

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**DERİN ÖĞRENMENİN MODÜLASYON SINIFLANDIRMA
UYGULAMALARI İÇİN ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ
ÜZERİNDE GERÇEKLENMESİ**

KORAY ALIÇ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2022**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENMENİN MODÜLASYON SINIFLANDIRMA UYGULAMALARI İÇİN ALANDA PROGRAMLANABİLİR KAPİ DİZİLERİ ÜZERİNDE GERÇEKLENMESİ

Koray ALIÇ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Gökhan SOYSAL

Modülasyon sınıflandırma hem askeri hem de sivil haberleşme sistemlerinde kritik öneme sahiptir. Modülasyon sınıflandırma uygulamaları ile askeri ortamlardan istihbarat elde etme, haberleşme sistemlerini karıştırma ve yayıncı tanıma gibi çalışmalar yapılmaktadır. Sivil ortamda ise haberleşme kanalının efektif olarak yönetilmesi ve alıcının değişen kanal koşullarına göre otomatik olarak doğru demodülatör tipinin seçilmesi gibi uygulamalar yapılmaktadır. Literatürde modülasyon sınıflandırma için olabilirlik tabanlı ve öznitelik tabanlı yöntemler sıklıkla kullanılmaktadır. Bunun dışında son zamanlarda derin öğrenme tabanlı yöntemler uygulanmaya başlanmıştır. Bu tez çalışmasında derin öğrenme tabanlı yöntemler ile haberleşme sinyallerinde modülasyon sınıflandırma başarımı incelenmiştir. Evrişimli sinir ağları (CNN) ve özyinelemeli sinir ağları (RNN) türünde mimariler oluşturularak, bu mimarilerin sınıflandırma başarımları nicemlenmiş ve nicemlenmemiş modeller için karşılaştırılmıştır. Derin öğrenme uygulamaları genel olarak grafik işlemci birimleri (GPU) üzerinde çalıştırılmaktadır. Bu tez çalışmasında derin öğrenme mimarisi alanda programlanabilir kapı dizileri (FPGA) üzerinde gerçekleştirilmiş ve çalıştırılmıştır. Derin öğrenme mimarisinin GPU üzerinden FPGA ortamına aktarılması ortaya çıkan nicemleme kaybı incelenmiştir. Sınıflandırma başarımlar oranı 0dB SNR seviyesinde yapılan testlerde ortalama %50 olurken, 10dB SNR seviyesinden itibaren bu oran %90 seviyesine ulaşmıştır. Nicemlenmiş modellerdeki performans kaybı ortalama %5 seviyesinde olmuştur.

Şubat 2022, 102 sayfa

Anahtar Kelimeler: modülasyon sınıflandırma, CNN, RNN, FPGA, nicemleme

ABSTRACT

Master Thesis

FIELD PROGRAMMABLE GATE ARRAY BASED IMPLEMENTATION OF DEEP LEARNING FOR MODULATION CLASSIFICATION

Koray ALIÇ

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Dr. Gökhan SOYSAL

Modulation classification is critical in both military and civilian communication systems. With modulation classification applications, studies such as obtaining intelligence from military environments, jamming communication systems, and recognition of transmitters can be done. In the civilian environment, applications such as effective management of the communication channel and automatically selecting the right demodulator type according to the changing channel conditions of the receiver are carried out. In literature, there are many likelihood-based and feature-based methods exist. Apart from this, deep learning-based methods have been applied recently. This thesis has studied modulation classification performance in communication signals with deep learning-based methods. Convolutional neural networks (CNN) and recursive neural networks (RNN) types of architectures were created, and the classification performances of these architectures were compared for quantized and Non-quantized models. Deep learning applications are generally run on graphics processor units (GPU). In this thesis, deep learning architecture is implemented and run on field-programmable gate arrays (FPGA). The quantization loss that occurs when transferring the deep learning architecture to the FPGA environment over the GPU has been scrutinized. While the classification success rate was 50% in tests performed at 0dB SNR level, this rate reached 90% above 10dB SNR level. Performance loss in quantized models averaged 5%.

February 2022, 102 pages

Key Words: modulation classification, CNN, RNN, FPGA, quantization

ÖNSÖZ ve TEŞEKKÜR

Çalışmalarımı yönlendiren, bilgi ve önerileriyle katkıda bulunan danışman hocam sayın Dr. Öğretim Üyesi Gökhan SOSYAL'a (Ankara Üniversitesi Elektrik ve Elektronik Mühendisliği Anabilim Dalı Öğretim Üyesi) ve Dr. Öğretim Üyesi Aykut KALAYCIOĞLU'na (Ankara Üniversitesi Elektrik ve Elektronik Mühendisliği Anabilim Dalı Öğretim Üyesi) teşekkür ederim.

Koray ALIÇ
Ankara, Şubat 2022

İÇİNDEKİLER

TEZ ONAYI	
ETİK	i
ÖZET	ii
ABSTRACT	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ.....	1
2. DERİN ÖĞRENME KAVRAMLARI	6
2.1 Yapay Zekâ ve Makine Öğrenmesi.....	6
2.2 Sinir Ağlarına Giriş	7
2.2.1 Algılayıcı	7
2.2.2 Çok katmanlı algılayıcılar	9
2.2.3 Aktivasyon fonksiyonları.....	10
2.2.3.1 Sigmoid fonksiyonu.....	11
2.2.3.2 Hiperbolik tanjant fonksiyonu.....	12
2.2.3.3 Doğrultulmuş doğrusal birim	13
2.2.4 Derin sinir ağları	14
2.2.5 Sinir ağları nasıl öğrenir?.....	15
2.2.6 Eksik öğrenme ve aşırı öğrenme.....	17
2.3 Evrişimli Sinir Ağları.....	18
2.3.1 Evrişim katmanı.....	20
2.3.2 Ortaklama katmanı.....	21
2.3.3 Tam bağlı katman	22
2.3.4 Özelleşmiş evrişimli sinir ağı yapıları	23
2.4 Özyinelemeli Sinir Ağları	24
2.4.1 Özelleşmiş özyinelemeli sinir ağı yapıları	26
2.5 Mimari eğitimi ve raporlanmasında kullanılan teknik terimler	26
2.5.1 Sınıflandırma matrisi.....	26
2.5.2 Epok (epoch)	28
2.5.3 Grup boyutu	28
2.5.4 Grup normalizasyonu	28
3. MODÜLASYON TEKNİKLERİ.....	30
3.1 Analog Modülasyon Türleri.....	30
3.1.1 Genlik modülasyonu	30
3.1.2 Frekans modülasyonu.....	30
3.2 Sayısal Modülasyon Türleri	31
3.2.1 Genlik kaydırmalı anahtarlama	31
3.2.1.1 Dur-başla anahtarlama.....	31
3.2.1.2 Dörtlü genlik kaydırmalı anahtarlama	32
3.2.1.3 Sekizli genlik kaydırmalı anahtarlama	33
3.2.2 Faz kaydırmalı anahtarlama.....	34
3.2.2.1 İkili faz kaydırmalı anahtarlama.....	34
3.2.2.2 Karesel faz kaydırmalı anahtarlama.....	35
3.2.2.3 8-16-32 Faz kaydırmalı anahtarlama.....	35

3.2.3 Genlik ve faz kaydırmalı anahtarlama	36
3.2.4 Karesel genlik modülasyonu	37
4. ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ.....	39
4.1 FPGA Çalışmalarında Nicemleme.....	39
4.2 FPGA Üzerinde Matematik İşlemlerinin Kaynak Kullanımı.....	40
5. YAPILAN ÇALIŞMALAR	42
5.1 Modülasyon Sinyalleri Veri Seti	42
5.2 Modülasyon Sınıflandırma Derin Öğrenme Yapıları.....	54
5.2.1 CNN-1 mimarisi	54
5.2.2 CNN-2 mimarisi	58
5.2.3 CNN-3 mimarisi	61
5.2.4 ResNet-1 mimarisi.....	65
5.2.5 ResNet-2 mimarisi.....	69
5.2.6 ResNet-3 mimarisi.....	73
5.2.7 ResNet-4 mimarisi.....	77
5.2.8 GRU mimarisi.....	81
5.2.9 LSTM mimarisi	85
5.2.10 Mimarilerin doğruluk oranı karşılaştırması	89
5.3 Mimarilerinin FPGA Üzerine Aktarılması.....	90
5.3.1 FPGA test ortamının kurulması	90
5.3.2 Mimarinin nicemlenmesi ve performans analizi.....	91
5.3.3 FPGA test sonuçları	93
6. TARTIŞMA ve SONUÇ.....	95
KAYNAKLAR	98
ÖZGEÇMİŞ.....	102

KISALTMALAR DİZİNİ

AI Yapay Zekâ
AM Genlik Modülasyonu
AM-DSB-SC Bastırılmış Taşıyıcılı Çift Yan Bant Genlik Modülasyonu
AM-DSB-WC Taşıyıcılı Çift Yan Bant Genlik Modülasyonu
AM-SSB-SC Bastırılmış Taşıyıcılı Tek Yan Bant Genlik Modülasyonu
AM-SSB-WC Taşıyıcılı Tek Yan Bant Genlik Modülasyonu
APSK Genlik ve Faz Kaydırmalı Anahtarlama
ASIC Uygulamaya Özgü Tümleşik Devre
ASK Genlik Kaydırmalı Anahtarlama
BN Grup Normalizasyonu
BPSK İkili Faz Kaydırmalı Anahtarlama
BRAM Blok Rastgele Erişimli Bellek
CLB Yapılandırılabilir Mantık Bloğu
CNN Evrişimli Sinir Ağı
CPU Merkezi İşlemci Birimi
dB Desibel
DL Derin Öğrenme
DNN Derin Sinir Ağı
DPU Derin Öğrenme İşlemci Birimi
DSP Sayısal İşaret İşlemcisi
FB Öznitelik Tabanlı
FF İki Durumlu Devre
FM Frekans Modülasyonu
FN Yanlış Negatif
FP Yanlış Pozitif
FPGA Alanda Programlanabilir Kapı Dizisi
FSK Frekans Kaydırmalı Anahtarlama
GMSK Gauss En Küçük Kaydırmalı Anahtarlama

GPU Grafik İşlemci Birimi
GRU Kapılı Özyinelemeli Birim
HDL Donanım Tanımlama Dili
LB Olabilirlik Tabanlı
LSTM Uzun-Kısa Süreli Hafıza
LUT Aramalı Tablo
ML Makine Öğrenmesi
MSK En Küçük Kaydırmalı Anahtarlama
MUX Çoklayıcı
OOK Dur-Başla Anahtarlama
OQPSK Ortogonal Karesel Faz Kaydırmalı Anahtarlama
PSK Faz Kaydırmalı Anahtarlama
QAM Karesel Genlik Modülasyonu
QPSK Karesel Faz Kaydırmalı Anahtarlama
ReLU Doğrultulmuş Doğrusal Birim
RNN Özyinelemeli Sinir Ağı
SNR Sinyal Gürültü Oranı
TP Doğru Pozitif

ŞEKİLLER DİZİNİ

Şekil 2.1 Algılayıcı (perceptron) genel yapısı	7
Şekil 2.2 AND ve XOR doğruluk tablosu üzerinde doğrusal sınıflandırma	8
Şekil 2.3 Çok katmanlı algılayıcı genel yapısı	9
Şekil 2.4 XOR algılayıcı şeması	10
Şekil 2.5 Sigmoid fonksiyonu ve türevi	12
Şekil 2.6 Hiperbolik tanjant fonksiyonu ve türevi	13
Şekil 2.7 ReLU aktivasyon fonksiyonu ve türevi	14
Şekil 2.8 Derin sinir ağları genel yapısı	14
Şekil 2.9 DNN yapısı içerisindeki algılayıcı gösterimi	15
Şekil 2.10 Öğrenci not tahmini örnek ağ yapısı	15
Şekil 2.11 (a) eksik öğrenme, (b) ideal öğrenme, (c) aşırı öğrenme	17
Şekil 2.12 Seyreltme işlemi örneği	18
Şekil 2.13 CNN mimarisi genel yapısı	19
Şekil 2.14 Ardışık CNN blokları ve sınıflandırma	19
Şekil 2.15 Evrişim işlemi, adım sayısı = 1	20
Şekil 2.16 Evrişim işlemi 0 ekleme, adım sayısı = 1	21
Şekil 2.17 Ortaklama 2x2, adım sayısı = 2	22
Şekil 2.18 Tam bağlı katman	22
Şekil 2.19 Inception CNN yapısı	23
Şekil 2.20 ResNet CNN yapısı	24
Şekil 2.21 Özyinelemeli sinir ağı genel yapısı	24
Şekil 2.22 Farklı uygulamalar için özelleşmiş RNN yapıları: (a) bire bir, (b) çoktan bire, (c) birden çokta, (d)(e) çoktan çokta	25
Şekil 2.23 Sınıflandırma matrisi örneği	26
Şekil 3.1 Genlik modülasyonu	30
Şekil 3.2 Frekans modülasyonu	31
Şekil 3.3 OOK Modülasyonu	32
Şekil 3.4 4ASK Modülasyonu	32
Şekil 3.5 4ASK yıldız kümesi şeması	33
Şekil 3.6 8ASK Modülasyonu	33
Şekil 3.7 8ASK yıldız kümesi şeması	34
Şekil 3.8 BPSK sembolü yıldız kümesi şeması	35
Şekil 3.9 QPSK sembolü yıldız kümesi şeması	35
Şekil 3.10 8PSK 16PSK 32PSK sembolleri yıldız kümesi şeması	36
Şekil 3.11 16APSK, 32APSK, 64APSK, 128APSK sembolleri yıldız kümesi şeması	37
Şekil 3.12 QAM16, QAM32, QAM64 yıldız kümesi şeması	37
Şekil 3.13 QAM128, QAM256 yıldız kümesi şeması	38

Şekil 4.1 Evrişim işleminin FPGA için blok şeması	41
Şekil 5.1 QPSK modülasyonu, 30dB SNR.....	43
Şekil 5.2 QPSK sinyali yıldız kümesi şeması	43
Şekil 5.3 QPSK sinyali alt-örneklenmiş yıldız kümesi şeması.....	44
Şekil 5.4 BPSK modülasyonu, 30dB SNR.....	44
Şekil 5.5 BPSK sinyali yıldız kümesi şeması.....	45
Şekil 5.6 BPSK sinyali alt-örneklenmiş yıldız kümesi şeması.....	45
Şekil 5.7 4ASK Modülasyonu 30dB SNR	46
Şekil 5.8 4ASK sinyali yıldız kümesi şeması.....	46
Şekil 5.9 4ASK sinyali alt-örneklenmiş yıldız kümesi şeması.....	47
Şekil 5.10 16APSK Modülasyonu, 30dB SNR	48
Şekil 5.11 16APSK sinyali yıldız kümesi şeması	48
Şekil 5.12 16APSK sinyali alt-örneklenmiş yıldız kümesi şeması.....	49
Şekil 5.13 128QAM Modülasyonu, 30dB SNR	50
Şekil 5.14 128QAM sinyali yıldız kümesi şeması	50
Şekil 5.15 128QAM sinyali alt-örneklenmiş yıldız kümesi şeması	51
Şekil 5.16 BPSK Modülasyonu, 10dB SNR	52
Şekil 5.17 BPSK Modülasyonu, 0dB SNR	52
Şekil 5.18 BPSK modülasyonu yıldız kümesi şeması SNR karşılaştırması.....	53
Şekil 5.19 QPSK modülasyonu yıldız kümesi şeması SNR karşılaştırması.....	53
Şekil 5.20 Evrişim blok yapısı	54
Şekil 5.21 CNN-1 mimarisi genel yapısı.....	55
Şekil 5.22 CNN-1 modeli sınıflandırma matrisi.....	55
Şekil 5.23 CNN1 modeli SNR'a göre doğruluk oranı.....	57
Şekil 5.24 CNN-2 mimarisi genel yapısı.....	58
Şekil 5.25 CNN-2 modeli sınıflandırma matrisi.....	58
Şekil 5.26 CNN2 modeli SNR'a göre doğruluk oranı.....	61
Şekil 5.27 CNN-3 mimarisi genel yapısı.....	61
Şekil 5.28 CNN-3 modeli sınıflandırma matrisi.....	62
Şekil 5.29 CNN-3 modeli SNR'a göre doğruluk oranı	65
Şekil 5.30 ResNet-1 blok yapısı.....	65
Şekil 5.31 ResNet-1 mimarisi genel yapısı	66
Şekil 5.32 ResNet-1 modeli sınıflandırma matrisi	66
Şekil 5.33 ResNet-1 modeli SNR'a göre doğruluk oranı	69
Şekil 5.34 ResNet-2 blok yapısı.....	69
Şekil 5.35 ResNet-2 modeli sınıflandırma matrisi	70
Şekil 5.36 ResNet-2 modeli SNR'a göre doğruluk oranı	73
Şekil 5.37 ResNet-3 blok yapısı.....	73
Şekil 5.38 ResNet-3 modeli sınıflandırma matrisi	74

Şekil 5.39 ResNet-3 modeli SNR'a göre doğruluk oranı	77
Şekil 5.40 ResNet-4 mimarisi iç yapısı	78
Şekil 5.41 ResNet-4 modeli sınıflandırma matrisi	78
Şekil 5.42 ResNet-4 modeli SNR'a göre doğruluk oranı	81
Şekil 5.43 GRU mimarisi genel yapısı	81
Şekil 5.44 GRU modeli sınıflandırma matrisi	82
Şekil 5.45 GRU modeli SNR'a göre doğruluk oranı	85
Şekil 5.46 LSTM mimarisi genel yapısı	85
Şekil 5.47 LSTM modeli sınıflandırma matrisi	86
Şekil 5.48 LSTM modeli SNR'a göre doğruluk oranı	89
Şekil 5.49 Mimarilerin doğruluk oranı grafiği	90
Şekil 5.50 HDL blok yapısı üst seviye yapısı	91
Şekil 5.51 HDL blok yapısı DPU kullanımı	91
Şekil 5.52 Nicemlenmiş mimarilerin doğruluk oranı grafiği	92

ÇİZELGELER DİZİNİ

Çizelge 2.1 Şekil 2.27’de verilen örnek sınıflandırma tablosu için performans tablosu	28
Çizelge 4.1 π ve e sayılarının 64-32-16 ve 8-bit nicemlenmiş ifadesi	39
Çizelge 4.2 Çarpma işleminde kaynak kullanımı	40
Çizelge 4.3 Toplama işleminde kaynak kullanımı	40
Çizelge 5.1 Kullanılan derin öğrenme modelleri ve F1 skorları	54
Çizelge 5.2 CNN-1 modeli detaylı performans raporu	56
Çizelge 5.3 CNN-1 modeli SNR’a göre F1 skoru değerlendirmesi	57
Çizelge 5.4 CNN-2 modeli detaylı performans raporu	59
Çizelge 5.5 CNN-2 modeli SNR’a göre F1 skoru değerlendirmesi	60
Çizelge 5.6 CNN-3 modeli detaylı performans raporu	63
Çizelge 5.7 CNN-3 modeli SNR’a göre F1 skoru değerlendirmesi	64
Çizelge 5.8 ResNet-1 modeli detaylı performans raporu	67
Çizelge 5.9 ResNet-1 modeli SNR’a göre F1 skoru değerlendirmesi	68
Çizelge 5.10 ResNet-2 mimarisi detaylı performans raporu	71
Çizelge 5.11 ResNet-2 mimarisi SNR’a göre F1 skoru değerlendirmesi	72
Çizelge 5.12 ResNet-3 mimarisi detaylı performans raporu	75
Çizelge 5.13 ResNet-3 mimarisi SNR’a göre F1 skoru değerlendirmesi	76
Çizelge 5.14 ResNet-4 modeli detaylı performans raporu	79
Çizelge 5.15 ResNet-4 mimarisi SNR’a göre F1 skoru değerlendirmesi	80
Çizelge 5.16 GRU modeli detaylı performans raporu	83
Çizelge 5.17 GRU mimarisi SNR’a göre F1 skoru değerlendirmesi	84
Çizelge 5.18 LSTM modeli detaylı performans raporu	87
Çizelge 5.19 LSTM mimarisi SNR’a göre F1 skoru değerlendirmesi	88
Çizelge 5.20 Mimarilerin 30dB SNR seviyesindeki nicemlenmiş ve nicemlenmemiş durumlardaki başarım oranları	92
Çizelge 5.21 Mimarilerin FPGA üzerinde yapılan testlerinde başarım oranının SNR(dB) seviyesine göre incelenmesi	93
Çizelge 5.22 Mimarilerin çıktı üretme hızları	93

1. GİRİŞ

Modülasyon sınıflandırma, alıcı tarafından alınan sinyalin modülasyon tipinin bilinmediği durumlarda sinyalin modülasyon tipinin belirlenmesi için kullanılan bir yöntemdir. Elektronik harp ortamında alınan bir sinyalin modülasyon tipinin belirlenmesi, alınan sinyal içerisindeki mesajın çözülebilmesini, mesaj içeriğinin karıştırılabilmesine veya sinyali gönderen birim hakkında fikir sahibi olunmasına yardımcı olur (Zhu vd. 2019). Sivil uygulamalarda ise alıcının değişen ortam ve kanal koşullarına göre otomatik olarak doğru demodülatör tipinin seçilmesini sağlar. Hem askeri hem de sivil uygulamalarda kritik öneme sahiptir.

Derin öğrenme yöntemlerinin uygulanabilir olmasından önce otomatik modülasyon sınıflandırma iki temel yaklaşım ile gerçekleştirilmekteydi. Bunlar, istatistiksel bilgiye dayalı olabilirlik prensibi tabanlı (likelihood-based (LB)) sınıflandırma ve öznitelik çıkarımı tabanlı (feature-based (FB)) klasik sınıflandırma yöntemleridir. LB metodu ile yüksek sınıflandırma başarımı elde edilebilir ancak çok fazla hesaplama ve işlem yükü gerektirmektedir. FB metodu ile sınıflandırma ise daha az karmaşık olan bir yapı içermektedir (Hazza vd. 2013). Ancak bu metot ile sınıflandırma yapabilmek için modüle edilmiş sinyal içerisinde özniteliklerin çıkarılması ve hangi özniteliklerin kullanılacağına seçilmesi gerekmektedir. Literatürde farklı öznitelik seçimleriyle yapılan çalışmalar mevcuttur. (Park vd. 2007) yaptıkları çalışmada sinyale ait anlık bant genişliği, faz ve frekans bilgilerini kullanmıştır. (Guan vd. 2004) çalışmada öznitelik olarak sinyalin fourier dönüşümü kullanılmıştır. (Park vd. 2008) yaptığı çalışmada ise dalgacık dönüşümünden faydalanmıştır.

Literatürde LB ve FB metotlarını kullanan birçok otomatik modülasyon sınıflandırma çalışması mevcuttur. (Wei ve Mendel 2000) çalışmada LB metodunu kullanmış ve karesel genlik modülasyonlarını (quadrature amplitude modulation (QAM)) sınıflandırmıştır. Çalışmada sinyal gürültü oranı (signal noise ratio (SNR)) ve alınan kaydın sembol uzunluğuna bağlı olarak başarımları değişen sonuçlar almıştır. Yaklaşık 10dB SNR seviyesinde 100 sembol veri ile %60, 200 sembol veri ile %70 ve 1000 sembol veri ile %90 başarımları raporlamıştır. (Huang ve Polydoros 1995) çalışmada faz kaydırmalı anahtarlamalı (phase shift keying (PSK)) sinyalleri, LB metodu ile sınıflandırmıştır. İdeal koşullarda ve 100 sembol uzunluğundaki veriler ile yaptığı

testlerinde ikili PSK (Binary PSK (BPSK)) ile karesel PSK (quadrature PSK (QPSK)) sinyallerini -2dB SNR seviyesinde bile yaklaşık %95 başarımla sınıflandırıldığını raporlamıştır. Aynı SNR seviyesi ve aynı metot ile QPSK ve 8PSK sinyallerini ise yaklaşık %60 başarımla sınıflandırabilmiştir. 8PSK ve 16PSK sinyalleri için ise 4dB SNR seviyesinde bile ancak %55 başarımla ulaşabilmiştir. Modülasyon şeması büyüdükçe sınıflandırma başarımının düştüğü gözükmemektedir. (Dobre vd. 2004) çalışmasında FB metodunu kullanmıştır. Öznitelik olarak sinyallerin döngüsel birikimini (cyclic cumulant (CC)) kullanmıştır. QAM ve 16QAM modüli sinyalleri sınıflandırmaya çalışmıştır. Sınıflandırma başarımının SNR ve sembol sayısı ile değişiminden bahsetmiştir. 500 sembol uzunluğundaki sinyal ile yaptığı çalışmada 5dB, 7dB ve 10dB SNR seviyesinde yaklaşık %75 sınıflandırma başarımı elde etmiştir. Sembol sayısı 900 olduğunda ise yaklaşık %95 oranında başarı sağlamıştır. (Park vd. 2008) çalışmasında ise dalgacık dönüşümü katsayıları öznitelik olarak kullanılmıştır. Destek vektör makinesi kullanarak 8 farklı sayısal modülasyon türünü sınıflandırmıştır. Sınıflandırılmaya çalışılan modülasyon türleri genlik kaydırmalı anahtarlama (amplitude shift keying (ASK)), 4ASK, frekans kaydırmalı anahtarlama (frequency shift keying (FSK)), 4FSK, faz kaydırmalı anahtarlama (phase shift keying (PSK)), QPSK, en küçük kaydırmalı anahtarlama (minimum shift keying (MSK)) ve 16QAM olarak verilmiştir. 10dB SNR seviyesinden itibaren %90 doğruluk oranı elde edildiği belirtilmiştir.

Günümüzde popüler olan derin öğrenme teknikleri ile geleneksel yöntemlerden çok daha başarılı sonuçlar elde edilmiştir. Derin öğrenme mimarileri öznitelik çıkarılması işlemini kendi kendine yaptığı için elle öznitelik oluşturulması ve seçilmesi sorunu ortadan kalkmıştır.

Literatürde farklı derin öğrenme mimarileri kullanarak modülasyon sınıflandırma yapan çalışmalar bulunmaktadır. (Karahana ve Kalaycıoğlu 2020) çalışmasında özyinelemeli sinir ağı (recurrent neural network (RNN)) yapılarından olan uzun-kısa süreli hafıza (long-short term memory (LSTM)) ve çift yönlü LSTM (*Bi*-LSTM) mimarileri karşılaştırmalı kullanılarak modülasyon sınıflandırma yapılmıştır. 11 farklı modülasyon sinyali ile yapılan çalışmada 5dB SNR seviyesinden itibaren LSTM ile %85, *Bi*-LSTM mimarisi ile de %91 doğruluk oranı elde edildiği raporlanmıştır. Modülasyonların yıldız küme şemasındaki benzerlik olması sınıflandırma başarımını düşürdüğü gösterilmiştir. 16QAM ve 64QAM sinyallerinin sınıflandırmasında problem yaşanmıştır. (Zhang vd.

2020) çalışmasında ise evrişimli sinir ağları (convolutional neural network (CNN)) kullanılarak 12 farklı sayısal modüleli sinyalin sınıflandırması yapılmıştır. Kullanılan CNN katmanlarının sayısının başarımlarına etkisi incelenmiştir. 10dB SNR seviyesinde 16 katmanlı CNN yapısı ile %89, 8 katmanlı CNN ile %84 ve 4 katmanlı CNN ile de %65 başarımlar elde edildiği raporlanmıştır. CNN tabanlı bir başka çalışma da (Dileep vd. 2020) tarafından yapılmıştır. Benzetim ortamında oluşturulmuş 8 farklı sayısal modülasyon ile yapılan çalışmada 2dB SNR seviyesinin üstünde %97 başarımlarına ulaşıldığı belirtilmiştir. (Vanhoy vd. 2019) çalışmasında hem CNN hem de LSTM mimarilerinin birleşimi bir yapıyı içeren derin sinir ağı yapısı kullanılmıştır. 20 farklı modülasyon ile yapılan çalışmada modülasyon şemasının büyüklüğünün başarımlarındaki etkisi anlatılmıştır. BPSK ve 4ASK gibi basit modülasyon şemasına sahip sinyaller 0dB SNR seviyesinde bile sırasıyla %80 ve %70 başarımlar göstermiştir. Bunun yanında 16PSK, 16QAM ve 32QAM gibi büyük modülasyon şemasına sahip sinyaller ise 18dB SNR seviyesinde bile %30 başarımlar oranını geçememiştir.

İncelenen çalışmaların bir kısmında benzetim ortamlarında yapay olarak üretilmiş sinyaller kullanılırken bazılarında ise gerçek ortamdan kaydedilmiş sinyaller kullanılmıştır. Çalışmalarda farklı veri setlerinin kullanılması nedeniyle bu çalışmaları doğrudan birbirleriyle karşılaştırmak doğru bir yaklaşım olmayacaktır. Ancak bütün çalışmalar SNR seviyesinin sınıflandırma başarımlarında önemli olduğunu göstermektedir. Ayrıca büyük modülasyon şemalarının sınıflandırılması görece daha küçük modülasyon şemalarının sınıflandırılmasına göre daha düşük başarımlar oranı göstermiştir. Başarımlar oranını etkileyen bir diğer önemli parametre de kaydedilen sinyalin uzunluğudur.

Derin öğrenme uygulamaları genel olarak merkezi işlemci birimi (control processing unit (CPU)) veya grafik işlemci birimleri (graphics processing unit (GPU)) donanımları üzerinde çalıştırılmaktadır. GPU paralel işlem yeteneğinin çok daha fazla olması sebebiyle çok daha hızlı sonuçlar üretmektedir. (Sundar vd. 2018, Aida-Zade vd. 2019) çalışmalarında CPU ve GPU kullanılarak derin öğrenme mimarilerinin eğitim süreleri karşılaştırılmıştır. Kullanılan CPU ve GPU'ların çekirdek sayıları ve saat frekanslarına bağlı olarak sonuçlar değişse de GPU kullanımı eğitim süresini 2 kat ile 80 kat arasında kısalttığı belirtilmiştir. CPU ve GPU üzerinde çalışma yapmak ve bu ortamda mimari tasarlamayı destekleyen birçok kütüphane bulunması nedeniyle çok tercih edilmektedir. Ayrıca bu donanımlara ulaşmak çok daha ucuz ve kolay olarak görülmektedir. CPU ve

GPU dışında alanda programlanabilir kapı dizileri (field programmable gate array (FPGA)) ve uygulamaya özgü tümleşik devre (application specific integrated circuit (ASIC)) ile yapılan derin öğrenme çalışmaları da bulunmaktadır. FPGA yapıları içlerinde sınırlı miktarda sayısal devre elemanı içermektedir. CPU ve GPU donanımlarında yapılan çalışmalarda kayan noktalı (floating-point) parametreler ve matematiksel işlemler yapılabilmektedir. Ancak FPGA donanımlarında kayan noktalı sayılarla işlem yapılması kaynak kullanımını arttırmaktadır. Bu nedenle FPGA üzerinde tasarım yapılırken kullanılacak parametrelerde nicemleme yoluna gidilmektedir ve sabit noktalı (fixed point) sayılar kullanılmaktadır. (Lee vd. 2019) çalışmasında farklı derin öğrenme mimarilerinde parametrelerin nicemlenmesiyle gerçekleşen performans kaybını raporlamıştır. Kayan noktalı olarak %84.11 başarımlı oranı elde ettiği çalışmada, parametreleri sabit noktalı 12-bit ile ifade ettiğinde %83.93, sabit noktalı 8 bit ile ifade ettiğinde ise %75.63 başarımlı oranı elde edildiğini göstermiştir. Başarımlı oranında düşüş yaşanırken FPGA uygulamaları için kaynak kullanımı, CPU ve GPU uygulamaları için ise güç kullanımı açısından avantaj oluşmaktadır. (Huang vd. 2017) çalışmasında CPU, GPU ve FPGA donanımları harcadıkları güç açısından karşılaştırılmıştır. Güç verimliliği açısından FPGA ile GPU'ya göre 4 kat, CPU'ya göre ise 50 kat daha iyi sonuç alınmıştır. Güç tüketiminin daha düşük olması soğutma ihtiyacının da bir o kadar az olmasını gerektirir. Daha küçük donanımlar ile derin öğrenme uygulamaları yapılmasının önü açılmaktadır. FPGA ortamında tasarım yapmak GPU ve CPU ortamında olduğu kadar kolay değildir. Bu alanda derin öğrenme mimarilerinin uygulanabilmesini kolaylaştıracak kütüphaneler yeni yeni oluşmaya başlamıştır. ASIC kullanımı ise maliyet açısından yeterince ulaşılabilir olmadığı için yaygınlaşmamış durumdadır.

Bu tez kapsamında FPGA üzerinde çalışacak bir derin öğrenme mimari tasarlanmıştır. Tasarlanan bu mimari Xilinx ailesinin FPGA ürünleri üzerinde gerçekleştirilmiştir. Mimarinin FPGA üzerine aktarılması için Xilinx Vitis-AI yazılımı ve TensorFlow kütüphanelerinden faydalanılmıştır. Bu tez çalışmasında (O'Shea vd. 2018) tarafından oluşturulmuş içerisinde gerçek ortamdan kaydedilmiş ve benzetim ortamlarında oluşturulmuş sinyallerin bulunduğu veri seti kullanılmıştır. Kullanılan veri seti içerisinde -20dB SNR seviyesinden 30dB SNR seviyesine kadar 2dB adımlarla kaydedilmiş sinyaller bulunmaktadır. Bu veri seti literatürde karşılaştırma veri seti olarak kullanılmaktadır. Yapılan çalışma ile modülasyon sınıflandırma gibi hem sivil hem de

askerî açıdan kritik öneme sahip bir probleme günümüzün gelişen teknolojilerinden birisi olan derin öğrenme algoritmaları ile FPGA üzerinde çalışan bir çözüm gösterilmiştir. Derin öğrenme algoritmalarının FPGA üzerinde gerçekleşmesi konusunda örnek olabilecek çalışma yapılmıştır ve modülasyon sınıflandırma dışında başka problemlerin de FPGA üzerinde çözümlenebilmesine yol göstermektedir.

Bu tezin 2.bölümünde derin öğrenme ile ilgili temel kavramlar ve mimariler anlatılmıştır. Derin öğrenme yapıları içerisindeki katmanlar incelenmiştir. 3.bölümde tez içerisinde incelenecek olan modülasyon sinyalleri ile ilgili genel bilgiler verilmiş ve yapıları incelenmiştir. 4.bölümde temel FPGA kavramları üzerinde durulmuştur. FPGA üzerinde tasarımı yapılırken nicemlemenin önemi açıklanmıştır. 5.bölümde çalışmada kullanılan veri seti incelenmiş, farklı derin öğrenme mimarileri kullanılarak eğitilmiş ve FPGA üzerinde gerçekleştirilmiştir.

2. DERİN ÖĞRENME KAVRAMLARI

Tezin bu bölümünde derin öğrenme uygulamalarında sıklıkla kullanılan temel kavramlar, fonksiyonlar ve mimariler anlatılmıştır.

