



DERİN ÖĞRENME İLE MÜZİK TÜR SINIFLANDIRMASI

Hasan Can CEYLAN

YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MART 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Hasan Can CEYLAN

21/03/2022

DERİN ÖĞRENME İLE MÜZİK TÜR SINIFLANDIRMASI
(Yüksek Lisans Tezi)

Hasan Can CEYLAN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mart 2022

ÖZET

Sınıflandırma işleminin, öğrenme sürecini kolaylaştırması ve zamandan sağladığı tasarruf dikkate alındığında, bu işlemin müzik öğrenimindeki katkısı yadsınamaz. Müzik sınıflandırmasında en geçerli ve etkili yöntemlerden biri müzik tür sınıflandırmasıdır. Müzik üretiminin hız kazanması ve veri sayısındaki önemli artış göz önüne alındığında, müzik türlerini sınıflandırma süreci artık insanlar tarafından yapılamayacak kadar karmaşık bir hale gelmiştir. Derin sinir ağlarının bu alandaki başarılı sonuçları da göz önünde bulundurularak, on farklı müzik türünü sınıflandırabilen bir derin öğrenme algoritması geliştirilmesi hedeflenmektedir. Çalışmada on farklı müzik türünü (blues, klasik, country, disko, hiphop, caz, metal, pop, reggae, rock) içeren GTZAN veri seti üzerinde işlemler yapılmıştır. Öznitelik olarak mel frekansı keprstral katsayıları (MFCC), sınıflandırıcı olarak ise evrişimsel sinir ağları (CNNs) kullanılmıştır. Veri setindeki müzik parçalarının ön işleme yapılarak MFCC değerleri elde edilmiş, daha sonra elde edilen bu veriler ile bir CNN modeli eğitilmiş ve test verileriyle modelin başarısı belirlenmiştir. Çalışma sonucunda, müzik türlerini sınıflandırma işlemini otomatik olarak gerçekleştiren bir model geliştirilmiştir.

Bilim Kodu : 90542
Anahtar Kelimeler : Derin Öğrenme, Müzik Türü Sınıflandırması, Evrişimsel Sinir Ağları, Mel Frekans Keprstral Katsayıları, Makine Öğrenmesi
Sayfa Adedi : 47
Danışman : Prof. Dr. Fırat HARDALAÇ

MUSIC GENRE CLASSIFICATION WITH DEEP LEARNING

(M. Sc. Thesis)

Hasan Can CEYLAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

March 2022

ABSTRACT

Allowing that the classification process facilitates the learning process and saves time, the contribution of this process to music learning is undeniable. One of the most valid and effective methods in music classification is music genre classification. Given the acceleration of music production and the significant increase in the amount of data, the process of classifying music genres has become too complex to be done by humans. Considering the successful results of deep neural networks in this field, the aim is to develop a deep learning algorithm that can classify 10 different music genres. In the study, operations have been performed on the GTZAN dataset, which includes ten different music genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock). Mel frequency cepstral coefficients (MFCC) have been used as features and convolutional neural networks (CNNs) have been used as classifiers. MFCC values have been obtained by preprocessing the music pieces in the data set, then a CNN model has been trained with these data and the success of the model has been determined with the test data. As a result of the study, a model has been developed that automatically performs the classification of music genres.

Science Code : 90542
Key Words : Deep Learning, Music Genre Classification, Convolutional Neural Networks, Mel Frequency Cepstral Coefficients, Machine Learning
Page Number : 47
Supervisor : Prof. Dr. Firat HARDALAC

TEŐEKKÜR

GerçekleőtirmiŐ olduđum tez alıŐmamda yardım ve katkılarıyla beni yönlendiren ve desteklerini esirgemeyen danışmanım Prof. Dr. Fırat HARDALA hocama, eđitim hayatım boyunca bana destek olan arkadaşlarıma ve hocalarıma, hayatımın her safhasında madden ve manen desteklerini üzerimde hissettiđim aileme teŐekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	3
3. MATERYAL.....	7
3.1. Veri Seti	7
3.2. İşaret İşleme	7
3.3. Ses Sinyallerinin İşlenmesi	7
3.4. Mel Frekans Kepstral Katsayıları (MFCC).....	10
4. METOT.....	13
4.1. Derin Öğrenme.....	13
4.2. Yapay Sinir Ağları (ANNs)	14
4.2.1. Aktivasyon fonksiyonları.....	16
4.2.2. Ağırlık güncelleme ve kayıp fonksiyonu	19
4.3. Derin Sinir Ağları.....	22
4.3.1. Evrimsel sinir ağları (CNNs).....	23
4.3.2. Düzenleme yöntemleri	25
5. DENEYSEL SONUÇLAR	29
5.1. Değerlendirme Metrikleri.....	29

	Sayfa
5.1.1. Karmaşıklık matrisi (Confusion matrix)	29
5.1.2. Doğruluk (Accuracy)	30
5.1.3. Duyarlılık (Recall)	30
5.1.4. Kesinlik (Precision).....	30
5.1.5. F-skoru (F-score).....	30
5.2. MFCC Parametreleri	30
5.3. Çalışma Ortamı	31
5.3.1. Araçlar.....	31
5.3.2. Donanım.....	32
5.4. CNN Parametre Seçimi ve Sonuçlar	32
5.4.1. Eğitim, test ve doğrulama oranı	32
5.4.2. Diğer parametreler	34
5.5. Önerilen Model ve Sonuçlar	35
6. SONUÇ VE ÖNERİLER	39
KAYNAKLAR	41
EKLER.....	45
EK-1. Önerilen CNN modeline ait kaynak kodu.....	46
ÖZGEÇMİŞ	47

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Literatürdeki çalışmalar	5
Çizelge 5.1. Test ve doğrulama oranına göre doğruluk	33
Çizelge 5.2. CNN parametre optimizasyonu	34
Çizelge 5.3. Sınıflara göre değerlendirme metrikleri.....	36
Çizelge 5.4. GTZAN veri setindeki sonuçlar.....	37



ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Blues-0 sinyal gösterimi	8
Şekil 3.2. Metal-0 sinyal gösterimi	8
Şekil 3.3. Blues-0 hızlı Fourier dönüşümü (FFT) grafiği	9
Şekil 3.4. Metal-0 FFT grafiği	9
Şekil 3.5. Blues-0 spektrogramı	10
Şekil 3.6. Metal-0 spektrogramı	10
Şekil 3.7. MFCC adımları	11
Şekil 3.8. Mel filtreleri	11
Şekil 3.9. Blues-0 MFCC	12
Şekil 3.10. Metal-0 MFCC	12
Şekil 4.1. Tek nöron yapay sinir ağı yapısı	15
Şekil 4.2. YSA yapısı	15
Şekil 4.3. Birim basamak fonksiyonu	16
Şekil 4.4. Sigmoid fonksiyonu	17
Şekil 4.5. Relu fonksiyonu	18
Şekil 4.6. Hiperbolik tanjant fonksiyonu	19
Şekil 4.7. Küçük öğrenme katsayısı	22
Şekil 4.8. İdeal öğrenme katsayısı	22
Şekil 4.9. Büyük öğrenme katsayısı	22
Şekil 4.10. Genel CNN yapısı	23
Şekil 4.11. Evrişim işlemi	24
Şekil 4.12. Maksimum ve ortalama ortaklama	25
Şekil 4.13. Normal sinir ağı yapısı	26
Şekil 4.14. Bırakma işlemi sonrası sinir ağı yapısı	27
Şekil 5.1. Karmaşıklık matrisi	29

Şekil	Sayfa
Şekil 5.2. Doğruluk grafiği	33
Şekil 5.3. Kayıp grafiği.....	34
Şekil 5.4. Önerilen CNN yapısı	35
Şekil 5.5. Önerilen model karmaşıklık matrisi	36



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

m'	Birincil Ham Moment
v'	İkincil Ham Moment
e	Euler Sayısı
f	Frekans
Σ	Toplam Sembolü

Kısaltmalar

Açıklamalar

ADAM	Adaptive Moment Estimation
ANN	Artificial Neural Network
CEL	Cross Entropy Loss
CNN	Convolutional Neural Network
CS	Compressive Sampling
DCT	Discrete Cosine Transform
DWCH	Daubechies Wavelet Transform
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
KNN	K Nearest Neighbours
LDA	Linear Discriminant Analysis
MAE	Mean Absolute Error
MFCC	Mel-Frequency Cepstral Coefficients
MSCM	Modulation Spectral Crest Measure
MSE	Mean Squared Error
MSFM	Modulation Spectral Flatness Measure
NASE	Normalized Audio Spectrum Envelope
OMSC	Octave Based Modulation Spectral Contrast
OSC	Octave Based Spectral Contrast

Kısaltmalar	Açıklamalar
RELU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Reccurent Neural Network
SPR	Statistical Patern Recognition
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
TANH	Hiperbolik Tanjant
XGB	Extreme Gradient Boosting
YSA	Yapay Sinir Ağı

1. GİRİŞ

İnsanođlu, var olduđu ilk günden beri karşılaştığı karmaşık olay ve olguları algılama ve anlama süreçlerini kolaylaştırmaya ve bir düzene sokmaya ihtiyaç duyar. Bu ihtiyacı karşılamaya yönelik ilk ve en temel yaklaşımlardan biri sınıflandırmadır [1]. Bu sebeple öğrenme sürecinin bir gerekliliđi olan sınıflandırma işlemi bilgiye daha kolay ulaşabilmenin bir ön koşulu olarak değerlendirilebilir. Nitekim hayatın her aşamasında kolaylık sağlayan düzen kavramının bilgi edinme sürecindeki önemi yadsınamaz [2]. Bu noktadan hareketle, anlama ve algılamamanın sağlanmasına yönelik gerçekleştirilen eylemlerde sınıflandırma yoluyla bir düzen oluşturmanın gerekliliđi belirlenebilir.

Müzik, algılama, ayırt etme gibi süreçler sonucunda öğrenmenin beklendiđi bir alandır. Bu alan özellikle internet kullanımının artışı ve akıllı telefonlar, bilgisayarlar ve tabletler yoluyla çok kısa sürelerde, çok sayıda veriye erişebilmenin getirdiđi kolaylıkla çok çeşitli bir yapıya bürünmüş ve bu gelişmeler neticesinde diđer bilim dalları gibi müzik alanı da girift bir yapıya dönüşmüş ve bu çok faktörlü yapıyı anlamak ve algılamak zorlaşmıştır. Günümüzde kullanılan ve müzik erişimini sağlayan uygulamaların öneri sistemlerinde de bu problem bir karmaşıklıđa sebep olmakta ve sistemlerin istenilen düzeylerde çalışmasına engel olmaktadır. Bu sebeple biriken ve çeşitlenen müzik alanında sistematik bir düzene ihtiyaç duyulmaktadır. [3] 'te de açıklandığı gibi bu düzeni sağlamada en geçerli yöntemlerden biri müzikleri türlerine göre sınıflandırmaktır.

Müzik türü sınıflandırma işleminin, bahsedilen problemlerin önemli bir kısmının aşılmasını sağlayacağı düşünülmektedir. Ancak bu işlem artan ve çeşitlenen müzik türleri ve parça sayısı nedeniyle insanlar tarafından yapılması neredeyse imkansız bir işlem haline dönüşmüştür. Ayrıca insanların müzik türü sınıflandırmasındaki ortalama başarıyı %70'tir [4]. Daha önce gerçekleştirilen otomatik müzik türü sınıflandırma çalışmalarında bu oranın aşılabildiđi gözlemlenmiştir. Bu doğrultuda, bu tez çalışmasında on farklı müzik türünü otomatik olarak sınıflandırabilen bir derin öğrenme modelinin geliştirilmesi amaçlanmıştır. Bu model sayesinde müzik sınıflandırma işlemi insanlara göre daha kısa sürede ve daha yüksek başarı oranı ile yapılabilmektedir.

Tezin organizasyonu şu şekildedir. Bölüm 1'de tezin konusu, amacı, önemi ve hangi probleme çözüm getirdiđi verilmektedir. Bölüm 2'de müzik türü sınıflandırması alanında

daha önce gerekleřtirilen alıřmalar sunulmaktadır. Blm 3'te tezde kullanılacak olan veri seti ve bu veri setindeki paraları sınıflandırma yapılabilecek řekle getirmemizi saėlayan yntemler aıklanmaktadır. Blm 4'te derin ėrenme, yapay sinir aėları ve bu yapılarda kullanılan bazı parametreler aıklanmaktadır. Blm 5'te yapılan deneyler ve bu deneyler sonucunda ortaya ıkan sonular sunulmaktadır. Son blm olan blm 6'da ise alıřmada elde edilen bulgular gz nne alınarak ıkarımlarda bulunulmakta ve bazı neriler verilmektedir.



2. LİTERATÜR TARAMASI

Otomatik ses analizi sisteminin temelinde öznitelik çıkarımı yer alır. Farklı çalışmalarda çok sayıda ve değişik tiplerde öznitelikler kullanılarak bu sistemler geliştirilmiştir. Başlarda çıkarılan öznitelikler kullanılarak istatistiksel yöntemler ile ses analizi işlemi gerçekleştirilirken, günümüzde gelişen teknoloji sayesinde bu işlem makine öğrenimi, derin öğrenme ve yapay sinir ağları gibi yapılar kullanılarak gerçekleştirilebilmektedir.

Bu tez çalışmasında ele alınan ve ses analizinin bir alt alanı olan müzik tür sınıflandırması uzun yıllardır ele alınan ve halen incelenmeye devam edilen bir alandır. Bu bölümde müzik tür sınıflandırması konusunun ele alındığı önceki çalışmalar kısaca açıklanmaktadır.

Çalışmada kullanılan veri setinin (GTZAN) ortaya konduğu [5]'te Tzanetakis ve Cook, tını, ritim ve perde bilgilerinden oluşan 3 özellik seti ortaya çıkarmışlardır. Çıkarılan özellikler ile istatistiksel örüntü tanıma (SPR) modellerinden Gauss karışımı modeli (GMM) ve k-en yakın komşu (KNN) algoritmalarını kullanarak 10 farklı müzik türünü sınıflandırmışlardır. Sınıflandırma sonucunda %61'lik bir doğruluk oranı elde etmişlerdir.

