

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI

YÜKSEK LİSANS TEZİ

Mustafa Bayram AYAZ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

EKİM 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI

YÜKSEK LİSANS TEZİ

**Mustafa Bayram AYAZ
(504201333)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

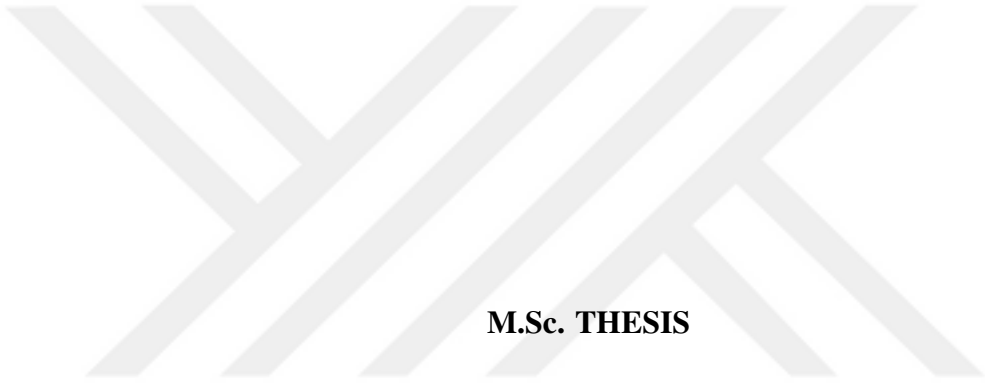
Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet Ertuğrul ÇELEBİ

EKİM 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

END TO END AUTOENCODER DESIGN FOR WIRELESS COMMUNICATION



M.Sc. THESIS

**Mustafa Bayram AYAZ
(504201333)**

Department of Electronics and Communication Engineering

Telecommunications Engineering Programme

Thesis Advisor: Prof. Dr. Mehmet Ertuğrul ÇELEBİ

OCTOBER 2023

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 504201333 numaralı Yüksek Lisans Öğrencisi Mustafa Bayram AYAZ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Mehmet Ertuğrul ÇELEBİ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ali Emre PUSANE**
Boğaziçi Üniversitesi

Prof. Dr. Hakan Ali ÇIRPAN
İstanbul Teknik Üniversitesi

Teslim Tarihi : **5 Ekim 2023**
Savunma Tarihi : **3 Ekim 2023**





Çok değerli aileme ve arkadaşlarıma...



ÖNSÖZ

Bu çalışmanın gerçekleştirilmesinde, yüksek lisans eğitimim boyunca değerli bilgilerini paylaşan ve çalışmamda motivasyon sağlayan Prof. Dr. Mehmet Ertuğrul Çelebi'ye teşekkür eder, saygılarımı sunarım. Ayrıca beni bu günlere sevgi ve saygı kelimelerinin anlamlarını bilecek şekilde yetiştirerek getiren ve benden hiçbir zaman desteğini esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

Ekim 2023

Mustafa Bayram AYAZ
(Elektronik ve Haberleşme Mühendisi)



İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
SEMBOLLER	xv
ÇİZELGE LİSTESİ	xvii
ŞEKİL LİSTESİ	xix
ÖZET	xxi
SUMMARY	xxiii
1. GİRİŞ	1
1.1 Tezin Kapsamı, Amacı ve Literatür Taraması	1
2. DERİN ÖĞRENME	9
2.1 Denetimli ve Denetimsiz Öğrenme	10
2.2 Yapay Sinir Ağlarının Temel Yapısı ve Bileşenleri	11
2.2.1 Derin öğrenmede kullanılan kavramlar	13
2.2.2 Derin öğrenmede kullanılan katmanlar, algoritmalar ve aktivasyon fonksiyonları	16
2.2.3 Optimizasyon algoritmaları	23
2.2.4 Geri yayılım (back propagation)	26
2.2.5 Kayıp fonksiyonları	27
2.3 Otokodlayıcılar	29
2.3.1 Otokodlayıcı mimarisi	30
2.3.2 Otokodlayıcı türleri	34
2.3.2.1 Evrişimli otokodlayıcılar (convolutional autoencoder, CAE)	34
2.3.2.2 Değişken otokodlayıcı (variational autoencoder, VAE)	35
2.3.2.3 Gürültü çözen otokodlayıcı (denoising autoencoder, DAE)	36
3. KABLOSUZ HABERLEŞME İÇİN OTOKODLAYICILAR	39
3.1 Alamouti Kodlaması ve Gönderici Çeşitlemesi	40
3.2 Kodlanmış ve Kodlanmamış Sistemler İçin Otokodlayıcılar	43
3.2.1 Uçtan uca haberleşme için otokodlayıcı uygulamaları	47
3.2.1.1 QPSK - Hamming(7,4) ile otokodlayıcı(7,4) performans karşılaştırması	48
3.2.1.2 64-QAM ve otokodlayıcı performans karşılaştırması	51
4. KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI	53
4.1 Uzaysal Çoğullama İçin Otokodlayıcı Modeli	54
4.1.1 Uzaysal çoğullama simülasyon sonuçları	56
4.2 Uzaysal Çeşitlilik İçin Otokodlayıcı Modeli	58
4.2.1 Sistem optimizasyonu	60
4.2.2 Verici çeşitliliği simülasyon sonuçları	61
5. SONUÇLAR	65
KAYNAKLAR	69
ÖZGEÇMİŞ	73



KISALTMALAR

AWGN	: Additive White Gaussian Noise (Toplamsal Beyaz Gauss Gürültüsü)
QPSK	: Quadrature Phase Shift Keying
8PSK	: 8 Phase Shift Keying
BLER	: Block Error Rate (Blok Hata Oranı)
ADAM	: Uyarlanabilir Moment Tahmini (Adaptive Moment Estimation)
RMSprop	: Karelerin Ortalamasının Karekökü Yayılımı (Root Mean Square Propagation)
CAE	: Evrişimli Otokodlayıcılar (Convolutional Autoencoder))
VAE	: Değişken Otokodlayıcı (Variational Autoencoder)
YZ	: Yapay Zeka
PCA	: Principal Component Analysis (Temel Bileşenler Analizi)



SEMBOLLER

M	: Sembol Kümesi Boyutu
k	: Sembol Başına Düşen Bit Sayısı
n	: Kullanılan Kanal Sayısı





ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 Derin sinir ağlarında hangi aktivasyon fonksiyonunun ne zaman kullanılacağına dair öneri	22
Çizelge 3.1 İki gönderici çeşitlemesi (transmit diversity) için sembol iletim sırası	40
Çizelge 3.2 Literatürde dikkate alınan farklı türde otokodlayıcılar	46
Çizelge 4.1 2 anten için sembol iletim sırası.....	62
Çizelge 4.2 2x1 otokodlayıcı modeli katmanları ve çıkış boyutları.....	63





ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 : Yapay zekâ ve alt kavramların kapsamı	10
Şekil 2.2 : Algılayıcı (perceptron) sinyal-akış grafiği.	12
Şekil 2.3 : ReLU fonksiyonu.....	22
Şekil 2.4 : tanh fonksiyonu.....	22
Şekil 2.5 : sigmoid fonksiyonu.....	22
Şekil 2.6 : Veri kümesinde minimuma yakınsamanın rastgele gradyan iniş ve gradyan iniş gösterimi	24
Şekil 2.7 : Otokodlayıcı genel yapısı	30
Şekil 2.8 : Tamamlanmamış Otokodlayıcı	33
Şekil 2.9 : Tam Otokodlayıcı.....	34
Şekil 2.10 : Gizli uzay gaussian dağılmaya zorlanır (bottleneck).....	36
Şekil 3.1 : Verici ve Alıcı Çeşitlilik Karşılaştırması	43
Şekil 3.2 : Otokodlayıcı olarak modellenmiş bir haberleşme sistemi	44
Şekil 3.3 : Otokodlayıcı modeli için kullanılan katman konfigürasyonu	49
Şekil 3.4 : Kodlanmamış QPSK ve Otokodlayıcı Blok Hata Oranı	50
Şekil 3.5 : Hamming(7,4) ve otokodlayıcı blok hata oranı karşılaştırması	50
Şekil 3.6 : Hem verici hem de alıcı sinir ağı ile gizli katmanlı otokodlayıcı yapısı	51
Şekil 3.7 : Geleneksel 64-QAM ve otokodlayıcı tabanlı sistem.....	52
Şekil 4.1 : N_t verici N_r alıcı MIMO sistem gösterimi	54
Şekil 4.2 : Uzaysal Çoğullama için Otokodlayıcı modeli	55
Şekil 4.3 : Uzaysal Çoğullama 2x1 ve Otokodlayıcı 2x1 BER performansı.....	57
Şekil 4.4 : Uzaysal Çoğullama 2x2 ve Otokodlayıcı 2x2 BER performansı.....	58
Şekil 4.5 : Uzaysal Çoğullama 3x2 ve Otokodlayıcı 2x3 BER performansı.....	58
Şekil 4.6 : Uzaysal Çeşitleme için Otokodlayıcı modeli.....	59
Şekil 4.7 : 2x1 Otokodlayıcı ve Alamouti karşılaştırması	64
Şekil 4.8 : Otokodlayıcının ürettiği takımyıldızı (constellation) grafiği	64



KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI

ÖZET

Bu çalışma, derin öğrenme ve kablosuz iletişim alanları arasındaki ilişkiyi inceleyen kapsamlı bir çalışmayı ele almaktadır. Çalışma, kablosuz haberleşme sistemlerinin verimliliğini ve güvenilirliğini artırmak amacıyla otokodlayıcıların nasıl kullanılabileceğini araştırmaktadır. Çalışmanın ana katkısı otokodlayıcılar ile tasarlanan bir haberleşme sisteminin farklı otokodlayıcı sistem konfigürasyonları ile performansını açıklamaktır.

Tezin ilk bölümü, derin öğrenme kavramlarına odaklanmaktadır. Denetimli ve denetimsiz öğrenme arasındaki farklar açıklanmakta ve ileri yönlü derin ağlar incelenmektedir. Ayrıca, derin öğrenmede kullanılan temel kavramlar ve algoritmalar ele alınmaktadır. Öğrenme parametreleri, aktivasyon fonksiyonları, optimizasyon algoritmaları ve geri yayılım (back propagation) gibi konular detaylı bir şekilde açıklanmaktadır. Kayıp fonksiyonları ve öğrenme algoritmaları da örneklerle birlikte incelenmektedir. Tasarlanan modelde sıklıkla kullanılan kategorik çapraz entropi (categorical cross entropy) kayıp fonksiyonu araştırılmıştır.

Kablosuz haberleşme temelleri ele alınarak otokodlayıcı ile ilişkisi incelenmiştir. Otokodlayıcıların mimarisi ve farklı türleri detaylı bir şekilde incelenmektedir. Evrişimli otokodlayıcılar, değişken otokodlayıcılar ve gürültü çözen otokodlayıcılar gibi farklı türlerin özellikleri vurgulanmaktadır.

Çalışmanın devamında uçtan uca otokodlayıcı tasarımı ele alınmaktadır. Sistem modelleri ve otokodlayıcı konfigürasyonları üzerinde durularak, farklı senaryolarda otokodlayıcı hata analizleri yapılmaktadır. QPSK, 8PSK gibi modülasyon türleri ile bunların otokodlayıcı yapılarıyla olan karşılaştırmalı hata analizleri yapılmıştır. Farklı işaret gürültü oranlarında (signal to noise ratio, SNR) otokodlayıcı model eğitilmiş ve hangi değerlerde otokodlayıcının yüksek hata performansı verdiği bilgileri çıkarılmıştır.

Alamouti sistemine alternatif olarak sunulan otokodlayıcı modeli tasarlanırken kanalda distorsiyon etkisi olarak Rayleigh sönümlemesi kullanılmıştır. MMSE tahmini Minimum Ortalama Karesel Hata ile elde edilir ve giriş sinyalinin kod çözücündeki temsili elde edilir. Bu çalışma sonucunda geleneksel 2 x 1 Alamouti sistemi ile bu sistemin otokodlayıcılarla tasarlanan modelinin bit hata oranı performansı açısından benzer sonuçlar verdiği görülmektedir. Uzaysal çeşitlilik için bir otokodlayıcı modeli tanıtılmış ve bu model 2x1 iletişim sistemlerinin verici ve alıcı işlemlerini tek bir uçtan uca kodlama yöntemi ile optimize etmektedir. Rayleigh sönümleme kanalı kullanılarak yapılan simülasyon sonuçları, otokodlayıcı tabanlı sistemin BER

performansını göstermektedir. Bu sonuçlar, mevcut geleneksel gönderici anten çeşitleme uygulamalarının performansı ile karşılaştırılmıştır.

Bu çalışma kapsamında derin öğrenme ile kablosuz iletişim arasındaki önemli bağlantılar vurgulanır ve kablosuz iletişim sistemlerindeki otokodlayıcıların potansiyeli incelenir. Bu tez, derin öğrenme ve kablosuz iletişim arasındaki ilişkinin daha iyi anlaşılmasını sağlar ve kablosuz iletişim sistemlerinin performansını artırmak için otokodlayıcıların kullanımını vurgular. Son bölümde araştırmanın sonuçları özetlenmektedir.



END TO END AUTOENCODER DESIGN FOR WIRELESS COMMUNICATION

SUMMARY

This study presents a comprehensive investigation of the relationship between deep learning and wireless communications. The study explores how autoencoders can be used to improve the efficiency and reliability of wireless communication systems. The main contribution of the study is to describe the performance of a communication system designed with autoencoders using different configurations of autoencoder systems. These configurations include the types of layers used and the effects of hyperparameters during model training.

The first section of the thesis focuses on deep learning concepts. It explains the differences between supervised and unsupervised learning and examines feedforward neural networks. Autoencoders, which operate based on unsupervised learning mechanisms and provide successful results in representing data, are discussed. Furthermore, fundamental concepts and algorithms used in deep learning are addressed. Topics such as learning parameters, activation functions, optimization algorithms, and backpropagation are explained in detail. Loss functions and learning algorithms are also examined with examples. The categorical cross entropy loss function, which is frequently used in the designed model, is investigated. It is an effective loss function for autoencoders to reconstruct the data as accurately as possible.

We also focused on the fundamentals of wireless communication. Topics such as Alamouti coding are explained in detail, followed by a discussion on how autoencoders can be used for wireless communication. The architecture and different types of autoencoders are extensively examined. Features of specific types of autoencoders, such as convolutional autoencoders, variational autoencoders, and denoising autoencoders, are highlighted. These types of autoencoders are designed for specific problems.

Compared to conventional communication systems, autoencoders have both advantages and disadvantages. Autoencoders can compress data by representing it with a reduced amount of information. This allows for efficient data storage and transmission. They can capture complex patterns and representations in the data, which can be beneficial for various tasks in communication systems. They also are capable of reconstructing corrupted or noisy data, making them robust in the presence of noise and channel impairments. They can help improve the reliability of communication systems. On the other hand, training autoencoders can be computationally intensive and time-consuming, especially for large-scale datasets. The optimization process

requires significant computational resources. Also, one of the disadvantages of autoencoders is that designing an effective autoencoder for a specific communication task requires careful consideration of various factors, such as network architecture, hyperparameter tuning, and training methodology. It can be challenging to optimize and fine-tune the autoencoder for optimal performance.

We explain end-to-end autoencoder design. Autoencoder design is thoroughly examined for both uncoded and coded systems. Uncoded systems refer to the traditional communication systems where the information is directly transmitted without any additional coding. In uncoded systems, the input data is encoded and modulated using a specific modulation scheme (e.g., BPSK or QPSK) and transmitted over the channel. At the receiver, the received signal is demodulated and decoded to recover the original data. The performance of uncoded systems is typically measured in terms of Block Error Rate (BLER), which represents the accuracy of data recovery. In conventional communication systems, channel coding techniques have been developed to improve the efficiency of information signal transmission, as the signal is subject to distortion during transmission through the channel. Coded systems involve the use of error-correcting codes in addition to modulation for reliable data transmission. The input data is first encoded using an error-correcting code, such as Hamming codes. The encoded data is then modulated and transmitted over the channel. At the receiver, the received signal is demodulated and decoded using appropriate decoding algorithms to recover the original data. The use of error-correcting codes helps in mitigating the effects of noise and channel impairments, improving the overall reliability of data transmission. These systems try to correct errors by adding extra bits to the information signal at the receiver.

The focus is on the design of end-to-end autoencoders specifically designed for wireless communications. A model for spatial diversity is introduced, aiming to optimize the entire communication system, including the transmitter and receiver. Autoencoder model emphasizes the importance of system optimization and presents simulation results to demonstrate the performance of the proposed autoencoder-based approach in achieving spatial diversity in wireless communication. In the autoencoder-based system, information is encoded in the encoding section of the autoencoder and reaches the channel layer where distorting channel effects are added. Here the encoder and decoder are designed for 2×1 system. The encoder output represents modulated signals and these symbols are processed in the channel. In the decoder, corrupted symbols are estimated by MMSE estimator and the closest representation to the input values is obtained in the output layer. The entire system is modeled similar to the Alamouti space time block coding architecture. There is end-to-end optimization in the system.

The encoder part of the autoencoder corresponds to the transmitter part of traditional systems, and the decoder part corresponds to the receiver part of traditional systems. While designing this autoencoder model, which is offered as an alternative to the Alamouti system, Rayleigh fading is used as a distortion effect in the channel. MMSE estimation is achieved with Minimum Mean Square Error (MMSE) and the representation of the input signal in the decoder is obtained. As a result of this study, it

is seen that the traditional 2 x 1 Alamouti system and the model of this system designed with autoencoders give similar results in terms of bit error rate performance.

The results section summarizes the findings and outcomes of the study. It emphasizes the significant connections between deep learning and wireless communication and highlights the potential of autoencoders in wireless communication systems.

This thesis provides a better understanding of the relationship between deep learning and wireless communication and emphasizes the use of autoencoders to enhance the performance of wireless communication systems.





1. GİRİŞ

Kablosuz haberleşme sistemleri günümüz dünyasında cihazların birbirleriyle iletişim kurabilmesi açısından önemli bir temel yapı taşı haline gelmiştir. Gelişen teknoloji ile birlikte birçok haberleşme sistemi daha yüksek hızlarda ve daha verimli halde çalıştırılmak üzere tasarlanmıştır. Cihazlar arasındaki haberleşmede alıcı ve verici arasındaki bozucu kanal etkileri bu sistemleri daha elverişli kullanmaya zorlamıştır. Günümüzdeki haberleşme sistemlerinde verici ve alıcı tarafında iletilen bilginin en doğru şekilde işlenebilmesi için çeşitli algoritmalar, haberleşme blokları geliştirilmiştir. İhtiyaca göre daha karmaşık veya görece daha basit sistemler geçmişten günümüze temel haberleşme bloklarında sıklıkla kullanılmış ve alternatifleri de sürekli üretilmektedir.

1.1 Tezin Kapsamı, Amacı ve Literatür Taraması

Bilgi teorisi ve modülasyon şemalarının analizi, belirli iletim hızları, bant genişlikleri ve sinyal-gürültü oranları (Signal to Noise Ratio, SNR) için ulaşılabilir bilgi yoğunluğu ve bit hata oranları konusunda sınırlar verir [1]. İki elektronik sistemin haberleşmesinde kullanılan kodlama ve kod çözme teknikleri uygulanan yöntemle bağlı olarak bazı temel zorlukları da beraberinde getirir. Kanalda bozulan bilginin yeniden kurtarılması için geliştirilen yöntemlerin hata düzeltme kabiliyeti, ne ölçüdeki bir hızda veriyi iletebilme, ne kadarlık bir verimde hatadan arındırma gibi konular alıcı ve verici arasında gerçekleşen tüm optimizasyon bloklarının maksimum verimde çalışmasını gerekli kılmıştır. Bu gereklilikten ötürü bir verici ve alıcı yardımıyla bir kaynaktan hedefe bir kanal üzerinden güvenilir mesaj iletimi, bir iletişim sisteminin temel gereksinimidir. Uygulamada, verici ve alıcı, haberleşme görevi için en uygun çözümü elde etmek amacıyla bir dizi çoklu bağımsız bloğa bölünür. Bu tür bir blok yapısı; her bloğun kendi içinde en iyi performansı verecek şekilde çalıştırılması, blok içi algoritmaların analizinin görece kolaylığı ve zorluğu, yine

kullanılan bu algoritmaların optimizasyonu gibi alt konulara bağımlılık olduğundan uçtan uca optimum bir performansa ulaştığı açık değildir [2]. Bir haberleşme sistemi tasarladığınızda burada kullanılan birbirinden bağımsız her bloğu kendi için optimize etme gerekliliği oluşur. Sistem tasarlandıktan sonra giriş bilgi sinyali ve çıkışta alınan girişe en yakın sinyal başarımlı ölçer. Bu açıdan bakıldığında zaman haberleşme sisteminin toplamdaki performansı sistemde kullanılan blokların minimum performansa sahip bloğu ile doğrudan ilişkilidir. Geleneksel haberleşme sistemlerinde bu sebeplerden dolayı uçtan uca ortak bir optimizasyon söz konusu değildir. Blokların optimizasyonu ve bu blokların birbirleriyle olan etkileşimindeki verim burada önemli olan performans ölçütüdür. Temel bir haberleşme sisteminde bahsi geçen bu bloklardan bazıları kaynak/kanal kodlama, modülasyon, kanal kestirimi (channel estimation), dengeleme (equalization) gibi bloklardır. Ayrıca alıcı tarafta senkronizasyon problemleri de giderilir ve bunun için algoritmalar mevcuttur [3]. Bu bloklarda gerçekleştirilen tüm işlemler çeşitli matematiksel yükleri de beraberinde getirir. Pratikte uygulanan sistemlerde ayrıca doğrusal olmayan güç amplifikatörleri kullanımı gibi bazı lineer olmayan durumların yakalanması söz konusudur [4]. Matematiksel açıdan da bu tür problemler; çözülmesi, işlenmesi, analizi açısından daha kompleks yapıdadırlar. Literatürdeki birçok mevcut sistem daha çok doğrusallık, durağanlık, tipik olarak Gauss yaklaşımlarına (Gaussianity) dayanır. Kanal nedeniyle oluşan bozulmalar, kusurlu güç amplifikatörler veya düşük çözünürlüklü analog dijital dönüştürücüler sebebiyle etkilenen sistemler, teoride oluşturulan modeller ile pratikte gerçekleşen sistem arasında farklılıklar oluşturur ve bu durum performansı negatif yönde etkiler. Halihazırda kullanılan sistemler ek olarak farklı ortam koşullarından kaynaklanan ekstra bozucu etkiler sebebiyle bahsedilen bloklarda ayrı ayrı yeni birer optimizasyon sorunu oluşabilir. Çözüm için, farklılaşan bozucu etkiler yukarıda bahsedilen her bir blok için (gerekirse) yeniden çözülmesi gereken bir problem haline alır. Halihazırda kullanılan kablosuz haberleşme sistemlerindeki algoritmalar dünyadaki mevcut iletişim sorunlarına çözüm olarak önerilmişse de bu uygulamalara alternatif bir bakış açısı da kaçınılmaz hale gelmiştir. Birçok verinin kullanıldığı günümüz çevrelerde veriyi işlemek ve istenilen çıkışları üretmek için veriyi manipüle etmek sıklıkla kullanılan bazı bilgisayar tabanlı yöntemleri ortaya çıkarmıştır. Herhangi

bir konudaki mevcut veriler, kendisine has özellikleri bakımından ayırt edilebilen bilgi parçacıklarıdır. Verilerin özgün özellikleri dolayısıyla sınıflandırma, kümeleme gibi bazı temel problemler farklı bakış açısıyla irdelenmeye başlanmıştır. En genel adıyla yapay zekâ (artificial intelligence, AI) verilerin manipülasyonunu, onları amaca en uygun şekilde yönetebilmeyi hedefleyen bir yapıdır. Günümüzde çok fazla alanda kullanılmaya, bakış açısı getirilmeye çalışılan yapay zekâ teknolojisi verinin olduğu birçok alanda etkin sonuçlar üretmektedir. Çünkü doğru veriye sahip olmak, karşılaşılan probleme karmaşık matematiksel işlemlere başvurmadan en optimum çözümü getirebilir. Amaç da bir probleme getirilen çözümü optimize etmektedir. Yapay zekâ temelde bilgisayarların zengin içeriğe sahip çevreyi veriler aracılığıyla anlaması ve bu kazanımla etkileşim kurmasını sağlamaktır. Tarihsel süreçte bu konu bazı alt başlıklarla irdelenmiştir. Derin öğrenme (deep learning, DL) olarak adlandırılan kavram, bu hedefe ulaşmak için etkin yaklaşımlardan biri olarak ortaya çıkmıştır. Sinir ağları olarak tanımlanmış birtakım katmanlarla, veriler sayısal eşdeğerleriyle temsil edilirler. Gelişmekte olan derin öğrenme yöntemleri konuşma işleme (speech processing) [5], nesnelerin interneti (internet of things, IoT) [6], otonom sürüş teknolojileri (autonomous driving technologies) [7] alanlarında sıklıkla kullanılmaktadır. Derin ağların daha iyi bir öğrenme becerisine sahip birçok problemde geçerli fonksiyon yaklaşımlarına sahip yöntemler olduğu literatürde belirtilmiştir [8]. Bu nedenle derin öğrenme matematiksel olarak modellenmesi zor olan karmaşık kanal koşullarına ve donanım kusurlarına rağmen haberleşme sistemlerini modellemek için kullanılabilir. Derin öğrenme tabanlı haberleşme sistemlerinde öğrenilen algoritmalar, matematiksel modeller ya da geleneksel algoritmalar yerine eğitim yoluyla uçtan uca performans için optimize edilmiş, modelde öğrenilen ağırlıklarla temsil edilir [9]. Derin öğrenme yaklaşımları, hesaplama hızı ve veri işleme kapasitesi sağlayan ayrıca paralel işleme mimarilerinin doğası gereği, büyük miktarda verinin işlenmesini sağlamayı gerektirir. Derin öğrenme sistemleri, paralelleştirilmiş işleme mimarileri aracılığıyla önemli bir hesaplama verimi üretmede büyük bir potansiyele sahiptir [9]. Derin öğrenme tabanlı haberleşme sistemleri, bağımsız blokları optimize etmek yerine uçtan uca performansı optimize etmeyi hedefler, dolayısıyla global performansa ulaşmak için geleneksel blok yapısına ihtiyaç

duymaz. Derin öğrenmenin haberleşme sistemlerinde kullanılması ile ilgili bir çalışma modülasyon tanıma uygulamalarıdır. Alınan sinyallerin farklı modülasyon bilgilerini ayırt etme yeteneğine sahip olan modülasyon tanıma, farklı haberleşme sistemleri arasındaki modülasyon tespitini kolaylaştırdığından haberleşme alanında önemli bir işlemdir. [10]'da yapılan çalışmada evrimsel sinir ağları (convolutional neural network, CNN) kullanılarak sinyal özellikleri çıkarılmış, derin öğrenme modeli oluşturulmuş ve 8 farklı modülasyon türünü sınıflandırarak probleminin çözümüne bir yaklaşım getirilmiştir. Geleneksel haberleşme sistemlerine bakıldığında kaynaktan alıcı kısma bir kanal üzerinden bilgi iletilirken bağımsız her blok performans açısından optimize edilse dahi bu, sistemde kullanılan haberleşme problemini global açıdan optimize etmek anlamına gelmez. Uçtan uca performans ölçütü birden fazla bloğun birlikte optimize edilmesi ya da tüm sistemin ortak bir şekilde optimize edilmesiyle sağlanır. Bu bağlamda bakıldığında geleneksel haberleşme sistemlerinin blok yapısına alternatif olarak uçtan uca yapılandırılmış bir optimizasyon problemi yeniden formüle edilerek yeni bir derin öğrenme tabanlı uygulama literatüre girmiştir. Bu yeni yaklaşım bir otokodlayıcı sistemi tarafından uçtan uca haberleşme sistemine uygulanmasına dayanır. Geleneksel haberleşme sistemleriyle karşılaştırılabilir bir performansa sahip olan otokodlayıcıların gelecek haberleşme sistemlerinde de kullanılması muhtemeldir. Otokodlayıcı tabanlı bir haberleşme sistemi, alıcı ve verici işlevlerini ortak eğiterek [2]'deki çalışmada tanıtılmıştır. Burada, haberleşme sistemi bir otokodlayıcı olarak yorumlanmış ve verici ve alıcı bileşenlerini tek bir işlemde ortaklaşa optimize eden uçtan uca yeniden oluşturma görevi olarak haberleşme sistemi tasarımının yeni bir yolu sunulmuştur. Bir derin öğrenme modeli kullanarak hata oranını en aza indirmek gibi istenen bir kayıp fonksiyonu için optimize edilmiş belirli bir kanal modeli için verici ve alıcı uygulamalarını öğrenmenin mümkün olduğunu göstermişlerdir. Vericiyi, kanalı ve alıcıyı otokodlayıcı olarak eğitilebilen tek bir derin sinir ağı olarak uygulamışlardır. Bu çalışma kapsamında geleneksel haberleşme sistemleri derin öğrenme uygulamalarından olan otokodlayıcı yapılarıyla gerçekleştirilmiş ve performans çıktıları benzer sonuçlar vermiştir.

