

**BİR HECE-TABANLI TÜRKÇE SESLİ İFADE TANIMA
SİSTEMİNİN TASARIMI VE GERÇEKLEŞTİRİMİ**

**A SYLLABLE-BASED SPEECH RECOGNITION SYSTEM
DESIGN AND IMPLEMENTATION FOR TURKISH
LANGUAGE**

BURCU CAN

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Bölümü için Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2007

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :.....
(Prof. Dr. Ersin TÖRECİ)

Üye (Danışman) :.....
(Yrd. Doç. Dr. Harun ARTUNER)

Üye :.....
(Yrd. Doç. Dr. Ali Ziya ALKAR)

ONAY

Bu tez/...../..... tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

...../...../.....

Prof. Dr. Erdem YAZGAN

Fen Bilimleri Enstitüsü Müdürü

aieme

BİR HECE-TABANLI TÜRKÇE SESLİ İFADE TANIMA SİSTEMİNİN TASARIMI VE GERÇEKLEŞTİRİMİ

Burcu Can

ÖZ

Sesli ifade tanıma sistemlerinde kullanılan yöntemler, tanınması amaçlanan dile göre değişmektedir. Türkçenin çekimli bir dil olması nedeniyle, sözcük tabanlı bir Türkçe sesli ifade tanıma sisteminin geliştirilmesi oldukça zordur. Bu sebeple, şu ana kadar geliştirilen Türkçe sesli ifade tanıma sistemlerinin büyük bir kısmı fonem tabanlı olmuştur. Fonem tabanlı sesli ifade tanıma sistemlerinin geliştirilmesi, fonem sayılarının her dilde sınırlı sayıda olması nedeniyle avantajlıdır.

Sesli ifade tanıma sistemlerinin kullandıkları birim, sistemin başarısını doğrudan etkiler. Birim boyutunun büyümesi, birimlerin birbirinden ayırt edilmesini kolaylaştıracaktır. Bu nedenle ayrışık sözcük tanıma sistemlerinin başarısı oldukça yüksektir. Fonemlerin ise boyut olarak nispeten küçük olması fonem tabanlı sesli ifade tanıma sistemlerinin başarısını olumsuz yönde etkilemektedir. Her iki birimin de dezavantajları göz önünde bulundurulduğunda, boyut olarak fonemden büyük sayı olarak ise Türkçede sınırlı bir birim olan hecenin kullanılmasına bu tez kapsamında karar verilmiştir. Türkçenin hecelemeğe yatkın bir dil olması ve Türkçe hece kurallarının basit olması Türkçe sesli ifade tanıma sisteminin hece tabanlı olarak geliştirilmek istenmesinde diğer nedenler olarak sayılabilir. Tez kapsamında geliştirilen sesli ifade tanıma sisteminde hecelerin sınıflandırılmasında Çok Katmanlı Nöron ağlarından yararlanılmıştır. Deney, farklı özellikteki çok katmanlı nöron ağları kullanılarak, sürekli sesli ifadeler için ve sürekli sesli ifadelerle dayalı hecelerin sınırları belirlenmiş olarak, olmak üzere iki farklı şekilde yapılmıştır.

Sonuç olarak, hece tabanlı bir Türkçe sesli ifade tanıma sistemi için gerekli deneysel uygulama ve deney platformu oluşturulmuştur.

Anahtar Kelimeler: Türkçe sesli ifade tanıma, hece-tabanlı, yapay sinir ağları, sayısal im işleme

Danışman: Yrd. Doç. Dr. Harun ARTUNER, Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Bilgisayar Mühendisliği Anabilim Dalı

A SYLLABLE-BASED SPEECH RECOGNITION SYSTEM DESIGN AND IMPLEMENTATION FOR TURKISH LANGUAGE

Burcu Can

ABSTRACT

Methods used in speech recognition systems change with the language which is aimed to recognize. Since the words in Turkish language are mostly produced by derivational and inflectional affixes to roots, it is very hard to develop a word-based Turkish speech recognition system. For this reason, most of the Turkish speech recognitions developed up to now have been phoneme-based. It is advantageous to develop phoneme-based speech recognition systems because the number of phonemes in every language is limited.

The unit used by speech recognition systems effects the system's success directly. As the unit's size grows, it becomes easier to discriminate the units from each other. For this reason, isolated word recognition systems are very successful. Also since it effects the phoneme-based speech recognition systems' success negatively of the units the size of the phonemes is smaller. When the disadvantages of these two units are considered, we decided to use syllables which are larger than phoneme in size and which is limited in number in Turkish language. The other reasons for developing the Turkish speech recognition as syllable-based are that the Turkish can be read syllable by syllable and the rules for syllables are very simple. Within the scope of this thesis work multi layer perceptrons are used to classify the syllables. Two separate experiments have been performed for continuous speech and for the separated syllables from continuous speech by using multi layer perceptrons.

In conclusion, an experimental application and an experiment platform have been implemented for a syllable-based Turkish speech recognition system.

Keywords: Turkish speech recognition system, syllable-based, artificial neural networks, digital signal processing

Advisor: Ast. Prof. Dr. Harun ARTUNER, Hacettepe University, Department of Computer Engineering, Computer Engineering Section

TEŞEKKÜR

Yazar, bu çalışmanın gerçekleşmesinde katkılarından dolayı, aşağıda adı geçen kişi ve kuruluşlara içtenlikle teşekkür eder.

Yrd. Doç. Dr. Harun Artuner (Tez danışmanı), tez konusunun belirlenmesini sağlamış, tez çalışmamın her aşamasında görüşleriyle ve geniş kaynak kütüphanesiyle yol gösterici olmuş ve tez çalışmasının gerçekleştirilmesi için gerekli araçları edinmemde yardımcı olmuştur. Tez metninin hazırlanması sırasında içeriğin oluşturulmasında önemli katkı sağlamış, tez metnini inceleyerek biçim ve içerik açısından düzeltilmesinde ve tezin sunulmasında gerekli her türlü desteği sağlamıştır.

Prof. Dr. Ersin Töreci, çalışmanın gerçekleştirilmesi için gerekli ortamı sağlamış ve tez metnini inceleyerek kolay anlaşılır ve çelişkisiz bir metnin ortaya çıkmasında yardımcı olmuştur.

Yrd. Doç. Dr. Ali Ziya Alkar tez metnini ayrıntılı olarak inceleyerek düzeltilmesinde yardımcı olmuş ve biçim ve içerik bakımından son şeklini almasında katkıda bulunmuştur.

Caner Şahin ve Ersin Er, tez çalışmamda görüşleriyle yol gösterici olmuşlardır.

Barış Atabay, İbrahim Tanrıverdi, Mustafa Daşgın tez çalışmamda gerekli ortamın sağlanmasında yardımcı olmuşlardır.

Ailem her daim olduğu gibi, tez çalışma sürecinde de hep yanımda olmuşlar, manevi destekleriyle hiçbir zaman beni yalnız bırakmamışlardır.

Eren Buğlalılar, her türlü maddi ve manevi desteğiyle beni hiç yalnız bırakmamış, tez metninin yazımında yabancı dille ilgili sorunların çözümünde yardımcı olmuştur.

Ayrıca Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü'ndeki diğer tüm araştırma görevlisi arkadaşlarım tez süresince moral destek vermişlerdir.

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZ	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
SİMGELER ve KISALTMALAR DİZİNİ	x
1 GİRİŞ	1
2 SESLİ İFADE TANIMLARI	3
2.1 Sesin Fiziksel Özellikleri	3
2.2 Sesli İfadelerde Ses, Fon (Sesbirim), Fonem, Hece ve Sözcük	6
2.2.1 Ses (<i>phone</i>), fonem (<i>phoneme</i>)	6
2.2.2 Hece	9
2.2.3 Sözcük	11
2.3 Türkçenin Parçalı Sesbirimleri	11
2.3.1 Ünlüler	11
2.3.2 Ünsüzler	12
3 SESLİ İFADE TANIMAYA GENEL BAKIŞ	15
3.1 Sesli İfade Tanıma Sistemleri	18
3.2 Sesli İfade Tanımaya İlişkin Özellikler	19
3.2.1 Konuşmacıdan Bağımsızlık	19
3.2.2 Ses Sinyalinin Niteliği	21
3.2.3 Alıştırma Gereği	21
3.2.4 Sözlük Büyüklüğü	21
4 SESLİ İFADE TANIMA SÜRECİ	23
4.1 Sesli İfadelerin Sayısallaştırılması	23
4.2 Sesli İfadeler Üzerinde Yürütülen Ön İşlemler	24
4.3 Özellik Vektörünün Çıkarılması	27
4.4 Özellik Vektörlerinin Sınıflandırılması	30
4.4.1 Hidden Markov Model	30
4.4.2 Time Warping-Dynamic Time Warping Yaklaşımı	36
4.4.3 Nöron Ağları Yaklaşımı	37
4.4.3.1 Denetimli Öğrenme (<i>Supervised Learning</i>)	42
4.4.3.1.1 İleri Beslemeli Nöron Ağları (Feedforward Neural Networks) ...	42
4.4.3.1.2 Özyineli Nöron Ağları (Recurrent Neural Networks)	45
4.4.3.2 Destekli Öğrenme (Semi-Supervised/Reinforcement Learning)	45
4.4.3.3 Denetimsiz Öğrenme (Unsupervised Learning)	46
5 BİR HECE TABANLI TÜRKÇE SESLİ İFADE TANIMA SİSTEMİNİN	
TASARIMI VE GERÇEKLEŞTİRİMİ	47
5.1 Deneysel Çalışmanın Aşamaları	47
5.1.1 Hece Sınırlarının Belirlenmesi (Etiketleme)	48
5.1.2 Hecelere Ait Özellik Vektörlerinin Çıkarılması	50
5.1.3 Sınıflandırıcı Sistem	51
5.2 Hece Etiketleme ve Hece Konum Tespiti Sistemi	51
5.3 Deneyler	58
6 SONUÇ, TARTIŞMA ve ÖNERİLER	71
7 KAYNAKLAR	73

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1 Ses dalgasının havada yayılımı	3
Şekil 2.2 İnsan kulağına ilişkin olarak sesin eş yükseklik eğrileri.....	5
Şekil 3.1 Sesli İfade Tanıma Sisteminin İşlevi	16
Şekil 3.2 Sesli ifadeye ait kaynak-kanal modeli	17
Şekil 3.3 Dilbilimsel Kod Çözücü	17
Şekil 4.1 Örnekselel/Sayısal Dönüştürücünün iç yapısı	23
Şekil 4.2 Pencereleme fonksiyonu	25
Şekil 4.3 Mel-Ölçeği Sayısal Filtre Dizisi (<i>Mel-Scale Filter Bank</i>)	29
Şekil 4.4 Hidden Markov Model'de durumlar ve durumlar arası geçişler	31
Şekil 4.5 İMKB'ye ilişkin bir Markov zinciri	33
Şekil 4.6 İMKB'ye ilişkin <i>Markov</i> zincirine ait çıktı olasılık yoğunluk fonksiyonu, başlangıç durum olasılıkları	35
Şekil 4.7 Bir fonem tabanlı sesli ifade tanıma sistemine ait <i>Markov</i> zinciri	36
Şekil 4.8 Farklı yapay sinir ağı modelleri a) yapısal olmayan, b) katmanlı, c) özyi- neli, d) modüler.....	38
Şekil 4.9 Bir nöronun çıktı değerinin hesaplanması.....	39
Şekil 4.10 Belirleyici yerel aktivasyon fonksiyonlar: a)doğrusal (<i>linear</i>), b) <i>threshold</i> , c) <i>sigmoidal</i>	40
Şekil 4.11 <i>Perceptron</i> modelleri a) Tek katmanlı nöron ağı, b) Çok katmanlı nöron ağı	42
Şekil 4.12 Geri Yayılım Öğrenme Algoritması	43
Şekil 4.13 b,d,g fonemlerini tanıma amaçlı geliştirilmiş bir zaman gecikmeli nöron ağı modeli.....	44
Şekil 4.14 Bir özyineli nöron ağı modeli.....	45
Şekil 5.1 Türkçe hecelerin sınıflandırılmasında kullanılan sistemin iş-akış çizelgesi	48
Şekil 5.2 devlet sözcüğüne ait dalga biçimi, 3 boyutlu spektrogram ve enerji değişimi grafikleri.....	50
Şekil 5.3 . <i>Neuro Solutions</i> ile modellenmiş 2 katmanlı yapay nöron ağı mimarisi	51
Şekil 5.4 Geliştirilen araç - Hece Etiketleme ve Hece Konum Tespiti Sistemi (HEKTİS).....	52
Şekil 5.5 HEKTİS kullanımı. Denemeler adlı esere ait sesli ifade sinyalinin görüntülenmesi.....	54
Şekil 5.6 HEKTİS'te hece etiketleme işlemi.....	55
Şekil 5.7 HEKTİS'te sıfırdan geçiş oranını gösterir grafiğin görüntülenmesi.....	56
Şekil 5.8 HEKTİS'te enerji grafiğinin görüntülenmesi	57
Şekil 5.9 HEKTİS yazılımında sinyal üzerinde ilerlenmesi	58
Şekil 5.10 Bir sözcüklük sesli ifade sinyaline ilişkin örnek enerji ve sıfırdan geçiş oranı ölçümü.....	63
Şekil 5.11 . <i>Rabiner</i> ve <i>Sambur</i> 'un sesli ifade sinyalinin başlangıç noktasını bulmak için geliştirdikleri algoritmaya ait akış çizgesi	65
Şekil 5.12 . <i>Rabiner</i> ve <i>Sambur</i> 'un sesli ifade sinyalinin bitiş noktasını bulmak için geliştirdikleri algoritmaya ait akış çizgesi	66
Şekil 5.13 Sürekli sesli ifade üzerinden sınıflandırıcı sistem aracılığı ile hecelerin tanınması.....	68
Şekil 5.14 Sürekli sesli ifadeye ait ayrık hecelerin sınıflandırıcı sistem aracılığı ile tanınması.....	68

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2.1 Sesin algılanabilen ve fiziksel nicelikleri arasındaki ilişki	4
Çizelge 2.2 Türkçede hece türleri (yabancı dillerden dilimize giren sözcüklerin içerdiği heceler de dahil).....	10
Çizelge 5.1 Montaigne'in Denemeler adlı eserinde geçen hece türleri (U: ünlü harf, Z: ünsüz harf) ve bu hecelere ait sayı ve oranlar.....	59
Çizelge 5.2 Gazetelerden elde edilen arşivde geçen hece türleri ve bu hece türlerine ait sayı ve oranlar	60
Çizelge 5.3 Montaigne'in Denemeler adlı eserinde en sık karşılaşılan, tek veya iki karakterden oluşan ilk 50 heceye ait rastlanma sıklıkları	62
Çizelge 5.4 Farklı gazetelerden elde edilen arşivde en sık karşılaşılan, tek veya iki karakterden oluşan ilk 50 heceye ait rastlanma sıklıkları	62
Çizelge 5.5 Geliştirilen sesli ifade tanıma sisteminin, sürekli sesli ifade sinyali ile farklı heceler için farklı özellikteki nöron ağları ile kullanımı	67
Çizelge 5.6 Geliştirilen sesli ifade sisteminin farklı sayıda ayırık heceler için farklı özellikteki nöron ağları ile kullanımı	69

SİMGELER ve KISALTMALAR DİZİNİ

ADPCM	Adapted Differential Pulse Code Modulation
APCM	Adaptive Pulse Code Modulation
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
DCT	Discrete Cosine Transform
DM	Delta Modulation
DPCM	Differantial Pulse Code Modulation
DTW	Dynamic Time Warping
FFT	Fast Fourier Transformation
HEKTİS	Hece Etiketleme ve Hece Konum Tespiti Sistemi
HMM	Hidden Markov Model
ICZT	Interval Zero Crossing Threshold
IPA	International Phonetic Alphabet
ITL	Interval Threshold Low
ITU	Interval Threshold Upper
İMKB	İstanbul Menkul Kıymetler Borsası
LFCC	Linear Frequency Cepstrum Coefficients
Log-PCM	Logarithmic Pulse Code Modulation
LPCC	Linear Predictive Cepstrum Coefficients
MFCC	Mel-Frequency Cepstral Coefficients
MLP	Multilayer Perceptron
NN	Neural Network
PCM	Pulse Code Modulation
TDNN	Time Delay Neural Network

1 GİRİŞ

İnsanlar arasında kullanılan en önemli iletişim aracı sestir. İnsan-bilgisayar arasında da yazının ve simgelerin yanı sıra sese dayalı etkileşimin kurulabilmesi kullanım kolaylığı açısından önem taşımaktadır. Ancak sesli ifadenin yazılı ifade gibi değişmez karakterlerden değil de kişiden kişiye değişen fonemlerden oluşması tanınmasını zorlaştırmaktadır. Bu da sesli ifade tanımada farklı yöntemlerin kullanılmasını gerektirmektedir.

Sesli ifade tanıma insan-bilgisayar arası iletişimde çok önemli faydalar sağlamaktadır. Gelişen sesli ifade tanıma teknolojileri sayesinde bilgisayara veri girişi çok daha hızlı bir şekilde yapılabilmektedir. Veri girişinin yanında sesli ifade teknolojileri sayesinde bilgisayara sesli komut vermek artık mümkün olmuştur. Böylece hem yeni veri girişinde hem de sesli komut kullanılmasıyla hız kazanılmaktadır. Bu amaçlar dahilinde düşünülecek olursa sesli ifade tanıma sistemlerinin kullanım alanlarından biri olarak etkileşimli sesli cevap sistemlerini (*interactive voice response systems*) saymak mümkündür. Çoğu şirket, müşterilerinin ihtiyacını karşılamak için arama merkezlerini kullanır. Bu amaçla, çok pahalı bir yöntem olmasına rağmen, arama merkezlerinde çalışmak üzere çok sayıda çalışan kiralarlar. Eğer sesli cevap sistemlerinin kullanıcıları ile haberleşmek üzere, sesli ifade tanıma sistemlerinin geliştirilmesiyle birlikte çalışana olan ihtiyaç azalacak ve çalışanların yaptığı işi sesli ifade tanıma sistemleri devralacaktır.

Sesli ifade tanıma sistemleri, dikte sistemlerinin kullanılmasına da imkan verecektir. Böylece bir editör program dahilinde kullanılabilecek bir sesli ifade tanıma sistemi yardımıyla, klavye yerine insan sesi ile bir belge yazılması hem daha az zaman harcayarak hem de daha kolay bir şekilde mümkün olabilecektir. Aynı şekilde duyma engelliler için televizyon yayınları, veya iş esnasında el ve gözlerini kullanan çalışanların yapması gereken kayıtlar; sesli ifade tanıma sistemlerinin kullanılmasını gerektirir.

Sesli ifade tanıma sistemlerinin yaratacağı bir başka kolaylık ise, ses ile komut verilmesini sağlamaktır. Böylece, örneğin bir televizyon uzaktan kumanda yerine insan sesi ile kumanda edilebilecek, bir kanal adı söylenerek değiştirilebilecek veya televizyon önceden belirlenen bir komut ile kapatılabilecektir. Bu teknolojiyi

evin diğ er ara ları i in de kullanmak m mk nd r. Hatta arabalarda, robotlarda, bilgisayarlardaki elektronik aletlerin ses ile y netilmeleri m mk nd r.

Sesli ifade tanıma sistemlerinin, d nyadaki bir  ok dil i in uygulanması ile birlikte, konu ma   z mleyiciler (*speech synthesizer*) ve makine  eviriciler (*machine translator*) ile b t nle tirilerek  eviri i lemleri i in kullanılabilirlerdir.

Sayılan kullanım alanları dı ında  ok farklı ama lar i in sesli ifade tanıma sistemlerini kullanmak m mk nd r. Ve bu sistemlerin kullanılmasıyla bir ok alanda insanlar tarafından harcanan i  g c , para ve zaman b y k oranda azalacaktır.

Bu tezin 2. b l m nde sesli ifadeye ili kin tanımlar verilmi tir.  ncelikle sesli ifadeye ait  zelliklerden bahsedilmi , ardından sesli ifadeyi olu turan birimler hakkında farklı diller ve T rk e ba lamında bilgi verilmi tir.

3. b l mde sesli ifade tanıma sistemlerinin t rlerine ve bu sesli ifade tanıma sistemlerini birbirinden ayıran farklı  zelliklere de inilmi tir.

4. b l mde sesli ifade tanıma s recinde izlenen adımlar genel hatlarıyla a ıklanmı tır.  ncelikle sesli ifadenin sayısalla tırılmasıyla ba lanan s re , pencereleme, *Fourier* d n   m  gibi bir takım  n i lemlerin ardından sesli ifadenin  zellik vekt rlerine ayrılmasıyla s rd r lmektedir. Son olarak bu  zellik vekt rlerinin sınıflandırılması i lemine ge ilmektedir. Sınıflandırma s recinde kullanılabilecek y ntemler de bu b l mde anlatılmı tır.

5. b l mde geli tirilen hece-tabanlı T rk e sesli ifade tanıma sisteminin tanıtımı yapılmı  ve deneysel  alı mada izlenen adımlar anlatılmı tır. Son olarak yapılan deneylerin sonu larına ait istatistiksel bilgiler verilmi tir.

6. b l mde ise genel olarak tez  alı ması dahilinde yapılan deneylerden elde edilen bilgiler irdelenmi  ve gelecekte bu  alı malara nelerin eklenebilece i  zerinde durulmu tur.

2 SESLİ İFADE TANIMLARI

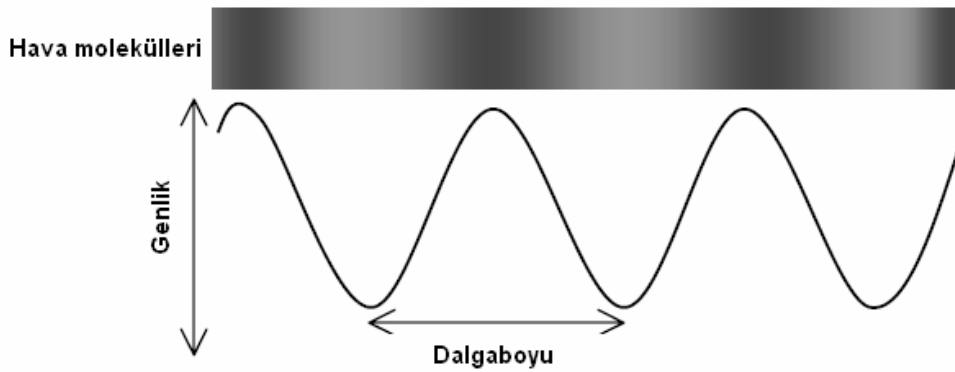
İnsanlar arasında iletişimi sağlayan en önemli araç, sestir. Sesli iletişim, yazılı iletişime göre daha kolay, daha hızlı olması sebebiyle daha fazla tercih edilen bir iletişim şekli olmaktadır.

Sesli iletişim kurmak isteyen birey tarafından öncelikli olarak beyinde tasarlanan düşünce, bir dil bağlamında sözcüklere dönüştürülerek, bir dizi sinirsel dönüştürümler sonucunda konuşma sistemi elemanları olan ağız, dil, dudak ve ses telleri aracılığıyla ses dalgaları biçiminde havaya yayılır ve iletişim kurulacak diğer birey tarafından algılanır, yorumlanır. (Somervuo, 1996)

Dilin ses yönünü inceleyen bilim dalı sesbilimdir (fonetik). Sesbilim, sesin ne gibi niteliklere sahip olduğunu, ses dalgalarıyla dinleyene nasıl ulaştığını, dinleyenin bu sesi nasıl algıladığını, kısacası dilin sesle ilgili olan tüm özelliklerini inceler. (Sever, 2000)

2.1 Sesin Fiziksel Özellikleri

Ses, gırtlaktaki ses tellerinin hava moleküllerini titreştirmesi sonucunda, bu hava moleküllerinin sıkışma ve dağılması ile enerjinin uygulandığı yöne paralel olarak boyuna basınç dalgaları oluşturması sonucunda meydana gelir (Şekil 2.1). (Huang, Acerd, Hon, 2001)



Şekil 2.1 Ses dalgasının havada yayılımı

Bu dalgalar bir sinüs dalgası ile ifade edilebilir. Bir sinüs dalgasını ifade edebilmek için, iki önemli parametre mevcuttur: genlik ve dalgaboyu. Frekans da (Hz) dalga

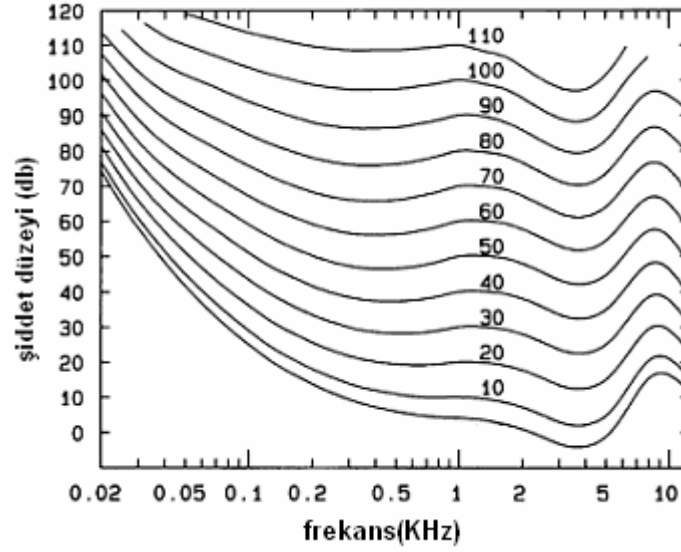
biçimi ölçmek için kullanılmaktadır. Havadaki ses basınç dalgasının hızı yaklaşık olarak $331.5 + 0.6T_c$ m/s şeklinde ifade edilir. Burada T_c *Celcius* cinsinden sıcaklık değeridir.

