İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEL SAYI KODLAMALI GENETİK ALGORİTMALARIN OPTİMİZASYONDA KULLANIMI

YÜKSEK LİSANS TEZİ

Uzay Müh. Ahmet ÖZTÜRK

Anabilim Dalı : UZAY MÜHENDİSLİĞİ

Programı : UZAY MÜHENDİSLİĞİ

MAYIS 2002

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEL SAYI KODLAMALI GENETİK ALGORİTMALARIN OPTİMİZASYONDA KULLANIMI

YÜKSEK LİSANS TEZİ

Uzay Müh. Ahmet ÖZTÜRK
(511991068)

Tezin Enstitüye Verildiği Tarih : 10 Mayıs 2002
Tezin Savunulduğu Tarih : 27 Mayıs 2002

Tez Danışmanı : **Prof.Dr. Süleyman TOLUN**
Diğer Jüri Üyeleri **Doç.Dr. Tülay YILDIRIM**
Doç.Dr. Kerem CIĞIZOĞLU

MAYIS 2002

ÖNSÖZ

Bu çalışmada optimizasyona yeni bir yaklaşım getirmiş olan Genetik Algoritmalar ve optimizasyonda kullanımları konu edilmiştir. Gerçel sayı kodlamalı Genetik Algoritmaların optimizasyonda kullanımı incelenerek uygulaması gerçekleştirilmiştir. Genetik Algoritmalar'ın tercih edilmesinin sebebi, sağladığı doğal ve esnek yöntemlerin yapay zeka, makine öğrenmesi, robotik ve diğer mühendislik konularındaki başarılı uygulamalarının motivasyonuna ek olarak geleneksel arama ve optimizasyon yöntemlerinin önerdiği faydalardan yararlanmak üzere onlarla melez kullanımının da mümkün olmasıdır.

Bu konuda araştırma yapmayı arzu edenler için bilgi sağlaması açısından konunun temelleri hakkında bilgi verilmesi ve optimizasyon problemlerinde kullanım hedefiyle yazılım geliştirilmesi amaçlanmıştır.

"Gerçel Sayı Kodlamalı Genetik Algoritmaların Optimizasyonda Kullanımı" gibi ilginç bir konuda sağladığı çalışma olanağı, çalışmadaki katkıları, yol göstericiliği ve bunların ötesinde doğruluk ve iyiniyet adına ayakta kalma kararlılığından dolayı danışmanım Prof. Dr. Süleyman Tolun'a teşekkürlerimi sunarım.

Normal şartlar altında sonuçlanması günle ifade edilebildiği halde çözüme kavuşması aylar süren danışman değişikliği girişimimi, göreve başlamasının hemen ardından sonuçlandıran Fen Bilimleri Enstitüsü Müdürü Prof. Dr. Mehmet Karaca'ya ve çalışma arkadaşlarına teşekkürlerimi sunarım.

İlgi, anlayış, teşvik ve yardımları sebebiyle Prof. Dr. Zeki Erim, Doç. Dr. İbrahim Özkol, Prof. Dr. Oktay Özcan, Prof. Dr. Ülgen Gülçat, Prof.Dr. Adil Yükselen, Doç. Dr. Serhat Şeker, Doç. Dr. Kerem Cığızoğlu, Doç. Dr Tülay Yıldırım, Doç.Dr. Mine Kalkan'a da ayrıca teşekkür ederim.

Varlıklarıyla keyif veren ve destek olan arkadaşlarıma ve yardımları sebebiyle özellikle Neslihan Doran'a teşekkürlerimi sunarım.

Varlığımın kaynağı ebeveynlerime emekleri, sağladıkları olanaklar ve duydukları güven dolayısıyla minnettarım.

Kanun, yönetmelik ve uygulamalar ile tanınmış hakların kullanımının engellenmediği, uydurma iddialarla bedel ödetilmeyen, yetkilerin kötüye kullanılmadığı, emeğin hiçe sayılmadığı ve entrikaların dönmediği bir sistem dilerim; en azından akademide. Aksi takdirde demezler mi kişiye "Profesör olmuşsunuz ama adam olamamışsınız" diye?

Mayıs 2002 Ahmet ÖZTÜRK

İÇİNDEKİLER

KISALTMALAR **IV**
TABLO LİSTESİ **V**
ŞEKİL LİSTESİ **VI**
SEMBOL LİSTESİ **VII**
ÖZET **VIII**
SUMMARY **IX**

1. GİRİŞ **1**

2. OPTİMİZASYON **2**
2.1. Tanımlar 2
2.2. Optimizasyon Problemlerinin Sınıflandırması 3
2.3. En Uygun(Optimal) Olma Koşulları 5
2.4. Arama Yöntemleri 7
2.4.1. Cebir Tabanlı Yöntemler 7
2.4.2. Sıralamalı Yöntemler 7
2.4.3. Rasgele Arama 7

3. EVRİMCİ ALGORİTMALAR **9**
3.1. Evrimci Programlama 9
3.2. Evrim Stratejileri 9
3.3. Genetik Algoritmalar 10
3.3.1. Temel İlkeler 11
3.3.2. Kromozom Gösterimi 11
3.3.3. Uygunluk Fonksiyonu 12
3.3.4. Olgunlaşmamış Yakınsama 12
3.3.5. Yavaş Sonlanma 12
3.3.5.1. Doğrusal Ölçekleme 13
3.3.5.2. Üssel Ölçekleme 13
3.3.5.3. Sigma Kesmesi 13
3.3.6. Seçim 13
3.3.6.1. Kesme Seçimi 14
3.3.6.2. Stokastik Evrensel Örnekleme 14
3.3.6.3. Rulet Tekerleği Seçimi 14
3.3.7. Çoğalma 14
3.3.8. Mutasyon 15
3.3.9. Yakınsama 16
3.3.10. Tersten Sıralama ve Yenidn Düzenleme 17
3.3.11. Çift Değerlilik ve Baskınlık 17
3.3.12. Epistasis 17

3.3.13. Aldanma 17
3.4. Genetik Algoritmalar Nasıl Çalışır? 18
3.4.1. Şema Teoremi 18
3.4.2. Yapıtaşı Hipotezi 19
3.5. Arama Uzayında Keşif ve Keşfin Kullanımı 19
3.6. Bilgi Tabanlı Teknikler 20
3.7. Kullanılabilirlik 21
3.8. Karşılaşılabilecek Genetik Algoritma Türleri 21
3.8.1. Basit Genetik Algoritma 21
3.8.2. Kararlı Durum Genetik Algoritması 21
3.8.3. Deme Genetik Algoritması 21
3.8.4. Düzensiz Genetik Algoritmalar 21
3.8.5. Breeder Genetik Algoritma 22

4. GENETİK ALGORİTMALAR'DA ÖĞRENME 23
4.1. Michigan Yaklaşımı 23
4.2. Pittsburgh Yaklaşımı 23
4.3. Yinelemeli Kural Öğrenme 24
4.4. Evrimci Yapay Sinir Ağları(EYSA) 24
4.4.1. EYSA Bağlantı Ağırlıklarının Evrimi 24
4.4.1.1. İki Tabanlı Gösterim 25
4.4.1.2. Gerçel Sayı Kodlamalı Gösterim 25
4.4.1.3. Evrimci Eğitim ve Geriyayınımlı Eğitim Karşılaştırması 27
4.4.1.4. Hibrid Eğitim Yaklaşımı 27

5. UYGULAMA 28

6. SONUÇLAR VE TARTIŞMA 31

KAYNAKLAR 32

EKLER 34

ÖZGEÇMİŞ 54

KISALTMALAR

GA	: Genetik Algoritmalar
EA	: Evrimci Algoritmalar
EP	: Evrimci Programlama
ES	: Evrimci Stratejiler
IRL	: Yinelemeli Kural Öğrenme (Iterative Rule Learning)
BGA	: Üretici(Breeder) Genetik Algoritması
YSA	: Yapay Sinir Ağı
EYSA	: Evrimci Yapay Sinir Ağı
MSE	: Hata karelerinin ortalaması(Mean Squared Error)
RMS	: Hata karelerinin ortalamasının karekökü (Root Mean Square)
AR	: Otoregresif(Autoregressive) Model
MA	: Hareketli Ortalama(Moving Average) Modeli
ARMA	: Otoregresif-Hareketli Ortalama Modeli

TABLO LİSTESİ

Sayfa No

Tablo B.1: İnsan beyni ve bilgisayar karşılaştırması …… 43

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 : Optimizasyon problemlerinin sınıflandırması..... 3
Şekil 2.2a : Kısıtsız optimizasyonda karşılaşılan minimum türleri..... 6
Şekil 2.2b : Kısıtlı optimizasyonda karşılaşılan minimum türleri..... 6
Şekil 3.1 : Genetik Algoritma akış diagramı..... 11
Şekil 3.2 : Stokastik Evrensel Örnekleme..... 14
Şekil 3.3 : Rulet Tekerleği Seçimi..... 15
Şekil 3.4 : Tek noktalı çaprazlama..... 15
Şekil 3.5 : Çok noktalı çaprazlama..... 15
Şekil 3.6 : Tek noktada mutasyon..... 16
Şekil 3.7 : Çok noktada mutasyon..... 16
Şekil 5.1 : Evrimci Yapay Sinir Ağı genel yapısı..... 26
Şekil 6.1 : Eğitimi gerçekleştirilen YSA topolojisi 28
Şekil 6.2 : Kromozomların isimlendirilmesi..... 29
Şekil 6.3 : Ağırlık değerlerinin kromozomu oluşturan diziliş tarzı..... 29
Şekil A.1 : EYSA'nın eğitiminde kullanılan veriler..... 35
Şekil A.2 : Zaman serisinin tüm değerlerine karşılık EYSA yazılımı GANN ve AR(3) simulasyonu 36
Şekil A.3 : Modellemede kullanılan gerçek değerlere karşılık EYSA yazılımı GANN ve AR(3) simulasyonu..... 37
Şekil A.4 : Modelin sınanmasında kullanılan gerçek değerlere karşılık EYSA yazılımı GANN ve AR(3) simulasyonu..... 38
Şekil A.5 : Adaptif öğrenme katsayılı YSA'nın çeşitli denemeler sonucu ulaşabildiği RMS değerleri..... 39
Şekil B.1 : Gerçek sinir hücresinin şematik gösterimi..... 42
Şekil B.2 : Yapay Sinir Hücresi..... 43
Şekil B.3 : Bias girişli Yapay Sinir Hücresi..... 44
Şekil B.4 : Eşik fonksiyonu..... 44
Şekil B.5 : Kısmi doğrusal fonksiyon..... 45
Şekil B.6 : Lojistik sigmoid fonksiyonu..... 45
Şekil B.7 : Hiperbolik tanjant fonksiyonu..... 46
Şekil B.8 : Tek tabakalı YSA..... 46
Şekil B.9 : Çok tabakalı YSA..... 47
Şekil B.10 : YSA'nın eğitimi..... 48
Şekil B.11 : Lojistik sigmod fonk.un eğim parametresiyle değişimi..... 51
Şekil B.12 : YSA ile zaman serisi modelleme işleminin gösterimi..... 53

SEMBOL LİSTESİ

x	**:** Bağımsız değişkenleri içeren vektör
i	**:** Tamsayı
f(x),g(x)	**:** Fonksiyon ifadesi
$\mathbf{f_i}$	**:** Uygunluk değeri
$\mathbf{o_i}$	**:** Hedeflenen uygunluk değeri
$\mathbf{H_{ij}}$	**:** Hessian matrisi
t	**:** Hedef değerler
$\mathbf{w_{ij}}$	**:** Bağlantı ağırlıkları
$\mathbf{I_j}$	**:** Giriş sinyalleri toplamı
a, b, d, k, q	**:** Sabitler
A	**:** Sabit değerlerden oluşmuş matris
V(x)	**:** Bağımsız değişken x'in değer kümesi
η	**:** 1. Komşuluk bölgesini tanımlamakta kullanılan keyfi değişken 2. YSA'da öğrenme katsayısı
N(y, η)	**:** y'nin η komşuluğundaki noktaların oluşturduğu küme
P, n	**:** Topluluk nüfusu
λ	**:** Yeni bireylerin sayısı
σ	**:** Kromozom topluluğunun uygunluk değerinin standard sapması
Npointer	**:** Örneklemeyle seçilecek birey sayısı
Φ	**:** Aktivasyon fonksiyonu
ε	**:** Hata terimi
μ	**:** Momentum terimi
α	**:** Lojistik sigmoid fonksiyonunun eğim parametresi

GERÇEL SAYI KODLAMALI GENETİK ALGORİTMALARIN OPTİMİZASYONDA KULLANIMI

ÖZET

Bu çalışmada optimizasyona yeni bir yaklaşım getirmiş olan Genetik Algoritmalar konu edilmiştir. Genetik Algoritmalar, esas itibarıyla evrim sürecinden esinlenerek önerilmiş arama yöntemleridir. Parametrelerin kendisi yerine parametrelerin oluşturduğu dizilişler üzerinden işlem yapar ve bu işlem çözüm adayı olarak kodlanmış birden fazla bireyle yürütülür. İki-tabanlı gösterimle, gerçel sayılarla veya sembolik olabilen kodlamada tercih gerçel sayılardan yana kullanılmıştır.

Geleneksel arama yöntemlerinden farklı olarak yalnızca uygunluk(hedef) fonksiyon değerlendirmesi yapan GA, türev veya benzeri yardımcı bilgilere ihtiyaç duymaz. 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, rasgele verilerin modellenmesi için çok uygundur. Bu amaçla genetik operatörler geliştirilmiştir.

Genetik Algoritmalar'ın öğrenme yeteneğine sahip sistemlere ve bu sistemlerin yoğun kullanıma sahip olan temsilcisi Yapay Sinir Ağları'na yönelik, Evrimci Yapay Sinir Ağları(EYSA) adıyla anılan uygulamaları da mevcuttur. Ayrı ayrı, çeşitli amaçlarla kullanılan bu yaklaşımların etkileşimli kullanımı konu olduğunda bahsedilebilecek uygulamalar hakkında da bilgi ve örnek uygulama sağlanmıştır. Sonrası için temel teşkil edecek nitelikte olan uygulamada yerel minimum problemi yaşamaksızın Yapay Sinir Ağının bağlantı ağırlıklarının optimizasyonu gerçekleştirilmiştir.

REAL CODED GENETIC ALGORITHMS IN OPTIMIZATION

SUMMARY

In this study, Genetic Algorithms that has introduced a new approach to the optimization is coped with. Essentially Genetic Algorithms are search methods that are inspired of natural evolution. It processes through the strings of parameters instead of the parameters themselves. Real valued representation is preferred to the other alternatives, namely, binary and symbolic representation.

Differing from the traditional methods by its search method that requires just a functional evaluation, GA does not make use of derivatives or such auxiliary knowledge. Traditional GA applications intensively deals with numerical optimization of functions. With its approved success relative to the other methods in optimisation of multimodal, noisy and discontinuous functions, GA is an appropriate approach in modelling the random data.

GA has applications concerned with the systems that have capability of learning and their well-known representative, namely, the Artificial Neural Networks under the topic of Evolutionary Neural Networks. Information and an example application about the hybrid usage of these distinct approaches is provided. With its property to constitute a kernel for further studies, optimal values of the connection weights are computed avoiding the local minimum.

1. GİRİŞ

Canlı toplulukları nesiller boyunca, ilk defa Charles Darwin'in "Türlerin Kökeni (The Origin of Species)" adlı kitabında açıkça kaleme aldığı şekilde, doğal seçim süreci ve güçlülerin hayatta kalması prensiplerine göre evrim geçirmişlerdir[1]. Doğadan esinlenerek tasarlanan ve gerçekleştirilen Genetik Algoritmalar, evrim sürecinin hesaplamalı modellerinin esas alındığı bilgisayar tabanlı arama ve problem çözme yöntemleridir. Geleneksel Genetik Algoritmalar uygulamalarının çoğunluğu, fonksiyonların sayısal optimizasyonunda yoğunlaşmıştır. Süreksizlik içeren, çok-tepeli, gürültülü verilerin ve fonksiyonların optimizasyonunda diğer yöntemlerden daha başarılı olduğu gösterilmiş olan Genetik Algoritmalar'ın geleneksel arama yöntemleriyle melez kullanımı da mümkündür. Bunlardan başka, esasen öğrenme algoritmalarına dahil olmayan Genetik Algoritmalar bir kısım öğrenme işlemleri için güçlü arama yöntemleri de önerir[2,3].

Bu tez çalışmasında, geleneksel optimizasyon ve arama yöntemleri hakkında verilen kısa bilgiyi takiben Genetik Algoritmalar ve optimizasyonda kullanımlarına ilişkin bilgi sunulmuştur. Yoğun ilginin ve yıllardır süregelmiş çalışmaların sonuçlarının tamamından bahsetmenin mümkün olmaması sebebiyle geliştirilen uygulamanın anlaşılmasını, konunun geneline yönelik fikir sahibi olunmasını ve uygulama geliştirilmesini sağlamaya yeterli temel bilgiler verilmiştir.

2. OPTİMİZASYON

Optimizasyon, genel anlamda, bir kavram üzerinde çeşitlemelerle elde edilen bilginin işletilmesiyle gerçekleştirilen iyileştirme çalışmasıdır. Özelde, hesaplamalı bilimlerin dalı olarak "En iyi nedir, hangisidir?", "Tek çözüm eldeki midir?" gibi sorulara sayısal değerle ifade edilebilecek bir cevap arayan ve Matematiksel Optimizasyon resmi başlığıyla anılan alandır. Mühendislik, mimari, ekonomi, matematik ve yaşamın diğer alanlarında ortaya çıkan optimizasyon problemlerinin çözümüne dönük çeşitli yöntemler geliştirilmiştir[4,5].

Bu bölümde, farklı gruplardaki optimizasyon problemlerinin çözümüne uygun olduğu için sıkça kullanılan yöntemler hakkında kısa bilgi ve terminoloji verildikten sonra geçen yüzyılın son çeyreğinden itibaren kullanılmaya başlanan, doğal evrim sürecini taklit ederek iyileştirmeye çalışan evrimci algoritmalardan ve çalışmaya konu olması sebebiyle özellikle Genetik Algoritmalar'dan bahsedilmektedir.

