AYIRMA ANALİZİNE MATEMATİKSEL PROGRAMLAMA VE YAPAY SİNİR AĞLARI YAKLAŞIMLARI

H.Hasan ÖRKCÜ

DOKTORA TEZİ
İSTATİSTİK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2009
ANKARA

H.Hasan ÖRKCÜ tarafından hazırlanan AYIRMA ANALİZİNE MATEMATİKSEL PROGRAMLAMA VE YAPAY SİNİR AĞLARI YAKLAŞIMLARI adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Hasan BAL …….
Tez Danışmanı, İstatistik Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile İstatistik Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Prof. Dr. Gülsüm HOCAOĞLU …….
İstatistik, Hacettepe Üniversitesi
Prof. Dr. Hasan BAL …….
İstatistik, Gazi Üniversitesi
Prof. Dr. A. Alptekin ESİN …….
İstatistik, Gazi Üniversitesi
Prof. Dr. Ayşen APAYDIN …….
İstatistik, Ankara Üniversitesi
Prof. Dr. İhsan ALP …….
İstatistik, Gazi Üniversitesi

Tarih: 10/06/2009

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onamıştır.

Prof. Dr. Nail ÜNSAL …….
Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

H.Hasan ÖRKCÜ

AYIRMA ANALİZİNE MATEMATİKSEL PROGRAMLAMA VE YAPAY SİNİR AĞLARI YAKLAŞIMLARI

(Doktora Tezi)

H.Hasan ÖRKCÜ

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2009

ÖZET

Ayırma analizi sosyal bilimlerde, finans, pazarlama ve muhasebe gibi iş alanlarında ve biyoloji gibi taksonomi ve sınıflama analizlerinin içerildiği diğer alanlarda geniş bir kullanımı olan bir araçtır. Bu çalışmada sınıflandırma problemleri için biri matematiksel programlamaya dayalı diğeri de yapay sinir ağlarına dayalı iki yeni yaklaşım önerilmiştir. Yeni matematiksel programlama modeli literatürde önerilen matematiksel programlama modellerinin güçlü özelliklerine dayanmaktadır. Sınıflandırma problemlerinde yapay sinir ağları da oldukça geniş bir kullanıma sahiptir. Yapay sinir ağlarının eğitilmesinde kullanılan geri yayılım algoritması yerel çözümlere yakalanma ve bazı durumlarda düşük sınıflandırma performansı vermesi gibi olumsuzluklara sahiptir. Bu çalışmada, yapay sinir ağlarının eğitilmesi sürekli parametreli genetik algoritma ile ele alınmış ve sürekli parametreli genetik algoritma ile eğitilmiş ağ yapısı sınıflandırma modellerinin çözümünde kullanılmıştır. Hem yeni önerilen matematiksel programlama modeli hem de sürekli parametreli genetik algoritma ile eğitilmiş ağ yapısı diğer geleneksel yöntemlerle beraber

literatürden alınan 12 farklı gerçek sınıflandırma probleminde ve simülasyon verisinde sınanmıştır.

Sonuçlar yeni önerilen matematiksel programlama modeli ve sürekli parametreli genetik algoritma ile eğitilmiş ağ yapısının diğer sınıflandırma yöntemlerine göre daha iyi olduğunu göstermektedir.

Bilim Kodu : **205.1.148**
Anahtar Kelimeler : **Ayırma Analizi, Matematiksel Programlama, Yapay Sinir Ağları Yaklaşımları, Genetik Algoritma.**
Sayfa Adedi : **154**
Tez Yöneticisi : **Prof. Dr. Hasan BAL**

MATHEMATICAL PROGRAMMING AND ARTIFICIAL NEURAL NETWORK APPROACHES TO DISCRIMINANT ANALYSIS

(Ph.D. Thesis)

H.Hasan ÖRKCÜ

GAZİ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

June 2009

ABSTRACT

Discriminant analysis is widely used research tool in social sciences, in business areas such as finance, marketing, and accounting, and in other areas involving taxonomical and classification analyses such as biology. In this study, a new mathematical programming model and an artificial neural networks approach are proposed for classification problems. New mathematical programming model bases on the strong features of some mathematical programming models in the literature. Artificial neural networks have also a wide ranging usage area in the classification problems. Back-propagation algorithm, used in the training of the artificial neural networks, has negative features such as being captured in the local solutions and low performance of classification in some states. In this study, training of the artificial neural networks is implemented with real-coded genetic algorithm and network structure that has been trained with real-coded algorithm has been used in the solutions of the classification models. Both newly proposed mathematical programming model and network structure, and other conventional discriminant methods were tested by using 12 different real classification data taken from the literature and simulation data.

The results show that classification success of new proposed mathematical programming model and artificial neural network approach trained with real-coded genetic algorithm are better than other classification methods.

Science Code : 205.1.148
Key Words : Discriminant Analysis, Mathematical Programming, Artificial Neural Network Approaches, Genetic Algorithm.
Page Number : 154
Adviser : Prof. Dr. Hasan BAL

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Prof. Dr. Hasan BAL'a ve manevi destekleriyle beni hiçbir zaman yalnız bırakmayan eşime, anneme ve babama teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

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

Simgeler	**Açıklama**
c	Grupları Ayıran Eşik Değer
h	Grup Sayısı
k	Değişken Sayısı
α	Ayırma Fonksiyonu Katsayıları
μ	Ortalama Vektörü
V	Varyans-Kovaryans Matrisi
Z	Veri Matrisi

Kısaltmalar	**Açıklama**
ADALINE	Uyarlanmış Doğrusal Ayırıcı
DBD	Delta-Bar-Delta
DEA-DA	Sueyoshi Veri Zarflama Analizi Ayırma Analizi Modeli
FLDF	Fisher'in Doğrusal Ayırma Fonksiyonu.
GA	Genetik Algoritma
GMFC	Genel Çok Fonksiyonlu Sınıflandırma Modeli
GPMED	Medyandan Sapmalar Hedef Programlama Modeli
GSFC	Genel Tek Fonksiyonlu Sınıflandırma Modeli
LOO	Leave-One-Out (Birini dışarıda tutma)
LSO	Leave-Some-Out (Bikaçını dışarıda tutma)

Kısaltmalar	Açıklama
LPMED	Medyandan Sapmalar Doğrusal Programlama Modeli
med	Medyan (ortanca)
MLP	Çok Katmanlı Ağ
MSD	Sapmalar Toplamının Minimizasyonu (Minimization Sum of Deviations)
MLM	Lam, Choo ve Moy'un Çok Gruplu Sınıflandırma Modeli
R&S	Ragsdale ve Stam'ın İki Aşamalı Modeli
VZA	Veri Zarflama Analizi
YSA	Yapay Sinir Ağları

1. GİRİŞ

Bilimin en temel yöntemlerinden bir tanesi olan sınıflandırma, insanoğlu tarafından üstlenilen en eski bilimsel uğraşlardan biridir. Sınıflandırma problemi, hangi sınıfa ait olduğu bilinen örnekler kullanılarak, yeni örneklerin sınıflarının belli bir doğruluk ile belirlenmesidir. Birçok alanda başarılı çalışmalar gerçekleştirebilmek için sınıflandırma problemlerinin etkin şekilde çözülmesine ihtiyaç duyulmaktadır. Bu nedenle, farklı disiplinlerden birçok araştırmacı sınıflandırma problemi üzerinde çalışmakta ve daha iyi sonuçlar elde etmek için yeni yöntem arayışları sürmektedir.

Sınıflandırma problemi, herhangi bir veri kümesinden seçilen eğitim kümesini kullanan belirli bir tanıma sistemi yani sınıflandırıcının geliştirilmesidir. Bu sınıflandırıcılar eğitim kümesindeki noktaların hangi sınıfta olduğunu belirlemenin yanı sıra eğitim kümesinde olmayan noktalar ile karşılaştırıldıklarında bu noktaların belirli bir olasılıkla hangi sınıfta olması gerektiğini söylemeye imkân veren sistemlerdir. Sınıflandırma problemleri "ayırma analizi (discriminant analysis)" teknikleri ile incelenmektedir.

Ayırma analizi, birimlerin gözlenen özelliklerine (ölçülen değişkenlere) göre uygun gruplarına atanması işlemi ile uğraşır. Ayırma analizi sınıflandırılması istenen birimlerin grup üyeliğini kestirmek, ayrık gruplardan birine birimleri sınıflandırmak ve gruplar arasındaki farklılığı bir ya da daha fazla gözlenebilir niteliğe dayalı olarak açıklamak amacıyla kullanılmaktadır.

Ayırma analizi sosyal bilimlerde, finans, pazarlama ve muhasebe gibi iş alanlarında tıbbi teşhis, hava tahmini, kredi kartı veya kredi almak için yapılan başvuruların geri ödemesinin yapılıp yapılmayacağının tahmin edildiği kredi değerlendirme, müşteri bölümleme ve sahtekârlık belirleme gibi birçok alanda ve biyoloji gibi taksonomi ve sınıflama analizlerinin içerildiği diğer alanlarda başarı ile uygulanmaktadır.

Bireyleri iki ya da daha fazla gruba sınıflandırmak için farklı kriterler kullanılmaktadır. Fisher'in, iki grup için ortaya atmış olduğu doğrusal ayırma

fonksiyonu, ayırma analizinde en çok kullanılan kriterdir [Anderson, 1984]. Fisher (1936), Normal dağılım varsayımı altında iki ya da daha fazla gruptan (yığın) gözlenmiş birimleri gruplardan birine sınıflandırmak için mevcut değişkenler üzerinden tanımlanacak doğrusal fonksiyonları önermiştir. Öyle ki bu doğrusal fonksiyonların katsayılar vektörü, gruplar arası farklılığı maksimum yapacak biçimde alınırlar.

İstatistiksel yöntemler kullanılarak elde edilen sonuçlar olasılıklarla ifade edilir fakat bu yöntemlerin kullanılabilmesi için gerekli olan varsayımların sağlanması çoğunlukla imkânsızdır. Sınıflandırma problemlerinin incelenmesinde istatistiksel yöntemlere alternatif olarak çok sayıda matematiksel programlama yöntemi geliştirilmiştir. Matematiksel programlama yöntemlerini kullanmak için yığınla ilgili herhangi bir varsayımın sağlanmasına gerek yoktur. Doğrusal programlama ile sınıflandırma problemlerinin incelenmesi ilk defa Fred ve Glover (1981a) tarafından yapılmıştır. Fred ve Glover, iki gruplu sınıflandırma problemleri için, sapmalar toplamının minimizasyonuna dayanan bir model önermişlerdir. Matematiksel programlama yöntemleri dağılımdan bağımsız ve sınıflama amaçları için çok kullanışlıdırlar. Matematiksel programlama yöntemleri ile çoğunlukla iki gruplu sınıflandırma problemleri incelenmiştir [Markowski ve Markowski (1985), Fred ve Glover (1986a, 1986b), Joachimsthaler ve Stam (1988), Koehler (1989), Koehler ve Erenguc (1990), Glover (1990), Rubin (1990), Lee ve Ord (1990), Stam ve Jones (1990), Stam ve Ragsdale (1992), Hosseini ve Armacost (1994), Lam ve ark. (1996), Lam ve Moy (1997, 2002), Sueyoshi (1999, 2001, 2004, 2006), Bal ve ark. (2006a, 2006b), Bal ve Örkcü (2007)].

Merkezi sinir sisteminin basitleştirilmiş bir modeli olan Yapay Sinir Ağları (Artificial Neural Networks) beyindeki sinirlerin çalışmasını taklit ederek sistemlere öğrenme, genelleme yapma, hatırlama gibi yetenekler kazandırmayı amaçlayan bir bilgi işleme sistemidir [Patterson, 1996]. Yapay Sinir Ağları (YSA) günümüzde mühendislik, elektrik-elektronik, haberleşme gibi çok çeşitli alanlarda ve ayrıca, başta regresyon analizi, zaman dizileri analizi ve ayırma analizi olmak üzere diğer geleneksel istatistiksel yöntemlere alternatif olarak da kullanılabilmektedir. Örüntü

tanıma (pattern recognition) kabiliyetleri temelinde sınıflandırma yapan ve veri üzerinde herhangi bir varsayım öne sürmeyen YSA ile de çoğunlukla iki gruplu sınıflandırma problemleri incelenmiştir [Patwo ve ark. (1993), Curram ve Mingers (1994), Holmstrom ve ark. (1997), Mengiameli ve West (1999)]. YSA, bankacılık sektöründe genellikle kredi değerlendirme ve iflas tahmini uygulamalarında kullanılmıştır [Leshno ve Spector (1996), Tam ve Kiang (1992), Pendharkar (2005)].

Bu çalışmanın amacı, sınıflandırma problemlerinin çözümünde etkin olarak kullanılabilecek yeni yaklaşımlar elde etmektir. Bu amaçla sınıflandırma problemleri için önerilen matematiksel programlama ve yapay sinir ağları yaklaşımları, kuvvetli ve zayıf yanları ile ayrıntılı bir şekilde incelenmiş ve literatürde önerilen yöntemlerden daha etkin olan biri matematiksel programlama açısından diğeri de yapay sinir ağları açısından iki yeni sınıflandırma yaklaşımı önerilmiştir. Yeni önerilen çok gruplu matematiksel programlama yaklaşımı bölüm 4.2'de ve yeni önerilen sürekli parametreli genetik algoritmalarla eğitilmiş çok katmanlı ağ yapısı sınıflandırma yaklaşımı ise bölüm 6.3'te verilmiştir.

Tezin ikinci bölümünde en genel hali ile ayırma analizine istatistiksel yaklaşımlar gözden geçirilmektedir. Bu amaçla Fisher'in (1936) doğrusal ayırma fonksiyonu ve karesel ayırma fonksiyonuna yer verilmiştir.

Üçüncü bölümde *iki gruplu* sınıflandırma problemleri için önerilen matematiksel programlama modelleri ayrıntıları ile gözden geçirilmektedir.

Dördüncü bölümde *çok gruplu* sınıflandırma problemleri için literatürde önerilen matematiksel programlama modelleri ayrıntıları ile incelenmiştir. Gochet ve ark. (1997) tarafından önerilen çok gruplu model ile metodolojik yönden diğer yaklaşımlardan daha güçlü olan Sueyoshi (2006)'nın çok gruplu modeli kombine edilmiş ve yeni bir, çok gruplu matematiksel programlama modeli önerilmiştir.

Beşinci bölümde yapay sinir ağ yapıları ve öğrenme algoritmaları üzerinde durulmuştur. Bu amaçla, ağ yapılarından çok katmanlı ağ yapısı başta olmak üzere

en çok kullanılan ağ yapıları ve öğrenme algoritmalarından da başta geri yayılım algoritması olmak üzere öğrenme algoritmaları tanıtılmıştır. Ayrıca Sueyoshi (1999)'dan alınan küçük bir örnek üzerinde yapay sinir ağlarının en basit modeli olan basit algılayıcı modeli ile sinir ağlarının sınıflandırma problemlerinde nasıl kullanıldığı örneklendirilmiştir.

Geri yayılım eğitim algoritmasında genellikle türeve bağlı eğim azalması yöntemi kullanılmaktadır. Eğim azalması yöntemi minimumu araştırılan çok katmanlı ağın gerçekleşen ve arzu edilen çıktı değerine bağlı olarak oluşturulan karesel hata fonksiyonunun çok fazla yerel minimumu olduğunda çoğunlukla yerel bir çözüm vermekte ve böylelikle düşük bir sınıflandırma başarısı ile karşılaşılmaktadır. Genetik algoritmalar karmaşık yapıdaki fonksiyonların minimumlarının bulunmasında kullanılan modern sezgisel yöntemlerden birisidir ve türev bilgisine ihtiyaç duymazlar [Goldberg, 1989].

Altıncı bölümde çok katmanlı ağ yapısının genetik algoritma yöntemi ile eğitilmesini ve genetik algoritma ile eğitilmiş çok katmanlı bir sinir ağının sınıflandırma problemlerinde nasıl kullanılacağı incelenmiştir. Bu amaçla, ikili kodlamalı ve sürekli parametreli genetik algoritma tekniği ele alınarak genetik algoritma yönteminin çok katmanlı ağ yapısının eğitiminde nasıl kullanılacağı tartışılmış ve genetik algoritmaların geri yayılım ve Levenberg-Marguardt algoritmaları gibi klasik algoritmalara göre üstünlüklerinden bahsedilmiştir.

Yedinci bölümde, iki gruplu ve çok gruplu durumlar için yöntemlerin karşılaştırılması yapılmaktadır. Bu amaçla Fisher'in doğrusal ayırma fonksiyonu, yeni önerilen çok gruplu matematiksel programlama yaklaşımı ve literatürde önerilen matematiksel programlama yaklaşımları, geri yayılım algoritması, Levenberg-Marguardt, ikili kodlamalı ve sürekli parametreli genetik algoritmalar ile eğitilmiş çok katmanlı ağ yapısının literatürde de kullanılan gerçek sınıflandırma problemi verileri ve simülasyon yöntemi ile sınıflandırma başarıları incelenmektedir.

Son bölümde elde edilen sonuçlar değerlendirilmektedir.

2. İSTATİSTİKSEL YAKLAŞIMLAR

İstatistik bilimi açısından ayırma fonksiyonu, özel bir istatistiksel karar verme fonksiyonudur. Hangi gruba sınıflandırılacağı tam olarak bilinemeyen bir birim, üzerinde bazı ölçümler yapılarak, bu birimin mevcut gruplardan hangisine en büyük olasılıkla sınıflandırılacağına istatistiksel yöntemlerle karar verilebilir. Burada birimin sonlu sayıdaki gruplardan birine sınıflandırılması zorunlu olmaktadır. Fisher (1936), bu problemin çözümü için söz konusu değişkenlerin doğrusal fonksiyonlarından hareket etmiştir. Şöyle ki;

$$Y = \alpha_1 x_1 + \alpha_2 x_2 + \ \ldots \ + \alpha_k x_k = \underline{\alpha}' \underline{x} \tag{2.1}$$

olsun. İki gruplu doğrusal ayırma analizinde, k boyutlu α ayırma vektörü ve c sabiti belirlenir. $\underline{\alpha}' \underline{x} \leq c$ ise ilgili birim 1. gruba, diğer durumda 2. gruba ait olarak sınıflandırılır. Bu doğrusal fonksiyon, farklı grupların kovaryans matrislerinin aynı olması durumunda uygundur. Aksi halde karesel fonksiyonların kullanımı analiz sonuçlarının sağlıklı olmasını sağlayacaktır [Lachenburch, 1975].