Tzanetakis ve Li [5]'te kullanılan özellik setlerini kullanmışlardır. Bu özellikleri destek vektör makinelerini (SVM) ve lineer diskriminant analiz (LDA) algoritmalarına girdi olarak vermişlerdir. Gerçekleştirilen çalışma sonucunda %71,1'lik bir sınıflandırma doğruluk oranı elde etmişlerdir ve insanların müzik türü sınıflandırmasındaki ortalama doğruluk oranını geçmişlerdir [6].

Li ve arkadaşları, GTZAN veri setinden tını, ritim, perde ve Daubechies dalgacık dönüşümü (DWCH) özelliklerini kullanmışlardır. KNN, LDA, SVM gibi makine öğrenmesi yöntemleri ile sınıflandırma işlemlerini gerçekleştirmişler ve SVM ile %79'luk bir doğruluk oranı elde etmişlerdir [7].

Li ve Ogihara, taksonomi yöntemi ile veri seti gruplara ayırarak sınıflandırma işlemini gerçekleştirmişlerdir. Bu yöntem ile elde ettikleri sonucun daha öncekine göre %0,7'lik bir gelişim kaydettiğini dile getirmişlerdir [8].

Lee ve arkadaşları, mel frekans kepstral katsayıları (MFCC), normalize ses spektrogramı (NASE) ve oktav tabanlı spektral karşıtlık (OSC) yöntemleri ile elde ettikleri öznelikleri kullanarak LDA ile GTZAN veri setinde %90,6'lık başarı elde etmişlerdir [9].

Lidy ve Rauber, GTZAN ve ISMIR veri setini kullanmışlardır. Ritim, ritim histogramları ve spektrum analizlerinden elde ettikleri öznelikleri kullanarak SVM ile sınıflandırma işlemini gerçekleştirmişler ve GTZAN veri setinde %74,9'luk bir doğruluk oranı elde etmişlerdir [10].

Chang ve arkadaşları, akustik özellikler, tını, doku özelliklerini, OSC, oktav tabanlı modülasyon spektral kontrastını (OMSC), modülasyon spektral düzlük ölçüsünü (MSFM) ve modülasyon spektral tepe ölçüsünü (MSCM) kullanarak bir özellik seti oluşturmuşlardır. Sıkıştırılmış örnekleme (CS) sınıflandırıcılarını kullanarak GTZAN veri setinde %92,7 lik bir başarı elde etmişlerdir [11].

Bu tez çalışmasında kullanılan ve önceki çalışmalarda da başarılı sonuçların elde edildiği evrimsel sinir ağı (CNN) ses sınıflandırması işlemi için bildiğimiz kadarıyla ilk kez Humphrey ve Bello tarafından kullanılmıştır. Farklı sınıflandırıcılar ile gerçekleştirilen çalışmalarda karşılaşılan güçlükleri gidermek amacıyla akor tanıma işleminde CNN'i kullanmışlar ve önceki modellere oranla daha başarılı sonuçlar elde etmişlerdir [12].

Zhang ve arkadaşları [13]'de iki farklı derin öğrenme modeli geliştirerek sınıflandırma başarısını geliştirmeyi hedeflemişlerdir. İlk modelde maksimum ortaklama ve ortalama ortaklama işlemlerini bir arada kullanmışlardır. İkinci modelde ise birincisine ek olarak kısayol bağlantıları kullanarak ağa farklı bir katmandan özellikler ekleyerek modeli oluşturmuşlardır [14]. GTZAN veri setinde %87,4'lük bir doğruluk oranı elde etmişlerdir.

Choi ve arkadaşları, özellik seti olarak olarak logaritmik mel-spektrogramları kullanarak CNN'den elde ettikleri özellikleri ve mel frekans kepstral katsayılarını birlikte kullanmışlardır. Bu özellik setlerini SVM kullanarak sınıflandırmışlar ve GTZAN veri setinde %89,8'lik bir doğruluk oranına ulaşmışlardır [15].

Feng ve arkadaşları, kısa süreli Fourier dönüşümü spektrogramlarını kullanmışlardır. Spektrogramları yinelemeli sinir ağlarına (RNN) ve CNN'e girdi olarak vermişlerdir. İki

durumdan elde edilen özellikleri birleştirmişler, birleştirilen özellik setini softmax katmanına girdi olarak iletmışler ve sınıflandırma işlemini tamamlamışlardır. GTZAN veri setinde %92 'lik doğruluk oranına ulaşmışlardır [16].

Hareesh Bahuleyan, 7 müzik türünden (hip-hop, blues, pop, reggae, rock, tekno, vokal) ve toplam 40540 örnekten oluşan veri setini kullanmıştır. İki farklı yaklaşım ile sınıflandırma işlemini gerçekleştirmiştir. Birincisi spektrogramları girdi olarak alan VGG-16 modeli, ikincisi ise çıkartılan frekans ve zaman düzlemi özniteliklerini girdi olarak alan makine öğrenmesi modelleridir. Son olarak en yüksek sınıflandırma başarımına sahip makine öğrenmesi ekstrem gradyan yükseliticisi (XGB) ve VGG-16 'yı toplu öğrenme (ensemble learning) ile kombine hale getirerek sınıflandırma işlemini tamamlamıştır. Sonuç olarak %65,0'lık doğruluk oranı elde etmiştir [17].

Nanni ve arkadaşları, müzik tür sınıflandırmasında derin öğrenme özellikleri, akustik özellikler ve görsel özellikleri incelemiştir. Yapılan çalışmalar sonucunda klasik yöntemlerle elde edilen özellikler ile CNN'in kullandığı özelliklerin farklı bilgiler içerdiğini gözlemlemişlerdir. Bu fark sebebiyle klasik ve CNN özellik vektörlerini birleştirip yeni bir özellik seti ortaya çıkarmışlardır. Bu özellik setini kullanarak SVM ile sınıflandırma işlemini gerçekleştirmişlerdir [18].

Çizelge 2.1'de GTZAN veri seti kullanılarak müzik türü sınıflandırması işleminin gerçekleştirildiği çalışmaların isimleri, yazarları ve elde edilen doğruluk oranları verilmektedir.

Çizelge 2.1. Literatürdeki çalışmalar

Yazar	Çalışma Adı	Doğruluk Oranı
Tzanetakis ve Cook [5]	Musical genre classification of audio signals	%61,00
Tzanetakis ve Li [6]	Factors in automatic musical genre classification of audio signals	%71,10
Li ve arkadaşları [7]	A comparative study on content-based music genre classification	%79,00

Çizelge 2.1 (devam). Literatürdeki çalışmalar

Yazar	Çalışma Adı	Doğruluk Oranı
Li ve Ogihara [8]	Music genre classification with taxonomy	%75,10
Lee ve arkadaşları [9]	Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features	%90,60
Lidy ve Rauber [10]	Evaluation of feature extractors and psycho-acoustic transformations for music genre classification	%74,90
Chang ve arkadaşları [11]	Music genre classification via compressive sampling	%92,70
Zhang ve arkadaşları [13]	Improved music genre classification with convolutional neural networks	%87,40
Choi ve arkadaşları [15]	Transfer learning for music classification and regression tasks	%89,80
Feng ve arkadaşları [16]	Music Genre Classification with Paralleling Recurrent Convolutional Neural Network	%92,00
Lee ve arkadaşları [19]	SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification	%82,10
Nanni ve arkadaşları [20]	Combining visual and acoustic features for audio classification tasks	%90,60
Sigtia ve Dixon [21]	Improved music feature learning with deep neural networks	%83,00

3. MATERYAL

Bu bölümde çalışmada kullanılan veri seti ve verileri derin öğrenme modelinin anlamlandırabileceği hale getirmemizi sağlayan işaret işleme teknikleri verilmektedir.

3.1. Veri Seti

Derin öğrenme modelleri, sınıflandırma, regresyon, kümeleme gibi işlemleri gerçekleştirirken verilere ihtiyaç duyar. Bu nedenle seçilecek veri setinin dağılımı ve doğruluğu çalışmanın verimliliği açısından büyük önem taşımaktadır. Seçilecek veri setinin alanda daha önce yapılmış çalışmalardaki kullanım oranı ve geçerliliği önemlidir. Çalışmanın doğruluğu ve geçerliliği bu sayede belirlenebilir. Müzik türü sınıflandırmasında önemli bir geçerliliğinin bulunması ve modelin daha doğru değerlendirilmesini sağlayacağı da göz önüne alınarak, bu çalışmada 10 farklı müzik türünü barındıran ve her tür için 100 farklı örnek içeren GTZAN [5] veri seti kullanılmıştır. Veri seti 30 saniyelik uzunluğa sahip, 22050 Hz frekans ve 16 bit olan örnekler içermektedir.

3.2. İşaret İşleme

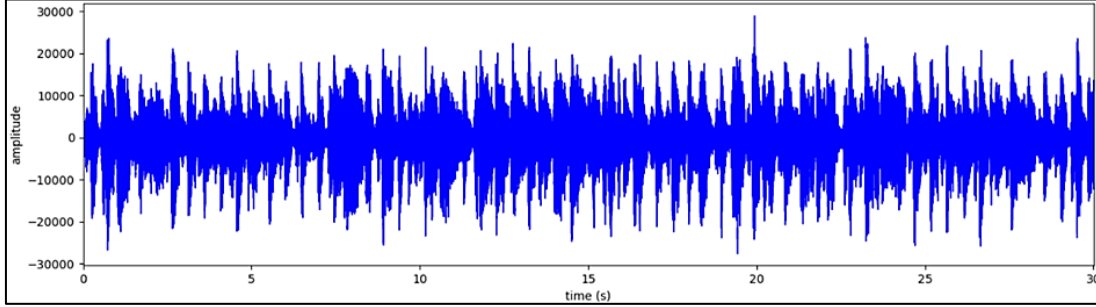
Günlük hayatta bulunan fiziksel büyüklükleri bir veya daha fazla değişkene bağlı olarak temsil etmemizi sağlayan kavrama işaret adı verilir. Ses sinyalleri de günlük hayatta karşılaştığımız bu fiziksel büyüklüklerden biridir [22].

İşaretleri sayısal olarak temsil ederek makine için anlamlı bilgiler elde etmek amaçlanır. Sayısal işaretleri işleyerek ise yeni bilgiler ve işaretler elde edilerek günlük hayattaki büyüklükler hakkında makineye daha fazla bilgi sağlanabilir. Ses işaretleri genlik, zaman, frekans ve faz bilgileri içeren büyüklüklerdir. Bu tez çalışmasında müzik kesitleri sayısal işaretler olarak inceleneceğinden bu müzik kesitlerinden anlamlı bilgiler elde etmek için sinyal işleme yöntemleri kullanılmaktadır.

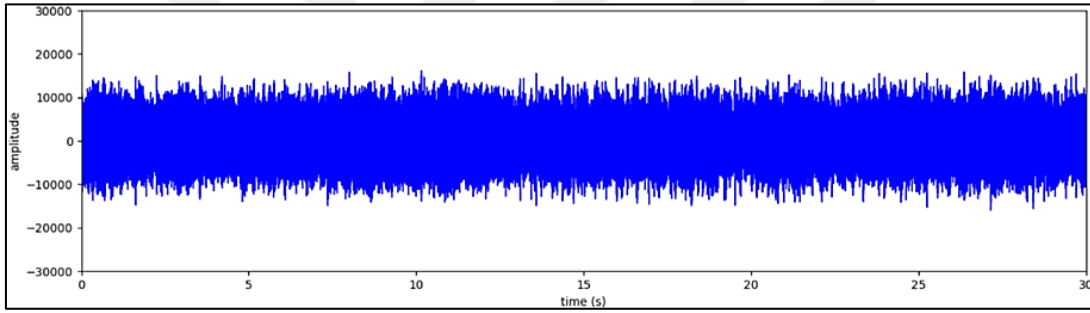
3.3. Ses Sinyallerinin İşlenmesi

Kullanılan müzik kesitleri ses sinyalleri olarak temsil edilen yapılardır. Bu sinyaller yönü ve şiddeti bulunan, zamana bağlı olarak sürekli değişen değerlere sahiptir. Ses sinyallerini bilgisayarın anlayabileceği hale getirmek için analog olan bu sinyaller belirlenen örnekleme

oranı ile örneklenecek sayısal sinyallere dönüştürülürler. Bu işlem sonucu elde edilen genlik ve zaman bilgileri ile oluşturulan grafikler ile sinyaller temsil edilir. Şekil 3.1 ve Şekil 3.2’de kullanılan veri setindeki iki farklı parçanın sinyal gösterimleri verilmektedir.



Şekil 3.1. Blues-0 sinyal gösterimi

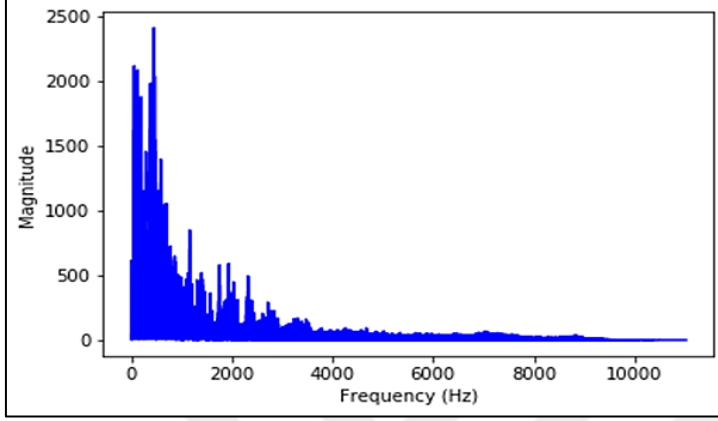


Şekil 3.2. Metal-0 sinyal gösterimi

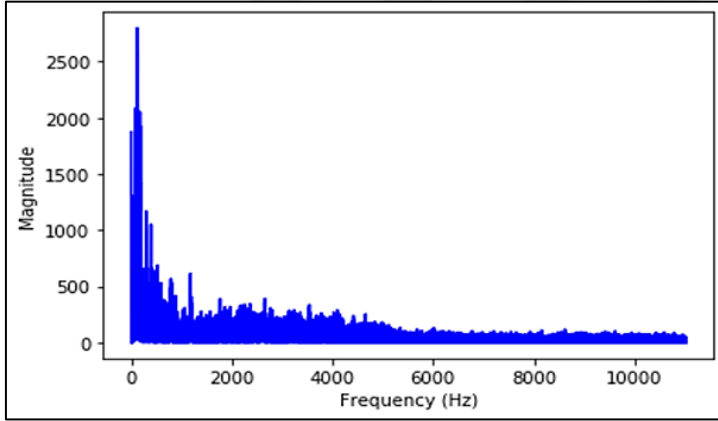
Zamana bağlı olarak sürekli değişen ve düzen içermeyen ses sinyallerinin hem karmaşık olması hem de elde edilecek bilgi ile yapılacak öğrenmenin yeterli verimliliğe sahip olamayacak olması sebebiyle bir bütün halinde incelenmemesi gerekir. Bu sebeple sinyaller belirli sayıda parçalara ayrılır ve bilgi elde etme işlemi bu parçalardan yola çıkarak gerçekleştirilir. Çerçeve uzunluğu ve aralık uzunluğu adı verilen iki değişken ise bu parçaların boyutunu belirler. Çerçeve uzunluğu, bir çerçevede bulunan örnek sayısını temsil ederken aralık uzunluğu ise her çerçevenin ne kadar örnek sonra başlayacağını belirleyen değişkendir [23].