2.1 Tanımlar

Bir optimizasyon probleminin gayesi, parametre sınırlarının muhtemel kısıtlarını gözeterek bir niteliğin iyileştirilmesini sağlayan uygun parametre kombinasyonunun bulunmasıdır. İyileştirilecek niteliğe hedef fonksiyon, arayış sırasında değeri değişebilen parametrelere kontrol veya karar değişkenleri ve parametrelerin alabileceği değerler ile ilgili sınırlamalara kısıt denir.

İyileştirme, hedef fonksiyonun en büyük veya en küçük değerinin bulunmasıyla gerçekleşir. Bir f fonksiyonunun en büyük değeri, $-f$ veya $1/f$ fonksiyonunun en küçük değeridir. Bu şartlarda, optimizasyon problemi matematik diliyle,

$x = (x_1, x_2, ..., x_n)^T$ n adet bağımsız değişkeni içeren vektör,

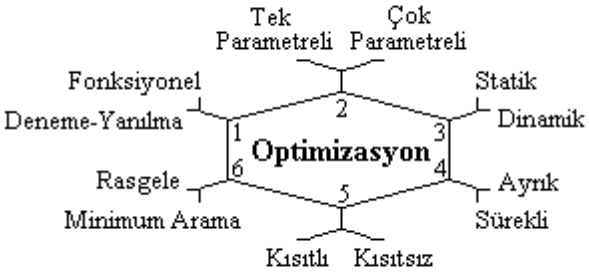
$c_i(x) = 0$, $i = 1,2,..., m'$ eşitlik kısıtlamaları,

$c_i(x) \geq 0$, $i = m' + 1,2,..., m$ eşitsizlik kısıtlamaları olmak üzere

$f(x)$ fonksiyonunun minimizasyonu olarak ifade edilir.

2.2 Optimizasyon Problemlerinin Sınıflandırması

Geliştirilmiş optimizasyon yöntemlerinin büyük bir bölümünün uygulanma alanı, belirli türdeki problemlerle kısıtlıdır. Sonuç alınabilmesi için problemin özellikleriyle uyumlu yöntem seçimi önem taşır. Optimizasyon problemlerinin sınıflandırması, hedef fonksiyonun, kısıtların, kontrol değişkenlerinin matematiksel özelliklerine ve arama yöntemine göre gerçekleşir. Şekil 2.1'de görüldüğü haliyle altı kategoride toplanabilecek yöntemler biribirini dışlamak zorunda değildir. Örneğin, bir dinamik optimizasyon problemi kısıtlı olabileceği gibi kısıtsız da olabilir[4].



Şekil 2.1. Optimizasyon problemlerinin sınıflandırması

Şekil 2.1'de görülen numaralandırmayı gözeterek kısaca bilgi verilecektir.

1. Deneme-yanılma yaklaşımıyla optimizasyon, gözlem verilerini üreten süreç hakkında pek bilgi sahibi olunmaksızın, verileri etkileyen parametrelerin ayarlanmaya çalışılmasıdır. Deneysel ve epey keşif yapılmasını sağlamış bir yöntemdir. Buna tezat yaklaşım ise bir sürecin matematik diliyle ifade edilebileceğini varsayan fonksiyonel yaklaşımdır. Genellikle teorisyenlerin tercih ettiği yaklaşımda çeşitli matematiksel yöntemler kullanılarak optimum çözüm bulunmaya çalışılır. Hedef fonksiyonun doğası, en önemli özelliklerin başında gelir. *b*'nin sabit değerli bir vektör ve *c*'nin bir sabit olduğu durumda, $f(x)$ ve kontrol değişkenleri x arasındaki ilişki 2.1 eşitliğine uyarsa fonksiyon doğrusaldır denir.

$$f(x) = b^T x + d \tag{2.1}$$

A, sabit değerlerden oluşmuş bir matris olmak üzere, ilişkiyi tasvir eden eşitlik (2.2) formunda ise fonksiyon kuadratiktir denir.

$$f(x) = x^T Ax + b^T x + d \tag{2.2}$$

2. Sadece bir tane hedef fonksiyon mevcutsa, optimizasyon 1-boyutludur. Daha fazla hedef fonksiyonun olduğu durumda çok-boyutlu optimizasyon gereklidir. Boyut artışına bağlı olarak optimizasyon güçleşir. Çalışmada 1-boyutlu optimizasyon üzerinde durulmuştur.
3. Dinamik optimizasyon, niteliğinde zamana bağlı değişim gözlenen verilerin kullanımıyla gerçekleştirilen iyileştirilme çalışmasıdır. Statik optimizasyonda, veriler zamandan bağımsızdır. Örnek problem olarak, aralarında muhtelif yollar olan iki şehir arasında ulaşım incelenecek olursa; Mesafe olarak en kısa yolun bulunması, yolların uzunluğu değişmediğinden, statik optimizasyon problemidir. Diğer şehire en kısa sürede ulaşmak ise yolların fiziksel durumu, trafik yoğunluğu gibi zamanla değişim gösteren parametrelere bağlı olduğundan dinamik optimizasyon problemidir.
4. Yapılacak işlerden oluşan bir listede uygun sıralamanın bulunması gibi, parametrelerin sonlu sayıda değer alabileceği problemler ayrık parametreli optimizasyon sınıfına girer. Uygun çözüm, parametrelerin belirli bir dizilişinde elde edildiği için kombinasyonel optimizasyon olarak da anılır. Parametrelerin sonsuz değer alma ihtimalinin olduğu durum, sürekli optimizasyon problemi olarak incelenir.
5. Çoğu zaman, parametrelerin alabileceği değerlere sınırlandırma getirilir. Kısıtlı optimizasyon, sınırlandırmaları tanımlayan eşitlik ve eşitsizlikleri, hedef fonksiyona katar. Kısıtsız optimizasyonda, parametreler herhangi bir değer alabilirler. Kısıtlı optimizasyonda, parametrelerin doğrusal denklem ve doğrusal kısıt terimleri olarak ifade edildiği durum doğrusal programlama sınıfını tanımlar. Parametrelerin doğrusal olmadığı durumda doğrusal olmayan programlama sınıfını tanımlar. Problemlerin çözümünde bilgisayar kullanımının artmasına rağmen, yukarıda geçen programlama terimi, bilgisayar programlamadan farklı anlam taşımaktadır.
6. İlk değerleri atandığı halde en düşük maliyet arayışına giren minimum arayıcı algoritmalar sıklıkla yerel minimuma yakalansa da hızlarından dolayı tercih

edilmektedirler. Daha yavaş işleyen ve rasgele değerler kullanarak arama yapan yöntemler, global minimumu bulmakta daha başarılıdır.

2.3 En Uygun(Optimal) Olma Koşulları

Yöntemlere geçmeden önce, aranan optimumun taşıdığı koşulları tanımlamak yerinde olacaktır. *f(x)* fonksiyonunun global optimumu x^*, dar anlamda (2.3) bağıntısı ile tanımlanır. *V(x)*, kontrol değişkeni *x*'in olası değerler kümesi olmak üzere,

$$f(x^*) < f(y) \quad \forall y \in V(x), y \neq x^* \tag{2.3}$$

Kısıt tanımlanmamış bir problem için *V(x)*, sonsuz geniştir.

Bir y^* noktası, yeterince yakın komşuluğundaki olası noktaların oluşturduğu küme $N(y^*, \eta)$ olmak üzere (2.4) bağıntısını sağlarsa *f(x)* fonksiyonunun güçlü yerel minimumu olarak anılır.

$$f(y^*) < f(y) \quad \forall y \in N(y^*, \eta), y \neq y^* \tag{2.4}$$

Benzer tarzda, zayıf yerel minimumun tanımı (2.5) eşitsizliği ile verilir.

$$f(y^*) \leq f(y) \quad \forall y \in N(y^*, \eta), y \neq y^* \tag{2.5}$$

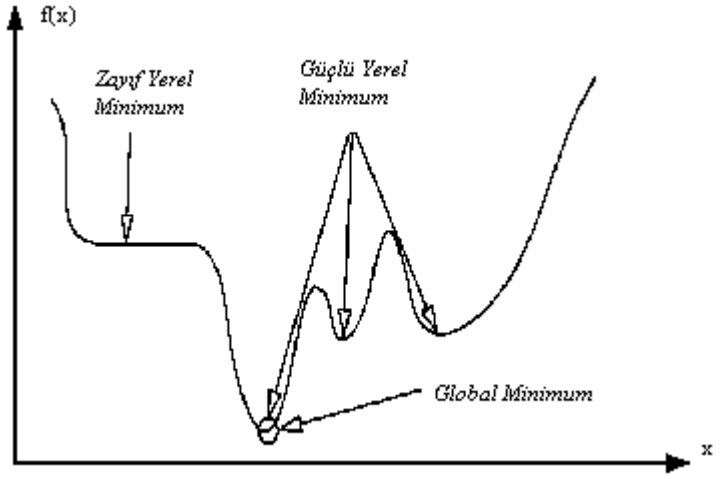
f(x) fonksiyonu için olası tüm *x* değerlerinde birinci ve ikinci türevlerin hesaplanabildiği durumda daha kolay tanımlanan optimal olma koşulları mevcuttur. *g(x), f(x)*'in gradyanı olmak üzere $g(x^*) = 0$ koşulunu sağlayan x^* noktası, *f(x)*'in durağan noktası olarak tanımlanır. İlk türev vektörü $\Delta f(x)$'in bileşenleri (2.6) denklemiyle hesaplanmaktadır.

$$g_i(x) = \frac{\partial f(x)}{\partial x_i} \tag{2.6}$$

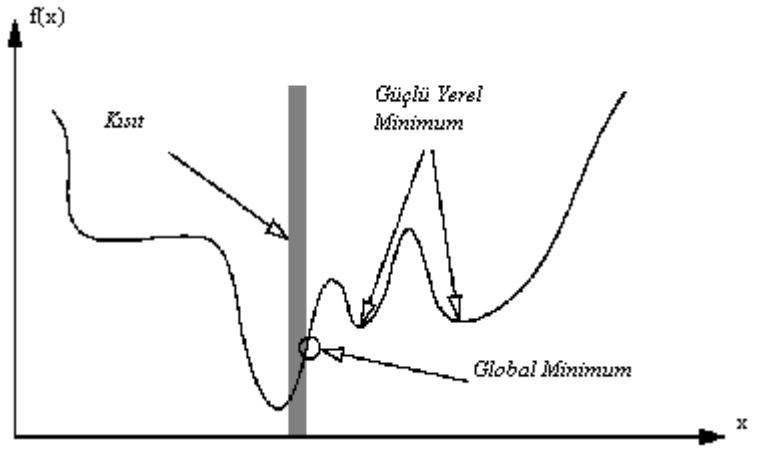
Bileşenleri (2.7) ile hesaplanan ikinci türevlerin oluşturduğu simetrik matrise Hessian matrisi denir.

$$H_{ij}(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \tag{2.7}$$

Eğer Hessian matrisi *H(x),* pozitif-belirli, yani, $u^T H(x^*)u > 0 \quad \forall u \neq 0$ ise x^* noktası, *f(x)*'in güçlü yerel minimumu olur. Tek değişkenli ve kısıtsız fonksiyonlardaki çeşitli durağan noktalar Şekil 2.2.a'da gösterilmiştir. Şekil 2.2.b'de de görüleceği gibi, kısıtlı optimizasyon probleminde durum biraz daha karmaşıktır; Kontrol değişkeninin alabileceği değerlerdeki sınırlama sebebiyle, global minimum, gerçek bir durağan nokta yerine bir uç değer alabilir. Kısıtlı problemler, dönüşüm yapılarak eşdeğer kısıtsız optimizasyon problemlerine dönüştürülebilmektedir[5]. Ancak, çalışmanın amacına hizmet etmemesi sebebiyle detayına girilmeyecektir.



Şekil 2.2.a Kısıtsız optimizasyonda karşılaşılan minimum türleri



Şekil 2.2.b Kısıtlı optimizasyonda karşılaşılan minimum türleri

2.4 Arama Yöntemleri

Bu bölümde, bir kısmı yalnızca kısıtlı uzayda uygulanabilirlikli çeşitli arama yöntemlerinden kısaca bahsedilecektir[5].

2.4.1 Cebir Tabanlı Yöntemler

Doğrudan ve dolaylı yöntemler olmak üzere iki grupta toplanırlar. Dolaylı yöntemler, hedef fonksiyonun gradyanını sıfıra eşitlemek suretiyle elde edilen ve genelde doğrusal olmayan denklem takımlarını çözerek yerel ekstremuma ulaşmaya çalışırken, doğrudan yöntemler, fonksiyon üzerinde belirlenen noktadaki gradyana bağlı hareket ederek arama yapar. Yerel yaklaşımla kısıtlı ve fonksiyonun türevlenebilirliğinin gerekli olması sebebiyle yeterince gürbüz(robust) değildirler. Buna karşın yoğun ilgi gören ve sıklıkla kullanılan bu yöntemler aramanın yönüne rehberlik etmesi için fonksiyonun gradyanını kullandığı için "Gradyan Azaltma Yöntemleri" adıyla da anılırlar. Sürekli fonksiyonların optimizasyonu için geliştirilmiş yöntemlerdir. Tek tepeli(veya tek çukurlu) fonksiyonlarda (unimodal functions) yüksek başarımlı olmasına rağmen çok-tepeli(veya çok çukurlu) fonksiyonlarda, tırmanılan ilk tepenin en yüksek(yada inilen çukurun en derin) olmaması halinde sonuç başarısızlıktır. Yerel-maksimum(minimum) sorunu olarak bilinen ve Şekil 2.2'de görülen bu durumda, yerel en yüksek(düşük) değere ulaşıldıktan sonra herhangi bir ilerleme kaydedilmesi söz konusu değildir.

2.4.2 Sıralamalı Yöntemler

Sonlu bir arama bölgesindeki veya ayrıklaştırılmış sonsuz arama uzayındaki elemanlardan her seferinde bir elemanı hedef fonksiyonda deneyerek arama yapan yöntemlerdir. Basitliği sebebiyle çekici görünse de arama uzayının genişliği aramada zaman maliyetini aşırıcı derecede arttırır.

2.4.3 Rasgele Arama

Geleneksel nitelendirmesinin yanlış olmayacağı yöntemlerin tatminkar başarımla çözemediği problemlerin sayısı, gerçek problemlerin süreksizlik, bozuntu ve çok-tepelilik içermesi nedeniyle azımsanmayacak derecededir. Sistematik yaklaşım öneren arama yöntemleri başarısızlığa uğradığı anda benimsenecek yaklaşım

sistematik olmayan yöntemler, yani rasgele arama yöntemleridir. Arama uzayındaki noktalar rasgele seçilip uygunluk değerlendirmesi yapılır. Arama, uygun değer bulunana kadar devam eder. Bu yöntem, keşiften elde edilen bilgiyi kullanmaz. Keşif ve keşfin kullanımına 3.5 bölümünde değinilmiştir.

Melez bir yaklaşım olarak, rasgele arama ve gradyan azaltmalı arama yöntemleri birleştirilerek yinelemeli tepe-tırmanma(çukur inme) araması yapılabilir. Bir tepe(çukur) bulunduktan sonra rasgele seçilen bir noktadan itibaren tepe(çukur) arama işlemi yeniden başlatılır. Çok fazla yerel maksimuma(minimuma) sahip olmayan fonksiyonlarda başarımı iyidir ve gerçekleştirilmesi açısından kolay bir yöntemdir. Ancak, her arama denemesi diğerinden izole olarak gerçekleştirildiğinden, eğrinin bütün şeklini gösteren resim elde edilemez. Arama süresince, arama uzayının her noktası eşit ilgi görür. Uygunluk değeri düşük bir bölgede değerlendirilen nokta sayısı, uygunluk değeri yüksek bir bölgede değerlendirilen nokta sayısına eşittir.

Rasgele arama kategorisine girdiği halde, bu noktaya kadar anlatılan yöntemlerden farklı yaklaşımla doğal evrim sürecini taklit ederek optimizasyon yapan yöntemler, Evrimci Algoritmalar(EA) genel başlığı altında incelenir. Evrimci Algoritmalar, tasarım ve gerçekleştirilmesinde evrim sürecinin hesaplamalı modellerinin esas alındığı bilgisayar tabanlı problem çözme sistemlerini tasvir etmekte kullanılan genel başlıktır. Evrimci Programlama, Evrim Stratejileri ve Genetik Algoritmalar bu alandaki baskın yöntemlerdir. Geliştiricilerinin benimsedikleri gösterim tarzı, seçim stratejileri, nüfus yönetimi ve genetik operatörlerin önemsenme derecesine göre farklılıklar arz ederler. EA'dan kısaca bahsedilecek ve özel olarak Genetik Algoritmalara(GA) odaklanılacaktır.

3. EVRİMCİ ALGORİTMALAR

Evrimci Algoritmalar, tasarım ve gerçekleştirilmesinde evrim sürecinin hesaplamalı modellerinin esas alındığı bilgisayar tabanlı problem çözme sistemlerini tasvir etmekte kullanılan genel başlıktır[2]. "Evrimci Programlama"(Fogel, Owens ve Walsh 1966), "Evrim Stratejileri" (Rechenberg 1973) ve "Genetik Algoritmalar" (Holland 1975) bu alandaki baskın yöntemlerdir. Geliştiricilerinin benimsedikleri gösterim tarzı, seçim stratejileri, nüfus yönetimi ve genetik operatörlerin önemsenme derecesine göre farklılıklar arz ederler.

3.1 Evrimci Programlama(EP)

Rasgele değerlerle yaratılan P adet kromozomun tamamı mutasyon operatörü vasıtasıyla P adet yeni birey elde etmekte kullanılır. İhtimalli turnuva seçimiyle, 2P nüfusa ulaşan topluluktan yeni nesili oluşturacak P birey seçilir. Topluluktaki uygunluk değeri en yüksek bireyin, erişilen optimumun kaybedilmemesi için her zaman var olması sağlanır.

