

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

135963

METİNDEN BAĞIMSIZ KAPALI SET KONUŞMACI SAPTAMA

YÜKSEK LİSANS TEZİ

Yavuz Selim FATİHOĞLU

135963

Anabilim Dalı: Elektronik Bilgisayar Eğitimi

Danışman: Yard. Doç. Dr. Melih İNAL

**TC. YÜKSEKÖĞRETİM KURULU
DÜŞÜNME MERKEZİ**

OCAK 2003

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

METİNDEN BAĞIMSIZ KAPALI SET KONUŞMACI SAPTAMA

YÜKSEK LİSANS TEZİ

Yavuz Selim FATİHOĞLU

Tezin Enstitüye Verildiği Tarih : 13.01.2003

Tezin Savunulduğu Tarih : 24.01.2003

Tez Danışmanı

Yard. Doç. Dr. Melih İNAL

(.....)

Üye

Yard. Doç. Dr. İsmail ERTÜRK

(.....)

Üye

Yard. Doç. Dr. A. Turan ÖZCERİT

(.....)

OCAK 2003

**TC. YÜKSEK ÖĞRETİM BAKANLIĞI
DOKÜMANİSYON MERKEZİ**

METİNDEN BAĞIMSIZ KAPALI-SET KONUŞMACI TANIMA

Yavuz Selim FATİHOĞLU

Anahtar Kelimeler: Genetik Algoritmalar, Benzetilmiş Tavlama, Doğrusal Öngörülü Kodlama, Konuşmacı Tanıma, Yapay Sinir Ağları, Kendi Kendini Organize Eden Ağlar (SOM)

Özet: Bu tez çalışmasında, geleneksel özellik çıkartım yöntemleri ve genetik algoritmalar benzetilmiş tavlama (GABT) yöntemi ile yapay sinir ağlarına dayalı konuşmacı tanıma uygulaması gerçekleştirilmiştir.

Özellik çıkartım vektörlerinin elde edilmesi aşamasında Mel-frekans ölçekli kepsral katsayılar yöntemi ve GABT gibi farklı iki yöntem uygulanmıştır. Elde edilen özellik vektörleri Kendi Kendini Organize Eden Ağlar (SOM) 'ın eğitiminde kullanılmıştır. SOM konuşma tanıma uygulamasının bir sonraki aşaması olan sınıflandırmada kullanılmıştır.

Yapılan çalışmada, 5 erkek, 5 bayan, toplam 10 konuşmacının yer aldığı ve çeşitli rakamları farklı oturumlarda iki kez telaffuz ettikleri CSLU (Center for Spoken Language Understanding) veritabanına dayalı konuşmacı seti kullanılmıştır. Bu telaffuzlardan 10 tanesinin eğitim, 10 tanesinin de test aşamasında kullanılması amaçlanmıştır. Çalışma Metinden Bağımsız Kapalı Set Konuşma Saptama uygulamasıdır.

Sonuç olarak özellik çıkartım aşamasında GABT , Mel frekans ölçekli kepsral katsayılara göre daha iyi sonuç verdiği görülmüştür.

TEXT INDEPENDENT CLOSED-SET SPEAKER IDENTIFICATION

Yavuz Selim FATİHOĞLU

Keywords: Genetic Algorithms, Simulated Annealing, Linear Predictive Coding, Speaker Recognition, Artificial Neural Networks, Self Organizing Maps

Abstract: In this thesis work Speaker Recognition application based on Artificial Neural Network is studied by using conventional feature extraction method and Genetic Algorithms and Simulated Annealing method.

In the stage of obtaining feature extraction vectors, two different methods as Mel-frequency cepstral coefficients method and the Genetic Algorithms Simulated Annealing were applied. The feature vectors which are obtained from the feature extraction have been utilized in the training of Self-Organizing Map (SOM). SOM is used for classification which is the next stage of speech recognition application. Text independent closed set speech recognition is implemented.

A speaker data set consisting of 5 male and 5 female speakers is used. The data set is based on CLSU (Center for Spoken Language Understanding) where the speakers uttered various numbers twice in different sessions. 10 of these utterances were aimed to be used in training while the others were aimed to be used in test stages.

It can be concluded that Genetic Algorithms and Simulated Annealing method gives better results than cepstral coefficients method based on Linear Predictive Coding in the feature extraction stage.

ÖNSÖZ ve TEŞEKKÜR

Konuşma tanıma uygulamaları gün ve gün önemini ve kullanım alanını geliştirmektedir. Daha iyi bir konuşma tanıma sistemine sahip olmak için farklı yöntemlerin bir araya getirilmesi gerekebilir. Bu karma yaklaşımlar sonucunda konuşma tanıma sistemlerinde daha iyiye doğru bir ilerleme olacağına inanıyorum. Yapılan bu tez çalışmasının ileride bu konu üzerinde çalışacaklar için yararlı bir kaynak olmasını dilerim.

Bu çalışmada yardımları, önerileri ve desteğinden dolayı danışmanım Değerli hocam Yard. Doç Dr. Melih İNAL' a, ayrıca çalışmam süresince yardımlarını esirgemeyen diğer tüm hocalarıma, arkadaşlarıma, çok değerli aileme teşekkürü borç bilir ve saygılarımı sunarım.

İÇİNDEKİLER

ÖZET.....	ii
ABSTRACT.....	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
İÇİNDEKİLER.....	v
SİMGELER ve KISALTMALAR.....	vii
ŞEKİLLER DİZİNİ.....	viii
TABLolar DİZİNİ.....	x
BÖLÜM 1. GİRİŞ.....	1
1.1. Tez Çalışmasının Amacı.....	1
1.2. Tez Çalışma Basamakları.....	2
1.3. Tez Çalışmasını Oluşturan Bölümler.....	2
BÖLÜM 2. KONUŞMA VE KONUŞMACI TANIMA SİSTEMİ.....	4
2.1. Özellik Çıkartım (Feature Extraction) Yöntemleri.....	9
2.1.1. Süzgeç Bankası (Filter Bank) Modeli.....	10
2.1.2 Güç Spektral Analizi (Power Spectral Analysis).....	10
BÖLÜM 3. GENETİK ALGORİTMALAR ve BENZETİLMİŞ TAVLAMANIN KARMA KULLANIMI.....	27
3.1. Genetik Algoritmalar.....	27
3.2. Genetik Algoritmaların Çalışması.....	31
3.3. Gerçel Kodlama.....	35
3.4. Genetik Algoritma Teori ve Hipotezleri.....	38
3.4.1 Temel içerenler teorisi.....	38
3.4.2 Yapı taşları hipotezi.....	39
3.5 Benzetilmiş Tavlama.....	40
3.5.1. Metallerde tavlama.....	41

3.5.2. Benzetilmiş Tavlama Prensipleri	42
BÖLÜM 4. AMAÇLANAN YSA SINIFLANDIRICISININ VE KONUŞMACI VERİTABANININ OLUŞTURULMASI.....	45
4.1. Öğrenme Modları.....	46
4.2. Öğrenme Kuralları	47
4.2.1. En küçük kareler yöntemi	47
4.2.1.1. Ağırlıkların hesaplanması	47
4.2.1.2. En dik eğim yöntemiyle ağırlıkların (W^{**} nin) bulunması.....	49
4.2.2. Geri yansıtma öğrenme kuralı.....	51
4.2.3. Yarışmacı öğrenme kuralı.....	51
4.3. YSA Sınıflandırıcı Modelleri.....	53
4.3.1. Çok katmanlı almaç - ÇKA	53
4.3.2. Kendi kendini organize eden ağ modeli	54
4.4. Yapay Ağaç Ağ Modeli (YAM)	55
4.5. Amaçlanan Yöntem ve Veritabanının Oluşturulması.....	57
4.5.1. Özellik çıkartım aşaması.....	57
4.5.1.1. Önışleme aşaması	58
4.5.1.2. Pencereleme aşaması	58
4.5.1.3. Mel-frekans ölçekli kepral katsayıların elde edilmesi	60
4.5.2. Veritabanının oluşturulması.....	61
4.5.3. Konuşmacı veritabanının özellik çıkartım işlemleri.....	66
4.6. Amaçlanan Yöntem	66
BÖLÜM 5. KONUŞMACI SAPTAMA UYGULAMALARI.....	68
5.1 SOM Sınıflandırıcısının Kullanıldığı Konuşmacı Saptama Uygulamaları.....	74
BÖLÜM 6. SONUÇ ve ÖNERİLER.....	86
KAYNAKLAR	88
EK: PROGRAM LİSTELERİ.....	91
ÖZGEÇMİŞ.....	93

SİMGELER DİZİNİ ve KISALTMALAR

W	: Ağırlık Vektörü
z^{-1}	: Geri kaydırma operatörü
X	: Giriş vektörü
ε	: Hata ifadesi
y	: Çıkış vektörü
d	: İstenen çıkış değeri
ξ	: Ağırlık vektörlerinin fonksiyonu
μ	: “adım”; ağırlık vektörünün minimum hataya yakınsama hızı
$\Delta W(t)$: t inci zaman adımındaki ağırlıkların değişimi
∇	: Gradient operatörü
η	: Öğrenme oranı.
(n)	: Kestirilmiş sinyal.
YSA	: Yapay Sinir Ağları
DÖK	: Doğrusal Öngörülü Kodlama
ÇKA	: Çok Katmanlı Almaç modeli
TKA	: Tek Katmanlı Almaç
SOM	: Kendi kendini organize eden ağ modeli (Kohonen Ağı)
AFD	: Ayrık Fourier Dönüşümü
AKD	: Ayrık Kosinüs Dönüşümü
BBM	: Birleştirilmiş Bellek Modeli
TIMIT	: Texas Instruments (TI) Massachusetts Ins. of Tech (MIT)
CSLU	: (C)enter for (S)poken (L)anguage (U)nderstanding
OGI	: (O)regon (G)raduate (I)nstitute
BT	: Benzetilmiş Tavlama
GA	: Genetik Algoritma
GABT	: Genetik Algoritmalar ve Benzetilmiş Tavlama
RIFF	: Resource Interchange File Format

ŞEKİLLER DİZİNİ

Şekil 2. 1: Genel konuşma sistemleri.....	4
Şekil 2. 2 Genel konuşmacı tanıma sistemi.....	8
Şekil 2. 3 Konuşma tanıma için sınıflandırıcı yapısı.....	9
Şekil 2. 4 Süzgeç Bankası modeli.....	10
Şekil 2. 5 F1 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	12
Şekil 2.6 F2 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	13
Şekil 2.7 F3 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	14
Şekil 2.8 F4 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	15
Şekil 2.9 F5 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	16
Şekil 2.10 M1 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	17
Şekil 2.11 M2 konuşmacısının izge grafiği,dalga şekli ve formant frekansları	18
Şekil 2.12 M3 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	19
Şekil 2.13 M4 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	20
Şekil 2.14 M5 konuşmacısının izge grafiği,dalga şekli ve formant frekansları.....	21
Şekil 2.15 Konuşma işaretinin Doğrusal Öngörülü Modeli	22
Şekil 3. 1 GA akış diyagramı.....	28
Şekil 3. 2 Tek kesim noktalı çaprazlama.....	33
Şekil 3. 3 İki kesim noktalı çaprazlama.....	34
Şekil 3. 4 Gerçel çaprazlama işlemi	36
Şekil 3. 5 Örnek temel içeren ve buna uyan diziler	38
Şekil 3. 6 Tanım aralıkları.....	39
Şekil 4. 1 Birbirine bağlı üç katmanlı bir ÇKA modeli.....	53

Şekil 4. 2 Kohonen Ağı (SOM).....	54
Şekil 4. 3_Yapay Ağaç ağ Modeli -YAM.....	56
Şekil 4. 4 DÖK incelemesi özellik çıkartım aşamaları.....	57
Şekil 4. 5 GABT yöntemi ile özellik çıkartım aşamaları	57
Şekil 4. 6 Hamming pencereleme işlemi örneği	59
Şekil 4. 7 Mel-frekans ölçekli filtre bankasının genlik frekans cevabı	60
Şekil 4. 8 Mel-ölçek fonksiyonu	61
Şekil 4. 9 F1 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	62
Şekil 4. 10 F2 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	63
Şekil 4. 11 F3 konuşmacısının 1. oturum ses dosyasına ait dalga şekli	63
Şekil 4. 12 F4 konuşmacısının 1. oturum ses dosyasına ait dalga şekli	63
Şekil 4. 13 F5 konuşmacısının 1. oturum ses dosyasına ait dalga şekli	64
Şekil 4. 14 M1 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	64
Şekil 4. 15 M2 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	64
Şekil 4. 16 M3 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	65
Şekil 4. 17 M4 konuşmacısının 1. oturum ses dosyasına ait dalga şekli	65
Şekil 4. 18 M5 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.....	65
Şekil 4. 19 Test aşamasında SOM ağının kullanıldığı yapı.....	67
Şekil 5. 1 İlk üç işlem biriminin diğer ağırlıklarla bağlantıları.	70
Şekil 5. 2 Eğitim sonrasında ilk üç işlem biriminin için ağırlık bağlantıları.....	70
Şekil 5. 3 Uygunluk değeri ve Parametre katsayı değeri.....	72
Şekil 5. 4 2000 nesil için GABT uygulama sonuçları ortalama değer grafiği	85
Şekil 5. 5 1000 nesil için GABT uygulama sonuçları ortalama değer grafiği.....	85
Şekil 5. 6 Mel-ölçek uygulama sonuçları ortalama değer grafiği.....	85

TABLULAR DİZİNİ

Tablo 2.1 Öngörü Hataları.....	6
Tablo 5.1 Eğitim vektörlerinin “i” çarpanı ve “i” çarpanı olmadan standart sapma ve varyans değerleri.....	73
Tablo 5.2 Eğitim ve test vektörleri için “i” çarpanı ve “i” çarpanı olmadan varyans değerlerinin karşılaştırması	74
Tablo 5.3 2000 iterasyon için Mel-frekans ölçekli katsayı değerleri uygulama sonuçları	76
Tablo 5.4 5000 iterasyon için Mel-frekans ölçekli katsayı değerleri uygulama sonuçları.....	76
Tablo 5.5 10000 iterasyon için Mel-frekans ölçekli katsayı değerleri için uygulama sonuçları.....	77
Tablo 5.6 20000 iterasyon için Mel-frekans ölçekli katsayı değerleri için uygulama sonuçları.....	77
Tablo 5.7 2000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları.....	78
Tablo 5.8 5000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları.....	79
Tablo 5.9 10000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları.....	79
Tablo 5.10 20000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları	80
Tablo 5.11 2000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları	81
Tablo 5.12 5000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları	81
Tablo 5.13 10000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları	82

Tablo 5.14 20000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları	82
Tablo 5.15 Mel-frekans ölçekli katsayı değerleri için uygulama sonuçlarının ortalamaları.....	83
Tablo 5.16 1000 nesil için GABT katsayı değerleri için uygulama sonuçlarının ortalamaları	84
Tablo 5.17 2000 nesil için GABT katsayı değerleri için uygulama sonuçlarının ortalamaları	84



BÖLÜM 1. GİRİŞ

Konuşma tanıma teknolojisi son on yılda büyük ilerlemeler göstermiştir. Daha bir kaç yıl öncesine kadar söylenen kelimeler arasında boşluk verilerek tanıma işlemi yapılabilirken, bugünlerde aralıksız konuşulan bir konuşma için bile konuşma tanıyıcı sistemler ticari anlamda kullanılmaktadır (Furui 1995). Bilgisayar teknolojiindeki gelişmeler sonucu, günümüzde gerçek zamanda konuşma ve konuşmacı tanıma gibi karmaşık uygulamalar gerçekleştirilmektedir.

1.1. Tez Çalışmasının Amacı

Tez çalışması metinden bağımsız kapalı set konuşmacı tanıma uygulaması olup çalışmada CSLU (Center for Spoken Language Understanding) veritabanından faydalanılarak hazırlanmış olan bir konuşmacı seti kullanılmıştır. Konuşmacı seti 5 bayan ve 5 erkekten oluşmaktadır. Bu tez çalışmasında; özellik çıkartım yöntemlerinin ve sınıflandırma yöntemlerinin hangi parametreleri ile daha uyumlu çalıştıkları, kullanılan yöntemlerden hangisinin en iyi sonucu verdiği ortaya konmaya çalışılmıştır. Yapılan uygulama çalışmaları ve elde edilen sonuçlar bölüm 6' da tablolar halinde gösterilmiştir.

Tez boyunca cevapları aranacak sorular;

- Hangi özellik çıkartım yöntemi ile özellik vektörleri en iyi şekilde elde ediliyor?
- Genetik algoritma ve benzetilmiş tavlama yöntemi ile elde edilmiş özellik vektörleri ile kepsral katsayılarla elde edilmiş özellik vektörlerinden hangisi sınıflandırıcı yapısı için uygun bir giriş olabilmektedir?
- SOM tabanlı YSA sınıflandırıcısı kullanılarak elde edilen konuşmacı tanıma başarı oranı yeterli mi?
- GABT yöntemi ile doğrusal öngörülü kodlama tabanlı kepsral katsayıların, özellik çıkartım vektörlerinin edilmesi başarılı mı?

1.2. Tez Çalışma Basamakları

1. Özellik çıkartım algoritmalarından Doğrusal Öngörülü Kodlama tabanlı kepsral katsayılardan yinelemeli ilişkisinden elde edilen mel-frekans kepsral katsayı yönteminin konuşma verisine uygulanması.
2. GABT karma yönteminin konuşma işaretine uygulanması sonucu özellik çıkartım vektörlerinin elde edilmesi.
3. YSA sınıflandırıcısı tasarımının yapılması ve oluşturulan SOM tabanlı YSA sınıflandırıcısına elde edilen özellik vektörlerinin uygulanması.
4. Yukarıda sayılan maddelerdeki çalışmaların sonuçları elde edilerek, genel bir değerlendirme yapıp, YSA sınıflandırıcı ile kullanılan özellik çıkartım algoritmalarının en uygun olanının belirlenmesi ve çalışma sonuçlarının değerlendirilmesi.

1.3. Tez Çalışmasını Oluşturan Bölümler

Tez çalışmasının diğer bölümleri sırasıyla aşağıdaki şekildedir:

Bölüm 2’de, Konuşma ve Konuşmacı Tanıma Sistemleri incelenmiştir. Konuşmacı tanıma sistemlerinin aşamaları, özellik çıkartım algoritmaları ve bu algoritmalarından en yaygın olarak kullanılan Doğrusal Öngörülü Kodlama (DÖK) yöntemi açıklanmıştır. Ardından özellik çıkartım aşamasında kullanılan GABT yönteminden söz edilecektir. Daha sonra, bu iki yöntemle elde edilen özellik vektörlerinin karşılaştırılması yapılarak ve sonuçlar değerlendirilecektir.

Bölüm 3’te, Genetik Algoritma (GA) ve Benzetilmiş Tavlama (BT) yöntemleri açıklanmıştır. Özellik çıkartım yöntemi olarak kullanılan GABT yönteminin yapısı, işleyişi hakkında gerekli bilgiler verilmiştir.

Bölüm 4’te, bu tez çalışmasında kullanılacak YSA sınıflandırıcılarından olan kendi

kendini organize eden ağ modeli (Self Organizing Map - SOM), eğiticişiz öğrenme algoritması ve öğrenme kuralları incelenmiştir. Daha sonra bu tez çalışmasında kullanılan, genel konuşmacı tanıma sistemindeki aşamalar, konuşmacı veritabanının oluşturulma aşamaları açıklanmıştır.

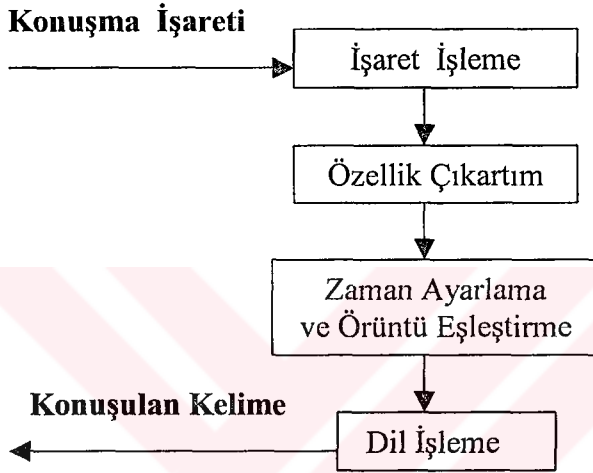
Bölüm 5'te, SOM (Self Organizing Map) YSA (Yapay sinir Ağı) tabanlı, metinden bağımsız ve kapalı set konuşmacı tanıma uygulamaları ve bu uygulamalara ilişkin sonuçlar açıklanmıştır. Bu çalışmalar aşağıdaki gibi özetlenmiştir.

CLSU Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Kapalı Set Konuşmacı Saptama Alanına Uygulanması: Bu çalışmada, CLSU veritabanı kullanarak metinden bağımsız kapalı set konuşmacı saptama uygulamaları yapılmıştır. Kapalı set konuşmacı saptama uygulaması 5 bayan, 5 erkek olmak üzere 10 kişi için uygulanmıştır. Uygulamada her oturum için 3 olmak üzere, 2 oturumda toplam 6 farklı telaffuz kullanılmıştır. Özellik çıkartım aşamasında iki ayrı yöntem uygulanmıştır. Mel-frekans ölçekli kepsral katsayılar ve GABT incelemesi sonucu elde edilen özellik vektörleri sınıflandırma aşamasında kullanılmıştır. Sınıflandırma aşamasında her konuşmacı için ayrı ayrı SOM ağları oluşturulup tek katmanda iki boyutlu 10X10 ve 20x20 işlem birimi değerleri kullanılarak, 2000 iterasyon, 5000 iterasyon, 10000 iterasyon, 20000 iterasyon değerleri için SOM ağının eğitimi yapılarak, konuşmacı saptama uygulamaları yapılmıştır.

Bölüm 6'da, yapılan çalışmaların sonuçları değerlendirilmiş, ayrıca konuşmacı tanıma sistem veriminin iyileştirilmesi için ileriye yönelik çeşitli öneriler yapılmıştır.

BÖLÜM 2. KONUŞMA VE KONUŞMACI TANIMA SİSTEMİ

Konuşma tanıma sistemlerinin bir bölümü olan özellik çıkartım yöntemlerinden söz etmeden önce konuşmacı tanıma sistemlerini incelemek daha doğru olur.



Şekil 2. 1 Genel konuşma tanıma sistemleri

Şekil 2.1'deki her birimin görevi aşağıda maddeler halinde sıralanmıştır:

1. İşaret işleme birimi, konuşma işaretinin bilgisayar ortamında sayısallaştırarak işlenebilmesi için kullanılır. Bu birimin amacı; örneklenmiş konuşma işaretini, genlik değişimlerinden, konuşmacının aksanı, vurgusu veya iletim ortamından kaynaklanan gürültüden bağımsız olarak üretmektir.
2. Özellik çıkartım birimi, işaret işleme biriminde üretilen işaretin özelliklerini çıkartarak istenmeyen bilgilerin elenmesi ve uzun bir konuşma verisinin kısa bir özetini çıkarmakta kullanılır. Bu aşama, konuşma verisinden elde edilecek parametre setinin hesaplanmasında kullanılır.

3. Zaman Ayarlama ve Örüntü Eşleştirme biriminde ise kelime sezimi için gerekli algoritmaların gerçekleştirilmesi işlemi yapılır. Bu algoritmalar, konuşulmuş kelimelerin, özellik çıkartım işlemi unda elde edilen vektörlere göre eşleştirme işlemini yapar. Zaman ayarlama; konuşma hızındaki farklılıklar sonucu, telaffuzlarda oluşan zamana bağlı değişimlere neden olan ses bilgilerinin ayarlanmasıdır.
4. Son olarak dil işleme birimi konuşulan kelimenin, kural tablosundan seçimi için kullanılır.

Geleneksel örüntü tanıma işlemi birçok alanda olduğu gibi konuşma tanıma alanında da yaygın bir şekilde kullanılır. Genel örüntü tanıma işlemi başlıca dört adımdan oluşur:

1. Özellik çıkartım, Şekil 2.1'deki gibi giriş işareti üzerinde yapılan bir takım işlemlerdir. Özellik çıkartımı için konuşma işaretine Ayırık Fourier dönüşümü (AFD), süzgeç bankası, güç spektral yada Doğrusal Öngörülü Kodlama (DÖK) incelemeleri gibi spektral teknikler uygulanır.
2. Örüntü eğitimi, aynı sınıfa ait konuşma seslerinin ilgili örüntüleri için özellikler oluşturmakta kullanılır. Genellikle “referans örüntüsü” olarak adlandırılan oluşan örüntü; bir takım ortalama alma tekniklerinden üretilmiş bir örnek yada referans örüntüsünün istatistiksel özelliklerini temsil eden bir model olabilir.
3. Örüntü sınıflandırmada, bilinmeyen test örüntüsü, her bir referans örüntü ile karşılaştırılır ve test örüntüsü ile her referans örüntüsü arasındaki uzaklık hesaplanır. Konuşma örüntülerini karşılaştırmak için, iki konuşma vektörü arasındaki uzaklık şeklinde tanımlanan yerel uzaklık ölçümü ile Dinamik Zaman Eğitimi (DZE) algoritması diye adlandırılan ve farklı oturumlara ait iki örüntüyü dengeleyen (zaman ölçeklemesi) genel zaman ayarlama işlemine ihtiyaç duyulur.

4. Mantıksal karar birimi, bilinmeyen test örüntüsü ile referans örüntü arasındaki benzerlik değerine bakarak, eşleşen en uygun referans örüntü veya örüntülerinin seçimini yapar.

DÖK Tabanlı Özellik Çıkartım Algoritmalarının İncelenmesi: Konuşma işaretinden özellik vektörlerinin çıkartımı için DÖK tabanlı parametre takımları kullanılabilir. Bu parametre takımının katsayıları: doğrusal öngörülü, yansıma ve kepsral (cepstral) katsayılar sayılabilir (Rabiner 1993). Yapılan çalışmada bu katsayılardan kepsral katsayıların, konuşmacı tanıma sisteminin verimi açısından en etkin yöntem olduğu görülmüştür (İnal 2000).

DÖK incelemesinde çeşitli yöntemler kullanılabilir. Özilişki (Autocorrelation) ve Kovaryans yöntemleri iki tanesidir. Yukarıda söz edildiği üzere DÖK tabanlı bu yöntemlerden elde edile katsayılar ve bu katsayıların dönüşümünden elde kepsral katsayılar verim açısından etkin bir yöntemdir. Özellik vektörlerinin elde edilmesinde amaç öngörü hatasının en aza indirilmesidir.

Bölüm 1.3' te söz edilen, tez çalışmasında kullanılan CSLU veritabanı içinden rasgele seçilen bir bayan ve bir erkek konuşmacının birinci oturumunda telaffuz ettikleri bir ses işareti için bu yöntemler uygulanmıştır. Her bir ses işareti 660' lık bölütlere ayrılıp Hamming pencereleme yöntemi uygulanmıştır, her bir bölütü temsil eden p katsayı derecesi, 8, 10, 12 seçilip bu değerler için ayrı ayrı öngörü hataları bulunmuştur. Tablo 2.1 her yöntem için hesaplanan öngörü hatalarını göstermektedir.

Tablo 2.1 Öngörü hataları

Öngörü Hataları							
Etiket	Özilişki katsayıları için	Özilişki kepsral katsayılar için	Kovaryans katsayıları için	Kovaryans kepsral katsayılar için	Mel-ölçekli kepsral katsayılar için	GABT katsayıları için	Derece (p)
F1	0.0681	0.2052	0.0682	0.2042	0.0635	0.0165	8
	0.0619	0.2252	0.0619	0.2239	0.0688	0.0154	10
	0.0687	0.2278	0.0687	0.2271	0.0637	0.0141	12
M1	0.1203	0.6104	0.1202	0.6037	0.1963	0.0279	8
	0.1364	0.6785	0.1362	0.6715	0.1749	0.0284	10
	0.1799	0.6362	0.1799	0.6296	0.1396	0.0275	12

Yaptığımız deneysel çalışmalar sonucu GABT yöntemi kullanılarak elde edilmiş olan özellik vektörlerinin ortalama karesel hataları, diğer yöntemleri ortalama karesel hatalarından daha düşük olduğu saptanmıştır. Bu sonuca göre GABT ve mel-frekans ölçekli kepsral katsayılar yöntemleri kullanılarak özellik vektörlerinin sınıflandırılması yoluna gidilmiştir.

Konuşmacı tanıma uygulamaları, konuşmacı saptama ve konuşmacı doğrulama olmak üzere iki sınıfa ayrılır;

Konuşmacı doğrulama, kendini sisteme tanıtan bir konuşmacının, kim olduğunun doğrulanarak sisteme kabul edilmesi ya da reddedilmesi işlemi olarak tanımlanabilir.

Konuşmacı saptama, sisteme giriş için başvuran konuşmacının bilinen konuşmacılar listesi içerisinde kim olduğunun saptanması işlemi olarak tanımlanabilir.

Konuşmacı saptama işlemi doğrulama işleminden daha zordur. Çünkü, sisteme sunulan konuşmacı, sistem tarafından bilinen tüm konuşmacılarla karşılaştırılmak zorundadır. Sistem; yapılan en iyi eşleştirme sonucuna göre konuşmacıyı saptamış olacaktır.

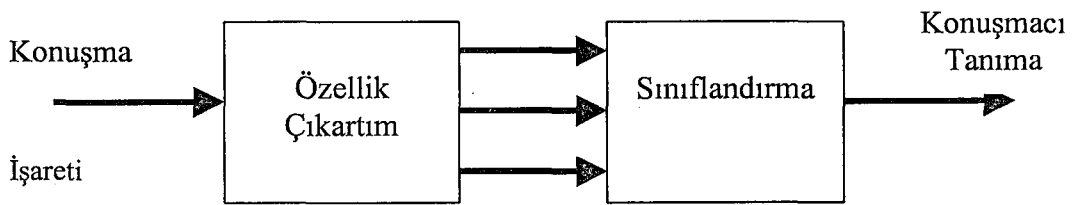
Konuşma doğrulama işleminde ise sisteme kendini tanıtan konuşmacı ile önceden sistem tarafından bilinen o konuşmacının karşılaştırılmasıdır. Bu nedenle konuşma doğrulama işlemi daha az karmaşık olup, hata yapma olasılığı daha azdır.

