

**YAPAY ZEKA TEKNOLOJİLERİ VE ZAMAN SERİLERİ
ANALİZİ İLE EKONOMİNİN BANKNOT TALEBİNİN TAHMİNİ:
TÜRKİYE GENELİ VE ESKİŞEHİR ŞUBESİ REESKONT
BÖLGESİNİN KARŞILAŞTIRILMASI**

İshak KUTLU

Uzmanlık Tezi

Türkiye Cumhuriyet Merkez Bankası

Eskişehir Şubesi

Eskişehir, Kasım 2023

**YAPAY ZEKA TEKNOLOJİLERİ VE ZAMAN SERİLERİ
ANALİZİ İLE EKONOMİNİN BANKNOT TALEBİNİN TAHMİNİ:
TÜRKİYE GENELİ VE ESKİŞEHİR ŞUBESİ REESKONT
BÖLGESİNİN KARŞILAŞTIRILMASI**

İshak KUTLU

Danışman

Doç. Dr. Emre ÇİMEN

Uzmanlık Tezi

Türkiye Cumhuriyet Merkez Bankası

Eskişehir Şubesi

Eskişehir, Kasım 2023

**TÜRKİYE CUMHURİYET MERKEZ BANKASI
UZMANLIK TEZİ DEĞERLENDİRME TUTANAĞI**

Eskişehir Şubesi Uzman Yardımcısı (14399) İshak KUTLU'nun, "Yapay Zeka Teknolojileri ve Zaman Serileri Analizi ile Ekonominin Banknot Talebinin Tahmini: Türkiye Geneli ve Eskişehir Şubesi Reeskont Bölgesinin Karşılaştırılması" başlıklı tezini görüşmek üzere tez değerlendirme komisyonumuz 20.11.2023 tarihinde toplanmıştır.

Tez çalışması ve yapılan tez savunması sonucunda aday, komisyon üyeleri tarafından karşılarında belirtilen şekilde değerlendirilmiştir:

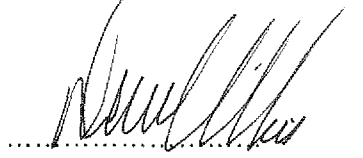
Komisyon Üyesi
Ad-Soyad / Unvan

Değerlendirme
(Başarılı / Başarısız)

İmza

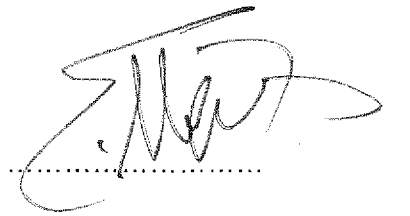
Muhammet Devrim ÜLKÜ
Eskişehir Şubesi Müdürü

Başarılı



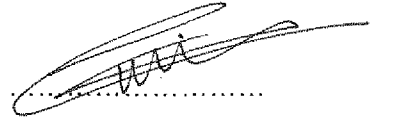
Emrah AKGÜL
Eskişehir Şubesi Müdür Yardımcısı

Başarılı



Doç. Dr. Emre ÇİMEN
Eskişehir Teknik Üniversitesi
Mühendislik Fakültesi

Başarılı



ÖNSÖZ

Bu tez çalışmasının ilk iki bölümünde yapay zeka teknolojilerinden yapay sinir ağları ve zaman serileri analizi teorik seviyede matematiksel ve istatistiksel yöntemlerle incelenmiştir. Üçüncü bölümde ise Python programlama dilinde tasarlanan ileri beslemeli yapay sinir ağı, zaman gecikmeli yapay sinir ağı, uzun kısa süreli bellek ve mevsimsel bütünleşik otoregresif hareketli ortalama modelleriyle, ülkesel ve bölgesel düzeyde banknot talebinin öngörü başarımları kupür bazında araştırılmıştır. Deneysel çalışma, Kaggle platformunda ve Jupyter notebookları üzerinde modüller şeklinde gerçekleştirilmiştir.

Tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve ortaya çıkartılmasında ilgi ve desteğini esirgemeyen, akademik bilgi ve tecrübelerinden yararlandığım, yönlendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren değerli danışmanım Doç. Dr. Emre Çimen'e; desteklerini ve anlayışlarını hiçbir zaman esirgemeyen kıymetli eşime ve kızıma sonsuz teşekkürlerimi sunarım.

Ayrıca Eskişehir Şubesinin banknot tahsil ve tediye verilerini sağlayan Emisyon Genel Müdürlüğü Banknot Planlama Müdürlüğünden Fırat Emre Karataş ve Ahmetcan Günay'a desteklerinden dolayı teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	i
İÇİNDEKİLER.....	ii
TABLO LİSTESİ.....	v
GRAFİK LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	xi
KISALTMA LİSTESİ.....	xii
SEMBOL LİSTESİ.....	xv
EK LİSTESİ.....	xviii
ÖZET.....	xix
ABSTRACT.....	xx
GİRİŞ.....	1

BİRİNCİ BÖLÜM

YAPAY ZEKA TEKNOLOJİLERİ.....	6
1.1. Yapay Sinir Ağlarına Giriş.....	8
1.2. Yapay Sinir Ağlarının Gelişimi.....	10
1.3. Yapay Sinir Ağlarının Genel Yapısı.....	11
1.3.1. Biyolojik Sinir Hücreleri.....	12
1.3.2. Yapay Nöronlar.....	15
1.3.3. Aktivasyon Fonksiyonları.....	18
1.4. Yapay Sinir Ağı Mimarileri.....	23
1.4.1. Tek Katmanlı İleri Beslemeli Ağlar.....	24
1.4.2. Çok Katmanlı İleri Beslemeli Ağlar.....	25
1.4.3. Geri Beslemeli ya da Yinelemeli Ağlar.....	26
1.5. Öğrenme Yöntemleri.....	27
1.5.1. Denetimli Öğrenme.....	27
1.5.2. Denetimsiz Öğrenme.....	28
1.5.3. Pekiştirmeli Öğrenme.....	28
1.6. Tek Katmanlı İleri Beslemeli Ağlar.....	29

1.6.1. Tek Katmanlı Algılayıcı ve Hebb Öğrenme Algoritması.....	29
1.6.2. Adaptif Doğrusal Eleman ve Delta Öğrenme Algoritması.....	39
1.7. Çok Katmanlı İleri Beslemeli Ağlar.....	47
1.7.1. Geri Yayılım Algoritması.....	49
1.7.1.1. Çıktı Katmanına İlişkin Sinaptik Ağırlıkların Analizi.....	52
1.7.1.2. Gizli Katmanlara İlişkin Sinaptik Ağırlıkların Analizi.....	54
1.7.1.2.1. İkinci Katmana İlişkin Sinaptik Ağırlıkların Analizi.....	54
1.7.1.2.2. Birinci Katmana İlişkin Sinaptik Ağırlıkların Analizi.....	57
1.7.1.3. Momentum Parametresi.....	60
1.8. Yapay Sinir Ağlarında Performans İyileştirmesi.....	62
1.8.1. Çapraz Doğrulama Yöntemleri.....	62
1.8.2. Aşırı Öğrenme ve Yetersiz Öğrenme Problemleri.....	64
1.8.3. Erken Durdurma Kriteri.....	67
1.8.4. Yerel Minimuma Yakınsama Problemi.....	69
1.9. Yapay Sinir Ağı Tasarımı.....	71

İKİNCİ BÖLÜM

ZAMAN SERİLERİ ANALİZİ.....	77
2.1. Durağanlık ve Birim Kök Analizi.....	77
2.1.1. Saf Rassal Yürüyüş Modeli.....	79
2.1.2. Sabit İçeren Rassal Yürüyüş Modeli.....	95
2.1.3. Trend İçeren Rassal Yürüyüş Modeli.....	102
2.2. Durağan Olmayan Serilerin Durağanlaştırılması.....	107
2.3. Durağanlığın Araştırılması.....	109
2.3.1. Otokorelasyon Fonksiyonu (ACF).....	109
2.3.2. Q İstatistiği ya da Ljung–Box (LB) İstatistiği.....	112
2.3.3. Genişletilmiş Dickey-Fuller Birim Kök Testi.....	114
2.4. Eşbütünleşme ve Genişletilmiş Engle-Granger Testi.....	116
2.5. Zaman Serisi Modelleri.....	119
2.5.1. Otoregresif Model (AR)	120
2.5.2. Hareketli Ortalama Modeli (MA).....	121
2.5.2.1. MA(q) Modelinin Türetilmesi.....	124
2.5.2.2. MA(q) Modelinin Durağanlık Analizi.....	126
2.5.2.2.1. MA(q) Modelinin Beklenen Değeri.....	126
2.5.2.2.2. MA(q) Modelinin Varyansı.....	127

2.5.2.2.3. MA(q) Modelinin Kovaryansları.....	128
2.5.2.2.4. MA(q) Modelinin Otokorelasyon Parametreleri.....	132
2.5.3. Otoregresif Hareketli Ortalama Modeli (ARMA)	134
2.5.4. Bütünleşik Otoregresif Hareketli Ortalama Modeli (ARIMA)	136
2.5.5. Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama Modeli (SARIMA).....	136
2.5.6. Dışsal Değişkenli Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama Modeli (SARIMAX).....	141
2.6. Box-Jenkins Yöntemi.....	142
2.6.1. Geçici Modelin Tasarlanması.....	143
2.6.2. Parametrelerin Tahmin Edilmesi.....	145
2.6.3. Model Yeterliliğinin Araştırılması.....	145
2.6.4. Model Öngörüsünün Değerlendirilmesi.....	146

ÜÇÜNCÜ BÖLÜM

TÜRKİYE GENELİ VE ESKİŞEHİR ŞUBESİ REESKONT BÖLGESİ: YAPAY SİNİR AĞLARI VE ZAMAN SERİLERİ ANALİZİ İLE BANKNOT TALEBİNİN TAHMİNİNE İLİŞKİN UYGULAMA.....	147
3.1. Performans Metrikleri.....	148
3.2. Banknot Talebinin Tahmini.....	149
3.2.1. Türkiye Geneli İçin Kupürlere Göre Tahminler.....	152
3.2.2. Eskişehir Şubesi İçin Kupürlere Göre Tahminler.....	165
3.3. Modellerin Değerlendirilmesi.....	178

DÖRDÜNCÜ BÖLÜM

SONUÇ VE ÖNERİLER.....	180
KAYNAKÇA.....	182
EKLER.....	185

TABLO LİSTESİ

Sayfa No

Tablo 3.1. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 200 TL	152
Tablo 3.2. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 100 TL	155
Tablo 3.3. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 50 TL	157
Tablo 3.4. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 20 TL	159
Tablo 3.5. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 10 TL	161
Tablo 3.6. Türkiye Geneli İçin Tasarlanan Modellerin Bilgileri: 5 TL	163
Tablo 3.7. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 200 TL....	165
Tablo 3.8. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 100 TL....	168
Tablo 3.9. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 50 TL.....	170
Tablo 3.10. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 20 TL....	172
Tablo 3.11. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 10 TL....	174
Tablo 3.12. Eskişehir Şubesi İçin Tasarlanan Modellerin Bilgileri: 5 TL.....	176
Tablo 3.13. Modellerin RMSE Metriğine Göre Kupür Bazında Öngörü Performansları.....	179

GRAFİK LİSTESİ

	<u>Sayfa No</u>
Grafik 1.1. Aktivasyon Geriliminin Aşamaları.....	14
Grafik 1.2.a. Adım Fonksiyonu.....	21
Grafik 1.2.b. Çift Kutuplu Adım Fonksiyonu.....	21
Grafik 1.2.c. Lojistik Fonksiyon.....	22
Grafik 1.2.d. Hiperbolik Tanjant Fonksiyonu.....	22
Grafik 1.2.e. Simetrik Rampa Fonksiyonu.....	22
Grafik 1.2.f. Doğrusal (Birim) Fonksiyon.....	22
Grafik 1.2.g. ReLU Fonksiyonu.....	22
Grafik 1.2.h. SELU Fonksiyonu.....	22
Grafik 1.3.a. Doğrusal Sınıflandırma.....	33
Grafik 1.3.b. Doğrusal Olmayan Sınıflandırma.....	33
Grafik 1.4. Gradyan Azalma: Sinaptik Ağırlıkların Güncellenmesi	45
Grafik 1.5. Gradyan Azalma: Adım Sayısı	46
Grafik 1.6. Momentum Parametresi.....	61
Grafik 1.7. Aşırı ve Normal Öğrenen MLP Topolojileri.....	65
Grafik 1.8.a. Yetersiz Öğrenme.....	66
Grafik 1.8.b. Normal Öğrenme.....	66
Grafik 1.8.c. Aşırı Öğrenme.....	66
Grafik 1.9. Varyans ve Sapma Arasındaki Değiş-Tokuş.....	67
Grafik 1.10. Erken Durdurma Kriteri.....	68
Grafik 1.11. Yerel Minimuma Yakınsama Problemi.....	70
Grafik 1.12. Gerçek Bir Hata Fonksiyonu Örneği.....	70

Grafik 2.1. AR(1) İçin ACF ve PACF: Durağan Seri.....	111
Grafik 2.2. AR(1) İçin ACF ve PACF: Durağan Olmayan Seri.....	111
Grafik 2.3. MA(1) İçin ACF ve PACF.....	123
Grafik 2.4. MA(2) İçin ACF ve PACF.....	124
Grafik 3.1.a. Türkiye Geneli: 200 TL İçin FNN Hata Optimizasyonu.....	153
Grafik 3.1.b. Türkiye Geneli: 200 TL İçin FNN Çıktısı.....	153
Grafik 3.1.c. Türkiye Geneli: 200 TL İçin TLNN Hata Optimizasyonu.....	153
Grafik 3.1.d. Türkiye Geneli: 200 TL İçin TLNN Çıktısı.....	153
Grafik 3.1.e. Türkiye Geneli: 200 TL İçin LSTM Hata Optimizasyonu.....	153
Grafik 3.1.f. Türkiye Geneli: 200 TL İçin LSTM Çıktısı.....	153
Grafik 3.1.g. Türkiye Geneli: 200 TL İçin Hataların Otokorelasyonu.....	153
Grafik 3.1.h. Türkiye Geneli: 200 TL İçin SARIMA Çıktısı.....	153
Grafik 3.2.a. Türkiye Geneli: 100 TL İçin FNN Hata Optimizasyonu.....	155
Grafik 3.2.b. Türkiye Geneli: 100 TL İçin FNN Çıktısı.....	155
Grafik 3.2.c. Türkiye Geneli: 100 TL İçin TLNN Hata Optimizasyonu.....	156
Grafik 3.2.d. Türkiye Geneli: 100 TL İçin TLNN Çıktısı.....	156
Grafik 3.2.e. Türkiye Geneli: 100 TL İçin LSTM Hata Optimizasyonu.....	156
Grafik 3.2.f. Türkiye Geneli: 100 TL İçin LSTM Çıktısı.....	156
Grafik 3.2.g. Türkiye Geneli: 100 TL İçin Hataların Otokorelasyonu.....	156
Grafik 3.2.h. Türkiye Geneli: 100 TL İçin SARIMA Çıktısı.....	156
Grafik 3.3.a. Türkiye Geneli: 50 TL İçin FNN Hata Optimizasyonu.....	158
Grafik 3.3.b. Türkiye Geneli: 50 TL İçin FNN Çıktısı.....	158
Grafik 3.3.c. Türkiye Geneli: 50 TL İçin TLNN Hata Optimizasyonu.....	158
Grafik 3.3.d. Türkiye Geneli: 50 TL İçin TLNN Çıktısı.....	158
Grafik 3.3.e. Türkiye Geneli: 50 TL İçin LSTM Hata Optimizasyonu.....	158
Grafik 3.3.f. Türkiye Geneli: 50 TL İçin LSTM Çıktısı.....	158
Grafik 3.3.g. Türkiye Geneli: 50 TL İçin Hataların Otokorelasyonu.....	158

Grafik 3.3.h. Türkiye Geneli: 50 TL İçin SARIMA Çıktısı.....	158
Grafik 3.4.a. Türkiye Geneli: 20 TL İçin FNN Hata Optimizasyonu.....	160
Grafik 3.4.b. Türkiye Geneli: 20 TL İçin FNN Çıktısı.....	160
Grafik 3.4.c. Türkiye Geneli: 20 TL İçin TLNN Hata Optimizasyonu.....	160
Grafik 3.4.d. Türkiye Geneli: 20 TL İçin TLNN Çıktısı.....	160
Grafik 3.4.e. Türkiye Geneli: 20 TL İçin LSTM Hata Optimizasyonu.....	160
Grafik 3.4.f. Türkiye Geneli: 20 TL İçin LSTM Çıktısı.....	160
Grafik 3.4.g. Türkiye Geneli: 20 TL İçin Hataların Otokorelasyonu.....	161
Grafik 3.4.h. Türkiye Geneli: 20 TL İçin SARIMA Çıktısı.....	161
Grafik 3.5.a. Türkiye Geneli: 10 TL İçin FNN Hata Optimizasyonu.....	162
Grafik 3.5.b. Türkiye Geneli: 10 TL İçin FNN Çıktısı.....	162
Grafik 3.5.c. Türkiye Geneli: 10 TL İçin TLNN Hata Optimizasyonu.....	162
Grafik 3.5.d. Türkiye Geneli: 10 TL İçin TLNN Çıktısı.....	162
Grafik 3.5.e. Türkiye Geneli: 10 TL İçin LSTM Hata Optimizasyonu.....	162
Grafik 3.5.f. Türkiye Geneli: 10 TL İçin LSTM Çıktısı.....	162
Grafik 3.5.g. Türkiye Geneli: 10 TL İçin Hataların Otokorelasyonu.....	163
Grafik 3.5.h. Türkiye Geneli: 10 TL İçin SARIMA Çıktısı.....	163
Grafik 3.6.a. Türkiye Geneli: 5 TL İçin FNN Hata Optimizasyonu.....	164
Grafik 3.6.b. Türkiye Geneli: 5 TL İçin FNN Çıktısı.....	164
Grafik 3.6.c. Türkiye Geneli: 5 TL İçin TLNN Hata Optimizasyonu.....	164
Grafik 3.6.d. Türkiye Geneli: 5 TL İçin TLNN Çıktısı.....	164
Grafik 3.6.e. Türkiye Geneli: 5 TL İçin LSTM Hata Optimizasyonu.....	164
Grafik 3.6.f. Türkiye Geneli: 5 TL İçin LSTM Çıktısı.....	164
Grafik 3.6.g. Türkiye Geneli: 5 TL İçin Hataların Otokorelasyonu.....	165
Grafik 3.6.h. Türkiye Geneli: 5 TL İçin SARIMA Çıktısı.....	165
Grafik 3.7.a. Eskişehir Şubesi: 200 TL İçin FNN Hata Optimizasyonu.....	166
Grafik 3.7.b. Eskişehir Şubesi: 200 TL İçin FNN Çıktısı.....	166

Grafik 3.7.c. Eskişehir Şubesi: 200 TL İçin TLNN Hata Optimizasyonu.....	166
Grafik 3.7.d. Eskişehir Şubesi: 200 TL İçin TLNN Çıktısı.....	166
Grafik 3.7.e. Eskişehir Şubesi: 200 TL İçin LSTM Hata Optimizasyonu.....	167
Grafik 3.7.f. Eskişehir Şubesi: 200 TL İçin LSTM Çıktısı.....	167
Grafik 3.7.g. Eskişehir Şubesi: 200 TL İçin Hataların Otokorelasyonu.....	167
Grafik 3.7.h. Eskişehir Şubesi: 200 TL İçin SARIMA Çıktısı.....	167
Grafik 3.8.a. Eskişehir Şubesi: 100 TL İçin FNN Hata Optimizasyonu.....	168
Grafik 3.8.b. Eskişehir Şubesi: 100 TL İçin FNN Çıktısı.....	168
Grafik 3.8.c. Eskişehir Şubesi: 100 TL İçin TLNN Hata Optimizasyonu.....	169
Grafik 3.8.d. Eskişehir Şubesi: 100 TL İçin TLNN Çıktısı.....	169
Grafik 3.8.e. Eskişehir Şubesi: 100 TL İçin LSTM Hata Optimizasyonu.....	169
Grafik 3.8.f. Eskişehir Şubesi: 100 TL İçin LSTM Çıktısı.....	169
Grafik 3.8.g. Eskişehir Şubesi: 100 TL İçin Hataların Otokorelasyonu.....	169
Grafik 3.8.h. Eskişehir Şubesi: 100 TL İçin SARIMA Çıktısı.....	169
Grafik 3.9.a. Eskişehir Şubesi: 50 TL İçin FNN Hata Optimizasyonu.....	171
Grafik 3.9.b. Eskişehir Şubesi: 50 TL İçin FNN Çıktısı.....	171
Grafik 3.9.c. Eskişehir Şubesi: 50 TL İçin TLNN Hata Optimizasyonu.....	171
Grafik 3.9.d. Eskişehir Şubesi: 50 TL İçin TLNN Çıktısı.....	171
Grafik 3.9.e. Eskişehir Şubesi: 50 TL İçin LSTM Hata Optimizasyonu.....	171
Grafik 3.9.f. Eskişehir Şubesi: 50 TL İçin LSTM Çıktısı.....	171
Grafik 3.9.g. Eskişehir Şubesi: 50 TL İçin Hataların Otokorelasyonu.....	171
Grafik 3.9.h. Eskişehir Şubesi: 50 TL İçin SARIMA Çıktısı.....	171
Grafik 3.10.a. Eskişehir Şubesi: 20 TL İçin FNN Hata Optimizasyonu.....	173
Grafik 3.10.b. Eskişehir Şubesi: 20 TL İçin FNN Çıktısı.....	173
Grafik 3.10.c. Eskişehir Şubesi: 20 TL İçin TLNN Hata Optimizasyonu.....	173
Grafik 3.10.d. Eskişehir Şubesi: 20 TL İçin TLNN Çıktısı.....	173
Grafik 3.10.e. Eskişehir Şubesi: 20 TL İçin LSTM Hata Optimizasyonu.....	173

Grafik 3.10.f. Eskişehir Şubesi: 20 TL İçin LSTM Çıktısı.....	173
Grafik 3.10.g. Eskişehir Şubesi: 20 TL İçin Hataların Otokorelasyonu.....	173
Grafik 3.10.h. Eskişehir Şubesi: 20 TL İçin SARIMA Çıktısı.....	173
Grafik 3.11.a. Eskişehir Şubesi: 10 TL İçin FNN Hata Optimizasyonu.....	175
Grafik 3.11.b. Eskişehir Şubesi: 10 TL İçin FNN Çıktısı.....	175
Grafik 3.11.c. Eskişehir Şubesi: 10 TL İçin TLNN Hata Optimizasyonu.....	175
Grafik 3.11.d. Eskişehir Şubesi: 10 TL İçin TLNN Çıktısı.....	175
Grafik 3.11.e. Eskişehir Şubesi: 10 TL İçin LSTM Hata Optimizasyonu.....	175
Grafik 3.11.f. Eskişehir Şubesi: 10 TL İçin LSTM Çıktısı.....	175
Grafik 3.11.g. Eskişehir Şubesi: 10 TL İçin Hataların Otokorelasyonu.....	175
Grafik 3.11.h. Eskişehir Şubesi: 10 TL İçin SARIMA Çıktısı.....	175
Grafik 3.12.a. Eskişehir Şubesi: 10 TL İçin FNN Hata Optimizasyonu.....	177
Grafik 3.12.b. Eskişehir Şubesi: 10 TL İçin FNN Çıktısı.....	177
Grafik 3.12.c. Eskişehir Şubesi: 10 TL İçin TLNN Hata Optimizasyonu.....	177
Grafik 3.12.d. Eskişehir Şubesi: 10 TL İçin TLNN Çıktısı.....	177
Grafik 3.12.e. Eskişehir Şubesi: 10 TL İçin LSTM Hata Optimizasyonu.....	177
Grafik 3.12.f. Eskişehir Şubesi: 10 TL İçin LSTM Çıktısı.....	177
Grafik 3.12.g. Eskişehir Şubesi: 10 TL İçin Hataların Otokorelasyonu.....	177
Grafik 3.12.h. Eskişehir Şubesi: 10 TL İçin SARIMA Çıktısı.....	177

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Yapay Zekanın Bileşenleri	7
Şekil 1.2. Biyolojik Sinir Hücresi	12
Şekil 1.3. Biyolojik Sinir Hücreleri Arasında Sinaptik Bağlantı	13
Şekil 1.4. Yapay Nöron.....	16
Şekil 1.5. Tek Katmanlı İleri Beslemeli Mimari.....	25
Şekil 1.6. Çok Katmanlı İleri Beslemeli Mimari.....	26
Şekil 1.7. Geri Beslemeli Mimari.....	27
Şekil 1.8. Tek Nöronlu ve Tek Katmanlı Algılayıcı.....	30
Şekil 1.9. Çok Nöronlu ve Tek Katmanlı Algılayıcı.....	34
Şekil 1.10. ADALINE.....	39
Şekil 1.11. Üç Katmanlı MLP	48
Şekil 1.12. Geri Yayılım Algoritması.....	50
Şekil 1.13. Rassal Örneklenen Çapraz Doğrulama.....	63
Şekil 1.14. Çok Katmanlı Çapraz Doğrulama.....	64
Şekil 1.15. MLP Eğitim ve Uygulama Aşamaları.....	72
Şekil 2.1. Box-Jenkins Yöntemi.....	143

KISALTMA LİSTESİ

ADALINE	: Adaptive Linear Element (Adaptif Doğrusal Eleman)
ADAM	: Adaptive Moment Estimation (Adaptif Moment Tahmini)
AIC	: Akaike Information Criterion (Akaike Bilgi Kriteri)
AI	: Artificial Intelligence (Yapay Zeka)
ANN	: Artificial Neural Network (Yapay Sinir Ağları)
ADF	: Augmented Dickey-Fuller (Genişletilmiş Dickey-Fuller)
AEG	: Augmented Engle-Granger (Genişletilmiş Engle-Granger)
ACF	: Autocorrelation Function (Otokorelasyon Fonksiyonu)
ATM	: Automated Teller Machine (Bankomat)
AR	: Autoregressive (Oto regresif)
ARCH	: Autoregressive Conditional Heteroskedasticity (Oto regresif Koşullu Varyans)
ARIMA	: Autoregressive Integration Moving Average (Bütünleşik Oto regresif Hareketli Ortalama)
ARIMAX	: Autoregressive Integration Moving Average Exogenous (Dışsal Değişkenli Bütünleşik Oto regresif Hareketli Ortalama)
ARMA	: Autoregressive Moving Average (Oto regresif Hareketli Ortalama)
BIC	: Bayes Information Criterion (Bayes Bilgi Kriteri)
CNN	: Convolutional Neural Network (Evrimsel Sinir Ağı)
DNN	: Deep Neural Network (Derin Sinir Ağı)
DF	: Dickey-Fuller
EG	: Engle-Granger
EDA	: Exploratory Data Analysis (Keşifsel Veri Analizi)
EVDS	: Elektronik Veri Dağıtım Sistemi

FNN	: Feed Forward Neural Network (İleri Beslemeli Sinir Ağı)
GSYİH	: Gayrisafi Yurt İçi Hasıla
GARCH	: Generalized Autoregressive Conditional Heteroskedasticity (Genelleştirilmiş Otoregresif Koşullu Varyans)
GPU	: Graphic Process Unit (Grafik İşleme Birimi)
KNN	: K Nearest Neighbor (K En Yakın Komşu)
LMS	: Least Mean Squares (En Küçük Ortalama Kareler)
LSTM	: Long Short-Term Memory (Kısa ve Uzun Dönemli Hafıza)
MAPE	: Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hata)
MSE	: Mean Square Error (Ortalama Hata Karesi)
MA	: Moving Average (Hareketli Ortalama)
MLP	: Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
MADALINE	: Multiple Adaptive Linear Element (Çoklu Adaptif Doğrusal Eleman)
NN	: Neural Network (Sinir Ağı)
PACF	: Partial Autocorrelation Function (Kısmi Otokorelasyon Fonksiyonu)
RELU	: Rectified Linear Unit (Düzeltilmiş Doğrusal Birim)
R-CNN	: Region-Based Convolutional Neural Network (Bölgeye Dayalı Evrimsel Sinir Ağı)
RegARIMA	: Regression Autoregressive Integration Moving Average (Regresyonlu Bütünleşik Otoregresif Hareketli Ortalama)
ResNet	: Residual Neural Network (Kalıntılı Sinir Ağı)
RSS	: Residual Sum of Squares (Hata Kareleri Toplamı)
RMSE	: Root Mean Square Error (Ortalama Hata Karesinin Karekökü)
SELU	: Scaled Exponential Linear Unit (Ölçekli Üssel Doğrusal Birim)
SIC	: Schwarz Information Criterion (Schwarz Bilgi Kriteri)
SAR	: Seasonal Autoregressive (Mevsimsel AR)

SARIMA	: Seasonal Autoregressive Integration Moving Average (Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama)
SARIMAX	: Seasonal Autoregressive Integration Moving Average Exogenous (Dışsal Değişkenli Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama)
SMA	: Seasonal Moving Average (Mevsimsel MA)
SVM	: Support Vector Machine (Destek Vektör Makinesi)
SMAPE	: Symmetric Mean Absolute Percentage Error (Simetrik Ortalama Mutlak Yüzde Hata)
TCMB	: Türkiye Cumhuriyet Merkez Bankası
TLNN	: Time Lagged Neural Network (Zaman Gecikmeli Sinir Ağı)
VAR	: Vector Autoregression (Vektör Otoregresif)
VECM	: Vector Error Correction Model (Vektör Hata Düzeltme Modeli)
VGG	: Visual Geometry Group (Görsel Geometri Grubu)
YOLO	: You Only Look Once (Nesnelerin genelleştirilmiş temsillerine dayanan yapay sinir ağı temelli nesne tanıma modeli)

SEMBOL LİSTESİ

YAPAY SİNİR AĞLARI İÇİN SEMBOL LİSTESİ

u	: Aktivasyon Gerilimi
$f(u)$	: Operasyonel Fonksiyon
$g(u)$	: Aktivasyon Fonksiyonu
w_i	: Sinaptik Ağırlıklar
x_i	: Girdi Sinyalleri
θ	: Aktivasyon Eşiği ya da Sabit
y	: Çıktı Sinyali
e	: Euler Sabiti
n	: Girdi Sayısı
t	: Örnek Sayısı
k	: Örnek Sırası
$\mathbf{w}$	: Ağırlık Vektörü
$\mathbf{W}$	: Ağırlık Matrisi
$\mathbf{x}$	: Girdi Vektörü
$\mathbf{X}$	: Girdi Matrisi
$\mathbf{u}$	: Aktivasyon Gerilimi Vektörü
$\mathbf{U}$	: Aktivasyon Gerilimi Matrisi
s	: Adım Sırası
α	: Öğrenme Parametresi
$\mathbf{y}$	: Çıktı Vektörü
$\mathbf{Y}$	: Çıktı Matrisi
$\mathbf{d}$	: Gerçek Değer Vektörü

D	: Gerçek Değer Matrisi
m	: Nöron Sayısı
E(.)	: Beklenen Değer
∇	: Matris ya da Vektör Türevi
Δ	: Türev
∂	: Kısmi Türev Operatörü
δ^i	: i. Katmanın Gradyan Vektörü
a^i	: i. Katmanın Ağırlıklandırılmış Girdi Vektörü
e^i	: i. Katmanın Hata Vektörü
λ	: Momentum Parametresi
L	: Katman Sırası

ZAMAN SERİLERİ ANALİZİ İÇİN SEMBOL LİSTESİ

y_t	: Zaman Serisi
μ	: Ortalama
σ	: Standart Sapma
σ^2	: Varyans
Var(.)	: Varyans
Cov(.)	: Kovaryans
E(.)	: Beklenen Değer
k	: Gecikme Uzunluğu
γ_k	: k. Gecikme Uzunluğunda Kovaryans
γ_0	: Varyans
t	: Gözlem Sırası
T	: Gözlem Sayısı
R^2	: Belirlilik Katsayısı
ρ	: Otokorelasyon Katsayısı

ε	: Hata Terimi
σ_{ε}^2	: Hata Teriminin (Sabit) Varyansı
m	: Maksimum Gecikme Uzunluğu
t'	: Trend Değişkeni
τ	: Tau İstatistiği
p	: AR Sürecinin Sırası
q	: MA Sürecinin Sırası
d	: Normal Fark
D	: Mevsimsel Fark
P	: Mevsimsel AR Sürecinin Sırası
Q	: Mevsimsel MA Sürecinin Sırası
s	: Mevsim Periyodu

EK LİSTESİ

Sayfa No

Ek 1. Türkiye Geneli ve Eskişehir Şubesi İçin SARIMA İstatistikleri	186
Ek 2. Türkiye Geneli İçin Kupür Bazında Emisyon Hacmi Verileri	192

ÖZET

Merkez bankaları parasal ve finansal istikrar hedeflerini gerçekleştirebilmeleri için para piyasası likiditesini en az hata ile tahmin etmeyi amaçlar. Emisyon hacmi ya da tedavüldeki banknot miktarı ise para piyasası likiditesinin en önemli otonom değişkenlerinden biridir. Emisyon hacmi makroekonomik değişkenlerle birlikte birçok dışsal faktörden güçlü bir şekilde etkilendiği için öngörülmesi de oldukça zordur. Bu çalışmada yapay zeka teknolojilerinden ileri beslemeli (ya da çok katmanlı algılayıcı), zaman gecikmeli ve uzun kısa süreli bellek tipi yapay sinir ağları ve zaman serileri analizine dayanan mevsimsel bütünleşik otoregresif hareketli ortalama modeli ile ülkesel ve (Eskişehir Şubesi özelinde) bölgesel seviyede, herhangi bir makroekonomik ya da dışsal değişken kullanılmadan emisyon talepleri öngörülme çalışılmıştır. Bu amaçla yapay sinir ağlarının ve zaman serilerinin yapısı ve işleyişi teorik çerçevede, matematiksel ve istatistiksel yöntemlerle detaylı olarak incelenmiştir. Sonrasında ülkesel ve bölgesel seviyedeki veri setleri kullanılarak, kupür bazında yapay sinir ağları ve zaman serilerine ilişkin toplam 48 modelin eğitimi ve tasarımı Python programlama dilinde gerçekleştirilmiştir. Söz konusu modellerin öngörü başarımları kıyaslanarak modellerin geliştirilmesine yönelik önerilerde bulunulmuştur.

Anahtar Kelimeler: Emisyon Hacmi, Öngörü, Yapay Zeka, Yapay Sinir Ağları, Zaman Serileri Analizi.

ABSTRACT

Central banks aim to predict money market liquidity with the minimum error to achieve their monetary and financial stability goals. Emission volume, or the amount of banknotes in circulation, is one of the most important autonomous variables of money market liquidity. It is also challenging to predict since the emission volume is strongly influenced by numerous external factors, along with macroeconomic variables. In this study, national and regional emission demands (specific to the Eskişehir Branch) have been attempted to be predicted using artificial intelligence technologies, namely feed-forward (or multi-layer perceptron), time-lagged, and long short-term memory type artificial neural networks, and a seasonal integrated autoregressive moving average model based on time series analysis. To accomplish this, the structure and functioning of artificial neural networks and time series have been extensively examined using mathematical and statistical methods within the theoretical framework. Subsequently, 48 models of artificial neural networks and time series, based on denominations, were trained and designed in the Python programming language using national and regional datasets. Through a comparison of the prediction performances of the mentioned models, recommendations have been provided for their improvement.

Keywords: Banknotes in Circulation, Forecasting, Artificial Intelligence, Artificial Neural Network, Time Series Analysis.

GİRİŞ

Merkez bankalarının temel görevi fiyat istikrarının yanında, özellikle de 2008 küresel ekonomik krizinden sonra, finansal istikrarı da sağlamak ve sürdürmektir. Merkez bankaları, parasal ve finansal istikrar hedeflerini gerçekleştirmek için para politikası araçlarını kullanmalarının yanı sıra, para biriminin mevcudiyeti, kalitesi ve güvenliği konusunda halkın güvenini kazanması gerekir. Bunu gerçekleştirmek için banknotlarına olan talebi öngörmeleri beklenir. Örneğin artan banknot talebinin karşılanması ve banknotların basımı için gerekli olan bileşenlerin tedarik planının yapılması, gelecek dönemlere ilişkin banknot talebinin öngörülmesini gerektirir (Miller, 2017). Benzer şekilde merkez bankaları, kurum içi banknot tedarik sürecinin koordine edilmesi ve planlanması, banknot işleme sistemlerine olan ilave ihtiyaçların değerlendirilmesi, uzun dönemli stoklama kapasitelerinin gözden geçirilmesi gibi bir dizi kritik problemle baş etmek için banknotlarına olan talebi en az hata ile öngörmek isterler. Böylece ekonomik karar birimlerinin para talebi doğrultusunda, para politikası araçlarını da kullanarak tedavüldeki banknot miktarını, diğer bir ifadeyle emisyon hacmini düzenlerler.

Diğer taraftan hedeflerine etkili bir şekilde ulaşmak isteyen merkez bankalarının para piyasası likiditesini doğru tahmin etmeleri gerekmektedir. Bununla birlikte para piyasası likiditesi, merkez bankasının tam kontrolü altında olmayan birkaç bağımsız faktörden etkilenmektedir. Dolaşımdaki para olarak da ifade edilen emisyon hacmi (ya da tedavüldeki banknot miktarı) en önemli otonom faktörlerden biridir. Makroekonomik değişkenlerin yanı sıra birçok dışsal faktörden güçlü bir şekilde etkilendiği için de emisyon hacminin belirlenmesi oldukça güçtür. Bu çerçevede merkez bankalarının gerçekleştirdikleri likidite tahmini (ya da öngörüsü) için dolaşımdaki para miktarı en büyük ve en değişken bileşendir (Balli ve Elsamadisy, 2010). Doğru bir likidite tahmini sağlamak ve buna bağlı olarak etkili para politikası uygulamak için gelecek dönemlere ilişkin dolaşımdaki para miktarının doğru tahmin edilmesi esastır.

Literatürde Putri ve diğerleri (2021), dolaşımdaki para miktarında meydana gelen değişimleri, takvim etkisini de içerecek şekilde ARIMAX ve bir yapay sinir ağı olan DNN (deep neural network) ile modellemiştir. Ayrıca ARIMAX doğrusal modeli, DNN ise doğrusal olmayan modeli temsil etmek üzere, iki modeli birleştirerek hibrit bir model tasarlamışlardır. Hibrit model trendi, mevsimsel hareketleri, takvim etkilerini ve doğrusal olmayan örüntüleri çıkarmada diğer iki modelden daha başarılı sonuçlar vermiştir.

Miller (2017), banknot talebini etkileyen olası faktörleri belirlemek ve değerlendirmek amacıyla, banka şubesi ve ATM sayısı, faiz oranı, nominal tüketim, işsizlik oranı gibi bir dizi değişkene sahip hata düzeltme modeli (error correction model) isimli ekonometrik bir zaman serisi modeli üzerinde çalışmıştır. Çalışmada Bank of England'ın geçmiş yıllara ilişkin gerçekleşen banknot talebi verileri kullanılmıştır. Gelecekte banknot talebini tam olarak bilmenin mümkün olmadığı, talebin ne olabileceğini ve nasıl etkilenebileceğini anlamak için modellerin sadece bir çerçeve sunduğu, gelecekte bir talep değişikliğine işaret edebilecek öncü göstergelerin belirlenmesinde yardımcı olabileceği sonucuna varılmıştır.

Bartzsch ve diğerleri (2015), banknotları küçük (5-10-20 euro), orta (50-100 euro) ve büyük (200-500 euro) olarak ayırarak Deutsche Bundesbank banknotlarına olan talebi incelemişlerdir. Ayrıca Almanya'da basılan banknotlara olan yabancı talebini, euro bölgesindeki diğer ülkelerin talebi ve euro bölgesi dışındaki ülkelerin talebi olarak iki bölümde incelemişlerdir. Çalışmada, öngörüler için ekonometrik (zaman serisi) modellerden olan VECM (vector error correction model) ve RegARIMA kullanılmıştır.

Ağaslan ve Gayaker (2020), makroekonomik değişkenlerin bulunmadığı yüksek frekanslı ARMA modelinin, makroekonomik değişkenlerin yer aldığı düşük frekanslı VAR (vector autoregression) ve GARCH (generalized autoregressive conditional heteroskedasticity) modellerinin öngörü başarısından daha iyi olduğunu gözlemiştir.

Assenmacher ve diğerleri (2019), işlem ve birikim amacıyla talep edilen para talebine odaklanarak İsviçre'nin banknot talebini etkileyen

faktörleri, günlük veriler üzerinden incelemişlerdir. Mevsimsel faktör olarak hafta içi günleri esas almışlardır. İşlem ya da birikim amacıyla talep edilen banknot miktarının banknotların nominal değerinin (kupür) büyüklüğüne göre değiştiği sonucuna varmışlardır. Diğer bir ifadeyle, büyük kupürlerin (1000-500-200-100 İsviçre frangının), özellikle de 1000 ve 200 nominal değerli İsviçre franklarının, esas olarak birikim amacıyla, küçük kupürlerin (50-20-10-5 İsviçre frangının) ise işlem amacıyla talep gördüğünü değerlendirmişlerdir. Modelleme sürecinde tüketici fiyat indeksi, GSYİH (gayrisafi yurt içi hasıla), döviz kuru gibi makroekonomik değişkenlere dayalı bir regresyon modelinden yararlanmışlardır. Araştırmada, makroekonomik değişkenlerin büyük kupürlerin talebi üzerinde daha belirleyici olduğu yönünde bulgulara rastlanmıştır. Birikim amacıyla tutulan banknot talebinin Bretton Woods sisteminin dağılmasıyla birlikte arttığı, 1990'ların ortalarında nispeten düşük olmasına rağmen ve 2000'li yılların başlangıcından ve 2008 küresel krizden bu yana önemli ölçüde arttığı vurgulanmıştır.

Finlandiya ile birlikte euro bölgesinden İspanya, Fransa, Almanya gibi ülkeleri kapsayan Wagner (2010)'in araştırmasında, günlük frekansa sahip ATM verileri kullanılmıştır. Modelleme aşamasında SARIMA ve VAR olmak üzere iki farklı zaman serisi modelinden yararlanılmıştır. Takvim etkisinin ön planda tutulduğu çalışmanın sonuçlarına göre banknot talebine ilişkin uzun dönemli bir trend bulgusuna rastlanmamıştır. Bununla birlikte günlük, haftalık ve aylık periyotlara işaret eden mevsimsel ve döngüsel davranış örüntüleri tespit edilmiştir.

Tedavüldeki banknot miktarının belirlenmesine ilişkin tahmin edilen modeller için literatürde bir ayırım söz konusudur. Bu ayırımın yapılmasında belirleyici olan faktör ise verilerin frekanslarıdır. Dolaşımdaki para miktarının belirlenmesinde kullanılan modeller genel olarak düşük frekanslı modeller ve yüksek frekanslı modeller olmak üzere ikiye ayrılmaktadır. Emisyon hacminin, düşük frekanslı modellerde ülkenin makroekonomik faktörlerinden etkilendiği, yüksek frekanslı modellerde ise büyük ölçüde hane halkının para talebi tercihi tarafından belirlendiği varsayılır (Ağaslan ve Gayaker, 2020).

Bu çalışmanın amacı ise banknot talebinin kaynağını değil, doğrudan doğruya gelecek dönemlere ilişkin emisyon hacminin yapay zeka teknolojileri ve zaman serileri modelleri ile yapılan öngörü performanslarının araştırılmasıdır. Zaman serileri modellerinden SARIMA ve yapay zeka teknolojilerinden ise FNN (MLP), TLNN ve LSTM tipi yapay sinir ağları Python programlama dili kullanılarak tasarlanmıştır. Bu çerçevede Türkiye geneli ve Eskişehir Şubesi Reeskont Bölgesi para talebi için kupür bazında toplam 48 model tasarlanmış ve eğitilmiştir.

Çalışmanın ilk bölümünde yapay sinir ağlarının gelişimi, türleri ve öğrenme yöntemleri gibi temel yapıları üzerinde durulmuştur. Devamında günümüzde de büyük ölçüde geçerliliğini koruyan ilk yapay sinir ağı olan algılayıcının (perceptron) ve bu yöntemden geliştirilen ADALINE ağına matematiksel yöntemlerle işleyişi incelenmiştir. Sonrasında ise yapay zeka alanında önemli bir dönüm noktası olarak kabul edilen MLP (multi layer perceptron) ve geri yayılım (backpropagation) algoritmaları detaylı olarak incelenmiştir. Son olarak yapay sinir ağlarında performans iyileştirmesine ilişkin yöntemler ve tasarım süreçleri üzerinde durularak çalışmanın ilk bölümü tamamlanmıştır.

İkinci bölümde zaman serilerinin en önemli karakteristiği durağanlık ve deneysel çalışmada kullanılan zaman serisi modelleri detaylı olarak matematiksel ve istatistiksel yöntemlerle incelenmiştir. Durağanlık ve birim kök kavramları, rassal yürüyüş modelleri üzerinden ele alınmıştır. Sonrasında durağanlığın tespiti, Dickey-Fuller birim kök testi, durağan olmayan serilerin durağanlaştırılması ve eşbütünleşme konuları üzerinde durulmuştur. Zaman serilerinin temel modellerinden olan AR ve MA süreçlerinin yapısı ve durağanlık analizi yapılmıştır. Bu analizlerden hareketle ARMA, ARIMA, SARIMA ve SARIMAX modelleri türetilmiştir. Son olarak literatürde de yaygın bir şekilde kabul görmüş, zaman serisi çalışmalarında takip edilecek prosedürleri ifade eden Box-Jenkins yöntemi açıklanmıştır.

Üçüncü bölümde birinci ve ikinci bölümde teorik çerçevesi çizilen yöntemlerle, Python programlama dili kullanılarak kapsamlı bir deneysel uygulama gerçekleştirilmiştir. Veri seti olarak kupür bazında Türkiye geneli ve

Eskişehir Şubesi Reeskont Bölgesinin emisyon hacimleri kullanılmıştır. Yapay sinir ağları kapsamında FNN (MLP), TLNN ve LSTM ağları, zaman serisi modeli olarak ise SARIMA modeli kullanılmıştır. Söz konusu modeller, kupür bazında ayrıştırılan Türkiye geneli ve Eskişehir Şubesi verilerine ayrı ayrı uygulanmıştır. Uygulama sürecinde yapay sinir ağlarından 36 farklı ağ, SARIMA modelinde ise 12 farklı model tasarlanmıştır. Çalışmada serilerin kendi değerleri dışında herhangi bir makroekonomik bağımsız değişken kullanılmamıştır. Bu bölümün sonunda deneysel çalışmada kullanılan modeller değerlendirilmiştir.

Sonuç ve öneriler bölümünde ise araştırmanın ulaştığı en önemli bulgulara yer verilmiş ve modellerin tahmin performanslarının geliştirilebilmesi için önerilerde bulunulmuştur.

BİRİNCİ BÖLÜM

YAPAY ZEKA TEKNOLOJİLERİ

Yapay zekanın en önemli isimlerinden biri olarak kabul edilen John McCarthy'e göre yapay zeka (artificial intelligence, AI), akıllı makineler, özellikle akıllı bilgisayar programları tasarlama bilimi ve mühendisliğidir (Tutorials Point, 2015, s.1). Bu çerçevede yapay zeka, bir bilgisayarı, bilgisayar kontrollü bir robotu veya bir yazılımı zeki insanların düşündüğü gibi zekice düşündürmenin bir yolu olarak ifade edilebilir.

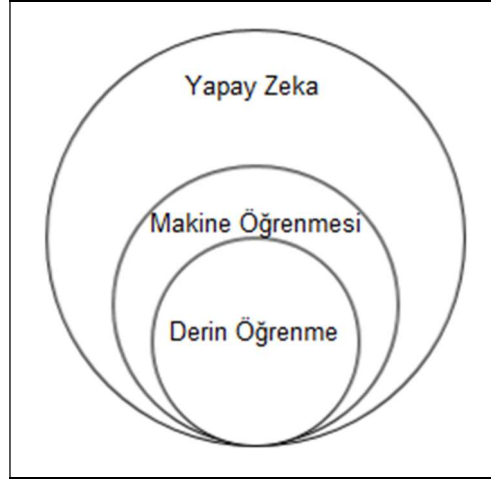
Yapay zeka, bilgisayar bilimi, istatistik, matematik ve mühendislik gibi alanlara dayalı bir bilim ve teknolojidir. Yapay zekanın temel motivasyonu, akıl yürütme, öğrenme ve problem çözme gibi insan zekasıyla ilişkili bilgisayar işlevlerinin geliştirilmesidir.

Bilgisayar programları ve yapay zeka programları çeşitli yönleriyle birbirinden farklılık gösterir. Bir bilgisayar programı, çözüme ilişkin adımların detaylı bir şekilde önceden tasarlandığı “belirli” problemleri çözebilir. Eğer problemin yapısında değişiklik meydana gelirse, bilgisayar programında da kapsamlı değişiklik yapılması gerekir. Söz konusu modifikasyonlar hızlı ve kolay olmayıp programın olumsuz etkilenmesine yol açabilir. Yapay zekaya sahip bir bilgisayar programı ise çözüme ilişkin adımların detaylı bir şekilde önceden tasarlanmasının gerekmediği “genel” problemleri çözebilir. Yapay zeka programları, son derece bağımsız bilgi parçalarını bir araya getirerek problemde meydana gelen değişiklikleri özümseyebilir. Bu nedenle programın yapısını etkilemeden en küçük bir bilgi parçası bile değiştirilebilir. Modifikasyonlar ise hızlı ve kolaydır.

Yapay zeka günümüzde oyun, doğal dil işleme, uzman sistemler, görüş sistemleri, konuşma tanıma, el yazısı tanıma ve zeki robotlar gibi alanlara uygulanmaktadır. Önde gelen bazı araştırma alanları ise yapay sinir

ağları, robotik, bulanık mantık, doğal dil işleme ve uzman sistemler olarak ifade edilebilir (Tutorials Point, 2015, s.1-3, 11).

Yapay zekanın bileşenleri Şekil 1.1’de gösterildiği gibi makine öğrenmesi (machine learning) ve derin öğrenme (deep learning) tekniklerinden oluşur.



Şekil 1.1. Yapay Zekanın Bileşenleri

Kaynak: Nwankpa ve diğerleri, 2018, s.2

Son yıllarda, yapay zekanın bir alt bileşeni olan makine öğrenimi alanı hızla genişlemeye başlamıştır. Bu alandaki araştırmalar, insan aktivitelerinin farklı yönlerine doğru hızlı bir çeşitlik kazanmıştır. Makine öğrenmesi, tahminler ve çıkarım gibi önemli görevleri gerçekleştirmek için istatistik ve bilgisayar bilimi ilkelerini kullanan bir çalışma alanıdır. Bu modeller, belirli bir sistemin girdileri ve çıktıları arasındaki matematiksel ilişkileri ortaya çıkaran yapılardır. Açıkça programlanmadan makineler öğrenme yeteneği vermeyi amaçlayan öğrenme süreci, modelin belirtilen görevi yerine getirebilmesi için model parametrelerini tahmin etme faaliyetidir (Nwankpa ve diğerleri, 2018).

Makine öğrenmesinin bir alt algoritması olan ve son 10 yılda yaygınlaşan derin öğrenme, genellikle CNN (convolutional neural network) adı verilen çok katmanlı yapay sinir ağlarından oluşur. Özellikle de görüntü sınıflandırma, nesne tanıma gibi görsel girdilerin doğrusal olmayan soyut temsillerini çıkartarak (extracting) öğrenen algoritmalarıdır. AlexNet, VGG (visual geometry group), inception (GoogLeNet), ResNet (residual neural network), R-CNN (region-based convolutional neural network), YOLO (you

only look once) ve DeepLab gibi derin öğrenme algoritmalarının öğrenme süreci oldukça uzun zaman gerektirir. Bu tür derin öğrenme algoritmaları yüksek verimlilikte çalışan GPU (graphic process unit) birimine, diğer bir ifadeyle gelişmiş grafik kartlarına ihtiyaç duyar. Bu çalışmanın konusu görsel sınıflandırma ya da nesne tanıma problemleri olmadığı için derin öğrenme yöntemleri üzerinde durulmayacaktır.

ID3, C4.5, twoing, Gini algoritmaları gibi karar ağaçları (decision tree), en yakın komşu (k nearest neighbor, KNN), naive Bayes, destek vektör makineleri (support vector machine, SVM), regresyon, lojistik regresyon, rassal orman (random forest), BFR, CURE, K-means, XGBoost, yapay sinir ağları (artificial neural network) gibi sayısız makine öğrenmesi algoritması bulunmaktadır. Model oluşturma sürecinde, somut olaya ilişkin probleme göre genellikle deneysel çalışmalarda en yüksek başarıyı sağlayan algoritma seçilir. Bunlardan yapay sinir ağları, problem çeşitliliğine göre uygulanabilirliği en geniş makine öğrenmesi yöntemidir. Bu bölümde makine öğrenmesi tekniklerinden, doğrusal ve/veya doğrusal olmayan öngörü problemlerinin çözümünde de yaygın olarak kullanılan yapay sinir ağlarına odaklanılmıştır.

1.1. Yapay Sinir Ağlarına Giriş

Sinir ağı (neural network, NN) olarak da ifade edilen bir yapay sinir ağı (artificial neural network, ANN), deneyimsel bilgiyi belleklemek ve kullanılabilir hale getirmek için basit işlem birimlerinden oluşan, büyük ölçüde paralel dağıtılmış bir işlemci olarak tanımlanabilir. Bir yapay sinir ağı iki açıdan insan beynine benzetilebilir.

- Bilgi, bir ağ tarafından öğrenme süreciyle çevreden elde edilir.
- Sinaptik ağırlıklar (ya da ağırlıklar) olarak bilinen nöronlar arası bağlantı, edinilen bilgileri belleklemek amacıyla kullanılır.

Öğrenme süreci, hedeflenen amaca ulaşmak için sinaptik ağırlıkları değiştiren ve “öğrenme algoritmaları” olarak adlandırılan yazılımsal prosedürler tarafından yerine getirilir. Sinaptik ağırlıkların değiştirilmesi yapay sinir ağlarının geleneksel yöntemleriyle gerçekleştirilebilir. Bununla birlikte,

insan beynindeki nöronların ölebilmesi ve yeni sinaptik bağlantıların oluşabilmesi gerçeğinden hareketle, bir yapay sinir ağının kendi topolojisini (yapısını) değiştirebilmesi de mümkündür (Haykin, 2009, s.2).

Doğrusal ve doğrusal olmayan problemlerin çözümünde kullanılabilen yapay sinir ağları oldukça geniş bir uygulama alanına sahiptir.

- Geleneksel yöntemlerle modellenmesi zor fonksiyon yaklaştırma/evrensel eğri uydurma (function approximation/universal curve fitting) problemlerinin çözümünde,
- Robotik, havacılık, uydular gibi makine-teçhizatların kalite, etkinlik ve güvenlik gereksinimlerinin karşılanması amacıyla süreç kontrol sistemlerinde,
- Nesne ve ses/konuşma algılama gibi görsel ve işitsel süreçlerde örüntü tanıma ve sınıflandırma problemlerinin yanı sıra resim, video gibi görsel nesnelerin ve ses, konuşma gibi işitsel çıktılarının üretilmesinde,
- Çeşitliliğe sahip verilerde, özellikle veri madenciliğinde benzerliklerin tespit edilmesi ve tanımlanması gibi kümeleme problemlerinin çözümünde,
- Finansal ürünlerin fiyat projeksiyonları, hava tahminleri gibi öngörü (forecasting) gerektiren problemlerin çözümünde,
- Dinamik programlama, sınırlandırılmış optimizasyon problemlerinin çözümünde,
- Görüntü ve sinyal işleme gibi bozuk ya da belirsiz verilerden ilişkisel hafıza kullanımıyla doğru örüntülerin çıkartılmasında yapay sinir ağlarından yararlanılır (Silva ve diğerleri, 2017, s.8; Fausett, 1994, s.7-11).

Sektörel açıdan ise yapay sinir ağları, otomotiv, bankacılık, elektronik, finans, eğlence, robotik, tıp, ulaşım gibi oldukça geniş bir uygulama alanı bulmuştur (Hagan ve diğerleri, 2014, s.5-7).

1.2. Yapay Sinir Ağlarının Gelişimi

McCulloch ve Pitts tarafından 1943 yılında, biyolojik nöronlardan (sinir hücrelerinden) esinlenerek geliştirilen matematiksel model, bilinen ilk yapay sinir ağı modeli olarak kabul edilir. Tek bir nörondan oluşan ve basit bir ikili eşik değer fonksiyonu kullanan bu model, modifiye edilerek sonraki çalışmalarda yaygın olarak kullanıldı (Mehrotra ve diğerleri, 1996, s.4-5).

McGill Üniversitesinde bir psikolog olan Donald Hebb, 1949 yılında kendi adıyla da literatüre girecek olan yapay sinir ağlarının ilk öğrenme kuralını tasarladı. Hebb'e göre iki nöron eş anlı olarak aktif olsaydı, onlar arasındaki bağlantının gücü de artmış olmalıydı. Hebb'in önermesi, sonradan iyileştirilerek bilgisayar simülasyonlarına olanak tanıdı. Ayrıca bu önerme 1972 yılında Kohonen ve Anderson tarafından geliştirilen korelasyon matrisi öğrenmesiyle de yakından ilgilidir (Fausett, 1994, s.22).

Çok sayıda araştırmacı, 1950'li yıllarda, biyolojik nöronlara dayanan matematiksel modeller geliştirmeye devam etti. Sonuç olarak, bu yıllarda çok sayıda öğrenme algoritması ve ağ topolojisi geliştirildi. Ancak 1957 ve 1958 yılları arasında, algılayıcının (perceptron) temel modelini ve "Mark I Perceptron" isimli ilk nörobilgisayarı geliştiren Frank Rosenblatt öne çıktı. Algılayıcı, ağırlıkların optimizasyonunda, bir nöronun arzulan davranışı gerçekleştirmesine göre onu bir ağırlıkla (parametreyle çarparak) ödüllendiren ya da cezalandıran Hebb öğrenme kuralını kullanır (Fausett, 1994, s.23; Mehrotra ve diğerleri, 1996, s.5).

Algılayıcı, basit örüntüleri ayırt edebilme kapasitesinden dolayı ilgi uyandırdı. Widrow ve Hoff, 1960 yılında, "uyarlamalı doğrusal element" (ADAPtive LINEar element) anlamına gelen "ADALINE" isimli bir ağ geliştirdi. Sonrasında ise "çoklu uyarlamalı doğrusal element" (Multiple ADALINE) anlamına gelen "MADALINE" tasarlandı. ADALINE ve MADALINE ağlarının öğrenmesi, "en küçük ortalama kare" (least mean square, LMS) modeli olarak da bilinen "delta kuralı"na dayanan bir ağdan oluşur.

Algılayıcıya dayanan yapay sinir ağları, 1969 yılında Minsky ve Papert tarafından "Perceptrons" isimli kitabın yayımlanmasıyla belirgin bir duraklama

yaşadı. Kitap, algılayıcı ve ADALINE gibi tek katmandan oluşan dönemin yapay sinir ağlarının, (XOR gibi çok temel mantıksal fonksiyonların çıktıları ve girdileri arasındaki ilişkiyi öğrenme konusundaki yetersizliği üzerinde durarak) doğrusal olmayan örüntüleri sınıflandırmasının mümkün olmadığını gösterdi (Silva ve diğerleri, 2017, s.6-7). Bunun bir sonucu olarak, 1970'li yıllarda, yapay sinir ağı çalışmalarına ayrılan fonlarda kesintiye gidildi ve araştırmacılar bu alanı terk etmeye başladı. Sadece az sayıda araştırmacı yapay sinir ağları üzerinde çalışmalarına devam etti (Kröse ve Smagt, 1996, s.13).

1980'li yıllarda, bilgisayarların hafıza ve işlemci kapasitelerinin artması gibi donanımsal gelişmeler, daha hızlı ve etkili optimizasyon algoritmalarının kavranması, biyolojik sinir sistemiyle ilgili yeni bulgular gibi sebeplerden dolayı yapay sinir ağlarına geri dönüş başladı. Rumelhart, Hinton ve Williams tarafından yazılan "Parallel Distributed Processing" (Paralel Dağıtık İşlemler) isimli kitap, çok katmanlı yapay sinir ağlarında, ağırlıkların optimizasyonunun mümkün olduğunu gösteren, "geri yayılım" (backpropagation) isimli bir algoritmayı gün ışığına çıkardı. Böylece Minsky ve Papert'in üzerinde durduğu XOR probleminden kaynaklanan öğrenme örüntüsü problemi çözülmüş oldu. Fizikçi John Hopfield, "Hopfield ağları" olarak bilinen, uyarlanabilir aktivasyonlar ve sabit parametrelere dayanan çok sayıda yapay sinir ağı geliştirdi. Kunihiro Fukushima, 1975 yılında önerdiği pozisyonu/rotasyonu bozuk karakterleri ayırt etmede başarısız olan "cognitron" isimli yapay sinir ağındaki eksikliği giderdi ve "neocognitron" isimli ağı geliştirdi. Çok sayıda araştırmacı, aktivasyonu ya da parametresi olasılık yoğunluk fonksiyonuna göre değişen (deterministik olmayan) yapay sinir ağlarını ifade eden "Boltzmann makinesi"ni (Boltzmann machine) araştırmaya başladı. Geri yayılım (backpropagation) algoritması, Hopfield ağları, neocognitron, Boltzmann makinesi gibi gelişmeler yapay sinir ağları üzerine araştırmaları motive ederek yeniden canlandırdı (Fausett, 1994, s.25-26; Silva ve diğerleri, 2017, s.7).

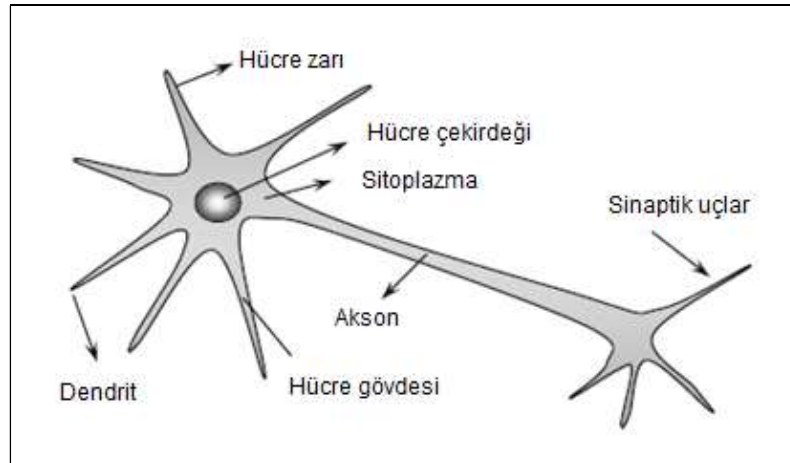
1.3. Yapay Sinir Ağlarının Genel Yapısı

Yapay sinir ağları, biyolojik sinir hücrelerinden esinlenerek geliştirildi. Temel olarak bir girdi ve çıktı katmanına ilave olarak, genellikle girdi ve çıktı

katmanları arasında bir veya daha fazla gizli katman içerirler. Girdi katmanından sunulan veriler, gizli katmanda yer alan nöronlar aracılığıyla çeşitli matematiksel ve istatistiksel yöntemlerle işlenerek çıktı katmanına iletilir. Çıktı katmanında bulunan nöron ya da nöronlar ise problemin çözümüne ilişkin bir değer döndürür. Takip eden alt başlıklarda, yapay nöronlar ve aktivasyon fonksiyonları gibi yapay sinir ağlarının temel bileşenlerinin yanı sıra, yapay sinir ağlarıyla yakından ilgili olan biyolojik sinir hücrelerinin temel işleyişleri de incelendi.

1.3.1. Biyolojik Sinir Hücreleri

Yapay sinir ağlarının en önemli karakteristiklerini açığa çıkarmak için biyolojik sinir hücrelerinin temel yapısını ve işleyişini incelemek yararlı olacaktır (Fausett, 1994, s.5). Düşünme ve öğrenme gibi insan beyni tarafından gerçekleştirilen bilgi işleme süreçleri, paralel çalışan biyolojik işlem bileşenleri tarafından yerine getirilir. Bu bileşenler temel olarak “dendrit”, hücre gövdesi olarak bilinen “soma” ve “akson”dan oluşur (Şekil 1.2).



Şekil 1.2. Biyolojik Sinir Hücresi

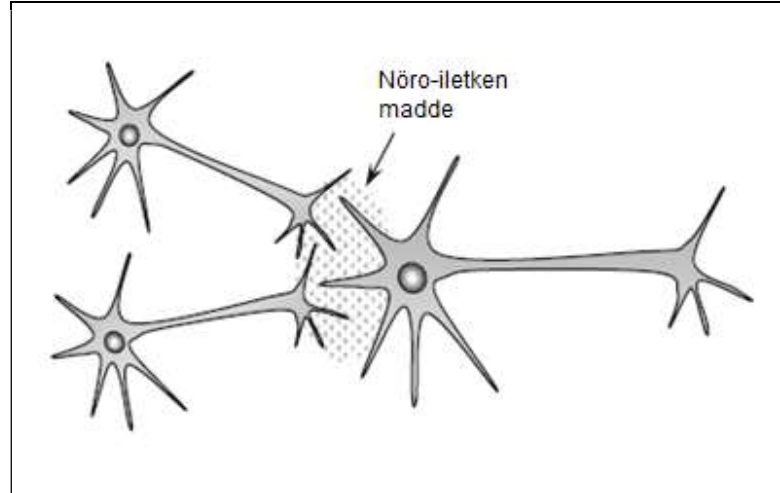
Kaynak: Silva ve diğerleri, 2017, s.9

Dendritler, ağaç dallarına benzeyen çok sayıda uzantıdan oluşur. Dendritlerin temel amacı, diğer bağlantı nöronlarından ya da dış çevreyle temas halinde olan sensör nöronlarından gelen uyarıları sürekli bir şekilde yakalamaktır.

Hücre gövdesi (soma), nöronun kendi aksonu boyunca bir elektrik uyarımı başlatıp başlatmayacağına işaret eden aktivasyon gerilimi üretmek için dendritlerden gelen tüm bilgilerin işlenmesinden sorumludur.

Akson, diğer bağlantı nöronlarına ya da kas dokusuna bağlı nöronlara elektrik sinyallerinin iletilmesini sağlayan tek bir uzantıdan oluşur. Aksonun bitiminde, sinaptik bağlantı uçları olarak adlandırılan saçaklar bulunur.

Dendritlerden gelen bilgilere göre hücre gövdesinin ürettiği aktivasyon gerilimi belirli bir eşiği aşarsa, söz konusu aşırı gerilim bir elektrik sinyali üreterek akson boyunca yayılır. Gerilimin üretildiği nöronun akson ucunda bulunan sinapslar ve sinyalin iletileceği nöron ya da nöronların dendritleri arasında bulunan bağlantı noktalarında herhangi bir fiziksel temas yoktur. Bu yüzden gerilimin üretildiği nörona ait sinapslar tarafından, sinyalin diğer nöron ya da nöronlara iletilmesinden sorumlu kimyasal bir nöro-iletken madde salgılanır (Şekil 1.3).

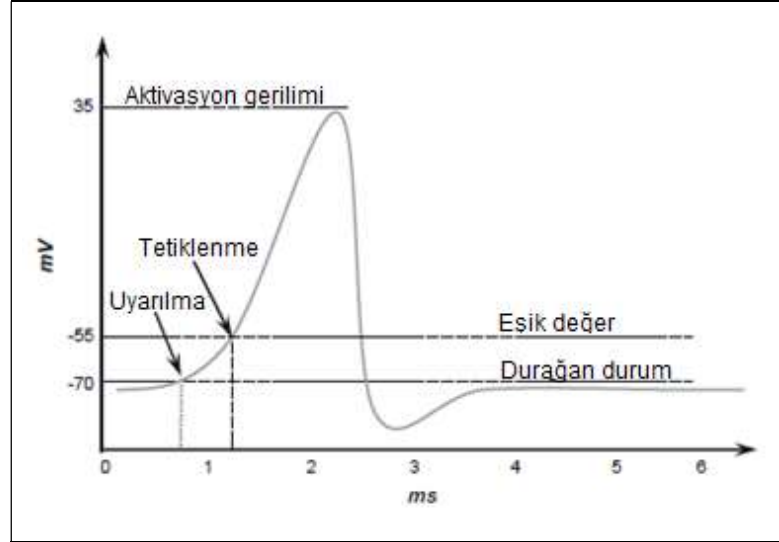


Şekil 1.3. Biyolojik Sinir Hücreleri Arasında Sinaptik Bağlantı

Kaynak: Silva ve diğerleri, 2017, s.10

İlk nöronun sinapslarında salgılanan nöro-iletken madde, diğer nöronun ya da nöronların dendritleri tarafından algılanarak hücre gövdesine/gövdelerine iletilir. Yukarıdaki süreç aynı şekilde diğer nöronlar arasında da yinelenir.