3.2 Evrim Stratejileri(ES)

İlk değer ataması ve değerlendirmenin ardından rasgele seçilen ebeveyn kromozomların yeniden düzenlenmesiyle yeni bireyler elde edilir. Yeni bireylerin bir kısmı mutasyon operatörüne tabi tutulur. Yeni bireylerin sayısı ebeveynlerin sayısı P'den daha fazladır. Var olmaya devam edecek bireylerin belirlenmesinde benimsenen iki yöntem mevcuttur. (P, λ) ile temsil edilen ilk yöntemde, yaratılan λ adet yeni bireyden P adedi seçilip ebeveynlerinin yerlerini alması sağlanarak yeni nesil oluşturulur. İkinci yöntem $(P+\lambda)$ ile temsil edilir. Bu yöntemde ebeveynler ve yeni bireylerden uygunluk değeri en yüksek olan P adet birey yeni nesili oluşturmak üzere seçilir. EP'den farklı olarak yeniden düzenlenme önemli bir operatördür.

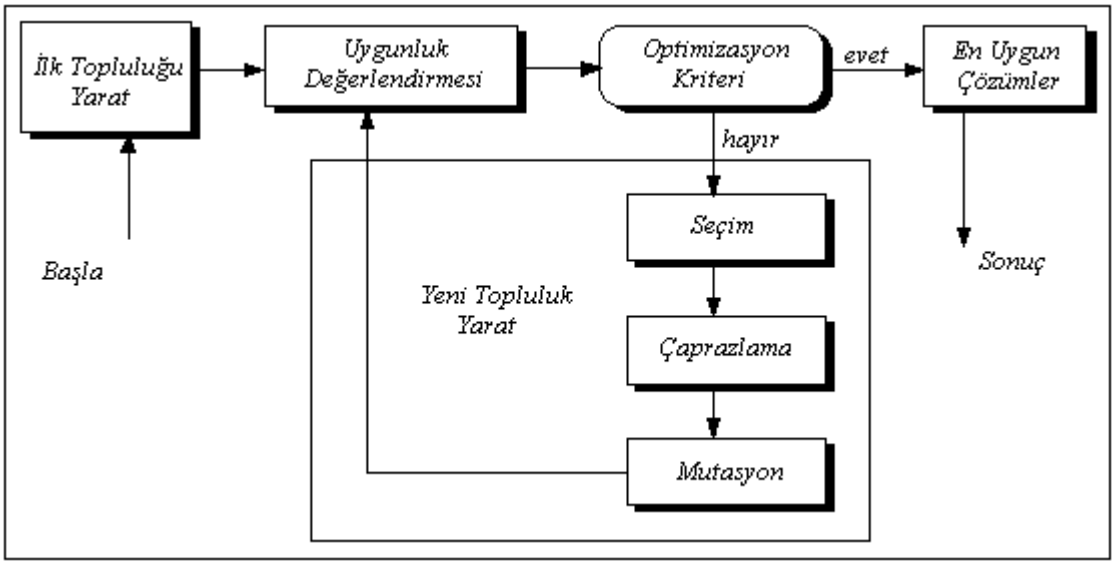
3.3 Genetik Algoritmalar

Metin boyunca GA kısaltması ile anılacak olan Genetik Algoritmalar, arama ve optimizasyon problemlerinin çözümünde başarıyla kullanılan ve popülerliği her geçen gün artan adaptif yöntemlerdir. Esin kaynağı canlı organizmalardır. Canlı toplulukları nesiller boyunca, ilk defa Charles Darwin'in "Türlerin Kökeni (The Origin of Species)" adlı kitabında açıkça kaleme aldığı şekilde, doğal seçim süreci ve güçlülerin hayatta kalması prensiplerine göre gelişmişlerdir[1]. Genetik Algoritmalar ismi ilk defa J.D. Bagley tarafından 1967 tarihli doktora tezinde kullanılmıştır[6]. Evrim sürecini model alan GA'nın temel prensipleri ise ilk defa J. Holland tarafından 1975 basımı "Doğal ve Yapay Sistemlerde Uyarlanma (Adaptation in Natural and Artificial Systems)" isimli kitapla ortaya konulmuştur[7,8]. Temelleri açıkça ortada olan evrim sürecini taklit eden GA, evrimde hangi biyolojik süreçlerin esas, hangilerinin tali önemde veya önemsiz olduğu halen araştırma konusu olması sebebiyle gelişime açık bir sahadır. Bundan başka, farklı disiplinlerdeki uygulamalar farklı bakış açıları kazandırmak suretiyle de GA'nın gelişimine katkı sağlamaktadırlar[8].

GA, her biri, eldeki problemin muhtemel çözümünü temsil eden dizilişler olan bireyler topluluğu üzerinde işlem yapar. Bu dizilişlerin her biri, benzetişim gereği, 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 nesilin içeriği, atalarının sahip olduğu uygunluk değeri arttırıcı özellikleri daha yüksek oranda barındırır. Böylelikle, uygun değerli parametreler nesil sayısı arttıkça toplulukta daha yaygın hale gelir. Eğer GA iyi tasarlanmış ise topluluk uygun bir çözüme yakınsayacaktır.
GA'nın gücünün kaynağı, diğer yöntemlerle çözümde zorlanabilinecek problemlerin de dahil olduğu geniş bir uzayda başarıyla uygulama geliştirme imkanı sağlaması ve yöntemin gürbüzlüğüdür(robust). GA, global optimumun belirlenmesini garanti etmese de kabul edilebilir derecede iyi bir çözümü, yeterince hızlı bulabilir. Özel çözüm tekniklerinin geliştirilmiş olduğu durumlarda dahi GA, daha etkin olarak sonucu bulmada veya hibrid kullanımla yöntemi iyileştirmede kullanılmaktadır[1].

3.3.1 Temel İlkeler

Şekil 3.1'deki gösterimle tanımlanabilecek olan 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 3.1 Genetik Algoritma akış diagramı

3.3.2 Kromozom Gösterimi

Uygun gösterimden kasıt, muhtemel çözümün bir dizi parametre ile temsil edilebileceği ilkesinin gayri ihtiyari kabulüne uygun olarak parametre sayısının belirlenmesidir. Parametreler, gösteriminde belirli bir alfabenin kullanıldığı genler olarak kodlanırlar. Genler bir araya gelerek kromozomları oluştururlar. Genlerin kromozom üzerindeki yerleri lokus adıyla anılır. Alfabe sembollerden, iki tabanlı sayılar olan {0,1}'den, tamsayılardan, gerçel sayılardan, matrislerden oluşabilir. GA'nın kullanılmaya başlandığı ilk zamanlarda, parametre değerleri için iki tabanlı gösterim daha uygun görülüp yaygın olarak kullanılmış olsa da gerçel değerli gösterim de mümkündür. Michalewicz, çalışmalarıyla göstermiştir ki arama uzayının doğasıyla uyumlu gösterimler daha iyi sonuçlar sağladığından daha verimlidir. Bu bağlamda, fonksiyon optimizasyonunda, kromozomların temsili için alt ve üst sınırlar dahilinde gerçel sayıların kullanımı alışılageldik iki tabanlı gösterime kıyasla

daha elverişlidir[9]. Yapılan çalışmada gerçel değerli gösterim kullanılmıştır. Gen bilimi terminolojisinde, belirli bir kromozomun gen içeriği, genotip olarak anılır. Genotip, bir organizmayı teşkil etmekte gerekli olan bilgiyi içerir. Teşkil edilmiş görüntü de fenotip olarak anılır. Aynı terimler GA için geçerlidir. Örneğin, bir tasarım işinde, belirli bir tasarımı temsil eden parametreler genotipi oluştururken gerçekleştirilen tasarım fenotiptir. Kromozomun uygunluk değeri, fenotipin başarımına dayanır. Bu da uygunluk fonksiyonu kullanılmak suretiyle kromozomdan hesaplanabilir.

3.3.3 Uygunluk fonksiyonu

Çözümü istenen her problem için bir uygunluk fonksiyonu belirlenmelidir. Uygunluk fonksiyonu, belirli bir kromozomun çözüme yakınlığının göstergesi olan uygunluk değerinin hesaplanmasında kullanılır[1,11].

3.3.4 Olgunlaşmamış Yakınsama

İlk topluluk rasgele değerlerle yaratıldığından bireylerin uygunluk değerleri ve belirli bir lokusa ait genler arasında ciddi 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)[1,11].

Holland'ın şema teorisi, bireylere, uygunluk değeriyle 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.

3.3.5 Yavaş Sonlanma

Olgunlaşmamış yakınsamaya karşıt sorun yavaş sonlanmadır. Epey nesil geçmesine rağmen topluluk yaklaştığı halde global minimumu konumlandıramayabilir. Ortalama uygunluk değeri yeterince yüksek değerli olup en iyi bireyin uygunluk değerine yakınsamış olabilir. Olgunlaşmamış yakınsamayı önlemede kullanılan

yöntemler bu soruna karşı da kullanılır[1,6,11]. Kullanılan yöntemlerle topluluğun etkin uygunluk değerinin varyansı arttırılır.

3.3.5.1 Doğrusal Ölçekleme

i. kromozomun uygunluk değeri f_i ile karşılık gelen hedef değeri o_i arasındaki doğrusal ilişki (3.1) denklemiyle tanımlanır. *a* ve *b* sabit değerleri tespit edilirken f_i ve o_i ortalamaları eşitlenmeye çalışılır[6,12].

$$f_i = ao_i + b \tag{3.1}$$

3.3.5.2 Üssel Ölçekleme

A.M. Gillies'in önerdiği yöntemde, probleme bağlı değişkenlik arz eden ve olgunlaşmamış yakınsamayı önleyecek tarzda belirlenmesi gereken bir *"k"* üssü ölçeklemede kullanılır. Gillies, tanımı (3.2) eşitliğiyle verilen yöntemi kullanırken *k=1.005* almıştır[6,12].

$$f_i = o_i^k \tag{3.2}$$

3.3.5.3 Sigma Kesmesi

S.Forrest'in önerdiği, doğrusal ölçeklemede karşılaşılması 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 (3.3) eşitliğiyle verilir[6,12].

$$f_i = o_i - (o_{ortalama} - c\sigma) \tag{3.3}$$

3.3.6 Seçim

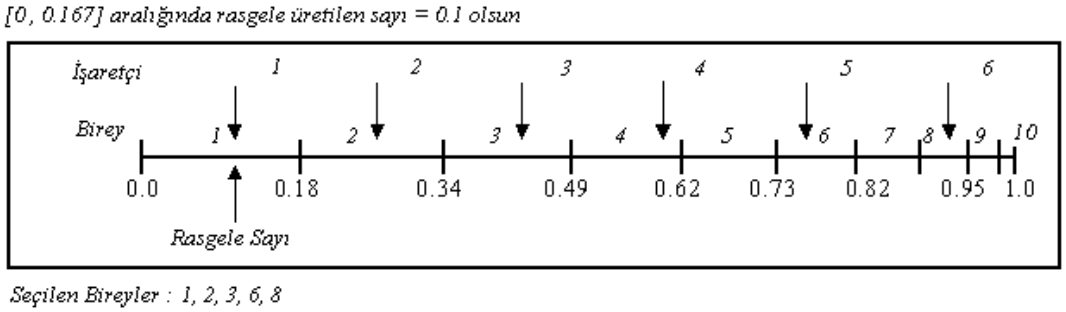
GA'da çözüme giden yol, bireylerin uygunluk değerinin artışını sağlayan gen içeriğinin edinilmesinden geçer. Bu sebeple, yeni nesili oluşturacak bireylerin seçimi hayati önemdedir. Önerilen çeşitli temel yaklaşımlar ve çeşitlemeleri mevcuttur. Sıklıkla kullanılan stratejiler, kesme seçimi, rulet tekerleği ve stokastik örneklemedir.

3.3.6.1 Kesme Seçimi

"En Güçlüler Yaşar" prensibinden hareketle uygunluk değerine göre büyükten küçüğe sıralanan bireylerden belirlenen sayıda en yüksek değerli bireyler seçilir, diğerleri yok edilir.

3.3.6.2 Stokastik Evrensel Örnekleme

James Baker(1987) tarafından önerilen yöntemde rulet stratejisine benzer yaklaşımla, bireyler doğru üzerine yerleştirilir. Seçilecek birey sayısına eşit sayıda işaretçinin, doğru üzerine eşit aralıklarla yerleştirilmesiyle örnekleme yapılır. Örneğin, seçilecek birey sayısı *Npointer=6* olduğunda, örnekleme periyodu *1/Npointer=0.167* olur ve ilk işaretçinin yeri [*0, 1/NPointer*] aralığında olmak koşuluyla rasgele seçilir[13].



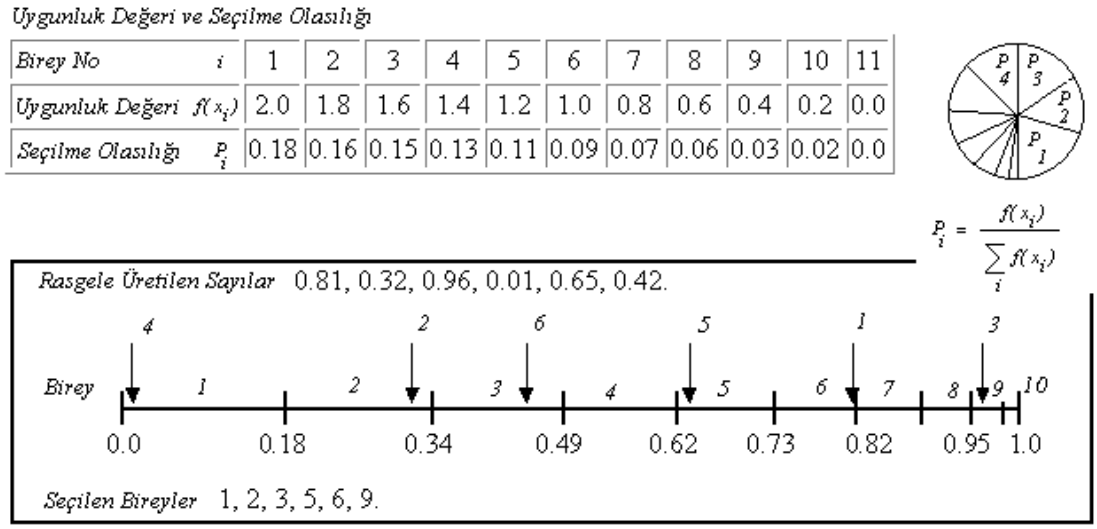
Şekil 3.2. Stokastik evrensel örnekleme

3.3.6.3 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 yerleştirilirler. Üretilen rasgele sayının rastladığı aralığın sahibi birey seçilir. Önceden belirlenen birey sayısına ulaşılana dek işlem tekrarlanır[13,14].

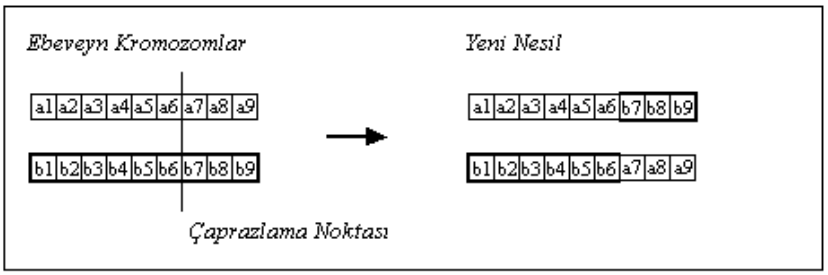
3.3.7 Çoğalma

Uygunluk değerini gözeten seçim stratejisi sonucu seçilen kromozomların çaprazlanmasıyla yeni nesili oluşturan bireylerin üretildiği evre, çoğalmadır. Ebeveyn olarak iki kromozom seçildikten sonra gerçekleşen çaprazlama işlemi tek veya çok noktalı olabilir. Rasgele belirlenen noktadan ikiye ayrılan kromozomlardan

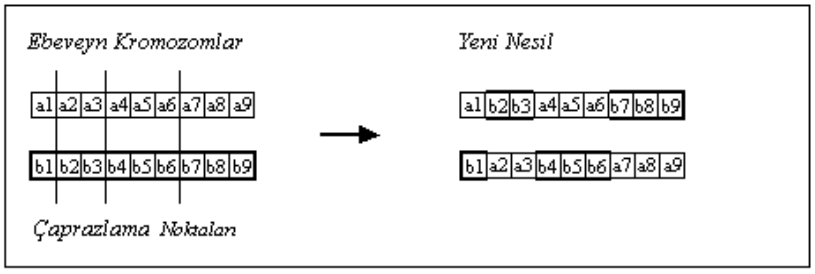


Şekil 3.3 Rulet tekerleği seçimi

baş ve kuyruk dizileri elde edilir. Şekil 3.4 ve 3.5'te görüldüğü gibi, baş veya kuyruk dizilerinin değiştokuşu sonrasında birleştirilen diziler yeni nesilin iki ferdi olarak kromozom havuzuna kaydedilirler.



Şekil 3.4 Tek noktalı çaprazlama



Şekil 3.5 Çok noktalı çaprazlama

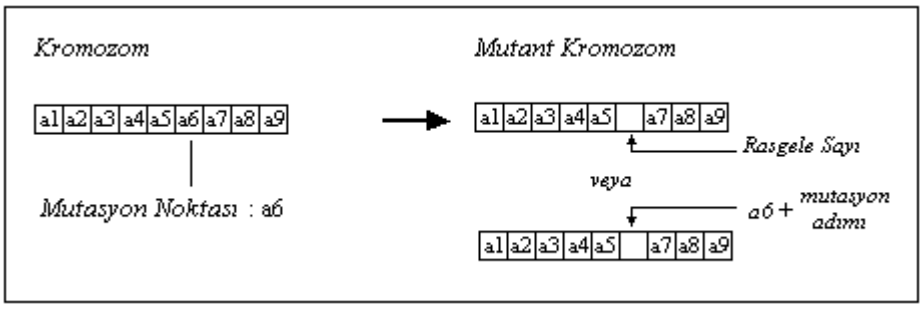
3.3.8 Mutasyon

Faydası tartışılmaya devam etmekle birlikte sıkça kullanılan bir diğer genetik operatör mutasyondur. Rasgele seçilen kromozomdaki bir veya birkaç geni rasgele

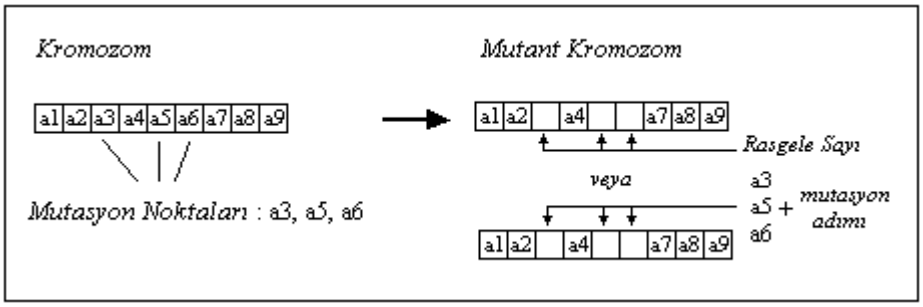
değişikliğe uğratan mutasyon operatörünün nesil başına uygulanma oranının düşük olması tavsiye edilir. Şekil 3.6'da tek noktada ve 3.7'de çok noktada mutasyon işlemi gösterilmiştir. İki tabanlı gösterimde, genin alabileceği değer { 0 , 1 } ile kısıtlı olduğundan 0→1'e, 1→0'a dönüşür. Gerçel sayılı gösterimde ise, gendeki değişim, rasgele belirlenen bir sayıyla yerdeğişimine bağlı olabileceği gibi mevcut değere mutasyon adımı olarak anılan küçük ilavelerle de gerçekleşebilir.

