

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GENETİK ALGORİTMALAR İLE RADYAL TEMELLİ
FONKSİYON AĞLARININ OPTİMİZASYONU**

Elektronik ve Haberleşme Mühendisi Gül YAZICI

FBE Elektronik ve Haberleşme Anabilim Dalı Haberleşme Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Tülay YILDIRIM

İSTANBUL, 2006

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	vii
ÖNSÖZ	viii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ	1
2. GENETİK ALGORİTMALAR	4
2.1 Temel Genetik Algoritma Operatörleri	5
2.1.1 Kromozom Gösterimi	6
2.1.2 Uygunluk Fonksiyonu	6
2.1.3 Olgunlaşmamış Yakınsama	6
2.1.4 Yavaş Sonlanma	7
2.1.4.1 Doğrusal Ölçekleme	7
2.1.4.2 Üssel Ölçekleme	7
2.1.4.3 Sigma Kesmesi	8
2.1.5 Seçim	8
2.1.5.1 Kesme Seçimi	8
2.1.5.2 Stokastik Evrensel Örnekleme	8
2.1.5.3 Rulet Tekerleği Seçimi	9
2.1.6 Çoğalma	9
2.1.7 Mutasyon	10
2.1.8 Yakınsama	10
2.1.9 Tersten Sıralama ve Yeniden Düzenleme	11
2.1.10 Çift Değerlilik ve Baskınlık	11
2.1.11 Epistasis	11
2.1.12 Aldanma	11
2.2 Genetik Algoritmalar Nasıl Çalışır	12
2.2.1 Şema Teoremi	12
2.2.2 Yapıtaşı Hipotezi	13
2.3 Arama Uzayında Keşif ve Keşfin Kullanımı	13
2.3.1 Bilgi Tabanlı Teknikler	14
2.3.2 Kullanılabilirlik	14
2.4 Genetik Algoritma Türleri	14
2.4.1 Basit Genetik Algoritma	14
2.4.2 Kararlı Durum Genetik Algoritması	15
2.4.3 Düzensiz Genetik Algoritmalar	15

2.4.4	Deme Genetik Algoritması	15
2.5	Genetik Algoritmelerde Öğrenme	15
2.5.1	Michigan Yaklaşımı	15
2.5.2	Pittsburgh Yaklaşımı	16
2.5.3	Yinelemeli Kural Öğrenme	16
2.6	Evrimsel Yapay Sinir Ağları	16
2.6.1	EYSA Bağlantı Ağırlıklarının Evrimi	16
2.6.1.1	İki Tabanlı Gösterim.....	17
2.6.1.2	Gerçek Sayı Kodlamalı Gösterim.....	17
2.6.1.3	Evrimsel ve Geri yayınımlı Eğitim Karşılaştırması.....	17
2.6.1.4	Hibrid Eğitim Yaklaşımı	18
3.	YAPAY SİNİR AĞLARI	19
3.1	Radyal Temelli Fonksiyon Ağları.....	21
3.2	RBF Ağları ile Çok Katmanlı Algılayıcıların Karşılaştırılması	24
3.3	Öğrenme Stratejileri	25
3.3.1	Rastgele Seçilmiş Merkezler	25
3.3.2	Merkezlerin Düzenli Seçimi.....	27
3.3.3	Merkezlerin Denetimli Seçimi.....	28
3.4	Tartışma	30
3.5	Boyut Laneti	31
3.6	Ortogonal En Az Kareler Metodu.....	31
4.	GA-RBF SİMULASYONU.....	34
4.1	İris Veri Kümesi.....	35
4.2	Tiroid Veri Kümesi	37
4.3	E.coli Veri Kümesi	39
4.4	Fetus Veri Kümesi	40
4.5	Cam Veri Kümesi	42
4.6	Lens Veri Kümesi	44
5.	SONUÇLAR.....	47
	KAYNAKLAR.....	49
	ÖZGEÇMİŞ.....	52

SİMGE LİSTESİ

b_j	Muhtemel bir bias
c_i	Merkez vektörü
d	Seçilen merkezler arasındaki uzaklık
E	Anlık değer fonksiyonu
f_i	i. kromozomun uygunluk değeri
M	Merkez sayısı
o_i	Uygunluk değerine karşılık gelen hedef değeri
p_i	Populasyon olasılık fonksiyonu
W_{ij}	i. gizli katman ile j. çıkış katmanı arası dönüşümdeki ağırlık
x	Giriş vektörü
$\varphi(x)$	Aktivasyon fonksiyonu
σ	Gauss genişliği

KISALTMA LİSTESİ

GA	Genetik Algorithms
RBF	Radial Basis Functions
OLS	Orthogonal Least Square Method
LMS	Least Mean Square
EYSA	Evrimsel Yapay Sinir Ağları
GRNN	Generalized Regression Neural Networks

ŞEKİL LİSTESİ

Şekil 2.1	Genetik algoritma akış diyagramı.....	5
Şekil 2.2	Tek noktalı çaprazlama.....	9
Şekil 2.3	Çok noktalı çaprazlama	10
Şekil 3.1	YSA mimarisinin temel elemanları	20
Şekil 3.2	Çok katmanlı ileri ve geri beslemeli ağ yapıları	21
Şekil 3.3	Radyal temelli ağ yapısı.....	23
Şekil 4.1	Algoritmanın çalışma akışı	34
Şekil 4.2	İris çiçeği-genetik optimizasyon sonuçları	37
Şekil 4.3	Tiroid hastalığı-genetik optimizasyon sonuçları.....	38
Şekil 4.4	E.coli bakterisi- genetik optimizasyon sonuçları	40
Şekil 4.5	Fetus gelişimi-genetik optimizasyon sonuçları.....	42
Şekil 4.6	Cam veri kümesi-genetik optimizasyon sonuçları	44
Şekil 4.7	Lens veri kümesi-genetik optimizasyon sonuçları.....	45

ÇİZELGE LİSTESİ

Çizelge 3.1	RBF ağı için, doğrusal ağırlıklar, merkezlerin yeri ve genişliği için adaptasyon formülleri.....	30
Çizelge 4.1	İris veri kümesine ait eğitme ve test örnek sayıları.....	36
Çizelge 4.2	İris veri kümesi için doğru sınıflanan örnek sayıları.....	36
Çizelge 4.3	Tiroid veri kümesine ait eğitme ve test örnek sayıları.....	38
Çizelge 4.4	Tiroid veri kümesi için doğru sınıflanan örnek sayıları.....	38
Çizelge 4.5	E.coli bakterisine ait eğitme ve test örnek sayıları.....	39
Çizelge 4.6	E.coli bakterisi için doğru sınıflanan örnek sayısı.....	39
Çizelge 4.7	Fetus veri kümesi için eğitme ve test örnek sayıları.....	41
Çizelge 4.8	Fetus veri kümesi için doğru sınıflanan örnek sayıları.....	41
Çizelge 4.9	Cam veri kümesine ait eğitme ve test kümesi örnek sayıları.....	43
Çizelge 4.10	Cam veri kümesi için doğru sınıflanan örnek sayıları.....	43
Çizelge 4.11	Lens veri kümesi için eğitim ve test örnek sayıları.....	45
Çizelge 4.12	Lens veri kümesi doğru sınıflanan örnek sayıları.....	45
Çizelge 4.13	Tüm veri kümelerine ait test başarımları ve sigma değerleri.....	46

ÖNSÖZ

Teknolojik gelişmenin önemli boyutlara ulaştığı günümüzde, insanoğlunun kendisini tanımaya yönelik çalışmaları da önemli aşamalar kaydetmiştir. Yapay zeka (artificial intelligence) kavramı ile insanın en önemli özellikleri olan düşünebilme ve öğrenebilme yetenekleri en önemli araştırma konuları durumuna gelmiştir. Özellikle son zamanlarda bilgisayar kullanımının hızla yaygınlaşması sonucunda yapay zeka çalışmaları da bir ivme kazanmıştır. İnsanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanan yapay zeka, aslında programlanmış bilgisayarlara düşünme yeteneği sağlama girişimidir. İnsan gibi düşünen ve davranan sistemlerin geliştirilmesine yönelik olarak 1950’li yıllardan beri süren yapay zeka çalışmaları, bir noktada insanı taklit etmeye yönelik olduğundan mühendislik, nöroloji ve psikoloji gibi alanlara da yayılmıştır.

Yapay zeka çalışmaları kapsamında ortaya çıkan ve bir noktada yapay zeka çalışmalarına destek sağlamakta olan farklı alanlardan bir tanesi de Yapay Sinir Ağları teknolojisidir. Dolayısıyla, yapay zeka alanının bir alt dalını oluşturan YSA teknolojisi öğrenebilen sistemlerin temelini oluşturmaktadır. İnsan beyninin temel işlem elemanı olan nöronu (neuron) şekilsel ve işlevsel olarak basit bir şekilde taklit eden YSA’lar, bu yolla biyolojik sinir sisteminin basit bir simülasyonu için oluşturulan programlardır. Bu şekilde, insanoğluna özgü deneyerek (yaşayarak) öğrenme yeteneğini bilgisayar ortamına taşıyabildiği düşünülen YSA teknolojisi bir bilgisayar sistemine inanılmaz bir “girdi veriden öğrenme” kapasitesi sağlamaktadır ve bir çok avantajlar sunmaktadır.

Evrim teorisinden yola çıkarak optimizasyon problemlerine başarılı çözümler getiren Genetik Algoritmalar ise, yapay sinir ağlarını destekleyici özellikler içermekte ve bu tezde incelenen sınıflama probleminde yapay sinir ağlarına katkı sağlamaktadır. Farklı veri kümeleri ile yapılan simülasyonlarda elde edilen sınıflama başarısının, genetik optimizasyon kullanıldığı zaman arttığı ispat edilmiştir.

Tezi hazırlama sürecimde, her aşamada bana yol gösteren değerli hocam Prof. Dr. Tülay Yıldırım’a, benden desteklerini esirgemeyen ailem ve arkadaşlarıma teşekkürlerimi sunarım.

ÖZET

Yapay sinir ağıları topolojisinin ağı performansına güçlü bir etkisi vardır, ancak optimal konfigürasyonun bulunması önemli bir sorundur. Radyal Temelli Fonksiyon (RBF) ağlarında merkezlerin yeri ve Gauss fonksiyonunun genişliği ağı performansı üzerinde kritik bir öneme sahiptir. Merkezler ve saklı katman genişlikleri kromozamlara kodlanarak bu iki kritik parametrenin Genetik Algoritmalar (GA) ile optimizasyonu hedeflenmiştir.. Genetik algoritma için uygunluk fonksiyonu test başarı oranıdır. Merkezlerin, veri kümesindeki satır değerleri olarak algılanması gereken bu yapıda optimize edilecek parametrelerin alabileceği maksimum ve minimum değerler algoritmanın işlemesi sırasında belirtilmektedir. Algoritma verilen sınır değerleri arasından en uygun çözümü bulmaya çalışmaktadır.

Optimizasyon işlemi tiroid hastalığı, iris çiçeği, e.coli bakterisi, fetus gelişimi, cam (glass) ve lens veri kümeleri üzerinde denenmiştir. Çalışmanın en önemli avantajı, eğitime setinin küçük bir kısmı kullanılarak başarılı bir sonuç vermesidir. Her bir veri kümesinden az sayıda örnek alınarak ağlar eğitilmiş ve tüm veri kümesi ile test edilmiştir. Seçilen veri kümelerinin zor sınıflanan bir karakteristikte olmasına dikkat edilmiş, bu şekilde GA-RBF başarımları daha iyi gözlenmiştir.

Tüm simülasyonların Matlab 7 programıyla gerçekleştirildiği bu yaklaşımda, elde edilen grafikler ve karşılaştırmalı tablolar GA-RBF optimizasyonunun RBF yaklaşımından daha üstün olduğunu ortaya koymuştur. GA algoritmasının çalışması uzun sürede gerçekleşse de, geniş ve zor sınıflanan veri kümeleri için daha cazip sonuçlar sunmaktadır. Simülasyon bölümünde ifade edilen sonuçlar ve karşılaştırmalar, önerilen yaklaşımın kabul edilebilirliğini açıkça ortaya koymaktadır.

Anahtar kelimeler: Genetik Algoritmalar, Radyal Temelli Fonksiyon Ağları, Merkez Seçimi, Gauss Fonksiyon Genişliği.

ABSTRACT

The effects of a neural network's topology on its performance are well known, yet the question of finding optimal configuration remains largely open. In Radial Basis Function (RBF) networks, placement of centers is said to have significant effect on the performance of network. In this paper, centers and widths of hidden layer are coded in a chromosome and these two critical parameters are determined with the optimization by Genetic Algorithms (GA). The maximum and minimum limit values of these parameters are defined in the algorithm whilst centers should be perceived as the number of lines in dataset.

This treatise aims to classify the test set with high accuracy while minimum number of instance is chosen from the train set. Iris plant, thyroid disease, escherichia coli bacteria, fetus, glass and lens datasets are used to compare the success between GA-RBF and RBF method.

The final conclusion is that GA-RBF approach is more effective than RBF and some numerical results indicate the applicability of the proposed approach, which all simulations realized by Matlab 7. Nevertheless, GA took a long training time to achieve these results. But for a large number of applications and rough datasets, genetic approach is an attractive solution for the design of efficient artificial neural networks.

Keywords: Genetic algorithms, radial basis functions, selection of centers, gauss function spread value.

1. GİRİŞ

Ünlü matematikçi J. HADAMARD “Gerek matematikte, gerek başka alanlarda buluş ve icatlar düşüncelerin bir araya getirilmesiyle gerçekleşir” demiştir. Bu söz bize genetik algoritmaların (GA) nasıl ortaya çıktığını açıklayabilir. Doğadan esinlenerek tasarlanan ve geliştirilen genetik algoritmalar, evrim sürecinin hesaplamalı modellerinin esas alındığı bilgisayar tabanlı arama ve optimizasyon problemlerinin çözümünde başarıyla kullanılan ve popülerliği her geçen gün artan adaptif yöntemlerdir. Genetik algoritmalar, bilgisayar üzerinde oluşan bir evrim şeklidir. Genetik algoritmaların amacı hem problemleri çözmek hem de evrimsel sistemleri modellemektir. Değişik planlama teknikleri, bir fonksiyonun optimizasyonu veya ardışık değerlerin tespitini içine alan bir çok problem tipi için çözüm geliştirmektedir. Genetik algoritma ile oluşturulan seçim, doğal topluluklara benzer bir şekilde bilgisayar hafızasına depo edilmiş kromozomlar üzerinde icra edilmektedir. Bilgisayara uyarlama tabiatla kıyaslanamayacak kadar basitleştirilmesine rağmen, genetik algoritmalar karışık olmasının yanında hayret uyandıracak kadar da ilginç yapıya sahiptirler (Turgut ve Arslan, 2001).

Geleneksel arama yöntemleriyle birlikte kullanımı mümkün olan genetik algoritmalar bir kısım öğrenme işlemleri için güçlü arama yöntemleri de önermektedir. Genetik algoritmalar kullanılarak yapay sinir ağlarındaki ağırlık, fonksiyon, katman sayısı gibi pek çok parametre belirlenebilir. Bu çalışmada genetik algoritmalar, bir yapay sinir ağı modeli olan Radyal Temelli Fonksiyon (Radial Basis Function-RBF) ağlarının gauss fonksiyonu genişliğini ve merkezlerinin pozisyonlarını bulmak için kullanılmıştır.

Hibrid yapıli RBF modeli, ileri beslemeli üç katmanlı bir ağıdır. RBF ağlarının klasik çok katmanlı algılayıcılardan üç konuda farkı bulunmaktadır: saklı katman ve çıkış katmanı arasında yalnızca bir eğitim kümesi vardır, aktivasyon fonksiyonları standart değildir ve öğrenme denetimli veya denetimsiz gerçekleşebilir (Burdsall ve Carrier, 1997).

Literatürde, genetik algoritmalar ve RBF ağlarının kullanıldığı çeşitli çalışmalar mevcuttur. Barreto, Barbosa ve Ebecken (2002), RBF ağlarının konfigürasyonu için genetik algoritmalar kullanılarak, klasik hesaplama problemlerini kontrol eden yeni bir çaprazlama operatörü tanımlamışlardır. RBF ağları parametrelerinin GA ile nasıl tanımlanacağı sorusuna yönelik olan çalışmada ise (Lacerda vd., 2001), önerilen GA, Hermit polinomal yaklaşım problemine uygulanmaktadır. Zuo, Liu ve Ruan (2004) tarafından yapılan çalışma RBF ağının saklı katmanını eğitmek için, GA kullanılarak ses değişimi üzerinedir. Topolojiyi sabitlemek ve

tüm özgür parametreleri hesaplamak için optimizasyon aracı olarak kullanılan genetik algoritma yaklaşımı (Sheta ve Jong, 2001), çalışmada zaman-serisi tahmininde kullanılmaktadır. Mak ve Cho'nun (1998) çalışmalarında, saklı nöron sayısı sabitlenmiştir ve genetik algoritma yalnızca merkezlerin yerlerini belirlemektedir. Merkezlerin genişliği ve çıkış katmanı ağırlıkları k-en yakın komşu yaklaşımı ile hesaplanmıştır. Whitehead ve Choate (1995), merkezlerin sayısını belirlemiş ancak pozisyonlarını ve genişliklerini farklı ve ilginç bir yolla geliştirmişlerdir. Her bir bireye ağı kodlamak yerine, radyal temelli fonksiyon ağı oluşturmak için her biri bir merkez olacak şekilde tüm kromozom kümesini birleştirmişlerdir. Her bir birey yalnızca bir saklı katmanı kodlar ve tüm populasyon yalnızca bir radyal temelli fonksiyon ağını ifade eder. Bir başka fikir, genetik algoritmaları bir destek aracı olarak kullanarak konfigürasyon işlemini hibridize etmektir. Chen, Wu ve Luk bu çalışmalarında (1999), radyal temelli fonksiyon ağları için iki seviyeli öğrenme yöntemini tanıtmışlardır. Algoritmanın iki ana parametresi dış seviyede GA tarafından optimize edilirken, düzenlenmiş bir ortogonal en küçük kareler (regularized orthogonal least squares-ROLS) algoritması, iç seviyedeki radyal temelli fonksiyon ağlarını kurmak için uygulanmıştır. Aslında en genel yaklaşım çıkış katmanındaki ağırlıkların cebirsel ya da azalan eğim yöntemi ile hesaplanırken, genetik algoritmaların ağ topolojisi ile merkezlerin yeri ve genişliğini tespit etmede kullanılmasıdır. Ancak elbette bu yöntemler arasında da pek çok fark bulunabilir. Örneğin Whitehead ve Choate'nın (1993) çalışmalarında, RBF merkezlerini belirlemek için aralık-doldurma eğrilerini geliştiren genetik yaklaşımdan bahsedilmektedir. Bu yaklaşımdaki temel fikir, m-boyutlu bir uzaydan kromozomun kodlandığı boyutsuz bir uzaya merkezlerin yerini tasarlamaktır. Bir diğer örnek de Maillard ve Gueriot'un (1997) çalışmasıdır. Burada temel fonksiyonlar Gauss ile sınırlı değildir ve evrime odaklanmıştır.