Aşağıda yer alan Grafik 1.1’de, biyolojik bir nöronun durağan durumu (resting potential) -70 mV (milivoltaj), alınan sinyalin diğer nöronlara iletilmesine ilişkin eşik değer (threshold potential) ise -55 mV olarak gösterilmiştir.



Grafik 1.1. Aktivasyon Geriliminin Aşamaları

Kaynak: Silva ve diğerleri, 2017, s.11

Herhangi bir nöronun uyarılması durumunda, maksimum değeri 35 mV olan aktivasyon gerilimi (activation potential) üretilir. Aktivasyon geriliminin süresi nöronlara göre değişmekle birlikte, maksimum değeri temsil eden 35 mV, uyarılan herhangi bir nöron için sabittir. Oluşan aktivasyon gerilimi, nöronun aksonları aracılığıyla diğer nöronlara aktarıldıktan hemen sonra, hücre zarı kimyasal reaksiyonlar sonucu repolarize olarak voltajı düşürür ve nöronun tekrar durağan duruma dönmesini sağlar (Silva ve diğerleri, 2017, s.9-11).

Biyolojik nöronlardan yapay sinir ağlarının nöronlarına aktarılan en önemli özellikler aşağıdaki gibi özetlenebilir.

- Bir nöron çok sayıda sinyal/girdi alır.
- Sinyaller/girdiler, nöron tarafından bir ağırlık ile değiştirilebilir.
- Nöron, ağırlıklandırılmış girdileri toplar.
- Ağırlıklandırılmış girdiler yeterli büyüklüğe ulaşırsa, nöron tek bir çıktı iletir.
- Bir nöronun çıktısı diğer nöronlara girdi olarak aktarılabilir.

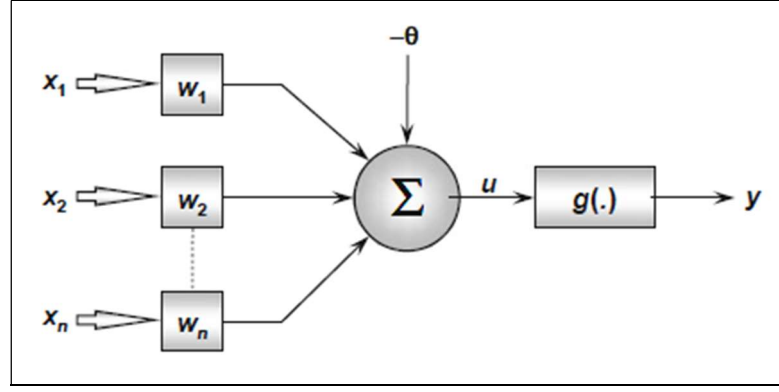
- Bilgi işleme yereldir, diğer bir ifadeyle nöronlar tarafından bağımsız olarak gerçekleştirilir.
- Bellek uzun ve kısa dönem olmak üzere dağıtılmıştır. Uzun süreli bellek, nöronların sinapslarında veya ağırlıklarında bulunur. Kısa süreli bellek ise nöronlar tarafından gönderilen sinyallere (gönderici nöron için çıktıya, alıcı nöronlar için girdiye) karşılık gelir.
- Bir sinapsın gücü deneyimle artabilir.
- Sinapslar için nöro-iletkenler uyarıcı (excitatory) veya durdurucu (inhibitory) olabilir.

“Hata toleransı” (fault tolerance), biyolojik sinir sisteminin yapay sinir ağıyla paylaştığı diğer bir önemli karakteristiktir. Biyolojik sinir sistemi üzerinden hata toleransı iki açıdan örneklendirilebilir. Birincisi, yıllar sonra karşılaşılan çocukluk arkadaşının tanınmasında olduğu gibi daha önce alınan sinyallerden bir ölçüye kadar farklı girdi sinyalleri ayırt edilebilir. İkincisi ise bir sinir hücresinde hasar meydana geldiğinde, diğer nöron ya da nöronların hasar görmüş sinir hücresinin işlevini üstlenmesidir (Fausett, 1994, s.6-7).

1.3.2. Yapay Nöronlar

Yapay sinir ağlarında işlem birimlerini ifade eden yapay nöronlar, biyolojik sinir hücrelerinin basitleştirilmiş modelleridir. Bu modeller sinir hücresinin elektrik sinyalleri üretmesine ve yaymasına ilişkin biyolojik çalışma ilkelerinden esinlenerek geliştirilmiştir (Silva ve diğerleri, 2017, s.11). Ancak yapay nöronların daima biyolojik nöronlarla birebir ilişkili olduğunu düşünmemek gerekir (Freeman ve Skapura, 1991, s.7).

Paralellik ve yüksek ilişkisellik/bağılantısallık gibi biyolojik bir sinir ağının temel özelliklerini içeren en basit yapay nöron modeli, McCulloch ve Pitts (1943) tarafından tasarlandı. Bu tasarım günümüzde de farklı yapay sinir ağı mimarilerinde en yaygın kullanılan yapay nöron modelidir.



Şekil 1.4. Yapay Nöron

Kaynak: Silva ve diğerleri, 2017, s.12

Bir ağdaki her bir nöron, Şekil 1.4'te gösterildiği gibi uygulanabilir. Biyolojik sinirlerin dendritleri tarafından toplanan elektrik uyarımlarına benzer bir şekilde, dış çevreden alınan (sinyal olarak da adlandırılan) girdiler, $x_1, x_2, x_3, \dots, x_n$ ile temsil edilir (Silva ve diğerleri, 2017, s.12). x_1, x_2 gibi ifadeler gözlem sırasını değil, birbirinden farklı girdileri (özellikleri, bağımsız değişkenleri) ifade eder.

Her bir x_i girdisi, kendi sinaptik ağırlığı w_i ile çarpılarak hesaplanır ve ağırlıklandırılmış girdilerden oluşan ilişkisel bilgi nörona ulaştırılır. Ağırlıklandırılmış girdiler “doğrusal birleştirici” (linear combiner) olarak da adlandırılan “operasyonel fonksiyon” tarafından toplanır (Haykin, 2009, s.10).

$$u = \left(\sum_{i=1}^n w_i x_i \right) - \theta \quad (1.1)$$

$x_0=1$ ve $w_0=-\theta$ olmak üzere, 1.1 no.lu denklem, i indisi 0'dan başlatılarak 1.2 no.lu denklemde gösterildiği şekilde de ifade edilebilir.

$$u = \sum_{i=0}^n w_i x_i \quad (1.2)$$

Girdilerin ağırlıklandırılmış toplamı ise çıktısı sınırlandırılan (sigmoid gibi) bir aktivasyon fonksiyonuna, 1.3 ve 1.4 no.lu denklemlerde gösterildiği gibi girdi olarak verilir (Haykin, 2009, s.10).

$$y=g(u) \quad (1.3)$$

$$y=g\left(\sum_{i=0}^n w_i x_i\right) \quad (1.4)$$

Yukarıdaki denklemler yapay nöronların temel bileşenlerini içermekte olup bunlara ilişkin temel kavramlar aşağıda açıklanmıştır.

- Girdiler (input signals): Girdi sinyali olarak da adlandırılan x_i ifadeleri dış çevreden ya da belirli bir uygulamadan elde edilen örneklem değerlerini temsil eder. Girdiler, öğrenme algoritmasının etkinliğini arttırmak amacıyla genellikle normalize edilirler.
- Sinaptik ağırlıklar (synaptic weights): w_i ifadeleri, her bir girdi değişkenini ağırlıklandırılmasında kullanılan değerlerdir. Amaç yapay nöronun işlevselliğine işaret eden değişkenlerin ilişkisini ölçmektir.
- Operasyonel fonksiyon (operational function, Σ): Birleştirme fonksiyonu olarak da bilinir. Bir aktivasyon gerilimi üretmek için ağırlıklandırılan girdi değerlerinin toplamını ifade eder. Ağırlıklandırılan girdi değerlerinin toplamı yerine, ortalaması, en büyüğü ya da en küçüğü gibi probleme göre farklı operasyonel fonksiyonlar da kullanılabilir.
- Aktivasyon eşiği (activation threshold, θ): Sabit (bias) olarak da ifade edilir. Uygun eşik değerini belirlemek için kullanılan bir değerdir. Aktivasyon gerilimini hesaplamak amacıyla kullanılır.
- Aktivasyon gerilimi (activation potential, u): Operasyonel fonksiyonun sonucu ile aktivasyon eşiği arasındaki farkı ifade eder. Nöronun çıktısı olan bu fark pozitif ($u \geq \theta$) ise nöron tarafından uyarıcı bir gerilim üretilir; fark negatif ($u \leq \theta$) ise uyarıcı bir gerilim üretilmez.
- Aktivasyon fonksiyonu (activation function, g): Nöronun çıktısını makul değerler arasında sınırlayan bir fonksiyondur. Farklı problemlerin çözümünde kullanılabilecek çok sayıda aktivasyon fonksiyonu çeşidi bulunur.

- Çıktı (output signal, y): Çıktı sinyali olarak da adlandırılan y ifadesi, belirli girdiler verilen nöronun aktivasyon fonksiyonu kullanarak ürettiği sonuç değeridir. Bu sonuç değeri, bir ağda sıralı olarak yer alan bağlantısız (aynı katmanda bulunan diğer) nöronlar tarafından girdi olarak kullanılabilir (Silva ve diğerleri, 2017, s.12).

1.3.3. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonu (activation function), aktivasyon gerilimini (u değerini) girdi olarak alarak regresyon, sınıflandırma, kümeleme gibi birbirinden farklı problemleri, doğasına uygun bir şekilde dönüştürür. Örneğin ikili bir sınıflandırma problemi için aktivasyon fonksiyonu 0 ve 1 şeklinde ikili (binary) bir değer üretebilir. Üretilen ikili değer, bir örneklemin belirli bir sınıfa ait olup olmadığına ilişkin soruya, aktivasyon fonksiyonun evet ya da hayır şeklinde verdiği bir yanıt olarak değerlendirilebilir.

Yapay nöronlarda, gerçek dünya probleminin türüne göre doğrusal ve doğrusal olmayan aktivasyon fonksiyonları kullanılır. Yapay nöronların doğrusal olmayan fonksiyonlar içerebilmesi, karmaşık problemlerin çözümünde kullanılabilmesine olanak tanır. Aktivasyon fonksiyonu, “transfer fonksiyonu” (transfer function) olarak da bilinir (Hagan ve diğerleri, 2014, s.37-38).

Adım (step), çift kutuplu adım (bipolar step), simetrik rampa (symmetric ramp) fonksiyonları türevlenemeyen, diğer bir ifadeyle eğimleri sabit aktivasyon fonksiyonlarına örnek verilebilir.

İşaret (signum), sınırlandırıcı (heaviside/hard limiter), eşik (threshold) gibi farklı isimlerle de adlandırılan adım fonksiyonu, eğer aktivasyon gerilimini ifade eden u değeri 0 değerine eşit veya ondan büyükse 1, değilse 0 değerini döndürür (Hagan ve diğerleri, 2014, s.38; Haykin, 2009, s.13; Mehrotra ve diğerleri, 1996, s.11-12). Aşağıda yer alan 1.5 no.lu denklemde örnek bir adım fonksiyonu yer almaktadır.

$$g(u)=\begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (1.5)$$

Simetrik sınırlandırıcı (symmetric hard limiter) olarak da adlandırılan çift kutuplu adım fonksiyonu, eğer aktivasyon gerilimi (u değeri) 0 değerinden büyükse 1, 0 değerine eşitse 0 ve 0 değerinden küçükse 1 değerini döndürür (Silva ve diğerleri, 2017, s.13-14). Aşağıdaki 1.6 no.lu denklemde örnek bir çift kutuplu adım fonksiyonu yer almaktadır.

$$g(u)=\begin{cases} 1, & u > 0 \\ 0, & u = 0 \\ -1, & u < 0 \end{cases} \quad (1.6)$$

Adım ve çift kutuplu adım fonksiyonlarında 0 sabiti yerine, θ gibi bir eşik değer de kullanılabilir (Fausett, 1994, s.41).

Simetrik rampa fonksiyonu, eğer aktivasyon gerilimi (u değeri) $[-a, a]$ arasında ise u, a değerinden büyükse a, küçükse -a değerini döndürür (Kröse ve Smagt, 1996, s.17). Aşağıda yer alan 1.7 no.lu denklem simetrik rampa fonksiyonuna örnek olarak verilmiştir.

$$g(u)=\begin{cases} a, & u > a \\ u, & -a \leq u \leq a \\ -a, & u < -a \end{cases} \quad (1.7)$$

Lojistik (logistic), hiperbolik tanjant (hyperbolic tangent), Gaussyan (Gaussian) ve doğrusal (linear) fonksiyonlar birinci dereceden türevlenebilen, diğer bir ifadeyle eğimleri değişebilen aktivasyon fonksiyonlarına örnek verilebilir.

Doğrusal ve doğrusal olmayan davranışlar arasında hassas bir dengeye sahip lojistik fonksiyon daima 0 ile 1 arasında bir değer döndürür. En yaygın kullanılan aktivasyon fonksiyonlarından (Hagan ve diğerleri, 2014, s.40; Haykin, 2009, s.14).

$$g(u) = \frac{1}{1+e^{-\beta u}} \quad (1.8)$$

1.8 no.lu denklemle temsil edilen lojistik fonksiyonda yer alan β ifadesi fonksiyonun büküm noktasındaki eğimini, e ifadesi ise Euler sabitini ifade eder.

Hiperbolik tanjant fonksiyonu, daima -1 ile 1 arasında bir değer döndürür.

$$g(u) = \frac{1-e^{-\beta u}}{1+e^{-\beta u}} \quad (1.9)$$

1.9 no.lu denklemle temsil edilen hiperbolik tanjant fonksiyonunda yer alan β ifadesi fonksiyonun büküm noktasındaki eğimini, e ifadesi ise Euler sabitini temsil eder. Öğrenme algoritmalarının yakınsaması (ağırlık matrisinin hesaplanması) açısından hiperbolik tanjant fonksiyonu genellikle lojistik fonksiyondan daha hızlıdır (Bishop, 1995, s.127).

Aşağıdaki 1.10 no.lu denklemle ifade edilen doğrusal fonksiyon (ya da birim fonksiyon), aktivasyon gerilimi u değerine eşit sonuç üretir (Hagan ve diğerleri, 2014, s.39).

$$g(u) = u \quad (1.10)$$

Doğrusal aktivasyon fonksiyonları, özellikle fonksiyon yaklaştırma işlemlerini gerçekleştiren yapay sinir ağlarında kullanılır. Fonksiyon yaklaştırma (function approximation) ya da evrensel eğri tahmini (universal curve fitting), belirli bir sürecin girdi/çıkı değişkenlerinin davranışlarını çıkartmayı ifade eder (Silva ve diğerleri, 2017, s.18).

Derin öğrenme uygulamaları için en yaygın kullanılan aktivasyon fonksiyonu ReLU (rectified linear unit), Nair ve Hinton (2010) tarafından önerilmiştir. ReLU, hızlı öğrenen ve başarılı bir aktivasyon fonksiyonudur. Sigmoid ve tanjant aktivasyon fonksiyonlarına kıyasla derin öğrenmede daha iyi performans ve genelleme kabiliyetine sahiptir. ReLU, neredeyse doğrusal bir işlevi temsil ettiği için gradyan yöntemlerle optimize edilmelerini kolaylaştıran doğrusal modellerin özelliklerini korur (Nwankpa ve diğerleri, 2018).

$$g(u) = \max(0, u) = \begin{cases} u, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (1.11)$$

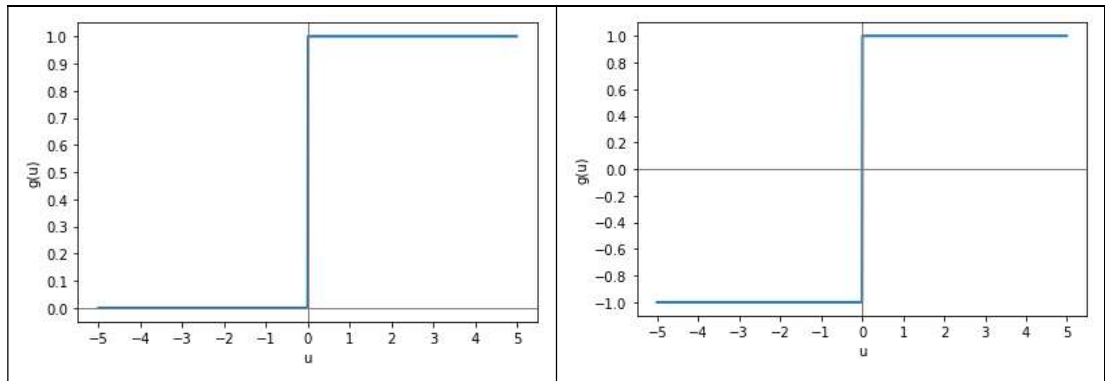
1.11 no.lu denklem, sıfırdan küçük girişlerin değerlerini düzelterek onları sıfır olarak kabul eder. Sıfıra eşit ya da sıfırdan büyük değerleri ise girdi değerine eşitler.

SELU (scaled exponential linear unit), Klambauer ve diğerleri (2017) tarafından önerilmiştir. SELU, kendi kendini normalleştiren (self-normalising) bir sinir ağı olarak tanıtıldı. Ortalama ve varyans, çoklu katmanlar aracılığıyla yayılarak sıfır ortalama ve birim varyansa yakınsayan bir değere sahiptir. Bu nedenle ortalama ve varyansı derin öğrenme uygulaması için uygun hale getirir ve güçlü düzenleme ile özellikleri hızlı ve etkili bir şekilde öğrenir (Nwankpa ve diğerleri, 2018).

$$g(u) = \max(0, u) = \tau \begin{pmatrix} u, & u > 0 \\ \alpha \exp(x) - \alpha & u \leq 0 \end{pmatrix} \quad (1.12)$$

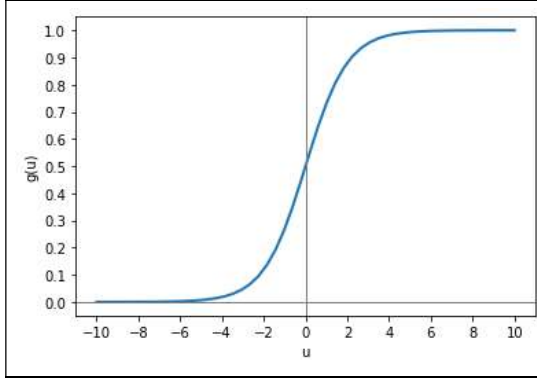
1.12 no.lu denklemde yer alan τ ölçekleme faktörünü, α ise negatif ağ girdilerini genellikle 1'e eşitlemek için doygunluk noktalarını kontrol eden ve yaklaşık değeri 1,6733 olan bir hiper parametredir.

Aşağıda yer alan Grafik 1.2.a, b, c, d, e, f, g, h'de ise bazı aktivasyon fonksiyonları Python ile görselleştirilmiştir.

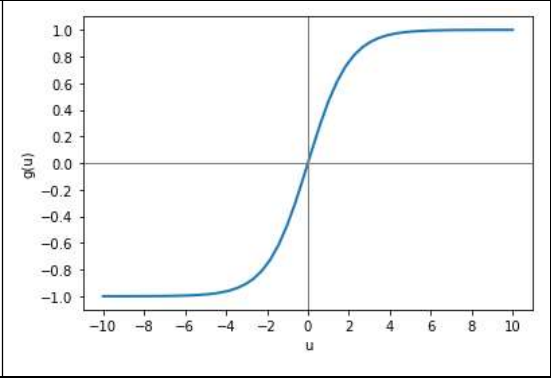


Grafik 1.2.a. Adım Fonksiyonu

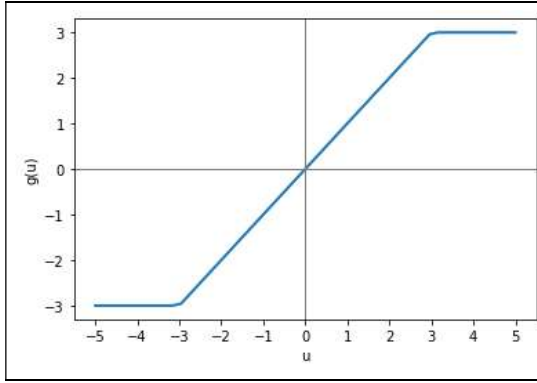
Grafik 1.2.b. Çift Kutuplu Adım Fonksiyonu



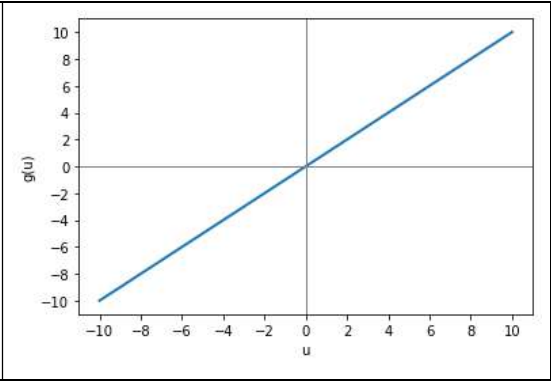
Grafik 1.2.c. Lojistik Fonksiyon



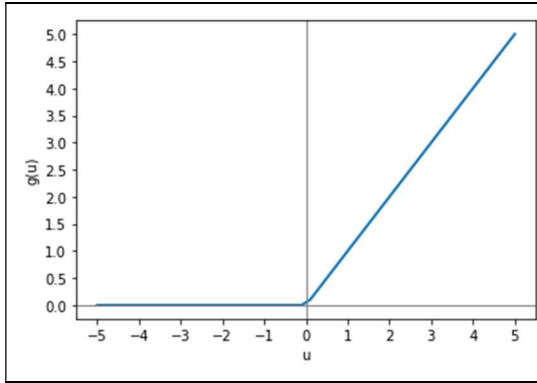
Grafik 1.2.d. Hiperbolik Tanjant Fonksiyonu



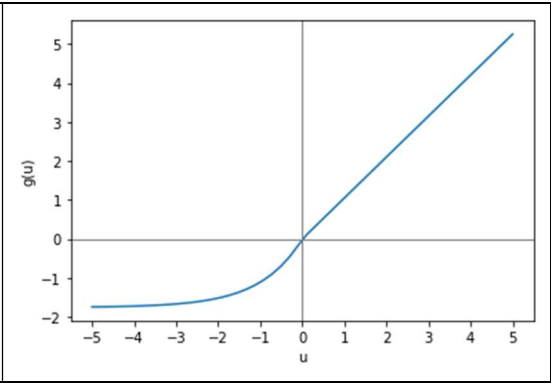
Grafik 1.2.e. Simetrik Rampa Fonksiyonu



Grafik 1.2.f. Doğrusal (Birim) Fonksiyon



Grafik 1.2.g. ReLU Fonksiyonu



Grafik 1.2.h. SELU Fonksiyonu

Grafik 1.2.a ve Grafik 1.2.c kıyaslandığında lojistik fonksiyon, β değeri sonsuza giderken adım fonksiyonuna yakınsar (Bishop, 1995, s.128). Ancak adım fonksiyonun aksine, lojistik fonksiyon tanımlı olduğu aralıkta türevlenebilir (Silva ve diğerleri, 2017, s.15).

Grafik 1.2.b ve Grafik 1.2.d kıyaslandığında hiperbolik tanjant fonksiyonu, β değeri sonsuza giderken çift kutuplu adım fonksiyonuna yakınsar. Ancak çift kutuplu adım fonksiyonun aksine, hiperbolik tanjant

fonksiyonu tanımlı olduğu aralıkta türevlenebilir. Lojistik ve hiperbolik tanjant fonksiyonu, “sigmoid fonksiyonlar” olarak da bilinir (Silva ve diğerleri, 2017, s.16-17).

Grafik 1.2.e ve Grafik 1.2.f’de ise simetrik rampa fonksiyonu ve doğrusal (birim) fonksiyon gösterilmektedir. Simetrik rampa fonksiyonu, birim fonksiyonu belirli değer aralıkları ile sınırlandıran bir fonksiyondur.

Grafik 1.2.g ve Grafik 1.2.h’de ise ReLU ve SELU fonksiyonları gösterilmektedir. ReLU, simetrik rampa fonksiyonun alabileceği değerleri tek yönde kısıtlayan doğrusal bir fonksiyondur. SELU ise bu kısıtlamayı doğrusal değil üssel bir şekilde gerçekleştirmektedir.

1.4. Yapay Sinir Ağı Mimarileri

Yapay sinir ağı “mimarisi”, nöronların sinaptik bağlantılarının birbirlerine yönlendirilerek aralarındaki ilişkilerin yapılandırılması olarak tanımlanabilir. Örneğin ağın katman sayısı, aynı ve/veya farklı katmanlarda bulunan nöronların birbirleriyle bağlantılı olup olmadıkları, ağın girdileri alış ve işleme şekli gibi yapısal özellikler ağın mimarisine işaret eder. Yapay sinir ağı “topolojisi” ise belirli bir mimari içinde ağın bürünebileceği farklı yapısal kompozisyonlar olarak ifade edilebilir. Diğer bir ifadeyle, aynı mimariye sahip iki ağdan biri 6, diğeri 12 nörondan oluşan farklı ağ topolojilerine sahip olabilir. Benzer şekilde aynı mimariye sahip iki ağdan birinin nöronları lojistik aktivasyon fonksiyonundan, diğeri nöronları ise hiperbolik tanjant aktivasyon fonksiyonundan oluşabilir. (Silva ve diğerleri, 2017, s.21; Mehrotra ve diğerleri, 1996, s.16-21).

Yapay sinir ağı mimarisi, ağın eğitiminde kullanılan öğrenme algoritmalarıyla çok yakından bağlantılıdır (Haykin, 2009, s.21). Yapay sinir ağı mimarisinin eğitilmesi, diğer bir ifadeyle nöronların ağırlıklarının ve eşik değerlerinin ayarlanması, ağın çıktılarını arzulanan değerlere yakın olacak şekilde tahmin etmeye çalışan “öğrenme algoritması” tarafından yerine getirilir (Silva ve diğerleri, 2017, s.21).

Bir yapay sinir ağı temel olarak girdi, gizli ve çıktı katmanları olmak üzere üç kısımdan oluşur.

Girdi katmanı (input layer), dış çevreden ya da belirli bir uygulamadan elde edilen, ağı eğitileceği örneklem değerlerini ve değişkenlerini (x_i ifadelerini) içerir. Bu girdiler, genellikle aktivasyon fonksiyonları tarafından üretilen limit değerler dahilinde normalize edilirler. Normalizasyon işlemi, ağı tarafından gerçekleştirilen matematiksel işlemler için daha iyi sayısal kesinlik sağlar. Girdi katmanında normalizasyon gibi veri ön işleme dışında herhangi bir işlem yapılmaz. Bu sebepten dolayı bazı kaynaklarda girdi katmanı bir katman olarak değerlendirilmez. Bu çalışmada da girdi katmanı (aksi belirtilmedikçe) bir katman olarak değerlendirilmemiştir.

Gizli katman ya da katmanlar (hidden layers), analiz edilen verilerle ilgili örüntüleri ortaya çıkarmaktan, diğer bir ifadeyle öğrenmeyi gerçekleştirmekten sorumlu olan nöronlardan oluşur.

Çıktı katmanı (output layer), bir veya daha fazla nörondan oluşabilir. Gizli katmanlarda yer alan nöronlar tarafından ortaya çıkartılan örüntülerin, nihai çıktılara dönüştürülmesinden ve sunulmasından sorumludur. Aynı bir gizli katmanın bulunmadığı ağlarda, gizli ve çıktı katmanları tek bir katmanda birleşerek aynı katmanda (çıktı katmanında) yer alabilir.

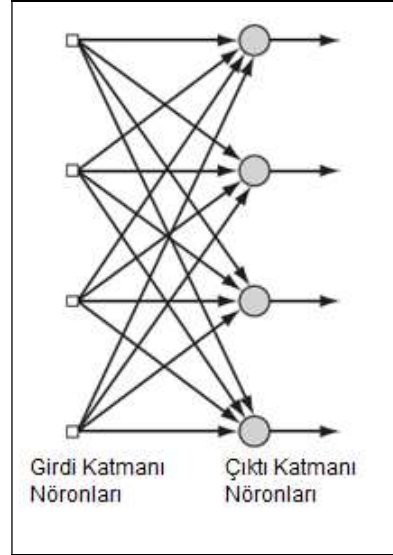
Yapay sinir ağı mimarileri, temel olarak üç kategoride sınıflandırılabilir.

- Tek katmanlı ileri beslemeli ağlar (single-layer feed-forward networks),
- Çok katmanlı ileri beslemeli ağlar (multi-layer feed-forward networks),
- Geri beslemeli ya da yinelemeli ağlar (back-forward or recurrent networks) (Haykin, 2009, s.21).

Takip eden alt başlıklarda söz konusu üç yapay sinir ağı mimarisi ana hatlarıyla tanıtılmıştır.

1.4.1. Tek Katmanlı İleri Beslemeli Ağlar

Bu yapay sinir ağı, sadece bir girdi katmanı ile nöronların yer aldığı çıktı katmanına sahip bir mimaridir. Girdi katmanında herhangi bir hesaplama yapılmadığı için “tek katmanlı” ağ olarak adlandırılır (Haykin, 2009, s.21). Şekil 1.5’te tek katmanlı ileri beslemeli bir yapay sinir ağı mimarisi gösterilmektedir.



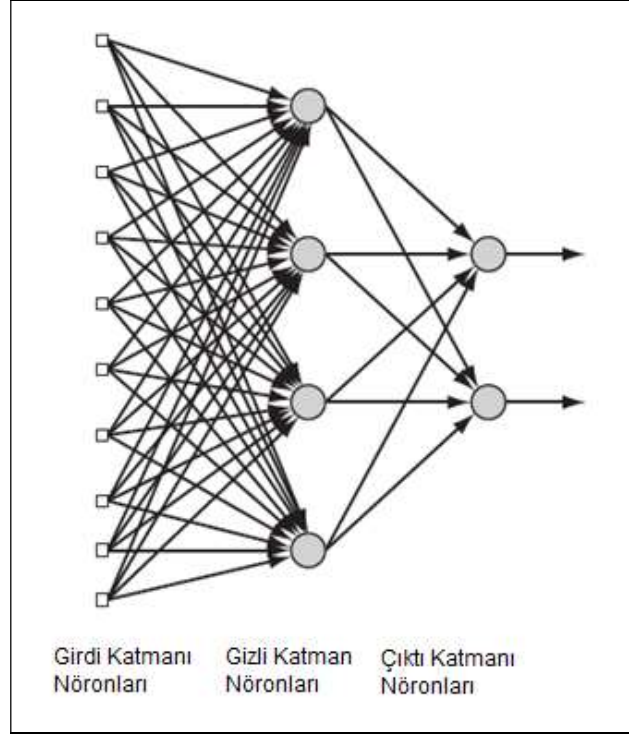
Şekil 1.5. Tek Katmanlı İleri Beslemeli Mimari

Kaynak: Haykin, 2009, s.21

Bilgi akışı girdi katmanından çıktı katmanına doğru tek bir yönde gerçekleşir. Girdi katmanındaki her bir eleman çıktı katmanındaki her bir nöronla doğrudan bağlantılıdır. Bu tür mimariye sahip ağlarda ağın nöron ve çıktı sayıları örtüşür. Tek katmanlı ileri beslemeli ağlar sınıflandırma ve doğrusal filtreleme problemlerinin çözümünde sıklıkla kullanılır (Silva ve diğerleri, 2017, s.22).

1.4.2. Çok Katmanlı İleri Beslemeli Ağlar

Bu yapay sinir ağı ise girdi ve çıktı katmanlarına ilave olarak, ağırlıkları hesaplayan nöronları içeren bir veya daha fazla gizli katmandan oluşur (Silva ve diğerleri, 2017, s.22). Şekil 1.6’da girdi ve çıktı katmanı ile bunların arasında yer alan bir gizli katmandan oluşan çok katmanlı ileri beslemeli bir yapay sinir ağı mimarisi gösterilmektedir.



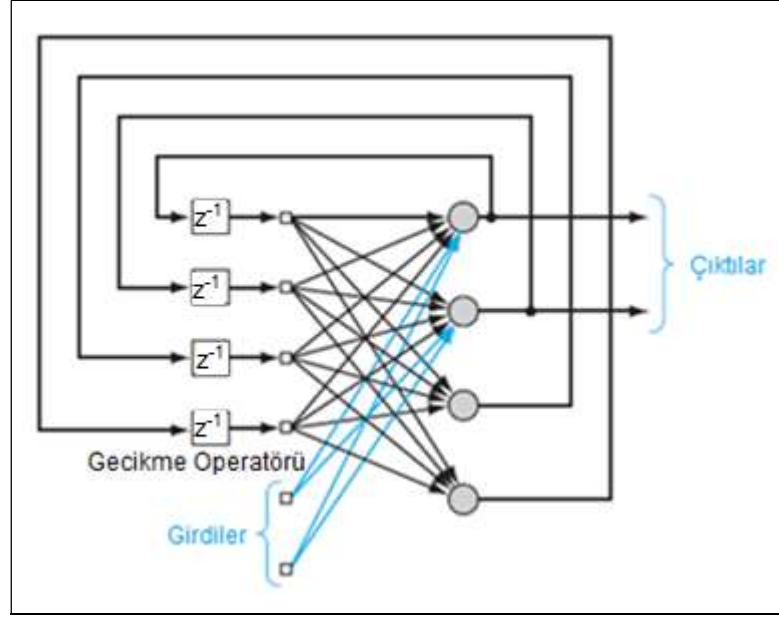
Şekil 1.6. Çok Katmanlı İleri Beslemeli Mimari

Kaynak: Haykin, 2009, s.22

Girdi katmanındaki her bir sinyal/özellik birinci (gizli) katmanın nöronlarının girdisidir. Birinci katmanın nöronlarının çıktısı ise çıktı katmanının nöronlarının (ya da nöronun) girdisidir. Her bir katman, kendinden önceki katmanın çıktısını girdi olarak alır. Katmanlarda bulunan nöronların her biri, (girdi katmanından çıktı katmanına doğru) kendisinden sonraki katmanda bulunan nöronların her biriyle bağlantılıdır. Girdiler, girdi katmanından çıktı katmanına doğru nöronlar tarafından işlenerek ilerler (Haykin, 2009, s.22-23). Ancak aynı katmanda bulunan nöronların birbiriyle herhangi bir bağlantısı bulunmamaktadır.

1.4.3. Geri Beslemeli ya da Yinelemeli Ağlar

Yinelemeli ağların (recurrent networks), ileri beslemeli ağlardan farkı, en az bir “geri besleme” (feedback) döngüsüne sahip olmalarıdır. Tek katmanlı ağlarda, her bir nöronun çıktısı, geriye doğru (kendisi dışındaki) tüm nöronlara girdi olarak verilir. Şekil 1.7’de tek gizli katmana sahip çok katmanlı bir yapay sinir ağında geri besleme süreci gösterilmektedir.



Şekil 1.7. Geri Beslemeli Mimari

Kaynak: Haykin, 2009, s.24

Ağdaki geri besleme döngüleri, öğrenme kapasitesi ve performansı üzerinde büyük bir etkiye sahiptir. Ayrıca ağın doğrusal olmayan nöronlar içerdiği varsayımında, geri besleme döngüleri, doğrusal olmayan ve (z^{-1} , z^{-2} gibi) zaman gecikmelerinden oluşan dinamik davranışlara sahiptir (Haykin, 2009, s.23).

1.5. Öğrenme Yöntemleri

Bir yapay sinir ağının eğitim süreci, nöronların çıktıları tarafından üretilen çözümleri genelleştirmek için sinaptik ağırlıkları ve eşik değerleri optimize eden ve “öğrenme algoritmaları” olarak adlandırılan bir dizi adımdan oluşur (Silva ve diğerleri, 2017, s.25). Sinaptik ağırlıkların ve eşik değerlerin hesaplanmasına göre eğitim yöntemleri, literatürde genellikle denetimli (supervised), denetimsiz (unsupervised) ve denetimli öğrenmenin özel bir türü olan pekiştirmeli (reinforcement) öğrenme olarak kategorize edilir.

1.5.1. Denetimli Öğrenme

Denetimli öğrenmede (supervised learning/training), her bir örneklemin girdisine karşılık beklenen (gözlemlenen) birer çıktı değeri bulunur. Bu yüzden denetimli öğrenme girdi/çıktı verilerine sahip “özellik/değer”

(attribute/value) matrisi kullanır. Sinaptik ağırlıkların ve eşik değerlerin hesaplanmasında, “eğitim seti” (training set) olarak adlandırılan bir veri seti ve bir öğrenme algoritması kullanılarak üretilen çıktılar ile beklenen çıktılar karşılaştırılır. Böylece elde edilen farkların/hataların tekrarlayan bir prosedürle optimize edilmesi amaçlanır. Yineleme prosedürü yardımıyla farkların ya da hataların kabul edilebilir bir seviyeye indirgenmesiyle eğitim süreci tamamlanmış olur. Eğitim setinde öğrenmeyi gerçekleştirmeye yetecek sayıda veri bulunmuyorsa, ağırlıklar ve eşik değerleri yakınsama (öğrenme) sürecini yerine getiremez. Benzer şekilde ağırlıklar çok sayıda katman ve nöron içermesi de “ezberleme” (memorization/overfitting) soruna yol açarak öğrenme sürecini olumsuz etkiler. (Silva ve diğerleri, 2017, s.26; Anderson ve McNeill, 1992, s.10-11). Her bir beklenen çıktıya karşılık gelen bir girdiyle eğitilen denetimli öğrenme ağırları “ilişkisel hafıza”ya (associative memory) sahiptir. Eğer beklenen çıktı ile girdi aynı ise “oto ilişkisel hafıza” (autoassociative memory), beklenen çıktı ile girdi farklı ise “ayrık ilişkisel hafıza” (heteroassociative memory) olarak adlandırılır (Fausett, 1994, s.15-16).

1.5.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede (unsupervised learning/training), denetimli öğrenmede olduğu gibi girdi değerlerinin yanında beklenen çıktı değerleri kullanılmaz. Bu öğrenme yaklaşımında yapay sinir ağı, veri setinin birbirine en çok benzeyen girdilerini aynı çıktı kümelerine atamak amacıyla sinaptik ağırlıkları ve eşik değerleri optimize eder (Fausett, 1994, s.16). Denetimsiz öğrenmede bir ağ, girdi setinin istatistiksel düzenliliğini ayırt eder hale geldiğinde, girdi setinin ayırt edici özelliklerini saptamak için iç temsillerini oluşturma yeteneğini edinir (Haykin, 2009, s.37). Diğer bir ifadeyle, denetimsiz öğrenme yöntemiyle eğitilen bir ağ, daha önce karşılaşmadığı bir örneğin karakteristiklerini, öğrenme sürecinde oluşturduğu örüntüleri (iç temsilleri) kullanarak ayırt edebilir.

1.5.3. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme (reinforcement learning/training), denetimli öğrenme tekniğinin bir türü olarak düşünülebilir. Çünkü ağırlıkların ürettiği çıktı ile

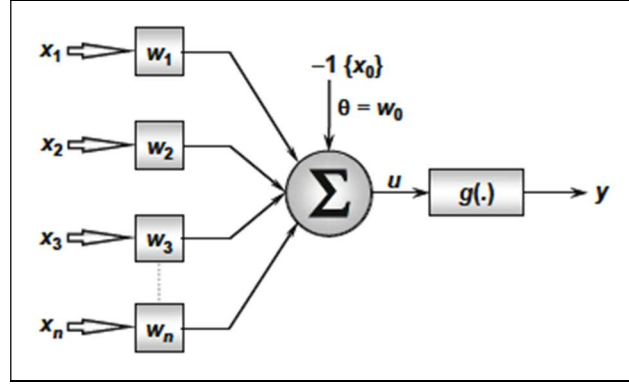
arzulanan çıktı arasındaki fark ağı tarafından sürekli olarak analiz edilir. Ancak denetimli öğrenmede olduğu gibi ağa beklenen çıktı değerlerinin tümü “doğrudan” verilmez. Pekiştirmeli öğrenme sürecinde, dış çevreyle sürekli etkileşim halinde elde edilen veriler, hataları minimize etmek amacıyla kullanılır. Ağı ürettiği çıktıların arzulanan çıktılara yaklaşıp yaklaşmamasına göre ağırlıklar, ödül-ceza yöntemi esas alınarak (ağı tarafından) aşamalı bir şekilde artırılır ya da azaltılır. Deneme-yanılma (heuristic) yöntemine dayanan pekiştirmeli öğrenme tekniğinde kullanılan öğrenme algoritmaları olasılıksal süreçler içeren stokastik karakteristiğe sahiptir (Silva ve diğerleri, 2017, s.26-27; Haykin, 2009, s.36).

1.6. Tek Katmanlı İleri Beslemeli Ağlar

Tek katmanlı ileri beslemeli ağlar, yapay sinir ağları alanında ilk geliştirilen ağlardır. Bunlardan en yaygın olanları tek katmanlı algılayıcı ve ADALINE olarak bilinen adaptif doğrusal elemandır. Tek katmanlı ağlar, sadece doğrusal problemlerin çözümünde kullanılabilir; doğrusal olmayan problemlerin çözümünde kullanılamaz.

1.6.1. Tek Katmanlı Algılayıcı ve Hebb Öğrenme Algoritması

Basit şekilleri sınıflandırabilen ve tek nörondan oluşan ilk “tek katmanlı algılayıcı” (single-layer perceptron), Frank Rosenblatt (1958) tarafından geliştirilmiştir. Geliştirilen en basit yapay sinir ağı olan tek katmanlı algılayıcı, girdi ve çıktı katmanına sahiptir (Bishop, 1995, s.98). 1960’lı ve 1970’li yıllarda birçok araştırmacıya ilham veren, denetimli öğrenmenin ilk modeli olan Rosenblatt’ın algılayıcısı, 1958 yılından günümüze geçerliliğini korumaktadır (Haykin, 2009, s.47). Şekil 1.8’de gösterilen tek katmanlı algılayıcının girdilerini (bağımsız değişkenlerini) x_i , girdi sayısını n , sinaptik ağırlıklarını w_i , aktivasyon eşiğini θ ve çıktısını (bağımlı değişkenini) y temsil etmektedir.



Şekil 1.8. Tek Nöronlu ve Tek Katmanlı Algılayıcı

Kaynak: Silva ve diğerleri, 2017, s.30

n ifadesi girdi/özellik sayısı (gözlem sayısı değil), u aktivasyon gerilimi, $x_0=1$ ve $w_0=\theta$ olmak üzere, tek nöronlu ve tek katmanlı ağın ağırlıklandırılmış girdilerini toplayan operasyonel fonksiyonu $f(u)$ ve nöronun aktivasyon fonksiyonu $g(u)$ aşağıda yer alan 1.13 ve 1.14 no.lu denklemlerde gösterilmektedir.

$$u=f(u)+\theta=\sum_{i=0}^n w_i x_i \quad (1.13)$$

$$y=g(u) \quad (1.14)$$

$\mathbf{w}$ ve $\mathbf{x}$ birer vektör olmak üzere, 1.13 no.lu denklem, aşağıda yer alan 1.15 no.lu denklemde gösterildiği gibi vektörel formda ifade edilebilir.

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}, \mathbf{w}' = [w_0 \quad w_1 \quad \dots \quad w_2 \quad w_n] \text{ ve } \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (1.15)$$

$\mathbf{w}'$ ifadesi $\mathbf{w}$ vektörünün devriğini (transpose) temsil etmektedir. $\mathbf{w}'$ vektörünün, 1.16 ve 1.17 no.lu denklemlerde gösterildiği gibi sağdan $\mathbf{x}$ vektörü ile çarpımı u skalerine (scaler) eşit olacaktır (Haykin, 2009, s.50).

$$u = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}^T \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [w_0 \quad w_1 \quad \dots \quad w_2 \quad w_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (1.16)$$

$$u = \mathbf{w}'\mathbf{x} \quad (1.17)$$

t örneklem sayısı ve k örneklem sırası olmak üzere, 1.16 (ya da 1.17) no.lu vektörel ifadenin skaler sonucu u, k. örneklem için aktivasyon gerilimini verir. Aktivasyon gerilimi ise aktivasyon fonksiyonunun girdisi olacaktır. Diğer taraftan ağırlık eğitimi sonucu $\mathbf{w}$ ağırlık vektörü hesaplandıktan (öğrenme süreci tamamlandıktan) sonra, aktivasyon gerilimi, herhangi bir $\mathbf{x}$ vektörünün (özellik kümesinin) sınıfını belirler (Hagan ve diğerleri, 2014, s.84). Bir başka ifadeyle, $\mathbf{w}$ ağırlık vektörü hesaplandığında, doğrusal olarak belirlenebilen sınıfları, birbirinden ayıran “karar sınırı”nın (decision boundary) n boyutlu örneklem uzayındaki konumunu belirlemede aktivasyon gerilimi kullanılır.

Yukarıdaki 1.16 (ya da 1.17) no.lu denklem sistemine, girdilerin tüm gözlem değerlerini ilave etmek için 1.18 no.lu denklemde gösterildiği gibi t adet $\mathbf{x}$ vektöründen oluşan $\mathbf{X}$ matrisi kullanılır.

$$\mathbf{X} = \begin{bmatrix} x_{1,0} & x_{1,1} & x_{1,2} & \dots & x_{1,t} \\ x_{2,0} & x_{2,1} & x_{2,2} & \dots & x_{2,t} \\ x_{3,0} & x_{3,1} & x_{3,2} & \dots & x_{3,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,0} & x_{n,1} & x_{n,2} & \dots & x_{n,t} \end{bmatrix} \quad (1.18)$$

$x_{i,k}$ ifadesi, i. girdinin (sinyalin, bağımsız değişkenin ya da özelliğin) k. gözlem değeri olmak üzere, $\mathbf{X}$ girdi matrisinin her bir satırı, girdi katmanında yer alan her bir girdinin (1 sabiti dahil) gözlem değerlerini ifade eder. O halde (tüm girdiler için) her bir örnekleme göre aktivasyon gerilimini elde etmek için $[(n+1)*1]$ boyutlu $\mathbf{w}'$ vektörü, sağdan $[(n+1)*t]$ boyutlu $\mathbf{X}$ matrisi ile çarpılır ve $[1*t]$ boyutlu bir $\mathbf{u}$ vektörü 1.19 ve 1.20 no.lu denklemlerde gösterildiği gibi elde edilir.

$$\mathbf{u} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}' \begin{bmatrix} x_{1,0} & x_{1,1} & x_{1,2} & \dots & x_{1,t} \\ x_{2,0} & x_{2,1} & x_{2,2} & \dots & x_{2,t} \\ x_{3,0} & x_{3,1} & x_{3,2} & \dots & x_{3,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,0} & x_{n,1} & x_{n,2} & \dots & x_{n,t} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_t \end{bmatrix} \quad (1.19)$$

$$\mathbf{u} = \mathbf{w}'\mathbf{X} \quad (1.20)$$

Yukarıdaki 1.19 (ya da 1.20) no.lu denklem sistemine göre bir nöron toplamda, örneklem sayısı kadar (t defa) aktivasyon gerilimi üreteceğinden, u artık bir skaler değil, t elemanlı vektörel bir büyüklüktür. Diğer taraftan **u** vektörü [1*t] boyutlu bir satır vektörü olmasına rağmen, yanında herhangi bir matris/vektör çarpanı bulunmadığından, [t*1] boyutlu bir sütun vektörü olarak ifade edilmiştir.

Aktivasyon fonksiyonu g(u) adım, çift kutuplu adım, lojistik fonksiyon gibi problemin türüne göre herhangi bir fonksiyon olabilir. Tek nöronlu algılayıcının çıktısı y, iki sınıftan biriyle ilgili olduğundan sadece iki değer alabilir. Bu yüzden algılayıcı, genellikle doğrusal ikili sınıflandırma (binary classification) problemlerinin çözümünde kullanılır. Bununla birlikte sınıfların doğrusal olarak ayrılabilmesi koşuluyla, birden çok nöron içeren bir algılayıcı ikiden daha fazla sınıfı ayrıştırılabilir (Haykin, 2009, s.48).

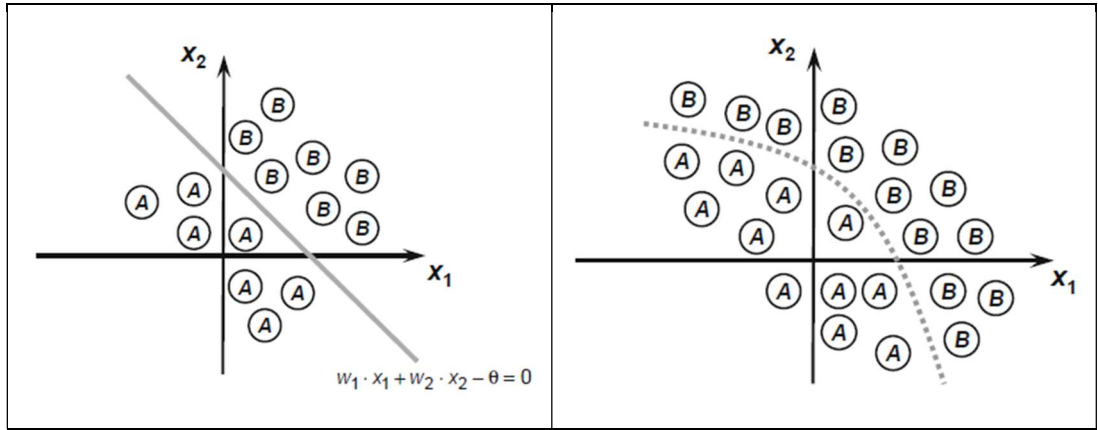
A ve B gibi iki farklı sınıfa ilişkin bir veri seti olduğu ve aktivasyon fonksiyonu olarak çift kutuplu adım fonksiyonunun kullanıldığı varsayımında, tek nöronlu algılayıcının çıktısı 1 ise A sınıfını, -1 ise B sınıfını ifade eder. Çift kutuplu adım fonksiyonunun matematikse formu 1.21 no.lu denklemde gösterilmektedir.

$$y=g(u)=\begin{cases} 1, & u \geq 0 \\ -1, & u < 0 \end{cases} \quad (1.21)$$

İki boyutlu uzayda “karar sınırı” (decision boundary) görselleştirmek amacıyla girdi sayısının n=2 olduğu varsayılmıştır. Bu çerçevede A ve B sınıflarını iki girdiyle doğrusal olarak birbirinden ayıran karar sınırı 1.22 no.lu denklemle gösterilebilir.

$$w_1x_1 + w_2x_2 + \theta = 0 \quad (1.22)$$

1.22 no.lu denklemin $n=2$ iken k . örneklem için $\mathbf{wx}'$ ifadesine eşit olduğu fark edilebilir. Ağırlık vektörünün (örneğe göre w_1 ve w_2) değerleri hesaplandıktan sonra, $x_1=0$ iken denklemin çözümü karar sınırının x_2 eksenini kestiği noktaya, $x_2=0$ iken denklemin çözümü karar sınırının x_1 eksenini kestiği noktaya işaret eder (Hagan ve diğerleri, 2014, s.84-85). Eşik değer θ ise karar sınırının orijinden farklı bir yerde konumlanmasını sağlar (Haykin, 2009, s.49).



Grafik 1.3.a. Doğrusal Sınıflandırma

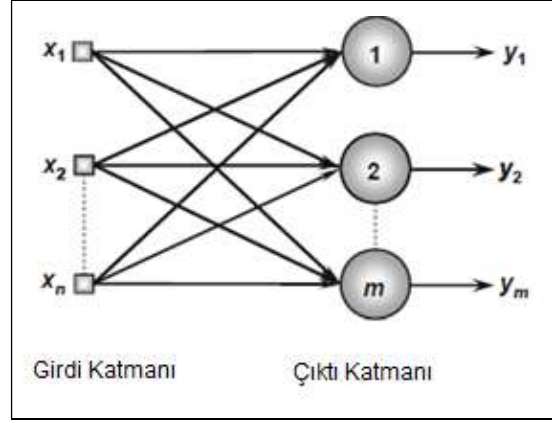
Grafik 1.3.b. Doğrusal Olmayan Sınıflandırma

Kaynak: Silva ve diğerleri, 2017, s.32

Kaynak: Silva ve diğerleri, 2017, s.33

Grafik 1.3.a'da algılayıcı iki farklı sınıfı doğrusal olarak birbirinden ayırmaktadır. Aktivasyon fonksiyonunun çıktısı 1 ise A sınıfını doğrunun altında, -1 ise B sınıfını doğrunun üstünde konumlandırır. Grafik 1.3.b'de ise algılayıcının doğrusal olmayan bir sınıflandırma problemini etkin bir şekilde çözemeyeceği gösterilmektedir.

Tek katmanlı algılayıcı birden fazla nöron da içerebilir. Şekil 1.9'da çok sayıda nörondan oluşan tek katmanlı algılayıcı gösterilmektedir.



Şekil 1.9. Çok Nöronlu ve Tek Katmanlı Algılayıcı

Kaynak: Silva ve diğerleri, 2017, s.22

m nöron sayısı olmak üzere, her bir girdinin her bir nöronla eşleştiği çok nöronlu ve tek katmanlı bir topolojide, girdi sayısı ile nöron sayısının çarpımı, eşik değer dahil $[m \cdot (n+1)]$ kadar ağırlık bulunur. $\mathbf{W}$ ağırlık matrisi olmak üzere, bu durumda ağırlıklar, 1.23 no.lu denklemde gösterildiği gibi ağırlık vektörü yerine ağırlık matrisi ile ifade edilir (Hagan ve diğerleri, 2014, s.83).

$$\mathbf{W} = \begin{bmatrix} w_{1,0} & w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,0} & w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ w_{3,0} & w_{3,1} & w_{3,2} & \dots & w_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{m,0} & w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} \quad (1.23)$$

$w_{j,i}$ ifadesi, j. nöronu i. girdiye bağlayan ağırlık değeri (ya da j. nöronun i. ağırlık değeri) olmak üzere, $\mathbf{W}$ ağırlık matrisinin her bir satırı, çıktı katmanında yer alan her bir nöronun (eşik değerleri dahil) girdi ağırlıklarını ifade eder. Bu çerçevede tek nöronlu ve tek katmanlı topoloji ile çok nöronlu ve tek katmanlı topolojinin ilişkisi, $\mathbf{W}$ ağırlık matrisinin devriğinin alınıp vektörlerine ayrılmasıyla 1.24 no.lu denklem yardımıyla açık bir şekilde gösterilebilir.

$$\mathbf{W}' = \begin{bmatrix} w_{1,0} & w_{2,0} & w_{3,0} & \dots & w_{m,0} \\ w_{1,1} & w_{2,1} & w_{3,1} & \dots & w_{m,1} \\ w_{1,2} & w_{2,2} & w_{3,2} & \dots & w_{m,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{1,n} & w_{2,n} & w_{3,n} & \dots & w_{m,n} \end{bmatrix} \quad (1.24)$$

$\mathbf{W}'$ ağırlık matrisinin sütunlarını ifade eden vektörler ise 1.25 no.lu denklem ile aşağıda gösterilmektedir.

$$\mathbf{w}'_1 = \begin{bmatrix} w_{1,0} \\ w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix}, \mathbf{w}'_2 = \begin{bmatrix} w_{2,0} \\ w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix}, \mathbf{w}'_3 = \begin{bmatrix} w_{3,0} \\ w_{3,1} \\ w_{3,2} \\ \vdots \\ w_{3,n} \end{bmatrix}, \dots, \mathbf{w}'_m = \begin{bmatrix} w_{m,0} \\ w_{m,1} \\ w_{m,2} \\ \vdots \\ w_{m,n} \end{bmatrix} \quad (1.25)$$

$\mathbf{w}'_1$ vektörünün yukarıdaki tek nöronlu ve tek katmanlı $\mathbf{w}$ vektörü ile (indis işaretleri dışında) aynı olduğu açıkça görülmektedir. Çıktı katmanında m adet nöron bulunduğundan, m adet $\mathbf{w}'$ vektörü vardır. Diğer bir ifadeyle $\mathbf{W}'$ matrisi, m adet nöron içeren tek katmanlı bir topolojide, m adet $\mathbf{w}'$ vektöründen oluşur. O halde her bir nöronun k. örnekleme göre aktivasyon gerilimini elde etmek için $[m \times (n+1)]$ boyutlu $\mathbf{W}$ matrisi, sağdan $[(n+1) \times 1]$ boyutlu $\mathbf{x}$ vektörü ile çarpılır ve nöronların aktivasyon gerilimlerini veren $[m \times 1]$ boyutlu bir $\mathbf{u}$ vektörü 1.26 ve 1.27 no.lu denklemlerde gösterildiği gibi elde edilir.

$$\mathbf{u} = \begin{bmatrix} w_{1,0} & w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,0} & w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ w_{3,0} & w_{3,1} & w_{3,2} & \dots & w_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{m,0} & w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_m \end{bmatrix} \quad (1.26)$$

$$\mathbf{u} = \mathbf{W}\mathbf{x} \quad (1.27)$$

Tek nöronlu ve tek katmanlı topolojinin aksine, çok nöronlu ve tek katmanlı topolojide k. örneklem için aktivasyon gerilimi u bir skaler değil, m elemanlı vektörel bir büyüklüktür. Çünkü her nöronun kendine ait bağımsız birer operasyonel fonksiyonu, aktivasyon gerilimi ve aktivasyon fonksiyonu gibi temel bileşenleri vardır. Bu durumda u vektörü, her bir nöronun k. örneklem için aktivasyon gerilimini verir. Her bir nöronun tüm örnekleme göre aktivasyon gerilimini elde etmek amacıyla, 1.27 no.lu denklem sisteminde $\mathbf{x}$ vektörü yerine $\mathbf{X}$ matrisi yazılarak 1.28, 1.29 ve 1.30 no.lu denklemler elde edilir.

$$\mathbf{U} = \begin{bmatrix} w_{1,0} & w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,0} & w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ w_{3,0} & w_{3,1} & w_{3,2} & \dots & w_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{m,0} & w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} \begin{bmatrix} x_{1,0} & x_{1,1} & x_{1,2} & \dots & x_{1,t} \\ x_{2,0} & x_{2,1} & x_{2,2} & \dots & x_{2,t} \\ x_{3,0} & x_{3,1} & x_{3,2} & \dots & x_{3,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,0} & x_{n,1} & x_{n,2} & \dots & x_{n,t} \end{bmatrix} \quad (1.28)$$

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,t} \\ u_{2,1} & u_{2,2} & u_{2,3} & \dots & u_{2,t} \\ u_{3,1} & u_{3,2} & u_{3,3} & \dots & u_{3,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{m,1} & u_{m,2} & u_{m,3} & \dots & u_{m,t} \end{bmatrix} \quad (1.29)$$

$$\mathbf{U} = \mathbf{W}\mathbf{X} \quad (1.30)$$

Daha önce belirtildiği gibi $[m*(n+1)]$ boyutlu $\mathbf{W}$ matrisi ile $[(n+1)*t]$ boyutlu $\mathbf{X}$ matrisi çarpılırsa, $[m*t]$ boyutlu bir $\mathbf{U}$ matrisi elde edilir. Her nöron toplamda örneklem sayısı kadar (t defa) aktivasyon gerilimi üreteceğinden, u artık bir vektör değil, $[m*t]$ elemanlı matristir.

Tek nöronlu ve tek katmanlı bir algılayıcı, çıktısı 0 ya da 1 olabileceğinden, doğrusal olarak ayrılabilen iki kategoriye sınıflandırılabilir. Her bir kategori farklı bir çıktı vektörü ile temsil edildiğinden, m sayıda nöron içeren tek katmanlı bir algılayıcı ise doğrusal olarak ayrılabilen 2^m adet kategoriye sınıflandırılabilir (Hagan ve diğerleri, 2014, s.87).

Denetimli öğrenme yöntemiyle eğitilen, Frank Rosenblatt'ın tek katmanlı algılayıcısı Hebb öğrenme algoritmasını kullanır. Bu algoritmaya göre üretilen çıktı ile beklenen çıktı arasında fark varsa, sinaptik ağırlıklar ve eşik değerler girdiye göre yeniden hesaplanır; üretilen çıktı ile beklenen çıktı örtüşüyorsa sinaptik ağırlıklar ve eşik değerler değişmeden aynı kalır. Bu süreç, üretilen çıktılar ile beklenen çıktılar benzer olana kadar tüm örneklem değerleri için (t defa) sıralı bir şekilde tekrar eder (Silva ve diğerleri, 2017, s.33).

Tek nörondan oluşan bir topoloji için Hebb öğrenme algoritmasının matematiksel formu 1.31 no.lu denklemde sunulmaktadır (Hagan ve diğerleri, 2014, s.206).

$$w_i^{s+1} = w_i^s + \alpha(d^k - y^k) \cdot x_i^k \quad (1.31)$$

1.31 no.lu denklemde s adım sırası olmak üzere, w_i^s i'inci girdinin önceki sinaptik ağırlığı, w_i^{s+1} i.'nci girdinin mevcut sinaptik ağırlığı, x_i^k i'inci girdinin k 'inci (gözlem) değeri, d^k k 'inci beklenen çıktının değeri, ağırlık k . çıktısını y^k , α ise 0'dan büyük 1'den küçük sabit bir değere sahip olup öğrenme parametresini ifade eder. Örneklem sayısı t olmak üzere, yukarıdaki denklem k . örneklem için ağırlıkların güncellenmesini işaret eder. Dolayısıyla her adım (epoch, yeniden hesaplama/öğrenme sürecini temsil eden s ifadesi), örneklem sayısı t kadar hesaplama süreci içerir.

$\mathbf{w}$ ve $\mathbf{x}^k$ $[(n+1)*1]$ boyutlu birer vektör olmak üzere, eşik değeri temsil eden 1 sabiti ile birlikte $(n+1)$ adet girdi ve her girdinin ağırlığı olduğuna göre 1.31 no.lu denklem vektörel formda 1.32 no.lu denklemde gösterildiği gibi ifade edilebilir (Hagan ve diğerleri, 2014, s.206).

$$\mathbf{w}^{s+1} = \mathbf{w}^s + \alpha(d^k - y^k) \mathbf{x}^k \quad (1.32)$$

Denklemde yer alan d^k ve y^k ifadeleri birer skalerdir. Denklemde göre toplam ağırlık sayısını işaret eden $(n+1)$ adet ağırlık, k . örneklem için “eş anlı” bir şekilde değişmektedir. Ağırlıkları k . örneklem yerine tüm örneklemelere göre ifade etmek için 1.33 no.lu denklemde gösterildiği gibi denkleme $\mathbf{X}$ matrisi ilave edilir.

$$\mathbf{w}^{s+1} = \mathbf{w}^s + \alpha \mathbf{X}(\mathbf{d} - \mathbf{y}) \quad (1.33)$$

$\mathbf{d}$ ve $\mathbf{y}$ ifadeleri örneklem sayısına işaret eden $(t*1)$ boyutlu birer vektör olduğu için farkları da $(t*1)$ boyutlu bir vektör olacaktır. $[(n+1)*t]$ boyutlu $\mathbf{X}$ matrisi, $\mathbf{d} - \mathbf{y}$ vektörü ile soldan çarpılarak $\mathbf{w}$ vektörü ile aynı boyutta $[(n+1)*1]$ boyutlu yeni bir vektör elde edilir. 1.33 no.lu denklem sistemi, tek nöronlu bir topolojide, n adet ağırlığın her s adımında (epoch) ya da öğrenme sürecinde tüm örneklem değerlerine göre değiştirildiğine işaret etmektedir.

Yukarıdaki yöntemden yararlanılarak m adet nörondan oluşan bir topoloji için Hebb öğrenme algoritmasının k . örnekleme göre matris formu 1.34

no.lu denklemde gösterildiği şekilde ifade edilebilir (Hagan ve diğerleri, 2014, s.92).

$$\mathbf{W}^{s+1} = \mathbf{W}^s + \alpha(\mathbf{d}^k - \mathbf{y}^k)\mathbf{x}^k \quad (1.34)$$

m adet nöronun her birinin eşik değerler dahil (n+1) adet ağırlıklandırılmış girdisi olduğundan ağırlıklar, $[m^*(n+1)]$ boyutlu $\mathbf{W}$ ağırlık matrisi ile ifade edilir. Bu durumda $\mathbf{d}^k$ ve $\mathbf{y}^k$ ifadeleri (m^*1) boyutlu birer vektörü temsil edecektir. $\mathbf{x}^k$ ifadesi ise k. örnekleme göre $[1^*(n+1)]$ boyutlu bir girdi vektörüdür. $[m^*1]$ boyutlu $\mathbf{d}^k - \mathbf{y}^k$ vektörünün sağdan $[1^*(n+1)]$ boyutlu $\mathbf{x}^k$ vektörü ile çarpımı, $\mathbf{W}$ ağırlık matrisiyle aynı boyutta, $[m^*(n+1)]$ boyutlu bir matris olacaktır. Bu durumda, toplam ağırlık sayısını ifade eden $m^*(n+1)$ adet ağırlık, k. örneklem için “eş anlı” bir şekilde değiştirilir. Ağırlıkları k. örneklem yerine tüm örneklere göre ifade etmek için 1.35 no.lu denklemde gösterildiği gibi denkleme $\mathbf{X}$ matrisi ilave edilir.

$$\mathbf{W}^{s+1} = \mathbf{W}^s + \alpha(\mathbf{D} - \mathbf{Y})\mathbf{X}^T \quad (1.35)$$

Tüm (t adet) örneklemlerin hesaba katıldığı çok nöronlu ağ topolojisinde, (m^*1) boyutlu $\mathbf{d}^k$ ve $\mathbf{y}^k$ vektörleri yerine ise (m^*t) boyutlu $\mathbf{D}$ ve $\mathbf{Y}$ matrisleri bulunur. Çünkü çıktı katmanında m adet nöron bulunduğundan t adet örneklemin her biri için m adet beklenen ve hesaplanan çıktı değeri olacaktır. Diğer taraftan (m^*t) boyutlu $\mathbf{D}$ ve $\mathbf{Y}$ matrislerinin soldan $[t^*(n+1)]$ boyutlu $\mathbf{X}^T$ matrisi ile çarpımı, $\mathbf{W}$ matrisiyle aynı boyutta $[m^*(n+1)]$ boyutlu bir matrise eşit olacaktır. Yukarıdaki denklem sistemi, çok (m adet) nöronlu bir topolojide, n adet ağırlığın her s adımında (epoch) ya da öğrenme sürecinde tüm örneklem değerlerine göre değiştirildiğine işaret etmektedir.

Alternatif bir bakış açısına göre yukarıda açıklanan öğrenme süreci, esasen, ağırlık değerlerini işaret eden eğim ile eşik değeri ifade eden sabitin sürekli olarak değiştirilerek karar sınırının en uygun şekilde konumlandırılmasıdır (Öztemel, 2006, s.61).

Eğer ağın çıktısı ile beklenen çıktı değerleri arasında bir fark olmazsa ya da en aza indirilirse, eğitim sürecinin tamamlandığı kabul edilir. Ağın eğitimi

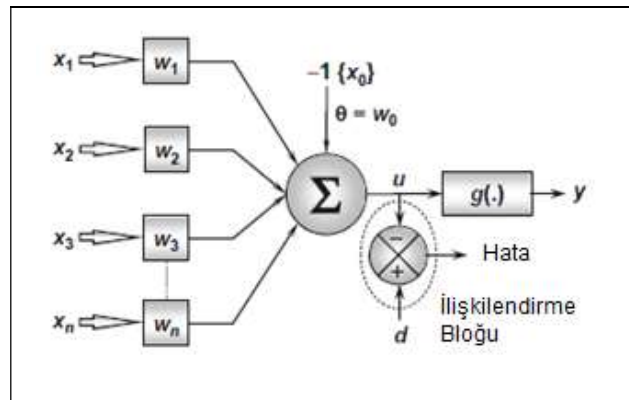
tamamlandıktan sonra, ağa girdi olarak yeni örneklemeler (ya da test seti) verildiğinde, örneklemelerin ait oldukları sınıflara göre atanması ağ tarafından gerçekleştirilir.

1.6.2. Adaptif Doğrusal Eleman ve Delta Öğrenme Algoritması

Adaptif doğrusal eleman (adaptive linear element, ADALINE), 1960 yılında Bernard Widrow ve Marcian Hoff tarafından geliştirilmiştir. ADALINE, aktivasyon fonksiyonun (türevlenebilir) doğrusal bir fonksiyon olması dışında, Frank Rosenblatt tarafından geliştirilen algılayıcı ile oldukça benzerdir. “En küçük ortalama kare” (least mean square, LMS) olarak da adlandırılan öğrenme algoritması, Rosenblatt’ın algılayıcısında kullanılan öğrenme algoritmasından çok daha güçlüdür (Hagan ve diğerleri, 2014, s.310).

ADALINE basit bir ağ olmasına rağmen, “delta öğrenme algoritması”nın geliştirilmesi, paralel sinyal işleme gibi birçok pratik probleme uygulanması, endüstriyel seviyede telefon ağlarının anahtarlanmasıyla kullanılan ilk yapay sinir ağı olması gibi sebeplerden dolayı yapay sinir ağı alanına önemli katkıları olmuştur. Ancak bunların en önemlisi, çok katmanlı ağlarda (multiple layer perceptron) “genelleştirilmiş delta öğrenme algoritması” olarak bilinen, delta öğrenme algoritmasının geliştirilmesidir (Silva ve diğerleri, 2017, s.41; Fausett, 1994, s.80-81).

