DERİN ÖĞRENME TABANLI KONUŞMA TANIMA SİSTEM TASARIMI

Burak KORCUKLU

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TABANLI KONUŞMA TANIMA SİSTEM TASARIMI

Burak KORCUKLU
0000-0002-9820-4444

Doç. Dr. Ahmet Emir DİRİK
0000-0002-6200-1717
(Danışman)

YÜKSEK LİSANS
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2021

TEZ ONAYI

Burak KORCUKLU tarafından hazırlanan "Derin Öğrenme Tabanlı Konuşma Tanıma Sistem Tasarımı" adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Ahmet Emir DİRİK

Başkan : Doç. Dr. Ahmet Emir DİRİK
0000-0002-6200-1717
Bursa Uludağ Üniversitesi,
Mühendislik Fakültesi,
Bilgisayar Mühendisliği Anabilim Dalı
İmza

Üye : Doç. Dr. Cemal HANİLÇİ
0000-0002-9174-0367
Bursa Teknik Üniversitesi,
Mühendislik ve Doğa Bilimleri Fakültesi,
Elektrik-Elektronik Mühendisliği Anabilim Dalı
İmza

Üye : Dr. Öğr. Üyesi Alkın YURTKURAN
0000-0003-2978-2811
Bursa Uludağ Üniversitesi,
Mühendislik Fakültesi,
Endüstri Mühendisliği Anabilim Dalı
İmza

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Aksel EREN
Enstitü Müdürü
.../....

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15/02/2020

Burak KORCUKLU

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME TABANLI
KONUŞMA TANIMA SİSTEM TASARIMI

Burak KORCUKLU

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ahmet Emir DİRİK

Doğal dil işleme, bilgisayarların, doğal yazı veya konuşma dilini nasıl anlayabileceğini ve manipüle edebileceğini çözümleyen araştırmalardan ve uygulamalardan oluşmaktadır. Konuşma işleme ise konuşma sinyalleri ve sinyal işleme yöntemlerini barındıran doğal dil işlemenin bir alt alanıdır. Konuşma sinyalleri çoğunlukla dijital gösterimleri üzerinden işlenerek farklı yöntemler ile yazı diline çevrilmektedir. Genellikle eğitim ve test aşamalarından oluşan bu süreç, eldeki etiketli veriler kullanılarak modeli eğitmek ve farklı etiketli veriler ile eğitilen modelin tutarlılığının ölçülmesini kapsamaktadır.

Tarih boyunca birçok araştırmacı konuşulan dili yazıya dönüştürmek için farklı yaklaşımlar ve yöntemler geliştirmişlerdir. Günümüzde özel firmaların geliştirdikleri çevrimiçi konuşma tanıma modelleri birçok çalışma alanında kullanılmaktadır. Geliştirilen bu modeller Saklı Markov Modeli (HMM), yapay sinir ağları, gürültü temizlemek için kullanılan algoritmalar, derin öğrenme algoritmaları ve fonem sözlüklerinin bir arada kullanılmasıyla gerçekleştirilmektedir. Bu modellerin kullanımı akıllı ev sistemleri, otomotiv, askeriye, sağlık gibi çeşitli alanlarda gün geçtikçe artmaktadır. Kullanılan modellerin çoğunlukla çevrimiçi çalışması, kullanıcı tarafından yeni geliştirmelere izin vermemesi ve yetersiz dil desteği sebebiyle hala geliştirilmesi gereken birçok yanı bulunmaktadır.

Bu tezde iki farklı doğal dilden yazıya dönüşüm modeli oluşturulmuştur. İlk model geleneksel yöntemlere alternatif olarak geliştiricinin işlem yükü ve karmaşıklığı daha az olan uçtan uca derin öğrenme yöntemi ile; ikincisi ise geleneksel yöntemlerle ön işlemeli bir süreç izlenerek gerçekleştirilmiştir. Bu modellerin konuşmacı bağımlılığı, veri seti boyutu, eğitim süresi gibi farklı koşullardaki başarıları saptanmaya çalışılmıştır. Ayrıca her iki modelin eğitim ve test aşamaları için gerekli veri setini oluşturmak amacıyla kullanıcılardan etiketli veri toplanabilecek ağ tabanlı bir yazılım geliştirilmiştir.

Anahtar Kelimeler: Konuşma tanıma, derin öğrenme, konuşmadan yazıya dönüşüm, sinyal işleme, doğal dil işleme

2021, vii + 60 sayfa.

ABSTRACT

MSc Thesis

DEEP LEARNING BASED
SPEECH RECOGNITION SYSTEM DESIGN

Burak KORCUKLU

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Ahmet Emir DİRİK

Natural language processing consists of research and applications on how computers can understand and manipulate natural writing or spoken language. Speech processing is a sub-field of natural language processing that includes speech signals and signal processing methods. Speech signals are mostly processed through digital representations and translated into written language with different methods. This process, which usually consists of training and testing phases, includes training the model using the labeled data at hand and measuring the consistency of the trained model with different labeled data.

Throughout history, many researchers have developed different approaches and methods to translate spoken language into writing. Today, online speech recognition models developed by private companies are used in many areas of work. These developed models are realized by using Hidden Markov Model (HMM), artificial neural networks, algorithms used for noise removal, deep learning algorithms and phoneme dictionaries together. The use of these models is increasing day by day in various fields such as smart home systems, automotive, military and health. The models used are mostly online, do not allow new developments by the user, and there are still many aspects that need to be improved due to insufficient language support.

In this thesis, two different natural language-to-text transformation models have been created. As an alternative to traditional methods, the first model uses end-to-end deep learning method with less processing load and complexity; The second one was carried out following a pre-processed course with traditional methods. The success of these models in different conditions such as speaker addiction, data set size, and duration of education was tried to be determined. In addition, a network-based software has been developed to collect labeled data from users in order to create the necessary data set for the training and testing stages of both models.

Key words: Speech recognition, deep learning, speech-to-text conversion, signal processing, natural language processing

2021, vii + 60 pages.

TEŞEKKÜR

Bu tezi yapmamda ve yazmamda bana her dağim desteğini esirgemeyen aileme, veri toplama sürecine katkıda bulunan arkadaşlarıma ve doğru yönlendirmeleri ve yapmış olduğu desteklerden ötürü danışmanım Doç. Dr. Ahmet Emir DİRİK'e teşekkürlerimi sunarım.

Burak KORCUKLU
15/02/2020

İÇİNDEKİLER

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
γ	Adım boyutu
W	Ağırlık
z	Ağırlıklı girdi
a	Aktivasyon fonksiyonu
K	Çerçeve sayısı
p	Frekans dilimi
h	Gizli katman vektörü
L	Katman
C	Maliyet
b	Sapma
$\hat{y}$	Tahmini çıktı
X	Veri seti
δ	Yardımcı miktar (Auxiliary quantity)
t	Zaman adımı
T	Zaman periyodu

Kısaltmalar	Açıklama
BLAS	Basic Linear Algebra Subprograms
CMVN	Cepstral Mean and Variance Normalization
CTC	Connectionist Temporal Classification
DTW	Dynamic Time Warping
FST	Finite-state Transducer
HMM	Saklı Markov modeli
IBM	International Business Machines
IDA	Institude for Defence Analysis
LAPACK	Linear Algebra Package
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coding
LSTM	Long Short-term Memory
MFCC	Mel-frequency Cepstrum
MLLT	Maximum Likelihood Linear Transform
NAG	Nesterov's Accelerated Gradient
PLP	Perceptual Linear Predictive
STC	Semi-tied Covariance
TSA	Tekrarlayan Sinir Ağı
VTLN	Vocal Tract Length Normalization
WFST	Weighted Finite-state Transducer
YSA	Yapay Sinir Ağı

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

1. GİRİŞ

Gelişen teknolojinin her geçen gün günlük yaşantımıza daha fazla dahil olmasıyla birlikte insan–bilgisayar etkileşimi alanında yapılan çalışmaların popülaritesi de artmıştır. Eskiden lüks olarak sayılan telefon, televizyon gibi aygıtlar dahi günümüzde orijinallerinden çok daha gelişmiş evrelere ulaşmış ve artık birçok teknolojik cihaz kullanımı tercihten çok zorunluluk haline dönüşmüştür. Akıllı ev sistemleri, sanal asistanlar, taşıt teknolojileri vb. insan–bilgisayar etkileşiminin ön planda olduğu teknolojik ürünler gün geçtikçe dünyada daha yaygın hale gelmektedir. Ayrıca bu teknolojik gelişmeler, birçok yeni iş sektörü doğurduğu gibi mevcut çalışma alanlarını da teknolojik olarak gelişmeye iterek bilinen endüstri düzeninin değişimine yol açmıştır. Makinelerin insanlarla bu kadar iç içe olduğu bir düzende, teknolojik cihaz ve yazılımlar ile insanlar arasındaki iletişimin kolaylığı büyük önem taşımaktadır.

Bilgisayarlı konuşma tanıma, otomatik konuşma tanıma, konuşmadan yazıya çeviri gibi farklı şekillerde de tanımlanan konuşma tanıma kavramı, insan sesinin makineler tarafından işlenerek söylenenlerin yazıya ya da anlaşılabilecek bir formata dönüştürülmesi anlamına gelmektedir. Makine ile kurulan tek taraflı ve başarılı iletişimin ardından bunun karşılığında makineden beklenen işlemler kolaylıkla programlanabilecek bir hal almaktadır. Yani konuşma verisi bir kez yazıya dönüştürülebildiğinde, makinenin bu komutları algılayarak istenilen görevleri gerçekleştirebilmesi yalnızca hayal gücü ile sınırlıdır. Doğal dil işlemenin alt adımı olan konuşma tanıma, yazının konuşmaya çevrildiği konuşma sentezi ile birlikte makinelerle çift taraflı sesli etkileşimi mümkün hale getirmektedir. Buna örnek olarak Amazon Alexa, Microsoft Cortana, Google Assistant ve Apple Siri gibi kullanımı oldukça yaygın olan sesli asistanlar verilebilir. Bu tarz sistemler kullanıcının söylediği komutları analiz ederek, gerekli işlemleri yaptıktan sonra kullanıcıya sesli geri dönüş sağlamaktadırlar. Bu komutlar örneğin internette sesli arama yapmak, hava durumu ve döviz kurunu öğrenmek gibi temel komutlar olabileceği gibi, internetten ürün sipariş vermek, bir şirketin geçmiş yıllardaki satış rakamlarını öğrenmek gibi karmaşık istekler dahi olabilmektedir. Asistanların akıllı ev sistemleri ve diğer cihazlar ile entegrasyonu sayesinde akıllı evlerdeki termostat, akıllı süpürgeler, aydınlatma vb. tüm teknolojik cihazlarla etkileşim bir üst seviyeye taşınmıştır. Bu sayede görme ve fiziksel engelli insanların makinelerle etkileşimi de sesli komut sistemleri

sayesinde daha konforlu hale gelmiştir. Konuşma sistemlerinin farklı alan ve pazarlara adapte olması ile birlikte cihazların manuel kullanımının önüne geçilmiş, filmlere altyazı oluşturulması gibi insan emeği gerektiren birçok görevin yerini bu işlemleri eş zamanlı gerçekleştirebilen konuşma tanıma yazılımları almıştır. Otomotiv sektöründe bu alandaki gelişmeler sayesinde insanların araç kullanırken telefona bakmadan mesaj yazmalarının ve okumalarının sohbet etmek kadar basitleşmesi ile telefon kaynaklı trafik kazalarının önüne geçilmeye çalışılmıştır. Konuşma tanıma sistemlerinin çağrı merkezlerinde kullanılmaya başlanmasının ardından hem çağrı denetimleri kolaylaşmış hem de müşterilerin problemlerinin daha etkili bir biçimde çözülebilmesi amaçlanmıştır. Yani konuşma tanıma sistemleri eğitim, milli savunma, robot bilimi gibi daha birçok alanda makinelerle etkileşimi kolaylaştırarak verimi bir üst seviyeye taşımaktadır. Bu doğrultuda bakıldığında mahkeme kayıtlarının dahi otomatik olarak tutulduğu teknolojik cihazlarla iletişimin insanlar arasındaki iletişim kadar basitleştiği bilim kurgu filmlerinden aşina olunan bir gelecek oldukça yakın görünmektedir.

Bilgisayar ile makinenin sözlü iletişimi alanındaki çalışmalar 1950'li yıllara kadar dayanmaktadır. Geçmiş yıllarda donanımsal eksiklikler, erişim zorlukları ve yazılım alanındaki gelişmelerdeki yetersizlikler bu alandaki çalışmaların tatmin edici sonuçlar vermesinin önüne geçmiş olsa da bugün konuşma tanıma alanındaki çalışmalar, aktif kullanılabilirlik seviyesine ulaşmış durumdadır. Konuşma tanıma sistemleri yeni öğrenen bir bebeğe benzetilebilmektedir. İnsanların konuşarak iletişim kurmalarını içgüdüsel olarak nitelendirsek de insanlar da doğdukları ilk günden itibaren çevrelerindeki konuşmaları dinleyerek ve bu konuşmaları anlamlandırarak öğrenmektedirler. Bu işlevi gerçekleştirecek bir yazılım tasarlanmak istendiğinde de izlenen yöntemler insan öğrenimiyle büyük ölçüde benzerlikler göstermektedir.

Konuşma tanıma sistem tasarımının en büyük zorluklarından biri insan faktörüdür. Dünyada konuşulan diller, konuşmacıların şiveleri ve ağızları, cinsiyetleri, vurgulamaları ve konuşma tonları gibi birçok varyasyonu kapsayan bir sistem tasarlamak, ihtiyaç duyulan veri miktarını önemli ölçüde arttırmaktadır. Bunun yanı sıra sistemi hem geliştirirken hem de kullanırken kayıtlardaki gürültüler, kayıt cihazı kalitesi gibi çevresel etmenlerin de göz önünde bulundurulması gerekmektedir. Hal böyle olunca kullanışlı bir sistem tasarlamak çok fazla iş gücü, etiketli verilerden oluşan büyük bir veri seti ve bu

verileri analiz edebilecek işlemci gücünü zorunlu kılmaktadır. Tarihsel süreçte yalnızca belirli bir amaca hizmet eden sistemlerin üzerine yoğunlaşılmasının temel sebebi de budur fakat yine de konuşmacıdan bağımsız, etkili kullanılabilen bir sistem geliştirilebilmesi uzun yıllar almıştır. Her ne kadar kullanım kolaylığı nedeniyle basit bir yapı olarak düşünülse de tarih boyunca gelişimi büyük veri, yapay zeka, makine öğrenmesi ve derin öğrenme kavramlarının ortaya çıkışına kadar günümüzde kullanılan sistemler seviyesine erişememiştir. Bu kavramların ortaya çıkışı, veri ön işleme aşamalarını büyük ölçüde küçültmüş ve sistematize etmiştir. Tez kapsamında hazırlanılan veri seti ile eğitilen ve test edilen, uçtan uca derin öğrenme yaklaşımı ve geleneksel yaklaşım kullanılarak oluşturulan ve çevrimdışı çalışabilen iki farklı konuşma modelinin başarıları ölçülerek analiz edilmiştir.

Çevrimdışı çalışabilen ve Türkçe dilini algılayabilen sistem geliştirme çalışmalarının önemi, mevcut durumda kullanılan yöntemlerin masraflı olması, internet erişimi gereksinimi ve Türkçe dil desteklerinin yetersiz kalmasıdır. Meeker'in (2017) Kaliforniya'da gerçekleştirilen Code Konferansı'nda sunduğu İnternet Trendleri Raporu'na göre bu alandaki çalışmaların öne çıkan isimlerinden Google, İngilizce dilindeki konuşmaları tanımada insan tanıma düzeyi olan %95 başarı oranına ulaşmıştır. Bu büyük bir başarı olmasına rağmen farklı dillerdeki oran İngilizceye kıyasla düşük kalmaktadır. Ayrıca Google, Apple gibi firmalar konuşma tanıma sistemlerini ticari kullanıma açarken yüksek meblağlarda ücret talep etmekte ve kullanılan çoğu sistem çevrimiçi olacağı için hem internet erişimine ihtiyaç duymakta, hem de istenildiğinde dışarıdan kısıtlamalara açık hale gelmektedir. Bu sebeplerden dolayı tezde açık kaynak kodlu ve geleneksel bir konuşma tanıma yazılımı olan Kaldi ile alt adımların birleşiminden oluşan bir yapı kurularak ve Derin Konuşma ile uçtan uca derin öğrenme kullanarak iki farklı yöntem ile çevrimdışı çalışabilen ve Türkçe komutları algılayabilen konuşma tanıma sistemleri oluşturulmuştur. Ardından başarı oranlarını etkileyen faktörler tespit edilmeye çalışılmıştır. Yöntemlerin başarı oranları konuşmacıya bağımlılık, veri seti boyutu, konuşmacı çeşitliliği gibi farklı durumlar göz önünde bulundurularak ölçülmüş ve kıyaslanmıştır.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

2.1. Doğal Dil İşleme

Dilbilimin amacı, sözlü ve yazılı kaynaklardan edinilen çok sayıda dilbilimsel gözlemi karakterize edip açıklayabilmektir. Bu alandaki araştırmaların bir kısmı insanların dili nasıl edindikleri, ürettikleri ve anladıklarının bilişsel yanıyla ilgilenirken diğer bir kısmı da dilbilimsel ifadeler ve dünya arasındaki ilişki ve iletişim kurulan dilin yapısı ile ilgilenmektedir. Günümüzde teknolojik gelişmeler sayesinde, dilbilimin öğretileri ışığında bir dili, dile özgü kavramları ve kuralları analiz etmek ve o dildeki sözlü ve yazılı iletişimi makineler vasıtasıyla sağlamak kolaylaşmış ve doğal dil işleme kavramı ortaya çıkmıştır.

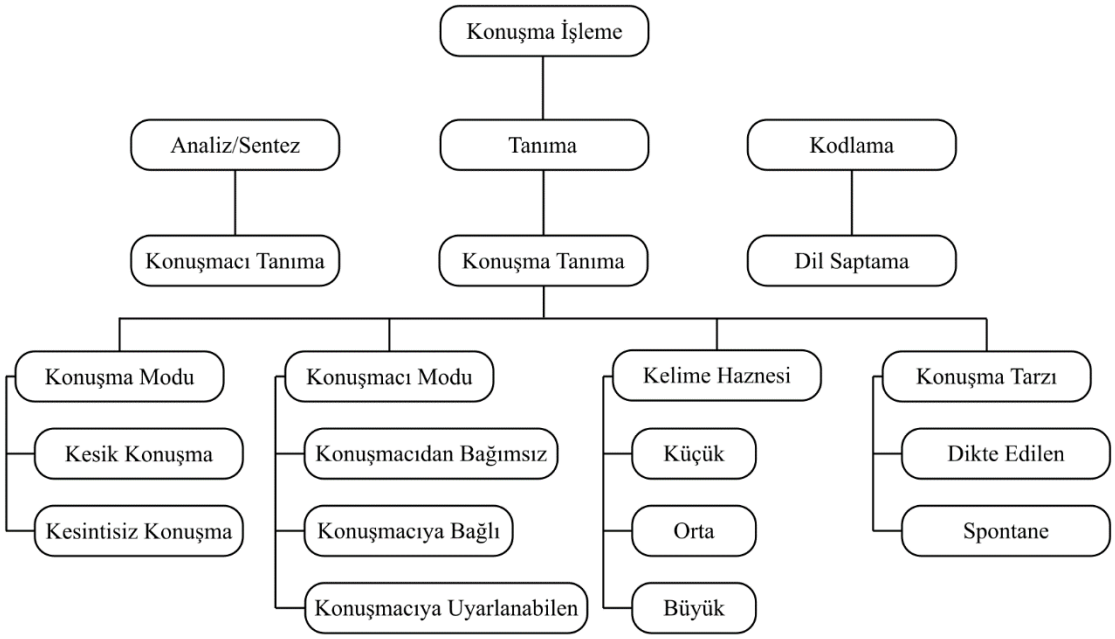
Doğal dil işleme, bilgisayarların, doğal yazı veya konuşma dilini nasıl anlayabileceğini ve manipüle edebileceğini çözümleyen araştırmalar ve uygulamalardan oluşmaktadır. Doğal dil işlemenin temellerini bilgisayar bilimleri, dil bilimi, matematik, elektrik-elektronik mühendisliği, yapay zeka, robot bilimi ve psikoloji gibi disiplinler oluşturmaktadır ve uygulamaları, makine dili çevirimi, doğal dilde yazılan yazıların işlenmesi ve özetlenmesi, kullanıcı arayüzleri, diller arası çeviri, konuşma tanıma, yapay zeka ve uzman sistemler gibi birçok alandaki çalışmalarda kullanılmaktadır.

Doğal dil işleme süreci temelde “doğal dil anlama” ve “doğal dil üretme” olarak iki ana bölüme ayrılabilir. Doğal dil anlama bölümünde amaç, girdinin işlenmeye ya da kullanışlı bir biçimde kullanıcıya sunulmaya hazırlanması ve dilin ayırıcı yönlerinin tespitidir. Öte yandan doğal dil üretme bölümünde ise istenilen bilgi eldeki veriler ışığında ve tanımlanan dil planları çerçevesinde kullanıcı tarafından anlaşılabilecek bir biçimde ortaya koyulmaktadır. Doğal dil işlemede girdi ve çıktılar sözlü veya yazılı formatta olabilmektedir. Tez kapsamında konuşmanın içeriği irdelenmeden konuşmacılardan toplanan konuşma girdileri kullanılarak, bu girdilerin yazı dilinde kullanıcıya sunulması amaçlanmıştır.

2.2. Konuşma Tanıma

Konuşma işleme, en geniş kapsamda konuşma sinyalleri ve sinyal işleme yöntemlerini barındıran bir çalışma alanıdır. Bu alandaki çalışmalarda sinyaller çoğunlukla dijital gösterimleri üzerinden işlenmektedirler. Yani konuşma işleme dijital sinyal işlemenin konuşma sinyallerine uyarlanan özel bir kolu olarak ifade edilebilmektedir.