Radyal Temelli Fonksiyon ağlarında merkezlerin yeri ve gauss fonksiyonunun genişliği ağı performansı üzerinde kritik bir öneme sahiptir. Bu tez çalışması, genetik algoritmalar kullanarak RBF ağından en iyi sınıflama sonucu almak için merkezlerin ve gauss fonksiyon genişliğinin belirlenmesi esasına dayanır. Bu parametreler kromozomlara kodlanarak, genetik algoritmalar yardımı ile yapılan optimizasyon sonucunda sınıflama problemi ele alınmıştır. İlk bölümde genetik algoritmalar geniş olarak yer verilmiştir. Genetik algoritmalarda kullanılan temel ilkeler açıklanmış ve bu algoritmaların nasıl çalıştığı anlatılmıştır. Genetik algoritma türleri ile genetik algoritmalarda öğrenmeden bahsedilerek ikinci bölüme geçilmiştir. Yapay sinir ağlarının anlatıldığı bu bölümde, ağırlıklı olarak teze konu olan radyal temelli fonksiyon ağlarına ve temel özelliklerine değinilmiştir. Üçüncü bölümde, Matlab 7.0 programı kullanılarak yapılan bilgisayar simülasyonu anlatılmıştır. Algoritmanın genel

özellikleri belirtilmiş, genetik algoritma parametrelerinin nasıl kullanıldığı açıklanmıştır. Optimizasyon işlemi tiroid hastalığı, iris çiçeği, e.coli bakterisi, fetus gelişimi, cam (glass) ve lens veri kümeleri üzerinde denenmiştir. Çalışmanın en önemli avantajı, eğitme setinin küçük bir kısmı kullanılarak başarılı bir sonuç vermesidir. Her bir veri kümesinden az sayıda örnek alınarak ağlar eğitilmiş ve tüm veri kümesi ile test edilmiştir. Her bir veri kümesine ait elde edilen grafikler ve karşılaştırmalı tablolar verilerek genetik algoritma kullanarak yapılan RBF optimizasyonunun başarımını göstermektedir. Son bölümde ifade edilen sonuçlar ve karşılaştırmalar, önerilen yaklaşımın kabul edilebilirliğini açıkça ortaya koymaktadır.

2. GENETİK ALGORİTMALAR

Genetik algoritmalar arama ve optimizasyon problemlerinin çözümünde başarıyla kullanılan ve popülerliği her geçen gün artan adaptif yöntemlerdir. Son yıllarda optimizasyonun önemi, bir çok büyük ölçekli birleştirici optimizasyon (combinatorial optimization) probleminin ve yüksek kısıtlı mühendislik problemlerinin günümüz bilgisayarları ile yaklaşık olarak çözülebilmelerinden dolayı daha da artmıştır. Genetik algoritmaların (GA) amacı böyle kompleks problemlerdir. Bu problemler, olasılıklı algoritmalar sınıfına ait olmakla birlikte rastgele algoritmalarından çok farklıdır (İşçi ve Korukoğlu, 2003).

Genetik Algoritmalar stokastik bir arama yöntemidir. GA, Darwin'in en iyi olan yaşar prensibine dayalı olarak biyolojik sistemlerin gelişim sürecini modellemektedir. GA tekniği, çözüm uzayının büyüklüğüne rağmen iyi bir çözüme kısa zamanda yakınsamaktadır. Genetik algoritmalarda canlı organizmalardan esinlenilmiştir. Canlı topluluklar nesiller boyunca, ilk defa Charles Darwin'in "*Türlerin Kökeni*" adlı kitabında açıkça kaleme aldığı şekilde, doğal seçim süreci ve güçlülerin hayatta kalması prensiplerine göre gelişmişlerdir (Beasley vd., 1993a).

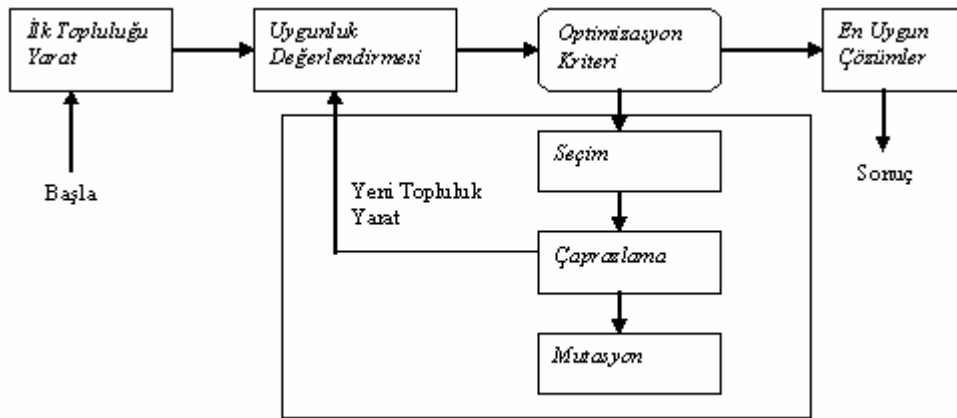
Genetik algoritmalar ismi ilk defa J.D Bagley tarafından 1967 tarihli doktora tezinde kullanılmıştır. Evrim sürecini temel alan GA'nın temel prensipleri ise ilk defa J. Holland tarafından 1975 yılında basılan "*Doğal ve Yapay Sistemlerde Uyarlanma*" isimli kitabında ortaya konulmuştur. Makine öğrenmesi (Machine Learning) konusunda çalışmalar yapan Holland, evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Genetik Algoritma ilk ismini biyoloji, ikinci ismini ise bilgisayar biliminden almaktadır. Sadece bir tane mekanik yapının öğrenme yeteneğinin geliştirilmesi yerine, böyle yapılardan oluşan bir topluluğun çoğalma, çiftleşme, değişim vb. genetik süreçlerden geçirilerek, başarılı (öğrenebilen) yeni bireylerin oluştuğu görülmüştür. Holland'ın çalışmalarının sonuçlarını açıkladığı kitabının 1975'de yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar yada kısaca GA olarak yerleşmiştir. Ancak 1985 yılında Holland'ın öğrencisi olarak doktorasını veren David E.Goldberg adlı inşaat mühendisi 1989'da konusunda bir klasik sayılan kitabını yayınlıncaya kadar, Genetik Algoritmaların pek pratik yararı olmayan araştırma konusu olduğu düşünülmekteydi. Halbuki Goldberg'in gaz borusu hatlarının denetimi üzerine yaptığı çalışma ona sadece 1985 National

Science Foundation Genç Araştırmacı ödülünü kazandırmakla kalmayıp, Genetik Algoritmaların pratik kullanımının da olabilirliğini kanıtlamıştır (Turgut ve Arslan, 2001).

Temelleri açıkça ortada olan evrim sürecini taklit eden GA, gelişime açık bir sahadır. GA, problemin muhtemel çözümünü temsil eden dizilişler olan bireyler topluluğu üzerinde işlem yapar. Bu dizilişlerden her biri kromozom olarak anılır. Her bireye sağladığı çözümün başarı derecesine göre bir uygunluk değeri atanır. Uygunluk değeri yeterince yüksek olan bireylerin, önceden belirlenmiş bir stratejisi olan seçim işlemi sonunda çaprazlama işlemiyle çoğalması sağlanarak yeni nesiller elde edilir. Elde edilen her yeni neslin içeriği, atalarının sahip olduğu uygunluk değeri artırıcı özellikleri daha yüksek oranda barındırır. Böylelikle uygun değerli parametreler nesil sayısı arttıkça toplulukta daha yaygın hale gelir. Eğer GA iyi tasarlanmışsa topluluk uygun bir çözüme yakınsayacaktır. GA'nın avantajı geniş bir uzayda başarıyla uygulama geliştirme imkanı sağlaması ve yöntemin gürbüzlüğüdür. GA, global optimumun belirlenmesini garanti etmese de kabul edilebilir bir çözümü yeterince hızlı bulabilir. GA daha etkin olarak sonucu bulmada veya hibrid kullanımla yöntemi iyileştirmede kullanılmaktadır (Beasley vd., 1993a).

2.1 Temel Genetik Algoritma Operatörleri

Şekil 2.1'de gösterilen standart GA'nın çalıştırılabilmesi için tespit edilmesi gereken temel noktalar; kromozom gösterimi, seçim stratejisi, çoğalma operatörleri, ilk topluluğun yaratılması, sonlandırma kriteri ve değerlendirme fonksiyonudur. Bu noktaların belirlenmesinde probleme uygunluğu gözetilmelidir.



Şekil 2.1 Genetik algoritma akış diyagramı

2.1.1 Kromozom Gösterimi

Problemin muhtemel çözümünün bir dizi parametre ile temsil edildiği ilkesine uygun olarak öncelikle parametre sayısının belirlenmesi gerekir. Parametreler gen olarak anılırlar ve birleşerek kromozomları oluştururlar. Kromozomların gösteriminde belirli bir alfabe kullanılır. Bu alfabe sembollerden, iki tabanlı sayılar olan $\{0,1\}$ 'den, tamsayılardan, gerçel sayılardan ve matrislerden oluşabilir. GA'nın yaygınlaşmaya başladığı ilk zamanlarda, parametre değerleri için iki tabanlı gösterim daha uygun görülüp yaygın olarak kullanılmıştır. Michalewicz, çalışmalarıyla arama uzayının doğasıyla uyumlu gösterimlerinin daha iyi sonuçlar sağladığını göstermiştir. Bu nedenle, fonksiyon optimizasyonunda, kromozomların iki tabanlı gösterime kıyasla gerçel sayılarla ifade edilmesi daha verimlidir.

Gen bilimi terminolojisinde, belirli bir kromozomun gen içeriğine genotip adı verilir. Genotip bir organizmayı teşkil etmekte gerekli olan bilgiyi içerir. Teşkil edilmiş görüntüye de fenotip denir. Aynı terimler genetik algoritmalar için de kullanılabilir. Örneğin bir tasarımda, belirli bir tasarımı temsil eden parametreler genotipi oluştururken gerçekleştirilen tasarım fenotip olur. Kromozom uygunluk değeri ise, fenotipin başarımına dayanır. Bu da uygunluk fonksiyonu kullanılmak suretiyle kromozomdan hesaplanabilir (Michalewicz, 1994).

2.1.2 Uygunluk Fonksiyonu

Çözümü istenen her problem için uygunluk fonksiyonu belirlenmelidir. Uygunluk fonksiyonu, belirli bir kromozomun çözüme yakınlığının göstergesi olan uygunluk değerinin hesaplanmasında kullanılır. Bu topluluk içerisinde uygunluk değeri yüksek olan bireyler seçilerek, en yüksek uygunluklu birey, uygunluğu düşük olan bireyin yerini almaktadır (Turgut ve Arslan, 2001). Yığındaki her birey ikili düzende veya tamsayı olarak kodlanmaktadır. Bu bireyler değerlendirme aşamasında deşifre edilerek belirli amaç fonksiyonu yada fonksiyonlarında gösterdikleri performanslarına (uygunluklarına) göre değerlendirilmektedir. Bireyler uygunluk değerlerinin yüksekliğine göre daha büyük bir olasılıkla seçilmektedir (Uçaner ve Özdemir, 2002).

2.1.3 Olgunlaşmamış Yakınsama

İlk topluluk rastgele değerlerle yaratıldığından bireylerin uygunluk değerleri ile belirli bir lokusa ait genler arasında ciddi bir farklılık olacaktır. Topluluk yakınsadıkça uygunluk

değerlerinin varyansı azalır. Çözüme yakınlığının göstergesi olan uygunluk değerinin değişimine bağlı olarak karşılaşılabilecek sorunlar da vardır. İlki olgunlaşmamış (premature, erken) yakınsama ve ikincisi yavaş sonlanmadır (slow finishing) (Öztürk, 2002).

Holland'ın şema teorisi, bireylere uygunluk değerleriyle orantılı çoğalma fırsatı tanınmasını önerir. Ancak topluluk nüfusunun sonlu olması zorunluluğu nedeniyle olgunlaşmamış yakınsama gerçekleşebilir. GA'nın nüfusu, sınırlı topluluklarda etkin çalışabilmesi için bireylerin kazanacağı çoğalma fırsatı sayısının ne çok fazla ne de çok az olacak şekilde denetlenmesi gerekir. Uygunluk değeri ölçeklenerek, erken nesillerde, aşırı uygunluk değerli bireylerin toplulukta hakimiyet kurması engellenir. Bu şekilde genetik algoritmanın uzayın yalnızca bir bölümünde arama yapması engellenmiş olur ve verimli bir sonuç elde edilir.

2.1.4 Yavaş Sonlanma

Olgunlaşmamış yakınsamaya karşıt sorun yavaş sonlanmadır. Epey nesil geçmesine rağmen topluluk yaklaştığı halde global minimumu konumlandırmayabilir. Ortalama uygunluk değeri yeterince yüksek değerli olup en iyi bireyin uygunluk değerine yakınsamış olabilir. Kullanılan yöntemlerle topluluğun etkin uygunluk değerinin varyansı artırılır [1].

2.1.4.1 Doğrusal Ölçekleme

i. kromozomun uygunluk değeri f_i ile karşılık gelen hedef değeri o_i arasındaki doğrusal ilişki (2.1) denklemiyle tanımlanır. a ve b sabit değerleri tespit edilirken f_i ve o_i ortalamaları eşitlenmeye çalışılır (Man vd., 1997).

$$f_i = a o_i + b \quad (2.1)$$

2.1.4.2 Üssel Ölçekleme

A.M Gillies'in önerdiği yöntemde, probleme bağlı değişkenlik arz eden ve olgunlaşmamış yakınsamayı önleyecek tazda belirlenmesi gereken bir “k” üssel ölçeklemede kullanılır. Gillies, tanımı (2.2) eşitliğiyle verilen yöntemi kullanırken $k=1.005$ almıştır (Man vd., 1997).

$$f_i = o_i^k \quad (2.2)$$

2.1.4.3 Sigma Kesmesi

S. Forrest'ın önerdiği, doğrusal ölçeklemede karşılaştırılması muhtemel negatif uygunluk değerini önleyecek şekilde ölçekleme yapan yöntemdir. c amaca uygun olarak seçilmiş sabit bir değer ve σ , topluluğun uygunluk değerinin standart sapması olmak üzere tanım (2.3) eşitliğiyle verilmiştir (Man vd., 1997).

$$f_i = o_i - (o_{\text{ortalama}} - c\sigma) \quad (2.3)$$

2.1.5 Seçim

GA'da çözüme giden yol, bireylerin uygunluk değerinin artışını sağlayan gen içeriğinin edinilmesinden geçer. Bu sebeple yeni nesili oluşturacak bireylerin seçimi hayati önemdedir. Bu operatör yeni topluluk içerisinde, uygunluğu yüksek bireylerin bulunmasını sağlamaktadır. Bireylerin uygunluk değerleri esas alınarak seçim işlemi yapılmaktadır. Seçim işlemi topluluk içerisinde uygunlukları düşük olan bireyleri eleyip, elenenler yerine uygunlukları yüksek olan bireylerin birkaç adet kopyasını yaparak tamamlanmaktadır. Önerilen çeşitli temel yaklaşımlar ve çeşitlemeler mevcuttur. Sıklıkla kullanılan stratejiler, kesme seçimi, rulet tekerleği ve stokastik örneklemedir.

2.1.5.1 Kesme Seçimi

“En Güçlüler Yaşar” prensibinden yola çıkarak uygunluk değerine göre büyükten küçüğe doğru sıralanan bireylerden belirlenen sayıda en yüksek değerli bireyler seçilir. Diğerleri yok edilir.

2.1.5.2 Stokastik Evrensel Örnekleme

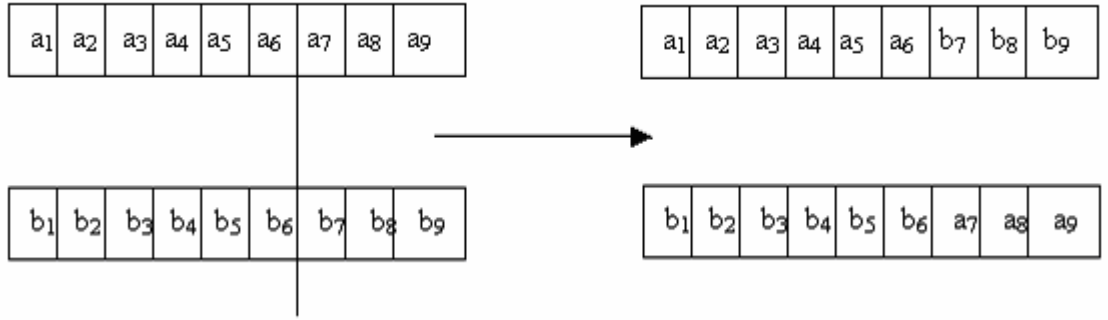
James Baker (1987) tarafından önerilen yöntemde rulet stratejisine benzer yaklaşımla, bireyler bir doğru üzerine yerleştirilir. Seçilecek birey sayısına eşit sayıda işaretçinin, doğru üzerine eşit aralıklarla yerleştirilmesiyle örnekleme yapılır. Örnek olarak, seçilecek birey sayısı $N_{pointer} = 6$ olduğunda, örnekleme periyodu $1/N_{pointer} = 0.167$ olur ve ilk işaretçinin yeri $[0, 1/N_{pointer}]$ aralığında olmak koşuluyla rasgele seçilir [2].

