

**OSMANLICA KARAKTERLERİN
YAPAY SİNİR AĞLARIYLA TANINMASI**

Davut ÖZHAN

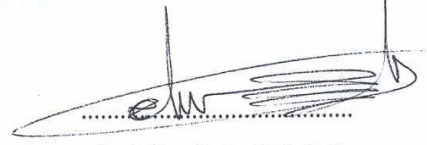
**YÜKSEK LİSANS TEZİ
ELEKTRİK EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2005

ANKARA

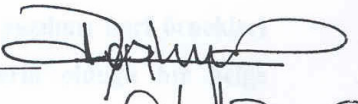
Davut ÖZHAN tarafından hazırlanan Osmanlıca Karakterlerin Yapay Sinir Ağları ile Tanınması adlı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylıyorum.





Prof. Dr. Çetin ELMAS

Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektrik Eğitimi Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

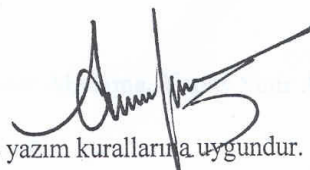
Başkan : Prof. Dr. İsmail COŞKUN 

Üye : Prof. Dr. Ömer Faruk BAY 

Üye : Prof. Dr. Çetin Elmas 

Üye : _____

Üye : _____



Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

**OSMANLICA KARAKTERLERİN YAPAY SİNİR AĞLARIYLA
TANINMASI
(Yüksek Lisans Tezi)**

Davut ÖZHAN

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Aralık 2005

ÖZET

Bu çalışmada el yazısı Osmanlıca harfleri tanımaya ilişkin bir Yapay Sinir Ağı (YSA) tasarlanmış ve uygulanmıştır. Uygulama ayrı yazılmış harf örnekleri üzerinde gerçekleştirilmiştir. Osmanlıca yazılı harflerin olduğu bir belge tarayıcıdan taranarak bilgisayar ortamına alınması işleminden sonra görüntü işleme teknikleri kullanılarak, taranan karakterler sayısal verilere dönüştürülmüştür. Verilerin düzenlenmesi için bir normalizasyon işleminden geçirilerek, YSA için giriş-çıkış değerleri elde edilmiştir. YSA'nın eğitim işlemi uygulamaları yapıldıktan sonra deneysel sonuçların değerlendirilmesi yapılmıştır.

**Bilim Kodu : 704.1.033
Anahtar Kelimeler : Resim İşleme, Karakter Algılama, Yapay Sinir Ağları,
Sayfa Adedi : 83
Tez Yöneticileri : Prof. Dr. Çetin ELMAS**

**RECOGNIZATION OF OTTOMAN TURKISH CHARACTERS BASED ON
ARTIFICIAL NEURAL NETWORKS**

(M.Sc. Thesis)

Davut ÖZHAN

GAZI UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

December 2005

ABSTRACT

In this thesis; an ANN (Artificial Neural Network) is designed and applied to recognize Ottoman Turkish Handwriting Characters. Application is written for separated character samples. First; a document which is written based on Ottoman Turkish Characters is scanned by scanner and get image of characters then applying image processing techniques to get digital data. Normalization process is applied for forming the data, and input-output values are obtained for ANN. After education process applications of ANN are completed, test results are criticized.

Science Code : 704.1.033
Key Words : Image Processing, Character Recognition, Artificial Neural Networks
Page Number : 83
Advisers : Prof. Dr. Çetin ELMAS

TEŐEKKÜR

Bu alıőmanın gerekleőtiriminde yardımcı ve teővik edici olan Hocam Sayın Prof. Dr. etin Elmas'a, ve ayrıca bu alanda yaptıkları alıőmalardan tecrübelerini aktaran Elektrik Eđitimi Bölümünden sayın Dr. Okan Bingöl, Araőtırma görevlileri Uđur Güven ve Serdar Birođuluna, arkadaşım Selahattin Gökdemir'e, ve desteklerini esirgemeyen aileme teőekkürü bir bor bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iii
ABSTRACT.....	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER	vi
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ	1
2. OSMANLICA	7
2.1. Tanım	7
2.2. Osmanlıca Yazı	9
3. GÖRÜNTÜ ANALİZİ VE KARAKTER TANIMA.....	12
3.1. Giriş.....	12
3.2. Optik Karakter Tanıma	12
3.3. İşaret ve Sayısal Görüntü	15
3.4. Görüntü Analizi.....	17
3.5. Gri Seviyeli Görüntü.....	18
3.6. İkili (binary) Görüntü.....	19
4. YAPAY SİNİR AĞLARI	20
4.1. Yapay Sinir Ağının Gelişimi.....	21
4.2. Bir Yapay Sinirin Ana Öğeleri.....	23
4.3. YSA' ların Üstünlükleri	26

	Sayfa
4.4. YSA' ların Sınıflandırılması	27
4.5. YSA' ların Uygulama Alanları	28
4.6. Yapay Sinir Ağı Mimarisi.....	28
4.7. YSA' larda Öğrenme.....	30
4.7.1. Danışmanlı öğrenme	31
4.7.2. Danışmansız öğrenme	33
4.7.3. Pekiştirerek öğrenme.....	33
4.8. Öğrenme Oranının Ağ Üzerindeki Etkisi.....	34
4.9. Momentum Teriminin Ağ Üzerindeki Etkisi	34
4.10. Gizli Katman Sinir Sayısının Ağ Üzerindeki Etkisi	34
4.11. Hata Farkı Değişkeninin Ağ Üzerindeki Etkisi	34
4.12. Danışmanlı Geri Yayılımlı Bir YSA Modeli	35
4.13. Arapça Karakter Tanıma Problemi İçin Kullanılan Yapay Sinir Ağlarının Karşılaştırılması	41
4.14. Segmentasyonsuz Kelime Tanıma Yönteminin Arapça İçin Kullanılması	51
5. OSMANLICA KARAKTERLERİN TANINIMASI İÇİN TASARLANAN YSA UYGULAMASI.....	61
5.1. Harf Karakterlerinin Tarayıcıdan Alınması	63
5.2. Resim İşleme.....	64
5.3. Resim Sayısallaştırma	67
5.4. Normalizasyon	68
5.5. Giriş-Çıkışların YSA'ya Uygun Hale Getirilmesi	69
5.6. YSA'nın Eğitilmesi.....	70
5.7. Deneysel Sonuçlar.....	73

	Sayfa
6. SONUÇ VE ÖNERİLER	78
KAYNAKLAR	80
ÖZGEÇMİŞ	84

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Osmanlıca harflerin başta, ortada ve sonda yazılışları.....	10
Çizelge 4.1. YSA türlerine göre uygulama alanları	28
Çizelge 4.2. En iyi sonucu veren ağların karşılaştırılması	51
Çizelge 4.3. Her biri bir sınıfa karşılık gelen 28 temel Arapça harf	52
Çizelge 4.4. Benzer şekle sahip Arapça harflerin aynı sınıfa atanması ile oluşan temel karakterler	52
Çizelge 4.5. Üç değişik test seti için elde edilen sonuçlar	60
Çizelge 5.1. Her bir harf karakteri için elde edilen deney sonuçları.....	74

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Osmanlıca genel alfabe	9
Şekil 3.1. Matris fonlu koordinat düzeni.....	16
Şekil 3.2. Bilgisayarla görme sistemi için genel blok diyagramı.....	17
Şekil 3.3. Gri tonlu bir görüntü	19
Şekil 3.4. İkili (Binary) görüntü	19
Şekil 4.1. Yapay bir sinir (düğüm).....	23
Şekil 4.2. Etkililik işlevleri.....	25
Şekil 4.3. Çok katmanlı bir YSA	30
Şekil 4.4. Danışmanlı öğrenme	32
Şekil 4.5. Danışmansız öğrenme	33
Şekil 4.6. Tek çıkışlı iki-katmanlı geri yayılmalı YSA.....	35
Şekil 4.7. Yerel minimum noktası	40
Şekil 4.8. Aynı temel şekle sahip Arapça karakterlerden bazıları	43
Şekil 4.9. ART1 ağının genel görünümü	44
Şekil 4.10. ARTMAP ağı'nın genel görünümü	45
Şekil 4.11. LVQ ağının yanlış sınıflandırdığı karakterler.....	48
Şekil 4.12. Kohenen ağı	49
Şekil 4.13. Tüm ağlar tarafından yanlış sınıflandırılan harfler	50
Şekil 4.14. Tanıma sisteminin genel şeması	53
Şekil 4.15. Tarayıcıdan alınan bir kelimenin görüntüsü	56

Sayfa

Şekil 4.16. Bir önceki şekilde görülen kelimenin görüntüsü üzerinde bir sembol sembol modelinin eşleşmesi	56
Şekil 4.17. Bir kelime görüntüsü üzerinde alt düzey bir şeklin kesişimi ve eşleşen alanlar	57
Şekil 4.18. Eşleme işlemindeki alt düzey şekiller. Her kare bir görüntü pikseline karşılı gelmektedir	58
Şekil 4.19. Birbiriyle en çok karıştırılan karakterler. karıştırma sıklığı soldan sağa doğru azalmaktadır.....	61
Şekil 5.1. Tasarlanan sistemin genel şeması	61
Şekil 5.2. Uygulama basamakları genel bilgileri içeren programın ön yüzü.....	62
Şekil 5.3. Tarayıcıdan taranan harf karakterini içeren belge	63
Şekil 5.4. Tarayıcıdan taranan bir harf karakteri.....	64
Şekil 5.5. BMP uzantılı bir resim	67
Şekil 5.6. Bir harf karakteri ve sayısal görüntüsü	68
Şekil 5.7. Normalizasyonda X ve Y uzunluklarını piksel cinsinden büyüklüğünü değiştirmek için kullanılan programın ön yüzü	69
Şekil 5.8. 10x10'luk YSA'nın girişlerini gösteren programın ön yüzü.....	69
Şekil 5.9. Tasarlanan ağ yapısı değerlerinin girildiği programın ön yüzü.....	71
Şekil 5.10. Uygulama için tasarlanan YSA ağ yapısı	73

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
ART	Adaptif Rezonans Teorisi
ARTMAP	Adaptif Rezonans Teori Eşleme
LVQ	Learning Vector Quantiser
Lpi	Lines Per Inch
ASCII	American Standart Code for Imformation
Rⁿ	ARTMAP Ağı İçin Giriş Vektörleri Kümesi
R^m	ARTMAP Ağı İçin Çıkış Vektörleri Kümesi
F1_a	ARTMAP Ağı İçin Örüntü Giriş Katmanı
F1_b	ARTMAP Ağı İçin Örüntü Giriş Katmanı
OCR	Optical Character Recognition
MSE	Mean Square Error
SSE	Sum Square Error
RMS	Root Mean Square
ρ	Vigilance Sabiti

1. GİRİŞ

İnsana özgü olarak bilinen yeteneklerinin kısmi olarak makinalara taklit ettirilmesi çok eski bir bilimsel düşünce olmasına karşılık, uygulama imkânları teknoloji ile paralel bir gelişim göstermiştir. Yapay zeka ile ilgili uygulamalarda kaydedilen ilerlemelerde, bilgisayar teknolojisinde sağlanan hızlı gelişmelerin önemli bir payı vardır. Çünkü Yapay zeka uygulamalarında, işleme giren veri sayısı ve ihtiyaç duyulan işlem hacmi büyük boyutlara ulaşmaktadır.

Yapay zeka, insan zekasına ait karar verme fonksiyonlarının sınırlı ölçülerde bilgisayarlara taklit ettirilmesidir. Yapay Zeka, zeka'nın "Bilgisayar yazılımı veya Tümüleşik yongalarla taklit edilmesidir". Yapay zeka; düşünme, anlama, kavrama, yorumlama ve öğrenme yapılarının bilgisayar programları ve tümleşik yongalarla taklit edilerek problemlerin çözümlerine uygulanması olarak ta ifade edilebilir (1).

Yapay zeka ile ilgili ilk fikirler 1950 yıllarında ortaya atılmaya başlanmış ve insan algılama sisteminin temelini oluşturan sinirlerin yapısından esinlenerek Yapay Sinir Ağı (YSA) modelleri geliştirilmiştir. Bireysel olarak sinirlere benzetilen YSA işlem elemanlarının, kendi aralarındaki bağlantı biçimleriyle oluşturdukları ağ yapısı ile yeni ve farklı bir bilgi işleme tekniği elde edilmektedir. Çünkü elde edilen model, öğrenme ve öğrenme kalitesine paralel olarak karar verme yeteneklerine sahiptir. Bu özellikler, herhangi bir uygulamaya ait insan bilgi ve tecrübesinin, ağ eğitimi sonucu elde edilen veriler ile bir sisteme aktarılabilmesine imkân vermektedir. Öğrenme aşamasının sistem başarımında öne çıkan bu çok önemli konumu nedeniyle, ağ eğitim algoritmaları sürekli geliştirilerek bu alanda yeni yöntemler üretilmektedir. Böylelikle, programlanabilir makinalara ayrıca eğitilebilirlik özelliği de ekleme süreci gün geçtikçe ilerleme kaydetmektedir. Bu doğrultudaki son gelişmeler, öğrenme kriterlerinin öğretilmesi yoluyla kendi kendine öğrenen sistem modelleri üzerinde yoğunlaşma çalışmalarını yansıtmaktadır (2).

Yapay zeka yöntemleri; çoğu yüksek teknoloji uygulamalarında aktif olarak yerini almaktadır. Ayrıca ekonomi ve sosyal bilimler gibi diğer alanlarda da karmaşık problemlerin çözümünde etkili bir biçimde kullanılmaktadır. Görüntü işleme ve tanıma alanında ise, görme yoluyla kalite kontrolü, ürün sınıflandırma, karakter tanıma, askeri amaçlı otomatik hedef tanıma, tıbbi görüntü analizi ve teşhis, robot hareketinin görme yoluyla denetimi gibi örnek uygulamalar gerçekleştirilmektedir (3).

Nesne tanıma işleminde ilk adım görüntünün kamera, dijital tarayıcı, ultrasonik tarayıcı ve benzeri araçlarla elde edilmesidir. Işık şiddetindeki değişim, yansıma, sahne görüntüsünün karmaşıklığı, görme açısındaki değişimler, görme mesafesi, görüntü işaretlerinde oluşan gürültü ve benzeri faktörler bu aşamada yer alan problemlerdir. Harici düzenlemeler dışında, alınan görüntüye yansıyan olumsuzlukların giderilmesi, azaltılması ve ayrıca tanıma aşamasına hazırlanmasında, görüntü işleme yöntemleri kullanılır. Görüntü iyileştirme sonrasında, algılanan görüntü içerisinde yer alan nesnenin genel görüntü çerçevesinden ayrıştırılması gerekmektedir. Sayısal formdaki görüntünün içerdiği renk ve ton farklılıkları, nesneyi arka zeminden ayırt etmeye yarayan özelliklerdir.

Bilgisayar teknolojisinin gelişimine paralel olarak, dökümanların bilgisayar ortamına aktarılması ve bunların işlenmesi çağımızda zorunlu bir ihtiyaç olmuştur. Ofis ve büro dökümanlarının dijital ortama aktarılması hem geriye dönük faaliyetlerin incelenmesine kolaylıklar getirmekte hemde zamandan tasarruf sağlamaktadır. Bunun yanında, günümüzde artık çok önem kazanan kütüphane arşivlerinin dijital ortama aktarılması gerekliliği ve arşivlerin çok büyük bir bölümünün Osmanlıca ve Arapça dillerinde yazıldığı gerçeği bilindiğinde konunun önemi daha da iyi anlaşılacaktır (2, 4).

10 adet Arapça rakamı tanıma konusunda yeterli başarı elde edilmiş olmasına rağmen Arapça harfleri özellikle yapay sinir ağları yardımıyla tanıma konusunda oldukça az sayıda yayın bulunmaktadır. Osmanlıca harfleri tanıma işlemi alfabe

oluşturan harflerin değişik durumlarda ortaya çıkan özelliklerinden dolayı oldukça zor bir iştir. Her harfin bulunduğu konuma göre dört değişik biçimi vardır.

Birbirine bağlı herhangi bir harf grubunda:

- 1) Başta
- 2) Ortada
- 3) Sonda
- 4) Diğer harflerden izole durumda olabilir.

Osmanlıca metinleri diğer dillerden ayıran önemli farklar şunlardır:

- Harfler birbirine bağlı gruplar şeklindedir. (Matbaa ya da el yazısı olması bu durumu değiştirmiyor.)
- Harfin şekli birbirine bağlı harf grubundaki konumuna göre değişmektedir.
- Aynı font için karakterler kalınlık olarak değişebilmekte, aynı karakterin bile değişik konumlarda değişik kalınlıkları olabilmektedir.
- Bazı karakterler düşey olarak aynı hizada olabilmektedir.

Osmanlıca metinler, matbaa ya da el yazısı olsun hep birbirine bağlı harf grupları içerdiği için, harfleri izole etme yani segmentasyon işlemi çok önem kazanmaktadır. Matbaa Osmanlıcasının tanınması bile, Latin el yazısını tanıma işleminden daha zor bir çalışmadır. Çünkü harf kalınlığı belirtildiği gibi aynı harf için bile değişik konumlarda değişmektedir (5).

Konu ile ilgili şu zamana kadar yapılmış diğer çalışmalardan ayrı olarak uygulamada 28 temel arapça harfe ilaveten 3 adet Farsçadaki pe (پ), ç (چ) ve (ج) harfleri de uygulamada kullanmışlardır.

Uygulama için tasarlanan sistem; Osmanlıca ayrıık harf karakterlerinin tarayıcıdan taranması basamağı ile başlayıp, resim işleme, resmin sayısallaştırılması, normalizasyon, giriş-çıkışların YSA'ya uygun hale getirilmesi, YSA' nın eğitilmesi ve en son sonuçların değerlendirilmesi basamağı ile son bulmaktadır.

El Yazısı Karakter Tanıma Alandaki Diğer Çalışmalar:

Fukushima (1982) tarafından gerçekleştirilen çalışma, elyazısı rakam tanıma alanındaki ilk örneklerdendir. Çalışmada konum ve şekil deformasyonuna ait değişimleri karşılayan, istatistik tabanlı bir algoritma geliştirilmiştir. Algoritma, piksel konumlarını referans alarak, şekil farklılıklarının olası durumlarını incelemekte ve karakter ölçek ve konumundan bağımsız olarak sınıflandırma işlemini gerçekleştirmektedir.

Lam ve Suen (1988) el yazısı posta kodlarını otomatik değerlendirilmesi amacıyla bir tanıma sistemi geliştirdi. Sistemde karakterin yapısal özellikleri kullanılmaktadır. Bu çalışmada sistem hızı açısından, iki aşamalı bir karar verme tekniğı kullanılmaktadır. İlk aşamada karakterin ana özellikleri ile hızlı tanıma işlemi denenmektedir. Eğer tanıma oranı güvenli sınırlar içerisinde değilse, doğruluk oranı yüksek bir tanıma sonucu üretmek amacıyla, ikinci aşamada karakterin daha detaylı özellikleri araştırılmaktadır.

Fukushima (1988) çok katmanlı ve hiyerarşik yapıda tasarlanmış bir ağ yapısı (neocognitron) kullanarak, piksel tabanlı tanıma sistemi gerçekleştirmiştir. Bu sistemde giriş katmanına uygulanan veriler, ağın her tabakasnda adım adım işlenerek bağlantıdaki diğer hücrelere aktarılır. Her katmanda karakterin farklı bir özelliğı teşhis edilmektedir. Katmanlar ilerledikçe seçiciliğın artması ve karmaşık özelliklerin ayrıştırılması sonucu, son katmanda tanıma sağlanmaktadır.

Mai ve Suen (1990) karakterin yapısal özelliklerinden elde ettikleri öznitelik bilgileri için, karakter sınıflarına göre tekrarlanma/oluşma sıklığına göre bir istatistik veri

tabanı oluşturmuşlardır. Sınıflandırmada YSA kullanımı ile birlikte, tanınamayan örneklerin teşhisinde, üretilen veri tabanına göre karar sağlanmaktadır.

Le Cun (1990) rakam karakterlerin normalize edilmiş formlarını kullanarak, piksel tabanlı bir tanıma uygulaması gerçekleştirmiştir. Uygulamada 4 katmanlı geri yayılım ağı (backpropagation network) tasarlanmıştır.

F. Kimura ve M. Shridhar (1991) rakam karakterlerin tanınmasında iki özellik setini birleştirerek kullanmaktadır. Bunlardan ilki, karakter sınır eğrisinden üretilen yerel doğrultu bileşenleridir. Diğeri ise karakter dış sınır eğrisinin sağ ve sol profillerinden elde edilen yapısal özelliklerdir. Sınıflandırma aşamasında ise tanıma başarısının yükseltilmesi amacıyla çoğul yöntem kullanımı ile tercih edilmiştir. Sınıflandırma illuminated, istatistik yöntemler ve karar aback kullanılarak bunların ayrı ayrı verimlilik oranları incelenmiştir. Yüksek tanıma oranı (%98) elde edilmekle birlikte, işlem yükü fazlalığı görülmektedir.

S. Whan Lee (1995) piksel tabanlı tanıma contemn benimsemiştir. Bununla birlikte, maskeleye yöntemleri ile karakterin alt bölgeler içerisindeki bölümleri icing ayrı ayrı yerel doğrultu bilgileri çıkartılmaktadır. Ayrıca alt bölgelerin içerdiği doğrultuların niteliğine göre, karakter bütünü temsil eden matris bilgisi üretilmektedir. Sınıflandırma aşamasında ise, elde edilen yerel ve küresel verilerin grup yapısına göre, her bir grubu karşılayan ve ayrı kümeler şeklinde bağlantı içeren çok katmanlı bir yapay sinir ağı tasarlanmıştır.

J. Basak (1995) karakterin iskelet formunu doğrudan giriş olarak kullanan, çok katmanlı yapay sinir ağı kullanarak el yazısı karakter tanıma sistemi gerçekleştirmiştir. Karakterin çizgisel bileşenlerine ait özniteliklerin oluşturulması, öz bağlantımlı (connectionist) olarak tasarlanan ağ yapısı üzerinde gerçekleşmektedir. 6 katman olarak tasarlanan ağın eğitim aşamasında, her katman üzerinde ayrı bir özniteliğin seçiciliği oluşturulur. Son katmanda ise bu bilgilerin entegrasyonu sağlanarak tanıma işlemi sağlanmıştır. Özellik çıkartımına göre düzenlenmiş özel hücre bağlantıları ile oluşturulan ağ yapısında, görüntü matrisini oluşturan nokta

sayısınınca giriş kullanılmıştır.

X. Li (1997) yazma anında (on-line) karakter tanıma contemn geliştirmiştir. Yöntemde, karakteri oluşturan çizgiler üzerinde baskın noktalar belirlenmektedir. Bu noktalar köşe noktaları ve düz çizgilerin orta noktaları olarak tanımlanmıştır. Sınıflandırma işlemi benzerlik ölçümleri ile gerçekleştirilmektedir.

Y.S. Hwang (1997) piksel tabanlı veri girişini benimsemiştir. Rakam karakterler üzerine yapılmış olan bu çalışmada, sınıflandırıcı olarak RBF (Radial Basis Function) ile eğitilen ağ yapısı kullanılmıştır.

D. Cheng ve H. Yan (1998) rakam karakterler üzerine gerçekleştirdiği çalışmada, karakter sınır eğrilerini temel alan bir yaklaşım sergilemiştir. Sınır eğrileri kullanılarak çevre uzunluğu, alan oranları, piksel uzaklık parametreleri gibi yapısal özelliklerin yanında, sınır eğrilerinin Fourier açılımından üretilen bilgiler tanıma amaçlı olarak kullanılmaktadır. Sınıflandırıcı olarak karar ağacı kullanılmıştır(5).

K. Yaman, ve diğerleri (2001) Ankara Hızlı Raylı Sistemde kameralar vasıtasıyla algılanan gri seviye görüntüleri, bilgisayar ortamına aktarılmıştır. Elde edilen sayısal değerler yardımı ile modelleme yapılmıştır.

Perdue University: Digital Image Processing Laboratory (2004) Image Halftoning diye adlandırılan bir yöntemle gray scale resimleri binary (sayısal) resimlere çevirme yaklaşımını gerçekleştirmişler.

A. Erdem ve Emre Uzun (2005) Yapay sinir ağları ile Türkçe Times New Roman, Arial ve Elyazısı karakterleri tanıma ile ilgili bir çalışma gerçekleştirmişler. Çalışma elyazısı karakterleri tanımada da başarılı olmuştur.

2. OSMANLICA

2.1. Tanım

13-20. yüzyıllar arasında Anadolu'da ve Osmanlı Devleti'nin hüküm sürdüğü yerlerde yaygın olarak kullanılmış olan, özellikle 15. yüzyıldan sonra Arapça ve Farsçanın etkisinde kalan türk yazın dili. Osmanlı Türkçesi ya da eski yazı olarak da bilinen Osmanlıca Arapça, Farsça ve Türkçenin karışımıdır ve Arap alfabesiyle yazılır (6).