Haberleşme sistemlerinde, özellikle MIMO sistemlerde, makine öğrenimi ve derin öğrenmenin kullanımı son yıllarda artmıştır. Sistem içinde belirli işlevleri yerine

getirmek için makine öğrenimi kullanılan [11] [12] çalışmaların yanısıra otokodlayıcı teknolojisi kullanarak sistemin tamamen uygulandığı [2] [13] çalışmalar mevcuttur. MIMO sistemlerinde kanal kestiriminin kullanılması [14] otokodlayıcıların kullanım alanlarından biridir. Araştırmalar, otokodlayıcı tekniklerinin OFDM sistemleri [15] ve moleküler haberleşme sistemleri [16] dahil olmak üzere çeşitli haberleşme sistemlerinin kalitesini artırabileceğini göstermektedir. Haberleşme sistemlerinde otokodlayıcı teknikleri, kapalı döngü otokodlayıcı ve açık döngü otokodlayıcı olmak üzere iki kategoriye ayrılarak incelenmiştir.

Haberleşme sistemlerinde otokodlayıcı kullanımına dair çalışmalar genellikle, Tek Giriş Tek Çıkış (SISO), MIMO Uzay Zaman Blok Kodu (MIMO-STBC) olarak belirli sistemlerle sınırlıdır. Ancak bu çalışmalar, verici anten sayısının iki olduğu durum için yalnızca AE-MIMO sisteminin performansını değerlendirmektedir. Bu nedenle, daha fazla verici anten içeren sistemlerde otokodlayıcı tekniklerinin etkinliğini değerlendirmek için daha fazla araştırmaya ihtiyaç vardır.

Alamouti sistemleri, kablosuz iletişim sistemlerindeki performansı artırmak ve hata oranını düşürmek amacıyla kullanılan önemli teknolojilerdir. Bu sistemler, uzaysal çeşitlilik (spatial diversity) olarak adlandırılan bir iletişim tekniğini kullanarak önemli avantajlar sunarlar. Uzaysal çeşitlilik birden fazla antenin kullanıldığı bir haberleşme sistemi tasarımıdır. Bu sistemlerde hem verici hem de alıcı tarafında birden fazla anten bulunabilir ve aynı frekans bandında birden fazla veri akışı iletimi gerçekleştirilir. Bu, aynı anda farklı yollar üzerinden veri iletimi sağlar. Uzaysal çeşitlilik, bu farklı yolların kullanılmasıyla gelen sinyalde gürültü ve hata oranının azaltılmasına yardımcı olur. Alamouti kodlaması özellikle kablosuz haberleşmede kullanılan bir çeşit uzaysal çeşitleme tekniğidir. Bu teknikte, iki antenli bir verici, veriyi özel bir matematiksel işlemle kodlar ve iki farklı anten üzerinden gönderir. Alamouti kodlamasında kullanılan Uzay-Zaman Blok Kodlaması (Space-Time Block Coding, STBC), kablosuz iletişim sistemlerinde kullanılan bir kodlama tekniğidir. Bu teknik, hem uzaysal hem de zamansal çeşitlilik sağlayarak iletişim sistemlerinin güvenilirliğini artırır. Verici, farklı antenler aracılığıyla aynı veriyi gönderir, ancak bu veri antenler arasında farklı bir şekilde kodlanır. Bu, aynı verinin farklı yollarla iletilmesi anlamına gelir ve bu da iletimin kanal üzerindeki çeşitli yollar boyunca farklı davranmasını sağlar. Alamouti

uzay zaman blok kodlaması aynı veriyi farklı zaman dilimlerinde gönderdiğinden zamansal çeşitlilik sağlar. İletim sembolleri belirli bir sıra ile gönderilir ve bu sıra iletim sırasında değiştirilir. Bu, kanal koşullarının zaman içinde değiştiği durumlarda iletişim kalitesini artırır. Alıcı tarafında, gelen sinyal parçaları çıkarılır ve orijinal veri yeniden elde edilir. STBC'nin önemli bir özelliği, alıcı tarafında karmaşık matematiksel işlemlere gerek duymadan veriyi geri çözebilmesidir.

Alamouti kodlaması, gelen sinyalleri karşılaştırarak hata düzeltme ve hata azaltma yetenekleri sunar. Uzaysal çeşitlilik, kablosuz iletişim sistemlerinde çoklu yol etkileri, sinyal zayıflaması ve çoklu giriş çoklu çıkış (MIMO) avantajlarından yararlanarak iletişim kalitesini artırır. Bu teknikler, daha güvenilir ve yüksek kapasiteli kablosuz iletişim sistemlerinin tasarlanmasına katkı sağlar [17].

Bu çalışmada kodlanmış ve kodlanmamış geleneksel sistemler ve otokodlayıcı tabanlı sistemler incelenmiştir. Geleneksel sistemler ile otokodlayıcı tabanlı sistemler benzer sonuçlar göstermektedir. Ayrıca çalışma Alamouti sisteminin otokodlayıcılar ile uzaysal çeşitlilik karşılaştırmasını içerir. Verilerin farklı antenler üzerinden alıcı kısma iletilmesi otokodlayıcı ile tasarlanmıştır. Otokodlayıcının kodlayıcı bölümü geleneksel sistemlerin gönderici (verici) kısmına, kod çözücü kısmı ise geleneksel sistemlerin alıcı kısmına karşılık gelir. Alamouti sistemine alternatif olarak sunulan bu otokodlayıcı modeli tasarlanırken kanalda bozucu etki olarak Rayleigh sönümlemesi kullanılmıştır. Kanal kestirimi Minimum Ortalama Kare Hatası (Minimum Mean Square Error, MMSE) ile sağlanarak kod çözücüde giriş sinyalinin temsili elde edilir. Bu çalışmanın sonucunda görülmektedir ki geleneksel 2 x 1 Alamouti sistemi ile bu sistemin otokodlayıcılar ile tasarlanmış modeli bit hata oranı performansı açısından benzer sonuçlar vermektedir.

Çalışma kapsamında uzaysal çoğullama (spatial multiplexing) konusu incelenmiş, bu türden bir sistem otokodlayıcı ile modellenmiştir. Farklı sayıda alıcı anten ile otokodlayıcı modeli eğitilmiş, geleneksel sistemler sistemler ile performans karşılaştırmaları yapılmıştır.

Bu tez kapsamında, Bölüm 2'de derin öğrenme temellerinden, yapay sinir ağlarının eğitimi esnasında kullanılan hiperparametrelerden, ağın optimizasyonu için

gerekli olan algoritmalarından ve derin öğrenmenin bir uygulaması olan otokodlayıcı yapılarından bahsedilmiştir. Bölüm 3’te geleneksel haberleşme sistemleri hakkında temel bilgilere ve bu sistemlerin otokodlayıcılar ile nasıl tasarlandıklarına, Alamouti sisteminin nasıl çalıştığına dair açıklamalara değinilmiştir. Bölüm 4’te ise Alamouti kodlamasına benzer bir uçtan uca otokodlayıcı tasarımına yer verilmiş, otokodlayıcı tabanlı sistemin Alamouti sistemine karşı uzaysal çeşitlilik açısından bit hata performans karşılaştırılmasına yer verilmiştir. Son olarak Bölüm 5’te çalışmanın sonuçları verilmiştir.





2. DERİN ÖĞRENME

Derin öğrenme, yapay sinir ağları temelli bir makine öğrenme yöntemidir ve son yıllarda büyük ilgi görmektedir. Bilgisayarın karmaşık problemleri çözme yeteneğini artırarak, veri analizi, tanıma, sınıflandırma ve tahmin gibi birçok alanda önemli başarılar elde etmektedir.

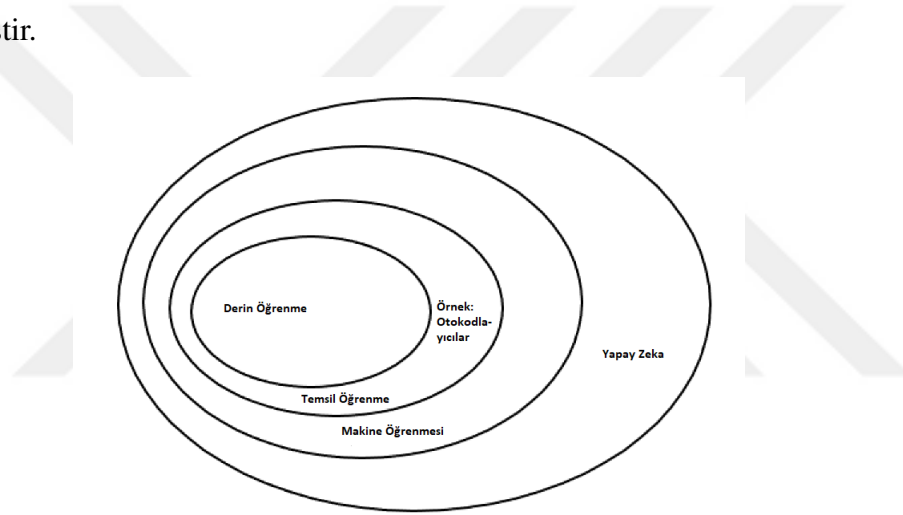
Derin öğrenme, çok katmanlı yapay sinir ağlarından oluşan bir modeldir. Bu yapay sinir ağları, beyindeki sinir hücrelerinin çalışma prensiplerinden esinlenerek tasarlanmıştır. Her bir sinir hücresi, veriyi alır, bu veriyi işler ve sonucunu diğer sinir hücrelerine ileterek bilgiyi ileriye taşır. Bu sinir hücrelerinin katmanlar halinde düzenlendiği bir yapıda, veri özellikleri hiyerarşik olarak öğrenilmektedir.

Derin öğrenme algoritması, veriye dayalı olarak ağırlıkları ve parametreleri otomatik olarak öğrenir. Büyük veri setlerinde derin öğrenme algoritması, yüksek işleme gücüyle derin ağları eğiterek karmaşık desenleri tespit edebilir ve veriye dayalı çıkarımlar yapabilir. Bu sayede, görüntü, metin, ses gibi farklı veri türleri üzerinde etkili bir şekilde çalışabilir.

Derin öğrenme, özellikle görüntü işleme, doğal dil işleme, konuşma tanıma, otonom araçlar, tıbbi teşhis ve finansal analiz gibi alanlarda büyük bir etki yaratmıştır. Örneğin, derin öğrenme yöntemleri, görüntüleri otomatik olarak sınıflandırabilir, metinleri anlayabilir ve konuşmaları tanıyabilir. Bununla birlikte, derin öğrenme yöntemlerinin karmaşıklığı ve veri yoğunluğu nedeniyle eğitim süreci zorlu olabilir ve uygun donanım kaynakları gerektirebilir.

Derin öğrenme, sürekli olarak gelişen bir alan olup araştırma ve uygulama alanında büyük bir potansiyele sahiptir. Yeni algoritmaların ve mimarilerin geliştirilmesi, daha iyi performans ve daha geniş bir uygulama yelpazesi sunmaktadır. Derin öğrenme, gelecekte birçok alanda daha da yaygınlaşacak ve günlük hayatımızı daha akıllı hale getirecektir. Veriyi doğru işleme ve algoritma bilgileri bu aşamada önemlidir. Özetle yapay zeka, teknolojik gelişmelerin ve araştırmaların odak noktasındadır. İnsan

benzeri zeka ve yetenekleri makinelerle birleştirme vizyonu, yapay zeka alanının gelecekte daha da önem kazanacağını göstermektedir. Bu alandaki ilerlemeler, günlük yaşamımızı dönüştürebilir, yeni fırsatlar yaratabilir ve insanlığın karşılaştığı zorluklara çözümler sunabilir. Yapay zekâ gerek konusu gerek uygulama alanı dolayısıyla geniş bir kavramdır. Yapay zekâ, makine öğrenimi ve derin öğrenme, akıllı davranan yazılımları tanımlamak için sıklıkla birbirinin yerine kullanılan üç terimdir. Bununla birlikte, aralarındaki temel ayrımları anlamak gerekir. Derin öğrenme, makine öğreniminin bir alt kümesidir ve makine öğrenimi, akıllıca bir şey yapan herhangi bir bilgisayar programı için bir kilit terim olan yapay zekanın bir alt kümesidir. Başka bir deyişle, tüm makine öğrenimi bir yapay zekâ uygulamasıdır, ancak tüm yapay zekâ uygulamaları bir makine öğrenmesi değildir. Şekil 2.1’de bu durum görsel olarak gösterilmiştir.



Şekil 2.1 : Yapay zekâ ve alt kavramların kapsamı

2.1 Denetimli ve Denetimsiz Öğrenme

Denetimli ve denetimsiz öğrenme, makine öğrenimi alanında yaygın olarak kullanılan iki önemli kavramdır. Bu kavramlar, algoritmalara temel bir yaklaşım ve veri türü sağlar.

Denetimli öğrenme, etiketli verilerin kullanıldığı bir öğrenme yaklaşımıdır. Bu yöntemde, eğitim veri seti, her bir örneğin doğru etiket veya çıktı değeri ile eşleştirildiği bir şekilde hazırlanır. Öğrenme algoritması, bu etiketli veri setini kullanarak girişlerle çıktılar arasındaki ilişkiyi modellemeyi ve yeni verileri doğru bir şekilde sınıflandırmayı veya tahmin etmeyi öğrenir. Örneğin, bir görüntünün

üzerindeki nesneleri tanımlamak veya bir e-postanın spam olup olmadığını belirlemek gibi görevler, denetimli öğrenme yöntemiyle ele alınabilir.

Denetimsiz öğrenme ise etiketsiz verilerin kullanıldığı bir öğrenme yaklaşımıdır. Bu yöntemde, eğitim veri seti sadece giriş verilerini içerir ve herhangi bir etiket veya çıktı bilgisi bulunmaz. Denetimsiz öğrenme algoritmaları, bu veri setindeki yapıyı ve desenleri keşfetmek için giriş verilerini analiz eder. Örneğin, veri gruplarını veya benzerlikleri bulmak, veri boyutunu azaltmak veya veriye bir temsil oluşturmak gibi görevler, denetimsiz öğrenme yöntemiyle gerçekleştirilebilir.

Denetimsiz öğrenme, denetimli öğrenmeden farklı, geniş makine öğrenimi yöntemi sınıfıdır. Denetimli öğrenmede amaç, girdi verilen bir etiketi (veya etiketler üzerinde bir dağılımı) öngören bir işlevi f , yani $\hat{y} = f(x)$ öğrenmek için giriş etiketi çiftlerini (x, y) kullanmaktır. Denetimsiz öğrenmede, böyle bir etiket veya başka bir hedef sağlanmaz. Veriler bir dizi örnek x 'ten oluşur ve amaç, x 'in kendisinin istatistiksel yapısı hakkında bilgi edinmektir. Denetimsiz öğrenmede modelin denetlenmesine ihtiyaç yoktur. Model gerekli sonuçları elde etmek için kendisi çalışır. Sonuçlara bağlı olarak öğretilmesi yerine, verilerden öğrenme işlevi gerçekleştirilir. Denetimsiz öğrenmede kümeleme algoritmaları bir örnek olarak verilebilir. Kümeleme algoritmaları istatistiksel modelleme çalışmalarının çoğunda kullanılabilir. Burada ağ yapısı direkt olarak gözlenen yapıdan ayarlanmaz. Ancak, eğer biliniyorsa, onu daha iyi açıklamamıza veya ondan tahminlerde bulunmamıza yardımcı olur. Kümeleme, verilerin / örneklerin doğal olarak oluşan gruplandırmasının belirlenmesidir (grup içi değişkenliği düşük ve grup değişkenliği arasında yüksek). Bu yöntemde işaretlenmemiş (unlabelled) veri üzerinden bilinmeyen bir yapıyı tahmin etmek için bir fonksiyon kullanılır. Burada girdi verisinin hangi sınıfa ait olduğu belirsizdir.

2.2 Yapay Sinir Ağlarının Temel Yapısı ve Bileşenleri

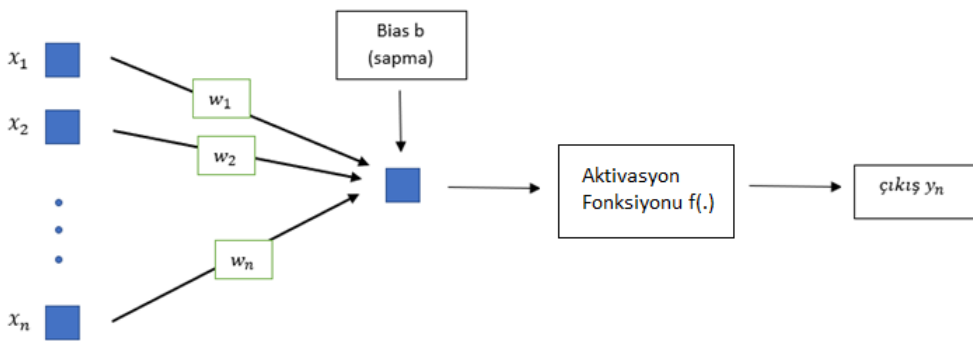
Algılayıcı, doğrusal olarak ayrılabilir olduğu söylenen örüntülerin sınıflandırılması için kullanılan bir sinir ağının en basit şeklidir.

Algılayıcı, yapay sinir ağlarının temel yapı taşlarından biridir. Basit bir yapay sinir hücresi olarak düşünülebilir. Bir algılayıcı, girdi verilerini ağırlıklarla çarparak ağırlıklı toplamalarını hesaplar ve ardından bir aktivasyon fonksiyonu kullanarak bir çıktı üretir. Bu çıktı, genellikle bir eşik değeriyle karşılaştırılarak bir sınıflandırma veya karar verme işlemi yapılır. Algılayıcı, girdi verilerini öğrenerek ve ağırlıklarını ayarlayarak belirli bir görevi başarmak için kullanılabilir.

İleri yönlü derin ağlar ise, birbirine bağlı algılayıcılardan oluşan çok katmanlı yapay sinir ağlarıdır. İleri yönlü derin ağlar bir girdi katmanı, bir veya daha fazla gizli katman ve bir çıktı katmanından oluşur. Her katmandaki algılayıcılar, bir önceki katmandaki çıktıları alır ve kendi ağırlıklarıyla çarparak yeni bir çıktı üretir. İleri yönlü derin ağlar, derin öğrenme için kullanılan en yaygın yapay sinir ağı modellerinden biridir.

Bu ağlar, birçok avantajı nedeniyle tercih edilen bir modeldir. Derin ağlar, yüksek seviyede karmaşıklığı ele alabilme yeteneğine sahiptir. Çok katmanlı yapıları sayesinde, veri özelliklerini öğrenebilirler. Ayrıca, giriş verilerindeki örüntüleri otomatik olarak keşfedebilir ve genellemeler yapabilirler. Öğrenme sürecinde geri yayılım algoritması kullanılarak ağırlıklar ayarlanır ve hedeflenen çıktılara yakın sonuçlar elde edilir.

Derin ağlar çeşitli uygulama alanlarında başarıyla kullanılmaktadır. Görüntü ve ses tanıma, doğal dil işleme, otomatik sürüş, nesne tanıma, duygu analizi gibi birçok alanda kullanılarak ileri düzeyde performans elde edilebilir [18].



Şekil 2.2 : Algılayıcı (perceptron) sinyal-akış grafiği.

Sonuç olarak, algılayıcılar temel sinir hücreleri olarak işlev görürken, ileri yönlü derin ağlar birçok algılayıcının bir araya gelmesiyle oluşan çok katmanlı yapay sinir ağı modelleridir. Karmaşık problemleri çözmek ve yüksek seviyede özellikleri öğrenmek

için kullanılır ve birçok uygulama alanında büyük başarı sağlar. Algılayıcının matematiksel bir ifadesi:

$$v = f\left(\sum_{i=0}^n w_i x_i + b\right) = f(\mathbf{w} \cdot \mathbf{x}) \quad (2.1)$$

şeklindedir. Burada $\mathbf{w}$ gerçel değerli ağırlıkların bir vektörü, n girişin özellik sayısı ve b ise sapma terimidir. $x_1, x_2, \dots, x_n$ algılayıcıya uygulanan girdilerdir. Şekil 2.2’de algılayıcının temel mimarisi gösterilmektedir. Algılayıcı modelini kullanarak doğru bir sınıflandırma yapmak için bir eşik değeri gerekmektedir. Aktivasyon fonksiyonu, bu eşik değerinin üzerindeki bir sınıf olarak etiketlerken, altındakileri diğer bir sınıf olarak etiketler. Algılayıcı genellikle veriyi iki kısma ayırmaya olanak tanır, bu nedenle lineer ikili bir sınıflandırıcı olarak da adlandırılır. Algılayıcı öğrenme algoritmasının amacı, pozitif ve negatif girişleri doğru bir şekilde sınıflandırabilen bir karar sınırı oluşturmaktır. Hedef, en iyi sınıflandırmayı sağlayan ağırlık ve sapma parametrelerini öğrenmektir.

2.2.1 Derin öğrenmede kullanılan kavramlar

Derin öğrenme uygulamalarında araştırmacılar çıktı üretmek istedikleri problemi çözebilmek için modelledikleri problemi temsil edecek en iyi özelliklerin çıkarılması üzerinde çalışmalar yapmışlardır. Bu özellikler arasında problemi temsil yeteneği en yüksek olanların seçilmesi doğru çıktıları üretmek açısından önem arz etmektedir. Dünya üzerindeki herhangi bir problemi derin öğrenme algoritmaları yardımıyla çözmek için birtakım makine öğrenmesi kavramlarının en uygun şekilde seçilmesi gerekmektedir. Bunun için algoritma ve parametre seçimlerinin gelişimi ile birlikte çok katmanlı yapay sinir ağının nasıl tasarlanacağı, ağırlık değerlerinin ne olacağı, ağın kaç katmandan oluşacağı, katmanların kaç nöron içereceği, veriler katmanlar arasında işlenirken hangi optimizasyon algoritmalarının çalışması gerektiği, en uygun aktivasyon fonksiyonunun seçilmesi gibi konular zaman içerisinde gelişim göstermiştir. Gerekli verilerin ağı beslemesi esnasında büyük öneme sahip olan hiperparametreler, eğitim esnasında özellikle ağırlıklar üzerinde güçlü etkisi vardır. Özellikle zaman içerisinde donanımsal gelişimler dolayısıyla hızlı çalışan ve daha

fazla katmandan oluşan karmaşık yapılar tasarlanabildiğinden hiperparametre seçimi ve optimizasyonu üzerinde daha fazla çalışma gerektiren bir konu haline gelmiştir.