2.1.5.3 Rulet Tekerleği Seçimi

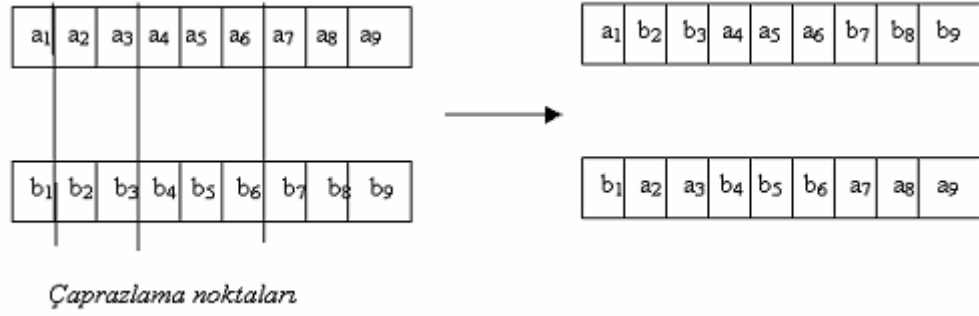
Stokastik bir yöntemdir. Bireyler uygunluk değerleriyle orantılı uzunluklarla, ardışık olarak bir doğru üzerine veya her bir diliminin alanı uygunluk değeriyle orantılı olacak şekilde rulet tekerleği üzerine yerleştirilir. Üretilen rastgele sayının aralığının sahibi birey seçilir. Arzu edilen birey sayısına ulaşılan dek işlem tekrarlanır (Rooij vd., 1996).

2.1.6 Çoğalma

Biyolojik terim olarak çaprazlama genel olarak, bir bireyin içerisindeki değişimi kastetmesine rağmen, çaprazlama terimi burada bireyler arasındaki benzer alt kromozomların değişimlerini ifade etmektedir. Genetik algoritmada çaprazlama iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. İki ebeveyn arasında seçilmiş olan sitelerdeki genlerin yerleri değiştirilerek çaprazlama işlemi tamamlanmaktadır. En çok kullanılan çaprazlama çeşitleri bir noktalı çaprazlama, iki noktalı çaprazlama, üniform çaprazlama ve sıralı çaprazlamadır (Turgut ve Arslan, 2001). Şekil 2.2 ve 2.3’de görüldüğü gibi, baş ve kuyruk dizilerinin değiş tokuşu sonrasında birleştirilen diziler yeni neslin iki ferdi olarak kromozom havuzuna kaydedilir.



Şekil 2.2 Tek noktalı çaprazlama



Şekil 2.3 Çok noktalı çaprazlama

2.1.7 Mutasyon

Genetik Algoritmalar sadece seçim ve çaprazlama operatörlerinden meydana gelselerdi çok hızlı bir şekilde yerel bir optimuma ulaşabilirlerdi. Toplulukta çeşitlilik yaratabilmek, çaprazlama sonucunda kaybolabilen iyi özellikleri geri kazanabilmek ve genel en iyiye ulaşabilmek için bireylerdeki kodlar belli bir olasılık ile değişime (mutasyon) uğratılmaktadır. GA'nın bu işleyişi önceden belirlenen bir durdurma koşulu sağlanıncaya kadar devam etmektedir (Uçaner ve Özdemir, 2002). Rasgele seçilen kromozomdaki bir veya birkaç geni rasgele değişikliğe uğratan mutasyon operatörünün nesil başına uygulanma oranının düşük olması tavsiye edilir. Sınırlı bir topluluk üzerinde çalışıldığında, toplulukta birkaç genetik bilginin erkenden kaybolma ihtimali bulunmaktadır. Örnek olarak, bir kromozomu oluşturan genlerin tamamı 0 yada 1 olabilmektedir. Böyle bir kromozomu çaprazlama operatörü ile değiştirmek mümkün olmamaktadır. Çaprazlama vasıtasıyla üretilmeyen uygunluk değeri yüksek kromozomların, mutasyon vasıtasıyla üretmek mümkün olmaktadır. Bunun yanında uygunluk değeri oldukça yüksek olan kromozomları bozma ihtimali de bulunmaktadır (Turgut ve Arslan, 2001). İki tabanlı gösterimde, genin alabileceği değer $\{0,1\}$ ile kısıtlı olduğundan 0, 1'e ; 1, 0'a dönüşür. Gerçek sayılı gösterimde ise, gendeki değişim, rasgele belirlenen bir sayıyla yerdeğişimine bağlı olabileceği gibi mevcut değere mutasyon adımı olarak anılan küçük ilavelerle de gerçekleştirilebilir.

2.1.8 Yakınsama

GA'nın doğru gerçekleştirildiği uygulamalarda, kromozom topluluğundaki en iyi ve ortalama uygunluk değerleri, bireylerin, evrim sonucu gelişimiyle birbirine ve global optimuma yakınsar. Uygunluk kriterine sağlayan birey yakınsamıştır denir. Topluluk ortalamasının en

iyi bireyin uygunluk değerine yakınsadığı durumda topluluk yakınsamış olur. Topluluk ve kromozom yakınsamasından başka, bir genin yakınsaması da tanımlanmıştır. Bir nesildeki kromozomların belirli bir lokusu %95 oranında aynı gene sahipse gen yakınsamıştır denir (Dejong, 1980).

2.1.9 Tersten Sıralama ve Yeniden Düzenleme

Genlerin sıralanışı çok önemlidir. Sıralamayı değiştirerek arama uzayını genişletmek için bir operatör kullanılır. Bu tersten sıralama operatörü, bir kromozom üzerindeki genlerden rasgele belirlenmiş iki lokus arasında kalanları ters sırayla yerleştirir.

2.1.10 Çift Değerlilik ve Baskınlık

İleri hayat formlarında, kromozomlar ikili sarmal düzenindedir. Genler iki şerit üzerine kodlanmıştır. Birbirinin alternatifi iki genin kodladığı yapı, çift değerli (diploid) kromozom adıyla anılır. Bugüne kadar olan GA çalışmaları ise tek şerit üzerine kodlanmış genlerle gerçekleştirilmiştir. Tek değerli yapı ise haploid kromozom adıyla anılır. Çift değerliliğin sağlayabileceği faydalar olmasına karşın, programlama ve işlem kolaylığı sağlamasından dolayı haploid yapı tercih edilmiştir. Zamana bağlı değişimin söz konusu olabileceği ortamlarda farklı iki çözümü barındıran diploid kromozomlar avantajlıdır. Aynı parametreyi kodlayan genlerden biri baskın diğeri çekinik olacaktır ve ortamdaki değişimle genler de baskınlık/ çekiniklik özelliğini değiştirebileceklerdir. Çift değerlilik, gende evrim sürecinden daha hızlı değişim sağlar (Öztürk, 2002).

2.1.11 Epistatis

Genlerarası etkileşim epistasis denilmektedir. Bir genin uygunluk değerine katkısı, diğer genlerin sahip olduğu değerlere bağlıdır. Epistasis çok fazla ise genetik algoritmalar verimli olmayacaktır. Çok düşük olduğunda ise diğer yöntemlerin başarımı genetik algoritmalara göre yüksek olacaktır.

2.1.12 Aldanma

Evrin süreci işledikçe global optimumu sağlayacak olan şemaların veya yapıtaşlarının toplulukta görülme sıklığı artacaktır. Bu optimal şemalar, çaprazlama operatörüyle nesiller geçtikçe bir araya toplanır ve global optimum sonucu sağlar. Global optimumu bulunmasına katkı sağlamayacak şemaların görülme sıklığının artışı ıraksamaya sebep olacaktır. Bu sonuç, aldanma olarak bilinir. Aldanma için epistasis gerekli fakat yeterli değildir.

2.2 Genetik Algoritmalar Nasıl Çalışır?

Genetik algoritmaların çalışma prensibini açıklamaya yönelik pek çok çalışma mevcuttur. Ancak evrim geçirerek problem çözen bilgisayar programları olan genetik algoritmaların başarısını kısmen izah edebilen iki ekol vardır. Bu yaklaşımlar, “Şema Teoremi” ve “Yapıtaşı Hipotezi”dir

2.2.1 Şema Teoremi

GA çalışmasını açıklamaya yönelik pek çok çalışma yapılmıştır. Bu çalışmalardan en çok bilineni John Holland tarafından ortaya atılan şema (schema ya da schemata) kuramıdır. Şema ikili dizileri göstermek için kullanılan bir gösterimdir. Holland bu kuramı ikili diziler üzerinde temel bir genetik algoritma için tanımlamıştır. Bu yöntemle göre genetik algoritma iyi yapıları ortaya çıkarır, çoğaltır ve birleştirir. Holland’ın şema teoremi hala genetik algoritmaların başarısını açıklamak için en başarılı teorem olarak kullanılır. Genel olarak bir genetik algoritma ikili kodlama (binary strings) sistemi üzerinde çalışır. Bu kodlamalar doğal sistemlerdeki kodlamalar ile ilgilidir. Genetik algoritmalar doğal genetiklerden kelimeleri ödünç alır. Genetik algoritmalarda oluşan başarılı bireyler incelenirse, bu bireyler arasındaki benzerlikler bulunabilir (İşçi ve Korukoğlu, 2003). Bu benzerliklerden yola çıkarak şemalar oluşturulabilir. İkili dizi kodlaması için aşağıdaki yöntem önerilebilir: 0,1 ve * (“ * ” o konumda 0 veya 1 olmasının önemsiz olduğunu gösterir). Örnek olarak ikinci biti 1, dördüncü ve altıncı bitleri 0 olan çözümlerin başarılı olduğu bir toplumda şu şema oluşturulabilir:

* 1 * 0 * 0

Bu şemaya uygun aşağıdaki ikili diziler yazılabilir:

010000, 010010, 011000, 011010, 110000, 110010, 111000, 111010

Görüldüğü gibi şemaların katılması ikili dizilerle gösterilen arama aralığını büyötmektedir. Arama aralığının büyümesinin sonucun bulunmasını zorlaştırması beklenir ancak durum böyle değildir.

Şema teoremi, GA'nın gücünü şablonların işlenme şekli ile izah eder. Kromozom topluluğunun bireylerine bir sonraki nesli oluşturmak üzere çoğalma fırsatı verilir. Bu işleme 'Çoğalma Denemesi' denir. Her bireyin kazanacağı fırsat sayısı, uygunluk değerinin yüksekliğiyle doğru orantılı olarak değişir. Böylelikle uygunluk değeri yüksek bireyler, sonraki nesile daha fazla gen aktarımında bulunurlar. Yüksek uygunluk değerinin kaynağı, sahip olduğu iyi bir şablonun varlığı kabul edilir ve yeni nesile aktarılan iyi şablonların çözüme ulaşma ihtimalini arttırdığı düşünölür.

Holland, arama uzayını keşfetmenin en uygun yolunun, kromozomların sahip oldukları uygunluk değerleriyle orantılı çoğalma denemesi fırsatı kazanmaları olduğunu göstermiştir. Her bir nesilde etkin olarak işlenebilecek şablon sayısı, n topluluk nüfusu olmak üzere, n^3 mertebesindedir. Üstü kapalı koşutluk (implicit parallelism) olarak bilinen özellik, GA'nın başarımının kısmi açıklamasıdır (Dejong, 1980).

2.2.2 Yapıtaşı Hipotezi

Goldberg'e göre, GA'nın gücü iyi yapıtaşları bulabilmesinden kaynaklanmaktadır. Yapıtaşları, kısa tanımlayıcı uzunluklu, birleştirildikleri takdirde başarıyı artırma eğiliminde olan uyumlu dizilişlerdir. Başarılı kodlama, ilişkili genlerin kromozom üzerindeki konumlarının yakın olduğu halde genlerarası etkileşiminin az olmasının sağlandığı durumda yapıtaşlarının oluşumunun teşvik edildiği kodlamadır. Bir genin toplam uygunluk değerine katkısı diğer genlerden bağımsız olsaydı, problemin çözümü, sırasıyla her gen için tepe-tırmanma yönteminin uygulanmasıyla bulunabilirdi ki, bu genelde mümkün olmayan bir çözüm yoludur (Beasley vd., 1993a).

2.3 Arama Uzayında Keşif ve Keşfin Kullanımı

Global maksimumun bulunması için etkin optimizasyon algoritmasının kullanması gereken iki teknik, arama uzayının yeni ve bilinmeyen bölgelerini araştırmak üzere yapılan keşif ve daha önce tetkik edilen noktalardan elde edilen bilginin daha iyi noktalar bulmak üzere

kullanılmasıdır. İyi bir arama algoritması, çelişen iki gereklilik arasında bir denge noktası bulmalıdır.

Holland göstermiştir ki, GA keşif ve keşfin kullanımını aynı anda ve en uygun şekilde birleştirmektedir. Teoride doğru olmasına karşın, uygulamada kaynağı Holland'ın basitleştirici kabulleri olan kaçınılmaz sorunlar mevcuttur. Bu kabuller:

- 1) Topluluk nüfusu sonsuzdur.
- 2) Uygunluk fonksiyonu, çözümün işe yararlılığı için doğru göstergedir.
- 3) Kromozomlardaki genlerarası etkileşim bariz değildir.

Birinci kabulün, uygulamada gerçekleştirilmesinin imkansızlığına bağlı olarak GA stokastik hataya açık olacaktır. Test fonksiyonlarının nispeten kolayca sağladığı ikinci ve üçüncü kabullerin, gerçek problemlerde sağlanması daha güç olabilir (Beasley vd., 1993a).

2.3.1 Bilgi Tabanlı Teknikler

Araştırmalarda, geleneksel yöntemler kullanan GA üzerinde yoğunlaşılmasıyla beraber arama uzayına ait bilgi kullanan göreve özel başka operatörler de tasarlanabilir. Göreve özel genetik algoritma başarımı yükselir. Probleme özel bilginin çaprazlama işleminde elverişli bir şekilde katıştırılabileceği ifade edilmektedir. Arama uzayı bilgisi, uygunluk değeri belirgin derecede düşük, kısıtları ihlal eden kromozomların çoğalmasının engellemesinde kullanılabilir. Böylece başarımı düşük bireylerin işlemleriyle vakit kaybedilmemiş olur. Arama uzayı bilgisi, iyi noktalar etrafında keşfi yoğunlaştıracak şekilde yerel iyileştirme operatörlerin tasarımı için kullanılabileceği gibi daha uygun noktalar civarında arama yapmak üzere kromozom topluluğunun ilk değer atamasında da kullanılabilir (Beasley vd., 1993b).

2.3.2 Kullanılabilirlik

Geleneksel GA uygulamalarının çoğunluğu, fonksiyonların sayısal optimizasyonunda yoğunlaşmıştır. Süreksizlik içeren, çok tepeli, gürültülü verilerin ve fonksiyonların optimizasyonunda diğer yöntemlerden daha başarılı olduğu gösterilmiş olan GA, rastgele verilerin modellenmesi için çok uygundur [1].

2.4 Genetik Algoritma Türleri

Literatürde çeşitli ön isimlerle anılan genetik algoritma türleri hakkında kısaca bilgi verilmiştir.

2.4.1 Basit Genetik Algoritma

İşleyişi, şekil 2.1' e uygun olan ve bir neslin çoğalarak yeni nesli oluşturduğu, örtüşmeyen toplulukların varlığına dayalı yaklaşımdır.

2.4.2 Kararlı Durum Genetik Algoritması

Örtüşen topluluklar kullanılır ve nesildeki yok edilecek birey sayısının kullanıcı tarafından belirlenmesine imkan verir.

2.4.3 Düzensiz Genetik Algoritmalar

Goldberg (1989) ve arkadaşları tarafından geliştirilen düzensiz genetik algoritmaların yaklaşımı, uygunluk değeri yeterince yüksek yapı taşlarının bulunup birleştirilerek daha yüksek uygunluk değerli bireylerin oluşturulması esasına dayanmaktadır.

2.4.4 Deme Genetik Algoritması

Deme genetik algoritması, kararlı durum genetik algoritmasını kullanarak birkaç topluluğun paralel evrimiyle sonuca ulaşmaya çalışır. Yeni neslin oluşturulması aşamasında bazı bireyler topluluklar arasında yer değiştirir.

2.5 Genetik Algoritmalarda Öğrenme

Evrim ve öğrenme, biyolojik uyarlamanın uzay-zamanda farklılıkla gerçekleşen iki türüdür. Evrim, tercihe bağlı çoğalma sonucu doğan bireylerden uygunluğu arttıracak farklılaşmalar gösteren ebeveynlerin yerini aldığı bir süreçtir. Öğrenme ise, bireyin var olduğu sürece uygunluğunu arttıracak değişimlere uğramasıdır. Görüldüğü üzere, süreçlerin zaman ölçekleri farklıdır. Evrim, genotipi değiştirirken öğrenme fenotipi etkiler. Fenotipteki değişim, genotipi

doğrudan etkilemez. Öğrenme ve evrimin etkileşimiyle ilgili detaylı bilgi için Stefano Nolfi ve Dario Floreano'nun çalışmalarından faydalanılabilir [3].

Öğrenme algoritmalarına dahil olmayan Genetik Algoritmalar bir kısım öğrenme işlemleri için güçlü arama yöntemleri önerir.

2.5.1 Michigan Yaklaşımı

Kromozomlar bireysel kurallardır ve kurallar kümesi topluluğu oluşturur. Kurallar ortamla etkileşimli olarak zamanla değişime uğrar. Michigan yaklaşımı tabanlı genetik öğrenme işlemi “Sınıflandırıcı Sistem” adıyla anılır.