Psikoakustik'te, bir sesin özellikle de bir konuşma sesinin algılanabilen özellikleri ile o sesi karakterize eden ölçülebilen fiziksel özellikleri arasında bir ayrım yapılır. Algılanabilen bir özellik tek bir fiziksel niceliğe bağlı olmaktan çok, diğer fiziksel nicelikler de bu algılanabilen özelliği etkileyebilmektedir. Dolayısıyla aradaki bağlantı biraz karmaşıktır. Sesin algılanabilen ve fiziksel özellikleri arasındaki ilişki Çizelge 2.1.'de verilmiştir.

Fiziksel Nicelik	Algılanan Nitelik
Şiddet	Yükseklik
Temel frekans	Perde (<i>pitch</i>)
İzgesel şekil (<i>spectral shape</i>)	Tını
Başlangıç/bitiş zamanı	Zamanlama
İki kulaklı duymada faz farkı	Konum

Çizelge 2.1 Sesin algılanabilen ve fiziksel nicelikleri arasındaki ilişki

Şiddet seviyesi daha yüksek olan seslerin genelde daha yüksek ses seviyesine sahip olmalarıyla değişen kulak duyarlılığı, sesin frekansı ve kalitesi ile de değişir. Bu tanımlı doğrular nitelikte olan ve ISO tarafından kabul edilen eş yükseklik eğrileri kuralına göre de, aynı yüksekliğe sahip sesleri elde etmek için farklı frekans değerlerinde farklı şiddet seviyeleri uygulama zorunluluğu vardır ve aynı frekans değerinde bir sesin şiddeti arttıkça yüksekliği de artar. İnsan kulağına ilişkin Şekil 2.2'de eş yükseklik eğrilerine ait bir çizim verilmiştir:



Şekil 2.2 İnsan kulağına ilişkin olarak sesin eş yükseklik eğrileri

Şekil 2.2'de en alttaki eğri insan kulağının duyabildiği en düşük yüksekliğe sahip ses yükseklik eğrisi, en üstteki eğri ise insan kulağının duyabildiği en yüksek ses yüksekliğine sahip yükseklik eğrisidir. Duyma duyarlılığı dış kulak kanalının birinci rezonans frekansına yakın olan 4000 Hz civarında en yüksek değerine ulaşır, ve ikinci rezonans frekansı olan 13 kHz civarında tekrar yükselişe geçer.

Sesin perdesi (*pitch*), sesin temel frekansı ile yakından ilişkilidir. Temel frekans ne kadar yüksekse, elde edilen ses perdesi de o kadar yüksek olur. Frekans sabit tutulduğunda ve sesin şiddeti artırıldığında sesin perdesi de değişecektir.

Farklı insanlara ait sesler farklı olarak algılanır. Bu, ses organlarının fiziksel niceliklerinden etkilenen temel frekans farklılıkları gibi ayrımlardan kaynaklanır. Örneğin erkeklerde ses organlarının kütle ve uzunlukça kadınlara göre daha büyük olması temel frekans ve dolayısıyla algılanan ses olarak farklı duyulmalarını sağlar.

Sesin fiziksel özelliğine ait bir başka kavram ise, o sesin tınısıdır. Tını, aynı yüksekliğe ve perdeye sahip olan seslerin farklı algılanmasının nedenini açıklayan bir kavramdır. Özellikle müzikte sıkça kullanılan bir kavramdır. Örneğin aynı süre içerisinde aynı oktavda çalınan aynı notanın bir flütle ya da bir kemanla çalınması o sesin farklı algılanmasına neden olacaktır. Bir sesin tınısı, o sesin izgesel (*spectral*) güç dağılımı, genlik ya da frekans dağılımının modülasyon derinliği,

harmonikler arasındaki harmonik olmama derecesi gibi birtakım fiziksel değişkenlere bağlıdır. (Huang, Acerd, Hon, 2001)

2.2 Sesli İfadelerde Ses, Fon (Sesbirim), Fonem, Hece ve Sözcük

Sesbiliminin bir çalışma konusu, herhangi bir sesi üretirken ses organlarının konumlarını tanımlamaktır. Amaç, herhangi bir sese ait, bütün sesbilimciler tarafından tanınmayan herhangi bir sesle karşılaşıldığı zaman yazı ile ifade edilebilecek genel bir gösterim elde etmektir. Herhangi bir dile ait alfabede yer alabilecek, çok fazla farklı konuşma sesi vardır. Dolayısıyla sesbilimcilerin kendi gösterim sistemlerini tasarlaması gerekmektedir. En eski ve en çok kabul edilen gösterim Uluslararası Fonetik Alfabeti (*International Phonetic Alphabet – IPA*)'dir. Alfabe, Romen karakterleri başta olmak üzere diğer dillerin alfabelerinden alınmış harflerden oluşmaktadır. IPA'da yer alan çoğu sembol yazıcılar tarafından basılamadığı için, yakın zamanda "Arphabet" (İleri Araştırma Projeleri Birimi – ARPA'nın kurulması ile birlikte) geliştirilmiştir. Ancak yine de, IPA halen sözlüklerde kelimelerin telaffuz gösterimleri için kullanılmaktadır. (Parson, 1986)

Konuşma sinyalinin alışılan ayrılma yöntemi ünlüler ve ünsüzler şeklindedir. Bu terimler sesbiliminde kullanılmasına rağmen, bunları kesin olarak tanımlamak zordur. Tanımlamada gösterilen çabalar genelde belirli bir dil bağlamında sahip olduğu işlevlere ait soruları içerir.

Bir dile ait sesli ifadenin farklı özellikler dikkate alındığında farklı biçimlerde parçalara ayrılması mümkündür. İzleyen bölümlerde bu birimlerin neler olabileceği ve bu birimler hakkında genel bilgi verilecektir.

2.2.1 Ses (*phone*), fonem (*phoneme*)

Sesbilim (*phonetics*), herhangi bir dilden bağımsız olarak konuşma sinyallerinin bir görüntüsü iken, fonembilim (*phonemics*) belirli bir dile ait konuşma sinyallerinin bir görüntüsüdür. Sesbilim, dilbilimine kaynak sağlar; fonembilim ise dilbiliminin bir koludur.

Sesbilimde, en küçük birim ses (*phone*) olarak adlandırılırken, fonembilimde en küçük birim fonem (*phoneme*)'dir. Sesler köşeli parantezler içinde yazılırlar: [b], fonemler ise, eğik çizgiler arasında yazılırlar: /b/.

Fonemin çalışan bir tanımını vermek gerekirse, belli bir dilde bir kelimeden diğerine fark eden en küçük ses birimi olarak tanımlamak mümkündür. Herhangi bir sesli ifadede eğer bir ses biriminin değiştirilmesi o sesli ifadenin anlamını da değiştiriyor ise, bu durumda değiştirilen birim aynı zamanda fonemdir. Eğer o ses biriminin değiştirilmesi herhangi bir farklılık yaratmıyor, ya da yeni oluşan sesli ifade o dil bağlamında anlamsız oluyor ise bu durumda değiştirilen birim bir fonem değildir.

Seslerin sayısı, onları ayırt etme yeteneğimizle sınırlı olup çok geniştir, fonemlerin sayısı ise bütün dillerde küçüktür. Bilinen en büyük fonem sayısı 45 (Chipewyan dili), en küçük ise 13 (Havai dili)'tür.

Ses, dilin en yalın ögesidir. Sesli ifadeden yazıya geçişte, her sese karşılık bir alfabetik simge verilir. Ancak ses sayısının çok fazla olması, sesli ifadeden yazıya geçişi karmaşılaştırır. Bu nedenle kimi sesler kümelenerek her kümeye tek bir simge verme yolu kullanılmaktadır. Bu bağlamda seslerin kümelenmesi fonem tabanında gerçekleşir. Fonemler, anlam ayırıcı özelliği bulunan ses kümeleridir. Yani anlam ayırıcı özelliği bulunmayan sesler tek bir fonem kümesi altında toplanır. Ses, diller üstü bir birimdir. Fonem ise anlam ayırıcı özellik içermesi nedeniyle belli bir dile özgü birimdir. (Artuner, 1994)

Her dilin kendine özgü bir alfabesi bulunmaktadır. Bu bağlamda Türk Alfabesi fonemik bir alfabe olarak benimsenebilir. Türkçenin her fonemine bir alfabetik simge (harf) atandığı söylenebilir.

Sesler bir fonem kümesine dahil edilirken izlenen yol, anlam ayırımına dayanmaktadır. Eğer bir sözcük içerisinde yer alan bir ses, başka bir sesle yer değiştirdiğinde bir anlam farklılığı oluyor ise, bu sesler farklı fonemlere aittir. Örneğin, *kaZak* ile *kaYak* örneğinde değiştirilen ses ile sözcüğün anlamı da değiştiği için, z ve y seslerinin farklı fonem kümelerinde yer aldıkları söylenebilir.

Ses, fonemlerin seslendirilmesiyle oluşur. Konuşma organlarının, sesli ifadeler oluşturmak için çalışması sonucunda çıkan birimlerdir. Ters bir tanım da yapılacak olursa, sesbirimlerin simgesel olarak ifade edilmesi sonucu oluşan simgeler fonem olarak adlandırılır.

Sesbirimlerin tanımlanabilmesi için ayırıcı özellikler (*distinctive features*) kuramı geliştirilmiştir (Jakobsan, Fant, Halle, 1951). Ayırıcı özellikler bir fonemin sahip olabileceği 12 tane özellikten oluşmaktadır. Her özellik akustik bir nitelik taşımakta ve olası iki değerden yalnız bir tanesini alabilmektedir. Ayrıca her özellik bir diğerinden bağımsızdır. Böylece her fonem ayırıcı özelliklerin oluşturduğu bir küme olarak düşünülebilmekte ve *boolean* bir vektörle ya da 12 bitlik ikili bir sayı ile tanımlanabilmektedir. Jakobson, Fant ve Halle tarafından belirlenen özellikler şu şekildedir (1952,1956):

- Sessellik (*Vocalic/Nonvocalic*): Tanımlı bir formant yapısının var olup olmamasına karşılık gelir.
- Ünlü/Ünsüz (*Consonantal/Nonconsonantal*): Ünlü özelliğindeki fonemler görece olarak daha küçük bir toplam enerjiye sahiptir.
- Sıkışık/Seyrek (*Compact/Diffuse*): İzgesel (*spectral*) enerjinin dağılımına karşılık gelir.
- Gergin/Gevşek(*Tense/Lax*): Gergin, daha geniş bir bant genişliği ve uzun bir sürede daha yüksek bir toplam enerjiye karşılık gelir.
- Ötümlü/Ötümsüz (*Voiced/Voiceless*): Ses tellerindeki titreşimlere bağlı olarak düşük frekanstaki bileşenlerin varlığına dair bir özelliktir. Ötümlü bir fonemde bu bileşenler bulunur.
- Genizden/Ağızdan (*Nasal/Oral*): Genizden meydana getirilen fonemler fazladan rezonansların eklenmesi ile daha geniş bir izgesel enerji dağılımı gösterir.
- Kesilmeli/Sürekli (*Discontinuous/Continuous*): Kesilmeli fonemler izgesel enerji yayılımında ani enerji değişiklikleri gösterir.
- Keskin/Yumuşak (*Strident/Mellow*): Keskin fonemler daha güçlü ve daha fazla gürültü bileşenleri içerir.
- Patlamalı/Patlamasız (*Checked/Unchecked*): Patlamalı fonemlerde enerji, patlama şeklinde görülür.

- Bas/Tiz (Grave/Acute): Bas seslerde düşük frekanslı rezonanslar hakim iken, tiz seslerde yüksek frekanslı rezonanslar hakimdir.
- *Flat/Plain*: Yüksek frekanslı rezonansların göreceli enerjilerine göre farklılık göstermektedir: *Flat* daha zayıf, *plain* ise daha güçlü özellikteki fonemleri tanımlar.
- Keskin/Düz(Sharp/Plain): Keskin fonemler, daha yüksek frekanslı rezonansların göreceli frekanslarında bir yükselme gösterirler.

Bu özellikler 4096 fonemi tanımlayabilmektedir. Bu nedenle, bu özelliklerden elde edilen her kombinasyonun herhangi bir dilde mutlaka yer alması beklenmemelidir.

Sesbirim kavramı genişletilerek parçalı sesbirimler ve parçalar üstü sesbirimler olarak ikiye ayrılır (Ergenç, 1989). Parçalı sesbirimleri, ünlü, ünsüz ve kayan ünlüleri içerir. Bu sesbirimler saptanırken anlam ayırıcı özelliklerinden faydalanılmaktadır. Bunun dışında süre, perde değişimi, ton, vurgu, ezgi gibi ayırıcılardan da faydalanılması ile meydana gelen sesbirimlere, parçalar üstü sesbirimler adı verilir. Ve parçalar üstü sesbirimleri sesli ifadenin bir kesimine bakarak belirlenemez, genellikle bağlama bağımlıdır.

2.2.2 Hece

Konuşma, yukarıda bahsedilen birim konuşma seslerinin ardı ardına sıralanması ile meydana gelir. Fonemler bir araya gelerek heceleri meydana getirir. Bu kural bütün dillerde geçerlidir. Ancak heceler farklı dillerde farklı biçimlerde olabildiği için sesbilimciler heceleri tanımlamakta zorluk çekmişlerdir. Diller için heceler, yapıları itibari ile farklılık göstermektedir. Bu yüzden tüm dillere özgü ortak bir hece tanımı çıkarılması mümkün olmamaktadır.

Türkçede her hecede mutlaka bir ünlü bulunur, ünlü olmadan hece kurulamaz. Sesli ifadeler, kaburgalar arası kaslarca biçimlendirilen göğüs atışları neticesinde oluşurlar. Her göğüs atışıyla birlikte ses telleri de titreşmeye başlar. Bu şekilde ünlü sesin çıkarılması gerçekleşir. Her hecede de tek bir ünlü ses olabileceği için, her göğüs atışı periyodunda çıkarılan sesler hece olarak anılırlar. Her atış sürecinde belirli basınçta soluk, ses yolundan dışarı çıkar. Bu basınçlı havaya ses tellerinin titreşimlerinin eklenmesiyle ünlü sesler çıkarılır. Kasların gevşemesiyle

birlikte geçen havanın azalması ya da durdurulması sırasında ses tellerinin titreşmesi ya da durması ünsüz seslerin çıkarılmalarını sağlar. (Artuner, 1994)

Heceler, sözcüklerin ses yapısını oluştururlar. Hece yapısının Türkçe için basit kurallar ile ifade edilebilmesi bu tez çalışması açısından önem taşımaktadır. Türkçede ünlüler tek başlarına hece özelliği gösterdikleri halde ünsüzler yanlarına ünlü almadan bir hece oluşturamazlar. Dolayısıyla Türkçe bir sözcükte kaç ünlü varsa, o kadar da hece var demektir. Çünkü, Türkçe bir hecede birden fazla ünlünün bulunması mümkün değildir. Ünsüzler, kendilerini takip eden ünlülerle birleşerek hece oluştururlar. Bu sebeple bir sözcük hecelerine ayrılırken ünsüz+ünlü şeklinde hecelenir. Bu durum, yan yana iki ünsüzün gelmediği durumlarda geçerlidir.

Günümüz Türkçesinde kullanılan 11 tür hece vardır. Bunlar Çizelge 2.2.'de belirtildiği gibidir:

1	Ü	<i>a-na</i>
2	ÜZ	<i>as-ker</i>
3	ZÜ	<i>ki-tap</i>
4	ÜZZ	<i>alt</i>
5	ZÜZ	<i>ka-sap</i>
6	ZÜZZ	<i>tank</i>
7	ZZÜZ	<i>spor</i>
8	ZZÜ	<i>kli-ma</i>
9	ZÜZZZ	<i>tekst</i>
10	ZZÜZZ	<i>flört</i>
11	ZZZÜZ	<i>stres</i>

Çizelge 2.2 Türkçede hece türleri (yabancı dillerden dilimize giren sözcüklerin içerdiği heceler de dahil)

Burada ilk 6 hece türü Türkçe sözcüklerde kullanılan hece türleridir. Diğerleri ise yabancı dillerden Türkçeye geçmiş sözcüklerde bulunan hece türleridir. (Demircan, 1979)

2.2.3 Sözcük

Hecelerin birleşmesiyle sözcükler oluşur. Türkçe, eklemeli bir dil olduğu için heceleri yan yana getirmek sözcük oluşturmak için yeterlidir. Ancak bazı durumlarda heceler yan yana gelirken değişikliğe uğrayabilir.

Türkçede hecelerin sözcük oluşturmaya dair bazı kuralları mevcuttur. Türkçede büyük ünlü uyumu, küçük ünlü gibi ses uyumlarına uymak şartıyla heceler yan yana gelebilir. Ancak yabancı dillerden dilimize girmiş sözcükler bu kurallara uymamaktadır.

Büyük ünlü uyumu olarak adlandırılan ses uyumu, bir sözcük içinde kalın ünlülerden sonra kalın ünlülerin, ince ünlülerden sonra ise ince ünlülerin gelmesi kuralını gerektirir. Küçük ünlü uyumu olarak adlandırılan ses uyumu ise, bir sözcüğün ilk hecesinde eğer düz ünlü var (a,e,ı,i) ise sonraki hecelerde de düz ünlü bulunması kuralını, eğer bir sözcüğün ilk hecesinde yuvarlak ünlü var ise bunu izleyen ilk hecede dar yuvarlak (u,ü) ya da düz (a,e) ünlü bulunması kuralını gerektirir. Ünlülere dair özellikler, Türkçenin parçalı sesbirimleri kapsamında ilerleyen bölümde anlatılacaktır.

Türkçe sesli ifade tanıma sistemlerinde bu kurallardan faydalanarak sesli ifade tanıma başarımını yükseltmek mümkündür.

2.3 Türkçenin Parçalı Sesbirimleri

2.3.1 Ünlüler

Ünlüler, ses telleriyle oluşturulan titreşimlerin ses yolunda rezonansa sokulması yoluyla elde edilen seslerdir. Ses yolunun herhangi bir konumunda bir engele uğramadan oluşurlar. Türkçede ünlü olarak nitelendirilen 16 sese karşılık, anlam ayırıcı olarak 8 tane sesbirim tanımlıdır.

/a/, /e/, /ı/, /i/, /u/, /ü/, /o/, /ö/

Ünlüler, çene açısına (dil-damak arası açıklık), dilin ağız içindeki konumuna ve dudakların biçimine göre sınıflandırılırlar.

a) Ünlülerin çene açısına göre sınıflandırılması (dil-damak arası açıklık):

Çene açısına göre, yani ağız açıklığına göre ünlüler dar veya geniş olarak belirlenir.

Dar ünlüler : /ı/, /i/, /u/, /ü/

Geniş ünlüler : /a/, /e/, /o/, /ö/

b) Ünlülerin dudakların biçimine göre sınıflandırılması:

Dudakların biçimine göre ünlüler düz ve yuvarlak olmak üzere sınıflandırılırlar.

Düz ünlüler : /a/, /e/, /ı/, /i/

Yuvarlak ünlüler : /o/, /ö/, /u/, /ü/

c) Ünlülerin dilin ağız içindeki konumuna göre sınıflandırılması:

Bu sınıflandırma biçiminde, diğerlerinden farklı olarak ünlü ses çıkarılırken ağzın aldığı biçim değil sesin çıkarılma noktası dikkate alınır. Dil, sesin çıkarılmasında önemli bir rol oynar. Dilin bölgelerine göre ünlüler; arka-dil, orta-dil ve ön-dil olmak üzere sınıflandırılırlar.

Arka-dil ünlüleri : /a/, /o/, /u/

Orta-dil ünlüleri : /ı/

Ön-dil ünlüleri : /e/, /i/, /ö/, /ü/

Ön-dil ünlüleri ise, sesin çıkarılırken aldığı biçime göre düz ya da yuvarlak olarak sınıflandırılabilirler.

Düz ön-dil ünlüleri : /e/, /i/

Yuvarlak ön-dil ünlüleri : /ö/, /ü/

2.3.2 Ünsüzler

Ses yolunun herhangi bir konumunda bir engele uğrayarak çıkan sesler ünsüz olarak adlandırılırlar. Ünsüzlerde ses teli titreşimleri önemli değildir, bu sesleri oluşturan daha çok sesteki kesilmelerdir.

Türkçede 20 tane ünsüz bulunmaktadır:

/b/, /c/, /ç/, /d/, /f/, /g/, /h/, /j/, /k/, /l/, /m/, /n/, /p/, /r/, /s/, /ş/, /t/, /v/, /y/, /z/

Türkçede ünsüzler çıkış noktasına, çıkış biçimine ve ses teli titreşimlerinin varlık ve yokluğuna göre sınıflandırılırlar.

a) Ünsüzlerin çıkış biçimlerine göre sınıflandırılması:

Havanın ses organlarından çıkış biçimi sesi etkiler. Buna göre ünsüzler patlamalı, çarpmalı, yan daralmalı, sızıcı ve genizden olmak üzere 5 sınıfa ayrılır.

Patlamalı ünsüzler : /b/, /d/, /g/, /p/, /t/, /k/

Çarpmalı ünsüzler : /r/

Yan daralmalı ünsüzler : /l/

Sızıcı ünsüzler : /c/, /ç/, /f/, /h/, /j/, /s/, /ş/, /v/, /y/, /z/

Geniz ünsüzleri : /m/, /n/

b) Ünsüzlerin çıkış yerlerine göre sınıflandırılması:

Havanın çıkış noktası da sesi etkiler. Buna göre ünsüzler çift dudak, dudak-diş, dilucu-dişardı, dilucu-dişeti, dil-öndamak, dilucu-öndamak, dil-artdamak ve gırtlak ünsüzleri olmak üzere 8 sınıfa ayrılırlar.

Çift dudak : /b/, /p/, /m/

Dudak-diş : /f/, /v/

Dilucu-dişardı : /d/, /t/

Dilucu-dişeti : /n/, /r/, /s/, /z/

Dil-öndamak : /c/, /ç/, /j/, /ş/, /y/

Dilucu-öndamak : /l/

Dil-artdamak : /k/, /g/

Grtlak : /h/

c) Ünsüzlerin ses tellerinin titreşimine göre sınıflandırılması:

Bazı ünsüz sesler meydana gelirken ses tellerinde titreşim oluşur, bu tür ünsüz seslere ötümlü adı verilir. Eğer ünsüz ses çıkarılırken ses tellerinde titreşim meydana gelmiyorsa böyle ünsüz seslere de ötümsüz adı verilir. (Ergenç, 1990)

Ötümlü ünsüzler : /b/, /c/, /d/, /g/, /j/, /l/, /m/, /n/, /r/, /v/, /y/, /z/

Ötümsüz ünsüzler : /p/, /t/, /k/, /ç/, /f/, /s/, /ş/, /h/

Burada sesbilim açısından önemli bir nokta da şudur ki; dünya dillerinde belli sayıda ünlü, ünsüz bulunsada bunların nitelikleri farklıdır. Örneğin Türkçedeki /a/ sesi ile, İngilizcedeki, Fransızcadaki, Farsçadaki ve Japoncadaki /a/ sesleri birbirinden farklıdır. Bu /a/ seslerinden bazıları /o/ya yakın, kimi genizden gelen kimi de Türkçedeki /a/ sesinden daha uzundur.

3 SESLİ İFADE TANIMAYA GENEL BAKIŞ

Sesli ifade tanıma sistemlerinde tanınacak en küçük birim, tanıma sistemlerini birbirinden ayıran özelliklerden biri olmaktadır. Sistemde tanınması amaçlanan dilin de özellikleri göz önünde bulundurularak birime ait özellikler belirlenmelidir. Örneğin Türkçenin çekimli bir dil olması, sözcük sayısını sınırsız yaptığı için Türkçe sesli ifade tanıma sistemlerinde tanınan birimin sözcük olmasını imkansızlaştırmaktadır. Bu yüzden şu ana kadar Türkçe için geliştirilen sesli ifade tanıma sistemlerinin büyük bir kısmı fonem tabanlı olmuştur. Bu tür tanıma sistemlerinde sesli ifadeyi oluşturan fonemlerin tek tek tanınması, daha sonra bunların birleştirilmesi ile sesli ifade tanınması sağlanır. Ancak birimlerin boyutunun küçük olması birbirlerinden ayırt edilmesini zorlaştırmakta, ve sistem başarımını olumsuz yönde etkilemektedir. Aynı zamanda fonemlerin birbirini etkilemesi de (*coarticulation*) sistem başarımını olumsuz yönde etkileyen bir diğer faktördür. Bu yüzden bu tez kapsamında fonemlerden meydana gelen ve daha büyük birimler olan hecelerin sesli ifade tanıma sistemi dahilinde tanınması amaçlanmıştır. Türkçede hece yapısının tek ünlü içermek gibi basit kurallardan oluşması, hece tabanlı tanımayı kolaylaştırmıştır.

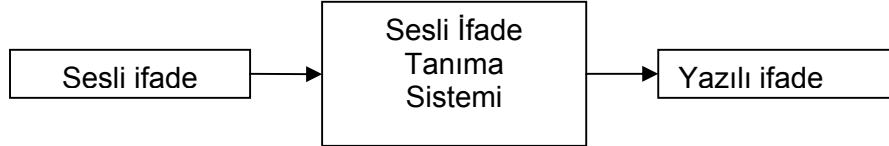
Sesli ifade tanıma sistemlerini birbirinden ayıran bir diğer özellik de sınıflandırmada kullanılan yöntemlerdir. Çok katmalı nöron ağları sesli ifade tanıma sistemlerinin tanınacak birimlerin sınıflandırılma sürecinde çok yaygın olarak kullanılmaktadır. Bu tez kapsamında da hecelerin sınıflandırılmasında farklı özellikteki çok katmanlı nöron ağları kullanılmıştır.

