KTO KARATAY
ÜNİVERSİTESİ

T.C.
KTO Karatay Üniversitesi
Fen Bilimleri Enstitüsü

ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
TEZLİ YÜKSEK LİSANS PROGRAMI

SEÇİCİ DERİN OTOKODLAYICILAR İLE SIRALI SES KAYNAKLARININ SEGMENTASYONU

Meryem Betül ÖZKARDAŞ

KONYA

Aralık 2018

SEÇİCİ DERİN OTOKODLAYICILAR İLE SIRALI SES KAYNAKLARININ SEGMENTASYONU

Meryem Betül ÖZKARDAŞ

KTO Karatay Üniversitesi Fen Bilimleri Enstitüsü

Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalı
Yüksek Lisans Programı

Yüksek Lisans Tezi

Aralık, 2018

Fen Bilimleri Enstitü Onayı

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Hüseyin Bekir YILDIZ

Bu tezli yüksek lisans tezinin yapılması gereken bütün gerekliliklerinin yerine getirdiğini onaylıyorum.

Anabilim Dalı Başkanı
Dr. Öğr. Üyesi H. Oktay ALTUN

Meryem Betül ÖZKARDAŞ tarafından hazırlanan SEÇİCİ DERİN OTOKODLAYICILAR İLE SIRALI SES KAYNAKLARININ SEGMENTASYONU başlıklı bu çalışma 12.12.2018 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jüri tarafından tezli yüksek lisans tezi olarak kabul edilmiştir.

Tez Danışmanı
Dr. Öğr. Üyesi H. Oktay ALTUN

Jüri Üyeleri

Başkan: Doç. Dr. Murat CEYLAN

Üye: Dr. Öğr. Üyesi H. Oktay ALTUN

Üye: Dr. Öğr. Üyesi Semih YUMUŞAK

Tez Bildirimi

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orjinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu bildirir aksi bir durumda aleyhime doğabilecek tüm hak ve kayıplarını kabullendiğimi beyan ederim.

Aralık-2018

Meryem Betül ÖZKARDAŞ

Özet

SEÇİCİ DERİN OTOKODLAYICILAR İLE
SIRALI SES KAYNAKLARININ SEGMENTASYONU

Meryem Betül ÖZKARDAŞ

KTO Karatay Üniversitesi,
Fen Bilimleri Enstitüsü,
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi H. Oktay ALTUN

Aralık 2018

Ses kaynaklarının ardışık biçimde kaydedildiği senaryolarda, bir ses kaynağının seçilip, diğer kaynakların silinmesi işini yapabilecek bir teknik geliştirdik. Bir derin otokodlayıcı mimarisini, bir ses kaynağını geçirirken, diğer bir kaynağı silecek şekilde eğittik, ve bu tekniğe seçici otokodlayıcı ismini verdik. Geliştirdiğimiz metodu, Türk klasik müziği enstrümanlarının (sanatçıların birinin çalıp diğerinin dinlediği ve sıralı şekilde seslerin kaydedildiği durumlar için), ardışık insan seslerinin ve ardışık hayvan seslerinin segmentasyonunda kullandık. Metot genel manada yarı çift yönlü haberleşmenin tek bir alıcıyla kaydedildiği durumlarda haberleşme kanallarından birini diğerlerinden izole etmekte kullanılabilir.

Anahtar kelimeler: Seçici Derin Otokodlayıcı, Ses Kaynağı Segmentasyonu, Sıralı Ses Kaynağı Ayrıştırma

Abstract

AN INTERLEAVED AUDIO SOURCE SEGMENTATION TECHNIQUE VIA DEEP AUTOENCODERS

Meryem Betül ÖZKARDAŞ

KTO Karatay University,
The Graduate School of Natural and Applied Sciences,
Master of Science Thesis in Electrical and Computer Engineering

Advisor: Asst. Prof. H. Oktay ALTUN

December 2018

In this thesis, we devised a technique for segmentation and isolation of a particular sound source from an interleaved audio source. We trained a deep auto-encoder architecture in a way to output desired signal source intact but suppress others by outputting zero. We tested our method in order to segment Turkish classical music instruments, male/female voices and animal voices. In general sense, the method can be utilized in several half-duplex communication scenarios where isolating a communication channel is desirable.

Keywords: Sound Source Segmentation, Selective Deep Autoencoder, Interleaved Sound Source Separation

Teşekkür

Bana olan sevgi ve saygısına minnettar olduğum sabırlı eşim Ebubekir ÖZKARDAŞ'a, tez çalışmam boyunca vaktinden fedakarlık eden 11 aylık oğlum Ali Mirza ÖZKARDAŞ'a, tüm zamanların en fedakar anneannesi kıymetli annem Candan DURAN'a, her zaman sırtımı yaslayabildiğim babam Mehmet DURAN'a ve en iyi arkadaşlarım kardeşlerime teşekkürlerimi sunuyorum.

Çalışmalarım boyunca bana destek olan arkadaşlarım ve meslektaşlarım Büşra MUTLU İPEK'e ve Esra Betül KEŞÇİ'ye, tezin yazımında ve programlamada kısmi katkıları olan Cihan ÇALIŞIR'a, Dervişe GÖKALP'e ve Rümeysa TATAR'a teşekkür ederim. Sayesinde öğrenmeyi, araştırmayı ve kendimi keşfetmeyi öğrendiğim, tez çalışmamın başından sonuna kadar desteğini eksik etmeyen, biz öğrencilerine değer verdiğini her zaman hissettiren kıymetli tez danışmanım Dr. Öğr. Üyesi H. Oktay ALTUN'a minnetimi ve şükranlarımı sunarım.

Meryem Betül ÖZKARDAŞ
Aralık-2018

İçindekiler

Şekil Listesi

Tablo Listesi

Kısaltmalar

Kısaltmalar	Açıklama
DSA	Derin Sinir Ağları
DNN	*Deep Neural Networks*
ICA	*Independent Component Analysis*
NTF	*Non-negative Tensor Factorization*
NMF	*Non-negative Matrix Factorization*
YSA	Yapay Sinir Ağları
RNN	*Recurrent Neural Network*
KSA	Konvolüsyonel Sinir Ağları
NLP	*Natural Language Processing*
KSA	Konvolüsyonel Sinir Ağları
ALS	*Alternating Least Squares*
STFT	*Short Time Fourier Transform*
SGD	*Stochastic Gradient Descent*
MSE	*Mean Square Error*
DCT	*Discrete Cosine Transform*
AE	*Autoencoder*
DC	*Direct Current*
AC	*Alternating Current*
k-NN	*k-Nearest Neighbors*

Semboller

Semboller	Açıklama
$\sum$	Toplam sembolü
x_i	Giriş matrisi
W_i	Sinir ağındaki ağırlık
θ_i	Eşik değeri
φ	Aktivasyon fonksiyonu
n	Girdi sayısı
y_i	Maliyet fonksiyonundaki gerçek değer
y_i^p	Maliyet fonksiyonunda tahmin değeri

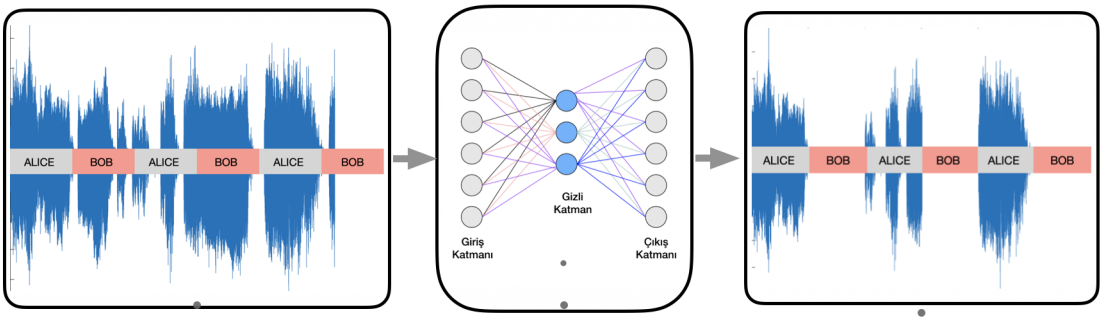
1 Giriş

Ses segmentasyonu, ardışık şekilde kaydedilmiş ses kaynaklarının homojen kısımlarına ayrıştırılması olarak ifade edilebilir. Bu homojenliğin kaynağı cinsiyet, enstrüman veya insan sesi, canlı seslerindeki farklılık ve buna benzer birçok doğal gerekçeyle oluşabilir. Segmentasyonda genellikle amaç benzer özellikteki sinyallerin gruplandırılarak farklı sinyal işleme veya sınıflandırma algoritmalarına tabi tutulmasıdır. Ses segmentasyonu otomatik transkripsiyon, yayın haberlerinde bölümleme, otomatik müzik analizi, stil tanıma, konuşmacı tanıma, otomatik konuşma tanıma ve buna benzer birçok uygulamada kullanılabilir.