2.5.2 Pittsburgh Yaklaşımı

Her bir kromozom tam bir kurallar kümesini kodlar. Çaprazlama ile yeni bir kurallar dizilişi ve mutasyon sonucu yeni bir kurallar kümesi elde edilir. Bazı durumlarda, değişken uzunluklu kural tabanları kullanılmaktadır.

2.5.3 Yinelemeli Kural Öğrenme

Michigan yaklaşımında olduğu gibi her bir kromozom bir kuralı temsil eder. Problemin çözümünü sağlayacak kurallar kümesini elde etmek için GA, aşağıda önerilene benzer yinelemeli düzende yer almalıdır.

- 1) Sistem için bir kural elde etmede GA kullan.
- 2) Kuralı nihai kurallar kümesine kat.
- 3) Kuralı değerlendir.
- 4) Şu ana kadar elde edilen kurallar çözüm sağlamaya yeterli değilse birinci adıma dön. Aksi takdirde çözümü sağlayan kuralı sun ve süreci bitir (Herrera vd., 2001).

2.6 Evrimci Yapay Sinir Ağları (EYSA)

Yapay Zeka başlığı altında öğrenme yeteneğiyle bilinen en popüler model yapay sinir ağlarıdır. Yapay Sinir Ağları ve Evrimci Arama süreçlerinin ortak kullanımı ile türetilen Evrimci Yapay Sinir Ağları (EYSA), sağladığı öğrenme yeteneği olan daha etkin yapay sistemler tasarım olanağı ile ilgi görmektedir. Ayrı ayrı çeşitli amaçlarla kullanılan bu

yaklaşımların etkileşimli kullanımı konu olduğunda bahsedilebilecek uygulamalar, YSA bağlantı ağırlık değerlerinin, YSA yapısının ve YSA öğrenme kurallarının belirlenmesidir.

2.6.1 EYSA Bağlantı Ağırlıklarının Evrimi

YSA’da öğrenme, denetimli veya denetimsiz gerçekleşir. Önceden belirlenmiş uygunluk kriteri gerçekleşmek üzere ağırlıkların ayarlanma süreci olan denetimli öğrenmede en çok kullanılan eğitim algoritması geri yayılımdır (Back Propagation).

Yerel minimum probleminin üstesinden gelmede, uygunluk değerlendirmesi için YSA’nın hata kriteri seçilerek, evrim sürecine işlerlik kazandırabilir. Sürecin belli başlı adımları aşağıda verilmiştir.

- 1) Kromozomlar, ağırlık değeri olarak atanır
- 2) Uygunluk değeri hesaplanır
- 3) Uygunluk değeri sonucu çoğalma süreci başlatılır
- 4) Genetik operatörler kullanılarak yeni nesilin elde edilir

Evrimci eğitim sürecinin, iki ana konusu kromozomları oluşturan ağırlık değerlerinin gösterimi ve benimsenen evrim sürecidir. Ağırlık değerlerinin temsilinde gerçel sayı veya iki tabanlı gösterim kullanılır.

2.6.1.1 İki Tabanlı Gösterim

Her ağırlık değeri, sabit uzunluklu bir bit dizisi ile temsil edilir. Kromozomlar ise bit dizilerinin birbirine eklenmesiyle elde edilir. İki tabanlı gösterimin kullanıldığı durumlarda doğrudan ikili tabanda kodlamanın yanısıra Gray kodu, üssel kodlama veya daha karmaşık bir kodlama kullanılabilir. Ancak burada gösterimin çözünürlüğü ağırlık değerinin hassasiyetini belirlediğinden büyük önem taşımaktadır. Çok kısa gösterim yakınsamayı zorlaştırır, uzun gösterim ise işlem yükünü ve süresini artırır. Kullanılacak bit sayısının optimizasyonu, kodlamada kullanılacak değer aralığı ve kodlama yöntemi açık bir problem teşkil eder.

2.6.1.2 Gerçel Sayı Kodlamalı Gösterim

İki tabanlı gösterimin eksikliklerini gidermek üzere her bir ağırlığa karşılık bir gerçel sayı ataması önerilen yöntemde, kromozomlar değerlerin dizi oluşturacak tarzda sıralanmasıyla oluşturulur. Çaprazlama işlemi, diziyi herhangi bir noktadan ya da genleri bölerek basamak

düzeyinde de gerçekleşebilir. Ebeveynlerin ortalamasını alan çaprazlama, rasgele mutasyon ve arama uzayına özel tanımlanabilecek işlemler, genlerin uygunluk değerini arttıracak gelişimi sağlayan diğer genetik operatörlerdir (Yao, 1999).

2.6.1.3 Evrimci ve Geri yayınımlı Eğitim Karşılaştırılması

GA gibi global arama algoritmalarına dayanan ve yerel minimumlara yakalanmayan evrimci eğitim, daha önce de belirtildiği üzere, karmaşık çok tepeli ve türevlenemeyen fonksiyonların da dahil olduğu geniş bir problem uzayına çözüm sağlayabilmesi nedeniyle caziptir. Eğitilmek istenen YSA'nın türüyle ilgili bir kısıtlama getirmez. Hata fonksiyonunun türevine ihtiyaç duymaması nedeniyle türevlenemeyen hata fonksiyonlarıyla karşılaştığında zorlanmaz. Evrimci algoritmaların ve özel olarak genetik algoritmaların global örnekleme yeteneği ince ayar yapabilme yeteneğinin önüne çıkar. Üç katmanlı ve ileri beslemeli YSA'lar için belirtilen avantajlarına karşın, evrimci algoritmaların hesaplama maliyeti yüksektir (Zhang ve Mühlenbein, 1993).

2.6.1.4 Hibrid Eğitim Yaklaşımı

Evrimeci eğitimin etkinliği, GA'nın global örnekleme yeteneği ve yerel arama yaklaşımının ince ayar becerisiyle birleştirilerek artırılabilir. Yüksek uygunluk değeri vaad eden bir bölgenin bulunması için GA kullanımını takiben bölgede arama yapmak üzere yerel arama yöntemleri işletilebilir (Zhang ve Mühlenbein, 1993).

3. YAPAY SİNİR AĞLARI

İnsanlığın doğayı araştırma ve taklit etme çabalarının en son ürünlerinden bir tanesi Yapay Sinir Ağları (YSA) teknolojisidir. Yapay Sinir Ağları, basit biyolojik sinir sisteminin çalışma şeklini simüle etmek için tasarlanan programlardır. Simüle edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözlemleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik (synaptic) bağlantıların ayarlanması ile olur. Yani, insanlar doğumlarından itibaren bir “yaşayarak öğrenme” süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitme yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıktı verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak sinapsların bağlantı ağırlıklarını bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.

YSA'lar, ağırlıklandırılmış şekilde birbirlerine bağlanmış bir çok işlem elemanından (nöronlar) oluşan matematiksel sistemlerdir. Bir işlem elemanı, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem elemanı, diğer nöronlardan sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır. Genelde, işlem elemanları kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır.

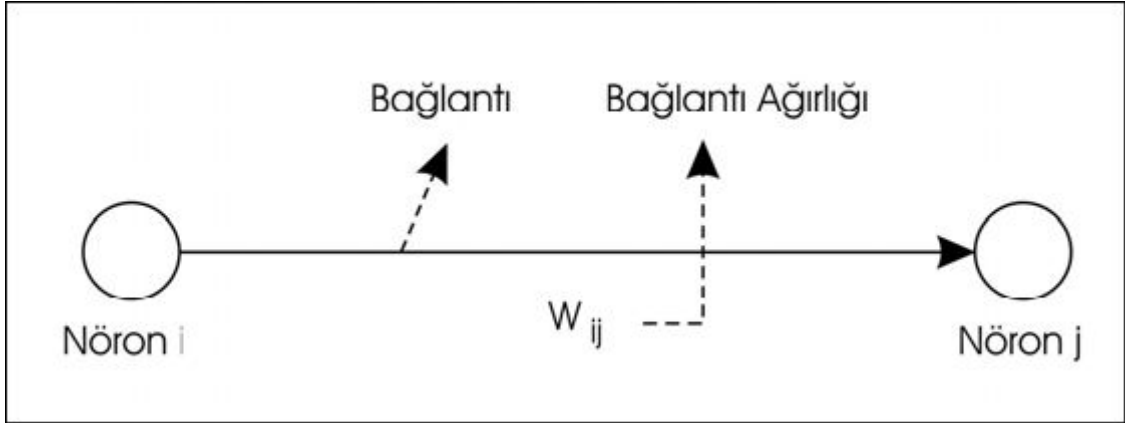
Sinirsel hesaplamanın merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. YSA'lar, geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem elemanı her hareketi sırasıyla gerçekleştirir. YSA'lar ise herbiri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem elemanlarından oluşmaktadır. En basit şekilde, bir işlem elemanı, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, işlem elemanlarının çalışma şekli yanıltıcı şekilde basittir. Sinirsel hesaplamanın

gücü, toplam işlem yükünü paylaşan işlem elemanlarının birbirleri arasındaki yoğun bağlantı yapısından gelmektedir.

Çoğu YSA'da, benzer karakteristiğe sahip nöronlar tabakalar halinde yapılandırılırlar ve transfer fonksiyonları eş zamanlı olarak çalıştırılırlar. Hemen hemen tüm ağlar, veri alan nöronlara ve çıktı üreten nöronlara sahiptirler.

YSA'nın ana ögesi olan matematiksel fonksiyon, ağın mimarisi tarafından şekillendirilir. Daha açık bir şekilde ifade etmek gerekirse, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA'ların davranışları, yani girdi veriyi çıktı veriyeye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir.

Bu bilgiler ışığında bakıldığında, YSA'ların yapısı üç ana eleman içermektedir ve Şekil 3.1'deki gibidir. Şekilden de görülebileceği gibi, YSA'ların yapısını oluşturan üç ana eleman temel işlem elemanı olan nöron, girdi ve çıktı yolunu sağlayan bağlantı ve bu bağlantıların sağlamlığını gösteren bağlantı ağırlığıdır (Yurtoğlu, 2005).

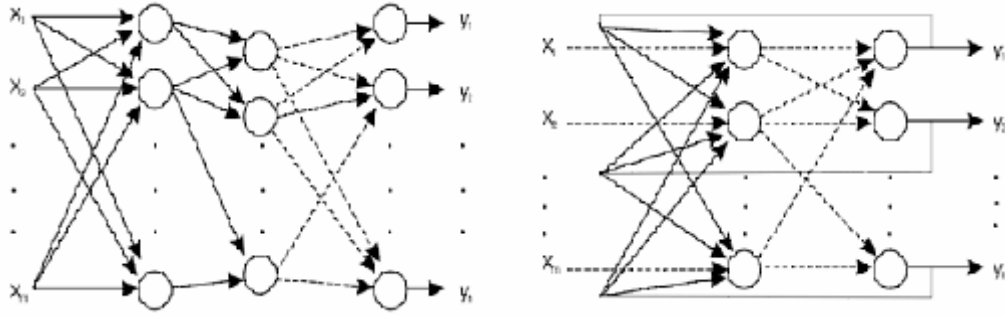


Şekil 3.1 YSA mimarisinin temel elemanları

YSA'ların çok sayıda farklı çeşitleri vardır. Bu farklılıkların kaynağı mimarisi, öğrenme yöntemi, bağlantı yapısı vb. olabilmektedir. Genel olarak, YSA'lar üç ana kritere göre sınıflandırılmaktadırlar. Bu kriterlerden biri öğrenme yöntemidir. Önceki bölümde belirtildiği gibi, temel olarak iki çeşit öğrenme algoritması vardır: eğitici öğrenme ve eğitici öğrenme. Her yöntemin kullandığı öğrenme kuralı değişebilmekteyse de, YSA'lar bu iki algoritmaya göre sınıflandırılırlar.

İkinci bir sınıflandırma, ağıın kullandığı veriye göre yapılmaktadır. Temel olarak, kalitatif ve kantitatif olmak üzere iki tür veri vardır. Kalitatif verilerle çalışan ağlar, ister yönlendirmeli ister yönlendirmesiz öğrenme kullansın, sınıflandırma ağları olarak bilinirler. Kantitatif veriler kullanan yönlendirmeli eğitime ise regresyon olarak adlandırılmaktadır.

Son sınıflandırma kriteri ise ağıın yapısıdır. Bazı ağlar ileri besleme şeklinde yapılandırılırken, bazı ağlar ise geri besleme yapısı içermektedir. İleri besleme sinir ağlarında, işlem elemanları arasındaki bağlantılar bir döngü oluşturmazlar ve bu ağlar girdi veriye genellikle hızlı bir şekilde karşılık üretirler. Geri beslemeli ağlarda (Recurrent Networks) ise bağlantılar döngü içerirler ve hatta her seferinde yeni veri kullanabilmektedirler. Bu ağlar, döngü sebebiyle girdinin karşılığını yavaş bir şekilde oluştururlar. Bu yüzden, bu tür ağların eğitime süreci daha uzun olmaktadır. Ayrıca, hem ileri besleme hem de geri yayılma olarak tanımlanabilecek ağ yapıları da mevcuttur (Yurtoğlu, 2005). Şekil 3.2’ de çok katmanlı ileri beslemeli ağ yapısı ile çok tabakalı geri beslemeli ağ yapısı gösterilmektedir.



Şekil 3.2 Çok katmanlı ileri ve geri beslemeli ağ yapıları

YSA çeşitleri arasında en çok bilinen ve kullanılan ağlar geri yayılma ile eğitilen çok katmanlı perceptron (Geri-Yayılmalı Ağ – Backpropagation Network), Radyal Tabanlı Fonksiyon (Radial Basis Function), Hopfield ve Kohonen sayılabilir. Çok fazla çeşit ve yoğun bir literatür olması nedeniyle, burada tüm ağ çeşitleri hakkında bilgi verilmemektedir. Bunun yerine, sadece bu çalışmada kullanılan radyal temelli fonksiyon ağları hakkında bilgi verilecektir.

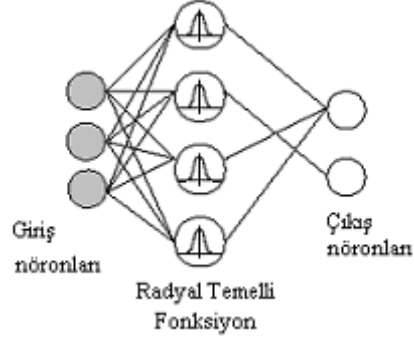
3.1 Radyal Temelli Fonksiyon Ağları

Denetimli bir sinir ağı dizaynı için çok çeşitli yollar izlenebilir. Çok katmanlı algılayıcı (denetimli olarak) dizaynı için geri yayılım (back propogation) algoritması istatistikte

“stokastik yaklaşım” olarak bilinen bir optimizasyon metodu uygulaması olarak değerlendirilebilir. Bir sinir ağı dizaynı çok boyutlu uzayda “eğri-uydurma problemi” olarak değerlendirildiğinde öğrenme, çok boyutlu uzayda taşınan dataya en iyi uyumu (best fit) sağlayan bir yüzey bulma ile eşdeğerdir, burada “best fit” bir takım istatistiksel yollar ile ölçülebilen anlamında kullanılmıştır. Benzer olarak genelleme de test verisini interpolate etmek için çok boyutlu uzayın kullanımına eşdeğerdir. Radyal temelli fonksiyonlar (RBF) ilk defa çok değişkenli gerçek bir interpolasyon probleminin çözümünde tanıtılmıştır. Bu konu hakkındaki ilk çalışmalar Powell (1985) tarafından yapılmıştır, şu anda ise RBF nümerik analiz konusundaki araştırmaların ana konusu durumundadır. Broomhead ve Lowe (1988) RBF’i sinir ağları dizaynı konusunda kullanan ilk kişilerdir. Moody ve Darken (1989), Renals (1989), ve Poggio ve Grosi (1990) yayınladıkları yayınlar ile RBF dizaynı ve uygulamaları alanlarına önemli katkıları sağlayan diğer kişilerdir. Poggio ve Grosi yayınlarında genelleştirilmesi gelişmiş yeni veri metodu olarak bu yapay sinir ağları sınıfına uygulanan düzenleme teorisinin kullanımı üzerinde durmuşlardır. En basit yapılı RBF’lerin ağ yapısı üç farklı katmandan oluşur. Giriş katmanı kaynak noktalarından (sensory units) oluşur. İkinci katman ise çok katmanlı algılamada farklı amaçlara hizmet eden yeteri kadar yüksek boyuta sahip saklı bir katmandır. Çıkış katmanı ise giriş katmanına uygulanan aktivasyon paternlerine ağın yanıtını sağlar. Saklı birim uzayından çıkış uzayına dönüşüm doğrusal iken, giriş uzayından saklı birim uzayına dönüşüm doğrusal değildir. Bu rasyonelin matematiksel doğrulaması ilk defa Cover (1965) tarafından daha önceki yayınlarında gösterilmiştir. Düşük boyutlu uzayda doğrusal olarak ayrılabilen, patern-sınıflama problemi yüksek boyutlu uzayda doğrusal olmayan bir şekilde çıkmaktadır, RBF ağı içindeki saklı birim uzayı boyutunun yüksek olmasının nedeni budur. Düzgün bir tasarımda, saklı birimlerin merkezleri adaptif yapılı ise, saklı birim uzayının boyutunu düşürmek mümkün olacaktır (Haykin, 1998).

Radyal tabanlı fonksiyon ağı tasarımı çok boyutlu uzayda eğri uydurma yaklaşımıdır. Bu nedenle RBF’in eğitimi çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşür. RBF’in genellemesi ise test verilerini interpolate etmek amacıyla, eğitim sırasında bulunan çok boyutlu yüzeyin kullanılmasına eşdeğerdir. RBF ağları, Şekil 3.2’de görüldüğü gibi giriş, orta ve çıkış olmak üzere üç katmandan oluşur, ancak giriş katmanından orta katmana dönüşüm herhangi bir ağırlıkla çarpılmadan direkt olarak yapılır. Orta katmandan çıkış katmanına dönüşüm ise bir ağırlıkla çarpılarak yapılır. Bu metodun temel çalışma prensibi N-boyutlu uzaydaki verileri farklı sınıflara veya kategorilere ayırıyor olmasıdır. Her kategori bir merkez içermekte ve girişlerin kategorilere ayrılması bu merkezler

sayesinde olmaktadır. Sınıflama N-boyutlu giriş vektörü ile merkez vektörü arasındaki öklit mesafesinin hesaplanmasıyla yapılmaktadır.