Osmanlıca terimi Tazminat Dönemi (1839–1876) aydınlarınca ortaya atılmıştır. Daha önceleri Türk lehçelerinin hepsine Türki (Türkçe) ya da lisan-ı Türki (Türk dili) deniyordu. 19. yüzyılda artan milliyetçilik hareketlerine karşılık, Osmanlı Devleti'nin siyasal bütünlüğünü korumak amacıyla yeni bir milliyetçilikle ortaya çıkan Tanzimat Aydınları, millet-i Osmaniye (Osmanlı milleti) kavramını geliştirdiler. Osmanlı toprakları üzerinde konuşulup yazılan Türkçeye de Osmani (Osmanlıca) ya da lisan-ı Osmani (Osmanlı dili) adını verdiler.

Türkler tarih boyunca farklı din ve kültürlerle bir arada yaşadıkları için farklı alfabeler kullanmışlardır. 5. yüzyıldan 20. yüzyıla değin yakın ilişki içinde buldukları kültürlerin etkisiyle Göktürk, Uygur, Sogd, Çin, Tibet, Nasturi-Süryani, Mani, Brahmi, Peçenek, Kuman, Yunan, İbrani, Slav, Arap ve Latin alfabeleri değişik dönemlerde kullanılmıştır. Bunlar arasında Türklerin büyük bölümü tarafından en uzun süre (11. yüzyıldan 20. yüzyıla değin) kullanılan Arap alfabesidir.

Tarihsel gelişimi açısından Osmanlıca üç döneme ayrılır:

- 1- Eski Osmanlıca ya da Eski Anadolu Türkçesi: (13-15. yüzyıllar arası)
- 2- Orta Osmanlıca ya da Klasik Osmanlıca: (16-19. yüzyıllar arası)
- 3- Yeni Osmanlıca (19. yüzyıl-20. yüzyılın başları)

Eski Osmanlıca (Eski Anadolu Türkçesi):

Türklerin büyük bölümü 10. yüzyıla değin Uygur harflerini kullanıyordu. İslamiyet'in kabul edilmesinin ardından, Arap kültürünün etkisiyle Arap harfleri kullanılmaya başladı. 15. yüzyıla değin dilde Arapça ve Farsça sözcük ve tamlamalar azdı. Öte yandan 15. yüzyılda İstanbul'da başlayan saray yaşamı Arap, İranlı sanat ve bilim çevrelerini kendisine çekti; Türkçe'nin yanı sıra, Arapça ve Farsça yüksek sınıf ve aydınlarca kabul görmeye başladı. Bu yabancı öğeler 15. yüzyıldan sonra özellikle nazımda arttı.

Orta Osmanlıca (Klasik Osmanlıca):

16. yüzyıldan başlayarak Arapça ve Farsça yalnızca sözcük kullanımı olarak değil, dilbilgisi açısından da Türkçe'yi etkilemeye başladı. 19. yüzyıla değin süren bu dönemde Arapça ve Farsça tamlamalar yalnızca isim soylu sözcüklere değil fiillere de girdi. Kökü yabancı bileşik sözcükler oluşturuldu, düzyazı dilinde kısa ve yalın tümcelerin yerini bağlaçlarla uzatılmış yabancı öğelerle dolu tümceler aldı.

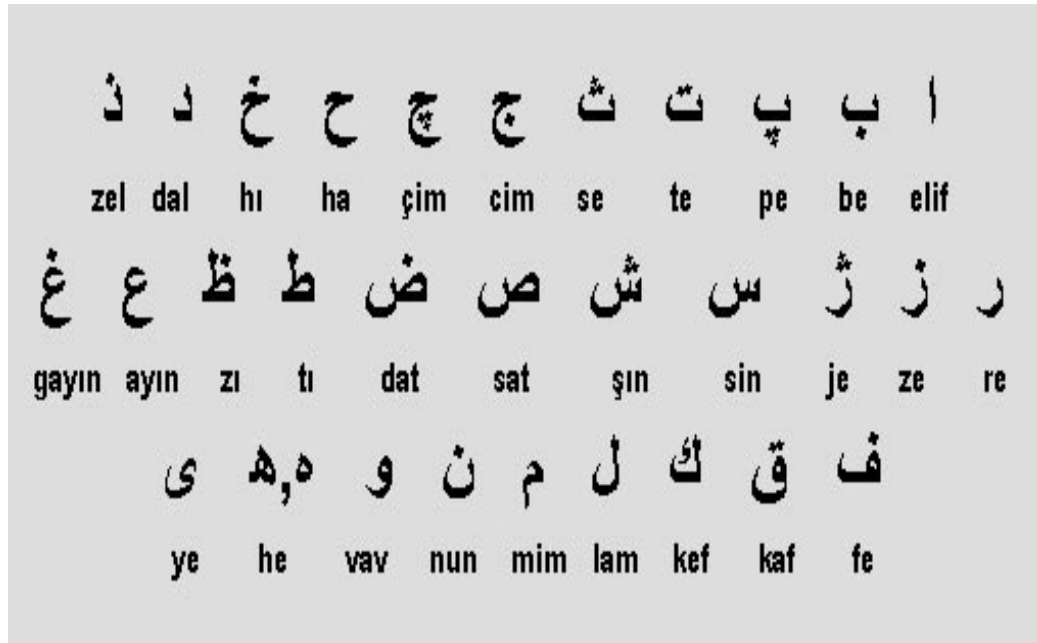
Yeni Osmanlıca:

19. yüzyılda Osmanlı Devleti'nde matbaanın kurulması, çeşitli konularda gazete ve dergilerin yayımlanması ve Batı kültürüne açılma çabaları düzyazıda etkisini gösterdi. Halkın anlayabileceği bir dille yazma girişimiyle dönemin yazar ve yayıncıları daha yalın bir dil kullanmaya başladılar. Yazıda ilk kez noktalama işaretleri kullanılırken, edebiyatta Batı etkisiyle roman, hikâye gibi yeni türlere ilgi arttı. Öte yandan Batı'dan alınan yeni kavramları (Batılılaşma, milliyetçilik, Osmanlıcılık vs.) karşılayacak Türkçe sözcüklerin olmaması sebebiyle yeniden Arapça ve Farsça sözcüklerle tamlamalara başvuruldu. Arapça ve Farsça sözcük köklerinden yeni sözcükler türetilti. Bu dönemin sonunda özellikle şiirde ağıdalı bir dil kullanılmasına (Edebiyat-ı Cedide, Fecr-i Âti) ve yabancı sözcüklere yer

verilmesine karşın, dilde yabancı öğelerden arındırma çabaları devam etti. Bu süreç 1928'deki Harf Devrimi'nden sonra daha da hızlandı.

2.2. Osmanlıca Yazı

Arap alfabesi 28 harften oluşur. Osmanlıcada Arap harflerinin yanı sıra Farsçadaki pe (پ), ç (چ) ve (ژ) harflerini de kullanmışlardır. Bu 31 harfin dışında Türkçedeki ince g ünsüzünü belirtmek için kef harfine bir çizgi eklenerek gef, genizsi n ünsüzü için üç nokta eklenerek nef (sağır kef, kâf-ı nunî), lam ile eliften lamelif, hemze ile h harfinin ünlü şekli olan hâ-i resmiye harfleri oluşturulmuştur.



Şekil 2.1. Osmanlıca genel alfabesi

Osmanlıca sağdan sola doğru yazılır. Arap harflerinde temel ve küçük harf ayrımı yoktur. Noktalama işaretlerinde kesin kurallar bulunmamaktadır. Arap harfleri sözcüklerin başında, ortasında ve sonunda farklı biçimde yazılır. Bazı harfler (dal, zel, re, je, vav, د, ذ, ر, ز, ژ, و) bir sonraki harfle birleşmez.

Çizelge 2.1. Osmanlıca harflerin başta, ortada ve sonda yazılışları

İsimleri	Harfler	Sonda (sağdan bitişik)	Ortada (her iki taraftan bitişik)	Başta (soldan bitişik)	Karşılıkları
elif	ا	ا	-	-	a, e
hemze	ء	ء	ء	ء	ʾ(a, e, i, u, ü)
be	ب	ب	ب	ب	b
pe	پ	پ	پ	پ	p
te	ت	ت	ت	ت	t
se	ث	ث	ث	ث	s
çim	ج	ج	ج	ج	c
çim	چ	چ	چ	چ	ç
ha	ح	ح	ح	ح	h
hu	خ	خ	خ	خ	h
dal	د	د	-	-	d
zel	ذ	ذ	-	-	z
re	ر	ر	-	-	r
ze	ز	ز	-	-	z
je	ژ	ژ	-	-	j
sin	س	س	س	س	s
şın	ش	ش	ش	ش	ş
sat	ص	ص	ص	ص	s
dat	ظ	ظ	ظ	ظ	d, z
tı	ط	ط	ط	ط	t
zı	ظ	ظ	ظ	ظ	z
ayın	ع	ع	ع	ع	ç, c
gayın	غ	غ	غ	غ	g, ğ (kalm)
fe	ف	ف	ف	ف	f
kaf	ق	ق	ق	ق	k
kef	ك	ك	ك	ك	k, g, ğ(y), n
gef	گ	گ	گ	گ	g, ğ
nef, sağır kef	ك	ك	ك	ك	n
lam	ل	ل	ل	ل	l
nım	م	م	م	م	m
nun	ن	ن	ن	ن	n
vav	و	و	-	-	v, o, ö, u, ü
he	ه	ه	ه	ه	h, e, a
lame lif	لا	لا	-	-	la
ye	ی	ی	ی	ی	y, i, i

Arap harflerinin Türkçe'deki zengin ünlü sistemini karşılamada yetersiz olduğu düşünülür. Örneğin Arap alfabesindeki elif (ا) Türkçe'deki a ve e ünlüsünün karşılığıdır ya da Türkçe'deki u, ü, o, ö ünlülerinin yerine Arapça da yalnızca (و) harfi vardır, bu aynı zamanda v ünsüzünün de karşılığıdır.

Osmanlıca da Arap rakamları kullanılır. Araplar Hint rakamlarıyla tanıştıktan sonra ondalık (onlu, 10 tabanlı) sayı sistemini benimsediler. Günümüzde de kullanılan bu sistem 8-11. yüzyıllar arasında Hintli ve Arap matematikçileri tarafından geliştirilmiştir. Rakamlar yazının tersine soldan sağa doğru yazılır.

Şekil 2.2. Osmanlıca rakamlar



3. GÖRÜNTÜ ANALİZİ VE KARAKTER TANIMA

3.1. Giriş

Bu bölümde, görüntüye ilişkin temel kavramlar, görüntü analizi ve optik karakter tanıma ile ilgili açıklamalar yapıldıktan sonra el yazısı karakter tanıma ile ilgili çalışmalara yer verilecektir.

3.2. Optik Karakter Tanıma

Karakter tanıma işlemini, insan-makine iletişimini kolaylaştırmak üzere elle veya matbaa harfleriyle yazılmış metinleri doğru biçimde bilgisayara tanıtmak olarak tanımlayabiliriz. Optik karakter tanıma, son yıllarda araştırmacıların üzerinde yoğun çalıştıkları konulardan biri olmuş ve günlük hayata inen uygulama örnekleri sergilenmiştir (7). Optik karakter tanımanın başlıca kullanım alanları şunlardır:

- Kütüphane dokümanlarının bilgisayar ortamına hızlı aktarılması,
- Posta dokümanlarının dağıtım öncesi hızlı sınıflandırılması,
- Bilgi ve belge yönetimi,
- Arşivlerin dijital ortama aktarılması,
- Ofis dokümanlarının (çek, senet vb.) otomatik sınıflandırılması ve değerlendirilmesi,
- Endüstride ürün veya malzemelerin etiketlerine göre seçimi, sınıflandırılması,
- Kalite kontrol ve otomatik paketleme,
- Görme engelliler için okuma desteği,

Makina baskısı karakter tanıma uygulamaları, düzenli yazılmış doküman veya simgelerin sayısal formda bilgisayar ortamına aktarılmasında önemli bir araç teşkil

etmektedir. Günümüzde, bu amaçla oluşturulan ticari yazılımların kullanımı giderek yaygınlaşmaktadır.

El yazısı karakter tanıma, optik karakter tanımanın önemli bir alt konusunu oluşturmakla birlikte, tanıma açısından daha karmaşık ve zor bir problemi ortaya koymaktadır. Bu alandaki temel zorluk, kişilere bağlı olarak sonsuz sayıda denebilecek yazma biçimlerinin bulunmasıdır. Dolayısıyla aynı sınıfa ait karakter örnekleri arasında büyük farklılıklar ortaya çıkmaktadır. Sözü edilen farklılıkları tanımlayan herhangi bir matematiksel model tanımlamak mümkün değildir. Ayrıca, farklı yazma biçimleri ile değişmeyen bir öznitelik bulunmamaktadır. Bu yüzden, aynı sınıfa ait örnekler için küçük değişimler üreten uygun bir öznitelik seti tanımlamak gereklidir. Bir el yazısı karakter tanıma yönteminin ihtiyaç duyduğu temel özellikler şu şekilde özetlenebilir.

Öznitelik çıkarma yöntemleri, yapısal ve istatistiksel bilgi tabanlı olmak üzere iki yaklaşım sergilemektedir. Yapısal yaklaşımda, verilen bir örneğin formel yapıda temel karakteristik tanımlaması elde edilmeye çalışılır. Örüntülerdeki anlamlı bilgiler, çoğunlukla özellikler setinin sayısal değerlerinde değildir. Fakat özelliklerin birbirine bağlanması veya aralarındaki ilişkiler şeklinde anlam kazanır. Başka bir deyişle, yapısal yaklaşımda her obje bileşenlerinin bir kompozisyonu olarak ele alınır. Örneğin bu yöntemlerde karakterin geometrik yapısal özellikleri (köşe noktaları, kesişimler, çizgisel ve yay bileşenleri, içbükey-dışbükey kriterleri vb.) üretilebilir. Üretilen özelliklerin hiyerarşik yapısı tanıma verisini oluşturmaktadır. Çoğu sözdizimsel (syntactical) sınıflandırma teknikleri bu gruba dahil edilir. İstatistiksel yaklaşımda, öznitelik denilen karakteristik ölçümler istatistik analiz yöntemleri ile üretilir. İstatistiksel tabanlı yöntemler, (kenar özellikleri, moment değişmezleri, dönüşüm teknikleri, fourier açılımı vb.) şekil pikselleri arasındaki komşuluklar ile bağımlılık ilişkileri ele alınarak şekil özelliklerini yansıtan bilgiler oluşturma temeline dayanır. El yazısı karakter tanıma alanında, yapısal ve istatistiksel yöntemlere dayalı çalışmaların yanında bunların birlikte kullanımına örnek teşkil eden çalışmalar da yer almaktadır.

El yazısı karakter tanıma uygulamalarını kullanımı açısından iki ana sınıfa ayrılmaktadır.

1- Yazım anında tanıma (On-line): Yazma sırasında, hareket dedektörleri ile kalem hareketleri değerlendirilerek tanıma yapılır.

2- Yazım sonrası tanıma (Off-line): Yazılmış karakterlerin optik algılama yolu ile analizi sonucu tanıma gerçekleştirilir.

Yazım anında tanımanın bir avantajı, yazma anında oluşan hareket biçimleri ile dinamik bilgilerin elde edilebilmesidir. Ayrıca optik algılama ile ilgili gürültü ve arka zemin uyumsuzlukları gibi problemler oluşmamaktadır. Yazma anında algılamanın mümkün olmadığı uygulamalar için, optik algılama ve yazım sonrası tanıma yöntemleri kullanılır

Optik karakter tanıma alanındaki ilk örnekler 1959 yılında Latin karakterleri tanıma çalışmaları ile başlamıştır. Bu dönemde kullanılan bilgisayarların günümüze oranla daha ilkel oluşu ve hız gerektiren işlem yükü uygulamaların temel zorluklarını oluşturmuştur. Daha sonra 1966' da Çin karakterlerini tanıma üzerine ilk araştırma örnekleri verilmeye başlanmıştır. Literatürde, Arapça karakter tanımaya ilişkin çalışmaların ise 1980'li yıllarda başladığı görülmektedir. Metin tanıma sistemlerinde, sembollerini segmente etme, yani her sembolü diğerlerinde izole ederek bağımsız olarak değerlendirmek en kritik işlem olarak karşımıza çıkar. Sembollerini doğru olarak tanımayı etkileyen faktörleri genel olarak şöyle sıralayabiliriz:

- Yazma biçimi
- Taranan sayfanın fiziksel durumu (tarayıcıya doğru şekilde konma biçimi)
- Sayfanın kalitesi
- Metinden başka oluşan lekeler
- Tarayıcının kalitesi ve çözünürlüğü

El yazısı bir metni tanıtmada, matbaa harflerinden oluşan metni tanıtmadan daha zordur. Bu yüzden pratik alanda matbaa harflerini tanıma daha çok uygulanmıştır. Çoğu zaman, tanınacak semboller birbirine bağlı parçalardan oluşabilir. Latin harfleri el yazısıyla yazıldığında harfler birbirine bağlıyken, osmanlıca metinler matbaa harfleriyle de yazılsa birbirine bağlı harf grupları şeklinde karşımıza çıkar. Yazının doğasından olduğu kadar, değişik nedenlerden (tarayıcı çözünürlüğü, yazıcı veya kağıt kalitesi) dolayı ortaya çıkan gürültü (noise) de harfleri bitiştirebilir veya tek bir harfi parçalayabilir. Bu sorun genellikle bağlılık veya kesimleme (connectivity & segmentation) olarak bilinir (2, 4, 5).

3.3. İşaret ve Sayısal Görüntü

Bir sisteme ait ve bilgi anlamı taşıyan değişimler işaret olarak tanımlanır. İşaret, oluşumunda etkin olan tek veya çok boyutlu değişken veya değişkenlerin bir fonksiyonu olarak modellenir. Fiziksel işaretler üzerinde işlem yapabilmek amacıyla, önce elektriksel forma dönüştürülür. Bu aşamada işleme yönelik, çeşitli dönüştürücü (mikrofon, kamera, ısı sensörü vb.) elemanlar kullanılır. Zamana göre davranışları açısından işaretleri iki ana gruba ayrılabilir.

Sürekli işaretler: Zaman ile süreklilik bağıntısı içeren fonksiyonlar ile tanımlanan işaretlerdir.

Ayrık işaretler: Sadece, belirli periyodik zaman aralıklarında tanımlanmış işaretlerdir.

Bilgisayar ortamında verilerin, sayısal formda ve seri olarak işlenmesi işaretlerin ayrık zamanlı forma dönüştürülmesini gerekli kılar. Bu yüzden, sürekli işaretler örnekleme yöntemleri kullanılarak ayrık formda tanımlanmaktadır.

Bir sayısal görüntü, satır ve sütun indisleri görüntü içerisinde herhangi bir noktayı tanımlayan elemanlardan meydana gelmiş bir matris olarak göz önüne alınabilir. Bu

matrisin her bir elemanının sayısal değeri, kendisine karşılık gelen noktalardaki gri seviye değerine sahiptir.

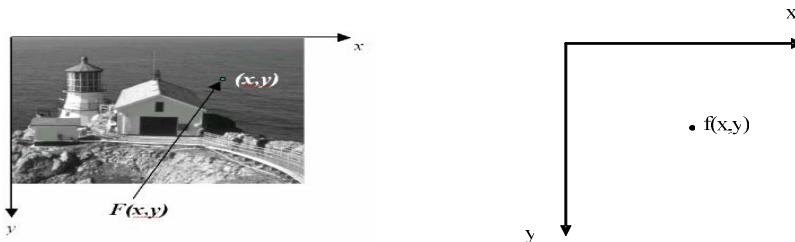
Bu sayısal dizinin veya matrisin her bir elemanına görüntü elemanı, resim elemanı veya "piksel" adı verilir. Pikseller bireysel olarak görüntüye ait renk ve parlaklık gibi niteliklerini yansıtırlar. Bunun gerisinde sözü edilen niteliklerin, herbir piksel için sayısal bilgi formuna dönüştürülmüş biçimi "sayısal görüntü" olarak tanımlanır.

Yüksek çözünürlüklü ve renk ayrıntılarını taşıyacak biçimde algılanmış bir görüntü, daha küçük piksellerden oluşmakta ve daha fazla sayısal veri ile temsil edilebilmektedir.

Sayısal görüntünün matematiksel açılımı iki boyutlu bir matris ile ifade edilir. Matris elemanları matris içi sıralanma ile düzeni ile tanımlanan piksel konumlarını temsil ederken, sayısal değer itibariyle renk ve ton gibi nitelikleri yansıtmaktadır. Böylelikle alınan bir G görüntüsü, genel biçimiyle AXB ölçekli bir matris halinde (3.1) eşitliğinde tanımlandığı gibi işlemlerde yer alır (2, 8).

$$G = \begin{bmatrix} g(1,1) & g(1,2) & \dots & g(1,B) \\ g(2,1) & g(2,2) & \dots & g(2,B) \\ \cdot & \cdot & \dots & \cdot \\ g(A,1) & g(A,2) & \dots & g(A,B) \end{bmatrix} \quad [3.1]$$

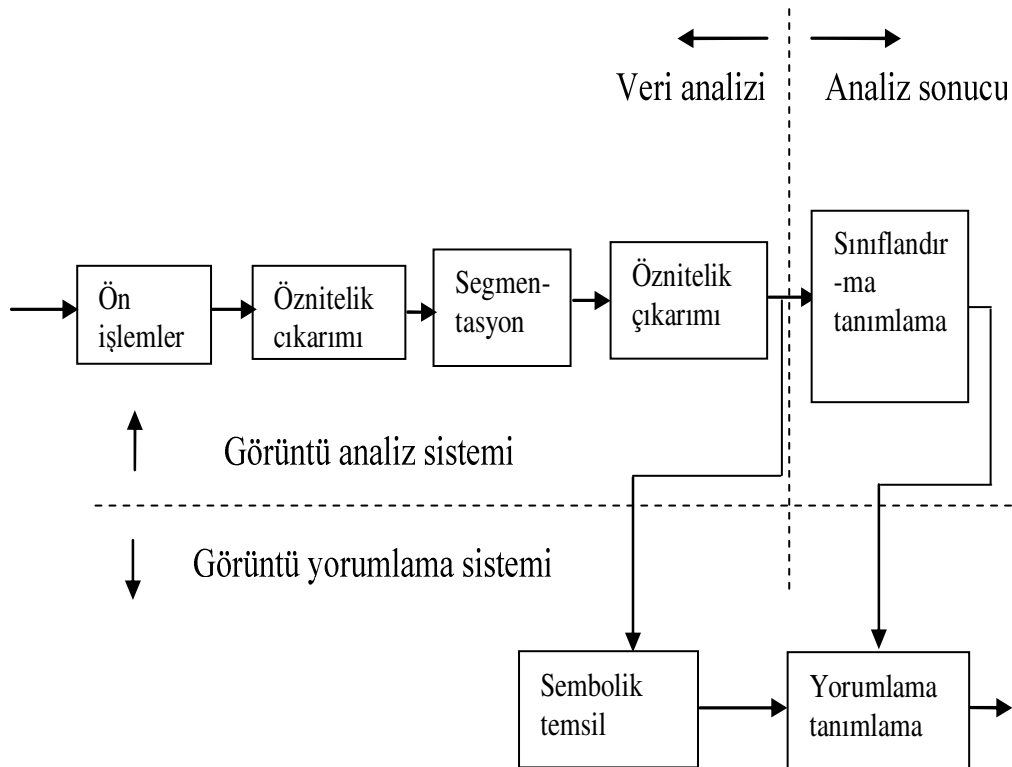
Bu formata göre herhangi bir pikselin konumu, Şekil 3.1 'de tanımlanan koordinat sistemi ile belirtilir.



Şekil 3.1. Matris fonlu koordinat düzeni

3.4. Görüntü Analizi

Bilgisayarla görme uygulamalarında, görüntü algılamadan tanıma aşamasına kadar gerçekleşmesi gereken bir dizi işlem yer almaktadır. Bunlara ilişkin klasik yöntemlerin bulunmasına karşılık, her işlem basamağı ayrı bir araştırma konusu olarak incelenmekte, uygulama alanına göre başarıyı arttıran ve işlem süresini azaltan yeni yöntemler üzerinde çalışmalar sürdürülmektedir. Bütün olarak ele alındığında, bir görme sistemi Şekil 3.2' de görüldüğü gibi genel bir yapı üzerine kurulmuştur. Alınan sayısal görüntü öncelikle ön işlemlerden geçirilir. Ön işlemlerin amacı, görüntü iyileştirme, geliştirme, onarma veya sonraki aşamaya hazırlık şeklinde özetlenebilir.



Şekil 3.2. Bilgisayarla görme sistemi için genel blok diyagramı

Öznitelik çıkartımı (feature extraction), yapılacak işlem için görüntüden bazı özellik bilgilerinin elde edilmesini amaçlar. Alınan görüntü içerisinde yer alan nesnelerin,

belirli dokuların veya özelliklerin seçilebilmesi ve belirgin hale getirilerek ayrıştırılması segmentasyon aşamasında gerçekleştirilir. Oysa bu işlemlerin sağlıklı olarak gerçekleştirilmesi her zaman sabit kurallar ile yapılamamaktadır. Çünkü alınan her görüntü, farklı renk veya ton özellikleri taşıyabilir. Bu yüzden segmentasyon öncesinde, görüntü yapısına ait öznitelik çıkarma yöntemlerine başvurulur.