Şekil 1.10’da tek nörondan oluşan ADALINE’nin mimarisi gösterilmektedir.



Şekil 1.10. ADALINE

Kaynak: Silva ve diğerleri, 2017, s.42

Tek nöronlu algılayıcıda olduğu gibi tek nöronlu ADALINE da iki elemanlı doğrusal sınıflandırma problemlerinde kullanılır. Ancak sınıfların doğrusal olarak ayrılabilmesi koşuluyla, birden fazla nörona sahip bir ADALINE, çoklu sınıflandırma problemlerinin çözümünde de kullanılabilir (Mehrotra ve diğerleri, 1996, s.60-61). Günümüzde de doğrusal sınıflandırma problemlerinde sıklıkla kullanılan ADALINE'in işleyişine ilişkin süreç Frank Rosenblatt tarafından geliştirilen algılayıcının işleyişiyle oldukça benzerdir. Bu çerçevede aşağıda yer alan 1.36 ve 1.37 no.lu denklemler, tek katmanlı algılayıcıda yer alan 1.13 ve 1.14 no.lu denklemlerle aynıdır.

$$u = \sum_{i=0}^n w_i x_i \quad (1.36)$$

$$y = g(u) \quad (1.37)$$

Aktivasyon gerilimi u , k . örnekleme göre (tek katmanlı algılayıcıda 1.17 no.lu denklem üzerinden detaylı olarak açıklandığı gibi) 1.38 no.lu denklem yardımıyla vektörel formda ifade edilebilir.

$$u = \mathbf{w}'\mathbf{x} \quad (1.38)$$

Benzer şekilde aktivasyon gerilimi u , tüm örnekleme göre (tek katmanlı algılayıcıda 1.20 no.lu denklem üzerinden detaylı olarak açıklandığı gibi) 1.39 no.lu denklem aracılığıyla vektörel formda ifade edilebilir.

$$\mathbf{u} = \mathbf{w}'\mathbf{X} \quad (1.39)$$

Birden fazla nörona sahip ADALINE ağının k . örnekleme göre (tek katmanlı algılayıcıda 1.27 no.lu denklem üzerinden detaylı olarak açıklandığı gibi) 1.40 no.lu denklem kullanılarak vektörel formda ifade edilebilir.

$$\mathbf{u} = \mathbf{W}\mathbf{x} \quad (1.40)$$

Birden fazla nörona sahip ADALINE ağının tüm örnekleme göre (tek katmanlı algılayıcıda 1.30 no.lu denklem üzerinden detaylı olarak açıklandığı gibi) 1.41 no.lu denklem ile matris formunda ifade edilebilir.

$$\mathbf{U} = \mathbf{W}\mathbf{X} \quad (1.41)$$

Bu çerçevede tek nöronlu ADALINE ile Rosenblatt tarafından geliştirilen algılayıcı arasındaki temel fark, ağırlıkların ve eşik değerlerin ayarlanmasında kullanılan öğrenme algoritmasıdır. ADALINE ağıının parametrelerinin hesaplanması süreci, günümüzde halen yaygın olarak kullanılan “Widrow-Hoff öğrenme algoritması”, “en küçük ortalama kare” (least mean square, LMS), “gradyan azalma” (gradient descent) olarak da adlandırılan “delta öğrenme algoritması”dır.

Ağın ürettiği k . aktivasyon gerilimi ile beklenen çıktı değeri arasındaki farkı temsil eden “ilişkilendirme bloğu” (association block), sinaptik ağırlıkların ve eşik değerin hesaplanmasında kullanılır (Silva ve diğerleri, 2017, s.43).

$$e^k = d^k - u^k \quad (1.42)$$

Gradyan azalma (delta kuralı), t adet örnek olduğu varsayımında, 1.42 no.lu denklemde gösterildiği üzere ağırlıkları hesaplamak için her bir beklenen örnek değeri d ile aktivasyon gerilimi u arasındaki farkı minimize etmeyi amaçlar. Daha teknik bir ifadeyle, gradyan azalma algoritması optimal w ağırlıklarını hesaplamak için tüm veri setinin (d ve u arasındaki) hata kareleri toplamını $E(w)$ olabildiğince azaltmayı amaçlar (Silva ve diğerleri, 2017, s.44). $\mathbf{w}$ herhangi bir nöronun ağırlık vektörü olmak üzere, gradyan azalma (ya da LMS), maliyet fonksiyonu olarak da bilinen ağırlıkların (istatistiksel anlamda) beklenen değerinin (expectation value, $E(\mathbf{w})$), herhangi bir andaki hata karelerinin toplamının minimize edilmesi olarak tanımlanabilir (Haykin, 2009, s.102; Nielsen, 2015, s.16; Kröse ve Smagt, 1996, s.28). Gradyan azalmanın matematiksel formu, 1.43 no.lu denklemle ifade edilebilir.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^t (d^k - u^k)^2 \quad (1.43)$$

1.43 no.lu denklemde, u^k yerine 1.36 no.lu denklem yazılarak 1.44 no.lu denklem elde edilir.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^t \left(d^k - \sum_{i=0}^n w_i x_i^k \right)^2 \quad (1.44)$$

$\mathbf{w}$ ve $\mathbf{x}^k$ birer vektör olmak üzere, 1.44 no.lu denklem $\mathbf{w}$ vektörü, sağdan $\mathbf{x}^k$ vektörünün devriğiyle çarpılarak 1.45 no.lu denklem yardımıyla vektörel formda ifade edilebilir.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^t (d^k - \mathbf{w}' \mathbf{x}^k)^2 \quad (1.45)$$

∇ ifadesi matrislerde/vektörlerde türev operatörünü temsil etmek üzere, 1.46 no.lu denklemde gösterildiği gibi $\mathbf{w}$ sinaptik ağırlıklar vektörüne göre kısmi türev alınır.

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \sum_{k=1}^t (d^k - \mathbf{w}' \mathbf{x}^k) (-\mathbf{x}^k) \quad (1.46)$$

1.46 no.lu denklemde, 1.38 no.lu denkleme göre $\mathbf{w}' \mathbf{x}^k$ yerine u^k skaleri yazılarak 1.47 no.lu denkleme ulaşılır.

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = - \sum_{k=1}^t (d^k - u^k) \mathbf{x}^k \quad (1.47)$$

$\nabla E(\mathbf{w})$ ifadesinin 0'dan büyük ve 1'den küçük sabit bir değere sahip α öğrenme katsayısı ile çarpımı, 1.48 ve 1.49 no.lu denklemlerde temsil edildiği üzere, $\mathbf{w}$ sinaptik ağırlık vektörünün mevcut ve bir önceki değeri arasındaki farkı ifade eden $\Delta \mathbf{w}$ kadarlık bir değişime yol açtığına işaret eder.

$$\Delta \mathbf{w} = - \alpha \nabla E(\mathbf{w}) \quad (1.48)$$

$$\Delta \mathbf{w} = - \alpha \sum_{k=1}^t (d^k - u^k) \mathbf{x}^k \quad (1.49)$$

α parametresine büyük bir değer atanması, hatalara göre güncellenen ağırlık matrisinin optimum değerlere yakınsamamasına; çok küçük bir değer atanması ise öğrenme sürecinin çok yavaş gerçekleşmesine yol açar.

(Fausett, 1994, s.82). Gradyan azalmada ortalama hata karelerinin minimize edilmesi amaçlanır. Bu çerçevede s ifadesi parametrelerin güncellenme adımlarını ifade etmek üzere, her s adımında ağırlık vektörü eğimin tersi yönde değişeceğinden, 1.50 no.lu denklemde gösterildiği gibi α parametresinin solundaki negatif işaret pozitif olarak ifade edilir (Silva ve diğerleri, 2017, s.45).

$$\mathbf{w}^{s+1} = \mathbf{w}^s + \alpha \sum_{k=1}^t (d^k - u^k) \mathbf{x}^k \quad (1.50)$$

1.50 no.lu denklemde $\mathbf{w}^{s+1}$ ve $\mathbf{w}^s$ vektörleri, $\Delta \mathbf{w}$ ifadesinin ayrıştırılmasından elde edilmiştir. Denklem k . örneklem için 1.51 nolu denklemde gösterildiği gibi (örnek sayısı t görmezden gelinerek) daha sade ifade edilebilir.

$$\mathbf{w}^{s+1} = \mathbf{w}^s + \alpha (d^k - u^k) \mathbf{x}^k \quad (1.51)$$

Tek katmanlı algılayıcıda yer alan 1.17 ve 1.32 no.lu denklemlerde belirtildiği gibi yukarıdaki 1.51 no.lu denklemde yer alan d^k ve u^k ifadeleri birer skalerdir. Diğer taraftan 1.51 no.lu denklem, delta öğrenme kuralını temsil eder. Delta öğrenme kuralının Hebb öğrenme kuralından farkı, beklenen çıktının y (hesaplanan çıktı) yerine u (aktivasyon gerilimi) ile kıyaslanmasıdır. 1.51 no.lu denkleme göre tüm ağırlıklar, Hebb öğrenme kuralında olduğu gibi k . örneklem için “eş anlı” bir şekilde değişmektedir. Ağırlıkları k . örneklem yerine tüm örneklemelere göre ifade etmek için 1.51 no.lu denkleme $\mathbf{X}$ matrisi ilave edilerek 1.52 no.lu denklem sistemi elde edilir.

$$\mathbf{w}^{s+1} = \mathbf{w}^s + \alpha \mathbf{X}(\mathbf{d} - \mathbf{u}) \quad (1.52)$$

$\mathbf{d}$ ve $\mathbf{u}$ ifadeleri, örneklem sayısına işaret eden $(t+1)$ boyutlu birer vektör olduğu için farkları da $(t+1)$ boyutlu bir vektör olacaktır. $[(n+1) \times t]$ boyutlu $\mathbf{X}$ matrisi, soldan $\mathbf{d} - \mathbf{u}$ vektörü ile çarpılarak $\mathbf{w}$ vektörü ile aynı boyutta $[(n+1) \times 1]$ boyutlu yeni bir vektör elde edilir. 1.52 no.lu denklem sistemi, tek nöronlu bir topolojide, n adet ağırlığın her s adımında (epoch) ya da öğrenme sürecinde tüm örneklem değerlerine göre eş anlı şekilde değiştirildiğini ifade eder.

m sayıda nörondan oluşan bir topolojinin girdi ağırlıklarını k. örnekleme göre “eş anlı” bir şekilde güncellemek için denkleme $m \times 1$ boyutlu $\mathbf{d}^k$ ve $\mathbf{u}^k$ vektörleri ilave edilir ve sağdan $[(n+1) \times 1]$ boyutlu $\mathbf{x}^k$ vektörünün devriği ile çarpılarak 1.53 no.lu denkleme ulaşılır (Hagan ve diğerleri, 2014, s.92). Bu sonuç tek katmanlı algılayıcıda yer alan 1.34 no.lu denklemlerle son derece benzerdir.

$$\mathbf{W}^{s+1} = \mathbf{W}^s + \alpha(\mathbf{d}^k - \mathbf{u}^k)\mathbf{x}^{rk} \quad (1.53)$$

m adet nöronun her birinin eşik değerler dahil $(n+1)$ adet ağırlıklandırılmış girdisi olduğundan ağırlıklar, $[m \times (n+1)]$ boyutlu $\mathbf{W}$ matrisi ile temsil edilir. Beklendiği gibi $[m \times 1]$ boyutlu $\mathbf{d}^k - \mathbf{u}^k$ vektörünün sağdan $[1 \times (n+1)]$ boyutlu $\mathbf{x}^{rk}$ vektörü ile çarpımı, $\mathbf{W}$ ağırlık matrisiyle aynı boyutta, $[m \times (n+1)]$ boyutlu bir matris olacaktır. Bu durumda, toplam ağırlık sayısını ifade eden $m \times (n+1)$ adet ağırlık, k. örneklem için “eş anlı” bir şekilde değiştirilir. Ağırlıkları k. örneklem yerine tüm örneklemlere göre ifade etmek için 1.54 no.lu denklemlerde gösterildiği gibi 1.53 no.lu denkleme $\mathbf{X}$ matrisi ilave edilir. Bu sonuç ise tek katmanlı algılayıcıda yer alan 1.35 no.lu denklemlerle, diğer bir ifadeyle Hebb öğrenme algoritmasıyla son derece benzerdir.

$$\mathbf{W}^{s+1} = \mathbf{W}^s + \alpha(\mathbf{D} - \mathbf{U})\mathbf{X}^T \quad (1.54)$$

Tüm örneklemlerin sayısı t olduğundan, $m \times t$ boyutlu $\mathbf{D} - \mathbf{U}$ matrisi ile $[(n+1) \times t]$ boyutlu $\mathbf{X}$ matrisinin devriğinin çarpımı, Hebb öğrenme algoritmasında olduğu gibi $[m \times (n+1)]$ boyutlu $\mathbf{W}$ matrisinin boyutu ile aynıdır.

Aşağıda yer alan Grafik 1.4’te gradyan azalma algoritmasının işleyişine ilişkin görsele yer verilmiştir.

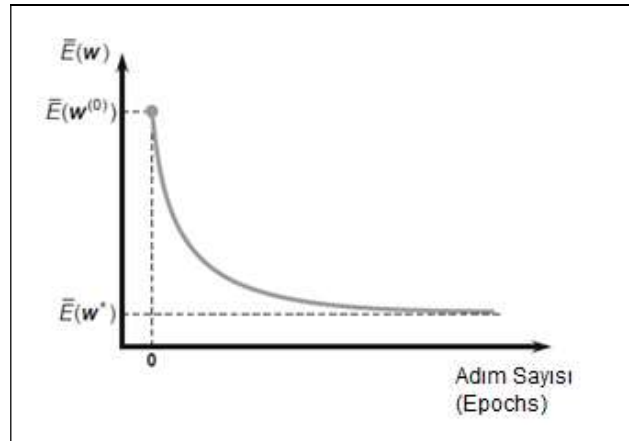
değerlerinin hesaba katıldığı ortalama hata kareleri $\bar{E}(w)$ kullanılarak durdurma kriteri koşuluna bağlanabilir.

$$\bar{E}(w) = \frac{1}{t} \sum_{k=1}^t (d^k - u^k)^2 \quad (1.55)$$

Öğrenme sürecini temsil eden s adımları arasında, 1.56 no.lu denklemde ifade edildiği gibi ortalama hata karesinin değişimi çok az ise algoritmanın yakınsamayı gerçekleştirmiş olduğu kabul edilebilir (Silva ve diğerleri, 2017, s.46-47).

$$\bar{E}(w^{s+1}) - \bar{E}(w^s) \leq \varepsilon \quad (1.56)$$

Çok küçük bir değere sahip ε ifadesi yakınsama için gerekli durdurma koşulunu (precision value) temsil etmektedir. Grafik 1.5'te ortalama hata kareleri fonksiyonunun adım (epoch) sayısına göre davranışı gösterilmektedir.



Grafik 1.5. Gradyan Azalma: Adım Sayısı

Kaynak: Silva ve diğerleri, 2017, s.49

Eğitim süreci tamamlandıktan sonra, hesaplanan w ağırlık vektörünün devriği, soldan yeni (ağın daha önce görmediği) örneklem girdisini ifade eden x^{yeni} vektörü ile çarpılarak aktivasyon gerilimi u elde edilir.

$$u = w'x^{yeni} \quad (1.57)$$

$$g(u) = g(w'x^{yeni}) \quad (1.58)$$

1.57 no.lu denklemde yer alan u değerini girdi olarak alan ve 1.58 no.lu denklemle ifade edilen aktivasyon fonksiyonunun çıktısı ise ağıın yeni örnekleme atadığı sınıfı ifade eder.

1.7. Çok Katmanlı İleri Beslemeli Ağlar

Kullanılan öğrenme algoritmasına göre çok sayıda türü bulunan ileri beslemeli çok katmanlı yapay sinir ağı (multi-layer feed-forward neural network, FNN) mimarisi, girdi ve çıktı katmanları arasında en az bir gizli katman içeren ağlardır. Bu çerçevede (en önemli/yaygın öğrenme algoritmalarından biri olan) “geri yayılım” (backpropagation) öğrenme algoritmasının uygulandığı FNN mimarileri, “çok katmanlı algılayıcı” (multi-layer perceptron, MLP) olarak da adlandırılır. Bununla birlikte her FNN ya da MLP mimarisi geri yayılım algoritması kullanmayabilir (Silva ve diğerleri, 2017, s.55; Bishop, 1995, s.116). Bu çerçevede “MLP” ifadesi, “geri yayılım” (backpropagation) öğrenme algoritmasının kullanılmadığı FNN mimarilerini de işaret edebilmektedir. Bu yüzden MLP ifadesi, bir FNN türü olarak değil, FNN mimarisi yerine kullanılan bir terim olarak düşünülebilir.

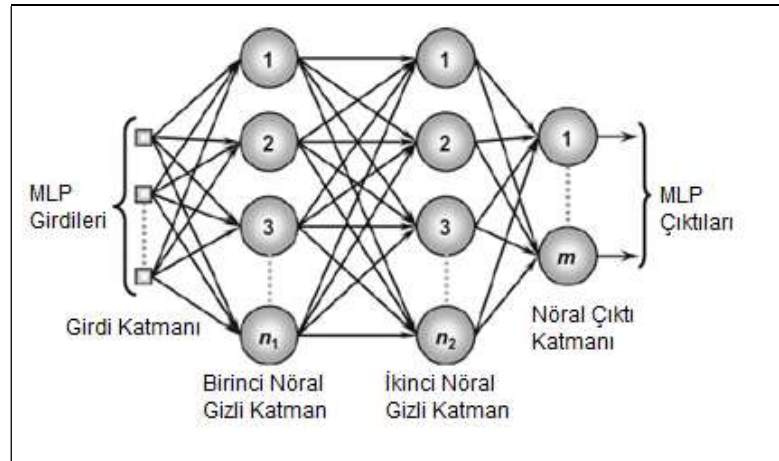
Eğitimi denetimli bir süreçle gerçekleştiren çok katmanlı FNN (ya da MLP) mimarisine sahip ağlar, sadece doğrusal problemlerin çözümünde kullanılan tek katmanlı algılayıcı ve ADALINE ağlarının aksine, doğrusal olmayan problemlerin çözümünde de yaygın olarak kullanılırlar (Mehrotra ve diğerleri, 1996, s.65).

Çok katmanlı ağların eğitimine ilişkin ilk öğrenme algoritması, 1974 yılında Paul Werbos tarafından tasarlanmış olmasına rağmen, yapay sinir ağı topluluğunda yaygınlık kazanmadı. Bununla birlikte Rumelhart ve diğerleri tarafından 1986 yılında yayımlanan “Paralel Dağıtılmış İşlemler” (Parallel Distributed Processing) isimli kitapta, hataların “geri yayılım” (backpropagation) olarak adlandırılan öğrenme algoritması tutarlı bir şekilde açıklanmış ve MLP ağlarının öğrenme sürecine başarılı bir şekilde uygulanmıştır (Silva ve diğerleri, 2017, s.55; Hagan ve diğerleri, 2014, s.358). Böylece MLP ağları, zaman serileri tahmini, örüntü tanıma, sistem optimizasyonu gibi farklı alanlarda çok çeşitli problemlerin çözümü için oldukça

geniş bir uygulama alanına ve tanınırlığı ulaşmış, yapay zeka çalışmalarına olan ilgiyi yeniden canlandırmıştır.

MLP ağlarında, gizli katmanların tüm nöronlarında doğrusal aktivasyon fonksiyonu kullanılması, ağı tek katmanlı bir algılayıcıya dönüştürür (Bishop, 1995, s.121). Bu yüzden MLP ağlarının gizli nöral katmanlarının tümünde doğrusal aktivasyon fonksiyonu kullanılmamalıdır. Bununla birlikte MLP mimarisinde, genellikle tek gizli katman ve ağı nöronlarında ise sigmoid tipi doğrusal olmayan aktivasyon fonksiyonları kullanılır (Kröse ve Smagt, 1996, s.33).

Özellikle de geri yayılım algoritmasını kullanan MLP topolojisinin eğitimi nispeten yavaş gerçekleşmesine rağmen, eğitilmiş bir ağı çıktı üretmesi çok hızlıdır. Tek katmanlı ağların örüntü/ilişki çıkartabilme yeteneği sınırlı olmasına rağmen, bir ya da daha fazla gizli katman içeren çok katmanlı mimariler herhangi bir problemin çözümünde çok daha başarılı sonuçlar üretebilmektedir. Bununla birlikte bazı uygulamalarda birden çok gizli katman faydalı olsa da genellikle tek bir gizli katman (diğer bir ifadeyle çıktı katmanıyla birlikte iki katmanlı mimari) yeterli olmaktadır (Fausett, 1994, s.290).



Şekil 1.11. Üç Katmanlı MLP

Kaynak: Silva ve diğerleri, 2017, s.30

Şekil 1.11'de gösterildiği gibi bir MLP mimarisi, örneğin çok sayıda nöron içeren tek katmanlı 3 farklı algılayıcının katmanlandırılarak bir araya getirilmesine benzetilebilir. MLP ağının girdileri, ilk nöral (gizli) katmanın çıktısı sonraki nöral (gizli) katmanın girdisi, son gizli katmanın çıktısı ise çıktı

katmanının girdisi olacak şekilde katmanlar arasından geçerek çıktı katmanına ulaşır. Her katmanda farklı sayıda nöron ve farklı aktivasyon fonksiyonları bulunabilir (Hagan ve diğerleri, 2014, s.358). Herhangi bir katmandaki nöronların tümü, önceki katmandaki nöronların tümüyle bağlantılıdır. MLP ağında “girdi akışı” soldan/sağa ya da girdi katmanından çıktı katmanına doğru gerçekleşir (Haykin, 2009, s.124).

1.7.1. Geri Yayılım Algoritması

Geri yayılım algoritması, hata geri yayılımı (backpropagation), geriye doğru hata yayılımı (back error propagation), genelleştirilmiş delta kuralı (generalized delta rule) olarak da bilinir. Bu şekilde eğitilen ağlar bazen çok katmanlı algılayıcılar veya MLP (multi-layer perceptron) olarak da adlandırılır (Gurney, 1997, s.103).

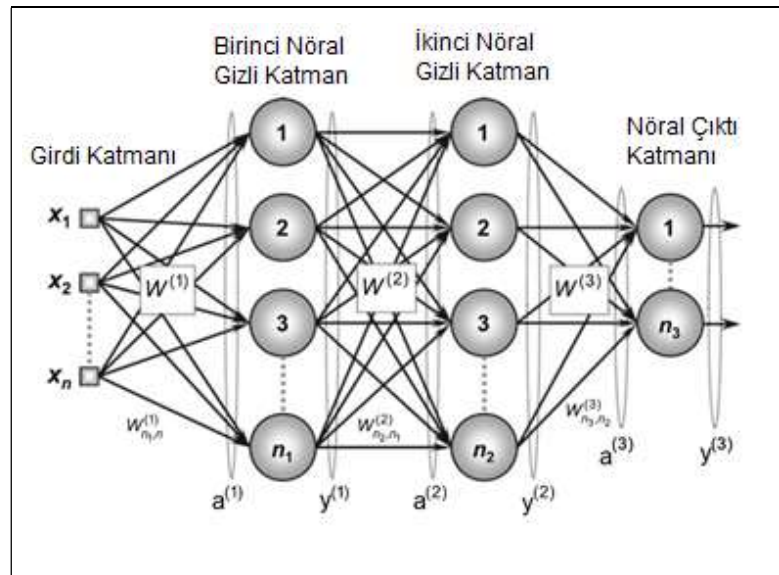
Eğitim sürecinde “geri yayılım” (backpropagation) algoritmasını kullanan MLP, ağırlıkların ve eşik değerlerin hesaplanmasında denetimli öğrenme yönteminden yararlanır. Geri yayılım, “hataların” çıktı katmanından girdi katmanına doğru geriye doğru aktarımına işaret eder (Fausett, 1994, s.289). Geri yayılım algoritması genellikle iki aşamalı uygulanır.

“İleri yayılım” (forward propagation) olarak adlandırılan ilk aşamada, yukarıdaki şekilde görüldüğü gibi bir örneklem ağa girdi olarak verilir ve nihai çıktılara karşılık gelen üretim sürecine kadar tüm nöral katmanlar arasında yayılır. Bu aşamada, herhangi bir değişikliğe uğramayan nöronların, mevcut sinaptik ağırlıkları ve eşik değerleri dikkate alınarak ağdan bir yanıt alınması amaçlanır.

Ağın çıktı katmanında birden fazla nöron varsa, nöronların sinaptik ağırlıklarını ve eşik değerlerini güncellemek için ağın ürettiği çıktı değerleriyle beklenen çıktı değerleri arasındaki fark kullanılır. “Geri yayılım” (backward propagation) olarak adlandırılan ikinci aşamada, söz konusu farklar sebebiyle, ağdaki tüm nöronların sinaptik ağırlıkları ve eşik değerleri güncellenir. Süreç, ağın ürettiği çıktı değerleriyle beklenen çıktı değerleri arasındaki farkın minimizasyonu ya da ağın ürettiği çıktıların beklenen çıktılara yakınsaması

sağlanana kadar yineleyici bir şekilde devam eder (Silva ve diğerleri, 2017, s.57-58). Geri yayılım algoritmasını türetmek için ileri beslemeli bir mimaride, doğrusal olmayan hata ve aktivasyon fonksiyonları türevlenebilir sürekli fonksiyonlar olmalıdır (Bishop, 1995, s.141).

Her katman kendinden önceki katmanın çıktısını girdi olarak alır. Birinci katmanın girdisi, girdi katmanıdır. İkinci katmanın girdisi, birinci katmanın çıktısıdır. Üçüncü katmanın girdisi, ikinci katmanın çıktısıdır. Süreç katman sayısı kadar bu şekilde devam eder. Son olarak çıktı katmanının girdisi ise kendisinden bir önceki katmanın çıktısı olacaktır (Hagan ve diğerleri, 2014, s.358).



Şekil 1.12. Geri Yayılım Algoritması

Kaynak: Silva ve diğerleri, 2017, s.59

Şekil 1.12'de, üssel ifadeler katman sırasını ve n^1 , n^2 ve n^3 ise sırasıyla birinci, ikinci ve üçüncü katmanlarda bulunan nöron sayılarını temsil etmek üzere, $\mathbf{W}^1$, $\mathbf{W}^2$, $\mathbf{W}^3$ ifadeleri, sırasıyla, birinci ve ikinci gizli katmanlar ile çıktı (üçüncü) katmanının j. nöronlarını, kendilerinden bir önceki katmanın i. nöronuna bağlayan $(n^1 \times n)$, $(n^2 \times n^1)$, $(n^3 \times n^2)$ boyutlu sinaptik ağırlıklar matrisini temsil etmektedir. $\mathbf{a}^1$, $\mathbf{a}^2$, $\mathbf{a}^3$ ifadeleri ise birinci, ikinci ve üçüncü (çıktı) katmanların j. nöronlarının ağırlıklandırılmış $(n^1 \times 1)$, $(n^2 \times 1)$, $(n^3 \times 1)$ boyutlu girdi vektörüdür. L katman sırası, $x_0=1$, $y_0^L=1$ ve $w_{j,0}^L=\theta_j^L$ olmak üzere, anlatım sadeliğini sağlamak amacıyla ilgili katmanda bulunan nöronların eşik değerleri,

1.59, 1.60 ve 1.61 no.lu denklemlerde gösterildiği gibi **W** ağırlık matrisinin içinde temsil edilmiştir.

$$a_j^1 = \sum_{i=0}^{n^1-1} w_{ji}^1 x_i = \mathbf{W}^1 \mathbf{x} = \mathbf{a}^1 \quad (1.59)$$

$$a_j^2 = \sum_{i=0}^{n^2-1} w_{ji}^2 y_i^1 = \mathbf{W}^2 \mathbf{y}^1 = \mathbf{a}^2 \quad (1.60)$$

$$a_j^3 = \sum_{i=0}^{n^3-1} w_{ji}^3 y_i^2 = \mathbf{W}^3 \mathbf{y}^2 = \mathbf{a}^3 \quad (1.61)$$

Girdisi $\mathbf{a}^1$, $\mathbf{a}^2$, $\mathbf{a}^3$ vektörleri olan birinci, ikinci ve üçüncü katmanların j. nöronlarının çıktısı ise sırasıyla, 1.62, 1.63 ve 1.64 no.lu denklemlerde gösterildiği gibi $(n^1 \times 1)$, $(n^2 \times 1)$, $(n^3 \times 1)$ boyutlu $\mathbf{y}^1$, $\mathbf{y}^2$, $\mathbf{y}^3$ vektörleridir.

$$\mathbf{y}^1 = g(\mathbf{a}^1) \quad (1.62)$$

$$\mathbf{y}^2 = g(\mathbf{a}^2) \quad (1.63)$$

$$\mathbf{y}^3 = g(\mathbf{a}^3) \quad (1.64)$$

Geri yayılım algoritmasını türetmek için ağ tarafından üretilen çıktılar ile beklenen çıktılar arasındaki sapma ölçülmelidir. Bu çerçevede, k ifadesi örneklem indisini temsil etmek üzere, çıktı katmanının j. nöronunun k. örnekleme göre hata kareleri (maliyet) fonksiyonu (cost function) 1.65 no.lu denklemde gösterilmiştir (Bishop, 1995, s.145; Silva ve diğerleri, 2017, s.61-62).

$$e(k) = \frac{1}{2} \sum_{j=1}^{n^3} \left(d_j(k) - y_j^3(k) \right)^2 \quad (1.65)$$

1.65 no.lu denklem, 1.66 no.lu denklem yardımıyla vektörel formda ifade edilebilir.

$$\mathbf{e}_k = \frac{1}{2} (\mathbf{d}_k - \mathbf{y}_k^3)' (\mathbf{d}_k - \mathbf{y}_k^3) \quad (1.66)$$

Sinaptik ağırlıkların çıktı katmanından girdi katmanına doğru güncellendiği geri yayılım algoritması, çıktı katmanının ağırlık matrisini temsil eden $\mathbf{W}^3$ ve gizli katmanların ağırlık matrislerini ifade eden $\mathbf{W}^1$ ve $\mathbf{W}^2$ olarak iki kısımda analiz edilebilir.

1.7.1.1. Çıktı Katmanına İlişkin Sinaptik Ağırlıkların Analizi

Çıktı katmanının eğitim süreci, beklenen çıktılarla ağ tarafından üretilen çıktılar arasında oluşan hatanın minimize edilmesine dayanır. Çıktı katmanındaki sinaptik ağırlıklar, 1.65 no.lu denklemde ifade edilen çıktı katmanının j. nöronunun k. örnekleme göre hatalarından hareketle, aktivasyon fonksiyonunun türevlenebilir olduğu varsayımında, zincir kuralı yardımıyla elde edilir.

$$\nabla \mathbf{e}^3 = \frac{\partial \mathbf{e}}{\partial \mathbf{W}^3} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^3} \cdot \frac{\partial \mathbf{y}^3}{\partial \mathbf{a}^3} \cdot \frac{\partial \mathbf{a}^3}{\partial \mathbf{W}^3} \quad (1.67)$$

1.67 no.lu denklemle temsil edilen zincir kuralına göre üç kısımdan oluşan türev denklemi, 1.61, 1.64 ve 1.65 no.lu denklemlerden hareketle çözülür.

$$\frac{\partial \mathbf{a}^3}{\partial \mathbf{W}^3} = \mathbf{y}^2 \quad (1.68)$$

$$\frac{\partial \mathbf{y}^3}{\partial \mathbf{a}^3} = g'(\mathbf{a}^3) \quad (1.69)$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^3} = -(\mathbf{d} - \mathbf{y}^3) \quad (1.70)$$

1.68, 1.69 ve 1.70 no.lu denklemler 1.67 no.lu zincir kuralına göre oluşturulan türev denkleminde yerlerine yazılır.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^3} = -(\mathbf{d} - \mathbf{y}^3) \cdot g'(\mathbf{a}^3) \cdot (\mathbf{y}^2)' \quad (1.71)$$

1.71 no.lu denklemde yer alan $g'(a^3)$ ifadesi, girdisi a^3 vektörü olan üçüncü katmanın aktivasyon fonksiyonunun türevini; $(y^2)'$ ifadesi ise üçüncü katmanın ağırlıklandırılmış girdi vektörünün devriğini temsil eder. $(d-y^3).g'(a^3)$ ifadesi (n^3*1) ve $(y^2)'$ ifadesi ise $(1*n^2)$ boyutlu birer vektördür. (Hagan ve diğerleri, 2014, s.366). Hata minimizasyonu amaçlandığı için 1.72 no.lu denklemde yer alan W^3 ağırlık matrisinde değişimin yönü türevin tersi yönde (negatif) 1.73 no.lu denklemde gösterildiği gibi olacaktır.

$$\Delta W^3 = -\alpha \frac{\partial e}{\partial W^3} \quad (1.72)$$

$$\frac{\partial e}{\partial W^3} = \alpha (d-y^3).g'(a^3).(y^2)' \quad (1.73)$$

δ^3 ifadesi, çıktı katmanının j. nöronunun lokal gradyan vektörü olmak üzere, 1.67 no.lu denklemde yer alan zincir kuralından hareketle 1.74 no.lu denklem şeklinde de ifade edilebilir. 1.74 no.lu denklemde, 1.64 ve 1.70 no.lu denklemler yerine yazılarak 1.75 no.lu denklem elde edilir.

$$\delta^3 = -\frac{\partial e}{\partial y^3} \cdot \frac{\partial y^3}{\partial a^3} \quad (1.74)$$

$$\delta^3 = (d-y^3).g'(a^3) \quad (1.75)$$

1.75 no.lu (n^3*1) boyutlu δ^3 vektörü, 1.73 no.lu türev denkleminde yerine yazılarak 1.76 no.lu denklem elde edilir.

$$\Delta W^3 = \alpha \delta^3.(y^2)' \quad (1.76)$$

s ifadesi adım indisi olmak üzere, ΔW^3 ifadesi s+1 ve s adımları arasındaki fark olduğundan, 1.76 no.lu denklem 1.77 no.lu denklemde gösterildiği gibi düzenlenir.

$$W^3(s+1) = W^3(s) + \alpha \delta^3.(y^2)' \quad (1.77)$$

1.77 no.lu denklemde yer alan α ifadesi geri yayılım algoritmasının öğrenme katsayısıdır. Daha önce belirtildiği gibi n^2 ve n^3 ikinci ve üçüncü

katmanların nöron sayıları olmak üzere, $\delta^3 \cdot (y^2)'$ vektörlerinin çarpımı, W^3 matrisinin boyutuna eşit ($n^3 \cdot n^2$) boyutlu bir matris olacaktır. Sonuç olarak geri yayılım algoritmasında, çıktı katmanının ağırlıklarının güncellenmesine işaret eden öğrenme süreci, 1.77 no.lu denklem üzerinden gerçekleşir.

1.7.1.2. Gizli Katmanlara İlişkin Sinaptik Ağırlıkların Analizi

Çıktı katmanının aksine, gizli katmanların nöronları çıktıların beklenen değerlerine doğrudan erişemez. Bu yüzden gizli katmanlarda sinaptik ağırlıklar, yukarıda gösterildiği gibi daha önce çıktı katmanında üretilen hata tahminleri kullanılarak hesaplanır. Diğer bir ifadeyle, çıktı katmanında elde edilen ağırlıklandırılmış girdi vektörleri (a^3), ikinci katmanın sinaptik ağırlıklarının (W^2) güncellenmesinde, ikinci katmanda elde edilen ağırlıklandırılmış girdi vektörleri (a^2) ise birinci katmanın sinaptik ağırlıklarının (W^1) güncellenmesinde kullanılır. Böylece çıktıların beklenen değerlerine doğrudan ulaşamadığı için hataların da doğrudan hesaplanamadığı gizli katmanlarda sinaptik ağırlıklar, çıktı katmanında başlayan ve aşamalı bir şekilde ilk sıradaki gizli katmana doğru devam eden geriye yayılım süreciyle güncellenir (Silva ve diğerleri, 2017, s.64). Dolayısıyla yukarıda gösterildiği gibi çıktı katmanının eğitim sürecinin tamamlanmasını müteakip, çıktı katmanından önceki katman olan ikinci katmanın eğitim sürecinden devam edilir.

1.7.1.2.1. İkinci Katmana İlişkin Sinaptik Ağırlıkların Analizi

Çıktı katmanında uygulanan zincir kuralı, ikinci katmanda yer alan nöronların sinaptik ağırlıklarının analiz edilmesi için 1.78 no.lu denklemde gösterildiği şekilde uyarlanır.

$$\nabla e^2 = \frac{\partial e}{\partial W^2} = \frac{\partial e}{\partial y^2} \cdot \frac{\partial y^2}{\partial a^2} \cdot \frac{\partial a^2}{\partial W^2} \quad (1.78)$$

1.78 no.lu denklemle temsil edilen zincir kuralına göre üç kısımdan oluşan türev denklemi, 1.60, 1.63 ve 1.65 no.lu denklemlerden hareketle çözülür.

$$\frac{\partial \mathbf{a}^2}{\partial \mathbf{W}^2} = \mathbf{y}^1 \quad (1.79)$$

$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{a}^2} = \mathbf{g}'(\mathbf{a}^2) \quad (1.80)$$

Ancak 1.79 ve 1.80 no.lu denklemleri 1.78 no.lu zincir kuralına göre oluşturulan türev denkleminde yerine yazmadan önce, ikinci katmanın türev denkleminin bir parçası olan $\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2}$ ifadesinin türevini almak amacıyla, çıktı katmanının girdi vektörü olan $\mathbf{a}^3$ ifadelerine göre 1.81 no.lu denklemde gösterildiği gibi yeni bir zincir kuralı oluşturulur.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^3} \cdot \frac{\partial \mathbf{a}^3}{\partial \mathbf{y}^2} \quad (1.81)$$

1.81 no.lu denklemin payında bulunan $\mathbf{a}^3$ ifadesi yerine, 1.61 no.lu denklemde $\mathbf{a}^3$ ifadesinin eşiti yazılarak 1.82 no.lu denklem elde edilir.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^3} \cdot \frac{\partial (\mathbf{W}^3 \mathbf{y}^2)}{\partial \mathbf{y}^2} \quad (1.82)$$

Bu aşamada 1.82 no.lu denklemin en sağındaki kısmın türevi alınabilir. Böylece 1.83 no.lu denkleme ulaşılır.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^3} (\mathbf{W}^3) \quad (1.83)$$

1.83 no.lu denklem geri yayılıma ilişkin iki önemli olguya işaret etmektedir. Birincisi $\mathbf{W}^3$ ifadesi, çıktı katmanının ağırlık matrisidir ve çıktı katmanının tüm nöronları $\mathbf{W}^3$ üzerinden ikinci katmanın j. nöronuyla bağlantılıdır. İkincisi ise $\mathbf{W}^3$ ifadesi, bir önceki (çıktı katmanındaki) aşamada gerçek hata değerlerine göre güncellenmiş ağırlık matrisidir. Bu çerçevede $\mathbf{W}^3$ ağırlık matrisinin güncellenmesinde kullanılan gerçek hata değerleri, $\mathbf{W}^3$ ağırlık matrisi üzerinden dolaylı bir şekilde geriye, ikinci katmana doğru yayılır.

1.74 no.lu denklem, $\mathbf{W}^3$ matrisinin devriğini sağdan çarpacak şekilde 1.83 no.lu denklemde yerine yazılarak 1.84 no.lu denklem elde edilir.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = -(\mathbf{W}^3)' \delta^3 \quad (1.84)$$

1.79, 1.80 ve 1.84 no.lu denklemler, 1.78 no.lu zincir kuralına göre oluşturulan türev denkleminde yerlerine yazılarak 1.85 no.lu denkleme ulaşılır.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = -(\mathbf{W}^3)' \delta^3 . g'(\mathbf{a}^2) . (\mathbf{y}^1)' \quad (1.85)$$

1.85 no.lu denklemde yer alan $g'(\mathbf{a}^2)$ ifadesi, girdisi $\mathbf{a}^2$ vektörü olan ikinci katmanın aktivasyon fonksiyonunun türevini; $(\mathbf{y}^1)'$ ifadesi ise ikinci katmanın ağırlıklandırılmış girdi vektörünün devriğini temsil eder. $(\mathbf{W}^3)' \delta^3 . g'(\mathbf{a}^2)$ ifadesi $(n^2 \times 1)$ ve $(\mathbf{y}^1)'$ ifadesi ise $(1 \times n^1)$ boyutlu birer vektördür. (Hagan ve diğerleri, 2014, s.366). Hata minimizasyonu amaçlandığı için 1.86 no.lu denklemde yer alan $\mathbf{W}^2$ ağırlık matrisinde değişimin yönü türevin tersi yönde (negatif), 1.87 no.lu denklemde gösterildiği gibi olacaktır.

$$\Delta \mathbf{W}^2 = -\alpha \frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} \quad (1.86)$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \alpha (\mathbf{W}^3)' \delta^3 . g'(\mathbf{a}^2) . (\mathbf{y}^1)' \quad (1.87)$$

δ^2 ifadesi, ikinci katmanın j. nöronunun lokal gradyan vektörü olmak üzere, 1.78 no.lu denklemde yer alan zincir kuralından hareketle 1.88 no.lu denklem şeklinde de ifade edilebilir. 1.88 no.lu denklemde, 1.63 ve 1.84 no.lu denklemler yerine yazılarak 1.89 no.lu denklem elde edilir.

$$\delta^2 = -\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \cdot \frac{\partial \mathbf{y}^2}{\partial \mathbf{a}^2} \quad (1.88)$$

$$\delta^2 = (\mathbf{W}^3)' \delta^3 . g'(\mathbf{a}^2) \quad (1.89)$$

1.89 no.lu denklem, üç katmanlı bir yapay sinir ağında çıktı katmanında üretilen hataların ikinci nöral katmana yayılımını ifade etmektedir.

1.89 no.lu ($n^2 \times 1$) boyutlu δ^2 vektörü, 1.87 no.lu türev denkleminde yerine yazılarak 1.90 no.lu denklem elde edilir.

$$\Delta \mathbf{W}^2 = \alpha \cdot \delta^2 \cdot (\mathbf{y}^1)' \quad (1.90)$$

s ifadesi adım indisi olmak üzere, $\Delta \mathbf{W}^2$ ifadesi s+1 ve s adımları arasındaki fark olduğundan, 1.90 no.lu denklem 1.91 no.lu denklemde gösterildiği şekilde düzenlenir.

$$\mathbf{W}^2(s+1) = \mathbf{W}^2(s) + \alpha \cdot \delta^2 \cdot (\mathbf{y}^1)' \quad (1.91)$$

Daha önce belirtildiği gibi n^1 ve n^2 birinci ve ikinci katmanların nöron sayıları olmak üzere, $\delta^2 \cdot (\mathbf{y}^1)'$ vektörlerinin çarpımı, $\mathbf{W}^2$ matrisinin boyutuna eşit ($n^2 \times n^1$) boyutlu bir matris olacaktır. Sonuç olarak geri yayılım algoritmasında, ikinci (çıkış katmanından önceki) katmanın ağırlıklarının güncellenmesine işaret eden öğrenme süreci, 1.91 no.lu denklem üzerinden gerçekleşir.

1.7.1.2.2. Birinci Katmana İlişkin Sinaptik Ağırlıkların Analizi

İkinci katmanda uygulanan zincir kuralı, birinci katmanda yer alan nöronların sinaptik ağırlıklarının analiz edilmesi için 1.92 no.lu denklemde gösterildiği şekilde uyarlanır.

$$\nabla \mathbf{e}^1 = \frac{\partial \mathbf{e}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} \cdot \frac{\partial \mathbf{y}^1}{\partial \mathbf{a}^1} \cdot \frac{\partial \mathbf{a}^1}{\partial \mathbf{W}^1} \quad (1.92)$$

1.92 no.lu denklemle temsil edilen zincir kuralına göre üç kısımdan oluşan türev denklemi, 1.59, 1.62 ve 1.65 no.lu denklemlerden hareketle çözülür.

$$\frac{\partial \mathbf{a}^1}{\partial \mathbf{W}^1} = \mathbf{x} \quad (1.93)$$

$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{a}^1} = g'(\mathbf{a}^1) \quad (1.94)$$

Ancak 1.93 ve 1.94 no.lu denklemleri 1.92 no.lu zincir kuralına göre oluşturulan türev denkleminde yerine yazmadan önce, birinci katmanın türev denkleminin bir parçası olan $\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1}$ ifadesinin türevini almak amacıyla, ikinci katmanın girdi vektörü olan $\mathbf{a}^2$ ifadelerine göre (ikinci katmanın türevinde açıklandığı şekilde) 1.95 no.lu denklemde gösterildiği gibi yeni bir zincir kuralı oluşturulur.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^2} \cdot \frac{\partial \mathbf{a}^2}{\partial \mathbf{y}^1} \quad (1.95)$$

1.95 no.lu denklemin payında bulunan $\mathbf{a}^2$ ifadesi yerine, 1.60 no.lu denklemde $\mathbf{a}^2$ ifadesinin eşiti yazılarak 1.96 no.lu denklem elde edilir.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^2} \cdot \frac{\partial (\mathbf{W}^2 \mathbf{y}^1)}{\partial \mathbf{y}^1} \quad (1.96)$$

Bu aşamada 1.96 no.lu denklemin en sağındaki kısmın türevi alınabilir. Böylece 1.97 no.lu denkleme ulaşılır.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{a}^2} (\mathbf{W}^2) \quad (1.97)$$

1.97 no.lu denklem, ikinci katmanın sinaptik ağırlık analizinde yapılan çıkarıma paralel bir şekilde geri yayılıma ilişkin iki önemli olguya işaret etmektedir. Birincisi $\mathbf{W}^2$ ifadesi, ikinci katmanın ağırlık matrisidir ve ikinci katmanın tüm nöronları $\mathbf{W}^2$ üzerinden birinci katmanın j. nöronuyla bağlantılıdır. İkincisi ise $\mathbf{W}^2$ ifadesi, bir önceki (ikinci katmandaki) aşamada $\mathbf{W}^3$ ağırlık matrisi üzerinden dolaylı bir şekilde hata değerlerine göre güncellenmiş ağırlık matrisidir. Bu çerçevede $\mathbf{W}^2$ ağırlık matrisinin güncellenmesinde kullanılan dolaylı hata değerleri, $\mathbf{W}^2$ ağırlık matrisi üzerinden yine dolaylı (ikinci katmandaki sürece benzer) bir şekilde geriye, birinci katmana doğru yayılır.

1.88 no.lu denklem, $\mathbf{W}^2$ matrisinin devriğini sağdan çarpacak şekilde 1.97 no.lu denklemde yerine yazılarak 1.98 no.lu denklem elde edilir.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} = -(\mathbf{W}^2)' \delta^2 \quad (1.98)$$

1.93, 1.94 ve 1.98 no.lu denklemler, 1.92 no.lu zincir kuralına göre oluşturulan türev denkleminde yerlerine yazılarak 1.99 no.lu denkleme ulaşılır.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^1} = -(\mathbf{W}^2)' \delta^2 . g'(\mathbf{a}^1) . (\mathbf{x})' \quad (1.99)$$

1.99 no.lu denklemde yer alan $g'(\mathbf{a}^1)$ ifadesi, girdisi $\mathbf{a}^1$ vektörü olan birinci katmanın aktivasyon fonksiyonunun türevini; $(\mathbf{y}^1)'$ ifadesi ise birinci katmanın ağırlıklandırılmış girdi vektörünün devriğini temsil eder. $(\mathbf{W}^2)' \delta^2 . g'(\mathbf{a}^1)$ ifadesi $(n^1 \times 1)$ ve $(\mathbf{x})'$ ifadesi ise $(1 \times n)$ boyutlu birer vektördür. (Hagan ve diğerleri, 2014, s.366). Hata minimizasyonu amaçlandığı için 1.100 no.lu denklemde yer alan $\mathbf{W}^1$ ağırlık matrisinde değişimin yönü türevin (eğimin) tersi yönde (negatif), 1.101 no.lu denklemde gösterildiği gibi olacaktır.

$$\Delta \mathbf{W}^1 = -\alpha \frac{\partial \mathbf{e}}{\partial \mathbf{W}^1} \quad (1.100)$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^1} = \alpha (\mathbf{W}^2)' \delta^2 . g'(\mathbf{a}^1) . (\mathbf{x})' \quad (1.101)$$

δ^1 ifadesi, birinci katmanın j. nöronunun lokal gradyan vektörü olmak üzere, 1.92 no.lu denklemde yer alan zincir kuralından hareketle 1.102 no.lu denklem şeklinde de ifade edilebilir. 1.102 no.lu denklemde, 1.62 ve 1.98 no.lu denklemler yerine yazılarak 1.103 no.lu denklem elde edilir.

$$\delta^1 = -\frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} \cdot \frac{\partial \mathbf{y}^1}{\partial \mathbf{a}^1} \quad (1.102)$$

$$\delta^1 = (\mathbf{W}^2)' \delta^2 . g'(\mathbf{a}^1) \quad (1.103)$$

1.103 no.lu denklem, üç katmanlı bir yapay sinir ağında çıktı katmanında üretilen hataların ilk nöral katmana yayılımını ifade etmektedir. 1.103 no.lu $(n^1 \times 1)$ boyutlu δ^1 vektörü, 1.101 no.lu türev denkleminde yerine yazılarak 1.104 no.lu denklem elde edilir.

$$\Delta \mathbf{W}^1 = \alpha \cdot \delta^1 \cdot (\mathbf{x})' \quad (1.104)$$

s ifadesi adım indisi olmak üzere, $\Delta \mathbf{W}^1$ ifadesi s+1 ve s adımları arasındaki fark olduğundan, 1.104 no.lu denklem 1.105 no.lu denklemde gösterildiği şekilde düzenlenir.

$$\mathbf{W}^1(s+1) = \mathbf{W}^1(s) + \alpha \cdot \delta^1 \cdot (\mathbf{x})' \quad (1.105)$$

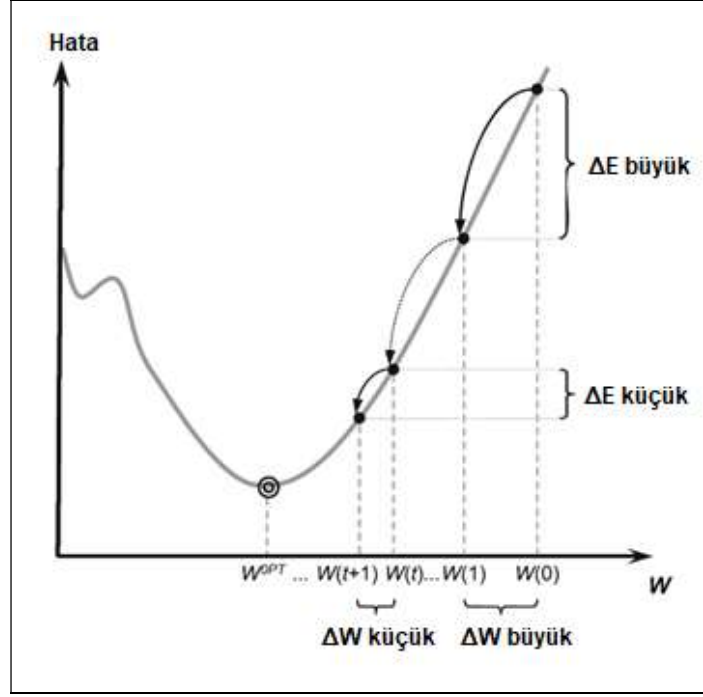
Daha önce belirtildiği gibi n ve n^1 girdi ve birinci katmanların nöron sayıları olmak üzere, $\delta^1 \cdot (\mathbf{x})'$ vektörlerinin çarpımı, $\mathbf{W}^1$ matrisinin boyutuna eşit ($n^1 \cdot n$) boyutlu bir matris olacaktır. Sonuç olarak geri yayılım algoritmasında, birinci (ikinci katmandan önceki) katmanın ağırlıklarının güncellenmesine işaret eden öğrenme süreci, 1.105 no.lu denklem üzerinden gerçekleşir.

1.7.1.3. Momentum Parametresi

Momentum parametresi (momentum parameter), s ve s+1 adımları arasında sinaptik ağırlık matrisinde gerçekleşen değişimin geri yayılım algoritmasına ilave edilmesidir. L ifadesi katmanları temsil etmek üzere, momentum parametresi 1.106 no.lu denklemde gösterildiği gibi geri yayılım öğrenme algoritmasına eklenir.

$$\mathbf{W}^L(s+1) = \mathbf{W}^L(s) + \lambda (\mathbf{W}^L(s) - \mathbf{W}^L(s-1)) + \alpha \cdot \delta^L \cdot (\mathbf{y}^{L-1})' \quad (1.106)$$

$\alpha \cdot \delta^L \cdot (\mathbf{y}^{L-1})'$ ifadesi öğrenme terimidir. 0 ile 1 arasında bir değere sahip λ ifadesi momentum parametresi olmak üzere, $\lambda (\mathbf{W}^L(s) - \mathbf{W}^L(s-1))$ ifadesi ise momentum terimidir. λ değeri 0 iken yukarıdaki denklem klasik geri yayılım algoritmasına dönüşür. λ değeri 0'dan farklı iken momentum terimi, ağırlık matrisinin yakınsama sürecine pozitif bir katkı sağlar (Silva ve diğerleri, 2017, s.71).



Grafik 1.6. Momentum Parametresi

Kaynak: Silva ve diğerleri, 2017, s.72

Grafik 1.6'da, ağırlık matrisinin s anındaki çözümü final çözümünden uzaksa, momentum teriminin ağırlık matrisi üzerindeki yakınsatma hızı artar. Diğer bir ifadeyle momentum terimi, ağırlık matrisini hata fonksiyonunun minimum noktasına daha hızlı ulaştırmaya çalışır. Ağırlık matrisinin s anındaki çözümü final çözüme yaklaştıkça, yani s ve $s-1$ anındaki hataların farkı azaldığı için momentum teriminin ağırlık matrisi üzerindeki yakınsatma hızı da yavaşlar. Öğrenme algoritmasına ilave edilen momentum parametresinin herhangi bir s anında öğrenme hızına katkısı $\frac{\alpha}{1-\lambda}$ kadardır (Silva ve diğerleri, 2017, s.72).

Momentum parametresi, sadece adım sayısını azaltmakla kalmaz, yakınsama adımlarını büyüttüğü için ağırlık matrisinin yerel minimumlara takılma olasılığını da düşürür (Fausett, 1994, s.306; Haykin, 2009, s.138). Diğer taraftan yüksek bir öğrenme katsayısının belirlenmesi durumunda, yakınsama süreci salınım (zikzak) yapacağından, hata fonksiyonunun minimum noktasına ulaşılamaz. Arzu edilmeyen bu salınımlardan kaçınmak amacıyla da momentum parametresi kullanılabilir (Kröse ve Smagt, 1996, s.37).

1.8. Yapay Sinir Ağlarında Performans İyileştirmesi

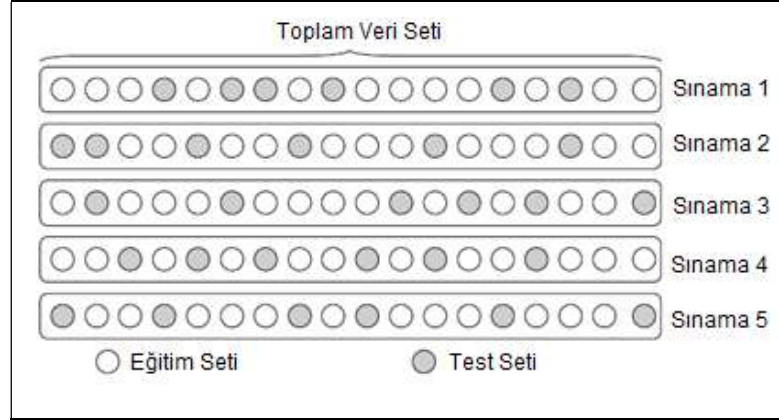
Belirli bir problemi çözmek için en uygun ağ topolojisi genellikle ampirik olarak belirlenir. Çünkü ağın tasarımı seçilen öğrenme algoritması, ağırlık matrisinin başlangıç değerleri, problemin karmaşıklığı, eğitim setinin dağılımı ve kalitesi gibi çok sayıda faktöre bağlıdır. Bir yapay sinir ağının performansını iyileştirmek için dikkat edilmesi gereken hususlar ve en yaygın uygulanan yöntemler takip eden alt başlıklarda ele alınmıştır.

1.8.1. Çapraz Doğrulama Yöntemleri

Çapraz doğrulama yöntemleri (cross-validation ya da holdout method), ağ topolojisinin belirlenmesinde en yaygın kullanılan istatistiksel yöntemlerden biridir. Yöntemin amacı, test seti dışında eğitim sürecinde kullanılan farklı bir doğrulama seti kullanılarak aday ağ topolojilerinin öğrenme performansını değerlendirmektir (Shwartz ve David, 2014, s.150; Haykin, 2009, s.171-172). Veri seti eğitim, doğrulama ve test seti olmak üzere üç farklı bölüme ayrılabilmesi gibi eğitim ve test seti olarak iki bölüme de ayrılabilir. Test setinin eğitim sürecinde kullanılmaması koşuluyla, çapraz doğrulama ayrı bir doğrulama seti yerine test seti kullanılarak da gerçekleştirilebilir (Silva ve diğerleri, 2017, s.97). Ağ topolojisinin seçiminde genellikle iki farklı çapraz doğrulama yöntemi kullanılır.

“Rassal örneklenen çapraz doğrulama” (random subsampling cross-validation) olarak adlandırılan ilk yöntemde tüm veri seti eğitim ve doğrulama seti olarak rassal bir şekilde ikiye bölünür. Ağın öğrenme sürecinde sadece eğitim seti kullanılır. Doğrulama veri seti ise öğrenme sürecinde kullanılmayan girdi ve beklenen çıktı değerlerine sahiptir. Aday olarak belirlenen her bir topolojinin doğrulama setinin girdilerini kullanarak ürettiği çıktılarla, doğrulama setinde bulunan beklenen çıktılar kıyaslanarak, diğer bir ifadeyle hatalar karşılaştırılarak performans değerlendirilmesi yapılır. Bu çerçevede en iyi performansı sunan ağ topolojisi, mevcut problemin çözümü için en uygun yapay sinir ağı olarak kabul edilir.

Şekil 1.13'te her bir sınamanın (trial) aynı 18 örneklemden oluştuğu 5 farklı sinama vardır. Ancak 12'si eğitim ve 6'sı doğrulama verisinden oluşan sınamaların her birinin eğitim ve doğrulama setleri rassal bir şekilde belirlenir. Tekrar eden her sınamada aday topoloji, öğrenme sürecinde eğitim setini, performans değerlendirmesinde ise doğrulama setini kullanır (Silva ve diğerleri, 2017, s.97).



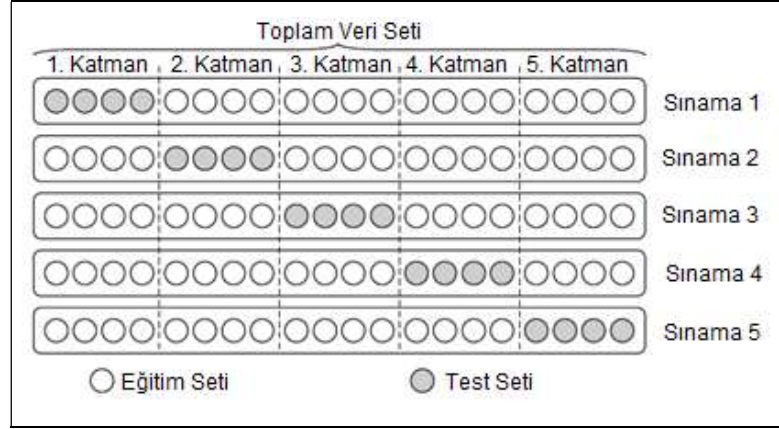
Şekil 1.13. Rassal Örneklene Çapraz Doğrulama

Kaynak: Silva ve diğerleri, 2017, s.98

Eğitim sürecinde hesaplanan ağırlık matrisi ve test setinin özellik/girdi vektörü kullanılarak ağırlık çıktıları elde edilir. Hesaplanan bu çıktıların, test/doğrulama setinin beklenen çıktılarına göre kıyaslanarak değerlendirilir. Aday topolojinin global performansı, sınamalar sonucunda elde edilen lokal performansların ortalaması, en büyüğü, en küçüğü gibi probleme özgü herhangi bir kritere göre belirlenebilir. Böylece aday topolojiler arasında en yüksek global performansa sahip ağ topolojisi seçilir.

İkinci yöntem ise “k-katmanlı/çok katmanlı çapraz doğrulama” (k-fold/multifold cross-validation) olarak adlandırılır. Bu yöntemde veri seti k adet katmana bölünür. k-1 katman eğitim seti, kalan 1 katman ise doğrulama seti olarak ayrılır. Öğrenme süreci, her katmandaki doğrulama seti hariç, tüm katmanlar üzerinde yineleyici bir şekilde devam eder (Haykin, 2009, s.175). Bu çerçevede 20 örneklemden oluşan bir veri seti, k=5 için 5 katman sayısına; k=4 için 4 katman sayısına sahip olacaktır. Uygulamada k değeri genellikle 5 ile 10 arasında seçilmektedir (Silva ve diğerleri, 2017, s.98).

Ağ performanslarının ölçülmesi için az sayıda veriye ihtiyaç duyan çok katmanlı çapraz doğrulama yöntemi, özellikle sınırlı sayıda veriyle çalışırken kullanılabilir (Shwartz ve David, 2014, s.149). İlk yöntemde olduğu gibi k-katmanlı çapraz doğrulama yönteminde de aday topolojiler arasında en yüksek global performansa sahip ağ topolojisi seçilir. Şekil 1.14'te 20 örneklemden oluşan ve k=5 katmana bölünmüş bir veri seti gösterilmektedir.



Şekil 1.14. Çok Katmanlı Çapraz Doğrulama

Kaynak: Silva ve diğerleri, 2017, s.99

Eğer $k=1$ ise çok katmanlı çapraz doğrulama yöntemi, “tek elemanlı çapraz doğrulama yöntemi” (leave-one-out cross-validation) olarak adlandırılan yöntemle dönüşür (Haykin, 2009, s.175).

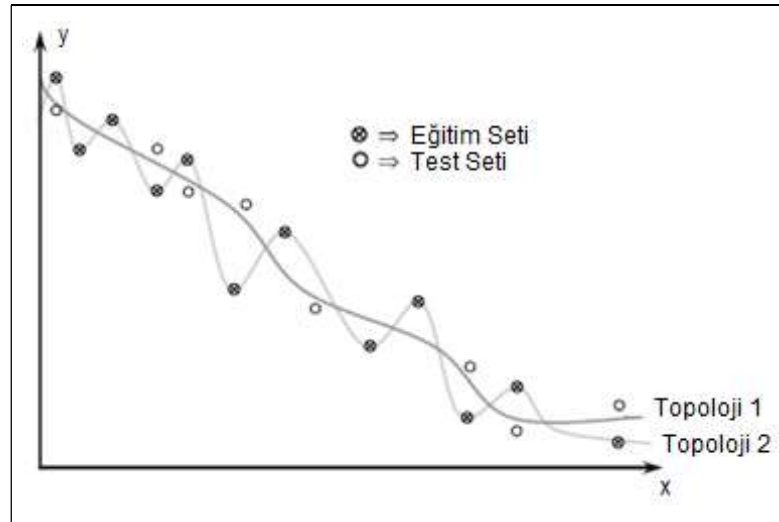
İki gizli katmana sahip aday MLP topolojisinde nöron sayılarını belirlemek için yukarıdaki yöntemden farklı olarak “Kolmogorov yöntemi” (Kolmogorov methods) de yaygın olarak kullanılır. Kolmogorov yöntemine göre n ifadesi girdi katmanında bulunan nöron sayısı (ya da girdi/bağımsız değişken sayısı) olmak üzere, birinci gizli katmanda $n(2n+1)$ ve ikinci gizli katmanda ise $(2n+1)$ adet nöron ile üç nöral katmanlı bir MLP tam olarak temsil edilebilir (Bishop, 1995, s.138). Benzer şekilde tek gizli katmana sahip aday MLP topolojisinde ise gizli katmanın nöron sayısı $(2n+1)$ olarak belirlenebilir (Silva ve diğerleri, 2017, s.100).

1.8.2. Aşırı Öğrenme ve Yetersiz Öğrenme Problemleri

Gizli katman ve/veya nöron sayılarının artırılması, ağ performansının test setinde iyi bir performans göstermesini garantilemez. Böyle bir durumda,

eğitim setinden örüntüleri ve/veya ilişkileri öğrenmek yerine, eğitim setindeki her bir veriyi ezberleyen ağı, eğitim seti için hata kareleri toplamı çok düşük, test seti için çok yüksek olacaktır. Arzu edilmeyen söz konusu olgu “aşırı öğrenme” ya da “ezberleme” (overfitting/memorization) olarak isimlendirilir. Bununla birlikte birçok problemin çözümünde genellikle tek gizli katmana sahip MLP topolojisinin yeterli olduğu kabul edilmektedir (Fausett, 1994, s.299).

Bir ağı öğrenmesini ifade eden genelleme yapabilme kabiliyeti, eğitim setindeki veri noktalarından daha az parametreye sahip olmasına dayanır. Bu yaklaşım “Ockham’ın usturası” (Ockham’s razor) kavramına işaret eder. Yapay sinir ağlarında eğitim setini yeterince temsil edebilen mümkün olan en basit ağı kullanılması tercih edilmelidir. Diğer bir ifadeyle bir problemi çözmek için basit bir ağ yeterliyse, karmaşık bir ağı kullanılması tercih edilmemelidir (Hagan ve diğerleri, 2014, s.380).



Grafik 1.7. Aşırı ve Normal Öğrenen MLP Topolojileri

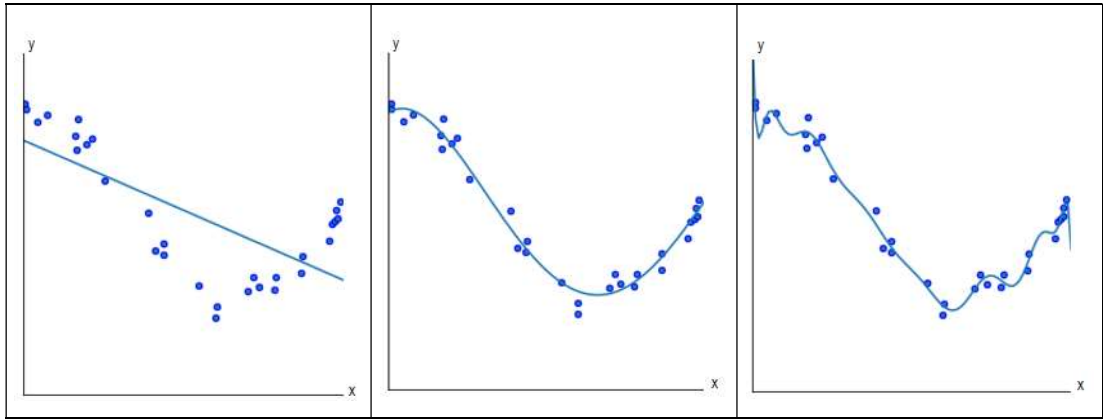
Kaynak: Silva ve diğerleri, 2017, s.102

Grafik 1.7’de 20 nöronlu oluşturulan varsayılan 2 no.lu topoloji ezberlemenin olduğu, 10 nöronlu oluşturulan varsayılan 1 no.lu topoloji ise ezberlemenin olmadığı senaryoyu temsil etmektedir. 2 no.lu topoloji, eğitim setinde bulunan tüm beklenen çıktıları eksiksiz bir şekilde doğru saptamasına rağmen, test setinde yer alan beklenen çıktıları tahmin etmede büyük ölçüde başarısızdır. Bu yüzden 2 no.lu topoloji, sapmalı (biased) bir tahmin edicidir. Diğer taraftan 1 no.lu topoloji, eğitim setinde bulunan beklenen çıktıları eksiksiz bir şekilde değil de yaklaşık olarak saptamasına rağmen, test setinde

yer alan beklenen çıktıları da makul sapmalarla tahmin etmede büyük ölçüde başarılıdır. Bu sebeple 1 no.lu topoloji, sapmasız(unbiased) bir tahmin edicidir ve 2 no.lu topolojiye tercih edilir (Silva ve diğerleri, 2017, s.102). Bu durum, 1 no.lu topolojinin genelleme yapabilme kabiliyetinin 2 no.lu topolojiye göre daha yüksek olduğunu ifade eder. Diğer bir ifadeyle 1 no.lu topolojinin test setine ilişkin hata kareleri toplamının 2 no.lu topolojiye göre daha düşük olduğuna işaret eder.

