KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI TABANLI KONUŞMACI TANIMA

DOKTORA TEZİ

Melih İNAL

Anabilim Dalı: Elektrik Eğitimi

Danışman: Yard.Doç.Dr. Erhan BÜTÜN

TEMMUZ 2001

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI TABANLI KONUŞMACI TANIMA

DOKTORA TEZİ

Melih İNAL

Tezin Enstitüye Verildiği Tarih : 21 Mayıs 2001

Tezin Savunulduğu Tarih : 13 Temmuz 2001

Tez Danışmanı

Yard.Doç.Dr. Erhan BÜTÜN

(.....)

Üye	Üye	Üye
Prof.Dr. Hasan DİNÇER	Doç.Dr. Yılmaz ÇAMURCU	Doç.Dr. Kadir ERKAN
(.....)	(.....)	(.....)

Üye

Yard.Doç.Dr. Engin ÖZDEMİR

(.....)

TEMMUZ 2001

YAPAY SİNİR AĞLARI TABANLI KONUŞMACI TANIMA

Melih İNAL

Anahtar Kelimeler: Yapay Sinir Ağları, Eğiticili ve Eğiticisiz Öğrenme, Konuşmacı Tanıma, Metne Bağlı Kapalı Set Konuşmacı Tanıma, Metinden Bağımsız Açık-Kapalı Set Konuşmacı Tanıma.

Özet: Bu çalışmada, çeşitli Yapay Sinir Ağları (YSA) tabanlı Konuşmacı Tanıma uygulamaları gerçekleştirilmiştir. Çok Katmanlı Almaç (ÇKA) ve Kendi Kendini Organize Eden (SOM) Yapay Sinir Ağları, eğiticili ve eğiticisiz öğrenme yöntemleridir. ÇKA ve SOM modelleri konuşmacı örüntüleri için sınıflandırıcı olarak kullanılmıştır.

Konuşmacı tanımada, özellik çıkartım önemli bir aşamadır. Bu çalışmada, özellik vektörlerinin çıkartımı için, Doğrusal Öngörülü Kodlama (DÖK) tabanlı çeşitli algoritmalar kullanılmıştır. Özellikle, kepstral katsayılar yöntemi en baskın algoritmadır. Çalışmalar, başlıca iki alanda incelenebilir: birincisi, çeşitli ÇKA mimarileri ile metne bağlı kapalı set konuşmacı tanıma ve ikincisi, SOM mimarileri ile metinden bağımsız açık-kapalı set konuşmacı tanıma uygulamalarıdır. Konuşmacı saptama uygulamalarında, SOM ağlarının çıkışında, karar birimi olarak, Birleştirilmiş Bellek Modeli (BBM) kullanılması amaçlanmıştır.

İlk alanda yapılan çalışmalarda, 10 konuşmacının yer aldığı ad ve soyadlarını telaffuz ettikleri, Türkçe konuşmacı seti kullanılmıştır. Her telaffuz 8 kez tekrarlanarak, 5 tanesi eğitim, 3 tanesi de test aşamasında kullanılmıştır. Konuşmacı sayısı ve telaffuz edilen kelime sayısı arttıkça, her konuşmacı için ÇKA sınıflandırıcısının oluşturulması ve eğitimi çok uzun zaman alır. Ayrıca sistemin tanıma verimi orantılı olarak düşer. ÇKA sınıflandırıcısının bir diğer dezavantajı ise belirli bir problem için, optimum ağ mimarisinin, deneme ve yanılma yoluyla bulunmasıdır.

İkinci alanda yapılan çalışmalarda, farklı SOM sınıflandırıcıları, Türkçe konuşmacı setinin eğitimi ve test edilmesi için, kullanılmıştır. SOM, ÇKA modeli ile karşılaştırıldığında, her bakımdan daha iyi sonuç vermiştir. Daha sonra, SOM mimarileri, TIMIT veritabanı için, yine sınıflandırıcı şeklinde kullanılmıştır. Yaptığımız çalışmalar, TIMIT veritabanını kullanan diğer çalışmalarla karşılaştırıldığında, diğer çalışmalar kadar iyi sonuç vermiştir.

ARTIFICIAL NEURAL NETWORKS BASED SPEAKER RECOGNITION

Melih İNAL

Keywords: Artificial Neural Networks, Supervised and Unsupervised Learning, Speaker Recognition, Text Dependent Closed Set Speaker Recognition, Text Independent Open-Closed Set Speaker Recognition.

Abstract: In this study, Various Artificial Neural Networks (ANN) based Speaker Recognition Applications are realized. Multilayer Perceptron (MLP) and Self Organizing Map (SOM) ANN are methods of the supervised and unsupervised learning scheme. MLP and SOM models are used as classifiers for speaker's patterns.

Feature Extraction is an important stage in the speaker recognition. In this study, Linear Prediction Coding (LPC) based various algorithms are used for extraction of the feature vectors. Especially cepstral coefficients method is the most satisfied algorithm. Studies can be examined in two major areas: first one is the text dependent closed set speaker recognition with various MLP architectures and second is text independent open-closed set speaker recognition with SOM architectures. At the SOM outputs, use of Associative Memory Model (AMM) as decision unit is proposed for the speaker identification applications.

In the first area Turkish speaker set is used and constituted by the 10 speakers with their name and surname. Each utterance is repeated 8 times, 5 of them is used in training and remaining in the test stage. When the number of words and speakers in the set increase, the MLP classifier would take too long to build and train. Also the recognition rate is dropped proportionally. Another weakness of MLP recognizers is the network architecture that is optimal for a specific problem should be found by trail and error.

In the second area, different SOM architectures are used as classifier for training and testing Turkish speaker set. When SOM is compared with MLP, SOM is found better than MLP in all aspects. And then SOM architectures are used again as classifier for TIMIT database. When our study is compared with different studies for TIMIT database, our studies give good results as much as the others.

ÖNSÖZ ve TEŞEKKÜR

Her alana rahatlıkla uygulanabilen Yapay Sinir Ağlarının; günümüzde yapılan çalışmalar doğrultusunda, bulanık mantık ve genetik algoritmalar gibi modern yöntemlerle melez kullanımları geliştirilmektedir. Bütün bu yöntemlerin kendine özgü özellikleri bir araya geldiğinde, mükemmelleşecek konuşma ve konuşmacı tanıma uygulamalarıyla, robotlar; artık bazı duygularını bir insanın tarzına yakın bir şekilde ifade ederken, dış çevreden algıladıkları verilere göre anlamlı seslerle tepki verebilecekleri, dünyanın sahip olduğu teknolojiye bakılırsa artık içinde bulunduğumuz bu yüzyılda gerçekleşecek gibi görülmektedir.

Bana bu konuda çalışma olanağı veren danışmanım Sayın Yard. Doç. Dr. Erhan BÜTÜN, tez izleme jürisi üyeleri Sayın Doç. Dr. Kadir ERKAN ve Yard. Doç. Dr. Engin Özdemir Hocalarıma teşekkürlerimi sunarım. Bu tez çalışmasında, önerilerini esirgemeyen Dr. Ruhi SARIKAYA Hocama teşekkürü bir borç bilirim. Son olarak çalışmalarımın sonuna kadar gösterdiği sabır ve manevi desteğinden dolayı sevgili eşim Seval İNAL'a da teşekkür ederim.

Bu tez çalışmasının özellikle Türkçe'de geliştirilecek çalışmalara yardımcı olmasını dilerim.

İÇİNDEKİLER

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
Δ W(t)	: t inci zaman adımındaki ağırlıkların değişimi
∇	: Gradient operatörü
$<\varepsilon>$	: Hata ifadesinin beklenen değeri
E_{vn}	: Vektör nicemleme değişimi

YSA	: Yapay Sinir Ağları
SMM	: Saklı Markov Modeli
DÖK	: Doğrusal Öngörülü Kodlama
ÇKA	: Çok Katmanlı Almaç modelli
TKA	: Tek Katmanlı Almaç
SOM	: Kendi kendini organize eden ağ modeli (Kohonen Ağı)
AFD	: Ayrık Fourier Dönüşümü
DZE	: Dinamik Zaman Eğriltimi
VN	: Vektör Nicemleme
HR	: Hatalı Reddetme
HK	: Hatalı Kabul
EHO	: Eşit Hata Oranı
GKM	: Gauss Karışımlı Model
EK	: Enyakın Komşuluk
LBG	: Linda-Buzo-Gray
YAM	: Yapay Ağaç Modeli
DYAM	: Değiştirilmiş Yapay Ağaç Modeli
BBM	: Birleştirilmiş Bellek Modeli
TIMIT	: Texas Instruments (TI) Massachusetts Ins. of Tech (MIT)

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

BÖLÜM 1. GİRİŞ

Günlük hayatta, hemen hemen her gün, nerdeyse herkes kendi kimliğini kanıtlamak zorunda kalmaktadır. Bugün kullanılan yöntemler, zaman kaybına ve bazı durumlarda uygulama hatalarına neden olmaktadır. Otomatik konuşmacı tanıma sistemleri yaşamı kolaylaştırmak adına, herhangi bir şifre hatırlamaya ya da elektronik ödemelerde bir kimlik denetimine bağlı kalmaksızın, bugünkü güvenlik sistemlerinin yerini almakta büyük aşamalar kaydetmektedir.

Konuşma tanıma teknolojisi son on yılda büyük ilerlemeler göstermiştir. Daha bir kaç yıl öncesine kadar söylenen kelimeler arasında boşluk verilerek tanıma işlemi yapılabilirken (Furui 1995), bu günlerde aralıksız konuşulan bir konuşma için bile konuşma tanıyıcı sistemler ticari anlamda kullanılmaktadır. Bilgisayar teknolojisindeki gelişmeler sonucu, günümüzde gerçek zamanda konuşma ve konuşmacı tanıma gibi karmaşık uygulamalar gerçekleştirilmektedir. Bu tez çalışması, konuşmacı tanıma uygulamalarını araştırmak ve bir Türkçe konuşmacı tanıma sistemi oluşturmak üzere hazırlanmıştır.

Dayanıklı bir konuşmacı tanıma sistemi sayesinde, artık kullanıcıların anahtar ya da akıllı kart taşımaya ya da herhangi bir kullanıcı şifresi hatırlamaya ihtiyacı yoktur. Çünkü, kişilerin sesi bu işlemlerin yerini almaktadır. Sınıflandırılabilen ses işaretleri, çalınması ya da unutulması güç, bir çeşit pasaport gibi kullanılabilir. Ses postası, bilgisayar aracılığı ile bir veritabanına erişim ya da alışveriş yapma gibi uygulamalar artık konuşmacı tanıma sistemlerinin kontrolünde gerçekleştirilebilir. Bir diğer kullanım alanı olarak adli olaylar sayılabilir; bir suçlamaya ilişkin alınmış ses kaydı, bilinen suçlu kayıtlarıyla karşılaştırılarak, asıl suçlu bulunabilir ya da yapılacak elemeler sonucunda araştırmanın sınırları daraltılabilir.

1.1. Tez Çalışmasının Amacı

Yapılan bir çok çalışmada, araştırmacılar; Yapay Sinir Ağları (YSA) kullanarak konuşmacı tanıma sistemi geliştirmektedirler. Bu tez çalışmasında, konuşmacı sınıflandırma işlemi için gerekli YSA mimarilerinin, hangi parametrelerle uyumlu bir şekilde çalıştığı konusu üzerinde yoğunlaşacaktır.

Araştırma Süresince Yanıtları Aranacak Sorular

- Hangi YSA mimarisi, konuşmacıları birbirinden ayırt etmek için gerekli bilgileri en iyi şekilde öğrenir.
- Konuşmacılar arasındaki farklılıkları, hangi özellik çıkartım algoritması en iyi şekilde ortaya koyar.
- Hangi özellik çıkartım algoritması, hangi YSA modeli ile en iyi sonucu verir.

1.2. Tez Çalışmasının Önemi

Konuşmacı tanıma ve YSA konusunda yapılan çalışmalarda, Saklı Markov Modeli'nin (SMM) zamana bağlı değişim gösteren işaretler için iyi bir yöntem olduğu ön görülmektedir. Fakat, bu yaklaşım, ses işaretlerinden elde edilen özellik çıkartım (feature extraction) vektörleri için yeterince iyi bir model oluşturmayabilir. Konuşmacı tanıma uygulamalarında iyi sonuç veren bir özellik çıkartım algoritması, YSA'lar için uygun bir taban oluştururken, SMM'lerle uyumlu bir çalışma göstermeyebilir. Eğer, konuşmacı tanıma uygulamalarında en iyi sonucu verecek YSA mimarisi ve özellik çıkartım algoritması belirlenebilirse, YSA'ların daha baskın olabileceği söylenebilir.

Otomatik konuşmacı tanıma alanındaki gelişmelere rağmen, artalan gürültüsü, kayıt ve test zamanı değişik mikrofon kullanılması, telefon hatlarındaki bozukluklar (girişim, yankı v.b.) ve konuşmacının sağlık durumu (soğuk algınlığı gibi) ses verimini düşürmektedir. Eğer kayıt ortamı ve iletişim koşulları hesaba katılmazsa sistem güvenirliliğinin %100 seviyelerinde olacağı kesindir.

1.3. Tez Çalışmasının Amaçlanan Basamakları

Yapılan araştırmalarda Türkçe dili için oluşturulmuş bir veritabanı olmadığından, bu çalışma boyunca toplanan konuşma verileri, ilerde Türkçe konuşmacı veritabanı oluşturulmak üzere kullanılabilir. Ayrıca, incelenen konuşmacı veritabanları içinde kullanımı esnek ve çok amaçlı işlenebilen bir veritabanı seçimi yapılarak, her iki veritabanı da, tasarlanan sınıflandırıcıların verimini belirlemek için denenecektir. Yapılan çalışmalarda, MATLAB programının 5.2 sürümü ve "Creative Sound Blaster 16 Value PNP" ses kartı kullanılmıştır.

Yapılan araştırmalar doğrultusunda, tez çalışması boyunca yapılması amaçlanan basamaklar aşağıdaki gibi maddeler halinde sıralanmıştır.

1. Türkçe konuşmacı verisinin hazırlanması ve literatürde kullanılan uygun bir konuşmacı veritabanın seçimi.

2. Özellik çıkartım algoritmalarının konuşma verisine uygulanması ve özellik çıkartım vektörlerinin elde edilmesi.

3. YSA sınıflandırıcı tasarımının yapılması ve değişik YSA algoritmalarına elde edilen özellik vektörlerinin uygulanması.

4. Literatürde yaygın bir şekilde kullanılan bir konuşmacı veritabanı kullanılarak, özellik çıkartım algoritması ve YSA sınıflandırıcısının bu veritabanına uygulanması.

5. Yukarıda sayılan maddelerdeki çalışmaların sonuçları elde edilerek, genel bir değerlendirme yapılıp en uygun YSA modeli ve özellik çıkartım algoritmasının belirlenmesiyle çalışmanın sonuçlarını değerlendirmek.

1.4. Tez Çalışmasını Oluşturan Bölümler

Tezin diğer bölümlerinde sırasıyla aşağıdaki konular anlatılmıştır:

Bölüm 2'de, Konuşma ve Konuşmacı Tanıma Sistemleri incelenmiştir. Konuşmacı tanıma sistemlerinin aşamaları, özellik çıkartım algoritmaları ve bu algoritmalardan en yaygın olarak kullanılan Doğrusal Öngörülü Kodlama (DÖK) algoritması açıklanacaktır. Daha sonra, Örüntü tanıma sınıflandırıcıları, konuşmacı tanıma uygulamaları açısından değerlendirilerek genel bir sınıflandırıcı yapısı tanımlanarak, bu yapının çeşitli aşamaları anlatılacaktır. Bu bölümde YSA'lar dışındaki sınıflandırıcılara değinilecektir. YSA sınıflandırıcıları bir sonraki bölümde ayrıca incelenecektir. Yine bu bölümde, literatürde söz edilen çeşitli konuşmacı veritabanları ve özellikleri incelenecektir.

Bölüm 3'te, bu tez çalışmasında kullanılacak YSA sınıflandırıcılarına ait öğrenme algoritmaları, öğrenme modları ve kuralları incelenmiştir. Ayrıca, konuşmacı tanıma alanında en yaygın kullanılan YSA sınıflandırıcıları tanıtılacaktır. Ayrıca, literatürde yer almış, Türkçe ve diğer dillerde yapılan makale ve tez çalışmalarının yer aldığı Konuşmacı Tanıma uygulamaları açıklanacaktır.

Bölüm 4'te, bu tez çalışmamızda kullanacağımız, genel konuşmacı tanıma sistemindeki aşamalar ve Türkçe konuşmacı veritabanının oluşturulma aşamaları açıklanacaktır. Daha sonra, Kendi kendini organize eden (Self Organizing Map-SOM) YSA sınıflandırıcısının çıkışında, karar birimi olarak Birleştirilmiş Bellek Modeli (BBM) ağının kullanıldığı, karma yapı açıklanacaktır.

Bölüm 5'te, eğiticili öğrenme algoritmalarından Çok Katmanlı Almaç (ÇKA) YSA sınıflandırıcılarının kullanıldığı, metne bağlı kapalı set konuşmacı tanıma uygulamalarına ilişkin çalışmalar açıklanacaktır. Bu çalışmalar maddeler halinde aşağıdaki gibi özetlenmiştir.

1. 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 kepstral (cepstral) katsayılar sayılabilir (Rabiner 1993). Yaptığımız çalışmada bu katsayılardan kepstral katsayıların, konuşmacı tanıma sisteminin verimi açısından en etkin yöntem olduğu görülmüştür (İnal 2000).

Bu çalışmada her telaffuza ait özellik vektörleri, Çok Katmanlı Almaç Modeli-ÇKA kullanılarak eğitilmiştir. Bu çalışmada kepstral katsayıları, yinelemeli bir yöntemle DÖK katsayılarından türetilmiştir.

2. ÇKA Mimarisinin Sistem Verimine Etkisi: Bu çalışmada, ÇKA mimarisinin sistem verimine etkisi araştırılmıştır. ÇKA sınıflandırıcılarının, saklı katmandaki işlem birimi sayısı, sırasıyla 16, 32 ve 64 olarak değiştirilmiştir. Her mimari için ağın eğitiminde, toplam karesel hata 0.001 seçilmiştir. Önceki çalışmadan farklı olarak antikonuşmacıların çıkış değerleri "-1" seçilmiştir. Bu sınıflandırıcılar, konuşmacı saptama sistemine uygulanarak elde edilen sonuçlar değerlendirilmiştir (İnal 2001).

3. Değişik ÇKA Mimarilerinin İncelenmesi: Bir önceki çalışmada saklı katmanlarda kullanılan işlem birimi sayısına göre ortalama verimler birbirine çok yakın çıkmıştır. Bu durum, saklı katman işlem birimi sayısının daha detaylı incelenmesini gerektirmektedir. Bu nedenle tekrar bir çalışma yapılmıştır. Çalışmada, saklı katmanda sırasıyla 8, 16, 24, 32 ve 64 işlem birimi için yine istenen çıkış değerleri ±1 seçilmiştir. ÇKA ağlarında sigmoid transfer fonksiyonu kullandığımızdan işlem birimleri çıkışları daima ±1 aralığında sınırlıdır. İstenen çıkış değerleri de ±1 seçilirse, daima kalıcı bir hata bandı ±1 civarında oluşabilir. Bu düşünce ile, saklı katman işlem birimi sayısı 24 olan ağın istenen çıkış değerleri ±0.9 olarak seçilerek yine diğer ağlar gibi eğitilmiştir. Bu ağ için işlem birimi sayısının 24 seçilmesinin nedeni; giriş katmanındaki işlem birimi sayısı 12 olduğundan yapılan çalışmalarda genellikle "saklı katmandaki işlem birimi sayısı giriş katmanındaki işlem birimi sayısının iki katı kadar seçilir" düşüncesidir.

Bölüm 6'da, SOM YSA ağı tabanlı, metne bağlı-metinden bağımsız ve açık-kapalı set konuşmacı tanıma uygulamaları ve bu uygulamalara ilişkin sonuçlar açıklanacaktır. Ayrıca, Türkçe metne bağlı kapalı set konuşmacı saptama sistemi

oluşturularak, gerçek zamanda test edilecektir. Yapılacak konuşmacı saptama uygulamalarında, SOM ve BBM karma ağ yapısının kullanılması amaçlanmaktadır. Bu çalışmalar maddeler halinde aşağıdaki gibi özetlenmiştir.

1. SOM Sınıflandırıcısının Kullanıldığı Türkçe Metne Bağlı Konuşmacı Tanıma Uygulamaları: Bu çalışmada her kullanıcı için ayrı ayrı SOM ağları oluşturulmuş olup tek katmanda iki boyutlu 10X10 işlem birimi kullanılarak, 5000 epok (epoch) için ağın eğitimi yapılarak, konuşmacı saptama ve doğrulama uygulamaları yapılmıştır.

2. Tüm Konuşmacılar için bir tek SOM Ağının Kullanıldığı Uygulama: Önceki çalışmada, Türkçe metne bağlı konuşmacı saptama uygulamasında, her konuşmacı için ayrı bir SOM sınıflandırıcı ağı tanımlanırken, bu çalışmada tek bir ağ kullanılarak bütün konuşmacı seti için sınıflandırıcı verimi araştırılmak üzere, tek katmanda iki boyutlu 20X20 işlem birimi kullanılarak, 10000 epok için ağın eğitimi yapılmıştır.

3. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Kapalı Set Konuşmacı Saptama Alanına Uygulanması: Bu çalışmada, TIMIT veritabanı kullanılarak metinden bağımsız kapalı set konuşmacı saptama uygulamaları yapılmıştır. Kapalı set konuşmacı saptama uygulaması sırasıyla 5, 10 ve 20 konuşmacı için gerçekleştirilmiştir. Öncelikle tüm konuşmacılara ait tek bir SOM ağı, iki boyutlu 25X25 işlem birimine sahip ağ mimarisi ile 10000 epok için eğitilmiştir. Eğitim süresinin etkisini araştırmak üzere, her konuşmacı için ayrı ayrı tanımlanmış SOM ağları, iki boyutlu 20X20 işlem birimine sahip ağ mimarisiyle hem 5000 hem de 10000 epok için eğitilmiştir. Daha sonra ağdaki işlem birimi sayısının etkisini araştırmak üzere yine iki boyutlu ağ mimarisi için 10X10 işlem birimi seçilerek ağın eğitimi 5000 epok için gerçekleştirilmiş ve elde edilen sonuçlar değerlendirilmiştir.

4. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Kapalı Set Konuşmacı Doğrulama Alanına Uygulanması: Bu çalışmada yine TIMIT veritabanı kullanılarak, 10 ve 20 konuşmacı için metinden bağımsız kapalı set konuşmacı

doğrulama uygulamaları yapılmıştır ve sonuçlar değerlendirilmiştir. 10 konuşmacının eğitimi için iki boyutlu 10X10 işlem birimine sahip ağ kullanılarak, 5000 epok için eğitimi yapılmıştır. 20 konuşmacı için iki boyutlu 20X20 işlem birimine sahip ağ kullanılarak, 10000 epok için eğitim yapılmış ve elde edilen sonuçlar değerlendirilmiştir.

5. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Açık Set Konuşmacı Saptama Alanına Uygulanması: Yine TIMIT veritabanı kullanılarak, bu çalışmada metinden bağımsız açık set konuşmacı saptama uygulaması 20 konuşmacı için iki boyutlu 20X20 işlem birimine sahip ağ kullanılarak, 10000 epok için eğitimi yapılarak sonuçlar değerlendirilmiştir. Bu çalışmada konuşmacıların sisteme kabul edilme eşik değeri %70 olarak tanımlanmıştır. Ağın eğitiminde kullanılmayan 18 sahte konuşmacı, SOM ağlarının test aşamasında kullanılmıştır.

6. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Açık Set Konuşmacı Doğrulama Alanına Uygulanması: Bu çalışmada yine TIMIT veritabanı kullanılarak, 10 ve 20 konuşmacı için metinden bağımsız açık set konuşmacı doğrulama uygulamaları yapılmıştır ve sonuçlar değerlendirilmiştir.10 konuşmacı için, iki boyutlu 10X10 işlem birimi kullanılan ağın, 5000 epok için eğitimi yapılmıştır. Bu uygulamada 10 sahte konuşmacı ağın test aşamasında kullanılmıştır. 20 konuşmacı için iki boyutlu 20X20 işlem birimi kullanılan ağın, 10000 epok için eğitimi yapılarak elde edilen sonuçlar değerlendirilmiştir. Ağın eğitiminde kullanılmayan 18 sahte konuşmacı, SOM ağlarının test aşamasında kullanılmıştır. Bu çalışmalarda da konuşmacıların sisteme kabul edilme eşik değeri %70 olarak tanımlanmıştır.

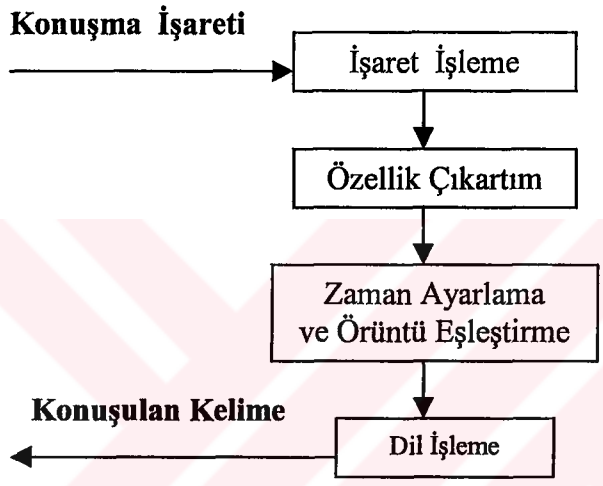
7. Türkçe Metne Bağlı Kapalı Set Konuşmacı Saptama Sistemi Oluşturulması: Bu çalışmada diğer çalışmalarda olduğu gibi MATLAB 5.2 programı yardımıyla önceden hazırladığımız Türkçe veritabanını kullanarak metne bağlı kapalı set konuşmacı saptama sistemi hazırlanmıştır. Sisteme, saptama işlemi için başvuran konuşmacı, sistem tarafından tanınırsa, kabul edilecektir. Eğer tanıma işleminde, eşleştirme yeterince iyi değilse tekrar yeni deneme yapılması sistem tarafından istenebilir ya da konuşmacı sisteme kabul edilmeden reddedilebilir. Bu çalışmaya

ilişkin sonuçlar değerlendirilerek, bu sistemin program çıktısı tez çalışmasının sonuna eklenmiştir.

Bölüm 7'de, yaptığımız tüm çalışmaların sonuçları değerlendirilerek, önceden yapılmış çalışmalarla karşılaştırılacaktır. Ayrıca, konuşmacı tanıma sistem veriminin iyileştirilmesi için çeşitli öneriler yapılacaktır.

BÖLÜM 2. KONUŞMA VE KONUŞMACI TANIMA SİSTEMLERİ

Konuşmacı tanıma sistemlerini incelemeden önce, konuşma tanıma sistemlerini incelemek daha doğru olur. Şekil 2.1'de genel konuşma tanıma modeli görülmektedir.



Şekil 2.1. Genel konuşma tanıma modeli

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

1. İşaret işleme birimi, konuşma işaretinin bilgisayar ortamında sayısallaştırarak işlenebilmesi için kullanılmıştır. Bu birimin amacı; örneklenmiş konuşma işaretini, genlik değişimlerinden, konuşmacının aksanı ve 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 sonucunda elde edilen vektörlere göre eşleştirme işlemini yapar. Zaman ayarlama; konuşma hızındaki değişimlerin sonucu, telaffuzlarda oluşan zamana bağlı bozulmalara neden olan ses bilgilerinin ayarlanmasıdır.

4. Son olarak dil işleme birimi konuşulan kelimenin, kural tablosundan seçimi için kullanılır.

Konuşma ve konuşmacı tanıma uygulamalarında, sayısallaştırılmış bir ses verisi artık işlenebilir örüntü verisine dönüştürülmüş olur. Bu amaçla istatistiksel örüntü tanıma işlemleri büyük önem arz etmektedir.

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 Ayrık Fourier Dönüşümü (AFD), süzgeç bankası ya da Doğrusal Öngörülü Kodlama (DÖK) incelemeleri gibi spektral inceleme teknikleri uygulanır.

2. Örüntü eğitimi, aynı sınıfa ait konuşma seslerinin ilgili bir veya birden fazla test örüntüleri için özellikler oluşturmakta kullanılır. Sonuçta oluşan ve genellikle "referans örüntüsü" olarak adlandırılan örüntü; bir takım ortalama alma tekniklerinden üretilmiş bir örnek ya da model olabilir, veya 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 Eğriltimi (DZE) algoritması diye adlandırılan ve farklı konuşmalara sahip iki

örüntüyü dengeleyen (zaman ölçeklemesi) bir 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.

Zaman Ayarlama ve Örüntü Eşleştirme modelleri, konuşmacıdan bağımsız konuşma tanıma sisteminde, konuşmacıya göre değişebilen; telaffuz ve bölgesel aksan farklılıkları gibi durumlarda kullanılabilir. Bu modelde, telaffuz ve model arasındaki benzerliği bulmak için bir maliyet ya da olasılık fonksiyonu kullanılır. Bu problemin karmaşıklığı, kelime büyüklüğü ve konuşma hızı ile ilgilidir. Zaman Ayarlama ve Örüntü Eşleştirme algoritmalarının en önemli özelliği, hem spektral hem de Zaman Eğriltimine (Warping) uğramış kelimeler veya akustik olayları eşleştirebilmesidir. Konuşma hızı, zaman ekseninde doğrusal olmayan eğriltimi arttırır. Konuşmanın zamana bağlı yapısını modelleyen örüntü eşleştirme algoritmalarının veya bir diğer adıyla kelime sezim algoritmalarının en çok kullanılanları; Dinamik Zaman Eğriltimi (DZE), Saklı Markov Modeli (SMM) ve Kelime Telaffuz Modelleridir.

DZE, model ve telaffuzdan elde edilen özellik vektörlerini karşılaştırmak için bir uzunluk ölçüm "Distortion Metric" algoritması kullanır ve bu algoritmaya göre tüm vektörler için yapılan ayarlamanın genel bir maliyetini bulur. Eğer model ve telaffuzların belirlenmesinde Vektör Nicemleme (VN) algoritması kullanılıyorsa, uzunluk ölçümü önceden hesaplanarak bir tabloda saklanabilir.

DZE algoritmaları en uygun ayarlama (alignment) yörüngeleri için her kelime modeli ile giriş telaffuzunu karşılaştırır. DZE, bir uzunluk ölçümü kullanarak, kelime modeli üzerinden bir minimizasyon problemi gibi ifade edilebilir. Ayrık telaffuz tanıma sistemleri için DZE algoritması, her telaffuzun bitim noktasında başlar. Sürekli konuşma tanıma sistemlerinde ise her telaffuzun başında ve sonunda DZE algoritması kullanılır. Böylelikle, DZE algoritması kelimelerin bölütlenmesinde de kullanılabilir.

DZE zaman ekseninde eğriltim işlemi için uygun bir algoritma olmasına rağmen dezavantajları da bulunmaktadır. Örneğin, eğer konuşmacının aksan ve telaffuz gibi değişimleri kelime modelinde sunulmuşsa, algoritma etkili olabilir. Bu nedenle bir kelimeye ait birden fazla kelime modeli yaygınca kullanılmaktadır. Konuşmacıya bağımlı konuşma tanıma sistemi için, kelime başına 2 ila 5 arasında değişen kelime modeli yeterli olmaktadır. Konuşmacıdan bağımsız tanımada ise kelime başına, 12 kelime modeli kullanmanın verimi arttırdığı görülmüştür. Kelime modeli; Vektör Nicemleme (VN) gibi algoritmalar kullanılarak belirlenebilir. Böylece DZE'nin bulacağı çözüm, konuşmacıya bağlı otomatik konuşma tanıma sistemleri için geçerli bir algoritma olabilir fakat konuşmacıdan bağımsız sistemler için olmayabilir. Bu durum özellikle geniş kelime tanıma sistemleri için doğrudur.