Bunun yanısıra kişisel bilgilerin saklanmasının hassas olduğu senaryolarda veya özel hayatın mahremiyetinin korunmasının gerekli olduğu durumlarda da ses segmentasyonu ve ardışık kaynakların ayrıştırılması önemlidir. Örneğin Alice ve Bob aynı ortamda sırasıyla konuşan iki kişi olsun. Alice ve Bob'un konuşmaları tek bir ses kayıt cihazından kaydediliyor olsun. Alice veya Bob'un ses kayıtlarından sadece bir tanesini otomatik biçimde seçerek diğer kişinin kaydını silmemizi gerektirecek durumlar olabilir: Bob'un sesi güvenlik sebebiyle gizlenirken Alice'in konuşmaları kamu otoriteleri, basın gibi üçüncü kişilere verilmek istenebilir. Diğer bir senaryoda ise makine öğrenme algoritmaları için tek bir kişinin sesi, diyalogdan çekilerek, tek bir kişiye ait sesle ulaşmak istenebilir. Çok büyük ses veri bankalarından, bir ses kaynağına ait seslerin izolasyonu maksadıyla da değerlendirilebilir.

Bu tez çalışmasında derin öğrenme tekniklerini kullanarak ses kaynaklarının segmentasyon problemini çözebilecek bir metodoloji öneriyoruz. Birbirine karışmış iki ses kaynağını ayrıştırabilmek için *otokodlayıcı* (AE) adı verilen özel bir derin öğrenme mimarisi kullanıyoruz. AE, girdi olarak verilen ses veya görüntü dosyasını aynı şekilde çıktı olarak veren bir mekanizmaya sahiptir. Ancak biz modelimizde, literatürdeki örneklerden farklı olarak sadece bir kaynağı AE ile eğiterek öğretmek yerine, hedef sesi bulmak için mimariye karışan hedef ses dışındaki ses kaynaklarını reddederek sıfırlayacak şekilde eğitiyoruz. Şekil 1.1'de Alice ve Bob'un aynı anda konuştuğu iki farklı karışık sesten Alice'n sesini ayıran ve diğer ses olan Bob'un sesini sıfırlarak bu sesi reddeden bir oto kodlayıcı model geliştiriyoruz. Böylelikle

giriş katmanında verilen kaynakları ayrıştırmayı hedefliyoruz.



Şekil 1.1: Alice ve Bob'a ait aralıklı karışım seslerinden Alice'in sesinin otokodlayıcı derin öğrenme modeli ile ayrılması

Bölüm 2'de, bu çalışma ile ilgili eş zamanlı veya farklı zamanlara ait araştırmacıların yapmış oldukları araştırmalardan ve çözüm önerilerinden bahsedilmiştir. Bölüm 3'de çalışmada ihtiyaç duyulan teknolojik yaklaşımların anlaşılabilmesi için derin öğrenme yaklaşımları hakkında temel bilgilere yer verilmiştir. Bölüm 4'te, ses kaynağının hangi uygulamalarla segmente edildiği, bu ayrışma esnasında derin öğrenme için hangi yöntemler ile sonuca ulaşıldığından bahsedilmiştir. Bölüm 5'te, literatürde uygulanan yöntemler ve bu tez çalışmasında elde edilen sonuçların karşılaştırılması ile birlikte önerilere yer verilmiştir. Bölüm 6'da ise bu tez çalışması için geliştirilen programın kaynak kodları sunulmuştur.

2 Literatür

Ses kaynağı bölümlemesi, bir müzik dosyasından saf sesli enstrüman ayrımı veya bu tez çalışmasının da konusu olan art arda veya iç içe seslerin olduğu bir ortamdan, tek bir sesin ayrıştırılması gibi konuların üzerinde çalışıldığı güncel bir alandır [1]. Çalışmalar henüz insan işitsel sistemi kadar başarılı olamasa da her geçen gün daha iyi sonuçlar alınmaktadır. Ses üzerindeki çalışmalara destek olacak bu tez, içinde bulunduğumuz zamanda güçlü bir potansiyele sahiptir.

Konuşma ayrımı (*speech separation*) ve konuşma kümeleme, son yıllarda kapsamlı bir çalışmanın konusu olmuştur. Wang ve ark. [2], konuşma ayrımını, derin öğrenme ile çözmeye çalışmışlardır. Hershey ve ark. [3], farklı kaynakların kümelenmesi ve bölümlenmesi için ayrımcı olarak eğitilmiş konuşma yerleştirmelerinin kullanıldığı, derin kümeleme denen bir yöntem önermektedir. Bunu yaparken, permütasyon içermeyen bir yöntem geliştirmek istediler ancak doğru sonuç elde edemediler. Farklı bir çalışmada ise, tek bir sesten iki ayrı konuşmacı ayrımı yapmak için etiket permütasyon problemini çözen araştırmacılar; Işık ve ark. [4] ve Yu ve ark. [5] bir DSA'yı eğiterek etiket permütasyon problemini başarıyla kullanan yöntemler tanıtmışlardır.

İdeal ikili maske (*ideal binary mask*), Narayanan ve ark. [6] ve Wang ve ark. [7], derin sinir ağlarını kullanarak iki aşamalı bir çerçeve önermişlerdir. İlk aşamada, araştırmacılar her bir çıktı boyutunu ayrı ayrı tahmin edebilmek için derin sinir ağlarını kullanırlar. İkinci aşamada ise birinci aşamadan elde edilen tahminleri filtrelemek için bir sınıflandırıcı (SVM) kullanılır. Bununla birlikte, önerilen çerçeve, çıktı boyutu yüksek olduğunda ölçeklendirilemez. Huang ve ark. [8], tüm özellik boyutlarını aynı anda tek bir sinir ağını kullanarak birlikte öngörebilen genel bir çerçeve tasarlamışlardır. Tahmin çıktıları sıklıkla zaman frekansı maskeleme işlevleri ile düzleştirildiği için, maskeleme işlevini ağlarla ortak olarak eğitmeyi hedeflemişlerdir.

Tekrarlayan sinir ağı (*Recurrent Neural Network (RNN)*), Maas ve diğ. [9], sağlam otomatik konuşma tanımada, konuşma gürültüsünün azaltılması için bir RNN

kullanılmasını önerdiler. Gürültülü bir sinyal verildiğinde araştırmacılar, temiz konuşmayı öğrenmek için bir RNN uygularlar. Kaynak ayırma senaryosunda, anlamlandırma çerçevesindeki bir hedef kaynağın doğrudan modellenmesinin, tüm kaynakları modelleyen çerçeveye kıyasla optimal olmadığını bularak, maskeleme ve ayrımcılık eğitimini gerçekleştirmek için farklı tahmin çıktılarından gelen bilgileri ve kısıtlamaları kullandılar. Huang ve ark. [10] ise tek bir sesin ayrıştırılması için, derin öğrenme modellerini (DNN-deep neural networks , RNN) kullanarak yumuşak maskeleme fonksiyonu ve en iyi model için optimizasyon yaklaşımı ile geliştirmişlerdir. Bir kadın ve bir erkekten oluşan tek ses dosyasını ayrıştırmak için TIMIT corpus yöntemini kullanmışlardır.

Negatif olmayan matris ayrıştırması (*non-negative matrix factorization*), Lee ve Seung [11] tarafından önerilen NMF algoritmaları, gözlem matrisini ve model arasındaki rekonstrüksiyon hatasını en aza indirerek, matrisleri giriş olmayan ve negatif olmayan olarak sınırlayarak ayrışmayı yapmışlardır. Doğal ses kaynaklarının zaman-frekans gösterimleri genellikle seyrektir, yani çerçevelerin ve frekansların çoğu aktif değildir. Araştırmaya göre seyrek spektrogramlar, her biri tek bir ses kaynağının parçalarını temsil eden, negatif olmayan bir ayrışmaya sahiptir. FitzGerald ve ark. [12] ise ek olarak, kümeleri bir k-en yakın komşular yaklaşımı (kNN) (k-nearest neighbours approach) kullanarak oluşturmuştur. Bu yaklaşımla, algoritmanın otomatik olarak kaynak ayrıştırmasını gerçekleştirebileceğini göstermişlerdir. Keyder [13] tez çalışmasında, NTF'nin en iyi kullanım algoritmasının değişimli en küçük kareler algoritması (alternating least squares, ALS) olduğunu belirtmiştir. Bu algoritma, alfa ve beta olarak bilinen maliyet fonksiyonlarının kullanılarak oluşturulduğu alfa ve beta algoritmalarıdır. Her iki algoritmanın ayrıştırma sonucu, farklı bir çok teste göre denenmiştir.

Bağımsız bileşen analizi (*(ICA) Independent Component Analysis)*, rastgele değişkenler, ölçümler veya sinyal kümelerinin altında yatan gizli faktörleri ortaya çıkarmak için istatistiksel ve hesaplamalı bir tekniktir. Mitianoudis [14], doktora tezinde ses kaynağı ayrıştırmak için ICA yöntemini kullanmıştır.

Seyrek kodlama (*sparse coding*), seyrek olmayan negatif kaynaklar için, kaynakların elde edilebileceği bir çarpım matrisi yoktur. Bunun yerine Virtanen [15], yapılan varsayımlar altında en uygun kaynakları bularak belirli bir ayırma algoritması geliştirilmiştir. Algoritma, Hoyer (2002) tarafından seyrek kodlama ile birleştirilerek, negatif olmayan matris faktorizasyonundan Lee ve Seung'dan [11] alınan fikirler kullanılarak tasarlanmıştır.