Daha önce bahsettiğimiz, zaman düzlemi üzerindeki grafikler ve genlik değerleri bir ses sinyalinin özelliklerinin temsil edilmesi için yeterli bilgiyi barındırmamaktadır. Bu sinyallerin farklı frekans değerlerine sahip birden fazla sinüzoidal sinyalin bir araya getirilmesi ile oluşturulduğu düşüncesinden yola çıkılarak geliştirilen ve sinyallerin frekans bölgesinde temsil edilmesini sağlayan yöntem Fourier dönüşümüdür. Bu dönüşüm ile

sinyalin zaman düzleminde frekans düzlemine geçişi sağlanır. Frekans ve genlik değerleri ile elde edilen yeni grafiklere de spektrum adı verilir. Şekil 3.3 ve Şekil 3.4'te daha önce zaman düzleminde grafikleri verilen sinyallerin Fourier dönüşümü sonrası elde edilen grafikleri verilmektedir.

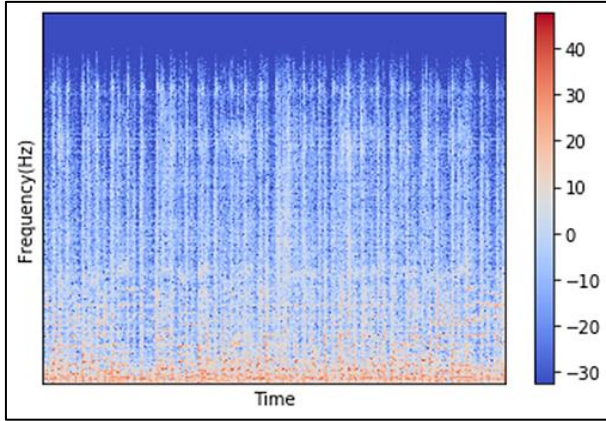


Şekil 3.3. Blues-0 hızlı Fourier dönüşümü (FFT) grafiği

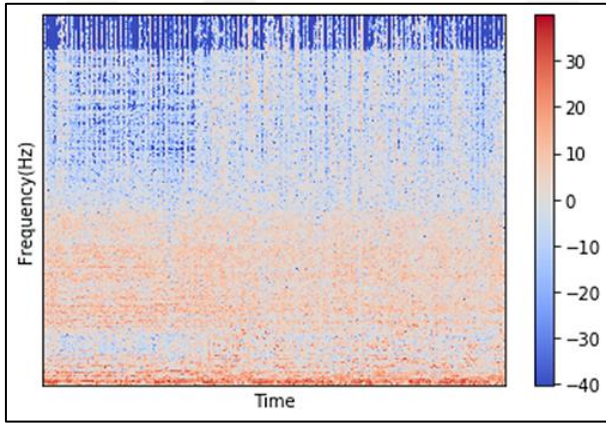


Şekil 3.4. Metal-0 FFT grafiği

Fourier dönüşümü sonrası elde edilen grafikler zaman bilgisi içermemektedir. Bu durumun önüne geçmek için ses sinyali pencerelere bölünür ve bu pencerelere Fourier dönüşümü uygulanır. Bu uygulamaya kısa zamanlı Fourier dönüşümü (STFT) adı verilir ve işlem sonucunda oluşan grafik ise spektrogram olarak isimlendirilir [23]. Şekil 3.5 ve Şekil 3.6'da daha önce grafikleri verilen örneklerin, spektrogram görüntüleri verilmektedir.



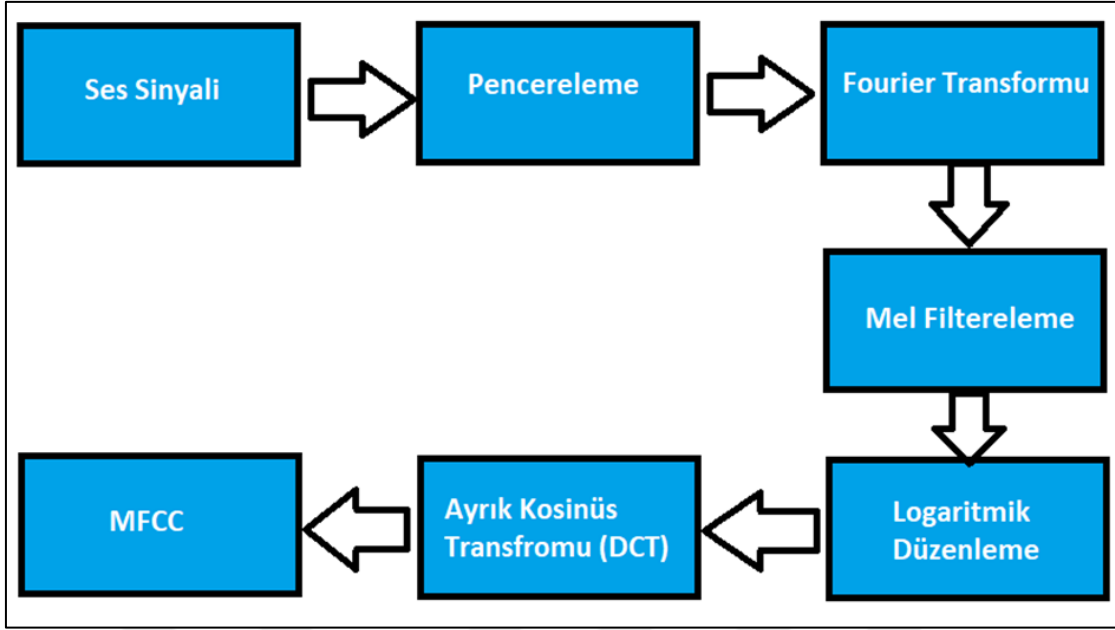
Şekil 3.5. Blues-0 spektrogramı



Şekil 3.6. Metal-0 spektrogramı

3.4. Mel Frekans Kepstral Katsayıları (MFCC)

Mel Frekans Kepstral Katsayıları (MFCC); insan ses algılama sistemine benzer şekilde ayırım yapılmasını sağlayan bir özneliktir. Bu sayede müzik sınıflandırma, ses tanıma gibi uygulamalarda yüksek başarımlara ulaşabilen bir yapıya sahiptir [24, 25]. MFCC Şekil 3.7'de gösterilen adımların uygulanması ile elde edilir.

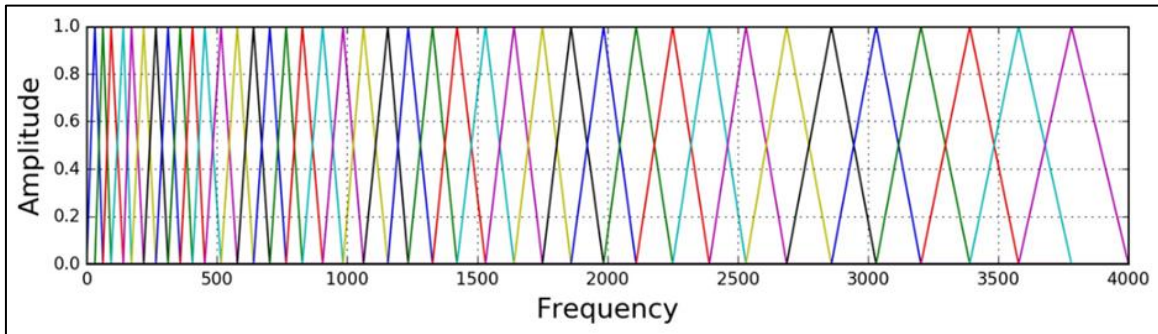


Şekil 3.7. MFCC adımları

MFCC'yi elde etmek için ses sinyali çerçvelere ayrılır ve bu çerçvelere Fourier dönüşümü uygulanır. Bu dönüşüm öncesinde çerçevelerin başlangıç ve bitiş noktalarındaki süreksizliği engellemek adına pencereleme yöntemi kullanılır. Daha sonra elde edilen işaretin frekansları mel frekansına çevrilir [26]. (Eş. 3.1)

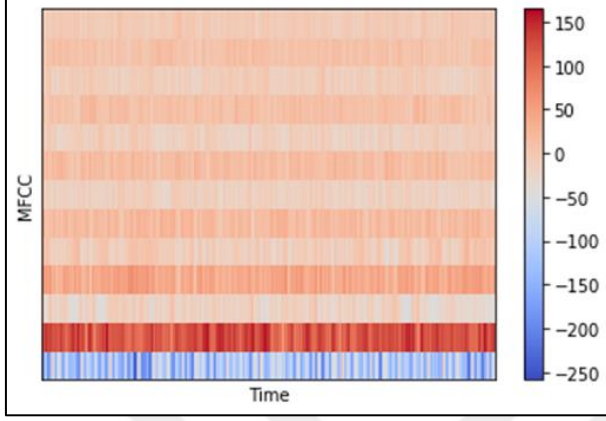
$$Mel(f) = 2595 \times \log_{10} \left(1 + \frac{f}{700} \right) \quad (3.1)$$

Sinyalin 1 kHz 'e kadar olan frekansları doğrusal olarak ölçeklendirilirken daha büyük frekanslar logaritmik olarak ölçeklendirilir. Şekil 3.8'de Mel filtrelerinin frekans spektrumu gösterimi verilmektedir.

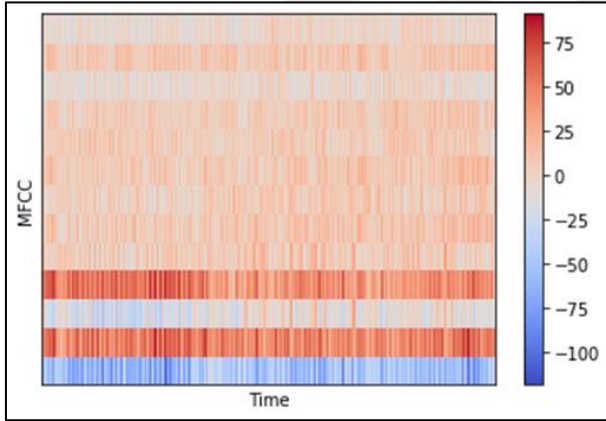


Şekil 3.8. Mel filtreleri [13]

Frekans düzenlemesi gerçekleştirildikten sonra ayrık kosinüs transformu uygulanır ve mel frekans kepstrum katsayıları elde edilir. Şekil 3.9 ve Şekil 3.10'da daha önce grafikleri gösterilen verilerin MFCC grafikleri verilmektedir.



Şekil 3.9. Blues-0 MFCC



Şekil 3.10. Metal-0 MFCC

Grafikler frekans değerlerine göre MFCC değerlerinin değişimini renk skalası ile temsil eden görüntüler şeklinde tanımlanabilir.

4. METOT

Bu bölümde çalışmada sınıflandırıcı olarak kullanılan derin sinir ağları ve bu yapıların temelini oluşturan yapay sinir ağları verilmektedir. Ek olarak modelin aşırı öğrenme yapmasını engellemek adına kullanılabilen olan düzenleme (regularization) yöntemleri de açıklanmaktadır.

