



T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY SİNİR AĞLARI KULLANARAK
KONUŞMACI TANIMA**

Süheyla BAŞARAN

**YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

BURSA-2007



**T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SİNİR AĞLARI KULLANARAK
KONUŞMACI TANIMA**

Süheyla BAŞARAN

**YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

BURSA-2007

T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI KULLANARAK
KONUŞMACI TANIMA

Süheyla BAŞARAN

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu Tez/...../200... tarihinde aşağıdaki jüri tarafından oybirliği/oy çokluğu ile kabul edilmiştir.

Yrd. Doç. Dr. Figen ERTAŞ
Danışman

.....

.....

ÖZET

Bu çalışmada TIMIT veri tabanından alınan bir test öbeği üzerinde Yapay Sinir Ağları ve geri yayılım algoritmaları kullanılarak metinden bağımsız konuşmacı tanıma uygulamaları geliştirilmiştir.

Konuşmacı tanıma aşamasından önce kişilere ait ses işaretlerinin Mel Frekansı Kepstrum Katsayıları (MFCC) çıkarılarak özellik vektörleri kümesi oluşturulmuştur. Daha sonra özellik vektörlerinin boyutlarındaki büyüklük problemini çözmek amacıyla özellik vektörlerini Vektör Nicemleme işleminden geçirilmiştir.

Konuşmacı tanıma işleminde geriye yayılım algoritması ve yapay sinir ağları ile verilen konuşma örneğinden konuşmacının kim olduğunu tespit etmeye çalışılmıştır.

MFCC katsayıları, Vektör Nicemleme adımları, geriye yayılım algoritması ve yapay sinir ağ yapılarında çeşitli parametrelerin değişmesinin konuşmacı tanıma üzerindeki etkisi incelenmiştir.

Anahtar Kelimeler: Konuşmacı tanıma, yapay sinir ağları, geriye yayılım, vektör nicemleme, MFCC

ABSTRACT

In thesis , It has been aimed to identify the speaker of a given TIMIT database sound with using Neural Network and back propagation algorithms independently from text.

Before recognition phase, the feature set of the training set of sound files were extracted by MFCC transformation. After for solving dimension of the training set of sound's problems Vector Quantization algorithm is used.

Back propagation neural network approach is used identifying of the speakers from the given speech data.

In this thesis parameters of MFCC's, Vector Quantization and Backpropagation Neural Networks were changed and investigated effects of this parameters to the recognition rate of system.

Keywords: Speaker Recognition, neural network, backpropagation, vector quantization, MFCC

İÇİNDEKİLER

TEZ ONAY SAYFASI.....	ii
ÖZET.....	iii
ABSTRACT	iv
İÇİNDEKİLER	v
KISALTMALAR DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
ŞEKİLLER DİZİNİ	ix
GİRİŞ	1
1. KAYNAK ÖZETLERİ	5
1.1. Konuşmacı Tanımda Uygulanan Yöntemler	6
1.1.1. Dinamik Zaman Eğilimli (Dynamic Time Warping).....	6
1.1.2. Vektör Nicemleme	7
1.1.3. En Yakın Komşuluk	7
1.1.4. Saklı Markov Modeli	7
1.1.5. Yapay Sinir Ağları	8
2. MATERYAL VE YÖNTEM.....	12
2.1 Özellik Vektörü Çıkarılması	13
2.1.1 Hızlı Fourier Dönüşüm Yöntemi	13
2.1.2 Doğrusal Tahmin Yöntemi.....	14
2.1.3 Doğrusal Algısal Tahmin (PLP)	14
2.1.4 RASTA.....	14
2.1.5 Mel Frekansı Kepstrum Katsayıları (MFCC)	15
2.1.5.1 Çerçeveleme.....	16
2.1.5.2 Pencereleme	16
2.1.5.3 Hızlı Fourier Dönüşümü	17
2.1.5.4 Mel Frekansına Çevirme.....	17
2.1.5.5 Kepstrum.....	18
2.2 Vektör Nicemleme	19
2.2.1. Ses Özellik Vektörünün Sıkıştırılması	21
2.3 Yapay Sinir Ağları	22

2.3.1 Yapay Sinir Ağlarının Yapısı Elemanları ve Mimarisi.....	24
2.3.2 Yapay Sinir Ağlarının Öğrenme Algoritmalarına Göre	
Sınıflandırılması	32
2.3.2.1 Eğitici Öğrenme	33
2.3.2.2 Eğitici Öğrenme	33
2.3.2.3. Takviyeli Öğrenme.....	33
2.3.3 Çok Katmanlı Alıcılar	33
2.3.3.1 Geri Yayılım Algoritması (Back Propagation)	35
2.3.3.2 Ağ Yapısının Tanımı	37
2.3.4 Toplam Karesel Hata ve Gradyent Azalma	39
2.3.5 Hatanın Kısmi Türevlerinin Ağırlık Vektörlerine Göre Bulunması .	40
2.3.5.1 Çıkış Katmanı ile Saklı Katman Arasındaki Ağırlıklar	40
2.3.5.2 Giriş Katmanı ile Saklı Katman Arasındaki Ağırlıklar	41
2.3.6 Levenberg-Marquardt Algoritması	43
2.4 TIMIT Veri Tabanı	45
3. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	47
3.1 Başarı Oranını Etkileyen Parametreler.....	48
3.1.1 Konuşmacı Tanıma Testleri	50
3.1.1.1 Kod Vektörü Boyutunun Başarıya Etkisi.....	50
3.1.1.2 Yapay Sinir Ağlarında Saklı Katman Nöron Sayısının	
Başarıya Etkisi	53
3.1.1.3 Yapay Sinir Ağlarında Transfer Fonksiyonun Başarıya	
Etkisi	57
3.1.1.4 Konuşmacı Sayısının Başarıya Etkisi	60
3.1.1.5 İterasyon Sayısının Başarıya Etkisi.....	64
3.1.1.6 Farklı Eğitim Algoritmalarının Başarıya Etkisi.....	67
3.1.2 Test Sonuçlarının Yorumlanması ve Öneriler.....	71
KAYNAKLAR	74
ÖZGEÇMİŞ.....	76
TEŞEKKÜR	77

KISALTMALAR DİZİNİ

DFT	-Ayrık Fourier Dönüşümü (Discrete Fourier Transform)
FFT	- Hızlı Fourier Dönüşümü (Fast Fourier Transform)
MFCC	- Mel-Frekans Kepstrum Katsayısı
DTW	- Dinamik Zaman Eğilimli(Dynamic Time Warping)
SMM	- Saklı Markov Modeli
YSA	-Yapay Sinir Ağları
LP	- Doğrusal Tahmin
LPC	-Doğrusal Tahmin Katsayısı (Linear Predictor Coefficients)
PLP	-Doğrusal Algısal Tahmin
RASTA	- Relative Spectra
MLP	- Çok Katmanlı Algılayıcılar(Multi Layer Perception)
LBG	- Linde-Buzo Gray

ÇİZELGELER

Çizelge 3.1 Kod vektör boyutu 16 iken farklı öğrenme oranlarında test sonuçları	50
Çizelge 3.2 Kod vektör boyutu 32 iken farklı öğrenme oranlarında test sonuçları	51
Çizelge 3.3 Kod vektör boyutu 32 iken farklı öğrenme oranlarında test sonuçları	52
Çizelge 3.4 Saklı katman nöron sayısı 15 iken farklı öğrenme oranlarında test sonuçları	54
Çizelge 3.5 Saklı katman nöron sayısı 20 iken farklı öğrenme oranlarında test sonuçları	55
Çizelge 3.6 Saklı katman nöron sayısı 30 iken farklı öğrenme oranlarında test sonuçları	55
Çizelge 3.7 Saklı katman nöron sayısı 30 iken farklı öğrenme oranlarında test sonuçları	56
Çizelge 3.8 Kod vektör boyutu 16 iken sigmoid fonksiyonun başarı test sonuçları....	58
Çizelge 3.9 Kod vektör boyutu 32 iken sigmoid fonksiyonun başarı test sonuçları....	58
Çizelge 3.10 Kod vektör boyutu 64 iken sigmoid fonksiyonun başarı test sonuçları..	59
Çizelge 3.11 On konuşmacı ile yapılan testlerin sonuçları	60
Çizelge 3.12 Onbeş konuşmacı ile yapılan testlerin sonuçları.....	61
Çizelge 3.13 Yirmi konuşmacı ile yapılan testlerin sonuçları	62
Çizelge 3.14 Otuz konuşmacı ile yapılan testlerin sonuçları	63
Çizelge 3.15 Farklı iterasyon sonuçları ile yapılan testlerin sonuçları.....	64
Çizelge 3.16 Gradyant azalan algoritma kullanılarak 20 kişi ile yapılan test sonuçları.....	67
Çizelge 3.17 Gradyant azalan algoritma kullanılarak 30 kişi ile yapılan test sonuçları.....	68
Çizelge 3.18 Momentumlu gradyant azalan algoritma kullanılarak 30 kişi ile yapılan test sonuçları.....	70

ŞEKİLLER DİZİNİ

Şekil 1.1 Konuşmacı tanıma şematik diyagramı.....	5
Şekil 1.2 Beş durumlu Saklı Markov modeli.....	8
Şekil 1.3 Konuşmacı tanımada kullanılan YSA mimarisi	9
Şekil 1.4 Higher Enstitüsünde yapılan konuşmacı tanıma sisteminin hata eğrisi.....	10
Şekil 2.1: Ses işaretine bir örnek.....	12
Şekil 2.2 MFCC algoritması akış diyagramı.....	16
Şekil 2.3 Çevreleme	16
Şekil 2.4 Mel Yayılımlı Filtre Bankası	18
Şekil 2.5 Vektör kuantalamının temel prensipleri.....	20
Şekil 2.6 Vektör kuantalamının temel adımları	20
Şekil 2.7 LBG algoritmasının akış diyagramı.....	22
Şekil 2.8 Sinir Sisteminin Blok Diyagramı.....	24
Şekil 2.9 Biyolojik Nöronun Genel Yapısı ve İşlevleri	25
Şekil 2.10 :Yapay Nöronun Genel Yapısı.....	26
Şekil 2.11 Yapay Nöronun Detaylı Yapısı.....	28
Şekil 2.12 En Çok Kullanılan Transfer Fonksiyonları.....	30
Şekil 2.13: YSA'ların Genel Yapısı.....	31
Şekil 2.14 Danışmalı Öğrenme Yapısı.....	32
Şekil 2.15 Eğiticişiz Öğrenme Yapısı	33
Şekil 2.16 Takviyeli Öğrenme Yapısı.....	34
Şekil 2.17 Çok Katmanlı Ağ Yapısı.....	35
Şekil 2.18 Çok Katmanlı Ağ Yapısı Elemanları	38
Şekil 2.19 Çok Katmanlı Bir Ağın Geri Yayılı Akış Şeması.....	43
Şekil 3.1 Konuşmacı tanımanın 3 temel yapısı.....	47
Şekil 3.2 Kod vektör boyutu 16 ve 0.06 öğrenme oranı ile yapılan testte hata değişim eğrisi.....	51
Şekil 3.3 Kod vektör boyutu 32 ve 0.06 öğrenme oranı ile yapılan testte hata değişim eğrisi.....	51
Şekil 3.4 Kod vektör boyutu 64 ve 0.04 öğrenme oranı ile yapılan testte hata değişim eğrisi.....	52
Şekil 3.5 Kod vektör boyutu 64 ve 10 saklı katman ile yapılan	

testte hata deęişim eęrisi.....	54
Şekil 3.6 Kod vektör boyutu 64 ve 20 saklı katman ile yapılan	
testte hata deęişim eęrisi.....	55
Şekil 3.7 Kod vektör boyutu 64 ve 30 saklı katman ile yapılan	
testte hata deęişim eęrisi.....	56
Şekil 3.8 Kod vektör boyutu 64 ve 40 saklı katman ile yapılan	
testte hata deęişim eęrisi.....	57
Şekil 3.9 Kod vektör boyutu 16 ve sigmoid transfer fonksiyon	
ile yapılan testte hata deęişim eęrisi.....	58
Şekil 3.10 Kod vektör boyutu 32 ve 30 saklı katman nöron sayısı	
ile yapılan testte hata deęişim eęrisi.....	59
Şekil 3.11 Kod vektör boyutu 64 ve 20 saklı katman nöron sayısı	
ile yapılan testte hata deęişim eęrisi.....	60
Şekil 3.12 Kod vektör boyutu 64, 15 kiři ve 10 saklı katman ile	
yapılan testte hata deęişim eęrisi.....	61
Şekil 3.13 Kod vektör boyutu 64, 15 kiři ve 50 saklı katman ile	
yapılan testte hata deęişim eęrisi.....	62
Şekil 3.14 Kod vektör boyutu 64, 20 kiři ve 20 saklı katman ile	
yapılan testte hata deęişim eęrisi.....	62
Şekil 3.15 Kod vektör boyutu 64, 20 kiři ve 20 saklı katman ile	
yapılan testte hata deęişim eęrisi.....	63
Şekil 3.16 İterasyon sayısı 300 iken yapılan testte hata deęişim eęrisi.....	64
Şekil 3.17 İterasyon sayısı 500 iken yapılan testte hata deęişim eęrisi	65
Şekil 3.18 İterasyon sayısı 750 iken yapılan testte hata deęişim eęrisi.....	65
Şekil 3.19 İterasyon sayısı 1000 iken yapılan testte hata deęişim eęrisi.....	66
Şekil 3.20 Gradyant azalan algoritma, 20 kiři ve 30 saklı katman ile	
yapılan teste ait hata deęişim eęrisi.....	68
Şekil 3.21 Gradyant azalan algoritma, 20 kiři ve 50 saklı katman ile	
yapılan teste ait hata deęişim eęrisi.....	68
Şekil 3.22 Gradyant azalan algoritma, 30 kiři ve 30 saklı katman ile	
yapılan teste ait hata deęişim eęrisi.....	68
Şekil 3.23 Gradyant azalan algoritma, 30 kiři ve 50 saklı katman ile	
yapılan teste ait hata deęişim eęrisi.....	69

Şekil 3.24 Momentum katsayısı 0,9, kişi sayısı 30 ve 50 saklı katman ile yapılan teste ait hata değişim eğrisi.....	70
Şekil 3.25 Momentum katsayısı 0,7, kişi sayısı 30 ve 50 saklı katman ile yapılan teste ait hata değişim eğrisi.....	71

GİRİŞ

Ses doğadaki canlılar arasında kullanılan en temel iletişim araçlarından biridir. Sesin insanlar arasında iletişim amacıyla kullanılmasıyla konuşma ortaya çıkmıştır. Dolayısıyla konuşma insanların sahip olduğu en temel iletişim araçlarından biri olmuştur. Başka bir ifadeyle, konuşma insanlar arasında bilgi taşıyan ve bunun için başka bir aracı gerektirmeyen en önemli iletişim aracıdır.

Konuşma iletişim alanlarından en ilginç ve heyecan verici olan, ses yoluyla insan-makine iletişimidir. Bilgisayarlarla haberleşmenin var olması, ne söylendiğinin ve kim tarafından söylendiğinin anlaşılması, iletişimin doğal ve rahat olmasını sağlar. Bu nedenle de konuşmacı tanıma ve konuşma tanıma sistemlerinin tasarımı ve gerçeklenmeleri uzun yıllar üzerinde durulan cazip bir yatırım alanı olmuştur.

Konuşmacı Tanıma, sunulan sözden konuşan kişinin kim olduğunun belirlenmesini kapsar. Günlük hayatta insanlar bu yetiye her gün istemsizce defalarca kez başvurmaktadır. Örneğin, sırtımız dönük olduğu halde seslenen bir tanıdığımızı yüzümüzü dönmeden tanımamız, evimizde kapımızı çalan kişinin kimliğini sesinden anlayıp ona göre açıp açmamaya karar vermemiz ve daha pek çok durum aynı yetinin kullanımını içerir.

Son zamanlarda yapılan deneysel ispatlar makinelerin özellikle konuşmacı doğrulama fonksiyonlarında, kısa ifadelerde ve çok sayıda konuşmacı olması durumunda dinleyicilerden daha iyi bir performans elde edebilme potansiyeline sahip olduklarını göstermektedir. Makinelerle karşılaştırıldığında insanların yeni bir sesi iyi bir şekilde öğrenebilmesinin daha uzun zaman aldığı ortaya çıkmaktadır.

Konuřmacı tanıma üzerine yapılacak alıřmalar iin uygulamanın amacına baėlı olarak farklı yntemler izlenebilir. Bu yntemlerin ayrıntılı aıklamaları ařaėıda yer almaktadır.

Ses iřareti kullanılarak  ana tanıma sistemi üzerine alıřmalar yapılmıřtır. Bu  ana dal; konuřma tanıma, dil tanıma ve konuřmacı tanımadır.

Konuřmacı tanımanın bir alt dalı olan Kimlik Saptama Sistemi, konuřmacı ifadesini giriř olarak alır ve bu test ifadesinin konuřmacılardan hangisine ait olduėunu belirler. Sonuta sistemin ıkıřında, sistem tarafından bilinen konuřmacılardan birinin adı belirlenir. Eėer test ifadesi bilinmeyen bir kiřiye aitse sistem red uyarısı verir.

Konuřmacı Doėrulamada ise giriřteki ses iřaretinin kimliėini beyan eden konuřmacıya ait olup olmadıėı onaylanır. ıkıřta beklenen olası cevaplar evet veya hayır karardır. Bu sistemin sonucunda test ifadesi cmleyi ifade eden konuřmacıya ait olduėu sonucu ıkabilir, ait deėilse sistem ret cevabı verir. Kimlik doėrulama sistemi ‘iddia ettiėin kiři misin?’ sorusunu cevaplar.

Metinden baėımsız konuřmacı tanıma ynteminde sistemin eėitilen veya test edilen ifadeler hakkında bilgisi yoktur. Bunun tersi olarak metine baėımlı sistemlerde eėitim ve test iin kullanılan ifadeler, sistemin belirlediėi parola kelimeler yada cmleler olmak zorundadır.

İnsan sesi, parmak izi ve retina gibi bireylere zg zellikler tařırlar. Bu nedenle de birok bilimsel alıřmaya ara olmuřtur. Bunların temelini ses tanıma, konuřma tanıma, konuřmacı tanıma ve bunları kullanan birok gvenlik sistemi oluřturmaktadır. Bu alıřmalarda temel ara insan sesidir bu nedenle de sesin kullanılabilmesi iin sayısallařtırılması ve kullanılacak amaca uygun zelliklerinin ıkartılması gerekmektedir.

Sesli ifade tanıma sistemlerinin alıřma ilkesi temel olarak, giriř verisinin daha nce kaydedilmiř řablonlarla karılařtırılmasına dayanır. Bu řablonlar fonem dzeyinde ya

da sözcük düzeyinde düzenlenmiş olabilir. Bu karşılaştırma sonucunda verinin en çok benzediği şablon onun simgesel gösterimi olarak kabul edilir. Ses işaretlerinin doğrudan karşılaştırılması çok zordur. Sesin şiddeti farklı olabileceğinden dolayı ses işaretleri üzerinde bir ön işlem yapılması gerekmektedir. Bu ön işlem özellik çıkarımı olarak adlandırılır. Alınan ses verisinden kısa süreli özellik vektörleri çıkarılarak daha önce sınıflandırılmış olan şablonlarla karşılaştırma yapılır. Burada elde edilen özellik vektörlerinin ses verilerini en iyi şekilde temsil etmesi, islenebilir oran ve büyüklükte olması ve ayırt edici özelliklere sahip olması gerekmektedir.

Dayanıklı bir konuşmacı tanıma sistemi sayesinde, artık kullanıcıların anahtar ya da akıllı kart taşımaya 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.

Konuşmacı tanıma üzerine ilk çalışmalar 1960–1970 yıllarında Pruzansky tarafından filtre bankaları ve iki sayısal işaretin spektogramlarının korelasyonu yapılarak benzerlik ölçüsünün araştırılması çalışmalarıyla başlamıştır. Daha sonra 1980 yıllarında Hidden Markov Model ve Vektör Nicemleme yöntemleri kullanılarak konuşmacı tanıma yöntemleri denenmiştir. Daha sonraki yıllarda Zaman Gecikmeli Sinir Ağları, Radyal Temelli Fonksiyonlar, Öğrenicili Vektör Nicemleme ve Çok Katmanlı Algılayıcılar gibi değişik yöntemler eklenmiştir.

Bilgisayar teknolojisindeki gelişmeler sonucu, günümüzde gerçek zamanda konuşma ve konuşmacı tanıma gibi karmaşık uygulamalar gerçekleştirilmektedir. Bu tezde yapay sinir ağlarının bir alt çalışma dalı olan Çok Katmanlı Algılayıcılar kullanarak konuşmacı tanıma yöntemi incelenmiştir.

Konuşmacı tanımda kullanılan ses verileri MFCC'ler kullanılarak özellik vektörlerine ayrılmıştır. Yapay sinir ağlarında giriş verilerinin boyutları büyük olması

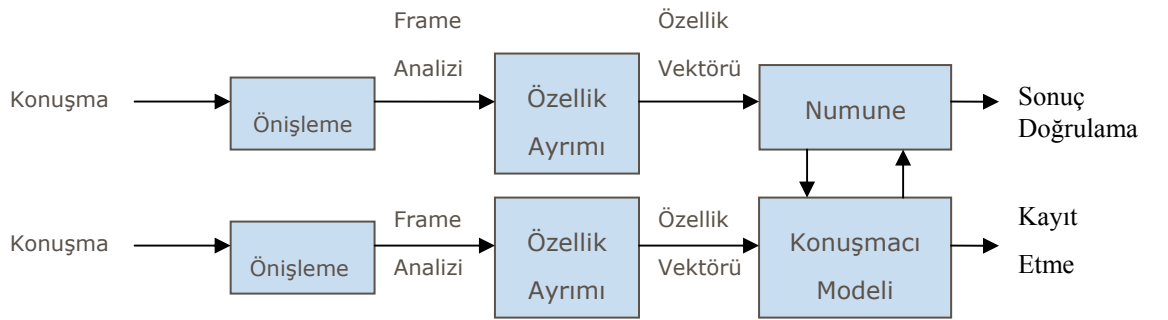
nedeniyle ön bir işlem daha uygulanmıştır. Özellik vektörleri, vektör nicemleme işleminden geçirilerek boyutları küçültülmüştür. Bu işlem sonucu, tanıma işlemi yapılacak kişi sayısı arttırıldığında hata oranında belirli ölçüde artış meydana gelmiştir.

Bu tezde çok katmanlı algılayıcılarda sistem parametrelerini değiştirerek en iyi tanıma oranının sağlandığı koşullar tespit edilmeye çalışılmıştır.

1. KAYNAK ÖZETLERİ

Konuşmacı tanımada kişilerin konuşmalarından ses özelliklerini belirleme ve bu seslere göre kişileri sınıflandırma gibi geniş bir çalışma alanı bulunmaktadır. Ses İşareti birçok mesaj taşımaktadır ve bu işaretlerle uzun yıllar boyunca birçok alt dal geliştirilerek çalışmalar yapılmıştır.

Konuşmacı tanımının genel çalışma yapısını ve kullanılan ara birimleri aşağıda oluşturulan Şekil 1.1 deki gibi gösterilebilir.



Şekil 1.1 Konuşmacı tanıma şematik diyagramı

Bu şemada her bir blok, sistemin bir parçasını temsil eder. Konuşmacı tanıma Özellik Çıkartma ve Modelleme olarak iki ana kısımdan oluşur. Ses işaretine girişte ön işleme yapılarak boyutu azaltılır. Konuşmacı tanımada özellik vektörü çıkarma önemli bir yer oluşturmaktadır. Bu şekilde kişileri temsil eden sayısal vektörler oluşur. Özellik vektörleri daha sonra önceden belirlenen modeli eğitmek için kullanılır. Sistemin en sonunda karar mekanizması vardır. Karar mekanizmasının girişindeki test vektörü ve eğitilmiş model kullanılarak test örneğindeki sesin hangi konuşmacıya ait olduğu tespit edilir.

1.1. Konuşmacı Tanımda Uygulanan Yöntemler

Ses özelliklerinin bazılarını yâda hepsini kullanarak birçok konuşmacı tanıma modeli oluşturulabilir. Hangi modelin kullanılacağı seçimi kullanılacak sesin çeşidine, istenilen performansa, eğitim ve güncelleştirmenin kolay olmasına, depolama ve hesaplama hususlarına dikkat edilerek yapılır. Aşağıda kullanılan en yaygın modeller hakkında özetler bulunmaktadır.

1.1.1 Dinamik Zaman Eğilimli (Dynamic Time Warping)

Konuşma tanımda en önemli problemlerden biri sesin zaman skalasının düzgün olmayışıdır. 1960 yıllarında Martin ve arkadaşlarının RCA laboratuvarlarında yaptığı çalışmalarda tanıma sonuçlarında büyük değişikliklere yol açan sesin başlangıç ve bitiş noktasını belirleyen zaman normalizasyon metodunu geliştirmiştir (Martin 1964). Aynı yıllarda Vintsyuk DTW kullanarak kelime tanıma çalışmaları yapmıştır (Vintsyuk 1968). Fakat 1980'lere kadar yapılan çalışmalar birçok ülkede bilinmemekteydi. 1970'lerden sonra haberleşme üzerine DTW ile beraber Viterbi algoritmaları beraber kullanılarak yapılan çalışmalarda bu yöntemin konuşmacı tanımda kaçınılmaz olduğu anlaşılmıştır.

Belirli bir sözcüğün seslendirilmesi, kişiden kişiye hatta aynı kişinin farklı zamanlarda seslendirmesi ile zaman içinde farklılık gösterebilmektedir. Aynı sözcüğün seslendirilmesi, bir seslendirmede uzun, bir seslendirmede ise daha kısa zamanda gerçekleştirilebilir. Aynı zamanda, ses işaretinde kimi fonemler daha uzun, kimileri ise daha kısa yer almaktadır. Dynamic Time Warping algoritması yardımıyla, bu iki seslendirme, zaman içinde yayılarak ya da daraltılarak birbirine yaklaştırılmaya çalışır. Yani bu iki seslendirmenin, zaman olarak örtüştürülmesi işlevi gerçekleştirilir.

DTW, sözcük tabanlı ses tanıma sistemlerinde etkin ve sıkça kullanılan bir yöntemdir. Bu yaklaşımla, çalışma anında tespit edilen sözcük kesimlemesi, sistemde kayıtlı sözcük şablonları ile seslendirme zamanları örtüştürülerek karşılaştırılması gerçekleştirilebilir.

Doddington filtre bankası kullanarak çıkardığı özellik vektörleri ve dinamik zaman eğilimli yöntem ile yaptığı çalışmada 6 saniye test ifade süresinde %0,8 oranında hata ile konuşmacı tanıma çalışması yapmıştır (Doddington 1985).