Derin öğrenmede kullanılan modelin yüksek başarımlı sağladığı birbirinden farklı hiperparametre grupları olabilmektedir. Bu bölümde bir problemin çözümüne yönelik tasarlanan modelde sıklıkla kullanılan bazı hiperparametreler açıklanacaktır.

Veri Seti Boyutu

Derin öğrenme uygulamalarında veri setinin büyüklüğü ve çeşitliliği öğrenme için en önemli faktörlerdendir. Veri setinin büyüklüğü iyi bir model için yeterli değildir, bunun dışında verinin çeşitliliği de önemlidir. Verinin çeşitliliği arttıkça oranda modelin başarımlı da artış gösterecektir.

Yığın Boyutu (Batch Size)

Yığın boyutu, bir eğitim veri setinin, bir seferde ağa verilen örneklerin sayısıdır. Derin öğrenme modelinin eğitimi sırasında, genellikle veri seti büyük olduğu için tüm veri setini aynı anda ağa beslemek mümkün olmaz. Bu nedenle, veri seti küçük gruplara bölünerek her bir grup/yığın (batch) ayrı ayrı ağa beslenir.

Yığın boyutu, eğitim sürecindeki birkaç faktörü etkiler. Öncelikle, bellek gereksinimlerini etkiler. Büyük bir yığın boyutu kullanıldığında, daha fazla bellek kullanılır. Ayrıca, işlemci ve Grafik İşlemci Birimi (Graphics Processing Unit, GPU) kullanımını da etkiler. Büyük bir yığın boyutu, işlemci ve GPU'nun daha yoğun bir şekilde çalışmasına neden olabilir.

Yığın boyutu ayrıca gradyan hesaplama ve güncelleme işlemlerini etkiler. Daha büyük bir yığın boyutu, daha hassas bir gradyan tahmini sağlar, ancak hesaplama maliyetini artırır. Öte yandan, daha küçük bir yığın boyutu, gradyan tahmininin daha az hassas olmasına neden olabilir, ancak daha hızlı hesaplamalar yapılmasına olanak sağlar.

Yığın boyutu seçimi, veri setinin boyutuna, kullanılan donanım kaynaklarına ve modelin karmaşıklığına bağlı olarak yapılmalıdır. Genel olarak, büyük yığın boyutlar daha hızlı hesaplamalara ve daha iyi gradyan tahminlerine yol açabilir, ancak daha fazla bellek ve işlemci gücü gerektirirken, küçük yığın boyutlar daha hızlı eğitim süreleri ve daha az bellek kullanımı sağlar.

Öğrenme Oranı (Learning Rate)

Derin öğrenme uygulamalarında nöron parametrelerinin güncellenmesi geriye doğru türev alınarak (geri yayılım, back-propagation) ve bu türevin öğrenme oranı dediğimiz parametre ile çarpılmasıyla çalışır. Çoğu durumda, öğrenme oranı model eğitimi sırasında manuel olarak ayarlanmalıdır ve bu ayarlama genellikle yüksek doğruluk elde edebilmek için gerekmektedir. Ağın eğitimi esnasında kullanılan “öğrenme oranı” parametresi sabit değer olarak belirlenebilir ya da öğrenme esnasında kademeli değiştirilebilir. Optimizasyon algoritmalarının doğru minimuma yakınsaması için ne kadar hızlı ilerlemesi gerektiği veya ne kadar doğrulukta ilerlemesi gerektiğini ifade eder. Öğrenme oranının küçük olması yayılım esnasında küçük adımlarla ilerleyeceğinden öğrenme işleminin uzun sürmesine neden olacaktır. Öğrenme oranı değerinin başlangıçta küçük olması, lokal optimum değere takılarak, global optimum değere hiç ulaşamamasına neden olabilir. Buna karşı yüksek olması da yakınsaması gereken değeri atlamaya ve istenmeyen değerlere yakınsamasına sebep olabilir.

Eğitim Tur Sayısı (Epoch)

Modelin ne kadarlık eğitimden geçeceği önemli bir sorundur. Eğer verilerin üzerinden az geçerse öğrenme olmaz fazla geçerse aşırı öğrenme gerçekleşir. Bunun dengede tutulması gerekir. Genellikle eğitim tur sayısı arttıkça başarı artar ama belirli değerden sonra öğrenme azalır. Model eğitilirken verilerin tümü aynı anda eğitim için giriş katmanını beslememelidir. Belli sayıda parçalar halinde eğitimde yer alırlar. İlk parça eğitilir, modelin başarımı test edilir, başarıma göre geri yayılım ile ağırlıklar güncellenir. Daha sonra yeni eğitim kümesi ile model tekrar eğitilip ağırlıklar tekrar güncellenir. Bu işlem her bir eğitim adımında tekrarlanarak model için en uygun ağırlık değerleri hesaplanmaya çalışılır. Bu eğitim adımlarının her birine “eğitim tur sayısı” denilmektedir. Eğitim tur sayısının artırılması daha iyi verim sağlamayabilir ancak tam tersine eğitim verisini ezberlemeye başlamasına neden olabilir ve yeni bir veride sonuçları istenilen performansta olmayabilir.

2.2.2 Derin öğrenmede kullanılan katmanlar, algoritmalar ve aktivasyon fonksiyonları

Derin öğrenme, yapay sinir ağlarının çok katmanlı yapılar kullanarak karmaşık problemleri çözmek için kullanıldığı bir makine öğrenme yöntemidir. Derin öğrenme ağları, birbirine bağlı olan birçok katmandan oluşur. En yaygın kullanılan derin öğrenme katmanlarından giriş katmanı, verilerin ağa girdi olarak verildiği ilk katmandır. Genellikle verilerin boyutunu uygun hale getirmek için ön işleme adımları burada gerçekleştirilir. Gizli katmanlar giriş katmanından sonra gelen ve çıktı katmanına kadar olan katmanlardır. Bu katmanlar arasında çeşitli sayıda bulunabilir ve her bir katman, girişleri alır, ağırlıkları ve aktivasyon fonksiyonu tarafından belirlenen işlemleri gerçekleştirir. Çıktı Katmanı, en son katmandır ve genellikle sonuçları veren çıktıları üretir. Örneğin, sınıflandırma problemlerinde, çıktı katmanı genellikle sınıf etiketlerini tahmin eder. Aktivasyon fonksiyonları, her bir katmandaki çıktıları belirli değerler arasına temsil edilmesini sağlayan bir bileşendir.

Yoğun Katman (Dense Layer):

Yoğun katman (Dense Layer), yapay sinir ağlarında sıklıkla kullanılan bir tür katmandır. Bu katman, tam bağlantılı bir yapıya sahiptir, yani her bir girdi birimi, her bir çıktı birimiyle bağlantılıdır. Ağırlıklar, her bir girdi birimine atanmış olan ağırlık değerlerini içeren bir matristir. Girdi, katmana verilen girdi değerlerini temsil eder. Sapma ise her bir çıktı birimine atanmış olan sapma değerlerini içeren bir vektördür. Aktivasyon fonksiyonu çıktı değerlerini sınırlamak veya lineer olmayan bir şekilde dönüştürmek için kullanılan bir fonksiyondur.

Yoğun bir katman, girişlerinin önceki katmandaki J nöronun çıktıları olduğu K nöronlardan oluşur. Sırasıyla $\mathbf{x} \in \mathbb{R}^J$ ve $\mathbf{y} \in \mathbb{R}^K$ olarak ifade edilirse, boyutu J ve K olan giriş ve çıkış vektörleriyle bir yoğun katman aşağıdaki şekilde ifade edilebilir:

$$\mathbf{y} = f(\mathbf{x}) = \varphi(\Theta\mathbf{x} + \mathbf{b}) \quad (2.2)$$

Burada $\mathbf{b} \in \mathbb{R}^K$ ve $\Theta \in \mathbb{R}^{K \times J}$ sırasıyla eğitilebilir sapma vektörü ve eğitilebilir ağırlık matrisini temsil eder ve $\varphi(\cdot)$ aktivasyon fonksiyonudur. Her bir çıkış $y_k = \varphi([\Theta]_k^\top \mathbf{x} + b_k)$ ise katmandaki k . nöron tarafından gerçekleştirilen işlemi temsil eder.

Uygun parametre seçimi ile çok katmanlı sinir ağları prensipte ideal fonksiyonlara yaklaşabilir, daha fazla gizli birimler (hidden units) daha iyi yaklaşımlar elde etmeyi sağlar.

Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN)

Yoğun katmanlar, bir boyutlu verileri işlemek için faydalı olsa da, genellikle iki veya üç boyutlu girişler konvolüsyonel katmanlar tarafından işlenir. Tam bağlı katmanlardan farklı olarak, Evrişimsel sinir ağlarında her nöron yalnızca aynı bölgeden gelen küçük bir giriş alt kümesine bağlıdır. Ağırlıklar, giriş özellikleriyle "konvolüsyon" yapılarak (çapraz korelasyon) çıktı üretmek için kullanılır. Evrişimsel sinir ağları, giriş verilerinde ilişkileri/yapıyı bulma görevlerinde özellikle başarılıdır ve bu nedenle görüntü veya dil temelli görevler için uygundur. Geleneksel tam bağlı katmanlara kıyasla, evrişimsel sinir ağlarının paylaşılan ağırlık mimarisi nedeniyle çok daha az ağırlığı vardır. Az sayıda ağırlık, geri yayılım sırasında hesaplanması gereken türev sayısını azaltır ve bu nedenle eğitim ve uygulama sürecinde önemli ölçüde daha hızlıdır.

İleri beslemeli sinir ağlarında tek boyutlu giriş alındığından, çok boyutlu verilerin tek boyutlu bir vektöre dönüştürülmesi gerekmektedir. Bu dönüşümle birlikte, verinin çok boyutlu özellikleri evrişimsel sinir ağları kullanılarak korunurken, diğer yöntemlerde kaybolabilir. Evrişimsel sinir ağları ile ayrıca uçtan uca tasarım da gerçekleştirilebilir [19] ancak bu tez kapsamının dışındadır.

Yığın Normalizasyon Katmanı (Batch Normalization Layer)

Eğitim sırasında, her katmana karşılık gelen çıktıların dağılımı, optimize edilebilir parametrelerin iyileştirilmesiyle birlikte değişir. Bu nedenle, her gizli katman, giriş dağılımındaki değişikliklere sürekli olarak uyum sağlamak için parametrelerini sürekli olarak yeniden ayarlamalıdır. Bu sorun derin sinir ağlarında daha da artar, çünkü ilk katmanın çıktı dağılımındaki bir değişiklik, son katmanların giriş dağılımı üzerinde önemli bir etkiye sahip olabilir. Bu sorunu çözmek için, yığın normalizasyon

katmanları girişlerini normalize eder, böylece ilgili dağılımlar optimize edilmiş ortalamalara ve varyanslara sahip olur. B_S yığın boyutu olmak üzere, bir yığın normalizasyon katmanının giriş vektörleri $\mathbf{x}^{[s]} \in \mathbb{R}^J, s \in \{1, \dots, B_S\}$ olarak ifade edilirse, her elemanın x_j ortalaması ve varyansı, yığın üzerinde tahmin edilebilir:

$$\mu_j = \frac{1}{B_S} \sum_{s=1}^{B_S} x_j^{[s]}, \text{ ve } \sigma_j^2 = \frac{1}{B_S} \sum_{s=1}^{B_S} \left(x_j^{[s]} - \mu_j \right)^2 \quad (2.3)$$

Her boyuta daha sonra sıfır ortalama ve birim varyansa sahip olacak şekilde ayrı ayrı normalizasyon işlevi uygulanır:

$$\hat{x}_j^{[s]} = \frac{x_j^{[s]} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}} \quad (2.4)$$

burada ϵ , sayısal kararlılığı sağlamak için eklenen küçük bir sabittir. Her j boyutunun ortalamaları ve varyansları daha sonra tipik olarak sırasıyla eğitilebilir parametreler γ_j ve β_j tarafından kontrol edilir:

$$y_j^{[s]} = \gamma_j \hat{x}_j^{[s]} + \beta_j \quad (2.5)$$

Girdi dağılımının bu yeniden parametrelendirilmesi, daha hızlı eğitim sağlar ve yapay sinir ağının [18] hem performansını hem de genelleme özelliklerini geliştirir.

Doğrusal Regresyon

Derin öğrenmede doğrusal regresyon, girdi ve çıktı değişkenleri arasındaki doğrusal ilişkileri yaklaşık olarak tahmin etmek için derin öğrenme modellerinde kullanılır. Derin öğrenme modelleri genellikle karmaşık doğrusal olmayan sorunları çözmek için kullanılır ancak doğrusal regresyon görevleri için de kullanılabilirler. Bu ilişki, basit bir doğrusal denklemle ifade edilir: $y = mx + b$. Burada, y çıktı, x girdi, m eğim, ve b sapma (bias) olarak adlandırılır. Derin öğrenmede doğrusal regresyonu uygulamanın bir yolu, doğrusal aktivasyon işlevine sahip tek katmanlı bir sinir ağı kullanmaktır. Bu, basit bir doğrusal regresyon modeline eşdeğerdir, ancak gradyan iniş ve geri yayılım gibi derin öğrenme teknikleri kullanılarak eğitilebilmesi avantajına sahiptir. Doğrusal regresyon, derin öğrenme bağlamında genellikle kayıp fonksiyonu olarak ortalama

karesele hata (Mean Squared Error - MSE) veya ortalama mutlak hata (Mean Absolute Error - MAE) gibi regresyon problemleri için kullanılan temel kayıp fonksiyonları ile çalışır. Bu kayıp fonksiyonları, tahmin edilen değerlerle gerçek değerler arasındaki farkları ölçer ve bu farkın minimize edilmesini amaçlar. MSE, en yaygın kullanılan kayıp fonksiyonlarından biridir ve aşağıdaki formülle ifade edilir:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.6)$$

Burada, N veri örneği sayısını, y_i gerçek değeri, ve $\hat{y}_i$ tahmini değeri temsil eder. MSE, tahmin edilen değerler ile gerçek değerler arasındaki farkların karelerinin ortalamasını alır.

Lojistik Regresyon (Logistic Regression)

Derin öğrenmede lojistik regresyon, sınıflandırma problemlerini çözmek için derin öğrenme modellerinin kullanılmasını ifade eder. Lojistik regresyon, bir örneğin belirli bir sınıfa ait olma olasılığını tahmin etmek için kullanılabilen denetimli bir öğrenme algoritmasıdır. Derin öğrenmede lojistik regresyon sigmoid aktivasyon fonksiyonuna sahip tek katmanlı sinir ağı ile kullanılabilir. Lojistik regresyonun genel bir ifadesi [20],

$$P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (2.7)$$

şeklindeir. Burada:

- $P(Y = 1 | X)$, bağımlı değişkenin (etiketin) 1 olma olasılığını temsil eder.
- $X = (X_1, X_2, \dots, X_n)$, bağımsız değişkenlerin (özelliklerin) vektörüdür.
- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$, modelin öğrenmesi gereken katsayılarıdır.

Lojistik regresyon, derin öğrenmenin daha karmaşık ve çok katmanlı modellerine kıyasla daha basit bir yapısı vardır. Ancak birçok uygulamada başlangıç noktası olarak kullanılabilir bir sınıflandırma algoritmasıdır. Derin öğrenme, lojistik regresyonu temel alarak daha karmaşık ve katmanlı sinir ağları geliştirerek başlar ve daha geniş ve karmaşık problemleri ele alabilir.

Aktivasyon Fonksiyonları

Aktivasyon fonksiyonu, bir sinir ağındaki bir nöronun çıktısını belirleyen matematiksel işlem veya dönüşüm işlevidir. Bu fonksiyon, nöronun gelen sinyali ne kadar "aktif" ettiğini veya hangi sınıfa ait olduğunu belirler. Literatürde yaygın olarak bulunan sinir ağı aktivasyon fonksiyonları [21]:

- **Doğrultucu lineer birim (Rectified linear unit, ReLU)**

Doğrultucu lineer birim (ReLU), derin öğrenme ve sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. ReLU işlevi, giriş değeri negatif ise sıfırı, pozitif ise giriş değerini çıkış olarak verir. ReLU işlevi, hesaplama açısından oldukça basit ve hızlıdır. Sadece bir eşik değeri kontrol edilir ve geri kalan işlem düşük maliyetlidir [22].

ReLU işlevi, pozitif değerlerde giriş değerini doğrudan aktarırken, negatif değerlerde sıfır döndürür. Bu, gradyanın düzgün ve etkin bir şekilde ilerlemesini sağlar, özellikle de giriş değerinin büyük bir kısmının negatif olduğu durumlarda. Sigmoid veya tanh gibi diğer aktivasyon fonksiyonlarına kıyasla, ReLU daha az kaybolan gradyan sorunu yaşar. Negatif değerlerde sıfır döndüren ReLU, gradyanın kaybolma riskini azaltır. Giriş değeri yaklaşık olarak sıfıra yaklaştığında gradyanın yavaşça azaldığı bir geçiş bölgesine sahiptir. Bu, gradyanın düzgün bir şekilde güncellenmesini sağlar ve eğitim sürecinde daha istikrarlı bir ilerleme sağlar.

$$\text{ReLU}(x) = \max(0, x) \quad (2.8)$$

- **Hiperbolik Tanjant (Hyperbolic tangent)**

Hiperbolik Tanjant, tanjant fonksiyonunun hiperbolik versiyonudur. Hiperbolik tanjant fonksiyonunun ortalama değeri sıfırdır, yani giriş değeri sıfır olduğunda çıkış değeri de sıfırdır. Bu özelliği, bazı durumlarda eğitim sürecini optimize etmek için faydalı olabilir. Hiperbolik tanjant fonksiyonunun çıkış değeri -1 ile 1 arasında sınırlıdır. Bu, çıktının belirli bir aralıkta kalmasını sağlar ve eğitim sürecindeki gradyanı düşük tutar. Hiperbolik tanjant fonksiyonunun bir dezavantajı, hesaplama maliyetinin sigmoid fonksiyonuna göre biraz daha yüksek olmasıdır. Bununla birlikte, sigmoid fonksiyonuna kıyasla simetrik bir çıkışa sahip olması ve sıfır merkezli olması, bazı

durumlarda tercih edilmesini sağlar.

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

• Sigmoid

Sigmoid fonksiyonunun çıkış değeri 0 ile 1 arasında sınırlıdır. Bu, çıktının belirli bir aralıkta kalmasını sağlar ve sınıflandırma problemlerinde olasılık değerlerini temsil etmek için kullanılabilir.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

• Softmax

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad (2.11)$$

Softmax fonksiyonu, çok sınıflı sınıflandırma problemlerinde kullanılan bir aktivasyon fonksiyonudur. Genellikle çıktı katmanında kullanılır ve her bir sınıf için olasılık değerlerini hesaplar.

Softmax fonksiyonunun amacı, giriş vektöründeki değerleri olasılık olarak yorumlayacak şekilde normalize etmektir. Her bir sınıfın olasılığını belirlerken, giriş vektöründeki her bir elemanın eksponansiyelini alır ve bu değerleri toplar. Ardından, her elemanın eksponansiyelini toplamın üzerine böler. Sonuç olarak, softmax fonksiyonu, tüm sınıfların olasılık dağılımını veren bir vektörü elde etmemizi sağlar.

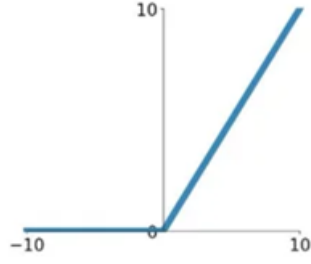
Softmax fonksiyonunun önemli bir özelliği, çıktı değerlerini [0, 1] aralığına sıkıştırması ve bu değerlerin toplamının 1 olmasıdır. Bu özellik, çıktıları olasılık olarak yorumlamamıza olanak tanır ve sınıflar arasında karşılaştırma yapmamızı kolaylaştırır.

Örneğin, bir görüntünün sınıflandırılması için softmax fonksiyonu kullanılıyorsa, her bir sınıfın olasılığını hesaplar ve en yüksek olasılığa sahip sınıfı tahmin olarak seçeriz.

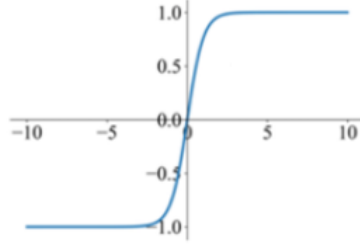
Softmax fonksiyonu, derin öğrenme modellerinde yaygın olarak kullanılır ve sınıflandırma problemlerinde etkili sonuçlar verir.

Bazı aktivasyon fonksiyonlarının grafiksel gösterimi aşağıdaki gibidir.

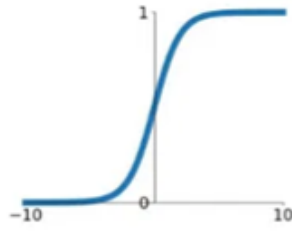
Derin öğrenme algoritmalarında problemin türüne göre ağın eğitimi esnasında seçilecek olan aktivasyon fonksiyonları hakkında yapılan bir araştırmaya tablo 2.1'de verilmiştir. Çalışma sonucu yorumlar ve bu fonksiyonların hangi durumda kullanılacağı belirtilmiştir [23]. Burada aktivasyon fonksiyonlarının belirli algoritmalarındaki



Şekil 2.3 : ReLU fonksiyonu



Şekil 2.4 : tanh fonksiyonu



Şekil 2.5 : sigmoid fonksiyonu

çalışma prensibine yönelik yorumlar ve bu aktivasyon fonksiyonlarının derin ağlarda kullanım önerisine ilişkin açıklamalar belirtilmiştir.

Çizelge 2.1 : Derin sinir ağlarında hangi aktivasyon fonksiyonunun ne zaman kullanılacağına dair öneri

Fonksiyon	Yorum	Kullanım Önerisi
Sigmoid	Sıfır merkezli olmadığı için eğitim sırasında kaybolan gradyan işlevine ve zikzak yapmaya eğilimlidir.	Tekrarlayan sinir ağında kullanılabilir.
Tanh	Kaybolan gradyan eğilimli (vanishing gradient)	
ReLU	Gizli katmanlar için en popüler işlev.	İlk seçim olmalıdır
SoftMax		Sınıflandırma ağlarında çıktı katmanı için kullanılır.

2.2.3 Optimizasyon algoritmaları

Derin öğrenme çalışmaları temelde bir optimizasyon problemini çözmektir. Makine öğrenmesinde hata oranını en aza indirmek için sıklıkla kullanılan bazı teknikler: Uyarlanabilir Moment Tahmini (ADAM, Adaptive Moment Estimation), Karelerin Ortalamasının Karekökü Yayılımı (RMSProp, Root Mean Square Propagation), gradyan iniş algoritmalarıdır. Bu algoritmalar arasında başarımları ve hız bakımından farklılıklar bulunmaktadır. Adaptif algoritmalar öğrenme hızını kendisi öğrenmektedir ve dinamiklerdir.

Gradyan İniş (Gradient Descent, GD)

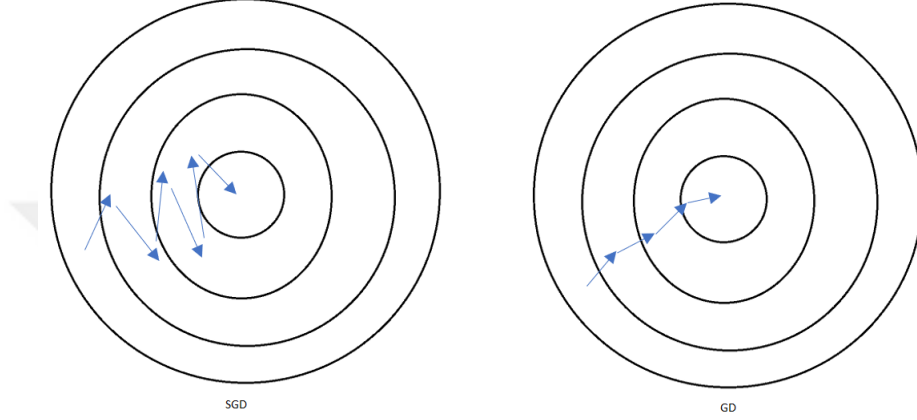
Gradyan iniş, bir kayıp fonksiyonunu optimize etmek için yinelemeli bir algoritmadır. Fonksiyonun negatif eğimi yönünde tekrar tekrar hareket ederek çalışır. Bir fonksiyonun bir noktadaki gradyanı, negatif gradyan yönünde hareket ederek gradyan inişi fonksiyonun minimumuna doğru hareket eder. Gradyan iniş, belirli bir kayıp fonksiyonunu en aza indiren model parametrelerini bulmayı amaçlar. Derin öğrenme bağlamında bu kayıp fonksiyonu, tahmin edilen çıktılar ile gerçek hedef değerler arasındaki hatayı ölçer. Temel kullanımında GD, her yinelemede eğitim veri kümesinin tamamını kullanarak gradyanı hesaplar, bu da onu büyük veri kümeleri için hesaplama açısından maliyetli hale getirir.