4.1. Derin Öğrenme

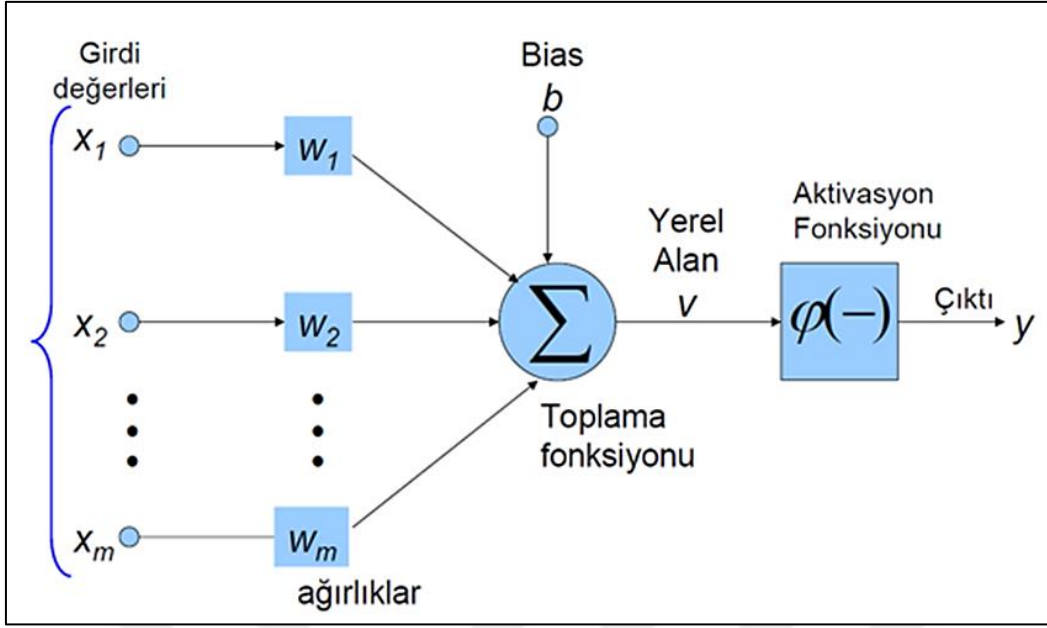
Derin öğrenme [28], makine öğrenmesinin bir alt alanıdır. Makine öğrenmesi [29], büyük veri kümelerini örneklerden öğrenme yöntemi olarak kullanan algoritmalar ile sınıflandırma, kümeleme, regresyon vb. gibi işlemleri gerçekleştirebilmemize olanak sağlayan bir öğrenme çeşididir. Derin öğrenmede ise bu gibi işlemleri gerçekleştirebilmek için çok sayıda katmandan oluşan yapay sinir ağları [30] kullanılır. Derin öğrenmenin günümüzde daha popüler bir yöntem haline gelmesinin temel sebebi ise donanım alanında kaydedilen ilerlemedir. Çünkü derin öğrenme yöntemleri matris ve vektör çarpımları gibi yüksek işlem gücü gerektiren işlemlerden yararlanarak daha önce bahsedilen (sınıflandırma, regresyon, kümeleme vb.) işlemleri gerçekleştirirler. Bu alanda, son zamanlarda kullanıma sunulan ve paralel işlem yapmamıza imkan sağlayan ekran kartları (gpu) daha önce eğitilmesi günler veya haftalar alan derin öğrenme modellerini saatler gibi sürelerde eğitilebilir hale getirmişlerdir. Daha önceleri tek bir eğitimin bile çok uzun süreler alması, derin öğrenme modellerinin hiper-parametre optimizasyonlarının gerçekleştirilmesini çok daha güç hale getirmekteydi. Bu sebeple tam olarak optimizasyonu gerçekleştirilememiş modeller kullanılmakta ve daha verimsiz sonuçlar ortaya çıkmaktaydı. Buna ek olarak gerçek zamanlı işlemlerin gerçekleştirilmesi de mümkün değildi. Kısaca donanım alanındaki ilerleme ve paralel hesaplama gücünün bize verdiği imkanlar sayesinde derin öğrenme daha doğru ve hızlı sonuçlar alabildiğimiz bir yöntem haline gelmiş ve buna bağlı olarak gün geçtikçe daha yaygın olarak kullanılmaya başlanmıştır [28].

Derin öğrenme, belirli bir veri kümesini kullanarak çıktılar tahmin etmemizi sağlayan yapay zekayı eğitmemize olanak tanır. Bu eğitim denetimli ve denetimsiz öğrenme ile gerçekleştirilebilir [31]. Denetimli öğrenme, etiketli girdilere göre elde edilecek çıktılara bağlı olarak kontrolün sağlandığı yöntemdir. Denetimsiz öğrenme ise, belirsiz durumlarda kullanılan ve kontrolün tamamen yapay zekaya bırakıldığı yöntem olarak tanımlanabilir.

Daha önce bahsedildiği üzere derin öğrenme, çok katmanlı sinir ağları ile problemlere çözümler üreten bir yapıdır. Bu sebeple bu bölümde öncelikle yapay sinir ağları açıklanmış daha sonra ise bu yapılarla beraber kullanılan derin öğrenme modellerine ve bu modellerin optimizasyonu için kullanılacak yöntemlere değinilmiştir.

4.2. Yapay Sinir Ağları (ANNs)

Bu bölümde yapay sinir ağı (YSA), YSA bileşenleri ve YSA kullanılarak yapılan sınıflandırma işlemi detaylı olarak açıklanmaktadır. Ses sinyalleri verilerinin sınıflandırılması, geleneksel sinyal işleme ve birçok istatistiksel yöntem kullanılarak oluşturulabilmesine rağmen, derin öğrenme ve makine öğrenmesi son yıllarda en başarılı yöntemler olarak öne çıkmışlardır. Bu yöntemler halen popüleritesini korumakta ve gelişerek ilerlemektedir. YSA'lar, makine öğrenmesi ve derin öğrenmede kullanılan ana araçlardan ve en temel parçalardan biri olan yapılardır. Bu sistemler, insan sinir sistemlerinin yapılarından ilham alınarak oluşturulmuştur. Bu sistemler birbirine bağlı katmanlardan oluşur ve bu katmanlar içerisinde düğüm adı verilen yapılar yer alır. Bu düğümler sinir sistemindeki biyolojik nöronlara benzer yapılar olarak düşünülebilir. Katmanlarda yer alan bu düğümler, sonraki katmandaki düğümlere ağırlıklarla bağlıdır. Bu ağırlıklar kullanılarak girdiden gelen bilgiler her katmanda ağırlıklardan faydalanılarak işlenir ve bir sonraki katmana aktarılır. YSA bu bilgileri kullanarak ve ağırlıkları güncelleyerek öğrenim işlemini gerçekleştirir. Bu ağırlıklar nöronlarda bulunan sinapslar olarak düşünülebilir [32]. Şekil 4.1' de tek nöronlu bir yapay sinir ağının genel yapısı verilmektedir.

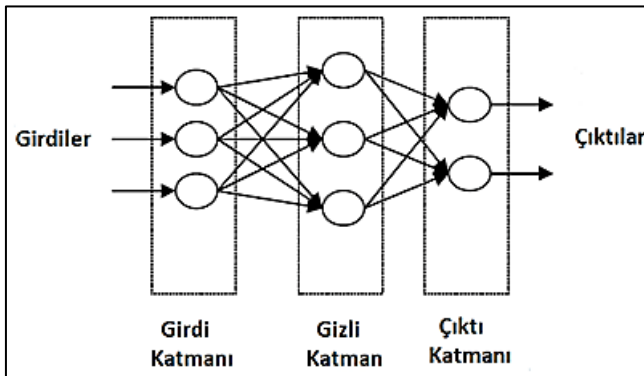


Şekil 4.1. Tek nöron yapay sinir ağı yapısı [33]

YSA'da her nöron, girdiye göre atanan ağırlıkları kullanarak toplama işlemini gerçekleştirir. Her girdi, nörona kendi ağırlığı kullanılarak beslenir. Daha sonra eşik değeri eklenir ve aktivasyon fonksiyonuna girdi olarak verilir. Kullanılan aktivasyon fonksiyonu tipine göre, toplamdan bir çıktı elde edilir. (Eş. 4.1)

$$\text{Çıktı} = \psi(b + \sum_{n=1}^m (X_n * W_n)) \quad (4.1)$$

Bir YSA genellikle bir giriş katmanı ve bir veya birden fazla gizli katman ve çıkış katmanından oluşur. Her katman önceden belirlenmiş sayıda nörona sahiptir. Girdi katmanı ise ağa verilen girdinin boyutuna eşit sayıda nörona sahiptir. Bu yapı Şekil 4.2' de verilmektedir.



Şekil 4.2. YSA yapısı [34]

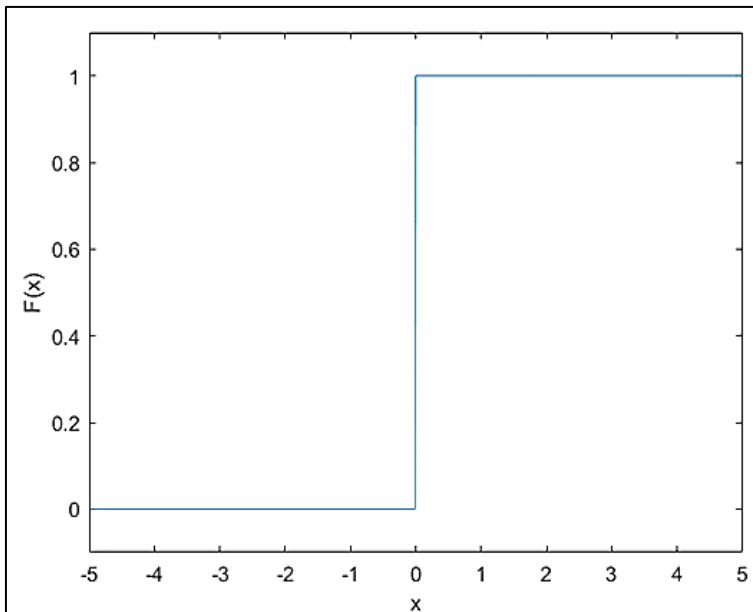
4.2.1. Aktivasyon fonksiyonları

Sinir ağlarının çıktısını ölçeklendirmek üzere kullanılan fonksiyonlar olarak tanımlanabilirler. Sınıflandırma işlemi her zaman lineer durumlar için ele alınmaz. Lineer olmayan durumlarda bu sınıflandırılmanın yapılabilmesi için modele lineer olmayan sınıflandırma sınırları eklemek gerekir. Aktivasyon fonksiyonları modele bu sınırlılıkları eklemek amacıyla kullanılırlar. Bu sayede sınıflandırma işlemi gerçekleştirilmemize olanak sağlarlar. Bu bölümde sıklıkla kullanılan aktivasyon fonksiyonları ve grafikleri verilmektedir.

Birim basamak fonksiyonu

En basit yapıdaki aktivasyon fonksiyonudur. Bir eşik değeri belirlenir ve nöronlar buna göre etkinleştirilirler. Ancak birim basamak fonksiyonu, lineer olarak ayrılamayan verileri sınıflandıramadığı için makine öğrenmesi ve derin öğrenme alanındaki son çalışmalarda tercih edilmemektedir. Verilerin doğrusal ayrımının gerçekleştirilebildiği durumlarda kullanılabilirler. Eş. 4.2 ve şekil 4.3'te birim basamak fonksiyonunun matematiksel gösterimi ve grafiği verilmektedir.

$$F(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4.2)$$

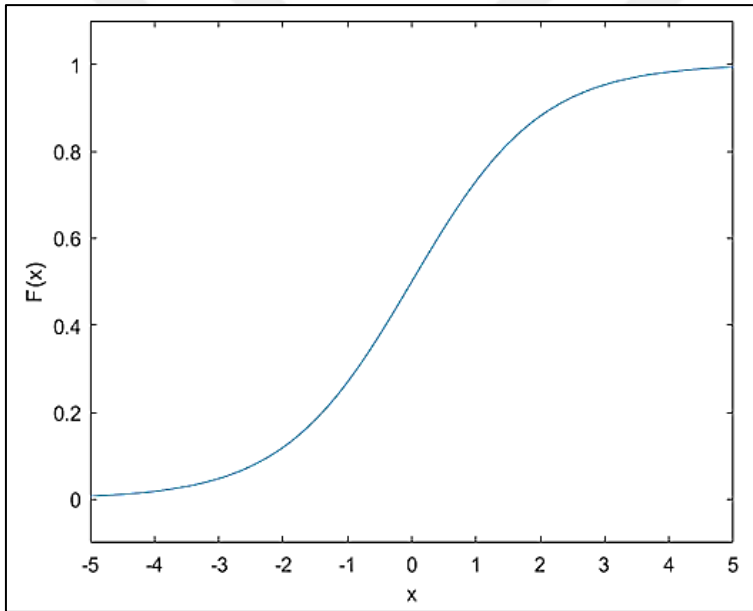


Şekil 4.3. Birim basamak fonksiyonu

Sigmoid fonksiyonu

Birim basamak fonksiyonunun yumuşatılmış (smoothened) versiyonu olarak düşünülebilir. Bu sayede doğrusal olarak ayrılamayan problemlerin çözümlenebilmesine olanak sağlar. 0 ile 1 arasında sınırlandırılmasından dolayı sınıflandırmanın ne kadar güvenilir olduğu bilgisini de içerir. Makine öğrenmesi ve derin öğrenme çalışmalarında sıklıkla kullanılan bir aktivasyon fonksiyonudur. Eş. 4.3 ve şekil 4.4'te sigmoid fonksiyonunun matematiksel gösterimi ve grafiği verilmektedir.

$$F(x) = \frac{1}{1+e^{-x}} \quad (4.3)$$

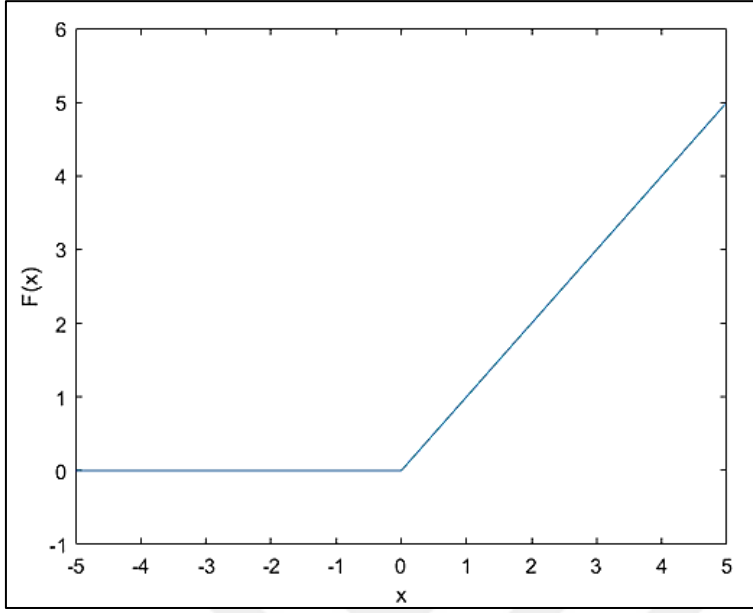


Şekil 4.4. Sigmoid fonksiyonu

Rektifiye lineer birim (Relu)

Relu, günümüzde sinir ağları tasarlanırken sıklıkla kullanılan ve doğrusal olmayan bir aktivasyon fonksiyonudur. Doğrusal olmaması lineer olmayan problemlere çözüm üretmemizi sağlar. Relu sıfırdan küçük ve eşit olan girdileri sıfıra eşitlerken sıfırdan büyük girdileri olduğu gibi bırakır. Eş. 4.4'ten anlaşılacağı üzere 0'dan küçük değerler için fonksiyon aktifleşmez. Bu sayede işlem hızından kazanç sağlar. Ancak bu durum ölü nöron sorununa neden olabilir. Eş. 4.4 ve şekil 4.5'te Relu fonksiyonunun matematiksel gösterimi ve grafiği verilmektedir.

$$F(x) = \max(0, x) \quad (4.4)$$



Şekil 4.5. Relu fonksiyonu

Softmax fonksiyonu

Softmax fonksiyonu çoklu sınıflandırma problemleri için kullanılan bir sigmoid fonksiyonu çeşididir. Verilen her bir girdinin bir sınıfa ait olma olasılığını gösteren 0 ile 1 arası çıktılar üretmektedir. Eş. 4.5'te softmax fonksiyonunun matematiksel ifadesi verilmektedir.