Konuşmacı tanıma sistemlerinde kullanılan konuşmacı seti, kapalı ve açık konuşmacı seti olarak ikiye ayrılır. Kapalı konuşmacı seti uygulamalarında, sisteme başvuran konuşmacı, sanki sistem tarafından bilinen bir konuşmacıymış gibi saptama ya da doğrulama işlemi yapılır, bu işleme “Kapalı Set Konuşmacı Tanıma İşlemi” denir. Aslında bu konuşmacı, sistemin yeni bir üyesi olabilir ve tanıma işlemine katılması muhtemel bir kişi ise bu durumda tanıyıcı sistem, bu konuşmacıyı da konuşmacı setine dahil ederek eğitim işlemine tabi tutabilir, bu işleme de “Açık Set Konuşmacı Tanıma İşlemi” denir. Bu durum başka bir problemi doğurmaktadır. Konuşmacı saptamada, bilinen konuşmacılar arasında yapılan eşleştirme yeterince iyi değilse ve

kimliğini iddia eden bu konuşmacı aslında tanıyıcı tarafından bilinen bir konuşmacı ise bu konuşmacının sanki yeni bir konuşmacıymış gibi saptama işlemine eklenmesi doğru olmaz. Doğrulama işleminde bu durum nispeten daha kolaydır. Kimliğini iddia eden konuşmacı eğer sistem tarafından bilinmiyorsa, doğrulama işlemi bu konuşmacıyı tanımak için yeni bir konuşmacı olarak eğitim işlemine katar. Bu yüzden her iki konuşmacı tanıma işleminin açık set konuşmacı uygulamalarında bir eşik değeri kullanılarak konuşan kişinin yeni bir konuşmacı olup olmadığı belirlenir.

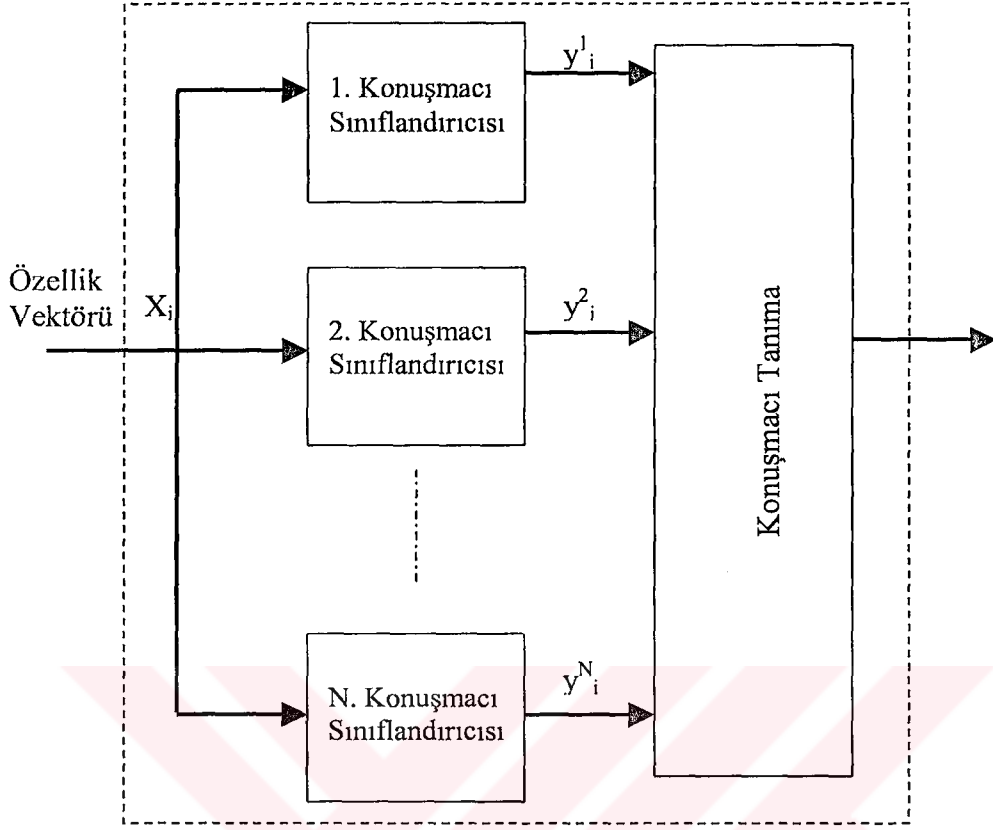
Konuşmacı tanıma uygulamalarında, tanıma işlemi için konuşmacılara belirli cümleleri telaffuz etmeleri istenebilir yani metin sınırlandırılması getirilebilir. Böyle işleme “Metne Bağlı Tanıma” denir. Aksine hiç bir sınırlandırma getirilmeksizin konuşmacı istediği telaffuzu yaparak tanıyıcıda bu telaffuzları tanıma yoluna gidebilir. Bu şekilde yapılan tanıma uygulamalarına “Metinden Bağımsız Tanıma” denir (Furui 1995).

Genel konuşması tanıma sistemi Şekil 2.2’ de gösterilmektedir. Şekil 2.2’ deki özellik çıkartım birimi; sayısallaştırılmış ses işaretinin özelliklerini çıkarmakta kullanılır. Bu aşama konuşma verisinden elde edilecek parametre setinin hesaplanmasında kullanılır. Sınıflandırma biriminde özellik çıkartım biriminden elde edilen ses özellikleri sınıflandırılarak saptama veya doğrulama işlemine uygulanır.



Şekil 2.2 Genel konuşmacı tanıma sistemi.

Konuşmacı tanıma sistemi, her konuşmacıyı ayrı bir sınıflandırıcısı olacak şekilde modeller. Şekil 2.3’deki her konuşmacı modeli, verilen özellik vektörü ile karşılaştırılır. Bu sonuca göre en iyi benzerliği gösteren konuşmacı modeli karar biriminde belirlenir.



Şekil 2.3 Konuşma tanıma için sınıflandırıcı yapısı

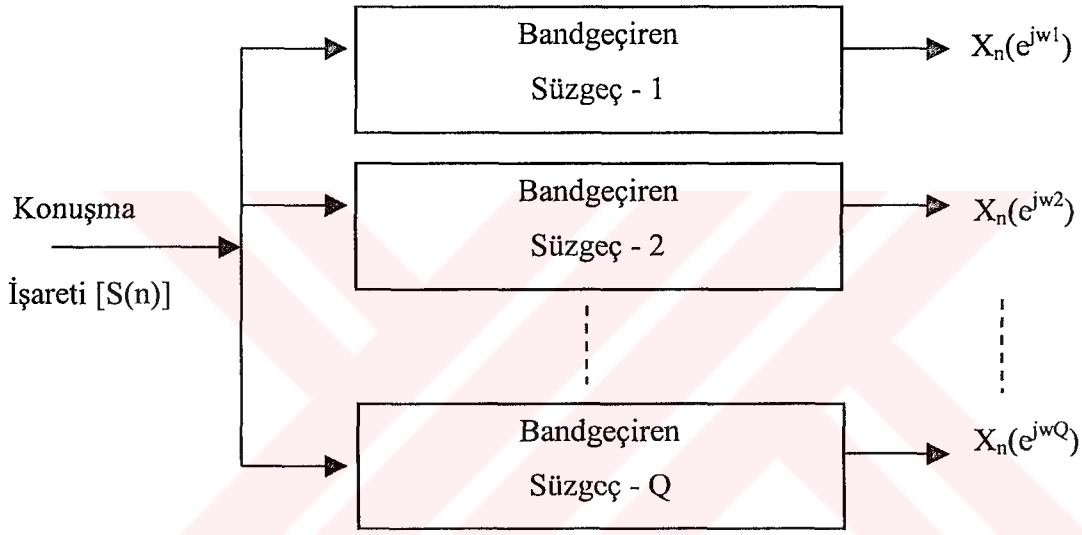
2.1. Özellik Çıkartım (Feature Extraction) Yöntemleri

Özellik çıkartım yöntemlerinin amacı; konuşma işaretinin parametrik özelliklerini belirlemektir. Bu yöntemlerin en sık kullanılanları aşağıda verilmiştir:

1. Süzgeç Bankası (Filter Bank) Modeli
2. Güç Spektral Analiz (Power Spectral Analysis)
3. Doğrusal Öngörülü Kodlama (Linear Predictive Coding)

2.1.1. Süzgeç bankası (Filter Bank) modeli

Süzgeç bankası modeli Şekil 2.4' de gösterilmiştir. Sayısallaştırılmış konuşma işareti $S(n)$, frekans aralıkları (telefon konuşma işaretleri için 30-3000 Hz, geniş band işaretler için 100-8000 Hz) işlenecek işaretin örnekleme frekansına göre Q adet bandgeçiren süzgeç bankası üzerinden geçirilmektedir. Sonuçta her bandgeçiren süzgeç, konuşma işaretinin birbirinden bağımsız değişik frekans aralıklarındaki $X(n)$ işaretlerini üretir.



Şekil 2. 4 Süzgeç Bankası modeli

2.1.2 Güç spektral analizi (Power Spectral Analysis)

Konuşma işaretiyle çalışmanın en yaygın tekniklerinden biri güç spektrumudur. Bir işaretin güç spektrumu, işaretin zamana göre frekans içeriğini açıklar. Konuşma işaretinin güç spektrumunu hesaplamamanın ilk adımı AFD' nin (Ayrık Fourier Dönüşümü) işarete uygulanmasıdır. AFD zaman bölgesindeki bir işaretin eşdeğer frekans bilgisini hesaplamak için kullanılır. AFD uygulanmış işaret, orijinal zaman bölgesindeki işaretin faz ve genlik bilgilerini içerir.

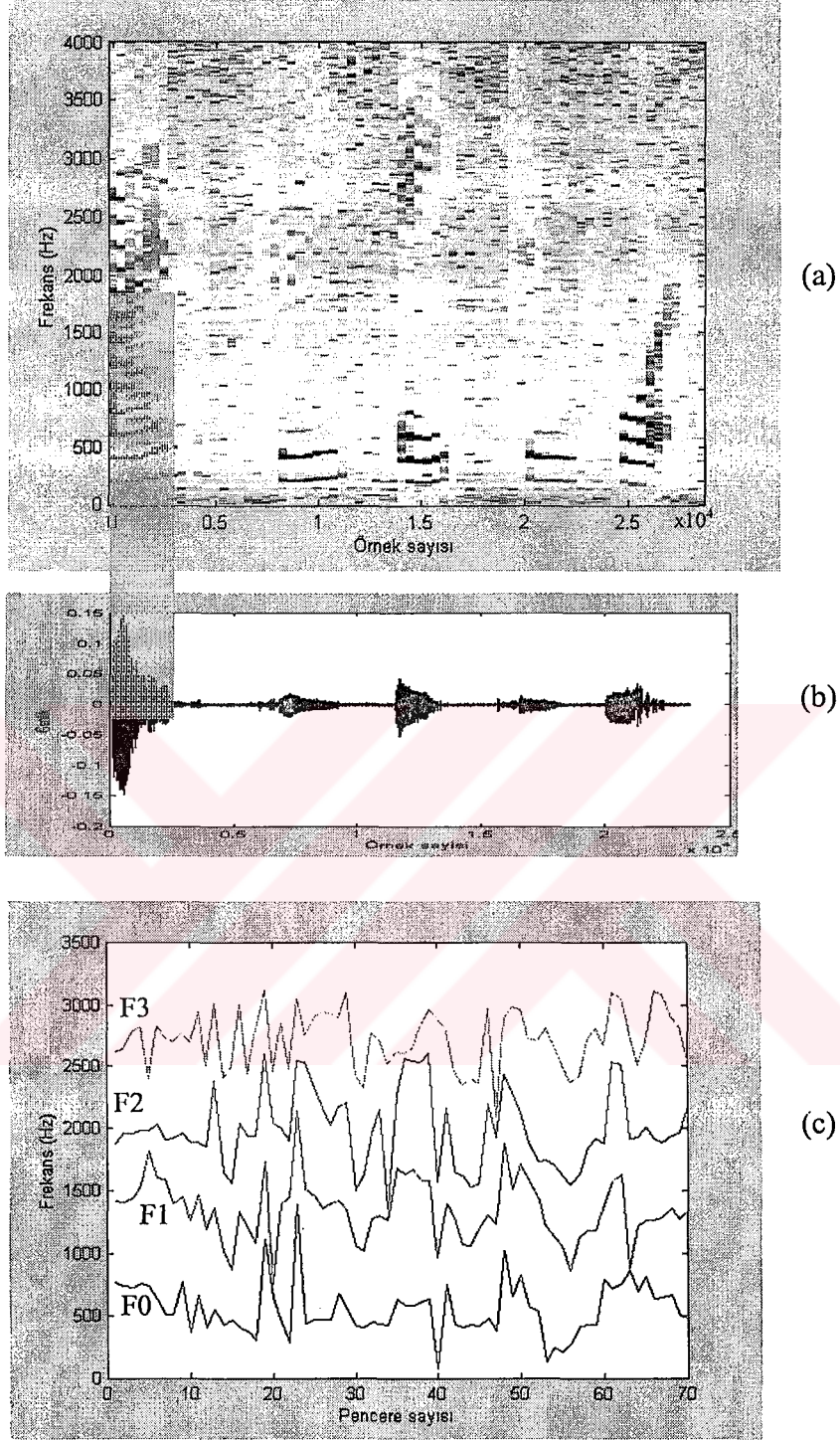
Ayrık Fourier Dönüşümü:

$$X(e^{j\theta}) = \sum_{n=-\infty}^{n=\infty} x(n)e^{j\theta n}, \theta = 2\pi f / T \Rightarrow 2\pi f / f_s \quad (2.1)$$

Ters Ayrık Fourier Dönüşümü:

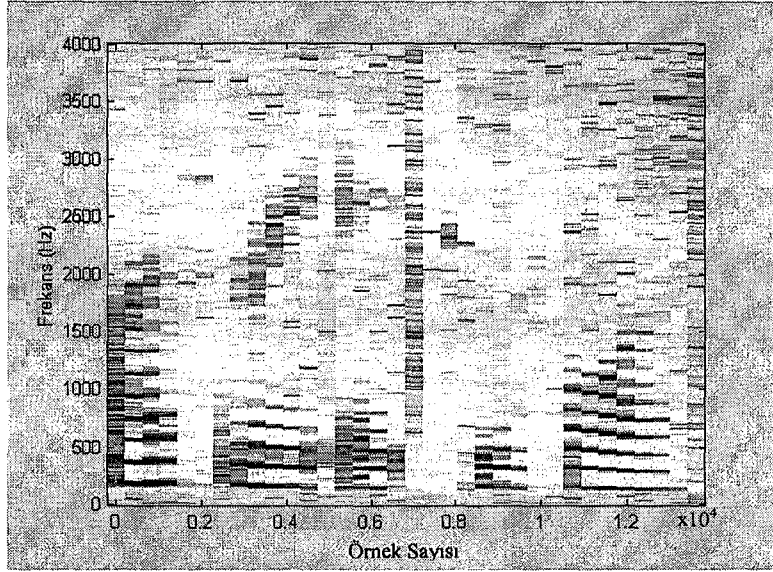
$$x(n) = \int_{-\pi}^{\pi} X(e^{j\theta}) e^{j\theta n} d\theta \quad (2.2)$$

Her sinyal temelde belirli bir frekanstaki sinyalin ve o frekansın katları kadar farklı sinyalin, değişik oranlarda doğrusal biçimde birleşmesinde oluşur. Aynı şekilde ses işaretinin de Fourier dönüşümü alınırsa zaman bölgesinden frekans bölgesindeki eşdeğerine geçiş yapılır. Bu sinyallerin katsayıları sinyalin karakteri hakkında önemli ölçüde ip ucu verir. Ses işareti pencerelere bölünür ve her birinin Fourier dönüşümü alınır ve sonuç olarak ortaya izge grafiği (spectrogram) ve sesin genel karakteristiğini veren bir grafik elde edilir. İzge grafiği; işaretin 3 bileşenin bir arada bulunduğu bir görsel gösterim şeklidir. Yatay eksen zamanı, düşey eksen ise frekansı gösterir. İzge grafiğindeki karanlık alanlar geniş bant izgegrafik incelemesinde, sesin karakteristiğini belirleyen formant frekanslarını gösterir. Bu alanlar ses yolunun (vocal tract) rezonansına karşılık gelir, farklı ses yolu şekilleri sonucu farklı formant frekansları elde edilir. Tez çalışmasının uygulama kısmında kullanılan ses veritabanı içindeki 5 erkek ve 5 bayan konuşmacının ses işaretlerinin karakteristiklerinin gösterildiği izge grafikleri ve formant frekansları Şekil 2.5 – 2.14’ de gösterilmiştir.

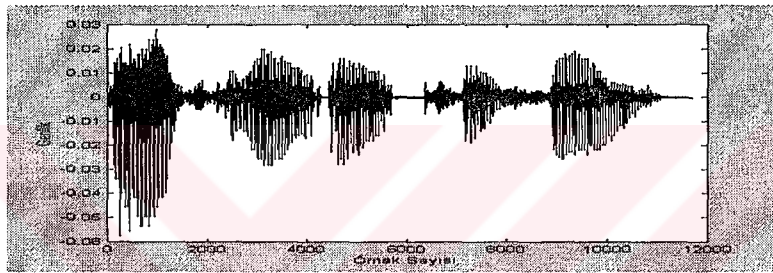


Şekil 2. 5

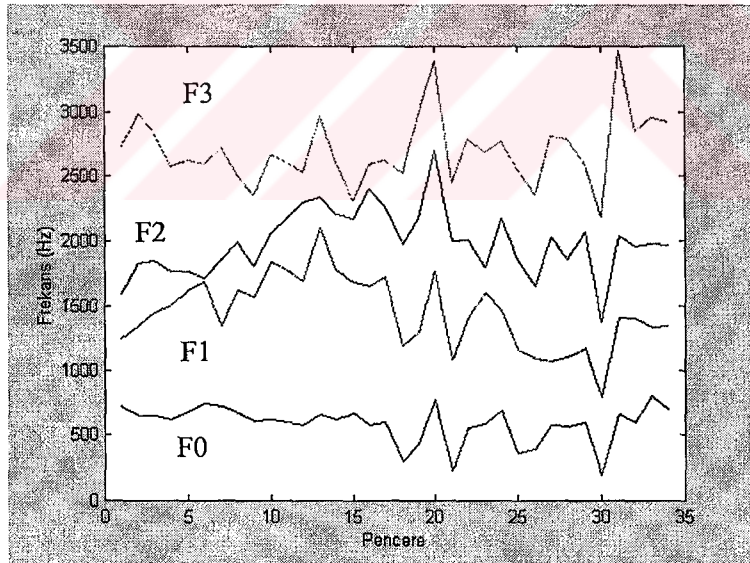
- (a) F1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
- (b) F1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
- (c) F1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



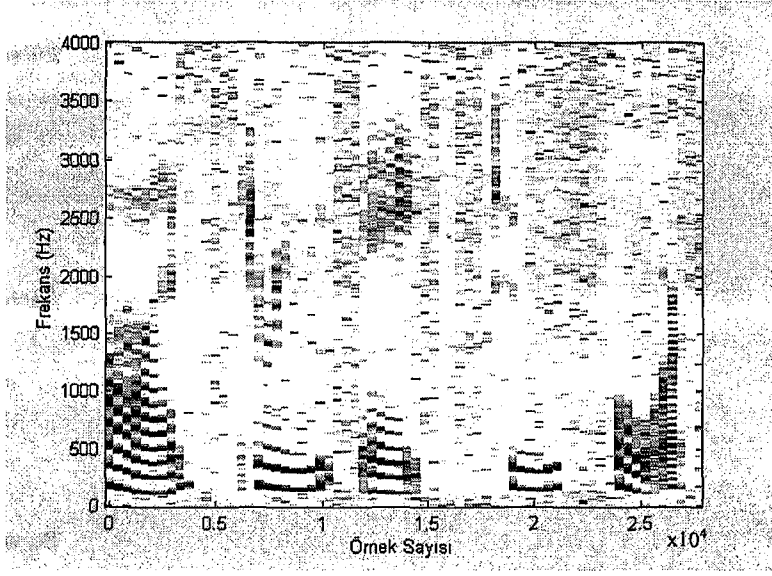
(b)



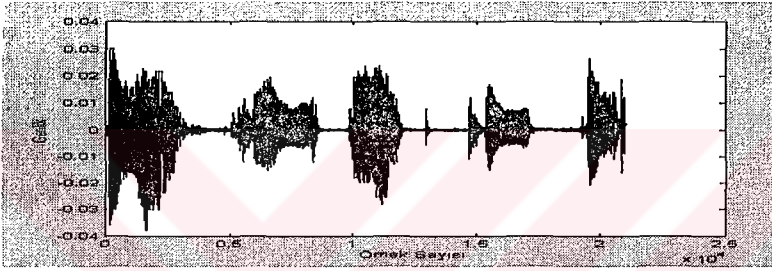
(c)

Şekil 2.6

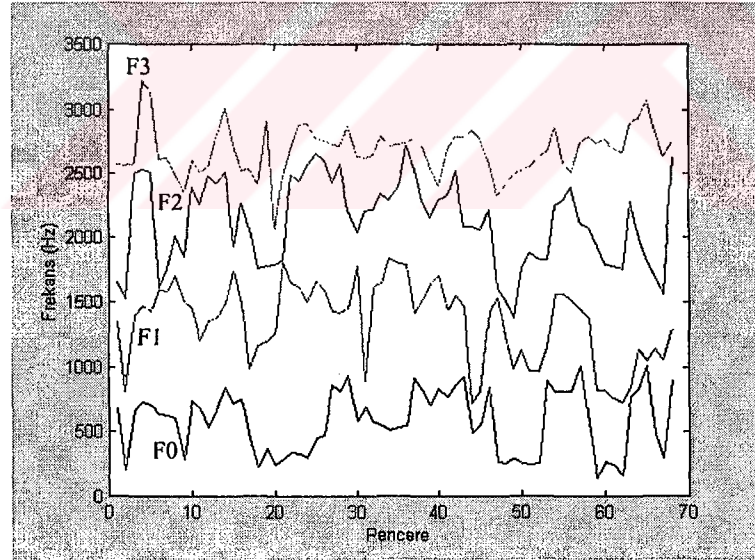
- (a) F2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) F2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) F2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



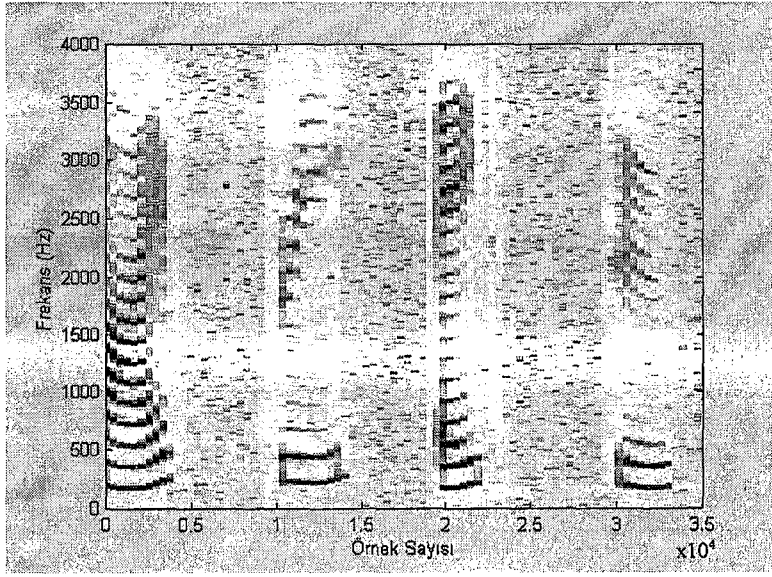
(b)



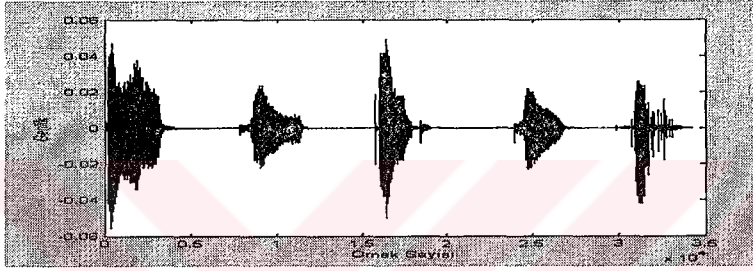
(c)

Şekil 2.7

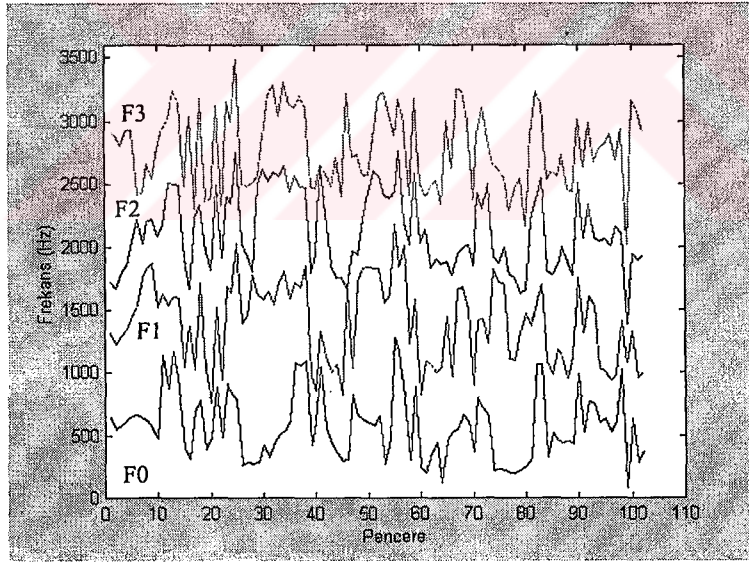
- (a) F3 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun izge grafiği
 (b) F3 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun dalga şekli.
 (c) F3 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



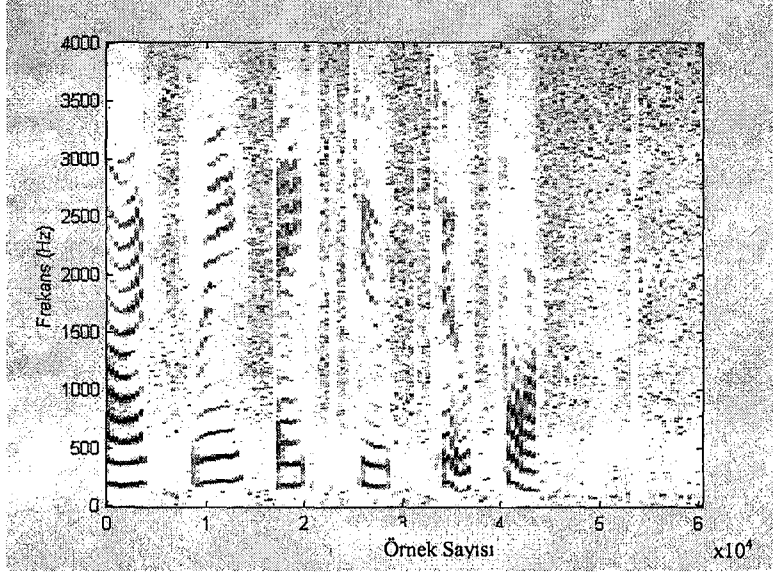
(b)



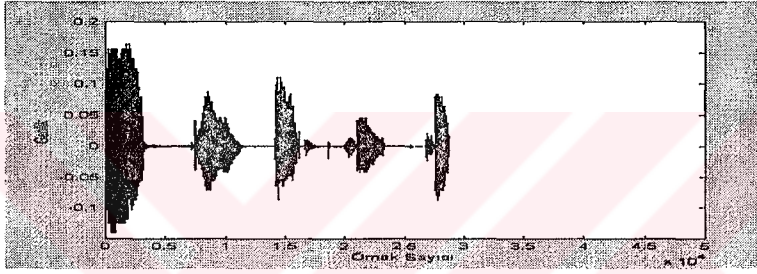
(c)

Şekil 2.8

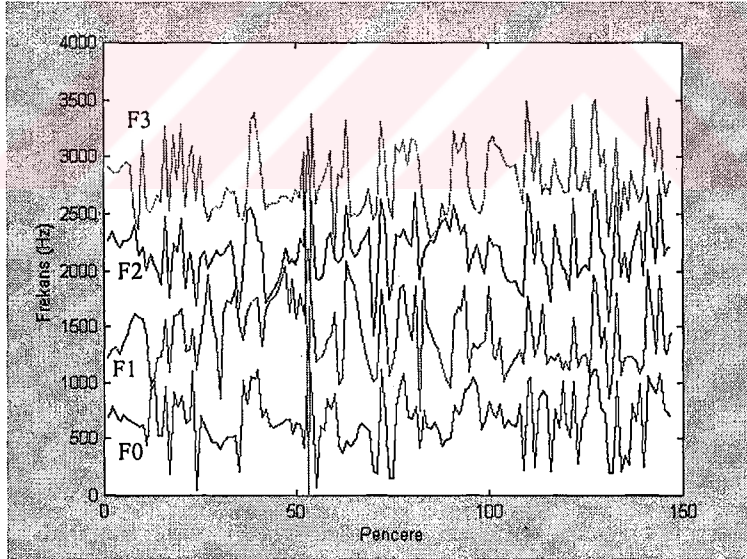
- (a) F4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) F4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) F4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



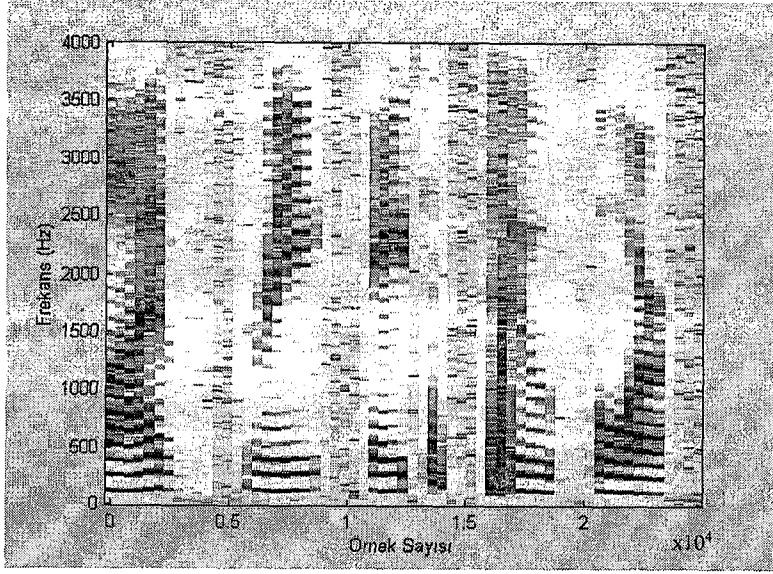
(b)



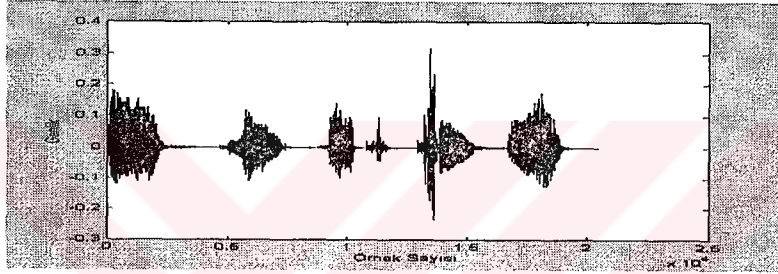
(c)

Şekil 2.9

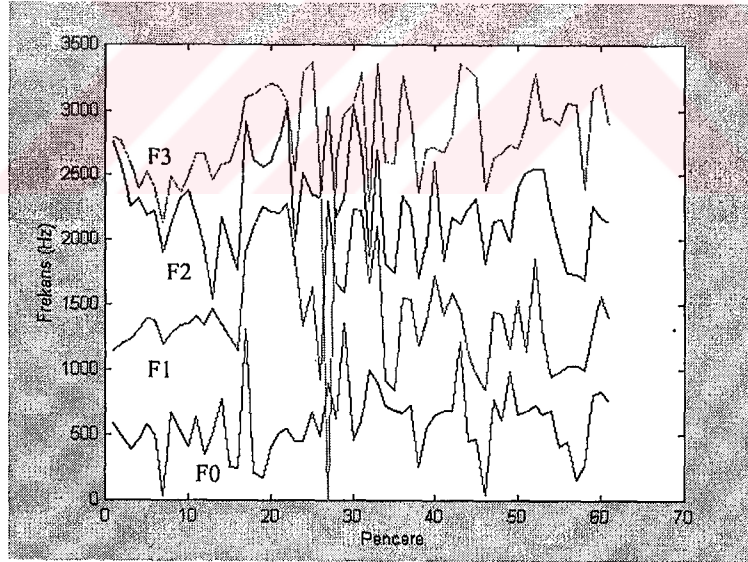
- (a) F5 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) F5 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) F5 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



(b)



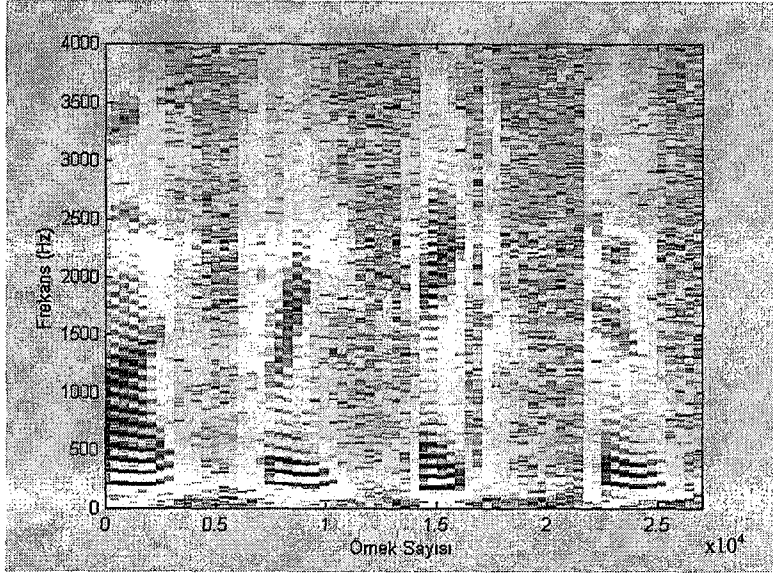
(c)

Şekil 2.10

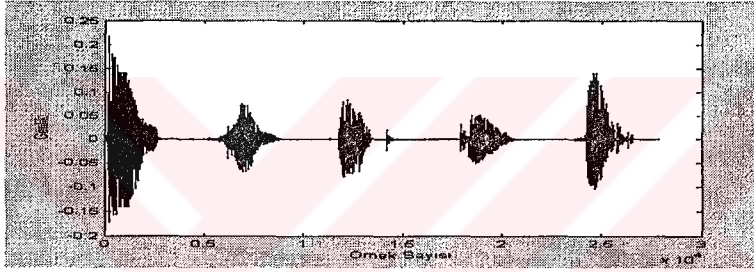
(a) M1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği

(b) M1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.