Şekil 3.3 Radyal temelli ağ yapısı

RBF’de uyarlanabilecek serbest parametreler; merkez vektörleri, radyal fonksiyonların genişliği ve çıkış katman ağırlıklarıdır. Çıkış katmanı doğrusal olduğundan ağırlıklar, eğitim düşme yada doğrusal en iyileme yöntemleri ile kolayca bulunabilir. Merkezler, girişler arasından rastgele ve sabit olarak seçilebilmekle birlikte RBF’in performansını iyileştirmek amacıyla merkez vektörlerinin ve genişliği uyarlanması için çeşitli yöntemler geliştirilmiştir. Merkez vektörleri, eğitim düşme yöntemine göre eğitici öğrenme algoritması ile uyarlanarak, dik en küçük kareler yöntemi ile yada kendiliğinden düzenlemeli yöntemle giriş örneklerinden öbekleme yapılarak belirlenebilir (Çoban, 2003).

RBF’in genel matematiksel ifadesi aşağıda belirtildiği gibidir.

$$s_j(x) = \sum_{i=1}^N w_{ij} \varphi_i(x) + b_j \quad (3.1)$$

Burada w_{ij} i. gizli katman ile j. çıkış nöronu arası dönüşümdeki ağırlığı göstermekte, b_j muhtemel bir bias, $\varphi(x)$ ise aktivasyon fonksiyonudur. RBF’ de orta katman aktivasyon fonksiyonu genellikle standart öklit uzaklıklarını üstel fonksiyondan geçiren Gauss fonksiyonudur ve aşağıdaki gibi tanımlanır.

$$\varphi_i(x) = \exp \left[-\frac{\|x - c_i\|^2}{2\sigma_i^2} \right] \quad (3.2)$$

Bu ifadede x giriş vektörü, c_i merkez vektörüdür.

3.2 RBF Ağları ile Çok Katmanlı Algılayıcıların Karşılaştırılması

Radyal Temelli Fonksiyonu ağları ve çok katmanlı yapay sinir ağları yapısı, doğrusal olmayan katmana sahip ileri beslemeli ağ örneklerindendir ve aynı zamanda yaygın yaklaşıtııcılarıdır. Ayrıca RBF ağlarının yaygın kullanımlarına ait Park ve Sandberg (1991) tarafından tanımlanan resmi bir ispat bulunmaktadır. Bu yüzden her zaman belirli bir MLP'yi tam doğrulukla taklit eden RBF ağının bulunması şaşırtıcı değildir. Bununla beraber bu iki ağın birbirinden farklılaştığı bazı önemli konular altta belirtilmiştir.

1. MLP bir ve birden fazla gizli katmana sahip olabilirken, temel yapıdaki bir RBF ağ sadece tek bir gizli katmana sahip olabilmektedir.
2. Genel olarak, gizli ya da çıkış katmanına konumlanmış MLP'nin hesaplama düğümleri genel bir nöron modeli kullanır. Öte yandan RBF ağının gizli katmanındaki hesaplama düğümleri oldukça farklıdır ve de ağın çıkış katmanında farklı bir amaca hizmet eder.
3. RBF ağına ait gizli katman doğrusal değildir halbuki çıkış katmanı doğrusaldır. Öte yandan genelde doğrusal olmayan yapıya sahip olan MLP'nin gizli ve çıkış katmanları sınıflandırıcı olarak kullanılır. Bununla beraber doğrusal olmayan regresyon problemlerinin çözümünde MLP kullanıldığında, genellikle çıkış için bir doğrusal katman tercih edilir.
4. RBF ağının herbir gizli birimi için aktivasyon fonksiyonun hesaplanması konusunda giriş vektörü ve merkez birim arasındaki öklit uzaklığı kullanılır. Öte yandan MLP'deki herbir gizli birim için aktivasyon fonksiyonu, giriş fonksiyonunun "iç çarpım"ını ve o birime ait sinaptik ağırlık vektörünü hesaplar.
5. MLP ağları doğrusal olmayan giriş-çıkış tasarımlarına global yaklaşımlar oluşturur. Sonuç olarak, MLP ağlarında çok az eğitim kümesi bulunurken ya da hiç bulunmazken, öte yandan üstel olarak bozulmuş, lokalize non-lineerliliği (Örn. Gauss fonksiyonları) kullanan RBF ağları doğrusal olmayan giriş çıkış tasarımına yerel yaklaşımlar oluşturur. Bu ağlar hızlı öğrenme yapısına ve eğitim kümesi gösterimine

göre indirgenmiş hassasiyete sahiptir. Halbuki birçok durumda arzulanan düzeydeki düzgünlük tasarımı için giriş boşluğunu aralamada çok sayıda radyal temelli fonksiyon gerekmektedir.

RBF ağının çıkış katmanına ait doğrusal karakteristik, bu yapının çok katmanlı algılayıcı yapısından çok Rosenblatt algılayıcı yapısına daha yakın olduğu anlamına gelir. Halbuki RBF ağ yapısı, giriş boşluğunun rasgele seçilmiş non-lineer dönüşümlerini yürütme yeteneğine sahip algılayıcılardan farklıdır. Bu, başka herhangi bir doğrusal algılayıcıyla değil fakat RBF ağı ile çözülebilen XOR problemi ile tarif edilebilir (Haykin, 1998).

3.3 Öğrenme Stratejileri

Radyal temelli fonksiyon ağları tarafından gerçekleştirilen öğrenme süreci şu şekilde görülür: Ağın çıkış birimleri ile ilgili doğrusal ağırlıklar, saklı katmanların doğrusal olmayan aktivasyon fonksiyonları ile karşılaştırıldıklarında başka bir zaman ölçeğinde gelişim gösterirler. Bu durumda saklı katmanların aktivasyon fonksiyonları bir nonlinear optimizasyon stratejisine bağlı olarak daha yavaş ilerlerken, çıkış katmanlarının ağırlıkları kendilerini lineer optimizasyon stratejileri sayesinde seri bir şekilde yenilerler. Burada vurgulanması gereken önemli nokta şudur; Radyal temelli fonksiyon ağlarının farklı katmanları farklı görevleri yerine getirir ve bu da bir ağın saklı katmanlar ve çıkış katmanlarının optimizasyonunu farklı tekniklerle, hatta belki de farklı zaman ölçekleri kullanılarak gerçekleştirmeyi mantıklı hale getirir.

RBF tasarımında radyal temel fonksiyonlarının merkezlerinin belirlenmesine göre farklı öğrenme stratejileri izlenebilir. Daha doğrusu, bunun için 3 adet yaklaşım vardır.

3.3.1 Rastgele Seçilmiş Sabit Merkezler

En basit yaklaşım, saklı katmanların aktivasyon fonksiyonlarının sabit radyal temel fonksiyonları tarafından tanımlanmış olduklarını varsaymaktır. Merkezlerin konumları eğitim kümesi içinde rastgele seçilir. Öğrenme kümesindeki verilerin problemi yeterince gözler önüne seren bir şekilde dağıldığını varsaydığımızda, bu yöntemi uygulamak akılcıdır. Radyal temelli fonksiyonlar için, standart sapması merkezlerin dağılımına göre sabit, isotropik bir Gauss fonksiyonu uygulayabiliriz. Bu durumda, t_i noktasında merkezlenmiş normalize bir RBF şöyle tanımlanabilir.

$$G(\|x - t_i\|^2) = \exp\left(\frac{M}{d^2} \|x - t_i\|^2\right), \quad i = 1, 2, \dots, M \quad (3.3)$$

Burada M merkezlerin sayısı ve d seçilen merkezlerin aralarındaki uzaklıktır. Aslında, bütün Gaus RBF'lerinin standart sapması (veya genişliği) aşağıdaki gibi sabit alınabilir :

$$\sigma = \frac{d}{\sqrt{2M}} \quad (3.4)$$

Standart sapmanın bu şekilde seçilmesi bize Gaus fonksiyonunun ne aşırı geniş ne de aşırı dar olacağını garanti eder, her iki uç nokta da istenmeyen durumlardır.

Bu yaklaşımda öğrenilmesi gereken parametreler, ağırlık çıkış katmanlarının lineer ağırlıklarıdır. Bunun basit bir yöntemi, sanki tersi (pseudoinverse) metodudur.

Bu durumda :

$$w = G' d \quad (3.5)$$

d bu denklemde öğrenme kümesinin istenilen cevabıdır. G' matrisi, G'nin sankitersidir ve aşağıdaki gibi tanımlanır :

$$G = \{g_{ij}\} \quad (3.6)$$

$$g_{ij} = \exp\left(-\frac{M}{d^2} \|x_j - t_i\|^2\right) \quad j = 1, 2, \dots, N; \quad i = 1, 2, \dots, M \quad (3.7)$$

Burada x_j , eğitim kümesinin j'inci giriş vektörüdür.

Bir matrisin sankitersini kullanan bütün algoritmalar tekil değer ayrıştırması (singular value decomposition) kullanır (Golub ve Van Loan, 1989; Haykin, 1991):

Eğer G gerçel bir N'e M matris ise, bu durumda aşağıdaki ortogonal matrisler tanımlıdır :

$$U = \{u_1, u_2, \dots, u_N\}$$

$$V = \{v_1, v_2, \dots, v_N\}$$

öyle ki :

$$U'GV = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_K), \quad K = \min(M, N) \quad (3.8)$$

U matrisinin sütunları, G'nin sol singüler (tekil) vektörleri, V'nin sütun vektörleri de G'nin sağ tekil vektörleri olarak adlandırılırlar. $\sigma_1, \sigma_2, \dots, \sigma_K$ ise G matrisinin singüler (tekil) değerleridirler. tekil değer ayrıştırması teoremine göre G matrisinin M'ye N sankitersi matrisi

$$G' = V \Sigma' U^T \quad (3.9)$$

Burada Σ' , G matrisinin tekil değerlerinden oluşan N'e N bir matristir.

$$\Sigma' = \text{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_K}, 0, \dots, 0\right) \quad (3.10)$$

Sankitersi matrislerin kullanımı ile ilgili daha etkili algoritmalar, Golub ve Van Loan (1989) ve Haykin (1991)'in kitaplarında daha ayrıntılı bir şekilde tartışılmaktadır.

3.3.2 Merkezlerin Düzenli Seçimi

İkinci yaklaşımda, çıkış katmanının doğrusal ağırlıkları denetimli öğrenme kuralı kullanarak hesaplanırken, radyal temelli fonksiyonlardaki merkezlerin yerlerinin kendi kendilerine düzenlenmiş şekilde hareket etmelerine izin verilir. Diğer bir deyişle, ağ hibrid öğrenme işlemine uğrar (Moody ve Darken, 1989; Lippman, 1989). Öğrenme işleminin kendi kendine düzenlenen elemanı, anlamlı veri bulunduğunda, giriş uzayının yalnızca o kısımlarında radyal temelli fonksiyonların merkezlerini yerleştirirken anlamlı bir yolla ağ kaynaklarını ayırmaya hizmet eder.

Saklı birimlerin merkezlerinin kendi kendine düzenlenmiş seçimleri için, standart k-en yakın komşu kurallı kullanılır (Moody ve Darken, 1989). Bu kural, giriş vektörü x 'i k-en yakın örneklerinde ifade edilen en çok kullanılan etiket ile belirleyerek ayırır. Diğer bir deyişle, k-enyakın komşular üzerindeki etiketleri araştırarak ve bir oy alarak, bir karar oluşturur (Duta ve Hart, 1973).

Çıkış katmanının doğrusal ağırlıklarını hesaplamak için kullanılan denetimli öğrenme işlemi için, least-mean square (LMS) gibi, verimli ve basit bir algoritma olan hata düzeltme öğrenme kuralı kullanılabilir. RBF ağının saklı birimlerindeki çıkışlar, LMS algoritmasının girişleri gibi davranır (Haykin, 1998).

3.3.3 Merkezlerin Denetimli Seçimi

Üçüncü yaklaşımda radyal temelli fonksiyonların merkezleri ve ağı diğer bağımsız parametreleri denetlenmiş bir öğrenme safhası geçirmektedir. Başka bir deyişle RBF ağı onun en genel formunu ele alır. Böyle bir işlem için en doğal aday LMS algoritmasının genelleştirilmesini sunan gradyan-inişli prosedürünü kullanarak en uygun şekilde gerçekleştirilen hata giderme öğrenmesidir.

Böyle bir öğrenme prosedürünün geliştirilmesindeki ilk adım değer fonksiyonunun anlık değerini belirlemektir.

$$E = \frac{1}{2} \sum_{j=1}^N e_j^2 \quad (3.11)$$

N'nin öğrenme işlemini üstlenen eğitim örneklerinin sayısını ve e_j 'nin hata sinyalini gösterdiği fonksiyon aşağıdaki gibi tanımlanmıştır.

$$\begin{aligned} e_j &= d_j - F^*(x_j) \\ &= d_j - \sum_{i=1}^M w_i G(\|x_j - t_i\|c_i) \end{aligned} \quad (3.12)$$

Burada ihtiyaç duyulan, “E”yi minimize etmek için w_i , t_i ve $\sum^{-1}_i$ ’yi bulmaktır.

Bu minimizasyonun sonuçları Çizelge 3.1’de toplanmıştır.

- Bu değer fonksiyonu lineer parametre w_i bakımından konvektir, fakat t_i ve $\sum^{-1}_i$ bakımından konveks değildir. Sonraki durumda t_i ve $\sum^{-1}_i$ ’nin optimum değerlerini araştırmada, parametreler uzayında yerel minimuma takılı kalabilir.
- w_i , t_i ve $\sum^{-1}_i$ için güncel denklemler, farklı öğrenme oran parametreleri olan η_1, η_2 ve η_3 atar.
- Gerideki propagasyon algoritmasından farklı olarak Çizelge 3.1’de tanımlanan RBF ağı için gradyan –inişli işlem hata geri-yayınım içermez.
- Gradyan vektörünün $\partial E / \partial t_i$ görevden bağımsız küme tepkisine benzer etkisi vardır (Poggio ve Girosi, 1990).

Gradyan-inişli prosedürünün başlaması parametreler uzayında araştırmanın, RBF ağı olarak standart patern-sınıflama metodunu gerçekleyerek bilinen kullanışlı bölgede araştırma için alanı sınırlayan başlangıç koşulundan başlaması için istenir. Böylece ağırlık uzayında istenmeyen bir yerel minimuma kesişme olasılığı azaltılır. Örneğin biz her sınıftaki her paternin tam bir gauss dağılımından çizildiğini varsayan standart gauss sınıflandırıcısı ile başlayabiliriz.

Buradaki açık soru şudur: Radyal temelli fonksiyonların merkez pozisyonlarını adapte ederek ne kazanabiliriz? Bu sorunun cevabı doğal olarak ilgi alanıyla orantılı uygulamaya bağlıdır. Bununla birlikte literatürde bahsedilen bazı sonuçların temelinde merkezlerin hareket etmelerine izin vermek düşüncesi yer alır.

RBF ağlarını kullanarak Lowe (1989) tarafından yapılan çalışma gösterir ki, gizli katman fonksiyonlarının aktivasyonunu belirleyen parametrelerin doğrusal olmayan optimizasyonundan minimal ağ konfigürasyonuna ihtiyaç duyulduğunda faydalanılır. Bununla birlikte Lowe'ye göre genelleştirmedeki aynı performans gizli katmandaki sabitlenmiş daha fazla daha geniş bir RBF ağı kullanılarak ve ağların çıkış katmanlarını doğrusal optimizasyonla adapte ederek elde edilir. Bu da gizli katmanlardaki sabitlenmiş daha fazla merkeze sahip ağ demektir.

Wettschereck ve Dietrich (1992) radyal temelli fonksiyonların sabitlenmiş merkezli ağını ayarlanabilir merkezli genelleştirilmiş radyal temelli fonksiyonlarla karşılaştırmışlardır. Sonraki durumda merkezlerin pozisyonları denetlenmiş öğrenmeyle belirlenir. Performans konuları NET konuşma görevi için yapılmıştır. Orjinal Net konuşma deneyi Sejnowski ve Rosenberg tarafından (1987) geri yayılım algoritması ile eğitilen çok katmanlı algılayıcı kullanarak gerçekleştirilmiştir. Bu deneyin amacı İngilizce heceleme yapay sinir ağının nasıl kendi fonetik telafuzuna adreslendiğini öğrenmektir. Wettschereck ve Dietrich tarafından yapılan deneyde NET konuşmada etki alanı aşağıdaki gibi özetlenir.

RBF ağları (denetimsiz öğrenme ile merkez pozisyonları ve denetimli öğrenme ile çıkış katmanı ağırlıkları), geri yayılım algoritması ile eğitilen çok katmanlı algılayıcılar gibi genelleştirme yapmamıştır.

Genelleştirilmiş RBF ağları (çıkış katmanı ağırlıkları gibi, merkezlerin pozisyonlarının denetimli öğrenmesi ile), çok katmanlı algılayıcıların genel performansını geçebilir (Haykin, 1998).

