sınıflı bir problemin uzantısı anlaşılır bir şekilde değildir, bir çok formülasyon mevcuttur (Abe, 2005: 39).

Parametlerin seçim sürecinin zaman alması: Bir destek vektör makinesinin eğitiminde, uygun kernelin ve parametrelerinin seçimine ihtiyaç duyulmaktadır. Verilen bir problemde optimal parametrelerin seçimine model seçimi denir. Destek vektör makinelerinde model seçimi, sürekli tekrarlanan destek vektör makinesi eğitimi

vasıtasıyla genelleme yeteneđi tahmin edilerek yapılmaktadır. Bu süreç oldukça zaman alıcı bir süreçtir (Abe, 2005: 39) (Suyken vd., 2003: 392)



DÖRDÜNCÜ BÖLÜM

YAPAY SİNİR AĞLARI

Bu bölümde öncelikle yapay sinir ağları ve modelleri anlatılacak daha sonra özörgütlemeli haritalar incelenecektir.

4.1 Yapay Sinir Ağları Tanımı ve Kullanım Alanları

Yapay sinir ağları, birçok bilim dalında karmaşık dünya problemlerinin modellenmesi için ortaya çıkmış sayısal modelleme araçlarıdır. Yapay sinirler ya da ağ olarak adlandırılan yoğun ölçüde birbirine bağlanmış uyarlanabilir basit uygulama elemanlarından oluşmuş yapılar, paralel bilgi işleme sistemleri olarak da tanımlamak mümkündür (Basheer & Hajmeer, 2000: 3).

Yapay sinir ağları, insan beyninin özelliği olan öğrenme vasıtasıyla yeni bilgiler üretebilme becerisini otomatik olarak bilgisayarlar ile gerçekleştiren sistemlerdir (Öztemel, 2012: 29)

Bir bilgisayara insan gibi düşünme yeteneği verebilmek için araştırmacıların kullanabileceği tek model insan beynidir ve insan beyni oldukça karmaşık bir modeldir. İnsan beyni sinir hücrelerinden oluşmaktadır ve bilgisayarlar, yapay sinir ağlarını kullanarak bu hücrelerin davranışlarını taklit etmeye çalışmaktadırlar (Heaton, 2008: 39).

İnsan beyni ve bilgisayarlar birbirlerinden farklı yapıdadırlar. İnsan beyni sinir hücresi olarak adlandırılan aynı anda çalışan 10^{11} tane işlemciden oluşurken bilgisayarlarda genellikle tek bir işlemci mevcuttur. İnsan beynindeki bu işlemci hücrelerin bilgisayarların işlemcilerine göre daha yavaş ve basit olduğu fakat buna rağmen yüksek bağlantısalılıktan dolayı hesap yapma gücüne sahip oldukları düşünülmektedir. “Bilgisayarda işlemci etkin, bellek edilgenken beyinde işlem ve belleğin birlikte ağ üzerine dağıtılmış olduğu düşünülür; işlem, ağdaki hücrelerce yapılırken bellek, aralarındaki kavşaklardır.” (Alpaydın, 2012: 197)

Yapay sinir ağıları, sayısal modellemenin mümkün olduğu çok çeşitli problemlerde uygulanabilmektedir. Kullanıldığı alanları aşağıdaki şekilde özetleyebiliriz (Hassoun, 1995: 1):

- Örüntü tanıma
- Ses tanıma
- Kümeleme
- Denetim
- Görüntü Sıkıştırma
- İnsanlar ve karmaşık fiziksel sistemler arasında uyarlabilir ara yüzler
- Fonksiyon kestirimi
- Çağrışımsal bellek
- Tahmin ve Kestirim
- Tümlşik optimizasyon
- Doğrusal olmayan sistem modelleme

4.2 Yapay Sinir Ağlarının Tarihi Gelişimi

İnsan beyni ve fonksiyonları üzerine yapılan çalışmalar uzun yıllar öncesine dayanmaktadır. İlk defa 1890 yılında beynin yapısı ve fonksiyonları hakkında bir yayın yayınlanmış olup 1911 yılında beynin sinir hücrelerinden meydana geldiği fikri benimsenmiştir. 1940'lı yıllardan önce bazı bilim adamlarının yapay sinir ağları üzerine çalışmaları olmasına karşın 1940'dan sonraki yıllarda yapılan çalışmaların, mühendislik değeri taşımasından dolayı, yapay sinir ağlarının ilk temelleri olduğu düşünülmektedir. 1940'lı yıllardan sonra yapay sinir ağlarıyla ilgili yapılan çalışmaları özetleyecek olursak (Öztemel, 2012: 37-41) (Elmas, 2010: 26-28) (Kriesel, 2005: 9-13) (Yegnanarayana, 2005: 21-24) :

1943 yılında Warren MacCulloch ve Walter Pitts ilk yapay sinir ağı modelini ana hatlarıyla oluşturmuşlardır.

1949 yılında Donald Hebb, yapay sinir ağları literatüründe temel bir öğrenme kuralı olan “*Hebbian öğrenme kuralı*” nı ortaya koymuştur.

1950 yılında nöropsikolog Karl Lasley, beyin bilgi depolamasının dağıtılmış sistemler olarak gerçekleştiğini savunmuştur.

1951 yılında Marvin Minsky doktora tezi için ilk nörobilgisayar olan *Snark*'ı geliştirmiş fakat hiçbir zaman pratikte uygulanamamıştır.

1956 yılında bilim adamları ve öğrencilerin katıldığı Dartmouth yaz araştırması projesinde ilk defa insan beyninin nasıl taklit edileceği tartışılmıştır.

1957-1958 yıllarında Frank Rosenblatt, Charles Wightman ve çalışma arkadaşları, "*Mark I perceptron*" adını verdikleri ilk başarılı nörobilgisayarı geliştirmişlerdir. Geliştirilen bu algılayıcı model, yapay sinir ağlarındaki çok katmanlı algılayıcıların temelini oluşturmaktadır.

1959 yılında Frank Rosenblatt, perceptron'un farklı versiyonlarını tanımlayarak "*Perceptron yakınsaklık teoremi*" ni formüle edip gerçekliğini ispatlamıştır.

1960 yılında Bernard Widrow ve Marcian E. Hoff, ADALINE (ADAPtive LInear NEuron) ve MADALINE (for many ADALINES) olarak tanımladıkları ağ modelini geliştirmişlerdir. Bu model gerçek dünya problemlerine uygulanmış ilk yapay sinir ağı modelidir.

1965 yılında Nils Nilsson, "*Öğrenen makineler*" adlı kitabında yapay sinir ağları hakkında yapılan çalışmalara yer vererek bu alanda ilk kitabın yayınlanmasını sağlamıştır.

1969 yılında Marvin Minsky ve Seymour Papert, yayınladıkları bir yayında yapay sinir ağı algılayıcılarının birçok önemli probleme çözüm bulamadığını öne sürmüşlerdir. (Doğrusal ayrılabilirlik ve XOR problemi) Bu iddia bu alanda yapılan çalışmaların oldukça yavaşlamasına neden olmuştur.

1972 yılında Teuvo Kohonen ve James A. Anderson çağrışımsal bellek ile ilgili doğrusal ilişkilendirici modeli hakkında birbirine çok benzeyen farklı yayınlar yayınlamışlardır.

1974 yılında Harvard Paul Werbos doktora tezi için “*Geriyayılım modeli*” adı verilen bir öğrenme modeli geliştirmiştir.

1976-1980 yılları arasında Stephen Grossberg yapay sinir ağlarını matematiksel olarak incelediği çok sayıda yayın çıkartmıştır ve Gail Carpenter ile birlikte “*Adaptif Rezonans Teorisi (ART)*” ni geliştirmişlerdir.

1982 yılında Teuvo Kohonen, “*Kohonen haritaları*” olarak da bilinen “*Özörgütlemeli haritalar (Self organizing maps - SOM)*” çalışmasını yayınlamıştır.

1982 ve 1984 yıllarında John Hopfield, “*Hopfield ağları*” nı ortaya çıkartmış ve bu yıllar arasında yapmış olduğu çalışmalarda yapay sinir ağlarının birçok problem için uygulanabileceğini göstermiştir. Bu çalışmaların sonucunda Geoffrey E. Hinton ve arkadaşları “*Boltzman makinesi*” ni geliştirmişlerdir. Hopfield 1985 yılında ise “Gezgin satıcı” probleminin çözümünde “*Hopfield ağları*” nı kullandığı bir makale yayınlamıştır.

1983 yılında Miyake ve Ito Fukushima, “*Neocognitron*” adı verilen elyazısını karakterlerini tanıyan bir yapay sinir ağı modeli geliştirmişlerdir.

1986 yılında David E. Rumelhart ve James MacClelland “*Paralel dağıtımli işleme*” konusundaki çalışmalarıyla ilgili yayınlar yayınlamışlardır. Bu çalışmalar vasıtasıyla çok katmanlı algılayıcılar ile XOR problemine çözüm bulunmuş ve 1969 yılında Marvin Minsky ve Seymour Papert’in tezi çürütülmüştür. Böylece yapay sinir ağlarına olan ilgi yeniden canlanmıştır.

1988 yılında David Broomhead ve David Lowe filtreleme problemlerinde başarılı sonuçlar veren çok katmanlı algılayıcılara alternatif olarak geliştirilen “*Radyal tabanlı fonksiyonlar*” ı ortaya çıkartmışlardır. Yine aynı yıl Donald Specht “*Olasılıksal ağlar*” ve daha sonra “*Genel regresyon ağları*” modellerini geliştirmiştir. Bu tarihten itibaren yapay sinir ağlarıyla ilgili çok sayıda sempozyum ve konferans düzenlenmiş ve sayısız uygulamalar gerçekleştirilmiştir.

4.3 Yapay Sinir Ağlarının Yapısı ve Ana Öğeleri

Bir yapay sinir ağı biyolojik sinir ağlarının yapısını ve fonksiyonlarını taklit etmeye çalışan matematiksel bir modeldir. Bu nedenle bu bölümde önce biyolojik sinir hücresinin yapısı daha sonra ise yapay sinir hücresinin yapısı ve ana öğeleri incelenecektir.

4.3.1 Biyolojik Sinir Hücresi

İnsan sinir sistemi vücutta buldukları bölgelere bağlı olarak farklı tipte ve uzunlukta milyonlarca sinir hücresinden oluşmaktadır. Şekil 19'da biyolojik sinir hücresinin şematik gösterimi yer almaktadır.

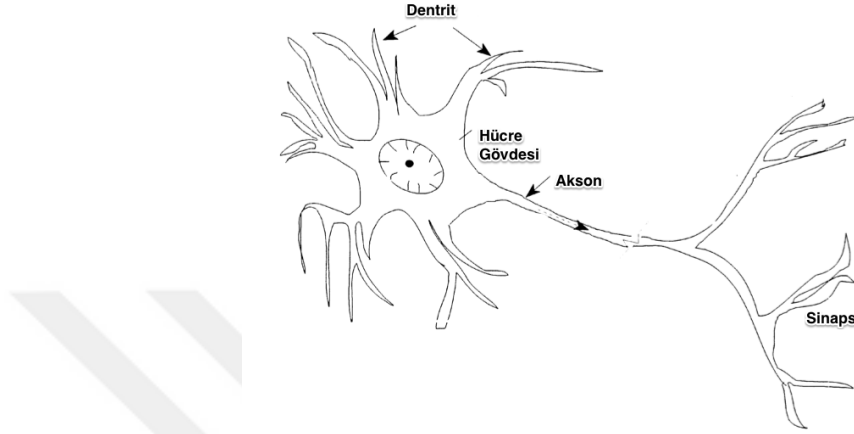
Temel bir biyolojik sinir hücresi dentrit, hücre gövdesi (soma), akson ve sinapslerden oluşmaktadır. Hücre gövdesinde bir çekirdek ve bir plazma bulunmaktadır. Hücre gövdesinden çıkan ve bir ağacın dallarına benzeyen dentritler, aksonlar aracılığıyla diğer sinir hücrelerinden gelen sinyalleri hücre gövdesine iletirler. Tipik bir sinir hücresindeki dentritlerin bu sinyalleri aldığı alan yaklaşık olarak $0,25 \text{ mm}^2$ 'dir (Basheer & Hajmeer, 2000: 4-5).

Akson, somanın ince uzun uzantısıdır. Bir aksonun uzunluğunun 1 metreden daha uzun olduğu da görülmektedir. Akson ile dentrit arasındaki boşluğa sinaps adı verilir. Komşu sinir hücreleri arasındaki sinyaller aksondan bu alana bırakılır. Dentritler tarafından alınan ve sinapse ulaşan sinyaller elektiriksel uyarılardır (Zurada, 1992: 27). Neurotransmitter vasıtasıyla iki sinir hücresi arasındaki bilgi alış verişi gerçekleşmektedir (Öztemel, 2012: 47).

Bir insan beyinde 10^{11} civarında sinir hücresi bulunmaktadır ve her bir sinir hücresi yaklaşık 10000 kadar başka sinir hücresiyle bağlantı kurmaktadır. Her bir sinir hücresinin sahip olduğu sinaps sayısı ise yaklaşık olarak 10^{14} civarındadır (Haykin, 1999: 6-8).

Şekil 19 - Biyolojik Sinir Hücresi

Kaynak: Basheer, I. ve Hajmeer, M., “Artificial Neural Networks: Fundamentals, Computing, Design, And Application”, **Journal of Microbiological Methods**, C: 43, 2000, p. 3.



4.3.2 Yapay Sinir Hücresi

Biyolojik sinir ağlarının yapısını ve fonksiyonlarını taklit etmeye çalışan yapay sinir hücrelerinin 5 temel ögesi bulunmaktadır: girdiler, ağırlıklar, toplama işlevi, aktivasyon işlevi ve çıktılar. Bir yapay sinirin çalışma prensibi Şekil 20’de yer almaktadır (Krenker vd., 2011: 3).

Girdiler: Yapay sinir hücresine çevreden gelen bilgileri ifade etmektedir. Bir yapay sinir hücresine çevreden olduğu kadar diğer hücrelerden de bilgilerin gelmesi mümkündür (Öztemel, 2012: 49).

Ağırlıklar: Her girdi değeri farklı ağırlıklarla (W_1, W_2, \dots, W_i) çarpılmaktadır. Bu ağırlıklar, girdilerin sinir hücresindeki etkisini belirlemektedir. Ağırlık değeri büyük ise o girdinin yapay sinirle bağlantısının güçlü olduğu, değer küçük ise bağlantısının zayıf olduğu anlaşılır (Elmas, 2010: 31).

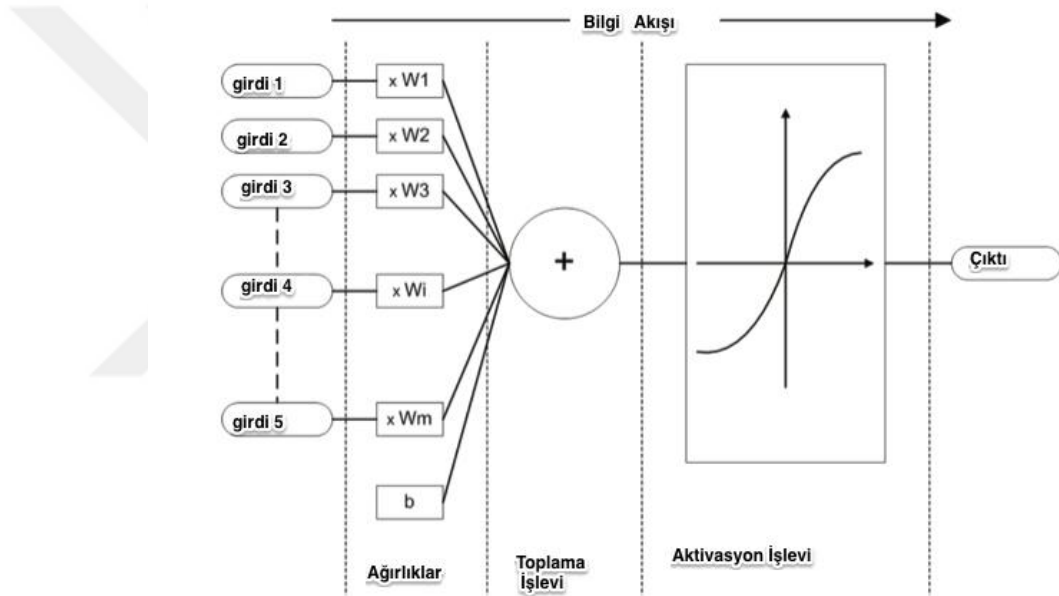
Toplama İşlevi: Bu işlev, ağırlıklandırılmış girdi değerlerini eşik değeri ile toplayarak net girdiyi hesaplamaktadır. Net girdinin hesaplanmasının gösterildiği 4.1 numaralı denklemde W ağırlıkları, X ise girdileri, b ise eşik değerini ifade etmektedir. Literatüre

bakıldığında bu formülün dışında daha karmaşık formüllerin de kullanıldığı görülmektedir (Krenker vd., 2011: 5).

$$Net_j = \sum_{i=1}^n W_{ij} X_{ij} + b \quad (4.1)$$

Şekil 20 - Yapay Sinir Hücresi

Kaynak: Krenker, A., Bester, J. ve Kos, A., "Introduction To Artificial Neural Networks", **Artificial Neural Networks-Methodological Advances and Biomedical Applications**, 2011, p. 3.



Aktivasyon İşlevi: Bu işlev, toplama işlevinin sonucuna karşılık çıktıyı belirlemektedir. Toplama işlevindeki gibi aktivasyon işlevinde de farklı formüller kullanılmaktadır. En yaygın olarak kullanılan formül (step fonksiyonu) aşağıda yer almaktadır (Basheer & Hajmeer, 2000: 6).

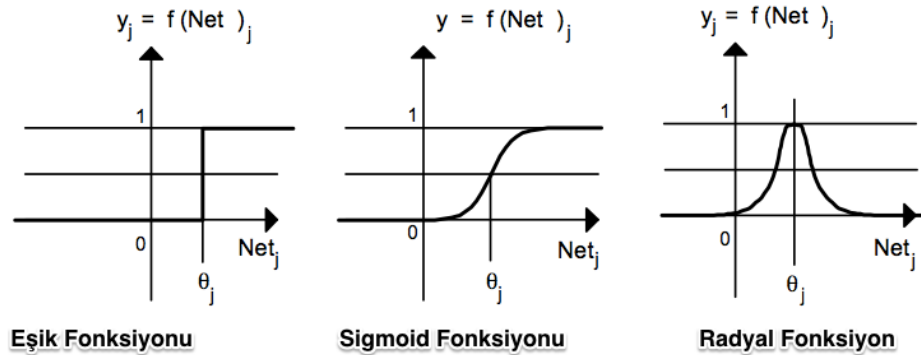
Şekil 21’de ise farklı aktivasyon işlevleri gösterilmektedir.

$$y = \begin{cases} 1 & \text{eğer } \sum_{i=1}^n W_{ij} X_{ij} \geq b \\ 0 & \text{eğer } \sum_{i=1}^n W_{ij} X_{ij} < b \end{cases} \quad (4.2)$$

Çıktılar: Aktivasyon işlevi sonucunda belirlenen değerleri ifade etmektedir. Üretilen çıktılar çevreye ya da diğer sinir hücrelerine gönderilmektedir. Ağ olarak gösteriminde birden çok çıktı değeri görünen bir yapay sinir hücresinin aslında tek bir çıktı değeri mevcuttur. Bu değer diğer yapay sinir hücrelerine girdi değeri olarak da iletilmektedir (Öztemel, 2012: 51).

Şekil 21 - Farklı Aktivasyon İşlevleri

Kaynak: Zupan, J., "Introduction To Artificial Neural Network Methods: What They Are And How To Use Them.", *Acta Chimica Slovenica*, C: 41, No: 3, 1994, p. 332.



Bir önceki bölümümüzde bir insan beyinde 10^{11} civarında biyolojik sinir hücresinin olduğunu, her bir sinir hücresi yaklaşık 10000 kadar başka sinir hücresiyle bağlantı içinde olduğunu ve her bir sinir hücresinin sahip olduğu sinaps sayısının ise yaklaşık olarak 10^{14} civarında olduğundan bahsetmiştik. Bunun aksine bir dijital bilgisayarda uyarlanabilen en karmaşık yapay sinir ağı 10 ile 10000 arasında sinir hücresine sahiptir ve her bir sinir hücresinin bağlantı sayısı 5 ile 100 arasındadır (Basheer & Hajmeer, 2000: 7).

İnsan beyni idrak etmeye yönelik olayları kısa sürede gerçekleştirirken hesaba dayalı işlemlerde süreye ihtiyaç duyar. Bunun aksine bilgisayarlar hesaba dayalı işlemleri çok

hızlı sonuca ulaştırırken idrak etmeye yönelik olaylarda yetersiz kalırlar. İnsan beyninin bu noktada üstün olan yanı sinirsel algılayıcılar ile kazanılmış ve nispi sınıflandırılmış bilgileri değerlendirebilmesidir (Elmas, 2010: 21).

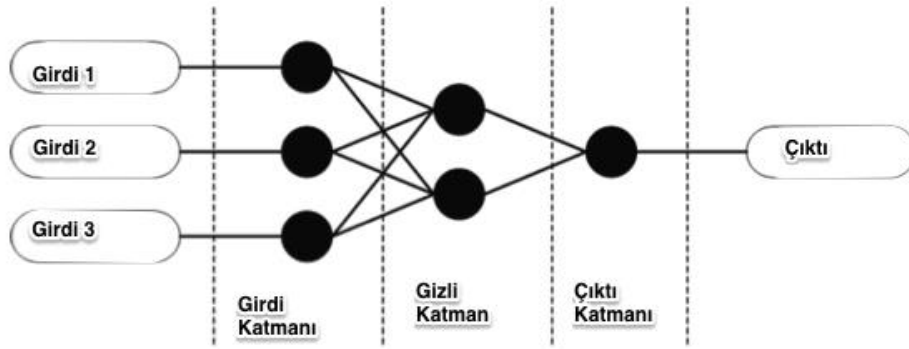
4.3.3 Yapay Sinir Ağlarının Yapısı

Farklı sayıda yapay sinir hücrelerinin bir araya gelmesiyle yapay sinir ağı oluşmaktadır. Yapay sinir ağındaki sinir hücreleri tek katman ya da iki, üç ve daha fazla katman halinde bir araya gelebilir (Zupan, 1994: 334)

Şekil 22’de bir yapay sinir ağı örneği yer almaktadır. Yuvarlak şekiller sinir hücrelerini temsil etmektedir ve bir yapay sinir ağı girdi katmanı, gizli katman ve çıktı katmanı olmak üzere 3 katmandan oluşmaktadır.

Şekil 22 - Bir Yapay Sinir Ağı Örneği

Kaynak: Krenker, A., Bester, J. ve Kos, A., “Introduction To Artificial Neural Networks”, **Artificial Neural Networks-Methodological Advances and Biomedical Applications**, 2011, p. 3.



Girdi katmanında yer alan sinir hücreleri çevreden gelen bilgileri gizli katmana iletmektedirler. Tek bir gizli katman olabileceği gibi birden fazla gizli katman da olabilmektedir. Girdi katmanından gelen bilgiler gizli katmanlarda aktivasyon işlevi vasıtasıyla işlenerek çıktı katmanına iletilir. Çıktı katmanı ise kendisine gelen bilgilere uygun olarak çıktı değerlerini oluşturur (Wali, 2014: 107).

Yapay sinir ağlarının çalışma prensibine baktığımız zaman basit bir tanımlama yapmak oldukça zordur. Çoğunlukla birbirine paralel bağlı basit aritmetik birimleri

kullanarak çoklu girdi ve çıktıları işleyen bir kara kutuya benzetilmektedir. Unutulmaması gereken en önemli şey, yapay sinir ağları yöntemlerinin girdi ve çıktılar arasında doğrusal bir ilişki olmadığı durumlarda en iyi şekilde çalışacağıdır. Yapay sinir ağları doğrusal ilişkiyi tanımlamak ya da bulmak için de kullanılabilir fakat sonuçlar diğer basit standart istatistiksel yöntemlerin sonuçlarından daha kötü çıkmaktadır (Zupan, 1994: 328).

Bir yapay sinir ağının girdi değerlerini çıktı değerlerine nasıl dönüştürdüğü yani sonuçları nasıl oluşturduğunu açıklayamaması bu yöntemin bir dezavantajıdır. Bu durum bu ağa olan güveni zedelemekle birlikte yapay sinir ağlarıyla yapılan başarılı uygulamaların varlığı bu yöneme olan ilgili sürekli arttırmaktadır (Öztemel, 2012: 54).

Şekil 23 - Kara Kutu Benzetimi

Kaynak: Zupan, J., "Introduction To Artificial Neural Network Methods: What They Are And How To Use Them.", *Acta Chimica Slovenica*, C: 41, No: 3, 1994, p. 32.



4.4 Yapay Sinir Ağlarında Temel Öğrenme Kuralları

Yapay sinir ağları için geliştirilmiş birçok öğrenme algoritması mevcuttur. Öğrenme algoritmaları ile tüm sinir hücresi çiftlerinin arasındaki bağlantı ağırlık değerlerinin ne şekilde değişeceği belirlenmektedir (Yegnanarayana, 2005: 31-32).

Farklı öğrenme tiplerine göre farklı öğrenme kuralları mevcuttur ve bu öğrenme kurallarının bir takım özellikleri vardır (Storkey & Valabregue, 1999: 869):

- Bir öğrenme kuralı yerel olabilir. Belirli bir bağlantının güncellenmesi sadece sinir hücrelerinin her iki tarafındaki bağlantıyla ilişkili bilgiye dayalıysa öğrenme kuralının yerel olduğu söylenir.
- Bir öğrenme kuralı artımlı olabilir. Öğrenme süreci, yeni örneği hafızaya almak için eski bir ağ düzenini daha önceden öğrenilmiş örneklere ihtiyaç duymadan güncelleyebiliyorsa öğrenme kuralının artımlı olduğu söylenir.
- Bir öğrenme kuralı, ağ düzeninin güncellenmesinde ya çok hızlı uygulama gerçekleştirebilmektedir ya da sınırlı bir süreç söz konusu olabilir.
- Bir öğrenme algoritmasının bir kapasitesi vardır. Kapasite, bir ağ içinde kaç tane örneğin olduğuna ilişkin ölçüyü ifade etmektedir.

Önemli öğrenme kuralları aşağıda yer almaktadır:

Hebb Kuralı: En eski öğrenme kuralı 1949 yılında Donald Hebb tarafından yazılan “*The Organization of Behavior*” kitabında yer alan öğrenme kuralıdır (Kuriscak vd., 2015: 27). Denetimsiz bir öğrenme kuralı olan *Hebb* kuralına göre eğer iki sinir hücresi eş zamanlı olarak aktif hale gelirse aralarındaki bağlantı güçlenecektir. Eğer iki sinir hücresi eş zamanlı olarak aktif hale gelmezse aralarındaki bağlantı kesilecektir. Pavlov’un bu öğrenme kuralını temel alarak yaptığı klasik koşullanma deneyi çok güzel bir örnek teşkil etmektedir. Bu deneyde, bir köpeğe ilk olarak zil çalındığı zaman köpeğin hiçbir tepki vermediği gözlemlenir. Yiyecek verildiğinde ise köpeğin salyası akar. Daha sonra köpeğe yiyecek verilirken aynı anda da zil çalınır. Zili duymak ve yiyeceğe salya akıtmak için sinir hücreleri aynı anda aktif hale gelir ve aralarında kuvvetli bir bağlantı oluşur. Daha sonra her zil çaldığında köpeğin salyası aktığı gözlemlenir (Marsland, 2009: 12).

Delta Kuralı: *Widrow-Hoff* kuralı ya da *Hata Düzeltme* kuralı olarak da bilinen ve denetimli bir öğrenme kuralı olan Delta kuralı, gerçekleşen çıktı değerini beklenen çıktı değerine adım adım yaklaştırmak için bağlantı ağırlıklarını sürekli olarak değiştirmeyi hedefler. Bu amacı gerçekleştirmek için de gerçekleşen çıktı değeri ve beklenen çıktı değeri arasındaki hata karelerinin ortalamasını minimize etmek ister (Haykin, 1999: 52).

Hopfield Kuralı: Bu öğrenme kuralı Hebb öğrenme kuralına benzerlik göstermektedir. Bu kural yapay sinir hücreleri arasındaki bağlantının ne kadar güçlendirilmesi ya da zayıflatılması konusunda yol gösterir. Eğer girdi ve çıktılarının her ikisi de aktif ise bağlantı ağırlıkları öğrenme katsayısı kadar kuvvetlendirilir. Girdi ve çıktılarının her ikisi de aktif değilse bağlantı ağırlıkları öğrenme katsayısı kadar azaltılır. Öğrenme katsayısı ise 0 ile 1 arasında yer alan ve kullanıcı tarafından atanan bir değeri ifade eder (Öztemel, 2012: 26) (Storkey & Valabregue, 1999: 870).