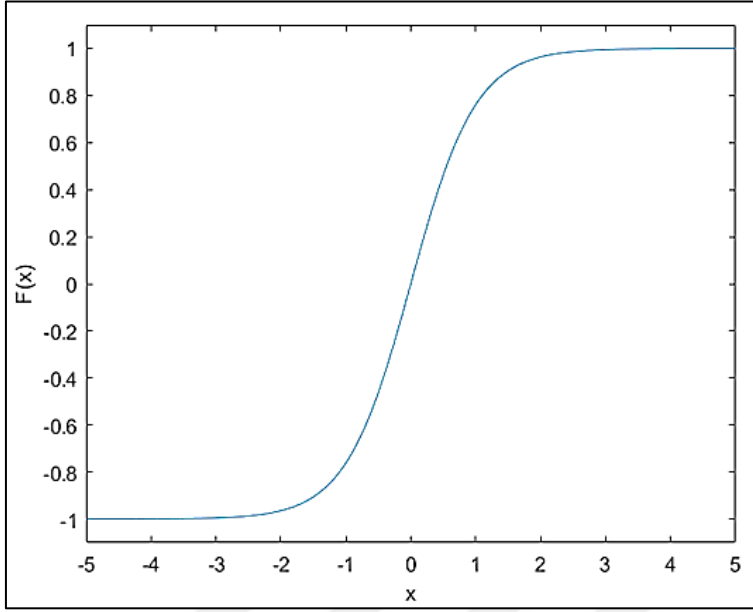
$$F(x)_j = \frac{e^{x_j}}{\sum_{k=1}^k e^{x_k}} \quad j=0,1,\dots,k \quad (4.5)$$

Hiperbolik tanjant fonksiyonu (tanh)

Hiperbolik tanjant, sigmoid fonksiyonunun ölçeklendirilmiş bir hali olarak düşünülebilir. Sigmoid fonksiyonundan farklı olarak, hiperbolik tanjant fonksiyonu orijine göre simetrik ve -1 ile 1 aralığında değerler alır. Sürekli ve türevlenebilir bir fonksiyondur bu sebeple doğrusal olmayan problemlerde de kullanılabilir. Sınıflandırma problemimizde ikiden fazla sınıf yoksa softmax yerine tercih edilebilir. İki grubun yer aldığı sınıflandırma problemlerinde negatif girdileri negatif değerlerle ve sıfır girdilerini sıfıra yakın olacak değerlerle haritalandırması bir avantaj sağlar. Eş. 4.6'da hiperbolik tanjant fonksiyonunun

(tanh) matematiksel ifadesi verilmektedir. Şekil 4.6'da hiperbolik tanjant fonksiyonun grafiği verilmektedir.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.6)$$



Şekil 4.6. Hiperbolik tanjant fonksiyonu

4.2.2. Ağırlık güncelleme ve kayıp fonksiyonu

Maliyet fonksiyonu olarak da adlandırılan kayıp fonksiyonu, sinir ağı modellerinin veri seti için ne kadar doğru çalıştığını belirlememizi sağlar. Bu tespiti eğitim süresince tahmin edilen ile gerçekteki etiketi kıyaslayarak gerçekleştirir ve kıyas sonucunda bir hata değeri döndürür. Kayıp fonksiyonunun çıktısı modelin başarısı ile ters orantılıdır, yani çıktındaki yükseliş başarısız bir modelin işaretidir. Kayıp fonksiyonu, türeve zıt olacak şekilde ağırlıkları güncellemek için geri yayılım sırasında hesaplanan hatayı kullanır. Aşağıda sıklıkla kullanılan kayıp fonksiyonları ve eşitlikleri verilmektedir.

Ortalama kare hatası (MSE)

$$L(y_n + y'_n) = 1/n \sum_n^N (y_n - y'_n)^2 \quad (4.7)$$

Karekök ortalama hata (RMSE)

$$L(y_n + y'_n) = \sqrt{1/n \sum_n^N (y_n - y'_n)^2} \quad (4.8)$$

Ortalama mutlak hata (MAE)

$$L(y_n + y'_n) = 1/n \sum_n^N |y_n - y'_n| \quad (4.9)$$

Çapraz entropi kaybı (CEL)

$$L(y_n + y'_n) = 1/n \sum_n^N |y_n * \log_{y'_n}| \quad (4.10)$$

Sinir ağı bu kayıp fonksiyonlarından elde edilen bilgileri kullanarak ağırlıklarını kayıp fonksiyonunu azaltacak şekilde günceller. Bu güncellemeyi ise çeşitli optimizasyon algoritmalarını kullanarak gerçekleştirilebilir. Bu tezde kullanılan optimizasyon algoritmalarının bazıları aşağıda açıklanmıştır.

Gradyan alçalış

Bu algoritma ile kayıp fonksiyonunun birinci mertebeden türevi, yani gradyanı, mevcut noktada hesaplanır ve kendi değeri kadar hareket ettirilerek ters yönde eğimi artırma işlemi gerçekleştirilir. Bu iki adım, her döngüdeki ağırlıklara uygulanır. Batch gradyan alçalışı (BGD), stokastik gradyan alçalışı (SGD) ve mini-batch gradyan alçalışı, yaygın olarak kullanılan ağırlık güncelleme yöntemlerinden bazılarıdır. SGD'de eğitim örnekleri rastgele karıştırılır, başka bir deyişle ağırlıklar her eğitim örneğinden sonra güncellenir [35]. Öte yandan, BGD'de tüm eğitim örnekleri ağırlık güncellemede kullanılır. SGD, BGD'den daha fazla hesaplama gerektirir ve daha hassas sonuçlar almamızı sağlar. Genelde SGD daha büyük veri kümeleri için ve BGD ise daha küçük veri kümeleri için uygun yöntemler olduğundan, SGD ve BGD'nin bir kombinasyonu olan mini-batch gradyan alçalışı geliştirilmiştir. Bu yöntemde ağırlıkları güncellemek için belirli sayıda örnekten oluşan bir batch kullanır.

Uyarlanabilir momentum tahmini (Adam) optimize edici

Momentum, gradyanın momentini önceki ağırlık güncellemelerini baz alarak kullanan ve bu sayede SGD'nin güncellemesini sağlayan terim olarak açıklanabilir. Adam [36], Eş. 4.11 ve 4.12'de gösterilen birinci ve ikinci dereceden momentumları kullanan bir optimize edicidir. Adam'ın kullanım amacı, büyük momentuma bağlı olarak global minimumun kaçırılmasını engellemektir.

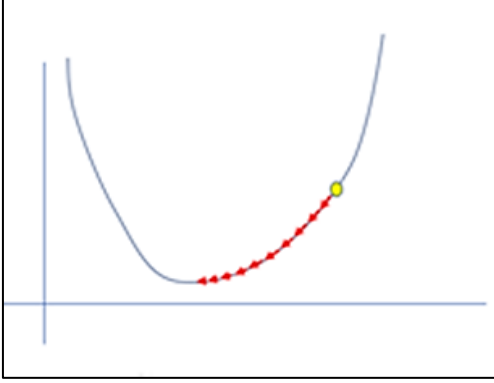
$$m'_t = \frac{m_t}{1-B^t_1} \quad (4.11)$$

$$v'_t = \frac{v_t}{1-B^t_2} \quad (4.12)$$

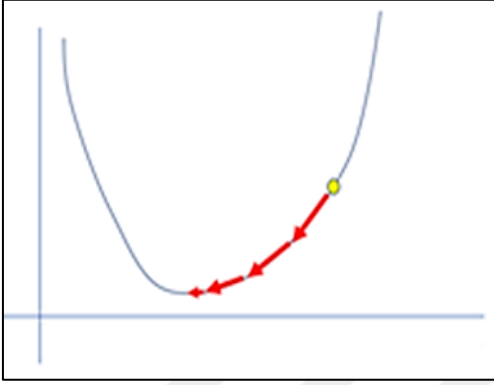
Parametre optimizasyonu ise Eş. 4.13'te ki gibi gerçekleştirilir:

$$Q_{t+1} = Q_t - \frac{\mu}{\sqrt{v'_t + \epsilon}} m'_t \quad (4.13)$$

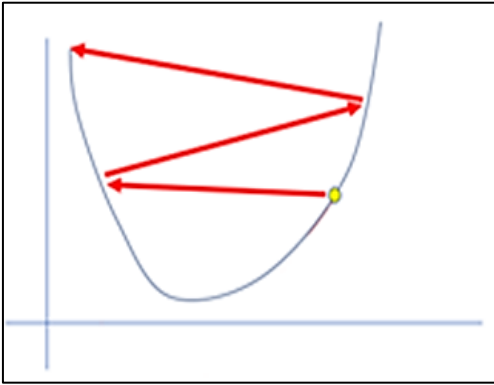
Kayıp fonksiyonunda minimum noktaya ulaşmak için optimize ediciler ile belirli büyüklükte güncellemeler gerçekleştirilir. Bu güncellemelerin oranını belirleyen değişkene ise öğrenme katsayısı ismi verilir. Büyük öğrenme katsayısı kayıp fonksiyonundaki minimum noktanın kaçırılmasına sebep olurken, küçük öğrenme katsayısı ise minimum noktaya ulaşmadan eğitimin tamamlanmasına veya bu noktaya çok daha fazla tekrar sonrasında ulaşmamıza neden olabilir. Bu sebeple öğrenme katsayısının seçimi modelin verimliliğini doğrudan etkiler. Şekil 4.7, 4.8 ve 4.9'da sırasıyla küçük, ideal ve büyük öğrenme katsayılarının seçimi durumunda eğitim adımları boyunca kayıp fonksiyonu üzerindeki ilerlemeler gösterilmektedir.



Şekil 4.7. Küçük öğrenme katsayısı [37]



Şekil 4.8. İdeal öğrenme katsayısı [37]



Şekil 4.9. Büyük öğrenme katsayısı [37]

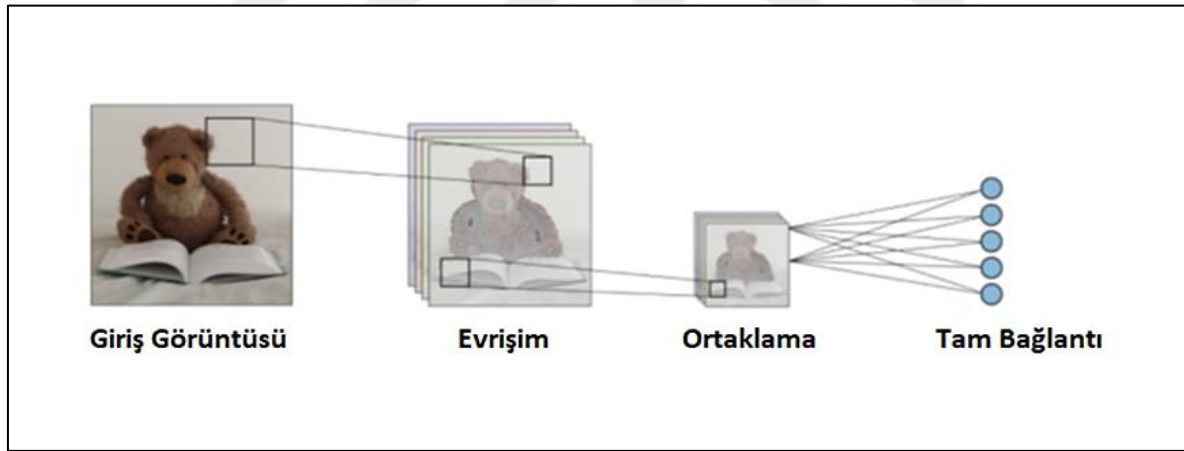
4.3. Derin Sinir Ağları

Derin sinir ağlarında “derin”, birden fazla gizli katmana sahip olmak anlamına gelir. Diğer bir deyişle derin sinir ağları, YSA’ların gizli katman sayısının artırıldığı versiyonları olarak düşünülebilir. Bu sayede veriler arasındaki ilişkileri daha iyi anlamlandırarak daha iyi

sonuçlar elde etmemizi sağlarlar. Bu tez çalışmasında derin sinir ağlarından biri olan CNN kullanılmıştır.

4.3.1. Evrişimsel sinir ağları (CNNs)

[38]'de ortaya konulan CNN'ler son yıllarda yüz tanıma, nesne algılama ve konuşma tanıma gibi çeşitli problemleri çözmek için kullanılan sinir ağı modelleridir [39]. CNN'lerin YSA'lara göre avantajları doğrusal olmayan problemlerdeki öznelik çıkarma yetenekleri ve çok boyutlu matrislerin kullanıldığı problemlerde klasik sinir ağlarına oranla kompleksliği daha az olan modellerin geliştirilebilmesidir. Örneğin 3 kanallı ve yüksek çözünürlüklü bir görüntünün, klasik sinir ağları ile temsil edilebilmesi için milyonlar seviyesinde giriş nöronuna ihtiyaç duyulacak karmaşıklık düzeyi çok yüksek bir sinir ağı mimarisi kullanmak gerekecektir. CNN'ler ise çok daha düşük komplekslikte mimariler ile bu problemleri çözebilmemizi sağlarlar. CNN'ler genel olarak şekil 4.10'da gösterilen katmanlardan oluşurlar.