Bu tez kapsamında geliştirilen hece-tabanlı sesli ifade tanıma sisteminin gerçekleştirimi iki farklı veri kümesi için denenmiştir. İlk olarak sürekli sesli ifadeler içinde önceden sisteme tanıtılan hecelerin tanınması sağlanmış, ardından sürekli sesli ifadeler içinde geçen hecelerin sınıflandırılması için bir algoritma *Rabiner* ve *Sambur*'un sınır bulma algoritması (Rabiner, Sambur, 1974) kullanılarak gerçekleştirilmiş ve sınırları bilinen hecelerin sesli ifade tanıma sistemi tarafından tanınması sağlanmıştır.

Sesli ifadenin üzerinde işlem yapmak güçtür, ve bu sesli ifadenin daha sonradan kullanabilmek, üzerinde farklı işlemler yapmak, saklamak, daha sonra kolayca ulaşabilmek için yazılı ifadeye dönüştürülmesi gerekir. Bu noktada bir sesli ifade

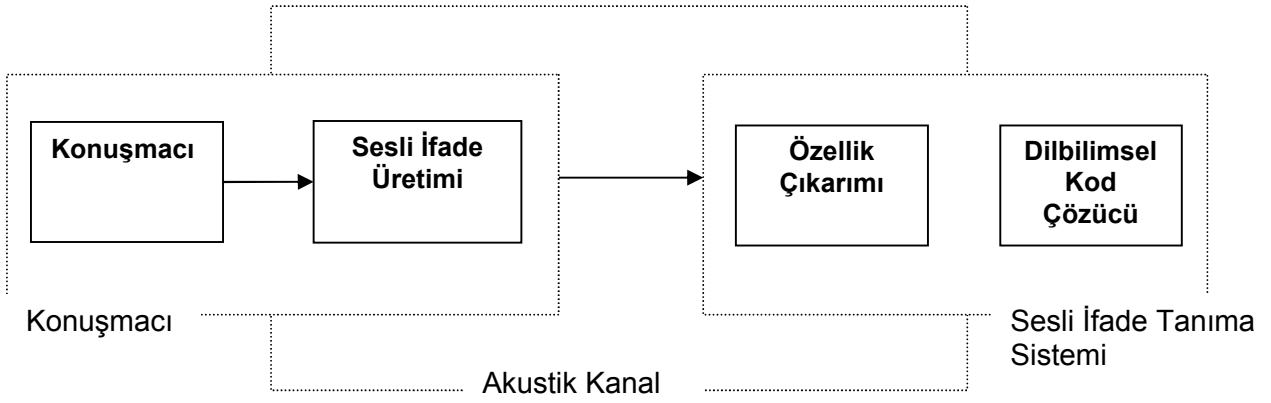
tanıma sistemi devreye girmektedir. Sesli ifade tanıma sisteminin amacı, insan-bilgisayar arası iletişimlerde insan-insan arası iletişimlerde olduğu gibi dinleyen konumunda bulunan insanın yerini almaktadır.

Bu durumda bir sesli ifade tanıma sisteminin görevi, verilen sesli ifadeyi yazılı ifadeye dönüştürmektir (Şekil 3.1).



Şekil 3.1 Sesli İfade Tanıma Sisteminin İşlevi

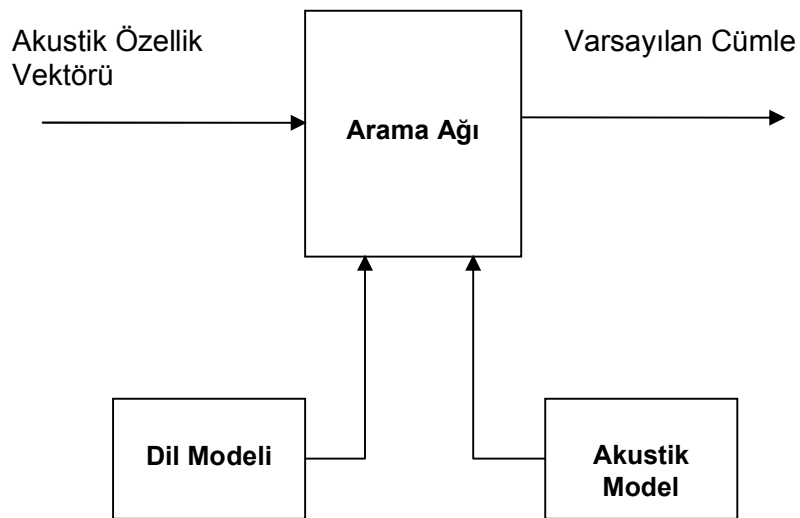
Bu süreçte öncelikli olarak sesin sayısallaştırılması gereklidir. Herhangi bir mikrofona sayısallaştırma işlemi gerçekleştirilmektedir. Sayısallaştırılan ses verisine ait sesli ifade sinyali birtakım ön işlemlere tabi tutulur. Sesli ifade sinyalinin tamamının üstünde işlem yapmak yerine sinyalin ilgili kısmını temsil edebilecek, o parçaya ait özellikleri içeren bir özellik vektörü kullanılmaktadır. Bu noktada sesli ifade sinyalinin hangi birim tabanında çalıştığı önem kazanır. Fonem tabanlı bir sesli ifade tanıma sistemi, sesli ifadenin yapı taşları olarak fonemleri kabul eder ve bu fonemler üstünden tanımayı gerçekleştirir. Bunun haricinde sözcük tabanlı, hece tabanlı olarak tanıma yapmak da mümkündür. Hangi birim üstünden tanıma yapılıyor ise, o birimin bulunması hedeflenen birim kümesindeki elemanlardan hangisine ait olduğuna dair bir sınıflandırma işlemi yapılacaktır. Bu durumda, o birime ait sinyal parçasının tamamını karşılaştırmak gereksiz ve zor bir yöntemdir ve özellik vektörü sayesinde sadece ilgili sinyal parçasını temsil eden belli sayıda özelliklerin karşılaştırılması yeterli olacaktır. Özellik vektörü çıkarırken dikkat edilecek husus, veri kaybına neden olmadan ilgili kısmı eksiksiz bir biçimde temsil edebilecek özelliklerin bulunmasıdır. Ve son adım olarak karşılaştırma işleminin ardından ilgili sesli ifade, dil simgelerine dönüştürülür. Kaynak-kanal modeli olarak adlandırılan ilgili işlem dizisi Şekil 3.2'de verilmiştir.



Şekil 3.2 Sesli ifadeye ait kaynak-kanal modeli

Bu haberleşme kanalının kaynağı, telaffuz edilecek kelimeler üstünde karar veren konuşmacı beynidir. Sesli ifade, konuşmacının ses yolunu da içeren karmaşık bir sistem aracılığıyla üretilir. Alıcı tarafında, bu sesli ifadeye ait dalga biçimi özellik çıkarıcı tarafından akustik özellik dizisine dönüştürülür. Akustik kanala ait şekil Şekil 3.2'de verilmiştir. Akustik özellik dizisi, dilbilimsel kod çözücüde (*linguistic decoder*) varsayılan sözcük dizisiyle eşleştirilir. Şekilde görüldüğü üzere, bir sesli ifade tanıma sistemi; özellik çıkarıcı ve dilbilimsel kod çözücüden oluşur.

Dilbilimsel kod çözücü üç önemli bileşen içerir: Akustik model, dil modeli, arama algoritması (kod çözme algoritması). Çözücüdeki işlem Şekil 3.3'te özetlenmiştir.



Şekil 3.3 Dilbilimsel Kod Çözücü

Kod çözücü, akustik modelden ve dil modelinden elde edilen olasılıkları kullanarak arama ağındaki en iyi yolu bulur. Bunun için öncelikle sesli ifade dalga biçimlerinin akustik özellik vektörlerine dönüştürülmesi gerekmektedir.

3.1 Sesli İfade Tanıma Sistemleri

Sesli ifade tanıma sistemleri farklı özellikler dikkate alındığında farklı şekillerde sınıflandırılabilir. Sesli ifadenin sürekli olup olmamasına göre sesli ifade tanıma sistemlerini şu şekilde sınıflandırmak mümkündür:

- **Ayrışık sözcük tanıma sistemleri (*Isolated word recognition*):** Bu yöntemde sözcükler birbirinden bağımsız olarak dikkate alındığından, birlikte seslendirme (*coarticulation*) sorunu bulunmamaktadır. Sözcükler kısa aralıklarla seslendirildiği için sözcüklerin başlangıç ve bitiş noktaları belirgindir ve bu durum tanıma işleminin daha kolay yapılmasına olanak sağlar. Zira sesli ifade tanımada en zor olanı sözcüklerin başlangıç ve bitiş noktalarının belirlenmesidir.
- **Sözcük yakalama sistemleri (*Word spotting systems*):** Bu sistemler, sürekli bir sesli ifadenin içinde belli birtakım sözcüklerin yakalanmasını sağlarlar. Bu sistemlerde *time warping* ile dinamik programlama teknikleri sıkça kullanılır, ve her sözcük bir şablon (*template*) ile ifade edilir. Tanıma işlemi, aranan şablonun sesli ifade içinde çakıştığı bir örüntü arama biçiminde gerçekleşmektedir.
- **Sürekli sesli ifade tanıma sistemleri (*Continuous speech recognition systems*):** Bu sistemler, ara verilmeden seslendirilen sözcüklerin tanınmasını amaçlar. Tanıma işlemi, sözcük sınırları bulunup sözcük tanımaya indirgenebilir veya sözcüğe göre daha alt düzey birimler üzerinden gerçekleştirilebilir. Sürekli sesli ifade tanıma, ayrışık sözcük tanıma ve sözcük yakalamaya göre daha zordur. Bunun birinci nedeni sözcük sınırlarının belli olmamasıdır. Sürekli sesli ifadede sözcük sınırlarının bulunması zor, hatta kimi zaman olanaksızdır. Sözcük başına gelen seslerin bazen kaybolması buna neden olarak gösterilebilir. Sürekli sesli ifade tanımayı zorlaştıran bir diğer neden de sesbirimlerinin, birlikte seslendirmeden dolayı (*coarticulation effects*) kendisinden önce ve sonra

gelen diğer sesbirimlerinden etkilenmeleridir. Son neden olarak da, vurgulama, duraklama gibi bürünlerden (*prosodics*) kaynaklanır. İsim, sıfat gibi bazı sözcükler yüksek enerjili seslerle ifade edilirken, bağlaçlar ve kısa süreli sözcükler çoğu kez düşük enerjili olmakta ya da yutulmaktadır. Bu da, sistemin sesli ifadeleri doğru bir biçimde yakalayabilmesine engel olmaktadır. Sözcük tabanlı yapılan sesli ifade tanımda, sözcük sayısı oldukça geniş tutulmalıdır. Bu durumdan kurtulmak için, sesli ifade tanımda, sözcük yerine sesbirim, hece gibi daha alt düzeydeki birimler üzerinden tanıma yapılabilir. Bu alt birimlerin sayısı sözcük sayısına göre daha az olmasına rağmen, burada da sesbirim sınırlarının belirlenmesi, birlikte söylenme etkilerinin göz önüne alınması gibi sorunlar ortaya çıkabilmektedir. Sesbirim tabanlı tanıma sistemlerinde bu sesbirimlerinin tanınması ve daha sonra bunların birleştirilerek sözcüklerin oluşturulması sağlanmaktadır.

Sesli ifade tanıma sistemlerinin sınıflandırılmasında ele alınabilecek bir başka kıstas ise konuşmacıya bağımlı olup olmamasıdır. Bu durumda sesli ifade tanıma sistemlerini ikiye ayırmak mümkündür:

- Kişiyeye bağımlı sesli ifade tanıma sistemleri (*Speaker dependent*)
- Kişiden bağımsız sesli ifade tanıma sistemleri (*Speaker independent*)

Sistemin kullanılış alanını artırmak için amaç, kişiden bağımsız bir sesli ifade tanıma sistemi olmalıdır. Fakat bunu başarmak kişiyeye bağımlı bir sistem geliştirmekten daha zordur.

3.2 Sesli İfade Tanımaya İlişkin Özellikler

Sesli ifade tanıma sistemleri konuşmacıdan bağımsızlık, ses sinyal niteliği, öğrenme yeteneği, sözlük büyüklüğü, dilbilgisi kullanımı gibi özelliklere sahiptir. Bu özellikler aşağıda özetlenmiştir:

3.2.1 Konuşmacıdan Bağımsızlık

Eğer bir sesli ifade tanıma sistemi rasgele konuşmacılara ait sesli ifadeleri tanıyabiliyorsa konuşmacıdan bağımsızlık özelliğinden söz edilir. Genelde sesli ifadeler konuşmacıya bağımlı öğeler içerdiğinden, konuşmacıdan bağımsızlık

sağlanması zor ancak önemli bir özelliktir. Konuşmacıdan konuşmacıya farklılık gösteren bu öğelerin, tanımayı etkilemeyecek biçimde ortadan kaldırılması konuşmacıdan bağımsızlığı sağlamaktadır. Sesli ifade tanıma sistemlerinde denenen konuşmacı sayısı önemli bir parametre olup tek, sınırlı ve sınırsız biçimde belirlenmektedir.

Konuşmacıdan bağımsızlığı sağlamak için genelde kullanılan 3 değişik yaklaşım bulunmaktadır. Birinci yaklaşımda, sesli ifadelerdeki konuşmacıya bağımlı öğeleri ortadan kaldıran *anlayışlı ön işleme birimi* kullanılmaktadır. Bu yaklaşımla konuşmacıdan bağımsızlığın sağlanabilmesi için sistemde *uzman spektrogram* kesimi yer almalı ve bu spektrogramlar yüksek duyarlılıkta incelenmelidir. Amaç, spektrogramlar üzerinde değişmez ortak parametreler saptamaktır. Eğer bu tür parametreler bulunabilirse konuşmacıdan bağımsız tanıma gerçekleştirilmiş olur ve konuşmacıya bağımlı tanıma kadar kolaylaşır.

Konuşmacıdan bağımsızlığı sağlamak için kullanılan ikinci yaklaşım, konuşmacılar arasındaki farklılıkları yakalayacak gösterim biçimleri kullanmaktır. Bu şekilde, konuşmacıların ses özelliklerinin kümelenmesi gerçekleştirilir. Uygulamada sistem sözlüğünde bulunan her sözcük birden fazla konuşmacı tarafından seslendirilmektedir.

Üçüncü yaklaşım ise konuşmacıya uyum sağlama yönteminin kullanılmasıyla gerçekleştirilir. Bu yönetime göre, daha önceden belirlenen bir takım parametreler yeni konuşmacıdan gelenlerle karşılaştırılarak aradaki farklar bulunur. Böylece sistemde her yeni konuşmacıyla birlikte sözcük parametreleri de değişmektedir. Sonuçta örnekler kümelere ayrılıp her küme için bir örnek küme vektörü üretilir. Ancak sözcük ve konuşmacı sayısının artması durumunda, karmaşıklığın artması ve sistemin genel başarımının düşmesinden bahsetmek mümkündür. Bu yüzden bu yöntemin kullanıldığı sistemlerde sistem sözlüğünü yeterince küçük tutmakta, konuşmacılar arasındaki farkların kolayca anlaşılması açısından fayda vardır.

Kullanılan yaklaşım ne olursa olsun, konuşmacıdan bağımsızlığı sağlamada sistemde alıştırmalar (*training*) sürecine gerek duyulur. Alıştırma süreci, sözcük tanıma alıştırmalarının çok sayıda yinelenmesiyle gerçekleştirilir.

3.2.2 Ses Sinyalinin Niteliği

Sesli ifadelerin kaydedildiği ortam ve kayıt koşulları, sesli ifadenin niteliğini dolayısıyla sesli ifade tanıma sürecini etkileyen önemli özelliklerdir. Kayıt için uygun olmayan, rasgele seçilmiş ortamlarda kaydedilen sesli ifadeleri tanımak, ortamdan kaynaklanan gürültüden dolayı zordur. Bu nedenle, kayıt ortamı olarak özel yalıtılmış, yansız bir oda seçilmelidir.

Kayıt esnasında kullanılan araçlar da, kaydedilen sesli ifadenin niteliğini belirlemektedir. Bu amaçla yüksek kaliteli elektronik donanımlar kullanılabileceği gibi telefon gibi özel amaçlı ve koşulların zorlandığı donanımlar da amaçlarına göre kullanılabilmektedir.

3.2.3 Alıştırma Gereği

Sesli ifade tanıma sistemlerinde alıştırma (*training*) sürecine ihtiyaç duyulup duyulmaması da bu sistemlerin özelliğini belirleyen bir başka kriterdir. Bu özellik dikkate alındığında sesli ifade tanıma sistemlerini şu şekilde sınıflandırmak mümkündür:

- Alıştırma süreci gerektirmeyenler (*without training*)
- Alıştırmanın başta bir kez uygulandığı sistemler (*fixed training*)
- Alıştırma sürecinin tanıma süreciyle iç içe olduğu ve yeni koşullara uyum sağlayan sistemler (*continuous training*)

3.2.4 Sözlük Büyüklüğü

Sesli ifade tanıma sistemlerinin tanıyabildiği ayrışık sözcük sayısı, sistemleri birbirinden ayıran bir başka özelliktir. Sesli ifade tanıma sistemi tarafından tanınabilen sözcüklerin oluşturduğu küme sistem sözlüğü ya da kısaca sözlük olarak adlandırılır. Sözlüğün büyümesi genelde başarıyı olumsuz yönde etkilemekte, aynı zamanda işlem gücünü ve hata oranını da artırmaktadır.

Sözlük büyüklüğüne göre sistemleri şu şekilde sınıflandırmak mümkündür:

- Küçük boy (100'den az sözcük içeren)
- Orta boy (100-1000 arası sözcük içeren)

- Büyük boy (1000'den daha fazla sözcük içeren)
- Çok büyük boy (10000'den daha fazla sözcük içeren)

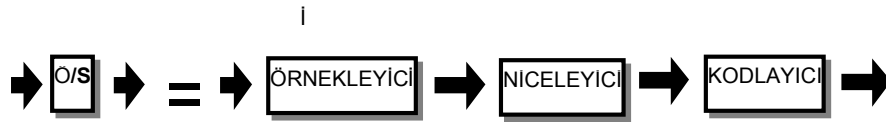
4 SESLİ İFADE TANIMA SÜRECİ

Sesli ifade tanıma süreci, kaydedilen örneksel sesli ifadenin sayısallaştırılmasıyla başlar. Sayısallaştırılan sesli ifade sinyali üzerinde bir takım ön işlemler yapıldıktan sonra sesli ifade sinyali özellik vektörlerine ayrılır ve tanımaya hazır hale getirilir. Özellik vektörlerine ayrılan sesli ifade sinyali son olarak bir sınıflandırılmaya tabi tutulur. Bu adım geleneksel sesli ifade sürecinin son adımıdır.

4.1 Sesli İfadelerin Sayısallaştırılması

Ses sinyalinin bilgisayar ortamına aktarılması için öncelikle bu sinyalin örnekselden sayısala dönüştürülmesi gerekir. Bu dönüşüm yoğun tımlleşik çevrim teknolojisini kullanan birimlerce sorunsuz bir şekilde yapılabilmektedir. Sesli ifadelerin sayısallaştırılmasında, örnekselden sayısala dönüşüm hızı ve bit türünden kod uzunluğu iki önemli kriterdir. Sesli ifade tanıma sistemlerinde örnekleme hızı genelde en az 8KHz ve üzerinde, sözcük genişliği de genellikle 12 bit ve üzerinde olmaktadır.

Örnekselden sayısala dönüşüm, örnekleme (*sample and hold*), niceleme (*quantization*) ve kodlama (*coding*) olmak üzere üç aşamada yapılır (Şekil 4.1).



Şekil 4.1 Örneksel/Sayısal Dönüştürücünün iç yapısı

Örnekleme, belirli zaman aralıklarında sinyal örneği alma işlemidir. *Nyquist* ilkesine göre belirlenen örnekleme sıklığı, örneklenen sinyal içindeki en büyük sıklığın en az iki katı olmalıdır. Aksi halde *spektrum örtüşmesi (aliasing)* yüzünden bozulmalar meydana gelir ve orijinal sinyal tekrar elde edilemez.

Niceleme, bir sinyalin sonlu sayıda ve eşit olmaması muhtemel ayrı aralıklara bölme ve her bir aralığa bir değer atama işlemidir.

Kodlama, örneklenen sinyal değerinin ikili koda dönüştürülmesi işlemidir. Kodlama işlemini gerçekleştirmek için alınan örneğin genliğine bakılır. Bu genliğe en yakın basamak hangisi ise o basamağın kodu gönderilir.

Alıcıda ters işlem yapılır. Önce, seri olarak gelen bit dizileri ikili sayıya dönüştürülür. Bu sayı bir sayısal/örneksel dönüştürücü yardımı ile gerilime çevrilir. Elde edilen basamaklı gerilim süzülerek örneksel sinyal tekrar elde edilir.

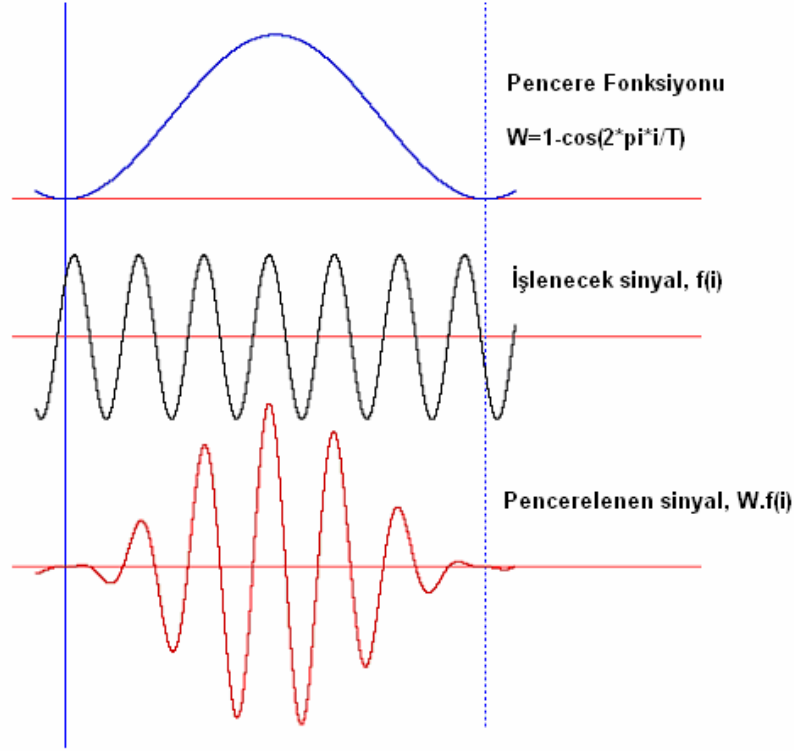
Sesli ifade sinyallerinin sayısallaştırılmasında değişik kodlama yöntemleri kullanılmaktadır (Rabiner, 1978). Bu yöntemler İngilizce adlandırmaları ile aşağıda verilmiştir:

- *PCM Pulse Code Modulation*
- *Log-PCM Logarithmic Pulse Code Modulation*
- *APCM Adaptive Pulse Code Modulation*
- *DPCM Differential Pulse Code Modulation*
- *ADPCM Adapted Differential Pulse Code Modulation*
- *DM Delta Modulation*

4.2 Sesli İfadeler Üzerinde Yürütülen Ön İşlemler

Sesli ifadelerin tanınma sürecine hazırlık amacıyla yapılan bazı ön işlemler vardır. Sayısallaştırılan sesli ifade sinyalleri üzerinde bu ön işlemler uygulanarak özellik vektörlerinin ayırıcı özellikleri artırılmaya çalışılır. Bu işlemlerden başlıcaları pencereleme, filtreleme, sıfır noktasını geçme sayısını hesaplama, enerji hesaplama ve *center clipping* işlemleridir.

Pencereleme işlemi, özellik vektörü çıkarımı esnasında sınırlı uzunluktaki sesli ifade verileri üzerinde uygulanır. Bu veriler doğrudan bir parametre hesaplama girişine veriliyor ise, dikdörtgen pencere kullanımından söz etmek mümkündür. Bu pencereleme yönteminin dışında veriler işlenmeden önce bir takım katsayılarla çarpılabilir. Bu da pencereleme işlemi dahilinde yapılmaktadır (Şekil 4.2).



Şekil 4.2 Pencereleme fonksiyonu

FFT (*Fast Fourier Transformation*) gibi işlemlere ilişkin algoritmalar belirli uzunlukta veri kümeleri üzerinde çalışabilmektedir. FFT ile zaman boyutundan sıklık boyutuna dönüşüm amaçlanmaktadır. Bu belirli uzunluktaki veri kümeleri, belli sayıda sözcük içerir. Bir seferde işleme alınacak sözcük kümesi, sinyal içinde bir pencere (*window*) olarak tanımlanır. Sesli ifadeleri belirli uzunluktaki sözcüklere ayırma işlemi ise pencereleme (*windowing*) olarak bilinir.