Kohonen Kuralı: Yarışmacı öğrenme kuralı ya da “Kazanan tümünü alır -Winners take it all” kuralı olarak da bilinen, denetimsiz bir öğrenme kuralı olan Kohonen kuralında bir yapay sinir ağındaki yapay sinir hücreleri aktif hale gelebilmek için birbirleriyle yarışır. Hebb öğrenme kuralında aynı anda bir kaç sinir hücresi aktif hale gelirken Kohonen kuralında sadece bir tane sinir hücresi aktif hale gelir (Haykin, 1999: 58). Aktif hale gelen yani kazanan hücre bağlantı ağırlıklarını değiştirerek komşu hücrelere göre daha güçlü hale gelmektedir (Elmas, 2010: 35).

4.5 Yapay Sinir Ağı Mimarilerinin Sınıflandırılması

İki ya da daha fazla yapay sinir hücresinin bir araya gelmesiyle bir yapay sinir ağı oluşmaktadır. Tek bir yapay sinir hücresi gerçek hayat problemlerinin çözümünde kullanışsızken yapay sinir ağları karmaşık gerçek hayat problemlerinin çözümünde oldukça yeteneklidir. Birbirlerinden ayrı yapay sinir hücrelerinin birbirlerine bağlanma şekline “*Topoloji*” adı verilmektedir. Bağlantının çeşitli yollardan yapılabilir olmasından dolayı çok sayıda farklı topolojiler söz konusu olmakla birlikte genellikle ileri beslemeli ve geri beslemeli olmak üzere iki ana topoloji ayrımı yapılmaktadır. Şekil 24’de bu iki topoloji örneği gösterilmektedir (Krenker vd., 2011: 6).

İleri Beslemeli Ağlar: Bir ileri beslemeli ağda, yapay sinir hücreleri katmanlara yerleştirilmiştir. Şekil 24’de de gösterildiği gibi bir katmanın her yapay sinir hücresi bir önceki katmanın yapay sinir hücreleriyle bağlantı halindedir. Sinyal yayılımı girdi katmanından çıktı katmanına doğru ağın gizli katmanlarından geçerek

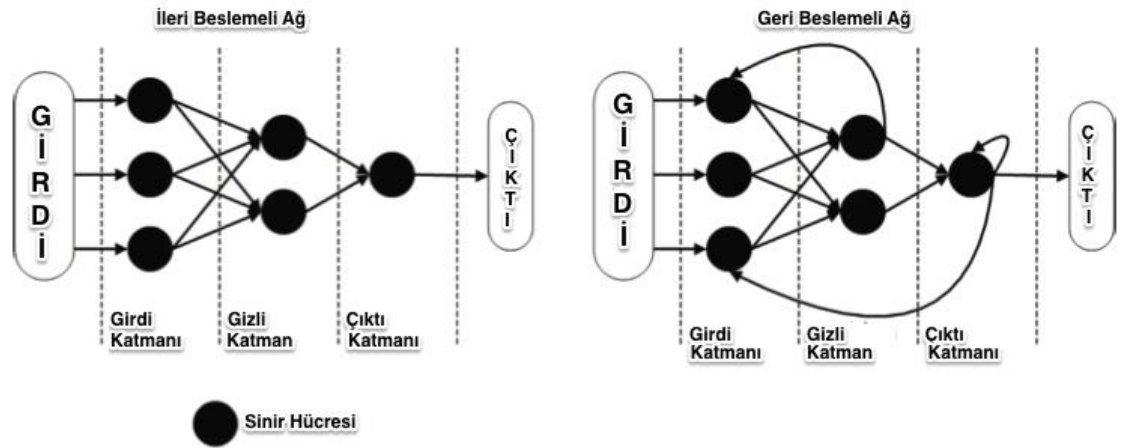
gerçekleşmektedir. Gizli yapay sinir hücreleri girdi özelliklerini temsil ederken, çıktı sinir hücreleri ağı tepkilerini ortaya çıkarmaktadır (Cruz vd., 2011: 47).

Tek katmanlı algılayıcılar, çok katmanlı algılayıcılar ve radyal tabanlı fonksiyon ağıları ileri beslemeli ağlardandır (Gardner & Dorling, 1998: 2628).

Geri Beslemeli Ağlar: Geri beslemeli bir ağı ileri beslemeli bir ağdan ayıran en önemli özellik en az bir geri besleme döngüsünün olmasıdır (Haykin, 1999: 23). Geri beslemeli ağlarda yapay sinir hücreleri çıktılarının ağı geri dönüşümü sağlanarak bu çıktılar girdi olarak kullanılmaktadır ve doğrusal olmayan çağrışımsal bellek gibi çalışmaktadır. İleri doğru bağlantılarla birlikte yapay sinir hücrelerinin kendi ya da kendinden önceki katmanda yer alan yapay sinir hücreleriyle de bağlantısı olmaktadır ve dinamik davranış sergilemektedirler (Alpaydın, 2012: 229) (Öztemel, 2012: 165).

Şekil 24 - Bir Yapay Sinir Ağına İleri ve Geri Beslemeli Topolojisi

Kaynak: Krenker, A., Bester, J. ve Kos, A., "Introduction To Artificial Neural Networks", **Artificial Neural Networks-Methodological Advances and Biomedical Applications**, 2011, p. 6.



Yarışmacı ağlar, özörgütlemeli harita (SOM) ağları, Hopfield ağı ve adaptif rezonans teori (ART) ağları geri beslemeli ağlardandır (Gardner & Dorling, 1998: 2628).

4.6 Tek Katmanlı Algılayıcılar

1943 yılında MacCulloch ve Pitts ilk yapay sinir ağı modelini ana hatlarıyla oluşturulmuş ve 1949 yılında yapay sinir ağları literatüründe temel bir öğrenme kuralı

olan “*Hebbian öğrenme kuralı*”nı ortaya koyulmuştur. 1958 yılında ise Rosenblatt, doğrusal ayrılabilir örüntü sınıflandırması için kullanılan *Perceptron* adı verilen basit tek katmanlı bir algılayıcı modeli geliştirmiştir ve geliştirdiği bu model çok katmanlı algılayıcıların temelini oluşturmuştur.

1960 yılında ise Widrow ve Hoff, ADALINE ve MADALINE olarak tanımladıkları ağ modelini geliştirmişlerdir. Bu modeller de tek katmanlı algılayıcı modellerine örnektir.

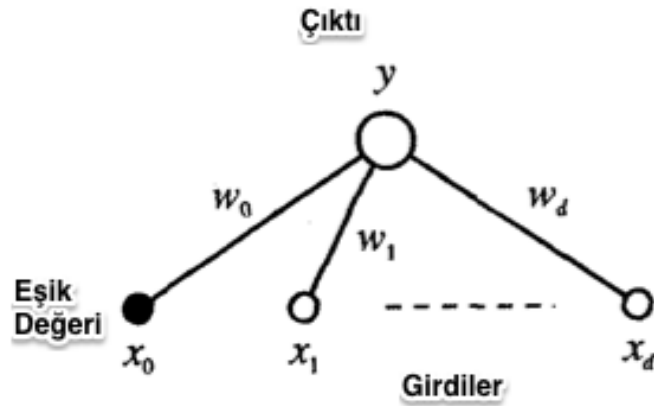
4.6.1 Perceptron

İlk defa 1958 yılında ise Rosenblatt tarafından doğrusal ayrılabilir örüntü sınıflandırması için kullanılan, tek katmanlı ileri beslemeli bir ağ olan *Perceptron* sadece girdi ve çıktı katmanlarından oluşmaktadır.

Şekil 25’de tek katmanlı basit bir algılayıcı modeli olan Perceptron modeli yer almaktadır. Şekil 25’de x_1, \dots, x_d girdileri, w_1, \dots, w_d bağlantı ağırlıklarını, $y(x)$ çıktıyı ifade etmektedir. w_0 eşik değeridir ve x_0 olarak gösterilen bu eşik değerinin girdisi her zaman 1 değerini almaktadır (Bishop, 1995: 79-81).

Şekil 25 - Perceptron Modeli

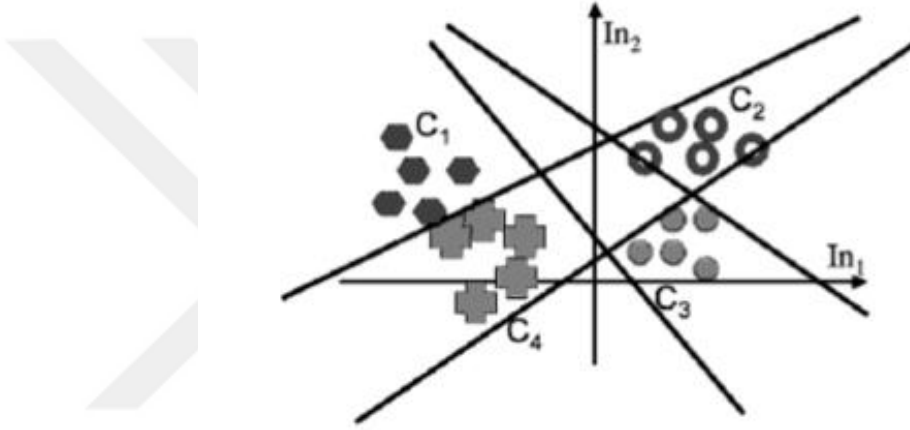
Kaynak: Bishop, C. M., *Neural Networks For Pattern Recognition*, Oxford, UK, Clarendon Press, 1995, p. 79.



Şekil 26’da ise C_1, C_2, C_3 ve C_4 olarak adlandırılan dört tane sinir hücresinin Perceptron tarafından hesaplanmış farklı karar sınırları yer almaktadır. Bu sınıflandırma probleminde sınıflar birbirlerinden doğrusal olarak ayrılmaktadır (Marsland, 2009: 34).

Şekil 26 - Perceptron Modeliyle Yapılan Sınıflandırma

Kaynak: Marsland, S., *Machine Learning An Algorithmic Perspective*, USA, A Chapman & Hall Book CRC Press, 2009, p. 34.



Perceptron öğrenme algoritması ise aşağıdaki adımlardan oluşmaktadır (Kasabov, 1996: 270):

Adım 1: Her girdi için ($x_i \in \mathbb{R}, i = 1, \dots, d$) rastgele küçük değerli bağlantı ağırlıkları ($w_{ij} \in \mathbb{R}, i = 0,1,2, \dots, n, j = 1,2, \dots, m$) tanımlanır.

Adım 2: y çıktısı, her j sinir hücresi için aktivasyon işlevi (g) kullanılarak aşağıdaki şekilde hesaplanır. w_0 değeri genelleme yapmak için eklenmiş bir değerdir $x_0 = 1$ olarak gösterdiğimiz girdinin ağırlığını ifade etmektedir.

$$y_j = g \left(\sum_{i=1}^d w_{ij} x_i + w_0 \right) \quad (4.3)$$

Adım 3: Bir önceki adımda tanımlanan çıktı fonksiyonu doğrusaldır ve çıktının eksi ya da artı olduğu iki sınıfı birbirinden ayırmak için kullanılmaktadır.

$$o_j = \begin{cases} 1 & \text{eğer } w_{ij}x_i > 0 \\ 0 & \text{eğer } w_{ij}x_i \leq 0 \end{cases} \quad (4.4)$$

Adım 4: Her bir sinir hücresinin hatası hedef çıktıdan (y_j) gerçekleşen çıktı (o_j) çıkartılarak hesaplanır.

$$Err_j = y_j - o_j \quad (4.5)$$

Adım 5: Her w_{ij} ağırlığı aşağıdaki şekilde güncellenir.

$$w_{ij}(t + 1) = w_{ij}(t) + \eta x_i \cdot Err_j \quad (4.6)$$

η öğrenme katsayısını ifade etmektedir ve 0 ile 1 arasında bir değerdir. Eğer hedef çıktı ile gerçekleşen çıktı değerleri birbirine eşit ise güncelleme yapılamadığını görülür.

Güncellenmenin büyüklüğü, hedef çıktı ile gerçekleşen çıktı değerleri arasındaki farka, girdi büyüklüğüne ve η ile gösterdiğimiz öğrenme katsayısına bağlıdır. Eğer hedef çıktı ile gerçekleşen çıktı değerleri arasındaki fark artarsa güncellenmenin büyüklüğü de artmaktadır. Bununla birlikte, girdi büyüdükçe ağırlığındaki güncelleme de daha yüksek oranda gerçekleşecektir. Öğrenme katsayısının küçük olması ise çok sayıda güncelleme gerekmesine sebep olur (Alpaydın, 2012: 205).

Adım 6: Hata vektörü Err_j , yeterli derecede düşene kadar 2'den 5'e kadar olan adımlar tekrarlanır.

1969 yılında Minsky ve Papert, Perceptron modelinin doğrusal olan problemlere çözüm bulabildiğini fakat XOR problemi gibi doğrusal olmayan problemlere çözüm bulamadığını öne sürmüşlerdir. Minsky ve Papert tarafından atılan bu iddia yapay sinir ağları alanında yapılan çalışmaların çoğunun durmasına sebep olmuştur.

4.6.2 XOR Problemi

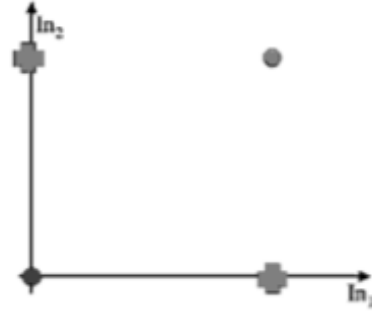
Perceptron modelinde sadece girdi ve çıktı katmanları bulunmakta ve gizli katmanlar yer almamaktadır. Bu nedenle doğrusal ayrılmayan girdileri sınıflandıramamaktadır.

Doğrusal ayrılamayan örüntü sınıflandırması da oldukça sık karşılaşılan problemlerdendir. XOR problemi bu duruma örnek olarak gösterilebilir (Haykin, 1999: 176).

Şekil 27 - XOR Probleminin Girdi ve Çıktıları

Kaynak: Marsland, S., *Machine Learning An Algorithmic Perspective*, USA, A Chapman & Hall Book CRC Press, 2009, p. 34.

ln_1	ln_2	t
0	0	0
0	1	1
1	0	1
1	1	0



XOR problemi için girdi ve çıktı değerleri şekil 27’de solda yer alan tabloda verilmektedir. Tablonun yanında yer alan şekilden de anlaşılacağı üzere iki sınıf birbirinden doğrusal olarak ayrılamamaktadır. Şekildeki iki boyutlu bir uzayda iki farklı sınıfa ait örneklerin farklı taraflarda kaldığı bir doğru çizilmesi mümkün olmamaktadır (Marsland, 2009: 34-35). Aşağıdaki koşulları sağlayan w_0, w_1 ve w_2 değerleri bulunmamaktadır (Alpaydın, 2012: 205):

$$w_0 \leq 0 \quad (4.7)$$

$$w_2 + w_0 > 0 \quad (4.8)$$

$$w_1 + w_0 > 0 \quad (4.9)$$

$$w_1 + w_2 + w_0 \leq 0 \quad (4.10)$$

1986 yılında ise Rummelhart ve MacClelland çok katmanlı algılayıcılar ile XOR problemine çözüm bulmuş ve Minsky ve Papert’in iddiasını çürütmüşlerdir. Böylece yapay sinir ağlarına olan ilgi yeniden canlanmış ve bu konuda yapılan araştırmalar artmaya başlamıştır.

4.6.3 ADALINE/MADALINE Modelleri

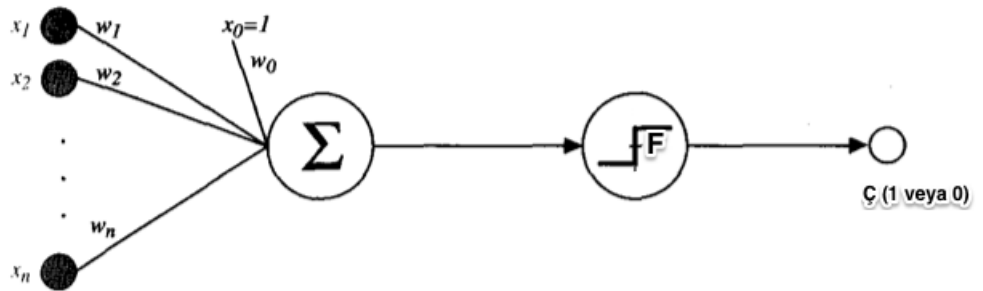
ADALINE, 1960 yılında Bernard Widrow ve Marcian E. Hoff tarafından geliştirilen ve Adaptif Doğrusal Nöron (ADaptive LInear NEuron) olarak tanımlanan ağ modelinin kısaltılmış halidir. MADALINE (for many ADALINES) ise birden çok ADALINE ünitesinin bir araya gelmesinden oluşmaktadır ve her iki ağın da öğrenme kuralları aynıdır.

ADALINE'nın ağ yapısı tek katmanlı algılayıcılara benzemektedir ve genellikle bir yapay sinir hücresinden oluşmaktadır. Şekil 28'de ADALINE ünitesinin yapısı yer almaktadır (Öztemel, 2012: 69).

ADALINE ağ modelinin öğrenme kuralı ise en küçük kareler olarak da tanımlanan delta öğrenme kuralıdır. Bu kurala göre, gerçek çıktı değerleri ile hedef çıktı değerleri karşılaştırılarak hata hesaplanır. Eğer hesaplanan hata sıfır ise ağırlıklarda değişiklik olmadan işlem biter. Fakat hata sıfırdan büyükse ağırlık değerlerinde değişiklik hata istenilen değere inene kadar devam ettirilir (Elmas, 2010: 21).

Şekil 28 - ADALINE Ünitesinin Yapısı

Kaynak: Öztemel, E., Yapay Sinir Ağları, 3. Basım, İstanbul, Türkiye, Papatya Yayıncılık, 2012, s.69.



4.7 Çok Katmanlı Algılayıcılar

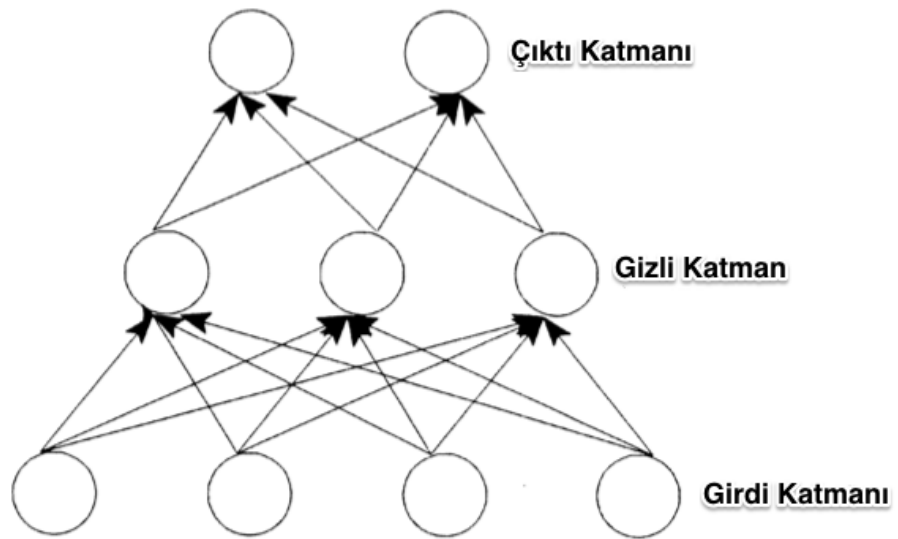
Perceptron modelinin doğrusal olmayan problemlere çözüm bulamadığını ve bu durumun yapay sinir ağlarına olan güveni azaltıp bu alandaki çalışmalarını ciddi anlamda duraklattığından daha önceki bölümlerde bahsedilmişti.

Doğrusal ayrılamayan sınıflandırma problemlerinin üstesinden gelebilmek için girdi katmanıyla çıktı katmanı arasına gizli katman adı verilen ek katmanlar yerleştirilerek çok katmanlı ileri beslemeli ağlar geliştirilmiştir. Şekil 29'da üç katmanlı bir algılayıcının yapısı gösterilmektedir (Basheer & Hajmeer, 2000: 7)

İleri beslemeli yapay sinir ağlarında dış çevreden alınan bilgi girdi katmanından gizli katmanlara oradan da çıktı katmanına doğru geri döngü olmaksızın tek bir yönde iletilmektedir. Katmanların sayısı, yapay sinir hücresinde kullanılan aktivasyon işlevinin tipiyle ya da yapay sinir hücreleri arasındaki bağlantıların sayısı ile ilgili herhangi bir sınırlama söz konusu değildir (Krenker vd., 2011: 7).

Şekil 29 - Çok Katmanlı Algılayıcıların Yapısı

Kaynak: Basheer, I. ve Hajmeer, M., "Artificial Neural Networks: Fundamentals, Computing, Design, And Application", **Journal of Microbiological Methods**, C: 43, 2000, p. 7.



Çok katmanlı bir algılayıcının ayırıcı özelliklerini özetleyecek olursak (Haykin, 1999: 156-157):

1. Ağdaki her bir yapay sinir hücresi doğrusal olmayan bir aktivasyon işlevine sahiptir. En yaygın olarak kullanılan işlev ise Sigmoid fonksiyonudur.
2. Ağda, girdi ve çıktı katmanlarının dışında bir ve birden fazla gizli katman bulunmaktadır. Bu gizli katmanlar ağa daha karmaşık görevlerin yerine getirilmesinde öğrenim kabiliyeti sağlamaktadır.
3. Ağda, ağın sinapsleri tarafından belirlenen yüksek seviyede bir bağlantı bulunmaktadır. Ağın bağlantısındaki bir değişiklik için sineptik bağlantıların ya da onların ağırlıklarının değişimi gereklidir.

4.7.1 Geriye Yayılım Algoritması

Çok katmanlı algılayıcılar, “*Geriye yayılım*” ya da “*Hatayı geriye yayma*” adı verilen bir algoritmayla öğrenmeyi gerçekleştirmektedirler. Geriye yayılım algoritması, ağın çıktı değerleriyle, hedef çıktı değerleri arasındaki hata karelerini minimize etmek ister. Hata karelerini minimize eden optimal bağlantı ağırlıklarını (w_{ij}) bulmak için de “*Eğimli iniş*” kuralını kullanır (Mitchell, 1997: 97).

Eğimli iniş yöntemine göre bağlantı ağırlıklarındaki değişim (Δw_{ij}), hata (E) eğiminin negatif yönünde ve ($t + 1$) iterasyonu için aşağıdaki şekilde hesaplanır (Kasabov, 1996: 273-275):

$$\Delta w_{ij}(t + 1) = -\eta \frac{\partial E}{\partial w_{ij}(t)} \quad (4.11)$$

Burada η öğrenme katsayısını ifade etmektedir. Eğimli iniş yöntemiyle bir kaç iterasyondan sonra hata (E) minimum değere ulaşacaktır. Tüm eğitim örnekleri için hatalar toplamı aşağıdaki şekilde hesaplanmaktadır.

$$E = \sum_{(p)} \sum_{(j)} Err_j^{(p)} \quad (4.12)$$

Bir örneğin (p) hatası $Err_j^{(p)}$ ise aşağıdaki şekilde hesaplanmaktadır.

$$Err_j^{(p)} = \frac{(y_j^{(p)} - o_j^{(p)})^2}{2} \quad (4.13)$$

i sinir hücresiyle j sinir hücresi arasındaki bağlantı ağırlığını değiştiren eğimli iniş kuralı 4.14 numaralı denklemde yer alan delta kuralı ya da 4.15 numaralı denklemde yer alan genelleştirilmiş delta kuralı ile ifade edilmektedir.

$$\Delta w_{ij}(t + 1) = \eta \cdot Err_j \cdot o_i \quad (4.14)$$

$$\Delta w_{ij}(t + 1) = \eta \cdot Err_j \cdot g'(u_j) o_i \quad (4.15)$$

Err_j , j sinir hücresi için hedef çıktı (y_j) ile gerçekleşen çıktı (o_j) arasındaki hatayı ($|y_j - o_j|$) simgeler. $g'(u_j)$ değeri ise g aktivasyon işlevinin ($\partial g / \partial u$) türevidir. u net girdi, o_i ise i sinir hücresi için çıktı değeridir. Aktivasyon işlevi g Lojistik fonksiyonu olduğunda $g'(u_j)$ değeri, $o_j(1 - o_j)$ olarak ifade edilir ve 4.15 numaralı formül aşağıdaki şekilde sadeleştirilir.

$$\Delta w_{ij}(t + 1) = \eta \cdot Err_j \cdot o_j(1 - o_j) \cdot o_i \quad (4.16)$$

4.7.2 İleri ve Geri Hesaplama

Geriye yayılım algoritmasındaki her iterasyon ileri ve geri hesaplama olmak üzere iki aşamadan oluşmaktadır.

İleri Hesaplama: Bu aşamada girdi katmanındaki girdiler herhangi bir değişiklik olmadan ara yani gizli katmanlardan geçerek çıktı katmanına doğru yayılır. Girdi katmanındaki tüm sinir hücrelerinden gelen bilgiler bağlantı ağırlıkları da göz önünde bulundurularak gizli katmandaki her sinir hücresine iletilir. Gizli katmandaki j sinir hücresine gelen net girdi aşağıdaki şekilde hesaplanır. v_{ij} değeri, i girdi katmanı hücresini j gizli katman hücresine bağlayan bağlantının ağırlığıdır. v ilk katmanın ağırlıklarını, w ise ikinci katmanın ağırlıklarını temsil etmektedir (Marsland, 2009: 54-55) (Öztemel, 2012: 78).

$$h_j = \sum_i v_{ij} \cdot x_i \quad (4.17)$$

j gizli katman hücresinin çıktısı ise bu net girdiye aktivasyon işlevinin uygulanmasıyla hesaplanmaktadır. Genellikle Sigmoid fonksiyonu kullanılmaktadır.

$$a_j = g(h_j) = \frac{1}{1 + \exp(-\beta h_j)} \quad (4.18)$$

4.18 numaralı denklemde yer alan β değeri, gizli katmanda bulunan j hücresine bağlı eşik elemanının ağırlığını ifade etmektedir. Gizli katmanın tüm sinir hücreleri ve çıktı katmanının sinir hücrelerinin çıktıları aynı şekilde hesaplanır ve çıktılar bulununca ileri hesaplama tamamlanmış olur.

Geri Hesaplama: Bu aşamada ise hedef çıktılar ile gerçekleşen çıktılar arasındaki fark yani hata hesaplanarak bu hata, ağırlıkların değişiminin hesaplanması için geriye doğru yayılır (Marsland, 2009: 55) (Öztemel, 2012: 79) (Elmas, 2010: 129-131).

Öncelikle toplam hata hedef çıktı değerleri (y_k) ile gerçekleşen çıktı (o_k) değerleri arasındaki farkın kareleri toplamı alınarak aşağıdaki şekilde hesaplanır. Daha sonra çıktı biriminin hatası δ_o kullanılarak ikinci katmandaki ağırlıklarda ilk güncelleme yapılır. Daha sonra δ_h hatası kullanılarak ilk katmandaki ağırlıklarda güncelleme yapılır.

$$\text{Toplam Hata} = \frac{1}{2} \sum_{k=1}^n (y_k - o_k)^2 \quad (4.19)$$

Çıktıdaki hatanın hesaplanması 4.20 numaralı denklemde ve gizli katmanlardaki hatanın hesaplanması 4.21 numaralı denklemde yer almaktadır.

$$\delta_{ok} = o_k(1 - o_k)(y_k - o_k) \quad (4.20)$$

$$\delta_{hj} = a_j(1 - a_j) \sum_k w_{jk} \delta_{ok} \quad (4.21)$$

Çıktı katmanındaki, ağırlıklar 4.22 numaralı denklemdeki gibi güncellenirken gizli katmandaki ağırlıklar 4.23 numaralı denklemdeki gibi güncellenmektedir.

$$w_{jk} \leftarrow w_{jk} + \eta \delta_{ok} a_j^{gizli} \quad (4.22)$$

$$v_{ij} \leftarrow v_{ij} + \eta \delta_{hj} x_i \quad (4.23)$$

Böylece ağıdaki bağlantı ağırlıkları güncellenmiş olacaktır. Her bir iterasyon aynı şekilde yapılarak öğrenme bitene kadar yinelenecektir.