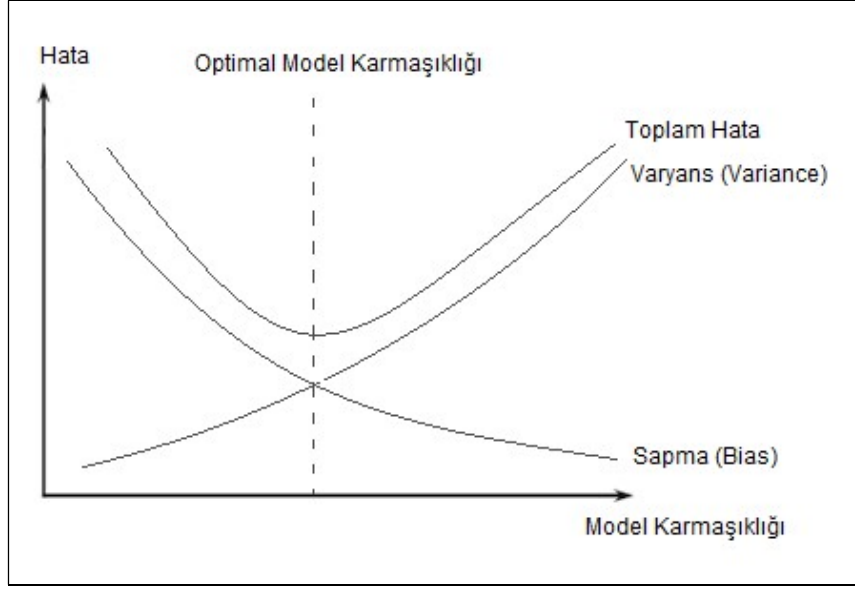
Az sayıda nöron ve/veya gizli katmandan oluşan bir MLP topolojisi ise örüntüleri ve/veya ilişkileri çıkarmakta yetersiz kalabilir. Böyle bir durumda, MLP'nin eğitim ve test setinin her ikisi için hata kareleri toplamı çok yüksek olacaktır. Arzu edilmeyen söz konusu olgu “yetersiz öğrenme” (underfitting) olarak isimlendirilir (Silva ve diğerleri, 2017, s.103).

Yukarıdaki açıklamalar çerçevesinde Grafik 1.8.a, b ve c’de noktalar veri setini, çizgiler ise alternatif modelleri temsil etmek üzere, yetersiz, normal ve aşırı öğrenme olguları görselleştirilmiştir.



Grafik 1.8.a. Yetersiz Öğrenme Grafik 1.8.b. Normal Öğrenme Grafik 1.8.c. Aşırı Öğrenme

Aşırı öğrenen ağların ağırlıklarının varyansı yüksek, sapması (bias) düşük ve hataları yüksek olur. Benzer şekilde yetersiz öğrenen ağların ağırlıklarının varyansı düşük, sapması (bias) yüksek ve hataları yine yüksek gerçekleşir. Ağırlıkların varyansı azaltıldıkça, sapması; sapması azaltıldıkça da varyansı artar. Dolayısıyla ağırlıkların varyansı ile sapması arasında bir değiş-tokuş (trade-off) vardır. Bu açıklamalar doğrultusunda Grafik 1.9’da eğitim setinin hatasına ilişkin varyans ve sapma arasındaki değiş-tokuş görselleştirilmiştir.

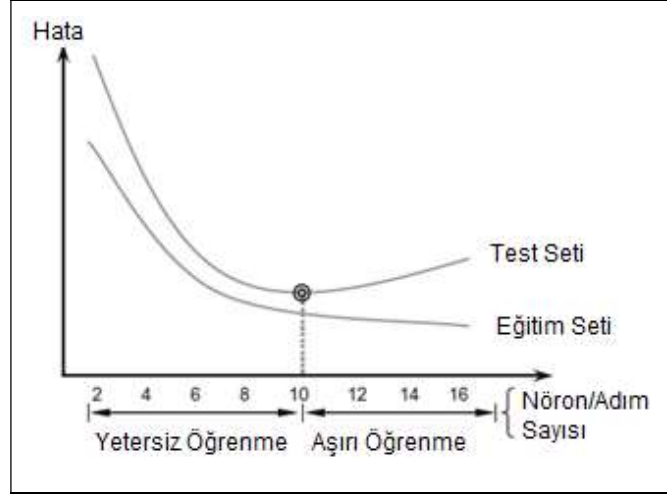


Grafik 1.9. Varyans ve Sapma Arasındaki Değiş-Tokuş

Örneklem sayısını arttırmak, varyansı ve sapmayı birlikte azaltmanın bir yoludur. Bununla birlikte örneklem sayısı sonsuza giderken varyans ile sapmanın her ikisinin de azalabildiği belirli bir sınır vardır. (Bishop, 1995, s.337). Grafik 1.9’da optimal model karmaşıklığının solundaki alan yetersiz öğrenmeyi, sağındaki alan ise aşırı öğrenmeyi yansıtır. Optimal model karmaşıklığı ise varyansın ve sapmanın ikisinin bir arada minimize edildiği noktayı temsil etmektedir. Bu çerçevede ağırlıkların varyansı ile sapmasını eş anlı olarak minimize etmede “erken durdurma kriteri” (early stopping) yöntemi yaygın olarak kullanılır.

1.8.3. Erken Durdurma Kriteri

Öğrenme algoritmaları teorik olarak genellikle hata karelerini minimize etmeye odaklanmış olsa da yapay sinir ağlarında mümkün olan en iyi performansa ulaşmak için eğitimi sonlandırma kriteri olarak hata kareleri toplamının minimize edilmesi amaçlanmaz. Eğitim setinde hata kareleri toplamının minimize edilmesi, yukarıda açıklandığı üzere ezberleme (memorization) problemine yol açar (Fausett, 1994, s.298). Bu sebeple regresyon analizinin aksine yapay sinir ağlarında (eğitim seti için) hata kareleri toplamının minimizasyonu değil, optimizasyonu amaçlanır. Bu çerçevede “erken durdurma” (early stopping) kriteri, ağırlıkların varyansı ile sapmasını minimum yapan optimum hata kareleri toplamı olarak tanımlanabilir.



Grafik 1.10. Erken Durdurma Kriteri

Kaynak: Silva ve diğerleri, 2017, s.104

Aşırı öğrenme ve yetersiz öğrenme problemi, gizli katman ve nöron sayısının yanı sıra, hata kareleri toplamının minimize edilmesi (ya da öğrenme algoritması) sürecinde yer alan yineleme/adım (epoch) sayısından da kaynaklanabilir. Grafik 1.10'da, az sayıda adım/nöron ile eğitilen modelin yetersiz öğrenme, çok sayıda adım/nöron ile eğitilen modelin ise aşırı öğrenme problemlerine maruz kalması gösterilmektedir. Buna göre öğrenme sürecinin 10'dan az adım/nöron sayısı ile gerçekleştirildiği durum, eğitim ve test setine ilişkin hataların minimize edilmediğine işaret etmektedir. Öğrenme sürecinin 10'dan fazla adım/nöron sayısı ile gerçekleştirildiği durum, eğitim setine ilişkin hataların az, test setine ilişkin hataların ise fazla olduğuna işaret etmektedir (Silva ve diğerleri, 2017, s.104). Benzer şekilde 10'dan az adım/nöron sayısının bulunduğu bölgede ağırlıkların varyansı düşük, sapması yüksek; 10'dan fazla adım/nöron sayısının bulunduğu bölgede ise ağırlıkların varyansı düşük, sapması yüksek olacaktır.

Yukarıdaki açıklamalar adım sayısı açısından değerlendirilecek olursa eğitim ve test setinin hataları azaldığı sürece aşırı öğrenme sürdürülür. Ancak eğitim setinin hataları azalmaya devam etmesine rağmen, test setinin hataları artmaya başlarsa öğrenme süreci sonlandırılır (Fausett, 1994, s.298). Ancak yukarıdaki şekilde doğrulama/test seti hataları, tek minimuma sahip bir eğri olarak ifade edilmesine rağmen, uygulamada birden fazla yerel minimum içerebilir. Yerel minimumlardan kaçınmak amacıyla, adım sayısının bir miktar

arttırılarak aşırı öğrenme alanına doğru uzanan daha yavaş bir erken durdurma kriteri kullanılabilir (Haykin, 2009, s.174).

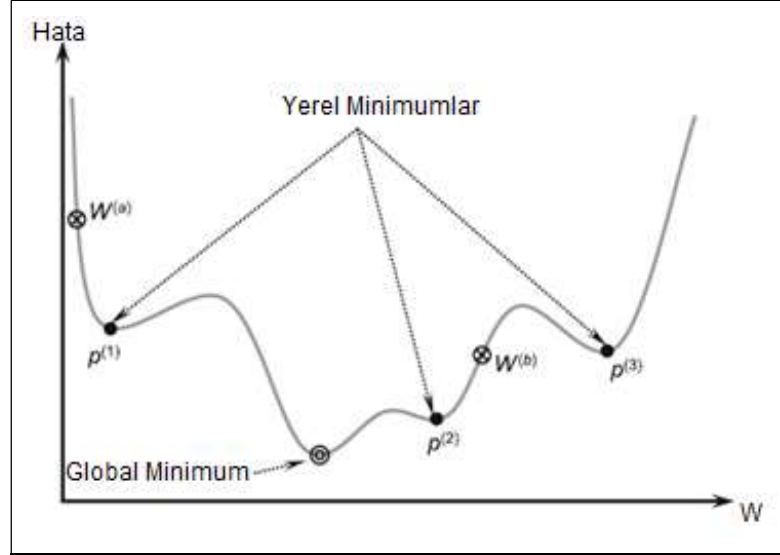
Diğer taraftan adım sayısında olduğu gibi gizli katmanların nöron sayısının belirli bir eşiğe kadar arttırılması da test ve eğitim setlerinin hatalarını düşürür. Ancak gizli katmanlardaki nöron sayısının arttırma limiti aşılsa, eğitim setinin hatası azalmaya devam etmesine rağmen, test setinin hataları ise artmaya başlar. Böyle bir durumda ağ, genelleme yapabilme kabiliyetini kaybederek verideki gürültüye uyum (fitting noise) göstermeye, diğer bir ifadeyle aşırı öğrenmeye (ezberlemeye) maruz kalmaya başlar (Kröse ve Smagt, 1996, s.44-45).

Bu yüzden öğrenme sürecinin 10 adım/nöron ile gerçekleştirildiği temsili durum eğitim setine ilişkin hataların minimize edilmediğine işaret etmesine rağmen, ağırlıkların varyansı ile sapmasını minimum yapan optimal adım/nöron sayısını ifade etmektedir. Benzer şekilde söz konusu temsili durum, test setine ilişkin hataların minimize edilmesi sebebiyle aşırı ya da yetersiz öğrenme senaryosunun gerçekleşmediği optimal adım/nöron sayısı olarak da değerlendirilebilir.

Alternatif durdurma kriteri olarak iki adım (epoch) arasında ortalama hata karelerinin (mean square error, MSE) belirli bir değerin altına düşmesi durumunda eğitim sonlandırılması önerilebilir. Ancak hata karelerinin belirli bir değerin altına düştüğü seviyenin yerel minimuma karşılık gelebileceği göz önünde bulundurulmalıdır. Aksi durumda ağırlık matrisi optimize edilemeyeceğinden yetersiz öğrenme problemi ortaya çıkar (Gurney, 1997, s.106).

1.8.4. Yerel Minimuma Yakınsama Problemi

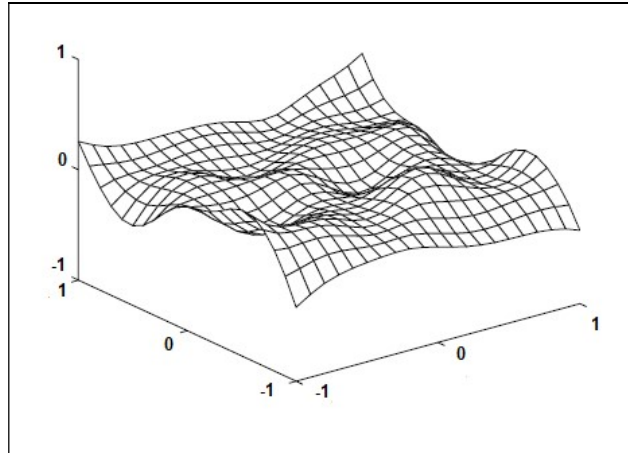
MLP tarafından üretilen hatalar doğrusal olmadığı için öğrenme sürecinde, ağırlık matrisinin arzu edilmeyen bir yerel minimumda belirlenmesi söz konusu olabilir.



Grafik 1.11. Yerel Minimuma Yakınsama Problemi

Kaynak: Silva ve diğerleri, 2017, s.105

Grafik 1.11'de W^a , W^b ve W^c noktaları ağırlık matrisinin başlangıç değerlerini; p^1 , p^2 ve p^3 noktaları ise başlangıç noktalarına karşılık gelen yerel minimumları ifade etmektedir. Eğer ağırlık matrisinin başlangıç değeri W^a noktası ise ağırlık matrisi p^1 , başlangıç değeri W^b noktası ise ağırlık matrisi p^2 ve başlangıç değeri W^c noktası ise ağırlık matrisi p^3 yerel minimumunda belirlenecektir. Ancak p^1 , p^2 ve p^3 noktalarından hiçbirisi doğrusal olmayan hata fonksiyonunun global minimumuna karşılık gelmemektedir.



Grafik 1.12. Gerçek Bir Hata Fonksiyonu Örneği

Kaynak: Kröse ve Smagt, 1996, s.38

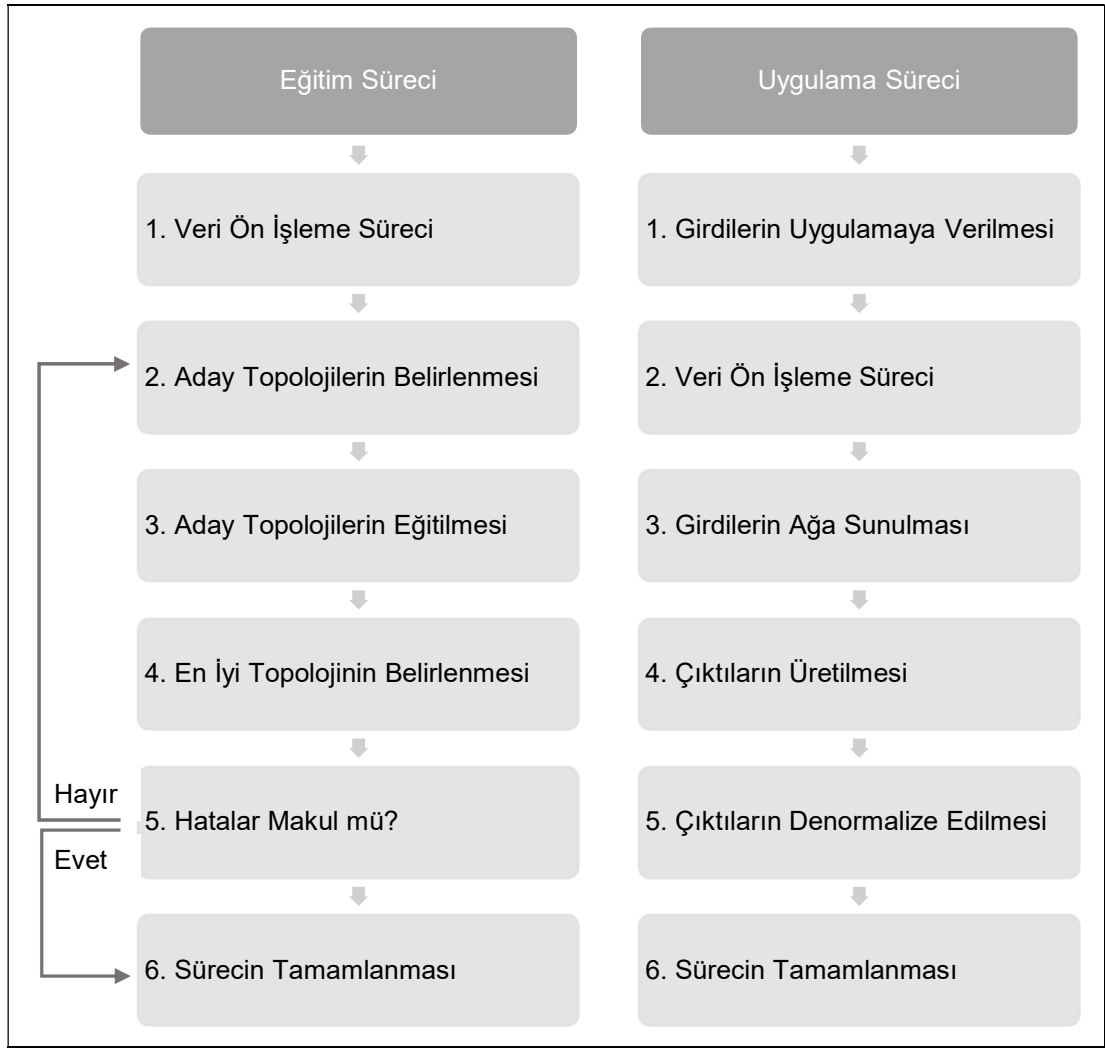
Grafik 1.12'de iki boyutlu girdiye ve tek boyutlu bir çıktıya, her birinde sigmoid aktivasyon fonksiyonunun kullanıldığı 10 gizli nörona ve çıktı katmanında doğrusal aktivasyon fonksiyonuna sahip bir MLP ağına ilişkin hata

fonksiyonunun yüzeyi gösterilmektedir. İki boyutlu bir girdiye sahip ağın eşik değeri ile birlikte üç parametresi olduğundan hata fonksiyonu üç boyutlu düzlemde gösterilebilmiştir. Grafik 1.12’de görüldüğü gibi hata fonksiyonu çok sayıda yerel minimuma sahiptir.

Bir ağın yerel minimuma yakınsama probleminden kaçınmak için ağırlık matrisi birden fazla kez (rassal olarak belirlenen) farklı başlangıç değerleriyle eğitilmelidir. Eğitimin sonucunda en az hatayı üreten ağırlık matrisinin başlangıç değeri, global minimuma işaret eder (Silva ve diğerleri, 2017, s.105). Bazen eğitime karşı yukarı yönde adımlara izin vermek ya da öğrenme algoritmasında momentum parametresine yer vermek yerel minimumlara takılmaktan kaçınmak için başvurulacak yöntemler arasındadır (Gurney, 1997, s.104). Diğer bir alternatif ise gizli katmanlarda bulunan nöron sayısının artırılması olabilir. Ancak gizli katmanlarda nöron sayısının artırılmasının da bir sınırı olduğu, sınırın aşılmasının yerel minimuma yakalanma olasılığını arttıracak göz önünde bulundurulmalıdır (Kröse ve Smagt, 1996, s.40).

1.9. Yapay Sinir Ağı Tasarımı

Aşağıda yer alan Şekil 1.15’te MLP tasarımına ilişkin eğitim ve uygulama aşamaları gösterilmektedir.



Şekil 1.15. MLP Eğitim ve Uygulama Aşamaları

Kaynak: Silva ve diğerleri, 2017, s.106

Şekil 1.15'te belirtilen eğitim ve doğrulama veri setini kullanan eğitim süreci özetle aşağıdaki aşamalardan oluşur.

1. Veri ön işleme aşaması eksik, hatalı, anomali vb. uygun olmayan verilerin tespit edilip düzeltilmesi, işlenmemiş verilerin görselleştirilmesi gibi keşifsel veri analizi (exploratory data analysis, EDA) ve verilerin 0 ile 1 arasında değerlere indirgenmesi anlamına gelen normalizasyon işlemi gerçekleştirilir. Z-skor ve max-min yöntemleri en yaygın kullanılan normalizasyon yöntemlerindendir.

$$x^n = \frac{x - \mu}{\sigma} \quad (1.107)$$

Aritmetik ortalaması μ , standart sapması σ olan bir x serisi, 1.107 no.lu denklemde gösterildiği gibi z-skor yöntemiyle normalize edilebilir. x^n normalize (ya da standardize) edilen seriyi ifade etmektedir.

$$x^n = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1.108)$$

1.108 no.lu denklemde gösterildiği gibi max-min normalizasyon yöntemine göre bir x serisinin her bir değerinden onun minimum değeri çıkartılarak elde edilen sonuç, maksimum ve minimum değerleri arasındaki farka bölünür. Max-min yöntemi, bir serinin değerlerini 0 ile 1 ya da -1 ile 1 arasında değerlere indirger (Aksu ve diğerleri, 2019).

2. Aday topolojilerin belirlenmesi aşamasında ağda yer alacak katman ve gizli nöron sayıları, kullanılacak aktivasyon fonksiyonları, çıktı katmanında yer alacak nöron sayısı gibi eğitilecek ağların topolojik özellikleri belirlenir.

3. Belirlenen topolojiler, veri ön işleme sürecinde oluşturulmuş girdi ve beklenen çıktı vektörleri aracılığıyla eğitilir.

4. Eğitim sonucunda aday ağ topolojilerinin performansları karşılaştırılır. Aday topolojiler arasında en iyi performansa sahip topoloji seçilir.

5. Aday topolojiler arasında en iyi performansa sahip topolojinin hatalarının kabul edilebilir olup olmadığı değerlendirilir. Hataların yüksek olduğu değerlendirilirse ikinci aşamaya geri dönülerek süreç tekrarlanır.

6. Eğer hataların makul olduğu değerlendirilirse ağın eğitimi tamamlanmış kabul edilir.

Şekil 1.26'da belirtilen test ya da gerçek girdileri kullanan uygulama süreci ise özetle aşağıdaki aşamalardan oluşur.

1. Eğitim sürecinin sonunda tasarlanan uygulamaya işlenmemiş girdiler sunulur.

2. Girdiler, normalizasyon (ya da standardizasyon) gibi veri ön işleme sürecine sokulur.

3. Veri ön işleme sürecinden geçmiş girdiler ağı sunulur. Bu aşamada beklenen çıktılara ilişkin herhangi bir veri seti yoktur.

4. Ağ, normalize edilmiş girdilere göre çıktıları hesaplar.

5. Girdiler normalize edildiği için çıktılar da normalize edilmiş değerlerde üretilir. Bu yüzden çıktılar denormalize edilir. Diğer bir ifadeyle çıktılar, kullanıcının yorumlayabileceği normal sayısal ya da kategorik değerlere dönüştürülür.

6. Eğitilen ağın, daha önce karşılaşmadığı girdilere göre ürettiği çözümler kullanıcıya sunulur.

MLP ağlarının eğitim sürecinde dikkat edilmesi gereken hususlar aşağıda özetlenmiştir.

- Katman veya gizli nöron sayısını arttırmak, ağın öğrenme (genelleme yapabilme) kapasitesini her zaman arttırmaz. Gizli nöron sayısı belirli bir sınırı aşarsa, “aşırı öğrenme/ezberleme” (overfitting/memorization) probleminden dolayı ağın öğrenme kapasitesi düşer. Benzer şekilde aynı genelleme yeteneğine sahip iki MLP topolojisinden, verideki gürültülere (noise) daha az duyarlı olduğu ve daha fazla bilgiyi çıkarabildiği için nöron sayısı az olan topoloji tercih edilir. Diğer bir ifadeyle Ockham’ın usturası yaklaşımıyla, aynı performansa sahip mimariler ya da topolojiler arasında daha basit olan ağ tercih edilir. Topolojinin genelleme yeteneğini değerlendirmek için daima test seti kullanılır.
- Veri seti dalgalanmaların daha az olduğu pürüzsüz (smooth) bir fonksiyon özelliği gösteriyorsa daha az sayıda gizli nöron, dalgalanmaların daha fazla olduğu bir fonksiyon özelliği gösteriyorsa da daha fazla sayıda nöron kullanılabilir (Kröse ve Smagt, 1996, s.43).
- Öğrenme sürecinde yerel minimumdan kaçınmak için ağırlık matrisinin başlangıç değerleri, rassal ve yinelemeli bir şekilde seçilir; öğrenme algoritmasına, 0 ile 1 arasında bir değere sahip

momentum parametresi (λ) dahil edilir. Hata fonksiyonu genellikle engebeli bir yüzeye sahip olduğundan, yerel minimuma yakalanmamak amacıyla eğime karşı yukarı yönde adımlara tolerans tanınabilir. Bir ölçüye kadar gizli nöron sayıları artırılarak da yerel minimuma yakalanma olasılığı düşürülebilir.

- Aktivasyon fonksiyonunun doygunluk bölgesinden kaçınmak için girdi ve çıktı örneklemeleri normalize edilir ve ağırlık matrisinin başlangıç değerleri, küçük değerler arasından seçilir.
- Gizli katmanlarda aktivasyon fonksiyonu olarak hiperbolik tanjant fonksiyonu kullanılabilir. Çünkü hiperbolik tanjant fonksiyonu, girdisinin işareti değiştiğinde çıktı değeri yerine işareti değişen bir fonksiyon (odd function) olduğundan, ağırlık yakınsama sürecini iyileştirir (Silva ve diğerleri, 2017, s.109). Benzer şekilde probleme göre değişmekle birlikte ReLU ve SELU aktivasyon fonksiyonları da (ya da bunların türevleri de) öncelikli olarak tercih edilebilir.
- Hata fonksiyonunun minimum noktasına ulaşılmasını engelleyen, sürekli salınımlara sebep olabilecek yüksek bir α öğrenme parametresi belirlemekten kaçınılmalıdır. Benzer şekilde yerel minimumlara takılabilecek ya da öğrenme sürecini uzatabilecek seviyede küçük bir α öğrenme parametresi belirlenmemelidir. α öğrenme parametresi 0'dan büyük 1'den küçük olacak şekilde deneysel yöntemlerle seçilebilir.
- Eğitim setinde, veri setindeki maksimum ve minimum değerlere sahip örneklemelerin bulunmasına özen gösterilir. Örneklem sayısı arttıkça ağırlık genelleme yapabilme kabiliyeti arttığından, ağırlık eğitimi mümkün olduğu kadar geniş örneklemle gerçekleştirilmelidir.
- Girdilerin karmaşıklık boyutunu indirmek ve fazlalıkları azaltmak için veri ön işleme tekniklerine ve/veya (Fourier dönüşümü, temel bileşen analizi, Wavelet dönüşümü gibi) özellik çıkarımı araçlarına başvurulur (Silva ve diğerleri, 2017, s.109).

- MLP eğitim sürecinde, adım (epoch) durdurma kriteri olarak belirli bir hata değeri (precision error) yerine maksimum adım sayısı belirlenebilir. Eğitim ve test setinin hataları birlikte azaldığı sürece, diğer bir ifadeyle ağırlık varyansı ve sapması birlikte azaldığı sürece eğitime devam edilir. Test setinin hataları artmaya, ağırlık varyansı ile sapması ayrılmaya başladığında ise eğitim sonlandırılır.

İKİNCİ BÖLÜM

ZAMAN SERİLERİ ANALİZİ

Zaman serileri, bir değişkenin günlük, aylık, yıllık vb. sıklıklarda zaman içinde gözlemlenen kesikli değerleri olarak tanımlanabilir (Hill ve diğerleri, 2011, s.7). Zaman serileri analizinin en önemli amaçlarından biri öngörü yapmaktır. Öngörü (forecasting) ise genellikle stratejik planlamaya odaklanan uzun dönemli öngörü; geçmiş verilerden paternlerin ortaya çıkarılmasına, modellemeye ve tanımlamaya dayanan orta ve kısa dönemli öngörü olarak sınıflandırılır. İstatistiksel metotlar orta ve kısa dönemli öngörüler için oldukça elverişlidir. Öngörü, karar verme ve planlama süreçleri için kritik bir girdidir (Montgomery ve diğerleri, 2008, s.1-2). Bu çalışmada orta ve kısa dönemli öngörülere odaklanılmaktadır.

Zaman serileri analizi temel olarak tek değişkenli (univariate) ve çok değişkenli (multivariate) modeller olarak incelenir. Tek değişkenli (univariate) modeller, sadece serinin geçmiş değerlerinde ve/veya bunlardan türetilen hata terimlerinin mevcut ve geçmiş dönem değerlerinde bulunan bilgileri kullanarak modelleme ve tahminleme yapmaya çalışır. Çok değişkenli (multivariate) modeller ise bir değişkendeki değişimi başka bir değişkenin mevcut ya da geçmiş değerlerinde bulunan bilgiyi kullanarak açıklamaya çalışır (Brooks, 2008, s.206-207). Tek değişkenli modellerde, y değişkeni dışında x gibi ilave bir açıklayıcı değişken kullanılmazken çok değişkenli modellerde ise y değişkeni dışında x gibi ilave bir açıklayıcı değişken ya da değişkenler kullanılır.

2.1. Durağanlık ve Birim Kök Analizi

Ekonometrik ve istatistiksel çalışmalarda yoğun olarak kullanılan zaman serileri modellerinin, klasik doğrusal regresyon modellerine kıyasla, kendine özgü zorlukları vardır. Zaman serileri üzerine yapılan uygulamalı

çalıřmalarda, seriler genellikle durađan olmamakla birlikte serilerin durađanlık (stationary) kořullarını sađlaması beklenir.

Zaman serileri, stokastik/rassal sũreçlerden elde edilen veri setleridir. Stokastik sũreç ise zaman içinde sıralanmıř rastgele deđiřkenlerin bir koleksiyonudur (Gujarati ve Porter, 2009, s.740). Stokastik sũrecin özel bir tũrũ olan gũçlü durađanlık (strictly stationary), zaman serilerinin gözlemlenen deđerlerine iliřkin ortak olasılık dađılımının (joint probability distribution) zaman içinde sabit kalmasıdır. Diđer bir ifadeyle gũçlü durađanlık, bir zaman serisinin ortalamasının (beklenen deđerinin), varyansının ve kovaryansının gecikme uzunluđundan bađımsız bir řekilde zaman içinde sabit kalmasına iřaret eder (Box ve diđerleri, 2016, s.22-24). Uygulamada genellikle gũçlü durađanlık kořulları sađlanamaz. Bu yũzden durađanlık, “zayıf durađanlık” (weakly stationary) ya da “kovaryans durađan” (covariance stationary) kavramı çerçevesinde analiz edilir. Zayıf durađanlık ise bir zaman serisinin ortalaması (beklenen deđer) ile varyansının zaman içinde sabit, iki dũnem arasındaki kovaryansının ise hesaplandıđı dũneme deđil, dũnemler arasındaki uzaklıđa (gecikme uzunluđuna) gũre deđerımesi olarak tanımlanır (Gujarati ve Porter, 2009, s.740).

$$\text{Ortalama} \quad : E(y_t) = \mu \quad (2.1)$$

$$\text{Varyans} \quad : \text{Var}(y_t) = E(y_t - \mu)^2 = \sigma^2 \quad (2.2)$$

$$\text{Kovaryans} \quad : \text{Cov}(y_t, y_{t+k}) = \gamma_k = E[(y_t - \mu)(y_{t+k} - \mu)] \quad (2.3)$$

Gecikme uzunluđu k ve zamana gũre gũzlem sırası t olmak ũzere, 2.1, 2.2 ve 2.3 no.lu denklemlerde, y_t deđerıkeninin zamana gũre sabit olan beklenen deđerini μ , varyansını σ^2 ve zamandan bađımsız gecikme uzunluđuna gũre deđerıřen kovaryansını γ_k temsil etmektedir.

Bir zaman serisinin beklenen deđerinin ve varyansının zamana gũre deđerımesi, serinin durađan olmadıđını ifade eder (Gujarati ve Porter, 2009, s.740). Durađan olmayan zaman serilerinde, deđerıkenler arasında anlamlı bir iliřki olmasa da R^2 deđerı ve t istatistikleri oldukça yũksek ıkar. Deđerıkenler

arasında sahte bir ilişkiye işaret eden söz konusu durum “sahte regresyon” (spurious regression) sorunu olarak adlandırılır (Hill ve diğerleri, 2011, s.482-483). Bu yüzden zaman serilerinde durağanlığın araştırılması, tahminlerin güvenilirliği bakımından son derece kritik kabul edilir. Durağanlık, genellikle “rassal yürüyüş modeli” (random walk model) üzerinden incelenir. Rassal yürüyüş modeli, sabit veya trend içerip içermemesine göre temel olarak üç kategoride incelenebilir.

2.1.1. Saf Rassal Yürüyüş Modeli

Bir rassal değişkenin kendisinden önceki değeri ile problemsiz hata teriminden oluşan model, rassal yürüyüş modeli (random walk model) olarak tanımlanır. Rassal salınımlar gösteren ve gerçek bir patern içermeyen rassal yürüyüş modeli durağan olmayan bir karakteristiğe sahiptir (Hill ve diğerleri, 2011, s.480). Sabit içermeyen rassal yürüyüş modeli “saf rassal yürüyüş modeli” olarak da adlandırılır (Gujarati ve Porter, 2009, s.768).

$$y_t = \rho y_{t-1} + \varepsilon_t \quad (2.4)$$

Esasen 2.4 no.lu denklem AR(1) sürecidir. Bununla birlikte $\rho=1$ iken AR(1) süreci rassal yürüyüş modeline dönüşür. AR(1) süreci ile rassal yürüyüş modelinin bir arada değerlendirilmesinin, durağanlığın daha kompakt bir şekilde analiz edilmesine olanak tanıdığı düşünülmektedir. Dolayısıyla bu çalışmada rassal yürüyüş modelinin tüm türlerinde söz konusu kompakt yaklaşım kabul edilmiştir. Bu açıklamalar göz önünde bulundurulmak koşuluyla, 2.4 no.lu denklem $t=1$, $t=2$ ve $t=3$ iken 2.5, 2.6 ve 2.7 no.lu denklemlerde gösterildiği gibi ifade edilebilir.

$$y_1 = \rho y_0 + \varepsilon_1 \quad (2.5)$$

$$y_2 = \rho y_1 + \varepsilon_2 \quad (2.6)$$

$$y_3 = \rho y_2 + \varepsilon_3 \quad (2.7)$$

Yukarıdaki denklem sisteminde y_1 'in eşiti, y_2 denklemi içine ve y_2 'nin bir önceki aşamada elde edilen eşiti de y_3 denklemi içine yazılarak 2.8, 2.9 ve 2.10 no.lu denklemlere ulaşılır.

$$y_2 = \rho(\rho y_0 + \varepsilon_1) + \varepsilon_2 \quad (2.8)$$

$$y_3 = \rho(\rho(\rho y_0 + \varepsilon_1) + \varepsilon_2) + \varepsilon_3 \quad (2.9)$$

$$y_3 = \rho^3 y_0 + \rho^2 \varepsilon_1 + \rho \varepsilon_2 + \varepsilon_3 \quad (2.10)$$

Yukarıdaki süreç gözlem sayısını temsil eden T kez tekrarlanırsa 2.11 no.lu denklem elde edilir.

$$y_t = \rho^T y_0 + \left(\sum_{t=1}^T \rho^{t-1} \varepsilon_{T-t+1} \right) \quad (2.11)$$

2.11 no.lu denklemde $\rho > 1$ durumu, $\rho^3 > \rho^2 > \rho$ olduğundan şokların etkisinin zaman içinde giderek arttığına işaret eder. Bu süreç “patlayan durum” (explosive case) olarak bilinir. $0 < \rho < 1$ durumu ise $\rho^3 < \rho^2 < \rho$ olduğundan şokların etkisinin zaman içinde giderek azaldığını ifade eder. $\rho = 1$ durumu ise seride “birim kök” (unit root) olduğuna işaret eder. Eğer $\rho = 1$ ise yukarıdaki süreç, y_t serisinin başlangıç değeri ile geçmiş dönem hata terimlerinin birikimli bir fonksiyonu olacaktır (Brooks, 2008, s.322).

$$y_t = y_0 + \left(\sum_{t=1}^T \varepsilon_t \right) \quad (2.12)$$

2.12 no.lu denklemde gösterildiği gibi $\rho = 1$ iken, diğer bir ifadeyle serinin birim kökü varken, y_t serisi tamamen rassal şoklar etkisi altındadır ve deterministik bir trende sahip değildir. Rassal şokların seri üzerindeki etkisinin zamanla kaybolmaması “sonsuz bellek” (infinite memory), hata terimlerinin toplamından oluşan trend de “stokastik trend” (stochastic trend) olarak adlandırılır (Gujarati ve Porter, 2009, s.742). y_t serisinin beklenen değerini elde etmek için 2.12 no.lu denkleme beklenen değer operatörü ilave edilerek 2.13 no.lu denkleme ulaşılır.

$$E[y_t] = E[y_0] + E\left[\left(\sum_{t=1}^T \varepsilon_t\right)\right] \quad (2.13)$$

Problemsiz hata terimlerini temsil eden $\varepsilon_1, \varepsilon_2, \varepsilon_3$ gibi ifadelerin beklenen değeri 0 değerine eşittir (Brooks, 2008, s.44). Başlangıç değerini ifade eden y_0 bir sabit ve bir sabitin beklenen değeri de kendisine eşit olduğundan, y_t serisinin beklenen değeri 2.14 no.lu denklemde gösterildiği gibi başlangıç değerine eşit olacaktır.

$$\mu = E[y_t] = y_0 \quad (2.14)$$

2.14 no.lu denklemin sonuca göre birim kök varken, saf rassal yürüyüş modelinde serinin beklenen değeri sabittir. Diğer taraftan $-1 < \rho < 1$ iken y_t serisinin beklenen değeri 2.11 no.lu eşitlikten türetilerek 2.15 no.lu denklem elde edilir.

$$\mu = \rho^t y_0 \quad (2.15)$$

Başlangıç değerini ifade eden y_0 değeri, birim kökün yokluğunda y_t serisini çok uzak bir geçmişte etkilediğinden genellikle ihmal edilerek 0 kabul edilir (Hill ve diğerleri, 2011, s.480). Bu çerçevede 2.16 no.lu denklemde ifade edildiği gibi serinin beklenen değerinin 0 olduğu kabul edilebilir.

$$\mu = 0 \quad (2.16)$$

Bu sonuç birim kök yokken, saf rassal yürüyüş modelinde serinin beklenen değerinin 0, diğer bir ifadeyle sabit olduğuna işaret eder.

Diğer taraftan serinin varyansı, $\rho = 1$ iken (birim kök varken) zamanın bir fonksiyonu olacaktır.

$$\text{Var}(y_t) = t\delta_\varepsilon^2 \quad (2.17)$$

2.17 no.lu denklemde yer alan δ_ε^2 ifadesi ise problemsiz hata teriminin varyansı olmak üzere, doğrusal regresyon analizinde y_t serisinin varyansı, hata teriminin varyansına eşit olduğu için 2.17 no.lu eşitlikte de y_t serisinin varyansı hata teriminin varyansına eşit olarak ifade edildi (Gujarati ve Porter, 2009, s.70). Birim kök varken y_t serisinin varyansının zamana göre sabit olmaması, onun durağan olmadığını ifade eder. Diğer taraftan birim kökün bulunmadığını temsil eden $-1 < \rho < 1$ durumunda, y_t serisinin varyansı 2.18 no.lu denklemle ifade edilebilir.

$$\text{Var}(y_t) = \frac{\delta_\varepsilon^2}{1-\rho^2} \quad (2.18)$$

2.18 no.lu eşitliğe göre birim kök varken y_t serisinin varyansının sonlu bir değere sahip olmaması, bir önceki çıkarıma uygun olarak onun durağan olmadığını ifade eder. Serinin varyansı ancak $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu (ya da sabit) bir değere sahip olabilir (Evans, 2003, s.229).

2.17 ve 2.18 no.lu y_t serisinin varyans denklemleri aşağıdaki adımlar takip edilerek elde edilebilir.

Varyansın durağanlık analizinin kolaylıkla yapılabilmesi amacıyla, 2.4 no.lu denklem esas alınarak y_t 'nin y_1 için beklenen değeri 2.19, beklenen değerinin karesi 2.20 ve karesinin beklenen değeri ise 2.24 no.lu denklemde sunulmuştur.

$$E(y_1) = A_1 = \rho y_0 \quad (2.19)$$

$$(E(y_1))^2 = A_1^2 = (\rho y_0)^2 \quad (2.20)$$

$$E(y_1^2) = E((A_1 + \varepsilon_1)^2) \quad (2.21)$$

$$E(y_1^2) = E(A_1^2 + 2A_1\varepsilon_1 + \varepsilon_1^2) \quad (2.22)$$

$$E(y_1^2) = A_1^2 + 2A_1 \underbrace{E(\epsilon_1)}_0 + E(\epsilon_1^2) \quad (2.23)$$

$$E(y_1^2) = A_1^2 + E(\epsilon_1^2) \quad (2.24)$$

2.19 no.lu denklemde y_1 'in beklenen değeri bir sabit olup A_1 'e eşit olduğu kabul edilmiştir. 2.4 no.lu denklem, $t=1$ için 2.19, 2.20 ve 2.21 no.lu denklemlerin içinde yerine yazılmıştır. Hata terimlerinin beklenen değeri 0 olduğu için diğer elemanlar ile çapraz çarpımlarının beklenen değerleri de 2.23 no.lu denklemde gösterildiği gibi 0'a eşit olacaktır.

Yukarıda y_1 için kabul edilen benzer yaklaşımlar y_2 ve y_3 için de aynı gerekçelerle kabul edilmiştir. Buna göre y_t 'nin y_2 için beklenen değeri 2.25, beklenen değerinin karesi 2.26 ve karesinin beklenen değeri ise 2.28 no.lu denklemde; y_t 'nin y_3 için beklenen değeri 2.29, beklenen değerinin karesi 2.30 ve karesinin beklenen değeri ise 2.32 no.lu denklemde sunulmuştur.

$$E(y_2) = A_2 = \rho^2 y_0 \quad (2.25)$$

$$(E(y_2))^2 = A_2^2 = (\rho^2 y_0)^2 \quad (2.26)$$

$$E(y_2^2) = E((A_2 + \rho\epsilon_1 + \epsilon_2)^2) \quad (2.27)$$

$$E(y_2^2) = A_2^2 + \rho^2 E(\epsilon_1^2) + E(\epsilon_2^2) \quad (2.28)$$

$$E(y_3) = A_3 = \rho^3 y_0 \quad (2.29)$$

$$(E(y_3))^2 = A_3^2 = (\rho^3 y_0)^2 \quad (2.30)$$

$$E(y_3^2) = E((A_3 + \rho^2\epsilon_1 + \rho\epsilon_2 + \epsilon_3)^2) \quad (2.31)$$

$$E(y_3^2) = A_3^2 + \rho^4 E(\epsilon_1^2) + \rho^2 E(\epsilon_2^2) + E(\epsilon_3^2) \quad (2.32)$$

2.25 no.lu denklemde y_2 'in beklenen değeri bir sabit olup A_2 'ye, 2.29 no.lu denklemde ise y_3 'in beklenen değeri bir sabit olup A_3 'e eşit olduğu kabul

edilmiştir. 2.4 no.lu denklem, t=2 için 2.25, 2.26 ve 2.27 no.lu denklemlerin içinde; t=3 için 2.29, 2.30 ve 2.31 no.lu denklemlerin içinde yerine yazılmıştır.

y_t 'nin varyansı 2.33 no.lu denklemde gösterildiği gibi hesaplanır.

$$\text{Var}(y_t) = E\left(\left(y_t - E(y_t)\right)^2\right) \quad (2.33)$$

2.33 no.lu denklemde, parantez içinde bulunan üslü ifade açılarak 2.34 no.lu denklem elde edilir.

$$\text{Var}(y_t) = E\left(y_t^2 - 2y_tE(y_t) + \left(E(y_t)\right)^2\right) \quad (2.34)$$

Parantez dışında yer alan beklenen değer operatörü parantez içine dağıtılarak 2.35 no.lu denkleme ulaşılır.

$$\text{Var}(y_t) = E(y_t^2) - 2 \underbrace{E(y_t)E(y_t)}_{(E(y_t))^2} + E\left[\underbrace{\left(E(y_t)\right)^2}_{(E(y_t))^2}\right] \quad (2.35)$$

Bir değişkenin beklenen değeri ya da beklenen değerinin karesi bir sabittir. Bir sabitin beklenen değeri ise sabitin kendisine eşittir. O halde bir değişkenin beklenen değerinin beklenen değeri yine o değişkenin beklenen değerine eşit olur. Benzer şekilde bir değişkenin beklenen değerinin karesinin beklenen değeri yine o değişkenin beklenen değerinin karesine eşit olur. Bu çerçevede 2.35 no.lu denklem, 2.36 no.lu denkleme dönüşür.

$$\text{Var}(y_t) = E(y_t^2) - 2\left(E(y_t)\right)^2 + \left(E(y_t)\right)^2 \quad (2.36)$$

$$\text{Var}(y_t) = E(y_t^2) - \left(E(y_t)\right)^2 \quad (2.37)$$

2.37 no.lu denklemden yola çıkılarak gözlem sayısı T=3 basitleştirici varsayımı altında her alternatif t gözlemi için y_t 'nin varyansı aşağıda analiz edilmektedir.

t=1 için y_t 'nin varyansı:

$$\text{Var}(y_1) = E(y_1^2) - (E(y_1))^2 \quad (2.38)$$

2.38 no.lu denklemde $(E(y_1))^2$ ve $E(y_1^2)$ ifadeleri yerine, 2.20 ve 2.24 no.lu denklemler yazılır.

$$\text{Var}(y_1) = A_1^2 + E(\varepsilon_1^2) - A_1^2 \quad (2.39)$$

$$\text{Var}(y_1) = E(\varepsilon_1^2) \quad (2.40)$$

$E(\varepsilon_1^2)$ ifadesi, hata teriminin t=1 döneminde varyansını temsil eder. δ_ε^2 ifadesi hata teriminin t=1, t=2, t=3 gibi tüm dönemlerde sabit olan varyansını temsil etmek üzere, 2.40 no.lu denklem 2.41 no.lu denklemle ifade edilebilir.

$$\text{Var}(y_1) = \delta_\varepsilon^2 \quad (2.41)$$

t=2 için y_t 'nin varyansı:

$$\text{Var}(y_2) = E(y_2^2) - (E(y_2))^2 \quad (2.42)$$

2.42 no.lu denklemde $(E(y_2))^2$ ve $E(y_2^2)$ ifadeleri yerine, 2.26 ve 2.28 no.lu denklemler yazılır.

$$\text{Var}(y_2) = A_2^2 + \rho^2 E(\varepsilon_1^2) + E(\varepsilon_2^2) - A_2^2 \quad (2.43)$$

$$\text{Var}(y_2) = \rho^2 E(\varepsilon_1^2) + E(\varepsilon_2^2) \quad (2.44)$$

$$\text{Var}(y_2) = \rho^2 \delta_\varepsilon^2 + \delta_\varepsilon^2 \quad (2.45)$$

t=3 için y_t 'nin varyansı:

$$\text{Var}(y_3) = E(y_3^2) - (E(y_3))^2 \quad (2.46)$$

2.46 no.lu denklemde $(E(y_3))^2$ ve $E(y_3^2)$ ifadeleri yerine, 2.30 ve 2.32 no.lu denklemler yazılır.

$$\text{Var}(y_3) = A_3^2 + \rho^4 E(\varepsilon_1^2) + \rho^2 E(\varepsilon_2^2) + E(\varepsilon_3^2) - A_3^2 \quad (2.47)$$

$$\text{Var}(y_3) = \rho^4 E(\varepsilon_1^2) + \rho^2 E(\varepsilon_2^2) + E(\varepsilon_3^2) \quad (2.48)$$

$$\text{Var}(y_3) = \rho^4 \delta_\varepsilon^2 + \rho^2 \delta_\varepsilon^2 + \delta_\varepsilon^2 \quad (2.49)$$

Yukarıdaki sonuçlar her alternatif t gözlemi için aşağıda özetlenmiştir.

$$t=1 \text{ iken: } \text{Var}(y_1) = \delta_\varepsilon^2 \quad (2.41)$$

$$t=2 \text{ iken: } \text{Var}(y_2) = \rho^2 \delta_\varepsilon^2 + \delta_\varepsilon^2 \quad (2.45)$$

$$t=3 \text{ iken: } \text{Var}(y_3) = \rho^4 \delta_\varepsilon^2 + \rho^2 \delta_\varepsilon^2 + \delta_\varepsilon^2 \quad (2.49)$$

Gözlem sayısı T iken 2.41, 2.45 ve 2.49 no.lu sonuçlar 2.50 no.lu denklemde gösterildiği gibi genel bir forma dönüştürülebilir.

$$\text{Var}(y_t) = \sum_{t=1}^T \rho^{2t-2} \delta_\varepsilon^2 \quad (2.50)$$

Gözlem sayısı T sonsuza giderken 2.50 no.lu denklem geometrik bir seri oluşturur.

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_\varepsilon^2 = \rho^0 \delta_\varepsilon^2 + \rho^2 \delta_\varepsilon^2 + \rho^4 \delta_\varepsilon^2 + \dots + \rho^{2t-2} \delta_\varepsilon^2 \quad (2.51)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_\varepsilon^2 = (1 + \rho^2 + \rho^4 + \dots + \rho^{2t-2}) \delta_\varepsilon^2 \quad (2.52)$$

2.52 no.lu denklemin A gibi bir sabite eşit olduğu kabul edilerek 2.53 no.lu denklemle ifade edilir ve geometrik seri aşağıdaki denklemlerde açıkça gösterildiği gibi çözülebilir.

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_\varepsilon^2 = (1 + \rho^2 + \rho^4 + \dots + \rho^{2t-2}) \delta_\varepsilon^2 = A \quad (2.53)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = (1 + \rho^2 + \rho^4 + \dots + \rho^{2t-2}) = \frac{A}{\delta_{\epsilon}^2} \quad (2.54)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = 1 + \rho^2 \underbrace{(1 + \rho^2 + \rho^4 + \dots + \rho^{2t-2})}_{\frac{A}{\delta_{\epsilon}^2}} = \frac{A}{\delta_{\epsilon}^2} \quad (2.55)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = 1 + \rho^2 \frac{A}{\delta_{\epsilon}^2} = \frac{A}{\delta_{\epsilon}^2} \quad (2.56)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = 1 = \frac{A}{\delta_{\epsilon}^2} - \rho^2 \frac{A}{\delta_{\epsilon}^2} \quad (2.57)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = 1 = \frac{A}{\delta_{\epsilon}^2} (1 - \rho^2) \quad (2.58)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = \frac{\delta_{\epsilon}^2}{1 - \rho^2} = A \quad (2.59)$$

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\epsilon}^2 = \frac{\delta_{\epsilon}^2}{1 - \rho^2} \quad (2.60)$$

2.60 no.lu geometrik serinin sonucu 2.50 no.lu denklemde yerine yazılır.

$$\text{Var}(y_t) = \frac{\delta_{\epsilon}^2}{1 - \rho^2} \quad (2.61)$$

Böylece 2.61 no.lu denklemde gösterildiği gibi 2.18 no.lu varyans denkleminin ispatı yapılmış olur.

Diğer taraftan T sonlu ve y_t 'nin birim kökü var ($\rho=1$) iken sabit içermeyen (saf) rassal yürüyüş modeli için y_t 'nin varyansı sabit olmayıp gözlem sırasını temsil eden t ile hata teriminin varyansının çarpımına eşittir. Diğer bir ifadeyle $\rho=1$ iken y_t 'nin varyansı zamana bağımlı bir şekilde değişim

göstermektedir. 2.17 no.lu varyans denklemini elde etmek amacıyla, aşağıda gösterildiği gibi 2.50 no.lu denklem açılarak $\rho=1$ için çözülür.

$$\sum_{t=1}^{T=\infty} \rho^{2t-2} \delta_{\varepsilon}^2 = \rho^0 \delta_{\varepsilon}^2 + \rho^2 \delta_{\varepsilon}^2 + \rho^4 \delta_{\varepsilon}^2 + \dots + \rho^{2t-2} \delta_{\varepsilon}^2 \quad (2.63)$$

$$\sum_{t=1}^{T=\infty} 1^{2t-2} \delta_{\varepsilon}^2 = 1^0 \delta_{\varepsilon}^2 + 1^2 \delta_{\varepsilon}^2 + 1^4 \delta_{\varepsilon}^2 + \dots + 1^{2t-2} \delta_{\varepsilon}^2 \quad (2.64)$$

$$\sum_{t=1}^{T=\infty} \delta_{\varepsilon}^2 = \delta_{\varepsilon}^2 + \delta_{\varepsilon}^2 + \delta_{\varepsilon}^2 + \dots + \delta_{\varepsilon}^2 \quad (2.65)$$

$$\sum_{t=1}^{T=\infty} \delta_{\varepsilon}^2 = T \delta_{\varepsilon}^2 \quad (2.66)$$

2.66 no.lu denklem y_t 'nin varyansının sabit olmadığına (zamana göre değiştiğine) işaret eder ve y_t 'nin t. gözlem değeri için varyansı 2.67 no.lu denklemde gösterildiği gibi ifade edilir.

$$\text{Var}(y_t) = t \delta_{\varepsilon}^2 \quad (2.67)$$

Böylece 2.67 no.lu denklemde gösterildiği gibi 2.17 no.lu varyans denkleminin ispatı yapılmış olur.

Diğer taraftan y_t serisinin kovaryansı ise varyansının bir fonksiyonu olduğundan, $\rho=1$ iken (birim kök varken) 2.68 no.lu denklemde gösterildiği gibi zamanın bir fonksiyonu olacaktır.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = (t-k) \delta_{\varepsilon}^2 \quad (2.68)$$

Birim kök varken y_t serisinin kovaryansının zamana göre sabit olmaması, diğer bir ifadeyle k gecikme uzunluğuna ilave olarak zamanın bir fonksiyonu olması, onun durağan olmadığını gösterir.

Varyansın analizine benzer şekilde birim kökün bulunmadığını temsil eden $-1 < \rho < 1$ durumunda, y_t serisinin kovaryansı 2.69 no.lu denklemle ifade edilir.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = \rho^k \frac{\sigma_\varepsilon^2}{1-\rho^2} \quad (2.69)$$

Serinin kovaryansı, varyansta yapılan çıkarıma uygun olarak sadece $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu (zamandan bağımsız) bir değere sahip olabilir. Ayrıca $|\rho| < 1$ iken, k gecikme uzunluğu arttıkça serinin kovaryansları 0 değerine yaklaşacaktır (Hill ve diğerleri, 2011, s.478).

2.68 ve 2.69 no.lu y_t serisinin kovaryans denklemleri aşağıdaki adımlar takip edilerek elde edilebilir.

Kovaryansın durağanlık analizinin kolaylıkla yapılabilmesi amacıyla, 2.4 no.lu denklem esas alınarak y_t 'nin y_1 ve y_2 için beklenen değerleri 2.70 ve 2.71, beklenen değerlerinin çapraz çarpımları 2.72 ve çapraz çarpımlarının beklenen değerleri 2.75 no.lu denklemde sunulmuştur.

$$E(y_1) = A_1 = \rho y_0 \quad (2.70)$$

$$E(y_2) = A_2 = \rho^2 y_0 \quad (2.71)$$

$$E(y_1)E(y_2) = A_1 A_2 \quad (2.72)$$

$$E(y_1 y_2) = E((A_1 + \varepsilon_1)(A_2 + \rho \varepsilon_1 + \varepsilon_2)) \quad (2.73)$$

2.73 no.lu denklemde, hata terimlerinin beklenen değerleri 0 olduğu için diğer elemanlarla çapraz çarpımlarının beklenen değerleri 0 olacaktır.

$$E(y_1 y_2) = A_1 A_2 + \rho E(\varepsilon_1^2) \quad (2.74)$$

$$E(y_1 y_2) = E(y_1 y_2) = A_1 A_2 + \rho \sigma_\varepsilon^2 \quad (2.75)$$

2.70 ve 2.71 no.lu denklemlerde y_1 ve y_2 'nin beklenen değeri birer sabit olup A_1 ve A_2 'ye eşit olduğu kabul edilmiştir. 2.4 no.lu denklem 2.70, 2.71, 2.72 ve 2.73 no.lu denklemlerin içinde yerine yazılmıştır.

Benzer yaklaşımlar y_2 ve y_3 ile y_3 ve y_4 için de aynı gerekçelerle kabul edilmiştir. Buna göre y_t 'nin y_2 ve y_3 için beklenen değeri 2.76 ve 2.77, beklenen değerlerinin çapraz çarpımları 2.78 ve çapraz çarpımlarının beklenen değeri 2.81 no.lu denklemde; y_3 ve y_4 için beklenen değeri ise 2.77 ve 2.82, beklenen değerlerinin çapraz çarpımları 2.83 ve çapraz çarpımlarının beklenen değeri 2.86 no.lu denklemde sunulmuştur.

$$E(y_2) = A_2 = \rho^2 y_0 \quad (2.76)$$

$$E(y_3) = A_3 = \rho^3 y_0 \quad (2.77)$$

$$E(y_2)E(y_3) = A_2 A_3 \quad (2.78)$$

$$E(y_2 y_3) = E((A_2 + \rho \varepsilon_1 + \varepsilon_2)(A_3 + \rho^2 \varepsilon_1 + \rho \varepsilon_2 + \varepsilon_3)) \quad (2.79)$$

$$E(y_2 y_3) = A_2 A_3 + \rho^3 E(\varepsilon_1^2) + \rho E(\varepsilon_2^2) \quad (2.80)$$

$$E(y_2 y_3) = A_2 A_3 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.81)$$

$$E(y_4) = A_4 = \rho^4 y_0 \quad (2.82)$$

$$E(y_3)E(y_4) = A_3 A_4 \quad (2.83)$$

$$E(y_3 y_4) = E((A_3 + \rho^2 \varepsilon_1 + \rho \varepsilon_2 + \varepsilon_3)(A_4 + \rho^3 \varepsilon_1 + \rho^2 \varepsilon_2 + \rho \varepsilon_3 + \varepsilon_4)) \quad (2.84)$$

$$E(y_3 y_4) = A_3 A_4 + \rho^5 E(\varepsilon_1^2) + \rho^3 E(\varepsilon_2^2) + \rho E(\varepsilon_3^2) \quad (2.85)$$

$$E(y_3 y_4) = A_3 A_4 + \rho^5 \delta_\varepsilon^2 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.86)$$

k iki dönem arasındaki gecikme uzunluğunu ve t dönemi temsil etmek üzere, y_t 'nin kovaryansı 2.87 no.lu denklemde gösterildiği gibi hesaplanır.

$$\text{Cov}(y_t, y_{t+k}) = E\left(\left(y_t - E(y_t)\right)\left(y_{t+k} - E(y_{t+k})\right)\right) \quad (2.87)$$

2.87 no.lu denklemde, parantez içinde bulunan ifadeler çarpılıp, beklenen değer operatörü parantez içine dağıtıldıktan sonra 2.88 no.lu denklem elde edilir.

$$\text{Cov}(y_t, y_{t+k}) = E(y_t y_{t+k}) - E(y_t)E(y_{t+k}) \quad (2.88)$$

2.88 no.lu denklemden yola çıkılarak gecikme uzunluğu $k=1$ ve gözlem sayısı $T=4$ basitleştirici varsayımında, her alternatif t gözlemi için y_t 'nin kovaryansları aşağıda analiz edilmektedir.

$k=1$ ve $t=1$ için y_t 'nin kovaryansı:

$$\text{Cov}(y_1, y_2) = E(y_1 y_2) - E(y_1)E(y_2) \quad (2.89)$$

2.89 no.lu denklemde $E(y_1)E(y_2)$ ve $E(y_1 y_2)$ ifadeleri yerine, 2.72 ve 2.75 no.lu denklemler yazılarak 2.91 no.lu denkleme ulaşılır.

$$\text{Cov}(y_1, y_2) = A_1 A_2 + \rho \delta_\varepsilon^2 - A_1 A_2 \quad (2.90)$$

$$\text{Cov}(y_1, y_2) = \rho \delta_\varepsilon^2 \quad (2.91)$$

$k=1$ ve $t=2$ için y_t 'nin kovaryansı:

$$\text{Cov}(y_2, y_3) = E(y_2 y_3) - E(y_2)E(y_3) \quad (2.92)$$

2.92 no.lu denklemde $E(y_2)E(y_3)$ ve $E(y_2 y_3)$ ifadeleri yerine, 2.78 ve 2.81 no.lu denklemler yazılarak 2.94 no.lu denkleme ulaşılır.

$$\text{Cov}(y_2, y_3) = A_2 A_3 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 - A_2 A_3 \quad (2.93)$$

$$\text{Cov}(y_2, y_3) = \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.94)$$

$k=1$ ve $t=3$ için y_t 'nin kovaryansı:

$$\text{Cov}(y_3, y_4) = E(y_3 y_4) - E(y_3)E(y_4) \quad (2.95)$$

2.95 no.lu denklemde $E(y_3)E(y_4)$ ve $\text{Cov}(y_3, y_4)$ ifadeleri yerine, 2.83 ve 2.86 no.lu denklemler yazılarak 2.97 no.lu denkleme ulaşılır.

$$\text{Cov}(y_3, y_4) = A_3 A_4 + \rho^5 \delta_\varepsilon^2 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 - A_3 A_4 \quad (2.96)$$

$$\text{Cov}(y_3, y_4) = \rho^5 \delta_\varepsilon^2 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.97)$$

Yukarıdaki sonuçlar her alternatif t gözlemi için aşağıda özetlenmiştir.

$$t=1 \text{ iken: } \text{Cov}(y_1, y_2) = \rho \delta_\varepsilon^2 \quad (2.91)$$

$$t=2 \text{ iken: } \text{Cov}(y_2, y_3) = \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.94)$$

$$t=3 \text{ iken: } \text{Cov}(y_3, y_4) = \rho^5 \delta_\varepsilon^2 + \rho^3 \delta_\varepsilon^2 + \rho \delta_\varepsilon^2 \quad (2.97)$$

Gecikme uzunluğu k ve gözlem sayısı T iken 2.91, 2.94 ve 2.97 no.lu sonuçlar 2.98 no.lu denklemde gösterildiği gibi genel bir forma dönüştürülebilir.

$$\text{Cov}(y_t, y_{t+k}) = \sum_{t=1}^{T-k} \rho^{2t+k-2} \delta_\varepsilon^2 \quad (2.98)$$

Gecikme uzunluğu k=1 ve gözlem sayısı T sonsuza giderken 2.98 no.lu denklem geometrik bir seri oluşturur.

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_\varepsilon^2 = \rho^1 \delta_\varepsilon^2 + \rho^3 \delta_\varepsilon^2 + \rho^5 \delta_\varepsilon^2 + \dots + \rho^{2t-1} \delta_\varepsilon^2 \quad (2.99)$$

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_\varepsilon^2 = (\rho + \rho^3 + \rho^5 + \dots + \rho^{2t-1}) \delta_\varepsilon^2 = A \quad (2.100)$$

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_\varepsilon^2 = (\rho + \rho^3 + \rho^5 + \dots + \rho^{2t-1}) = \frac{A}{\delta_\varepsilon^2} \quad (2.101)$$

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho + \rho^2 \underbrace{(\rho + \rho^3 + \rho^5 + \dots + \rho^{2t-1})}_{\frac{A}{\delta_{\varepsilon}^2}} = \frac{A}{\delta_{\varepsilon}^2} \quad (2.102)$$

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho + \rho^2 \frac{A}{\delta_{\varepsilon}^2} = \frac{A}{\delta_{\varepsilon}^2} \quad (2.103)$$

$$\sum_{t=1}^{T-1} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho \frac{\delta_{\varepsilon}^2}{1-\rho^2} \quad (2.104)$$

2.99 ve 2.103 arasındaki adımlar $k=2$ ve $k=3$ için tekrarlandığında 2.105 ve 2.106 no.lu sonuçlar elde edilir.

$$\sum_{t=1}^{T-2} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho^2 \frac{\delta_{\varepsilon}^2}{1-\rho^2} \quad (2.105)$$

$$\sum_{t=1}^{T-3} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho^3 \frac{\delta_{\varepsilon}^2}{1-\rho^2} \quad (2.106)$$

2.104, 2.105 ve 2.106 no.lu sonuçlara göre gecikme uzunluğu k ve gözlem sayısı T iken y_t 'nin kovaryansları 2.107 ve 2.108 no.lu denklemlerde gösterildiği gibi genel bir forma dönüştürülebilir.

$$\sum_{t=1}^{T-k} \rho^{2t+k-2} \delta_{\varepsilon}^2 = \rho^k \frac{\delta_{\varepsilon}^2}{1-\rho^2} \quad (2.107)$$

$$\text{Cov}(y_t, y_{t+k}) = \rho^k \frac{\delta_{\varepsilon}^2}{1-\rho^2} \quad (2.108)$$

Böylece 2.108 no.lu denklemde gösterildiği gibi 2.69 no.lu kovaryans denkleminin ispatı yapılmış olur.

2.61 no.lu varyans denkleminde görüleceği üzere $\frac{\delta_{\varepsilon}^2}{1-\rho^2}$ ifadesi y_t 'nin varyansına eşittir. 2.108 no.lu denklemde $\frac{\delta_{\varepsilon}^2}{1-\rho^2}$ ifadesi yerine $\text{Var}(y_t)$ yazılarak, y_t 'nin kovaryansları 2.109 no.lu denklemle de ifade edilebilir.

$$\text{Cov}(y_t, y_{t+k}) = \rho^k \text{Var}(y_t) \quad (2.109)$$

Ayrıca y_t 'nin birim kökü varken, diğer bir ifadeyle $\rho=1$ iken sabit içermeyen (saf) rassal yürüyüş modeli için y_t 'nin kovaryansları asimptotik açıdan (yani n sonsuza giderken) varyanstan dolayı 2.110 no.lu denklemde gösterildiği üzere tanımsız olmaktadır.

$$\text{Cov}(y_t, y_{t+k}) = \rho^k \frac{\delta_\varepsilon^2}{1-1} = \frac{\delta_\varepsilon^2}{0} = \text{Tanımsız} \quad (2.110)$$

Diğer taraftan T sonlu ve y_t 'nin birim kökü varken sabit içermeyen (saf) rassal yürüyüş modeli için y_t 'nin kovaryansları, gözlem sırası t ve gecikme uzunluğu k arasındaki fark ile hata teriminin varyansının çarpımına eşit olacaktır. 2.68 no.lu kovaryans denklemini elde etmek amacıyla, aşağıda gösterildiği gibi 2.98 no.lu denklem açılarak $\rho=1$ için çözülür.

$$\text{Cov}(y_t, y_{t+k}) = \sum_{t=1}^{T-k} \rho^{2t+k-2} \delta_\varepsilon^2 = \sum_{t=1}^{T-k} 1^{2t+k-2} \delta_\varepsilon^2 \quad (2.111)$$

$$\text{Cov}(y_t, y_{t+k}) = \sum_{t=1}^{T-k} \delta_\varepsilon^2 = \underbrace{\delta_\varepsilon^2 + \delta_\varepsilon^2 + \delta_\varepsilon^2 + \dots + \delta_\varepsilon^2}_{T-k} \quad (2.112)$$

$$\text{Cov}(y_t, y_{t+k}) = (T-k) \delta_\varepsilon^2 \quad (2.113)$$

2.113 no.lu denklem y_t 'nin kovaryanslarının sadece gecikme uzunluğuna göre değil, zamana göre de değiştiğine işaret eder ve y_t 'nin t . gözlem için kovaryansları 2.114 no.lu denklemde gösterildiği gibi ifade edilebilir.

$$\text{Cov}(y_t, y_{t+k}) = (t-k) \delta_\varepsilon^2 \quad (2.114)$$

2.114 no.lu denklemde de açıkça görüleceği üzere $\rho=1$ iken y_t 'nin kovaryansları zamanın bir fonksiyonu olduğundan, zamana göre değişmektedir.

Böylece 2.114 no.lu denklemde gösterildiği gibi 2.68 no.lu varyans denkleminin ispatı yapılmış olur.

y_t 'nin birim kökü yok ve $\rho=0$ iken sabit içermeyen (saf) rassal yürüyüş modeli için y_t 'nin kovaryansları, 2.115 no.lu denklemde gösterildiği gibi zamandan ve gecikme uzunluğundan bağımsız olarak 0 değerine eşit olur.

$$\text{Cov}(y_t, y_{t+k}) = 0^k \frac{\sigma_\varepsilon^2}{1-0^2} = 0 \quad (2.115)$$

y_t 'nin birim kökü yok ($\rho \neq 1$) ve $\rho \neq 0$, yani ρ ifadesi 0 ile 1 arasında bir değere sahip iken sabit içermeyen (saf) rassal yürüyüş modeli için y_t 'nin kovaryansları, zamandan bağımsız olarak sadece gecikme uzunluğu k 'ye göre değişir.

$$\text{Cov}(y_t, y_{t+k}) = \rho^k \frac{\sigma_\varepsilon^2}{1-\rho^2} \quad (2.116)$$

Ancak gecikme uzunluğu k arttıkça, y_t 'nin kovaryansları ρ^k ifadesinden dolayı 0 değerine yaklaşacaktır. Bu durum zaman serilerinde genellikle karşılaşılan bir durağanlık türü olup zayıf durağanlık olarak adlandırılır.

Özetle saf rassal yürüyüş modeli ya da AR(1) süreci için durağanlık koşulu $|\rho| < 1$ olarak ifade edilebilir.