Çok değişkenli gözlemlerden oluşan iki veya daha fazla grup arasında ayırım yapmak ve kaynağı bilinmeyen gözlemleri bu gruplardan birine sınıflandırmak için kullanılan ayırma analizi yöntemi, uygulanabilmesi ve sonuçların sağlıklı olması için bazı varsayımların sağlanmasına bağlıdır [Anderson, 1984]. Bu varsayımlar:

a) Ortalama vektörleri birbirinden farklı iki veya daha fazla grup söz konusu olmalıdır.
b) Her bir gruptaki gözlemler, iki veya daha fazla değişken tarafından karakterize edilmelidir.
c) Analizde kullanılacak her bir örnek, çok değişkenli normal dağılımlı bir yığından seçilmiş olmalıdır.
d) Doğrusal ayırma fonksiyonunun uygun olması için, yığınların kovaryans matrisleri aynı olmalıdır.

k değişkene sahip birimlerden oluşan iki veya daha fazla grup bulunmaktadır. Bireyler, gruplar arasında ayırma gücü en yüksek olacak biçimde belli sayıda doğrusal bağıntılar yardımıyla sınıflandırılmaktadır. Birimlere ait k değişkeni, k boyutlu uzayda tanımlayacak öyle eksenler bulunmalı ki veri toplulukları bu eksenler boyunca birbirinden olabildiğince ayrılsın. İkiden çok grup olması durumunda bulunacak ayırma fonksiyonu sayısı; h grup sayısını, k değişken sayısını göstermek üzere $\min(h-1,k)$ olur. İki grup olması durumunda tek bir ayırıcı fonksiyon grupları birbirinden ayırırken, çok grup olması durumunda tek ayırıcı fonksiyon tüm grupları ayırma da yeterli olmamaktadır. Bu nedenle 2. hatta 3. ayırıcı fonksiyonlara ihtiyaç duyulmaktadır. Bulunacak ayırıcı fonksiyonlardan ilki gruplar arasında en büyük ayırımı sağlayan olacaktır. İkinci fonksiyon, ilki ile ilişkisi olmayan ve ilk ayırıcı fonksiyondan sonra gruplar arasında en iyi ayırımı sağlayan bağıntı olacaktır. Öteki fonksiyonlarda benzer biçimde yorumlanabilir.

Ayırıcı fonksiyonların bulunması Fisher'in belirlediği

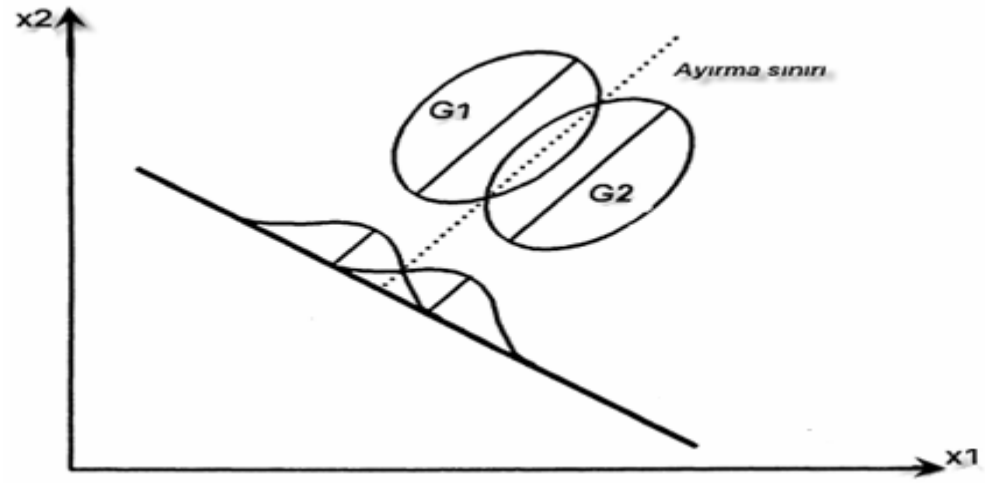
$$\lambda = \frac{v'Bv}{v'Wv} \tag{2.2}$$

iki varyans oranı biçimindedir. Buradan elde edilen λ özdeğerine karşılık gelen özvektörler yukarıda belirtilen koşulları sağlayan ayırıcı fonksiyonlar olacaktır. Bu nedenle ayırmayı maksimum yapan v değeri

$$\left|W^{-1}B-\lambda I\right|v=0 \tag{2.3}$$

eşitliğinden elde edilir. Burada $B=\sum_{j=1}^{k} n_j(\bar{X}_j-\bar{X})(\bar{X}_j-\bar{X})'$ olmak üzere gruplar arası varyans matrisi ve $W=\sum_{j=1}^{k}\sum_{i=1}^{n_j}(X_{ij}-\bar{X}_j)(X_{ij}-\bar{X}_j)'$ olmak üzere grup içi varyans matrisidir.

İki değişken için Fisher'in ayırma fonksiyonu Şekil 2.1'de verilmektedir.



Şekil 2.1. İki değişken için Fisher'in ayırma fonksiyonu

Bireylerin k değişkenli normal dağılımlı ve ortak varyans-kovaryans matrisine sahip yığından gelmesi durumu için Anderson (1984) tarafından geliştirilen yöntem, Fisher (1936)'nın kullandığı yöntemin devamı biçimindedir. Yöntem h grubun birbiri ile kıyaslanması esasına dayanmaktadır. Bu durumda $\binom{h}{2} = h(h-1)/2$ tane ayırıcı fonksiyon bulunması gerekmektedir. Bu fonksiyonlar karşılaştırılan iki grubun olasılık yoğunluk fonksiyonlarının birbirine oranlanmasından elde edilmektedir.

$$f_i(X) = \frac{1}{(2\pi)^{1/2}\left|V\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)'V^{-1}(x-\mu)\right] \tag{2.4}$$

$i.$ gruba ait yoğunluk fonksiyonu olmak üzere, varyans-kovaryans matrisleri eşit olan i ve j gibi iki grubun karşılaştırılması amacıyla,

$$f_{ij} = \frac{f_i(X)}{f_j(X)} = d_{ij} = (\mu_1 - \mu_2)'V^{-1}X - \frac{1}{2}(\mu_1 - \mu_2)'V^{-1}(\mu_1 + \mu_2) \tag{2.5}$$

fonksiyonu kullanılmaktadır. Bu fonksiyon yardımıyla $i.$ ve $j.$ grupları birbirinden ayıran bölgeler tanımlanmaktadır. Eğer $f_{ij}(X) \geq 0$ ise birey $i.$ gruba (R_i bölgesine), aksi halde $j.$ gruba (R_j bölgesine) sınıflandırılır.

Verilerin normal dağıldığı ancak varyans-kovaryans matrislerinin farklı olması durumunda kullanılan karesel fonksiyon,

$$Q(X) = \frac{1}{2}\log\frac{|S_j|}{|S_i|} - \frac{1}{2}(\overline{X}_i' S_i^{-1} \overline{X}_i - \overline{X}_j' S_j^{-1} \overline{X}_j + X'(S_i^{-1}\overline{X}_i - S_j^{-1}\overline{X}_j) - \frac{1}{2}X'(S_i^{-1} - S_j^{-1})X \qquad (2.6)$$

olarak tanımlanmaktadır. Birey, $Q(X) < 0$ ise R_j bölgesine, $Q(X) > 0$ ise R_i bölgesine sınıflandırılır.

Kovaryans matrislerinin eşit olmadığı durumlarda, doğrusal ayırma fonksiyonunun ayırma gücü etkilenmektedir. Marks ve Dunn (1974), ayırma analizinde gruplardaki birim sayısı büyüklüğü üzerinde çalışmışlar ve küçük örneklerde, gruplara ilişkin kovaryans matrisleri arasındaki farklılık fazla değilse, Fisher'in doğrusal ayırma fonksiyonunun karesel ayırma fonksiyonuna göre daha iyi sonuç verdiğini göstermişlerdir.

3. İKİ GRUPLU MATEMATİKSEL PROGRAMLAMA YAKLAŞIMLARI

Doğrusal programlama yöntemleri ile sınıflandırma problemlerinin incelenmesi ilk defa Fred ve Glover (1981a) tarafından yapılmıştır. Fred ve Glover, sapmalar toplamının minimizasyonuna dayanan bir sınıflandırma modeli önermişlerdir. Bajgier ve Hill (1982) iki gruplu sınıflandırma probleminde, doğrusal programlama modelleri ile istatistiksel yöntemlerin deneysel bir karşılaştırmasını vermişlerdir. Yapılan bu ilk çalışmalardan sonra, sapmalar toplamının minimizasyonu, yanlış sınıflandırılmış birimlerin sayısının minimizasyonu ve gruplar arası uzaklıkların maksimizasyonu gibi sınıflandırma kriterlerine dayalı birçok model geliştirilmiştir [Markowski ve Markowski (1985), Fred ve Glover (1986a, 1986b), Joachimsthaler ve Stam (1988), Koehler (1989), Koehler ve Erenguc (1990), Glover (1990), Rubin (1990), Lee ve Ord (1990), Stam ve Jones (1990), Stam ve Ragsdale (1992), Hosseini ve Armacost (1994), Lam ve ark. (1996), Lam ve Moy (1997, 2002), Sueyoshi (1999, 2001, 2004, 2006), Bal ve ark. (2006a, 2006b), Bal ve Örkcü (2007)]. Fisher'in orijinal yöntemine benzer olarak, doğrusal programlama modelleri dağılımdan bağımsız ve sınıflama amaçları için kullanışlıdır [Koehler ve Erenguc, 1990].

k değişkenli iki gruplu sınıflandırma problemi göz önüne alınsın. Z; G_1 ve G_2 gruplarından alınan n çaplı bir örneğin k. değişken skorlarını gösteren $k \times n$'lik bir matris olsun. Değişken katsayıları $\alpha_1, \alpha_2, \ldots, \alpha_k$ ile ve j. birime ait sınıflandırma skoru ise $S_j = \sum_{i=1}^{k} \alpha_i Z_{ij}$ olarak tanımlanır ($j = 1, 2, \ldots, n$). Bir birimin ait olduğu grubun tayin edilmesi o birimin sınıflandırma skorunun değerine bağlıdır. Fred ve Glover (1981a) tarafından önerilen basit MSD (Minimization Sum of Deviations-sapmalar toplamının minimizasyonu) sınıflandırma modeli,

$$Min \sum_{j \in G_1} S_j^+ + \sum_{j \in G_2} S_j^-$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + S_j^+ \geq c \quad j \in \mathrm{G}_1 \tag{3.1}$$

$$\sum_{i=1}^{k} \alpha_i Z_{ij} - S_j^- \leq c \quad j \in \mathrm{G}_2$$

biçiminde tanımlanabilir. Burada $Z_{ij} \geq 0$, $S_j^+ \geq 0$, $S_j^- \geq 0 \left(j = 1, 2, \ldots, n\right)$, $\alpha_i \left(i = 1, 2, \ldots, k\right)$ ve c işaretçe serbest (pozitif veya negatif değerler alabilen) değişkenlerdir. Eş. 3.1 ile verilen model doğrusal programlama modelidir ve Simpleks algoritması ile çözülür. Bu modelin çözülmesi ile α_i ve c değerleri elde edilerek, herhangi bir birimin sınıflandırma skoru elde edilebilir. Bir birimin sınıflandırma skoru (S_j) c'ye eşit ya da büyük ise G_1'e, diğer durumda G_2'ye sınıflandırılacaktır. $S_j^+ \geq 0$ olması, G_1 grubundaki $j.$ birimin yanlış sınıflandırıldığını; $S_j^- \geq 0$ olması, G_2 grubundaki $j.$ birimin yanlış sınıflandırıldığını göstermektedir.

Lam ve ark. (1996), MSD modelinin yaptığı işlemi iki adıma ayırmışlardır. Birincisi değişken ağırlıklarının bulunması, ikincisi de sınıflandırma için ayırma değerlerinin belirlenmesidir. Fakat bu model grup ortalamasının sınıflandırma skorundan sapmalar toplamını minimize eden bir amaç fonksiyonundan yararlanmaktadır. Lam ve ark. (1996)'nın modeli (LCM),

LCM 1

$$Min \sum_{j \in G_1} S_j^+ + \sum_{j \in G_2} S_j^-$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \mu_{1i} \right) + S_j^+ \geq 0 \qquad j \in \mathrm{G}_1 \tag{3.2}$$

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \mu_{2i} \right) - S_j^- \leq 0 \qquad j \in \mathrm{G}_2$$

$$\sum_{i=1}^{k} \alpha_i \left(\mu_{1i} - \mu_{2i} \right) \geq 1$$

ile formüle edilebilir. Eş. 3.2 modelinde, $S_j^+ \geq 0$, $S_j^- \geq 0 \left(j = 1, 2, \ldots, n \right)$, α_i $\left(i = 1, 2, \ldots, k \right)$ serbest değişken, μ_{1i}; G_1 grubundaki birimlerin *i.* değişkenine ait ortalama ve μ_{2i}; G_2 grubundaki birimlerin *i.* değişkenine ilişkin ortalamadır.

Eş. 3.2 modeli yardımı ile değişkenlerin ayırma katsayıları, α_i $\left(i = 1, 2, \ldots, k \right)$, bulunarak bireylerin skorları (S_j) elde edilir. Bu modelde birimlerin sınıflandırma skorları bulunduğu grubun ortalama skoruna yaklaştırılarak ayırma katsayılarına ulaşılır. Birimlerin sınıflandırma skorları (S_j),

LCM 2

$$\min \sum_{j=1}^{n} h_j$$

Kısıtlar: (3.3)

$$S_j + h_j \geq c \qquad j \in \mathrm{G}_1$$

$$S_j - h_j \leq c \qquad j \in \mathrm{G}_2$$

modelinde kullanılarak sınıflandırma yapılır. Eş. 3.3 modelinde, $h_j \geq 0 \left(j = 1, 2, \ldots, n \right)$, c serbest değişkendir. Görüldüğü gibi sınıflandırma işlemi iki aşamada yapılmaktadır. Bu modeller Simpleks algoritması ile çözülür.

Bal ve ark. (2006a) LCM modelinde ortalamadan sapmalar yerine medyan (ortanca) değerinden sapmalar toplamını minimize ederek özellikle Üstel, χ^2 ve F dağılımı gibi çarpık dağılımdan gelen değişkenlere sahip problemlerde başarı ile kullanılabilecek iki aşamalı LPMED modelini önerdiler. Medyan (ortanca) değerinden sapmaların kullanılmasının sebebi, çarpık dağılımlı veya normal dağılımlı olmayan yığınlardan seçilen örneklerde medyanın ortalamadan daha hassas sonuçlar vermesi ve gerçek hayat problemlerinin birçoğunun da bahsedilen dağılımlar gibi çarpık dağılımlı bir durumu yansıtabileceği olarak ifade edilmektedir. LPMED modelinin özellikle çarpık dağılımlarda LCM modeline göre yüksek sınıflandırma başarısına sahip olduğu birçok farklı durumu kapsayan simülasyon çalışması ile gösterilmiştir. LPMED modeli LCM modeli gibi iki aşamadan oluşmaktadır. Bal ve ark. (2006b) LCM ve LPMED modellerinin iki aşamada iki ayrı doğrusal programlama modelini çözerek yaptığı işlemi öncelikli hedef programlama ile tek bir sınıflandırma modeline indirgemişlerdir. Bal ve ark. (2006b) tarafından önerilen hedef programlamaya dayalı sınıflandırma modeli (GPMED)

$$Min\ a = \left\{ \sum_{j \in G_1, G_2}^{n} \left(S_j^- + S_j^+ \right), \sum_{j=1}^{n} h_i \right\}$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - med_{1i} \right) + S_j^- - S_j^+ = 0 \quad , \quad j \in \mathrm{G}_1 \tag{3.4}$$

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - med_{2i} \right) + S_j^- - S_j^+ = 0 \quad , \quad j \in \mathrm{G}_2$$

$$\sum_{i=1}^{k} \alpha_i \left(med_{1i} - med_{2i} \right) \geq 1$$

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + h_j \geq c \quad , \quad j \in \mathrm{G}_1$$

$$\sum_{i=1}^{k} \alpha_i Z_{ij} - h_j \leq c \quad , \quad j \in \mathrm{G}_2$$

ile formüle edilmektedir. Eş. 3.4 modelinde, $S_j^+ \geq 0$, $S_j^- \geq 0$ $(j = 1, 2, \ldots, n)$, α_i $(i = 1, 2, \ldots, k)$ işaretçe serbest değişken, med_{1i}; G_1 grubundaki birimlerin *i.* değişkenine ait medyan (ortanca) ve med_{2i}; G_2 grubundaki birimlerin *i.* değişkenine ilişkin medyan (ortanca) değeridir.

Bu modelde ilk öncelikte $\sum_{j \in G_1, G_2}^{n} \left(S_j^- + S_j^+ \right)$ sapmaları minimum yapılarak α_i ağırlıkları ve her bir birime ait $S_j = \sum_{i=1}^{k} \alpha_i Z_{ij}$ ayırma skor değerleri elde edilmektedir. İlk öncelik korunarak, ilk öncelikte elde edilen S_j ayırma skorları yardımıyla ikinci öncelikte $\sum_{j=1}^{n} h_i$ sapmalarının minimum yapılması ile birimlerin sınıflandırılması yapılmaktadır. Bir birim sınıflandırma skoru grupları birbirinde ayıran *c* eşik değerinden büyük ya da eşitse G_1 grubuna diğer durumda G_2 grubuna atanmaktadır. LCM ve LPMED modellerinin aksine, GPMED modelinde ağırlık değerleri ve grupları birbirinden ayıran eşik değer aynı anda elde edilmektedir. Bal ve ark. (2006) LCM1 modelinin yani LCM yönteminin ilk aşaması çoklu optimal çözümlere sahip olduğu durumlarda GPMED modelinin çoklu optimal çözümlerin arasından daha yüksek doğru sınıflandırma başarısına sahip çözümleri seçtiğini birçok farklı durumu kapsayan simülasyon çalışmalarında göstermişlerdir. LCM1 modelinin tek optimal çözümleri varsa LCM modeli ile GPMED aynı sınıflandırma başarısına sahip olacaktır. Bir doğrusal programlama modelinin çoklu optimal çözümlere sahip olması çoğu defa mümkün olabilir.

Sınıflandırma problemlerinde, bazı birimlerin hangi gruba ait olduğu tam olarak belirlenememektedir. İki gruplu durum için, G_1'deki bazı birimler diğer gruba (G_2'ye) ait olabilir veya G_2'deki bazı birimler G_1 grubuna ait olabilir. Böyle bir durum "*çakışma*" olarak adlandırılır ve önemli bir yanlış sınıflandırma kaynağıdır

[Stam ve Ragsdale (1992), Sueyoshi (1999)]. Ayırma analizinde çakışma olmayan durum ve çakışma durumu sırasıyla Şekil 3.1 ve Şekil 3.2'de gösterilmektedir.