4.8 Yapay Sinir Ağlarında Dikkat Edilmesi Gereken Konular

Bir ağın eğitimine başlamadan önce bazı konular üzerine düşünülmesi ve değerlendirilmesi gerekir. Bu konular aşağıda yer almaktadır ve bu konuların bazıları sadece geriye yayılım algoritmasıyla eğitilen yapay sinir ağlarını ilgilendirirken diğerleri tüm yapay sinir ağı tiplerini ilgilendirmektedir.

- Veritabanı Büyüklüğü ve Bölümlenmesi
- Girdi ve Çıktıların Temsil Edilmesi
- Ağdaki İlk Ağırlıkların Belirlenmesi
- Öğrenme ve Momentum Katsayısı
- Yakınsama ve Yerel Minimum
- Aktivasyon İşlevi
- Gizli Katmanların Büyüklüğü
- Öğrenme Biçimleri
- Verilerin Ölçeklendirilmesi
- Genelleme, Aşırı Eğitim ve Eğitimi Durdurma Zamanı

4.8.1 Veri Tabanı Büyüklüğü ve Bölümlenmesi

Verilerden geliştirilen modeller genellikle veri tabanı büyüklüğüne bağlıdır. Eğitim için kullanılan verinin problemin nüfuz alanındaki olası bilinen tüm varsayımlarının üstesinden gelebilecek büyüklükte olması gerekir. Bir yapay sinir ağı gelişimi, ana veri tabanının üç alt kümeye bölümlenmesini gerektirir: Eğitim, test ve geçerlik. Veri tabanının üç alt kümeye bölümlenmesi Şekil 30'da yer almaktadır.

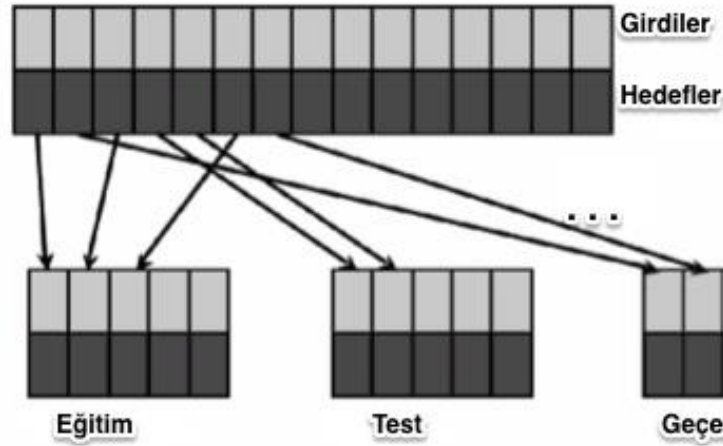
Eğitim alt kümesi, problemin nüfuz alanına ait tüm verileri içermeli ve eğitim aşamasında ağın ağırlıklarını güncellemek için kullanılmalıdır. Test alt kümesi, öğrenme süreci sırasında eğitimsiz veriler için ağın yanıtlarını kontrol etmek amacıyla

kullanılır. Test alt kümesinde kullanılan veriler eğitimde kullanılan verilerden farklı olmalıdır. Verilerin üçüncü bölümü ise diğer iki alt kümeden farklı örnekler içeren geçerlik alt kümesidir. Bu alt küme en iyi ağı seçtikten sonra ağı sınamak ya da sinir sisteminde uygulanmadan önce doğruluğunu onaylamak için kullanılır (Basheer & Hajmeer, 2000: 18).

Yapay sinir ağlarında sadece eğitim ve test alt kümeleri de kullanılabilir. Üçüncü alt küme olan geçerlik alt kümesi genellikle aşırı uyum sorununu önlemek ya da eğitim sürecinin durdurma noktasını belirlemek için kullanılır. Özellikle küçük veri setlerinde hem geçerlik hem de test için bir test alt kümesi kullanmak yaygındır. Eğitim ve test örneklerinin seçimi yapay sinir ağlarının performansını etkilemektedir.

Şekil 30 - Eğitim, Test ve Geçerlik Alt Kümleri

Kaynak: Marsland, S., *Machine Learning An Algorithmic Perspective*, USA, A Chapman & Hall Book CRC Press, 2009, p. 67.



Bahsedilen bu veri setlerinin büyüklüğünün belirlenmesiyle ilgili olarak herhangi bir matematiksel kural yoktur. Bu bölümlenme işlemi farklı şekillerde yapılabilir ya da rastgele atanabilir. Eğitim ve test alt kümelerinin kullanıldığı durumlarda genellikle literatürde yapılan çalışmaların 90:10, 80:20 ya da 70:30 oranlarında bölümlendiği görülmektedir (Zhang vd., 1998: 50).

Eğitim, test ve geçerlik alt kümelerinin kullanıldığı durumlarda ise genellikle çok verinin olduğu durumlarda 50:25:25 oranında, çok verinin olmadığı durumlarda ise

60:20:20 oranında bir bölümlenme yapıldığı görülmektedir. Eğitim, test ve geçerlik alt kümelerinin ayrılmasında k-katlamalı çapraz geçerlik yönteminin kullanılması da söz konusudur (Marsland, 2009: 68).

4.8.2 Girdi ve Çıktıların Gösterimi

Uygun veri gösterimi başarılı bir YSA tasarımında önemli rol oynamaktadır. Girdi ve çıktılar sürekli, kesikli ya da her ikisinin bir karışımı olabilir. Örnek olarak, her bir girdi değerinin farklı sınıflardan birine ait olduğu ve çıktı değerinin de bu sınıflardan birine atandığı bir sınıflandırma probleminde tüm girdi ve çıktılar 0 ve 1 gibi ikili sayılarla temsil edilebilir (Basheer & Hajmeer, 2000: 19).

Yapay sinir ağları sadece sayılarla çalışmaktadır ve sayısal olmayan faktörlerin de göz önünde bulundurulması gerekiyorsa onların da mutlaka rakamlarla gösterilmesi gerekir. Hem girdi hem de çıktı değerlerinin sayısal olarak gösterimi sağlanmalıdır (Öztemel, 2012: 94).

4.8.3 Ağdaki İlk Ağırlıkların Belirlenmesi

Eğitilecek ağın ağırlıkları genellikle pozitif ya da negatif küçük rastgele değerlerle başlatılır. Başlangıç durumu nihai çözümü kuvvetle etkilemektedir. Eğer tüm ağırlıklar eşit ağırlık değerleriyle işe başlarsa ve çözüm farklı ağırlıkların geliştirilmesini gerektiriyorsa ağ düzgün eğitilmeyebilir. Ağ, eğitim süresince rastgele faktörler ya da girdi örneklerinin rastgele karakterleri tarafından rahatsız edilmedikçe iç temsil sürekli olarak simetrik ağırlıklar olarak sonuçlanabilir (Zurada, 1992: 208).

Buradaki soru rastgele değerler verilen ağdaki ilk ağırlıkların ne kadar küçük olacağı ve bunun ne kadar önemli olduğudur? Bunu öğrenebilmek için öncelikle başlangıç ağırlıklarına sıfıra yakın değerler daha sonra da büyük değerler vererek sonuçları karşılaştırabiliriz. Bununla birlikte değerlerin neden küçük olması gerektiğini anlamak için Sigmoid fonksiyonunun şekline de bakabiliriz. Eğer başlangıç ağırlıkları 1 ya da -1'e çok yakın olursa yani büyük değerler verilirse Sigmoid girdilerinin de 1 ve -1'e çok yakın olacağı ve böylece sinir hücresinin çıktılarının 0 ya da 1 olacağı görülmektedir. Yani Sigmoid doyma noktasına gelecek, minimum ya da maksimum

noktasına ulaşacaktır. Eğer ağırlıklar çok küçük yani sifıra yakın değerler alırsa girdiler sifıra yakın olacak ve sinir hücresinin çıktısı doğrusal olacak ve doğrusal bir model elde edilecektir. Bu sebeple başlangıç ağırlıklarının belirlenmesi biraz daha dikkat gerektirmektedir. İlk ağırlıkların belirlenmesinde yaygın kullanılan bir formül de $-1/\sqrt{n} < w < 1/\sqrt{n}$ formülüdür. Burada w , ilk ağırlıkları n ise girdi katmanındaki düğüm sayısını ifade etmektedir (Marsland, 2009: 57).

4.8.4 Öğrenme ve Momentum Katsayısı

Öğrenme ve momentum katsayılarının belirlenmesi, ağıdaki ilk ağırlıkların belirlenmesi kadar ağıın performansını etkilemektedir. Ağıdaki ağırlıkların değişim miktarı öğrenme katsayısı ile belirlenmektedir. Öğrenme katsayısının büyük olması, öğrenme için gerekli adımları azaltmakta ve ağıın toplam hatasında iyileştirme sağlamaktadır. Fakat büyük öğrenme katsayıları ağıın yerel minimuma takılmasına ve salınım yapmasına sebep olarak iyi bir yakınsamanın oluşmasına engel olmaktadır. Öğrenme katsayısının küçük olması ise öğrenme süresini uzamasına sebep olmaktadır (Öztemel, 2012: 99) (Elmas, 2010: 131).

Öğrenme katsayısıyla ilgili olarak literatüre bakıldığında Wythoff (1993) 0.1-1.0 aralığında, Zupan ve Gasteiger (1993) 0.3-0.6 aralığında ve Fu (1995) 0.0-1.0 aralığındaki değerlerin kullanılmasının başarılı sonuçlar vereceğini tavsiye etmişlerdir. Bu değerlerin dışında bazı problemlerde eğitim boyunca değişen uyarlanabilir öğrenme katsayısı da kullanılmaktadır. Eğitimin başında minimum değerden uzakta olunmasından dolayı büyük öğrenme oranlarıyla eğitime başlanır ve minimum noktaya yaklaştıkça öğrenme oranı düşürülür. Minimum noktaya olan uzaklık tahmin edilemeyeceği için çeşitli sezgisel yöntemler önerilmektedir (Basheer & Hajmeer, 2000: 20).

Bir hesaplamada öğrenme katsayısının büyük olması ağıın yerel minimuma takılmasına ve salınım yapmasına neden olduğundan bahsedilmişti. Bunu ortadan kaldırabilmek ve daha iyi çözümler elde edebilmek için hesaplamaya momentum katsayısı ilave edilmektedir (Kasabov, 1996: 275). Momentum katsayısı, bir önceki yinelemedeki ağırlık değişiminin belirli bir oranını yeni değişim miktarına ilave edilmesidir.

Momentum katsayısı α ile gösterilir ve eğer ağı yerel minimuma takıldıysa bir sıçramayla buradan kurtulmasına ve daha iyi sonuçlar bulunmasına yardımcı olur (Öztemel, 2012: 99). Momentum katsayısı eğer küçük bir değer alırsa yerel minimumdan kurtulma ihtimali zorlaşabilir. Bu sebeple uygun bir momentum katsayısı seçilmelidir. Kesin bir değer olmamakla birlikte genellikle 0.6-0.9 aralığında katsayılar tercih edilmektedir (Kriesel, 2005: 114).

Momentum katsayısı hesaplamaya aşağıdaki şekilde eklenmektedir (Kasabov, 1996: 275).

$$\Delta w_{ij}(t + 1) = \eta \cdot Err_j \cdot o_j(1 - o_j) \cdot o_i + \alpha \cdot \Delta w_{ij}(t) \quad (4.24)$$

4.8.5 Yakınsama ve Yerel Minimum

Geriye yayılım algoritması, ağı çıktığı değerleriyle, hedef çıktığı değerleri arasındaki hatayı azaltacak optimal bağlantı ağırlıklarını bulmak için “*Eğimli iniş*” yöntemini kullanmaktadır. Çok katmanlı ağlar için hata yüzeyi birçok farklı yerel minimum içerebildiğinden dolayı “*Eğimli iniş*” bunlardan herhangi birinde sıkışıp kalabilir. Sonuç olarak çok katmanlı ağlar üzerinde geriye yayılım sadece hatadaki yerel minimuma doğru yakınsamayı garanti etmekte, global minimum hataya yakınsamayı garanti etmemektedir. Global minimum hataya yakınsamayı garanti edememesine rağmen geriye yayılım, pratikte oldukça etkili olarak kullanılan bir fonksiyon yaklaşırma yöntemidir (Mitchell, 1997: 104).

4.8.6 Aktivasyon İşlevi

Transfer işlevi olarak da tanımlanan aktivasyon işlevi, ateşleme yoğunluğunu belirlemek amacıyla bir sinir hücresine çarpan tüm sinyallerin ağırlıklı toplamını dönüştürmek için gereklidir. (Basheer & Hajmeer, 2000: 20).

Yaygın olarak kullanılan aktivasyon işlevleri aşağıda yer almaktadır (Zhang vd., 1998: 47) :

1. Sigmoid fonksiyonu:

$$f(x) = (1 + \exp(-x))^{-1} \quad (4.25)$$

2. Hiperbolik tanjant fonksiyonu:

$$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x)) \quad (4.26)$$

3. Sinüs ya da cosinüs fonksiyonu:

$$f(x) = \sin(x) \text{ ya da } f(x) = \cos(x) \quad (4.27)$$

4. Doğrusal fonksiyon:

$$f(x) = x \quad (4.28)$$

4.8.7 Gizli Katmanların Büyüklüğü

Geriye yayılım algoritmasının bir ilginç özelliği de ağ içindeki gizli birim katmanlarındaki kullanışlı ara ifadeleri keşfetme yeteneğidir (Mitchell, 1997: 106). Gizli katmanlar ve gizli katman sinir hücreleri başarılı sinir ağı uygulamalarında önemli bir rol oynamaktadır.

Sinir ağlarının bir özelliği algılamasına, verideki bir örüntüyü yakalamasına ve girdi/çıkış değişkenleri arasında karmaşık doğrusal olmayan eşleşme yapabilmesine izin veren gizli katmanlardaki sinir hücreleridir. Gizli katmanların olmadığı basit tek katmanlı algılayıcılar doğrusal istatistiksel yöntemlerle eş değer görülmektedir. Yapılan çalışmalar incelendiğinde özellikle tahmin problemlerinde tek gizli katmanın yeterli olduğu, bununla birlikte iki gizli katmana sahip ağların da bazı problem tiplerinde yaygın olarak kullanıldığı görülmektedir (Zhang vd., 1998: 42-43).

Gizli katman ve sinir hücrelerinin sayısının çok olması genellemeyi ve ağın öğrenme süresini kötüleştirmektedir. Öğrenme süresinin uzun olması ise algılayıcının aşırı öğrenmesine sebep olmaktadır (Alpaydın, 2012: 218). Bu sebeple gizli katman sayısının belirlenmesi oldukça önemli bir konudur.

Çok katmanlı algılayıcılarda bir problem için kaç tane gizli katman ve sinir hücresi kullanılmasıyla ilgili kesin bir yöntem mevcut değildir. Uygulayıcılar genellikle kendi tecrübelerinden yola çıkarak gizli katman ve sinir hücrelerinin sayılarını belirlemektedirler (Öztemel, 2012: 104).

4.8.8 Ağın Eğitilmesi

Yapay sinir ağları eğitilirken genellikle iki türlü öğrenme gerçekleştirilmektedir (Basheer & Hajmeer, 2000: 22):

- 1) Grup Öğrenme
- 2) Çevrimiçi Öğrenme

Öğrenme süreci boyunca tüm eğitim setinin eksiksik sunumuna “dönem” adı verilmektedir. Grup öğrenmede ağırlık güncellemesi, bir dönem boyunca oluşan tüm eğitim örneklerinin ağı sunulmasından sonra gerçekleşmektedir (Haykin, 1999: 172).

Çevrimiçi öğrenmede ise her eğitim örneği ağı sunulduktan sonra ağırlık güncellemesi gerçekleşmektedir (Wilson & Martinez, 2003: 1429). Örnekler teker teker geldiği ve bütün eğitim setine öğrenmenin başında sahip olunmadığı için genellikle çevrimiçi öğrenme kullanılmaktadır. Hata, tüm örnekler üzerinde değil tek tek her örnek için tanımlanmaktadır (Alpaydın, 2012: 203).

Grup öğrenmeyi ve çevrimiçi öğrenmeyi karşılaştırdığımızda yapılan araştırmalardan özellikle büyük eğitim örnekleri için çevrimiçi öğrenmenin grup öğrenmeden daha hızlı olduğu görülmektedir. Çevrimiçi öğrenmenin daha az belek gereksinimi vardır. Eğitim verisinin gereğinden fazla olduğu durumlarda (veri seti aynı örüntünün bir çok kopyasını içeriyorsa) örneklerin her seferinde tek tek sunulmasından dolayı çevrimiçi öğrenme daha avantajlıdır. Bununla birlikte, çevrimiçi öğrenme stokastik yapısından dolayı yerel minimumdan zaman zaman kurtulmaktadır (Wilson & Martinez, 2003: 1432).

Özetlemek gerekirse, çevrimiçi öğrenmenin bir kaç dezavantajı olmasına karşın oldukça yaygın kullanılmasının iki önemli nedeni bulunmaktadır (Haykin, 1999: 172):

- Uygulaması kolay bir algoritmadır.
- Büyük ve zor problemler için etkili sonuçlar sağlamaktadır.

4.8.9 Verilerin Ölçeklendirilmesi

Yapay sinir ağlarında problemle ilgili örnekler toplanırken, örnekler ile problem uzayı farklı ölçeklere sahip olabilir. Ya da bazen girdi ve çıktı değerleri çok büyük ya da çok küçük değerlerden oluşabilir. Bu girdi ve çıktıların belirli aralıkta ölçeklendirilmesi gerekir. Girdi ve çıktıların ölçeklendirilmesi yapay sinir ağlarının performansını doğrudan etkilemektedir (Öztemel, 2012: 101-102).

Verilerin ölçeklendirilmesinde çoğunlukla kullanılan formüller aşağıda yer almaktadır (Zhang vd., 1998: 49-50).

- $[0,1]$ aralığına doğrusal dönüşüm:

$$x_n = (x_0 - x_{min}) / (x_{max} - x_{min}) \quad (4.29)$$

- $[a, b]$ aralığına doğrusal dönüşüm:

$$x_n = (b - a)(x_0 - x_{min}) / (x_{max} - x_{min}) + a \quad (4.30)$$

- İstatiksel ölçeklendirme:

$$x_n = (x_0 - x) / s \quad (4.31)$$

- Basit ölçeklendirme:

$$x_n = x_0 / x_{max} \quad (4.32)$$

4.8.10 Genelleme, Aşırı Eğitim ve Eğitimi Durdurma Zamanı

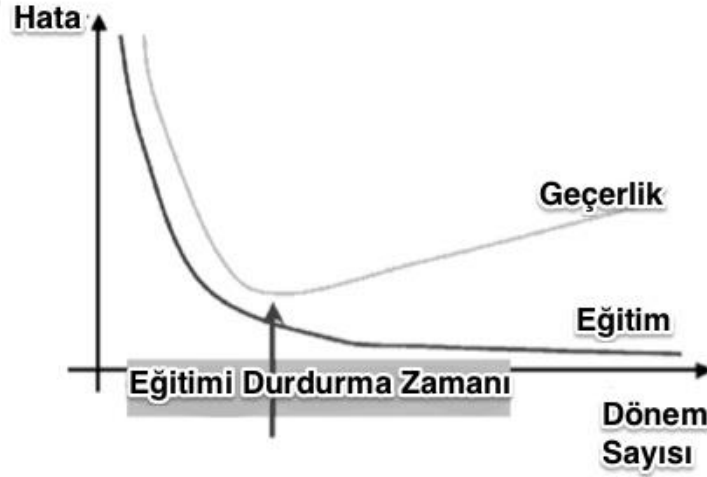
Yapay sinir ağlarının asıl amacı, eğitim örneklerinden genelleme yaparak tüm olası girdiler hakkında bilgi üretebilmektir. Ağın iyi bir genelleme yapıp yapmadığından emin olabilmek için de gerekli sayıda eğitimin gerçekleştirilmesi gerekmektedir. Bununla birlikte gereğinden az eğitim gibi gereğinden fazla eğitimin gerçekleştirilmesi de tehlikelidir. Eğer eğitim çok uzun sürerse aşırı eğitim gerçekleşecek ve gerçek fonksiyonla birlikte hata ve gürültü de öğrenilecektir (Marsland, 2009: 66).

Eğitimin başlangıcında tüm ağırlıklar sıfıra yakın bir değerdedir ve öğrenme boyunca önemli ağırlıklar sıfırdan uzaklaşmakta daha büyük değerler almaktadır. Fakat öğrenme süresinin uzun tutulması tüm ağırlıkların sıfırdan uzaklaşıp önem kazanmasına sebep olacaktır. Öğrenme sürdükçe model daha karmaşık hale gelecek ve bu durum iyi genelleme yapılamamasına neden olacaktır (Alpaydın, 2012: 218).

Aşırı eğitim probleminin üstesinden gelebilmek için bazı teknikler bulunmaktadır. Bunlardan biri “Ağırlık azaltma” tekniğidir. Bu teknikte, her yinelemede her bir ağırlık azaltılmaktadır. Başka bir ifadeyle ağ ağırlıklarının toplam büyüklüğüne ilişkin bir ceza terimi ekleyerek hata tanımı biçimlendirilmektedir. Bu yaklaşımın amacı, ağırlık değerlerini küçük tutarak modelin karmaşık hale gelmesini engellemektir (Mitchell, 1997: 111).

Şekil 31 - Eğitimi Durdurma Zamanı

Kaynak: Marsland, S., *Machine Learning An Algorithmic Perspective*, USA, A Chapman & Hall Book CRC Press, 2009, p. 69.



Bu problemin üstesinden gelebilmek için kullanılan bir diğer başarılı teknik ise algoritmaya test örneklerine ilaveten geçerlik örneklerinden oluşan bir veri seti sağlamaktır. Şekil 31 incelendiğinde iki öğrenme eğrisi olduğu görülmektedir. Birisi eğitim alt kümesine, diğeri ise geçerlik alt kümesine aittir. Eğitim alt kümesine ait hata değeri dönem sayısı arttıkça monoton bir şekilde azalırken, geçerlik alt kümesine ait

hata değeri önce monoton bir şekilde azalarak minimum değere ulaşmakta daha sonra eğitim devam ettikçe artmaya başlamaktadır. İşte bu minimum noktadan sonra eğitim verisine gürültü de dahil olmaktadır ve bu noktada eğitimin durdurulması gerekir. Bu kurala “*Erken durdurma*” kuralı adı verilmektedir (Haykin, 1999: 216).

4.9 Öz örgütlemeli Harita Ağları (Self Organizing Maps)

Kohonen (1995) tarafından geliştirilen öz örgütlemeli harita (SOM) ağları, çok boyutlu özelliklere sahip nesnelere daha küçük boyutlu uzayda kümeleyen bir yöntemdir. Öz örgütlemeli haritaların amacı; bir küme içerisindeki örneklerin benzerlik derecesini maksimize etmek, farklı kümelere ait örneklerin benzerliklerini minimize etmek ve sonuçların daha küçük boyutlu bir uzayda temsil edilmesini sağlamaktır (Hsu vd., 2009: 7949).

SOM ağları, geri beslemeli bir yapay sinir ağı modeli olmakla birlikte bölüm 2.5.2’de anlatılan denetimsiz öğrenme yöntemlerinden biridir ve bölüm 4.4’de anlatılan yarışmacı öğrenme kuralını kullanmaktadır.

SOM ağları çoğunlukla sınıflandırma yapmak amacıyla kullanılmaktadır. Sınıflandırmanın dışında özellik çıkarımı, örüntü tanıma, konuşma analizi ve ilgili birçok uygulamada kullanılmaktadır. Bazı araştırmalarda Som ağlarının zaman serilerinin kısa dönemli talep tahmininde kullanıldığı da görülmektedir. Ekonomi, fizik, kimya alanlarında ve medikal uygulamalarda veri analizlerinin yapılmasında sıklıkla tercih edilmektedir (Simon vd., 2005:1797) (Ismail vd., 2011: 10575) (Chen, 2012: 2275).

4.9.1 Özörgütlemeli Harita Ağlarının Yapısı

SOM ağlarının tasarımı insan beyninin serebral korteksinin fonksiyonlarından esinlenerek yapılmıştır. Serebral korteks görme, duyma, dokunma gibi sinyalleri işlemek için farklı bölümlere ayrılmaktadır. Korteks bu sinyalleri aldığı ilk önce onları sınıflandırmakta daha sonra işlenmesi için ilgili bölümlere eşleştirmektedir. Korteksin her bir bölümünde, benzer işleve sahip nöronlar yakından ilişkilidir ve sinyallerin hızlı ve doğru bir şekilde işlenmesini sağlar. Gelen sinyallerin

sınıflandırılarak ilgili işleme bölümlerine eşleştirilmesine “*Topografik haritalama*” denilmektedir ve SOM ağlarının temel konseptini oluşturmaktadır (Yu vd., 2015: 83).

Temel olarak SOM ağları girdi katmanı ve çıktı katmanından oluşmaktadır. Girdi katmanı, dışarıdan çok boyutlu bir uzaydan gelen girdi sinyallerini kabul etmekte, çıktı haritası olarak da adlandırılan çıktı katmanı ise bu sinyalleri genellikle iki boyutlu bir yapı şeklinde düzenlemektedir. Her giriş nöronu her çıkış nöronuyla bağlantılıdır ve her bir bağlantı bir ağırlık değeriyle ilişkilendirilmektedir.

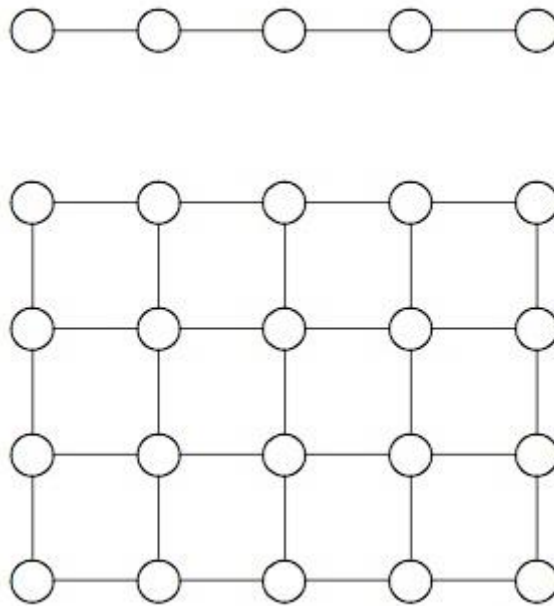
Çıkış nöronları kendi kendilerini örgütleyerek bir harita düzenlemektedirler ve benzer ağırlıktaki nöronlar birbirlerine yakın olarak konumlanmaktadır (Ismail vd., 2011: 10575). Başka bir ifadeyle nöronlar komşuluk ilişkileri vasıtasıyla birbirleriyle bağlantılıdır. Bu komşuluk ilişkisine ise “*Topoloji*” adı verilmektedir. SOM ağlarının eğitilmesi topoloji tarafından yüksek derecede etkilenmektedir. Şekil 32’de tek boyutlu ve iki boyutlu topoloji örnekleri yer almaktadır (Kriesel, 2005: 172).

Şekil 32 - Özörgütlemeli Harita Ağlarında Topoloji Örnekleri

Kaynak: Kriesel, D., “A Brief Introduction Neural Networks”, (Çevrimiçi),

<http://www.dkriesel.com/media/science/neuronaleetze-en-zeta2-2col-dkrieselcom.pdf>, 27 Mayıs

2015, p.173.

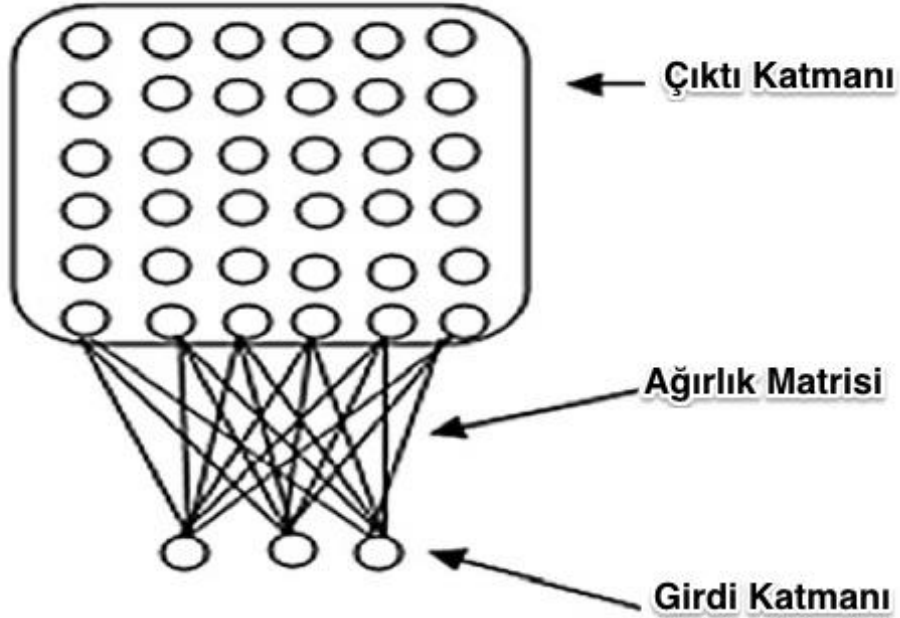


SOM ağlarının genel yapısı ise Şekil 33’de yer almaktadır. Şekil 33’de girdi katmanının m sayıda nöron ve çıktı katmanının ise $n_x \times n_y$ sayıda x satır ve y sütundan oluşan dikdörtgen bir harita şeklinde düzenlendiğini düşünelim. Girdi katmanındaki her bir nöron çıktı katmanındaki her bir nöronla bağlantılı olduğu için, çıktı katmanındaki her bir nöronun m kadar girdi katmanı ile bağlantısı bulunmaktadır. Bu bağlantıların her biri sineptik ağırlıklara sahiptir.