2.1.2. Sabit İçeren Rassal Yürüyüş Modeli

Saf rassal yürüyüş modelinde izlenen benzer yöntemlerle, sabit içeren rassal yürüyüş modelinin de durağanlık analizi yapılabilir.

$$y_t = \beta_0 + \rho y_{t-1} + \varepsilon_t \quad (2.117)$$

Saf rassal yürüyüş modeli konusunda ele alınan yaklaşıma benzer şekilde, 2.117 no.lu denklem de sabit içeren AR(1) sürecidir. Bununla birlikte $\rho=1$ iken 2.117 no.lu denklemle ifade edilen AR(1) süreci sabit içeren rassal

yürüyüş modeline dönüşür. 2.117 no.lu model $t=1$, $t=2$ ve $t=3$ iken aşağıda gösterildiği gibi ifade edilebilir.

$$y_1 = \beta_0 + \rho y_0 + \varepsilon_1 \quad (2.118)$$

$$y_2 = \beta_0 + \rho(\beta_0 + \rho y_0 + \varepsilon_1) + \varepsilon_2 \quad (2.119)$$

$$y_3 = \beta_0 + \rho(\beta_0 + \rho(\beta_0 + \rho y_0 + \varepsilon_1) + \varepsilon_2) + \varepsilon_3 \quad (2.120)$$

$$y_3 = \beta_0 + \rho\beta_0 + \rho^2\beta_0 + \rho^3y_0 + \rho^2\varepsilon_1 + \rho\varepsilon_2 + \varepsilon_3 \quad (2.121)$$

Yukarıdaki süreç gözlem sayısını temsil eden T kez tekrarlanırsa 2.122 no.lu eşitlik elde edilir.

$$y_t = \rho^T y_0 + \left(\sum_{t=1}^T \rho^{t-1} \beta_0 \right) + \left(\sum_{t=1}^T \rho^{t-1} \varepsilon_{T-t+1} \right) \quad (2.122)$$

Eğer $\rho=1$ iken yukarıdaki süreç, 2.123 no.lu denklemde gösterildiği gibi y_t serisinin başlangıç değeri ve zamanla artan sabit ile geçmiş dönem hata terimlerinin birikimli bir fonksiyonu olacaktır.

$$y_t = y_0 + \left(\sum_{t=1}^T \beta_0 \right) + \left(\sum_{t=1}^T \varepsilon_t \right) \quad (2.123)$$

2.123 no.lu denklem t . gözlem için genelleştirilerek 2.124 no.lu denklem elde edilir.

$$y_t = y_0 + t\beta_0 + \left(\sum_{t=1}^T \varepsilon_t \right) \quad (2.124)$$

2.124 no.lu eşitlikte yer alan $\sum_{t=1}^T \varepsilon_t$ ifadesi stokastik trendi, $t\beta_0$ değeri ise her t adımımda sabit bir şekilde arttığından deterministik bir trendi ifade eder (Hill ve diğerleri, 2011, s. 481). y_t serisinin beklenen değerini elde etmek için 2.124 no.lu denkleme beklenen değer operatörü ilave edilerek 2.125 no.lu denklem elde edilir.

$$E[y_t] = E[y_0] + E[t\beta_0] + E\left[\left(\sum_{t=1}^T \varepsilon_t\right)\right] \quad (2.125)$$

Başlangıç değerini ifade eden y_0 gibi $t\beta_0$ ifadesi de bir sabit ve bir sabitin beklenen değeri de kendisine eşit olduğundan, y_t serisinin beklenen değeri 2.126 no.lu denklemde gösterildiği gibi olacaktır.

$$E[y_t] = y_0 + t\beta_0 \quad (2.126)$$

Birim kökün varlığı durumunda serinin beklenen değeri sabit olmayıp zamanın bir fonksiyonudur.

Diğer taraftan birim kökün bulunmadığını temsil eden $-1 < \rho < 1$ durumunda, y_t serisinin beklenen değeri 2.127 no.lu denklemle ifade edilebilir (Hill ve diğerleri, 2011, s.479-480).

$$\mu = \frac{\beta_0}{1-\rho} \quad (2.127)$$

2.127 no.lu denklem, birim kök yokken, sabit içeren saf rassal yürüyüş modelinde serinin beklenen değerinin zamandan bağımsız (sabit) olduğuna işaret eder.

2.127 no.lu y_t serisinin ortalama denklemi aşağıdaki adımlar takip edilerek elde edilebilir.

2.117 no.lu denkleme beklenen değer operatörü ilave edilerek 2.129 no.lu denkleme ulaşılır.

$$E(y_t) = E(\beta_0) + E(\rho y_{t-1}) + E(\varepsilon_t) \quad (2.128)$$

$$E(y_t) = \beta_0 + \rho E(y_{t-1}) \quad (2.129)$$

Gözlem sayısı $T=3$ basitleştirici varsayımında, 2.129 no.lu denklem kullanılarak her alternatif t gözlemi için y_t 'nin beklenen değerleri 2.130, 2.132 ve 2.135 no.lu denklemlerde gösterildiği gibi olur.

$$E(y_1) = \beta_0 + \rho y_0 \quad (2.130)$$

$$E(y_2) = \beta_0 + \rho E(\beta_0 + \rho y_0) \quad (2.131)$$

$$E(y_2) = \beta_0 + \rho \beta_0 + \rho^2 y_0 \quad (2.132)$$

$$E(y_3) = \beta_0 + \rho y_2 \quad (2.133)$$

$$E(y_3) = \beta_0 + \rho(\beta_0 + \rho \beta_0 + \rho^2 y_0) \quad (2.134)$$

$$E(y_3) = \beta_0 + \rho \beta_0 + \rho^2 \beta_0 + \rho^3 y_0 \quad (2.135)$$

Yukarıda y_t 'nin beklenen değerleri için gösterilen süreç, gözlem sayısı T iken toplam sembolü yardımıyla genel bir forma dönüştürülebilir.

$$E(y_t) = \rho^t y_0 + \sum_{t=1}^T \rho^{t-1} \beta_0 \quad (2.136)$$

Gözlem sayısı T sonsuza giderken 2.136 no.lu denklem geometrik bir seri oluşturur. Bu geometrik seri aşağıda gösterildiği gibi çözülür.

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = \rho^t y_0 + \beta_0 + \rho \beta_0 + \rho^2 \beta_0 + \rho^3 \beta_0 + \dots + \rho^{t-1} \beta_0 \quad (2.137)$$

2.137 no.lu denklem içindeki $\rho^t y_0$ ifadesi, t sonsuza giderken 0 değerine eşit olacağından denklemden çıkartılabilir.

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = (1 + \rho + \rho^2 + \rho^3 + \dots + \rho^{t-1}) \beta_0 = A \quad (2.138)$$

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = (1 + \rho + \rho^2 + \rho^3 + \dots + \rho^{t-1}) = \frac{A}{\beta_0} \quad (2.139)$$

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = 1 + \rho \underbrace{(1 + \rho + \rho^2 + \rho^3 + \dots + \rho^{t-1})}_{\frac{A}{\beta_0}} = \frac{A}{\beta_0} \quad (2.140)$$

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = 1 + \rho \frac{A}{\beta_0} = \frac{A}{\beta_0} \quad (2.141)$$

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = 1 = (1-\rho) \frac{A}{\beta_0} \quad (2.142)$$

$$\sum_{t=1}^{T=\infty} \rho^{t-1} \beta_0 = \frac{\beta_0}{1-\rho} \quad (2.143)$$

2.143 no.lu geometrik serinin sonucu 2.136 no.lu denklemde yerine yazılır.

$$E(y_t) = \frac{\beta_0}{1-\rho} \quad (2.144)$$

Böylece 2.144 no.lu denklemde gösterildiği gibi 2.127 no.lu ortalama (beklenen değer) denkleminin ispatı yapılmış olur.

Ayrıca y_t 'nin birim kökü varken, diğer bir ifadeyle $\rho=1$ iken sabit içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri asimptotik açıdan (yani T sonsuza giderken) tanımsız olmaktadır.

$$E(y_t) = \frac{\beta_0}{1-1} = \frac{\beta_0}{0} = \text{Tanımsız} \quad (2.145)$$

Diğer taraftan n sonlu ve y_t 'nin birim kökü var ($\rho=1$) iken sabit içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri sabit olmayıp gözlem sırasını temsil eden t ile sabitin çarpımı ile y_t 'nin başlangıç değerinin toplamına eşittir. Diğer bir ifadeyle $\rho=1$ iken y_t 'nin beklenen değeri zamana bağımlı bir şekilde değişim göstermektedir.

$$E(y_t) = \rho^t y_0 + \sum_{t=1}^T \rho^{t-1} \beta_0 \quad (2.146)$$

$$E(y_t) = \rho^t y_0 + \beta_0 + \rho \beta_0 + \rho^2 \beta_0 + \dots + \rho^{t-1} \beta_0 \quad (2.147)$$

$$E(y_t) = 1y_0 + \beta_0 + 1\beta_0 + 1^2\beta_0 + \dots + 1^{t-1}\beta_0 \quad (2.148)$$

$$E(y_t) = y_0 + \underbrace{\beta_0 + \beta_0 + \beta_0 + \dots + \beta_0}_T \quad (2.149)$$

$$E(y_t) = y_0 + T\beta_0 \quad (2.150)$$

2.150 no.lu denklem y_t 'nin beklenen değerinin sabit olmadığına (zamana göre değiştiğine) işaret eder ve y_t 'nin t. gözlem değeri için beklenen değeri 2.151 no.lu denklemle ifade edilebilir.

$$E(y_t) = y_0 + t\beta_0 \quad (2.151)$$

y_t 'nin birim kökü yok ($\rho \neq 1$) ve $\rho = 0$ iken sabit içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri, 2.153 no.lu denklemde gösterildiği üzere zamandan bağımsız olarak β_0 sabitine eşit olur.

$$E(y_t) = \frac{\beta_0}{1-0} \quad (2.152)$$

$$E(y_t) = \beta_0 \quad (2.153)$$

Serinin varyansı ise $\rho = 1$ iken (birim kök varken) saf rassal yürüyüş modelinde olduğu gibi 2.154 no.lu denklemde gösterildiği üzere zamanın bir fonksiyonu olacaktır.

$$\text{Var}(y_t) = t\delta_\epsilon^2 \quad (2.154)$$

Birim kök varken y_t serisinin varyansının zamana göre sabit olmaması, onun durağan olmadığını ifade eder. Diğer taraftan birim kökün olmadığını ifade eden $-1 < \rho < 1$ durumunda, y_t serisinin varyansı 2.155 no.lu denklemle ifade edilebilir.

$$\text{Var}(y_t) = \frac{\delta_\epsilon^2}{1-\rho^2} \quad (2.155)$$

2.155 no.lu eşitliğe göre birim kök varken y_t serisinin varyansının sonlu bir değere sahip olmaması, saf rassal yürüyüş modelinde olduğu gibi onun

durağan olmadığını ifade eder. Serinin varyansı ancak $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu (ya da sabit) bir değere sahip olabilir.

2.154 ve 2.155 no.lu varyans denklemleri, “Saf Rassal Yürüyüş Modeli” konusunda detaylı olarak açıklanan varyansa ilişkin adımlar takip edilerek elde edilebilir.

Benzer şekilde y_t serisinin kovaryansı ise varyansının bir fonksiyonu olduğundan, $\rho=1$ iken (birim kök varken serinin kovaryansı da 2.156 no.lu denklemde gösterildiği gibi gecikmenin ve zamanın bir fonksiyonu olacaktır.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = (t-k)\delta_\varepsilon^2 \quad (2.156)$$

Birim kök varken y_t serisinin kovaryansının zamana göre sabit olmaması, diğer bir ifadeyle k gecikme uzunluğuna ilave olarak zamanın bir fonksiyonu olması, onun durağan olmadığını gösterir.

Varyansın analizine benzer şekilde birim kökün olmadığını ifade eden $-1 < \rho < 1$ durumunda, y_t serisinin kovaryansı 2.157 no.lu denklemle ifade edilebilir.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = \rho^k \frac{\delta_\varepsilon^2}{1-\rho^2} \quad (2.157)$$

Serinin kovaryansı, varyansta yapılan çıkarıma uygun olarak sadece $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu bir değere sahip olabilir. Serinin kovaryansının sadece k gecikme uzunluğunun bir fonksiyonu olması, onun zamandan bağımsız bir şekilde belirlendiğini gösterir. Ayrıca $|\rho| < 1$ iken, k gecikme uzunluğu arttıkça serinin kovaryansları 0 değerine yaklaşacaktır (Hill ve diğerleri, 2011, s.478).

2.156 ve 2.157 no.lu varyans denklemleri, “Saf Rassal Yürüyüş Modeli” konusunda detaylı olarak açıklanan kovaryansa ilişkin adımlar takip edilerek elde edilebilir.

Özetle sabit içeren rassal yürüyüş modeli ya da sabit içeren AR(1) süreci için durağanlık koşulu, saf rassal yürüyüş modelinde olduğu gibi $|\rho| < 1$ olarak ifade edilebilir.

2.1.3. Trend İçeren Rassal Yürüyüş Modeli

Zaman serisinin içindeki trend, kronolojik bir zaman aralığının kendisinin ya da karesinin deterministik bir fonksiyonu olabilir. Bu çerçevede rassal yürüyüş modeline 2.158 no.lu denklemde gösterildiği üzere t' gibi bir trend değişkeni ilave edilebilir.

$$y_t = \beta_0 + \beta_1 t' + \rho y_{t-1} + \varepsilon_t \quad (2.158)$$

Rassal yürüyüş modeline ilave edilen t' trend değişkeni, zamanın deterministik bir fonksiyonu ise serinin deterministik bir trende sahip olduğunu, tahmin edilemez bir yapıda ise stokastik bir trende sahip olduğunu ifade eder (Gujarati ve Porter, 2009, s.745).

2.158 no.lu model $t=1$, $t=2$ ve $t=3$ iken 2.159, 2.160 ve 2.162 no.lu denklemlerle ifade edilebilir.

$$y_1 = \beta_0 + \beta_1 t' + \rho y_0 + \varepsilon_1 \quad (2.159)$$

$$y_2 = \beta_0 + \beta_1 t' + \rho(\beta_0 + \beta_1 t' + \rho y_0 + \varepsilon_1) + \varepsilon_2 \quad (2.160)$$

$$y_3 = \beta_0 + \beta_1 t' + \rho(\beta_0 + \beta_1 t' + \rho(\beta_0 + \beta_1 t' + \rho y_0 + \varepsilon_1) + \varepsilon_2) + \varepsilon_3 \quad (2.161)$$

$$y_3 = \beta_0 + \rho\beta_0 + \rho^2\beta_0 + \beta_1 t' + \rho\beta_1 t' + \rho^2\beta_1 t' + \rho^3 y_0 + \rho^2\varepsilon_1 + \rho\varepsilon_2 + \varepsilon_3 \quad (2.162)$$

Yukarıdaki süreç gözlem sayısını temsil eden T kez tekrarlanırsa 2.163 no.lu eşitlik elde edilir.

$$y_t = \rho^T y_0 + \left(\sum_{t=1}^T \rho^{t-1} \beta_0 \right) + \left(\sum_{t=1}^T \rho^{t-1} \beta_1 t' \right) + \left(\sum_{t=1}^T \rho^{t-1} \varepsilon_{T-t+1} \right) \quad (2.163)$$

Eğer $\rho=1$ iken yukarıdaki süreç, 2.164 no.lu denklemde gösterildiği gibi y_t serisinin başlangıç değeri, zamanla artan sabit ve trend ile geçmiş dönem hata terimlerinin birikimli bir fonksiyonu olacaktır.

$$y_t = y_0 + \left(\sum_{t=1}^T \beta_0 \right) + \left(\sum_{t=1}^T \beta_1 t \right) + \left(\sum_{t=1}^T \varepsilon_t \right) \quad (2.164)$$

$t' = 1+2+3 \dots + T = \frac{T(T+1)}{2}$ olduğundan 2.164 no.lu eşitlik t . gözlem için genelleştirilerek 2.165 no.lu denklem elde edilir.

$$y_t = y_0 + \beta_0 t + \left(\frac{t(t+1)}{2} \right) \beta_1 + t \varepsilon_t \quad (2.165)$$

2.165 no.lu eşitlikte yer alan $t \varepsilon_t$ ifadesi stokastik trendi, $t \beta_0$ ifadesi deterministik trendi ve $\frac{t(t+1)}{2} \beta_1$ ise güçlendirilmiş deterministik trendi ifade eder (Hill ve diğerleri, 2011, s. 482). Birim kök varken y_t serisinin beklenen değeri ise 2.166 no.lu denklemde gösterilmektedir.

$$E(y_t) = y_0 + \beta_0 t + \left(\frac{t(t+1)}{2} \right) \beta_1 \quad (2.166)$$

2.166 no.lu denkleme göre y_t serisinin beklenen değeri iki parçalı deterministik trende sahiptir. Birim kök varken serinin beklenen değeri, zamanın bir fonksiyonu olmasına (sabit olmamasına) rağmen, deterministik bir trende sahiptir (Hill ve diğerleri, 2011, s. 480).

Yukarıdaki açıklamalar çerçevesinde, trend içeren rassal yürüyüş modelinin beklenen değeri için detaylı bir durağanlık analiz yapmak amacıyla, 2.163 no.lu denkleme beklenen değer operatörü ilave edilerek 2.167 no.lu denklem ulaşılır.

$$E(y_t) = E(\rho^T y_0) + E\left(\sum_{t=1}^T \rho^{t-1} \beta_0\right) + E\left(\sum_{t=1}^T \rho^{t-1} \beta_1 t\right) + E\left(\sum_{t=1}^T \rho^{t-1} \varepsilon_{T-t+1}\right) \quad (2.167)$$

Hata terimlerinin beklenen değeri 0 değerine, sabitlerin beklenen değerleri ise kendilerine eşit olduğundan 2.167 no.lu denklem, 2.168 no.lu denklemde gösterildiği gibi sadeleştirilerek ifade edilebilir.

$$E(y_t) = \rho^T y_0 + \sum_{t=1}^T \rho^{t-1} \beta_0 + \sum_{t=1}^T \rho^{t-1} \beta_1 t' \quad (2.168)$$

Gözlem sayısı T sonsuza giderken, 2.168 no.lu denklemde yer alan $\sum_{t=1}^T \rho^{t-1} \beta_0$ ve $\sum_{t=1}^T \rho^{t-1} \beta_1 t'$ ifadeleri birer geometrik seri oluşturur. Saf rassal yürüyüş modelinde kullanılan yöntemler yardımıyla elde edilen geometrik serilerin sonuçları 2.168 no.lu denklemde yerlerine yazılarak 2.170 no.lu denklem elde edilir.

$$E(y_t) = \rho^T y_0 + \frac{\beta_0}{1-\rho} + \frac{\beta_1 t'}{1-\rho} \quad (2.169)$$

$$E(y_t) = \rho^T y_0 + \frac{\beta_0 + \beta_1 t'}{1-\rho} \quad (2.170)$$

y_t 'nin birim kökü varken, diğer bir ifadeyle $\rho=1$ iken trend içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri asimptotik açıdan (yani T sonsuza giderken) 2.171 no.lu denklemde gösterildiği gibi tanımsız olmaktadır.

$$E(y_t) = y_0 + \frac{\beta_0 + \beta_1 t'}{1-1} = y_0 + \frac{\beta_0 + \beta_1 t'}{0} = \text{Tanımsız} \quad (2.171)$$

Diğer taraftan T sonlu ve y_t 'nin birim kökü var ($\rho=1$) iken trend içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri sabit olmayıp, 2.168 no.lu denklemde hareketle türetilen 2.172 no.lu denklemle ifade edilebilir.

$$E(y_t) = 1y_0 + \sum_{t=1}^T 1\beta_0 + \sum_{t=1}^T 1\beta_1 t' \quad (2.172)$$

Yukarıdaki denklem y_t 'nin beklenen değerinin sabit olmadığına (zamana göre değiştiğine) işaret eder ve y_t 'nin t. gözlem değeri için beklenen

değeri 1.173 no.lu denklemde gösterildiği gibi ifade edilerek 2.166 no.lu beklenen değer denklemi ispat edilir.

$$E(y_t) = y_0 + \beta_0 t + \left(\frac{t(t+1)}{2} \right) \beta_1 \quad (2.173)$$

y_t 'nin birim kökü yok ($\rho \neq 1$) ve $\rho \neq 0$, yani ρ ifadesi 0 ile 1 arasında bir değere sahip iken trend içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri 2.170 no.lu denklem yardımıyla analiz edilebilir.

$$E(y_t) = \rho^T y_0 + \frac{\beta_0 + \beta_1 t}{1-\rho} \quad (2.170)$$

Gözlem sayısı T sonsuza giderken ρ^t ifadesi 0 değerine yakınsayacağından, trend içeren rassal yürüyüş modeli için y_t 'nin deterministik bir trende sahip beklenen değeri 2.174 no.lu denkleme eşit olur.

$$E(y_t) = \frac{\beta_0 + \beta_1 t}{1-\rho} \quad (2.174)$$

y_t 'nin birim kökü yok ($\rho \neq 1$) ve $\rho = 0$ iken trend içeren rassal yürüyüş modeli için y_t 'nin beklenen değeri 2.176 no.lu denkleme eşit olur.

$$E(y_t) = \frac{\beta_0 + \beta_1 t}{1-0} \quad (2.175)$$

$$E(y_t) = \beta_0 + \beta_1 t \quad (2.176)$$

Bu sonuç birim kökün yokluğunda, serinin zamanın (trendin) bir fonksiyonu olmasına rağmen, beklenen değerinin deterministik bir trende sahip olduğuna işaret eder.

y_t serisinin varyansı, $\rho = 1$ iken (birim kök varken) saf rassal yürüyüş modelinde olduğu gibi 2.177 no.lu denklemde gösterildiği üzere zamanın bir fonksiyonu olacaktır.

$$\text{Var}(y_t) = t\delta_\varepsilon^2 \quad (2.177)$$

Birim kök varken y_t serisinin varyansının zamana göre sabit olmaması, onun durağan olmadığını ifade eder. Diğer taraftan $-1 < \rho < 1$ iken y_t serisinin varyansı 2.178 no.lu denklemle ifade edilebilir.

$$\text{Var}(y_t) = \frac{\delta_\varepsilon^2}{1-\rho^2} \quad (2.178)$$

2.179 no.lu eşitliğe göre birim kök varken y_t serisinin varyansının sonlu bir değere sahip olmaması, saf rassal yürüyüş modelinde olduğu gibi onun durağan olmadığını ifade eder. Serinin varyansı ancak $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu (sabit) bir değere sahip olabilir.

2.178 ve 2.179 no.lu varyans denklemleri, “Saf Rassal Yürüyüş Modeli” konusunda detaylı olarak açıklanan varyansa ilişkin adımlar takip edilerek elde edilebilir.

Benzer şekilde serinin kovaryansı ise varyansının bir fonksiyonu olduğundan, $\rho=1$ iken (birim kök varken) serinin kovaryansı da 2.179 no.lu denklemde gösterildiği üzere zamanın ve gecikmenin bir fonksiyonu olacaktır.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = (t-k)\delta_\varepsilon^2 \quad (2.179)$$

Birim kök varken y_t serisinin kovaryansının zamana göre sabit olmaması, diğer bir ifadeyle k gecikme uzunluğuna ilave olarak zamanın bir fonksiyonu olması, onun durağan olmadığını gösterir.

Varyansın analizine benzer şekilde $-1 < \rho < 1$ iken y_t serisinin kovaryansı 2.180 no.lu denklemle ifade edilebilir.

$$\text{Cov}(y_t, y_{t+k}) = \gamma_k = \rho^k \frac{\delta_\varepsilon^2}{1-\rho^2} \quad (2.180)$$

Serinin kovaryansı, varyansta yapılan çıkarıma uygun olarak sadece $-1 < \rho < 1$ ya da $|\rho| < 1$ iken sonlu bir değere sahip olabilir. Ayrıca $|\rho| < 1$ iken, k gecikme uzunluğu arttıkça serinin kovaryansları 0 değerine yaklaşacaktır.

2.179 ve 2.180 no.lu kovaryans denklemleri, “Saf Rassal Yürüyüş Modeli” konusunda detaylı olarak açıklanan kovaryansa ilişkin adımlar takip edilerek elde edilebilir.

Özetle trend içeren rassal yürüyüş modeli için durağanlık koşulu, saf ve sabit içeren rassal yürüyüş modellerinde olduğu gibi $|\rho| < 1$ olarak ifade edilebilir.

2.2. Durağan Olmayan Serilerin Durağanlaştırılması

Bir serinin kendi gecikmesine göre farkı bütünleşme derecesi (integrated degree) olarak tanımlanır. Bütünleşme derecesi “d” ifadesi ile temsil edilir. Eğer y_t serisi durağan ise y_t serisinin sıfırıncı dereceden durağan olduğu ifade edilir ve $I(0)$ ile sembolize edilir (Brooks, 2008, s.326). Durağan olmayan bir y_t serisi, 2.181, 2.182 ve 2.183 no.lu denklemlerde gösterildiği gibi birinci farkı alınarak durağanlaştırılabilir.

$$y_t = y_{t-1} + \varepsilon_t \quad (2.181)$$

$$y_t - y_{t-1} = \varepsilon_t \quad (2.182)$$

$$\Delta y_t = \varepsilon_t \quad (2.183)$$

Δy_t serisi durağan ise serinin birinci dereceden bütünleşik ya da durağan olduğu ifade edilir ve $I(1)$ ile temsil edilir. Zaman serileri genellikle birinci dereceden durağandır (Evans, 2003, s.238). Bununla birlikte y_t serisi birinci dereceden de durağan değilse, 2.184, 2.185, 2.186 ve 2.187 no.lu denklemlerde gösterildiği gibi serinin ikinci farkı alınarak durağanlığı araştırılır.

$$\Delta y_t - \Delta y_{t-1} = \varepsilon_t \quad (2.184)$$

$$(y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = \varepsilon_t \quad (2.185)$$

$$y_t - 2y_{t-1} + y_{t-2} = \varepsilon_t \quad (2.186)$$

$$\Delta^2 y_t = \varepsilon_t \quad (2.187)$$

Serinin durağanlığı sağlanana kadar fark alma işlemi yapılabilir. Fark alma işleminin sonunda durağanlık sağlanmakla birlikte, seri bir hareketli ortalama (MA) sürecine dönüşür (Brooks, 2008, s.326).

Durağan olmayan serilerin yukarıda gösterildiği gibi kendi gecikmelerine göre farklarının alınarak durağanlaştırılması “fark durağan süreç” (difference stationary process) olarak adlandırılır (Gujarati ve Porter, 2009, s.745).

Diğer taraftan t' trendi temsil etmek üzere, bir y_t serisi 2.88 no.lu denklemde gösterildiği gibi deterministik bir trend içeriyor olabilir.

$$y_t = \beta_0 + \beta_1 t' + \varepsilon_t \quad (2.188)$$

Ancak söz konusu deterministik trend durağan olmayabilir. Bu durumda serinin birinci farkı alınır, modelin MA(1) sürecine dönüşmesi karşılığında durağanlık sağlanır (Brooks, 2008, s.323).

$$y_t = \beta_0 + \beta_1 t' + \varepsilon_t \quad (2.189)$$

$$y_{t-1} = \beta_0 + \beta_1 (t'-1) + \varepsilon_{t-1} \quad (2.190)$$

2.189 no.lu denklem ve 2.190 no.lu denklem çıkarılarak seri durağanlaştırılır ve 2.191 no.lu denklemde gösterildiği üzere MA(1) süreci elde edilir.

$$\Delta y_t = \beta_1 + \varepsilon_t - \varepsilon_{t-1} \quad (2.191)$$

Durağan olmayan deterministik bir trende sahip serinin trendden arındırılarak durağanlaştırılması (detrending) “trend durağan süreç” (trend stationary process) olarak adlandırılır (Gujarati ve Porter, 2009, s.745).

Sabit içeren rassal yürüyüş modeli ile trendin birleşiminden oluşan, aşağıda sunulan 2.192 no.lu modelin ise iki farklı durumu vardır.

$$y_t = \beta_0 + \beta_1 t' + \rho y_{t-1} + \varepsilon_t \quad (2.192)$$

İlk durumda $\beta_0 \neq 0$ $\beta_1 \neq 0$ ve $\rho=1$ iken y_t serisi durağan olmamakla birlikte, y_t serisinin birinci dereceden bütünleşik, diğer bir ifadeyle fark durağan olması halinde, deterministik bir trend etrafında salınımlar yaptığı söylenebilir (Evans, 2003, s.239; Gujarati ve Porter, 2009, s.745-746). Söz konusu durum 2.193 no.lu denklemle ifade edilmiştir.

$$\Delta y_t = \beta_0 + \beta_1 t' + \varepsilon_t \quad (2.193)$$

İkinci durumda ise $\beta_0 \neq 0$ $\beta_1 \neq 0$ ve $\rho < 1$ iken y_t serisi durağandır ve sıfırıncı dereceden bütünleşik y_t serisi deterministik bir trend etrafında salınımlar yapmaktadır. Söz konusu durum 2.194 no.lu denklemle temsil edilmiştir.

$$y_t = \beta_0 + \beta_1 t' + \rho y_{t-1} + \varepsilon_t \quad (2.194)$$

$\beta_0 \neq 0$ $\beta_1 \neq 0$ ve $\rho < 1$ iken 2.63 no.lu denklem durağan bir AR(1) süreci ile deterministik bir trendin birleşimini ifade eder (Gujarati ve Porter, 2009, s.746).

2.3. Durağanlığın Araştırılması

Bir zaman serisinin durağan olup olmadığını anlamak için serinin doğrudan grafiği incelenebileceği gibi otokorelasyon fonksiyonu da analiz edilebilir. Ancak otokorelasyon fonksiyonu üzerinden durağanlığın araştırılması daha güvenilir ve yaygın kullanılan bir yöntemdir. Benzer şekilde, aralarında farklılıklar olmakla birlikte, Q istatistiği ya da Dickey-Fuller testi gibi daha formal teknikler de sıklıkla kullanılır. Özellikle de paket programlarda ya da Python kütüphanelerinde otokorelasyon fonksiyonu analizi, Q istatistiği ve Dickey-Fuller testi gibi yöntemlerle oldukça yaygın bir şekilde karşılaşılır.

2.3.1. Otokorelasyon Fonksiyonu (ACF)

Bir değişkenin kendinden önceki gecikmesi ile olan doğrusal ilişkinin yönünü ve gücünü temsil eden otokorelasyon, zamandan bağımsız k gecikme uzunluğuna sahip kovaryansının varyansına bölümüyle elde edilir (Box ve

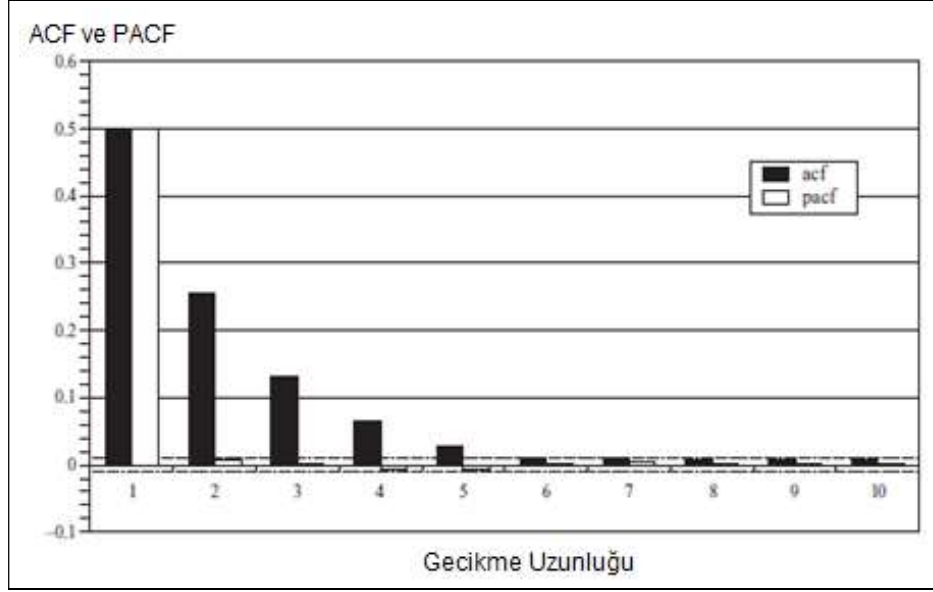
diğerleri, 2016, s.24-25). Serinin beklenen değeri μ olmak üzere, otokorelasyon 2.195 no.lu denklemde gösterildiği gibi ifade edilir.

$$\rho_k = \frac{\sum_{t=1}^T (y_t - \mu)(y_{t+k} - \mu)}{\sum_{t=1}^T (y_t - \mu)^2} \quad (2.195)$$

γ_k , serinin k gecikme uzunluğunda otokovaryansını ve γ_0 varyansını temsil etmek üzere, 2.195 no.lu eşitlik, 2.196 no.lu denklemde gösterildiği gibi daha sade ifade edilebilir.

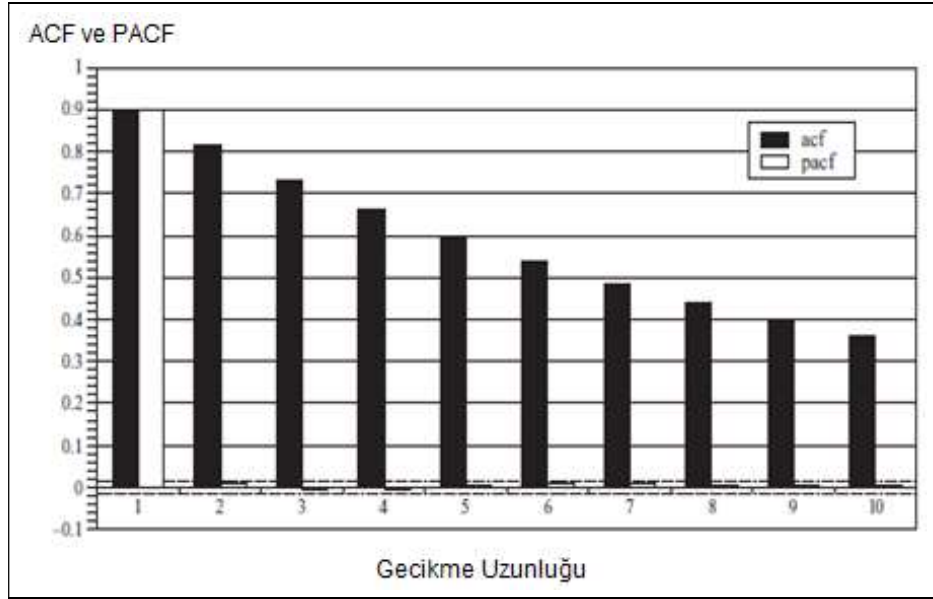
$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (2.196)$$

Belirli bir gecikme uzunluğunda hesaplanan otokorelasyon değerlerinin grafik üzerinde gösterimi, otokorelasyon fonksiyonu (autocorrelation function, ACF) ya da korelogram (correlogram) olarak adlandırılır (Hanke ve Wichern, 2014, s.41). Otokorelasyon, korelogram üzerinde serinin durağanlığı konusunda kullanışlı bilgiler sağlar. Otokorelasyon değerleri 0'ın etrafından salınım yapıyorsa serinin durağan olduğu çıkarımı yapılabilir. Eğer 1 veya 1'e yakın otokorelasyon değerleri k gecikme uzunluğuna doğru yavaş bir şekilde azalıyorsa, serinin durağan olmadığına; hızlı bir şekilde azalıyorsa da serinin durağan olduğuna işaret eder (Montgomery ve diğerleri, 2008, s.32-33). Grafik 2.1 ve Grafik 2.2'de dikey ekseninde otokorelasyon değerlerinin, yatay ekseninde ise gecikmelerin yer aldığı korelogramda, durağan ve durağan olmayan birer seri örneği sunulmuştur.



Grafik 2.1. AR(1) İçin ACF ve PACF: Durağan Seri

Kaynak: Brooks, 2008, s.227



Grafik 2.2. AR(1) İçin ACF ve PACF: Durağan Olmayan Seri

Kaynak: Brooks, 2008, s.227

Gecikme uzunluğunun seçiminde Akaike (Akaike information criteria, AIC) ve Schwarz (Schwarz information criteria, SIC) bilgi kriterleri yaygın olarak kullanılmaktadır (Gujarati ve Porter, 2009, s.753). Schwarz bilgi kriteri, literatürde Bayes bilgi kriteri (Bayes information criterion, BIC) olarak da adlandırılır. Her iki bilgi kriteri de modele ilave edilen her parametrede, hata kareleri toplamı (residual sum of squares, RSS) üzerinden modeli cezalandırmaya dayanır. Bu yüzden iki bilgi kriterinde de gecikme (model) seçimi, düşük bilgi kriteri (AIC ya da BIC) değerine göre yapılır. T gözlem

sayısı, $\ln$ doğal logaritma ve k pozitif bir tam sayı olmak üzere, optimum gecikme uzunluğu k için AIC ve SIC/BIC 2.197 ve 2.198 no.lu denklemlerde gösterildiği gibi hesaplanabilir.

$$\text{AIC} \quad : \text{AIC}(k) = \ln \left[\frac{\text{RSS}}{T} \right] + (k+1) \frac{2}{T} \quad (2.197)$$

$$\text{SIC/BIC} \quad : \text{BIC}(k) = \ln \left[\frac{\text{RSS}}{T} \right] + (k+1) \frac{\ln(T)}{T} \quad (2.198)$$

AIC ile BIC arasındaki fark, sadece ikinci terimlerin payında yer alan ifadelerdir. BIC denkleminin payında yer alan $\ln(T)$ ifadesinden dolayı, örneklem büyüdükçe model AIC yöntemine kıyasla daha fazla cezalandırılır. Bunun bir sonucu olarak örneklem küçük ise tutarlılık (consistency) açısından SIC, örneklem büyük ise etkinlik (efficiency) bakımından ise AIC daha başarılıdır (Stock ve Watson, 2020, s.579-580; Montgomery ve diğerleri, 2008, s.59).

Durağanlığın araştırılmasında kullanılacak maksimum gecikme uzunluğu için 2.199 no.lu denklemle ifade edilen, Schwert (1989, s.151)'in önerdiği yöntem de kullanılabilir.

$$k=12 \left(\frac{T}{100} \right)^{\frac{1}{4}} \quad (2.199)$$

2.3.2. Q İstatistiği ya da Ljung–Box (LB) İstatistiği

Otokorelasyonun beyaz gürültüye (white noise) uyumunu ölçen, “portmanteau” testler kategorisinde yer alan Q istatistiği, Box ve Pierce (1970) tarafından geliştirilmiştir. Q istatistiği, belirli bir gecikme uzunluğunda, serinin tüm ρ_k değerlerinin aynı anda 0 değerine eşitliğini test eder (Montgomery ve diğerleri, 2008, s.56-57).

$$Q=T \sum_{k=1}^m \rho_k^2 \quad (2.200)$$

2.200 no.lu denklemde k gecikme uzunluğunu, ρ_k otokorelasyon parametresini, T gözlem sayısını ve m ise maksimum gecikme uzunluğunu ifade eder. Q istatistiği büyük örneklerde m serbestlik derecesi ile χ^2 (ki-kare) dağılımına uyar.

Ljung ve Box (1978) ise Q istatistiğini, küçük örneklerde de m serbestlik derecesi ile χ^2 dağılımına uyacak şekilde geliştirmişlerdir. Literatürde ve uygulamada Q istatistiği olarak genellikle Ljung ve Box tarafından geliştirilen 2.201 no.lu denklem kullanılır.

$$Q = T(T+2) \sum_{k=1}^m \frac{\rho_k^2}{T-k} \quad (2.201)$$

Q istatistiğiyle otokorelasyon parametrelerinin tümünün 0 değerine eşitliğini sınamak için hipotez testi 2.202 ve 2.203 no.lu ifadelerde gösterildiği gibi kurulur.

$$h_0 : \rho_1 = \rho_2 = \rho_3 = \dots = \rho_k = 0 \quad (2.201)$$

$$h_1 : \rho_1 ; \rho_2 ; \rho_3 ; \dots ; \rho_k \neq 0 \quad (2.202)$$

Boş hipotezi temsil eden h_0 , serinin tüm ρ_k değerlerinin 0 değerine eşit olduğunu, alternatif hipotezi temsil eden h_1 ise ρ_k değerlerinden en az birinin 0 değerinden farklı olduğunu ifade eder.

Q istatistiği, χ^2 değerinden büyük ise boş hipotez reddedilir. Boş hipotezin reddi serinin otokorelasyon katsayılarının tümünün birden 0 değerine eşit olmadığına, en az bir ρ_k değerinin 0 değerinden farklı olduğuna işaret eder.

Q istatistiği, α anlamlılık seviyesinde ve m serbestlik derecesinde ki-kare χ^2 değerinden küçük ise boş hipotez kabul edilir. Boş hipotezin kabulü serinin otokorelasyon katsayılarının tümünün birden 0 değerine eşit olduğuna işaret eder (Brooks, 2008, s.209-211; Gujarati ve Porter, 2009, s.753-754).

Q istatistiği serinin birim kökünün varlığını değil, ρ_k değerlerinin doğrudan doğruya 0 değerine eşitliğini, diğer bir ifadeyle “güçlü durağanlığı”

(strictly stationary) test eder. Q istatistiğine göre boş hipotez reddedilse bile ρ_k değerleri “zayıf durağanlık” için yeterli sayılabilecek -1 ile 1 arasında 0’a yakın bir değere sahip olabilir.

2.3.3. Genişletilmiş Dickey-Fuller Birim Kök Testi

Bir seride birim kökün varlığını araştıran Dickey-Fuller birim kök testi, “portmanteau” testlerin aksine güçlü durağanlığı değil, zayıf durağanlığı sınar.

$$y_t = \rho y_{t-1} + \varepsilon_t \quad (2.203)$$

Rassal yürüyüş modelini ve aynı zamanda AR(1) sürecini ifade eden 2.203 no.lu eşitlikte ρ parametresi otokorelasyon katsayısıdır ve $-1 \leq \rho \leq 1$ değerleri arasındadır. ε_t ise problemsiz hata terimini temsil etmektedir. Seride birim kök olasılığı göz önüne alınarak doğrudan ρ parametresi test edilmez. Bunun yerine 2.204 ve 2.205 no.lu denklemlerde gösterildiği üzere eşitliğin her iki tarafından da y_{t-1} ifadesi çıkarılarak y_t serisinin birinci farkı elde edilir (Gujarati ve Porter, 2009, s.754).

$$y_t - y_{t-1} = \rho y_{t-1} - y_{t-1} + \varepsilon_t \quad (2.204)$$

$$\Delta y_t = (\rho - 1)y_{t-1} + \varepsilon_t \quad (2.205)$$

$\delta = \rho - 1$ olmak üzere 2.205 no.lu eşitlik 2.206 no.lu denklemle ifade edilebilir.

$$\Delta y_t = \delta y_{t-1} + \varepsilon_t \quad (2.206)$$

Birim kökün varlığını sınamak için hipotez testi 2.207 ve 2.208 no.lu ifadelerde gösterildiği gibi sol kuyruk testi şeklinde kurulur (Hill ve diğerleri, 2011, s.484).

$$h_0 : \delta = 0 \quad (2.207)$$

$$h_1 : \delta < 0 \quad (2.208)$$

Boş hipotezi temsil eden h_0 , $\delta = 0$ iken $\delta = \rho - 1$ olduğundan serinin birim kökünün olduğunu, alternatif hipotezi temsil eden h_1 ise serinin birim kökünün olmadığını ifade eder.

Dickey-Fuller test istatistiği standart t-istatistiği yöntemleriyle hesaplanmakla birlikte, büyük örnekleme dahi normal dağılıma uymaz. Bu yüzden kritik değerler için Monte Carlo simülasyonlarından hesaplanan τ (tau) istatistiğinden yararlanılır (Stock ve Watson, 2020, s.586).

t-istatistiğinin mutlak değeri, τ kritik değerinin mutlak değerinden büyük ise boş hipotez reddedilir. Boş hipotezin reddi seride birim kök bulunmadığına, serinin durağan olduğuna işaret eder. t-istatistiğinin mutlak değeri, τ kritik değerinin mutlak değerinden küçük ise boş hipotez kabul edilir. Boş hipotezin kabulü ise seride birim kök bulunduğuna, serinin durağan olmadığına işaret eder (Brooks, 2008, s.328).

Dickey-Fuller birim kök testi rassal yürüyüş modelinin sabit içermeyen ve içeren formları ile trend içeren farklı formları olduğundan 2.209, 2.210 ve 2.211 no.lu denklemlerde gösterildiği üzere üç farklı şekilde uygulanır (Gujarati ve Porter, 2009, s.755).

$$\text{Sabitli} : \Delta y_t = \delta y_{t-1} + \varepsilon_t \quad (2.209)$$

$$\text{Sabitli} : \Delta y_t = \beta_0 + \delta y_{t-1} + \varepsilon_t \quad (2.210)$$

$$\text{Trendli} : \Delta y_t = \beta_0 + \beta_1 t + \delta y_{t-1} + \varepsilon_t \quad (2.211)$$

Dickey-Fuller birim kök testi, hata terimleri arasında otokorelasyon olmadığı, hata terimlerinin problemsiz olduğu varsayımı yapar. Ancak hata terimleri arasında otokorelasyon sorununun varlığı durumunda, t-istatistikleri yüksek hesaplanacağından (oversized), kabul edilmesi gereken boş hipotezin yanlış bir şekilde reddedilmesi söz konusu olur. Diğer bir ifadeyle hata terimleri arasında otokorelasyon sorununun varlığı, durağan olmayan bir serinin yanlış bir şekilde durağan olarak kabul edilmesine yol açar. Hata terimleri arasında olası otokorelasyon sorununun çözümü amacıyla, p bağımlı değişkenin gecikme uzunluğu olmak üzere, yukarıdaki üç farklı modele de bağımlı

değişkenin birinci farklarının p kadar gecikmeleri 2.212, 2.213 ve 2.214 no.lu denklemlerde gösterildiği şekilde ilave edilir (Brooks, 2008, s.329).

$$\text{Sabitsiz} : \Delta y_t = \delta y_{t-1} + \sum_{i=1}^p \phi_i \Delta y_{t-i} + u_t \quad (2.212)$$

$$\text{Sabitli} : \Delta y_t = \beta_0 + \delta y_{t-1} + \sum_{i=1}^p \phi_i \Delta y_{t-i} + u_t \quad (2.213)$$

$$\text{Trendli} : \Delta y_t = \beta_0 + \beta_1 t' + \delta y_{t-1} + \sum_{i=1}^p \phi_i \Delta y_{t-i} + u_t \quad (2.214)$$

u_t olası otokorelasyon sorunun giderildiği problemsiz hata terimi olmak üzere, uygulamada Dickey-Fuller test istatistiği olarak genişletilmiş Dickey-Fuller (Augmented Dickey-Fuller, ADF) test istatistiği kullanılır. Her üç modelin birim kök testinde de δ parametresinin 0'a eşitliği test edilir. Ayrıca $\delta=0$ eşitliğinin sınanmasında kullanılan kritik değerler üç modelde de birbirinden farklıdır. Gecikme uzunluğu p'nin belirlenmesi amacıyla Akaike, Schwarz gibi bilgi kriterleri kullanılabilir (Gujarati ve Porter, 2009, s.756-757).

2.4. Eşbütünleşme ve Genişletilmiş Engle-Granger Testi

Eşbütünleşme (cointegration), bağımlı ve bağımsız değişkenler ayrı birer seri olarak durağan olmasalar da ortak bir stokastik trendi paylaşması olarak tanımlanabilir. Seriler arasındaki uzun dönem ilişkiye işaret eden eşbütünleşme durumunda, seriler durağan olmamasına rağmen sahte regresyon olgusundan söz edilemez. Eşbütünleşme ilişkisi yaklaşımının ve testinin geliştiricilerinden Granger, eşbütünleşme testinin sahte regresyon probleminden kaçınmak için bir ön test olarak düşünülebileceğini ifade etmiştir (Hill ve diğerleri, 2011, s.488-489; Gujarati ve Porter, 2009, s.762).

Eşbütünleşme olgusu, 2.215 ve 2.216 no.lu denklemlerde gösterildiği üzere serilerin birinci farklarının alınarak durağanlaştırılmasına benzer.

$$y_t = y_{t-1} + \varepsilon_t \quad (2.215)$$

$$\Delta y_t = \varepsilon_t \quad (2.216)$$

y_t ve x_t serilerinin durağan olmadığı varsayımında, y_t serisinin bağımlı ve x_t serisinin bağımsız değişken olduğu bir modelde, y_t serisinin 2.216 no.lu denklemle temsil edilen durağan formuna benzer, 2.217 no.lu denklemde gösterildiği gibi y_t ve x_t serilerinin doğrusal kombinasyonundan oluşan farklı bir durağan form olabilir.

$$y_t - x_t = u_t \quad (2.217)$$

2.217 no.lu eşitlikte u_t problemsiz hata terimini temsil etmektedir. y_t ve x_t serilerinin birinci (aynı) dereceden bütünüleşik (durağan) olduğu varsayımında, y_t ve x_t arasındaki bağlantıya işaret eden $y_t - x_t$ ifadesinin doğrusal kombinasyonu, sıfırıncı dereceden “eşbütünüleşme ilişkisi” (cointegrating relationship) olarak adlandırılır (Hanke ve Wichern, 2014, s.321-322). Benzer şekilde y_t ve x_t serilerinin ikinci (aynı) dereceden bütünüleşik (durağan) olduğu varsayımında, $y_t - x_t$ ifadesinin doğrusal kombinasyonu, birinci dereceden eşbütünüleşme ilişkisine işaret eder.

Eşbütünüleşme ilişkisinin sınanması amacıyla önerilen Engle-Granger testi, Dickey-Fuller testinde olduğu gibi sabit içermeyen ve içeren formları ile trend içeren üç farklı formu 2.218, 2.219 ve 2.220 no.lu denklemlerle gösterilmiştir (Hill ve diğerleri, 2011, s.489).

$$\text{Sabitsiz} \quad : y_t = \beta_1 x_t + \varepsilon_t \quad (2.218)$$

$$\text{Sabitli} \quad : y_t = \beta_0 + \beta_1 x_t + \varepsilon_t \quad (2.219)$$

$$\text{Trendli} \quad : y_t = \beta_0 + \varphi t + \beta_1 x_t + \varepsilon_t \quad (2.220)$$

ε_t problemsiz hata terimi olmak üzere, y_t ve x_t serilerinden oluşan modelin hata terimleri durağan ise serilerin eşbütünüleşik olduğu çıkarımı yapılır. Ancak hata terimleri doğrudan gözlemlenemediğinden, 2.218, 2.219 ve 2.220 no.lu modeller yardımıyla hata terimlerinin tahmin edilmesi gerekir. Söz konusu tahmin değerleri 2.221, 2.222 ve 2.223 no.lu denklemlerle temsil edilmiştir.

$$\text{Sabitsiz} : \hat{\varepsilon}_t = y_t - \hat{\beta}_1 x_t \quad (2.221)$$

$$\text{Sabitli} : \hat{\varepsilon}_t = y_t - \hat{\beta}_0 - \hat{\beta}_1 x_t \quad (2.222)$$

$$\text{Trendli} : \hat{\varepsilon}_t = y_t - \hat{\beta}_0 - \hat{\phi}t' - \hat{\beta}_1 x_t \quad (2.223)$$

Tahmin edilen hata terimlerinin durağanlığı, 2.224, 2.225 ve 2.226 no.lu denklemler yardımıyla Dickey-Fuller testi kullanılarak test edilir.

$$\hat{\varepsilon}_t = \rho \hat{\varepsilon}_{t-1} + u_t \quad (2.224)$$

$$\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1} = (\rho - 1) \hat{\varepsilon}_{t-1} + u_t \quad (2.225)$$

$$\Delta \hat{\varepsilon}_t = \delta \hat{\varepsilon}_{t-1} + u_t \quad (2.226)$$

Hata terimlerinin beklenen değeri 0 olduğundan, Engle-Granger testine konu 2.224, 2.225 ve 2.226 no.lu modellerde sabit terim bulunmamaktadır. Hata terimlerinde birim kökün varlığı, y_t ve x_t serileri arasında eşbütünleşme olmadığını, yokluğu ise seriler arasında eşbütünleşme olduğunu ifade eder. Birim kökün varlığını sınamak için hipotez testi, 2.227 ve 2.228 no.lu ifadelerde gösterildiği üzere, Dickey-Fuller testinde olduğu gibi sol kuyruk şeklinde kurulur (Hill ve diğerleri, 2011, s.489).

$$h_0 : \delta = 0 \quad (2.227)$$

$$h_1 : \delta < 0 \quad (2.228)$$

Boş hipotezi temsil eden h_0 , $\delta = 0$ iken $\delta = \rho - 1$ olduğundan hata terimlerinde birim kökün olduğunu, alternatif hipotezi temsil eden h_1 ise hata terimlerinde birim kökün olmadığını ifade eder.

Engle-Granger testi, hata terimlerinin tahmin edilen değerlerine dayandığından, testin kritik değerleri Dickey-Fuller testinin kritik değerlerinden farklıdır. Eşbütünleşme testinde, Engle-Granger tarafından hesaplanan kritik değerler kullanılır (Gujarati ve Porter, 2009, s.763).

t-istatistiğinin mutlak değeri, Engle-Granger test istatistiğini ifade eden τ kritik değerinin mutlak değerinden büyük ise boş hipotez reddedilir. Boş

hipotezin reddi hata terimlerinde birim kökün bulunmadığına, y_t ve x_t serilerinin eşbütünleşik olduğuna işaret eder. t-istatistiğinin mutlak değeri, τ kritik değerinin mutlak değerinden küçük ise boş hipotez kabul edilir. Boş hipotezin kabulü hata terimlerinde birim kökün bulunduğuna, y_t ve x_t serilerinin eşbütünleşik olmadığına işaret eder.

Engle-Granger testi, Dickey-Fuller testinde olduğu gibi hata terimleri arasında otokorelasyon olmadığı, hata terimlerinin problemsiz olduğu varsayımı yapar. Hata terimleri arasında olası otokorelasyon sorununun çözümü amacıyla, p (bağımlı değişkeni temsil eden) tahmin edilen hata teriminin gecikme uzunluğu olmak üzere, eşbütünleşme testinde kullanılan modele, tahmin edilen hata teriminin birinci farklarının p kadar gecikmeleri 2.229 no.lu denklemde gösterildiği şekilde ilave edilir (Hill ve diğerleri, 2011, s.489-490).

$$\Delta \hat{\varepsilon}_t = \delta \hat{\varepsilon}_{t-1} + \sum_{i=1}^p \phi_i \Delta \hat{\varepsilon}_{t-i} + u_t \quad (2.229)$$

u_t olası otokorelasyon sorununun giderildiği problemsiz hata terimi olmak üzere, uygulamada Engle-Granger test istatistiği olarak genişletilmiş Engle-Granger (Augmented Engle-Granger, AEG) test istatistiği kullanılır (Gujarati ve Porter, 2009, s.763). Ayrıca $\delta=0$ eşitliğinin sınanmasında kullanılan kritik değerler Dickey-Fuller testinde olduğu gibi üç modelde de birbirinden farklıdır (Hill ve diğerleri, 2011, s.489).

2.5. Zaman Serisi Modelleri

Zaman serileri modelleri esas olarak otoregresif model ve hareketli ortalama modellerinin analizine dayanır. Otoregresif model (autoregressive model, AR), AR süreci; hareketli ortalama modeli (moving average model, MA) ise MA süreci olarak da adlandırılır. Bu iki model, bir arada tek bir model olarak tasarlandığında ARMA (autoregressive moving average model) ismini alır. Serinin durağanlığına ilişkin bilginin modele dahil edilmesiyle, model ARIMA (integrated autoregressive moving average model) olarak adlandırılan modele dönüşür. ARIMA modeline 3, 6, 12, aylık ya da haftalık vb. (esasen özel birer

AR ve MA süreci olarak ifade edilebilen) periyodik gecikmelerin modele dahil edilmesiyle son derece başarılı bir model olan SARIMA (seasonal integrated autoregressive moving average model) modeli elde edilir. Son olarak SARIMA modeline, (bağımlı değişken niteliğindeki) serinin dışında (bağımsız değişken niteliğinde) yeni bir ya da daha fazla seri ilave edilirse SARIMAX (seasonal integrated autoregressive moving average model with exogenous variables) olarak adlandırılan son derece kompleks bir model elde edilir. Bunların yanında ARCH (autoregressive conditional heteroskedasticity) ve GARCH (generalized autoregressive conditional heteroskedasticity) gibi varyansa odaklanan zaman serisi modelleri de yaygın olarak kullanılmaktadır. Bu çalışmada zaman serilerinin en temel yapıları olan AR ve MA süreçleri ile ARMA, ARIMA, SARIMA ve SARIMAX modelleri üzerinde durulmuştur.

2.5.1. Otoregresif Model (AR)

Otoregresif model, “Durağanlık ve Birim Kök Analizi” konusunda detaylı olarak ele alındığı için bu bölümde kısaca özetlenerek önemli görülen bazı konulara değinilecektir.

Otoregresif model, serinin durağan olması koşuluyla, bir değişkenin t dönemindeki değerinin, $t-1$, $t-2$ gibi kendi geçmiş dönem değerlerinin doğrusal bir fonksiyonuna göre koşullu beklenen değeri olarak ifade edilebilir. Gecikme sırası p olmak üzere, p . gecikmeye sahip bir otoregresif model $AR(p)$ şeklinde ifade edilir.

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.230)$$

2.230 no.lu modelde yer alan y_t ifadesi bağımlı değişkeni, y_{t-1} , y_{t-2} gibi y_t değişkeninin gecikmeleri bağımsız değişkenleri ve ϕ_1 , ϕ_2 gibi ifadeler ise parametreleri temsil eder. Modelin en sağında bulunan ε_t terimi ise doğrusal modelin tüm varsayımlarını sağlayan problemsiz hata (white noise) terimini ifade etmektedir. Bir AR modeli, 2.231 no.lu denklemde gösterildiği üzere, y_t değişkeninin beklenen değerini ifade eden ϕ_0 gibi bir sabit terim/parametre de içerebilir (Stock ve Watson, 2020, s.565-566; Pindyck ve Rubinfeld, 1991, s.478).

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.231)$$

2.231 no.lu eşitlik, p gecikme uzunluğu ve i gecikme sırası olmak üzere, toplam sembolü kullanılarak 2.232 no.lu denklemle daha sade ifade edilebilir.

$$y_t = \phi_0 + \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \varepsilon_t \quad (2.232)$$

Eğer gecikme uzunluğu p=1 ise AR(1) süreci olarak adlandırılır ve 2.233 no.lu denklemle ifade edilir.

$$y_t = \phi_0 + \phi_1 y_{t-1} + \varepsilon_t \quad (2.233)$$

Gecikme uzunluğu p=2 ise AR(2) süreci olarak adlandırılır ve 2.234 no.lu denklemle ifade edilir.

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t \quad (2.234)$$

Modelin gecikmelerini, diğer bir ifadeyle AR sürecini belirlemek için gecikme uzunluğunun seçiminde, Akaike (AIC) ve Schwarz/Bayes (SIC/BIC) bilgi kriterleri yaygın olarak kullanılır (Stock ve Watson, 2020, s.579-580).

2.5.2. Hareketli Ortalama Modeli (MA)

Hareketli ortalama modeli (ya da süreci), bir değişkenin t dönemindeki değerinin, t-1, t-2 gibi serinin geçmiş dönem problemsiz hata terimlerinin doğrusal bir fonksiyonu olduğunu temsil eder (Brooks, 2008, s.211). Burada yer alan “hareketli ortalama modeli” terimi, istatistikte kullanılan (aritmetik) hareketli ortalama ile farklı bir kavrama, diğer bir ifadeyle problemsiz hata terimlerinin mevcut ve geçmiş dönem doğrusal kombinasyonlarının zaman içinde değişmesine işaret eder (Hanke ve Wichern, 2014, s.361).

Modelin derecesi q olmak üzere, q. dereceye sahip bir hareketli ortalama modeli MA(q) şeklinde 2.235 no.lu denklemde gösterildiği gibi ifade edilir.

$$y_t = \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.235)$$

2.105 no.lu modelde yer alan y_t ifadesi bağımlı değişkeni, ε_{t-1} , ε_{t-2} gibi problemsiz hata teriminin gecikmeleri bağımsız değişkenleri ve θ_1 , θ_2 gibi ifadeler ise parametreleri temsil eder. Modelin en sağında bulunan ε_t terimi ise AR(p) sürecinde olduğu gibi doğrusal modelin tüm varsayımlarını sağlayan problemsiz hata (white noise) terimini ifade etmektedir. Bir MA modeli (bağımsız değişkenlerin y_t üzerindeki etkisi yokken) y_t değişkeninin beklenen değerini ifade eden θ_0 gibi bir sabit terim/parametre de içerebilir (Brooks, 2008, s.211).

$$y_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.236)$$

2.236 no.lu eşitlik toplam sembolü kullanılarak 2.237 no.lu denklemle daha sade ifade edilebilir.

$$y_t = \theta_0 + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \quad (2.237)$$

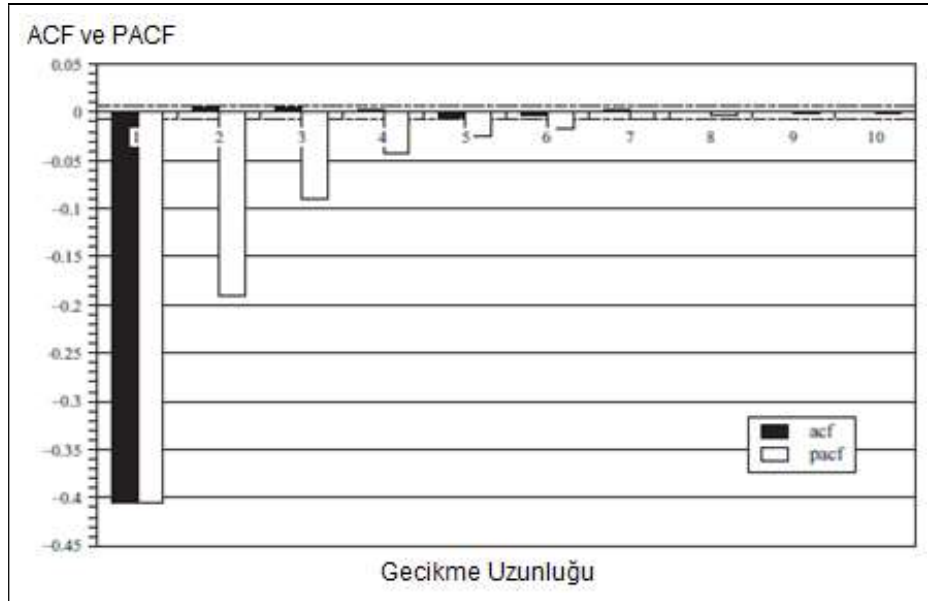
AR sürecinin aksine bir MA süreci, θ_0 , θ_1 , θ_2 gibi parametrelerin değerlerinden bağımsız olarak her zaman durağandır (Montgomery ve diğerleri, 2008, s.235).

AR süreci, serinin geçmiş değerlerinin ağırlıklandırıldığı bir model olmasına rağmen, MA süreci ise dışsal şoklardan etkilenen bir model olarak düşünülebilir. Bu yüzden oynaklığın çok az olduğu pürüzsüz (smooth) bir seri, AR modeli için çok yüksek, MA modeli için çok düşük parametre değerlerine sahip olurken; oynaklığın çok fazla olduğu belirgin şoklara maruz kalan bir seri, AR modeli için çok düşük, MA modeli için çok yüksek parametre değerlerine sahip olacaktır. Bağımlı değişkeni birkaç dönem etkileyen dışsal bir şok, MA sürecinin derecesini ifade eden q değerinin birden büyük olacağına işaret edebilir. Benzer şekilde serinin frekansı arttıkça, diğer bir ifadeyle serinin gözlem aralığı yıllık, aylık, haftalık gibi azaldıkça, MA sürecinin q değeri artabilir. Örneğin aylık gözlemlere dayanan bir seriyle modellenen bir MA

sürecinin q değeri, yıllık gözlemlere dayanan aynı seriyle modellenen MA sürecinin q değerinden daha fazla olacaktır (Evans, 2003, s.236).

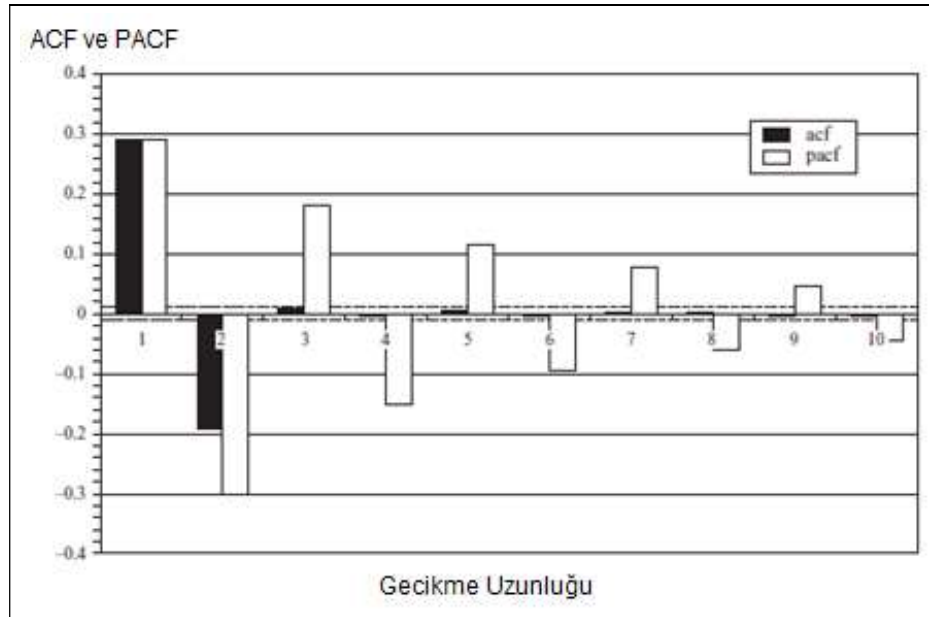
MA(q) sürecinin q . gecikmeden sonra kovaryansının 0 olması, kısa dönemli bir hafızaya sahip olduğu anlamına gelir. Örneğin MA(1) sürecinde, y_t dönemi y_{t-1} dönemiyle bir korelasyona sahip olduğundan, sadece y_{t+1} dönemi tahmin edilebilir. Çünkü MA(1) süreci $t-1$ döneminden önceki şokları unuttur. Bu yüzden MA sürecinin geleceğe ilişkin öngörü kabiliyeti sadece birkaç dönemle sınırlıdır (Pindyck ve Rubinfeld, 1991, s.475).

MA sürecinin derecesini ifade eden q değeri, otokorelasyon fonksiyonu (ACF) incelenerek belirlenebilir. Korelogramda, otokorelasyon değerinin aniden 0'a yaklaştığı gecikmeden bir önceki gecikme q değeri olarak kabul edilebilir. Kısmi otokorelasyon fonksiyonu (PACF) ise AR sürecinin aksine MA sürecinde aşamalı bir şekilde 0 değerine yakınsadığından, q değerini belirlemede kullanışlı değildir (Chatfield, 1996, s.58).



Grafik 2.3. MA(1) için ACF ve PACF

Kaynak: Brooks, 2008, s.226



Grafik 2.4. MA(2) için ACF ve PACF

Kaynak: Brooks, 2008, s.226

Örneğin Grafik 2.3'te yer alan korelagram (ACF), otokorelasyon değerinin birinci gecikmeden sonra aniden 0 değerine yaklaşması MA(1) sürecine; Grafik 2.4'te yer alan korelagram ise otokorelasyon değerinin ikinci gecikmeden sonra aniden 0 değerine yaklaşması MA(2) sürecine işaret eder.