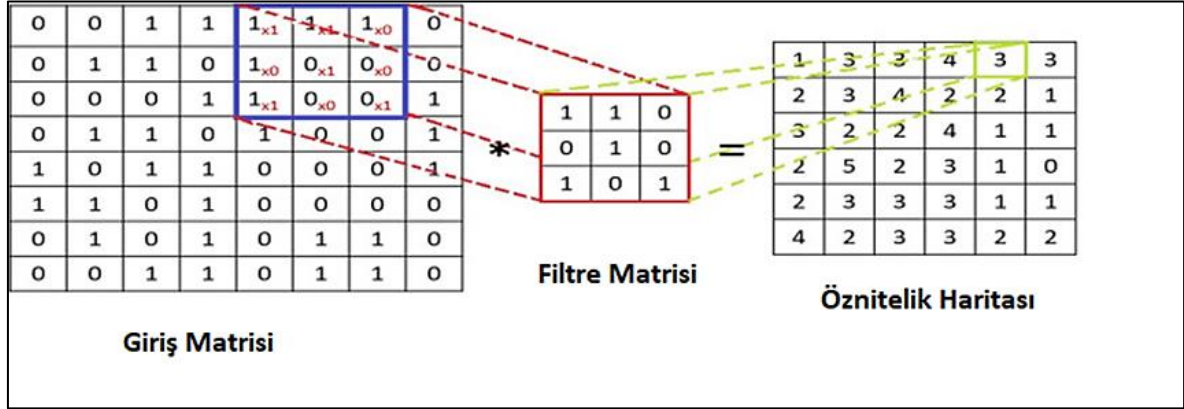


Şekil 4.10. Genel CNN yapısı [40]

Evrişim katmanı

Evrişim katmanı girdi verilerinden bir özellik haritası elde etmek amacıyla kullanılan katmandır. Filtre matrisleri girdi matrisleri üzerinde kaydırılır ve filtre matrisi ve girdi matrisi değerleri noktasal çarpım işlemine tabi tutulur. Bu işlem girdinin tamamına uygulanır. İşlemler gerçekleştirilirken adım aralığı ve filtre boyutu gibi değişkenler optimize edilebilir. Çarpım sonucunda elde edilen değerler öznelik haritası veya özellik haritası olarak bilinen matrisi oluştururlar. Evrişim katmanında ağa doğrusal olmayanlık

kazandırmak için aktivasyon fonksiyonları kullanılır. Relu, softmax, hiperbolik tanjant CNN’lerde sıklıkla kullanılan aktivasyon fonksiyonlarıdır. Şekil 4.11’de 8x8 boyutundaki bir girdi matrisine 3x3 filtre matrisi ile uygulanan evrişim işlemi verilmektedir.



Şekil 4.11. Evrişim işlemi [41]

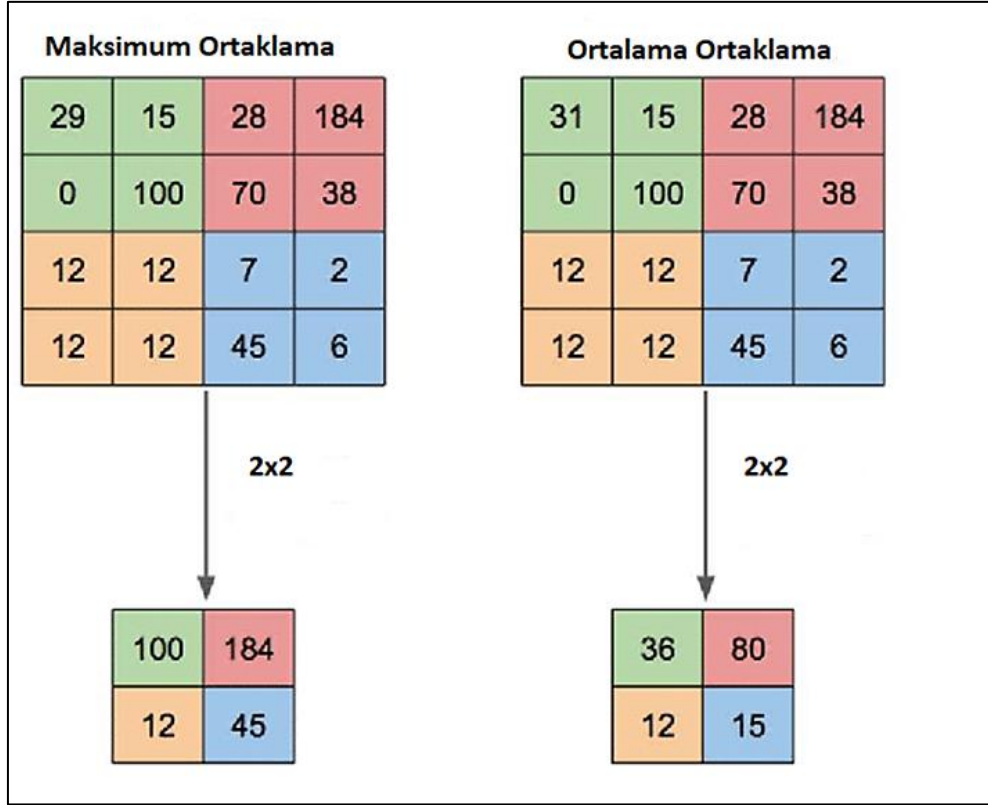
Ortaklama katmanı

Bu katmanda evrişim katmanı sonucunda elde edilen öznitelik matrislerine bir boyut azaltma uygulanır. Ortaklama katmanında girdiye yine bir filtre matrisi uygulanarak çıktı matrisi elde edilir. İki tip ortaklama yöntemi bulunmaktadır.

Maksimum Ortaklama: Burada işlem gerçekleştirilen alanda bulunan en yüksek değere sahip değer çıktı matrisine yazılır.

Ortalama Ortaklama: Bu yöntemde ise işlem gerçekleştirilen alandaki değerlerin ortalaması çıktı matrisine yazılır.

Ortaklama katmanında birçok bilgi kaybolmasına rağmen kompleksliği azaltma, verimliliği artırma ve aşırı öğrenme ihtimalini minimize etme gibi faydaları bulunmaktadır. Şekil 4.12’de örnek ortaklama işlemleri gösterilmektedir.



Şekil 4.12. Maksimum ve ortalama ortaklama [42]

Tam bağlantı katmanı

Tam bağlantı katmanında, ortaklama katmanı sonucunda elde edilen matrisler tek boyutlu hale getirilir ve sınıflandırma işlemi gerçekleştirilir. Bu katmanın son katmanında bulunan nöron sayısı sınıflandırma yapılacak olan sınıf sayısına eşit sayıdadır. Evrişim katmanının aksine bu katmanda aktivasyon fonksiyonu olarak genellikle softmax aktivasyon fonksiyonu kullanılır ve 0 ile 1 arasında olasılıklar şeklinde tahmin edilen çıktılar verilir.

4.3.2. Düzenleme yöntemleri

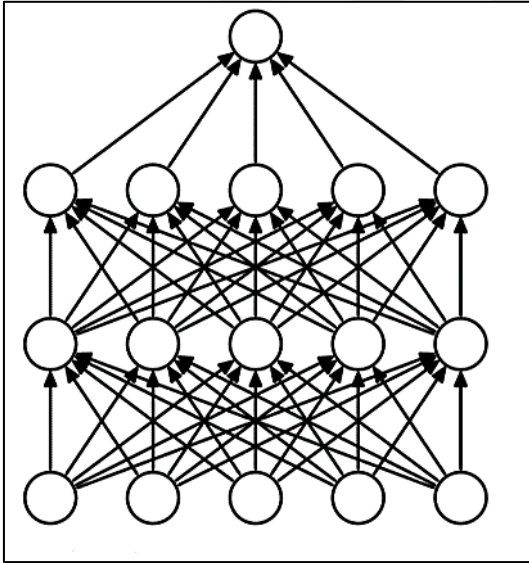
Bu bölümde, ağırlık aşırı öğrenme (ezberleme) yapmasını engellemek amacıyla kullanılan düzenleme yöntemlerinin bir kısmı açıklanmaktadır.

Bırakma operasyonu (Dropout)

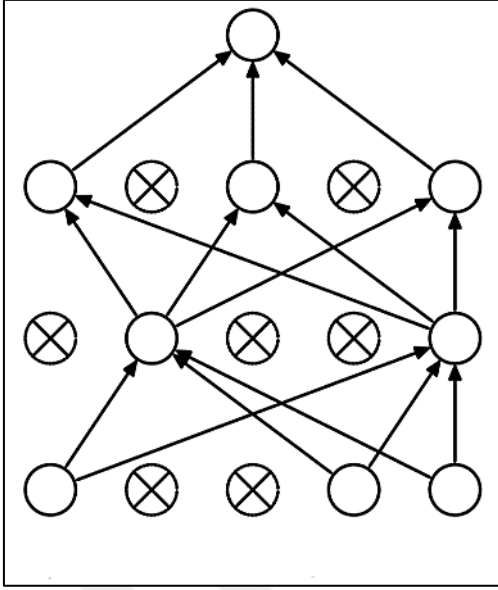
Aşırı öğrenme, makine öğrenmesi, YSA'lar ve derin sinir ağlarında karşılaşılan ortak bir sorundur. Sinir ağları, kullanılan girdi ve çıktılar arasında ilişkiler kurarak etiketlenmemiş

verilerin sınıflarının tahminini gerçekleştirirler. Test verilerinde başarısız bir sınıflandırma yapılan ancak eğitim verilerinde başarılı sonuçlar alınan sinir ağlarının aşırı öğrenme gerçekleştirdiği söylenebilir. Sinir ağını daha fazla eğitim verisi ile beslemek ağın farklı özellikleri temsil edebilir hale gelmesini sağlayacağından bu durumu aşmak için kullanılabilir yöntemlerdendir. Ancak yüksek sayıda eğitim verisi ağın karmaşıklığını da artırmış olacaktır. Bu durum ise daha fazla eğitim sürelerine ihtiyaç duyulmasına neden olacaktır. Aşırı öğrenmeyi gidermek için çeşitli araştırmalar yürütülmüştür. Srivastava ve arkadaşları [43]'de sinir ağından rastgele nöronların çıkarılmasının ardından kalan nöronların diğer nöronlara ait önemli bilgileri de tutmaya başladıklarını keşfettiler.

Bırakma işlemi, aşırı öğrenmeyi azaltarak ağ performansını artırmayı amaçlayan bir iyileştirme yöntemidir [43]. Her eğitim adımında belirlenen oranda nöron çıkarılır ve bu sayede ağın genelleştirme yeteneği artırılmış olur. Bu oran 0 ile 1 aralığında seçilen bırakma oranı ile belirlenir. Şekil 4.13'te normal ağ yapısı şekil 4.14'te ise aynı ağın bırakma işlemi uygulandıktan sonraki yapısı gösterilmektedir.



Şekil 4.13. Normal sinir ağı yapısı [44]



Şekil 4.14. Bırakma işlemi sonrası sinir ağı yapısı [44]

Erken durdurma (Early stopping)

Erken durdurma da eğitim verilerinin bir kısmı doğrulama seti olarak kullanılır. Doğrulama seti eğitimin her tekrarında ağıın öğrenme durumunu kontrol etmek için kullanılır. Bu sayede doğrulama verilerindeki hata oranı takip edilebilir ve bu oran yükselmeye başladığında eğitim durdurularak aşırı öğrenme problemi engellenebilir. Ancak bu işlem gerçekleştirilirken tüm verileri temsil edebilecek oranda bir doğrulama seti seçilmelidir aksi halde öğrenme tam anlamıyla sağlanmadan erken durdurma gerçekleştirilmiş olur.

Ağırlık azaltma (Weight decay)

Ağırlık azaltma, kayıp fonksiyonuna L1 veya L2 gibi bir düzenleme terimi eklenmesi ile aşırı öğrenmeyi azaltmayı hedefleyen başka bir yöntemdir. L1, ağırlıkların mutlak değerlerinin toplamıdır ve seyrek ağırlık matrisleri elde etmemizi sağlar. L2, ağırlıkların karelerinin toplamıdır ve hesaplamayı daha verimli hale getirir.



5. DENEYSEL SONUÇLAR

Bu bölümde gerçekleştirilen deneylerin sonuçları değerlendirme metrikleri göz önüne alınarak verilmektedir. Ayrıca seçilen parametrelerin seçim sebepleri ve sonuçları da gösterilmektedir.

5.1. Değerlendirme Metrikleri

Bu bölümde oluşturulan modelin başarımını değerlendirmek için kullanılacak olan değerlendirme metrikleri açıklanmaktadır.

5.1.1. Karmaşıklık matrisi (Confusion matrix)

Karmaşıklık matrisi kısaca test verileri için beklenen ve elde edilen sonuçların karşılaştırıldığı tablo olarak tanımlanabilir. Şekil 5.1’de iki sınıf için kullanılan örnek bir karmaşıklık matrisi verilmektedir. Bu tabloda bulunan değişkenler kullanılarak çeşitli değerlendirme metrikleri elde edilir ve bu metrikler modelin başarımını değerlendirmek için kullanılabilir.

Karmaşıklık Matrisi	Pozitif Tahmin	Negatif Tahmin
Gerçek Pozitif	Doğru Pozitif (TP)	Yanlış Negatif (FN)
Gerçek Negatif	Yanlış Pozitif (FP)	Doğru Negatif (TN)

Şekil 5.1. Karmaşıklık matrisi

Doğru Pozitif (TP): Pozitif olan değer pozitif olarak doğru tahmin edildiği veri sayısıdır.

Yanlış Pozitif (FP): Negatif olan değer pozitif olarak yanlış tahmin edildiği veri sayısıdır.

Yanlış Negatif (FN): Pozitif olan değer negatif olarak yanlış tahmin edildiği veri sayısıdır.

Doğru Negatif (TN): Negatif olan değer negatif olarak doğru tahmin edildiği veri sayısıdır.

5.1.2. Doğruluk (Accuracy)

Doğru Pozitif (TP) ve Doğru Negatif (TN) toplamının, tüm tahminlerin toplamına oranıdır.

Eş. 5.1’de doğruluk denklemi verilmektedir.

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+FN+TN} \quad (5.1)$$

5.1.3. Duyarlılık (Recall)

Doğru pozitif (TP) değerinin, doğru pozitif (TP) ve yanlış negatif (FN) değerlerinin toplamına oranıdır. Eş. 5.2 ‘de duyarlılık denklemi verilmektedir.

$$\text{Duyarlılık} = \frac{TP}{TP+FN} \quad (5.2)$$

5.1.4. Kesinlik (Precision)

Doğru pozitif (TP) sayısının, doğru pozitif (TP) ve yanlış pozitif (FP) toplamına oranıdır. Eş. 5.3’te kesinlik denklemi verilmektedir.