3.3.9 Yakınsama

GA'nın doğru gerçekleştirildiği uygulamalarda, kromozom topluluğundaki en iyi ve ortalama uygunluk değerleri, bireylerin, evrim sonucu gelişimiyle biribirine ve global optimuma yakınsar. Uygunluk kriterini sağlayan birey yakınsamıştır denir. Topluluk ortalamasının, en iyi bireyin uygunluk değerine yakınsadığı durumda topluluk yakınsamış olur. Topluluk ve kromozomun yakınsamasından başka, genin



Şekil 3.6 Tek noktada mutasyon



Şekil 3.7 Çok noktada mutasyon

yakınsaması da tanımlanmıştır. Bir nesildeki kromozomların belirli bir lokusu %95 oranında aynı gene sahipse gen yakınsamıştır denir[1,15].

3.3.10 Tersten Sıralama ve Yeniden Düzenleme

Genlerin sıralanışı çok önemlidir. Sıralamayı değiştirerek arama uzayını genişleten bir operatör tersten sıralamadır. Bu operatör, bir kromozom üzerindeki genlerden rasgele belirlenmiş iki lokus arasında kalanları ters sırayla yerleştirir.

3.3.11 Çift Değerlilik ve Baskınlık

İleri hayat formlarında, kromozomlar ikili sarmal düzenindedir, genler iki şerit üzerine kodlanmıştır. Biribirinin alternatifi iki genin kodlandığı yapı, çift değerli(diploid) kromozom adıyla anılır. Bugüne kadar olan GA çalışmaları tek şerit üzerine kodlanmış genlerle gerçekleştirilmiştir. Tek şeritli yapı haploid kromozom adıyla anılır. Çift değerliliğin sağlayabileceği faydalar olmasına karşı programlama ve işlem kolaylığı sağlamasından dolayı haploid yapı tercih edilmiştir. Zamana bağlı değişimin sözkonusu 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 değiştirebilecektir. Çift değerlilik, gende evrim sürecinden daha hızlı değişim sağlar.

3.3.12 Epistasis

Genlerarası etkileşim epistasis adıyla anılır. Bir genin uygunluk değerine katkısı, diğer genlerin sahip olduğu değerlere bağlıdır. Epistasis çok fazla ise GA verimli olmayacaktır. Çok düşük olduğunda ise diğer yöntemlerin başarımı GA'ya göre yüksek olacaktır.

3.3.13 Aldanma

Evrim süreci işledikçe, global optimumu sağlayacak olan şemaların veya yapıtaşlarının toplulukta görülme sıklığı artacaktır. Bu optimal şemalar, çaprazlama operatörüyle, nesiller geçtikçe biraraya 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.

3.4 Genetik Algoritmalar Nasıl Çalışır?

Yaratıcılarınca dahi tam olarak anlaşılamadığı halde, doğal seçim sürecine benzeşimle evrim geçirerek problem çözen bilgisayar programları[16] olan GA'nın iyi çalışmasını garantileyebilmek amacıyla deneye dayalı kuralların bulunmasına dönük araştırmaların sonucu ulaşılmış ve kabul görmüş bir genel teori henüz yoktur[1]. Yine de başarılı uygulamalar geliştirilmesinde yardımcı olan ve GA'nın başarısını kısmen izah edebilen iki ekol vardır. Sırasıyla izah edilmiş olan bu iki yaklaşım Şema Teoremi ve Yapıtaşı Hipotezi'dir[1]. Şema teorisinin bir özelliği de GA'nın sahip olduğu aleni ve üstü örtülü paralellik(koşutluk) özelliklerinden ikincisini açıklamasıdır. Aleni paralellik, çözüm sağlayacağı umulan birden fazla parametre kombinasyonunun işletilmesinden doğar[17].

3.4.1 Şema Teoremi

GA'nın çalışmasını izaha yönelik ilk özenli çalışma J. Holland'ın Şema Teoremi'dir. İki tabanlı gösterimde {0,1,#} değerlerini alabilen genlerden oluşan belirli bir örüntüye şema denir. Bir kromozomun genleri, # değeri herhangi bir değere karşılık gelmek üzere, {0,1} değerlerine sahip bir şablonla birebir uyuşursa, o şemaya sahip kabul edilir. Örneğin, '1010' dizilişli kromozomun sahip olduğu bazı şemalar: '10##', '#0#0', '##1#', '101#'dir. Şemanın derecesi, #'den farklı olan değerlerin sayısına eşittir. Yukarıdaki şemalar için dereceler, sırasıyla, (2, 2, 1, 3)'tür. Şemanın tanımlayıcı uzunluğu(defining length), en dıştaki #'den farklı değerlerin kapalı aralığında kalan gen sayısıdır ve örnek şemalar için değerler, yine sırasıyla olmak üzere, (2, 3, 1, 3)'tür.

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 düşünülür.

Holland, arama uzayını keşfetmenin en uygun yolunun, kromozomların sahip oldukları uygunluk değeriyle orantılı çoğalma denemesi fırsatı kazanmaları olduğunu göstermiştir. Bu yolla, iyi şablonlar ardışık nesiller boyunca üssel artan deneme şansı yakalarlar. Ayrıca, yine Holland göstermiştir ki bir kromozom pek çok şablona sahip olduğu durumda, 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[1].

3.4.2 Yapıtaşı Hipotezi

Goldberg'e göre, GA'nın gücü, iyi yapıtaşları bulabilmesinden kaynaklanmaktadır. Yapıtaşları, kısa tanımlayıcı uzunluklu, birleştirildikleri takdirde başarımı artırma eğiliminde olan uyumlu dizilişlerdir. Başarılı kodlama, ilişkili genlerin kromozom üzerindeki konumlarının yakın olduğu halde genlerarası etkileşimin az olmasının sağlandığı durumda yapıtaşlarının oluşumunun teşvik edildiği kodlamadır. Tercih edilmeyen bir durum olmasına karşın epistasis yani, genler arasında etkileşim olur ve yukarıda belirtilen başarılı kodlama koşulu kolaylıkla sağlanamaz.

Bir genin toplam uygunluk değerine katkısı diğer genlerden bağımsız olsaydı, problemin çözümü, sırasıyla her gen için tepe-tırmanma yönteminin uygulanmasıyla bulunabilirdi ki genelde mümkün olmayan bir çözüm yoludur.

Bu başlık altında incelenmekte olan iki önemli ve ilginç soru:

1) Genele yönelik başarılı kodlamayı sağlayabilecek kodlama tasarımı gerçekleştirilebilir mi? Evet ise nasıl?
2) Eğer mümkün değilse, bu durumda GA başarımını arttırabilecek değişiklikler mümkün müdür? Evet ise nedir?[1]

3.5 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.

Sade rasgele arama, keşif konusunda iyi olduğu halde keşfin kullanımı söz konusu değildir. Tepe-tırmanma yöntemi ise az keşif yapmasına karşı keşfin kullanımı konusunda başarılıdır. Bu iki yöntemin birleştirilerek kullanımı gayet verimli olabilir. Ancak, daha fazla keşif yapmaya karar vermeden önce mevcut keşfin kullanımına ne kadar süreyle devam edileceği konusunda dengenin bulunması kolay olmayabilir.

Holland göstermiştir ki GA, keşif ve keşif 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) Toplululuk nüfusu sonsuzdur.
2) Uygunluk fonksiyonu, çözümün işe yararlığı için doğru göstergedir.
3) Kromozomdaki 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[1].

3.6 Bilgi-tabanlı Teknikler

GA çalışmalarında gelenekselleşmiş operatörlerle beraber arama uzayına ait bilgi kullanan göreve özel tasarlanmış yeni operatörler de kullanılabilir. Göreve özel GA'nın başarımı yükselir[18].

Probleme özel bilgi çaprazlama işleminde elverişli bir şekilde katıştırılabilir[18]. Arama uzayı bilgisi, uygunluk değeri belirgin derecede düşük, kısıtları ihlal eden kromozomların çoğalmasının engellenmesinde kullanılabilir. Böylece, başarımı düşük bireylerin işlemleriyle vakit kaybı önlenmiş olur[18].

Arama uzayı bilgisi, iyi noktalar etrafında keşfi yoğunlaştıracak şekilde yerel iyileştirme operatörlerinin tasarımı için kullanılabileceği gibi daha uygun noktalar civarında arama yapmak üzere kromozom topluluğunun ilk değer atamasında da kullanılabilir[18].

3.7 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, rasgele verilerin modellenmesi için çok uygundur[11].

Öğrenme yeteneğine sahip sistemlere yönelik uygulamaları da olan GA'nın, ekonomik modelleme ve piyasa işlemleri[11] gibi belirli bir durumu analiz ederken kural tabanlı gelişim göstermesi sağlanabilir.

3.8 Karşılaşılabilecek Genetik Algoritma Türleri

Literatürde çeşitli ön isimlerle anılan genetik algoritmalar vardır. Bunlar hakkında detaya girilmeksizin bilgi verilmiştir.

3.8.1 Basit Genetik Algoritma

İşleyişi şekil 3.1'e uygun olan ve bir neslin çoğalarak yeni nesli oluşturduğu, örtüşmeyen toplulukların varlığına dayalı yaklaşımdır[6].

3.8.2 Kararlı Durum Genetik Algoritması

Örtüşen topluluklar kullanılır ve nesildeki yok edilecek birey sayısının kullanıcı tarafından belirlenmesine olanak sağlanır[6].

3.8.3 Deme Genetik Algoritması

Kararlı durum genetik algoritmasını kullanarak birkaç topluluğun paralel evrimiyle sonuç arar. Yeni nesilin oluşturulması aşamasında bazı bireyler topluluklar arasında göç ettirilir[6].

3.8.4 Düzensiz Genetik Algoritmalar

Goldberg ve arkadaşları tarafından geliştirilen düzensiz genetik algoritmaların yaklaşımı, uygunluk değeri yeterince yüksek yapı taşlarının bulunup birleştirilerek daha yüksek uygunluk değerli bireylerin oluşturulmasıdır[6].

3.8.5 Üretici Genetik Algoritma

BGA, GA ile ES arasında bir konumdadır. BGA'da kullanılan seçim stratejisi kesme seçimidir[25]. Bir nesildeki, miktarı sayı veya yüzde oran olarak atanmış en iyi uygunluk değerli bireylerin yaşamasına izin verilir ve diğerleri yok edilir.

GA teorisi ve yaklaşım arasındaki bağlantının yanısıra seçim, yeniden düzenlenme ve mutasyon operatörlerinin incelemeleri de [25]'te bulunabilir.

4. GENETİK ALGORİTMALAR'DA ÖĞRENME

Evrim ve öğrenme, biyolojik uyarlanmanın uzay-zamanda farklılıkla gerçekleşen iki türüdür. Evrim, tercihe bağlı çoğalma sonucu doğan bireylerden uygunluğu arttıracak farklılaşmalar gösterenlerin, ebeveynlerin yerini aldığı süreçtir. Öğrenme ise, bireyin var olduğu sürece uygunluğunu arttıracak değişimlere uğramasıdır. Görüldüğü üzere, süreçlerin zaman ölçekleri farklıdır. Diğer bir farklılıkta, evrim genotipi değiştirirken öğrenme fenotipi etkiler. Fenotipteki değişim, genotipi doğrudan etkilemez. Öğrenme ve evrimin etkileşimiyle ilgili yürütülen belli başlı çalışmalar hakkında detaylı bilgi için Stefano Nolfi ve Dario Floreano'nun çalışmalarına bakılabilir[19].

Öğrenme algoritmalarına dahil olmayan Genetik Algoritmalar bir kısım öğrenme işlemleri için güçlü arama yöntemleri önerir. Bu bağlamda, GA'nın makine öğrenmesi problemlerinde kullanımına dönük çalışmalar mevcuttur[3]. Michigan, Pittsburgh ve Yinelemeli Kural Öğrenme(IRL) önerilen üç alternatif yaklaşımdır[3].

4.1 Michigan yaklaşımı

Kromozomlar bireysel kurallardır ve topluluk, kurallar kümesidir. Kurallar, ortamla etkileşimli olarak zamanla değişime uğrar. Michigan yaklaşımı tabanlı genetik öğrenme işlemi "Sınıflandırıcı Sistem" adıyla da anılır.

4.2 Pittsburgh yaklaşımı

Her bir kromozom tam bir kurallar kümesini kodlar. Çaprazlama ile yeni bir kurallar dizilişi ve mutasyon sonucu yeni kurallar elde edilir. Bazı durumlarda, değişken uzunluklu kural tabanları kullanılmaktadır. Smith tarafından 1980'de önerilen bu yöntem ve yönteme güncel bakış açıları [3]'de bulunabilir.

4.3 Yinelemeli Kural Öğrenme

Michigan yaklaşımında olduğu gibi her bir kromozom bir kuralı temsil eder. Fakat Michigan yaklaşımının aksine, yalnızca en iyi bireyin çözümün bir parçası olduğundan hareketle diğer bireyler gözardı edilir. Problemin çözümünü sağlayacak kurallar kümesini elde etmek için GA, aşağıda önerilene benzer bir yinelemeli düzende yer almalıdır.

1. Sistem için bir kural elde etmekte 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[3].

4.4 Evrimci Yapay Sinir Ağları

Yapay Zeka genel başlığı altında öğrenme yeteneğiyle bilinen en popüler model Yapay Sinir Ağları'dır. Ek B'de Yapay Sinir Ağları hakkında bilgi verilmiştir.

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[20,21,22]. Genel çerçevesi şekil 4.1'de verilen etkileşimli üç evrim süreciyle başarımı en iyileştirilmiş YSA tasarımı gerçekleştirilebilir. Bölüm 5'te verilen örnek uygulamada, YSA bağlantı değerlerinin evrimci algoritma kullanımıyla belirlenmesi konu edilmiştir.

4.4.1 EYSA Bağlantı Ağırlıklarının Evrimi

YSA'da öğrenme, denetimli veya denetimsiz gerçekleşebilir. Önceden belirlenmiş uygunluk kriteri gerçeklenmek üzere ağırlıkların ayarlanma süreci olan denetimli öğrenmede en sık kullanılan eğitim algoritması geriyayınımdır(backpropagation). Gradyan azaltarak arama yapan yöntem, YSA çıktısı ile hedef değer arasındaki

farktan doğan hatanın karesinin ortalaması veya bu ortalamanın karekökünün minimizasyonunu hedefler. Daha önce de belirtildiği üzere, çok tepeli fonksiyonlarda yerel minimum problemiyle başedemez ve türevlenemeyen fonksiyonlarla işlem yapamaz. Başarılı uygulamaları görmek ve daha detaylı bilgi edinmek için [20,21,22]'ye ve referanslarına bakılabilir.

Yerel minimum probleminin üstesinden gelmekte, uygunluk değerlendirmesi için YSA'nın hata kriteri seçilerek, evrim sürecine işlerlik kazandırılabilir. Sürecin belli başlı adımları aşağıda verilmiştir.