Pencereleme işlemi sırasında, incelenen pencere dahilindeki sinyal kesiminin önünde ve arkasındaki kesimlerin ele alınmıyor olması sinyalin bir dikdörtgenle maskelenmiş biçiminin incelenmesi gibi bir sonuç doğurur. Bu durumda sinyalin, işlenen pencere dışında sıfır düzeyindeymiş gibi ele alınması sorun yaratır. Bu sorunu hafifletmek üzere pencereleme işlemi, sinyalin uygun bir fonksiyon ile *convolution* işlemine tabi tutulması olarak düşünülür. *Convolution* işlemi zaman ekseninde, sinyal ile seçilen pencere fonksiyonunun özel bir çarpımıdır.

Convolution işlemi

$$y(t) = x(t) \otimes w(t)$$

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau)w(\tau)d\tau \quad (4.1.a)$$

şeklinde ifade edilmektedir. Burada x sinyali, w *convolution* fonksiyonunu, \otimes ise *convolution* işlemini göstermektedir. *Convolution* işleminin kesikli fonksiyonlar için kullanılan biçimi de aşağıdaki gibidir (Oppenheim, 1989):

$$y[k] = x[k] * w[k]$$

$$y[k] = \sum_{-\infty}^{\infty} x[i] \cdot w[k-i] = \sum_{-\infty}^{\infty} w[i] \cdot x[k-i] \quad (4.1.b)$$

Convolution fonksiyonu olarak bazı özel fonksiyonlar da kullanılmaktadır. Bu özel fonksiyonlar kullanılarak uygulanan pencereleme yöntemlerini şu şekilde sıralamak mümkündür:

Rectangular Window

$$w[i] = \begin{cases} 1 & \text{eğer } 0 \leq i \leq n-1 \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.2)$$

Hamming Window

$$w[i] = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi \cdot i}{n-1}\right) & \text{eğer } 0 \leq i \leq n-1 \\ 0 & \text{değilse} \end{cases} \quad (4.3)$$

Hanning Window

$$w[n] = \begin{cases} \frac{1}{2} \left\{ 1 - \cos\left(\frac{2\pi n}{L-1}\right) \right\} & \text{eğer } 0 \leq n \leq L-1 \\ 0 & \text{değilse} \end{cases} \quad (4.4)$$

Barlett Window

$$w[i] = \begin{cases} \frac{2i}{n-1} & \text{eğer } 0 \leq i \leq \frac{n-1}{2} \\ 2 - \frac{2i}{n-1} & \text{eğer } \frac{n-1}{2} < i \leq n-1 \\ 0 & \text{değilse} \end{cases} \quad (4.5)$$

Blackman Window

$$w[i] = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi \cdot i}{n-1}\right) - 0.08\cos\left(\frac{4\pi \cdot i}{n-1}\right) & \text{eğer } 0 \leq i \leq n-1 \\ 0 & \text{değilse} \end{cases} \quad (4.6)$$

Kaiser Window

İçerdiği β parametresi sayesinde, diğer pencereleme fonksiyonlarına dönüşebilen özel bir pencereleme fonksiyonudur.

Filtreleme işlemi esnasında, sinyal istenmeyen sıklıklardan arındırılır. Sıfırdan geçme sayısını hesaplamadaki amaç, sinyalin sıklık değişimlerini gözlemek; enerji hesaplamadaki amaç ise sesli ifadenin bazı özelliklerini önceden belirleyebilmektir. Bu iki veri, özellikle sinyalin kesimlenmesi amacıyla kullanılmaktadır.

Center clipping işlemi, sesli ifadenin gürültüden ayrılması için kullanılan bir başlangıç yöntemidir. Uygulama esnasında sinyaldeki sıfır noktasına yakın kesimler sinyalden çıkarılır.

4.3 Özellik Vektörünün Çıkarılması

Özellik vektörü çıkarmadaki amaç, sesli ifade verilerinin sesli ifadeyi tanımlayabilecek vektörlere indirgenmesini sağlamaktır. Elde edilen özellik vektörleri, bir sonraki aşama olan sınıflandırmada kullanılırlar. Farklı özellik vektörü çıkarma yaklaşımları mevcuttur. Genel olarak bu yaklaşımlar iki ayrı sınıfta ele alınabilir. Birinci sınıfa ait yaklaşımlar parametrik olarak nitelendirilebilirken, ikinci sınıfa ait yaklaşımlar parametrik olmayan yöntemleri içermektedir.

Özellik vektörlerinin çıkarılmasında kullanılan parametrik yöntemler, bir sesli ifade üretim sistemi öngörürler. Bu yöntemlerde sesli ifadenin kendisi bir çıkış olarak düşünülür ve bu çıkıştan yola çıkarak sesli ifade üretim sisteminin girişi tahmin edilmeye çalışılır. Bu esnada giriş ve çıkış arasında bir sesli ifade üretim fonksiyonu oluşturulur. Bu fonksiyonun parametreleri sesli ifade tanıma sistemi tarafından kullanılacak olan özellik vektörleri olmaktadır.

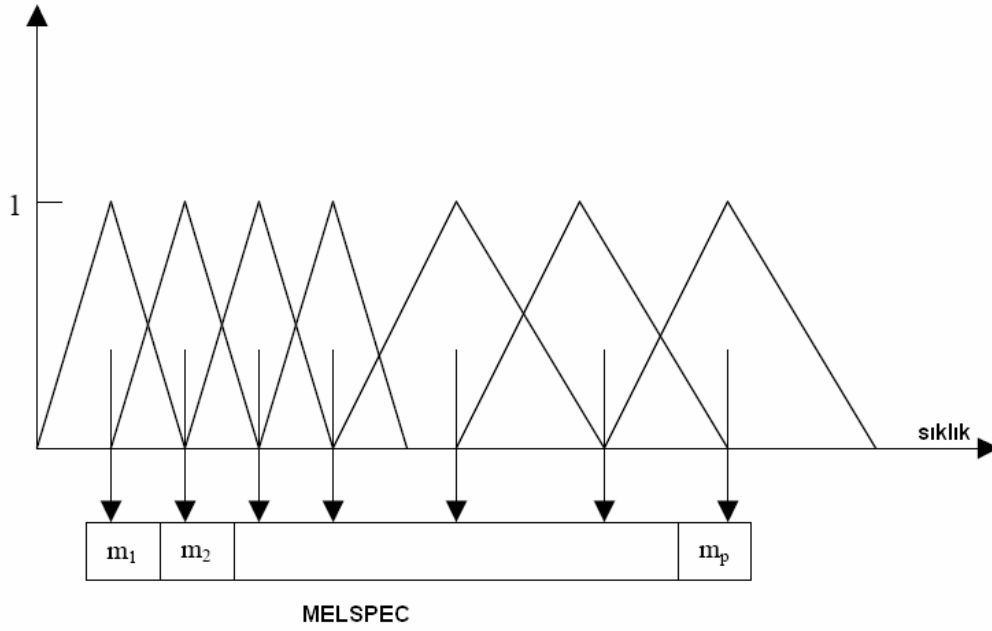
Parametrik olmayan yöntemler ise, sesli ifade sinyali üzerinde pencereler halinde ilerleyerek sinyal üzerinde bazı dönüşümler uygulanması esasına dayanır. Kullanılacak pencere türünün ve uzunluğunun belirlenmesi, bu yöntemlerin ilk aşamalarını oluşturur. Pencere üzerinde daha sonra bir boyut dönüştürme işlemi yapılır. Örneğin Fourier dönüşümünde genlik-zaman boyutu, sıklık-zaman boyutuna dönüştürülür. Daha sonra bu dönüşüm sonucunda elde edilen veriler bir takım iyileştirme işlemleri sonucunda sınıflandırma aşamasına hazır hale getirilir.

Sayısal filtre dizisi tekniği (*filter bank*), özellik vektörü çıkarımında çok sık kullanılan yöntemlerden bir tanesidir. Bu teknik, kulağın çalışma ilkesi esas alınarak geliştirilmiştir. Sesli ifadeyi meydana getiren sıklık bölgeleri filtre dizileri ile ayrılmaya çalışılarak, her sıklık bölgesine ilişkin elde edilen değerler dizisi incelenen sinyali parametrik olarak tanımlama ve modellemede kullanılır. Teknik, tanımlanan her merkez sıklık değeri için değişmez band genişliğinde filtreler kullanılmasını gerektirir. Filtrelerin merkez sıklıkları, akustik sıklık değerinden algılanan sıklık değerine dönüşüm fonksiyonu ile bulunur. Bu bağlamda yaygın olarak, *mel* ve *bark* adlı iki fonksiyon kullanılmaktadır.

Özellik vektörü genelde *mel* sıklığı eksenini esas alınarak bulunmaktadır. *Mel* sıklığına dönüşüm fonksiyonu şu şekildedir:

$$mel = 2595 \log_{10} \left(1 + \frac{f}{700.0} \right) \quad (4.7)$$

Sayısal filtre dizisini oluşturmak için, pencerelenen sesli ifade sinyalinin *Fourier* dönüşümüne ait büyüklüğü ilgili filtre kazancı ile çarpılır ve sonuçlar biriktirilir. Bu sonuçlardan her biri sayısal filtre dizisi kanalındaki izgesel büyüklüğü gösteren bir ağırlıklandırılmış toplam içerir.



Şekil 4.3 Mel-Ölçeği Sayısal Filtre Dizisi (*Mel-Scale Filter Bank*)

Sayısal filtre dizisindeki genliklerin logaritmalarına kesikli kosinüs dönüşümünün (*discrete cosine transform - DCT*) uygulanıp, düşük dereceli terimlerin alınmasıyla özellik vektörü olarak kullanılabilecek, Mel-Frequency Cepstral Coefficients (*MFCC*) adı verilen *cepstrum* katsayıları elde edilir (Şekil 4.3):

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^N \cos \left\{ m_j \left(\frac{\pi * i}{N} (j-0.5) \right) \right\} \quad i = 1, \dots, M \quad (4.8)$$

Burada N , sayısal filtre dizisindeki kanal sayısını, M ise kesikli kosinüs dönüşümünden elde edilen katsayıların sayısını ifade eder.

Cepstrum katsayıları ile modellemede söz konusu katsayılar, hem *Fourier* dönüşümü sonucu elde edilen değerlere hem de *linear prediction* katsayılarına dayalı olarak hesaplanabilir. *Fourier* dönüşüm değerlerine dayalı olarak hesaplanan *cepstrum* katsayıları genelde *Linear Frequency Cepstral Coefficients* (*LFCC*) olarak anılır. Eğer *Fourier* dönüşüm değerleri hesaplanırken kullanılan sıklık değerleri *mel* sıklıkları olarak ele alınırsa, bu dönüşüm değerlerine göre hesaplanan *cepstrum* katsayıları, *Mel Frequency Cepstral Coefficients* (*MFCC*) olarak adlandırılır. *Cepstrum* katsayıları, eğer *LPC*

değerlerine dayalı olarak hesaplanıyor ise *Linear Predictive Cepstrum Coefficients* (LPCC) olarak adlandırılır.

Özellik vektörü oluşturulurken, *cepstrum* katsayılarına ek olarak enerji terimi de kullanılabilir. Tüm bu katsayıları durağan olarak nitelendirmek mümkündür. Durağan katsayılara bir de bu katsayıların birinci ve ikinci dereceden türevleri ilave edilerek özellik vektörleri oluşturulabilir.

4.4 Özellik Vektörlerinin Sınıflandırılması

Özellik vektörlerinin sınıflandırılmasında bir çok yöntem bulunmaktadır:

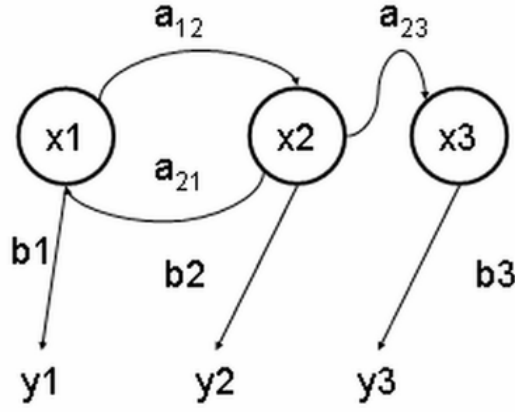
- *Hidden Markov Model* (HMM)
- *Time Warping-Dynamic Time Warping* (DTW)
- Nöron Ağları (*Neural Network*-NN)

4.4.1 Hidden Markov Model

Hidden Markov Model (HMM) sesli ifade tanımada kullanılan yaygın yöntemlerden bir tanesidir (Tebelskis, 1979). Yöntemin popüler olmasının bir nedeni, sahip olduğu zengin matematiksel yapı; diğer bir neden ise uygun bir biçimde uygulandığında başarılı sonuçlar elde edilmesidir.

Hidden Markov Model ile sinyal değişken (stokastik) olarak modellenir. Bir sesli ifade tanıma sistemi dahilinde kullanılan HMM oluşturma sürecinde, ardışık kısa süreli sesli ifade kesimleri birlikte ele alınır ve ardı ardına gelebilecek bu kesimler için bir model oluşturulup, bu model aracılığıyla uzun süreli sesli ifadelerin tanınması sağlanır.

Bu model *Markov* zincirini esas alarak her sözcük ya da fonem için bir zincir üretir. Ses organları sesi üretmek için sesi çeşitli pozisyonlara taşır. Meydana gelen sesin değişikliğinin derecesi vardır. Bu yüzden mümkün çıkışların bir kümesini bir durum ile ilişkilendirmek mümkündür. Bir organın her pozisyonuna bir durum atanırsa, durumlar arası geçişlerden söz edilebilir.



Şekil 4.4 Hidden Markov Model'de durumlar ve durumlar arası geçişler

Bir HMM durumlar ve bu durumlar arası geçişlerden oluşur (Şekil 4.4). HMM'de durumlar ve durumlar arası geçişler, olasılık temsiline dayanır.

Bir Markov zinciri, minimum büyüklükte bellek içeren bir rasgele işlemler sınıfını modeller. Öncelikle, kesikli zamanlı Markov zincirini ele alalım. $X=X_1, X_2, \dots, X_n$ sonlu, kesikli bir alfabe olan $O=\{o_1, o_2, \dots, o_M\}$ seçilen bir rasgele değişkenler dizisi olsun. Bayes' kuralına göre (Tebelskis, 1979):

$$P(X_1, X_2, \dots, X_n) = P(X_1) \prod_{i=2}^n P(X_i | X_{i-1}) \quad (4.9)$$

olarak verilir. Burada, $X_{i-1} = X_1, X_2, \dots, X_{i-1}$ 'dir. $P(X_i | X_{i-1}) = P(X_i | X_{i-1})$ olmak şartıyla, X rasgele değişkenlerinin birinci dereceden bir Markov zinciri oluşturdukları söylenebilir. Sonuç olarak, birinci dereceden Markov zinciri şu şekilde yazılabilir:

$$P(X_1, X_2, \dots, X_n) = P(X_1) \prod_{i=2}^n P(X_i | X_{i-1}) \quad (4.10)$$

$P(X_i | X_{i-1}) = P(X_i | X_{i-1})$ eşitliği aynı zamanda Markov varsayımı (*Markov assumption*) olarak da bilinir. Bu varsayım, dinamik veri dizilerini modellemek için çok az bellek kullanır, çünkü verilen bir zamanda rasgele bir değişkenin olasılığı sadece bir önceki zamandaki değerine bağlıdır. Markov zinciri, eğer zaman dizini olan i değeri atılırsa, zamanla değişmeyen (durağan) olayları modellemek için kullanılır.

$$P(X_i = s | X_{i-1} = s') = P(s | s') \quad (4.11)$$

Eğer X_i bir durumla ilişkilendirilirse; Markov zinciri, $P(s | s')$ olasılık fonksiyonu ile durum geçişleri belirtilen bir sonlu durum süreci ile gösterilir. Bu sonlu durum gösteriminin kullanılmasıyla birlikte, Markov varsayımının anlamını şu şekilde yorumlamak mümkün olur: Markov zincirinin verilen bir zamanda belli bir durumda olma olasılığı sadece, o Markov zincirinin bir önceki zamandaki durumuna bağlıdır.

$\{1, \dots, N\}$ ile etiketlenmiş N tane farklı durumdan oluşan ve t zamanında s_t durumunda bulunan bir Markov zinciri göz önünde bulundurulacak olursa, bu Markov zincirinin parametreleri şu şekilde tanımlanabilir:

$$a_{ij} = P(s_t = j | s_{t-1} = i) \quad 1 \leq i, j \leq N \quad (4.12)$$

$$\pi_i = P(s_1 = i) \quad 1 \leq i \leq N \quad (4.13)$$

Burada a_{ij} i durumundan j durumuna geçiş olasılığı, ve π_i Markov zincirinin i durumunda başlayacağı ilk olasılık değeridir. Geçiş ve başlangıç olasılıkları şu kısıtlara bağlıdır:

$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N \quad (4.14)$$

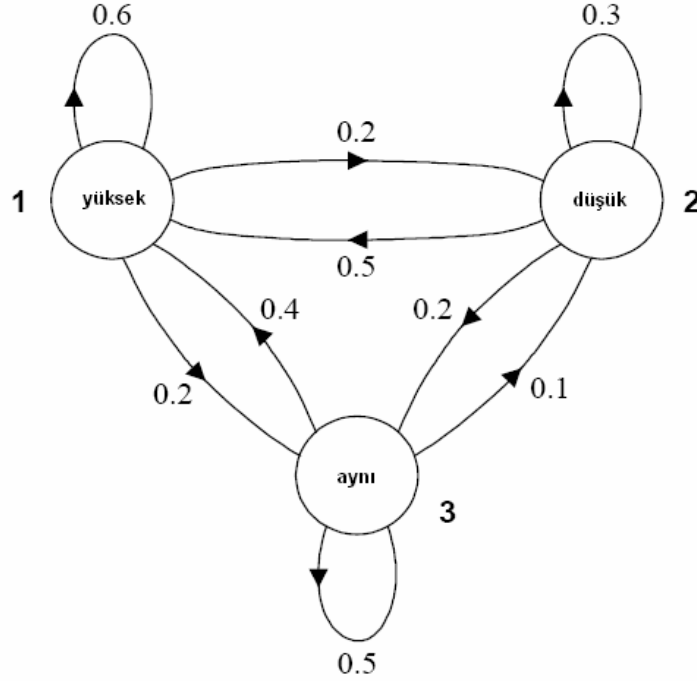
$$\sum_{j=1}^N \pi_j = 1; \quad (4.15)$$

Üstte açıklanan Markov zinciri aynı zamanda gözlenebilir Markov zinciri olarak da adlandırılır çünkü sürecin çıktısı, her birinin bir X_i gözlenebilir durumuyla ilişkili olduğu durumların t zaman birimindeki kümesi şeklindedir. Diğer bir söyleyişle X gözlenebilir olay zinciri ile, $S = s_1, s_2, \dots, s_n$ Markov zinciri durum dizisi arasında birebir bir ilişki söz konusu değildir. İstanbul Menkul Kıymetler Borsası'na (İMKB) ilişkin üç durumlu bir Markov zinciri Şekil 4.5'te verilmiştir. Her günün sonunda ortalama bu üç durumdan bir tanesinde olacaktır:

durum 1 – yüksek (bir önceki günün diziniyle karşılaştırıldığında)

durum 2 – düşük (bir önceki günün diziniyle karşılaştırıldığında)

durum 3 – aynı (bir önceki günün diziniyle karşılaştırıldığında)



Şekil 4.5 İMKB'ye ilişkin bir Markov zinciri

İMKB Markov zincirine ait parametre, şu şekilde bir durum-geçiş olasılık matrisini içerebilir:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

Ve olası bir başlangıç durum olasılık matrisi şu şekilde olabilir:

$$\pi = (\pi_i)^t = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.3 \end{pmatrix}$$

5 ardışık "yüksek" durumu için olasılığın bulunması istendiğinde, yüksek-yüksek-yüksek-yüksek-yüksek gözlenen dizisine (1,1,1,1,1) durum dizisine karşılık geldiği için; olasılık:

$$P(5 \text{ ardışık "yüksek" gün}) = P(1,1,1,1,1)$$

$$\pi_1 a_{11} a_{11} a_{11} a_{11} = 0.5 \times (0.6)^4 = 0.0648$$

Markov zincirinde her bir durum deterministik olarak gözlenebilir bir olayla ilişkilendirilir, bu durumda her bir durumda üretilen çıktı rasgele değildir. Markov zincirinin doğal bir uzantısı olarak gelişen bir yaklaşım da verilen herhangi bir durumda çıktı gözlem sembollerinin üretildiği deterministik olmayan bir süreç önerir. Böylece gözlem, durumun olasılık fonksiyonudur. Bu yeni model saklı Markov modeli (*hidden Markov model*) olarak bilinir. Bu model, alttaki olasılıklı süreci (durum dizisi) doğrudan gözlenebilir olmayan, bir çift katmanlı olasılıklı süreç olarak görünür. Bu yüzden Markov modelinin önüne "saklı" kelimesi de eklenmiştir. Saklı Markov modelinin durumu durumu gizli de olsa, modellenen veriyle ilgili açık bir bilgi içerir. Örneğin Şekil 4.6'da verilen İMKB ortalamasına ait saklı Markov modelinde 1. durum, borsa fiyatlarının yükseldiği bir piyasayı; 2. durum ise borsa fiyatlarının düştüğü bir piyasayı göstermektedir.

Bir saklı Markov modeli şu özelliklerle tanımlanır:

- $O = \{o_1, o_2, \dots, o_M\}$ - Bir çıktı gözlem alfabesi ile. Gözlem sembolleri, modellenen sistemin fiziksel çıktısına karşılık gelir. İMKB ortalamasına ait saklı Markov modelinde, çıktı gözlem alfabesi 3 kategoriden oluşur: $O = \{\text{yüksek}, \text{düşük}, \text{aynı}\}$.
- $\pi = \{1, 2, \dots, N\}$ - Durum uzayını gösteren durumlar kümesi. s_t , t zamanındaki durumu ifade etmektedir. İMKB ortalaması düşünüldüğünde ilgili saklı Markov modelinde, durum; borsa fiyatlarının yükseldiği bir piyasayı, borsa fiyatlarının düştüğü bir piyasayı veya kararlı bir piyasayı gösterir.
- $A = \{a_{ij}\}$ - Bir geçiş olasılık matrisi. a_{ij} i durumundan j durumuna bir geçişin olma olasılığı:

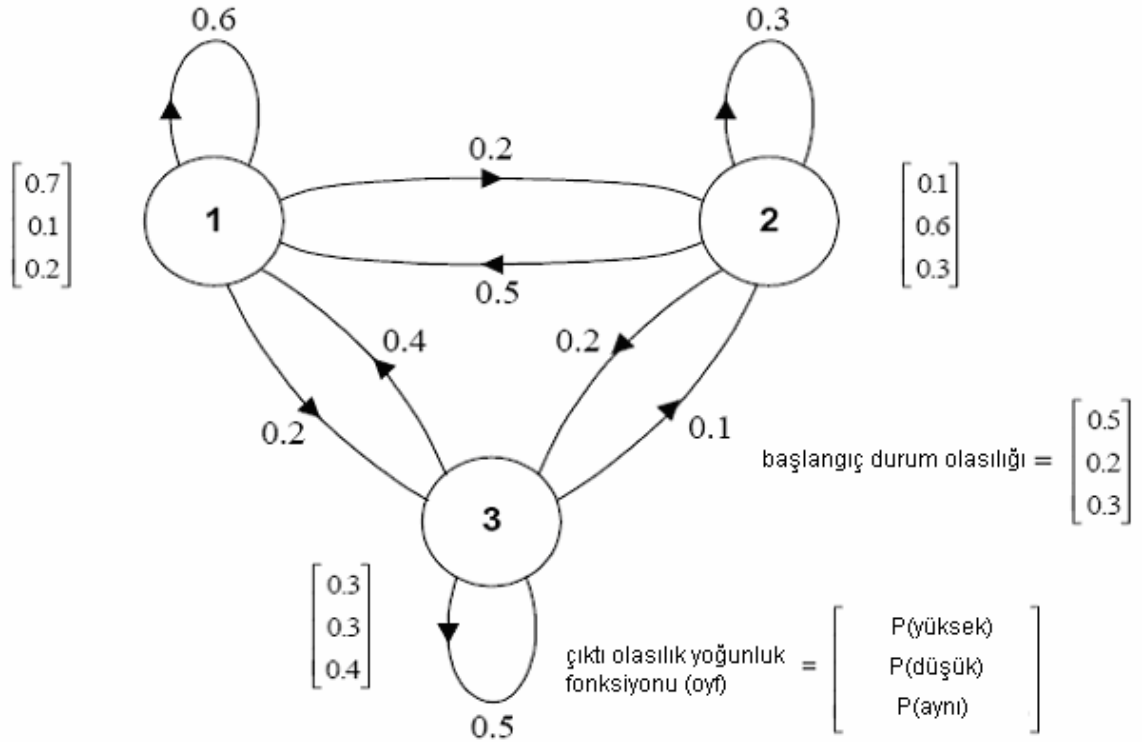
$$a_{ij} = P(s_t = j | s_{t-1} = i) \quad (4.16)$$

- $\mathbf{B} = \{b_i(k)\}$ - $b_i(k)$ 'nin, i durumu verildiğinde o_k yayılım sembolünün olasılığını gösterdiği, bir çıktı olasılık matrisi. $\mathbf{X} = X_1, X_2, \dots, X_t$ saklı Markov modelinin gözlenen çıktısı olsun. Durum dizisi $S = s_1, s_2, \dots, s_t$ gözlenen değildir (gizli), ve $b_i(k)$ şu şekilde yazılabilir:

$$b_i(k) = P(X_t = o_k | s_t = i) \quad (4.17)$$

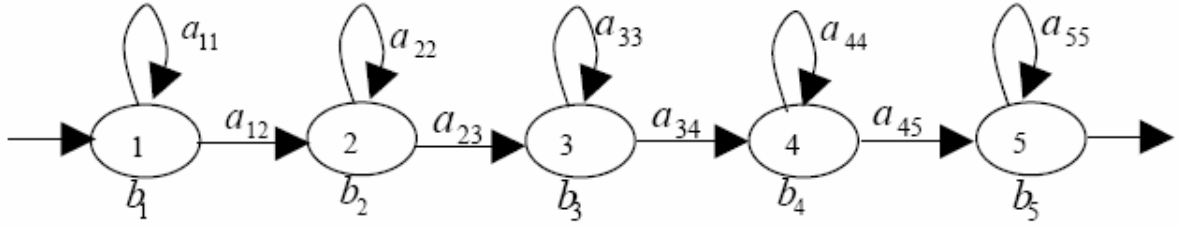
- $\boldsymbol{\pi} = \{\pi_i\}$ - Bir başlangıç durum dağılımı.

$$\pi_i = P(s_0 = i) \quad 1 \leq i \leq N \quad (4.18)$$



Şekil 4.6 İMKB'ye ilişkin *Markov* zincirine ait çıktı olasılık yoğunluk fonksiyonu, başlangıç durum olasılıkları

Sesli ifade bağlamında gözlemler özellik vektörleridir. HMM durumları ise temel alınan sesli ifade birimlerine denk gelir. Bu durumda amaç, saklı olan durum dizisini gözlemlerden yararlanarak bulmaktır. Fonem tabanlı bir sesli ifade tanıma sisteminde bulunan durum dizisi fonem dizisine denk gelir (Şekil 4.7). Her fonem için ayrı bir model tanımı yapılır.