Kelime telaffuz modelleri, her kelimenin birden fazla farklı telaffuzları için sonlu sayıda farklı durum modeli içerir. Telaffuz modelindeki durumlardan her hangi bir tanesi istenen değere ulaştığında kelime tanınmış olur. Bu modelin iyi bir yöntem olmasını sağlayan bir diğer özelliği ise; konuşmacıların aksan ve üslup değişimlerini çok iyi sezinleyebilmesidir. Modelin dezavantajı ise, sözlükteki her kelime için belirli sayıda model oluşturmak gerekliliğidir. Ek olarak bazı kelimelerin oldukça farklı telaffuz şekillerinin olmasıdır. Bu durum da her telaffuz için önemli sayıda eğitim yapılmasını gerektirmektedir.

Bu bilgiler ışığında konuşmacı tanımada benzer işlemler yapılacaktır. Tanıyıcı sistem bir konuşmacı setindeki konuşmacılara ait örüntüleri sınıflandırıldıktan sonra, artık o sete ait konuşmacıları da sınıflandırılmış olur. Bu tez çalışmasında, örüntü tanıma ve sınıflandırma işlemi yapılacaktır.

Konuşmacı tanıma uygulamaları, iki sınıfa ayrılır;

Konuşmacı Saptama: Sisteme giriş için başvuran konuşmacının bilinen konuşmacılar listesi içerisinden kim olduğunun saptanması işlemi olarak tanımlanabilir.

Konuşmacı Doğrulama: Kendini sisteme tanıtan bir konuşmacının, kim olduğunun doğrulanarak sisteme kabul edilmesi ya da reddedilmesi 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. Sistem tarafından karşılaştırılacak konuşmacıların sayısı fazla ise doğal olarak saptama işlemi daha uzun sürecektir.

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 da daha azdır. Karşılaştırma sonucuna göre, eğer eşleştirme oranı yeterince yüksekse konuşmacı sisteme kabul edilir. Eğer oran çok düşük yada yeterince yüksek değilse, konuşmacı daha fazla konuşma örneğine göre bir kaç kez test edilir, sonuç değişmez ise konuşmacı sistem tarafından reddedilir.

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).

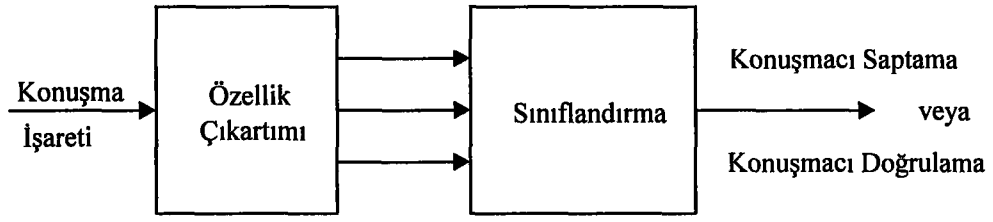
Metne bağlı yapılan tanıma işleminde, konuşmacının telaffuz edeceği kelime ya da cümleler tanıyıcıya çok iyi öğretilmelidir. Çünkü, telaffuzun zaman eksenindeki değişikliği ya da konuşmacının soğuk algınlığı gibi sesinin akustik özelliklerini değiştirecek olumsuz durumlar tanıyıcı sistemi yanıltabilir. Bir başka olumsuz durum, metne bağlı tanıma yapıldığı için sahtekar bir konuşmacı bir başka konuşmacının telaffuzunu yüksek teknolojiye sahip bir ses kaydedici yardımıyla kayıt ederek taklit etme yoluna gidebilir. Bu da sistem güvenliğini olumsuz yönde etkileyecektir. Konuşma işaretinin Zaman Eğriltimi (Time Warping), metne bağlı konuşmacı tanıma uygulamalarında, tanıma işleminin gerçekleşmesi için yeterli olabilir. Zaman Eğriltim işlemi Yapay Sinir Ağı tabanlı algoritmalarla da gerçekleştirilebilir (Levine 1995).

Metinden bağımsız yapılan tanıma işleminde, rastsal telaffuz yapılabildiğinden sahte kayıtlara karşı daha fazla güvenlidir. Canlı bir kayıt işlemiyle yapılan test ya da birkaç kelimelik bir cümle konuşmacıya telaffuz ettirilebilir. Ancak konuşmacıya telaffuz ettirilecek cümle ya da kelimeler yeterince rasgele seçilmezse ve tanıyıcı sistem, taklidi kolay telaffuzları içeriyorsa, yine de aldatıcı, sahte kayıtlar tanıyıcının güvenirliliğini sarsabilir.

Bir konuşmacı doğrulama sisteminde iki tip hata kriteri vardır: Hatalı Reddetme (HR) ve Hatalı Kabuletme (HK). HR kriteri, sisteme kabul edilmesi gereken doğru bir konuşmacının reddedilmesi, HK ise sisteme kabul edilmemesi gereken sahte bir konuşmacının kabulü olarak tanımlanabilir. HR ile HK kriterlerinin kesişimi Eşit Hata Oranı (EHO) (HR $\cap$ HK = EHO) şeklinde tanımlanabilir. Bu kesişimin sonucuna göre, HR kriterinin düşürülmesi, HK'nın artışına neden olacağı gibi tersi bir durum da söz konusu olabilir. Bu nedenle sistem güvenliği açısından uzlaşıcı bir denge kurulmalıdır (Krishnamoorthy 1998)

Genel konuşmacı tanıma sistemi Şekil 2.2 'de görülmektedir. Tanıyıcı sistem, önce istenen özellikleri konuşma işaretinden elde eder. Elde edilen özellikler daha sonra

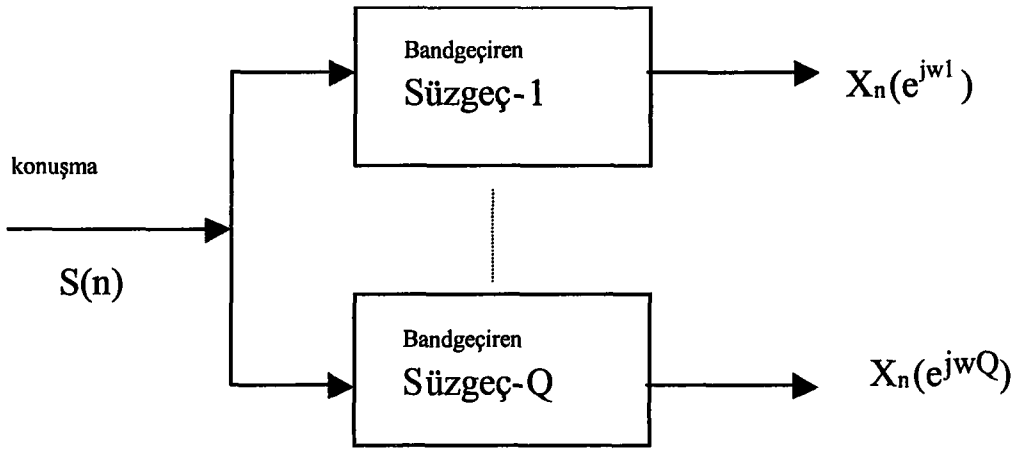
saptama ya da doğrulama işlemi için gerekli kararı vermek üzere sınıflandırıcıya giriş olarak uygulanır.



Şekil 2.2. Konuşmacı tanıma sistemi

2.1. Özellik Çıkartım Yöntemleri

Geçen bir kaç yıl boyunca bir çok araştırmacı, bugün örüntü tanıma işleminde karşılaşılabilecek sınıflandırma problemlerinin çözümü için gerekli parametrelerin elde edilmesinde çeşitli özellik çıkartım yöntemleri geliştirmişlerdir. Özellik çıkartım yöntemlerin amacı; konuşma işaretinin parametrik özelliklerini belirlemektir. Bu yöntemlerden bir tanesi kısa-zaman spektrum incelemeleridir. Bu inceleme türünün temeli Süzgeç Bankası Modelidir. Süzgeç bankası modelinin genel yapısı Şekil 2.3'te görülmektedir. Sayısallaştırılmış konuşma işareti S(n), frekans aralıkları (telefon konuşma işaretleri için 30-3000 Hz, genişband işaretler için 100-8000 Hz) işlenecek işaretin örnekleme frekansına göre Q adet bandgeçiren süzgeç bankası üzerinden geçirilmektedir. Süzgeç bankası modelinin her bandgeçiren süzgeci, konuşma işaretinin birbirinden bağımsız değişik frekans aralıklarındaki X(n) işaretlerini üretir.



Şekil 2.3. Süzgeç bankası inceleme modeli

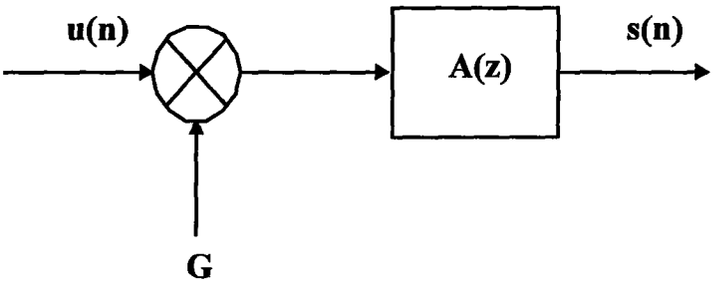
Konuşmacı tanıma işleminde kullanılan özellik çıkartım yöntemlerinden, Süzgeç bankası modeline ek olarak, ses perdesi saptaması (pitch), konuşmacının örnek telaffuzlarına göre karakteristik ses frekans aralığını veren formant frekanslarının (formant frequincies) ve konuşma yoğunluğu (intensity) belirlenmesi ve son olarak Doğrusal Öngörülü Kodlama (DÖK) "Linear Prediction Coding (LPC)" yöntemleri sayılabilir.

2.2. Doğrusal Öngörülü Kodlama (DÖK) Modeli

Konuşmacı tanımada DÖK işleminin ne kadar önemli ve bu derece yaygın bir aşama olduğunu anlamak için aşağıdaki nedenleri sıralayabiliriz (Rabiner 1993):

1. DÖK, konuşma işaretinin iyi bir modelini oluşturur. Konuşma işaretinin sessiz ve geçiş bölgeleri boyunca, DÖK modeli sesli bölgelere göre daha az etkili olmasına rağmen konuşma tanıma alanında hala etkili bir modeldir.
2. DÖK modeli; sesi üreten ses genlik (vocal tract) karakteristiklerini en iyi ifade eder.
3. DÖK analitik olarak ifade edilebilen bir modeldir. Yöntem matematiksel açıdan doğru ve basittir. Doğrudan yazılım veya donanım olarak uygulanabilir. DÖK işlemindeki hesaplamalar, süzgeç bankası modelindeki tüm sayısal uygulamalardan oldukça az ve basittir.
4. DÖK modeli tanıma uygulamalarında iyi sonuç verir. DÖK tabanlı konuşma ve konuşmacı tanıyıcı sistemlerin verimi, süzgeç bankası tabanlı olanlarla karşılaştırılmayacak derecede daha iyidir.

Şekil 2.4'te bir doğrusal öngörülü kodlama modeli görülmektedir.



Şekil 2.4. Konuşma İşaretinin Doğrusal Öngörülü Kodlama Modeli

Konuşma işaretinden özellik vektörlerinin çıkartımı için DÖK incelemesi yapılarak DÖK tabanlı parametre takımları bulunabilir. Bu parametre takımının katsayıları: doğrusal öngörülü (a) ve kepstral (c) katsayılar sayılabilir.

DÖK modeline, n anında verilen konuşma örneği s(n), kendisinden önceki k konuşma örneklerinin doğrusal bir birleşimi şeklinde tanımlanabilir (Kayran 1992):

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_k s(n-k) \qquad (2.1)$$

Eşitlik 2.1'de s(n) ifadesine; n anında verilen s(n) konuşma örneğinin kestirimi denir. s(n) konuşma örneği ve kestirimi arasında oluşan fark ifadesine e(n) öngörü hatası denir. İnceleme penceresi boyunca oluşan hatanın, ortalama karesel ifadesinin minimum değere düşürülmesi için *a* parametre takımındaki katsayılar yenilenir (Makhoul 1975). Yenilenen katsayılar artık bu inceleme penceresi boyunca sabit olarak kalır. Eşitlik 2.1'e Şekil 2.4'te gösterildiği gibi Gu(n) terimini de ekleyerek aşağıdaki eşitlik elde edilebilir:

$$s(n) = \sum_{i=1}^{k} a_i s(n-i) + Gu(n) \qquad (2.2)$$

Eşitlik 2.2'de u(n) uyartım işareti ve G uyartım kazancı olarak adlandırılır. Eşitlik 2.2 'ye z ayrık zaman dönüşümünü uygularsak Eşitlik 2.3 elde edilebilir:

$$S(z) = \sum_{i=1}^{k} a_i z^{-i} S(z) + GU(z) \qquad (2.3)$$

Eşitlik 2.3'teki sistemin transfer fonksiyonu aşağıdaki gibi ifade edilebilir:

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{k} a_i z^{-i}} = \frac{1}{A(z)} \qquad (2.4)$$

DÖK modelinden türetilen kepstral (cepstral) katsayılar bir diğer özellik çıkartım yöntemi olup, bu alanda uygulanan en baskın yöntemdir. Kepstral katsayıları, işaretin güç spektrumunun logaritmasının ters Fourier dönüşümü şeklinde ifade edilebilir. Bu

katsayılar, konuşma işaretinin ses genlik yanıtını ifade eder. Kepstral katsayılar, DÖK modelindeki *a* parametre takımındaki katsayıların yinelemeli ilişkisinden aşağıdaki gibi elde edilebilir (Farell 1994):

$$c_1 = a_1$$
$$c_n = \sum_{i=1}^{n-1} (1 - \frac{i}{n}) a_i c_{n-i} + a_n \qquad (2.5)$$
$$1 < n \leq p$$

Eşitlik 2.5'te *p*, *a* ve *c* parametre takımı katsayılarının derecesini göstermektedir.

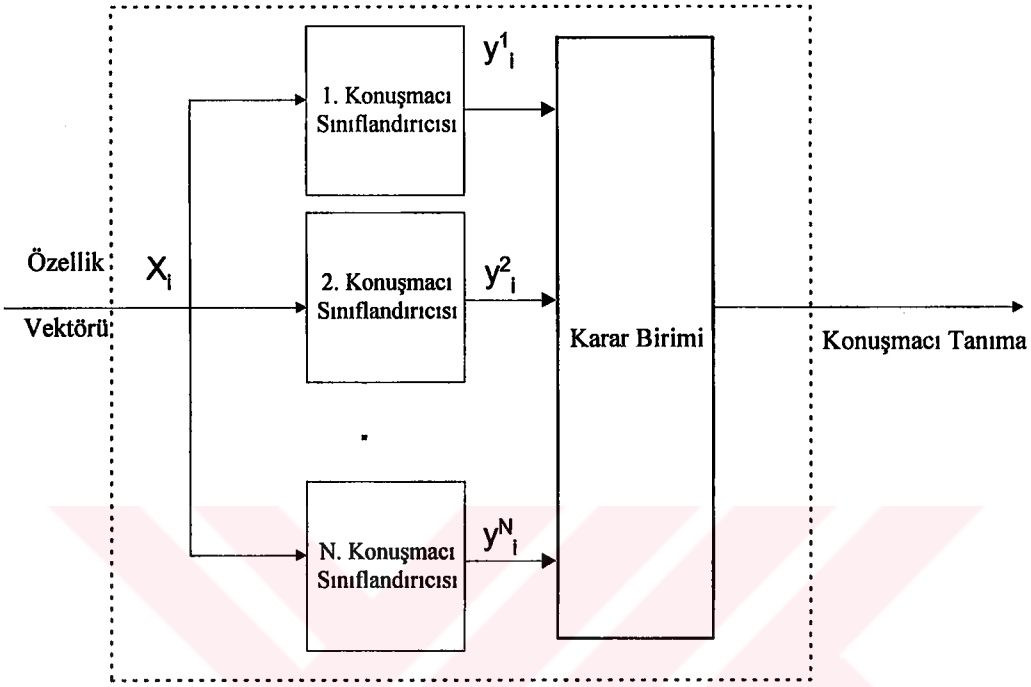
2.3. Örüntü Tanıma Sınıflandırıcıları

Konuşmacı tanıma sistemi, her konuşmacıyı ayrı bir sınıflandırıcısı olacak şekilde modeller. Şekil 2.2'deki genel konuşmacı tanıma sistemine göre sınıflandırma aşaması Şekil 2.5'teki gibidir. Şekil 2.5'teki her konuşmacı modeli, verilen özellik vektörü ile karşılaştırılır. Bu sonuca göre en iyi benzerliği gösteren konuşmacı modeli, bu özellik vektörü için belirlenmiş olur.

Konuşmacı tanıma işleminde kullanılacak, farklı örüntü sınıflandırıcıları vardır. Ayrıca bu sınıflandırıcıların denenmesi için değişik amaçlar için hazırlanmış konuşmacı veritabanları mevcuttur. Bu bölümde örüntü sınıflandırmada kullanılan bir takım teknikler ve konuşmacı veritabanları incelenecektir. Bu tekniklerden ilki, bir test ve referans telaffuzuna ait özellik vektörleri arasındaki uzaklığın hesaplandığı Euclid uzaklığıdır. Bir başka teknik olarak, Vektör Nicemleme (VN) sınıflandırıcısı konuşmacı tanıma alanına uygulanmıştır (Morgan 1991). Ayrıca örüntü sınıflandırıcıları olarak çeşitli YSA modelleri kullanılmaktadır. Bu tez YSA uygulamaları üzerine yoğunlaştığı için YSA sınıflandırıcıları ayrı bir bölümde incelenecektir.

Konuşmacı tanımada kullanılan Vektör Nicemleme gibi baskın sınıflandırıcılar, eğiticisiz öğrenme algoritmalarına temel oluşturur. Konuşmacı tanımada popüler bir VN tabanlı sınıflandırıcı olarak Ayrık SMM sayılabilir. Bu çalışma, daha çok YSA'lar üzerinde yoğunlaştığı için SMM tabanlı uygulamalar incelenmemiştir. Bir

başka eğiticisiz sınıflandırıcı ise Gauss Karışımlı Model (GKM) (Gaussian Mixture Models) sayılabilir. Bu yöntem, her konuşmacıya ait eğitim vektörlerini Gauss karışımlarının toplamları şeklinde ifade eder (Haykin 1999).



Şekil 2.5. Konuşmacı Tanıma Sistemi için Sınıflandırıcı Yapısı

Belirli konuşmacı modelleri için tasarlanmış sınıflandırıcılar hem eğiticili hem de eğiticisiz öğrenme algoritmalarını kullanabilirler. Eğiticili öğrenme algoritmalarını kullanan sınıflandırıcılar (ÇKA, Karar Ağaçları gibi), her konuşmacı modeli için, bütün konuşmacılara ait konuşma verisi ile ifade edilir. Böylece, bu sınıflandırıcılar hem konuşmacı hem de "antikonuşmacı" (sınıflandırılması yapılacak konuşmacı dışında kalan diğer konuşmacılar) konuşma verileri kullanarak tasarlanır. Eğiticisiz öğrenmeli sınıflandırıcılarda (Vektör Nicemleme, Gauss Karışımlı Model gibi) ise, her konuşmacı modeli sadece o konuşmacının konuşma verisi ile eğitilir. Bu sınıflandırıcılar aşağıdaki bölümlerde incelenmiştir.

2.3.1. Enyakın komşuluk (EK)

EK ya da daha genel anlamda kEK (Burada k=1,2,... gibi komşulukları ifade etmektedir) bir sınıfa ait verinin *Olasılık Yoğunluk Fonksiyonunu* tahmin eden bir yöntem olup, sınıflandırıcı şeklinde kullanılabilir. En basit kEK sınıflandırıcısı EK olup (kEK'de k=1 ise EK olarak adlandırılır) çalışma algoritması şu şekilde özetlenebilir: Her özellik vektörü ve o özellik vektörüne verilmiş etiketten oluşan eğitim vektörü ile verilen bir test vektörü arasındaki uzaklıklar hesaplanır. Enyakın komşuluk adını alan bu algoritmanın isminden anlaşılacağı üzere, hesaplanan bu uzaklıklara göre en yakın yani minimum uzaklıktaki eğitim vektörünün etiketi test vektörüne atanarak, test edilen vektörün hangi konuşmacıya ait olduğu belirlenmiş olur.

kEK (k>1) yöntemi, yeni bir vektörün hangi sınıfa ait olduğunun seçiminde kullanılır. Seçim işlemi, en yakın k tane uzaklıkların ya da komşulukların belirlenmesi üzerine kurulmuştur. Örneğin, verilmiş bir test vektörünün en yakın k komşulukları bulunmuş olsun. Daha sonra, bu k komşulukları arasındaki muhtemel en yakın komşuluk bulunarak, test vektörünün hangi sınıfa ait olduğunun seçimi yapılmış olur. Seçimi yapılan sınıfın etiketi ilgili test vektörüne atanır. Genellikle *k* tek sayı seçilir. Eğer *k* çift sayı seçilirse, hesaplanan uzaklıklar arasında aynı uzaklığa sahip iki ayrı sınıf belirlenebilir. Yani bir test vektörüne ait iki ayrı sınıf oluşmuş olur. Bu durumda sınıflar arasından seçim yapmak mümkün olmayacaktır.

Konuşmacı tanıma uygulamasında EK ya da kEK yöntemi kullanılarak, test vektörlerinin sınıflandırılması için ilk önce test vektörlerinin etiketleri bulunur. Her sınıfa ait uzaklık değerleri hesaplanır. EK yöntemi genellikle metinden bağımsız konuşmacı doğrulama ve saptama uygulamalarında kullanılır.

EK sınıflandırıcıları, bütün konuşma verisi arasından bir test vektörünün en yakın komşuluğunu bulabilmek için, bütün eğitim verisinin saklanması ve ayrıntılı bir karşılaştırma yapılması fazla hafıza alanı ve yoğun hesaplamalar gerektirdiğinden kullanım maliyeti çok yüksektir.

2.3.2. Vektör nicemleme (VN)

Vektör Nicemleme algoritmaları genelde, eğiticisiz öğrenme algoritmalarına girer. Bu nedenle, öğrenme süresince sınıfları ifade eden etiketler kullanılmaz. VN işlemi, eğitim verisini belirli sınıflara kendiliğinden sınıflandırmaktadır. K-ortalama (K-means) yöntemi, Vektör Nicemlemede kullanılan, popüler bir algoritmadır. Bu algoritmada K sınıf sayısını ifade eder. K-ortalama algoritması, VN yönteminde olduğu gibi kodlama işleminde de kullanılmaktadır. K-ortalama ve VN algoritmalarından; Linda-Buzo-Gray (LBG) ve Vektör Nicemleme Öğrenme Metodu gibi yöntemler kullanılmaktadır (Deller 1993).

VN algoritması sınıflandırıcı olarak şu şekilde kullanılabilir: İlk önce, $\{x_1,x_2,\ldots,x_L\}$ eğitim vektörlerinden oluşan özellik uzayı, $\{s_1,s_2,\ldots,s_N\}$ gibi bölgelere bölünür ($N<=L$). $1<= i <= N$ olup her s_i bölgesi; verinin farklı öbeklerini ifade eder. Her öbeğin merkezi (centroids) hesaplanır. Hesaplanan $\{c_1,c_2,\ldots,c_N\}$ merkezlerine kural tablosu denir. Bir test vektörünü sınıflandırmak için, test vektörü ile kural tablosundaki her merkez arasındaki uzaklık bulunur. Hesaplanan uzaklığa göre, kural tablosundaki minimum uzaklığa sahip vektör seçilerek, test vektörü ile eşleştirilir (Rabiner 1993).

Konuşmacı tanıma sistemlerine VN sınıflandırıcısı aşağıdaki gibi uygulanabilir: Bir konuşmacıya ait özellik vektörlerinden o konuşmacının kural tablosu hazırlanır. Bu işlem, konuşmacı setindeki tüm konuşmacılar için tekrarlanır. Konuşmacı saptama işlemi için, test edilecek telaffuza ait özellik vektörü, her konuşmacıya ait kural tablosuna uygulanır. Verilmiş bir kural tablosu için, test vektörüne en yakın merkez bulunur ve aynı kural tablosu için bu merkeze olan uzaklıklar toplanır. Her konuşmacıya ait kural tablosuna göre hesaplanan uzaklıklardan test vektörüne minimum uzaklıkta olan konuşmacı seçilir.

Konuşmacı doğrulama işlemi için, test vektörleri sadece doğrulanması istenen konuşmacı modeline uygulanır. Minimum uzaklıkların toplamları hesaplanarak test vektörleri sayısına bölünür. Elde edilen değer, bir eşik değeri ile karşılaştırılarak konuşmacının sisteme kabul edilip edilmeyeceğine karar verilir. Özellik vektörleri birbirine en çok benzeyen konuşmacılardan (eşkonuşmacılar-cohorts) elde edilmiş

göreceli bir değere göre uyarlanmış bir eşik değer de kullanılabilir. Bu yönteme eşkonuşmacı normalizasyonu (cohort normalization) denmektedir (Rosenberg 1992).

VN sınıflandırıcısı, konuşmacı tanıma sistemlerinde, çeşitli özellik çıkartım vektörleri ile kullanılmıştır. Başlıca özellik vektörleri Doğrusal Öngörülü Katsayılar ve Kepstral katsayılardır. Kepstral katsayıların VN ile oluşturduğu kombinasyonun, konuşmacı tanıma sistemlerinde, iyi sonuç verdiği söylenmiştir (Farrell 1994).

2.3.3. Ağaç yapılı vektör nicemleme

VN sınıflandırıcısı konuşma tanıma uygulamalarında kullanılan başarılı bir yöntemdir. Fakat, geniş bir konuşma seti için kural tablosu hazırlamak ve karşılaştırmaları yapmak sistemi hantallaştıracaktır. Bu sorunu hafifletmenin yolu, en yakın merkezlerin karşılaştırmasını organize edecek ağaç yapılı vektör nicemleme algoritmasının kullanılmasıdır.

Ağaç yapılı VN algoritması, en yakın merkeze ulaşmak için bir kol oluşturarak hem zamandan hem de hafızadan kazanç sağlar. Fakat bu avantaja rağmen algoritma, test vektörünü optimum sınıfla eşleştireceği kesin değildir.

2.3.4. Karar ağaçları

En çok istatistik alanında kullanılan karar ağaçları, bir karar mekanizmasının kurallarını ifade eder. En popüler olanları, C4, ID3, CART ve Bayes karar ağaçlarıdır. Bir karar ağaç yapısında, her ağacın dal oluşturmayan (nonterminal-nonleaf) her düğümü bir karar durumunu gösterir ve ağacın her yeşeren dalı ise bir sınıfa aittir. Yeşeren düğümler giriş verisinin özel bölümlerini ifade eder. Bir test vektörü, oluşturulacak bir karar çerçevesinde, her düğümde değerlendirilerek alt düğümlere yöneltilecektir. Karar verme yapısı, test vektörü nihai bir düğüme ulaşıncaya kadar devam eder. Sonuçta, bu test vektörüne, ilgili düğümün içerdiği sınıfa ait etiket atanmış olacaktır.

Karar ağaçlarının konuşmacı tanıma sistemine uygulanması için, ilk olarak tüm konuşmacılara ait özellik vektörlerinden eğitim verisi oluşturulur. Bir konuşmacıya ait konuşma verisinin özellik vektörlerine "bir", diğer konuşmacıların özellik

vektörlerine ise "sıfır" değeri etiketlenir. Etiketlenmiş eğitim verisine göre her konuşmacı için ikili karar ağacı eğitilir. İkili karar ağacının kolları her bir sınıfın etiketini gösterir. Bu etiketlerde "bir" değeri ilgili konuşmacıya ait sınıfları, "sıfır" değeri ise o konuşmacıya ait olmayan "antikonuşmacı" sınıfları içerecektir. Karar ağacının kolları, ait oldukları sınıf etiketleriyle birlikte bu etiketlerin olasılıklarını içerir. Konuşmacı saptama işleminde, test edilecek telaffuzlara ait bütün özellik vektörleri her karar ağacına uygulanır. Her konuşmacının karar ağaç olasılıkları, saptanması istenen konuşmacının belirlenmesinde kullanılır.

2.3.5. Konuşmacı veritabanları

Değişik amaçlar için hazırlanmış konuşmacı veritabanları, bir çok araştırmacı tarafından kullanılmaktadır. En çok kullanılan veritabanlarından dokuz tanesi aşağıda sıralanmıştır (Campbell 1999).

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

NIST: Telefon santrali kayıtlarından elde edilmiş konuşmacı tanıma veritabanı.

OGI: Oregon Graduate Institute-OGI bir çok dilde gerçekleştirilmiş 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 kayıt edilmiştir (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 tekrarlamış olduğu 10 konuşma verisini içerir. Ofis ortamında 3.8 KHz örnekleme frekansında kayıt edilen rakamların telaffuzu şeklindeki (pin numarası gibi) konuşma verisi Linguistic Data Consortium (LDC) lisansı altında kullanıcıların hizmetine sunulmaktadır.

http://www.ldc.upenn.edu/Catalog/readme_files/yoho.readme.html

KING (LDC): 1992 yılında LDC tarafından, 51 bay konuşmacıdan telefon hattı üzerinden değişik mikrofon 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.

ANDOSL : Avustralya Ulusal Konuşma Dili Veritabanı.

http://andosl.anu.edu.au/andosl/general_info

Bu tez çalışmasında kullanacağımız Türkçe veritabanı 7 Bay ve 3 Bayan konuşmacının ad ve soyadlarını 8 kez tekrarladıkları metne bağlı konuşmacı tanıma veritabanıdır. Konuşma verilerinin 5 tanesi eğitim, 3 tanesi de sınıflandırıcının test aşamasında kullanılmıştır. Yapılan araştırmalara göre halen Türk dilinde hazırlanmış bir konuşmacı veritabanı bulunmamaktadır. Bu çalışmalar sonunda, yapacağımız uygulamalar için kullanılacak bu konuşmacı kayıtları ilerde bir veritabanı oluşturmada kullanılabilir.

Tez çalışmasında kullanılan bir diğer veritabanı ise metinden bağımsız TIMIT veritabanıdır. New England bölgesinden 14 bayan ve 24 bay konuşmacıdan alınan, örnekleme frekansı 16 KHz ve tek kanallı WAV uzantılı dosyalar şeklinde kayıt edilmiş, konuşma verileri kullanılmıştır.

BÖLÜM 3. YSA SINIFLANDIRICILARI

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 sinapslerden oluşur. Bu biyolojik sinir hücreleri (işlem birimleri) ve sinapsler (bağlantılar) bilgisayar ortamında geliştirilen algoritmalar sayesinde bir benzetimin ürünü olarak geliştirilmiştir. Bu yüzden de yapay adını almıştır. Bağlantılar taşıdığı işareti bir işlem birimden diğerine iletirler. 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 bir çıkış üretebilir, aksi halde çıkış üretemez.

Bölüm 2.3'te anlatılan klasik sınıflandırıcılara alternatif olarak Çok Katmanlı Almaç (ÇKA) (Multilayer Perceptron), Kendi Kendini Organize Eden Ağ Modeli (Self Organizing Map - SOM), Zaman Gecikmeli YSA, Radyal Tabanlı Fonksiyon Ağı, Vektör Nicemleme Öğrenme Metodu (VNÖM), Öngörülü YSA Modeli, Yapay Ağaç Ağ Modeli (YAM) ve diğer YSA modelleri sayılabilir (Morgan 1991).

Bu bölümde, YSA'ların öğrenme modları, öğrenme kuralları, konuşma tanıma alanında en çok kullanılan YSA sınıflandırıcıları, bir alt bölümde tez çalışmasında SOM ağıyla birlikte kullanılması amaçlanan Birleştirilmiş Bellek Modeli (BBM) ağı (Associative Memory Model) açıklanacaktır. Ayrıca, bir başka alt bölümde de Türkçe ve diğer dillerde, konuşmacı tanıma alanında yapılmış çalışmalardan söz edilecektir.

YSA'lar genellikle katmanlardan oluşur, Şekil 3.1'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 birimleriyle olan bağlantısı yasaklanmış olabilir (İnal 1996).

Bir YSA eğitim sonunda, ağın 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.