Stokastik gradyan iniş (Stochastic Gradient Descent, SGD), her yinelemede tek bir eğitim örneği veya küçük bir eğitim örnekleri alt kümesi kullanarak model parametrelerini güncelleyen bir gradyan iniş çeşididir. Bu, özellikle büyük veri kümeleri için SGD'yi gradyan inişinden çok daha hızlı hale getirir. Ancak SGD, her yinelemede eğitim verilerinin yalnızca küçük bir alt kümesini kullandığından, gradyan inişinden daha az doğru olabilir.

SGD, eğitim hızını hızlandırmak için tasarlanmıştır. SGD yöntemi, yalnızca bir rastgele eğitim örneğinin (x_i, y_i) kaybını hesaplayarak parametreyi günceller [24]. SGD, parametreleri daha sık güncellediği için minimuma GD'den genellikle daha hızlı yakınsar.

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} J_i (\theta_t; (x^{(i)}, y^{(i)})) \quad (2.12)$$

Burada θ sinir ağının ağırlıklarını ve sapmalarını, $J(\theta)$ kayıp fonksiyonunu, $-\nabla_{\theta} J(\theta_t)$, negatif gradyanı, α ise öğrenme oranını temsil eder. Şekil 2.6'de gösterildiği gibi, SGD yöntemi global minimuma ulaşmak için gradyan inişten daha fazla adım gerektirse de, güncelleme için daha az hesaplama kullanıldığı için eğitim seti büyük olduğunda yakınsama çok daha hızlıdır.



Şekil 2.6 : Veri kümesinde minimuma yakınsamanın rastgele gradyan iniş ve gradyan iniş gösterimi

Bir n eğitim örneği kümesi $(x^{i:i+n}, y^{(i:i+n)})$ üzerinde bir parametre güncellemesi 2.13 denkleminde belirtilmiştir.

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} J_i (\theta_t; (x^{(i:i+n)}, y^{(i:i+n)})) \quad (2.13)$$

Uyarlanabilir Moment Tahmini (Adaptive Moment Estimation, ADAM)

Uyarlanabilir moment tahmini optimizasyon algoritması [25], derin öğrenmede sıklıkla kullanılan bir algoritmadır. Bu yöntem, gradyanın üstel olarak ağırlıklı ortalamasını ve aynı zamanda gradyanın karesini (gradyanın birinci ve ikinci momentlerini) alır. Ardından, RMSProp gibi parametreleri günceller, ancak gradyanın ve gradyanın karesinin sapmasının düzeltilmiş versiyonunu kullanır.

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) \nabla_{\theta} J_i (\theta_t) \quad (2.14)$$

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) (\nabla_{\theta} J_i (\theta_t))^2 \quad (2.15)$$

$$v_t^g = \frac{v_t}{1 - \beta_1^t} \quad (2.16)$$

$$s_t^g = \frac{s_t}{1 - \beta_2^t} \quad (2.17)$$

$$\theta_{t+1} = \theta_t - \alpha \frac{v_t^g}{\sqrt{s_t^g + \epsilon}} \quad (2.18)$$

Üst indis g güncellenmiş ifadeleri gösterir. β_1 and β_2 ayrıca seçilmesi gereken iki hiper parametredir. Çok çeşitli farklı sinir ağı mimarileri için farklı amaçlarda kullanılır. Eğitim için uygun bir öğrenme oranı seçmek önemlidir. Öğrenme oranı çok büyük ise sinir ağı düşük hata değerlerine yakınsamayabilir, çok küçük öğrenme oranları ise eğitim sürecini çok yavaşlatır. Öğrenme oranının seçimi, mevcut probleme bağlıdır. Eğitim sürecini gözlemlemek için 0.1 gibi büyük bir öğrenme oranıyla başlanılabilir ve ardından 0.01, 0.001 gibi daha küçük değerler denenebilir. Örneğin, [0.001, 0.01] gibi makul bir daha küçük öğrenme oranı aralığını bildikten sonra, en uygun olanını bulmak için bu aralıktan rastgele örneklem yapılabilir.

Bir öğrenme oranı, başlangıçta eğitim sürecini hızlandırabilir, ancak minimum etrafında dalgalanabilir ve gerçekten yakınsamayabilir. Eğer bu öğrenme oranı daha küçük bir değer olarak değiştirilirse, tüm eğitim süreci verimsiz olur. Genellikle eğitim süreci boyunca öğrenme oranını kademeli olarak azaltmak faydalı olur. Öğrenme oranını azaltmanın bazı örnekleri,

$$\alpha = \begin{cases} \alpha_0 \frac{1}{1+\eta t} \\ \alpha_0 \eta^t \\ \alpha_0 \frac{c}{\sqrt{t}} \end{cases} \quad (2.19)$$

şeklinde. Burada α_0 , öğrenme oranının başlangıç değerini, t mevcut eğitim sayısını, $\eta \in (0, 1)$ azalma oranını ve $c > 0$, ihtiyaca göre ayarlanmış bir sabiti gösterir.

Kök Ortalama Karesel Yayılma (Root Mean Square Propagation, RMSProp)

Kök ortalama karesel yayılma (RMSProp), grup gradyan inişini hızlandırabilen bir yöntemdir. Bu yöntem, gradyanın karesinin ortalamasını alır ve gradyanı bu ortalama karenin kareköküne böler.

$$v_t = \beta v_{t-1} + (1 - \beta) (\nabla_{\theta} J_i(\theta_t))^2 \quad (2.20)$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\nabla_{\theta} J_i(\theta_t)}{\sqrt{v_t + \epsilon}} \quad (2.21)$$

burada ϵ , gradyanın sıfıra bölünmesini önlemek için kullanılan küçük bir değerdir.

2.2.4 Geri yayılım (back propagation)

Backpropagation, kısaca "hata geriye yayılma" olarak bilinir ve derin öğrenmede kullanılan temel bir algoritmadır. Bu algoritma, yapay sinir ağlarını eğitmek için kullanılır. Ağı iç parametrelerini (ağırlıklar ve sapmalar) belirli bir kayıp fonksiyonunu en aza indirecek şekilde ayarlamasına olanak tanıyan bir optimizasyon tekniğidir. Backpropagation, kayıp fonksiyonunun model parametrelerine göre gradyanlarının hesaplandığı ve bu gradyanların parametreleri güncellemek için kullanıldığı gradyan tabanlı bir optimizasyonun bileşenidir.

Ağın ileri geçişi (forward pass) sırasında, girdi verisi sinir ağından geçirilerek tahmin edilen çıktılar hesaplanır. Belirli bir katman için önceki katmandan gelen girdilere ağırlıklar (W) ve sapmalar (b) dahil edilerek ağırlıklı toplamı hesaplanır $z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)}$. Toplama bir aktivasyon fonksiyonu (örneğin, sigmoid, ReLU) uygulanır $a^{(l)} = f(z^{(l)})$. Son katmanın çıktısı tahmin edilen değerleri temsil eder.

Tahmin edilen değerler ile gerçek hedef değerler arasındaki kayıp (hata) uygun bir kayıp fonksiyonunu kullanılarak hesaplanır (örneğin, regresyon için ortalama karesel hata, sınıflandırma için çapraz entropi).

Geri yayılma sırasında, model parametrelerine göre kayıp fonksiyonunun gradyanları katman katman hesaplanır ve çıkış katmanından başlayarak geriye doğru yayılır. Belirli bir katman l için, ağırlıklı toplam $z^{(l)}$ göre gradyan $\frac{\partial L}{\partial z^{(l)}}$ zincir kuralını kullanarak hesaplanır.

Ağırlık ve sapma değerlerinin güncellenmesi için seçilen optimizasyon algoritması (örneğin, gradyan inişi) kullanılır.

$$W^{(l)} = W^{(l)} - \alpha \cdot \frac{\partial L}{\partial W^{(l)}} \quad (2.22)$$

$$b^{(l)} = b^{(l)} - \alpha \cdot \frac{\partial L}{\partial b^{(l)}} \quad (2.23)$$

Burada, α öğrenme hızını temsil eder ve parametre güncellemelerini kontrol eder. Eğitim sırasında ileri ve geri geçiş birkaç kez tekrarlanır, belirli bir sayıda iterasyon (epoch) veya yakınsama elde edilene kadar devam eder. İleri geçiş, kayıp hesaplama, geri geçiş ve parametre güncellemeleri eğitim sırasında tekrarlanır. Zaman içinde sinir ağı, verilerden karmaşık temsiller öğrenir ve kayıp fonksiyonunu minimize ederek doğru tahminler yapabilme yeteneğini geliştirir.

2.2.5 Kayıp fonksiyonları

Kayıp fonksiyonu tasarlanan modelin hata oranını aynı zamanda başarımını ölçen fonksiyondur. Derin ağların son katmanı kayıp fonksiyonun tanımlandığı katmandır. Kayıp fonksiyonu temelde modelin yaptığı tahminin, gerçek değerden ne kadar farklı olduğunu hesaplamaktadır. Bu nedenle iyi tahmin eden bir model oluşturulmamış ise gerçek değer ile tahmin edilen değer arasındaki fark yüksek olacak dolayısıyla kayıp değeri yüksek olacak, iyi modele sahip olan ağlarda kayıp değeri az olacaktır. Birebir aynı olduğu durumda kayıp 0 olacaktır.

Bir eğitim kriterleri tanımlandıktan sonra, yinelenen optimizasyon prosedürleri uygulamasına girilir. Doğrulama seti (validation set) performansının izlenmesi, modelde genellemenin eğitim yinelemeleri arasında en iyi olduğu tahmin edilen bir noktada durmasına izin verir. Buna erken durdurma (early stopping) denir. Çok katmanlı sinir ağları ve genel olarak derin öğrenme için en sık kullanılan optimizasyon prosedürleri rastgele gradyan iniş varyantları veya doğal gradyan yöntemleridir. İkinci dereceden ve doğal gradyan yöntemleri, büyük sinir ağları için hesaplama olarak maliyetlidir, çünkü bunlar, parametre sayısının karesine eşit boyuttaki matrisleri içerir. Daha küçük veri kümelerinde veya hesaplama paralel hale getirilebildiğinde, ikinci dereceden yöntemlerin hesaplama avantajı vardır, çünkü bunların paralelleştirilmeleri kolaydır ve birim zaman başına birçok güncelleme gerçekleştirebilirler. Tüm bu

kriterler hesaba katıldığında özellikle çıkış katmanlarında kullanılan bazı algoritmalar aşağıdaki gibidir.

Kategorik Çapraz Entropi (Categorical Cross Entropy)

Kategorik Çapraz Entropi, sınıflandırma problemlerinde yaygın olarak kullanılan bir kayıp fonksiyonudur. Bu kayıp fonksiyonu, tahmin edilen sınıf dağılımıyla gerçek sınıf dağılımı arasındaki farkı ölçerek modelin performansını değerlendirir.

Kategorik Çapraz Entropi'nin formülü aşağıdaki gibi ifade edilir:

$$L(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i) \quad (2.24)$$

L kayıp fonksiyonunu temsil eder. y gerçek sınıf dağılımını temsil eden bir vektördür. Bu vektörde, gerçek sınıf için ilgili etiket 1 olarak ifade edilirken diğer etiketler 0 olarak ifade edilir.

$\hat{y}$ tahmin edilen sınıf dağılımını temsil eden bir vektördür. Bu vektörde, modelin her sınıf için tahmin ettiği olasılıklar bulunur.

Formüldeki toplam, sınıf sayısı kadar olan i değeri için gerçekleştirilir. Her bir y_i ve $\hat{y}_i$ gerçek ve tahmin edilen sınıf olasılıklarını temsil eder. Kategorik Çapraz Entropi, gerçek sınıf ve tahmin edilen sınıf arasındaki farkı hesaplar. Eğer gerçek sınıf 1 olarak etiketlenmişse ($y_i = 1$) kayıp değeri tahmin edilen sınıf olasılığının negatif logaritmasıdır $-\log(\hat{y}_i)$. Eğer gerçek sınıf 0 olarak etiketlenmişse ($y_i = 0$), bu terim kayba katkıda bulunmaz. Modelin doğru sınıfı doğru bir şekilde tahmin etmesi durumunda, kayıp fonksiyonunun değeri düşük olur. Ancak, yanlış sınıflandırmalar veya belirsiz tahminler durumunda kayıp fonksiyonu değeri artar.

Bu kayıp fonksiyonu, sınıflandırma problemlerinde modelin eğitiminde sıklıkla kullanılır. Modelin hedeflenen sınıfı doğru bir şekilde tahmin etmesini teşvik eder ve istenmeyen tahmin hatalarının azaltılmasına yardımcı olur.

İkili Çapraz Entropi (Binary Cross Entropy, BCE)

Eğer bir derin öğrenme modelinin (örneğin otokodlayıcının) çıkış katmanının aktivasyon fonksiyonu bir sigmoid fonksiyonu ise, nöron çıkışlarını 0 ile 1 arasında sınırlar ve giriş özellikleri 0 ile 1 arasında normalize edilirse, burada L_{CE} ile gösterilen

ikili çapraz entropi kaybı fonksiyonu kullanılabilir. Bu kayıp fonksiyonu genellikle sınıflandırma problemlerinde kullanılır, ancak otokodlayıcılar için de geniş çalışma alanı vardır [26]. 2.25 denkleminde olduğu gibi ifade edilir:

$$L_{CE} = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^n [x_{j,i} \log \tilde{x}_{j,i} + (1 - x_{j,i}) \log (1 - \tilde{x}_{j,i})] \quad (2.25)$$

Burada $x_{j,i}$, i . gözlemin j . bileşenidir. M , eğitim veri setindeki gözlem sayısıdır. Toplam, tüm gözlemler kümesi ve vektörlerin tüm bileşenleri üzerinde gerçekleştirilir. Bu kayıp fonksiyonunu minimize etmek, girişi mümkün olduğunca benzer bir şekilde yeniden oluşturmakla eşdeğerdir. L_{CE} 'yi en aza indirmek için $\tilde{x}_i$ 'nin hangi değerleri alması gerektiğini şu şekilde ifade edilir :

$$\frac{\partial L_{CE}}{\partial \tilde{x}_i} = 0 \quad i = 1, \dots, M \quad (2.26)$$

İkili çapraz entropi L_{CE} değeri, $i = 1, \dots, M$ için $x_i = \tilde{x}_i$ olduğunda en aza indirilmiş olur.

2.3 Otokodlayıcılar

Otokodlayıcılar, beklenen çıktının girdinin tam bir kopyası olduğu denetimsiz öğrenme algoritmalarıdır. Bunu başarmak için, otokodlayıcılar, verilerden iyi temsiller öğrenmeye çalışan, verilerin gizli bir temsilini bulur. Bu yapı girişinden aldığı veriyi çıkışında da orijinaline en benzer şekilde tekrar oluşturmak için eğitilir [18].

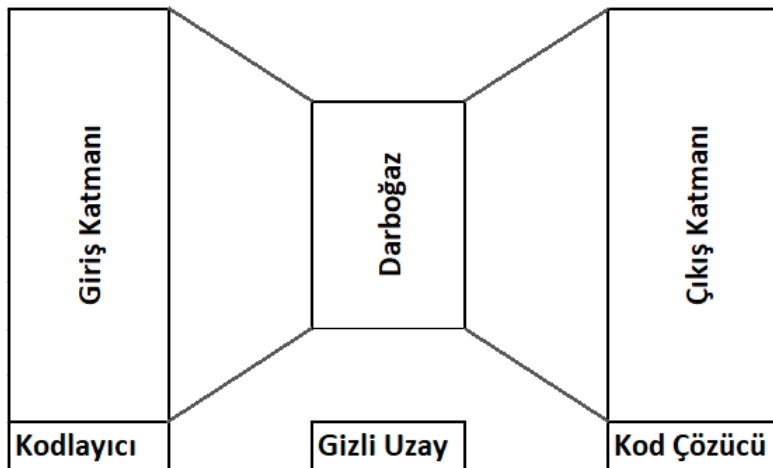
Otokodlayıcılar , tipik olarak Ortalama Karesel Hatayı (Mean Squared Error, MSE) en aza indirmeye çalışarak, optimizasyon kriterleri olarak yeniden yapılandırma maliyetini kullanarak bir girdinin ara, daha düşük boyutlu kodlamasını öğrenir. Ham girdileri daha düşük boyutlu gizli seyrek gösterime kodlayan bir kodlayıcıdan ve çıktı olarak girdi vektörü için bir tahmin oluşturan bir kod çözücünden oluşurlar [27]. Otokodlayıcılar, veri sıkıştırma ve sıkıştırılmış verilerin iletiminde yaygın olarak kullanılabilir. Otokodlayıcılarla veri sıkıştırmanın nerede kullanılabileceği konusunda bazı temel fikirler vermek için, burada otokodlayıcıların nasıl çalıştığı incelenmelidir.

Otokodlayıcılar, kendi kendini denetleyen bir şekilde çalışan bir tür sinir ağıdır. Dolayısıyla, otokodlayıcılarda üç ana yapı taşı vardır: kodlayıcı, gizli uzay gösterimi (ya da diğer adıyla darboğaz (bottleneck)) ve kod çözücü. Bir otokodlayıcının amacı; ağı, giriş verisinin en önemli kısımlarını yakalaması için eğiterek, tipik olarak boyutsallığı azaltmak için, daha yüksek boyutlu bir veri için daha düşük boyutlu bir gösterimi (kodlamayı) öğrenmektir.

Kullanım alanlarına örnek olarak; özellik çıkarma (feature extraction), boyut azaltma, aykırı veri çıkarma söylenebilir. Bu anlamda otokodlayıcı, temel bileşenler analizi (Principal Component Analysis-PCA) ile bazı benzerlikler göstermektedir. PCA’de tıpkı otokodlayıcı gibi büyük boyutlu verileri daha az boyutla temsil etmek için kullanılabilen bir boyut indirgeme methodudur. Aralarındaki temel farklılık; PCA, bu işlemi verileri daha düşük boyutlu bir uzaya taşıyan doğrusal dönüşümler ile yaparken, otokodlayıcı sahip olduğu birçok katmandan oluşabilen ileri beslemeli ağ mimarisi ile doğrusal olmayan problemler için de bunu gerçekleyebilmektedir.

2.3.1 Otokodlayıcı mimarisi

Otokodlayıcılar temelde 3 bölümden oluşur. İlk olarak, girdi, sıkıştırıldığı ve gizli uzay gösterimi adı verilen katmanda depolandığı kodlayıcıdan geçer, ardından kod çözücü orijinal girdiyi koddan yeniden açar (decompression). Otokodlayıcının temel amacı, girdi ile aynı çıktıyı elde etmektir. Bu sistemde giriş ve çıkış aynı boyutta olmalıdır.



Şekil 2.7 : Otokodlayıcı genel yapısı

Bir otokodlayıcı, giriş verilerini minimum hatayla yeniden temsil edebilen bir sinir ağı sınıfıdır. Otokodlayıcılar başlangıçta boyut indirgeme için tanıtılmıştır, çünkü giriş verilerini öğrenilen çok boyutlu bir gizli uzay olan $z \in \mathbb{R}^{1 \times d}$ boyutuna sıkıştırır ve gizli uzay temsilden giriş boyutuna yeniden veri oluşturur. Bir otokodlayıcı, bir kodlayıcı ve bir çözücü olarak ifade edilebilir. Kodlayıcı f_e , boyutu d olan giriş verilerini $x \in \mathbb{R}^{1 \times d}$, öğrenilen bir çok boyutlu gizli uzayı olan $z \in \mathbb{R}^{1 \times d}$ boyutuna sıkıştırır. Burada z , denklem 2.27 tarafından verilen kodlamadır ve W_e ve b_e kodlayıcının ağırlıkları ve sapmaları olup, eğitilebilir kodlayıcı parametreleri θ_e tarafından birlikte temsil edilir. ϕ herhangi bir doğrusal olmayan aktivasyon fonksiyonudur. Çözücü f_d , veriyi $x' \in \mathbb{R}^{1 \times d}$ giriş boyutuna gizli uzay temsilden d_l boyutuna yeniden oluşturma yeteneğine sahip başka bir sinir ağıdır. Burada x' , denklem 2.28 tarafından verilen gizli uzay temsilden giriş boyutuna yeniden oluşturulan veridir. W_d ve b_d çözücünün ağırlıkları ve sapmalarıdır ve θ_d tarafından birlikte temsil edilir. Otokodlayıcı parametreleri $\theta = [\theta_e, \theta_d]$ ile karakterize edilir. Otokodlayıcılar, kodlayıcının ve kod çözücünün ağırlıklarının birbirinin transpozu olduğu yani $W_e = W_d^T$ ve katmanların aktivasyon fonksiyonuna sahip olduğu durumda temel bileşenler analizine (PCA) indirgenebilir [28].

$$z = f_e(x) = \phi(W_e x + b_e) \quad (2.27)$$

$$x' = f_d(f_e(x)) = f_d(z) = \phi(W_d z + b_d) \quad (2.28)$$

Kodlayıcı (Encoder)

Bir kodlayıcı, girdiyi gizli bir uzay temsiline sıkıştıran ve girdi görüntüsünü küçültülmüş bir boyutta sıkıştırılmış bir temsil olarak kodlayan, ileri beslemeli, tamamen bağlı bir sinir ağıdır. Sıkıştırılmış görüntü, orijinal görüntünün bozuk halidir.

Gizli uzay gösterimi (bottleneck)

Ağın bu kısmı, kod çözücüye beslenen girişin indirgenmiş temsilini içerir. Darboğaz, kodlayıcıdan kod çözücüye bilgi akışını kısıtlamak için vardır, böylece yalnızca en önemli bilgilerin geçmesine izin verir. Darboğaz, bir verinin sahip olduğu maksimum bilginin içinde tutulacağı şekilde tasarlandığından, darboğazın girdinin

bilgi temsiliinin oluřturulmasına yardımcı olduđunu sylenebilir. Bu nedenle, kodlayıcı-kod  yapıısı, veri biimindeki bir grntden en fazlasını ıkarmamıza ve ađ iindeki eřitli girdiler arasında yararlı iliřkiler kurmamıza yardımcı olur. Girdinin sıkıřtırılmıř bir temsili olarak bir darbođaz, sinir ađının girdiyi ezberlemesini ve verilerin ařırı đrenilmesini (overfitting) engeller. Genel bir kural olarak, darbođaz ne kadar kkse, overfitting riski o kadar dřk olur. Ancak ok kk darbođazlar saklanabilir bilgi miktarını kısıtlar, bu da nemli bilgilerin kodlayıcının katmanlarından dođru olmayan bilgi ıkma olasılıđını artırır.

Kod  (decoder)

Kod  de kodlayıcı gibi ileri beslemeli bir ađdır ve kodlayıcıya benzer bir yapıya sahiptir. Bu ađ, girdiyi koddan orijinal boyutlara geri dndrmekten sorumludur.

Utan uca bir model olarak otokodlayıcılar etiketlenmemiř (ıkıřa giriřin olması gereken veri bilgisinin verilmediđi durum) bir veri kmesini alıp $\hat{x}$ ıktısı, orijinal x giriřinin yeniden yapılandırılması ile grevli đrenme problemi olarak aıklanabilir. Bu ađ, orijinal girdi ile yeniden yapılandırılmıř veri arasındaki farkları len yeniden oluřturma hatası en aza indirilerek eđitilebilir. Kodlayıcı, x giriřini, giriř verilerine kıyasla daha kk boyutları olan darbođaz katmanı ile gizli bir alana eřler. Daha sonra, kod , girdi verisini yeniden yapılandırmak ve retmek iin bu gizli alanı kullanacaktır. Ađın akıřındaki temel ama giriř vektr, sıkıřtırılmıř bir gsterimden, genellikle $\hat{x}$ olarak gsterilen bir ıktı vektrne yaklařmaktır. Bu, rneđin ortalama kare hatası kullanılarak bir yeniden yapılandırma kriterinin en aza indirilmesiyle gerekleřtirilebilir.

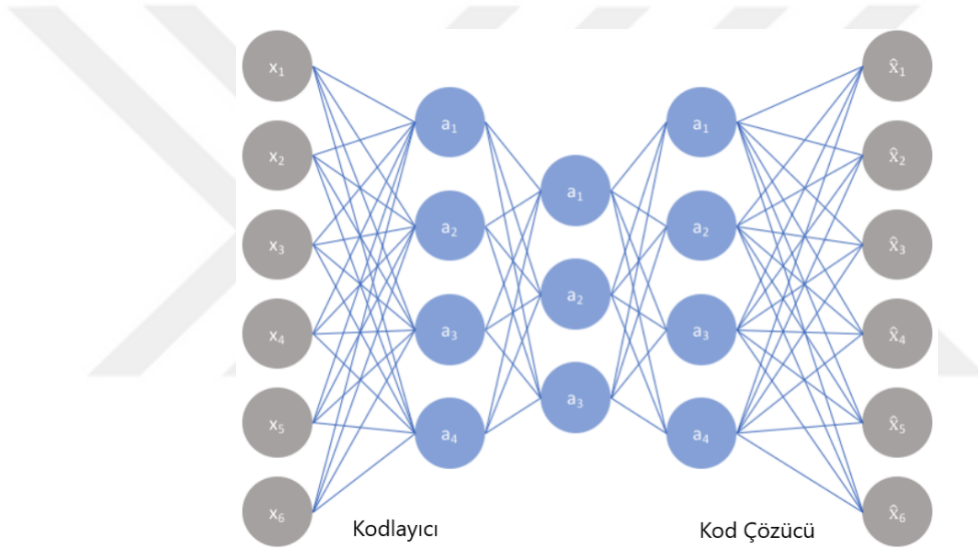
$$L(x, g(f(x))) = \|x - \hat{x}\|^2 \quad (2.29)$$

Gizli katmanın boyutsal olarak ok daha kk olması geređi, ađın dođru bir řekilde yeniden yapılandırmak iin veri reten dađılımın nemli bir temsilini đrenmesi gerektiđini garanti eder. Bu nedenle, ađın yalnızca kimlik iřlevini đrenmesi, yani girdiyi ıktı katmanına basite kopyalamaması gerekir.