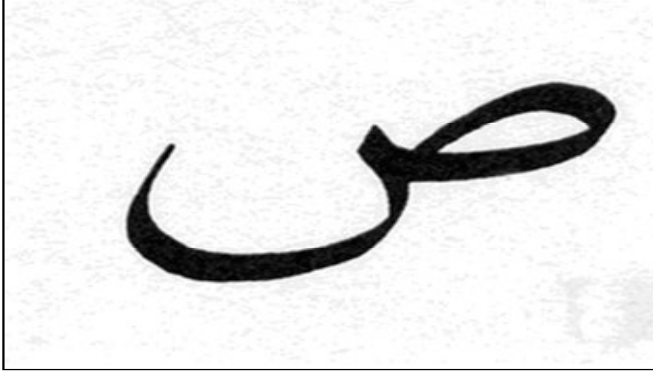
Elde edilen öznitelikler bu aşamada, görüntüdeki ihtiyaç duyulan ve fark edilebilen özelliklerin ayrıştırılmasında önemli rol oynamaktadır. Örneğin, segmentasyon işlemi, görüntü içerisindeki nesnelere arka zeminden ayrıştırılması sağlanabilir.

Segmentasyon sonrasında, görüntü içerisinde yer alan nesne veya dokular üzerinde uygulama amacına göre, ikinci bir öznitelik çıkartma işlemi yapılır. Elde edilen öznitelik bilgileri görüntüye ait özelliklerin sayısal verilerle temsil edilmesini sağlar. Bu aşama sonunda gerekirse benzer nesnelere veya bölgelere ayrı gruplar halinde etiketlenerek sınıflandırılabilir (2, 5, 8).

3.5. Gri Seviyeli Görüntü

Makina görmesine dayalı olarak birçok uygulamada, algılanan görüntünün renk özelliklerinin temsil edilmesine gerek yoktur. Görüntü, siyah beyaz arası gri tonlardan oluşacak şekilde algılanabilir veya renk özellikleri, algılama sonrasında ilgili gri tonlara dönüştürülebilir.

Bu durumdaki bir görüntü içerisinde, ışık şiddeti, yansıma gibi özellikler muhafaza edilebilmektedir (Şekil 3.3). Görüntüyü oluşturan pikseller ise, gri tonları yansıtacak şekilde farklı gri seviye kodları ile temsil edilmektedir. 8 bitlik piksel tanımlama formatında, gri tonlara ait pikseller 0-255 arasında seviye değerleri ile temsil edilir. Burada 0 beyaz, 255 ise siyah rengi temsil etmektedir.

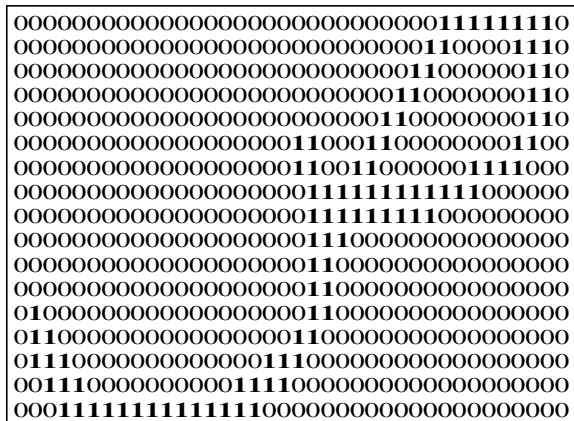


Şekil 3.3. Gri tonlu bir görüntü

3.6. İkili (binary) Görüntü

Görüntü piksellerinin taşıdığı gri tonların seçilen bir eşik seviyesine göre siyah beyaz olarak iki seviyeye dönüştürülmesi sonucu, ikili görüntü elde edilmektedir. İkili görüntüde nesneyi tanımlayan siyah pikseller 1, arka zemini oluşturan beyazlar ise, 0 değeri ile temsil edilir. Şekil 3.4' de ikili görüntü örneği görülmektedir. Tanıma amaçlı öznetelik çıkarma yöntemleri ikili görüntü üzerinde de başarı ile kullanılabilmektedir (2-4, 9).

Bu aşamada eşik değerinin belirlenmesi, sağlıklı bir dönüşümün elde edilmesinde önem kazanmaktadır.



Şekil 3.4. İkili (Binary) görüntü

4. YAPAY SİNİR AĞLARI

Yapay sinir ağı (YSA), insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır. YSA, bir başka deyişle, biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır. YSA zaman zaman bağlantıcılık (connectionism), paralel dağıtılmış işlem, sinirsel-işlem, doğal zeka sistemleri ve makine öğrenme algoritmaları gibi isimlerle de anılmaktadır (10, 11).

YSA bir programcının geleneksel yeteneklerini gerektirmeyen, kendi kendine öğrenme düzenekleridir. Bu ağlar öğrenmenin yanı sıra, ezberleme ve bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptir.

YSA insan beyninin bazı organizasyon ilkelerine benzeyen özellikleri kullanmaktadırlar. YSA bilgi işleme sistemlerinin yeni neslini temsil ederler. Genel olarak YSA model seçimi ve sınıflandırılması, işlev tahmini, en uygun değeri bulma ve veri sınıflandırılması gibi işlerde başarılıdır. Geleneksel bilgisayarlar ise özellikle model seçme işinde verimsizdir ve sadece algoritmaya dayalı hesaplama işlemleri ile kesin aritmetik işlemlerde hızlıdırlar.

Bir sinir ağının en temel bileşeninin beyinin yapısından esinlenerek geliştirilmelerine karşın, bazı özellikleri beyinle aynı değildir. Bazı özelliklerinin ise beyinde biyolojik karşılığı yoktur. Bununla birlikte sinir ağları biyolojik beyinle büyük benzerlikler gösterir ve bu nedenle terimlerin büyük kısmı sinirbilim (neuroscience) biliminden alınmıştır.

YSA' nın yararlı ve ilgi çekici birçok özellikleri vardır. İlk özellik öğrenebilme yeteneğidir. YSA uyarlanabilir ve esneyebilir. Tek bir YSA, herhangi bir işleve ilke olarak yaklaşabilir. YSA' nın yapısını ve öğrenme kurallarını değiştirmek zorunda değiliz, sadece öğretim materyalini değiştirerek öğrenmeyi sağlayabiliriz. Daha da

önemlisi öğretim materyalini değiştirmek temel olarak yeni bir çevreyle tanışmak gibidir. YSA' nın en büyük üstünlükleri, öğrenme kabiliyeti olması ve farklı öğrenme algoritmaları kullanabilmesidir. Bunun yanı sıra en sık belirtilen sakıncaları ise sistemin çalışmasının analiz edilememesi ve öğrenme işleminde başarılı olunamama riski olmasıdır.

4.1. Yapay Sinir Ağının Gelişimi

İnsan beyni hakkındaki çalışmalar binlerce yıl öncesine dayanır. Modern elektroniğin gelişmesi ile birlikte, bu düşünce işlemini kullanmaya çalışmak doğal bir hale gelmiştir. İlk YSA modeli 1943 yılında, bir sinir hekimi olan Warren McCulloch ile bir matematikçi olan Walter Pitts (12) tarafından gerçekleştirilmiştir. McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden esinlenerek, elektrik devreleriyle basit bir sinir ağını modellemişlerdir.

1948 yılında Wiener "Cybernetics" isimli kitabında, sinirlerin çalışması ve davranış özelliklerine değindi. 1949 da ise Hebb (13) "Organization of Behavior" isimli kitabında, öğrenme ile ilgili temel teoriyi ele aldı. Hebb kitabında öğrenebilen ve uyum sağlayabilen sinir ağları modeli için temel oluşturacak Hebb Kuralı'nı ortaya koymuştur.

1957 yılında Frank Rosentblatt'ın (14) Perceptron'u gerçekleştirmesinden sonra YSA alanındaki gelişmeler hızlanmıştır. Perceptron, beyin işlevlerini modelleyebilmek amacıyla yapılan çalışmalar neticesinde ortaya çıkan tek katmanlı eğitilebilen ve tek çıkışa sahip olan YSA' dır.

1959 yılında, Bernand Widrow ve Marcian Hoff (15) Stanford Üniversitesinde ADALINE ve MADALINE diye adlandırdıkları ağ modellerini geliştirdiler. MADALINE, telefon hatlarında oluşan yankıları yok eden bir uyarlanabilir süzgeç olarak kullanılmış, gerçek dünya sorunlarına uygulanmış olan ilk sinir ağıdır ve hala kullanımda bulunmaktadır (17).

1969 yılında Minsky ve Papert (18) Perceptron'un yetersizliğini görmüşler ve XOR problemini çözemediğini ispatlamışlardır. Bunun için iki katmanlı ileri beslemeli ağların kullanılabileceğini ileri sürmüşler ve tek katmanlı ağlardaki birçok sınırlamayı ortadan kaldırdığını göstermişlerdir. Fakat gizli katmanların ağırlıklarının nasıl değiştirileceği konusunda bir yöntem önerememişlerdir. Bu soruna Rumelhart (19) ve arkadaşları geri yayımlı YSA yöntemi ile bir çözüm getirmişlerdir.

1970' li yıllarda yapay sinir ağları hakkında birkaç araştırmacı çalışmasına rağmen, Grossberg ve Kohonen önemli ilerlemeler kaydettiler (20), Grossberg (21) biyolojik sinir sistemlerinin temel prensiplerinden yararlanarak kendine ait uyarlanabilir rezonans teorisini (ART) geliştirmiştir ve ağın dinamik yapısıyla birinci dereceden diferansiyel denklemleri modellemiştir. Bu fikirler Carpenter ve Grossberg tarafından sonraki çalışmalarına esas teşkil edip ART' yi üç sınıfa ayırmışlardır. Bunlar ART1 (22), ART2 (23), ART3 (24)' dir. Bunlar kendi kendine organize örnek sınıflandırma algoritmalarının sinir uygulamalarıdır. Diğer taraftan Kohonen çalışmalarında kendi kendine organize sistem haritalarını geliştirmiştir (25). Bulduğu bu ağ yapısı ile özel ve farklı örnekleri ayarlayarak sinirleri organize etmiştir.

1970 yılında Werbos (26) geri yayımlı YSA algoritmasını geliştirmiştir. Onun ilk pratik uygulaması dinamik model tahmini olmuştur. 1980' li yılların başında Hopfield geriye yayımlı ağların eğitmeleri için kendi adını verdiği Hopfield modelini ortaya koymuştur (27, 28). Sonraları Kosko, Hopfield ve Grossberg' e ait fikirlerinin bazılarında yararlanarak uyarlanabilir iki yönlü ilişkili hafıza (BAM) modeli geliştirmiştir (29). Diğer önemli modeller daha sonraki yıllarda bulunmuştur.

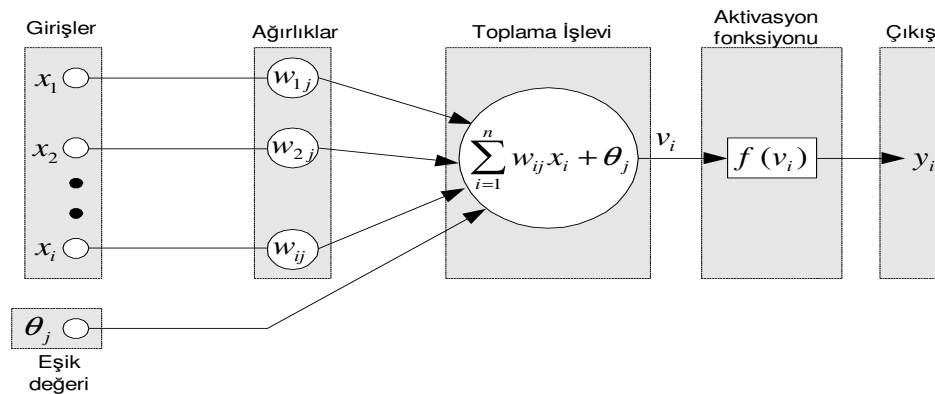
1980' li yıllarının ortalarında Parker (30), Rumelhart ve arkadaşları (31) geri beslemeli ileri yayımlı sinir ağını tekrar ele almışlardır Hinton, Sejnowski ve Ackley (32, 33) Hopfield ağı çeşitlerinden Boltzmann makinesini geliştirmişlerdir. Onların bu çalışmaları diğer araştırmacıların çalışmalarında yön vermiştir. Fukuda ve Meireles yapmış oldukları çalışmalarında son zamanlarda YSA' ların çok farklı alanlarda geniş uygulama alanları bulduklarını belirtmişlerdir (20, 34).

4.2. Bir Yapay Sinirin Ana Öğeleri

YSA, birbirine bağlı çok sayıda işlem elemanlarından oluşmuş, genellikle paralel işleyen yapılar olarak adlandırılabilir. YSA' daki işlem elemanları (düğümler) basit sinirler olarak adlandırılır. Bir YSA, birbirleriyle bağlantılı, çok sayıda düğümlerinden oluşur. YSA, insan beyni gibi, öğrenme hatırlama ve genelleme yeteneği ne sahiptirler (10).

Nöronlar sinir ağını oluşturan basit işlevlere sahip işlemcilerdir. Bir nöron yapısı içerisinde üç ana bölüm bulunur. Bunlar sırası ile ağırlıklar, toplama işlevi ve aktivasyon (etkinlik işlevi) fonksiyonudur. Şekil 4.1'de sinir ağlarında kullanılan genel bir sinir modeli verilmiştir. $i = 1, \dots, n$ için x_i ağırlık girişlerini gösterir. w_{ij} i 'inci giriş ile j 'inci düğüm arasındaki bağlantı ağırlıklarını gösterir.

Şekil 4.1'de görüleceği gibi giriş, bağlantılar üzerindeki ağırlıklar ile çarpılarak bir toplayıcıya uygulanmakta ve elde edilen toplam, nöronun aktivasyon fonksiyonundan geçirilerek çıkışlar hesaplanmaktadır. Eş. 4.1' de bir sinir için sistemin girişlerine göre ağırlık toplamı gösterilmiştir.



Şekil 4.1. Yapay bir sinir (düğüm)

$$v_i(t) = w_{1j}x_1(t) + w_{2j}x_2(t) + \dots + w_{ij}x_n(t) + \theta_j = \sum_{i=1}^n w_{ij}x_j(t) + \theta_j \quad [4.1]$$

olur. Sonuç olarak toplama işleminin sonucu aktivasyon fonksiyonunun geçirilir

$$y_i = f(v_i) \quad [4.2]$$

Tüm YSA bu temel yapıdan türetilmiştir. Bu yapıdaki farklılıklar YSA' nın farklı sınıflandırılmalarını sağlar. Bir yapay sinirin öğrenme yeteneği, seçilen öğrenme algoritması içerisinde ağırlıkların uygun bir şekilde ayarlanmasına bağlıdır.

Girişler: Girişler (x_1, x_2, \dots, x_n) çevreden aldığı bilgiyi sinire getirir. Girişler, kendinden önceki sinirlerden veya dış dünyadan sinir ağına gelebilir. Bir sinir genellikle gelişi güzel birçok girdileri alır.

Ağırlıklar: Ağırlıklar $(w_{1j}, w_{2j}, \dots, w_{nj})$ yapay sinir tarafından alınan girişlerin sinir üzerindeki etkisini belirleyen uygun katsayılardır. Her bir giriş kendine ait bir ağırlığa sahiptir. Bir ağırlığın değerinin büyük olması, o girişin yapay sinire güçlü bağlanması ya da önemli olması, küçük olması zayıf bağlanması ya da önemli olmaması anlamına gelmektedir.

Toplama İşlevi: Toplama işlevi v_i , sinirde her bir ağırlığın ait olduğu girişlerle çarpımının toplamlarını eşik θ_j değeri ile toplayarak etkinlik işlevine gönderir. Bazı durumlarda toplama işlevi bu kadar basit bir işlem yerine, en az (min), en çok (max), çoğunluk veya birkaç normalleştirme algoritması gibi çok daha karmaşık olabilir.

Aktivasyon Fonksiyonları (Etkinlik İşlevi): Toplama işlevinin sonucu, etkinlik işlevinden $f(v_i)$ geçirilip çıkışa iletilir. Bir etkinlik işlevinin kullanım amacı, zaman söz konusu olduğunda toplama işlevinin çıkışının değişmesine izin vermektir. Etkinlik işlevi, Şekil 4.2' de gösterilen eğrilerin biri şeklinde olabilmektedir. Bu etkinlik işlevlerini matematiksel ifadeleri aşağıda sıralanmıştır (35).

Şekil 4.2.a' da verilen basamak işlevi Eş. 4.3' de olduğu gibi ifade edilir.

$$f(v_i) = \begin{cases} 1 & ; v_i \geq 0 \\ 0 & ; v_i \leq 0 \end{cases} \quad [4.3]$$

Şekil 4.2 (b)' de verilen kesikli doğrusal işlevi Eş. 4.4' de olduğu gibi ifade edilir. Bu işlev doğrusal sigmoid işlev gibi tanımlanır.

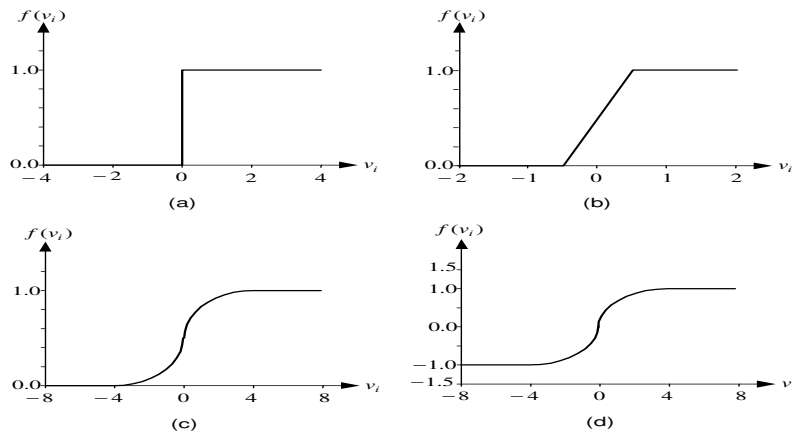
$$f(v_i) = \begin{cases} 1 & ; v_i \geq 0,5 \\ v_i + 0,5 & ; -0,5 \leq v_i \leq 0,5 \\ 0 & ; v_i \leq -0,5 \end{cases} \quad [4.4]$$

Şekil 4.2 (c)' de verilen tek yönlü sigmoid işlevi Eş. 4.5' de olduğu gibi ifade edilir.

$$f(v_i) = \frac{1}{1 + e^{-v_i}} \quad [4.5]$$

Şekil 4.2 (d)' de verilen çift yönlü sigmoid (tanjant) işlevi Eş. 4.6' da olduğu gibi ifade edilir.

$$f(v_i) = \frac{1 - e^{-v_i}}{1 + e^{-v_i}} \quad [4.6]$$



Şekil 4.2. Etkinlik işlevleri a) İşaret işlevi b) Doğrusal işlev
c) Tek yönlü sigmoid işlev d) Çift yönlü sigmoid işlev

Çıkış işlevi: Çıkış $y_i = f(v_i)$, etkinlik işlevi sonucunun dış dünyaya veya diğer sınırlara gönderildiği yerdir. Bir sinirin bir tek çıkışı vardır. Sinirin bu çıkışı, kendinden sonra gelen herhangi bir sayıdaki diğer sınırlara giriş olabilir.

Her bir düğümde bir çıkış işaretine izin verilir. Bu işaret diğer yüzlerce sinir hücrelerinin girişi olabilir. Bu durum biyolojik sinirde olduğu gibidir. Biyolojik sinirde de birçok giriş varken sadece bir çıkış etkinliği vardır. Düğüm çıkışı etkinlik işlevinin sonucuna eşdeğerdir. Fakat bazı ağ yapıları, komşu düğümler arasında yarışma oluşturmak için etkinlik sonuçlarını düzenleyebilir. Böylece yarışmacı girişler hangi düğümün öğrenme ya da uyma işlemine katılacağına karar verilmesinde yardımcı olur.

4.3. YSA' ların Üstünlükleri

YSA' nın klasik sistemlere göre sağladıkları üstünlükler aşağıda açıklanmıştır (36).

Doğrusal olmayan sistemlere uygulanabilirliği: YSA bu özeliğinden dolayı doğadaki hemen hemen her probleme uygulanabilmesini sağlamaktadır.

Öğrenme: YSA' ları diğer yaklaşımlardan ayıran temel bir özelliktir. YSA' ları bir problemi öğrenebilmesi için problemin giriş ve çıkış verilerine karşılık gelen çıkış verilerinin veya sadece verilerinin sağlanması gerekir. Giriş ve çıkış verileri ile tanımlanan öğrenme kümesi yeteri sayıda örnek içermelidir.

Genelleme yapma: Eğitim sürecinin ardından YSA eğitim kümesi dışındaki veriler için de çıkışlar üretebilir. YSA' nın ürettiği çıkışların kabul edilebilir düzeyde olup olmadığı, test verilerinin bilinen sistem çıkışı ile YSA' nın çıkış değerleri karşılaştırılarak varılabilir. Bu uygulama genellemenin testi olarak bilinir.

Adaptasyon: YSA tanımı yada parametreleri değişen probleme veya sisteme uygun çözümler sağlamak için tekrar tekrar eğitilebilir. Eğitim gerçek zamanda gerçekleştirilir.

Veri İşleme: Biyolojik sistemlerde veri dağınık yapıda saklanır. YSA'da ise, ağırlıklar üzerine paralel olarak dağıtılmış şekilde temsil edilmekte, korunmakta ve işlenmektedir. Ağırlıklar herhangi bir anda YSA'nın problemle ilgili olarak sahip olduğu bilgiyi ifade eder.

Hataya ve gürültüye karşı duyarlılık ve tolerans: YSA'nın genelleme yeteneği gerçekleştirilen testlerle sınırlanır. Genelleme, yeterli seviyede ise giriş vektörünü tanımlayan parametrelerden bazılarının verilmemesi veya sağlanmaması durumunda yada ağırlıklarından bazılarının bozulmaması veya kullanılamaz duruma gelmesi halinde de giriş vektörünü karşılayan çıkış verilerinin YSA tarafından kabul edilebilir doğrulukta üretebilmesidir.

Donanım: YSA'lar paralel yapısı nedeniyle büyük ölçekli entegre devre (VLSI) teknolojisi ile gerçekleştirilebilir. Bu özellik, YSA'nın hızlı bilgi işleme yeteneğini ve örnek tanıma, işaret işleme, sistem kimliklendirme ve denetim gibi gerçek zaman uygulamalarında kullanımı artırır.

4.4. YSA'ların Sınıflandırılması

YSA'nın düğümleri ve bağlantıları çok değişik biçimlerde bir araya getirilebilir. Ağlar bu düğüm ve bağlantı mimarilerine göre değişik isimler alırlar. Yapay sinir ağ mimarileri, sınırlar arasındaki bağlantıların yönlerine göre veya ağ içindeki işaretlerin akış yönlerine göre birbirlerinden ayrılmaktadır. Buna göre YSA için, ileri beslemeli (feedforward) ve geri beslemeli (feedback veya recurrent) ağlar olmak üzere iki temel ağ mimarisi vardır.

4.5. YSA'ların Uygulama Alanları

Günümüzde pek çok alanda YSA'nın uygulamalarına rastlamak mümkündür. Yapısal farklılıklarına göre uygulamalarda kullanılan YSA çeşitlilik gösterebilir (37).

Çoğunlukla uygulamalarda karşılaşılan YSA' türlerine göre Çizelge 4.1'de verilmiştir.

4.6. Yapay Sinir Ağı Mimarisi

YSA düğüm veya sinir olarak adlandırılan çok sayıdaki işlem elemanının bir araya gelmesinden oluşur. Psikolog Frank Rosentblatt' ın öğrenme yeteneğine sahip olan bir örnek tanıma aygıtını ortaya çıkarmış olduğu 1958 yılından bu yana, hiyerarşik sinir ağı en çok üzerinde çalışılmış olan bir ağ yapısı olmuştur. Bir hiyerarşik sinir ağı, düğümlerin sıralı bir şekilde art arda bağlanmasından oluşur.

Genellikle ağ yapısı öğrenme algoritmasının seçiminde belirleyicidir. Bu nedenle seçilen ağ yapısı üzerinde kullanılacak öğrenme algoritmasının seçimi, çoğunlukla ağ mimarisine de bağlıdır. Bazen tasarımda öğrenme algoritmasının seçimi veya kurgulanması ağ yapısının tasarımı veya kurgulanması ile paralel veya önce de gerçekleştirilebilir. Bu genellikle basit sınıflandırıcı ağ yapıları (MLP ağları) veya basit öğrenme algoritmaları için geçerlidir.