2.1 Yapay Zekâ ve Makine Öğrenmesi

Yapay zeka (Artificial Intelligence - AI), insan beyninin karar verme ve problem çözme yeteneklerini taklit ederek akıllı makine ve bilgisayar programları yapmak için kullanılan mühendislik bilimidir. En basit haliyle AI, problem çözmek için bilgisayar bilimi ve veri gruplarını birleştiren bir alandır. Tarihsel olarak incelendiğinde AI çalışmaları İngiliz matematikçi Alan Turing tarafından ikinci dünya savaşında tasarlanan Bombe makinesine kadar gitmektedir. Tasarlanan bu makine ilk elektromekanik bilgisayar olarak kabul edilmektedir ve şifreli mesajların çözülmesi amacıyla kullanılmıştır (Haenlein ve Kaplan 2019). Alan Turing, 1950 yılında yaptığı yayınla akıllı makineler ve bu makinelerin zekâ seviyelerinin nasıl test edileceğini tanımlamıştır. Turing testi adı verilen bu çalışma hâlâ güncelliğini korumaktadır (Turing 1950). AI, terim olarak ise ilk 1956 yılında John McCarthy tarafından Dartmouth konferansında kullanılmıştır (Benko ve Lanyi 2009). Düzenlenen konferansta AI resmi olarak ortaya koyulmuş ve insan beynini taklit edebilecek makineler inşa etmek için çalışma alanları oluşturmaya çalışılmıştır. Bu konferans ile başlayan çalışmaların ardından insanla diyalog kurabilen veya hanoi kuleleri problemi gibi basit problemleri çözebilen bilgisayar programları geliştirilmiştir (Haenlein ve Kaplan 2019).

Makine öğrenmesi (Machine Learning - ML) ise yapay zeka çalışmalarının istatistik ve veri madenciliği çalışmalarından etkilenmiş bir alt kümesi olarak düşünülebilir (Gama ve Carvalho 2010). (Alpaydin 2020) makine öğrenmesi tanımını “Makine öğrenimi, örnek verileri veya geçmiş deneyimleri kullanarak bir performans kriterini optimize etmek için yapılmış bilgisayar programıdır” şeklinde yapmıştır. Tarihsel olarak incelendiğinde ise ML terimi ilk kez (Samuel 1959) tarafından ortaya koyulmuştur. ML temel olarak verileri sınıflandırmak veya verilen verileri kullanarak bir tahmin yapmak için kullanılmaktadır. ML uygulamaları denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) olmak üzere üç temel şekilde eğitilmektedir. Denetimli öğrenme için hem giriş hem de çıkış bilgisi programa verilir ve programın girişler ile çıkışları ilişkilendiren kuralı öğrenmesi

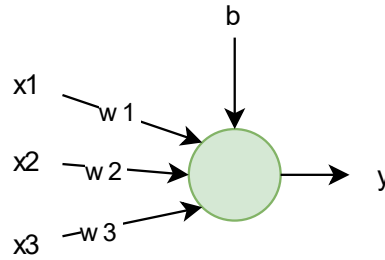
beklenir. Denetimsiz öğrenmede programa sadece girişler verilir ve programın girişler içerisindeki desenleri veya sınıfları bulması beklenmektedir. Pekiştirmeli öğrenmede ise ardışık olarak verilen girişlerin ulaşılması istenen hedefe olumlu veya olumsuz yöndeki katkısı programa geri beslenerek, programın en uygun hareket tarzını öğrenmesi beklenir (Alpaydin, 2020). Günümüz ML uygulamaları ile bilgisayarlı görü, konuşma tanıma ve görüntü sınıflandırma gibi problemler çözülebilmektedir.

2.2 Sinir Ağlarına Giriş

Sinir ağları (Neural Networks – NN) veya daha genel adıyla yapay sinir ağları (Artificial Neural Networks – ANN), ML algoritmalarında sıklıkla kullanılan metotlardan birisidir. AI çalışmalarında olduğu gibi insan beyninden esinlenilerek oluşturulmuş bir yapıdır. İnsan beyninde nöron adı verilen sinir hücreleri bulunmaktadır. Bu sinir hücrelerinin birbiriyle haberleşmesi insan beynini bütün çalışma mantığını oluşturmaktadır. ANN çalışmaları da bu nöron yapısından esinlenilerek oluşturulmuştur. Giriş verisi, nöron adı verilen ve katmanlar halinde ardışık olarak birbirine bağlanan yapı içerisinde analiz edilir (Theobald 2017). ANN yapısının temellerini oluşturmak için algılayıcı (Perceptron) kavramını incelemek gerekir.

2.2.1 Algılayıcı

ANN yapısının en basit parçası algılayıcı (perceptron) olarak tanımlanabilir. Birbirinden doğrusal olarak ayrıştırılabilecek verilerin sınıflandırılması için kullanılmaktadır. Literatürde sıklıkla gösterilen algılayıcı yapısı Şekil 2.1’de gösterilmiştir. x_1, x_2, x_3 sistemin girişlerini; w_1, w_2, w_3 ağırlıklandırma katsayılarını; b değeri ön eğilimlenleme değerini, y değeri de çıkışı ifade eder. Tek bir nörondan oluşan en basit sinir ağı modeli olarak ifade edilebilir (Haykin 2008).



Şekil 2.1 Algılayıcı (perceptron) genel yapısı

Eğitim sürecinde nörona denetimli öğrenme biçiminde girişler ile çıkışlar verilir, ağırlıklandırma katsayılarının ve ön eğilimlenme değerinin öğrenilmesi beklenmektedir. Öğrenme işlemi tamamlandıktan sonra sisteme verilen yeni girişlere göre y değerinin belirli bir eşik değerinin altında veya üstünde olması durumuna ile sınıflandırma yapılmaktadır. Eşitlik (2.1)'de y değerinin hesaplanma şekli verilmiştir. Eşitlik (2.1)'de kullanılan f fonksiyonuna ikili basamak aktivasyon fonksiyonu denmektedir. Eşitlik (2.2)'de bu örnek için kullanılan aktivasyon fonksiyonu gösterilmiştir. Bölüm 2.2.3'de aktivasyon fonksiyonları kullanım amacı ve literatürde kullanılan diğer aktivasyon fonksiyonları verilmiştir.

$$y = f\left(\sum_{i=1}^{i=3} x_i w_i + b\right) \quad (2.1)$$

$$f(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (2.2)$$

Tek katmanlı algılayıcı yöntemi ile sadece doğrusal olarak birbirinden ayrılabilen sınıflandırma problemleri çözülebilir. Literatürde bu durumu örneklemek için Şekil 2.2'de verilen mantık işlemlerinden olan ve (AND) işlemi ile dışlayan veya (XOR) işlemi örnek olarak sıklıkla kullanılmaktadır.

	0	1
0	0	0
1	0	1

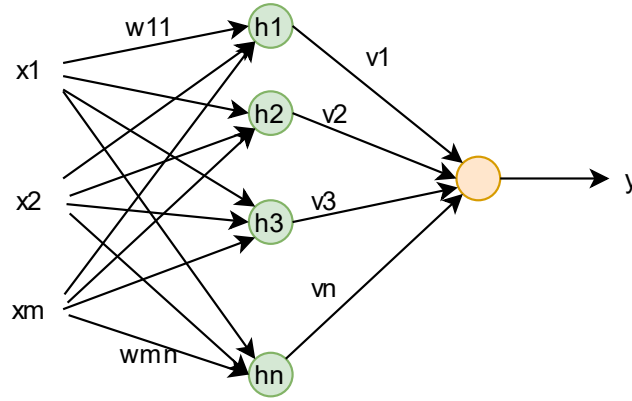
	0	1
0	0	1
1	1	0

Şekil 2.2 AND ve XOR doğruluk tablosu üzerinde doğrusal sınıflandırma

AND işlemi tek bir doğrusal fonksiyon ile gruplandırılabilirken, XOR işlemini gruplayabilmek için ise iki doğrusal fonksiyona ihtiyaç duyulmaktadır. XOR problemi tek bir algılayıcı ile çözülememektedir. Algılayıcı kullanarak daha karmaşık problemlerin çözülebilmesi için ise çok katmanlı algılayıcılar (multilayer perceptrons) yöntemi kullanılmaktadır (Zhao vd. 2002).

2.2.2 Çok katmanlı algılayıcılar

Algılayıcı kavramında sadece bir nöron bulunmaktaydı. Çok katmanlı algılayıcı (Multilayer perceptrons – MLP) ise bu nöronlardan arka arkaya birden fazla bağlanarak katmanlı bir yapı oluşturulmuştur. Literatürde sıklıkla tarif edilen MLP yapısı Şekil 2.3’de gösterilmiştir (Zhao vd. 2002). Modelin giriş olarak $x_1, x_2, \dots, x_m$ uygulanmıştır. Saklı katman olarak ifade edilen h katmanında $h_1, h_2, \dots, h_n$ nöronları bulunmaktadır. Sistemin çıkışı da y olarak ifade edilmiştir.



Şekil 2.3 Çok katmanlı algılayıcı genel yapısı

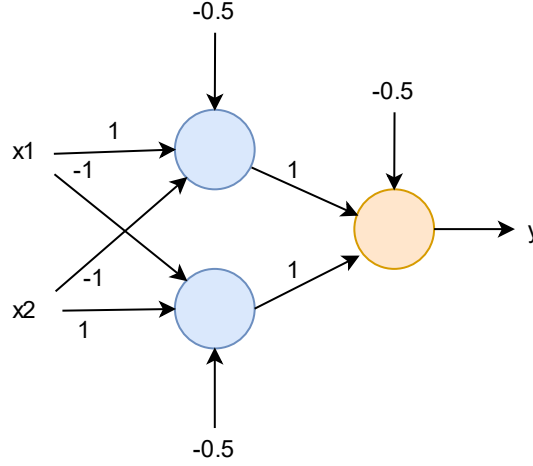
x_1 girişi ile h_1 nöronu arasında w_{11} katsayısı bulunmaktadır. x_m girişi ile h_n nöronu arasında da w_{mn} katsayısı bulunmaktadır. Benzer şekilde diğer bütün girişler ve nöronlar için de bu katsayılar bulunmaktadır. Saklı katman ile çıkış katmanı arasında da benzer şekilde v_1, v_2, v_3, v_n katsayıları bulunmaktadır. Tekli algılayıcı yapısında olduğu gibi her nöronun ayrıca b_i ön eğilimle değerleri bulunmaktadır. Nöronların çıkışları eşitlik (2.3) ve (2.4) ile hesaplanmaktadır.

$$h_i = f\left(\sum_{k=1}^m x_k w_{ki} + b_i\right) \quad (2.3)$$

$$y = f\left(\sum_{k=1}^n h_k v_k + b\right) \quad (2.4)$$

Şekil 2.2 ile tanımlanan XOR sınıflandırma probleminin çözümü için MLP yapısı Şekil 2.4’te gösterilmiştir. Modelin girişine verilen x_1, x_2 değerleri, gösterilen katsayılar ve ön eğilimlenme değerleriyle işleme sokulduğunda XOR işleminin doğruluk tablosunun

oluşturabildiği görülmektedir. Nöronların her birinde eşitlik (2.2)'de verilen ikili basamak aktivasyon fonksiyonu kullanılmıştır. Aynı problemi farklı katsayılar ve ön eğilimlenme değerleriyle çözebilecek MLP bulunmaktadır (Zhao vd. 2002).



Şekil 2.4 XOR algılayıcı şeması

2.2.3 Aktivasyon fonksiyonları

Aktivasyon fonksiyonu bir nöronun giriş ve çıkış arasında lineer olmayan bir ilişki kurmayı ve nöronlar üzerindeki bilgiyi bu doğrusal olmayan eşitlik ile bir sonraki katmana taşımayı hedefler (Guo vd. 2019). Aktivasyon fonksiyonunun doğrusal olmaması gerekmektedir. Aktivasyon fonksiyonun doğrusal olması durumunda, doğrusal fonksiyonların doğrusal kombinasyonu da doğrusal olacağı için tanımlanan çok katmanlı model tek katmanlı bir doğrusal model haline dönüşecektir. Bu durumu örneklemek için eşitlik (2.5),(2.6) ile ifade edilen ardışık bağlı iki nöron denklemini ele alabiliriz. İlk katmanın girişleri x_1, x_2 ; bu girişlere ait katsayılar w_1, w_2 ve ön eğilimlenme değeri de b_1 olarak kullanılmıştır. İlk katmanın çıkışı olan y_1 değeri de w_3 katsayısı ve b_2 ön eğilimlenme değeriyle ikinci katmanın çıkışını oluşturmaktadır.

$$y_1 = x_1 w_1 + x_2 w_2 + b_1 \quad (2.5)$$

$$y_2 = y_1 w_3 + b_2 \quad (2.6)$$

Aktivasyon fonksiyonu kullanılmadığı durumda y_1 değeri eşitlik (2.6)'da yerine yazılabilir ve eşitlik (2.7)'de verilen denklem elde edilebilir.

$$y_2 = x_1 w_1 w_3 + x_2 w_2 w_3 + b_1 w_3 + b_2 \quad (2.7)$$

Denklemlerde kullanılan w_1, w_2, w_3, b_1, b_2 değerlerinin eğitim süreci sonucunda bulunan sabit değerler olduğu bilinmektedir. Sabit değerlerin çarpımı veya toplamı da bir sabit değer oluşturacağına göre, $w_1 w_3 = w_a$, $w_2 w_3 = w_b$, $b_1 w_3 + b_2 = b_a$ şeklinde ifade edilebilirler. Yeni katsayılar ile eşitliğin tekrar yazılması durumunda eşitlik (2.8) elde edilmektedir. Elde edilen denklem tek katmanlı algılayıcı denklemine dönüşmüştür. Kurulan çok katmanlı yapı kaybolmuştur.

$$y_2 = x_1 w_a + x_2 w_b + b_a \quad (2.8)$$

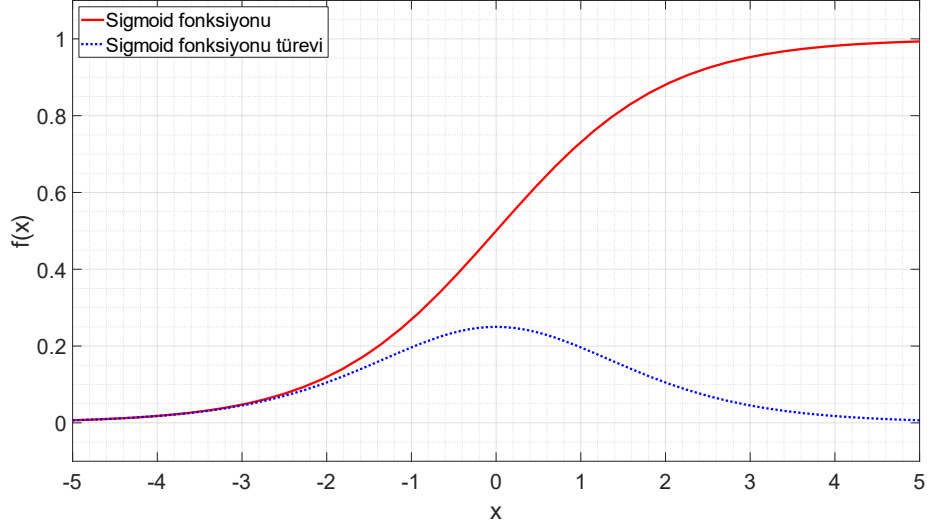
Temel olarak kullanılan bazı aktivasyon fonksiyonları şunlardır: sigmoid, tanjant hiperbolik, doğrultulmuş doğrusal birim.

2.2.3.1 Sigmoid fonksiyonu

Sigmoid fonksiyonu girişi ne olursa olsun çıkışı 0 ile 1 arasında limitlidir. Matematiksel ifadesi eşitlik (2.9)'te verilmiştir.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.9)$$

Doğrusal olmayan bir fonksiyondur. Şekil 2.5'teki grafik incelendiğinde, x değerinin -2 ile +2 arasında fonksiyonun değeri hızlıca değişmektedir. Diğer yandan uç noktalar gidildiğine ise fonksiyonun türevi sıfıra yaklaşmaktadır. Uç noktalara gidildiğinde öğrenme yavaşlarken orta noktalarda ise öğrenme daha hızlı gerçekleşmektedir.



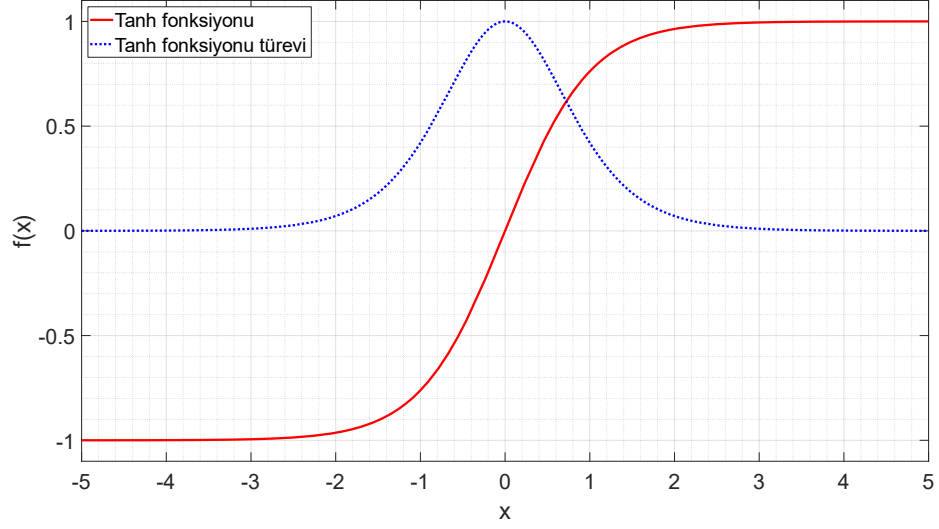
Şekil 2.5 Sigmoid fonksiyonu ve türevi

2.2.3.2 Hiperbolik tanjant fonksiyonu

Şekil 2.6’da gösterilen hiperbolik tanjant ($\tanh$) fonksiyonu, sigmoid fonksiyonuna çok benzemektedir. Aralarındaki tek fark $\tanh$ fonksiyonunun orijin etrafında simetrik olmasıdır. Çıkış değerleri -1 ile 1 arasında limitlidir. $\tanh$ fonksiyonunun matematiksel ifadesi eşitlik (2.10)’da verilmiştir.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.10)$$

Sigmoid fonksiyonunda olduğu gibi uç değerlere gidildikçe öğrenme yavaşlamaktadır. Değer aralığı daha geniş olduğu için öğrenme hızı sigmoid fonksiyonuna göre daha yüksektir.



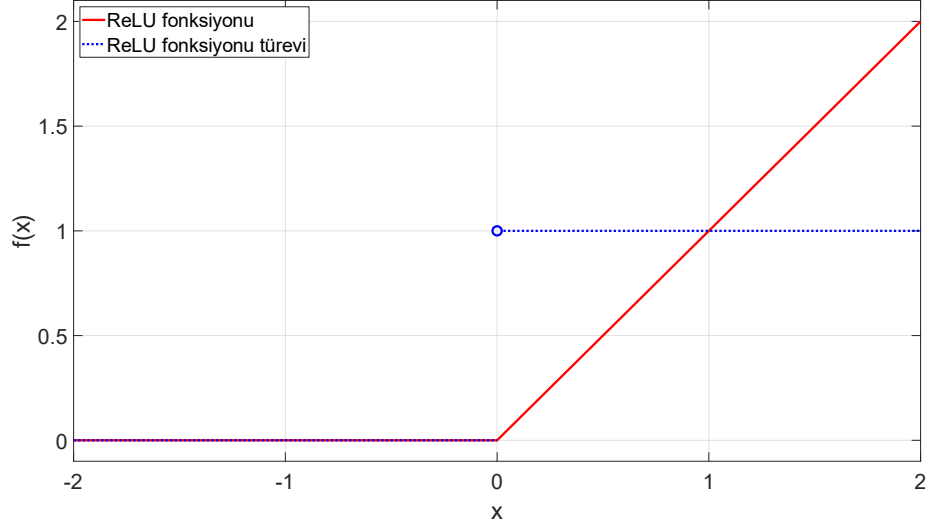
Şekil 2.6 Hiperbolik tanjant fonksiyonu ve türevi

2.2.3.3 Doğrultulmuş doğrusal birim

Şekil 2.7’de gösterilen doğrultulmuş doğrusal birim (Rectified linear unit (ReLU)) ile sadece değeri sıfırdan büyük olan nöronlar aktifleştirilmektedir. Matematiksel ifadesi eşitlik (2.11)’de verilmiştir.

$$f(x) = \max(0, x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.11)$$

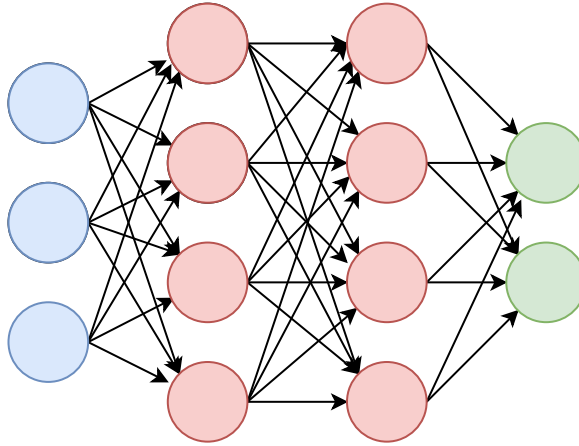
Negatif değerli nöronlar doğrudan sıfırlandığı için çok daha hızlı çalışır. Gerçeklenmesi çok kolaydır. İşlem yükü gerektirmediği için çok katmanlı mimarilerde rahatlıkla kullanılmaktadır ve günümüzde kullanılan mimariler içerisinde en çok kullanılan aktivasyon fonksiyonudur.



Şekil 2.7 ReLU aktivasyon fonksiyonu ve türevi

2.2.4 Derin sinir ağları

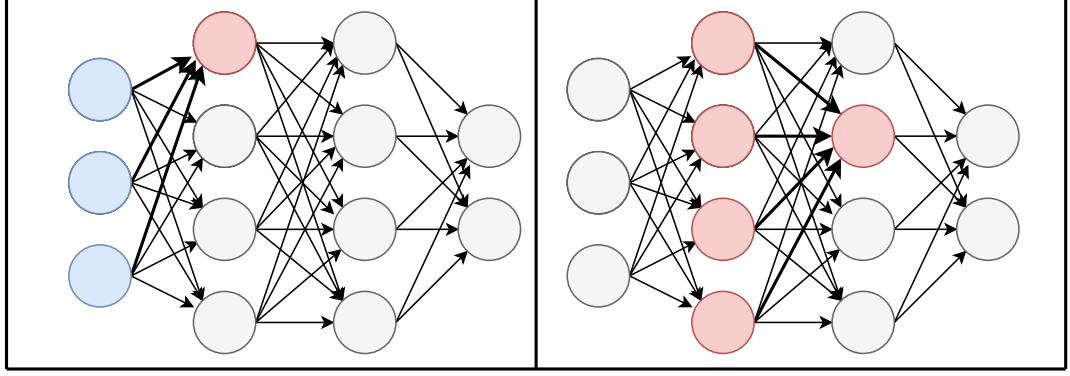
Algılayıcılar kullanılarak sinir ağları oluşturulmaktadır. Literatürde derin sinir ağları (Deep Neural Network – DNN) Şekil 2.8’de görüldüğü gibi tanımlanmaktadır (Musiol 2016). Verilen modelde mavi nöronlar giriş katmanını, kırmızı nöronlar saklı katmanları ve yeşil nöronlar da çıkış katmanını belirtmektedir. İçerisinde birden fazla saklı katman bulunması nedeniyle derin ismini almaktadır.



Şekil 2.8 Derin sinir ağları genel yapısı

Şekil 2.8’de verilen DNN yapısı küçük parçalar halinde incelendiğinde içerisinde MLP yapısını barındırdığı gözükmemektedir. Şekil 2.9’da DNN yapısı içerisindeki algılayıcıların

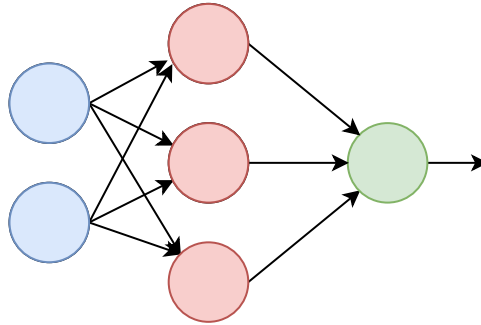
yapısı gösterilmiştir. Soldaki şekilde girişler, saklı katman içerisindeki bir nörona bağlandığı görülmektedir. Bu yapı Şekil 2.1’de verilen algılayıcı yapısının aynısıdır. Sağdaki şekilde de bir saklı katmandan, bir diğer saklı katmana ilerleyen algılayıcı gösterilmektedir.



Şekil 2.9 DNN yapısı içerisindeki algılayıcı gösterimi

2.2.5 Sinir ağları nasıl öğrenir?

NN yapılarının öğrenme mantığını kavramak için (Amini 2019) tarafından verilen örnek problem incelenebilir. Bir dersten öğrencilerin alacağı notu, öğrencinin katıldığı ders sayısı ve bitirme projesine harcadığı çalışma süresine göre tahmin edilmeye çalışılsın. Tanımlanan problemde sistemin iki girişi ve bir çıkışı olması gerektiği anlaşılmaktadır. Kullanılacak saklı katman sayısı örneğin basitleştirilmesi açısından tek olarak seçilmiştir. Bu durumda Şekil 2.10’da gösterildiği gibi bir yapı kurulduğu kabul edilebilir.



Şekil 2.10 Öğrenci not tahmini örnek ağ yapısı

Mavi olarak gösterilen giriş katmanı ile kırmızı olarak gösterilen saklı katman arasında W^1 katsayılarının, saklı katman ile yeşil olarak gösterilen çıkış katmanı arasında da W^2

katsayılarının olduğunu kabul edelim. Matematiksel ifadelerin daha kolay yazılabilmesi için modelin girişlerini x_i ve hesaplanan çıkışı da $\hat{y}$ değişkenleriyle ifade edilsin. Modelin eğitilebilmesi ve W değerlerinin bulunabilmesi için bir veri setine ihtiyaç duyulmaktadır. Bu veri seti her bir öğrencinin katıldığı ders sayısını (x_1), projeye harcadığı süreyi (x_2) ve dersten aldığı notu (y) içermelidir. Eğitim süreci rastgele olarak W katsayılarının belirlenmesi ile başlar ve rastgele katsayılar ile tahmin edilen not ile gerçekte olması gereken not arasındaki hataya göre katsayıların güncellenmesi beklenir. Olması gereken değer ile tahmin edilen değer arasındaki hatayı tanımlamak için kayıp fonksiyonu (Loss Function) denilen fonksiyonlar kullanılmaktadır. Literatürde en yaygın olarak kullanılan kayıp fonksiyonlarından birisi kareler hatası fonksiyonudur ve eşitlik (2.12)'de verildiği gibi ifade edilebilir (Zhang vd. 2021). Aşağı verilen bir örnek veri ve mevcut W değerleri için tahmin edilen değer ile olması gereken değer farkının karesi hesaplanmaktadır.

$$l(W) = (\hat{y} - y)^2 \quad (2.12)$$

Bütün modelin kayıp fonksiyonunun hesaplanması, her bir örnek veri için eşitlik (2.12) ile yapılır ve ortalaması alınır (2.13). Eşitlik (2.13) ile yapılan hesaplama maliyet fonksiyonu (cost function) da denmektedir.

$$L(W) = \frac{1}{n} \sum_{i=1}^n l_i(W) \quad (2.13)$$

Yapılan eğitimin amacı maliyet fonksiyonu değerini en küçük yapacak W parametrelerini bulmaktır. W değerlerinin bulunması için optimizasyon problemlerinin en çok kullanılan yöntemlerinden birisi olan gradyan inişi (gradient decent) algoritması sinir ağları içerisinde açık ara en sık kullanılan yöntemdir (Ruder 2016). Gradyan inişi algoritmasının sözde kod ile tanımı şu şekilde yapılabilir:

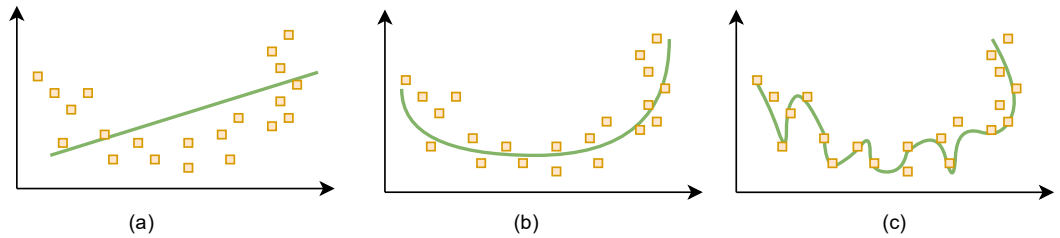
1. W katsayıları rastgele olarak seçilir.
2. Maliyet fonksiyonunun gradyanı $\frac{\partial L(W)}{\partial W}$ hesaplanır.
3. W katsayıları güncellenir. $W \leftarrow W - \eta \frac{\partial L(W)}{\partial W}$
4. W değerleri modelde yerine yazılır ve 2.adıma geri dönülür.

Algoritmanın 3.adımında kullanılan η değişkeni öğrenme katsayısı olarak tanımlanır. Parametrelerin ne kadar büyük adımlarla güncelleneceğini belirler. Değerin çok küçük seçilmesi eğitim süresinin çok uzamasına, çok büyük kullanılması da optimum noktanın etrafında salınma girip hiçbir zaman optimum noktaya yakınsayamamasına neden olabilir. Eğitim döngüsünü sonlandırmak için eğitim döngü sayısının önceden sabit belirlenmesi, $L(W)$ değerindeki değişimlerin belirli bir eşik değerinin altına inmesi veya $L(W)$ değerinin belirli bir döngü sayısı boyunca hiç küçülememesi gibi metotlar kullanılmaktadır (Ruder 2016).

Eğitimin sonlanması ardından tasarlanan ağa, eğitim veri seti içerisinde kullanılmayan bir öğrencinin bilgileri verilmesi ile öğrencinin dersten hangi notu alacağı tahmin edilebilmektedir.

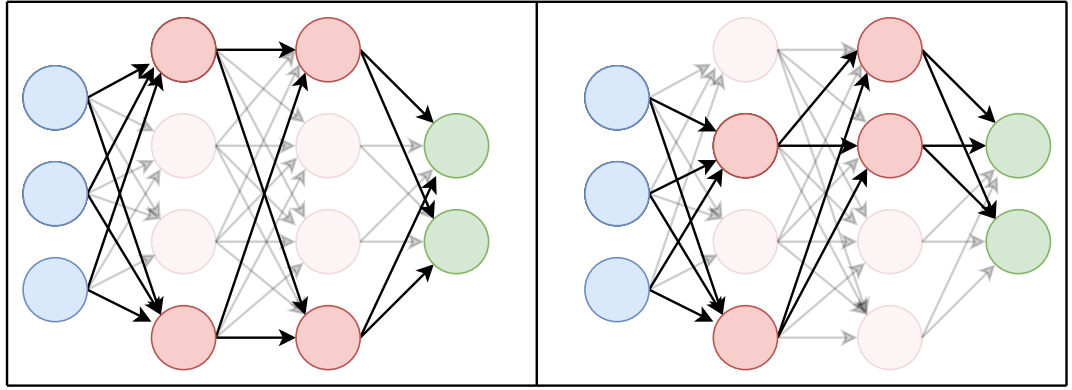
2.2.6 Eksik öğrenme ve aşırı öğrenme

Öğrenme süreci sonunda elde edilen model verilerin tamamını temsil edecek genel bir formül oluşturmalıdır. Sadece eğitim verilerini temsil edebilen veya verilerin bir kısmını hiç temsil edemeyen modeller kullanılmamalıdır. Bu şekildeki modellere, modeli test etmek için yeni bir veri verildiğinde model doğru tahmini üretemeyecektir (Amini, 2019). Şekil 2.11’de (a) grafiğinde gösterildiği üzere eğitilen model verileri temsil edememektedir. Model verileri temsil edecek kapasiteye sahip değildir ve eksik öğrenme(underfitting) durumu gerçekleşmiştir. Aynı şeklin (c) grafiğinde ise aşırı öğrenme(overfitting) durumu gerçekleşmiştir. Model çok karmaşık bir yapıda oluşturulmuş veya problemin genel olarak ifade edilmesini engelleyecek derecede çok fazla parametre içermektedir. (b) grafiğinde ise ideal durum gösterilmiştir. Model, verilen verilerin genel yapısını ifade edecek şekilde ideal öğrenmeyi gerçekleştirmiştir.



Şekil 2.11 (a) eksik öğrenme, (b) ideal öğrenme, (c) aşırı öğrenme

Eksik öğrenme problemi modelin yeterince derin bir yapıda olmaması, öğrenme döngüsünün yeterince uzun tutulmaması veya yeterince büyük bir veri seti kullanılmaması gibi birçok nedenden ötürü gerçekleşmiş olabilir (Naraan ve Tagliarini 2005). Aşırı öğrenme durumu da benzer şekilde modelin çok karmaşık bir yapıda kurulması, kullanılan veri seti içerisinde gürültü oluşturacak verilerin bulunması gibi nedenlerden gerçekleşmiş olabilir. Sinir ağlarında aşırı öğrenme durumunu engellemek için seyreltme(dropout) adı verilen bir yöntem bulunmaktadır ve literatürde sıklıkla kullanılmaktadır (Musiol, 2016). Şekil 2.12’de seyreltme işleminin sinir ağları üzerindeki örneği gösterilmiştir. Modelin eğitimi süresince uygulanan bu yöntemde, her eğitim döngüsünde modelin saklı katmanlarında bulunan nöronların bazıları rastgele olarak modelden çıkarılır. Böylelikle modelin sadece belirli nöronlar üzerinden gelişmesi engellenir.



Şekil 2.12 Seyreltme işlemi örneği

Şekil 2.12’de ilk örnekte her iki saklı katmandan ikinci ve üçüncü nöronlar çıkarılırken, diğer örnekte ilk saklı katmandan birinci ve üçüncü nöron ile ikinci saklı katmandan üçüncü ve dördüncü nöronlar modelden çıkarılmıştır. Bu nöronların seçimi tamamen rastgele yapılmaktadır.