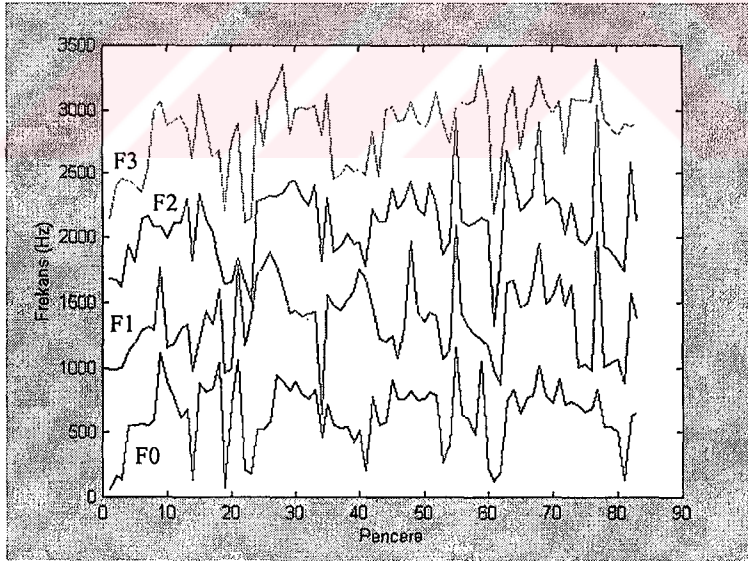
(c) M1 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



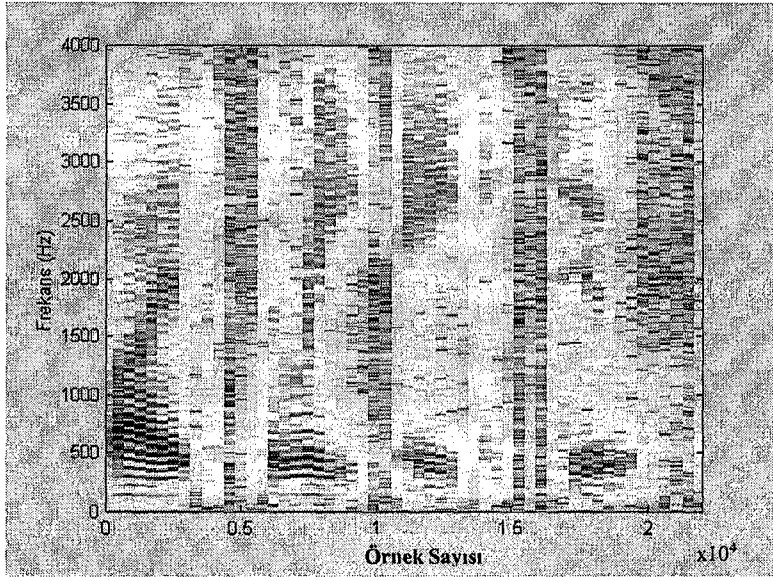
(b)



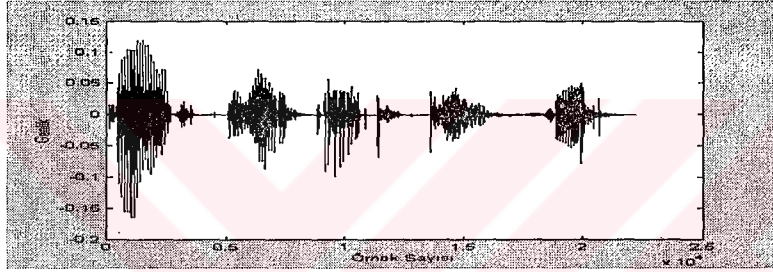
(c)

Şekil 2.11

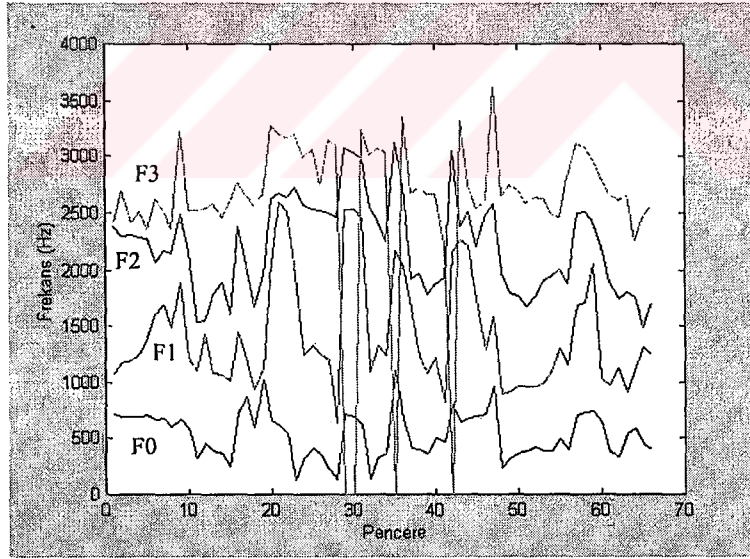
- (a) M2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) M2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) M2 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



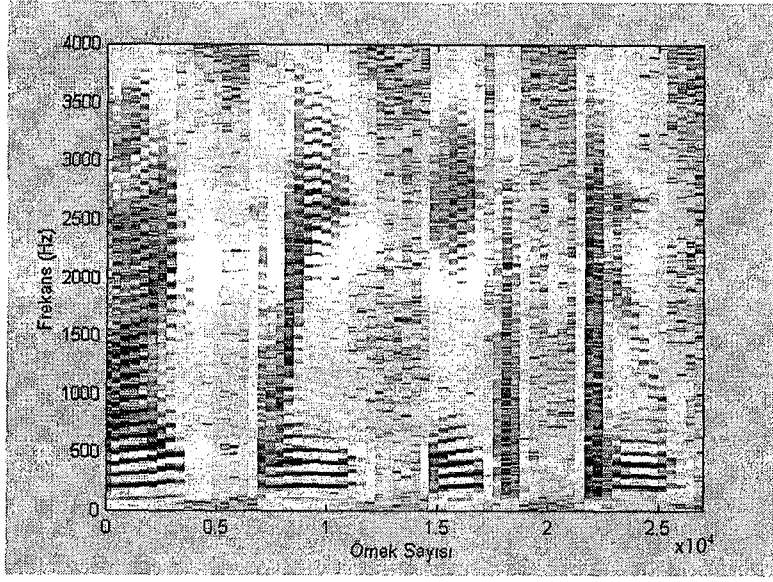
(b)



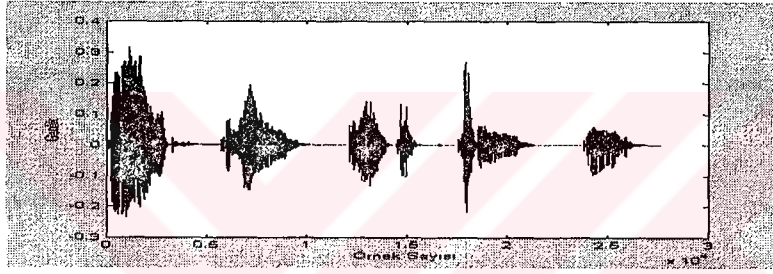
(c)

Şekil 2.12

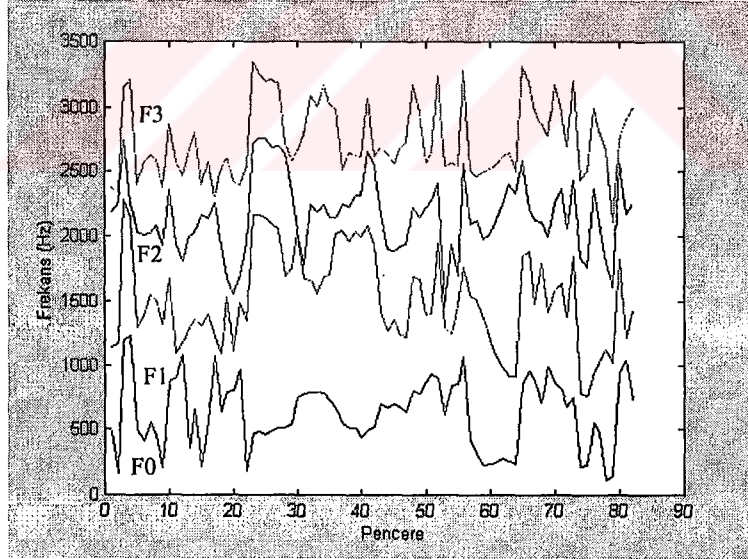
- (a) M3 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) M3 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) M3 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



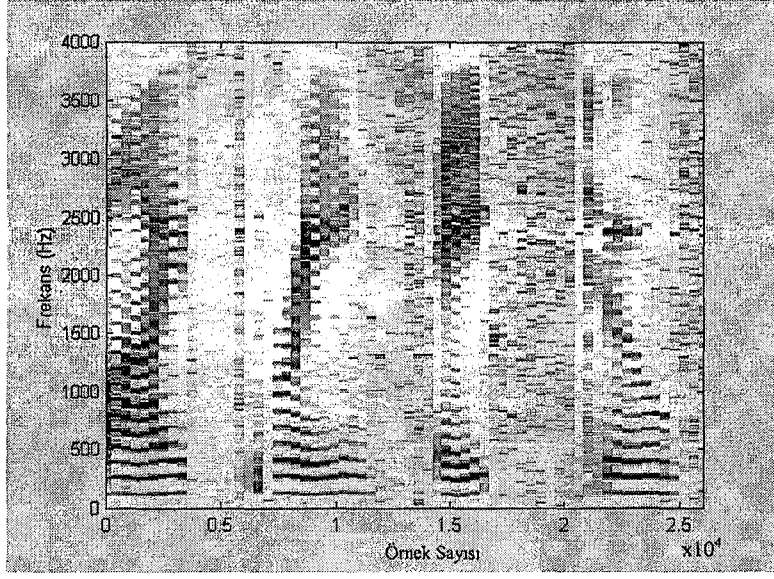
(b)



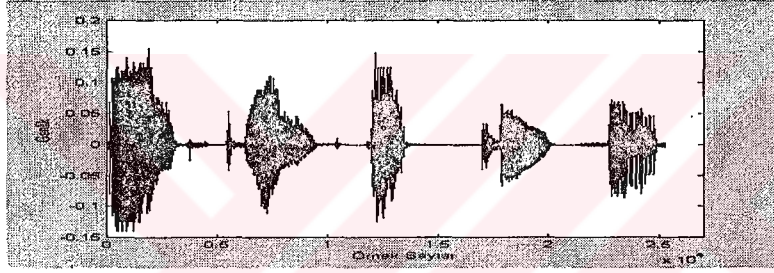
(c)

Şekil 2.13

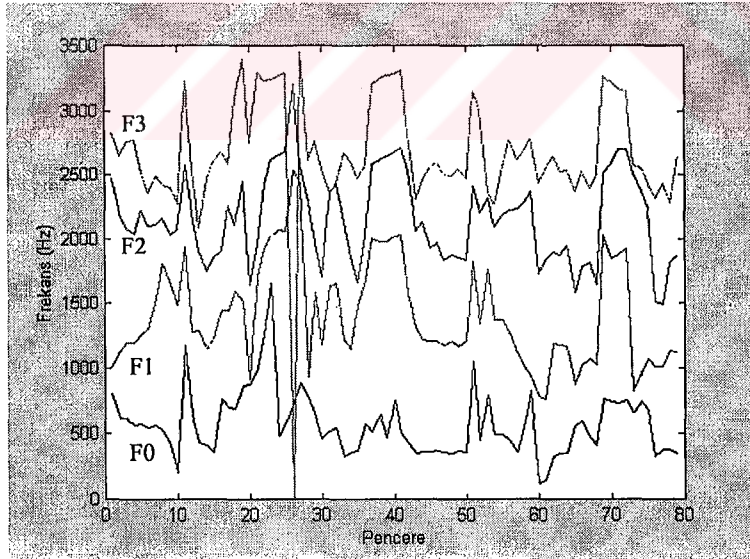
- (a) M4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun izge grafiği
 (b) M4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun dalga şekli.
 (c) M4 konuşmacısının 1. oturumdaki “5 3 8 2 4” telaffuzunun formant frekansları (F0,F1,F2,F3).



(a)



(b)



(c)

Şekil 2.14

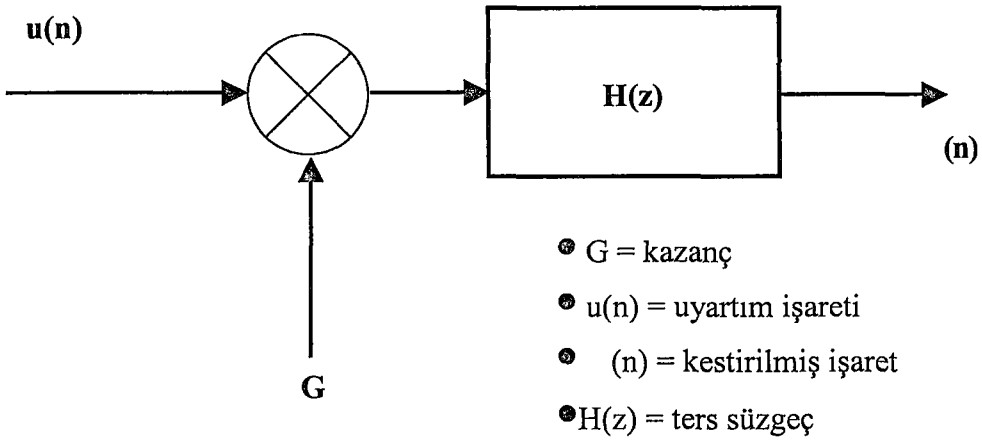
- (a) M5 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun izge grafiği
 (b) M5 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun dalga şekli.
 (c) M5 konuşmacısının 1. oturumdaki "5 3 8 2 4" telaffuzunun formant frekansları (F0,F1,F2,F3).

2.1.3 Doğrusal öngörülü kodlama modeli

Konuşmacı tanımada Doğrusal Öngörülü Kodlama (DÖK) son derece yaygın kullanılan bir aşamadır. Konuşma işaretinden özellik vektörlerinin çıkartımı için DÖK incelemesi yapılarak DÖK tabanlı parametre takımları bulunabilir. Bu parametre takımı katsayıları doğrusal öngörülü ve kepsral katsayılarıdır. Şekil 2.15' te konuşma işaretinin doğrusal öngörülü modeli görülmektedir.

Doğrusal Öngörülü Kodlama Modelinin Avantajları

- DÖK, konuşma işaretinin bir modelini oluşturur. Model; konuşma işaretinin sessiz ve geçiş bölgeleri boyunca, sesli bölgelere göre daha az etkili olmasından dolayı konuşma tanıma alanında etkin bir yöntemdir.
- DÖK modeli, ses yolu karakteristiklerini en iyi şekilde ifade eder.
- DÖK analitik olarak ifade edilebilen bir modeldir. Matematiksel açıdan doğru ve basittir. DÖK işlemindeki hesaplamalar, süzgeç bankası modelindeki tüm sayısal uygulamalardan oldukça az ve basittir.
- DÖK modelinin kullanıldığı konuşma ve konuşmacı tanıma sistemlerin verimi, süzgeç bankası tabanlı olanlarla karşılaştırılmayacak derecede iyidir (İnal 2000).



Şekil 2. 15 Konuşma işaretinin Doğrusal Öngörülü Modeli.

Şekil 2.5'te DÖK modeli oluşturulacak konuşma verisini; n anında verilen $s(n)$ örneğini, kendisinden önceki k konuşma örneklerinin doğrusal bir birleşimi şeklinde tanımlar (Kayran 1992).

$$(n) \approx a_1 s(n-1) + a_2 s(n-1) + \dots + a_k s(n-k) \quad (2.3)$$

$$e(n) = s(n) - (n) \quad (2.4)$$

Eşitlik 2.5'te (n) ifadesine; n anındaki konuşma örneğinin kestirimi denir. $s(n)$ konuşma örneği ve kestirimi arasında oluşan fark ifadesine eşitlik 2.4'te görüldüğü üzere $e(n)$ öngörü hatası denir. İnceleme penceresi boyunca oluşan hatanın, ortalama karesel ifadesinin minimum değere düşürülmesi için a parametre takımındaki katsayılar yenilenir. Yenilenen katsayılar artık bu inceleme penceresi boyunca sabit kalır. Şekil 2.15'te verilen modelden;

$$(n) = \sum_{i=1}^k a_i s(n-i) + Gu(n) \quad (2.5)$$

elde edilir. Eşitlik 2.5'te $u(n)$ uyarım işareti, G uyarım kazancı olarak adlandırılır. Eşitlik 2.5'e z ayırık dönüşümü uygularsak aşağıdaki eşitlik elde edilir.

$$S(z) = \sum_{i=1}^k a_i z^{-i} S(z) + GU(z) \quad (2.6)$$

ve Eşitlik 2.6'teki sistemin transfer fonksiyonu aşağıdaki gibidir.

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{A(z)} \quad (2.7)$$

DÖK modelinden türetilen kepsral (cepstral) katsayılar bir diğer özellik çıkartım yöntemi olup, bu alanda uygulanan en baskın yöntemdir. Kepsral katsayıları, işaretin güç spektrumunun logaritmasının ters Fourier dönüşümü şeklinde ifade edilebilir. Bu katsayılar, konuşma işaretinin ses genlik yanıtını ifade eder. Kepsral katsayılar, DÖK modelindeki a parametre takımındaki katsayıların yinelemeli ilişkisinden aşağıdaki gibi elde edilebilir (Farell 1994):

$$\begin{aligned}
c_1 &= a_1 \\
c_n &= \sum_{i=1}^{n-1} \left(1 - \frac{i}{n}\right) a_i c_{n-i} + a_n \\
1 < n &\leq p
\end{aligned} \tag{2.8}$$

Eşitlik 2.8.'de p , a ve c parametre takımı katsayılarının derecesini göstermektedir.

2.2 Konuşmacı Veritabanları

Değişik amaçlar için hazırlanmış konuşmacı veritabanları, birçok araştırmacı tarafından kullanılmaktadır. En çok kullanılan veritabanlarından dokuz tanesi aşağıda sıralanmıştır (Campbell 1999). Ayrıca uygulama kullanılan veritabanı olan CSLU' dan söz edilmiştir.

TIMIT: Texas Institute Massachusetts Institute of Technology-TIMIT veritabanı ABD'nin 8 farklı bölgesinden, 192 bayan 438 bay konuşmacıdan hem eğitim hem de test işlemleri için kaydedilmiş, metinden bağımsız toplam 10 konuşma verisine sahiptir. Ayrıca TIMIT veritabanlarının türevleri olan CTIMIT, HTIMIT, NTIMIT veritabanları değişik koşullarda (farklı mikrofon, hücresel ve şebeke telefonlarından iletilerek) kayıt edilmiştir.

NIST: Telefon santrali kayıtlarından elde edilmiş konuşmacı veritabanı.

OGI: Oregon Graduate Institute-OGI birçok dilde gerçekleştirilmiş olan telefon konuşmaları ve bölgesel özellikler gösteren konuşmacı tanıma veritabanıdır. 100 konuşmacının 47 bay ve 53 bayan konuşmacısı olup, telefon hattı üzerinden değişik koşullar altında 2 yıl gibi uzun bir periyot içinde 12 kez arama sonucu konuşma verileri kaydedilmiştir.(Cole 1998)

SIVA: Bilimsel çalışmalar için İtalyanca dilinde oluşturulmuş bir veritabanı olup, 20 erkek konuşmacının 7 farklı telefon cihazından 18 kez tekrarlanmış konuşma verilerini içerir.

YOHO(LDC): 106 bay ve 32 bayan konuşmacının 4 kez tekrarlanmış olduğu 10 konuşma verisini içerir. Ofis ortamında 3.8 kHz örnekleme frekansında kayıt edilen rakamların telaffuzu şeklindeki konuşma verisi Linguistic Data Consortium (LDC) lisansı altında kullanıcıların hizmetine sunulmaktadır.

KING(LDC): 1992 yılında LDC tarafından, 51 bay konuşmacıdan telefon hattı üzerinden değişik mikrofön özellikleri altında yapılmış kayıtları içerir.

POLYCOST: Polycost 14 ülkeden ana dili İngilizce olmayan konuşmacılardan (74 bay ve 59 bayan) konuşmacıdan edinilmiş bir veritabanıdır.

POLYVAR: Polyvar bir konuşmacı doğrulama veritabanı olup özellikle İsviçre dolaylarında yaşayan yerli ve yabancı Fransız konuşmacılardan oluşmuştur. Toplam 160 saatlik İsveç dili ve Fransızca konuşmalarını içerir. 41 konuşmacının her biri 10 kez, 31 konuşmacının da her biri 2-10 kez arasında yaptıkları telefon görüşmeleri kayıt edilmiştir.

Center for Spoken Language Understanding (CSLU) Speaker Verification Ver. 1.1 : 500 katılımcıyı içeren telefon görüşmelerinin saklandığı, Oregon Graduate Institute – OGI tarafından geliştirilmiş konuşmacı tanıma veritabanı kitaplığıdır. Her katılımcı kaydı iki yıllık periyot içinde 12 oturumda yapıp, katılımcılara göz rengi, tipik bir günlerini nasıl geçirdikleri gibi bazı sorular sorulmuş. Bunun yanında aşağıda listesi verilmiş kelime ve cümle grupları söylenmiştir. Katılımcılar ilk ay boyunca haftada 2 defa aramışlardır, ikinci ve üçüncü ayda arama yapmamışlardır. Dördüncü ayda bir arama yapmışlardır, beşinci ve altıncı ayda arama yapmamışlardır. Bu sistem 3 kez tekrarlanarak 2 yıllık periyot içine dağılmıştır.

Veritabanındaki ses dosyaları RIFF (Resource Interchange File Format) standardında 16 bit biçimindedir. Veritabanındaki telaffuzlardan bazıları aşağıdaki şekildedir;

Tez uygulamasında kullanılan CSLU veritabanı içinden seçilen 5 erkek, 5 bayan konuşmacının söyledikleri rakam telaffuzları;

5 3 8 2 4	}	Eğitim aşamasında kullanılan telaffuzlar
4 0 7 1 3		
1 9 oh 5 4		
6 1 oh 9 7	}	Test aşamasında kullanılan telaffuzlar
2 8 3 7 6		
0h 5 2 3 9		



BÖLÜM 3. GENETİK ALGORİTMALAR ve BENZETİLMİŞ TAVLAMANIN KARMA KULLANIMI

Bu bölümde, benzetilmiş tavlama ve genetik algoritmalar hakkında bilgi verilecektir. Daha sonra tez çalışması içinde özellik çıkartım vektörlerinin elde edilmesinde kullanılan, benzetilmiş tavlama ve genetik algoritmaların karma kullanım ve bu kullanımın uygulamasından bahsedilecektir.

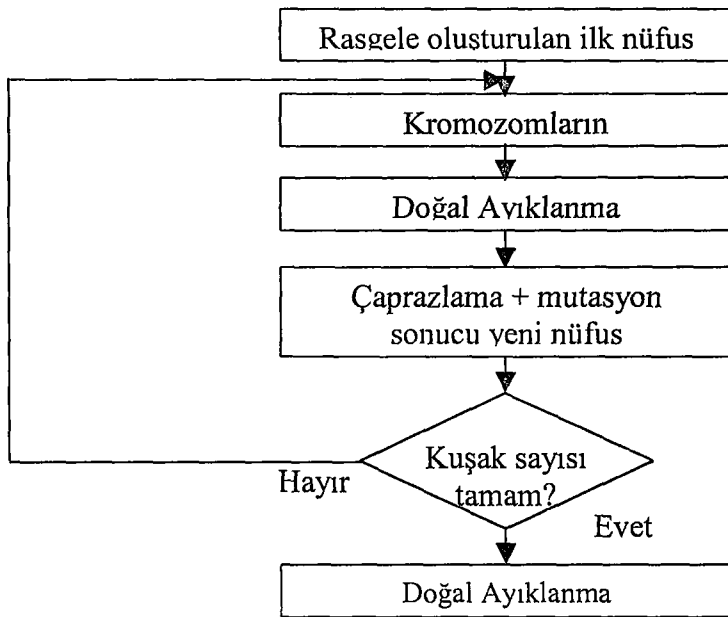
3.1. Genetik Algoritmalar

Genetik algoritmalar (GA) temelleri genetik bilimi ve bu bilimde yer alan doğal ayıklama denilen kavramlara dayandırılan eniyileme teknikleridir. Bu açıdan bakıldığında değişkenler doğal olarak kromozom üzerindeki genler olarak temsil edilirler. GA çözümler uzayından bir grup aday çözüm alınarak bir nüfus oluşturur. Sonra doğada da yer aldığı üzere doğal ayıklanma ve mutasyon ile çaprazlama genetik işlemlerini uygulayarak daha iyi ve daha uyumlu kromozomları bulur. Doğal ayıklanma ile en iyi uyum sağlayan kromozomları bulur. Doğal ayıklanma ile en iyi uyum sağlayan kromozomların gelecek kuşaklara iletilmesi kesinleşmiş olur. Çaprazlama işlemi ile GA iki ebeveyn (anne-baba) kromozomun genlerini birleştirerek iki yeni kromozom (çocuklar) oluşturur. Bu çocukların ortama anne babalarından daha iyi uyum göstermeleri yani optimum çözüme daha yakın olmaları beklenir. Mutasyon işlemi çözümler uzayında yeni alanların keşfedilmesini yani çeşitliliğin artmasını sağlar. GA' nın iyi çalışmasının sebeplerinden birisi doğal ayıklanma sayesinde tepeye tırmanma (hill-climbing) kabiliyeti ve çaprazlama ile mutasyon sonucu daha iyiye gidiş özelliklerinden birleşimini bünyesinde bulundurmasıdır (Louis ve Rawlins 1993).

Bir GA Şekil 3.1'deki akış diyagramında görüldüğü gibi dört adımın birleşmesinden oluşur. Birinci adım ilk nüfusu oluşturacak kromozomların rasgele olarak

seçilmesidir. Nüfusun büyüklüğü eniyileme süresince sabit kalır ve bu araştırmacının kontrolüne bırakılmış bir karardır. İkinci adımda değerlendirme yani kromozomların uygunluk değerlerinin hesaplanması gerçekleştirilir. Üçüncü adım doğal ayıklanma adımındır. Bu adımda uygunluk değeri en yüksek olan kromozomların bir veya daha fazla kopyası bir eşleştirme havuzuna atılırlar. Uygunluk değeri düşük olanlar nüfustan ayıklanırlar. Dördüncü adımda çaprazlama ve mutasyon işlemleri yer alır. Eşleştirme havuzunda bulunan kromozomlardan rasgele iki tanesi (ebeveynler) seçilir. Bu kromozomların eşleştirme olasılığı yine araştırmacının seçeceği bir değerdir ve genellikle yüksek bir değere ayarlanır (0,8-1.0 gibi). Eğer bu ebeveynlerin eleştirilmelerine izin verilmişse bir çaprazlama işlemi uygulanır ve iki ebeveyn arasında gen değişimi gerçekleştirilerek iki çocuk üretilir. Yeni nüfusa bu çocuklar ilave edilir. Eşleştirmeye izin verilmemişse bu ebeveyn kromozomlar değişmeden nüfusa katılırlar.