Stam ve Ragsdale (1992) ayırma fonksiyonunun bulunması için iki aşamalı bir yöntem önermişlerdir. Bu yöntemde, çakışma durumunda olan yani sınıflandırılması zor olan birimler ilk aşamada tanımlanır ve bu birimler ikinci aşamada yeniden incelenir. Yöntemin ilk aşaması Eş. 3.5 modeli ile verilmektedir. Bu yönteme iki aşamalı MSD yaklaşımı da denilmektedir.

$$Min \sum_{j \in G_1} S_{1j} + \sum_{j \in G_2} S_{2j}$$

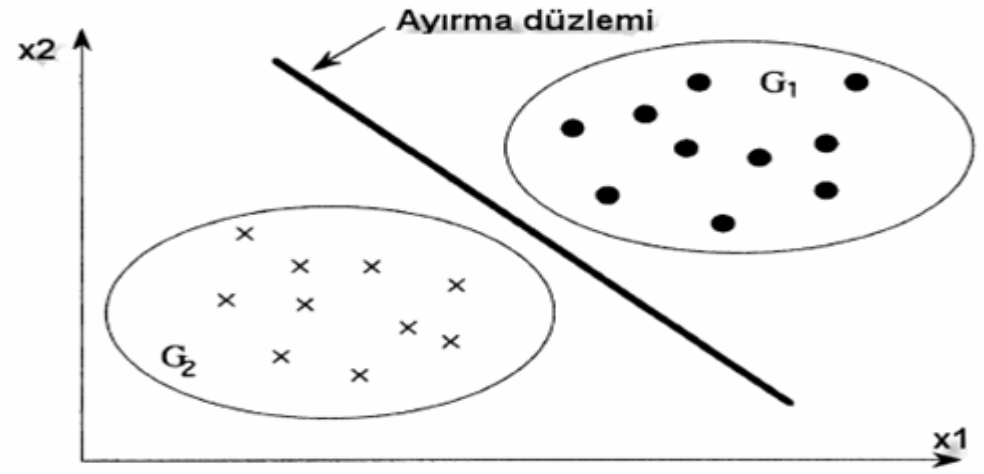
Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + S_{1j} \geq 1 \qquad j \in G_1 \tag{3.5}$$

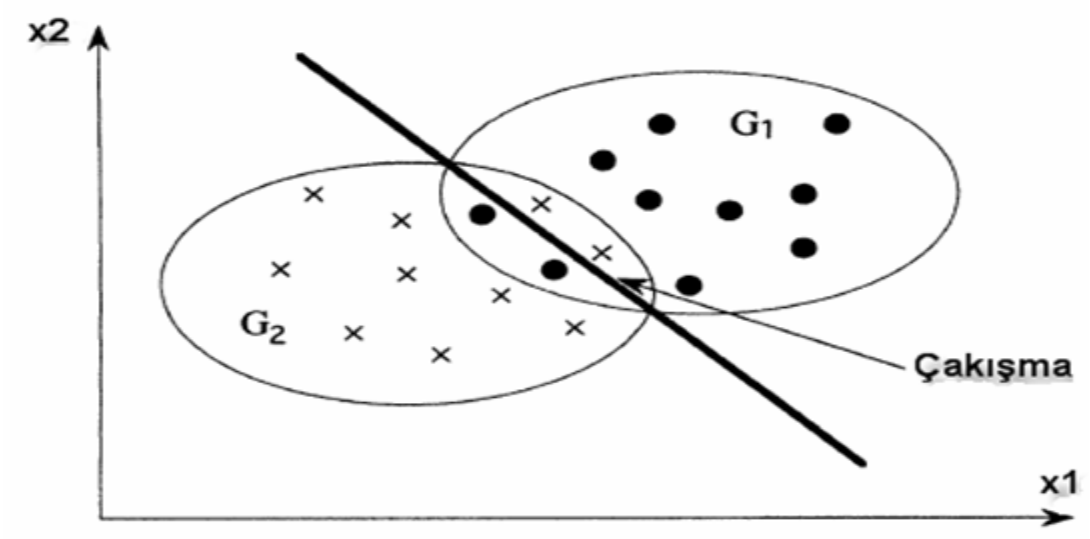
$$\sum_{i=1}^{k} \alpha_i Z_{ij} - S_{2j} \leq 0 \qquad j \in G_2$$

$S_{1j}, S_{2j} \geq 0$, α_i işaretçe serbest

Burada, G_1 grubu için sağ taraf sabiti 1 ve G_2 grubu için sağ taraf sabiti 0 değerleri gruplar arasındaki çakışmayı belirlemek için konulmuştur.



Şekil 3.1. Ayırma analizinde çakışma olmayan durum



Şekil 3.2. Ayırma analizinde çakışma durumu

Eş. 3.5 ile verilen model Simpleks algoritması ile çözülür. α^*, Eş. 3.4 modelinden elde edilen optimum değer olsun. Değeri Z_{im} ile gösterilmiş yeni örneklenen *m*. birimin G_1, G_2 ya da $G_1 \cap G_2$ (çakışma) durumlarından hangisine ait olduğu

a) $\sum_{i=1}^{k} \alpha_i^* Z_{im} \geq 1$ ise birimin G_1 grubuna ait olduğu sonucuna varılır.

b) $0 < \sum_{i=1}^{k} \alpha_i^* Z_{im} < 1$ ise birimin $G_1 \cap G_2$ 'ye ait olduğu sonucuna varılır.

c) $\sum_{i=1}^{k} \alpha_i^* Z_{im} \leq 0$ ise birimin G_2 grubuna ait olduğu sonucuna varılır.

kriterleri ile belirlenir. Bu kriterlere göre, tüm birimler ($G_1 \cup G_2$) aşağıda verilen alt kümelere ayrılabilir.

$$C_1 = \left\{ j \in G_1 \,\middle|\, \sum_{i=1}^{k} \alpha_i^* Z_{ij} \geq 1 \right\}$$

$$C_2 = \left\{ j \in G_2 \;\middle|\; \sum_{i=1}^{k} \alpha_i^* Z_{ij} \le 0 \right\}$$

$D_1 = G_1 - C_1$ ve $D_2 = G_2 - C_2$

Burada C_1 ve C_2 alt grupları sırasıyla G_1 ve G_2 gruplarına doğru olarak sınıflandırılan birimlerden oluşmaktadır. Benzeri olarak D_1 ve D_2 alt grupları çakışma durumunda olan ya da ilk aşamada yanlış sınıflandırılan birimlerden oluşmaktadır. İkinci aşamada C_1 ve C_2 gruplarında yer alan birimlerin gruplarını bozmadan D_1 ve D_2 gruplarında yer alan birimler yeni bir ayırma fonksiyonu yardımı ile değerlendirilir.

Yöntemin ikinci aşaması Eş. 3.6 modeli ile verilmektedir.

$$Min \sum_{j \in D_1} S_{1j} + \sum_{j \in D_2} S_{2j}$$

Kısıtlar:

$$\begin{aligned}
&\sum_{i=1}^{k} \alpha_i Z_{ij} \ge 1 && j \in C_1 \\
&\sum_{i=1}^{k} \alpha_i Z_{ij} \le 0 && j \in C_2 \\
&\sum_{i=1}^{k} \alpha_i Z_{ij} + S_{1j} \ge c && j \in D_1 \\
&\sum_{i=1}^{k} \alpha_i Z_{ij} - S_{2j} \le c && j \in D_2 \\
&c \le 1 \\
&S_{1j}, S_{2j} \ge 0,\ \alpha_i \text{ işaretçe serbest}
\end{aligned} \qquad (3.6)$$

Eş. 3.6 ile verilen model Simpleks algoritması ile çözülür. c^* ve α_i^* sırasıyla Eş. 3.6 modelinden elde edilen ayırma skoru ve ağırlık tahminleri olsunlar. İlk aşamada çakışma grubunda tanımlanan $m.$ birimin,

a) $\sum_{i=1}^{k} \alpha_i^* Z_{im} > c^*$ ise birimin G_1 grubuna ait olduğu sonucuna varılır.

b) $\sum_{i=1}^{k} \alpha_i^* Z_{im} < c^*$ ise birimin G_2 grubuna ait olduğu sonucuna varılır.

Sueyoshi (1999), Veri Zarflama Analizi (Data Envelopment Analysis-DEA) ile ayırma analizi arasındaki faklılıkları ve benzerlikleri hedef programlama ışığında incelediği çalışmasında, Veri Zarflama Analizinin (VZA) metodolojik gücünü ayırma analizi ile birleştirerek DEA-DA (Data Envelopment Analysis-Discriminant Analysis) adını verdiği yeni bir sınıflandırma modeli önermiştir. İlk defa Charnes ve ark. (1978) tarafından önerilen ve CCR oran formu olarak bilinen VZA yöntemi, çeşitli etkinliklerin göreli olarak hesaplanması için geliştirilmiş bir yönetim bilimi tekniğidir [Cooper ve ark., 2000]. Temel etkinlik modeli olan CCR modeli dışında da çeşitli etkinlik hesaplamaları için geliştirilmiş farklı VZA modelleri de mevcuttur. Bu modellere örnek olarak BCC modeli, toplamsal model ve çarpımsal model verilebilir. Sueyoshi toplamsal VZA modelinin ayırma gücünü ayırma analizi ile birleştirerek DEA-DA sınıflandırma modelini önermiştir.

Sueyoshi, Stam ve Ragsdale (1992) tarafından önerilen iki aşamalı modele benzer olarak, ilk aşamada çakışma durumunun tespit edildiği ikinci aşamada ise çakışma durumunda olan birimlerin sınıflandırıldığı iki aşamalı bir model önermiştir [Sueyoshi, 1999].

Aşama 1: Çalışmayı belirleme ve sınıflandırma

Çakışma belirleme süreci veya DEA-DA'nın birinci aşaması Eş. 3.7 modelinde verildiği gibi formüle edilebilir:

$$Min \sum_{j \in G_1} S_{1j}^{+} + \sum_{j \in G_2} S_{2j}^{-}$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + S_{1j}^{+} - S_{1j}^{-} = d \qquad j \in G_1$$

$$\sum_{i=1}^{k} \beta_i Z_{ij} + S_{2j}^{+} - S_{2j}^{-} = d - \eta \qquad j \in G_2 \tag{3.7}$$

$$\sum_{i=1}^{k} \alpha_i = 1$$

$$\sum_{i=1}^{k} \beta_i = 1$$

$\alpha_i, \beta_i, S_{1j}^{+}, S_{1j}^{-}, S_{2j}^{+}, S_{2j}^{-} \geq 0$, d işaretçe serbest

Eş. 3.7 modelinde, α_i ayırma fonksiyonunun *i*. ağırlığıdır. S_{1j}^{+} ve S_{1j}^{-} sırasıyla G_1 grubuna ilişkin, d ayırma skorundan $\sum_{i=1}^{k} \alpha_i Z_{ij}$ parçalı doğrusal ayırma fonksiyonuna olan pozitif ve negatif sapmaları göstermektedir. $S_{1j}^{+} > 0$ olması G_1 grubundaki *j*. birimin yanlış sınıflandırılmış olduğunu göstermektedir. Bu nedenle yanlış sınıflandırmayı minimum yapmak için amaç fonksiyonunda $\sum_{j \in G_1} S_{1j}^{+}$ minimum yapılmak istenmektedir. $S_{1j}^{-} > 0$ olması G_1 grubundaki *j*. birimin doğru sınıflandırılmış olduğunu göstermektedir. β_i diğer ayırma fonksiyonunun *i*. ağırlığıdır. S_{2j}^{+} ve S_{2j}^{-} ise sırasıyla G_2 grubuna ilişkin, $d - \eta$ ayırma skorundan $\sum_{i=1}^{k} \beta_i Z_{ij}$ başka bir parçalı doğrusal ayırma fonksiyonuna olan pozitif ve negatif sapmaları göstermektedir. η çok küçük pozitif sayıdır ve tüm ağırlıkların sıfır olması gibi uygun olmayan sonuçlardan kaçınmak için modele dahil edilmiştir. $S_{2j}^{-} > 0$ olması G_2 grubundaki *j*. birimin yanlış sınıflandırılmış olduğunu göstermektedir. Bu nedenle yanlış sınıflandırmayı minimum yapmak için amaç fonksiyonunda $\sum_{j \in G_2} S_{2j}^{-}$ minimum yapılmak istenmektedir. $S_{2j}^{+} > 0$ olması G_2 grubundaki *j*. birimin doğru sınıflandırılmış olduğunu göstermektedir.

Birinci aşama; G_1'deki birimlerin G_2 içinde yer alması ya da G_2'deki birimlerin G_1 içinde yer alması biçimindeki yanlış sınıflandırmaları minimize etmek için tasarlanmıştır. Denklem sisteminin önemli bir özelliği ise tüm gözlenen faktörlerin iki ayrı parça halindeki doğrusal fonksiyonlar ile bağlantılı olmasıdır ($\sum_{i=1}^{k} \alpha_i = 1$ ve $\sum_{i=1}^{k} \alpha_i Z_{ij}$ ile $\sum_{i=1}^{k} \beta_i = 1$ ve $\sum_{i=1}^{k} \beta_i Z_{ij}$).

Değeri Z_{im} ile gösterilmiş *m.* birimin G_1, G_2 ya da $G_1 \cap G_2$ (çakışma) durumlarından hangisine ait olduğu aşağıda verilen kriterle belirlenir.

a) $\sum_{i=1}^{k} \alpha_i^* Z_{im} > d^* \geq \sum_{i=1}^{k} \beta_i^* Z_{im}$ veya $\sum_{i=1}^{k} \alpha_i^* Z_{im} \leq d^* < \sum_{i=1}^{k} \beta_i^* Z_{im}$ ise *m.* birimin çakışma bölgesinde yer aldığı ve $G_1 \cap G_2$'ye ait olduğu sonucuna varılır.

b) $\sum_{i=1}^{k} \alpha_i^* Z_{im} > d^*$ ve $\sum_{i=1}^{k} \beta_i^* Z_{im} \geq d^*$ ($\sum_{i=1}^{k} \alpha_i^* Z_{im} = \sum_{i=1}^{k} \beta_i^* Z_{im} = d^*$ dahil) ise çakışma olmadığı ve *m.* birimin G_1 grubuna ait olduğu sonucuna varılır.

c) $\sum_{i=1}^{k} \alpha_i^* Z_{im} < d^*$ ve $\sum_{i=1}^{k} \beta_i^* Z_{im} < d^*$ ise *m.* birimin G_2 grubuna ait olduğu sonucuna varılır.

İlk aşamada ele alınan ve Eş. 3.6 modeli ile ifade edilen DEA-DA modeli G_1 ile G_2 arasında iki farklı ayırma fonksiyonu oluşturmaktadır. Birinci aşama ile son ayırma fonksiyonu belirlenmemekte, bir çakışma olup olmadığı tespit edilmektedir.

Aşama 2: Çakışmayı ele alma

Çakışmanın ($G_1 \cap G_2$) varlığı, bir birimin her iki gruba da dahil olabileceğini ifade eder.

$$Min \sum_{j \in G_1} S_{1j}^{+} + \sum_{j \in G_2} S_{2j}^{-}$$

Kısıtlar:

$$\sum_{i=1}^{k} \gamma_i Z_{ij} + S_{1j}^{+} - S_{1j}^{-} = c \qquad j \in G_1$$

$$\sum_{i=1}^{k} \gamma_i Z_{ij} + S_{2j}^{+} - S_{2j}^{-} = c - \eta \qquad j \in G_2 \tag{3.8}$$

$$\sum_{i=1}^{k} \gamma_i = 1$$

$\gamma_i, S_{1j}^{+}, S_{1j}^{-}, S_{2j}^{+}, S_{2j}^{-} \geq 0$, c işaretçe serbest

Eş. 3.8 modelinden γ_i^{*} elde edildikten sonra, $G_1 \bigcap G_2$'deki tüm birimler, $\sum_{i=1}^{k} \gamma_i^{*} Z_{ij}$ sınıflandırma skor değerleri ile ayırma skoru c^{*} karşılaştırılarak G_1 ya da G_2'den birine

$\sum_{i=1}^{k} \gamma_i^{*} Z_{im} \geq c^{*}$, $j \in G_1 \bigcap G_2$ ise G_1 grubuna

$\sum_{i=1}^{k} \gamma_i^{*} Z_{im} < c^{*}$, $j \in G_1 \bigcap G_2$ ise G_2 grubuna

eşitlikleri yardımıyla sınıflandırılır. Eş. 3.8 modeli, yalnızca çakışma ($G_1 \bigcap G_2$) özelliği olan birimlerin sınıflandırılmasında kullanılır. Ağırlık katsayıları DEA-DA modelinde bir yüzdelik ifadesi ile ölçülür. Böyle bir yüzdelik ifadesi ayırma fonksiyonu oluşturmada her bir faktörün katkı düzeyi ile ilgili bilgi sağlar.

Sueyoshi tarafından sunulan orijinal DEA-DA formülasyonu bazı eksikliklere sahiptir [Sueyoshi, 2001].

a) VZA-DA üç tane parçalı doğrusal ayırma fonksiyonu (iki tanesi çakışma belirlemede ve bir tanesi sınıflandırmada) ve iki tane ayırma skoru (d ve c)

üretmektedir. Ayırma fonksiyonlarının sayısının çokluğu hesaplama etkinliğini azaltmaktadır.

b) Üç tane parçalı doğrusal ayırma fonksiyonu şeklinde ifade edilen DEA-DA modeli sadece negatif olmayan ağırlık tahminleri bulacak şekilde formüle edilmiştir.

c) DEA-DA, birimlerden ölçülen değişkenlerde (Z_{ij}) negatif değerlere izin vermemektedir. Böyle bir eksiklik özellikle finans ve bankacılık gibi birçok değişkenin negatif değerlere sahip olduğu veri setlerinde modelin kullanılamayacağını göstermektedir.

Orijinal DEA-DA'ya ilişkin eksikliklerin üstesinden gelmek için, Sueyoshi genişletilmiş DEA-DA (Extended DEA-DA) modelini önermiştir [Sueyoshi, 2001]. Genişletilmiş DEA-DA modeli de, orijinal DEA-DA modeline benzer olarak çakışma durumunun belirlendiği ve sınıflandırmanın yapıldığı iki aşamadan oluşmaktadır.

Aşama 1: Çakışmayı belirleme ve sınıflandırma
Çakışma belirleme süreci veya genişletilmiş DEA-DA'nın birinci aşaması Eş. 3.9 modelinde olduğu gibi

$$Min \sum_{j\in G_1} S_{1j}^{+} + \sum_{j\in G_2} S_{2j}^{-}$$

Kısıtlar:

$$\sum_{i=1}^{k} \lambda_i Z_{ij} + S_{1j}^{+} - S_{1j}^{-} = d + 1 \qquad j \in G_1 \tag{3.9}$$

$$\sum_{i=1}^{k} \lambda_i Z_{ij} + S_{2j}^{+} - S_{2j}^{-} = d \qquad j \in G_2$$

$$\sum_{i=1}^{k} |\lambda_i| = 1$$

$S_{1j}^{+}, S_{1j}^{-}, S_{2j}^{+}, S_{2j}^{-} \geq 0$, λ, d işaretçe serbest.

formüle edilebilir. Eş. 3.9 modelinde S_{1j}^{+} ve S_{1j}^{-} sırasıyla G_1 grubuna ilişkin, d ayırma skorundan $\sum_{i=1}^{k}\alpha_i Z_{ij}$ parçalı doğrusal ayırma fonksiyonuna olan pozitif ve negatif sapmaları göstermektedir. Benzer olarak S_{2j}^{+} ve S_{2j}^{-}'de sırasıyla G_2 grubuna ilişkin, d ayırma skorundan $\sum_{i=1}^{k}\alpha_i Z_{ij}$ parçalı doğrusal ayırma fonksiyonuna olan pozitif ve negatif sapmaları göstermektedir. $S_{1j}^{+} > 0\ (j \in G_1)$ ve $S_{2j}^{-} > 0\ (j \in G_2)$ yanlış sınıflandırmanın varlığını göstermektedir.