Çizelge 4.1. YSA türlerine göre uygulama alanları

Uygulama	Yapay Sinir Ağı Mimarisi
Tahmin	Geri yayımlı ağ
	ADALİNE
	Perceptron
	Radyal temelli fonksiyon(RBF) ağı
Model Sınıflandırma	Karşı yayma ağı
	Vektör kuantalamalı öğrenme(LVQ)
	Kohonen ağı
	Geri yayımlı ağ
	Radyal temelli fonksiyon(RBF) ağı
	Boltzman makinası
Perceptron	
Veri İlişkilendirme	İki yönlü ilişkili hafıza (BAM)

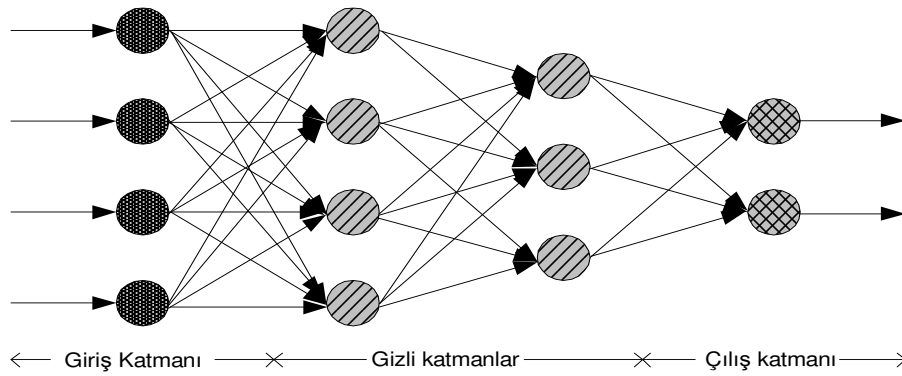
Çizelge 4.1.(Devam) YSA türlerine göre uygulama alanları

	Boltzman makinası
	Hamming ağı
	Hopfield ağı
Sistem kimliklendirme ve kontrol	Perceptron
	Geri yayımlı ağ
	Radyal temelli fonksiyon(RBF) ağı
	Hopfield ağı
	Kohonen ağı
Resim işleme, görüntü	Perceptron
	Geri yayımlı ağ
	Radyal temelli fonksiyon(RBF) ağı
	İki yönlü ilişkili hafıza (BAM)
	Vektör kuantalamalı öğrenme(LVQ)

Bu tür ağların özelliği sahip oldukları basit dinamikleridir. Eğer giriş katmanına bir işaret girerse, bu işaret sinirler arasındaki bağlantılar yoluyla diğer katmana iletilir. Giriş katmanının düğümleri tarafından öteki katmana iletilmeden önce bu işaret üzerinde basit bir işlem yerine getirilir. Bu süreç çıkış katmanına erişinceye kadar tekrar edilir. Ağ içerisindeki düğümler katmanlar halinde yerleştirilir. İlk katmandaki girişe verilen bilgi ağ içinde ileriye doğru yayılır. Her katmandaki düğümlere sadece önceki katmandaki düğümlerden girişlere izin verilir. Bir düğüm kendinden sonraki her hangi bir düğüme bağlanabilirken kendisine asla bağlanmaz. Son katmandaki işaretler ağın çıkışıdır. Geri beslemeli veya tekrarlanan ağlarda en azından bir düğümün geriye yayıldığı bir dönüş bağlantısı vardır. Tekrarlanan ağlar da tamamen veya parçalı olarak geri besleme yollarına sahiptirler. Bu tür ağların tasarımları ve davranışları oldukça karmaşıktır.

Sinir ağlarında istenen sonucun elde edilmesi için ağın uyarlanabilir olması gerekir. Bunu sağlamak için uygun değerli ağırlıklar ve doğru bağlantılar seçilmelidir. Ağ bu şartları karşılayabilmek için sistemin davranışlarını öğrenmeli ya da kendi kendini örgütlemelidir. Öğrenme kalıcı yenilenmeler için gözlemlenebilir veya eğitim

faaliyetlerinden çıkan sonuçlar olarak tanımlanabilir. Öğrenme YSA' nın ayrılmaz bir parçasıdır. Öğrenme; giriş değerlerine veya bu girişlerin çıkışlarına bağlı olarak ağın bağlantı ağırlıklarını değiştiren veya ayarlayan öğrenme kuralı ile gerçekleştirilir. Ara (gizli) katmanları olmayan, sadece giriş ve çıkış katmanı olan ağlar, karmaşık işlevleri hesaplama yeteneğinden yoksundur. Bu nedenle karmaşık hesaplamalar için oluşturulan ağlarda en az bir ara katman olmalıdır. Burada her bir sinir doğrusal olmayan bir etkinlik işlevini hesaplayabilmektedir. Şekil 4.3'de iki gizli katmana sahip dört katmanlı bir YSA görülmektedir.



Şekil 4.3. Çok katmanlı bir YSA

Bu tür ağlar, giriş katmanı, bir veya birden çok gizli katman ve çıkış katmanından oluşmaktadır. Bir katmandaki her sinir, bir sonraki katmanın bütün sinirleri ile bağlantılıdır. Aynı katmandaki sinirler arasında veya geri-besleme şeklinde bağlantıları yoktur.

Giriş katmanından alınan bilgiler, giriş katmanı ve gizli katman arasında bulunan bağlantı ağırlıkları ile çarpılıp gizli katmana iletilmektedir. Gizli katmandaki sinirlere gelen girişler toplanarak aynı şekilde gizli katman ile çıkış katmanı arasındaki bağlantı ağırlıkları ile çarpılarak çıkış katmanına iletilir. Çıkış katmanındaki sinirler de kendisine gelen bu girişleri toplayarak buna uygun bir çıkış üretirler. Burada giriş katmanından çıkış katmanına doğru, gizli katmanlar üzerinden tek yönlü bir iletişim mevcuttur.

4.7. YSA' larda Öğrenme

YSA' da bilgi, ağdaki bağlantıların ağırlıklarında depolanır. Bir ağda öğrenme kısaca, istenen bir işlevi yerine getirecek şekilde ağırlıkların ayarlanması sürecidir. YSA' da öğrenme, sınırlar arasındaki ağırlıkların değiştirilmesi ile gerçekleşmektedir. Buna göre sınırlar arası bağlantılar üzerindeki ağırlıkların, belirli bir yöntem (öğrenme kuralları) uyarınca dinamik olarak değiştirilebilen ağlar eğitilebilir. Eğitilebilen yani öğrenebilen ağlar, yeni şekilleri tanıyabilir (şekil tanıma) veya verilen bir girişin hangi sınıfa ait olduğuna karar verebilir (sınıflandırma). YSA' da öğrenme düğümler arasındaki ağırlıkların, düğümlerdeki etkinlik veya aktarım işlevlerinin değişkenlerinin ayarlanmasıyla yapılmaktadır.

Ağların eğitimi için kullanılan öğrenme kuralları genellikle danışmanlı öğrenme (supervised learning), danışmansız öğrenme (unsupervised learning) ve pekiştirerek öğrenme (reinforcement learning) olmak üzere üç öğrenme yöntemi başlığı altında toplanabilir.

4.7.1. Danışmanlı öğrenme

YSA' da gerçek çıkış istenen çıkışla kıyaslanır. Rasgele değişen ağırlıklar ağ tarafından öyle ayarlanır ki, bir sonraki döngüde gerçek çıkış ile istenen çıkış arasında daha yakın karşılaştırma üretebilsin. Öğrenme yöntemi, bütün işleme elemanlarının anlık hatalarını en aza indirmeye çalışır. Bu hata azaltma işlemi, kabul edilebilir doğruluğa ulaşana kadar ağırlıklar devamlı olarak derlenir.

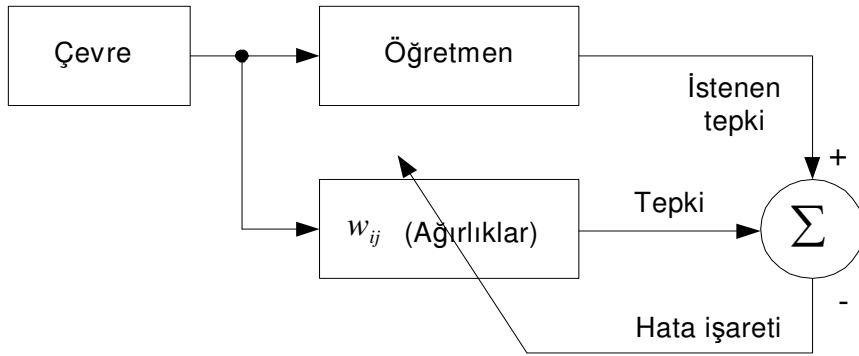
Danışmanlı öğrenmede, YSA kullanılmadan önce eğitilmelidir. Eğitme işlemi, sınır ağına giriş ve çıkış bilgileri sunmaktan oluşur. Bu bilgiler genellikle eğitme kümesi olarak tanımlanır. Yani, her bir giriş kümesi için uygun çıkış kümesi ağa sağlanmalıdır.

Birçok uygulamada, ağı gerçek veriler uygulanmak zorundadır. Bu eğitime safhası uzun zaman alabilir. Sınır ağı, belirli bir sıralamadaki girişler için istenen istatistiksel doğruluğu elde ettiği zaman eğitime işlemi tamamlanmış kabul edilir ve eğitime işlemi bitirilir. Öğrenim aşaması tamamlandıktan sonra ağı kullanılmaya başlandığında, bulunan ağırlıkların değeri sabit olarak alınır ve bir daha değiştirilmezler. Bazı ağı yapılarında ağı çalışırken çok düşük oranda eğitime izin verilir. Bu işlem ağların değişen koşullara uyum sağlamasına yardımcı olur.

Eğer sistemin önemli olan özellikleri ve ilişkileri öğrenmesi gerekiyorsa, o zaman eğitime kümesi, bütün ihtiyaç duyulan bilgileri içermesi gerekir. Eğer ağı sadece bir örnekle eğitilirse, bir olay için çok hassas olan bütün ağırlıklar kümesi, bir sonraki olayda yeterli çözüm vermez. Yeni şeyler öğrenme safhasında eski olaylar unutulabilir. Sonuç olarak, sistem gerekli bilgilerle birlikte öğrenmek zorundadır.

Danışmanlı öğrenmede giriş ve çıkış çiftlerinden oluşan eğitim bilgileri vardır. Ağı giriş bilgisine göre ürettiği çıkış değerini, istenen değerle karşılaştırarak ağırlıkların değiştirilmesinde kullanılacak bilgiyi elde eder. Girilen değerle istenen değer arasındaki fark hata değeri olarak önceden belirlenen değerden küçük oluncaya kadar eğitime devam edilir. Hata değeri istenen değer altına düştüğünde tüm ağırlıklar sabitlenerek eğitim işlemi sonlandırılır. Eğitim işlemi sırasında her bir eğitim bilgisi çifti için oluşan hata değerine göre ağırlıkların değiştirilmesine 'örüntü kipi' öğrenme, tüm eğitim kümesi için hataların toplanarak toplam hata değerine göre ağırlıkların değiştirilmesine ise 'küme kipi' öğrenme denilmektedir.

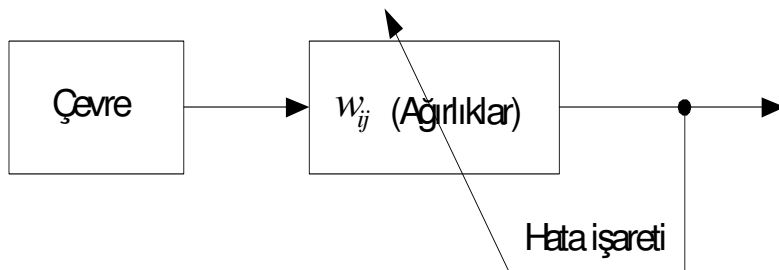
Danışmanlı öğrenmeye çok katmanlı perceptron (multilayer perceptron), geriye yayılım (backpropagation), delta kuralı, Widrow-Hoff veya en küçük karelerin ortalaması (least mean square) ve uyarlanabilir doğrusal eleman anlamına gelen ADALINE örnek olarak verilebilir. Danışmanlı öğrenme Şekil 4.4' de gösterilmiştir.



Şekil 4.4. Danışmanlı öğrenme

4.7.2. Danışmansız öğrenme

Danışmansız öğrenmede bir danışman veya öğretmen, sinir ağın girişin hangi veri parçası sınıfına ait olduğunu veya ağın nerede iyi sonuç vereceğini söylemez. Ağ veriyi üyeleri birbirinin benzeri olan öbeklere yol gösterilmeksizin ayırır. Danışmansız olarak eğitilebilen ağlar istenen ya da hedef çıkış olmadan giriş bilgilerinin özelliklerine göre ağırlık değerlerini ayarlar. Danışmansız öğrenme danışmanlı öğrenmeye göre çok daha hızlıdır. Ayrıca matematik algoritmaları da daha basittir. Danışmansız öğrenmeye yarışmacı öğrenme (competitive learning), Kohonen'in Özörgütlemeli harita ağları (self-organizing), Hebbian öğrenme, Grossberg öğrenme gibi öğrenme kuralları örnek olarak verilebilir. Şekil 4.5'de danışmansız öğrenme ana hatlarıyla gösterilmiştir.



Şekil 4.5. Danışmansız öğrenme

4.7.3. Pekiştirerek öğrenme

Yaparak öğrenme veya pekiştirerek öğrenme yönteminde ağa doğrudan gerçek ağ çıkışı verilmez, ağ çıkışının iyi veya kötü olarak değerlendirmesini yapar. Performans bilgisi genellikle ikili sayıdır ve denetim hareketleri sırasının başarısını gösterir.

4.8. Öğrenme Oranının Ağ Üzerindeki Etkisi

Öğrenme oranı ağ performansı üzerinde önemli bir etkiye sahiptir. Küçük öğrenme oranı değerleri için eğitime işlemi uzun zaman alırken bu değerin büyütülmesi ile eğitime işlemi daha kısa zamanda gerçekleşmektedir. Öğrenme oranının artırılması durumunda öğrenme için gerekli adım sayısında azalma meydana gelmektedir. Öğrenme oranının artırılması ağın toplam hatası üzerinde bir iyileştirme meydana getirmektedir. Fakat öğrenme oranına çok büyük değerlerin verilmesi durumunda pek bir yakınsama olayı meydana gelmemektedir.

4.9. Momentum Teriminin Ağ Üzerindeki Etkisi

Hesaplamalara momentum teriminin ilave edilmesi ağ performansı üzerinde etkili olduğu gözlenmiştir. Momentum teriminin hesaplamaya katılması adım sayısında ve toplam ağ hatasında bir düşüş meydana getirmektedir. Momentum katsayısı yüksek alındığında ağdaki toplam hatanın sifıra doğru daha fazla bir eğilimle yaklaştığı görülmektedir.

4.10. Gizli Katman Sinir Sayısının Ağ Üzerindeki Etkisi

Gizli katmandaki sinir sayılarının artırılması simülasyon sırasında hem hafıza hem de CPU'nun yükünü artırmaktadır. Fakat öğrenme işlemi daha hassas tamamlanmaktadır. Gizli katman sinir sayısının az miktarda alınması ağın hatırlama yeteneğini olumsuz yönde etkilemektedir. Bu yüzden bu çalışmada ara katman sinir

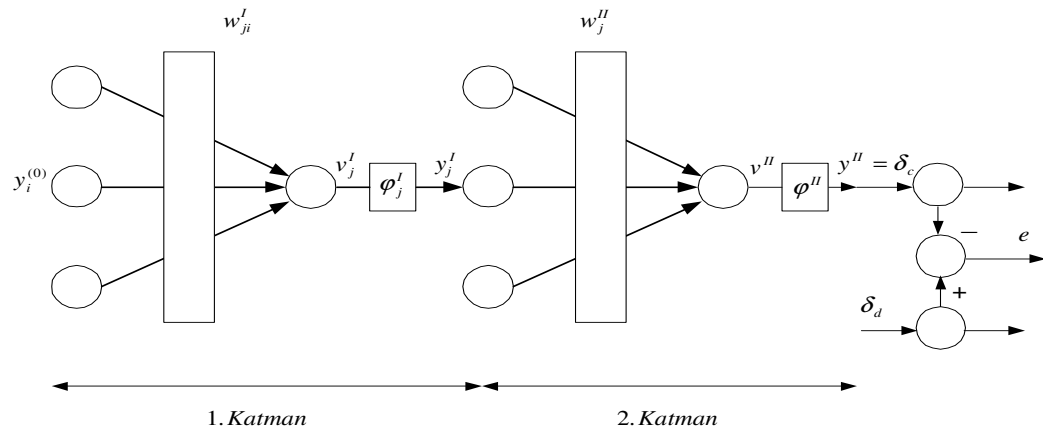
sayısı, giriş katmanındaki ve çıkış katmanındaki sinir sayıları da gözetilerek deneme yanılma yöntemi ile belirlenmiştir.

4.11. Hata Farkı Değişkeninin Ağ Üzerindeki Etkisi

Hata farkı değişken değeri ya da tolerans, sonuçta istenilen toplam çıkış hatasındaki hassasiyetle ilgilidir. Küçük tolerans değerleri ağdaki toplam çıkış hatasını daha fazla sifira doğru yaklaştırarak bağlantı ağırlıklarındaki hassasiyeti de aynı oranda artırmaktadır. Aynı zamanda tolerans değerinin küçülmesi eğitim süresini ve adım sayısını arttırmaktadır.

4.12. Danışmanlı Geri Yayılımlı Bir YSA Modeli

Şekil 4.6' da tek çıkışlı iki-katmanlı geri yayılımlı bir YSA modeli görülmektedir. Geri yayılım tekniği, sinir ağlarının ağırlıklarının ayarlanmasında çok güçlü bir metottur (108). Tek çıkışlı iki-katmanlı sinir ağlarındaki ağırlıkların aşağıdaki eşitliklere göre her adımında güncelleştirilir.



Şekil 4.6. Tek çıkışlı iki-katmanlı geri yayılımlı YSA

I. katmandaki ağırlık değerinin bulunması Eş 4.7' de verilmiştir.

$$w_{ji}^I(k) = w_{ji}^I(k-1) + \Delta w_{ji}^I(k) \quad [4.7]$$

II. katmandaki ağırlık değerinin bulunması Eş 4.8' de verilmiştir.

$$w_j^II(k) = w_j^II(k-1) + \Delta w_j^II(k) \quad [4.8]$$

Eş. 4.7 ve Eş. 4.8' deki $\Delta w_{ji}^I(k)$ ve $\Delta w_j^II(k)$ ağırlık değişimlerinin bulunması için gerekli hesaplamalar aşağıdaki gibi gerçekleştirilir.

Hata $e = e(+)$ istenen çıkış değeri $\delta_d = \delta_d(t)$ ile gerçek çıkış değeri $\delta_c = \delta_c(t)$ arasındaki farktır. Buradan hata Eş. 4.9' da olduğu gibi hesaplanır.

$$e(t) = \delta_d(t) - \delta_c(t) \quad [4.9]$$

Performans veya performans indeksi (E), hatanın (e) karesinin yarısı olarak Eş. 4.10' daki gibi ifade edilir.

$$E = \frac{1}{2}e^2 = \frac{1}{2}(\delta_d - \delta_c)^2 \quad [4.10]$$

Hatanın geriye yayılım algoritmasında, performans indeksinin (E) minimize edilmesine çalışılır. Şekil 4.6' da iki-katmanlı sinir ağının genel gösterimi verilmektedir. $y^II = \delta_c$ ifadesi ikinci katmanın çıkışıdır yani sinir ağının çıkışıdır. y_j^I ise birinci katmandaki sinirin çıkışıdır. Bu çıkış ikinci katmanın girişidir. Çıkış katmanından itibaren performans indeksini Eş. 4.11' deki ifade edilir.

$$\frac{\partial E}{\partial w_j^II} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y^II} \frac{\partial y^II}{\partial v^II} \frac{\partial v^II}{\partial w_j^II} \quad [4.11]$$

$y_j^I = \frac{\partial v^{II}}{\partial w_j^{II}}$ ifadesi Eş 4.11' de yerine konularak Eş 4.12 elde edilir.

$$\frac{\partial E}{\partial w_j^{II}} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y^{II}} \frac{\partial y^{II}}{\partial v^{II}} \cdot y_j^I \quad [4.12]$$

Eş. 4.12' nin sağ tarafındaki türevli kısmı δ^{II} gibi bir değişkenle ifade edilirse Eş. 4.13 elde edilir.

$$\delta^{II} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y^{II}} \frac{\partial y^{II}}{\partial v^{II}} \quad [4.13]$$

Buradan Eş. 4.12 yeniden düzenlenirse Eş 4.14 elde edilir.

$$\frac{\partial E}{\partial w_j^{II}} = \delta^{II} y_j^I \quad [4.14]$$

Eş 4.13' deki her bir türevin bileşeninin açılımı aşağıdaki gibi yapılır. Eş. 4.15 performans indeksi ile hata arasındaki türevin sonucudur.

$$\frac{\partial E}{\partial e} = e \quad [4.15]$$

Eş. 4.16' da hata ile II. katmanın çıkışı (ağın gerçek çıkışı) arasındaki türevin sonucu verilmiştir.

$$\frac{\partial e}{\partial y^{II}} = -1 \quad [4.16]$$

Elde edilen bu değerler Eş. 4.13' de yerine konulursa Eş. 4.17 elde edilir.

$$\delta'' = -e\phi''(v'') \quad [4.17]$$

$$\phi''(v'') = \frac{\partial y''}{\partial v''} \quad [4.18]$$

II. katmanın ağırlık değişimi ($\Delta w_j''$), öğrenme katsayısı (λ'') ile orantılıdır.

$$\Delta w_j''(k) = \lambda'' \frac{\partial E}{\partial w_j''}(k) \quad [4.19]$$

$$\Delta w_j''(k) = \lambda'' \delta''(k) y_j^I(k) \quad [4.20]$$

Eş. 4.20' de elde edilen ağırlık değişimi Eş. 4.8' de yerine konularak II. Katmanın ağırlık değerleri yenilenir.

II. katman için yapılan tüm ağırlık ayarlanmaları I. Katman içinde yapılacaktır.

$$\frac{\partial E}{\partial w_{ji}^I} = \frac{\partial E}{\partial y_j^I} \frac{\partial y_j^I}{\partial v_j^I} \frac{\partial v_j^I}{\partial w_{ji}^I} \quad [4.21]$$

Eş. 4.21' in en sağ tarafındaki türevli kısım (p_i) gibi bir değişkenle ifade edilirse Eş. 4.22 elde edilir.

$$p_i = \frac{\partial v_j^I}{\partial w_{ji}^I} \quad [4.22]$$

Eş. 4.22, Eş. 4.21' de yerine konulursa Eş. 4.23 elde edilir.

$$\frac{\partial E}{\partial w_{ji}^I} = \frac{\partial E}{\partial y_j^I} \frac{\partial y_j^I}{\partial v_j^I} p_i \quad [4.23]$$

Eş. 4.23' ün sağ tarafındaki türevli kısımlar (δ_j^I) gibi bir değişkenle ifade edilirse Eş. 4.24 elde edilir.

$$\delta_j^I = \frac{\partial E}{\partial y_j^I} \frac{\partial y_j^I}{\partial v_j^I} \quad [4.24]$$

Eş. 4.24 tekrar yazılacak olursa Eş 4.25 elde edilir.

$$\frac{\partial E}{\partial w_{ji}^I} = \delta_j^I p_i \quad [4.25]$$

I. katman içinde delta kuralı (Δ) uygulanırsa ağırlık değişim değerleri,

$$\Delta w_{ji}^I = \lambda^I \frac{\partial E}{\partial w_{ji}^I} \quad [4.26]$$

veya

$$\Delta w_{ji}^I = \lambda^I \delta_j^I p_i \quad [4.27]$$

Eş. 4.27, Eş 4.20' de olduğu II. Katman için yapılan ağırlık ayarlamasına benzerdir. Burada λ^I değişkeni I. katman için kullanılan öğrenme sabitidir. Eş 4.28' deki türevli kısım δ_j^I gibi bir değişkenle gösterilirse,

$$\delta_j^I = \frac{\partial E}{\partial y_j^I} \frac{\partial y_j^I}{\partial v_j^I} = [\delta^II \ w_j^{II}] [\phi'^I (v_j^I)] \quad [4.28]$$

Burada Eş 4.28' in sağ tarafındaki ifadeler Eş 4.29 ve Eş. 4.30' da verilmiştir.

$$\frac{\partial y_j^I}{\partial v_j^I} = \varphi'^I(v_j^I) \quad [4.29]$$

$$\frac{\partial E}{\partial y_j^I} = \delta^II w_j^II \quad [4.30]$$

$$\frac{\partial E}{\partial y_j^I} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y^II} \frac{\partial y^II}{\partial v^II} \frac{\partial v^II}{\partial y_j^I} = \delta^II w_j^II \quad [4.31]$$

Burada Eş. 4.31' in sağ tarafındaki ifadeler Eş 4.32 ve Eş. 4.33' de verilmiştir

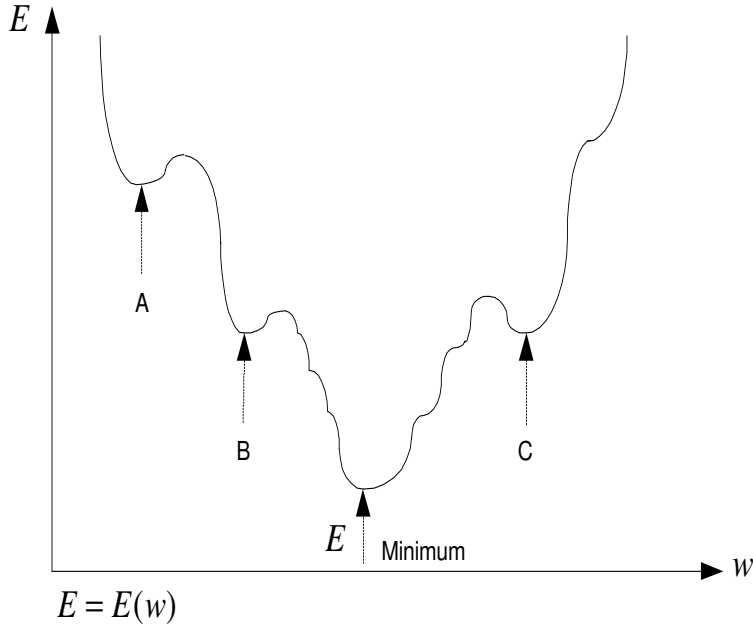
$$\delta^II = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y^II} \frac{\partial y^II}{\partial v^II} \quad [4.32]$$

$$w_j^II = \frac{\partial v^II}{\partial y_j^I} \quad [4.33]$$

Her bir katmandaki λ^I ve λ^II öğrenme katsayıları değişkendir. Ayrıca yerel minimum noktasını yakalamak içinde momentum katsayısı μ değişkeni kullanılır. Bu değişken $0 < \mu < 1$ aralığında seçilmelidir. Her bir katmanın ağırlık ayarlamaları için momentum katsayısı Eş. 4.34 ve Eş 4.35' deki gibi kullanılır.

$$w_{ji}^I(k+1) = w_{ji}^I(k) + \mu^I [w_{ji}^I(k) - w_{ji}^I(k-1)] \quad [4.34]$$

$$w_j^II(k+1) = w_j^II(k) + \mu^II [w_j^II(k) - w_j^II(k-1)] \quad [4.35]$$



Şekil 4.7. Yerel minimum noktası

Ağırlık vektörü ile performans indeksi arasındaki değişim Şekil 4.7’ de gösterilmektedir. Şekilde görüldüğü gibi değişik yerel minimum noktalar (A, B, C) vardır. Öğrenme ve momentum katsayıları öyle ayarlanmalıdır ki gerçek minimum noktası elde edilebilsin.