Daha sonra nüfus içindeki kromozomların çeşitliliğini arttırmak için mutasyon işlemi uygulanır. Mutasyon olma olasılığı yine araştırmacı tarafından kararlaştırılır ve genellikle düşük bir değerdir (0.01 - 0.1 gibi). Böylelikle iyi kromozomların bozulmaması sağlanır. Mutasyon basit olarak bir genin değerinin değiştirilmesidir. Bunun sonucunda da nüfus tamamen yeni üretilmiş kromozomlardan oluşmuştur ve ikinci adımdan itibaren aynı işlemler belirli sayıdaki kuşak boyunca tekrarlanır (De Jong 1993).



Şekil 3. 1 GA akış diyagramı

GA' yı diğer eniyileme yaklaşımlarından ayırtan özelliklerden birisi GA' nın değişkenlerin kendilerini değil değişkenleri temsil eden kromozomları eniyilemesidir. Bu özellik GA' nın çok esnek olmasına ve değişkenlerin birbirlerinden çok farklı oldukları durumlarda dahi eniyilemenin başarılı olmasına yol açar. Değişkenleri GA temsil etme şekli açısından, en yaygın olarak kullanılan üç tür kodlama; ikili sistem, tamsayı ve gerçel sayı kodlamalarıdır.

GA geleneksel eniyileme yöntemlerinden birkaç açıdan farklıdır. Öncelikle GA paralel ve küresel bir arama yöntemi olduğundan aramaya çok sayıda farklı nokta ile başlar ve küresel bir çözüm bulma olasılığı çok daha fazladır. Ayrıca arama uzayı ile ilgili hiçbir ön bilgisi yoktur. Bu yüzden basittir ve çok farklı problemlere uygulanabilir. Ancak GA' lar büyük nüfuslar söz konusu olduğunda yavaştır ve hassas çözümlerde iyi sonuç vermeyebilir (Reeves 1993).

Benzetilmiş Tavlama (BT) eniyileme ve yüksek dereceli problemlerde kullanılan diğer bir yöntemdir. BT enerji seviyesini en küçükmeye çalışırken belirlediği bir sonraki nokta ya da enerji seviyesini rasgele seçer.

Şekil 3.1'de yer alan akış diyagramı çerçevesinde her yeni nüfus oluşturulduğunda bir öncekine göre daha iyi kromozomlar üretilmesi beklenen bir sonuçtur. Ancak GA' nın her bir nüfusu incelemesinde, gerçekte, nüfustaki her yeni üretilen kromozomun etrafında daha yüksek veya daha düşük uygunluk değerine sahip pek çok komşu kromozom vardır. Eğer yeni kuşak değerlendirilmeden önce her yeni kromozom kendi yakın komşu kromozomlarından biriyle değiştirilirse daha hızlı yakınsama ve hassas ayarlama yapılabilir. Böylelikle BT ile birleştirilen GA iki önemli olumsuz yanından kurtulmuş olacaktır. GA tarafından üretilen her bir yeni kromozoma BT uygulanabilir. BT yeni üretilen her bir kromozom için rasgele diğer bir kromozom seçer. BT' nin seçtiği kromozomun uygunluk değeri diğerinden daha yüksekse bu yeni kromozom nüfusa eskisinin yerine katılır. Eğer BT' nin seçtiği kromozomun uygunluk değeri daha düşükse o zaman daha ileride kullanılmak üzere belirli bir olasılık dahilinde nüfusa katılabilir. BT GA ile birleştirildiğinde düşük bir sıcaklık çarpanı T ve hızlı bir tavlama çarpanı p kullanılır. Bu, yeni kromozomun

uygunluk deęerinin eskisinden düşük olması halinde onun yeni nüfusa katılma olasılıęının düşük olmasını sağlar.

Bilindięi üzere DÖK incelemesinde bir konuşma örneęi kendisinden p önceki örneklerin bir parametre takımı yardımıyla kestirimi yapılmaktadır. Burada p , katsayıların derecesini göstermektedir. Bu parametre takımı en küçük kareler yöntemi gibi geleneksel hata en küçükleme algoritmaları kullanılarak incelemenin yapıldığı pencereye uygulanarak katsayıların optimum deęeri belirlenir. Daha sonra belirlenen parametre takımı, incelemenin yapıldığı pencere için deęiştirilmeden aynı kalır.

Bu tez çalışmasında da iki önemli eniyileme yöntemi genetik algoritma ile benzetilmiş tavlama birleştirilerek oluşturulan algoritma bu parametre katsayılarının kestiriminde kullanılmıştır. Genetik algoritmalar ve benzetilmiş tavlama hakkındaki genel bilgilerden bir sonraki bölümde söz edilecektir..

Genetik algoritmalar doğada var olan bir yarışma ortamında, ancak daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildięi biyolojik olaylara benzetilerek geliştirilmiş paralel ve küresel (global) bir arama teknięidir. Arama uzayında aynı anda bir çok noktayı deęerlendirdięi için küresel çözüme ulaşması olasılığı fazladır. Genetik algoritmalar Darwin'in evrim teorisi kavramını kullanırlar. Darwin teorisi, temel olarak, canlıların varlıęının "en iyinin yaşaması" kuralına dayandığını vurgular. Darwin'e göre, ayrıca, mevcut organizmalar arasında meydana gelen üreme, çaprazlama ve mutasyon işlemlerinin sonucunda yeni tür canlılar oluşur.

Evrim teorisinin problemlere daha doğal bir yoldan çözüm aramak amacıyla algoritmalara uygulanmıştır. Önce problem için olabilecek uygun farklı çözümler, yani diziler oluşturulur. Sonra bu çözümlerin ne derece iyi sonuç verdięine bakılır. Bütün olası çözümler arasından iyi olanların bir kısmı seçilir, dięerleri ise elenerek "en iyinin yaşaması" kuralı uygulanır. Seçilen çözümlere üreme, çaprazlama ve mutasyon işlemleri uygulanarak bir önceki kuşaęa göre daha iyi sonuç vermesi beklenen yeni bir olası çözümler kuşaęı oluşturulur. Bu şekildeki yeni kuşak üretimi

ve deęerlendirilmesi iřlemleri istenene ok yakın bir kuřak yakalanıncaya kadar devam eder.

Genetik algoritmalar yeterli ve etkili bir arařtırma gerektiren problemlere uygun bir yaklařım getirdięinden iřletme, fen ve mhendislik alanlarında gittike daha fazla uygulanan bir yntemdir. Genetik algoritmalar fonksiyon eniyileme, grnt ve ses iřleme, sistem tanıma, kontrol, robot sistemleri, planlama ve mhendislik tasarımları gibi birok farklı alanlarda kullanılmıřtır (Kristinsson ve Dumont 1992; Man et al. 1996).

3.2. Genetik Algoritmaların alıřması

Bir genetik algoritmanın alıřması basit olmakla beraber etkili bir tekniktir. alıřmasının basitlięi ve etkili olması genetik algoritma yaklařımının en ekici iki zellięidir. Genetik algoritmalar geniř ve karmařık arama uzaylarında bile kresel zmlere ulařma konusunda bařarılıdırlar. Genetik algoritmaların alıřması iin drt ęe gereklidir.

1. Problemin zmlerini temsil edecek kromozomlar
2. Problemin zm nfusunu oluřturmak iin bir yntem,
3. zmn uygunluk derecesini belirleyecek bir ama fonksiyon,
4. Sonraki kuřaktaki dizilere (bilgi deęiřimi amacıyla) uygulanacak genetik iřlemler (Chang 1997).

Birok problemin zmnde iyi sonular veren genetik algoritmalar ařaęıdaki  iřlemi kullanır :

1. reme
2. aprazlama
3. Mutasyon

Genetik algoritmalar, yarışma ortamında daha kuvvetli olan bireylerin yaşama olasılığının daha fazla olduğu ve doğal seçim ya da doğal ayıklanma denilen biyolojik işleme çok benzeyen bir yol takip ederler. Buna göre, problemin sonuçta bulunacak çözümü bir dizi parametreyle temsil edilebilecek bir birey olacaktır. Biyolojideki genler ve kromozomlar olarak düşünülen bu parametreler ikili tipte bulunan diziler şeklinde temsil edilirler. Kromozomun problemi çözmedeki başarısı amaç fonksiyona yakınlık derecesi ile ölçülür. Bu genellikle 'uygunluk değeri' (fitness value) olarak adlandırılan pozitif bir değerdir. Genetik algoritma kromozomların uygunluk değerlerini kullanarak onları birbiriyle karşılaştırır.

Genetik evrim içerisinde uygunluğu daha fazla olan bir kromozomun daha kaliteli ürün vermesi beklenir. Bu da probleme daha iyi bir çözüm bulunduğu anlamına gelir. Bir genetik algoritma uygulaması gerçekleştirileceğinde içinde kromozomların bulunacağı ve bunların rasgele seçilebileceği bir havuz oluşturulur. Havuzun büyüklüğü probleme göre belirlenir. Genetik işlemin her çevriminde o anki seçilmiş kromozomlardan yeni bir kuşak oluşturulur. Bu işlemin başarılı olabilmesi genellikle 'ebeveynler' veya 'eşleştirme havuzu' olarak adlandırılan bir grup kromozomun özel bir seçme işlemiyle seçilmesine bağlıdır. Ebeveynlerin genleri bir sonraki kuşağı oluşturmak üzere karıştırılarak tekrar birleştirilir. Bu evrim işlemi sonucunda 'daha iyi' kromozomun daha fazla sayıda ürün vermesi ve böylelikle sonraki kuşakta daha fazla yaşama şansına sahip olması beklenir. Görüldüğü gibi bu doğadaki 'en iyinin yaşaması' mekanizmasına benzemektedir. Doğada uygunluk, bireyin yetişmesi ve genlerinin sonraki nesillere iletilmesini engelleyen hastalıklar, düşmanlar ve benzeri zorluklara karşı koyabilme ve hayatta kalabilme kabiliyetidir. Yapay bir düzenlemede ise oluşturulan dizinin yaşaması ya da ölmesine karar veren onun amaç fonksiyon değeridir.

Üreme işlemi birkaç yolla yapılabilir. Genellikle bu işlem bir rulet tekerleği yöntemi ile gerçekleştirilir. Bu yöntemde belirli bir andaki her dizi tekerlek üzerinde kendi uygunluk değeri oranında bir bölüm ile temsil edilir. Tekerlek döndürülüp durduğunda belirlenmiş noktaya isabet eden dizi artık yeni kuşağın bir üyesidir. Böylelikle her dizi bir sonraki kuşakta kendi uygunluk değerinin yüksekliği oranında

tekrar üremiş olacaktır. Seçilen bu diziler o anki nüfusu oluştururlar ve daha sonraki işlemler için eşleştirme havuzuna alınırlar.

Bir genetik algoritmanın kuşaklar üzerinde çalışmasının nedeni diziler arasındaki doğrudan bilgi alışverişi sebebiyle bir sonraki kuşağın daha iyi sonuç vereceği beklentisidir. Bu alışveriş çaprazlama işlemi aracılığı ile gerçekleştirilir. Eğer çaprazlama ile bir fayda sağlanamıyorsa genetik algoritma, kuşaklarla çalışmayan daha basit başka algoritmalarından daha iyi sonuç vermeyecekti (Jones 1995). Hatta rasgele arama yöntemlerinden belki biraz daha iyi olabilecekti (Spears ve Anand 1991). Klasik tek kesim noktalı çaprazlama işlemi iki adımda gerçekleştirilir. Önce havuzda bulunan yeni üretilmiş diziler rasgele eşleştirilirler. Sonra 1 (bir) ile dizi uzunluğunun bir eksiği arasında $[1, l-1]$ rasgele bir k tamsayısı seçilir. Eşleştirilmiş iki dizide $(k+1)$ inci karakter ve sonrasının karşılıklı olarak yerleri değiştirilir. Böylece iki yeni dizi elde edilmiş olur. Örnek olarak seçilen P ve R dizileri aşağıdaki eşitliklerde görülmektedir.

$$P = 01101 \quad (3.1)$$

$$R = 11000 \quad (3.2)$$

k sayısı 3 olarak belirlenirse dizilerin kesileceği yerler aşağıdaki şekildeki gibi olur:

$$P = \boxed{01101} \quad R = \boxed{11000} \quad (a)$$

$$p = \boxed{01100} \quad r = \boxed{01101} \quad (b)$$

Şekil 3.2 Tek kesim noktalı çaprazlama.

(a) Çaprazlamadan önce P ve R ebeveynleri

(b) Çaprazlamadan sonra elde edilen p ve r çocukları

Tek kesim noktalı çaprazlamadan sonra tekdüze (uniform) ve çok kesim noktalı çaprazlama işlemleri de uygulanmıştır. Tekdüze çaprazlamada dizi içindeki her bitin %50 olasılıkla, diğer ebeveynin aynı yerdeki bitiyle karşılıklı olarak değiştirilmesi

söz konusudur. Çaprazlama sonrası oluşan yeni bireyler her bir ebeveynden alınan genlerin karışımı olur. Etkin kesim noktası sayısı sabit olmamakla birlikte ortalama olarak dizinin uzunluğunun yarısı kadar olur (MAN et al, 1996). Tekdüze çaprazlamaya örnek olarak 3.1 ve 3.2 eşitliklerinde verilen P ve R ebeveynlerine ait 1., 4. ve 5. bitlerin seçildiği düşünülürse çaprazlama işleminden sonra p ve r dizileri aşağıdaki gibi gösterilebilir:

$$p = 1 \ 1 \ 1 \ 0 \ 0 \quad (3.3)$$

$$r = 0 \ 1 \ 0 \ 0 \ 1 \quad (3.4)$$

Çok kesim noktalı çaprazlama için ise yine Denklem 3.1 ve 3.2'deki P ve R ebeveynleri ile örnek iki nokta olarak 1 ve 4 alınırsa yapılan işlem ve yeni elde edilen diziler Şekil 3.3'de görülmektedir.

Kullanılacak çaprazlama tekniğinin tercihi konusu tartışmalıdır. Genel bir yorum olarak bu çaprazlama tekniklerinin her birisinin bazı problemler için özellikle daha iyi olduğu ve diğer problemler için ise çok iyi olmadığı söylenebilir.

$$\begin{array}{l}
 P = \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \qquad R = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 0 \\ \hline \end{array} \quad (a) \\
 p = \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 \\ \hline \end{array} \qquad r = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 0 & 0 \\ \hline \end{array} \quad (b)
 \end{array}$$

Şekil 3.3 İki kesim noktalı çaprazlama .

(a) Çaprazlamadan önce P ve R ebeveynleri

(b) Çaprazlamadan sonra elde edilen p ve r çocukları

Üreme ve çaprazlama sadece rasgele sayı üretimi, dizilerin kopyalanması ve bazı kısımlarının karşılıklı yer değişimi işlemleriyle son derece basit olarak gerçekleştirilebilmektedir. Çaprazlama ile elde edilen bilgi alışverişiyle üreme işleminin birleştirilmesi genetik algoritmaların gücünün çoğunu oluşturmaktadırlar. Bu iki işlem dizi içinde var olan özellikleri etkili bir biçimde arayıp tekrar

birleřtirerek alıřırlar. Bazen gelecekte ok yararlı olabilecek nemli genetik malzeme yani belirli yerlerdeki 0' lar yada 1' leri kaybetmeleri olasılıđı vardır. nc iřlem olan mutasyon, yapay genetik sistemlerde bu řekildeki erken kayıplara karřı koruyucu olarak grev yapar. Mutasyon ok dřk olasılıkla ancak yine rasgele olarak kullanılır. Mutasyon dizi iindeki bir bitin deđerinin deđiřtirilmesinden ibarettir. Standart genetik algoritmalarda mutasyon ikinci derecede bir rol oynar (Goldberg 1989).

İkili dizi iindeki bitler genleri temsil ederler ve bu grř halen geerli olan temel ieren (schema) teorisi vasıtasıyla Holland tarafından da desteklenmektedir. Bazı arařtırmacılar daha da ileri giderek mutasyon iřleminin grevinin sadece aprazlama kullanılarak gerekleřtirilebileceđi dřncesiyle mutasyonu hi kullanmayabilirler (Rosca ve Ballard 1995). Ancak mutasyon oranının ok daha yksek hatta aprazlamadan daha fazla olması gerektiđini savunan arařtırmacılar da vardır (Hinterding et al. 1995). Aslında her iki tarafın da grř teorik olarak dođrulanmamıřtır. Ne aprazlamanın ne de mutasyonun herhangi bir aıdan diđerine gre daha gl olduđu konusunda teorik bir gsterim bulunmamaktadır.

3.3. Gerel Kodlama

İkili kodlanmış GA' lar basitliđinden dolayı gerek donanım uygulamalarında sıklıkla kullanılmakla birlikte bilgisayarla yapılan benzetimler iin gerel kodlanmış GA' lar daha uygun olabilmektedir (Eshelman ve Schaffer 1995). Gerel kodlanmış genetik algoritmaların stnlkleri ařađıdaki gibi verilebilir :

- a) İkili kodlanmış bit dizilerinde olduđu gibi her hesaplama iin gerel sayıya dnřtrme gereksinimi olmadıđından daha etkindir.
- b) Gerel kodlama ile istenen deđerlerin daha yksek kesinlikle ifade edilmesi ikili kodlamada grlen ve bu eksiklikten kaynaklanan kayıpları nler.
- c) Farklı aprazlama ve mutasyon tekniklerinin uygulanması konusunda daha fazla serbestlik sađlar.

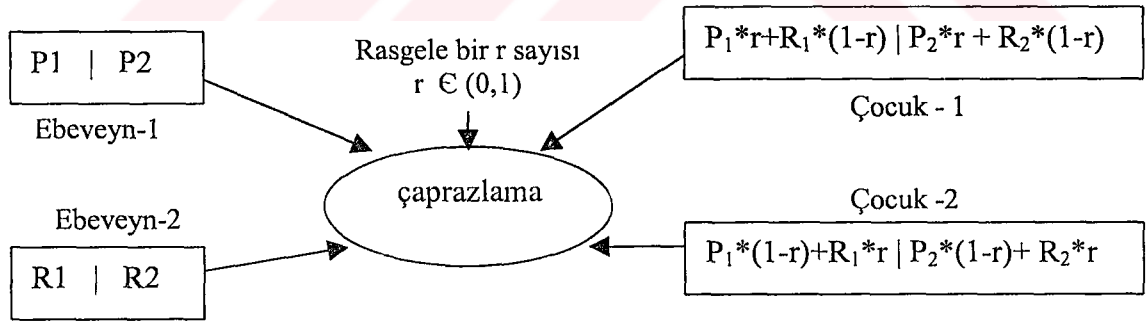
Bununla birlikte, son maddede belirtilen serbestlik daha çok parametre belirlenmesi ve daha fazla sayıda karar verilmesini gerektirir (Wright 1991). Gerçel kodlamada üreme, çaprazlama ve mutasyon işlemleri aşağıdaki gibi gerçekleştirilebilir

- a) Üreme : Bir sonraki nüfusu oluşturmak üzere o anki nüfusun uygunluk değerleri en iyi olan yarısı seçilir. Böylece genetik olarak iyi olan bireyler ebeveyn olabilir.
- b) Çaprazlama : Rasgele olarak seçilen iki ebeveyn birbiriyle genetik bilgi değişiminde bulunur ve oluşan iki yeni birey nüfusa eklenir. Bu işlem nüfusun diğer yarısı da tamamlanıncaya kadar tekrarlanır. Çaprazlama işlemini matematiksel olarak ifade etmek için ebeveynin birine P diğerine R denebilir. Çaprazlamadan sonra oluşan yeni üyeler C_1 ve C_2 aşağıdaki eşitliklerdeki gibi gösterilebilir.

$$C_1 = r * P + (1-r) * R \quad (3.5)$$

$$C_2 = (1-r) * P + r * R \quad (3.6)$$

Buradaki r , (0-1) aralığında rasgele bir sayıdır.



Şekil 3. 4 Gerçel çaprazlama işlemi

Gerçel çaprazlama işlemi Şekil 3.4' te gösterilmiştir. Eğer bireyler, her biri ayrı bir parametreyi temsil eden birden fazla reel sayıdan oluşuyorsa bu durumda çaprazlama için bir veya birkaç kesim yeri belirlenir.

Ebeveynler P için ($P_1 .P_2 .. P_n$) ve R için ($R_1 .R_2 .. R_n$) ise sadece bir kesim yeri ile çaprazlama aşağıdaki gibi gerçekleştirilir:

$$i=1 \rightarrow n_c \quad CP_i = P_i*r+R_i*(1-r) \quad (3.7)$$

$$i=n_{c+1} \rightarrow n \quad CP_i = P_i*(1-r)+R_i*r \quad (3.8)$$

$$i=1 \rightarrow n_c \quad CR_i = P_i*(1-r)+R_i*r \quad (3.9)$$

$$i=n_{c+1} \rightarrow n \quad CR_i = P_i*r+R_i*(1-r) \quad (3.10)$$

Burada n_c çaprazlama için rasgele belirlenen kesim yeri, CP ve CR' ler çaprazlama sonucu yeni oluşan bireyler, bunlar yine (0-1) aralığında rasgele bir sayıdır. Çaprazlama sonucunda elde edilen yeni bireyler aşağıda gösterilmektedir.

$$CP = CP_1.CP_2 \dots CP_n \quad (3.11)$$

$$CR = CR_1.CR_2 \dots CR_n \quad (3.12)$$

c) Mutasyon: Mutasyon oranı belirli bir olasılık dahilinde gerçekleştirilir ve bireylerin genetik bilgisinin değiştirilmesiyle yeni bilgi kazanılmasını sağlar. Seçilen bireyin $P = P_1 . P_2 \dots P_n$ olması durumunda mutasyona uğramış P şöyle elde edilir:

$$P_i = P_i + (r_i - 0.5)*2* P_{imax} \quad (3.13)$$

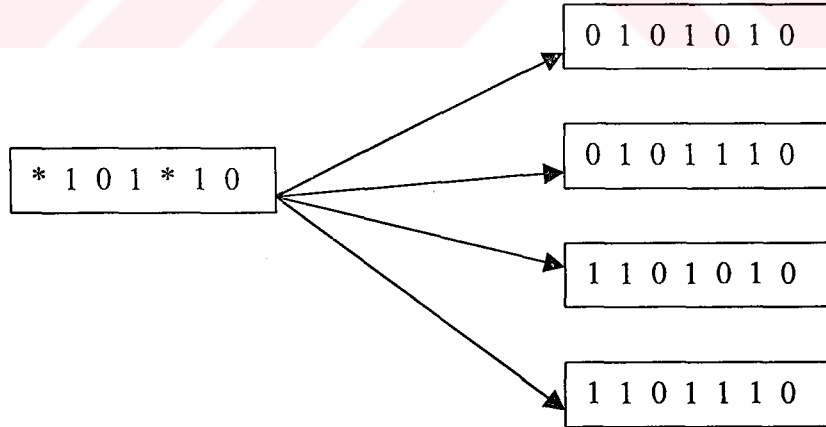
P_{imax} , P_i ' in mutasyon sonucunda uğrayabileceği en büyük değişme ve r_i ' ler, (0-1) aralığında rasgele seçilmiş sayılardır.

3.4. Genetik Algoritma Teori ve Hipotezleri

Holland'ın temel içeren teorisi ve buna bağlı olarak yapı taşları (building blocks) hipotezi, genetik algoritmaların her nüfusa gittikçe iyileşme göstermesi ve aranan nüfus daha fazla yakınsamasını açıklar.

3.4.1 Temel içerenler teorisi

Genetik algoritmaların tasarımı büyük ölçüde Holland'ın temel içerenler (schemata-schema'nın çoğulu) teorisine dayanır. Bir ya da daha fazla ortak özelliği olan bir grup diziyeye verilen isimdir. Temel içerenler dizi içine "önemseme" veya "dikkate alma" (don't care) anlamına gelen '*' işareti yerleştirilerek oluşturulur. İçindeki '*' lar dışında tüm değerleri aynı olan dizilerin tamamını bir temel içeren temsil eder. Temel içeren içindeki '*' ların sayısı r olduğunda her temel içeren tam olarak 2^r kadar diziyeye uyar. Şekil 3.5' de örnek alınan bir temel içeren ve buna uyan dört farklı dizi gösterilmiştir.



Şekil 3. 5 Örnek temel içeren ve buna uyan diziler

Buradaki '*' işaretinin sadece bir sembol olduğundan, dolayısıyla genetik algoritma tarafından kesinlikle işlenemeyeceğine dikkat edilmelidir. Bu sembol sayesinde belli

uzunluktaki diziler arasında olabilecek bütün benzerliklerin tanımlanabilmesi mümkün olur.

Dizi içindeki her pozisyon 1, 0 yada '*' olabileceğinden, olası farklı temel içeren sayısı, l dizi uzunluğu için 3^l kadardır. İlk bakışta uzunluğun 5 olduğu bir örnek için $2^5=32$ tane dizi yerine $3^5=243$ temel içeren ile ilgilenmek işleri zorlaştırıyor gibi görünüyor. Ancak dizilerin sadece kendileri göz önüne alındığında elimizde yalnızca dizi sayısı kadar bilgi var demektir. Oysa diziler, onların uygunluk değerleri ve diziler arasındaki benzerlikler de dikkate alındığında aramayı yönlendirebilecek pek çok bilgiye sahip olunur. Diziler arasındaki benzerlikleri hesaba katmakla ne kadar bilgi edinileceği o nüfus (popülasyon) içinde kaç tane temel içeren bulunduğu bağlıdır. Temel içerenlerin kesin sayısı o özel nüfus içindeki dizilerle ilgilidir. Beş bit uzunluğundaki bir dizi, her bit ya gerçek değerini ya da '*' sembolünü taşıyacağından 2^5 temel içeren içerir. O halde n elemanlı bir nüfus içindeki temel içeren sayısı l dizi uzunluğundaki dizilerin çeşitliliğine bağlı olarak 2^l ile $n2^l$ arasında bir rakamdır.

3.4.2 Yapı taşları hipotezi

Uygunluk değeri yüksek ve tanım aralığı kısa (short-defining-length) olan temel içerenler “yapı taşları” olarak adlandırılırlar. Temel içerenler içinde değeri belli olan bitler arasındaki uzaklık tanım aralığıdır. Aşağıdaki şekilde tanım aralığı 4 ve 1 olan iki farklı dizi görülmektedir.

1	*	*	*	0
---	---	---	---	---

(a)

*	*	1	1	*
---	---	---	---	---

(b)

Şekil 3. 6 Tanım aralıkları

- Tanım aralığı 4 olan dizi
- Tanım aralığı 1 olan dizi

Yapı taşları kavramı problemin karmaşıklığını azaltır. Dizi içinde bitlerin tamamının değerinin belli olduğu birkaç iyi bireyi tesadüfen bulmaya çalışmak yerine yapı taşlar; önemli olan bir bireyin küçük parçaları başka önemli bireylerin küçük parçalarıyla birçok kuşak boyunca birleştirilerek istenen uygunluk değerine yakınsayan birey veya bireyler üretir. Genetik algoritmalar bir kuşak içerisinde birçok temel içereni üzerinde işlem yapabilirler. "Doğrudan paralellik" (implicit parallelism) olarak adlandırılan bu özellikleri, etkili bir eniyileme işlemi sağlar.

Yapı taşları kuşaktan kuşağa gittikçe artar. Bunun görülebilmesi için önemli temel içerenlerin tekrar üreme, çaprazlama ve mutasyon işlemlerinden nasıl etkilendikleri incelenmelidir. Tekrar üremenin etkisi kolaylıkla görülür. Uygunluk değeri yüksek olan temel içerenlerin seçilme olasılığı da yüksek olduğundan bir sonraki kuşağa artan sayılarda iletilirler. Yalnızca tekrar üreme işlemi arama uzayında yeni noktalara ulaşma konusunda bir fayda sağlamayacağından çaprazlama işlemi uygulanır. Şekil 3.6. (a)' da görülen, tanım aralığı 4 olan temel içerenin bozulma olasılığı, Şekil 3.6. (b)' de yer alan tanım aralığı 1 olan temel içerenine göre daha yüksektir. Sonuç olarak tanım aralığı kısa olan temel içerenler çaprazlama sonrasında önemli bilgiler kaybedilmeden kalır ve tekrar üreme işlemiyle iyi bir örnekleme oranında seçilirler. Mutasyon ise belirli bir temel içereni bozacak kadar sık oranlarda değildir. O halde yapı taşları bozulmadan kuşaktan kuşağa geçerek nüfusun ortalama uygunluk değerini artırır ve çözüme ulaşmada önemli rol oynarlar (Forrest ve Mitchell 1993).

Çaprazlamanın görevi yapı taşlarının bozulmadan bilgi değişiminde bulunması ve daha yüksek uygunluk değerli yapı taşları oluşturmaktır. Mutasyon ise daha yüksek seviyelerde çeşitlilik ve uzayın farklı bölgelerine ulaşma imkanı sağlayabilir. Böylelikle yeni yapı taşları oluşturma olasılığı artar. Ancak bu, belirli yerlerdeki önemli bilgilerin kaybolmasına ve yapı taşlarının kaybedilmesine yol açabilir.

3.5 Benzetilmiş Tavlama

Benzetilmiş tavlama teknikleri ısıtılmış katı maddelerde soğuma esnasında kristallerin yeniden düzenlenmeleri gerçeğinden esinlenmiştir. Çağlar öncesinde demircilerin de fark ettiği gibi daha yavaş soğutma daha mükemmel kristal yapıya

yol açmaktadır. Bunun nedeni ısıl enerji tarafından sağlanan hareket etme ve yer değiştirme kolaylığının, atomların yerel en iyi şekillendirmeden kaçmalarına ve bunun yerine küresel olarak en iyi şekillendirmeye ulaşmalarına imkan vermesidir

3.5.1. Metallerde tavlama

Tavlama (annealing) terimi bir metalin belirli bir süre için sıcaklığının artırıldığı sonra yavaş yavaş soğutulduğu ısıl bir işlemi anlatır. Tavlama aşağıdaki amaçlar için yapılır:

- a. Stresi azaltmak
- b. Yumuşaklık, dayanıklılık ve esnekliği arttırmak
- c. Belirli bir mikro yapıyı oluşturmak

Tavlama sonucunda tekrar kristalleşme gerçekleşir ve metalin mekanik özelliklerini iyileştiren moleküler yapı değişiklikleri olur.

Tavlama üç aşamada gerçekleştirilir;

- a. İstenen sıcaklığa kadar ısıtma
- b. Belli bir sıcaklıkta tutma
- c. Soğutma

Soğutma işlemi genellikle oda sıcaklığına kadar olur. Bu işlemlerde zaman önemli bir parametredir. Sıcaklığın değişimi çok hızlı olursa metalde şekil bozukluğu hatta çatlama olabilir. Ayrıca gerçek tavlama süresi gerekli değişim hareketleri için yeterli derecede uzun olmalıdır. Tavlama sıcaklığı da yine önemli bir konudur. Sıcaklık artırılarak tavlama işlemi hızlandırılabilir.