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (5.3)$$

5.1.5. F-skoru (F-score)

F skoru, duyarlılık ve kesinlik değerinin harmonik ortalamasıdır. Sınıflandırıcıların kıyaslanmasında sıklıkla kullanılır ve yanlış pozitif ve yanlış negatif durumları hakkında bilgi elde etmemizi sağlar. Eş. 5.4’te F-skoru denklemi verilmektedir.

$$\text{FSkoru} = \frac{2 \cdot \text{Duyarlılık} \cdot \text{Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}} \quad (5.4)$$

5.2. MFCC Parametreleri

İnsanların müzik türlerini sınıflandırma probleminde [26]’da açıklandığı üzere 3 saniye üzerinde uzunlukta ses dinlemeleri durumunda sınıflandırma doğrulukları herhangi bir

iyileşme kaydetmemektedir. Müziği insanlar için üretilmiş bir sanat ürünü olarak değerlendirecek olursak geliştirilecek modelin de insan tepkisine benzer tepkiler verecek şekilde geliştirilmesi gerektiği kanısına varabiliriz. Bu sebeple 3'er saniyelik örnekler üzerinden MFCC değerleri elde edilmiştir. Python'un librosa kütüphanesi kullanılarak, [6]'da olduğu gibi n-mfcc, hop-length ve n-fft değerleri sırasıyla 13, 512 ve 2048 olarak seçilmiştir ve bunun sonucunda elde edilen 13x130 boyutundaki MFCC değerleri 26x65 boyutlarında olacak şekilde düzenlenerek kullanılmıştır.

5.3. Çalışma Ortamı

Bu bölümde tezde kullanılan yazılım araçları ve donanımlar açıklanmaktadır. Bu çalışmada programlama dili olarak Python kullanılmaktadır.

5.3.1. Araçlar

Bu bölümde çalışma boyunca kullanılan yazılım araçları açıklanmaktadır.

Jupyter notebook

Jupyter Notebook, çok sayıda programlama dili ile kullanılabilen, kod düzenlenmeye ve çalıştırmaya yarayan açık kaynaklı bir web uygulamasıdır. Çalışma boyunca geliştirilen kodlar bu araç kullanılarak derlenmiş ve çalıştırılmıştır.

Numpy

Numpy, Python programlama dili için geliştirilmiş olan ve nümerik işlemleri hızlı bir şekilde gerçekleştirmemize olanak sağlayan bir hesaplama kütüphanesidir. Matriks, tensör gibi yapılar üzerinde işlemler gerçekleştirebilmemize olanak sağlar.

Pandas

Pandas, Python programlama dili için geliştirilmiş makine öğrenmesi alanında sıklıkla kullanılan açık kaynaklı bir veri analizi ve işleme aracıdır. Pandas, Numpy kütüphanesinin üzerine inşa edilmiş bir kütüphanedir.

Tensorflow

Tensorflow, makine öğrenimi alanında kullanılan hesaplamaları, işlemleri ve görselleştirmeleri gerçekleştirebilmemize olanak sağlayan açık kaynaklı bir yazılım kütüphanesidir. Tensorflow ile bir makine öğrenmesi modeli geliştirebilir, eğitebilir ve geliştirilen bu modeli kaydedebilirsiniz. Çalışmada kullanılan Keras uygulama arayüzü, Tensorflow kütüphanesinin altyapısını kullanarak gerekli işlemleri gerçekleştirdiğinden, dolaylı bir kullanım söz konusudur.

Keras

Keras, Python programlama dili için geliştirilmiş olan bir yapay sinir ağı uygulama arayüzüdür. Tensorflow veya Theano altyapısını kullanarak çalışır. Katmanlar halinde sinir ağlarının oluşturulabilmesine olanak sağlayan yapısı sayesinde daha kısa kodlar ile karmaşık modeller geliştirilebilmesine olanak sağlar.

5.3.2. Donanım

Bu tez çalışması boyunca gerçekleştirilen deneyler iki farklı donanım ortamı üzerinde gerçekleştirilmiştir. İlk donanım ortamı Nvidia GTX1660Ti ekran kartının ve Nvidia'nın Cuda altyapısının kullanıldığı fiziksel donanım ortamıdır. İkinci donanım ortamı ise Google Colab ile bulut ekran kartları üzerinden deneylerin gerçekleştirildiği sanal donanım ortamıdır. Google Colab ortamında iki farklı bulut ekran kartı kullanılarak deneyler gerçekleştirilmiştir. Bu bulut ekran kartları Nvidia Tesla T4 ve Nvidia Tesla K80'dir.

5.4. CNN Parametre Seçimi ve Sonuçlar

Bu bölümde yapılan deneyler ile farklı hiper parametrelerin modelin başarımı ile ilişkisi ele alınmakta ve en başarılı sonuçların elde edildiği modelin yapısı verilmektedir.

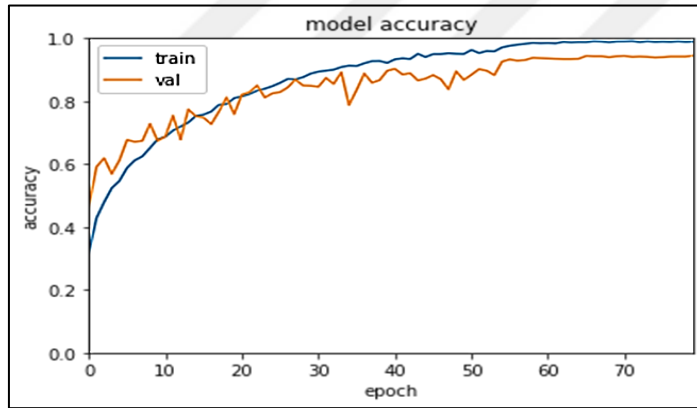
5.4.1. Eğitim, test ve doğrulama oranı

Makine öğrenmesi ve derin öğrenmede veri setinin kümelenmesi gerçekleştirilmemişse ilk olarak ele alınması gereken parametre verilerin eğitim, test ve doğrulama kümelerine ayrılmasıdır. Kullanılan veri seti GTZAN'da veriler önceden kümelenmediği için farklı

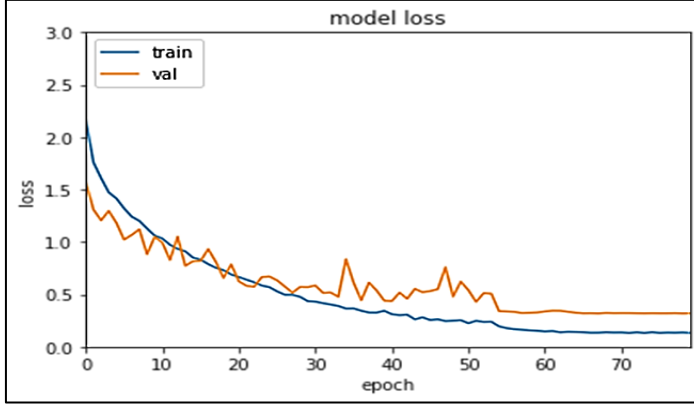
oranlarda seçilen kümeleme oranlarının sonuçları incelenmiştir. Çizelge 5.1’de farklı oranlardaki test ve doğrulama kümeleri yüzdeleri için elde edilen test doğruluk oranları verilmektedir. Şekil 5.2 ve 5.3’te ise en yüksek doğruluk oranının elde edildiği %10 test ve %10 doğrulama ayırımı için eğitim süresince ki doğruluk ve kayıp grafikleri verilmektedir. Çalışmada sonraki hiper parametre seçimleri bu oran kullanılarak gerçekleştirilmektedir.

Çizelge 5.1. Test ve doğrulama oranına göre doğruluk

Test Oranı	Doğrulama Oranı	Doğruluk
%10	%10	%94,30
%15	%15	%93,73
%10	%20	%93,30
%20	%10	%92,20
%20	%20	%91,05
%30	%10	%90,90
%10	%30	%91,76



Şekil 5.2. Doğruluk grafiği



Şekil 5.3. Kayıp grafiği

5.4.2. Diğer parametreler

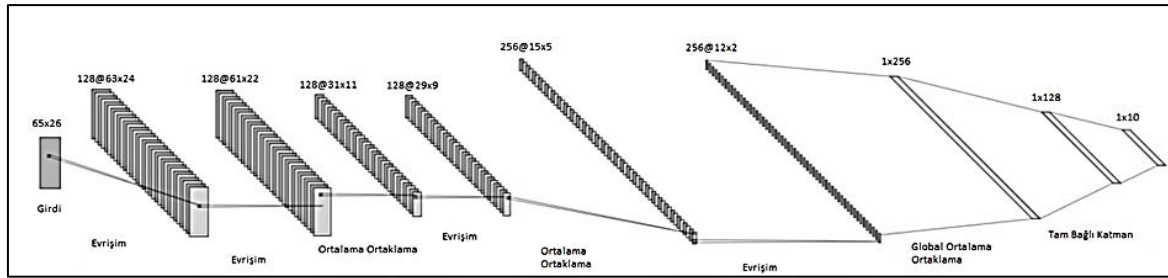
Derin öğrenme modellerinde çok sayıda parametre bulunması ve bu parametrelerin modelin başarısını etkilemesi sebebiyle optimum değerlerinin bulunması doğru modelin geliştirilmesinde önemli bir etkidir. Parametre optimizasyonu için başlangıç değerleri: 32 batch boyutu, 128 filtre, Relu-Softmax fonksiyonları, Adam optimize edici ve ortalama ortaklamadır. Başlangıç parametrelerinin teker teker değiştirilmesi sonucunda elde edilen doğruluk oranları çizelge 5.2’de verilmektedir.

Çizelge 5.2. CNN parametre optimizasyonu

Parametre Çeşidi	Parametre	Doğruluk
Batch Boyutu	128	%93,40
	64	%92,73
	32	%94,30
	16	%94,20
Filtre Sayısı	256	%93,90
	128	%94,30
	64	%90,60
	32	%86,20
Aktivasyon Fonksiyonu	Relu-Softmax	%94,30
	Hiperbolik Tanjant-Softmax	%92,30
	Relu-Sigmoid	%93,00
	Hiperbolik Tanjant-Sigmoid	%94,20
Optimize Edici	Adam	%94,30
	SGD	%63,20
	RMSProp	%94,20
	Adamax	%87,10
	Adagrad	%66,50
	Nadam	%94,10
Ortaklama Katmanı	Maksimum Ortaklama	%93,10
	Ortalama Ortaklama	%94,30

5.5. Önerilen Model ve Sonuçlar

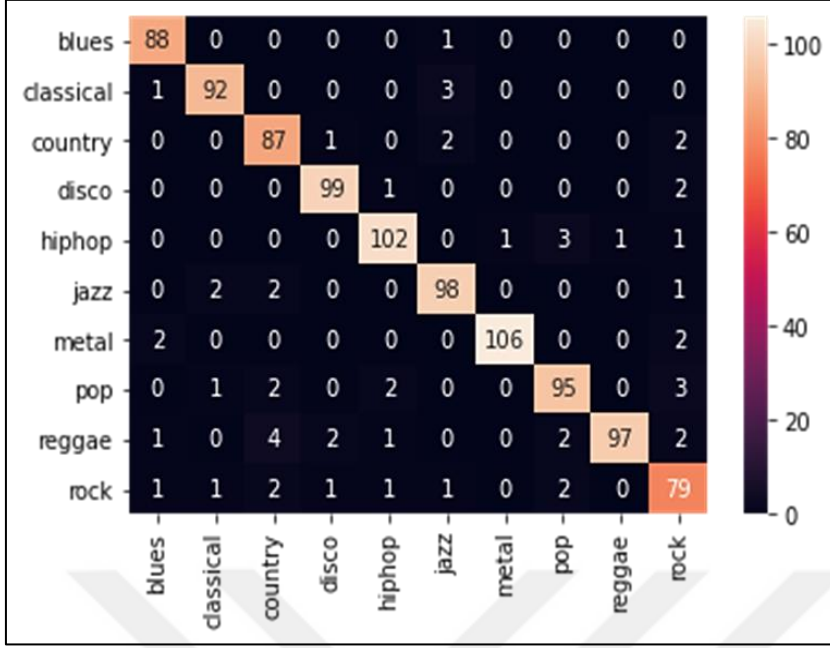
Gerçekleştirilen parametre optimizasyonları sonrasında en başarılı sonuçların elde edildiği model şekil 5.4'te gösterilmektedir. Eğitim geri yayılım algoritması kullanılarak 80 tekrar olacak şekilde gerçekleştirilmiştir. Optimize edici olarak Adam, hata fonksiyonu olarak ise seyrek kategorik çapraz entropi (Sparse Categorical Cross Entropy) kullanılmıştır. Ayrıca EK-1'de önerilen CNN modeline ait kaynak kodu verilmektedir.



Şekil 5.4. Önerilen CNN yapısı

Modelin daha doğru bir şekilde değerlendirilmesi adına eğitim ve test verilerinin yerlerinin değiştirilerek modelin eğitildiği k-katlamalı çapraz doğrulama modele uygulanmış ve %93,4'lük bir test doğruluk oranı ortaya çıkmıştır. Daha önceki çalışmaların bazılarında bu oran geçilse de [45]'te açıklandığı üzere GTZAN veri setinde bulunan gürültü ve hatalar sebebiyle elde edilebilecek anlamlı en yüksek doğruluk oranı %94,50'tir. Bu sebeple bu tez çalışmasında en son teknoloji (state of the art) olarak %94,50 oranı baz alınmaktadır.

Önerilen modelin %94,30'luk test doğruluk oranına sahip bir katlaması için karmaşıklık matrisi şekil 5.5'te verilmektedir. Ayrıca yine aynı katlama için çizelge 5.3'te önerilen modelin sınıf bazında değerlendirilmesi adına her sınıf için duyarlılık, kesinlik ve F-skoru değerleri verilmiştir.