4.13. Arapça Karakter Tanıma Problemi İçin Kullanılan Yapay Sinir Ağlarının Karşılaştırılması

Bu çalışmada, araştırmacılar 28 izole Arapça karakter içeren bir veri tabanını, 5 değişik yapay sinir ağını eğitmede ve test etmede kullanmışlardır. Yapay sinir ağını eğitmede kullanılan giriş örüntüleri, 32x32 piksel boyunda siyah-beyaz görüntüler şeklinde, üzerinde hiçbir ön işleme ve özellik çıkarımı yapılmadan ham veri olarak verilmiştir. 1024 birim genişliğindeki giriş katmanı, karakterleri belirleyen özellikleri kaybetmemek için seçilmiş optimum değerdir. Daha küçük boyutlarda, karakterlerin şekli bozulmakta ve gürültü karışmaktadır (4).

4.13.1. Karşılaştırma amacıyla kullanılan yapay sinir ağları

Arapça karakterlerini sınıflandırma performanslarının karşılaştırıldığı yapay sinir ağları şunlardır:

- Çok Katmanlı Geri Beslemeli Ağ
- Öğreticisiz Eğitimli ART1 (Adaptif Rezonans Teorisi) Ağı
- Öğreticili ARTMAP Ağı
- Learning Vector Quantiser (LVQ) Ağı
- Kohonen Ağı

Yukarıdaki ağlar üzerinde hiçbir önışleme ve özellik çıkarımı yapılmadan 28 değişik Arapça harfinin sınıflandırılması için kullanılmıştır. İlk dört yapay sinir ağı, SNNS (Zell et al.1993) adlı yazılım paketi ile sonucusu MATLAB (Demuth and Beale, 1992) ile gerçekleştirilmiştir.

A- Çok katmanlı algılayıcı

Çok katmanlı algılayıcı, tek katmanlı algılayıcının bir uzantısıdır. Algılayıcı sinir hücresi, n boyutlu x giriş vektörünün iki değişik sınıftan hangisine ait olduğunu bulmak için kendisinde sonlanan ağırlıkların toplamından belli bir eşik (threshold) değerini hesaplar. Oluşan y değeri daha sonra, algılayıcının çıkış fonksiyonu $f(y)$ ' ye geçirilir. Bunun için, çıktı değeri olarak -1 veya $+1$ üreten bipolar lineer olmayan aktivasyon fonksiyonu kullanılmıştır.

Çok katmanlı algılayıcının ilk katmanındaki her bir sinir hücresi, bir hiper düzlem (iki boyutlu uzay için bir çizgi) ile ayrılmış iki karar bölgesi oluşturur. İkinci katmandaki her sinir hücresi, ilk katmandaki hücrelerin çıkışlarını alarak, vektör uzayında bir hiper küp oluştururlar. Bunu, giriş değerlerini mantıksal olarak VE işleminden geçirerek, ilk katmandaki her hiperdüzlemin kesişim kümesini alarak yaparlar.

Çok katmanlı algılayıcının gerçekleştirimi:

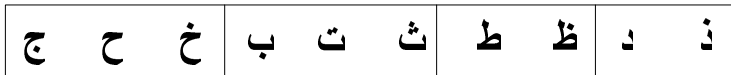
Çok katmanlı algılayıcı SNNS yazılım paketinin 'BIGNET' özelliği kullanılarak gerçekleştirilmiştir. Bu özellik sayesinde, düzgün yapıdaki büyük yapay sinir ağları kısa bir sürede oluşturulmaktadır.

Ağ, momentumlu geri yayımlı eğitim algoritmasını kullananan geri beslemeli yapıda olduğundan, momentum değeri şu anki ağırlık değişimine bir önceki ağırlık değişiminin belli oranda eklenmesini sağlamaktadır. Bu katsayı değeri, geri yayılma hata yüzeyindeki yalancı (false) minimum noktasından uzaklaşmak için ağırlıklara bir ivme kazandırır. (Rumelhart and McClelland, 1986)

Bu çalışmada iki değişik mimariye sahip çok katmanlı algılayıcı kullanılmıştır. İki yapıda da giriş ve çıkış hücre sayısı sırasıyla 1024 ve 28 olmak üzere aynıdır. Birinci yapıda 84 hücreli tek gizli katman ikinci yapıda ise 112 ve 56 hücreli olmak üzere iki gizli katman kullanılmıştır.

Çok katmanlı algılayıcı için elde edilen sonuçlar:

Çok katmanlı algılayıcı, 30 iterasyon sonucu eğitime setindeki 1435 görüntü için oldukça iyi bir yakınsama sağlamıştır. Standart geri yayılma ile, ağın yakınsaması 180 iterasyon sonucu ve daha yüksek ortalama kare hatası ile sağlanabilmiştir. Araştırmacılar, bunun nedenini yerel minimum noktası ile açıklamaktadırlar. 2 katmanlı ve 3 katmanlı ağların ikisi de eğitime setindeki görüntüleri %100 doğrulukta sınıflandırmıştır. Test setindeki görüntüler için bu oran %73 olarak belirlenmiştir. Yanlış sınıflandırılan karakterlerin %30' u aynı temel şekle sahip karakterler olarak bulunmuştur. Aşağıda bu özellikteki karakterlerden bazıları görülmektedir.



Şekil 4.8. Aynı temel şekle sahip Arapça karakterlerden bazıları

17 temel karakter şekli için doğru sınıflandırma oranı yaklaşık %80 'dir.

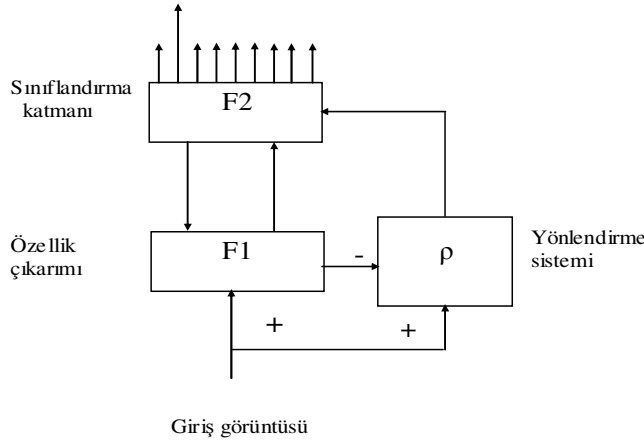
B- Adaptif rezonans teorisi (ART1) ağı

ART1 ve ART2, genel olarak ART olarak adlandırılan ağ yapılarının iki değişik sınıfıdır. Bu ağlar, önceden öğretilen görüntüleri sınıflandırma ve bu görüntüler vasıtasıyla oluşan sınıflardan hiçbirine ait olmayan yeni görüntüler için yeni sınıflar oluşturabilme özelliğine sahiptirler. ART2 ağları örneksel (Analog) giriş değerleri, ART1 ise ikili (binary) giriş değerleri içeren görüntüleri sınıflandırır. Bu çalışmada siyah-beyaz karakter matrisleri kullanıldığından ART1 ağı kullanılmıştır.

ART1 ağ yapısı iki katman içerir. F1 katmanı özellik çıkarımı işlemi yaparken, F2 katmanı sınıflandırmayı gerçekleştirir. F2 katmanındaki her hücre F1 katmanı tarafından çıkarılan özellik değerleri ile ağırlıkların çarpımlarının toplamını alarak, belli bir anda sadece kendisinin aktif olmasını sağlamak üzere diğer hücrelerle yarışır. Yarışı kazanan hücre, önceden öğrendiği sınıfa ait kritik özellikler kümesini geri besleme değeri olarak F1 katmanına verir. Bu pozitif geri besleme değeri, eğer giriş vektörü de benzer özellikleri içeriyorsa, F1 katmanında 'rezonans' etkisine neden olur.

Üretilen rezonans miktarı, vigilance (ρ) denilen bir parametre ile karşılaştırılır. Eğer F1 katmanındaki rezonans yeterli miktarda ise, giriş vektörü F2 katmanındaki yarıştı kazanan hücrenin belirttiği sınıfa dahil demektir. Bunun tersi olursa yarıştı kazanan hücrenin ödülü elinden alınarak, yeni bir F2 katman hücresi aktif hale getirilir.

Eğer hiçbir F2 hücresi giriş vektörü özelliklerini tam olarak temsil edemezse, yani o ana kadar öğrenilen sınıflardan hiçbirine ait değilse, önceden kullanılmayan F2 katmanı hücresi, bu yeni özellikleri tanımak üzere eğitilir ve yeni bir sınıf oluşturur. F2 katmanının tüm hücreleri kullanılırsa, ağın toplam kapasitesine ulaşılmış demektir ve artık yeni bir sınıf eklenmez (Carpenter and Grossberg, 1987-1988)



Şekil 4.9. ART1 Ağının genel görünümü

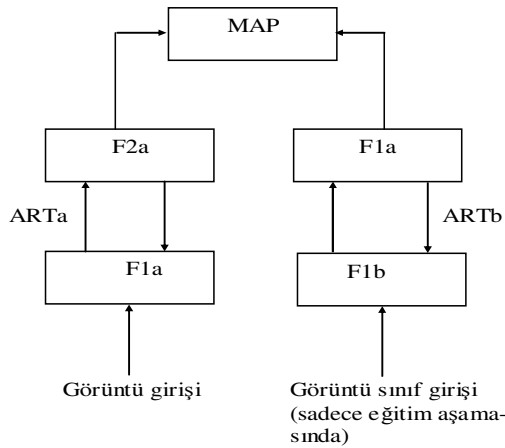
ART1 ağının gerçekleştirimi:

SNNS Yazılım paketinde, bir ART1 ağının gerçekleştirimi için gerekli olan, F1 ve F2 katmanı boyları, ağırlıklara ilk değerlerin verilmesi, vigilance parametresinin ve öğrenme fonksiyonu seçimi işlemleri bulunmaktadır. F1 katmanı, giriş matrisi boyuna (32x32) eşit sayıda hücre içerirken, F2 katmanı her arapça harf için bir tane olmak üzere 28 hücre içermektedir. ART1 ağı, F2 katmanının hücre sayısını eğitime işlemi sırasında kendi kendine ayarlamasına rağmen, bu sayının eğitmenin başında belirlenmesi algoritmanın etkinliğini artıracaktır. Her sınıf için bir sinir hücresi atanması, aynı sınıfa ait karakterlerin benzer özellikler içermesini, farklı sınıftaki karakterlerin ise tamamen farklı özelliklerde olduğunun baştan belirlenmesi demektir. Böylece, her bir sınıfa ait vektörler çok iyi gruplanarak, sınıf sınırlarının kesişmemesi sağlanmış olur. Tüm 28 çıkış hücreleri tespit edilinceye kadar, her eğitime adımında vigilance parametresinde açık bir artış görülecek şekilde, birden çok eğitime yapılmıştır.

ARTMAP ağları:

ARTMAP ağları, ARTa ve ARTb olmak üzere iki ART1 modeli ile eşleme (map) alanı olarak belirlenmiş bir katman içerir. (Şekil 4.10.) Eşleme alanı, R^n kümesindeki

giriş vektörlerini, R_m kümesindeki çıkış vektörleri ile ilişkilendirme işlemini öğreticili eğitim aşamasında yerine getirir. Eğitim vektörü çiftleri (sınıflandırılacak giriş vektörü ve beklenen çıkış değerleri) F1a ve F1b katmanlarına uygulanmıştır. Test aşamasında ise, önceden verilmeyen bir giriş vektörü F1a' ya uygulanırken, F1b' ye buna karşılık gelen vektör uygulanmaz. Bu giriş vektörü, daha sonra önceden öğrenilen b çıkışlarından biriyle eşleştirilir. Her bir sınıflandırma hatasında, vigilance parametresi otomatik olarak artırılarak öğrenme sağlanır. Bu işlem, doğru sınıflandırma sağlanıncaya kadar devam eder. Vigilance parametresi, her vektör girişinden önce sıfırlanarak, yeni bir sınıf oluşturulmadan önce mevcut sınıflandırmalar arasında bir seçim yapmaya zorlar. (Carpenter at al, 1991)



Şekil 4.10. ARTMAP Ağı'nın genel görünümü

ARTMAP ağı'nın gerçekleştirimi:

ARTMAP ağı, SNNS paketinin BIGNET özelliği kullanılarak gerçekleştirilmiştir. Katman boyları F1a için 1024 hücre, F2a için 84 hücre, F1b için 28 hücre ve F2b için 28 hücre şeklindedir. ARTMAP ağı için gerekli ilk ağırlık değerleri verme, öğrenme ve günleme fonksiyonu seçimi yapılmıştır. Ayrıca, ρ_a , ρ_b ve ρ vigilance parametreleri için, sırasıyla 0.3, 1.0, ve 1.0 değerleri kullanılmıştır.

ART1 ve ARTMAP ağı için elde edilen sonuçlar:

ART1 ağıının performansı oldukça kötü çıkmıştır. Düşük vigilance değerlerin de bile, ciddi sınıflandırma hataları oluşmuştur. Farklı sınıflara ait karakterler aynı sınıfa aitmiş gibi görünmüş ya da bir karakter değişik test adımlarında birbirinden farklı sınıflara atanmıştır. Bunun nedeni yazma şeklindeki değişim sonucu biribiri arasında önemli özellik farkları olan karakter sınıflarının ortaya çıkmasıdır. Ortalama bir performans sağlayan uygun bir vigilance değeri bulmak mümkün olmamıştır.

ARTMAP ağıının işlem hızı yönünden de performansı oldukça kötü bulunmuştur. Ağıın yapısının çok büyük olması (1.7 milyon gösterge) nedeniyle, çok kullanıcı bir DEC istasyonunda 1435 karakter için bir eğitime iterasyonu birkaç gün almıştır. Ölçme zamanının çok uzun olması nedeniyle, bu ağıın sınıflandırma performansını tespit etmek mümkün olmamıştır.

C- Learning vector quantiser (LVQ) ağıları

LVQ ağıları, vektör çiftlerinin eğitimi için ilk vektör konum referanslarını belirlemek için öğreticisiz eğitime yöntemini kullanır. Bunu yaparken belirli bir uzaklık ölçütü (genellikle, Öklid) kullanır. Bu eğitime, daha sonra öğreticili eğitime işlemi ile devam eder. Bu aşamada karar bölgesi sınırları, bu referans vektörlerini giriş vektörüne doğru yaklaştırmak (doğru sınıflandırma durumunda) veya uzaklaştırmak (hatalı-sınıflandırma durumunda) suretiyle ayarlanır. Her görüntü sınıfı, birden fazla referans vektörüne atanarak, vektör uzayında giriş vektörü sınıfları tek bir öbek oluşturmayacak şekilde kompleks ve ayrık karar sınıfları oluşturulur. Her sınıf için referans vektör sayısı genellikle eğitime işleminden önce belirlenir, fakat dinamik olarak da atanabilir. (Zell et al, 1993)

LVQ ağıının gerçekleştirimi:

SNNS yazılım paketi, Dinamik LVQ (Zell et al,1993) adıyla tek tip LVQ ağıının gerçekleştirimine izin vermektedir. Bu ismin verilmesinin sebebi, sınıf içi hücrelerin (aynı-sınıfla ilişkili hücreler), eğitime algoritması yardımıyla dinamik olarak

atanmasıdır. Bunun sonucu olarak, iyi gruplanmış sınıf vektörleri için sadece tek bir hücre atanırken, vektör uzayında daha kompleks öbekler oluşturan sınıf vektörleri için daha fazla sayıda hücre atanabilir.

SNNS yardımıyla ağın gerçekleştirimi için, sadece giriş katmanının hücre sayısını belirtmek yeterlidir. Çünkü diğer tüm hücreler eğitime aşamasında dinamik olarak atanmaktadır. LVQ ağı için gerekli öğrenme ve güncleme fonksiyonlarının seçimi ile birlikte aşağıdaki eğitime parametrelerinin seçimi de yapılabilmektedir:

n^+ Bir giriş vektörünün doğru sınıflandırılması durumunda sınıf vektörünün giriş vektörüne doğru kaydırılma adımı değeridir. ($n^+ = 0.03$)

n^- Bir giriş vektörünün yanlış sınıflandırılması sonucu sınıf vektörünün giriş vektöründen uzaklaşma adım değeridir. ($n^- = 0.03$)

iterasyon sayısı :Yeni sınıf referans vektörleri atanmadan önce yapılan toplam iterasyon sayısı 3 olarak belirlenmiştir.

Bu ağ, eğitime setinden toplam 1435 örnek kullanılarak eğitilmiştir.

LVQ ağı için elde edilen sonuçlar:

SNNS yazılım paketi yardımıyla gerçekleştirilen Dinamik LVQ ağı, ortalama 16 iterasyon sonucu 1435 eğitime seti görüntülerini %100 doğrulukta sınıflandırma başarıları göstermiştir. Her sınıf için bir adet olmak üzere toplam 28 sinir hücresi, dinamik olarak her eğitime adımında üretilmiştir. Ortalama sınıflandırma doğruluk oranı 28 değişik karakter sınıfı için %65 iken, 17 temel karakter sınıfı için bu değer %73 olarak bulunmuştur. Yanlış sınıflandırma aşağıdaki karakterler için ortaya çıkmıştır.



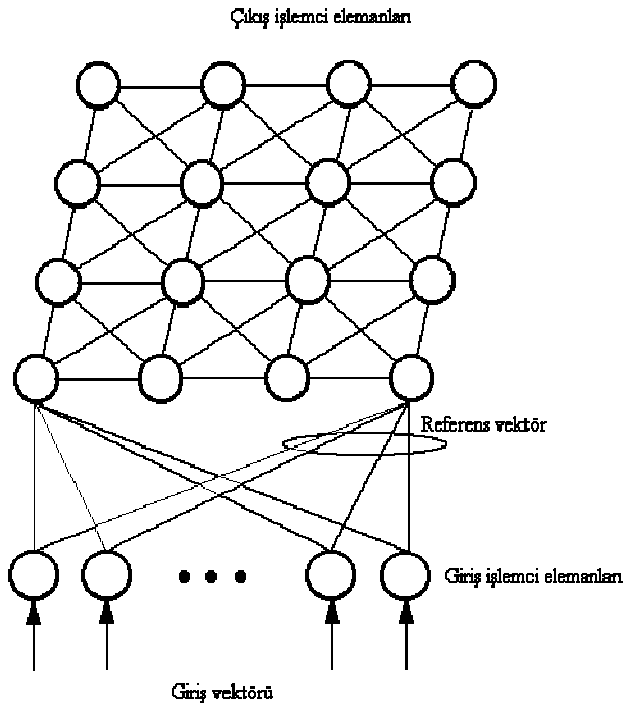
Şekil 4.11. LVQ Ağının yanlış sınıflandırdığı karakterler

D- Kohonen ağları (kendini organize edebilen ağlar)

Kohonen ağları, öğreticisiz eğitilebilen tek katmanlı ağlardır. Sinir hücrelerinin, benzer giriş vektörü gruplarını tanımayı öğrenmeleri, komşu hücrelerinde bu vektörlere tepki göstermelerini sağlayarak gerçekleştirilir. Bir sinir hücresinin bir boyutlu ve iki boyutlu olmak üzere değişik boyutlarda komşuları vardır. Eğitim aşamasında, giriş vektörünün doğru sınıflandırma hakkını kazanmasından sonra sinir hücresi ve komşuları giriş vektörüne bağlı olarak ağırlıklarını ayarlayabilirler. Eğitimin başında ağırlıklar çok küçük random değerlerdir. Eğitim ilerledikçe ağırlık vektörleri giriş vektörlerinin bulunduğu vektör uzayına doğru kaymaya başlar. Böylece, komşuluk derecesi sadece yarışı kazanan sinir hücresini gösterecek şekilde 1'e düşer. Böylece sadece kendisinin ağırlıkları, giriş vektörüne yakın olacak şekilde ayarlanır. Bu tip ağlar, giriş vektör yoğunluğunun yüksek olduğu alanlara daha çok vektör toplanmasına neden olur. Öğrenme işlemi genelde Kohonen öğrenme, algoritmasına dayanır. (Hecht,1990- Demuth and Beale,1992) Bu algoritmanın tek boyutlu biçiminde ağırlıklar aşağıdaki gibi ifade edilir.

$$W_m^{yeni} = W_m^{eski} + k(x - W_m^{eski})Am \quad [4.36]$$

Burada W_m , m. sinir hücresinin ağırlık vektörü, $0 \leq k \leq 1$ öğrenme oranı, x giriş vektörüdür. A, komşuluk alanını tanımlamaktadır ve merkez hücre için 1, komşu hücreler için 1'den büyük katsayılara sahiptir.



Şekil 4.12. Kohonen Ağı

Kohonen ağının gerçekleştirimi:

Özellik eşleyen ağlar olan Kohonen ağları, girişleri vektör formunda aldığı için, eğitime ve test karakter matrislerinin her kolonu bir vektör olacak şekilde ağa verilmiş, giriş vektörleri daha sonra normalize edilmiştir. Bu yüzden giriş katmanı 1024 hücreden oluşmuştur. Her karakter sınıfı için birden fazla olmak üzere toplam 100 sinir hücresi çıkış katmanı için kullanılmıştır. Ağın eğitime işlemi toplam 1000 itirasyon sonucu gerçekleştirilmiştir.

Kohonen ağı için elde edilen sonuçlar:

MATLAB' da gerçekleştirilen Kohonen ağı için doğru sınıflandırma oranı 28 karakter sınıfı için %51, 17 temel karakter sınıfı için %54 olarak bulunmuştur. Araştırmacılar, sonucun çok kötü olmasını, giriş vektörlerinin yanlış öbeklenmesine bağlamaktadırlar. Ayrıca, bu tip ağların ham veri üzerinde ön işleme ve özellik çıkarma işlemleri yapılmayan karakter tanıma problemlerinde yetersiz kalacağı

sonucuna ulaşmaktadırlar.

4.13.2. Genel sonuçlar:

En iyi doğru sınıflandırma oranı çok katmanlı algılayıcı ve LVQ ağları için bulunmuştur. Aşağıdaki iki harf tüm ağlar tarafından yanlış sınıflandırılmıştır.



Şekil 4.13. Tüm ağlar tarafından yanlış sınıflandırılan harfler

Kabul edilebilir oranda sonuç veren üç ağ (Çok Katmanlı Algılayıcı, LVQ ve Kohonen) için doğru sınıflandırma oranları diğer ağlara nazaran daha iyi sonuçlar vermiştir. Bu durum Çizelge Ek-1.1'de verilmiştir.

Çizelge 4.2. En iyi sonucu veren ağların karşılaştırılması

Ağ	28 Sınıf İçin	17 Sınıf İçin
ÇKA	%73	%81
LVQ	%67	%74
Kohonen	%50	%54

Yanlış sınıflandırmanın nedeni, el yazısı karakterlerin yazımı için belli bir şablon belirlenmediğinden dolayı karakterlerin çok fazla deformasyona uğramasıdır. Bu yüzden, deformasyon karşısında da doğru sonuç alınabilecek şekilde önışleme ve özellik çıkarımı gerekmektedir.