Konuşma işleme ile eldeki konuşma verisi analiz edilip sentezlenebilmekte, konuşmanın içeriği tespit edilebilmekte ve eldeki veri minimum kayıpla küçültülecek şekilde kodlanabilmektedir. Konuşma içeriği tespit edilebildiğinde ise konuşmacıyı tanımak, konuşulan dili saptamak ve ses verisinden anlamlı metinler oluşturabilmek mümkün hale gelmektedir. Tezin temel konusu olan konuşma tanıma, konuşulan dilin bilgisayar tarafından yazıya dönüştürülmesini sağlayan metodolijileri ve teknikleri geliştiren bilgisayarlı dilbilimin interdisiplinel bir alanıdır.

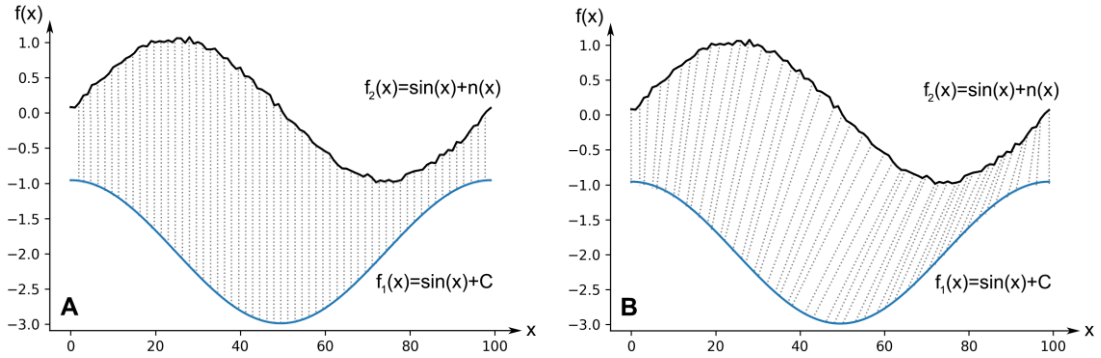


Şekil 2.1. Konuşma işleme adımları ve veri setinin değişiklik gösteren unsurları

Şekil 2.1'de görülebileceği gibi veri setinin (konuşma külliyatının) seçiminde konuşma tanıma işlemini etkileyebilecek birçok etken bulunmaktadır. Özel bir konuşma tanıma sistemi/uygulaması geliştirirken veri setinin özelliklerinin ve geliştirme yönteminin ihtiyaca yönelik olarak belirlenmesi gerekmektedir. Komut algılaması istenilen bir uygulama tasarımında konuşma modu kesik, konuşma tarzı dikte edilen konuşma

verilerinden oluşan bir veri setinin seçimi amaca daha iyi hizmet edebileceği gibi cümle tanımada daha çok spontane konuşulan ve kesintisiz bir veri seti daha işlevsel hale gelmektedir. Bu sebeplerden dolayı veri seti oluşturulurken konuşmacı sayısına, öğe sayısına, (sayı, kelime, cümle, diyalog vb.) konuşma stiline, kayıt ortamına (sessiz, gürültülü, telefon vb.), sinyal frekansına ve sinyale uyarlama yöntemine (kelime, cümle vb.) dikkat etmek, alınan sonuçları büyük ölçüde etkilemektedir.

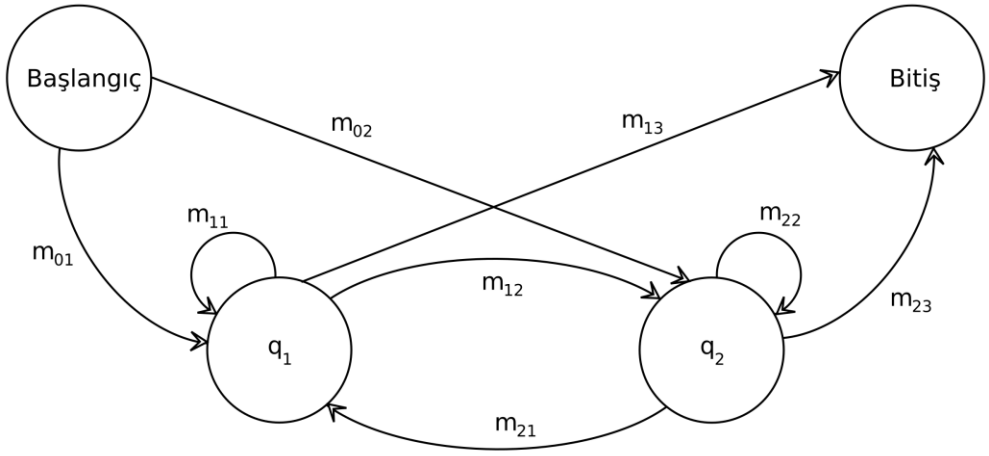
1950'li yıllarda üzerine çalışılmaya başlanan bir alan olan konuşma tanıma, 60'lı yılların sonuna kadar kayda değer bir gelişme kaydedememiştir. Başlangıç yıllarında yapılan öne çıkan çalışmada 1952 yılında Bell Labs araştırmacıları Davis ve ark. (1952) tek konuşmacılı rakam tanıma için her bir ifadenin güç spektrumundaki yerel maksimumlarını (formant) tespit ederek çalışan "Audrey" isimli bir sistem kurmuşlardır. 1962 yılında Uluslararası İş Makineleri (IBM) mühendislerinden Dersch, on altı İngilizce kelimeyi anlayabilen "Shoebox" makinesinin konuşma tanıma özelliğini 1962 Dünya Fuarı'nda sergilemiştir (Anonim 2020). Geliştirilen makine ile ulaşılmak istenen hedef, rakamları ve artı, eksi gibi temel matematiksel operatörleri tanıyarak temel matematik işlemlerini çıktı olarak kullanıcıya sunmaktır. Bu işlem, cihaza bağlı bir mikrofondan komutların algılanarak elektriksel vurulara çevrilmesi, ardından bu vurular bir ölçüm devresi ile sınıflandırılarak röle sistemi aracılığıyla cihaza bağlı bir toplama makinesine iletilmesi şeklinde gerçekleşmektedir. 1966 yılında ilk olarak Nagoya Üniversitesi'nden Itakura ve Nippon Telgraf ve Telefon'dan Saito tarafından Doğrusal Öngörücü Kodlama'nın (LPC) konuşma tanımada kullanılması önerilmiştir (Gray 2010). LPC yöntemi ile ulaşılmaya çalışılan hedef, ses sinyallerinden konuşma verilerini filtreleyebilmektir. Bu dönemdeki konuşma tanıma adına en büyük gelişme ise Sovyet araştırmacıların Dinamik Zaman Atlama (DTW) algoritmasını icat etmeleri olmuştur. DTW algoritması temel olarak iki farklı ses verisini eşleştirmek için en uygun bükülmeleri tespit etmeye dayanmaktadır. Buradaki bükülme, iki sinyalin karşılaştırılması sonucu elde edilen uzaklığı temsil etmektedir. Aynı kişi tarafından söylenen aynı kelimeler, farklı şekilde telaffuz edilebilecekleri için benzer ancak farklı sinyaller ortaya koyabilmektedirler. DTW algoritması bu kelimeleri eşleştirebilmek için sinyalleri uygun şekilde ayarlamakta ve sinyaller arasındaki en kısa mesafeyi bulmaya çalışmaktadır (Paliwal ve ark. 1982).



Şekil 2.2. Sinyal hizalamaları **A)** Doğrusal hizalama **B)** DTW algoritması ile hizalama

Giorgino'nun (2009) çalışmaları ile hazırlanan dtw-python paketi kullanılarak oluşturulan Şekil 2.2.'de verilen iki farklı sinyalin benzerlik oranları hesaplanırken DTW algoritmasının hizalama prensibi görülebilmektedir. Araştırmacılar bu yöntem sayesinde 200 kelimelik bir kelime dağarcığı üzerinde çalışabilen bir konuşma tanıma sistemi oluşturmuşlardır ancak konuşmacıdan bağımsız tanıma sorunu çözümsüz kalmıştır. Yine aynı dönemlerde Stanford Üniversitesi'nde yüksek lisans öğrencisi olarak konuşma tanıma üzerine çalışan Reddy ve ark. (1974), satranç oynamak için sözlü komutları tanıyabilen ve akıcı konuşmaları geçmiş araştırmalara göre daha iyi algılayabilen bir sistem geliştirmiştir.

1960'larda Baum ve Petrie'nin (1966), Savunma Analizi Enstitüsü'nde (IDA) Markov zincirlerinin matematiğini geliştirmesinin ardından, Carnegie Melon Üniversitesi'nden Baker (1975) bu araştırmaları temel alarak, konuşma tanıma için Saklı Markov modelini (HMM) kullanmaya başlamıştır (Rabiner 1989). HMM'nin temeli, Markov zincirini büyütmeye dayanmaktadır (Jurafsky ve Martin 2020). Bir Markov zinciri, her biri belirtilen bir kümeden değerler alabilen rastgele değişken dizilerinin olasılıklarını bulmaya yarayan bir modelden oluşmaktadır. Bu kümeler sözcükler, etiketler gibi herhangi bir şeyi temsil eden semboller olabilmektedir. Markov zincirlerinde mevcut duruma bakarak gelecek durum hakkında tahminde bulunmak mümkündür ancak mevcut durumdan önceki durumların gelecek durumların tahmini üzerinde herhangi bir etkisi yoktur.

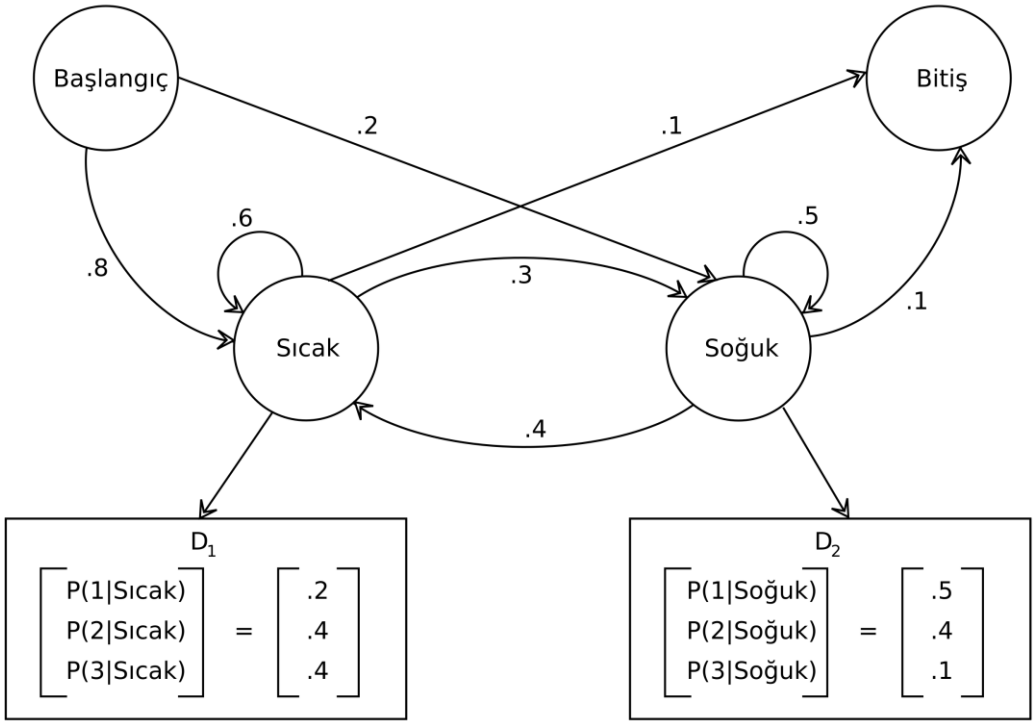


Şekil 2.3. Markov zincirinde durumlar ve olasılıklar

$Q = q_1, q_2, \dots, q_N$ olarak tanımlanan bir durum değişken sekansı ele alındığında Markov varsayımı Denklem 2.1'deki gibi hesaplanabilmektedir. Şekil 2.3'te grafikteki düğümler durumları, kenarlar ise geçiş olasılıklarını temsil etmektedir. Markov zincirini oluşturmak için $M = m_{01}, m_{02}, \dots, m_{n1}, \dots, m_{nn}$ geçiş olasılık matrisi oluşturulmaktadır. Her bir m_{ij}, i'inci durumdan j'inci duruma geçiş olasılığını temsil etmektedir.

$$\mathbb{P}(q_i = m | q_1 \dots q_{i-1}) = \mathbb{P}(q_i = m | q_{i-1}) \tag{2.1}$$

Markov zincirleri hava durumu tahmini gibi gözlemlenebilen durumların olasılığını hesaplamak için kullanışlıdır fakat karşılaşılabilecek her durumda yeterli olamamaktadır. Örneğin Eisner'in (2002) hava durumu tahmini modelinde hava durumu tahmin edilecek bölgenin herhangi bir geçmiş verisi bulunmamaktadır. Tahmin için eldeki tek kaynak bir bölge sakininin o gün kaç dondurma yediğini içeren bir günlüktür. Bu günlükten dolaylı olarak havanın sıcak ya da soğuk olduğu tahmin edilmek istendiğinde, 'O' belirli bir günde yenen dondurma sayısı, 'Q' dondurma yenilmesine neden olan 'sıcak' ve 'soğuk' durumlarının saklı sekansının hesaplanması gerekmektedir (Şekil 2.4). Bu süreçte oluşturulan ve çıktı olasılıkları (emission probability) olarak da adlandırılan $D = d_i(o_t)$ gözlem olasılıkları dizisi, her biri bir i durumundan üretilen o_t gözlem olasılıklarını içermektedir.



Şekil 2.4. Hava durumu tahmini için oluşturulan Saklı Markov modeli

HMM'lerin konuşma tanımada kullanımı, araştırmacıların akustik, dil ve sözdizimi gibi farklı bilgi kaynaklarını birleşik bir olasılık modelinde birleştirmesine olanak sağlamıştır. 1980'li yıllarda HMM'lerin popülaritesinin artmasıyla birlikte IBM çalışanı Fred Jelinek ve takımı HMM'den faydalanarak "Tangora" adlı sesle etkinleştirilen bir daktilo icat etmişlerdir (Anonim 2020). 1980'li yılların ortalarına gelindiğinde Tangora, bu modelleme tekniği ile yirmi bin kelimelik bir kelime dağarcığını işleyebilecek bir duruma gelmiştir ve oluşturulan daktiloya bir grafik ekran eklenmiştir. Jelinek'in istatistiksel yaklaşımı, insan beyninin dili algılama biçimine daha az önem vermiştir ve konuşma tanımayı HMM gibi istatistiksel bir modelleme yöntemi kullanarak anlamaya çalışmıştır. Bu yaklaşım konuşma dilinin birçok ortak özelliğini hesaba katamayacak kadar basit olduğundan dilbilimciler tarafından tartışma konusu olmuştur (Huang ve ark. 2014). Bununla birlikte, HMM'nin konuşmayı modellemede oldukça kullanışlı bir yol olduğu kanıtlanmıştır ve 1980'lerde DTW'nin yerini alarak konuşma tanımada kullanılan baskın algoritma haline gelmiştir.

1990'ların sonununa gelindiğinde, konuşma tanımada olasılık modellerinin yanı sıra yapay zeka alanındaki gelişmeler sayesinde çoğunlukla ileri beslemeli yapay sinir ağları (YSA) ile birleştirilen HMM'ler gibi gelenekselleşmiş yaklaşımlar kullanıldığı görülmektedir (Bourlard ve Morgan 1994). Tarih boyunca birçok farklı yaklaşım kullanılmış olsa da bu alanda çığır açan bir yenilik olan YSA ve derin öğrenme kavramlarının konuşma tanıma sistemlerine adaptasyonu ile elde edilen başarılar büyük ölçüde artmıştır. Günümüzdeki konuşma tanıma çalışmalarının çoğu, Hochreiter ve Schmidhuber (1997) tarafından yayınlanan ve özelleşmiş bir tekrarlayan sinir ağı (TSA) olan uzun kısa süreli hafıza (LSTM) adı verilen derin öğrenme yöntemini kullanmaktadır. LSTM yöntemini kullanan TSA'lar, kaybolan gradyan probleminin önüne geçmekte ve binlerce adım önce meydana gelen olaylardan dahi faydalanarak derin öğrenmeyi gerçekleştirebilmektedir. Bu yenilik, araştırmacılar tarafından hızla benimsenmiştir ve akustik modelleme, dil modelleme gibi görevlerde derin öğrenme teknikleri kullanılmaya başlanmıştır.

2.3. Yapay Zeka Uygulamalarında Öğrenme Türleri

Makine öğrenmesi ve yapay zeka uygulamalarındaki ana hedef, deneyimlerden yola çıkarak bilgi edinmek ve bu bilgilerden yararlı kavramları sentezlemektir. Bu doğrultuda gerçekleştirilebilecek kapsamlı görevler olabileceği gibi yalnızca belirli bir işlevi yerine getiren daha küçük çaplı görevler de mevcuttur. Bu geniş yelpazede genel geçer tek bir yöntem olamayacağından yapay zeka alanındaki uygulamalarda da birçok farklı yaklaşım ve yöntem geliştirilmiştir. Bu yöntemlerin ayrı ayrı ya da birlikte kullanımıyla çeşitlendirilmesi mümkün olsa da öğrenme türleri, öğrenme için kullanılan veri seti türü ve problem çözme yaklaşımına göre denetimli öğrenme, denetimsiz öğrenme ve takviyeli öğrenme olarak üç temel gruba ayrılabilmektedir.

Denetimli öğrenme: Denetimli öğrenmede oluşturulan sinir ağı etiketli bir veri setini eğitim verisi olarak kullanmakta ve görülemeyen durumlar için tahminlerde bulunmaktadır (Mohri ve ark. 2018). Bu tip öğrenmenin amacı, her girdi için istenen çıktıyı üretmektir. Bu yüzden öğrenme süreci boyunca tahmin edilen çıktının olması gereken çıktıdan ne kadar uzak olduğunun ölçütü olan maliyet fonksiyonunun, yanlış çıkarımları en aza indirecek şekilde olması hedeflenmektedir. Örüntü tanıma ve regresyon gibi problemler denetimli öğrenmeye uygun problemlerdir. Denetimli

öğrenme, aynı zamanda konuşma tanıma, hareket tespiti gibi sıralı verilere de uygulanabilmektedir. Bu yöntem, işlem sürecinde mevcut aşamaya kadar elde edilen çözümlerin kalitesi hakkında sürekli geri bildirim sağlayan bir öğretmen vasıtasıyla öğrenme gibi düşünülebilir.

Denetimsiz öğrenme: Denetimsiz öğrenmede amaç etiketsiz verilerden oluşan bir veri seti ile öğrenme algoritmasını eğitmektir (Nassif ve ark. 2019). Buradaki ana hedef, verideki temel yapıyı ya da dağıtım modellerini belirleyerek veriler hakkında daha fazla bilgi edinmektir. Etiketsiz verilerden kendi başına anlam çıkaran algoritma, tüm girdilerin istatistiksel yapısını yansıtan belirli bir girdi örüntüsü oluşturmaya çalışmaktadır. Bu sayede farklı girdiler, her bir girdi nesnesinden çıkarılan özelliklere göre kümelenmektedir (Rojas 1996). Denetimsiz öğrenme yöntemi istatistiksel dağılım, sıkıştırma, kümeleme, filtreleme gibi problemlerin çözümünde kullanılabilen bir yöntemdir.

Takviyeli öğrenme: Takviyeli öğrenmenin temelleri etkileşim ile öğrenmeye dayanmaktadır (Sutton ve Barto 2018). Her bir zaman bloğunda, kullanıcı bir eylem gerçekleştirmekte ve ortam, bazı kurallara göre bir gözlem ve anlık bir maliyet oluşturmaktadır. Bu tip öğrenmede ortam genellikle durumlar $q_1, q_2, q_3, \dots q_n \in Q$ ve eylemler $a_1, a_2, a_3, \dots a_n \in A$ olacak şekilde Markov karar süreci şeklinde modellenebilmektedir ve amaç, ağırlıkları kümülatif maliyeti en aza indiren eylemleri gerçekleştirecek şekilde yapılandırmaktır. Durum geçişleri bilinmediğinden x girdisinin anlık maliyet dağılımı $\mathbb{P}(c_t|\, q_t)$, gözlem dağılımı $\mathbb{P}(x_t|\, q_t)$ ve geçiş dağılımı $\mathbb{P}(q_{t+1}|q_t, a_t)$ kullanılmakta ve akış süreci eylemler üzerinden koşullu dağılım ile gözlemlenecek şekilde tanımlanmaktadır. Takviyeli öğrenme ile YSA'lar ve HMM'lerin birlikte kullanımı araç yönlendirme, video oyunları, doğal kaynak yönetimi vb. sıralı karar verme görevlerinin gerçekleştirilmesinde büyük kolaylık sağlamaktadır.