2.5.2.1. MA(q) Modelinin Türetilmesi

MA sürecinde, problemsiz hata teriminin gecikmelerini ifade eden ε_{t-1} , ε_{t-2} gibi bağımsız değişkenler gözlemlenebilir değildir. Gözlemlenebilir olmayan değişkenlerden de bir MA modeli tahmin edilemez. Bununla birlikte durağan bir AR(1) sürecinden sonsuz gecikmeli bir MA modeli elde edilebilir (Evans, 2003, s.233-235). AR(1) sürecinin MA(q) sürecine dönüşümüne ilişkin işlemler aşağıda gösterilmiştir.

$$y_t = \phi_0 + \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \varepsilon_t \quad (2.238)$$

$p=1$ iken AR(1) süreci için 2.238 no.lu denklem kullanılarak 2.239 no.lu denklem elde edilir.

$$y_t = \phi_0 + \phi_1 y_{t-1} + \varepsilon_t \quad (2.239)$$

$y_{t-1} = \phi_0 + \phi_1 y_{t-2} + \varepsilon_{t-1}$ olduğundan, 2.239 no.lu eşitlikte yerine yazılarak 2.240 ve 2.241 no.lu denklemler elde edilir.

$$y_t = \phi_0 + \phi_1 (\phi_0 + \phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \quad (2.240)$$

$$y_t = \phi_0 + \phi_1 \phi_0 + \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.241)$$

$y_{t-2} = \phi_0 + \phi_1 y_{t-3} + \varepsilon_{t-2}$ olduğundan, 2.241 no.lu eşitlikte yerine yazılarak 2.242 ve 2.243 no.lu denklemler elde edilir.

$$y_t = \phi_0 + \phi_1 \phi_0 + \phi_1^2 (\phi_0 + \phi_1 y_{t-3} + \varepsilon_{t-2}) + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.242)$$

$$y_t = \phi_0 + \phi_1 \phi_0 + \phi_1^2 \phi_0 + \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.243)$$

Yukarıdaki süreç T sonsuza giderken T kez yinelenirse, 2.244 no.lu denkleme ulaşılır.

$$y_t = \phi_0 + \phi_1 \phi_0 + \phi_1^2 \phi_0 + \dots + \phi_1^{T-1} \phi_0 + \phi_1^T y_{t-T} + \phi_1^{T-1} \varepsilon_{t-(T-1)} + \dots + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.244)$$

2.244 no.lu eşitlikte yer alan ve tümü birer sabiti ifade eden ϕ_0 ve ϕ_1 parametreleriyle, bunların kombinasyonundan oluşan tüm parametrelerin θ_0 gibi bir sabite eşit olduğu kabul edilebilir. Bu durumda 2.244 no.lu eşitlik, 2.245 no.lu denklemle daha sade ifade edilebilir.

$$y_t = \theta_0 + \phi_1^T y_{t-T} + \phi_1^{T-1} \varepsilon_{t-(T-1)} + \dots + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.245)$$

AR(1) sürecinin durağanlık koşulunu ifade eden $-1 \leq \phi_1 \leq 1$ geçerli iken ϕ_1^T ifadesi 0 değerine yakınsayacağından 2.245 no.lu eşitlik, sağ tarafta bulunan son y_{t-T} ifadesinden de kurtarılmış olur.

$$y_t = \theta_0 + \phi_1^{T-1} \varepsilon_{t-(T-1)} + \dots + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.246)$$

$q = T-1$ olmak üzere, $\phi_1 = \theta_1$, $\phi_1^2 = \theta_2$, ... $\phi_1^{T-1} = \theta_q$ şeklinde 2.246 no.lu MA(q) sürecinde bulunan tüm ε terimlerinin parametreleri 2.247 ve 2.248 no.lu denklemlerde gösterildiği gibi θ parametreleriyle ifade edilebilir.

$$y_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.247)$$

$$y_t = \theta_0 + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \quad (2.248)$$

Böylece MA(q) sürecinin gözlemlenebilir olmayan bağımsız değişkenleri, durağan bir AR(1) süreci yardımıyla yukarıda gösterildiği gibi elde edilir.

2.5.2.2. MA(q) Modelinin Durağanlık Analizi

MA sürecinin durağanlık analizi, cebirsel kolaylık olması amacıyla, genellikle θ_0 sabit terimi modelden çıkartılarak sabitsiz MA süreci üzerinden yapılır.

2.5.2.2.1. MA(q) Modelinin Beklenen Değeri

MA(q) modelinin beklenen değeri sabit olup 0 değerine eşittir.

$$y_t = \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \quad (2.249)$$

$$E[y_t] = E \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + E[\varepsilon_t] \quad (2.250)$$

2.249 ve 2.250 no.lu denklemlerde yer alan θ_i ifadeleri birer sabit ve sabitlerin beklenen değeri kendisine eşittir. ε_t ise problemsiz hata terimi ve problemsiz hata teriminin kendisinin ve gecikmelerinin beklenen değeri birbirine eşit olup 0 değerine eşittir; $E[\varepsilon_t] = E[\varepsilon_{t-1}] = E[\varepsilon_{t-2}] = \dots = E[\varepsilon_{t-q}] = 0$.

$$E[y_t] = \theta_1 0 + \theta_2 0 + \dots + \theta_q 0 + 0 \quad (2.251)$$

$$E[y_t] = 0 \quad (2.252)$$

2.251 ve 2.252 no.lu denklemlerde gösterildiği gibi MA(1), MA(2), MA(3) gibi herhangi bir MA sürecinin beklenen değeri 0 değerine eşittir.

Duranlık açısından değerlendirilecek olursa, MA süreci, beklenen değerin sabit olmasına ilişkin durağanlık koşulunu θ değerlerinden bağımsız olarak sağlar.

2.5.2.2.2. MA(q) Modelinin Varyansı

δ_ϵ^2 ifadesi hata teriminin sabit olan varyansı olmak üzere, MA(q) modelinin varyansı sabit olup $(1+\theta_1^2+\theta_2^2 \dots + \theta_q^2)\delta_\epsilon^2$ değerine eşittir.

Bir serinin karesinin beklenen değeri 2.253 no.lu denklemde gösterildiği gibi varyansı ifade eder.

$$E[y_t^2] = E \left[\left(\sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t \right)^2 \right] \quad (2.253)$$

$$E[y_t^2] = E[\theta_1^2 \epsilon_{t-1}^2 + \theta_2^2 \epsilon_{t-2}^2 + \dots + \theta_q^2 \epsilon_{t-q}^2 + \zeta\zeta + \epsilon_t^2] \quad (2.254)$$

$$E[y_t^2] = \theta_1^2 E[\epsilon_{t-1}^2] + \theta_2^2 E[\epsilon_{t-2}^2] + \dots + \theta_q^2 E[\epsilon_{t-q}^2] + E[\zeta\zeta] + E[\epsilon_t^2] \quad (2.255)$$

2.254 ve 2.255 no.lu denklemlerde yer alan “çapraz çarpımlar” anlamında kullanılan “çç”, hata terimlerinin farklı gecikmelerinin beklenen değerlerinin çapraz çarpımlarını içerdiğinden 0 değerine eşit olacaktır. Söz konusu durum 2.256 ve 2.257 no.lu denklemlerde gösterilmiştir. Bu çerçevede 2.258 no.lu denklem, 2.255 no.lu denklemin sadeleştirilmiş halini temsil eder.

$$E[\zeta\zeta] = \theta_1 \theta_2 E[\epsilon_{t-1}] E[\epsilon_{t-2}] + \theta_1 \theta_3 E[\epsilon_{t-1}] E[\epsilon_{t-3}] + \dots \quad (2.256)$$

$$E[\zeta\zeta] = 0 \quad (2.257)$$

$$E[y_t^2] = \theta_1^2 E[\epsilon_{t-1}^2] + \theta_2^2 E[\epsilon_{t-2}^2] + \dots + \theta_q^2 E[\epsilon_{t-q}^2] + E[\epsilon_t^2] \quad (2.258)$$

Bir serinin karesinin beklenen değeri varyansı temsil ettiğinden, $E[\epsilon_t^2]$ ifadesi de problemsiz hata teriminin (ϵ_{t-1} , ϵ_{t-2} gibi tüm gecikmelerinde sabit olan) δ_ϵ^2 ile temsil edilen varyansını ifade eder; $E[\epsilon_t^2] = E[\epsilon_{t-1}^2] = E[\epsilon_{t-2}^2] = \dots = E[\epsilon_{t-q}^2] = \delta_\epsilon^2$.

y_t serisinin varyansı γ_0 olmak üzere, 2.259 ve 2.260 no.lu denklemlerde gösterildiği gibi sade bir şekilde ifade edilebilir.

$$\gamma_0 = \theta_1^2 \delta_\varepsilon^2 + \theta_2^2 \delta_\varepsilon^2 + \dots + \theta_q^2 \delta_\varepsilon^2 + \delta_\varepsilon^2 \quad (2.259)$$

$$\gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2) \delta_\varepsilon^2 \quad (2.260)$$

MA(1), MA(2), MA(3) gibi MA süreçlerinin varyans analizini basit bir şekilde yapabilmek için 2.260 no.lu denklemle ifade edilen MA(q) sürecinin varyansı, 2.261 no.lu denklemle daha sade olarak ifade edilebilir.

$$\gamma_0 = \delta_\varepsilon^2 \left(1 + \left(\sum_{i=1}^q \theta_i^2 \right) \right) \quad (2.261)$$

2.261 no.lu denklem yardımıyla MA(1), MA(2), MA(3) gibi herhangi bir MA sürecinin varyansı kolaylıkla elde edilebilir. MA(1), MA(2), MA(3) gibi süreçler, 2.261 no.lu denklem kullanılarak 2.262, 2.263 ve 2.264 no.lu denklemlerle ifade edilmiştir.

$$\text{MA(1)} \quad : \quad \gamma_0 = \delta_\varepsilon^2 (1 + \theta_1^2) \quad (2.262)$$

$$\text{MA(2)} \quad : \quad \gamma_0 = \delta_\varepsilon^2 (1 + \theta_1^2 + \theta_2^2) \quad (2.263)$$

$$\text{MA(3)} \quad : \quad \gamma_0 = \delta_\varepsilon^2 (1 + \theta_1^2 + \theta_2^2 + \theta_3^2) \quad (2.264)$$

Duranlık açısından değerlendirilecek olursa, MA süreci, varyansın zamandan bağımsız bir şekilde belirlenmesine, diğer bir ifadeyle varyansın sabit olmasına ilişkin durağanlık koşulunu sağlar.

2.5.2.2.3. MA(q) Modelinin Kovaryansları

k gecikme uzunluğu ve γ_k MA(q) modelinin k. gecikme uzunluğuna sahip kovaryansı (ya da otokovaryansı) olmak üzere, MA(q) sürecinin herhangi bir k gecikmesine ilişkin kovaryansı 2.265 ve 2.266 no.lu denklemlerle ifade edilebilir.

$$\text{Cov}[y_t, y_{t-k}] = \gamma_k = E[y_t y_{t-k}] \quad (2.265)$$

$$\gamma_k = E \left[\left(\left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \right) \left(\left(\sum_{i=1}^q \theta_i \varepsilon_{t-i-k} \right) + \varepsilon_{t-k} \right) \right] \quad (2.266)$$

2.266 no.lu denklem aşağıda gösterildiği gibi açılabilir.

$$\gamma_k = E \left[\left((\theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}) + \varepsilon_t \right) \left((\theta_1 \varepsilon_{t-1-k} + \theta_2 \varepsilon_{t-2-k} + \dots + \theta_q \varepsilon_{t-q-k}) + \varepsilon_{t-k} \right) \right] \quad (2.267)$$

2.267 no.lu denklemde, gecikme uzunluğu $k=1$ olmak üzere, $MA(q)$ sürecinin 1 dönem gecikmeli kovaryansı 2.268 no.lu denklemle ifade edilir.

$$\gamma_1 = E \left[\left((\theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}) + \varepsilon_t \right) \left((\theta_1 \varepsilon_{t-2} + \theta_2 \varepsilon_{t-3} + \dots + \theta_q \varepsilon_{t-q-1}) + \varepsilon_{t-1} \right) \right] \quad (2.268)$$

Hata terimlerinin farklı gecikmelerinin beklenen değerlerinin çarpaz çarpımları 0 değerine eşit olduğundan 2.268 no.lu denklem, aşağıda 2.269 ve 2.270 no.lu denklemlerde gösterildiği gibi düzenlenebilir.

$$\gamma_1 = \theta_1 E[\varepsilon_{t-1}^2] + \theta_2 \theta_1 E[\varepsilon_{t-2}^2] + \theta_3 \theta_2 E[\varepsilon_{t-3}^2] + \dots + \theta_q \theta_{q-1} E[\varepsilon_{t-q}^2] \quad (2.269)$$

$$\gamma_1 = (\theta_1 + \theta_2 \theta_1 + \theta_3 \theta_2 + \dots + \theta_q \theta_{q-1}) \sigma_\varepsilon^2 \quad (2.270)$$

Yukarıdaki yöntemle, gecikme uzunluğu $k=2$, $k=3$ ve $k=q$ olmak üzere, $MA(q)$ sürecinin 2, 3 ve q dönem gecikmeli kovaryansları da 2.171, 2.272, 2.273 ve 2.274 no.lu denklemlerde gösterildiği gibi elde edilebilir.

$$k=1 \quad : \quad \gamma_1 = (\theta_1 + \theta_2 \theta_1 + \theta_3 \theta_2 + \dots + \theta_q \theta_{q-1}) \sigma_\varepsilon^2 \quad (2.271)$$

$$k=2 \quad : \quad \gamma_2 = (\theta_2 + \theta_3 \theta_1 + \theta_4 \theta_2 + \dots + \theta_q \theta_{q-2}) \sigma_\varepsilon^2 \quad (2.272)$$

$$k=3 \quad : \quad \gamma_3 = (\theta_3 + \theta_4 \theta_1 + \theta_5 \theta_2 + \dots + \theta_q \theta_{q-3}) \sigma_\varepsilon^2 \quad (2.273)$$

.

.

.

$$k=q \quad \gamma_q = \theta_q \sigma_\varepsilon^2 \quad (2.274)$$

Yukarıdaki sonuçlar, MA sürecinin kovaryanslarının zamandan bağımsız bir şekilde sadece k gecikme uzunluğuna göre belirlendiğini göstermektedir.

MA sürecinde, gecikme uzunluğu, MA sürecinin derecesinden büyük ($k > q$) ise kovaryanslar 0 olacaktır. Aşağıda yer alan 2.275 no.lu denklem, 2.266 no.lu denklemin açılmış halini temsil etmektedir.

$$Y_k = E \left[\left((\theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}) + \varepsilon_t \right) \left((\theta_1 \varepsilon_{t-1-k} + \theta_2 \varepsilon_{t-2-k} + \dots + \theta_q \varepsilon_{t-q-k}) + \varepsilon_{t-k} \right) \right] \quad (2.275)$$

Örneğin $k=q+1$ için 2.275 no.lu denklem yeniden düzenlenerek 2.276 no.lu denklem elde edilir.

$$Y_{q+1} = E \left[\left((\theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}) + \varepsilon_t \right) \left((\theta_1 \varepsilon_{t-2-q} + \theta_2 \varepsilon_{t-3-q} + \dots + \theta_q \varepsilon_{t-2q-1}) + \varepsilon_{t-q} \right) \right] \quad (2.276)$$

$k=q+1$ iken parantez içindeki ifadelerin tüm çarpımları, hata terimlerinin farklı gecikmelerinin beklenen değerlerinin çapraz çarpımları olacağından, 2.277 no.lu denklemde gösterildiği üzere, sonuç 0 değerine eşit olacaktır.

$$Y_{q+1} = 0 \quad (2.277)$$

Dolayısıyla MA sürecinin kovaryansları q . gecikmeden sonra $AR(p)$ sürecinde olduğu gibi aşamalı bir şekilde 0 değerine yaklaşmak yerine, doğrudan 0 değerine eşit olacaktır.

Belirli bir k gecikme uzunluğunda $MA(1)$, $MA(2)$, $MA(3)$ gibi MA süreçlerinin kovaryans analizini basit bir şekilde yapabilmek için 2.278 no.lu denklemde gösterildiği gibi $MA(q)$ sürecinin kovaryanslarının daha sade olarak ifade edilmesi yararlı olacaktır.

$$Y_k = \begin{cases} (\theta_k + \theta_{k+1} \theta_1 + \theta_{k+2} \theta_2 + \dots + \theta_q \theta_{q-k}) \delta_\varepsilon^2 & \text{ve } k=1, 2, \dots, q \\ 0 & \text{ve } k > q \end{cases} \quad (2.278)$$

2.278 no.lu denklemin daha sade biçimi, toplam sembolü kullanılarak 2.279 no.lu denklemle ifade edilebilir.

$$\gamma_k = \begin{cases} \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} & ; \theta_0=1 \text{ ve } d=1, 2, \dots, q \\ 0 & ; d < k \text{ veya } q < k \end{cases} \quad (2.279)$$

2.279 no.lu denklem kullanılarak $k=1$, $k=2$ ve $k=3$ iken MA(1) sürecinin kovaryansları 2.280, 2.281 ve 2.282 no.lu denklemlerle kolaylıkla ifade edilebilir.

$$\text{MA}(1); \quad q=k=1 \quad : \gamma_1 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 \theta_1 \quad (2.280)$$

$$\text{MA}(1); \quad q < k=2 \quad : \gamma_2 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = 0 \quad (2.281)$$

$$\text{MA}(1); \quad q < k=3 \quad : \gamma_3 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = 0 \quad (2.282)$$

Yukarıdaki denklem sistemine göre MA(1) sürecinin birinci gecikmeden sonraki kovaryanslarının tümü 0 değerine eşittir.

Benzer şekilde 2.279 no.lu denklem kullanılarak $k=1$, $k=2$ ve $k=3$ iken MA(2) sürecinin kovaryansları 2.283, 2.284 ve 2.285 no.lu denklemlerle ifade edilebilir.

$$\text{MA}(2); \quad q > k=1 \quad : \gamma_1 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 (\theta_1 + \theta_2 \theta_1) \quad (2.283)$$

$$\text{MA}(2); \quad q = k=2 \quad : \gamma_2 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 \theta_2 \quad (2.284)$$

$$\text{MA}(2); \quad q < k=3 \quad : \gamma_3 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = 0 \quad (2.285)$$

Yukarıdaki denklem sistemine göre MA(2) sürecinin ikinci gecikmeden sonraki kovaryanslarının tümü 0 değerine eşittir.

Son olarak, 2.279 no.lu denklem kullanılarak $k=1$, $k=2$ ve $k=3$ iken MA(3) sürecinin kovaryansları 2.286, 2.287 ve 2.288 no.lu denklemlerle ifade edilebilir.

$$\text{MA}(3); \quad q > k=1 \quad : \quad \gamma_1 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 (\theta_1 + \theta_2 \theta_1 + \theta_3 \theta_2) \quad (2.286)$$

$$\text{MA}(3); \quad q > k=2 \quad : \quad \gamma_2 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 (\theta_2 + \theta_3 \theta_1) \quad (2.287)$$

$$\text{MA}(3); \quad q = k=3 \quad : \quad \gamma_3 = \sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k} = \sigma_\varepsilon^2 \theta_3 \quad (2.288)$$

Yukarıdaki denklem sistemine göre de MA(3) sürecinin üçüncü gecikmeden sonraki kovaryanslarının tümü 0 değerine eşit olacaktır.

Sonuç olarak, MA sürecinin kovaryansları zamandan bağımsız bir şekilde sadece k gecikme uzunluğuna göre belirlendiği için durağanlık koşulunu sağlar. Ayrıca $k > q$ iken, diğer bir ifadeyle MA sürecinin derecesini temsil eden q . gecikmeden sonra kovaryanslar daima 0 değerine eşittir.

2.5.2.2.4. MA(q) Modelinin Otokorelasyon Parametreleri

2.289 no.lu denklemde k gecikme uzunluğu olmak üzere, ρ_k k . gecikmeye sahip otokorelasyon parametresini ifade eder.

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (2.289)$$

Yukarıda elde edilen, MA(q) modelinin varyansını ve kovaryanslarını temsil eden 2.261 ve 2.279 no.lu denklemler, 2.289 no.lu denklemde yerine yazılarak 2.290 no.lu denkleme ulaşılır.

$$\rho_k = \frac{\sigma_\varepsilon^2 \sum_{d=1}^q \theta_d \theta_{d-k}}{\sigma_\varepsilon^2 \left(1 + \left(\sum_{i=1}^q \theta_i^2 \right) \right)} \quad (2.290)$$

Pay ve paydada çarpan konumundaki hata terimlerinin varyansları sadeleştirilerek 2.291 no.lu denkleme ulaşılır.

$$\rho_k = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} \quad (2.291)$$

MA(q) sürecinin q. gecikmeden sonraki kovaryansları 0 değerine eşit olduğundan, MA(q) sürecinin q. gecikmeden sonraki otokorelasyonları da 0 olacaktır. Örneğin 2.292 ve 2.293 no.lu denklemlerde gösterildiği üzere, k=q+1 iken bir MA(q) sürecinin (q+1)'inci otokorelasyon katsayısı 0 olur.

$$\rho_{q+1} = \frac{0}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} \quad (2.292)$$

$$\rho_{q+1} = 0 \quad (2.293)$$

MA(1), MA(2), MA(3) gibi MA süreçlerinin 0 değerinden farklı otokorelasyon parametreleri, 2.291 no.lu denklem yardımıyla kolaylıkla elde edilebilir. MA(1), MA(2) ve MA(3) süreçlerinin farklı gecikme uzunluklarına göre 0 değerinden farklı otokorelasyon katsayıları 2.294 ile 2.299 arasındaki denklemlerle gösterilmiştir.

$$\text{MA(1); } q=k=1 : \rho_1 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_1}{1 + \theta_1^2} \quad (2.294)$$

$$\text{MA(2); } q < k=1 : \rho_1 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_1 + \theta_2 \theta_1}{1 + \theta_1^2 + \theta_2^2} \quad (2.295)$$

$$\text{MA(2); } q=k=2 : \rho_2 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2} \quad (2.296)$$

$$\text{MA(3); } q < k=1 : \rho_1 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_1 + \theta_2 \theta_1 + \theta_3 \theta_2}{1 + \theta_1^2 + \theta_2^2 + \theta_3^2} \quad (2.297)$$

$$MA(3); \quad q < k = 2 \quad : \rho_2 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_2 + \theta_3 \theta_1}{1 + \theta_1^2 + \theta_2^2 + \theta_3^2} \quad (2.298)$$

$$MA(3); \quad q = k = 3 \quad : \rho_3 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_3}{1 + \theta_1^2 + \theta_2^2 + \theta_3^2} \quad (2.299)$$

Benzer şekilde MA(q) sürecinin farklı gecikme uzunluklarına göre 0 değerinden farklı otokorelasyon parametreleri 2.300 ile 2.303 arasındaki denklemlerle gösterilmiştir.

$$MA(q); \quad q < k = 1 \quad : \rho_1 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_1 + \theta_2 \theta_1 + \theta_3 \theta_2 + \dots + \theta_q \theta_{q-1}}{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2} \quad (2.300)$$

$$MA(q); \quad q < k = 2 \quad : \rho_2 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_2 + \theta_3 \theta_1 + \theta_4 \theta_2 + \dots + \theta_q \theta_{q-2}}{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2} \quad (2.301)$$

$$MA(q); \quad q < k = 3 \quad : \rho_3 = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_3 + \theta_4 \theta_1 + \theta_5 \theta_2 + \dots + \theta_q \theta_{q-3}}{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2} \quad (2.302)$$

·
·
·

$$MA(q); \quad q = k \quad : \rho_q = \frac{\sum_{d=1}^q \theta_d \theta_{d-k}}{\left(1 + \left(\sum_{i=1}^q \theta_i^2\right)\right)} = \frac{\theta_q}{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2} \quad (2.303)$$

2.5.3. Otoregresif Hareketli Ortalama Modeli (ARMA)

AR(p) ve MA(q) modellerinin birleşimiyle ARMA(p,q) modeli elde edilir. Bu sebeple ARMA (p,q) modeli, serinin durağan olması koşuluyla, bir değişkenin t dönemindeki değerinin, t-1, t-2 gibi kendi geçmiş dönem değerleri ile mevcut ve geçmiş dönem problemsiz hata terimlerinin doğrusal bir fonksiyonu olduğunu temsil eder (Brooks, 2008, s.223; Makridakis ve diğerleri, 1998, s.344). 2.304 no.lu denklem ARMA (p,q) modelini temsil etmektedir.

$$y_t = \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \quad (2.304)$$

AR(p) ve MA(q) modellerinde olduğu gibi ARMA (p,q) modeli de bir sabit terim içerebilir (Montgomery ve diğerleri, 2008, s.253). 2.305 no.lu denklem ise sabit terim içeren ARMA (p,q) modelini temsil etmektedir.

$$y_t = \alpha_0 + \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \varepsilon_t \quad (2.305)$$

2.306 ve 2.307 no.lu denklemlerde gösterildiği gibi p=1 ve q=0 iken ARMA(1,0) modeli AR(1) modeline, p=0 ve q=1 iken ARMA(0,1) modeli MA(1) modeline dönüşür (Enders, 2004, s.51).

$$\text{AR}(1): y_t = \alpha_0 + \phi_1 y_{t-1} + \varepsilon_t \quad (2.306)$$

$$\text{MA}(1): y_t = \alpha_0 + \theta_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.307)$$

ARMA(1,1) modeli ise 2.308 no.lu denklemle ifade edilebilir.

$$y_t = \alpha_0 + \phi_1 y_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t \quad (2.308)$$

ARMA(1,1) modelinin durağanlık koşulları AR(1) modelinin durağanlık koşullarıyla aynıdır (Mills, 2019, s.44).

ARMA(p,q) sürecinde p ve q dereceleri (ya da gecikmeleri), sırasıyla, otokorelasyon ve kısmi otokorelasyon paternleri tarafından belirlenebilir. Modelin AR bölümünün derecesini belirlemek için “Otoregresif Model” konusunda açıklandığı gibi PACF, MA bölümünün derecesini belirlemek için “Hareketli Ortalama Modeli” konusunda açıklandığı gibi ACF korelogramları incelenebilir. p ve q dereceleri 0’dan büyük olmak üzere, ARMA(p,q) sürecinde, ACF ve PACF paternleri aşamalı bir şekilde 0 değerine doğru yaklaşır. Uygulamada p ve q değerleri nadiren 2 değerini aşar (Hanke ve Wichern, 2014, s.363; Brooks, 2008, s.224-225).

AR ve MA modellerinin derecelerinin belirlenmesinde, en küçük kareler (ordinary least squares) ve maksimum olabilirlik (maximum likelihood) optimizasyon yöntemlerini kullanan Box-Jenkins ve Hannan-Rissanen prosedürü, Kalman filtresi, Akaike ve Bayes/Schwarz bilgi kriterleri gibi hata kareleri toplamının (residual sum of squares) minimizasyonuna dayalı çok sayıda yineleyici yöntem vardır (Chatfield, 1996, s.59; Mills, 2019, s.46-48). Günümüzde söz konusu yöntemler birçok paket programı ve/veya yazılım kütüphanesi tarafından araştırmacıya sunulmaktadır.

2.5.4. Bütünleşik Otoregresif Hareketli Ortalama Modeli (ARIMA)

Bütünleşik otoregresif hareketli ortalama modeli, yaygın bilinen adıyla ARIMA modeli, durağan ve durağan olmayan zaman serilerini temsil edebilen doğrusal model sınıfıdır. Serinin kendi gecikmeleri dışında bağımsız değişken içermeyen ARIMA, ağırlıklı olarak verideki otokorelasyon paternlerine dayanır. ARIMA literatürde Box-Jenkins yöntemi olarak da bilinir (Hanke ve Wichern, 2014, s.355).

Bütünleşme derecesi d olan $AR(p)$ ve $MA(q)$ modellerinin birleşimiyle $ARIMA(p,d,q)$ modeli elde edilir. $ARIMA(p,0,0)$ durağan bir $AR(p)$ sürecini, $ARIMA(0,0,q)$ $MA(q)$ sürecini, $ARIMA(p,0,q)$ ise durağan bir $ARMA(p,q)$ sürecini ifade eder. Eğer seri sıfırıncı dereceden bütükleşik değilse, serinin durağan olana kadar farkları alınarak durağanlık koşulları sağlanır. Birinci farkları durağan bir süreç $ARIMA(p,1,q)$, ikinci farkları durağan bir süreç ise $ARIMA(p,2,q)$ olarak ifade edilir (Gujarati ve Porter, 2009, s.776-777).

2.5.5. Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama Modeli (SARIMA)

Aylık, haftalık gibi güçlü periyodik paternler gösteren, öngörülebilir mevsimsel davranışlara sahip zaman serilerinin ARIMA modeline dahil edilmesiyle mevsimsel ARIMA (seasonal ARIMA, SARIMA) elde edilir. s mevsimsel gecikme (bağımlılık), D mevsimsel bütünleşme derecesi, P mevsimsel AR ve Q mevsimsel MA süreçlerinin derecesi olmak üzere, genel

gösterimi $SARIMA(p,d,q) \times (P,D,Q)_s$ olarak ifade edilebilir (Montgomery ve diğerleri, 2008, s.282-283).

SARIMA, serinin kendi gecikmelerine dayanan otokorelasyonlarından türetilen AR ve MA süreçleri ile periyodik gecikmelerine dayanan otokorelasyonlarının hesaba katıldığı mevsimsel AR (seasonal AR, SAR) ve MA (seasonal MA, SMA) süreçlerinden oluşur. ACF ve PACF korelogramlarında s, 2s, 3s gibi paternleri takip eden ve mevsimsel gecikmeyi ifade eden s değeri, zaman serisinin karakteristiğine göre değişebilir (Enders, 2004, s.112). Çeyreklik zaman serilerinde s değerinin 4, aylık zaman serilerinde ise 12 olması beklenebilir (Hanke ve Wichern, 2014, s.388).

$$y_t = \phi_1 y_{t-1} + \phi_{12} y_{t-12} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.309)$$

2.309 no.lu denklem AR(1) ve MA(1) süreçlerine ilave olarak, mevsimsel bağımlılığı 12 dönem olan SAR(1) ve SMA(1) süreçlerinden oluşmaktadır. Dolayısıyla 2.309 no.lu denklem $SARIMA(1,0,1) \times (1,0,1)_{12}$ modelini temsil eder.

$$y_t = \phi_1 y_{t-1} + \phi_{12} y_{t-12} + \phi_{24} y_{t-24} + \varepsilon_t \quad (2.310)$$

2.310 no.lu denklem ise AR(1) ve SAR(2) süreçlerinden oluşan $SARIMA(1,0,0) \times (2,0,0)_{12}$ modelini temsil eder (Hanke ve Wichern, 2014, s.396). $SARIMA(p,d,q) \times (P,D,Q)_s$ modelinin genel formu ise 2.311 no.lu denklemle ifade edilebilir.

$$y_t = \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) + \left(\sum_{i=1}^P \phi_{i,s} y_{t-i,s} \right) + \left(\sum_{i=1}^Q \theta_{i,s} \varepsilon_{t-i,s} \right) + \varepsilon_t \quad (2.311)$$

Durağan olmayan mevsimsel serilerin, modeli en iyi şekilde belirleyebilmesi için (normal fark alma işlemine) ilave olarak bir mevsimsel farkın alınması gerekebilir (Hanke ve Wichern, 2014, s.388). Söz konusu durum 2.312, 2.313 ve 2.314 no.lu denklemlerle gösterilmektedir.

$$\text{Normal Fark (d)} : \Delta y_t = y_t - y_{t-1} \quad (2.312)$$

$$\text{Mevsimsel Fark (D)} : \Delta_{12}y_t = y_t - y_{t-12} \quad (2.313)$$

$$\text{Birleşik Fark} : \Delta\Delta_{12}y_t = \Delta(\Delta_{12}y_t) \quad (2.314)$$

Birleşik fark, seride normal fark alma işlemine ilave olarak mevsimsel fark alma işlemini ifade etmektedir. Birleşik farkı y_t ve y_t 'nin gecikmeleri türünden elde etmek için birleşik fark işlemi içinde, 2.315 no.lu denklemde gösterildiği üzere, $\Delta_{12}y_t$ yerine $y_t - y_{t-12}$ yazılır.

$$\Delta\Delta_{12}y_t = \Delta(y_t - y_{t-12}) \quad (2.315)$$

$y_t - y_{t-12}$ ifadesinin solunda yer alan Δ fark operatörü, mevsimsel farkı temsil eden $y_t - y_{t-12}$ ifadesiyle onun bir dönem gecikmesi $y_{t-1} - y_{t-13}$ ifadesinin farkının alınması gerektiğine işaret eder. Söz konusu durum, 2.316 no.lu denklemde gösterilmektedir.

$$\Delta\Delta_{12}y_t = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \quad (2.316)$$

Böylece birleşik fark işlemi y_t ve y_t 'nin gecikmeleri türünden 2.317 no.lu denklemde gösterildiği şekilde elde edilir (Hanke ve Wichern, 2014, s.394).

$$\Delta\Delta_{12}y_t = y_t - y_{t-12} - y_{t-1} + y_{t-13} \quad (2.317)$$

w_t ifadesi birleşik farkı temsil etmek üzere, 2.317 no.lu eşitlik, 2.318 no.lu denklemle de ifade edilebilir.

$$w_t = y_t - y_{t-12} - y_{t-1} + y_{t-13} \quad (2.318)$$

SARIMA(1,1,1)x(1,1,1)₁₂ modeli, normal ve mevsimsel farklar içerdiğinden, w_t ifadesi yardımıyla 2.319 no.lu denklemde gösterildiği gibi yazılabilir.

$$w_t = \phi_1 w_{t-1} + \phi_{12} w_{t-12} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.319)$$

Modelde yer alan w_{t-1} ve w_{t-12} ifadelerinin y_t türünden açılımı ise 2.320 ve 2.321 no.lu denklemlerde gösterilmektedir.

$$w_t = \phi_1(y_{t-1} - y_{t-13} - y_{t-2} + y_{t-14}) + \phi_{12}(y_{t-12} - y_{t-24} - y_{t-13} + y_{t-25}) + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.320)$$

$$w_t = \phi_1 y_{t-1} - \phi_1 y_{t-2} + \phi_1 y_{t-14} - (\phi_1 + \phi_{12}) y_{t-13} + \phi_{12} y_{t-12} - \phi_{12} y_{t-24} + \phi_{12} y_{t-25} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.321)$$

Son olarak w_t yerine de w_t ifadesinin eşiti $y_t - y_{t-12} - y_{t-1} + y_{t-13}$ yazılır ve model en genel biçimiyle, 2.322 no.lu denklem yardımıyla, y_t türünden ifade edilebilir.

$$y_t = (1 + \phi_1) y_{t-1} - \phi_1 y_{t-2} + \phi_1 y_{t-14} - (1 + \phi_1 + \phi_{12}) y_{t-13} + (1 + \phi_{12}) y_{t-12} - \phi_{12} y_{t-24} + \phi_{12} y_{t-25} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.322)$$

SARIMA(p,d,q)x(P,D,Q)_s modelinde d ve D değerleri genellikle 1 değerini aşmaz (Chatfield, 1996, s.60). Bununla birlikte Bell ve Hillmer (1984)'in araştırmasında gösterdiği gibi (durağan bir seriyi temsil eden) ARMA ve (durağan bir mevsimsel seriyi ifade eden) mevsimsel ARMA parametreleri birlikte belirlenir ve tahmin edilir. Böyle bir durumda mevsimsellikten arındırılmış veri (seasonally adjusted data) kullanılmasından kaçınılmalıdır (Enders, 2004, s.94).

“Mevsimsellik çarpanı” (multiplicative seasonality) ya da “çoklu mevsimsellik” (multiple seasonalities), mevsimsel etkilerin özel bir türünü işaret eder. Aylık veriler üzerinde çalışıldığı varsayımında, mevsimsellik çarpanı, sadece birbirini takip eden yılların aynı aylarının değil, aynı yılın birbirini takip eden aylarının da mevsimsellik üzerinde etkisi (birbiriyle ilişkili) olabileceği fikrine dayanır (Box ve diğerleri, 2016, s.308). Mevsimsel etkilerin ve ARIMA modelinin etkileşimine olanak tanıyan mevsimsellik çarpanı, SARIMA(1,0,1)x(1,0,1)₁₂ modeli üzerinden karşılaştırmalı olarak 2.323 ve 2.324 no.lu denklemlerde gösterilmektedir.

$$y_t = \phi_1 y_{t-1} + \phi_{12} y_{t-12} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \varepsilon_t \quad (2.323)$$

$$y_t = \phi_1 y_{t-1} + \phi_{12} y_{t-12} + \phi_1 \phi_{12} y_{t-13} + \theta_1 \varepsilon_{t-1} + \theta_{12} \varepsilon_{t-12} + \theta_1 \theta_{12} \varepsilon_{t-13} + \varepsilon_t \quad (2.324)$$

2.323 no.lu modelin AR(1) ve MA(1) süreçlerinden hareketle, SAR ve SMA süreçlerine, sırasıyla, y_{t-13} ve ε_{t-13} şeklinde mevsimsel bağımlılığın birer dönem gecikmeleri ilave edilerek çoklu mevsimsellik faktörü modele dahil edilmiş ve 2.324 no.lu model elde edilmiştir. 2.324 no.lu modelde yer alan y_{t-13} mevsim faktörüne göre AR(1) sürecini, ε_{t-13} ise mevsim faktörüne göre MA(1) sürecini temsil ettiğinden, y_{t-13} ve ε_{t-13} ifadeleri, sırasıyla, ϕ_{12} ve θ_{12} mevsim çarpanlarına ilave olarak ϕ_1 ve θ_1 parametrelerine sahiptirler. ϕ_1 ve θ_1 parametreleri ise sırasıyla, AR(1) ve MA(1) süreçlerinin parametreleridir. Diğer bir ifadeyle 2.324 no.lu modelde tahmin edilen parametreler ϕ_1 , ϕ_{12} , θ_1 , θ_{12} olup 2.323 no.lu modelde tahmin edilen parametre sayısına eşittir (Enders, 2004, s.95). Söz konusu durum 2.325 no.lu denklemde gösterilmektedir.

$$y_t = \phi_1 y_{t-1} + \phi_{12} (y_{t-12} + \phi_1 y_{t-13}) + \theta_1 \varepsilon_{t-1} + \theta_{12} (\varepsilon_{t-12} + \theta_1 \varepsilon_{t-13}) + \varepsilon_t \quad (2.325)$$

Yukarıdaki açıklamalardan hareketle, SARIMA(p,d,q)x(P,D,Q)_s modeline dahil edilen çoklu mevsimsellik faktörünün matematiksel formu, Evans (2003, s.249)'ın önerdiği formun genelleştirilmesiyle 2.326 no.lu denklemde gösterildiği gibi ifade edilebilir.

$$\begin{aligned} y_t = & \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) \\ & + \left(\sum_{i=1}^P \phi_{i,s} \left(y_{t-i,s} + \sum_{i=1}^p \phi_i y_{t-i-s} \right) \right) \\ & + \left(\sum_{i=1}^Q \theta_{i,s} \left(\varepsilon_{t-i,s} + \sum_{i=1}^q \theta_i \varepsilon_{t-i-s} \right) \right) + \varepsilon_t \end{aligned} \quad (2.326)$$

2.322 no.lu eşitliği ifade eden SARIMA(1,1,1)x(1,1,1)₁₂ modeline mevsimsellik çarpanını eklemek için M notasyonu 2.325 no.lu eşitlikte yer alan

$\phi_{12}\phi_1y_{t-13}+\theta_{12}\theta_1\varepsilon_{t-13}$ ifadesini temsil etmek üzere, M'nin birleşik (normal ve mevsimsel) farkının alınarak, sonucun 2.322 no.lu eşitliğe ilave edilmesi yeterli olacaktır. 2.327, 2.328 ve 2.329 no.lu denklemlerde M ifadesinin birleşik farkının alınması aşamaları gösterilmektedir.

$$M=\phi_{12}\phi_1y_{t-13}+\theta_{12}\theta_1\varepsilon_{t-13} \quad (2.327)$$

$$\Delta\Delta_{12}M=\phi_{12}\phi_1(y_{t-13}-y_{t-25}-y_{t-14}+y_{t-26})+\theta_{12}\theta_1\varepsilon_{t-13} \quad (2.328)$$

$$\Delta\Delta_{12}M=\phi_{12}\phi_1y_{t-13}-\phi_{12}\phi_1y_{t-25}-\phi_{12}\phi_1y_{t-14}+\phi_{12}\phi_1y_{t-26}+\theta_{12}\theta_1\varepsilon_{t-13} \quad (2.329)$$

2.329 no.lu sonuç 2.322 no.lu eşitliğe ilave edilerek mevsimsellik çarpanının dahil edildiği SARIMA(1,1,1)x(1,1,1)₁₂ modelinin en genel formu olan 2.330 no.lu denklem elde edilir (Hyndman, 2014, s.84).

$$\begin{aligned} y_t &= (1+\phi_1)y_{t-1}-\phi_1y_{t-2}+(\phi_1-\phi_{12}\phi_1)y_{t-14}-(1+\phi_1+\phi_{12}-\phi_{12}\phi_1)y_{t-13} \\ &+ (1+\phi_{12})y_{t-12}-\phi_{12}y_{t-24}+(\phi_{12}-\phi_{12}\phi_1)y_{t-25}+\phi_{12}\phi_1y_{t-26} \\ &+ \theta_1\varepsilon_{t-1}+\theta_{12}\varepsilon_{t-12}+\theta_{12}\theta_1\varepsilon_{t-13}+\varepsilon_t \end{aligned} \quad (2.330)$$

2.5.6. Dışsal Değişkenli Mevsimsel Bütünleşik Oto regresif Hareketli Ortalama Modeli (SARIMAX)

Dışsal değişken olarak çoklu zaman serilerini kullanan SARIMAX (Seasonal Integrated Autoregressive Moving Average Models with Exogenous Variables), bağımlı değişkeni tahmin etmek için çok değişkenli zaman serilerinden yararlanan ARIMA modelinin gelişmiş bir türü olarak tanımlanabilir (Alharbi ve Csala, 2022, s.2). r modele dahil edilen maksimum dışsal değişken sayısı ve α ise modele eklenen dışsal değişkenlerin (x'lerin) parametresi olmak üzere, çoklu mevsimsellik faktörü içermeyen SARIMAX modelinin genel formu, 2.331 no.lu denklemde gösterildiği gibi modele dışsal değişken faktörünün dahil edilmesiyle elde edilir.

$$\begin{aligned}
y_t = & \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) \\
& + \left(\sum_{i=1}^P \phi_{i,s} y_{t-i,s} \right) + \left(\sum_{i=1}^Q \theta_{i,s} \varepsilon_{t-i,s} \right) \\
& + \left(\sum_{i=1}^r \alpha_i x_i \right) + \varepsilon_t
\end{aligned} \tag{2.331}$$

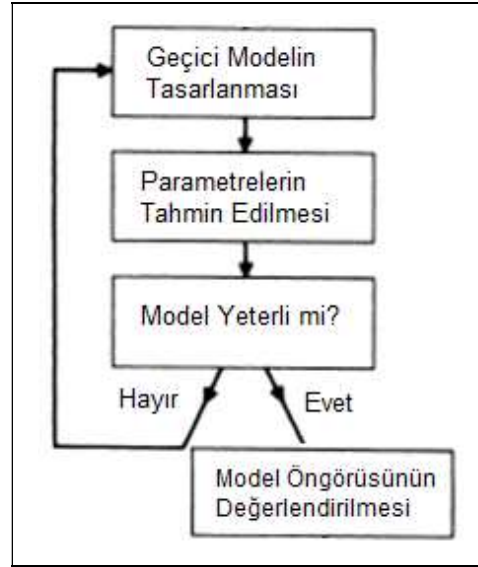
Benzer şekilde çoklu mevsimsellik faktörü içeren SARIMAX modelinin genel formu ise 2.332 no.lu denklemde gösterildiği gibi olacaktır.

$$\begin{aligned}
y_t = & \left(\sum_{i=1}^p \phi_i y_{t-i} \right) + \left(\sum_{i=1}^q \theta_i \varepsilon_{t-i} \right) \\
& + \left(\sum_{i=1}^P \phi_{i,s} \left(y_{t-i,s} + \sum_{j=1}^p \phi_j y_{t-i-s} \right) \right) \\
& + \left(\sum_{i=1}^Q \theta_{i,s} \left(\varepsilon_{t-i,s} + \sum_{j=1}^q \theta_j \varepsilon_{t-i-s} \right) \right) \\
& + \left(\sum_{i=1}^r \alpha_i x_i \right) + \varepsilon_t
\end{aligned} \tag{2.332}$$

2.6. Box-Jenkins Yöntemi

Box-Jenkins yöntemi, zaman serilerinden oluşan ARMA tipi modellerin belirlenmesi (identification), parametrelerinin tahmin edilmesi (fitting), test edilmesi (controlling) ve tahmin edilen parametrelerden öngörü yapılması (forecasting) için bir dizi prosedürü işaret eder (Hanke ve Wichern, 2014, s.356). ARMA(p, q) modellerinin yanı sıra ARIMA(p, d, q), SARIMA(p, d, q)(P, D, Q) ve SARIMAX(p, d, q)(P, D, Q) gibi ARMA tipi modellerin inşası için Box-

Jenkins yöntemine yaygın olarak başvurulur. Box-Jenkins yönteminin aşamaları Şekil 2.1'de gösterilmiştir.



Şekil 2.1. Box-Jenkins Yöntemi

Kaynak: Box ve diğerleri, 2016, s.16

Temel olarak dört aşamadan oluşan Box-Jenkins yöntemi, takip eden alt başlıklarda detaylı olarak açıklanmıştır.

2.6.1. Geçici Modelin Tasarlanması

İlk olarak serinin durağan olup olmadığı araştırılır. Mevsimsellik içermeyen ve durağan olmayan serilerde, durağanlık sağlanana kadar fark alma işlemi yapılır. Mevsimsellik içeren ve durağan olmayan serilerin mevsimsellikten arındırılması amacıyla, mevsimsel farkları da alınır (Makridakis ve diğerleri, 1998, s.347). Durağan bir serinin fazladan farklarının alınması (overdifferencing), ilave otokorelasyonlara yol açarak modelin karmaşıklığını artırır. Örneğin modelin yeterliliğini sağlayan en basit formu $ARIMA(0,1,0)$ olması gerekirken, fazladan farkları alınan bir seride $ARIMA(0,2,1)$ şeklinde ilave (sahte) bir $MA(1)$ süreci ortaya çıkabilir (Box ve diğerleri, 2016, s.183). Durağanlık için serinin logaritmik ya da üssel dönüşümü de yapılabilir (Makridakis ve diğerleri, 1998, s.358).

Modelin yapısı serinin durağan formu üzerinden belirlenmelidir. ACF korelogramında otokorelasyonların 0 değerine aşamalı bir şekilde yaklaşması modelde AR sürecinin olabileceğine, PACF korelogramında kısmi

otokorelasyonların 0 değerine aşamalı bir şekilde yaklaşması ise modelde MA sürecinin olabileceğine işaret eder. ACF ve PACF korelogramlarında, sırasıyla, otokorelasyonların ve kısmi otokorelasyonların her ikisinin de 0 değerine aşamalı bir şekilde yaklaşması ise modelde AR ve MA süreçlerinin birlikte yer alabileceğini ifade eder (Box ve diğerleri, 2016, s.183). Mevsimsel gecikme uzunluğuna karşılık gelen ACF ve PACF değerleri anlamlı bir şekilde 0'dan farklı ise sırasıyla AR ve MA tipi mevsimsel etkilerin varlığına işaret edebilir (Makridakis ve diğerleri, 1998, s.348).

ARIMA(p,d,q) modelinin, olası AR ve MA süreçlerine göre genel yapısı oluşturulduktan sonra, p ve q derecelerinin belirlenmesi amacıyla, modelin MA kısmı için ACF, AR kısmı için PACF değerlerinin aniden 0 değerine yaklaşmasını ifade eden kesilmeler araştırılabilir. PACF korelogramının p. gecikmeden sonra kesilmesi modelin AR sürecine ilişkin p derecesini, ACF korelogramının q. gecikmeden sonra kesilmesi ise modelin MA sürecine ilişkin q derecesini tespit etmede kullanılabilir (Box ve diğerleri, 2016, s.181-183). Modelde AR ve MA süreçlerinin bir arada yer alması durumunda, modelin p ve q derecelerini belirlemek daha zor olacaktır. Model seçiminde hata kareleri toplamının (residual sum of squares) minimizasyonu ya da olasılığın en yüksek olduğu değer kullanılabilir. Ancak bu yöntem, modeldeki parametre sayısı arttıkça hata kareleri toplamı azalacağından ya da olasılık değeri artacağından her zaman güvenilir sonuçlar vermeyebilir. Bunun yerine modele ilave parametre eklenmesinin cezalandırıldığı düzeltilmiş R^2 , Akaike ve Bayes/Schwarz gibi bilgi kriterleri kullanılabilir (Makridakis ve diğerleri, 1998, s.361-362).

Genel olarak model seçimi, daha az parametre içeren basit bir modelin daha fazla parametre içeren karmaşık bir modele tercih edilmesini ifade eden "tutumluluk ilkesi"ne (principle of parsimony) göre yapılır. Sınırlı bir veriyle çalışırken, "aşırı öğrenme"ye (overfitting) işaret eden, çok sayıda parametre kullanılarak oluşturulan bir model veriyi çok iyi bir şekilde temsil edebilmesine (ezberlemesine) rağmen, öngörü başarısı oldukça zayıf kalacaktır. En iyi öngörü sonuçları için amaç, verinin temel özelliklerini temsil edebilen en basit modelin geliştirilmesi olmalıdır (Hanke ve Wichern, 2014, s.363-364).

2.6.2. Parametrelerin Tahmin Edilmesi

Model seçimi yapıldıktan sonra, hata kareleri toplamının minimizasyonu ile modelin parametreleri tahmin edilir. Hata kareleri toplamının minimizasyonunda doğrusal olmayan, yinelemeli (iterative) ya da doğrusal yöntemler kullanılabilir. Tahmin edilen model parametrelerinin olasılık değerlerini ifade eden p değerleri, t ya da Z gibi istatistikleri değerlendirilerek anlamlı bulunmayan parametreler modelden çıkartılır (Makridakis ve diğerleri, 1998, s.359-361).

Hata terimlerinin ortalama hata kareleri (residual mean square) ve varyansı hesaplanır. Ortalama hata kareleri modellerin karşılaştırılmasında ve değerlendirilmesinde oldukça kullanışlıdır. Ayrıca ortalama hata kareleri öngörünün hata sınırlarını hesaplamakta da kullanılır (Hanke ve Wichern, 2014, s.365-366).

2.6.3. Model Yeterliliğinin Araştırılması

Model yeterliliğinin araştırılması, modelin öngörü amacıyla kullanılmadan önce yeterliliğinin test edilmesini ifade eder. Modelin yeterliliği, tahmin edilen modelin hata terimlerinin öngörünün geliştirilmesinde kullanılamamasına, diğer bir ifadeyle tamamen rassal dağıldığına işaret eder.

Tahmin edilen modelin hata terimlerinin normal dağılıma sahip olup olmadığının araştırılması için histogram ya da normal olasılık dağılımı grafikleri kullanılabilir. Benzer şekilde hata terimlerine ilişkin grafik, hata terimlerinin dağılımında anomali olup olmadığının tespit edilmesinde kullanışlı olabilir (Hanke ve Wichern, 2014, s.366).

İyi bir öngörü performansına sahip bir modelin, hata terimleri arasındaki otokorelasyonun 0 değerine yakın, daha spesifik ifade edilirse, T gözlem sayısı olmak üzere, 0 etrafında maksimum $\pm 2/\sqrt{T}$ kadar salınımlarda (ya da güven aralıklarının sınırları dahilinde) bulunması beklenir. Hata terimlerinin normal ve/veya mevsimsel gecikmelerdeki otokorelasyonunun anlamlı olması, modelin yetersiz olduğuna işaret eder. Hata terimlerinin rassal

dağılmadığına işaret eden böyle bir durumda modelin üzerinde değişiklikler yapılması ya da yeni bir model seçilmesi gerekir.

Hata terimleri arasındaki otokorelasyonun 0 değerine eşit olup olmadığını test etmek için Ljung ve Box tarafından geliştirilen Q istatistiği kullanılabilir (Makridakis ve diğerleri, 1998, s.365).

Model inşası aşamasında kişisel kanılar (judgment) önemli bir rol oynar. Aynı yeterliliğe sahip iki model arasında, öngörünün doğasına dayalı bir seçim yapılabilir. Benzer şekilde sıra dışı koşullarla açıklanabilen büyük değerlere sahip birkaç hata terimi görmezden gelinebilir (Hanke ve Wichern, 2014, s.367).

2.6.4. Model Öngörüsünün Değerlendirilmesi

Yeterli bir model elde edildikten sonra, öngörünün güven aralığı belirlenerek geleceğe ilişkin bir ya da birkaç dönemlik öngörü yapılabilir. Belirli bir güven aralığı seviyesinde, daha uzak bir öngörü periyodu daha geniş bir güven aralığına işaret eder. Çünkü öngörü periyodunun artması, daha uzak bir geleceğe ilişkin tahminlerin yapılmasını gerektirdiği için belirsizliğin artması anlamına gelir.

Gelecek dönemlere ilişkin veriler ulaşılabilir olduğunda, aynı model, farklı bir dönemden itibaren revize edilmiş öngörülerin yapılmasında kullanılabilir.

Serinin paternleri zamanla değişirse, yeni veriler modelin parametrelerini tekrar tahmin etmede kullanılabilir ya da gerekli olması durumunda yeni bir model geliştirilebilir. Bu çerçevede öngörü hatalarının takip edilmesi yararlı olabilir. Son yapılan öngörülerin hataları, sürekli bir şekilde, bir önceki yapılan öngörülerin hatalarından daha büyük olma eğilimindeyse, modelin yeniden değerlendirilmesi gerektiğine işaret eder. Benzer şekilde son tahmin hataları, sürekli bir şekilde, pozitif (underpredicting) ya da negatif (overpredicting) olma eğilimindeyse aynı durum geçerlidir (Hanke ve Wichern, 2014, s.367).

ÜÇÜNCÜ BÖLÜM

TÜRKİYE GENELİ VE ESKİŞEHİR ŞUBESİ REESKONT BÖLGESİ: YAPAY SİNİR AĞLARI VE ZAMAN SERİLERİ ANALİZİ İLE BANKNOT TALEBİNİN TAHMİNİNE İLİŞKİN UYGULAMA

Bu bölümde, yapay zeka teknolojilerinin ve zaman serileri modellerinin, Türkiye geneli ve Eskişehir Şubesi Reeskont Bölgesinin banknot talebiyle ilgili tahmin performansları incelenmiştir. Yapay zeka teknolojileriyle banknot talebinin tahmin edilmesinde, özellikle zaman serilerine ilişkin örüntülerin çıkartılmasında yaygın bir şekilde kullanılan FNN, TLNN ve LSTM tipi çok katmanlı üç farklı yapay sinir ağı eğitilmiştir. Zaman serileri analiziyle banknot talebinin tahmin edilmesinde ise SARIMA modeli kullanılmıştır. Hem yapay sinir ağları hem de zaman serileri analizi alanında veri ön işleme, veri analizi, istatistiksel testler, veri görselleştirme, zaman serileri modellerinin ve yapay sinir ağlarının tasarlanması gibi süreçlerin tümü Python programlama dilinde çeşitli kütüphanelerden de yararlanılarak modüller şeklinde kodlanmıştır.

Tedavüle sürülen banknot miktarına ilişkin Türkiye geneli verileri TCMB'nin EVDS portalından, Eskişehir Şubesi verileri ise TCMB Emisyon Genel Müdürlüğü Banknot Planlama Müdürlüğünden tedarik edilmiştir.

Aylık frekansa sahip veriler, nominal değeri 200, 100, 50, 20, 10 ve 5 TL olan kupür kompozisyonundan oluşmaktadır. Veri seti 2009 yılının ocak ayından 2022 yılının aralık ayına (Aralık ayı dahil) kadar olan dönemde, kupür bazında tedavüle sürülen (ya da tedavülden çekilen) banknot adetlerini içermektedir. Nominal değeri 200 TL olan kupür, 2009 yılında tedavüle sürüldüğünden, analizlerin kupür bazında bütünlüğünün korunması amacıyla, veri seti 2009 yılının ocak ayından itibaren oluşturulmuştur. Model çıktılarının okunabilirliğini arttırmak amacıyla, veriler 1000'e bölünerek paket sayısına dönüştürülmüştür. Herhangi bir kupür için 2009 yılının ocak ayı verisi, önceki

yıllar da dahil olmak üzere tedavüle sürülen paket bazında toplam banknot miktarını ifade eder.

Eskişehir Şubesinin Reeskont Bölgesi Eskişehir, Kütahya ve Bilecik illerinden oluşur. Eskişehir Şubesinin Afyonkarahisar’da bir Banknot Deposu bulunmasına rağmen, Afyonkarahisar ili, Eskişehir Şubesinin Reeskont Bölgesine dahil değildir. Bununla birlikte Eskişehir Şubesi tarafından tedavüle sürülen ya da tedavülden çekilen banknot miktarına ilişkin verilere, Afyonkarahisar Banknot Deposu verileri de dahil edilmiştir. Böylece Eskişehir Şubesi tarafından karşılanan banknot talebini bir bütün olarak analiz etmek mümkün olmuştur.

3.1. Performans Metrikleri

Performans metrikleri özellikle de test setine ilişkin öngörü başarımlarını ölçmek amacıyla kullanılır. Hata kareleri ortalaması (mean square error, MSE), 3.1 no.lu denklemde gösterildiği gibi y_i gerçek değerler ile $\hat{y}_i$ tahmin değerlerinin farklarının kareleri toplamının örnek sayısı n ifadesine bölümüdür.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.1)$$

MSE metriğinde hata karelerinin alınmasının sebebi, hataların toplamı aşamasında, negatif sapmaların pozitif sapmaları götürmesinin engellemektir. MSE metriğinde hata kareleri alındığından dolayı, problemin sayısal ölçeğinden uzaklaşılır. Sonucun kolaylıkla yorumlanabilmesi amacıyla MSE metriğinin karekökü alınarak 3.2 no.lu denklemde gösterildiği gibi hata karelerinin ortalamasının karekökü (root mean square error, RMSE) elde edilir.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.2)$$

Öngörü literatüründe sıklıkla kullanılan RMSE, ölçeğe bağımlı bir performans metriği olması sebebiyle farklı değişkenlere sahip modellerin

öngörü performanslarının karşılaştırılmasında önerilmez (Hyndman ve Koehler, 2006). Bu çerçevede örneğin, 200 TL'lik kupüre ilişkin öngörünün RMSE metriği ile 100 TL'lik kupüre ilişkin öngörünün RMSE metriğinin kıyaslanması doğru olmayacaktır.

Öngörü doğruluğunu ölçmek için 3.3 no.lu denklemde gösterildiği gibi ortalama mutlak yüzde hatası (mean absolute percentage error) da yaygın olarak kullanılır (Hyndman ve Koehler, 2006).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3.3)$$

MAPE, ölçek uyumsuzlukları, negatif veya sıfıra yakın değerler için uygulanması önerilmez. Çünkü MAPE metriği, pozitif tahmin hatalarını, negatif tahmin hatalarından daha ağır bir şekilde cezalandırır. Bu sebeple ölçek uyumsuzlukları, negatif veya sıfıra yakın değerler için 3.4 no.lu denklemde gösterildiği üzere simetrik ortalama mutlak yüzde hatası (symmetric mean percentage error, SMAPE) kullanılabilir (Wagner, 2010).

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \quad (3.4)$$

3.2. Banknot Talebinin Tahmini

Yapay zeka teknolojileri çerçevesinde tasarlanan FNN, TLNN ve LSTM tipi çok katmanlı yapay sinir ağları, kupür bazında Türkiye geneli ve Eskişehir Şubesi verileri kullanılarak ayrı ayrı eğitilmiştir. Bu çerçevede her bir yapay sinir ağı mimarisi kullanılarak kupür bazında, her bir kupür için üç farklı ağ kullanılarak Türkiye geneli banknot talebi için 18, Eskişehir Şubesi tarafından karşılanan banknot talebi için 18 olmak üzere, toplam 36 ağ topolojisi oluşturularak eğitim gerçekleştirilmiştir.

Uygulama sürecinde kullanılan FNN, çok katmanlı geri yayımlı ve ileri beslemeli ağ mimarisine sahiptir. Girdi değerleri olarak serinin gecikmelerini kullanan TLNN çok katmanlı, geri yayımlı ileri beslemeli ağlardandır. LSTM

ise geri beslemeli ya da yinelemeli ağ mimarisine sahip yapay sinir ağıdır. Tüm ağlarda girdi olarak bağımlı değişkeni ifade eden gerçek çıktı değerleri kullanılmıştır. Bu yüzden uygulama sürecinde yararlanılan tüm ağlarda tek tip değişken (univariate) kullanılmıştır.

Yapay sinir ağları, zaman serileri (regresyon) analizinin aksine durağanlık, hata terimlerinin birbirinden bağımsız belirlenmesi ya da varyansının sabit olması gibi herhangi bir kısıt gerektirmez. Diğer taraftan yapay sinir ağları zaman serileri analizi gibi sadece doğrusal değil, doğrusal olmayan örüntüleri de çıkarabilir.

Veri seti, eğitim (training), doğrulama (validation) ve test seti olarak üçe bölünmüştür. Eğitim seti 2009 yılının ocak ayından 2020 yılının aralık ayına kadar olan 144 aylık veriyi, doğrulama seti 2021 yılının Ocak ayından Aralık ayına kadar olan 12 aylık veriyi ve öngörü (forecasting) dönemini ifade eden test seti ise 2022 yılının Ocak ayından Aralık ayına kadar olan 12 aylık veriyi içermektedir.

Ağların eğitiminde belirli herhangi bir yöntem bulunmadığından, topoloji seçimine işaret eden farklı nöron ve aktivasyon fonksiyonları gibi hiper parametrelerin kombinasyonları deneysel uygulamalarla belirlenmiştir. Tüm ağların maliyet/kayıp fonksiyonunda “ortalama hata kareleri” (MSE), öğrenme sürecinde ise “dereceli azalma” (gradient descent) öğrenme algoritması türü olan “Adam” (adaptive moment estimation) yöntemi kullanılmıştır. Ağların öğrenme parametreleri 0,001, momentum parametreleri ise 0,01 olarak belirlenmiştir.

Ağların eğitiminde kullanılan veriler Z-skor ile indirgenmiştir. Benzer şekilde ağların eğitimi sürecinde adım (epoch) sayısını belirlemede “erken durdurma” (early stopping) kriterinden yararlanılmıştır. Maliyet fonksiyonu üzerinde bulunan yerel minimumlardan kaçınmak amacıyla adım sayısı, erken durdurma kriterinin 50 adım sonrasının kontrol edilmesi ve doğrulama hatasının artmaya başlaması durumunda geriye doğru en iyi adıma geri dönülmesi şeklinde optimize edilmiştir. Bu çerçevede ağlara ilişkin hata optimizasyon süreçleri de görselleştirilerek öngörü grafiği ile birlikte

sunulmuştur. Ağların bir defada işleyeceği veri miktarı (batch size) 5 belirlenmiştir. Ağırlık matrisinin oluşturulması sürecinde, ağırlıkların başlangıç değerleri algoritma tarafından rassal olarak atanmıştır.

Zaman serileri analizi çerçevesinde ise tasarlanan SARIMA, yapay sinir ağlarında olduğu gibi kupür bazında, Türkiye geneli ve Eskişehir Şubesi verileri kullanılarak ayrı ayrı modellenmiştir. Bu çerçevede araştırma, Türkiye geneli banknot talebi için 6, Eskişehir Şubesi tarafından karşılanan banknot talebi için 6 olmak üzere, toplam 12 SARIMA modeli üzerinde gerçekleştirilmiştir. Durağan olmayan seriler normal fark ve/veya mevsimsel farkların alınması suretiyle durağanlaştırılmıştır. Benzer şekilde hata terimleri arasında olası otokorelasyon sorunlarından kurtulmak amacıyla, ilave fark alma işlemi ve/veya normal fark ile mevsimsel fark kombinasyonlarını değiştirmek yeterli olmuştur. Her öngörü grafiği ile birlikte modele ilişkin hata terimlerinin otokorelasyon fonksiyonu da görselleştirilerek sunulmuştur. Zaman serileri kapsamında toplam 12 model üzerinde araştırma yapıldığından, tüm modeller için değişen varyans sorununa ilişkin genel bir çözüm bulunamamıştır. Bu kapsamda araştırmanın amacını ve odağını dağıtacağı kaygısıyla, değişen varyans sorunu için modele özgü tekil çözümlere yer verilmemiştir. SARIMA modellerinin geliştirilmesi aşamasında açık kaynak bir Python kütüphanesi olan “pmdarima” kütüphanesinden yararlanılmıştır.

Verilerin dosyadan okunup işlenmesinden modellerin tasarlanması, eğitilmesi ve öngörülerin yapılmasına kadar uçtan uca tüm süreçlerin inşasında kullanılan Python kütüphaneleri aşağıda listelenmiştir.

- | | | |
|--------------|--------------|---------------|
| ▪ tensorflow | ▪ matplotlib | ▪ random |
| ▪ keras | ▪ calendar | ▪ statsmodels |
| ▪ sklearn | ▪ datetime | ▪ pmdarima |
| ▪ numpy | ▪ math | ▪ tabulate |
| ▪ pandas | ▪ statistics | ▪ inflect |

3.2.1. Türkiye Geneli İçin Kupürlere Göre Tahminler

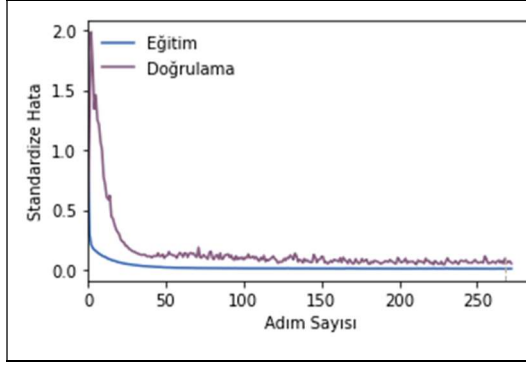
Türkiye geneli için kupür bazında tahmin yapmak amacıyla, 200 TL'lik kupür verilerine göre yapay sinir ağlarının mimarisi ve topolojisi ile SARIMA modeli dahil yapay sinir ağlarının performans metrikleri Tablo 3.1'de sunulmuştur.

TABLO 3.1. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 200 TL

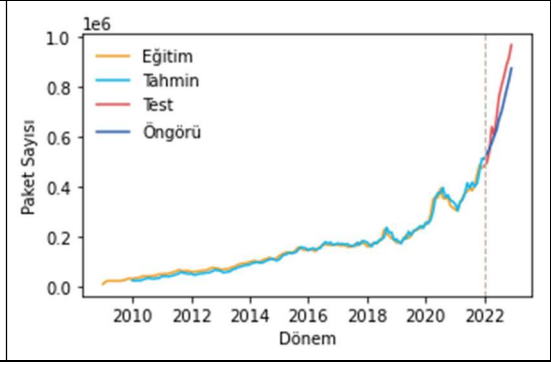
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12	7	
Gizli Nöronlar	8	6	3	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	268	78	65	
Eğitim RMSE (paket)	13.238	14.808	18.529	12.057
Test RMSE (paket)	73.188	106.747	70.721	69.491
Eğitim SMAPE	5,28	3,22	5,26	6,52
Test SMAPE	4,41	6,21	4,14	7,82

Türkiye geneli 200 TL'lik kupür verileriyle eğitilen üç yapay sinir ağı arasında, test setine ilişkin SMAPE ve RMSE metriklerine göre FNN ve LSTM yakın bir öngörü performansı göstermiştir. Bu veri seti için TLNN ağının, diğer iki ağı göre eğitim performansı daha iyi olmasına rağmen, test performansı nispeten daha zayıftır. FNN ve LSTM performansına yakın olmakla birlikte test setine ilişkin RMSE metriğine göre en iyi tahmin performansı SARIMA(2,2,0)x(0,1,1)₁₂ modelindedir (Tablo 3.1). SARIMA(2,2,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 1'de yer almaktadır.

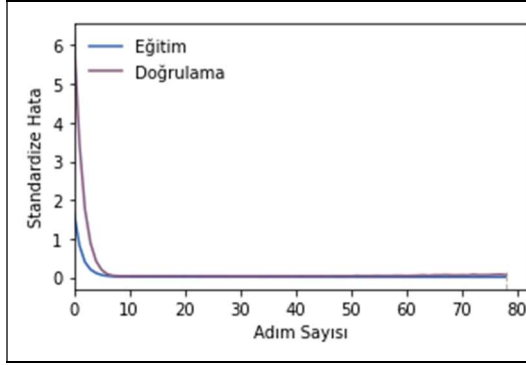
Grafik 3.1.a, b, c, d, e, f, g ve h'de Türkiye geneli 200 TL'lik kupür verileri kullanılarak elde edilen, her üç ağı hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile SARIMA(2,2,0)x(0,1,1)₁₂ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.