Doğrusal programlamada mutlak değer ifadesi $(\sum_{i=1}^{k}|\lambda_i| = 1)$ doğrudan ele alınamayacağı için, model yeniden düzenlenmelidir. $\lambda_i^{+} = \left(|\lambda_i| + \lambda_i\right)/2$, $\lambda_i^{-} = \left(|\lambda_i| - \lambda_i\right)/2$ ve $\lambda_i = \lambda_i^{+} - \lambda_i^{-}$ olarak tanımlanır. λ_i^{+} ve λ_i^{-} üzerine yapılan tanımlamalar $\lambda_i^{+}\lambda_i^{-} = \left(\frac{|\lambda_i|^2 - \lambda_i^2}{4}\right) = 0$ şeklinde olduğundan, $\lambda_i^{+}\lambda_i^{-} = 0$ doğrusal olmayan şartı ve $\lambda_i^{+}, \lambda_i^{-} \geq 0$ eşitsizliği her zaman sağlanır. λ_i'nin λ_i^{+} ve λ_i^{-} olarak ayrılması Z_{ij}'de sadece pozitif değerlerin değil negatif değerlerin de olmasını mümkün kılmaktadır. Buradan, $\sum_{i=1}^{k}|\lambda_i| = 1$ kısıtı modelden çıkartılır ve yerine λ_i^{+} ve λ_i^{-}'ye bağlı tanımlamalar yazılırsa,

$$Min \sum_{j \in G_1} S_{1j}^{+} + \sum_{j \in G_2} S_{2j}^{-}$$

Kısıtlar:

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right)Z_{ij} + S_{1j}^{+} - S_{1j}^{-} = d + 1 \quad j \in G_1$$

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right)Z_{ij} + S_{2j}^{+} - S_{2j}^{-} = d \qquad j \in G_2 \tag{3.10}$$

$$\sum_{i=1}^{k}\left(\lambda_i^+ + \lambda_i^-\right) = 1$$

$S_{1j}^+, S_{1j}^-, S_{2j}^+, S_{2j}^-, \lambda_i^+, \lambda_i^- \geq 0$, d işaretçe serbest.

elde edilir. $\lambda_i^*\left(=\lambda_i^+ - \lambda_i^-\right)$ ve d^* Eş. 3.10 modelinin optimum çözümleri olsunlar. Değeri Z_{im} ile gösterilmiş m. birimin G_1, G_2 ya da $G_1 \bigcap G_2$ (çakışma) durumlarından hangisine ait olduğu aşağıda verilen kriterle belirlenir.

a) $\sum_{i=1}^{k}\lambda_i^* Z_{im} > d^* + 1$ ise m. birimin G_1 grubuna ait olduğu sonucuna varılır.

b) $d^* + 1 > \sum_{i=1}^{k}\lambda_i^* Z_{im} > d^*$ ise m. birimin $G_1 \bigcap G_2$ 'ye ait olduğu sonucuna varılır.

c) $d^* \geq \sum_{i=1}^{k}\lambda_i^* Z_{im}$ ise m. birimin G_2 grubuna ait olduğu sonucuna varılır.

Yukarıdaki kriterlere göre, tüm birimler ($G_1 \bigcup G_2$) aşağıda verilen alt kümelere ayrılabilir.

$$C_1 = \left\{ j \in G_1 \,\middle|\, \sum_{i=1}^{k}\lambda_i^* Z_{ij} > d^* + 1 \right\}$$

$$C_2 = \left\{ j \in G_2 \,\middle|\, \sum_{i=1}^{k}\lambda_i^* Z_{ij} \leq d^* \right\}$$

$D_1 = G_1 - C_1$ ve $D_2 = G_2 - C_2$

Aşama 2: Çakışmayı ele alma

Çakışmanın ($G_1 \bigcap G_2$) varlığı tanımlandıktan sonra, bir yeni ayırma skoru (c) modelin içine sadece $D_1 \bigcup D_2$ alt grubunda yer alan birimleri sınıflandıracak şekilde konulur.

$$Min \sum_{j\in D_1} S_{1j}^{+} + \sum_{j\in D_2} S_{2j}^{-}$$

Kısıtlar:

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} \geq d+1 \qquad j \in C_1$$

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} \leq d \qquad j \in C_2 \qquad (3.11)$$

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} + S_{1j}^{+} - S_{1j}^{-} = c \qquad j \in D_1$$

$$\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} + S_{2j}^{+} - S_{2j}^{-} = c \qquad j \in D_2$$

$$\sum_{i=1}^{k}\left(\lambda_i^{+} + \lambda_i^{-}\right) = 1$$

$$d \leq c \leq d+1$$

$S_{1j}^{+}, S_{1j}^{-}, S_{2j}^{+}, S_{2j}^{-}, \lambda_i^{+}, \lambda_i^{-} \geq 0$, c ve d işaretçe serbest

Eş. 3.11 modelinde ilk aşamada doğru sınıflandırılmış tüm birimler $\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} \geq d+1,\ j \in C_1$ ve $\sum_{i=1}^{k}\left(\lambda_i^{+} - \lambda_i^{-}\right) Z_{ij} \leq d,\ j \in C_2$ ile sınırlandırılmıştır. Bu kısıtlarla ilgili tüm sapma değişkenleri modelden çıkartılmıştır. $d+1$ (C_1 alt grubu için ayırma skoru) ve d (C_2 alt grubu için ayırma skoru) kısıtları altında, yeni c ayırma skoru, çakışma durumundaki gözlemlerin sapmalarının minimum yapılması ile belirlenir. c^{*} ve $\lambda_i^{*}\left(=\lambda_i^{+} - \lambda_i^{-}\right)$ sırasıyla modelden elde edilen ayırma skoru ve ağırlık katsayıları olsunlar. İlk aşamada çakışma grubunda tanımlanan *m.* birim, aşağıdaki kurala göre sınıflandırılabilir:

a) $\sum_{i=1}^{k} \lambda_i^{*} Z_{im} > c^{*}$ ise birimin G_1 grubuna ait olduğu sonucuna varılır.

b) $\sum_{i=1}^{k} \lambda_i^{*} Z_{im} < c^{*}$ ise birimin G_2 grubuna ait olduğu sonucuna varılır.

4. ÇOK GRUPLU MATEMATİKSEL PROGRAMLAMA YAKLAŞIMLARI

Bu bölümde çok gruplu sınıflandırma problemi için literatürde önerilen matematiksel programlama yöntemleri ayrıntılı bir şekilde incelenmekte ve literatürde önerilen yöntemlerin kuvvetli özelliklerine dayanan yeni bir, çok gruplu matematiksel programlama modeli önerisi yer almaktadır.

4.1. Literatürde Önerilen Çok Gruplu Matematiksel Programlama Modelleri

İki gruplu ayırma analizi problemi için önerilen çok sayıda matematiksel programlama yaklaşımı olmasına rağmen, çok az sayıda çok gruplu matematiksel programlama yaklaşımı önerilmiştir. İki gruplu durumdan çok gruplu duruma genişletmenin en kolay yolu tüm iki gruplu kombinasyonları kullanmaktır [Fred ve Glover, 1981b]. Herhangi bir iki gruplu formülasyon bu çok gruplu ayırma yönteminde kullanılabilir. İki gruplu durum için bir karma tamsayılı sınıflandırma modeli,

$$Min \sum_{j=1}^{n} y_j$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i Z_{ij} - My_j \le c - \varepsilon \qquad , \ j \in G_1 \tag{4.1}$$

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + My_j \ge c + \varepsilon \qquad , \ j \in G_2$$

olarak verilmektedir [Pavur ve Loucopoulos, 1995]. Burada, Z_{ij}, j. birimin i. değişkenine ilişkin gözlem değeri, α_i, i. değişkenin ağırlığı (katsayısı), y_j, j. birimin yanlış sınıflandırılıp sınıflandırılmadığını gösteren iki değerli bir değişken, c iki grubu birbirinden ayıran ayırma skoru ve c serbest bir değişkendir. ε ise küçük bir pozitif sayıdır ve herhangi bir birimin ayırma fonksiyonu üzerinde olmaması için modele dahil edilmektedir. h grup sayısı olmak üzere, G_1, G_2, . . . , G_h gruplarının

tüm ikili çiftlerinin $h(h-1)/2$ tane ayırma fonksiyonu kurmak için kullanılması gerekmektedir.

Daha geçerli bir yol ise Gehrlein (1986) tarafından geliştirilen genel tek fonksiyonlu sınıflandırma modelini ya da genel çok fonksiyonlu sınıflandırma modelini kullanmaktır.

Gehrlein (1986) ikiden fazla grup için tasarlanmış özel bir karma tamsayılı programlama modeli önermiştir. Genel tek fonksiyonlu sınıflandırma modeli (GSFC),

$$Min \sum_{j=1}^{n} y_j$$

Kısıtlar:

$$\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} - My_j \le U_r \qquad , \forall j \in G_r ,\ r = 1,\ 2, \ldots , h \tag{4.2}$$

$$\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} + My_j \ge L_r \qquad , \forall j \in G_r ,\ r = 1,\ 2, \ldots , h$$

$$U_r - L_r \ge e' \qquad , r = 1,\ 2, \ldots , h$$

$$L_r - U_t + MJ_{rt} \ge e \qquad , \forall G_r ,\ \forall G_t ,\ r \ne t ,\ r = 1,\ 2, \ldots , h$$

$$L_t - U_r + MJ_{tr} \ge e \qquad , \forall G_r ,\ \forall G_t ,\ r \ne t ,\ r = 1,\ 2, \ldots , h$$

$$J_{rt} + J_{tr} = 1 \qquad , r = 1,\ 2, \ldots , h ;\ r \ne t$$

olarak tanımlanmaktadır. Burada, h; grup sayısı, α_0; bir kayma sabiti (bir eşik değeri), U_r; G_r grubuna atanan aralığın son üst noktası ($r = 1, \ldots , h$), L_r; G_r grubuna atanan aralığın son alt noktası ($r = 1, \ldots , h$), e'; bir gruba atanan aralığın minimum genişliği, e; bitişik aralıklar arasındaki aralığın (oluşacak boşluğun) minimum büyüklüğü, M; pozitif büyük bir sabit, k; değişken sayısı ve n; toplam birim sayısı olarak tanımlanmaktadır. J_{rt} değişkeni ise

$$J_{rt} = \begin{cases} 1, & G_r \text{ grubu } G_t \text{ grubundan önce ise} \\ 0, & \text{diğer durumlarda} \end{cases}$$

olarak tanımlanmaktadır.

Bu modelde, y_j $(j = 1, 2, \ldots, n)$ ve J_{rt} $(r, t = 1, 2, \ldots, h;\ r \neq t)$ değişkenleri ikili değerler alan değişkenler α_i $(i = 1, 2, \ldots, k)$, L_r ve U_r $(r = 1, 2, \ldots, h)$ değişkenleri ise serbest değişkenlerdir.

Eş. 4.2 ile verilen model, her bir birim için bir doğrusal $\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij}$ ayırma skoru tanımlamakta ve böyle bir ayırma skorunun $[L_r, U_r]$ aralığının içine düşüp düşmediğini kontrol etmektedir. Bu işlem $\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} - My_j \leq U_r$ ve $\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} + My_j \geq L_r$ kısıtları ile yapılmaktadır. $U_r - L_r \geq e'$ kısıtı her bir grubun e' ya da daha fazla genişlikteki aralıklara atanmasını sağlamaktadır. $L_r - U_t + MJ_{rt} \geq e$, $L_t - U_r + MJ_{tr} \geq e$ ve $J_{rt} + J_{tr} = 1$ kısıtları ise grup aralıkları arasında çakışma olmamasını garanti etmektedir. e değeri bitişik gruplar arasındaki minimum boşluk (aralık) olarak tanımlanmaktadır ve küçük pozitif bir sayı olduğu varsayılmaktadır. Ayırma skor değeri gruplar arasındaki boşluğa düşen bir birim yanlış sınıflandırılmış sayılmaktadır [Pavur ve Loucopoulos, 1995].

Kısıtlardaki M sayısı yanlış sınıflandırılmış bir birimin kendi grubuna ilişkin aralığın en yakın uç noktasına olan maksimum sapmayı tanımlamaktadır. $L_r - U_t + MJ_{rt} \geq e$ ve $L_t - U_r + MJ_{tr} \geq e$ kısıtlarındaki M sayısı her bir gruba ilişkin aralıkların oluşturulmasında önemli bir rol oynamaktadır. $J_{rt} + J_{tr} = 1$ kısıtından $r \neq t$ olmak üzere herhangi iki grup için ya J_{rt} ya da J_{tr} 1 olmak zorundadır, yani J_{rt} ve J_{tr} aynı anda değere sahip olamazlar. J_{rt}'nin 1 olduğu varsayılsın. Bu durumda $L_r - U_t + M \geq e$ yani $M \geq U_t - L_r + e$ olacaktır. Eğer U_t aralıkların en

büyük uç noktası ve L_r aralıkların en küçük uç noktası ise, M değeri başka bir yoruma da sahip olmaktadır. M en sağda olan aralığın uç noktası ile en soldaki aralığın en uç noktası arasındaki mümkün maksimum farkı tanımlamaktadır.

Eş. 4.2 ile verilen model birimlerin mutlaka bir gruba sınıflandırılacağını garanti etmemektedir. Bir birime ait ayırma skoru en sağdaki aralığın da sağında bir değere sahip olabilir. Bu durum veride aşırı derecede aykırı değer söz konusu olduğu zaman oluşmaktadır.

Gehrlein (1986) ve aynı zamanlarda Choo ve Wedley (1985) bir genel çok fonksiyonlu sınıflandırma modeli (GMFC) önermişlerdir.

$$Min \sum_{j=1}^{n} y_j$$

Kısıtlar:

$$\alpha_{r0} + \sum_{i=1}^{k} \alpha_{ri} Z_{ij} - \alpha_{t0} - \sum_{i=1}^{k} \alpha_{ti} Z_{ij} + My_j \geq e \qquad , \forall j \in G_r ,\ j = 1,\ 2, \ldots, n \qquad (4.3)$$

$$r = 1,\ 2, \ldots, h ,\ r \neq t$$

Bu modelde, α_{ri}; G_r grubundaki Z_{ij} değişkeninin ağırlığı ve α_{r0}: G_r grubu için kayma sabiti (G_r grubu için bir eşik değeri) olarak tanımlanmaktadır. Burada, α_{ri} ağırlık değişkenleri işaretçe serbest değişkenlerdir. y_j ise GSFC modelinde tanımlandığı gibidir.

GMFC modeli bir birimi en büyük ayırma skoruna sahip grup içine sınıflandırmaktadır. Modeldeki kısıt ile her bir grup için bir bireysel ayırma fonksiyonu oluşturulmaktadır. Eğer grup sayısı ve değişken sayısı çok büyürse, GMFC modeli GSFC modelinden çok daha fazla serbest değişken ve kısıtlamaya sahip olacaktır [Loucopoulos ve Pavur, 1997].

GSFC modeli yanlış sınıflandırılmış birimlerin sayısının minimize edildiği bir amaç fonksiyonuna sahiptir. Amaç fonksiyonu yanlış sınıflandırma sapmalar toplamının minimize edilmesi ile yer değiştirilirse, her bir y_j ikili değişkeni d_{ju} ve d_{jl} negatif olmayan sürekli değişken çifti ile yer değiştirecektir. Böylelikle modelin hesaplama karmaşıklığı ve zorluğu anlamlı bir şekilde azalacaktır. Bu model Eş. 4.4 ile sunulmaktadır ve sapmalar toplamının minimize edilmesi için genel tek fonksiyonlu sınıflandırma modeli

$$Min \sum_{j=1}^{n} \left(d_{ju} + d_{jl} \right)$$

Kısıtlar:

$$\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} - d_{ju} \le U_r \qquad , \forall j \in G_r ,\ r = 1,\ 2, \ldots , h \tag{4.4}$$

$$\alpha_0 + \sum_{i=1}^{k} \alpha_i Z_{ij} + d_{jl} \ge L_r \qquad , \forall j \in G_r ,\ r = 1,\ 2, \ldots , h$$

$$U_r - L_r \ge e' \qquad , r = 1,\ 2, \ldots , h$$

$$L_r - U_t + M J_{rt} \ge e \qquad , \forall G_r ,\ \forall G_t ,\ r \ne t ,\ r = 1,\ 2, \ldots , h$$

$$L_t - U_r + M J_{tr} \ge e \qquad , \forall G_r ,\ \forall G_t ,\ r \ne t ,\ r = 1,\ 2, \ldots , h$$

$$J_{rt} + J_{tr} = 1 \qquad , r = 1,\ 2, \ldots , h ;\ r \ne t$$

olarak verilmektedir [Pavur ve Loucopoulos, 1995]. Eş. 4.4 modelinde, d_{ju} ; U_r uç noktasının sağ tarafına yanlış sınıflandırılmış bir birimin U_r uç değerinden sapması ve d_{jl}; L_r alt noktasının sol tarafına yanlış sınıflandırılmış bir birimin L_r alt noktasından sapması olarak tanımlanmaktadır. Burada, d_{ju} ve d_{jl} sapma değişkenleri negatif olmayan değişkenlerdir. J_{rt}, J_{tr}, α_0, α_i, L_r ve U_r değişkenleri ise GSFC modelinde tanımlandığı gibidir.

GSFMSD modeli gruplara aralık atamaktadır ve yalnızca bir tane ayırma fonksiyonu kullanmaktadır. Ayrıca yanlış sınıflandırma uzaklığını kısıtlamak (sınırlamak) için

büyük M sayısına gerek duyulmamakla birlikte aralıkların bütünlüğünü (oluşmasını) sağlamak için büyük M sayısı hala kullanılmaktadır.

Bal (1999) doğrusal hedef programlama ile üç gruplu sınıflandırma problemini ele almıştır. Grupları bilinen k değişkenli birim olduğu varsayılmakta ve her birimin skorlarını uygun grup aralıkları içine yerleştirecek ayırma fonksiyonunun ağırlıklarının bulunması amaçlanmaktadır. Bu düşünce daha açık olarak α_i, i. değişkenin ağırlığı, Z_{ij}, j. birimin i. değişken değeri olmak üzere, Z_{ij} değerleri ve G_r $(r = 1,\ 2,\ 3)$ grupları biliniyorken $S_j = \sum_{i=1}^{k} \alpha_i Z_{ij}$ birim skorlarını uygun gruplara ayıracak, gruplara ait L_r, U_r alt ve üst sınırları ile α_i ayırıcı ağırlığının bulunmasına dayanmaktadır. Buna göre üç gruplu ($h = 3$) matematiksel programlama modeli,

$$L_r \le \sum_{i=1}^{k} \alpha_i Z_{ij} \le U_r,\ j \in G_r,\ r = 1,\ 2,\ 3$$

$$L_1 \le U_1 \le L_2 \le U_2 \le L_3 \le U_3$$

kısıtlarını sağlayacak L_r, U_r ve α_i değerlerinin araştırılmasından ibaret olur.