3.1. Öğrenme Modları

İki çeşit öğrenme modu vardır: eğiticili (supervised) ve eğiticisiz (unsupervised) öğrenme modları. Eğiticili öğrenme modu; adından da anlaşılacağı gibi öğretici gerektirir. Bunun aksine eğiticisiz öğrenme de ise ağın kendisi bir öğretici gibi kendi yaklaşımlarını kendisi oluşturur.

Eğiticili 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.

Eğiticili 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 eğiticili öğrenme algoritması işlem birimleri arasında işbirliği yapmalıdır. Böyle bir tasarıda, 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 konusudur.

Eğiticisiz öğrenme modunda, ağın ağırlıkları dışardan etkiler kullanılarak ayarlanmaz. Bunun yerine, ağın performansını kendi kendine izlemesi söz

konusudur. Ağ, giriş işaretine ve ağın önceden belirtilmiş fonksiyonuna göre yenileme işlemini yapar. Ağa doğru ya da yanlış olduğu söylenmemiş olsa bile, ağ kendini nasıl organize edeceği hakkında yine de bir miktar bilgiye sahip olmalıdır. Yani, ağ kendi çağrışımını kendisi yaratır.

Eğiticisiz öğrenme modunda, işlem birimleri arasındaki yarışma (rekabet), 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. Örneğin, işlem birimleri, yatay veya düşey kenarlar ya da sağ ve sol kenarlar gibi çeşitli örüntü özellikleri arasındaki ayırımı yapmak için organize olabilirler.

3.2. Öğrenme Kuralları

Bu bölümde tez çalışmasında kullanılacak öğrenme kuralları açıklanmıştır.

3.2.1. En küçük kareler yöntemi

Bir işlem biriminin istenen çıkış değeri ve hedef değer arasındaki farkı (delta) minimuma indirgemek için bu birimler arasındaki bağlantıların kuvvetliliğinin, sürekli olarak sağlanması fikri üzerine kurulu bir kuraldır. Verilmiş $\{x_1,x_2,...,x_L\}$ 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...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.

3.2.1.1. 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:

$$\varepsilon_k = d_k - y_k \tag{3.1}$$

Karesel hata veya beklenen hata değeri şu şekilde tanımlanır:

$$\langle \varepsilon_k^2 \rangle = \frac{1}{L} \sum_{k=1}^{L} \varepsilon_k^2 \tag{3.2}$$

Eşitlik 3.2'de L; eğitim setindeki giriş vektörlerinin sayısıdır. Eşitlik 3.1 kullanılarak karesel hata ifadesi aşağıdaki gibi genişletilebilir:

$$\langle \varepsilon_k^2 \rangle = \langle (d_k - W^t . X_k)^2 \rangle \tag{3.3}$$

Şimdi de Eşitlik 3.3'teki ifadenin parantez karesini elemanlara dağıtırsak, aşağıdaki ifade elde edilmiş olur:

$$\langle \varepsilon_k^2 \rangle = \langle d_k^2 \rangle + W^t \langle X_k X_k^t \rangle W - 2 \langle d_k X_k^t \rangle W \tag{3.4}$$

$R=\langle X_k X_k^t \rangle$ 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 3.4 aşağıdaki şekilde tekrar yazılabilir:

$$\xi = \langle d^2 \rangle + W^t R W - 2 p^t W \tag{3.5}$$

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:

$$\frac{\partial \xi(W)}{\partial W} = 2RW - 2p \tag{3.6}$$

$$2RW^* - 2p = 0 \Rightarrow RW^* = p \tag{3.7}$$

$$W^*=R^{-1}p \tag{3.8}$$

W^* ifadesi ağırlıkların çözüm kümesini göstermektedir.

ξ skaler olmasına rağmen Eşitlik 3.6'daki ξ'nin kısmi türev ifadesi ise bir vektördür: Eşitlik 3.6, ξ 'nin gradientidir, yani $\nabla\xi$ bir vektördür:

$$\nabla\xi=\left[\frac{\partial\xi}{\partial W_1},\frac{\partial\xi}{\partial W_2},\frac{\partial\xi}{\partial W_3},\ldots\frac{\partial\xi}{\partial W_n}\right]^t \tag{3.9}$$

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

3.2.1.2. 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 global bir minimuma ulaşıncaya 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:

$$W(t+1)=W(t)+\Delta W(t) \tag{3.10}$$

Eşitlik 3.10'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:

$$W(t+1)=W(t)-\mu\nabla\xi(W(t)) \tag{3.11}$$

Her bir iterasyonda $\nabla\xi(W(t))$ değerinin belirlenmesi gerekir. Eşitlik 3.6 ve 3.9, $\nabla\xi(W(t))$'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. $\varepsilon_k^2(t)$, hatanın karesi belirlenir, o anki ağırlık vektörünün kullanılmasıyla aşağıdaki eşitlik elde edilir:

$$\varepsilon_k^2(t)=(d_k-W^t(t)X_k)^2 \tag{3.12}$$

3. $\varepsilon_k^2(t)$'nin yani hatanın gradientini hesaplamak için Eşitlik 3.3'teki gibi beklenen hata değerinin ($<\varepsilon_k^2>$) kullanılmasıyla aşağıdaki eşitlikler elde edilir:

$$\nabla\varepsilon_k^2(t)\approx\nabla<\varepsilon_k^2> \tag{3.13}$$

$$\nabla\varepsilon_k^2(t)=-2\varepsilon_k(t)X_k \tag{3.14}$$

4. Eşitlik 3.14'teki gradient formülünü kullanarak Eşitlik 3.11'e göre ağırlık vektörü aşağıdaki gibi yenilenir:

$$W(t+1)=W(t)+2\mu\varepsilon_k X_k \tag{3.15}$$

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 3.15, 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 parametresinin bu kararsız durumu ortadan kaldırır.

3.2.2. 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 ağa 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 ağa 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 yansıtmak 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 ağın eğitimi belirli kriterlere göre eğitildikten sonra kullanılır. Eğitim sırasında hatanın gradient ifadesi iki nokta arasına taklıp, osilasyona girip lokal bir minimaya saplanıp kalabilir.

3.2.3. Yarışmacı öğrenme kuralı

Bu kural sadece eğiticisiz 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:

$$y_k = \begin{cases} 1 & \text{eğer } v_k > v_j \text{ ve } j \neq k \\ 0 & \text{aksi halde} \end{cases} \quad (3.16)$$

Eşitlik 3.16 '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:

$$\sum w_{kj} = 1 \quad (3.17)$$

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

$$\Delta w_{kj} = \begin{cases} \eta(X_j - w_{kj}(n)) \\ 0 \end{cases} \quad (3.18)$$

Eşitlik 3.18'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:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau}\right), \quad n = 0,1,2,... \quad (3.19)$$

Eşitlik 3.19'da τ zaman sabitidir. Eşitlik 3.18 ve 3.19 kullanılarak bir sonraki ayrık zaman için yeni ağırlıklar aşağıdaki gibi hesaplanır:

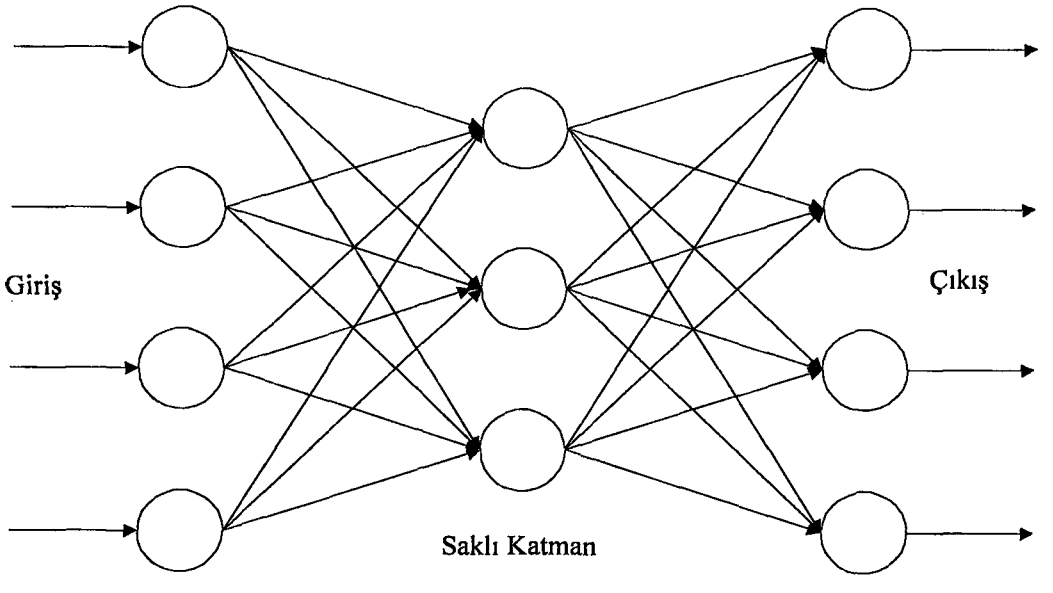
$$w_{kj}(n+1) = w_{kj}(n) + \eta(X - w_{kj}(n)) \quad (3.20)$$

3.3. YSA Sınıflandırıcı Modelleri

İstatistiksel örüntü tanıma tabanlı sınıflandırıcıların, öbekleme (clustering) işlemine alternatif bir yöntem ise eğiticili öğrenme algoritmasıdır. Eğiticili sınıflandırıcının en popüler olanı YSA'lardır. Son yıllarda yapılan çalışmalar YSA'ların, konuşma işleme alanında, önemli rol üstlendiğini göstermektedir. Aşağıdaki alt bölümlerde tez çalışmasında ve konuşmacı tanıma alanında en yaygın kullanılan sınıflandırıcılar anlatılmıştır.

3.3.1. Çok katmanlı almaç-ÇKA

Genellikle standart geri yansıtma eğitim algoritmasının kullanıldığı tipik bir Çok Katmanlı Almaç yapısı Şekil 3.1'de görülmektedir. Geri yansıtma 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 hata değeri, daha sonra çıkış katmanından bir önceki katmandaki işlem birimlerinin ağırlıklarını yenilemek üzere geriye yansıtılır. Yansıtma işlemi giriş katmanına ulaşıncaya kadar tekrarlanır. Yukarıdaki bütün bu işlemler, eğitim setindeki örüntülerin tümü için tekrar edilir.

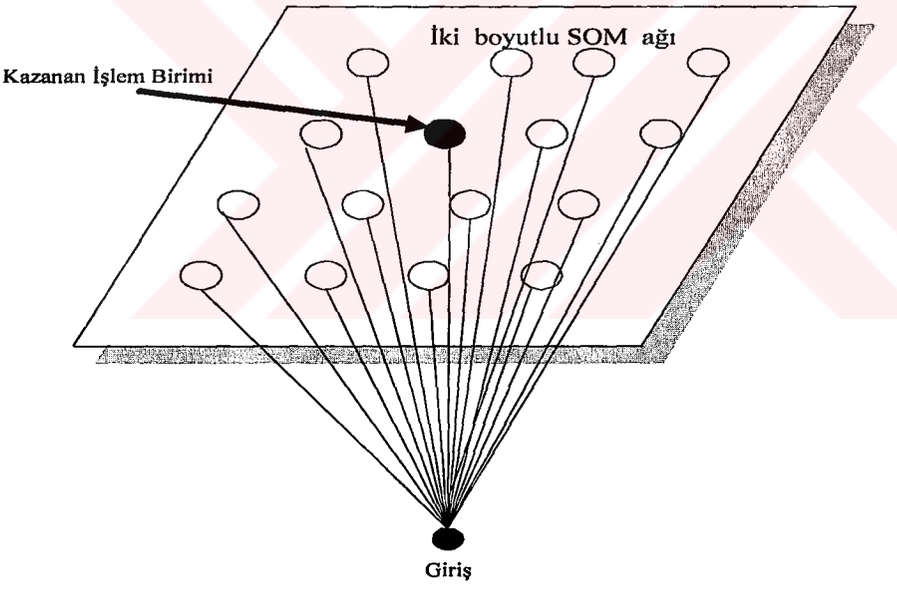


Şekil 3.1. Birbirine tam bağlı üç katmanlı bir ÇKA modeli

Şekil 3.1 'deki gibi bir ÇKA, sınırlı giriş değerlerine sahip problemlerin çözümü için uygun bir modeldir. Fakat, uzun süreli eğitim gerektirmesi, algoritmanın bir optimum genel çözüm noktası yerine bir yerel minimum noktaya saplanıp kalması da olumsuz taraflarıdır.

3.3.2. Kendi kendini organize eden ağ modeli (Self Organizing Map - SOM)

Kohonen özellik haritası olarak da bilinen SOM; 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 yarışı, en yüksek skorlu çıkışı üreten işlem biriminin kazandığı öğrenme işlemi; yarışmacı öğrenme algoritması olarak özetlenebilir. Şekil 3.2'de temel SOM yapısı görülmektedir.



Şekil 3.2. Kohonen Ağı (SOM)

Şekil 3.2'ye göre SOM ağına uygulanan bir girişe göre; yarışı 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 eğitimini tamamlamışsa, ilgili konuşmacı

için belirli çıkışları etkin olacaktır. Bu çıkışlara göre hangi konuşmacının seçileceğine karar verecek bir yapı olmalıdır. Bu nedenle karar birimi olarak birleştirilmiş bellek modeli YSA mimarisinin, SOM ağının çıkışında kullanılması amaçlanmıştır.

3.3.3. Yapay ağaç ağ modeli – YAM

YAM modeli, karar ağaç yapısı ile ileri beslemeli ağ modeli özelliklerinin birleştirilmesinden 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 altkümelere böler, böylece her dalın düğümü tek bir sınıfı ifade eder.

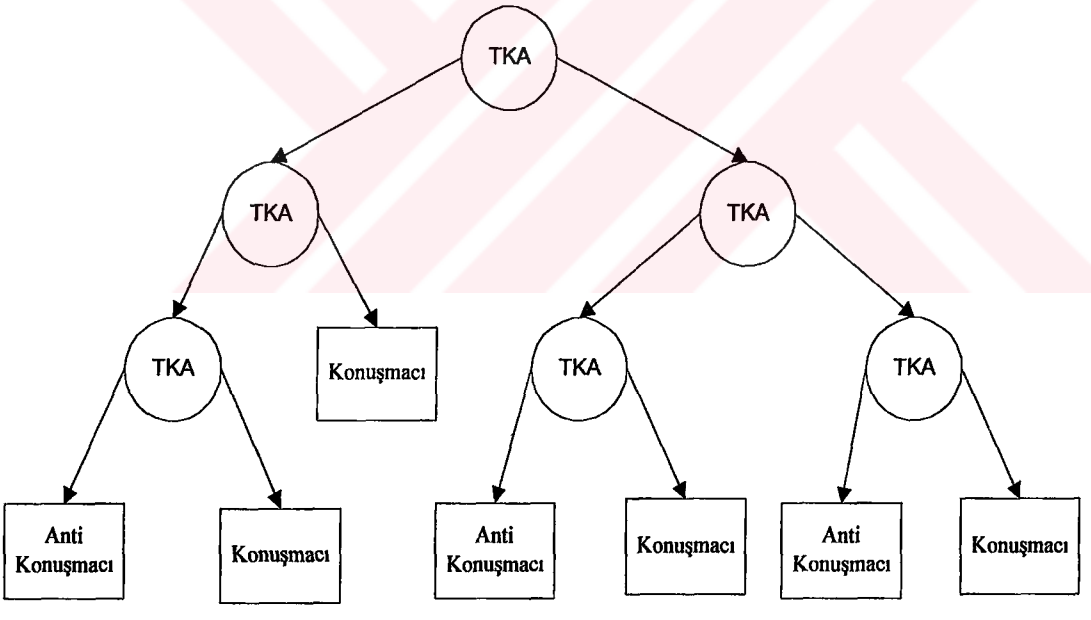
YAM yinelemeli eğitimi aşağıdaki gibidir: Bir düğümdeki eğitim verisi seti için, eğer verilen veri bu düğümdeki sınıfa ait ise ilgili düğüm başka kol ya da kollar oluşturmaz. Aksi durumda, eğitim verisi, bu düğümün çocuklarını oluşturacak, bir kaç altkümeye bölünür. Bu işlem tüm eğitim verisi için tekrarlanır. 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 optimum 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 konuşmacının telaffuzlarından üretilir ve önceki bölümlerde açıklandığı gibi etiketlendirilerek, eğitim seti oluşturulur. YAM'a uygulanan eğitim verisinin tümü tamamen sınıflandırılıncaya kadar, yinelemeli bir şekilde hem öğrenmeyi hem de mimarisini geliştirir. Tipik bir ikili yapay ağaç modeli Şekil 3.3'de görülmektedir.

Şekil 3.3'te "Konuşmacı" etiketli kollar, "bir" değerine sahip sınıfları, "Anti Konuşmacı" etiketli kollar ise, "sıfır" değerine sahip sınıfları göstermektedir.

Şekil 3.3'e göre, bir sınıfa ait eğitim verisinin aynı sınıfı ifade eden kola karşılık gelmesi gerekmemektedir, muhtemelen bir kaç 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 nitelendirecektir. Konuşmacı doğrulama için, tanınması istenen konuşmacıya ait modele bütün özellik vektörleri uygulanır. Her vektör, bu konuşmacıya ait olup olmama durumuna göre etiketlenir. Eğer test vektörlerinin bu konuşmacıya ait olma yüzdeleri, önceden saptanan bir eşik değerini aşıyorsa konuşmacı doğrulanmış olacaktır.



Şekil 3.3. Yapay Ağaç Ağ Modeli –YAM

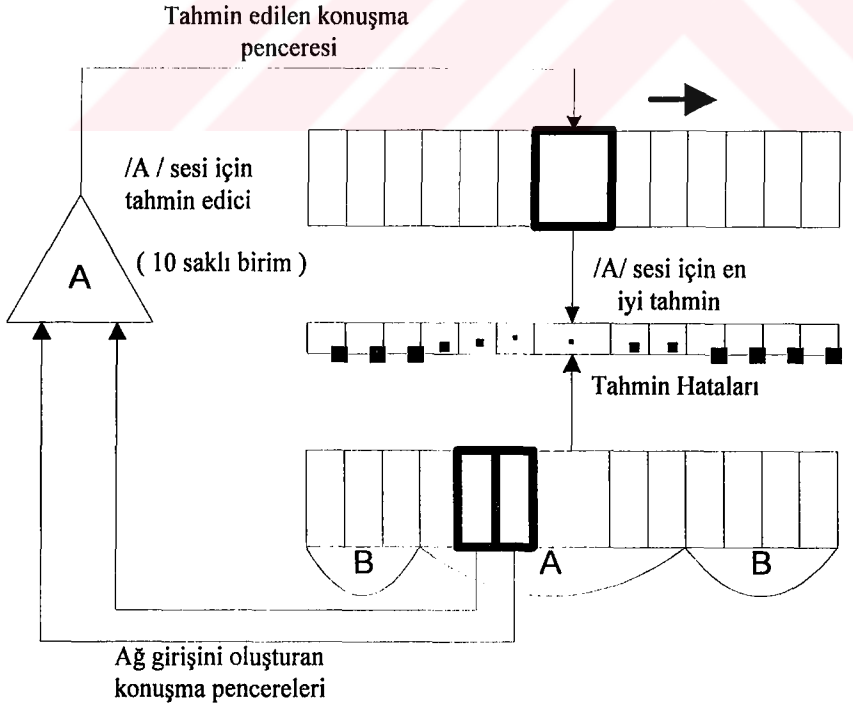
3.3.4. Öngörülü YSA modeli

Konuşmanın belli başlı pencereleri, Öngörülü YSA modelinin girişlerini oluşturur. Modelin çıkışları ise, bir sonraki pencerenin tahminini içerir. Her sese ait bir model kurularak, birden fazla model kullanma yoluyla, bu modellerin tahmin hatası karşılaştırılarak; en küçük tahmin hatasına sahip model bu konuşmanın ilgili bölütü

için en iyi eşleştirme olarak seçilebilir. Bu durum sınıflandırma ağında tam tersinedir: ağ, girişine uygulanan konuşma bölütünü, çıkışta verilen sınıflardan birine sınıflandırır.

Tebelskis tez çalışmasında, "bağlantılı öngörülü YSA" – Linked Predictive Neural Net (LPNN) adını verdiği mimaride, akustik (ses dağılım biçimi) modeller gibi tasarlanmış, öngörülü ağlar kullanmıştır (Tebelskis 1995). Şekil 3.4'te tahmin etme yoluyla ses tanıma işlemi yapan öngörülü ağ modeli görülmektedir.

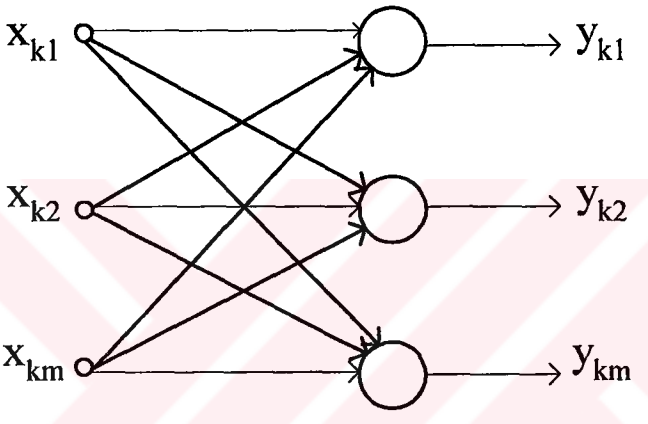
LPNN ağı, Şekil 3.4'te gösterildiği gibi, tahmin etme yoluyla ses tanıma işlemini yerine getirir. Üçgen şeklinde gösterilen ağ, sürekli konuşmanın K tane penceresini (K en az 2 seçilmelidir) saklı katmandaki birimlerden geçirme yoluyla, konuşmanın bir sonraki penceresini tahmin etmeye çalışır. Tahmin edilen pencere daha sonra, gerçek pencere değeri ile karşılaştırılır. Eğer, hata yeterince küçük ise, konuşmanın bu bölütü için ağın iyi bir model oluşturduğu söylenebilir. Bir dildeki çeşitli sesler için her sese ait bir tanıyıcı LPNN modeli tasarlanabilir. Bu düşünceden yola çıkarak, tanıtılacak kelimeler de aynı yöntemle modellenebilir.



Şekil 3.4. Öngörülü ağın temel işlem şekli

3.4. Birleştirilmiş Bellek Modeli (BBM)

BBM ağı, doğrusal işlem birimlerinden tek katmanlı bir yapıya sahip olup, ağın giriş ve çıkış vektörleri arasında doğrusal bir ilişki kurar. Şekil 3.5'te BBM ağının yapısı görülmektedir. Şekildeki x_k giriş, y_k çıkış vektörlerini ifade etmektedir. Ağın girişine uygulanan vektörlerin istenen çıkış değerlerine göre oluşturacağı ağırlıkların bir kez hesaplanması sonucunda ağın eğitimi sağlanmış olur. Test aşamasında da ağın girişine uygulanan vektör ile ağırlıkların skaler çarpımından elde edilen değere göre hangi çıkışın etkin olacağı belirlenmiş olur.



Şekil 3.5. Birleştirilmiş Bellek Modeli Yapısı

Şekil 3.5'e göre giriş ve çıkış vektörlerinin boyutunu gösteren *m* değeri aynı zamanda ağın işlem birimi sayısını göstermektedir. x_k, y_k vektör çiftleri arasında ilişki sağlayacak eşitlik aşağıdaki gibi ifade edilebilir:

$$y_k = W(k)\, x_k, \qquad k = 1, 2, \ldots, q \tag{3.21}$$

Eşitlik 3.21'de W(k); (x_k, y_k) çiftlerinden elde edilen ağırlık matrisini göstermektedir. Aynı ifade *i.* işlem birimi için ifade edilirse, Eşitlik 3.21 başka bir biçimde yeniden tanımlanmış olur:

$$y_{ki} = \sum_{j=1}^{m} w_{ij}(k) x_{kj}, \quad i = 1, 2, \ldots, m \tag{3.22}$$

Eşitlik 3.22'de $w_{ij}(k)$, j=1,2,...,m için, i. işlem biriminin k. örüntü çiftlerine göre oluşturduğu bağlantıların ağırlıklarını ifade etmektedir. Matris notasyonunu

kullanarak Eşitlik 3.22 aşağıdaki gibi tekrar yazılabilir:

$$y_{ki} = [w_{i1}(k),\ w_{i2}(k),\ .\ .\ .,\ w_{im}(k)] \begin{bmatrix} x_{k1} \\ x_{k2} \\ \cdot \\ \cdot \\ \cdot \\ x_{km} \end{bmatrix}, i = 1,2,\ldots,m \qquad (3.23)$$

Eşitlik 3.23'ün sağındaki sütun vektörü x_k gibi diğer ifadelerin de açık biçimi yazılırsa aşağıdaki eşitlik elde edilir:

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{11}(k) & w_{12}(k) & \cdots & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & \cdots & w_{2m}(k) \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1}(k) & w_{m2}(k) & \cdots & w_{mm}(k) \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \qquad (3.24)$$

Eşitlik 3.24; Eşitlik 3.21'de tanımlanmış ifadenin matris dönüşümü yapılmış şeklidir. Ağırlık matrisinin *m x m* boyutunda olmasının nedeni giriş ve çıkış vektörlerinin aynı boyutta olmasından kaynaklanmaktadır. Bu matris dönüşümünü kullanarak, eğitim aşamasında ağın ağırlık vektörleri aşağıdaki eşitlikte olduğu gibi kapalı biçimde ifade edilebilir:

$$W = Y X^T \qquad (3.25)$$

Eşitlik 3.25'te X^T ifadesi giriş vektörlerinin devriğini ifade etmektedir. Eşitlik 3.25 kullanılarak elde edilen ağırlık vektörleri için BBM modelinin test amacıyla girişine uygulanacak vektörlerin sınıflandırılması için kullanılabilir. Bu amaçla Eşitlik 3.25'te, eşitliğin her iki tarafı X vektörü ile çarpılırsa, aşağıdaki ifade elde edilebilir:

$$W X = \hat{Y} X^T X \qquad (3.26)$$

Eşitlik 3.26'da $\hat{Y}$ test aşamasında W ve X ifadelerine göre hesaplanacak çıkış değerini göstermektedir. Eşitliğin sağ tarafında X^TX ifadesi, I birim matrisini

verdiğine göre, bu eşitlik yeniden düzenlenirse tahmini yapılacak Y çıkışı aşağıdaki eşitliğe göre bulunabilir:

$$\hat{Y} = W X \tag{3.27}$$

BBM ağı gürültüsüz giriş vektörleriyle elde edilmiş ağırlıklar için oldukça doğru çıkışlar üretmektedir. Bu nedenle ağın eğitiminde kullanılacak giriş değerleri itinalı bir şekilde seçilmelidir.

3.5. Konuşmacı Tanıma Alanında Yapılmış Çalışmalar

Bu bölümde, yapılan araştırmalar doğrultusunda, gerçekleştirilmiş YSA tabanlı konuşmacı tanıma çalışmalarına ait tez ve makalelerin sonuçları, çeşitli yönleriyle incelenecektir.

Paoloni v.d. çalışmalarında iki Öngörülü YSA modelini kullanmışlardır (Paoloni 1996). Bir tanesi tüm konuşmacı seti için eğitilen "Global Öngörülü YSA" şeklinde adlandırılan model, diğeri ise her konuşmacı için eğitilen modeldir. Her konuşmacıya ait modelin tahmin edilememe etkisi, her konuşmacıya ait tahmine dayalı YSA modelinin, tüm konuşmacılara ait global modele normalize edilmesi sonucunda ortadan kaldırılabilir. Global modelin kullanımı ile 3 dakikalık konuşma verisinin eğitimi süresince Eşit Hata Oranı (EHO) % 26.5'ten % 3.7'ye düşürülmüştür. Sistemin test aşamasında 20 saniyelik konuşma verisi için EHO % 0.55'ler seviyesindedir. Her konuşmacı için farklı hatalı reddetme (HR) ve hatalı kabul_etme (HK) kriterleri olacağı düşünülerek her biri için değişik eşik değerleri tanımlanmıştır. Her konuşmacı için değişik eşik değerleri tanımlanmadığında, 20 saniyelik veri için EHO % 4.5 oranında gerçekleşmiştir.

Kitamura ve Takei çalışmasında iki model kullanan bir sistem geliştirmiştir (Kitamura 1996). Bu ağlardan ilki, Statik Özellik Modelidir. Bu model, konuşmanın karakteristik özelliklerini içeren Vektör Nicemleme özellik haritasının betimlendiği SOM Ağı Modelidir. Bir diğer model ise; Dinamik Özellik Modelidir. Bu model, 3 katmanlı bir Öngörülü YSA modelini içermektedir. Dinamik özellik modelinin çıkışı,

her konuşmacıya ait örüntüleri öğrenen, Öngörülü YSA modeline giriş olarak uygulanmıştır. Test süresince, tek boyutlu mel-kepstral katsayıları statik özellik modeline, iki boyutlu mel-kepstral katsayıları ise dinamik özellik modeline uygulanmıştır. Statik model çıkışı; girişine uygulanan konuşma vektörü ile bu konuşmanın özellik vektörü arasındaki vektör nicemleme değişimini (E_{vn}) verir. Dinamik modelin giriş olarak uygulandığı Öngörülü YSA çıkışı da o anki konuşma değeri ile tahmini yapılan değer arasındaki tahmin hatasını (E_{tahmin}) verir. Sonuçta bu iki hatanın birleşiminden oluşan bir hata değeri (E) hesaplanır. Konuşma süresi 0.68 saniye olan bir konuşma verisi için sistemin tanıma oranı % 99'larda olup, bu denli yüksek oran iki boyutlu mel-kepstral katsayıları sayesinde sağlanmıştır.

Farrell v.d. çalışmalarında DARPA (Defense Advanced Research Projects Agency) TIMIT (TI-Texas Instruments MIT-Massachusetts Institute of Technology) veritabanını kullanarak değişik çalışmalar gerçekleştirmişlerdir (Farrell 1994). Bunlardan ilki, 5, 10 ve 20 konuşmacı için kapalı set konuşmacı saptama uygulamasıdır. Bu uygulamada, Vektör Nicemleme ve Ağaç Yapılı Vektör Nicemleme sınıflandırıcıları kullanılmıştır. Her iki sınıflandırıcı için kural tablosunun büyüklüğü 16'dan 128'e kadar değiştirilmiştir. Bu sınıflandırıcıların sonuçları Tablo 3.1 ve 3.2'de gösterilmiştir.

Tablo 3.1. Vektör Nicemleme Sınıflandırıcısının Performansı

Kural Tablosu Büyüklüğü	5 Konuşmacı	10 Konuşmacı	20 Konuşmacı
16	%100	%98	%90
32	%100	%98	%92
64	%100	%98	%95
128	%100	%98	%96

Tablo 3.2. Ağaç Yapılı Vektör Nicemleme Sınıflandırıcısının Performansı

Kural Tablosu Büyüklüğü	5 Konuşmacı	10 Konuşmacı	20 Konuşmacı
16	%96	%92	%83
32	%100	%92	%90
64	%100	%96	%90
128	%100	%94	%88

Bir diğer sınıflandırıcı ise kEK (k En yakın Komşuluk) yöntemidir. Çalışmada k komşuluğu [1..7] olarak değiştirilmiştir. Sonuçlar Tablo 3.3'te gösterilmiştir. Tablo 3.3'e göre en yakın komşuluk değerinin artması sistem performansını arttırmamaktadır. Buna göre, k değerinin nasıl seçileceği hakkında bir sonuca varılamamakta olup, genel anlamda geniş konuşmacı seti için k değerinin de büyük seçilmesi gerektiği fikri savunulmaktadır.

Tablo 3.3. kEK Sınıflandırıcısının Performansı

En Yakın Komşuluk	5 Konuşmacı	10 Konuşmacı	20 Konuşmacı
1	%96	%96	%85
3	%84	%90	%85
5	%96	%96	%89
7	%100	%96	%92

Kapalı set konuşmacı doğrulama için bir diğer sınıflandırıcı ise ÇKA modelidir. Geri yansıtma algoritmasıyla eğitilen ÇKA modelinde, tek saklı katmanda sırasıyla 16, 32 ve 64 işlem birimi kullanılmıştır. Çalışmada, birden fazla saklı katmanın kullanılması, sistem performansını artırmadığı görülmüştür. Çalışmanın sonuçları Tablo 3.4'te gösterilmiştir. Saklı katman işlem birimi sayısını arttırmanın, sistem performansını artırmadığı görülmektedir. Bunun nedeni; işlem birimi sayısının arttırılması, özelik vektörleri sınıflarının daha fazla alt sınıflara bölünmesine neden olacaktır. Bu durum ağın yerel bir minimuma saplanıp kalmasıyla sonuçlanabilir.