3.5.2. Benzetilmiş tavlama prensipleri

Benzetilmiş tavlama (simulated annealing) termodinamik prensiplerine dayanan sayısal bir eniyileme tekniğidir. Metal parçacıkları, belli bir dereceye kadar ısıtıldıktan sonra sıcaklığın yavaş yavaş düşürülmesiyle gerçekleştirilen soğutma işlemi sırasında, bu yeni düşük enerji durumuna uygun olarak yeni bir dengeye girmek üzere kendilerini tekrar düzenlerler. Parçacıkların enerji durumlarının toplamı o metalin yapısı şeklinde düşünülebilir. Bir parçacığın herhangi bir enerji seviyesinde olma olasılığı Boltzman dağılımı kullanılarak hesaplanabilir. Metalin sıcaklığı düştükçe Boltzman dağılımı en düşük enerji seviyesine sahip parçacık yapısına doğru yaklaşır.

Bir sistemin yapısal parçacıkların olası enerji durumlarının toplamıdır. Bu durumda, sistem T sıcaklığında dengede ise sistemin verilen bir s yapısında bulunma olasılığı $\pi_T(s)$; E(s) enerji seviyesine ve Boltzman dağılımına bağlıdır.

$$\pi_T(s) = \frac{\exp\left[\frac{-E(s)}{kT}\right]}{\sum_{w \in S} \exp\left[\frac{-E(w)}{kT}\right]} \quad (3.17)$$

Denklem 3.17' deki Boltzman sabiti, S ise olası yapıların bütünüdür.

T sıcaklığında ısı dengede olan bir sistemin davranışının incelenmesi için i anındaki sistem yapısının q olduğu düşünülebilir, i+1 anındaki yapı için bir r adayı rasgele üretilir ve Denklem 3.18' de verilen p olasılığı dahilinde kabul edilir :

$$p = \frac{\pi_T(r)}{\pi_T(q)} = \exp\left[\frac{-(E(r) - E(q))}{kT}\right] \quad (3.18)$$

Eğer p 1' den büyükse ($p > 1$) bu, r' nin enerjisi kesinlikle q' nun enerjisinden düşük anlamına gelir ve i+1 anı için yeni yapı olarak r yapısı olarak kabul edilir. Eğer p 1' den küçük veya eşitse ($p \leq 1$), bu durumda r yapısı yeni yapı olarak p olasılığında kabul edilir. Bu yüzden daha yüksek enerji durumlarına ait yapılara da dönüş olabilir,

i zamanı sonsuza giderken sistemin s yapısının bulunma olasılığının $\pi_T(s)$ 'e eşit olduğu ve yapı dağılımının Boltzman dağılımına yakınlaştığı ispatlanmıştır. Buna göre düşük sıcaklıklarda Boltzman dağılımı gereğince düşük enerji durumları baskındır (Jeong 1996). Yeni yapı çok kötü olduğu yüksek enerjili durumların kötü olmayanlara oranla kabul edilme olasılıkları daha azdır. Bu özellik benzetilmiş tavlama algoritmasının yerel en iyiden uzaklaşmasına yol açar.

Düşük enerjili yapılar elde etmek için sadece sıcaklığı düşürmek yeterli değildir. Sistem sıcaklığının önce artırıldığı ve her sıcaklık derecesinde ısı dengeye ulaşılması için yeterli zaman verilerek kademeli olarak azaltıldığı bir tavlama işlemi uygulanmalıdır. Eniyileme problemlerinin tavlama işlemine benzer bir şekilde çözülebilmesi için aşağıdaki hazırlayıcı adımlar atılmalıdır (Jeong 1996) :

1. Eniyileme probleminin kendisi ile ilgili fiziksel kavramların benzerlikleri tanımlanmalıdır. Eniyileme probleminde enerji fonksiyonu amaç fonksiyon haline gelmekte ve parçacıkların yapısı da parametre değerlerinin yapısı haline gelmektedir. Düşük enerjili bir yapı bulmak en iyiye yakın bir çözüm bulma anlamına gelmekte ve sıcaklık ise benzetim için bir kontrol parametresi olmaktadır.

2. Uygun bir soğutma düzeneği seçilmelidir. Buna göre yapılacak soğutma işlemi sırasında bir dizi azalan sıcaklık değeri ve her sıcaklık derecesine harcanacak zamanın büyüklüğü belirlenmelidir.

3. Yeni yapılar yani yeni çözümler üretmek ve seçmek için bir yol belirlenmelidir.

Soğutma işlemi gerçekteki tavlama işlemine benzer şekilde yavaştır. Yakınsama için yeterli bir soğutma düzeneği denklem 3.19'da verilmiştir. Buna göre verilen bir sıcaklık sırası (T_i) ve c sabiti için $i \rightarrow \infty$ sistemin s yapısında bulunma olasılığı $\pi_0(s)$ ' dir.

$$i \rightarrow \infty \text{ ve } T_i \geq \frac{c}{\log i} \Rightarrow T_i \rightarrow 0 \quad (3.19)$$

Sıcaklık değerinin başlangıçta yüksek olması arama uzayında büyük adımlar atılmasına ve arama uzayının keşfedilmesine olanak verir . Metallerin fiziksel olarak tavlama işlemine benzer şekilde T sıcaklığının yavaş yavaş düşürülmesi, yüksek enerjili kötü bir yapısının yada eniyileme problemindeki düşük uygunluk değerli bir çözümün kabul edilme olasılığı zamanla azalır. Sıcaklığın düşürüldüğü soğutma düzeneği BT (Benzetilmiş Tavlama)' nin başarısı için önemli bir noktadır. Soğutma düzeneğinde yer alan iki önemli parametre seçilen yeni yapıyla eskisi arasındaki enerji farkı (adımın büyüklüğü), yani eski çözümle yeni çözüm arasındaki uygunluk değeri farkı ve T değeridir.

BT' nin en iyi özelliği yerel çözümden uzaklaşabilme kabiliyetidir. O yüzden genel çözümü bulabilme yeteneği başlama noktasına bağlı değildir. Ayrıca BT' nin uygulanması da oldukça kolaydır. Buna karşın başarısının adım büyüklüğü ve sıcaklık parametrelerine bağlı olması istenen bir durum değildir. Üstelik diğer eniyileme tekniklerine göre daha fazla amaç fonksiyon değerlendirmesi gerektirir. BT, "yönlü rasgele yürüme" (biased random walk) olarak düşünülebilir. Diğer eniyileme tekniklerinden farklı olarak, çözümler uzayında, daha iyi çözüme ulaşmak için belirli bir kurala bağlı hareketler gerçekleştirmek yerine rasgele seçilmiş adımlar atar.

BÖLÜM 4. AMAÇLANAN YSA SINIFLANDIRICISININ VE KONUŞMACI VERİTABANININ OLUŞTURULMASI

YSA'ların sınıflandırma özellikleri sayesinde, konuşmacı tanıma sistemlerinin vazgeçilmez unsurları haline gelmiştir. YSA'lar, insan beyninin çalışma prensibinin bilgisayar ortamında benzetilmesi şeklinde açıklanabilir. İnsan beyninde olduğu gibi YSA'lar da sinir hücrelerinin birbiriyle bağlantısını ve bilgi iletimini sağlayan sinaplardan oluşur. Bu biyolojik sinir hücreleri (işlem birimleri) ve sinapslar (bağlantılar) bilgisayar ortamında algoritmalar sayesinde bir benzetimin ürünü olarak geliştirilmiştir. Bu yüzden de yapay adını almıştır. Bağlantılar taşıdığı işareti bir işlem birimden diğerine iletilirler. Taşınan bu işaretler belli bir ağırlık değerine sahiptir. Bir işlem biriminin çıkışı; kendisine gelen ağırlıklarla, bu ağırlıkların bağlı olduğu girişlerin çarpımlarının toplamından oluşur. Bu toplam bir eşik değere ulaşıyorsa ilgili birim çıkış üretebilir, aksi halde çıkış üretemez.

Bu bölümde, YSA'ların öğrenme modları, öğrenme kuralları, konuşma tanıma alanında en çok kullanılan YSA sınıflandırıcıları ve bir alt bölümde tez çalışmasında kullanılması amaçlanan SOM sınıflandırıcısı açıklanacaktır.

YSA'lar genellikle katmanlardan oluşur, Şekil 4.1.'de birbirine tam bağlı üç katmanlı bir yapay sinir ağı modeli görülmektedir. Her katmandaki işlem birimi sayısı, uygulamanın özelliğine göre değişmektedir. Uygulamanın amacına göre bütün işlem birimleri birbirine tam bağlı olmayabilir, bazı ağırlıkların bir üst katmandaki işlem birimiyle olan bağlantısı yasaklanmış olabilir (İnal 1996).

Bir YSA eğitim sonunda, ağırlık çıkışında istenen değere göre ağırlıklarını ayarlayarak, girişine uygulanan örüntüleri öğrenebilir. YSA'ların öğrenme algoritmaları, değişik ağ yapılarına uygun öğrenme modu ve kurallarına göre hem hız hem de uygulama alanları bakımından farklılıklar gösterir.

4.1. Öğrenme Modları

İki çeşit öğrenme modu vardır: eğitici (supervised) ve eğitici (unsupervised) öğrenme modları. Eğitici öğrenme modu; adından da anlaşacağı gibi öğretici gerektirir. Bunu aksine eğitici öğrenme de ise ağın kendisi bir öğretici gibi kendi yaklaşımlarını kendisi oluşturur.

Eğitici öğrenme modunda, ağırlıklar, ilk olarak rasgele verilerek, ağ tarafından ayarlanır, bir sonraki iterasyonda, sinir ağının o anki çıkışı, istenen çıkışla karşılaştırılarak, ağırlıklar, oluşacak hatayı azaltacak şekilde ayarlanır.

Eğitici öğrenmede, sinir ağı bir sistem içinde aktif olarak kullanılmadan önce eğitilmelidir. Eğitim; ağa sunulan giriş ve çıkış verilerini içerir. Bu veriye eğitim seti adı verilmektedir. Bu set içinde, her giriş için istenen çıkış değeri yer almaktadır. Eğitim işlemi çok uzun sürebilir. Öğrenme işlemine fazla devam edildiğinde, ağırlıklar daha fazla değişmez. Bu tekniği kullanarak, bir ağ modeli; sınıflandırma, karar verme, bilgi ezberleme veya genelleme gibi işleri yapabilir.

Bir eğitici öğrenme algoritması işlem birimleri arasında işbirliği yapmalıdır. Böyle bir tasarımda, kümeler birlikte çalışarak birbirlerini uyarmaya çalışacaktır. Eğer dışardan bir giriş, kümedeki herhangi bir düğümü etkinleştirirse, bu kümenin tümünde etkinleşme artar. Tersine, bir düğüme gelen giriş, kümedeki etkinliği düşürücü yönde olursa, bu küme üzerinde girişin yasaklayıcı etkisi söz konusu olur.

Eğitici öğrenme modunda, ağın ağırlıkları dışarıdan etkiler kullanılarak ayarlanamaz. Bunun yerine, ağın performansı kendi kendine izlemesi söz konusudur. Ağ giriş işaretine ve ağın önceden belirtilmiş fonksiyonuna göre yenileme işlemi yapar. Ağa doğru ya da yanlış olduğu söylenmemiş olsa bile, ağ kendini nasıl organize edeceği hakkında yine de bir miktar bilgiye sahip olmalıdır. Ağ kendi çağrışımını kendi yapar.

Eğitici öğrenme modunda, işlem birimleri arasındaki yarışma, öğrenme için temel oluşturur. Rekabet halindeki kümelerin eğitimi, belirli grupların belirli uyarılara karşı

yanıtlarını kuvvetlendirir. Bu gruplar hem birbirleriyle hem de uygun bir yanıt ile ilişkilendirilir.

4.2. Öğrenme Kuralları

Bu kısımda aşağıdaki alt bölümlerde açıklanan öğrenme kurallarından genel olarak söz edilecektir.

4.2.1. En küçük kareler yöntemi

Bir işlem biriminin istenen çıkış değeri ve hedef değer arasındaki farkı (delta) minimuma indirmek için bu birimler arasındaki bağlantıların kuvvetliliğinin, sürekli olarak sağlanması fikri üzerine kurulu bir kuraldır. Verilmiş $\{x_1, x_2, \dots, x_L\}$ gibi bir X vektörü, W ağırlık seti ile bir y çıkış değeri üretecektir. Her bir giriş vektörü d_k hedef değerine sahip olacaktır. Burada $k=1 \dots L$ değerlerini almaktadır. Bir tek ağırlık vektörünün her bir giriş vektörü ile ilişkilendirilerek istenilen çıkış değerinin bulunması pek kolay değildir. Bu yüzden en küçük kareler yöntemi geliştirilmiştir.

4.2.1.1. Ağırlıkların hesaplanması

X_k giriş vektörüne göre, hedef vektör d_k ile optimum ağırlık seti W^* için, istenilen hedef çıkış ve o anki gerçek çıkış arasındaki fark minimize edilmelidir. Bu işlem her giriş vektörü için tekrarlanmalıdır. Burada amaç, giriş vektörlerine bağlı olarak oluşan karesel hatayı minimize etmektir.

Giriş vektörü k için o anki çıkış değeri y_k ise, ilgili hata ifadesi aşağıdaki eşitlikte görüldüğü gibi hesaplanır:

$$\varepsilon_k = d_k - y_k \quad (4.1)$$

Karesel hata veya beklenen hata değeri şu şekilde tanımlanır:

$$\langle \varepsilon_k^2 \rangle = \frac{1}{L} \sum_{k=1}^L \varepsilon_k^2 \quad (4.2)$$

Eşitlik 4.2’de L ; eğitim setindeki giriş vektörlerinin sayısıdır. Eşitlik 4.1 kullanılarak karesel hata ifadesi aşağıdaki gibi genişletilebilir:

$$\langle \varepsilon_k^2 \rangle = \langle (d_k - W' \cdot X_k)^2 \rangle \quad (4.3)$$

Şimdi de Eşitlik 4.3’teki ifadenin parantez karesini elemanlara dağıtırsak, aşağıdaki ifade elde edilmiş olur:

$$\langle \varepsilon_k^2 \rangle = \langle d_k^2 \rangle + W' \langle X_k X_k' \rangle W - 2 \langle d_k X_k' \rangle W \quad (4.4)$$

$R = \langle X_k X_k' \rangle$ giriş korelasyon matrisi, bir $P = \langle d_k X_k' \rangle$ vektörü ve $\xi = \langle \varepsilon_k^2 \rangle$ şeklinde tanımlanmış olsun. Bu ifadeleri kullanarak Eşitlik 4.4 aşağıdaki şekilde tekrar yazılabilir:

$$\xi = \langle d^2 \rangle + W' R W - 2 P' W \quad (4.5)$$

Bu eşitlik ξ ‘yi , W ağırlık vektörünün bir fonksiyonu şeklinde gösterir. Bir başka deyişle $\xi = \xi(W)$ ‘dir. En küçük kareler yöntemine göre ağırlık vektörünü bulmak için:

$$\frac{\partial \xi(W)}{\partial W} = 2 R W - 2 P \quad (4.6)$$

$$2 R W^* - 2 P = 0 \Rightarrow R W^* = P \quad (4.7)$$

$$W^* = R^{-1} P \quad (4.8)$$

W^* ifadesi ağırlıkların çözüm kümesini göstermektedir.

ξ skaler olmasına rağmen Eşitlik 4.6’daki ξ ‘nin kısmi türev ifadesi ise bir vektördür: Eşitlik 4.6’ da ξ ‘nin gradientidir, yani $\nabla \xi$ bir vektördür:

$$\nabla \xi = \left[\frac{\partial \xi}{\partial W_1}, \frac{\partial \xi}{\partial W_2}, \frac{\partial \xi}{\partial W_3}, \dots, \frac{\partial \xi}{\partial W_n} \right]' \quad (4.9)$$

Bütün bu işlemler, $\xi(W)$ fonksiyon eğiminin sıfır olduğu bir noktayı gösterir. Genelde bu nokta genel minimal ya da maksima olabilir.

4.2.1.2. En dik eğim yöntemiyle ağırlıkların (W^* 'nin) bulunması

Ağırlık yüzeyinden rasgele seçilmiş bir ağırlık noktası, en dik eğim yönü aşağı doğru olacak şekilde belirlenir. Bu işlem genel bir minimuma ulaşınca kadar tekrarlanır. En dik eğimin yönü her noktadaki değişimlere diktir ve bu yön daima minimum noktayı göstermez. Bu işlemde ağırlık vektörü değişken olduğundan, ağırlık değişimleri t zaman adımlarına göre gösterilmiştir. İlk ağırlık vektörü $W(0)$ ve t anındaki ağırlık vektörü de $W(t)$ ile gösterilmektedir. Her adımda bir sonraki ağırlık vektörü aşağıdaki gibi hesaplanacaktır:

$$W(t+1) = W(t) + \Delta W(t) \quad (4.10)$$

Eşitlik 4.10'da $\Delta W(t)$; t zaman adımındaki W'nin değişimidir. Yüzeydeki her noktanın en dik eğim yönüne bakılmalıdır. Bu yüzden yüzeyin gradientini hesaplayarak en dik yukarı eğim yönü bulunabilir. Sonrada gradientin negatifi alınarak en dik eğimin yönü belirlenmiş olur. Değişimin büyüklüğü, gradientin μ gibi bir sabitle çarpılarak aşağıdaki ifade elde edilebilir:

$$W(t+1) = W(t) - \mu \nabla \xi(W(t)) \quad (4.11)$$

Her bir iterasyonda $\nabla \xi(W(t))$ değerinin belirlenmesi gerekir. Eşitlik 4.6 ve 4.9, $\nabla \xi(W(t))$ 'nin belirlenmesinde kullanılabilir. Fakat, W^* 'nin analitik belirlenmesi yapılmalıdır. Bunun içinde R ve P değerlerinin bilinmesi gerekir. Bunu gerçekleştirmek çok zor olduğu için, gradientin bulunması için aşağıdaki gibi bir yaklaşım yapılmıştır.

İterasyondaki her adım için, sırasıyla aşağıdaki maddeler gerçekleştirilmelidir:

1. Ağ modelinin girişlerine X_k , giriş vektörü uygulanır.

2. $\epsilon_k^2(t)$, hatanın karesi belirlenir, o anki ağırlık vektörünün kullanılmasıyla aşağıdaki eşitlik elde edilir:

$$\epsilon_k^2(t) = (d_k - W^1(t)X_k)^2 \quad (4.12)$$

3. $\epsilon_k^2(t)$ 'nin yani hatanın gradientini hesaplamak için Eşitlik 4.3'teki gibi beklenen hata değerinin ($\langle \epsilon_k^2 \rangle$) kullanılmasıyla aşağıdaki eşitlikler elde edilir:

$$\nabla \epsilon_k^2(t) \approx \nabla \langle \epsilon_k^2 \rangle \quad (4.13)$$

$$\nabla \epsilon_k^2(t) = -2\epsilon_k(t)X_k \quad (4.14)$$

4. Eşitlik 4.14'teki gradient formülünü kullanarak Eşitlik 4.11'e göre ağırlık vektörü aşağıdaki gibi yenilenir:

$$W(t+1) = W(t) + 2\mu\epsilon_k X_k \quad (4.15)$$

5. 1'den 4'e kadar olan adımlar diğer giriş vektörleri ile, hata kabul edilebilir bir değere indirilene kadar ya da belli bir sayıda tekrarlanır.

Eşitlik 4.15, en küçük kareler yöntemini ifade eder. μ parametresi ağırlık vektörünün minimum hataya yakınsama hızını belirtmektedir. Her iterasyonda ağırlık vektöründeki değişimler küçük değerli tutulmalıdır. Eğer değişimler çok büyük olursa, ağırlık vektörü aranan genel bir minimum (global minima) nokta bulamaz ya da kazara yakınsayabilir. μ parametresi bu kararsız durumu ortadan kaldırır.

4.2.2. Geri yansıtma öğrenme kuralı

Hataların geri yansıtılması tekniği; Delta öğrenme kuralının genelleştirilmiş halini kullanır. Bu işlem iki fazda gerçekleşmektedir. İlk faz, “ileri doğru faz” olup, giriş vektörü sunulduktan sonra ve ileriye doğru ağıya yansıtılarak her işlem birimi için geçerli tüm çıkışlar hedeflenen çıkış değeri ile karşılaştırılıp, hata ifadesi hesaplanır. İkinci fazda ise “geriye doğru faz” olup, birinci fazda hesaplanmış olan hata ifadesi şimdi de geri yönde işleme konur. Bu iki faz tamamlandıktan sonra yeni girişler ağıya işlenmek üzere sunulur.

Her bir düğümün kendisinden önce gelen katmana, ne kadar hata oluşturduğunu anlamak için; bir düğümün ileri doğru fazda her girişe göre hatayı ne kadar arttırdığını hesaplamak mümkündür. Daha sonra, bu hatayı geriye doğru yansıtma koşulu ile ağırlıkları bu iki faz sonucunda uygun değere ayarlamak mümkündür.

Geri yansıtma kuralı yavaş olup, hem eğitim işlemi hem de test işlemi aynı anda yapılamaz. Öncelikle ağı eğitimi belirli kriterlere göre eğitildikten sonra kullanılır. Eğitim sırasında hatanın gradient ifadesi iki nokta arasına takılıp, osilasyona girip lokal bir minimuma saplanıp kalabilir.

4.2.3. Yarışmacı öğrenme kuralı

Bu kural sadece eğitimcisz öğrenme ağı uygulamalarında kullanılmaktadır. Bu yöntemde her işlem birimi çıkış üretmek için rekabet eder. En büyük çıkışa sahip işlem birimi “kazanan işlem birimi” olarak anılır ve diğer işlem birimlerini yasaklayabilme özelliğine sahiptir. Sadece kazanan işlem biriminin çıkış vermesine izin verilmiştir ve sadece kazanan işlem birimi ile ona komşuluk eden birimlerin ağırlıklarının uyarılmasına izin verilir. Yarışmayı kazanan işlem biriminin çıkışına “1” değeri atanırken diğer tüm işlem birimi çıkışları “0” değerini alır. Bu durum aşağıdaki eşitlikte görüldüğü gibi ifade edilebilir:

$$y_k = \begin{cases} 1 & \text{eğer } v_k > v_j \text{ ve } j \neq k \\ 0 & \text{aksi halde} \end{cases} \quad (4.16)$$

Eşitlik 4.16 'da v_k parametresi k. işlem birimine gelen tüm girişlerin oluşturduğu bağlantının ifadesidir.

w_{kj} ; j. işlem birimini k. işlem birimine bağlayan ağırlıkları ifade etsin. Bu ağırlıklar, ağ ilk tanımlandığında, tümünün pozitif bir değere sahip olduğu ve aşağıdaki eşitlikte görüldüğü gibi, tüm bağlantıların ağırlık değerleri birbirine eşittir:

$$\sum w_{kj} = 1 \quad (4.17)$$

Girişler ağa sunuldukça yarışı kazanan işlem birimine bağlı ağırlıklar her defasında yenilenecek öğrenme sağlanmış olur. N ayrık zaman adımında $w_{kj}(n)$ ağırlığı öğrenme sonucunda aşağıdaki eşitlikte görüldüğü gibi yenilenir:

$$\Delta w_{kj} = \begin{cases} \eta(X_j - w_{kj}(n)) \\ 0 \end{cases} \quad (4.18)$$

Eşitlik 4.18'de η öğrenme oranını, X giriş örüntülerini ifade etmektedir.

Algoritmada kullanılan η öğrenme oranı parametresi, η_0 gibi bir ilk değerden başlayarak, n ayrık zaman değeri sonunda, asla sıfır olmadan, sıfıra doğru azalır. Öğrenme oranı parametresi aşağıdaki gibi tanımlanabilir:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau}\right), \quad n = 0, 1, 2, \dots \quad (4.19)$$

Eşitlik 4.19'da τ zaman sabitidir. Eşitlik 4.18 ve 4.19 kullanılarak bir sonraki ayrık zaman için yeni ağırlıklar aşağıdaki gibi hesaplanır:

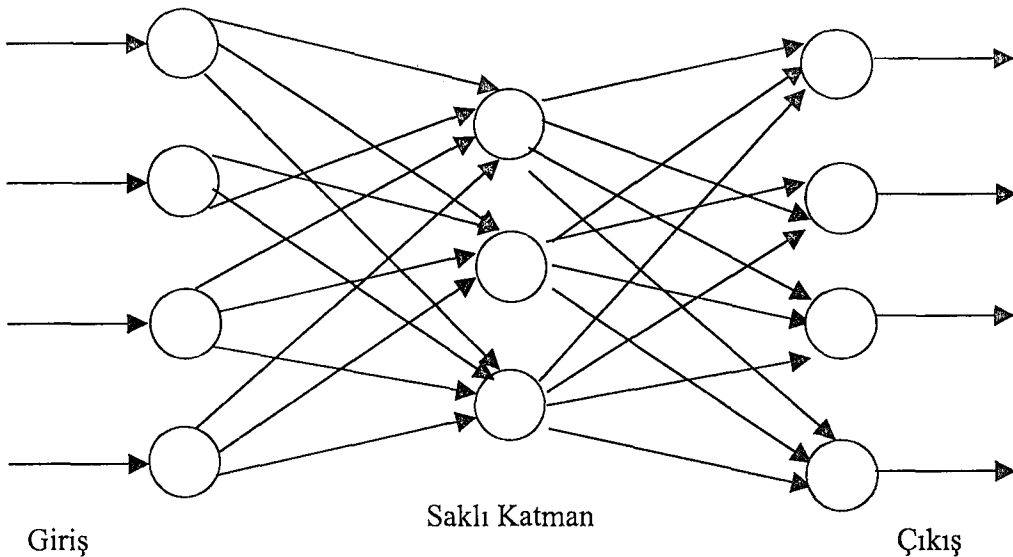
$$w_{kj}(n+1) = w_{kj}(n) + \eta(X_j - w_{kj}(n)) \quad (4.20)$$

4.3. YSA Sınıflandırıcı Modelleri

İstatistiksel örüntü tanıma tabanlı sınıflandırıcıların, öbikleme (clustering) işlemine alternatif bir yöntem ise YSA sınıflandırıcılarıdır. Bunların en popüler olanları ÇKA (Çok Katmanlı Almaç) modeli, kendi kendini organize eden ağ modeli (Self Organizing Map-SOM), YAM (Yapay Ağaç Modeli) , Öngörülü YSA modelidir. Aşağıdaki alt kısımlarda tez çalışmasında ve konuşmacı tanıma alanında en yaygın kullanılan sınıflandırıcılar anlatılmıştır.

4.3.1. Çok katmanlı almaç - ÇKA

Genellikle standart geri yansıtımlı eğitim algoritmasının kullanıldığı tipik bir Çok Katmanlı Almaç yapısı Şekil 4.1’de görülmektedir. Geri yansıtımlı öğrenme algoritması; önceden tanımlanmış giriş ve çıkış örüntü setlerini iki fazda “yansıtma-uyarlama” öğrenir. Ağın girişine örüntü uygulandıktan sonra, diğer üst katmanlara çıkış üretilinceye kadar yansıtılır. Yansıtma sonucunda elde edilen çıkış, daha sonra istenen çıkışla karşılaştırılır ve oluşan hata, her çıkış birimi için hesaplanır. Hesaplanan bu hata değeri, daha sonra çıkış katmanından bir önceki katmandaki işlem birimlerinin ağırlıklarını yenilemek üzere geriye yansıtılır. Yansıtma işlemi giriş katmanına ulaşınca kadar tekrarlanır. Yukarıdaki bütün bu işlemler, eğitim setindeki örüntülerin tümü için tekrar edilir.

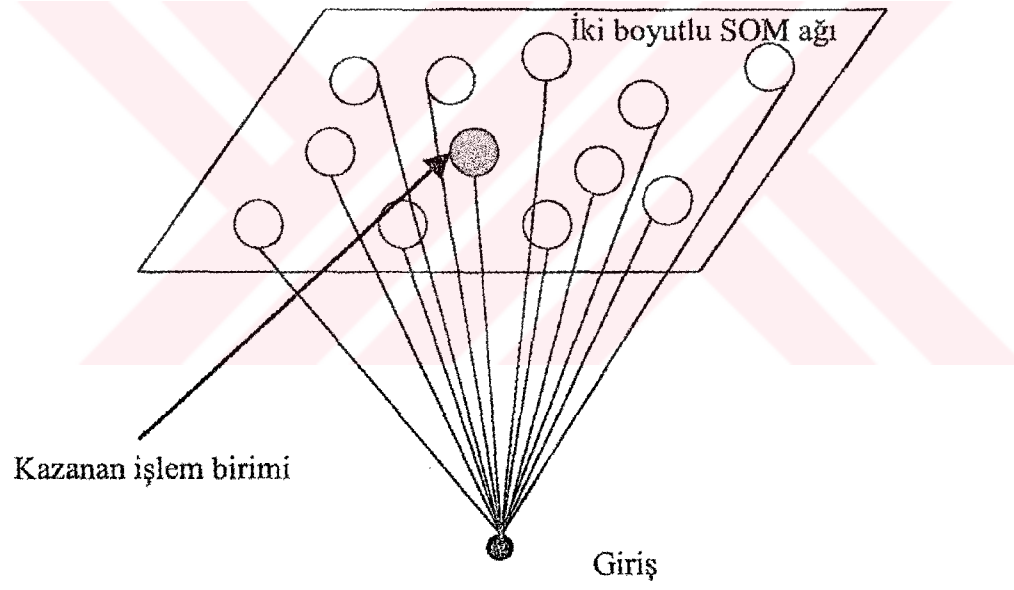


Şekil 4. 1: Birbirine tam bağlı üç katmanlı bir ÇKA modeli.

Şekil 4.1'deki gibi bir ÇKA, sınırlı giriş değerlerine sahip problemlerin çözümü için uygun bir modeldir. Fakat, uzun süreli eğitim gerektirmesi, algoritmanın bir genel çözüm noktası yerine bir yerel minimum noktaya saplanıp kalması da olumsuz taraflarıdır.

4.3.2. Kendi kendini organize eden ağ modeli

Kohonen özellik haritası olarak da bilinen SOM (Self Organizing Map); girişleri, farklı sınıflara ayırma işleminde kullanılır. Giriş katmanına uygulanan her yeni girişe göre ağın çıkışları değiştirilerek öğrenme sağlanmış olur. Her işlem biriminin diğer işlem birimlerine karşı yarıştığı ve bu yarışta, en yüksek skorlu çıkışı üreten işlem biriminin kazandığı öğrenme işlemi; yarışmacı öğrenme algoritması olarak özetlenebilir. Şekil 4.2'de temel SOM yapısı görülmektedir.



Şekil 4. 2 Kohonen Ağı (SOM)

Şekil 4.2'ye göre SOM ağına uygulanan bir girişe göre; yarışta kazanıp "1" çıkışını üreten işlem birimi diğer çıkışları yasaklayabildiğinden dolayı, diğer işlem birimlerinin çıkışları "0" olur. SOM ağının test aşamasında, bir konuşmacıya ait konuşma verisi önceden her konuşmacı için eğitilmiş SOM ağlarına uygulanır. SOM

ağının çıkışı ilgili konuşmacı verisine göre hangi işlem birimlerinin çıkış üreteceğini söylemektedir. Eğer SOM ağı yeterince iyi eğitilmiş ise, ilgili konuşmacı için belirli çıkışları etkin olacaktır. Bu çıkışlara göre hangi konuşmacının seçileceğine karar verilmelidir.