1.1.2 Vektör Nicemleme

Bu yöntem sesin birçok kere pencerelenmiş halini kullanan metin bağımlı kalıp modellerinden biridir. Konuşmacıların ses örnekleri kullanılarak her bir konuşmacı için vektör nicelendirilmiş bir kod kitabı oluşturulur. Daha sonra oluşturulan bu kod kitapları konuşmacılar ayırt etmek için karakteristik bir referans oluşturur. Bu modelin çıkışında giriş örneğinin eşleştirme sonucunu bulmak için giriş işareti ile kod kitabındaki kod kelimeleriyle aralarındaki uzaklığın mesafesine bakılır ve en az mesafede olan kod kitabıyla eşleştirilerek sonuca varılır (Linde ve ark. 1980).

1.1.3 En Yakın Komşuluk

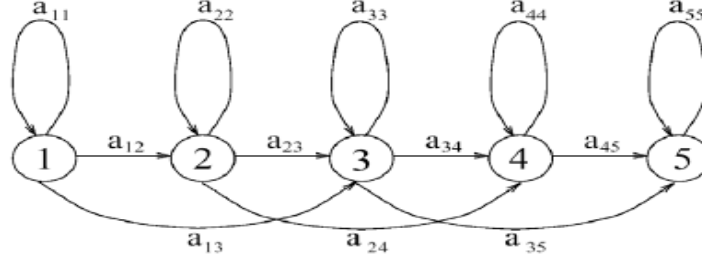
Bu model DTW ve vektör nicemlemenin birleşimi olarak da düşünülebilir. Bu modelde eğitim aşamasında elde edilen dataların hepsi korunur, kod kitabı oluşturmak için herhangi bir sınıflandırma yapılmaz. En yakın komşuluk mesafesi giriş ile saklanan kalıp arasındaki minimum mesafedir. Çıkıştaki eşleştirmeyi yapmak için tüm girişlerin en yakın komşuluk mesafelerinin ortalaması alınarak hesaplanır. Daha sonra çıkıştaki eşleştirme skorları birleştirilerek olasılık oranının yakınsamasında kullanılır.

Higgins ve Wohlford 'un kepstrum katsayılarını kullanarak en yakın komşuluk yöntemiyle 11 kişi ile yaptığı çalışmada 2,5 saniye test ifadesi ile %10 hata oranı ile metinden bağımsız konuşmacı tanıma çalışması yapmıştır (Higgins ve Wohlford 1986).

1.1.4 Saklı Markov Modeli

Bu metottaki temel fikir ses işaretinin parametrik bir rasgele işlem olarak ifade edilebilmesidir. Saklı Markov modeli iki stokastik süreç içerir. İlk olan Markov Modeli zaman ile ilgili değişiklikleri kullanır ve durumları içeren Markov zinciri üretir. Diğer

süreç gözlemlenebilir ve özellik parametreleri veya gözlemler denilen rasgele değişkenler içerir (Koç 1999).



Şekil 1.2 Beş durumlu Saklı Markov modeli

Ses tanıma probleminde bir SMM'in altında yatan düşünce bir ses işaretinin en iyi parametrik bir rasgele süreç olarak karakterize edilebilmesidir. Böylece stokastik süreç parametreleri kesin ve iyi tanımlanmış bir şekilde hesaplanabilir.

SMM teorisi herhangi bir sonlu durum otomasyonuna uygulanırken lineer bir durumlar dizisi kullanılır. Durumlar normalize edilmiş bir zaman ekseninde zaman noktaları olarak açıklanabilir. Konuşma hızı değişimlerini hesaplamak için her durumun solda olduğu, genel olarak üç tip olası geçiş vardır: sonraki duruma git, aynı duruma geri dön ve bir durum ileri atla. Benzer yapı Şekil 1.2'de görülebilir. Uygulamada sadece gözlem dizisi bilinir ve temelde olan durum dizisi bilinmez. Bu yapıya Saklı Markov Model denmesinin nedeni budur.

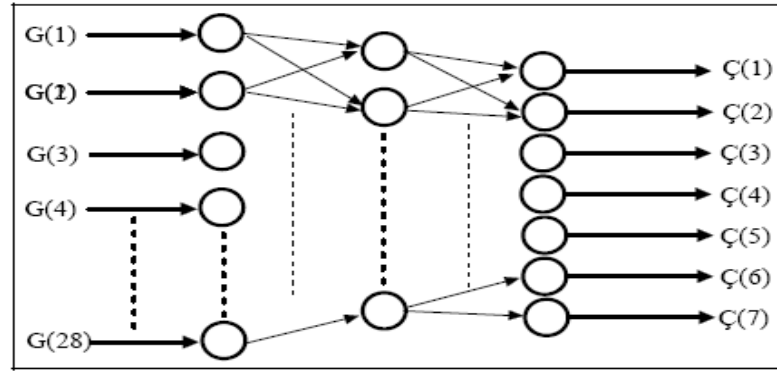
Che ve Lin YOHO veri tabanının kullanarak SMM yöntemi ile yaptığı metinden bağımsız konuşmacı tanıma çalışmasında 2,5 saniye ifade süresiyle %0,56 hata oranı elde etmiştir (Che ve Lin 1995).

1.1.5 Yapay Sinir Ağları

YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli

şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçekleştirilebilir.

Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını ihtiva eder.



Şekil 1.3 Konuşmacı tanımada kullanılan YSA mimarisi

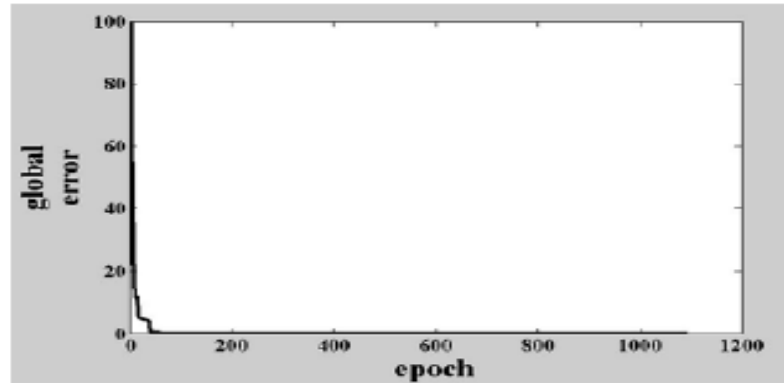
Yapay sinir ağları hesaplama ve bilgiyi işleme konusunda birçok çalışmada başarılı sonuçlar vermiştir. Bu başarının nedeni paralel dağılmış yapısı, öğrenebilme ve genelleme yeteneğidir. Bu özelliği nedeniyle yapay sinir ağları karmaşık problemleri çözebilme yeteneğini gösterir.

Yapay Sinir Ağları (YSA) herhangi bir eşleme için eğitilebildiğinden ses tanımada ses işaretinin sınıflandırılması (modellenmesi) aşamasında kullanılabilir.

Abdul Ahad ve arkadaşlarının yaptığı çalışmada yapay sinir ağlarının en sık kullanılan yapısı olan çok katmanlı algılayıcıların konuşma tanıma üzerindeki başarısı incelenmiştir. Bu çalışmada 0'dan 9'a kadar olan sayılardan oluşmuş veri bankası kullanılmıştır. Her bir ifade 100 kere tekrarlanarak toplam 1000 ifadeden oluşan bir veri bankası oluşturulmuştur. Bu veri bankası kullanılarak konuşma tanıma için en ideal

yapay sinir ağı modeli bulunmaya çalışılmıştır. Bu çalışmada 2 katmanlı ağı kullanıldığında öğrenme oranının etkisi ve iterasyon sayısının en etkin tanıma etkisi incelenmiştir. Sistemin en iyi performansı %0,01 hata oranıyla elde edilmiştir ve öğrenme oranının değişik değerleri sistemin başarısını doğrudan etkilemediği görülmüştür. Ayrıca birkaç yüz iterasyondan sonra sonuca varıldığı belirtilmektedir (Ahad ve ark. 2002).

Higher Enstitüsünde Sajid A.Marhon ve arkadaşları tarafından yapılan yapay sinir ağlarıyla konuşmacı tanıma çalışmasında ortalama %80 tanıma başarısı sağlanmıştır. Bu çalışmada toplam 8 kişiden 5 adet eğitim için 5 adet test için toplam 80 adet veri alınarak veri bankası oluşturulmuş ve konuşmacı tanıma bu örneklerle yapılmıştır. Bu çalışmada eğitilen sistemin toplam hata değerinin değişim eğrisi aşağıdaki şekilde gösterilmiştir (Marhon ve ark. 2006).



Şekil 1.4 Higher Enstitüsünde yapılan konuşmacı tanıma sisteminin hata eğrisi

Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği bölümünde çok katmanlı algılayıcılar kullanılarak 4 kişiden alınan 72 adet ses ile konuşmacı tanıma çalışması yapılmıştır. Bu çalışmada ses örneklerinin mel frekans kepsrum katsayıları çıkarılarak geri yayılım algoritması uygulandığında %57 oranında başarı sağlanmıştır (Moralı ve Aygün 2006).

Afyon Kocatepe Üniversitesi ve Celal Bayar Üniversitelerinde yapılan çalışmada 7 farklı kişiden alınan 'a' sesli harfinin öznelikleri bulunmuş ve bunlar sesli harfin alındığı kişiyi bulacak şekilde eğitilmiş çok katmanlı ağı yapısı eğitilmiştir. Daha sonra

eđitim setinde olmayan verilerle ađın bařarı sı test edilmiř ve 7 kiřide ortalama %90 tanıma bařarı sı elde edilmiřtir (Caner ve Vakkas 2005).

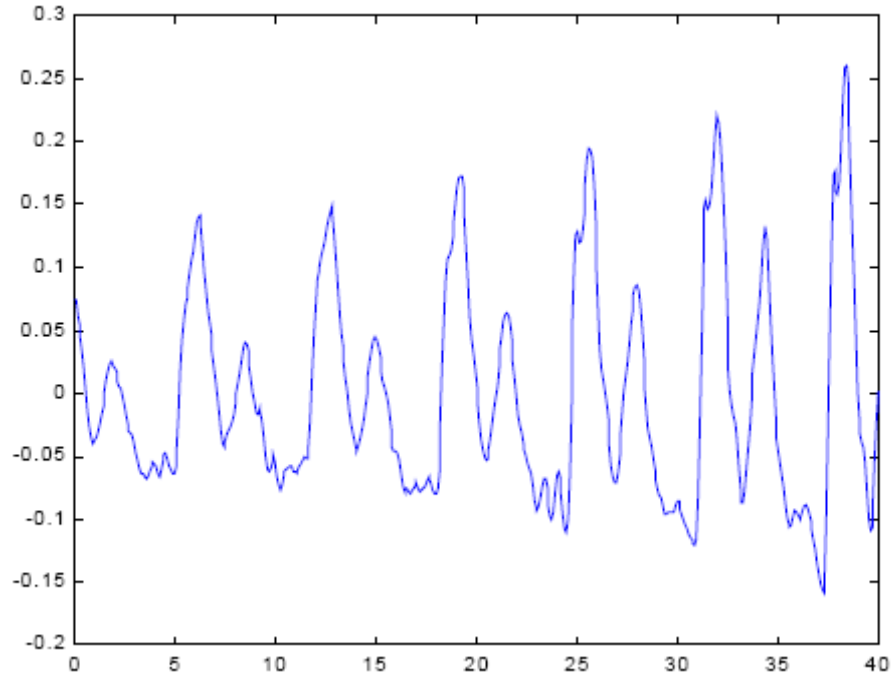
İpek Bariř ve arkadařları tarafından 7 kiřiden alınan ses rneklerinden 260 tane keprstral katsayı hesaplanmıř ve bunlardan bir veri tabanı oluřturulmuřtur. Daha sonra MATLAB programı tarafından yazılan YSA, oluřturulan veri tabanına gre eđitilip test edilmiřtir. YSA' nın ıkıřında konuřmacıların kimlikleri %90'lık dođruluk payıyla bulunmuřtur (Bariř ve ark. 2002).

YSA metotları farklı tipteki yapıları đrenebilir. Bađlantılardaki ađırlık deđerlerini deđerřtirerek ncelik verdiđi yapıyı deđerřtirebilir ve bu řekilde bařarı oranı artırır. Yapay sinir ađlarının bu yapısı zellikle ses tanıma ve yorumlanmasında etkilidir (Bennani ve Gallnari 1995). Yapay sinir ađları ile yapılan konuřmacı tanıma alıřmalarına bakıldıđında genel olarak az sayıda kiři ile ve kısa ifadelerle tanıma alıřması yapıldıđı grlmektedir. Bu tezin amacı konuřmacı tanıma yapılırken kiři sayısını artırmak ve cmle bazında konuřmacı tanıma alıřması yapmaktır. TIMIT veri tabanında bulunan kiřilere ait cmlelerden oluřan veri tabanı incelendiđinde karřımıza ıkan ilk sorun oluřacak veri tabanının boyutunun bykldr. Bu tezin ilk adımı veri tabanının byklđnn azaltılması ve bu sayede daha ok konuřmacı ile test yapmaktır. Bu amala tezde, yapay sinir ađları ile konuřmacı tanıma zerine yapılan testler ve sonuları hakkında bilgi verilecektir.

2. MATERYAL VE YÖNTEM

Konuşmacı tanıma çalışmalarında insan sesi temel araçtır bu nedenle de sesin kullanılabilmesi için sayısallaştırılması ve amaca uygun bir özelliklerini çıkartılması gerekmektedir. Özellik vektörlerinin ses verilerini en iyi şekilde temsil etmesi, islenebilir oran ve büyüklükte olması ve ayırt edici özelliklere sahip olması gerekmektedir.

Ses işareti zamanla yavaş değişen bir işarettir bu nedenle de durağanımsı olarak da adlandırılır. Şekil 2.1’ de bir ses işareti örneği görülmektedir. Genelde çalışmalar kısa zaman periyotlarında (5 ile 100 milisaniye) yapılmıştır çünkü ses karakteristiği çok daha durağandır. Bununla birlikte uzun zaman periyotlarında işaret karakteristiği değişerek konuşulan farklı konuşma seslerini de yansıtır. Bu nedenle de ses işaretinin özelliklerini çıkarmada kısa zaman spektral analiz kullanılmaktadır.



Şekil 2.1 Ses işaretine bir örnek

2.1 Özellik Vektörü Çıkarılması

Konuşmacı tanıma sistemlerinde kullanılan özellik çıkarma yöntemlerinden bazılarını örnek olarak Hızlı Fourier Dönüşümü, Cepstral Katsayılarının kullanılması, Doğrusal Tahmin Yöntemi, Doğrusal Algısal Tahmin yöntemi, RASTA gibi yöntemler verilebilir. Bu özellik çıkarma yöntemleri hakkında genel bilgiler aşağıda değinilmiştir.

2.1.1 Hızlı Fourier Dönüşüm Yöntemi

Konuşma işaretiyle çalışmanın en genel tekniklerinden biri güç spektrumudur. Güç spektrumu konuşma işaretinin zaman içindeki frekans bilgilerini gösterir. Konuşma işaretinin güç spektrumunun hesaplanmasındaki ilk aşama Ayrık Fourier Dönüşümünün (Discrete Fourier Transform) yapılmasıdır. Ayrık Fourier Transform eşit zaman aralığındaki frekans bilgisini gösterir. DFT direk olarak kullanılırsa zaman alıcı bir algoritmadır fakat eğer ses düzgün parçalanırsa çok hızlı bir şekilde hesaplanabilir ve bu da Hızlı Fourier Dönüşümü (Fast Fourier Transform) algoritmasının temelidir.

Hızlı Fourier Dönüşümü ses işaretini meydana geldiği frekanslara ayırır. Normal bir ses işareti de renklerin farklı farklı renklerin birleşmesiyle oluşması gibi çeşitli frekanslardaki seslerin birleşmesi ile oluşur. Ses işaretinin ham şekli insan aynı sözcüğü söylese de farklı görüntüler almaktadır. Dolayısıyla ses işaretini ham hali (zaman ve genlik) ile algılamak zordur. Bu nedenle çeşitli şekillerde ses işareti işlenir ve bu şekilde tanınmaya çalışılır. FFT bu tekniklerden birisidir. Aynı sözcüğü ifade eden iki ses işaretine ham hali ile bakıldığında işaretlerin benzerliği zor görünür. Ama işaretlerin FFT' sine bakıldığında benzerliği daha kolay görünür. Ancak FFT'si alınan işaretler de bazı ufak farklılıklar içerirler. Bu dönüştürme işlemi, ses işaretinin belirleyici ve tanımda faydalı olan özelliklerini çıkarma amacı ile ses verisi üzerinde uygulanmaktadır.

2.1.2 Doğrusal Tahmin Yöntemi

Sesli ifade çözümleme tekniklerinden en güçlüsü doğrusal tahmin (*Linear Prediction*) yöntemidir. Bu yöntem sesli ifade parametrelerinin tahmininde kullanılan en etkin yöntemlerden biridir. Bu parametreler, temel frekans, formant frekansları, spektrum, gırtlak modelleyen fonksiyonlar ve sesli ifadenin sıkıştırılmasında kullanılan kodlama teknikleridir. Bu metodun önemi ürettiği doğru sonuçlara ve bu sonuçlara ulaşırken görece olarak daha hızlı çalışmasıdır. LP çözümlemenin dayandığı temel fikir, bir sesli ifade örneğinin geçmiş sesli ifade örneklerine dayanarak yaklaşık olarak elde edilebileceğidir. Su andaki örnek sesli ifade ile doğrusal olarak tahmin edilen sesli ifade arasındaki farkların karelerinin toplamı en aza indirilmeye çalışılarak, sesli ifadenin tahminini sağlayacak bir dizi birim katsayı bulunabilir. Bu katsayılara tahmin edici katsayılar (*Linear Predictor Coefficients*) denir ve tahmin edilene sesli ifadenin doğrusal olarak birleştirilmesi sırasında kullanılan ağırlıklandırma katsayıları olarak da tanımlanabilirler.

2.1.3 Doğrusal Algısal Tahmin (PLP)

LP tekniğinde sesli ifade modellenirken tüm frekanslardaki sesler eşdeğer tutulmaktadır. Bu durum insan kulağıyla uyumlu değildir. 800 Hz değerinden daha düşük frekanslarda duyma miktarı frekansla birlikte düşer. İnsan kulağı daha çok duyma frekans aralığının ortasındaki frekanslara duyarlıdır. Bu sorunu çözmek için birçok çalışma yapılmıştır. Bu çalışmalardan biri de bulunan LP katsayılarının *mel* skalasına uyarlanması olmuştur. Bir başka yaklaşım da LP tekniği uygulamadan önce sesli ifadenin güç spektrumunun alınmasıdır (Hermansky 1990). PLP yöntemi de bu yaklaşımı kullanmaktadır.

2.1.4 RASTA

Özellik vektörü oluşturmada kullanılan RASTA yönteminde, sesli ifade içindeki çevresel etkilerin, yani gürültünün, modellenmesine dayalı bir sesli ifade modelleme yöntemi kullanılır. RASTA ismi, görece spektrum (RelAtive SpecTrA) ifadesinin

kısaltılmasıyla elde edilmiştir. Yukarıda belirtilen PLP yöntemi üzerine gürültü modelleme tekniği eklenerek elde edilen bir yöntemdir.

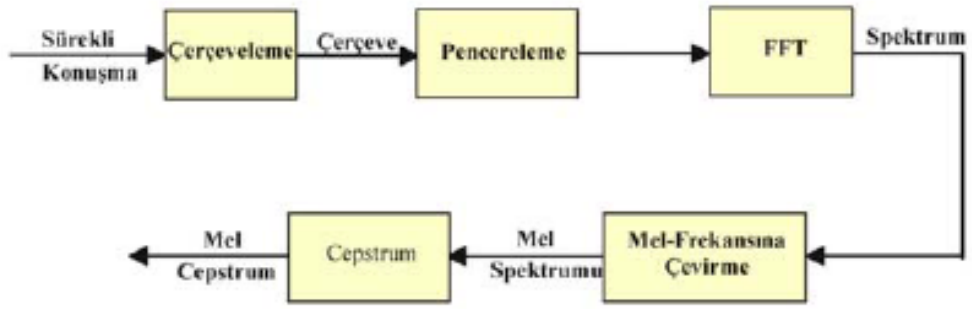
RASTA yönteminin dayandığı temel insan kulağının sesli ifadeyi algılamasının daha önceki seslerden önemli derecede etkilendiğidir. Yani sesli ifadenin algılanması daha önce duyulan seslere bağlıdır. Daha değişik bir ifadeyle algılama su andaki ses ile önceki ses arasındaki spektral farka bağlıdır. Bu durumda insan kulağı yavaş değişen seslere daha az duyarlıdır denebilir.

Yapılan sesli ifade çözümlemesinin yavaş değişen seslere daha az duyarlı yapılması insan kulağının bu özelliğinin de modellenmesini sağlar. Bunu yapmak için daha önce belirtilen PLP yönteminde kullanılan filtreleme yönteminde değişiklikler yapılmıştır. Kullanılan filtreler spektral sıfır değeri keskinleştirilmiş, yani sıfır frekans düzeyine aniden inen filtrelerle değiştirilmiştir. Böylece frekanslardaki yavaş değişimlerin etkisi azaltılmıştır (Hermansky and Morgan 1994).

2.1.5 Mel Frekansı Kepstrum Katsayıları (MFCC)

MFCC algoritmasını ilk kez Davis ve Mermelstein 1980 yılında duyurmuştur. Duyurulduktan sonra en çok tercih edilen algoritmalarından biri olmuştur (Davis ve Mermelstein 1980). MFCC kullanılarak yapılan çalışmalarda tanıma oranı oldukça yüksek olduğu tespit edilmiştir.

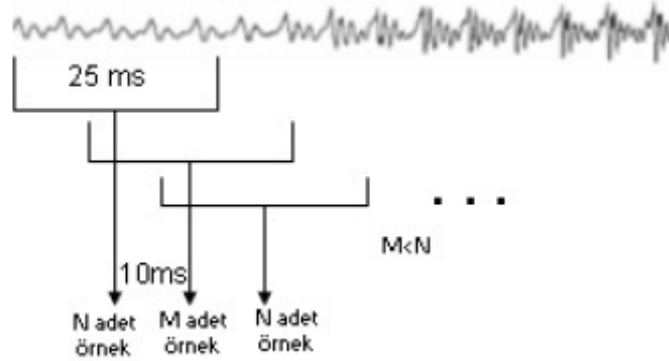
MFCC işlemi veya algoritmasının blok diyagramı Şekil 2.2' de verilmiştir. Konuşma girişi 10000 Hz'den yüksek bir frekansta örneklenmelidir. Bu frekans eşiğinin seçilmesinin sebebi aliasing yani düşük frekanslarda ve çözünürlükte yapılan örneklemede yeterli nokta örneklenemeyeceği için eğimin değiştiği yerlerde sapmalar olacak ve gerçek ses işaretinden farklı bir işaret örneklenmiş olacaktır. Bu örneklenen işaretler 5 kHz' e kadar bütün frekansları yakalayacaktır. 5 kHz, insan sesinin aşağı yukarı bütün enerji seviyelerini içerir. Daha önce belirtildiği gibi MFCC insan kulaklarının davranışlarını taklit eder. Ayrıca MFCC değişimlerden, ses dalga yapısından çok daha az etkilenir.



Şekil 2.2 MFCC algoritması akış diyagramı

2.1.5.1 Çerçeveleme

Bu aşamada konuşma sürekli işareti N adet örnekten oluşan çerçevelere ve komşu M örnekten oluşan çerçevelere bölünür. ($M < N$) İlk çerçeve N örnekten oluşurken ikinci çerçeve ilk çerçeve den M örnek sonra başlar ve ilk çerçevenin $N-M$ çerçeve kadar üzerine biner. (Örneğin $N=256, M=100$)



Şekil 2.3 Çerçeveleme

2.1.5.2 Pencereleme

Sıradaki işlem her çerçeveyi pencereleyerek, işaretin başı ve sonundaki süresiz bölümleri minimize etmektir. Buradaki amaç, pencereyi kullanarak çerçevelerin başı ve sonundaki bilgi içermeyen bölümleri kırmak dolayısıyla spektral bozulmayı engellemektir. Eğer pencereyi $w(n)$, $0 \leq n \leq N-1$, N her çerçevedeki örnek sayısı olarak kabul edersek, işaretin pencerelenmiş hali

$$y(n) = x(n)w(n), \quad 0 \leq n \leq N-1 \quad (2.1)$$

2.1.5.3 Hızlı Fourier Dönüşümü

Sıradaki işlem, N örnekli her çerçevenin zaman domeninden, frekans domenine çevrilmesi için Hızlı Fourier dönüşümü uygulanmasıdır. FFT, Ayırık Fourier dönüşümü (DFT) uygulamak için hızlı bir algoritmadır. N örneklilik bir set $\{x_n\}$ için şöyle tanımlanabilir:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn / N}, \quad n=0,1,2,\dots,N-1 \quad (2.2)$$

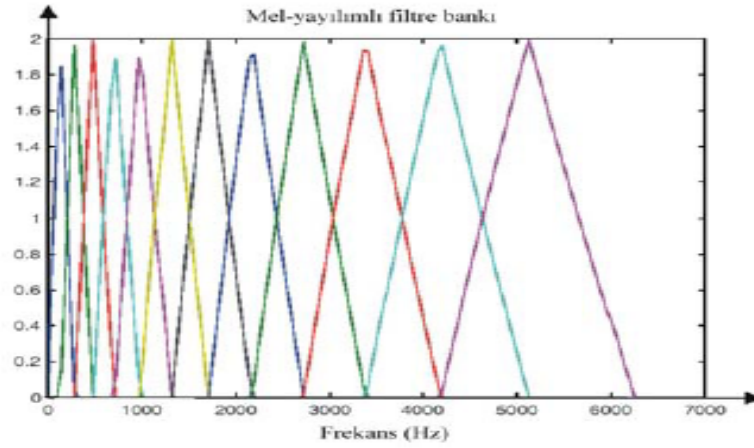
Bu aşamadan sonraki sonuç spektrogram ve periyodogram diye anılır.

2.1.5.4 Mel Frekansına Çevirme

Daha öncede belirtildiği gibi, psikofiziksel çalışmalar, insanın frekans içeriklerini algılamasının lineer bir skalada gerçekleşmediğini göstermektedir. Böylece f frekanslı her ses için “mel” skalası denen sübjektif bir skala hesaplanır. Mel-Frekansı 1000 Hz’in altında lineer, 1000 Hz’in üstünde logaritmik dağılıma sahip bir skaladır. Bir referans noktası olarak alınan 1 kHz’lik ses, insan duyma eşiğinin 40 dB üstünde olmakla birlikte 1000 mel olarak tanımlanır. Bundan dolayı Hz türünden bilinen bir frekansın mel eşdeğerini bulmak için şu formül kullanılabilir:

$$\text{mel}(f) = 2595 \cdot \log(1 + f / 700) \quad (2.3)$$

Bu sübjektif spektrumu simüle etmek için bir yol mel skalasında düzgün olarak yerleştirilmiş bir filtre bankası kullanmaktır (Şekil 2.4). Bu filtre bankı üçgen bant geçiren ve aralıkları ve bant genişliği sabit mel frekansı aralığına bağlı bir banktır. Yayılışı ve band genişliği sabit bir mel frekansı aralığına sahiptir.