Tablo 3.4. ÇKA Sınıflandırıcısının Performansı

Saklı Katman İşlem Birimi Sayısı	5 Konuşmacı	10 Konuşmacı	20 Konuşmacı
16	%96	%90	%90
32	%96	%90	%82
64	%88	%94	%85

Farrell'in yaptığı bir diğer çalışma karar ağaçlarıdır. Bu çalışmada C4, ID3, CART ve Bayes karar ağaçlarını kullanmıştır. Bu çalışmanın sonuçları Tablo 3.5'te görülmektedir. Bu yöntemler içinde en iyi sonucu Bayes algoritması vermiştir. Bilgisayar belleğinin yetersiz kalmasından dolayı 20 konuşmacı için CART algoritması sonuç üretememiş ve tablonun ilgili alanına * sembolü konulmuştur.

Tablo 3.5. Karar Ağaçları Sınıflandırıcısının Performansı

Karar Ağacı Algoritması	5 Konuşmacı	10 Konuşmacı	20 Konuşmacı
ID3	%88	%88	%79
C4	%92	%84	%73
CART	%80	%76	*
BAYES	%92	%92	%83

Farell, kapalı set konuşmacı saptama uygulaması için son olarak yapay ağaç modelini kullanmıştır. Ayrıca bu çalışmada algoritma üzerinde değişiklikler yaparak her iki modelin de performansları değişik budama (pruning) seviyeleri için karşılaştırmıştır. Değiştirilmiş yapay ağaç modelinin, orijinal yapay ağaç modelinden temel farkı; sınıf etiketlerine ek olarak her budak için bir güven değeri (confidence measure) kullanılmasıdır (güven değeri, giriş vektörlerinin bir konuşmacıya ait olma yüzdesine denir). Örneğin, x özellik vektörlerinin, j. konuşmacının modeli Sj'ye uygulanmasıyla elde edilecek konuşmacı olasılığı aşağıdaki gibi hesaplanabilir:

$$P_{YAM}(x|S_j) = \frac{M}{N+M} \tag{3.28}$$

Eşitlik 3.28'de N antikonuşmacı vektör sayısını M ise konuşmacı vektör sayısını belirtmektedir. Aynı olasılık dağılımı Değiştirilmiş Yapay Ağ Modeli (DYAM) için aşağıdaki gibi hesaplanmıştır:

$$P_{DYAM}(x|S_j) = \frac{\sum_{i=1}^{M} c_i^1}{\sum_{j=1}^{N} c_j^0 + \sum_{i=1}^{M} c_i^1} \tag{3.29}$$

Eşitlik 3.29'da c^1 ve c^0 sırasıyla konuşmacı ve antikonuşmacı güven değerlerini göstermektedir. Güven değerlerinin eklenmesi, Budanmış Yapay Ağaç Modeli için önemli iyileştirmeler sağlamaktadır. Her iki model değişik budama seviyeleri için değerlendirilmiştir. Bu sonuçlar Tablo 3.6'da verilmiştir.

Tablo 3.6. YAM ve DYAM Sınıflandırıcısının Performansı

Budama Seviyesi (yam/dyam)	5 Konuşmacı (yam/dyam)	10 Konuşmacı (yam/dyam)	20 Konuşmacı (yam/dyam)
4	%80 / %92	%66 / %88	%67 / %75
5	%88 / %96	%84 / %90	%76 / %87
6	%96 / %100	%92 / %94	%82 / %93
7	%92 / %96	%92 / %98	%91 / %96
8	%92 / %92	%90 / %96	%89 / %94
budamasız	%92 / %92	%90 / %90	%89 / %89

Tablo 3.6'da DYAM'ın YAM'a göre daha iyi sonuç verdiği ve budama işleminin her durumda sistem performansını arttırdığı görülmüştür. Fakat farklı konuşmacı sayısı için en iyi sonuçlar farklı budama seviyelerinde elde edilmiştir. Buradan çıkarılacak

sonuç; konuşmacı sayısının artmasıyla, budama seviyelerinin arttırılmasının doğru olacağıdır.

Kapalı set konuşmacı saptama uygulamasına ilişkin en iyi performansa sahip yöntemler, farklı sınıflandırıcılar için karşılaştırılacak olursa, DYAM modelinin 10 konuşmacı için uygun budama seviyesinde (seviye 7) Vektör Nicemleme yöntemiyle aynı performansı sağladığı ve Ağaç Yapılı Vektör Nicemleme Yönteminden daha iyi sonuç verdiği görülmektedir. Buradan çıkarılacak genel sonuç, DYAM modeli ve Vektör Nicemleme yöntemlerinin kapalı set konuşmacı saptama uygulamaları için en uygun yöntemler olduğu söylenebilir.

Yine Farell'in çalışmasında, VN ve DYAM yöntemleri konuşmacı doğrulama sistemine uygulanmıştır. İlk uygulamada sistem tarafından bilinen 10 konuşmacı ve sistem tarafından bilinmeyen (eğitim aşamasında kullanılmamış) sahte olarak nitelendireceğimiz 10 konuşmacı kullanılmıştır. Her küme içindeki konuşmacılar; 5 bayan 5 de bay konuşmacıdan oluşmaktadır. Her DYAM modelinde 7. budama seviyesi kullanılarak 10 konuşmacı eğitilmiştir.

DYAM modelinin performansı, model tarafından eğitimde kullanılan ve doğru şekilde sınıflandırılmış konuşmacıların oranının, sınıflandırılmayanlara oranlanmasıyla bulunmuştur. Kural tablosu büyüklüğü 128 olan bir VN sınıflandırıcısı için modelin performansı: özellik vektörlerinin toplam uzaklık ölçümlerinin, test vektörleri sayısına oranlanmasıyla ölçülmüştür.

Eşkonuşmacı normalizasyonu olarak bilinen bir teknik kullanılarak VN sınıflandırıcısının performansı iyileştirilebilir. Konuşmacı doğrulama sistemlerinde kullanılan klasik konuşmacı seçim eşitsizliği aşağıdaki gibi tanımlanabilir:

$$P(X|I) > T(I) \qquad (3.30)$$

Burada P(X|I), I konuşmacısına ait X özellik vektörlerinin bu konuşmacıya ait olma olasılığını göstermektedir. T(I) ise ilgili konuşmacı için tanımlanmış eşik değerini ifade etmektedir. Eşitlik 3.30'daki sabit T(I) eşik değerini kullanmak yerine aşağıdaki eşitsizlikte olduğu gibi uyarlamalı eşik değeri tanımlanabilir:

$$P(X|I) > P(X|\bar{I}) \quad (3.31)$$

Eşitlik 3.31'de P(X|I) olasılığı, I. konuşmacıya ait X özellik vektörünün I dışındaki konuşmacılara ait olma olasılığından büyükse, ilgili konuşmacı doğrulanmış olacaktır. Burada karşılaşabilecek problem ise, $P(X|\bar{I})$ antikonuşmacı olasılığının nasıl tahmin edileceğidir. I Konuşmacısına en yakın konuşmacı modelleri, (I Konuşmacısının eşkonuşmacıları) antikonuşmacı olasılığının tahmininde kullanılacak bir tekniğe zemin oluşturabilir.

Bir konuşmacı doğrulama işleminde, özellik vektörleri I Konuşmacı modeline uygulandığı gibi I'nın eşkonuşmacılarına da uygulanır. Daha sonra, elde edilen I Konuşmacısına ait olasılık, bazı metrik yöntemler kullanılarak, eşkonuşmacı değerlerine oranlanır. Kullanılacak metrik; sınıflandırıcıya uygun olacak şekilde, maksimum, minimum ya da ortalama alan bir yöntem olabilir.

Bu çalışmada, Eşitlik 3.28 ve 3.29'daki her iki değişik eşik değeri kullanılmıştır. VN yöntemi için, bir konuşmacının, doğrulama olasılığının eşkonuşmacılara ait minimum değerlere normalize edilerek bulunmuştur. DYAM yönteminde ise eşkonuşmacıların ortalaması alınmış değerleri kullanılarak normalizasyon yapılmıştır. EHO değeri eşkonuşmacı normalizasyonu kullanılmadan VN için %3.6 DYAM için %2.4 bulunmuştur. Eşkonuşmacı normalizasyonu kullanıldığında ise bu oran, sırasıyla VN için %2.2 ve DYAM için %1.8 olduğu görülmüştür. Eşkonuşmacı normalizasyonun her iki yöntemde de iyileştirici bir sonuç verdiği görülmüştür.

Krishnamoorthy, TIMIT veritabınından 10 konuşmacıyı kullanarak YAM modelini eğitmiş ve test etmiştir (Krisnamoorthy 1998). Rastgele seçilen bu 10 konuşmacı verisi 16 kHz'de örneklenmiş olup, bu veriler 10 ms'lik kısmı birbiriyle örtüşen 20 ms'lik pencereler kullanılarak 13 mel-kepstral katsayılarına dönüştürülmüştür. Bu katsayıların ilki ilgili pencerenin konuşma enerjisini ifade ettiğinden işleme katılmadan, geri kalan 12 katsayı kullanılmıştır. Elde edilen sonuçlar aşağıdaki maddelerde açıklanmaktadır:

- YAM modelinin seviyelerini artırarak performansı iyileştirilebilir. Fakat, belirli bir seviyeden sonra sistem kesime uğramaktadır.

- Düğümlere ilk değer ataması rasgele yapılabilir. Fakat, uygulama sonuçlarına göre, eğer bir düğüme ve alt kollarına da aynı ağırlık değerleri, ya da ağırlıklar çok az değiştirilerek atanırsa; daha az iterasyonla çözüme ulaşmak mümkündür.

- Konuşmacı doğrulama işlemi için 5 seviyeli YAM modelinde güven sınır değeri değiştirilerek, Tablo 3.7'deki sonuçlar elde edilmiştir.

Tablo 3.7'den çıkardığımız yoruma göre %70 güven sınır değeri en iyi performansı vermektedir. YAM modelinin bir konuşmacıyı her seviyesinde sınıflandırabilme yüzdesi de Tablo 3.8'de görülmektedir.

Tablo 3.7. Güven değerine göre HK ve HR yüzdeleri

Güven Değeri (%)	Hata Kabul HK-%	Hata Redetme HR-%
90	0.0	100
80	0.0	87.50
75	0.0	37.50
70	0.0	0.0
60	20	0.0

Tablo 3.8. Düğüm seviyelerine göre YAM'ın eğitim verisini sınıflandırabilme yüzdeleri

YAM Düğüm Seviyesi	Eğitim Verisinin Sınıflandırılma Yüzdesi
1	%79.44
2	%90.10
3	%96.35
4	%96.95

Tablo 3.8'den çıkaracağımız sonuç, YAM modelinin düğüm seviyesi artıkça, ağın sınıflandırma hatası düşmektedir.

Krishnamoorthy, bir başka çalışmasında *DÖK* incelemesini kullanarak *VN* yöntemiyle konuşmacı saptama uygulaması yapmıştır (Krisnamoorthy 1998b). Aynı modele mel-kepstral katsayıları uygulanarak, her iki uygulamanın sonuçları karşılaştırılmıştır. TIMIT veritabanı kullanılarak, 16 kHz'de örneklenmiş 10 konuşmacı seçilmiştir. HTK simülatörü kullanılarak mel-kepstral katsayıları konuşma verisinden üretilmiştir. VN yöntemi kullanılarak 16 ve 32 büyüklüğünde iki kural tablosu oluşturulmuş ve elde edilen sonuçlar Tablo 3.9'da karşılaştırılmıştır.

Tablo 3.9. Kural tablosu büyüklüğüne göre VN sınıflandırıcısının DÖK ve Mel-Kepstral Katsayıları kullanılarak elde edilen performansları

Kural Tablosu Büyüklüğü	DÖK Katsayıları Kullanılarak	Mel-Kepstral Katsayıları Kullanılarak
16	%92.5	%93.75
32	%95.0	%97.5

Tablo 3.9'a göre, VN kural tablosu büyüdükçe performans artmaktadır. Ayrıca, DÖK ve Mel-kepstral katsayıları uygun özellik çıkartım yöntemleri olup Mel-Kepstral katsayılar; DÖK tabanlı uygulamalardan daha iyi sonuç vermektedir.

Öğünç, tezinde, insanları; ses özelliklerini kullanarak tanıyan bir mesaj dinleme/bırakma sistemi gerçekleştirilmiştir. Bu tezin ses işleme aşamasında DÖK tabanlı katsayılar kullanılmıştır. Turbo Pascal 5.0 programının kullanıldığı çalışmada, sınıflandırma aşamasında minimum uzaklık hesaplama algoritması ve konuşmanın Özilişki Matrisi kullanılmıştır (Öğünç 1990).

Öncül, "Kısa Eğitme Süreli Bir Konuşmacı Tanıma Dizgesi Tasarımı Ve Gerçekleştirilmesi" tezini hazırlamıştır. Bu çalışmada, konuşmacıya ait özellikler DÖK katsayıları cinsinden elde edilip karşılaştırmalar Dinamik Zaman Eğriltimi algoritmasıyla yapılmış olup, sistem 14 kişilik deney grubu için % 92'lik başarı göstermiştir (Öncül 1993).

Altınçay, çalışmasında bir konuşmacı tanıma sistemi geliştirmiştir. Laboratuvar ortamında, on konuşmacıyı tanıyan bir Dinamik Zaman Eğriltimi programı kullanılmıştır. Konuşmacı tanıma sistemlerinde kullanılan değişik parametreler incelenmiştir (Altınçay 1995).

Seven, konuşma ve konuşmacı tanıma sistemi oluşturulmasında iki değişik ses tanıma yöntemi kullanmaktadır. İlki kEK- k inci En Yakın Komşuluk yöntemi, ikincisi de ÇKA YSA modelidir. Bu iki yöntem, ses örneklerinin sınıflandırılmasında iki değişik yaklaşımı temsil eder (Seven 1997).

Konuşmacı gurubu 26 bayan ve 22 erkek konuşmacıdan oluşturulmuş olup her konuşmacı 0-9 rakamlarını on kere tekrarlamıştır. Toplanan 4800 ses örneği iki kümeye ayrılıp, sistemin eğitim ve test aşamasında kullanılmıştır.

İlk uygulama kEK tabanlı kapalı set konuşmacı saptama uygulaması olup k sırasıyla 1, 3, 5 ve 7 seçilmiştir. Cinsiyet ayrımına karşı sistemin başarısını araştırmak üzere ayrı ayrı sadece bayan ve bay konuşmacıların olduğu uygulamalar gerçekleştirilmiştir. Daha sonra hem bayan hem de bay konuşmacıların yer aldığı bir uygulama gerçekleştirmiştir. Uygulamalara göre, sistem verimi açısından cinsiyetin pek önemi olmadığı görülmüştür. Bu sonuçlar sırasıyla Tablo 3.10, 3.11 ve 3.12'de gösterilmiştir.

Tablo 3.10. Kapalı Set Konuşmacı Saptama kEK Uygulaması
(Tüm Konuşmacılar Erkek)

k Komşuluk Sayısı	5 Konuşmacı (%)	10 Konuşmacı (%)	20 Konuşmacı (%)
1	100	90	83
3	100	92	90
5	100	95	93
7	100	100	97

Tablo 3.11. Kapalı Set Konuşmacı Saptama kEK Uygulaması

(Tüm Konuşmacılar Bayan)

k Komşuluk Sayısı	5 Konuşmacı (%)	10 Konuşmacı (%)	20 Konuşmacı (%)
1	100	85	78
3	100	92	83
5	100	96	90
7	100	100	95

Uygulamalarda k komşuluk değerinin kaç alınacağı pek de açık değildir. Konuşmacı seti büyüdükçe k değerinin de arttırılması önerilen bir yöntemdir.

Tablo 3.12. Kapalı Set Konuşmacı Saptama kEK Uygulaması

(Konuşmacı Cinsiyetleri Karışık)

k Komşuluk Sayısı	6 Konuşmacı (3Byn/3Erk)	10 Konuşmacı (5Byn/5Erk)	20 Konuşmacı (10Byn/10Erk)	48 Konuşmacı (22Byn/26Erk)	Ortalama Verim
1	% 100	% 82	% 75	% 70	% 81.8
3	% 100	% 88	% 80	% 77	% 86.3
5	% 100	% 92	% 85	% 82	% 89.8
7	% 100	% 96	% 92	% 86	% 93.5

Seven 'in diğer çalışması ise ÇKA modeli olup kapalı set konuşmacı saptama uygulaması için, geri yansıtma ağı ile tek saklı katman için işlem birimi sayısını sırasıyla 8, 16, 32 ve 64 olarak değiştirmiştir. Tablo 3.13'te ÇKA modeline ilişkin konuşmacı saptama sistem verimi gösterilmektedir. Saklı katman sayısını artırmanın sistem verimini iyileştirmediği görülmüştür.

Tablo 3.13. ÇKA Modeli Kapalı Set Konuşmacı Saptama Verimi

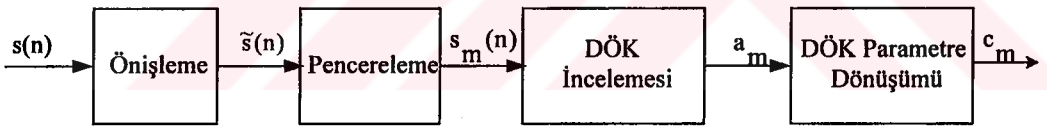
Saklı Katman İşlem Birimi Sayısı	Ortalama Saptama Yüzdesi (%)
8	83.5
16	88.7
32	92.5
64	89.8

BÖLÜM 4. AMAÇLANAN YÖNTEMLER VE TÜRKÇE KONUŞMACI VERİTABANININ OLUŞTURULMASI

Bu bölümde, tez çalışmamızda kullanacağımız, Şekil 2.2'deki genel konuşmacı tanıma sistemindeki aşamalar ve Türkçe konuşmacı veritabanı oluşturma aşamaları anlatılacaktır. Daha sonra, SOM sınıflandırıcısının çıkışında, karar birimi olarak BBM ağının kullanıldığı, karma yapı açıklanacaktır.

4.1. Özellik Çıkartım Aşaması

Şekil 2..2'deki Özellik çıkartımı 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 özellik çıkartım aşamasının blok diyagramı Şekil 4.1'de görülmektedir.



Şekil 4.1. Özellik çıkartım aşamaları

4.1.1. Önişleme aşaması

Sayısallaştırılmış konuşma işareti, s(n), aşağıda transfer fonksiyonu verilen birinci dereceden sayısal süzgeçten geçirilerek, işaretin ani frekans değişimlerine bağlı yüksek seviyeli genlik değişimlerini bastırarak, işaretin ani değişim gösteren geçiş bölgeleri yumuşatılmıştır.

$$H(z) = 1 - \tilde{a}z^{-1} \qquad (4.1)$$

Önişleme girişine uygulanan s(n) konuşma işaretine göre, önişleme çıkışında elde edilen $\tilde{s}(n)$ ifadesinin fark denklemi, aşağıdaki eşitlikte görüldüğü gibidir:

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1) \tag{4.2}$$

Eşitlik 4.2'deki, birinci dereceden sayısal süzgeç katsayısı $\tilde{a}$, [0.9-1.0] aralığında seçilmektedir. Bir çok konuşma ve konuşmacı tanıma uygulamalarında bu değer 0.95 seçilmiştir (Rabiner 1993).

4.1.2. Pencereleme işlemi

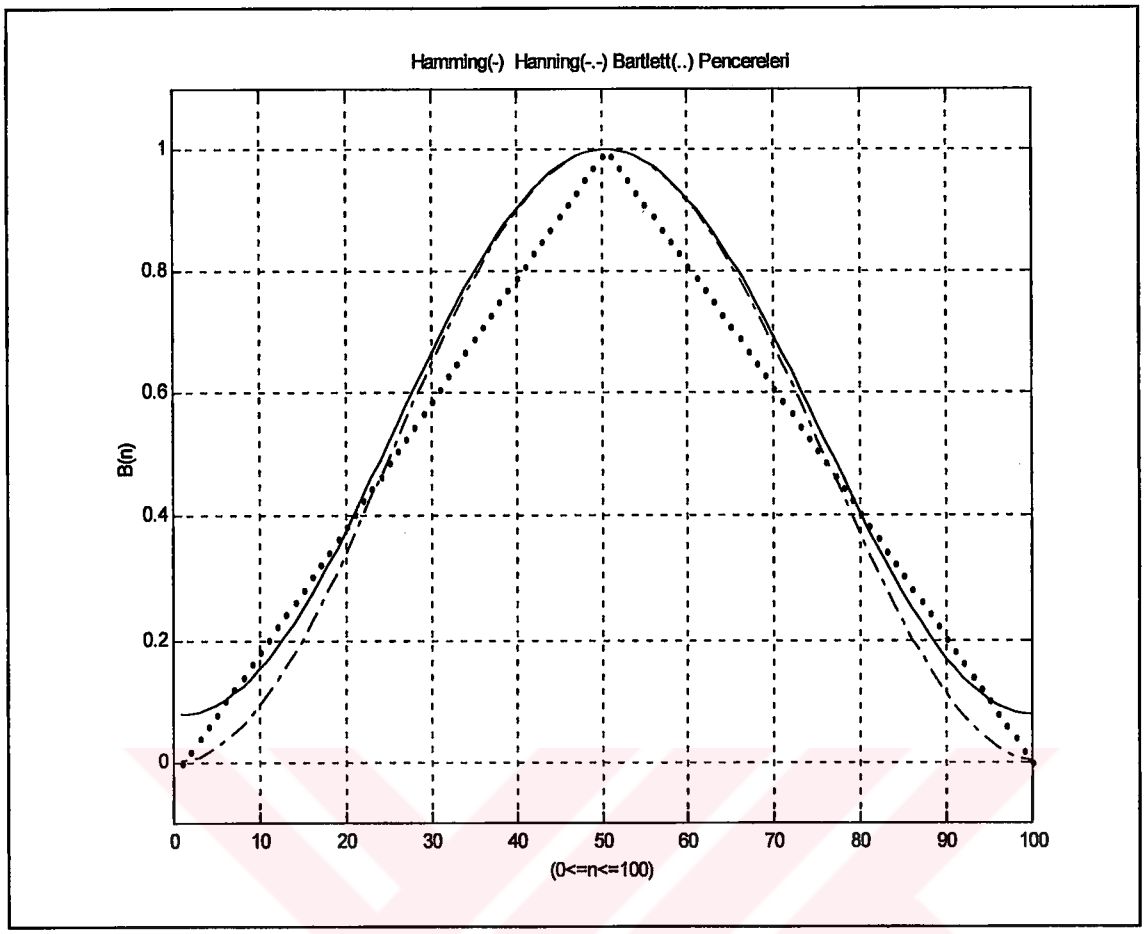
Bir sonraki özellik çıkartım aşaması, sayısal süzgeçten geçirilmiş konuşma işaretinin, pencerelenmesi işlemidir. Pencereleme işleminin amacı, konuşma işaretinin her bir bölütünün başlangıç ve bitimi arasındaki süreksizliği minimuma indirmektir. Bir başka değişle, ilgili bölütün başlangıç ve bitişini sıfıra yakın bir noktadan başlayarak sivrileştirmektir. B(n) pencere katsayıları kullanılarak, konuşma bölütünün uzunluğu N olan (0<= n <=N-1), her konuşma bölütü için pencerelenen işaret aşağıdaki eşitlikteki gibidir:

$$s_m(n) = \tilde{s}(n)B(n) \tag{4.3}$$

DÖK inceleme yönteminde kullanılan, tipik Hamming penceresinin ifadesi Eşitlik 4.4'teki gibidir:

$$B(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \le n \le N-1 \tag{4.4}$$

Hamming penceresi dışında kullanılan Hanning, Bartlett gibi pencereler de bulunmaktadır. Şekil 4.2'de bu pencerelerin 100 nokta için değişim şekli görülmektedir. Şekil 4.2 incelendiğinde Hamming penceresinin başlangıç ve bitiş noktalarında sıfır olmadığı görülmektedir. Bu durum pencerenin, bölütün başlangıç ve bitim değerlerini sıfıra kadar çökertmediğini göstermektedir. Böylelikle işaretin anlamlı olabilecek değerleri, tekrarlanan pencereler boyunca korunmuş olacaktır.



Şekil 4.2. Pencereleme işleminde kullanılan pencere çeşitleri

Bir sonraki özellik çıkartım aşaması ise DÖK incelemesi olup Bölüm 2.2'de DÖK modelinde açıklandığı gibi a parametre takımındaki katsayılar, konuşma işaretinin her $s_m(n)$ bölütü için, bulunmuştur.

Konuşma işaretinin tüm bölütleri için bulunan a_m parametre takımındaki katsayılar kullanılarak, son aşama olan DÖK parametre dönüşüm aşamasında, Eşitlik 2.5'teki yinelemeli yönteme göre c_m parametre takımı hesaplanmıştır.

4.2. Türkçe Konuşmacı Veritabanının Oluşturulması

Biriktirilen verilerin kalitesi; kayıt ortamı, donanım ve yazılımlar, örnekleme oranı, örnekleme frekansı ve bu işaretlere uygulanan ön işlem aşamaları gibi faktörlere bağlıdır. Bu işlemlere ilişkin detaylı bilgiler aşağıdaki alt bölümlerde verilmiştir. Konuşmacıların adını ve soyadını telaffuz ettikleri örnek dalga şekilleri;

Şekil 4.3 - Şekil 4.12'da gösterilmiştir. Bu dalga şekilleri, konuşma örneklerinin, zaman-genlik ilişkisini göstermekte olup, genlik ± 1 aralığına ölçeklenmiştir.

4.2.1. Kayıt ortamı

Kayıt ortamı; kayıt yapılan bilgisayar dışında çalışır durumda iki bilgisayarın daha yer aldığı, 20 m^2'lik alana sahip bir ofistir. Ofiste çalışan kişilerin o anda sessiz de olsa hareket halinde olmaları ve çalışan bilgisayarlardan kaynaklanan, düşük seviyeli artalan gürültüsü oluşmuştur. Her kayıt başlamadan önce ortam gürültüsünün seviyesini belirlemek için kayıtların başında sessiz bölümler bırakılmıştır. Kayıt ortamını, düşük seviye gürültülü ofis ortamı olarak açıklayabiliriz.

4.2.2. Donanım ve yazılım

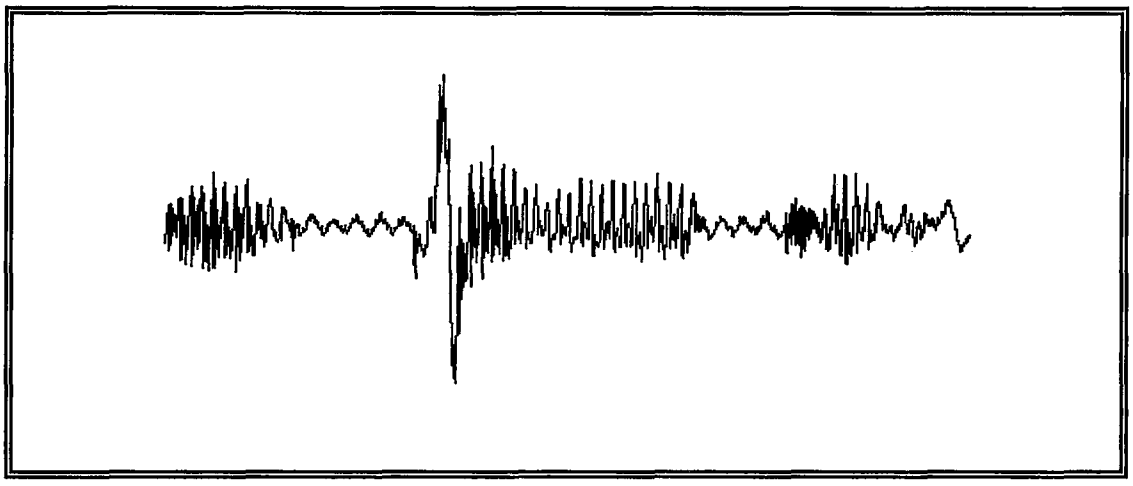
Pentium 233 MMX, 48 Mb bellek kapasitesi olan kişisel bilgisayarda, 16-bit Creative Sound Blaster ses kartı ve 500 Ω dinamik, harici, masaüstü mikrofonun kullanıldığı sistemde; kayıt etme ve kayıttan yürütme işlemleri için yine Creative 'in kendi yazılımı kullanılmıştır.

4.2.3. Örnekleme frekansı

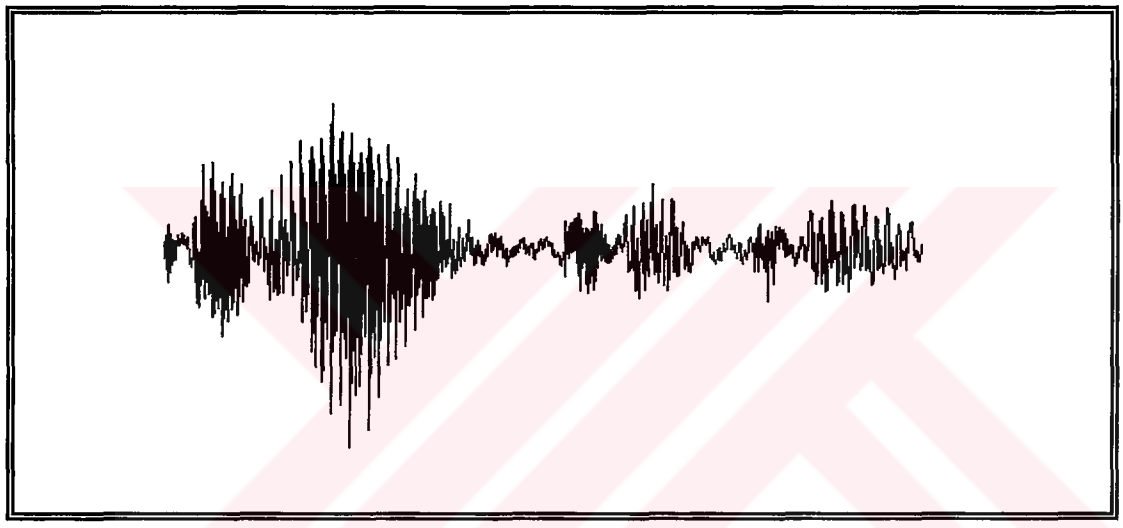
Örnekleme frekansı 11 kHz seçilmiştir. Bunun dışında 22 ya da 44 kHz'lik kayıtlar da yapmak mümkün olmasına rağmen, fazla alan, daha uzun eğitim ve işleme süresi gerektirmektedir.

Konuşma işaretinin yeterince temsil edilebilmesi için 8 bitlik örnekleme oranı yeterli olmasına rağmen daha doğru sonuçlar elde etmek için bu çalışmada örnekleme oranı 16 bit seçilmiştir. Kayıtlarda çift kanal kullanımı daha fazla işlem ve alan kaplayacağı düşüncesiyle, tek kanal kullanılmıştır.