1) Kromozomların, ağırlık değeri olarak atanması
2) Uygunluk değerinin hesaplanması
3) Uygunluk değerini gözeterek çoğalma sürecinin başlatılması
4) Genetik operatörleri kullanarak yeni neslin elde edilmesi

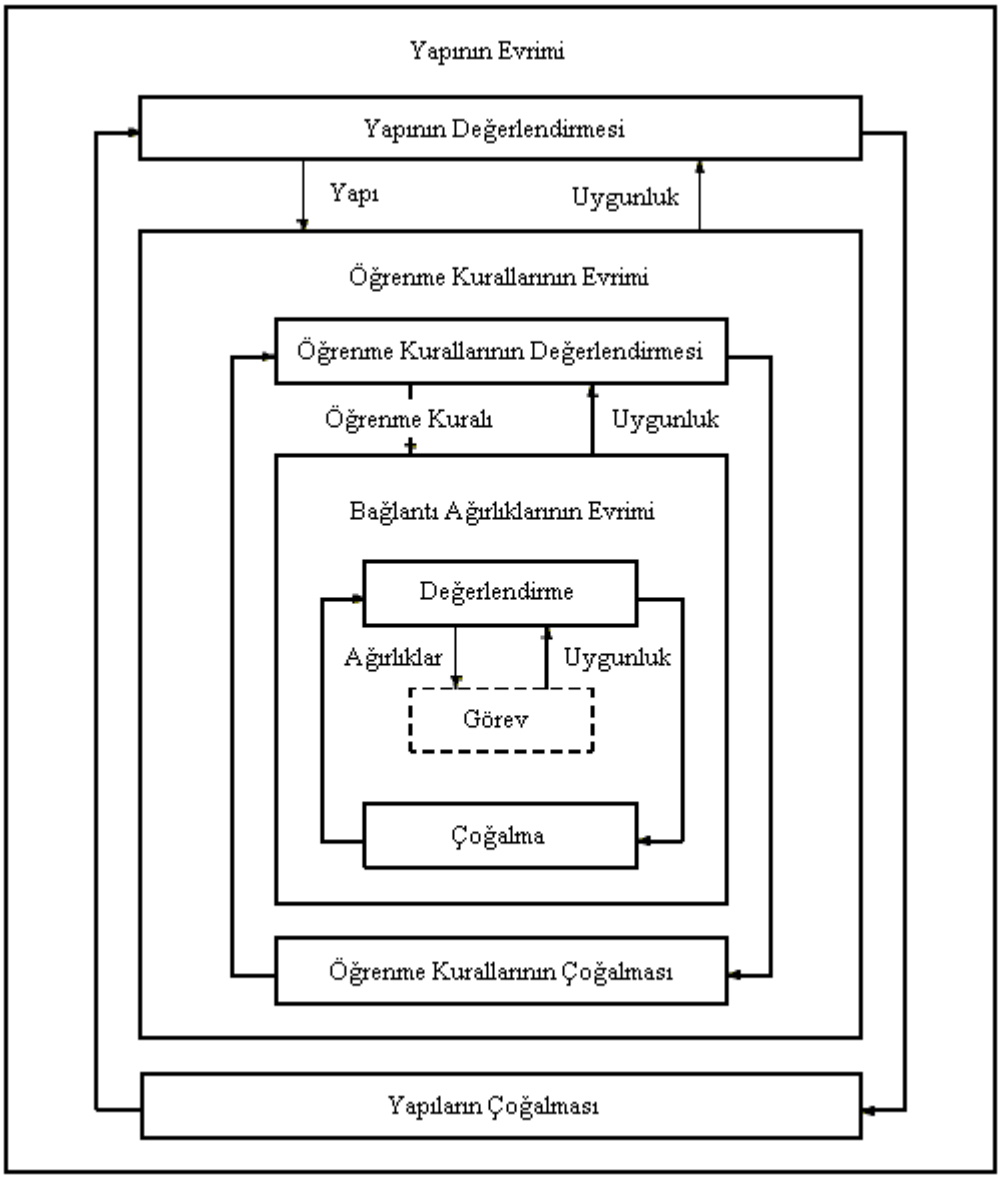
Kromozomları oluşturan ağırlık değerlerinin gösterimi ve benimsenen evrim süreci, evrimci eğitim sürecinin iki ana başlığını teşkil eder. Ağırlık değerlerinin temsilinde gerçel sayılı veya iki tabanlı gösterim kullanılabilir.

4.4.1.1 İki Tabanlı Gösterim

Her ağırlık değeri, sabit uzunluklu bit dizisi ile temsil edilir. Kromozomlar ise bit dizilerinin biribirine eklenmesiyle elde edilir. İki tabanlı gösterim kullanıldığı durumlarda doğrudan iki tabanında kodlamanın yanısıra Gray kodu, üssel kodlama veya daha karmaşık bir kodlama kullanılabilir. Gösterimin çözünürlüğü, belirlenmeye çalışılan ağırlık değerinin hassasiyetini belirlediğinden hayati önem taşır. Çok kısa gösterim yakınsamayı imkansızlaştırabilecekken uzun gösterim işlem yükünü ve süresini arttırır. Kullanılacak bit sayısının optimizasyonu, kodlamada kullanılacak değer aralığı, kodlama yöntemi açık problemdir.

4.4.1.2 Gerçel Sayı Kodlamalı Gösterim

İki tabanlı gösterimin noksanlarını gidermek üzere her bir ağırlığa karşılık bir gerçel sayı ataması önerilen yöntemde kromozomlar, değerlerin dizi oluşturacak tarzda sıralanmasıyla oluşturulur [10,21,22]. Çaprazlama işlemi, diziyi herhangi bir noktadan bölerek gerçekleşebileceği gibi genleri bölerek basamak düzeyinde de gerçekleşebilir. Ebeveynlerin ortalamasını alan çaprazlama, rasgele mutasyon ve arama uzayına özel tanımlanabilecek işlemler, genlerin uygunluk değerini arttıracak



Şekil 4.1 Evrimci Yapay Sinir Ağı genel yapısı

gelişimi sağlayan diğer genetik operatörlerdir. Montana ve Davis, çalışmalarında gelişime açık evrimci yaklaşımın eğitim süresinin, ilgilendikleri problem için, geriyayınım algoritmasına göre daha kısa olduğu sonucuna ulaşmışlardır[21]. Benzer sonuca ulaşan Bartlett ve Downs EYSA'nın boyutlarının artımıyla yakınsama süreleri arasındaki farkın evrimci algoritmalar lehine açıldığı sonucuna ulaşmışlardır. Bu sonucun genel olarak doğru olduğunun teyit edilmesi için daha fazla çalışma yapılması gerekmektedir[21].

4.4.1.3 Evrimci Eğitim ve Geriyayınımlı Eğitim Karşılaştırması

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 sebebiyle caziptir. Hata fonksiyonun türevine ihtiyaç duymaması nedeniyle türevlenemeyen hata fonksiyonlarıyla karşılaşıldığında sıkıntı yaşamaz. Ayrıca, eğitilmek istenen YSA'nın türüyle ilgili kısıtlama getirmez. Algoritmik açıdan, EA'nın ve özel olarak GA'nın global örnekleme yeteneği ince ayar yapabilme yeteneğinin önüne çıkar. Üç katmanlı ve ileri beslemeli YSA'lar için belirtilen avantajlarına karşı, evrimci algoritmaların hesaplama maliyeti yüksektir[22].

4.4.1.4 Hibrid Eğitim Yaklaşımı

Evrimci eğitimin etkinliği, GA'nın global örnekleme yeteneği ve yerel arama yaklaşımının ince ayar becerisiyle birleştirilerek arttırılabilir. Yüksek uygunluk değeri vaad eden bir bölgenin bulunması için GA kullanımını takiben bölgede arama yapmak üzere yerel arama yöntemleri işletilebilir.

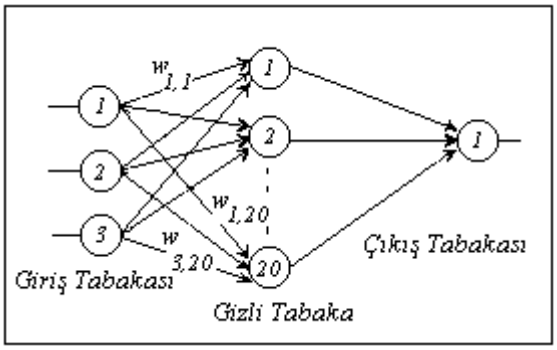
Belew, ağırlıkların ilk değer atamasını GA kullanarak yaptıktan sonra bulduğu yeterince iyi değerleri iyileştirmek için geri-yayınım algoritması kullanmıştır[21]. Sistemin başarımı, sade GA'ya ve sade geri-yayınıma göre nispeten artmıştır.

5. UYGULAMA

İleri-beslemeli yapay sinir ağının denetimli eğitimi, hatanın gradyanını gözeten türeve dayalı yöntemlerle gerçekleştirilmeye çalışılan klasik bir problemdir. Hedef fonksiyonun sürekli ikinci türeve sahip olduğu durumda geleneksel yöntemlerin bir çözüme yakınsama hızı daha yüksektir. Ancak, geleneksel yöntemlerde karşılaşılan en ciddi sorun yerel minimuma takılmadır[belanche].

MATLAB ortamında yazılımı gerçekleştirilen genetik operatörler, ileri beslemeli yapay sinir ağının bağlantı ağırlıklarının hedeflenen doğrulukta bulunmasında kullanılmıştır. Böylelikle yapay sinir ağının eğitimi, genetik algoritmaların kullanımıyla gerçekleştirilmiştir.

Eğitimi gerçekleştilen yapay sinir ağının topolojisi Şekil 5.1'de gösterilmiştir. Giriş katmanı 3, gizli katmanı 20 ve çıkış katmanı 1 nöron içeren bir yapı önerilmişse de. farklı nöron sayılarının atanması mümkündür. Gizli katmandaki nöronların aktivasyon fonksiyonu lojistik sigmoid ve çıkış katmanındaki nöronunki doğrusal aktivasyon fonksiyonudur.



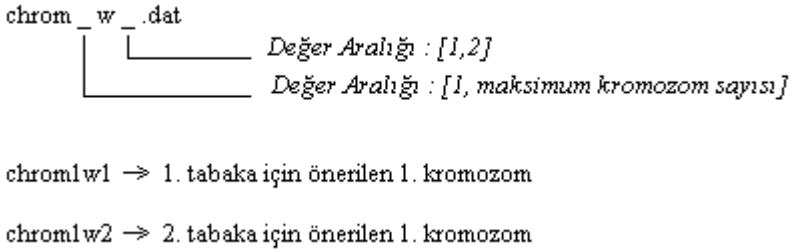
Şekil 5.1 Eğitimi gerçekleştirilen YSA topolojisi

Önerilen yapıyla, günlük ortalama debi verisinden ibaret zaman serisi modellemesi yapılmıştır. Veri, [0.1,0.9] aralığına doğrusal ölçeklenmiştir. Uygunluk değerlendirmesi için hedef değer ve YSA çıkışı arasındaki farkın karesinin ortalama değerinin(MSE) veya bu değerin karekökünün(RMS) tersi kullanılabilir. Azalan hata

teriminin uygunluk değerini arttırdığının bilincinde olarak uygunluk değerinin maksimizasyonu yerine RMS'in minimizasyonu hedeflenmiş ve değerlendirme bu ilkeyle yapılmıştır. (5.1) eşitliğiyle tanımlanan hata kriteri *RMS=0.035* olarak atanmıştır.

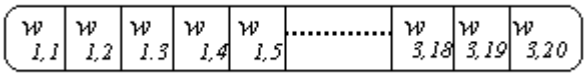
$$RMS = \left[\frac{1}{N} \sum_{i}^{N} \varepsilon_i^2 \right]^{\frac{1}{2}} \tag{5.1}$$

Muhtemel çözümlerin kodlandığı kromozomlar, giriş-gizli katman bağlantıları ve gizli-çıkış katman bağlantıları için eşli olarak yaratılıp şekil 6.2'deki gösterimle isimlendirilip kaydedilmiştir.



Şekil 6.2 Kromozomların isimlendirilmesi

Ağırlık değerleri MATLAB yazılımının RANDN fonksiyonu ile belirlenerek şekil 6.3'deki dizilişe göre yerleştirilmişlerdir. RANDN fonksiyonunun kullanımıyla ağırlıkların alabileceği değerler ve dolayısı ile arama uzayı sınırlandırılmış olmaktadır.



Şekil 6.3 Ağırlık değerlerinin kromozomu oluşturan diziliş tarzı

Kullanılan genetik operatörler,

1. Tek noktalı çaprazlama
2. Çok noktalı çaprazlama
3. Aritmetik ortalamalı çaprazlama
4. Tek noktada mutasyon
5. Çok noktada mutasyondur.

Bunlardan başka, RMS değeri hata kriterine ulaşmadığı fakat atanmış bir komşuluğuna yaklaştığı zaman adaptif öğrenme katsayılı geriyayınım algoritması da bir operatör gibi devreye girer ve sonuca ulaşmaya çalışır. Kullanılan geriyayınım yazılımı için [24] numaralı referansa başvurulabilir.

Herhangi bir operatörün işlemesi sonucu oluşan birey veya bireylerin değerlendirmesi yapılır. Değerlendirme sonucu, hatası ebeveynlerinden daha düşük olan kromozomlar topluluk nüfus sayacı arttırılarak topluluğa sıradaki numarayla kaydedilir. Topluluk nüfusunun sonsuz olması imkansız olduğundan bir kısıtlama getirilmesi kaçınılmazdır. Bu bağlamda getirilen kısıtlamayla, yazılım koşturulurken kontrol noktasına gelindiği anda topluluk nüfusu 100'ü geçmiş ise uygunluk değeri en yüksek 20 birey seçilir ve diğerleri yok edilerek işleme devam edilir. Sürecin sonlandırılması için sağlanması gereken iki koşuldan ilkinin tanımlanmış hata veya uygunluk değerine ulaşılması olduğu daha önce belirtilmişti. İkinci koşul da maksimum nesil sayısıdır. Algoritma, pratik anlam taşıyabilmesi için sonlu bir sürede sonuca yakınsayabilmelidir. Yazılımda atanan maksimum nesil sayısı 50'dir.

Uygulama, literatürde Üretici Genetik Algoritma (Breeder Genetic Algorithm, BGA) adıyla anılan yaklaşımla benzeşmektedir. BGA hakkında detaylı bilgi [22]'de bulunabilir.

Modelleme ve simulasyon sonuçlarına ilişkin grafikler Ek A'da verilmiştir.

6. SONUÇLAR VE TARTIŞMA

Bu çalışmada optimizasyona yeni bir yaklaşım getirmiş olan Genetik Algoritmalar ve optimizasyonda kullanımları konu edilerek temelleri hakkında bilgi verilmiş ve optimizasyon problemlerinde kullanım hedefiyle yazılım geliştirilmiştir. Geliştirilen yazılım, uygunluk fonksiyonu tanımlanan herhangi bir 1-boyutlu optimizasyon probleminde doğrudan kullanılabileceği gibi gerekli ilavelerle çok boyutlu optimizasyon problemlerinde de kullanılabilir. Sunulan örnek uygulama, gerçel sayı kodlamalı genetik algoritmaların optimizasyonda kullanımının yanısıra bir diğer biyolojik kökenli yaklaşım olan ve öğrenme yeteneğiyle bilinen yapay sinir ağlarının evrimci modellenmesi hakkında da bilgi vermektedir. Ayrıca, kullanılan verinin gerçek olması ve modelleme sonucunun başarılı olması sebebiyle de anlam taşımaktadır.

İleriye dönük yapılabilecek çalışmalar, ilki GA yazılımıyla, ikincisi uygulamaya konu olan EYSA ve üçüncüsü zaman serisi modellemeye ilgili üç ayrı başlık altında toplanabilir.

1. GA yazılımıyla ilgili yapılabilecekler:
    - Seçenek sağlamak üzere operatörlerin çoğaltılması,
    - Yazılımın modüler hale getirilmesi ve kullanıcı dostu tasarıma kavuşturulması,
    - Nihai olarak MATLAB ile uyumlu bir araçkutusu haline getirilmesidir.
2. Örnek uygulamanın şekil 4.1'de verilen EYSA çerçevesini tamamlamak üzere geliştirilmesidir.
3. Örnek uygulamanın ve çalışmanın esin kaynağı olan eldeki rüzgar hızı ve yönü bilgisinden ibaret zaman serisinin modellenmesi ve verideki boşlukların geliştirilecek model ile doldurulmasıdır.

KAYNAKLAR

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

[2] **Bodenhofer, U., Herrera, F.**, 1997. Revised version of lecture notes from "Preprints of the International Summer School: Advanced Control-Fuzzy, Neural, Genetic", 1-69

[3] **Herrera, F., Magdalena, L.**, A Genetic Learning Process For The Scaling Factors, Granularity and Contexts of The Fuzzy Rule-Based System Data Base, www.researchindex.com, Erişim tarihi: Mart 2002

[4] **Haupt, R.L., Haupt, S.E.**, 1998. Practical Genetic Algorithms, John Wiley&Sons Inc.

[5] **Computational Science Education Project**, 1995. Mathematical Optimization, www.researchindex.com, Erişim tarihi: Mart 2002

[6] **Demirel, L.**, 1999. Genetic Algorithms in Engineering Optimization, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[7] **Tsoukalas,L., Uhrig, R.E.**, 1997. Fuzzy and Neural Approaches in Engineering, John Wiley&Sons Inc., New York

[8] **Koza, J.**, introduction-to-genetic-algorithms.pdf, www.researchindex.com, Erişim tarihi: Şubat 2002

[9] **Michalewicz, Z.**, 1994. Genetic Algorithms + Data Structure = Evolution Programs, AI Series, Springer-Verlag, New York.

[10] **Goldberg, D.E.**, Real Coded Genetic Algorithms, Virtual Alphabets and, Blocking, www.researchindex.com, Erişim tarihi: Mart 2002

[11] **Busetti, F.**, Genetic Algorithms Overview, www.researchindex.com, Erişim tarihi: Mart 2002

[12] **Man, K.F., Tang, K.S., Kwong,S., Halang, W.A.**, 1997. Genetic Algorithms for Control and Signal Processing, Springer-Verlag London Ltd.

[13] **Pohlheim, H.**, 1997. GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB, www.geatbx.com/docu/index.html, Erişim tarihi: Mart 2002

[14] **van Rooij, A.J.F., Jain, L.C., Johnson, R.P.**, 1996. Neural Network Training Using Genetic Algorithms, World Scientific Publishing Co. Pte Ltd.

[15] **DeJong, K.**, 1980. Adaptive System Design: A Genetic Approach. IEE Trans SMC, 10, 566-574.

[16] **Holland, J.**, 1992. Genetic Algorithms, Scientific American, 44-50

[17] **Houck, R.C.**, Joines, J.A., Kay, M.G., A Genetic Algorithm for Function Optimization: A Matlab Implementation

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

[19] **Nolfi, S., Floreano, D.**, Learning and Evolution, www.researchindex.com, Erişim tarihi: Mart 2002

[20] **Yao, X.**, 1993. A Review of Evolutionary Artificial Neural Networks, International Journal of Intelligent Systems, 8, 539-567

[21] **Yao, X.**, September 1999. Evolving Artificial Neural Networks, Proceedings of the IEEE, 87(9): 1423-1447

[22] **Zhang, B.T., Mühlenbein, H.**, 1993. Evolving Optimal Neural Networks Using Genetic Algorithms with Occam's Razor, Complex Systems, 7(3): 199-220

[23] **Werbos, P.J.**, 1994. The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, John Wiley&Sons Inc.

[24] **Hines, W.**, 1997. Matlab Supplement to Fuzzy and Neural Approaches in Engineering, John Wiley&Sons Inc.

[25] **Belanche, Ll.,** 1999. A Case Study in Neural Network Training with Breeder Genetic Algorithm, LSI Technical Report LSI-00-07-R, Universitat Politecnica de Catalunya.