Şekil 2.4 Mel Yayılımlı Filtre Bankası

Bu filtre bankının frekans domeyninde uygulandığına dikkat ederek, işaretin FFT ile frekans domeynine çevrilmiş haline uygulanır. Bu mel dönüştürücü filtre bankındaki her filtre frekans domeyninde bir histogram kutusu (kutuların birbiri üzerine binmesi) oluşturur.

2.1.5.5 Kepstrum

Bu son aşamada, logaritması alınmış mel spektrumunun frekans domeninden tekrar zaman domenine çevrilmesi gerekmektedir. Sonuç, Mel Frekansı Kepstrum Katsayıları (Mel Frequency Cepstrum Coefficients) (MFCC) adını alır. Verilen çerçevenin analizinde kepsral gösterim, işaretin bölgesel spektral özelliklerini çok güzel temsil ve tasvir etmektedir. Mel Spektrum Katsayıları (ve onun logaritması) gerçek sayılar oldukları için, Ayrık Kosinüs dönüşümü (DCT) ile onları zaman domeynine çevirmek mümkündür. Son aşamanın sonucu olan mel güç spektrumu katsayıları S_k , $k = 1, 2, \dots, K$ olarak gösterebilir. S_k 'lar kullanılarak C_n (MFCC'ler) aşağıdaki bağlantılarla hesaplanır.

$$C_n = \sum_{k=1}^K (\log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad n=1, 2, \dots, K \quad (2.4)$$

C_0 dikkate alınmamıştır. Çünkü giriş işaretinin ortalama değerini içermektedir ki konuşan kişinin kimliği hakkında çok az bilgi içerir. Mel Frekans skalasında tanımlanmış kısa vadeli güç spektrumunun logaritmasının kosinüs dönüşümü bu sonucu verir. Bu katsayı setine *akustik vektör* denir. Yani giriş ses işareti akustik vektörlere dönüştürülmüş olmuştur.

2.2 Vektör Nicemleme

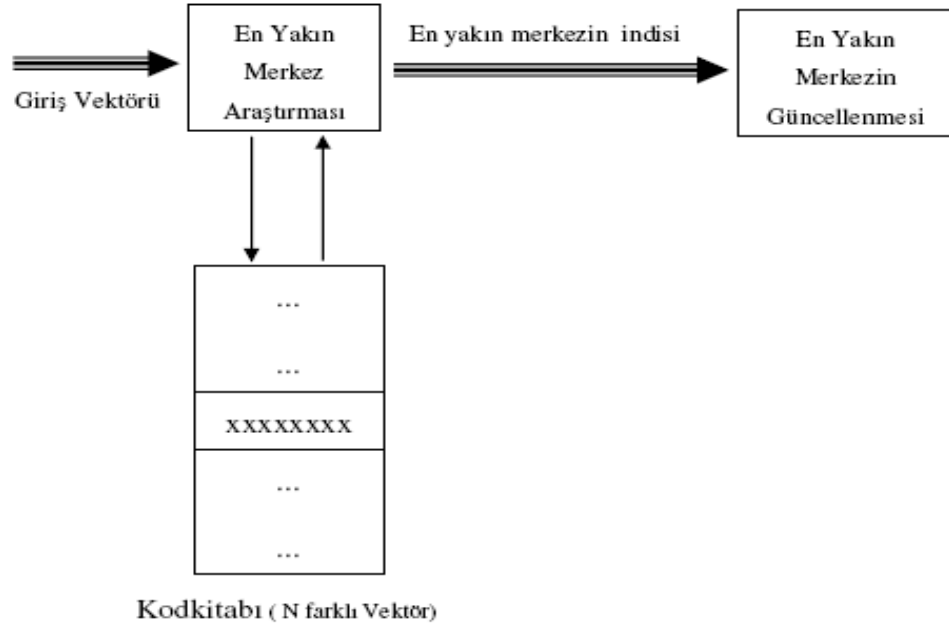
Vektör nicemleme, depolanmak ya da iletilmek istenen verinin, kendisini en iyi şekilde yansıtabilen daha az miktarda veriyle ifade edilmesidir. En iyi şekilde yansıtan ifadesinde anlatılmak istenen, verinin miktarını azaltırken oluşacak kaybın (bozulmanın) en az olduğu veri kümesini bulabilmektir (Duda ve ark. 2001).

Benzerliği uzaklık ile yorumlayacak olursak, vektör nicemleme algoritmaları, aynı merkezin örnekleri birbirine yakın, farklı merkezlerin örnekleri birbirinden uzak olacak şekilde merkezlerin yerleri belirlemeyi amaçlar (Duda ve ark. 2001).

Vektör nicemleme algoritmalarının temel yapısı, giriş uzayını kendisini en iyi şekilde ifade edecek belli sayıdaki çıkış kümesi ile belirlemeye dayanmaktadır. Bu, girişi gruplandırmak olarak da tanımlanabilir. Öncelikle giriş uzayının kaç adet çıkış ile ifade edileceği belirlenir. Çıkış sayısı ne kadar büyük seçilirse çıkıştaki bozulma o oranda daha düşük olmaktadır.

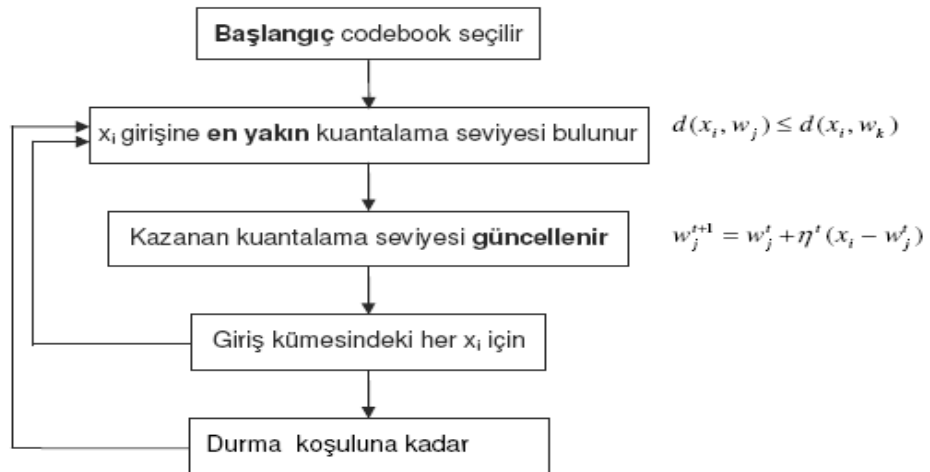
Literatürde her bir çıkış -nicemleme seviyesi-, kodvektör; bu çıkışlar kümesine ise kod tablosu adı verilir. Giriş uzayından seçilen bir giriş vektörü için nicemlemenin temel yapısı Şekil 2.5' de görüldüğü gibidir.

İlk adımda çıkış olarak ifade edilecek nicemleme seviyelerine ilk değer ataması yapılmalıdır. Daha sonrasında giriş vektörüne en yakın nicemleme seviyesi belirlenir. Kullanılan vektör nicemleme algoritmasına göre en yakın nicemleme seviyesi veya en yakın seviye ile birlikte komşuları güncellenir.



Şekil 2.5 Vektör nicemlemenin temel prensipleri

Vektör niceme algoritmalarının adımları Şekil 2.6’da gösterildiği gibidir. Algoritmanın adımları olan niceme seviyelerine ilk değer atama, en yakın seviyenin hesaplanması ve niceme seviyelerinin güncellenmesi için farklı yöntemler geliştirilmiştir. Bu adımlar için kullanılacak metotlar niceme başarısını doğrudan etkilemektedir.



Şekil 2.6 Vektör nicemlemenin temel adımları

2.2.1. Ses Özellik Vektörünün Sıkıştırılması

Bu tezde konuşmacı tanımada kullanılacak özellik vektörlerinin boyutunun sıkıştırılması amaçlanmıştır. Bu amaçla özellik vektörlerinden yararlanarak, söz konusu konuşmacı için VQ kod kitabının hesaplanmasına sıra gelmiştir. Oldukça iyi bilinen bir algoritma olan LBG algoritması (Linde ve ark. 1980), L adet eğitim vektör setini M adet kod kitabı vektörü halinde demetlemek için kullanılır. Algoritma aşağıdaki tekrarlanan prosedür ile gerçekleştirilir:

1. Vektörlük kod kitabının dizaynı: Bu bütün eğitim vektörleri setinin centroid'i (kod kelimesi) olacaktır.

2. y_n diye tanımlanan her kod kitabını bölerek kod kitabının büyüklüğü aşağıdaki kural uygulanarak iki katına çıkarılır:

$$y_n^+ = y_n(1 + \varepsilon) \quad (2.5)$$

$$y_n^- = y_n(1 - \varepsilon) \quad (2.6)$$

n 'in değeri 1 ile kod kitabının o an ki boyutu arasında değişir ve ε 'da bölme parametresidir.

3. En yakın komşu araştırması: Her eğitim vektörü için, o anki kod kitabındaki en yakın kod kelimesi bulunur(benzerlik ölçümü şeklinde), ve ilgili hücreye bu vektör atanır(en yakın kod kelime'sine ortak olarak).

4. Centroid (kod kelime) güncellemesi: her hücredeki kod kelimeyi, o hücreye atanmış eğitim vektörlerinin kod kelimesini kullanıp kod kelimesini güncelle.

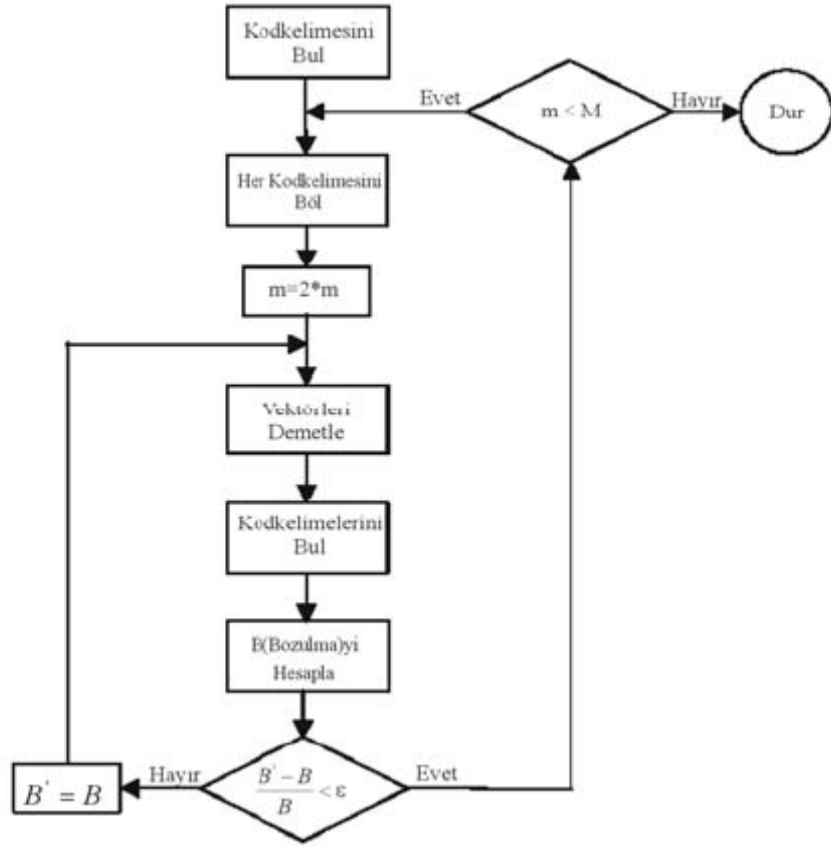
İterasyon 1: üçüncü ve dördüncü adımları tekrarlayarak ortalama uzaklık belirtilen eşik değerden düşük olana kadar devam edilmelidir.

İterasyon 2: iki, üç ve dördüncü adımları tekrarlayarak kod kitabının boyutu M olana kadar devam edilir.

LBG algoritması bu aşamalardan geçerek M boyutunda kod kitabı oluşturacaktır. İlk başta 1 vektörlük kod kitabı ile başlanır, ardından bölme tekniği kullanılarak 2 vektörlük bir kod kitabı oluşturulmaya çalışılır. Daha sonra tekrar bölme tekniği kullanılarak M vektörlü kod kitabı bulunana kadar aynı işlemlere devam edilir.

Şekil 2.7 LBG algoritmasının detaylı aşamalarını akış diyagramında göstermektedir. “Vektörleri Demetle” her eğitim vektörünü en yakın kod kitabı ile ilişkili bir demete bağlayan en yakın komşuyu arama prosedürüdür.

“Kod kelimelerini Bul” ise centroidleri güncelleme prosedürüdür. “Bozulmayı Hesapla”, prosedürün hangi değere yakınsadığını görmek için en yakın komşuyu arama sırasında eğitim vektörlerinin uzaklıklarını toplar.



Şekil 2.7 LBG algoritmasının akış diyagramı

2.3 Yapay Sinir Ağları

Teknolojik gelişmenin önemli boyutlara ulaştığı günümüzde, insanoğlunun kendisini tanımaya yönelik çalışmaları da önemli aşamalar kaydetmiştir. Yapay zekâ (artificial intelligence) kavramı ile insanın en önemli özellikleri olan düşünebilme ve öğrenebilme yetenekleri en önemli araştırma konuları durumuna gelmiştir. Özellikle son

zamanlarda bilgisayar kullanımının hızla yaygınlaşması sonucunda yapay zekâ çalışmaları da bir ivme kazanmıştır.

İnsanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanan yapay zeka, aslında programlanmış bilgisayarlara düşünme yeteneği sağlama girişimidir. İnsan gibi düşünen ve davranan sistemlerin geliştirilmesine yönelik olarak 1950'li yıllardan beri süren yapay zeka çalışmaları, bir noktada insanı taklit etmeye yönelik olduğundan mühendislik, nöroloji ve psikoloji gibi alanlara da yayılmıştır.

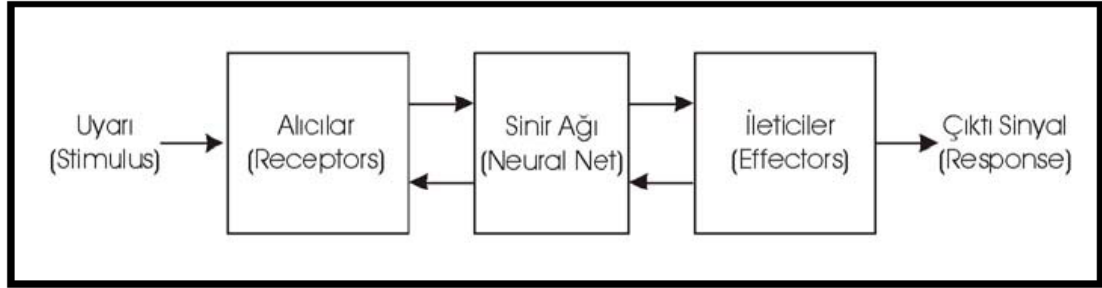
Yapay sinir ağlarının dayandığı ilk hesaplama modelinin temelleri 1940'ların başında araştırmalarına başlayan W.S. McCulloch ve W.A. Pitts'in, 1943 yılında yayınladıkları bir makaleyle atılmış olmuştur. Daha sonra 1954 yılında B.G. Farley ve W.A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturulmuştur. 1960 yılı ise ilk neural bilgisayarın ortaya çıkış yılıdır. 1963 yılında basit modellerin ilk eksiklikleri fark edilmiş, ancak başarılı sonuçların alınması 1970 ve 1980'lerde termodinamikteki teorik yapıların doğrusal olmayan ağların geliştirilmesinde kullanılmasına kadar gecikmiştir.

1970'li yılların sonlarında ve 1980'li yılların başlarındaki ilerleme, yapay sinir ağları alanına ilginin yeniden canlanması bakımından önemlidir. Bu hareketi bir kaç faktör etkilemiştir. Örneğin, ayrıntılı kitaplar ve konferanslar çok farklı alanlarda uzmanlaşmış insanlara bir forum imkânı ve dolayısı ile bir etkileşim sağlamıştır. Akademik programlar oluşturulmuş ve en önemli üniversitelerde dersler açılmıştır. Artan ilgi ile beraber bu alandaki araştırmalara yönelik fonlar da artmış ve enstitüler ortaya çıkmıştır. Sağlanan önemli ilerleme yapay sinir ağları alanında daha ileri araştırmalar için gerekli ilgi ve bilgi birikimini sağlamıştır. Sinir sistemi tabanlı işlemciler oluşturulmakta ve karmaşık problemlerin çözümüne yönelik uygulamalar gelişmektedir. Kısacası, bu alan günümüzde bir geçiş dönemi içinde görülmektedir.

2.3.1 Yapay Sinir Ağlarının Yapısı Elemanları ve Mimarisi

Önceden de belirtildiği gibi, yapay sinir ağları insan beyninden esinlenilmiş yapılardır. Bu yüzden, YSA'ların yapısını anlamak için insan beyninin yapısını ve çalışma şeklini anlamak önemlidir. Bununla birlikte, tahmin edilebileceği gibi insan beyni, karmaşık yapısı ile anlaşılması ve anlatılması zor bir konudur. Hatta beynin çalışma şekli hala tam olarak anlaşılamamıştır.

İnsan beyni, sinir sisteminin merkezini oluşturan temel elemandır. En basit şekilde, sürekli olarak iletilen bilgiyi alır, idrak eder (işler) ve uygun kararları vererek gerekli yerlere iletir. Çok basit görünmekle birlikte aslında oldukça karmaşık olan bu yapının Haykin tarafından kullanılan basit bir gösterimi Şekil 2.8' de sunulmaktadır (Haykin 1999).



Şekil 2.8 Sinir Sisteminin Blok Diyagramı

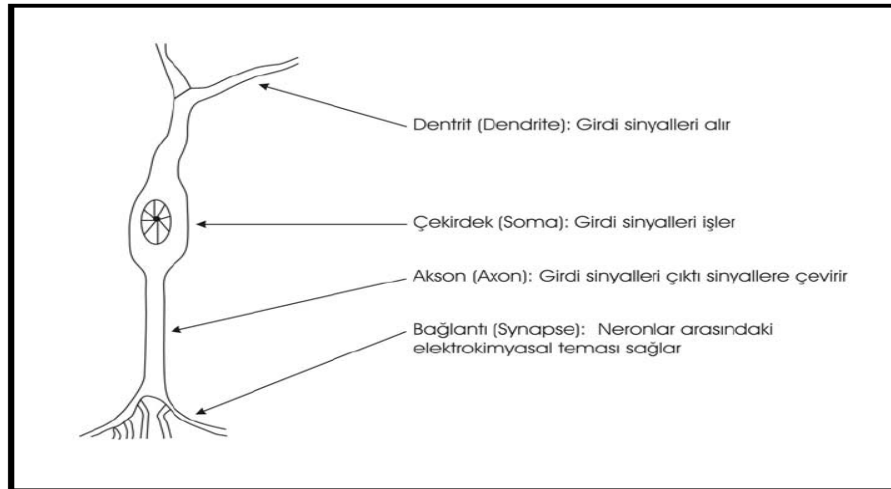
Şekilden takip edilebileceği gibi, dışarıdan veya başka bir organdan gelen işaretler (uyarı) alıcılar yoluyla sinir ağına iletilir. İşaretler burada işleminden geçirilerek çıktı işaretler oluşturulur. Oluşturulan çıktı işaretler ise ileticiler yoluyla dış ortama veya diğer organlara iletilirler. Şekil 2.8' de Sinir Ağı olarak gösterilen ortadaki bölüm, yani beyin, sinir sisteminin merkezi konumundadır. Beynin temel yapı taşları ise sinir hücreleri, diğer bir ifadeyle nöronlardır. Beyin, işlevini birbirleri arasında yoğun bağlantılar bulunan bu yapı taşları ile yerine getirir. Nöronların en belirgin özelliği, vücudun diğer bölümlerinin tersine yeniden üretilmeyen belirli bir hücre türü olmasıdır. Beynin diğer temel yapısal ve fonksiyonel birimleri olan bağlantılar (synapses) nöronlar arası etkileşimi sağlarlar. Dolayısıyla, beynin yapısında bu bağlantılar da önemli bir yer

tutmaktadır. Zaten beynin oldukça etkin çalışan bir organ olmasının temel sebebi de bu yoğun bağlantılı yapıdır. Bu yapı sayesinde beyin, bugünkü bilgisayar teknolojisinden kat kat daha etkin bir şekilde çalışabilmektedir.

Doğal olarak, YSA' ların temel elemanları da yapay nöronlardır. Bu nöronlar, aralarındaki bağlantılar oluşturularak ve tabakalar halinde gruplandırılarak yapay sinir ağları oluşturulmaktadır.

İnsanın beyinsel gücünün bu yapı taşları birkaç genel işleve sahiptirler. Bir biyolojik nöron, temel olarak, diğer kaynaklardan girdiler alır, belirli bir şekilde bunları birleştirir, sonuç üzerinde bir işlem (genelde doğrusal olmayan) uygular ve nihai sonucu üretir. Şekil 2.9 bir nöronun dört temel elemandan oluşan genel yapısını ve bu dört eleman arasındaki ilişkiyi göstermektedir.

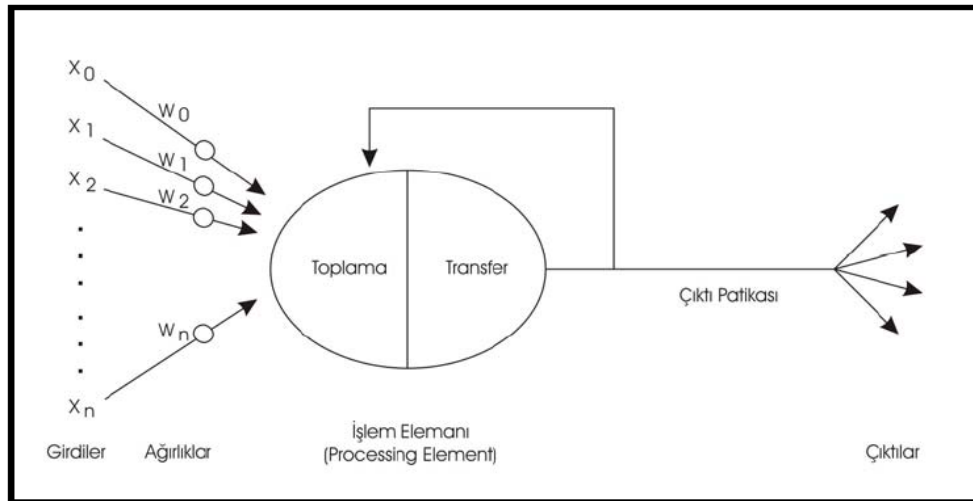
Şekilden de görüldüğü gibi, tüm doğal nöronlar dört temel bileşene sahiptir. Bu bileşenler biyolojik isimleri ile bilinirler: dentrit (dendrite), çekirdek (soma), akson (axon) ve bağlantı veya sinaps (synapse). Dentritler, çekirdeğin saça benzeyen uzantılarıdır ve girdi kanalları olarak işlev görürler. Bu girdi kanalları diğer nöronların sinapsları aracılığıyla girdilerini alırlar. Daha sonra çekirdek, gelen bu işaretleri zaman içinde işler. Çekirdek, bu işlenmiş değeri bir çıktıya dönüştürdükten sonra bu çıktıyı akson ve sinapslar aracılığıyla diğer nöronlara gönderir.



Şekil 2.9: Biyolojik Nöronun Genel Yapısı ve İşlevleri

Yapay sinir ağlarının temel işlem elemanı olan yapay nöronlar, doğal nöronların dört temel fonksiyonunu simüle ederler. Bir yapay nöronun temel yapısı, genel haliyle, Şekil 2.10'da sunulmaktadır.

Şekil 2.10'da girdi değerler $x(i)$ matematiksel sembolü ile gösterilmiştir ve bu gösterimde $i = 0, 1, 2, \dots, n$ değerlerini almaktadır. Bu girdi değerlerin her biri bir bağlantı ağırlığıyla çarpılmaktadır. Bu ağırlıklar ise $w(i)$ ile gösterilmektedir. En basit yapıda, bu çarpımlar toplanır ve bir transfer fonksiyonuna gönderilerek sonuç üretilir. Bu sonuç daha sonra bir çıktıya dönüştürülür. Bu elektronik uygulama değişik toplama fonksiyonları ve transfer fonksiyonları kullanabilir ve farklı ağ yapılarında uygulanabilir.



Şekil 2.10 Yapay Nöronun Genel Yapısı

Şekil 2.10 McCulloch ve Pitts tarafından tanımlanan, biyolojik nöronun basit matematiksel modelinin gösterimi olarak da düşünülebilir. McCulloch-Pitts modeli incelendiğinde, toplama fonksiyonu olarak doğrusal bir fonksiyon ve transfer fonksiyonu olarak birim adım fonksiyonu kullanılmış olduğu görülmektedir. Temelde biyolojik nörona benzer bir şekilde, işlem elemanı toplama fonksiyonu sonucunun belirli bir eşik değerinin altında veya üstünde olmasına göre çıktısını iki ihtimal arasından seçerek oluşturmaktadır (McCulloch ve Pitts 1943). Bu matematiksel modelin fonksiyonel gösterimi aşağıdaki gibidir.

$$y_i = \Psi(g(x)) = \Psi\left(\sum_{j=0}^n w_{ij}x_j - Q_i\right) : x_i = (x_0, x_1, \dots, x_n) \in R \quad (2.7)$$

Bu gösterimde, y_i çıktı değeri, $\Psi(.)$ transfer fonksiyonunu, $g(.)$ toplama fonksiyonunu, w_{ij} bağlantı ağırlıklarını ve Q_i ise i nolu nöron için eşik değerini göstermektedir. $\Psi(.)$ transfer fonksiyonu birim adım fonksiyonudur:

$$\Psi(.) = \begin{cases} 1; g(.) \geq 0 \\ 0; g(.) < 0 \end{cases} \quad (2.8)$$

$g(.)$ toplama fonksiyonu ise x_j girdilerinin doğrusal bir bileşimi olarak düşünülmüştür:

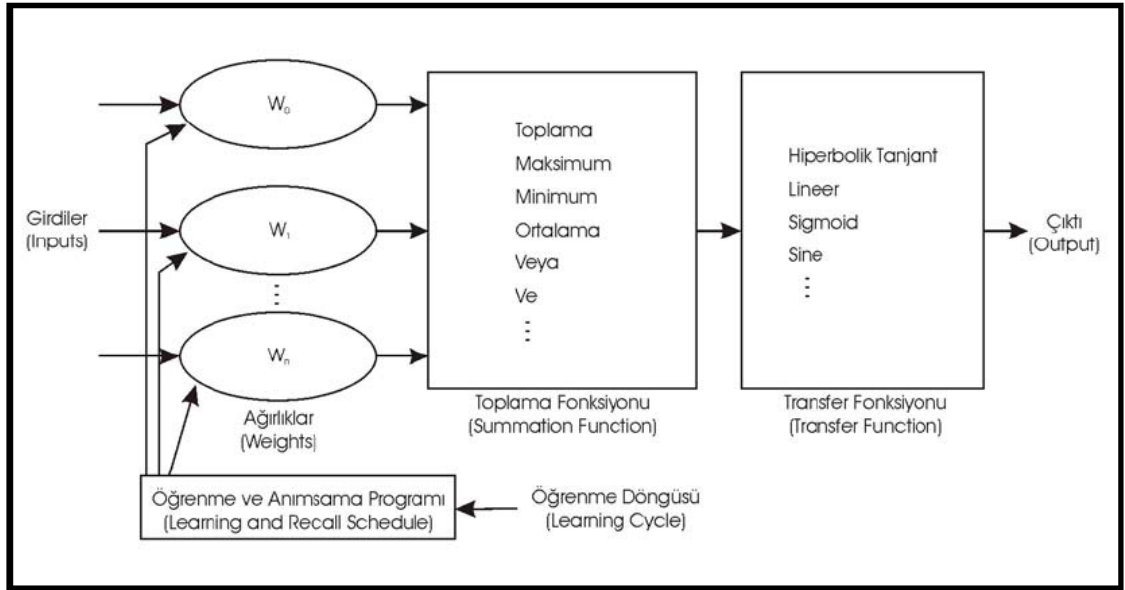
$$g(.) = \sum_{j=0}^n w_{ij} x_j - Q_i \quad (2.9)$$

McCullogh-Pitts modelinde transfer fonksiyonu olarak birim adım fonksiyonu ve toplama fonksiyonu olarak doğrusal bir fonksiyon kullanılmakla birlikte YSA alanındaki gelişmelerle beraber farklı fonksiyonların kullanılabilmesi sağlanmıştır. Şekil 2.11 içinde YSA modellerinde kullanılabilecek farklı fonksiyonlara bazı örnekler sunulmaktadır. Basitçe görülebileceği gibi, hangi fonksiyon kullanılırsa kullanılsın, bir nöron modelinin matematiksel ve fonksiyonel gösterimi değişmeyecektir.