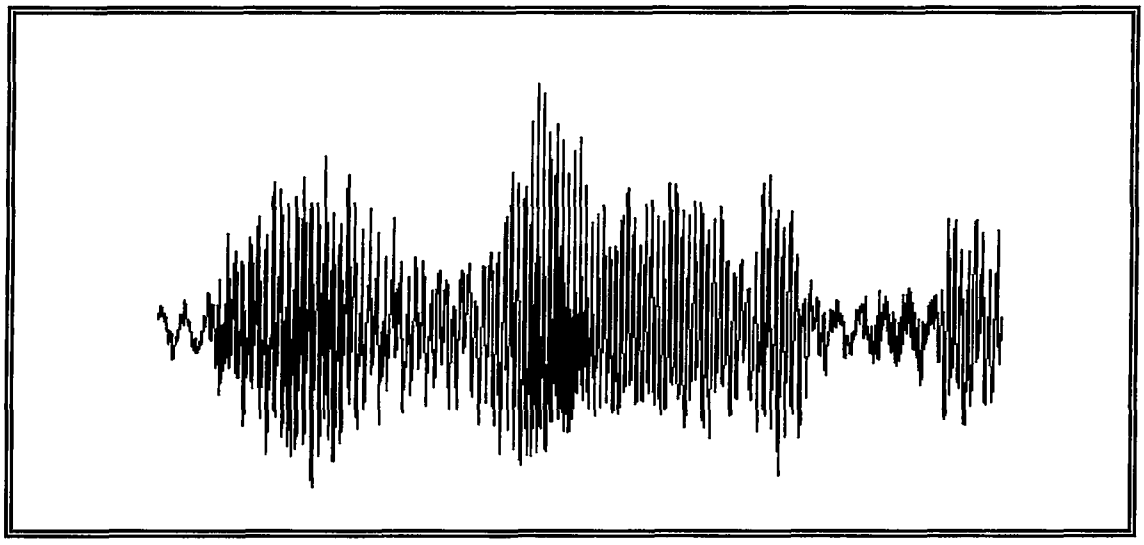
Genellikle bir çok konuşma ve konuşmacı tanıma uygulamalarında, örnekleme frekansı 8-20 kHz arasında değişen oranlarda, örnekleme oranı 16 bit ve kayıtlarda tek kanal seçilmektedir.



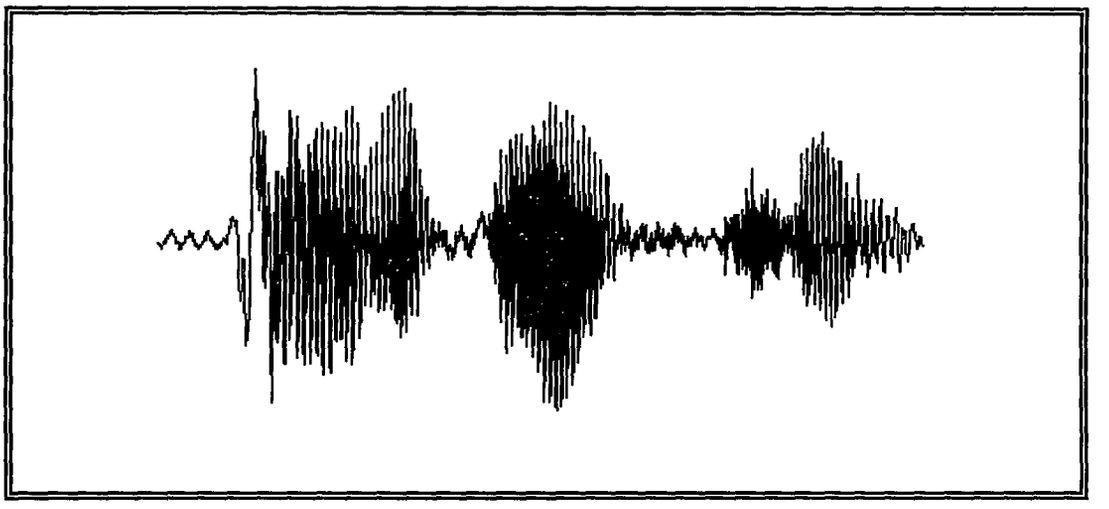
Şekil 4.3. "Alper Metin" telaffuzuna ait dalga şekli



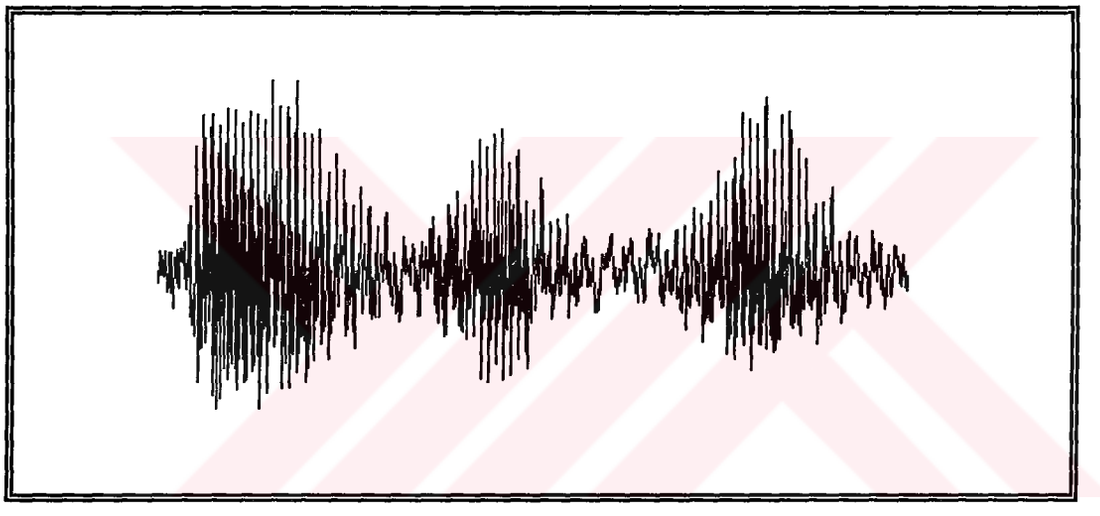
Şekil 4.4. "Celal Çeken" telaffuzuna ait dalga şekli



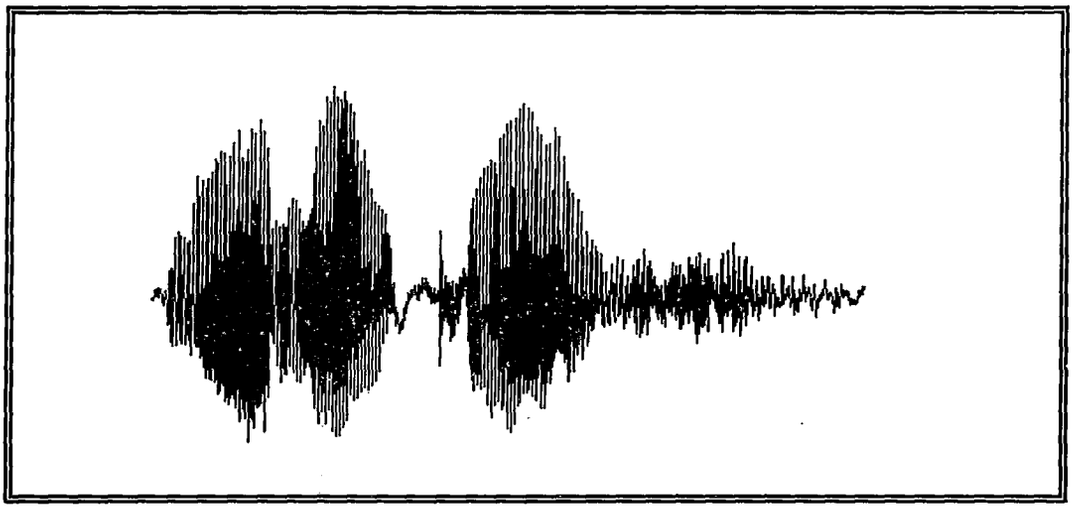
Şekil 4.5. "Erhan Bütün" telaffuzuna ait dalga şekli



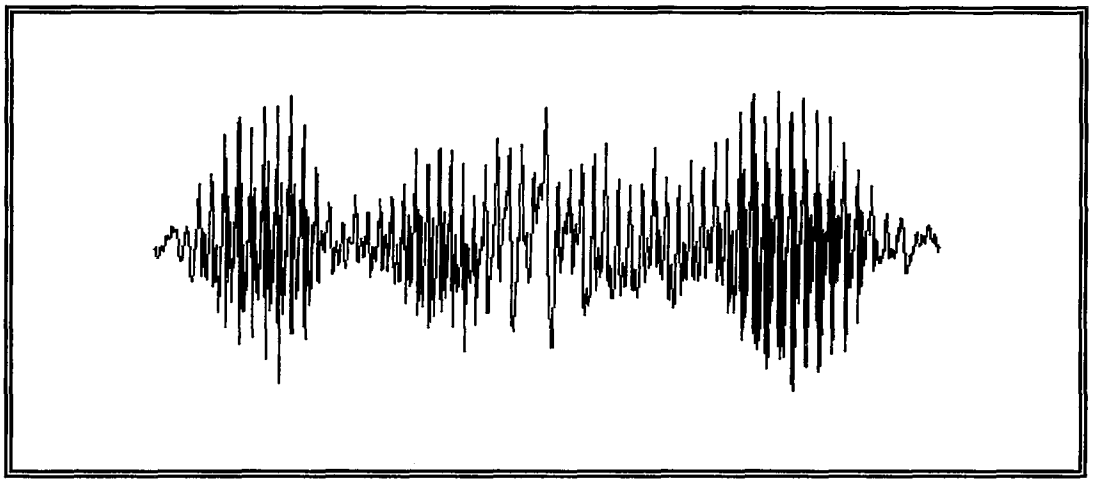
Şekil 4.6. "Faruk Arkçı" telaffuzuna ait dalga şekli



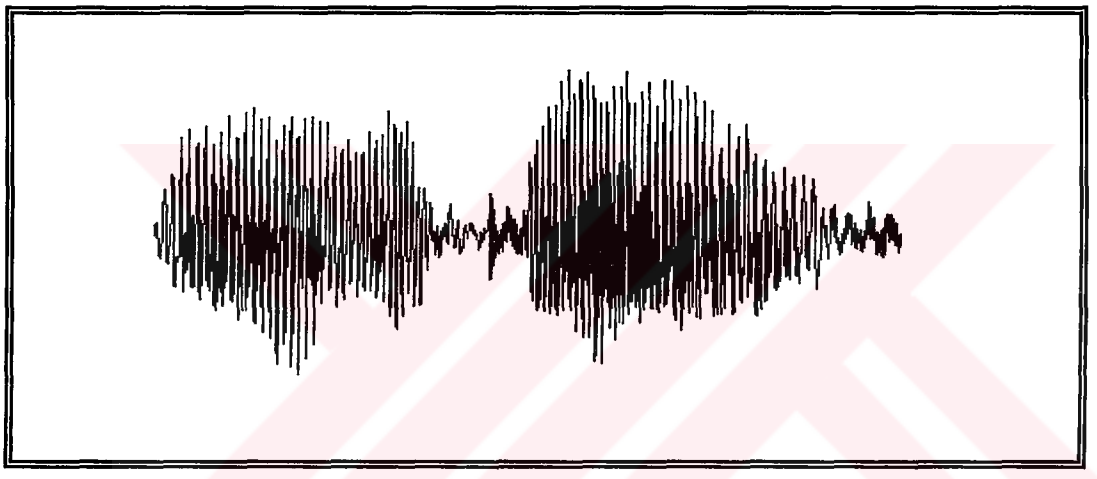
Şekil 4.7. "Hüseyin Çirkin" telaffuzuna ait dalga şekli



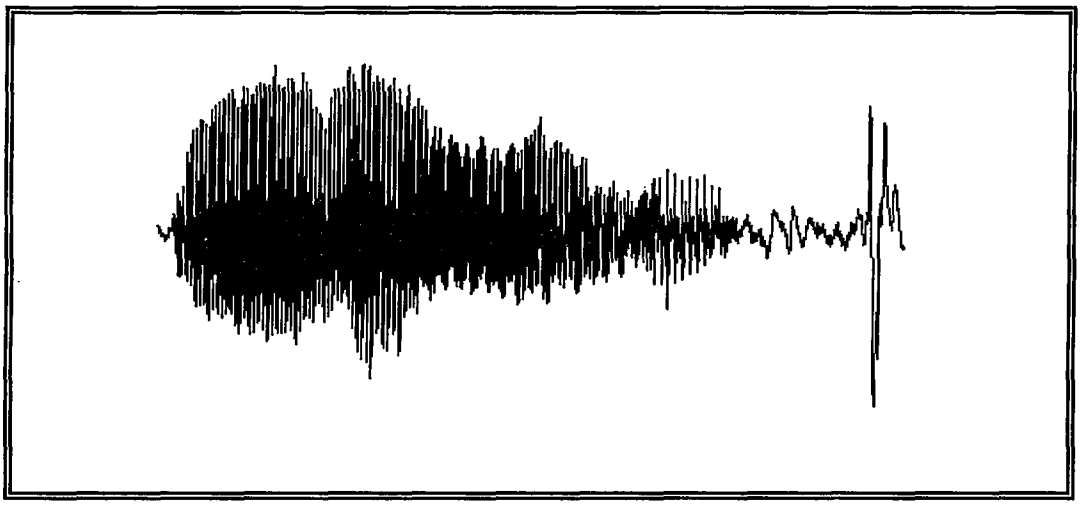
Şekil 4.8. "Melek Özcan" telaffuzuna ait dalga şekli



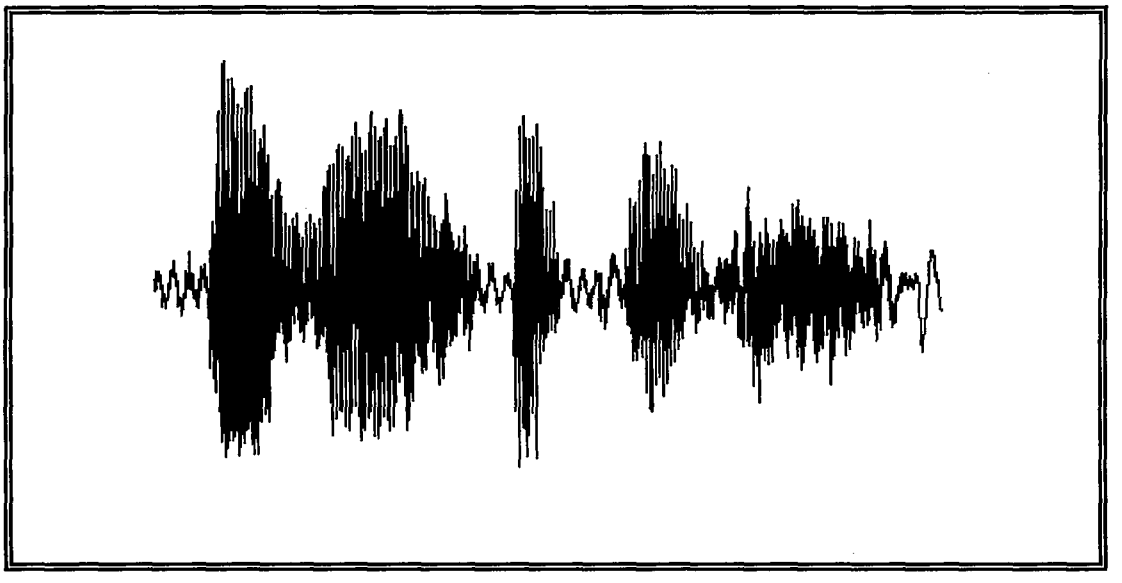
Şekil 4.9. "Melih İnal" telaffuzuna ait dalga şekli



Şekil 4.10. "Namık Yener" telaffuzuna ait dalga şekli



Şekil 4.11. "Nuran Yılmaz" telaffuzuna ait dalga şekli



Şekil 4.12. "Seval Ataş" telaffuzuna ait dalga şekli

4.2.4. Ses örneklerinin hazırlanması

Her konuşmacının 8 kez tekrarladığı konuşma örneklerinden dolayı, dalga şekillerinin saklandığı dosyalar doğal olarak büyük olmaktadır. Bu dosyaların tek başına özellik çıkartım algoritmalarına uygulanması oldukça güç olacağından, Sound Blaster kartının "Wave Studio" yazılımı kullanılarak her ayrı telaffuz farklı dosyalara alınmış ve ses dosyalarındaki sessiz bölgeler kayıttan el ile silinmiştir. Yapılan bu ön işlemler oldukça zahmetli ve uzun bir aşamadır. Ayrıca, unutulmamalıdır ki, ön işleme ve özellik çıkartım işlemi en uygun şekilde yapılmış bir verinin; içereceği anlamlı bilgiler daha fazladır.

4.2.5. Türkçe konuşmacı veritabanının özellikleri

Türkçe konuşmacı veritabanı için, yaşları 22-55 arasında değişen 10 konuşmacının kendi ad ve soyadlarını 8 kez tekrarladıkları metne bağlı, kapalı set konuşmacı tanıma uygulamaları yapılmak üzere oluşturulmuştur. Konuşmacılardan 3 tanesi bayan, 7 tanesi erkek seçilmiştir. Önceden söz edilen kayıt ortamı altında 10 konuşmacıya ait 80 telaffuz elde edilmiştir.

4.2.6. Türkçe konuşmacı veritabanının özellik çıkartım işlemleri

Konuşma örneklerine uygulanan özellik çıkartım aşamaları maddeler imleri halinde sıralanmıştır.

- Konuşma işaretleri, transfer fonksiyonu Eşitlik 4.1'deki gibi birinci dereceli sayısal süzgeçten geçirilmiştir. Süzgeç katsayısı 0.98 seçilmiştir.

- Konuşma işaretinin her 330'luk bölütü için Hamming pencereleme yöntemi kullanılmıştır. Ayrıca her 330'luk bölüt için uygulanan pencereleme işlemi, pencereler arası geçişte veri kaybını önlemek için; her 110 örnekte bir tekrarlanmıştır (overlap). Buna göre 330 örneklik bir pencere 33 ms.'lik konuşma parçasını ifade ettiğine göre; her 110 örneklik kısım için pencereler tekrarlandığına göre, konuşma işareti boyunca her 11 ms.'lik aralık için bir pencereleme işlemi yapılmaktadır

- Bu çalışmada, özellik çıkartım yöntemlerinden DÖK incelemesi kullanılmış ve 12 inci dereceden DÖK tabanlı doğrusal öngörülü a parametre takımı konuşma işaretinin her bölütü için hesaplanmıştır.

- DÖK incelemesi sonucunda elde edilen a parametre takımından kepstral c parametre takımı Eşitlik 2.5'e göre türetilmiştir.

4.3. Amaçlanan Yöntem

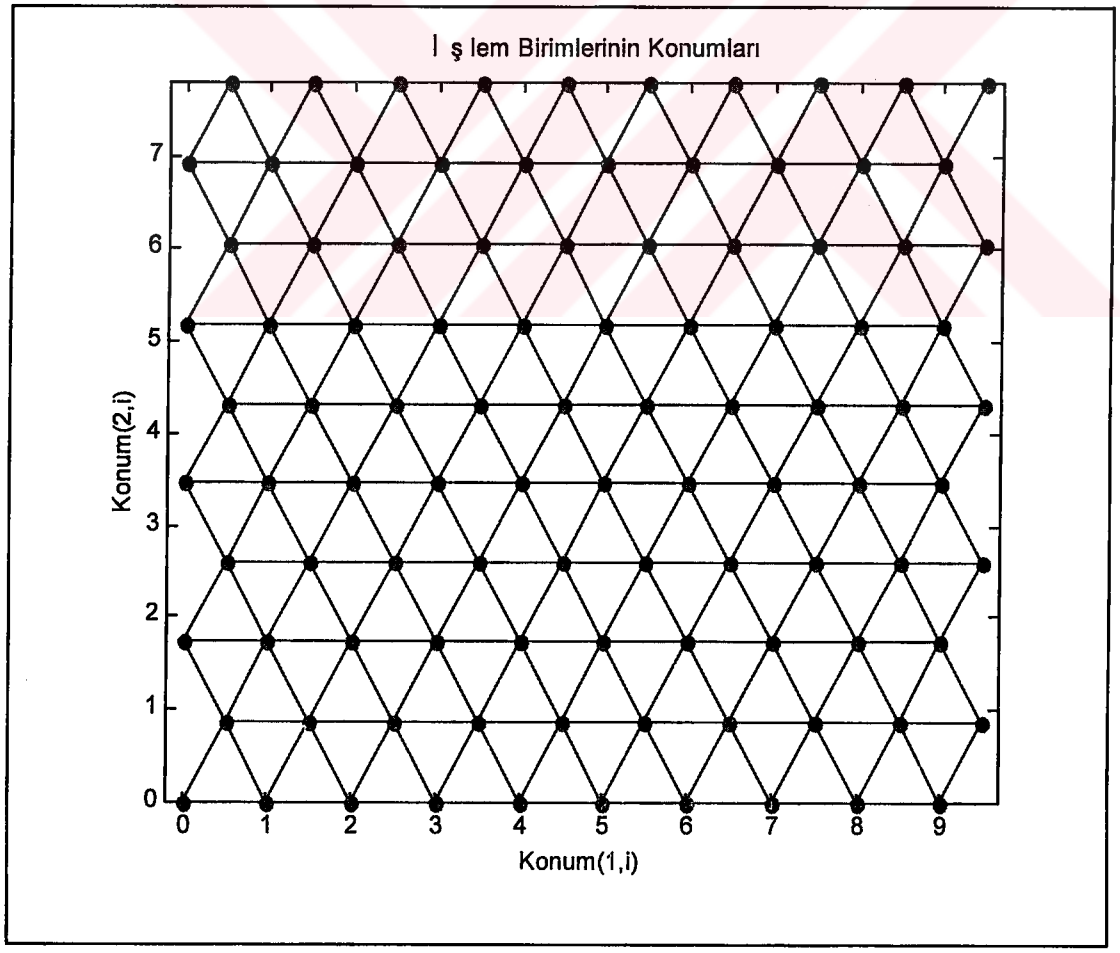
Çalışmalarımızda, ÇKA YSA sınıflandırıcılarının dışında, Şekil 2.5'te görülen özellik vektörlerinin uygulandığı sınıflandırıcı yapısına göre, SOM ve BBM ağlarından oluşan, karma bir sınıflandırıcı yapısının kullanılması amaçlanmaktadır.

SOM YSA sınıflandırıcısının eğitiminde, Bölüm 3.2.3'te açıklanan eğiticisiz öğrenme algoritmalarından Yarışmacı Öğrenme algoritması kullanılmıştır. SOM ağı eğiticisiz bir öğrenme algoritması kullandığı için ve ağ kendi çağrışımlarını kendisi yaptığından eğiticili öğrenme algoritmalarına göre daha hızlı yakınsamaktadır.

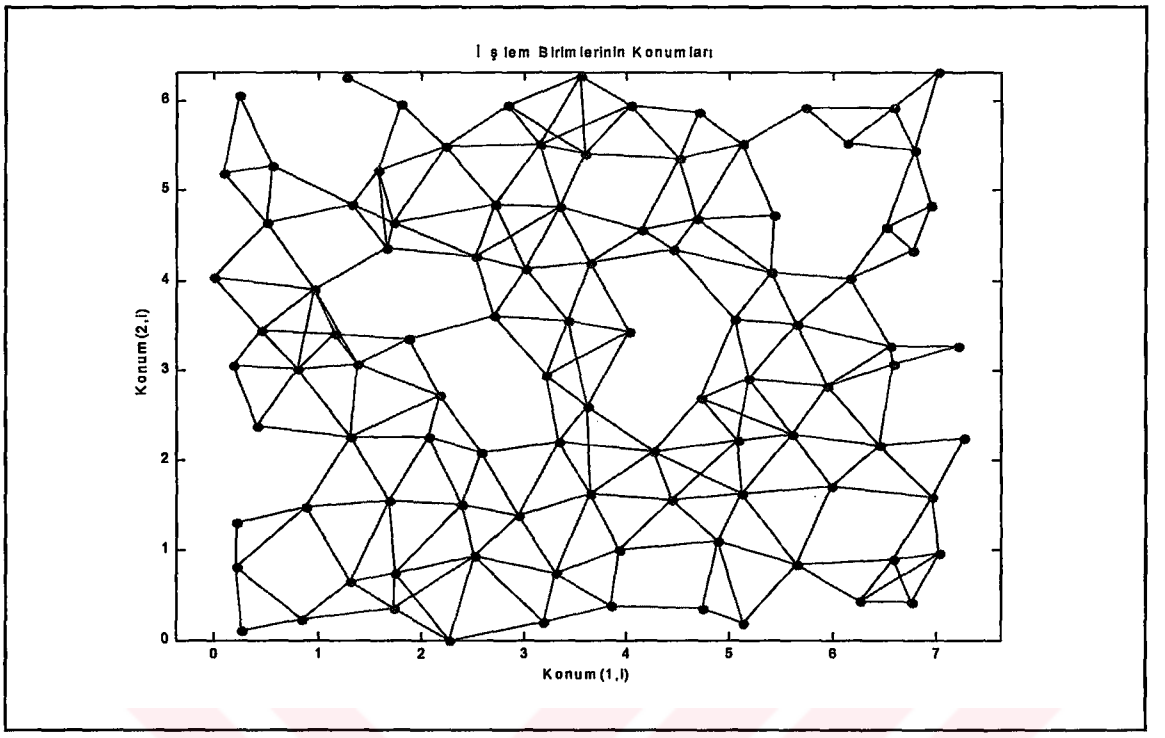
Yarışmacı öğrenme kuralına göre yarışı kazanarak çıkış üreten işlem biriminin çıkışı "1" olacaktır. Yarışı kazanan işlem birimlerinin diğer işlem birimlerinin çıkış üretmesini yasaklayabilme özelliğinden dolayı diğer işlem birimlerinin çıkışı "0" olacaktır. Yarışı kaybeden işlem birimleri çıkış üretemez.

SOM ağının yapısı gereği, ağdaki işlem birimlerinin konumlarına göre tek ya da çok boyutlu ağ oluşturmak mümkündür. Çok boyutlu ağ mimarisinde, ağın katmanları çok boyutlu değil, işlem birimlerinin konumları itibariyle birden fazla boyut olduğu söylenebilir; SOM ağı, daima tek katmanlıdır.

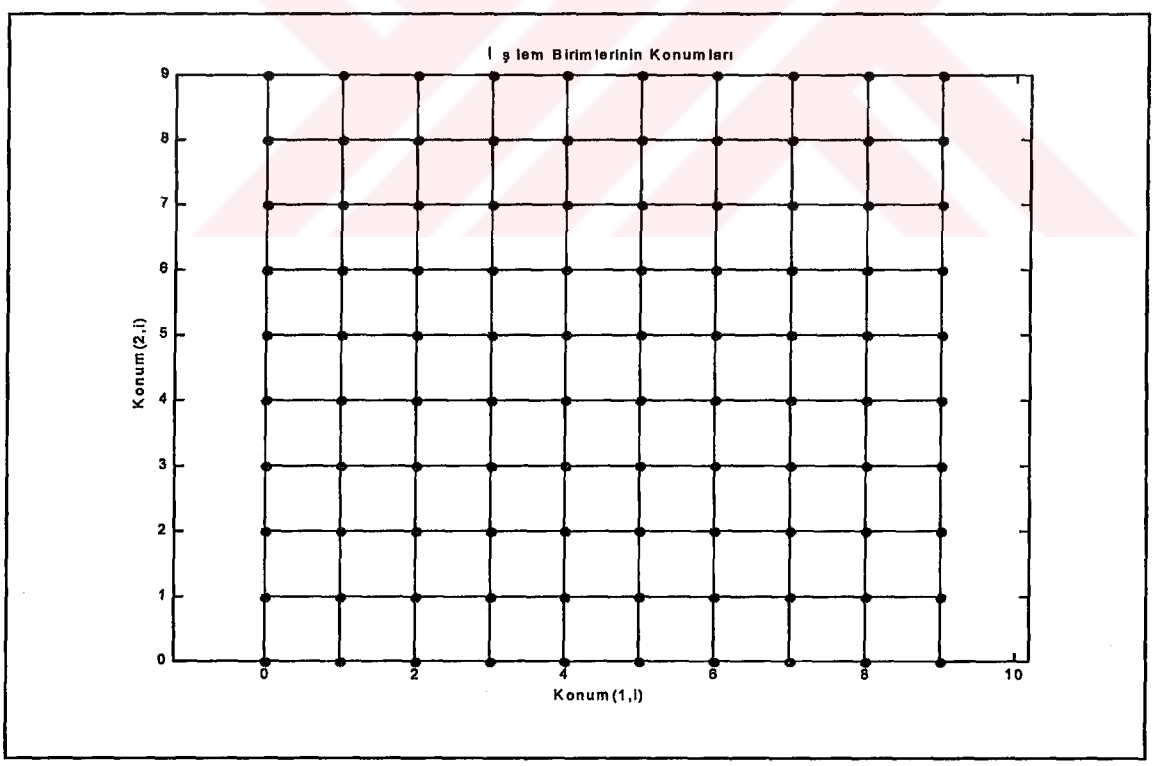
Eğitim öncesi SOM ağındaki işlem birimlerinin konumları Şekil 4.13 a)'da görüldüğü gibi üçgen konumunda seçilmiştir. Şekil 4.13 b) ve c)'de görüldüğü gibi rasgele ya da kare konumunda da seçilebilir.



a) Üçgen konum



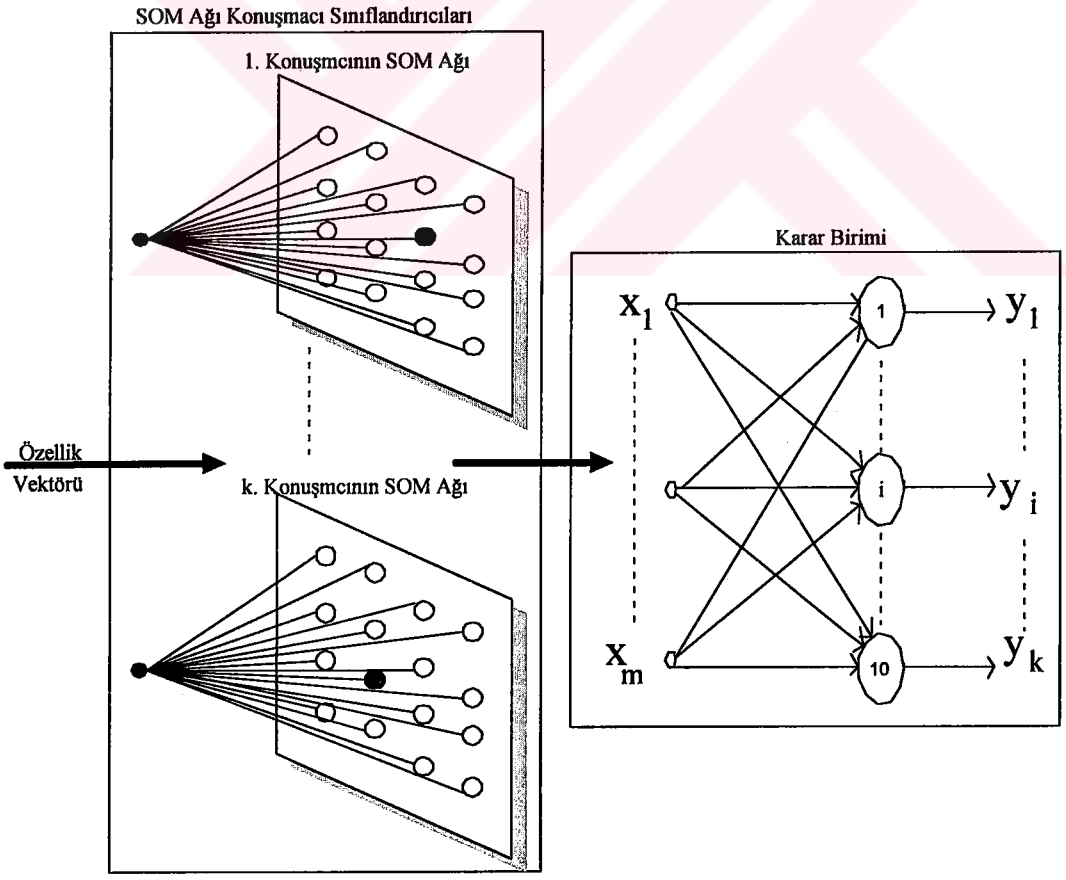
b) Rasgele konum



c) Kare konum

Şekil 4.13. Eğitim öncesi SOM ağındaki işlem birimlerinin konumları

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 kolay olmamaktadır. SOM ağı gibi eğiticisiz öğrenme algoritmalarının kullanıldığı, çıkışı gözlemlenecek ağlarda BBM gibi hem matematiksel ifadesi kolay hem de gerçek sınıflandırıcının karakteristiğini bozmayacak doğrusal ağlara ihtiyaç vardır. BBM ağı, konuşmacı saptama işleminde, bir konuşmacıya ait verilerin ağa sunulmasıyla, bu verilerin hangi konuşmacıya ait olabileceği konusunda bir fikir verecektir. Bu amaçla, Şekil 4.14'te görülen karma yapıda BBM ağı, eğitimi tamamlanmış SOM ağlarının çıkışlarında kullanılarak, konuşmacının saptanması için bir karar birimi olarak kullanılmıştır. Şekil 4.14'te görülen karma yapı incelendiğinde, Şekil 2.5'teki konuşmacı tanıma sistemi için sınıflandırıcı yapısında olduğu görülmektedir.