2.3 Evrişimli Sinir Ağları

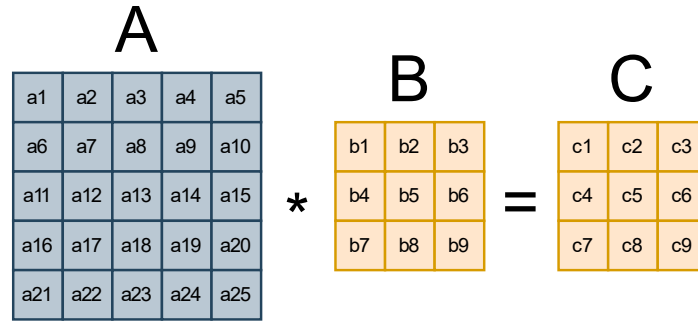
DNN yapıları içerisinde en popüler olan yapılardan birisi evrişimli sinir ağlarıdır (convolutional neural network – CNN). İsmi, içerdiği doğrusal matematiksel operasyon olan evrişim işleminden almaktadır. CNN yapıları evrişim katmanı, ortaklama katmanı ve tam bağlı katman (fully connected layer) bir diğer ismiyle yoğun katmanı (dense layer)

2.3.1 Evrişim katmanı

Giriş matrisi her bir filtre ile evrişim işlemi uygulanır. Eşitlik (2.14)'de 3x3 boyutunda A giriş matrisi ile 3x3 boyutundaki B filtre matrisinin evrişim işlemi verilmiştir.

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} * \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \sum_{i=1}^9 a_i b_i \quad (2.14)$$

Eşitlik (2.14)'de örnek olması açısından giriş matrisi ile filtre matrisinin boyları eşit kullanılmıştır. Ancak giriş matrisinin boyutu, filtre matrisinin boyutundan büyüktür. Bu durumda filtre matrisi, giriş matrisi üzerinde belirli adım sayıları ile kaydırılarak evrişim işlemi uygulanır. Şekil 2.15'te gösterilen örnekte B filtre matrisi, A giriş matrisi üzerinde birer adım kaydırılarak evrişim uygulanmış ve C matrisi elde edilmiştir. Eşitlik (2.15),(2.16) ve (2.17) ile bazı C matrisi elemanlarının hesaplanması gösterilmiştir.



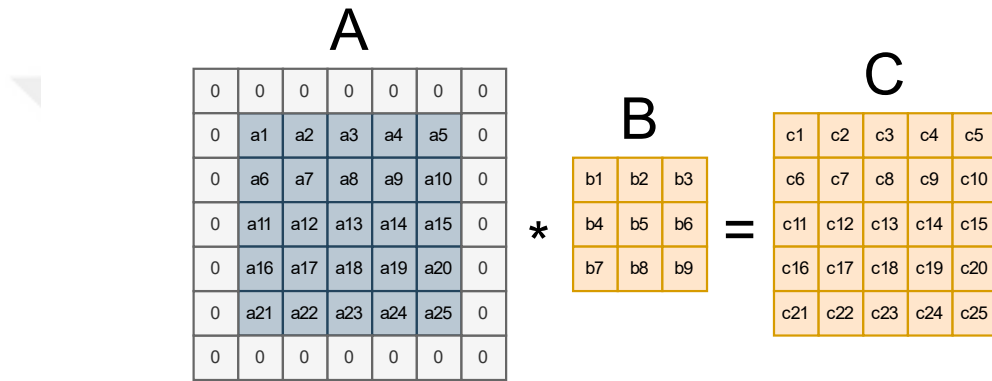
Şekil 2.15 Evrişim işlemi, adım sayısı = 1

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_6 & a_7 & a_8 \\ a_{11} & a_{12} & a_{13} \end{bmatrix} * \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = c_1 \quad (2.15)$$

$$\begin{bmatrix} a_2 & a_3 & a_4 \\ a_7 & a_8 & a_9 \\ a_{12} & a_{13} & a_{14} \end{bmatrix} * \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = c_2 \quad (2.16)$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} * \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = c_7 \quad (2.17)$$

Şekil 2.15’te görüldüğü üzere 5x5 boyutundaki giriş matrisi ile 3x3 boyutundaki filtre matrisine bir adım kayma miktarı ile evrişim işlemi uygulandığında 3x3 boyutunda bir matris elde edilmiştir. Bu durum giriş matrisinin çerçevesindeki noktalarda bulunan bilginin kaybolması ihtimalini ortaya çıkarmaktadır. Bu durumu önlemek için kullanılan sıfır ekleme(zero padding) adı verilen basit bir yöntem bulunmaktadır. Şekil 2.16’da giriş matrisine 0 satır ve sütunları eklenerek boyutu 7x7 gibi kullanılmış ve evrişim işlemi uygulanmıştır. Böylece giriş verisinin çerçevesinde bulunabilecek bilginin kaybolması engellenmektedir (Albawi vd. 2018).

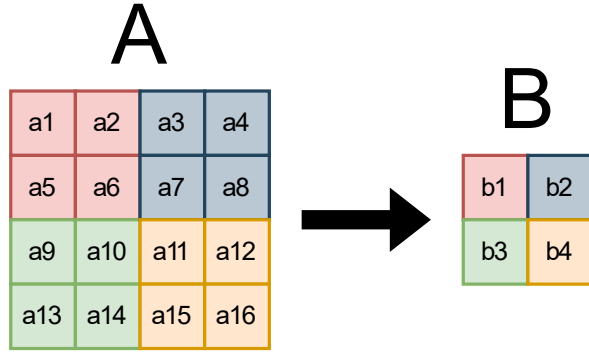


Şekil 2.16 Evrişim işlemi 0 ekleme, adım sayısı = 1

Evrişim katmanı kendi içerisinde aktivasyon fonksiyonlarını da içerir. Bölüm 2.2.3’te anlatılan aktivasyon fonksiyonları bu katman içerisinde de yer almaktadır.

2.3.2 Ortaklama katmanı

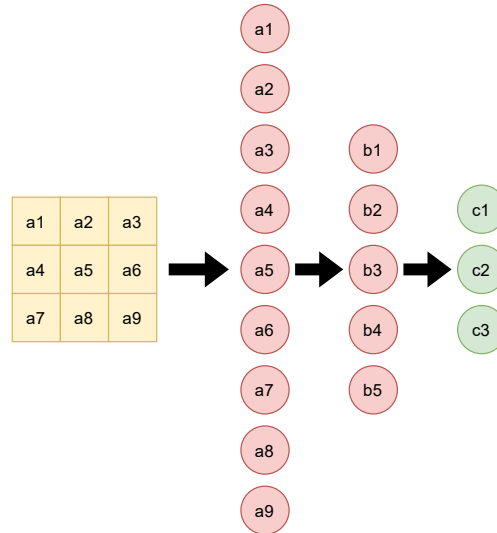
Ortaklama katmanının (pooling layer) temel amacı veri üzerinde alt örnekleme yaparak ardışık katmanlardaki karmaşıklığı azaltmaktır. Görüntü işleme problemleri için aynı görüntü üzerindeki çözünürlüğün düşürülmesi gibi kabul edilebilir. İki farklı ortaklama yöntemi bulunmaktadır. Bunlar maksimum ortaklama (max pooling) ve ortalama ortaklamadır (average pooling). Şekil 2.17’de 4x4 boyutundaki giriş matrisi, ortalama boyutu 2x2 ve adım sayısı 2 olacak şekilde ortaklama işleminden geçirilmiştir (Musiol 2016). Ortalama ortaklama işlemiyle b_1 değerinin hesaplanması için a_1, a_2, a_5 ve a_6 değerlerinin ortalaması alınır. Maksimum ortaklama işleminde ise bu değerlerden en büyüğü seçilir. Evrişim işleminde olduğu gibi adım sayısı kadar ilerlenir ve tekrar ortaklama işlemi uygulanır.



Şekil 2.17 Ortaklama 2x2, adım sayısı = 2

2.3.3 Tam bağlı katman

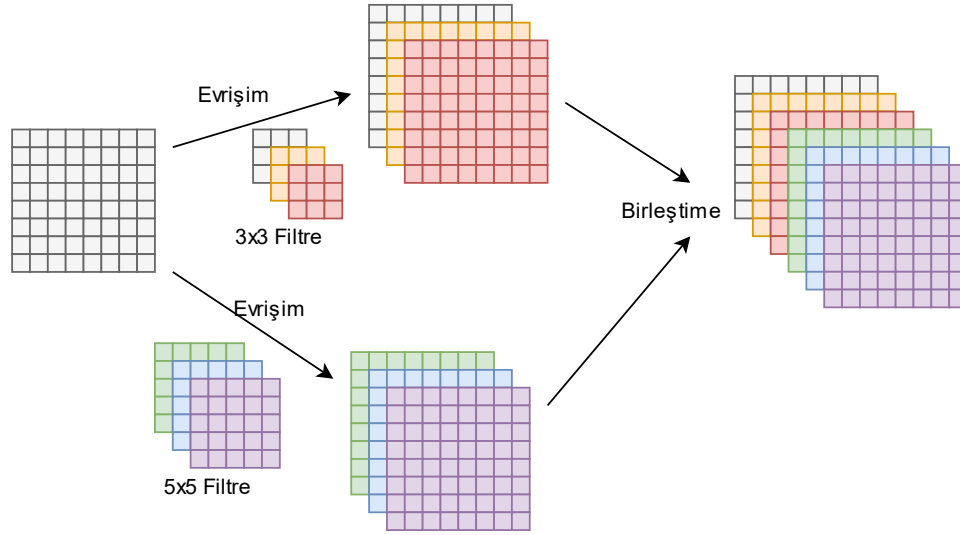
Art arda bağlanmış CNN katmanlarının arkasından tam bağlı katman (fully connected layer) gelmektedir. Bölüm 2.2.2’de anlatılan çok katmanlı algılayıcılar ile aynı yapıya sahiptir. Matris formundaki veri Şekil 2.18’de görüldüğü gibi düzleştirilmiş bir şekilde işleme alınır. Düzleştirilmiş A dizisi ile B dizisindeki nöronların her biri birbirine bağlıdır. Tam bağlı katman en çok parametre içeren katmandır. Bu nedenle çok fazla işlem yükü getirmektedir (Albawi vd. 2018).



Şekil 2.18 Tam bağlı katman

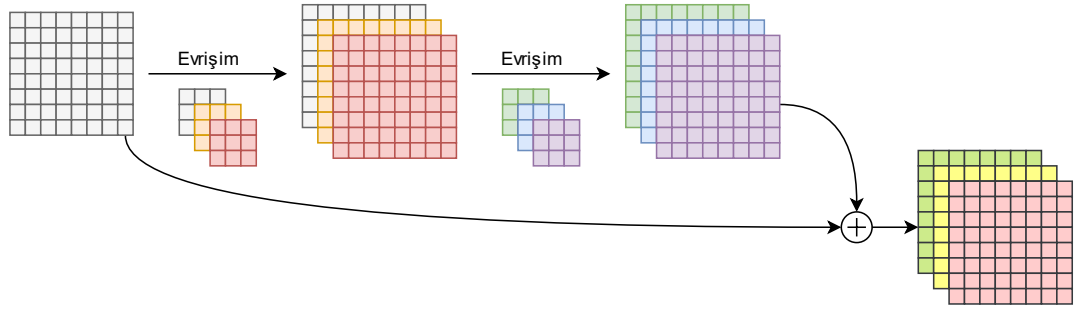
2.3.4 Özelleşmiş evrişimli sinir ağı yapıları

Literatürde CNN mimarisini temel alarak özelleşmiş farklı ağ yapıları bulunmaktadır. Bunlarda temel iki tanesi Inception ve ResNet isimleriyle literatürde geçmektedir. Şekil 2.19’da Inception modelinin literatürdeki yapısını örnekleyen bir yapı gösterilmiştir. Yapının özelliği farklı filtre boyutlarıyla evrişim işlemine giren blokların daha sonra birleştirilerek kullanmasıdır. Farklı filtreler kullanarak giriş verisi içerisindeki farklı bilgileri ortaya çıkarmaktır. Şekilde görüldüğü üzere 3x3 boyutundaki filtre matrisleriyle ve 5x5 boyutundaki filtre matrisleriyle ayrı ayrı evrişim işlemi uygulanmış, çıkan matrislerin birleşimi modelin arkasından gelecek bir sonraki katmana gönderilmiştir.



Şekil 2.19 Inception CNN yapısı

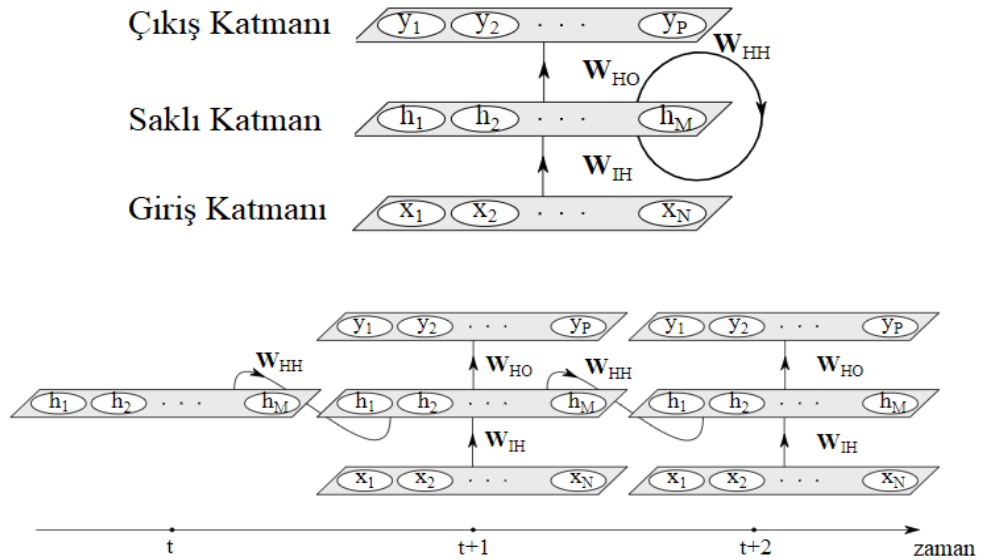
Şekil 2.20’de literatürdeki ResNet CNN yapısına bir örnek gösterilmiştir. Bu yapıda seçilen bazı katmanlar kısa yoldan ilerletilerek mimarinin içerisindeki bir başka katmana eklenmektedir. Mimarinin girişinde bulunan bilgilerin mimarinin derinliklerine gidildikçe kaybolması durumunda, kısa yoldan gelen bu katman mimarinin derinliklerinde tekrardan bilginin oluşturulmasını sağlar (Targ vd. 2016).



Şekil 2.20 ResNet CNN yapısı

2.4 Özyinelemeli Sinir Ağları

Özyinelemeli sinir ağları (recurrent neural network (RNN)) mimarisi daha önceden giriş olarak aldığı önemli verileri hatırlama yeteneğine sahiptir. RNN içerisinde bulundurduğu hafıza elemanları sayesinde sıralı gelen veri grupları üzerinde çalışmak için çok uygundur. Zaman serisi, konuşma, yazı, ses, hava durumu gibi veri gruplarında çalışma yapmak için idealdir.

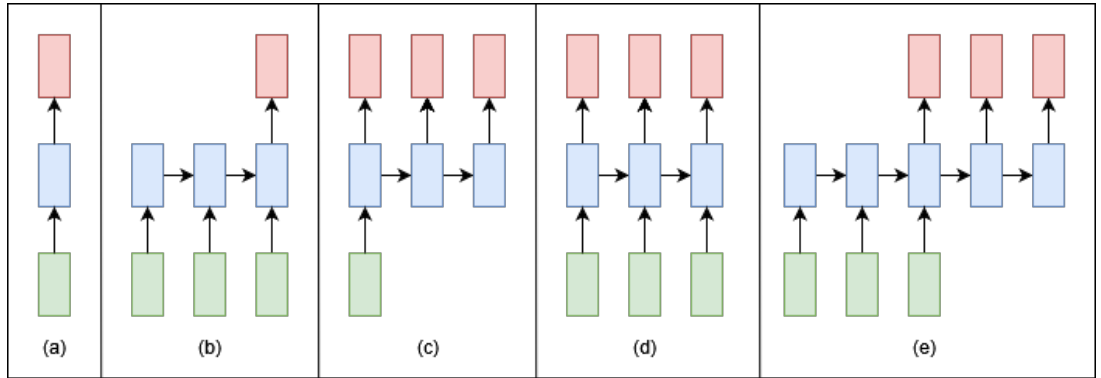


Şekil 2.21 Özyinelemeli sinir ağı genel yapısı

Şekil 2.21’de RNN mimarisinin genel yapısı gösterilmiştir (Salehinejad vd. 2017). RNN mimarisi de genel olarak 3 katmandan oluşur. Bunlar giriş, özyinelemeli saklı ve çıkış

katmanlarıdır. Şekil 2.21’de de görüldüğü üzere giriş katmanı N elemanlı diziden oluşmaktadır. Bu katmanın girişleri $\{\dots, x_{t-1}, x_t, x_{t+1}, \dots\}$ gibi zamana bağlıdır. Giriş katmanı ile saklı katman arasında W_{IH} ağırlıklandırma katsayıları bulunur. Saklı katman M elemanlı bir diziden oluşmaktadır ve saklı katmanlar zaman içerisinde birbirlerine bağlıdır. Birbirlerine bağlı saklı katmanlar arasında ise W_{HH} ağırlıklandırma katsayıları bulunmaktadır. Saklı katmanlar ayrıca bulundukları zaman diliminde çıkış katmanına da bağlıdır. Bu bağlantı arasında ise W_{HO} ağırlıklandırma katsayıları olduğu şekil 2.21’de görülmektedir (Salehinejad vd. 2017) Farklı problemlerin çözümü için özelleşmiş farklı RNN mimari yapıları bulunmaktadır. Bu mimari yapıları Şekil 2.22’de görsel olarak ifade edilmiştir. Bu farklı mimari yapıların kullanılacağı uygulamalara şu örnekler verilebilir (Alom vd. 2019):

- Bire Bir : Standart sınıflandırma problemi, görüntü sınıflandırma
- Çoktan Bire: Giriş olarak verilen bir cümleden, cümlemin olumlu mu yoksa olumsuz mu duygu içerdiğinin anlaşılması
- Birden Çoka: Giriş olarak tek bir görüntü verildiği, çıkış olarak ise görüntünün içeriğinin yorumlanması
- Çoktan Çoka: Giriş olarak alınan bir metnin bir dilden başka bir dile çevrilmesi
- Çoktan Çoka: Giriş olarak video çerçevelerinin alınıp, her çerçevenin etiketlenmesi



Şekil 2.22 Farklı uygulamalar için özelleşmiş RNN yapıları: (a) bire bir, (b) çoktan bire, (c) birden çoka, (d)(e) çoktan çoka

2.4.1 Özelleşmiş özyinelemeli sinir ağı yapıları

Literatürde RNN yapısını esas alan özelleşmiş ağ yapıları bulunmaktadır. Literatürde uzun-kısa süreli hafıza(long-short term memory – LSTM) ve kapılı tekrarlayan birim (gated recurrent unit – GRU) isimleriyle kullanılmaktadır. LSTM ve GRU mimarileri RNN yapısının kısa süreli hafızasını geliştirmek için sunulmuştur (Yang vd. 2020). Bu mimarilerin temelleri ve iç yapıları (Fu vd. 2016, Yang vd. 2020) çalışmalarıyla ayrıntılı olarak anlatılmaktadır.

2.5 Mimari Eğitimi ve Raporlanmasında Kullanılan Teknik Terimler

2.5.1 Sınıflandırma matrisi

Sınıflandırma işleminin sonuçlarının gösterildiği matristir. Karmaşıklık matrisi (confussion matrix) ve karışıklık matrisi şeklinde de kullanılmaktadır. Şekil 2.23'te bir sınıflandırma matrisi örneği verilmiştir. Örnek olarak isimleri A, B, C ve D olan 4 farklı grubun sınıflandırılması gösterilmektedir. Matrisin bir eksenini gerçekte olan sınıfı belirtirken, diğer eksenini yapılan test sonucunda tahmin edilen sınıfı belirtmektedir. Her sınıfa ait 100 veri teste sokulmuştur. Sisteme test girişi olarak verilen A sınıfına ait 100 verinin tamamı A olarak sınıflandırıldığı gösterilmektedir. C sınıfı olarak verilen verilerin ise 2 tanesi yanlış sınıflandırıldığı ve B olarak tahmin edildiği gösterilmiştir.

Gerçek Sınıf	A				
	B	1	97	1	1
	C		2	98	
	D	7			93
		A	B	C	D
		Tahmin Sınıfı			

Şekil 2.23 Sınıflandırma matrisi örneği

Bu matris üzerindeki değerleri kullanarak her sınıfın ne kadar başarılı bir şekilde sınıflandırıldığı ifade eden kesinlik (precision), duyarlılık (recall) ve F1 skoru değerleri

bulunmaktadır. Bu değerleri hesaplayabilmek için de gerçek pozitif (TP), yalancı pozitif (FP) ve yalancı negatif (FN) kavramlarına ihtiyaç duyulmaktadır.

Gerçek Pozitif : Bir durumun varlığının doğru olarak belirlenmesidir. Şekil 2.23’de verilen örnek için B sınıfına ait 100 örneğin 97 tanesi B olarak tahmin edilmiştir. Gerçek pozitif sayısı B sınıfı için 97 olmuştur.

Yalancı Pozitif : Bir durumun varlığının yanlış olarak belirlenmesidir. B sınıfına ait 100 örneğin 3 tanesi B sınıfı dışında olarak tahmin edilmiştir. B sınıfı için yalancı pozitif sayısı 3 olmuştur.

Yalancı Negatif : Bir durumun var olmayışının yanlış olarak belirlenmesidir. B sınıfına ait olmayan 2 örnek, B sınıfı olarak tahmin edilmiştir. B sınıfı için yalancı negatif sayısı 2 olmuştur.

Kesinlik: Gerçek pozitiflerin, gerçek pozitifler ve yalancı pozitiflerin toplamına oranıdır.

Duyarlılık : Gerçek pozitiflerin, gerçek pozitifler ve yalancı negatiflerin toplamına oranıdır.

F1-skoru : kesinlik ve duyarlılık değerlerine göre hesaplanır. Eşitlik (2.18)’de verilmiştir. Sınıflandırma başarımıyla ilgili yorum yapmak için en faydalı bilgi F1 skorudur.

$$F1 = 2 \times \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (2.18)$$

Şekil 2.23’te verilen örnek için bu değerler şu şekilde hesaplanabilir: A sınıfı için TP sayısını 100, FP sayısının ise 0 olduğu ve FN sayısının da 8 olduğu görülmektedir. A sınıfının sınıflandırılma kesinliği $\frac{100}{100+0} = 1.0$ ve duyarlılığı $\frac{100}{100+1+7} = 0.93$ olarak hesaplanabilir. Hesaplanan değerler eşitlik (2.18)’de yerine konulduğunda F1 skoru 0.964 olarak hesaplanmaktadır. Bütün sınıflar için bu hesaplamalar yapılmıştır ve çizelge 2.1’de verilmiştir. A sınıfı kesinlik olarak en yüksek değeri elde etse de F1 skoru olarak üçüncü sıradadır. C sınıfı ise 0.98 kesinliğe sahip olmasına rağmen, duyarlılık değeri de yüksek olduğu için en yüksek F1 skorunu elde etmiştir. F1 skoru, sınıflandırma başarımını karşılaştırmak için daha uygun bir parametredir. Kesinlik ve duyarlılık değerleri ile sınıflandırma başarımı hakkında doğrudan bir yorum yapılamaz. Kesinlik değerinin çok yüksek ve duyarlılık değerinin çok düşük olduğu veya kesinlik değerinin çok düşük ve duyarlılık değerinin çok yüksek olduğu durumlar oluşabilir. F1 skorunun 1 olması %100 başarımla ifade eder.

Çizelge 2.1 Şekil 2.27’de verilen örnek sınıflandırma tablosu için performans tablosu

	Kesinlik	Duyarlılık	F1-skoru
A	1.0	0.93	0.964
B	0.97	0.98	0.975
C	0.98	0.99	0.985
D	0.93	0.99	0.959

2.5.2 Epok (epoch)

Bir veri setinin tamamının 1 kez mimariden geçirilmesi ve eğitim parametrelerinin güncellenmesi 1 epok olarak tanımlanır. Epok sayısı eğitimin ne zaman tamamlanacağını belirtir. Epok sayısı 100 olarak tanımlanmışsa, veri setinin tamamı 100 kere mimariye verilir ve her turda parametreler güncellenir. 100 turun sonunda eğitim sonlandırılır.

2.5.3 Grup boyutu

Grup boyutu (batch size) mimarinin başarımlı oranı ve çalışma süresini etkileyen önemli parametrelerden birisidir. Veri seti küçük parçalara bölünerek mimariye verilebilir. Bu parçaların boyutuna grup boyutu denir. İçerisinde 100 veri bulunan bir veri seti için grup boyutu 20 olarak tanımlanmışsa. Öncelikle 20 verilik grup mimariye verilir ve parametre güncellemesi yapılır. 5 tur sonunda veri setinin tamamı 1 kez mimariye verilmiş ve 1 epok tamamlanmış olunur. Epok sayısı kadar döngü devam eder. (Radiuk 2018) çalışmasında grup boyutunun görüntü sınıflandırma problemi için başarımlı oranı ve çalışma süresini nasıl etkilediğini incelemiştir. Grup boyutunun artırılması sınıflandırma başarımlı oranını artırırken, mimarinin öğrenme süresini uzattığını raporlamıştır.

2.5.4 Grup normalizasyonu

Grup normalizasyonu (batch normalization – BN) ağırları daha hızlı eğitmek ve daha yüksek doğruluk oranı sağlamak için kullanılan en popüler yöntemlerden birisidir. İlk olarak (Ioffe ve Szegedy 2015) tarafından önerilmiştir. BN işleyişi şu şekilde açıklanabilir: Eğitilecek parametrelerin güncellenmesi kayıp fonksiyonu ve gradyan inişi

ile yapıldığı bölüm 2.2.5'te anlatılmıştı. Bu güncelleme her katman için ayrı ayrı yapılmaktadır. Bir katmanın çıkışı arkasında gelen katmanın girişi olduğu için, öndeki katmanlardan birinde yapılan parametre güncellemesi arkada kalan katmanları da etkileyecektir. Arkadaki katmanlarda daha önceden eğitilmiş parametrelerin tekrar eğitilmesi ihtiyacı ortaya çıkacaktır ve eğitim süresini uzatacaktır. Bu nedenle her katmanın çıkışını belirli bir aralıkta ve dağılımda limitleyecek normalizasyon işlemi yapılmaktadır. Öndeki katmanlarda yapılan parametre güncellemesinin ardından normalizasyon işlemi gerçekleşeceği için bir arkadan gelen katmanın girişlerinde büyük değişiklikler gözlenmeyecektir. Literatürde BN uygulamasının etkilerini inceleyen çalışmalar bulunmaktadır. (Thakkar vd. 2018) çalışmasında görüntü sınıflandırma problemi için BN uygulamasının etkisini incelemiştir. BN kullanımının özellikle derin sinir ağlarında başarı oranını önemli ölçüde arttırdığını raporlamıştır.

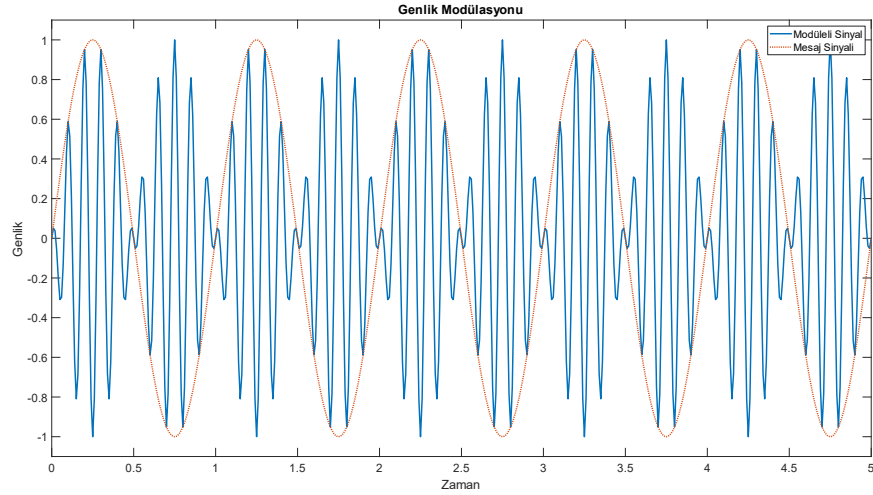
3. MODÜLASYON TEKNİKLERİ

Bu bölümde modülasyon sınıflandırma problemi içerisinde birbirlerinden ayrıştırılmaya çalışılan modülasyon tipleri yıldız küme şemaları veya modüleli sinyal şekilleri üzerinden gösterilmiştir.

3.1 Analog Modülasyon Türleri

3.1.1 Genlik modülasyonu

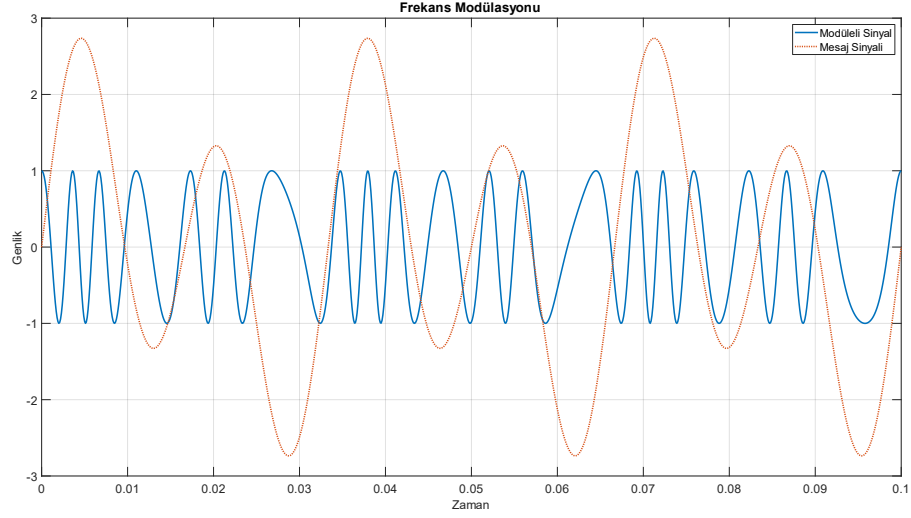
Taşıyıcı sinyalin genliğinin, gönderilmek istenen mesajın genliğine göre değiştiği modülasyon türüne genlik modülasyonu (amplitude modulation -AM) denir. Şekil 3.1’de iletilecek mesaj sinyali ve bu sinyalin genlik modüleli görüntüleri verilmiştir.



Şekil 3.1 Genlik modülasyonu

3.1.2 Frekans modülasyonu

Taşıyıcı sinyalin frekansının gönderilmek istenen sinyalin genliğine göre değiştiği modülasyon türüne frekans modülasyonu(frequency modulation – FM) denir. Şekil 3.2’de iletilecek bir mesaj sinyali ve bu sinyalin frekans modüleli hali gösterilmiştir.



Şekil 3.2 Frekans modülasyonu

3.2 Sayısal Modülasyon Türleri

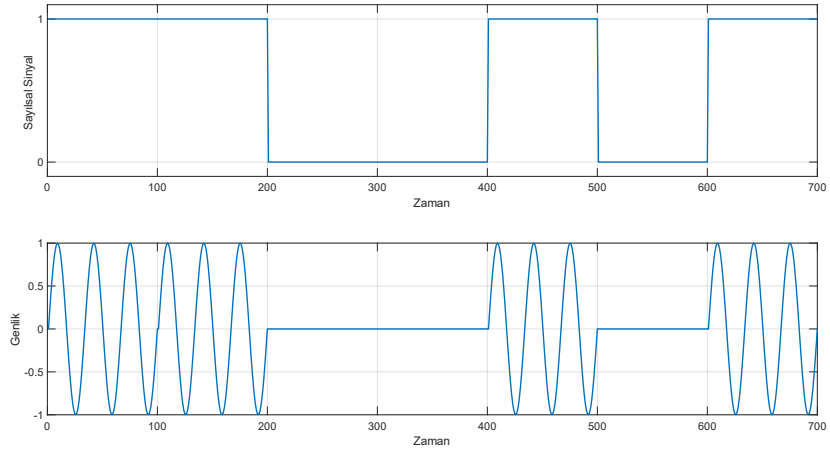
3.2.1 Genlik kaydırmalı anahtarlama

İletilecek sayısal mesaj, sabit frekanslı referans sinyalinin genliği değiştirilerek ifade edilir. Bir genlik kaydırmalı anahtarlama (amplitude shift keying - ASK) sembolü ile kaç bitlik verinin iletebileceğine göre şekillenmiş birden fazla ASK yöntemi bulunmaktadır. Bir ASK sembolü ile kaç bitlik verinin ifade edileceğine göre şekillenmiş farklı ASK tipleri mevcuttur.

- Dur -Başla Anahtarlama (On-Off Keying - OOK)
- Dörtlü genlik kaydırmalı anahtarlama (4 Amplitude Shift Keying - 4ASK)
- Sekizli genlik kaydırmalı anahtarlama (8 Amplitude Shift Keying - 8ASK)

3.2.1.1 Dur-başla anahtarlama

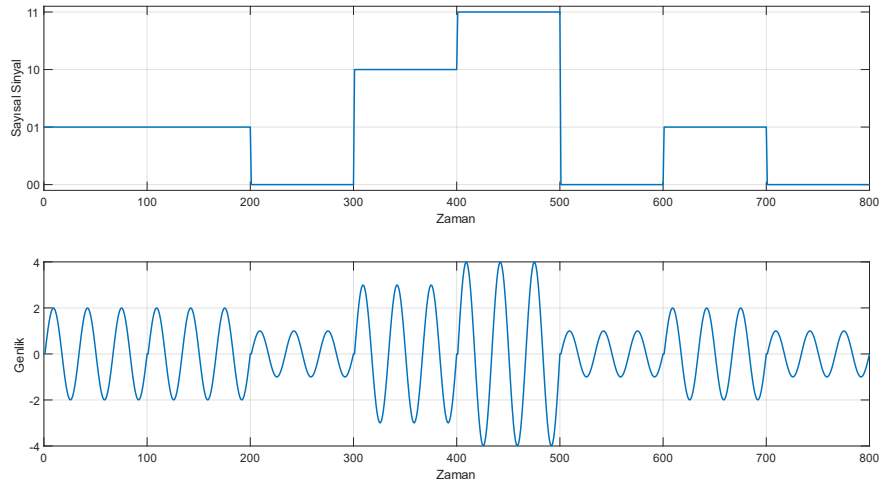
Şekil 3.3'te OOK modülasyonu ile iletilen sayısal işaret ve bu işaretin modüle edilmiş biçimi gösterilmiştir. Sayısal olarak 1 sembolünü göndermek için sabit genlikli bir sinüs işareti gönderilirken, 0 sembolünü göndermek için işaret susturulmuştur.