Giriş nöronu i ile çıkış nöronu j arasındaki bağlantının ağırlığı ise w_{ij} olarak gösterilmektedir. Başlangıçta bu bağlantı ağırlıklarına rastgele değerler verilmekte ve ağırlık eğitimi boyunca bu değerler düzeltilmektedir (Chen, 2012: 2275).

Şekil 33 - Özörgütlemeli Harita Ağlarının Yapısı

Kaynak: Ismail, S., Shabri, A. Ve Samsudin, R., “A Hybrid Model Of Self-organizing Maps (SOM) And Least Square Support Vector Machine (LSSVM) For Time-Series Forecasting”, **Expert Systems with Applications**, C: 38, 2011, p. 10575.



4.9.2 Özörgütlemeli Harita Ağlarının Eğitilmesi

SOM ağlarının eğitilmesi, bağlantı ağırlıklarına rastgele küçük değerler atandıktan sonra rekabet, işbirliği, uyum olmak üzere üç aşamadan oluşmaktadır (Yu vd., 2015: 83) (Haykin, 1999: 447-448):

- 1) **Rekabet:** Her girdi örneği için ağdaki nöronlar diskriminant fonksiyonunun ilgili değerlerine göre birbirleriyle yarışmaktadırlar. Bu diskriminant fonksiyonu nöronlar arasındaki rekabetin temelini oluşturmaktadır. Diskriminant fonksiyonunun en büyük değerini alan nöron yarışmanın kazananı olarak ilan edilmektedir.
- 2) **İşbirliği:** Yarışmanın kazananı ilan edilen nöron, uyarılmış nöronların topolojik komşuluğunun mekansal konumunu belirler ve böylece komşu nöronlar arasındaki işbirliğinin temellerini sağlar.
- 3) **Uyum:** Bu süreç uyarılmış nöronların, girdi örneklerinin bağlantı ağırlıklarıyla ilgili uygun ayarlamaları boyunca diskriminant fonksiyon değerlerini arttırmalarını sağlamaktadır.

4.9.3 Özörgütlemeli Harita Ağlarının Algoritması

n boyutlu bir girdi uzayında, girdi örneklerinin $x = \{x: i = 1, \dots, n\}$ ve girdi toplam nöron m ile ifade edildiğini, girdi nöronu i ile çıktı nöronu j arasındaki bağlantı ağırlıklarının $w_j = \{w_{ji} = j: 1, \dots, m; i = 1, \dots, n\}$ olarak tanımlandığını düşünelim. SOM ağlarının algoritma adımları aşağıdaki şekildedir oluşmaktadır (Haykin, 1999: 454) (Rojas, 1996: 396) (Wali, 2014: 136):

Başlangıç Adımı: İlk adımda her bir bağlantı ağırlığı w_j için rastgele seçilmiş değerler atanır.

Örnekleme Alma Adımı: x girdi vektöründen bir örnek alınarak ağa gösterilir.

Eşleştirme Adımı: x girdi vektörüyle her nöronun ağırlık vektörleri (w_j) arasındaki uzaklıklar Öklid mesafesi kullanılarak aşağıdaki şekilde hesaplanır:

$$d_j(x) = \sum_{i=1}^n (x_i - w_{ji})^2 \quad (4.33)$$

Ağırlık vektörü girdi vektöründe en yakın mesafede olan nöron kazanan nöron olarak ilan edilir.

Güncelleme Adımı: Kazanan nöron ilan edildikten sonra kazanan nöron ve bu nöronun komşu nöronlarının ağırlıkları aşağıdaki şekilde güncellenir (Chen, 2012: 2276).

$$w_{(t+1)} = w_{(t)} + \eta(t) \cdot T_{j,k(x)}(t) \cdot (x_i - w_{ji}) \quad (4.34)$$

4.34 numaralı denklemde yer alan t dönemi, $\eta(t)$ öğrenme katsayısını, $T_{j,k(x)}(t)$ ise j nöronuyla $k(x)$ kazanan nöronu arasındaki uzaklığa bağlı topolojik komşuluğu ifade etmektedir.

$j = k(x)$ olması durumunda ise $T_{j,k(x)}(t) = 1$ olur ve $|j - k(x)|$ değeri arttıkça $T_{j,k(x)}(t)$ değeri azalmaktadır ve Gauss fonksiyonuna dayanarak aşağıdaki şekilde gösterilmektedir (Alpaydın, 2012: 241) (Öztemel, 2012:182). Gauss fonksiyonu dışında kullanılan diğer topoloji fonksiyon örnekleri Şekil 34’de yer almaktadır (Kriesel, 2005: 178).

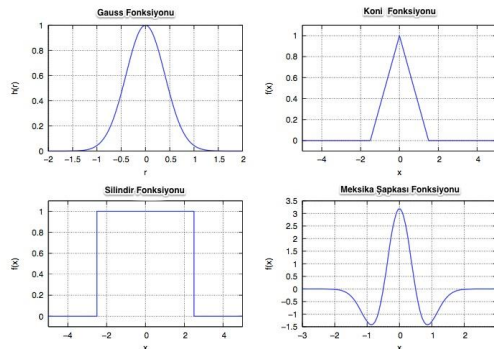
$$T_{j,k(x)}(t) = \exp(-\|d_j - d_{k(x)}\|/\sigma^2) \quad (4.35)$$

Şekil 34 - Topoloji Fonksiyonları

Kaynak: Kriesel, D., “A Brief Introduction Neural Networks”, (Çevrimiçi),

<http://www.dkriesel.com/media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf>, 27 Mayıs

2015, p.178.

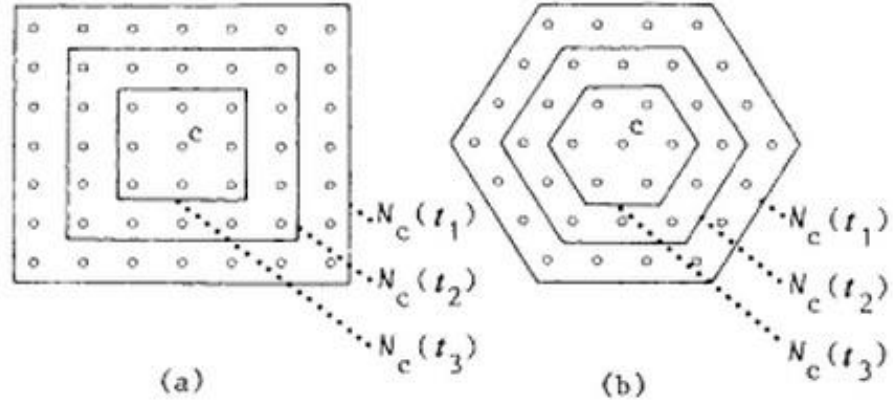


4.35 numaralı formüldeki d_j ve $d_{k(x)}$, j ve $k(x)$ nöronlarının pozisyonlarını göstermekte, σ ise topolojik komşuluğun genişliğini tanımlamaktadır. Yakınsama için topolojik komşuluğun genişliği eğitim sırasında zamanla iterasyonlarla azaltılmaktadır. Aynı şekilde öğrenme katsayısı değerinin de iterasyonlarla zaman içinde azaldığı görülür.

Şekil 35’de bir nöronun dikdörtgen ve altıgen biçiminde belirlenmiş topolojik komşuluk örnekleri yer almaktadır. C kazanan nöronu N_c ise mekânsal komşuluğu temsil etmektedir ve bu nöronun komşu nöronları dikdörtgen ve altıgen içinde yer alan nöronlardan oluştuğu görülmektedir. N_c ’nin yarıçapı her yinelemede azalmaktadır. $t_1 < t_2 < t_3$ olması durumunda $N_c(t_3) < N_c(t_2) < N_c(t_1)$ olmaktadır. Eğitim başladığında mekânsal komşuluğun yarıçapı büyükken zamanla küçülmektedir (Zurada, 1992: 424).

Şekil 35 - Bir Nöronun Topolojik Komşuluk Örnekleri ($t_1 < t_2 < t_3$)

Kaynak: Kohonen, T., *Self-Organization And Associative Memory*, Third Edition, Berlin: Springer-Verlag, 1989., p.132.

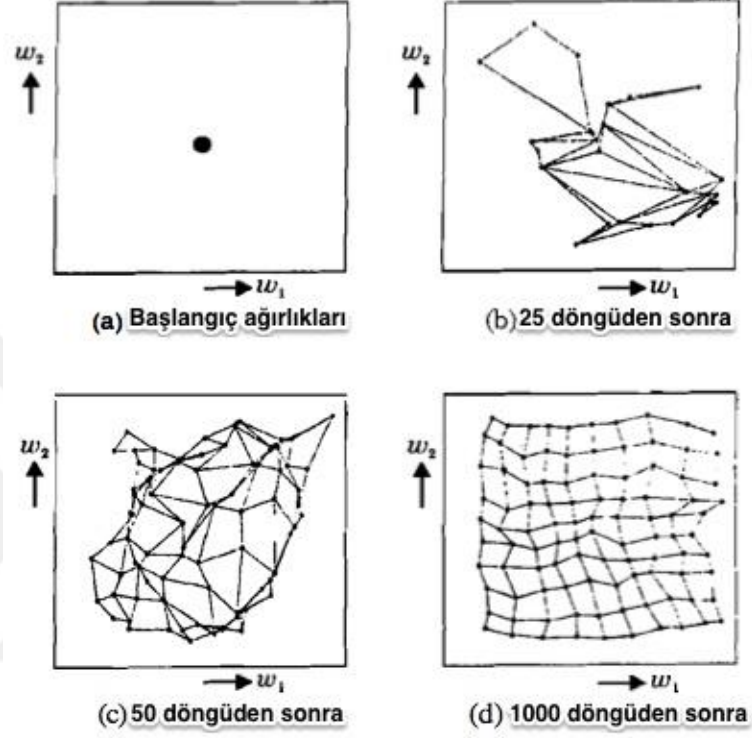


Sürdürme Adımı: Yeniden 2. Adıma dönülür ve bu süreç hedeflenen amaca ulaşıncaya kadar tekrar edilir.

Şekil 36’da 2 boyutlu girdi uzayından 2 boyutlu çıktı uzayına haritalama örneği, Şekil 37’de 1 boyutlu girdi uzayından 1 boyutlu çıktı uzayına haritalama örneği ve Şekil 38’de ise 2 boyutlu girdi uzayından 1 boyutlu çıktı uzayına haritalama örneği yer almaktadır.

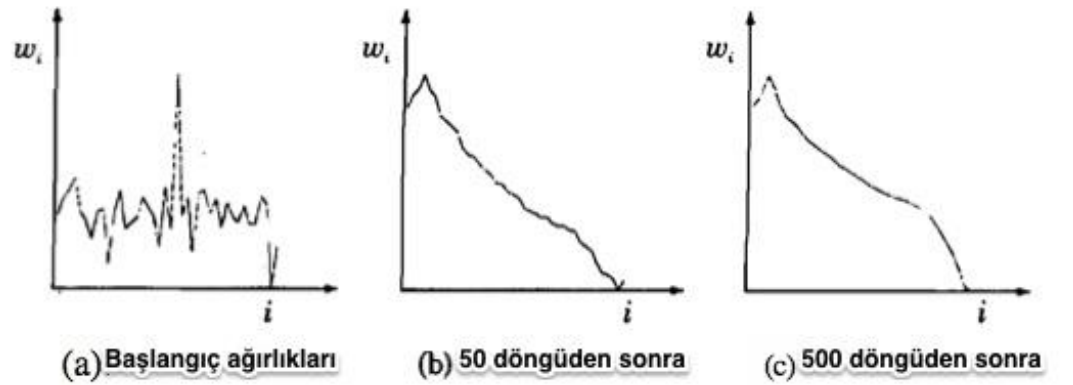
Şekil 36 - SOM Ağları Haritalama Örneği 1

Kaynak: Yegnanarayana, B., *Artificial Neural Network*, Eastern Economy Edition, New Delhi, India, Prentice-Hall of India, 2005, p.227.



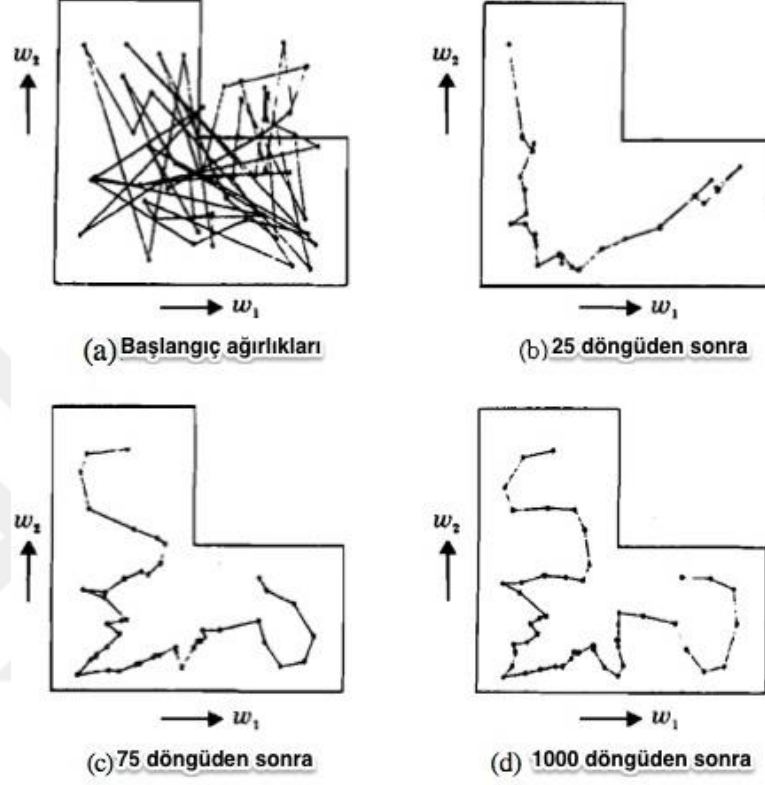
Şekil 37 - SOM Ağları Haritalama Örneği 2

Kaynak: Yegnanarayana, B., *Artificial Neural Network*, Eastern Economy Edition, New Delhi, India, Prentice-Hall of India, 2005, p.228.



Şekil 38 - SOM Ağları Haritalama Örneği 3

Kaynak: Yegnanarayana, B., **Artificial Neural Network**, Eastern Economy Edition, New Delhi, India, Prentice-Hall of India, 2005, p.228.



BEŞİNCİ BÖLÜM

UYGULAMA

Bu tezin uygulamasında e-ticaret sektöründe faaliyet gösteren bir işletmenin aralıklı talep yapısına sahip ürünlerinin talep tahmininde; SES, Croston yöntemi ve modifikasyonları, DVM ve bir yapay sinir ağı modeli olan ÇKA ve SOM ağları kullanılarak talep tahmini yapılmıştır. Daha sonra bu yöntemlerin hataları karşılaştırılarak performansları ölçülmüştür.

5.1 Uygulamanın Amacı ve Kapsamı

Talep tahmini ve planlama, işletmelerin başarısını doğrudan etkilemekte ve müşteri memnuniyetinin artırılmasında önemli rol oynamaktadır. Düzgün talep yapısına sahip ürünlerin talep tahmini için birçok yöntem başarılı sonuçlar verirken aralıklı talep yapısına sahip ürünlerin talep tahmininde bu yöntemler başarılı sonuçlar vermemektedir. Klasik zaman serilerinden farklı olan ve çoğu zaman diliminde sıfır talep gören aralıklı talebin tahmini ve planlaması talep büyüklüğünün ve şeklinin değişkenliğinden dolayı oldukça zordur.

Bu çalışmada, e-ticaret sektöründe faaliyet gösteren bir işletmenin aralıklı talep yapısına sahip ürünlerinin talep tahmini ve planlaması için öncelikle basit üstel düzeltme yöntemi ve literatürde sıklıkla kullanılan aralıklı talep tahmini için geliştirilmiş Croston yöntemi, Syntetos & Boylan ve daha sonra Leven & Segerstedt tarafından değişikliğe uğratılan Croston yönteminin modifikasyonları uygulanmıştır. Daha sonra gelişmiş makine öğrenme yöntemlerinden DVM ve bir yapay sinir ağı modeli olan ÇKA ve SOM ağları kullanılıp her bir yöntemin performansı ölçülerek hangi yöntemin daha başarılı sonuçlar verdiği araştırılmıştır.

5.2 Veri Setinin Hazırlanması

Çalışmada Türkiye'nin önde gelen bir e-ticaret işletmesinin gerçek verileri kullanılmıştır. Bahsi geçen e-ticaret işletmesi 9 ana kategoride satış yapmaktadır. Her

bir ana kategorinin altında alt kategoriler yer almaktadır. Aralıklı talebe sahip ürünlerin toplam stok değerinin büyük bir kısmının genellikle havacılık, otomotiv, askerlik ve bilgi işlem sektöründe rastlandığı bilgisinden yola çıkarak bu işletmenin *Elektronik* ana kategorisinin altında yer alan *Bilgisayarlar* alt kategorisinin verileri kullanılmıştır. Bu kategoride tablet, masaüstü, dizüstü bilgisayar, monitör ve mini masaüstü olmak üzere toplam 640 adet ürün mevcuttur ve her bir ürünün 01.05.2014 - 31.05.2015 tarihleri arasındaki toplam 396 gün için günlük talep miktarları temin edilmiştir.

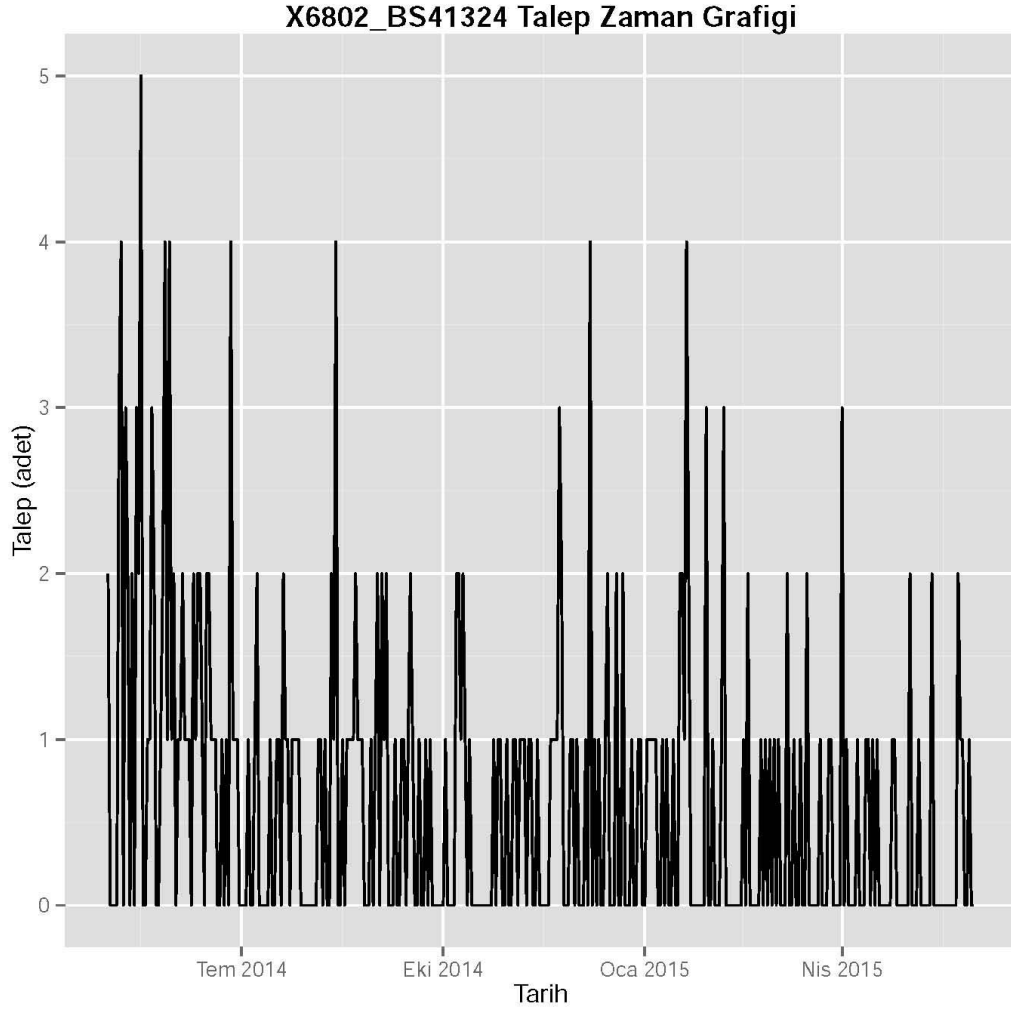
5.3 Aralıklı Talebin Sınıflandırılması

Bilgisayarlar alt kategorisinde 640 adet ürün bulunmaktadır. Bu ürünlerden aralıklı talep yapısına sahip olan ürünlerin belirlenebilmesi için Syntetos ve diğerleri (2005) tarafından geliştirilen sınıflandırma tekniği kullanılmıştır. Bu sınıflandırma talebi; değişken talep, düzgün talep, düzgün olmayan talep ve aralıklı talep olmak üzere dört kategoriye ayırmaktadır ve bu sınıflandırmayı yaparken iki parametreyi esas almaktadır. Bu parametrelerden birisi talebin sıklığı yani iki ardışık talep arasında geçen zaman periyodlarının ortalaması (p), diğeri ise talep büyüklüğünün değişkenlik/varyasyon katsayısının karesi (CV^2) parametreleridir. Bu teknikte bu parametreler için belirlenen eşik değerler $p = 1.32$ ve $CV^2 = 0.49$ değerleridir ve $p > 1.32$, $CV^2 \leq 0.49$ durumunu sağlayan ürünler aralıklı talep yapısına sahip olan ürünlerdir.

640 adet ürün için p ve CV^2 değerleri hesaplandığında $p > 1.32$, $CV^2 \leq 0.49$ koşuluna uyan toplam 98 adet ürün tespit edilmiştir. Başka bir ifadeyle 640 adet üründen 98 adet ürün aralıklı talep yapısına sahiptir. Bu hesaplama yapılırken xlsx formatında gelen veriler R programı ile okunup data.frame nesnesi içinde saklanmıştır ve talebin gerçekleşmediği günlere 0 değer atanmıştır. Daha sonrasında her bir ürün için p ve CV^2 değerleri hesaplatılmıştır.

Şekil 39’da aralıklı talep yapısına sahip bir ürünün 01.05.2014 - 31.05.2015 tarihleri arasındaki gün bazındaki talep miktarlarının grafiği yer almaktadır. Grafikten de anlaşılacağı gibi aralıklı talep yapısına sahip ürünlerin talebi rastgele gerçekleşmekte ve çoğu zaman diliminde talep miktarı sıfır olmaktadır.

Şekil 39 - Aralıklı Talebe Sahip Bir Ürünün Talep Zaman Grafiği

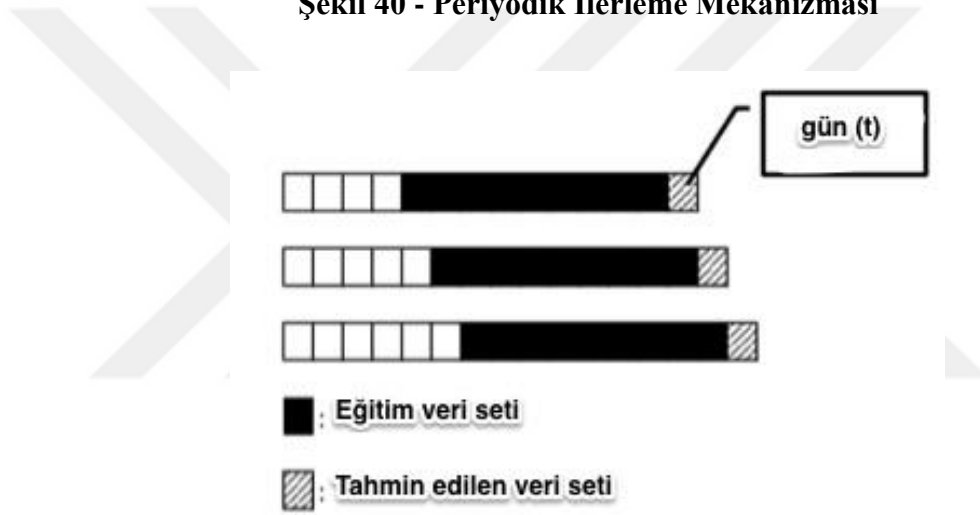


5.4 Veri Analizi ve Bulgular

640 adet ürün için p ve CV^2 değerleri hesaplanıp $p > 1.32$, $CV^2 \leq 0.49$ koşuluna uyan toplam 98 adet ürün tespit edildikten sonra elde edilen aralıklı talebe sahip ürünlerin talep verilerinin %80'i eğitim için kalan %20'si de test için ayrılmıştır. 01.05.2014 - 31.05.2015 tarihleri arasındaki 396 günlük talebin ilk 316 günlük bölümü eğitim kümesi, son 80 günlük bölümü ise test kümesi olarak kullanılmıştır. Veri üzerinde sırasıyla SES, Croston yöntemi, Syntetos & Boylan ve daha sonra Leven & Segerstedt tarafından değişikliğe uğratılan Croston yönteminin modifikasyonları, gelişmiş

makine öğrenme yöntemlerinden DVM ve bir yapay sinir ağı modeli olan ÇKA ve SOM ağıları uygulanmıştır. Uygulamada tüm yöntemler için periyodik ilerleme (rolling) mekanizması kullanılmıştır. Periyodik ilerleme mekanizması tahmin doğruluğunun artırılması için kullanılan etkili bir yöntemdir. Düzensiz bir yapıya sahip verinin olması durumunda talep tahmin doğruluğunu arttırmak için her yeni dönemi tahmin ederken en son döneme ait veriyi de kullanarak tahmin gerçekleştirilmektedir (Akay & Atak, 2007: 1672). Periyodik ilerleme mekanizması şekil 40'da gösterilmektedir.

Şekil 40 - Periyodik İlerleme Mekanizması



5.4.1 Basit Üstel Düzeltme Yöntemi

Basit üstel düzeltme yöntemi (SES) kısa dönemli talep tahmininde en yaygın olarak kullanılan istatistiksel yöntemlerden biridir. Her yeni tahmin, bir önceki tahmin ile tahminin ve gerçekleşen değer arasındaki farkın yüzdesinin toplamına eşittir. Aşağıdaki şekilde göstermek mümkündür (Stevenson, 2009: 83-84) :

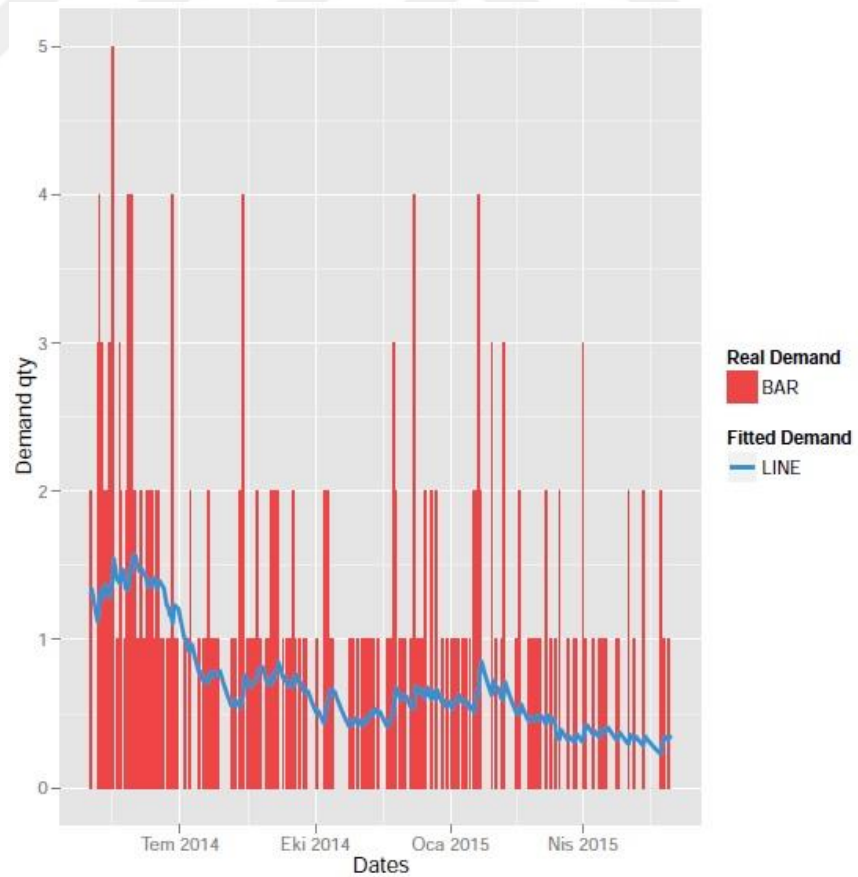
$$F_t = F_{t-1} + \alpha (D_{t-1} - F_{t-1}) \quad (5.1)$$

F_t t dönemi için tahmin değerini, F_{t-1} bir önceki dönem için tahmin değerini, D_{t-1} bir önceki dönem için gerçekleşen talep değerini ve $(D_{t-1} - F_{t-1})$ tahmin hatasını temsil etmektedir. α değeri ise düzeltme katsayısıdır ve 0 ile 1 arasında bir değer almaktadır. Düzeltme katsayısı geçmiş verilerin etkisini düzene sokmaktadır.