4.4. Yapay Ağaç Ağ Modeli (YAM)

YAM modeli, karar ağaç yapısı ile ileri beslemeli ağ modeli özelliklerinin birleştirilmesinde oluşan bir sınıflandırıcıdır. Ağaçtaki her düğüm, tek katmanlı almaç (TKA) yapısındadır. YAM mimarisi eğitim süresi boyunca değişebildiğinden kendi kendini organize etme özelliği vardır. YAM, sıralı karar işlemini uygulamak üzere bir ağaç mimarisini kullanır. YAM'ın her seviyesindeki düğümler, giriş vektörlerini özel bölümlere ayırır. Böylece YAM'ın giriş uzayını homojen alt kümelere böler, böylece her dalın düğümü tek bir sınıfı ifade eder.

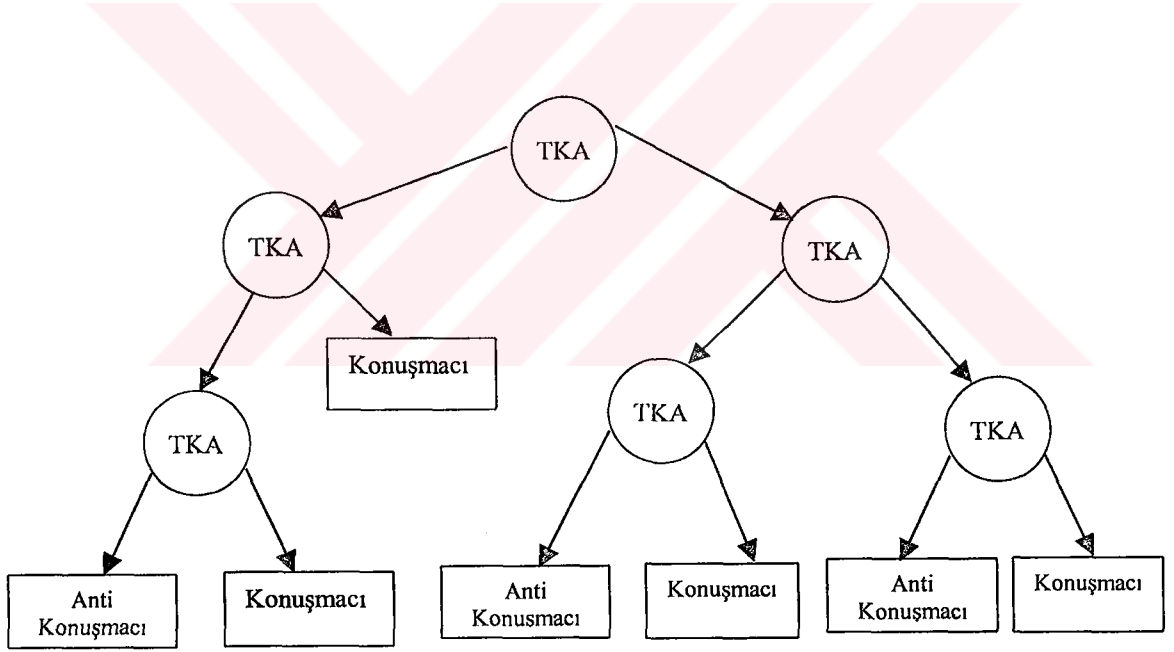
YAM yinelemeli eğitimi aşağıdaki gibidir: Bir düğümdeki eğitim veri seti için, eğer verilen veri bu düğümdeki sınıfa ait ise ilgili düğüm başka kol ya da kollar oluşturmaz. Aksi durumda, eğitim verisi, bu düğümün çocuklarını oluşturacak, birkaç alt kümeye bölünür. Bu işlem tüm eğitim verisi için tekrarlanır. Eğitilmiş bir YAM'ın eğitim seti üzerindeki performansı %100 olacaktır. Fakat, aşırı eğitim yüzünden, YAM en iyi genelleştirmeyi yapmayabilir. YAM mimarisindeki bazı kolların budanması (pruning) ile en uygun genelleştirme sağlanabilir. Budama işlemi, elverişsiz veriyi içeren alt dalların kaldırılmasıyla sağlanır.

Konuşmacı tanıma sistemi için, ikili yapay ağaç modeli her konuşmacı için oluşturulur. Eğitimde kullanılacak özellik vektörleri, her konuşmacı için oluşturulur. Eğitimde kullanılacak özellik vektörleri, her konuşmacının telaffuzlarından üretilir ve etiketlenilerek, eğitim seti oluşturulur. YAM'a uygulanan eğitim verisinin tümü sınıflandırılınca kadar, yinelemeli bir şekilde hem öğrenmeyi hem de mimarisini geliştirir. Tipik ikili yapay ağaç modeli Şekil 4.3' te görülmektedir.

Şekil 4.3' te "Konuşmacı" etiketli kollar, "bir" değerine sahip sınıfları, "Anti-Konuşmacı" etiketli kollar ise, "sıfır" değerine sahip sınıfları göstermektedir. Şekil

4.3' e göre, bir sınıfa ait eğitim verisinin aynı sınıfı ifade eden kola karşılık gelmesi gerekmektedir, muhtemelen birkaç kola dağılacaktır.

Konuşmacı saptama için, test vektörleri, tüm konuşmacıların eğitilmiş YAM modeline uygulanır ve test vektörlerinin her modele ait olma yüzdeleri hesaplanır. Eğer uygulama kapalı konuşmacı setine göre yapılıyorsa, en büyük yüzdeliğe sahip modelin konuşmacısı seçilir. Eğer açık konuşmacı seti uygulaması yapılıyorsa, en büyük yüzdelik değer bir eşik değer ile karşılaştırılır. Eğer yüzdelik değer eşik değerinin üstünde ise konuşmacı seçilir, aksi halde sınıflandırıcı; bu konuşmacıyı, "konuşmacı seti dışındadır" şeklinde nitelendirilecektir. Konuşmacı doğrulama için, tanınması istenen konuşmacıya ait olma yüzdeleri, önceden saptanan bir eşik değerini aşıyorsa konuşmacı doğrulanmış olacaktır.



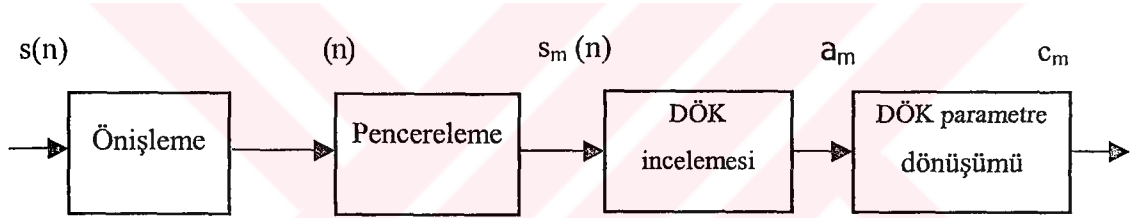
Şekil 4.3 Yapay Ağaç ağ Modeli - YAM

4.5. Amaçlanan Yöntem ve Veritabanının Oluşturulması

Bu bölümde, Şekil 2.2' deki genel konuşmacı tanıma sistemindeki aşamalar ve konuşmacı veritabanı oluşturma aşamaları anlatılacaktır.

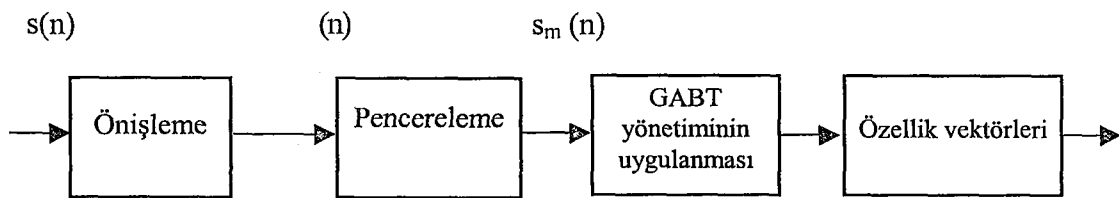
4.5.1. Özellik çıkartım aşaması

Şekil 2.2' deki özellik çıkartım aşamasında DÖK incelemesi yapılacak konuşma işaretinin DÖK tabanlı parametre takımları bulunmadan önce, konuşma işaretine bir takım ön işlemler uygulanmıştır. Bu işlemlerin ve DÖK incelemesinin yer alacağı çıkartım aşamasının blok diyagramı Şekil 4.4' de görülmektedir. Benzer şekilde GABT yöntemi özellik çıkartımı yapmak üzere Şekil 4.5' teki aşamalardan yararlanılmıştır.



Şekil 4. 4 DÖK incelemesi özellik çıkartım aşamaları.

Benzer şekilde GABT yöntemi, özellik çıkartımı yapmak üzere Şekil 4.4' teki yapının bir benzeri DÖK incelemesi ve DÖK parametre dönüşümü blokları yerine kullanılarak aynı konuşma işaretleri için GABT tabanlı özellik vektörleri üretilmiştir.



Şekil 4.5 GABT yöntemi ile özellik çıkartım aşamaları

4.5.1.1 Önışleme aşaması

Sayısallaştırılmış konuşma ışıareti $s(n)$, aşığıdaki transfer fonksiyonu verilen birinci dereceden sayısal süzgeçten geçirilerek, ışıaretin ani frekans deęişimlerine baęlı yüksek seviyeli genlik deęişimlerini bastırarak, ışıaretin ani deęişim gösteren geçiş bölgeleri yumuşatılmıştır.

$$H(z) = 1 - \bar{a} z^{-1} \quad (4.28)$$

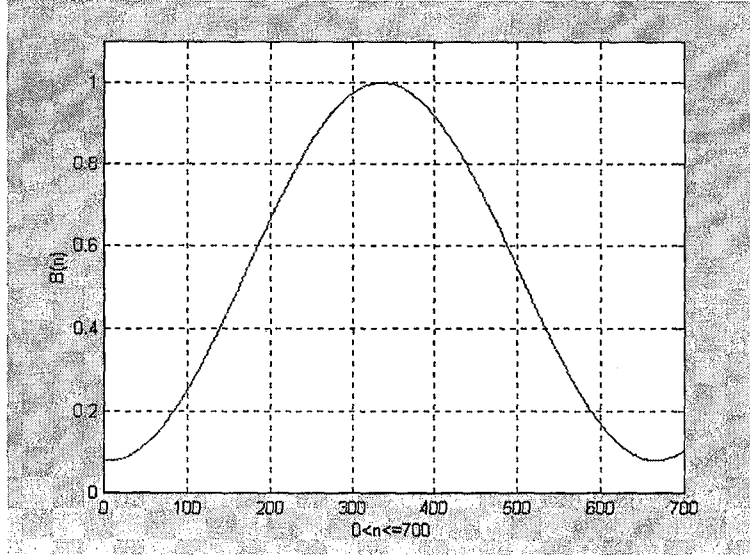
Önışleme girişine uygulanan $s(n)$ konuşma ışıaretine göre, çıkışta elde edilen (n) ifadesinin fark denklemi, aşığıdaki eşitlikle görüldüğü gibidir:

$$s(n) = s(n) - \bar{a} s(n-1) \quad (4.29)$$

4.5.1.2. Pencereleme aşaması

Bir sonraki özellik çıkartım aşaması, sayısal süzgeçten geçirilmiş konuşma ışıaretinin, pencerelenmesi işlemdir. Pencereleme işleminin amacı, konuşma ışıaretinin her bir bölütünün başlangıç ve bitimi arasındaki süreksizliği minimuma indirmektir. Başka bir deyişle ilgili bölütün başlangıç ve bitişini sıfıra yakın bir noktadan başlayarak deęişimini belirginleştirmiştir.

Pencereleme işlemi; ışıareti tekrarlayan (overlapping) pencereler şeklinde böler. Şekil 4.6' da örnek olarak 700 nokta için Hamming penceresinin deęişim şekli görülmektedir. Şekil 4.6' da Hamming penceresinin başlangıç ve bitiş deęerlerinin sıfır olmadığı görülmektedir. Böylece ışıaretin anlamlı olabilecek deęerleri tekrarlanan pencereler boyunca korunmuş olacaktır.



Şekil 4.6 Hamming pencereleme işlemi örneği.

$B(n)$ pencere katsayıları kullanılarak, konuşma bölütünün uzunluğu N olan ($0 \leq n \leq N-1$) her konuşma bölütü için işaret aşağıdaki eşitlikteki gibidir:

$$s_m(n) = B(n) s(n) \quad (4.30)$$

Uygulamalarda kullanılan tipik Hamming pencereleme yönteminin eşitliği aşağıdaki gibidir:

$$B(n) = 0.54 - 0.46 \cos(2\pi n / (N-1)) \quad (4.31)$$

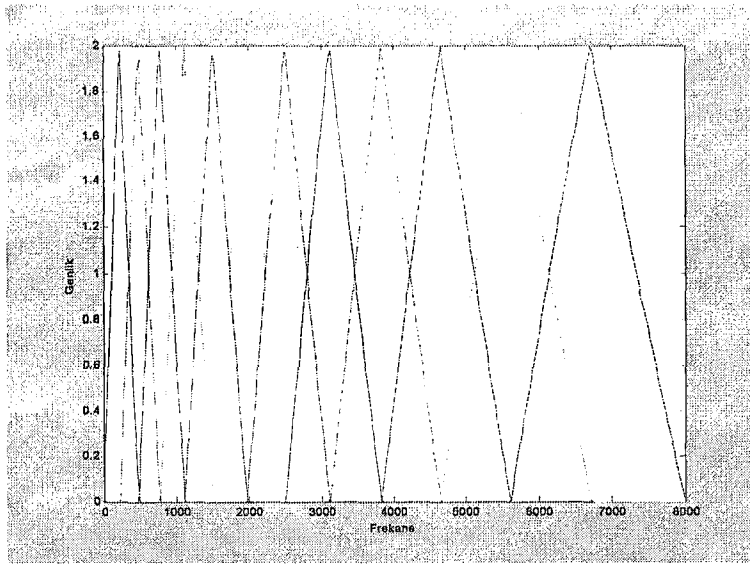
Hamming penceresi dışında Hanning, Barlett gibi pencereler de vardır.

Bir sonraki özellik çıkartım aşaması ise DÖK incelemesi olup Bölüm 2.1.3'te DÖK modelinde açıklandığı gibi a parametre takımındaki katsayılar, konuşma işaretinin her $s_m(n)$ bölütü için bulunmuştur. Hesaplanan bu a_m parametre takımındaki katsayılar kullanılarak, son aşama olan DÖK parametre dönüşüm aşamasında, Eşitlik 2.8'deki yinelemeli yöntemle göre c_m parametre takımı hesaplanmıştır. Ayrıca Bölüm 3' te söz edilen GABT yöntemi ile de pencerelenmiş konuşma işaretine ait parametre takımı katsayıları elde edilmiştir

4.5.1.3. Mel-frekans ölçekli kepsral katsayıların elde edilmesi

Mel-frekans ölçekli kepsral katsayılar (Mfcc) ses işlemede sıkça kullanılır (Krishnamoorthy, 1998). Mel-frekans ölçekli kepsral katsayıların elde edilmesi aşağıdaki işlem basamakları ile gerçekleştirilir;

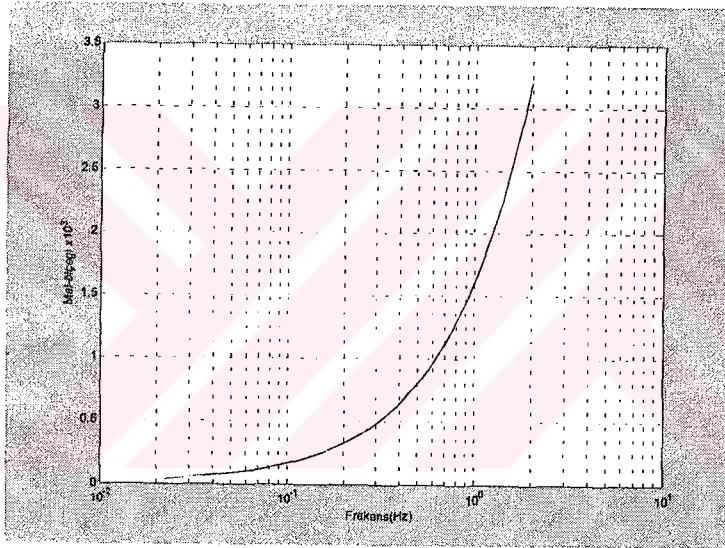
- Konuşma işareti birbiri içine geçmiş tekrarlanan (overlap) penceler şekilde bölütlere ayrılır.
- Bölütler, Eşitlik 4.30' daki gibi yöntemlerinden biri ile (genellikle Hamming yöntemi) etkiletilir..
- Her bölüte Ayrık Fourier Dönüşümü (AFD) uygulanıp güç spektrumu elde edilir.
- Elde edilen güç spektrumu Şekil 4.7' deki gibi mel-boşluklu filtre bankasından geçirilir.
- Filtre çıkışındaki elde edilen katsayıların değerlerin logaritmasının ayrık kosinüs dönüşümü (AKD) alınarak, Mel-frekans ölçekli kepsral katsayılar (mfcc) hesaplanmış olur.



Şekil 4.7 Mel-frekans ölçekli filtre bankasının genlik frekans cevabı.

Her bir filtrenin genlik frekans cevabı Şekil 4.7' deki gibi üçgen şeklindedir ve bitişik iki filtrenin merkezinden sıfıra doğru doğrusal olarak azalır. Mel-frekans ölçeği temelde gerçek frekans (Hz) ile algılanan frekans arasında bir eşleştirme işlemine dayanır. İnsan duyu sistemi doğrusal anlamda frekansları algılamadığı için mel-frekans ölçeğinin gerçek frekans (Hz) yerine kullanılması daha anlamlıdır (İnal 2002) Şekil 4.8' de görüldüğü gibi 1 kHz' in altındaki eşleme yaklaşık olarak doğrusal, üstündeki ise logaritmik kabul edilir. Eşitlik 4.32'de deneysel yolla bulunmuş bu ilişki ifade edilmiştir:

$$F_{mel} = \ln\left(\frac{1 + F_{Hz}}{700}\right) * 1127.01 \quad (4.32)$$



Şekil 4.8 Mel-ölçek fonksiyonu.

4.5.2. Veritabanının oluşturulması

Tez çalışmasının uygulama aşamasında kullanılacak olan konuşmacı veritabanı Bölüm 2.3'te de açıklanmış olan CSLU veritabanından yararlanılarak elde edilmiştir.

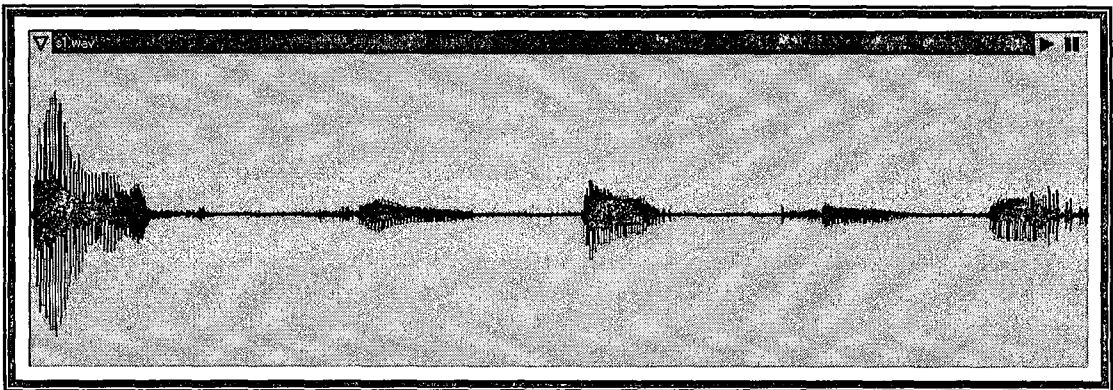
CSLU, Bölüm 2.3'te de anlatıldığı gibi Amerikanın çeşitli eyaletlerinden elde edilmiş telefon ve cep telefonu görüşmelerinin tutulduğu bir veritabanıdır. Tez çalışmasında bu veritabanı içinden seçilmiş olan toplamı 10 olmak üzere, 5 bayan 5 erkek konuşmacı kullanılmıştır. Toplam 12 olmak üzere, 6 adet eğitim, 6 adet test

erkek konuşmacı kullanılmıştır. Toplam 12 olmak üzere, 6 adet eğitim, 6 adet test için ses dosyası kullanılmıştır. Konuşmacılar “5 3 8 2 4”, “4 0 7 1 3”, “1 9 0 5 4” rakamlarının telaffuzlarını ikişer defa farklı oturumlarda söylemişlerdir. Bu telaffuzlar eğitim için kullanılmıştır, ayrıca “6 1 0 9 7”, “2 8 3 7 6”, “0 5 2 3 9” rakamlarının telaffuzlarını da yine aynı şekilde ikişer defa farklı oturumlarda söylemişlerdir, bu telaffuzlarda test amaçlı kullanılmıştır. Böylece 10 konuşmacıya ait toplam 120 telaffuz elde edilmiştir.

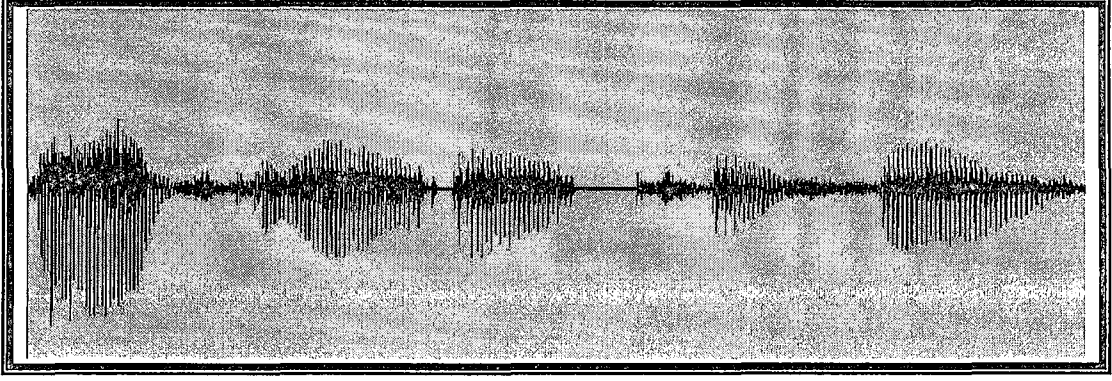
Bu telaffuzlar 10 konuşmacı için farklı oturumlarda telaffuz ettikleri farklı rakamların söylenmesinden elde edilmiştir. Uygulamada eğitim ve test vektörleri için farklı telaffuzlar kullanıldığı için yapılan çalışma metinden bağımsız kapalı set konuşmacı tanıma uygulamasıdır.

Veritabanındaki 5 erkek ve 5 bayan konuşmacı erkekler için M1, M2, M3, M4, M5 , bayanlar için F1, F2, F3, F4, F5 olacak şekilde etiketlenmişlerdir. Telaffuzların her biri “egtm” ve “test” şeklinde ve her bir telaffuzda F1, F2, F3, F4, F5 ve M1, M2, M3, M4, M5 şeklinde ayrı klasörler içinde tutulmaktadır. Sonuçta her bir eğitim ve test vektörü için oluşturulacak olan özellik vektörleri veya gerekli ağ modelleri kendi klasörleri içinde saklanmış olacaktır.

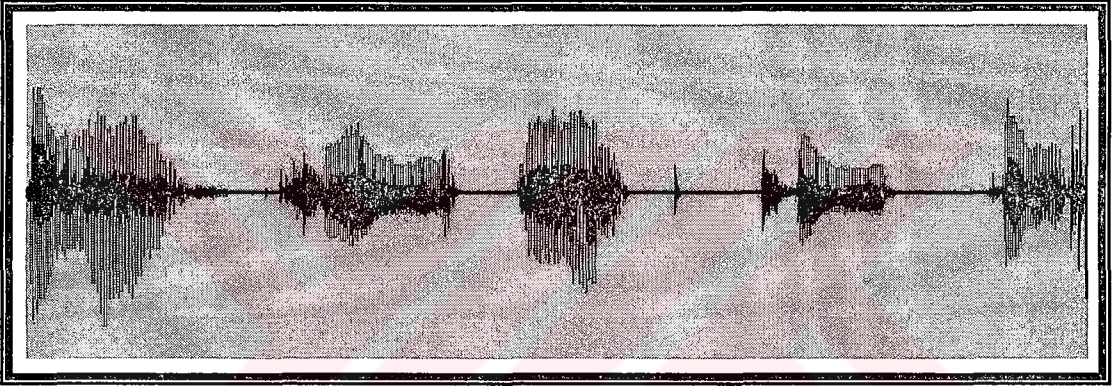
Telaffuzların örnek dalga şekilleri Şekil 4.9. – 4.18’ da gösterilmiştir. Bu dalga şekilleri telaffuzların zamana göre genliklerini ifade etmektedir.



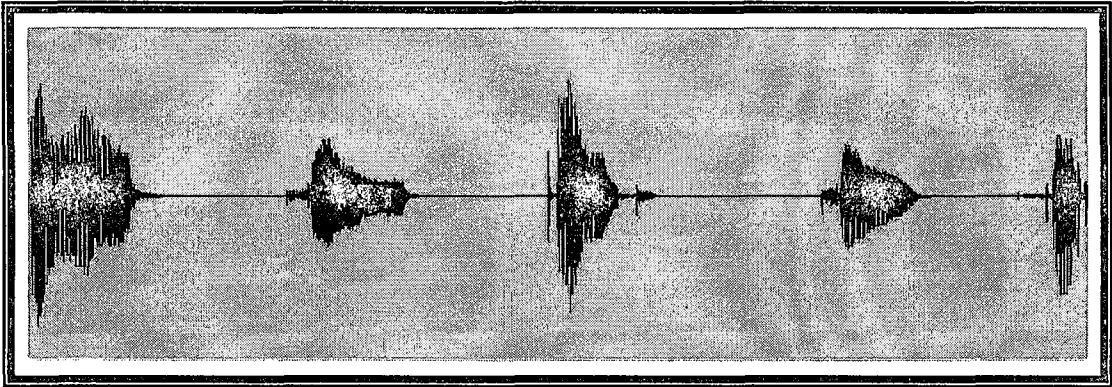
Şekil 4. 9 F1 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



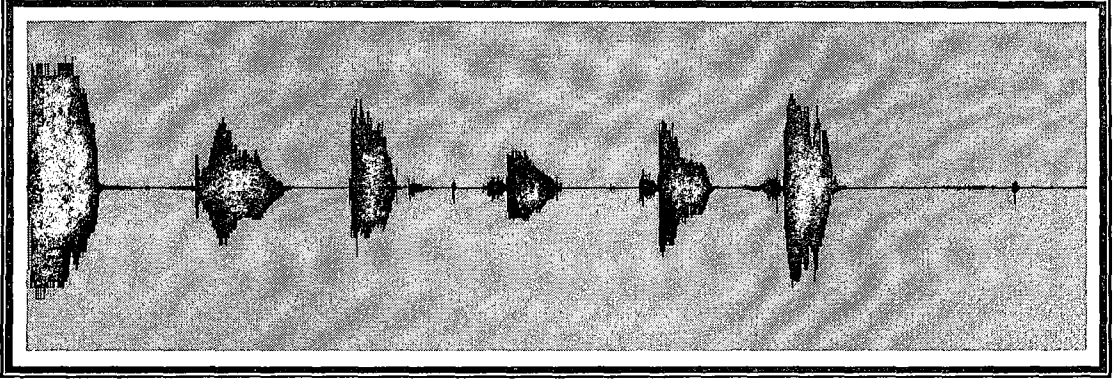
Şekil 4. 10 F2 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



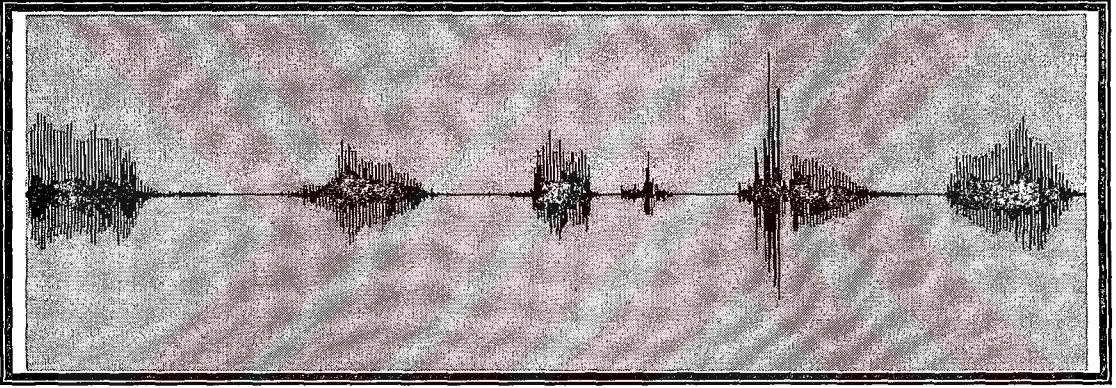
Şekil 4. 11 F3 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



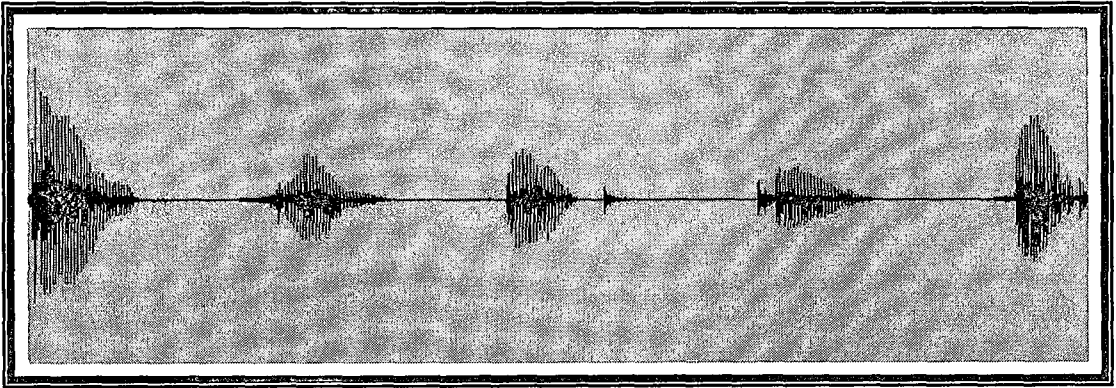
Şekil 4. 12 F4 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



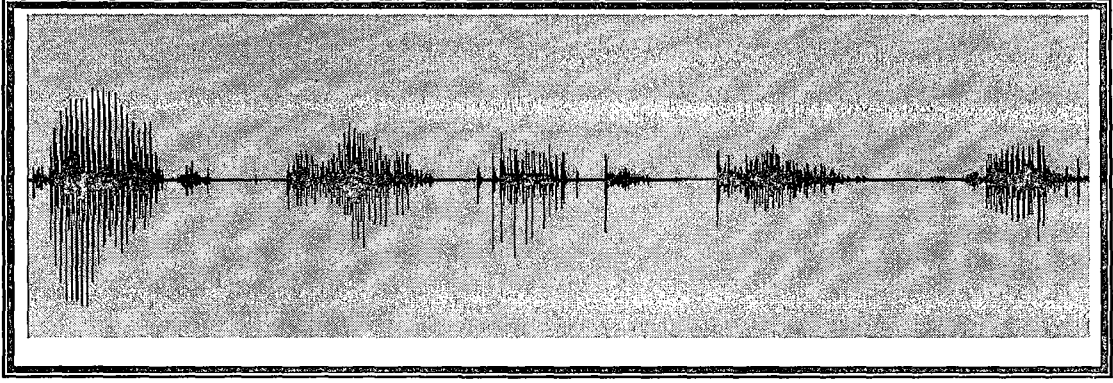
Şekil 4. 13 F5 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



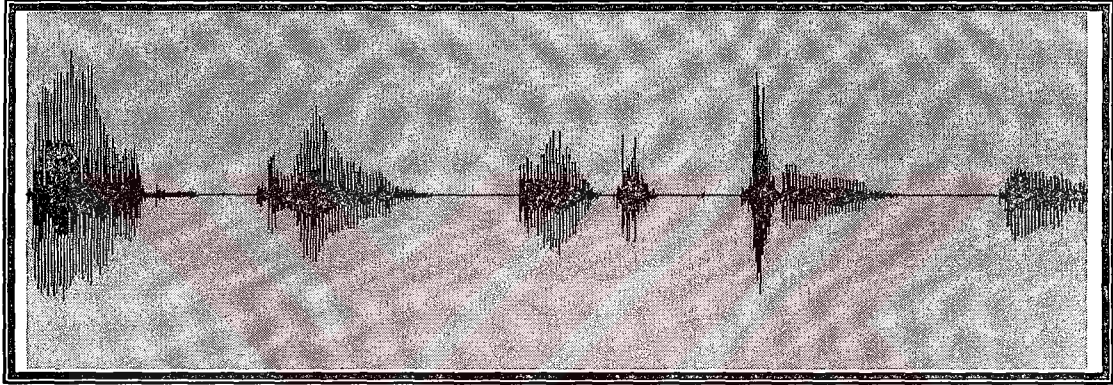
Şekil 4. 14 M1 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



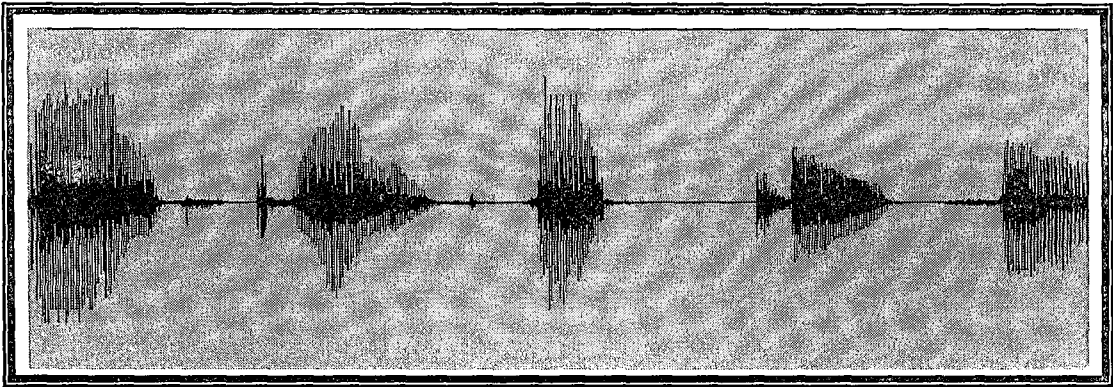
Şekil 4. 15 M2 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



Şekil 4. 16 M3 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



Şekil 4.17 M4 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.