Otokodlayıcılar (*autoencoder*) son yıllarda, ses dosyalarının ayrımı için kullanılan popüler yaklaşımlardan bir tanesi haline gelmiştir. Bu tez çalışmasında kullanılan segmentasyonun yanında, Lu ve ark. [16] tarafından gürültü azaltma ve konuşma

iyileştirme için kullanılmıştır. Girdi olarak verilen gürültülü ve temiz ses sinyalleri ile modeli eğiterek, büyük veri setleri ile test etmişlerdir. Seslerin gürültüden tam olarak ayrılmaması problemine çözüm olarak Kameoke ve ark. [17] çok kanallı otokodlayıcı yöntemini denemişlerdir. Otokodlayıcıların birçok avantajı olsa da, yüksek karmaşıklı yapılarda verilerin kaybıyla karşılaşılabilir. Optimize edilerek çözülmek istense de, optimize edilecek çok fazla parametre olması sebebiyle sistem oldukça karmaşıktır. Tez çalışmamızda geliştirdiğimiz yöntem, art arda devam eden ses kaynaklarından istenilen kaynağı çıkarıp diğerlerini sıfırlayan otomatik kodlayıcıdır. Verilen girdi ile çıktıyı eşleyerek aralarındaki ilişkiyi öğrenen kodlayıcı çeşitli amaçlar için de kullanılabilir.

Ses içeriği analizi uygulamalarının iki bölümde incelendiğini kabul edersek: Birincisi, bir ses akışının homojen segmentasyonu ve ikincisi bir konuşma akışını farklı hoparlörlerle bölümlere ayırmaktır. Foote [18], bir ses veya müzik dosyasında değişiklik olan noktaları otomatik olarak bulmak için bir yöntem geliştirmişlerdir. Modellemek için sinyali kullanmıştır. Bu nedenle belirli akustik ipuçlarına ihtiyaç duymamış ve modeli eğitmemiştir. Lu ve ark. [19], 2 saatlik eğitim verilerini kullanarak, bir ses akışını farklı ses türlerine ayırmışlardır. Burada, sınıflandırıcı destek vektör makineleri ve doğrusal spektral çiftler ile vektörel entegrasyona sahip en yakın komşular yöntemi kullanılmıştır. Janku ve Hyniova [20], MMI tarafından denetlenen ağaç tabanlı vektör niceleyici ve ileri beslemeli sinir ağının, çevresel sesleri ve konuşmayı tanımak ve bölümlemek için bir ses akışında kullanılabileceğini öne sürdüler. Seslerin ayrılma problemini Yadav [21], bir ses akımını sesli bölgeler bazında bölümlere ayırarak çözmüştür.

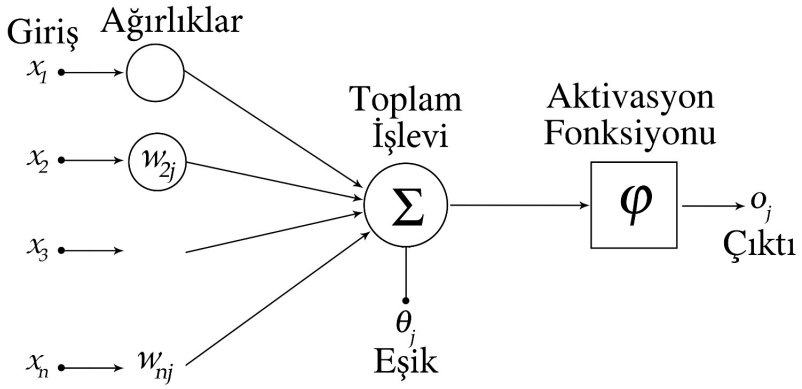
3 Kuramsal Temeller

Bu bölümde, tez çalışmasının araştırma, geliştirme ve yazım aşamasında faydalanılan konu başlıklarına değinilmiştir. İnsan ve makine etkileşiminin bir ürünü olan yapay zeka yaklaşımının, zaman içerisinde insan hayatı için vazgeçilmez olması ve yaşamı kolaylaştıracağı kaçınılmazdır. Yapay zeka içerisinde bi yaklaşım olan Doğal Dil İşleme (NLP), doğal dillerin yapısının çözümlenerek bilgisayar ortamında anlaşıldığı veya yeniden üretildiği bir sistemdir. NLP'nin uygulama alanlarının geliştirilmesi birçok donanımsal ürünün ortadan kalkmasına sebep olabilir. Ses komutu algılayan yazılımların gelişmesi ile, klavye ve fare gibi aygıtların kullanımları azalabilir.

NLP'nin bilim dünyasında ve günlük hayattaki kullanım alanları belgelerin otomatik çevrilmesi, soruları yanıtlayabilen makineler, otomatik konuşan ve komut algılayan makineler, konuşmaların süzgeçten geçirilmesi, yeniden konuşma üretilmesi ve metinlerin otomatik çözümlenmesi gibi birçok konu ile örneklenebilir.

3.1 Yapay Sinir Ağları

Yapay sinir ağları (YSA); en kısa tanımıyla insan beyninin çalışma ve düşünebilme yeteneğinden yola çıkılarak, biyolojik sinir ağlarını taklit eden bir yapıdır [22]. Her biri kendi hafızasına sahip işlem elemanlarının birbirleriyle kurdukları bilgi işleme yapılarıdır.



Şekil 3.1: Yapay sinir ağları yapısı

YSA'nın en önemli özelliği, farklı yerlerden elde ettiği verileri analiz ederek ve karşılaştırarak, hiç görmediği örnekler üzerinde uygulayıp yeni bir karar verebilmesidir. Bu yapay öğrenme yaklaşımı, bilim alanında birçok problemin çözümünü hızlandırmaktadır.

Tablo 3.1: Yapay sinir ağları ve biyolojik sinir ağları karşılaştırması

Biyolojik Sinir Sistemi	**Yapay Sinir Sistemi**
Nöron	İşlem Elemanı
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

YSA günümüzde birçok alanda kullanılmaktadır; ses tanıma, örüntü tanıma, üretim sistemlerinin optimizasyonu, mekanik parçaların verimli kullanım sürelerinin tahmini, resim işleme, spam maillerin filtrelenmesi, kan analizi, insan beyninin modellenmesi, parmak izi ile tanınma, araçların otomatik denetlenmesi, düşünebilen araçlar ve robotlar için en iyi rotanın belirlenmesi, lineer olmayan denetim alanları, hava durumunun yorumlanması, karakter el yazısı tanıma, hastalıkların teşhisi ve tedavi önerileri, radar algılanması, veri madenciliği olarak örnekler verilebilir [23].

YSA, katmanlı bir yapıya sahiptir. Yapay nöronlar katmanlar içinde yapılandırılmıştır. Giriş katmanı, gizli katman ve çıkış katmanından oluşmaktadır. Her katman, önceki katmandaki verileri alarak, ürettiği katmanı sonraki katmana aktarır. Bu katmanlı yapı ne kadar çoksa, model o kadar derin bir yapıya dönüşür. Giriş ve çıkış katmanının arasındaki gizli katman, verilerin işlendiği katmandır.

3.1.1 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, hücreye gelen girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Seçilen aktivasyon fonksiyonu kurulan ağın performansını önemli ölçüde etkiler. *Sigmoid* fonksiyonu, yapay sinir ağlarında kullanılan sürekli ve türevlenebilir bir fonksiyondur. *Sigmoid* fonksiyonu sıfırla bir arasında olasılık değerleri ürettiğinden son katmanda daha tutarlı sonuçlar almak için kullanılır. Bu özelliği ile ortalama kare hesabı kayıp fonksiyonu ve Adam optimizasyon fonksiyonu ile verdiği sonuç, diğer aktivasyon fonksiyonlarınkinden daha iyidir.

$ReLU$ en yaygın kullanılan aktivasyon fonksiyonlarından biridir. $ReLU$'nun yararları ise, sadece birleşme sürecini hızlandıracak olan, pozitif ve negatif değerlerin geçmediği değerlere izin verir ve ölü bir nöronun ortaya çıkma olasılığını ortadan kaldırır ya da azaltır. Ölü nöron, kullanılmayan bilgiler olarak tanımlanabilir. $ReLU$ fonksiyonu negatif girdiler için 0 değerini alırken, x pozitif girdiler için x değerini almaktadır.

3.2 Derin Öğrenme

Derin öğrenme, makine öğrenmesi içerisinde verileri birden fazla özellik seviyesinde inceleyen bir yaklaşımdır. Bu seviyeler belli bir hiyerarşi içerisinde, üst katmanların alt katmanlardan öğrenerek türediği yapılardır [24]. Burada kullanılan katman sayısı ne kadar olursa derinlik aynı oranda artar ve öğrenmenin çok katmanlı yapılarda otomatik olarak gerçekleşmesi hedeflenir. Özellikle, büyük verilerden ihtiyaç duyulan özelliklerin saptandığı sistemler oluşturmak için YSA katmanlarından farklı olarak çok katmanlı derin sinir ağların kullanılması olarak adlandırılan Derin Öğrenme yaklaşımlarının geliştirilmesi oldukça önemlidir.