Tamamlanmamıř Otokodlayıcılar (Under Complete Autoencoders)

Orta katmandaki dđmlerin sayısı giriř dđmlerinin sayısından azsa, ađa tamamlan-

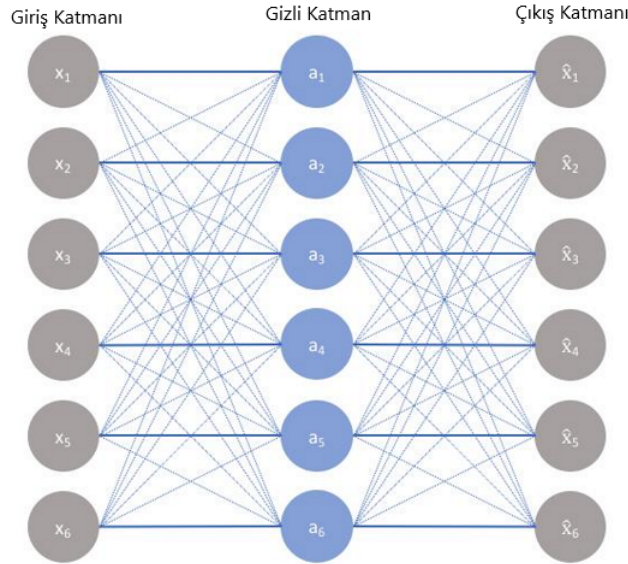
mamış otokodlayıcılar adı verilir. Tamamlanmamış otokodlayıcılar, verilerin önemli özelliklerini yakalamak için kod dağılımlarının kısıtlandığı en basit otokodlayıcı türleridir. Yani, girdi verilerinin en önemli özelliklerini yakalamak için tasarlanmıştır. Bu nedenle, boyutluluğun azaltılması, ana hedeflerinden biridir. Tamamlanmamış otokodlayıcılar, Temel bileşen analizinin gelişmiş versiyonu olarak kabul edilebilir. Temel bileşen analizi yalnızca verilerin doğrusal özelliklerini yakalarken, tamamlanmamış otokodlayıcılar herhangi bir hiper boyutlu şemayı yakalayabilir. Sinir ağlarının özellik çıkarma yeteneği, tamamlanmamış otokodlayıcıların veri temsili için herhangi daha yüksek boyutlu kopyasını bulmasını ve bunu daha düşük boyutluluğa kodlamasını sağlar. [29] çalışmasında, boyut indirgeme, tamamlanmamış otokodlayıcıların pratik uygulaması olarak gösterilmektedir.



Şekil 2.8 : Tamamlanmamış Otokodlayıcı

Tam Otokodlayıcılar (Complete Autoencoders)

Eğer kod katmanındaki düğüm sayısı giriş katmanındaki düğüm sayısı ile aynı ise, ağa tam otokodlayıcı denir. Bu durumda, orta katmandaki (kod katmanı, darboğaz) düğüm sayısı giriş vektörünün boyutuna eşittir. Yani, tam otokodlayıcı durumunda kod katmanının boyutunu belirlenmez, bunun yerine giriş katmanının boyutu ile eşit olur. Şekil 2.9, tam otokodlayıcıların temsilini göstermektedir.



Şekil 2.9 : Tam Otokodlayıcı

Aşırı Tam otokodlayıcılar (Over Complete Autoencoders)

Aşırı tam otokodlayıcılar, öznelikler arasındaki ilişkilerin giriş katmanı ile aynı sayıda düğüme özetlenemediği durumlarda, model gizli özellik haritalarını çıkarmak için kod katmanına daha fazla düğüm eklemesi yapması gerekecektir. Eğer kodun boyutu giriş/çıkış katmanındaki düğüm sayısından daha büyük ise, ağ aşırı tam kodlayıcılar olarak adlandırılır. Aşırı tam otokodlayıcılarda kod katmanı artık darboğaz olarak nitelendirilmez çünkü giriş katmanı boyutundan daha küçük değerlere sınırlı değildir.

2.3.2 Otokodlayıcı türleri

Farklı türlerde otokodlayıcı türleri vardır ve bunlardan bazıları aşağıda kısa açıklama ile belirtilmiştir.

2.3.2.1 Evrişimli otokodlayıcılar (convolutional autoencoder, CAE)

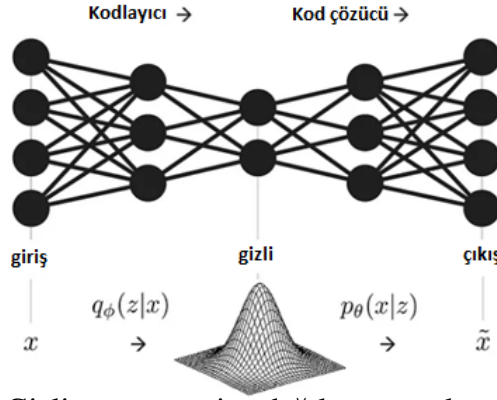
Evrişimli Otokodlayıcılar, derin öğrenme ve otokodlayıcıların bir kombinasyonudur. Evrişimli otokodlayıcılar, giriş verilerini öğrenmek, temsil etmek ve yeniden oluşturmak için evrişimli sinir ağlarından (Evrişimli Sinir Ağları, Convolutional Neural Networks, CNN) oluşan bir yapı kullanır. Evrişimli otokodlayıcılar, genellikle görüntü verilerinin boyutsal yapısını koruyan ve verinin önemli özelliklerini çıkaran

etkili bir şekilde öğrenirler. Bunun için, kodlayıcı ve çözücü olmak üzere iki temel bileşene sahiptirler. Kodlayıcı bölümü, Giriş verilerini daha düşük boyutlu bir temsile dönüştürmek için evrişimli katmanlar, örnekleme (pooling) katmanları ve tam bağlı katmanlar içeren bir yapı kullanır. Bu katmanlar, verinin önemli özelliklerini çıkarmak için farklı filtreleri kullanır ve veriyi sıkıştırır. Bu kodlama süreci, verinin önemli özelliklerini yakalama ve özünü çıkarma yeteneğine sahiptir. Çözücü bölümü ise, kodlanmış temsili alır ve orijinal giriş verisini yeniden oluşturmak için evrişimli katmanlar, ters örnekleme (unpooling) katmanları ve aktivasyon fonksiyonları kullanır. Bu adım, kodlayıcı tarafından elde edilen düşük boyutlu temsili yüksek boyutlu giriş verisine geri dönüştürme görevini üstlenir [30]. Evrişimli Otokodlayıcılar, denetimsiz öğrenme görevlerinde ve veri sıkıştırma uygulamalarında kullanılırlar. Özellikle görüntü işleme ve video analizi gibi alanlarda popülerdirler. Ayrıca veri görselleştirme gibi görevlerde de kullanışlıdır.

2.3.2.2 Değişken otokodlayıcı (variational autoencoder, VAE)

Değişken otokodlayıcı (VAE), denetimsiz ve denetimli öğrenmede yaygın olarak uygulanan üretken bir modeldir. Değişken otokodlayıcı, gizli uzaydaki vektörlerin dağılımını normal dağılıma zorlayan denetimsiz otokodlayıcı modelidir [31]. Otokodlayıcıların, özellik çıkarmada denetimsiz öğrenmede sahip oldukları yeteneklere rağmen ezberleme olasılığı olan önemli bir sorunu vardır. Gizli katman veya darboğaz gibi bir düğüme sahip olsa bile, ağ için ezberleme olasıdır. Sinir ağı kodlayıcı ve kod çözücü içeren otokodlayıcıda kodlayıcının giriş değişkeni x 'i aldığını ve bunu bazı koşullu dağılım $q(z|x)$ tarafından açıklandığı gibi gizli değişken z 'ye eşlediğini varsayalım. Kod çözücü daha sonra gizli uzay değişkeni z 'yi alır ve onu bir koşullu dağılım $p(x|z)$ ile x 'e geri eşler. Otokodlayıcılar aşırı öğrenme gibi ciddi sorunlardan etkilenme eğilimindedir ve gizli değişken genellikle çok seyrek bir dağılıma sahip olacaktır ki bu da istenmeyen bir durumdur. Bu sorunları azaltmanın bir yolu, gizli değişkeni Gauss dağılımı gibi belirli bir dağılıma uymaya zorlayarak düzenleştirme eklemeektir. Bu da kayıp fonksiyonuna KL-sapma (Kullback-Leibler) terimi eklenerek elde edilebilir.

Bir deęişken otokodlayıcı, esas olarak normal bir otokodlayıcı ile aynıdır, ancak gizli alan deęişkeninin belirli bir dağılıma uymasına neden olan kayıp işlevine ek bir düzenlileştirme ile birlikte gelir.



Şekil 2.10 : Gizli uzay gaussian dağılmaya zorlanır (bottleneck).

Deęişken otokodlayıcı, görüntü işleme [32], metin veri işleme [33] dahil olmak üzere birçok alanda kullanılmaktadır. Kör kanal denkleştirme (blind channel equalization) için deęişken otokodlayıcı kullanan bir yöntem de [34]’de önerilmiştir ve bu çalışma, kablosuz MIMO sistemindeki ön kodlama (hüzme oluşturma) problemini çözmek için deęişken otokodlayıcının faydalarını göstermektedir. Veri iletimi alanında deęişken otokodlayıcı kullanan bir yöntem [35] önerilmiştir.

2.3.2.3 Gürültü çözen otokodlayıcı (denoising autoencoder, DAE)

Gürültü çözen otokodlayıcıların temel amacı, gürültülü verileri temizlemek için öğrenme süreci boyunca verinin içsel yapısını yakalamaktır. Bu süreç, modelin veri üzerindeki gürültüyü anlamasını ve gürültüden arındırılmış bir temsil oluşturmalarını sağlar. Böylece, gürültü çözen otokodlayıcı, veri iyileştirme, gürültü azaltma ve anormallik tespiti gibi uygulamalarda etkili bir şekilde kullanılabilir. Gürültü giderme, ham girdi verilerine eklenen rastgele gürültüyü otokodlayıcıya beslemeden önce kaldırmayı ifade eder [36]. Bu nedenle, bir gürültü çözen otokodlayıcı bozulmuş verileri alır ve onu gürültüden arındırması ve gürültüsüz veriler üretmesi beklenir. Bir gürültü çözen otokodlayıcının çıktısı, geri yayılım sırasında kayıp fonksiyonunu hesaplarken gürültülü olanla deęil, gerçek veri örneęiyle karşılaştırılır. Bu nedenle, bir gürültü çözen otokodlayıcı, gürültülü bir girdideki bozulmayı ortadan kaldırmaktan sorumludur ve model performansını deęerlendirmek için çıktısı karşılık gelen gerçek

örnekle (gürültüsüz) karşılaştırılır. Denklem 2.30, $f(x)$ 'in gürültülü girişi x kayıp fonksiyonu olarak kullanılır.

$$\mathcal{L}(x, \hat{x}) = |\hat{x} - g(f(x))| \quad (2.30)$$

Böylece verinin özelliklerini öğrenmeden girdiyi çıktıya kopyalamaktan kaçınılır. Bu otokodlayıcılar, orijinal bozulmamış girişi kurtarmak için eğitilirken kısmen bozuk bir girdi alır. Model, eklenen gürültüyü ortadan kaldırmak için doğal verileri tanımlayan daha düşük boyutlu bir doğru giriş verilerini eşlemek için bir vektör alanını öğrenir. Bu sayede, kodlayıcı en önemli özellikleri çıkaracak ve verilerin daha sağlam bir temsilini öğrenecektir. Dolayısıyla, kod çözücü küçük bozulmalara karşı dirençli olacaktır.





3. KABLOSUZ HABERLEŞME İÇİN OTOKODLAYICILAR

Kablosuz haberleşme, günümüzde hayatımızın ayrılmaz bir parçası haline gelmiştir. Akıllı telefonlar, tabletler, dizüstü bilgisayarlar ve diğer birçok cihaz, kablosuz ağlar üzerinden birbirleriyle iletişim kurmaktadır. Bu kablosuz haberleşme sistemlerinde, iletim kanalındaki olumsuz etkilerden kaynaklanan hataları düzeltmek veya azaltmak amacıyla otokodlayıcılar kullanılabilir.

Kablosuz haberleşme, veri iletiminin kablosuz haberleşme kanalları üzerinden gerçekleştirildiği bir haberleşme yöntemidir. Kablosuz haberleşme sistemleri modülasyon, kanal kodlama, demodülasyon, kod çözme gibi birçok farklı bileşenden oluşur.

Dijital haberleşme, analog sinyaller kullanarak dijital bilgiyi gönderme sürecidir. Dijital modülasyon, bir dizi ikili dijital bilgiyi yani bitleri analog taşıyıcı dalga üzerine aktarma sürecidir. Kodlanacak bit akışı önce K -bitlik kelimelerin bir alfabetine bölünür. Örneğin, $K = 2$ ise mümkün olan 2-bitlik kelimeler $(1, 1)$, $(0, 0)$, $(1, 0)$ ve $(0, 1)$ olabilir. Bu kelimeler daha sonra sembollere, genellikle karmaşık sayılara, eşlenir. Kelimelerin sembollerle eşleştirilmesi, her bir kelimeye karşılık gelen karmaşık düzlemdeki farklı olası sembol/noktalardan oluşan takımyıldızı (constellation) tarafından tanımlanır. Dijital modülasyon tekniği olan Faz Kaydırmalı Anahtarlama (PSK) kelimeleri taşıyıcı dalga fazını değiştirerek temsil eder.

Otokodlayıcılar, verileri özel bir şekilde kodlama ve kod çözme yöntemleriyle korumak için kullanılan hata düzeltme teknikleridir. Otokodlayıcılar, hata düzeltme kodları olarak da adlandırılır ve hedeflenen veri iletileri üzerinde bir dizi işlem gerçekleştirerek verilerin doğruluğunu artırmaya çalışır. Bu, iletim kanalındaki gürültü, sinyal zayıflaması ve diğer bozucu faktörler nedeniyle oluşabilecek hataları düzeltmeye veya algılamaya yardımcı olur.

Otokodlayıcıların çalışma prensibi, veriyi kaynakta kodlamak, iletim kanalı boyunca kodlanmış veriyi iletmek ve hedefte kod çözme işlemi yapmaktır. Kaynakta kodlama,

verinin belirli bir algoritma kullanılarak hata düzeltme kodları ile birlikte kodlanmasını sağlar. Kod çözme işlemi ise, alınan kodlanmış verinin hataları düzeltme veya algılama işlemleriyle geri orijinal veriye dönüştürülmesini hedefler.

Kablosuz haberleşme sistemlerinde otokodlayıcılar, veri bütünlüğünü korumak ve hata düzeltme yeteneklerini artırmak için yaygın olarak kullanılır. Özellikle yüksek hızlı veri iletimi gerektiren uygulamalarda, otokodlayıcılar hataların etkisini minimize ederek güvenilir bir iletişim sağlar.

3.1 Alamouti Kodlaması ve Gönderici Çeşitlemesi

Alamouti kodlaması, kablosuz iletişimde MIMO (Çoklu Giriş, Çoklu Çıkış) sistemlerde kullanılan bir çeşit uzaysal çeşitleme kodlamasıdır [17]. Bu kodlama tek bir anten yerine birden fazla antenin kullanıldığı sistemlerde kullanılır ve özellikle kablosuz haberleşmede hata oranını azaltmak ve iletim hızını artırmak amacıyla tasarlanmıştır.

Alamouti kodlaması, iki ardışık zaman diliminde gönderilen sembollerin bir çeşit çeşitleme tekniği ile kodlanmasına dayanır. 2 verici ve 1 alıcı sistemler için Alamouti, belirli bir sembol zamanı içinde iki sinyalin aynı anda iletilmesi olarak çalışır. Birinci gönderici antenden iletilen sembol x_1 ile gösterilir ve ikinci gönderici antenden iletilen sembol x_2 ile gösterilir. Sonraki sembol zamanında, sembol $(-x_2^*)$ birinci gönderici antenden iletilir ve sembol x_1^* ikinci gönderici antenden iletilir. Burada * konjuge işlemidir.

Bu kodlama, uzay-zaman kodlama kullanılarak yapılır. Bu iletim şeması tablo 3.1 'de gösterilmiştir.

Çizelge 3.1 : İki gönderici çeşitlemesi (transmit diversity) için sembol iletim sırası

	Gönderici Anten 1	Gönderici Anten 2
t zamanı	x_1	x_2
t + T zamanı	$-x_2^*$	x_1^*

Alamouti kodunda çeşitlilik, alıcı tarafındaki maximum likelihood (ML) sinyal tespitine dayanmaktadır [17]. İki kanal kazancının, h_1 ve h_2 , ardışık iki sembol zamanı boyunca değişmediği varsayılmaktadır.

$$h_1(t) = h_1(t + T_s) = h_1 = |h_1| e^{j\theta_1} \quad (3.1)$$

$$h_2(t) = h_2(t + T_s) = h_2 = |h_2| e^{j\theta_2} \quad (3.2)$$

Burada $|h_i|$ ve θ_i , i . antenin genlik kazancı ve faz dönüşünü belirtir. n_1, n_2 toplamsal beyaz Gauss gürültüsüdür. Alınan sinyal,

$$y_1 = h_1 x_1 + h_2 x_2 + n_1 \quad (3.3)$$

$$y_2 = -h_1 x_2^* + h_2 x_1^* + n_2 \quad (3.4)$$

ile ifade edilir. Bu ifadeler $h_1(t) = h_1(t + T)$ ve $h_2(t) = h_2(t + T)$ varsayımı altında matris formunda aşağıdaki gibi ifade edilir.

$$\begin{bmatrix} y_1 \\ y_2^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ -h_2^* & h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (3.5)$$

Denklemin her iki tarafını da kanal matrisinin kompleks transpozunu ile (Hermitian) çarpıldığında denklem,

$$\begin{bmatrix} h_1 & h_2 \\ -h_2^* & h_1^* \end{bmatrix} \begin{bmatrix} y_1 \\ y_2^* \end{bmatrix} = \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix} \quad (3.6)$$

$$\tilde{y}_1 = (|h_1|^2 + |h_2|^2) x_1 + \tilde{n}_1 \quad (3.7)$$

$$\tilde{y}_2 = (|h_1|^2 + |h_2|^2) x_2 + \tilde{n}_2 \quad (3.8)$$

şeklindedir. Alıcıda maksimum olabilirlik (ML) yapısı:

$$x_{i,ML} = Q \left(\frac{\tilde{y}_i}{|h_1|^2 + |h_2|^2} \right), i = 1, 2 \quad (3.9)$$

ile ifade edilir ve $Q(\cdot)$ verilen takımyıldız kümesi için bir iletim sembolünü belirleyen bir karar fonksiyonunu belirtir.

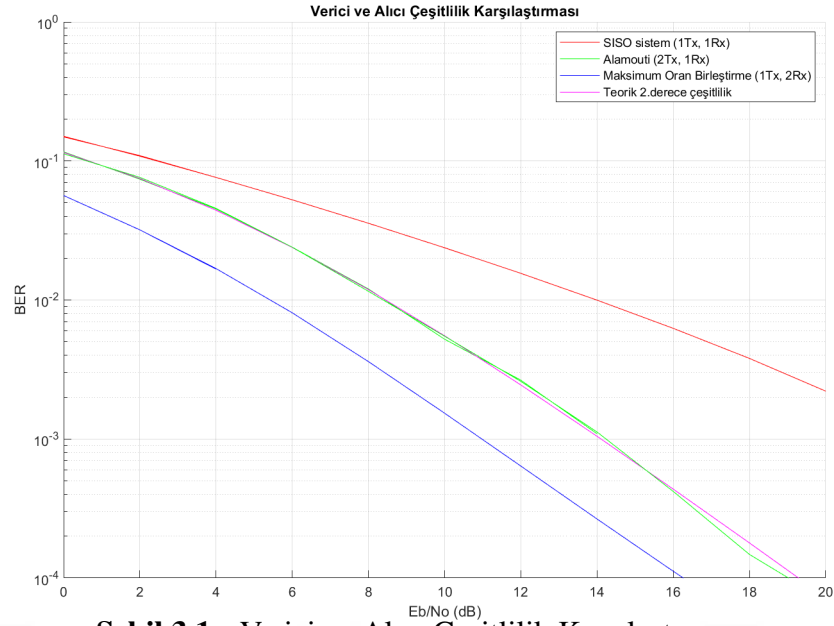
Kablosuz bir kanaldaki sinyal gücü rastgele dalgalanarak veya sönümlenerek bozulmaya uğrar. Çeşitlilik bozulmanın haberleşmedeki etkisini azaltmak için güçlü bir tekniktir. Çeşitlilik teknikleri, sinyalin birden fazla haberleşme yolu üzerinden iletilmesine dayanır. Gönderici veya alıcıdaki birden fazla anten, alıcıya zaman, frekans veya uzayda bağımsız bir şekilde iletim yolları üzerinden sinyal iletimine olanak tanıyan uzaysal çeşitlilik kazancı sunar [37]. Bu şekilde, sinyalin kopyalarının daha bozucu bir sönümleme yaşamaması olasılığı artar ve bu da sinyal alımının kalitesini ve güvenilirliğini artırır. Çeşitlilik derecesi (diversity order), N_d ile gösterilen, alıcı tarafında birleştirilebilen iletilen sinyallerin bağımsız kopyalarının sayısıdır. Bir $N_t \times N_r$ (gönderici anten sayısı x alıcı anten sayısı) MIMO sisteminde verici ile alıcı arasında $N_t N_r$ bağımsız yayılma yolu bulunabileceğinden, uzaysal çeşitliliğin (spatial diversity) aşağıdaki gibi ifade edilebilir.

$$\max \{N_d\} = N_t N_r \quad (3.10)$$

Haberleşme sisteminin performansı, çeşitlilik derecesi arttıkça iyileşir. Uzaysal çeşitliliğin diğer bir ölçüsü, Şekil 3.1'deki eğrinin lineer olduğu yüksek SNR bölgesinde eğrinin eğimini kullanmaktır. Elde edilen eğim, sistemin çeşitlilik kazancı olarak tanımlanır.

Maksimum oranlı alıcı birleştirme (Maximal-ratio receiver combining, MRRC), alıcıdaki birden fazla antenden gelen sinyalleri birleştirerek sinyal-gürültü oranını artırmayı amaçlayan bir çeşit çeşitleme ve birleştirme tekniğidir. Çeşitlilik karşılaştırması E_b/N_0 ve hata olasılığı açısından Şekil 3.1'de gösterildiği şekildedir.

Verici çeşitliliği sistemi, alıcı çeşitliliği sistemine benzer bir hesaplama karmaşıklığına sahiptir. Şekil 3.1'de de görüldüğü gibi iki verici anten ve bir alıcı antenin kullanılması, bir verici anten ve iki alıcı antenden oluşan MRC sistemle aynı çeşitlilik (eğimler benzer) performansını sağlar. Verici çeşitliliğinin, Sistemlerde kullanılan antenlerin



Şekil 3.1 : Verici ve Alıcı Çeşitlilik Karşılaştırması

toplam gücü verici çeşitliliği ve alıcı çeşitliliğinde benzer ayarlandığından, verici çeşitliliği MRC alıcı çeşitliliğine 3 dB'lik bir dezavantaja sahiptir.

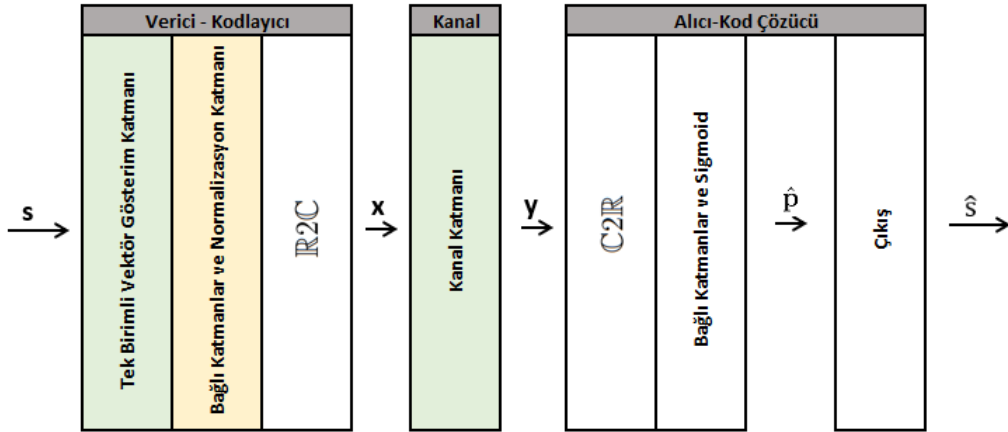
3.2 Kodlanmış ve Kodlanmamış Sistemler İçin Otokodlayıcılar

Bir haberleşme sistemi, iletilen bitlerin geri kazanılmasıyla ilgili olarak bir otokodlayıcı olarak görülebilir, çünkü haberleşme sisteminde, verici (kodlayıcı) tarafında bit eşleme ve alıcı (kod çözücü) tarafında bit çözme işlemleri kanal bozulmalarına (darboğaz katmanı) karşı dayanıklı şekilde gerçekleştirilir. Otokodlayıcı tabanlı sistemler, herhangi bir haberleşme hızı (R) için sinir ağı tabanlı verici ve alıcı düğümlerin uçtan uca öğrenmesiyle veri odaklı tasarımlar gerçekleştirmek için potansiyel bir çözüm olarak ortaya çıkmıştır [2]. Kablosuz haberleşme ağındaki bit çözme sorunu, otokodlayıcı modelinde bir sınıflandırma sorunu olarak kabul edilir. Bu nedenle model, kodlayıcının girişi ile kod çözücünün çıktısı arasındaki çapraz entropi kaybını en aza indirerek optimize edilir.