Şekil 4.7 Bir fonem tabanlı sesli ifade tanıma sistemine ait *Markov* zinciri

Her bir fonem için böyle bir model oluşturulduğu düşünülürse ardı ardına gelen fonem zincirleri bu modellerin ardı ardına sıralanması ile modellenebilir. Bu durumda her bir fonemin son durumundan bir sonraki fonemin ilk durumuna geçişi söz konusudur. Böylece tüm sesli ifade tanıma sistemi bir Markov işlemi olarak tanımlanabilir. Böyle bir model, farklı uzunluktaki fonem dizilerini tanımak için kullanılabilir.

HMM geliştirme aşaması üç ana adımdan oluşur. Bunlar:

- Hesaplama: Bu aşamada, verilen bir model için belirli bir gözlem dizisinin olasılığı hesaplanır.
- Çözümleme: Bu aşamada, verilen bir model için belirli bir gözlem dizisini oluşturan durum dizisi bulunur.
- Öğrenme: Bu son aşamada ise verilen bir model için bir dizi gözlemi oluşturan bir modelin olasılığının yüksek olması için model parametreleri belirlenir.

4.4.2 Time Warping-Dynamic Time Warping Yaklaşımı

Aynı sözcük ya da foneme ait sesli ifade sinyalinin uzunlukları, aynı kişi seslendirdiğinde dahi değişebilmektedir. Bu yöntem ile, sinyaldeki bu uzunluk farklılıklarının yok edilmesi amaçlanır. Bu, sözcüğün ya da fonemin sinyalinin referans şablonu ile aynı zaman aralığında olabilmesi için zaman ekseninde daralma ya da genişleme yaparak gerçekleştirilir. Sözcük tanıma ya da fonem için genel olarak *Dynamic Time Warping* (DTW) yöntemi kullanılmaktadır. Bu yöntemde zaman eksenini doğrusal olmayan bir biçimde genişletilip daraltılarak referans şablonu ile tanınacak olan sesli ifade kesiminin başlangıç ve bitiş

zamanları akıřtırılmaya alıřılır. Ama, karřılařtırmanın aynı zaman aralıklarında yapılmasını saėlamaktır.

4.4.3 Nron Aėları Yaklařımı

Nron aėları, sesli ifade tanımada son zamanlarda ok yaygın olarak kullanılmaktadır. ok farklı trlerde nron aėları mevcuttur. Ancak bunların hepsi 4 temel zelliėe sahiptir:

- İřlem birimleri kmesi
- Baėlantılar kmesi
- Hesaplama iřlemi
- Eėitim iřlemi

İřlem Birimleri

Bir nron aėı, insan beynindeki nronlara karřılık gelen ok sayıda basit iřlem biriminden oluřur. Tm bu birimler kořutluėu saėlayacak řekilde, eř zamanlı olarak alıřırlar. Sistemdeki tm hesaplama, bu birimler tarafından gerekleřtirilir, onların iřlemlerini ngren veya yneten, sistemde herhangi bařka iřlemci yoktur. Her bir anlık zaman biriminde, her iřlem birimi yerel girdilerine sayısal bir fonksiyon uygular ve sonucu (aktivasyon deėeri) komřu birimlere iletir.

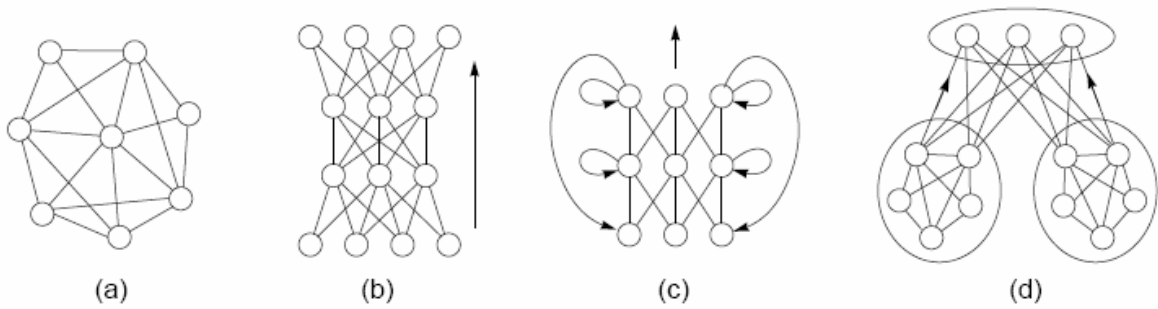
Bir aėdaki birimler; evreden veriyi alan girdi birimleri, i veri gsterimini oluřturan saklı birimler ve kararlar ile denetim sinyallerini gsteren ıktı birimleri olmak zere ayrılırlar.

Baėlantılar

Bir aėdaki birimler, verilen herhangi bir topolojiyi oluřturmak zere baėlantılar (veya aėırlıklar) ile organize olurlar. Her aėırlık, $-\infty$ 'dan ∞ 'a deėiřen bir gerel deėere sahiptir. Bir aėırlıėın deėeri (gc), o birimin komřu birimler zerinde ne kadar etkili olduėunu gsterir; pozitif bir aėırlık deėeri o birimin diėerini uyarmasını saėlarken, negatif bir aėırlık deėeri o birimin komřu birimi engellemesine neden olur. Aėırlıklar oėunlukla tek ynldr (girdi birimlerinden ıktı birimlerine doėru)

ancak girdi ve çıktı birimleri arasında herhangi bir ayrım yok ise iki yönlü de olabilirler.

Bütün ağırlıkların değerleri, bir ağına verilen girdi örüntüsü ile nasıl bir sayısal reaksiyona neden olacağını belirler. Böylece ağırlıklar, bir ağına uzun dönemli belleğini meydana getirirler. Ağırlıklar, bir eğitim sürecinin sonucunda değişirler, fakat bu değişim zaman alır; çünkü birikmiş bilginin değişmesi yavaş gerçekleşir. O anlık verilen girdinin ağıda meydana getirdiği değişim ise kısa zamanlı bellek olarak bilinir.



Şekil 4.8 Farklı yapay sinir ağı modelleri a) yapısal olmayan, b) katmanlı, c) özyineli, d) modüler

Bir ağı, çok farklı topolojiler ile bağlanabilir (Şekil 4.8). En çok kullanılan topolojiler; yapısal olmayan, katmanlı, özyineli ve modüler ağlardır. Her topoloji, amaca uygun olarak belli bazı uygulamalar için kullanılır. Örneğin;

- Yapısal olmayan ağlar, en çok örüntü tamamlama uygulamalarında kullanılmaktadır.
- Katmanlı ağlar, en çok örüntü birleştirme uygulamalarında kullanılmaktadır.
- Özyineli ağlar, en çok örüntü sıralama uygulamalarında kullanılmaktadır.
- Modüler ağlar ise, daha basit düzeydeki bileşenlerden karmaşık sistemler oluşturmak için kullanılırlar.

Hesaplama

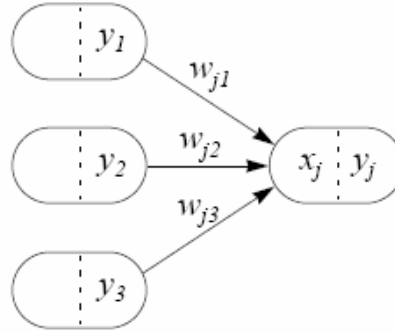
Hesaplama, her zaman bir girdi örüntüsünün ağına verilmesi ile başlar. Bunun ardından diğer işlem birimleri, eşzamanlı veya eşzamanlı olmayan bir şekilde

aktivasyon değerlerini hesaplarlar. İleri beslemeli ağlarda aktivasyon değerlerinin iletilmesi, girdi katmanından çıktı katmanına doğru tek yönlü olurken; özyineli ağlarda ağ içindeki aktivasyon değerlerinin iletilmesi sırasında geri dönüşler de olabilmektedir.

Ağdaki herhangi bir birim iki aşamada güncellenir: ilk olarak net girdi hesaplanır, ardından net girdi değerinin bir fonksiyonu olarak birimin aktivasyon çıktısı hesaplanır. Bir j biriminin net girdisi olan x_j , girdilerinin ağırlıklandırılmış toplamı biçiminde hesaplanır:

$$x_j = \sum_i y_i w_{ji} \quad (4.19)$$

Burada y_i , birimin çıktı aktivasyon değerini w_{ji} ise i biriminden j birimine olan ağırlık değerini ifade etmektedir (Şekil 4.9).



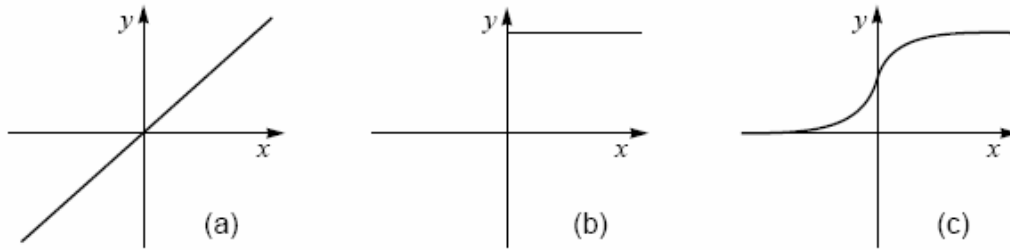
Şekil 4.9 Bir nöronun çıktı değerinin hesaplanması

Genellikle, net girdi değeri bir θ , bias terimi ile dengelenir. Bu durumda net girdi değerini şu şekilde yazmak mümkündür:

$$x_j = \sum_i y_i w_{ji} + \theta_j \quad (4.20)$$

Ancak uygulamada bias terimi, aktivasyon değeri $y_0 = 1$ olan hayali bir birime bağlı w_{j0} ağırlık değeri olarak hesaba katılır. Böylece yukarıda verilen formülde hesap aralığı, hayali birimi de kapsayacak biçimde genişletilir.

Net girdi X_j 'nin hesaplanmasının ardından, çıktı aktivasyonu y_j , X_j 'nin bir fonksiyonu olarak hesaplanır. Aktivasyon fonksiyonu belirleyici (*deterministic*) veya olasılıklı (*stochastic*) olabilir. Belirleyici aktivasyon fonksiyonları genelde 3 biçimde olurlar: doğrusal (*linear*), *threshold*, *sigmoidal*. İlgili fonksiyonlar Şekil 4.10'da verilmiştir.



Şekil 4.10 Belirleyici yerel aktivasyon fonksiyonlar: a)doğrusal (*linear*), b)*threshold*, c)*sigmoidal*

Doğrusal fonksiyon basit olarak; $y=x$ olarak ifade edilir. Ancak bu fonksiyon pek etkili değildir, çünkü doğrusal birimlerden oluşan birden fazla katman aynı işlevselliğe sahip tek bir katman gibi davranır. Bunun için genelde doğrusal olmayan aktivasyon fonksiyonları kullanılmaktadır. Bunlardan biri de *threshold* fonksiyonudur. Bu fonksiyona göre;

$$y = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (4.21)$$

eşitsizliği kullanılarak çıktı aktivasyon değeri hesaplanır.

Threshold fonksiyonundan daha etkili olan bir diğer doğrusal olmayan fonksiyon ise *sigmoidal* fonksiyondur. *Sigmoidal* fonksiyona göre çıktı aktivasyonu şu şekilde hesaplanır:

$$y = \frac{1}{1 + \exp(-x)} \text{ veya } y = \tanh(x) \quad (4.22)$$

Sigmoidal fonksiyon, *threshold* fonksiyonundan farklı olarak süreklilik özelliğini gösterir ve türevi alınabilmektedir. Ayrıca geri yayılım (*backpropagation*) algoritmasını da desteklemektedir.

Eğitim

Nöron ağı'nın eğitilmesi aşamasında, verilen girdi örüntüleri için ağı'nın istenen değerleri üretmesini sağlamak üzere bağlantılar uyarlanır. İşlem genelde ağırlıkların değiştirilmesi şeklinde olur, ancak eğitim aşamasında topolojinin değiştirilmesi, örneğin bazı bağlantıların yok edilip, yeni bağlantılar eklenmesi de mümkündür.

Nöron ağıları doğrusal olmadıkları ve katmanlı bir yapıya sahip oldukları için ağırlıkları *gradient descent* gibi ancak yinelemeli (*iterative*) bir süreç ile eğitilebilir. Bu süreçte eğitim kümesinden birden fazla geçilmektedir, her bir geçiş *epoch* olarak anılmaktadır. Öğrenilen ağırlıklar bütün ağı yayıldığı için bu ağırlıkların her bir geçişte belli oranda değiştirilmesi gerekir, bu oranı belirleyen sabit değer ise öğrenme oranıdır (*learning rate*- ϵ). Öğrenme oranı için iyi bir değer bulmak büyük önem taşır. Bu değer çok küçük olması, eğitimin sonsuza kadar sürmesine; çok büyük olması ise, bir önceki geçişte edinilen bilginin yok olmasına neden olur. En uygun öğrenme oranını bulmada herhangi bir yöntem yoktur, ancak farklı değerlerin denenmesiyle en uygun öğrenme oranı bulunabilmektedir.

Nöron ağıları, öğrenme süreçlerine göre sınıflandırılabilirler. Bu durumda öncelikle öğrenme süreçlerini sınıflandırmak gerekir. Üç farklı türde öğrenme süreci olabilir:

Denetimli öğrenme (*Supervised learning*): Bu öğrenme sürecinde ağı eğitmek için bir öğretici kullanılır. Öğretici, basitçe çıktı katmanında ağı'nın ne üretmesi gerektiğini söyler. Bir girdi-çıkı örneği ağına verilir ve bir dizi hesaplama işlemleri sonucunda elde edilen çıktı, doğru çıktı ile karşılaştırılır. Ağırlıklar, daha iyi çıktıyı üretmek için yeniden ayarlanır ve bu işlem, kabul edilebilir bir hata seviyesine erişinceye dek devam eder.

Destekli öğrenme (*Semi-supervised/reinforcement learning*): Destekli öğrenme sürecinde de bir öğreticinin varlığı gerekmektedir. Ancak bu öğretici çıktının ne olması gerektiğini ağına söylemez, bunun yerine üretilen çıktının doğru veya yanlış olduğunu belirtir.

Denetimsiz öğrenme (*Unsupervised learning*): Denetimsiz öğrenme sürecinde bir öğreticiye gerek duyulmaz. Bu süreçte ağı, girdi-çıkı eşleştirmesini düzenlemek

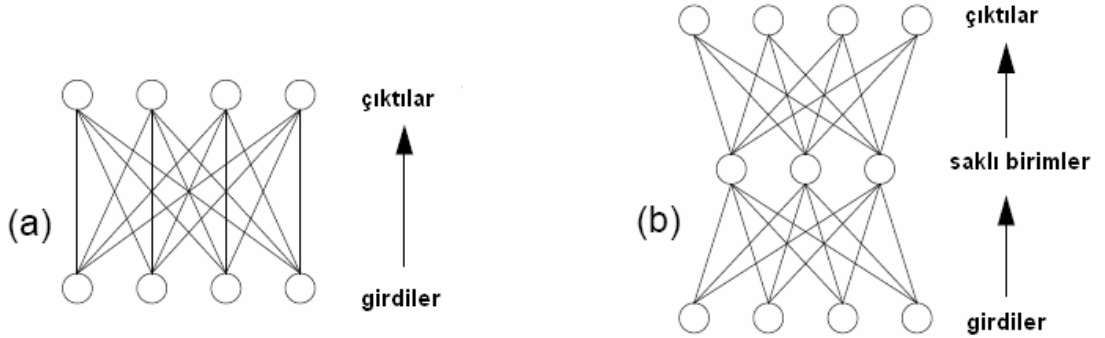
için kendi ölçütlerini geliştirir. Bu nedenle, denetimsiz öğrenme türünü kullanan ağlar, kendi kendine organize olan ağlar olarak adlandırılırlar.

4.4.3.1 Denetimli Öğrenme (*Supervised Learning*)

Denetimli öğrenme sürecini kullanan farklı türde nöron ağları mevcuttur. Bunları iki sınıfta incelemek mümkündür: İleri beslemeli ağlar (*Feedforward networks*), özyineli ağlar (*recurrent networks*)

4.4.3.1.1 İleri Beslemeli Nöron Ağları (*Feedforward Neural Networks*)

Algılayıcı (*perceptron*), denetimli öğrenmeyi kullanan en basit düzeydeki ileri beslemeli nöron ağı türüdür. Tek katmanlı veya çok katmanlı olabilir (Şekil 4.11).



Şekil 4.11 *Perceptron* modelleri a) Tek katmanlı nöron ağı, b) Çok katmanlı nöron ağı

Çok katmanlı nöron ağlarının eğitilmesinde geri yayılım öğrenme algoritması (*backpropagation learning algorithm*) kullanılmaktadır. Anlaşılması kolay bir algoritma olması nedeniyle geri yayılım öğrenme algoritması, en yaygın kullanılan öğrenme algoritmasıdır. Algoritmada, hatalar çıkıştan girişe doğru azaltılmaya çalışıldığı için geri yayılım ismini almıştır. Geri yayılım öğrenme algoritması, ağ çıkışındaki mevcut hata düzeyine göre her bir katmandaki ağırlıkların yeniden hesaplanması esasına dayanır.

Geri yayılım öğrenme algoritmasında öncelikle ağda ileri doğru bir hesaplama yapılır ve her bir birimin aktivasyon çıkışları hesaplanır. Amaç, çıkışta bulunan değerlerin bulunması gereken değerlere yaklaştırılmasını sağlamaktır. İleri doğru yapılan hesaplamanın ardından, çıkış katmanından geriye doğru her bir işlem birimi için hata oranı hesaplanır ve bu hatayı düşürecek şekilde ağırlık değerleri

g nlenir. Geriye doęru hatanın hesaplanması ve aęırlıkların g nlenmesi, geri yayılım algoritmasının temelini oluřturmaktadır.

Geri yayılım algoritması řekil 4.12'de verilmiřtir:

- Bařlangıç aęırlık deęerleri belirlenir
- Hata oranı b y k olduęu s rece
 - Her bir eęitim  r nt s  i in
 - Aęa giriř deęerleri uygulanır
 - Ara katmanlar ve  ıkıř katmanındaki t m n ronların  ıkıřları hesaplanır:

$$O_{pj}(\text{net}_j) = \frac{1}{1 + e^{-\lambda \text{net}_j}}$$

$$\text{net}_j = \sum_k O_{pk} W_{jk}$$

-  ıkıř katmanında hata hesaplanır.

$$\delta_{pj} = (T_{pj} - O_{pj}) O_{pj} (1 - O_{pj})$$

- Ara katman(lar)da bir sonraki katmanın hesapladıęı hataya g re geriye hata oranları hesaplanır.

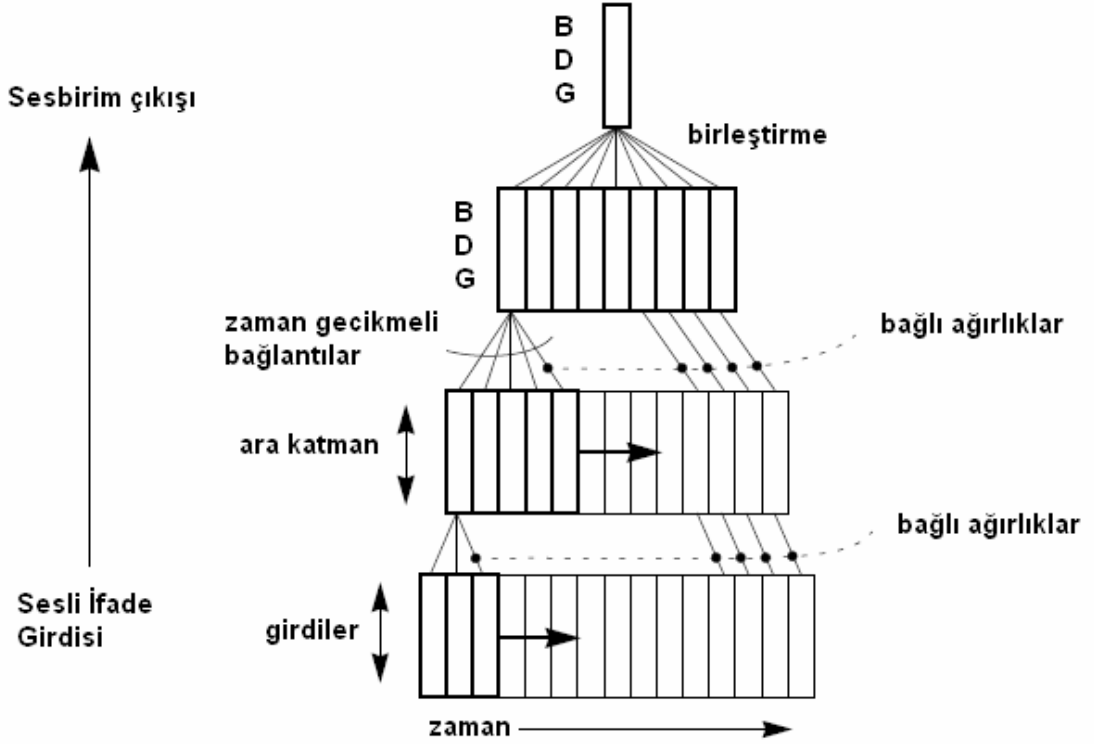
$$\delta_{pj} = O_{pj} (1 - O_{pj}) \sum_k \delta_{pk} W_{kj}$$

- Aęırlık deęerleri hata oranlarına g re yeniden g nlenir.

$$\Delta W_{ji}(t) = \beta \delta_{pj} O_{pj}$$

řekil 4.12 Geri Yayılım  ęrenme Algoritması

Çok katmanlı nöron ağlarının bir başka türü ise zaman gecikmeli nöron ağlarıdır (*Time Delay Neural Networks-TDNNs*). Sesbirim tanımak üzere geliştirilmiş bir zaman gecikmeli nöron ağ yapısı Şekil 4.13'te verilmiştir.



Şekil 4.13 b,d,g fonemlerini tanıma amaçlı geliştirilmiş bir zaman gecikmeli nöron ağı modeli

Zaman gecikmeli nöron ağları iki boyutlu girdiler ile çalışır. Sesli ifade tanıma bağlamında yatay boyut zaman olur. Zaman gecikmeli nöron ağlarının üç temel mimari özelliği vardır:

Zaman gecikmeleri sıradüzensel bir yapıya sahiptir. Daha yüksek düzeydeki birimler geçici bağlamları ele alır ve yüksek düzeyde bir özellik tespitine imkan verirler.

Ağırlıklar zaman eksenini boyunca bağlanmıştır. Böylece, farklı geçici noktalardaki ağırlıklar aynı değeri paylaşırlar.

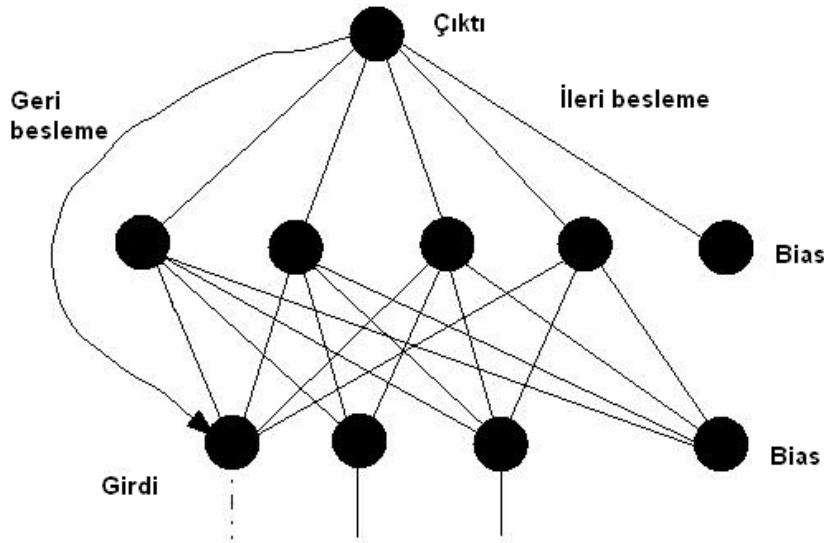
Çıktı birimleri, zaman içinde dağılmış olup bulunan yerel özelliklerin sonuçlarını geçici olarak birleştirir. Böylece örüntüler, rastlandıkları andan bağımsız olarak tanınabilmektedir.