Çizelge 3.1 RBF ağı için, doğrusal ağırlıklar, merkezlerin yeri ve genişliği için adaptasyon formülleri

1. *Doğrusal ağırlıklar* (çıkış katmanı)

$$\frac{\partial E(n)}{\partial w_i(n)} = \sum_{j=1}^N e_j(n) G(\|x_j - t_i(n)\| c_i)$$

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial E(n)}{\partial w_i(n)} \quad i = 1, 2, \dots, M$$

2. *Merkezlerin pozisyonu* (saklı katman)

$$\frac{\partial E(n)}{\partial t_i(n)} = 2w_i(n) \sum_{j=1}^N e_j(n) G'(\|x_j - t_i(n)\| c_i) \Sigma_i^{-1} [x_j - t_i(n)]$$

$$t_i(n+1) = t_i(n) - \eta_2 \frac{\partial E(n)}{\partial t_i(n)} \quad i = 1, 2, \dots, M$$

3. *Merkezlerin genişliği* (saklı katman)

$$\frac{\partial E(n)}{\partial \Sigma_i^{-1}(n)} = -w_i(n) \sum_{j=1}^N e_j(n) G'(\|x_j - t_i(n)\| c_i) Q_{ji}(n)$$

$$Q_{ji}(n) = \|x_j - t_i(n)\| \|x_j - t_i(n)\|^T$$

$$\Sigma_i^{-1}(n+1) = \Sigma_i^{-1}(n) - \eta_3 \frac{\partial E(n)}{\partial \Sigma_i^{-1}(n)}$$

3.4 Tartışma

RBF ağ yapısı, saklı birimlerinin oluşumunun çıkış birimlerinden farklı olması nedeniyle alışılmışın dışındadır. RBF saklı birimlerin dizaynında temel teşkil etmektedir. günümüzde sayısal analizin belli başlı çalışma alanlarından biri olan RBF ağ teorisi, radyal temelli fonksiyonlar ile yakından ilişkilidir (Singh 1992). Bir diğer ilginç nokta da çıkış katmanının doğrusal ağırlıklarla ayarlanabilir parametreler kümesi sağlamasıdır. Lineer adaptif filtreleme ile ilgili olarak literatür taranırsa pek çok bilgi bulunabilir (Haykin,1991) (Haykin, 1998).

3.4.1 Boyut Laneti

Standart RBF ağırları gibi, kernel tipi yaklaşımlar da, giriş uzayının boyutuna bağlı olarak, saklı birimlerin sayısındaki üstel artış nedeniyle “boyut laneti” olarak adlandırılan sıkıntıdan müzdariptir.

Boyut laneti, özellikle görüntü ve ses tanıma gibi geniş ölçekli problemlerin çözümünde daha güçlü bir hal almaktadır. Uzantılarının öklit mesafe matrisi tabanlı olduğu, sabit Gauss radyal temelli fonksiyonları kullanan standart RBF ağı, baz fonksiyonunun merkezleri etrafında hiper-küre oluşturarak işlem yaptığı için boyut lanetinden müzdariptir. Bu problemi hafifletmek için hiper-küre yapısı, ayarlanabilir yöne sahip hiper-elips yapısı ile yer değiştirilebilir. Bu durum örnek olarak, ağırlıklı değer fikrine dayanan genellenmiş Gauss radyal temelli fonksiyonlar ile sonuçlandırılmaktadır (Poggio ve Girosi, 1990; Lowe, 1989). Bu nedenle, hiper yüzey yaklaşımı (tekrar yapılandırma) daha az sayıda saklı birim kullanımına imkan veren fakat adapte etmek için daha çok parametreye sahip olan bir tasarıdır. Saha et al.(1991), üstel Gauss radyal temelli fonksiyonun bir polinomla yer değiştirildiği, yönlendirilmiş radyal temelli olmayan fonksiyon ağırları (oriented non-radial basis functions - ONRBF) taslağına başvurur. Özellikle, i. saklanmış birimin aktivasyon fonksiyonu;

$$\varphi_i(x) = \exp\left(-\left\|T_i \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}\right\|^2\right) \quad (3.13)$$

ile ifade edilir. $x_1, x_2, x_3, \dots, x_p$ giriş vektörü x 'in elemanlarıdır. T_i ise p 'ye $(p+1)$ bir matristir.

$T_i, i=1,2,3,\dots,M$ matrisi giriş vektörünü dönüştürürler ve bu dönüşümler, giriş vektörünün çevrimine, ölçeklenmesine, rotasyonuna karşılık gelir.

3.4.2 Ortogonal En Az Kareler Methodu

Bir diğer konu da RBF ağının, eğitim kümesinin “aşırı uydurma” olmasını engelleme yönünde düzenlemektir. Bu amaca ulaşmak için, alternatif yol olarak OLS prosedürü kullanılır.(Chen, 1991). Bu metod, lineer beklenen tepkinin $d(n)$ olarak tanımlandığı doğrusal regresyon modelinden kaynaklanır (Haykin, 1991):

$$d(n) = \sum_{i=1}^M x_i(n)a_i + e(n), \quad n = 1, 2, \dots, N \quad (3.14)$$

Denklemde, a_i model parametreleri, $x_i(n)$ bağlanım parametreleri ve $e(n)$ rezidüyü (hatayı) ifade eder. Matris sistemi kullanılarak, denklem (3.14)

$$d = Xa + e \quad (3.15)$$

şeklinde (aşağıdaki denklemler sağlandığında) tekrar ifade edilebilir.

$$\begin{aligned} d &= [d(1), d(2), \dots, d(N)]^T \\ a &= [a_1, a_2, \dots, a_M]^T \\ X &= [x_1, x_2, \dots, x_M] \\ x_i &= [x_i(1), x_i(2), \dots, x_i(N)]^T, \quad 1 \leq i \leq M \end{aligned}$$

$$e = [e(1), e(2), \dots, e(N)]^T$$

Bağlanım vektörü x_i temel vektörler kümesidir. Denklem 3.15'in en az kare çözümü matrsten X_a temel vektörler tarafından aralanmış boşluğa, beklenen tepki vektörü d 'nin izdüşümü olarak düşürülmesi ile elde edilir. OLS metodu adını bu yöntemden almaktadır. OLS metodu bağlanım vektörlerinin $x_1, x_2, x_3, \dots, x_m$ dönüşümünü kapsar ve $u_1, u_2, u_3, \dots, u_m$ ile gösterilen ortogonal temel vektörlere tekabül eder. Örnek olarak standart Gram-Schmidt ortogonalizasyon prosedürü alttaki dönüşümü gerçekleştirmek için kullanılır.

$$u_1 = x_1$$

$$\alpha_{ik} = \frac{u_i^T x_k}{u_i^T u_k}, \quad 1 \leq i \leq k$$

$$u_k = x_k - \sum_{i=1}^{k-1} \alpha_{ik} x_i$$

$$k = 2, \dots, M \text{ iken.}$$

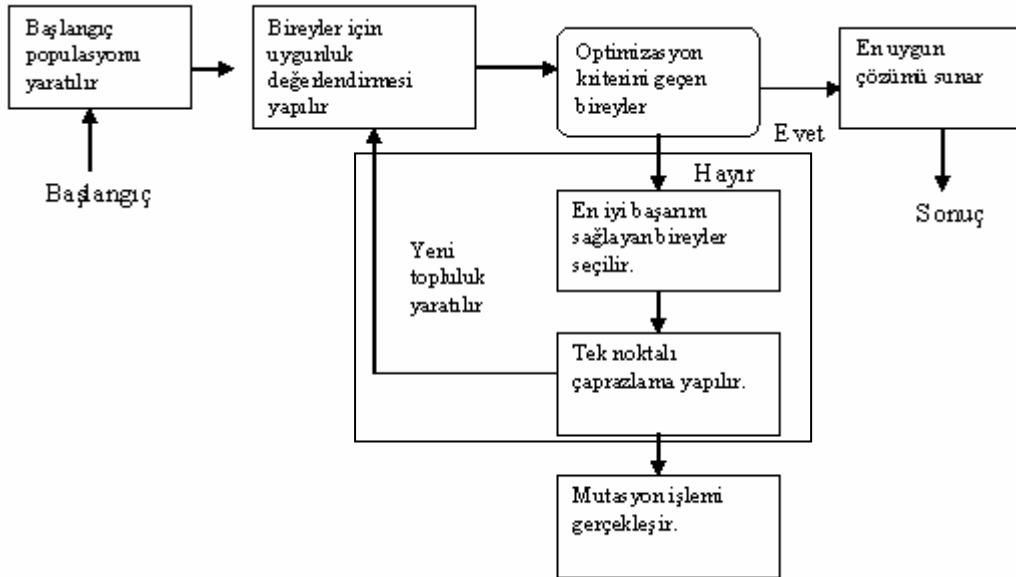
Sinir ağırları bağlamında, OLS öğrenme prosedürü, RBF merkezleri $t_1, t_2, t_3, \dots, t_M$ ' yi $M < N$ iken, eğitim kümesi vektörlerinin $x_1, x_2, x_3, \dots, x_N$ bir alt grubu olarak tayin eder. Yeterli ağ

performansı oluşana kadar merkezlerin birer birer belirlenmesi yoluna gidilir (Gram-Schmidt ortogonalizasyon prosedürü takip edilir). Prosedürün herbir adımında beklenen tepkiye ait değişken sayısı adım adım artarak maksimize olur. Böyle yapılarak OLS öğrenme prosedürü, beklenen tepkinin açıklanamayan tayin edilmiş varyansı için merkezleri rasgele seçilmiş RBF ağından daha küçük saklı katmanı olan bir RBF ağı oluşturur. Sonuç olarak OLS öğrenme prosedürü az sayıdaki RBF ağı ile iyi numerik özellikler elde etmeye yarayan kullanışlı bir yapı sunar (Haykin, 1998).

4. GA-RBF SİMULASYONU

Bu çalışmada çeşitli veri kümelerinin sınıflandırılması için kullanılan RBF ağıının optimizasyonu genetik algoritmalar kullanılarak gerçekleştirilmiştir. Ağdan en yüksek test performansını elde etmek için; ağa uygulanacak örnek paternlerin (yani merkezlerin) ve gauss fonksiyonunun genişliği genetik algoritma ile belirlenmeye çalışılmıştır. Bu aşamada veri kümesinden simülasyon için minimum sayıda örnek alınarak, geriye kalan test kümesinin yüksek başarı oranı ile sınıflandırılması amaçlanmıştır.

Algoritmada ilk olarak bir populasyon yaratılır. Gerçek kodlamalı dijital dizilimine sahip bu bireyler kromozomlarla ifade edilir. Kromozomun her bir biti “gen” olarak adlandırılır. Ardından her bir birey için uygunluk değerlendirilmesi yapılır. Burada amaç uygunlukları küçük olan kromozomların elenip, daha iyi uygunluk fonksiyonuna sahip olan kromozomların seçilmesidir. Optimizasyon kriterini geçen değerler, en uygun çözümü sunar. Ancak bu değeri geçemeyen populasyon elemanları için seçim işlemi uygulanmaktadır. Bu istenen çözüm sağlanıncaya kadar devam ettirilir. Seçilen kromozomlar arasında çaprazlama ve gerekirse mutasyon işlemi yapılır. Ve kromozomun uygunluk değerine tekrar bakılır. Bu işlem tatmin edici bir sonuç alana kadar devam eder (Hatanaka vd., 2003).



Şekil 4.1 Algoritmanın çalışma akışı

Her bir kromozom ya da birey, her bir sınıf için başlangıç pozisyonlarının merkez kümesini kodlar.

$$\text{Birey} = [\{b_1^1, \dots, b_{k_1}^1\}, \dots, \{b_1^n, \dots, b_{k_n}^n\}]$$

Burada n, veri kümesinin sınıf sayısını ifade eder. k_i , i. sınıf için belirlenecek olan merkez sayısıdır. Örneğin 3 sınıfı olan ve her birindeki örnek sayısı 50 olan iris kümesi için $n=3$ 'tür ve k_i her sınıf için 10 olarak belirlenmiştir.

Genetik algoritma için uygunluk fonksiyonu test başarımlar oranıdır.

$$\text{Uygunluk} = \frac{\text{doğru_sınıflanan} - \text{eğitmede_kullanılan}}{\text{veri_kümesi} - \text{eğitmede_kullanılan}} \times 100$$

Yukarıdaki eşitlikte doğru sınıflanan ve tüm veri kümesindeki örnek sayısından eğitimde kullanılan örnek sayısını çıkarmamızın amacı, yalnızca test başarımlarını bulmaktır. Aksi takdirde hem test hem eğitim başarımlarını bulmuş oluruz. Eğitimde kullanılan örnek sayısı, her bir sınıf için belirlenen merkez sayılarının toplamıdır. Merkezlerin, veri kümesindeki satır değerleri olarak algılanması gereken bu yapıda optimize edilecek parametrelerin alabileceği maksimum ve minimum değerler algoritmanın işlemesi sırasında belirtilmektedir. Algoritma verilen sınır değerler arasından en uygun çözümü bulmaya çalışmaktadır.

Bireylerin seçimi, en iyi uygunluk değerine sahip olan bireyleri seçmeye yöneliktir. Seçim işleminde rulet tekerleği stratejisi kullanılmıştır. Populasyonun büyüklüğü N iken ve i. bireyin uygunluğu, $uygunluk(i)$ ile belirtilirken, i. bireyin seçilme olasılığı:

$$p_i = \frac{uygunluk(i)}{\sum_{j=1}^N uygunluk(j)}$$

p_i , populasyondaki olasılık yoğunluk fonksiyonunu ifade eder.

Mutasyon her veri kümesi için sabit 2 olarak seçilmiştir. Böylelikle sistemin lokal minimumda takılmasını önleyecek kadar sayıdaki bireylerde genetik bozulma yoluna gidilmiştir.

4.1 Iris Veri Kümesi

Örnek bir uygulama olarak 3 farklı iris çiçeğinin 5 özellik değerine göre sınıflandırılması problemi optimize edilmiştir. 150 örnek ve 3 sınıfa sahip bu veri kümesi için her bir sınıftan 10 örnek eğitim kümesi olarak alınmıştır. Genetik algoritmanın bize sağladığı avantaj en iyi

sonucu verecek satırları kendisinin belirlemesidir. Algoritma eğitme kümesi ile simüle edilip, tüm veri kümesi ile test edildiğinde 149 örneğin doğru sınıflandırıldığı tespit edilmiştir.

Çizelge 4.1’de iris veri kümesine ait eğitme ve test örnek sayıları verilmiştir. Eğitme kümesi için 30, test kümesi içinse 120 örnek alınarak hem RBF-GA hem de yalnızca RBF ağları kullanılarak simüle edilen iris çiçeği probleminde, doğru sınıflandırılan örnek sayıları Çizelge 4.2’de verilmiştir.

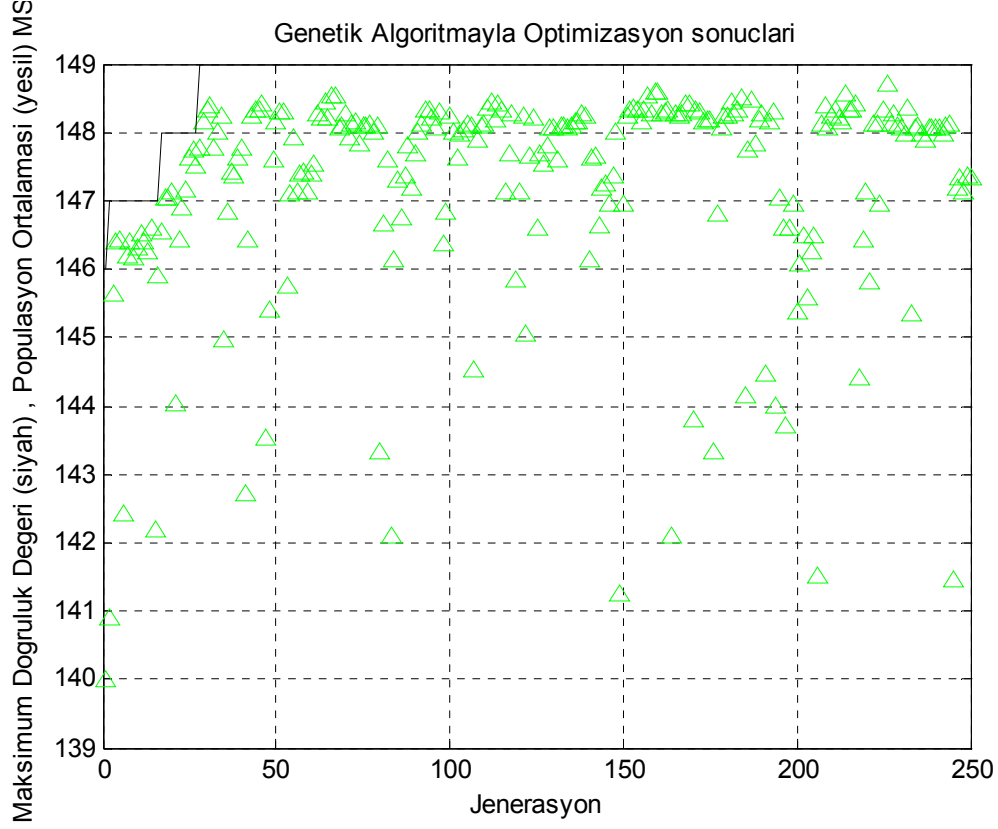
Çizelge 4.1 Iris veri kümesine ait eğitme ve test örnek sayıları

İris	Toplam Veri Kümesi	Eğitme Veri Kümesi	Test Veri Kümesi
Sınıf 1	50	10	40
Sınıf 2	50	10	40
Sınıf 3	50	10	40

Çizelge 4.2 Iris veri kümesi için doğru sınıflanan örnek sayıları

İris	Sınıf 1	Sınıf 2	Sınıf 3
RBF-GA	50	49	50
RBF	43	43	25

Şekil 4.2’de genetik algoritma ile yapılan optimizasyon işlemi sonucunda simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir. Buna göre, üçgenler populasyon ortalamasını ifade eder, bu da her jenerasyonda elde edilen test başarı değerlerinin ortalaması demektir. Siyah eğriler ise maksimum doğruluk değerini ifade etmektedir.



Şekil 4.2 İris çiçeği-genetik optimizasyon sonuçları

Bu algoritma için jenerasyon sayısı 250 olarak alınmış ve gauss fonksiyonun genişliği 523 olarak belirlenmiştir. Grafik incelendiğinde işlemin 50 jenerasyondan sonra da aynı sonucu vereceği görülmektedir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir ağı programında da kullanılmıştır. σ değeri yine 523 alınarak program çalıştırıldığında elde edilen test başarımları Çizelge 4.13’de verilmiştir.