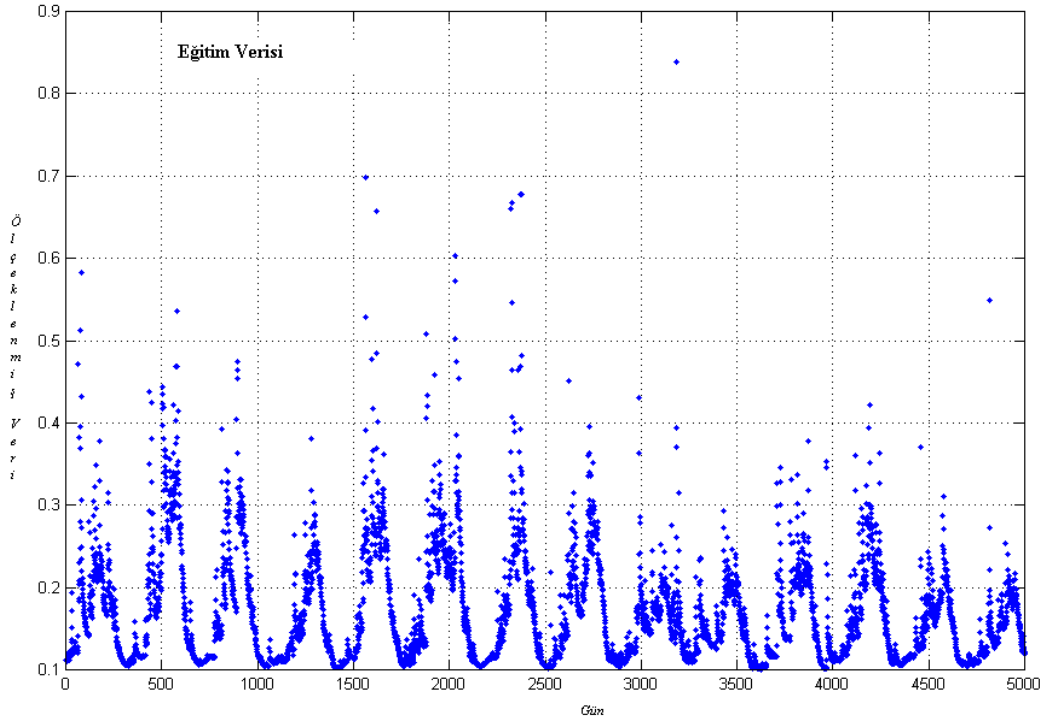
EKLER

EK A SİMULASYON SONUÇLARI 35

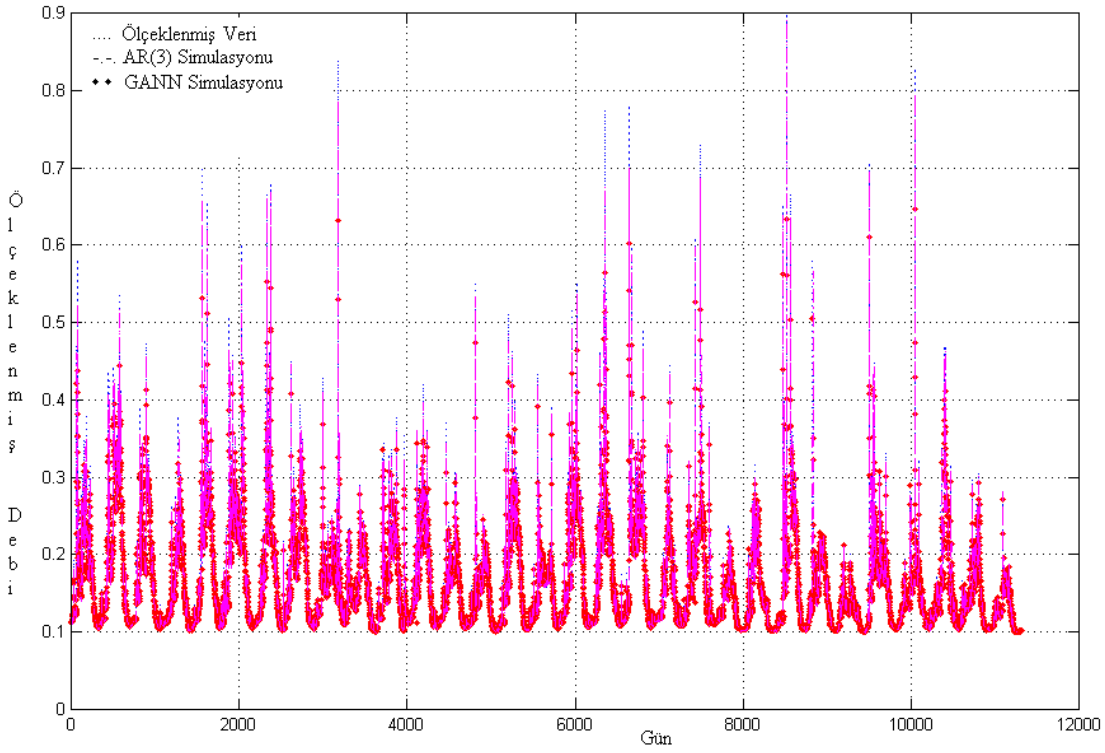
EK B YAPAY SİNİR AĞLARI 41

EK A

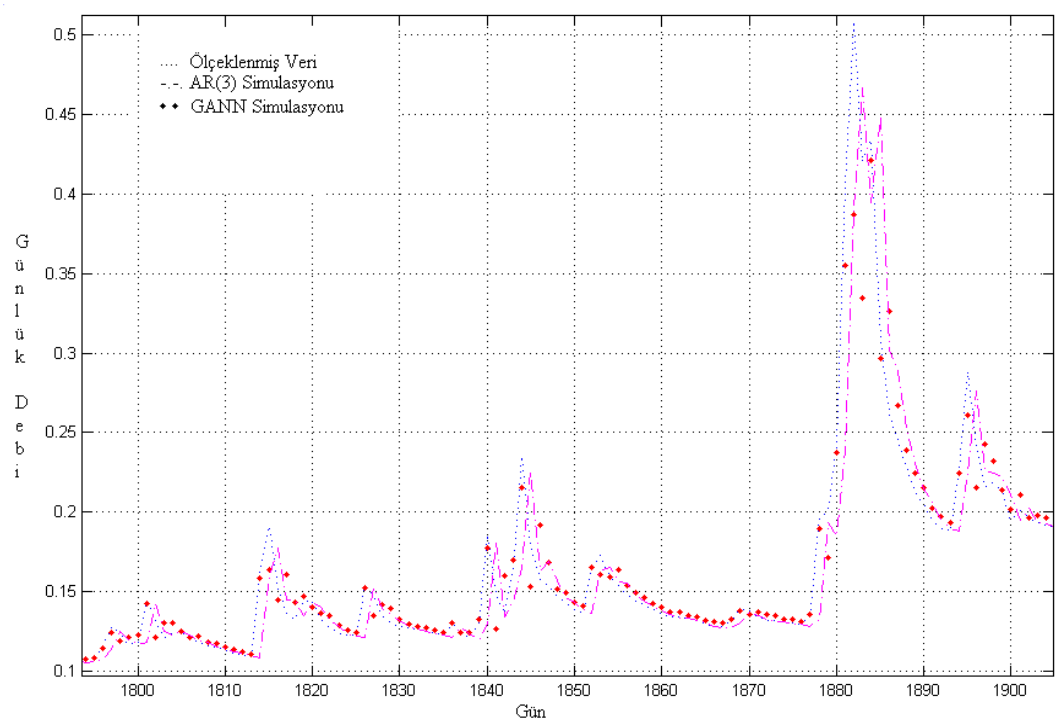
SİMULASYON SONUÇLARI



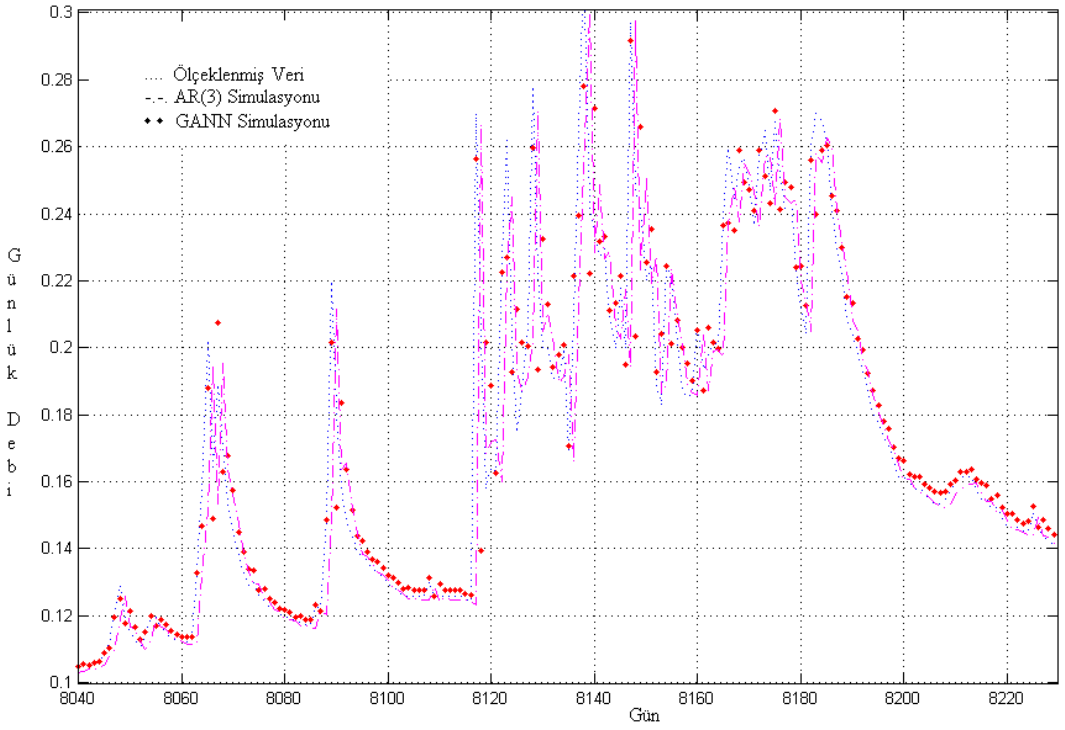
Şekil A.1 EYSA'nın eğitiminde kullanılan veriler



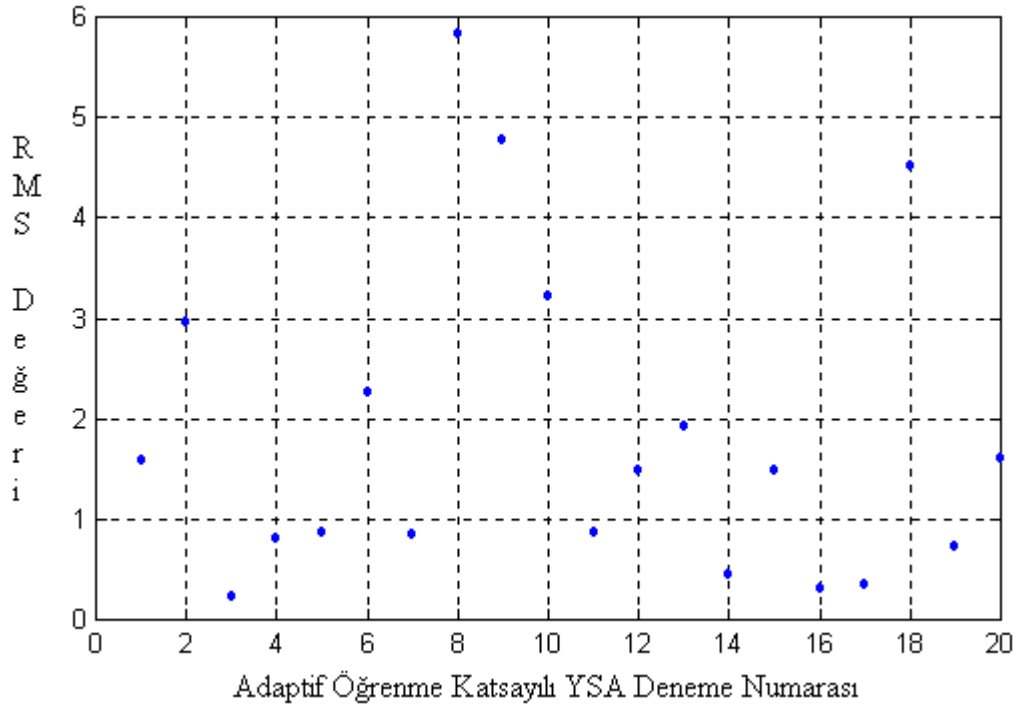
Şekil A.2 Zaman serisinin tum değerlerine karşılık EYSA yazılımı GANN ve AR(3) simulasyonu



Şekil A.3 Modellemede kullanılan gerçek değerlere karşılık EYSA yazılımı GANN ve AR(3) simulasyonu



Şekil A.4 Modelin sınanmasında kullanılan gerçek değerlere karşılık EYSA yazılımı GANN ve AR(3) simulasyonu



Şekil A.5 Adaptif öğrenme katsayılı YSA'nın çeşitli denemeler sonucu ulaşılabildiği RMS değerleri

EK B

YAPAY SİNİR AĞLARI

B.1 YAPAY SİNİR AĞLARI

"Yapay Zeka" terim olarak, 1956'da Darthmouth Koleji'nde Rockefeller Vakfı'nın himayesinde düzenlenen, "Öğrenmenin ve Zekanın Diğer Belirleyici Niteliklerinin Her Yönünde Bilgisayarların ve Simulasyonun Potansiyel Kullanımı" konulu konferansta ortak kullanım kazanmıştır. Yapay zeka, genel anlamda, zeka kullanımını gerektiren faaliyetlerle ilişkili insani düşünme sürecini taklit etme girişiminde olan bilgisayar işlemleri olarak tanımlanabilir. Kendiliğinden öğrenme, doğal dil anlama, görüntü-ses hatırlama, oyun oynama, matematiksel problem çözme, robotik, uzman sistemler, yapay sinir ağları, Genetik Algoritmalar gibi konular yapay zekanın altbaşlıkları olarak incelenir[7].

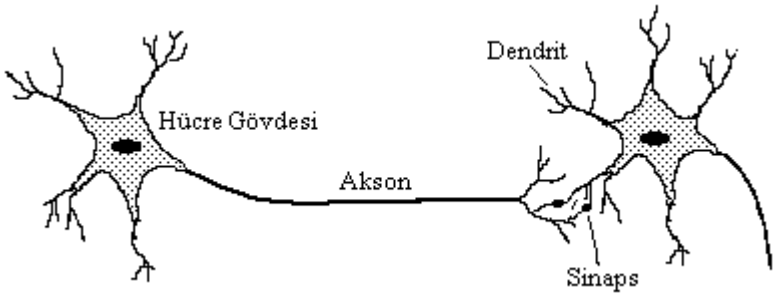
Bu çalışmada kullanılan YSA ve GA, kökeni canlıların biyolojik veya davranış görüngülerine dayanan yaklaşımlardır. Ayrı ayrı kullanılabildikleri gibi sinerji sağlamak üzere melez kullanımları da mümkündür.

Bu bölümde, öncelikle yapay sinir ağlarının biyolojik kökeninden bahsedilecek ve daha sonra yapay sinir ağları hakkında genel bilgi verilecektir.

B.1.1 Yapay Sinir Ağlarının Biyolojik Kökeni

İnsan beyni, düşünme, hatırlama ve problem çözme yeteneklerine sahip karmaşık bir sistemdir. Beyin fonksiyonlarının bilgisayarla taklit edilmesine yönelik girişimlerin başarısı henüz kısmi olmaktan öteye gidememiştir.

Bu karmaşık yapının temel birimi nörondur. Şekil B.1'de gerçek sinir hücresinin şematik gösterimi verilmiştir. Nöron, dendritler aracılığıyla sinyalleri alır ve birleştirir. Bileşke sinyalin yeterince güçlü olduğu durumda nöron ateşleme yapar ve sinyal, terminalleri aracılığıyla diğer nöronların dendritleriyle bağlantılı olan akson boyunca yol alır. Akson boyunca yol alarak nörona ulaşan sinyaller, elektriksel yükü değişken hızlarda ileten sıvı ile dolu çok küçük boşluklardan geçerler. Bu boşluklar, sinaptik bağlantılar olarak anılır. Sinaptik bağlantının empedans veya kondüktans değerinin ayarlanması, bellek oluşumu ve öğrenmeyi sağladığından kritik önemdedir[7].



Şekil B.1 Gerçek sinir hücresinin şematik gösterimi

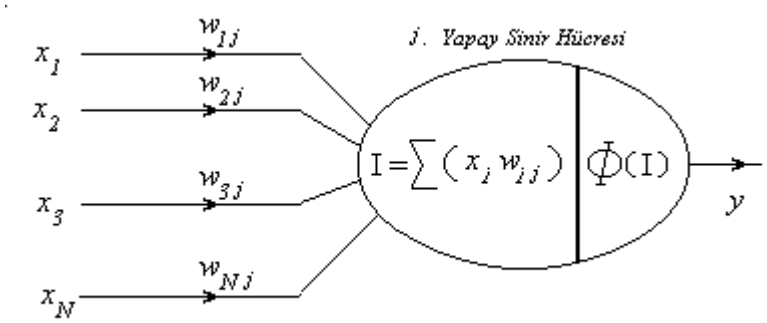
Benzeşim yapılacak olursa, beyin, doğrusal olmayan paralel bir bilgisayar olarak düşünülebilir. Yapısında yer alan nöronları düzenleme yeteneği sayesinde algılama, tanıma gibi olguları bilgisayarlardan daha hızlı yapabilmektedir. Gigahertz mertebesine ulaşan saat frekansına rağmen seri modda çalışıyor olmaları sebebiyle hala yeterince hızlı olamayan bilgisayarların başarımını arttırmak üzere yapay sinir ağı entegreleri geliştirilmektedir. Tablo B.1'de insan beyni ve bilgisayarların karşılaştırılması verilmiştir.