Zaman gecikmeli nöron ağları geri yayılım algoritması kullanılarak eğitilirler. Tek farklı nokta ise, ağırlıklarının birbirinden bağımsız olarak değil de, ortalama hataya göre güncellenmesidir.

4.4.3.1.2 Özyineli Nöron Ağları (Recurrent Neural Networks)

Sabit süreli zaman pencerelerinden kaçınmak için özyineli nöron ağı kavramı (*Recurrent Neural Networks*) ortaya çıkmıştır. Bu ağ türü ile daha önceden edinilmiş bilgiler, belirlenmemiş bir süre boyunca ağda tutulabilmektedir.

Bir sürekli nöron ağı, geriye doğru besleme özelliğinin eklendiği, bir dizi standart hata yayımlı ağı birleşimi şeklinde düşünülebilir (Şekil 4.14). Giriş ve çıkış vektörleri, iç ve dış olmak üzere ikiye ayrılır. İç çıkış verileri bir sonraki vektörün iç giriş kesimine geri besleme olarak verilir. Böylece zaman içinde bir süreklilik sağlanır.



Şekil 4.14 Bir özyineli nöron ağı modeli

Özyineli nöron ağlarının farklı türleri bulunmaktadır. Bunlardan en bilinenleri; Hopfield, Elman, Jordan ağlarıdır.

4.4.3.2 Destekli Öğrenme (Semi-Supervised/Reinforcement Learning)

Destekli öğrenmede, bir öğretici ağı çıkışları için olması gereken hedefleri sağlamaz, ancak ağı davranışını iyi ya da kötü olarak nitelendirir.

4.4.3.3 Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenmede herhangi bir öğretici bulunmaz, ağ verilen girdideki düzenlilikleri kendi tespit etmek durumundadır. Kendinden düzenlemeli ağlar (*Self-Organizing Networks*) girdi verisini sıkıştırmak, kümelemek, sınıflandırmak ve eşleştirmek için kullanılabilir.

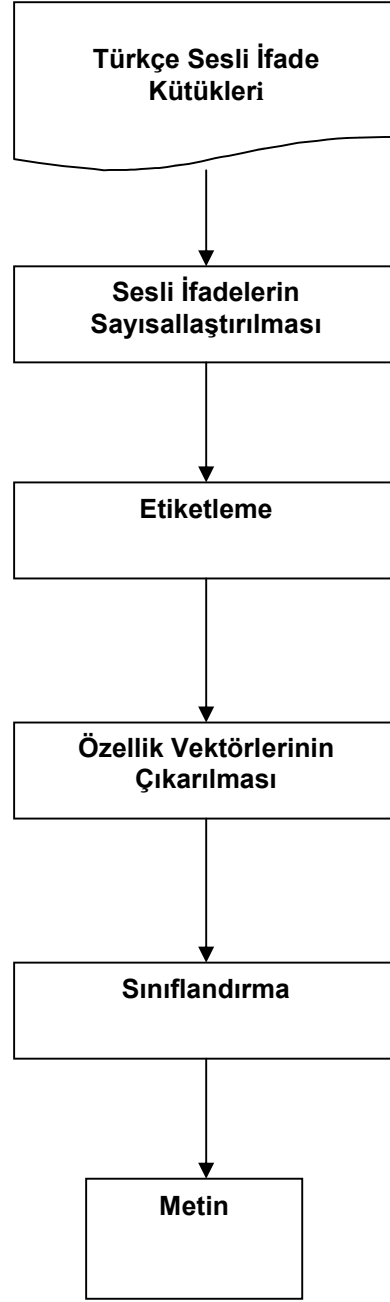
5 BİR HECE TABANLI TÜRKÇE SESLİ İFADE TANIMA SİSTEMİNİN TASARIMI VE GERÇEKLEŞTİRİMİ

Tez kapsamında hece tabanlı (*syllable based*), konuşmacıya bağımlı (*speaker dependent*) bir sesli ifade tanıma sisteminin tasarımı ve gerçekleştirimi hedeflenmiştir.

Sesli ifade tanıma sürecine hazırlık aşamasında, sesli ifadenin görüntülenmesi, üzerinde bir takım ön işlemlerin yapılması, sesli ifadenin enerji ve sıfırdan geçiş oranlarının hesaplanması, hece sınırlarının belirlenip etiketlenmesi gibi amaçlara yönelik bir araç geliştirilmiştir. Bu araç yardımı ile sesli ifade tanıma süreci boyunca gerekli olacak deney verileri, hece sınırları belirlenerek hazırlanmış ve bu verilerin özellik vektörleri çıkarılarak sesli ifade tanıma sürecinin son adımı olarak yapay sinir ağları kullanılarak sınıflandırma işlemi gerçekleştirilmeye çalışılmıştır.

5.1 Deneysel Çalışmanın Aşamaları

Deneysel çalışma 3 ana adımdan oluşmaktadır. İlk adımda, geliştirilen araç yardımıyla sesli ifadenin hece sınırları belirlenmiş ve etiketlenmiştir. İkinci adımda, edinilen hecelere ait özellik vektörleri MFCC (*Mel Frequency Cepstrum Coefficients*) kullanılarak çıkarılmış, son adımda ise bu özellik vektörleri çok katmanlı yapay sinir ağları modelinin eğitimi için kullanılarak ilgili sesli ifade tanıma sisteminin gerçekleştirimi tamamlanmıştır.



Şekil 5.1 Türkçe hecelerin sınıflandırılmasında kullanılan sistemin iş-akış çizelgesi

Geliştirilen kişiye bağımlı, hece tabanlı sesli ifade tanıma sistemine ait işlem dizisi şeması Şekil 5.1'de verilmiştir:

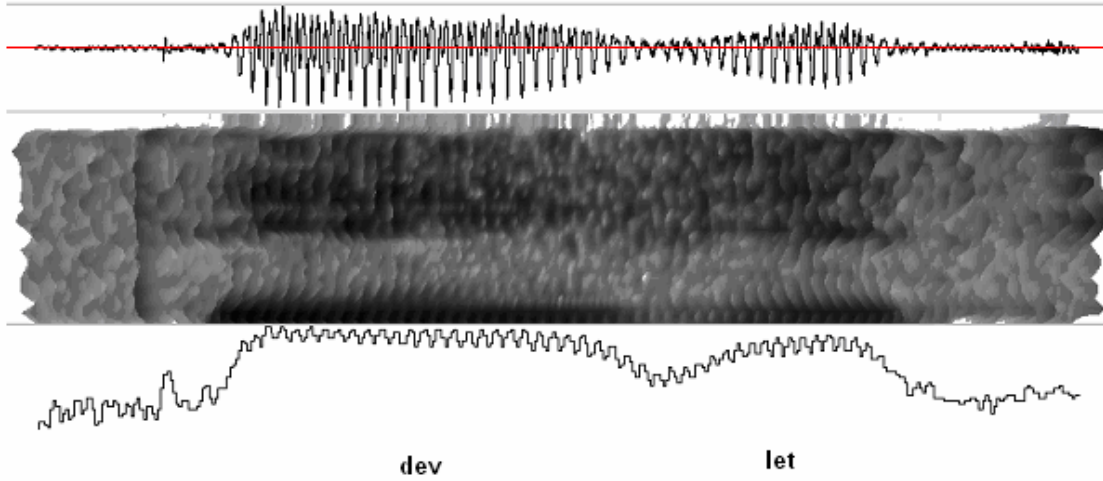
5.1.1 Hece Sınırlarının Belirlenmesi (Etiketleme)

Sesli ifade tanıma sistemi sürecinde eğitim verilerin hazırlanması büyük önem taşır. Bu tez kapsamında sesli ifade tanıma sisteminin eğitilmesi sırasında

kullanılacak veriler, el ile kesimleme yolu izlenerek elde edilmiştir. El ile kesimleme şu düzeylerde yapılabilir:

- Sözcük düzeyi: Sözcük sınırlarının belirlenmesinde spektrum ve sesli ifadenin ses dalgası kullanılabilir. Sinyalin genliğinin sıfır olduğu ve spektrumun beyaz olduğu kesimler sessizlik olarak, diğer kesimler ise sözcük olarak etiketlenir. Ses kartı aracılığıyla sesli ifade çalınarak da kesimleme işlemi gerçekleştirilebilir ya da kesimlemenin doğruluğu test edilebilir.
- Hece düzeyi: Hece sınırlarının belirlenmesinde de, sözcük düzeyinde olduğu gibi spektrum ve sesli ifadeye ait ses dalgası kullanılabilir. Sinyale ait genliğin sıfır olduğu kesimler sessizlik olarak etiketlenir. Bunun haricinde kalan kesimlerde ise, enerjinin yükseldiği bölgeler ünlü seslerin bulunduğu kesimleri yani heceleri ifade eder. Bu bilgiden yola çıkarak, enerjinin yükseldiği kesimler dinlenerek hecelerin gerçek sınırlarını el ile belirlemek mümkündür.
- Fonem düzeyi: Bu düzeyde daha çok sinyalin frekans dönüşümü kullanılır. Spektrumdaki değişiklikler fonem değişimini gösterir. Bu düzeyde de kesimlerin seslendirilmesi faydalı olabilir, ancak fonemlerin birbirini etkilemesi (*coarticulation*) sebebiyle yüksek doğrulukta kesimleme mümkün olmamaktadır. Yani fonem sınırları sözcüklerde olduğu gibi kolay bulunamaz.

Şekil 5.2'de *devlet* sözcüğüne ait dalga biçimi, 3 boyutlu spektrogram çizimi ve enerji değişimi verilmiştir. Spektrogramda yer alan koyu bölgeler ünlü harflerin bulunduğu konumları dolayısıyla hece birimlerinin varlığını göstermektedir. Aynı şekilde ünlü harflerin bulunduğu bölgelerde enerji değerleri yüksek olmaktadır. Hece konumlarının, bu şekilde spektrogramdan ve enerjiden de faydalanılarak bulunması mümkündür.



Şekil 5.2 devlet sözcüğüne ait dalga biçimi, 3 boyutlu spektrogram ve enerji değişimi grafikleri

Bu tez kapsamında hece düzeyinde bir kesimleme yapılmıştır. Kesimleme yapılırken geliştirilen ve tanımı verilen araç kullanılmıştır.

5.1.2 Hecelere Ait Özellik Vektörlerinin Çıkarılması

Özellik vektörlerinin çıkarılmasında *mel-cepstrum* yöntemi kullanılmıştır. MFCC katsayılarını içeren özellik vektörleri 256 örnekten oluşan çerçeveler (23 ms) için hesaplanmış ve çerçeveler yeni özellik vektörlerinin hesaplanması için sesli ifade sinyali üzerinde 64 örnek (5.8 ms) kaydırılmıştır. Her bir 256 örneklik pencere için 13 MFCC katsayısı hesaplanmış, böylece özellik vektörleri 13 boyutlu olarak belirlenmiştir.

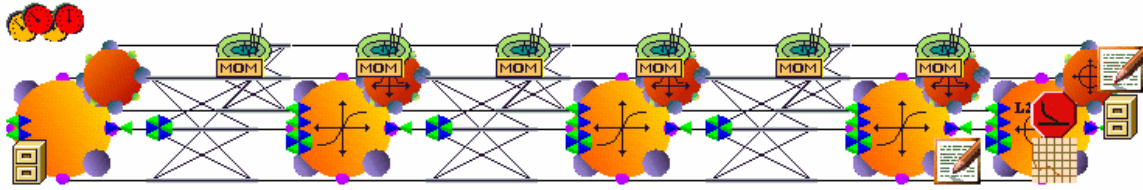
Tez kapsamında, heceler 1000 örnekten oluşan pencereler (90 ms) dahilinde işlenmiş, ve her bir pencerede 256'lık çerçeveler dahilinde heceyi oluşturan sesli ifade sinyalinin özellik vektörleri çıkarılmıştır. Sonuç olarak her bir hece 12 tane 13 boyutlu özellik vektörü ile ifade edilmiştir. 12*13 boyutlu (156) özellik vektörleri sınıflandırıcı sistem olan çok katmanlı yapay nöron ağlarına eğitmek üzere verilmeden önce tek boyutlu kolon matrisine dönüştürülmüştür. Bu şekilde her bir hece 156 değerli bir özellik vektörü ile ifade edilmiştir.

5.1.3 Sınıflandırıcı Sistem

Sesli ifade tanıma sistemlerinde en çok kullanılan tanıma yöntemi Çok Katmanlı Nöron Ağı (Multi Layer Perceptron) modelidir. Bu tez kapsamında da, hecelerin tanınması amacıyla çok katmanlı nöron ağı modeli kullanılmıştır.

Genel bir çok katmanlı nöron ağı mimarisinde, giriş katmanlı, ara katmanlar ve çıkış katmanı mevcuttur. Tez dahilinde oluşturulan çok katmanlı sinir ağı modellerinin giriş katmanı, sesli ifadeyi oluşturan hecelere ait özellik vektörlerinin sisteme verilmesi işlevini görür. Özellik vektörleri, daha önce de bahsedildiği üzere *MFCC* uygulanarak çıkarılmıştır. Çıkış katmanı ise, verilen sesli ifadede tanınacak olan hecenin sınıfını verir.

Tez kapsamında yapay nöron ağlarının modellenmesinde *Neuro Solutions* yazılımı kullanılmıştır. Şekil 5.3'te 15 hece için 2 katmanlı bir yapay sinir ağının bu yazılım ile modellenmesi sonucunda elde edilmiş çizimi verilmektedir:

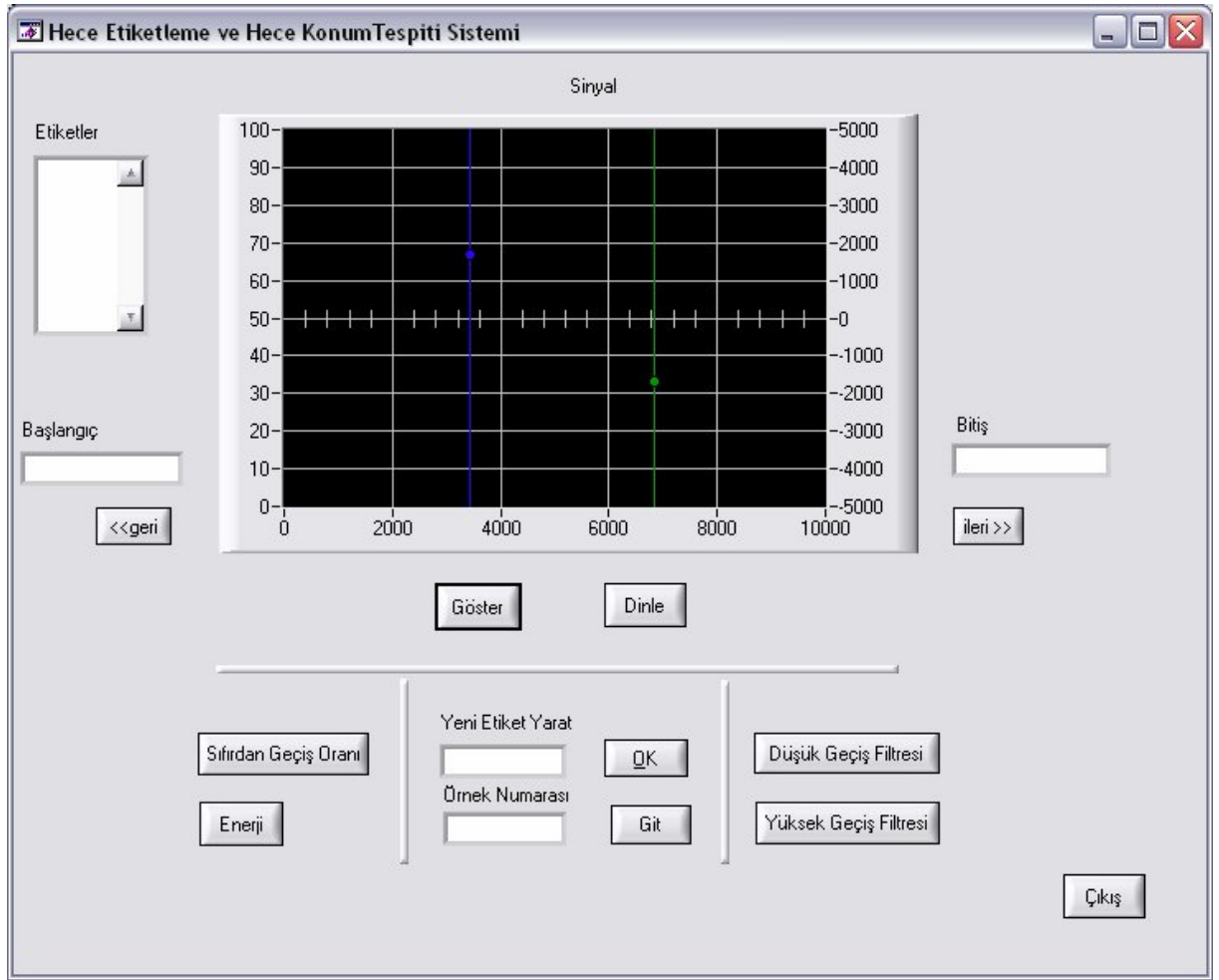


Şekil 5.3 . *Neuro Solutions* ile modellenmiş 2 katmanlı yapay nöron ağı mimarisi

5.2 Hece Etiketleme ve Hece Konum Tespiti Sistemi

Araç, *National Instruments CVI* ortamında geliştirilmiştir. Sesli ifade verisi olarak Kültür Bakanlığı'nın görme engelliler için hazırlamış olduğu sesli kitaplardan biri olan Montaigne'in *Denemeler* adlı eseri kullanılmıştır.

Geliştirilen araç ilk çalıştırıldığında elde edilen görüntüsü Şekil 5.4'te verilmiştir:



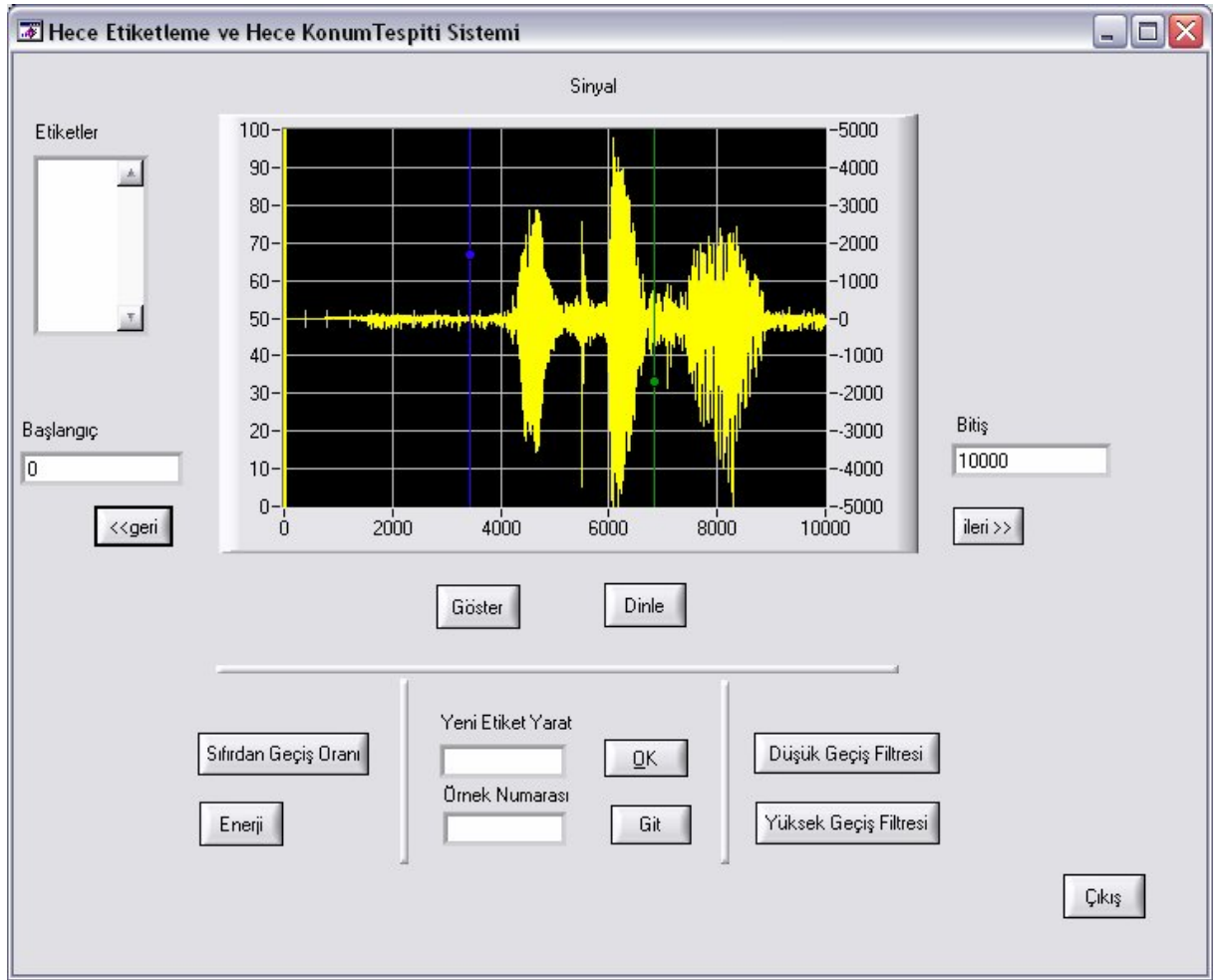
Şekil 5.4 Geliştirilen araç - Hece Etiketleme ve Hece Konum Tespiti Sistemi (HEKTİS)

Hece Etiketleme ve Hece Konum Tespiti Sistemi (HEKTİS), sesli ifadenin sınıflandırma sürecine hazırlık aşamasında üzerinde yapılması gereken birtakım ön işlemlere olanak sağlar. Bu ön işlemler;

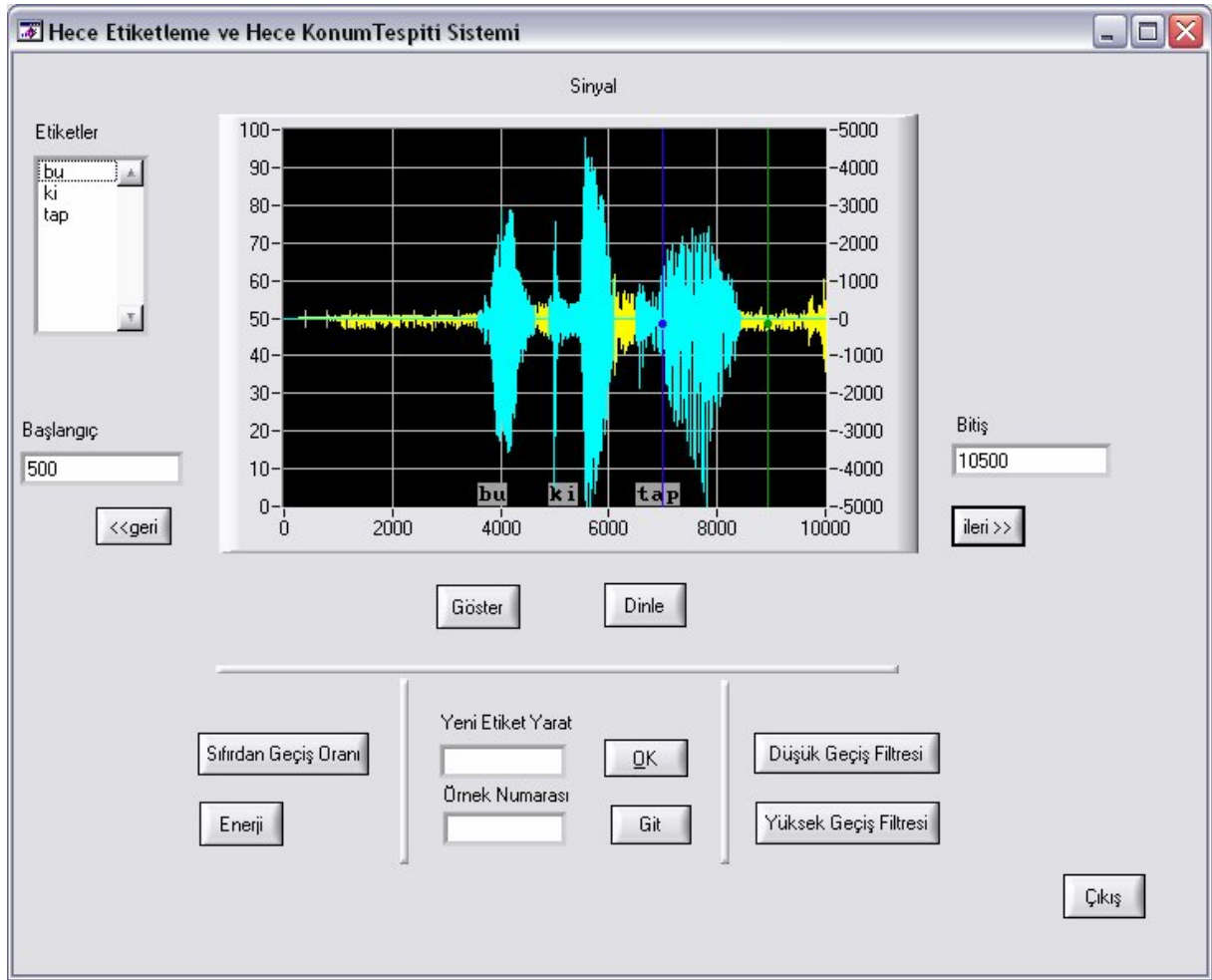
- Sesli ifadeye ait hecelerin etiketlenmesi,
- Görüntülenen sesli ifade sinyali üzerinde daha önceden etiketlenmiş heceleri konumları ile görüntüleyebilme,
- Sinyal üzerinde istenen konuma gitme,
- Görüntülenen sesli ifade sinyalinin ilgili kısmı üzerinde imleçler yardımıyla sınırları belirlenen sesli ifadenin dinlenmesi,

- Görüntülenen sesli ifade sinyalinin ilgili kısmına ait sıfırdan geçiş oranını gösterir grafiğin görüntülenmesi,
- Görüntülenen sesli ifade sinyalinin ilgili kısmına ait enerji grafiğinin görüntülenmesi,
- Görüntülenen sesli ifade sinyali üstünde Düşük Geçiş Filtresi (*Low Pass Filter*) uygulanması,
- Görüntülenen sesli ifade sinyali üstünde Yüksek Geçiş Filtresi (*High Pass Filter*) uygulanması işlemlerini içerir.