Bu yapıda kullanılan fonksiyonlar temelde uygulama konusuna bağlıdır. Bazı uygulamalar ikili (binary) veri seti gerektirir. Bu uygulamalara örnek olarak konuşma tanımlama ve metin tanıma uygulamaları verilebilir. Bu tür uygulamalar, doğal olarak, sadece toplama fonksiyonu içeren nöronlardan oluşan ağları kullanamaz. Bu ağlar mantıksal fonksiyonları kullanabilir ve bu fonksiyonlar toplama ve transfer fonksiyonlarına dahil edilebilir. Veya bazı uygulamalar basit bir şekilde toplama ve belirli bir eşik değeri ile karşılaştırma şeklinde bir işleme gereksinim duyabilirler. Bu şekilde evet/hayır, doğru/yanlış veya 0/1 gibi iki olası sonuç arasından seçim

yapılabilir. Bazı fonksiyonlar ise girdi verileri zamanla ilişkilendirebilir ve bu şekilde zamana bağlı ağlar oluşturulabilir.

Şekil 2.11 'de, daha önce tanımlanan basit bir yapay nöron yapısının daha detaylı bir şeması sunulmaktadır. Şekilde, girdi değerler işlem elemanına üst sol bölümden girmektedir. İşlemde ilk adım, bu girdi değerlerin her birinin ilgili ağırlıklarla $w(i)$ ağırlıklandırılmalarıdır. Bir nöron genellikle, eşanlı olarak birçok sayıda girdi alır. Her girdinin kendi nispi ağırlığı vardır. Bu ağırlıklar, biyolojik nöronların değişen sinaptik etkililikleri ile aynı görevi üstlenirler. Her iki durumda da, bazı girdiler diğerlerine göre daha önemli hale gelirler. Bu sayede, işlem elemanının bir sinirsel tepki üretmesi işleminde daha fazla etkili olurlar. Ayrıca, ağırlıklar girdi işaretin güçlülüğünü belirleyen adaptif katsayılardır. Yani, girdinin bağlantı gücünün bir ölçüsüdür. Bu bağlantı güçleri, çeşitli eğitime setlerine göre değiştirilebilirler.



Şekil 2.11 Yapay Nöronun Detaylı Yapısı

Ağırlıklandırmadan sonra, bu değiştirilmiş girdiler toplama fonksiyonuna gönderilirler. Toplama fonksiyonunda, adından da anlaşılacağı gibi, genelde toplama işlemi yapılmaktadır fakat birçok farklı işlem çeşidi toplama fonksiyonu için kullanılabilir. Toplama fonksiyonu, bu basit çarpımlar toplamına ek olarak, minimum, maksimum, mod, çarpım veya çeşitli normalizasyon işlemlerinden birisi olabilir.

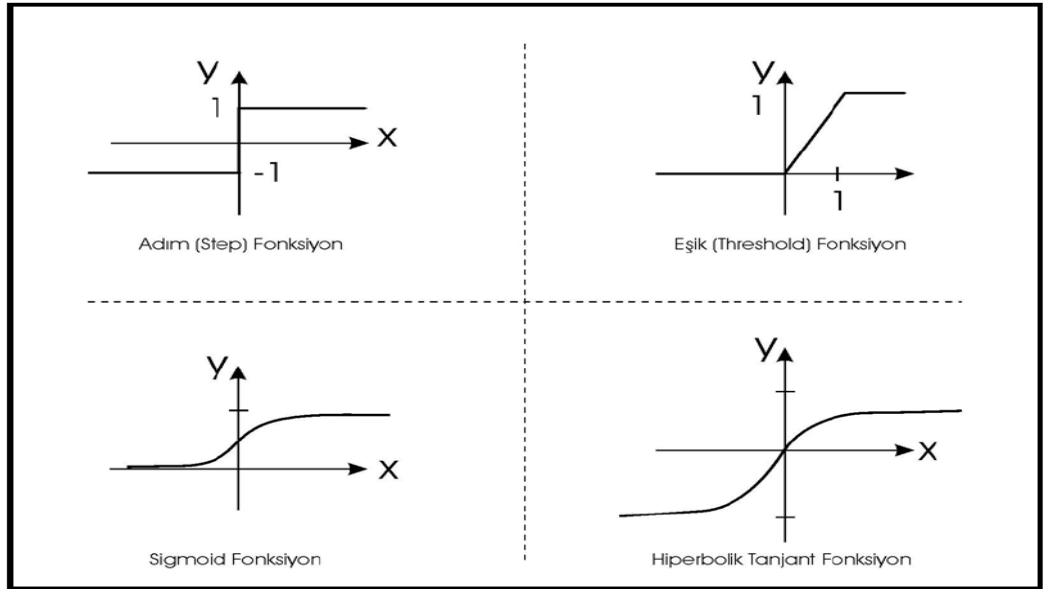
Girdileri birleştirecek olan algoritma genellikle seçilen ağ mimarisine de bağlıdır. Bu fonksiyonlar farklı şekilde değerler üretebilir ve sonra bu değerler ileri doğru gönderilir. Ek olarak, uygulamacı kendi fonksiyonunu oluşturup toplama fonksiyonu olarak kullanabilir. Bazı toplama fonksiyonları, transfer fonksiyonuna iletmeden önce, sonuçları üzerinde ilave işlemler yaparlar. Bu işlem aktivasyon fonksiyonu olarak adlandırılan işlemdir. Bir aktivasyon fonksiyonu kullanmanın amacı toplama fonksiyonu çıktısının zamana bağlı olarak değişmesini sağlamaktır. Fakat aktivasyon fonksiyonu literatürü henüz tam olarak gelişmemiştir. Bundan dolayı, çoğu yapay sinir ağında birim aktivasyon fonksiyonu kullanılmaktadır. Birim aktivasyon fonksiyonu ise bir aktivasyon fonksiyonu kullanılmaması ile aynı anlama gelmektedir. Ayrıca, aktivasyon fonksiyonu, her işlem birimi için ayrı ayrı kullanılan bir bileşenden ziyade ağın genel bir bileşenidir. Yani, oluşturulan bir ağ yapısında, tüm işlem elemanları aynı aktivasyon fonksiyonunu kullanırlar.

Sonraki aşamada toplama fonksiyonunun çıktısı transfer fonksiyonuna gönderilir. Bu fonksiyon, aldığı değeri bir algoritma ile gerçek bir çıktıya dönüştürür. Transfer fonksiyonu genellikle doğrusal olmayan bir fonksiyondur. Doğrusal fonksiyonlar genelde tercih edilmez çünkü doğrusal fonksiyonlarda çıktı, girdi ile orantılıdır. Bu durum, ilk YSA denemelerinin başarısızlıkla sonuçlanmasının temel nedenidir (Minsky ve Papert 1969). Genellikle kullanılan transfer fonksiyonları eşik, sigmoid, hiperbolik tanjant vb. fonksiyonlardır. Bu fonksiyonlar arasında en çok kullanılanlar Şekil 2.12 'de örneklenmiştir. Transfer fonksiyonunun sonucu genellikle işlem elemanının çıktısıdır. Transfer fonksiyonunun çalışma yapısı Şekil 2.12'de sigmoid fonksiyon kullanılarak örneklenmektedir. Sigmoid transfer fonksiyonu, toplama fonksiyonundan gelen ve şekilde 'Toplam' olarak gösterilen değeri alır ve sıfır ile bir arasında bir değere dönüştürür. Sıfır ile bir arasındaki bu değer transfer fonksiyonunun ve dolayısıyla işlem elemanının çıktısıdır ve dış ortama veya girdi olarak başka bir nörona iletilir.

Transfer fonksiyonu işlemi öncesinde, sisteme tekdüze (uniform) dağılmış bir rastsal hata eklenebilmektedir. Bu rastsal hatanın kaynağı ve büyüklüğü, ağın öğrenme işlemi sürecinde belirlenir. Sisteme böyle bir hata teriminin eklenmesinin sebebi, insan

beyninin işlevinin, içinde bulunduğu ortamın şartlarından (örnek olarak sıcak/soğuk olmasından) etkileniyor olmasıdır. Bu yüzden, YSA literatüründe rastsal hata ekleme işlemi “sıcaklık (temperature)” olarak da adlandırılmaktadır. Günümüzde, rastsal hata kullanımı fiilen tam olarak yerleşmemiştir ve halen bir araştırma süreci içerisinde. Ayrıca, bazı ağlarda, transfer fonksiyonunun çıktısı üzerinde başka işlemler, ölçeklendirme ve sınırlandırma, yapılabilmektedir.

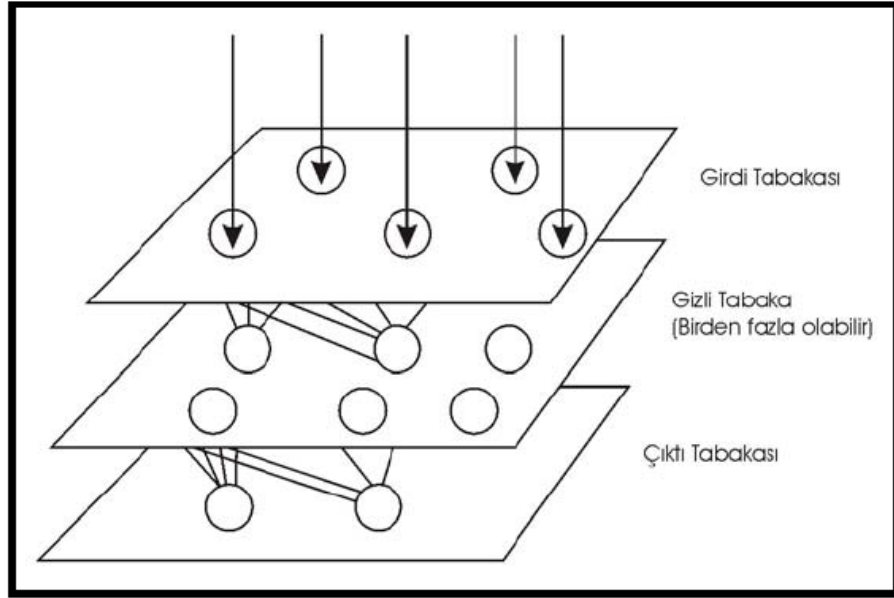
Transfer fonksiyonundan çıkan değer işlem elemanının da çıktısıdır. Fakat bazı durumlarda işlem elemanının bu çıktıyı bir çıktı fonksiyonu ile bir dönüşüme uğratması gerekebilmektedir. Bu çıktı, ağın yapısına göre, girdi olarak başka bir işlem elemanına veya bir dış bağlantıya gönderilir.



Şekil 2.12 En Çok Kullanılan Transfer Fonksiyonları

YSA'lar da, yapay nöronlar basit bir şekilde kümelendirilmektedirler. Bu kümelendirme tabakalar halinde yapılmaktadır ve daha sonra bu tabakalar bir diğerine ilişkilendirilmektedir. Temel olarak, tüm YSA'lar benzer bir yapıya sahiptirler. Böyle bir genel yapı Şekil 2.13'de gösterilmektedir. Bu yapıda, bazı nöronlar girdileri almak için bazı nöronlar ise çıktıları iletmek için dış mekân ile bağlantılı haldedirler. Geri kalan tüm nöronlar ise gizli tabakalardadırlar, yani sadece ağ içinde bağlantıları vardır.

Önemli olan nokta, bir sinir ağının, bir nöronlar yumağından daha komplike olması gerektiğidir. YSA'ların ilk yıllarında, bazı araştırmacılar, nöronlar arasındaki bağlantıları rastsal olarak oluşturmuşlar ve olumsuz sonuçlarla karşılaşmışlardır. Bir yapıyı tasarım etmenin en kolay yolu elemanları tabakalandırmaktır. Burada tabakalandırmanın üç bölümü vardır. Bunlar, nöronları tabakalar halinde gruplandırmak, tabakalar arasındaki bağlantıları gruplandırmak ve son olarak ise toplama ve transfer fonksiyonlarını gruplandırmaktır.



Şekil 2.13: YSA'ların genel yapısı

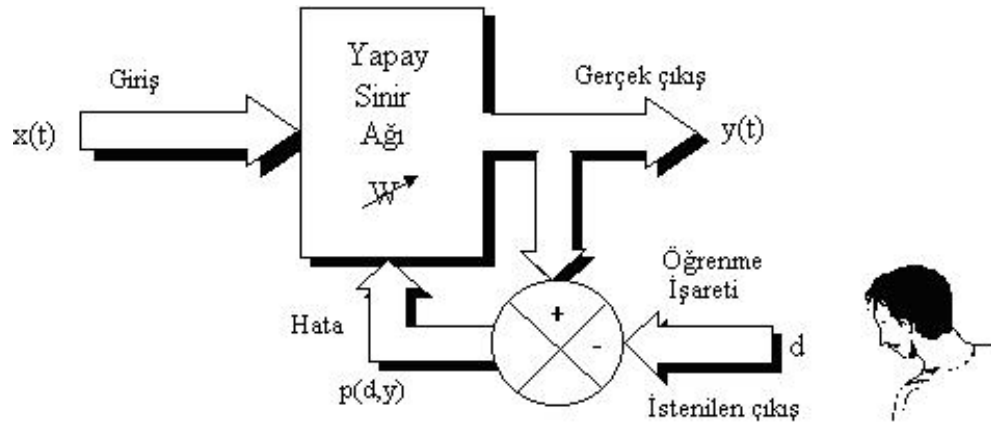
Tek tabaka ya da tek eleman içeren bazı başarılı ağlar oluşturulabilmesine rağmen çoğu uygulamalar en az üç tabaka (girdi tabakası, gizli tabaka ve çıktı tabakası) içeren ağlara ihtiyaç duymaktadır. Girdi tabakası, dışarıdan girdileri alan nöronları içerir. Ayrıca, önemli olan bir nokta, girdi tabakasındaki nöronların girdi değerler üzerinde bir işlem uygulamamasıdır. Sadece girdi değerleri bir sonraki tabakaya iletirler ve bu yüzden de bazı araştırmacılar tarafından ağların tabaka sayısına dahil edilmezler. Çıktı tabakası ise çıktıları dışarı ileten nöronları içeren tabakadır. Girdi ve çıktı tabakaları tek tabakadan oluşurken bu iki tabaka arasında birden fazla gizli tabaka bulunabilir. Bu gizli tabakalar çok sayıda nöron içerirler ve bu nöronlar tamamen ağ içindeki diğer nöronlarla bağlantılıdır.

2.3.2 Yapay Sinir Ağlarının Öğrenme Algoritmalarına Göre Sınıflandırılması

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Birtakım metot ve kurallar, gözlem ve eğitime ile ağdaki ağırlıkların değiştirilmesi sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

2.3.2.1 Eğitici Öğrenme

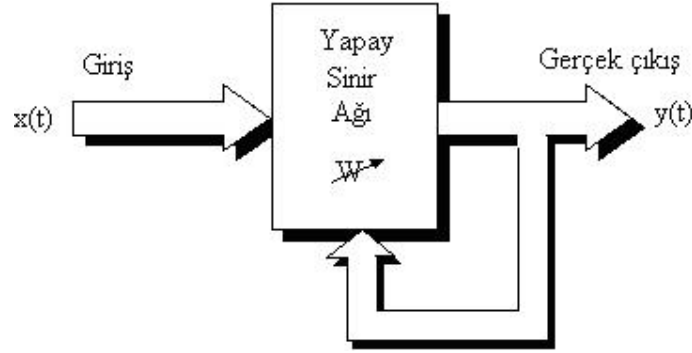
Bu tip öğrenmede, YSA'ya örnek olarak bir doğru çıkış verilir. Bu öğrenmede ağın ürettiği çıktılar ile hedef çıktılar arasındaki fark hata olarak ele alınır ve bu hata minimize edilmeye çalışılır. Bunun için de bağlantıların ağırlıkları en uygun çıkışı verecek şekilde değiştirilir. Bu sebeple danışmanlı öğrenme algoritmasının bir “öğretmene” veya “danışmana” ihtiyacı vardır. Şekil 2.14 'de danışmanlı öğrenme yapısı gösterilmiştir. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.



Şekil 2.14 Danışmanlı öğrenme yapısı

2.3.2.2 Eğitici-siz Öğrenme

Eğitici-siz öğrenme moduna "Kendi kendine öğrenilebilen mod" da denilmektedir. Bu öğrenme modunda eğitim seti kullanılmamaktadır. Ağ, birbirine benzer giriş bilgilerini gruplamakta veya giriş bilgisinin hangi gruba ait olduğunu göstermektedir. Ağ eğitimi için sadece giriş bilgileri yeterli olmakta, referans alınacak(eğitici) bilgiye ihtiyaç duyulmamaktadır. Ağın performansını kendiliğinden izlenmesi söz konusudur. Ağ, giriş işaretlerinin yönüne veya düzenine bakmakta ve ağın fonksiyonuna göre ayarlama yapmaktadır. Ağ kendini nasıl organize edeceği hakkında bir miktar bilgiye sahip olmalıdır.



Şekil 2.15 Eğitici-siz öğrenme yapısı

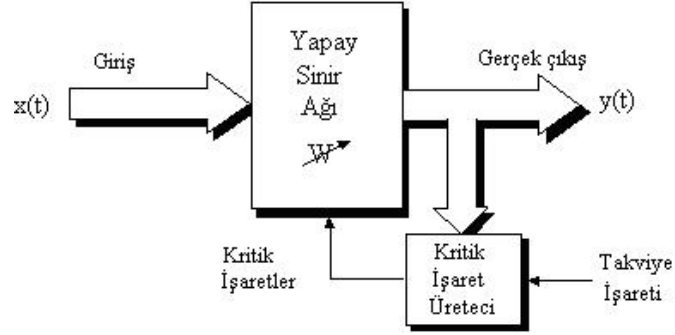
2.3.2.3. Takviyeli Öğrenme

Bu öğrenme kuralı eğitici-li öğrenmeye yakın bir metottur. Denetimsiz öğrenme algoritması istenilen çıkışın bilinmesine gerek duymaz. Hedef çıktıyı vermek için "öğretmen" yerine, burada YSA'ya bir çıkış verilmemekte fakat elde edilen çıkışın verilen girişe karşılık iyiliğini değerlendiren bir kriter kullanılmaktadır. Optimizasyon problemlerini çözmek için Hilton ve Sejnowski'nin geliştirdiği Boltzman kuralı veya genetik algoritma tasdikli öğrenmeye örnektir.

2.3.3 Çok Katmanlı Algılayıcılar

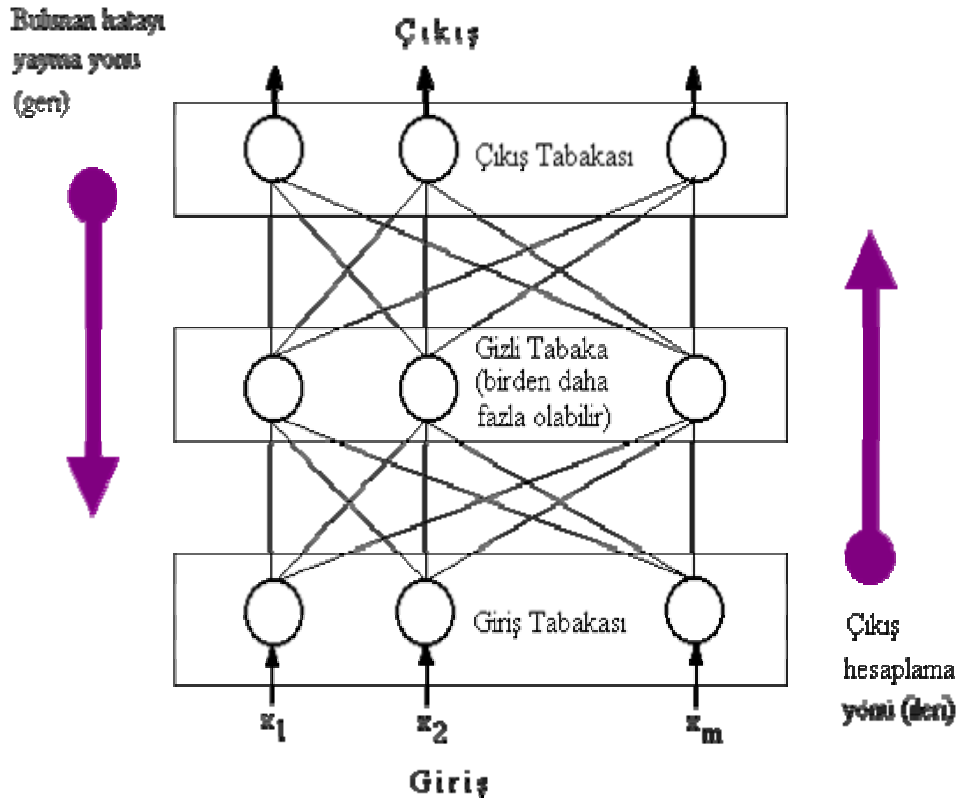
Birçok katmanlı sinir ağı modeli, Şekil 2.17'da gösterilmiştir. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan sinir ağı modeli olmuştur.

Birçok öğretim algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir. Bir MLP modeli, bir giriş, bir veya daha fazla ara ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemler giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı ise, deneme-yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir.



Şekil 2.16 Takviyeli öğrenme yapısı

MLP ağlarında, ağı bir örnek gösterilir ve örnek neticesinde nasıl bir sonuç üreteceği de bildirilir (danışmanlı öğrenme). Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve çıkış katmanından da çıkışlar elde edilir. Kullanılan eğitim algoritmasına göre, ağı çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak hata minimuma düşünceye kadar ağı ağırlıkları değiştirilir.



Şekil 2.17 Çok katmanlı ağ yapısı

2.3.3.1 Geri Yayılım Algoritması (Back Propagation)

Karmaşık verilerin sınıflandırılmasında kullanılan etkin YSA modellerinden birisi; ilk olarak Werbos tarafından düzenlenen daha sonra Parker, Rumelhart ve McClelland tarafından geliştirilen geri yayılım ağıdır (Back Propagation Network). İlk uygulamaları yazılı metinden söz sentezi, robot kollarının kontrolüdür. Bu tezde Konuşmacı Tanıma aşamasında Çok Katmanlı Ağların uygulaması yapılacaktır.

Geri Yayılım Algoritması günümüzde en yaygın kullanılan öğrenimi kolay sonuçları etkin bir YSA'dır. Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğretme algoritmasıdır.

Yayınma ve uyum gösterme (Propagate - Adapt) olmak üzere iki aşamada işlemleri gerçekleştiren geri yayılım algoritması, katmanlar arasında tam bir bağlantının bulunduğu çok katmanlı (Multilayer) , İleri Beslemeli (Feed Forward) ve Denetimli (Supervised) olarak eğitilen bir YSA modelidir.

Geri yayılım algoritması (Backpropagation) birçok uygulamalarda kullanılmış en yaygın öğrenme algoritmasıdır. Anlaşılması kolay ve tercih edilen öğretme algoritmasıdır. Bu algoritma hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayımlı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır. Bir geri yayımlı ağ modelinde giriş, gizli ve çıkış olmak üzere 3 katman bulunmakla birlikte problemin özelliklerine göre gizli katman sayısını artırabilmek mümkündür.

Giriş katmanı; giriş veri gruplarının ağa sunulduğu terminallerdir. Bu katmanda nöron sayısı giriş veri sayısı kadardır ve her bir giriş nöronu bir veri alır. Burada veri işlenmeden bir sonraki katman olan gizli katmana geçer.

Gizli katman; ağın temel işlevini gören katmandır. Bazı uygulamalarda ağda birden fazla gizli katman bulunabilir. Gizli katman sayısı ve katmandaki nöron sayısı probleme göre değişir, tamamen ağ tasarımcısının kontrolündedir ve onun tecrübesine bağlıdır. Bu katman; giriş katmanından aldığı ağırlıklandırılmış veriyi probleme uygun bir fonksiyonla işleyerek bir sonraki katmana iletir. Bu katmanda gereğinden az nöron kullanılması giriş verilerine göre daha az hassas çıkış elde edilmesine sebep olur. Aynı şekilde gerektiğinden daha çok sayıda nöron kullanılması durumunda da aynı ağda yeni tip veri gruplarının işlenmesinde zorluklar ortaya çıkar.

Çıkış katmanı; ağının en uç katmanıdır. Gizli katmandan aldığı veriyi ağın kullandığı fonksiyonla işleyerek çıktısını verir. Çıkış katmanındaki nöron sayısı ağa sunulan her verinin çıkış sayısı kadardır. Bu katmandan elde edilen değerler yapay sinir ağının söz konusu problem için çıkış değerleridir.

Geri yayımlı sinir ağında bir katmandan bir başka katmana, aradaki katmanı atlayarak geçebilmek mümkün değildir.

Bir giriş verisinin ağın ilk katmanında yer alan düğümlere uygulandığında en üst katman olan çıkış katmanına erişinceye kadar bu veri üzerinde çeşitli işlemler

gerçekleştirilir. Bu işlemlerin sonucunda elde edilen fiili çıktı, olması gereken çıktı ile karşılaştırılır. Fiili ve olması gereken değerler arasındaki fark her çıktı düğümü için bir hata işareti olarak hesaplanır. Hesaplanan hata işaretleri her çıktı düğümüne karşı gelen ara katmandaki düğümlere aktarılır. Böylece ara katmandaki düğümlerin her biri toplam hatanın sadece hesaplanan bir kısmını içerir. Bu süreç her katmandaki düğümler toplam hatanın belirli bir kısmını içerecek şekilde giriş katmanına kadar tekrarlanır. Elde edilen hata işaretleri temel alınarak bağlantı ağırlıkları her düğümde yeniden düzenlenir. Bu düzenleme tüm verilerin kodlanabileceği bir duruma ağın yakınsamasını sağlar.