Düşük bir değere sahip olan düzeltme katsayısı geçmiş verilerin üzerinde durulmasını sağlayarak daha çok sabit talep için uygundur, fakat sistematik değişikliklere karşı hızlı tepki verilememesine neden olacaktır. Yüksek bir değere sahip olan düzeltme katsayısı ise en son yapılan gözlemlerin üzerinde durulmasını sağlayarak talep değiştiğinde hızlı tepki verilmesi gereken durumlar için uygundur (Wallström & Segerstedt, 2010: 626).

SES yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 3’de, eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 41’de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgiler ise tahmin değerlerini ifade etmektedir.

**Şekil 41 - SES Yöntemine Göre Bir Ürünün Talep Zaman Grafiği
(Gerçekleşen ve Tahmini Talep)**



Tablo 3 - SES Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	SES	Gün	Gerçekleşen	SES	Gün	Gerçekleşen	SES	Gün	Gerçekleşen	SES
1	1	0,4343	21	0	0,4147	41	0	0,3485	61	2	0,2848
2	0	0,3543	22	1	0,3979	42	0	0,3349	62	0	0,3509
3	0	0,3398	23	0	0,4222	43	1	0,3217	63	0	0,3376
4	2	0,3259	24	0	0,4051	44	1	0,3483	64	0	0,3247
5	0	0,3942	25	0	0,3889	45	0	0,3736	65	0	0,3124
6	0	0,3780	26	0	0,3735	46	0	0,3591	66	0	0,3004
7	0	0,3626	27	1	0,3588	47	0	0,3451	67	0	0,2888
8	0	0,3479	28	0	0,3840	48	0	0,3317	68	0	0,2777
9	0	0,3338	29	0	0,3689	49	0	0,3188	69	0	0,2669
10	1	0,3203	30	0	0,3544	50	0	0,3063	70	0	0,2564
11	0	0,3478	31	1	0,3405	51	2	0,2942	71	0	0,2463
12	0	0,3338	32	1	0,3664	52	0	0,3607	72	0	0,2365
13	0	0,3203	33	0	0,3912	53	0	0,3469	73	2	0,2271
14	1	0,3073	34	1	0,3759	54	0	0,3335	74	1	0,2965
15	1	0,3353	35	0	0,4004	55	1	0,3207	75	1	0,3232
16	0	0,3621	36	1	0,3846	56	0	0,3468	76	0	0,3488
17	0	0,3475	37	0	0,4088	57	0	0,3335	77	0	0,3356
18	0	0,3335	38	0	0,3927	58	0	0,3207	78	1	0,3229
19	0	0,3201	39	0	0,3773	59	0	0,3083	79	0	0,3484
20	3	0,3072	40	0	0,3626	60	0	0,2964	80	0	0,3353

5.4.2 Croston Yöntemi

Literatürdeki çalışmalardan düzgün talep yapısına sahip ürünlerin talep tahmininde kullanılan klasik yöntemlerin aralıklı talep yapısına sahip ürünlerin talep tahmininde iyi sonuçlar vermediği bilinmektedir. Bu sebeple veri setine aralıklı talep tahmini için geliştirilmiş Croston yöntemi uygulanmıştır.

Croston yöntemi için R programındaki forecast paketinin Croston fonksiyonu kullanılmıştır. Bu fonksiyon ile alfa değeri de fonksiyon içerisinde kestirilmektedir. Croston tarafından aralıklı talep yapısına sahip ürünler için geliştirilen yöntemin algoritması 5.2 ve 5.3 numaralı denklemlerde yer almaktadır (Croston, 1972: 302) :

t = zaman aralıklarının gözden geçirilmesi

η = talebin sıfır olmadığı dönemlerin sayıları

$y_{t=}$ t dönemi için tahmin

Z_{η} = ortalama (μ)ve standart sapma (σ) ile talep miktarının sıfır olmayan gözlemleri

p = gelişler arasındaki sürenin ortalaması

q = talepler arasındaki süre (p ortalamayla geometrik dağılıma uygun dağılır.)

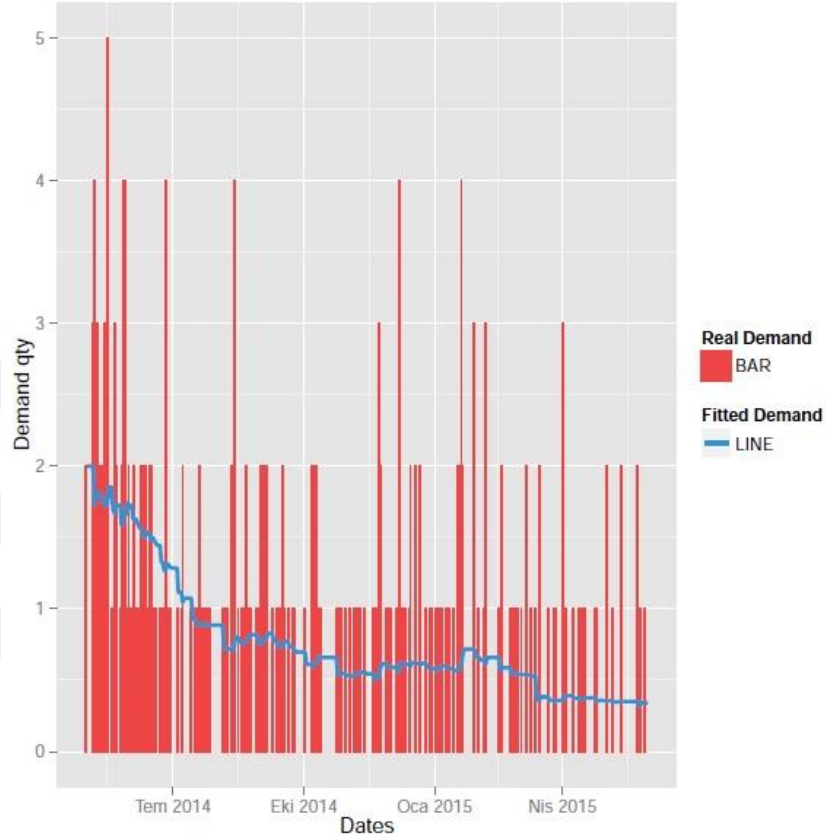
$$\left. \begin{aligned} \bar{z}_{\eta} &= \alpha y_t + (1 - \alpha) \bar{z}_{\eta-1} \\ \bar{p}_{\eta} &= \bar{p}_{\eta-1} (1 - \alpha) + \alpha q \\ \bar{y}_t &= \bar{z}_{\eta} / \bar{p}_{\eta} \\ q &= 1 \end{aligned} \right\} y_t \neq 0 \quad (5.2)$$

$$\left. \begin{aligned} \bar{z}_{\eta} &= \bar{z}_{\eta-1} \\ \bar{p}_{\eta} &= \bar{p}_{\eta-1} , \\ q &= q + 1 \end{aligned} \right\} y_t = 0 \quad (5.3)$$

Croston yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 4’de, eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 42’de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgi ise tahmin değerlerini ifade etmektedir.

Croston yönteminde talebin sıfır olduğu zaman periyotlarından sonra tahminler geçersiz kalmakta ve stok tükenme riskiyle karşılaşılan durumlarda elverişli olmamaktadır. Tahminler sadece talep meydana geldiğinde güncellenmekte ve talebin her dönemde güncellenmesi durumunda basit üstel düzeltme yöntemiyle aynı olmaktadır.

**Şekil 42 - Croston Yöntemine Göre Bir Ürünün Talep Zaman Grafiği
(Gerçekleşen ve Tahmini Talep)**



Tablo 4 - Croston Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	Croston	Gün	Gerçekleşen	Croston	Gün	Gerçekleşen	Croston	Gün	Gerçekleşen	Croston
1	1	0,5279	21	0	0,3898	41	0	0,3761	61	2	0,3482
2	0	0,3616	22	1	0,3898	42	0	0,3761	62	0	0,3514
3	0	0,3616	23	0	0,3915	43	1	0,3761	63	0	0,3514
4	2	0,3616	24	0	0,3915	44	1	0,3492	64	0	0,3514
5	0	0,3825	25	0	0,3915	45	0	0,3578	65	0	0,3514
6	0	0,3825	26	0	0,3915	46	0	0,3578	66	0	0,3514
7	0	0,3825	27	1	0,3915	47	0	0,3578	67	0	0,3514
8	0	0,3825	28	0	0,3735	48	0	0,3578	68	0	0,3514
9	0	0,3825	29	0	0,3735	49	0	0,3578	69	0	0,3514
10	1	0,3825	30	0	0,3735	50	0	0,3578	70	0	0,3514
11	0	0,3578	31	1	0,3735	51	2	0,3578	71	0	0,3514
12	0	0,3578	32	1	0,3634	52	0	0,3568	72	0	0,3514
13	0	0,3578	33	0	0,3724	53	0	0,3568	73	2	0,3514
14	1	0,3578	34	1	0,3724	54	0	0,3568	74	1	0,3195
15	1	0,3485	35	0	0,3743	55	1	0,3568	75	1	0,3323
16	0	0,3593	36	1	0,3743	56	0	0,3482	76	0	0,3421
17	0	0,3593	37	0	0,3761	57	0	0,3482	77	0	0,3421
18	0	0,3593	38	0	0,3761	58	0	0,3482	78	1	0,3421
19	0	0,3593	39	0	0,3761	59	0	0,3482	79	0	0,3400
20	3	0,3593	40	0	0,3761	60	0	0,3482	80	0	0,3400

5.4.3 Croston Yönteminin Modifikasyonları

Croston (1972) hem talep büyüklüğünü hem de talebin meydana gelme sıklığını göz önünde bulundurarak aralıklı talep yapısına sahip ürünler için bir yöntem geliştirmiştir. Croston yönteminin böyle bir tahmin süreci için teorik üstünlüğü olmasına rağmen diğer basit talep tahmin yöntemleriyle kıyaslandığında gözlemsel kanıtlar, bu yöntemin biraz daha iyi sonuçlar verdiğini fakat bazı gözlemlerde başarısız kaldığını göstermektedir. Syntetos & Boylan (2001) ve Leven & Segerstedt (2004) tarafından bu yöntem geliştirilmeye çalışılmıştır.

5.4.3.1 Syntetos & Boylan Modifikasyonu

Syntetos ve Boylan (2001) Croston yöntemini geliştirebilmek amacıyla yöntem üzerinde bir çalışma yapmış ve talebin beklenen tahminindeki matematiksel türevinde bir hata bulmuşlardır (Syntetos & Boylan, 2001: 457-458).

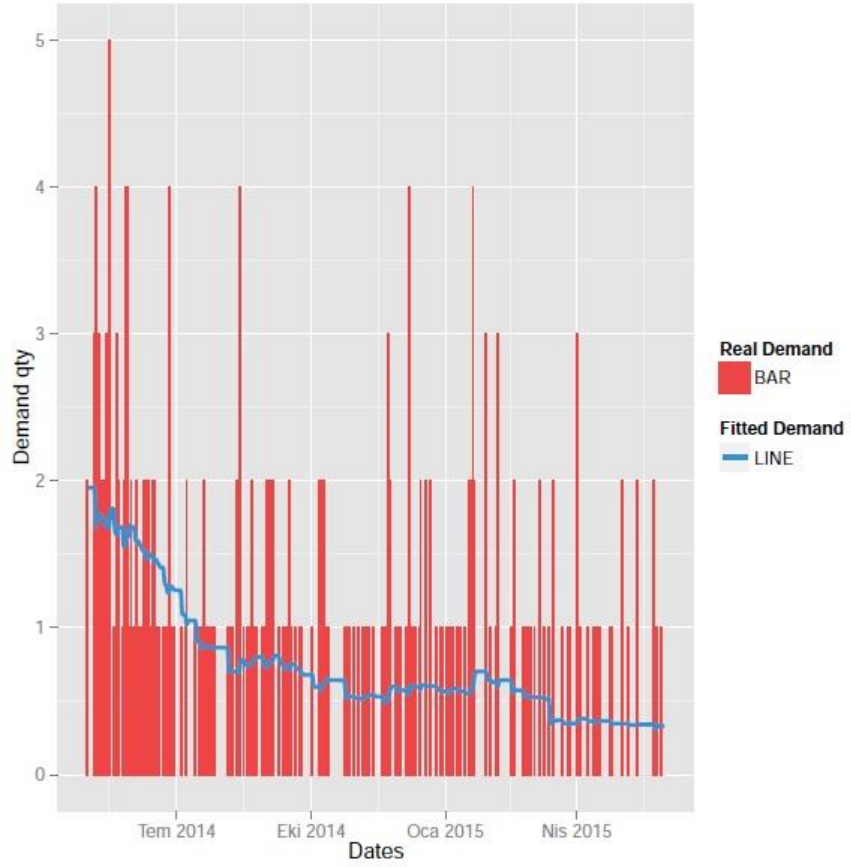
Croston yönteminde yer alan yanılığın düzeltmek amacıyla Syntetos ve Boylan (2005) Croston yöntemine aşağıda yer alan denklemi yönteme eklemiştir. Bu yöntem “Yakınlık yöntemi” olarak da bilinmektedir. Aşağıdaki denklemde yer alan α , \bar{p}_η için düzgünleştirme parametresidir (Babiloni vd., 2010: 122).

$$\bar{y}_t = \left(1 - \frac{\alpha}{2}\right) \frac{\bar{z}_\eta}{\bar{p}_\eta} \quad (5.4)$$

Croston Syntetos ve Boylan yöntemi için R programındaki forecast paketinin Croston fonksiyonuna 5.4 numaralı denklem eklenerek hesaplama yapılmıştır ve optimize edilmiş alfa değerleri kullanılmıştır.

Croston S&B yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 5’de, eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 43’de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgi ise tahmin değerlerini ifade etmektedir.

Şekil 43 - Croston S&B Yöntemine Göre Bir Ürünün Talep Zaman Grafiği (Gerçekleşen ve Tahmini Talep)



Tablo 5 - Croston S&B Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	CrostonS&B	Gün	Gerçekleşen	CrostonS&B	Gün	Gerçekleşen	CrostonS&B	Gün	Gerçekleşen	CrostonS&B
1	1	0,5156	21	0	0,3789	41	0	0,3663	61	2	0,3390
2	0	0,3515	22	1	0,3789	42	0	0,3663	62	0	0,3419
3	0	0,3515	23	0	0,3808	43	1	0,3663	63	0	0,3419
4	2	0,3515	24	0	0,3808	44	1	0,3399	64	0	0,3419
5	0	0,3720	25	0	0,3808	45	0	0,3485	65	0	0,3419
6	0	0,3720	26	0	0,3808	46	0	0,3485	66	0	0,3419
7	0	0,3720	27	1	0,3808	47	0	0,3485	67	0	0,3419
8	0	0,3720	28	0	0,3631	48	0	0,3485	68	0	0,3419
9	0	0,3720	29	0	0,3631	49	0	0,3485	69	0	0,3419
10	1	0,3720	30	0	0,3631	50	0	0,3485	70	0	0,3419
11	0	0,3475	31	1	0,3631	51	2	0,3485	71	0	0,3419
12	0	0,3475	32	1	0,3532	52	0	0,3474	72	0	0,3419
13	0	0,3475	33	0	0,3624	53	0	0,3474	73	2	0,3419
14	1	0,3475	34	1	0,3624	54	0	0,3474	74	1	0,3098
15	1	0,3384	35	0	0,3644	55	1	0,3474	75	1	0,3230
16	0	0,3495	36	1	0,3644	56	0	0,3390	76	0	0,3331
17	0	0,3495	37	0	0,3663	57	0	0,3390	77	0	0,3331
18	0	0,3495	38	0	0,3663	58	0	0,3390	78	1	0,3331
19	0	0,3495	39	0	0,3663	59	0	0,3390	79	0	0,3311
20	3	0,3495	40	0	0,3663	60	0	0,3390	80	0	0,3311

5.4.3.2 Leven & Segerstedt Modifikasyonu

Leven ve Segerstedt (2004) de Croston yöntemi üzerinde bir düzeltme yaparak her bir dönemdeki ortalama talep için aşağıdaki denklemi yönteme eklemiştir (Babiloni vd., 2010: 123).

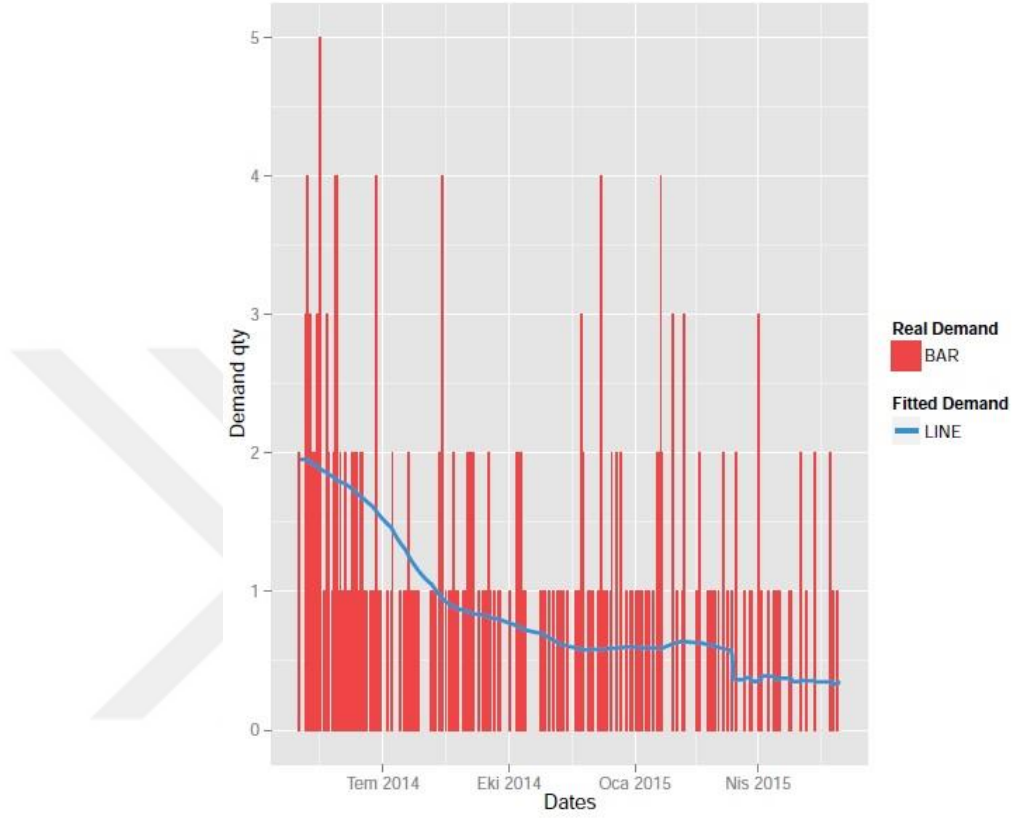
$$\bar{y}_t = \alpha \left(\frac{\bar{z}_\eta}{\bar{p}_\eta} \right) + (1 - \alpha) \bar{y}_{t-1} \quad (5.5)$$

Croston Syntetos ve Boylan yöntemi için R programındaki forecast paketinin Croston fonksiyonuna 5.5 numaralı denklem eklenerek hesaplama yapılmıştır ve optimize edilmiş alfa değerleri kullanılmıştır.

Croston yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 6'da eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 44'de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgi ise tahmin değerlerini ifade etmektedir.

R programındaki forecast paketinin SES, Croston ve modifikasyonları için her bir ürün için kestirilen optimum alfa değerleri ise Tablo 7'de yer almaktadır.

**Şekil 44 - Croston L&S Yöntemine Göre Bir Ürünün Talep Zaman Grafiği
(Gerçekleşen ve Tahmini Talep)**



Tablo 6 - Croston L&S Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	Croston L&S	Gün	Gerçekleşen	Croston L&S	Gün	Gerçekleşen	Croston L&S	Gün	Gerçekleşen	Croston L&S
1	1	0,5450	21	0	0,3610	41	0	0,3745	61	2	0,3561
2	0	0,3653	22	1	0,3610	42	0	0,3745	62	0	0,3478
3	0	0,3653	23	0	0,3910	43	1	0,3745	63	0	0,3478
4	2	0,3653	24	0	0,3910	44	1	0,3745	64	0	0,3478
5	0	0,3636	25	0	0,3910	45	0	0,3502	65	0	0,3478
6	0	0,3636	26	0	0,3910	46	0	0,3502	66	0	0,3478
7	0	0,3636	27	1	0,3910	47	0	0,3502	67	0	0,3478
8	0	0,3636	28	0	0,3898	48	0	0,3502	68	0	0,3478
9	0	0,3636	29	0	0,3898	49	0	0,3502	69	0	0,3478
10	1	0,3636	30	0	0,3898	50	0	0,3502	70	0	0,3478
11	0	0,3790	31	1	0,3898	51	2	0,3502	71	0	0,3478
12	0	0,3790	32	1	0,3727	52	0	0,3581	72	0	0,3478
13	0	0,3790	33	0	0,3653	53	0	0,3581	73	2	0,3478
14	1	0,3790	34	1	0,3653	54	0	0,3581	74	1	0,3455
15	1	0,3569	35	0	0,3728	55	1	0,3581	75	1	0,3255
16	0	0,3519	36	1	0,3728	56	0	0,3561	76	0	0,3356
17	0	0,3519	37	0	0,3745	57	0	0,3561	77	0	0,3356
18	0	0,3519	38	0	0,3745	58	0	0,3561	78	1	0,3356
19	0	0,3519	39	0	0,3745	59	0	0,3561	79	0	0,3424
20	3	0,3519	40	0	0,3745	60	0	0,3561	80	0	0,3424

Tablo 7 - SES, Croston ve Modifikasyonları İçin Optimum Alfa Değerleri

Ürün No	SES	Croston ve Modifikasyonları	Ürün No	SES	Croston ve Modifikasyonları
1	0,1832	0,1067	50	0,0001	0,0609
2	0,0727	0,0225	51	0,0837	0,0182
3	0,1473	0,0501	52	0,1472	0,0588
4	0,0534	0,0389	53	0,0244	0,0318
5	0,0749	0,1217	54	0,1212	0,1119
6	0,1993	0,1012	55	0,1110	0,1283
7	0,0001	0,0341	56	0,0859	0,0989
8	0,1035	0,1130	57	0,0943	0,0771
9	0,1508	0,1074	58	0,0653	0,0652
10	0,3429	0,0061	59	0,0693	0,1300
11	0,1450	0,0069	60	0,0427	0,0962
12	0,1370	0,0161	61	0,0857	0,0541
13	0,0346	0,1606	62	0,0001	0,0741
14	0,1499	0,0082	63	0,1365	0,0680
15	0,0640	0,0674	64	0,0994	0,1303
16	0,0902	0,0182	65	0,0862	0,1103
17	0,0277	0,0313	66	0,1089	0,2432
18	0,2639	0,0139	67	0,0001	0,0701
19	0,1875	0,0391	68	0,0001	0,0832
20	0,0766	0,0308	69	0,1037	0,0782
21	0,1499	0,0703	70	0,0705	0,0834
22	0,1136	0,0176	71	0,0571	0,1040
23	0,0936	0,0507	72	0,0589	0,0558
24	0,1550	0,1088	73	0,0001	0,0407
25	0,1129	0,1224	74	0,0936	0,1180
26	0,3301	0,0452	75	0,0408	0,1550
27	0,0590	0,0301	76	0,2421	0,0082
28	0,2351	0,0211	77	0,1497	0,0609
29	0,1213	0,0093	78	0,0993	0,0502
30	0,0262	0,0601	79	0,0867	0,1248
31	0,0732	0,0608	80	0,0810	0,0818
32	0,0509	0,0591	81	0,0918	0,0273
33	0,0155	0,0428	82	0,0490	0,0808
34	0,0245	0,0702	83	0,0001	0,0571
35	0,0355	0,0525	84	0,1307	0,1140
36	0,1398	0,0074	85	0,1288	0,1097
37	0,0442	0,0466	86	0,1316	0,1212
38	0,0747	0,0140	87	0,0990	0,1993
39	0,0245	0,0670	88	0,0137	0,0501
40	0,0512	0,0276	89	0,0001	0,0365
41	0,0343	0,0575	90	0,0857	0,1125
42	0,0465	0,1014	91	0,0716	0,1089
43	0,0825	0,0989	92	0,0819	0,0243
44	0,0359	0,0868	93	0,0607	0,0578
45	0,0138	0,0558	94	0,0997	0,1047
46	0,1996	0,0261	95	0,0457	0,0995
47	0,0831	0,0731	96	0,0524	0,0719
48	0,0257	0,0504	97	0,0739	0,0507
49	0,0244	0,0665	98	0,1285	0,1990

5.4.4 Destek Vektör Makinesi Yöntemi

DVM yöntemi için MATLAB 2014a ve R v3.2.2 araçları ile “LIBSVM v3.20 -- A Library for Support Vector Machines” paketi kullanılmıştır. Bu paket sınıflandırma problemleri için C-SVC ve ν -SVC, dağıtım tahmini için tek sınıflı SVM, regresyon için ν -SVR ve \mathcal{E} -SVR yöntemlerini desteklemektedir. LIBSVM uygulamada iki adımı içermektedir. İlk adım bir model oluşturabilmek için veri setinin eğitilmesi, ikinci adım ise bu modeli kullanarak test veri setinin bilgilerini tahmin etmektir.

Uygulamada öncelikle LIBSVM kütüphanesinden “svmtrain” fonksiyonu kullanılarak eğitim verisi üzerinden model oluşturulmuştur. Bu fonksiyon için “svm_type” olarak “4 -- nu-SVR” ve “kernel_type” olarak “2 -- radial basis function” olarak seçilmiş diğer parametreler varsayılan olarak kullanılmıştır. DVM için kullanılan tüm parametreler tablo 9’da üzeri sarı işaretlenerek belirtilmiştir.

Deneyle sırasında farklı parametre seçenekleri ile de modeller oluşturulmuş ve tahmin yapılmıştır. Ancak en iyi sonuçların bu parametre ayarları ile alındığı görülmüştür. Aşırı eğitim olup olmadığını görebilmek için 10 kat çapraz geçiş yapılarak aşırı eğitimin olmadığı görülmüştür.

Daha sonra oluşturulan model ve “svmpredict” fonksiyonu kullanılarak test verisi üzerinde tahminler yapılmıştır.