Facebook, Apple, Microsoft ve Google gibi teknoloji firmaları son yıllarda Derin Öğrenme üzerine araştırmalarını genişletmişlerdir. Facebook, face.com'u satın alarak, fotoğraf üzerinden yüz tanıma özelliğini artırmayı hedeflemiştir. Novauris şirketini kendi bünyesine dahil eden Apple, SİRİ üzerinde çalışmalarını artırmıştır. Microsoft, SwiftKey şirketini satın alarak, 100 dil için çalışan klavye uygulaması geliştirmeyi planlamıştır. Google bünyesinde bulunan Deep Mind firması ile derin öğrenme araştırmalarını sürdürmektedir [25].

Derin öğrenmenin bilinen uygulama alanları arasında sınıflandırma, görüntü işleme, video işleme, sinyal işleme ve bu çalışma içerisinde de ihtiyaç duyduğumuz doğal dil işleme gibi konular yer almaktadır.

3.2.1 Otokodlayıcı

Otokodlayıcı, girdi olarak verilen değerleri çıktı katmanında tekrar oluşturan denetimsiz bir öğrenme modelidir. Mimari olarak otomatik kodlayıcılar; bir girdi katmanı, bir çıkış katmanı ve bu iki katmanı birbirine bağlayarak öğrenme işlemini gerçekleşiren gizli katmanlardan oluşmaktadır. Otokodlayıcılar, ilk olarak 1986'da geri yayılım problemini kapsamlı şekilde inceleyen bir makalede belirtilmiştir [26]. Bu fikrin dikkat çekmesiyle, sonraki yıllarda daha fazla araştırma makalesine konu olmuştur. Günümüze kadar gelişerek ortaya çıkan bu öğrenme algoritması, gizli düğüm parametrelerinin rastgele oluşturulduğu ve çıktı ağırlıklarının hesaplandığı yapı haline gelmiştir. Böylece geri yayılımdan daha hızlı bir şekilde öğrenen aşırı öğrenme makinesi şeklinde adlandırılmıştır.

Otokodlayıcılar, etiketli eğitim veri setini kullanan, doğrusal olmayan boyut düşürme yöntemi ile çok katmanlı bir derin öğrenme sinir ağıdır. Otokodlayıcılar giriş katmanında bir dizi $\{x^{(1)}, x^{(2)}, x^{(3)}..., x^{(n)}\}$ n boyutlu x giriş verisini alarak, y, $\{y^{(1)}, y^{(2)}, y^{(3)}..., y^{(n)}\}$ n boyutlu çıkış verisini oluşturan bir denetimli öğrenme yöntemidir [27]. Bu sinir ağı yapısı, giriş katmanı, gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanından gizli katmana doğru orta katmanlarda nöron sayısı azalmaktadır. Giriş katmanındaki nöron sayısını, çıkış katmanındaki nöron sayısına uygun şekilde yapılandırmak amacıyla boyut düşürme yapılmaktadır [28]. Otokodlayıcılar 4 adet parametreden oluşurlar:

Kod Büyüklüğü; orta katmandaki düğümlerin az sayıda olması daha çok veri sıkıştırma anlamına gelmektedir. Katman sayısı; otokodlayıcılar istenilen derinliğe sahip olabilir. Katmanlardaki Düğüm sayısı; otokodlayıcılar, her katmanı orta katmana doğru azalan düğümlere ve orta katmandan çıkışa doğru aynı oranda katmanların giderek arttığı düğümlere sahiptir. Kod çözücü, kodlayıcının simetriğidir. Kayıp fonksiyonu; Kayıp fonksiyonu olarak ortalama kare hata (mean squarred error) kullanılabilir.

3.3 Ayrık Kosinüs Dönüşümü (DCT)

Ayrık kosinüs dönüşümü (DCT), ayrık Fourier dönüşümünün (DFT) özelliklerine sahip olan spektral bir dönüşümdür. DCT'nin kullanımı çeşitli dalga sayılarının sadece kosinüs fonksiyonlarını temel fonksiyonlar olarak kullandığı ve gerçek değerli sinyaller ve spektral katsayılar üzerinde çalıştığı için bazı alanlardaki kullanımı açısından bakıldığında DFT'den popülerdir.

DCT'ler, bilim ve mühendislik alanındaki, küçük boyuttaki yüksek frekanslı bileşenlerin atılabileceği kayıp ses ve görüntülerin sıkıştırılmasından kısmi diferansiyel denklemlerin sayısal çözümü için spektral yöntemlere kadar birçok uygulama için önemlidir. Dekor korelasyonu, enerji sıkıştırma, onarılabilirlik, simetri ve diklik gibi özelliklere de sahip bir dönüşümdür. DCT, sinyalin enerjisini, sinyal kalitesini düşürmeden sinyalin boyutunu küçültme seçeneği sunan düşük frekans bölgelerine paketler ve enerji sıkıştırma özelliği sayesinde yüksek korelasyonlu sinyaller için mükemmel enerji sıkıştırma gösterir.

DCT sinyal dönüşümü komşu değerler arasındaki gereksizliğin kaldırılması gibi avantajlara sahiptir. Bu, bağımsız olarak kodlanabilecek ilişkisiz dönüşüm katsayılarına yol açar. DCT katsayılarından orijinal sinyalin yeniden inşası ise ters ayrık kosinüs dönüşümü (IDCT) olarak adlandırılır. Bu çalışmada ise ses sinyalleri üzerinde çalıştığımız için, bir 1-D dizisinin DCT'sine, özellikle DCT-II'ye ve bunun tersi olan IDCT-II'ye odaklanmaktayız.

En yaygın tek boyutlu DCT bir sinyalin uzunluk dizisi N olarak tanımlanmış olup; DCT denklemi, Denklem 3.1'de, IDCT denklemi ise denklem 3.2'de görülmektedir. Denklem 3.3'da ise DCT ve IDCT'de kullanılanılmakta olan *w(k)* fonksiyonu tanımlanmaktadır.

$$y(k) = w(k) \sum_{k=1}^{N} x(k) cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), k = 1, 2..., N \tag{3.1}$$

$$x(k) = \sum_{k=1}^{N} w(k) y(k) cos\left(\frac{\pi(2n-1)}{2N}\right), k = 1, 2..., N \tag{3.2}$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases} \tag{3.3}$$

Bu nedenle, ilk dönüşüm katsayısı, numune dizisinin ortalama değeridir. Bu değere doğru akım (DC) Katsayısı denir. Diğer tüm dönüşüm katsayıları alternatif akım (AC) Katsayıları olarak adlandırılır.

3.4 Segmantasyon

Ses akışını homojen olarak ayırmak için kullanılır. Amaç farklı nitelikteki bölgeleri ele almaktır. Ses bölümleme farklı ses tiplerinde kullanılır. Müzik ve gürültüyü, kadın sesi ve erkek sesini, konuşma ve sessizlik olarak örnek verilebilir. Segmentasyonun farklı kullanım alanları bulunmaktadır. Bir kullanıcının duymak istediklerini hızlıca bulmasına izin verir, algısal segmentasyon tekniği ve etkileşimli dinleyici kontrolü ile uygulanır. Diğer bir kullanım alanı ise gerçek zamanlı yayın haberleri transkripsiyonu ve konuşmacı kimliği belirlemek amacıyla kullanılır. Kullanım alanlarını belirlerken farklı segmentasyon metotları vardır. Ve her bir metodun farklı özellikleri bulunmaktadır. Enerji tabanlı bölümlendirme; model tabanlı bölümlendirme, metrik tabanlı bölümlendirmedir. Enerji tabanlı bölümlendirme de ses akışındaki sessizlik kısımlarını algılamak için ses enerjisini ölçüp eşikleyerek sonuca ulaşılabilir. Gürültü kapısı bu yaklaşımın en basit örneğidir. Enerji tabanlı bölümlemede; sınırların akustik değişikliklerle doğrudan bağlantısının olmaması olumsuz bir özelliktir. Model tabanlı bölümlemede; her akustik sınıf için bir takım istatistiksel modeller tanımlanmıştır. Model olarak genellikle çok değişkenli Gauss modeli kullanılmıştır. Kullanıldığı yerler ise; konuşma, müzik, arka plan gürültüsü, sessizlik, telefon konuşması vb. Model parametreleri eğitim verilerinden tahmin edilir. Çok değişkenli Gauss modelinde parametreler ortalama ve kovaryans matrisidir. Bu parametrelerin tahmini için farklı yöntemler vardır: *Maximum Likelihood Estimation* ve *Expectation Maximization* yöntemleri gibi. Optimum parametreleri hesaplamak için doğrudan geliştirilen bazı kapalı form ifadelerinin kullanılabileceği tüm matematiksel ayrıntıların incelenmesi gerekmez. Model tabanlı bölümlemede; akustik özellikler segmentasyon sınırları ile bağlantılıdır. Metrik tabanlı bölümleme ise segment sınırları devam eden iki hareketli bitişik pencere arasındaki benzerlik/mesafe içeriği tarafından belirlenir. İki komşu pencere çok değişkenli Gauss dağılımları ile modellenmiştir. İki pencerenin ses akışı üzerinde hareket etmesine izin verir. Segment sınırları, yerel maksimum ve önceden tanımlanmış bir eşik tarafından belirlenir. Metrik tabanlı algoritmayı tasarlarken dikkat edilmesi gereken seyler vardır. Bunlar; uzaklık fonksiyonu seçilmelidir, pencere boyutu, pencere hareketi hızı (zaman artışı) ve eşik değeri belirlenmelidir [29]

4 Metodoloji

Bu bölümde projede kullanılan verilerin nasıl elde edildiği ve projeye nasıl entegre edildiğinden bahsedilerek, veriler üzerinde gerçekleştirilen ayrıştırma ve öğrenme işlemi ile ilgili bilgi verilmiştir.