Görüntülenmesi istenen sesli ifade sinyali, *Göster* düğmesi aracılığıyla görüntülenebilir. Montaigne'in *Denemeler* adlı sesli kitabının *HEKTİS* yazılımı ile görüntülenmesi sırasında elde edilen yazılıma ait görüntü Şekil 5.5'te verilmiştir:



Şekil 5.5 HEKTİS kullanımı. Denemeler adlı esere ait sesli ifade sinyalinin görüntülenmesi

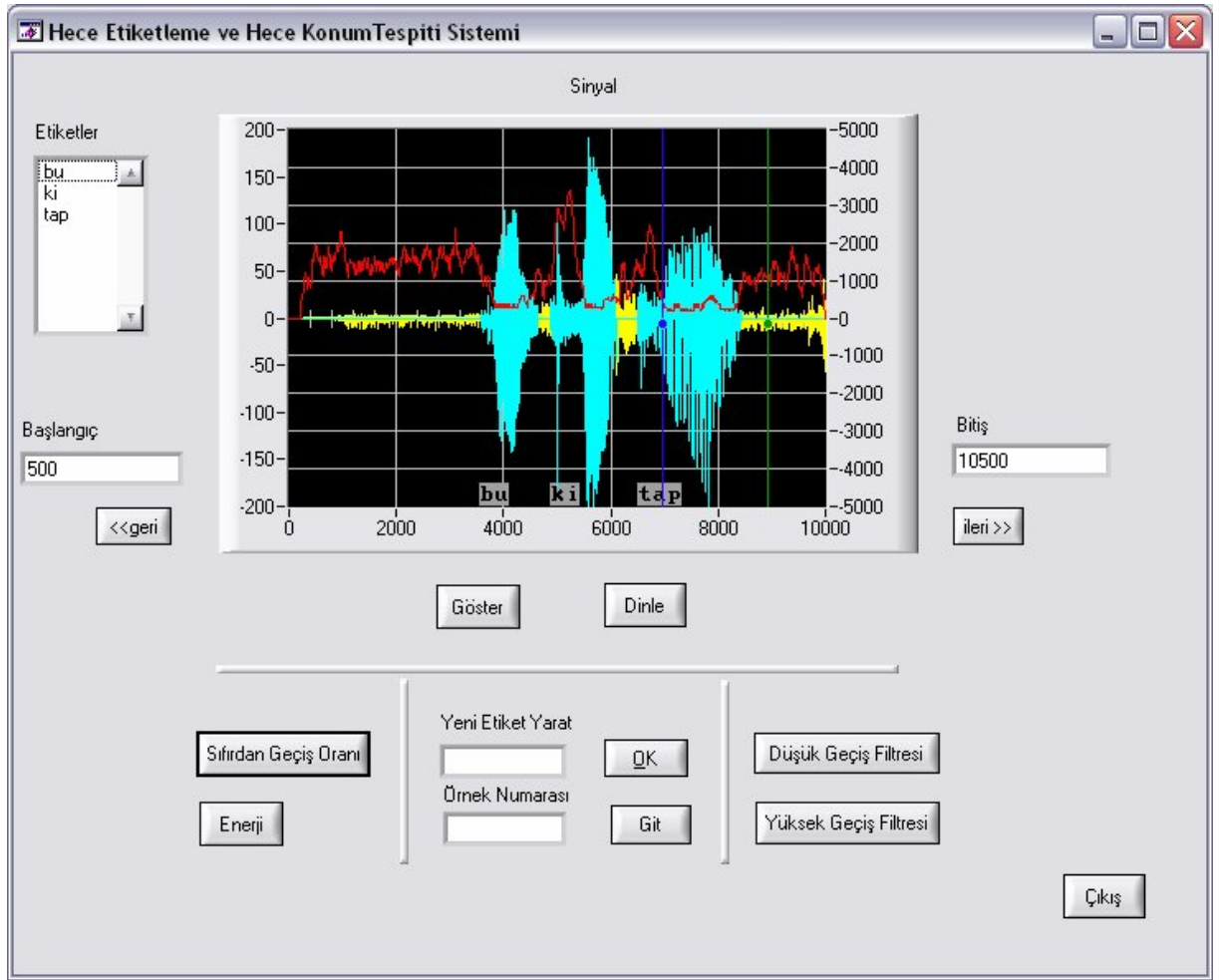


Şekil 5.6 HEKTİS'te hece etiketleme işlemi

Sinyal üzerinde farklı renklerle gösterilen imleçler yardımı ile sınırları işaretlenen sinyalin belli bir kısmı dinlenebilir ve bu şekilde sınırı belirlenen hece *Yeni Etiket Yarat* alanına yazılarak bir metin kütüğüne kaydedilir. Bu şekilde, *bu*, *ki* ve *tap* hecelerinin etiketlenmesi sonucunda elde edilen görüntü Şekil 5.6'da verilmiştir:

Görüldüğü üzere, hecelerin sinyal içindeki konumları ve ne oldukları bilgisi sinyalin üstünde görüntülenebilmektedir. Aynı zamanda, görüntülenen sesli ifade sinyalinde etiketlenmiş heceler de liste halinde solda bulunan *Etiketler* alanında verilmektedir.

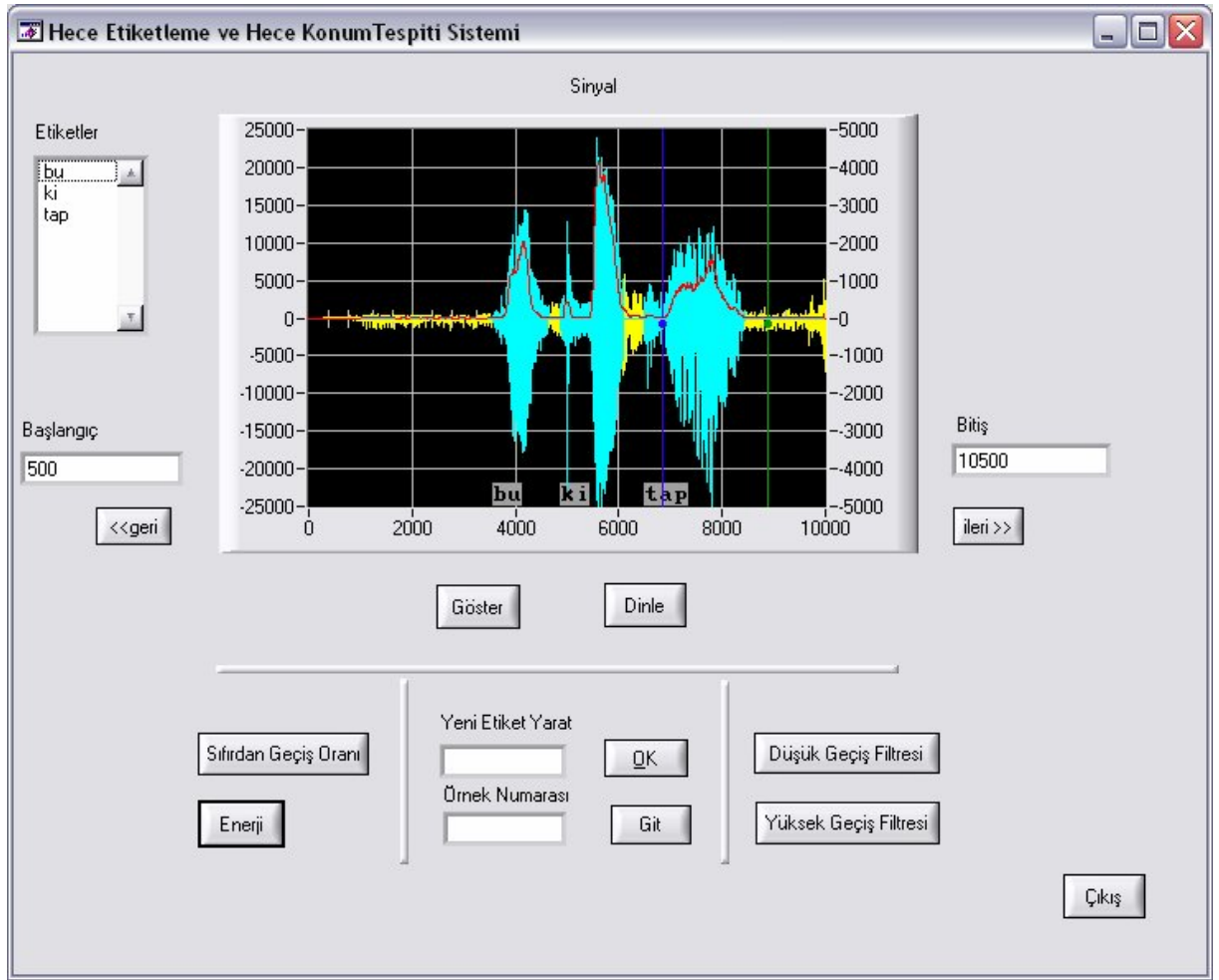
Görüntülenen sesli ifade sinyaline ait sıfırdan geçiş oranını (*zero crossing rate*) gösterir grafiğin görüntülenmesi ile elde edilen görüntü Şekil 5.7'de verilmiştir:



Şekil 5.7 HEKTİS'te sıfırdan geçiş oranını gösterir grafiğin görüntülenmesi

Görüldüğü üzere sessizliklerde, hece aralarında ve ünsüz harflerin bulunduğu alanlarda sıfırdan geçiş oranı büyük bir değere sahip olmaktadır. Ünlü harflerin söylendiği alanlarda ise, bu oran en düşük düzeyde olmaktadır.

Görüntülenen sesli ifade sinyaline ait enerji grafiğinin görüntülenmesi ile elde edilen görüntü ise Şekil 5.8'de verilmiştir:

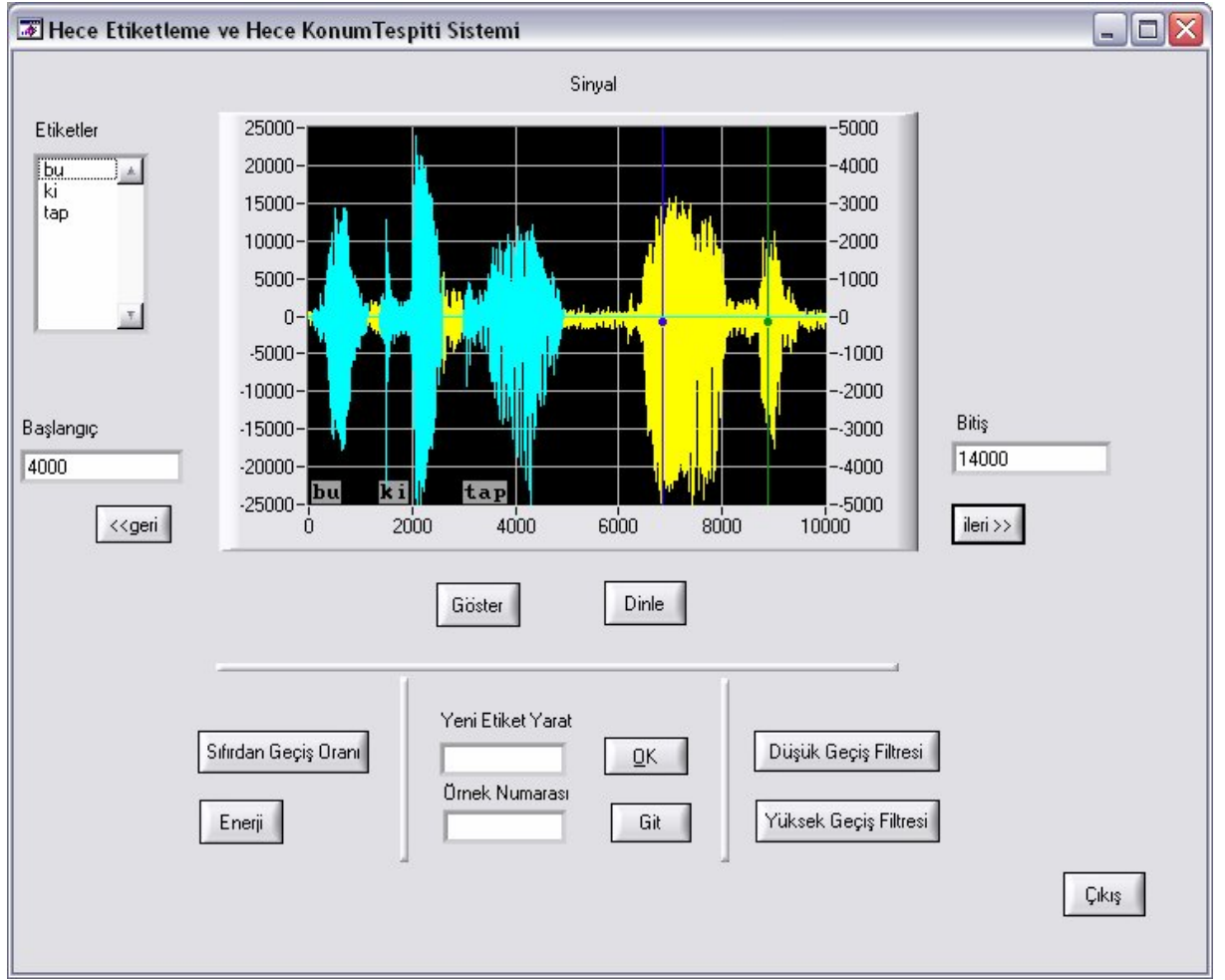


Şekil 5.8 HEKTİS'te enerji grafiğinin görüntülenmesi

Görüldüğü üzere enerji değeri ünlü harflerde en yüksek düzeye ulaşmakta, ünsüz harflerde ve sessiz alanlarda en düşük düzeyde olmaktadır.

Sesli ifade sinyaline ait kütükler çok büyük boyutlara sahip olabilmektedir. Bu açıdan sinyal işleme yazılımlarında performans kistası da göz önünde bulundurulmalıdır. HEKTİS yazılımının performansını yüksek tutmak adına hem kullanılan bellek açısından hem de diske erişim sayısı açısından önemli noktalara dikkat edilmiştir. Sistemin az bellek kullanmasını sağlamak amacıyla sesli ifade kütüğünün tamamı yerine sadece görüntülenen kısım bellekte tutulmaktadır. Ayrıca diske erişim sayısını da en düşük düzeyde tutmak amacıyla, *ileri* ve *geri* düğmeleri aracılığıyla bir sonraki sinyal penceresi görüntülenmek istendiğinde sinyalin sadece o an bellekte olmayan kısmı diskten çekilmektedir. Yani, 10000 örnek boyutunda olan bir sonraki sinyal penceresinin tamamının tekrar diskten çekilmesi yerine, sadece kaydırma miktarı olan 500 örnek diskten çekilmektedir.

Şekil 5.9'da *ileri* düğmesi ile sinyalin ilerleyen kısımlarının görüntülenmesine ait HEKTİS yazılımı görüntüsü verilmektedir:



Şekil 5.9 HEKTİS yazılımında sinyal üzerinde ilerlenmesi

Sinyal üzerinde *ileri*, *geri* düğmeleri ile ilerlemek dışında örnek numarası da verilerek istenen konuma gidilebilmektedir. *Başlangıç* ve *Bitiş* alanlarında sinyalin hangi örnekleri arasının görüntülendiği bilgisi gösterilmektedir.

Düşük Geçiş Filtresi ve *Yüksek Geçiş Filtresi* ile sesli ifade sinyali üstünde filtre işlemleri yapılabilir.

5.3 Deneyler

Türkçe hece birimlerini sınıflandırmak için, farklı sayılarda hece içeren farklı eğitim kümelerinden örnekler kullanılmıştır. Buradaki amaç, Türkçede mevcut olan tüm heceleri tanımaktan ziyade, tüm heceler içinden en sık kullanılan hecelerin oluşturduğu bir kümenin tanınmasını sağlamaktır. Farklı metin kütüklerinden en sık

kullanılan heceler incelendiğinde görülmüştür ki; en sık kullanılan yaklaşık 50 hece dahi Türkçedeki tüm hecelerin yaklaşık yarısını kapsamaktadır. Amaç, bu şekilde küçük bir hece kümesinin tanınmasıyla başlanan bir süreçte genellemeye gitmektir.

Bu tez dahilinde tek karakterli ve iki karakterli hecelerin tanınması amaçlanmıştır. İki karakterli hecelerin süre olarak uzun olmaları, farklı giriş sayısına sahip yapay sinir ağı modellerinin kullanılmasını gerektirmektedir. Montaigne'in *Denemeler* adlı eseri incelendiğinde görülmüştür ki; tek ve iki karakterli heceler tüm hecelerin %63'ünü oluşturmaktadır (Çizelge 5.1.). Bu demektir ki, bu hece kümesinin %100 başarıyla tanınması durumunda, %63'lük bir başarı oranına sahip bir sesli ifade tanıma sistemi elde edilecektir.

Hece Türü	Hece Sayısı	Geçiş Oranı
-----	-----	-----
Ü	88200	6.821
UZ	72427	5.601
ZU	660465	51.079
UZZ	3300	0.255
ZUZ	452468	34.993
ZZU	3030	0.234
ZUZZ	8240	0.637
ZZUZ	2217	0.171
ZUZZZ	243	0.019
ZZUZZ	415	0.032
ZZZUZ	156	0.012
NA	1874	0.145

Çizelge 5.1 Montaigne'in Denemeler adlı eserinde geçen hece türleri (U: ünlü harf, Z: ünsüz harf) ve bu hecelere ait sayı ve oranlar

Elde edilen istatistikleri desteklemek amacıyla çok daha geniş bir metin kümesi farklı gazetelerin arşivlerinden elde edilmiştir. Bu arşiv de, *Denemeler* adlı eseri destekler niteliktedir. Tek ve iki karakterli heceler, tüm heceler içinde %62'lik bir paya sahiptir (Çizelge 5.2.).

Hece Türü	Hece Sayısı	Geçiş Oranı
-----	-----	-----
U	2234978	5.064
UZ	2218630	5.027
ZU	23076088	52.282
UZZ	134266	0.304
ZUZ	15591250	35.324
ZZU	102679	0.233
ZUZZ	503822	1.141
ZZUZ	99484	0.225
ZUZZZ	8418	0.019
ZZUZZ	12895	0.029
ZZZUZ	8819	0.020
NA	146528	0.332

Çizelge 5.2 Gazetelerden elde edilen arşivde geçen hece türleri ve bu hece türlerine ait sayı ve oranlar

Denemeler adlı eser dahilinde 1.270.930 toplam heceye rastlanırken, bu hecelerden en sık karşılaşılan tek ve iki karakterli ilk 50 hecesi ele alındığında, bu hece grubunun 560.277 toplam sayısı ile %44'lük bir paya sahip olduğu görülmektedir. Bu demektir ki, farklı hece barındıran bu küçük hece kümesinin bile tanınmasının gerçekleştirilmesi ile, verilecek bir sesli ifadenin neredeyse yarısı tanınabilecektir. Bundan sonra tanınan hece grubuna eklenen hecelerin performanstan götördükleri farka oranla tanımaya getirecekleri başarı oranı giderek azalacaktır. Bu noktada, yitirilen performansla oranla başarı oranına

getirilecek katkının küçüklüğü göz önüne alındığında, uygulama kapsamında hece sayısının artırılması "*değer mi?*" sorusu önem kazanır.

Çizelge 5.3.'te referans alınan Montaigne'in *Denemeler* adlı eserinde geçen Türkçe hecelerden en sık karşılaşılan tek ve iki karakterli 50 heceye ait rastlanma sıklıkları verilmiştir.

	Hece	Sayı
	-----	-----
1	la	32720
2	le	30261
3	de	25520
4	da	20927
5	a	19770
6	ya	19679
7	o	19640
8	i	18910
9	di	18655
10	bir	18421
11	ma	18359
12	ri	15377
13	ne	13361
14	ra	12979
15	rl	12846
16	bi	12630
17	ve	12604
18	ka	12274
19	ki	11474

20	lar	11411
21	ni	10996
22	na	10937
23	li	10644
24	du	10625
25	me	10271
26	sa	10128
27	e	9904
28	ler	9459
29	u	9450
30	dl	9386
31	nl	9246
32	re	9167
33	si	8919
34	te	8863
35	bu	8713
36	ye	8701
37	ta	8541
38	ll	8130
39	ol	8066
40	yor	8031
41	ge	7968

42	sl	7787
43	ti	7742
44	ce	7389
45	den	7234
46	dan	6481
47	Gi	6362
48	nu	6240
49	gi	6214
50	se	6127
51	ba	5775
52	Gl	5756
53	tl	5752
54	ha	5662
55	dU	5499
56	lan	5384
57	ken	5363
58	mi	5219
59	nln	5141
60	Cin	4747
61	rin	4694
62	lu	4687

Çizelge 5.3 Montaigne'in Denemeler adlı eserinde en sık karşılaşılan, tek veya iki karakterden oluşan ilk 50 heceye ait rastlanma sıklıkları

Bu sıklıklar farklı gazete arşivlerinden elde edilen istatistiklerle karşılaştırıldığında hemen hemen aynı sonuçların varlığı gözlemlenerek ispatlanmıştır (Çizelge 5.4.).

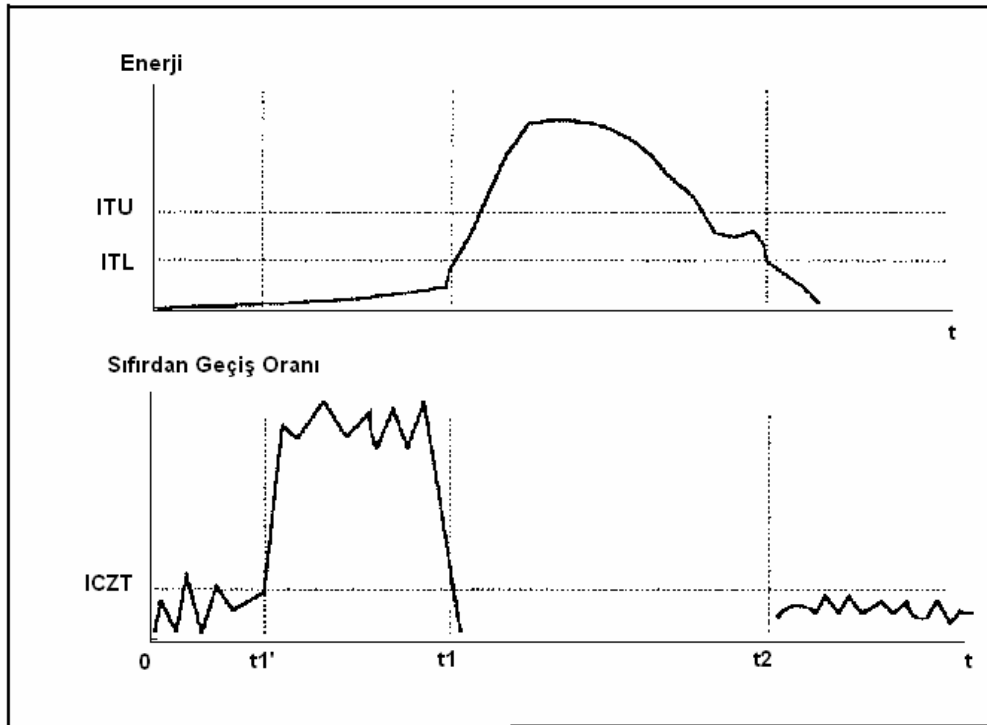
	Hece	Sayı
	-----	-----
1	la	1008010
2	le	952203
3	de	848172
4	da	733955
5	ya	681083
6	i	629565
7	ma	610199
8	di	608240
9	a	595075
10	ki	571590
11	li	547574
12	ka	524175
13	ra	501999
14	ri	493178
15	ye	486717
16	yUz	431247
17	o	410076
18	bir	400117
19	ta	389435
20	ve	386466
21	ne	384813
22	si	384594
23	bin	373411
24	rl	370788
25	me	364623
26	na	351190
27	ti	348170
28	sa	332398
29	te	326975
30	ni	324037
31	nl	318010
32	sl	316945
33	tl	314219
34	re	312085
35	bu	294963
36	se	280685
37	ge	279925
38	dl	272423
39	ll	265713
40	e	261869
41	lar	258709
42	mi	254869
43	ba	249545
44	ha	243904
45	bi	243551
46	yor	234002
47	ler	221433
48	Gi	220792
49	Gl	210116
50	do	198970
51	ce	197117
52	nIn	196913
53	nin	196227
54	lan	192458
55	al	185614
56	du	177227
57	ko	176770
58	nu	176497
59	gi	167761

Çizelge 5.4 Farklı gazetelerden elde edilen arşivde en sık karşılaşılan, tek veya iki karakterden oluşan ilk 50 heceye ait rastlanma sıklıkları

Sesli ifade tanıma sisteminin geliştirilmesi sürecinde, gazetelerden elde edilen arşivin seslendirilmiş hallerinin elde edilememesi nedeniyle, sesli kitabı bulunan Montaigne'in *Denemeler* adlı eseri tercih edilmiştir. Sık kullanılan heceler göz önüne alındığında kitap, büyük gazete arşivinin küçük bir temsili niteliğindedir.