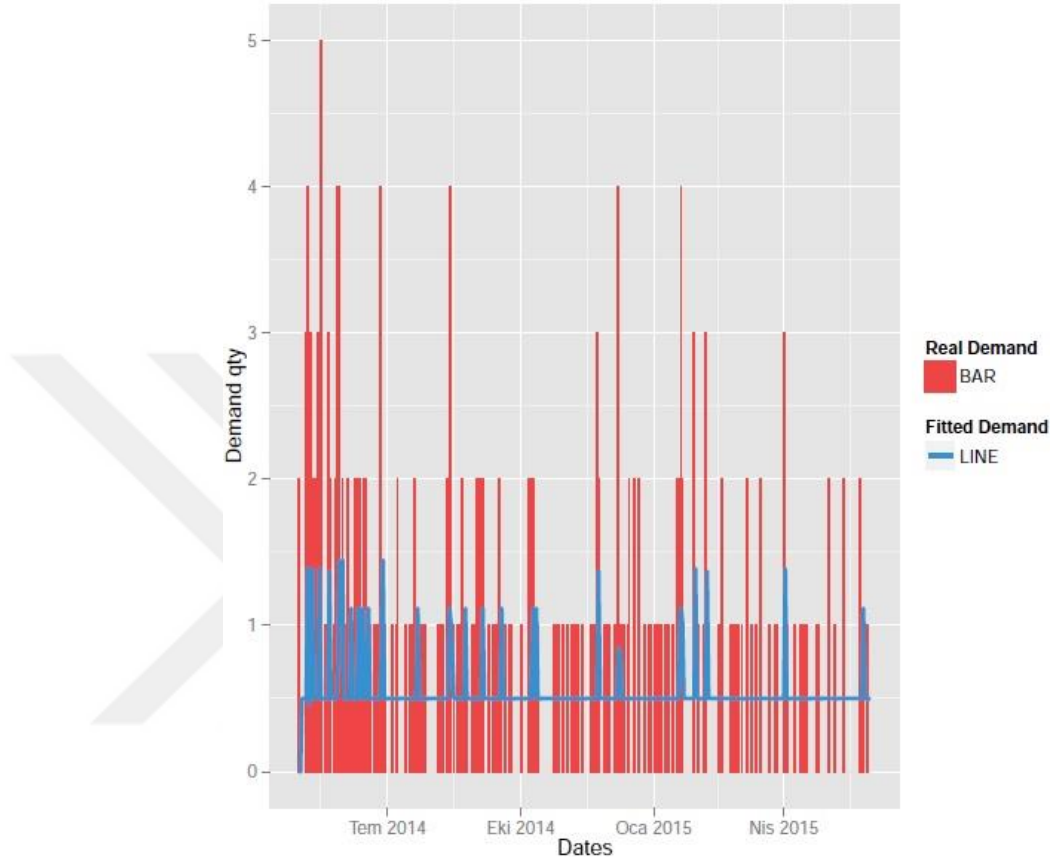
DVM yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 9’da eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 45’de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgi ise tahmin değerlerini ifade etmektedir.

Tablo 8 - DVM Yöntemi İçin Kullanılan Parametreler

options:

- s svm_type : set type of SVM (default 0)
 - 0 -- C-SVC (multi-class classification)
 - 1 -- nu-SVC (multi-class classification)
 - 2 -- one-class SVM
 - 3 -- epsilon-SVR (regression)
 - 4 -- nu-SVR (regression)
- t kernel_type : set type of kernel function (default 2)
 - 0 -- linear: $u \cdot v$
 - 1 -- polynomial: $(\gamma u \cdot v + \text{coef0})^{\text{degree}}$
 - 2 -- radial basis function: $\exp(-\gamma |u-v|^2)$
 - 3 -- sigmoid: $\tanh(\gamma u \cdot v + \text{coef0})$
 - 4 -- precomputed kernel (kernel values in training_set_file)
- d degree : set degree in kernel function (default 3)
- g gamma : set gamma in kernel function (default $1/\text{num_features}$)
- r coef0 : set coef0 in kernel function (default 0)
- c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
- n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
- p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
- m cachesize : set cache memory size in MB (default 100)
- e epsilon : set tolerance of termination criterion (default 0.001)
- h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
- b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
- wi weight : set the parameter C of class i to $\text{weight} \cdot C$, for C-SVC (default 1)
- v n: n-fold cross validation mode
- q : quiet mode (no outputs)

**Şekil 45 - DVM Yöntemine Göre Bir Ürünün Talep Zaman Grafiği
(Gerçekleşen ve Tahmini Talep)**



Tablo 9 - DVM Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	DVM	Gün	Gerçekleşen	DVM	Gün	Gerçekleşen	DVM	Gün	Gerçekleşen	DVM
1	1	0,5002	21	0	1,3887	41	0	0,5002	61	2	0,5002
2	0	0,4997	22	1	0,5002	42	0	0,5002	62	0	0,5003
3	0	0,4997	23	0	0,4997	43	1	0,5002	63	0	0,5004
4	2	0,5002	24	0	0,4997	44	1	0,4997	64	0	0,5002
5	0	0,5003	25	0	0,5002	45	0	0,4998	65	0	0,5002
6	0	0,5004	26	0	0,5002	46	0	0,4997	66	0	0,5002
7	0	0,5002	27	1	0,5002	47	0	0,5002	67	0	0,5002
8	0	0,5002	28	0	0,4997	48	0	0,5002	68	0	0,5002
9	0	0,5002	29	0	0,4997	49	0	0,5002	69	0	0,5002
10	1	0,5002	30	0	0,5002	50	0	0,5002	70	0	0,5002
11	0	0,4997	31	1	0,5002	51	2	0,5002	71	0	0,5002
12	0	0,4997	32	1	0,4997	52	0	0,5003	72	0	0,5002
13	0	0,5002	33	0	0,4998	53	0	0,5004	73	2	0,5002
14	1	0,5002	34	1	0,4997	54	0	0,5002	74	1	0,5003
15	1	0,4997	35	0	0,4997	55	1	0,5002	75	1	1,1171
16	0	0,4998	36	1	0,4997	56	0	0,4997	76	0	0,4998
17	0	0,4997	37	0	0,4997	57	0	0,4997	77	0	0,4997
18	0	0,5002	38	0	0,4997	58	0	0,5002	78	1	0,5002
19	0	0,5002	39	0	0,5002	59	0	0,5002	79	0	0,4997
20	3	0,5002	40	0	0,5002	60	0	0,5002	80	0	0,4997

5.4.5 Çok Katmanlı Algılayıcılar Yöntemi

Yapay sinir ağı literatüründe doğrusal olmayan modelleme ile ilgili birçok yöntem bulunmaktadır. Geriye yayılım algoritmasıyla öğrenmeyi gerçekleştiren çok katmanlı algılayıcılar (ÇKA) bu yöntemlerden en yaygın olarak kullanılan yöntemdir. ÇKA ve geriye yayılım algoritması bölüm 4.7’de anlatılmaktadır.

ÇKA yöntemi için MATLAB 2014a ve R v3.2.2 araçları ile “Neural Network Toolbox” paketi kullanılmıştır. Uygulama sırasında 3 katman kullanılmıştır. Bu katmanlar girdi katmanı, gizli katman ve çıktı katmanıdır. Girdi katmanında bir önceki dönemin sonunda gerçekleşen talep değeri ile talep değeri sıfır olmayan son iki talep arasındaki dönem sayısı yer almaktadır. Çıktı katmanı ise mevcut dönem için tahmin edilen talep değerini ifade etmektedir. Gizli katmanda ise 3 nöron kullanılmıştır. Öğrenme katsayısı 0,1 ve momentum katsayısı 0.9 olarak belirlenmiştir. Model Şekil 46’da yer almaktadır.

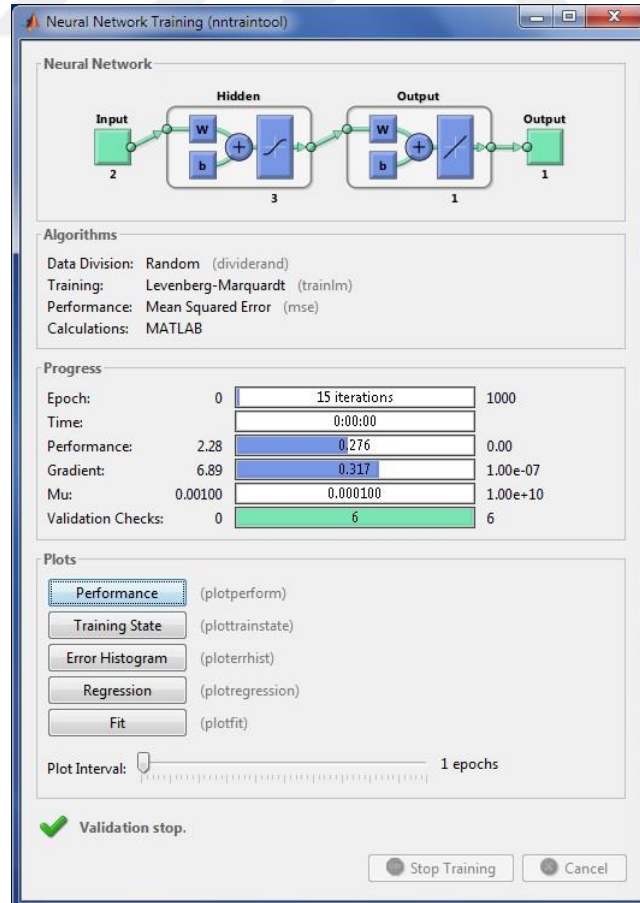
“fitnet” ve “train” fonksiyonları kullanılarak öğrenme gerçekleştirilmiş ve daha sonra “sim” fonksiyonu ile eğitilmiş veriler kullanılarak tahmin yapılmıştır. “Data Division” olarak “Random”, “Training” olarak “Levenberg-marquardt” ve “Performance” olarak “Mean Squared Error” kullanılmıştır. Diğer parametreler için tablo 10’da gösterildiği gibi varsayılan değerler kullanılmıştır.

ÇKA yöntemine göre hesaplanmış bir ürün için 80 günlük test verisinin gerçekleşen talepler ile tahmini talep değerleri Tablo 11’de eğitim verisi ve test verisine ait talep zaman grafiği ise Şekil 47’de yer almaktadır. Grafikte kırmızı çizgiler gerçekleşen talebi, mavi çizgi ise tahmin değerlerini ifade etmektedir

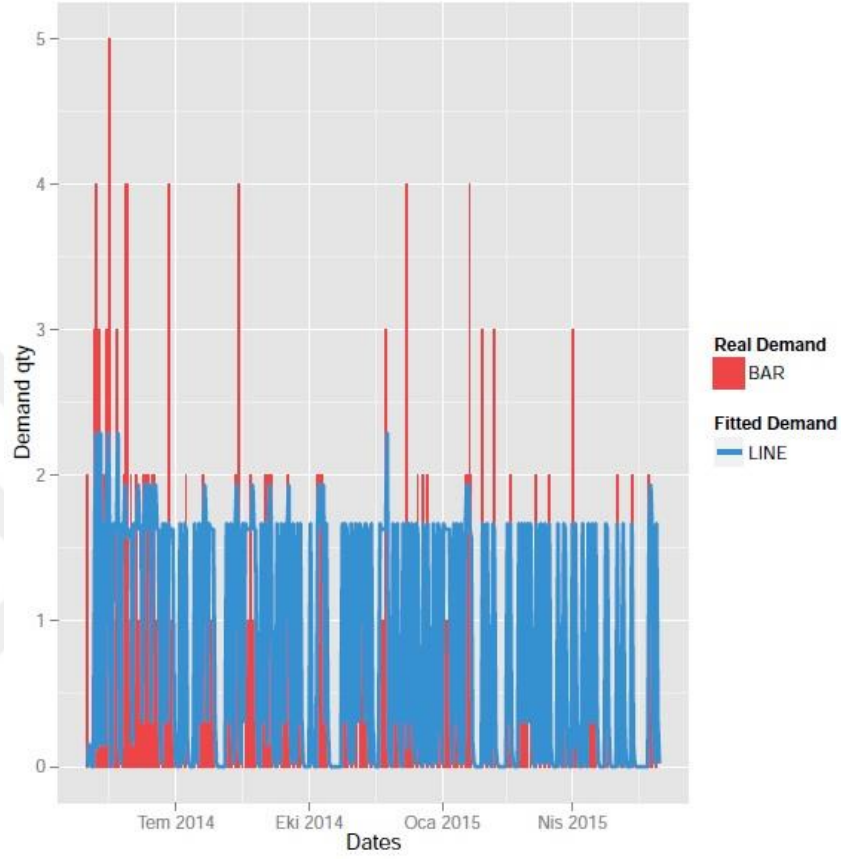
Tablo 10 - ÇKA Yöntemi İçin Kullanılan Parametreler

net.trainParam.epochs	1000	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.max_fail	6	Maximum validation failures
net.trainParam.min_grad	1e-7	Minimum performance gradient
net.trainParam.mu	0.001	Initial mu
net.trainParam.mu_dec	0.1	mu decrease factor
net.trainParam.mu_inc	10	mu increase factor
net.trainParam.mu_max	1e10	Maximum mu
net.trainParam.show	25	Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine	false	Generate command-line output
net.trainParam.showWindow	true	Show training GUI
net.trainParam.time	inf	Maximum time to train in seconds

Şekil 46 - ÇKA Modeli



**Şekil 47 - ÇKA Yöntemine Göre Bir Ürünün Talep Zaman Grafiği
(Gerçekleşen ve Tahmini Talep)**



Tablo 11 - ÇKA Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	ÇKA	Gün	Gerçekleşen	ÇKA	Gün	Gerçekleşen	ÇKA	Gün	Gerçekleşen	ÇKA
1	1	1,6698	21	0	0,1415	41	0	0,0000	61	2	1,6698
2	0	0,3094	22	1	1,6698	42	0	0,0000	62	0	0,1516
3	0	0,0212	23	0	0,3094	43	1	1,6698	63	0	0,0212
4	2	1,6698	24	0	0,0212	44	1	1,6305	64	0	0,0000
5	0	0,1516	25	0	0,0000	45	0	0,3094	65	0	0,0000
6	0	0,0212	26	0	0,0000	46	0	0,0212	66	0	0,0000
7	0	0,0000	27	1	1,6698	47	0	0,0000	67	0	0,0000
8	0	0,0000	28	0	0,3094	48	0	0,0000	68	0	0,0000
9	0	0,0000	29	0	0,0212	49	0	0,0000	69	0	0,0000
10	1	1,6698	30	0	0,0000	50	0	0,0000	70	0	0,0000
11	0	0,3094	31	1	1,6698	51	2	1,6698	71	0	0,0000
12	0	0,0212	32	1	1,6305	52	0	0,1516	72	0	0,0000
13	0	0,0000	33	0	0,3094	53	0	0,0212	73	2	1,6698
14	1	1,6698	34	1	1,6698	54	0	0,0000	74	1	1,9349
15	1	1,6305	35	0	0,3094	55	1	1,6698	75	1	1,6305
16	0	0,3094	36	1	1,6698	56	0	0,3094	76	0	0,3094
17	0	0,0212	37	0	0,3094	57	0	0,0212	77	0	0,0212
18	0	0,0000	38	0	0,0212	58	0	0,0000	78	1	1,6698
19	0	0,0000	39	0	0,0000	59	0	0,0000	79	0	0,3094
20	3	1,6698	40	0	0,0000	60	0	0,0000	80	0	0,0212

5.4.6 SOM Ağları Yöntemi

SOM ağları, çok boyutlu özelliklere sahip nesnelere daha küçük boyutlu uzayda kümeleyen bir yöntemdir. Öz örgütlemeli haritaların amacı; bir küme içerisindeki örneklerin benzerlik derecesini maksimize etmek, farklı kümelerine ait örneklerin benzerliklerini minimize etmek ve sonuçların daha küçük boyutlu bir uzayda temsil edilmesini sağlamaktır.

VQTAM (Vector-Quantized Temporal Associative Memory / Vektör nicemlenmiş geçici çağrışimsal bellek) yöntemi, zaman serilerinin tahmini, robotik ve sistem kontrol uygulamalarında karşılaşılan dinamik girdi-çıkı eşleştirmelerini öğrenmek ve SOM'u etkinleştirmek için önerilmiş bir yöntemdir. t dönemdeki $x(t)$ 'de SOM iki bölümden oluşmaktadır. Birinci bölüm öğrenilecek dinamik eşleştirmenin girdisi hakkında bilgileri bulunduran $x^{in}(t) \in \mathbb{R}^p$, yi ifade etmektedir. İkinci bölüm ise bu eşleştirmenin istenen girdisini içeren $x^{out}(t) \in \mathbb{R}^q$ 'yu ifade etmektedir. Nöron i 'nin ağırlık vektörü $w_i(t)$ ve bu vektörün boyutu buna bağlı olarak artmaktadır. Bu değişiklikler aşağıdaki şekilde formüle edilmektedir (Barreto & Souza, 2006: 788-790)

$$x(t) = \begin{pmatrix} x^{in}(t) \\ x^{out}(t) \end{pmatrix} \text{ ve } w_i(t) = \begin{pmatrix} w^{in}(t) \\ w^{out}(t) \end{pmatrix} \quad (5.6)$$

$w^{in}(t) \in \mathbb{R}^p$ ve $w^{out}(t) \in \mathbb{R}^q$ sırasıyla, istenen eşleştirmenin girdi ve çıktıları hakkında bilgiyi sağlayan ağırlık vektörlerinin parçalarıdır. $x^{in}(t)$ ve $x^{out}(t)$ vektörlerini oluşturmak için seçilen değişkenlere bağlı olarak SOM, ileri ve ters eşleştirmelerin öğrenilmesi için kullanılabilir. Bu vektörlerin mutlaka aynı boyutluluğa sahip olmasına gerek yoktur. Genellikle $p > 1$ ve $q = 1$ olması istenir. Böylece aşağıdaki tanımlamalar uygulanabilir.

$$x^{in}(t) = [y(t)y(t-1) \dots y(t-p+1)]^T \quad (5.7)$$

$$x^{out}(t) = s(t) \quad (5.8)$$

$s(t)$, her bir zaman dönemi t 'deki iletilme sinyalini, $y(t)$ ilgili kanalın çıktısını, p dengeleyicinin sırasını ve T aktarıcı vektörünü ifade etmektedir. Öğrenme sırasında, t zaman dönemindeki kazanan vektör sadece $x^{in}(t)$ esas alınarak belirlenir.

$$x^*(t) = \arg\{\|x^{in}(t) - w^{in}(t)\|\}_{i \in A} \quad (5.9)$$

Ağırlıkların güncellenmesi için $x^{in}(t)$ ve $x^{out}(t)$ kullanılır:

$$w_i^{in}(t-1) = w_i^{in}(t) + \alpha(t)h(i^*, i; t)[x^{in}(t) - w^{in}(t)] \quad (5.10)$$

$$w_i^{out}(t-1) = w_i^{out}(t) + \alpha(t)h(i^*, i; t)[x^{out}(t) - w^{out}(t)] \quad (5.11)$$

$\alpha(t)$, 0 ile 1 arasındaki öğrenme katsayısını ve $h(i^*, i; t)$ Gauss topolojik fonksiyonunu ifade etmektedir. Bu fonksiyon bölüm 4.9.3'de anlatılmaktadır.

VQTAM tarafından uygulanan SOM tabanlı çağrışımsal bellek yöntemi, fonksiyon yaklaşımı amaçları için kullanılabilir. Önce SOM eğitilmekte ve çıktısı $Z(t)$ yeni bir girdi vektörü için öğrenilmiş kod çizelgesi vektörlerinden tahmin edilmektedir.

$$Z(t) = w_{i^*}^{out}(t) \quad (5.12)$$

$w_{i^*}^{out}(t)$, kazanan nöron $i^*(t)$ 'nin ağırlık vektörüdür. Kanalın dengelenmesi için $q = 1$ 'dir. VQTAM esaslı dengeleyicinin çıktısı, 5.12 numaralı denklemin sayısal biçimidir. $\hat{s}(t)$ her bir zaman dönemi t 'deki tahmin edilmiş iletilme sinyalini ifade etmektedir.

$$Z(t) = \hat{s}(t) = w_{1,i^*}^{out}(t) \quad (5.13)$$

VQTAM yöntemi, SOM gibi yarışmacı sinir ağlarına açıkça bir hata sinyali hesaplamadan girdi ve çıktılarının eşleştirilmesinde bağlantı kurmaya izin vermektedir. VQTAM daha önce de bahsedildiği gibi fonksiyon yaklaşımı amaçları için kullanılabilir. Bununla birlikte bir vektör niceme yöntemi olduğu için doğru bir genelleme yapabilmek için yüksek sayıda nörona ihtiyaç duymaktadır.

VQTAM yönteminin performansını geliştirebilmek için eğitilmiş VQTAM mimarisinden tasarlanmış yerel RBF (Radial Basic Function – Dairesel Tabanlı/Gaussian Fonksiyonu) modelini kullanmak gerekmektedir. Birçok sinir ağı mimarisi yerel doğrusal modeller aracılığıyla doğrusal olmayan girdi ve çıktılarını eşleştirmelerini yaklaşık olarak tahmin etmeyi amaçlamaktadır. Bu yaklaşımın arkasında yatan fikir, girdi uzayının küçük bir kısmının her bir yeni girdi vektörü için eşleştirme çıktısının tahmininde kullanılmasıdır.

VQTAM mimarisinde, yerel modellemede temel fonksiyonların merkezlerini ve RBF mimarisinin gizli çıktı ağırlıklarını ayarlamak için ihtiyaç duyulan şey $1 < K < q$ olmasıdır. Bu amaçla, $\{i_1^*, i_2^*, \dots, i_K^*\}$ olarak ifade edilen kazanan ilk K nöronunun ağırlık vektörlerinin kullanılması önerilmektedir.

$$i_1^* = \underset{\forall i}{\operatorname{argmin}} \{ \|x^{in}(t) - w^{in}(t)\| \}$$

$$i_2^* = \underset{\forall i \neq i_1^*}{\operatorname{argmin}} \{ \|x^{in}(t) - w^{in}(t)\| \}$$

$$\begin{aligned} & \cdot \quad \cdot \quad \cdot \\ & \cdot \quad \cdot \quad \cdot \\ & \cdot \quad \cdot \quad \cdot \end{aligned}$$

$$i_K^* = \underset{\forall i \neq \{i_1^*, \dots, i_{K-1}^*\}}{\operatorname{argmin}} \{ \|x^{in}(t) - w^{in}(t)\| \} \quad (5.14)$$

Tahmin edilen çıktı ise aşağıda yer almaktadır.

$$Z(t) = \frac{\sum_{k=1}^K w_{1,i_k^*}^{out}(t) G_{i_k^*}(x^{in}(t))}{\sum_{k=1}^K G_{i_k^*}(x^{in}(t))} \quad (5.15)$$

Yerel RBF mimarisi KRBF mimarisi olarak inşa edilmektedir. $K = 1$ ve $K = q$ olması durumunda VQTAM ve global RBF mimarilerinin KRBF'nin örnek parçası haline geldiği görülmektedir. KRBF modelinde, mevcut girdi vektörüne en yakın K örnek vektöründen tek bir yerel RBF modeli kurulmaktadır. SOM esaslı filtrelerin

rekabetçi doğası, yapılara “modüler ağlar” adı verilen bir mimari seviyesi sunmaktadır. Yerel uzman ağlar tarafından oluşturulan modüler ağlar bir ana ağ tarafından koordine edilmekte ve aynı girdiyle çalışmaktadır. Bir yerel uzman filtresi rekabetçi ağının çıktısı aşağıda yer almaktadır.

$$z(t) = \sum_{i=1}^q g_i(t) z_i(t) \quad (5.16)$$

g_i , i . uzman filtrenin çıktısının $z_i(t)$, ağın tüm çıktılarına katkısını ölçen ana çıktıyı ifade etmektedir.

KRBF mimarisinde her bir uzman filtrenin ağın son çıktısına katkısının derecesi, mevcut girdi vektörüne merkezi en yakın K en aktif tabanlı fonksiyonun seçimiyle açık bir mekanizma tarafından kısıtlanmaktadır. Böylece, ana ağın değerleri aşağıdaki şekildedir.

$$g_i(t) = \begin{cases} \frac{G_i(x^{in}(t))}{\sum_{i=1}^q G_i(x^{in}(t))} & , \text{eğer } i \in \{i_1^*(t), \dots, i_K^*(t)\} \\ 0, & \text{yoksa} \end{cases} \quad (5.17)$$

q gizli birimlerin sayısını ifade etmektedir. Ağın çıktısı ise aşağıdaki şekildedir.

$$Z(t) = \hat{s}(t) = \sum_{i=1}^q g_i(t) w_{1,i}^{out}, \quad i = 1, \dots, K \quad (5.18)$$

SOM Ağları yöntemi için MATLAB 2014a ve R v3.2.2 araçları ile “SOM Toolbox v2.1” paketi kullanılmıştır. Uygulamada öncelikle SOM Toolbox kütüphanesinden “som_randinit” ve “som_seqtrain” fonksiyonları kullanılarak eğitim verisi üzerinden model oluşturulmuştur. Diğer parametreler varsayılan olarak kullanılmıştır.

“som_randinit” fonksiyonu için tablo 12’de yer alan parametreler kullanılmıştır. Bu fonksiyon ile topolojik komşuluk dikdörtgen olarak belirlenerek iki boyutlu bir haritalama yapılmıştır ve girdi nöron sayısı 16 olarak seçilmiştir. Nöron sayısı 4, 8 ve 32 seçilerek de deneyler gerçekleştirilmiş fakat en iyi öğrenmenin 16 nöronla sağlandığı görülmüştür. Nöron sayısının az olması kötü öğrenmeye sebep olurken, nöron sayısının fazla olması da ağın ezberlemesine neden olmaktadır.

Tablo 12 - Som_randinit Fonksiyonu İçin Kullanılan Parametreler

'msize' - map size: 16x1 2-D SOM map
'lattice' - map lattice: 'hexa' or 'rect': rect
'shape' - map shape: 'sheet', 'cyl' or 'toroid': sheet

“som_seqtrain” fonksiyonu içinse tablo 13’de yer alan parametreler kullanılmıştır. Bu fonksiyon ile öğrenme yapılmıştır. Eğitim sırasında zamansal olarak son 1 veri kullanılmıştır. Örneğin 1. günün verisi kullanılarak 2. günün tahmini yapılarak öğrenme sağlanmıştır. Öğrenme yapılırken Gauss topoloji fonksiyonu kullanılmıştır ve 50 döngü yapılmıştır.

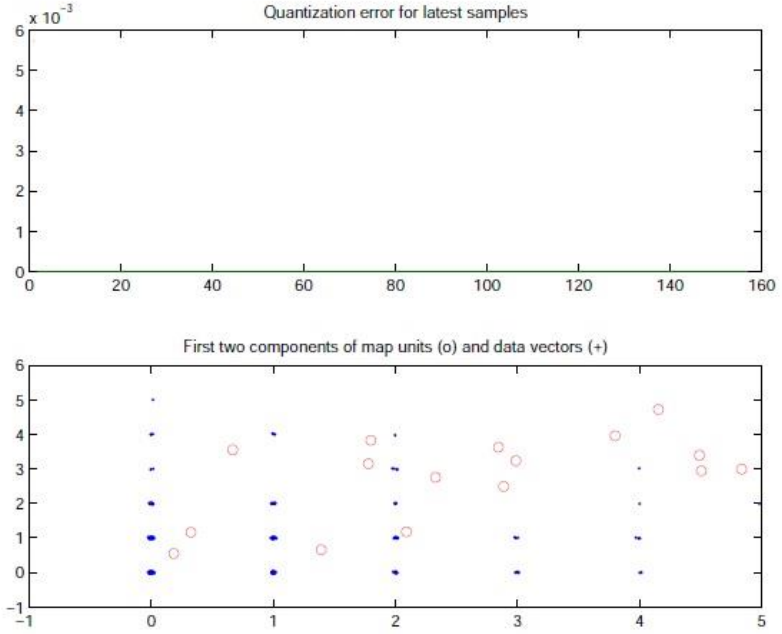
Tablo 13 - Som_seqtrain Fonksiyonu İçin Kullanılan Parametreler

'radius' - neighborhood radiuses, length 1, 2 or trainlen: [8 0]
'trainlen' - training length: 50
'sample_order' - order of samples: 'random' or 'ordered': random
'neigh' - neighborhood function, 'gaussian', 'cutgauss', 'ep' or 'bubble': gaussian

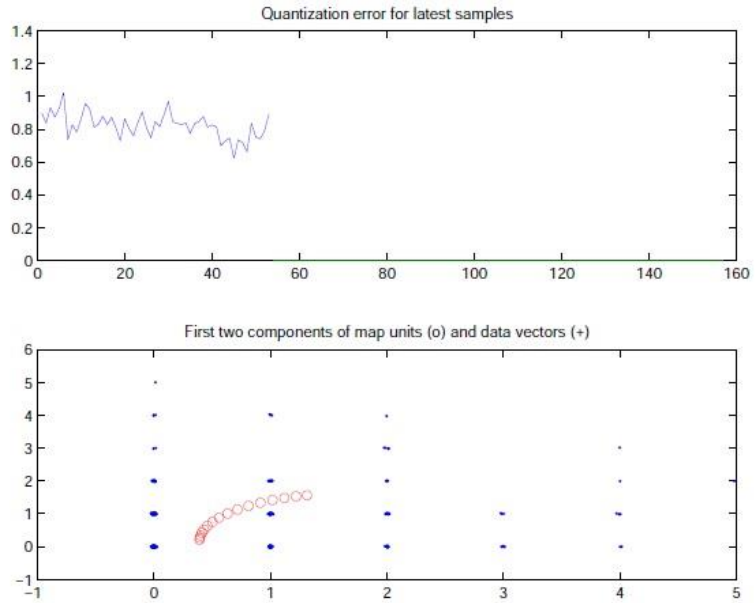
Daha sonra oluşturulan model ve “som_bmus” fonksiyonu kullanılarak VQTAM tabanlı bir dengeleyici olan KRBF tekniğiyle ile test verisi üzerinde tahminler yapılmıştır.