Haberleşme sistemi için otokodlayıcı kavramının kullanılması ilk olarak [38] ve [2] makalelerinde tanıtılmıştır. [2] makalesinde haberleşme sistemi bir otokodlayıcı olarak yorumlanmış, verici ve alıcı bileşenlerini tek bir süreçte birleştirerek uçtan uca yeniden yapılandırma görevi olarak haberleşme sistemi tasarımına yeni bir yaklaşım

sunulmuştur. Verilen bir kanal modeli için, blok hata oranını en aza indirmek gibi istenen bir kayıp fonksiyonu için optimize edilen verici ve alıcı uygulamalarının derin öğrenme modeli kullanılarak öğrenilebileceği gösterilmiştir. Verici, kanal ve alıcı bir otokodlayıcı olarak eğitilebilen bir model olarak uygulanmıştır. Verici ve alıcı tam bağlı sinir ağları olarak temsil edilirken, aralarındaki AWGN kanalı istenen gürültü varyansı ile bir gürültü katmanı olarak temsil edilir. Bu şekilde, haberleşme sistemi, $M = 1, 2, \dots, M$ olası mesajlardan biri olan s 'yi öğrenerek haberleşme kanalına karşı dirençli bir şekilde iletilen sinyalin x 'in temsilini oluşturmaya çalışan bir otokodlayıcı olarak görülebilir. Alıcıda, alınan sinyalden öğrenerek, orijinal mesaj s mümkün olan en düşük hata ile $\hat{s}$ olarak yeniden oluşturulur. [2]'deki çalışmada kodlayıcının girişi ve kod çözücünün gerçek çıktısı, k bitin 2^k boyutunda bir tek birimli vektörüdür. Bu nedenle, kod çözücü son katmanında Softmax aktivasyon fonksiyonunu kullanır ve sembol bazlı otokodlayıcı modeli kategorik çapraz entropi kaybını en aza indirerek optimize edilir.

[39] çalışmasında ise, kodlayıcının girişi ve kod çözücünün gerçek çıktısı k bit olarak kabul edilir. Bu nedenle, kod çözücü son katmanında sigmoid aktivasyon fonksiyonunu kullanır ve bit bazlı otokodlayıcı modeli, ikili çapraz entropi kaybını en aza indirerek optimize edilir.



Şekil 3.2 : Otokodlayıcı olarak modellenmiş bir haberleşme sistemi

Kablosuz haberleşme sistemleri için tasarlanan otokodlayıcılarda şekil 3.2'de de belirtildiği gibi genellikle ilk katman bir tek birimli (one-hot) katmandır. Bu katman, Q bitlik $s \in \{0, 1\}^Q$ vektörünü, yalnızca s 'nin ondalık gösterimine karşılık gelen konumda bir tane olmak üzere sadece sıfırlar içeren 2^Q boyutunda bir vektöre

dönüştürür. Örneğin, eğer $Q = 2$ ise, $[0, 0]^\top [1, 0, 0, 0]^\top$ 'a dönüştür, $[0, 1]^\top [0, 1, 0, 0]^\top$ 'a dönüştür, $[1, 0]^\top [0, 0, 1, 0]^\top$ 'a dönüştür ve $[1, 1]^\top [0, 0, 0, 1]^\top$ 'e dönüştür. Bir sonraki katman, normalizasyon katmanıdır:

$$f^{(1)}(\mathbf{a}) = \frac{\sqrt{2^Q}}{\|\Theta\|_F} \Theta \mathbf{a} = \mathbf{W} \mathbf{a} \quad (3.11)$$

burada $\Theta \in \mathbb{R}^{2 \times 2^Q}$, eğitilebilir katsayıların bir matrisidir ve $\mathbf{W} \in \mathbb{R}^{2 \times 2^Q}$, sütun vektörlerinin ortalama enerjisini bir birime sahip olacak şekilde normalize edilir. Son olarak, $\mathbb{R}2\mathbb{C}$ katmanları, 2 gerçekteki sayıyı tek bir karmaşık sayıya dönüştürür. Tek birimlik vektör kodlaması, normalizasyon ve $\mathbb{R}2\mathbb{C}$ katmanlarının birleşimi, birim ortalama enerjiye sahip olan sembol x 'i üretir, yani $\mathbb{E} [|x|_2^2] = 1$ olur.

AWGN katmanı, kanal katmanı olarak ifade edilir:

$$y = x + n \quad (3.12)$$

Burada, $n \sim \mathcal{CN}(0, \sigma^2)$, varyansı σ^2 olan bir karmaşık AWGN kanalıdır. Bu tür bir kanal türevlenebilir olduğundan, gradyanın alıcıdan vericiye geri yayılmasına izin verir. İlk alıcı katmanı, bir karmaşık sayıyı iki gerçekteki sayıya dönüştüren bir $\mathbb{C}2\mathbb{R}$ katmanıdır. Ardından, kod çözücü bloğunda eğitilebilir parametrelere sahip çoklu yoğun katmanlar eğitilir. Son katman boyutu 2^Q olup 2.10 denkleminde tanımlanan sigmoid aktivasyon fonksiyonunu kullanır. Sigmoid fonksiyonunun çıktıları $[0, 1]$ aralığındadır ve bu nedenle bir olasılık vektörü $\hat{\mathbf{p}} = [\hat{p}_0, \dots, \hat{p}_{Q-1}]^\top$ olarak yorumlanabilir, burada her bir giriş $\hat{p}_q$, y verildiğinde bir q . bitinin bir eşitlik olma olasılığını tahmin eder:

$$\hat{p}_q = \hat{P}(b_q = 1 | y) \quad (3.13)$$

Uçtan uca haberleşme sistemlerinin optimizasyonu için otokodlayıcı tasarımlarının örnekleri Tablo 3.2'de verilmiştir.

Çizelge 3.2 : Literatürde dikkate alınan farklı türde otokodlayıcılar

REFERANS	SENARYO	KATMAN	Kodlanmış/Kodlanmamış Sistem
[2]	Tek kullanıcı - 2 kullanıcı (MIMO), sembol tabanlı otokodlayıcı, AWGN, Rayleigh sönmüleme	Yoğun Katman	Hem kodlanmış (4/7) hem kodlanmamış sistem
[39]	Tek kullanıcı, sembol ve bit tabanlı otokodlayıcı, AWGN, Rayleigh sönmüleme	Yoğun Katman	Kodlanmamış Sistem
[40]	Tek kullanıcı, sembol tabanlı otokodlayıcı	Yoğun Katman	Kodlanmamış Sistem
[19]	Tek kullanıcı, sembol tabanlı otokodlayıcı, AWGN, Rayleigh sönmüleme	Evrişimli Katman	Hem kodlanmış hem kodlanmamış sistem
[41]	Tek kullanıcı, sembol tabanlı otokodlayıcı, AWGN	Yoğun Katman	Kodlanmış sistem (4/7)
[42]	Tek kullanıcı - 2 kullanıcı (MIMO), sembol tabanlı otokodlayıcı, AWGN	Yoğun Katman	Kodlanmamış Sistem
[43]	Tek kullanıcı, sembol tabanlı otokodlayıcı, AWGN	Yoğun Katman, Evrişimli Katman	Kodlanmış sistem (4/7)
[44]	Tek kullanıcı, bit tabanlı otokodlayıcı, AWGN	Yoğun Katman	Kodlanmamış Sistem
[45]	Tek kullanıcı, sembol tabanlı otokodlayıcı, AWGN	Yoğun Katman, Evrişimli Katman	Kodlanmamış Sistem

Otokodlayıcılar, derin öğrenmedeki denetimsiz öğrenme için en yaygın yöntemdir. Derin sinir ağları, eğitim amaçları için etiketli veri gerektirir ama etiketler denetimsiz öğrenmede sağlanmaz. Otokodlayıcı, her verinin kendi tarafından etiketlendiğini varsayar. Bir otokodlayıcı modelinin eğitim süreci tamamlandığında, otokodlayıcının çıktısı girdinin yakın bir tahminidir (ideal olarak girdi ile aynıdır). Otokodlayıcılar veriyi daha düşük bir boyutta sıkıştırırlar ve bu şekilde tasarlanırken, nöron sayısı ilk katmandan orta katmana doğru giderek azalır ve orta katmanda minimuma ulaşır, ortadan son katmana doğru giderek artar. Otokodlayıcıların kablosuz haberleşme sistemlerinde kullanılmasının bir avantajı da girdiyi yeterli doğrulukla çıktıda yeniden oluşturması ve orta katmanın verinin sıkıştırılmış bir versiyonunu içermesidir. Otokodlayıcılar, kablosuz iletişimde birçok sıkıştırma sorununa başarıyla uygulanmıştır.

Tablo 3.2’de detayları verilen çalışmalara göre, birçok çalışma, sadece "sembol tabanlı otokodlayıcı" modelini kullanarak otokodlayıcı tabanlı farklı blok kodlu modülasyon tasarımlarına odaklanmıştır: AWGN kanallarında: $R = 2/7, 4/7, 6/7$ hızları için [2], [43], [46]; Rayleigh sönmüleme kanallarında: $R = 4/8$ hızı için [12], $R = 4/7$ hızı için [47] çalışmaları literatürde mevcuttur. Otokodlayıcılar bu blok kodlanmış modülasyon tabanlı sistemleri elde etmek için alternatif sağlasa da, düşük blok uzunluklarında çalışmaları gibi konulardan dolayı geliştirilmesi gerekmektedir. Yani, bu çalışmalar düşük hızlı iletimi ele almış ve otokodlayıcıların BPSK, QPSK modülasyon-demodülasyon ve özellikle (7, 4) Hamming kodunu temel kanal kodlama tekniği olarak kullanarak önemli hata performans kazanımları göstermiştir. Bu nedenle, düşük bellek ve uygulama karmaşıklığı gerektiren kısa blok uzunluklarına sahip sistemler için yüksek hız gereksinimlerine yönelik (farklı) blok kodlu modülasyon tasarımları için otokodlayıcı tabanlı sistemlerin geliştirilmesi önem kazanmaktadır.

3.2.1 Uçtan uça haberleşme için otokodlayıcı uygulamaları

AWGN (Additive White Gaussian Noise, Toplamsal Beyaz Gauss Gürültüsü) kanalında temel otokodlayıcı, kablosuz haberleşme sistemlerinde kullanılan bir

haberleşme yöntemidir. Bu otokodlayıcılar, veri iletimini optimize etmek ve hata oranını düşürmek amacıyla kullanılır.

Otokodlayıcılar, haberleşme sistemindeki verileri kodlama ve kod çözme işlemleriyle işlerler. Gönderici (verici) tarafında veriler, özel bir kodlama işleminden geçirilerek kodlanır ve kanala iletmek üzere hazırlanır. Alıcı tarafında ise iletilen veriler, otokodlayıcı modeline uygun şekilde çözülerek orijinal verilere geri dönüştürülür.

AWGN kanalında otokodlayıcı, haberleşme sırasında ortaya çıkan gürültünün etkisini en aza indirmek için çeşitli yöntemler kullanır. Bu sayede haberleşme performansı artar ve hata oranı düşer. Ayrıca, otokodlayıcılar iletişim sisteminin güvenilirliğini artırmak için hata düzeltme kodlarıyla birlikte kullanılabilir.

3.2.1.1 QPSK - Hamming(7,4) ile otokodlayıcı(7,4) performans karşılaştırması

Bu başlık altında, AWGN kanalda temel otokodlayıcının yapısı, eğitim süreci ve performansı incelenmektedir. Kanal kodlama için modellenen otokodlayıcılar genelde (n, k) gösterimiyle belirtilir. Bir haberleşme sistemi, n kanal kullanımı üzerinden iletilen k bilgi bitinden oluşan bir mesajı iletmeyi amaçlar. Mesaj kümesinin boyutu $M = 2^k$ ve haberleşme hızı $R = k/n$ [bit/kanal kullanımı] olarak belirlenir. Bu haberleşme sistemi için otokodlayıcı modelinin katmanları ve karşılık gelen otokodlayıcı blokları şekil 3.3'te verilmiştir. Sinyal x , iletilen mesaj s için kanal üzerinden iletilir ve alıcı, alınan sinyale dayanarak $\hat{s}$ mesajını tespit eder.

Otokodlayıcının girişi, iletilen mesaj s 'in vektör temsilidir. Ardından, $ReLU$ ve aktivasyon fonksiyonları uygulanan yoğun katmanlardan geçirilir. Bu başlık altında [2] çalışması referans alınmıştır. Model oluşturulurken uygulanan hiper parametre değerleri verilmiştir. Referans alınan çalışmada BPSK modülasyon türüne ait BLER analizi incelenirken tezin bu aşamasında QPSK modülasyon türüne ait analizler yapılmıştır. Yine bu çalışma kapsamında tek derin katman yerine farklı sayıda katmanın kullanılmasıyla eğitim sürecinde optimal çözüme ulaşılmıştır. Normalizasyon katmanı, bir genlik kısıtlaması $|x_i| \leq 1 \forall i$, bir enerji kısıtlaması $\|\mathbf{x}\|_2^2 \leq n$ veya bir ortalama güç kısıtlaması $\mathbb{E}[|x_i|^2] \leq 1 \forall i$ uygulayarak verici güç sınırlamasını sağlar. Gürültü katmanı, varyansı $\sigma^2 = (2RE_b/N_0)^{-1}$ olan Gauss

Tek Birimli Vektör	Kodlayıcı
Tam Bağlı Katman 1	
ReLU	
Tam Bağlı Katman 2	
Normalizasyon	
AWGN	
Tam Bağlı Katman 3	Kod Çözücü
ReLU	
Tam Bağlı Katman 4	
Softmax	

Şekil 3.3 : Otokodlayıcı modeli için kullanılan katman konfigürasyonu

gürültüsünü ekler. Alıcı, *ReLU* ve softmax aktivasyon fonksiyonları uygulayan katmanlardan oluşur. Otokodlayıcının çıktısı, $\mathbf{p} \in (0, 1)^M$ aralığında olup en yüksek olasılığa sahip olan alınan mesaj $\hat{s}$ 'tir. Otokodlayıcının performansı, $\Pr(\hat{s} \neq s)$ olarak ölçülen BLER metriği ile değerlendirilir.

Bu çalışmada kullanılan bazı derin öğrenme parametreleri şu şekildedir :

Maksimum Epoch (tur sayısı) : 100

Mesaj Sayısı : 1000000

Yığın boyutu : 2000

Eğitimin gerçekleştirildiği E_b/N_0 değeri : 7

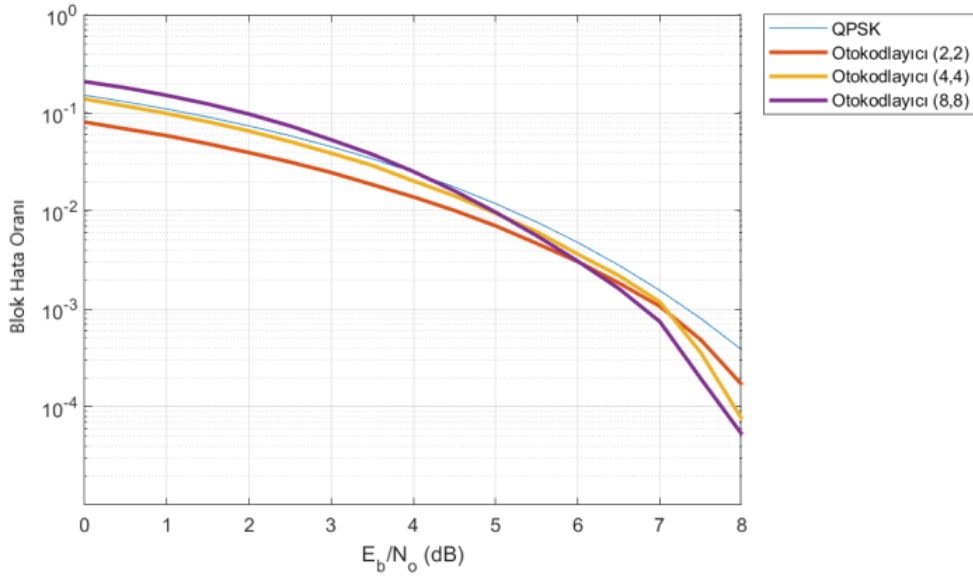
Optimizasyon algoritması : ADAM

Öğrenme oranı : Değişken (başlangıç : 0.01)

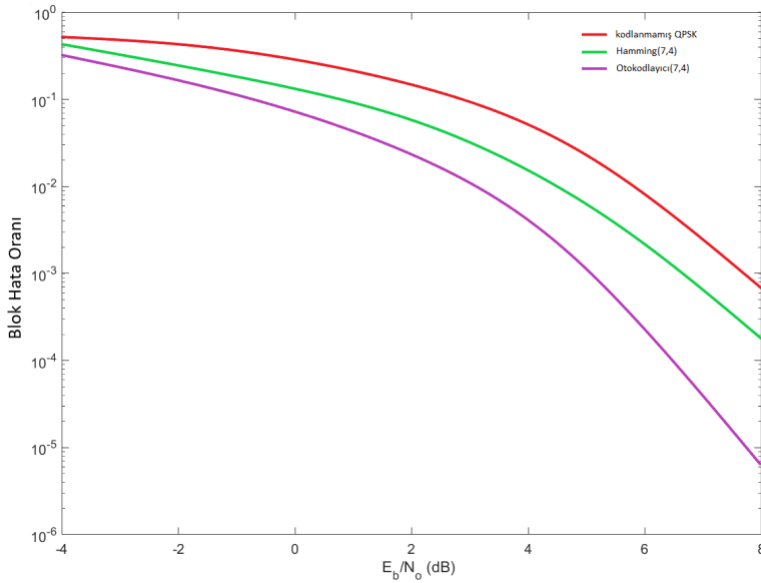
Kayıp Fonksiyonu : Kategorik Çapraz Entropi

Şekil 3.4 ve 3.5, kodlanmamış sistem ve kodlanmış sistemin otokodlayıcı ile aynı haberleşme hızına sahip geleneksel bir haberleşme sisteminin blok hata oranı olarak ifade edilen performans metriği $\Pr(\hat{s} \neq s)$ olan hata performansını göstermektedir. Geleneksel sistem, Hamming kodu ile QPSK modülasyonu kullanır. Burada Hamming kodlama için sert kararlı kod çözme tekniği uygulanmıştır. Ayrıca, kodlanmamış

sistem için (n, k) notasyonu $(2, 2)$, $(4, 4)$, $(8, 8)$ olarak ifade edilmiş ve performans karşılaştırması sunulmuştur. Sonuç olarak, otokodlayıcı, Hamming kodlu QPSK kadar güvenilir bir performans sergilemektedir. BLER performansı, iletimin güvenilirliğini değerlendirir. Bununla birlikte, verimliliğin sistemin gecikme süresi ve karmaşıklığı açısından da değerlendirilmesi gerekir. Yapay sinir ağlarının paralel işlem gücüne göre, otokodlayıcının gecikme ve karmaşıklık açısından üstün olması beklenir. Bu konular bu tez kapsamında incelenmemiştir ve gelecekteki araştırmalar için önerilebilir.



Şekil 3.4 : Kodlanmamış QPSK ve Otokodlayıcı Blok Hata Oranı

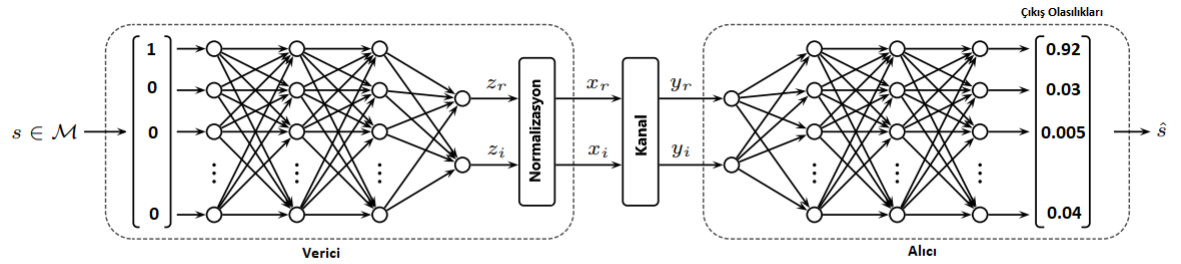


Şekil 3.5 : Hamming(7,4) ve otokodlayıcı blok hata oranı karşılaştırması

Grafikte dikey eksenle belirtilen blok hata oranı, iletilen veri bloklarının tamamının hatalı şekilde alınma olasılığını ifade eder. Her veri bloğu, iletişim sisteminden bağımsız olarak bir bütün olarak ele alınır. Eğer en az bir bit hatalı olarak alınırsa, blok hata olarak kabul edilir. Bit hata oranı, iletilen her bir bitin doğruluğunu belirlerken, blok hata oranı bir veri bloğunun tamamının hatalı alınma olasılığını gösterir. Dolayısıyla, blok hata oranı daha geniş bir perspektif sağlar ve sistem performansını blok bazında değerlendirirken, bit hata oranı her bir bitin performansını değerlendirir.

3.2.1.2 64-QAM ve otokodlayıcı performans karşılaştırması

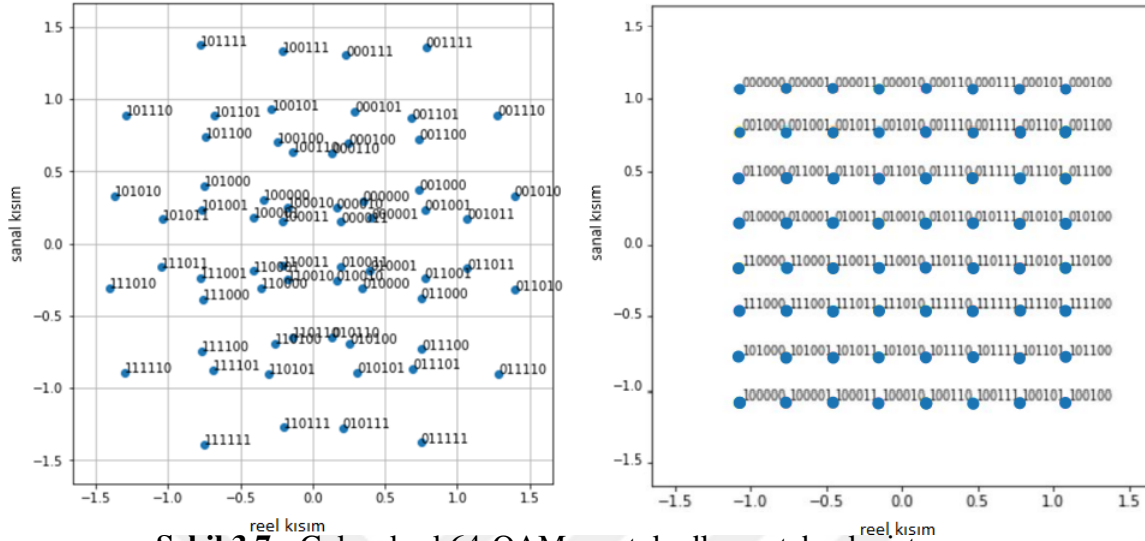
Otokodlayıcı tabanlı sistemde kodlayıcı çıkışı ve kod çözücü girişi arasındaki değerler kompleks sayılardır. Otokodlayıcı yapısında blok olarak bu durum şekil 3.6’te gösterilmiştir.



Şekil 3.6 : Hem verici hem de alıcı sinir ağı ile gizli katmanlı otokodlayıcı yapısı

Otokodlayıcılar farklı mesajlar ve kanal gürültü yapılandırmaları üzerinde birçok eğitim yığını kullanılarak eğitilir. Özellikle, hem verici hem de alıcıdaki tüm nöronların ağırlıkları ve sapmaları, denklem 2.24 ile verilen çapraz entropi kaybı ile ilişkili olarak optimize edilir. Optimizasyon, uygun bir öğrenme oranıyla rastgele gradyan inişi varyasyonu kullanılarak gerçekleştirilir. Eğitimden sonra, öğrenilen takımyıldızı değeri normalizasyon katmanının çıktısından elde edilebilir. Şekil 3.6’de de görüldüğü gibi x iki boyutludur ve x_r ve x_i , iletilen takımyıldızı noktalarının gerçek ve sanal kısımları olarak ifade edilir. Otokodlayıcının öğrenilen alıcı kısmı, gözlenen sinyalden (y) $\hat{s}$ mesajını çözebilen bir sınıflandırıcıdır.

64-QAM, 6 bitlik bir sembolü temsil eden bir modülasyon şemasıdır. Her sembol, 64 farklı faz ve genlik kombinasyonu ile ilişkilendirilir. Bu nedenle, 64-QAM sistemi, geniş bir sembol çeşitliliği sunar ve daha yüksek veri iletim hızlarına olanak sağlar.