Ayrıca aynı temel sınıfa ait karakterler için ortaya çıkan hatalı sınıflandırma gösteriyor ki, harfi belirleyen sadece tek bir noktanın üzerinde durmalı ve özellik çıkarımını bu ayrıntıyı göz önüne alarak yapmalıdır.

Çizelge 4.3. Her biri bir sınıfa karşılık gelen 28 temel Arapça harf

Sınıf No	Harf	Sınıf No	Harf	Sınıf No	Harf	Sınıf No	Harf
1	ا	8	د	15	ض	22	ك
2	ب	9	ذ	16	ط	23	ل
3	ت	10	ر	17	ظ	24	م
4	ث	11	ز	18	ع	25	ن
5	ج	12	س	19	غ	26	ه
6	ح	13	ش	20	ف	27	و
7	خ	14	ص	21	ق	28	ى

Çizelge 4.4. Benzer şekle sahip Arapça harflerin aynı sınıfa atanması ile oluşan temel karakterler

Sınıf No	Harf Grubu	Sınıf No	Harf Grubu	Sınıf No	Harf Grubu
1	ا	8	ظ	15	ه
2	ب ت ث	9	ع غ	16	و
3	ج ح خ	10	ف ق	17	ى
4	ذ	11	ك		
5	ز	12	ل		
6	س ش	13	م		
7	ص ض	14	ن		

4.14. Segmentasyonsuz Kelime Tanıma Yönteminin Arapça İçin Kullanılması

Bu çalışmada araştırmacılar, matbaa harfleriyle yazılmış Arapça kelimeleri segmentasyon yapmadan tanıyabilen bir sistem tasarlamış ve gerçekleştirmişlerdir. Kullanılan yöntem, Arapça bir metni oluşturan harflerin, alt düzeyde parçalarla ifade edilmesine dayanmaktadır. Tanıma aşamasında, alt düzey parçalar

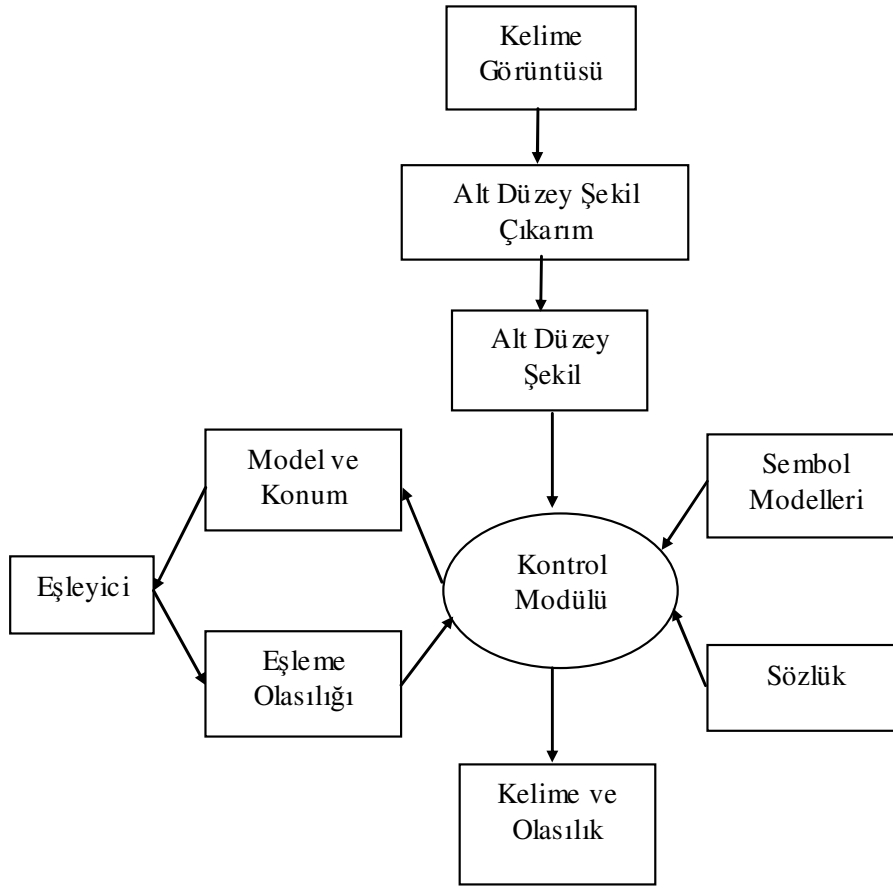
matematiksel morfoloji işlemleri kullanılarak tespit edilmektedir. Tanıma sistemi, elde edilen şekilleri önceden belirlenen karakter modelleriyle eşleştirmeye çalışmaktadır. Eşleşen karakter modelleri uzaysal düzenleme ile elde edilmektedir (Al-Badr and Haralick,1995).

Karakterleri tek tek bulan segmentasyon yöntemine göre, tüm kelime için yapılan işlemin avantajı, tanıma doğruluğunun kelime bazında ele alınabilmesidir. İlk yapılan deneylerde sadece Arapça metin içeren gürültüsüz temiz bir sayfadaki 42.000 kelime için %99.4'lük doğru sınıflandırma oranı elde edilmiştir. Tarayıcıdan alınmış bir sayfadaki kelimeler için bu oran %73 olarak bulunmuştur.

Sistem yeni fontları da otomatik olarak öğrenecek şekilde tasarlanmıştır. Yeni bir fontu sisteme öğretmek için yapılması gereken, fonttaki tüm karakterlerin gürültüsüz bir görüntüsünün girilmesidir.

4.14.1. Tanıma sistemi

Kelime tanıma sistemine giriş olarak verilen bilgi, matbaa harfleriyle yazılmış Arapça bir kelimedir. İlk etapta kelime için bir baseline tespit edilmektedir. Baseline tespiti, yatay olarak görüntünün taranıp, siyah piksellerin yoğun olarak bulunduğu bölgenin belirlenmesi olarak ifade edilebilir. Daha sonra, izole edilen kelime görüntüsü üzerinde önceden tanımlanmış alt düzey şekiller elde edilmektedir. Bu görüntü üzerinde erozyon işlemi şeklinde morfolojik dönüşümler uygulanarak, bu alt düzey şekillerle eşleşen kesimler çıkarılmaktadır. Erozyon işlemi sırasında, alt düzey K şekli, A görüntüsü üzerinde kaydırılmaktadır. Eğer K, A üzerinde bulunursa K'nın gösterdiği x, A'nın K ile erozyonundan bulunan piksel kümesine dahildir.



Şekil 4.14. Tanıma sisteminin genel şeması

4.14.2 Kelime tanıma

Kelimeleri belirlemek için, tanıma sistemi bir grup sembol modeli kullanmaktadır. Her sembol bir karaktere karşılık gelen sabit bir şekildir. Sembol modeli ise, karakteri alt düzey şekillerle ifade eden bir tanımlamadır. Sembol modelleri, tanıma işlemiyle birlikte yapılan eğitime aşamasında üretilmektedir.

Sistemin amacı, sembol modellerinin uzaysal yerleşimlerini bulup, birebir eşleşen ya da çok az değişen karakterlerle eşleştirmektir. Modellerin uzaysal yerleşimi bir semboller dizisi üretmekte ve bu da tanınması istenen kelimenin karşılığı olmaktadır. Sistem, sembol modellerinin en iyi uzaysal yerleşimini bulmak için düzlemsel arama yapmaktadır. Arama uzayındaki her durum, eşleşen sembol modellerinin oluşturduğu

bir grubu belirtmektedir. Bu durumlar, kelime görüntüsünün koordinatlarına bağı olarak belli bölgelerdir ve tanınacak kelimeleri oluşturan sembollerin o anki karşılığıdır.

Olabilecek arama durumlarını azaltmak için kelime yapıları ve sembol modellerinin uzaysal yerleşimi hakkında bazı kabullenmeler yapılmıştır. Kelime yapısı üzerindeki kabullenmeler:

a) Font türü ve büyüklüğü kelime içinde değişmemektedir.

b) Kelime içindeki komşu harfler yazım kurallarına uyumludur (Mesela, bir kelimenin sonunda, başta bulunması gereken bir harf şekli olmamalıdır).

c) Bir kelime, başta bulunabilecek bir harf şekliyle başlayıp sonda bulunabilecek bir harf şekliyle bitmelidir.

Arama olasılıklarını azaltmak için ayrıca Arapça bir sözlük kullanılmakta ve sadece sözcükte bulunan harfler tanınmaktadır. Sembol modellerinin uzaysal yerleşimi için getirilen kısıtlama ise, her sembolün baseline'ının kelime baseline'ı ile çakışma zorunluluğudur. Ayrıca sembolleri çevreleyen kutular (bounding box) birbirlerinin üzerine gelmeyecek şekilde olmalı, aralarında sadece küçük bir boşluk olmalı ve kelimeyi çevreleyen kutucuğun tamamen içinde yer alacak büyüklükte olmalıdır.

Kelime tanıma sistemi, kelimenin en sağından itibaren başlamakta ve belli bir grup sembol modelini kelimenin ilk karakterini tanımakta kullanılmaktadır. Daha sonra geriye kalan modeller, eşleme sistemine aktarılmakta diğer karakterler için en yüksek olasılıktaki uzaysal yerleşim bulunmaktadır. Eşleşen modellerin her biri bir durum (state) olarak kabul edilmektedir. Aynı işlem tüm kelime boyunca devam etmektedir.

4.14.3. Kelimenin doğru tanınma olasılığı

M, bir sembol modeli ve t bu modelin kelime görüntüsü koordinatları üzerinde bir ifadesi olmak üzere n adet sembol modelinin uzaysal yerleşimi $\{(M_1, t_1) \dots (M_n, t_n)\}$ olarak gösterilmektedir. Modeller, merkezlerinin x koordinatına göre sıralanmıştır. Ayrıca, 5 kelime üzerinde tespit edilen ve eşlemede kullanılmamış olan alt düzey şekillerin bir alt kümesi, I kelimenin görüntüsü ve M modeli ile belirtilebilen karakter $I(M)$ olsun. Bu durumda, $I(M_1) \dots I(M_n)$ karakterlerinden oluşan dizi, bilinmeyen bir kelimeye karşılık gelen giriş verisine göre tanıma sisteminin ürettiği sonuçtur.

Uzaysal bir yerleşimin olasılığı, içerdiği modellerin eşleşme olasılıklarından hesaplanır. Dönüştürüldüğü kelime bölgesiyle eşleşebilen her sembolün olasılığı eşleyici kesim tarafından belirlenir. Bu yüzden, her sembolün eşleşme olasılığı, diğer eşleşmelerden bağımsız olarak ele alınır. O halde, bir kelime için olasılık dağılımı etkileyici kesim tarafından belirlenir. Bu yüzden, her sembolün eşleşme olasılığı, diğer eşleşmelerden bağımsız olarak ele alınır. O halde, bir kelime için olasılık dağılımı

$$P(\{M_1, t_1\}, \dots, \{M_n, t_n\}, S / I) = \left(\prod_i^n P(\{M_i, t_i\} / I) \right) x P(S / I) \quad [4.37]$$

Yukarıdaki denklemin sağ tarafındaki ifadenin ilk terimi, model eşleme olasılıkları çarpımıdır. Eğer sistem bir kelime elde ettiğine karar verirse (kelimenin sonlandığını kabul ederse), geriye kalan tespit edilemeyen alt düzey şekilleri fazlalık olarak kabul etmektedir. Fazlalık olan diğer alt düzey şekillerin olasılığı (ikinci terim), her bağımsız fazlalık şeklin olasılıkları çarpımıdır.

4.14.4. Model eşleme

Bir sembol modelinin ele alınması, modeli kapsayan kutunun kelime görüntüsünde nereye yerleştirileceğinin belirlenmesiyle olmaktadır. Bu modelin, kelime içindeki

uygun bölgeyle eşleşmesi için, modeli oluşturan alt düzey şekillerin o bölgeyi oluşturan alt düzey şekillerle birebir eşleşmesi gerekir Modellerin herhangi bir alt düzey şeklinin eşleşmesi için:

- Kendisine benzeyen bir alt düzey şeklin kelime görüntüsü içinde bulunması
- Eşleme alanı hesaplanması (Eşleşen şekillerin ne kadar farklı olduğunun bulunması)
- Eşleşme olasılığının hesaplanması

Bir sembol modeli, kelime bölgesine dönüştürüldüğü zaman sistem modelin alt düzey şekilleriyle örtüşen kelime şekillerini belirler. Kelime üzerinde örtüşen bu bölge, modelin bu alt düzey şeklinin tüm olası deformasyonlarını karşılayabilecek olmalıdır.

Eşleşme hatası, bir alt düzey şekil ile örtüşen kelime bölgesindeki toplam piksel sayısl olarak hesaplanmaktadır. (Kanungo et al,1993) İzleyen sayfada tarayıcıdan alınmış bir kelime görüntüsü ve bu kelime üzerinde bir sembol modelinin 4 varyasyonu için kesişim bölgeleri gösterilmiştir.



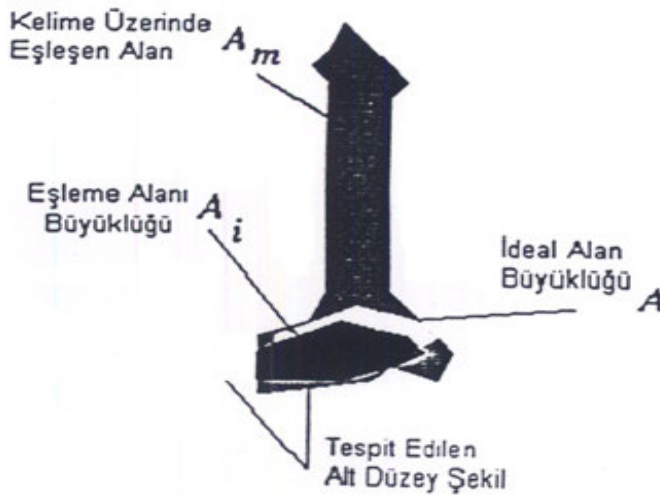
Şekil 4.15. Tarayıcıdan alınan bir kelimenin görüntüsü



Şekil 4.16. Bir önceki şekilde görülen kelimenin görüntüsü üzerinde bir sembol sembol modelinin eşleşmesi

T dönüşümündeki bir sembol modeli ile bir I görüntüsü arasındaki eşleşme olasılığı, modelin tüm alt düzey şekillerinin eşleşme olasılıklarının birleşimi olarak hesaplanır. Hesaplamayı kolaylaştırmak için, her alt düzey şeklin eşleşme olasılığının diğerlerinden bağımsız olduğu kabul edilmiştir. Böylece

$$P((M, t) / I) = P_M(A_1, \dots, A_n, P_1, \dots, P_n) = \left(\prod_{i=1}^n P_M(A_i) \right) \times P_M(D_1, \dots, D_n) \quad [4.38]$$



Şekil 4.17. Bir kelime görüntüsü üzerinde alt düzey bir şeklin kesişimi ve alanlar

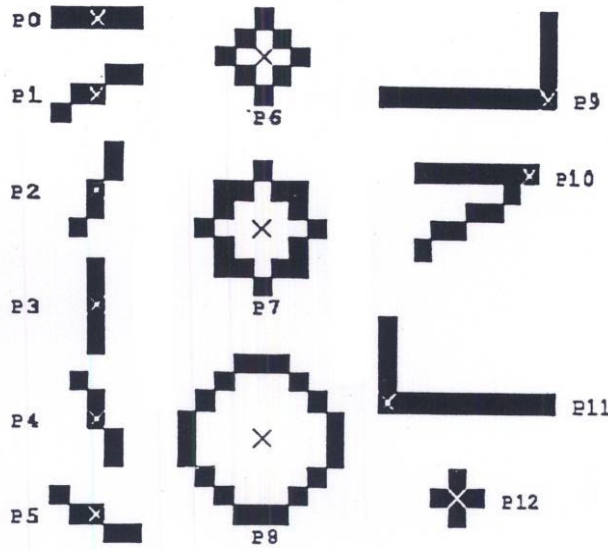
Burada A, modelin ilgili alt düzey şeklinin alanı, A_i eşleşme alanının büyüklüğü, A_M kelime üzerinde eşleşen alanın büyüklüğü, a ve c dağılım parametreleridir. Bu alanlar Şekil 4.18'de gösterilmiştir.

Dağılım parametreleri, çok sayıda gürültülü sembol görüntüleri içeren eğitime seti kullanılarak Bayesian kestirimi yardımıyla bulunmuştur. (Al-Badr, 1995)

4.14.5. Sonuçlar

Tanıma sistemi C diliyle geliştirilmiş ve Sun Spare 10 bilgisayarında çalıştırılmıştır. Sistemin tanınması için eğitildiği toplam sembol sayısı 156'dır. Bunlar arasında 28

Arapça harf, 20 noktalama işareti, 10 rakam ve 13 adet yazmada kullanılan yardımcı işaretler bulunmaktadır. Deneyler için kullanılan sözlük 42.000'in üzerinde kelime içermektedir ve değişik gazetelerden, magazinlerden ve kitaplardan toplanmıştır. Tanıma sistemi, Şekil 4.18'de gösterilen 13 değişik alt düzey şekli kullanmaktadır.



Şekil 4.18. Eşleme işlemindeki alt düzey şekiller. Her kare bir görüntü pikseline karşılıklı gelmektedir. Her şeklin merkezi + (Artı) ile gösterilmiştir

Tanıma sistemi, üzerlerine Kanungo modeli kullanılarak gürültüler eklenmiş, sentetik olarak üretilen izole semboller ile eğitilmiştir. (Kanungo et al, 1993) Sentetik olarak üretilen semboller de ilk etapta gürültü yoktur. Çünkü, postscript formatından görüntü formatına on-line olarak çevrilmiştir.

Gürültü parametreleri $\alpha_0 = \beta_0 = 1,0$, $\alpha = \beta = 1,5$ $c_0 = 0,0$ ve $e = 3$ olarak belirlenmiştir. Eğitim seti her sembol için 500 gürültülü görüntü olmak üzere, toplam 78 000 sembol içermektedir. Eğitim ve test için kullanılan bir Arapça font Nadeem Machintosh ve fon büyüklüğü 12'dir.

Test setindeki veriler, bir gazete magazininden seçilen yedi sayfadan elde edilmiştir. Bu sayfalar, 300 dpi çözünürlükte bir tarayıcı ile bilgisayar ortamına aktarılmıştır. Bu verilerden 3 değişik test seti elde edilmiştir:

- 1) İdeal Set: Yedi sayfalık dökümanın gürültüsüz hale dönüştürülmesinden sonra elde edilen kelimeler.
- 2) Gürültülü Set: Yukarıda bahsedilen kelimelere sentetik olarak eklenen gürültü ile elde edilen veriler.
- 3) Gerçek Set: Tarayıcıdan alınan yedi sayfalık görüntü verisi.

Yukarıda bahsedilen test setinin her birinden kullanılan kelime sayısı, doğru tanıma oranı ve işlem süresi aşağıdaki Çizelge 4.5' de listelenmiştir. Tanıma hatalarının çoğu birbirine çok benzer sembollerin birbirine karıştırılması şeklinde ortaya çıkmıştır. Çizelge 4.6, solda sağa doğru sırasıyla birbirine en çok karıştırılan sembol çiftlerini göstermektedir.

Sistem oldukça yavaş çalışmaktadır. Bir kelimenin tanınması için algoritmanın en kötü çalışma zamanı, kelime boyuyla üssel oranda artmaktadır. Kısa kelimeler için tanıma hızlıyken, bazı uzun kelimeler için 300 saniyeden fazla sürmektedir.

Çizelge 4.5. Üç değişik test seti için elde edilen sonuçlar

Test Seti Türü	Kelime Sayısı	Tanıma Oranı	Süre/Kelime(msn)
İdeal	3787	% 99,39	16,718
Gürültülü	3522	% 95,60	59,760
Gerçek	830	% 73,13	224,593

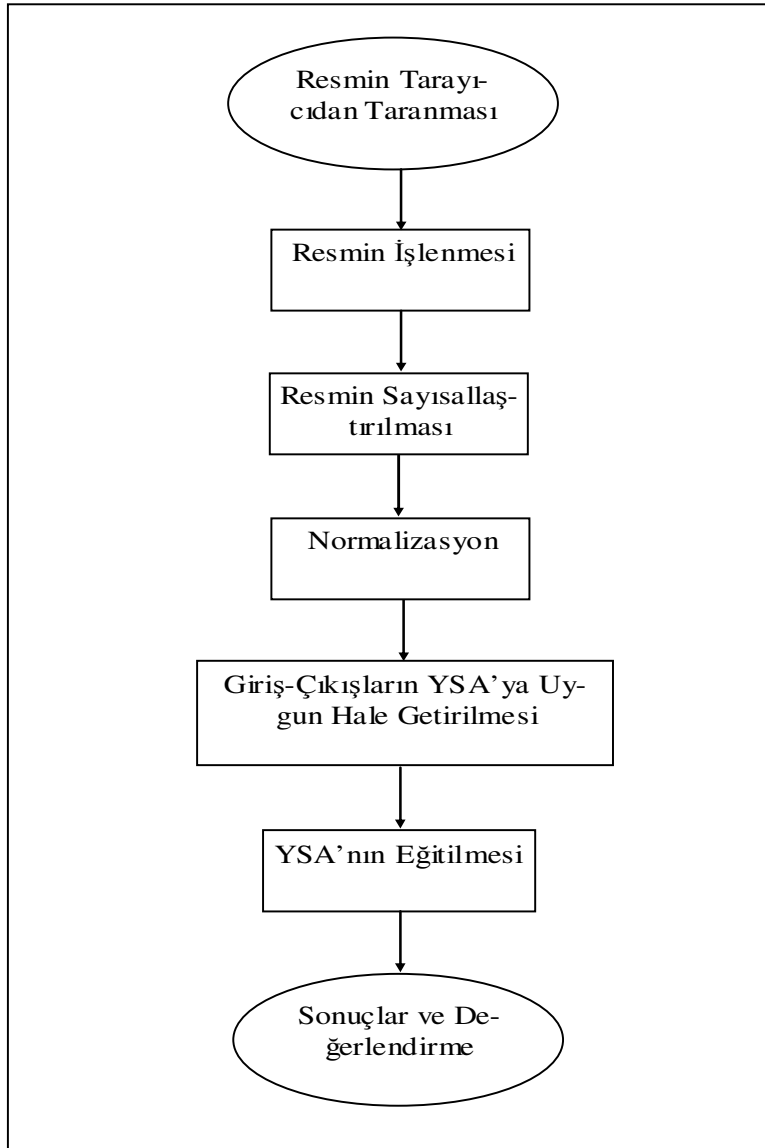
Bu çalışmada, kelimelerin doküman sayfaları arasından önceden çıkarılmış olduğu ve yönlerinin değiştirilmediği varsayılmıştır. Bu sistemin pratikte uygulanabilmesi için kelimelerin otomatik olarak metinden çıkarılması gerekmektedir.

لا	ج	لا	لا	لا	أ	خ	»	«	ض
لا	ح	لا	ن	لا	ا	ح	«	»	ص

Şekil 4.19. Birbiriyle en çok karıştırılan karakterler. karıştırma sıklığı soldan sağa doğru azalmaktadır

5. OSMANLICA KARAKTERLERİN TANINMASI İÇİN TASARLANAN YSA UYGULAMASI

Uygulama için tasarlanan sistem; Osmanlıca ayrik harf karakterlerinin tarayıcıdan resimlerin taranması basamağı ile başlayıp, resim işleme, resmin sayısallaştırılması, normalizasyon, giriş-çıkışların YSA'ya uygun hale getirilmesi, YSA'nın eğitilmesi ve en son sonuçların değerlendirilmesi basamağı ile son bulmaktadır. Sistemin her bir basamağı aşağıda detaylı bir şekilde açıklanacaktır.



Şekil 5.1 Tasarlanan sistemin genel şeması

Osmanlıca ayrıık harf karakterlerinin bilgisayar destekli tanınması uygulamasında tasarlanan sistemin genel şemasından hemen sonra incelemeler için uygulama genel başlıkları ve simülasyon basamakları ile ilgili genel bilgiler verilmiştir. Şekil 6.2' de genel uygulama basamaklarını gösteren şekil verilmiştir.

Uygulama Osmanlıca ayrıık harf karakterlerinin bilgisayar destekli tanınması uygulamasıdır.

Genel olarak:

- 1- Genel Bilgiler
- 2- Resim İşleme
- 3- Resim Tanıma

Üç ana başlık altında açıklanacaktır.

1- GENEL BİLGİLER

Karakter tanımanın amacını, insana özgü okuma özelliğini, çeşitli uygulamalar kullanarak bilgisayarlara taklit ettirmek olarak belirtebiliriz. Osmanlıca harfleri tanıma işlemi harflerin özelliklerinden dolayı oldukça zor bir iştir. Latin harfleri tanıma sistemleri alanında fazlaca çalışma olmasına rağmen Arapça harfleri tanıma sistemleri konusunda yeterince çalışma yoktur. Osmanlıca harfler de, temelde Arapça harflerinden oluşmaktadır. Bu çalışmada el yazısı Osmanlıca harfleri tanımaya ilişkin bir uygulama tanıtılmış ve uygulanmıştır. Uygulama ayrıık yazılmış harf örnekleri üzerinde gerçekleştirilmiştir. Osmanlıca yazılı harflerin olduğu bir belge tarayıcıdan taranarak bilgisayar ortamına alınması işleminden sonra görüntü işleme teknikleri kullanılarak, taranan karakterler sayısal verilere dönüştürülmüştür. Verilerin düzenlenmesi için bir normalizasyon işlemi yapılarak, YSA için giriş- çıkış değerleri elde edilmiştir.