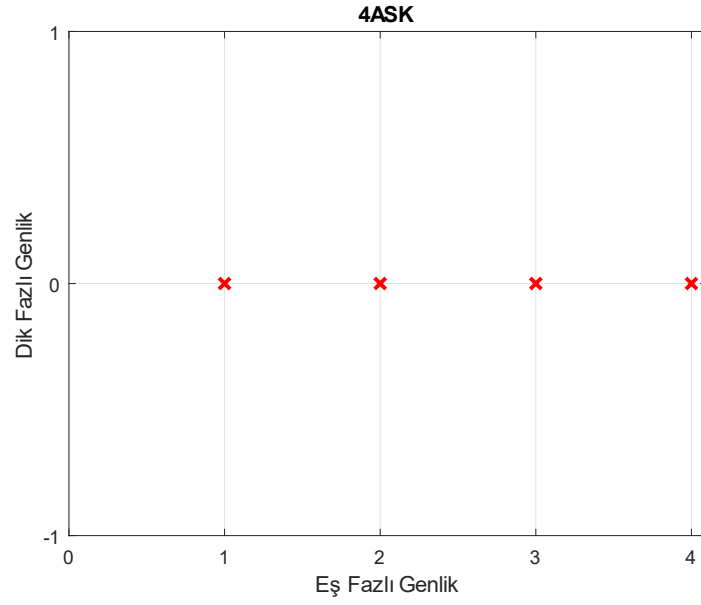
Şekil 3.3 OOK Modülasyonu

3.2.1.2 Dörtlü genlik kaydırmalı anahtarlama

Şekil 3.4’te görüldüğü üzere dört farklı genlik seviyesine sahip sinüs sinyalleri bulunmaktadır. Bir ASK sembolü ile iki bitlik veri iletimi yapılabilmektedir. Her farklı genlik seviyesi iki bitlik veriyi temsil eder. Sinyaller arasında bir faz farkı yoktur. Şekil 3.5 ile verilen yıldız kümesi şeması da sinyaller arasında faz farkı olmadığını, sadece genliklerinin farklı olduğunu ifade etmektedir.



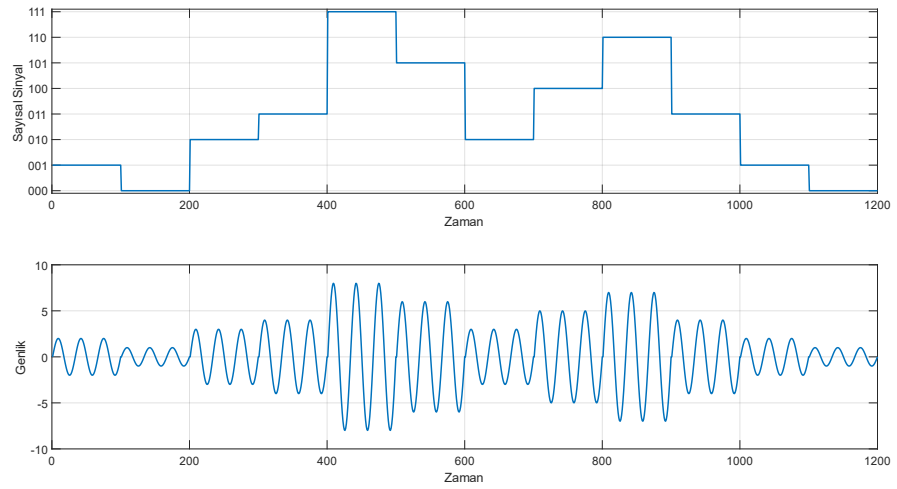
Şekil 3.4 4ASK Modülasyonu



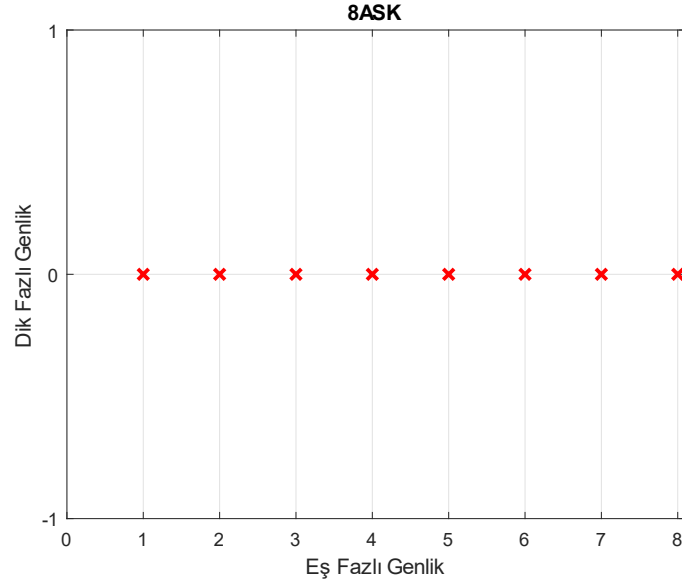
Şekil 3.5 4ASK yıldız kümesi şeması

3.2.1.3 Sekizli genlik kaydırmalı anahtarlama

Şekil 3.6’da görüldüğü üzere sekiz farklı genlik seviyesine sahip sinüs sinyalleri bulunmaktadır. Sekiz farklı genlik seviyesi ile her genlik seviyesi üç bitlik veri ifade edebilmektedir. 4ASK modülasyonunda da olduğu gibi bu sinyaller arasında da faz farkı yoktur. Şekil 3.7 ile sembollerin yıldız kümesi şeması üzerinde dağılımı ve semboller arasında faz farkı olmadığı gösterilmiştir.



Şekil 3.6 8ASK Modülasyonu



Şekil 3.7 8ASK yıldız kümesi şeması

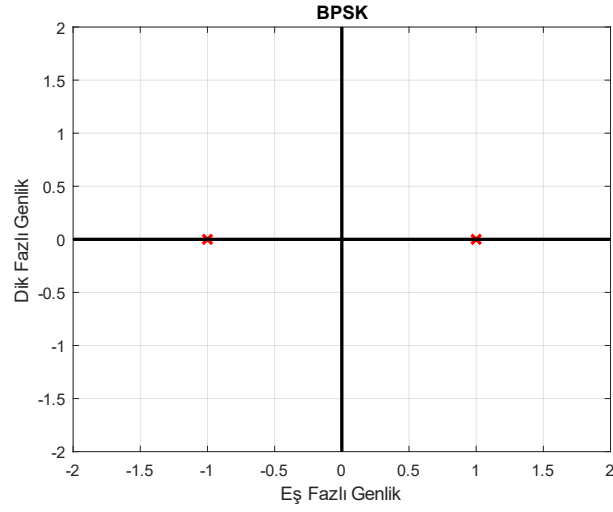
3.2.2 Faz kaydırmalı anahtarlama

İletilecek sayısal mesaj, sabit frekanslı referans sinyalinin fazı değiştirilerek ifade edilir. Bir sembol ile kaç bitlik sayısal verinin iletebileceğine göre şekillenmiş birden fazla faz kaydırmalı anahtarlama (phase shift keying – PSK) yöntemi bulunmaktadır.

- İkili (Binary) faz kaydırmalı anahtarlama (BPSK)
- Karesel (Quadrature) faz kaydırmalı anahtarlama (QPSK)
- Sekizli, on altılı ve otuz ikili faz kaydırmalı anahtarlama (8PSK, 16PSK, 32PSK)

3.2.2.1 İkili faz kaydırmalı anahtarlama

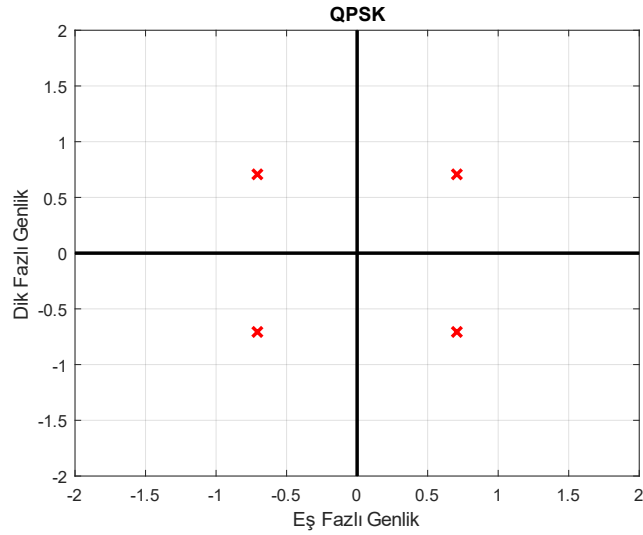
Bir BPSK sembolü ile 0 veya 1 sayısal mesajları iletebilir. Şekil 3.8’de ideal BPSK sembolü yıldız kümesi şeması verilmiştir. İletilen her sembolün genliği birbirine eşittir. Semboller arasında sadece 180 derece faz farkı bulunmaktadır.



Şekil 3.8 BPSK sembolü yıldız kümesi şeması

3.2.2.2 Karesel faz kaydırmalı anahtarlama

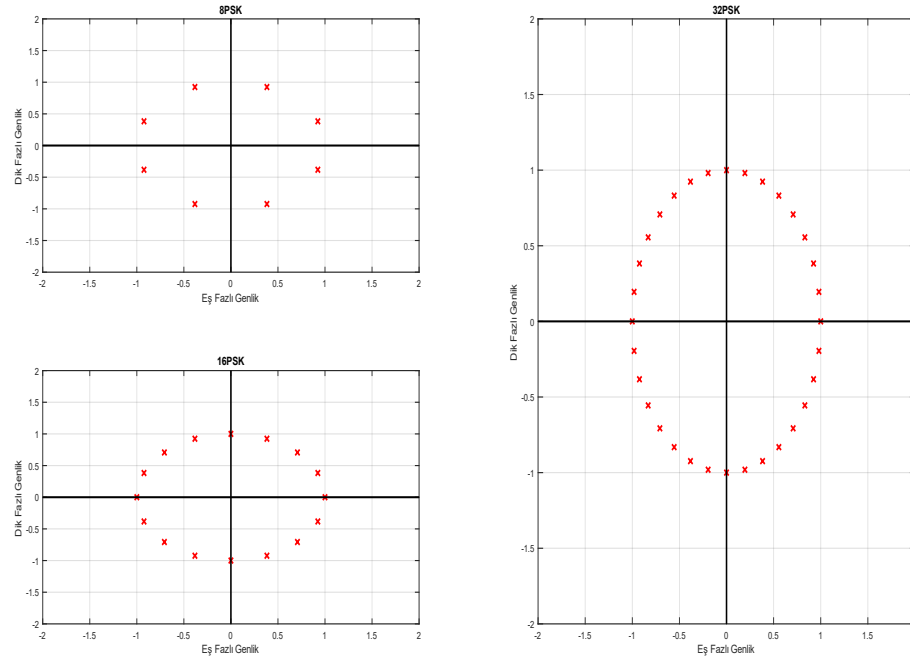
Bir QPSK sembolü ile 00,01,10 veya 11 sayısal mesajı iletilebilir. Semboller arasında 90 derecelik faz farkı bulunmaktadır. Şekil 3.9’da da bu faz farkı gözükmemektedir. BPSK’de olduğu gibi QPSK’de de her sembolün genliği birbirine eşittir.



Şekil 3.9 QPSK sembolü yıldız kümesi şeması

3.2.2.3 8-16-32 Faz kaydırmalı anahtarlama

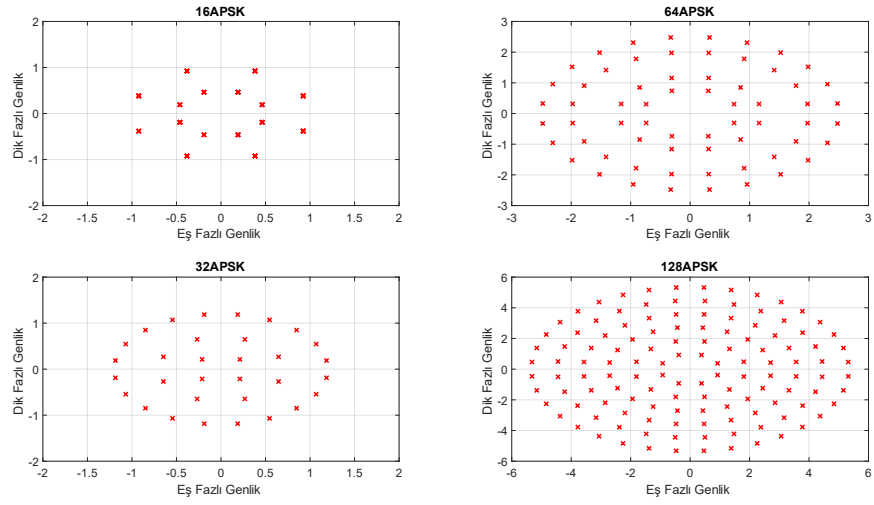
Bir 8PSK sembolü ile üç bitlik, 16PSK sembolü ile dört bitlik ve 32PSK sembolü ile beş bitlik veri taşınabilir. Modülasyon derecesi büyüdükçe semboller arasındaki faz farkı



Şekil 3.10 8PSK 16PSK 32PSK sembolleri yıldız kümesi şeması

3.2.3 Genlik ve faz kaydırmalı anahtarlama

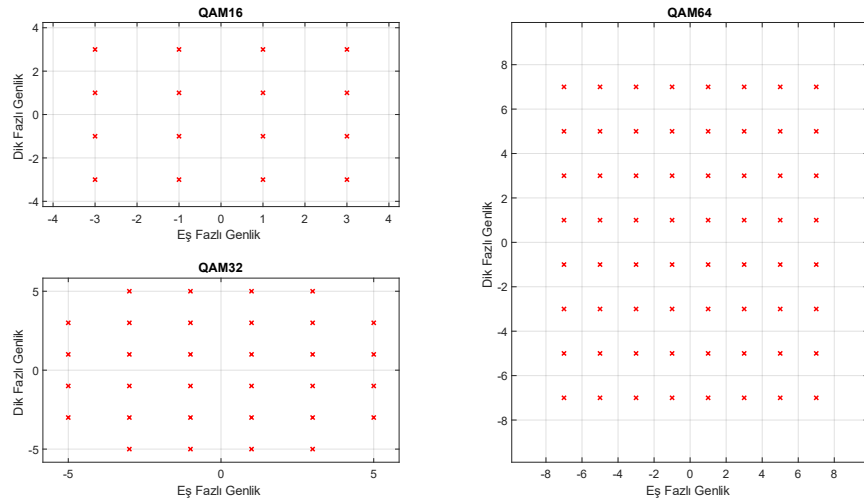
Genlik ve faz kaydırmalı anahtarlama(amplitude-phase shift keying – APSK), ASK ve PSK modülasyon türlerinin birlikte kullanılmış şeklidir. Semboller arasında hem faz hem de genlik farkı bulunmaktadır. ASK ve PSK’da olduğu gibi bir sembol ile kaç bitlik verinin taşınabileceğine göre özelleşmiş farklı APSK seçenekleri mevcuttur. Şekil 3.11’de görüldüğü üzere, semboller değiştikçe sinyallerin hem fazı hem de genlikleri değişmektedir.



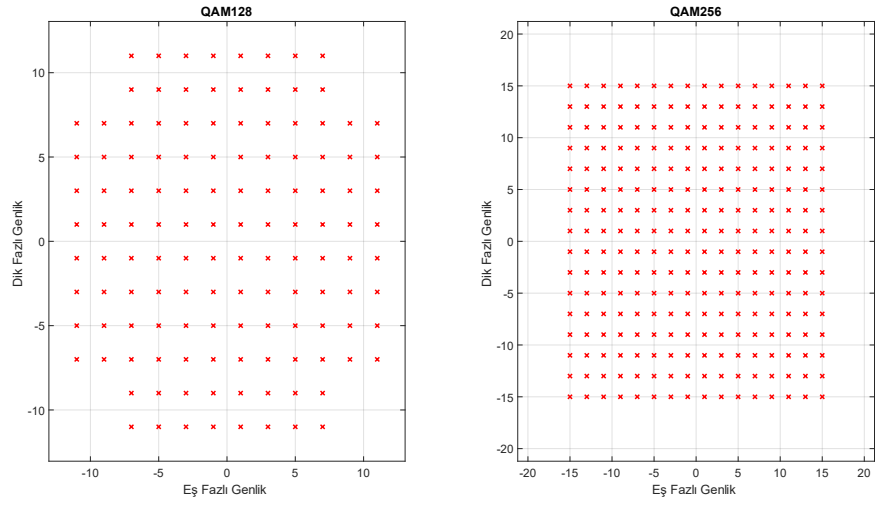
Şekil 3.11 16APSK, 32APSK, 64APSK, 128APSK sembollerini yıldız kümesi şeması

3.2.4 Karesel genlik modülasyonu

Karesel genlik modülasyonu (quadrature amplitude modulation – QAM), APSK modülasyonuna çok benzer bir yapıya sahiptir (Baldi vd. 2012). Sembollerin hem fazları hem de genlikleri birbirinden farklıdır. Bir sembol ile iletilebilecek maksimum bit sayısına göre gruplanmış farklı QAM tipleri bulunmaktadır. Şekil 3.12 ve Şekil 3.13'te yıldız kümesi şemaları verilmiştir.



Şekil 3.12 QAM16, QAM32, QAM64 yıldız kümesi şeması



Şekil 3.13 QAM128, QAM256 yıldız kümesi şeması

4. ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ

Alanda programlanabilir kapı dizileri (field programmable gate array - FPGA) içerisinde yapılandırılabilir mantık blokları(configurable logic blocks - CLB), blok rastgele erişimli bellek (block random access memory - BRAM) ve sayısal işaret işlemcisi(digital signal processor - DSP) bulunduran yarı iletken devre elemanlarıdır (Amagasaki vd. 2018). İçlerinde bulunan bu yapılar donanım tanımlama dilleriyle (hardware description language - HDL) programlanarak farklı uygulamalar için kullanılmaktadır. CLB yapıları içerisinde aramalı tablo (look-up table - LUT), iki durumlu devre (flip flop - FF) ve çoklayıcı (multiplexer - MUX) elemanları bulunur. CLB blokları temel mantık operasyonları, toplama, çıkarma işlemleri ve sayaç gibi operasyonların gerçekleştirilmesi için kullanılır. DSP'ler ile çarpma işlemi gerçekleştirilebilir. FF ile saat darbeleri senkron devreler kurulur. BRAM ise hafıza elemanı olarak kullanılır. FPGA kullanımının en büyük avantajı paralel işlem kabiliyetidir. FPGA içerisindeki bloklar saat darbeleri ile senkron olacak şekilde aynı anda işlem yapabilir. Bir FPGA içerisinde aynı anda yüzlerce toplama, çıkarma ve çarpma işlemi yapılabilir.

4.1 FPGA Çalışmalarında Nicemleme

Her FPGA içerisinde limitli sayıda CLB, BRAM ve DSP yapısı bulunmaktadır. Bu da yapılan tasarımın kaynak kullanımı açısından dikkatli planlanmasını gerektirmektedir. Yazılım dillerinde tam sayılar genellikle 32-bit olarak saklanır. Kayan noktalı (floating point) bir sayıyı ifade etmek için ise 32-bit veya 64-bit veri saklanması gerekmektedir. Bunların 32-bit veya 64-bit olarak ifade edilmesi yazılım projelerinin bir çoğu için problem yaratmazken, donanım tanımlama dilleri ile FPGA için tasarım yaparken kritik hale gelmektedir. Kullanılacak sayılar, yapılacak işlemler minimum bit sayısı ile ifade edilecek şekilde ayarlanmalıdır.

Çizelge 4.1 π ve e sayılarının 64-32-16 ve 8-bit nicemlenmiş ifadesi

	64-bit	32-bit	16-bit	8-bit
π	3.141592653589793	3.141592653468251	3.1416015625	3.15625
e	2.718281828459046	2.718281827867031	2.71826171875	2.71875

Çizelge 4.1'de π ve e sayılarının 64-32-16 ve 8 bitlik sabit noktalı (fixed point) olarak ifade edilecek şekilde nicemlenmiş halleri verilmiştir. Daha çok bit ile ifade etmek sayının daha hassas bir çözünürlük ile kullanılmasını sağlamaktadır. Ancak FPGA üzerinde

tasarım yaparken kaynak kullanımı açısından olumsuz etki yapmaktadır. Donanım tasarımcısının bu noktada ilgilenilen probleme göre hem kaynak kullanımı minimumda tutacak hem de yapacağı çalışmanın sonucu etkilemeyecek en uygun çözünürlük ile çalışması gerekmektedir. Literatürde FPGA üzerinde nicemleme hatasını ve donanımın kaynak kullanımını optimize etmeye çalışan çalışmalar bulunmaktadır (Roy ve Banarjee 2005).

4.2 FPGA Üzerinde Matematik İşlemlerinin Kaynak Kullanımı

Çizelge 4.2’de iki sayısının çarpımının FPGA üzerinde ne kadar kaynak tüketeceği gösterilmiştir. Çarpılacak sayıların bit uzunluğuna karşılık tüketeceği DSP sayısı raporlanmıştır.

Çizelge 4.2 Çarpma işleminde kaynak kullanımı

A Sayısı Bit Uzunluğu	B Sayısı Bit Uzunluğu	Kullanılan DSP Sayısı
64	64	16
64	32	8
32	32	4
32	16	2
16	16	1
8	8	1

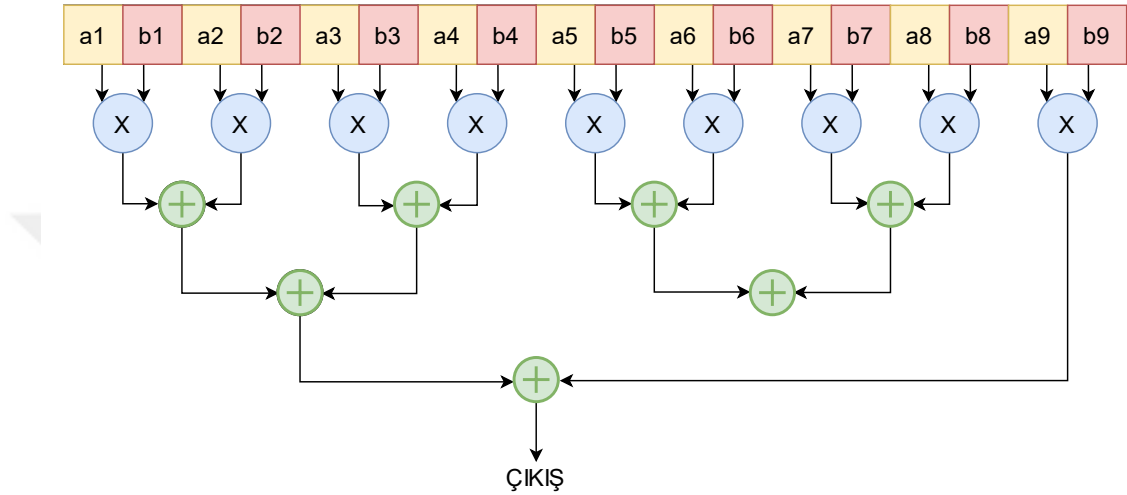
Çizelge 4.3’te iki sayısının toplamının FPGA üzerinde ne kadar kaynak tüketeceği gösterilmiştir. Toplanacak sayıların bit uzunluğuna karşılık tüketeceği LUT sayısı raporlanmıştır.

Çizelge 4.3 Toplama işleminde kaynak kullanımı

A Sayısı Bit Uzunluğu	B Sayısı Bit Uzunluğu	Kullanılan LUT Sayısı
64	64	32
64	32	32
32	32	16
32	16	16
16	16	8
16	8	8
8	8	4

Eşitlik (2.12)’de verilen evrişim işleminde 9 çarpma işlemi yapılması ve çıkan sonuçların toplanması istenmiştir. Bu işlemlerin her birinin 32-bit sayılar kullanılarak yapılması durumunda yaklaşık olarak 36 DSP ve 128 LUT kullanılacağı öngörülmektedir. Aynı

işlem 8-bit olarak yapılması durumunda ise yaklaşık 9 DSP ve 32 LUT kullanılacaktır. Bu kaynak kullanım tahminleri devrenin gerçekleştirilmesi durumunda değişiklik gösterebilir. Donanım tanımlama dilinde yazılmış kodları, FPGA içerisine yerleştirilebilir hale getiren sentezleyiciler bazı optimizasyon yöntemleri uygulayarak kaynak kullanımında farklılık yaratabilirler. Şekil 4.1’de eşitlik (2.14) ile verilen evrişim işlemini FPGA üzerinde kurmak için çizilmiş bir blok şema verilmiştir.



Şekil 4.1 Evrişim işleminin FPGA için blok şeması

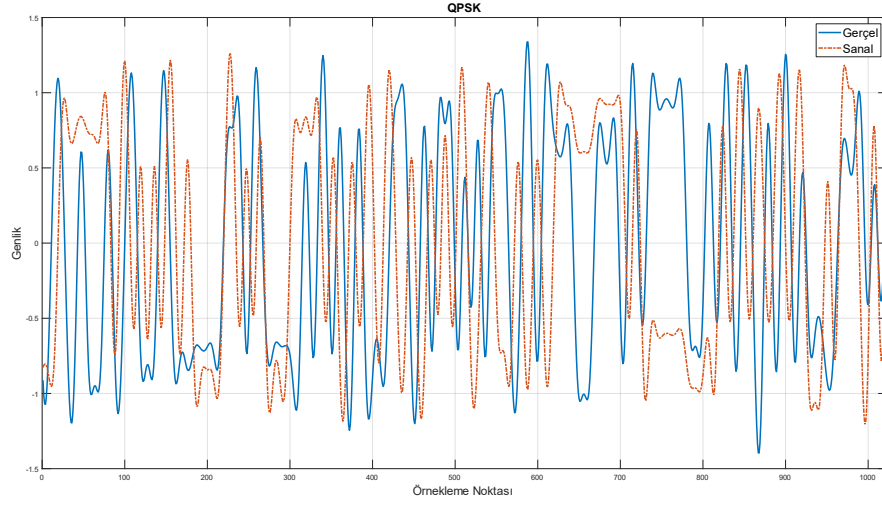
5. YAPILAN ÇALIŞMALAR

5.1 Modülasyon Sinyalleri Veri Seti

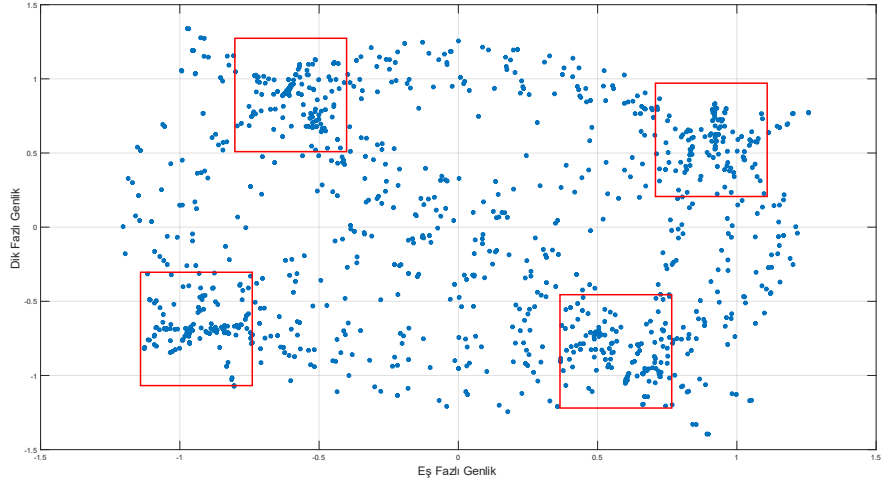
Derin öğrenme uygulamalarında veri seti, tasarlanacak modelin eğitilebilmesi açısından çok önemlidir. Bu nedenle literatürde bir çok makalede kullanılan açık kaynak bir veri setinden faydalanılmıştır (O'Shea vd. 2018). Bu veri seti içerisinde OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK ve OQPSK modülasyon türlerine ait sinyaller vardır. Her modülasyon türü için -30dB'den 20dB'ye kadar SNR seviyesine sahip kayıtlar alınmıştır. Her SNR seviyesi ve modülasyon tipine karşılık 4096 adet veri kaydı bulunmaktadır. Toplamda yaklaşık 2.5 milyon örnek veri bulunmaktadır.

Veri seti içerisindeki sinyallere frekans kayıklığı, faz kayıklığı, çok yollu sönmelenme gibi rastgele olarak değişen bozucu etkiler uygulanmıştır. Sayısal modüledi sinyaller ayrıca kök yükseltilmiş kosinüs darbe şekillendirme filtresinden (root-raised cosine pulse shaping filter) geçirilmiştir. Veri seti içerisindeki her bir kayıt 1024 örnekleme noktasından oluşmaktadır.

Şekil 5.1'de veri setinden alınmış QPSK modülasyonuna ait 30dB SNR seviyesine sahip bir sinyal gösterilmektedir. Sinyalin gerçel ve sanal kısımları çizdirilmiştir. Sadece sinyale bakarak modülasyon türüyle ilgili yorum yapmak mümkün değildir. Bu nedenle Şekil 5.2'de sinyalin yıldız kümesi şeması çizdirilmiştir. Kare içerisine alınmış bölgelerde sembollere ait bir gruplanma olduğu düşünülebilir fakat ideal QPSK şemasından çok uzak bir görünüme sahiptir. Bütün sembollerin ayrıca bir çember oluşturmaya çalıştığı da görülmektedir. Bu çember de sinyal büyüklüğünün neredeyse sabit olduğunu söyleyebilir ancak çemberin iç bölgesinde bulunan noktalar QPSK sembollerinin yıldız kümesi şeması ile uyumlu değildir. Veri seti tekrar incelendiğinde her bir QPSK sembolünün birden fazla örnekleme noktası ile örneklendiği düşünülmektedir.

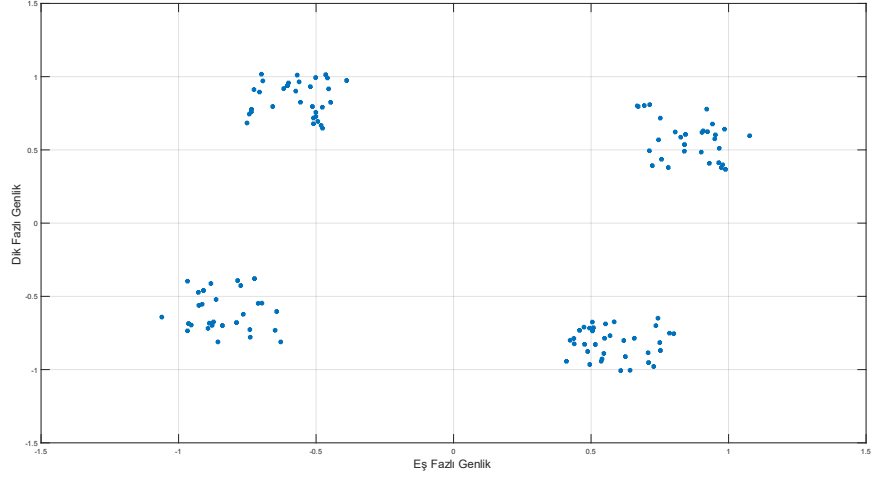


Şekil 5.1 QPSK modülasyonu, 30dB SNR



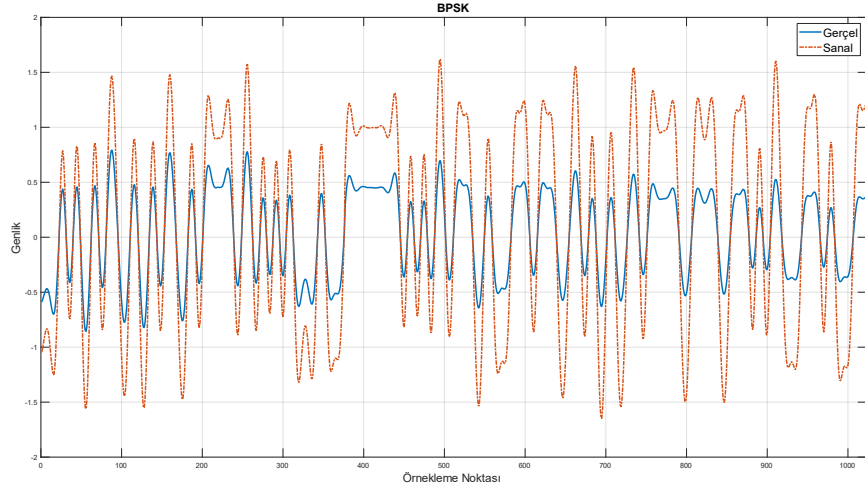
Şekil 5.2 QPSK sinyali yıldız kümesi şeması

Şekil 5.3'te QPSK sinyalinin 8'de 1 olacak şekilde alt örneklenmiş halinin yıldız kümesi şeması verilmiştir. İdeal QPSK şemasına benzer bir görünüm ortaya çıkmıştır. Şekil 5.2'de bulunan anlamsız noktalar kaybolmuştur. Birbirleri arasında yaklaşık 90° faz farkı bulunan ve genlikleri birbirlerine yaklaşık eşit olan gruplar gözlemlenmiştir.

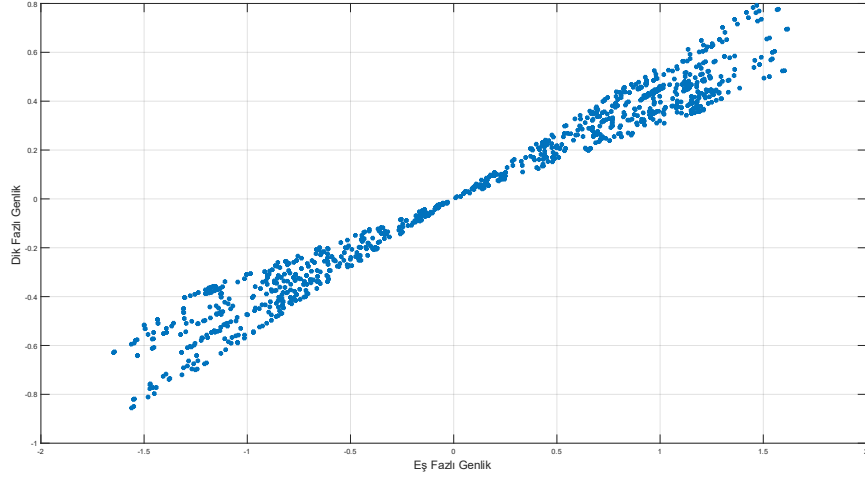


Şekil 5.3 QPSK sinyali alt-örneklelenmiş yıldız kümesi şeması

Şekil 5.4'te BPSK modülasyonuna ait 30dB SNR seviyesindeki bir sinyal verilmiştir. Doğrudan sinyali inceleyerek modülasyon türüne ait bir yorum yapmak mümkün olmadığı için Şekil 5.5'te sinyale ait yıldız kümesi şeması çizdirilmiştir. Genlik seviyesi ve fazı değişen bir sinyal var gibi gözükmemektedir fakat bu durum BPSK şeması ile uyumlu değildir.