Şekil 48’de bağlantı ağırlıkları rastgele atanmış 16 tane nöron kırmızı yuvarlaklar şeklinde yer almaktadır. Mavi noktalar ise bir sonraki gün gerçekleşecek olan talep miktarlarını ifade etmektedir. Bu bilgiler eğitim verisi üzerinden gelen verilerdir ve girdileri oluşturmaktadır. Girdilerle her nöronun ağırlık vektörleri arasındaki uzaklıklar hesaplanarak girdi vektörüne en yakın olan nöron kazanan nöron ilan edilir ve kazanan nöronla komşu nöronlarının ağırlıkları yeniden güncellenir. Şekil 49’da 17 döngüden sonraki haritalama, şekil 50’de 34 döngüden sonraki haritalama ve şekil 51’de 50 döngüden sonraki haritalama yer almaktadır. Her döngüyle birlikte tahmin hatası da azalmaktadır. Uygulamada 100 döngü de denenmiş fakat tahmin hatasında büyük bir değişiklik olmadığı ve öğrenme çok uzun sürdüğü için tercih edilmemiştir.

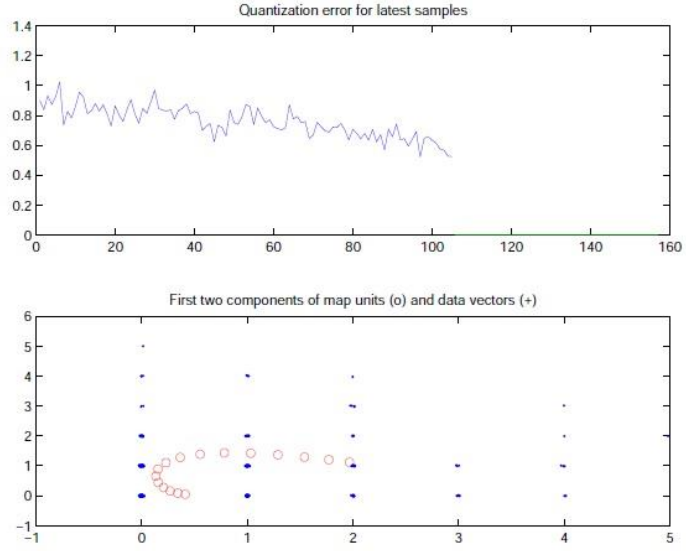
Şekil 48 - Bir Ürün İçin SOM Ağları Haritalama Örneği (Başlangıç Ağırlıkları)



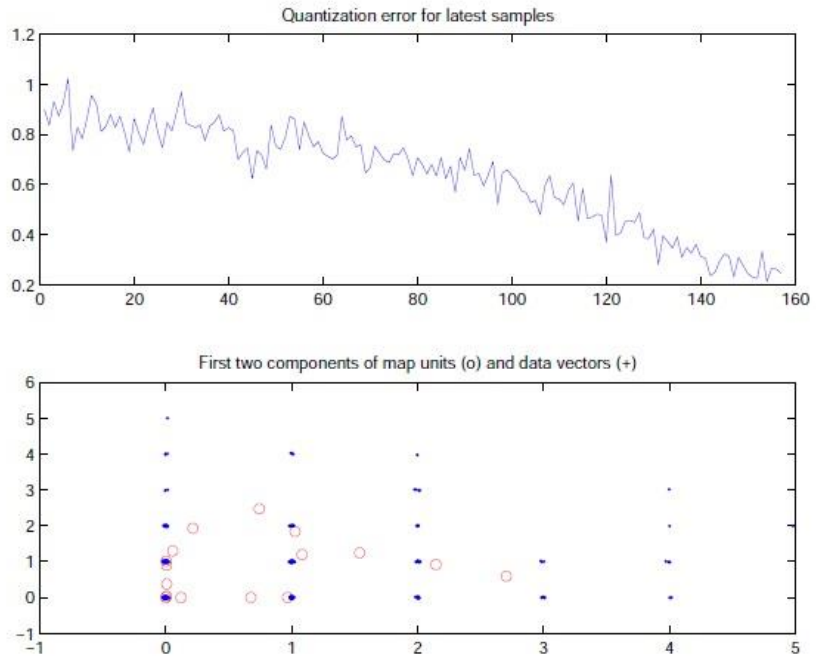
Şekil 49 - 17 Döngüden Sonra Bir Ürün İçin SOM Ağları Haritalama Örneği



**Şekil 50 - 34 Döngüden Sonra Bir Ürün İçin SOM Ağları Haritalama
Örneği**



**Şekil 51 - 50 Döngüden Sonra Bir Ürün İçin SOM Ağları Haritalama
Örneği**



Deneyleer sırasında farklı parametre seenekleri ile de modeller oluřturulmuř ve tahmin yapılmıřtır. Farklı parametrelerle yapılan deney sonularının tahmin hataları (MSE, RMSE, MASE, MAD, sMAPE) karřılařtırılmıř ve Tablo 14’de yer alan parametrelerin kullanılmasına karar verilmiřtir.

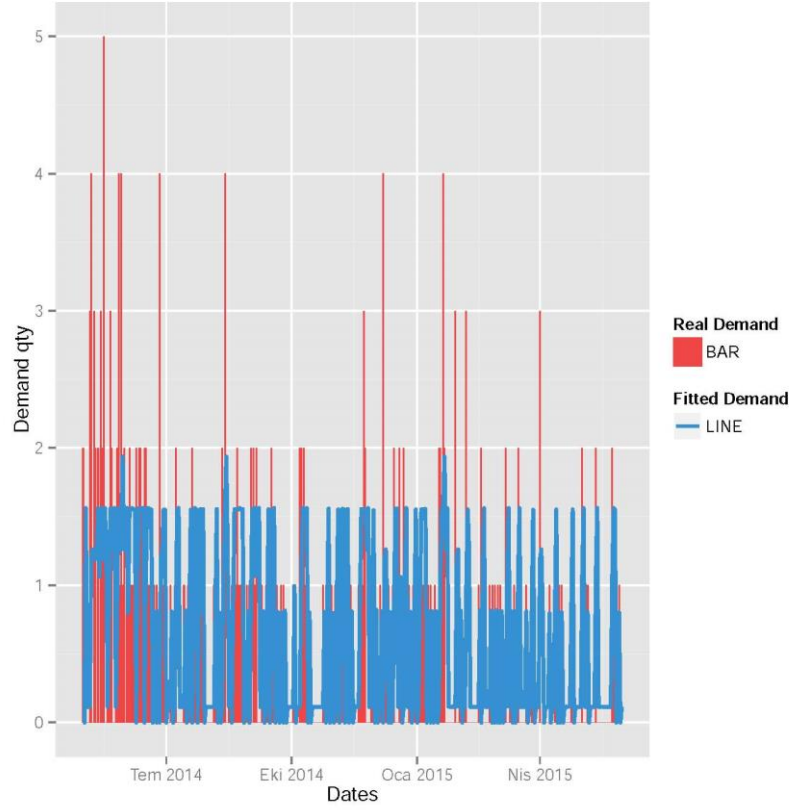
Tablo 14 - SOM Aėlarında Kullanılan Parametreler

Parametreler	Deėerler
map size	16x1
k	4
lattice	rect
shape	sheet
trainlen	50

SOM aėlarında kullanılan parametreleri zetlemek gerekirse, topolojik komřuluk dikdrtgen olarak belirlenerek iki boyutlu bir haritalama yapılmıřtır. Girdi nron sayısı 16 olarak seilmiřtir ve kazanan nronun belirlenmesinde $k=4$ yani en yakındaki 4 nron belirlenmiřtir. Dng sayısı ise 50 olarak belirlenmiřtir. Nron sayısının az olması kt ėrenmeye sebep olurken, nron sayısının fazla olması da aėın ezberlemesine neden olmaktadır. Bu sebeple optimum nron sayısının belirlenmesi nemlidir.

SOM aėları yntemine gre hesaplanmıř bir rn iin 80 gnlk test verisinin gerekleřen talepler ile tahmini talep deėerleri Tablo 15’de eėitim verisi ve test verisine ait talep zaman grafiėi ise Őekil 52’de yer almaktadır. Grafikte kırmızı izgiler gerekleřen talebi, mavi izgi ise tahmin deėerlerini ifade etmektedir.

Şekil 52 - SOM Ağları Yöntemine Göre Bir Ürünün Talep Zaman Grafiği (Gerçekleşen ve Tahmini Talep)



Tablo 15 - SOM Ağları Yöntemine Göre Gerçekleşen ve Tahmin Edilen Talep Değerleri

Gün	Gerçekleşen	SOM	Gün	Gerçekleşen	SOM	Gün	Gerçekleşen	SOM	Gün	Gerçekleşen	SOM
1	1	0,4501	21	0	0,7436	41	0	0,0440	61	2	0,8228
2	0	0,0059	22	1	0,4485	42	0	0,0436	62	0	0,2167
3	0	0,0446	23	0	0,0059	43	1	0,4522	63	0	0,0443
4	2	0,8228	24	0	0,0442	44	1	1,1433	64	0	0,0432
5	0	0,2167	25	0	0,0441	45	0	0,0059	65	0	0,0434
6	0	0,0440	26	0	0,0444	46	0	0,0446	66	0	0,0447
7	0	0,0441	27	1	0,4504	47	0	0,0439	67	0	0,0435
8	0	0,0438	28	0	0,0059	48	0	0,0441	68	0	0,0436
9	0	0,0438	29	0	0,0445	49	0	0,0435	69	0	0,0441
10	1	0,4434	30	0	0,0442	50	0	0,0441	70	0	0,0441
11	0	0,0059	31	1	0,4451	51	2	0,9686	71	0	0,0444
12	0	0,0444	32	1	1,1442	52	0	0,2167	72	0	0,0443
13	0	0,0446	33	0	0,0059	53	0	0,0438	73	2	0,8227
14	1	0,4462	34	1	0,4539	54	0	0,0436	74	1	0,4593
15	1	1,1980	35	0	0,0059	55	1	0,4450	75	1	1,1435
16	0	0,0059	36	1	0,4453	56	0	0,0059	76	0	0,0059
17	0	0,0443	37	0	0,0059	57	0	0,0440	77	0	0,0442
18	0	0,0443	38	0	0,0444	58	0	0,0443	78	1	0,4541
19	0	0,0441	39	0	0,0444	59	0	0,0436	79	0	0,0059
20	3	0,9686	40	0	0,0438	60	0	0,0441	80	0	0,0445

5.5 Aralıklı Talep Yapısına Sahip Ürünlerin Talep Tahmini İçin Kullanılan Yöntemlerin Karşılaştırılması

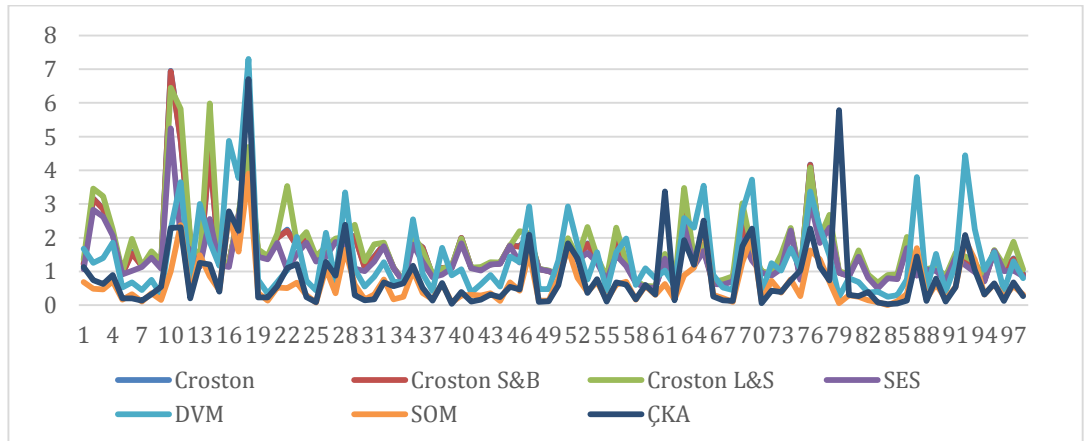
Veri üzerinde sırasıyla SES, Croston yöntemi, Syntetos & Boylan ve daha sonra Leven & Segerstedt tarafından değişikliğe uğratılan Croston yönteminin modifikasyonları, DVM, ÇKA ve SOM ağırları yöntemleri uygulanarak tahmin yapıldıktan sonra tahmin yapılan 80 günlük dönemin gerçekleşen talep değerleriyle tahmin değerleri kullanılarak performanslar ölçülmüştür. Bahsedilen tahmin yöntemlerinin performanslarını ölçmek ve karşılaştırma yapabilmek amacıyla Bölüm 1.5’de yer alan tahmin hatası ölçütlerinden MSE, RMSE, MASE, MAD ve sMAPE ölçütleri hesaplanmıştır.

MSE ölçütüne göre yöntemlerin karşılaştırılması tablo 16’de ve şekil 53’de yer almaktadır.

Tablo 16 - MSE Ölçütüne Göre Yöntemlerin Karşılaştırılması

Yöntem	MSE	Sıralama
Croston	1,3469	4
Croston S&B	1,6284	7
Croston L&S	1,3739	5
SES	1,3024	3
DVM	1,4357	6
SOM	0,6583	1
ÇKA	0,9712	2

Şekil 53 - MSE Ölçütüne Göre Yöntemlerin Karşılaştırılma Grafiği



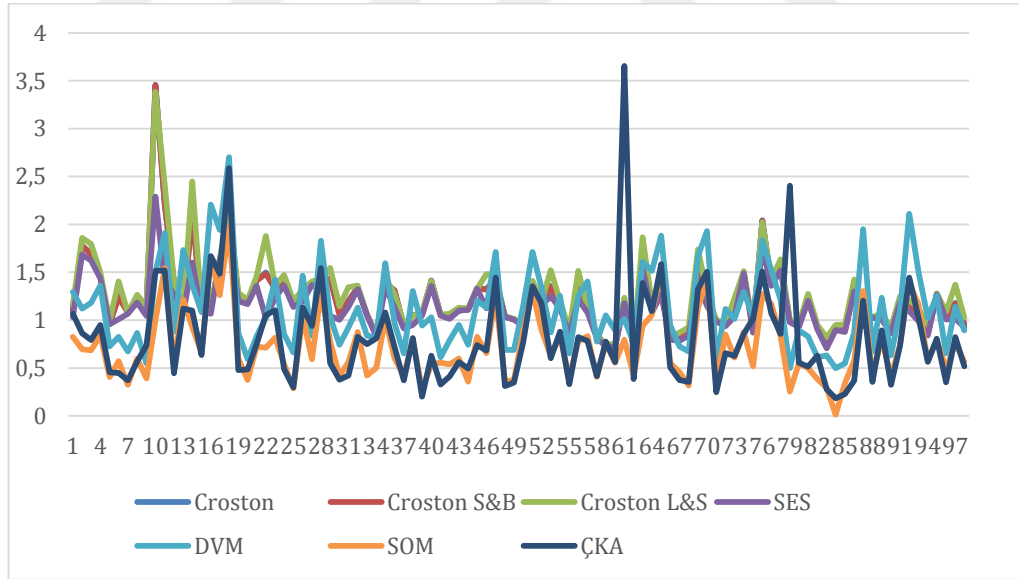
MSE ölçütü göz önünde bulundurulduğunda en başarılı yöntemin SOM ağırları daha sonra ise ÇKA olduğu görülmektedir.

RMSE ölçütüne göre yöntemlerin karşılaştırılması tablo 17’de ve şekil 54’de yer almaktadır.

Tablo 17 - RMSE Ölçütüne Göre Yöntemlerin Karşılaştırılması

Yöntem	RMSE	Sıralama
Croston	1,0714	4
Croston S&B	1,2202	7
Croston L&S	1,0791	5
SES	1,0593	3
DVM	1,1182	6
SOM	0,7299	1
ÇKA	0,8280	2

Şekil 54 - RMSE Ölçütüne Göre Yöntemlerin Karşılaştırılma Grafiği



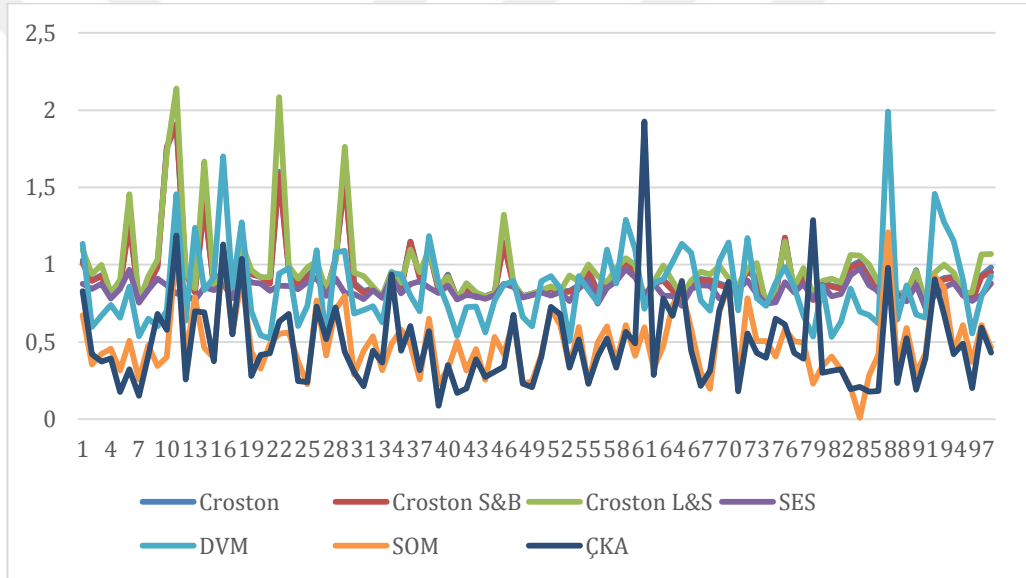
RMSE ölçütü göz önünde bulundurulduğunda da yine en başarılı yöntemin SOM ağırları daha sonra ise ÇKA olduğu görülmektedir.

MASE ölçütüne göre yöntemlerin karşılaştırılması tablo 18’de ve şekil 55’de yer almaktadır.

Tablo 18 - MASE Ölçütüne Göre Yöntemlerin Karşılaştırılması

Yöntem	MASE	Sıralama
Croston	0,8242	4
Croston S&B	0,9268	7
Croston L&S	0,8297	5
SES	0,8018	3
DVM	0,8600	6
SOM	0,4921	2
ÇKA	0,4905	1

Şekil 55 - MASE Ölçütüne Göre Yöntemlerin Karşılaştırılma Grafiği



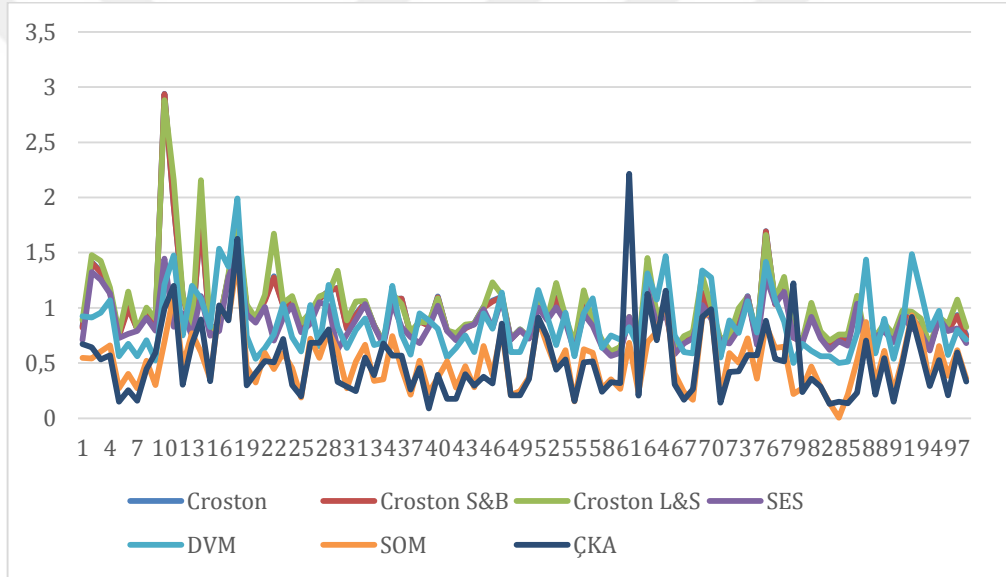
MASE ölçütü göz önünde bulundurulduğunda en başarılı yöntemin ÇKA daha sonra SOM ağırları olduğu görülmektedir ve bu iki yöntemin hatalarının birbirine çok yakın olduğu gözlemlenmektedir.

MAD ölçütüne göre yöntemlerin karşılaştırılması tablo 19'da ve şekil 56'da yer almaktadır.

Tablo 19 - MAD Ölçütüne Göre Yöntemlerin Karşılaştırılması

Yöntem	MAD	Sıralama
Croston	0,8435	4
Croston S&B	0,9529	7
Croston L&S	0,8501	5
SES	0,8173	3
DVM	0,8720	6
SOM	0,5140	1
ÇKA	0,5148	2

Şekil 56 - MAD Ölçütüne Göre Yöntemlerin Karşılaştırılma Grafiği



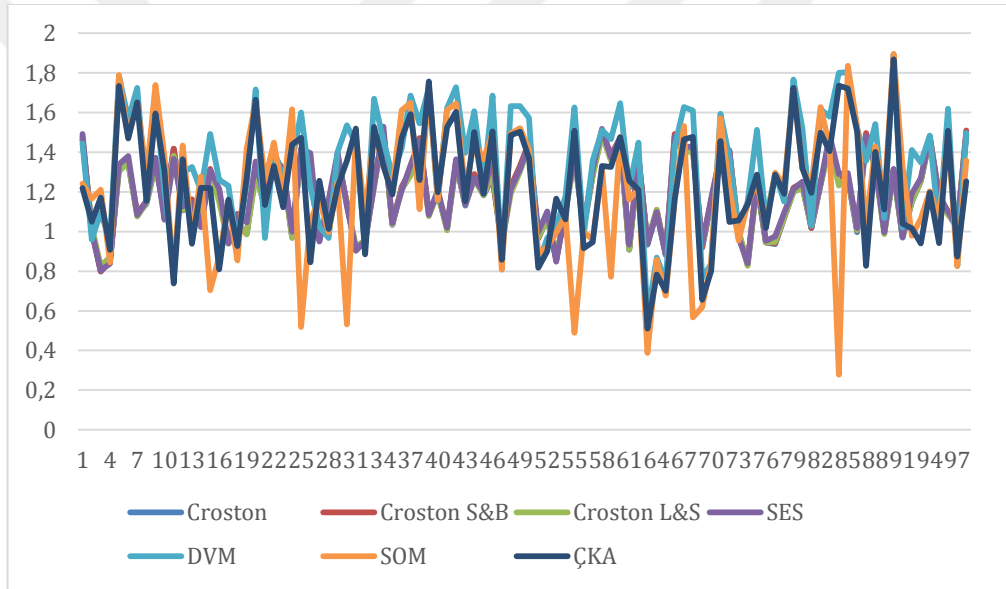
MAD ölçütü göz önünde bulundurulduğunda da SOM ağırları ve ÇKA yöntemlerinin hata değerlerinin neredeyse aynı ve diğer yöntemlerden daha başarılı olduğu gözlemlenmektedir.

sMAPE ölçütüne göre yöntemlerin karşılaştırılması tablo 20'de ve şekil 57'de yer almaktadır.

Tablo 20 - sMAPE Ölçütüne Göre Yöntemlerin Karşılaştırılması

Yöntem	sMAPE	Sıralama
Croston	1,3034	4
Croston S&B	1,1877	1
Croston L&S	1,3037	5
SES	1,3183	6
DVM	1,3342	7
SOM	1,2059	2
ÇKA	1,2370	3

Şekil 57 - sMAPE Ölçütüne Göre Yöntemlerin Karşılaştırılma Grafiği



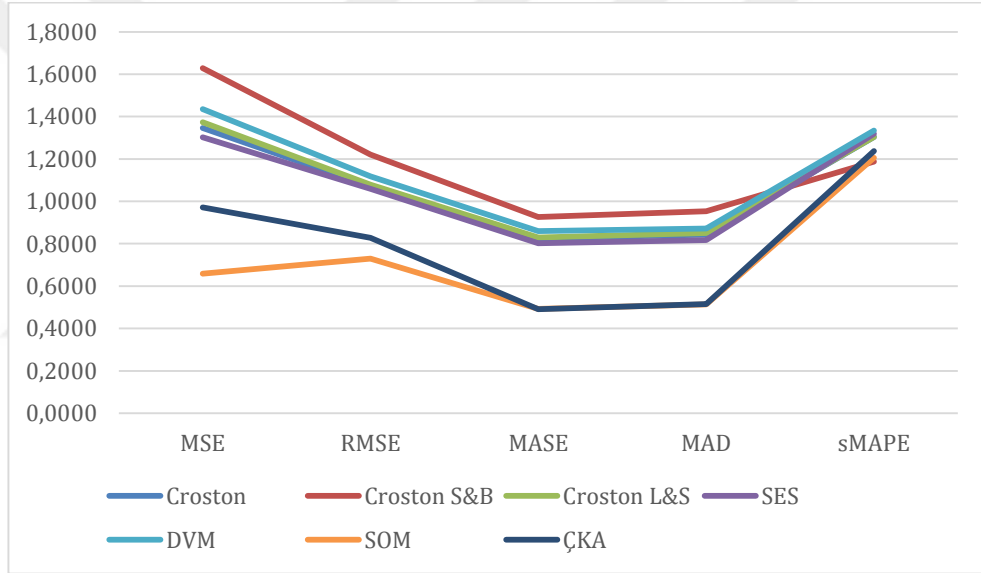
sMAPE ölçütü göz önünde bulundurulduğunda Syntetos & Boylan tarafından değişikliğe uğratılan Croston yönteminin modifikasyonun daha başarılı olduğu, bu yöntemi sırasıyla SOM ve ÇKA yöntemlerinin izlediği görülmektedir.

Hesaplanan tüm ölçütlere göre yöntemlerin karşılaştırılması tablo 21’de ve şekil 58’de yer almaktadır.

Tablo 21 - Tüm Ölçütlere Göre Yöntemlerin Karşılaştırılması

Yöntem	MSE	RMSE	MASE	MAD	sMAPE
Croston	1,3469	1,0714	0,8242	0,8435	1,3034
Croston S&B	1,6284	1,2202	0,9268	0,9529	1,1877
Croston L&S	1,3739	1,0791	0,8297	0,8501	1,3037
SES	1,3024	1,0593	0,8018	0,8173	1,3183
DVM	1,4356	1,1181	0,8600	0,8720	1,3342
SOM	0,6583	0,7299	0,4921	0,5140	1,2059
ÇKA	0,9712	0,8280	0,4905	0,5148	1,2370

Şekil 58 - Tüm Ölçütlere Göre Yöntemlerin Karşılaştırma Grafiği



Bütün ölçütler göz önünde bulundurularak yapılan karşılaştırma sonucunda tahmin hatasının en düşük olduğu yöntemin SOM ağırları ve bu yönteme çok yakın değerlere sahip olan ÇKA yönteminin olduğu görülmektedir. Ayrıca kullanılan diğer yöntemlerin hata değerlerinin birbirine yakın ve daha kötü sonuçlar verdiği gözlenmektedir.

SONUÇ VE ÖNERİLER

Talep tahmini ve planlama bir işletmenin başarısını doğrudan etkilediği için büyük önem taşımaktadır. Düzgün talep yapısına sahip ürünlerin talep tahmininde klasik yöntemler başarılı sonuçlar verirken çoğu zaman döneminde rastgele gerçekleşen ve talebi sıfır olan aralıklı talep yapısına sahip ürünlerin talep tahmininde bu yöntemler yetersiz kalmaktadır ve iyi sonuçlar vermemektedir.