İleri besleme safhasında, giriş tabakasındaki nöronlar veri değerlerini doğrudan gizli katmana iletirler. Gizli katmandaki her bir nöron kendi giriş değerlerini ağırlandırarak toplam değer hesap ederler ve bunları bir taşıma fonksiyonu ile işleyerek bir ileriki tabakaya veya doğrudan çıkış katmanına iletirler. Katmanlar arasındaki ağırlıklar başlangıçta rasgele küçük rakamlardan seçilir.

Çıkış katmanındaki, her bir nöron ağırlıklandırılmış değeri hesaplandıktan sonra, bu değer yine taşıma fonksiyonu ile karşılaştırılarak mevcut hata minimize edilmeye çalışılır. Hata değeri belli bir mertebeye ininceye kadar iterasyon işlemine devam edilir ve böylece ağın eğitim aşaması tamamlanmış olur. Katmanlar arasındaki bağlantılardaki ağırlık değerleri eğitimi tamamlamış ağdan alınarak deneme safhasında kullanılmak üzere saklanır.

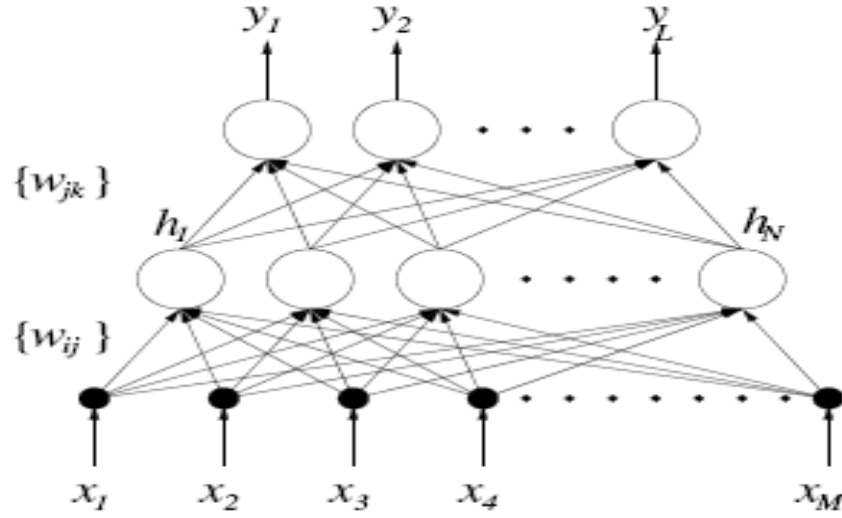
Geri yayılım algoritması, gradyen azalan ve MLP'leri eğitmede en çok kullanılan temel bir algoritmadır. Bu algoritma tezde konuşmacı tanıma için oluşturulan Çok Katmanlı Ağ yapısını eğitmekte kullanılmıştır. Bu algoritmayı anlayabilmek için her adımı aşağıdaki gibi anlatılmıştır.

2.3.3.2 Ağ Yapısının Tanımı

Ağ yapısındaki elemanları ve birbiriyle bağlantılarını açıklamak için Şekil 2.18 kullanılmıştır. Şekil 2.18 'de siyah halkalarla gösterilen giriş katmanıdır. Burada bir işlem yapılmaz girişleri ağa yönlendirmek amacıyla bu şekilde gösterilir. Giriş bilgileri

w_{ij} ağırlıkları ile çarpılarak her bir saklı katmana iletilir. Her bir node için girişlerin ağırlıklı toplamaları hesaplanır ve bu transfer fonksiyonu için giriş olarak kullanılır. Genel olarak çok katmanlı ağlarda transfer fonksiyon olarak sigmoid kullanılır:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$



Şekil 2.18 Çok katmanlı ağ yapısı elemanları

Saklı katmandaki bilgiler daha sonra w_{jk} ağırlıkları kullanılarak çıkış katmanına iletilir. Her bir çıkışın değeri saklı katmanın değerinin ağırlıklı toplamaları alınarak hesaplanır. Örneğin çıkış ağındaki herhangi bir k nodunu düşünelim. Çıkış değeri y_k kabul edilirse;

$$y_k = f(\text{net}_k) \quad (2.11)$$

Burada net_k saklı katmandaki h_j çıkış nodelarının ağırlıklı toplamalarıdır ve aşağıdaki gibi matematiksel olarak ifade edilebilir.

$$\text{net}_k = \sum_{j=1}^N w_{jk} h_j \quad (2.12)$$

Aynı şekilde her bir saklı katmandaki j nodunun çıkışı aşağıdaki matematiksel ifadeyle gösterilir.

$$h_j = f(\text{net}_j) \quad (2.13)$$

$$\text{net}_j = \sum_{i=1}^M w_{ij} x_i \quad (2.14)$$

x_i ağ yapısının girişlerini göstermektedir.

2.3.4 Toplam Karesel Hata ve Gradyent Azalma

İşlem kolaylığı olması açısından ağı girişlerini x giriş vektörü şeklinde gösterilebilir.

$$X = [x_1 \ x_2 \ \dots \ x_M]^T$$

Aynı şekilde ağı çıkışlarını y çıkış vektörü şeklinde gösterilebilir.

$$Y = [y_1 \ y_2 \ \dots \ y_L]^T$$

Ağ için eğitim vektörlerinin kümesi K adet giriş vektörü ve bunlara ait beklenen çıkışların oluşturduğu çiftlerden oluşan kümedir.

$$\text{Eğitim Seti} = \{(x_1, d_1), (x_2, d_2), \dots, (x_l, d_l), \dots, (x_k, d_k)\}$$

Ağın girişine X eğitim setinden herhangi bir giriş vektörü uygulandığında, ağ yapısı çıkışta (2.11)'deki denkleme göre gerçek çıkış değeri oluşturur. Giriş vektörüne göre karesel hata her bir noddaki karesel hataların toplamı alınarak bulunur.

$$[\text{Karesel Hata}] = \frac{1}{2} \sum_{k=1}^L (y_k - d_k)^2 \quad (2.15)$$

Toplam karesel hata ifadesi E eğitim setindeki tüm giriş/çıkış çiftleri için aşağıdaki gibi gösterilir.

$$E = \frac{1}{2} \sum_{l=1}^K \sum_{k=1}^L (y_{kl} - d_{kl})^2 \quad (2.16)$$

Sinir ağını eğitirken asıl amaç uygun ağırlık vektörlerini bularak toplam karesel hatayı minimize etmektir. Ağırlık vektörleri gradyent azalan algoritma ile eğitirler. Bu algoritmayı matematiksel olarak her bir ağırlık vektörünü Δw gibi küçük miktarla hatanın azalma yönüne bakarak güncelleme yöntemi olarak ifade edilebilir.

$$w(t+1) = w(t) + \Delta w(t)$$

$$\Delta w(t) = -\varepsilon \frac{\partial E}{\partial w t} \quad (2.17)$$

$w(t)$ t anındaki ağırlık vektörünü gösterir, $w(t+1)$ ise güncellenmiş ağırlık vektörüdür. 2.17 eşitliğindeki denklem Genel Delta Kuralı olarak da ifade edilir.

2.3.5 Hatanın Kısmı Türevlerinin Ağırlık Vektörlerine Göre Bulunması

2.3.5.1 Çıkış Katmanı ile Saklı Katman Arasındaki Ağırlıklar

İlk önce j saklı katman nodu ile k çıkış katman nodu arasındaki ağırlık vektörünü w_{jk} olduğunu düşünülür. Burada $\frac{\partial E}{\partial w_{jk}}$ bulmaya çalışılmalıdır. Bunun için Chain kuralı kullanarak aşağıdaki eşitlik yazılabilir.

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} \quad (2.18)$$

Denklem 2.16 aşağıdaki eşitliği gösterir.

$$\frac{\partial E}{\partial y_k} = y_k - d_k \quad (2.19)$$

Kısmı türevde sadece belli bir öge dikkate alınır. Burada k çıkış nodu w_{jk} ağırlık vektörüne bağlıdır.

$$\frac{\partial y_k}{\partial net_k} = y_k (1 - y_k) \quad (2.20)$$

Denklem 2.20 sigmoid fonksiyonunun türevi alınarak bulunmuştur.

$$\frac{\partial net_k}{\partial w_{jk}} = h_j \quad (2.21)$$

Denklem 2.12'nin türevi alındığında denklem 2.21 elde edilir. Buraya kadar elde edilen kısmi türevler denklem 2.18'de yerine konarak aşağıdaki eşitlik elde edilir.

$$\frac{\partial E}{\partial w_{jk}} = (y_k - d_k) y_k (1 - y_k) h_j \quad (2.22)$$

Denklem 2.22 hatanın w_{jk} ağırlık vektörüne göre kısmi türevini göstermektedir. Bu denklem kullanılarak saklı katmanla çıkış katmanı arasındaki ağırlık vektörlerinin gradient azalan yöntemle güncellenmesi yapılabilir.

2.3.5.2 Giriş Katmanı ile Saklı Katman Arasındaki Ağırlıklar

Giriş katmanı ile saklı katman arasındaki ağı w_{ij} olarak gösterebilir. Tekrar eşitlik 2.12' den başlayarak $\frac{\partial E}{\partial w_{ij}}$ 'ye Chain kuralı uygulayarak kısmi türev alınır.

$$\frac{\partial E}{\partial w_{ij}} = \sum_{k=1}^L \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{h_j} \frac{h_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (2.23)$$

Çıkış katmanlarının y_k değeri incelendiğinde her bir çıkışın w_{ij} ağırlık vektörüne bağlı olduğu görülür. Bu kısmi türevin ilk iki terimi daha önce çıkarılmıştır. Bir sonraki terim için eşitlik 2.12' i kullanabilir.

$$\frac{\partial net_k}{\partial h_j} = w_{jk} \quad (2.24)$$

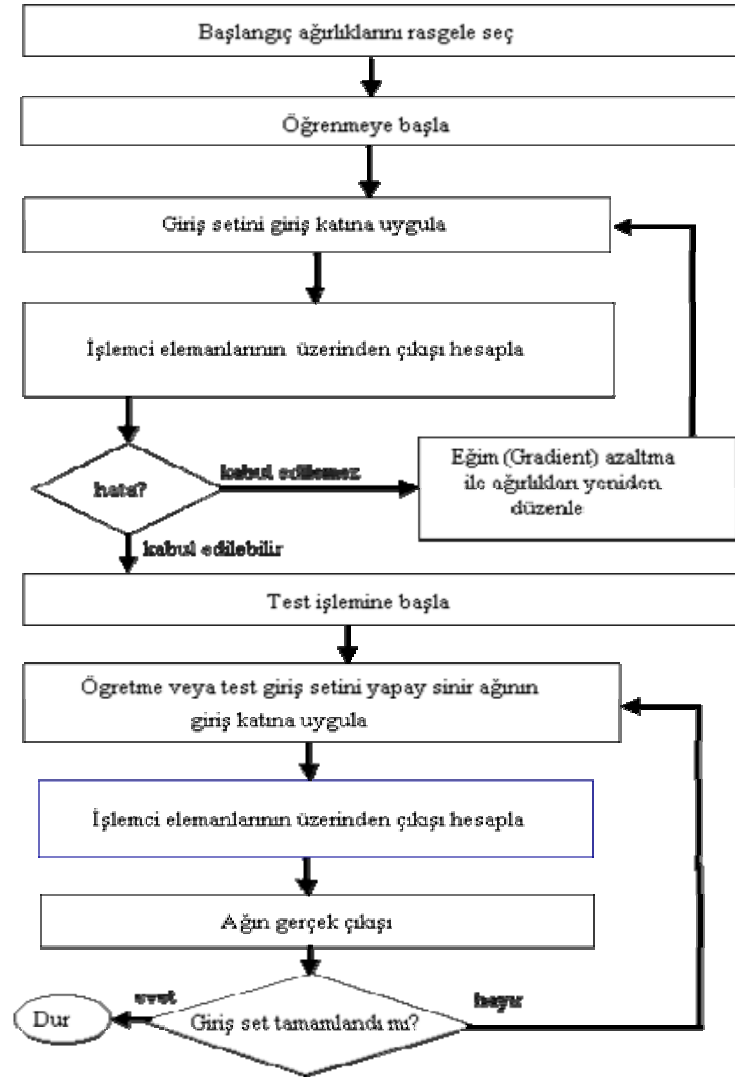
$$\frac{\partial h_j}{\partial net_j} = h_j(1 - h_j) \quad (2.25)$$

$$\frac{\partial net_j}{\partial w_{ij}} = x_i \quad (2.26)$$

Yukarıda verilen eşitlikleri Denklem 2.23’de yerine koyarsak aşağıdaki eşitlik elde edilir.

$$\frac{\partial E}{\partial w_{ij}} = \sum_{k=1}^L (y_k - d_k) y_k (1 - y_k) w_{jk} h_j (1 - h_j) x_i \quad (2.27)$$

Eşitlik 2.27’de giriş katmanı ile saklı katman arasındaki ağırlıklara göre kısmi türevi çıkarılmıştır. Eşitlik 2.22 ve 2.27 kullanılarak çok katmanlı bir ağda Gradient Azalan Algoritma uygulanabilir. Bu algoritmayı daha anlaşılır olabilmesi için Şekil 2.19’de akış şeması kullanılabilir.



Şekil 2.19 Çok Katmanlı Bir Ağın Geri Yayılım Akış Şeması

2.3.6 Levenberg-Marquardt Algoritması

İleri beslemeli ağlarda kullanılan öğrenme algoritmaları, performans fonksiyonunu en küçük yapacak ağırlıkları ayarlayabilmek için, performans fonksiyonunun gradyenini kullanırlar. Geriye yayılım algoritması da, ağ boyunca gradyen hesaplamalarını geriye doğru yapar. En basit geriye yayılım öğrenme algoritması gradyen azalması algoritmasıdır. Bu algoritmada ağırlıklar, performans fonksiyonunun azalması yönünde ayarlanır. Fakat bu yöntem, pek çok problem için çok yavaş kalmaktadır. Bu algoritmadan daha hızlı, daha yüksek performanslı algoritmalar da vardır.

Hızlı algoritmalar genel olarak iki kategoriye ayrılabilir. İlk kategorideki algoritmalar, deneme yanılma tekniklerini kullanarak, standart gradyen azalması (steepest descent) yönteminden daha iyi sonuçlar verebilir. Deneme-yanılma işlemlerini kullanan geriye yayılım algoritmaları; momentum terimli geriye yayılım, öğrenme hızı değişen geriye yayılım ve esnek geriye yayılım algoritmalarıdır. Hızlı algoritmaların ikinci kategorisin deki algoritmalar, standart sayısal optimizasyon yöntemlerini kullanır. Bu algoritmalar; eşlenik gradyen öğrenme algoritması, Newton öğrenme algoritmaları ve Levenberg – Marquardt öğrenme algoritmasıdır.

Öğrenme algoritmaları, kendisinden önce geliştirilen algoritmalara alternatif olarak ortaya çıkmıştır ve önceki algoritmaların iyi yönlerini geliştirip, kötü yönlerini azaltmaya yönelmiştir. Levenberg – Marquardt algoritması da, Newton ve Gradyen Azalması algoritmalarının en iyi özelliklerinden oluşur ve kısıtlamalarını ortadan kaldırır.

Eşlenik gradyen öğrenme algoritmasına alternatif olarak sunulan Newton yöntemlerinde, temel adım Hessian matrisini elde etmektir. Hessian matrisi, performans fonksiyonunun ağırlıklara göre ikinci dereceden türevlerinden oluşan bir matristir.

$$H(n) = \frac{\partial^2 E(n)}{\partial w^2 (n-1)} \quad (2.28)$$

Burada H Hessian matrisi, E performans fonksiyonu, w ağırlık sinaptik ağırlığıdır. Performans fonksiyonu, duruma göre toplam ani hata veya ortalama karesel hata olarak alınabilir. Bu çalışmada, ileri beslemeli ağların çoğunda olduğu gibi, performans fonksiyonu olarak, ortalama karesel hata kullanılmıştır.

$$E(n) = E_{\text{ort}}(n) = \frac{1}{N} \sum_1^N \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.29)$$

Burada N eğitim kümesindeki toplam örüntü sayısını, e_j hata işaretini, C ağın çıkış katmanındaki bütün nöronları içeren kümeyi göstermektedir. d_j istenen değer, y_j ağın çıkışı olmak üzere hata işareti,

$$e_j(n) = d_j(n) - y_j(n) \quad (2.30)$$

olarak bulunabilir. Hessian matrisi hesaplandıktan sonra, tersi bulunarak ağırlıklar yenilenebilir. Ancak Hessian matrisi çok karmaşık ve ileri beslemeli bir yapay sinir ağı için hesaplanması zor bir matristir. Newton yöntemlerinin içinde, ikinci dereceden türevlerin hesaplanmadan işlem yapılan bir sınıf vardır. Bu sınıftaki yöntemler, quasi – Newton yöntemleri olarak adlandırılırlar. Quasi – Newton yöntemleri, algoritmanın her iterasyonunda, Hessian matrisinin yaklaşık bir şeklini kullanır.

Levenberg – Marquardt algoritması da quasi – Newton yöntemleri gibi, Hessian matrisinin yaklaşık değerini kullanır. Levenberg – Marquardt algoritması için Hessian matrisinin yaklaşık değeri şu şekilde bulunabilir:

$$H(n) = J^T(n)J(n) + \mu I \quad (2.31)$$

2.31 denklemindeki μ Marquardt parametresi, I ise birim matristir. Burada J matrisi, Jakobien matrisi olarak adlandırılır ve ağ hatalarının ağırlıklara göre birinci türevlerinden oluşur:

$$J(n) = \frac{\partial e(n)}{\partial w(n-1)} \quad (2.32)$$

2.32 denkleminde e, ağ hataları vektörüdür. Jakobien matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir. Ağın gradyeni,

$$g(n) = J^T(n)e(n) \quad (2.33)$$

olarak hesaplanır ve ağırlıklar (2.34) denkleminde göre değiştirilir:

$$w(n+1) = w(n) - [H(n)]^{-1} g(n) \quad (2.34)$$

Marquardt parametresi, μ , skaler bir sayıdır. Eğer μ sıfırsa, bu yöntem yaklaşık Hessian matrisini kullanan Newton algoritması; eğer μ büyük bir sayı ise, küçük adımlı gradyen azalması yöntemi haline gelir. Newton yöntemleri, en küçük hata yakınlarında daha hızlı ve kesindir. Her başarılı adımdan sonra, yani performans fonksiyonunun azalmasında μ azaltılır ve sadece deneme niteliğindeki bir adım performans fonksiyonunu yükseltecekse μ artırılır. Bu yöntemle, algoritmanın her iterasyonunda, performans fonksiyonu daima azaltılır.

2.4 TIMIT Veri Tabanı

TIMIT ses veritabanı ses örneklerinin akustik ve fonetik yapıları hakkında bilgi edinmek amacıyla oluşturulmuştur. Bu ses örnekleri otomatik konuşma ve konuşmacı tanıma sistemleri için yapılan araştırma geliştirme çalışmalarında sıkça kullanılmaktadır. Bu tezde kullanılan ses örnekleri de TIMIT' den alınmıştır.

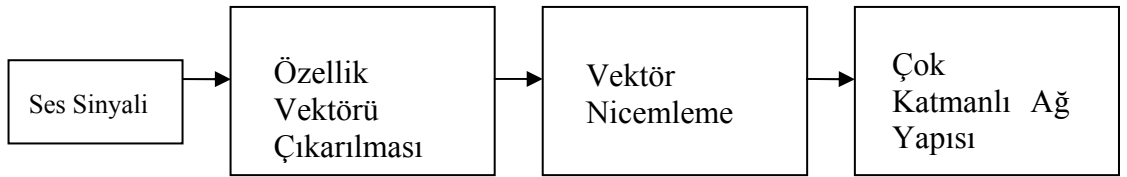
TIMIT üç ayrı kuruluşun ortak çalışması sonucu DARPA-ISTO sponsorluğunda oluşturulmuştur. TIMIT oluşturulması çalışmasında Massachusetts Teknoloji Enstitüsü (MIT), Stanford Araştırma Enstitüsü (SI) ve Texas Instrument (TI) beraber çalışmıştır.

TIMIT, Birleşik Devletlerde bulunan 8 farklı lehçedeki 630 insandan alınmış cümlelerden oluşmaktadır. Her kişiden 10 farklı cümle alarak toplam 6300 cümle elde edilmiştir. Ses verileri yüksek kaliteli mikrofon kullanılarak sessiz ortamlarda kaydedilmiştir.

3. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışmasında TIMIT veri öbeğinden alınan ses örnekleri kullanılarak oluşturulan yapay sinir ağ yapısı MATLAB paket programı kullanılarak eğitilmiştir. Eğitilen ağ yapısı konuşmacılardan alınmış test verileriyle test edilmiş ve konuşmacı tanımadaki başarı oranı tespit edilmeye çalışılmıştır.

Bu tezin genel amacı konuşmacı tanıma sistemini etkileyen parametrelerin belirlenmesi ve bunun tanıma başarı oranına olan etkilerinin araştırılmasıdır. Bu amaçla konuşmacı tanıma yapısı 3 ana alt yapıya bölünerek hazırlanmıştır.



Şekil 3.1 Konuşmacı tanımanın 3 temel yapısı

İlk aşamada TIMIT ses verisinde DR1 lehçesindeki kişilerin 10 adet cümlesi eğitim ve test için iki kısma ayrılır. Kişilere ait 8 adet cümle eğitim aşamasında kullanılırken 2 adet cümle ise test için kullanılmaktadır.

Konuşmacı tanıma kişilere ait ses örneklerini işlenerek özellik vektörleri çıkartılır. Daha önceki bölümlerde özellik vektörü çıkarma yöntemleri ayrıntılı olarak incelenmiştir. Bu yöntemlerden son yıllarda en çok kullanılan ve konuşmacı tanıma başarılı olan Mel Frekansı Kepstrum Katsayıları kullanılmıştır.

Yapay Sinir Ağlarında özellik vektörlerini kullanmak istendiğinde özellik vektörlerinin boyutunun büyük olması nedeniyle birçok problemle karşılaşmıştır. Bunların başında tanıma işlemi yapılacak kişi sayısı gelmektedir. Kişi sayısının artması

nedeniyle kullanılacak eğitim vektörü seti büyümektedir. Buda yapay sinir ağlarında eğitimin yavaşlamasına hatta bellek problemlerine neden olmaktadır.

Bu problemi ortadan kaldırmak amacıyla vektör nicemleme yöntemi uygulanmıştır. Özellik vektörü LBG algoritması kullanılarak kod kitapları haline getirilmiştir. Bu şekilde özellik vektörlerinin boyutu küçültülmüş ve daha çok kişi ile konuşmacı tanıma sistemi kurmaya imkân sağlamıştır. Vektör nicemleme işleminin önemli dezavantajlarından biri özellik vektörlerinin ayırt etme etkinliğini azaltmasıdır. Bu nedenle tez çalışması sırasında başarı oranını doğrudan etkileyen parametreler arasında ilk sırayı almaktadır.

Bu tezde kullanılan yapay sinir ağ yapısı çok katmanlı algılayıcılardan oluşmaktadır. Çok katmanlı ağlar giriş katmanı, saklı katmanlar ve çıkış katmanından oluşmaktadır. Konuşmacı tanımada kullanılan yapı 3 katmandan oluşmaktadır. Şekil 3.1’ de üç katmanlı ağ yapısı gösterilmektedir.

Çok katmanlı yapıda giriş katmanındaki nöron sayısı kullanılacak olan MFCC katsayısına bakılarak seçilir. Çıkış katmanı ise konuşmacı setindeki kişi sayısına göre seçilir. Saklı katmanlarda kullanılan nöron sayısı ise test aşamasında değiştirilerek konuşmacı tanıma olan etkisi incelenmiştir.

3.1 Başarı Oranını Etkileyen Parametreler

Bu tezde kullanılan 3 ana yapı ayrı ayrı incelendiğinde, her bir yapının kendine ait farklı parametreleri olduğu görülmektedir. Bu durum konuşmacı tanıma sisteminin birçok parametreden etkilenebileceğini göstermektedir. Başarılı bir tanıma yapılabilmesi için en uygun parametre değerlerinin tespit edilmesi gerekmektedir. Bu şekilde sistemin ideal çalışma şartları için en uygun parametre değerleri belirlenebilir.

Konuşmacı tanıma sisteminin ilk adımı olan özellik vektörü çıkarıldıktan sonra ikinci aşama olarak oluşan vektörün boyutunun azaltılması çalışması yapılmıştır.

Bu ikinci aşamada çıkarılan özellik vektörleri, vektör nicemleme işleminden geçirilerek kullanılmaktadır. Bu aşamada oluşturulacak kod kitabının boyutu değiştirilerek vektör nicemlemenin konuşmacı tanımaya etkisi incelenebilir.

Kod kitabındaki vektörlerin boyutunun düşmesi vektör nicemlemenin bozulma hatasının yükselmesine neden olmaktadır. Bu durum özellik vektörlerinin tanımda olan başarısını etkilemektedir. Bu tezde özellik vektörleri 16, 32, 64 boyutlarında kod kitabı oluşturularak kullanılmıştır.

Son aşama olan yapay sinir ağ sistemi birçok parametre içermektedir. Giriş ve çıkış nöron sayılarının kullanılan MFCC'ler ile ve kişi sayısı ile orantılı olması nedeniyle değişiklik yapılamamaktadır. Saklı katmanda kullanılacak nöron sayısı seçimi için kesin bir tanım bulunmaması nedeniyle nöron sayısı değiştirilerek başarıya etkisi incelenebilir. Ayrıca saklı katmanların sayısı ve bu katmanlardaki nöron sayılarının konuşmacı tanıma başarı oranında yapacağı değişikliklere bakılabilir.

Yapay sinir ağları giriş vektörlerini alarak çıkıştaki hedef vektörleri arasındaki hata oranına bulur bu hata oranını minimize etmek için katmanlarda nöronlar arasında bulunan ağırlık vektörlerini sayısal değerlerini değiştirir. Bu eğitim algoritmasında ağırlıkların hataya göre değişimini etkileyen öğrenme oranı katsayısı eğitim algoritmasını doğrudan etkileyen bir parametredir. Bu parametrenin değişimi çok katmanlı ağ yapısının istendiği gibi eğitilmesinde etkili olacağından konuşmacı tanıma oranını da etkileyecektir.

Çok katmanlı ağ yapısında her bir katmanda farklı transfer fonksiyonlar kullanılabilir. Konuşmacı tanımda amaç çıkış katmanında, test edilen kişiye ait nöronun çıkış değerinin 1 olması diğer kişilere ait nöronların değerinin ise 0 ya da -1 olmasıdır. Bu nedenle hiperbolik tanjant sigmoid veya sigmoid fonksiyon kullanılabilir.