4.1 Veri Kümesi ve Araçlar Hakkında

Ses segmentasyonu deneyi için, Türk musikisi enstrümanlarından ney ve kanun seslerinden oluşan, ve ikinci bir deney olarak da kadın ve erkek seslerinden oluşan iki çalışma yaptık. Her iki çalışmada kullanılan veriler 20 dakikadan daha uzun ses kayıtları olup ve her birinin örnekleme frekansı 44,1 kHz'dir. Modeli oluştururken bu verileri yalnız ney, yalnız kanun veya yalnız erkek, yalnız kadın ses vektörleri şeklinde girdi olarak kullanıyoruz.[1]

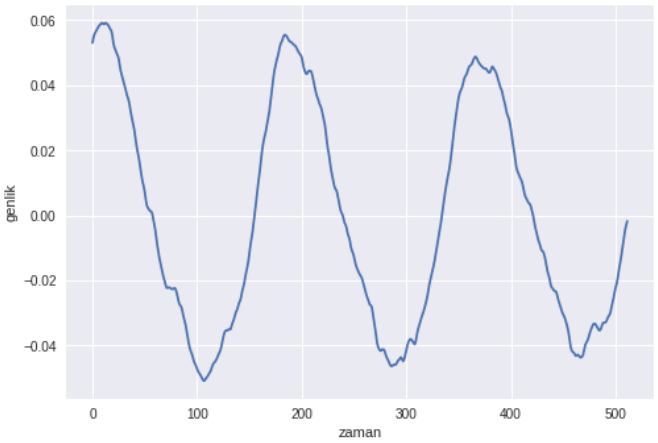
Projeyi, Pyhton programlama dilini kullanarak geliştirdik. Python, çok geniş kullanım alanı olan bir programlama dilidir ve en önemli unsurlarından biri çok hızlı işlem yapabilmesidir. Farklı dallarda hazırlanmış kütüphaneleri ile sürekli yenilenen ve test edilen bir programlama dilidir. Projede kullanılan kütüphane ise Keras'tır. Bu kütüphanenin tercih edilmesinin sebebi, diğerlerine göre daha gelişmiş bir kütüphane olmasının yanısıra modüler çalışmalar için uygun olmasıdır.

Tez için proje geliştirme aşamasında, önceden yapılan örnekleri incelerken Python ve MATLAB kullanıldığı tespit edildi. Pyhton kütüphanelerinin daha işlevsel olduğuna karar verilerek, yazılım burada gerçekleştirildi, ek olarak sonuçların gürültüden arındırılması aşamasında MATLAB kullanıldı. Kütüphane olarak ise; keras, numpy ve scipy kullanıldı. Keras, derin ağlar oluşturmak için geliştirilmiş bir kütüphanedir ve derin öğrenme alanında da kullanılan Tensorflow kütüphanesi üzerinden çalışır. Bu çalışmada keras kullanılmasının sebebi gelişmiş bir kütüphane

[1]Kullandığımız kaynak ses dosyalarına ve bunların iç içe geçirilerek önerdiğimiz metoda girdi olarak verilmiş hallerine; ayriyeten metodun çıktısı olan ses dosyalarına `http://bit.do/thesisdocuments` web adresinden ulaşılabilir.

olması ve hızlı çalışmasıdır. Numpy, proje içerisinde kullanılan aktivasyon fonksiyonlarını içeren bir kütüphanedir.

Modeli eğitmek için ilk başta okutulan ney ve kanun ses dosyalarının ve kadın/erkek ses dosyalarının belirli bır kısmını kesiyoruz. Kestiğimiz parçalar zaman penceresinde olduğundan daha iyi bir sonuç için öncelikle 512'lik parçalar şeklinde DCT formunu alıp frekans penceresine geçiyoruz. DCT formunda gelen bilgi, vektörün ilk dörtte birlik kısmında yığıldığından her vektörün ilk çeyreğini kesiyoruz. Şekil 4.1'de bir sinyal parçasının ilk hali ve Şekil 4.2'de DCT formundaki hali ve Şekil 4.3'de DCT formunun ilk çeyreği verilmiştir.

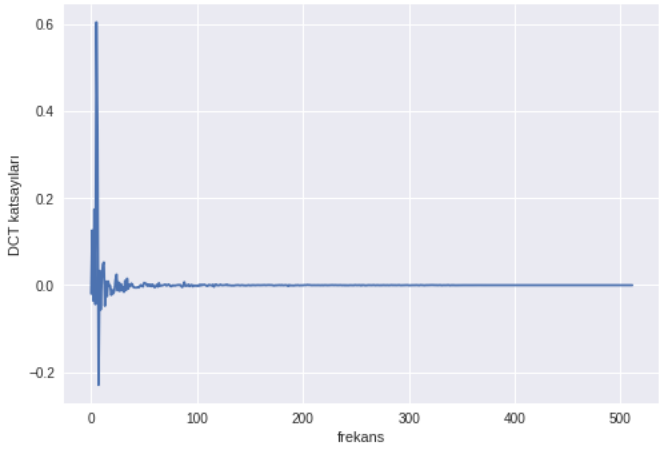


Şekil 4.1: Orijinal ney sesi örneği

4.2 Derin Otokodlayıcı Mimarimiz

Kurduğumuz AE modeli, ney ve kanun sesini öğrettiğimiz sistemde, ney ses verisini girdi olarak verip, yine ney sesini çıktı olarak veren; ney ve kanun sesini girdi olarak verip, yalnızca ney sesini çıktı olarak veren ve kanun sesini girdi olarak verdiğimizde hiçbir ses çıkışı ile karşılaşmayacağımız bir algoritma ile kaynak ayrıştırma problemleri için çözüm önerisi sunar.

Python üzerinde kurduğumuz model ardışık nöron katmanlarından oluşan simetrik bir yapıya sahip. Kodlayıcı (encoder) kısmında girdinin modele geldiği 128 nörondan oluşan ilk katman, ardından sırayla 256, 512, 1024 nörondan oluşan 3 katman eklenmiştir. Kod çözücü (decoder) kısmında kodlayıcı yapısının en alt katmanı



Şekil 4.2: DCT dönüşümü alınmış ses sinyal örneği. Ses kaynağı ney enstrümanıdır.

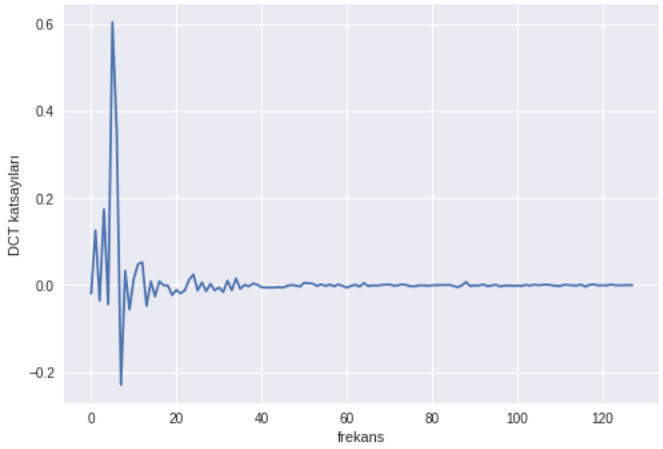
(1024 nörondan oluşan katman) simetri ekseni alınarak diğer katmanlar kopyalanmış ve yine 128 çıkışla sonlandırılmıştır. Aktivasyon fonksiyonu olarak, son katman hariç sıfırdan küçük değerleri sıfırlayan, sıfırdan büyük değerleri olduğu gibi bırakan relu fonksiyonunu kullandık. Son katmanda en iyi sonuç dağılımı için sigmoid fonksiyonunu ekledik. Şekil 4.5'de modelin yapısı gösterilmiştir.