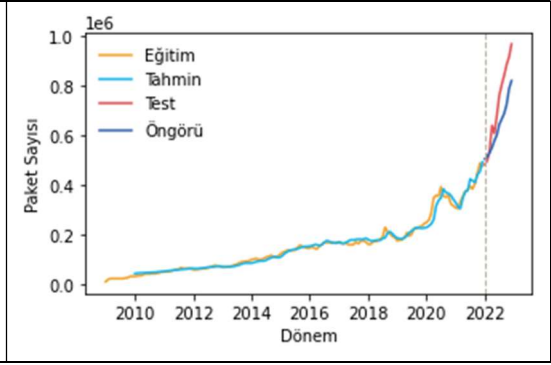
Grafik 3.1.a. Türkiye Geneli: 200 TL için FNN Hata Optimizasyonu



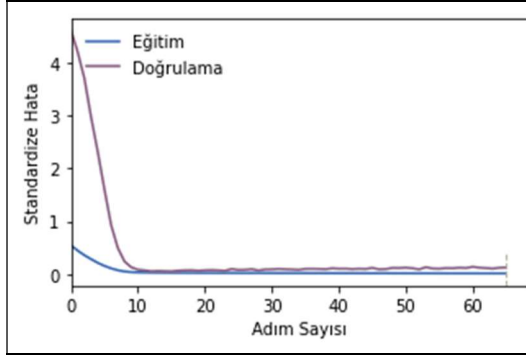
Grafik 3.1.b. Türkiye Geneli: 200 TL için FNN Çıktısı



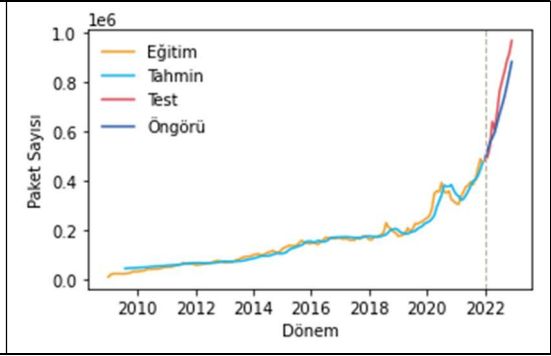
Grafik 3.1.c. Türkiye Geneli: 200 TL için TLNN Hata Optimizasyonu



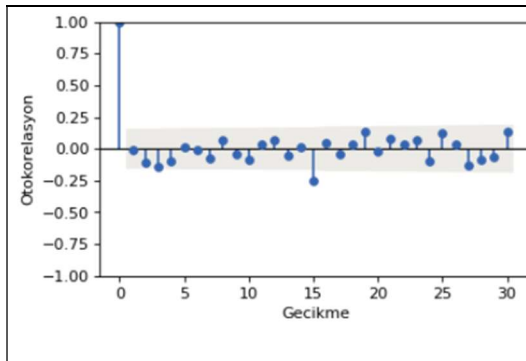
Grafik 3.1.d. Türkiye Geneli: 200 TL için TLNN Çıktısı



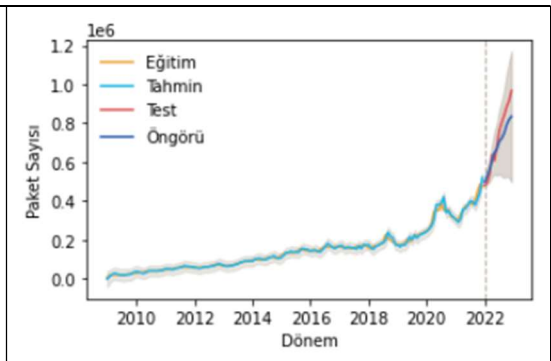
Grafik 3.1.e. Türkiye Geneli: 200 TL için LSTM Hata Optimizasyonu



Grafik 3.1.f. Türkiye Geneli: 200 TL için LSTM Çıktısı



Grafik 3.1.g. Türkiye Geneli: 200 TL için Hataların Otokorelasyonu



Grafik 3.1.h. Türkiye Geneli: 200 TL için SARIMA Çıktısı

Hata optimizasyonu sürecinin gösterildiği Grafik 3.1.a, c ve e’de yer alan “Standardize Hata”, Z-skor ile standardize edilen gerçek çıktı ile hesaplanan çıktı değerleri arasındaki farkı ifade etmektedir. Ağ çıktılarına ilişkin Grafik 3.1.b, d ve f’de ise eğitim ve eğitim verilerinin tahmini ile test ve test dönemine ilişkin ağların öngörüsünü temsil etmektedir. Öngörü dönemi, 2022 yılının tümünü kapsamaktadır. Genel olarak üç ağın da Türkiye genelinde 200’lik kupüre olan talep eğilimini yakaladığı ve bu bilgiyi öngörü döneminde başarılı bir şekilde kullandığı görülmektedir. Benzer şekilde bir zaman serisi modeli olan SARIMA(2,2,0)x(0,1,1)₁₂ modelinin de öngörü dönemine ilişkin eğilimi başarılı bir şekilde takip ettiği Grafik 3.1.h’de görülmektedir. Grafik 3.1.g’de gösterildiği üzere, hata terimlerinin otokorelasyon değerlerinin büyük ölçüde gri alanla işaretli güven aralıklarının sınırları dahilinde kalması, SARIMA(2,2,0)x(0,1,1)₁₂ modelinde otokorelasyon sorunu olmadığına işaret etmektedir. Otokorelasyon sorununun yokluğuna ilişkin benzer sonuca, Ek-1’de sunulan Tablo-1’den görüleceği üzere, 0,80 olan Prob(Q) değerinin 0,05 seviyesinin üzerinde olmasından da ulaşılabılır.

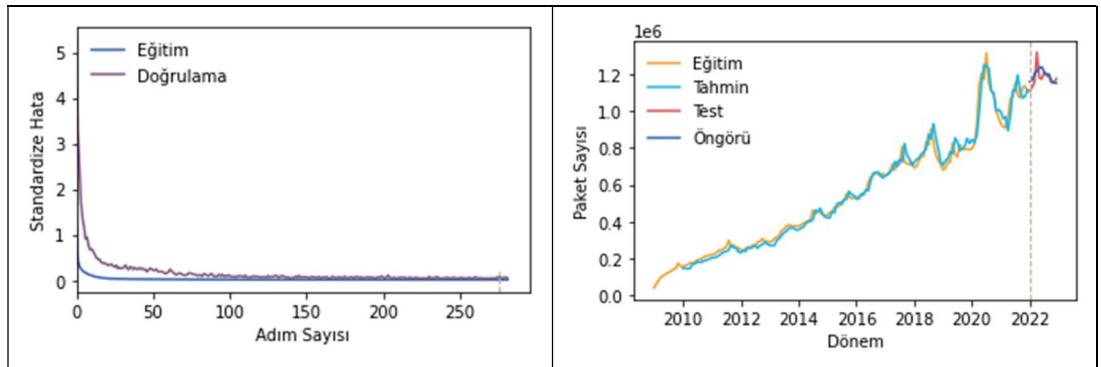
200 TL’lik kupüre olan talebin son yıllara kadar Kurban ve Ramazan Bayramı dönemlerinde önemli bir mevsimsel değişkenlik göstermediği ifade edilebilir. Ancak 2019 yılından sonra talebin üssel bir şekilde arttığı görülmektedir (Grafik 3.1.b).

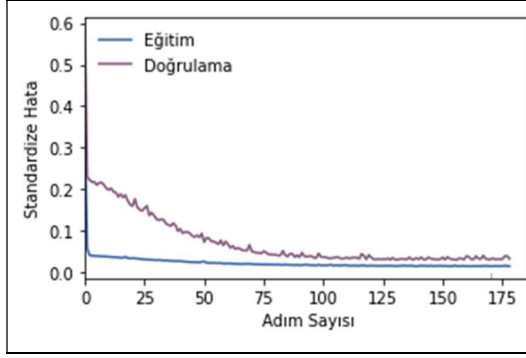
Türkiye geneli 100 TL’lik kupür verilerine göre yapay sinir ağlarının mimarisi ve topolojisi ile SARIMA modeli dahil yapay sinir ağlarının performans metrikleri Tablo 3.2’de sunulmuştur.

TABLO 3.2. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 100 TL

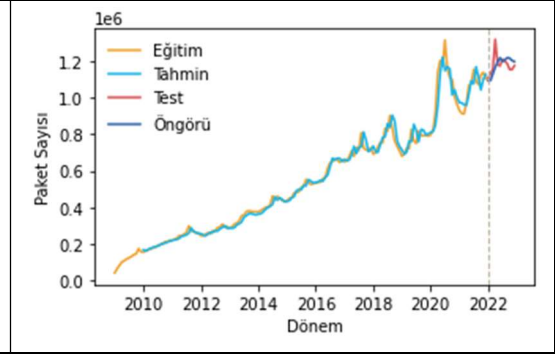
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 10, 11, 12	8	
Gizli Nöronlar	10	8	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	275	170	169	
Eğitim RMSE (paket)	39.233	38.829	39.263	37.620
Test RMSE (paket)	45.462	54.632	49.430	60.112
Eğitim SMAPE	2,74	1,98	2,51	4,62
Test SMAPE	1,43	1,65	1,09	3,71

Türkiye geneli 100 TL'lik kupür verileriyle eğitilen üç yapay sinir ağı arasında, SMAPE ve RMSE metriklerine göre FNN en iyi öngörü performansını göstermiştir. TLNN ağı, diğer iki ağa göre daha iyi eğitim performansına sahip olmasına rağmen, yaklaşık 9.000 paketlik daha fazla sapma ile görece daha zayıf bir öngörü (test) performansına sahiptir. Eğitim setine ilişkin RMSE metriği hariç, diğer tüm metriklerde en kötü tahmin performansı SARIMA(1,0,0)x(0,1,1)₁₂ modelindedir (Tablo 3.2). SARIMA(1,0,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 2'de yer almaktadır. Grafik 3.2.a, b, c, d, e, f, g ve h'de Türkiye geneli 100 TL'lik kupür verileri kullanılarak elde edilen, her üç ağın hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile SARIMA(1,0,0)x(0,1,1)₁₂ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.

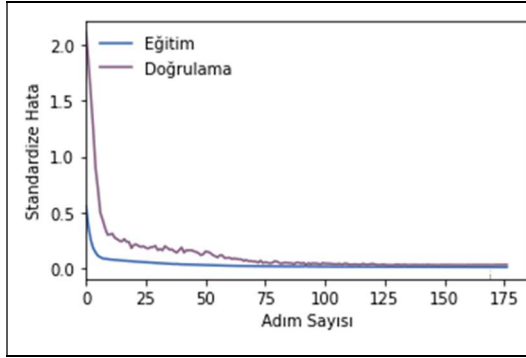
**Grafik 3.2.a. Türkiye Geneli: 100 TL İçin FNN Hata Optimizasyonu****Grafik 3.2.b. Türkiye Geneli: 100 TL İçin FNN Çıktısı**



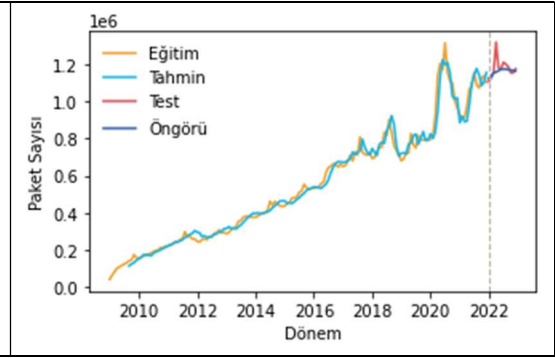
Grafik 3.2.c. Türkiye Geneli: 100 TL için TLNN Hata Optimizasyonu



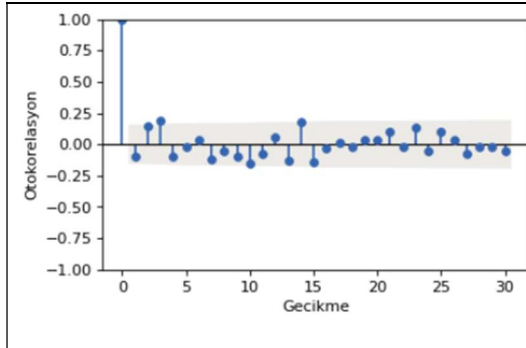
Grafik 3.2.d. Türkiye Geneli: 100 TL için TLNN Çıktısı



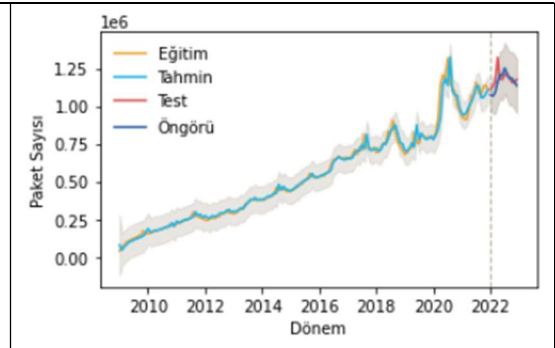
Grafik 3.2.e. Türkiye Geneli: 100 TL için LSTM Hata Optimizasyonu



Grafik 3.2.f. Türkiye Geneli: 100 TL için LSTM Çıktısı



Grafik 3.2.g. Türkiye Geneli: 100 TL için Hataların Otokorelasyonu



Grafik 3.2.h. Türkiye Geneli: 100 TL için SARIMA Çıktısı

Genel olarak tüm modellerin Türkiye genelinde 100 TL'lik kupüre olan talebin eğilimini yakaladığı ve bu bilgiyi öngörü döneminde başarılı bir şekilde kullandığı görülmektedir (Grafik 3.2.b, d, f, h). Ancak öngörü/test döneminde mevsimsel etkiyi en iyi tahmin eden ağın FNN olduğu anlaşılmaktadır. Hata terimlerine ilişkin otokorelasyon fonksiyonu, $SARIMA(1,0,0) \times (0,1,1)_{12}$ modelinde otokorelasyon sorunu olmadığına işaret etmektedir (Şekil 3.2.g). Otokorelasyon sorununun yokluğuna ilişkin benzer sonuca, Ek-1'de sunulan

Tablo-2’de görüleceği üzere, 0,23 olan Prob(Q) değerinin 0,05 seviyesinin üzerinde olmasından da ulaşılabılır.

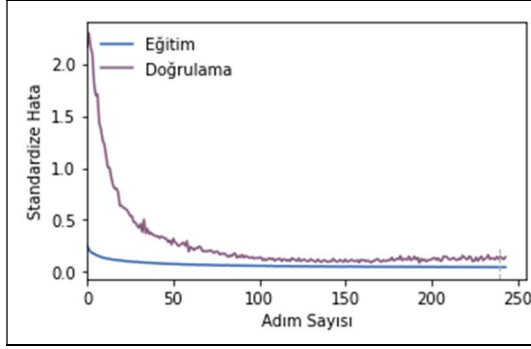
100 TL’lik kupüre olan talep, özellikle Kurban Bayramı dönemlerinde zirve noktasına ulaşmaktadır. Takip eden birkaç ay sonrasında ise normal seyrine geri döndüğü, mevsimsel sebeplerle talepte dalgalanmalar meydana geldiği, talebin nispeten dik bir trend etrafında doğrusal bir şekilde sürekli arttığı anlaşılmaktadır. 2020 yılında 100 TL’lik kupüre olan talepte ciddi bir mevsimsel dalgalanma olmasına rağmen, trendde herhangi bir değişiklik gözlenmemiştir (Grafik 3.2.b).

Türkiye geneli 50 TL’lik kupür verilerine göre eğitilen ağlara ve SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.3’te sunulmuştur.

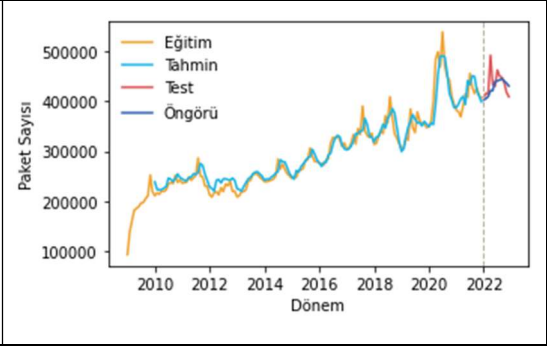
TABLO 3.3. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 50 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 4, 12, 13	8	
Gizli Nöronlar	6	9	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	239	131	87	
Eğitim RMSE (paket)	18.161	18.576	18.775	22.961
Test RMSE (paket)	23.598	26.489	21.754	22.056
Eğitim SMAPE	2,09	1,81	2,16	6,45
Test SMAPE	1,77	1,97	1,57	4,19

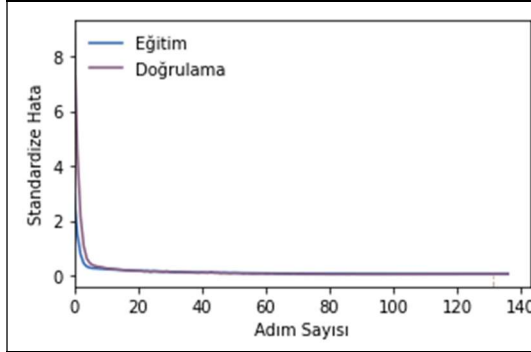
Türkiye geneli 50 TL’lik kupür verileriyle eğitilen üç yapay sinir ağının da eğitim performansı birbirine yakın olmasına rağmen, öngörü/test döneminde LSTM ağı en iyi performansı göstermiştir. Test verisine ilişkin RMSE metriğine göre LSTM ve SARIMA(1,1,0)x(0,1,1)₁₂ modeli birbirine yakın bir öngörü performansında bulunduğu anlaşılmaktadır (Tablo 3.3). SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1’de sunulan Tablo 3’te yer almaktadır. Grafik 3.3.a, b, c, d, e, f, g ve h’de Türkiye geneli 50 TL’lik kupür verilerine ilişkin hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile SARIMA(1,1,0)x(0,1,1)₁₂ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.



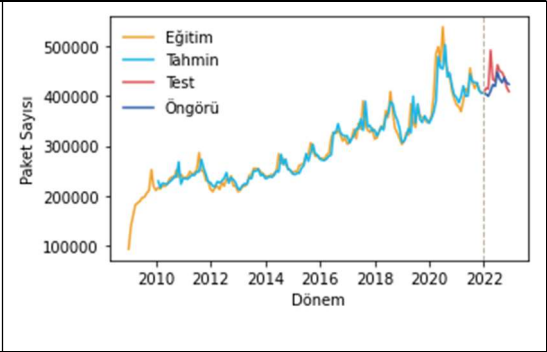
Grafik 3.3.a. Türkiye Geneli: 50 TL için FNN Hata Optimizasyonu



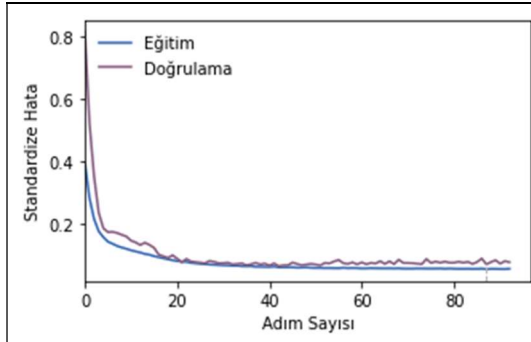
Grafik 3.3.b. Türkiye Geneli: 50 TL için FNN Çıktısı



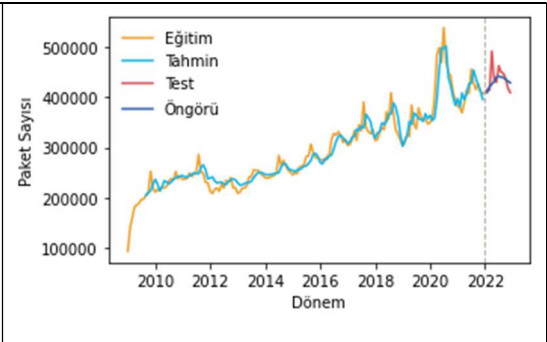
Grafik 3.3.c. Türkiye Geneli: 50 TL için TLNN Hata Optimizasyonu



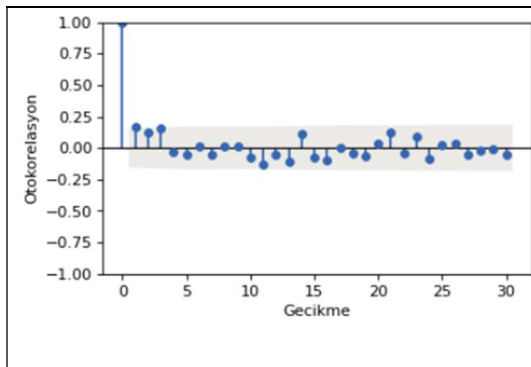
Grafik 3.3.d. Türkiye Geneli: 50 TL için TLNN Çıktısı



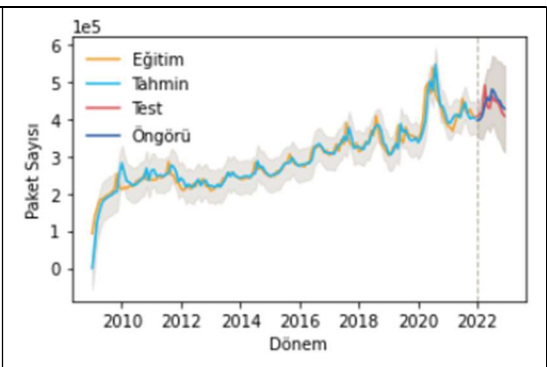
Grafik 3.3.e. Türkiye Geneli: 50 TL için LSTM Hata Optimizasyonu



Grafik 3.3.f. Türkiye Geneli: 50 TL için LSTM Çıktısı



Grafik 3.3.g. Türkiye Geneli: 50 TL için Hataların Otokorelasyonu



Grafik 3.3.h. Türkiye Geneli: 50 TL için SARIMA Çıktısı

LSTM ağı en iyi performansa sahip olmakla birlikte, genel olarak tüm modellerin öngörü/test döneminin eğilimlerini benzer bir örüntüyle tahmin ettiği görülmektedir (Grafik 3.3.b, d, f, h). Ancak mevsimsel hareketleri yakalamada, herhangi bir otokorelasyon sorunu barındırmayan SARIMA(1,1,0)x(0,1,1)₁₂ modelinin, LSTM ağından daha başarılı olduğu ifade edilebilir.

50 TL'lik kupüre olan talep, 100 TL'lik kupüre olan talep ile oldukça benzer bir yapıdadır. Özellikle Kurban Bayramı dönemlerinde zirve noktasına ulaşmaktadır. Takip eden birkaç ay sonrasında ise normal seyrine geri döndüğü, mevsimsel sebeplerle talepte dalgalanmalar meydana geldiği, talebin dik olmayan bir trend etrafında doğrusal bir şekilde sürekli arttığı anlaşılmaktadır. 2020 yılında 100 TL'lik kupürde görülen mevsimsel dalgalanma, 50 TL'lik kupürde de belirgin bir şekilde kendini göstermektedir. Bununla birlikte söz konusu talep artışı, trendde herhangi bir değişikliğe sebep olmamıştır (Grafik 3.3.b).

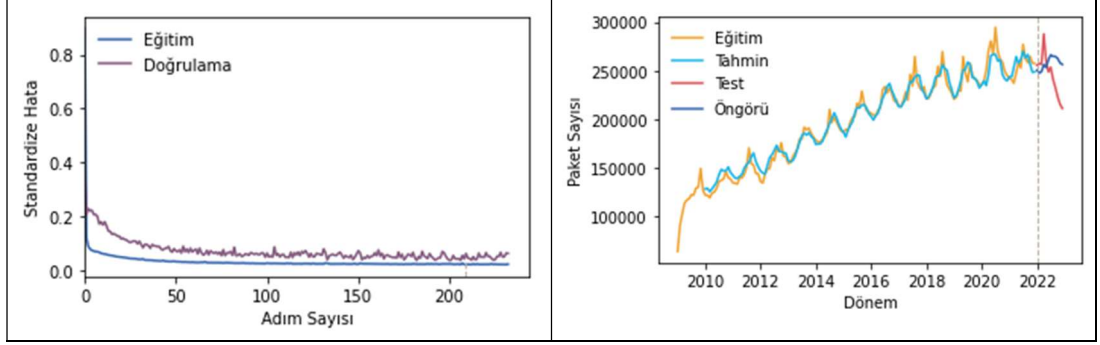
Türkiye geneli 20 TL'lik kupür verilerine göre eğitilen ağlara ve SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.4'te sunulmuştur.

TABLO 3.4. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 20 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12, 13	9	
Gizli Nöronlar	8	6	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	209	86	117	
Eğitim RMSE (paket)	7.758	8.273	8.235	12.653
Test RMSE (paket)	26.612	28.123	24.402	35.062
Eğitim SMAPE	1,47	1,51	1,56	5,74
Test SMAPE	4,52	4,44	3,93	11,51

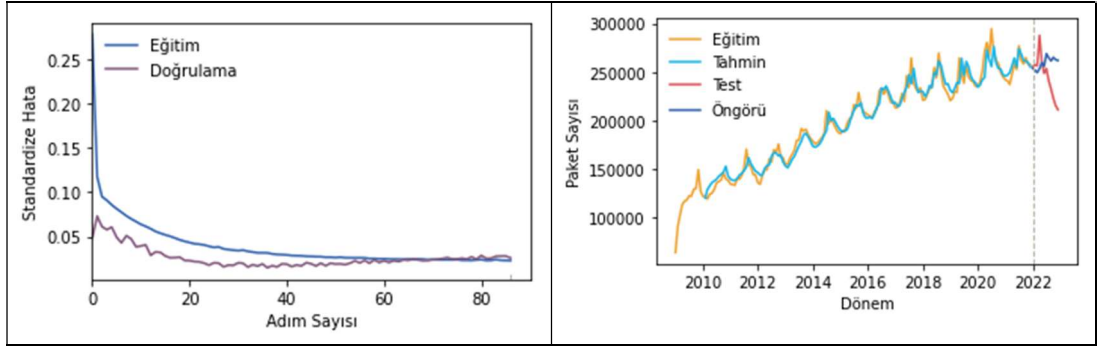
Türkiye geneli 20 TL'lik kupür verileriyle eğitilen üç yapay sinir ağının da eğitim performansı birbirine yakın olmasına rağmen, en iyi öngörü performansını LSTM ağı göstermiştir. Hata terimlerinin otokorelasyonuna ilişkin herhangi bir sorun barındırmayan SARIMA(1,1,0)x(0,1,1)₁₂ modelinin ise diğer üç model arasında en başarısız öngörüye sahip model olduğu

söylenabilir (Tablo 3.4). SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1’de sunulan Tablo 4’te yer almaktadır. Grafik 3.4.a, b, c, d, e, f, g ve h’de Türkiye geneli 20 TL’lik kupür verilerine ilişkin ağ çıktılarının ve SARIMA(1,1,0)x(0,1,1)₁₂ modelinin görsellerine yer verilmiştir.



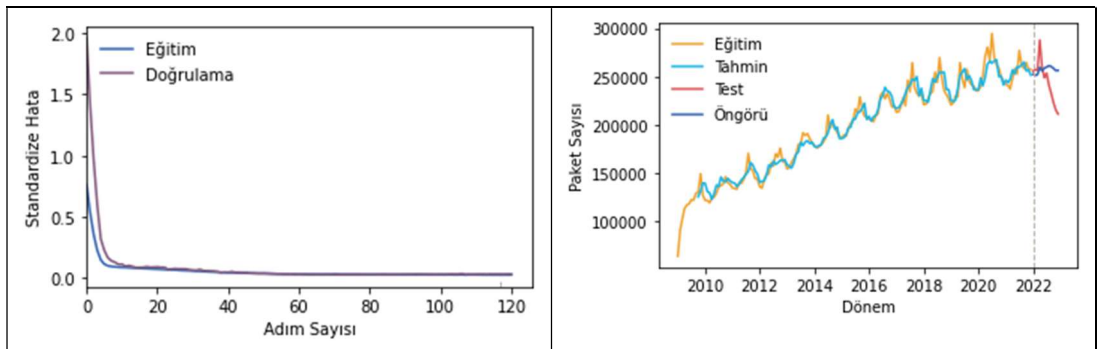
Grafik 3.4.a. Türkiye Geneli: 20 TL İçin FNN Hata Optimizasyonu

Grafik 3.4.b. Türkiye Geneli: 20 TL İçin FNN Çıktısı



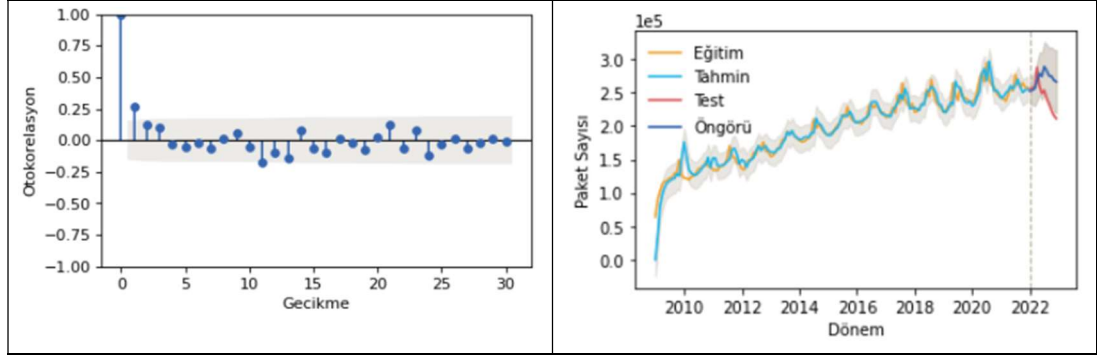
Grafik 3.4.c. Türkiye Geneli: 20 TL İçin TLNN Hata Optimizasyonu

Grafik 3.4.d. Türkiye Geneli: 20 TL İçin TLNN Çıktısı



Grafik 3.4.e. Türkiye Geneli: 20 TL İçin LSTM Hata Optimizasyonu

Grafik 3.4.f. Türkiye Geneli: 20 TL İçin LSTM Çıktısı



Grafik 3.4.g. Türkiye Geneli: 20 TL İçin Hataların Otokorelasyonu

Grafik 3.4.h. Türkiye Geneli: 20 TL İçin SARIMA Çıktısı

Öngörü dönemine ilişkin Türkiye geneli 20 TL'lik kupür talebinde, her dört model tarafından çıkartılan örüntüye uygun olmayan aşağı yönlü önemli bir kırılma olduğu görülmektedir. Diğer bir ifadeyle kullanılan modellerin hiçbirinin söz konusu kırılmayı öngörmeye başarılı olamadığı söylenebilir (Grafik 3.4.b, d, f, h).

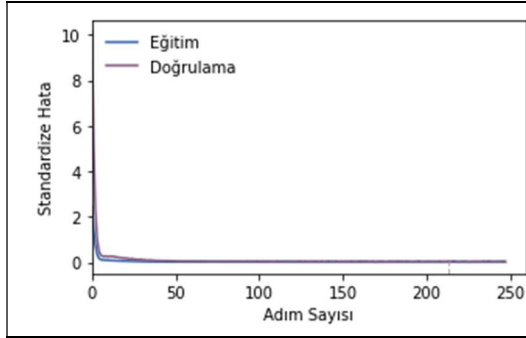
20 TL'lik kupüre olan talep, Kurban Bayramı dönemlerinden kaynaklanan mevsimsel dalgalanmalarla düzenli bir yapıya sahiptir. Ancak 2022 yılının ikinci yarısından sonra talebin önemli ölçüde azaldığı ve trendin altına düştüğü gözlenmektedir (Grafik 3.4.b).

Türkiye geneli 10 TL'lik kupür verilerine göre eğitilen ağlara ve SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.5'te sunulmuştur.

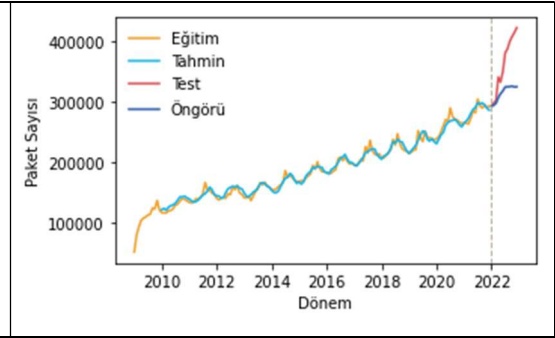
TABLO 3.5. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 10 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12	7	
Gizli Nöronlar	5	5	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	213	86	248	
Eğitim RMSE (paket)	6.053	6.930	6.854	10.389
Test RMSE (paket)	57.871	48.915	20.989	61.663
Eğitim SMAPE	1,24	1,32	1,34	5,30
Test SMAPE	6,55	5,55	2,39	14,06

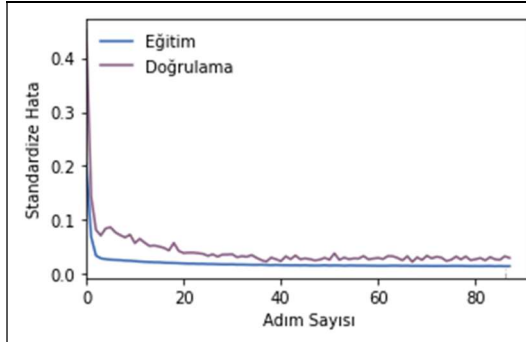
Türkiye geneli 10 TL'lik kupür verileriyle eğitilen üç yapay sinir ağı arasında, SMAPE ve RMSE metriklerine göre LSTM önemli bir farkla en iyi öngörü performansına sahiptir. SARIMA(1,1,0)x(0,1,1)₁₂ modelinin ise en başarısız öngörüye sahip olduğu anlaşılmaktadır (Tablo 3.5). SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 5'te yer almaktadır. Grafik 3.5.a, b, c, d, e, f, g ve h'de Türkiye geneli 10 TL'lik kupür verilerine ilişkin ağ çıktılarının ve SARIMA(1,1,0)x(0,1,1)₁₂ modelinin görsellerine yer verilmiştir.



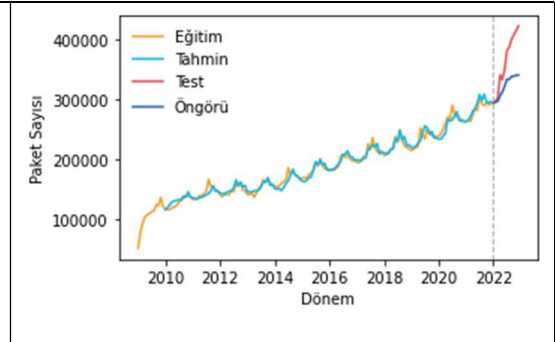
Grafik 3.5.a. Türkiye Geneli: 10 TL için FNN Hata Optimizasyonu



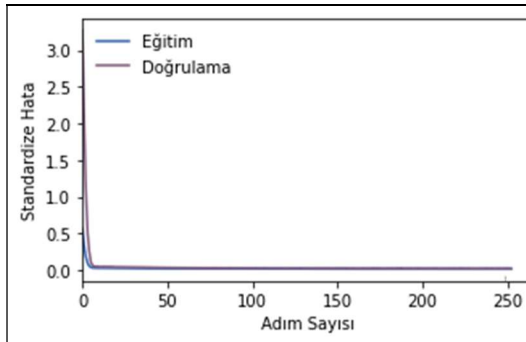
Grafik 3.5.b. Türkiye Geneli: 10 TL için FNN Çıktısı



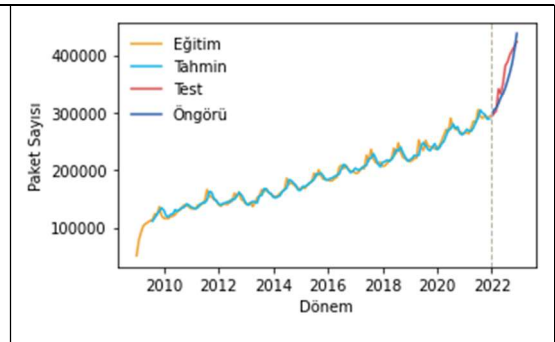
Grafik 3.5.c. Türkiye Geneli: 10 TL için TLNN Hata Optimizasyonu



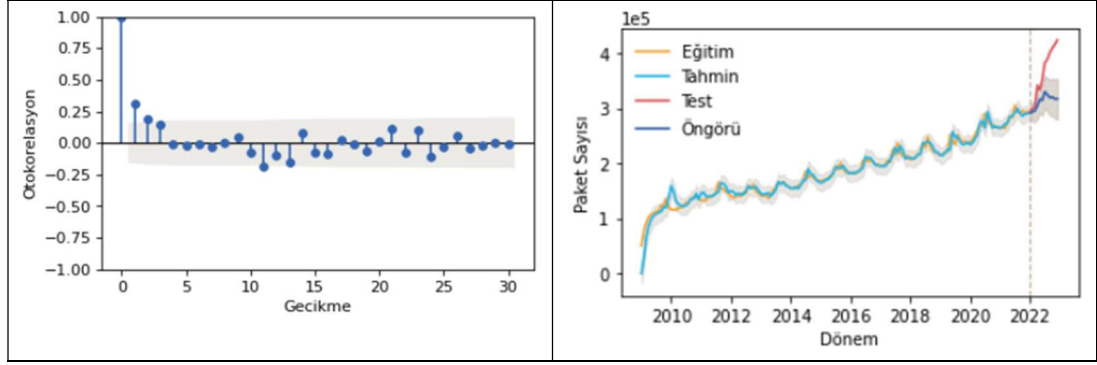
Grafik 3.5.d. Türkiye Geneli: 10 TL için TLNN Çıktısı



Grafik 3.5.e. Türkiye Geneli: 10 TL için LSTM Hata Optimizasyonu



Grafik 3.5.f. Türkiye Geneli: 10 TL için LSTM Çıktısı



Grafik 3.5.g. Türkiye Geneli: 10 TL İçin Hataların Otokorelasyonu

Grafik 3.5.h. Türkiye Geneli: 10 TL İçin SARIMA Çıktısı

Öngörü dönemine ilişkin Türkiye geneli 10 TL'lik kupür talebinde yukarı yönlü önemli bir kırılma olduğu görülmektedir (Grafik 3.5.b, d, f, h). Söz konusu kırılmayı başarılı bir şekilde takip edebilen tek model LSTM ağı olmuştur.

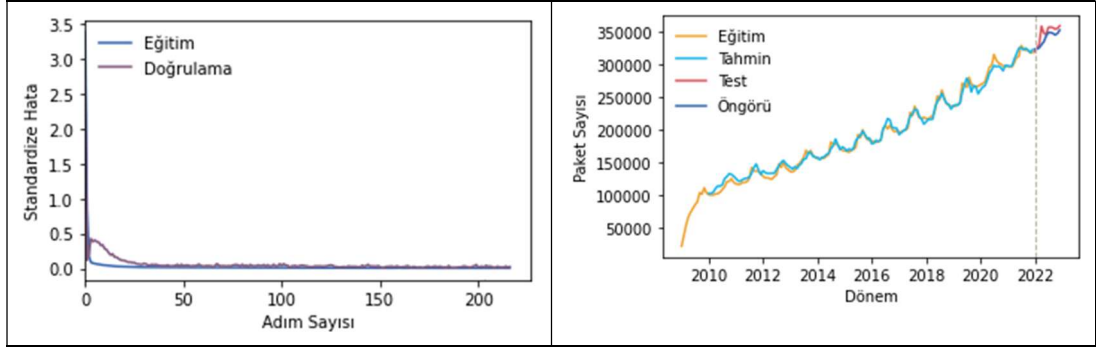
10 TL'lik kupüre olan talep de 20 TL'lik kupür gibi Kurban Bayramı dönemlerinden kaynaklanan mevsimsel dalgalanmalarla düzenli bir yapıya sahiptir. Ancak 20 TL'lik kupürün aksine, 2022 yılının ikinci yarısından sonra talebin önemli ölçüde arttığı ve trendin üzerine çıktığı gözlenmektedir. (Grafik 3.4.b). Bu durum 20 TL'lik kupürün talebi ile 10 TL'lik kupürün talebi arasında bir tür ikamenin olabileceğini düşündürmektedir.

Türkiye geneli 5 TL'lik kupür verilerine göre eğitilen ağlara ve SARIMA(0,1,2)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.6'da sunulmuştur.

TABLO 3.6. TÜRKİYE GENELİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 5 TL

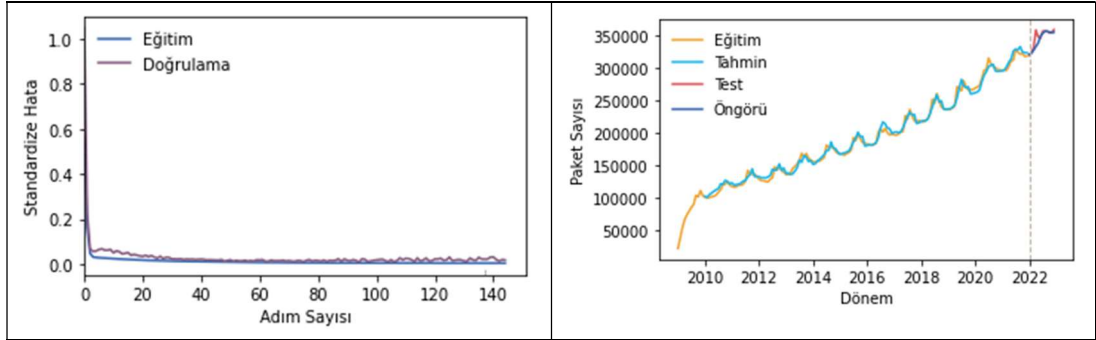
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12	8	
Gizli Nöronlar	8	5	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	203	137	114	
Eğitim RMSE (paket)	6.069	5.732	6.810	7.006
Test RMSE (paket)	10.880	8.202	7.578	13.147
Eğitim SMAPE	1,37	1,19	1,41	4,83
Test SMAPE	1,23	0,69	0,83	3,11

Türkiye geneli 5 TL'lik kupür verileriyle eğitilen LSTM ağının, SMAPE ve RMSE metriklerine göre eğitim performansı diğer iki ağı göre nispeten daha zayıf olmasına rağmen, öngörü/test dönemine ilişkin performansı diğer modellerden daha başarılıdır (Tablo 3.6). SARIMA(0,1,2)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 6'da yer almaktadır. Grafik 3.6.a, b, c, d, e, f, g ve h'de Türkiye geneli 5 TL'lik kupür verilerine ilişkin ağ çıktılarının ve SARIMA(0,1,2)x(0,1,1)₁₂ modelinin görsellerine yer verilmiştir.



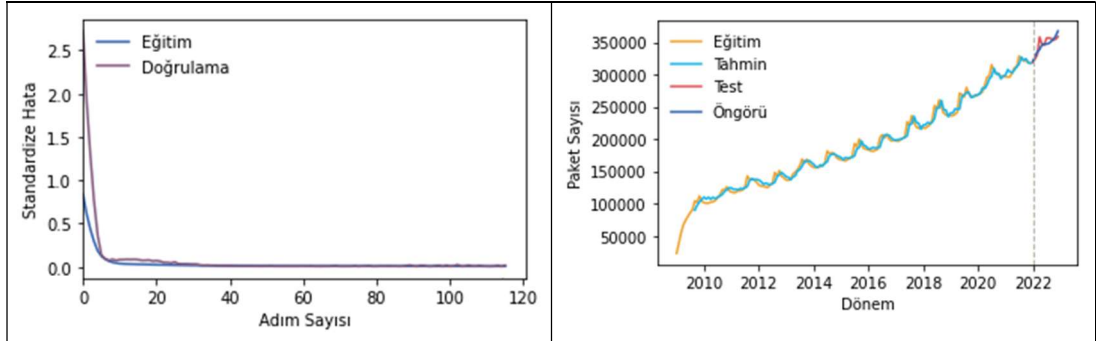
Grafik 3.6.a. Türkiye Geneli: 5 TL İçin FNN Hata Optimizasyonu

Grafik 3.6.b. Türkiye Geneli: 5 TL İçin FNN Çıktısı



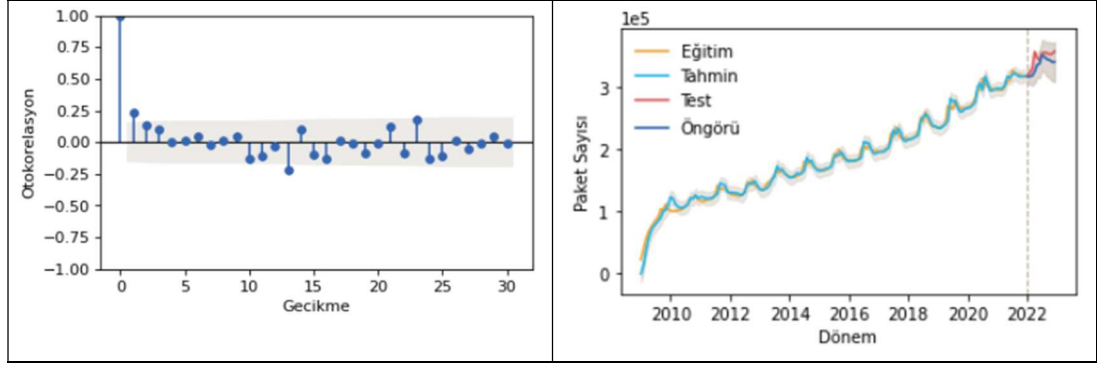
Grafik 3.6.c. Türkiye Geneli: 5 TL İçin TLNN Hata Optimizasyonu

Grafik 3.6.d. Türkiye Geneli: 5 TL İçin TLNN Çıktısı



Grafik 3.6.e. Türkiye Geneli: 5 TL İçin LSTM Hata Optimizasyonu

Grafik 3.6.f. Türkiye Geneli: 5 TL İçin LSTM Çıktısı



Grafik 3.6.g. Türkiye Geneli: 5 TL İçin Hataların Otokorelasyonu

Grafik 3.6.h. Türkiye Geneli: 5 TL İçin SARIMA Çıktısı

Öngörü dönemine ilişkin Türkiye geneli 5 TL'lik kupür talebini en iyi tahmin eden LSTM ağı olmasına rağmen, dört modelin çıkarttığı örüntünün de birbirine benzer/yakın olduğu görülmektedir (Grafik 3.6.b, d, f, h).

5 TL'lik kupüre olan talep Kurban Bayramı dönemlerinden kaynaklanan mevsimsel dalgalanmalarla son derece düzenli bir yapıya sahiptir. Trendde herhangi belirgin bir değişiklik gözlenmemektedir (Grafik 3.4.b).

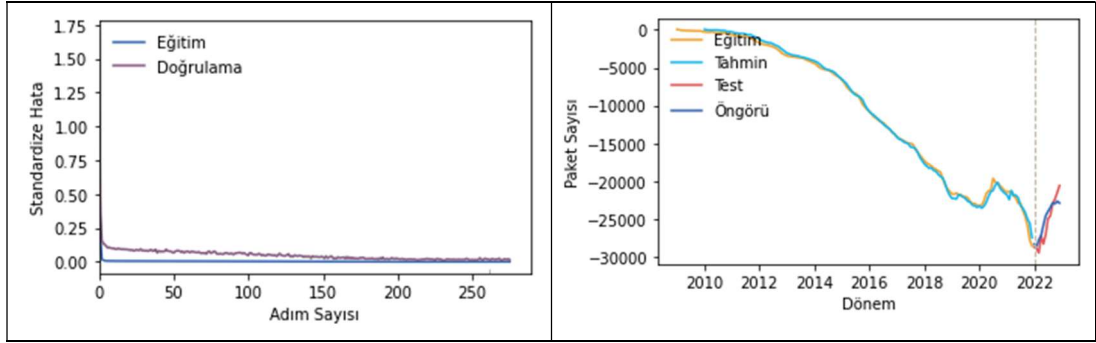
3.2.2. Eskişehir Şubesi İçin Kupürlere Göre Tahminler

Eskişehir Şubesi Reeskont Bölgesi için kupür bazında tahmin yapmak amacıyla, 200 TL'lik kupür verilerine göre yapay sinir ağlarının mimarisi ve topolojisi ile SARIMA modeli dahil yapay sinir ağlarının performans metrikleri Tablo 3.7'de sunulmuştur.

TABLO 3.7. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 200 TL

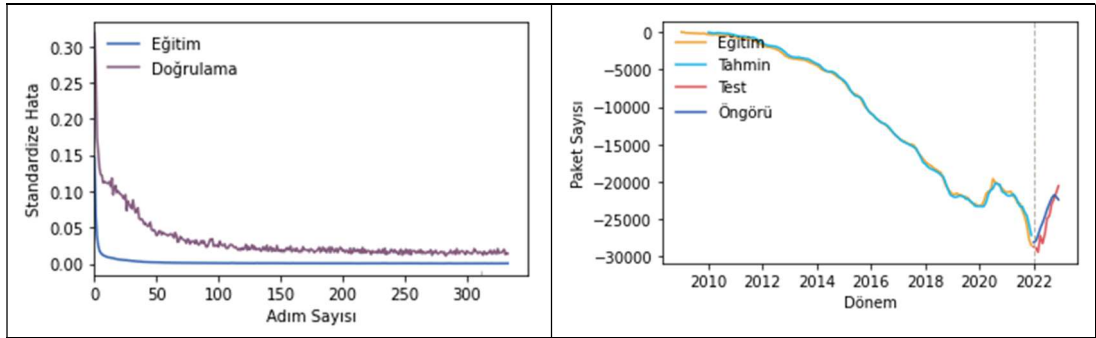
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 4, 11, 12	6	
Gizli Nöronlar	8	8	3	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	261	312	320	
Eğitim RMSE (paket)	430	380	408	228
Test RMSE (paket)	1.437	1.560	1.900	13.762
Eğitim SMAPE	9,86	7,46	3,41	6,12
Test SMAPE	2,31	2,72	3,20	35,63

Eskişehir Şubesinin 200 TL'lik kupür verileriyle eğitilen üç yapay sinir ağı arasında, RMSE metriğine göre FNN ve TLNN oldukça yakın bir performans göstermiştir. Bu veri seti için LSTM ağının hem eğitim hem de öngörü/test performansı diğer iki ağa göre nispeten daha zayıftır. Ancak test dönemine ilişkin SMAPE metriğine göre üç ağın da benzer bir performansa sahip olduğu anlaşılmaktadır. Diğer taraftan test setine (öngörü dönemine) ilişkin en kötü tahmin performansı $SARIMA(2,2,0) \times (1,0,0)_{12}$ modelindedir (Tablo 3.7). Söz konusu SARIMA modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 7'de yer almaktadır. Grafik 3.7.a, b, c, d, e, f, g ve h'de Eskişehir Şubesi 200 TL'lik kupür verileri kullanılarak elde edilen, her üç ağın hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile $SARIMA(2,2,0) \times (1,0,0)_{12}$ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.



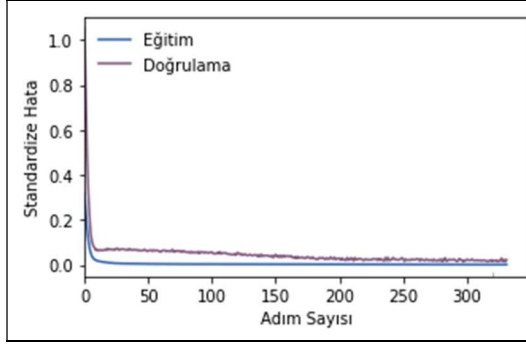
Grafik 3.7.a. Eskişehir Şubesi: 200 TL için FNN Hata Optimizasyonu

Grafik 3.7.b. Eskişehir Şubesi: 200 TL için FNN Çıktısı

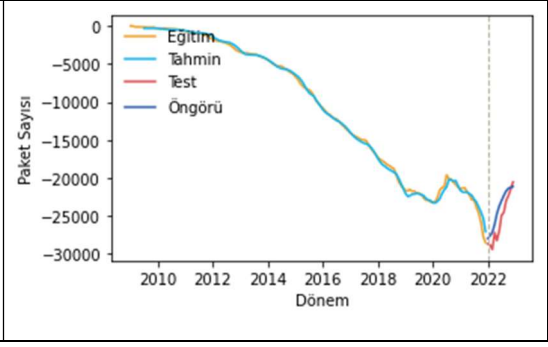


Grafik 3.7.c. Eskişehir Şubesi: 200 TL için TLNN Hata Optimizasyonu

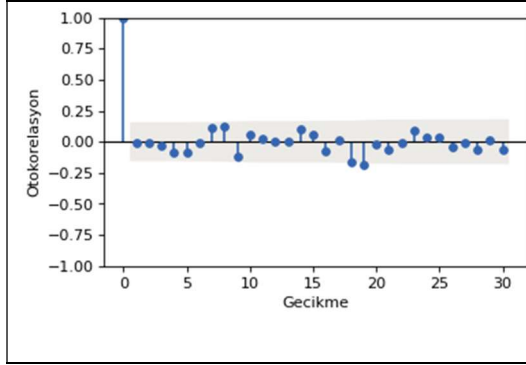
Grafik 3.7.d. Eskişehir Şubesi: 200 TL için TLNN Çıktısı



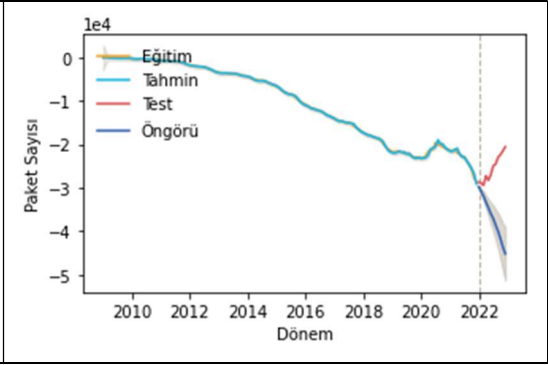
Grafik 3.7.e. Eskişehir Şubesi: 200 TL için LSTM Hata Optimizasyonu



Grafik 3.7.f. Eskişehir Şubesi: 200 TL için LSTM Çıktısı



Grafik 3.7.g. Eskişehir Şubesi: 200 TL için Hataların Otokorelasyonu



Grafik 3.7.h. Eskişehir Şubesi: 200 TL için SARIMA Çıktısı

Eskişehir Şubesi özelinde, öngörü dönemine ilişkin 200 TL'lik kupürde gözlenen yukarı yönlü hareketi, genel olarak tüm yapay sinir ağlarının yakaladığı görülmektedir (Grafik 3.7.b, d, f). Ancak SARIMA(2,2,0)x(1,0,0)₁₂ modeli ise yukarı yönlü hareketi izlemede başarısız bir öngörüye sahiptir (Grafik 3.7.h).

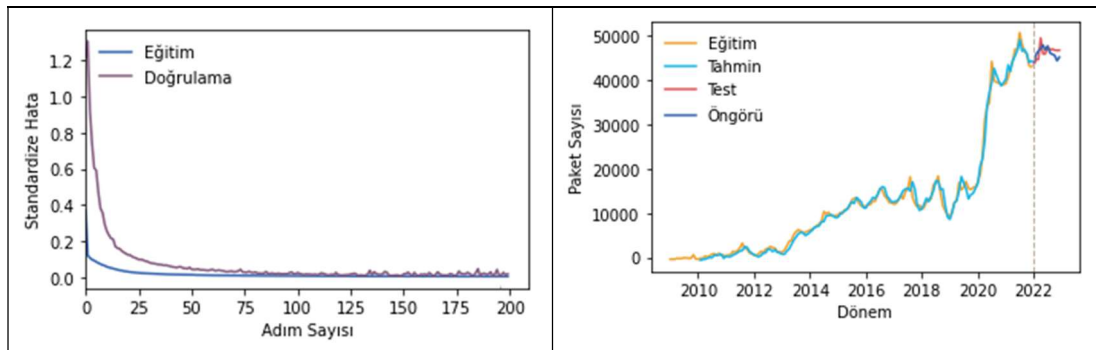
Eskişehir Şubesi özelinde 200 TL'lik kupüre pozitif bir talep yoktur. Söz konusu negatif talep diğer bölgelerden piyasaya sürülen 200 TL'lik banknotların Eskişehir Şubesi tarafından tedavülden çekildiğine işaret etmektedir. Belirgin bir mevsimsel etki gözlenmemekle birlikte, Türkiye genelinde olduğu gibi 2022 yılında Eskişehir Şubesi özelinde de 200 TL'lik kupüre olan talebin artmaya başladığı izlenmektedir (Grafik 3.7.b).

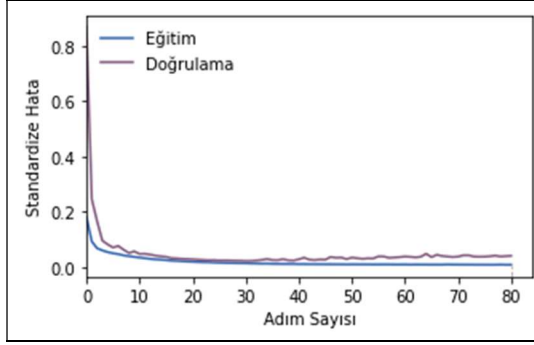
Eskişehir Şubesi 100 TL'lik kupür verilerine göre yapay sinir ağlarının mimarisi ve topolojisi ile SARIMA modeli dahil yapay sinir ağlarının performans metrikleri Tablo 3.8'de sunulmuştur.

TABLO 3.8. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 100 TL

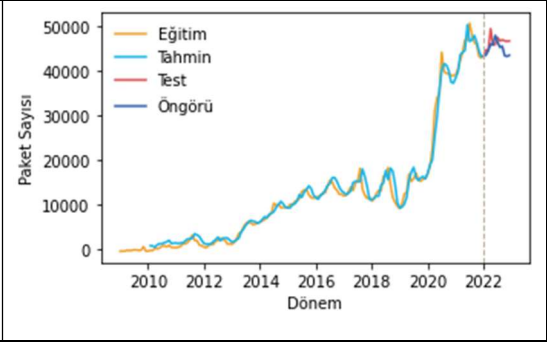
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	13	1, 2, 3, 4, 5, 11, 12, 13	7	
Gizli Nöronlar	8	8	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	195	80	137	
Eğitim RMSE (paket)	1.212	1.571	1.362	1.359
Test RMSE (paket)	1.488	2.150	2.130	1.906
Eğitim SMAPE	11,63	11,55	10,47	19,73
Test SMAPE	1,43	1,87	2,03	3,00

Eskişehir Şubesinin 100 TL'lik kupür verileriyle eğitilen üç yapay sinir ağı arasından FNN, SMAPE ve RMSE metriklerine göre öngörü/test dönemine ilişkin tahminlerde en iyi performansa sahip ağ olarak öne çıkmaktadır. SARIMA(0,1,2)x(1,0,0)₁₂ ise LSTM ağına yakın bir öngörü performansı göstermiştir (Tablo 3.8). SARIMA(0,1,2)x(1,0,0)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 8'de yer almaktadır. Grafik 3.8.a, b, c, d, e, f, g ve h'de Eskişehir Şubesi 100 TL'lik kupür verileri kullanılarak elde edilen, her üç ağın hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile SARIMA(0,1,2)x(1,0,0)₁₂ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.

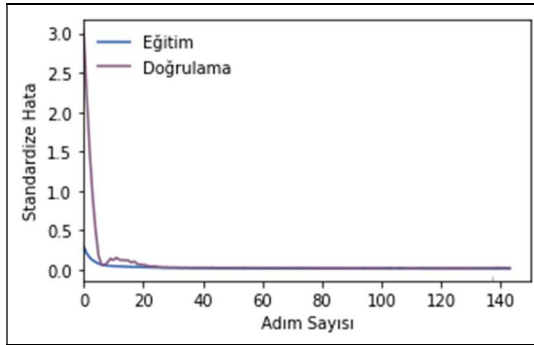
**Grafik 3.8.a. Eskişehir Şubesi: 100 TL için FNN Hata Optimizasyonu****Grafik 3.8.b. Eskişehir Şubesi: 100 TL için FNN Çıktısı**



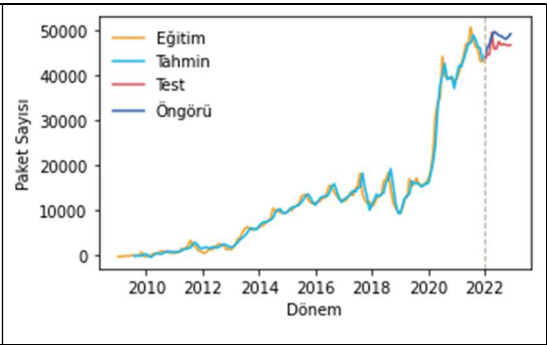
Grafik 3.8.c. Eskişehir Şubesi: 100 TL için TLNN Hata Optimizasyonu



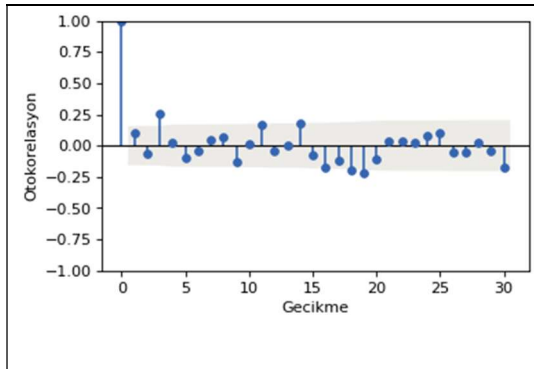
Grafik 3.8.d. Eskişehir Şubesi: 100 TL için TLNN Çıktısı



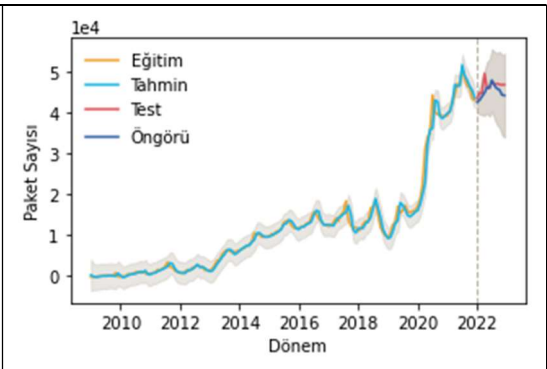
Grafik 3.8.e. Eskişehir Şubesi: 100 TL için LSTM Hata Optimizasyonu



Grafik 3.8.f. Eskişehir Şubesi: 100 TL için LSTM Çıktısı



Grafik 3.8.g. Eskişehir Şubesi: 100 TL için Hataların Otokorelasyonu



Grafik 3.8.h. Eskişehir Şubesi: 100 TL için SARIMA Çıktısı

Eskişehir Şubesinin 100 TL'lik kupür verileriyle yapılan tahminlerde, dört modelin de benzer örüntülere dayalı öngöründe bulunduğu görülmektedir (Grafik 3.8.b, d, f, h). SARIMA modelinin hata terimleri arasında sistematik bir otokorelasyon olmadığı anlaşılmaktadır (Grafik 3.8.g). SARIMA(0,1,2)x(1,0,0)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 8'de yer almaktadır. Bölgesel seviyede 100 TL'lik kupüre olan talep, Kurban Bayramı dönemlerinde zirve noktasına ulaşmaktadır. Takip eden birkaç ay sonrasında ise normal seyrine geri döndüğü, mevsimsel sebeplerle

talepte dalgalanmalar meydana geldiği gözlenmektedir. Talebin, 2020 yılına kadar dik olmayan bir trend etrafında doğrusal bir şekilde sürekli arttığı anlaşılmaktadır. Ancak 2020 yılında 100 TL'lik kupüre olan talebin trendin önemli ölçüde üzerine çıktığı ve bu seviyelerde tutunduğu anlaşılmaktadır (Grafik 3.8.b). Türkiye geneli, 2020 yılındaki 100 TL'lik kupüre olan talep artışının mevsimsel olduğu belirgin olmasına rağmen, Eskişehir Şubesi özelinde ise mevsimsel olmadığı söylenebilir.

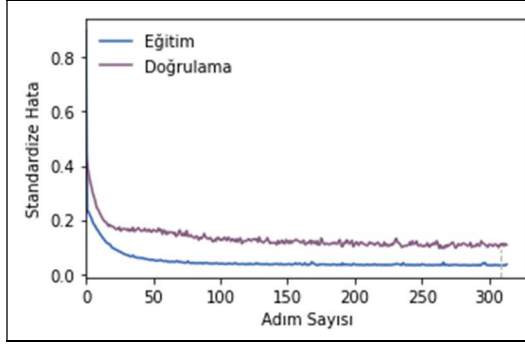
Eskişehir Şubesi 50 TL'lik kupür verilerine göre eğitilen ağlara ve SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.9'da sunulmuştur.

TABLO 3.9. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 50 TL

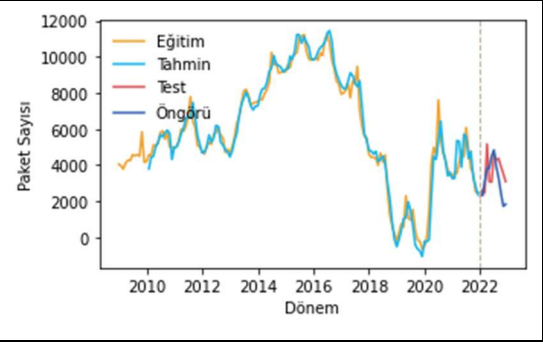
	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	13	1, 2, 3, 4, 11, 12, 13	9	
Gizli Nöronlar	7	7	6	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	309	126	123	
Eğitim RMSE (paket)	598	714	708	748
Test RMSE (paket)	1.054	678	810	862
Eğitim SMAPE	9,68	9,25	9,36	19,13
Test SMAPE	14,23	5,52	7,28	21,49

RMSE ve SMAPE metriklerine göre diğer iki ağla kıyaslandığında TLNN ağının eğitim performansı görece zayıf olmasına rağmen, öngörü/test performansının daha başarılı olduğu görülmektedir. SARIMA(1,1,0)x(0,1,1)₁₂ modeli ise eğitim ve test verisine ilişkin RMSE metriklerine göre LSTM ağı ile yakın bir performans göstermiştir (Tablo 3.9).

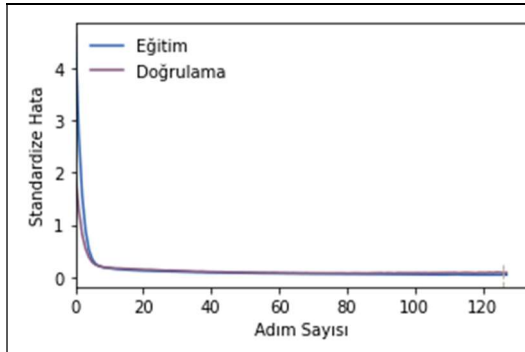
Grafik 3.9.a, b, c, d, e, f, g ve h'de Eskişehir Şubesi 50 TL'lik kupür verilerine ilişkin hata optimizasyonu süreci ile eğitim ve öngörü sonuçları ile SARIMA(1,1,0)x(0,1,1)₁₂ modelinin hata terimlerinin otokorelasyonu ve öngörü sonuçları görselleştirilmiştir.



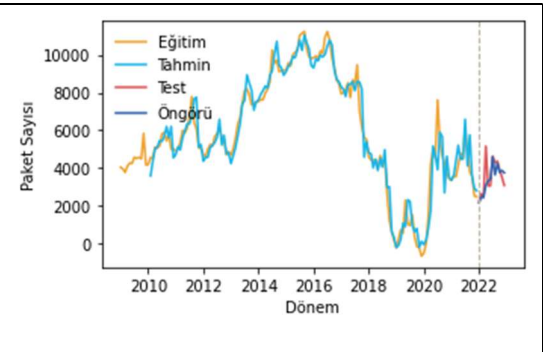
Grafik 3.9.a. Eskişehir Şubesi: 50 TL için FNN Hata Optimizasyonu



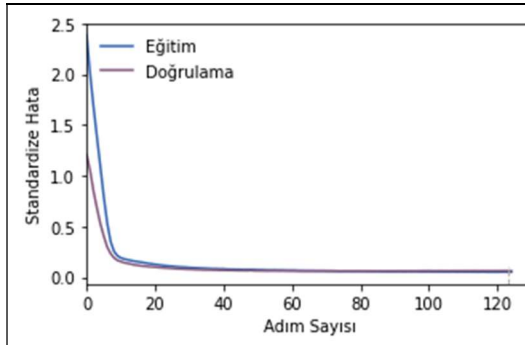
Grafik 3.9.b. Eskişehir Şubesi: 50 TL için FNN Çıktısı



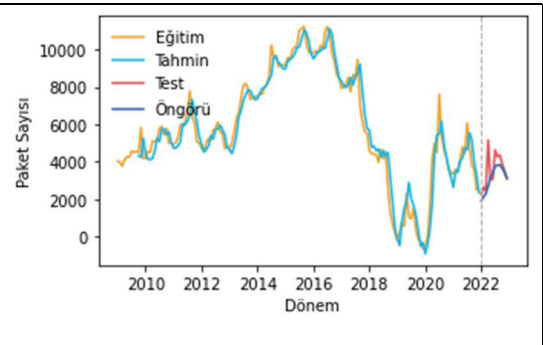
Grafik 3.9.c. Eskişehir Şubesi: 50 TL için TLNN Hata Optimizasyonu



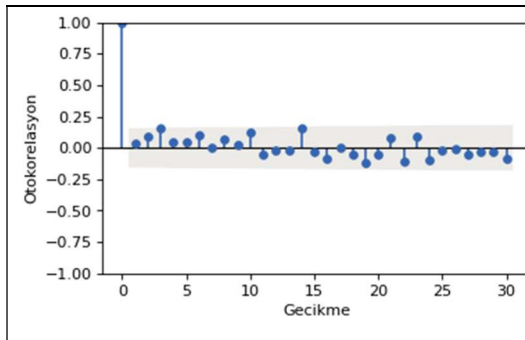
Grafik 3.9.d. Eskişehir Şubesi: 50 TL için TLNN Çıktısı



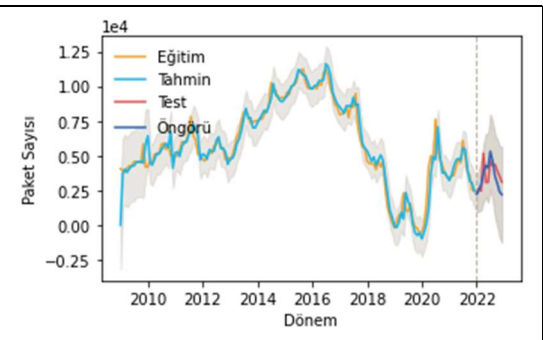
Grafik 3.9.e. Eskişehir Şubesi: 50 TL için LSTM Hata Optimizasyonu



Grafik 3.9.f. Eskişehir Şubesi: 50 TL için LSTM Çıktısı



Grafik 3.9.g. Eskişehir Şubesi: 50 TL için Hataların Otokorelasyonu



Grafik 3.9.h. Eskişehir Şubesi: 50 TL için SARIMA Çıktısı

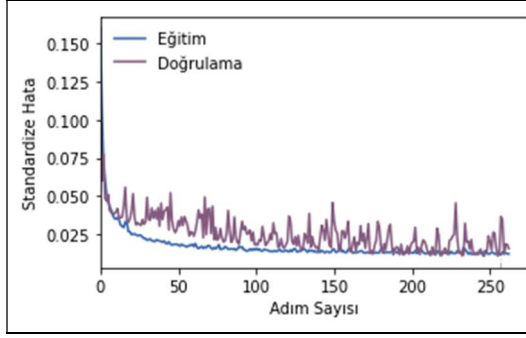
Tüm modellerin öngörü/test dönemi tahminlerinin birbirine benzer olduğu anlaşılmaktadır (Grafik 3.9.b, d, f, h). Bu veri seti için SARIMA(1,1,0)x(0,1,1)₁₂ modelinin öngörü/test döneminde yapay sinir ağlarına kıyasla mevsimsel hareketlere daha hızlı tepki vermesi dikkat çekmektedir. SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin tüm istatistikler Ek-1’de sunulan Tablo 9’da yer almaktadır. Eskişehir Şubesi özelinde 50 TL’lik kupüre olan talep, Türkiye genelinde olduğu gibi 100 TL’lik kupüre olan talep ile benzer bir yapıya sahip değildir. Bölgesel seviyede 50 TL’lik kupüre olan talep, belirgin bir trende sahip olmamakla birlikte Kurban Bayramı dönemlerinde artmaktadır. Ancak 2020 yılında 50 TL’lik kupürde önemli bir talep artışı olmasına rağmen, 2016 yılından sonra talebin genel olarak azalma eğiliminde olduğu ifade edilebilir (Grafik 3.9.b).