Şekil 4.14. SOM ve BBM ağının kullanıldığı karma ağ yapısı

BBM ağının eğitimi için, önce her konuşmacıya ait SOM ağının eğitimi yapılır. Daha sonra SOM ağının eğitiminde kullanılan vektörlere göre SOM ağları test edilerek, elde edilen çıkışlara göre BBM ağının eğitim seti oluşturulur.

Örneğin, konuşmacı seti 10 kişiden oluşan ve her konuşmacı için 10X10 işlem biriminin yer aldığı SOM ağları kullanılarak oluşturulan bir konuşmacı saptama sisteminde, BBM ağının girişinde 100 işlem birimi ve BBM ağının çıkışında da her konuşmacıyı gösteren 10 işlem birimi kullanılmalıdır. Her SOM ağının çıkışında 100 işlem birimi olduğuna göre, tek bir eğitim vektörü için, SOM ağında hangi işlem birimi çıkış üretmiş ise, BBM ağının eğitim vektöründe de o işlem biriminin çıkışı "1", diğerlerinin çıkışları "0" seçilir. Daha sonra, bu eğitim vektörü hangi konuşmacıya ait ise, BBM ağında o konuşmacının hedef değeri "1", diğerlerinin hedef değeri "0" seçilir. SOM ağlarının tüm eğitim vektörleri için elde edilen değerler, vektörler halinde biriktirilerek, BBM ağının eğitim seti oluşturulur. Elde edilen eğitim vektörleri ve hedef vektörlere göre, BBM ağının ağırlıkları hesaplanarak, ağın eğitimi sağlanmış olur. Bundan sonra bir test vektörü, bütün SOM ağlarına uygulanarak elde edilen çıkışlara göre BBM ağının eğitim vektörü oluşturularak, önceden hesaplanan ağırlıklarla çarpımından test vektörünün hangi konuşmacıya ait olduğu saptanmış olacaktır.

BÖLÜM 5. ÇKA TABANLI KONUŞMACI TANIMA UYGULAMALARI

Bu bölüm, eğiticili öğrenme algoritmalarından ÇKA YSA sınıflandırıcılarının kullanıldığı, metne bağlı kapalı set konuşmacı tanıma uygulamalarına ilişkin çalışmaları içerecektir. Bu uygulamaya ilişkin veriler, Bölüm 4.2'de özellikleri anlatılan Türkçe konuşmacı veritabanı kullanılarak hazırlanmıştır.

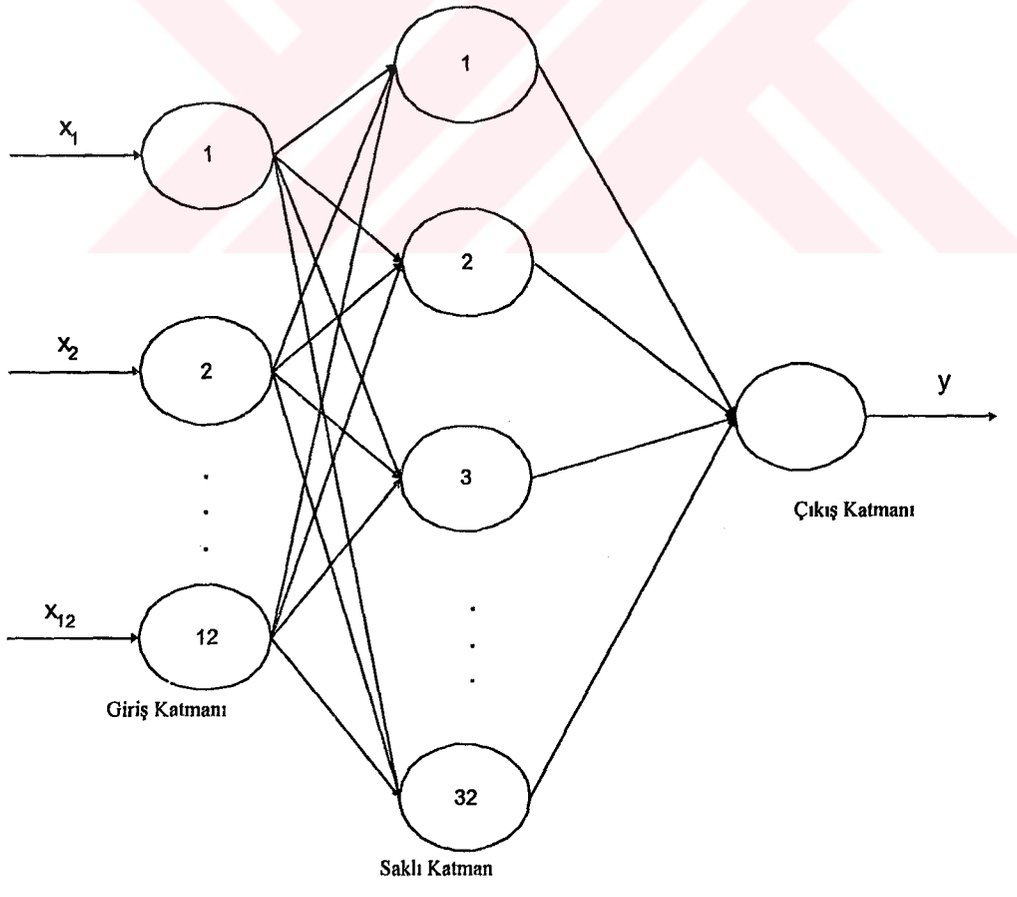
ÇKA YSA modelini konuşmacı tanıma sistemine uygulamak için tüm konuşmacılara ait özellik vektörleri oluşturulur. Tanınması istenen konuşmacıya ait konuşma verisinin özellik vektörlerine "1", diğer konuşmacıların özellik vektörlerine ise "0" ya da "-1" değeri etiketlenir. Bütün konuşmacılara ait özellik vektörleri kullanılarak, ÇKA modelinin eğitimi yapılır. Belirli bir konuşmacıya ait test vektörleri için eğitilmiş ÇKA modelinin çıkışı "1", diğer konuşmacılara ait test vektörleri için "0" ya da "-1" değerini vermelidir.

Konuşmacı saptama işleminde, bütün test vektörleri önceden eğitilmiş her ÇKA ağına uygulanır ve her ÇKA modelinin çıkışı, girişine uygulanan her test vektörünün üreteceği çıkışlar için ayrı ayrı toplanır. En büyük çıkışı üreten ağın konuşmacısı, test vektörüne ait konuşmacı olarak saptanır. Konuşmacı doğrulama işleminde ise, bir konuşmacıya ait test vektörleri, ilgi konuşmacıya ait önceden eğitilmiş ağa uygulanır. Üretilen çıkışların hepsi toplanarak test vektörleri sayısına bölünür. Elde edilen çıkış, belirlenen eşik değerini aşıyorsa, bu konuşmacı doğrulanmış olur aksi halde konuşmacı reddedilir.

Geniş konuşmacı seti için ÇKA ağının eğitiminde karşılaşılan problemlerden biri de, eğitim vektörlerinin büyük çoğunluğunun yasaklayıcı (sıfır) etiketlere sahip olmasıdır. Bu durum ağın çıkışını kesime götürüp, ağ; her test vektörünün "yasaklayıcı" özelliğe sahip olma durumunu öğrenir. Bu olumsuz etkiyi ortadan kaldırmak için, ağın eğitiminde "1" etiketli test vektörlerinin sayısını artırıcı gürültülü vektörler kullanılabilir. Böylelikle, yasaklayıcı ve uyarıcı vektörlerin

dağılımı eşit düzeye getirilmiş olur. Bu durum geniş konuşmacı seti için, yapılacak uzun süreli eğitim sonunda bile, sınıflandırıcının istenen değere yakınsamasını güçleştirecek ya da geciktirecektir. Bu olumsuzluğun da etkisini azaltmak için, uyarıcı yapay test vektörleri yaratmak yerine, yasaklayıcı vektörlerin sayısını azaltmak için vektör nicemleme gibi sıkıştırma işlemi uygulanabilir. Böylece hem eğitim seti azaltılmış hem de yasaklayıcı-uyarıcı test vektörleri arasındaki denge kurulmuş olacaktır. Çalışmamızda, VN işlemi için, K-ortalamalı Linda-Buzo-Gray (LBG) algoritması kullanılarak, her konuşmacı için o konuşmacı dışında kalan diğer konuşmacıların eğitim ve test vektörleri, sıkıştırılmıştır.

Çalışmalarımızda Şekil 5.1'deki ÇKA mimarisi kullanılmıştır. ÇKA mimarisinin sistem verimine etkisini araştırmak üzere yaptığımız çalışmalarda, tek saklı katmanda farklı sayıda işlem birimi kullanılmıştır (İnal 2001). Özellik vektörleri olarak; DÖK incelemesi yapılmış, 12 dereceden katsayılar kullanıldığı için giriş katmanı işlem birimi sayısı 12 seçilmiştir.



Şekil 5.1. Yapılan çalışmalarda kullanılan ÇKA modeli

5.1. DÖK Tabanlı Özellik Çıkartım Algoritmalarının İncelenmesi

Konuşma işaretinden özellik vektörlerinin çıkartımı için DÖK tabanlı parametre takımları olarak doğrusal öngörülü a(n), yansıma K(n) ve kepstral (cepstral) katsayılar c(n) sayılabilir. Bu parametre takımlarından yansıma katsayıları, doğrusal öngörülü katsayılardan özilişki fonksiyonu tabanlı yinelemeli bir yöntemle türetilebilir (Rabiner 1993).

Yaptığımız çalışmada bu katsayılardan kepstral katsayıların, sistemin konuşmacı tanıma verimi açısından en etkin yöntem olduğu görülmüştür (İnal 2000). Bu çalışmada kepstral katsayıları, yinelemeli bir yöntemle DÖK katsayılarından Eşitlik 2.5'teki gibi türetilebilir. Eşitlik 2.5'teki p, kepstral katsayıların terim sayısını (derecesini) göstermekte olup, 12 seçilmiştir. ÇKA Modeli, eğiticili öğrenme algoritmalarından Geri Yansıtma öğrenme kuralı ile eğitilmiştir. DÖK katsayılarının terim sayısı 12 olduğundan, ağın giriş katmanı işlem birimi sayısı 12 seçilmiştir. Ağın tek bir saklı katmanında 32 ve çıkış katmanında ise 1 işlem birimi kullanılmıştır. DÖK tabanlı katsayıların sistem verimi açısından sonuçları Tablo 5.1'de gösterilmiştir. Her konuşmacıya ait doğru sınıflandırılmış vektör sayısının, ilgili konuşmacıya ait bütün vektör sayısına bölünmesiyle, her konuşmacıya ait yüzdelik verimler bulunmuştur. Daha sonra bütün konuşmacılara ait yüzdelik verimlerin ortalaması alınarak, sistem verimi hesaplanmıştır.

Tablo 5.1. Doğrusal Öngörülü Tabanlı Katsayıların Sistem Verimi

DÖK TABANLI YÖNTEMLER	TANIMA VERİMİ %
DÖK Katsayıları (a)	82
Yansıtma Katsayıları K)	84
Kepstral Katsayılar (c)	91

Bu çalışmada bir konuşmacıya ait ÇKA modelinin eğitiminde o konuşmacının dışında kalan diğer tüm konuşmacıların konuşma verisini, ilgili konuşmacının verisi büyüklüğüne göre, VN yöntemlerinden LBG algoritması kullanılarak sıkıştırılmıştır. VN algoritması; ÇKA modelinin bir konuşmacıya ait olmayan verilerinin "0" olarak

seçilmesi sonucu; ağın sürekli yasaklayıcı vektörlere göre eğitilmesi yerine hem yasaklayıcı hem de o konuşmacının vektörlerini ifade eden uyarıcı vektörlerin sayısını eşitleyerek, ağın her iki durum için ortaya koyacağı yaklaşımı dengelemek için kullanılmıştır.

Daha sonra sıkıştırılan bu verilerin (antikonuşmacılar) ağ çıkışında istenen çıkış değerleri (hedef) "0", ilgili konuşmacının verisi ise "1" olarak seçilmiştir. ÇKA Ağı, toplam karesel hata 0.001 oluncaya kadar eğitilmiştir. Öğrenme oranı η, 0.01 seçilmiştir.

5.2. ÇKA Mimarisinin Sistem Verimine Etkisi

Bu çalışmada, ÇKA mimarisinin sistem verimine etkisi araştırılmıştır. ÇKA sınıflandırıcılarının, saklı katmandaki işlem birimi sayısı, sırasıyla 16, 32 ve 64 (net_16, net_32, net_64) olarak değiştirilerek her farklı mimari için ağın eğitimi toplam karesel hata 0.001 oluncaya kadar eğitilmiştir. Önceki çalışmadan farklı olarak antikonuşmacıların çıkış değerleri "-1" seçilmiştir. Çıkış değerlerinin "-1" seçilmesinin nedeni; çıkışın "0" seçilmesi durumunda ağın kararsız bir konuma gelmesini ve ağırlıkların yerel bir minimumda saplanıp kalmasını engellemektir. Sınıflandırıcılarda birden fazla saklı katman kullanılmasının sistem verimini arttırmadığı görülmüştür. Bu sınıflandırıcılar, konuşmacı saptama sistemine uygulanarak elde edilen sonuçlar Tablo 5.2'de gösterilmiştir.

Daha sonra en iyi sonucu veren sınıflandırıcı (net_16) konuşmacı doğrulama sistemine uygulanmıştır. Şekil 5.2'de görüldüğü gibi Eşit Hata Oranı-EHO %3 olarak bulunmuştur (İnal 2001).

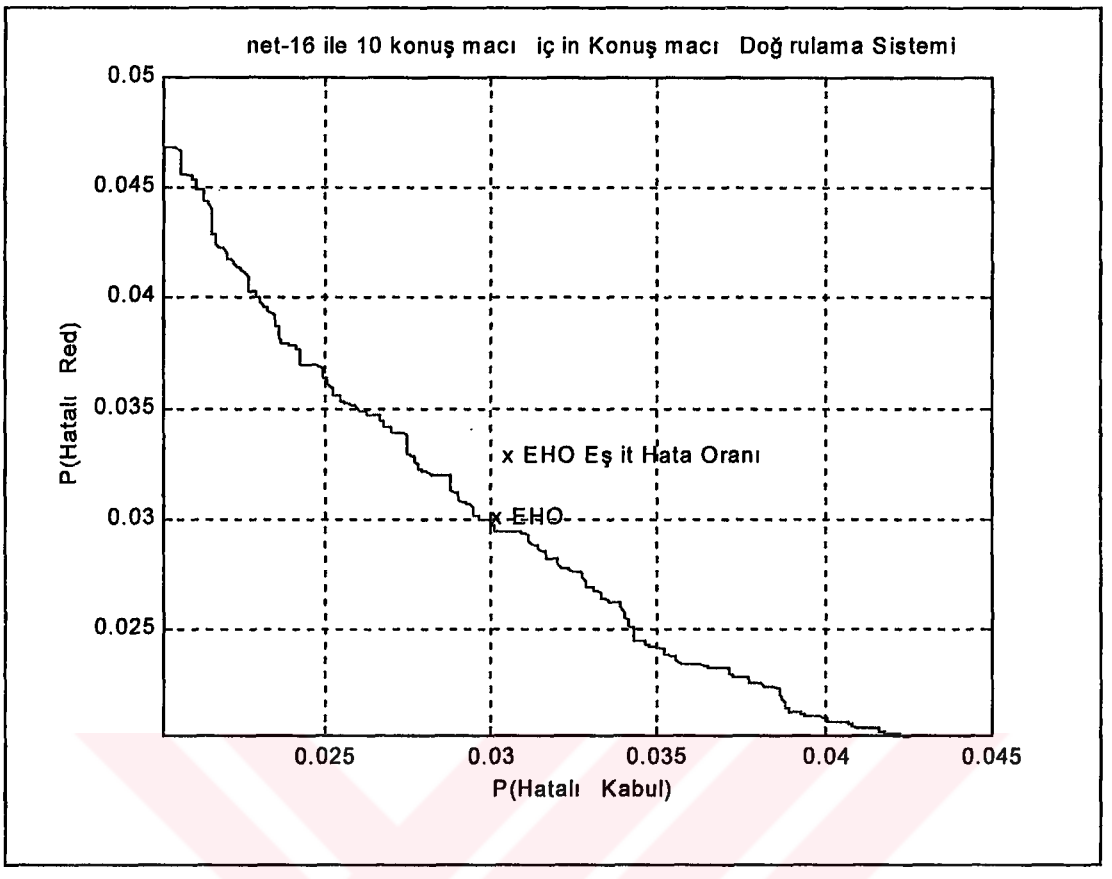
Tablo 5.2 incelendiğinde saklı katmanlarda kullanılan işlem birimleri sayısına göre ortalama verimler birbirine çok yakın çıkmıştır. Buna göre, net_16 mimarisinin verimi 0.04 kadar net_64 mimarisinden iyi olması; saklı katmanda daha az işlem birimi kullanıldığında verimin iyileştirilebileceği düşüncesini doğurmuştur. Bu nedenle tekrar bir çalışma yapılmıştır. Bu çalışmaya ilişkin sonuçlar Tablo 5.3'te gösterilmiştir. Çalışmada, saklı katmanda sırasıyla 8, 16, 24, 32 ve 64 işlem birimi

seçilerek, yine istenen çıkış değerleri ±1 seçilmiştir. Tasarlanan ağ modellerinin her birinin eğitimi 1000 epok için tekrarlanmıştır.

ÇKA ağlarında sigmoid transfer fonksiyonu kullandığımızdan işlem birimleri çıkışları daima ±1 aralığında sınırlıdır. İstenen çıkış değerleri de ±1 seçilirse, daima kalıcı bir hata bandı ±1 civarında oluşabilir. Ağın o anda hesaplanan çıkışı büyük olasılıkla hiçbir zaman ±1 olmayacağından, ağ osilasyona girip yerel bir minimumda salınım yapabilir. Bu nedenle Tablo 5.3'ün en son sütununda *net_24_09* adlı ağın istenen çıkış değerleri ±0.9 olarak seçilerek yine diğer ağlar gibi eğitilmiştir. Bu ağ için işlem birimi sayısının 24 seçilmesinin nedeni; giriş katmanındaki işlem birimi sayısı 12 olduğundan yapılan çalışmalarda genellikle "saklı katmandaki işlem birimi sayısı, giriş katmanındaki işlem birimi sayısının iki katı kadar seçilir" düşüncesidir. Bu çalışmaya ilişkin en iyi sonucu veren net_8 ağına ilişkin konuşmacı doğrulama işlemi Şekil 5.3'te görülmektedir. Konuşmacı doğrulama işlemlerinin gösterildiği şekillerde, P(Hatalı Kabul) ve P(Hatalı Red) eksenleri yüzdelik hataları göstermektedir. Tablo 5.2 ve 5.3'teki sonuçlar saklı katmandaki işlem birimi sayısının deneysel yolla belirlenebileceğini göstermektedir.

Tablo 5.2. ÇKA Sınıflandırıcılarının Konuşmacı Saptama Sistem Verimleri.

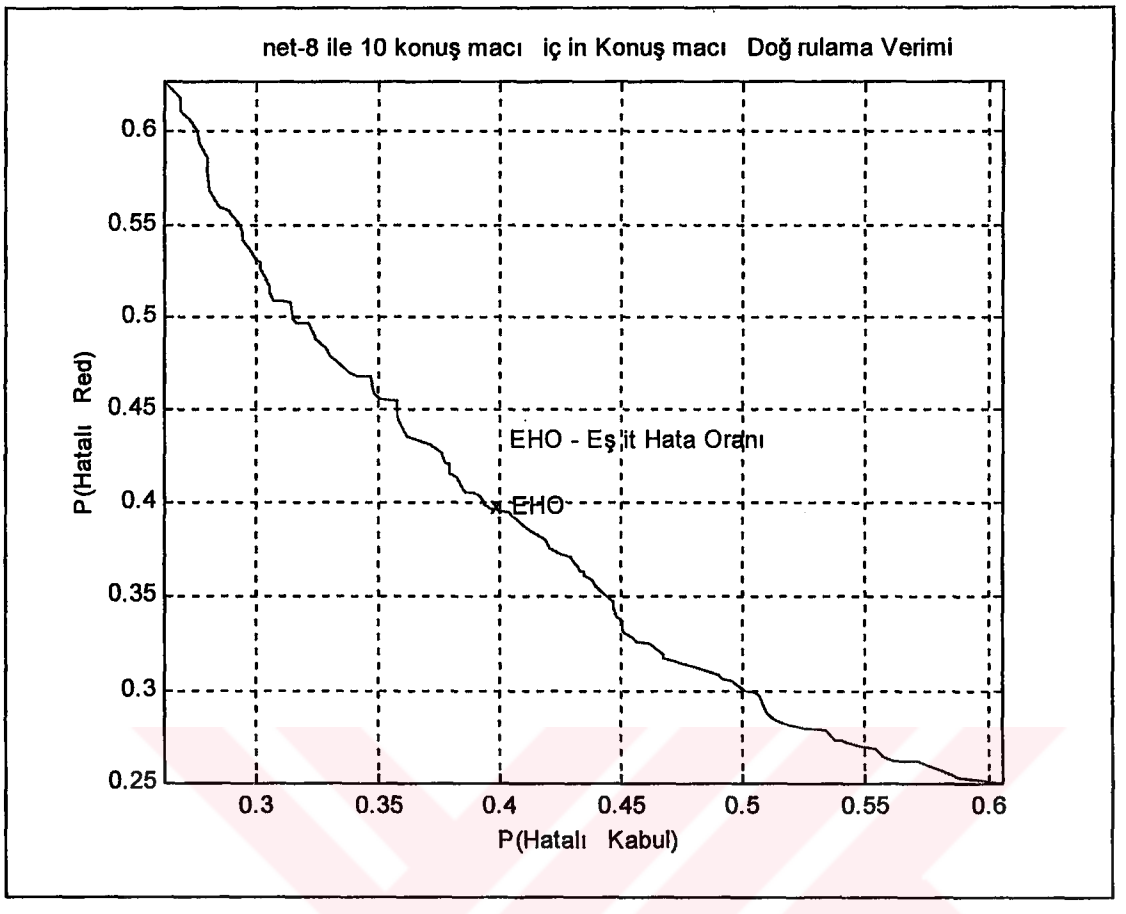
Konuşmacı	**net_16**	**net_32**	**net_64**
Melih	90,708	88,895	88,193
Alper	96,031	92,374	93,491
Celal	85,918	92,382	86,328
Faruk	93,394	89,108	91,367
Namık	92,255	92,003	92,197
Hüseyin	93,209	95,533	95,692
Erhan	90,302	87,105	89,430
Melek	97,982	95,587	94,888
Nuran	91,480	92,207	95,450
Seval	91,266	94,589	95,154
Ortalama	**92,255**	**91,978**	**92,219**



Şekil 5.2. ÇKA Sınıflandırıcısının Konuşmacı Doğrulama Sistemindeki Verimi

Tablo 5.3. ÇKA Sınıflandırıcıların Farklı Mimarilere Göre Sistem Verimleri.

Konuşmacı	**net_8**	**net_16**	**net_24**	**net_32**	**net_64**	***net_24_09***
Alper	87.952	82.530	81.928	84.337	72.892	*68.072*
Celal	71.311	63.115	55.738	51.639	57.377	*79.508*
Erhan	66.860	67.442	58.721	65.116	71.512	*62.791*
Faruk	81.757	70.270	68.919	56.757	80.405	*70.946*
Hüseyin	88.953	84.302	76.744	83.140	66.860	*69.767*
Melek	82.237	87.500	76.974	85.526	72.368	*78.947*
Melih	74.561	67.544	74.561	68.421	73.684	*69.298*
Namık	80.723	75.287	65.663	63.855	74.699	*74.096*
Nuran	82.759	74.713	74.138	70.690	78.736	*75.287*
Seval	83.908	77.108	66.092	82.184	82.759	*68.391*
Ortalama	**80.102**	**74.981**	**69.948**	**71.167**	**73.129**	**71.710**



Şekil 5.3. net_8 ÇKA Sınıflandırıcısının Konuşmacı Doğrulama Verimi

BÖLÜM 6. SOM TABANLI KONUŞMACI TANIMA UYGULAMALARI

Bu bölümde, SOM YSA ağı tabanlı, metinden bağımsız-metne bağlı ve kapalı-açık set konuşmacı tanıma uygulamaları ve bu uygulamalara ilişkin sonuçlar açıklanacaktır. Ayrıca, Türkçe metne bağlı kapalı set konuşmacı saptama sistemi oluşturularak, gerçek zamanda test edilecektir. Yapılan konuşmacı saptama uygulamalarında, Şekil 4.14'te açıklanan karma ağ yapısı kullanılmıştır.

6.1. SOM Sınıflandırıcısının Kullanıldığı Türkçe Metne Bağlı Kapalı Set Konuşmacı Tanıma Uygulamaları

Bu uygulamaya ilişkin veriler, Bölüm 4'te özellik çıkartım aşaması anlatılan, Türkçe konuşmacı veritabanından alınmış ses örnekleri kullanılarak hazırlanmıştır. DÖK tabanlı 12 inci dereceden kepstral katsayılar özellik çıkartım vektörleri kullanılarak eğitim ve test vektörleri oluşturulmuştur. Bu alt bölüm; eğiticisiz öğrenme algoritmalarından Yarışmacı Öğrenme Algoritmasının kullanıldığı SOM YSA sınıflandırıcılarına ilişkin çalışmaları içerecektir. SOM ağlarının eğitiminde, hedeflenen toplam karesel hata 0.0001 seçilmiştir. Belirlenen epok sayısına göre eğitim yapıldığından, hedeflenen toplam karesel hataya erişilmeden de eğitim bitirilebilir ya da hedeflenen toplam karesel hataya kadar sürdürülebilir. Başlangıçta öğrenme oranı η, 0.9 seçilmiştir. SOM ağlarının eğitim öncesi işlem birimlerinin konumları Şekil 4.13 a)'da görüldüğü gibi üçgen konumlarda seçilmiştir.

6.1.1. SOM sınıflandırıcısının kullanıldığı konuşmacı tanıma uygulamaları

Bu çalışmada, her kullanıcı için ayrı ayrı SOM ağları oluşturulmuş olup, ağın eğitiminde işlem birimleri iki boyutlu bir düzende 10X10 şeklinde seçilmiştir. Ağın eğitimi tüm konuşmacı vektörleri için 5000 kez tekrarlanmıştır. Tablo 6.1'de her konuşmacı için ayrı ayrı oluşturulan SOM sınıflandırıcıları kullanılarak konuşmacı saptama uygulamalarına ilişkin, her konuşmacının sınıflandırılma verimi görülmektedir. Tablo 6.2'de de yine aynı konuşmacılara ait konuşmacı doğrulama sonuçları verilmiştir. Tablo 6.1 incelendiğinde, konuşmacılara ait konuşmacı saptama

verimlerinin ortalaması %97.455 olduğu görülmektedir. Tablo 6.2'ye göre Hatalı Kabul oranlarının Hatalı Reddetme oranlarına göre daha yüksek olduğu görülmektedir. Bunun nedeni, konuşmacılara ait sınıflandırıcılar o konuşmacının vektörlerini diğer konuşmacıların vektörlerine göre daha iyi öğrenmiş olmalarıdır.

Tablo 6.1. Her Konuşmacı için SOM Ağı Oluşturulmuş Konuşmacı Saptama Verimi

Konuşmacılar	**Sınıflandırıcı Tanıma Verimi (%)**
Alper	98.810
Celal	97.541
Erhan	96.023
Faruk	99.342
Hüseyin	97.159
Melek	97.403
Melih	94.737
Namık	94.048
Nuran	100.000
Seval	99.432
Ortalama	**97.455**

Tablo 6.2. Her Konuşmacı için SOM Ağı Oluşturulmuş Sınıflandırıcılara ait Konuşmacı Doğrulama Sonuçları

Konuşmacılar	**Hatalı Kabul (%)**	**Hatalı Red (%)**
Alper	5.234	3.196
Celal	4.782	2.790
Erhan	5.625	3.452
Faruk	3.562	4.681
Hüseyin	6.142	1.147
Melek	5.526	3.274
Melih	6.432	2.125
Namık	3.217	3.837
Nuran	3.347	2.845
Seval	5.326	2.693

6.1.2. Tüm konuşmacılar için bir tek SOM ağının kullanıldığı Türkçe metne bağlı kapalı set konuşmacı tanıma uygulaması

Önceki çalışmada her konuşmacı için ayrı bir SOM sınıflandırıcı ağı tanımlanırken, tek bir ağ kullanılarak bütün konuşmacı seti için sınıflandırıcı verimi araştırılmak üzere, iki boyutlu 20X20 işlem birimi yapısındaki SOM ağı, 10000 iterasyon için eğitilerek, bu çalışma yapılmıştır. Çalışmaya ilişkin sonuçlar Tablo 6.3'te gösterilmiştir.

Tablo 6.3. Tüm Konuşmacılar için Oluşturulmuş tek bir SOM Ağı Sınıflandırıcısına ait Konuşmacı Saptama Verimi

Konuşmacılar	**Sınıflandırıcı Tanıma Verimi (%)**
Alper	81.033
Celal	81.870
Erhan	81.837
Faruk	84.893
Hüseyin	82.506
Melek	85.269
Melih	91.120
Namık	85.787
Nuran	83.109
Seval	82.861
Ortalama	**84.029**

İşlem birimi sayısı 4 katına ve iterasyon sayısı da 2 katına çıkarıldığında tek bir SOM ağının, 10 konuşmacının yer aldığı eğitim seti için hiç de önemsenmeyecek sonuçlar verdiği Tablo 6.3'teki verilere göre söylenebilir. Sonuçların bu denli iyi olması uygulamanın metne bağlı kapalı set konuşmacı tanıma uygulaması oluşu ve konuşmacı sayısının az oluşundan kaynaklanmaktadır.

6.2. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Kapalı Set Konuşmacı Tanıma Alanına Uygulanması

Bu bölümde TIMIT veritabanını kullanarak, çeşitli metinden bağımsız kapalı set konuşmacı saptama ve doğrulama uygulamaları incelenecektir. Bu uygulamaya ilişkin veriler, Bölüm 4.1'de özellik çıkartım aşaması açıklanan aşamalar göre hazırlanmıştır. Bu aşamalar, maddeler halinde sıralanmıştır:

- Konuşma işaretleri, transfer fonksiyonu Eşitlik 4.1'deki gibi birinci dereceli sayısal süzgeçten geçirilmiştir. Süzgeç katsayısı 0.98 seçilmiştir.

- Konuşma işaretinin her 330'luk bölütü için Hamming pencereleme yöntemi kullanılmıştır. Ayrıca her 330'luk bölüt için uygulanan pencereleme işlemi, pencereler arası geçişte veri kaybını önlemek için; her 110 örnekte bir tekrarlanmıştır.

- Bu çalışmada, özellik çıkartım yöntemlerinden DÖK incelemesi kullanılmış ve 12 inci dereceden DÖK tabanlı doğrusal öngörülü a parametre takımı konuşma işaretinin her bölütü için hesaplanmıştır.

- DÖK incelemesi sonucunda elde edilen a parametre takımından kepstral c parametre takımı Eşitlik 2.5'e göre türetilmiştir.

DÖK tabanlı 12 inci dereceden kepstral katsayılar özellik çıkartım vektörleri kullanılarak SOM ağlarının eğitim ve test vektörleri oluşturulmuştur. SOM ağlarının eğitiminde, öğrenme oranı η, 0.9 seçilmiştir. SOM ağlarının eğitim öncesi işlem birimlerinin konumları Şekil 4.13 (a)'da görüldüğü gibi üçgen konumlarda seçilmiştir.