Konuşmacı tanıma sistemlerinde önemli unsurlardan biride eğitim ve test süresidir. Konuşmacı tanıma başarısını etkileyen parametreler incelenirken her bir parametrenin eğitim ve test süresine olan etkisi de incelenmiştir. Ayrıca eğitim ve test sürelerini

doğrudan etkileyen ağın eğitimi için kullanılan iterasyon sayısı değiştirildiğinde tanıma başarısının nasıl değiştiği incelenmiştir.

3.1.1 Konuşmacı Tanıma Testleri

Bu tez çalışmasında Matlab paket programı kullanılarak kişileri tanıma sistemi geliştirilmiştir. Bu tanıma işleminin en başarılı olduğu parametreleri bulmak amacıyla geliştirilen sistem üzerinde testler yapılmıştır.

3.1.1.1 Kod Vektörü Boyutunun Başarıya Etkisi

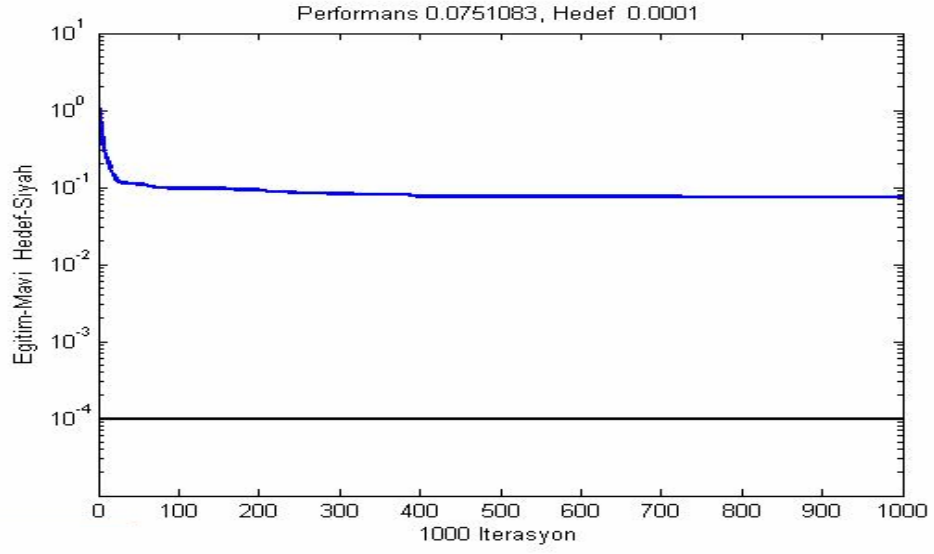
Bu testin amacı farklı kod vektörü boyutlarında oluşturulan sistemin başarı oranının tespit edilmesidir. Kod vektör boyutları 16,32 ve 64 olarak değiştirilmiştir. Yapılan testin diğer parametre değerleri ve başarı oranları Çizelge 3.1’de verilmiştir.

Özellik vektörlerinin nicemlenmesiyle elde edilen kod vektörleri, konuşmacı tanımda yapay sinir ağlarının kullanılmasında kolaylıklar sağlamıştır. Fakat bunun yanı sıra özellik vektörlerinin boyutunun küçülmesi, kişileri ayırt etmede başarı oranını düşürmektedir.

Yapılan ilk testte saklı katmandaki nöron sayısı 10 ve katmanlardaki transfer fonksiyonlar hiperbolik tanjant sigmoid olarak seçilmiştir. Burada kod vektörünün boyutu ve öğrenme oranı değiştirilerek konuşmacı tanıma başarı oranına etkisi incelenmiştir. Yapay sinir ağı eğitimi için program 1000 iterasyon ile çalıştırılmıştır.

Çizelge 3.1 Kod vektör boyutu 16 iken farklı öğrenme oranlarında test sonuçları

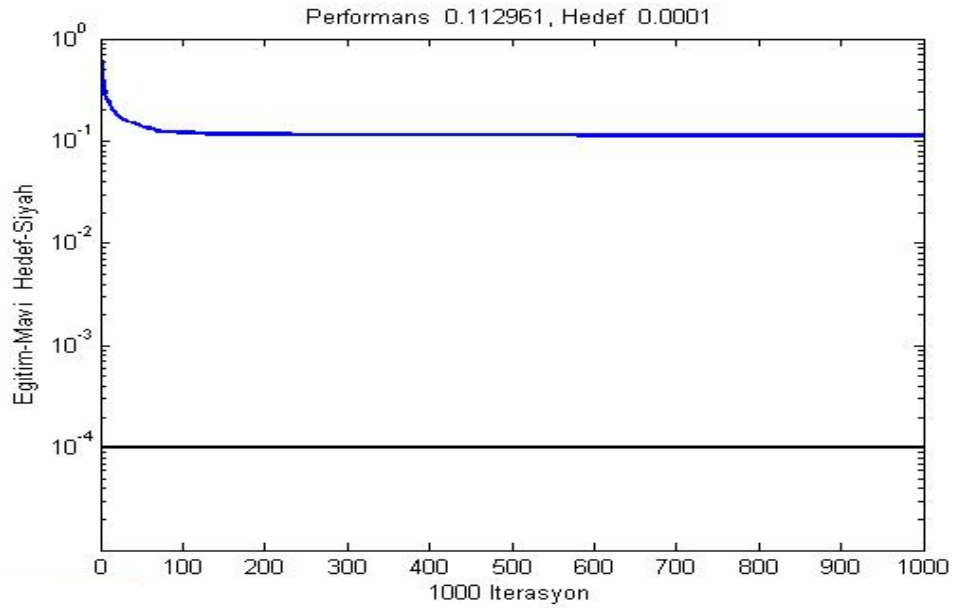
Mel Frekans K. Katsayısı	Vektör Nicemleme	Konuşmacı Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	16	10	0.02	412 sn	1,1 sn	0.12	45%
12	16	10	0.04	413 sn	0,9 sn	0,05	55%
12	16	10	0.06	430 sn	1,2 sn	0,07	65%
12	16	10	0.08	413 sn	1.2 sn	0.03	55%



Şekil 3.2 Kod vektör boyutu 16 ve 0.06 öğrenme oranı ile yapılan testte hata değişim eğrisi

Çizelge 3.2 Kod vektör boyutu 32 iken farklı öğrenme oranlarında test sonuçları

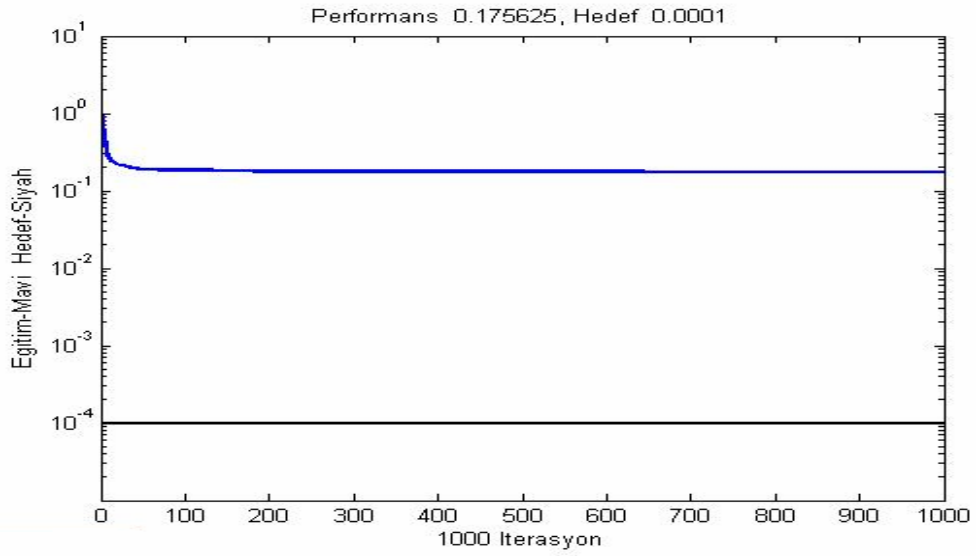
Mel Frekans K. Katsayısı	Vektör Nicemleme	Konuşmacı Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	32	10	0.02	753 sn	1,3 sn	0.14	55%
12	32	10	0.04	754 sn	1.7 sn	0,11	50%
12	32	10	0.06	766 sn	1,3 sn	0,11	55%
12	32	10	0.08	753 sn	1.3 sn	0,16	55%



Şekil 3.3 Kod vektör boyutu 32 ve 0.06 öğrenme oranı ile yapılan testte hata değişim eğrisi

Çizelge 3.3 Kod vektör boyutu 64 iken farklı öğrenme oranlarında test sonuçları

Mel Frekans K. Katsayısı	Vektör Nicemleme	Konuşmacı Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	10	0.02	1466 sn	4 sn	0,18	60%
12	64	10	0.04	1568 sn	2,5 sn	0,17	70%
12	64	10	0.06	1355 sn	3,0 sn	0,16	60%
12	64	10	0.08	1339 sn	3,1sn	0,21	70%



Şekil 3.4 Kod vektör boyutu 64 ve 0.04 öğrenme oranı ile yapılan testte hata değişim eğrisi

Yapılan testlerin sonuçları Çizelge 3.1, Çizelge 3.2 ve Çizelge 3.3’de gösterilmiştir. Test sonuçlarına bakıldığında ilk olarak artan kod vektör boyutu başarı oranını artırdığı gözlenmiştir. Bunun nedeni vektör nicemleme yapılırken özellik vektörlerinin boyutunda yaşanan değişiklikten dolayı oluşan kayıplardır. Kod vektör boyutunun küçülmesiyle ayırt edici özelliklerde oluşan kayıplar arttığından konuşmacı tanıma başarı oranı oldukça azalmıştır. Burada en ideal 64 kod vektör boyutunda konuşmacı tanıma testlerinin yapılmasıdır.

Her bir kod vektör boyutu ile yapılan testlere ait hata değişim eğrisi örnekleri Şekil 3.1, Şekil 3.2 ve Şekil 3.3’de gösterilmiştir.

Testte ikinci olarak yapay sinir ağlarında geri yayılım algoritmasının çalışma prensibinin bir parçası olan öğrenme oranı değişikliğinin konuşmacı tanıma başarıları incelenmiştir.

Konuşmacı tanıma başarı oranlarına bakıldığında öğrenme oranının değişimi her bir test için farklılık göstermiştir. Öğrenme oranı 0.02 olduğunda çok katmanlı algılayıcılarda geriye yayılım algoritması sırasında sistemin yerel minimum noktasına yakalanma ihtimali çok yüksektir. Geriye yayılım algoritması çalışması sırasında yerel minimuma yakalanırsa sistem hedef sonuca ulaşamaz ve beklenen sonuçlar elde edilemez. Başarı oranlarına bakıldığında diğer öğrenme oranlarına göre en düşük sonuçlar 0.002 değerinde çıkmıştır.

Geriye yayılım algoritması çalışırken yüksek öğrenme oranı kullanıldığında sistem büyük adımlarla değişim gösterir. Bu durum sistemin doğru sonucu bulmasında olumsuz etki oluşturur. Tanıma oranlarına bakıldığında da bu etki konuşmacı tanıma sisteminde de görülmektedir. Öğrenme oranı için testlere bakılarak genelleme yapılamaz. Bu genelleme ancak her durum için farklı öğrenme oranlarında testler yapılarak karar verilebilir.

Kod vektör boyutunun büyüklüğünün artması sistemin başarı performansını olumlu yönde etkilerken veri tabanını oluşturan vektörlerin boyutunun artması nedeniyle eğitim süresinin de artmasına neden olmuştur. Aynı şekilde test süresi de artış göstermiştir.

3.1.1.2 Yapay Sinir Ağlarında Saklı Katman Nöron Sayısının Başarıya Etkisi

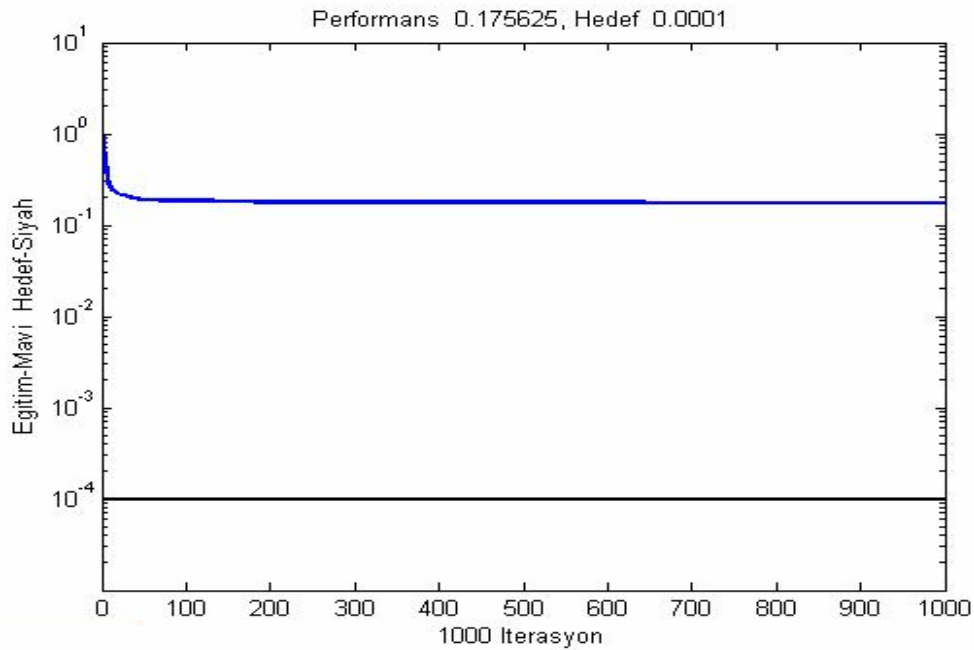
Yapay sinir ağlarında çok katmanlı algılayıcılar kullanılarak yapılan çalışmalarda çeşitli katman sayıları değiştirilerek en iyi sonuca ulaşmaya çalışılabilir.

Bu tez çalışmasında da hazırlanan konuşmacı tanıma sisteminde bulunan çok katmanlı algılayıcının saklı katmanında bulunan nöron sayısı değiştirilerek başarı oranına etkisi incelenmiştir.

Bu test 10 kişinin ses örnekleri kullanılmış, vektör nicemlemede kod kitabı boyutu 64 olarak alınmış, transfer fonksiyon hiperbolik tanjant sigmoid olarak seçilmiştir. Giriş katman sayısı MFCC'ye göre 12 olarak alınmış, çıkışta ise tanınacak kişi sayısı kadar nöron kullanılmıştır. Bu nöron sayıları sistemin yapısından zorunlu olarak geldiği için sadece saklı katmandaki nöron sayısı değiştirilerek test yapılmış ve sonuçları Çizelge 3.4, Çizelge 3.5, Çizelge 3.6 ve Çizelge 3.7'de gösterilmiştir.

Çizelge 3.4 Saklı katman nöron sayısı 15 iken farklı öğrenme oranlarında test sonuçları

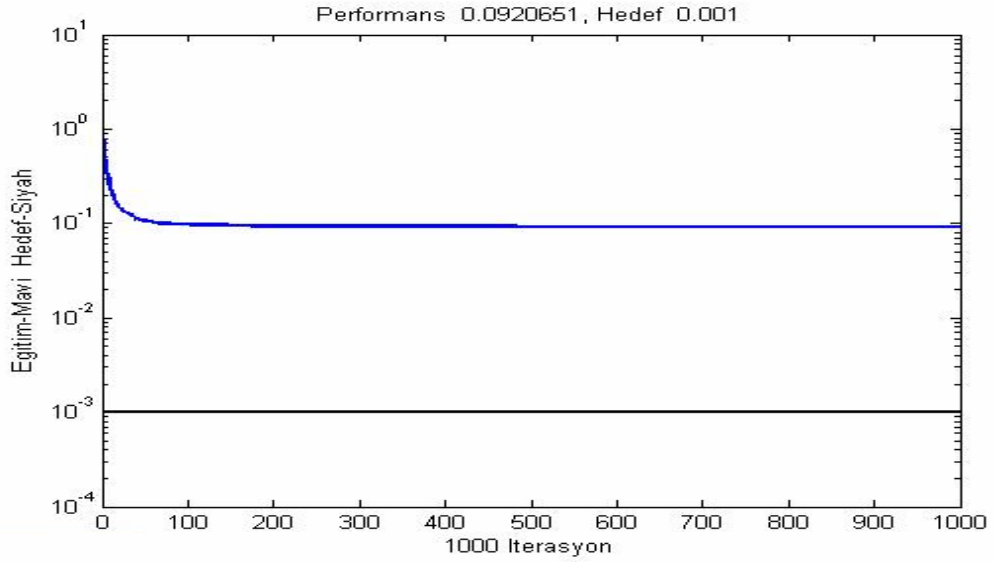
Mel Frekans K. Katsayısı	Vektör Nicemleme	Nöron Sayısı	YSA Ö. Oran	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	10	0.02	1466 sn	4 sn	0,18	60%
12	64	10	0.04	1568 sn	2,5 sn	0,17	70%
12	64	10	0.06	1355 sn	3,0 sn	0,16	60%
12	64	10	0.08	1339 sn	3,1sn	0,21	70%



Şekil 3.5 Kod vektör boyutu 64 ve 10 saklı katman ile yapılan testte hata değişim eğrisi

Çizelge 3.5 Saklı katman nöron sayısı 20 iken farklı öğrenme oranlarında test sonuçları

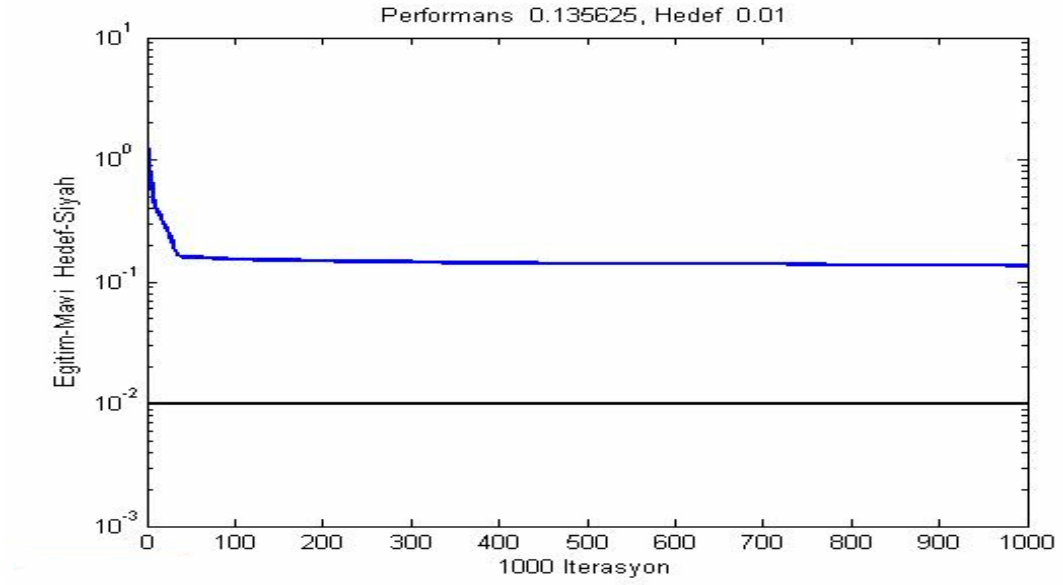
Mel Frekans K.Katsayısı	Vektör Nicemleme	Nöron Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	20	0.02	3320 sn	2,2 sn	0,10	70%
12	64	20	0.04	3218 sn	2,1sn	0,09	75%
12	64	20	0.06	3246 sn	3,4 sn	0,10	80%
12	64	20	0.08	6483 sn	9,6 sn	0,09	75%



Şekil 3.6 Kod vektör boyutu 64 ve 20 saklı katman ile yapılan testte hata değişim eğrisi

Çizelge 3.6 Saklı katman nöron sayısı 30 iken farklı öğrenme oranlarında test sonuçları

Mel Frekans K.Katsayısı	Vektör Nicemleme	Nöron Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	30	0.02	5417 sn	2.8 sn	0,06	75%
12	64	30	0.04	8476 sn	2,9 sn	0,13	75%
12	64	30	0.06	6173 sn	3,1 sn	0,04	80%
12	64	30	0.08	6403 sn	5,6 sn	0,04	80%

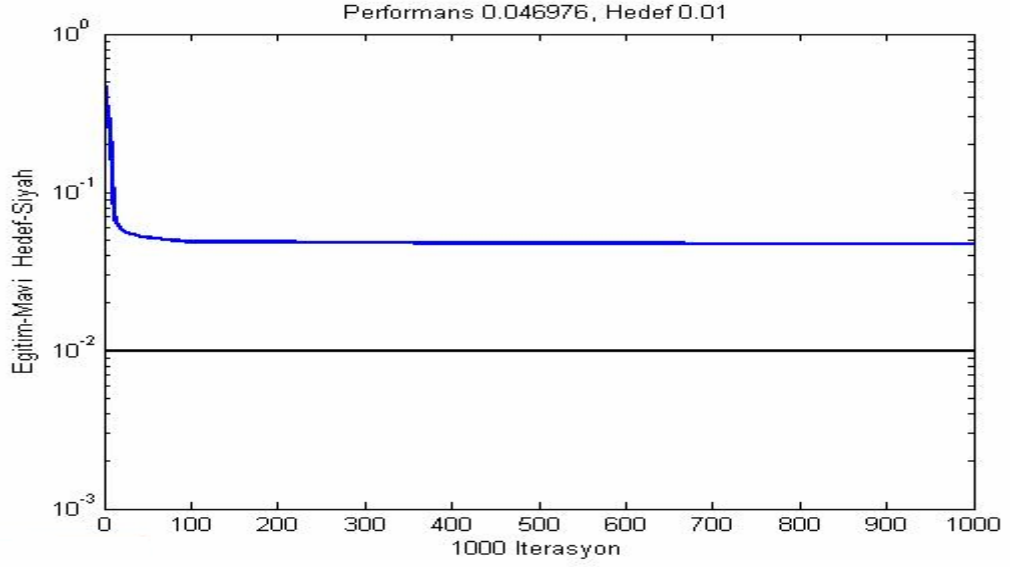


Şekil 3.7 Kod vektör boyutu 64 ve 30 saklı katman ile yapılan testte hata değişim eğrisi

Nöron sayısının artması sistemi daha karmaşık hale gelmekte bu durum eğitim ve test sürelerinin uzamasına olmasına yol açmaktadır. Burada yapılan testlerde de öğrenme oranı incelenmiştir. Farklı öğrenme oranlarında yüksek başarılar elde edilmiştir. Bu testte de görüldüğü gibi öğrenme oranının başarıya etkisi hakkında kesin bir genelleme yapılamamaktadır.

Çizelge 3.7 Saklı katman nöron sayısı 30 iken farklı öğrenme oranlarında test sonuçları

Mel Frekans K.Katsayısı	Vektör Nicemleme	Nöron Sayısı	YSA Ö. Oranı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	40	0.02	6547 sn	3.3 sn	0,03	85%
12	64	40	0.04	6816 sn	3,8 sn	0,04	85%
12	64	40	0.06	6173 sn	3,1 sn	0,04	90%
12	64	40	0.08	6403 sn	5,6 sn	0,04	85%



Şekil 3.8 Kod vektör boyutu 64 ve 40 saklı katman ile yapılan testte hata değişim eğrisi

Test sonuçları incelendiğinde katman sayısının artması başarı oranında yükselmeye neden olmuştur. Ayrıca artan katman sayısı Şekil 3.5, Şekil 3.5, Şekil 3.7 ve Şekil 3.8’de görüldüğü üzere sistemin minimum hata değerine yaklaşmasını sağlamıştır. Burada saklı katmandaki nöronlar sayesinde öğrenme aşamasında giriş ve çıkışlar arasındaki hata değeri düşmüştür ve sistem eğitimi başarılı olduğu için konuşmacı tanımaya yansımıştır.

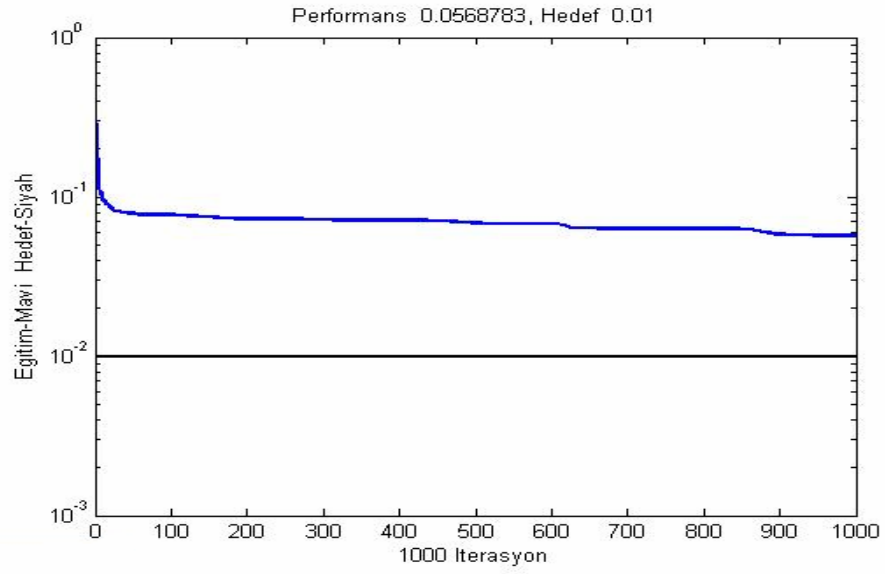
3.1.1.3 Yapay Sinir Ağlarında Transfer Fonksiyonun Başarıya Etkisi

Bundan önce yapılan testlerin hepsi hiperbolik tanjant sigmoid kullanılarak yapılmıştır. Bu transfer fonksiyonu eğitim sırasında çıkışı +1 ve -1 arasında değerler vermektedir. Sigmoid transfer fonksiyonunda ise çıkışlar 0 ve +1 arasında değişmektedir.

Bu test yapılırken öğrenme oranı 0.05 olarak sabit alınmıştır. Saklı katmandaki nöron sayısı değiştirilerek hiperbolik tanjant sigmoid transfer fonksiyonundan farklı test sonuçlarında gözlenmiştir. Yapılan testlere ait konuşmacı tanıma başarı sonuçları Çizelge 3.8, Çizelge 3.9 ve Çizelge 3.10’da gösterilmiştir.