Eskişehir Şubesi 20 TL’lik kupür verilerine göre eğitilen ağlara ve SARIMA(1,1,0)x(0,1,1)₁₂ modeline ilişkin bilgiler Tablo 3.10’da sunulmuştur.

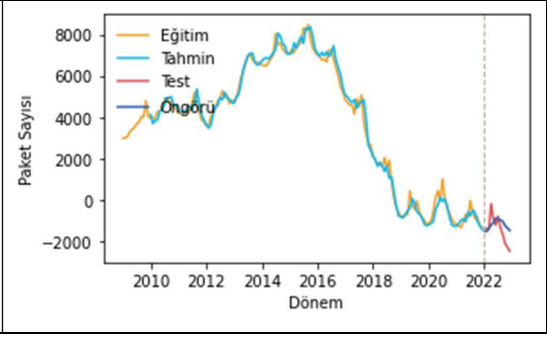
TABLO 3.10. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 20 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12, 13	9	
Gizli Nöronlar	9	6	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	257	295	106	
Eğitim RMSE (paket)	337	335	433	440
Test RMSE (paket)	599	495	521	606
Eğitim SMAPE	10,09	10,65	13,20	2,55
Test SMAPE	19,04	16,41	17,29	8,26

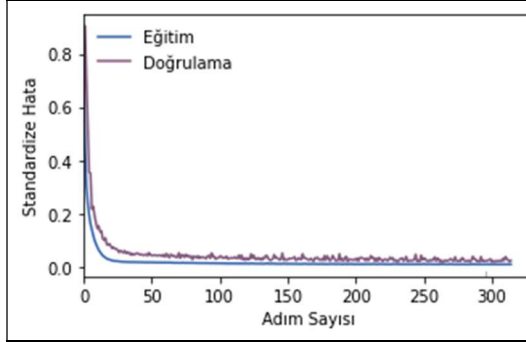
RMSE ve SMAPE metriklerine göre genel olarak tüm modellerin öngörü performansı birbirine yakın olmasına rağmen, TLNN ağının öngörü performansının daha iyi olduğu görülmektedir. SARIMA(1,1,0)x(0,1,1)₁₂ modelinin ise diğer üç model arasında en başarısız öngörüye sahip model olduğu söylenebilir (Tablo 3.10). Grafik 3.10.a, b, c, d, e, f, g ve h’de Eskişehir Şubesi 20 TL’lik kupür verilerine ilişkin ağ çıktılarının ve SARIMA(1,1,0)x(0,1,1)₁₂ modelinin görsellerine yer verilmiştir.



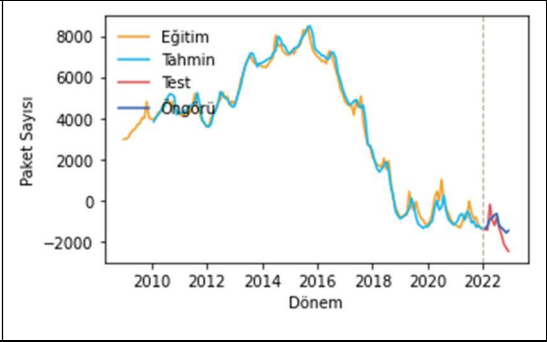
Grafik 3.10.a. Eskişehir Şubesi: 20 TL için FNN Hata Optimizasyonu



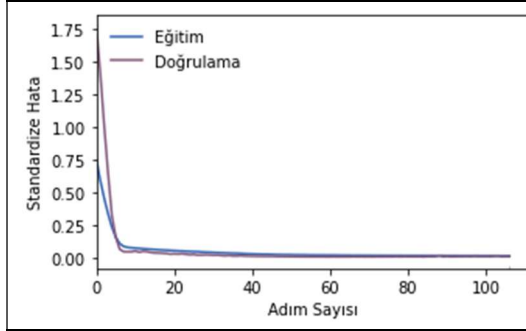
Grafik 3.10.b. Eskişehir Şubesi: 20 TL için FNN Çıktısı



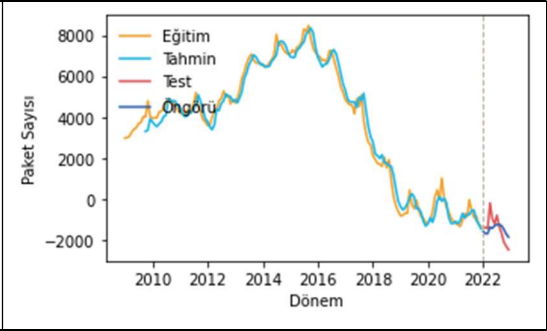
Grafik 3.10.c. Eskişehir Şubesi: 20 TL için TLNN Hata Optimizasyonu



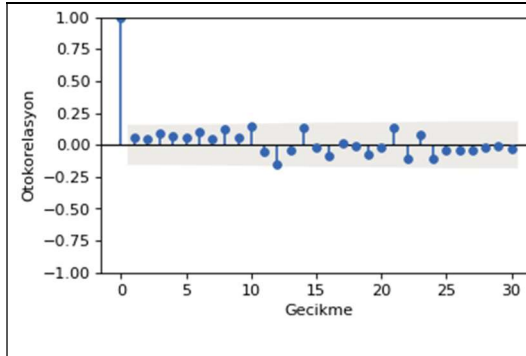
Grafik 3.10.d. Eskişehir Şubesi: 20 TL için TLNN Çıktısı



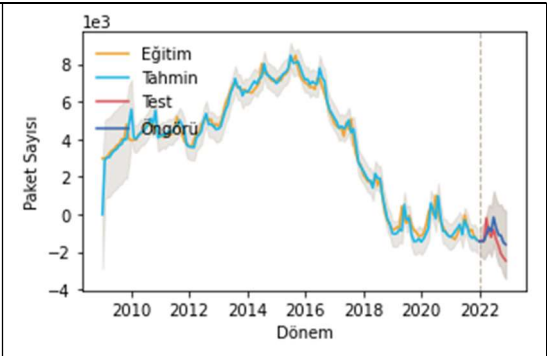
Grafik 3.10.e. Eskişehir Şubesi: 20 TL için LSTM Hata Optimizasyonu



Grafik 3.10.f. Eskişehir Şubesi: 20 TL için LSTM Çıktısı



Grafik 3.10.g. Eskişehir Şubesi: 20 TL için Hataların Otokorelasyonu



Grafik 3.10.h. Eskişehir Şubesi: 20 TL için SARIMA Çıktısı

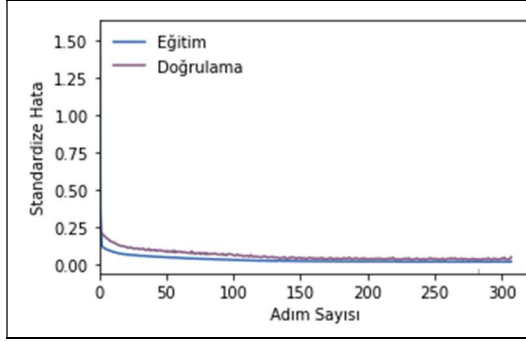
Tüm modellerin öngörü/test dönemi tahminleri birbirine benzer olmakla birlikte, $SARIMA(1,1,0) \times (0,1,1)_{12}$ modelinin öngörü/test döneminde mevsimsel hareketleri takip etmede diğer modellerden daha başarılı olduğu söylenebilir (Grafik 3.10.b, d, f, h). Hata terimleri arasında sistematik bir otokorelasyonun bulunmadığı $SARIMA(1,1,0) \times (0,1,1)_{12}$ modeline ilişkin tüm istatistikler Ek-1’de sunulan Tablo 10’da yer almaktadır. Bölgesel seviyede 20 TL’lik kupüre olan talep, belirgin bir trende sahip olmamakla birlikte Kurban Bayramı dönemlerinde mevsimsel dalgalanmalara sahiptir. 2016 yılından sonra talebin azalma eğiliminde olduğu görülmektedir. 2018 yılından sonra ise sıfırın etrafında dalgalanan ve zaman zaman sıfırın altına düşen bir talep söz konusudur. 2021 yılından itibaren ise 20 TL’lik kupüre olan bölgesel talep negatif seviyelerdedir (Grafik 3.10.b). Söz konusu durum diğer bölgelerden tedavüle sürülen 20 TL’lik banknotların Eskişehir Şubesi tarafından tedavülden çekildiğine işaret etmektedir.

Eskişehir Şubesi 10 TL’lik kupür verilerine göre eğitilen ağlara ve $SARIMA(1,1,0) \times (1,0,0)_{12}$ modeline ilişkin bilgiler Tablo 3.11’de sunulmuştur.

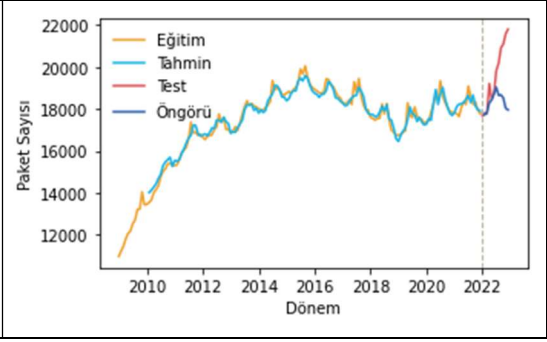
TABLO 3.11. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 10 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	13	1, 2, 3, 11, 12, 13	9	
Gizli Nöronlar	5	5	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	282	213	168	
Eğitim RMSE (paket)	256	282	285	927
Test RMSE (paket)	1.888	1.803	1.782	2.123
Eğitim SMAPE	0,57	0,57	0,58	2,55
Test SMAPE	3,35	3,35	3,32	8,26

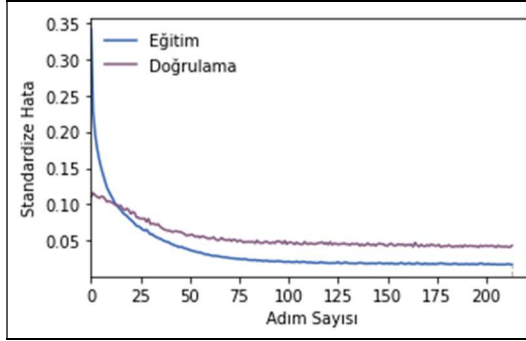
RMSE ve SMAPE metriklerine göre tüm modellerin büyük ölçüde benzer performans gösterdiği, öngörü döneminde nispeten büyük sapmalara sahip olduğu anlaşılmaktadır (Tablo 3.11). Grafik 3.11.a, b, c, d, e, f, g ve h’de Eskişehir Şubesi 10 TL’lik kupür verilerine ilişkin ağ çıktılarının ve $SARIMA(1,1,0) \times (1,0,0)_{12}$ modelinin görsellerine yer verilmiştir.



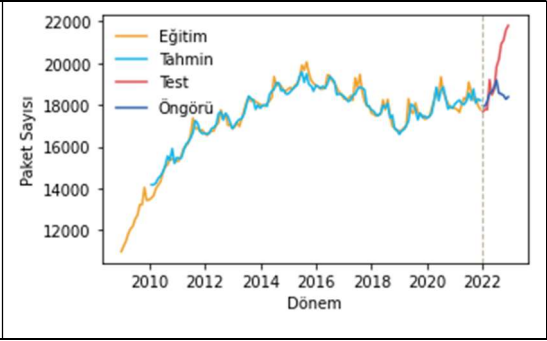
Grafik 3.11.a. Eskişehir Şubesi: 10 TL için FNN Hata Optimizasyonu



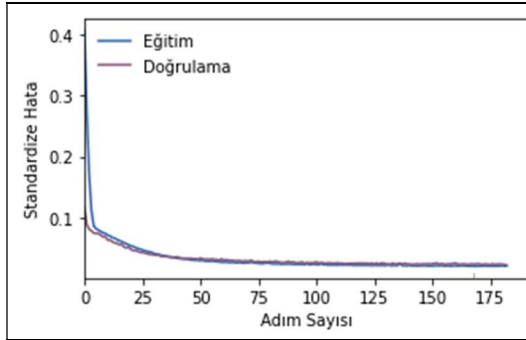
Grafik 3.11.b. Eskişehir Şubesi: 10 TL için FNN Çıktısı



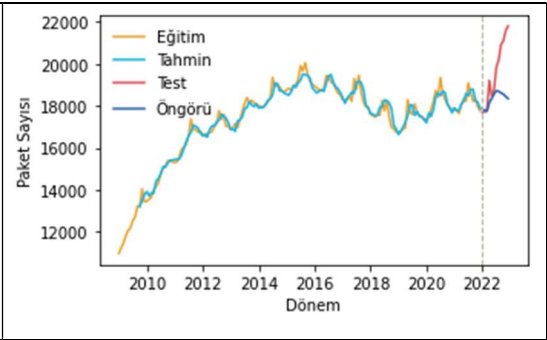
Grafik 3.11.c. Eskişehir Şubesi: 10 TL için TLNN Hata Optimizasyonu



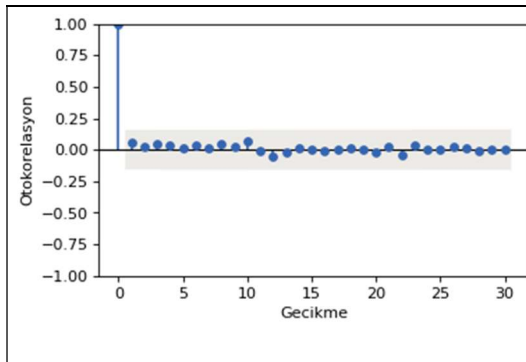
Grafik 3.11.d. Eskişehir Şubesi: 10 TL için TLNN Çıktısı



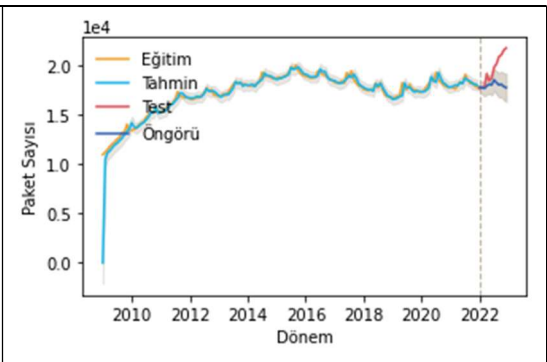
Grafik 3.11.e. Eskişehir Şubesi: 10 TL için LSTM Hata Optimizasyonu



Grafik 3.11.f. Eskişehir Şubesi: 10 TL için LSTM Çıktısı



Grafik 3.11.g. Eskişehir Şubesi: 10 TL için Hataların Otokorelasyonu



Grafik 3.11.h. Eskişehir Şubesi: 10 TL için SARIMA Çıktısı

Eskişehir Şubesinin 10 TL'lik verilerinde önceki dönemlerin örüntüsüne uymayan yukarı yönlü sert hareket hiçbir model tarafından öngörülemediği (Grafik 3.11.b, d, f, h).

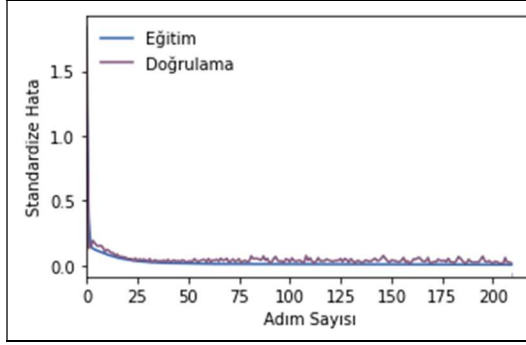
Diğer kupürlerde olduğu gibi 10 TL'lik kupüre olan talep de Kurban Bayramı dönemlerinden kaynaklanan mevsimsel dalgalanmalar belirgindir. 2016 yılından itibaren 10 TL'lik kupüre olan talep, nispeten yatay bir seyre sahiptir. Ancak 2022 yılından itibaren, Türkiye genelinde olduğu gibi Eskişehir Şubesi özelinde de talebin belirgin bir şekilde arttığı gözlenmektedir. (Grafik 3.11.b).

Eskişehir Şubesi 5 TL'lik kupür verilerine göre eğitilen ağlara ve SARIMA(3,2,0)x(0,1,0)₁₂ modeline ilişkin bilgiler Tablo 3.12'de sunulmuştur.

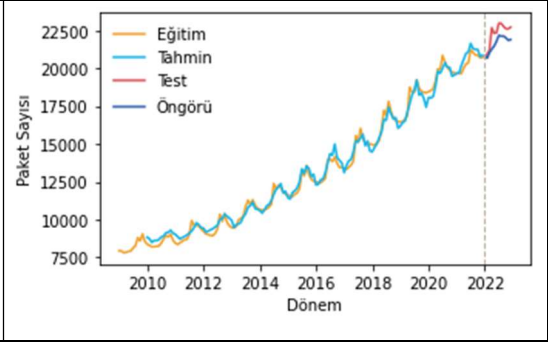
TABLO 3.12. ESKİŞEHİR ŞUBESİ İÇİN TASARLANAN MODELLERİN BİLGİLERİ: 5 TL

	FNN	TLNN	LSTM	SARIMA
Katman Sayısı	2	2	2	
Girdi Nöronları	12	1, 2, 3, 11, 12, 13	8	
Gizli Nöronlar	6	6	5	
Çıktı Nöronu	1	1	1	
Aktivasyon Fonksiyonu	selu	selu	selu	
Adım Sayısı	209	120	97	
Eğitim RMSE (paket)	355	290	450	892
Test RMSE (paket)	736	935	530	530
Eğitim SMAPE	1,15	0,85	1,34	4,11
Test SMAPE	1,46	1,85	0,89	1,78

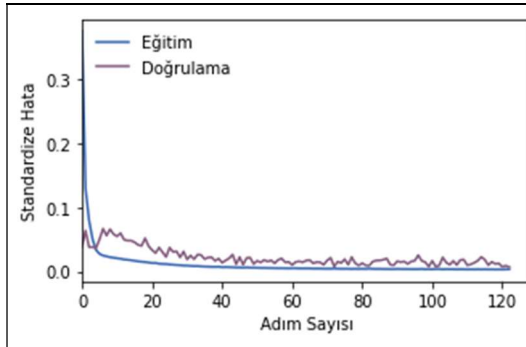
RMSE ve SMAPE metriklerine göre LSTM ağının ve SARIMA(3,2,0)x(0,1,0)₁₂ modelinin eğitim performansı görece zayıf olmasına rağmen, öngörü/test dönemi performansı diğer iki yapay sinir ağından daha başarılıdır (Tablo 3.12). SARIMA(3,2,0)x(0,1,0)₁₂ modeline ilişkin tüm istatistikler Ek-1'de sunulan Tablo 12'de yer almaktadır. Grafik 3.12.a, b, c, d, e, f, g ve h'de Eskişehir Şubesi 5 TL'lik kupür verilerine ilişkin ağ çıktılarının ve SARIMA(3,2,0)x(0,1,0)₁₂ modelinin görsellerine yer verilmiştir.



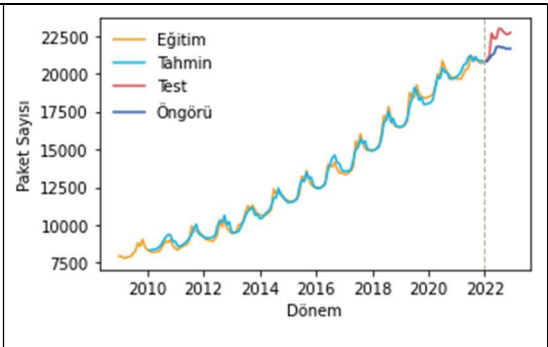
Grafik 3.12.a. Eskişehir Şubesi: 5 TL İçin FNN Hata Optimizasyonu



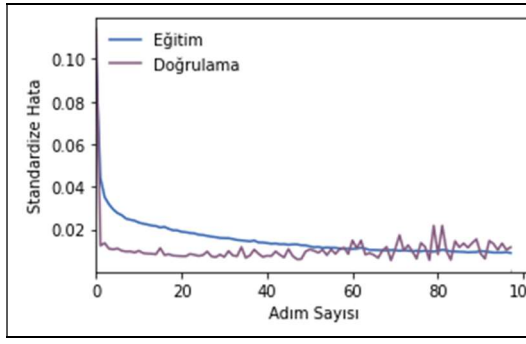
Grafik 3.12.b. Eskişehir Şubesi: 5 TL İçin FNN Çıktısı



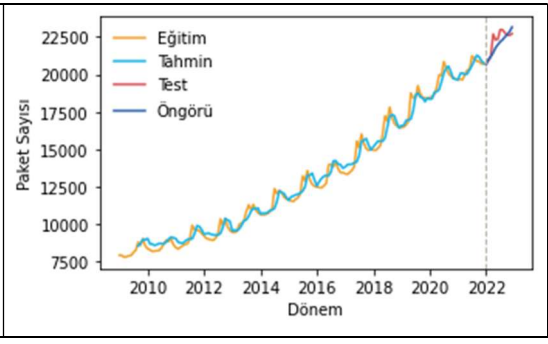
Grafik 3.12.c. Eskişehir Şubesi: 5 TL İçin TLNN Hata Optimizasyonu



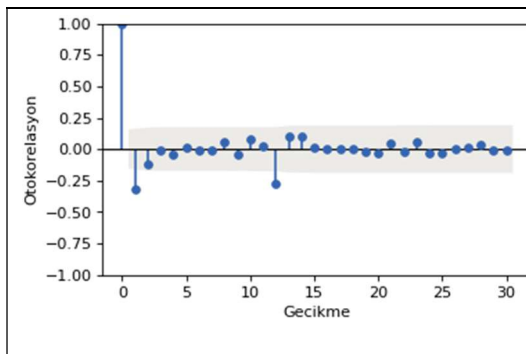
Grafik 3.12.d. Eskişehir Şubesi: 5 TL İçin TLNN Çıktısı



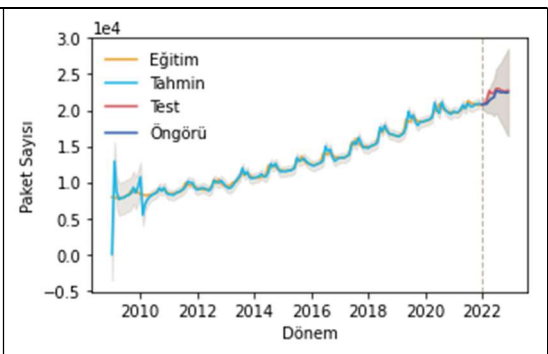
Grafik 3.12.e. Eskişehir Şubesi: 5 TL İçin LSTM Hata Optimizasyonu



Grafik 3.12.f. Eskişehir Şubesi: 5 TL İçin LSTM Çıktısı



Grafik 3.12.g. Eskişehir Şubesi: 5 TL İçin Hataların Otokorelasyonu



Grafik 3.12.h. Eskişehir Şubesi: 5 TL İçin SARIMA Çıktısı

LSTM ağıının ve SARIMA(3,2,0)x(0,1,0)₁₂ modelinin RMSE metriğine göre öngörü performansları aynı olmasına rağmen, Grafik 3.12.b, d, f, h incelendiğinde, SARIMA(3,2,0)x(0,1,0)₁₂ modelinin öngörü dönemine ilişkin mevsimsel hareketleri takip etmede daha başarılı olduğu ifade edilebilir.

Türkiye genelinde olduğu gibi Eskişehir Şubesi özelinde de 5 TL'lik kupüre olan talep Kurban Bayramı dönemlerinden kaynaklanan mevsimsel dalgalanmalarla son derece düzenli bir yapıya sahiptir. Trendde herhangi belirgin bir değişiklik gözlenmemektedir (Grafik 3.12.b).

3.3. Modellerin Değerlendirilmesi

Üzerinde çalışılan veri seti çerçevesinde ise FNN, TLNN ve LSTM yapay sinir ağları ile SARIMA modeli arasında birinin diğerine üstünlüğünden söz etmek mümkün görünmemektedir. Çeşitliliği yüksek birbirinden farklı veri setleri üzerinde yapılan araştırmada, veri setinin ya da problemin yapısına göre FNN, TLNN, LSTM ve SARIMA modelleri arasında öngörüye ilişkin performans farklılıkları gözlenmiştir. Bu sonuç da tüm problemlere uygulanacak tek bir iyi model olmadığına işaret etmektedir. Bir problemin çözümüne ilişkin en iyi modelin deneysel yöntemlerle derinlemesine yapılan araştırmalarla tespit edilebileceği söylenebilir.

Tablo 3.13'ten anlaşıldığı üzere test verisine ilişkin RMSE metriğine göre Türkiye geneli ve Eskişehir Şubesi düzeyinde, yapay sinir ağları ve SARIMA modelinin kupür bazında öngörü performansları arasında farklılıklar vardır. Türkiye geneli için öngörülerde 6 kupürün 4'ünde LSTM, SARIMA modeli ve diğer ağlardan daha iyi performans göstermiştir. 100 TL'lik ve 200 TL'lik kupürlerde, sırasıyla, FNN ve SARIMA modelinin başarımı LSTM ağıının bir miktar üzerindedir. Eskişehir Şubesi için yapılan öngörülerde, SARIMA modeli bazı kupürlerde yapay sinir ağlarına yakın bir öngörü performansına sahip olsa da FNN, LSTM ve TLNN ağları tarafından yapılan tahminlerin daha az hataya sahip olduğu görülmüştür.

TABLO 3.13. MODELLERİN RMSE METRİĞİNE GÖRE KUPÜR BAZINDA ÖNGÖRÜ PERFORMANSLARI

	FNN (Test RMSE, paket)	TLNN (Test RMSE, paket)	LSTM (Test RMSE, paket)	SARIMA (Test RMSE, paket)
200 TL (Türkiye)	73.188	106.747	70.721	69.491
100 TL (Türkiye)	45.462	54.632	49.430	60.112
50 TL (Türkiye)	23.598	26.489	21.754	22.056
20 TL (Türkiye)	26.612	28.123	24.402	35.062
10 TL (Türkiye)	57.871	48.915	20.989	61.663
5 TL (Türkiye)	10.880	8.202	7.578	13.147
En İyi Skor Toplamı (TR)	1	0	4	1
200 TL (Eskişehir)	1.437	1.560	1.900	13.762
100 TL (Eskişehir)	1.488	2.150	2.130	1.906
50 TL (Eskişehir)	1.054	678	810	862
20 TL (Eskişehir)	599	495	521	606
10 TL (Eskişehir)	1.888	1.803	1.782	2.123
5 TL (Eskişehir)	736	935	530	530
En İyi Skor Toplamı (Esk)	2	2	2	1
En İyi Skor Toplamı (TR+Esk)	3	2	6	2

Trendle aynı ya da ters yönlü kırılmaları yakalamada (Şekil 3.1, 3.4, 3.5, 3.7 ve 3.11) yapay sinir ağlarının SARIMA modelinden daha başarılı olduğu ifade edilebilir. Özellikle de LSTM ağı, söz konusu kırılmaları öngörmede, SARIMA modeli ve diğer yapay sinir ağlarına göre nispeten daha başarılıdır. Ancak RMSE metrikleri yansıtmamasına rağmen, Şekil 3.3, 3.9, 3.10 ve 3.12’de yer alan görsellerden, SARIMA modelinin yapay sinir ağlarına kıyasla mevsimsel hareketleri daha başarılı bir şekilde takip ettiği söylenebilir.

DÖRDÜNCÜ BÖLÜM

SONUÇ VE ÖNERİLER

Temel görevleri parasal ve finansal istikrarı sürdürmek olan merkez bankaları, bu görevlerini yerine getirmek için para politikası araçlarını kullanarak para piyasası likiditesini düzenlemeye çalışır. Para piyasası likiditesi ise içinde önemli bir otonom faktör olan ve makroekonomik değişkenlerin dışında birçok dışsal faktörden etkilenen emisyon hacmini (dolaşımdaki para miktarını) barındırır. Bu durum ise para piyasası likiditesini öngörmeyi zorlaştırır. Diğer taraftan banknotların üretimi, dağıtımı gibi kurum içi operasyonel süreçlerin etkin bir şekilde yönetilebilmesi için gelecek dönemlere ilişkin dolaşımdaki para miktarını en az hata ile tahmin etmek isterler. Bu motivasyonla FNN, TLNN, LSTM yapay sinir ağları ile zaman serisi modeli olan SARIMA yardımıyla, Türkiye geneli ve Eskişehir Şubesi Reeskont Bölgesi düzeyinde kupür bazında emisyon hacminin öngörülmesine ilişkin Python ile kapsamlı bir deneysel uygulama gerçekleştirilmiştir. Araştırmancının bulgularına göre her duruma uygun en iyi tek bir model olmadığı sonucuna varılmıştır.

Yapay sinir ağlarının tasarım ve eğitim süreci hem uygulama tarafından gerçekleştirilen hesaplama hem de gerektirdiği insan emeği bakımından zaman serilerine göre daha fazla maliyete sahiptir. Ancak öğrenme süreci sonunda ağırlık matrisinin elde edilmesiyle, öngörüler hızlı bir şekilde yapılabilir. Bununla birlikte yapay sinir ağlarının en önemli avantajlarından biri, regresyon analizine dayalı zaman serileri modellerinde olduğu gibi hata terimlerinde otokorelasyonun ya da değişen varyans probleminin bulunmaması gibi bir dizi istatistiksel varsayıma gereksinim duymamasıdır. Ancak zaman serileri modelleri istatistiksel varsayımların geçerliliğine bağımlı olduğundan, zaman serileri analizine dayalı bir model tasarlamak ve geliştirmek, genellikle, yapay sinir ağlarına kıyasla daha yüksek bir matematiksel karmaşıklık seviyesine sahiptir. Ayrıca yapay sinir ağları,

sadece doğrusal değil, doğrusal olmayan ilişkileri de ortaya çıkartabildiği için bazı kupürlerde (sadece doğrusal ilişkileri ortaya çıkarabilen) SARIMA modelinden çok daha isabetli öngörülere sahiptir. Bu iki özellik yapay sinir ağlarını esnekleştirerek gerçek dünya problemlerine daha uygulanabilir kılmaktadır.

Model öngörülerinin gerçek dünya karar süreçlerinde kullanılmak istenmesi durumunda, belirli bir veri setine ilişkin tüm modellerin çıktılarının (benzerlikleri ve farklılıkları da göz önünde tutularak) birlikte değerlendirilmesi, model çıktılarının doğrudan uygulanması yerine modellerin karar verici için bir tür karar destek sistemi olarak görülmesi önerilir.

Diğer taraftan ülkesel ve bölgesel düzeyde emisyon hacimlerinde, Ramazan ve Kurban Bayramı dönemlerinde önemli dalgalanmalar meydana gelmektedir. Öngörülerde oluşan hataların büyük kısmının söz konusu dönemlerden kaynaklandığı düşünülmektedir. Çünkü hicri takvime göre belirlenen Ramazan ve Kurban Bayramı dönemleri, miladi takvime göre her yıl yaklaşık 11 gün geriye gelmektedir. Söz konusu durum ise yaklaşık 6 yılda bir Ramazan ve Kurban Bayramı dönemlerinin bir ay geriye gelmesi ile sonuçlanmaktadır. Bu çerçevede, söz konusu takvim farkı göz önüne alınarak emisyon hacmi verileri düzeltilebilir. Böylece veri seti tarafında bahsi geçen düzeltmenin yapay sinir ağları ve zaman serisi modellerinin öngörü performansı üzerindeki etkisi başka bir çalışmanın konusu olabilir. Ayrıca SARIMA modelinin hata terimlerinde değişen varyansın varlığı, hata terimlerinin varyansına dayalı olarak geliştirilen ARCH ve GARCH modellerinin de başarılı sonuçlar verebileceğine işaret etmektedir. Aynı veri setleri üzerinde SARIMA yerine ARCH ve GARCH modellerinin kullanılması, yapay sinir ağlarıyla öngörü performanslarının kıyaslanması ayrı bir araştırma konusu olabilir.

KAYNAKÇA

- Ağaslan, E. ve Gayaker, S. (Eylül 2020). Türkiye Ekonomisindeki Emisyon Hacminin Düşük ve Yüksek Frekanslı Modeller ile Öngörüsü. *Bankacılar Dergisi*, 114, 30-49.
- Aksu, G., Güzeller, O. C. ve Eser, M. T. (Mart 2019). The Effect of the Normalization Method Used in Different Sample Sizes on the Success of Artificial Neural Network Model. *International Journal of Assessment Tools in Education*, 6, 2, 170-192.
- Alharbi, F. R. ve Csala, D. (Ekim 2022). A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach. *Inventions*, 7, 94, 1-21.
- Anderson, D. ve McNeill, G. (1992). Artificial Neural Networks Technology. New York: Kaman Sciences Corporation.
- Artificial Intelligence: Intelligent Systems. (2015). Tutorials Point.
- Assenmacher, K., Seitz, F. ve Tenhofen, J. (Ağustos 2019). The Demand for Swiss Banknotes: Some New Evidence. *Swiss Journal of Economics and Statistics*, 155,14, 1-22.
- Balli, F. ve Elsamadisy, E. M. (Mart 2010). Modeling the Currency in Circulation for the State of Qatar. *International Journal of Islamic and Middle Eastern Finance and Management*, 5, 4, 321-339.
- Bartzsch, N., Seitz, F. ve Setzer, R. (Haziran 2015). The Demand for Euro Banknotes in Germany: Structural Modelling and Forecasting. *Munich Personal RePEc Archive*, 64949, 1-50.
- Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C. ve Ljung, G. M. (2016). Time Series Analysis: Forecasting and Control. (Beşinci basım). New Jersey: John Wiley and Sons.
- Brooks, C. (2008). Introductory Econometrics for Finance. (İkinci basım). New York: Cambridge University Press.
- Chatfield, C. (1996). The Analysis of Time Series an Introduction. (Beşinci basım). London: Chapman and Hal/Crc.
- Enders, W. (2004). Applied Econometric Time Series. (İkinci basım). New Jersey: John Wiley and Sons.

- Evans, M. K. (2003). Practical Business Forecasting. Cornwall: Blackwell Publishers.
- Fausett, L. V. (1994). Fundamentals of Neural Networks: Architectures, Algorithms and Applications. New Jersey: Pearson Education.
- Freeman, J. A. ve Skapura, D. M. (1991). Neural Networks: Algorithms, Applications, and Programming Techniques. USA: Addison-Wesley Publishing Company.
- Gujarati, D. N. ve Porter, D. C. (2009). Basic Econometrics. (Beşinci basım). New York: McGraw-Hill Companies.
- Gurney, K. (1997). An Introduction to Neural Networks. London: UCL Press.
- Hagan, M. T., Demuth, H. B., Beale, M. H. ve Jesus, O. D. (2014). Neural Network Design. (İkinci basım). Oklahoma: Stillwater.
- Hanke, J. E. ve Wichern, D. (2014). Business Forecasting. (Dokuzuncu basım). London: Pearson Education.
- Haykin, S. (2009). Neural Networks and Learning Machines. (Üçüncü basım). New Jersey: Pearson Education.
- Hill, R. C., Griffiths, W. E. ve Lim, G. C. (2011). Principles of Econometrics. (Dördüncü basım). New Jersey: John Wiley and Sons.
- Hyndman, R. J. (2014). Forecasting: Principles and Practice. Erişim: 8 Nisan 2023, <https://robjhyndman.com/uwafiles/fpp-notes.pdf>.
- Hyndman, R. J. ve Koehler, A. B. (Kasım 2006). Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22, 4, 679-688.
- Kröse, B. ve Smagt, P. V. D. (1996). An Introduction to Neural Networks. (Sekizinci basım). Amsterdam: The University of Amsterdam.
- Ljung, G. M. ve Box, G. E. P. (1978). On a Measure of Lack of Fit in Time Series Models. *Biometrika*, 65, 2, 297-303.
- Makridakis, S., Wheelwright, S. C. ve Hyndman, R. J. (1998). Forecasting Methods and Applications. (Üçüncü basım). New York: John Wiley ve Sons.
- McCulloch, W. S. ve Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, 5, 115-133.
- Mehrotra, K., Mohan, C. K. ve Ranka, S. (1996). Elements of Artificial Neural Networks. Massachusetts: The MIT Press.

- Miller, C. (25-27 Nisan 2017). Addressing the Limitations of Forecasting Banknote Demand. *International Cash Conference*, Deutsche Bundesbank, Frankfurt.
- Mills, T. C. (2019). Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting. London: Elsevier.
- Montgomery, D. C., Jennings, C. L. ve Kulahci, M. (2008). Introduction to Time Series Analysis and Forecasting. New Jersey: John Wiley and Sons.
- Nielsen, M. A. (2015). Neural Networks and Deep Learning. Eriřim: 8 Nisan 2023, <https://freecomputerbooks.com/Neural-Networks-and-Deep-Learning.html>.
- Nwankpa, C. E., Ijomah, W., Gachagan, A. ve Marshall, S. (Kasım 2018). Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. Eriřim: 8 Nisan 2023, <https://arxiv.org/abs/1811.03378>.
- Öztemel, E. (2006). Yapay Sinir Ağları. (İkinci basım). İstanbul: Papatya Yayıncılık Eğitim.
- Pierce, D. A. (Temmuz 1970). Distribution of Residual Autocorrelations in the Regression Model with Autoregressive-Moving Average Errors. *Journal of the Royal Statistical Society*, 33, 1, 140-146.
- Pindyck, R. S. ve Rubinfeld, D. L. (1991). Econometric Models and Economic Forecasts. (Üçüncü basım). New York: McGraw-Hill.
- Putri, J. A., Suhartono, Prabowo, H. ve Salehah, N. A. (Ocak 2021). Forecasting Currency in East Java: Classical Time Series vs. Machine Learning. *Indonesian Journal of Statistics and Its Applications*, 5, 2, 284-303.
- Schwert, G. W. (Nisan 1989). Tests for Unit Roots: A Monte Carlo Investigation. *Journal of Business and Economic Statistics*, 7, 2, 147-159.
- Shwartz, S. S. ve David, S. B. (2014). Understanding Machine Learning: From Theory to Algorithms. New York: Cambridge University Press.
- Silva, I. N. D., Spatti, D. H., Flauzino, R. A. ve Liboni, L. H. B. (2017). Artificial Neural Networks: A Practical Course. Switzerland: Springer International Publishing.
- Stock, J. H. ve Watson, M. W. (2020). Introduction to Econometrics. (Dördüncü basım). London: Pearson Education.
- Wagner, M. (Temmuz 2010). Forecasting Daily Demand in Cash Supply Chains. *American Journal of Economics and Business Administration*, 2, 4, 377-383.

EKLER

Ek-1: Türkiye Geneli ve Eskişehir Şubesi İçin SARIMA İstatistikleri

TABLO 1. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 200 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(2, 2, 0)x(0, 1, [1], 12)	Log Likelihood	-1543.036			
Date:	Sun, 09 Jul 2023	AIC	3094.072			
Time:	14:17:14	BIC	3105.895			
Sample:	01-01-2009	HQIC	3098.877			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.7231	0.062	-11.590	0.000	-0.845	-0.601
ar.L2	-0.4807	0.080	-6.015	0.000	-0.637	-0.324
ma.S.L12	-0.5935	0.107	-5.560	0.000	-0.803	-0.384
sigma2	1.835e+08	2.1e-10	8.72e+17	0.000	1.83e+08	1.83e+08
=====						
Ljung-Box (L1) (Q):	0.06	Jarque-Bera (JB):	390.47			
Prob(Q):	0.80	Prob(JB):	0.00			
Heteroskedasticity (H):	42.29	Skew:	-0.92			
Prob(H) (two-sided):	0.00	Kurtosis:	10.91			
=====						

TABLO 2. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 100 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 0, 0)x(0, 1, [1], 12)	Log Likelihood	-1729.974			
Date:	Sun, 09 Jul 2023	AIC	3467.948			
Time:	14:17:44	BIC	3479.827			
Sample:	01-01-2009	HQIC	3472.775			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

intercept	9579.4937	3776.145	2.537	0.011	2178.385	1.7e+04
ar.L1	0.8780	0.030	29.554	0.000	0.820	0.936
ma.S.L12	-0.6719	0.087	-7.692	0.000	-0.843	-0.501
sigma2	2.03e+09	0.036	5.67e+10	0.000	2.03e+09	2.03e+09
=====						
Ljung-Box (L1) (Q):	1.46	Jarque-Bera (JB):	805.13			
Prob(Q):	0.23	Prob(JB):	0.00			
Heteroskedasticity (H):	23.92	Skew:	1.37			
Prob(H) (two-sided):	0.00	Kurtosis:	14.26			
=====						

TABLO 3. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 50 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)	Log Likelihood	-1620.735			
Date:	Sun, 09 Jul 2023	AIC	3247.470			
Time:	14:18:14	BIC	3256.358			
Sample:	01-01-2009	HQIC	3251.081			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.3407	0.063	-5.411	0.000	-0.464	-0.217
ma.S.L12	-0.6585	0.097	-6.769	0.000	-0.849	-0.468
sigma2	5.057e+08	9.79e-11	5.16e+18	0.000	5.06e+08	5.06e+08
=====						
Ljung-Box (L1) (Q):	0.18	Jarque-Bera (JB):	142.69			
Prob(Q):	0.67	Prob(JB):	0.00			
Heteroskedasticity (H):	3.69	Skew:	0.68			
Prob(H) (two-sided):	0.00	Kurtosis:	7.70			
=====						

TABLO 4. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 20 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)	Log Likelihood	-1512.343			
Date:	Sun, 09 Jul 2023	AIC	3030.686			
Time:	14:22:33	BIC	3039.574			
Sample:	01-01-2009	HQIC	3034.298			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.4403	0.040	-10.933	0.000	-0.519	-0.361
ma.S.L12	-0.5849	0.060	-9.694	0.000	-0.703	-0.467
sigma2	9.48e+07	2.05e-10	4.62e+17	0.000	9.48e+07	9.48e+07
=====						
Ljung-Box (L1) (Q):	2.01	Jarque-Bera (JB):	26.14			
Prob(Q):	0.16	Prob(JB):	0.00			
Heteroskedasticity (H):	1.54	Skew:	0.20			
Prob(H) (two-sided):	0.14	Kurtosis:	5.06			
=====						

TABLO 5. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 10 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)	Log Likelihood	-1481.384			
Date:	Sun, 09 Jul 2023	AIC	2968.769			
Time:	14:30:02	BIC	2977.657			
Sample:	01-01-2009	HQIC	2972.381			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.4181	0.037	-11.414	0.000	-0.490	-0.346
ma.S.L12	-0.5035	0.046	-11.019	0.000	-0.593	-0.414
sigma2	5.598e+07	2.71e-10	2.07e+17	0.000	5.6e+07	5.6e+07
=====						
Ljung-Box (L1) (Q):	2.50	Jarque-Bera (JB):	58.93			
Prob(Q):	0.11	Prob(JB):	0.00			
Heteroskedasticity (H):	1.15	Skew:	0.22			
Prob(H) (two-sided):	0.62	Kurtosis:	6.11			
=====						

TABLO 6. TÜRKİYE GENELİ İÇİN SARIMA İSTATİSTİKLERİ: 5 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(0, 1, 2)x(0, 1, [1], 12)	Log Likelihood	-1445.030			
Date:	Sun, 09 Jul 2023	AIC	2898.061			
Time:	14:32:48	BIC	2909.912			
Sample:	01-01-2009	HQIC	2902.877			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ma.L1	-0.4110	0.037	-11.085	0.000	-0.484	-0.338
ma.L2	0.3035	0.054	5.586	0.000	0.197	0.410
ma.S.L12	-0.3699	0.045	-8.248	0.000	-0.458	-0.282
sigma2	2.9e+07	3.02e-10	9.59e+16	0.000	2.9e+07	2.9e+07
=====						
Ljung-Box (L1) (Q):	1.12	Jarque-Bera (JB):	59.20			
Prob(Q):	0.29	Prob(JB):	0.00			
Heteroskedasticity (H):	1.54	Skew:	0.60			
Prob(H) (two-sided):	0.14	Kurtosis:	5.91			
=====						

TABLO 7. ESKİŞEHİR ŞUBESİ İÇİN SARIMA İSTATİSTİKLERİ: 200 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(2, 2, 0)x(1, 0, 0, 12)	Log Likelihood	-1057.238			
Date:	Sun, 09 Jul 2023	AIC	2122.477			
Time:	17:19:42	BIC	2134.624			
Sample:	01-01-2009	HQIC	2127.411			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.6838	0.035	-19.662	0.000	-0.752	-0.616
ar.L2	-0.3317	0.037	-9.078	0.000	-0.403	-0.260
ar.S.L12	0.4957	0.058	8.608	0.000	0.383	0.609
sigma2	5.247e+04	2827.389	18.559	0.000	4.69e+04	5.8e+04
=====						
Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	498.89			
Prob(Q):	0.94	Prob(JB):	0.00			
Heteroskedasticity (H):	56.47	Skew:	-0.86			
Prob(H) (two-sided):	0.00	Kurtosis:	11.65			
=====						

**TABLO 8. ESKİŞEHİR ŞUBESİ REESKONT BÖLGESİ İÇİN SARIMA İSTATİSTİKLERİ:
100 TL**

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(0, 1, 2)x(1, 0, [], 12)	Log Likelihood	-1340.184			
Date:	Sun, 09 Jul 2023	AIC	2688.367			
Time:	17:19:57	BIC	2700.541			
Sample:	01-01-2009	HQIC	2693.312			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ma.L1	-0.2031	0.045	-4.550	0.000	-0.291	-0.116
ma.L2	0.3502	0.062	5.654	0.000	0.229	0.472
ar.S.L12	0.4820	0.064	7.530	0.000	0.357	0.608
sigma2	1.858e+06	8.47e+04	21.934	0.000	1.69e+06	2.02e+06
=====						
Ljung-Box (L1) (Q):	1.52	Jarque-Bera (JB):	1767.63			
Prob(Q):	0.22	Prob(JB):	0.00			
Heteroskedasticity (H):	30.53	Skew:	2.48			
Prob(H) (two-sided):	0.00	Kurtosis:	18.78			
=====						

TABLO 9. ESKİŞEHİR ŞUBESİ İÇİN SARIMA İSTATİSTİKLERİ: 50 TL

SARIMAX Results						
=====						
Dep. Variable:	y			No. Observations:	156	
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)			Log Likelihood	-1130.697	
Date:	Sun, 09 Jul 2023			AIC	2267.394	
Time:	17:20:01			BIC	2276.282	
Sample:	01-01-2009			HQIC	2271.005	
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.3051	0.049	-6.212	0.000	-0.401	-0.209
ma.S.L12	-0.6835	0.078	-8.800	0.000	-0.836	-0.531
sigma2	4.177e+05	3.45e+04	12.102	0.000	3.5e+05	4.85e+05
=====						
Ljung-Box (L1) (Q):	0.17	Jarque-Bera (JB):	206.59			
Prob(Q):	0.68	Prob(JB):	0.00			
Heteroskedasticity (H):	3.41	Skew:	1.19			
Prob(H) (two-sided):	0.00	Kurtosis:	8.39			
=====						

TABLO 10. ESKİŞEHİR ŞUBESİ İÇİN SARIMA İSTATİSTİKLERİ: 20 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)	Log Likelihood	-1039.218			
Date:	Sun, 09 Jul 2023	AIC	2084.435			
Time:	17:20:06	BIC	2093.324			
Sample:	01-01-2009	HQIC	2088.047			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.3250	0.058	-5.633	0.000	-0.438	-0.212
ma.S.L12	-0.3123	0.092	-3.398	0.001	-0.492	-0.132
sigma2	1.195e+05	1.06e+04	11.228	0.000	9.86e+04	1.4e+05
=====						
Ljung-Box (L1) (Q):	0.06	Jarque-Bera (JB):	34.53			
Prob(Q):	0.80	Prob(JB):	0.00			
Heteroskedasticity (H):	2.36	Skew:	0.48			
Prob(H) (two-sided):	0.00	Kurtosis:	5.21			
=====						

TABLO 11. ESKİŞEHİR ŞUBESİ İÇİN SARIMA İSTATİSTİKLERİ: 10 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(1, 1, 0)x(1, 0, 0, 12)	Log Likelihood	-1097.936			
Date:	Sun, 09 Jul 2023	AIC	2201.872			
Time:	17:20:08	BIC	2211.003			
Sample:	01-01-2009	HQIC	2205.581			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-0.3448	0.050	-6.865	0.000	-0.443	-0.246
ar.S.L12	0.6041	0.052	11.535	0.000	0.501	0.707
sigma2	7.453e+04	4940.620	15.085	0.000	6.48e+04	8.42e+04
=====						
Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	87.90			
Prob(Q):	0.95	Prob(JB):	0.00			
Heteroskedasticity (H):	1.67	Skew:	0.94			
Prob(H) (two-sided):	0.07	Kurtosis:	6.17			
=====						

TABLO 12. ESKİŞEHİR ŞUBESİ İÇİN SARIMA İSTATİSTİKLERİ: 5 TL

SARIMAX Results						
=====						
Dep. Variable:	y	No. Observations:	156			
Model:	SARIMAX(3, 2, 0)x(0, 1, 0, 12)	Log Likelihood	-1041.533			
Date:	Sun, 09 Jul 2023	AIC	2091.065			
Time:	17:20:13	BIC	2102.889			
Sample:	01-01-2009	HQIC	2095.870			
	- 12-01-2021					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

ar.L1	-1.1938	0.055	-21.793	0.000	-1.301	-1.086
ar.L2	-0.8194	0.089	-9.191	0.000	-0.994	-0.645
ar.L3	-0.3176	0.075	-4.241	0.000	-0.464	-0.171
sigma2	1.255e+05	1.1e+04	11.414	0.000	1.04e+05	1.47e+05
=====						
Ljung-Box (L1) (Q):	1.08	Jarque-Bera (JB):	37.34			
Prob(Q):	0.30	Prob(JB):	0.00			
Heteroskedasticity (H):	2.09	Skew:	-0.00			
Prob(H) (two-sided):	0.01	Kurtosis:	5.51			
=====						

Ek-2: Türkiye Geneli İçin Kupür Bazında Emisyon Hacmi Verileri

TABLO 13. TÜRKİYE GENELİ İÇİN KUPÜR BAZINDA EMİSYON HACMI VERİLERİ

Dönem	200T	100T	50T	20T	10T	5T
2009-01	8.876.914	39.951.103	93.505.642	64.092.306	50.694.248	22.980.385
2009-02	17.860.948	63.006.557	139.763.666	90.841.136	78.469.785	39.772.471
2009-03	21.334.586	80.618.029	159.515.041	102.041.422	91.379.726	54.386.575
2009-04	21.311.212	99.272.284	182.185.538	113.360.739	102.870.382	67.043.738
2009-05	20.933.523	106.420.142	185.968.818	116.945.558	106.685.369	74.303.404
2009-06	21.050.993	115.661.991	189.705.081	118.271.255	108.725.758	80.488.726
2009-07	20.895.294	122.506.134	196.520.892	122.386.183	112.014.303	86.055.827
2009-08	21.214.158	129.593.623	197.785.993	122.156.041	113.429.738	90.523.300
2009-09	22.991.606	140.061.499	205.815.777	129.304.749	123.787.086	104.414.032
2009-10	24.770.657	144.230.295	211.271.346	130.115.468	122.148.390	101.922.905
2009-11	29.375.982	173.212.630	252.641.491	149.377.116	136.229.854	111.745.695
2009-12	30.356.988	155.535.572	219.043.310	127.276.730	120.430.064	104.290.084
2010-01	29.624.755	154.197.762	211.329.925	121.970.059	115.768.123	101.335.124
2010-02	32.190.158	161.019.008	217.138.757	121.792.133	115.785.742	100.849.073
2010-03	32.998.770	164.604.650	214.347.747	119.285.076	115.310.541	100.795.919
2010-04	34.037.783	174.600.686	220.726.209	124.158.655	119.188.144	102.464.706
2010-05	39.439.876	175.862.536	219.714.185	125.188.915	119.616.337	102.894.406
2010-06	40.525.781	181.406.929	222.919.030	129.076.287	122.373.281	104.913.394
2010-07	39.516.816	189.498.672	234.213.008	135.940.556	128.446.488	109.240.970
2010-08	40.810.342	195.406.272	238.180.772	137.133.196	130.205.656	112.976.560
2010-09	40.530.127	198.148.603	235.498.396	138.848.967	135.251.578	121.273.494
2010-10	41.893.730	211.470.452	252.181.233	146.353.792	139.327.502	121.345.383
2010-11	44.169.866	210.438.508	238.110.250	139.936.243	138.099.135	125.859.379
2010-12	47.977.732	219.430.021	243.057.557	138.634.596	136.015.500	120.132.703
2011-01	47.729.411	219.150.182	236.639.268	134.659.869	132.994.279	117.989.758
2011-02	49.696.609	224.213.395	237.474.752	134.189.694	132.474.041	117.037.889
2011-03	48.913.995	229.818.868	238.637.996	133.366.897	133.300.957	117.466.260
2011-04	51.890.312	244.454.038	249.140.093	140.348.577	139.163.916	120.191.281
2011-05	55.215.030	243.211.189	242.260.248	139.603.065	138.240.488	119.875.061
2011-06	56.998.324	252.943.247	247.037.529	143.536.838	141.952.685	121.470.340
2011-07	58.752.175	260.552.235	251.973.093	150.678.373	146.980.316	126.130.517
2011-08	65.106.263	297.084.764	286.621.498	170.476.466	166.105.557	142.631.830
2011-09	61.860.680	268.990.985	250.668.473	154.168.149	154.156.023	137.480.096
2011-10	61.064.745	272.011.181	249.259.406	152.985.961	153.516.539	137.054.299
2011-11	61.606.172	257.979.535	230.622.510	144.306.607	148.397.189	137.109.516
2011-12	61.505.297	258.622.193	230.142.172	144.142.683	146.911.486	133.152.592
2012-01	57.075.021	244.857.439	213.340.269	135.903.089	139.861.194	129.328.842
2012-02	57.261.372	242.556.537	208.718.579	134.563.941	137.313.138	126.941.955
2012-03	58.280.896	253.412.124	217.186.575	143.706.225	140.599.564	127.101.534
2012-04	60.147.422	260.322.707	219.619.940	149.169.917	140.852.456	125.748.528
2012-05	61.080.684	253.922.118	213.181.990	149.230.487	140.149.746	125.105.088
2012-06	62.600.438	272.569.862	227.338.803	160.547.518	147.345.325	129.150.043

2012-07	63.477.895	267.438.640	219.804.317	157.473.241	145.803.107	131.305.364
2012-08	66.816.675	288.715.330	234.732.387	170.064.117	159.469.402	148.102.284
2012-09	70.899.261	291.051.587	231.268.848	166.664.276	154.183.828	142.729.306
2012-10	74.462.551	307.270.962	240.215.218	175.768.281	161.557.540	151.156.890
2012-11	72.130.294	290.314.606	219.998.047	162.115.595	148.577.060	142.183.530
2012-12	71.395.075	294.409.974	220.574.565	161.599.412	146.705.187	140.028.397
2013-01	66.794.313	286.419.368	209.003.151	154.606.465	140.928.515	136.466.007
2013-02	66.688.962	293.066.356	211.480.054	155.393.011	140.592.005	136.334.930
2013-03	69.024.593	307.237.314	219.383.688	161.114.990	143.718.020	137.801.883
2013-04	68.110.780	313.244.810	219.430.444	165.147.319	136.236.312	145.937.846
2013-05	72.158.988	323.575.556	224.293.190	169.602.332	143.364.747	148.624.765
2013-06	76.476.240	352.991.379	241.554.149	179.708.862	151.739.054	152.835.091
2013-07	79.315.772	357.417.960	242.230.559	179.745.265	155.057.036	158.643.544
2013-08	84.536.617	375.917.422	255.585.052	191.829.974	165.937.323	168.879.309
2013-09	88.741.269	380.768.374	254.861.837	189.043.591	162.046.336	162.785.093
2013-10	89.726.757	378.044.990	254.417.667	190.875.748	166.715.085	168.678.374
2013-11	91.157.792	375.360.964	248.056.828	185.443.891	160.784.527	161.427.227
2013-12	92.923.900	376.333.928	244.800.016	181.433.444	157.704.299	158.535.921
2014-01	97.465.166	375.155.105	239.671.130	178.610.762	154.751.019	156.103.649
2014-02	99.798.887	382.471.190	239.397.831	176.239.368	153.763.607	156.025.265
2014-03	102.068.471	390.356.888	240.921.630	176.941.139	155.162.391	156.387.199
2014-04	98.540.334	397.967.012	242.636.885	179.652.643	159.470.251	157.197.330
2014-05	97.984.223	402.811.983	244.660.353	182.294.812	162.177.910	159.707.891
2014-06	101.739.433	413.423.517	251.467.049	186.590.566	165.846.895	162.393.915
2014-07	107.459.219	461.686.460	284.363.859	210.054.420	185.655.421	181.519.736
2014-08	109.296.865	439.972.730	264.761.884	196.746.394	175.060.359	177.499.530
2014-09	114.962.541	460.561.545	274.616.914	202.264.757	177.772.226	179.022.208
2014-10	110.398.455	448.949.501	263.483.270	197.315.640	174.637.282	177.510.676
2014-11	105.578.770	436.112.663	255.388.625	192.185.290	170.551.278	171.980.930
2014-12	112.278.493	434.761.277	251.112.909	188.078.942	167.903.171	168.846.332
2015-01	120.889.244	435.724.073	245.831.377	186.539.647	167.792.558	168.016.774
2015-02	127.703.094	449.307.668	249.934.429	189.517.221	169.377.465	167.552.643
2015-03	131.081.533	452.801.437	248.130.158	188.349.148	168.230.708	166.015.336
2015-04	136.222.570	479.412.462	261.451.979	197.096.718	173.917.635	168.647.511
2015-05	133.627.681	480.193.421	262.790.791	201.680.862	176.972.488	170.228.987
2015-06	135.555.697	485.397.265	266.044.277	203.263.277	180.373.700	173.874.366
2015-07	135.686.705	509.500.465	282.484.940	216.504.574	194.556.205	193.000.959
2015-08	146.405.928	516.829.336	283.579.927	212.738.572	188.282.106	188.296.656
2015-09	155.838.836	553.172.012	306.670.378	228.808.906	200.392.003	200.097.142
2015-10	149.759.126	536.664.843	290.514.692	216.353.697	190.038.810	191.305.349
2015-11	147.021.673	525.102.847	279.739.984	209.079.575	184.094.962	185.423.063
2015-12	143.710.211	530.645.128	279.555.084	207.865.994	184.238.758	184.459.024
2016-01	144.069.584	532.115.741	276.305.391	205.971.012	182.754.577	183.365.035
2016-02	150.634.710	537.744.136	275.640.071	204.641.376	181.374.234	181.933.375
2016-03	142.954.770	537.562.317	272.687.651	203.279.441	181.543.079	180.938.673
2016-04	140.791.182	557.759.687	282.849.754	209.580.922	185.274.992	182.648.497

2016-05	151.988.052	564.169.624	285.082.400	211.521.585	187.275.347	184.077.064
2016-06	158.457.142	615.180.973	316.201.296	230.325.952	205.162.121	201.541.211
2016-07	167.941.212	643.770.833	328.159.932	233.536.348	208.408.029	206.717.873
2016-08	166.098.973	650.589.363	325.219.786	227.040.251	202.520.671	201.430.868
2016-09	166.477.672	663.977.681	331.677.894	232.525.029	208.155.135	207.158.745
2016-10	163.852.330	659.506.193	323.480.886	226.619.286	201.696.093	200.709.302
2016-11	165.109.450	646.492.736	310.350.394	218.593.351	197.187.150	197.017.518
2016-12	166.725.601	662.174.494	316.753.690	220.104.589	198.195.672	198.341.364
2017-01	166.403.296	648.991.431	304.341.303	213.009.151	194.792.562	197.502.158
2017-02	162.195.961	656.233.781	306.685.140	213.661.807	194.470.971	196.499.376
2017-03	162.837.793	678.242.221	316.604.210	218.973.150	198.753.509	198.822.269
2017-04	163.970.696	710.106.603	333.776.690	227.916.364	205.010.872	201.235.986
2017-05	157.533.639	680.271.978	314.713.427	219.819.931	202.025.120	202.651.348
2017-06	157.762.526	727.964.069	345.839.075	246.156.691	225.783.572	226.684.005
2017-07	158.273.011	711.268.304	333.988.684	234.224.193	215.360.889	221.398.698
2017-08	168.209.903	808.965.341	390.532.248	264.303.874	236.059.955	236.167.542
2017-09	161.842.216	725.251.844	341.097.576	238.476.683	219.152.277	226.099.864
2017-10	172.127.618	715.024.272	332.151.861	232.927.473	213.216.937	218.797.139
2017-11	174.110.527	710.845.423	327.807.897	229.678.478	211.473.442	216.733.507
2017-12	169.554.334	726.450.856	336.294.830	233.593.646	213.959.193	219.296.613
2018-01	160.012.169	692.118.735	314.097.283	220.984.928	206.598.948	217.208.855
2018-02	159.511.477	699.463.255	316.591.180	221.557.292	207.367.079	217.055.828
2018-03	169.893.394	730.414.303	329.801.108	226.905.934	211.337.414	219.350.523
2018-04	170.456.576	755.508.551	340.562.025	232.847.362	216.440.694	222.078.541
2018-05	173.462.829	754.056.299	335.726.288	234.390.520	220.017.412	229.403.084
2018-06	180.196.692	832.837.543	371.216.001	254.393.735	237.844.241	251.894.683
2018-07	188.018.087	817.116.595	356.931.458	243.885.869	228.217.278	245.593.235
2018-08	227.659.959	903.510.018	409.229.283	269.514.595	247.252.567	260.653.946
2018-09	210.823.779	829.608.073	368.019.982	248.066.949	230.925.602	249.268.470
2018-10	202.298.398	762.546.275	335.719.390	235.249.971	222.276.150	240.138.170
2018-11	191.613.691	734.419.711	326.141.368	231.215.173	219.659.308	237.547.878
2018-12	187.259.880	709.580.365	315.531.966	226.880.567	218.025.582	236.224.191
2019-01	173.955.904	680.612.981	303.501.241	220.572.485	214.626.591	235.944.260
2019-02	176.061.680	691.827.747	310.013.905	222.831.654	216.662.393	236.321.862
2019-03	178.263.440	723.595.353	326.739.308	229.813.682	220.321.880	238.559.732
2019-04	186.084.487	721.559.208	322.596.872	229.130.316	220.794.196	241.502.430
2019-05	206.530.807	827.043.267	384.587.124	264.329.378	252.310.685	271.009.214
2019-06	192.982.158	765.239.593	346.633.439	247.124.367	240.770.714	268.918.381
2019-07	200.563.976	750.668.623	337.721.213	238.561.880	234.229.002	265.014.029
2019-08	224.023.915	817.882.807	378.992.839	257.333.644	251.621.714	279.955.855
2019-09	223.161.286	794.020.643	361.231.242	245.056.404	241.874.276	270.069.385
2019-10	229.522.241	791.949.115	356.246.699	241.402.285	240.031.778	266.263.475
2019-11	232.247.512	795.335.328	356.079.365	240.049.307	239.703.011	266.815.047
2019-12	241.992.686	790.083.950	347.456.729	235.442.861	236.805.509	266.477.347
2020-01	247.535.062	796.587.082	349.685.786	235.451.102	239.110.692	268.880.885
2020-02	257.142.979	823.209.689	360.833.895	239.655.217	242.522.161	270.643.422

2020-03	285.383.761	919.470.298	399.844.438	250.202.340	250.749.622	273.208.922
2020-04	346.870.189	1.143.269.575	484.925.844	270.956.227	261.304.271	283.753.252
2020-05	356.955.085	1.205.876.280	499.241.219	280.369.244	270.983.816	296.123.714
2020-06	354.005.691	1.172.899.568	469.592.562	266.338.578	267.000.289	297.951.194
2020-07	391.068.065	1.316.906.348	539.127.370	294.449.344	290.224.129	314.970.231
2020-08	353.691.164	1.154.493.106	469.995.791	269.430.535	275.801.141	306.464.535
2020-09	349.475.664	1.112.080.178	449.620.610	261.830.633	271.067.358	301.851.006
2020-10	357.023.592	1.099.904.957	442.375.103	258.708.214	270.244.944	300.136.939
2020-11	323.189.064	1.004.843.223	410.350.928	249.429.046	266.284.379	297.533.055
2020-12	315.102.176	965.723.845	394.821.586	245.148.344	264.764.535	296.987.094
2021-01	306.527.685	930.818.552	382.274.203	242.015.240	264.006.795	296.415.361
2021-02	303.573.135	914.156.768	379.166.206	240.801.340	265.218.987	296.275.853
2021-03	332.350.843	912.388.863	369.256.946	237.102.153	263.278.403	296.117.815
2021-04	353.046.912	975.402.582	390.829.142	247.981.304	273.838.059	302.570.252
2021-05	373.079.969	1.069.281.187	415.125.234	258.191.862	285.383.586	310.564.661
2021-06	380.173.239	1.080.290.517	408.057.290	252.688.664	282.102.572	311.853.424
2021-07	393.642.283	1.156.240.403	455.955.612	276.964.623	304.964.058	328.545.322
2021-08	382.791.917	1.089.948.622	428.223.702	262.227.198	293.625.408	324.021.270
2021-09	404.963.541	1.073.297.277	415.666.909	258.273.445	290.026.176	321.079.953
2021-10	448.091.495	1.126.800.900	427.323.218	264.630.193	295.435.349	321.760.217
2021-11	486.567.384	1.139.371.226	411.428.578	257.724.882	291.203.728	317.635.401
2021-12	474.580.684	1.104.170.985	407.426.182	257.409.815	293.480.026	317.979.752
2022-01	481.096.389	1.109.924.012	408.287.309	255.514.584	293.640.189	319.111.658
2022-02	497.265.084	1.133.209.949	416.280.696	257.283.590	298.872.635	324.515.742
2022-03	536.988.900	1.156.964.863	414.462.949	256.325.932	303.020.625	330.531.775
2022-04	638.516.567	1.321.227.028	492.098.002	287.655.658	341.371.004	358.111.030
2022-05	607.139.509	1.182.217.377	433.696.785	259.899.037	333.201.634	347.953.716
2022-06	679.146.649	1.175.573.445	430.303.614	248.622.340	351.298.271	345.120.102
2022-07	759.833.311	1.211.961.353	462.798.070	253.523.980	381.708.021	356.133.999
2022-08	803.252.845	1.202.318.922	450.563.194	241.029.833	388.653.136	356.642.904
2022-09	841.239.443	1.188.051.244	447.920.533	232.458.763	401.113.999	355.786.575
2022-10	884.125.726	1.156.475.012	437.782.669	223.058.624	408.885.214	353.742.253
2022-11	915.396.164	1.155.800.749	418.189.025	215.476.147	415.705.135	354.590.651
2022-12	966.978.002	1.176.729.088	409.561.726	211.205.340	423.514.089	358.560.009

Kaynak: EVDS