Sesli ifade tanıma sistemlerine ait önemli bir zorluk, bir eğitim kümesi oluşturma aşamasıdır. Bu aşamada, sesli ifade hangi birim üstünden tanınacaksa o birime ait sınırlar bulunmalı ve bu sınırlar dahilindeki sesli ifade verisine ait özellik vektörleri çıkarılmalıdır. Bu amaçla, veriler el ile hazırlanabileceği gibi farklı algoritmalar kullanılarak bu işlem otomatik olarak da (*automatic segmentation*) gerçekleştirilebilir. Bu şekilde zamandan önemli ölçüde kazanılmaktadır.

Bu düşünceden yola çıkarak, hece tanıma sistemini gerçekleştirme sürecinde, ilk olarak hece sınırlarının bulunmasına yönelik farklı algoritmaların geliştirilmesi işlemi gerçekleştirilmiştir. Bu aşamada, L.R. Rabiner ve M.R. Sambur'a (Rabiner, Sambur, 1974) ait sesli ifade bitiş noktası algoritması (*speech endpoint algorithm*) üzerinden yola çıkarak otomatik hece ayırma işleminin farklı denemeler ile gerçekleştirilmesi denenmiştir.

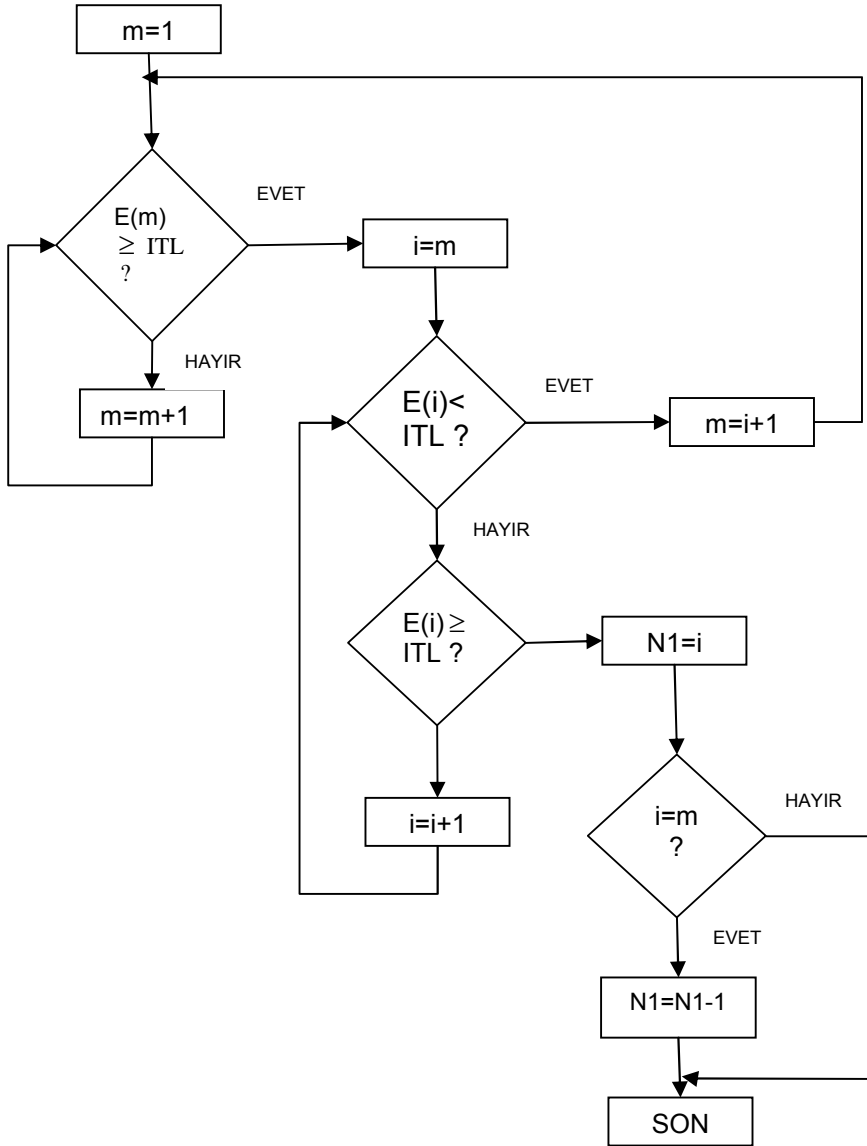


Şekil 5.10 Bir sözcüklük sesli ifade sinyali için örnek enerji ve sıfırdan geçiş oranı ölçümü

Şekil 5.10'da sözcüğün başlangıç ve bitiş noktalarının bulunmasına ilişkin *Rabiner ve Sambur*'un (Rabiner, Sambur, 1974) enerji ve sıfırdan geçiş oranı (*zero crossing rate*) fonksiyonlarının değişimlerini kullanan başlangıç-bitiş noktaları belirleme yöntemi verilmiştir. Bu yöntemle göre, sinyalin enerji ve sıfırdan geçiş oranı fonksiyonları üstünde birtakım değerler belirlenir ve bu değerlerin aşılması durumunda sınır noktaları belirlenir. Enerji fonksiyonu üzerinde *ITL* (*Interval Threshold Low*) adlı değer aşıldığında t_1 başlangıç noktası, *ITL* değerinin altına düştüğünde ise t_2 bitiş noktası belirlenir. Sözcüklerin kesin başlangıç bitiş noktalarının belirlenebilmesi için, enerji fonksiyonunun yanında sıfırdan geçiş oranı değişimleri de dikkate alınır. Sıfırdan geçiş oranı fonksiyonu üzerinde, *ICZT* (*Interval Zero Crossing Threshold*) olarak adlandırılan düzeyin geçildiği nokta (t_1'), sesli ifade sinyalinin t_1 yerine yeni başlangıç noktası olur. Sinyal üzerindeki gürültüden dolayı bu noktanın kolayca belirlenmesi çoğu zaman mümkün olmamaktadır. Bu sebeple, *ICZT* düzeyinin, belli bir süre içinde, ardı ardına, örneğin en az üç kez aşılması koşulu aranarak başlangıç noktası belirlenebilir.

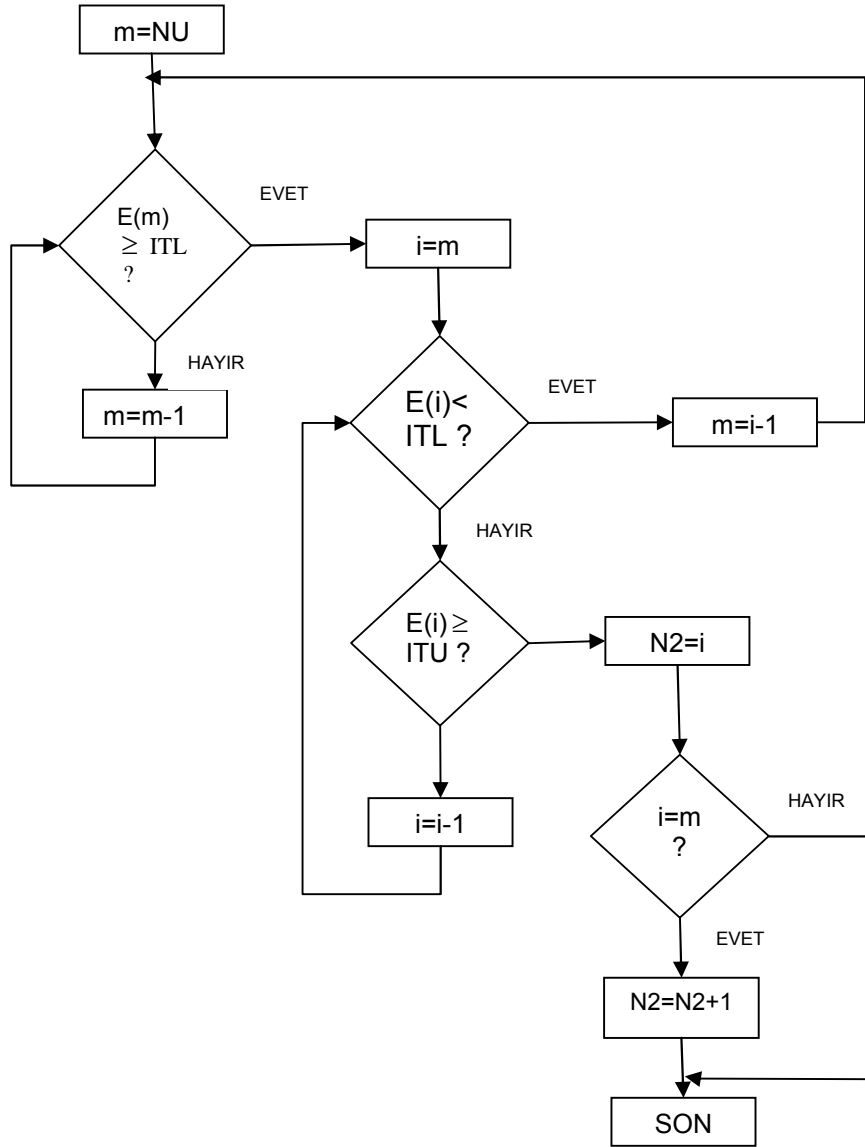
Başlangıç ve bitiş noktalarını belirleme yönteminin iyileştirilebilmesi için, *ITL* olarak adlandırılan tek bir eşik düzeyi değeri yerine, *ITL* ve *ITU* (*Interval Threshold Upper*) olarak adlandırılan iki farklı eşik düzeyi kullanılabilir. Eğer sesli ifade sinyalinin genlik düzeyi düşükse *ITL*, değilse *ITU* eşik değeri kullanılabilir. Sinyal üzerindeki gürültüden dolayı bu eşik değerleri de yanlış belirlenebilmektedir. *ICZT* eşik değerinde olduğu gibi, bu eşik değerlerinde de ardı ardına n nokta için geçilmesi şartı aranabilir.

Rabiner ve Sambur'un başlangıç ve bitiş noktalarının bulunması için geliştirdikleri algoritmanın başlangıç noktası için uyarlanmış hali Şekil 5.11'de verilmiştir.



Şekil 5.11 . *Rabiner* ve *Sambur*'un sesli ifade sinyalinin başlangıç noktasını bulmak için geliştirdikleri algoritmaya ait akış çizgesi

Aynı algoritmayı, bitiş noktası için de uyarlamak mümkündür (Şekil 5.12).



Şekil 5.12 . Rabiner ve Sambur'un sesli ifade sinyalinin bitiş noktasını bulmak için geliştirdikleri algoritmaya ait akış çizgesi

Rabiner ve Sambur, sinyale ait başlangıç ve bitiş noktasını tespit etmek amacıyla bu algoritmayı geliştirmişlerdir. Bu tez kapsamında, bu algoritma hece sınırlarını tespit etmek amacıyla uyarlanıp kullanılmıştır. Montaigne'in *Denemeler* adlı eserine ait sesli kitap üzerinde uygulandığında %44'lük bir başarı elde edilmiştir. Sonuçlar incelendiğinde görülmüştür ki; hataların önemli bir kısmı patlamalı ünsüzler olan /b/, /d/, /g/, /p/, /t/, /k/ 'nin varlığının tespit edilememesinden kaynaklanmaktadır. Dolayısıyla bu harflerden biri ile başlayan hecelerde patlamalı ünsüzden sonra gelen ünlüden itibaren o hecenin başladığı belirlenebilmektedir.

Hece tabanlı sesli ifade tanıma sisteminin gerçekleştirilmesi aşamasında ikinci olarak, sürekli sesli ifade üzerinden daha önce sınıflandırıcı sisteme öğretilen hecelerin tanınmasına çalışılmıştır. Bu amaçla, sesli ifade üzerinde 1000 örneklik (90 ms) bir pencere 60 örnek (5.4 ms) kaydırılarak, o anki pencere dahilindeki sesli ifadenin hangi heceye karşılık geldiği daha önce eğitilen sınıflandırıcı sistem yardımıyla bulunmaya çalışılmıştır. Farklı sayıdaki hecelerin farklı özelliklerdeki çok katmanlı nöron ağlarınca tanınmasına ilişkin istatistiki sonuçlar Çizelge 5.5'te verilmiştir.

Yöntem	MLP	MLP	MLP	MLP
Tanınan Farklı Hece Sayısı	42	42	42	55
Sınıflandırma Başarı Oranı	20/32	7/18	21/32	6/19
Sınıflandırma Başarı Yüzdesi	%62.5	%38.8	%65.6	%31.5
Katman Sayısı	4	5	3	6
Katmanlardaki Nöron Sayısı	156,42,42,42,42,42	156,42,42,42,42,42,42	156,42,42,42,42	156,50,50,50,50,50,50,55
İterasyon Sayısı	5151	16040	3332	72134
Eğitim Süresi	28 dakika	1 saat 58 dakika	13 dakika 28 saniye	18 saat 29 dakika

Çizelge 5.5 Geliştirilen sesli ifade tanıma sisteminin, sürekli sesli ifade sinyali ile farklı heceler için farklı özellikteki nöron ağları ile kullanımı

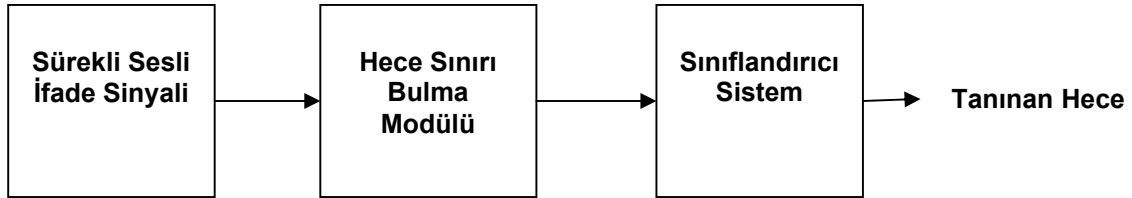
Çizelge 5.5'te verilen deneyler heceler için herhangi bir sınır bulma modülü kullanılmadan doğrudan sürekli sesli ifade sinyali üzerinden gerçekleştirilmiştir. Bunun için sürekli sesli ifade üzerinde bir pencere kaydırılmış ve bu pencerenin o an içerdiği sesli ifadenin bir hece içerip içermediği, eğer bir hece içeriyorsa bu hecenin ne olduğu sınıflandırıcı sistem aracılığıyla bulunmuştur. Sürece dahil işlem adımlar Şekil 5.13'te verilmiştir.



Şekil 5.13 Sürekli sesli ifade üzerinden sınıflandırıcı sistem aracılığı ile hecelerin tanınması

Elde edilen sonuçlardan, kullanılan çok katmanlı sinir ağlarının özelliklerinin (katman sayısı, katmanlardaki nöron sayıları vs) sesli ifade tanıma sisteminin başarısını nasıl etkilediği gözlemlenmiştir. Ve katman sayısının artmasının sistemin başarısına doğru oranda bir etki yapmadığı görülmüştür. 42 farklı hecenin 3 katmanlı bir nöron ağıyla en iyi oranda tanınabileceği, 55 farklı hece için ise daha karmaşık yapıda nöron ağlarının kullanılması gerektiği sonucuna varılmıştır.

Hece tabanlı sesli ifade tanıma sisteminin gerçekleştirilmesinde son aşama olarak, bir hece sınırı bulma modülünün var olduğu varsayımına dayanarak sesli ifadeden ayrıştırılmış hecelerin sınıflandırıcı sistem aracılığıyla hangi sınıfa dahil olduklarını bulma yoluna gidilmiştir. (Şekil 5.14)



Şekil 5.14 Sürekli sesli ifadeye ait ayrık hecelerin sınıflandırıcı sistem aracılığı ile tanınması

Farklı sayıdaki hecelerin farklı özelliklerdeki çok katmanlı nöron ağlarınca tanınmasına ilişkin istatistiki sonuçlar Çizelge 5.6.'da verilmiştir.

Yöntem	MLP	MLP	MLP	MLP	MLP	MLP
Taninan Farklı Hece Sayısı	5	10	15	20	25	30
Eğitim Kümesi Boyu	134	249	399	537	692	833
Deneme Kümesi Boyu	50	66	82	88	84	85
Sınıflandırma Başarı Oranı	45/50	53	54	52	41	28
Sınıflandırma Başarı Yüzdesi	%90	%80.3	%65.8	%59.1	%48.8	%33
Ara Katman Sayısı	1	2	1	3	3	4
Katmanlardaki Nöron Sayıları	156,5,5	156,8,8,10	156,18,15	156,24,24,24,20	156,25,25,25,25	156,30,30,30,30,30
Kullanılan Transfer Fonksiyonu	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
İterasyon Sayısı	95	830	336	1804	1915	2468
Eğitim Süresi	2 saniye	4 dakika 36 saniye	18 saniye	7 dakika 42 saniye	13 dakika 10 saniye	27 dakika 15 saniye

Çizelge 5.6 Geliştirilen sesli ifade sisteminin farklı sayıda ayırık heceler için farklı özellikteki nöron ağı ile kullanımı

Deneyin bu aşamasında farklı hece sayıları için farklı özellikteki nöron ağıları kullanılmıştır. Hece sayısının artmasıyla birlikte elde edilen başarı oranı düşmüştür. Ve tanınması istenen farklı hece sayısının artmasının daha karmaşık nöron ağıları kullanılmasını gerektirdiği sonucuna varılmıştır.

6 SONUÇ, TARTIŞMA ve ÖNERİLER

Tez kapsamında hece tabanlı bir Türkçe sesli ifade tanıma sisteminin tasarımı ve gerçekleştirimi yapılmıştır. Bu sürece ait her adımın sonunda yeni bir sonuç elde edilmiştir.

Tez çalışması sürecinde öncelikle sesli ifade tanıma sistemlerinin ana zorluklarından biri olan veri kümesi hazırlama aşamasına dair çalışmalar yapılmıştır. Bu amaçla, bir hece sınırı bulma algoritması geliştirilmiştir. İkinci aşamada sınırları bulunan hecelere ait özellik vektörlerinin çıkarılmasına dair deneyler yapılmıştır. Bu noktada, özellik vektörlerinin boyutlarının değişmesinin tanımaya olan başarımına etkisi incelenmiştir. Hazırlanan veri kümesine ait özellik vektörlerinin sesli ifade tanıma sistemine öğretilmesi amacıyla çok katmanlı nöron ağları kullanılmıştır. Deneylerde farklı mimarilerdeki çok katmanlı nöron ağları kullanılmıştır. Ve görülmüştür ki; öğretilen farklı hece sayısının artması çok daha karmaşık çok katmanlı nöron ağlarının kullanımını gerektirmektedir. Bu da sesli ifade tanıma sisteminin başarısını olumsuz yönde etkilemektedir. Bu yüzden hece sayısının artması farklı özellikteki nöron ağlarının kullanımını gerektirmektedir.

Deney aşamasında iki farklı yol ile sistemin başarımı test edilmiştir. İlk olarak tanınacak sesli ifadeye ait hecelere yönelik bir sınır bulma işlemi yapılmadan, doğrudan sürekli sesli ifade üzerinde bir hece tanıma işlemi gerçekleştirilmiştir. Ve sürekli sesli ifade tanıma sistemi üzerinde bir pencere kaydırılarak, bu pencere dahilindeki sesli ifadede bir hece yer alıp almadığı, eğer bir hece var ise bu hecenin ne olduğu daha önceden eğitilen çok katmanlı nöron ağı ile öğrenilmiştir. İkinci deneyde ise sürekli sesli ifade üzerinden el ile sınırları bulunmuş hecelerin hangi sınıfa dahil oldukları daha önceden eğitilen çok katmanlı nöron ağı ile öğrenilmiştir. Sonuç olarak görülmüştür ki; ayırık hecelerin sesli ifade tanıma sistemince tanınmasına ilişkin başarı sürekli sesli ifade üzerinden yapılan tanımaya göre daha düşük olmuştur. Bu da hecelerin el ile sınırlarının bulunmasının etkili bir yöntem olmadığını, sesli ifade tanıma sistemlerinde otomatik olarak hece sınırı bulan bir modülün kullanılmasının önemini göstermektedir.

Yapılan çalışmada Türkçe için kullanılabilir hece sayısının 3500 civarında olması ve bahsedilen ilk 3 hece kalıbının (Ü, ÜZ, ZÜ) tüm heceler içinde dahi rastlanma sıklığına ait oranın %63 olması hece tabanlı sesli ifade tanıma yaklaşımının önemini

ortaya çıkarmaktadır. Bu demektir ki, sadece bu 3 tür hece grubunun sisteme tanıtılmasıyla bile, %63'lük başarıya sahip bir sesli ifade tanıma sistemi elde etmek mümkün olabilecektir. Sesli ifade tanıma sistemini modüllerden oluşan bir bütün olarak ele alacak olursak, bu tez kapsamında sadece bahsi edilen sesli ifade tanıma sisteminin hece tabanı geliştirilmeye çalışılmıştır. Sesli ifade tanımadaki tamamlanması gereken bir diğer aşama ise %37'lik değinilmeyen kesimin sonraki aşamalarda sözcük üretme ve doğrulama algoritmaları ile kapatılmasıdır. Yapılmasının gerekli olduğu düşünülen bir diğer çalışma ise, hecelerden sözcüklere geçiş ve düzeltmelere yönelik olmasıdır.

Gelecek çalışma olarak, etkili bir hece sınırı bulma algoritması ile, geliştirilen sesli ifade tanıma sisteminin başarımı artırılabilir. Tez kapsamında çalışılması gereken bir diğer konu ise, boyutları sabit olmayan hecelerin sesli ifade tanıma sistemine öğretilmesidir. Böylece sadece tek ve iki karakterli hecelerin değil, ikiden fazla karakter içeren hecelerin de tanınması mümkün olabilecektir.

7 KAYNAKLAR

Arkın, E., 2004. Türkçe Sesbirimlerinin Sınıflandırılması için Bir Bulanık Sinir Ağı'nın Tasarımı ve Gerçekleştirimi, Master Tezi. Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara.

Artuner, H., 1994. Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve Gerçekleştirimi, Doktora Tezi. Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara.

Bülbül, H. İ., Karacı, A., 2007. Bilgisayar Ortamında Sesli Komutları Tanıma: Örüntü Tanıma Yöntemi, Kastamonu Eğitim Dergisi, Cilt 15, No 1, 45-62s.

Demircan, Ö., 1979. Türkiye Türkçesinin Ses Düzeni Türkiye Türkçesinde Sesler, Ankara

Ergenç, İ., 1989, Türkiye Türkçesinin Görevsel Sesbilimi, Engin Yayınları.

Ganapathiraju, A., Hamaker, J., Picone, J., Ordowski, M., Doddington, G. R., 2001. Syllable-Based Large Vocabulary Continuous Speech Recognition. IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 4.

Halliday, D., Resnick, R., 1981. Fundamentals of Physics, John Wiley and Sons.

İkizler, N., Çavdar, İ.H., Ekim, G., 2005. Türkçede Konuşmacıdan Bağımsız Ayrık Hece Tanıma Sistemi. Elektronik Mühendisliği Bölümü, Karadeniz Teknik Üniversitesi, Trabzon.

Huang, X., Acerd, A., Hon, H., 2001. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall.

Mengüşoğlu, E., 1999. Bir Türkçe Sesli İfade Tanıma Sisteminin Kural Tabanlı Tasarımı ve Gerçekleştirimi, Master Tezi. Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara.

Nagarajan, T., Murthy, A., Hegde, R. M., 2003. Segmentation of Speech into Syllable-Like Units. Department of Computer Science and Engineering, Indian Institute of Technology, Madras.

Parson, T., 1986. Voice and Speech Processing, McGraw-Hill.

Rabiner, R. L., Sambur, M. R., 1974. An Algorithm for Determining the Endpoints of Isolated Utterances. The Bell System Technical Journal.

Savcı, F. Y., 1994. Sesli İfade Tanıma için Otomatik Bir Özellik Çıkarımı Dizgesinin Tasarımı ve Gerçekleştirimi, Master Tezi. Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara.

Sever, H., 2000. Dilbilim ve Genel Tarihine Genel Bakış. <http://www.caginpulisi.com.tr/57/44-45.htm>

Somervuo, P., Kurimo, M., 1996. Using the Self-Organizing Map to Speed up the Probability Density Estimation for Speech Recognition with Mixture Density HMMs. International Conference on Spoken Language Processing.

Tebelskis, J., 1995. Speech Recognition Using Neural Networks, Master Tezi. School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Yu., K., 2006. Adaptive Training for Large Vocabulary Continuous Speech Recognition, Doktora Tezi. Hughes College and Cambridge University Engineering Department.

Zurada, J. M., 1992, Introduction to Artificial Neural Systems, West Publishing Company, St. Paul.

ÖZGEÇMİŞ

Adı Soyadı : Burcu CAN

Doğum Yeri : Erzurum

Doğum Yılı : 1983

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise 1997-2001: Süleyman Demirel Anadolu Lisesi, ANKARA

Lisans 2001-2005: Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, ANKARA

Yabancı Dil:

İngilizce : Anlama ve okuma çok iyi. Konuşma iyi.

Fransızca : Başlangıç düzey

İş Tecrübesi:

2005-2007 : Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara
Araştırma Görevlisi