Tablo B.1 İnsan beyni ve bilgisayar karşılaştırması

	İşlemci Eleman	*Eleman Boyutu*	*İşlem Hızı*	*İşlem Tarzı*	*Hata Toleransı*	*Öğrenme Yeteneği*	*Zeka*
	10^{14} Sinaps 10^{10} Nöron	10^{-6} m	100 Hz	*paralel, dağınık*	*var*	*var*	*genelde var*
	10^{8} Tranzistör	10^{-6} m	10^{9} Hz	*seri, merkezi*	*yok*	*biraz*	*kazandırılmaya çalışılıyor*

B.1.1.1 Yapay Sinir Hücresi

Sinir ağları biribirine paralel olarak çalışan basit elemanlardan oluşur. Gerçek bir sinir hücresinin birimlerine eşdeğer bileşenlerle modellenen yapay sinir hücresi şekil B2'de gösterilmiştir. Gövdenin giriş birimi olan bağlantıların herbirinin kendine ait bir ağırlık çarpanı vardır. Ağırlık değeri pozitif veya negatif olabilir. Uygulanan sinyallerin ağırlık değeriyle çarpımları, iki kısımdan oluşan gövdenin ilk kısımında toplanır. Bu toplam, ikinci kısmı tanımlayan aktivasyon fonksiyonunun argümanı olur.



Şekil B.2 Yapay Sinir Hücresi

B.1.1.2 Nöron Modelleri

Nöronlar, yapay sinir ağlarının bilgi işleyen yapısal elemanlarıdır. Bir nöronun yapısında üç temel yapıtaşı bulunur:

1. Sinaps adı verilen bağlantılar: Her sinapsın kendine ait w_{ij} ile gösterilen bir ağırlık çarpanı vardır. Bu ifadede i ile söz konusu nöron, j ile sinapsın giriş uygulanan ucu tanımlanmaktadır. Ağırlık çarpanı pozitif değerli olabileceği

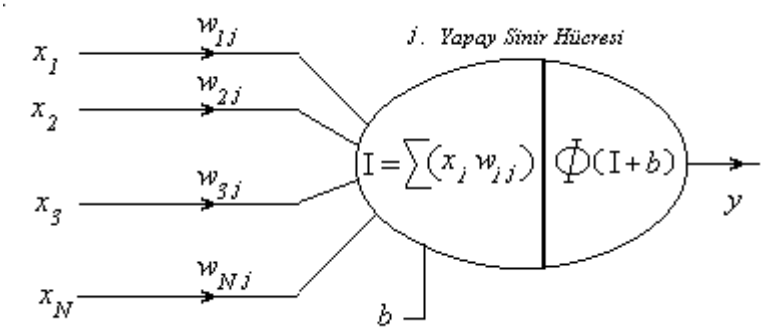
gibi negatif değerli de olabilir.

2. Toplayıcı: Uygun ağırlıkların uygulanmış olduğu giriş sinyallerini toplamak için kullanılır.
3. Aktivasyon fonksiyonu: Nöronun çıkışının genliğini kısıtlamak için kullanılır. Genelde bir nöronun normalize edilmiş genliği [0,1] veya [-1,1] kapalı aralığında ifade edilir.

Ayrıca, şekil B.3'te görüldüğü gibi nöronlar dışarıdan etki eden biaslara da sahip olabilirler. Bias pozitif veya negatif değerli olmasına bağlı olarak aktivasyon fonksiyonunun net girişini arttırmak veya azaltmak için kullanılır. Giriş sinyali x_i, sinaptik ağırlıklar w_{ij}, giriş sinyalleri toplamı I_j, bias b_j, aktivasyon fonksiyonu $\Phi(\ )$ ve nöronun çıkış sinyali y_j olmak üzere bir j nöronu matematiksel olarak şöyle tarif edilir:

$$I_j = \sum_{i=1}^{N} w_{ij} x_i \tag{B1.1}$$

$$y_j = \Phi(I_j + b_j) \tag{B1.2}$$



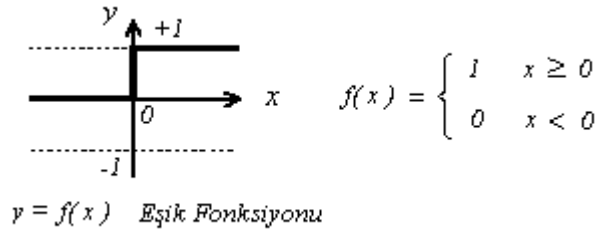
Şekil B.3 Bias girişli Yapay Sinir Hücresi

B.1.1.3 Aktivasyon Fonksiyonu Çeşitleri

$arg = I_j + b_j$ tanımlaması kullanılarak bir nöron için aktivasyon fonksiyonu $\Phi(arg)$ ifadesiyle gösterilir. Aktivasyon fonksiyonunun üç temel tipi takip eden alt başlıklarda verilmiştir.

B.1.1.3.1 Eşik Fonksiyonu

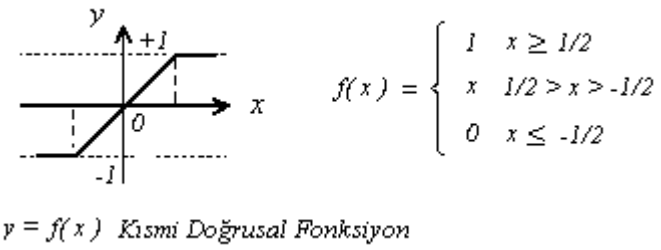
Eşik fonksiyonu kullanılarak yapılmış bir nöron literatürde McCulloch-Pitts modeli olarak adlandırılır. Fonksiyonun grafiği şekil B.4'te gösterilmiştir.



Şekil B.4 Eşik Fonksiyonu

B.1.1.3.2 Kısmi Doğrusal Fonksiyon

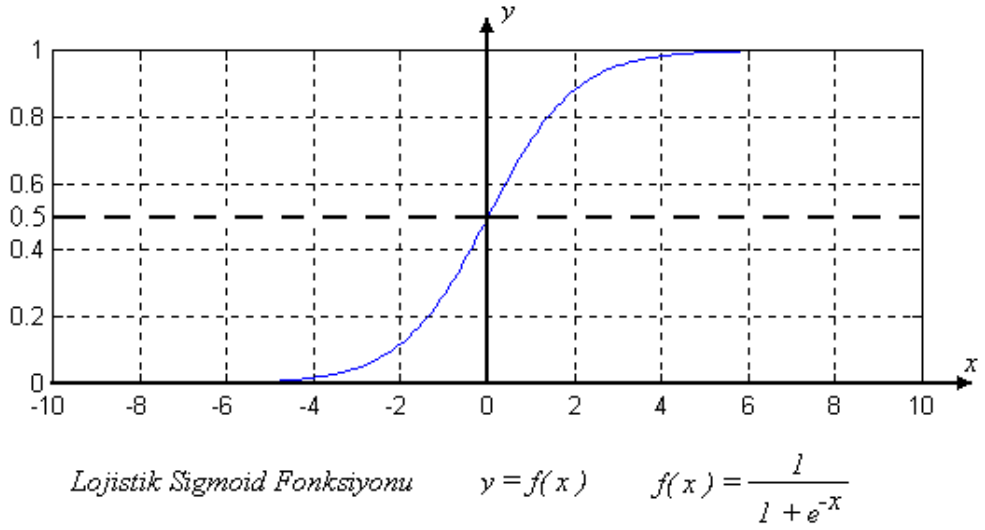
Doğrusal olmayan bir genlik artımı sağlayan bu aktivasyon fonksiyonu şekil B.5'te gösterimiştir. Eğer doğrusal bölgedeki genlik arttıran katsayı yeterince büyük alınırsa parçalı doğrusal fonksiyon eşik fonksiyonunua dönüşür.



Şekil B.5 Kısmi doğrusal fonksiyon

B.1.1.3.3 Sigmoid Fonksiyonu

Yapay sinir ağları oluşturulurken en çok kullanılan aktivasyon fonksiyonudur. Doğrusal ve doğrusal olmayan davranışlar arasında denge sağlayan sürekli artan bir fonksiyon olarak tanımlanır. Sigmoid fonksiyona bir örnek lojistik fonksiyondur ve şekil B.6'da gösterilmiştir.

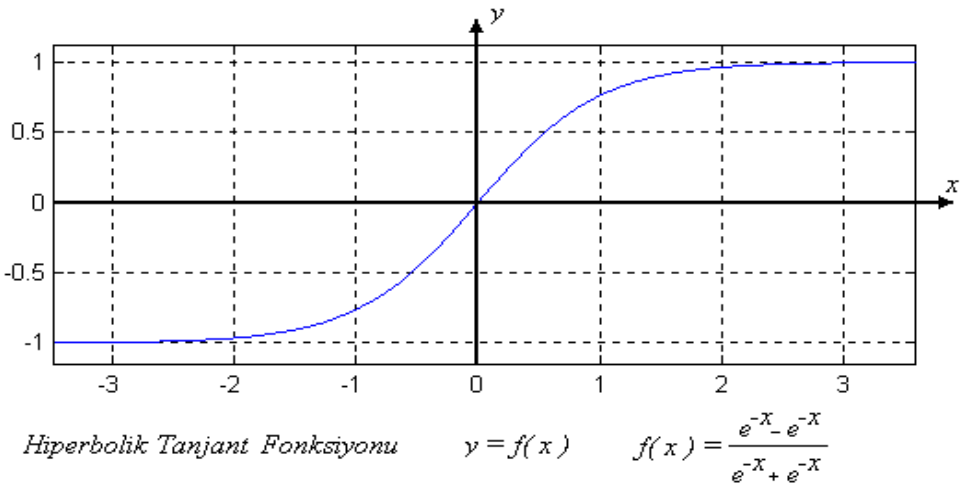


Şekil B.6 Lojistik Sigmoid fonksiyonu

Görüleceği üzere sigmoid fonksiyonunun türevi alınabilirken eşik fonksiyonunun türevi alınamaz.
Hiperbolik tanjant fonksiyonu da sigmoid fonksiyon örneğidir ve şekil B.7'de görülebilir.

B.1.2 Yapay Sinir Ağı

Yapay Sinir Ağı, öngörülen sayıda yapay sinir hücresinin, veri işlemek amacıyla belirli bir mimaride yapılandırılmasıyla şekillenir. Bu yapı, genellikle,



Şekil B.7 Hiperbolik Tanjant fonksiyonu

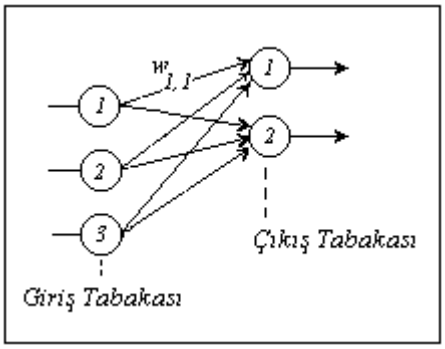
numaralandırılan birkaç katmandan oluşur. İlk katman, çoğunlukla numaralandırılmayan, giriş katmanıdır. Bu katmanın numaralandırılmaya katılmayışının sebebi, giriş katmanındaki elemanların ağırlık çarpanları ve aktivasyon fonksiyonlarının olmaması sebebiyle veri girişinden başka bir işlem yapmamalarıdır. Çıkış katmanı da son katmandır. Tercihe bağlı olarak farklı sayıda olabilen diğer ara katmanların ortak adı gizli katmandır.

B.1.2.1 Ağ Yapıları

Ağ yapıları tek katmanlı ileri beslemeli, çok katmanlı ileri beslemeli ve döngülü yapay sinir ağları olmak üzere üç temel başlıkta toplanabilir.

B.1.2.1.1 Tek Katmanlı İleri Beslemeli Sinir Ağları

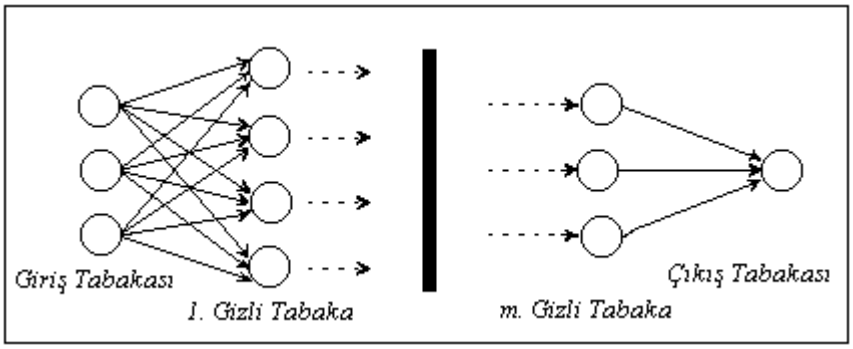
Tek katmanlı ileri beslemeli yapay sinir ağı en basit ağ yapısıdır. Bir giriş katmanı ve bir çıkış katmanı vardır. Örnek yapısı şekil B.8'de gösterilmiştir. Bu tip bir ağda bilgi girişten çıkışa doğru ilerler yani ağ ileri beslemedir. Tek katmanlı olarak isimlendirilmesinin sebebi, giriş katmanının veri üzerinde hiçbir işlem yapmadan veriyi çıkış katmanına iletmesidir.



Şekil B.8 Tek katmanlı yapay sinir ağı

B.1.2.1.2 Çok Katmanlı İleri Beslemeli Sinir Ağları

Bu tip yapay sinir ağlarının özelliği, şekil B.9'da da görüleceği üzere bir veya daha fazla gizli katman içermesidir. Gizli katmanların amacı giriş ve çıkış katmanları arasında gerekli bir takım işlemler yapmaktır. Giriş katmanı geniş olduğu zaman gizli katmanlar sayesinde yüksek dereceli istatistiksel veri elde edilebilir. Çok katmanlı yapılarda *(n).* katmanın çıkış sinyalleri *(n+1).* katmanın giriş sinyalleri olarak kullanılır. *m* adet giriş düğümü, ilk gizli katmanında h_1 adet nöron, ikinci gizli katmanında h_2 adet nöron ve çıkış katmanında *q* adet nöron bulunan bir çok katmanlı ileri besleme ağı *m-h1-h2-q* ağı olarak adlandırılır. Eğer her katmanda bulunan nöronlar bir sonraki katmanın tüm nöronlarına bağlı ise bu tip ağa tam bağlantılı ağ denir. Eğer bu sinaptik bağlantılardan bazıları eksikse ağ, kısmi bağlantılı ağ adını alır.



Şekil B.9 Çok katmanlı yapay sinir ağı

B.1.2.1.3 Döngülü Yapay Sinir Ağları

Döngülü yapay sinir ağlarının ileri beslemeli ağlardan farkı en az bir adet geri besleme çevriminin olmasıdır.

Yukarıda verilen sınıflandırmada, bağlantıların simetrik veya asimetrik olması durumuna göre alt sınıflar ortaya çıkar. *i* nöronundan *j* nöronuna yönelik bir bağlantı varsa *j*'den *i*'ye yönelik bir bağlantı da vardır. Bu iki bağlantının ağırlıkları $w_{ij}=w_{ji}$ eşitse bağlantı simetriktir denir. Eşitsizlik durumunda, bağlantı asimetrik olur.

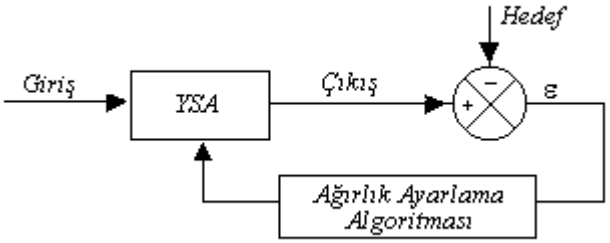
Farklı katmanlara ait nöronların bağlantısına, katmanlararası(interlayer) bağlantı denir. Aynı katmandaki nöronların bağlantısına, katmaniçi(intralayer) bağlantı, komşu olmayan katmanlardaki nöronların bağlantısınada katmanlarötesi(supralayer) bağlantı denir. Bunlardan başka, bir nöron kendisiyle bağlantılı olabilir. Sıkça kullanılan bir terim olan tam-bağlantılılık, bir katmana ait tüm nöronların komşu katmandaki tüm nöronlarla bağlantılı olduğu durumu tanımlar.

B.1.2.2 YSA'da Öğrenme ve Hatırlama

YSA'nın gerçekleştirdiği iki temel fonksiyon, öğrenme ve hatırlamadır. Öğrenme, ağırlık değerlerinin, bir giriş vektörüne karşılık arzu edilen çıkış vektörünü sağlamak üzere uyarlanmasıdır. YSA'nın, belirli bir girişe, ağırlık değerlerine uygun bir çıkış üretmesi de hatırlama olarak tanımlanır.
Ağırlık değerlerinin ayarlandığı öğrenme süreci denetimli, denetimsiz olabilir. Aralarındaki farkın kaynağı eğitim verisinin sınıflandırmasını yapan denetim mekanizmasının olup olmadığıdır. Bu durumda, denetimsiz öğrenme sürecinde

öğrenmenin yanısıra eğitim verisinin sınıflandırması da başarılması gereken bir başka görevdir.

Mühendislik uygulamalarının büyük çoğunluğu denetimli öğrenmeyi kullanır. Yapay sinir ağına, yapması istenilen göreve dayalı bir dizi örnek bilgi verilerek sinir ağı eğitilir. Burada amaç belirli bir giriş için hedef bir çıkış elde etmektir. Hedef çıkış, denetmen tarafından sağlanır. Elde edilen çıkışla hedef karşılaştırıldığı zaman hedefe ulaşılamamışsa bağlantıların ağırlıkları benimsenen öğrenme yaklaşımına göre ayarlanarak işlem tekrarlanır. Blok diagram olarak şekil B.10'da gösterilmiştir.



Şekil B.10 YSA'nın eğitimi

B.1.2.3 Geriyayınım Algoritması

Uygun ağırlıkların bulunması için en sık kullanılan yöntem geriyayınımdır. Geriyayınım algoritması, bir hedef fonksiyonu minimize etmek üzere tasarlanmış optimizasyon tekniğidir[24]. En sık kullanılan hedef fonksiyon hatanın karesidir. ε, hata terimini göstermek üzere, hata (B1.3.a) ve hedef fonksiyonun tanımlanmasında kullanılan hatanın karesi (B1.3.b) eşitliğiyle verilir.

$$\varepsilon = \varepsilon_q = \left[T_q - \Phi_{qk}(\mathrm{I})\right] \tag{B1.3.a}$$

$$\varepsilon^2 = \varepsilon_q^2 = \left[T_q - \Phi_{qk}(\mathrm{I})\right]^2 \tag{B1.3.b}$$

Yukarıda kullanılan gösterimde, k katman numarası, q nöron numarasını belirlemektedir. Delta kuralıyla ifade edildiği üzere, ağırlık değerindeki değişim, hatanın karesinin ağırlığa göre değişim oranıyla orantılıdır.