Şekil 3.7 : Geleneksel 64-QAM ve otokodlayıcı tabanlı sistem

Öte yandan, otokodlayıcı tabanlı 6 bitlik sembol temsil eden sistemlerde, otokodlayıcılar kullanılarak hata düzeltme yetenekleri sağlanır. Bu nedenle, otokodlayıcı tabanlı 64-QAM sistemlerinde, hata düzeltme yetenekleri sağlanarak güvenilir bir otokodlayıcı tabanlı haberleşme sistemi elde edilebilir. Otokodlayıcılar ile eğitilen bir sistem için öğrenilen takımyıldızı Şekil 3.7’te gösterilmiştir ve 64-QAM modülasyonu ile karşılaştırılmıştır. Görüldüğü üzere semboller ayrıştırılabilmektedir. Bu da düşük hata oranı sağlar.

4. KABLOSUZ HABERLEŞME İÇİN UÇTAN UCA OTOKODLAYICI TASARIMI

Çoklu Giriş Çoklu Çıkış (MIMO) haberleşme sisteminin uygulanması, özellikle sönmleme etkisi nedeniyle kablosuz bir kanalda güvenilir iletişim için kullanılan sistemlerdendir. Birden çok anten teknolojisi, iletişim sistemlerinde girişim azaltma ve önleme, uzaysal çeşitlilik gibi faydalar sağlar. MIMO'nun temel fikri, vericiden ve alıcıdan örneklenen sinyallerin uzaysal alanda belirli bir yöntemle birleştirildiği veya iletişimin Bit Hata Oranı (BER) açısından kalitesini artırmak için veri çeşitliliği eklediği bir yöntemdir.

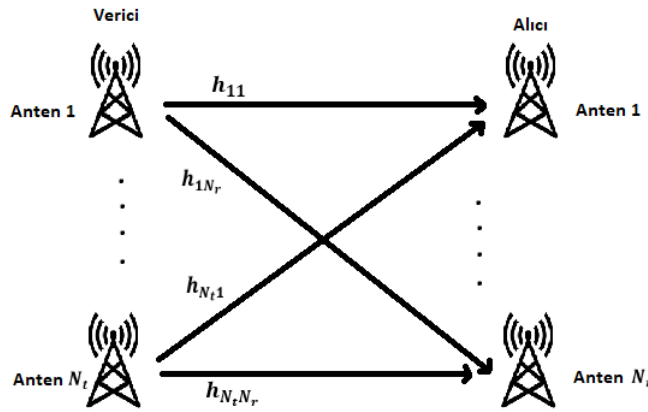
Çoklu anten sistemlerindeki otokodlayıcı teknikleri çoğunlukla kapalı döngü otokodlayıcılar (closed loop autoencoders) ve açık döngü otokodlayıcılar (open loop autoencoders) olarak iki kategoriye ayrılır. Kapalı döngü otokodlayıcı sistemlerinde, kodlayıcı ve kod çözücü kanal durumu bilgisine (Channel State Information, CSI) sahiptir ve kodlayıcı iletim sembolünü ve kanal durum bilgisini aynı anda kodlar [13]. Öte yandan, açık döngü otokodlayıcı sistemleri sadece kod çözücünün kanal durum bilgisini tahmin etmesine izin verir. Vericinin bu bilgiye sahip olmaksızın [45] otokodlayıcı tabanlı haberleşme sistemleri, iletim sembolünü veya bitlerini kullanarak kodlama ve çözme işlemini gerçekleştirir.

Bu bölümde, uzaysal çeşitlilik ve uzaysal çoğullama sistemlerinin verici ve alıcı işlemleri tek bir uçtan uca kodlama yöntemi ile ele alınmıştır. Belirli kanal koşulları için bir hata oranını en aza indirmek için verici ve alıcı ortaklaşa optimize edilir ve otokodlayıcılar kullanarak geleneksel gönderici anten çeşitleme sistemlerinin performansına ulaşılması hedeflenmiştir. Eğitim sırasında Rayleigh sönmleme kanalı kullanılmıştır. Sonuçlar, geliştirilen otokodlayıcı tabanlı sistemin BER performansını gösterir. Göndericide çeşitleme uygulamalarının performansı ile karşılaştırıldığında geleneksel sistemler için otokodlayıcı kullanımı benzer sonuçlar vermiştir. Bu bölümün devamında uzaysal çoğullama ve uzaysal çeşitlilik konuları incelenmiştir ve

bu sistemlerin otokodlayıcı modellerinin genel katman yapıları şekil 4.2 ve 4.6'daki gibidir. Bu modeller geleneksel çoğullama ve çeşitleme bloklarına benzer yapıdadır. Bu bölümün devamında çeşitleme ve çoğullama için öncelikle genel ifadeler belirtilmiş ve simülasyon sonuçları bölümlerinde çalışma kapsamında ele alınan özel durumlar incelenmiştir.

4.1 Uzaysal Çoğullama İçin Otokodlayıcı Modeli

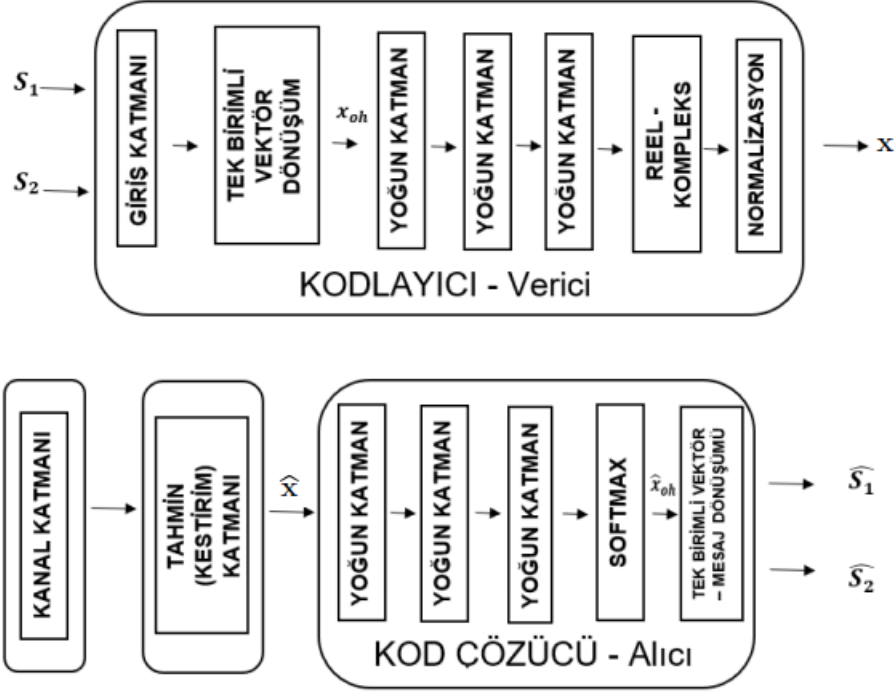
Uzaysal Çoğullama (Spatial Multiplexing), MIMO haberleşme sistemlerinde veri iletim hızını artırmak için kullanılan bir tekniktir. Bu teknik, birden fazla antenin kullanıldığı bir iletişim ortamında veri iletimini iyileştirmek için tasarlanmıştır. Uzaysal çoğullama, birden fazla veri akışının aynı anda ve aynı frekansta gönderilmesini sağlar.



Şekil 4.1 : N_t verici N_r alıcı MIMO sistem gösterimi

MIMO haberleşme sistemi N_t verici anteni ve N_r alıcı anteninden oluşur. Gönderilmek istenen veri uzaysal çoğullamada gönderici antenlerden aynı anda iletilir. Alıcı kısımda kanal bozulmalarına uğrayan sinyal kanal dengeleme, kestirim işlemleri sonrasında yeniden elde edilir. Çalışmanın bu kısmında kestirim tekniği olarak minimum ortalama kare hatası (minimum mean square error, MMSE) kullanılmıştır. Böyle bir sistem için yoğun katmanlar, normalizasyon katmanı, kanal ve kestirim katmanlarından oluşan otokodlayıcı modeli şekil 4.2'teki gibidir.

Otokodlayıcı modelin geleneksel sistemlerdeki bloklara karşılık gelen kodlayıcı (verici), kod çözücü (alıcı) bölümleri bulunur. Otokodlayıcının kodlayıcı çıkışı



Şekil 4.2 : Uzaysal Çoğullama için Otokodlayıcı modeli

modüle edilmiş sinyaller olarak düşünülebilir. Modeldeki x (şekil 4.2) otokodlayıcının kodlayıcı çıkışındaki sembollerdir. Uzaysal çoğullama kapsamında bakıldığında x sembolleri, N_t anten aracılığıyla tek bir zamanda iletilir. Rayleigh sönümleme etkisinin bulunduğu kanalda (H) sinyal bozulmalara uğrar ve alıcı kısımda da sinyale AWGN eklenir. Genel olarak bu sistemin ifadesi aşağıdaki gibidir.

$$y = Hx + n \quad (4.1)$$

y , $N_r \times 1$ boyutlu alınan sinyali; H , $N_r \times N_t$ boyutlu kanal kazancını ve n , $N_r \times 1$ boyutlu beyaz Gauss gürültüsünü ifade eder. N_t verici antenin ve N_r alıcı antenin bulunduğu bir sistemde bu antenler arasında sinyal iletim yolu sayısı $N_t \times N_r$ çarpımıdır. Bu sayıda farklı yoldan bilgi sembolleri alıcıya ulaşır. Dolayısıyla bu da kanal matrisinin boyutlarını belirler. Alıcıda, vericiye geri bildirim olmadan kanal bilgisinin (perfect channel knowledge) olduğu, yani açık döngülü (open loop) bir uzaysal çoğullama sistemi varsayılır. Alıcıya gelen sinyale MMSE kestirimi uygulanır.

$$\hat{x} = H^H \left(HH^H + \frac{1}{SNR} I_{N_r} \right)^{-1} y \quad (4.2)$$

$\hat{x}$ tahmin edilen sembol vektörünü ($N_t \times 1$), H^H hermitian işlevini (kompleks transpoz), y alıcıya gelen sinyal vektörünü, $\mathbf{I}_{N_r}$ boyutu $N_r \times N_r$ olan birim matrisi ifade eder. Tahmin edilen semboller geleneksel sistemlerde demodülasyon işlemiyle bilgi bitlerine dönüştürülürken otokodlayıcı modelde tahmin edilen semboller otokodlayıcının kod çözücü bloğunu besler. Yoğun katmanlar ve çıkış katmanı ile birlikte girişte otokodlayıcı modele verilen bilgi bitleri yeniden oluşturulur.

4.1.1 Uzaysal çoğullama simülasyon sonuçları

Uzaysal Çoğullama için $N_t \times N_r$ gösterimiyle 2x1, 2x2 ve 2x3 sistemler incelenmiştir. 3 sistem de şekil 4.2’de verilen modele benzer şekilde modellenmiştir. Otokodlayıcı modellerin geleneksel MIMO-MMSE sistemler için BER performansı karşılaştırması yapılmıştır.

2 verici anten ve 2 alıcı antenin bulunduğu bir 2x2 otokodlayıcı sistemi için denklem 4.1 aşağıdaki formdadır.

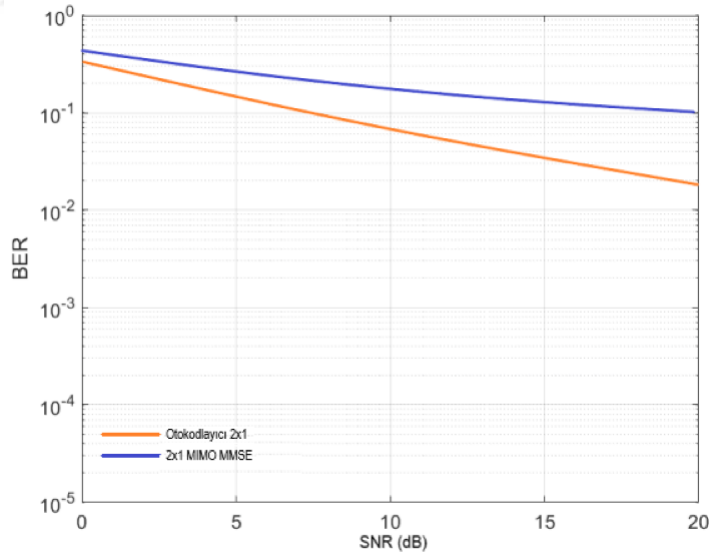
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (4.3)$$

Alıcıya gelen sinyal y bir 2x2 otokodlayıcı modeli için boyutları 2x1 olan bir vektördür. Kanal matrisi alıcı verici arasındaki farklı iletim yollarındaki bozulmaları temsil eder ve boyutu 2x2’dir. Otokodlayıcı modeldeki kanal katmanı Rayleigh sönümleme etkisini modeller. Rayleigh kanalında reel ve sanal kısımları Gauss dağılımlı olan kanal katsayıları üretilir. Alınan sinyal vektörü y , tahmin (kestirim) katmanını besler ve çıkış kod çözücü bloğuna iletilir. Kod çözücü çıktısı tahmin edilen değerleri giriş değerleri ile karşılaştırır ve kayıp değeri hesaplar. Kayıp fonksiyonu olarak kategorik çapraz entropi kaybı kullanılmıştır. Modelin eğitimi boyunca yığın boyutu 1024 ve 2048 olarak belirlenmiştir.

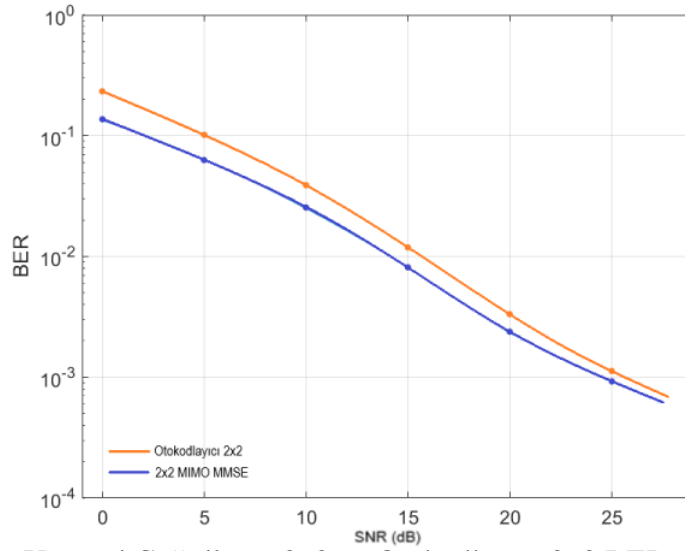
Öğrenme aktarması (transfer learning), bir yapay sinir ağı modelinin başka bir görevde öğrendiği bilgileri veya yetenekleri kullanarak yeni bir görevi çözmek için uyarlanması veya yeniden eğitilmesi anlamına gelir. Bu yaklaşım, bir modelin bir görevde öğrendiği özellikleri veya temsilleri başka bir görevde kullanarak, daha az veriyle veya daha kısa sürede sonuçlar elde etmeye yardımcı olur. Öğrenme aktarmasının

temel fikri, bir modelin bir görevde öğrendiği genel bilgi veya özelliklerin, farklı bir görevde de işe yarayabileceğidir. Öğrenme aktarması sayesinde, önceden belirlenen bir başlangıç SNR değerinde eğitilmiş bir modelin öğrendiği temsilleri yeni bir görevde başlangıç noktası olarak kullanılır. Öğrenme aktarması çalışmasına benzer olarak otokodlayıcı 5-20 dB alt ve üst limit arasında farklı değerlerde eğitilmiştir. Bu bakımdan öğrenme aktarmasına benzer yapıdadır. Eğitilen model genel olarak 100000 ve değişen sayıda test veri seti üzerinde test edilmiştir. Test verisi bilgisayar ortamında sentetik olarak üretilmiş giriş bilgi bit dizisidir.

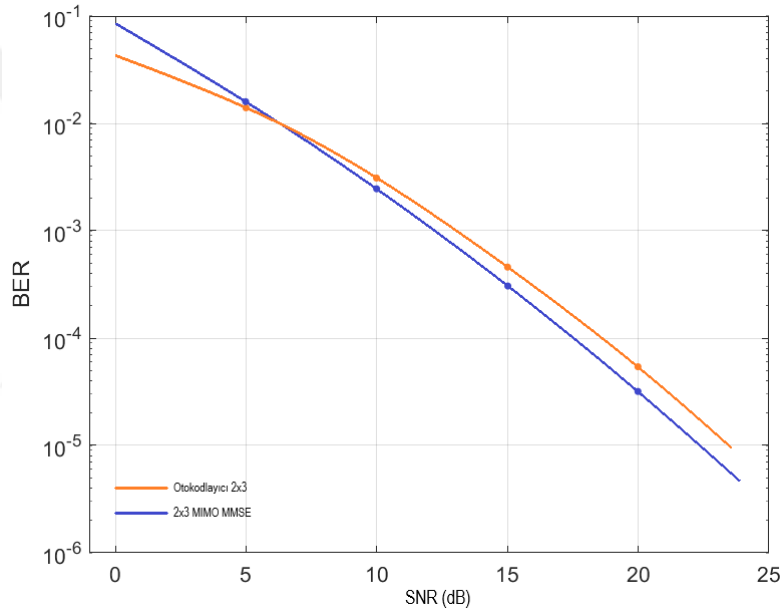
Otokodlayıcı ile eğitilen model ve geleneksel 2x1, 2x2, 2x3 MIMO sistemlerin BER performansı karşılaştırmaları denklem 4.1 ve 4.2 bağlı olarak verilmiştir. İlgili performans karşılaştırması şekil 4.3, 4.4 ve 4.5'te gösterilmiştir. Otokodlayıcı model geleneksel sistem ile performans karşılaştırmasında 2x1 sistemler için aynı SNR değerlerinde daha düşük BER değerlerinde sonuçlara ulaşılmış, 2x2 ve 2x3 sistemler için benzer sonuçlar vermiştir.



Şekil 4.3 : Uzaysal Çoğullama 2x1 ve Otokodlayıcı 2x1 BER performansı



Şekil 4.4 : Uzaysal Çoğullama 2x2 ve Otokodlayıcı 2x2 BER performansı

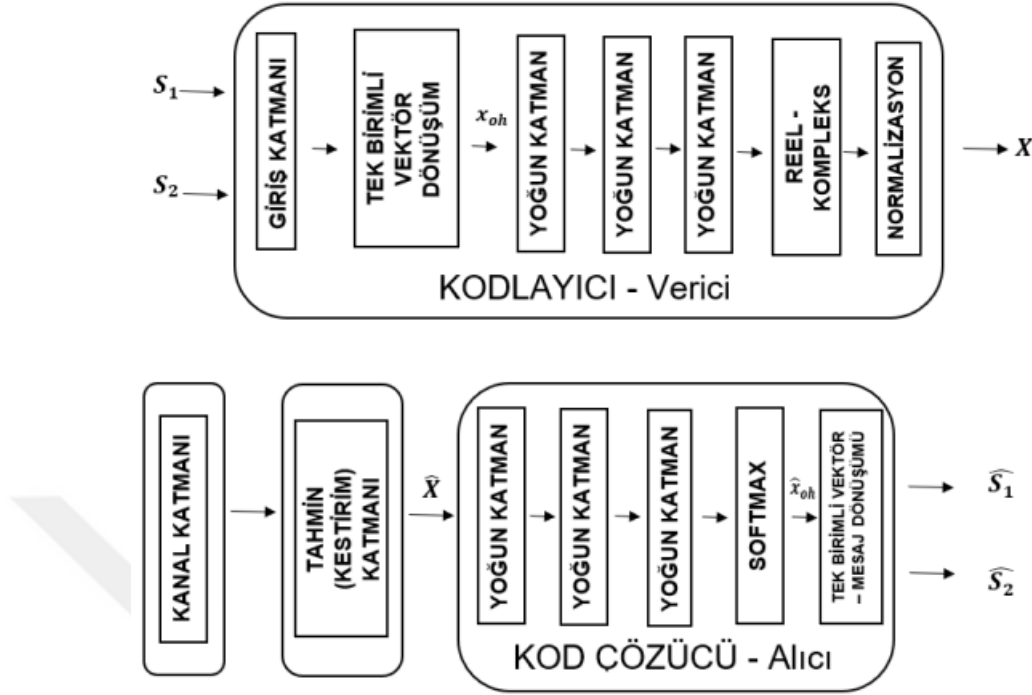


Şekil 4.5 : Uzaysal Çoğullama 3x2 ve Otokodlayıcı 2x3 BER performansı

4.2 Uzaysal Çeşitlilik İçin Otokodlayıcı Modeli

Sistem, N_t verici anten ve N_r alıcı anten olarak ele alınmış, kanal etkileri iletilen mesajlar üzerine eklenerek sistem modellenmiştir. Otokodlayıcı birkaç yoğun katman otokodlayıcının kodlayıcı bölümü, herhangi bir öğrenme parametresi bulunmayan Lambda katmanı olarak adlandırılan kanalın bozucu etkilerinin modellendiği katman ve yine yoğun katmanlardan oluşan otokodlayıcının kod çözücü bölümü bulunur.

Lambda katmanında (kanal katmanında) sembollere Rayleigh sönümlerme etkileri eklenir.



Şekil 4.6 : Uzaysal Çeşitleme için Otokodlayıcı modeli

Uzaysal çeşitlilik için oluşturulan otokodlayıcı modeli şekil 4.6’da görüldüğü gibi sinir ağı ile modellenir. x_{oh} tek birimli (one-hot) vektör gösterimidir. $N_t \times N_r$ otokodlayıcı sisteminde, kodlayıcı kısımda, s mesajlar, N_t anten aracılığıyla iletilir. Kodlayıcı, N_t mesaj akışını $s = [s_1, s_2, \dots, s_{N_t}]^T$ olarak alır ve bunları $N_t \times N_{cu}$ boyutlu uzay zaman kodu $\mathbf{X}$ ’e dönüştürür. N_{cu} uzay zaman kodunun zaman bileşenini ifade eder. N_t verici antenden oluşan otokodlayıcı modelde kodlayıcının normalizasyon katmanının işlevi, iletim gücünü $\mathbf{X}$ matrisinin her sütununu ayrı ayrı normalize etmektir. $\mathbf{X}$ ’in her sütunu x vektörü ile ifade edilirse, $\|x\|_2^2 \leq P$ olur. Alınan sinyal şu şekilde temsil edilir:

$$\mathbf{Y} = \sqrt{\frac{1}{N_t}} \mathbf{H} \mathbf{X} + \mathbf{N} \quad (4.4)$$

$\mathbf{X}$, boyutu $N_t \times N_{cu}$ olan sembolleri (kodlayıcının ürettiği uzay-zaman kodu) ifade eder. $\mathbf{H}$ boyutu $(N_r \times N_t)$ olan karmaşık elemanlara sahip kanal matrisidir. Burada $\mathbf{H} \mathbf{X}$, verici anten sayısına göre normalize etmek için $\sqrt{N_t}$ ’ye bölünür. $\mathbf{N}$ alıcının eklediği

$N_r \times N_{cu}$ boyutlu sıfır ortalamalı Gauss rastgele değişkenleri olarak modellemiş ve varyansları σ^2 olarak ifade edilir. $\mathbf{X}$ Reel-Kompleks dönüşüm katmanı sonrasında oluşturulur. Reel-Kompleks katmanına bir önceki yoğun katmandan $2 \times N_t \times N_{cu}$ çarpımı kadar düğüm çıktısı gelir. Diğer bir ifadeyle Reel-Kompleks katmanından önceki son katman çıkışının uzunluğu $2 \times N_t \times N_{cu}$ olan gerçek değerli bir vektördür. $N_t \times N_{cu}$ sayısı kadar eleman sembollerin gerçek bölümü temsil ederken, $N_t \times N_{cu}$ sayısı kadar eleman karmaşık sinyalin sanal bölümünü temsil eder. Sonrasında bu vektör $N_t \times N_{cu}$ boyutlu matrise dönüştürülür. Bu matrisin her elemanı otokodlayıcı sembollerini ifade eder.

Bu türden bir kestirim için, tahmin edilen sinyal şu şekilde hesaplanır:

$$\hat{\mathbf{X}} = \mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H + \frac{N_t}{\rho} \mathbf{I}_{N_r} \right)^{-1} \mathbf{Y}, \quad \rho = \frac{P}{\sigma^2} \quad (4.5)$$

burada ρ sinyal-gürültü oranını belirtir. Tahmin edilen değerler reel ve sanal kısımlarına ayrılarak otokodlayıcının kod çözücü katmanlarına bağlanır. Softmax katmanı öncesi yoğun katmanlar tarafından öğrenilen sinyal çıkışta giriş sinyallerine en yakın çıktıyı üretir.