Buna göre her bir sınıftan yalnızca 10 örnek alınarak çalıştırılan RBF programı, GA ile yapılan optimizasyonla kıyaslandığında daha düşük bir sonuç vermektedir. Bu da RBF-GA ağ yapısının başarısını açıkça ortaya koymaktadır.

4.2 Tiroid Veri Kümesi

Tiroid veri kümesi 3 sınıfa bölünmüş 215 örneğe sahiptir. Tiroid veri kümesinde birinci sınıfta 150, ikinci sınıfta 35 ve üçüncü sınıfta 30 örnek mevcuttur. Tiroid veri kümesine ait eğitme ve test örnek sayıları Çizelge 4.3’te gösterilmektedir. İlk sınıftan 25, ikinci sınıftan 7 ve üçüncü sınıftan 3 örnek alınarak ağ eğitildiğinde doğru sınıflandırılan örnek sayısı Çizelge 4.4’de verilmiştir.

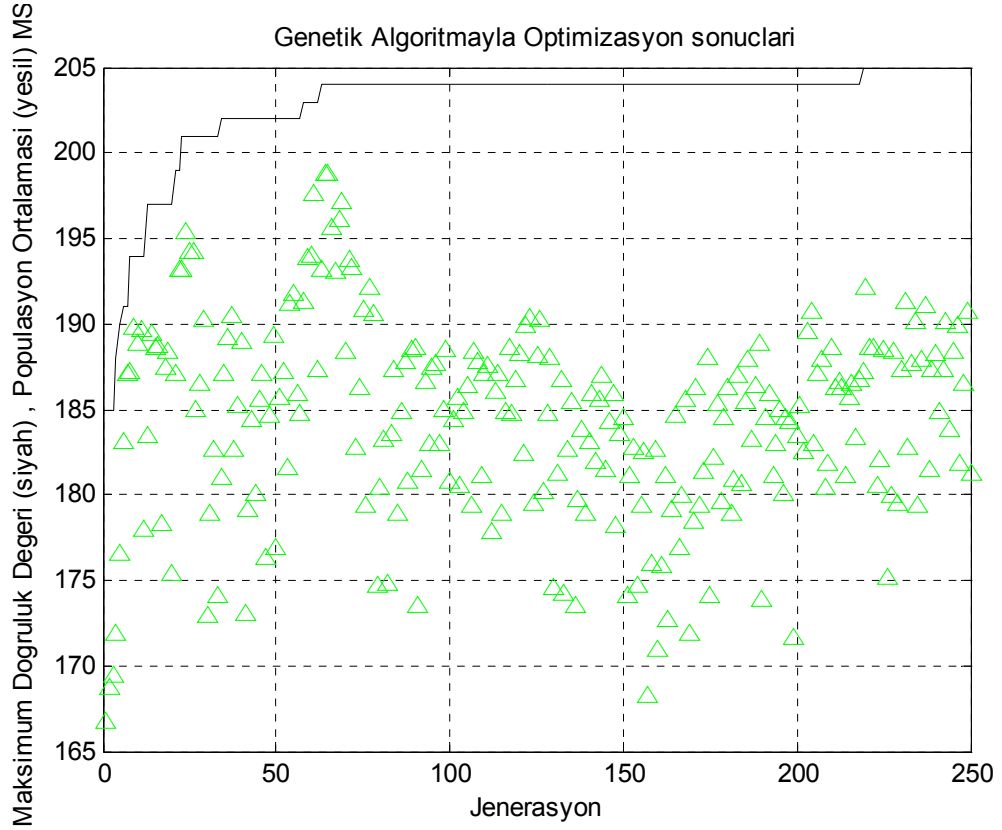
Çizelge 4.3 Tiroid veri kümesine ait eğitime ve test örnek sayıları

Tiroid	Toplam Veri Kümesi	Eğitime Veri Kümesi	Test Veri Kümesi
Sınıf 1	150	25	125
Sınıf 2	35	7	28
Sınıf 3	30	3	27

Çizelge 4.4 Tiroid veri kümesi için doğru sınıflanan örnek sayıları

Tiroid	Sınıf 1	Sınıf 2	Sınıf 3
RBF-GA	149	32	24
RBF	150	0	1

Şekil 4.3’de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir.

**Şekil 4.3** Tiroid hastalığı-genetik optimizasyon sonuçları

Bu veri kümesi için jenerasyon sayısı 250 olarak alınmış ve gauss fonksiyonunun genişliği 7 olarak belirlenmiştir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir

ağı programında da kullanılmıştır. σ değeri yine 7 alınarak program çalıştırıldığında elde edilen test başarımlar oranları Çizelge 4.13’de verilmiştir.

4.3 Escherichia Coli Veri Kümesi

E.coli, bazı türlerinin insan ve hayvan sağlığı için güçlü toksik madde içerdiği bir bakteridir. Bu çalışmadaki veri kümesi 1996 yılında Nakai’den alınmış ve Horton tarafından desteklenmiştir. E.coli veri kümesi 8 sınıfa bölünmüş 336 örneğe sahiptir. Her bir örnek dizilim adı, sınıf adı ve 8 nitelik ile ifade edilir. E.coli bakterisine ait veri kümesinin sınıf dağılımı Çizelge 4.5’te göstermektedir. Bu eğitim ve test veri dağılımını kullanarak elde edilen doğru sınıflanan örnek sayısı Çizelge 4.6’ da verilmiştir. Genetik algoritma kullaarak elde edilen RBF optimizasyonu Şekil 4.3’te gösterilmektedir. Bu veri kümesi ile çalışılırken jenerasyon sayısı 300 olarak tanımlanmıştır.

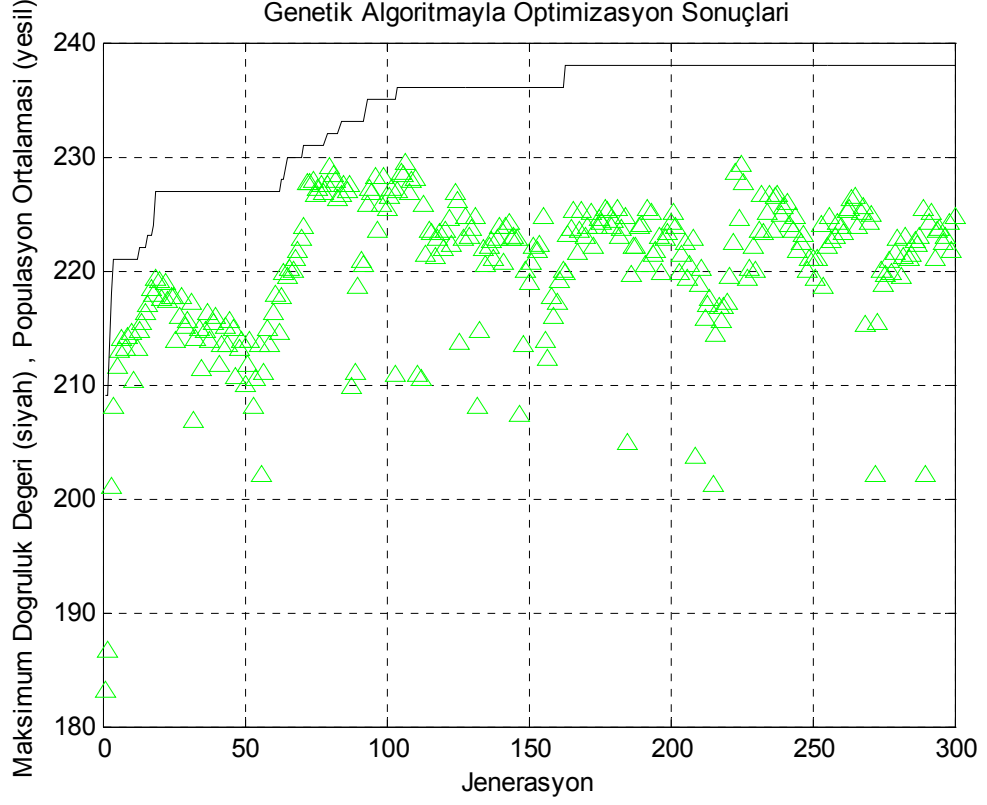
Çizelge 4.5 E.coli bakterisine ait eğitime ve test örnek sayıları

Sınıf	Toplam Veri Kümesi	Eğitime Veri Kümesi	Test Veri Kümesi
cp	143	23	120
im	77	17	60
pp	52	12	40
imU	35	5	30
om	20	3	17
omL	5	2	3
imL	2	1	1
imS	2	1	1

Çizelge 4.6 E.coli bakterisi için doğru sınıflanan örnek sayısı

E.COLI	RBF-GA	RBF
Sınıf 1	129	95
Sınıf 2	46	27
Sınıf 3	37	43
Sınıf 4	16	5
Sınıf 5	6	1
Sınıf 6	3	2
Sınıf 7	1	0
Sınıf 8	0	0

Şekil 4.3’de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir.



Şekil 4.4 E.coli bakterisi- genetik optimizasyon sonuçları

Bu veri kümesi için jenerasyon sayısı 300 olarak alınmış ve gauss fonksiyonun genişliği 924 olarak belirlenmiştir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir ağı programında da kullanılmıştır. σ değeri yine 924 alınarak program çalıştırıldığında elde edilen test başarımları Çizelge 4.13’de verilmiştir.

4.4 Fetus Veri Kümesi

Bu uygulamada amaç, fetus gelişiminin normal veya anormal olup olmadığının belirlenmesidir. Bunun için, gebelik sırasında fetusla plesanta arasındaki bağlantıyı sağlayan ve anneden gıda alımı ile bebekten atıkların temizlenmesini sağlayan bağ damarlarındaki (Umbilical Artery-UA) kan akış hızı dalga şekillerinin dopler ultrason ölçüm sonuçları kullanılmıştır. Haftalık ultrason ölçüm sonuçları ağa giriş bilgisi olarak verilmiş ve fetus gelişimi hakkında çıkış bilgisi elde edilmiştir. Burada örnekler dört parametre ile temsil

edilmektedir. İlk parametre, hafta olarak gebelik süresidir (Week Index-WI). Diğerleri ise; ultrason ışını ile kan akış yönü arasındaki açıdan bağımsız olan, kan akış hızı dalga şekilleri ölçüm sonuçlarıdır. Bunlar: Sistolik/Diastolik(S/D) oranı, Direnç İndeksi (Resistance Index-RI) ve Pulsatility İndeksi (PI) dir. 143'ü normal, 67'si anormal olmak üzere toplam 210 örnek ele alınmıştır. Çizelge 4.7, veri kümesine ait eğitim ve test örnek sayılarını göstermektedir. İlk sınıftan 23, ikinci sınıftan 17 örnek alınarak ağ eğitildiğinde doğru sınıflandırılan örnek sayısı Çizelge 4.8' de verilmiştir.

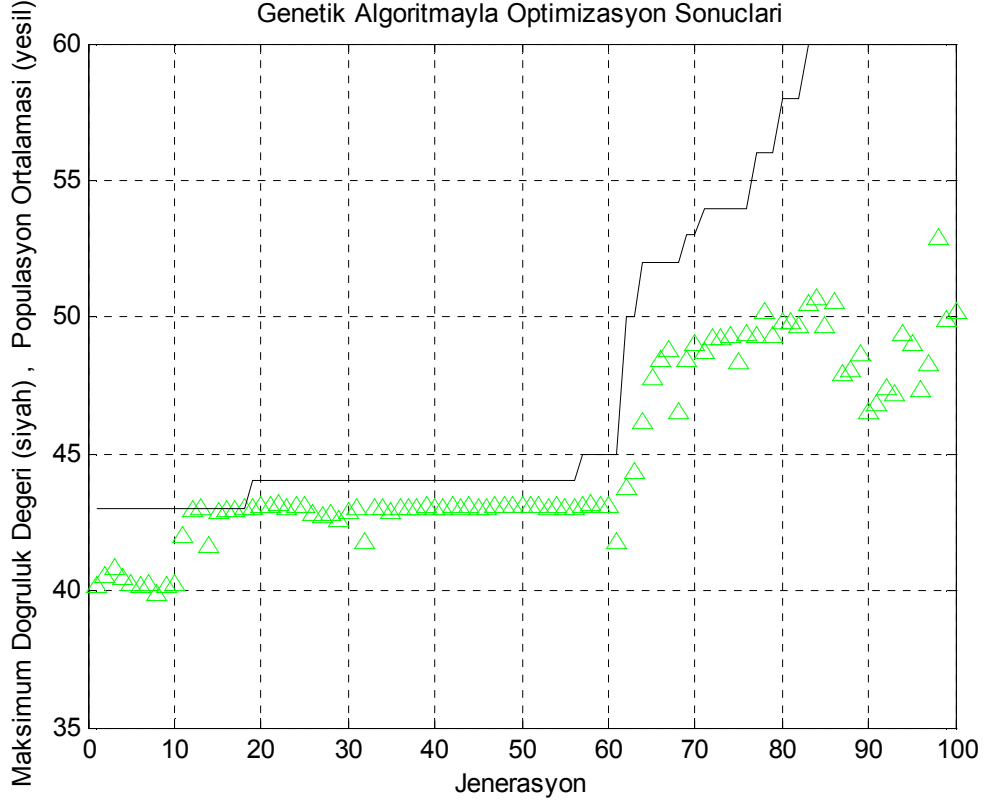
Çizelge 4.7 Fetus veri kümesi için eğitim ve test örnek sayıları

Sınıf	Toplam Veri Kümesi	Eğitim Veri Kümesi	Test Veri Kümesi
Normal	143	23	120
Anormal	67	17	60

Çizelge 4.8 Fetus veri kümesi için doğru sınıflanan örnek sayıları

Fetus	RBF-GA	RBF
Sınıf 1	40	28
Sınıf 2	20	18

Şekil 4.4'de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir.



Şekil 4.5 Fetus gelişimi-genetik optimizasyon sonuçları

Bu veri kümesi için jenerasyon sayısı 100 olarak alınmış ve gauss fonksiyonun genişliği 0.464 olarak belirlenmiştir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir ağı programında da kullanılmıştır. σ değeri yine 0.464 alınarak program çalıştırıldığında elde edilen test başarımlar oranları Çizelge 4.13’ de verilmiştir.

4.5 Cam Veri Kümesi

Cam veri kümesi Araştırma Merkez Kuruluşu, Berkshire’ da yaratılmıştır. Veri kümesi 6 sınıfa ayrılmış 214 örnekten oluşmaktadır. Veri kümesindeki her bir örnek tanım numarası, dokuz kimyasal ölçüm (Ri: kırıcı indis, Na:sodyum, Mg: magnezyum, Al:alüminyum, Si:silicon, K:potasyum, Ca:kalsiyum, Ba:baryum ve Fe:demir) ve 1 ile 7 arasında bir sınıf numarası ile ifade edilir. 4. sınıf veri kümesinden kaldırılmış ve bu çalışmada hiç kullanılmamıştır. Çizelge 4.9’da veri kümesinin sınıf dağılımı gösterilmektedir.

Çizelge 4.9 Cam veri kümesine ait eğitime ve test kümesi örnek sayıları

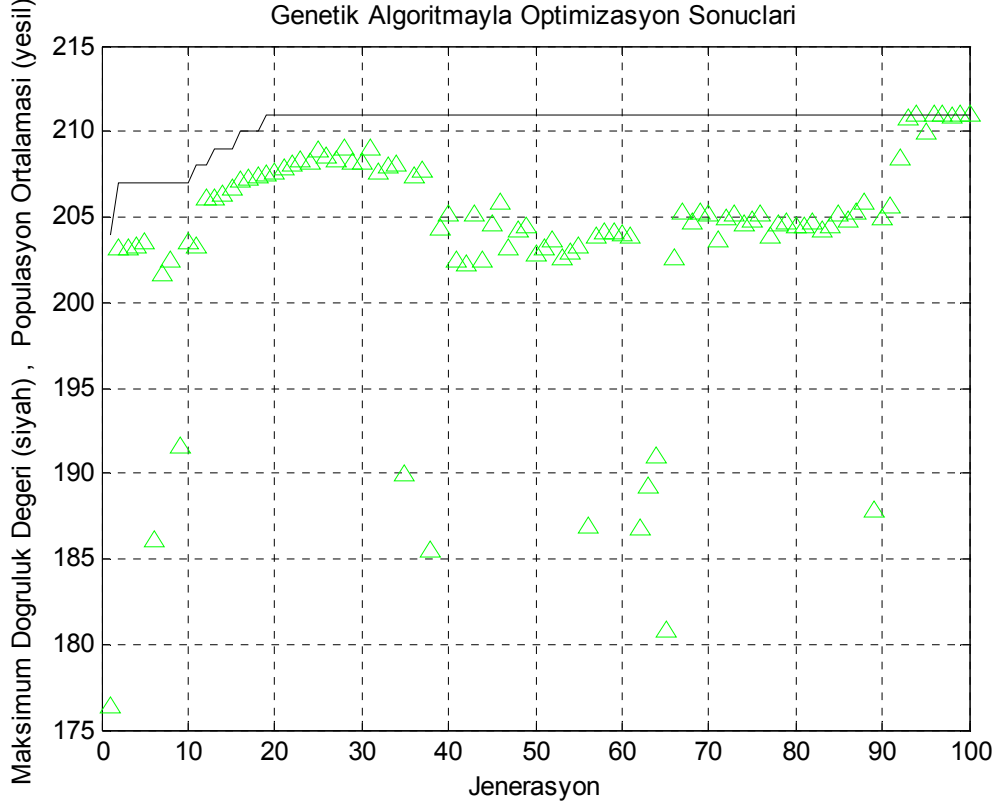
CAM	Toplam Veri Kümesi	Eğitime Veri Kümesi	Test Veri Kümesi
Sınıf 1	70	23	47
Sınıf 2	76	17	59
Sınıf 3	17	7	10
Sınıf 5	13	3	10
Sınıf 6	9	2	7
Sınıf 7	29	9	20
Toplam	214	61	153

Birinci sınıftan 23, ikinci sınıftan 17, üçüncü sınıftan 7, beşinci sınıftan 3, altıncı sınıftan 2 ve yedinci sınıftan 9 örnek alınarak ağ eğitildiğinde RBF-GA ve RBF algoritmalarında doğru sınıflanan örnek sayısı Çizelge 4.10’de verilmiştir.