Doğrusal hedef programlaması ile sınıflama probleminde $S_j = \sum_{i=1}^{k} \alpha_i Z_{ij}$ skorlarından grupları ayırmaya yarayan L_r ve U_r grup sınırları hedef değerleri olarak alınarak bu değerlerden sapmaları minimize eden çözümler elde edilebilir. Her birim bulunduğu grup aralığı içinde aşağıda verildiği gibi skorlanmaktadır.

$$L_1 \le \sum_{i=1}^{k} \alpha_i Z_{ij} \le U_1,\ j \in G_1$$

$$L_2 \le \sum_{i=1}^{k} \alpha_i Z_{ij} \le U_2,\ j \in G_2$$

$$L_3 \le \sum_{i=1}^{k} \alpha_i Z_{ij} \le U_3 ,\ j \in G_3$$

Hedef değerlerine uygun sapma değişkenlerinin konulmasıyla doğrusal hedef programlamaya dayalı sınıflandırma modeli

$$Min\ a = \sum_{j=1}^{n} \left(n_j^L + p_j^L \right)$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + n_j^L - p_j^L = L_r ,\ j \in G_r ,\ r = 1,\ 2,\ 3 \qquad (4.5)$$

$$\sum_{i=1}^{k} \alpha_i Z_{ij} + n_j^U - p_j^U = U_r ,\ j \in G_r ,\ r = 1,\ 2,\ 3$$

elde edilmiş olur. Eş. 4.5 modelinde, α_i $(i = 1,\ \ldots,\ k)$, L_r ve U_r $(r = 1,\ 2,\ 3)$ serbest değişkenler, n_j^L, p_j^L, n_j^U ve p_j^U sapmaları ise negatif olmayan değişkenlerdir. Modelde değişkenlerin ayırıcı ağırlığı ile grupların alt ve üst sınır değerleri aynı anda bulunmaktadır. Gruplara ait sınırlar $U_r + \varepsilon \le L_{r+1},\ r = 1, 2;\ \varepsilon \ge 1$ kısıtlarının kullanılmasıyla birbirinden tam olarak ayrılabilmektedir. Bal (1999) tarafından verilen üç gruplu doğrusal hedef programlama modeli h grup sayısı $(r = 1,\ \ldots,\ h)$ olmak üzere $h > 3$ için de kolaylıkla genişletilebilir.

Lam ve Moy (1996), Lam ve ark. (1996) tarafından geliştirilen iki gruplu sınıflandırma modelini çok gruplu duruma genişletmişlerdir. Lam ve ark. (1996)'nın iki gruplu sınıflandırma modelinde sınıflandırma işlemi bireysel sınıflandırma skorlarının kendi grup ortalama skorlarından sapmalarının minimize edilmesi temeline dayanmaktadır. MLM olarak isimlendirilen modelin çok gruplu duruma genişletme işlemi ise aşağıda verilmektedir.

i. değişkene ilişkin ortalama

$$\overline{z}_i^r = \frac{1}{n_r}\sum_{j\in G_r} Z_{ij},\ r = 1,\ \ldots,\ h$$

olarak tanımlanır. Burada n_r, G_r grubundaki birim sayısıdır. n ise $n = n_1 + \ \ldots\ + n_h$ olmak üzere toplam birim sayısını ifade etmektedir. $u = 1,\ \ldots,\ h-1$, $v = u+1,\ \ldots,\ h$ olmak üzere, her bir (u,v) çifti için GP olarak isimlendirilen model

GP

$$Min \sum_{j\in G_u,\, j\in G_v}^{n} d_j$$

Kısıtlar:

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \overline{z}_i^u\right) + d_j \geq 0,\ j \in G_u \qquad (4.6)$$

$$\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \overline{z}_i^v\right) - d_j \leq 0,\ j \in G_v$$

$$\sum_{i=1}^{k} \alpha_i \left(\overline{z}_i^u - \overline{z}_i^v\right) \geq 1$$

şeklinde tanımlanmaktadır. Eş. 4.6 modelinde, α_i $(i = 1,\ \ldots,\ k)$ serbest değişkenler ve $j \in G_u$ ve $j \in G_v$ için $d_j \geq 0$'dır. $\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \overline{z}_i^u\right) + d_j \geq 0$ ve $\sum_{i=1}^{k} \alpha_i \left(Z_{ij} - \overline{z}_i^v\right) - d_j \leq 0$ kısıtları ile $r.$ gruptaki birimlerin sınıflandırma skorlarını $r.$ grubun ortalama sınıflandırma skoruna mümkün olduğunca yaklaştırmaya yarar $(r = u, v)$.

Eş. 4.6 modeli yardımıyla her (u,v) grup çifti için elde edilen α_i yardımıyla G_u ve G_v'deki bireylerin S_j sınıflandırma skor değerleri bulunur. Ardından da $u = 1,\ \ldots,\ h-1$ ve $v = u+1,\ \ldots,\ h$ olmak üzere grupları ayırmada yararlanılacak c_{uv} değerleri GP1 olarak isimlendirilen

GP1

$$Min \sum_{j \in G_u} \sum_{u=1}^{h-1} \sum_{v=u+1}^{h} d_{juv} + \sum_{j \in G_v} \sum_{u=1}^{h-1} \sum_{v=u+1}^{h} d_{juv}$$

Kısıtlar:

$$S_{juv} + d_{juv} \geq c_{uv},\ u = 1,\ \ldots,\ h-1,\ v = u+1,\ \ldots,\ h,\ j \in G_u \tag{4.7}$$

$$S_{juv} - d_{juv} \leq c_{uv},\ u = 1,\ \ldots,\ h-1,\ v = u+1,\ \ldots,\ h,\ j \in G_v$$

modelin çözülmesiyle elde edilir. Burada, tüm c_{uv} değerleri serbest değişkenler ve tüm d_{juv} değerleri ise negatif olmayan değişkenler olarak tanımlanmıştır. Eş. 4.7 ile verilen GP1 modelinde tüm kesme değerleri yani c_{uv} değerleri eş zamanlı olarak elde edilmektedir.

$h(h-1)/2$ sayıdaki problemlerden oluşan GP ve GP1 modelleri, GP'nin sapma değişkenlerinin minimize edilmesine birinci öncelik verilerek öncelikli hedef programlaması ile çözülerek çok gruplu sınıflandırma yapılabilir. Ayrıca yanlış sınıflandırılmış birimlerin sayısını minimize etmek için doğrusal programlama modelindeki (GP1) amaç fonksiyonu değiştirilirse, Eş. 4.8 ile verilen ve GP2 olarak isimlendirilen model elde edilmiş olur.

GP2

$$Min \sum_{j \in G_u} \sum_{u=1}^{h-1} \sum_{v=u+1}^{h} y_{juv} + \sum_{j \in G_v} \sum_{u=1}^{h-1} \sum_{v=u+1}^{h} y_{juv}$$

Kısıtlar:

$$S_{juv} + My_{juv} \geq c_{uv},\ u = 1,\ \ldots,\ h-1,\ v = u+1,\ \ldots,\ h,\ j \in G_u \tag{4.8}$$

$$S_{juv} - My_{juv} \leq c_{uv},\ u = 1,\ \ldots,\ h-1,\ v = u+1,\ \ldots,\ h,\ j \in G_v$$

Eş. 4.8 modelinde, tüm c_{uv} değerleri serbest değişkenler, tüm y_{juv} değerleri 0 ya da 1 değerlerini alabilen ikili değişkenler ve M ise pozitif büyük bir sayıdır. $y_{juv} = 1$ ise

j. birimin ait olduğu gruba atanmayacağını, $y_{juv} = 0$ ise ait olduğu gruba atanacağını gösterir.

Gochet ve ark. (1997) çok gruplu sınıflandırma problemi için LP^q yaklaşımını önerdiler.

Her bir birim yalnız ve yalnızca tek bir gruba ait olmak üzere, birimlerin gruplarının bir sonlu $S = \{1, \ldots, s\}$ kümesi göz önüne alınsın. Her bir gruptaki birim sayısı n_r, $r \in S$ olsun ve her bir birimin ait olduğu grup başlangıçta bilinsin. r. gruba ($r \in S$) ait olan örnek birimlerinin bir kümesi $P_r = \{1, \ldots, n_r\}$ olsun. Her bir birim k tane değişkenin içerildiği $(k+1)$ boyutlu bir $Z_{i0} = 1$ olmak üzere $Z_i = (Z_{i0}, Z_{i1}, \ldots, Z_{ik})$ sütun vektörüne sahiptir. r. gruptaki her birim için ($j \in P_r$) değişken vektörü $Z_{jr0} = 1$ olmak üzere $Z_{jr} = (Z_{jr0}, Z_{jr1}, \ldots, Z_{jrk})$ ile gösterilsin.

$s(k+1)$ boyutlu $\alpha^r = (\alpha_{r0}, \alpha_{r1}, \ldots, \alpha_{rk})$ satır vektörünün tahmin edilmesi istenmektedir ve r grubunda yer alan her bir j birimi için $\alpha^r z_r$ $(r = 1, \ldots, s)$ doğrusal ayırma skoru aranmaktadır. h. gruptaki j. birim için sınıflandırma kuralı

$$\alpha^h z_r = \max_{r \in S} \{\alpha^r z_j\} \tag{4.9}$$

ile belirlenmektedir. Eş. 4.9 kuralı bir birimi en büyük sınıflandırma skoruna sahip olan gruba atar. α^r, $r \in S$ vektörleri Eş. 4.9'daki kuralı optimal yapacak şekilde belirlenmektedir. Bu işlem örnek birimler üzerinde "uyumluluk (goodness)" ve "uyumsuzluk (badness)" ölçülerinin birleştirilmesiyle tanımlanacak olan kritere göre yapılacaktır. $S_{\sim r} = S / \{r\}$, r. grup haricindeki tüm grupların kümesini göstersin. $y^+ = \max(0, y)$, $y^- = \min(0, y)$ ve $y \in \Re$ olmak üzere $y = y^+ - y^-$ olarak ifade edilsin.

Herhangi bir $j \in P_r$, ($r \in S$) birimi için eğitim setindeki uyumluluk Eş. 4.10 ile verilen $G_{rt}\left(\alpha^r, \alpha^t\right)$ ile ölçülmektedir. Eş. 4.10'da, j. birimin kendi r. grubuna göre $\alpha^r Z_{jr}$ ayırma skoru, j. birimin $t \in S_{\sim}$ kalan gruplarına göre $\alpha^t Z_{jr}$ ayırma skorlarının ikili olarak karşılaştırılmasıdır.

$$G_{rt}\left(\alpha^r, \alpha^t\right) = \left(\alpha^r Z_{ir} - \alpha^t Z_{ir}\right)^+, \; j \in P_r, \; t \in S_{\sim r}, \; r \in S \tag{4.10}$$

$G_{rt}\left(\alpha^r, \alpha^t\right)$'nin kesin pozitif değerleri tercih edilmektedir ve büyük değerleri daha iyidir. Benzer olarak, t grubuna göre $j \in P_r$ biriminin uyumsuzluğu Eş. 4.11'de verildiği gibi tanımlanmaktadır.

$$B_{rt}\left(\alpha^r, \alpha^t\right) = \left(\alpha^r Z_{ir} - \alpha^t Z_{ir}\right)^-, \; j \in P_r, \; t \in S_{\sim r}, \; r \in S \tag{4.11}$$

$B_{rt}\left(\alpha^r, \alpha^t\right)$'nin küçük değerleri tercih edilir, ideal olanı sıfırdır. $j \in P_r$ birimi için uyum ve uyumsuzluğun birleştirilmesi $G_r^j(\alpha)$ ve $B_r^j(\alpha)$ olarak, sırasıyla Eş. 4.12 ve 4.13'de verilmektedir.

$$G_r^j(\alpha) = G_r^j\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{t \in S_{\sim r}} G_{rt}^j\left(\alpha^r, \alpha^t\right), \; j \in P_r, \; r \in S \tag{4.12}$$

$$B_r^j(\alpha) = G_r^j\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{t \in S_{\sim r}} B_{rt}^j\left(\alpha^r, \alpha^t\right), \; i \in P_r, \; r \in S \tag{4.13}$$

r. gruptaki tüm birimler için uyum ve uyumsuzluk $G_r(\alpha)$ ve $B_r(\alpha)$ olarak, sırasıyla Eş. 4.14 ve Eş. 4.15'de verilmektedir.

$$G_r(\alpha) = G_r\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{j \in P_r} G_r^j\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{j \in P_r} \sum_{t \in S_{\sim r}} G_{rt}^j\left(\alpha^r, \alpha^t\right), \; r \in S \tag{4.14}$$

$$B_r(\alpha) = B_r\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{j \in P_r} B_r^j\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{j \in P_r} \sum_{t \in S_{\sim r}} B_{rt}^j\left(\alpha^r, \alpha^t\right), \; r \in S \tag{4.15}$$

Tüm gruplar için toplam uyumluluk ve uyumsuzluk ölçüleri sırasıyla Eş. 4.16 ve Eş. 4.17'de verilmektedir.

$$G(\alpha) = G\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{r \in S} G_r(\alpha) = \sum_{r \in S} \sum_{j \in P_r} \sum_{t \in S_{\sim r}} G_{rt}^j\left(\alpha^r, \alpha^t\right) \tag{4.16}$$

$$B(\alpha) = B\left(\alpha^1, \ldots, \alpha^s\right) = \sum_{r \in S} B_r(\alpha) = \sum_{r \in S} \sum_{j \in P_r} \sum_{t \in S_{\sim r}} B_{rt}^j\left(\alpha^r, \alpha^t\right) \tag{4.17}$$

Sırasıyla Eş. 4.16 ve 4.17 ile verilen toplam uyum ve uyumsuzluk ölçüleri kavramsal olarak iki gruplu durum için çoğu araştırmacı tarafından verilen iç ve dış sapma değişkenleri ile benzer yapıdadır [Östermark ve Höglund, 1998].

$G(\alpha) = B(\alpha) = 0$ elde edilmesi, sınıflama hakkında herhangi bir kullanışlı bilgi vermez yani bu durumda bütün birimler gruplardan herhangi birine sınıflandırılmaktadır. Uygun olmayan çözümlerle karşılaşmamak için uygun bir normalleştirme dönüşümüne ihtiyaç vardır. $G(\alpha) - B(\alpha) < 0$ olması yani toplam uyumsuzluğun toplam uyumluluktan daha büyük olduğu durumlar genellikle başarılı olmayan sonuçlara yol açacağı beklenilen bir durumdur. Buradan, arzu edilmeyen sonuçlardan kaçınmak için, $q > 0$ olmak üzere

$$G(\alpha) - B(\alpha) = q \tag{4.18}$$

normalleştirme dönüşümü önerilmiştir. Toplam uyumsuzluğun β_{rt}^j, toplam uyumluluğun ise γ_{rt}^j değişkenleri ile temsil edildiğini varsayalım. Toplam uyumsuzluğun minimize yapılmaya çalışıldığı LP^q modeli Eş. 4.19 ile verilmektedir.

$$Min \sum_{r \in S} \sum_{t \in S_{\sim r}} \sum_{j \in P_r} \beta_{rt}^j$$

Kısıtlar:

$$\beta_{rt}^j + \left(\alpha^r - \alpha^t\right) Z_{jr} - \gamma_{rt}^j = \varepsilon, \quad \forall j \in P_r,\ t \in S_{\sim r},\ r \in S \tag{4.19}$$

$$\sum_{r \in S} \sum_{t \in S_{\sim r}} \sum_{j \in P_r} \left(\gamma_{rt}^j - \beta_{rt}^j\right) = q$$

$$\beta_{rt}^j,\ \gamma_{rt}^j \geq 0, \qquad \forall j \in P_r,\ t \in S_{\sim r},\ r \in S$$

α^r, α^t serbest değişkenler.

Sueyoshi (2001) geliştirdiği DEA-DA modelini karma-tamsayılı matematiksel programlama yaklaşımların da genişletmiştir ve çok gruplu durum için önerdiği modeli de karma-tamsayılı modele dayanmaktadır. Sueyoshi'nin iki gruplu durum için önerdiği karma-tamsayılı modeli Eş. 4.20 ile verilmektedir.

$$Min \sum_{j \in G_1} y_j + \sum_{j \in G_2} y_j$$

Kısıtlar:

$$\sum_{i=1}^{k} \lambda_i Z_{ij} - c + My_j \geq 0 \qquad , \ j \in G_1 \tag{4.20}$$

$$\sum_{i=1}^{k} \lambda_i Z_{ij} - c - My_j \leq -\varepsilon \qquad , \ j \in G_2$$

$$\sum_{i=1}^{k} |\lambda_i| = 1$$

Eş. 4.20 modelinde, λ_i $(i = 1, \ldots, k)$ ve c serbest değişkenler, y_j $(j = 1, \ldots, n)$ ise 0 ve 1 değerlerini alabilen ikili değişkenlerdir. $y_j = 1$ olması $j.$ birimin yanlış sınıflandırıldığını göstermektedir. Eş. 4.20 modelindeki M büyük bir pozitif sayıyı ε ise küçük bir pozitif sayıyı simgelemektedir. Amaç fonksiyonu y_j ikili değişkenleri sayesinde yanlış sınıflandırılmış birimlerin toplam sayısını minimize edecek şekilde tasarlanmıştır. Ayırma skoru G_1 grubu için c ile G_2 grubu için ise $c - \varepsilon$ ile ifade edilmektedir. ε sayısı herhangi bir birimin ayırma fonksiyonu üzerinde olmasını engellemek için konulmaktadır. $i.$ değişken için ağırlık tahmini (katsayısı) λ_i ile ifade edilmektedir ve tüm Z_{ij} gözlemleri $\sum \lambda_i Z_{ij}$ ayırma fonksiyonu ile bağlantılıdır. Ayrıca ağırlık değerlerinin mutlak değerlerinin toplamı 1'eşitlenmektedir. Bu durum veride negatif değerlere de izin verilmesini sağlamaktadır.

$\sum_{i=1}^{k}|\lambda_i|=1$ kısıtı nedeniyle Eş. 4.20 modeli doğrudan çözülememektedir. λ_i ağırlıkları $\lambda_i^+=\left(|\lambda_i|+\lambda_i\right)/2$ ve $\lambda_i^-=\left(|\lambda_i|-\lambda_i\right)/2$ olmak üzere, $\lambda_i=\lambda_i^+-\lambda_i^-$ olarak tanımlanırsa Eş. 4.21 ile verilen model elde edilir.

$$Min \sum_{j\in G_1} y_j + \sum_{j\in G_2} y_j$$

Kısıtlar:

$$\sum_{i=1}^{k}\left(\lambda_i^+-\lambda_i^-\right)Z_{ij}-c+My_j\geq 0 \qquad , \ j\in G_1 \tag{4.21}$$

$$\sum_{i=1}^{k}\left(\lambda_i^+-\lambda_i^-\right)Z_{ij}-c-My_j\leq -\varepsilon \qquad , \ j\in G_2$$

$$\sum_{i=1}^{k}\left(\lambda_i^++\lambda_i^-\right)=1$$

$$\xi_i^+\geq\lambda_i^+\geq\varepsilon\xi_i^+ \qquad i=1,\ 2,\ \ldots,\ k$$

$$\xi_i^-\geq\lambda_i^-\geq\varepsilon\xi_i^- \qquad i=1,\ 2,\ \ldots,\ k$$

$$\xi_i^++\xi_i^-\leq 1 \qquad i=1,\ 2,\ \ldots,\ k$$

$$\sum_{i=1}^{k}\left(\xi_i^++\xi_i^-\right)=k$$

Eş. 4.21 modelinde, λ_i^+ ve λ_i^- $(i=1,\ \ldots,\ k)$ negatif olmayan değişkenler, c serbest değişken, y_j $(j=1,\ \ldots,\ n)$ ve ξ_i^+ ve ξ_i^- $(i=1,\ \ldots,\ k)$ ise 0 ve 1 değerlerini alabilen ikili değişkenlerdir. $\xi_i^++\xi_i^-\leq 1$, $\xi_i^-\geq\lambda_i^-\geq\varepsilon\xi_i^-$ ve $\xi_i^-\geq\lambda_i^-\geq\varepsilon\xi_i^-$ kısıtları λ_i^+ ve λ_i^- için alt ve üst sınırlar oluşturmakta ve ayrıca $\lambda_i^+\lambda_i^-=0$ olması sağlanmaktadır. Eş. 4.21 modeli için en önemli eleştiri M ve ε sayılarının seçimidir.