6.2.1. TIMIT veritabanı ile SOM ağı sınıflandırıcısının konuşmacı saptama alanına uygulanması

Bu çalışmada, TIMIT veritabanını kullanarak metinden bağımsız kapalı set konuşmacı saptama uygulamaları yapılmıştır. Kapalı set konuşmacı saptama uygulaması sırasıyla 5, 10 ve 20 konuşmacı için gerçekleştirilmiştir. Öncelikle tüm

konuşmacılara ait tek bir SOM ağı iki boyutlu 25X25 işlem birimine sahip ağ mimarisi ile 10000 epok için eğitilmiştir. Bu çalışmaya ilişkin sonuçlar Tablo 6.4 (a) (b) (c)'de gösterilmiştir. Sonuçlara bakıldığında tek bir ağın tüm konuşmacılar için iyi bir sınıflandırıcı olduğu söylenemez.

Bir başka çalışmada ise her konuşmacı için ayrı ayrı SOM YSA'ları tasarlanarak, eğitim süresinin etkisini araştırmak üzere yine 5,10 ve 20 konuşmacı için, iki boyutlu 20X20 işlem birimine sahip ağ mimarisi ile 10000 epok için eğitilmiştir. Bu çalışmaya ait sonuçlar Tablo 6.5 (a) (b) (c)'de görülmektedir. Tablo incelendiğinde 5 ve 10 konuşmacı için verim birbirine yakın iken konuşmacı sayısı 20 olduğunda verim %1 oranında düşmektedir. Bu durum, SOM ağının, konuşmacı sayısının fazla olduğu tanıma sistemlerinde de rahatlıkla kullanılabileceğini göstermektedir.

Eğitim süresinin etkisini araştırmak üzere yine aynı ağ mimarisi şimdi de 5000 epok için eğitilmiştir. Bu çalışmaya ilişkin sonuçlar Tablo 6.6 (a) (b) (c)'de gösterilmiştir. Tablo 6.5 ve 6.6 karşılaştırıldığında, epok sayısı yarıya indirildiğinde ortalama konuşmacı verimleri %1 oranında düştüğü gözlemlenmiştir. %1'lik bu düşüşün eğitim süresinin kısa tutulması açısından mantıklı olduğu düşünülmektedir.

Ağ mimarisinde işlem birimi sayısının etkisini araştırmak üzere 10000 epok için iki boyutlu ağ mimarisi 10X10 işlem birimi kullanılarak eğitilmiştir. Bu çalışmaya ilişkin sonuçlar Tablo 6.7 (a) (b) (c)'de gösterilmiştir. Tablo 6.5 ve Tablo 6.7 karşılaştırıldığında, işlem birimi sayısı yarıya indirildiğinde sistem veriminin %6-7 oranında düşmesi, dayanıklı bir konuşmacı tanıma sistemi için kabul edilemez oranlardır. Bu nedenle TIMIT Veritabanı ile yapılan metinden bağımsız kapalı set konuşmacı tanıma uygulaması için SOM ağının 20X20 işlem birimi ile 5000 epok için eğitilmesinin yeterli olacağı görülmüştür. Tablolarda konuşmacı etiketleri olarak görünen "F" bayan konuşmacıları ve "M" ise erkek konuşmacıları ifade etmektedir.

Tablo 6.4. Tüm konuşmacılara ait tek bir SOM Ağının kullanıldığı konuşmacı saptama uygulamaları

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	52.573
F2	50.781
M1	40.355
M2	38.521
M3	46.911
Ortalama	**45.828**
(a) 5 Konuşmacı için	

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	51.297
F2	49.596
F3	50.260
F4	33.953
F5	41.585
M1	42.520
M2	38.340
M3	50.145
M4	48.314
M5	42.626
Ortalama	**44.864**
(b) 10 Konuşmacı için	

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	54.147
F2	50.176
F3	54.044
F4	40.683
F5	42.640
F6	52.263
F7	32.499
F8	37.001
F9	45.296
F10	45.146
M1	41.694
M2	38.730
M3	45.222
M4	44.710
M5	38.860
M6	36.933
M7	42.321
M8	39.606
M9	37.465
M10	42.800
Ortalama	**43.112**
(c) 20 Konuşmacı için	

Tablo 6.5. Her konuşmacı için ayrı SOM Ağının kullanıldığı konuşmacı saptama uygulamaları

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	99.476
F2	95.868
M1	95.050
M2	96.233
M3	100.000
Ortalama	**97.325**
(a) 5 Konuşmacı için	

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	99.476
F2	98.585
F3	99.000
F4	97.527
F5	99.194
M1	97.512
M2	97.368
M3	96.465
M4	98.429
M5	94.902
Ortalama	**97.846**
(b) 10 Konuşmacı için	

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	99.476
F2	100.000
F3	99.500
F4	97.527
F5	97.177
F6	100.000
F7	94.788
F8	99.286
F9	90.476
F10	97.285
M1	95.025
M2	96.992
M3	90.909
M4	94.764
M5	92.941
M6	91.406
M7	98.222
M8	96.578
M9	97.154
M10	96.596
Ortalama	**96.305**
(c) 20 Konuşmacı için	

Tablo 6.6. Her konuşmacı için ayrı ayrı tanımlanmış, 5000 epok ile 20X20 işlem birimine sahip ağ mimarisiyle eğitilmiş SOM Ağının kullanıldığı konuşmacı saptama uygulamaları

Konuşmacı Etiketleri	**Saptama Verimi (%)**
F1	99.476
F2	97.170
M1	92.537
M2	93.609
M3	98.990
Ortalama	**96.356**
(a) 5 Konuşmacı için	

Konuşmacı Etiketleri	**Saptama Verimi (%)**
F1	97.382
F2	99.057
F3	99.500
F4	95.406
F5	98.790
M1	98.507
M2	94.737
M3	96.465
M4	98.429
M5	90.196
Ortalama	**96.847**
(b) 10 Konuşmacı için	

Konuşmacı Etiketleri	**Saptama Verimi (%)**
F1	97.906
F2	93.396
F3	98.500
F4	95.053
F5	98.387
F6	96.943
F7	95.765
F8	97.500
F9	88.312
F10	96.833
M1	91.045
M2	96.992
M3	90.404
M4	91.623
M5	96.863
M6	90.625
M7	98.667
M8	95.817
M9	98.780
M10	97.447
Ortalama	**95.343**
(c) 20 Konuşmacı için	

Tablo 6.7. Her konuşmacı için ayrı ayrı tanımlanmış, 10000 epok ile 10X10 işlem birimine sahip ağ mimarisiyle eğitilmiş SOM Ağının kullanıldığı konuşmacı saptama uygulamaları

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	91.099
F2	95.283
M1	92.557
M2	93.842
M3	94.364
Ortalama	**93.429**
(a) 5 Konuşmacı için	

Konuşmacı Etiketleri	Saptama Verimi (%)
F1	93.717
F2	99.057
F3	95.000
F4	93.640
F5	94.758
M1	93.532
M2	90.977
M3	91.919
M4	92.670
M5	83.529
Ortalama	**92.880**
(b) 10 Konuşmacı için	

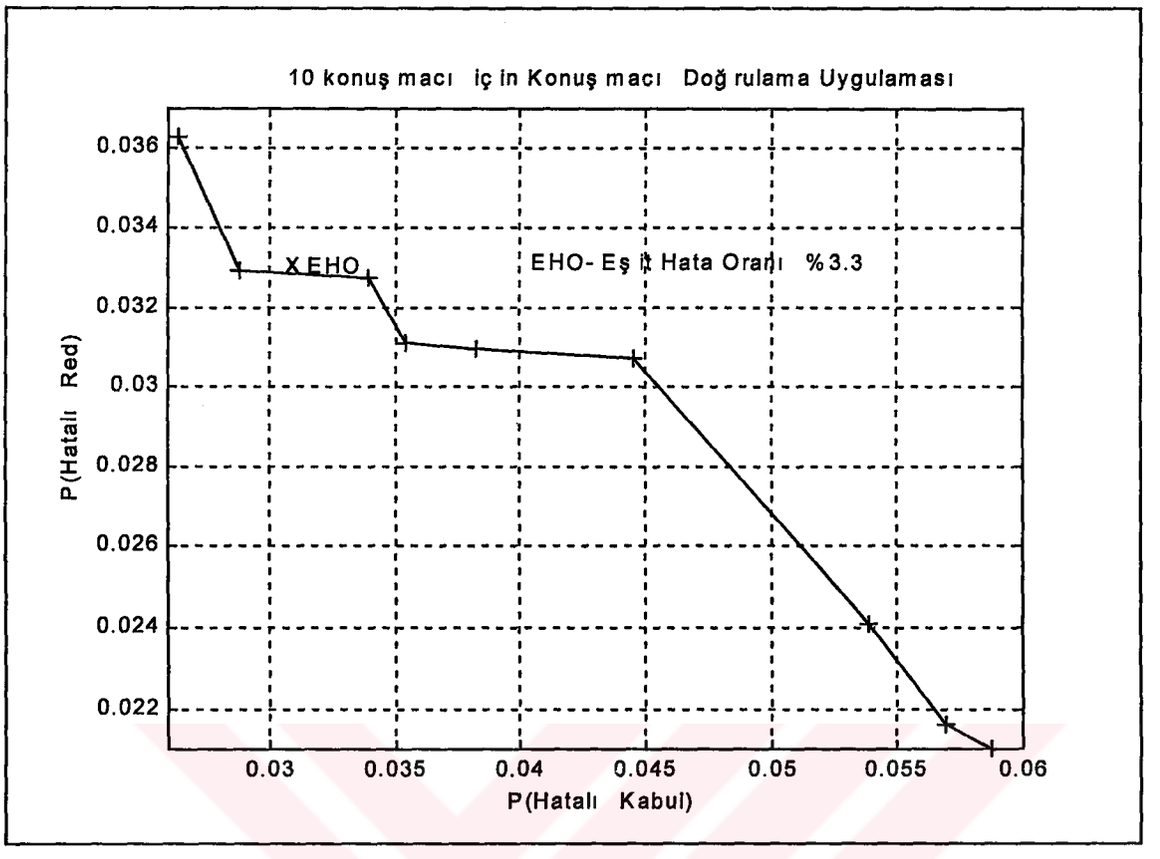
Konuşmacı Etiketleri	Saptama Verimi (%)
F1	84.293
F2	95.755
F3	86.500
F4	86.926
F5	92.742
F6	95.633
F7	88.599
F8	93.214
F9	90.043
F10	89.140
M1	93.035
M2	91.353
M3	90.909
M4	86.911
M5	89.020
M6	82.813
M7	91.111
M8	93.916
M9	92.276
M10	92.340
Ortalama	**90.327**
(c) 20 Konuşmacı için	

6.2.2. TIMIT veritabanı ile SOM ağı sınıflandırıcısının konuşmacı doğrulama alanına uygulanması

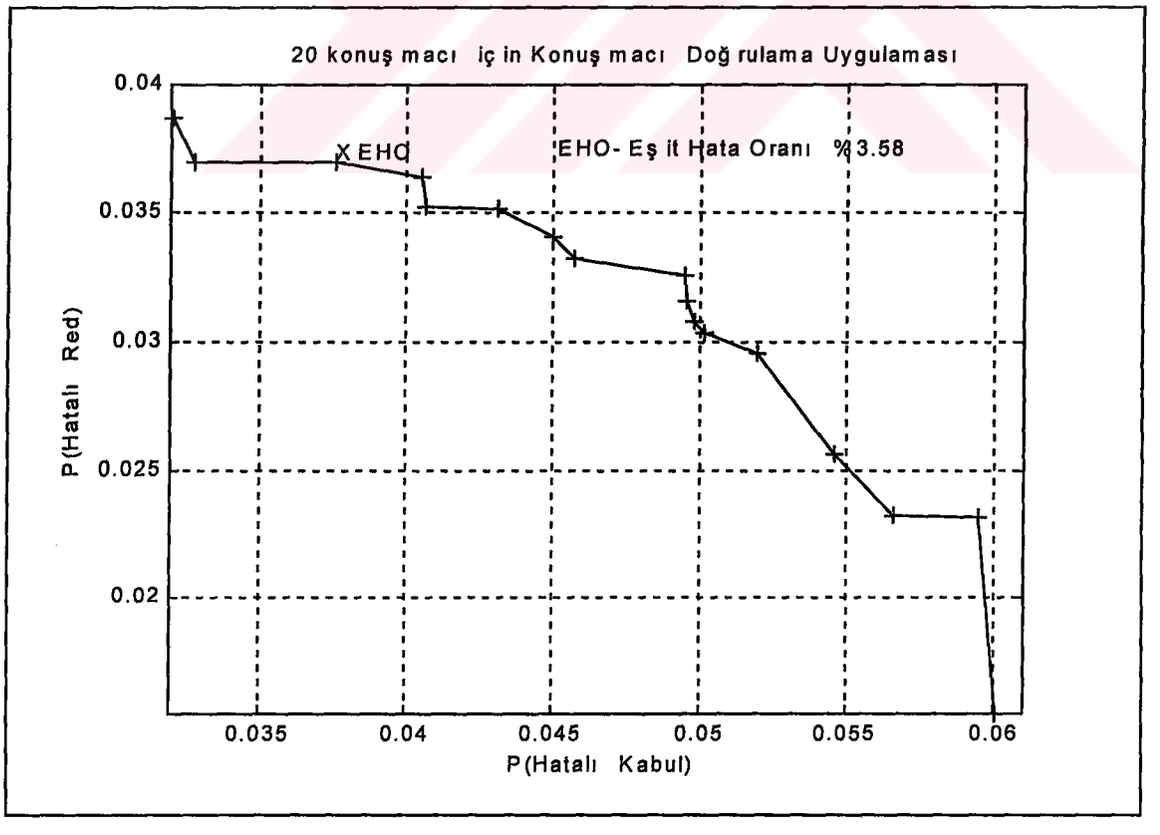
Bu çalışmada yine TIMIT veritabanı kullanılarak, 10 ve 20 konuşmacı için metinden bağımsız kapalı set konuşmacı doğrulama uygulamaları yapılmıştır. İlk doğrulama uygulamasında 10 konuşmacı için iki boyutlu ağ mimarisine sahip 10X10 işlem birimi kullanılarak 5000 epok için eğitilmiştir. Bu uygulamada konuşmacılara ilişkin sonuçlar Tablo 6.8'de gösterilmiştir. Eşit Hata Oranı ise Şekil 6.1'de görüldüğü gibi %3.3 bulunmuştur. Daha sonra 20 konuşmacı için iki boyutlu ağ mimarisine sahip 20X20 işlem birimi kullanılarak 10000 epok için eğitilmiştir. Bu uygulamaya ilişkin sonuçlar Tablo 6.9'da gösterilmiştir. Eşit Hata Oranı ise Şekil 6.2'de görüldüğü gibi %3.58 bulunmuştur.

Tablo 6.8. 10 konuşmacı için Konuşmacı Doğrulama Uygulaması

Konuşmacı Etiketleri	Hatalı Kabul (%)	Hatalı Red (%)
F1	5.90	2.41
F2	5.70	3.27
F3	5.87	2.08
F4	3.83	3.11
F5	5.39	3.63
M1	4.45	3.07
M2	3.54	2.16
M3	2.89	3.09
M4	3.39	2.10
M5	2.64	3.29



Şekil 6.1. 10 konuşmacı için Konuşmacı Doğrulama Uygulaması



Şekil 6.2. 20 konuşmacı için Konuşmacı Doğrulama Uygulaması

Tablo 6.9. 20 konuşmacı için Konuşmacı Doğrulama Uygulaması

Konuşmacı Etiketleri	Hatalı Kabul (%)	Hatalı Red (%)
F1	6.14	3.52
F2	5.95	0.97
F3	6.18	1.54
F4	4.57	2.33
F5	6.03	3.15
F6	5.66	3.41
F7	4.31	3.32
F8	4.96	0.96
F9	3.28	2.95
F10	4.98	3.52
M1	5.19	2.32
M2	4.05	3.25
M3	3.75	3.08
M4	4.06	3.03
M5	3.21	3.87
M6	5.01	1.07
M7	6.01	3.70
M8	4.95	3.70
M9	4.50	2.56
M10	5.46	3.64

6.3. TIMIT Veritabanı ile SOM Ağı Sınıflandırıcısının Metinden Bağımsız Açık Set Konuşmacı Tanıma Uygulamaları

Bu bölümde TIMIT veritabanı kullanılarak, çeşitli, metinden bağımsız açık set konuşmacı saptama ve doğrulama uygulamaları yapılmış ve bu uygulamalara ilişkin sonuçlar değerlendirilmiştir. Konuşma verilerinin özellik çıkartım aşamaları Bölüm 6.2'de açıklanmıştır.

6.3.1. TIMIT ve SOM ağı ile açık set konuşmacı saptama uygulaması

TIMIT veritabanını kullanarak metinden bağımsız açık set konuşmacı saptama uygulaması yapılmıştır. Açık set konuşmacı saptama uygulaması 20 konuşmacı için gerçekleştirilmiştir. SOM ağı iki boyutlu 20X20 işlem birimine sahip ağ mimarisi ile 10000 epok için eğitilmiştir. Ağın eğitiminde kullanılmayan 18 sahte konuşmacı, SOM ağlarının test aşamasında kullanılmıştır. Uygulamanın açık set konuşmacı tanıma uygulaması oluşundan dolayı, eşik değeri oranı %70 olarak tanımlanmıştır. Sonuçlar Tablo 6.10'da gösterilmiştir.

Tablo 6.10. 20 konuşmacı için Açık Set Konuşmacı Saptama Uygulaması

Konuşmacı Etiketleri	**Saptama Verimi (%)**
F1	78.825
F2	77.557
F3	79.632
F4	75.534
F5	81.129
F6	80.633
F7	75.703
F8	78.939
F9	81.530
F10	79.186
M1	76.081
M2	74.839
M3	76.705
M4	73.791
M5	87.451
M6	81.261
M7	82.492
M8	77.898
M9	77.197
M10	82.310
Ortalama	**78.935**

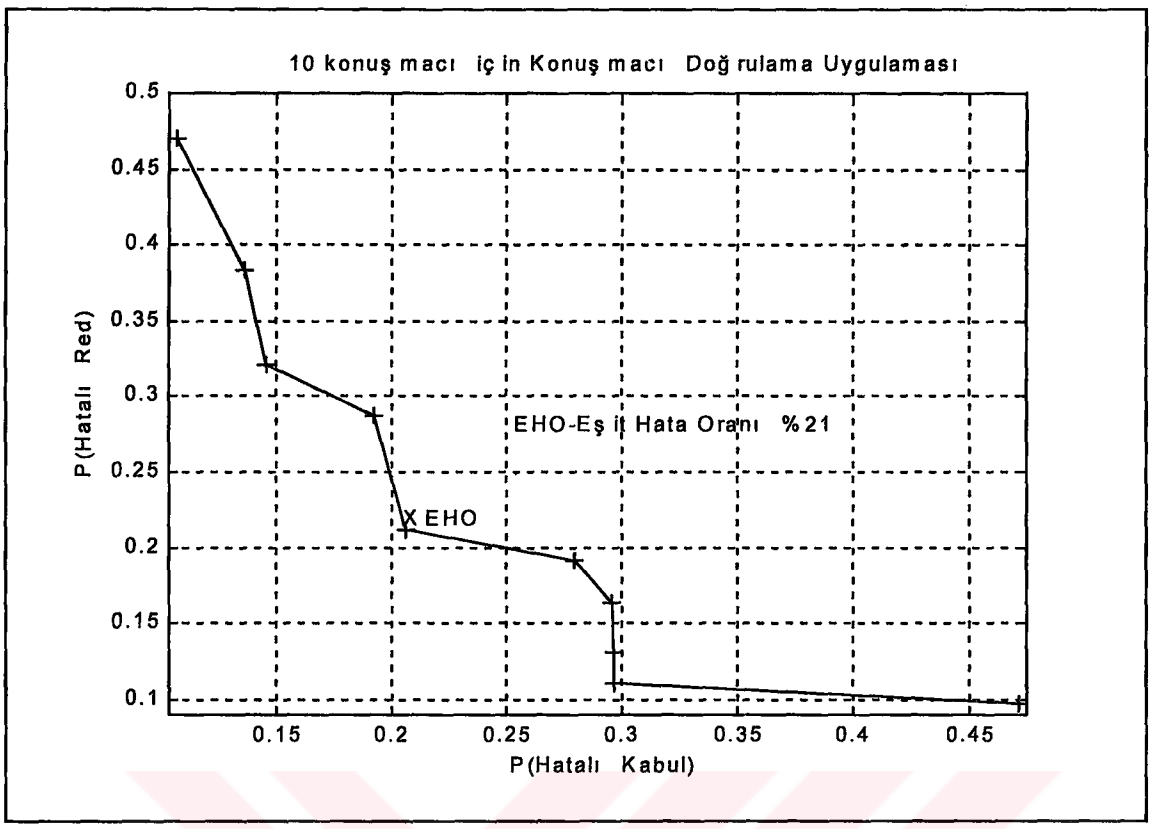
6.3.2. TIMIT ve SOM ağı ile açık set konuşmacı doğrulama uygulanması

Bu çalışmada yine TIMIT veritabanı kullanılarak, 10 ve 20 konuşmacı için metinden bağımsız açık set konuşmacı doğrulama uygulamaları yapılmıştır. İlk doğrulama uygulamasında 10 konuşmacı için iki boyutlu ağ mimarisine sahip 10X10 işlem birimi kullanılarak 5000 epok için eğitilmiştir. Bu uygulamada 10 sahte konuşmacı ağın test aşamasında kullanılmıştır.

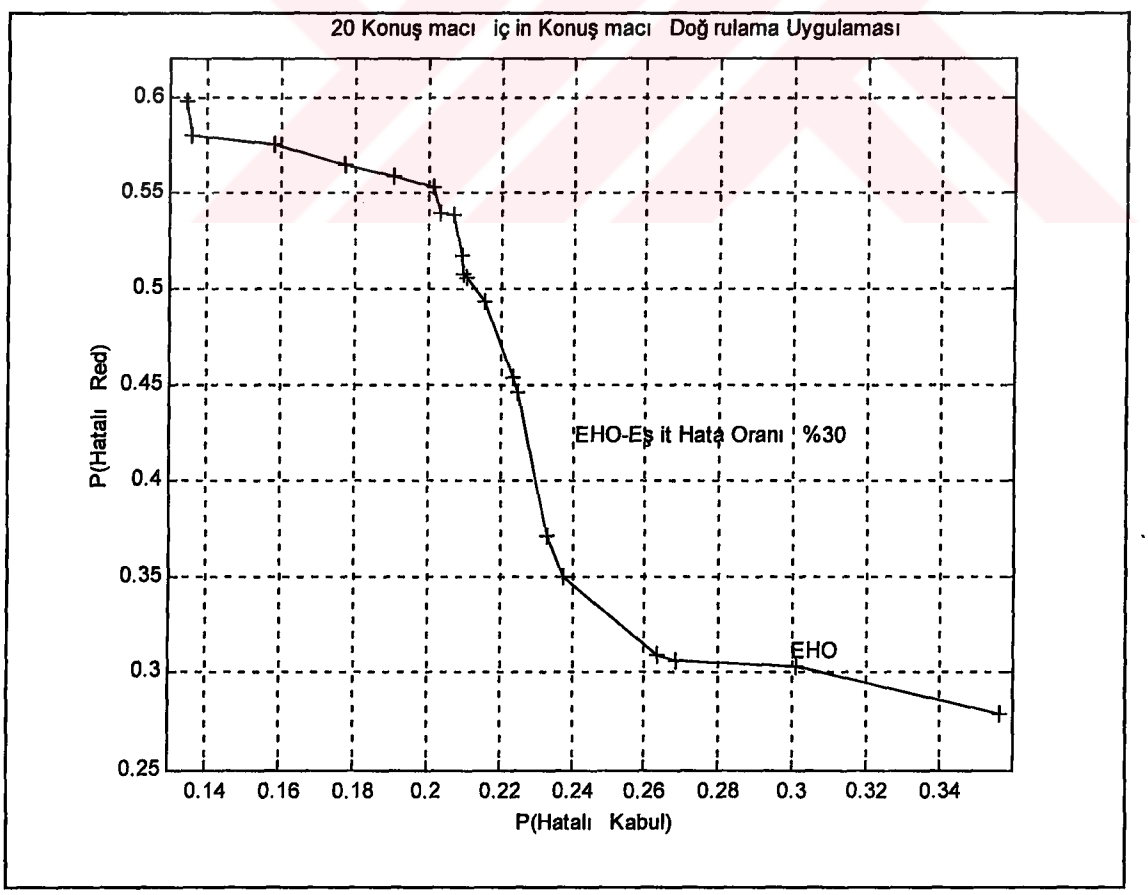
Bu uygulamaya ilişkin sonuçlar Tablo 6.11'de gösterilmiştir. Eşit Hata Oranı ise Şekil 6.3'te görüldüğü gibi %21 bulunmuştur. Daha sonra 20 konuşmacı için iki boyutlu ağ mimarisine sahip 20X20 işlem birimi kullanılarak 10000 epok için eğitilmiştir. Bu çalışmada da 18 sahte konuşmacı ağın test aşamasında kullanılmıştır. Bu uygulamaya ilişkin sonuçlar Tablo 6.12'de gösterilmiştir. Eşit Hata Oranı ise Şekil 6.4'te görüldüğü gibi %30 bulunmuştur. Açık set konuşmacı tanıma uygulaması olduğundan eşik değer %70 tanımlanmıştır.

Tablo 6.11. 10 konuşmacı için Açık Set Konuşmacı Doğrulama Uygulaması

Konuşmacı Etiketleri	Hatalı Kabul (%)	Hatalı Red (%)
F1	47.170	28.690
F2	19.242	21.181
F3	10.733	19.089
F4	13.668	11.095
F5	14.579	13.079
M1	20.626	16.385
M2	29.690	9.713
M3	29.614	32.086
M4	29.663	38.284
M5	27.962	46.942



Şekil 6.3. 10 konuşmacı için Açık Set Konuşmacı Doğrulama Uygulaması



Şekil 6.4. 20 konuşmacı için Açık Set Konuşmacı Doğrulama Uygulaması

Tablo 6.12. 20 konuşmacı için Açık Set Konuşmacı Doğrulama Uygulaması

Konuşmacı Etiketleri	Hatalı Kabul (%)	Hatalı Red (%)
F1	20.344	56.441
F2	21.550	27.938
F3	35.621	58.035
F4	26.364	57.539
F5	22.332	34.968
F6	13.596	59.788
F7	19.073	53.876
F8	13.441	53.923
F9	20.704	30.908
F10	26.827	50.753
M1	17.744	37.067
M2	30.093	50.604
M3	20.968	45.300
M4	20.921	30.573
M5	21.084	30.339
M6	20.163	51.735
M7	23.313	44.520
M8	15.821	55.320
M9	22.461	49.357
M10	23.763	55.855

TIMIT Veritabanı ile metinden bağımsız açık set konuşmacı tanıma uygulamalarında eşik değeri %70'ten küçük tanımlandığında, sahte konuşmacıların sisteme kabul edilmesinin kolaylaştığı görülmüştür. Eşik değerin %70'ten büyük seçilmesi durumunda, konuşmacı setinden olan konuşmacıların da sistem tarafından reddedilmesi mümkün olmaktadır. Bu nedenle, tanımlanacak eşik değerinin optimum bir değerde seçilmesi gerekmektedir. Bunun dışında sabit bir eşik değer seçilmesi yerine, konuşmacı setindeki her konuşmacının sisteme kabul edilme yüzdesine bağlı olarak dinamik bir eşik değeri seçimi önerilmektedir.

6.4. Türkçe Metne Bağlı Kapalı Set Konuşmacı Tanıma Sistemi

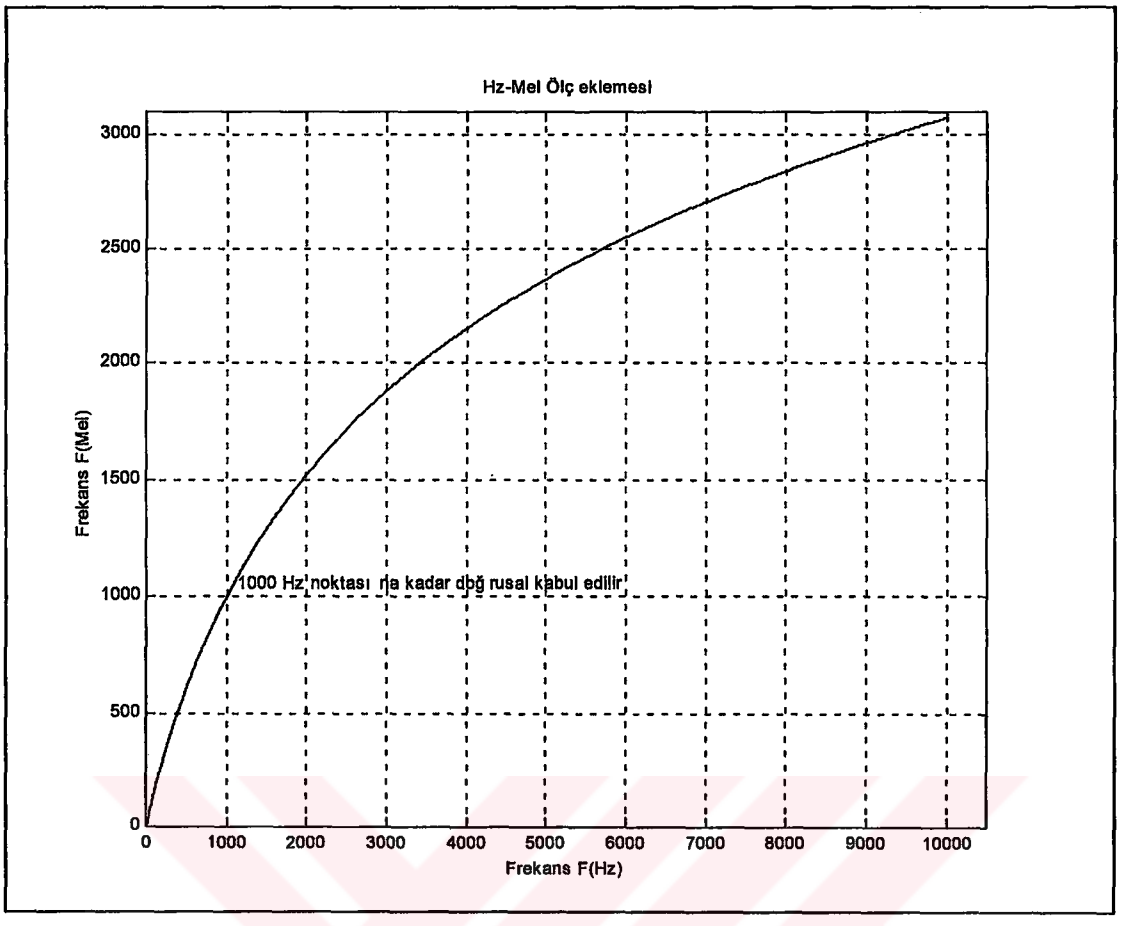
Türkçe metne bağlı kapalı set konuşmacı tanıma sisteminin her aşaması, MATLAB programında ayrı ayrı menüler şeklinde hazırlanmıştır. Türkçe veritabanı kullanılarak eğitilen SOM ağı mimarisi; iki boyutlu bir katmanda 10X10 işlem biriminden oluşmaktadır. Yaptığımız diğer çalışmalarda 12 inci dereceden kepstral katsayılar kullanılırken, bu çalışmada Mel frekans ölçekli kepstral katsayılar kullanılmıştır.

Yapılan araştırmalarda, Mel frekans ölçekli kepstral katsayıların Hertz ölçekli olanlara göre, dahi iyi sonuç verdiği görülmüştür. Mel ifadesi; sesteki frekansa bağlı değişimleri daha iyi algılayan ölçü birimi şeklinde açıklanır. Ölçekleme oranı 1 Hertz için 1.6089 Mel frekansı şeklindedir. İnsan duyu sisteminde de frekans değişiminin algılanması doğrusal olmadığı gibi, bu değişimlerin de fiziksel frekans (Hz) ile doğrusal ilişkisi yoktur. Araştırmacıların deneysel yollarla yaptığı çalışmalar sonucunda aşağıdaki eşitlikteki gibi bir yaklaşımı öngörmüşlerdir: (http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html).