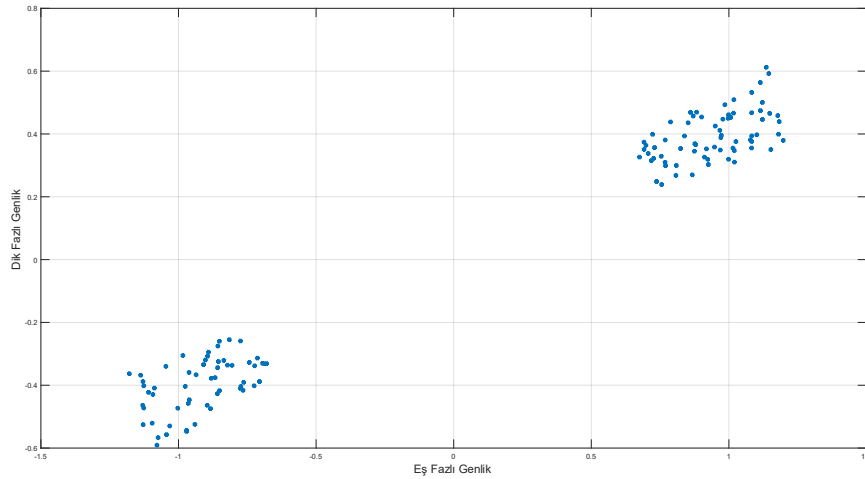


Şekil 5.4 BPSK modülasyonu, 30dB SNR



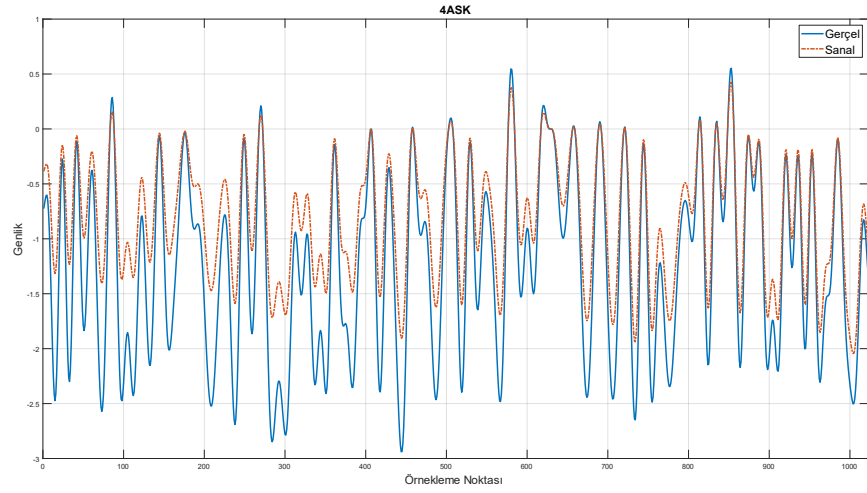
Şekil 5.5 BPSK sinyali yıldız kümesi şeması

Şekil 5.6'da 8'de 1 olacak şekilde alt örneklenmiş BPSK sinyali yıldız kümesi şeması verilmiştir. Aralarında 180° faz farkı bulunan iki grup oluşmuştur. Bu grupların ortalama genlikleri birbirine eşit gözükmemektedir. İdeal BPSK şemasına benzemeye başlamıştır. İdeal şemada sembol gruplarının fazları 0° ve 180° olarak tanımlanırken Şekil 5.6'da bir sembol grubunun fazı yaklaşık 20° , diğeri ise yaklaşık 200° 'dir. Sinyallere uygulanan faz kayıklığı bozucu etkisi kendisini göstermektedir.

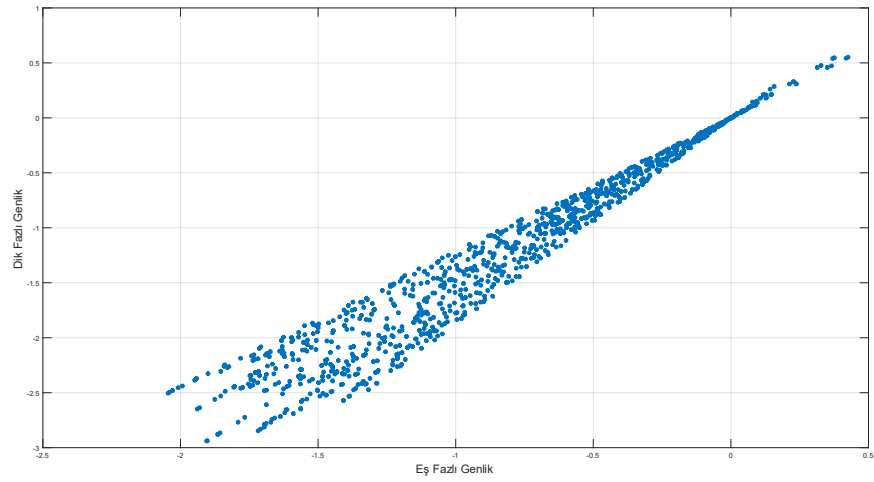


Şekil 5.6 BPSK sinyali alt-örneklenmiş yıldız kümesi şeması

Şekil 5.7’de 4ASK modülasyonuna ait sinyal gösterilmiştir. Sadece sinyal incelendiğinde Şekil 5.4’te verilen BPSK sinyalinden bir farkı gözükmemektedir. Şekil 5.8’de verilen alt örnekleme yapılmamış yıldız kümesi şeması da yine Şekil 5.5’te verilen BPSK yıldız kümesi şemasına benzemektedir. Ancak sinyal dikkatli incelendiğinde BPSK yıldız kümesi şemasında semboller 0 etrafında simetrik dağılırken, 4ASK’ya ait yıldız kümesi şemasında semboller hep artan genlik seviyesi ile dizilmiştir.

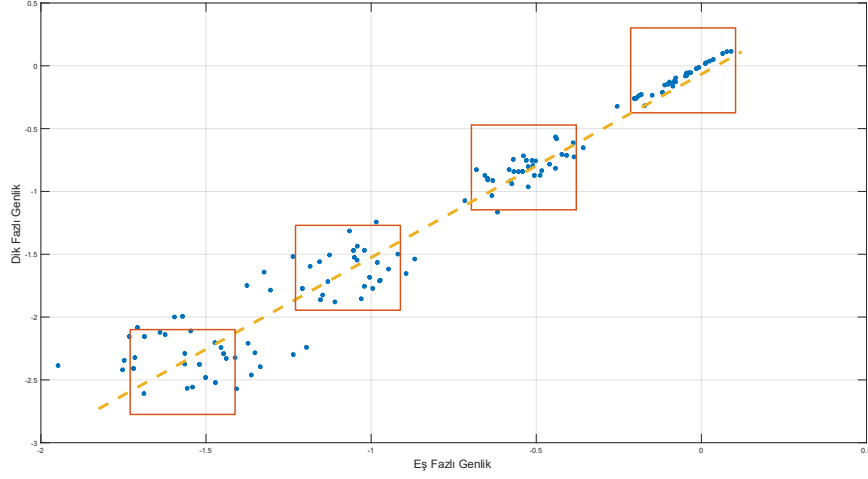


Şekil 5.7 4ASK Modülasyonu 30dB SNR



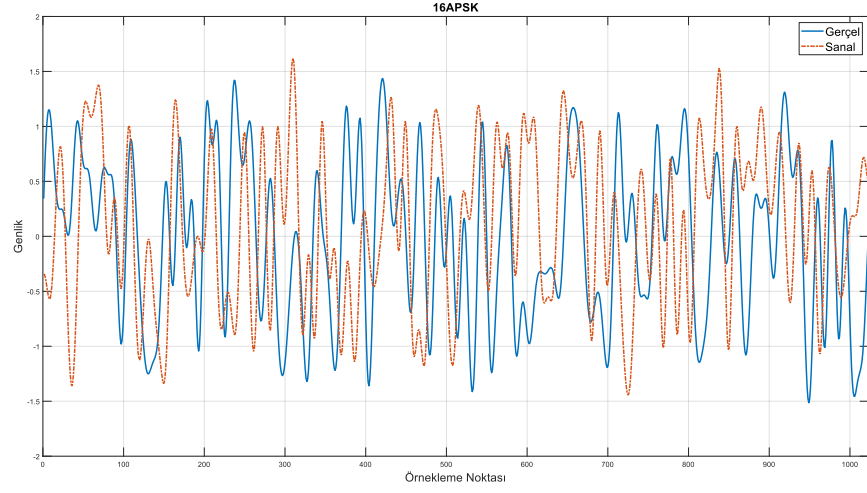
Şekil 5.8 4ASK sinyali yıldız kümesi şeması

Şekil 5.9’da 4ASK sinyalinin 8’de 1 oranla alt örnekleme hızı ile örnekleme yapılmış yıldız kümesi şeması verilmiştir. Kareler ile gruplanmış 4 farklı genlik seviyesine ait sembollerin bulunduğu görülmektedir. Bütün gruplar kesikli çizgi ile gösterilen yaklaşık 260° sabit faz çizgisi üzerinde dağılmıştır. İdeal 4ASK modülasyonunda olduğu gibi sembollerin sadece genliklerinin değiştiği, fazlarının sabit kaldığı söylenebilmektedir.

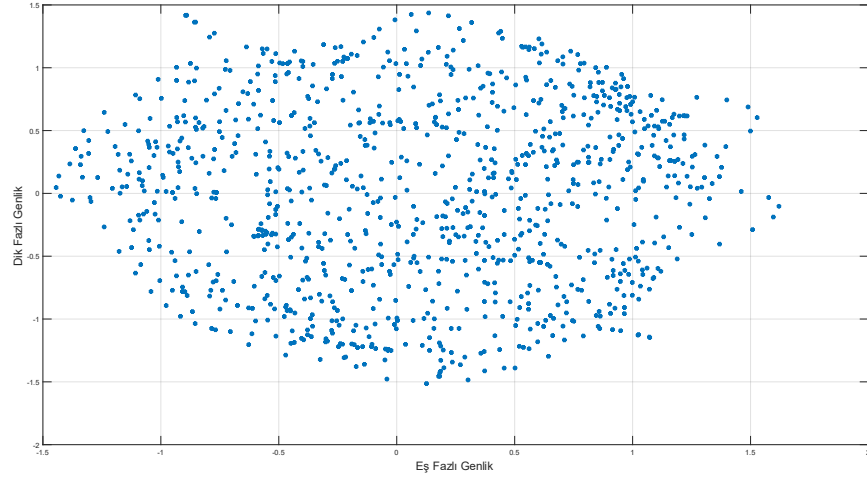


Şekil 5.9 4ASK sinyali alt-örnekleme hızı ile örnekleme yapılmış yıldız kümesi şeması

Şekil 5.10’da 16APSK modülasyonuna ait sinyal verilmiştir. Sinyali inceleyerek modülasyon türüne yönelik bir tahmin yapmak mümkün gözükmemektedir. Şekil 5.11’de verilen yıldız kümesi şeması üzerinden de yorum yapmak mümkün değildir. 0 etrafında eşit dağılmış semboller kümesi bir çok farklı modülasyon türüne ait olabilmektedir.



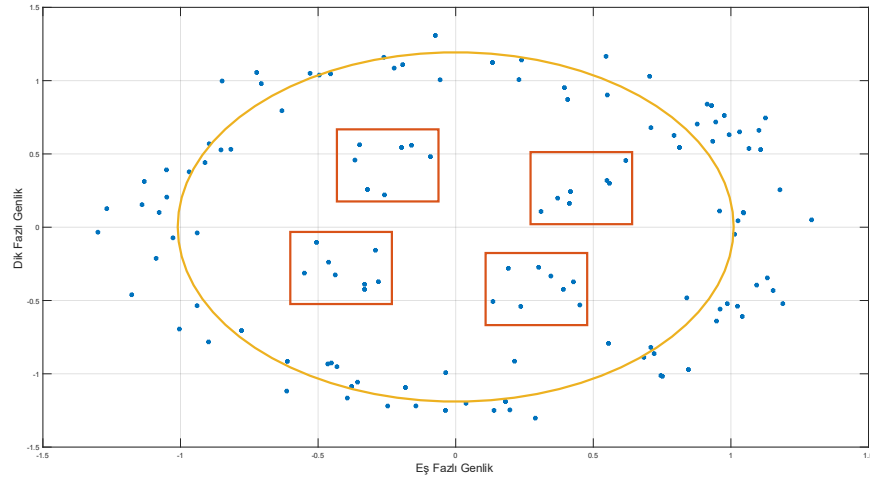
Şekil 5.10 16APSK Modülasyonu, 30dB SNR



Şekil 5.11 16APSK sinyali yıldız kümesi şeması

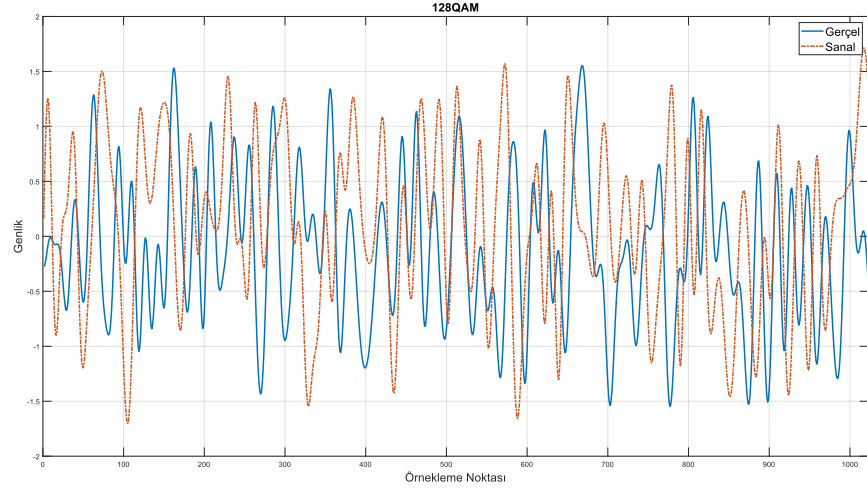
Şekil 5.12’de 16APSK modülasyonu alt örneklenmiş yıldız kümesi şeması verilmiştir. Kareler ile gruplanmış semboller ve bu sembollerin etrafında çember oluşturmuş semboller bulunmaktadır. 16 farklı APSK sembolünün 4 tanesi kareler içerisinde net olarak gözükmemektedir. Ancak çember üzerine dağılacak diğer 12 sembolü net bir şekilde gruplamak mümkün değildir. Daha karmaşık yıldız kümesi şemasına sahip modülasyon türleri incelendikçe semboller arasında ayırım yapmak ve modülasyon türünü yorumlamak daha zor bir hal alacaktır. Veri kaydı 1024 örnekleme noktasıyla alındığına

göre ve her sembol için 8’de 1 alt örnekleme yapıldığı için bir kayıt içerisinde maksimum 128 sembol bulunabilmektedir. 16 farklı sembol içeren 16APSK modülasyonu için her sembol en fazla 8 kere tekrarlanmış olabilir. Bozucu etkilerin bulunduğu bir sinyal kaydında bunların ayırık ayırık gruplandığını görmek mümkün olmayabilir. Şekil 5.12’de kareler ile gruplanmış semboller incelendiğinde, her kare içerisine 7 veya 8 sembolün düştüğü görülmektedir. Ortalamada her sembolün 8 kere tekrarlanmış olduğu bu şekilde doğrulanabilmektedir.

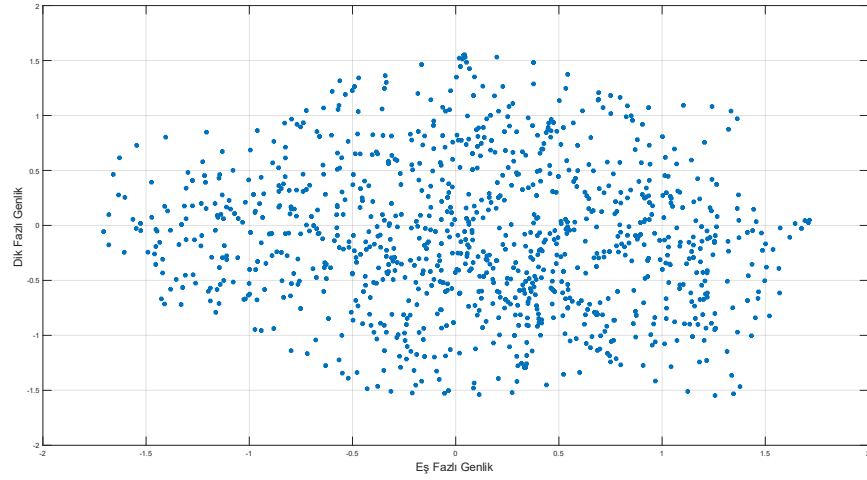


Şekil 5.12 16APSK sinyali alt-örneklenmiş yıldız kümesi şeması

Şekil 5.13’te 128QAM modülasyonuna ait sinyal gösterilmiştir. Şekil 5.14’te bu sinyale ait yıldız kümesi şeması verilmiştir.

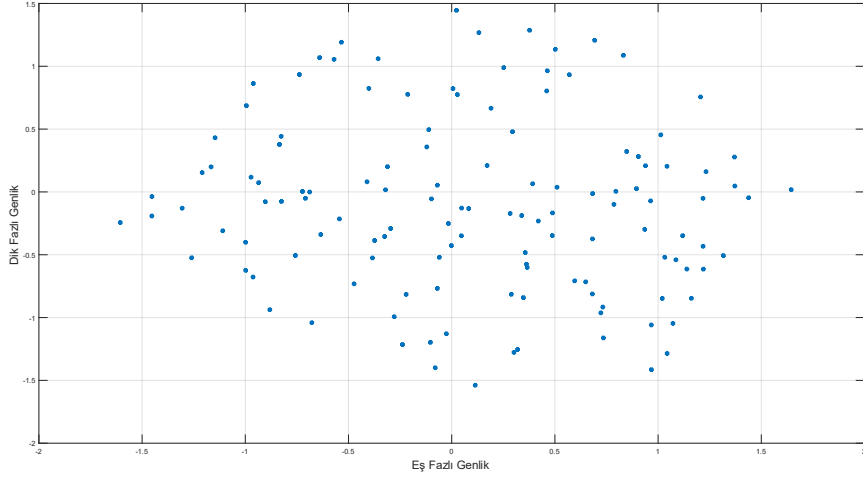


Şekil 5.13 128QAM Modülasyonu, 30dB SNR



Şekil 5.14 128QAM sinyali yıldız kümesi şeması

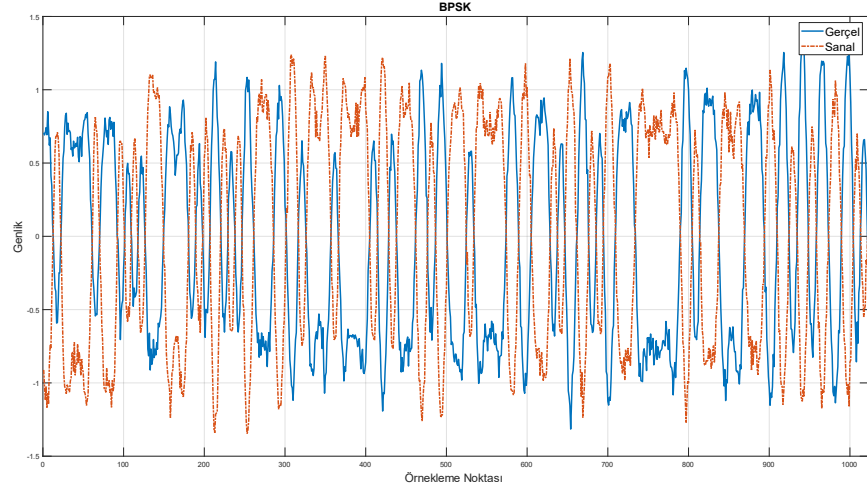
Şekil 5.15'te ise 128QAM modülasyonu alt örneklenmiş yıldız kümesi şeması verilmiştir. 128 farklı sembol içeren 128QAM modülasyonu için sembollerin gruplanması mümkün değildir. Bütün semboller birbirinden farklı olsa bile, her sembol en fazla bir kez bulunuyor olabilir. Bazı sembollerin tekrar ettiği de düşünülürse, kayıt içerisinde hiç bulunmayan semboller bile olabilir.



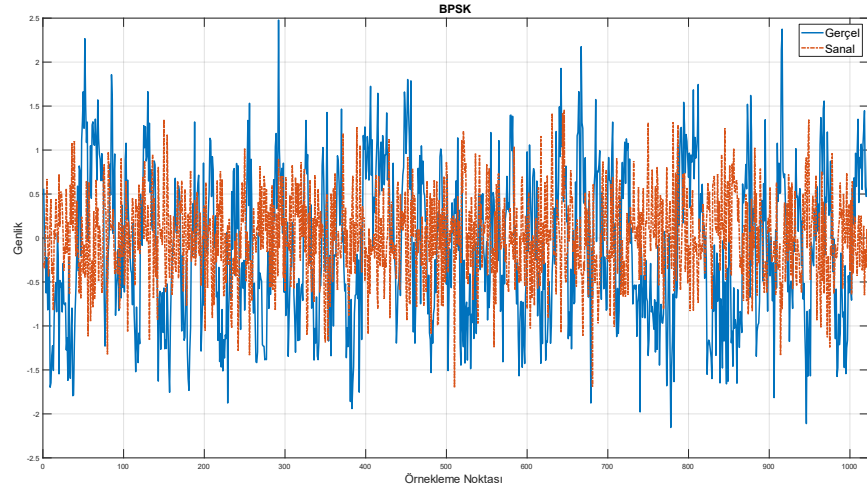
Şekil 5.15 128QAM sinyali alt-örneklenmiş yıldız kümesi şeması

64APSK-128APSK-64QAM-256QAM gibi karmaşık modülasyon türlerinde de semboller yıldız kümesi şeması ile ayırmak mümkün olmayacaktır. Özellikle 256QAM modülasyonunda bir kayıt içerisinde bütün sembollerin bulunması hiçbir zaman mümkün değildir. En iyi ihtimalle 256 farklı sembolün yalnızca yarısı bir kayıt içerisinde bulunuyor olabilir.

Sinyalin SNR seviyesi de modülasyon türünün belirlenebilmesinde önemli rol almaktadır. Şekil 5.4'te BPSK modülasyonuna ait 30dB SNR seviyesinde bir sinyal gösterilmişti. Şekil 5.16 ve Şekil 5.17'de yine BPSK modülasyonuna ait farklı sinyallerin 10dB SNR ve 0 dB SNR seviyesindeki halleri çizdirilmiştir. 30dB SNR seviyesinde sinyal şeklinde bozulmalar yokken, 10dB SNR seviyesinde gürültünün etkisi görülmeye başlanmıştır. 0dB SNR seviyesinde ise sinyali ayırt etmek artık mümkün değildir. SNR seviyesi düşükçe sinyalin şekli de daha da bozulacaktır.

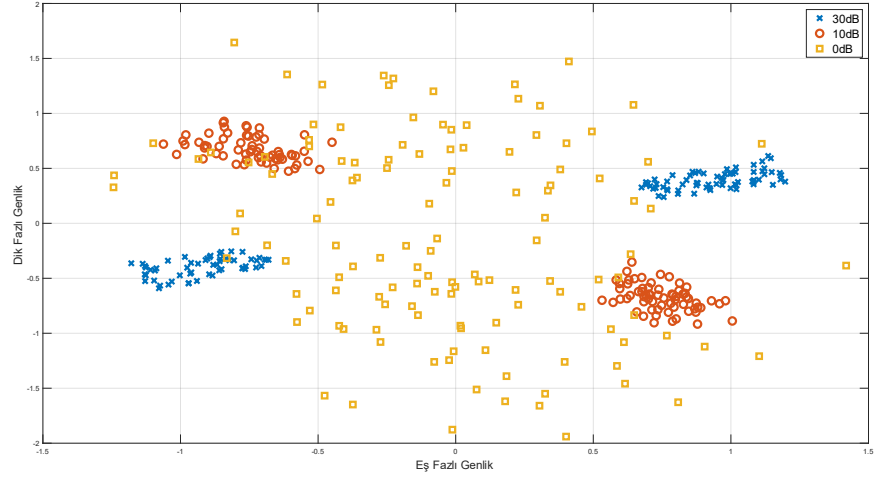


Şekil 5.16 BPSK Modülasyonu, 10dB SNR



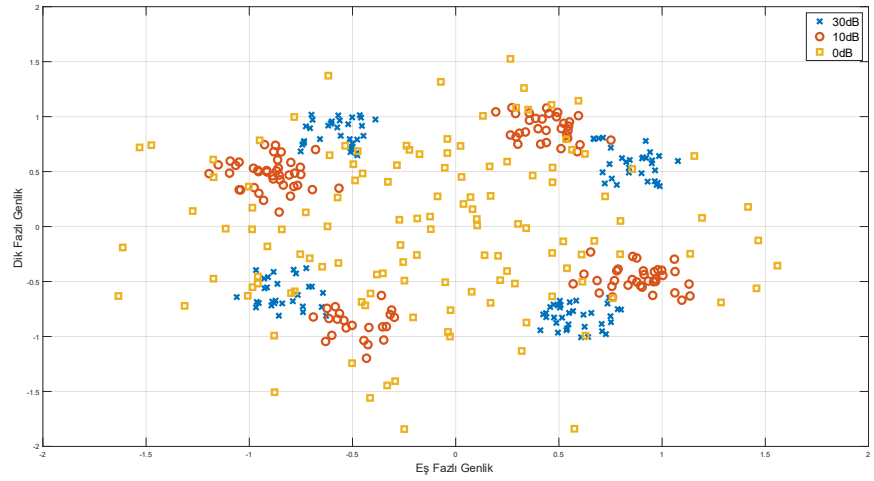
Şekil 5.17 BPSK Modülasyonu, 0dB SNR

Şekil 5.18’de 30dB, 10dB ve 0dB SNR seviyelerine sahip BPSK sinyallerinin alt örneklenmiş yıldız kümesi şemaları verilmiştir. BPSK gibi basit bir yıldız kümesi şemasına sahip modülasyon tipinde bile 0dB SNR seviyesinde gruplaşmaları görmek mümkün olmamıştır. 10dB ve 30dB SNR seviyelerinde ise sembollerin 180° faz farkı ile gruplandığı net şekilde gözükmektedir.



Şekil 5.18 BPSK modülasyonu yıldız kümesi şeması SNR karşılaştırması

Şekil 5.19’da da QPSK sinyalinin farklı SNR seviyelerindeki alt örneklenmiş yıldız kümesi şeması verilmiştir. BPSK sinyallerinde olduğu gibi 30dB ve 10dB SNR seviyelerinde sembollerin gruplandığı görülmektedir. 0 dB SNR seviyesinde ise semboller hakkında yorum yapmak mümkün değildir.



Şekil 5.19 QPSK modülasyonu yıldız kümesi şeması SNR karşılaştırması

5.2 Modülasyon Sınıflandırma Derin Öğrenme Yapıları

Farklı derin öğrenme mimarileriyle çalışılarak bu mimarilerin performansları karşılaştırılmıştır. Bütün SNR seviyelerinde sinyaller içeren veri setiyle yapılan testlerin F1 skorları Çizelge 5.1’de verilmiştir.

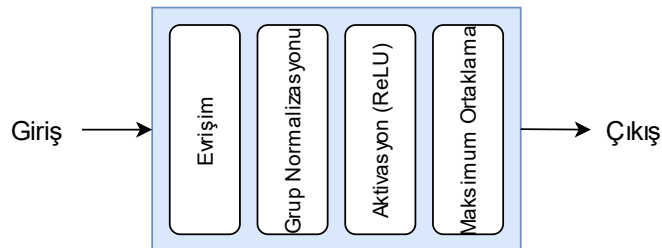
Çizelge 5.1 Kullanılan derin öğrenme modelleri ve F1 skorları

Model	F1 Skoru
CNN-1	0.62
CNN-2	0.64
CNN-3	0.62
ResNet-1	0.61
ResNet-2	0.61
ResNet-3	0.65
ResNet-4	0.65
GRU	0.55
LSTM	0.58

Bütün mimarilerin eğitiminde grup boyutu 128 olarak kullanılmıştır. Ayrıca 5 epok üst üste mimari eğitimde iyileşme sağlayamazsa eğitim durdurulmuştur. Böylece mimarinin eğitimi için uzun süre beklenmesinin önüne geçilmiştir. Veri setinin %80’i modelin eğitimi için kullanılırken kalan %20 kısmı da test için kullanılmıştır.

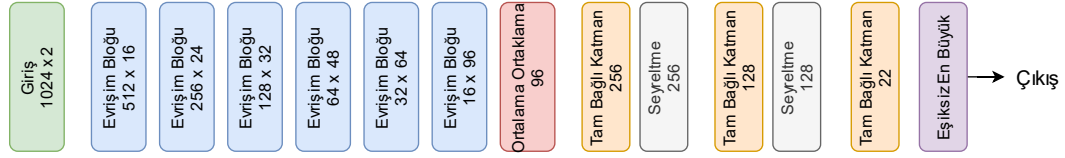
5.2.1 CNN-1 mimarisi

Denenen ilk CNN mimarisi, Matlab tarafından modülasyon sınıflandırma için oluşturulmuş örnek projeden esinlenerek kurulmuştur (Anonymous 2021a). Şekil 5.20’de kurulan mimariye ait evrişim, normalizasyon, aktivasyon ve ortalama katmanlarını içeren bir blok gösterilmiştir. Bu blok yapısında artarda 6 adet bağlanması ile mimari oluşturulmuştur.



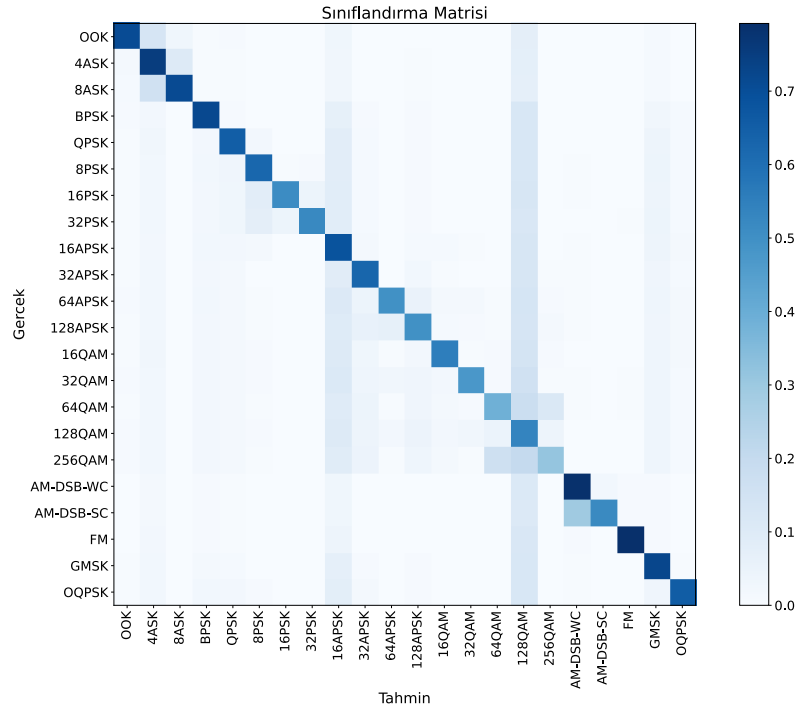
Şekil 5.20 Evrişim blok yapısı

Şekil 5.21’de kurulan mimarının genel yapısı ve her bir katmanın çıkış boyutları verilmiştir. Evrişim katmanında kullanılan filtrelerin boyutu 8x1 olarak ayarlanmıştır. Kurulan ağ yapısı 156894 adet eğitilebilir parametre içermektedir.



Şekil 5.21 CNN-1 mimarisi genel yapısı

Şekil 5.22’de CNN-1 modelinin test sonuçlarını gösteren sınıflandırma matrisi verilmektedir. Testte -20dB’den 30dB’ye kadar SNR seviyesine sahip sinyaller kullanılmıştır. Model 128QAM ve 16APSK modülasyonlarını seçmeye eğilimli gözükmemektedir.



Şekil 5.22 CNN-1 modeli sınıflandırma matrisi

Çizelge 5.2’de her bir modülasyon türüne ait kesinlik, duyarlılık ve F1 skoru değerleri verilmiştir. 128QAM daha baskın olarak seçildiği için en düşük kesinlik puanı ve en düşük F1 skorunu almıştır. 16APSK modülasyonu da 128QAM’dan sonra en düşük

kesinlik değerine sahiptir ancak duyarlılık puanı daha yüksek olduğu için F1 skoru olarak çok düşük değer almamıştır.

Çizelge 5.2 CNN-1 modeli detaylı performans raporu

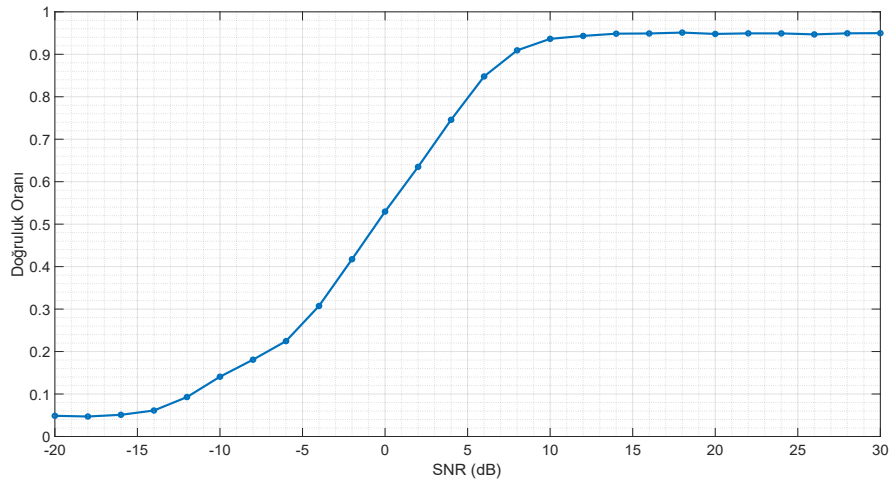
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.84	0.71	0.77
4ASK	0.51	0.75	0.61
8ASK	0.84	0.71	0.77
BPSK	0.67	0.72	0.69
QPSK	0.68	0.65	0.66
8PSK	0.72	0.62	0.67
16PSK	0.9	0.51	0.65
32PSK	0.9	0.52	0.66
16APSK	0.31	0.69	0.43
32APSK	0.63	0.63	0.63
64APSK	0.79	0.5	0.61
128APSK	0.6	0.5	0.54
16QAM	0.83	0.55	0.66
32QAM	0.87	0.48	0.61
64QAM	0.61	0.39	0.48
128QAM	0.17	0.53	0.26
256QAM	0.62	0.31	0.41
AM-DSB-WC	0.69	0.79	0.74
AM-DSB-SC	0.96	0.52	0.67
FM	0.93	0.79	0.86
GMSK	0.55	0.72	0.62
OQPSK	0.8	0.66	0.72
Ortalama	0.7	0.6	0.62

SNR seviyesinin sınıflandırmaya olan etkisini görmek için sadece belirli SNR seviyesine sahip sinyaller ile test tekrarlanmıştır. Çizelge 5.3'te SNR ile F1 skoru değerlendirmesi yapılmıştır. Basit olarak değerlendirilebilecek OOK, BPSK, FM, GMSK, OQPSK gibi modülasyon sinyalleri 0dB SNR seviyesinde bile %90 üzeri bir performansla sınıflandırılabilmiştir. Ancak daha büyük modülasyon şemaları içeren 256QAM, 128QAM ve 64QAM sinyalleri 30dB SNR seviyesinde bile %80 başarımlarına ancak ulaşabilmiştir. Bölüm 5.1'de veri seti içerisindeki sinyaller incelenirken, büyük modülasyon şemasına sahip sinyaller için sınıflandırmanın problemli olacağı açıklanmıştır.