Aralıklı talep yapısına sahip ürünlerin talep büyüklüğü ve şeklinin değişikliği tahmin ve planlama yapılmasını zorlaştırmaktadır. Bu zorluğun üstesinden gelebilmek için Croston (1972) özellikle aralıklı talebe sahip ürünler için bir tahmin yöntemi geliştirmiş daha sonra bu yöntem Syntetos & Boylan ve Leven & Segerstedt tarafından değişikliğe uğratarak farklı modifikasyonları ortaya çıkmıştır. Yapılan çalışmalarda bu yöntemlerin basit üstel düzeltme yöntemiyle karşılaştırıldığı görülmektedir. Fakat bu yöntemlerin de çok iyi sonuçlar vermediği görülmüş ve son zamanlarda daha gelişmiş makine öğrenme yöntemleriyle çalışmalar yapılmıştır.

Bu çalışmada e-ticaret sektöründe faaliyet gösteren bir işletmenin aralıklı talep yapısına sahip ürünlerinin talep tahmininde; basit üstel düzeltme, Croston yöntemi ve modifikasyonları, makine öğrenme yöntemlerinden DVM ve bir yapay sinir ağı modeli olan ÇKA ve SOM ağları kullanılarak talep tahmini yapılmıştır. Daha sonra bu yöntemlerin hataları karşılaştırılarak performansları incelenmiştir.

DVM, istatistiksel öğrenme teorisi üzerine kurulmuş makine öğrenmesinin en yaygın olarak kullanılan algoritmalarından biridir. Özellikle sınıflandırma ve regresyon problemlerinin çözümünde kullanılmaktadır. Aralıklı talebe sahip ürünlerin talep tahmininde yapılan çalışmalar incelendiğinde bu yöntemde sıfır değer almayan dönemlerin tahmininde olasılık yerine hiper düzlemden olan uzaklık baz alındığı için aralıklı talep tahmininde çok başarılı sonuçlar alınmadığı gözlemlenmektedir.

ÇKA, tek katmanlı algılayıcıların doğrusal olmayan problemlere çözüm bulamaması ve bu problemin üstesinden gelebilmek için geliştirilmiş bir yapay sinir ağı modelidir. Bu modelde girdi katmanıyla çıktı katmanı arasına gizli katman adı verilen ek katmanlar yerleştirilerek çok katmanlı ileri beslemeli ağlar geliştirilmiştir. Aralıklı

talep tahmini çalışmaları incelendiğinde bu yöntemin sıklıkla kullanıldığı görülmektedir.

Yine bir yapay sinir ağı modeli ona insan beyninin serebral korteksinin fonksiyonlarından esinlenerek tasarlanan SOM ağları ise çok boyutlu özelliklere sahip nesnelere daha küçük boyutlu uzayda kümeleyen ve haritalayan bir yöntemdir. SOM ağları daha çok özellik çıkarımı, örüntü tanıma, konuşma analizi gibi alanlarda uygulanmış talep tahmininde yaygın olarak kullanılmamıştır. Aralıklı talep tahmininde ise herhangi bir uygulama mevcut değildir. Talep tahmininde SOM ağlarını etkinleştirmek için VQTAM yönteminin kullanıldığı görülmektedir. VQTAM yöntemi, dinamik girdi-çıkı eşleştirmelerini öğrenmek için önerilmiş bir yöntemdir.

Tezin uygulama bölümünde öncelikle e-ticaret sektöründe faaliyet gösteren bir işletmeden alınan talep verileri üzerinde Syntetos ve diğerleri (2005) tarafından geliştirilen sınıflandırma tekniği uygulanmıştır. İki ardışık talep arasında geçen zaman periyodlarının ortalaması ve talep büyüklüğünün değişkenlik/varyasyon katsayısının karesi dikkate alınarak aralıklı talebe sahip olan 98 adet ürün bulunmuştur.

Bu ürünler için sırasıyla SES, Croston, Croston S&B, Croston L&S, DVM, ÇKA ve SOM yöntemleri kullanılarak tahmin yapılmıştır. Tahmin yapılırken talep verilerinin %80'i eğitim için, kalan %20'si de test için ayrılmıştır. Başka bir ifadeyle toplam 396 günlük verinin eğitim için ayrılan 316 günlük bölümünde eğitim yapılmış ve 80 günlük test verisi üzerinde öğrenme gerçekleştirilmiştir. Uygulamada tüm yöntemler için periyodik ilerleme (rolling) mekanizması kullanılmıştır. Başka bir ifadeyle düzensiz bir yapıya sahip verinin olması sebebiyle talep tahmin doğruluğunu arttırmak için her yeni dönemi tahmin ederken en son döneme ait veriyi de kullanarak tahmin gerçekleştirilmiştir.

Daha sonra elde edilen sonuçlar MSE, RMSE, MASE, MAD ve sMAPE hata ölçütleri göz önünde bulundurularak karşılaştırılmıştır. Bu karşılaştırma sonucunda en düşük hatayı SOM ağlarının ve bu yöntemle çok yakın değerlere sahip olan ÇKA yönteminin verdiği görülmektedir. Diğer yöntemlerinin hata oranlarının ise birbirlerine yakın olduğu ve performanslarının daha düşük olduğu gözlemlenmektedir.

Bu tez çalışmasında talep tahmini yapılırken talebi etkileyen diğer faktörler (fiyat, rekabet, kampanya vb.) dikkate alınmamıştır. Daha sonraki çalışmalarda bu faktörleri de göz önünde bulundurarak ve parametre optimizasyonu yapılarak tahmin hatası iyileştirilebilir. Bunun yanında hibrid yaklaşımlarla ya da son zamanlarda yaygın olarak kullanılmaya başlanan aşırı öğrenme makineleri ile de daha iyi tahmin sonuçları sağlanabilir.



KAYNAKÇA

- Abe, S.: **Support Vector Machines For Pattern Classification**, London, Springer Science + Business Media, 2005
- Akay, D. ve Atak M.: “Grey Prediction With Rolling Mechanism For Electricity Demand Forecasting Of Turkey”, **Energy**, C: 32, 2007, pp. 1670-1675.
- Alpaydın, E.: **Yapay Öğrenme**, 2. Baskı, İstanbul, Boğaziçi Üniversitesi Yayınevi, 2012
- Altay, N., Rudisill, F. ve Litteral, L. A.: “Adapting Wright's Modification Of Holt's Method To Forecasting Intermittent Demand”, **Int. J. Production Economics**, C: 111, 2008, pp. 389-408.
- Auria, L. ve Rouslan, A. M.: **Support Vector Machines (SVM) As A Technique For Solvency Analysis**, Berlin, DIW Berlin, 2008.
- Ayhan, S. ve Erdoğan, Ş.: “Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi”, **Eskişehir Osmangazi Üniversitesi İİBF Dergisi**, C: 9, No: 1, 2014, s.175-198.
- Babai, M. Z., Ali, M. M., ve Nikolopoulos, K.: “Impact Of Temporal Aggregation On Stock Control Performance Of Intermittent Demand Estimators: Empirical Analysis”, **Omega**, C: 40, No: 6, 2012, pp. 713-721.
- Babiloni, E., Cardoso, M., Albarracın, J. M. ve Palmer, M. E.: “Demand Categorisation, Forecasting And Inventory Control For Intermittent Demand”, **South African Journal of Industrial Engineering**, C: 21, No: 2, 2010, pp.115-130.
- Bao, Y., Wang, W. ve Zhang, J.: “Forecasting Intermittent Demand by SVMs Regression”, **IEEE International Conference on Systems Man and Cybernetics**, 2004, pp. 461-465.
- Barreto G.A. ve Souza L.G.: “Adaptive Filtering With The Self Organizing Map: A Performance Comparison”, **Neural Networks**, C: 19, 2006, pp. 785-798.
- Basak, D., Pal, S. ve Patranabis, D. C.: “Support Vector Regression”, **Neural Information Processing - Letters and Reviews**, C: 11, No: 10, 2007, pp. 203-224.

- Basheer, I. ve Hajmeer, M.: “Artificial Neural Networks: Fundamentals, Computing, Design, And Application”, **Journal of Microbiological Methods**, C: 43, 2000, pp. 3-31.
- Bell, J.: **Machine Learning: Hands-On For Developers And Technical Professionals**, Kindle Edition, NY, USA, John Wiley & Sons Inc, 2014.
- Bishop, C. M.: **Neural Networks For Pattern Recognition**, Oxford, UK, Clarendon Press, 1995.
- Bousquet, O., Boucheron, S. ve Lugosi, G.: “Introduction To Statistical Learning Theory”, **Advanced Lectures on Machine Learning**, C: 3176, 2004, pp. 169-207.
- Bowersox, D. J. ve Closs, D. J.: **Logistical Management: The Integrated Supply Chain Proces**, New York, USA, McGraw Hill Companies, 1996.
- Bowersox, D. J., Closs, D. J. ve Cooper, M. B.: **Supply Chain Logistics Management**, New York, USA, McGraw-Hill Education, 2002.
- Boylan, J. ve Syntetos, A.: “The Accuracy Of A Modified Croston Procedure”, **International Journal of Production Economics**, C: 107, No: 2, 2007, pp. 511-517.
- Brunato, M. ve Battiti, R.: “Statistical Learning Theory For Location Fingerprinting In Wireless LANs”, **Computer Networks**, C: 47, 2005, pp. 825-845.
- Byun, H. ve Lee, S. W.: “Applications Of Support Vector Machines For Pattern Recognition: A Survey”, **First International Workshop SVM**, 2002, pp. 213-237.
- Camastra, F. ve Vinciarelli, A.: **Machine Learning For Audio, Image And Video Analysis**, London, Springer Science+Business Media, 2010.
- Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. ve Li, H.: “Learning To Rank: From Pairwise Approach To Listwise Approach”, **24th International Conference on Machine Learning**, 2007, pp. 139-146.
- Chamanbaz, M., Dabbene, F., Tempo, R., Venkatamaran, V. ve Wang, Q.G.: “A Statistical Learning Theory Approach For Uncertain Linear And Bilinear Matrix Inequalities”, **Automatica**, C: 50, 2014, pp. 1617-1625.

- Chapelle, O., Schölkopf, B. ve Zien, A.: **Semi-Supervised Learning**, London, UK, The MIT Press, 2006.
- Chen, M. Y.: “Visualition And Dynamic Evaluation Model Of Corporate Financial Structure With Self-organizing Map And Support Vector Regression”, **Applied Soft Computing**, 12, 2012, pp. 2274-2288.
- Croston, J. D.: “Forecasting And Stock Control For Intermittent Demand”, **Operational Research Quarterly**, C: 23, No: 3, 1972, pp. 289-303.
- Cruz, R. M., Peixoto, H. M. ve Magalhaes, R. M.: “Artificial Neural Networks And Efficient Optimization Techniques For Applications In Engineering”, **Artificial Neural Networks-Methodological Advances and Biomedical Applications**, 2011, pp. 45-68.
- Eaves, A. H. ve Kingsman, B. G.: “Forecasting For The Ordering And Stock-holding Of Spare Parts”, **Journal of the Operational Research Society**, C: 55, No: 4, 2004, pp. 431-437.
- Efron, B.: “Bootsrap Method: Another Look At The Jackknife”, **Annals of Statistics**, C: 7, 1979, pp.1-26.
- Elmas, P.: **Yapay Zeka Uygulamaları**, Ankara, Türkiye, Seçkin Yayıncılık, 2010.
- Evgeniou, T., Poggio, T., Pontil, M. ve Verri, A.: “Regularization And Statistical Learning Theory For Data Analysis”, **Computational Statistics & Data Analysis**, C: 38, 2002, pp. 421-432.
- Evgeniou, T., Pontil, M. ve Poggio, T.: “Statistical Learning Theory: A Primer”, **International Journal of Computer Vision**, C: 38, No: 1, 2000, pp. 9-13.
- Fildes, R.: “The Evaluation Of Extrapolative Forecasting Methods. International”, **Journal of Forecasting**, C: 8, 1992, pp. 81-98.
- Fumera, G. ve Roli, F.: “Support Vector Machines With Embedded Reject Option”, **First International Workshop - SVM**, 2002, pp. 68-82.
- Gardner, M. ve Dorling, S.: “Artificial Neural Networks (The Multilayer Perceptron) - A Review Of Application”, **In The Atmospheric Sciences. Atmospheric Environment**, C: 32, No: 14/15, 1998, pp.2627-2636.

- Ghiani, G., Laporte, G. ve Musmanno, R.: **Introduction To Logistics Systems Planning And Control**, Chichester, England, John Wiley & Sons Ltd, 2004.
- Ghobbar, A. A. ve Friend, C.: “Evaluation Of Forecasting Methods For Intermittent Parts Demand In The Field Of Aviation: A Predictive Model”, **Computers & Operations Research**, C: 30, 2003, pp. 2097-2114.
- Ghobbar, A. A. ve Friend, C. H.: “Sources Of Intermittent Demand For Aircraft Spare Parts Within Airline Operations”, **Journal of Air Transport Management**, C: 8, 2002, pp.221-231.
- Gutierrez, R. S., Solis, A. O. ve Mukhopadhyay, S.: “Lumpy Demand Forecasting Using Neural Networks”, **Int. J. Production Economics**, C:111, 2008, pp. 409-420.
- Harrington, P.: **Machine Learning In Action**, NY, USA, Manning Publications Co, 2012.
- Hassoun, M. H.: **Fundamentals Of Artificial Neural Networks**, Newyork, USA, Massachusetts Institute of Technology, 1995.
- Hastie, T., Tibshironi, R. ve Friedman, J.: **The Elements Of Statistical Learning**, S. Edition, NY, USA, Springer Science + Business Media, 2009.
- Haykin, S.: **Neural Networks: A Comprehensive Foundation**, Second Edition, Singapur, Prentice Hall, 1999.
- Heaton, J.: **Introduction To Neural Networks For C**, Chesterfield, USA, Heaton Research Inc, 2008.
- Heizer, J. ve Render, B.: **Operations Management Sustainability And Supply Chain Management**, Eleventh Edition, London, England, Pearson Education Limited, 2014.
- Hsu, S. H., Hsieh, J. P. A., Chih, T. C. ve Hsu, K.-C.: “A Two-stage Architecture For Stock Price Forecasting by Integrating Self-organizing Map And Support Vector Regression”, **Expert Systems with Applications**, C: 36, 2009, pp.7947-7951.
- Hua, Z. S., Zhang, B., Yang, J. ve Tan, D. S.: “A New Approach Of Forecasting Intermittent Demand For Spare Parts Inventories In The Process Industries”, **Journal of the Operational Research Society**, C: 58, No: 1, 2007, pp. 52-61.

- Hua, Z. ve Zhang, B.: “A Hybrid Support Vector Machines And Logistics Regression Approach For Forecasting Intermittent Demand Of Spare Parts”, **Applied Mathematics and Computation**, C: 181, 2006, pp.1035-1048.
- Hyndman, R. J. ve Koehler, A. B.: “Another Look At Measures Of Forecast Accuracy”, **International Journal of Forecasting**, C: 22, 2006, pp.679-688.
- Ismail, S., Shabri, A. Ve Samsudin, R.: “A Hybrid Model Of Self-organizing Maps (SOM) And Least Square Support Vector Machine (LSSVM) For Time-Series Forecasting”, **Expert Systems with Applications**, C: 38, 2011, pp. 10574-10578.
- Johnston, F. R. Ve Boylan, J. E.: “Forecasting For Items With Intermittent Demand”, **Journal of the Operational Research Society**, 47, 1996, pp. 113-121.
- Johnston, F. R., Boylan, J. E. ve Shale, E. A.: “An Examination Of The Size Of Orders From Customers, Their Characterisation And The Implications For Inventory Control Of Slow Moving Items”, **The Journal Of The Operational Research Society**, C: 54, No: 8, 2003, pp. 833-837.
- Kasabov, N. K.: **Foundations Of Neural Networks, Fuzzy Systems, And Knowledge Engineering**, London, England, The MIT Press, 1996.
- Kohonen, T. : **Self-Organization And Associative Memory**, Third Edition, Berlin: Springer-Verlag, 1989.
- Kostenko, A. V. Ve Hyndman, R. j.: “Viewpoint - A Note On The Demand Categorization Of Demand Pattern”, **Journal of the Operational Research Society**, C: 57, 2006, pp. 1256-1258.
- Kourentzes, N.: “Intermittent Demand Forecasts With Neural Networks”, **International J. Production Economics**, C: 143, 2013, pp. 198-206.
- Kourentzes, N. “On Intermittent Demand Model Optimisation And Selection”, **Int. J. Production Economics**, C: 156, 2014, pp. 180-190.
- Krenker, A., Bester, J. ve Kos, A.: “Introduction To Artificial Neural Networks”, **Artificial Neural Networks-Methodological Advances and Biomedical Applications**, 2011, pp. 3-19.

- Kriesel, D.: “A Brief Introduction Neural Networks”, (Çevrimiçi), http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf, 27 Mayıs 2015.
- Kulkarni, P.: **Reinforcement And Systemic Machine Learning For Decision Making**, New Jersey, USA, John Willey & Sons Inc, 2012.
- Kuriscak, E., Marsalek, P., Stroffek, J. ve Toth, P. G.: “Biological Context Of Hebb Learning In Artificial Neural Networks”, **Neurocomputing**, C: 152, 2015, pp. 27-35.
- Leven, E. ve Segerstedt, A.: “Inventory Control With A Modified Croston Procedure And Erlang Distribution”, **International Journal of Production Economics**, C: 90, 2004, pp. 361-367.
- Levis, A. A. ve Papageorgiou, L. G.: “Customer Demand Forecasting via Support Vector Regression Analysis”, **Chemical Engineering Research and Design**, C: 83 No: A8, 2005, pp. 1009-1018.
- Liu, T.Y.: **Learning To Rank For Information Retrieval**, Berlin, Germany, Springer Verlag, 2011.
- Luxburg, U. Ve Schölkopf, B.: “Statistical Learning Theory: Models, Concepts, and Results”, **Handbook of The History of Logic**, C: 10, 2011, pp. 651-707.
- Marsland, S.: **Machine Learning An Algorithmic Perspective**, USA, A Chapman & Hall Book CRC Press, 2009.
- Mitchell, T. M.: **Machine Learning**, NY, USA, MacGraw Hill Education, 1997.
- Mohri, M., Rostamizadeh, A. ve Talwalkar, A.: **Foundations Of Machine Learning**, NY, USA, The MIT Press, 2012.
- Moura, M. d., Zio, E., Lins, I. D. ve Droguett, E.: “Failure and Reliability Prediction by Support Vector Mchines Regression Of Time Series”, **Reliability Engineering and System Safety**, C: 96, 2011, pp. 1527-1534.
- Nahmias, S.: **Production & Operations Analysis**, Sixth Edition, New York, USA, McGraw Hill Education, 2013.

- Navarrete, P. ve Solar, J. R.: “On The Generalization Of Kernel Machines”, **First International Workshop - SVM**, 2002, pp. 24-40.
- Nikolopoulos, K., Syntetos, A., Boylan, J., Petropoulos, F. ve Assimakopoulos, V.: “An Aggregate-disaggregate Intermittent Demand Approach (ADIDA) To Forecasting: An Empirical Proposition And Analysis”, **Journal of the Operational Research Society**, C: 62, No: 3, 2010, pp. 544-554.
- Nilsson, N. J.: **Introduction To Machine Learning (An Early Draft Of A Proposed Textbook)**, CA, USA, 1998.
- Nonmember, H.: “A Short Introduction To Learning To Rank”, **IEICE TRANS. INF. & SYST**, C: E94–D, No: 10, 2011, pp. 1-9.
- Orhunbilge, N.: **Uygulamalı Regresyon ve Korelasyon Analizi**, Gözden Geçirilmiş 2. Baskı, İstanbul: İ.Ü. Basım ve Yayınevi, 2002.
- Öztemel, E.: **Yapay Sinir Ağları**, 3. Basım, İstanbul, Türkiye, Papatya Yayıncılık, 2012.
- Platt, J. C.: “Sequential Minimal Optimization: A Fast Algorithm For Training Support Vector Machines”, **Advances in Kernel Methods: Support Vector Learning**, 1999, pp.185-208.
- Rao, A. V.: “Forecasting And Stock Control For Intemittent Demands”, **Operational Research Quarterly**, C: 24, No: 4, 1973, pp. 639-640.
- Regattieri, A., Gamberi, M., Gamberini, R. ve Manzini, R.: “Managing Lumpy Demand For Aircraft Spare Parts”, **Journal of Air Transport Management**, C: 11, 2005, pp. 426-431.
- Rojas, R.: **Neural Network A Systematic Introduction**, Berlin, Springer-Verlag, 1996.
- Schölkopf, B. ve Smola, A. J.: **Learning With Kernels: Support Vector Machines, Regularization, Optimization, And Beyond**, Cambridge, The MIT Press, 2002.
- Shale, E. A., Boylan, J. E. ve Johnston, F. R.: “Forecasting For Intermittent Demand: The Estimation Of An Unbiased Average”, **The Jorunal Of The Operational Research Society**, C: 57, 2006, pp. 588-592.
- Shalev-Schwartz, S. ve Ben-David, S.: **Understanding Machine Learning From Theory To Algorithms**, NY, USA, Cambridge University Press, 2014.

- Shenstone, L. ve Hyndman, R. J.: “Stochastic Models Underlying Croston's Method For Intermittent Demand Forecasting”, **Journal of Forecasting**, C: 24, 2005, pp. 389-402.
- Silver, E. A., Pyke, D. F. ve Peterson, R.: **Inventory Management And Production Planning And Scheduling**, Third Edition, New York, USA, John Wiley & Sons, 1998.
- Simon, G., Lendasse, A., Cottrell, M., Fort, J. Ve Verleysen, M.: “Time Series Forecasting: Obtaining Long Term Trends With Self-organizing Maps”, **Pattern Recognition Letters**, C: 26, 2005, pp. 1795-1808.
- Snyder, R. D., Ord, J. K. Ve Beaumont, A.: “Forecasting The Intermittent Demand For Slow-mowing Inventories: A Modelling Approach”, **International Journal of Forecasting**, C: 28, 2012, pp. 485-496.
- Steinwart, I. ve Christmann, A.: **Support Vector Machines**, NY, USA, Springer Science+Business Media, 2008.
- Stevenson, W. J.: **Operations Management**, Tenth Edition, New York, USA, McGraw Hill International Edition, 2009.
- Storkey, A. ve Valabregue, R.: “The Basins Of Attraction Of A New Hopfield Learning Rule”, **Neural Networks**, C: 12, 1999, pp. 869-876.
- Suyken, J., Horvath, G., Micchelli, C. ve Vandewalle, J.: “Advances In Learning Theory: Methods, Models And Application”, **NATO-ASI Series 3: Computer and Systems Sciences**, Amsterdam, The Netherlands: IOS Press, 2003.
- Syntetos, A. A. ve Boylan, J. E.: “On The Bias Of Intermittent Demand Estimates”, **Int. J. Production Economics**, C: 71, 2001, pp. 457-466.
- Syntetos, A. A. ve Boylan, J. E.: “The Accuracy Of Intermittent Demand Estimates”, **International Journal of Forecasting**, C: 21, 2005, pp. 303-314.
- Syntetos, A. A., Boylan, J. E. ve Croston, J. D.: “On The Categorization Of Demand Patterns”, **Journal of the Operational Research Society**, C: 56, No: 5, 2005, pp. 495-503.
- Şahin, M., Kızılaslan R. ve Demirel Ö.F.: “Forecasting Aviation Spare Parts Demand Using Croston Based Method and Artificial Neural Networks”, **Journal of Economic and Social Research**, C: 15, No: 2, 2013, pp. 1-21.

- Takma, Ç. ve Atıl, H.: “Bootstrap Metodu Ve Uygulanışı Üzerine Bir Çalışma: 2. Güven Aralıkları, Hipotez Testi Ve Regresyon Analizinde Bootstrap Metodu”, **Ege Üniv. Ziraat Fak. Derg.** , C: 43, No: 2, 2006, s. 63-72.
- Teunter, R. H., Syntetos , A. A. ve Babai, M. Z.: “Intermittent Demand: Linking Forecasting To Inventory Obsolescence”, **European Journal of Operational Research**, C: 214, 2011, pp. 606-615.
- Teunter, R. ve Sani, B.: “On The Bias Of Croston's Forecasting Method”, **Journal of Operatinal Research**, C:194, No: 1, 2009, pp.177-183.
- Vapnik, V.: **Estimation Of Dependences Based On Empirical Data**, Second Edition, NY, USA, Springer Science-Business Media Inc, 2006.
- Vapnik, V. N.: “An Overview Of Statistical Learning Theory”, **IEEE Transactions On Neural Networks**, C: 10, No: 5, 1999, pp. 988-999.
- Varghese, V. ve Rossetti, M.: “A Classification Approach For Selecting Forecasting Techniques For Intermittent Demand”, **Proceedings of the 2008 Industrial Engineering Research**, USA, 2008, pp. 863-868.
- Wali, A.: **Clojure For Machine Learning**, Birmingham, UK, Packt Publishing, 2014.
- Wallström, P. ve Segerstedt, A.: “Evaluation Of Forecasting Error Measurements And Techniques For Intermittent Demand”, **Int. J. Production Economics**, C: 128, 2010, pp. 625-636.
- Wang, X., Wen, J., Zhang, Y. ve Wang, Y.: “Real Estate Price Forecasting Based On SVM Optimized by PSO”, **Optik**, C: 125, 2014, pp. 1439-1443.
- Willemain, T. R., Smart, C. N. ve Schwarz, H. F.: “A New Approach To Forecasting Intermittent Demand For Service Parts Inventories”, **International Journal of Forecasting**, C: 20, 2004, pp. 375-387.
- Willemain, T. R., Smart, C. N., Shockor, J. H. ve DeSautels, P. A.: “Forecasting Intermittent Demand In Manufacturing: A Comparative Evaluation Of Croston's Method”, **Int. J. Forecasting**, C: 10, No: 4, 1994, pp. 529-538.

- Wilson, D. ve Martinez, T.: “The General Inefficiency Of Batch Training For Gradient Descent Learning”, **Neural Networks**, C:16, 2003, pp.1429-1451.
- Wright, D. J.: “Forecasting Data Published At Irregular Time Intervals Using An Extension Of Holt's Method”, **Management Science**, C: 32, No: 4, 1986, 499-510.
- Wu, Q.: “The Forecasting Model Based On Wavelet V-support vector Machine”, **Expert Systems with Applications**, C: 36, 2009, pp. 7604-7610.
- Yegnanarayana, B.: **Artificial Neural Network**, Eastern Economy Edition, New Delhi, India, Prentice-Hall of India, 2005.
- Yom-Tov, E.: “Advanced Lectures On Machine Learning”, **An Introduction to Pattern Classification**, C: 3176, 2004, pp. 1-21.
- Yu, H., Khan, F. ve Garaniya, V.: “Risk-based Fault Detection Using Self-Organizing Map”, **Reliability Engineering and System Safety**, C: 139, 2015, pp. 82-96.
- Yu, X., Qi, Z. ve Zhao, Y.: “Support Vector Regression For Newspaper/Magazine Sales Forecasting”, **Procedia Computer Science**, C:17, 2013, pp. 1055-1062.
- Zhang, G., Patuwo, B. E. ve Hu, M. Y.: “Forecasting With Artificial Neural Networks: The State Of The Art”, **International Journal of Forecasting**, C: 14, 1998, pp. 35-62.
- Zhu, X. ve Goldberg, A. B.: **Introduction To Semi-Supervised Learning**, Ed. by. R. J. Brachman, ve T. G. Dietterich, NY, US, Morgan&Claypool Publishers. 2009.
- Zontul, M., Kaynar, O. ve Bircan, H.: “SOM Tipinde Yapay Sinir Ağlarını Kullanarak Türkiye'nin İthalat Yaptığı Ülkelerin Kümelenmesi Üzerine Bir Çalışma”, **C.Ü. İktisadi ve İdari Bilimler Dergisi**, C: 5, No: 2, 2004, pp.47-68.
- Zupan, J.: “Introduction To Artificial Neural Network Methods: What They Are And How To Use Them.”, **Acta Chimica Slovenica** , C: 41, No: 3, 1994, pp. 327-352.
- Zurada, J.: **Introduction To Artificial Neural Systems**, St. Paul, MN, USA, West Publishing Company, 1992.

ÖZGEÇMİŞ

Derya SAATÇIOĞLU, 1981 yılında İstanbul'da doğmuştur. Lisans eğitimini İstanbul Üniversitesi, Ulaştırma ve Lojistik Yüksek Okulu'nda 2004 yılında tamamlamıştır. 2004-2007 yılları arasında bir Lojistik firmasında Lojistik uzmanı olarak çalışmıştır. 2010 yılında ise İstanbul Üniversitesi İşletme Anabilim Dalı Üretim Bilim Dalı'nda yüksek lisansını tamamlamıştır. 2011 yılından beri Beykoz Lojistik Meslek Yüksekokulu Lojistik bölümünde Öğretim Görevlisi olarak çalışmakta olan Derya Saatçioğlu, evli ve iki çocuk annesidir.