Şekil 4. 18 M5 konuşmacısının 1. oturum ses dosyasına ait dalga şekli.

4.5.3. Konuşmacı veritabanının özellik çıkartım işlemleri

Veritabanı setindeki konuşmacı örnek seslerine uygulanan özellik çıkartım aşamaları aşağıdaki şekildedir.

1. Konuşma işaretleri, transfer fonksiyonu Eşitlik 4.28' deki gibi birinci dereceli sayısal süzgeçten geçirilmiştir. Süzgeç katsayısı 0.98 olarak seçilmiştir.
2. Konuşma işaretinin her 660'lık bölüt için Hamming pencereleme yöntemi kullanılmıştır. Ayrıca her 660'luk bölüt için uygulanan pencereleme işlemi, pencereler arası geçişte veri kaybını önlemek için; her 330'luk (genelde bölütün yarısı kadar) örnekte bir tekrarlanmıştır.
3. Pencerelemiş ses sinyalinin parametre takımı katsayılarını elde etmek için Tablo 2.1' deki gibi öngörü hatası en az olan iki yöntemden tez çalışmasında yararlanılmıştır. Bu yöntemlerden öngörü hatası en düşük olan GABT yönteminin uygulaması aşamasında 100 bireyden oluşan bir nüfus kullanılmıştır, çaprazlama olasılığı 0.8, mutasyon oranı 0.1 ve iterasyon değerleri 1000 ve 2000 için uygulama gerçekleştirilmiştir. Elde edilen özellik vektörleri sınıflandırma aşamasında kullanılmıştır. Ayrıca bir diğer özellik çıkartım yöntemi olarak Mel-frekans ölçekli kepsral katsayılar Bölüm 4.6.1.3' de açıklandığı şekilde elde edilmiştir. Mel ölçeği insan kulağının algılama karakteristiklerini eşleştirmek için yapılan bir ayarlamadır ve konuşma tanıma sistemlerinde performansı artırır (Krishnamoorthy, 1998). Bu özellik çıkartım yöntemlerinden Mel-frekans ölçekli kepsral katsayılar (mfcc) ve GABT yöntemi ile özellik vektörleri üretilerek SOM YSA kullanılarak sınıflandırılmış ve sonuçlar daha sonraki bölümde karşılaştırılmıştır.

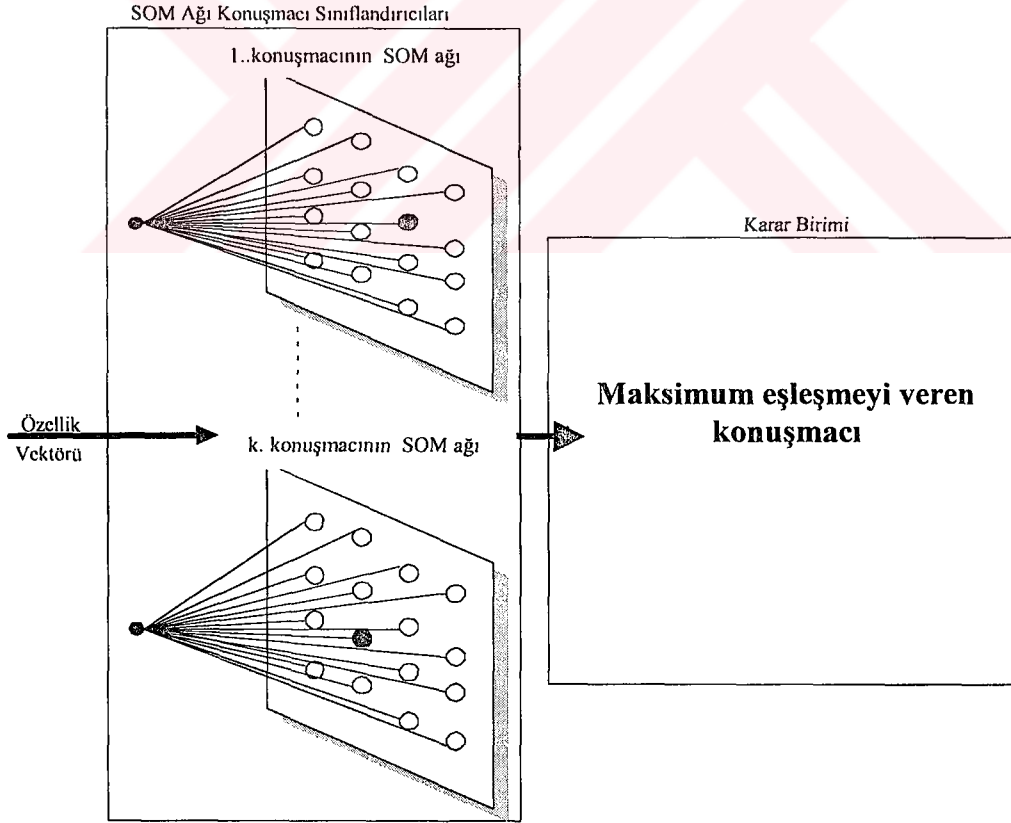
4.6. Amaçlanan Yöntem

SOM YSA sınıflandırıcısının eğitiminde, Bölüm 4.2.3'te açıklanan eğitimcisz öğrenme algoritmalarından Yarışmacı Öğrenme algoritması kullanılmıştır. SOM ağı eğitimcisz bir öğrenme algoritması kullandığı için ve ağı kendi çağrışımlarını kendisi

yaptığından eğiticili öğrenme algoritmalarına göre daha hızlı yakınsamaktadır (İnal 2002).

Yarışmacı öğrenme kuralına göre yarış kazanarak çıkış üreten işlem biriminin çıkışı "1" olacaktır. Yarış kazanan işlem birimlerinin diğer işlem birimlerinin çıkış üretmesini yasaklayabilme özelliğinden dolayı diğer işlem birimlerinin çıkışı "0" olacaktır. Yarış kaybeden işlem birimleri çıkış üretmez.

Bilindiği üzere SOM ağının çıkışları, ağdaki her işlem biriminin indislerini göstermektedir. Bir SOM ağı test edildiğinde sadece hangi çıkışın ilgili girişe göre etkin olduğu gözlemlenebilir. Buna göre bir konuşmacıya ait veri SOM ağına sunulduğunda çıkışta elde edilen indislerin hangi konuşmacıya ait olduğunu bulmak gerekir. Bu nedenle Şekil 4.19' daki yapıda karar birimi şeklinde bir blok kullanılmıştır.



Şekil 4. 19 Test aşamasında SOM ağının kullanıldığı yapı.

BÖLÜM 5. KONUŞMACI SAPTAMA UYGULAMALARI

Konuşmacı saptama uygulamalarının program kodları Matlab 5.2 programında yazılmıştır. Uygulamalar PIII-550 MHZ. işlemcili , 128 MB bellek ve 15 GB sabit disk kapasiteli bir kişisel bilgisayar üzerinde gerçekleştirilmiştir. Yazılan kodların çıktıları EK kısmında sunulmuştur.

Tez çalışmasının uygulama aşaması 3 bölüme ayrılabilir:

1. Konuşmacı veritabanı setinin hazırlanması. Konuşmacı seti iki kısımdan oluşmaktadır. Eğitim ve test için kullanılmak üzere iki konuşmacı seti oluşturulmuştur. Konuşmacı seti ile ilgili bilgiler Bölüm 4.5.2 Veritabanının oluşturulması kısmında açıklanmıştır.

2. Özellik çıkartım işlemleri yukarıda adı geçen konuşmacı veritabanı setin iki alt kısma ayrılan özellik çıkartım kısmından birincisi DÖK tabanlı mel-frekans ölçekli kepsral katsayıların kullanıldığı özellik çıkartım bölümüdür. Özellik çıkartım ile ilgili gerekli açıklamalar Bölüm 4.5.1. de açıklandığı şekilde uygulanmıştır. İkinci kısım ise tez çalışmasının amaçlarından biri olan konuşma işaretinin parametrik özelliklerini en iyi şekilde elde etme prensibi çerçevesinde düşünülüp Bölüm 3' te sunulan GABT yönteminin kullanılarak özellik çıkartım işleminin gerçekleştirildiği bölümdür.

3. Uygulama aşamasının son bölümünde ise farklı iki yaklaşımla elde edilmiş olan özellik vektörlerinin sınıflandırılmasıdır. Sınıflandırıcı olarak SOM YSA kullanılmıştır. SOM YSA sınıflandırıcısının eğitiminde, Bölüm 4.1'de açıklanan öğrenme metotlarından yarışmacı öğrenme algoritması kullanılmıştır.

Tez çalışmasında kullanılan konuşmacı veritabanı CSLU Speaker Recognition V 1.1 setinden elde edilmiştir. 5 bayan ve 5 erkek konuşmacı olmak üzere 10 konuşmacı

seçildi. Bu konuşmacılar çeşitli sayıları ikişer kez telaffuz ettiği, ayrı oturumlarda kayıt edilmiş ses dosyaları kullanılmıştır. Eğitim için 60, test için 60 olmak üzere toplam 120 farklı ses dosyası kullanılmıştır. Her konuşmacı telaffuzları daha önceden F1, F2, F3, F4, F5 (bayan konuşmacılar) ve M1, M2, M3, M4, M5 (erkek konuşmacılar) şeklinde etiketlenilerek konuşmacıların elde edilen özellik vektörleri kendi etiketleri ile adlandırılmış klasörlerde 'mat' dosyaları şeklinde saklanmıştır.

Yapılan çalışmalar sonucunda en iyi sonucu veren iki yöntem özellik çıkartım aşamasında kullanılmıştır. Bu yöntemler mel-frekans ölçekli katsayılar ve GABT yöntemidir. Bu çalışmada CSLU veritabanı kullanılarak bölüm 4.5.1' de anlatılan özellik çıkartım aşamaları aşağıdaki parametreler seçilerek uygulanmıştır:

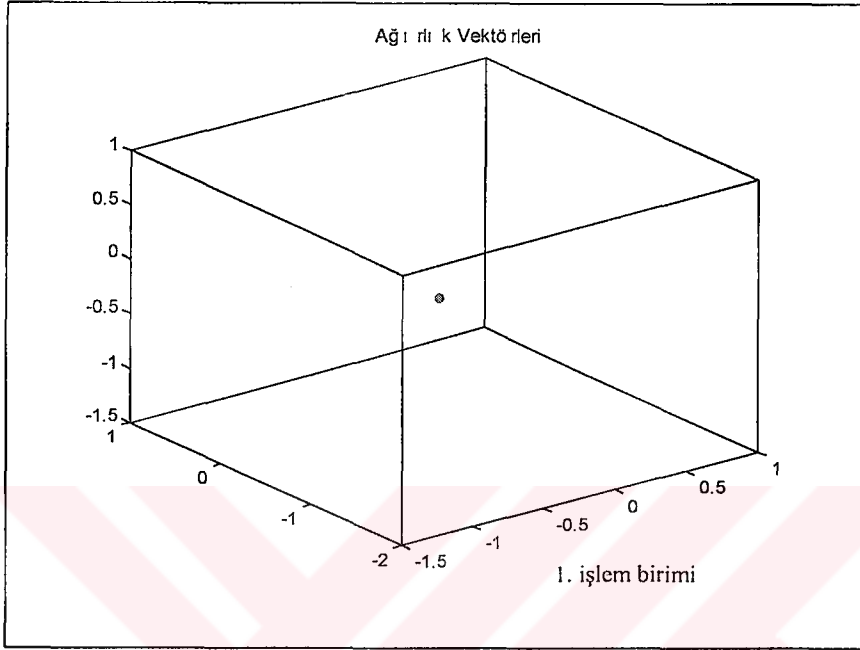
1 Konuşma işaretleri Eşitlik 4.28'deki gibi birinci dereceli sayısal süzgeçten geçirilmiştir.

2. Konuşma işaretinin her 660' lık bölütü için Hamming pencereleme yöntemi uygulanmış. Sinyali 660 'lık bölütlere ayrılma sebebi örnekleme frekansıdır. Kullanılan veritabanındaki ses dosyalarının örnekleme frekansı 8 kHz'tir, kullanacağımız katsayı (p) değeri 12 seçildiğinden, $8000/12$ değerinden yaklaşık olarak bölüt değeri 660 olarak bulunur. Her 330'luk pencerede bir, pencereler arası geçişte veri kaybını önlemek için , tekrarlama (overlap) yapılmıştır.

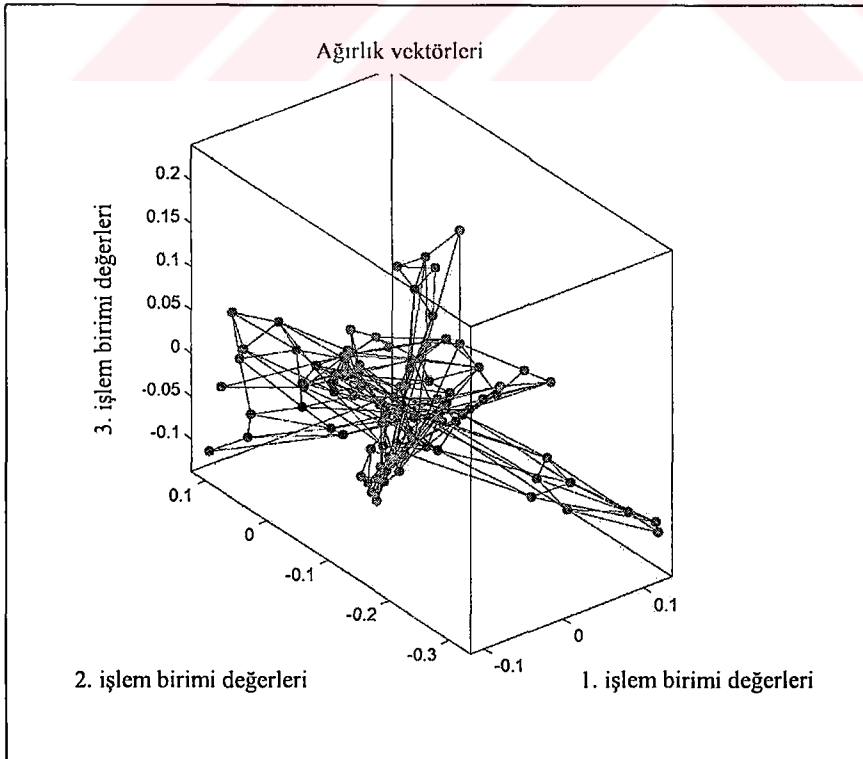
3. Bu çalışmada, bölütlere ayrılmış her pencere için üretilen özellik vektörü katsayısı 12 seçilmiştir. Her bölüt 12. dereceden katsayılar şeklinde ifade edilmiştir.

Elde edilen özellik vektörleri sınıflandırıcı yapısında kullanılan SOM ağlarının eğitim ve test vektörlerini oluşturmuştur. SOM ağlarının eğitiminde, öğrenme oranı η 0.9 olarak seçilmiştir.

Eğitime başlamadan önceki ilk üç işlem biriminin ağırlık vektörleri Şekil 5.1' de gösterilmiştir. Şekil 5.2' de eğitim sonrasında ağırlık vektörlerinin durumu gösterilmiştir.



Şekil 5.1. İlk üç işlem biriminin diğer ağırlıklarla bağlantıları



Şekil 5.2 Eğitim sonrasında ilgili işlem birimlerinin ağırlıkları

Bu çalışmada kullanılan diğer bir yöntem olan GABT, özellik çıkartım işlemi için Turbo Pascal' da yazılmış bir program kullanılmıştır. GABT yönteminde 12. dereceden özellik çıkartım katsayılarını elde etmek amacıyla kromozom uzunluğu 12, nüfus büyüklüğü 100 olarak belirlenmiştir. Yeni nesillerin oluşturulması sırasında gerçekleşecek olan çaprazlama işleminin olasılık değeri (pcross) 0.8 , mutasyon değeri (chmax) 0.1 seçilmiştir. Uygunluk fonksiyonu, sinyalin kendisinden önceki n adet örneğin doğrusal bir birleşiminden oluştuğu göz önüne alındığında, oluşacak kestirim hatasının en az olması gerektiği anlamına gelir. Bu durum örnek bir konuşma penceresine uygulanan GABT yöntemi sonucunda, kestirimi yapılan katsayıların kuşak boyunca değişimi, kuşak boyunca uygunluk değerinin değişimi Şekil 5.3' te gösterilmiştir. Şekil 5.3 (b) incelendiğinde amaç fonksiyonunun uygunluk değeri nesiller boyunca artmıştır. Şekil 5.3 (a)' da bir parametre katsayı değerinin değişimi gösterilmiştir.

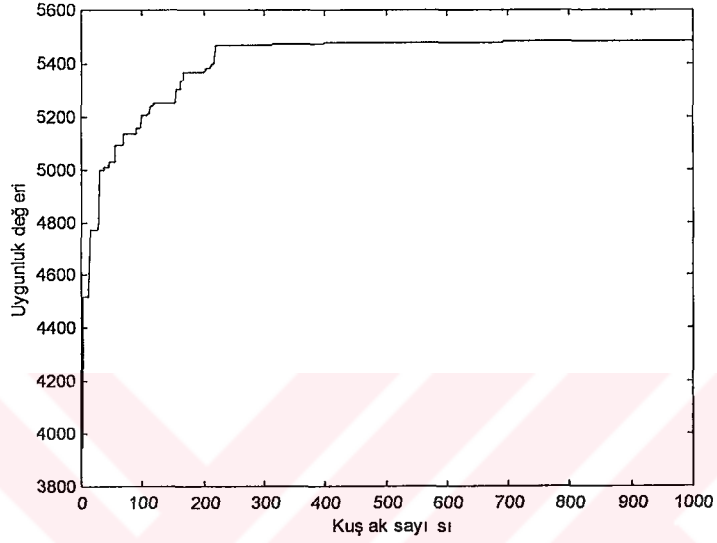
GABT' de gerçel kodlama kullanılarak 1000 ve 2000 nesil için uygulama gerçekleştirilmiştir. İşleme sokulacak işaretin pencere uzunluğu 660 olarak seçilmiştir. Ek bölümünde verilen Matlab programında yazılmış kod aracılığı ile elde edilen konuşma işaretinin her bir penceresi için 12 katsayılı bir dizi metin (text) dosyasına kaydedilmiştir. Program sonucu elde edilen özellik vektörleri SOM ağlarının eğitim ve test vektörleri için kullanılmıştır.

Metinden bağımsız kapalı set konuşmacı saptama uygulamasında sınıflandırıcı olarak kullanılan SOM YSA yapısında, kapalı set konuşmacı saptama uygulaması, 10 konuşmacı için gerçekleştirilmiştir.

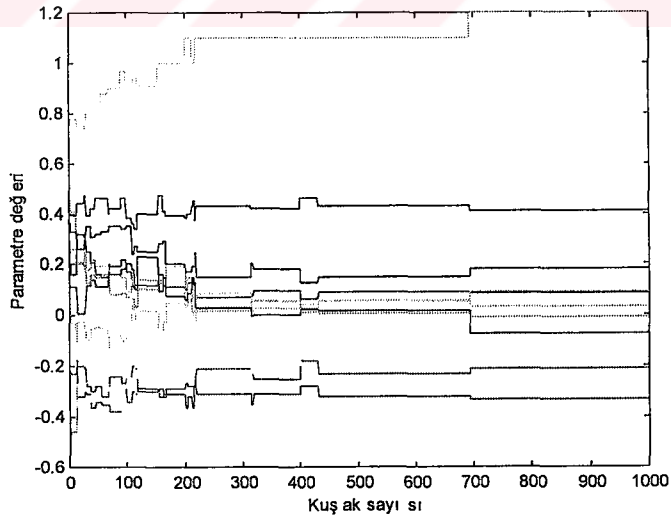
Her bir konuşmacı için ayrı ayrı SOM ağları oluşturulmuş ve eğitim sırasıyla 2000, 5000, 10000 ve 20000 iterasyonda iki boyutlu 10x10 işlem birimi ve iki boyutlu 20x20 işlem birimi ağ mimarileri kullanılmıştır. Burada iki boyutlu mimari, tek katmanda işlem birimlerinin konumunu ifade etmektedir.

Ayrıca eğitim aşamasında SOM ağlarının ayırt edici özelliğini arttırmak için her konuşmacının eğitimi verisi kendi adıyla etiketlenen tamsayı değeri (1, 2, 3, ...,

10) ile çarpılarak, test aşamasında da çarpılan bu tamsayı değerleri göz önünde bulundurulmuştur. Veritabanı içindeki tüm konuşmacılar için eğitim ve test özellik vektörleri için “i” çarpanı ve “i” çarpanı olmadan standart sapma ve varyansları hesaplanmıştır. Bu tamsayı değerinin SOM ağının giriş uzayının varyansını ve standart sapmasını değiştirmedeği Tablo 5.1– 5.2’ de görüldüğü gibi gözlemlenmiştir.



(a)



(b)

Şekil 5.3 Uygunluk değeri ve Parametre katsayı değeri

(a) Nesil boyunca uygunluk değeri değişimi

(b) Nesil boyunca parametre değeri değişimi

Tablo 5.1 Eğitim ve test vektörleri için “i” çarpanı ve “i” çarpanı olmadan standart sapma değerlerinin karşılaştırması

Standart Sapma			
Çarpansız		Çarpanlı	
Eğitim	Test	Eğitim	Test
0.2240	0.2197	0.2240	0.2197
0.2474	0.2446	0.4948	0.4893
0.2358	0.2351	0.7075	0.7054
0.2481	0.2470	0.9925	0.9880
0.2514	0.2508	1.2571	1.2542
0.2507	0.2505	1.5043	1.5030
0.2372	0.2365	1.6605	1.6557
0.2592	0.2558	2.0740	2.0466
0.2473	0.2441	2.2258	2.1967
0.2503	0.2505	2.5033	2.5046

Tablo 5.2 Eğitim ve test vektörleri için “i” çarpanı ve “i” çarpanı olmadan varyans değerlerinin karşılaştırması

Varyans			
Çarpansız		Çarpanlı	
Eğitim	Test	Eğitim	Test
0.0502	0.0483	0.0502	0.0483
0.0612	0.0598	0.2448	0.2394
0.0556	0.0553	0.5006	0.4975
0.0616	0.0610	0.9850	0.9761
0.0632	0.0629	1.5803	1.5731
0.0629	0.0627	2.2628	2.2589
0.0563	0.0559	2.7573	2.7414
0.0672	0.0654	4.3014	4.1886
0.0612	0.0596	4.9541	4.8257
0.0627	0.0627	6.2666	6.2729

Tablo 5.1 ve Tablo 5.2’deki sonuçlara bakıldığında, “i” katsayı değeri ile çarpılmış olan özellik vektörlerinin standart sapma ve varyans değerlerinin aynı sınır değerleri içine girmediği ve bunun sonucunda ayırt ediciliğin daha da arttığı görülmektedir. “i” katsayı çarpanı eğitim vektörlerine uygulandığı şekilde test vektörlerine de uygulanmıştır. Sonuç olarak “i” çarpanı ile eğitilen bir ağın kendisine uygulanan test vektörlerine göre saptama oranının daha yüksek olduğu yapılan uygulamalarda görülmüştür.

5.1 SOM Sınıflandırıcısının Kullanıldığı Konuşmacı Saptama Uygulamaları

Bu çalışmada, CSLU veritabanını kullanarak metinden bağımsız kapalı-set konuşmacı saptama uygulamaları yapılmıştır. Uygulama 5 erkek, 5 bayan olmak üzere toplam 10 konuşmacı seçilmiştir. Her konuşmacı için ayrı ayrı SOM YSA’ları oluşturulmuş, işlem biriminin etkisini incelemek için ağ mimarisi iki boyutlu 10x10

ve 20x20 şeklinde seçilmiştir. Ağ eğitiminde iterasyonun etkisini araştırmak amacıyla da uygulamalar 2000, 5000, 10000 ve 20000 iterasyon için gerçekleştirilmiştir.

İlk uygulamada Mel-frekans ölçekli kepsral katsayılardan oluşan özellik vektörleri sırası ile 2000, 5000, 10000 ve 20000 iterasyon değerleri için eğitilmiştir. Bu çalışmaya ait uygulama sonuçları Tablo 5.3- 5.6' de görülmektedir.

Tablolar incelendiğinde , “i” katsayı çarpanı ile elde edilen ortalama verim, çarpansız katsayı değeriyle elde edilene ortalama verime göre daha fazla olduğu görülmüştür. Örneğin Tablo 5.3 incelendiğinde, “i” katsayı çarpanı ile elde edilen ortalama verim, çarpansız katsayı değeriyle elde edilene ortalama verime göre yaklaşık %12' lik bir iyileşme sağlamıştır. %12' lik bir iyileşme konuşmacı tanıma işlemi için oldukça önemli bir değerdir.

Bununla beraber, işlem biriminin etkisi araştırıldığında, işlem biriminin yapılan çalışmada verimi yaklaşık %1-2 seviyelerinde arttırmıştır. Sonuç olarak işlem birimi sayısı verimi arttırmaya yönelik bir parametre olduğu söylenebilir. Fakat bu işlemin ağır hesaplama yükünü arttırdığı düşünülürse, verimdeki %1-2' lik artış önemsenecek kadar yüksek düzeyde değildir. Yapılan farklı bir çalışmada işlem birimi sayısının sistem verimi açısından etkin bir parametre olmadığı görülmüştür.

Ayrıca iterasyon değerleri açısından incelendiğinde, iterasyon sayısı arttığı sürece verim artmaktadır. Bununla birlikte, iterasyon sayısının artması ağır daha uzun süre eğitilmesi anlamına gelir. Sonuç olarak daha fazla eğitilen ağır daha iyi sonuç verdiği görülmüştür.