Çizelge 5.3 CNN-1 modeli SNR'a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.05	0.21	0.99	1.00	1.00	1.00
4ASK	0.03	0.24	0.77	0.99	1.00	1.00
8ASK	0.00	0.28	0.77	0.99	1.00	1.00
BPSK	0.02	0.07	1.00	1.00	1.00	1.00
QPSK	0.01	0.01	0.83	1.00	1.00	1.00
8PSK	0.03	0.05	0.30	1.00	1.00	1.00
16PSK	0.00	0.00	0.20	0.97	0.99	0.98
32PSK	0.02	0.02	0.21	0.97	0.98	0.99
16APSK	0.04	0.06	0.41	1.00	1.00	1.00
32APSK	0.00	0.00	0.42	1.00	1.00	1.00
64APSK	0.00	0.00	0.07	0.94	0.95	0.96
128APSK	0.01	0.02	0.15	0.92	0.95	0.94
16QAM	0.00	0.00	0.14	1.00	1.00	1.00
32QAM	0.00	0.00	0.12	0.94	0.97	0.97
64QAM	0.03	0.04	0.09	0.67	0.74	0.74
128QAM	0.09	0.10	0.09	0.76	0.82	0.82
256QAM	0.01	0.01	0.15	0.63	0.68	0.68
AM-DSB-WC	0.04	0.33	0.71	0.92	0.92	0.93
AM-DSB-SC	0.00	0.32	0.66	0.92	0.92	0.92
FM	0.04	0.53	1.00	1.00	1.00	1.00
GMSK	0.04	0.05	1.00	1.00	1.00	1.00
OQPSK	0.00	0.01	0.90	1.00	1.00	1.00

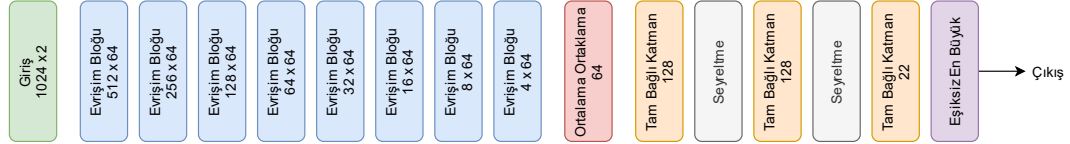
Şekil 5.23'te -20dB'den 30dB SNR seviyesine kadar 2 dB aralıklarla alınmış doğruluk oranı grafiği verilmiştir. Yaklaşık 10 dB seviyesinden sonra modelin genel başarımları %95 seviyesinde doyuma ulaşmıştır.



Şekil 5.23 CNN1 modeli SNR'a göre doğruluk oranı

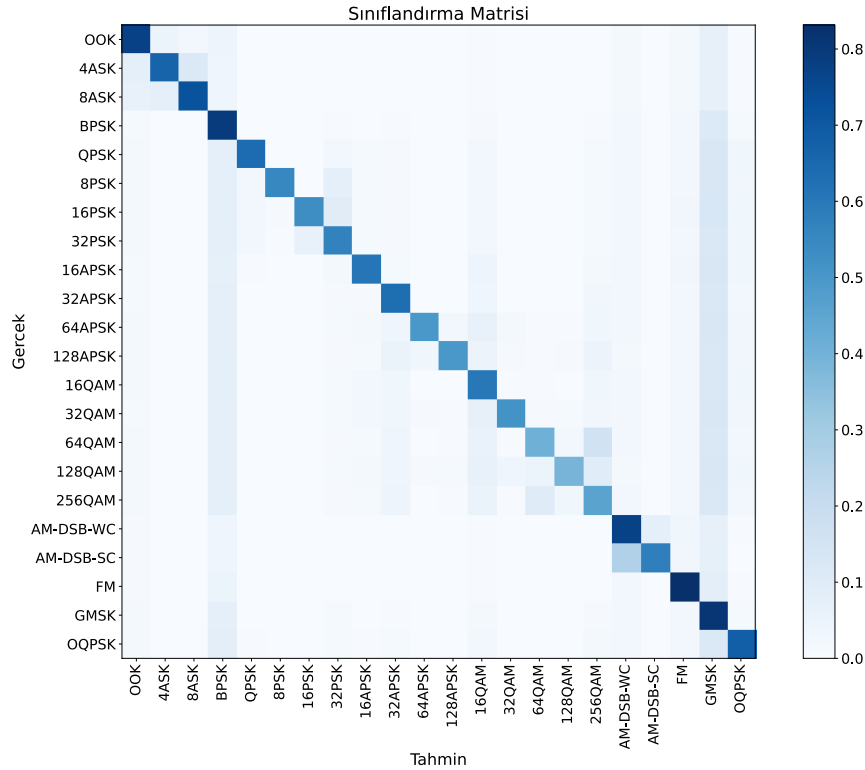
5.2.2 CNN-2 mimarisi

Şekil 5.24'te CNN-2 mimarisinin genel yapısı verilmiştir. Evrişim blokları Şekil 5.20'de olduğu gibi kullanılmıştır. CNN-1 mimarisinden farklı olarak evrişim bloklarının sayısı 8'e yükseltilmiştir. Ayrıca evrişim bloklarında kullanılan filtrelerin sayısı değiştirilmiştir. Bu ağ yapısı içerisinde 218390 adet eğitilebilir parametre bulunmaktadır.



Şekil 5.24 CNN-2 mimarisi genel yapısı

Şekil 5.25'te CNN2 modelinin sınıflandırma matrisi verilmiştir. Yapılan testte -20 dB'den 30dB'ye kadar bütün SNR seviyelerinden sinyaller kullanılmıştır. 64-128-256QAM modülasyonları CNN1 modelinde olduğu gibi gözükmemektedir. Ayrıca model GMSK ve BPSK modülasyonunu seçmek yönünde eğilimli olduğu gözükmemektedir.



Şekil 5.25 CNN-2 modeli sınıflandırma matrisi

Çizelge 5.4 F1 skorları incelendiğinde GMSK ve BPSK modülasyonlarının problemlili olduğu görülmektedir.

Çizelge 5.4 CNN-2 modeli detaylı performans raporu

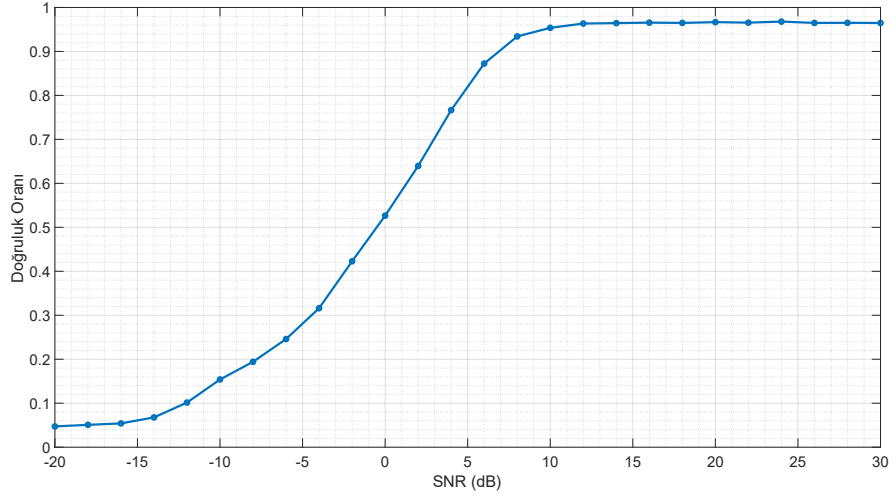
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.67	0.78	0.72
4ASK	0.82	0.66	0.73
8ASK	0.82	0.72	0.77
BPSK	0.37	0.79	0.51
QPSK	0.87	0.64	0.74
8PSK	0.98	0.55	0.71
16PSK	0.89	0.53	0.66
32PSK	0.65	0.57	0.6
16APSK	0.78	0.61	0.68
32APSK	0.65	0.63	0.64
64APSK	0.9	0.5	0.64
128APSK	0.91	0.5	0.64
16QAM	0.5	0.6	0.55
32QAM	0.87	0.52	0.65
64QAM	0.7	0.41	0.52
128QAM	0.83	0.39	0.53
256QAM	0.47	0.46	0.47
AM-DSB-WC	0.54	0.77	0.63
AM-DSB-SC	0.89	0.58	0.7
FM	0.63	0.83	0.71
GMSK	0.27	0.81	0.4
OQPSK	0.62	0.68	0.65
Ortalama	0.71	0.61	0.63

Çizelge 5.5 incelendiğinde SNR seviyesinin artmasıyla 64-128-256QAM dışında kalan bütün modülasyonların %90 başarımlarının üstüne çıktığı görülmektedir.

Çizelge 5.5 CNN-2 modeli SNR’a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.04	0.21	0.99	1.00	1.00	1.00
4ASK	0.03	0.23	0.69	1.00	1.00	1.00
8ASK	0.02	0.25	0.78	1.00	1.00	1.00
BPSK	0.07	0.10	1.00	1.00	1.00	1.00
QPSK	0.00	0.00	0.83	1.00	1.00	1.00
8PSK	0.01	0.01	0.16	1.00	1.00	1.00
16PSK	0.00	0.00	0.10	0.98	0.99	0.99
32PSK	0.02	0.02	0.44	0.99	0.99	0.99
16APSK	0.00	0.00	0.47	1.00	1.00	1.00
32APSK	0.01	0.01	0.45	1.00	1.00	1.00
64APSK	0.01	0.01	0.09	0.97	0.98	0.98
128APSK	0.00	0.01	0.13	0.96	0.98	0.98
16QAM	0.04	0.06	0.17	1.00	1.00	1.00
32QAM	0.00	0.00	0.12	0.98	0.99	0.99
64QAM	0.01	0.01	0.12	0.77	0.86	0.85
128QAM	0.04	0.04	0.04	0.85	0.90	0.89
256QAM	0.00	0.01	0.17	0.73	0.81	0.81
AM-DSB-WC	0.06	0.44	0.74	0.93	0.93	0.94
AM-DSB-SC	0.00	0.09	0.71	0.93	0.93	0.93
FM	0.06	0.47	1.00	1.00	1.00	1.00
GMSK	0.05	0.07	1.00	1.00	1.00	1.00
OQPSK	0.01	0.01	0.93	1.00	1.00	1.00

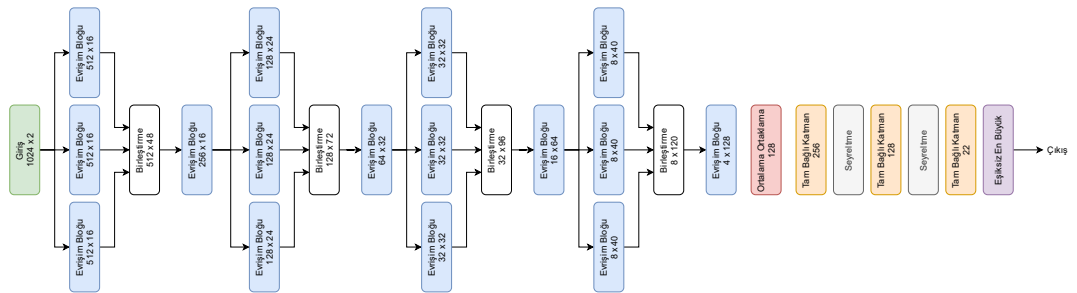
Şekil 5.26’da SNR’a göre doğruluk oranı grafiği verilmiştir. CNN1 modelinde de olduğu gibi 10dB seviyesinden sonra doyuma ulaşmıştır. Doyum noktasındaki ortalama doğruluk oranı %96 olarak görülmektedir.



Şekil 5.26 CNN2 modeli SNR'a göre doğruluk oranı

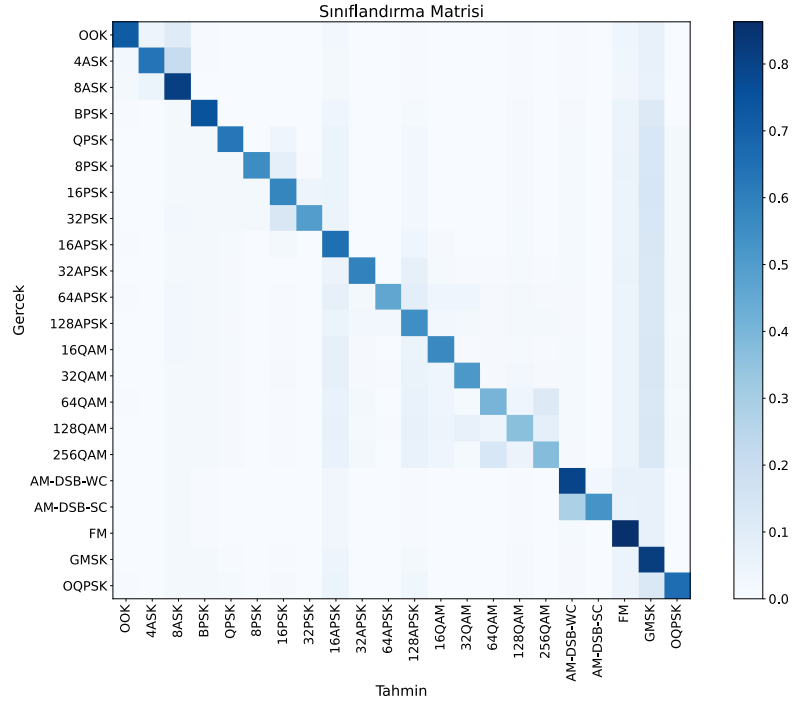
5.2.3 CNN-3 mimarisi

Şekil 5.27'de CNN-3 mimarisi genel yapısı verilmiştir. CNN-1 ve CNN-2 modellerinden farklı olarak paralel bağlanmış evrişim blokları gözükmemektedir. Bu paralel bağlanmış bloklar içerisindeki filtre boyutları birbirinden farklıdır. CNN-1 ve CNN-2 modellerinde evrişim blokları içerisinde 8x1 boyutunda filtreler kullanılmıştır. Bu paralel bloklarda ise 2x1, 4x1 ve 8x1 boyutlarında filtreler kullanılmıştır. 3 farklı filtre boyutuyla oluşturulan bloklar birleştirilerek tekrar evrişim bloğuna sokulmuştur. Tekli evrişim blokları içerisinde ise yine 8x1 boyutunda filtreler kullanılmıştır. Bu ağ yapısı içerisinde 151062 adet eğitilebilir parametre bulunmaktadır.



Şekil 5.27 CNN-3 mimarisi genel yapısı

Şekil 5.28'de CNN-2 modelinin sınıflandırma matrisi verilmiştir. Modelin GMSK modeline eğilimli olduğu görülmektedir.



Şekil 5.28 CNN-3 modeli sınıflandırma matrisi

Çizelge 5.6 incelendiğinde GMSK modülasyonu en düşük F1 skorunu almıştır. Kesinlik değeri çok düşük olduğu için F1 skorunda 64-128-256QAM modülasyonlarından bile daha düşük değer almıştır.

Çizelge 5.6 CNN-3 modeli detaylı performans raporu

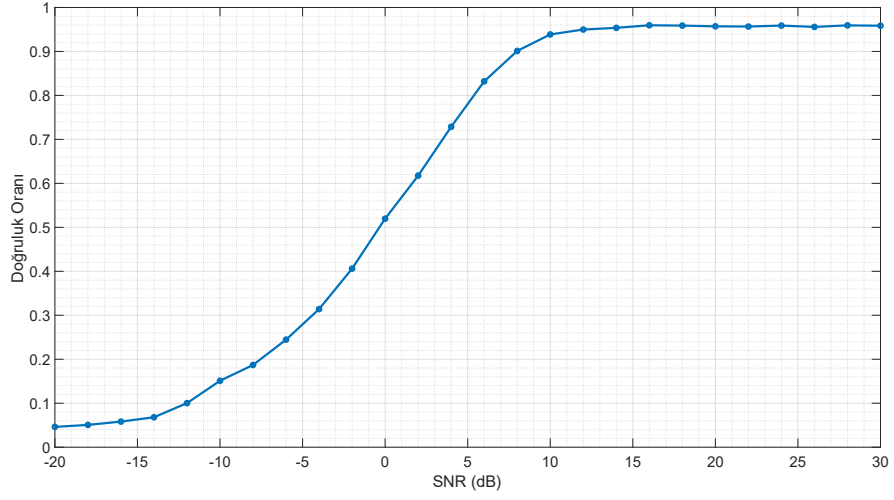
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.89	0.72	0.79
4ASK	0.88	0.64	0.74
8ASK	0.55	0.81	0.65
BPSK	0.71	0.75	0.73
QPSK	0.8	0.63	0.7
8PSK	0.92	0.55	0.69
16PSK	0.64	0.58	0.61
32PSK	0.89	0.49	0.63
16APSK	0.4	0.65	0.5
32APSK	0.85	0.59	0.7
64APSK	0.93	0.46	0.62
128APSK	0.45	0.55	0.49
16QAM	0.7	0.57	0.63
32QAM	0.76	0.51	0.61
64QAM	0.65	0.41	0.5
128QAM	0.57	0.36	0.44
256QAM	0.62	0.38	0.47
AM-DSB-WC	0.62	0.79	0.7
AM-DSB-SC	0.96	0.53	0.68
FM	0.45	0.86	0.59
GMSK	0.26	0.82	0.4
OQPSK	0.77	0.66	0.71
Ortalama	0.69	0.6	0.62

Çizelge 5.7’de SNR’ın yükselmesi ile sınıflandırma performansının düzeldiği görülmektedir. Beklenildiği üzere 64-128-256QAM modülasyonlarının sınıflandırılmasında sorun devam etmektedir. Modelden bağımsız olarak bu modülasyon grubunun ayrıştırılması diğer modülasyon türlerinin ayrıştırılması kadar başarılı değildir.

Çizelge 5.7 CNN-3 modeli SNR'a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.02	0.23	0.98	1.00	1.00	1.00
4ASK	0.02	0.17	0.72	0.99	0.99	0.99
8ASK	0.04	0.34	0.78	0.99	0.99	0.99
BPSK	0.02	0.07	1.00	1.00	1.00	1.00
QPSK	0.00	0.00	0.77	1.00	1.00	1.00
8PSK	0.00	0.00	0.21	1.00	1.00	1.00
16PSK	0.00	0.00	0.34	0.96	0.98	0.99
32PSK	0.01	0.00	0.15	0.96	0.98	0.99
16APSK	0.05	0.06	0.41	1.00	1.00	1.00
32APSK	0.00	0.01	0.39	1.00	1.00	1.00
64APSK	0.00	0.00	0.08	0.94	0.96	0.96
128APSK	0.01	0.01	0.13	0.93	0.96	0.96
16QAM	0.00	0.00	0.15	1.00	1.00	1.00
32QAM	0.00	0.00	0.13	0.93	0.96	0.96
64QAM	0.00	0.00	0.06	0.67	0.78	0.77
128QAM	0.06	0.06	0.01	0.74	0.82	0.81
256QAM	0.00	0.00	0.13	0.62	0.70	0.70
AM-DSB-WC	0.07	0.39	0.76	0.93	0.93	0.94
AM-DSB-SC	0.00	0.01	0.66	0.93	0.93	0.93
FM	0.05	0.42	1.00	1.00	1.00	1.00
GMSK	0.09	0.10	1.00	1.00	1.00	1.00
OQPSK	0.01	0.02	0.89	1.00	1.00	1.00

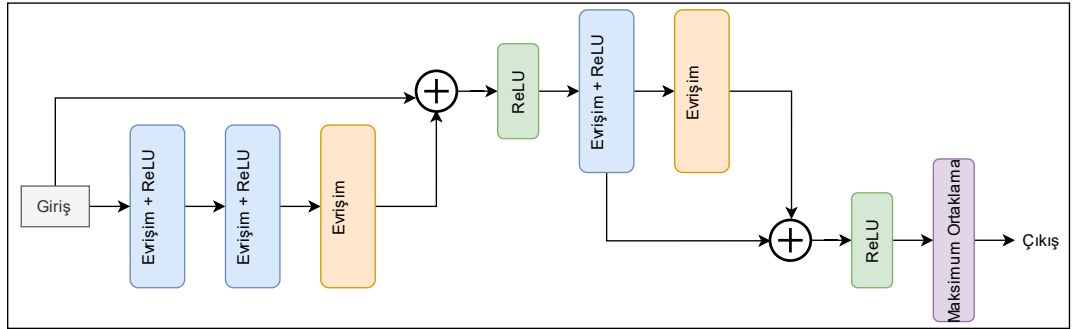
Şekil 5.29 incelendiğinde diğer modellerde de olduğu gibi 10dB SNR seviyesinden itibaren doğruluk oranı doyuma ulaşmıştır. Görülen doğruluk oranı yaklaşık olarak %96 seviyesindedir.



Şekil 5.29 CNN-3 modeli SNR'a göre doğruluk oranı

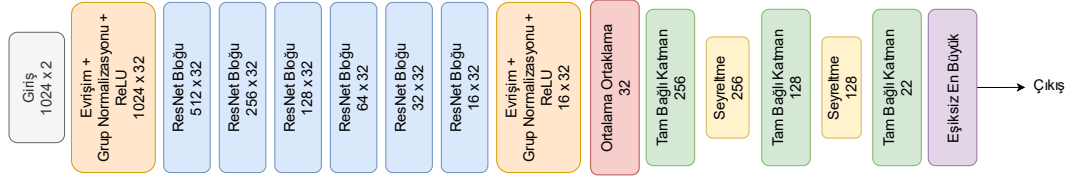
5.2.4 ResNet-1 mimarisi

Şekil 5.30'da ResNet mimarisi içerisinde tekrarlı olarak kullanılan bloğun yapısı verilmiştir. Bu bloktan artarda bağlanarak oluşturulan mimari ise şekil 5.31'de verilmiştir.



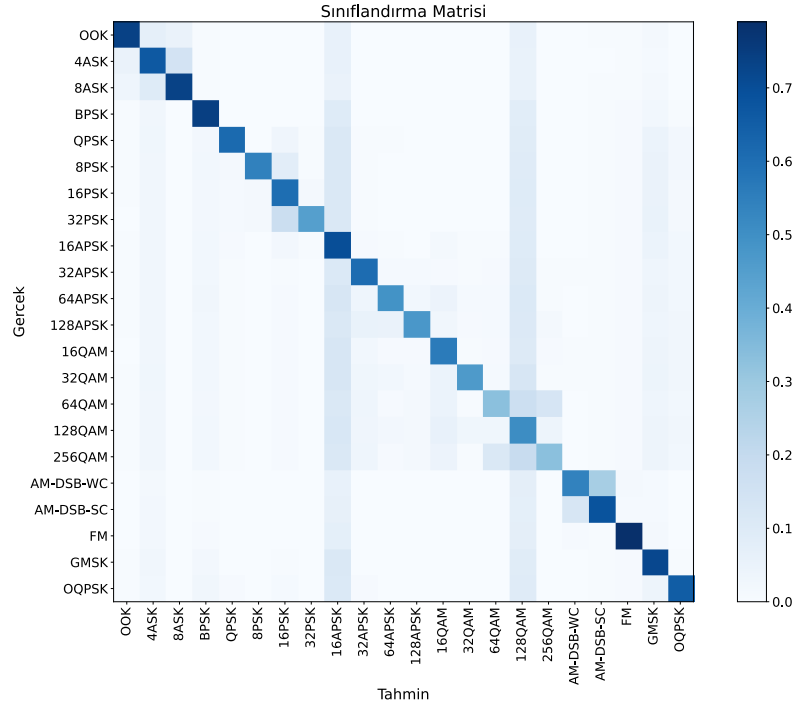
Şekil 5.30 ResNet-1 blok yapısı

Mimari içerisinde 179670 adet eğitilebilir parametre bulunmaktadır. Mimari içerisinde evrişim katmanlarında kullanılan filtreler 5x1 boyutundadır. Bu mimari Xilinx firması tarafından modülasyon sınıflandırma problemi için oluşturulan örnek projeden alınmıştır (Anonymous 2021b).



Şekil 5.31 ResNet-1 mimarisi genel yapısı

Şekil 5.32’de bütün SNR seviyelerinde sinyaller ile yapılan test için sınıflandırma matrisi verilmiştir. Model 16APSK ve 128QAM modülasyonlarını seçmeye eğilimli olduğu gözükmemektedir.



Şekil 5.32 ResNet-1 modeli sınıflandırma matrisi

Çizelge 5.8 incelendiğinde tamamıyla sınıflandırılmayan bir modülasyon türü olmadığı gözükmemektedir. 64-128-256QAM modülasyonları yine düşük F1 skorunda kalmıştır. Bunlara ek olarak 16APSK modülasyonu da düşük kesinlik değerinden dolayı düşük F1 skoru elde etmiştir.

Çizelge 5.8 ResNet-1 modeli detaylı performans raporu

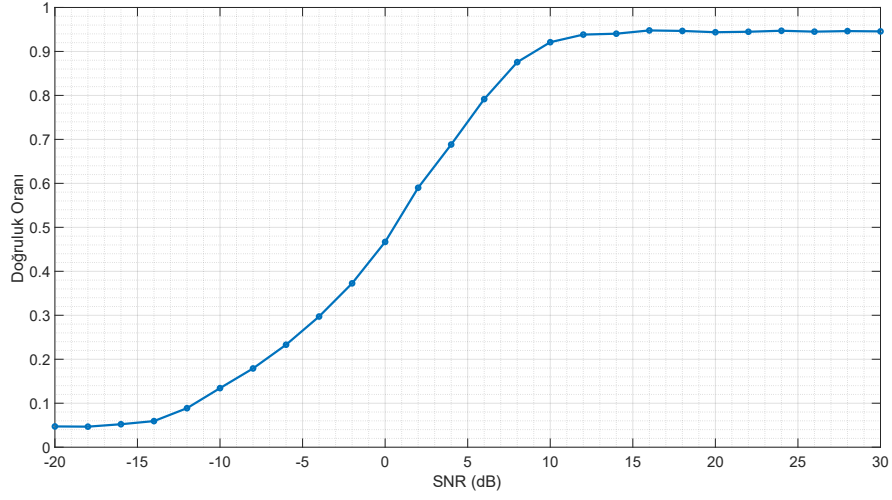
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.83	0.73	0.78
4ASK	0.51	0.66	0.58
8ASK	0.79	0.73	0.76
BPSK	0.67	0.74	0.7
QPSK	0.88	0.61	0.72
8PSK	0.95	0.54	0.69
16PSK	0.61	0.6	0.61
32PSK	0.89	0.45	0.59
16APSK	0.25	0.7	0.37
32APSK	0.69	0.61	0.65
64APSK	0.77	0.49	0.6
128APSK	0.82	0.47	0.6
16QAM	0.66	0.56	0.61
32QAM	0.85	0.47	0.6
64QAM	0.63	0.33	0.43
128QAM	0.2	0.51	0.29
256QAM	0.62	0.33	0.43
AM-DSB-WC	0.74	0.54	0.63
AM-DSB-SC	0.65	0.68	0.67
FM	0.84	0.79	0.81
GMSK	0.51	0.72	0.59
OQPSK	0.68	0.65	0.66
Ortalama	0.68	0.59	0.61

Diğer modellerde olduğu gibi SNR arttıkça sınıflandırma başarısının arttığı çizelge 5.9’da gösterilmiştir.

Çizelge 5.9 ResNet-1 modeli SNR'a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.01	0.26	0.97	1.00	1.00	1.00
4ASK	0.04	0.24	0.68	0.99	1.00	1.00
8ASK	0.00	0.26	0.72	0.99	1.00	1.00
BPSK	0.03	0.06	1.00	1.00	1.00	1.00
QPSK	0.03	0.04	0.58	1.00	1.00	1.00
8PSK	0.03	0.02	0.25	1.00	1.00	1.00
16PSK	0.02	0.02	0.16	0.92	0.98	0.98
32PSK	0.00	0.00	0.18	0.92	0.98	0.98
16APSK	0.04	0.04	0.32	1.00	1.00	1.00
32APSK	0.00	0.00	0.38	1.00	1.00	1.00
64APSK	0.00	0.00	0.07	0.92	0.98	0.98
128APSK	0.00	0.00	0.11	0.92	0.98	0.98
16QAM	0.00	0.00	0.09	0.99	1.00	1.00
32QAM	0.00	0.00	0.11	0.93	0.98	0.98
64QAM	0.00	0.01	0.15	0.68	0.81	0.81
128QAM	0.08	0.08	0.05	0.75	0.85	0.85
256QAM	0.03	0.04	0.06	0.65	0.75	0.74
AM-DSB-WC	0.05	0.39	0.45	0.90	0.90	0.90
AM-DSB-SC	0.01	0.18	0.53	0.89	0.89	0.89
FM	0.05	0.35	1.00	1.00	1.00	1.00
GMSK	0.06	0.07	0.98	1.00	1.00	1.00
OQPSK	0.01	0.03	0.81	1.00	1.00	1.00

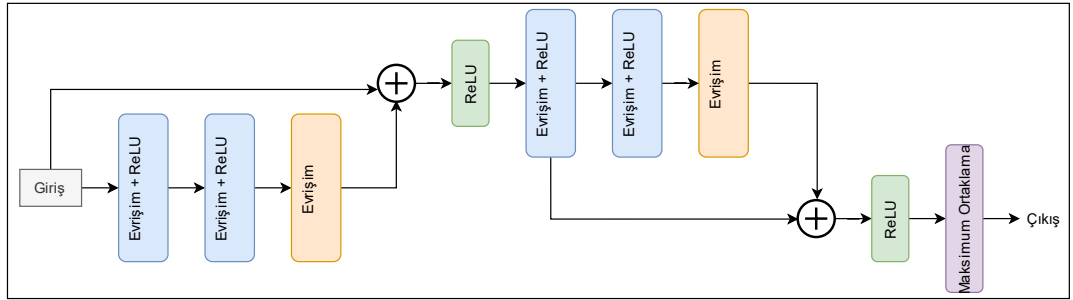
Şekil 5.33'te ResNet-1 modelinin SNR'a göre doğruluk oranı grafiği verilmiştir. 10dB SNR seviyesiyle beraber sınıflandırma başarıımı doyuma ulaşmıştır. Doyum noktasında yaklaşık doğruluk oranı %94 olarak gözükmemektedir.



Şekil 5.33 ResNet-1 modeli SNR'a göre doğruluk oranı

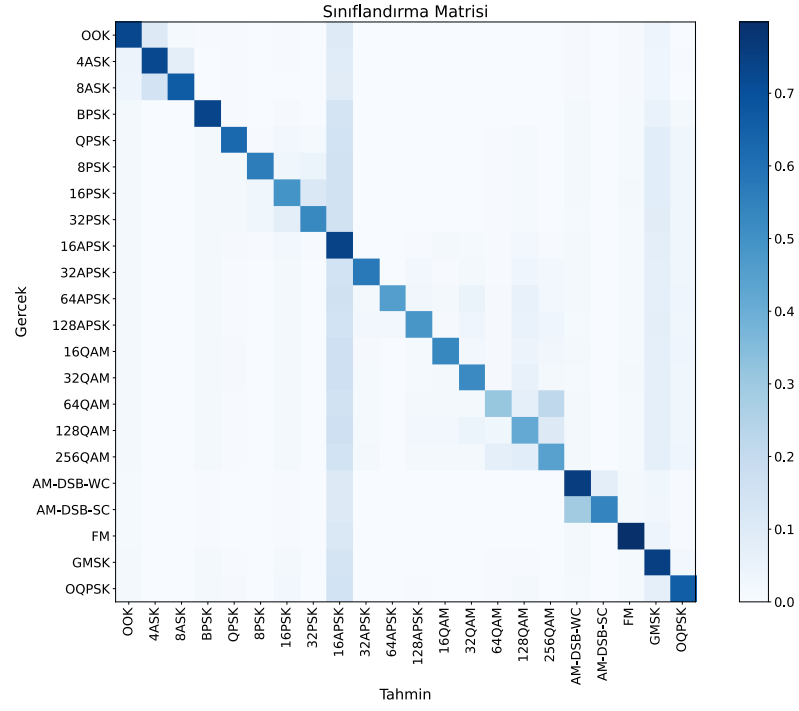
5.2.5 ResNet-2 mimarisi

ResNet-1 mimarisinden farklı olarak resnet bloklarının iç yapısı değiştirilmiştir. ResNet-1 mimarisinde evrişim içeren 5 katman bulunurken, ResNet-2 mimarisinde 6. evrişim katmanı eklenmiştir. Oluşan yeni bloğun iç yapısı Şekil 5.34'te verilmiştir. Bu blok Şekil 5.31'de olduğu gibi birebir şekilde kullanılmıştır. Bu mimari içerisinde 241494 adet eğitilebilir parametre bulunmaktadır.



Şekil 5.34 ResNet-2 blok yapısı

Şekil 5.35'te ResNet-2 modelinin tüm SNR seviyesindeki sinyaller ile yapılan testinin sınıflandırma matrisi verilmiştir. Model 16APSK modülasyonunu seçmeye eğilimli olduğu gözükmemektedir. Diğer modellerde olduğu gibi 64-128-256 QAM modülasyonları birbirinden ayırtılamamış gözükmemektedir.



Şekil 5.35 ResNet-2 modeli sınıflandırma matrisi

Çizelge 5.10’da ResNet-2 mimarisinin bütün SNR seviyelerinde sinyalleri içeren testi için alınmış detaylı performans raporu verilmiştir. Sınıflandırma matrisinde de görüldüğü üzere en düşük değer 16APSK modülasyonundadır.