Sueyoshi (2006), Eş. 4.21 ile verilen karma-tamsayılı modelin çok gruplu duruma genişlemesini önce üç gruplu durum için vermiştir.

$$Min \sum_{j\in G_1} y_j + \sum_{j\in G_2} y_j + \sum_{j\in G_3} y_j$$

Kısıtlar:

$$\sum_{i=1}^{k}\left(\lambda_i^+ - \lambda_i^-\right) Z_{ij} - c_1 + My_j \geq 0 \qquad , \ j \in G_1 \tag{4.22}$$

$$\sum_{i=1}^{k}\left(\lambda_i^+ - \lambda_i^-\right) Z_{ij} - c_1 - My_j \leq -\varepsilon \qquad , \ j \in G_2$$

$$\sum_{i=1}^{k}\left(\lambda_i^+ - \lambda_i^-\right) Z_{ij} - c_2 + My_j \geq 0 \qquad , \ j \in G_2$$

$$\sum_{i=1}^{k}\left(\lambda_i^+ - \lambda_i^-\right) Z_{ij} - c_2 - My_j \leq -\varepsilon \qquad , \ j \in G_3$$

$$\sum_{i=1}^{k}\left(\lambda_i^+ + \lambda_i^-\right) = 1$$

$$\xi_i^+ \geq \lambda_i^+ \geq \varepsilon\xi_i^+ \qquad i = 1,\ 2,\ \ldots,\ k$$

$$\xi_i^- \geq \lambda_i^- \geq \varepsilon\xi_i^- \qquad i = 1,\ 2,\ \ldots,\ k$$

$$\xi_i^+ + \xi_i^- \leq 1 \qquad i = 1,\ 2,\ \ldots,\ k$$

$$\sum_{i=1}^{k}\left(\xi_i^+ + \xi_i^-\right) = k$$

Eş. 4.22 modelinde, λ_i^+ ve λ_i^- $(i = 1,\ \ldots,\ k)$ negatif olmayan değişkenler c_1 ve c_2 serbest değişkenler y_j $(j = 1,\ \ldots,\ n)$ ve ξ_i^+ ve ξ_i^- $(i = 1,\ \ldots,\ k)$ ise 0 ve 1 değerlerini alabilen ikili değişkenlerdir. Modelde c_1 ayırma skoru G_1 ve G_2 gruplarını ayırmak için, c_2 ayırma skoru ise G_2 ve G_3 gruplarını ayırmak için kullanılmaktadır. Birimler aşağıda verilen kurala göre sınıflandırılmaktadır.

$\sum_{i=1}^{k} \lambda_i^* Z_{im} \geq c_1^*$ ise G_1 grubuna

$c_1^* - \varepsilon \geq \sum_{i=1}^{k} \lambda_i^* Z_{im} \geq c_2^*$ ise G_2 grubuna

$c_2^* - \varepsilon \geq \sum_{i=1}^{k} \lambda_i^* Z_{im}$ ise G_3 grubuna

sınıflandırma yapılmaktadır. Çok gruplu durum için ise genel bir model $r = 1, \ldots, h$ için Eş. 4.23 ile verilmektedir.

$$Min \sum_{r=1}^{h} \sum_{j \in G_r} y_j$$

Kısıtlar:

$$\sum_{i=1}^{k} \left(\lambda_i^+ - \lambda_i^- \right) Z_{ij} - c_r + M y_j \geq 0 \qquad , \; j \in G_r \,, \; r = 1, \ldots, h-1 \qquad (4.23)$$

$$\sum_{i=1}^{k} \left(\lambda_i^+ - \lambda_i^- \right) Z_{ij} - c_r - M y_j \leq -\varepsilon \qquad , \; j \in G_{r+1} \,, \; r = 1, \ldots, h-1$$

$$\sum_{i=1}^{k} \left(\lambda_i^+ + \lambda_i^- \right) = 1$$

$$\xi_i^+ \geq \lambda_i^+ \geq \varepsilon \xi_i^+ \qquad i = 1, 2, \ldots, k$$

$$\xi_i^- \geq \lambda_i^- \geq \varepsilon \xi_i^- \qquad i = 1, 2, \ldots, k$$

$$\xi_i^+ + \xi_i^- \leq 1 \qquad i = 1, 2, \ldots, k$$

$$\sum_{i=1}^{k} \left(\xi_i^+ + \xi_i^- \right) = k$$

Eş. 4.23 modelinde, λ_i^+ ve λ_i^- $(i = 1, \ldots, k)$ negatif olmayan değişkenler c_r $(r = 1, \ldots, h-1)$serbest değişkenler y_j $(j = 1, \ldots, n)$ ve ξ_i^+ ve ξ_i^- $(i = 1, \ldots, k)$ ise 0 ve 1 değerlerini alabilen ikili değişkenlerdir. Birimler aşağıda verilen kurala göre sınıflandırılmaktadır.

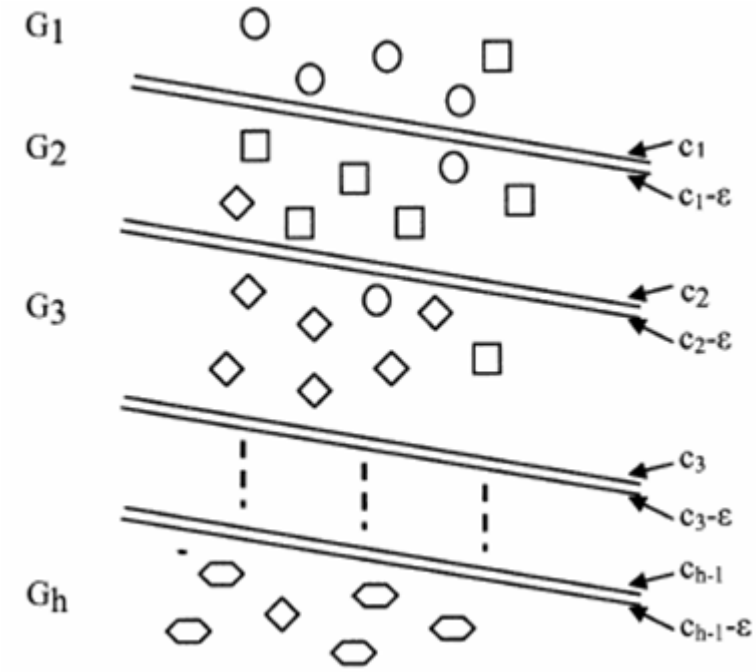
$\sum_{i=1}^{k} \lambda_i^* Z_{im} \geq c_1^*$ ise G_1 grubuna

$c_{r-1}^* - \varepsilon \geq \sum_{i=1}^{k} \lambda_i^* Z_{im} \leq c_r^*$ ise G_r $(r = 1, \ldots, h-1)$ grubuna

$c_{h-1}^{*} - \varepsilon \geq \sum_{i=1}^{k} \lambda_i^{*} Z_{im}$ ise G_h grubuna

sınıflandırma yapılmaktadır. Eş. 4.23 modeli ile $h-1$ tane farklı (c_1^{*}'dan c_{h-1}^{*}'a kadar) ayırma skoru elde edilmektedir. Ayırma skorları aynı λ_i^{*} ($i = 1, \ldots, k$) ağırlık değerleri ile elde edilmektedir. Şekil 4.1'de çoklu grupların sınıflandırılması gösterilmektedir.

Eş. 4.23 modeli çözüldüğünde optimal çözümün $c_1^{*} > c_2^{*} > \ldots > c_{h-1}^{*}$ olarak sağlanması gerekir. Çözüm bu durumu sağlamıyorsa $c_1 > c_2 + \varepsilon$, $c_2 > c_3 + \varepsilon$, . . . , $c_{h-2} > c_{h-1} + \varepsilon$ kısıtlarının Eş. 4.23 ile verilen modele eklenmesi gerekmektedir.



Şekil 4.1. Sueyoshi'nin çok gruplu sınıflandırma modelinin gösterimi

Sueyoshi tarafından verilen karma-tamsayılı yaklaşımı çok gruplu sınıflandırmanın özel bir türünü çözebilir. Burada özel bir türle kastedilen Şekil 4.1'den de görülebileceği gibi veri yapısının sıralı halde olmasıdır. Eğer veri seti gruplara göre

sıralı halde değilse mümkün olmayan çözümlerle veya düşük sınıflandırma oranları ile karşılaşılabilir.

4.2. Yeni Önerilen Çok Gruplu Sınıflandırma Modeli

Çok gruplu sınıflandırma problemi için önerilen matematiksel programlama yaklaşımlarının hem avantajları hem de dezavantajları vardır. Gehrlein (1986) tarafından önerilen tek fonksiyonlu sınıflandırma modeli ve Sueyoshi (2006) tarafından önerilen çok gruplu DEA-DA modeli veri setinin gruplara göre sıralı bir şekilde olduğu varsayımına dayanmaktadır ve ayrıca bu modeller karma tamsayılı sınıflandırma modelleridir. Bal (1999) tarafından önerilen ve birimlerin sınıflandırma skorlarının, grupların sınıflandırma skoru aralıklarından sapmaların minimum yapılmaya çalışıldığı doğrusal hedef programlamaya dayalı sınıflandırma modeli de veri yapısının sıralı olması varsayımına dayanmaktadır. Gehrlein'in çok fonksiyonlu modeli veri yapısında böyle bir düzenin olmasına ihtiyaç duymazken, modelde uygun bir normalleştirme kısıtının olmaması ve modelin karma-tamsayılı bir model olması çoğu durumda ağırlık katsayılarının sıfır olması gibi mantıklı olmayan veya mümkün olmayan çözümlerle karşılaşma olasılığını artırmaktadır. Lam ve Moy (1996) tarafından önerilen ve birimlerin sınıflandırma skorlarının grubun ortalama sınıflandırma skoruna yaklaştırıldığı modelde ise tüm mümkün ikili sınıflandırma modellerinin incelenmesi gerekmektedir ve sınıflandırma işlemi iki aşamada yapılmaktadır. Gochet ve ark. (1997) tarafından önerilen LP^q yaklaşımında veri yapısında özel bir sıralama varsayımı olmadığı gibi bu modeldeki tüm değişkenler negatif olmayandır ve ayrıca model tamsayılı bir model değildir. Bu modelin dezavantajı ise normalleştirme kısıtının uygun olmaması ve q sabitinin seçimi problemidir. Bu model q sabitinin uygun bir şekilde belirlenememesi gibi bazı durumlarda mümkün olmayan çözümler vermektedir [Östermark ve Höglund, 1998].

Sueyoshi'nin iki gruplu ve çok gruplu sınıflandırma problemleri için önerdiği DEA-DA modelleri veride negatif değişkenlere de izin vermesi ve birimleri bir konveks küme içerisine alarak sınıflandırmaya çalışması gibi özelliklerinden dolayı metodolojik olarak diğer modellerden oldukça güçlüdür [Sueyoshi, 2006]. Veri

yapısının özel bir sırada olması varsayımı bir kenara bırakılarak, Sueyoshi'nin DEA-DA modelindeki güçlü özellikler Gochet'in sınıflandırma modeli ile kombine edilebilir. Sueyoshi'nin DEA-DA modeli ve Gochet'in çok gruplu sınıflandırma modelinin bir kombinasyonu olan yeni sınıflandırma modeli Eş. 4.24 ile verilmektedir.

$$Min \sum_{r=1}^{h} \sum_{t \neq r}^{h} \sum_{j=1}^{n} n_{rt}^{j}$$

Kısıtlar:

$$\alpha_{r0} + \sum_{i=1}^{k} \alpha_{ri} Z_{ij} - \alpha_{t0} - \sum_{i=1}^{k} \alpha_{ti} Z_{ij} + n_{rt}^{j} - p_{rt}^{j} = \varepsilon \ , \ \forall j \in G_r \ , \ j = 1,\ 2,\ \ldots,\ n \qquad (4.24)$$

$$\sum_{r=1}^{h} \sum_{i=0}^{k} \alpha_{ri} = 1 \qquad , \ r = 1,\ 2,\ \ldots,\ h \ , \ r \neq t$$

$$\alpha_{ri} \geq 0 \qquad , \ r = 1,\ 2,\ \ldots,\ h \ , \ i = 0,\ 1,\ \ldots,\ k$$

Eş. 4.24 modeli ile bir birim en büyük sınıflandırma skoruna sahip olduğu gruba atanır. ε değeri çok küçük pozitif bir sayıdır ve herhangi bir birimin ayırma skorunun ayırma düzlemi üzerinde olmasını engellemek için modele konulmuştur. Modelde r. grupta yer alan birimlerin r. grup için oluşturulan $\alpha_{r0} + \sum_{i=1}^{k} \alpha_{ri} Z_{ij}$ ayırma fonksiyonunun, t. grup için oluşturulan $\alpha_{t0} + \sum_{i=1}^{k} \alpha_{ti} Z_{ij}$ ayırma fonksiyonundan ayırımı yolu ile gruplarına atanması istendiğinden $\alpha_{r0} + \sum_{i=1}^{k} \alpha_{ri} Z_{ij} - \alpha_{t0} - \sum_{i=1}^{k} \alpha_{ti} Z_{ij} \geq \varepsilon$ olması amaçlanmaktadır. Bu kısıta negatif n_{rt}^{j} ve pozitif p_{rt}^{j} sapma değişkenleri ilave edilerek ve istenmeyen n_{rt}^{j} sapma değişkenlerinin toplamı minimum yapılmaya çalışılarak birimlerin uygun gruplarına sınıflandırılması yapılmaktadır. İstenmeyen n_{rt}^{j} negatif sapma değişkenlerinin minimum yapılmasıyla $\alpha_{r0} + \sum_{i=1}^{k} \alpha_{ri} Z_{ij} - \alpha_{t0} - \sum_{i=1}^{k} \alpha_{ti} Z_{ij} \geq \varepsilon$ olması mümkün olduğunca sağlanacak ve grupların birbirinden ayırımı yapılabilecektir.

Eş. 4.19 ile verilen ve Gochet (1997) tarafından önerilen LP^q yaklaşımında modele $\sum_{r\in S}\sum_{t\in S_{-r}}\sum_{j\in P_r}\left(\gamma_{rt}^{j}-\beta_{rt}^{j}\right)=q$ kısıtı eklenerek tüm ayırma katsayılarının sıfır olması gibi uygun olmayan çözümlerden kurtulmak amaçlanmaktadır. Burada q pozitif bir sayıdır. q sabitinin farklı seçimlerinden elde edilecek ayırma fonksiyonları ve birimlerin doğru sınıflandırılması farklı şekillerde etkilenebilir. Seçilecek her q sabiti için farklı bir sonuç elde etmek mümkündür.

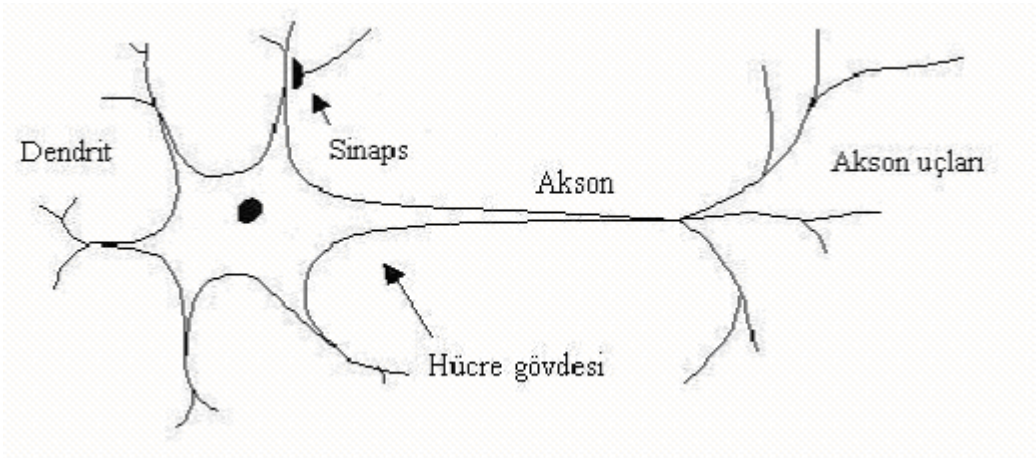
Bu çalışmada yeni önerilen matematiksel programlama modelinde uygun olmayan çözümlerden kaçınmak için ayırma fonksiyonu katsayılarının toplamının 1 olması amaçlanmıştır. Modele $\sum_{r=1}^{h}\sum_{i=0}^{k}\alpha_{ri}=1$ kısıtı eklenerek uygun olmayan çözümler engellenmeye çalışılmıştır. Böylelikle en az bir tane ayırma katsayısı değeri pozitif olarak elde edilmekte ve Sueyoshi tarafından önerilen sınıflandırma modellerine benzer olarak da birimler bir konveks küme içinde sınıflandırılmaktadır.

5.YAPAY SİNİR AĞLARI YAKLAŞIMLARI

Bu bölümde yapay sinir ağları ile ilgili temel kavramlar ve yapay sinir ağlarının eğitilmesinde kullanılan klasik geri yayılım algoritmasının sınıflandırma problemlerinin çözümünde kullanımı üzerinde durulmaktadır.

5.1. Yapay Sinir Ağları ile İlgili Temel Kavramlar

Yapay Sinir Ağları (YSA) beyindeki sinirlerin çalışmasını taklit ederek sistemlere öğrenme, genelleme yapma, hatırlama gibi yetenekler kazandırmayı amaçlayan bir bilgi işleme sistemidir. Bu durumda YSA'yı anlayabilmek için öncelikle biyolojik sinir ağlarını tanımak gerekir. Biyolojik sinir ağları insan beyninde bulunan milyarlarca sinir hücresinden oluşur. İnsan beyninde 10^{10} adet sinir hücresi ve hücreler arasında akışı sağlayan $6x10^{13}$'ten fazla bağlantı bulunmaktadır. Basit bir biyolojik sinir hücresi Şekil 5.1'de görülmektedir.



Şekil 5.1. Biyolojik sinir hücresi.

Bugün hala insan beyninin nasıl çalıştığı tam olarak bilinmemektedir. YSA konusundaki çalışmaların, beynin gizemi çözülmeye başladıkça sıçramalar göstereceği düşünülmektedir. İnsan beyni, çok hızlı çalışabilen mükemmel bir bilgisayar gibi görülebilir. Biyolojik sinir ağlarının performansları küçümsenemeyecek kadar yüksektir ve son derece karmaşık olayları bile

işleyebilecek yetenektedir. Bu nedenle YSA modelleri kullanılarak bu yeteneğin bilgisayara kazandırılması amaçlanmaktadır [Elmas, 2003].