4.2.1 Ortalama Kareler Hatası (Mean Square Error) Fonksiyonu ve Adam Optimizasyon Metodu

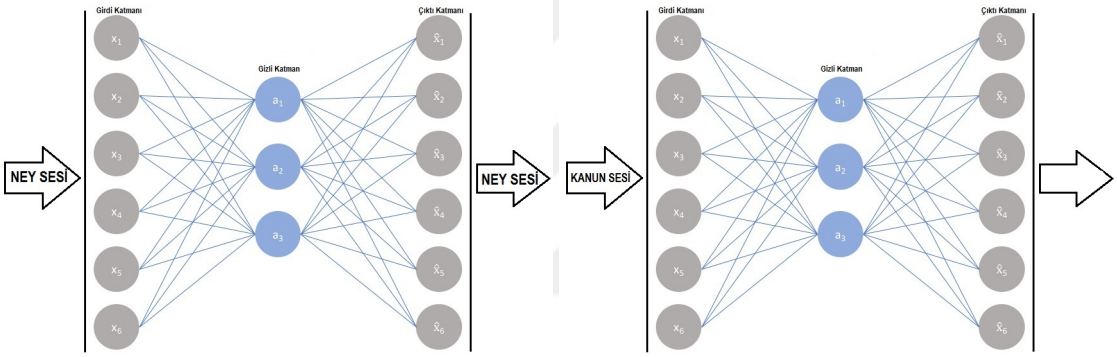
Ortalama Kare Hatası fonksiyonu en sık kullanılan regresyon kaybı fonksiyonudur. Bu fonksiyon hedef değişkenimiz ile öngörülen değerler arasındaki kare uzaklıkların toplamıdır. Ortalama Kare Hatası, aşağıdaki denklem 4.1 kullanılarak bulunmaktadır. Her turdan sonra hata hesabı yapılır ve optimizasyon fonksiyonu yardımıyla bu hata oranı düşürülmeye çalışılır.

$$MSE = \frac{\sum_{i=0}^{n} (y_i - y_i^p)^2}{n} \tag{4.1}$$

Adam, derin öğrenme yaklaşımlarında kullanılan *Stochastic Gradient Descent* (SGD) ailesine ait Adadelta'ya benzer bir algoritmadır. AdaDelta'dan farklı olarak parametrelerin her birinin öğrenme oranlarının yanısıra momentum değişikliklerini de önbellekte (cache) saklar; yani RMSprop ve momentumu birleştirir. Bu çalışmada en iyi sonuç elde etmek için ortalama kareler hesabı ve Adam fonksiyonu kullanıldı. SGD içerisinde bulunan metotlardan Adam kullanma sebebimiz, ses dosyalarını projeye entegre ederken .dct uzantısı kullanmamızdan kaynaklanıyor.



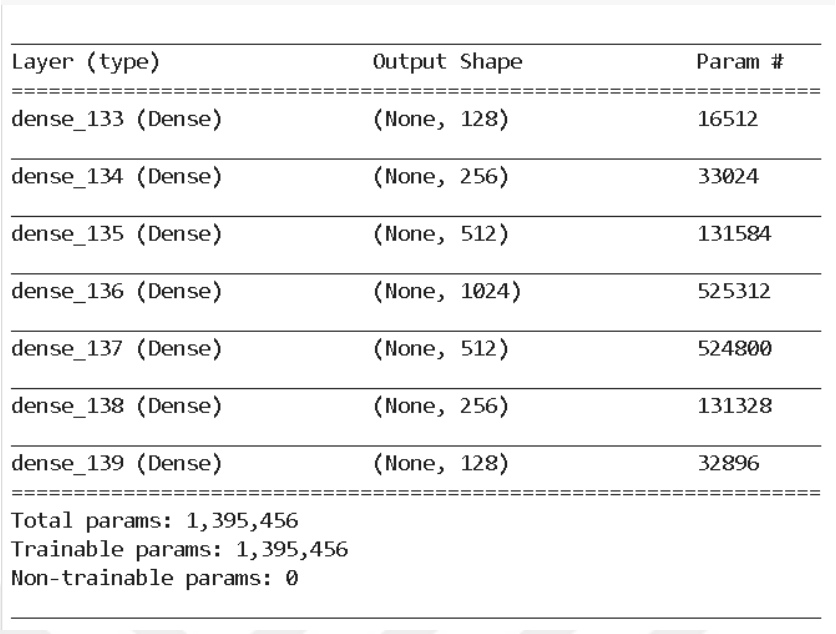
Şekil 4.3: DCT dönüşümünün ilk çeyreği. Derin otokodlayıcı öğrenme algoritmasına enerji sıkışmasından dolayı sadece ilk çeyreği verilmiştir.



(a) Ney sesini öğrenen AE yapısı (b) Kanun sesini öğrenen AE yapısı

4.2.2 Model Başarısının Ölçülmesi

Modelin eğitim başarısını test etmek için, eğitim verisinin bir kısmını modele tekrar veriyoruz ve modelin verdiği veriye göre, modelin öğrenip öğrenmediğini test ediyoruz. Şekil 4.8'de mavi renkle çizilen sinyal modeli eğitirken kullandığımız ney sesinden bir parça; yeşil renkli olan sinyalse modelin tahmin ettiği sinyal. Görüldüğü gibi minik bir gürültüyle de olsa tahmin oldukça iyi düzeydedir. Ek olarak kanun sesini sfırlamayı öğrenen modelimize kanun sesinden oluşan bir sinyal örneği dizisini atıp sesleri sıfırlamasını bekliyoruz. Örneğin Şekil 4.9'de mavi sinyal, modele verilen kanun vektörüyken; sıfıra yakın bir doğrultuda ilerleyen yeşil sinyal modelin tahmin ettiği vektördür. Buradan modelin kanun sesini sıfırlamayı da öğrendiğini teğit etmiş oluyoruz. Aynı zamanda tahmin edilen kısmın sadece



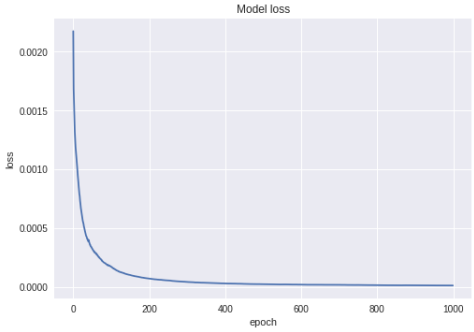
Layer (type)	Output Shape	Param #
dense_133 (Dense)	(None, 128)	16512
dense_134 (Dense)	(None, 256)	33024
dense_135 (Dense)	(None, 512)	131584
dense_136 (Dense)	(None, 1024)	525312
dense_137 (Dense)	(None, 512)	524800
dense_138 (Dense)	(None, 256)	131328
dense_139 (Dense)	(None, 128)	32896

Total params: 1,395,456
Trainable params: 1,395,456
Non-trainable params: 0

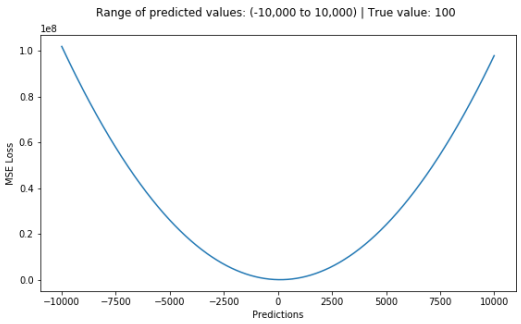
Şekil 4.5: Kullanılan seçici derin otokodlayıcı mimarisine ait detaylar

çizdirilmesi yeterli olmayacağından, eğitim verisinin bir kısmı test için ayrılmıştır. Bu iki kısım da ayrı ayrı ses dosyasına dönüştürülmüştür. Eğitim için kullanılan ses ile modelin bu sese karşılık tahmin ettiği vektörler birleştirilerek oluşturulan ses kıyaslanmıştır. Yine modelin başarılı olduğu görülmüştür.

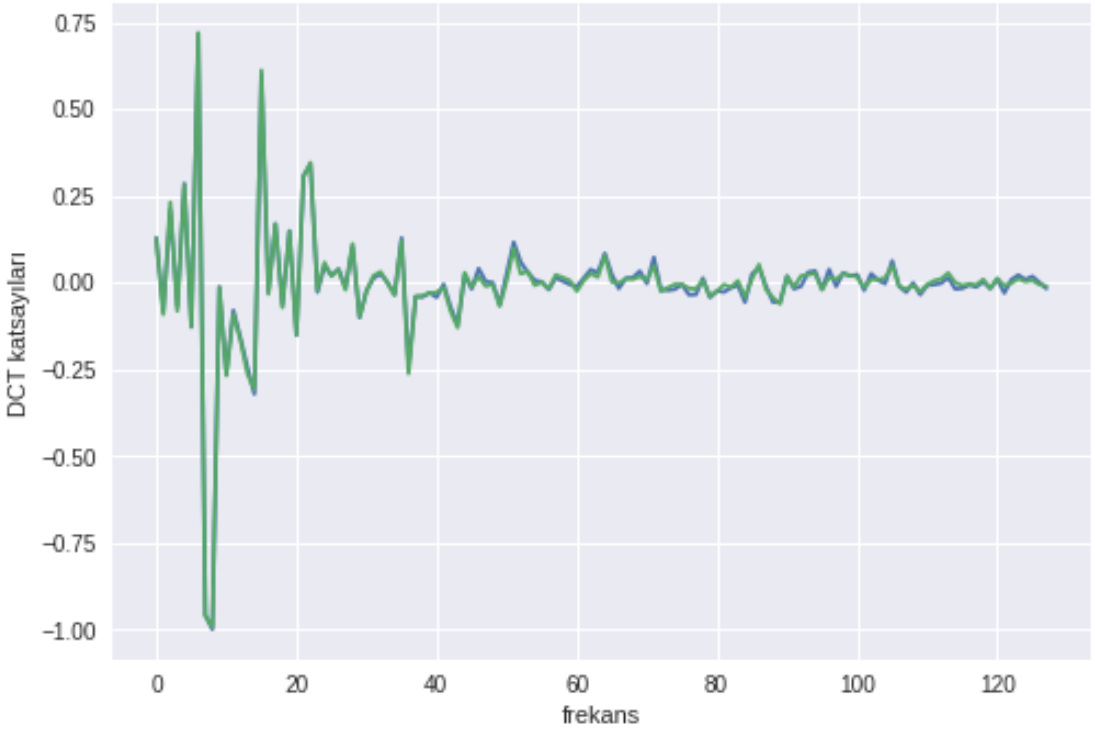
Modelin test başarısını ölçebilmek için modeli eğittiğimiz veriden farklı bir veri saklanmış ve modeli test etmek için kullanılmıştır. Şekil 4.10'de görüldüğü gibi mavi renkli ney sinyali modelin bilmediği bir kısımdan olmasına rağmen model bu sinyalin ney olduğunu tahmin etmiştir. Tahmin sinyali yeşil renkle çizdirilmiştir. Aynı adım kanun için tekrarlanmış ve model bilmediği bir kanun sinyalinin kanun olduğunu tahmin ederek bu sinyali sıfırlamıştır.