Çizelge 5.10 ResNet-2 mimarisi detaylı performans raporu

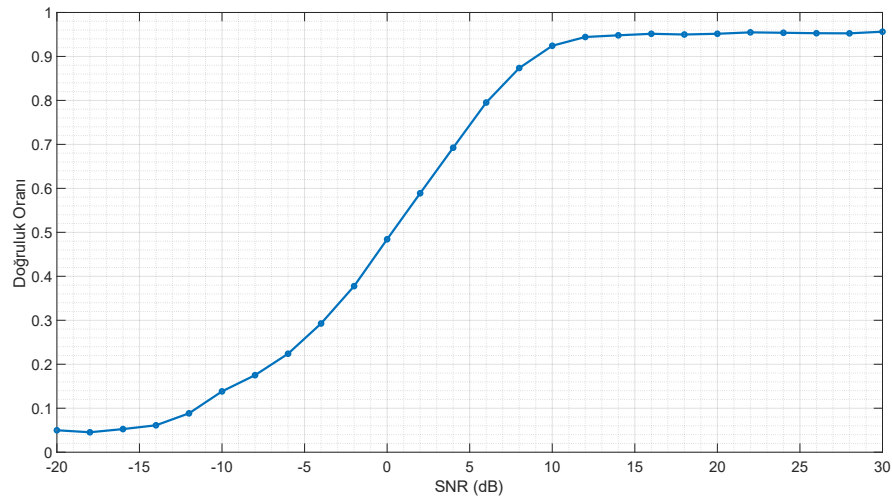
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.65	0.72	0.69
4ASK	0.72	0.72	0.72
8ASK	0.88	0.66	0.76
BPSK	0.74	0.73	0.74
QPSK	0.83	0.62	0.71
8PSK	0.87	0.56	0.68
16PSK	0.59	0.49	0.54
32PSK	0.75	0.52	0.62
16APSK	0.2	0.74	0.32
32APSK	0.85	0.57	0.68
64APSK	0.94	0.45	0.61
128APSK	0.78	0.48	0.6
16QAM	0.81	0.53	0.64
32QAM	0.71	0.52	0.6
64QAM	0.65	0.31	0.42
128QAM	0.46	0.42	0.44
256QAM	0.5	0.45	0.47
AM-DSB-WC	0.58	0.75	0.66
AM-DSB-SC	0.86	0.54	0.66
FM	0.77	0.8	0.78
GMSK	0.36	0.75	0.49
OQPSK	0.6	0.66	0.62
Ortalama	0.69	0.59	0.61

SNR seviyesinin yükselmesiyle ResNet-2 mimarisinin performansının değişimi çizelge 5.11’de verilmiştir. 10dB seviyesiyle beraber 16APSK modülasyonunda görülen düşük skor kaybolmuştur.

Çizelge 5.11 ResNet-2 mimarisi SNR'a göre F1 skoru değerlendirmesi

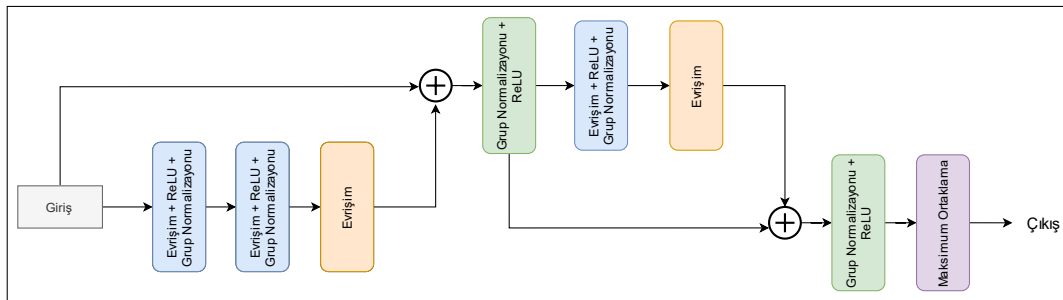
Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.03	0.16	0.98	1.00	1.00	1.00
4ASK	0.05	0.38	0.71	0.99	1.00	1.00
8ASK	0.00	0.19	0.69	0.99	1.00	1.00
BPSK	0.03	0.06	1.00	1.00	1.00	1.00
QPSK	0.03	0.02	0.66	1.00	1.00	1.00
8PSK	0.02	0.02	0.28	1.00	1.00	1.00
16PSK	0.01	0.03	0.21	0.92	0.97	0.98
32PSK	0.00	0.00	0.19	0.93	0.97	0.98
16APSK	0.06	0.07	0.34	1.00	1.00	1.00
32APSK	0.00	0.01	0.37	1.00	1.00	1.00
64APSK	0.00	0.00	0.03	0.95	0.97	0.98
128APSK	0.00	0.00	0.10	0.93	0.97	0.98
16QAM	0.00	0.00	0.12	1.00	1.00	1.00
32QAM	0.00	0.00	0.15	0.93	0.98	0.98
64QAM	0.00	0.01	0.02	0.62	0.77	0.78
128QAM	0.03	0.03	0.05	0.74	0.84	0.85
256QAM	0.00	0.00	0.16	0.64	0.72	0.73
AM-DSB-WC	0.06	0.44	0.65	0.89	0.89	0.89
AM-DSB-SC	0.00	0.02	0.46	0.86	0.86	0.85
FM	0.05	0.40	1.00	1.00	1.00	1.00
GMSK	0.07	0.09	0.99	1.00	1.00	1.00
OQPSK	0.01	0.02	0.78	1.00	1.00	1.00

Şekil 5.36'da ResNet-2 modelinin SNR'a göre doğruluk oranı değişimi verilmiştir. Diğer modellerde de olduğu gibi 10dB seviyesinden sonra model doyuma ulaşmıştır. Doyum noktasında görülen doğruluk oranı yaklaşık %95'tir.



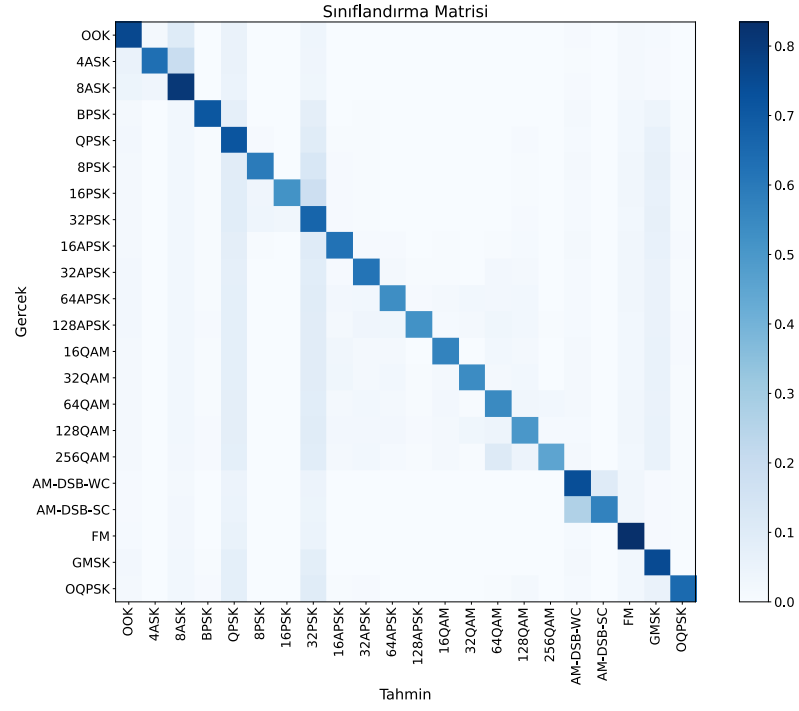
5.2.6 ResNet-3 mimarisi

Şekil 5.31’de verilen ResNet-1 mimarisiyle aynı yapıyı kullanmaktadır. Sadece ResNet bloklarının iç yapısında değişiklik yapılmış ve bu katmanlara grup normalizasyonu eklenmiştir. Şekil 5.31’de kullanılan ve ResNet bloğu olarak kullanılan bloğun iç yapısı Şekil 5.37’de olduğu gibi güncellenmiştir. Mimari 181718 adet eğitilebilir parametre içermektedir.



Şekil 5.37 ResNet-3 blok yapısı

Şekil 5.38’de ResNet-3 modelinin bütün SNR seviyelerinde sinyaller içeren testi için sınıflandırma matrisi verilmiştir. QPSK ve 32PSK modülasyonlarını seçmeye eğilimli olduğu söylenebilir. 64-128-256QAM modülasyonlarındaki belirsizlik daha az olarak gözükmemektedir.



Şekil 5.38 ResNet-3 modeli sınıflandırma matrisi

Çizelge 5.12’de ResNet-3 mimarisinin detaylı performans raporu verilmiştir. Bütün SNR seviyelerinde sinyalleri içeren bu çizelgede 64-128-256QAM modülasyonları ortalama %60 başarımla göstermiştir. Bu skora en yakın diğer sonuç CNN-2 mimarisinde elde edilmiştir ve o mimarideki F1 skoru ortalaması yaklaşık %50’dir.

Çizelge 5.12 ResNet-3 mimarisi detaylı performans raporu

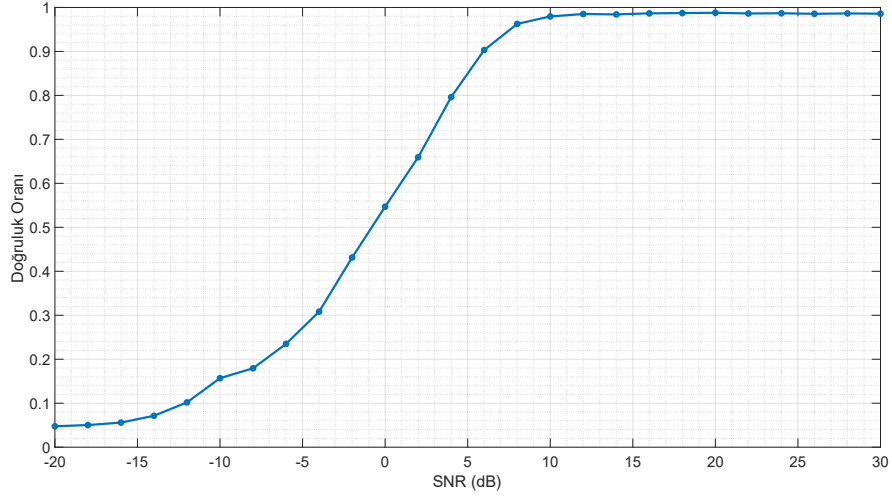
Modülasyon	Keskinlik	Duyarlılık	F1-Skoru
OOK	0.64	0.75	0.69
4ASK	0.92	0.63	0.75
8ASK	0.52	0.81	0.63
BPSK	0.89	0.71	0.79
QPSK	0.32	0.71	0.45
8PSK	0.86	0.59	0.7
16PSK	0.95	0.52	0.67
32PSK	0.28	0.66	0.39
16APSK	0.77	0.62	0.69
32APSK	0.76	0.62	0.68
64APSK	0.78	0.53	0.63
128APSK	0.92	0.52	0.66
16QAM	0.84	0.57	0.68
32QAM	0.86	0.54	0.66
64QAM	0.66	0.55	0.6
128QAM	0.68	0.5	0.58
256QAM	0.9	0.45	0.6
AM-DSB-WC	0.59	0.74	0.66
AM-DSB-SC	0.84	0.57	0.68
FM	0.62	0.83	0.71
GMSK	0.45	0.75	0.56
OQPSK	0.85	0.65	0.74
Ortalama	0.72	0.63	0.65

Çizelge 5.13'te ResNet-3 mimarisinin SNR'a göre F1 skoru değerlendirmesi verilmiştir. 10dB SNR seviyesinin üstünde 64-128-256 QAM modülasyonları yaklaşık %90 başarımlarına ulaşmış gözükmemektedir. 30dB SNR seviyesinde 64-128-256QAM modülasyonlarının ortalama F1 skoru %97 olmuştur. Bu skora en yakın sonuç CNN-2 mimarisinde alınmıştır ve o mimarideki ortalama F1 skoru %82'dir.

Çizelge 5.13 ResNet-3 mimarisi SNR'a göre F1 skoru değerlendirilmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.07	0.20	0.99	1.00	1.00	1.00
4ASK	0.02	0.19	0.75	1.00	1.00	1.00
8ASK	0.04	0.24	0.77	1.00	1.00	1.00
BPSK	0.04	0.06	1.00	1.00	1.00	1.00
QPSK	0.03	0.04	0.90	1.00	1.00	1.00
8PSK	0.01	0.01	0.16	1.00	1.00	1.00
16PSK	0.02	0.03	0.31	0.99	1.00	1.00
32PSK	0.03	0.04	0.21	1.00	1.00	1.00
16APSK	0.00	0.00	0.50	1.00	1.00	1.00
32APSK	0.00	0.00	0.47	1.00	1.00	1.00
64APSK	0.01	0.01	0.17	1.00	1.00	1.00
128APSK	0.00	0.00	0.12	1.00	1.00	1.00
16QAM	0.01	0.00	0.19	1.00	1.00	1.00
32QAM	0.02	0.02	0.12	1.00	1.00	1.00
64QAM	0.01	0.01	0.10	0.92	0.98	0.98
128QAM	0.02	0.02	0.04	0.93	0.97	0.96
256QAM	0.03	0.04	0.18	0.88	0.95	0.94
AM-DSB-WC	0.05	0.28	0.74	0.93	0.92	0.93
AM-DSB-SC	0.01	0.38	0.67	0.92	0.91	0.91
FM	0.06	0.43	1.00	1.00	1.00	1.00
GMSK	0.07	0.09	1.00	1.00	1.00	1.00
OQPSK	0.01	0.01	0.96	1.00	1.00	1.00

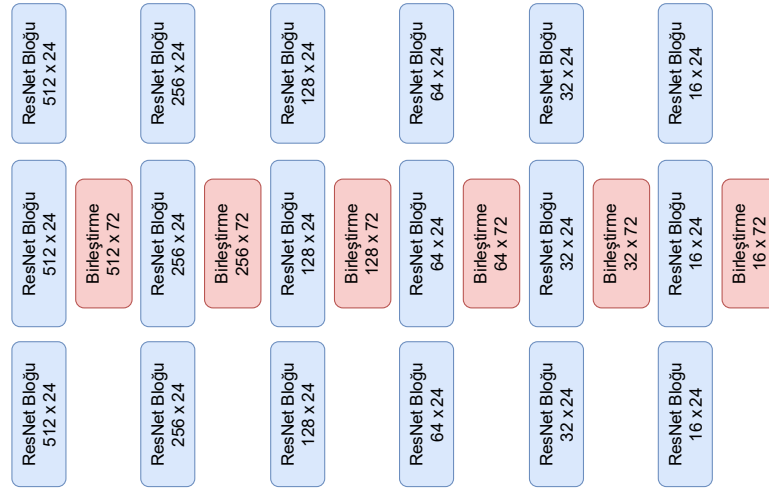
Şekil 5.39'da ResNet-3 mimarisinin SNR'a göre doğruluk oranı grafiği verilmiştir. Diğer modellerde de olduğu gibi 10dB seviyesi doyum noktası olarak gözükmektedir. Doyum noktasındaki doğruluk oranı yaklaşık %98 olarak gözükmektedir.



Şekil 5.39 ResNet-3 modeli SNR'a göre doğruluk oranı

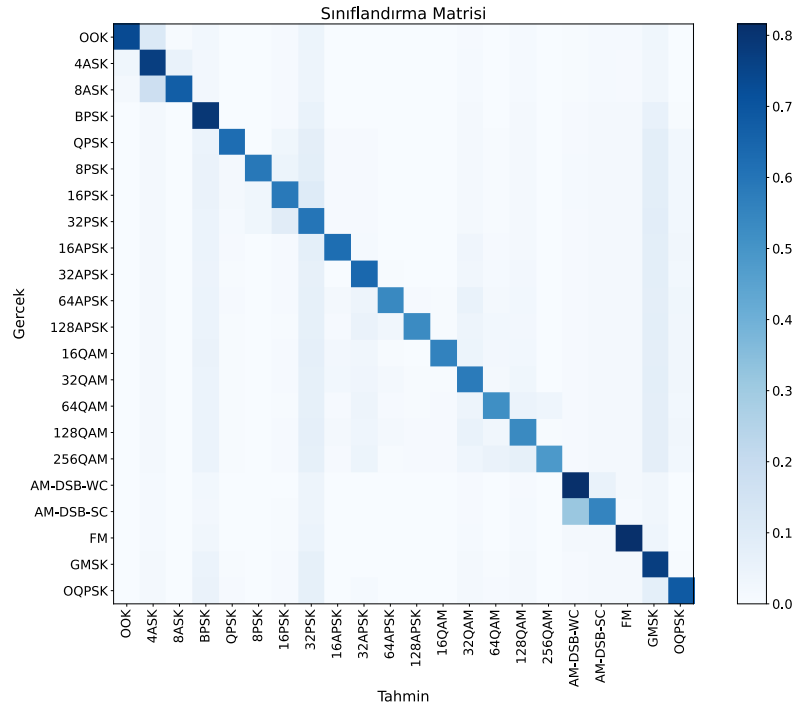
5.2.7 ResNet-4 mimarisi

ResNet-4 mimarisi CNN-3 ve ResNet-3 mimarilerinin birleşimi olması planlanarak kurulmuştur. ResNet mimarilerinin en iyisi ile CNN mimarilerinin en iyi sonuç veren yapılarının ortaklanması hedeflenmiştir. Şekil 5.31'de verilen ResNet-1 mimarisi iç yapısındaki iki evrişim bloğu arasında bulunan altılı ResNet bloğu yapısı değiştirilmiştir. Seri olarak bağlanan 6 resnet bloğunun yerine Şekil 5.40'da gösterilen CNN-3 mimarisinden örneklenerek kurulmuş yapı eklenmiştir. Seri olarak 6 ve her seri katmanda paralel olarak 3 ResNet bloğu kullanılmıştır. Bu ResNet bloklarının içerisindeki evrişim bloklarını filtre boyutları sırasıyla 2x1, 4x1 ve 8x1 olarak belirlenmiştir. Kullanılan mimari içerisinde 308054 adet eğitilebilir parametre bulunmaktadır.



Şekil 5.40 ResNet-4 mimarisi iç yapısı

Şekil 5.41’de ResNet-4 modelinin sınıflandırma matrisi verilmiştir. Model BPSK, 32PSK ve GMSK modülasyonlarını seçmeye eğilimli denilebilir. ResNet-3 mimarisindeki benzer bir 64-128-256 QAM sınıflandırması görülmektedir.



Şekil 5.41 ResNet-4 modeli sınıflandırma matrisi

Çizelge 5.14’te ResNet-4 modelinin detaylı performans durumu verilmiştir. ResNet-3 mimarisine çok yakın sonuçlar elde edilmiştir.

Çizelge 5.14 ResNet-4 modeli detaylı performans raporu

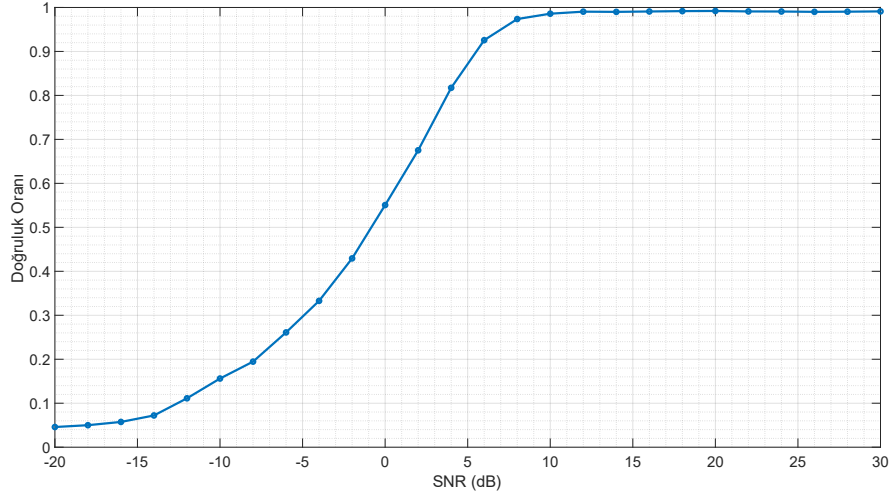
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.91	0.73	0.81
4ASK	0.58	0.77	0.66
8ASK	0.9	0.67	0.77
BPSK	0.47	0.79	0.59
QPSK	0.86	0.62	0.72
8PSK	0.91	0.59	0.71
16PSK	0.68	0.58	0.63
32PSK	0.31	0.6	0.41
16APSK	0.82	0.62	0.71
32APSK	0.67	0.64	0.65
64APSK	0.87	0.54	0.66
128APSK	0.92	0.53	0.67
16QAM	0.93	0.56	0.7
32QAM	0.53	0.58	0.56
64QAM	0.71	0.52	0.6
128QAM	0.55	0.53	0.54
256QAM	0.92	0.48	0.63
AM-DSB-WC	0.63	0.81	0.71
AM-DSB-SC	0.61	0.55	0.58
FM	0.73	0.82	0.77
GMSK	0.36	0.77	0.49
OQPSK	0.66	0.68	0.67
Ortalama	0.71	0.64	0.65

Çizelge 5.15'te ResNet-4 mimarisinin SNR'a göre F1 skoru değişimi verilmiştir. Diğer modellerde olduğu gibi 10 dB eşik seviyesi olarak görülmektedir. Bu seviyenin üstünde sınıflandırma başarımları çok artmaktadır.

Çizelge 5.15 ResNet-4 mimarisi SNR'a göre F1 skoru değerlendirilmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0.79	0.02	0.16	0.99	1.00	1.00
4ASK	0.69	0.05	0.29	0.76	1.00	1.00
8ASK	0.70	0.05	0.18	0.76	1.00	1.00
BPSK	0.65	0.05	0.10	1.00	1.00	1.00
QPSK	0.69	0.02	0.02	0.91	1.00	1.00
8PSK	0.71	0.00	0.00	0.33	1.00	1.00
16PSK	0.64	0.01	0.01	0.36	1.00	1.00
32PSK	0.59	0.02	0.03	0.20	1.00	1.00
16APSK	0.66	0.02	0.03	0.57	1.00	1.00
32APSK	0.68	0.00	0.00	0.47	1.00	1.00
64APSK	0.67	0.00	0.00	0.09	1.00	1.00
128APSK	0.66	0.00	0.00	0.09	1.00	1.00
16QAM	0.70	0.00	0.00	0.13	1.00	1.00
32QAM	0.58	0.04	0.05	0.22	1.00	1.00
64QAM	0.61	0.01	0.02	0.13	0.98	1.00
128QAM	0.49	0.05	0.06	0.06	0.97	0.99
256QAM	0.60	0.02	0.02	0.09	0.95	0.98
AM-DSB-WC	0.69	0.04	0.32	0.72	0.93	0.94
AM-DSB-SC	0.65	0.03	0.21	0.66	0.93	0.93
FM	0.79	0.04	0.52	1.00	1.00	1.00
GMSK	0.48	0.07	0.09	1.00	1.00	1.00
OQPSK	0.67	0.01	0.03	0.95	1.00	1.00

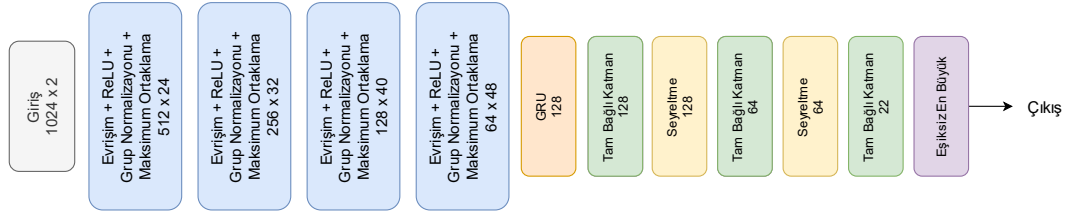
Şekil 5.42'de ResNet-4 modelinin SNR'a göre doğruluk oranı grafiği verilmiştir. 10 dB doyum noktasından sonra modelin başarımlı oranı %99 seviyesinin üzerine ulaşmıştır. İncelenen modeller içerisinde ulaşılan en yüksek doğruluk oranıdır. ResNet-3 mimarisinde %98 doğruluk oranına ulaşılırken kullanılan model de yaklaşık 180000 parametre bulunurken, ResNet-4 modelinde ise yaklaşık 300000 adet parametre bulunmaktadır.



Şekil 5.42 ResNet-4 modeli SNR'a göre doğruluk oranı

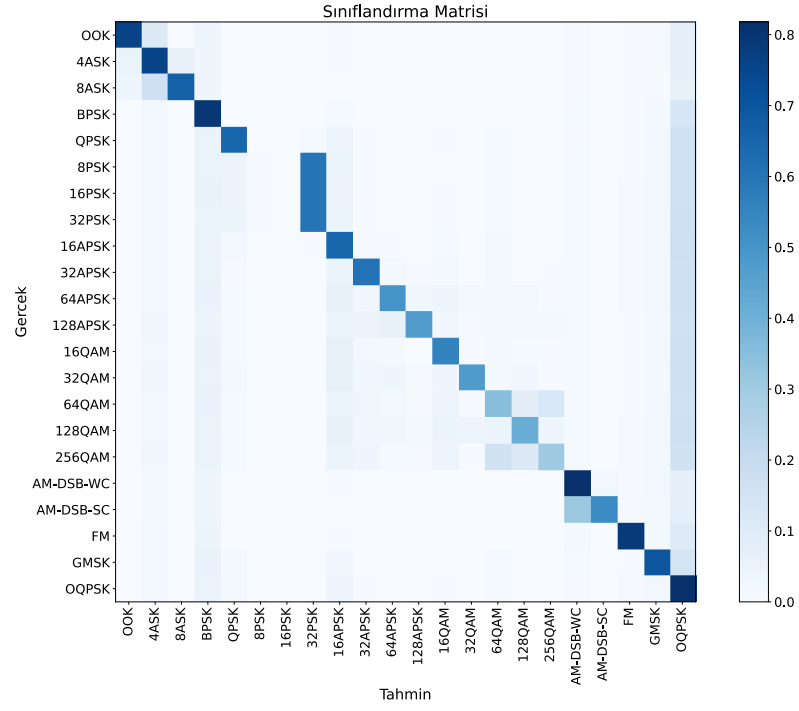
5.2.8 GRU mimarisi

Kurulan mimari Şekil 5.43'te verilmiştir. Bu ağ yapısı içerisinde 103014 adet eğitilebilir parametre bulunmaktadır.



Şekil 5.43 GRU mimarisi genel yapısı

Şekil 5.44'de GRU modelinin sınıflandırma matrisi verilmiştir. Modele giriş olarak verilen 8-16-32PSK modülasyonlarının tamamı 32PSK olarak sınıflandırılmıştır. Ayrıca OQPSK modülasyonunu seçmeye eğilimli olduğu görülmektedir. Diğer modellerde olduğu 64-128-256 QAM modülasyonlarının sınıflandırılması problemlidir.



Şekil 5.44 GRU modeli sınıflandırma matrisi

Çizelge 5.16’da GRU modelinin detaylı performans sonuçları verilmiştir. 8-16-32 PSK modülasyonlarının sınıflandırmasında karşılaşılan problem F1 skorlarında görülmektedir. 8-16 PSK modülasyonlarının F1 skoru 0 olarak gelmiştir.

Çizelge 5.16 GRU modeli detaylı performans raporu

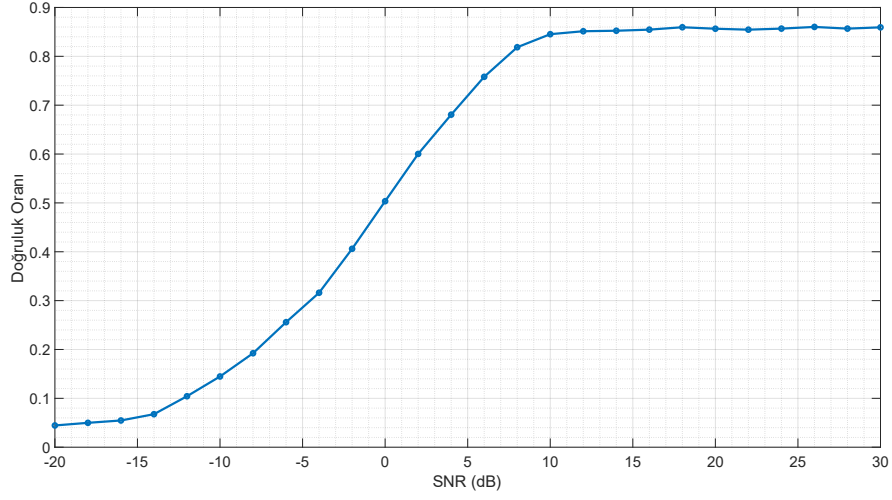
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.89	0.75	0.82
4ASK	0.51	0.75	0.61
8ASK	0.91	0.66	0.77
BPSK	0.44	0.8	0.56
QPSK	0.7	0.65	0.67
8PSK	0.35	0	0.01
16PSK	0	0	0
32PSK	0.33	0.6	0.42
16APSK	0.47	0.65	0.54
32APSK	0.7	0.61	0.65
64APSK	0.72	0.5	0.59
128APSK	0.86	0.47	0.61
16QAM	0.67	0.56	0.61
32QAM	0.86	0.48	0.62
64QAM	0.48	0.35	0.41
128QAM	0.61	0.41	0.49
256QAM	0.61	0.3	0.4
AM-DSB-WC	0.61	0.82	0.7
AM-DSB-SC	0.95	0.53	0.68
FM	0.86	0.79	0.82
GMSK	0.64	0.7	0.67
OQPSK	0.22	0.81	0.34
Ortalama	0.61	0.55	0.55

Çizelge 5.17’de GRU mimarisinin SNR’a göre F1 skoru değerlendirmesi gösterilmiştir. SNR seviyesi ne kadar yükselirse yükselsin 8-16 PSK modülasyonlarının F1 skoru 0 olarak kalmıştır. 8-16 PSK modülasyonlarının tamamı 32PSK olarak sınıflandırıldığı için 32 PSK modülasyonunun da F1 skoru düşük kalmıştır. Diğer modülasyon türleri ise SNR’ın yükselmesi ile iyileşme göstermişlerdir. 64-128-256 QAM modülasyonlarının 30dB SNR seviyesindeki ortalama F1 skoru yaklaşık %71 olmuştur.

Çizelge 5.17 GRU mimarisi SNR'a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0	0.24	0.99	1	1	1
4ASK	0.07	0.34	0.76	1	1	1
8ASK	0	0.02	0.75	1	1	1
BPSK	0.07	0.1	1	1	1	1
QPSK	0	0.01	0.59	1	1	1
8PSK	0	0	0	0.02	0.01	0.02
16PSK	0	0	0	0	0	0
32PSK	0	0	0.39	0.5	0.51	0.5
16APSK	0.03	0.07	0.36	1	1	1
32APSK	0	0.01	0.42	1	1	1
64APSK	0	0	0.15	0.91	0.95	0.96
128APSK	0	0	0.03	0.9	0.94	0.95
16QAM	0	0	0.17	0.99	1	1
32QAM	0	0	0.05	0.93	0.96	0.97
64QAM	0.01	0.02	0	0.63	0.7	0.72
128QAM	0	0	0.07	0.72	0.8	0.8
256QAM	0	0	0.15	0.56	0.63	0.62
AM-DSB-WC	0.04	0.54	0.77	0.93	0.94	0.94
AM-DSB-SC	0	0.01	0.66	0.92	0.93	0.93
FM	0.04	0.42	1	1	1	1
GMSK	0.05	0.06	1	1	1	1
OQPSK	0.08	0.11	0.81	1	1	1

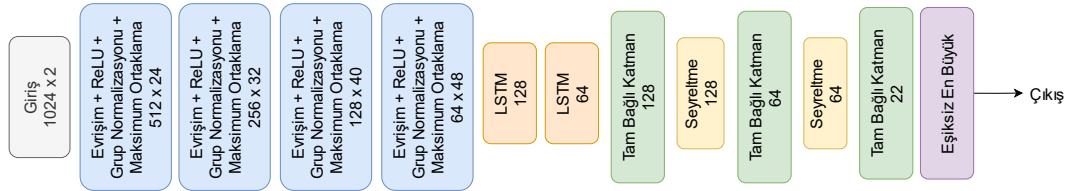
Şekil 5.45'te GRU modelinin SNR'a göre doğruluk oranı grafiği verilmiştir. Diğer modellerde de olduğu gibi 10dB SNR seviyesinde doyuma ulaşmıştır. Doyum noktasındaki doğruluk oranının yaklaşık %86 olduğu görülmektedir. Doyum noktasındaki başarımlar oranının düşük olmasının en büyük nedeni 8-16-32PSK modülasyonlarının sınıflandırılmamasıdır.



Şekil 5.45 GRU modeli SNR'a göre doğruluk oranı

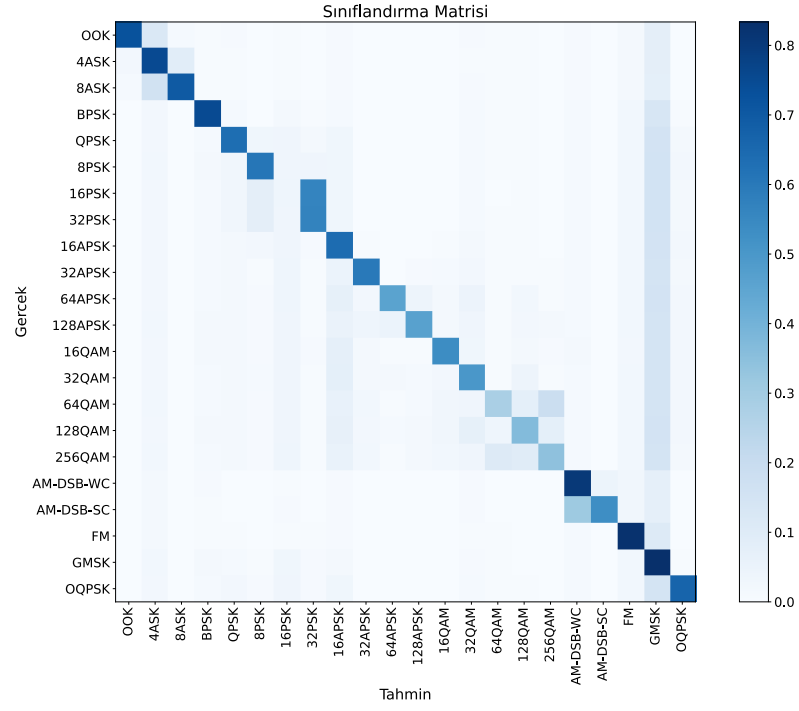
5.2.9 LSTM mimarisi

Kurulan ağ yapısı Şekil 5.46'da verilmiştir. GRU mimarisiyle çok benzer bir yapıda kurulmuştur. GRU katmanı yerine 2 adet LSTM katmanı eklenmiştir. 166502 adet eğitilebilir parametre içermektedir.



Şekil 5.46 LSTM mimarisi genel yapısı

Şekil 5.47'de LSTM mimarisine ait sınıflandırma matrisi verilmiştir. 16PSK ve 32PSK modülasyonlarında karışıklık görülmektedir. Sisteme verilen 16PSK sinyalleri 32PSK olarak sınıflandırılmış gözükmemektedir. Ayrıca model GMSK modülasyonunu seçmeye eğilimlidir.



Şekil 5.47 LSTM modeli sınıflandırma matrisi

Çizelge 5.18’de LSTM modelinin detaylı performans raporu verilmiştir. 16PSK modülasyonun F1 skoru sınıflandırma matrisinin de gösterdiği gibi çok düşük gelmiştir. Ayrıca modelin GMSK modülasyonunu seçmeye eğilimi GMSK F1 skorunu da düşürmüştür.