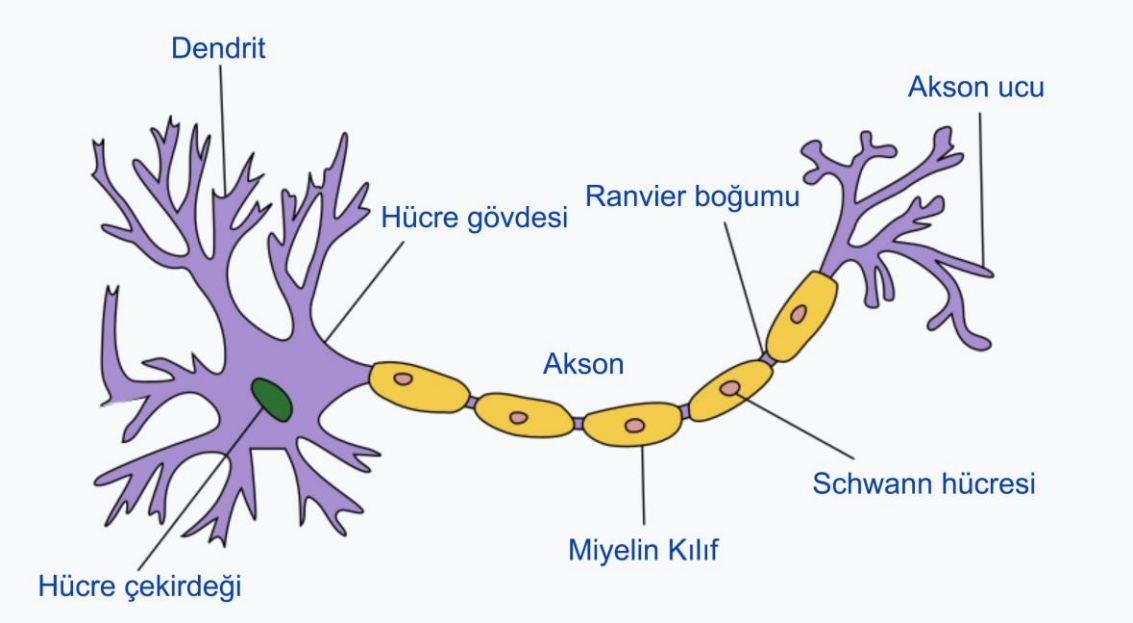
2.4. Derin Öğrenme

Yapay zekanın alt alanı olan derin öğrenme, insan beyninin bilgileri işleme şeklini ve bu bilgilerden bir karara varmak için kendi içinde oluşturduğu örüntüleri taklit etmektedir. Teknolojideki gelişmelerle birlikte internet kullanımının artması, birçok sektörde verilerin sanal ortama taşınması, sosyal medya uygulamaları ile kişisel bilgilere

kolaylıkla erişilebilmesi dünyanın her yerinden farklı formatlarda muazzam boyutlarda veri elde edilebilmesine olanak sağlamıştır. Büyük veri olarak adlandırılan bu veriler, çok çeşitli kaynaklardan toplanabilmekte ve paylaşılabilmektedir. Bu kadar çeşitli, büyük miktarda ve dağınık olan veriyi insan beyni ile analiz etmek ve karmaşık verilerden çıkarımlarda bulunmak yıllar sürebileceğinden bu tarz görevlerin gerçekleştirilmesinde derin öğrenme algoritmalarının kullanımı gün geçtikçe yaygınlaşmaktadır.

2.4.1. Yapay sinir ağları

YSA, insan ve hayvan beyinlerini oluşturan biyolojik sinir ağlarından esinlenilerek oluşturulan bilgi işlem sistemleridir (Chen ve ark. 2019). Bir YSA, biyolojik bir beyine ait olan ve Şekil 2.5'te görülebilen nöronları ilkel bir şekilde modelleyen ve yapay nöronlar adı verilen bağlı birimler veya düğümlerin bir araya gelmesinden oluşmaktadır. İnsan beynindeki nöronlar arasındaki bağlantılar, ilgili hücrenin sinapsları vasıtasıyla bağlı olduğu diğer nöronlara sinyal iletmesi şeklinde gerçekleşmektedir. Biyolojik sinir hücreleri temel alınarak geliştirilen yapay bir nöron da sinyal aldığında onu işlemekte ve ona bağlı nöronlara sinyal göndermektedir. Yapay nöronlar arasındaki bağlantılardaki sinyaller sayısal değerlerdir ve her bir nöronun çıktısı, o nöronun girdilerinin toplamının doğrusal olmayan bir fonksiyonu tarafından hesaplanmaktadır. Nöronlar ve kenar adı verilen bağlantılar, öğrenme sürecinin gerçekleşmesi amacıyla süreç boyunca ayarlanan ağırlıklara sahiplerdir. Ağırlık, bir bağlantıdaki sinyalin gücünü artırmakta veya azaltmaktadır. Bu sayede ağırlıklar temel alınarak doğru çıkarımların yapılması hedeflenmektedir. Farklı durumlarda öğrenme işlemlerini gerçekleştirmek adına yapay nöronların ayrıca bir eşik değeri de olabilmektedir. Öyle ki bir sinyal yalnızca toplam sinyal bu eşiği geçerse gönderilebileceği şekilde aktarım kısıtlanabilmektedir ve öğrenme süreci istenilen doğrultuda ilerleyebilmektedir. YSA'larda nöronlar genellikle katmanlar halinde toplanmaktadırlar ve farklı katman girdileri üzerinde farklı dönüşümler gerçekleştirilebilmektedir. Sinyaller çoğunlukla katmanları birden çok kez geçerek ilk katmandan (girdi katmanı) son katmana (çıktı katmanı) ulaşmaktadırlar.

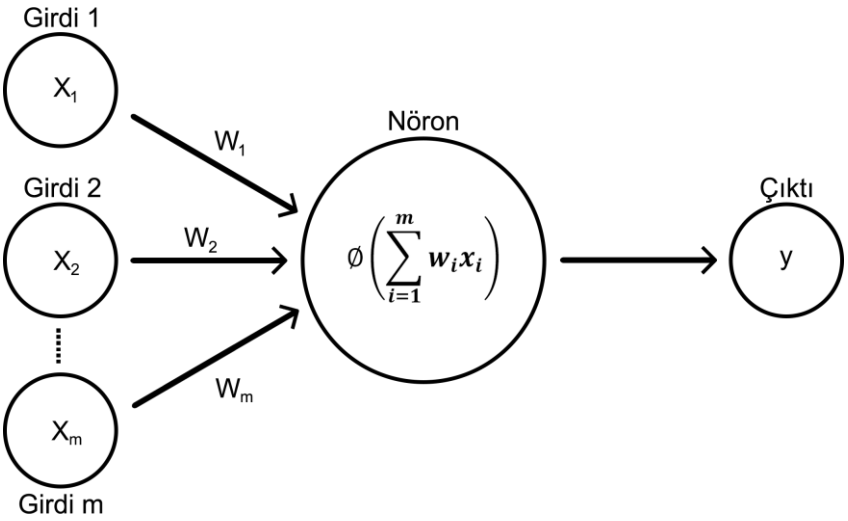


Şekil 2.5. Biyolojik sinir hücresi (Anonim 2020)

2.4.2. Yapay nöron yapısı

YSA'lar kavramsal olarak biyolojik nöronlardan türetilen Şekil 2.6'daki gibi yapay nöronlardan oluşmaktadır. Her bir yapay nöronun girdileri vardır ve diğer birçok nörona gönderilebilen tek bir çıktı üretmektedir. Girdiler, görüntüler veya belgeler gibi harici bir veri örneğinin değerleri veya diğer nöronların çıktıları olabilmektedir. Nörona gelen değerler, programlama kısıtları içinde işlemden geçirilir ve nöron çıktısı üretilir. Nöronlarda gerçekleşen işlemler sonucunda yapay bir sinir ağının son çıkış nöronlarının çıktıları ise, bir görüntüdeki bir nesneyi tanımak ya da konuşma sinyalinden kelime tespiti gibi görevleri yerine getirmektedir.

Nöronun çıktısını bulmak için, öncelikle tüm girdilerden nörona olan bağlantıların ağırlıklarının ağırlıklı toplamı hesaplanmaktadır. Ardından bu toplama bir sapma eklenmektedir. Elde edilen ağırlıklı toplam aktivasyon olarak da adlandırılabilmektedir. Bu ağırlıklı toplam daha sonra çıktıyı üretmek için genellikle doğrusal olmayan bir aktivasyon fonksiyonundan geçirilmektedir. İlk girdiler, resimler ve sayılar gibi harici verilerdir. Son çıktılar ise yüz tanıma, sınıflandırma, ses tanıma gibi ağın programlanma amacına uygun sonuçlardan oluşmaktadır.



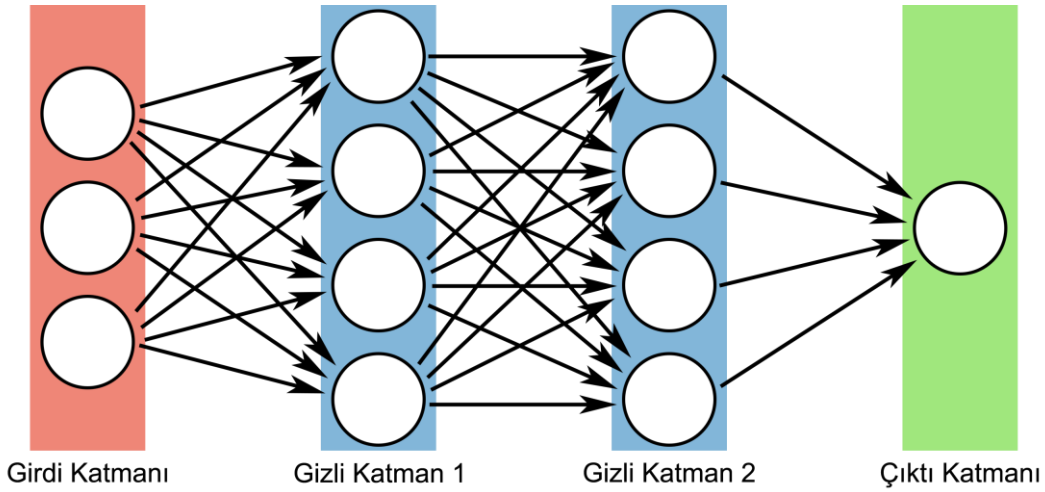
Şekil 2.6. Yapay nöron yapısı

2.4.3. Yapay sinir ağlarında öğrenme

Yapay sinir ağları, her biri bilinen bir girdi ve çıktı içeren örnekleri, girdi ve çıktı arasında olasılık ağırlıklı ilişkiler oluşturarak ve bu ilişkileri kendi bünyesinde depolayarak işlemekte ve eğitim adı verilen öğrenme sürecini temel olarak bu şekilde gerçekleştirmektedir. Bir sinir ağının eğitimi, genellikle ağın işlemler sonucunda elde edilen çıktısı ile bir hedef çıktı arasındaki farkı belirleyerek gerçekleştirilir. Bu fark bize maliyeti vermektedir. Ağ, bünyesindeki ağırlıklı ilişkilerini bir öğrenme kuralına göre ve hesaplanan maliyet değerini kullanarak ayarlamaktadır. Birbirini izleyen ayarlamalar ile sinir ağının, hedef çıktıya giderek daha fazla benzeyen çıktılar üretmesi hedeflenmektedir. Yeterli sayıda ayarlamadan sonra eğitim belirli ölçütlere göre sonlandırılmaktadır.

Şekil 2.7'dekine benzer bir yapıya sahip olan bu tür sistemler, genellikle gerçekleştirilmek istenen göreve özgü kurallarla programlanmak yerine, örnekleri dikkate alarak görevleri gerçekleştirmeyi öğrenmektedirler. Örnek olarak görüntü tanıma alanında YSA kullanılarak geliştirilmek istenen bir uygulamada manuel olarak "köpek" veya "köpek değil" olarak etiketlenen örnek görüntüler analiz edilerek elde edilen sonuçlar diğer görüntülerdeki köpekleri tanımlamak için kullanılabilmekte ve köpekleri içeren görüntüleri tanımlama işlemi öğrenilebilmektedir. Bu işlem YSA sayesinde köpeğe dair önceden öğrenilmiş belirleyici bilgiler olmadan işlenen örneklerden otomatik olarak köpeğe dair tanımlayıcı özellikler üretilerek gerçekleştirilmektedir. Konuşma

tanımada da sistem benzer şekilde ilerlemektedir. Bir sinir ağı, girdi olarak verilen konuşma verilerine dair kimin konuştuğu, söylenen kelimenin ek özellikleri, konuşanın duygu durumu gibi özelliklere ait spesifik tanımlamalar yapılmamasına karşın, yeterli sayıda etiketli veri kullanılarak eğitilirse bu çıkarımları istatistiksel metotlar ve algoritmalar ile tespit edebilmektedir.



Şekil 2.7. Temsili bir yapay sinir ağı yapısı

2.4.4. Aktivasyon fonksiyonu

Aktivasyon fonksiyonu, ağırlıklı toplamı hesaplayarak ve buna daha fazla sapma ekleyerek bir nöronun aktive edilip edilmeyeceğine karar vermektedir. Aktivasyon fonksiyonunun amacı, bir nöronun çıktısının doğrusal olmasını engellemektir ve genel denklemi

$$f(x) = \sum_{i=1}^{m} (w_i x_i) + b$$

olarak tanımlanabilmektedir.

Bir sinir ağında, nöronların ağırlıkları ve sapma değerleri çıktıdaki maliyete göre güncellenmektedir ve bu süreç, geri yayılma olarak bilinmektedir. Aktivasyon fonksiyonları, ağırlıklar ve sapma değerleri güncelleme maliyeti ile birlikte gradyanlar da sağladığı için geri yayılma mümkün olmaktadır. Veri setinin boyutu, içeriği, kullanılan yöntem ve gerçekleştirilmek istenen görev göz önünde bulundurularak farklı tip görevler

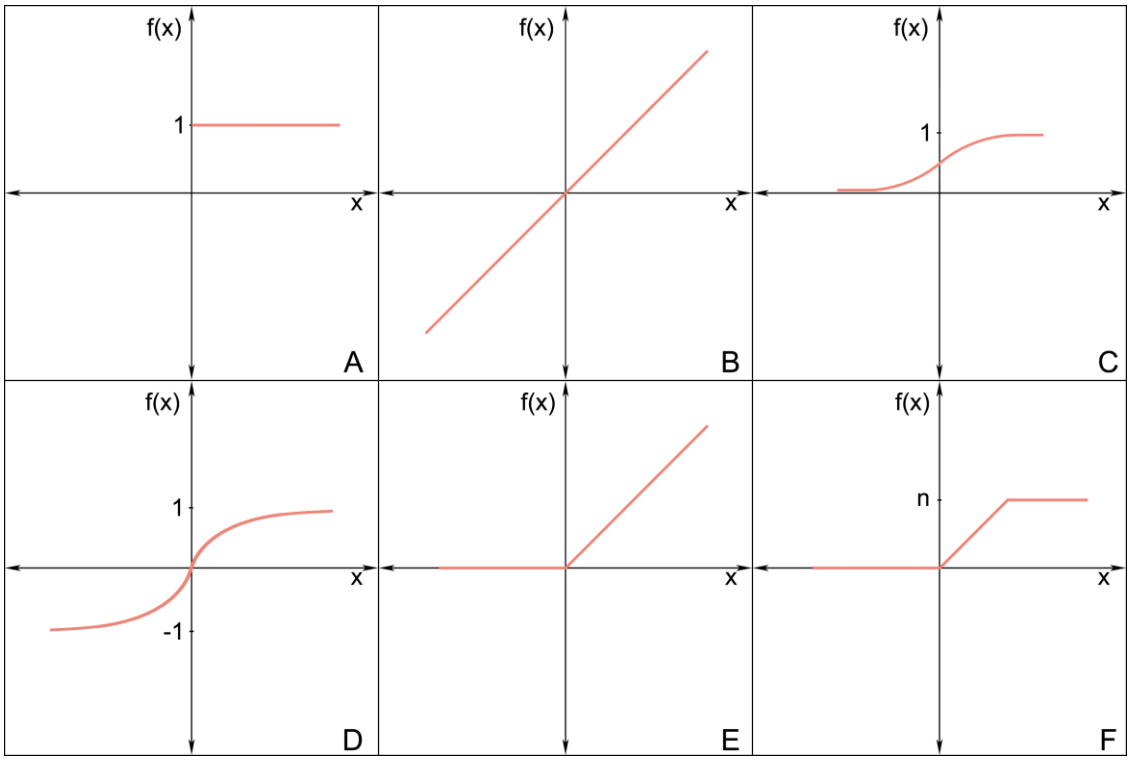
için farklı tip aktivasyon fonksiyonları kullanılmaktadır. Çizelge 2.1'de sıklıkla kullanılan aktivasyon fonksiyonları ve matematiksel ifadeleri, görülebilmektedir.

Çizelge 2.1. Sık kullanılan aktivasyon fonksiyonlarının matematiksel ifadeleri

Aktivasyon Fonksiyonu	**Denklem**	**Kapsadığı Aralık**
İkili Basamak Fonksiyonu	$f(x) = \begin{cases} x < 0 \; için \; 0 \\ x \geq 0 \; için \; 1 \end{cases}$	{0,1}
Doğrusal Fonksiyon	$f(x) = x$	(-∞,∞)
Sigmoid Fonksiyon	$f(x) = \frac{1}{1 + e^{-x}}$	(0,1)
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	(-1,1)
ReLu	$f(x) = \begin{cases} x < 0 \; için \; 0 \\ x \geq 0 \; için \; x \end{cases}$	[0,∞]
Kırpılmış ReLu	$f(x) = \begin{cases} x < 0 \; için \; 0 \\ 0 \leq x < n \; için \; x \\ x \geq n \; için \; n \end{cases}$	[0, n]

Farklı aktivasyon fonksiyonları, farklı sonuçlar üreteceğinden, tasarlanan yapıda probleme en uygun aktivasyon fonksiyonun kullanılması gerekmektedir. Çizelge 2.1'de açık formülü gösterilen fonksiyonlardan ikili basamak fonksiyonu yalnızca iki değer alabileceği için yalnızca iki sınıfın olduğu sınıflandırma işlemlerinde etkin kullanılabilmektedir. Doğrusal fonksiyon, YSA'larda kolay kullanım sağlamasına rağmen geriye yayılım esnasında türevi sabit kalacağı için gradyanda bir değişiklik olmamaktadır. Maliyet oranı güncellenemediğinden bu yöntem birden fazla katmanlı yapılarda fazla kullanışlı değildir. Sigmoid fonksiyon yaygın kullanılan aktivasyon fonksiyonlarından biridir. Doğrusal olmayan ve 0 ile 1 arasında değer alabilen bu fonksiyonda x değişkenindeki bir farklılık fonksiyon sonucunda büyük değişiklik gösterebilmektedir. Ancak fonksiyonun türevi (Şekil 2.8c) belirli bir noktadan sonra 0'a yaklaşmakta ve öğrenim durmaktadır. Hiperbolik tanjant fonksiyonu da davranış özellikleri bakımından sigmoid fonksiyona benzemektedir ve sigmoid fonksiyonun biraz

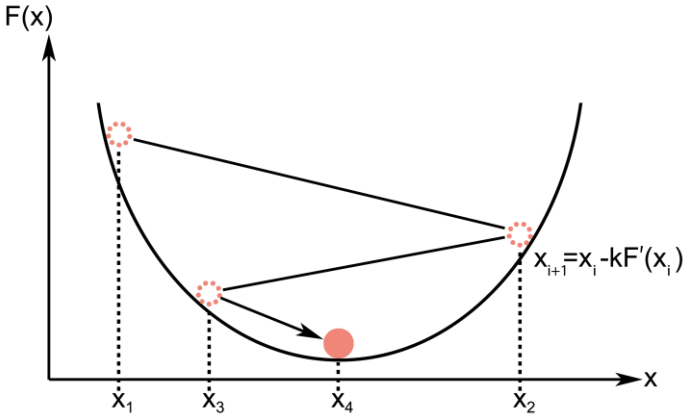
daha genişletilmiş bir halidir. ReLu fonksiyonu yine yaygın kullanılan aktivasyon fonksiyonlarından biridir. Tüm nöronları aynı anda aktive etmeyerek ağı seyrekleştirdiği için bu sayede hem performansı hem de hesaplama hızını arttırmaktadır.



Şekil 2.8. Aktivasyon fonksiyonu grafikleri **A)** İkili Basamak Fonksiyonu **B)** Doğrusal Fonksiyon **C)** Sigmoid Fonksiyon **D)** ReLu **E)** Kırpılmış ReLu

2.4.5. Gradyan inişi

Gradyan inişi, maliyet fonksiyonunu (maliyeti) en aza indiren bir *F* fonksiyonunun parametrelerinin değerlerini (katsayıları) bulmak için kullanılan bir optimizasyon algoritmasıdır. Şekil 2.9'daki fonksiyon grafiğini bir kaseye benzetirsek bu kaseye atılan topun birkaç sekmeden sonra kasenin dibine ulaşacağı metaforundaki gibi gradyan inişinde de temel amaç en az maliyete ulaşmaktır.

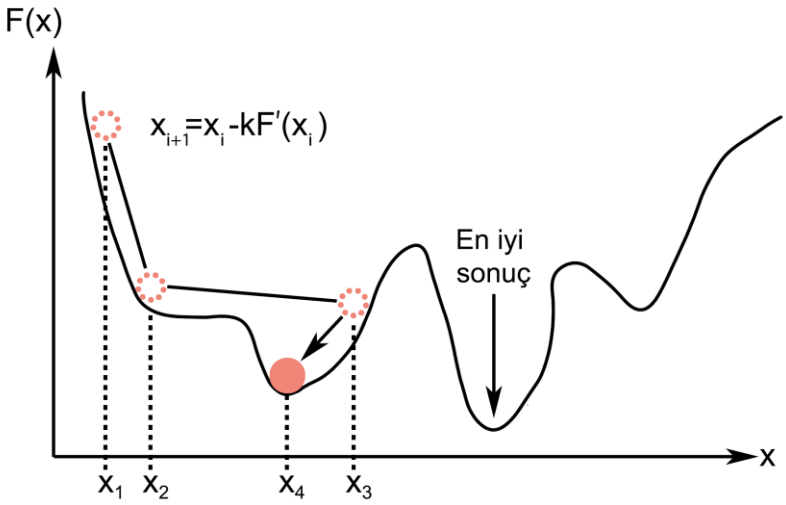


Şekil 2.9. Gradyan inişi iki boyutlu gösterimi

Gradyan inişi, eğer çok değişkenli fonksiyon $F(x)$ bir k noktasının çevresinde tanımlanmış ve türevlenebilir ise, o zaman k noktasından bu noktadaki F'nin negatif gradyanı yönünde gidilirse $(-\nabla \mathrm{F(k)})$, $F(x)$'in azalma hızı maksimum olur prensibine dayanmaktadır. Buradan yola çıkarak eğer adım boyutu $\gamma \in \mathbb{R}_+$ için $k_{n+1} = k_n - \gamma \nabla F(k_n)$ yeterince küçükse $F(k_n) \geq F(k_{n+1})$ olmaktadır. $\gamma \nabla F(k)$ teriminin k'den çıkarılmasının nedeni, gradyan karşısında yerel minimuma doğru hareket etmek istenmesidir. Bu durumda x_0'ın, F'nin yerel minimumu olduğu varsayımıyla yola çıkıldığında, x değerleri $x_{n+1} = x_n - \gamma_n \nabla F(x_n), n \geq 0$ koşulunu sağlayan $x_0, x_1, x_2, \ldots$ şeklinde bir dizi elde edilmektedir. Bunun sonucunda Denklem 2.2'deki gibi tekdüze bir dizi oluşmaktadır ve x_n dizisinin istenen yerel minimuma yakınsaması hedeflenmektedir.

$$F(x_0) \geq F(x_1) \geq F(x_2) \geq \cdots \tag{2.2}$$

γ adım boyutunun değerinin her yinelemede değişmesine izin verildiğini de göz önünde bulundurarak, F fonksiyonunun dışbükey olduğu ve ∇F'nin Lipschitz koşulunu sağladığı durumlarda yerel ya da global minimuma yakınsama garanti edilebilmektedir. Yani F fonksiyonu dışbükey olduğunda gradyan inişi global çözüme yakınsayabilir.