Tablo 5.3 2000 iterasyon için Mel-frekans ölçekli katsayı değerleri uygulama sonuçları

Mel-ölçekli katsayı değerleri için uygulama sonuçları (%)				
2000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	10x10	10x10	10x10
F1	85,678	83,950	84,390	85,448
F2	87,169	87,416	90,767	90,620
F3	82,912	82,217	91,698	92,588
F4	84,162	79,010	95,986	94,472
F5	81,694	76,565	95,619	95,213
M1	83,339	83,477	96,223	96,003
M2	83,684	82,116	97,585	97,444
M3	83,356	82,115	97,180	96,546
M4	82,535	79,559	96,295	95,679
M5	82,916	80,399	98,538	98,335
Ortalama	82,713	81,682	94,428	94,235

Tablo 5.4 5000 iterasyon için Mel-frekans ölçekli katsayı değerleri uygulama sonuçları

Mel-ölçekli katsayı değerleri için uygulama sonuçları (%)				
5000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,815	83,883	85,318	85,049
F2	86,624	87,585	87,823	94,110
F3	82,657	81,906	91,950	94,546
F4	84,375	80,466	93,540	95,030
F5	82,929	77,899	94,636	95,707
M1	83,966	81,375	97,609	97,713
M2	83,611	81,825	98,384	98,006
M3	83,207	82,009	96,662	97,394
M4	82,838	79,783	96,434	96,944
M5	83,718	80,500	97,641	96,112
Ortalama	83,874	81,723	94,000	95,061

Tablo 5.5 10000 iterasyon için Mel-frekans ölçekli katsayı değerleri için uygulama sonuçları

Mel-ölçekli katsayı değerleri için uygulama sonuçları (%)				
10000 iterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	85,407	85,684	86,671	86,518
F2	87,400	86,884	90,036	94,011
F3	82,883	82,405	93,533	96,561
F4	83,583	80,620	94,406	94,171
F5	83,055	77,982	94,535	95,436
M1	84,606	81,762	96,238	97,248
M2	83,941	82,629	97,310	96,915
M3	83,408	81,151	97,516	97,740
M4	83,338	80,529	96,719	96,198
M5	83,148	80,579	99,514	96,527
Ortalama	84,077	82,022	94,648	95,132

Tablo 5.6 20000 iterasyon için Mel-frekans ölçekli katsayı değerleri için uygulama sonuçları

Mel-ölçekli katsayı değerleri için uygulama sonuçları (%)				
20000 iterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,853	85,864	88,296	85,978
F2	86,953	87,483	89,519	93,581
F3	83,306	84,466	91,714	95,920
F4	84,688	81,534	94,533	95,527
F5	82,972	79,376	93,675	95,062
M1	83,193	84,183	95,415	96,695
M2	84,921	82,791	98,620	97,810
M3	84,037	83,088	96,217	96,718
M4	84,216	81,164	96,707	97,256
M5	83,039	83,107	97,707	99,537
Ortalama	84,218	83,306	94,240	95,408

Tüm konuşmacılar için ortalama verimler GABT yöntemi ile karşılaştırılabilmesi amacı ile Tablo 5.15’ de gösterilmiştir.

Bu çalışma da yapılan bir diğer konuşmacı saptama uygulaması, GABT yöntemi ile elde edilen özellik vektörlerinin kullanıldığı uygulamadır. GABT yönteminde özellik çıkartım vektörlerinin incelenmesi 1000 ve 2000 nesil için yapılmıştır. 1000 nesil için GABT uygulaması ile ilgili sonuçlar Tablo 5.7 –5.10 gösterilmiştir.

Tablo 5.7 2000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(1000 nesil) (%)				
2000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,673	81,593	83,631	81,720
F2	85,772	87,902	97,889	91,782
F3	82,295	79,857	97,735	94,908
F4	84,641	83,201	96,309	91,976
F5	81,416	75,473	97,304	96,223
M1	83,312	81,482	99,094	97,750
M2	83,787	79,454	99,210	97,372
M3	84,615	80,819	98,785	98,985
M4	82,682	80,027	99,741	99,224
M5	82,898	81,003	99,177	99,226
Ortalama	82,345	81,081	96,887	94,917

Tablo 5.8 5000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları (1000 nesil) (%)				
5000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,328	82,354	87,854	85,932
F2	87,165	86,456	97,090	96,458
F3	82,218	79,986	98,457	97,115
F4	84,145	83,530	97,495	96,970
F5	80,719	77,210	97,979	97,145
M1	83,827	80,525	98,534	98,704
M2	83,295	78,903	98,614	97,798
M3	85,224	82,479	99,153	98,446
M4	83,327	79,548	99,483	99,254
M5	84,052	80,507	99,506	98,854
Ortalama	83,830	81,150	97,416	96,668

Tablo 5.9 10000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(1000 nesil) (%)				
10000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	85,480	83,516	86,324	87,026
F2	87,020	87,302	96,859	96,333
F3	83,955	81,604	98,157	97,658
F4	84,578	83,429	97,353	96,196
F5	82,155	76,749	97,673	97,146
M1	83,293	81,934	98,859	97,977
M2	82,584	80,863	99,029	98,663
M3	83,802	84,060	98,292	98,070
M4	83,144	79,798	99,513	99,224
M5	84,767	81,634	99,484	99,235
Ortalama	84,078	82,089	97,154	96,753

Tablo 5.10 20000 iterasyon 1000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(1000 nesil) (%)				
20000 iterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,557	84,931	87,954	88,532
F2	86,896	87,861	97,523	97,107
F3	83,611	82,459	98,348	97,543
F4	84,004	82,921	97,030	96,534
F5	81,860	78,754	97,605	96,622
M1	83,962	82,299	98,974	98,506
M2	83,451	81,998	98,902	98,581
M3	85,695	83,482	99,334	98,048
M4	84,185	81,156	99,741	99,741
M5	84,553	83,666	99,514	99,283
Ortalama	84,278	82,953	97,492	97,050

Tablo 5.7- 5.10 incelendiğinde, 1000 nesil için GABT ile “i” çarpansız elde edilen katsayıların kullanıldığı uygulama verimi ile aynı sınıftaki Mel-frekans ölçekli değerlerin verimleri oldukça yakın çıkmıştır. GABT uygulamasında “i” katsayı çarpanı kullanılarak yapılan çalışmada verimi, aynı karşılaştırma için daha yüksek bir saptama verimi göstermiştir.

Tablo 5.11– 5.14 ‘ te 2000 nesil için GABT uygulama sonuçları gösterilmiştir. Ardından her iki yöntem için ortalama verimlerin genel ortalamaları Tablo 5.16 ve Tablo 5.17’ de gösterilmiştir.

Tablo 5.11 2000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(2000 nesil) (%)				
2000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	83,679	83,113	87,564	86,490
F2	87,252	86,411	95,975	95,540
F3	82,838	80,477	97,645	95,889
F4	84,468	81,613	97,189	95,606
F5	82,101	75,274	97,535	96,827
M1	84,489	80,310	99,030	98,434
M2	83,135	79,998	99,185	98,169
M3	84,232	82,152	98,358	98,046
M4	83,792	78,867	99,741	99,282
M5	84,016	80,795	99,256	99,615
Ortalama	84,000	80,901	97,148	96,390

Tablo 5.12 5000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(2000 nesil) (%)				
5000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,666	81,857	88,622	88,190
F2	86,640	88,127	95,957	95,464
F3	82,342	80,169	97,738	96,468
F4	85,028	85,104	97,276	97,191
F5	80,579	76,289	97,406	97,175
M1	84,380	82,857	98,750	98,156
M2	83,627	80,165	99,059	98,129
M3	85,627	82,110	98,229	98,014
M4	83,004	81,317	99,224	99,741
M5	85,085	80,762	99,129	98,355
Ortalama	84,098	81,876	97,139	96,688

Tablo 5.13 10000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(2000 nesil) (%)				
10000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	84,176	82,755	89,820	88,260
F2	88,065	87,255	95,972	95,254
F3	82,772	81,164	97,870	96,283
F4	84,002	82,592	98,163	96,084
F5	81,663	77,014	97,460	96,552
M1	84,992	83,003	98,633	97,576
M2	82,768	79,388	99,210	97,943
M3	84,677	83,919	98,358	98,429
M4	83,883	81,530	99,282	99,483
M5	84,569	82,887	99,278	99,126
Ortalama	84,157	82,151	97,405	96,499

Tablo 5.14 20000 iterasyon 2000 nesil için GABT katsayı değerleri için uygulama sonuçları

GABT katsayı değerleri için uygulama sonuçları(2000 nesil) (%)				
20000 İterasyon	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
Konuşmacı etiketi	10x10	20x20	10x10	20x20
F1	85,075	83,868	89,877	88,284
F2	87,815	87,376	96,661	96,249
F3	84,073	83,391	97,300	97,310
F4	85,205	84,600	97,132	95,747
F5	82,066	77,906	97,577	96,253
M1	83,451	82,366	99,117	97,766
M2	83,222	81,586	99,002	98,651
M3	85,139	83,782	99,044	98,559
M4	82,958	82,442	99,741	99,741
M5	84,181	83,079	99,484	98,540
Ortalama	84,318	83,040	97,493	96,710

2000 nesil için uygulama sonuçları incelendiğinde her iki durum içinde, “i” çarpanlı veya “ı” çarpansız olarak, saptama veriminin daha yüksek olduğu görülmektedir. Elde edilen sonuçlara göre GABT yöntemi ile elde edilen katsayı değerlerinin Mel-frekans ölçekli katsayılara göre daha iyi sonuç vermiştir.

Tablo 5.15 – Tablo 5.17’ de, her bir iterasyon değeri için ortalama verimler gösterilmiştir. Ortalama verimlerin genel ortalamaları alındığında GABT yöntemi ile elde edilen özellik vektörlerinin kullanıldığı uygulamaların daha iyi sonuç verdiği görülmüştür.

Tablo 5.15 Mel-frekans ölçekli katsayı değerleri için uygulama sonuçlarının ortalamaları

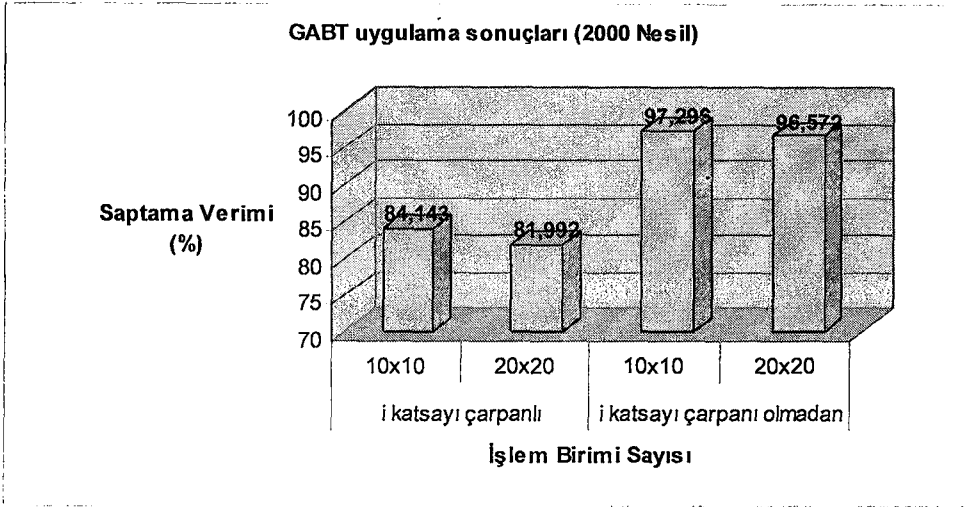
Mel- ölçekli katsayı değerleri için uygulama sonuçları ortalamaları (%)				
İterasyon sayısı	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
	10x10	20x20	10x10	20x20
2000	82,713	81,683	94,428	94,235
5000	83,874	81,723	94,000	95,061
10000	84,077	82,022	94,648	95,132
20000	84,218	83,306	94,240	95,408
Ortalama	83,720	82,183	94,329	94,960

Tablo 5.16 1000 nesil için GABT katsayı değerleri için uygulama sonuçlarının ortalamaları

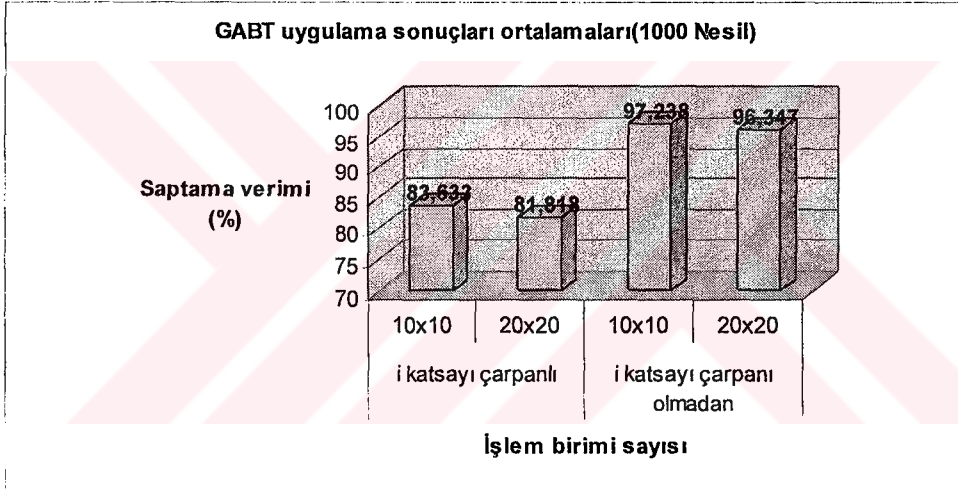
GABT katsayı değerleri için uygulama sonuçları(1000 nesil) (%)				
İterasyon sayısı	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
	10x10	20x20	10x10	20x20
2000	82,345	81,081	96,887	94,917
5000	83,830	81,150	97,416	96,668
10000	84,078	82,089	97,154	96,753
20000	84,278	82,953	97,492	97,050
Ortalama	83,633	81,818	97,238	96,347

Tablo 5.17 2000 nesil için GABT katsayı değerleri için uygulama sonuçlarının ortalamaları

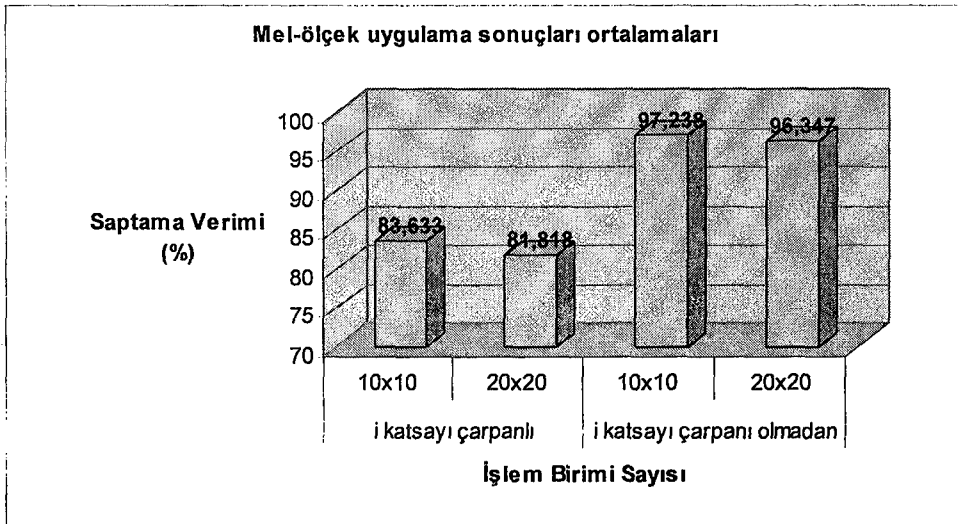
GABT katsayı değerleri için uygulama sonuçları(2000 nesil) (%)				
İterasyon sayısı	katsayı olmadan		katsayılı	
	İşlem Birimi Sayısı		İşlem Birimi Sayısı	
	10x10	20x20	10x10	20x20
2000	84,000	80,901	97,148	96,390
5000	84,098	81,876	97,139	96,688
10000	84,157	82,151	97,405	96,499
20000	84,318	83,040	97,493	96,710
Ortalama	84,143	81,992	97,296	96,572



Şekil 5.4 2000 nesil için GABT uygulama sonuçları ortalama değer grafiği..



Şekil 5.5 1000 nesil için GABT uygulama sonuçları ortalama değer grafiği..



Şekil 5.6 Mel-ölçek uygulama sonuçları ortalama değer grafiği.

BÖLÜM 6. SONUÇ ve ÖNERİLER

Yapılan çalışmalarda ilgili sonuçlar ve yorumlar aşağıda maddeler halinde sıralanmıştır.

1- Tez çalışmasındaki konuşma saptama işleminin ilk aşaması olan özellik vektörlerinin elde edilmesi için iki farklı yöntemden yararlanılmıştır. Bunlar GABT ve mel-ölçekli kepsral katsayılardır. Yapılan deneysel çalışmaların sonucunda GABT ve Mel-frekans ölçekli katsayı değerlerinin öngörü hatalarının daha az olduğu görülmüştür (Bkz. Tablo 2.1). Bu nedenle özellik vektörlerinin çıkartım aşamasında GABT ve Mel-frekans ölçekli kepsral katsayılar kullanılmıştır.

2- Oluşturulan konuşmacı veritabanı için CSLU veritabanı içinden rasgele 5 bayan 5 erkek konuşmacı seçilmiştir. Konuşmacıların farklı oturumlarda söyledikleri rakam telaffuzları SOM ağının eğitimin ve test aşamalarında kullanılmıştır. Konuşmacılar belir bir telaffuz ile sınırlandırılmamışlardır. Bu nedenle tanıma uygulaması “Metinden Bağımsız Tanıma” dır.

3- Sınıflandırma aşamasında eğitimcisz öğrenme ağlarından biri olan kendi kendini organize eden ağ modeli, SOM YSA kullanılmıştır. Bölüm 2.’de söz edilen özellik vektörlerinin çıkartım yöntemlerinden elde edilen değerler SOM YSA modeli için giriş oluşturur. SOM YSA modeli girişinde, her konuşmacının test ve eğitim vektörleri değeri kendi etiket numarası ile çarpılarak ayırt edicilik artırılmıştır. Özellik vektörlerinin standart sapma ve varyans değerlerinin aynı sınır değeri içinde olmadığı bunun sonucunda ayırt ediciliğin daha da arttığı görülmüştür (Bkz. Tablo 5.1 ve Tablo 5.2).

SOM YSA modelinin kullanıldığı metinden bağımsız kapalı set konuşmacı saptama uygulaması gerçekleştirilmiştir. Uygulama mel-frekans ölçekli katsayı değerleri ve GABT katsayı değerleri kullanılarak iki kısımda gerçekleştirilmiştir. Her iki uygulama için 10x10 ve 20x20 ağ mimarisine sahip , 2000, 5000, 10000 ve 20000

iterasyon deęerleri için eęitilmiş olan SOM YSA yapısı kullanılmıştır. Mel-frekans ölçekli katsayılar için yapılan uygulama sonucu elde edilen saptama verimlerini genel ortalama deęerleri Tablo 5.15' de gösterilmiştir. Elde edilen sonuçlar incelendiğinde, katsayı çarpanlı 10x10 ve 20x20 aę mimarisine sahip, 2000, 5000, 10000 ve 20000 iterasyon deęerleri için eęitilmiş aęın saptama verimlerinin genel ortalaması %94.96 (Bkz. Tablo 5.15) olarak bulunmuştur. Bununla beraber GABT uygulaması 1000 ve 2000 nesil için ayrı ayrı gerçekleştirip uygulamalar (Bkz. Tablo 5.7 - 5.14) sunulmuştur. Tablolar incelendiğinde 1000 nesil ve 2000 nesil için elde edilen deęerlerin birbirlerine oldukça yakın olduęu görülmüştür. Bu sebepten GABT uygulaması için 1000 nesil deęerinin yeterli olduęu söylenebilir.

İterasyon deęerlerine göre tablolar incelenirse, daha yüksek iterasyon deęerleri için daha yüksek saptama verimi sağlanıldığı görülmüştür. Aę eęitimi için kullanılan bu deęerlerin yüksek olması daha fazla sürenin harcanması gerektięi anlamına geliyor. Yapılan uygulama çevrim dıőı yapıldığı için zamandan baęımsızdır, fakat çevrim içi bir uygulama için zaman faktörü de devreye gireceğinden, iterasyon deęeri de sonucu etkileyecek daha önemli bir parametre olacaktır.

Tablolar incelendiğinde katsayı çarpanı ile elde edilen ortalama verim, çarpansız katsayı deęeriyle elde edilen ortalama verime göre %12'lik bir iyileşme sağlamıştır ki bu oran konuşma tanıma uygulamaları için yüksek bir deęerdir (Bkz. Tablo 5.3). Buna karşın işlem birimi sayısının %1-2' lik verim artışı sağladığı görülmüştür. Fakat işlem birimi sayısının aęın hesaplama yükünü arttırdığı düşünülürse, %1-2'lik bu artış oranı dikkate alınacak kadar yüksek deęildir.

Sonuç olarak GABT tabanlı yaklaşımının mel-frekans ölçekli kepsral katsayılarla oranla daha iyi sonuç verdięi görülmüştür. Bu tez çalışmasında yapılan çalışmalar doğrultusunda, aőağıda sunulmuş çalışmaların yapısı önerilir.

- GABT uygulaması için, GA parametrelerinden nüfus, mutasyon, çaprazlama gibi parametrelerini deęiştirerek daha iyi sonuçlar elde edilebilir.
- Sınıflandırma aşamasında tek bir SOM YSA modeli tasarlanıp, saptama verimi çalışmaları yapılabilir.

KAYNAKLAR

1. BOUTSINAS, B., VRAHATIS, M. N., 2001. Artificial Intelligence 132 , 1-38. ELSEVIER.
2. CAMPBELL, J., REYNOLDS, F., D., A., 1999. Corpora For The Evaluation of Speaker Recognition Systems. ICASSP 15-19 March.
3. CHANG, T., 1997. Genetic Algorithms in Battlefield Communications. PC AI, Sept./Oct., 28-30
4. COLE, R., NOEL, M., and NOEL, V., 1998. The CSLU Speaker Recognition Corpus In Proc. Of the Internatinal Conf On Spoken Language Processing (ICSLP), Sydney, Australia.
5. DELLER, J.R., PROAKIS, J.G., and HANSEN, J.H.L., 1993. Discrete Time Processing of Speech Signals, McMillan Pub. Co.
6. DE JONG, K.A., 1993. Genetic Algorithms Are Not Function Optimizers. Foundations of Genetic Algorithms. 2, Morgan Kaufmann Publishers, San Mateo, California, 5-17.
7. ESHELMAN, L.J., SCHAFFER, J.D., 1993. Real-Coded Genetic Algorithms and Interval-Shcemata. Foundations of Genetic Algorithms.2, Morgan Kaufmann Publishers, San Mateo Calirifornia, 187-202.
8. FARELL, K. R., MAMMONE, R., J., and ASSALEH, K.,T., Jan. 1994. Speaker Recognition Using Neural Networks and Conventional Classifiers, IEEE Trans. On Speech and Audio Processing, Vol.2, No.1, part II.
9. FORREST, S., MITCHELL, M., 1993. Relative Building-Block Fitness and the Building-Block Hypothesis. Foundations of Genetic Algorithms.2, Morgan Kaufmann Publishers, San Mateo Calirifornia, 109-126
10. FURUI, S., 1995. Speaker Recognition, Tokyo Institute of Technology, Department of Computer Science, NATO ASI Series from Statistics to Neural Networks Vol:136.
11. GOLDBERG, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Reading. MA: Addison-Wesley Publishing Company.
12. HAYKIN, S., 1999. Neural Networks A Comprehensive Foundation. by Prentice-Hall, Inc.

13. İNAL, M., BÜTÜN, E., ERKAN, K., YILDIRIM, M., ÇEKEN, C., 2000. Comparison of Linear Predictive Analysis Methods for ANN-Based Speaker Identification. NEUREL-2000, 5th Seminar on Neural Network Application in Electrical Engineering, Faculty of Elec. Eng. Univ. of Belgrade, YU,109-112.
14. İNAL, M., FATİHOĞLU, Y.S., 2002. Self-Organizing Map and Associative Memory Model,Hybrid Classifier For Speaker Recognition. NEUREL-2002, Seminar on Neural Network Application in Electrical Engineering, Faculty of Elec. Eng. Univ. of Belgrade, YU.
15. İNAL, M., 1996. İTÜ Triga Mark-II Reaktörünün Yapay Sinir Ağıyla Kontrolü. Yüksek Lisans Tezi, KOU, Fen Bilimleri Ens., Elektronik-Bilgisayar Eğt. A.B.D., sayfa 60-63.
16. JONES, T., 1995. Crossover, Macromutaion, and Population-Based Search. Foundations of Genetic Algorithms. 3, Morgan Kaufmann Publishers, San Mateo, California, 73-80.
17. KAYRAN, A.H., Ekim 1992, Rasgele İşaretler ve İşaret İşlemedeki Uygulamaları, İ.T.Ü., Elektrik-Elektronik Fakültesi, Dijital İşaret İşleme Lisansüstü Ders Notları.
18. KRISHNAMOORTY, M., 1998. Speaker Identification using LPC features, CAIP Center (Center for Advanced Information Processing) Rutgers University.
19. KRISTISTINSSON, and DUMONT, 1992. System Identification and Control Using Genetic Algorithms. IEEE Transactions on Syst. Man. And Cyber., Vol.22, No:5, 1033-1046
20. LOUIS, S.J. RAWLINS, G.J.E. 1993. Syntatic Analysis of Convergence in Genetic Algorithms. Foundations of Genetic Algorithms. 2, Morgan Kaufmann Publishers, San Mateo, California, 141-151.
21. MAN, K.F., TANG, K.S., KWONG, S., 1996. Genetic Algorithms: Concepts and Applications. IEEE Transactions On Industrial Electronics. Vol.43 No:5,519-533
22. RABINER, L. and JUANG, B.H., 1993. Fundamentals of Speech Recognition, Prentice Hall Signal Processing Series.
23. REEVES, C.R., 1993. Using Genetic Algorithms with Small Populations. Foundations of Genetic Algorithms. 2, Morgan Kaufmann Publishers, San Mateo, California, 141-151.
24. ROSCA, J.P., BALLARD, D.H., 1995. Causality in Genetic Programming. Foundations of Genetic Algorithms.3, Morgan Kaufmann Publishers, San Mateo Calirfornia, 256-263.

25. SAWARAGI, T., UMEMURA, J., KATAI, O., IWAI, S., 1996. Fusing Multiple Data and Knowledge Sources for Signal Understanding by Genetic Algorithm. IEEE Transactions on Industrial Electronics, Vol. 43, No.3, 411-420
26. SPEARS, W.M., ANAND, V., 1991. A Study of Crossover Operators in Genetic Programming. Proceedings of the International Symposium on Methodologies for Intelligent Systems, Methodologies for Intelligent Systems, eds. RAS Z.W. and ZEMANKOVA M., Berlin: Springer-Verlag, 542, 409-418.
27. WRIGHT, A.H., 1991. Genetic Algorithms for Real Parameter Optimization. Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo California, 205-218



EK: PROGRAM LİSTELERİ

Mel-frekans ölçekli kepsral katsayılar ve GABT yöntemiyle elde edilmiş test ve eğitim vektörlerinin elde edilmesi için kullanılan matlab program kodları aşağıda verilmiştir. Bununla beraber metinden bağımsız kapalı-set konuşmacı saptama uygulaması için kullanılan SOM YSA sınıflandırıcı yapısının eğitim ve test aşamasında kullanılan program kodları da sunulmuştur.

%Özellik çıkartım vektörlerinin Benzetilmiş Tavlama ve Genetik Algoritmalar %ile elde edilmesi için kullanılan timit_uret.m dosyası

```
clear all; clc;
for i=1:5,
train=timit_egtm(['F' int2str(i)]);
dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\egtm\' ['F' int2str(i)] '\train.mat...
train']
    eval(['save ' dosyapath])
    test=timit_test(['F' int2str(i)]);
    dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\test\' ['F' int2str(i)] '\test.mat...
test']
eval(['save ' dosyapath])
end
for i=1:5,
train=timit_egtm(['M' int2str(i)]);
dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\egtm\' ['M' int2str(i)] '\train.mat...
train']
eval(['save ' dosyapath]) test=timit_test(['M' int2str(i)]);
dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\test\' ['M' int2str(i)] '\test.mat...
test']
eval(['save ' dosyapath])
end
```

%Test vektörleri için kullanılan timit_test.m fonksiyon dosyası

```
function vekt=timit_test(ad)
AA=[]; for k=1:6, ck=['s',int2str(k)];
dosyad=[ck '.wav'] dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\test\' ad '\'];
oku=[dosyapath dosyad]; [S,Fs,Format]=readwav(oku);
A=mfcc_gen(S,Fs,format); AA=[AA;A]; end vekt=AA;
```

%Eğitim vektörleri için kullanılan timit_test.m fonksiyon dosyası

```
function vekt=timit_egtm(ad)
AA=[]; for k=1:6, ck=['s',int2str(k)]
dosyad=[ck '.wav'] dosyapath=['C:\Matlab\work\yavuz_tez\ses\sayilar\egtm\' ad '\'];
oku=[dosyapath dosyad]; [S,Fs,Format]=readwav(oku);
A=mfcc_gen(S,Fs,Format);
AA=[AA;A]; end vekt=AA;
```

%GABT yöntemiyle özellik vektörlerinin elde edilmesi için kullanılan Turbo Pascal programını çağıran fonksiyon

```
function ver=mfcc_gen(S,Fs,Format)
NN=2;F=0:0.1:1;MM=linspace(-0.9,0.9,length(F));
a=fir2(NN,F,MM);
s=S(2:length(S))-a(1)*S(1:length(S)-1);
%*****LPC Analysis*****
N=660;%pencere uzunluğu
M=round(N/2);%yaklaşık yarıısı
ss=enframe(s,hamming(N),M);
nfil=floor(3*log(Fs));
p=12;%filtre derecesi
ii=1;w=[];
while(ii<=size(ss,1))
d=ss(ii,:);
save c:\bp\bin\al.dat d -ascii
dos('c:\bp\bin\gen.exe');
load c:\bp\bin\Ppar.dat;
w=[w;Ppar(size(Ppar,1),2:11)];
ii=ii+1
end
ver=w;
```

%GABT ile elde edilen özellik vektörlerinin SOM ağı eğitiminin yapıldığı timit_gen.m matlab programı dosyası

```
clear all;
clc;
fm={'F1' 'F2' 'F3' 'F4' 'F5' 'M1' 'M2' 'M3' 'M4' 'M5' };
N=10;
for i=1:10,
clear net;
ad=strvcat(fm(i));
disp(ad);
eval(['load ' 'C:\Matlab\work\yavuz_tez\ses\sayilar\egtm\' ad \'train.mat']);
P=i*train';
clear train;
net=newsom(minmax(P),[N N],'hextop','dist');
net.trainParam.epochs=5000;
net.trainParam.show=500;
net=train(net,P);
eval(['save ' 'C:\Matlab\work\yavuz_tez\net_gen\net1_' ad '.mat net']);
end;
```

ÖZGEÇMİŞ

1975 yılında Adapazarı'nda doğdu. İlkokul öğrenimini Gölcük, ortaokul öğrenimini İzmit ve lise öğrenimini Gebze S.T.F.A Anadolu Teknik Lisesi Bilgisayar bölümünde tamamladı.

1994 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Öğretmenliği bölümünü kazandı. 1998 yılında bu bölümden başarı ile mezun oldu. 30 Eylül 1998 yılında Adapazarı Anadolu Kız Meslek Lisesi' nde Bilgisayar Öğretmenliği görevine başladı. Öğretmenliğinin ikinci yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsünün Elektronik-Bilgisayar Eğitimi Anabilim Dalında yüksek lisansa başladı.

2001 yılı başında Kocaeli Üniversitesi Fen Bilimleri Enstitüsünde Araştırma Görevlisi kadrosu ile göreve başladı. Halen Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi bölümünde araştırma görevlisi olarak görev yapmaktadır.