2- RESİM İŞLEME

Osmanlıca yazılı harflerin olduğu bir belge tarayıcıdan taranarak bilgisayar ortamına alınması işleminden sonra görüntü

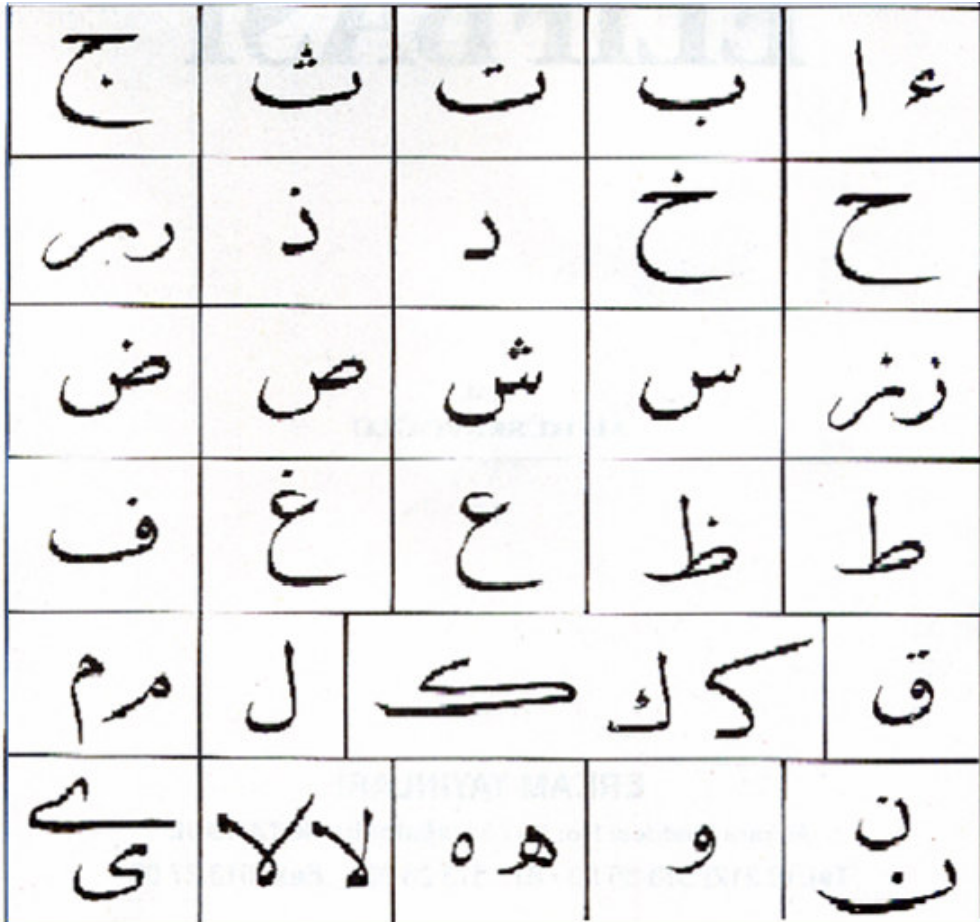
3- RESİM TANIMA VE YSA YAPISI

Uygulama için tasarlanan yapay sinir ağı; 100 girişli, 25'er düğüm elemanlı iki gizli (ara) katman ve 31 çıkışlı Çok Katmanlı Geri Yayımlı Ağ yapısı tasarlanmıştır.

Şekil 5.2. Uygulama basamakları ile ilgili genel bilgileri içeren programın ön yüzü

5.1. Harf Karakterlerinin Tarayıcıdan Taranması

Ayrık osmanlıca harf karakterlerin olduğu bir belge tarayıcıdan taranarak resimlerin bilgisayar ortamına aktarılması işlemi uygulama sistemin ilk basamağını oluşturmaktadır. Belge üzerinde ayrık harf karakterlerinin yanında belge üzerinde bulunan gürültüler de bulunmaktadır. Taranan belgede harf karakterleri ayrık durumda bulunmamaktadır. Ayrık harf karakterlerinin tek harf şeklinde işlenmesi için belge değişik resim işleme-düzenleme programları (Photoshop, Paint) kullanılmıştır. Resimlerdeki gürültülerin giderilmesinde de bu programların değişik fonksiyonlarından yararlanılmıştır. Aşağıda Şekil 5.3' de taranan belge ve belge üzerinde harf karakterleri gösterilmiştir.



Şekil 5.3. Tarayıcıdan taranan harf karakterini içeren belge

Ayrık harf karakterlerinin tek harf şeklinde işlenmesi için belge değişik resim işleme-düzenleme programları (Photoshop, Paint) kullanılmıştır. Şekil 5.4'de tarayıcıdan taranan tek bir harf karakteri gösterilmiştir.



Şekil 5.4. Tarayıcıdan taranan bir harf karakteri

5.2. Resim İşleme

Burada taranan resim işlenebilmesi için resim BMP resim formatında kaydedilmelidir.

Tarayıcıdan alınan karakter görüntüsü harici düzenlemeler dışında, alınan görüntüye yansıyan olumsuzlukların giderilmesi, azaltılması ve ayrıca tanıma aşamasına hazırlanmasında, görüntü işleme yöntemleri kullanılır.

Görüntü iyileştirme sonrasında, algılanan görüntü içinde yer alan nesnenin genel görüntü çerçevesinden ayrıştırılması gerekmektedir. Sayısal formdaki görüntünün içerdiği renk ve ton farklılıkları, nesneyi arka zeminden ayırt etmeye yarayan

özeliklerdir. Uygulamada harf karakterlerinin tanınması işleminden önce Windows ortamı için BMP dosya sistemi ile ilgili bilgi verilecektir.

BMP Grafik formatı:

BMP dosya sistemi hem OS/2 hem de Windows ortamı için resim (imaj) kaydetmek için tasarlanmıştır. Bu bölümde sadece Windows ortamı için BMP dosya biçimi anlatılacaktır.

BMP dosyalar temel olarak üç kısımdan oluşur.

1. Header (Başlık)

BMP dosyanın başlığında, görüntü hakkında istenilen her türlü bilgiyi bulunabiliriz. İlk iki byte her zaman için "BM" karakterlerini içerir. Bir dosyanın BMP olup olmadığını anlamak için ilk iki karakteri okuyup, bu iki karakterin "BM" olup olmadığını kontrol etmek yeterlidir. Sonraki 4 byte dosya uzunluğunu verir.

Offset 18. ve 22. sırasıyla görüntünün X ve Y uzunluklarını piksel cinsinden büyüklüğünü içerir. Bundan sonraki iki byte ise piksel başına kullanılan bit sayısını gösterir ve 1, 4, 8 ve 24 değerlerinden birini içerir. Bu bilgi sayesinde BMP dosyasının kaç renkli olduğunu anlayabiliriz. 1, 4, 8, 24 değerlerine bağlı olarak sırasıyla resim 2, 16, 256 ve 16 milyon renklidir.

2. Renk Tablosu

Başlık kısmından sonra renk haritası gelir. Bu harita VGA renk tablosunun içeriğine yazılacak renk bilgilerini içerir. Her renk bu haritada 4 byte'lık yer kaplar. Bu 4 byte'lık bilginin ilk üçü sırasıyla rengin mavi, yeşil ve kırmızı (ters sırada) bileşenlerine karşılık gelir. Sonuncusu ise kullanılmaz. Renklerin bileşenleri

saklanırken 0-255 arası değerler kullanıldığında VGA renk tablosunun içeriğine bu değerler 4'e bölünerek yazılır.

3. Görüntü Verisi

Renk haritasından sonra görüntüyü içeren veri kısmı bulunur. Bu veri kısmı ters kaydedilmiştir. Yani okuyacağımız ilk veri görüntünün en alt sol köşesine aittir. Görüntü kaydedilirken yatayda soldan sağa dikeyde ise aşağıdan yukarıya taranarak kaydedilmiştir. 256 renkli görüntülerde her pikse 18 bit, 16 renkli görüntülerde ise her piksel 4 bit'lik yer kaplar. Diğer önemli bir nokta ise tüm satır bilgilerinin 4 byte'ın katı şeklinde kaydedilmesidir.

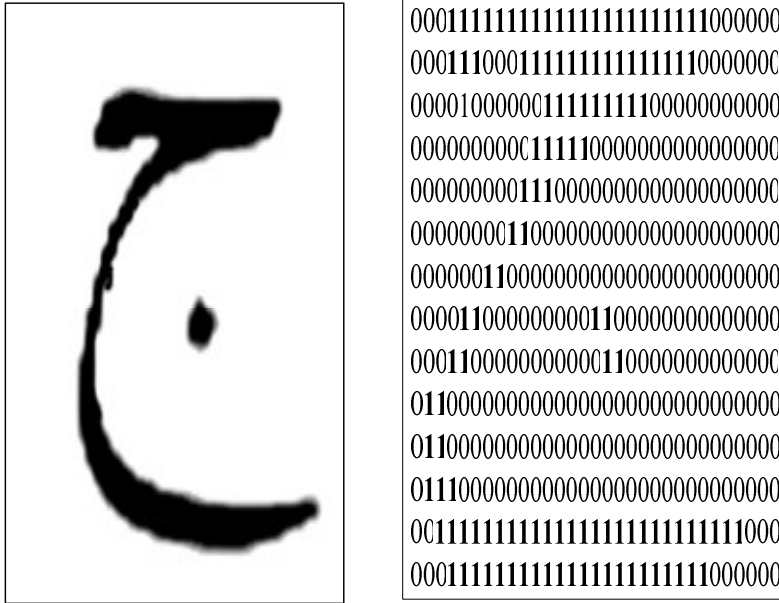


Şekil 5.5. BMP uzantılı bir resim

5.3. Resim Sayısallaştırma

Görüntü piksellerinin taşıdığı gri tonların seçilen bir eşik seviyesine göre siyah beyaz olarak iki seviyeye dönüştürülmesi sonucu, ikili görüntü elde edilmektedir. İkili görüntüde nesneyi tanımlayan siyah pikseller 1, arka zemini oluşturan beyazlar ise, 0 değeri ile temsil edilir. Bu sayısal dönüştürmede, tarama işleminden sonra resimde

bulunan arka zemin gürültüleri yukarıda ismi verilen resim işleme programları yardımı ile gidermek gerekir. Aksi takdirde pikseller bu nesneyi tanımlayan gerçek veriler olmaktan uzaklaşacaktır.

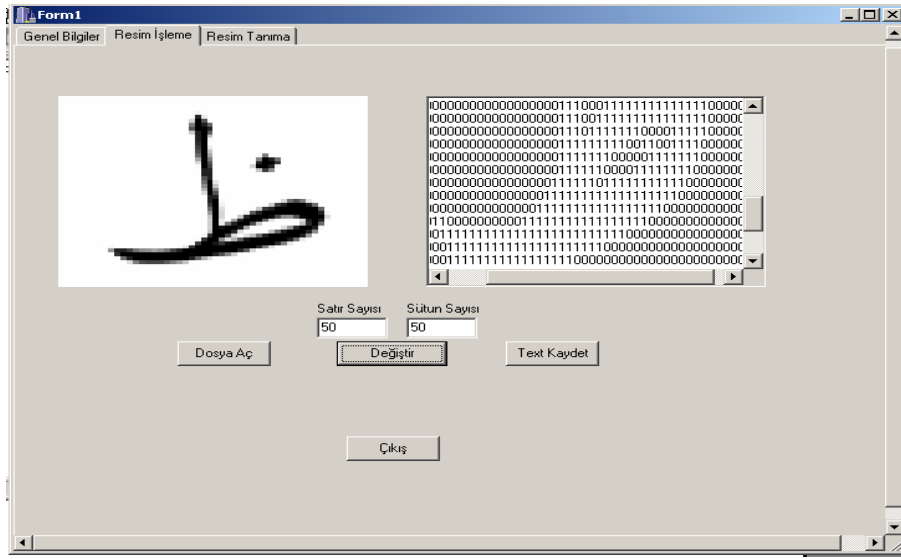


Şekil 5.6. Bir harf karakteri ve sayısal görüntüsü

5.4.Normalizasyon

Resim işleme uygulamalarında, işleme giren veri sayısı ve ihtiyaç duyulan işlem hacmi büyük boyutlara ulaşmaktadır. BMP resim formatında Offset 18. ve 22. sırasıyla görüntünün X ve Y uzunluklarını piksel cinsinden büyüklüğünü içermektedir. Burada harf karakterinin mümkün olduğunca orijinal görüntüsünü bozmadan, sayısallaştırılmış resimde satır-sütun değerleri değiştirilir. Aşağıda uygulanan programın ön yüzünde sayısal resim istenen satır-sütun uzunluk değerlerinde küçültülebilir.

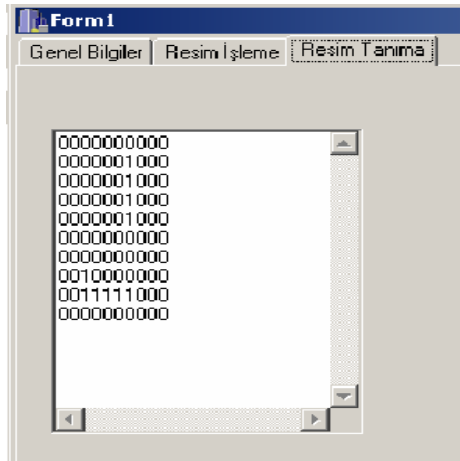
Bu satır-sütun değerleri değiştirilirken gerekli hassasiyet gösterilmediği takdirde harf karakterinin orijinalliği kaybolmaktadır.



Şekil 5.7. Normalizasyon için görüntünün X ve Y uzunluklarını piksel cinsinden büyüklüğünü değiştirmek için kullanılan programın ön yüzü

5.5. Giriş-Çıkışların YSA'ya Uygun Hale Getirilmesi

Yapay Sinir Ağları uygulamalarında giriş ve çıkış verilerinin YSA'nın yapısına uygun hale getirilmesi gerekmektedir. Uygulamada YSA giriş verisi için 10x10 boyutunda bir giriş matrisi, çıkışlar için de ayrıık harf karakter sayısınca 31 çıkış verisi gerekmektedir. Normalizasyon işleminden sonra program bu verileri kendisi yapmaktadır.



Şekil 5.8. 10x10'luk YSA'nın girişlerini gösteren programın ön yüzü

5.6. YSA'nın Eğitilmesi

YSA yapısı:

Bir YSA tasarımında temel amaç, YSA sisteminin mümkün olduğunca basit bir yapıda tasarlanmasına dikkat etmektir. Bu karmaşıklığın artması öğrenme sürecini doğrudan arttırmakta ve yapının performansını olumsuz yönde etkileyebilmektedir.

YSA'nın en uygun parametrelerle tasarlanması durumunda YSA sürekli olarak kararlı veya istikrarlı sonuçlar üretebilecek ve bu sayede gürbüz bir sistem tasarımı gerçekleştirilebilecektir. Ayrıca sistemin tepki süresinin yeterince kısa olabilmesi içinde ağ büyüklüğünün yeterince küçük olması gerekir. İhtiyaç duyulan toplam hesaplama da bu sayede sağlanmış olacaktır. Bunun için uygun olan yaklaşımı belirlemek YSA uygulama başarısını arttıracaktır. Gereğinden fazla sayıda işlemci eleman içeren ağ yapılarında, daha düşük genelleme kabiliyetleriyle karşılaşılacağı unutulmamalıdır. Bunun yanında belirli sayıların üzerinde nöron kullanmak da YSA performansını çoğu zaman yükseltmemektedir.

Uygun YSA mimarisininin veya yapısının seçimi, uygulama problemine bağlı olarak değişmektedir. Uygulanacak problemin veri yapısı, bu seçimi etkileyecek diğer önemli bir husustur. Uygulama problemi bir sınıflandırma problemi ise problemin zorluğuna ve ayrıştırılacak sınıf sayısına göre, MLP mimarisi ile başlamak doğru bir yaklaşım olacaktır.

YSA uygulamalarında, en uygun sayıdaki katman ve katmanlardaki nöron sayısını bulmak için her hangi bir kural yoktur. Genellikle sistemin karmaşıklığı, gizli katmanların ve gizli katmanlardaki nöronların sayısının artmasıyla giderilir. Çalışmanın başında sinir ağı bir giriş katmanı, bir gizli katmanı ve bir çıkış katmanı olarak üç katmanlı seçilmiştir. Deneme ve yanılma yoluyla bu sistem için en uygun ağ yapısı bir giriş katmanı, iki gizli katman ve bir çıkış katmanıdır. Sırasıyla her bir katmandaki düğüm sayıları ise 100, 25, 25 ve 31 olarak bulunmuştur. Burada çıkış

katmanındaki düğüm sayısı, her bir ayrık harf karakteri için bir düğüm olacak şekilde tasarlanmıştır.

Her bir nöronun çıkışı ağırlıklar üzerinden bir sonraki katmandaki nöronun girişi olarak bağlanmıştır. Aynı zamanda her nörona bir eşik değeri ekleyerek ağın daha hızlı öğrenmesini ve öğrenme kapasitesinin artırılması sağlanmıştır. Her bir katmandaki nöronun çıkışına aktivasyon fonksiyonu olarak hiperbolik tanjant sigmoid aktivasyon fonksiyonu kullanılmıştır.

Şekil 5.9. Tasarlanan ağ yapısı değerlerinin girildiği programın ön yüzü

Eğitim algoritması:

Bir YSA yapısı seçiminden sonra uygulama başarısını belirleyen en önemli faktör ise öğrenme algoritmasıdır. Seçilen YSA yapısı için en uygun öğrenme algoritmasının seçimi ağ yapısının seçiminden sonra çözülmesi gereken en önemli problemdir.

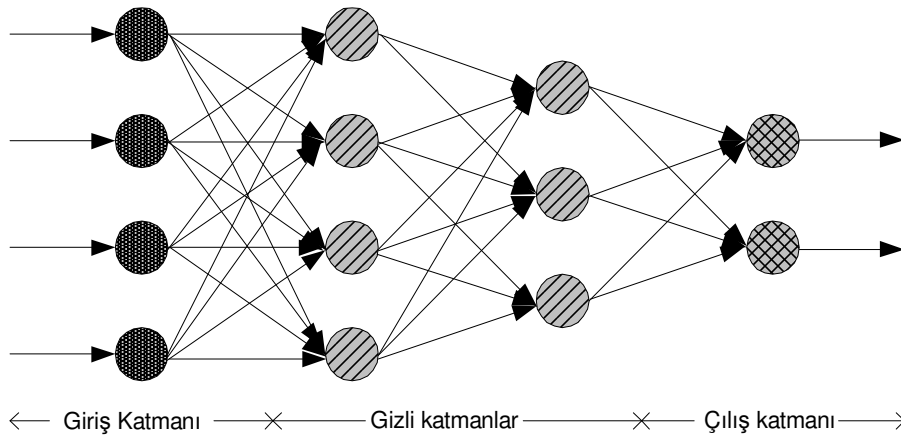
Genellikle ağ yapısı öğrenme algoritmasının seçiminde belirleyicidir. Bu nedenle seçilen ağ yapısı üzerinde kullanılacak öğrenme algoritmasının seçimi, çoğunlukla ağ mimarisine de bağlıdır.

YSA'ların az veriyle öğrenebilmeleri veya genelleme yapabilmeleri YSA'ları birçok probleme uygulanması için cazip kılmaktadır. Bunun sebebi ise deneysel olarak fazlaca veri elde etmenin maliyeti arttırmasıdır. Uygulamada kullanılan her bir ayrık harf karakterinin tanınması için taranan harflerin işlenen sayısal verileri kullanıldığı için ağların eğitimi için kullanılan öğrenme kuralları olarak danışmanlı öğrenme kuralı seçilmiştir. Bu öğrenme kuralında, gerçek çıkış istenen çıkışla kıyaslanır. Rasgele değişen ağırlıklar ağ tarafından öyle ayarlanır ki, bir sonraki döngüde gerçek ile istenen çıkış arasında daha yakın karşılaştırma üretebilsin.

Öğrenme yöntemi, işleme elemanlarının anlık hatalarını en aza indirmeye çalışır. Bu hata azaltma işlemi, kabul edilebilir doğruluğa ulaşıncaya kadar ağırlıklar devamlı olarak güncellenir. Delta kuralı, ağırlıkları değiştiren eşitlik ağın toplam karesel hatasını en aza indirmek için tasarlanmıştır. Hata olabilecek en küçük bir değere düştüğünde eğitime durdurulmuştur. Çünkü gereğinden fazla eğitime yaptırmak ağda ezberlemeye gitmekte ve doğru sonuç vermemektedir. Sinir ağı eğitiminde çevrim dışı eğitime kullanılmıştır.

Hesaplamalara momentum teriminin ilave edilmesi ağ performansı üzerinde etkili olduğu gözlenmiştir. Momentum teriminin hesaplamaya katılması adım sayısında ve toplam ağ hatasında bir düşüş meydana getirmektedir. Öğrenme katsayısı ağ performansı üzerinde önemli bir etkiye sahiptir. Küçük öğrenme katsayısı değerleri için eğitime işlemi uzun zaman alırken bu değerlerin büyütülmesi ile eğitime işlemi daha kısa zamanda gerçekleşmektedir. Öğrenme katsayısı arttırılması durumunda öğrenme için gerekli adım sayısında azalma meydana gelmektedir. Öğrenme katsayısı arttırılması ağın toplam hatası üzerinde bir iyileştirme meydana getirmektedir. Fakat öğrenme katsayısına çok büyük değerlerin verilmesi durumunda pek bir yakınsama olayı meydana gelmemektedir.

Uygulama için tasarlanan yapay sinir ağı; 100 girişli, 25'er düğüm elemanlı iki gizli (ara) katman ve 31 çıkışlı Çok Katmanlı Geri Yayımlı Ağ yapısı ve öğrenme algoritmasında öğrenme katsayısı ve momentum katsayıları; 0.01 olarak tasarlanan ağ yapısı Şkil 5.10 de gösterilmiştir.