Genel anlamda YSA, beynin bir işlevini yerine getirme yöntemlerini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Bir YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur. YSA öğrenme algoritmaları ile öğrenme sürecinden geçtikten sonra, bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip olurlar. YSA yapılarına göre farklı öğrenme yaklaşımları kullanırlar.

Kohonen, YSA'yı şöyle tanımlamaktadır [Patterson, 1996].

"Yapay sinir ağları paralel olarak bağlantılı ve çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran hiyerarşik bir organizasyonudur"

Haykin, YSA' yı şöyle tanımlamaktadır [Elmas, 2003].

"Bir sinir ağı, bilgiyi depolamak için doğal eğilimi olan basit birimlerden oluşan paralel dağıtılmış bir işlemcidir"

İlk YSA modeli 1943 yılında McCulloch ve Pitts tarafından geliştirilmiştir. 1949 yılında Hebb, YSA'daki öğrenme için başlangıç noktası sayılabilecek bir kuralı geliştirmiştir. Ortaya atılan bu öğrenme kuralı, o dönemde bir sinir ağının öğrenme işini nasıl gerçekleştirebileceği konusunda fikir vermekle birlikte, bugün halen geçerli olan öğrenme kurallarından birçoğunun da temelini teşkil etmektedir. 1958 yılında Frank Rosentblatt'ın basit algılayıcıyı (perceptron) geliştirmesiyle YSA alanındaki gelişmeler hız kazanmıştır [Şen, 2004a]. Basit algılayıcı model tek katmanlı, eğitilebilen ve tek çıkışlı bir YSA modelidir. 1960 yılında Widrow ve Hoff tarafından ADALINE (Adaptive Linear Combiner) ağ modelleri geliştirilmiştir. 1969 yılında Minsky ve Papert basit algılayıcının yetersizliğini görmüşler ve VE/VEYA(XOR) problemlerini çözemediğini ispatlamışlardır. Basit algılayıcının birçok konuda yetersiz kaldığının gösterilmesi YSA üzerindeki çalışmaları bir süre

durdurmuştur. 1980'li yılların başında yapılan çalışmalarla YSA yeniden yaygın hale gelmiştir. Hopfield 1982 yılında doğrusal olmayan ağları geliştirmiştir. Kohonen ve Anderson tarafından yapılan çalışmalar sonucunda danışmansız öğrenen ağların geliştirilmesiyle çalışmalar yeniden ivme kazanmıştır. 1986 yılında Rumelhart ve ark. tarafından çok tabakalı algılayıcı tipi ağlar için "geri yayılma" olarak adlandırılan bir eğitme algoritması geliştirilmiştir [Patterson, 1996]. 1987 yılında Elektrik Elektronik Mühendisliği Enstitüsü (Instute of Electrical Electronic Engineering (IEEE)) tarafından sinir ağlarını konu alan ilk uluslararası konferans 1800'ü aşkın katılımcıyla gerçekleştirilmiştir. Optik ve dijital teknolojilerdeki gelişmeler ve yeni teknolojilerin keşfedilmesiyle bilgisayarların hızları artmış ve bu sayede YSA uygulamaları da hız kazanmıştır. Günümüzde YSA'lar üzerinde yapılan çalışmalar dünyada büyük hızla devam etmektedir [Patterson, 1996].

5.1.1. Yapay sinir ağlarının özellikleri

YSA'lar giriş ve çıkışları bulunan kara kutular gibi düşünülebilir. Birçok girişi ve birçok çıkışı vardır. Kara kutunun işlevi basitçe matematiksel bir fonksiyonu temsil etmek şeklinde açıklanabilir.

Şekil 5.2. Yapay sinir ağlarının genel görünümü

YSA'larda her bir işlem birimi, basit bir anahtar görevi yapar ve şiddetine göre, kendisine gelen sinyalleri zayıflatır ya da kuvvetlendirir. Böylece sistem içindeki her birim belli bir ağırlığa sahip olmuş olur. Her birim sinyalin gücüne göre açık ya da kapalı duruma geçer ve basit bir tetikleyici görev üstlenir. Ağırlıklar işlem birimleri

arasında ilgi kurmayı sağlar. Yapay sinir ağları araştırmalarının odağındaki soru, ağırlıkların sinyalleri nasıl değiştirmesi gerektiğidir. Bu noktada her hangi bir formdaki bilgi girişinin ne tür bir çıkışa çevrileceği, değişik modellerde farklılık göstermektedir. Bir yapay sinir ağ tasarımında, bilgisayarda saklı olan bilgi tüm sisteme yayılmış küçük yük birimlerinin birleşmesinden oluşmaktadır. Ortama yeni bir bilgi aktarıldığında yerel büyük bir değişiklik yerine tüm sistemde küçük bir değişiklik meydana gelmektedir.

YSA'ların iki temel özelliği bu sistemlere kuvvetli bilgi işlem özelliği kazandırmaktadır. İlki bu sistemlerin paralel işlemci yapısı, ikincisi ise öğrenebilme ve genelleme yapabilme özellikleridir. Genelleme yapabilmesi, sistemin eğitimi sırasında öğrenmediği bazı bilgileri tutarlı sonuç olarak verebilmesidir. YSA'ların avantaj ve yetenekleri aşağıda verildiği gibi özetlenebilir.

1. Bir nöron (sinir hücresi) doğrusal olmayan bir yapıya sahiptir. Bu sebeple nöronlardan meydana gelen YSA'lar da doğrusal olmayan özelliğe sahiptir. Özellikle doğrusal olmayan girdi bilgilerini işleyen sistemler için bu önemli bir avantajdır.

2. Ağırlık değerleri üzerinde yapılacak değişiklikler sayesinde YSA sistemleri bulundukları ortama kolayca uyum sağlayabilme özelliğine sahiptir. Belli bir ortam için eğitilmiş olan bir sistem, ortam değişikliği durumunda tekrar eğitilebilir.

3. Sınıflama yapabilen bir YSA modeli sadece verilen girdinin hangi sınıfa ait olduğunu belirlemekle kalmaz, aynı zamanda öğrenme oranı kullanarak belli bir hassasiyet altında gerçekleştirir. Bu da sistemin sınıflama performansını arttırır.

4. YSA modelleri yapılan performans testleri ile hataya dayanıklı sistemler olarak kabul edilmektedir. Örneğin bir nöronun veya nöronun bir bağlantısının hizmet dışı olması durumunda da sistem doğruya yakın sonuçlar verebilir. Bunda en önemli

etmen, YSA sistemlerinin bilgiyi paralel işlemcilerde saklama özelliğidir [Sağıroğlu ve ark., 2003].

5. YSA'lar, paralel işlem olanağı sağlar ve bu sayede hesaplama zamanını azaltır. Adaptasyon kabiliyeti sayesinde, bazı ağ tipleri kendi kendini eğitme ve dinamik sistemler gibi kararlı hale gelebilme özelliğine sahiptirler. Genelleme kabiliyeti sayesinde eğitme aşamasında kullanılan giriş değerleri haricindeki değerlerce de kabul edilebilir sonuçlar verebilmektedir.

6. YSA'larda hafıza birçok birim üzerine dağıldığı için gürültüye yani veriler içerisindeki kirliliğe karşı koyma özelliği kazandırmaktadır.

7. Nöron yapısı, birçok yapay sinir ağı modelinde aynı özellikleri göstermektedir. Bu genelleme farklı uygulamalar için kullanılan teori ve öğrenme algoritmalarının paylaşılabilirliğini sağlamaktadır. Böylece modüler ağlar oluşturmak mümkün olmaktadır [Şen, 2004a].

Sinir ağlarının dezavantajları ise aşağıda verildiği gibi özetlenebilir [Elmas, 2003].

1. YSA'ların donanım bağımlı çalışmaları önemli bir sorun olarak görülebilir. Ağların özellikle, gerçek zamanlı bilgi işleyebilmeleri paralel çalışabilen işlemcilerin varlığına bağlıdır. Ayrıca bir ağın nasıl oluşturulması gerektiğini belirleyecek kuralların olmaması da başka bir dezavantajdır. Her problem farklı sayıda işlemci gerektirebilir. Bazı problemleri çözebilmek için gerekli olan paralel işlemcilerin tamamını bir arada paralel olarak çalıştırmak mümkün olmayabilir.

2. Probleme uygun ağ yapısının genellikle deneme yanılma yoluyla belirlenebiliyor olması da önemli bir problemdir. Eğer problem için uygun bir ağ oluşturulmaz ise çözümü olan bir problemin çözülememesi veya performansı düşük çözümlerin elde edilmesi söz konusu olabilir. Bu aynı zamanda bulunan çözümün en iyi çözüm

olduğunu da garanti etmez. Yani YSA'lar bazı durumlarda kabul edilebilir çözümler üretemeyebilirler.

3. Bazı ağlarda öğrenme katsayısı gibi ağın parametre değerleri, her katmanda olması gereken düğüm sayısı, katman sayısı gibi değerlerin belirlenmesinde bir kural olmaması da başka bir problemdir.

4. Ağın çözeceği problemin ağa tanıtımı çok önemlidir. YSA'lar sadece nümerik bilgilerle çalışmaktadır. Bu nedenle de problemin nümerik gösterime dönüştürülmesi gerekir. Uygun bir gösterim mekanizmasının kurulamamış olması problemin çözümünü engelleyebilmekte veya düşük performanslı bir çözüm elde edilmesine neden olabilmektedir.

5. Ağın eğitiminin ne zaman bitirileceğine karar vermek için geliştirilmiş bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlanması için yeterli görülmektedir. Fakat sonuçta en iyi öğrenmenin gerçekleştiği söylenememektedir. Sadece iyi çözümler üretebilen bir ağ oluştu denilmektedir. Bir diğer sorunda ağın davranışlarının açıklanamamasıdır.

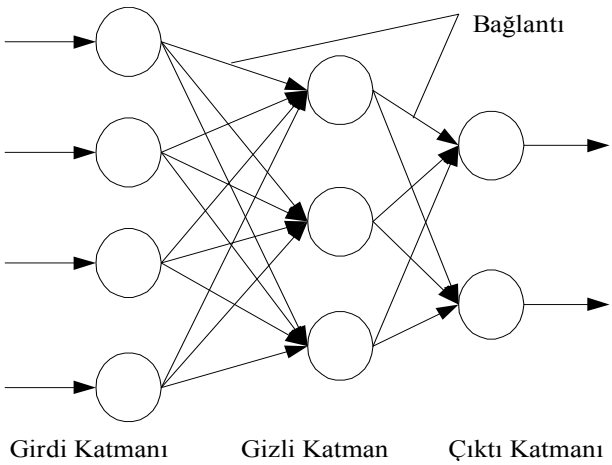
5.1.2. Bir yapay sinir ağının yapısı ve ana öğeleri

Yapay sinir hücreleri bağlantılar aracılığıyla bir araya gelerek YSA'yı oluştururlar. Şekil 5.3' de bir YSA örneği verilmiştir.

Genel olarak bir sinir ağını oluşturan katmanlar, girdi katmanı, gizli katman ve çıktı katmanıdır. Katmanlar ağdaki sinirlerin aynı doğrultu üzerinde bir araya gelmelerinden oluşur. Bu katmanlar kısaca şöyle özetlenebilir [Patterson, 1996]:

Girdi katmanı: Dış dünyadan bilgileri alarak gizli katmanlara gönderir.

Gizli katman: Girdi katmanından gelen bilgilerin işlenerek çıktı katmanına gönderildiği katmanlardır. Bir ağda birden fazla gizli katman olabilir. Ara katman olarak da isimlendirilmektedir.



Şekil 5.3. Bir yapay sinir ağı örneği

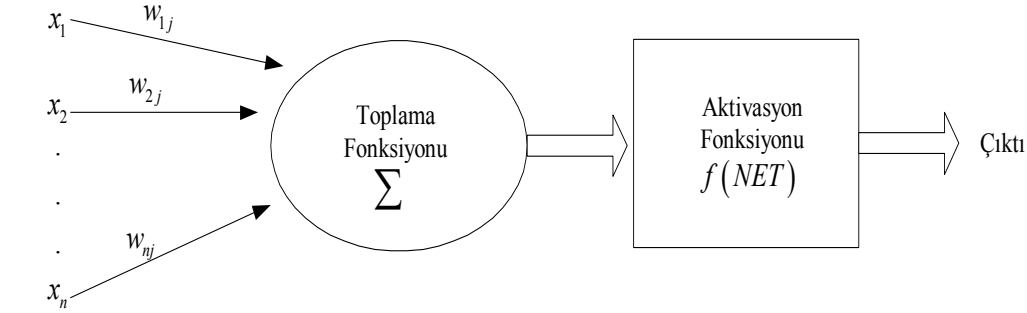
Çıktı katmanı: Gizli katmanlardan gelen bilgileri işleyerek gereken çıktıyı dış dünyaya gönderir. YSA' da işlem elemanları (düğümler) basit sinirler olarak adlandırılır. Yapay bir sinir hücresinin yapısı Şekil 5.4'de gösterilmiştir. Her sinirin 5 temel elemanı vardır. Bunlar: girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu (etkinlik işlevi) ve hücrenin çıktısıdır [Elmas, 2003].

Girdiler: Bir yapay sinir hücresine dış dünyadan gelen bilgilerdir. Bu bilgiler kendinden önceki sinirden veya dış dünyadan gelebilir.

Ağırlıklar: Yapay sinir tarafından alınan girişlerin sinir üzerindeki önemini belirleyen katsayılardır (w_1, w_2, . . . , w_n). Girişlerin hücre içerisindeki etkisini belirler ve her bir girişin kendine ait bir ağırlığı vardır.

Toplama fonksiyonu: En yaygın olarak kullanılan toplama fonksiyonu ağırlıklı toplamı bulmaktır. Bu fonksiyon ile hücreye gelen net girdi hesaplanır. Toplama fonksiyonu, sinirde her bir ağırlığın ait olduğu girişlerle çarpımının toplamlarını eşik değeri (θ_j) ile toplayarak aktivasyon fonksiyonuna gönderir.

$$NET = \sum_{i=1}^{n} w_{ij} x_i + \theta_j \tag{5.1}$$



Şekil 5.4. Yapay bir sinir hücresi

$f\left(wx + \theta_j\right)$ şeklinde nöron çıkışı hesaplanır. Buradaki **w** ağırlıklar matrisi, **x** ise girdi matrisidir. n girdi sayısı olmak üzere;

$$\mathbf{w} = \left(w_1, w_2, \ldots, w_n\right)$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n\right)$$

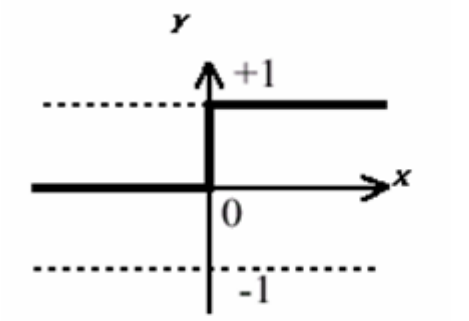
şeklinde yazılabilir. Formule edilirse;

$$NET = \sum_{i=1}^{n} w_{ij} x_i + \theta_j \quad \text{ve çıktı} = f\left(NET\right) = f\left(\sum_{i=1}^{n} w_{ij} x_i + \theta_j\right) \tag{5.2}$$

elde edilir. Eş. 5.2'de görülen f aktivasyon fonksiyonudur. Genelde doğrusal olmayan aktivasyon fonksiyonunun çeşitli tipleri vardır.

Aktivasyon fonksiyonu: Toplama fonksiyonu ile elde edilen net girdiden hücrenin üreteceği çıktıyı belirleyen fonksiyondur. İlgilenilen problemin türüne ve ağ yapısına bağlı olarak çeşitli aktivasyon fonksiyonları kullanılabilir. Şekil 5.5'de eşik aktivasyon fonksiyonunun grafiği görülmektedir. Eşik aktivasyon fonksiyonu net çıkışında eğer net değeri sıfırdan küçükse sıfır, sıfırdan daha büyük bir değer ise $+1$ değeri verir. Eşik aktivasyon fonksiyonunun -1 ile $+1$ arasında değişeni ise Signum

aktivasyon fonksiyonu olarak adlandırılır. Signum aktivasyon fonksiyonu, net giriş değeri sıfırdan büyükse $+1$, sıfırdan küçükse -1, sıfıra eşitse sıfır değerini verir.

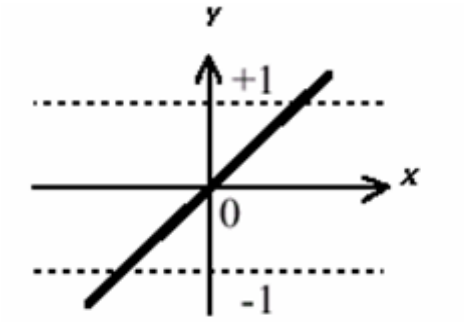


Şekil 5.5. Eşik aktivasyon fonksiyonu.

Doğrusal aktivasyon fonksiyonu

$$f(x) = x \tag{5.3}$$

şeklinde ifade edilir. Doğrusal aktivasyon fonksiyonunun çıkışı girişine eşittir. Sürekli çıkışlar gerektiği zaman çıkış katmanındaki aktivasyon fonksiyonunun doğrusal aktivasyon fonksiyonu olabildiğine dikkat edilmelidir. Şekli 5.6'da doğrusal aktivasyon fonksiyonu görülmektedir.



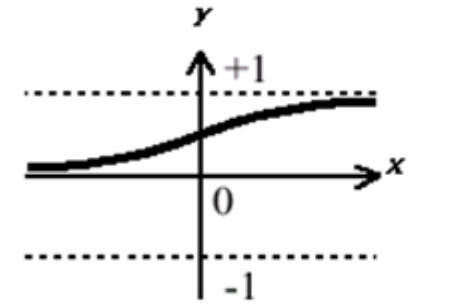
Şekil 5.6. Doğrusal aktivasyon fonksiyonu

Lojistik fonksiyonu,

$$f(x) = lojistik(x) = \frac{1}{1+e^{-\beta x}} \tag{5.4}$$

şeklinde ifade edilir. Eş. 5.4'deki β eğim sabiti olup genelde bir olarak seçilmektedir. Lojistik fonksiyon olarak da adlandırılmaktadır. Bu fonksiyonun

doğrusal olmadığından dolayı türevi alınabilmektedir böylece geri yayınımlı ağlarda kullanmak mümkün olabilmektedir. Şekil 5.7'de logaritma sigmoid transfer fonksiyonu görülmektedir.



Şekil 5.7. Logaritma Sigmoid aktivasyon fonksiyonu.

Diğer bir aktivasyon fonksiyonu olan hiperbolik tanjant aktivasyon fonksiyonu da doğrusal olmayan türevi alınabilir bir fonksiyondur. -1 ile $+1$ arasında çıkış değerleri üreten bu fonksiyon lojistik fonksiyona benzemektedir. Denklemi

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{5.5}$$

olarak ifade edilir.

Bu aktivasyon fonksiyonlarından başka fonksiyonlar da vardır. Yapay sinir ağında hangi aktivasyon fonksiyonunun kullanılacağı probleme bağlı olarak değişmektedir. Yukarıda verilen fonksiyonlar en genel aktivasyon fonksiyonlarıdır.