Şekil 2.10. Gradyan inişinde yerel minimum

F fonksiyonunun dışbükey olmadığı durumlarda bulunan sonuç yerel minimumda takılıp optimal sonuca ulaşmanın önüne geçilebilmektedir. Yerel minimuma takılmış olan bir gradyan inişi örneği Şekil 2.10'da görülebilir. Yerel minimumda takılma ihtimali stokastik gradyan inişi ile düşürülebilmektedir. Stokastik gradyan inişinin gradyan inişinden en temel farkı kullanılan gradyanın, gradyan inişindeki gibi tüm veri seti kullanılarak oluşturulan gerçek gradyan yerine tüm veri setinden rastgele veriler seçilerek oluşturulan daha küçük bir veri setini kullanarak oluşturulan yaklaşık bir gradyan değeri olmasıdır. Bu sayede daha farklı yerlerden veriler toplanılarak yerel minimumdan kaçmak mümkün olup, hesaplama yükü azalacağından çalışma hızı da artmaktadır.

2.4.6. Geriye yayılım

Geriye yayılım, ağırlıkları temel alarak ileri beslemeli bir sinir ağının kayıp fonksiyonunun gradyanını hesaplamaktadır. Yapay sinir ağlarının eğitiminde başlangıç ağırlıkları rastgele belirlenmektedir ve sinir ağının istenilen görevi öğrenebilmesi için bu ağırlıkların eğitim sürecinde güncellenmesi gerekmektedir. Eğitim sürecinde her bir girdiden elde edilen çıktı ile orijinal çıktı arasındaki fark geriye yayılım ile ağırlıklar güncellenerek minimize edilmektedir. Geriye yayılımın çalışma prensibi özetlemek gerekirse bir sinir ağına (x, y) girdi ve çıktı çifti verildiğinde kayıp;

$$C\left(y, f^L\left(W^L f^{L-1}\left(W^{L-1} \dots f^2\left(W^2 f^1\left(W^1 x\right)\right) \dots\right)\right)\right)$$

olarak gösterilebilmektedir. Kaybı hesaplamak için x girdisi ile başlanarak her katmanın ağırlıklı girdisi z^L ve katman L'nin çıktısı a^L aktivasyonu olacak şekilde ileri yönde işlem yapılmaya başlanmaktadır. Geri yayılım için ise a^L aktivasyonunun yanı sıra z^L'de değerlendirilen $(f^L)'$ türevlerinin de daha sonra kullanılabilmesi için hafızaya alınması gerekmektedir. Girdilere göre kaybın türevi zincir kuralı ile hesaplanmaktadır.

$$\frac{dC}{da^L} \cdot \frac{da^L}{dz^L} \cdot \frac{dz^L}{da^{L-1}} \cdot \frac{da^{L-1}}{dz^{L-1}} \cdot \frac{dz^{L-1}}{da^{L-2}} \ \dots \ \frac{da^1}{dz^1} \cdot \frac{\partial z^1}{\partial x} \tag{2.3}$$

Denklem 2.3'te elde edilen terimler kayıp fonksiyonunun türevi ve aktivasyon fonksiyonlarının türevlerine tekabül ettiğinden ağırlık matrisi fonksiyonu Denklem 2.4'teki gibi düzenlenebilmektedir.

$$\frac{dC}{da^L} \cdot (f^L)' \cdot W^L \cdot (f^{L-1})' \cdot W^{L-1} \ \dots \ (f^1)' \cdot W^1 \tag{2.4}$$

∇ gradyanı, çıktının türevinin girdi cinsinden transpoze halidir. Bu nedenle matrisler transpoze edilerek çarpma sırası tersine çevrilmekte ve Denklem 2.5 elde edilmektedir.

$$\nabla_x C = (W^1)^T \cdot (f^1)' \dots \cdot (W^{L-1})^T \cdot (f^{L-1})' \cdot (W^L)^T \cdot (f^L)' \cdot \nabla_{a^L} C \tag{2.5}$$

Geriye yayılım kavramı temel olarak, Denklem 2.5'i yol üzerindeki her katmanda gradyanın hesaplanarak geriye doğru değerlendirmekten oluşmaktadır. Hesaplanan ağırlıklar gradyanından faydalanılarak L katmanındaki maliyet olarak tanımlanabilecek kısmi çarpımlar için yardımcı miktarı δ^L Denklem 2.6'da ifade edilmektedir.

$$\delta^L := \ (f^L)' \cdot (W^{L+1})^T \ \dots \cdot (W^{L-1})^T \cdot (f^{L-1})' \cdot (W^L)^T \cdot (f^L)' \cdot \nabla_{a^L} C \tag{2.6}$$

δ^L, L düzeyindeki düğüm sayısına eşit uzunlukta bir vektördür ve her bir terim, ilgili düğümün değerine atfedilen maliyettir. Bu durumda L katmanının ağırlık gradyanı $\nabla_{W^L} C = \delta^L (a^{L-1})^T$ şeklinde ifade edilebilmektedir. Girdiler sabit, ağırlıklar değişkendir. δ^L vektörü öz yinelemeli olarak Denklem 2.7 şeklinde ifade edilebilmektedir ve ağırlık gradyanları, her katman için matris çarpımlarından faydalanılarak hesaplanabilmektedir.

$$\delta^{L-1} := \ (f^{L-1})' \cdot (W^L)^T \cdot \delta^L \tag{2.7}$$

Geriye yayılım ile ileriye dönük bilgi işleme arasında iki temel farklılık bulunmaktadır. İlk olarak δ^{L-1} vektörünü δ^{L} cinsinden hesaplamak, L katmanlarının tekrar tekrar çarpılmasının önüne geçmektedir. Ayrıca maliyeti geriye doğru yaymak, her adımın bir vektörü (δ^{L}) ağırlık matrisleriyle $(W^{L})^{T}$ ve aktivasyonların türevleriyle $(f^{L-1})'$ çarpmak anlamına gelmektedir. Bunun tersine daha önceki bir katmandaki değişikliklerden başlayarak ileriye doğru çarpmak, her çarpımın iki matrisin çarpılması ile sonuçlanması demektir ve bu durum daha fazla işlem yüküne sebep olmaktadır.

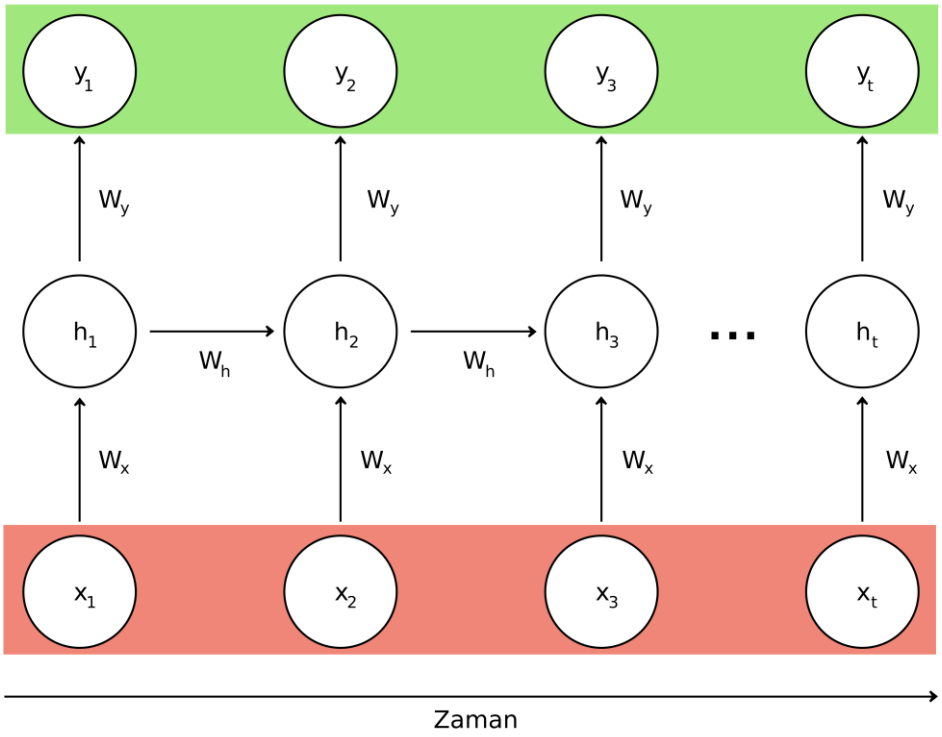
Geriye yayılım süreci, bütün bir YSA eğitim süreci içinde ele alınırsa, süreç işleyişi aşağıdaki gibi 5 temel adım ile özetlenebilmektedir:

- Eğitim süreci ağırlıklara sıfıra yakın (sıfıra eşit olmayan) değerler verilerek veri setindeki ilk girdi ile başlatılmaktadır.
- İleri yayılım: Nöronlar soldan sağa her bir nöronun etkisi ağırlıklarla sınırlı olacak şekilde etkinleştirilmekte ve çıktı katmanına kadar aktivasyon ilerletilmektedir. Ardından tahmin edilen sonuç gerçek sonuçla karşılaştırılarak hata oranı ölçülmektedir.
- Geri yayılım: Ölçülen hata oranı sağdan sola geriye yayılmaktadır. Bu süreçte ağırlıklar hatadan ne kadar sorumlu olduklarına göre güncellenmektedir. Ağırlıkların ne kadar güncelleneceğine öğrenme oranı karar vermektedir.
- Veri setindeki her bir girdi için ileri yayılma ve geri yayılma işlemlerinin sırasıyla tekrar edilmesi gerekmektedir. Ağırlıklar her gözlemden sonra (pekiştirmeli öğrenme) veya bir grup gözlemden sonra (toplu öğrenme) güncellenmektedir.
- Tüm veri seti YSA'dan geçerek tamamlandığında bir iterasyon gerçekleşmiş olmaktadır ve belirtilen iterasyon sayısına ya da koşula ulaşılana kadar iterasyonlar devam etmektedir.

2.4.7 Tekrarlayan sinir ağları

Geri besleme bağlantıları olan bir sinir ağı şeklinde ifade edilebilen ve sıralı veya zamanla değişen kalıpları öğrenmek için tasarlanan tekrarlayan sinir ağları (TSA) 1990'larda araştırma ve geliştirmelerin önemli bir odağı olmuştur (Medsker ve Jain 2013). Standart bir YSA'da girdiler nöronlara gelmekte ve çıktılar oluşmaktadır. TSA'larda ise nöronlar zaman dizisi boyunca kendilerine de bağlanmaktadırlar. Bu konsept, nöronların bir

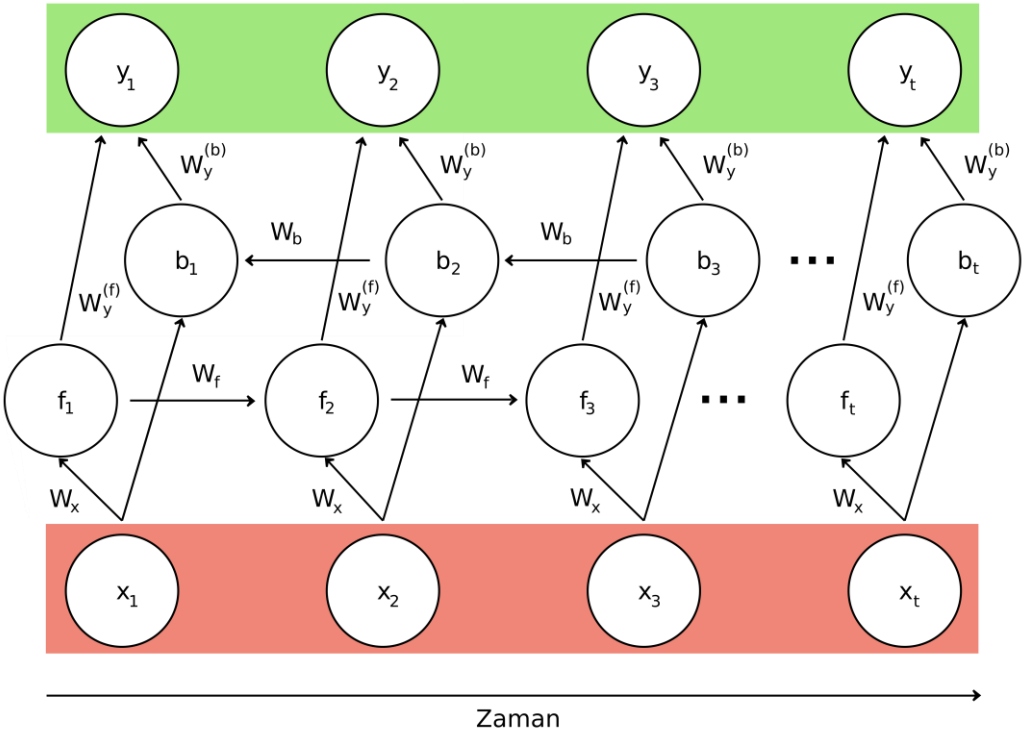
önceki adımda ilgili nöronda ne olduğunu hatırladığı kısa zamanlı bir hafızası olduğu ve bir sonraki zaman adımındaki ilgili nörona bilgi aktarabileceği şeklinde tanımlanabilmektedir. YSA'larda öğrenme, eğitim sürecinin sonunda gerçekleşmekte ve öğrenilen bilgiler hatırlanarak problem çözümünde kullanılmaktadır. TSA'lar buna ek olarak süreç boyunca da bir önceki zaman adımında ilgili nöronun bilgilerini hatırlamakta ve bu sayede genel öğrenme sürecinin başarısını arttırmaktadır.



Şekil 2.11. Tekrarlayan sinir ağı yapısı

YSA'lardaki temel amaç, verilen bir girdiden bir çıktı üretmektir. Bu işlem Şekil 2.11'deki gibi x girdi katmanı ve y çıktı katmanı olmak üzere bir dizi girdiye tekrar takrar uygulandığında ve bu tekrarlar sırasında gizli katmanlar arasında veri alışverişi olduğunda eldeki ağ TSA'ya dönüşmektedir. Bu tip sinir ağlarında girdi ve gizli durumdan bir çıktı elde etmenin yanı sıra gizli durumlar girdiye göre güncellenmektedir ve bir sonraki girdiyi işlerken bu veriler kullanılmaktadır. TSA'lar yalnızca önceki zaman adımlarından öğrenebileceği gibi sonraki zaman adımlarından da öğrenebilmektedir. Bu yapıdaki TSA'lar çift yönlü TSA olarak adlandırılmaktadır. Şekil 2.12'de de görülebileceği üzere y çıktısı önceki durum bilgilerini taşıyan f ve sonraki durum

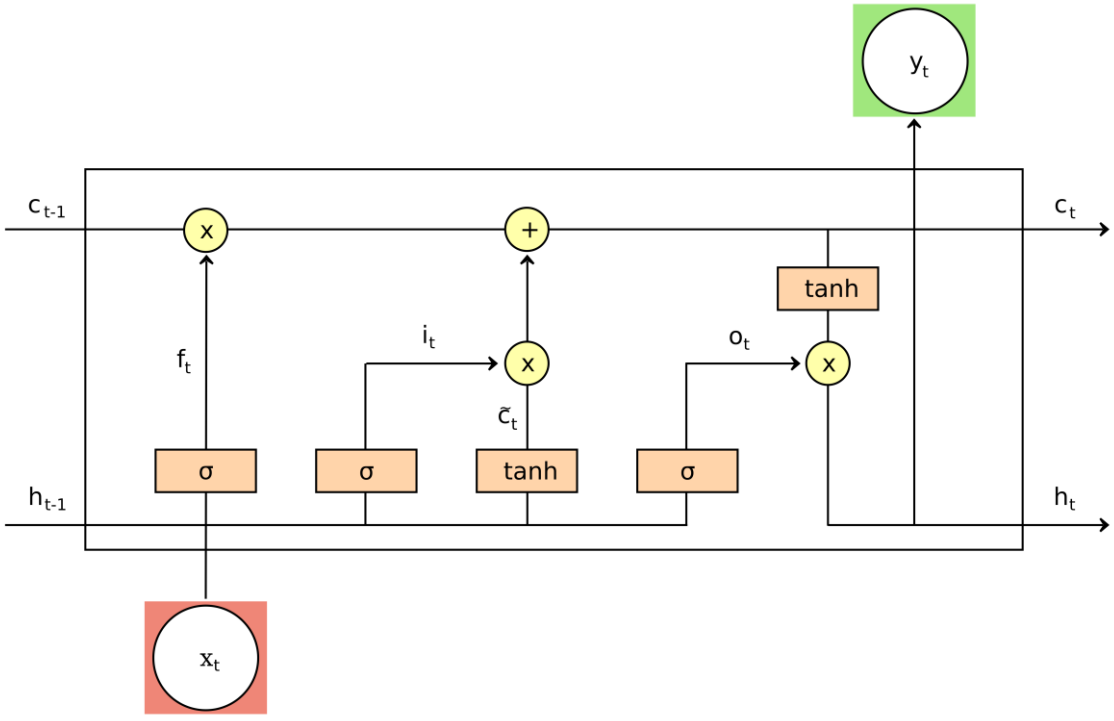
bilgilerini taşıyan b gizli katmanlarının ikisinden de bilgi alabilmektedir. Çift yönlü TSA'lar çoğunlukla el yazısı tanıma, konuşma tanıma gibi bağlam bilgisi gereken alanlarda kullanılmaktadır.



Şekil 2.12. Çift yönlü tekrarlayan sinir ağı yapısı

Örneğin konuşma tanıma gibi sabit boyutlu olmayan girdilerde bu tip sinir ağlarını kullanmak büyük avantaj sağlamaktadır. Girdi olarak alınan konuşma verisi sınıflandırılmak üzere parçalara bölüneceğinden ve her bir parçanın boyutu değişkenlik gösterebileceğinden önceki ve sonraki durumlar hakkında bilgi sahibi olmak verimi arttırmaktadır. Bu durumlar hakkında bilgi sahibi olma işlemi beraberinde hesaplama yaparken geçmiş ve gelecek verilerin ne kadarının hesaba katılacağı sorusunu getirmektedir. Doğal dil işlemede girdi olarak verilen cümlenin tamamına bakmak faydalı olabilmektedir ancak konuşma tanımada tüm girdiye bütünüyle bakmak gereksiz olacağından verinin, kullanılacak yönteme göre kelimelere, hecelere ya da seslere bölündüğü varsayılarak hesaba katılacak bölge belirlenmektedir. Aksi halde mevcut durum ve faydalanılacak bilgi arasındaki mesafe artmakta ve sinir ağı bu bilgileri eşleştiremez duruma gelmektedir.

Zamanda geriye doğru akan hata sinyallerinin süreçte patlama ya da kaybolma eğiliminde olması standart TSA'ların uzun süreli hafıza gerektiren işlemleri teoride gerçekleştirebiliyor olmasına karşın öğrenme işlemi pratikte beklenen performansı göstermemektedir. Geri yayılımda hatanın zamansal gelişimi ağırlık boyutlarına üstel olarak etki etmektedir ve bazı durumlarda eğim, süreçte kaybolacak kadar küçülerek ağırlık değişimini engellemekte hatta eğitimi durdurabilmektedir. Kaybolan gradyan problemi adı verilen bu durum, geriye yayılımın gradyanları zincir kuralıyla hesapladığı hiperbolik tanjant fonksiyonu gibi aktivasyon fonksiyonlarında ortaya çıkabilmektedir. n katmanlı bir ağda ilk katmanların ağırlıklarını hesaplamak için n adet sıfıra yakın sayının çarpılması ve gradyanın n ile üssel olarak azalarak ilk katmanların eğitiminin yavaşlaması ile sonuçlanmaktadır. Kaybolan gradyan probleminin önüne geçmek için en yaygın kullanılan yöntemlerden biri LSTM mimarisidir. Öyle ki günümüzde TSA'lar ile bütünleşmiştir denilebilir.



Şekil 2.13. Standart uzun kısa süreli hafıza yapısı

Örnek bir YSA'nın tekrarlayan katmanındaki standart LSTM yapısı Şekil 2.13'teki gibi ifade edilebilmektedir. Buna göre LSTM'in temel farkı c olarak ifade edilen hücre durumudur. LSTM hücre durumuna bilgi ekleme ve çıkarma işlemini kapılar yardımıyla

gerçekleştirmektedir. İlk aşama olarak sigmoid fonksiyonu ile hücre durumundan hangi bilgilerin atılacağı $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$ denklemine göre belirlenmektedir. Sonraki aşamada yine bir sigmoid fonksiyonu ile hangi bilgilerin güncelleneceği $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ denklemi ile belirlenerek, hiperbolik tanjant fonksiyonu ile eklenecek yeni değerlerin vektörleri $\tilde{c}_t = \tan(W_c \cdot [h_{t-1}, x_t] + b_c)$ denklemiyle oluşturularak hücre durumu $c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$ denklemine göre güncellenmektedir. Son aşama olarak ise çıktıya gönderilecek veriye karar verilmektedir. Bu aşamada son bir sigmoid fonksiyonuyla hücre durumunun hangi kısımlarının çıktıya gönderileceğine $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ denklemine göre karar verilmektedir. Ardından hücre durumu hiperbolik tanjant fonksiyonundan geçirilerek karar verilen kısımların güncellenmesi için $h_t = o_t \cdot \tan(c_t)$ denklemindeki gibi son sigmoid fonksiyon çıktısı ile çarpılarak çıktı üretilmektedir.