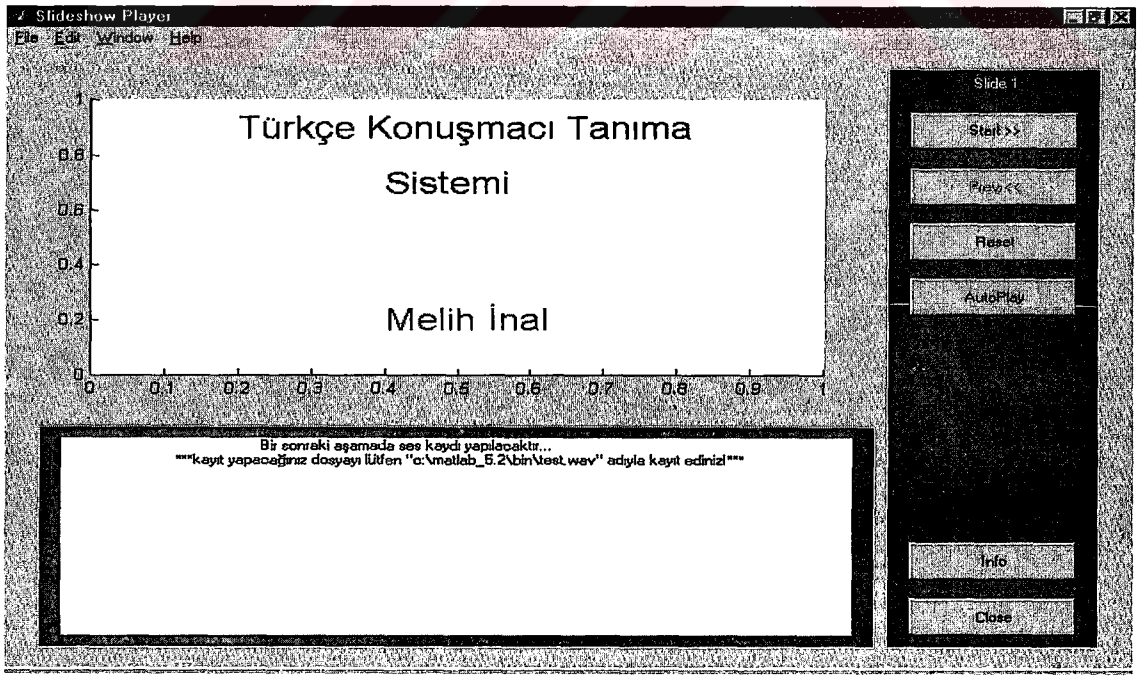
$$F_{mel} = \ln(1+F_{Hz}/700)*1127.01048 \qquad (6.1)$$

Eşitlik 6.1'e göre 0-10000 Hz frekans değerlerinin mel frekans karşılığı Şekil 6.5'te görülmektedir. Şekil 6.5'e göre 1 kHz'in altındaki ölçeklemeler yaklaşık olarak doğrusal kabul edilirken, 1 kHz'in üstündeki ölçeklemeler ise logaritmik şekilde artmaktadır.

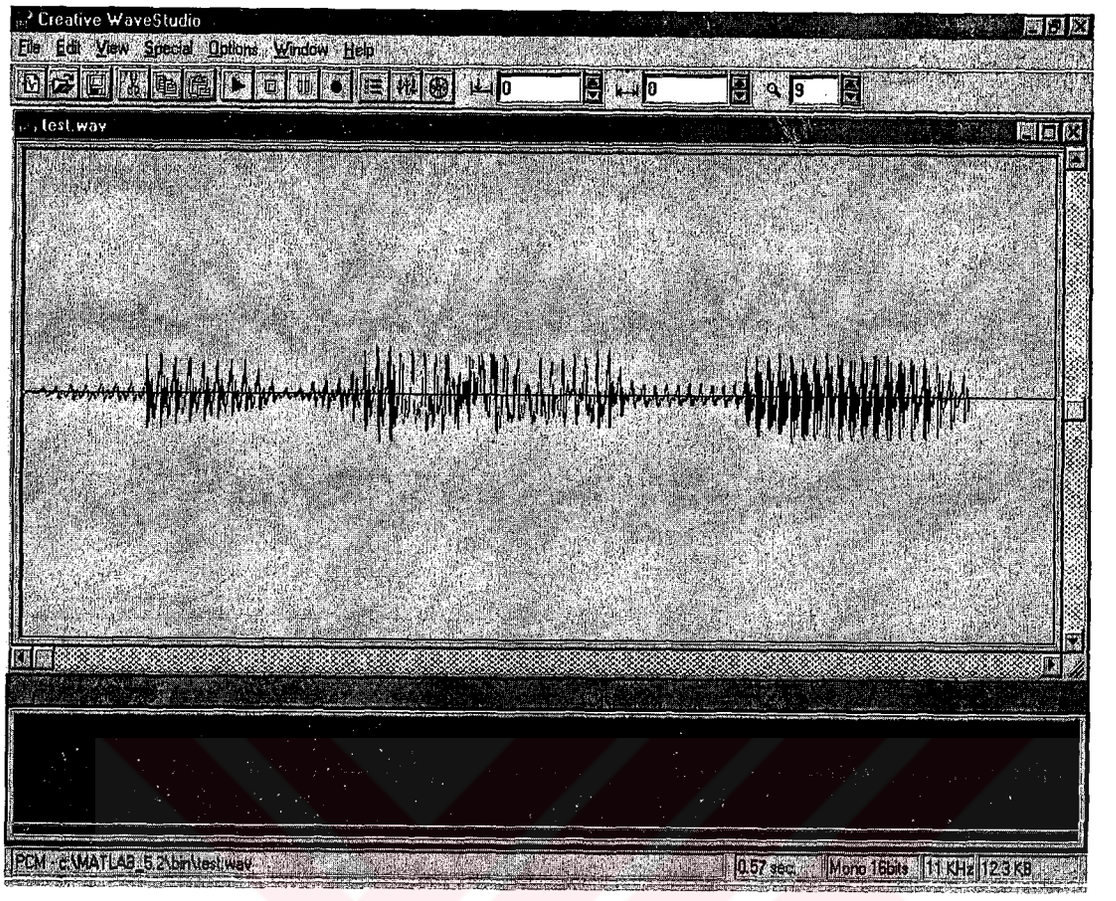
Bölüm 4.3'te açıklandığı gibi SOM ve BBM karma ağlarının eğitim tamamlandıktan sonra, gerçek zamanda ses kaydı yapılmak üzere Creative'in kendi programına geçilir. MATLAB programından Creative Programına geçiş, doğrudan Şekil 6.6'da görülen menü üzerinden sağlanmaktadır. Ses kaydının yapıldığı ekran Şekil 6.7'de görülmektedir. Daha sonra kayıt yapılarak "test.wav" dosyasında saklanan test amaçlı konuşma işaretinin özellik vektörleri üretilerek, önceden eğitimi yapılmış SOM ve BBM ağına uygulanarak, konuşmacının saptaması yapılır.



Şekil 6.5. Hz-Mel Frekans İlişkisi



Şekil 6.6. Konuşmacı Tanıma Sistemi Slayt Gösterimi



Şekil 6.7. Test amaçlı ses kaydının yapıldığı Creative WaveStudio Menüsü

Konuşmacılara ait SOM ağlarının eğitiminden sonra, eğitim vektörlerine göre elde edilen çıkışlar referans olarak seçilmiştir. Referans vektörlerin tüm konuşmacılara göre birbirleriyle olan benzerliklerini gözlemlemek üzere Tablo 6.13'teki veriler elde edilmiştir. Bu işlemin amacı, referans vektörleri birbirine benzeyen konuşmacıların belirlenmesidir. Böyle bir karşılaştırma yapmak için tüm eğitim vektörlerinin adedi eşit seçilmiştir. Tablo 6.13'ün satırlarındaki her veri o konuşmacının sütunlardaki konuşmacılara göre oluşan yüzdelik benzerliklerini göstermektedir. Tablo 6.13 incelendiğinde konuşmacıya ait benzeme yüzdesi ve kendi skoruna en yakın yüzde koyu renklerle belirtilmiştir. Her konuşmacıya ait referans vektörlerine göre, 10X10, 100 işlem birimi içinden en çok çıkış üreten işlem birimi; diğer konuşmacılarda ya hiç üretilmemekte ya da birkaç kez üretildiği görülmüştür. Böylece, her konuşmacının SOM Ağı, konuşmacıya ait vektörler için iyi bir genelleme yaptığı görülmektedir.

Tablo 6.14'te konuşmacıların test vektörlerine göre sınıflandırma yüzdeleri görülmektedir. Her konuşmacının ve kendisine en yakın konuşmacının SOM ağları sınıflandırılma yüzdeleri koyu renklerle, her satırda verilmiştir. Tablo 6.14 incelendiğinde her konuşmacının sınıflandırıcısı, diğer konuşmacıların sınıflandırıcılarına göre %50 civarında kendi test vektörlerini daha iyi sınıflandırıldıkları görülmektedir. Gerçek zamanda yapılan testlerde sınıflandırma veriminin daha düşük çıktığı fakat yine de iyi bir kayıt alındığında %50'lik verim farkının korunduğu görülmüştür. Bu durum, eğitim aşamasında mümkün olduğunca fazla ses kaydının kullanılması gerektiğini göstermiştir. Sistemi oluşturan yazılıma ait MATLAB kodu, tez çalışmasının sonunda, ek bölümde sunulmuştur.

Tablo 6.13. Konuşmacıların referans vektörlerinin birbirine benzeme yüzdeleri

	Alper	Celal	Erhan	Faruk	Hüseyin	Melek	Melih	Namık	Nuran	Seval
Alper	**100**	55	66	72	74	66	66	72	**76**	66
Celal	66	**100**	65	71	**81**	73	68	76	74	60
Erhan	71	58	**100**	70	70	68	61	71	**81**	68
Faruk	77	64	70	**100**	74	68	67	74	71	65
Hüseyin	72	66	63	67	**100**	63	66	71	**74**	66
Melek	73	67	70	70	72	**100**	72	72	**73**	66
Melih	75	65	65	71	**77**	74	**100**	66	69	60
Namık	76	67	70	73	**77**	69	61	**100**	76	70
Nuran	74	61	**74**	64	74	64	59	70	**100**	67
Seval	77	58	73	70	78	69	61	77	**80**	**100**

Tablo 6.14. Konuşmacıların test vektörlerine göre sınıflandırılma yüzdeleri

	Alper	Celal	Erhan	Faruk	Hüseyin	Melek	Melih	Namık	Nuran	Seval
Alper	**88**	**52**	48	25	44	23	28	35	26	26
Celal	39	**83**	42	27	**46**	27	26	32	28	28
Erhan	**59**	36	**93**	27	36	24	26	37	23	27
Faruk	35	35	43	**91**	32	23	29	**50**	28	37
Hüseyin	**44**	37	34	14	**95**	33	26	25	25	27
Melek	44	**56**	34	24	49	**94**	25	32	43	53
Melih	25	43	**48**	23	27	31	**83**	40	35	24
Namık	51	51	**60**	39	43	36	40	**91**	27	39
Nuran	51	43	53	33	39	39	25	32	**89**	**53**
Seval	41	**53**	34	27	39	47	15	37	37	**83**

BÖLÜM 7. SONUÇ VE ÖNERİLER

Yapılan çalışmalara ilişkin sonuçlar ve yorumlar aşağıda maddeler halinde sıralanmıştır.

1. DÖK tabanlı özellik çıkartım algoritmalarından kepstral katsayıların, ÇKA sınıflandırıcılarla yapılan çalışmada %91 verimle diğer katsayılara göre %10 civarında daha iyi sonuç verdiği görülmüştür (Bkz. Tablo 5.1) (İnal 2000). Bu nedenle yaptığımız çalışmalarda DÖK tabanlı kepstral katsayılar kullanılmıştır.

2. ÇKA sınıflandırıcısının kullanıldığı metne bağlı kapalı set konuşmacı tanıma uygulamasında, ÇKA sınıflandırıcının saklı katmandaki işlem birimleri sayısı sırasıyla 16, 32 ve 64 seçilerek sistem verimine etkisi incelenmiştir. Saklı katman işlem birimi sayısı 16 olan ağın ortalama verimi %92.255 bulunmuştur (Bkz. Tablo 5.2). Aynı ağın konuşmacı doğrulama uygulamasına göre EHO %3 olarak bulunmuştur (Bkz. Şekil 5.2) (İnal 2001). Seven, yine ÇKA tabanlı kapalı set konuşmacı tanıma verimi için; tek saklı katmanlı 32 işlem biriminin kullanıldığı ağ ile %92.5 olarak bulmuştur (Bkz. Tablo 3.13) (Seven 1997). Farklı konuşmacı seti için karşılaştırılan ÇKA mimarilerine göre çıkarılan sonuç, saklı katman işlem birimi sayısının deneysel yolla bulunmasıdır.

3. SOM sınıflandırıcısının kullanıldığı, hazırladığımız Türkçe veritabanı ile yapılan metne bağlı kapalı set konuşmacı tanıma uygulaması yapılmıştır. Konuşmacı saptama ve doğrulama uygulamalarına ilişkin sonuçlar sırasıyla; ortalama verim %97.455 (Bkz. Tablo 6.1) ve EHO %3.962 olarak bulunmuştur. Tüm konuşmacılar için ayrı ayrı tanımlanan SOM ağları yerine tek bir SOM ağı kullanılarak ortalama sistem verimi %84.029 bulunmuştur. Sonuçların bir tek SOM ağı ile bu denli iyi olması uygulamanın metne bağlı kapalı set konuşmacı saptama uygulaması oluşu ve konuşmacı sayısının az oluşuna bağlanabilir.

Türkçe konuşmacı veritabanı ile yapılan ÇKA ve SOM ağlarının kullanıldığı çalışmalara göre; SOM ağının ÇKA ağından %5 daha iyi verimle konuşmacıları

sınıflandırdığı görülmüştür. Buradan, konuşmacı tanıma sistemlerinde, SOM ağının sınıflandırıcı olarak tercih edilebileceği sonucu çıkmaktadır.

4. TIMIT veritabanı 20X20 işlem birimine sahip SOM ağ sınıflandırıcısı ile 10000 epok için eğitimi yapılan metinden bağımsız kapalı set konuşmacı tanıma uygulamasında elde edilen sonuçlar ve diğer araştırmacıların değişik modellerle elde ettiği sonuçlar Tablo 7.1'de karşılaştırılmıştır.

Tablo 7.1. TIMIT veritabanı ile yapılan değişik çalışmaların karşılaştırılması

Araştırmacı	Konuşmacı Sayısı	Yöntem	Ortalama Verim	Bkz. Tablo
Farrell v.d.	5	YAM/DYAM	%96/%100	Tablo 3.6
İnal	5	SOM	%97.325	Tablo 6.5
Farrell v.d.	10	YAM/DYAM	%92/%98	Tablo 3.6
Krishnamoorthy	10	YAM	%96.95	Tablo 3.8
İnal	10	SOM	%97.846	Tablo 6.5
Farrell v.d.	20	YAM/DYAM	%91/%96	Tablo 3.6
İnal	20	SOM	%96.305	Tablo 6.5

Tablo 7.1 incelendiğinde değişik araştırmalarla yaptığımız çalışmanın verimleri karşılaştırılmıştır. YAM/DYAM yöntemleriyle yapılan çalışmaların değişik budama seviyelerinde en iyi sonuçları, belirlenen konuşmacı sayılarına göre gösterilmiştir. Tabloya göre çalışmamızda kullandığımız SOM ağının yapılmış diğer çalışmalara alternatif olabileceği görülmüştür.

SOM ağının eğitim süresinin (Bkz. Tablo 6.6) ve işlem birimi (Bkz. Tablo 6.7) sayısının, sistem verimine etkisini incelemek üzere iki ayrı çalışma daha yapılmıştır. Yapılan çalışmalara göre eğitim süresi yarıya indirildiğinde ve işlem birimi sayısı 400'den 100'e çekildiğinde sistem verimi sırasıyla %1 ve %5 oranında düştüğü görülmüştür. %1'lik düşüş epok sayısı için kabul edilebilir bir değerdir. Çünkü, tüm konuşmacı ağlarının 10000 epok yerine 5000 epok için eğitilmesi, eğitim süresinin yarıya inmesi demektir. Daha kısa zamanda eğitimin tamamlanabilmesi de sistemin gerçek zamanda eğitilerek, kullanılabileceğini göstermektedir. İşlem birimi sayısının 400'den 100'e çekilmesi durumunda, verimin %5 düşmesi kabul edilemez. Çünkü %5'lik hata bir konuşmacı tanıma sistemi için önemli bir orandır. Bu uygulamadan

çıkardığımız sonuca göre, iki boyutlu 20X20 işlem birimi yapısındaki SOM ağlarının, 5000 epok için eğitilmesi yeterlidir.

ÇKA ağı antikonuşmacı verilerini sıkıştırmak için VN algoritmaları kullanırken, SOM ağı, sadece ilgili konuşmacının verisini kullanılarak eğitildiği için eğitim süresi hem daha kısa hem de tanıma verimi daha yüksektir. Bu durumda, gerçek zamanda SOM ağı eğitiminin yapıldığı konuşmacı tanıma uygulamaları yapmak mümkündür. Bu mümkün olmasa bile, bir Analog/Sayısal dönüştürücü kartına sahip bir mikrodenetleyici sayesinde, önceden eğitilmiş SOM ağlarının ağırlıklarının yer aldığı bir EPROM entegre kullanılarak konuşmacı tanıma sistemi oluşturmak mümkündür.

Sonuç olarak, metne bağlı ve metinden bağımsız konuşmacı tanıma uygulamalarında DÖK tabanlı kepstral katsayıların kullanıldığı SOM YSA modelinin iyi sonuç verdiği, hazırlanan Türkçe veritabanı ve TIMIT veritabanı ile yapılan uygulamalarda görülmüştür.

Bu tez çalışmasında yaptığımız çalışmalar doğrultusunda, aşağıda maddeler halinde sunulmuş çalışmaların yapılması önerilmektedir:

- Yapılan çalışmalarda sınıflandırıcı verimini arttırmak için, benzerlik gösteren konuşmacı vektörleri için *eşkonuşmacı normalizasyon* algoritmalarının kullanılması önerilmektedir.

- Aynı telaffuzlar, farklı zaman aralıklarında söylenmiş olabileceğinden, bu telaffuzlar için, Dinamik Zaman Eğriltim algoritmalarının ya da Zaman Gecikmeli YSA modellerinin SOM sınıflandırıcılarından önce kullanılması önerilmektedir.

- Sistem veriminin daha çok iyileştirilmesini sağlamak açısından, Şekil 4.14'te gösterilen karma yapıyla, zamana bağlı değişim gösteren işaretler için iyi bir model olan Saklı Markov Modellerinin oluşturacağı melez yapıların, geliştirilmesi önerilmektedir (Cerf 1994, Dugast 1994, Kamaric 1998, Rabiner 1986, Renals 1994, Rigoll 1994, Sun 1994, Wellekens 1993, Zavaliagkos 1994).

KAYNAKLAR

1. ALTINÇAY, H., November 1995. Implementation of a Speaker Recognition System. A master thesis submited to the graduate school of natural and applied sciences of Middle East Technical University.

2. BREWER, D., 1998. A Brief Survey of Speech Recognition Technology. http://lcg-www.uia.ac.be/~erikt/st98/splinks.html (Erişim Tarihi:10.07.2001)

3. CAMPBELL, J., REYNOLDS, F., D., A., 1999. Corpora For The Evaluation of Speaker Recognition systems. ICASSP 15-19 March.

4. CERF, P., WEIYE M., and COMPERNOLLE, D., 1994. Multilayer Perceptron as Labelers for HMM. IEEE Trans. On Speech And Audio Processing, Vol.2, No. 1, Part II, Jan. 1994, pp. 185-193.

5. 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.

6. DELLER, J.R., PROAKIS, J.G., and HANSEN, J.H.L., 1993. Discrete Time Processing of Speech Signals, McMiian Pub. Co.

7. DUGAST, C., DEVILLERS, L. and AUBERT, X. 1994. Combining TDNN and HMM in a Hybrid System for Improved Continuous-Speech Recognition. IEEE Trans. On Speech And Audio Processing, Vol.2, No. 1, Part II, Jan., pp. 217-223.

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. FURUI, S., 1995. Speaker Recognition, Tokyo Institute of Technology, Department of Computer Science, NATO ASI Series from Statistics to Neural Networks Vol:136.

10. HAYKIN, S., 1999. Neural Networks A Comprehensive Foundation. by Prentice-Hall, Inc.

11. 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.

12. 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.

13. İNAL, M., BÜTÜN, E., ERKAN, K., METİN, A., 2001. MLP Neural Net Classifiers using Cepstral Coefficients for Speaker Recognition Application. SCS 2001, Internatinal Symposium on Signals, Circuit & Systems July 10-11 2001, Iasi, ROMANIA.

14. KAMARIC, R., April 1998. Hidden Markov Models and Neural Networks for Speech Recognition, Ph.D. Thesis, Technical University of Denmark

15. KAYRAN, A., H., Ekim 1992. Rasgele İşaretler ve İşaret İşlemedeki Uygulamaları, İ.T.Ü., Elektrik-Elektronik Fakültesi, Dijital İşaret İşleme Lisansüstü Ders Notları.

16. KITAMURA, T., TAKEI, S.1996. Speaker recognition model using two dimensional mel-cepstrum and predictive neural network, ICSLP 96.

17. KRISHNAMOORTHY, M., 1998. Speaker Verification using Neural Tree Network, CAIP Center; Center for Advanced Information Processing" Rutgers University, project report.

18. KRISHNAMOORTHY, M., 1998b. Speaker Identification using LPC Features, CAIP Center " Center for Advanced Information Processing" Rutgers Uniiversity, project report.

19. LEVINE, E., 1995. A Time Warping Neural Networks", Proc. Of the Int. Conf. On Acousitics, Speech and Signal Processing, Dept. of Electrical Engineering, Stanford University, Stanford CA 94305 USA,

20. MAKHOUL, J., April 1975. Linear Prediction: A Tutorial Review, Invited Paper, IEEE proc. Vol.63 , pp.561-580.

21. MORGAN, D., P., SCOFIELD, C., L., 1991. Neural Networks and Speech Processing, by Kluwer Academic Publishers, Boston / Dordrecht /London

22. ÖĞÜNÇ, A., S., Feb. 1990. Implementation of A Speaker Recognition System. A master thesis in Electrical and Electronics Engineering. Middle East Technical University.

23. ÖNCÜL, N., Ocak 1993. Kısa Eğitme Süreli Bir Konuşmacı Tanıma Dizgesi Tasarımı ve Geçekleştirilmesi. Hacettepe Uni., FBE Enst. Eletrik ve Elo. Müh. Yüksek Lisans Tezi.

24. PAOLONI, A. v.d., 1996. Predictive neural networks in text independent speaker verification, ICSLP 96.

25. RABINER, L., and JUANG, B.H., Jan 1986.An Introduction to Hidden Markov Models, IEEE ASSP Magazine.

26. RABINER, L., and JUANG, B.H., 1993. Fundamentals of Speech Recognition, Prentice Hall Signal Processing Series.

27. RENALS, S., MORGAN, N., BOURLARD, H., COHEN, M., and FRANCO, H., Jan.1994. Connectionist Probability Estimators in HMM Speech Recognition, IEEE Trans. On Speech&Audio Proc., Vol.2, No.1, pp.161-174.

28. RIGOLLL, G., Jan. 1994. Maximum Mutual Information Neural Networks for Hybrid Connectionist-HMM Speech Recognition Systems, IEEE Trans. On Speech And Audio Processing, Vol.2, No. 1, Part II, pp. 175-184.

29. ROSENBERG, A., E., DELONG, J., LEE, C., H., JUANG, B., H., and SOONG, F., K., Oct. 1992. The use of cohort normalized scores for speaker recognition, Proc. ICSLP92.

30. SEVEN, A., 1997. Small Vocabulary Word And Speaker Recognition Using Artificial Neural Networks. M.Sc. in Computer Eng., İTÜ.

31. SUN, D., X., 1994. Topics on Hidden Markov Models and their applications in Speech Recognition, Dept. of Applied Mathematics and Statistics, SUNY, Stony Brook, NY 11794-3600, Joint Statistical Meetings in Toronto.

32. TEBELSKIS, J., May 1995. Speech Recognition using Neural Networks, Phd Thesis, School of Computer Science Carnegie Mellon University, Pittsburg, Pennsylvania.

33. WELLEKENS, C., J., 1993. Improved Hidden Markov Models for Speech Recognition Through Neural Network Learning, NATO ASI Series from Statistics to Neural Networks Vol:136 1993

34. ZAVALIAGKOS, G., ZHAO, Y., SCHWARTZ, R., and MAKHOUL, J., Jan. 1994. A Hybrid Segmental Neural Net / Hidden Markov Model System for Continuous Speech Recognition, IEEE Trans. On Speech And Audio Processing, Vol.2, No. 1, Part II, pp.151-160.

EK : PROGRAM LİSTELERİ

Bu ekte; önerilen Türkçe Konuşmacı Tanıma Sistemine ilişkin konuşmacılar için hazırlanan SOM sınıflandırıcılarının eğitiminin yapıldığı ve BBM kullanılarak hangi konuşmacının sisteme sunulacağının seçimini yapan MATLAB slayt gösterim programlarının listeleri verilmektedir.

%SOM Ağlarının Eğitimi ve BBM ağırlıklarının seçimi SOM_OUT.M

```
clear all;
isim={'alper', 'celal','erhan', 'faruk','huseyin', 'melek', 'melih', 'namk','nuran', 'seval'};
N=10;
P_BBM=[];al=[];
for i=1:10,
P=[];
ad=strvcat(isim(i));
eval(['load ' 'c:\matlab_5.2\bin\'ad '\som\train_'eval('ad')]);
trn=eval(['train_' strvcat(isim(i))]);
P=[P trn' ];
disp(ad);
net=newsom(minmax(P),[N N]);
net.trainParam.epochs=10000;
net.trainParam.show=100;
net=train(net,P);
t_trn=sort(vec2ind(sim(net,trn')))';
eval(['a_'ad '=sort(t_trn);'])
eval(['net_'ad '=net;'])
P_BBM=[P_BBM ; t_trn];
al=[al ; size(t_trn,1)];
end

%BBM'nin eğitimi
target=zeros(10,size(P_BBM,1));
target(1,1:al(1))=1;
bas=al(1);
for i=2:10
bas=bas+1;
son=bas+al(i)-1;
target(i,bas:son)=1;
bas=son;
end
egt=zeros(N*N,size(P_BBM,1));
for i=1:size(P_BBM,1),
egt(P_BBM(i),i)=1;
end
w=target*egt';
```

% Hesaplanan veriler daha sonra kullanılmak üzere HEPSİ.MAT dosyasına kayıt edilmiştir

save c:\matlab_5.2\bin\hepsi a_melih net_melih a_alper net_alper a_celal net_celal net_erhan a_erhan a_faruk net_faruk a_huseyin net_huseyin a_nuran net_nuran a_seval net_seval a_melek net_melek a_namk net_namk w

% HEPSİ.MAT Dosyası kullanılarak SOM Ağlarının test edilmesi

```
clear all;
isim={'alper', 'celal','erhan', 'faruk','huseyin', 'melek', 'melih', 'namk','nuran', 'seval'};
N=10;
load hepsi;
for f=1:10,
ww=strvcat(isim(f));
eval(['load ' 'c:\matlab_5.2\bin\' eval('ww') '\som\test_'ww]);
eval(['dene=test_'ww ';']);
k=1;
egitim=[];
while (k<=size(isim,2)),
ad=strvcat(isim(k));
Y=vec2ind(sim(eval(['net_'ad]),dene))';
for j=2:size(Y,1)
if Y(j)~=Y(j-1) d=[d;Y(j)];
end
end
egt=zeros(N*N,size(d,1));
for si=1:size(d,1),
egt(d(si),si)=1;
end
egitim=[egitim egt];
k=k+1;
end
d_dene=w*egitim;
clc;
disp(isim)
```

```
playshow slayt
else
%========== Slide 1 ==========
slide(1).code={
'cla;',
'axis off;',
'text(0.2,0.9,"Türkçe Konuşmacı Tanıma","FontSize",20,"color","b");',
'text(0.4,0.7,"Sistemi","FontSize",20,"color","b");',
'text(0.4,0.2,"Melih İnal","FontSize",20,"color","r");',
" };
slide(1).text={
'Sonraki aşamalarda konuşmacılara ait test vektörleri, her konuşmacının SOM Ağına uygulanarak sınıflandırma verimleri incelenecektir bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...',
"};
%========== Slide 2 ==========
slide(2).code={
'hesapla("alper");' };
slide(2).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...',
"};
 %========== Slide 3 ==========
slide(3).code={
'hesapla("celal")' };
slide(3).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...',
"};
 %========== Slide 4 ==========
slide(4).code={
'hesapla("erhan");' };
slide(4).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
 %========== Slide 5 ==========
slide(5).code={
'hesapla("faruk");' };
slide(5).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
```

```
slide(8).code={
'hesapla("melih");' };
slide(8).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
%========== Slide 9 ==========
slide(9).code={
'hesapla("namk");' };
slide(9).text={
'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
%========== Slide 10 ==========
slide(10).code={
'hesapla("nuran");' };
slide(10).text={
 'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
 %========== Slide 11 ==========
 slide(11).code={
 'hesapla("seval");' };
 slide(11).text={
 'Bir sonraki slayta geçmek için lütfen "Next" butonuna basınız...'};
end
```

% slayt.m fonksiyonunda kullanılan hesapla fonksiyonun çıktısı

```
function hesapla(ww)
load hepsi;
clc;
isim={'alper', 'celal','erhan', 'faruk','huseyin', 'melek','melih', 'namk' ,'nuran','seval'};
nick={'alp', 'cll','erh', 'frk','hsy', 'mlk','mlh', 'nmk' ,'nrn','svl'};
eval(['load ' 'c:\matlab_5.2\bin\' eval('ww') '\som\test_'ww]);
eval(['dn=test_'ww ';']);
S=sprintf('%s %s',ww,'için test sonuçları');
cla;
axis([0 1 0 1.2]);
axis off;
text(0.2,1.1,S,'FontSize',15,'color','b');
bak=[];
for k=1:10,
ad=strvcat(isim(k));
Y=sort(vec2ind(sim(eval(['net_'ad]),dn')));
Y=Y';d=[];
d=Y(1);
for j=2:size(Y,1)
if Y(j)~=Y(j-1) d=[d;Y(j)];
end
end
eval(['a=a_'ad ';'])
ilk=[];
ilk=a(1);
for j=2:size(a,1)
if a(j,1)~=a(j-1,1) ilk=[ilk;a(j,1)];
```

```
end
end
yn=[];
for i=1:size(d,1)
yn=[yn ; size(find(d(i)==ilk),1)];
end
dogru=size(find(yn),1);
ata=dogru*100/size(d,1);
s=sprintf('%s %0.0f ',strvcat(nick(k)),ata);
if strcmp(ww,strvcat(isim(k))) text(0.3,1.1-k/10,s,'FontSize',15,'color','b');
else text(0.3,1.1-k/10,s,'FontSize',10,'color','k');
end
bak=[bak ; ata];
end
```

ÖZGEÇMİŞ

1971 yılında Mardin'de doğdu. İlk, orta öğrenimini Mardin'de tamamladı. Lise öğrenimini Ankara Gazi Teknik Lisesi Bilgisayar İşletim Teknisyenliği Bölümü'nde tamamladı.

1988 yılında girdiği Gaziantep Üniversitesi Fizik Mühendisliği Bölümü'nde bir yıl İngilizce hazırlık eğitiminden sonra 1989 yılında ÖSYM sınavı ile girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Teknolojisi Öğretmenliği Bölümü'nden 1993 yılında mezun oldu. Eylül 1993 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü'nde Araştırma Görevlisi olarak göreve başladı.

Ekim 1994 yılında girdiği, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi A.B.D.'da Yüksek Lisansını 1996 yılında tamamlayarak aynı yıl girdiği, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektrik Eğitimi A.B.D.'da Doktora Eğitimi yapmaktadır.

Eylül 1997 yılından beri aynı bölümde Öğretim Görevlisi olarak görev yapmaktadır.