Şekil 5.10 Uygulama için tasarlanan YSA ağ yapısı genel şeması

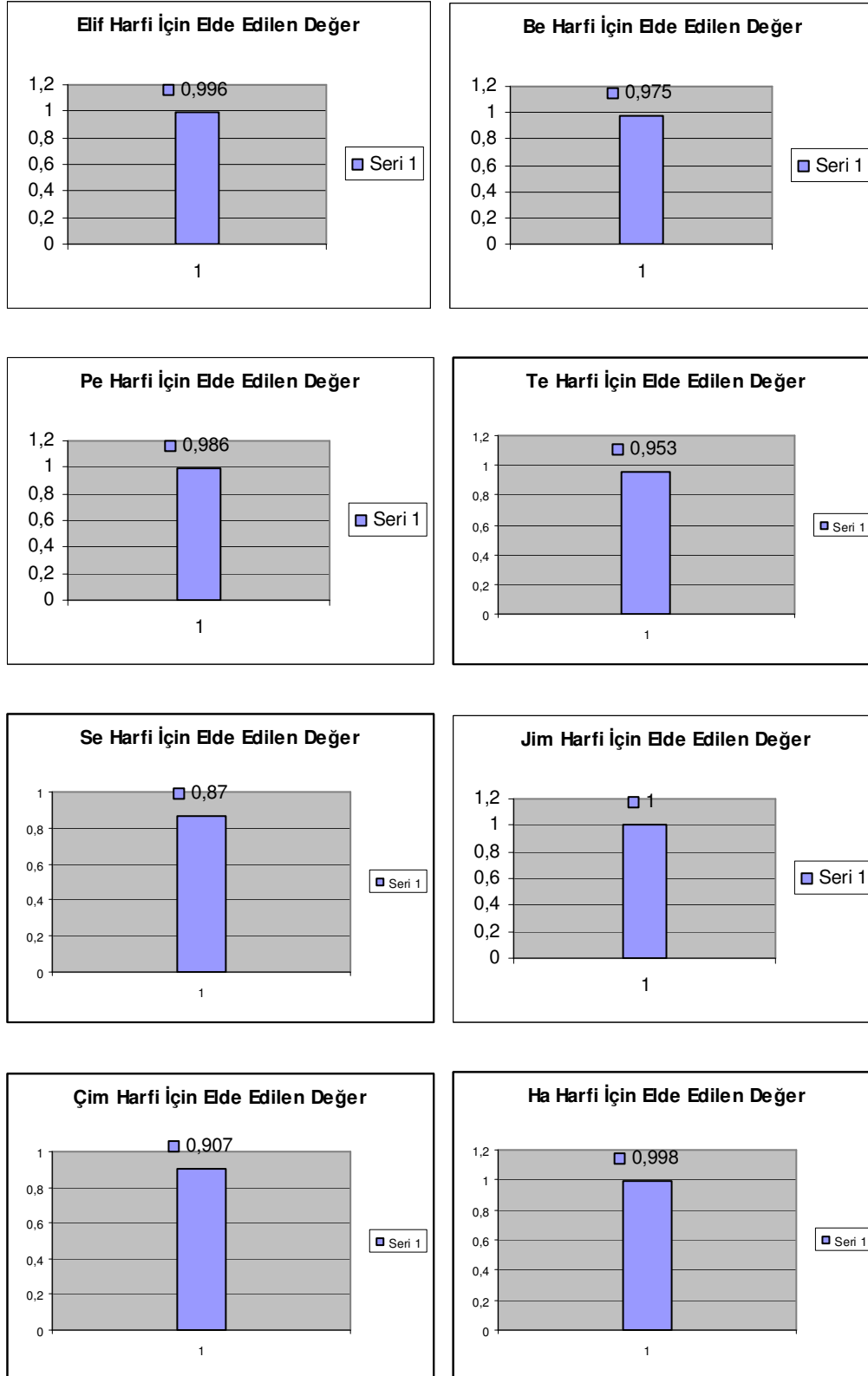
5.7. Deneysel Sonuçlar

İşlenen veriler uygulama safhasında, her bir harf karakteri için bilinen gerçek değer ile eğitime sonucunda çıkan değer arasında farkın(hatanın) en az olması durumuna kadar ağı eğitime devam edilir. Ancak burada eğitime süresince elde edilen *yanıltıcı* en iyi değerlerin gerçekten en iyi sonuçlar olup olmadığı durumudur. Uygulayıcı bunu göz önünde bulundurmalıdır. Tasarlanan uygulama için elde edilen deneysel sonuçlar aşağıdaki çizelgede her harf karakteri için verilmiştir. Uygulamada her harf için:

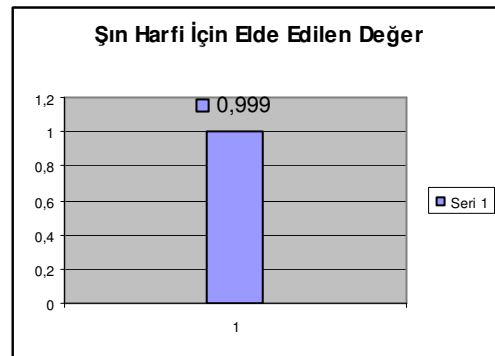
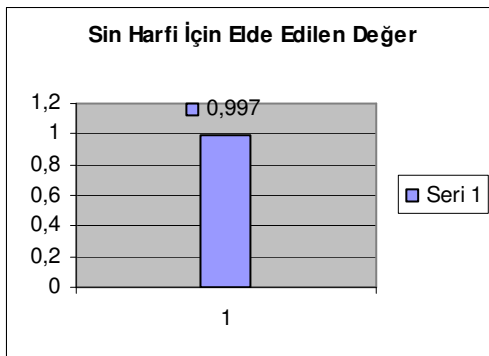
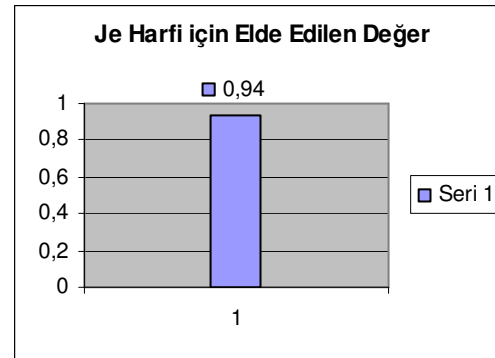
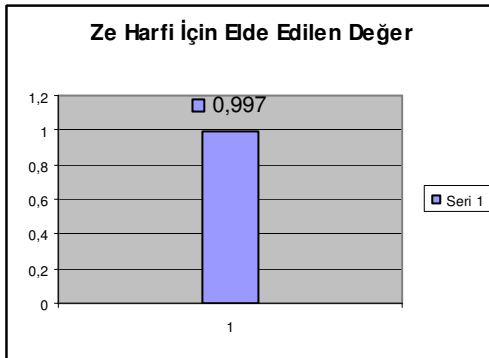
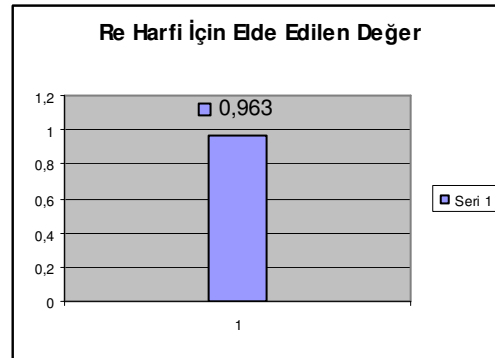
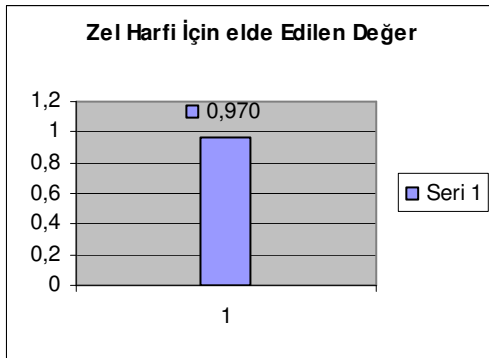
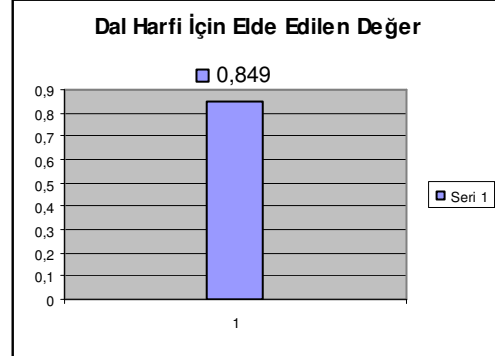
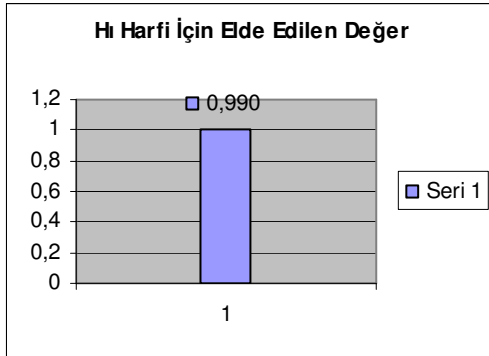
Gerçek değer: 1

Çıkan değer: 1 'e en yakın çıkan değer olarak alınmıştır.

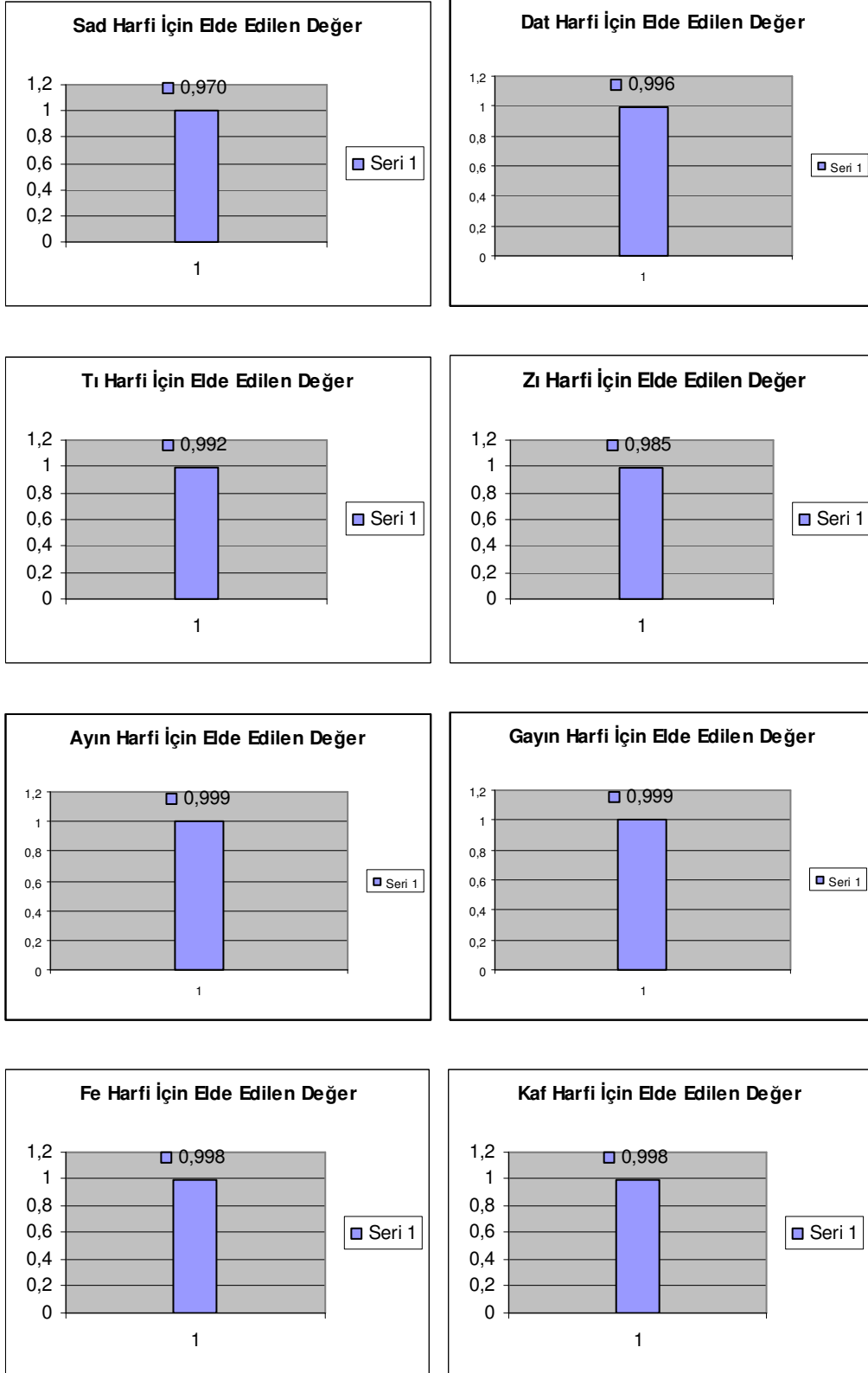
Çizelge 5.1. Her bir harf karakteri için elde edilen deney sonuçları



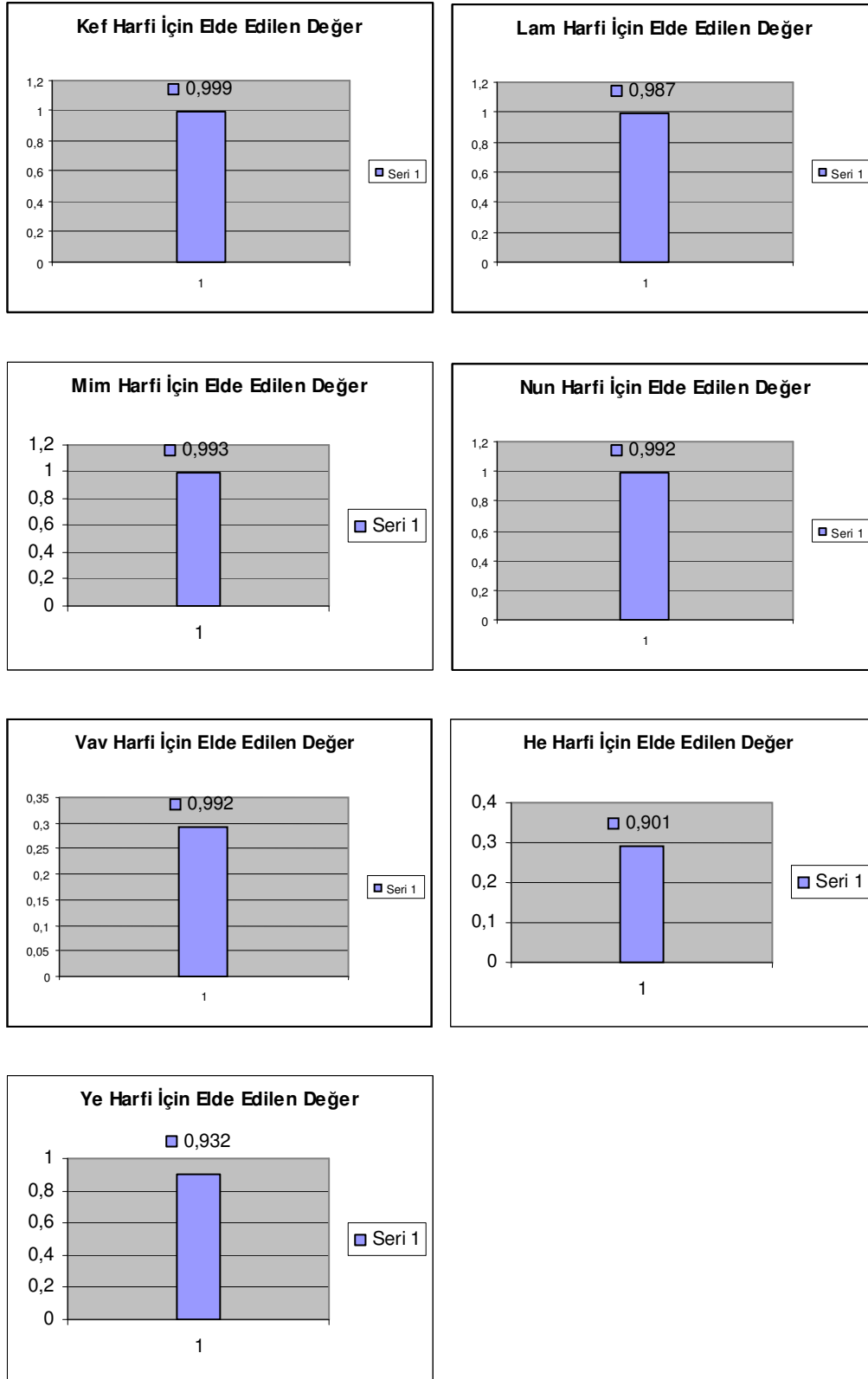
Çizelge 5.1. (Devam) Her bir harf karakteri için elde edilen deney sonuçları



Çizelge 5.1. (Devam) Her bir harf karakteri için elde edilen deney sonuçları



Çizelge 5.1. (Devam) Her bir harf karakteri için elde edilen deney sonuçları



6. SONUÇ VE ÖNERİLER

Bu tezde, osmanlıca ayırık harf karakterleri el yazıları, Yapay Sinir Ağları (YSA) ile tanınması uygulaması yapılmıştır.

Uygulamanın en önemli kısmını YSA teşkil etmektedir. Tasarlanan yapay sinir ağı; 100 girişli, 25'er düğüm elemanlı iki gizli (ara) katman ve 31 çıkışlı Çok Katmanlı Geri Yayımlı Ağ yapısı ve öğrenme algoritmasında öğrenme katsayısı ve momentum katsayıları; 0.01 olan ağ yapısıdır. Uygulama değerlendirildiğinde sonuçları açısından başarılı olmuştur.

Harf karakterlerin taranması, taranan resimlerin programlar yardımı ile işlenmesi kısmı ile YSA kısmı uygulamanın ayrı zaman bölümlerini teşkil etmektedir. Tarama ve resimlerin işlenmesi olan birinci kısım uygulamayı yapan kişi ile ilgili iken, YSA kısmı ise daha çok kullanılan verilerin fazlalığına ve bu verileri işleyen bilgisayarlara bağlı bir durumdur.

Tasarlanan uygulamada; resim işleme uygulamalarında işleme giren veri sayısının çok olması programların hantallaşması, hızlı çalışma ve erken sonuç vermesini etkilemektedir.

Karakter tanımlama uygulamalarında verilerinin işlenmesi ve bu verilerin orijinal karakterleri betimlemesi uygulamanın çok önemli kısmını teşkil ettiği bilinmelidir. Uygulamanın bu önemli bölümünden sonra kullanılan YSA'nın özellikleride uygulamanın başarısı üzerinde çok önemli etkileri sahiptir. YSA'nın aşağıda belirtilen özellikleri uygulamanın başarısı açısından son derece önemlidir.

- Probleme uygun olan YSA yapısı veya mimarisi seçimi,
- Problemin kabul edilebilir çözümü için YSA giriş ve çıkış sayılarının en uygun veya en az sayıda seçimi,

- Arakatman nöron sayılarının en uygun sayıda belirlenmesi,
- Arakatman sayısının seçimi,
- Kullanılacak öğrenme algoritmasının YSA yapısına uygun olması,
- Veri normalizasyon yaklaşımı,
- Seçilen transfer fonksiyonunun yapısı,
- Uygun veri tipinin ve sayısının belirlenmesi,

Seçimi olarak sıralanabilir.

Uygulamada bir diğer önemli husus taranan belgeler üzerinde orijinal resim yanında diğer gürültülerinde bulunması resim işlemeyi gerekli kılmaktadır. Ancak işlenen resimlerde gürültülerin temizlenmesinin yanında orijinal verilerin bir kısmının da kayıp olması, uygulamada hata oranını yükseltmektedir. Resimlerin çok fazla sayısal veri içermesi bir normalizasyon gerektirmesi ve uygulanan normalizasyonun iyi seçilememesi durumunda bu hata oranı daha da artmaktadır.

Uygulamanın Yapay Sinir Ağı (YSA) eğitim kısmı (Intel Pentium 1.60 GHz, 512 RAM) özellikli bir bilgisayarda 10-14 saat sürmüştür. Uygulamanın tarama ve resimlerin işlenmesi kısmı ise uygulayıcıya bağlı olarak değişmekle birlikte 3-4 saat sürmektedir.

KAYNAKLAR

- 1 Elmas, Ç., “Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)”, *Seçkin Yayıncılık*, Ankara, 22-37.(2003).
- 2 Çelik, H. “Yapay Sinir Ağı Kullanarak El Yazısı Rakam Tanıma”, Doktora Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 2-5, 10-16. (1999).
- 3 Alpaydın, E., Akın, L., Aratma, S., Yağcı, M., “Yapay Sinir Ağları İle Görüntü Tanıma”, *TÜBİTAK Proje, EEEAG-41*, Ankara, 8-15. (1994).
- 4 Alpaydın, E., Kaynak, C., Alimoğlu, F., “Yapay Sinir Ağları İle Elle Yazılmış Sayıların Optik ve Kalem Tabanlı Tanınması”, *TÜBİTAK Proje, EEEAG-143*, Ankara, 7-14. (1997).
- 5 Öztürk, A., “Osmanlıca Karakterlerin Bilgisayar Destekli Tanınması”, Yüksek Lisans Tezi, *Gebze İleri Teknoloji Enstitüsü Fen Bilimleri Enstitüsü*, Gebze, 1-5. (1998).
- 6 Türkmenoğlu, T., “Osmanlıca’ya İlk Adım, Eski Harfleri Okuma Klavuzu”, *Milenyum Yayınları*, İstanbul, 13-57. (2000).
- 7 Erdem, A., Uzun, E., *Gazi Üniv. Müh. Mim. Fak. Der.*, 20 (1) 13-19. (2005).
- 8 Yaman, K., Sarucan, A., Atak, M., Aktürk, N., *Gazi Üniv. Müh. Mim. Fak. Der.* 16 (1): 20-24 (2001).
- 9 A. Buman, C., “Digital Image Processing Laboratories Image Halftoning” *Purdue University*, 11, 1-4 (2004).
- 10 Elmas, Ç., “Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)”, *Seçkin Yayıncılık*, Ankara, 22-37. (2003).
- 11 Elmas, Ç., Sağıroğlu, Ş., Çolak, İ. and Bal, G., “Nonlinear Modelling of a Switched Reluctance Drive Based on Neural Networks”, *MELECON’ 94*, Antalya, (2): 809-812 (1994).
- 12 McCulloch, W. and Pitts, W., “A Logical Calculus Of The Ideas Immanent in Nervous Activity”, *Bulletin Mathematical Biophysics*, (5): 115-137, (1943).
- 13 Hebb, D., “The Organization Of Behavior: A Neuropsychological Theory”, *Wiley & Sons*, New York, 1-368. (1949).

- 14 Rosenblatt, F., "The Perceptron: A Probabilistic Model For Information Storage & Organization in The Brain", *Psychological Review*, (65): 386-408 (1958).
- 15 Rosenblatt, F., "The Perceptron: A Probabilistic Model For Information Storage & Organization in The Brain", *Psychological Review*, (65): 386-408 (1958).
- 16 Widrow, B. and Lehr, M. A., "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", *Proceedings The IEEE*, 78 (9): 1415-1442 (1990).
- 17 Minsky, M. and Papert, S., "Perceptrons: an Introduction to Computational Geometry", *MIT Press*, Cambridge, 1-275 (1969).
- 18 Rumelhart, D. E. and McClelland, J. L., "Parallel Distributed Processing: Explorations in The Microstructure of Cognition", *MIT Press*, Cambridge, 1-1208 (1986).
- 19 Fukuda, T. and Shibata, T., "Theory and Applications Neural Networks for Industrial Control Systems", *IEEE Transactions on Industrial Electronic*, 39 (6): 472-489 (1992).
- 10 Grosberg, B., "Adaptive Pattern Classification And Universal Recoding, II: Feedback, Expectation, Olfaction And Illusions", *Biolog. Cybernetics*, (23): 187-202 (1976).
- 21 Carpenter, G. A. and Grosberg, B., "A Massively Parallel Architecture For A Self-Organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics and Image Processing*, (37): 54-115 (1983).
- 22 Carpenter, G. A. and Grosberg, B., "Art 2: Self- Organization Of Stable Category Recognition Codes For Analog Output Patterns", *Applied Optic*, (26): 4919-4930 (1987).
- 23 Carpenter, G. A. and Grosberg, B., "Art 3 Hierarchical Search: Chemical Transmitters in Self-Organizing Pattern Recognition Architectures", *in Proc. Int. Joint Conf. on Neural Network*, (2): 30-33 (1990).
- 24 Kohonen, T., "Self-Organized Formation Of Topologically Correct Feature Maps", *Biolog. Cybernetics*, 43: 59-69 (1982).
- 25 Werbos, P., "Beyond Regression: New Tools For Prediction And Analysis in The Behavioral Sciences", *Ph.D. Dissertation, Harvard University*, Cambridge, MA, 45-65. (1974).

- 26 Hopfield, J. J., “Neural Network And Physical Systems With Emergent Collective Computational Abilities”, *Proc. Natl. Acad. Sci.*, 79: 2554-2558 (1982).
- 27 Hopfield, J. J., “Neurons With Graded Response Have Collective Computational Properties Like Those Of Two- State Neurons”, *Proc. Natl. Acad. Sci.*, 81: 3088-3092 (1984).
- 28 Kosko, B., “Adaptive Bidirectional Associative Memories”, *Appl. Optics*, 26: 4947-4960 (1987).
- 29 Parker, D., “Learning–Logic”, *Center for Computational Research in Economics and Management Science, M.I.T., Tech. Rep.* TR-47, Cambridge, MA. 76-87. (1985).
- 30 Rumelhart, D. E., Hinton, G. E. and Williams, R. J., “Paralel Distributed Processing”, *Cambridge, MA: The MIT Pres*, 5-23. (1986).
- 31 Hinton, G. E., Sejnowski, T. J. and Ackley, D. H., “Boltzmann machines: Constraint Satisfaction Networks That Learn”, *Carnegie-Mellon University, Dept. of Computer Science, Pittsburgh, PA, Tech. Rep.* CMU-CS-84-119 (1986).
- 32 Hinton, G. E. and Sejnowski, T. J., “Learning and Relearning in Boltzmann Machines, in Paralel Distributed Processing”, *Camdridge, MA: The MIT Press* Chapter 7, 1: 12-34. (1986).
- 33 Meireles, M. R. G., Almeida, P. E. M., and Simoes, M. G., “A Comprehensive Rewiew for Industrial Applicability of Artificial Neural Networks”, *IEEE Transactions on Industrial Electronic*, 50 (3): 585-601 (2003).
- 34 Meireles, M. R. G., Almeida, P. E. M., and Simoes, M. G., “A Comprehensive Rewiew for Industrial Applicability of Artificial Neural Networks”, *IEEE Transactions on Industrial Electronic*, 50 (3): 585-601 (2003).
- 35 Sađırođlu, Ő., BeŐdok, E. ve Erler, M., “Mühendislikte Yapay Zeka Uygulamaları-I, Yapay Sınır Ađları”, *Ufuk Yayıncılık*, Kayseri, 40-42 (2003).
- 36 Vas, P., “Artificial-Intelligence-Based electrical Machines and Drives”, *Oxford University Press*, Newyork, 100-101 (1999).
- 37 Liut, D.A., “Neural Network And Fuzzy Logic Learning And Control Of Lineer And Nonlinear Dynamic Systems”, Doctor of Philosophy, *Virginia State University Polytechnic Institute*, Virginia, 86-93 (1999).

ÖZGEÇMİŞ

Davut Özhan 1980 Batman doğumlu, ilköğretimi Mardin’de bitirmiştir. Mardin Teknik ve Endüstri Meslek Lisesi Elektrik Bölümünden 1997 yılında mezun olmuştur. 1998 Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Öğretmenliği Bölümünü kazanmıştır. 2002 yılında bu bölümden mezun olmuştur. 2003 yılında Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik Eğitimi Bölümünde Yüksek Lisans’a başlamıştır.