Şekil 5.5. Önerilen model karmaşıklık matrisi

Çizelge 5.3. Sınıflara göre değerlendirme metrikleri

Tür	Duyarlılık	Kesinlik	F-Skoru
Blues	%98,88	%94,62	%96,70
Classical	%95,83	%95,83	%95,83
Country	%94,57	%89,69	%92,06
Disco	%97,06	%96,12	%96,59
Hiphop	%94,44	%95,33	%94,88
Jazz	%95,15	%93,33	%94,23
Metal	%96,36	%99,07	%97,70
Pop	%92,23	%93,14	%92,68
Reggae	%88,99	%98,98	%93,72
Rock	%89,77	%85,87	%87,78
Tüm Türlerin Ortalaması	%94,20	%94,33	%94,22

Deneysel sonuçta en yüksek başarımlar blues ve metal türlerinde elde edilmiştir. En düşük oranlar ise rock ve country türlerinde elde edilmiştir. Kesin olmamakla birlikte bu durumun nedeni 3 saniyelik parçaların bu iki tür için yeterli bilgiyi içermemesi veya bu iki türde benzer seslerin bulunması olabilir. Kesin bir yargıya varılamamasının nedeni, derin öğrenmede hala tam olarak inceleyemediğimiz alanların bulunmasıdır. Çizelge 5.4'te daha

önce GTZAN veri setine uygulanmış çalışmaların bazılarının doğruluk oranları ve önerilen modelin doğruluk oranı verilmektedir.

Çizelge 5.4. GTZAN veri setindeki sonuçlar

Çalışma	Test Doğruluğu
Önerilen Model	%93,40
Lee ve arkadaşları	%82,10
Choi ve arkadaşları	%89,80
Lee ve arkadaşları	%90,60
Nanni ve arkadaşları	%90,60
Zhang ve arkadaşları	%87,40
Feng ve arkadaşları	%92,00



6. SONUÇ VE ÖNERİLER

Müziğin çok unsurlu ve karmaşık yapısı nedeniyle, müziği alt sınıflara ayırarak incelemek bu karmaşıklığı azaltmakta ve dolayısı ile öğrenme ve anlama süreçlerini daha hızlı ve kolay hale getirmektedir. Bu ayrımı yaparken kullanabileceğimiz en geçerli yöntemlerden biri de müzikleri türlerine göre sınıflandırmaktır. Müzik öğrenimi ve öğretimi sürecinde müzik tür sınıflandırması, algılama ve yorumlamayı kolaylaştırması ve öğrenme sürecinde daha belirgin çizgiler oluşturması açısından büyük önem taşımaktadır. Özellikle teknolojik gelişmeler ve internet erişiminin son derece kolay bir hale gelmesi ile birlikte, üretim ve tüketim hızı artmış, bu durum müzik alanında da kendini göstermiştir. Teknolojinin sağladığı bu kolaylıklar hem çok sayıda parça üretilmesine hem de farklı türler geliştirmeye imkan tanımaktadır. Bu durumun sonucunda ortaya çıkan karmaşık yapıda insan eliyle bir sınıflandırma işlemi yapabilmek daha zor bir hal almış dolayısı ile bu işlemi yapay zeka kullanımıyla gerçekleştirme fikri ön plana çıkmıştır. Bu tez çalışmasının ana fikri de derin öğrenmenin bu sınıflandırma işlemini insanlardan daha hızlı ve daha başarılı sonuçlarla gerçekleştirebileceğidir.

Yapılan araştırmalar sonucunda, daha önceki çalışmalarda kullanılan modellerin sonuçları ve konu ile ilgili deneyler dikkate alınarak, evrimsel sinir ağları ve 26x65 boyutundaki mel frekans kepsral katsayıları kullanılarak sınıflandırma işleminin gerçekleştirilmesine karar verilmiştir. Çalışma boyunca evrimsel sinir ağı hiper-parametrelerinin sınıflandırma başarımına etkisi doğruluk, kesinlik, f skoru gibi değerlendirme metrikleri açısından ele alınmıştır. GTZAN veri setinin sınıflandırma başarımı açısından tür bazında değerlendirilmesi gerçekleştirilmiş ve çıkarımlarda bulunulmuştur. Geliştirilen evrimsel sinir ağı modeli, GTZAN veri setine uygulanan ve birçok özelliğin bir arada kullanıldığı daha komplike modeller ile kıyaslanabilecek seviyede bir doğruluk oranını yakalamıştır. Ek olarak, daha küçük verilerin kullanılması, daha az eğitim süresi ve daha düşük bellek kullanımı ile sonuçlanmıştır.

İlerleyen çalışmalarda, daha fazla tür ve parçayı içeren veri setlerinin oluşturulması ve kullanımı ile daha fazla sayıda müzik türünü daha doğru sınıflandırabilecek modeller geliştirilebilir. Ancak oluşturulacak veri setlerinin dengeli ve doğru bir biçimde oluşturulması ve uygun temsiller kullanılarak modellere verilmesi ile bu sonuç mümkün

olacaktır. Bu sayede müziğin öğrenme ve öğretme sürecinde daha fazla katkı sunacak modellerin ortaya çıkarılması sağlanabilir.

Müzik türlerini otomatik olarak sınıflandırabilecek modellerin geliştirilmesi ve günümüzde yaygın bir şekilde kullanıma sahip olan müzik uygulamalarının öneri sistemlerinde var olan algoritmalara destek olacak bir şekilde entegrasyonunun sağlanması ile bu sistemlerin verimliliği ve doğruluğu artırılabilir. Bu sayede kullanıcılara daha kullanışlı bir uygulama ara yüzü sunulması mümkün olacaktır.



KAYNAKLAR

1. Ersoy İ. (2017). Müzikte Tür Kavramı: Müzik türleri sınıflandırmasında yeni bir model önerisi. *EÜ Devlet Türk Musikisi Konservatuvarı Dergisi*, 11(1), 1-16.
2. Soykan Ö.N. (2012). Müzik nedir? Felsefî bir araştırma. *Doğu Batı*, 62(1), 29-42.
3. Stowell D., Dixon S. (2011, 26 October). *MIR in school? Lessons from ethnographic observation of secondary school music classes*. Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR), Miami, USA 347–352.
4. Perrot, D., Gierdigen, R. (1999). Scanning the dial: an exploration of factors in identification of musical style. *Proceedings of the society for music perception cognition*, Evanston, USA, 88.
5. Tzanetakis, G., Cook, P., (2002). Musical genre classification of audio signals. *IEEE Transaction on Speech and Audio Processing*, 10(1), 261-275.
6. Li, T., Tzanetakis, G. (2003, 19-22 October). *Factors in automatic musical genre classification of audio signals*. IEEE Workshop on Application of Signal Processing to Audio and Acoustic, New York, USA, 143-146.
7. Li, T., Ogihara, M., Li, Q. (2003, July). *A comparative study on content-based music genre classification*. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, 282-289.
8. Li, T., Ogihara, M. (2005, 23 March). *Music genre classification with taxonomy*. IEEE International Conference on Acoustics, Speech, and Signal Processing, Pennsylvania, USA, 197.
9. Lee, C., Shih, J., Yu, K., Lin, H. (2009). Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Transactions on Multimedia*, 11(4), 670-682.
10. Lidy, T., Rauber, A. (2005, 11-15 September). *Evaluation of feature extractors and psycho-acoustic transformations for music genre classification*. Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR), London, UK, 34-41.
11. Chang, K., Jang, J., Iliopoulos, C. (2010, 9-13 August). *Music genre classification via compressive sampling*. Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), Utrecht, Netherland, 387-392.
12. Humphrey, E., Bello, J. (2012, 12-15 December). *Rethinking automatic chord recognition with convolutional neural networks*. Proceedings of the 11th International Conference on Machine Learning and Application (ICMLA), Florida, USA, 387-392.
13. Zhang, W., Lei, W., Xu, X., Xing, X. (2016, 8-12 September). *Improved music genre classification with convolutional neural networks*. Proceedings of the Interspeech, San Francisco, USA, 3304-3008.

14. He, K., Zhang, X., Ren, S., Sun, J. (2016, 27-30 June). *Deep residual learning for image recognition*. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nevada, USA, 770-778.
15. Choi, K., Fazekas, G., Sandler M.B., Cho, K. (2017, 23-27 October). *Transfer learning for music classification and regression tasks*. Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 141-149.
16. Feng, L., Liu, S., Yao, J. (2017). Music genre classification with paralleling recurrent convolutional neural network. *arXiv*, arXiv:1712.08370.
17. Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv*, arXiv:1712.08370.
18. Nanni, L., Costa, Y.M.G., Aguiar, R.L., Silla, C.N., Brahnam, S. (2018). Ensemble of deep learning visual and acoustic features for music genre classification. *Journal of New Music Research*, 47(4), 383-397.
19. Lee, J., Park, J., Kim, K., Nam, J. (2018). SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification. *Applied Sciences*, 8(1), 150.
20. Nanni, L., Costa, Y.M.G., Lucia, D.R., Silla, C.N., Brahnam, S. (2017). Combining visual and acoustic features for audio classification tasks. *Pattern Recognition Letters*, 88(1), 49-56.
21. Sigtia, S., Dixon, S. (2014, 4-9 May). *Improved music feature learning with deep neural networks*. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 6959-6963.
22. Elbir A., (2020). *Akustik Özellikler Kullanan Müzik Öneri Sistemi*, Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 111.
23. Karatana, A., Yildiz, O. (2017, 15-18 May). *Music genre classification with machine learning techniques*. 25th IEEE Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 1-4.
24. Kızrak, M.A., Bolat, B. (2015, 16-19 May). *Classification of classic turkish music makams by using deep belief networks*. 23rd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey, 527-530.
25. Molau, S. (2001, 7-11 May). *Computing Mel-frequency Cepstral Coefficients on the Power Spectrum*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Utah, USA, 73-76.
26. Gupta, S., Jaafar, J., Ahmad, W.W., Bansal, A. (2013). Feature extraction using mfcc. *Signal&Image Processing International Journal (SIPIJ)*, 4(4), 101.
27. Yusnita, M., Paulraj, M., Yaacob, S., Yusuf, R., Shahrman, A. (2013). Analysis of accent-sensitive words in multi-resolution mel-frequency cepstral coefficients for classification of accent in Malaysian English. *International Journal of Automotive Mechanical Engineering*, 7(1), 1053-1073.

28. Goodfellow I, Bengio Y, Courville A. (2016). *Deep Learning* (First Edition). Cambridge, MA: MIT Press, 1-26.
29. Chang, S., Cohen, T., Ostdiek, B. (2018). What is the machine learning?. *Physical Review D*, 97(5), 1-6.
30. Jain, A.K., Mao, J. (1996). Artificial neural networks: A tutorial. *IEEE Computer*, 29(3), 31-44.
31. İnternet: Delua J. (2015). Supervised vs. Unsupervised Learning: What's the Difference?. Web: <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning> Son Erişim Tarihi: (21.01.2022)
32. Basheer, I. A., Hajmeer, M. (2000). Artificial neural networks: Fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1), 3–31.
33. İnternet: Uzm. Alpay (2021). Yapay Sinir Ağları Nasıl Oluşur?. Web: <https://www.psikolojik.gen.tr/yapay-sinir-aglari.html> Son Erişim Tarihi: (20.01.2022)
34. İnternet: Tuzen E. (2017). Yapay Sinir Ağlarının Çalışma Prensibi. Web: <https://yapayzeka.ai/yapay-sinir-aglarinin-calisma-prensibi/> Son Erişim Tarihi: (22.01.2022)
35. Bottou, L. (2010, 22-27 August). *Large-scale machine learning with stochastic gradient descent*. Proceedings of 19th International Conference on Computational Statistics (COMPSTAT), Paris, France, 177-186.
36. Kingma, D., Ba, J. (2015, 7-9 May). *Adam: A method for stochastic optimization*. International Conference on Learning Representations, California, USA, 12.
37. İnternet: Jordan J. (2018). Setting the Learning Rate of Your Neural Network. Web: <https://www.jeremyjordan.me/nn-learning-rate/> Son Erişim Tarihi: (23.01.2022)
38. Boser, B., Cun, L., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2(1), 396-404.
39. Zang, H., Cheng, L., Ding, T., Cheung, K.W., Liang, Z., Wei, Z., Sun, G. (2018). Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network. *IET Generation, Transmission & Distribution*, 12(20), 4557–4567.
40. İnternet: Khvatova K. (2019). Is Convolutional Neural Network a Black Box? Not Anymore. Web: <https://medium.com/@kristinakhvatova/is-convolutional-neural-network-a-black-box-not-anymore-caca1429952d> Son Erişim Tarihi: (21.01.2022)
41. Balocco, S. (Editor). (2020). *Intravascular Ultrasound*. E-book: Elsevier Imprint, 141-158.
42. Yani, M. (2019). *Application of transfer learning using convolutional neural network method for early detection of terry's nail*. Journal of Physics: Conference Series, 1201(1), 12052.

43. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
44. İnternet: Ampadu H. (2021). Dropout in Deep Learning. Web: <https://ai-pool.com/a/s/dropout-in-deep-learning> Son Erişim Tarihi: (23.01.2022)
45. Sturm, H.A. (2013). The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint, arXiv:1306.1461*.





EKLER

EK-1. Önerilen CNN modeline ait kaynak kodu

```

#Model oluşturma
model = keras.Sequential()
#1st Layer
model.add(keras.layers.Conv2D(128, (3, 3), activation='relu', input_shape=X_train.shape[1:],
                             kernel_regularizer=keras.regularizers.l2(0.001)))
model.add(keras.layers.Conv2D(128, (3, 3), activation='relu', kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.AveragePooling2D((3, 3), strides=(2, 2), padding='same'))
#model.add(keras.layers.Dropout(0.1))
model.add(keras.layers.BatchNormalization())
#2nd Layer
model.add(keras.layers.Conv2D(128, (3, 3), activation='relu', kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.AveragePooling2D((3, 3), strides=(2, 2), padding='same'))
#model.add(keras.layers.Dropout(0.1))
model.add(keras.layers.BatchNormalization())
#3rd Layer
model.add(keras.layers.Conv2D(256, (4, 4), activation='relu', kernel_regularizer=keras.regularizers.l2(0.0001)))
model.add(keras.layers.GlobalAveragePooling2D())
#model.add(keras.layers.Dropout(0.1))
model.add(keras.layers.BatchNormalization())
#flatten layer
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(256, activation='relu'))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(128, activation='relu'))
model.add(keras.layers.Dropout(0.5))
#output layer
model.add(keras.layers.Dense(10, activation='softmax'))
reduce_lr = ReduceLRonPlateau(monitor='loss', factor=0.1,
                              patience=3, min_lr=0.000001)
#Optimizer
optimizer = keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history=model.fit(X_train, y_train, validation_data=(X_validation, y_validation), callbacks=[reduce_lr],
                 batch_size=32, epochs=80)

```



GAZİ GELECEKTİR..