Çizelge 3.8 Kod vektör boyutu 16 iken sigmoid fonksiyonun başarı test sonuçları

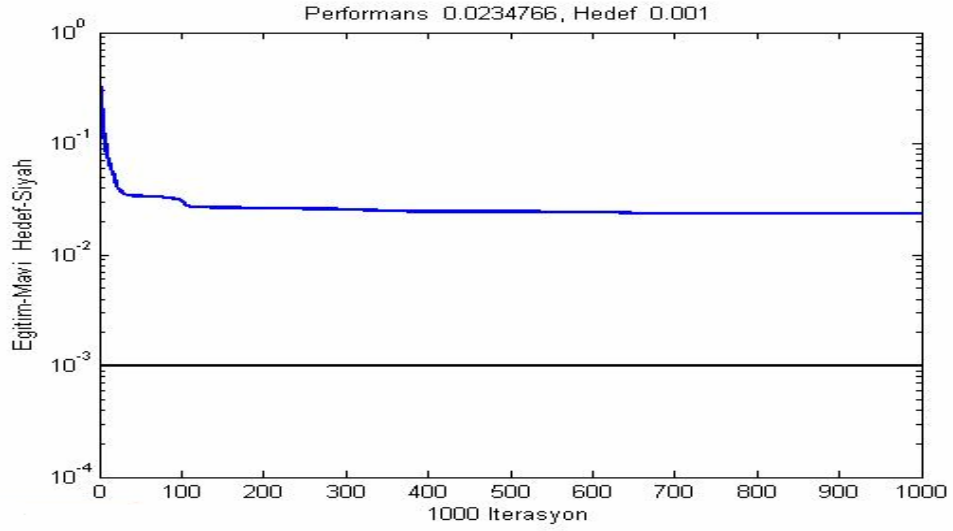
Mel Frekans Kepstrum Katsayısı	Vektör Nicemleme	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	16	10	415 sn	0,8 sn	0,07	20%
12	16	20	1027sn	0.8 sn	0,05	30%
12	16	30	3731 sn	1,9 sn	0,16	35%



Şekil 3.9 Kod vektör boyutu 16 ve sigmoid fonksiyon ile yapılan testte hata değişim eğrisi

Çizelge 3.9 Kod vektör boyutu 32 iken sigmoid fonksiyonun başarı test sonuçları

Mel Frekans Kepstrum Katsayısı	Vektör Nicemleme	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	32	10	1453 sn	2,9 sn	0,03	50%
12	32	20	1691sn	2,1 sn	0,01	60%
12	32	30	3028 sn	1,3 sn	0,02	60%

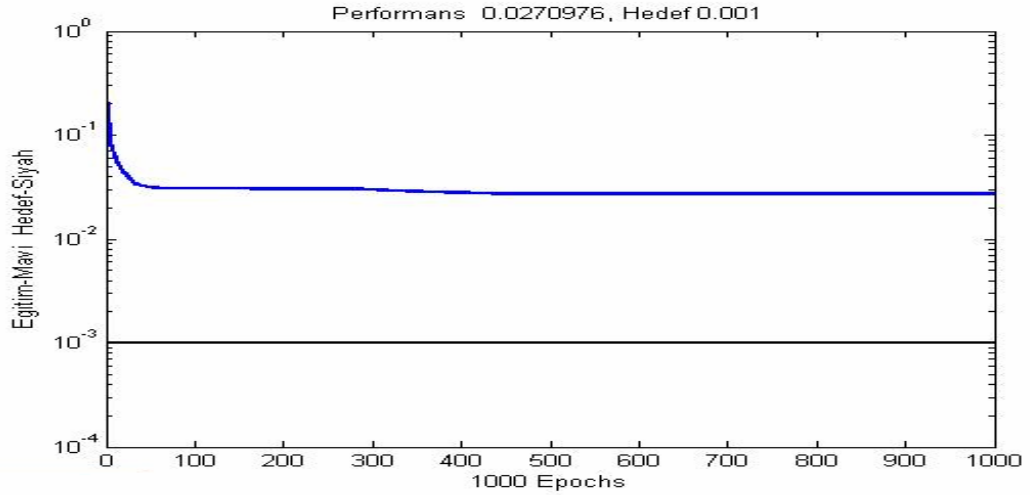


Şekil 3.10 Kod vektör boyutu 32, 30 saklı nöron sayısı ile yapılan testte hata değişim eğrisi

Çizelge 3.10 Kod vektör boyutu 64 iken sigmoid fonksiyonun başarı test sonuçları

Mel Frekans Kepstrum Katsayısı	Vektör Nicemleme	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	10	1453 sn	2,3 sn	0,04	70%
12	64	20	3240 sn	2,0 sn	0,02	75%
12	64	30	5349 sn	2,1 sn	0,01	70%

Test sonuçları kıyaslandığında sigmoid transfer fonksiyonun başarı oranı daha düşük olduğu ortaya çıkmıştır. Bunun nedeni çıkış değerlerinin 0 ile 1 aralığında olmasının hata olasılığını artırmasıdır. Şekil 3.9, Şekil 3.10 ve Şekil 3.11’de farklı kod vektör boyutları için sigmoid transfer fonksiyon ile yapılan testlere ait hata değişim eğrileri gösterilmektedir.



Şekil 3.11 Kod vektör boyutu 64, 20 saklı nöron sayısı ile yapılan testte hata değişim eğrisi

3.1.1.4 Konuşmacı Sayısının Başarıya Etkisi

Konuşmacı tanıma çalışması yapılırken en önemli adımlardan biri tanıma işleminin yapılacağı kişi sayısıdır. Yapay sinir ağlarında kişi sayısı önemli rol oynamaktadır. Geriye yayılım algoritmasının çalışma yapısından kaynaklanan bellek problemi nedeniyle kişi sayısı arttığında MATLAB yazılı belleğin yetmediği yönünde uyarı vermektedir. Bu nedenle testler yapılırken kişi sayısı kademeli olarak artırılmıştır. Otuz kişi sayısından sonra sistem hata uyarısı vermeye başladığı için yapay sinir daha fazla konuşmacı için eğitilememiştir.

Konuşmacı sayısı için yapılan testlerde en iyi performansa sahip kod vektör boyutu 64 kullanılmıştır. Öğrenme oranı ise sabit tüm testlerde 0.05 olarak alınmıştır. Saklı katmandaki transfer fonksiyon ise hiperbolik tanjant sigmoid kullanılmıştır.

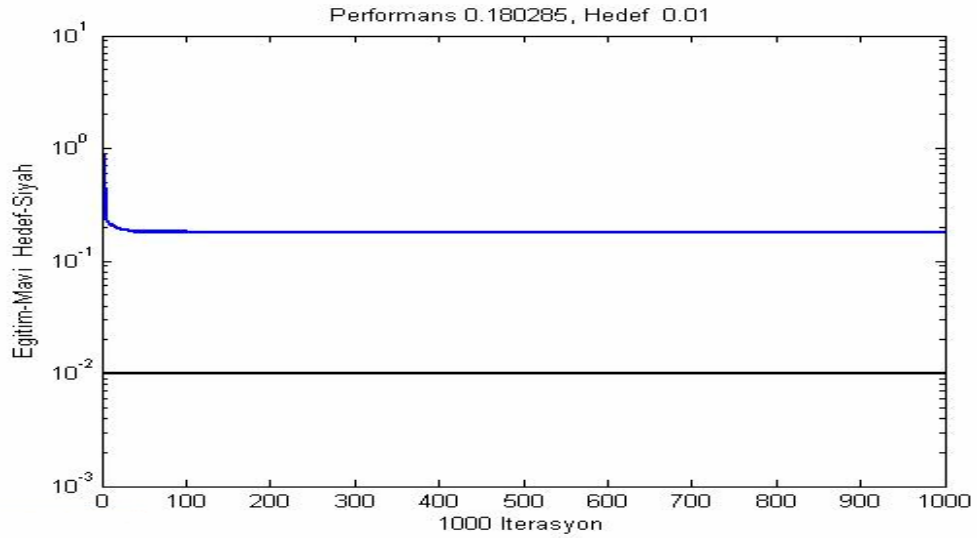
Çizelge 3.11 On konuşmacı ile yapılan testlerin sonuçları

Mel Frekans	Konuşmacı	Nöron	Eğitim	Test	Hata	Başarı
Kepstrum Katsayısı	Sayısı	Sayısı	Süresi	Süresi	Değ.	Oranı
12	10	10	1568 sn	2,5 sn	0,17	70%
12	10	20	3218sn	2,1 sn	0,09	75%
12	10	30	8476sn	2,9 sn	0,13	75%
12	10	40	8816 sn	3.8 sn	0.04	85%
12	10	50	8903sn	4.9 sn	0.04	75%

Çizelge 3.12 Onbeş konuşmacı ile yapılan testlerin sonuçları

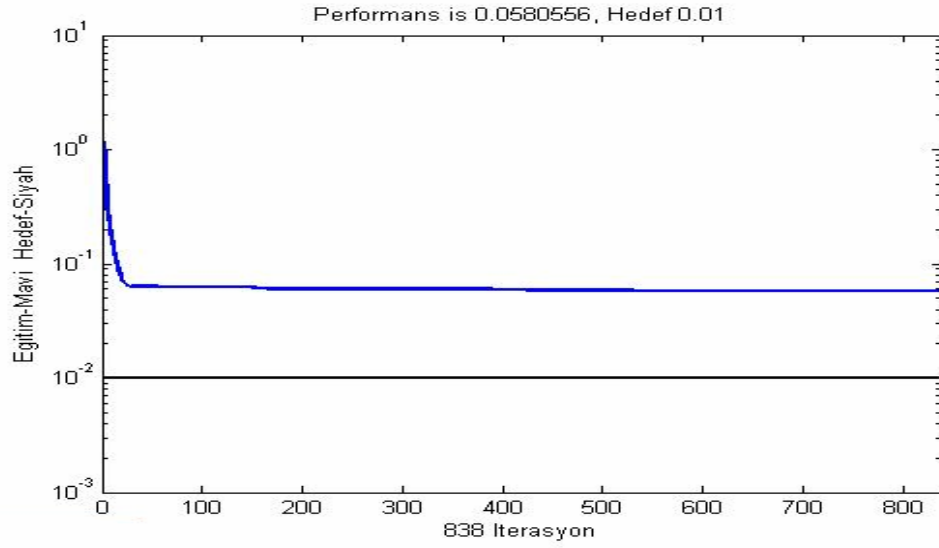
Mel Frekans Kepstrum Katsayısı	Konuşmacı Sayısı	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	15	10	6411 sn	6,2 sn	0,18	53,3%
12	15	20	9675sn	6,6 sn	0,09	63,3%
12	15	30	16961 sn	8,1 sn	0,09	63,3%
12	15	40	20972 sn	8,7 sn	0,07	53,3%
12	15	50	36213sn	8,9 sn	0,05	50%

Kişi sayısı artması ile sistemin başarı performansının düşmeye başladığı Çizelge 3.11, Çizelge 3.12, Çizelge 3.13 ve Çizelge 3.14' de bulunan test sonuçlarında görülmektedir.



Şekil 3.12 Kod vektör boyutu 64, 15 kişi ve 10 saklı nöron ile yapılan testte hata değişim eğrisi

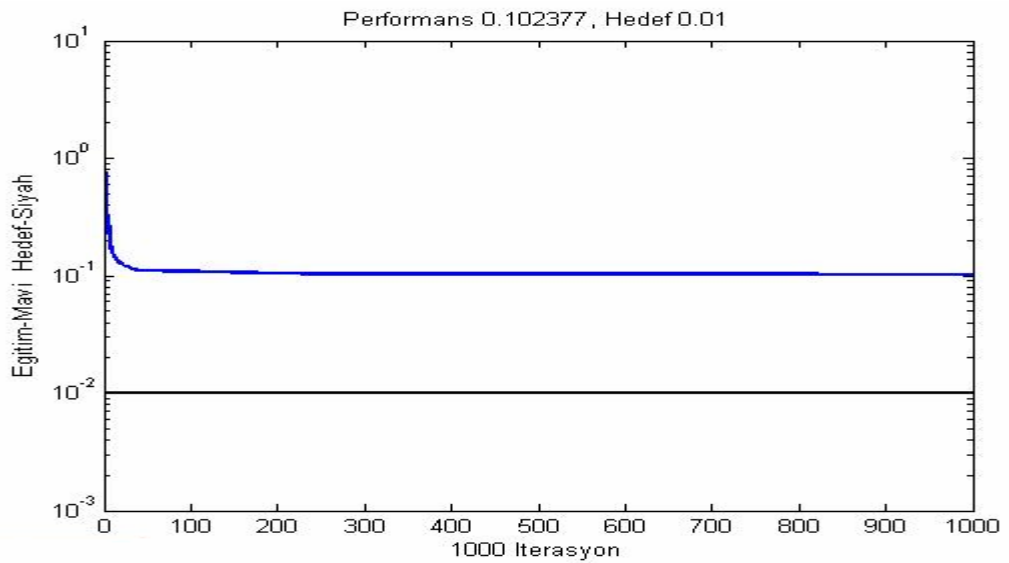
Yapay sinir ağlarıyla yapılan konuşmacı tanıma sisteminde kişi sayısının artmasıyla birlikte çıkış katmanındaki nöron sayısı da artmaktadır. Test sonuçlarına bakıldığında az sayıda konuşmacıyla yapılan testlerin başarı oranı yüksekken kişi sayısının artmasıyla konuşmacı tanıma oranı düşmektedir.



Şekil 3.13 Kod vektör boyutu 64, 15 kişi ve 50 saklı nöron ile yapılan testte hata değişim eğrisi

Çizelge 3.13 Yirmi konuşmacı ile yapılan testlerin sonuçları

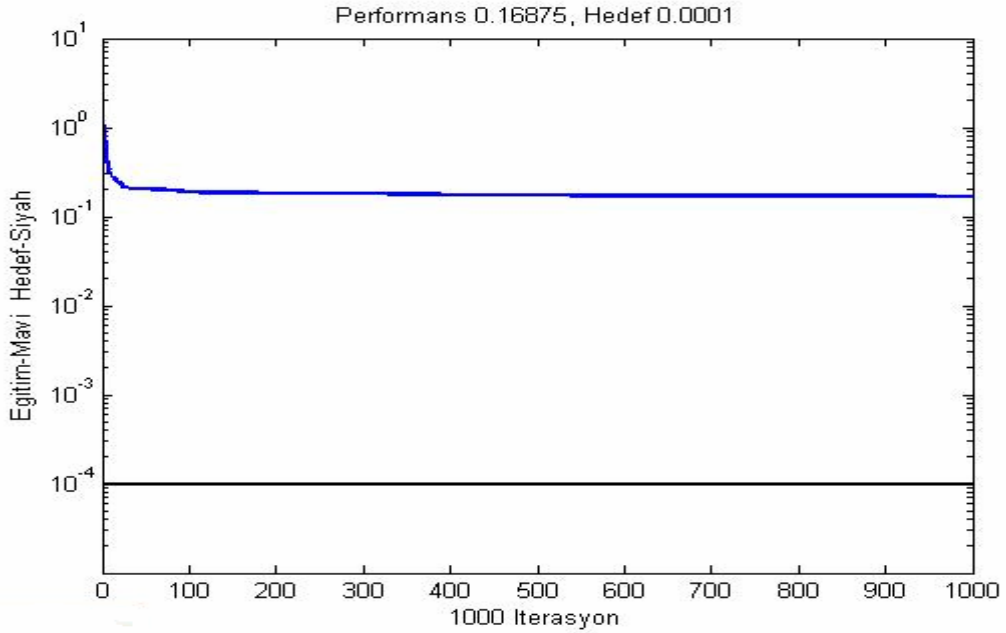
Mel Frekans Kepstrum Katsayısı	Konuşmacı Sayısı	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	20	10	8491 sn	7,2 sn	0,16	40 %
12	20	20	20743 sn	11,5sn	0,10	45%
12	20	30	24181 sn	12,0 sn	0,11	50 %
12	20	40	29372 sn	12,6 sn	0,11	47,5 %
12	20	50	36176 sn	13,2 sn	0,12	40%



Şekil 3.14 Kod vektör boyutu 64, 20 kişi ve 20 saklı nöron ile yapılan testte hata değişim eğrisi

Çizelge 3. 14 Otuz konuşmacı ile yapılan testlerin sonuçları

Mel Frekans Kepstrum Katsayısı	Konuşmacı Sayısı	Nöron Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	30	10	9834 sn	7,9 sn	0,18	36.6%
12	30	20	11046 sn	12,6sn	0,17	40 %
12	30	30	20034 sn	11,2 sn	0,16	43,3%
12	30	40	30170 sn	14,7 sn	0,15	43,3%
12	30	50	40341 sn	16,1 sn	0,16	40 %



Şekil 3.15 Kod vektör boyutu 64, 30 kişi ve 30 saklı nöron ile yapılan testte hata değişim eğrisi

Kişi sayısının artması ile yapay sinir ağ yapısında kullanılacak eğitim ve test grubunun vektör boyutları artmaktadır. Bu tanıma sisteminin başarı oranını düşürmekte buda sistemin performansının konuşmacı sayısı arttıkça daha kötüye gittiğini göstermektedir. Ayrıca Şekil 3.12, Şekil 3.13, Şekil 3.14 ve Şekil 3.15’de hata değişim eğrileri incelendiğinde kişi sayısı arttıkça sistemin hatası arttığı gözlenmiştir. Kişi sayısı ve saklı katman sayısı artmasının sisteme en büyük etkisi eğitim ve test süresinin büyük ölçüde artması olmuştur.

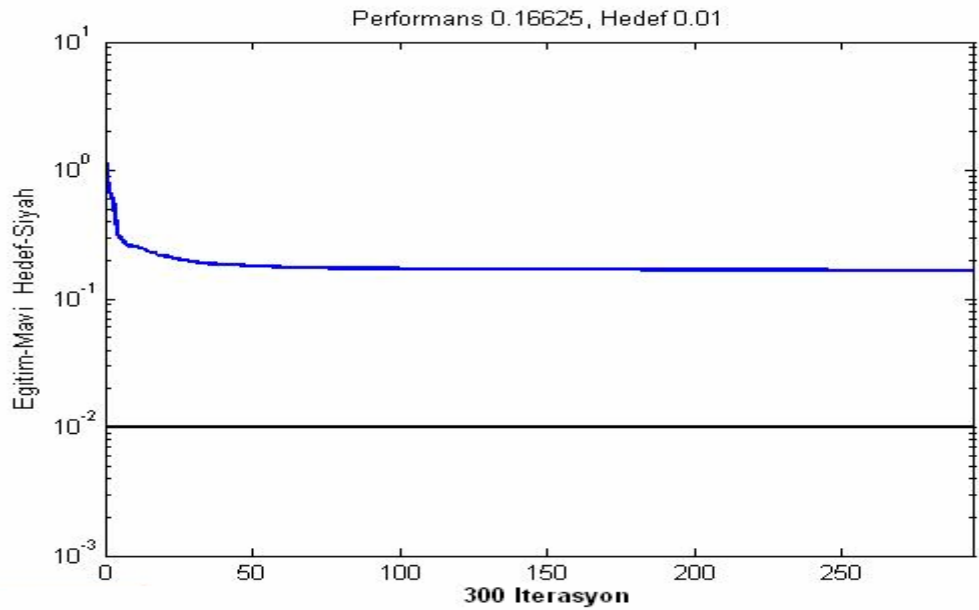
3.1.1.5 İterasyon Sayısının Başarıya Etkisi

Konuşmacı tanımada yapay sinir ağının eğitilmesi önemli bir aşamadır. Bu eğitim süresini belirleyen 2 unsur vardır. Bunlarda biri hedeflenen hata değeri diğeri ise iterasyon sayısıdır. Sistem hedeflenen hata değerine ulaşırsa veya istenen iterasyon sayısına ulaşırsa eğitimi durdurur.

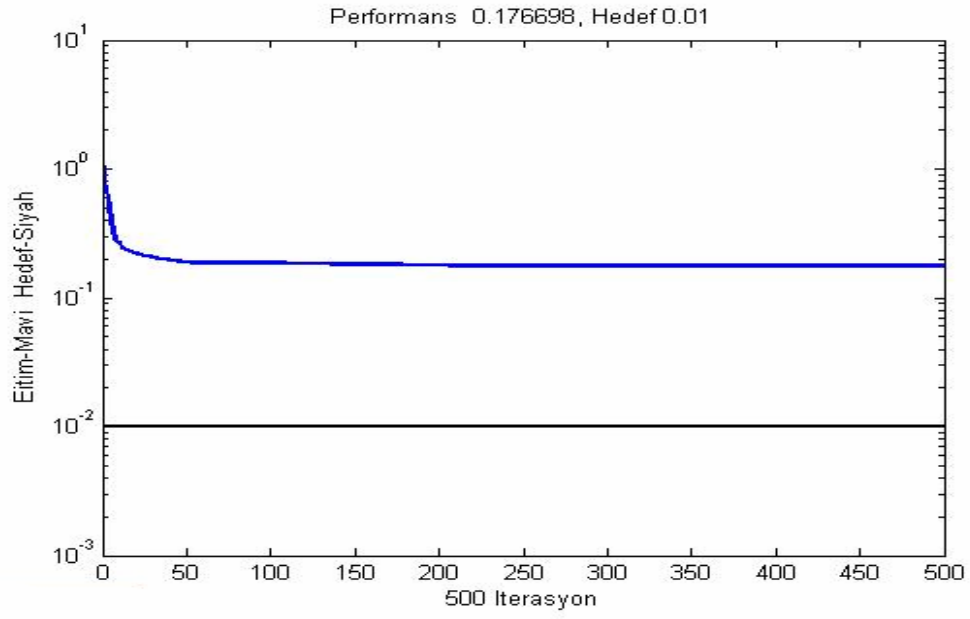
Kod vektörü 64, konuşmacı sayısı 10 ve öğrenme oranı 0,05 seçilerek iterasyon sayısı değiştirilmiş ve başarı sonuçları aşağıda Çizelge 3.15’de verilmiştir.

Çizelge 3.15 Farklı iterasyon sonuçları ile yapılan testlerin sonuçları

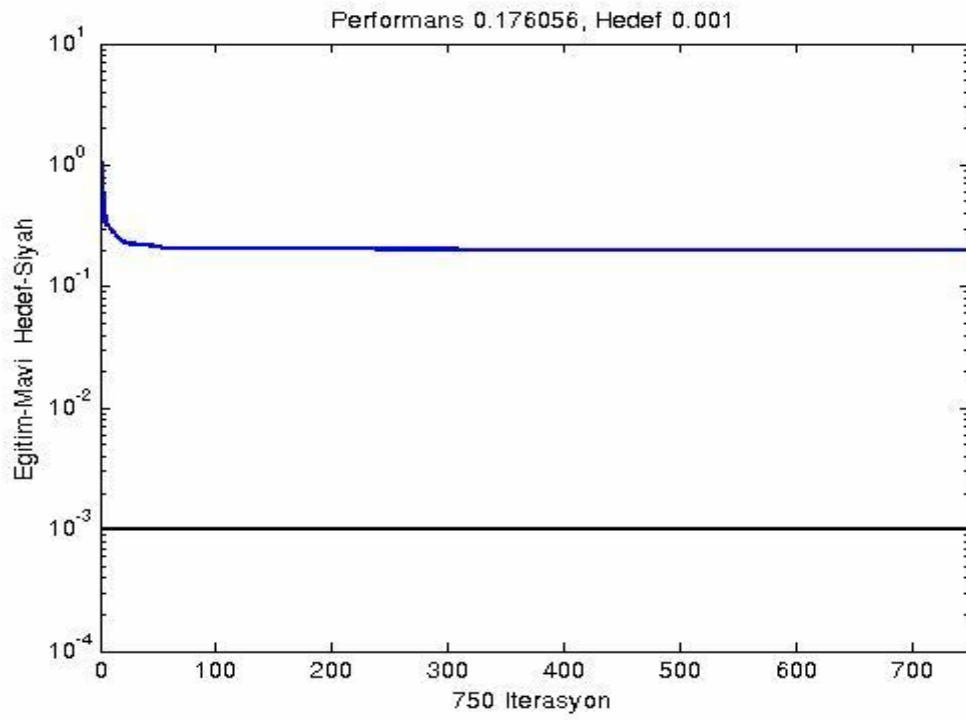
Mel Frekans Kepstrum Katsayısı	Vektör Nicemleme	İterasyon Sayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	64	300	430 sn	2,4 sn	0,16	65%
12	64	500	724 sn	2,21 sn	0,17	70%
12	64	750	1288 sn	2,3 sn	0,17	70%
12	64	1000	1568 sn	2,5 sn	0,16	70%



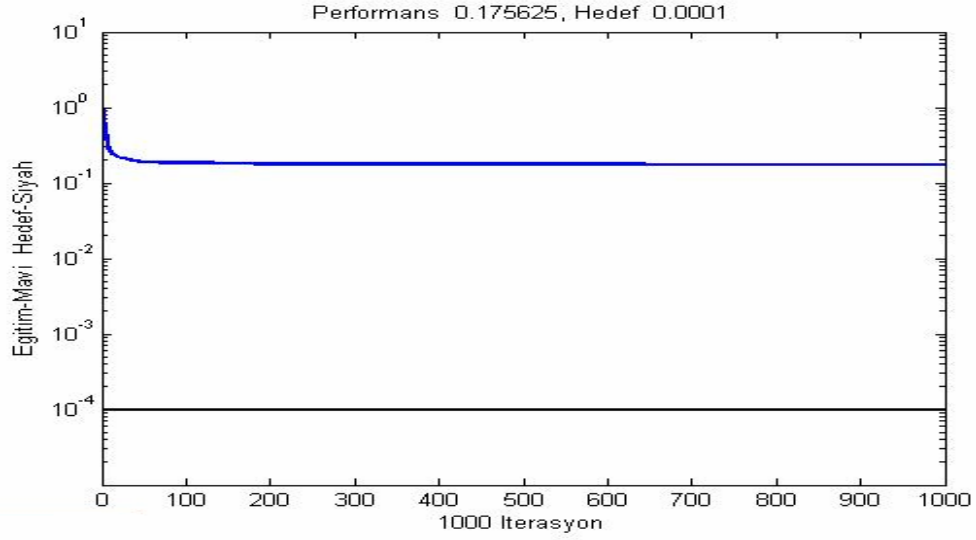
Şekil 3.16 İterasyon sayısı 300 iken yapılan testte hata değişim eğrisi



Şekil 3.17 İterasyon sayısı 500 iken yapılan testte hata değişim eğrisi



Şekil 3.18 İterasyon sayısı 750 iken yapılan testte hata değişim eğrisi



Şekil 3.19 İterasyon sayısı 1000 iken yapılan testte hata değişim eğrisi

Yukarıda Şekil 3.16, Şekil 3.17, Şekil 3.18 ve Şekil 3.19’da sırası ile 350, 500, 750 ve 1000 iterasyon sayısı kullanılarak yapılan testlerin hata değişim eğrileri görülmektedir. Bu testlerde iterasyon sayısı dışında diğer test parametreleri değiştirilmemiştir. Bu şekilde farklı iterasyon sayısının başarıya olan etkisi incelenmiştir.

Farklı iterasyon sayıları ile yapılan testlerin başarı sonuçları incelendiğinde artan iterasyon sayısının başarı ile doğrusal olarak değişmediği görülmektedir. 300 iterasyonla yapılan testte hata oranı diğer testlere göre daha fazla çıktığı ve başarının daha düşük olduğu sonucu çıkmaktadır. İterasyon sayısı artması ile hata oranı düşmekte ve başarı oranı yükselmektedir. Fakat 750 ve 1000 iterasyon ile yapılan testler incelendiğinde hatanın ve başarı oranının değerinin birbirine çok yakın olduğu gözlenmektedir. Buradan anlaşılabacağı üzere belirli bir iterasyon sayısından sonra sistemin hata değeri çok küçük oranda değişmekte ve bu değişim sistemin performansını etkilememektedir. Bu nedenle hatanın değişimi kontrol edilerek sistemin en iyi sonuç vereceği iterasyon sayısına karar verilebilir. Bu tezde yapılan testlerde hata değeri 1000 iterasyondan sonra çok küçük değişimler gösterdiği için daha fazla iterasyon kullanılmamıştır.