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon_q^2}{\partial w_{pq,k}} \tag{B1.4.a}$$

Kısmi türev, zincir kuralı kullanımıyla açılarak

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} \frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} \frac{\partial \mathrm{I}_{q,k}}{\partial w_{pq,k}} \tag{B1.4.b}$$

Bu eşitlikte,

$$\frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} = -2\left[T_q - \Phi_{q,k}(\mathrm{I})\right] \tag{B1.5.a}$$

$$\frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} = \alpha \Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] \text{ ve} \qquad \text{(B1.5.b)}$$

$$\mathrm{I}_{q,k} = \sum_{p=1}^{n} w_{pq,k} \Phi_{p,j}(\mathrm{I}) \qquad \text{(B1.5.c)}$$

$$\frac{\partial \mathrm{I}_{q,k}}{\partial w_{pq,k}} = \Phi_{p,j}(\mathrm{I}) \qquad \text{(B1.5.d)}$$

(B.1.5) eşitlikleri (B1.4.b) eşitliğine yerleştirilirse,

$$\Delta w_{pq,k} = -\eta_{p,q}(-2\alpha)\left[T_q - \Phi_{q,k}(\mathrm{I})\right]\Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right]\Phi_{p,j}(\mathrm{I}) \qquad \text{(B1.6)}$$

Esasen geriyayılacak hata terimi olan δ aynı zamanda kısa bir gösterim elde etmek için

$$\delta_{pq,k} = (2\alpha)\left[T_q - \Phi_{q,k}(\mathrm{I})\right]\Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] \qquad \text{(B1.7)}$$

olarak tanımlanırsa,

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon_q^2}{\partial w_{pq,k}} = -\eta_{p,q} \delta_{pq,k} \Phi_{p,j}(\mathrm{I}) \qquad \text{(B1.8)}$$

N, iterasyon sayacı olmak üzere, (N+1). adım için ayarlanacak ağırlık değeri

$$w_{pq,k}(N+1) = w_{pq,k}(N) - \eta_{p,q} \delta_{pq,k} \Phi_{p,j}(\mathrm{I}) \qquad \text{(B1.9)}$$

(B1.9) eşitliğinde tanımlanan işlem, uygun ağırlık değerlerine ulaşabilmek için çıkış katmanının tüm nöronlarına uygulanır. Hedef değerlere ulaşılamamasının bir sebebi hatalı çıkış katmanı ağırlıklarıyken diğeri, gizli katmanın ürettiği hatalı çıkışlardır. Gizli katmanın ağırlıklarının ayarlanmasında kullanılan denklemler, hedef değer olmaksızın hesaplanması gereken hata terimi $\delta_{hp,j}$ haricinde aynıdır. $\delta_{hp,j}$, çıkış katmanında bağlantılı olduğu her nöronun hata terimine katkı yapan gizli katman nöronlarının herbiri için ayrıca hesaplanmalıdır.

Hatanın ağırlıklara göre gradyanı başlangıç noktası alınıp zincir kuralıyla devam edilirse,

$$\Delta w_{hp,j} = -\eta_{h,p} \frac{\partial \varepsilon_q^2}{\partial w_{hp,j}} \qquad \text{(B1.10.a)}$$

$$= -\eta_{h,p} \sum_{q=1}^{r} \frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} \frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} \frac{\partial \mathrm{I}_{q,k}}{\partial \Phi_{p,j}(\mathrm{I})} \frac{\partial \Phi_{p,j}(\mathrm{I})}{\partial \mathrm{I}_{p,j}} \frac{\partial \mathrm{I}_{p,j}}{\partial w_{hp,j}} \qquad \text{(B1.10.b)}$$

(B1.10.b) Eşitliğinin sağ tarafındaki terimleri ayrı ayrı ele alacak olursak,

$$\frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} = (-2)\left[T_q - \Phi_{q,k}(\mathrm{I})\right] \qquad \text{(B1.11.a)}$$

$$\frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} = \alpha \Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] \quad \text{(B1.11.b)}$$

$$\frac{\partial \mathrm{I}_{q,k}}{\partial \Phi_{p,j}(\mathrm{I})} = w_{pq,k} \quad \text{(B1.11.c)}$$

$$\frac{\partial \Phi_{p,j}(\mathrm{I})}{\partial \mathrm{I}_{p,j}} = \alpha \Phi_{p,j}(\mathrm{I})\left[1 - \Phi_{p,j}(\mathrm{I})\right] \quad \text{(B1.11.d)}$$

$$\frac{\partial \mathrm{I}_{p,j}}{\partial w_{hp,j}} = x_h \text{'dir.} \quad \text{(B1.11.e)}$$

(B1.11) eşitlikleri (B1.10.b) eşitliğinde yerlerine koyulursa,

$$\frac{\partial \varepsilon^2}{\partial w_{hp,j}} = \sum_{q=1}^{r} (-2)\left[T_q - \Phi_{q,k}(\mathrm{I})\right]\alpha \Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] w_{pq,k}\, \alpha \Phi_{p,j}(\mathrm{I})\left[1 - \Phi_{p,j}(\mathrm{I})\right] x_h \quad \text{(B1.11.a)}$$

$$= \sum_{q=1}^{r} \delta_{pq,k}\, w_{pq,k}\, \alpha \Phi_{p,j}\left[1 - \Phi_{p,j}(\mathrm{I})\right] x_h \quad \text{(B1.11.b)}$$

$$\delta_{hp,j} = \delta_{pq,k}\, w_{pq,k}\, \frac{\partial \Phi_{p,j}(\mathrm{I})}{\partial \mathrm{I}_{p,j}} \quad \text{(B1.12)}$$

Son adım olarak, gizli katman nöronunun (N+1). iterasyon için hesaplanan değeri,

$$w_{hp,j}(N+1) = w_{hp,j}(N) - \eta_{hp}\, x_h\, \delta_{hp,j} \quad \text{(B1.13)}$$

Tüm gizli katman nöronları için değişiklik yapıldıktan sonra yeni girişler uygulanır ve süreç yeniden başlar. Hedeflenen hata kriterine ulaşılana dek iterasyon devam eder. Hata kriterine ulaşıldığında eğitim tamamlanmış olur.

B.1.2.4 Geriyayınımlı Eğitimi Etkileyen Etmenler

Geriyayınımlı eğitimin başarımını arttırmak için yapılabilecek bir takım düzenlemeler önerilmiştir.

B.1.2.4.1 Bias

Her bir nöron için bir bias elemanı ilave edilebilir. Aktivasyon fonksiyonunun apsisi kestiği noktayı öteleyerek nöronun eşik seviyesinde değişiklik etkisi yaratır. Genelde, eğitim hızını olumlu etkiler. Giriş elemanlarının biası (+1) olmak durumunda olmasına karşın diğer biaslar herhangi bir değer alabilir ve eğitilebilir.

B.1.2.4.2 Momentum

Hareketli bir cismin momentumunun etkisine benzer, eğitim sürecinin yönünün korunması sağlar. Bunun için, ağırlık ayarlaması sırasında, önceki ağırlık değişimiyle orantılı bir terim ilave edilir. μ , momentum terimi olmak üzere,

$$w_{pq,k}(N+1) = w_{pq,k}(N) - \eta_{p,q}\, \delta_{pq,k}\, \Phi_{p,j}(\mathrm{I}) + \mu \Delta w_{pq,k}(N) \quad \text{(B1.14)}$$

Yerel minimumdan kurtulmayı sağlayabildiği için ilgi gören düzenlemelerdendir. Ancak, geriyayınımın olamadığı gibi momentum ilavesi de her derde deva değildir.[7]

B.1.2.4.3 Öğrenme Katsayısı(η)

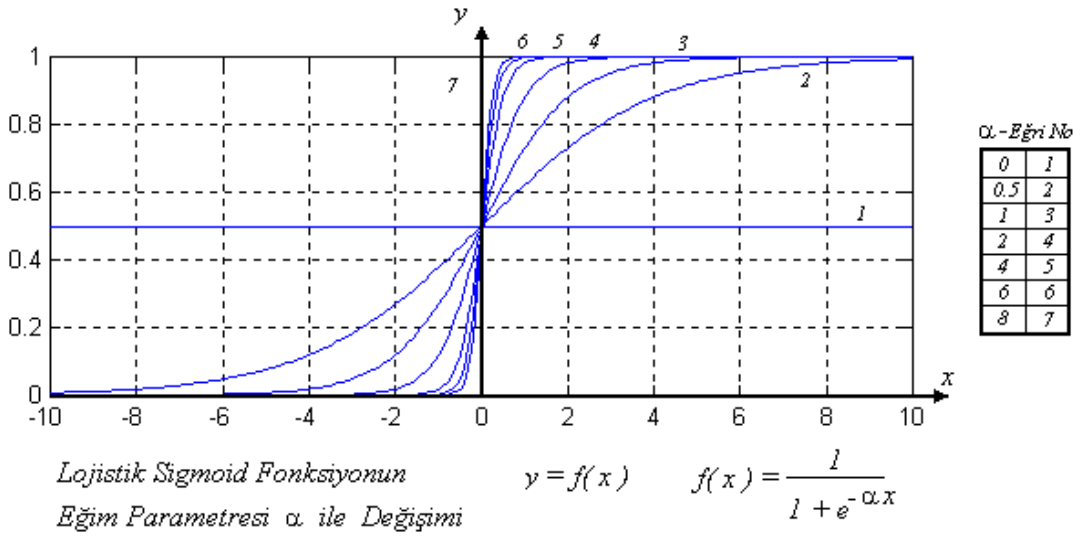
Pozitif değerli olmak zorundaki öğrenme katsayısı 2'den büyük seçildiğinde YSA'nın kararsızlığına, 1'den büyük seçildiğinde de çözüme ulaşmaktansa salınım yapmasına sebep olur[7]. Öğrenme katsayısı için $(0,1)$ uygun aralıktır. Bu aralıkta seçilen katsayının büyüklüğüyle, öğrenme adım aralığı orantılıdır. Eğitim verisinin değişim oranına uygun katsayı seçilmelidir. Uyarlanabilir öğrenme katsayısı kullanımı da başvurulabilecek yöntemlerdendir.

B.1.2.4.4 Sigmoid Fonksiyonunun Eğim Parametresi(α)

Eğim parametresine bağlı değişim şekil B.11'de gösterilmiştir. Artan ağırlık değerleri, nöronun, sigmoid fonksiyonun eğiminin(türevinin) çok küçük değerli bölgelerinde işlem yapmasına sebep olur. Geri yayımlanan hata terimi türevle orantılı olduğundan, düşük eğimde yeterli eğitim gerçekleşmez. Eğimin ayarlanması, eğitim süresini ve başarısını doğrudan etkiler.

B.1.2.4.5 Yerel Minimum

Geriyayınım algoritmasının sıkıntı yaratan yanı, yerel minimuma takılmasıdır. Algoritma, gradyan azaltma yöntemini kullandığı için hata yüzeyinin eğimi negatif olduğu sürece ağırlıkların minimuma ulaşmayı sağlayacak şekilde ayarlandığı kabul edilir. Hata yüzeyi, kolaylıkla düşülebilecek fakat kurtulmanın mümkün olamayabileceği tepe ve çukurları barındırabilir. Atanan ilk ağırlık değerleri, civarında, azalan gradyan yönünde arama yapılacak noktayı belirler. Rasgele seçilen ilk noktadan, global minimuma kadar olan mesafede daha önce de değinilen yerel minimum problemini yaratabilecek hata değerleriyle karşılaşmak muhtemeldir.



Şekil B.11 Lojistik sigmoid fonksiyonunun eğim parametresiyle değişimi

B.1.3 Eğitim ve Test Verisi Seçimi

Yapay sinir ağının eğitimi ve sınaması için toplanan veri sistemin düzgün çalışma uzayını kapsamalıdır. Örnek kayıtlarının çalışma uzayının sınırlarını belirlediği ve yapay sinir ağlarının yalnızca eğitildiği çalışma aralığı için güvenilir sonuç verebildiği, yani, ekstrapolasyon yeteneğinin güvenilemeyecek derecede kısıtlı olduğu unutulmamalıdır. Genel özelliklerin net olarak belirlenmesi için örnek kayıdı koleksiyonunun geniş olması tercih edilir. Bu kayıtların bir kısmı eğitim aşamasında kullanılırken bir kısmı sınama aşamasında ağın genelleştirme yeteneğinin teyidi amacıyla kullanılır. Sınamanın başarısızlığı durumunda, sınama amacıyla kullanılan kayıtların bir kısmı eğitim verisine katılarak eğitim ve sınama işlemleri kabul edilebilir bir performans kriterine kadar tekrarlanır.

Yapay sinir ağı eğitiminde karşılaşılan temel bir sorun ezberlemedir. Yapay sinir ağının eğitim sürecindeki hata seviyesi, test sürecindeki hata seviyesine göre bariz farklılıklar gösterdiği takdirde ezberleme sorunu ile karşılaşılmış olur. Bu da tanımlanması istenen fonksiyel ilişkiden ziyade eğitim verisindeki gürültü gibi tuhaflıkların da öğrenildiği anlamına gelir. Ezberlemeyi azaltmak için yapılabilecekler:

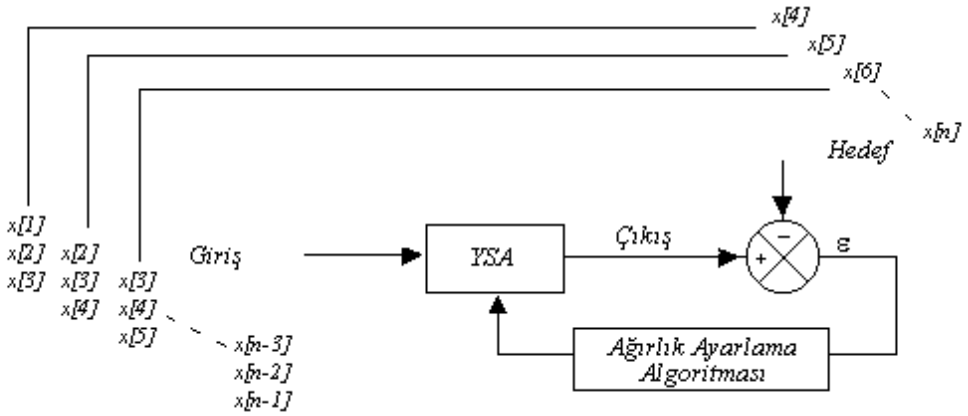
- Eğitimde kullanılan kayıt sayısını arttırarak gürültünün ortalamasının kendiliğinden düşmesini sağlamak,
- Serbest parametre olan nöronların sayısını kullanılması gerekenin asgarisi ile sınırlamak,
- Eğitimi, ezberleme başlamadan kesmek. Yöntem, çapraz değerlemeli eğitim olarak adlandırılır. Esası, eğitim sırasında ezberleme kontrolü yapmaya dayanır. Eğitim aşamasında biri eğitim, diğeri kontrol için olmak üzere iki veri grubu kullanılır. Her epokun sonunda her iki grup için hatanın RMS değeri hesaplanır ve kontrol kümesinin hatasında değişim olmadığı halde eğitim kümesinin hatasının azalmaya devam ettiği aşama tespit edilmeye çalışılır. Bu aşamada eğitim kesilir.

B.1.4 Zaman Serisi Modelleme

YSA, geçmişe ait verilerin oluşturduğu zaman serisinin modellenmesi ve geleceğe dönük kestirim yapılması işlemlerinde artan bir ilgi görmekte ve kullanılmaktadır. Mertebesi belirlenmiş AR, MA veya ARMA modeli kullanılarak yapılan kestirime benzer olarak, uzunluğu YSA'nın giriş elemanlarının sayısına eşit bir pencere, zaman serisi üzerinde adım adım ilerleyerek giriş olarak uygulanacak zaman serisi elemanlarını tanımlar. İşlem şekil B.12'de gösterilmiştir. Bu işlem, hem modelleme hem de kestirim sırasında kullanılır. Modelleme sırasında, pencere uzunluğunun *p* ile verildiği durumda, serinin *(p+1)*. elemanı hedef değer olarak alınır. Seri boyunca ilerleyen kayan pencerenin belirlediği veriler için hedeflenen hata kriterine ulaşılana dek YSA eğitimi devam eder. Geleceğe dönük kestirim, bir adımdan daha fazlası için yapılabilirse de hata artacaktır. Hatanın, bir çıkış değerine göre azaltılmaya çalışılması çözümün doğruluğunu arttıracaktır.

Bundan başka, farkı, eğitimde kullanılacak veriden kaynaklanan iki alternatif modelleme yaklaşımından birincisinde ilk adım, mevsimsel değişimler gibi periyodikliği bariz bileşenlerin veriden çıkarılmasıdır. Gerikalan, detay niteliğindeki veri üzerinden YSA eğitilir. Eğitilen YSA'nın kestirimlerinin, daha önce tespit

edilmiş periyodik bileşenlerle toplanmasıyla işlem tamamlanır. Diğer yaklaşımda eğitim verisi, ardışık elemanlar arasındaki farkın veya değişim oranının hesaplanmasıyla elde edilir.



Şekil B.12 YSA ile zaman serisi modelleme işleminin gösterimi

ÖZGEÇMİŞ

27 Aralık 1974' te Eski Cuma'da doğdu. İlköğrenimini Cebeci İlkokulu'nda tamamladıktan sonra orta öğrenimine Nişantaşı Anadolu Lisesi'nde devam etti. İTÜ Uçak ve Uzay Bilimleri Fakültesi Uzay Bilimleri ve Teknolojisi bölümünden mezuniyetinin ardından İTÜ Fen Bilimleri Enstitüsü Uzay Mühendisliği Yüksek Lisans Programı'na kayıt oldu. İyi derecede İngilizce ve başlangıç seviyesinde Fransızca bilmektedir.