Çizelge 5.18 LSTM modeli detaylı performans raporu

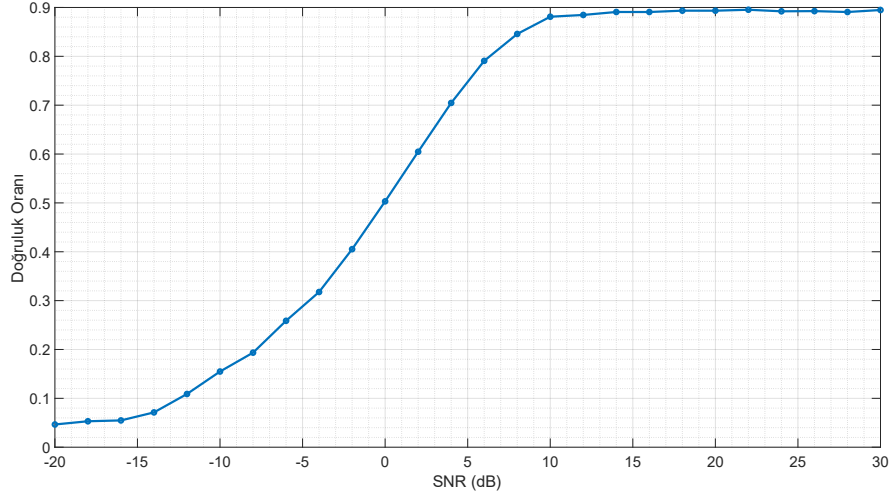
Modülasyon	Kesinlik	Duyarlılık	F1-Skoru
OOK	0.95	0.72	0.82
4ASK	0.52	0.75	0.62
8ASK	0.87	0.7	0.78
BPSK	0.77	0.75	0.76
QPSK	0.69	0.63	0.66
8PSK	0.68	0.61	0.64
16PSK	0.07	0.04	0.05
32PSK	0.45	0.56	0.5
16APSK	0.47	0.64	0.54
32APSK	0.77	0.6	0.67
64APSK	0.83	0.46	0.59
128APSK	0.8	0.46	0.59
16QAM	0.78	0.53	0.63
32QAM	0.55	0.5	0.53
64QAM	0.57	0.28	0.38
128QAM	0.53	0.37	0.43
256QAM	0.54	0.34	0.42
AM-DSB-WC	0.6	0.8	0.69
AM-DSB-SC	0.92	0.53	0.67
FM	0.63	0.83	0.72
GMSK	0.24	0.83	0.37
OQPSK	0.72	0.67	0.69
Ortalama	0.63	0.57	0.58

Çizelge 5.19’da LSTM mimarisinin SNR’a göre F1 skoru verilmiştir. 16PSK modülasyonu 30dB SNR seviyesinde bile performans gösterememiştir. Bu mimari 16PSK modülasyonunu sınıflandıramamaktadır. Diğer modülasyonlar SNR seviyesi arttıkça %90 değerlerine ulaşmıştır. 64-128-256QAM modülasyonları ise diğer modellerde de olduğu gibi tam başarıma ulaşamamıştır.

Çizelge 5.19 LSTM mimarisi SNR’a göre F1 skoru değerlendirmesi

Modülasyon	-20dB	-10dB	0dB	10dB	20dB	30dB
OOK	0	0.11	0.99	1	1	1
4ASK	0.06	0.35	0.73	1	1	1
8ASK	0	0.16	0.75	1	1	1
BPSK	0.04	0.09	1	1	1	1
QPSK	0.03	0.05	0.66	1	1	1
8PSK	0	0	0.38	0.99	1	0.99
16PSK	0.02	0.08	0	0	0	0
32PSK	0.01	0.03	0.23	0.68	0.68	0.67
16APSK	0.06	0.04	0.32	1	1	1
32APSK	0.01	0.01	0.42	1	1	1
64APSK	0	0	0.01	0.89	0.94	0.94
128APSK	0	0	0.14	0.88	0.93	0.93
16QAM	0	0	0.14	0.99	1	1
32QAM	0.04	0.04	0.18	0.89	0.95	0.95
64QAM	0.02	0.01	0	0.56	0.65	0.68
128QAM	0.03	0.02	0	0.67	0.74	0.73
256QAM	0	0	0.12	0.57	0.61	0.63
AM-DSB-WC	0.05	0.52	0.75	0.91	0.92	0.92
AM-DSB-SC	0	0.09	0.65	0.91	0.92	0.91
FM	0.06	0.42	1	1	1	1
GMSK	0.08	0.11	1	1	1	1
OQPSK	0	0.01	0.87	1	1	1

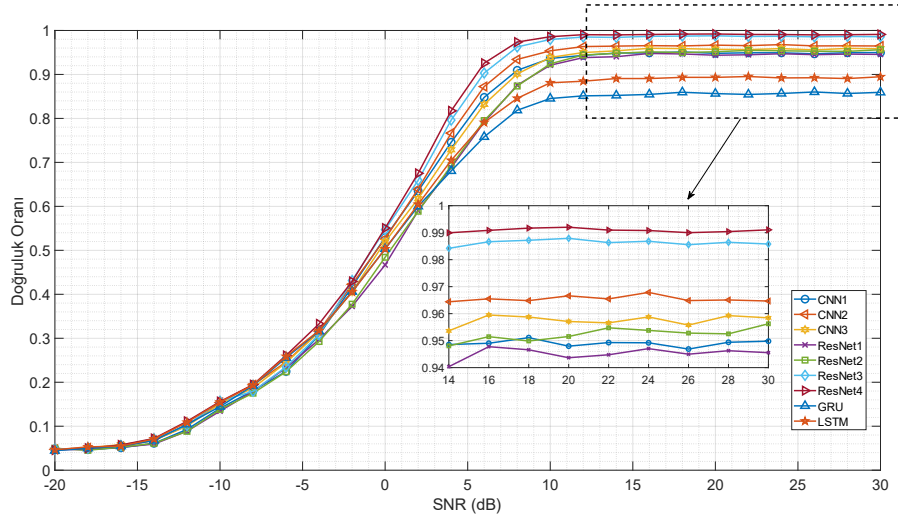
Şekil 5.48’de LSTM modelinin SNR’a göre doğruluk oranı grafiği verilmiştir. 10dB SNR değerinde doyum noktası olan %90 doğruluk oranına ulaşmıştır.



Şekil 5.48 LSTM modeli SNR'a göre doğruluk oranı

5.2.10 Mimarilerin doğruluk oranı karşılaştırması

Denenen bütün mimarilerin SNR'a göre doğruluk oranı grafiği Şekil 5.49'da verilmiştir. 10dB SNR seviyesi bütün mimariler için doyum noktası olarak belirtilebilir. En iyi performans %99 doğruluk oranı ile ResNet4 mimarisinde elde edilmiştir. 14dB - 30dB SNR seviyesinde performans grafiklerini daha detaylı incelemek için büyütülmüş şekilde verilmiştir. ResNet3 mimarisi ile ResNet4 mimarisi arasındaki %1'den daha düşük bir fark olduğu görülmektedir. ResNet3 ve ResNet4 mimarilerinin arkasında yaklaşık %97 doğruluk oranı ile CNN2 modeli bulunmaktadır. %94 ile %96 gibi küçük bir aralıkta sıkışmış 4 farklı mimari mevcuttur. GRU ve LSTM modelleri %90 doğruluk oranının altında kalmıştır.

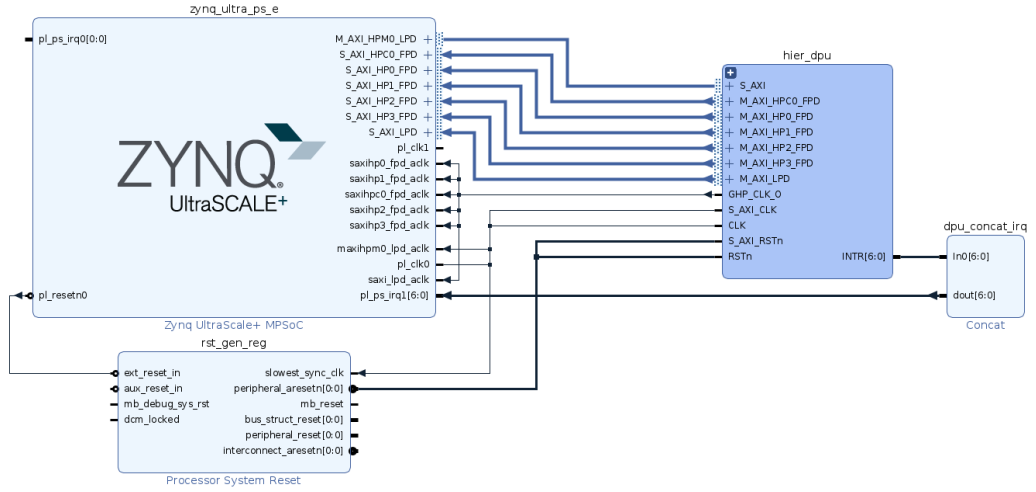


Şekil 5.49 Mimarilerin doğruluk oranı grafiği

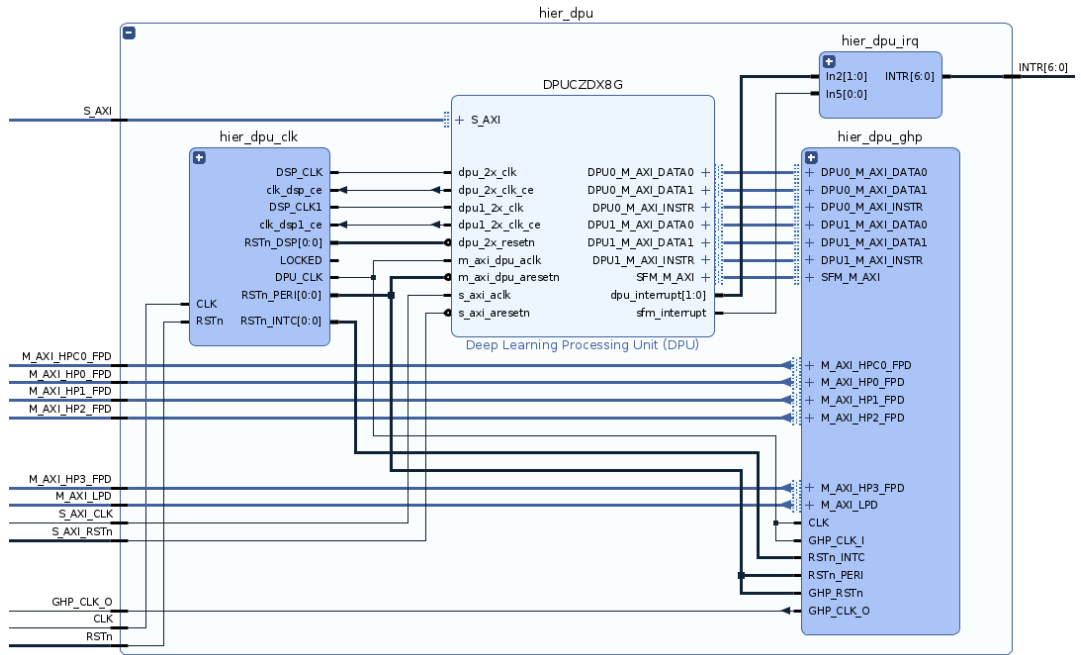
5.3 Mimarilerinin FPGA Üzerine Aktarılması

5.3.1 FPGA test ortamının kurulması

Tasarlanan mimarilerin testi için Xilinx ürünü ZCU104 geliştirme kartı kullanılmıştır. Kullanılan test kartının ilklendirilmesi için yine Xilinx tarafından sağlanan ilklendirme dosyaları kullanılmış, (Anonymous 2021c) ile belirtilen süreçler uygulanmıştır. Bu ilklendirme dosyaları ile içerisinde Xilinx derin öğrenme işlemci birimi (DPU) bulunan test ortamı oluşturulmuştur. Oluşturulan yapı Şekil 5.50 ve Şekil 5.51 ile gösterilmiştir. Şekil 5.50 ile gösterilen blok yapıda, FPGA içerisinde çalışan yapının üst seviye gösterimi yapılmıştır. Zynq işlemci birimi ve DPU bloklarının üst seviyeden bağlantısı gösterilmiştir. Şekil 5.51’de ise DPU bloğunun bağlantıları üzerindeki ayrıntılar verilmiştir. Kurulan bu yapının kaynak kullanımı 55000 LUT, 100000 FF, 250 BRAM ve 690 DSP şeklinde gerçekleşmiştir. ZCU104 kartı ile sadece CNN mimarisine ait yapılar çalıştırılabilmektedir. RNN tabanlı mimarilerinde aynı yöntemle FPGA üzerinde çalıştırılabilmesi için Xilinx’in daha üst seviye geliştirme kartlarına ihtiyaç duyulmaktadır. RNN mimarisine ait denenen modellerin başarımlarının görece daha düşük olması da göz önüne alınarak GRU ve LSTM mimarileri çalışmanın devamında incelenmemiştir.



Şekil 5.50 HDL blok yapısı üst seviye yapısı



Şekil 5.51 HDL blok yapısı DPU kullanımı

5.3.2 Mimarinin nicemlenmesi ve performans analizi

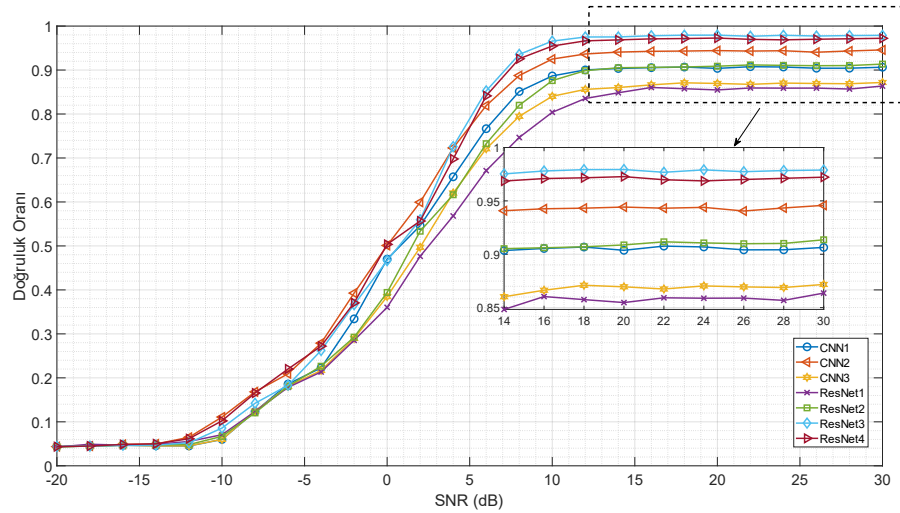
İncelenen mimarilerin içerisinde bulunan bütün parametreler kayan noktalı olarak yer almaktadır. Bu da her bir parametrenin 32-bit olarak kaynak kullanmasını gerektirmektedir. Mimarileri FPGA üzerine aktarmadan önce bölüm 4.1 ve 4.2’de incelenen nicemleme ve FPGA üzerinde kaynak kullanımı konuları dikkate alınarak,

mimari içerisinde bulunan 32-bit ile ifade edilen parametreler 8-bit olacak şekilde nicemlenmiştir. Nicemleme işlemi Xilinx Vitis-AI ve TensorFlow kütüphaneleri ile yapılmıştır. Şekil 5.52’de nicemlenmiş mimarilerin doğruluk oranı grafiği verilmiştir. Nicemlenmiş modellerde beklenildiği üzere bir performans kaybı yaşanmıştır. 30dB SNR seviyesinde yaşanan performans kaybı Çizelge 5.20’de gösterilmiştir.

Çizelge 5.20 Mimarilerin 30dB SNR seviyesindeki nicemlenmiş ve nicemlenmemiş durumlardaki başarımları

	Nicemlenmemiş	Nicemlenmiş
CNN1	%95.0	%90.6
CNN2	%96.8	%94.4
CNN3	%95.8	%87.2
ResNet1	%94.5	%86.4
ResNet2	%95.6	%91.4
ResNet3	%98.5	%97.8
ResNet4	%99.1	%97.2

Şekil 5.52’deki grafikte gösterildiği üzere yaklaşık 10 dB seviyesi nicemlenmiş mimariler için de doyum noktası olarak görülmektedir. Her mimari farklı oranlarda performans kaybı yaşamıştır. En iyi performansın ResNet4 modelinden ResNet3 modeline geçtiği görülmektedir. Yaklaşık %3 başarımları farkıyla bunların arkasında CNN2 modeli bulunmaktadır.



Şekil 5.52 Nicemlenmiş mimarilerin doğruluk oranı grafiği

5.3.3 FPGA test sonuçları

İncelenen mimarilerin FPGA üzerinde çıktı üretme hızları ve başarımları karşılaştırılmıştır. Çizelge 5.21’de SNR seviyesine göre başarımları verilmiştir. Genel olarak başarımları mimarilerin eğitim sonuçlarına benzer çıkmıştır. Her testte 10000 veri kullanılmıştır.

Çizelge 5.21 Mimarilerin FPGA üzerinde yapılan testlerinde başarımlarının SNR(dB) seviyesine göre incelenmesi

	-20	-10	0	2	4	6	8	10	20	30
CNN1	0.04	0.05	0.43	0.52	0.60	0.69	0.78	0.82	0.84	0.85
CNN2	0.04	0.12	0.47	0.55	0.67	0.77	0.85	0.89	0.91	0.91
CNN3	0.04	0.06	0.38	0.50	0.61	0.71	0.78	0.81	0.85	0.85
ResNet1	0.04	0.07	0.36	0.48	0.58	0.67	0.75	0.81	0.86	0.87
ResNet2	0.05	0.06	0.39	0.53	0.61	0.73	0.82	0.88	0.91	0.91
ResNet3	0.05	0.08	0.47	0.56	0.72	0.85	0.93	0.97	0.98	0.98
ResNet4	0.04	0.10	0.51	0.56	0.70	0.85	0.94	0.97	0.98	0.98

Mimarilerin tamamı aynı FPGA bloklarını kullanmaktadır. Her mimari için farklı bir FPGA tasarımı yoktur. Mimari içerisindeki parametre sayısı, katmanların sayısı ve kurulan mimarinin yapısı çıktı üretme hızını etkilemektedir. Çizelge 5.22’de mimarilerin çıktı üretme hızları verilmiştir.

Çizelge 5.22 Mimarilerin çıktı üretme hızları

	test/s
CNN1	1203
CNN2	1094
CNN3	678
ResNet1	500
ResNet2	458
ResNet3	459
ResNet4	174

ResNet-3 ve ResNet-4 mimarileri yaklaşık olarak aynı başarımlarına sahip olsalar da çıktı üretme hızları birbirinden çok farklıdır. Resnet-3 mimarisi yaklaşık 2.5 kat daha hızlı çalışmaktadır. CNN-2 mimarisi ResNet-3 ve ResNet-4 mimarilerine göre yaklaşık %5 daha düşük başarımlarına sahipken, çıktı üretme hızı açısından ResNet-3 mimarisine göre yaklaşık 2.5 kat daha hızlıdır.

Mimarilerin kuruluş yapıları ve parametre sayıları çıktı üretme hızını etkilemektedir. CNN1 ve CNN2 modelleri birbirleriyle tamamen aynı yapıya sahiptir. Aralarındaki tek fark evrişim katmanlarında bulunan filtrelerin sayılarıdır. CNN2 modelinde daha fazla sayıda filtre bulunmaktadır ve bu da parametre sayısını arttırmaktadır. CNN1 mimarisi içerdiği yaklaşık 157 bin parametre ile saniyede 1203 çıktı üretebilirken, CNN2 mimarisi 218 bin parametre ile saniyede 1094 çıktı üretebilmiştir. CNN-3 mimarisi bu mimarilere benzese de içerisindeki paralel evrişim katmanları çıktı üretme hızını etkilemektedir. 152 bin parametre içermesine rağmen saniyede sadece 678 çıktı üretebilmektedir. ResNet4 mimarisi de içerdiği 308 bin parametre ile en büyük mimaridir ve saniyede 174 çıktı üretebilmektedir. Hem parametre sayısının çok olması hem de içerisinde paralel evrişim blokları bulundurması çıktı hızını ciddi ölçüde etkilemiştir.

6. TARTIŞMA ve SONUÇ

Bu tezde modülasyon sınıflandırma amaçlı olarak derin öğrenme mimarilerinin FPGA üzerinde kullanımı incelenmiştir. Veri seti içerisindeki OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK modülasyon sinyallerinin sınıflandırması yapılmıştır. Modülasyon türünün sınıflandırması amacıyla 9 farklı derin öğrenme mimarisi kurulmuştur.

SNR seviyesinin başarımlarına olan etkisi incelenmiştir. Tasarlanan mimarilerin tamamında başarımları SNR seviyesiyle birlikte yükselmektedir. Bütün modülasyon sinyalleri için 10dB SNR seviyesi başarımlarının doyuma ulaştığı seviye olarak belirlenmiştir. Görece daha basit modülasyon türlerinden olan OOK, 4ASK, 8ASK, BPSK, QPSK, FM, GMSK ve OQPSK gibi sinyaller 0dB SNR seviyesinde bile yaklaşık %80 başarımları sağlayabilmektedir. Daha karmaşık modülasyon türlerinden olan 64APSK, 128APSK, 32QAM, 64QAM, 128QAM ve 256QAM sinyalleri ise 0dB SNR seviyesinde çok düşük başarımları göstermiştir. Modülasyon şeması büyüdükçe sembollerin ayırt edilebilirliği düşmektedir ve sınıflandırma yapılamamaktadır. Bu büyük modülasyon şemaları birlikte incelendiğinde yıldız küme şemalarının üst üste geldiği görülmektedir. Özellikle 64QAM, 128QAM ve 256QAM gibi modülasyon şemalarının sembolleri birbirleri içerisinde karışmaktadır. Veri setinin uzunluğundan da kaynaklı olarak bu sembollerin birbirlerinden ayrıştırılamaması beklenen bir sonuçtur.

9 farklı derin öğrenme mimarisi kurulmuş ve bunların performansları karşılaştırılmıştır. Benzetim ortamında yapılan testlerde nicemleme yapılmadan incelenen mimariler içinde en yüksek performansı ResNet-3 ve ResNet-4 modelleri vermiştir. ResNet-4 modeli %1'den daha düşük bir farkla daha iyi sonuç üretmiştir. En düşük sonucu ise GRU mimarisi vermiştir. Yüksek performans gösteren mimarilerde parametre nicemlemesi yapılmış ve modeller tekrar test edilmiştir. Yine benzetim ortamında yapılan testlerde bu sefer en iyi sonucu ResNet-3 mimarisi göstermiştir. İncelenen CNN-1-2-3 ve ResNet-1-2-3-4 mimarileri genel olarak bakıldığında evrişimli sinir ağı yapısına girmektedir. GRU ve LSTM mimarileri ise öz-yinelemeli sinir ağı yapısındadır. Literatürde evrişimli sinir ağı özellikle uzamsal bağılılığı olan görüntü sınıflandırma, bilgisayarlı görü gibi

problemler için avantajlı görülmektedir. Diğer yandan özyinelemeli sinir ağları ise zamansal bağımlılığı olan sinyal işleme problemleri için avantajlıdır. Modülasyon sınıflandırma problemi zamansal bağımlılığı olan bir veri grubu içerdiği için özyinelemeli sinir ağları ile çözülmesi beklenmektedir. Ancak veri setinin uzunluğu arttıkça özyinelemeli ağların başarımlar performansı olumsuz etkilenmektedir. Yapılan çalışmada veri setinin uzunluğu 1024 iken, literatürde özyinelemeli sinir ağları için veri setinin uzunluğunun 128'den fazla olmaması gerektiği vurgulanmaktadır. Ayrıca bu çalışma kapsamında kullanılan geliştirme kartı da özyinelemeli sinir ağlarının gerçekleştirilmesi için yeterli değildir.

Bu çalışmada derin öğrenme tabanlı yöntemler kullanılmıştır. Literatürde öznitelik tabanlı ve olabilirlik tabanlı geleneksel yöntemler de bulunmaktadır. Bu metotlar ile başarılı sınıflandırmalar yapılabilmektedir. Ancak veri seti içerisindeki modülasyon çeşitliliği arttıkça başarımlar oranları düşmektedir. İncelenen veri seti içerisinde 22 farklı modülasyon türü bulunmaktadır. Her modülasyon şemasını sınıflandırabilecek şekilde özniteliklerin çıkarılması noktasında geleneksel yöntemler dezavantajlıdır. Daha az çeşitliliğe sahip sınıflandırma uygulamaları için geleneksel yöntemler de uygulanabilir.

Nicemlenen mimariler Xilinx Vitis-AI ve Tensorflow kütüphaneleri kullanılarak FPGA geliştirme kartı üzerinde çalıştırılmıştır. Genel olarak benzetim ortamında elde edilen sonuçlara benzer sonuçlar çıkmıştır. ResNet-3 ve ResNet-4 modelleri birbirleriyle yaklaşık olarak aynı performansı üretmiştir. Bu mimarilerden sonra en başarılı sonucu CNN-2 mimarisi göstermiştir. Mimarilerin çıktı üretme hızları ayrıca karşılaştırılmıştır. En yavaş çıktı üreten model ResNet-4, en hızlı çıktı üreten model de CNN-1 modeli olmuştur. CNN-2 modeli ise hem en hızlı 2.mimari olması hem de başarımlar oranı olarak ResNet-3 mimarisinden ortalamada yalnızca %5 geride olması nedeniyle en optimum mimari olarak görülmektedir.

Bu tezde uygulanan yöntemle derin öğrenme mimarilerinin FPGA üzerinde gerçekleştirilmesi için üst düzey FPGA tasarımı veya sayısal tasarım bilgisine ihtiyaç yoktur. Derin öğrenme problemleri için bilgisayar ortamında tasarım yapar gibi tasarım yapıp FPGA ortamına aktarılabilir. HDL bilgisi gerektirmemektedir. Uygulanan yöntem basitlik açısından avantajlıdır.

Sonuç olarak, bu tez çalışmasında derin öğrenme mimarilerinin FPGA üzerinde gerçekleştirilmesi modülasyon sınıflandırma problemi üzerinden gösterilmiştir. Klasik derin öğrenme yöntemleriyle oluşturulan farklı derin öğrenme mimarileri FPGA üzerinde çalıştırılmıştır. Modülasyon sınıflandırma başarımlarını arttıran etmenler olarak SNR seviyesi ve modülasyon şemasının basitliği olduğu gösterilmiştir.

Bu tez çalışmasının devamında yapılabilecek geliştirmeler içerisinde, kurulan FPGA mimarisini her mimari için ortak olacak şekilde bir tasarım yerine mimariye özel bir blok tasarımı yapıp daha az kaynak kullanır hale getirmesi yer almaktadır. Bu çalışmada kullanılan veri seti 1 sembolü 8 örnekle ifade etmektedir. Kurulan mimarilerin farklı sembol örnek oranına sahip veri setleri için göstereceği performansın incelenmesi de gelecek çalışmalar içerisinde yer almaktadır. Ayrıca veri setindeki her bir kayıt 1024 örnek uzunluğundadır. Daha uzun ve daha kısa veri setleri kullanarak başarımlarındaki değişim gözlemlenebilir. Daha uzun veri setinin kullanılması karmaşık modülasyon şemalarının daha düşük SNR seviyelerindeki başarımlarını arttırabileceği düşünülmektedir. İncelenen modülasyon sınıflandırma problemi dışında derin öğrenme kullanılarak görüntü sınıflandırma, video sınıflandırma gibi birçok problem de FPGA ortamında gerçekleştirilebilir.

KAYNAKLAR

- Aida-Zade, K., Mustafayev, E., & Rustamov, S. (2019). Comparison of deep learning in neural networks on CPU and GPU-based frameworks. 11th IEEE International Conference on Application of Information and Communication Technologies, AICT 2017 - Proceedings.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2018). Understanding of a convolutional neural network. Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, 2018-January, 1–6.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. In: Electronics (Switzerland) (Vol. 8, Number 3). MDPI AG.
- Alpaydin, E. 2020. Introduction to machine learning. The MIT Press, USA
- Amagasaki, M., Shibata, Y., Amagasaki, M., & Shibata, Y. (2018). FPGA Structure. Principles and Structures of FPGAs, 47–86.
- Anonymous. 2021a. Web Sitesi: <https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html> Erişim Tarihi: 5.12.2021
- Anonymous. 2021b. Web Sitesi: https://github.com/Xilinx/Vitis-AI-Tutorials/tree/master/Design_Tutorials/10-RF_modulation_recognition Erişim Tarihi: 5.12.2021
- Anonymous. 2021c. Web Sitesi: https://www.xilinx.com/html_docs/xilinx2019_2/vitis_doc/installation.html Erişim Tarihi: 19.12.2021
- Baldi, M., Chiaraluce, F., de Angelis, A., Marchesani, R., & Schillaci, S. (2012). A comparison between APSK and QAM in wireless tactical scenarios for land mobile systems. Eurasip Journal on Wireless Communications and Networking, 2012(1), 1–14
- Baldominos, A., Saez, Y., & Isasi, P. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. Applied Sciences 2019, Vol. 9, Page 3169, 9(15), 3169.
- Benko, A., & Sik Lányi, C. (2009) History of Artificial Intelligence, In: Encyclopedia of Information Science and Technology, Second Edition. (2010)
- Dileep, P., Das, D., & Bora, P. K. (2020). Dense layer dropout based CNN architecture for automatic modulation classification. 26th National Conference on Communications, NCC 2020.
- Dobre, O. A., Bar-Ness, Y., & Su, W. (2004). Robust QAM modulation classification algorithm using cyclic cumulants. 2004 IEEE Wireless Communications and Networking Conference, WCNC 2004, 2, 745–748.
- Fu, R., Zhang, Z., & Li, L. (2017). Using LSTM and GRU neural network methods for traffic flow prediction. Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016, 324–328.

- Gama, J., & de Carvalho, A. C. P. L. F. (2009). Machine Learning In: Encyclopedia of Information Science and Technology, Second Edition. (2010)
- Guan, H. B., Ye, C. Z., & Li, X. Y. (2004). Modulation classification based on spectrogram. Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 6, 3551–3556.
- Guo, Y., Sun, L., Zhang, Z., & He, H. (2019). Algorithm Research on Improving Activation Function of Convolutional Neural Networks. Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019, 3582–3586.
- Haenlein, M., & Kaplan, A. (2019). A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence: 61(4), 5–14.
- Haykin, S. (2008). Neural Networks and Learning Machines. Pearson Prentice Hall New Jersey USA 936 pLinks, 3, 906.
- Hazza, A., Shoaib, M., Alshebeili, S. A., & Fahad, A. (2013). An overview of feature-based methods for digital modulation classification. 2013 1st International Conference on Communications, Signal Processing and Their Applications, ICCSPA 2013.
- Huang, C., Ni, S., & Chen, G. (2017). A layer-based structured design of CNN on FPGA. Proceedings of International Conference on ASIC, 2017-October, 1037–1040.
- Huang, C. Y., & Polydoros, A. (1995). Likelihood Methods For MPSK Modulation Classification. IEEE Transactions on Communications, 43(234), 1493–1504.
- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (pp. 448–456). PMLR. <https://proceedings.mlr.press/v37/ioffe15.html>
- Karahan, S. N., & Kalaycioglu, A. (2020). Deep Learning Based Automatic Modulation Classification with Long-Short Term Memory Networks. 2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings.
- Lee, H., Byun, Y., Hwang, S., Lee, S., & Lee, Y. (2019). Fixed-Point Quantization of 3D Convolutional Neural Networks for Energy-Efficient Action Recognition. Proceedings - International SoC Design Conference 2018, ISOC 2018, 129–130.
- Naraan, S., & Tagliarini, G. (2005). An analysis of overfitting in MLP networks. Proceedings of the International Joint Conference on Neural Networks, 2, 984–988.
- O'Shea, T. J., Roy, T., & Clancy, T. C. (2018). Over-the-Air Deep Learning Based Radio Signal Classification. IEEE Journal on Selected Topics in Signal Processing, 12(1), 168–179.
- Park, C. S., Choi, J. H., Nah, S. P., Jang, W., & Kim, D. Y. (2008). Automatic modulation recognition of digital signals using wavelet features and SVM. International Conference on Advanced Communication Technology, ICACT, 1, 387–390.
- Park, C. S., Jang, W., Nah, S. P., & Kim, D. Y. (2007). Automatic modulation recognition using support vector machine in software radio applications. International Conference on Advanced Communication Technology, ICACT, 1, 9–12.

- Radiuk, P. M. (2018). Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. *Information Technology and Management Science*, 20(1).
- Roy, S., & Banerjee, P. (2005). An algorithm for trading off quantization error with hardware resources for MATLAB-based FPGA design. *IEEE Transactions on Computers*, 54(7), 886–896.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint*, arxiv:1609.04747.
- Sai Sundar, K. v., Bonta, L. R., Reddy, A. K. B., Baruah, P. K., & Sankara, S. S. (2018). Evaluating Training Time of Inception-v3 and Resnet-50,101 Models using TensorFlow across CPU and GPU. *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, 1964–1968.
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent Advances in Recurrent Neural Networks. *Arxiv:1801.01078v3*
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229.
- Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Targ, S., Almeida, D., & Enlitic, K. L. (2016). Resnet in Resnet: Generalizing Residual Architectures. *Arxiv:1603.08029*
- Thakkar, V., Tewary, S., & Chakraborty, C. (2018). Batch Normalization in Convolutional Neural Networks - A comparative study with CIFAR-10 data. *Proceedings of 5th International Conference on Emerging Applications of Information Technology, EAIT 2018*.
- Theobald, O. (2017). *Machine Learning for Absolute Beginners: A Plain English Introduction*. 163.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(October), 433–460.
- Vanhoy, G., Thurston, N., Burger, A., Breckenridge, J., & Bose, T. (2019). Hierarchical Modulation Classification Using Deep Learning. *Proceedings - IEEE Military Communications Conference MILCOM, 2019-October*, 20–25.
- Wei, W., & Mendel, J. M. (2000). Maximum-likelihood classification for digital amplitude-phase modulations. *IEEE Transactions on Communications*, 48(2), 189–193.
- Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, 98–101.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). *Dive into Deep Learning Release 0.17.0*.

- Zhang, D., Ding, W., Wang, H., & Zhang, B. (2020). Autocorrelation Convolution Networks Based on Deep Learning for Automatic Modulation Classification. Proceedings of the 15th IEEE Conference on Industrial Electronics and Applications, ICIEA 2020, 1561–1565.
- Zhao, Y., Deng, B., & Wang, Z. (2002). Analysis and study of perceptron to solve XOR problem. Proceedings - 2nd International Workshop on Autonomous Decentralized System, IWADS 2002, 168–173.
- Zhu, L., Gao, Z., & Zhu, Z. (2019). Blind Modulation Classification via Accelerated Deep Learning. 2019 IEEE 5th International Conference on Computer and Communications, ICC 2019, 2102–2107.