3. MATERYAL ve YÖNTEM

3.1. Veri Seti ve Sözlük Hazırlanması

Bir konuşma tanıma modeli oluşturmanın en önemli ve belirleyici noktalarından biri, amaca yönelik bir veri setinin oluşturulmasıdır. Yöntemler ne kadar iyi geliştirilmiş olursa olsun, yanlış veya amaç dışı oluşturulmuş bir veri seti ile eğitildiği takdirde herhangi bir anlam ifade etmeyecektir. Tez kapsamında doğru veri setinin oluşturulması amacıyla öncelikli olarak geliştirilmesi planlanan modelin hedefi göz önünde bulundurularak veri toplanacak insanların ne söylemelerinin istendiği belirlenmiştir. Seçilen kelimeler herhangi bir sektöre ait olmayan, gündelik hayatta sık kullanılan kelimelerden oluşmaktadır. Ardından veri toplama aşaması için internet platformu uygun görülerek kullanıcılardan veri toplamak amacıyla hazırlanan internet sitesi vasıtasıyla konuşmalar kaydedilerek etiketlenmiştir. Eldeki verinin uygun şekilde bölünmesinin ardından oluşturulan sözlükte kullanılan kelimeler temel alınarak geleneksel yöntemde kullanılmak üzere bir fonem sözlüğü oluşturulmuş ve ortaya çıkan veri seti eğitim ve test amacıyla kullanılmıştır.

Oluşturulan sözlük, yöntemleri test edebilmek ve tez sürecinde yeterli sayıda veriye ulaşmak amacıyla Türkçe dilinde sık kullanılan kelimelerden oluşmaktadır. Veri seti ise Çizelge 3.1'de görülebilecek yüz farklı kelimenin bulunduğu sözlükteki kelimelerin karışık olarak üçerli bir şekilde okunmasından oluşan ses dosyalarını ve buna karşılık gelen konuşmacı adı ve konuşma metinlerini içermektedir. Konuşmacı bilgisi Derin Konuşma modelinde bir önem arz etmese de geleneksel modelde göz önünde bulundurulmaktadır. Eğitim için kullanılacak kelimeler seçilirken sözlükte "yedi" ve "yeni" gibi fonetik olarak birbirine benzeyen sözcükler ve "hala" gibi farklı konuşmacılar tarafından farklı telaffuz edilebilecek kelimeler de barındırılmasına dikkat edilmiştir. Bu sayede eğitilen modelin telaffuzu birbirine yakın kelimelerdeki ve aynı etiketli kelimeye karşılık gelen farklı telaffuzlara göre eğitilmesinin sonucundaki başarısını ayrı ayrı ölçmenin mümkün olması hedeflenmiştir.

Çizelge 3.1. Oluşturulan sözlükteki kelimeler ve tek konuşmacılı modellerin eğitiminde kullanılma sayıları

adam (23)	altı (34)	amca (24)	anne (32)	artık (31)
asla (27)	az (37)	baba (29)	bay (36)	bayan (25)
benim (31)	bey (27)	beş (31)	bir (17)	bunun (20)
burada (25)	bütün (29)	büyük (33)	dakika (26)	dayı (32)
devam (33)	dokuz (27)	dostum (29)	dört (27)	efendim (29)
erkek (24)	eski (30)	evet (29)	eğer (33)	fazla (26)
gel (24)	geldi (20)	geliyor (23)	gerçekten (34)	gidelim (26)
git (36)	gitti (31)	güzel (29)	hakkında (30)	hala (26)
hanım (27)	harika (25)	hayır (25)	herkes (20)	hiç (35)
hiçbir (25)	imdat (33)	istiyorum (38)	iyi (28)	içinde (32)
işte (26)	iki (22)	kadın (25)	kardeş (30)	kim (37)
kimin (25)	kötü (27)	küçük (26)	lanet (23)	lütfen (26)
merhaba (34)	naber (22)	nasıl (20)	nasılsın (34)	neden (21)
nerede (30)	olsun (27)	on (26)	onun (29)	orada (27)
para (24)	pekala (24)	saat (44)	sadece (25)	saniye (39)
sanırım (24)	sekiz (32)	selam (22)	senin (24)	sonra (22)
sıfır (34)	tamam (26)	teşekkür (22)	var (34)	yardım (26)
yedi (24)	yeni (26)	yeğen (26)	yok (31)	yoksa (23)
zaman (17)	çirkin (28)	çocuk (35)	çünkü (28)	önce (27)
önemli (25)	özür (35)	üç (35)	şey (29)	şimdi (34)

Çizelge 3.2. Oluşturulan sözlükteki kelimeler ve tek konuşmacılı modellerin testinde kullanılma sayıları

adam (8)	altı (6)	amca (5)	anne (8)	artık (12)
asla (8)	az (8)	baba (8)	bay (8)	bayan (8)
benim (3)	bey (7)	beş (5)	bir (3)	bunun (7)
burada (12)	bütün (3)	büyük (8)	dakika (7)	dayı (4)
devam (8)	dokuz (7)	dostum (8)	dört (7)	efendim (5)
erkek (8)	eski (9)	evet (6)	eğer (8)	fazla (4)
gel (4)	geldi (4)	geliyor (5)	gerçekten (4)	gidelim (6)
git (4)	gitti (7)	güzel (12)	hakkında (4)	hala (4)
hanım (8)	harika (11)	hayır (9)	herkes (7)	hiç (6)
hiçbir (5)	imdat (7)	istiyorum (7)	iyi (11)	içinde (6)
işte (9)	iki (8)	kadın (6)	kardeş (7)	kim (5)
kimin (6)	kötü (8)	küçük (8)	lanet (6)	lütfen (10)
merhaba (11)	naber (5)	nasıl (5)	nasılsın (6)	neden (6)
nerede (7)	olsun (7)	on (5)	onun (5)	orada (6)
para (8)	pekala (7)	saat (7)	sadece (8)	saniye (5)
sanırım (5)	sekiz (12)	selam (6)	senin (7)	sonra (10)
sıfır (7)	tamam (5)	teşekkür (4)	var (6)	yardım (6)
yedi (9)	yeni (9)	yeğen (8)	yok (10)	yoksa (8)
zaman (6)	çirkin (9)	çocuk (6)	çünkü (9)	önce (5)
önemli (10)	özür (8)	üç (8)	şey (4)	şimdi (3)

Çizelge 3.3. Oluşturulan sözlükteki kelimeler ve birden çok konuşmacılı modellerin eğitiminde kullanılma sayıları

adam (183)	altı (202)	amca (190)	anne (190)	artık (188)
asla (201)	az (207	baba (175)	bay (197)	bayan (181)
benim (209)	bey (207)	beş (211)	bir (167)	bunun (172)
burada (195)	bütün (198)	büyük (212)	dakika (188)	dayı (199)
devam (202)	dokuz (202)	dostum (203)	dört (216)	efendim (201)
erkek (182)	eski (203)	evet (203)	eğer (203)	fazla (194)
gel (187)	geldi (204)	geliyor (181)	gerçekten (193)	gidelim (199)
git (198)	gitti (206)	güzel (199)	hakkında (184)	hala (179)
hanım (202)	harika (183)	hayır (186)	herkes (185)	hiç (189)
hiçbir (196)	imdat (213)	istiyorum (197)	iyi (208)	içinde (197)
işte (197)	iki (215)	kadın (173)	kardeş (179)	kim (196)
kimin (179)	kötü (194)	küçük (178)	lanet (197)	lütfen (207)
merhaba (184)	naber (174)	nasıl (191)	nasılsın (201)	neden (204)
nerede (201)	olsun (183)	on (207)	onun (184)	orada (185)
para (194)	pekala (179)	saat (224)	sadece (195)	saniye (195)
sanırım (188)	sekiz (174)	selam (202)	senin (170)	sonra (181)
sıfır (203)	tamam (192)	teşekkür (181)	var (212)	yardım (203)
yedi (197)	yeni (189)	yeğen (204)	yok (182)	yoksa (177)
zaman (203)	çirkin (190)	çocuk (203)	çünkü (187)	önce (210)
önemli (216)	özür (196)	üç (204)	şey (200)	şimdi (196)

Çizelge 3.4. Oluşturulan sözlükteki kelimeler ve birden çok konuşmacılı modellerin testinde kullanılma sayıları

adam (57)	altı (55)	amca (43)	anne (45)	artık (52)
asla (48)	az (54)	baba (58)	bay (44)	bayan (45)
benim (44)	bey (43)	beş (50)	bir (42)	bunun (44)
burada (49)	bütün (48)	büyük (47)	dakika (44)	dayı (41)
devam (55)	dokuz (42)	dostum (56)	dört (45)	efendim (42)
erkek (54)	eski (59)	evet (40)	eğer (56)	fazla (54)
gel (54)	geldi (56)	geliyor (37)	gerçekten (42)	gidelim (46)
git (48)	gitti (48)	güzel (56)	hakkında (46)	hala (53)
hanım (57)	harika (57)	hayır (52)	herkes (48)	hiç (60)
hiçbir (48)	imdat (45)	istiyorum (36)	iyi (45)	içinde (50)
işte (45)	iki (48)	kadın (49)	kardeş (56)	kim (62)
kimin (54)	kötü (61)	küçük (39)	lanet (51)	lütfen (46)
merhaba (58)	naber (68)	nasıl (37)	nasılsın (53)	neden (47)
nerede (48)	olsun (43)	on (34)	onun (37)	orada (53)
para (65)	pekala (52)	saat (45)	sadece (42)	saniye (43)
sanırım (53)	sekiz (59)	selam (47)	senin (44)	sonra (57)
sıfır (52)	tamam (51)	teşekkür (42)	var (42)	yardım (42)
yedi (41)	yeni (46)	yeğen (42)	yok (46)	yoksa (52)
zaman (38)	çirkin (50)	çocuk (38)	çünkü (58)	önce (58)
önemli (45)	özür (46)	üç (46)	şey (52)	şimdi (38)

Şekil 3.1. Veri toplamak için hazırlanan internet sitesinin kullanıcı arayüzü

Veri setinin istenilen formatta ve içerikte oluşturulabilmesi için farklı insanlardan veri toplamak amacıyla üyelik gerektiren bir internet sitesi hazırlanmıştır. Şekil 3.1'de arayüzü görülebilecek internet sitesi, doğru veriler toplama amacıyla kapalı kullanıma açılarak kullanıcılardan istenilen formatta veriler toplanmıştır. İnternet sitesi, oluşturulan kelime listesinden rastgele üç farklı kelimeyi kullanıcıya sunarak bu kelimeleri kaydetmesini sağlamaktadır. Konuşma kayıtlarının üçerli bir şekilde kullanıcıdan alınmasının nedeni, hem kelimelerin tek tek okunması yerine birlikte okunarak daha akıcı söylemlerden oluşan bir veri seti oluşturmak hem de kullanıcılardan alınan veri sayısını arttırmaktır. Ardından kullanıcı kaydını dinleyerek, yanlış veya bozuk kayıt durumunda ilgili kaydın yanındaki "-" butonu ile kaydı yüklenecekler listesinden çıkarabilmektedir. Her beş kayıtta kullanıcıdan yükle butonuna basarak kayıtları sisteme yüklemesi beklenmektedir. Yüklemenin ardından kullanıcı kayıt işlemine devam etmektedir. Sisteme yüklenen kayıtlar, ilgili kullanıcının kayıt klasöründe, kullanıcı adı ve kayıt tarihine göre oluşturulan benzersiz dosya adları ile tutulmaktadır. Ayrıca etiketli veri oluşturmak amacıyla tüm kayıtları barından bir Excel tablosunda kayıt adları ve ilgili kayıtta söylenen konuşma içeriği tutulmaktadır. Her bir yükleme işleminin ardından bu tablo güncellenmektedir. Veri setinin olası farklı çalışmalarda da kullanılabilmesi adına üyelik sırasında kullanıcıdan cinsiyet, uyruk ve doğum tarihi bilgileri de alınarak veri tabanında saklanmaktadır. Oluşturulan veri setinde kayıtları bulunan konuşmacıların

bilgileri Çizelge 3.5'de gösterilmektedir. Elde edilen konuşma verileri, model girdilerini standart bir hale dönüştürmek amacıyla tek kanallı (mono channel) 16 kHz ve 16 bit formatında olacak şekilde yeniden düzenlenmiştir. Veri seti Derin Konuşma modeli için hazırlanırken tüm dosyalar tek bir klasörde toplanmış ve veri setini ayırmak amacıyla ilgili dosya isimlerinden ve bu ses dosyalarına karşılık gelen etiketlerden oluşan eğitim, test ve doğrulama için üç adet Excel tablosu oluşturulmuştur. Veri seti eğitim ve test için bölünürken verinin %80'i eğitim için %20'si ise eğitilen modelin test edilmesi için kullanılmaktadır. Eğitim için ayrılan kısmın yine %80'i modeli eğitmek %20'si ise her iterasyondan sonra eğitilen modelleri sınayıp hangi model üzerinden devam edileceğini saptamak amacıyla doğrulama için kullanılmaktadır. Sonuç olarak elimizdeki veri seti %64 eğitim, %16 doğrulama ve %20 test olacak şekilde üçe ayrılmıştır. Geleneksel model için veri seti oluşturulurken ise eldeki verinin %80 eğitim için, %20'si ise test için ayrılmıştır. Ayrıca varsayılan Kaldi yapısı gereği dosya isimlerini ve etiketlerini tutmak için hazırlanan tablolarda ses verilerinin hangi konuşmacılara ait olduğu da belirtilmiştir.

Çizelge 3.5. Tek konuşmacı ve çok konuşmacılı modellerin eğitimi ve testi için kullanılan veri setindeki konuşmacılar ve özellikleri

	Tek Konuşmacı	**Birden Çok Konuşmacı**								
Konuşmacı	a	a	b	c	d	e	f	g	h	i
Cinsiyet	e	e	k	e	k	e	e	e	e	e
Yaş	28	28	28	25	52	25	25	54	27	29
Kelime	1165	1529	1170	990	999	231	692	441	997	1051

3.2. Derin Konuşma Tanıma Modeli

Geleneksel konuşma tanıma sistemleri, adımlara bölünerek boru hattı şeklinde gerçekleştirilen algoritmalardan ve elle tasarlanmış ön işleme aşamalarından oluşan karmaşık düzenlere dayanmaktadır. Derin Konuşma yöntemi, derin öğrenmenin bu işleme aşamalarının yerini aldığı, uçtan uca bir konuşma sistemi olarak tanımlanabilmektedir. Uçtan uca yaklaşımıyla tasarlanan konuşma modelleri konuşma tanıma görevlerinde yeterli veri ile eğitildiğinde yüksek performans sağlarken aynı zamanda kullanıcıya yönelik işlem yükü bakımından çok daha basittir. Bu tip sistemler

doğrudan verilerden öğrendiğinden, konuşmacı adaptasyonu veya gürültü filtreleme için özel bileşenlere ihtiyaç duymamaktadır. Aksine farklı konuşmacılar ve gürültülü kayıtlara karşı geleneksel yöntemlere kıyasla üstünlük sağlayabilmektedir.

Geleneksel konuşma sistemleri, özelleştirilmiş girdiler, akustik modeller ve HMM'ler dahil olmak üzere çok sayıda büyük ölçüde tasarlanmış işleme aşaması kullanmaktadır. Bu ardışık düzenleri iyileştirmek için, yazılım geliştiricilerinin özellikleri ve modeli optimize etmek için büyük çaba sarf etmesi gerekmektedir. Konuşma tanıma süreci derin öğrenme algoritmalarıyla birlikte, genellikle akustik modelleri geliştirerek, tasarlanan sistemlerin performansını büyük ölçüde iyileştirmiştir. Bu gelişme önemli olsa da, derin öğrenme geleneksel konuşma tanıma sistemlerinde hala sınırlı bir rol oynamaktadır. Örneğin kullanılan sisteme derin öğrenme algoritmaları dahil edilmiş olsa bile gürültülü konuşma tanıma gibi durumlarda çalışabilen bir sistem oluşturmak için, sistemin geri kalanında zahmetli ön işlemler kullanılması gerekmektedir. Buna karşılık, temel alınan Hannun ve ark. (2014) tarafından geliştirilen Derin Konuşma, TSA kullanarak derin öğrenmeyi uçtan uca uygulamaktadır. Model, istenilen çıktıları üretmek için uçtan uca eğitildiğinde, yeterli veri ve hesaplama gücüyle, gürültülü konuşmaları ayırt etmeyi veya farklı konuşmacılara göre değişebilen konuşma türlerini kendi başına öğrenebilir duruma gelmektedir.

Bu tarz sistem tasarımları, farklı zorlukları da beraberinde getirmektedir. Öncelikli olarak uçtan uca bir derin öğrenme yapısı kurulduğundan ağı eğitmek için çok fazla sayıda etiketli veri gerekmektedir. Bu durum da doğal olarak veri seti oluşturmanın zorluğunu ve eğitim için ihtiyaç duyulan hesaplama gücünü büyük ölçüde arttırmaktadır. Bu zorlukların önüne geçebilmek adına sistemin gürültülü konuşma tanımadaki başarısı göz önünde bulundurularak araştırmacılar tarafından konuşma verilerine gürültü eklenerek veri seti boyutunu arttırmak ve eğitim işlemlerini grafik işlemcileri üzerinden paralel olarak gerçekleştirilerek eğitim sürecini hızlandırmak gibi çözümler geliştirmişlerdir. Yöntemin başarısı sonucunda Mozilla, geliştirilen yöntemi Google'nin tensorflow kütüphanesinden faydalanarak açık kaynaklı bir araç olarak geliştiricilere sunmuştur.

3.3. Derin Konuşma Modelinde Tekrarlayan Sinir Ağı Yapısı

Tasarlanan sistemin özü, konuşma spektrogramlarını analiz etmek ve buna karşılık gelen Türkçe metin çözümlemelerini oluşturmak için eğitilmiş bir TSA'dan oluşmaktadır. Tek bir konuşma verisinde konuşma sinyalinin x, buna karşılık gelen etiketin y olduğu varsayılırsa konuşma veri seti $X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(3)}), \dots\}$ şeklinde ifade edilmektedir. İfadedeki her bir $x^{(i)}$ ifadesi, zaman dilimlerinin ses özelliklerinin bir vektörü $x_t^{(i)}, t = 1, \dots, T^{(i)}$ olduğu, $T^{(i)}$ uzunluğundaki bir zaman serisidir. Spektrogramlar ayırt edici özellik olarak kullanılacağından $x_{t,p}^{(i)}$, t zamanında ses çerçevesindeki p'inci frekans bölmesinin gücünü göstermektedir. Geliştirilen TSA'nın asıl amacı, verilerden elde edilen bir x girdi dizisini, $c_t \in \{a, b, c, \dots, boşluk, tırnak\ işareti, null\}$ ve $\hat{y}_t = \mathbb{P}(c_t, x)$ koşullarında oluşturulacak y için bir karakter olasılıkları dizisine dönüştürmektir.