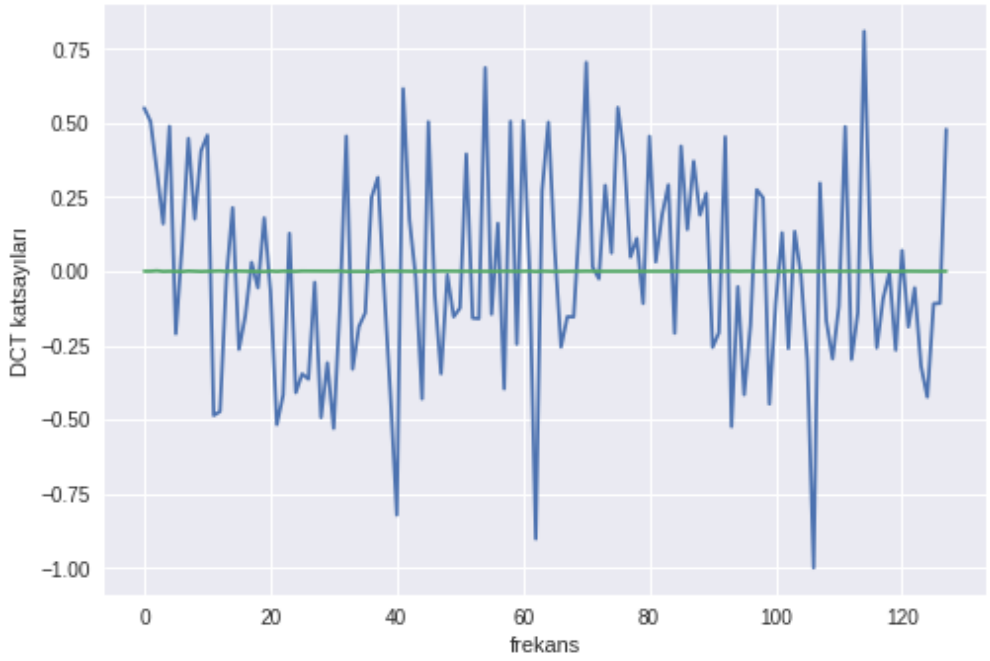
Şekil 4.6: Her bir epoch değerlerine karşılık modelin kayıp değerleri.



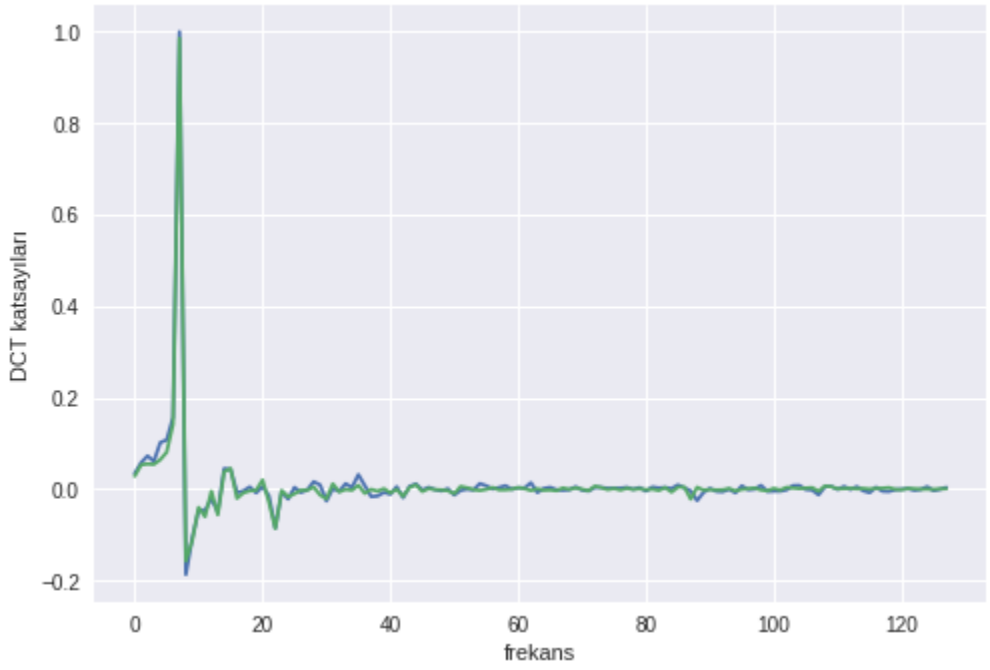
Şekil 4.7: Her bir tahmin değerine göre ortalama kare fonksiyon değerleri.



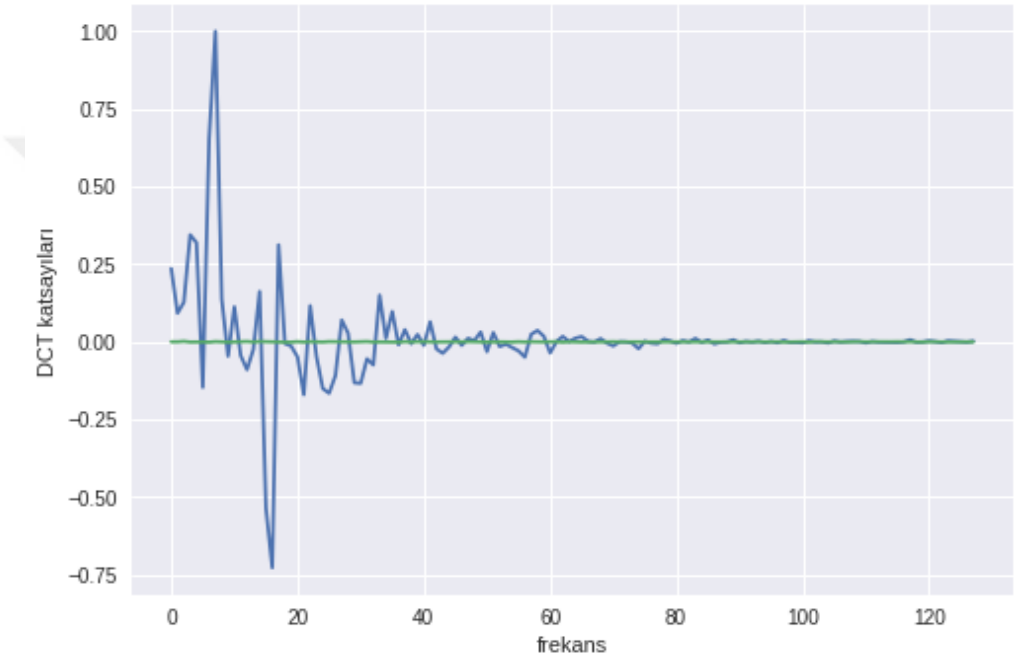
Şekil 4.8: Eğitim için kullanılan, modele girdi olarak verilen ney sinyali örneğinin DCT katsayıları (modele eğitim sırasında gösterilmiş ses parçası, mavi renkte) ve seçici derin otokodlayıcı modelin çıktısının DCT katsayıları (yeşil renkte)



Şekil 4.9: Eğitim için kullanılan, modele girdi olarak verilen kanun sinyali örneğinin DCT katsayıları (modele eğitim sırasında gösterilmiş ses parçası, mavi renkte) ve seçici derin otokodlayıcı modelin çıktısının DCT katsayıları (yeşil renkte)



Şekil 4.10: Test için kullanılan, modele girdi olarak verilen ney sinyali örneğinin DCT katsayıları (modele daha önce gösterilmemiş ses parçası, mavi renkte) ve seçici derin otokodlayıcı modelin çıktısının DCT katsayıları (yeşil renkte)



Şekil 4.11: Test için kullanılan, modele girdi olarak verilen kanun sinyali örneğinin DCT katsayıları (modele daha önce gösterilmemiş ses parçası, mavi renkte) ve otokodlayıcı modelin çıktısının DCT katsayıları (yeşil renkte)

5 Sonuç

Bu tezde seçici derin otokodlayıcılar mimarisini kullanarak, ardışık ses kaynaklarının segmente edilmesinde ve bunun doğal sonucu olarak ayrıştırılmasında kullanılabilecek bir metot geliştirdik. Metodun ses kaynaklarıyla sınırlı kalmaksızın half-duplex kanal kayıtlarının ayrıştırılmasında da kullanılabileceğine inanıyoruz. Geliştirilen teknik konuşmacı tanıma ve konuşmacı ses kümelemesinden tutun da ses bankalarının taranarak belli ses kaynaklarının ayrıştırılmasına kadar çok geniş bir uygulama alanında kullanım alanı bulabilir.