4.2.1 Sistem optimizasyonu

Kategorik çapraz entropi kayıp fonksiyonu, ağ parametrelerini eğitimi için gradyan inişi ile optimizasyon için kullanılır. Bu durumda, kategorik çapraz entropi kayıp fonksiyonu (categorical cross entropy, CCE) L_{CCE} aşağıdaki gibi verilir:

$$L_{CCE}(\mathbf{x}_{oh}, \hat{\mathbf{x}}_{oh}) = -\frac{1}{B} \sum_{i=0}^B \sum_{j=1}^{2^{kN_t}} \mathbf{x}_{oh,j} \log(\hat{\mathbf{x}}_{oh,j}) \quad (4.6)$$

burada x_{oh} , tek birimli giriş vektörünü belirtir, $\hat{\mathbf{x}}_{oh}$ j tahmin çıktısı olasılığını temsil eder, B yığın büyüklüğüdür. Adam yöntemi, gradyan kaybı ve geri yayılım güncellenen parametrelere dayalı olarak ağ parametrelerinin kaybını hesaplamak için kullanılır. Adam algoritması ile, ağırlık güncellemeleri geriye doğru yayılım algoritmasını kullanarak kayıp gradyanına dayalı olarak hesaplanır. Bu durumda, sırasıyla bir ileri geçiş (forward pass), ve bir geriye doğru geçiş (backward pass),

hesapları yapılır. Otokodlayıcının eğitimi esnasında her bir iterasyonda tahmin edilen $\mathbf{x}_t$ değerlerinin otokodlayıcı (AE) fonksiyonu ile basit bir gösterimi:

$$\hat{\mathbf{x}}_{\text{oh}}^n = f_{\text{AE}}(\mathbf{x}_{\text{oh}}^n; \theta_{\text{AE}}^n) \quad (4.7)$$

şeklindedir. Burada $f_{\text{AE}}(\cdot)$ otokodlayıcının işlemlerini ve θ_{AE} eğitim parametrelerini, n iterasyon sayısını temsil eder. Kayıp değeri, $\delta\theta_n$, $L(\mathbf{x}_{\text{oh}}^n, \hat{\mathbf{x}}_{\text{oh}}^n)$ ifadesinin θ_{AE}^n değerine göre türevi alınarak hesaplanır. Bu ifade şu şekildedir :

$$\delta\theta_{\text{AE}}^n = -\alpha \frac{\partial L_n}{\partial \theta_{\text{AE}}^n} = -\alpha \frac{\partial L_{\text{CCE}}^n(\mathbf{x}_{\text{oh}}^n, \hat{\mathbf{x}}_{\text{oh}}^n)}{\partial \theta_{\text{AE}}^n} \quad (4.8)$$

Öğrenme oranı α ile gösterilir. Sistemin parametreleri $(n + 1)$ 'inci yinelemede güncellenir:

$$\theta_{\text{AE}}^{n+1} = \theta_{\text{AE}}^n + \delta\theta_{\text{AE}}^n \quad (4.9)$$

Hata minimum değere ulaştığında veya belirli sayıda yinelemeye ulaşıldığında eğitim sona erer. Otokodlayıcı sisteminin parametreleri daha sonra vericiden alıcıya aynı anda optimize edilir.

Geriye doğru yayılımında, kanal katmanında kanal matrisi ile çarpma işlemi hem ileriye hem de geriye doğru geçişlerin sonuçlarına etki eder. Sonuç olarak, sistem bu şekilde Rayleigh kanal sönümlemesi etkisi ile eğitilir.

4.2.2 Verici çeşitliliği simülasyon sonuçları

Uzay zaman kodlaması ile verici çeşitliliği 2×1 sistem için t ve $t + T$ olmak üzere 2 farklı zamanda sağlanır. Bu durumda $N_{cu} = 2$ 'dir.

2×1 otokodlayıcı model için kodlayıcı çıkışındaki $\mathbf{X}$ 'in elemanları 4 kodlayıcı sembolüdür. Dolayısıyla bu sistemde kodlayıcı çıkışı 2×2 boyutlu bir matris olacaktır. Bu matrisin ilk sütunu t anında gönderilen sembollerini temsil ederken 2.sütunu $t + T$ anında gönderilen sembollerini temsil eder. Alamouti sistemine benzer olarak

uzay-zaman kodlaması otokodlayıcı modelde kullanılmış olup farklı zamanlarda alıcıya gelen sinyal otokodlayıcı sisteminde aşağıdaki gibidir :

$$\begin{aligned} \mathbf{y}_1 = \mathbf{y}(t) &= \mathbf{h}_1 \mathbf{x}_{11} + \mathbf{h}_2 \mathbf{x}_{21} + \mathbf{n}_1 \\ \mathbf{y}_2 = \mathbf{y}(t + T) &= \mathbf{h}_1 \mathbf{x}_{12} + \mathbf{h}_2 \mathbf{x}_{22} + \mathbf{n}_2 \end{aligned} \quad (4.10)$$

Burada $\mathbf{y}(t)$ ve $\mathbf{y}(t + T)$, zaman t ve $t + T$ 'de alınan sinyalleri temsil ederken, $\mathbf{n}_1$ ve $\mathbf{n}_2$ alıcı gürültüsünü temsil eden karmaşık rastgele değişkenlerdir. Otokodlayıcı çıkışında elde edilen sembollerin (geleneksel haberleşme sistemlerindeki modüle edilmiş semboller olarak düşünülebilir) 2 verici antenden gönderim biçimleri ilk zaman diliminde, yani t anında gönderici antenlerden sırasıyla x_{11} ve x_{21} şeklindeyken, $t + T$ anında da x_{12} ve x_{22} 'dir. Çalışmada antenlerden gönderilen sembol dizisi kuralının diğer bir gösterimi tablo 4.1'de belirtildiği gibidir.

Çizelge 4.1 : 2 anten için sembol iletim sırası

	Gönderici Anten 1	Gönderici Anten 2
t zamanı	x_{11}	x_{21}
t + T zamanı	x_{12}	x_{22}

Alamouti kodlama, sembolleri iki farklı zamanda gönderme prensibine dayalı olduğundan kodlayıcı ilk zaman aralığında, t , her antenden iki farklı sembol gönderir. Sembollerin değiştirilmiş halleri $t + T$ zamanında gönderilir.

Otokodlayıcı sembolleri tablo 4.1 olduğu gibi Alamouti verici kuralına benzer ve Alamouti verici kuralından farklı kombinasyonlara sahip iki ardışık zaman dilimine eşleyerek eğitilir.

Alınan sinyal vektörü $\mathbf{y}$, minimum ortalama kare hatası (MMSE) kullanılarak algılanmıştır. Bu bölümde, 2x1 sistemin Bit Hata Oranı (BER) performansı, Rayleigh kanalı üzerinde değerlendirilir. Sonuçlar 2x1 Alamouti sistemine göre karşılaştırılır. Tüm sistem, Keras kütüphaneleri kullanılarak Python'da uygulanmıştır. Otokodlayıcı modelde toplam iletim gücü P olacak şekilde normalize edilmiştir. Sistem 5-20 E_b/N_0 değerleri arasında belirli aralıklarla artırılarak eğitilmiştir. Optimizasyon yöntemi olarak Adam optimizasyon algoritması kullanılarak, öğrenme hızı 0.01 ve 0.0001

değerleri arasında güncellenmiştir. Yığın boyutu 2048 ve 1000000 veri seti üzerinde model eğitilmiştir.

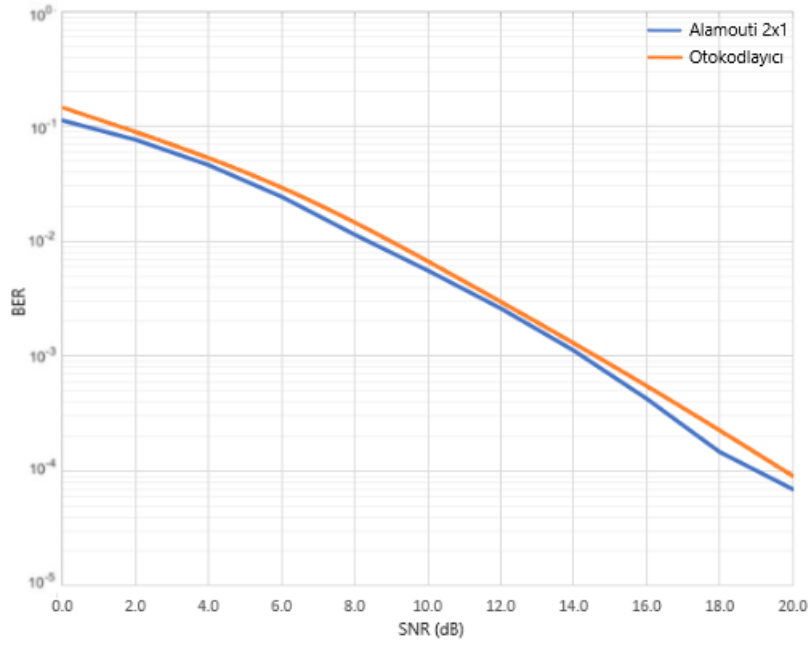
Şekil 4.7, hem Alamouti STBC kodlaması hem de otokodlayıcı sistemi için BER performans sonuçlarını göstermektedir. Sonuçlarda, Alamouti kodlamaya benzer performans elde ediyoruz. Eğrilerdeki eğimler benzer olduğundan otokodlayıcı model benzer verici çeşitlemesi performansı sunar. Bu da otokodlayıcı tabanlı sistemin verici çeşitlemesinde geleneksel Alamouti sistemine karşı bir alternatif sunar.

Otokodlayıcı modelin katmanları ve bu katmanlarda kullanılan çıkış boyutlarının örnek gösterimi tablo 4.2’de belirtildiği gibidir.

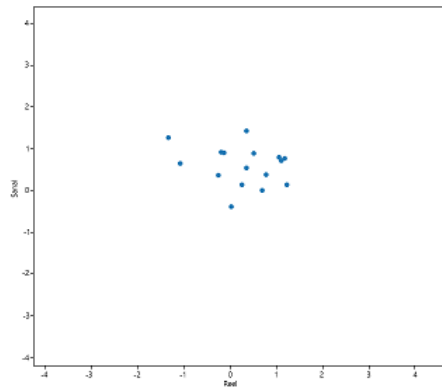
Çizelge 4.2 : 2x1 otokodlayıcı modeli katmanları ve çıkış boyutları

KATMANLAR	BOYUTLAR
Giriş Katmanı	Yığın Boyutu,2
Tek birimli vektör dönüşüm	Yığın Boyutu,4
Yoğun Katman	Yığın Boyutu,16
Yoğun Katman	Yığın Boyutu,32
Yoğun Katman	Yığın Boyutu,16
Kodlayıcı Çıkışı	Yığın Boyutu,8
Kod Çözücü Girişi	Yığın Boyutu,8
Yoğun Katman	Yığın Boyutu,8
Yoğun Katman	Yığın Boyutu,8
Yoğun Katman	Yığın Boyutu,4
Softmax	Yığın Boyutu,4

SNR aralığı boyunca otokodlayıcı tabanlı model, standart Alamouti performansına benzer performans göstererek umut verici sonuçlar göstermektedir. SNR yükseldikçe, önerilen model ile Alamouti kodlaması arasındaki performans farkı benzerdir. Ayrıca sistemin oluşturduğu takımyıldızı grafiği şekil 4.8’de verilmiştir.



Şekil 4.7 : 2x1 Otokodlayıcı ve Alamouti karşılaştırması



Şekil 4.8 : Otokodlayıcının ürettiği takımyıldızı (constellation) grafiği

5. SONUÇLAR

Bu çalışma, kablosuz iletişim sistemlerinde uçtan uca otokodlayıcı tasarımının önemi ve etkinliği üzerine bir araştırma sunmaktadır. Otokodlayıcılar, veri kodlama, özellik çıkarma ve hata düzeltme gibi kritik işlevleri gerçekleştirmek için derin öğrenme temelli bir yaklaşım sunmaktadır. Bu çalışmada, kablosuz haberleşme sistemleri ve otokodlayıcılar arasındaki ilişkiler çoğullama ve çeşitlilik açısından incelenmiştir.

Kablosuz iletişim sistemlerinde veri transferinde karşılaşılan hata ve gürültü problemleri, haberleşme performansını olumsuz etkileyebilir. Bu nedenle, hata düzeltme teknikleri büyük önem taşımaktadır. Bu çalışmada, 2 verici anten kullanılan sistemlerin otokodlayıcılar ile çeşitlilik kazancı elde edilmesi konusu incelenmiştir. Otokodlayıcılar veriyi sıkıştırma ve özellik çıkarma nitelikleriyle geleneksel haberleşme sistemlerinde uygulanabilir. Bu, kablosuz haberleşme sistemlerinde daha güvenilir ve hatadan daha az etkilenen veri iletimini sağlamak için etkili bir çözüm sunar.

Otokodlayıcıların eğitimi ve parametre ayarlaması optimizasyon algoritmaları aracılığıyla gerçekleştirilir. Gradyan inişi, Uyarlanabilir Moment Tahmini ve Karelerin Ortalamasının Karekökü Yayılımı gibi optimizasyon algoritmaları, otokodlayıcıların eğitiminde sıklıkla kullanılan yöntemlerdir. Bu algoritmalar, eğitim sürecinde modelin hedeflenen performansa ulaşmasını sağlamak için parametreleri günceller ve hata düzeltme sürecini optimize eder. Bu çalışmada Adam optimizasyon algoritması otokodlayıcının uçtan uca optimizasyonu için kullanılmıştır.

Otokodlayıcı tabanlı oluşturulan çeşitleme sisteminde MMSE kestirim uygulaması incelenmiştir. MMSE ile tahmin edilen bir değer ile gerçek değeri karşılaştırıldığında ortalama kare hatası (MSE) minimuma getirilerek bir otokodlayıcı modeli ele alınmıştır. MMSE tarafından tahmin edilen değerler gerçek değere yaklaştırılarak geleneksel sistemlere yakın performans sonuçları elde edilmiştir.

Sistem performansı ile hesaplama karmaşıklığı arasındaki bir dengenin oluşturulması, MIMO haberleşmesinin performansının geliştirilmesinde önemli bir yapıtaşı haline gelmiştir. Kablosuz haberleşme sistemlerinde uzaysal çeşitlilik elde etmek için önerilen otokodlayıcı modelinde MIMO haberleşme sistemlerinin uzaysal çeşitlilik sağlamak için kullanıldığına dair temel bir açıklama sunulmuştur. MIMO'nun temel fikri, vericiden ve alıcıdan gelen sinyallerin uzaysal alanda birleştirilmesi veya çeşitlilik eklenmesi yoluyla haberleşme kalitesini artırmaktır.

Bu çalışmada kullanılan verici çeşitliliği için otokodlayıcı modeli, 2 verici anten ve 1 alıcı anten içeren bir sistem için tasarlanmıştır. Bu sistemde, gönderici ve alıcı, haberleşme kanalının durumunu göz önünde bulundurarak uçtan uca optimize edilmiştir. Özellikle, otokodlayıcılar kullanılarak geleneksel gönderici anten çeşitleme sistemlerinin performansına benzer bir performans elde edilmiştir. Ayrıca, çalışmadaki otokodlayıcı modeli, veri iletimi sırasında Rayleigh sönümlemesi etkilerini modeller. Bu, haberleşme kanalının gerçek dünyadaki bozulmaları hesaba katarak gerçekçi sonuçlar elde etmeyi amaçlar.

2x1, 2x2 ve 2x3 sistemlerde uzaysal çoğullama için otokodlayıcı modeli eğitilmiştir. 2x1 bir sistem için otokodlayıcı modeli aynı SNR değerlerinde daha düşük BER performansı göstermiştir. 2x2 ve 2x3 uzaysal çoğullama sisteminde ise otokodlayıcı geleneksel sistemler ile benzer sonuçlar vermiştir.

Çalışmada otokodlayıcı modelinin eğitim sürecini ve sistem optimizasyonunu ayrıntılı olarak açıklanmıştır. Eğitim sırasında kullanılan kayıp fonksiyonu, veri seti ile tahmin edilen veri arasındaki hata miktarını hesaplayarak ağı optimize etmek için kullanılır. Ağırlıkların güncellenmesi, Adam yöntemi kullanılarak yapılır. Bu süreç, modelin öğrenmesini ve performansını iyileştirmiştir.

Ayrıca, öğrenme aktarımı (transfer learning) kavramı da çalışmada açıklanmıştır. Öğrenme aktarımı, modelin daha verimli bir şekilde yeni görevleri çözmesine yardımcı olmaktadır. Bu bağlamda, otokodlayıcı modelinin öğrenme aktarımı benzeri bir yapı kullanarak farklı SNR aralıklarında eğitildiği belirtilmiştir.

Çalışma sonunda otokodlayıcı modelinin performansı sunulmuştur. Özellikle, Rayleigh kanalı üzerinde yapılan simülasyon sonuçları, otokodlayıcı modelinin Bit

Hata Oranı (BER) performansını göstermektedir. Bu sonuçlar, geleneksel Alamouti kodlamasına göre otokodlayıcı modelinin benzer bir performans sergilediğini göstermektedir. Sinyal-Gürültü Oranı arttıkça, otokodlayıcı tabanlı modelin performansının iyileştiği görülmüştür.

Otokodlayıcılar ile tasarlanan haberleşme sistemlerinin farklı kanal koşullarında ve farklı kestirim algoritmaları ile modellenmesi gelecekteki çalışmalar için umut verici bir konudur. Elde edilen sonucun çeşitli hiperparametre ayarlamaları yapılarak iyileştirilmesi mümkündür. Çalışma otokodlayıcı tabanlı bir sistem tasarımının kablosuz haberleşmede uzaysal çeşitlilik elde etmek için kullanılabileceğini göstermektedir. Bu sistem, geleneksel çeşitleme tekniklerine bir alternatif sunmaktadır.





KAYNAKLAR

- [1] **Shannon, C.E.** (1948). A mathematical theory of communication, *The Bell system technical journal*, 27(3), 379–423.
- [2] **O’shea, T. ve Hoydis, J.** (2017). An introduction to deep learning for the physical layer, *IEEE Transactions on Cognitive Communications and Networking*, 3(4), 563–575.
- [3] **Rice, M.** (2009). *Digital communications: a discrete-time approach*, Pearson Education India.
- [4] **Schenk, T.** (2008). *RF imperfections in high-rate wireless systems: impact and digital compensation*, Springer Science & Business Media.
- [5] **Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. ve diğerleri** (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal processing magazine*, 29(6), 82–97.
- [6] **Yao, S., Zhao, Y., Zhang, A., Hu, S., Shao, H., Zhang, C., Su, L. ve Abdelzaher, T.** (2018). Deep learning for the internet of things, *Computer*, 51(5), 32–41.
- [7] **Huang, Y. ve Chen, Y.** (2020). Survey of state-of-art autonomous driving technologies with deep learning, *2020 IEEE 20th international conference on software quality, reliability and security companion (QRS-C)*, IEEE, s.221–228.
- [8] **Hornik, K., Stinchcombe, M. ve White, H.** (1989). Multilayer feedforward networks are universal approximators, *Neural networks*, 2(5), 359–366.
- [9] **Wang, T., Wen, C.K., Wang, H., Gao, F., Jiang, T. ve Jin, S.** (2017). Deep learning for wireless physical layer: Opportunities and challenges, *China Communications*, 14(11), 92–111.
- [10] **Peng, S., Jiang, H., Wang, H., Alwageed, H., Zhou, Y., Sebdani, M.M. ve Yao, Y.D.** (2018). Modulation classification based on signal constellation diagrams and deep learning, *IEEE transactions on neural networks and learning systems*, 30(3), 718–727.
- [11] **Neumann, D., Wiese, T. ve Utschick, W.** (2018). Learning the MMSE channel estimator, *IEEE Transactions on Signal Processing*, 66(11), 2905–2917.

- [12] **Samuel, N., Diskin, T. ve Wiesel, A.** (2019). Learning to detect, *IEEE Transactions on Signal Processing*, 67(10), 2554–2564.
- [13] **Erpek, T., O’Shea, T.J., Sagduyu, Y.E., Shi, Y. ve Clancy, T.C.** (2020). Deep learning for wireless communications, *Development and Analysis of Deep Learning Architectures*, 223–266.
- [14] **He, H., Wen, C.K., Jin, S. ve Li, G.Y.** (2018). Deep learning-based channel estimation for beamspace mmWave massive MIMO systems, *IEEE Wireless Communications Letters*, 7(5), 852–855.
- [15] **Van Luong, T., Ko, Y., Matthaiou, M., Vien, N.A., Le, M.T. ve Ngo, V.D.** (2020). Deep learning-aided multicarrier systems, *IEEE Transactions on Wireless Communications*, 20(3), 2109–2119.
- [16] **Mohamed, S., Dong, J., Junejo, A. ve diğ erleri** (2019). Model-based: End-to-end molecular communication system through deep reinforcement learning auto encoder, *IEEE Access*, 7, 70279–70286.
- [17] **Alamouti, S.M.** (1998). A simple transmit diversity technique for wireless communications, *IEEE Journal on selected areas in communications*, 16(8), 1451–1458.
- [18] **Goodfellow, I., Bengio, Y. ve Courville, A.** (2016). *Deep learning*, MIT press.
- [19] **Wu, N., Wang, X., Lin, B. ve Zhang, K.** (2019). A CNN-based end-to-end learning framework toward intelligent communication systems, *IEEE Access*, 7, 110197–110204.
- [20] **Hastie, T., Tibshirani, R., Friedman, J.H. ve Friedman, J.H.** (2009). *The elements of statistical learning: data mining, inference, and prediction*, cilt 2, Springer.
- [21] **Nwankpa, C., Ijomah, W., Gachagan, A. ve Marshall, S.** (2018). Activation functions: Comparison of trends in practice and research for deep learning, *arXiv preprint arXiv:1811.03378*.
- [22] **Nair, V. ve Hinton, G.E.** (2010). Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th international conference on machine learning (ICML-10)*, s.807–814.
- [23] **Szanda, T.** (2021). Review and comparison of commonly used activation functions for deep neural networks, *Bio-inspired neurocomputing*, 203–224.
- [24] **Ruder, S.** (2016). An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747*.
- [25] **Kingma, D.P. ve Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.
- [26] **Michelucci, U.** (2022). An introduction to autoencoders, *arXiv preprint arXiv:2201.03898*.

- [27] **O'Shea, T.J., Corgan, J. ve Clancy, T.C.** (2016). Unsupervised representation learning of structured radio communication signals, *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, IEEE, s.1–5.
- [28] **Kramer, M.A.** (1991). Nonlinear principal component analysis using autoassociative neural networks, *AIChE journal*, 37(2), 233–243.
- [29] **Charte, D., Charte, F., García, S., del Jesus, M.J. ve Herrera, F.** (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines, *Information Fusion*, 44, 78–96.
- [30] **Masci, J., Meier, U., Cireşan, D. ve Schmidhuber, J.** (2011). Stacked convolutional auto-encoders for hierarchical feature extraction, *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, Springer, s.52–59.
- [31] **Zhao, T. ve Li, F.** (2021). Variational-autoencoder signal detection for MIMO-OFDM-IM, *Digital Signal Processing*, 118, 103230.
- [32] **Turhan, C.G. ve Bilge, H.S.** (2018). Variational autoencoded compositional pattern generative adversarial network for handwritten super resolution image generation, *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, IEEE, s.564–568.
- [33] **Miao, Y., Yu, L. ve Blunsom, P.** (2016). Neural variational inference for text processing, *International conference on machine learning*, PMLR, s.1727–1736.
- [34] **Caciularu, A. ve Burshtein, D.** (2018). Blind channel equalization using variational autoencoders, *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, s.1–6.
- [35] **Lopez-Martin, M., Carro, B. ve Sanchez-Esguevillas, A.** (2019). Variational data generative model for intrusion detection, *Knowledge and Information Systems*, 60, 569–590.
- [36] **Vincent, P., Larochelle, H., Bengio, Y. ve Manzagol, P.A.** (2008). Extracting and composing robust features with denoising autoencoders, *Proceedings of the 25th international conference on Machine learning*, s.1096–1103.
- [37] **Paulraj, A.J., Gore, D.A., Nabar, R.U. ve Bolcskei, H.** (2004). An overview of MIMO communications-a key to gigabit wireless, *Proceedings of the IEEE*, 92(2), 198–218.
- [38] **O'Shea, T.J., Karra, K. ve Clancy, T.C.** (2016). Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention, *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, s.223–228.

- [39] **Cammerer, S., Aoudia, F.A., Dörner, S., Stark, M., Hoydis, J. ve Ten Brink, S.** (2020). Trainable communication systems: Concepts and prototype, *IEEE Transactions on Communications*, 68(9), 5489–5503.
- [40] **Dörner, S., Cammerer, S., Hoydis, J. ve Ten Brink, S.** (2017). Deep learning based communication over the air, *IEEE Journal of Selected Topics in Signal Processing*, 12(1), 132–143.
- [41] **Morocho-Cayamcela, M.E. ve Lim, W.** (2020). Accelerating wireless channel autoencoders for short coherence-time communications, *Journal of Communications and Networks*, 22(3), 215–222.
- [42] **Wu, D., Nekovee, M. ve Wang, Y.** (2020). Deep learning-based autoencoder for m-user wireless interference channel physical layer design, *IEEE Access*, 8, 174679–174691.
- [43] **Sadeghi, M. ve Larsson, E.G.** (2019). Physical adversarial attacks against end-to-end autoencoder communication systems, *IEEE Communications Letters*, 23(5), 847–850.
- [44] **Balevi, E. ve Andrews, J.G.** (2020). Autoencoder-based error correction coding for one-bit quantization, *IEEE Transactions on Communications*, 68(6), 3440–3451.
- [45] **Aoudia, F.A. ve Hoydis, J.** (2019). Model-free training of end-to-end communication systems, *IEEE Journal on Selected Areas in Communications*, 37(11), 2503–2516.
- [46] **Chen, X., Cheng, J., Zhang, Z., Wu, L., Dang, J. ve Wang, J.** (2020). Data-rate driven transmission strategies for deep learning-based communication systems, *IEEE Transactions on Communications*, 68(4), 2129–2142.
- [47] **Lu, Y., Cheng, P., Chen, Z., Li, Y., Mow, W.H. ve Vucetic, B.** (2020). Deep autoencoder learning for relay-assisted cooperative communication systems, *IEEE Transactions on Communications*, 68(9), 5471–5488.

ÖZGEÇMİŞ

Ad SOYAD: Mustafa Bayram AYAZ

Öğrenim Durumu:

- **Lisans** : 2020, İstanbul Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- 2020 - 2021 yılları arasında P.I.WORKS şirketinde Haberleşme Mühendisi.
- 2021 - ... AR-GE Mühendisi