Oluşturulan TSA modelinin yapısı bir $h^{(0)}$ girdi katmanı, 5 adet gizli katman ve bir çıktı katmanından meydana gelmektedir. Her bir L katmanındaki gizli birimler $h^{(L)}$ ile, girdi sinyali ise x ile temsil edilmektedir. İlk üç gizli katman tekrarlayan yapıda değildir. İlk katman için her t anında çıktı, her iki taraftaki dokuz adet K çerçevenin bağlamı ile birlikte x_t spektrogram çerçevesine bağlıdır. Diğer tekrarlamayan katmanlar her t anında bağımsız veriler ile işlem yapmaktadır. Sonuç olarak ilk üç katman Denklem 3.1'deki gibi hesaplanabilmektedir.

$$h_t^{(L)} = g(W^{(L)} h_t^{(L-1)} + b^{(L)}) \tag{3.1}$$

Denklem 3.1'de $g(z) = \min\{\max\{0, z\}, 20\}$ kırpılmış ReLu (clipped ReLu) aktivasyon fonksiyonunu, $W^{(L)}$ ağırlık matrisi ve $b^{(L)}$ sapmayı temsil etmektedir. Kırpılmış ReLu kullanımının amacı, tekrarlayan katmandaki aktivasyonların aşırı yüklenerek hata vermesinin önüne geçmektir. Dördüncü katman çift taraflı tekrarlayan bir katmandır. Bu katman ileri doğru tekrarlayan $h^{(f)}$ ve geriye doğru tekrarlayan $h^{(b)}$ olmak üzere iki takım gizli birimden oluşmaktadır. $h^{(f)}$, $t = 0$ anından $t = T^{(i)}$ anına kadar Denklem 3.2'deki gibi sıralı olarak i'inci ifadeyi hesaplarken, $h^{(b)}$ ise $t = T^{(i)}$ anından $t = 0$ anına kadar Denklem 3.3'teki gibi sıralı olarak tersten hesaplama yapmaktadır.

$$h_t^{(f)} = g(W^{(4)}h_t^{(3)} + W_t^{(f)}h_{t-1}^{(f)} + b^{(4)}) \quad (3.2)$$

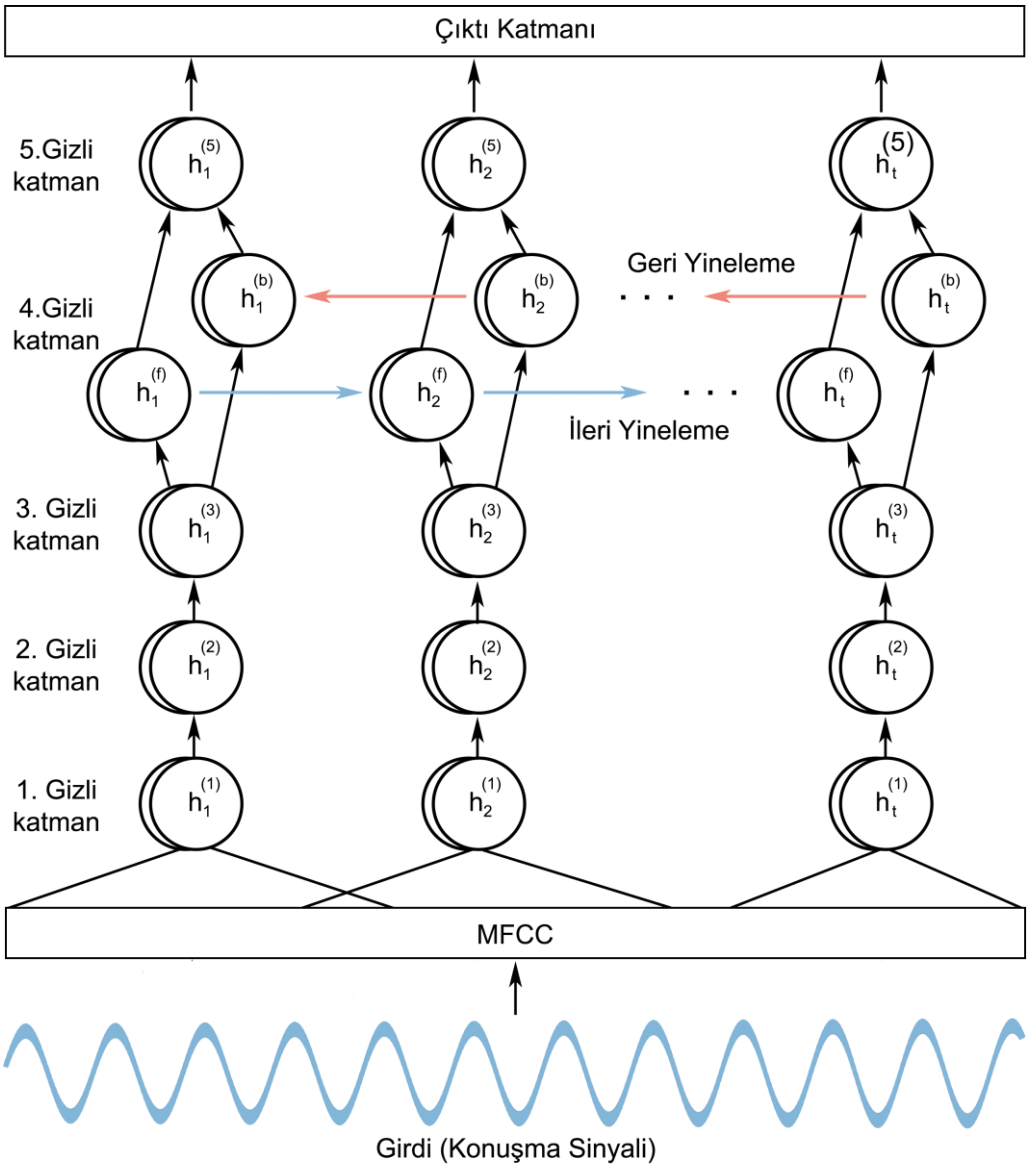
$$h_t^{(b)} = g(W^{(4)}h_t^{(3)} + W_t^{(b)}h_{t+1}^{(b)} + b^{(4)}) \quad (3.3)$$

Tekrarlamayan bir katman olan beşinci katman ise hem ileri doğru hesaplayan hem de geriye doğru hesaplayan birimleri girdi olarak almakta ve $h^{(6)}$ olarak ifade edilen çıktı katmanına bağlanmaktadır. Çıktı katmanı alfabedeki her karakter k ve zaman dilimi t için tahmin edilen karakter olasılıklarını veren standart Softmax fonksiyonudur.

$$h_{t,k}^{(6)} = \hat{y}_{t,k} \equiv \mathbb{P}(c_t = k|x) = \frac{\exp(W_k^{(6)}h_t^{(5)} + b_k^{(6)})}{\sum_j \exp(W_j^{(6)}h_t^{(5)} + b_j^{(6)})} \quad (3.4)$$

$\mathbb{P}(c_t|x)$ için tahmin hesaplandıktan sonra, tahmindeki maliyeti ölçmek için Bağlantısal Zamansal Sınıflandırma (CTC) kaybı $L(\hat{y}, y)$ hesaplanmaktadır. Eğitim aşamasında referans y karakter dizisinin ağ çıktıları doğrultusunda $\nabla_{\hat{y}} L(\hat{y}, y)$ gradyanı hesaplanabilmektedir. Bu noktadan itibaren ağ boyunca geriye yayılım ile tüm model parametleri göz önünde bulundurularak gradyan hesaplamak mümkün hale gelmektedir. Geriye yayılım aşamasında Nesterov Hızlandırılmış Gradyan (NAG) kullanılmaktadır.

TSA modelinin tamamlanmış hali Şekil 3.2'de gösterilmektedir. Yalnızca tek bir tekrarlayan katman kullanılan yapı oldukça basittir. Modelin ileri yineleme ve geri yineleme aşamalarında LSTM kullanılmıştır. Eğitim sırasında yalnızca ileri besleme katmanlarında %5 oranında bir bırakma uygulanmaktadır. Bırakma ile rastgele seçilen nöronlar göz ardı edileceğinden aşırı uyum göstermenin önüne geçilmektedir. Tasarlanan sistem karakter tahmini temelli ilerlediğinden test aşamasındaki hataların çoğu harf yutma, hece yutma gibi durumlardan meydana gelmektedir. Bu olasılığı azaltarak başarı oranını arttırabilmek adına kullanılan kelime külliyatı test aşamasında sisteme verilerek tahmin edilen kelime bir kelime eşleştirme algoritmasından geçirilerek bu tarz küçük hataların önüne geçilmesi sağlanmıştır.

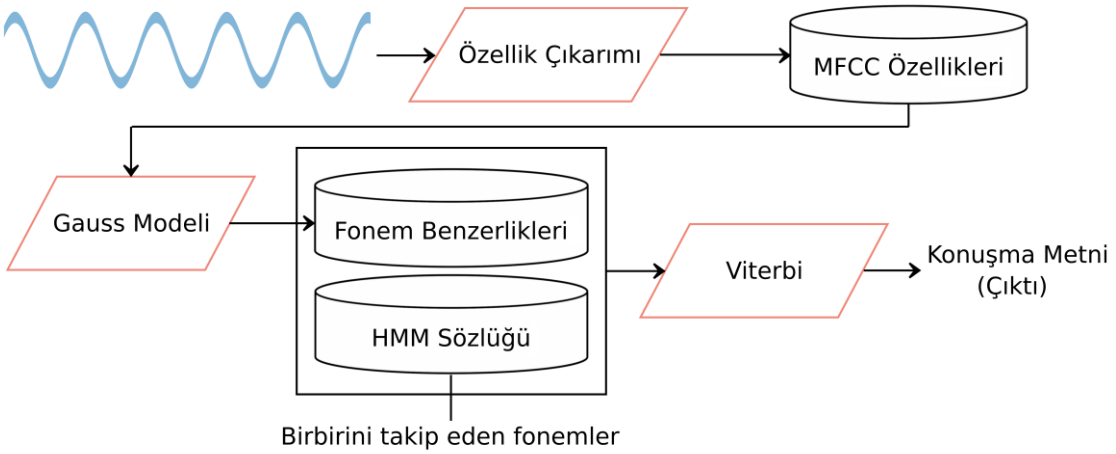


Şekil 3.2. Derin Konuşma TSA şeması

3.4. GMM-HMM Tabanlı Geleneksel Konuşma Tanıma Modeli

Geleneksel model oluşturulurken faydalanılan Kaldi yazılımı, Povey ve ark. (2011) tarafından geliştirilen, konuşma tanıma ve sinyal işleme görevlerini gerçekleştirmek için kullanılan açık kaynaklı bir yazılımdır. Açık kaynaklı olması, algoritmalarının kolay düzenlenebilir olması, lineer cebir işlemleri için Temel Lineer Cebir Alt Programları (BLAS) ve standart Lineer Cebir Paketini (LAPACK) içeren kapsamlı bir matris kütüphanesine sahip olması ve Sonlu Durum Dönüştürücüler (FST) ile etkileşiminin efektif olması bu yazılımı projeye göre özelleşmiş sistemler tasarlamaya elverişli hale getirmektedir. Kaldi yazılımı ile oluşturulan geleneksel model, derin öğrenmenin

konuşma tanıma alanında sağladığı kolaylıklardan önce sık kullanılan MFCC özellik çıkarımı, HMM ve Gauss Karışım Modellerini (GMM) temel alan bir modeldir ve akış şeması Şekil 3.3'te görülebilmektedir.



Şekil 3.3. GMM-HMM temelli geleneksel konuşma tanıma yöntemi akış şeması

3.4.1. Özellik çıkarımı

Kaldi yazılımında özellik çıkarımı ve dalga okuma algoritması standart Mel Frekansı Sepstral Katsayıları (MFCC) ve Algısal Doğrusal Tahmin (PLP) özelliklerini belirlemeye dayanmaktadır. Özellik çıkarımı yöntemi olarak ise yaygın kullanıma sahip olan Ses Yolu Uzunluğu Normalizasyonu (VTLN), Sepstral Ortalama ve Varyans Normalizasyonu (CMVN), Doğrusal Ayırıcı Analizi (LDA), Küresel Yarı-bağlı Kovaryans/Maksimum Olasılık Doğrusal Dönüşüm (STC/MLLT) yöntemlerini kullanmaktadır. MFCC özelliklerini belirlemek adına konuşma girdisi 25 ms'lik pencerelere bölünerek her aşamada pencereler 10 ms kaydırılmaktadır.

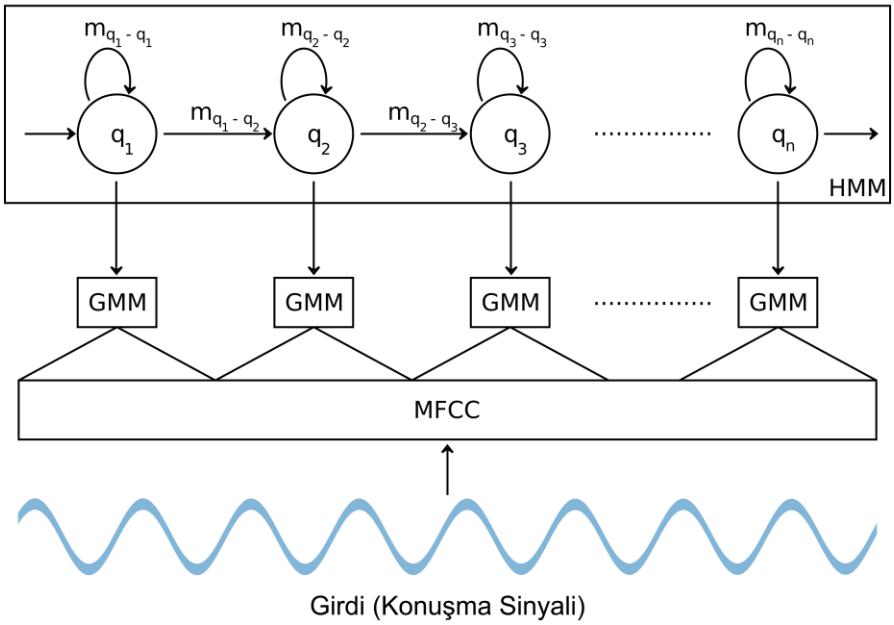
3.4.2. Akustik modelleme

Konuşma tanımada akustik modelleme, fonemler ile konuşma sinyallerinin arasındaki ilişkilendirmeyi temsil etmektedir. Geleneksel konuşma modelinde akustik modeller HMM ve GMM'lerin birlikte kullanılmasıyla oluşturulmaktadır. Konuşma tanımadaki temel amaç bir W kelime dizisini X özellik vektörleri şeklinde ifade edebilmektir.

$$W' = argmax\ P(X|W)P(W) \tag{3.5}$$

Tahmin edilen kelime dizisi W', akustik model $P(X|W)$ ve $P(W)$ dil modeline göre olasılıkların çarpımı şeklinde Denklem 3.5'teki gibi ifade edilebilmektedir. Geleneksel modelde oluşturulan akustik model özellik vektörleri dizisini kelimeler yerine veri setindeki kelimelere göre oluşturulan fonem sözlüğünü temel alarak modellemektedir. Konuşma setindeki kayıtlar kelimelerin rastgele bir şekilde üçerli okunmasından oluştuğu için oluşturulan modelde kelimelerin birbirinden sonra gelme olasılıklarını belirleyen dil modelinin belirleyici bir etkisi bulunmamaktadır.

GMM-HMM temelli akustik model Şekil 3.4'teki gibi temsil edilebilmektedir. Konuşma bölümünün akustik özelliklerinin olasılık dağılımının belirlenmesinde GMM; konuşma sinyallerinin zaman sırasına göre olasılığını (hangi fonemlerin birbirini takip edebileceğini) belirlemede ise HMM kullanılmaktadır.



Şekil 3.4. HMM-GMM temelli akustik model

Kaldi yazılımında anlamlı kelimeleri oluşturacak fonemlerin seçiminde karar ağaçlarından faydalanılmaktadır. Karar ağaçlarının oluşturulmasında Young ve ark. (1994) tarafından geliştirilen durum bağlama yöntemi temel alınmıştır. Kullanılan yöntemde büyük eğitim setlerini idare edebilmek için, ağaç oluşturma yalnızca HMM durumları içinde kodlanan istatistiklere dayanmaktadır ve orijinal verilere doğrudan bir referans yapılmamaktadır.

3.4.3. Çizge oluşturulması

Tüm eğitim ve kod çözme algoritmaları Ağırlıklı Sonlu Durum Dönüştürücüleri (WFST) kullanmaktadır. Standart WFST yapısında, kod çözme grafiğindeki girdiler içeriğe bağlı durumlara karşılık gelmektedir. Kaldi sisteminde bu girdiler sayısaldır ve bu girdilere oluşturulan pdf (olasılık dağılım fonksiyonu) numaraları atanmaktadır. Fakat bu pdf numaraları oluşturulurken aynı pdf numarasının birden fazla sesi temsil etmesi mümkün olduğundan standart yaklaşım FST belirlemede, Viterbi yolundan ses dizilerinin analizi ve geçiş olasılıklarının eğitilmesi sırasında bir takım problemlere yol açabilmektedir. Kaldi'de bu sorunun çözümü için FST girdileri, pdf numarası, bu numaraya karşılık gelen ses ve bu ses için topoloji spesifikasyonu içindeki geçiş kullanılarak tanımlanan yeni geçiş numaralarından oluşmaktadır. Sonuç olarak model, bire bir eşleşmiş geçiş kimlikleri ve geçiş olasılığı parametrelerinden oluşmaktadır. Kod çözümleme çizgeleri, Mohri ve ark.'nın (2002) tasarımı temel alınarak tasarlanmıştır. Tasarlama sürecinde FST'nin belirsiz kaldığından emin olmak için uygulanan ağırlık bastırımı (weight pushing) algoritması, belirsiz olmayan (istatistiksel) dil modellerinde probleme sebebiyet verebileceği için grafiğin oluşturulmasının her adımında stokastik olduğundan emin olmak şartıyla yeni adıma geçerek göz ardı edilmiştir.

3.5. Google Speech API

Google Speech API, Google tarafından YSA modelleri kullanılarak geliştirilen, her birinin başarı oranları değişiklik göstermekle birlikte seksenden fazla dilde kullanıcıların konuşma verilerini yazıya dönüştürmesini sağlayan bir uygulama programlama arayüzüdür. Kapalı kaynaklı ve ticari amaçlı olması sebebiyle kullandığı yöntemin detayları kullanıcılarla paylaşılmamaktadır. Derin Konuşma ve geleneksel modeli test etmek için kullanılan veri setleri, Google Speech API ile günümüzde aktif kullanılan ve konuşma tanımada en başarılı modellerden biri olan Google konuşma modeli kullanılarak test edilmiş ve modellerin başarısı kıyaslanarak kullanılabilirliği tespit edilmeye çalışılmıştır. Google Speech API kullanılarak gerçekleştirilen testlerde, verilerin Türkçe dilinde olduğu belirtilerek API tarafından sağlanan konuşma uyarlaması özelliği ile veri setindeki tüm kelimeler belirtilerek konuşma tanıma özelleştirilmiş ve bu kelimelerin yazıya dönüştürülme doğruluğu arttırılmıştır.

4. BULGULAR ve TARTIŞMA

Derin konuşma modelinin eğitimi ve testinde Mozilla'nın python dilinde Google Tensorflow kütüphanesini kullanarak geliştirdiği açık kaynaklı konuşma tanıma motoru temel alınmıştır. Geleneksel model de yine Kaldi kitinin geliştiricileri tarafından sunulan, c++ ve kabuk programlarından (shell script) oluşan açık kaynaklı yazılım kullanılarak adapte edilmiştir. Hazırlanan veriyi Google API ile test etmek için ise python dilinde bir test programı yazılmıştır. İşlemler, Ubuntu 16.04 işletim sistemine ve Nvidia 950M ekran kartına sahip bir bilgisayarda gerçekleştirilmiştir. Sistem kalitesinin önemi Derin Konuşma yönteminin eğitiminde ve testinde ortaya çıkmaktadır. Derin öğrenme işlemleri CPU'ya kıyasla daha hızlı olacağı için Nvidia tarafından geliştirilen CUDA sayesinde GPU üzerinden gerçekleştirilmiştir ancak kullanılan ekran kartı ile 3 iterasyondan (epoch) oluşan Derin Konuşma modelinin eğitimi tek konuşmacıda 18 saat; birden çok konuşmacıda ise 4 gün sürmüştür.

Eldeki veri setinin modellere tek konuşmacı, farklı konuşmacılar ve veri sayısı değişkenleri şeklinde farklı uygulamaları ile modellerin konuşmacı bağımlılığı ve eğitimde kullanılan veri miktarın modeller üzerine etkisi incelenmiştir. Hata oranları Kelime Hata Oranı (WER) ve Cümle Hata Oranı (SER) olarak adlandırılan ve konuşma tanımada en yaygın kullanılan başarı ölçütleri ile hesaplanmaktadır. WER, bir kayıttaki kelimeleri tespit etmedeki doğruluk oranını gösterirken SER, cümlenin bütünün doğru tespit edilip edilemediğini göstermektedir.

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{4.1}$$

WER, Denklem 4.1'de görüldüğü şekilde hesaplanabilmektedir. Bu denklemde S harf hatası bulunan veya komple yanlış tahmin edilen kelime sayısını, D tespit edilemeyerek cümleden silinen kelimelerin sayısını, I cümleye eklenen fazladan kelime sayısını ve N ilgili konuşmada bulunan toplam kelime sayısını temsil etmektedir. Toplam kelime sayısı aynı zamanda yanlış tahmin edilen kelimeler, silinen kelimeler ve doğru tahmin edilen kelimelerin toplamı şeklinde de ifade edilebilmektedir. SER ise WER'e nazaran hesaplanması çok daha kolaydır. Eğer tahmin edilen cümle içinde eksik veya fazla kelime ya da yanlış tahmin edilen bir kelime yoksa cümle doğru kabul edilmekte ve yanlış cümlelerin tüm cümlelere oranı hesaplanarak sonuç elde edilmektedir.

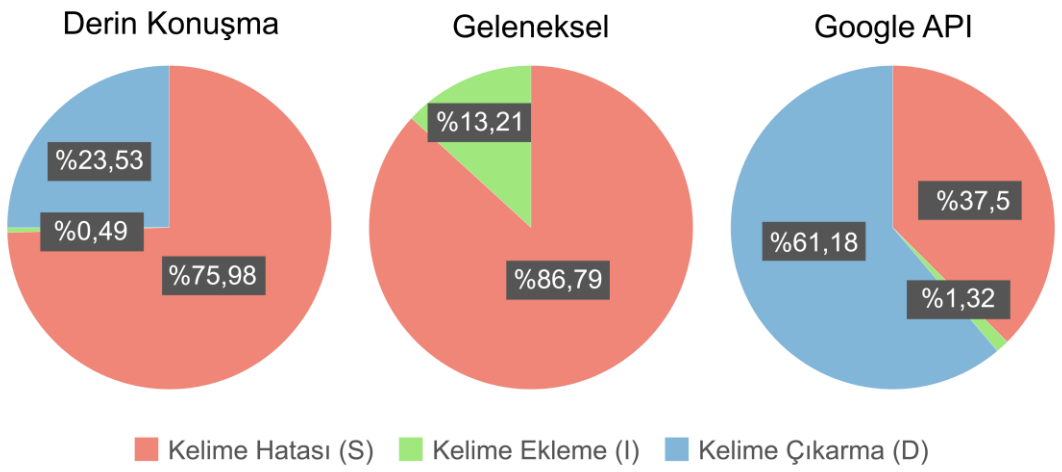
WER ve SER hesaplama tekniklerinden faydalanılarak ilk olarak tek konuşmacı için 935'i eğitim 230'u test için kullanılan ve toplam 1165 kayıttan oluşan veri seti ile yapılan testlerin Derin Konuşma, geleneksel model ve Google API başarı oranları Çizelge 4.1'de görülebilmektedir. Tek konuşmacılı modelde Derin Konuşma modeli, YSA eğitiminde çok fazla sayıda veriye ihtiyaç duyulduğundan hata oranları diğer yöntemlere nazaran daha yüksek çıkmıştır. Buna karşın eğitim setinin boyutu geleneksel model için yeterli olmuştur ve daha düşük hata oranları elde edilmiştir. Geleneksel modelin başarı oranının yüksek olmasının temel sebeplerinden biri kelime külliyatının kısıtlı olmasıdır. Modelin eğitimi ve testi için kullanılacak fonem sözlüğü de göz önünde bulundurulduğunda istatistiksel eşleştirmeler, eğitim için kullanılan düşük sayıdaki konuşma sinyali ile yüksek başarı sağlamıştır. Derin konuşma modelinde ise modele girdi olarak fonem sözlüğü yerine yalnızca Türk alfabesindeki harfler verildiği için ve tahminler karakter bazlı yapıldığı için daha fazla hatalı tahmin bulunmaktadır. Kelime hata oranları cümle hata oranları ile karşılaştırıldığında geleneksel modeldeki farkın orantısal olarak daha yüksek olduğu görülebilmektedir. Bu da hataların farklı kayıtlara yayıldığı anlamına gelmektedir. Geleneksel model, Derin Konuşma modeline kıyasla hatalı tahmin edilen her bir kayıtta daha fazla kelimeyi doğru tahmin etmeyi başarabilmiştir.