Tezde içinde sadece iki kaynak bulunan senaryolar için, farklı enstrüman sesleri ve kadın/erkek sesleri için tekniğimizi denedik ve oldukça başarılı sonuçlar elde ettik. Birden fazla kaynak bulunması durumlarında sistemin performansının denenmesi veya ses kaynaklarının ve ses benzerliklerinin artması durumundaki performansın ne olacağı konusu ileriki çalışmalarda ele alınacaktır. Örneğin iki farklı kişiye ait ardışık kadın seslerinin segmente edilebilmesi durumu gibi. Çalışma, seçici derin otokodlayıcıların, ardışık sinyal segmentasyonundaki potansiyeli hakkında oldukça ümit verici sonuçlar ortaya koymaktadır. Öte yandan, ardışık değil de üst üste binen seslerin ayrıştırılmasında önerdiğimiz modelin etkisiz olduğu görülmüştür.

6 Ekler

```
!pip install −U −q PyDrive
import os
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials


auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

from google.colab import drive
drive.mount("/content/drive")
######kullanilacak dosyalarin okutulmasi
import wave
from scipy.io import wavfile
import matplotlib.pyplot as plt
from matplotlib.pyplot import *
import soundfile as sf
######yeterli uzunlukta kesilmesi
ney_stereo,fs_ney=sf.read('/content/drive/My Drive/Ney.wav')
kanun_stereo, fs_kanun = sf.read('/content/drive/My Drive/Kanun.wav')
ney_mono=ney_stereo[0:128401388,0]#25600000
kanun_mono=kanun_stereo[0:128401388,0]
######dct ve normalizasyon uygulanmasi
from scipy.fftpack import  idct, dct
import numpy as np
dct_window_size=512
ney_dct=[]
kanun_dct=[]
ney=[]#ise yarayan 512*1lik dct parcalari kaydetmek icin
kanun=[]
safney=[]
```

```
safkanun=[]#saf ney ve kanun seslerinin kaydettik, kiyaslama icin.
maxval_ney=[]
maxval_kanun=[]#tahmin edilen degerleri denormalize etmek icin kaydededicez
for j in range(0,150000):
    r=dct(ney_mono[j*dct_window_size:(j+1)*dct_window_size],norm='ortho')
    r1=dct(kanun_mono[j*dct_window_size:(j+1)*dct_window_size],norm='ortho')
    if np.mean(r)!=0 and np.mean(r1)!=0:#birisi bile 0 olmayacak.
        ney_dct.append((r[0:128].astype('float32')/np.max(np.abs(r[0:128] +
    ↪ 0.00000000000000000001))).T)
        kanun_dct.append((r1[0:128].astype('float32')/np.max(np.abs(r1[0:128] +
    ↪ 0.00000000000000000001))).T)
        safney.append((ney_mono[j*dct_window_size:(j+1)*dct_window_size]))#ilk halleri
        safkanun.append((kanun_mono[j*dct_window_size:(j+1)*dct_window_size]))
        ney.append(r)#dctleri ekledik
        kanun.append(r1)
        maxval_ney.append(np.max(np.abs(r[0:128])))
        maxval_kanun.append(np.max(np.abs(r1[0:128])))
ney_dct=np.asarray(ney_dct)
kanun_dct=np.asarray(kanun_dct)
ney=np.asarray(ney)#ilk 128lik kismini degistirdiklerimiz
kanun=np.asarray(kanun)
maxval_ney=np.asarray(maxval_ney)#maksimum değerler
maxval_kanun=np.asarray(maxval_kanun)
safney=np.asarray(safney)#maksimum değerler
safkanun=np.asarray(safkanun)



#######egitim datasinin olusturulmasi
inp_train=[]
out_train=[]
zero=np.zeros([1,128])
for i in range(0,20000):
    if i<100000:
        inp_train.append(ney_dct[i])
        out_train.append(ney_dct[i])
    else:
        inp_train.append(kanun_dct[i-100000])
        out_train.append(zero[0])

inp_train=np.asarray(inp_train)
out_train=np.asarray(out_train)
inp_train=(inp_train+1)/2
out_train=(out_train+1)/2
######modelin kurulmasi  ve egitilmesi
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras import backend as K
```

```
model = Sequential()
model.add(Dense(128,activation='relu',input_dim=128))
model.add(Dense(256,activation='relu'))
model.add(Dense(512,activation='relu'))
model.add(Dense(1024,activation='relu'))
model.add(Dense(512,activation='relu'))
model.add(Dense(256,activation='relu'))
model.add(Dense(128,activation='sigmoid'))
model.compile(loss=keras.losses.mean_squared_error,
        optimizer=keras.optimizers.Adam(lr=0.001, beta_1=0.999, beta_2=0.999, epsilon=
    ↪ None, decay=0.0, amsgrad=False))

###verinin ilk ve son kisminndan 5er biner tane vektoru test icin ayiirdik
model.fit(inp_train[5000:195000],out_train[5000:195000],verbose=1,epochs=1000,batch_size
    ↪ =100)
######test etme
predictions = model.predict(inp_train)
```

Kod 6.1: Tez çalışmasına temel olan programlama çalışmasında geliştirilen kodlar. Python dilinde Keras kütüphaneleri kullanılarak geliiştirilmiştir.

Kaynaklar

[1] Saadia Zahid, Fawad Hussain, Muhammad Rashid, Muhammad Haroon Yousaf, and Hafiz Adnan Habib. Optimized audio classification and segmentation algorithm by using ensemble methods. *Mathematical Problems in Engineering*, 2015, 2015.

[2] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[3] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 31–35. IEEE, 2016.

[4] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-channel multi-speaker separation using deep clustering. *arXiv preprint arXiv:1607.02173*, 2016.

[5] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 241–245. IEEE, 2017.

[6] Arun Narayanan and DeLiang Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7092–7096. IEEE, 2013.

[7] Yuxuan Wang and DeLiang Wang. Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1381–1390, 2013.

[8] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, pages 477–482, 2014.

[9] Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. Recurrent neural networks for noise reduction in robust asr. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[10] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1562–1566. IEEE, 2014.

[11] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[12] Derry Fitzgerald, Matt Cranitch, and Eugene Coyle. Shifted non-negative matrix factorisation for sound source separation. 2005.

[13] M Altug Keyder. *Blind Audio Source Separation Using Nonnegative Tensor Factorization Techniques*. PhD thesis, 2008.

[14] Nikolaos Mitianoudis. *Audio source separation using independent component analysis*. PhD thesis, Citeseer, 2004.

[15] Tuomas Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *ICMC*, pages 231–234, 2003.

[16] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440, 2013.

[17] Hirokazu Kameoka, Li Li, Shota Inoue, and Shoji Makino. Semi-blind source separation with multichannel variational autoencoder. *arXiv preprint arXiv:1808.00892*, 2018.

[18] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.

[19] Lie Lu, Hong-Jiang Zhang, and Hao Jiang. Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing*, 10(7):504–516, 2002.

[20] Ladislava Smítková Janku and Katerina Hyniová. Application of feed-forward neural network and mmi-supervised vector quantizer to the task of content based audio segmentation by co-operative unmanned flying robots. In *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, pages 111–115. IEEE, 2010.

[21] Jainath Yadav and K Sreenivasa Rao. Detection of vowel offset point from speech signal. *IEEE Signal Processing Letters*, 20(4):299–302, 2013.

[22] Vasif V Nabiyev. Yapay zeka. *ISBN*, 975(347):985, 2005.

[23] Mustafa Furkan Keskenler and Eyüp Fahri Keskenler. Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2):8–18.

[24] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[25] Antonio Regalado. Is google cornering the market on deep learning. *Technology Review*, 29, 2014.

[26] David E Rumelhart and James L McClelland. Parallel distributed processing: explorations in the microstructure of cognition. volume 1. foundations. 1986.

[27] Otokodlayıcılar. `http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/`. Erişim tarihi: 02/12/2018.

[28] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 90–98. SIAM, 2017.

[29] Ses bölütleme. `http://www.music.mcgill.ca/~ich/classes/mumt611_07/presentations/shiyong/shiyong07audio.pdf`. Erişim tarihi: 07/12/2018.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: ÖZKARDAŞ, Meryem Betül
Uyruğu	: TC
Doğum Yeri ve Tarihi	: 10/09/1992
Medeni Hali	: Evli
Tel	: +90 0544 723 42 02
Fax	: -
e-mail	: meryembetulduran@outlook.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	: Fatih Üniversitesi	Temmuz-2014
Yüksek Lisans	: KTO Karatay Üniversitesi	Aralık-2018
Phd	: -	

Yabancı Dil

İngilizce(İyi)

Yayınlar

-