3.1.1.6 Farklı Eğitim Algoritmalarının Başarıya Etkisi

Daha önce yapılan testlerde hızlı eğitim algoritması olarak da bilinen Levenberg-Marquardt eğitim algoritması kullanılmıştır. Eğitim algoritması yapay sinir ağları uygulamalarında önemli yer tutmaktadır. Yapılacak çalışmanın özellikleri incelenerek ve testler yapılarak en iyi performans sağlayan eğitim algoritmasına ulaşılabilir.

Bu çalışmada 2 farklı eğitim algoritması daha uygulanmıştır. Bunlar gradyant azalan algoritma ve momentumlu gradyant azalan algortimadır. Momentum geri yayılım algoritmasında ağırlıkların bir önceki ağırlık değişiminin belirli bir oranında olmasını sağlar. Böylece sistemin yerel minimumlara takılma olasılığı azalır.

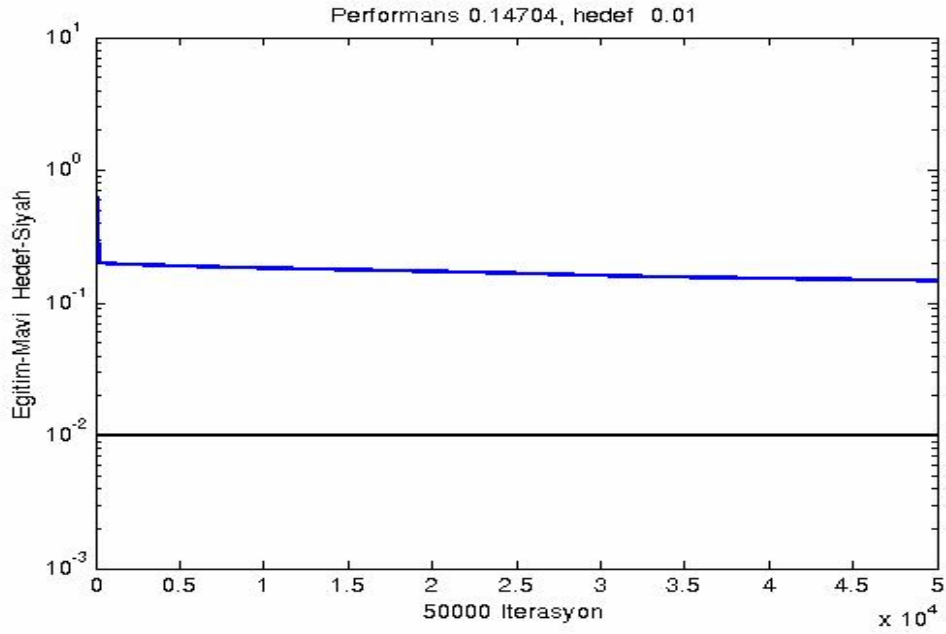
Bu testler 20 ve 30 kişiden alınan ses örneklerinin 64 kod vektör boyutu kullanılarak ve öğrenme oranı 0,06 seçilerek yapılmış ve saklı katman sayısı değiştirilerek başarıya etkisi incelenmiştir.

Çizelge 3.16 Gradyant azalan algoritma kullanılarak 20 kişi ile yapılan test sonuçları

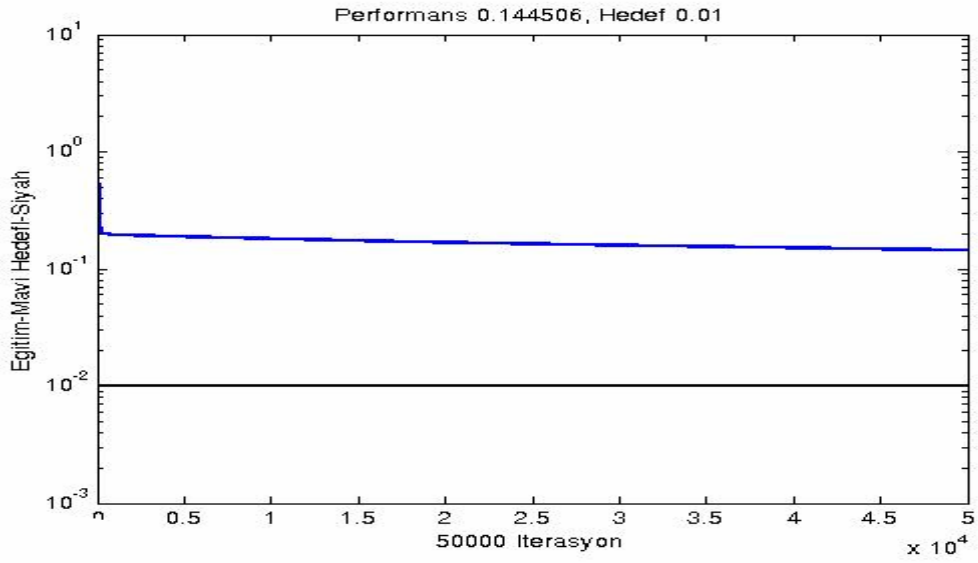
Mel Frekans	Kişi	Saklı Katman	Eğitim	Test	Hata	Başarı
Kepstrum Katsayısı	Sayısı	Sayısı	Süresi	Süresi	Değ.	Oranı
12	20	20	1763 sn	5,7 sn	0,17	35 %
12	20	30	2503 sn	5,5 sn	0,14	45 %
12	20	40	2451 sn	5,4 sn	0,15	45 %
12	20	50	2802 sn	6,5 sn	0,14	47,5 %

Çizelge 3.16’da yirmi kişi, Çizelge 3.17 ise otuz kişi ile yapılan testlerde gradyant azalan algoritma ile eğitilen sistemin konuşmacı tanıma başarı sonuçları gösterilmiştir.

Şekil 3.20 ve Şekil 3.21’de yirmi kişi, Şekil 3.22 ve Şekil 3.23 ise otuz kişi ile yapılan testlerde saklı katmandaki nöron sayısı değiştiğinde sistemin hata değişim eğrisinin değişimi gösterilmiştir.



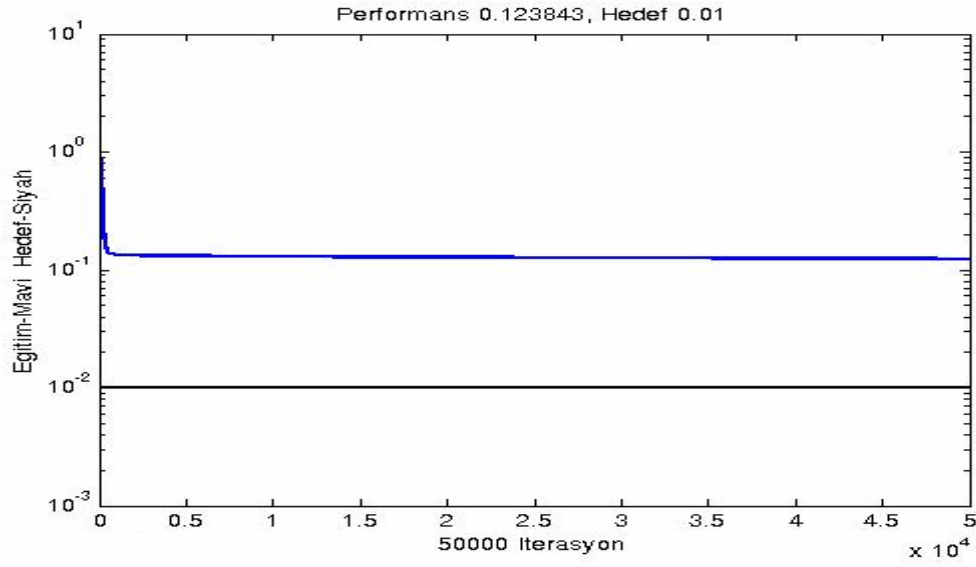
Şekil 3.20 Gradyant azalan algoritma ve 30 saklı nöron ile yapılan teste ait hata değişim eğrisi



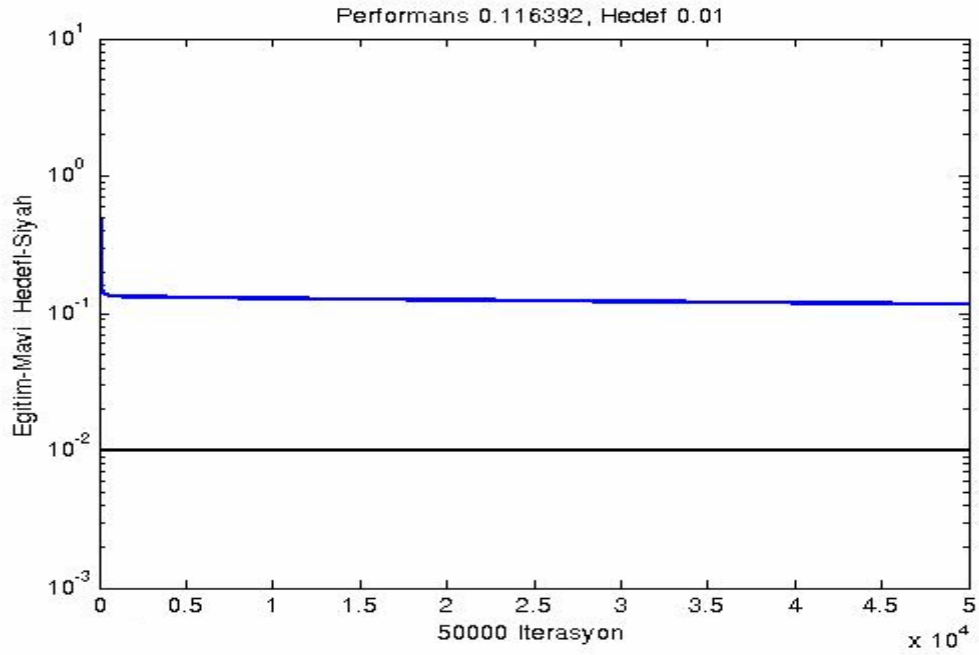
Şekil 3.21 Gradyant azalan algoritma ve 50 saklı nöron ile yapılan teste ait hata değişim eğrisi

Çizelge 3.17 Gradyant azalan algoritma kullanılarak 30 kişi ile yapılan test sonuçları

Mel Frekans	Kişi	Saklı Katman	Eğitim	Test	Hata	Başarı
Kepstrum Katsayısı	Sayısı	Sayısı	Süresi	Süresi	Değ.	Oranı
12	30	20	3377 sn	8,7 sn	0,127	25 %
12	30	30	4183 sn	7,8 sn	0,123	30 %
12	30	40	4896 sn	9,0 sn	0,119	33 %
12	30	50	5736 sn	9,2 sn	0,116	40 %



Şekil 3.22 Gradyant azalan algoritma, 30 saklı katman ile yapılan teste ait hata değişim eğrisi



Şekil 3.23 Gradyant azalan algoritma, 50 saklı nöron ile yapılan teste ait hata değişim eğrisi

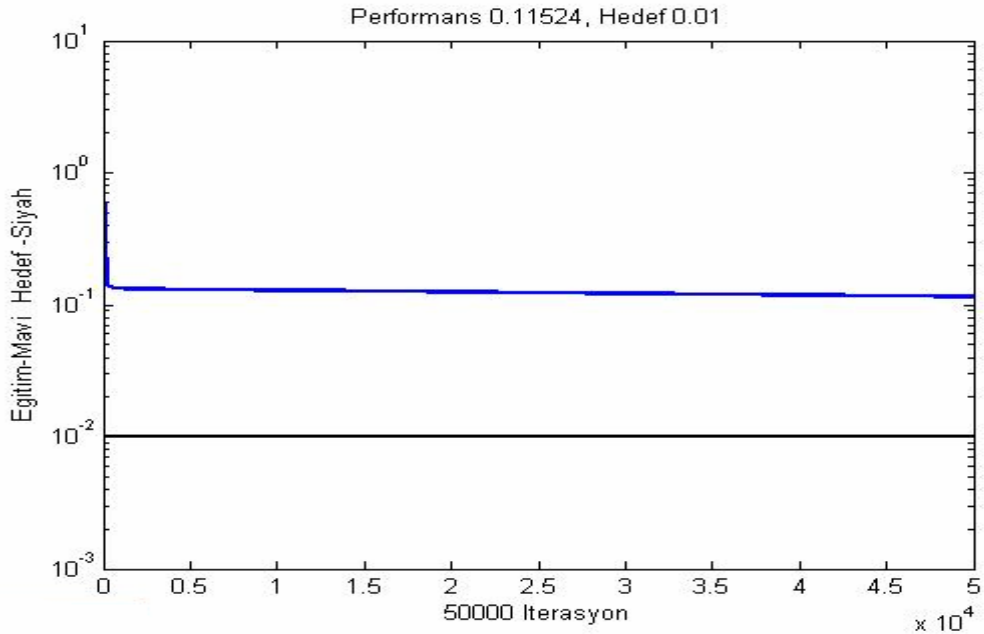
Test sonuçları kıyaslandığında gradyant azalan algoritmanın eğitim ve test sürelerinin daha düşük olduğu dikkat çekmektedir. Bunun yanında bu öğrenme algoritması Levenberg-Marquardt öğrenme algoritmasına üstünlük sağlayamadığı görülmektedir. Konuşmacı tanımada ki başarı oranları yaklaşık sonuçlar vermiştir.

Gradyant azalan algoritma kullanılırken ağırlık öğrenme esnasındaki salınımları önlemek amacıyla momentum katsayısı kullanılır. Momentum katsayısı 0 ile 1 arasında seçilir. Momentum kullanıldığında, ağırlıklara göre eğim ardışık iki iterasyonda aynı işaretli ise ağırlıklara uygulanacak düzeltme artar, aksi takdirde azalır.

Momentum katsayısı, her iterasyonda hesaplanan ağırlık hatasının türevi ile orantılıdır ve uygun değerleri geri yayılım algoritmasının doğru olan minimuma yaklaşmasını sağlar. Momentum katsayısı ile YSA'nın öğrenme oranında belirli bir hızlanma elde edilir.

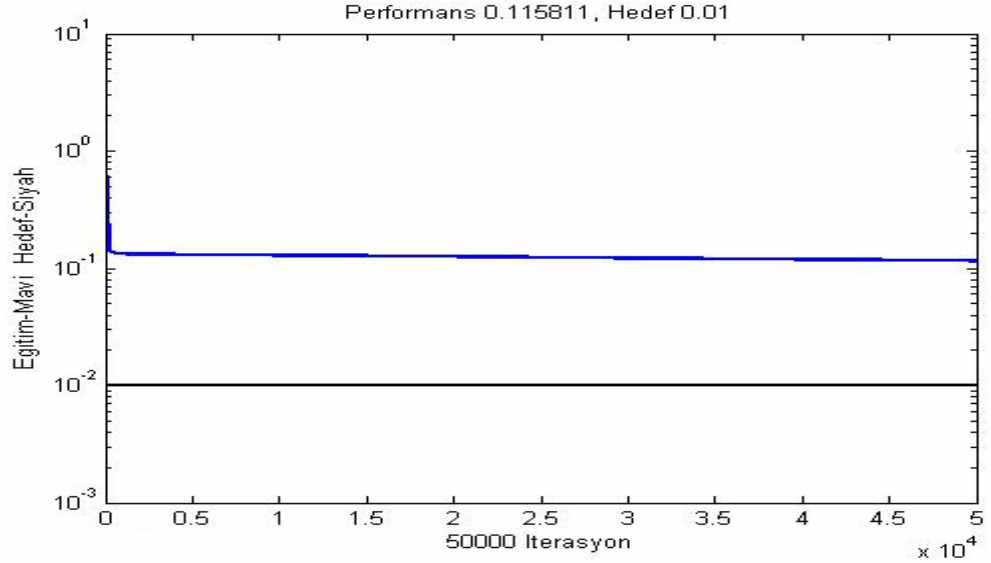
Çizelge 3.18 Momentumlu gradyant azalan algoritma kullanılarak 30 kişi ile yapılan test sonuçları

Mel Frekans Kepstrum Katsayısı	Kişi Sayısı	Momentun Katsayısı	Eğitim Süresi	Test Süresi	Hata Değ.	Başarı Oranı
12	30	0,9	5929 sn	16,9 sn	0,115	53,3 %
12	30	0,7	5907 sn	17,1 sn	0,116	48,3 %
12	30	0,5	6789 sn	16,8 sn	0,115	46,6 %
12	30	0,3	8479 sn	18,4 sn	0,115	46,6 %



Şekil 3.24 Momentum katsayısı 0,9, kişi sayısı 30 iken yapılan teste ait hata değişim eğrisi

Çizelge 3.18’de otuz kişi ile yapılan testleri aynı parametreler kullanarak birde eğitim algoritması olarak momentumlu gradyant azalan algoritma kullanılmış ve momentum katsayısı sisteme eklenerek tekrarlanmış ve başarıya olan etkisi incelenmiştir. Farklı momentum katsayıları ile yapılan testlere ait hata değişim eğrileri Şekil 3.24 ve Şekil 3.25’de gösterilmiştir.



Şekil 3.25 Momentum katsayısı 0,7, kişi sayısı 30iken yapılan teste ait hata değişim eğrisi

Momentum kullanılarak yapılan testler eğitim ve test süresinde artışa neden olsa da sistemin performansını olumlu etkilemiş ve tanıma oranını yükseltmiştir. Bu testlerde momentum katsayısının farklı değerleri için alınan başarı sonuçlarına bakıldığında momentum katsayısı için genelleme yapılamamaktadır.

3.1.2 Test Sonuçlarının Yorumlanması ve Öneriler

Yapılan tüm testlere genel bakıldığında yapay sinir ağlarıyla kurulan sistemin ne kadar karmaşık bir yapıya sahip olduğu görülebilir. Bu tez çalışmasında için Matlab Toolbox kullanılarak testler yapılmıştır. Bunun nedeni sistemin parametrelerinin değiştirilmesi sırasında kolaylık sağlanmasıdır. Çünkü yapay sinir ağlarında katman sayısı ve konuşmacı sayısı değişimi oluşturulacak ağ yapısını doğrudan etkilemektedir. Bu karmaşık yapının sahip olduğu birçok parametre konuşmacı tanımadaki başarıyı olumlu veya olumsuz yönde etkilemektedir.

Test sonuçlarına genel bakıldığında ilk olarak göze çarpan vektör nicemleme işleminin tanımadaki büyük etkisidir. Eğer vektör niceme 16 ve 32 kod vektör boyutlarında yapılırsa sistemin başarı performansı düşük olmaktadır. Bu durumda en ideal 64 kod vektör boyutudur. Bu boyut artırmak yapılacak testlerde kullanılan kişi sayısını düşürdüğü için daha fazla artırılmamıştır.

Yapay sinir ağlarındaki parametrelere bakıldığında ise öğrenme oranının sistemin başarısını etkilediği görülmektedir. Öğrenme oranını her bir test için deneme yanılma yöntemiyle ideal değeri bulunabilir. Yapay sinir ağlarına genel bakıldığında öğrenme oranının çok küçük ve çok büyük değerlerin kullanılmaması tavsiye edilebilir. Ayrıca katman sayısının artması sistemin eğitimini olumlu yönde etkilediği ve konuşmacı tanıma başarısını arttırdığı gözlenmektedir. Fakat katman sayısının artışı ile başarı oranının değişimi lineer değildir. Belirli bir katman sayısından sonra sistemin başarı oranı artmamakta buna rağmen eğitim ve test süresi çok uzamaktadır. Buda gerçekleştirilebilir sistemler için dezavantaj oluşturmaktadır.

Levenberg-Marquardt öğrenme algoritması ile iterasyon sayısı değiştirilerek yapılan testlerde iterasyon sayısının artışın olumlu etkisi gözlenmiştir. Bu algoritma hızlı eğitim özelliğine sahip olması nedeniyle iterasyon sayısının artışı belirli bir rakamdan sonra başarıyı çok fazla etkilememiştir. Bunun nedeni artan iterasyon sayısı ile hata değişiminin yavaşlaması nedeniyle sonucu etkilememesidir.

Farklı öğrenme algoritmaları ile yapılan testlerde momentumlu gradyant azalan algoritma ile 30 kişi ile yapılan testlerde diğer algoritmalara oranla daha iyi sonuçlar elde edilmiştir. Momentum katsayısı, her iterasyonda hesaplanan ağırlık hatasının türevi ile orantılıdır ve uygun değerleri geri yayılım algoritmasının doğru olan minimuma yaklaşmasını sağlar. Öğrenme ve momentum katsayılarının uygun değerleri ile çözüm için hızlı ve kararlı yaklaşımların elde edilmesi mümkün olmaktadır. İleride yapılacak çalışmalarda farklı öğrenme ve momentum katsayıları kombinasyonu yapılarak en ideal değerler bulunabilir.

Bu tezde yapılan testlere bakılarak hazırlanan sistemin konuşmacı sayısının artmasından kötü etkilendiği gözlenmektedir. Bu nedenle çok sayıda konuşmacının tanınması yapılacak çalışmalarda yapay sinir ağ yapısının kullanıldığında başarı oranının düşük olacağı söylenebilir.

Bu konuda ileride yapılacak çalışmalarda farklı bir yazılım kullanılarak performans artırılabilir. Ayrıca kişilerin ses özellik vektörlerinin boyutlarının değiştirilmesi sırasında oluşan kayıpların olumsuz etkisini azaltabilecek çalışmaların yapay sinir ağlarıyla yapılacak konuşmacı tanıma işleminde başarıyı yükselterek daha geniş konuşmacı setleri kullanılmasına imkân verebilir.

KAYNAKLAR

- Ahad, A., Fayyaz, A., Mehmood T. 2002. Speech Recognition using Multilayer Perception. Students Conference Proceedings, IEEE Volume 1 p.103-109
- Bennani, Y., Gallnari, P. 1995. Neural Networks for discrimination and modernization of speakers. Speech Communication, vol.17, p.159–175
- Caner, M. ve Üstün, S.V. 2006. Yapay Sinir Ağları İle Konuşmacı Kimliği Tanıma Uygulaması. Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi, Sayı 2, Cilt 12, s.279–284
- Che, C., Lin, Q. 1995. Speaker recognition using HMM with experiments on the YOHO database. Eurospeech, Madrid, ESCA, p.625-628
- Davis, S.B., Mermelstein, P., 1980 Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentence, IEEE Trans. on Acoustic, Speech and Signal Processing, p. 357–366
- Doddington, G. R. 1985. Speaker recognition –Identifying People by their Voices. IEEE ,Vol.73, No.11, p.1651-1664
- Duda, R. O., Hart, P. E. ve Stork, D.G. 2001. Pattern Classification 2nd Edition. Wiley-Interscience Publication, USA.
- Furui, S. 1994. An Overview of Speaker Recognition Technology. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, ESCA 94, p.1-9
- Haykin, S. 1999. Neural Networks. A Comprehensive Foundation, Upper Saddle River, N.J., Prentice Hall,
- Hermansky, H. 1990. Perceptual Linear Predictive (PLP) Analysis of Speech. Journal Of Acoustic Society of America. vol. 87, no. 4. p. 1738–1752
- Hermansky, H., Morgan, N. 1994. RASTA processing of speech. IEEE Transactions on Speech and Audio Processing, vol. 2, no. 4, p. 578–589

Higgins, A.L., Wolfhord, R.E. 1986.A New Method of Text Independent Speaker Recognition. Uluslararası Akustik, Konuşma ve Sinyal İşleme Konferansı, IEEE, p869-872

Koç, Alpay. 1999. Acoustic Feature Analysis For Robust Speech Recognition. Bilkent Üniversitesi Elektrik Elektronik Mühendisliği Anabilimdalı Başkanlığı.

Linde, Y., Buzo, A., Gray, R. 1980. An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications, Vol.28, p.84-95

Marhon, S.A., Duaa, N.U. 2006. The Higher Institute For Industry.Libya.

McCullogh, W. S. ve Pitts, W. 1943, A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, p. 115-133

Martin, T.B. 1964. Speech recognition by feature abstraction techniques. Tech.Report ALTDR-64-176, Air Avionics Lab.

Moralı, İ.A., Aygün, F.F. 2006. Çok Katmanlı Algılayıcı ve Geriye Yayılım Algoritması ile Konuşmacı Tanıma. Dokuz Eylül Üniversitesi, İzmir,

Reynolds, D. A. 2002.An Overview of Automatic Speaker recognition Technology. MIT Lincoln Laboratory, Lexington, MA USA.

Saraç, T. 2004. Yapay Sinir Ağları Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Bölümü Anabilim Dalı.

Song, F, K. Rosenberg, A. E., Juang B. H. 1987. A Vector Quantization Approach to Speaker Recognition, *AT&T Technical Journal*, Vol.66-2, p.14-26

Vintsyuk, T.K. 1968. Speech discrimination by dynamic programming.Kibernetika, 4(2), p. 81–88

ÖZGEÇMİŞ

02.11.1981 Sarıkamış doğumlu olan Süheyla Başaran ilk ve orta öğrenimini Kars'ta tamamlamıştır. Daha sonra Erzincan Fen Lisesinde lise eğitimine devam etmiştir. Daha sonra okul değişikliği yaparak liseyi Eskişehir Cumhuriyet lisesinde bitirmiştir.

Lisans eğitimini 1999–2003 yılları arasında Dokuz Eylül Üniversitesi Elektrik Elektronik Mühendisliği bölümünde tamamlamıştır. Lisans tezinde dijital filtrelerde ‘Least Square Error Design’ yöntemi kullanılarak FIR filtrelerin hata oranını azaltma çalışmaları yapmıştır.

Uludağ Üniversitesi Elektronik Mühendisliği bölümünde yüksek lisans eğitimine başlamış ve burada Konuşmacı Tanıma yöntemleri hakkında tez çalışması yapmaktadır.

Halen TEDAŞ Kars İl Müdürlüğünde Elektrik Mühendisi olarak çalışmaya devam etmektedir.

TEŞEKKÜR

Bu çalışmayı yapmamda desteğini ve ilgisini hiç eksik etmeyen ve yüksek lisansa başladığım günden itibaren bana her konuda yardımcı olan danışman hocam Yrd. Doç. Dr. Figen ERTAŞ' a teşekkür ediyorum.

Ayrıca takıldığım noktalarda danıştığım ve birçok konuda beni bilgisiyle destekleyen Araştırma Görevlisi Cemal HANİLCİ' ye de yardımlarından dolayı teşekkür etmek istiyorum.

Eğitimim boyunca her zaman yanımda olan benden sevgisini, güvenini ve desteğini hiç esirgemeyen en zor anlarda bile her şeyin üstesinden gelebileceğime beni inandıran annem babam ve kardeşlerime sonsuz teşekkür ediyorum. Ayrıca her konuda bana yardımcı olan arkadaşım Seyhan Mutlu' ya yardımları ve desteği için teşekkür ediyorum.