Çizelge 4.1. Tek konuşmacı tarafından eğitilen modellerin test hata oranları (230 kayıt, 690 kelime)

Model	WER%	SER%
Derin Konuşma	%29.56	%60.43
Geleneksel (GMM-HMM)	%13.04	%30.87
Google API	%22,02	%52,17

Hatalı tahminlerin sebebi hakkında daha derin bilgi sahibi olabilmek adına yanlış kelime tahmini, kelime ekleme ve kelime çıkarma durumları ayrı ayrı hesaplanıp hataya sebep olan bu durumların hatalı tahminler içerisindeki yüzde oranları Şekil 4.1'de görülmektedir. Burada dikkat çeken unsur geleneksel modelin gürültüleri daha zor saptayarak, fazladan sesleri de tahmin etmeye çalışmasıdır. Bu da kelime ekleme oranının diğer modellere kıyasla daha yüksek olmasına sebep olmuştur. Ayrıca fonemlerle istatistiksel eşleştirme yapılırken yanlış bir eşleştirmeden arta kalan konuşma sinyalleri de eşlenerek tahmin edilmeye çalışılmaktadır. Derin konuşma modelinde ise hataların

büyük kısmı harflerin doğru tahmin edilemeyerek kelimenin yanlış oluşturulmasından ya da rastgele harfler algılanarak anlamsız çıktılar oluşturulmasından kaynaklanmaktadır.



Şekil 4.1. Tek konuşmacı tarafından eğitilen modellerin hatalı tahminlerindeki kelime hatalarının, kelime eklemelerin ve çıkarmaların yüzde oranları

Tek konuşmacı tarafından eğitilen modellerin hangi kelimelerin tahminlerinde daha çok hata yaptıklarını ve bunun olası sebeplerini tespit etmek adına test setindeki her bir kelimenin modellere göre hatalı tahmin sayısının ilgili kelimenin testte karşılaşılma sayısına göre yüzde oranı hesaplanarak Çizelge 4.2, Çizelge 4.3 ve Çizelge 4.4'te belirtilmiştir. Derin konuşma modelinde çıktılar karakter temelli tahmin edilmeye çalışıldığından hatalı tahminlerin çoğunlukla belirleyici özellik çıkarımlarının zor olduğu karakterlerden oluşan kelimelerden meydana geldiği söylenebilmektedir. Geleneksel modelde elde edilen sonuçlar ve hatalı tahminlerin detayları incelendiğinde ise hataların çoğunlukla birbirini takip eden fonemlerin benzerliklerinden kaynaklandığı tespit edilmiştir. Örneğin "yeni" ve "yedi" kelimeleri, "onun" ve "on" kelimeleri, "kimin" ve "kim" kelimeleri, "gitti" ve "git" kelimeleri tahminlerde sıklıkla birbirleri ile karıştırılmışlardır. Spontane ve/veya gürültülü konuşmalarda model sinyalin anlamlandırabildiği kısımları üzerinden tahminde bulunmaya çalıştığından, hece ve ses yutma ya da anlaşılamama durumlarında farklı eşleştirmeler doğabilmektedir. Tespit edilen bir diğer durum ise "herkes" ve "erkek eski", "yoksa" ve "yok hala" gibi kelimenin iki farklı kelime olarak algılanarak hem yanlış tahmin hem de kelime ekleme sonucu ile gerçekleşen hatalardır.

Çizelge 4.2. Tek konuşmacıdan oluşan veri seti ile eğitilen ve test edilen Derin Konuşma modelinin kelime başına yüzde hata oranları

adam (%87,5)	altı (%33,33)	amca (%40)	anne (%12,5)	artık (%41,67)
asla (%100)	az (%50)	baba (%50)	bay (%62,5)	bayan (%37,5)
benim (%0)	bey (%57,14)	beş (%20)	bir (%66,67)	bunun (100%)
burada (%25)	bütün (%0)	büyük (%25)	dakika (%0)	dayı (%25)
devam (%50)	dokuz (%14,29)	dostum (%12,5)	dört (%28,57)	efendim (%0)
erkek (%12,5)	eski (%11,11)	evet (%83,33)	eğer (%0)	fazla (%0)
gel (%0)	geldi (%0)	geliyor (%0)	gerçekten (%25)	gidelim (%83,33)
git (%25)	gitti (%0)	güzel (%25)	hakkında (%0)	hala (%75)
hanım (%25)	harika (%9,09)	hayır (%55,56)	herkes (%85,71)	hiç (%16,67)
hiçbir (%0)	imdat (%0)	istiyorum (%0)	iyi (%63,64)	içinde (%0)
işte (%0)	iki (%50)	kadın (%16,67)	kardeş (%14,29)	kim (%40)
kimin (%66,67)	kötü (%12,5)	küçük (%62,5)	lanet (%16,67)	lütfen (%70)
merhaba (%0)	naber (%40)	nasıl (%20)	nasılsın (%33,33)	neden (%33,33)
nerede (%42,86)	olsun (%71,43)	on (%0)	onun (%80)	orada (%0)
para (%25)	pekala (%0)	saat (%14,29)	sadece (%37,5)	saniye (%0)
sanırım (%0)	sekiz (%33,33)	selam (%0)	senin (%57,14)	sonra (%30)
sıfır (%14,29)	tamam (%0)	teşekkür (%0)	var (%0)	yardım (%16,67)
yedi (%55,56)	yeni (%44,44)	yeğen (%62,5)	yok (%0)	yoksa (%12,5)
zaman (%66,67)	çirkin (%11,11)	çocuk (%16,67)	çünkü (%0)	önce (%20)
önemli (%10)	özür (%25)	üç (%25)	şey (%25)	şimdi (%33,33)

Çizelge 4.3. Tek konuşmacıdan oluşan veri seti ile eğitilen ve test edilen geleneksel modelin kelime başına yüzde hata oranları

adam (%50)	altı (%0)	amca (%0)	anne (%0)	artık (%0)
asla (%50)	az (%62,5)	baba (%25)	bay (%12,5)	bayan (%25)
benim (%0)	bey (%14,29)	beş (%0)	bir (%0)	bunun (%14,29)
burada (%0)	bütün (%0)	büyük (%12,5)	dakika (%0)	dayı (%0)
devam (%0)	dokuz (%0)	dostum (%0)	dört (%0)	efendim (%0)
erkek (%12,5)	eski (%0)	evet (%0)	eğer (%0)	fazla (%50)
gel (%25)	geldi (%0)	geliyor (%0)	gerçekten (%0)	gidelim (%33,33)
git (%25)	gitti (%0)	güzel (%0)	hakkında (%0)	hala (%75)
hanım (%0)	harika (%9,09)	hayır (%11,11)	herkes (%42,86)	hiç (%0)
hiçbir (%0)	imdat (%0)	istiyorum (%0)	iyi (%54,55)	içinde (%0)
işte (%0)	iki (%0)	kadın (%0)	kardeş (%0)	kim (%40)
kimin (%100)	kötü (%12,5)	küçük (%0)	lanet (%0)	lütfen (%0)
merhaba (%0)	naber (%40)	nasıl (%60)	nasılsın (%0)	neden (%33,33)
nerede (%14,29)	olsun (%0)	on (%40)	onun (%0)	orada (%0)
para (%62,5)	pekala (%0)	saat (%0)	sadece (%0)	saniye (%0)
sanırım (%0)	sekiz (%0)	selam (%0)	senin (%42,86)	sonra (%10)
sıfır (%0)	tamam (%20)	teşekkür (%0)	var (%0)	yardım (%33,33)
yedi (%11,11)	yeni (%88,89)	yeğen (%25)	yok (%0)	yoksa (%25)
zaman (%33,33)	çirkin (%11,11)	çocuk (%16,67)	çünkü (%0)	önce (%0)
önemli (%0)	özür (%0)	üç (%0)	şey (%25)	şimdi (%0)

Çizelge 4.4. Tek konuşmacıdan oluşan veri seti ile test edilen Google Speech modelinin kelime başına yüzde hata oranları

adam (%0)	altı (%33,33)	amca (%20)	anne (%12,5)	artık (%8,33)
asla (%62,5)	az (%62,5)	baba (%12,5)	bay (%25)	bayan (%0)
benim (%0)	bey (%14,29)	beş (%40)	bir (%33,33)	bunun (%71,43)
burada (%16,67)	bütün (%0)	büyük (%37,5)	dakika (%14,29)	dayı (%0)
devam (%12,5)	dokuz (%28,57)	dostum (%12,5)	dört (%28,57)	efendim (%0)
erkek (%12,5)	eski (%0)	evet (%33,33)	eğer (%25)	fazla (%75)
gel (%50)	geldi (%0)	geliyor (%0)	gerçekten (%0)	gidelim (%33,33)
git (%0)	gitti (%14,29)	güzel (%16,67)	hakkında (%0)	hala (%0)
hanım (%37,5)	harika (%18,18)	hayır (%22,22)	herkes (%14,29)	hiç (%66,67)
hiçbir (%20)	imdat (%0)	istiyorum (%0)	iyi (%54,55)	içinde (%0)
işte (%0)	iki (%25)	kadın (%16,67)	kardeş (%0)	kim (%40)
kimin (%50)	kötü (%37,5)	küçük (%25)	lanet (%33,33)	lütfen (%0)
merhaba (%0)	naber (%0)	nasıl (%0)	nasılsın (%0)	neden (%33,33)
nerede (%14,29)	olsun (%0)	on (%20)	onun (%40)	orada (%16,67)
para (%62,5)	pekala (%28,57)	saat (%57,14)	sadece (%12,5)	saniye (%0)
sanırım (%20)	sekiz (%66,67)	selam (%33,33)	senin (%85,71)	sonra (%0)
sıfır (%28,57)	tamam (%0)	teşekkür (%25)	var (%33,33)	yardım (%16,67)
yedi (%11,11)	yeni (%0)	yeğen (%75)	yok (%0)	yoksa (%25)
zaman (%33,33)	çirkin (%11,11)	çocuk (%16,67)	çünkü (%22,22)	önce (%20)
önemli (%0)	özür (%25)	üç (%12,5)	şey (%25)	şimdi (%0)

Aynı yöntemler kullanılarak birden çok konuşmacı ile eğitilen modellerin, eğitimde konuşma verileri kullanılan konuşmacılar tarafından test edilmesi sonucu elde edilen hata oranları Çizelde 4.5'te görülebilmektedir. Gerçekleştirilen eğitim ve testler sonucu tüm modellerin hata oranlarının tek konuşmacılı deneylere kıyasla düştüğü gözlemlenmiştir.

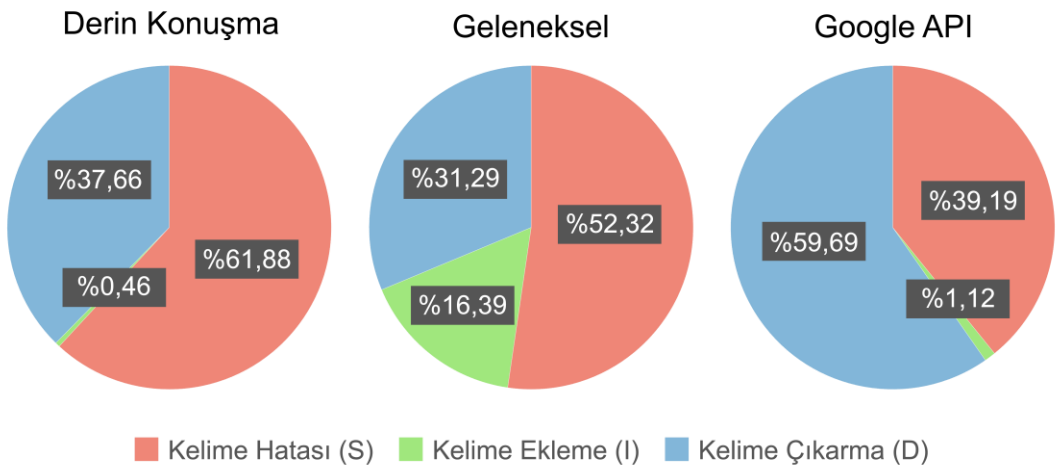
Çizelge 4.5. Birden çok konuşmacı tarafından eğitilen modellerin test hata oranları (1619 kayıt, 4857 kelime)

Model	WER%	SER%
Derin Konuşma	%17,76	%39,03
Geleneksel (GMM-HMM)	%12,43	%29,77
Google API	%14,65	%34,52

Google API testlerinin başarısının yükselmesinin sebebi, farklı konuşmacılardan alınan kayıtlarda, tek konuşmacı ile yapılan testlerde kullanılan kayıtlara kıyasla daha temiz kayıtlar bulunmasıdır. Birden çok konuşmacı ile oluşturulan geleneksel modelin başarısındaki artış tek konuşmacı ile eğitilen modele kıyasla yok denilecek kadar azdır. GMM-HMM temelli geleneksel modelde konuşma külliyatının çeşitliliği değişmediği için veri seti ve konuşmacı sayısındaki artış eğitime fazla bir katkı sağlayamamıştır. Buna karşın temiz kayıtlarda Google ile kıyaslandığında optimal sonucu verdiği gözlemlenen geleneksel modelin, test setlerine göre kullanılan gürültülü kayıtlarda da yüksek başarı elde ettiği söylenebilmektedir. Birden çok konuşmacılı deneylerde asıl ön plana çıkan durum ise Derin Konuşma modelinin başarısındaki artış olmuştur. Birden çok konuşmacılı modelde kullanılan veri seti, tek konuşmacılı modele göre daha fazla veri içerse de söz konusu yapay sinir ağlarını eğitmek olduğunda veri seti boyutu oldukça küçük kalmaktadır. Buna rağmen Derin Konuşma modelinin başarısı ile diğer modeller arasındaki fark büyük ölçüde azalmış, bazı konuşmacıların kayıtlarını doğru tahmin etmede daha yüksek başarı oranları elde edilmiştir.

Birden çok konuşmacı ile eğitilen ve test edilen modellerin tahminlerindeki hatalara sebep olan durumlar Şekil 4.2'de görülmektedir. Elde edilen sonuçlar detaylı incelendiğinde Derin Konuşma modelindeki kelime çıkarma sayısının artmasındaki sebepler irdelenmiş ve tahmin edilen kelimeler doğru olsa da birçok durumda iki farklı kelimenin birleşik tek bir kelime olarak algılandığı tespit edilmiştir. Gürültülü ve

spontane konuşmalardan oluşan kayıtlarda kelimeler arasındaki boşlukların tespit edilememesi otomatik olarak eksik kelime tahmini ile sonuçlanmaktadır. Geleneksel modelde de aynı şekilde çıkarma sayıları artmıştır fakat geleneksel model fonemlerden kelimeleri eşleştirmeye dayanan istatistiksel bir yöntem izlediğinden çıktılar incelendiğinde Derin Konuşma modelindekine benzer bir duruma rastlanmamıştır.



Şekil 4.2. Birden çok konuşmacı tarafından eğitilen modellerin hatalı tahminlerindeki kelime hatalarının, kelime eklemelerin ve çıkarmaların yüzde oranları

Konuşmacılardan toplanan kayıtlar, her bir konuşmacının konuşma tarzı ve kelimeleri kaydederken kullandığı mikrofon kalitesi farklı olduğundan büyük farklılıklar göstermektedir. Konuşmacıların hepsinden internet sitesinde karşılarına çıkan kelimeleri okumaları beklenmesine rağmen her insanın konuşma hızı, telaffuz biçimi farklılık gösterebileceğinden birbirinden bağımsız sonuçlar elde edilmiştir. Konuşma kayıtları başarı oranını etkileyen özelliklerine göre Çizelge 4.6'daki gibi sınıflara ayrılabilmektedirler. Bu ayrım modellerin farklı durumlardaki başarısının gözlemlenebilmesine imkan tanımaktadır. Kayıt kalitesi, kaydın alındığı mikrofon kalitesine ve kayıt alınan ortamın gürültüsüne göre değişiklik gösterebilmektedir. Kayıtlar incelenerek gürültü boyutları tespit edilmeye çalışılmıştır. Farklı ortam ve zamanlarda alınan kayıtların bir kısmının gürültüsüz diğer kısmının gürültülü olduğu durumlar karışık olarak belirtilmiştir. Konuşma modu, konuşmacıların karşılarına çıkan üç kelimeyi ayrı kelimeler şeklinde mi yoksa bir cümlenin öğeleri gibi art arda mı okuduğunu temsil etmektedir. Bazı konuşmacıların kayıtları, kayıt sayısının fazla olması sebebiyle kesintili konuşmadan kesintisiz konuşmaya doğru hızlanmıştır ve bu durumlar

karışık olarak tabloda belirtilmiştir. Konuşma tarzı ise okunan kelimelerin dikte edilerek mi yoksa gündelik hayatta kullanıldığı gibi spontane şekilde mi telaffuz edildiğini göstermektedir. Her iki durumu da barındıran setler yine karışık olarak nitelendirilmiştir.

Çizelge 4.6. Eğitim ve test aşamalarında kullanılan veri setindeki kayıtların özellikleri

Konuşmacı	Kayıt Kalitesi	Konuşma Modu	Konuşma Tarzı
a (tek konuşmacı)	Gürültülü	Karışık	Spontane
a	Karışık	Karışık	Spontane
b	Gürültüsüz	Kesintili	Dikte Edilen
c	Yüksek Gürültülü	Karışık	Spontane
d	Düşük Gürültülü	Karışık	Spontane
e	Gürültüsüz	Kesintili	Karışık
f	Karışık	Kesintili	Dikte Edilen
g	Karışık	Karışık	Dikte Edilen
h	Yüksek Gürültülü	Kesintili	Dikte Edilen
i	Gürültüsüz	Karışık	Karışık

Kayıtlar hakkındaki bilgiler ışığında testler sonucundaki başarı konuşmacı bazlı incelendiğinde modellerin çalışma prensipleri ve farklı durumlardaki başarı oranları daha iyi tespit edilebilmektedir. Çizelge 4.7 ve Çizelge 4.8'de birden çok konuşmacı ile eğitilen ve test edilen modellerin farklı konuşmacılara göre kelime ve cümle hata oranları gösterilmektedir. Elde edilen en yüksek başarıların gürültüsüz bir ortamda, kesintili ve dikte edilen bir konuşma ile gerçekleştirilen kayıtlara sahip "b" konuşmacısının kayıtlarında; en düşük başarı oranlarının ise yüksek gürültülü kayıtlara sahip "c" konuşmacısın kayıtlarında olduğu görülmektedir. Modeller ayrı ayrı incelendiğin Derin Konuşma modelinin kişiye özelleşmiş bir model olmamasından dolayı büyük ölçüde kayıt özelliklerinden etkilendiği söylenebilmektedir. "e" konuşmacısının başarı oranları incelendiğinde eğitimde en az sayısal paya sahip olmasına karşın yüksek bir başarı oranı elde edildiği görülmektedir. Buna karşın geleneksel modelin diğer modellere kıyasla "a" ve "c" konuşmacılarındaki başarısının yüksek olup "f" ve "g" konuşmacılarında konuşma özellikleri de göz önünde bulundurulduğunda beklenenin altında kalması, konuşmacıların eğitimde kullanılan kayıt sayısının, model testinde ilgili konuşmacının konuşmasını

tahmin etmedeki etkisini göstermektedir. Kelime hatalarının cümlelere dağılım oranları tüm modellerde benzerlik göstermektedir ve farklı sonuçlar oluşturmamaktadır.