Çizelge 4.10 Cam veri kümesi için doğru sınıflanan örnek sayıları

CAM	RBF-GA	RBF
Sınıf 1	70	70
Sınıf 2	76	9
Sınıf 3	16	3
Sınıf 5	11	3
Sınıf 6	9	2
Sınıf 7	29	9

Şekil 4.5’de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir.



Şekil 4.6 Cam veri kümesi-genetik optimizasyon sonuçları

Bu veri kümesi için jenerasyon sayısı 100 olarak alınmış ve gauss fonksiyonun genişliği 10.9 olarak belirlenmiştir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir ağı programında da kullanılmıştır. σ değeri yine 10.9 alınarak program çalıştırıldığında elde edilen test başarımları Çizelge 4.13’ de verilmiştir.

4.6 Lens Veri Kümesi

Lens veri kümesi, doğal özellik-değer çiftlerinin tüm mümkün olabilecek kombinasyonlarının belirtildiği, bütün bir veri kümesidir. Veri kümesinin yalnızca 24 örneği vardır. Bunlar tanım numarası, dört özellik (1: genç olma, 2: pre-presbyopic, 3: presbybic), görülür reçete (1: miyop, 2: hipermetrop), astigmat (1: evet, 2: hayır), gözyaşı üretim oranı (1: az, 2: normal) ve sınıf sayısıdır. Birinci sınıf 4, ikinci sınıf 5, üçüncü sınıf 15 örneğe sahiptir. Çizelge 4.11’de eğitim ve test örnek sayıları verilmiştir. Birinci sınıftan 2, ikinci sınıftan 3 ve üçüncü sınıftan 5 örnek alınarak ağ eğitildiğinde doğru sınıflanan örnek sayısı Çizelge 4.12’de verilmiştir.

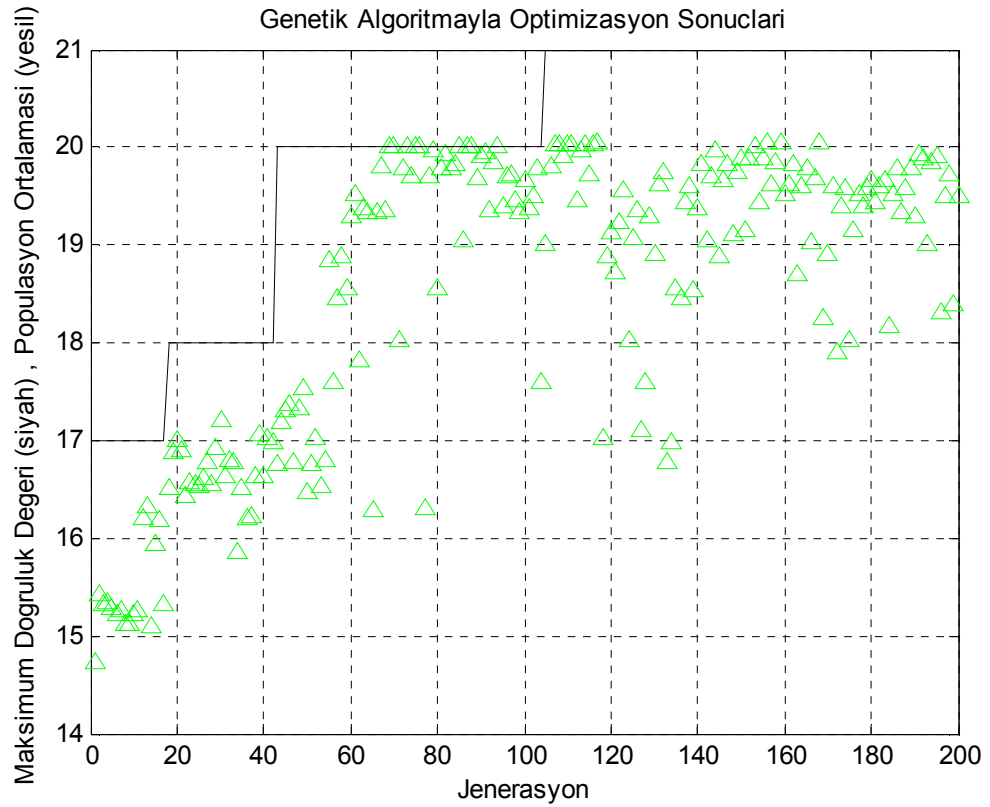
Çizelge 4.11 Lens veri kümesi için eğitim ve test örnek sayıları

Lens	Toplam Veri Kümesi	Eğitme Veri Kümesi	Test Veri Kümesi
Sınıf 1	4	2	2
Sınıf 2	5	3	2
Sınıf 3	15	5	10

Çizelge 4.12 Lens veri kümesi doğru sınıflanan örnek sayıları

Lens	RBF-GA	RBF
Sınıf 1	4	2
Sınıf 2	5	5
Sınıf 3	12	12

Şekil 4.5’de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin jenerasyon sayısına bağlı olarak değişimi gösterilmektedir.

**Şekil 4.7** Lens veri kümesi-genetik optimizasyon sonuçları

Bu veri kümesi için jenerasyon sayısı 200 olarak alınmış ve gauss fonksiyonun genişliği 764 olarak belirlenmiştir. Aynı veri kümesi sadece RBF ağı kullanılarak çalıştırılan bir yapay sinir ağı programında da kullanılmıştır. σ değeri yine 764 alınarak program çalıştırıldığında elde edilen test başarımları Çizelge 4.13’ de verilmiştir.

Çizelge 4.13 Tüm veri kümelerine ait test başarımları ve sigma değerleri

Test Başarım Oranı	RBF-GA(%)	RBF(%)	Sigma Değeri
İris	99.1	67.5	523
Tiroid	94.4	64.4	7
E.coli	63.9	40	924
Fetus	95	73	0.464
Cam	98	22	10.9
Lens	78	64	764

Tüm veri kümeleri ile yapılan optimizasyon işlemi sonucunda GA-RBF yaklaşımının RBF’ten daha başarılı olduğu görülmektedir. GA-RBF’in başarımları farklı veri kümeleri üzerinde de denenebilir. Ama özellikle, seçilen veri kümelerin zor sınıflama karakteristiği göz önüne alınırsa, GA-RBF’in başarımlarının değişmeyeceği aşikardır.

5. SONUÇLAR

Bu çalışma, bir yapay sinir ağı modeli olan Radyal Temelli Fonksiyon ağlarında, kritik öneme sahip olan merkezlerin seçimi ve Gauss fonksiyonu genişliğinin belirlenmesi üzerine Genetik Algoritmalarından faydalanmayı amaç edinmiştir. Genetik algoritmalar evrim teorisi üzerine kurulu olduğundan en iyinin seçimi prensibine bağlı olarak kritik parametrelerin optimizasyonunda son derece başarılı bir performans göstermektedir. RBF’de uyarlanabilecek serbest parametreler; merkez vektörleri, radyal fonksiyonların genişliği ve çıkış katman ağırlıklarıdır. Çıkış katmanı doğrusal olduğundan ağırlıklar, eğim düşme yada doğrusal en iyileme yöntemleri ile kolayca bulunabilir. Merkezlerin, girişler arasından rastgele ve sabit olarak seçildiği RBF ağlarında, genetik algoritmalar en iyi değere sahip olanların seçilmesi yolunda RBF başarımını güçlendirecek bir yaklaşım sergilemektedir.

Genetik Algoritmada ilk olarak bir populasyon yaratılır. Bireylerin kromozomlarla ifade edildiği bu yapıda, her bir birey için uygunluk değerlendirmesi yapılır. Böylelikle uygunlukları küçük olan kromozomlar elenecek, daha iyi uygunluk fonksiyonuna sahip olan kromozomlar seçilecektir. Optimizasyon kriterini geçen değerler, en uygun çözümü sunar. Ancak bu değeri geçemeyen populasyon elemanları için seçim işlemi uygulanmaktadır. Bu istenen çözüm sağlanıncaya kadar devam ettirilir. Seçilen kromozomlar arasında çaprazlama ve gerekirse mutasyon işlemi yapılır ve kromozomun uygunluk değerine tekrar bakılır. Bu işlem tatmin edici bir sonuç alana kadar devam eder. Bu şekilde RBF ağı sahip olabileceği en iyi merkez değerlerine kavuşmuş olur ve optimal konfigürasyonun sağlanması sayesinde, sınıflama problemi daha iyi bir sonuç verir.

Algoritma pek çok probleme uyarlanabilir. Bu tez çalışmasında çeşitli veri kümelerinin sınıflanması problemi üzerinde durulmuştur. İris çiçeği, tiroid hastalığı, e.coli bakterisi, fetus gelişimi, cam ve lens veri kümeleri üzerinde algoritmanın başarımı gözlenmiştir.

Algoritma az sayıda eğitme kümesi ile çalışmaktadır, ancak belirlenen sayıdaki girişler elle programa girildiğinden, her bir veri kümesi ayrı uğraş gerektirmektedir. Yazılım modüler hale getirilerek daha kullanıcı dostu tasarıma dönüştürülebilir.

Programın en önemli avantajı, zor sınıflanan veri kümeleri üzerinde başarılı bir performans göstermesidir ve bunu az sayıda merkez seçerek başarmaktadır.

Genetik algoritmaların çalışma süresi RBF’e göre uzun da olsa, RBF’in sağladığı başarımın yanında gözardı edilebilir. Çalışma süresi jenerasyon sayısı ile doğru orantılı olduğundan,

jenerasyon sayısı azaltılabilir. Ancak yüksek jenerasyon sayısı ile algoritma yine de denenmeli ve başarımın artıp artmayacağı gözlenmelidir. Bazı veri kümeleri düşük sayıda jenerasyon ile aynı başarımı gösterebilir. Örneğin iris veri kümesini 50 jenerasyondan fazla çalıştırmaya gerek olmadığı grafikten (Şekil 4.2) gözlenebilmektedir. Tiroid veri kümesi de en yüksek populasyon ortalamasını (Şekil 4.3) 50 ve 100. jenerasyonlar arasında vermektedir. Bu nedenle en iyi başarım, veri kümesini farklı jenerasyon sayılarında çalıştırarak bulunabilir.

Genetik algoritmalarla yapılan bu optimizasyon problemi farklı yapay sinir ağı modelleriyle denenebilir. Tezde kullanılan algoritmanın deneme amacıyla, sadece birkaç veri kümesi için RBF'nin benzeri bir ağ yapısı olan GRNN (Generalized Regression Neural Networks) modeliyle çalıştırılması sonucu da genetik algoritma optimizasyonunun, optimizasyon yapılmaksızın gerçekleştirilen simulasyon sonuçlarından yine daha iyi olduğu gözlenmiştir. Bu da sonraki çalışmalar için bir adım olmaktadır.

KAYNAKLAR

Turgut, P., Arslan, A., (2001), "Sürekli Bir Kirişte Maksimum Momentlerin Genetik Algoritmalar ile Belirlenmesi", Cilt:3, Sayı:3, sh:1-9, Ekim 2001, DEÜ, İzmir.

Burdsall, B., Carrier, C., (1997), "GA-RBF: A Self-Optimising RBF Network", In: Proceedings of the Third International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA'97), pages 348--351. Springer-Verlag, April 1997.

Barreto, A., Barbosa, H., Ebecken, N., (2002), "Growing Compact RBF Networks Using a Genetic Algorithm", Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN'02).

Lacerda, E.G.M., Carvalho, A.C.P.L.F., Ludermir, T.B., "Evolutionary Optimization of RBF Networks" 2001 Jun;11(3):287-94.

Zuo, G., Liu, W., Ruan, X., "Genetic Algorithm Based RBF Neural Network for Voice Conversion", Proceedings of the 5" World Congress on Intelligent Control and Automation. June 15-19. 2004, Hangzhou. P.R. China.

Sheta, A.F., Jong, K., (2001), "Time-Series Forecasting Using GA-Tuned Radial Basis Functions", Information Sciences, Special Issue on Evolutionary Algorithms, Volume 133 , Issue 3-4 (April 2001), Pages: 221 – 228.

Mak, M.W., Cho., K.W., (1998), "Genetic Evolution of Radial Basis Function Centers For Pattern Classification", In *Proc. Of The 1998 IEEE International Joint Conference on Neural Networks*, pages 669-673, 1998. Volume 1.

Whitehead, B.A., Chocate, T.D., (1995), "Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths For Time Series Prediction", *IEEE Trans. On Neural Networks*, 7(4):869-880, July 1995.

Chen, S., Wu, Y., Luk, B.L., (1999), "Combined Genetic Algorithm Optimization and Regularized Orthogonal Least Squares Learning for Radial Basis Function Networks" *IEEE-NN*,10(5):1239, September 1999.

Whitehead, B.A., Chocate, T.D., (1993), "Evolving Space Filling Curves to Distribute Radial Basis Functions Over an Input Space", *IEEE Trans. On Neural Networks*, 5(1):15-23, 1993.

Maillard, E.P., Gueriot, D., (1997), "RBF Neural Networks, Basis Functions and Genetic Algorithm", In *Int. Conf. On Neural Networks*, pages 2187-2192. IEEE, 1997, Volume 4.

İşçi, Ö., Korukoğlu, S., (2003), "Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama", Cilt:10, Sayı:2, 2003, CBÜ, Manisa.

Beasley, D., Bull D.R., Martin R.R., (1993), "An Overview of Genetic Algorithms:Part 1. Fundamentals", *Univ. Computing*, 15(2), 58-69.

Beasley, D., Bull D.R., Martin R.R., (1993), "An Overview of Genetic Algorithms:Part 2. Research Topics", *University. Computing*, 15(4), 170-181.

- Michalewicz, Z., (1994), "Genetic Algorithms + Data Structure = Evolution Programs", AI Series, Springer-Verlag, New York.
- Uçaner, M., Özdemir, O. (2002), "Genetik Algoritmalar ile İçme Suyu Şebekelerinde Ek Klorlama Optimizasyonu", Cilt:17, No:4, 157-170, 2002, GÜ, Ankara.
- Öztürk, A., (2002), "Gerçek Sayı Kodlamalı Genetik Algoritmaların Optimizasyonda Kullanımı", Yüksek Lisans Tezi, 2002, İTÜ, İstanbul.
- Man, K.F., Tang, K.S., Kwong, S., Halang, W.A., (1997), "Genetic Algorithms for Control and Signal Processings", Springer-Verlag London Ltd.
- Rooij, A.J.F., Jain, L.C., Johnson, R.P., (1996), "Neural Network Training Using Genetic Algorithms", World Scientific Publishing Co.Pte Ltd.
- Dejong, K., (1980), "Adaptive System Design: A Genetic Approach", IEEE Transaction on Systems, Man, and Cybernetics , Vol. SMC-10 , no. 9, Sep 1980, pp. 566-574.
- Herrera, F., Cordón, O., Magdalena, L., Villar, P., (2001), "A Genetic Learning Process For The Scaling Factors, Granularity and Contexts of The Fuzzy Rule-Based System Data Base" Information Science, Vol 136, Num 1-4, pp 85-107. 2001.
- Yao, X., (1999), "Evolving Artificial Neural Networks", Proceedings of the IEEE, 87(9):1423-1447.
- Zhang, B.T., Mühlenbein, H., (1993), "Evolving Optimal Neural Networks Using Genetic Algorithms with Occam's Razor", Complex Systems, 7(3):199-220.
- Yurtoğlu, H., (2005), "Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi", Yayın No:DPT:2683, Ekonomik Modeller ve Stratejik Araştırmalar Genel Müdürlüğü, Şubat 2005.
- Haykin, S., "Neural Network, A Comprehensive Foundation" Prentice Hall, Second Edition, 1998.
- Çoban Y., Kapanoğlu B., Yıldırım T., (2003), "Standart Öklid mesafesi hesaplayan bir CMOS Analog Devrenin RBF Ağlarına Uyarlanması", Elektrik-Elektronik-Bilgisayar Mühendisliği 10. Ulusal Kongresi, sf. 350-352, İstanbul, Türkiye, 18-21 Eylül, 2003.
- Hatanaka, T., Kondo, N., Uosaki, K., (2003), "Multi-Objective Structure Selection for Radial Basis Function Networks Based on Genetic Algorithm", Evolutionary Computation, Volume 2, Issue , 8-12 Dec. 2003 Page(s): 1095 - 1100 Vol.2
- Yazıcı, G., Polat Ö., Yıldırım, T., (2006), "Genetic Optimizations for Radial Basis Function and General Regression Neural Networks", 5th Mexican International Conference on Artificial Intelligence, LNAI, Springer-Verlag, November 13-17, 2006, Apizaco, Mexico (kabul edildi).
- Yazıcı, G., Yıldırım T., (2006), "Genetik Algoritmalar İle Radyal Temelli Fonksiyon Ağlarının Optimizasyonu", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu 2006 (ASYU INISTA 2006) Bildiriler kitabı, syf:241, Yıldız Teknik Üniversitesi, İstanbul.

İnternet Kaynakları

- [1] Busetti, F., “Genetic Algorithms Overview”, www.researhindex.com.
- [2] Pohlheim, H., (1997), “GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with Matlab”, www.geatbx.com/docu/index.html.
- [3] Nolfi, S., Floreano, D., “Learning and Evolution”, www.research.com.
- [4] <http://www.ics.uci.edu/~mlearn/MLRepository.html>

ÖZGEÇMİŞ

Doğum Tarihi	01.01.1981	
Doğum Yeri	Zonguldak	
Ortaokul	1992-1999	Zonguldak Atatürk Anadolu Lisesi
Lise	1996-1999	Zonguldak Fen Lisesi
Lisans	1999-2003	Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü
Yüksek Lisans	2003-2006	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Müh. Anabilim Dalı, Haberleşme Programı

Çalıştığı Kurumlar

2004-2005	Telista Bilişim A.Ş
2005-Devam	Beko Elektronik A.Ş