Çizelge 4.7. Birden çok konuşmacı tarafından eğitilen modellerin konuşmacı bazlı yüzde kelime hata oranları

Konuşmacı	Modellere Göre Kelime Hata Oranları (WER%)		
	Derin Konuşma	**Geleneksel (GMM-HMM)**	**Google API**
a	%15,25	**%5,66**	%18,41
b	%4,70	**%2,42**	**%2,42**
c	%35,52	**%17,00**	%36,70
d	%17,83	%18,83	**%8,50**
e	**%7,97**	%10,86	%18,84
f	%28,50	%20,53	**%14,73**
g	**%15,15**	%20,45	%15,91
h	%15,91	**%10,71**	%10,72
i	%17,14	%16,34	**%10,16**

Çizelge 4.8. Birden çok konuşmacı tarafından eğitilen modellerin konuşmacı bazlı yüzde cümle hata oranları

Konuşmacı	Modellere Göre Cümle Hata Oranları (SER%)		
	Derin Konuşma	**Geleneksel (GMM-HMM)**	**Google API**
a	%34,97	**%16,33**	%39,54
b	%11,11	**%4,70**	**%4,70**
c	%69,70	**%38,38**	%86,87
d	%42,50	%45,00	**%21,00**
e	**%21,74**	%28,26	%45,65
f	%56,52	%44,20	**%34,78**
g	**%37,50**	%45,45	%46,59
h	%36,68	%26,13	**%25,63**
i	%39,05	%42,38	**%24,76**

Çizelge 4.9. Birden çok konuşmacıdan oluşan veri seti ile eğitilen ve test edilen Derin Konuşma modelinin kelime başına yüzde hata oranları

adam (%24,56)	altı (%12,73)	amca (%9,3)	anne (%17,78)	artık (%13,46)
asla (%20,83)	az (%12,96)	baba (%18,97)	bay (%31,82)	bayan (%22,22)
benim (%20,45)	bey (%25,58)	beş (%20)	bir (%33,33)	bunun (%15,91)
burada (%18,37)	bütün (%27,08)	büyük (%27,66)	dakika (%13,64)	dayı (%21,95)
devam (%16,36)	dokuz (%11,9)	dostum (%19,64)	dört (%8,89)	efendim (%26,19)
erkek (%25,93)	eski (%15,25)	evet (%12,5)	eğer (%10,71)	fazla (%11,11)
gel (%24,07)	geldi (%17,86)	geliyor (%8,11)	gerçekten (%7,14)	gidelim (%21,74)
git (%39,58)	gitti (%18,75)	güzel (%17,86)	hakkında (%2,17)	hala (%22,64)
hanım (%14,04)	harika (%7,02)	hayır (%15,38)	herkes (%10,42)	hiç (%18,33)
hiçbir (%31,25)	imdat (%13,33)	istiyorum (%5,56)	iyi (%26,67)	içinde (%8)
işte (%17,78)	iki (%29,17)	kadın (%2,04)	kardeş (%12,5)	kim (%22,58)
kimin (%12,96)	kötü (%31,15)	küçük (%30,77)	lanet (%19,61)	lütfen (%19,57)
merhaba (%17,24)	naber (%16,18)	nasıl (%21,62)	nasılsın (%5,66)	neden (%12,77)
nerede (%12,5)	olsun (%18,6)	on (%32,35)	onun (%24,32)	orada (%13,21)
para (%32,31)	pekala (%23,08)	saat (%17,78)	sadece (%7,14)	saniye (%6,98)
sanırım (%15,09)	sekiz (%27,12)	selam (%17,02)	senin (%15,91)	sonra (%12,28)
sıfır (%15,38)	tamam (%11,76)	teşekkür (%2,38)	var (%23,81)	yardım (%11,9)
yedi (%34,15)	yeni (%34,78)	yeğen (%16,67)	yok (%17,39)	yoksa (%9,62)
zaman (%26,32)	çirkin (%4)	çocuk (%15,79)	çünkü (%17,24)	önce (%5,17)
önemli (%15,56)	özür (%23,91)	üç (%15,22)	şey (%7,69)	şimdi (%15,79)

Çizelge 4.10. Birden çok konuşmacıdan oluşan veri seti ile eğitilen ve test edilen geleneksel modelin kelime başına yüzde hata oranları

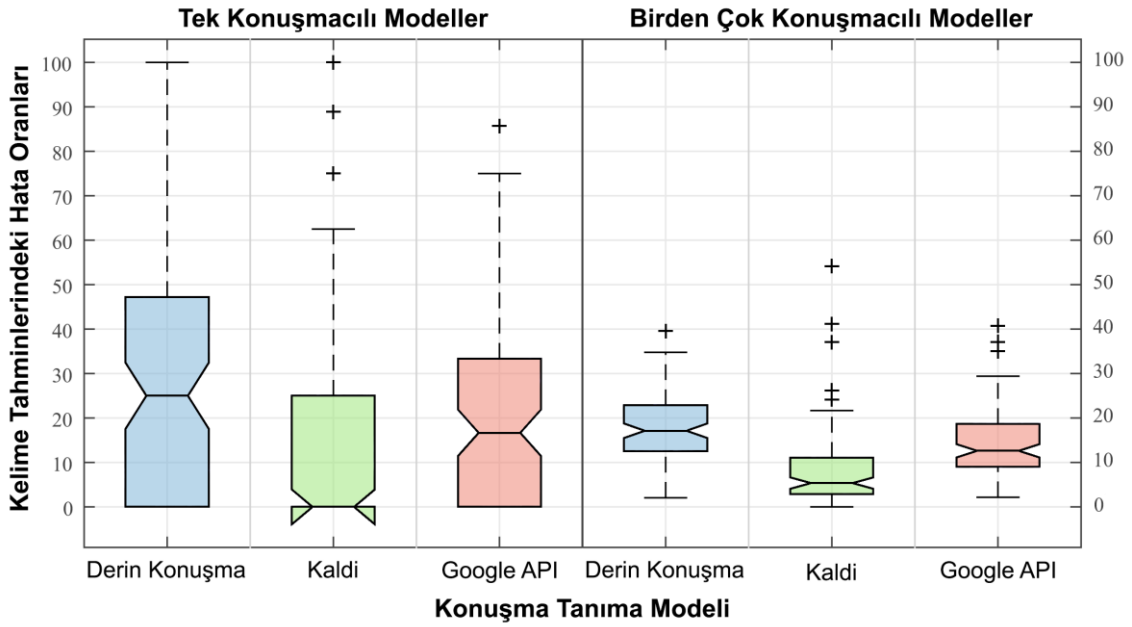
adam (%7,02)	altı (%1,82)	amca (%4,65)	anne (%2,22)	artık (%3,85)
asla (%2,08)	az (%16,67)	baba (%5,17)	bay (%4,55)	bayan (%2,22)
benim (%13,64)	bey (%20,93)	beş (%10)	bir (%26,19)	bunun (%11,36)
burada (%0)	bütün (%16,67)	büyük (%4,26)	dakika (%2,27)	dayı (%7,32)
devam (%5,45)	dokuz (%4,76)	dostum (%14,29)	dört (%2,22)	efendim (%7,14)
erkek (%14,81)	eski (%6,78)	evet (%2,5)	eğer (%1,79)	fazla (%5,56)
gel (%37,04)	geldi (%3,57)	geliyor (%2,7)	gerçekten (%0)	gidelim (%2,17)
git (%54,17)	gitti (%18,75)	güzel (%1,79)	hakkında (%8,7)	hala (%20,75)
hanım (%8,77)	harika (%5,26)	hayır (%5,77)	herkes (%4,17)	hiç (%21,67)
hiçbir (%4,17)	imdat (%4,44)	istiyorum (%2,78)	iyi (%8,89)	içinde (%4)
işte (%8,89)	iki (%4,17)	kadın (%0)	kardeş (%10,71)	kim (%24,19)
kimin (%3,7)	kötü (%21,31)	küçük (%5,13)	lanet (%9,8)	lütfen (%4,35)
merhaba (%3,45)	naber (%2,94)	nasıl (%18,92)	nasılsın (%5,66)	neden (%17,02)
nerede (%4,17)	olsun (%2,33)	on (%41,18)	onun (%13,51)	orada (%7,55)
para (%9,23)	pekala (%3,85)	saat (%2,22)	sadece (%0)	saniye (%6,98)
sanırım (%3,77)	sekiz (%15,25)	selam (%2,13)	senin (%11,36)	sonra (%3,51)
sıfır (%7,69)	tamam (%13,73)	teşekkür (%0)	var (%14,29)	yardım (%7,14)
yedi (%9,76)	yeni (%13,04)	yeğen (%9,52)	yok (%15,22)	yoksa (%0)
zaman (%2,63)	çirkin (%6)	çocuk (%2,63)	çünkü (%5,17)	önce (%3,45)
önemli (%0)	özür (%4,35)	üç (%8,7)	şey (%7,69)	şimdi (%2,63)

Çizelge 4.11. Birden çok konuşmacıdan oluşan veri seti ile test edilen Google Speech modelinin kelime başına yüzde hata oranları

adam (%5,26)	altı (%12,73)	amca (%11,63)	anne (%15,56)	artık (%11,54)
asla (%25)	az (%40,74)	baba (%8,62)	bay (%25)	bayan (%4,44)
benim (%15,91)	bey (%23,26)	beş (%8)	bir (%11,9)	bunun (%27,27)
burada (%14,29)	bütün (%16,67)	büyük (%17,02)	dakika (%11,36)	dayı (%7,32)
devam (%10,91)	dokuz (%11,9)	dostum (%16,07)	dört (%11,11)	efendim (%14,29)
erkek (%9,26)	eski (%10,17)	evet (%20)	eğer (%25)	fazla (%14,81)
gel (%18,52)	geldi (%3,57)	geliyor (%10,81)	gerçekten (%9,52)	gidelim (%19,57)
git (%16,67)	gitti (%12,5)	güzel (%12,5)	hakkında (%6,52)	hala (%15,09)
hanım (%14,04)	harika (%10,53)	hayır (%7,69)	herkes (%6,25)	hiç (%35)
hiçbir (%22,92)	imdat (%4,44)	istiyorum (%11,11)	iyi (%26,67)	içinde (%10)
işte (%11,11)	iki (%18,75)	kadın (%8,16)	kardeş (%10,71)	kim (%37,1)
kimin (%12,96)	kötü (%14,75)	küçük (%28,21)	lanet (%17,65)	lütfen (%2,17)
merhaba (%5,17)	naber (%7,35)	nasıl (%5,41)	nasılsın (%9,43)	neden (%12,77)
nerede (%8,33)	olsun (%9,30)	on (%29,41)	onun (%21,62)	orada (%9,43)
para (%29,23)	pekala (%23,08)	saat (%8,89)	sadece (%2,38)	saniye (%16,28)
sanırım (%13,21)	sekiz (%22,03)	selam (%8,51)	senin (%20,45)	sonra (%14,04)
sıfır (%11,54)	tamam (%5,88)	teşekkür (%11,9)	var (%14,29)	yardım (%9,52)
yedi (%2,44)	yeni (%19,57)	yeğen (%28,57)	yok (%4,35)	yoksa (%7,69)
zaman (%13,16)	çirkin (%22)	çocuk (%7,89)	çünkü (%17,24)	önce (%13,79)
önemli (%6,67)	özür (%23,91)	üç (%19,57)	şey (%17,31)	şimdi (%10,53)

Derin Konuşma'da kelime hatalarının tek konuşmacılı modele kıyasla daha dengeli olmasından dolayı hata sebebinin kelime özellikleri yerine kayıt özelliklerinden kaynaklandığı düşünülmektedir (bkz. Çizelge 4.9). Geleneksel modelde ise yine tek konuşmacılı modelde olduğu gibi fonem benzerliği olan kelimelerde başarı oranları daha düşüktür (bkz. Çizelge 4.10).

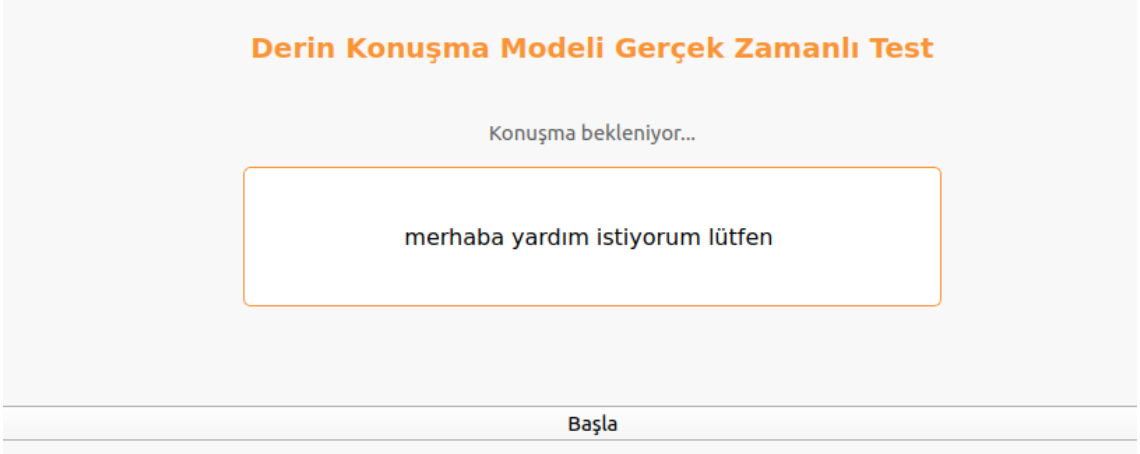
Model testlerindeki kelime tahmin hata oranlarını gösteren kutu grafiği Şekil 4.3'te görüldüğü gibidir. Eğitim ve testte kullanılan kayıt ve konuşmacı sayısının artmasının yanı sıra konuşma özellikleri ve kayıt kalitesinin değişmesi ile birlikte uç noktalardaki hatalar azalmış, model tahminleri daha tutarlı hale gelmiştir. Yine grafikten görülebileceği üzere bu durumdan en çok etkilenen model, eğitim aşamasındaki veri sayısının da artması sayesinde Derin Konuşma modelidir.



Şekil 4.3. Testler sonucunda kelime tahminlerindeki hata oranlarını gösteren kutu grafiği

Derin Konuşma modelini gerçek zamanlı test etmek adına birden çok konuşmacı tarafından eğitilen model kullanılarak python dilinde ve kullanıcı arayüzü için PyQt5 yazılımından da faydalanılarak bir masaüstü uygulaması oluşturulmuştur. Mikrofon girdilerini kesintisiz dinleyen uygulama, ses algılandığında dinleme moduna geçerek konuşma tamamlandığında elde edilen veriyi metne çevirerek kullanıcıya göstermektedir. Çevirme işleminin konuşma süresince değil de konuşma işlemi bittiğinde

tamamlanmasının temel sebebi geliştirilen modelde geri yineleme işleminin kullanılmasıdır. Örnek ekran görüntüsü Şekil 4.4'te görülebilecek program yalnızca modeli gerçek zamanlı test etmek için geliştirildiğinden arayüzü oldukça basit tutulmuştur. Geliştirilen uygulama kullanılarak kaydedilen örnek test demolarına https://youtu.be/7vDvPkH5wdk ve https://youtu.be/Qfv-Vx0nl9E internet adreslerinden ulaşılabilmektedir.



Şekil 4.4. Derin Konuşma modelinin gerçek zamanlı testi için geliştirilen yazılımın arayüzü

5. SONUÇ

Gerçekleştirilen eğitimler, testler ve test çıktılarının analizleri doğrultusunda modellerin farklı özellik ve boyutlardaki veri setleri ile işlenmesi sonucunda başarı oranlarının değişimleri gözlemlenmiştir. Elde edilen sonuçlara göre uçtan uca derin öğrenme ile gerçekleştirilen Derin Konuşma yöntemi veri seti boyutundan büyük ölçüde etkilenmektedir. Ayrıca konuşmacı bağımlılığı GMM-HMM temelli geleneksel modele göre beklenildiği üzere daha düşük çıkmıştır. Geleneksel modelin başarısı konuşma külliyatının kısıtlanmış olmasına, akustik modele sağlanan fonem sözlüğüne ve konuşmacı etiketli kayıtlar tarafından eğitilmesine bağlanmaktadır. Sınırlı bir konuşma külliyatını algılayabilecek ve konuşmacıya bağımlı bir yöntem geliştirilmek istendiğinde, olasılıklara göre eşleştirme tabanlı çalışan geleneksel yöntemler eğitim hızı ve başarı oranlarının yüksek olması sebebiyle tercih edilebilmektedir. Dezavantaj olarak ise verileri etiketlemek, fonem sözlüğünün hazırlanması ve bileşenlerin oluşturulması geliştiriciye büyük yük olmaktadır. Bu tarz modeller bütün bir dile uygulanmak istendiğinde dile uygun fonem sözlüğünün hazırlanması bile başlı başına büyük bir külfettir. Derin konuşma modelinde ise konuşma verileri yalnızca konuşma sinyalleri ve bu sinyale karşılık gelen metinlerden oluşmaktadır. Modellemede TSA'lardan faydalanıldığı için özellik çıkarımı Google Tensorflow kütüphanesi ile gerçekleştirilmektedir. Derin öğrenme kullanımı iki temel dezavantaja yol açmaktadır. İlk olarak ağın öğrenebilmesi için geleneksel yöntemlere kıyasla fazla sayıda etiketli veriye ihtiyaç duyulmaktadır. Bu sebeple derin öğrenme algoritmalarından faydalanılarak oluşturulmak istenen modeller için çeşitli etiketli veri toplama yöntemlerinin geliştirilmesi gerekebilmektedir. Bir diğer konu ise derin öğrenme ağlarının eğitimi yüksek işlemci gücü gerektirmektedir. Aksi halde bir dilin tamamını tanıyabilecek bir modelin oluşturulması yıllar sürebilecek bir görev haline dönüşmektedir. Grafik işlemci ünitelerinin hızla gelişmesi sayesinde eğitim süreçleri çok kısa sürelere inebilmektedir ancak güçlü bir donanımsal altyapı olmadan kapsamlı bir eğitim ne yazık ki mümkün değildir.

İlerleyen çalışmalarda derin öğrenmenin çalışma prensibinden ve elde edilen sonuçlardan yola çıkarak veri setini büyütmek ve çeşitlendirmek amacıyla geliştirilen internet sitesine kullanıcı kontrolü koyulması hedeflenmektedir. Bu sayede konuşmacıların kayıtlarının doğruluğunu yine diğer kullanıcılar kontrol edebileceği için site halka açık kullanıma

sunulabilecektir. Aynı zamanda veri setinin büyümesi sayesinde kelimeleri ayrı ayrı tanıyan ve 100 kelimelik bir dağarcığa sahip olan sistem, Türkçe dilindeki tüm kelimelere genişletilip, cümle tanımaya adapte edilebilecektir. Veri setini genişletmek adına sesli kitaplar, Türkçe film altyazıları gibi kaynaklardan veri toplanarak veri setinin çeşitlendirilmesi de planlanmaktadır. Elde edilen tüm bu verilerin işlenmesi büyük bir işlem yükü getireceğinden eğitim aşamasının birçok grafik işlemci ünitesi üzerinden paralel olarak gerçekleştirilmesi işlem yükünü bölerek süreci hızlandıracaktır ve büyük verilerin eğitimi çok daha kısa sürelere indirgenebilecektir.

KAYNAKLAR

Anonim 2020a. IBM Shoebox https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html-(Erişim tarihi:20.12.2020)

Anonim 2020b. Pioneering Speech Recognition https://www.ibm.com/ibm/history/ibm100/us/en/icons/speechreco/-(Erişim tarihi:20.12.2020)

Anonim 2020c. Structure of a typical neuron https://en.wikipedia.org/wiki/Dendrite-(Erişim tarihi:20.12.2020)

Baker, J. K. 1975. The DRAGON System-An Overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1): 24–29. https://doi.org/10.1109/TASSP.1975.1162650

Baum, L. E., Petrie, T. 1966. Statistical inference for finite state markov chains. *The Annals of Mathematical Statistics*, 37: 1554–1563. https://doi.org/10.1214/aoms/1177699147

Bourlard, H. A., Morgan, N. 1994. Connectionist Speech Recognition. Kluwer Academic Publishers, Boston, 261 pp. https://doi.org/10.1007/978-1-4615-3210-1

Chen, Y., Lin, Y., Kung, C., Chung, M., Yen, I. 2019. Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes. *Sensors (Basel)*, 19(9): 2047. https://doi.org/10.3390/s19092047

Davis, K. H., Biddulph, R., Balashek, S. 1952. Automatic Recognition of Spoken Digits. J*ournal of the Acoustical Society of America*, 24(6): 637–642. https://doi.org/10.1121/1.1906946

Eisner, J. 2002. An interactive spreadsheet for teaching the forward-backward algorithm. Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, (July): 10–18. https://doi.org/10.3115/1118108.1118110

Giorgino, T. 2009. Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7): 1–24. https://doi.org/10.18637/jss.v031.i07

Gray, R. M. 2010. Linear Predictive Coding and the Internet Protocol. Now Publishers Inc, USA, 148 pp.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A. Y. 2014. Deep Speech: Scaling up end-to-end speech recognition. , *ArXiv*, abs/1412.5567.

Hochreiter, S., Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, X., Baker, J., Reddy, R. 2014. A historical perspective of speech recognition. *Communications of the ACM*, 57(1): 94–103. https://doi.org/10.1145/2500887

Jurafsky, D., Martin, J. H. 2009. Speech and Language Processing. Prentice Hall, USA, 615 pp.

Medsker, L. R., Jain, L. C. 2013. Recurrent Neural Networks Design and Applications. *Journal of Chemical Information and Modeling*, 53(9): 1689–1699.

Meeker, M. 2017. Internet Trends 2017.. Kleiner Perkins Caufield & Byers (KPCB), California.

Mohri, M., Pereira, F., Riley, M. 2002. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech & Language*, 16(1): 69–88.

Mohri, M., Rostamizadeh, A., Talwalkar, A. 2018. Foundations of Machine Learning. The MIT Press, London, 486pp.

Nassif, A., B., Shahin, I., Attili, I., Azzeh, M., Shaalan, K. 2019. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access*, 7: 19143-19165. https://doi.org/10.1109/ACCESS.2019.2896880

Paliwal, K. K., Agarwal, A., Sinha, S. S. 1982. A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition. *Signal Processing*, 4(4): 329–333. https://doi.org/10.1016/0165-1684(82)90009-3

Povey, D., Ghoshal, A.,, Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. 2011. The Kaldi Speech Recognition Toolkit. Automatic Speech Recognition and Understanding Workshop (2011)

Rabiner, L. R. 1989. Tutorial on Hmm and Applications. *Proceedings of the IEEE*, 77(2): 257–286. https://doi.org/10.1109/5.18626

Reddy, D. R., Erman, L. D., Fennell, R. D., Lowerre, B. T., Neely, R. B. 1974. The HEARSAY Speech Understanding System. *Journal of the Acoustical Society of America*, 55(2): 409. https://doi.org/10.1121/1.3437267

Rojas, R. 1996. Neural Networks, Springer, Berlin, 502 pp. https://doi.org/978-3-642-61068-4

Sutton, R. S., Barto, A. G. 2018. Reinforcement Learning: An Introduction, The MIT Press, London, 526 pp.

Young, S. J., Odell, J. J., Woodland, P. C. 1994. Tree-based state tying for high accuracy acoustic modelling. ARPA workshop Human Language Technology, 1994, Princeton, New Jersey, pp. 307-312. https://doi.org/10.3115/1075812.1075885

ÖZGEÇMİŞ

Adı Soyadı : Burak KORCUKLU
Doğum Yeri ve Tarihi : Yıldırım 29.03.1992
Yabancı Dil : İngilizce

Eğitim Durumu
Lise : Özel Tan Fen Lisesi
Lisans : Ege Üniversitesi Bilgisayar Mühendisliği
Yüksek Lisans : Bursa Uludağ Üniversitesi Bilgisayar Mühendisliği

Çalıştığı Kurum/Kurumlar : ByGO Digital, Bursa Uludağ Üniversitesi

İletişim (e-posta) : burakkorcuklu@gmail.com

Yayınları :

Karpat, F., Dirik, A. E., Dogan, O., Kalay, O. C., Korcuklu, B., Yuce, C. 2020. A Novel AI-Based Method for Spur Gear Early Fault Diagnosis in Railway Gearboxes. Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı, 1–6. https://doi.org/10.1109/asyu50717.2020.9259819