Hücrenin çıktısı: Aktivasyon fonksiyonu tarafından belirlenen hücre çıktısıdır. Dış dünyaya ya da başka bir sinire gönderilir. Bir sinir hücresinden çıkan tek bir çıktı değeri vardır.

5.1.3. Yapay sinir ağlarının oluşturulmasında dikkat edilmesi gereken hususlar

Yapay Sinir Ağlarında uygulama başarısı uygun parametrelerin seçimine bağlıdır. Uygun parametrelerin seçimi ise kolaylıkla yapılamamaktadır. Uygun parametrelerin seçiminde karşılaşılan birçok güçlükler vardır. Bunlar,

- Probleme uygun olan YSA yapısının seçimi
- Problemin kabul edilebilir çözümü için YSA giriş ve çıkış sayılarının en uygun veya en az sayıda seçimi
- Gizli katman hücre sayılarının en uygun sayıda belirlenmesi
- Gizli katman sayısının seçimi
- Kullanılacak öğrenme algoritmasının YSA yapısına uygun olması
- En uygun öğrenme algoritması parametrelerinin seçimi
- Seçilen veri kodlama yapısı
- Veri normalizasyon yaklaşımı
- Seçilen transfer fonksiyonunun yapısı
- Toplama fonksiyonu tipi
- Uygun performans fonksiyonu seçimi
- Uygun iterasyon sayısı
- Ön işleme ve son işleme
- Uygun veri tipinin ve sayısının belirlenmesi

olarak sıralanabilir [Sağıroğlu ve ark., 2003].

5.1.4. Yapay sinir ağlarının uygulama alanları

Başarılı YSA uygulamaları incelendiğinde ağların, doğrusal olmayan, çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu verilerinin olması durumunda ve problemin çözümü için özellikle bir matematiksel modelin ve algoritmanın bulunmaması hallerinde yaygın olarak kullanıldıkları görülmektedir. Bu amaçla geliştirilmiş ağlar genel olarak şu işlevleri yerine getirmektedir:

- Sınıflandırma
- Örüntü tanıma
- Doğrusal olmayan sistem modelleme
- Doğrusal olmayan sinyal işleme
- Optimizasyon
- Zaman serileri analizi
- İlişkilendirme veya örüntü eşleştirme
- Olasılık fonksiyonlarının kestirimleri

Yukarıda belirtilen teorik konular dışında finansal konulardan mühendisliklere ve tıp bilimine kadar pek çok alanda da YSA uygulamalarından bahsetmek mümkündür.

- Bankalardan kredi isteyen müracaatları değerlendirme
- Ürünün pazardaki performansını tahmin etme
- Kredi kartı hilelerini tahmin etme
- Zeki araçlar ve robotlar için optimum rota belirleme
- Güvenlik sistemlerinde ses ve parmak izi tanıma
- Robot hareket mekanizmalarının kontrol edilmesi
- Kalite kontrolü
- Radar sinyallerini sınıflandırma
- Kan hücreleri reaksiyonları ve kan analizlerini sınıflandırma

Günümüzde hemen hemen her alanda YSA çalışmalarını görmek mümkündür. Çünkü gerçek hayatta karşılaşılan problemlerin çoğu doğrusal olmayan modeller gerektirmektedir. Bu durumda da geleneksel yöntemler ile çözüm üretilmesi zorlaşmakta veya imkânsız olmaktadır. YSA'lar geleneksel yollarla çözümü zor olan problemlere daha doğru ve gerçekçi çözümler üretmeyi amaçlamaktadır [Sağıroğlu ve ark., 2003].

5.1.5. Basit algılayıcı yapay sinir ağı ve iki gruplu sınıflandırma problemine uygulanması

Basit algılayıcı ağ yapısı, gizli katmanı olmayan sadece girdi ve çıktı katmanından oluşan bir yapıdır. Aktivasyon fonksiyonu Şekil 5.5'de gösterilen eşik aktivasyon fonksiyonudur. Basit algılayıcı ağ algoritması aşağıda verilmektedir.

1. Tüm ağırlık ve ayırıcı doğrunun sabit değerlerine (w_0) küçük rasgele sayılar verilir.

2. Ağa girdi seti ve ona karşılık gelen çıktı seti gösterilir ve net girdi hesaplanır $NET = \sum_{i=1}^{n} w_{ij} x_i + \theta_j$.

3. Net girdi eşik değeri ile kıyaslanır. Net girdinin eşik değerinden büyük ya da küçük olmasına göre çıktı değeri 0 ve 1 değerinden birisini alır.

$$Çıktı = \begin{cases} 1, & NET > 0 \\ 0, & NET \le 0 \end{cases}$$

4. Eğer beklenen çıktı ile gerçekleşen çıktı aynı ise, ağırlıklarda herhangi bir değişiklik olmaz. Ağ, beklenmeyen bir çıktı üretmiş ise iki durum söz konusudur.

1.Durum: Ağın beklenen çıktısı 0 iken gerçekleşen çıktısı 1 ise, ağırlık değerleri azaltılır.

$w_n = w_0 - x$

2. Durum: Ağın beklenen çıktısı 1 iken, gerçek çıktısı 0 ise, ağırlık değerleri artırılır.

$w_n = w_0 + x$

5. 2–4 adımlar bütün girdi setindeki örnekler için doğru sınıflandırmalar yapılıncaya kadar tekrar edilir [Elmas, 2003].

Sueyoshi (1999)'dan alınan, DEA-DA modeli ile Fisher'in istatistiksel sınıflandırma modeli ve bazı matematiksel programlama modellerinin örneklendirildiği hipotetik veri Çizelge 5.1'de verilmektedir. Hipotetik örnek üzerinde Fisher, MSD, Stam ve Ragsdale'nin iki aşamalı MSD, DEA-DA ve basit algılayıcı sinir ağı algoritması ile elde edilen ayırma fonksiyonları incelenmektedir.

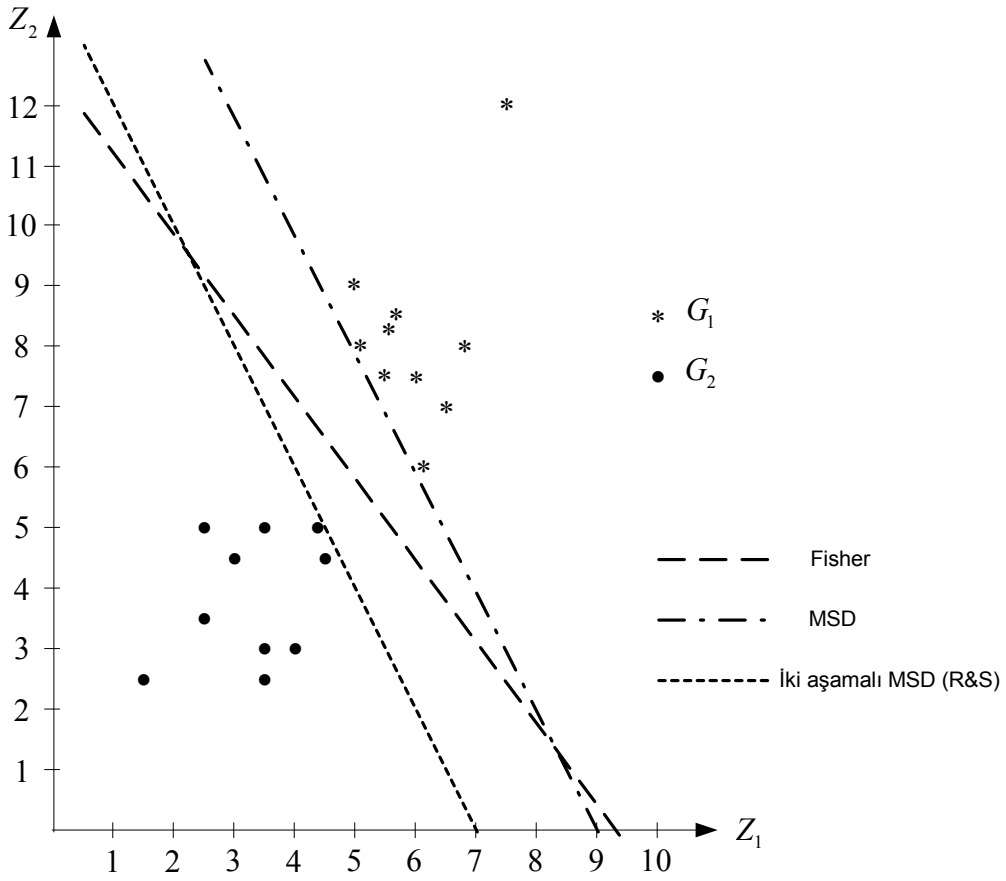
Fisher, MSD ve iki aşamalı MSD modelleri tarafından elde edilen ayırma fonksiyonları Şekil 5.8'de gösterilmektedir.

Çizelge 5.1. Basit algılayıcı model için hipotetik veri

Birim	Z_1	Z_2	Grup	Birim	Z_1	Z_2	Grup
1	5	9	G_1	11	3,5	5	G_2
2	5,6	8,5	G_1	12	4,5	4,5	G_2
3	5	8	G_1	13	4,5	5	G_2
4	5,5	7,5	G_1	14	3	4,5	G_2
5	6	6	G_1	15	2,5	3,5	G_2
6	5,8	8,3	G_1	16	3,5	2,5	G_2
7	7,5	12	G_1	17	1,5	2,5	G_2
8	6,5	7	G_1	18	4	3	G_2
9	7	8	G_1	19	3,5	3	G_2
10	6	7,5	G_1	20	2,5	5	G_2

İki aşamalı MSD yaklaşımı ile birinci aşamada elde edilen ayırma fonksiyonu $f = 0,50Z_1 + 0,25Z_2 - 3,5$ olarak elde edilmektedir ve çakışma durumu olmadığından ikinci aşamaya geçilmemiştir. Bu ayırma fonksiyonuna göre de tüm birimler doğru olarak sınıflandırılmaktadır.

Fisher'in istatistiksel yaklaşımı ile elde edilen ayırma fonksiyonu $f = 2,55Z_1 + 1,86Z_2 - 23,9$ olarak elde edilmektedir ve bu ayırma fonksiyonuna göre tüm birimler doğru olarak sınıflandırılmaktadır.

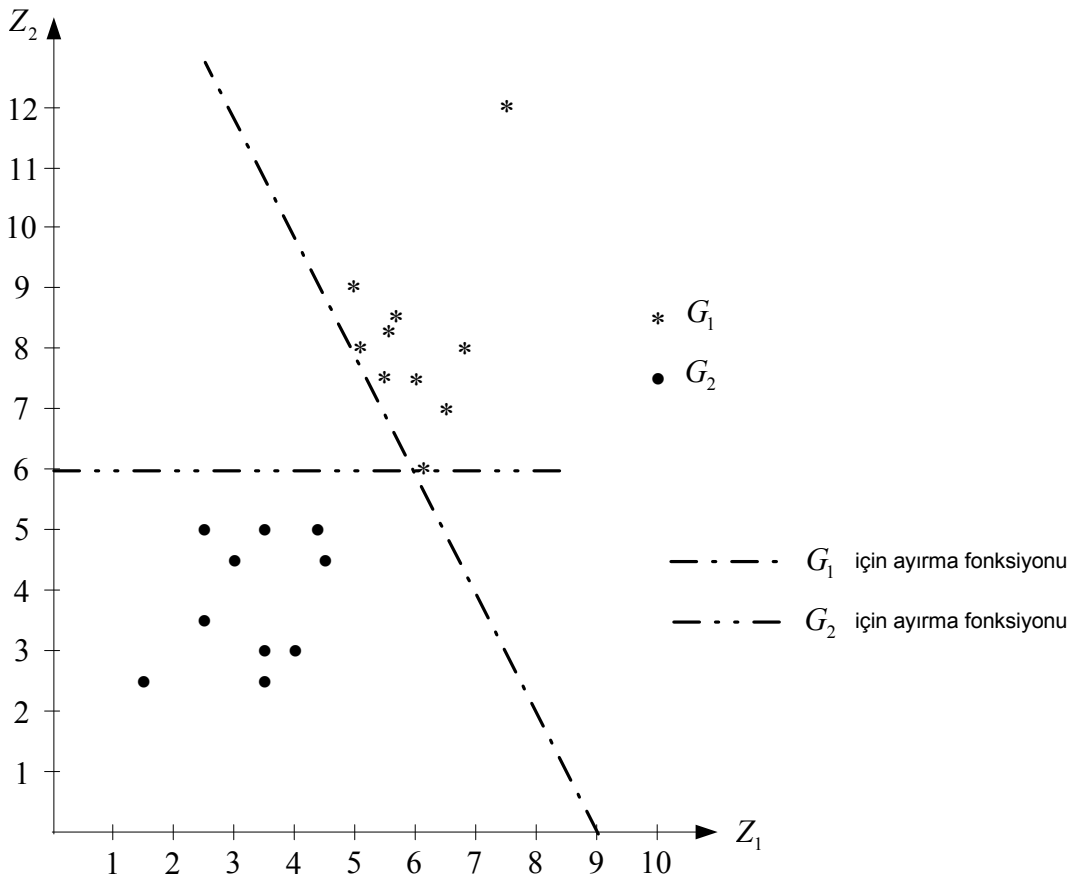


Şekil 5.8. Örnek için Fisher, MSD ve iki aşamalı MSD ayırma fonksiyonları

MSD yaklaşımı ile elde edilen ayırma fonksiyonu $f = 0,25Z_1 + 0,125Z_2 - 2,25$ olarak elde edilmektedir ve bu ayırma fonksiyonuna göre tüm birimler doğru olarak sınıflandırılmaktadır.

DEA-DA modeli tarafından elde edilen ayırma fonksiyonları Şekil 5.9'da gösterilmektedir.

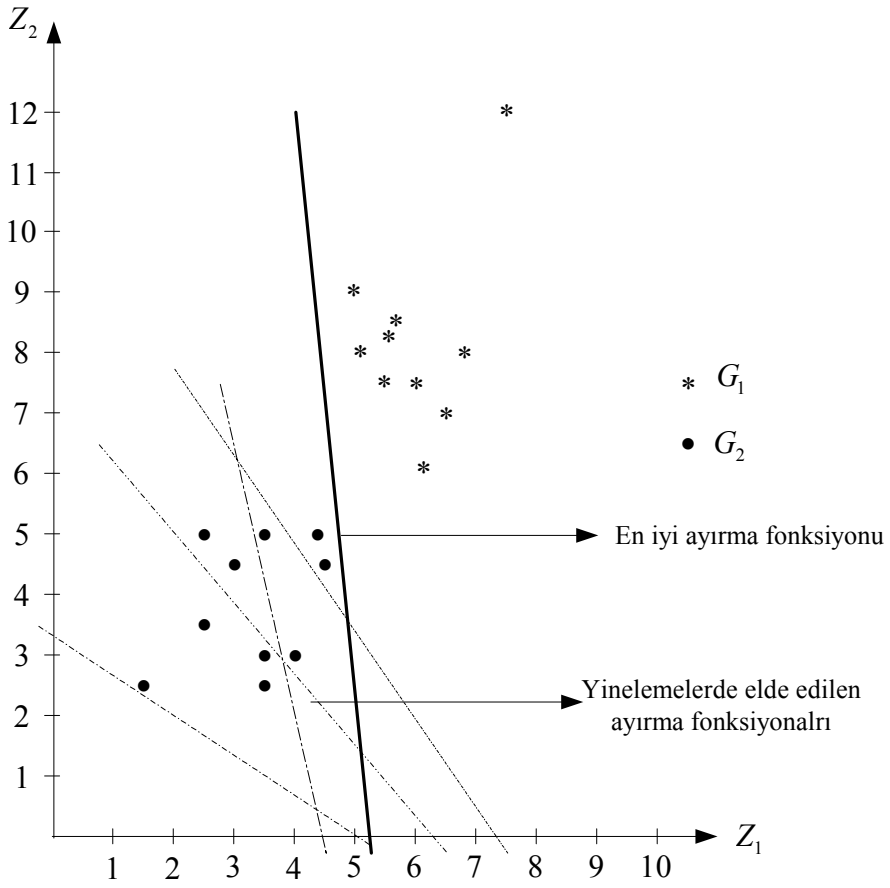
DEA-DA yaklaşımı ile birinci aşamada elde edilen ayırma fonksiyonları G_1 ve G_2 grupları için sırasıyla $f = 0,667Z_1 + 0,333Z_2 - 6$ ve $f = 0Z_1 + 1Z_2 - 6$'dir. Şekil 5.9'dan da görüldüğü gibi ilk aşamada çakışma durumu olmadığından ikinci aşamaya geçilmemiştir. Bu ayırma fonksiyonuna göre de tüm birimler doğru olarak sınıflandırılmaktadır.



Şekil 5.9. Örnek için DEA-DA ayırma fonksiyonları

Aynı örnek üzerinde basit algılayıcı ağ yapısını kullanarak iki grubu birbirinden en iyi şekilde ayıracak ayırma fonksiyonunu elde edilmesi de incelenmiştir.

Başlangıç ayırma fonksiyonu, $w_{01} = 0,4$, $w_{02} = 0,6$ ve $\theta_1 = -2$ olmak üzere rasgele olarak $f = 0,4Z_1 + 0,6Z_2 - 2$ alınsın. Başlangıç ayırma fonksiyonuna göre G_1 grubundaki tüm birimler doğru sınıflandırılırken, G_2 grubundaki tüm birimler ise yanlış sınıflandırılmıştır. Basit algılayıcı algoritması uygulandığında en iyi ayırma fonksiyonu, $f = -1,7Z_1 + 15,6Z_2 - 82$ olarak elde edilmektedir. Şekil 5.10'da basit algılayıcı ağ yapısı ile başlangıç ayırma fonksiyonu, ilk birkaç yinelemede elde edilen ayırma fonksiyonları ve en iyi ayırma fonksiyonu yer almaktadır. $f = -1,7Z_1 + 15,6Z_2 - 82$ ayırma fonksiyonu ile tüm birimler doğru sınıflandırılmıştır.



Şekil 5.10. Örnek için basit algılayıcı ayırma fonksiyonu

Ele alınan örnekte G_1 ve G_2 gruplarında yer alan birimler birbirinden tam olarak ayrılmaktadır ve ele alınan yöntemlerin hepsi birimleri tam olarak sınıflandırabilmektedir.

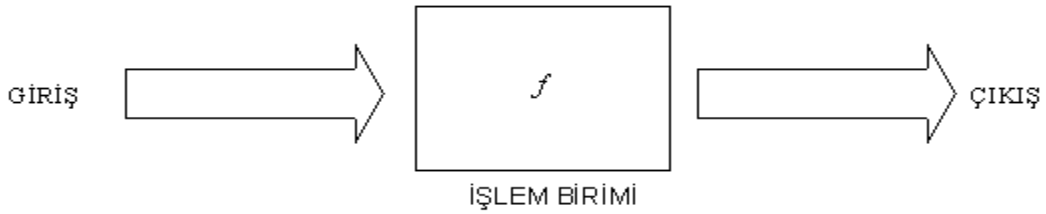
5.2. Ağ Yapıları

Yapay sinir ağları düğüm veya sinir olarak adlandırılan çok sayıdaki işlem elemanının bir araya gelmesinden oluşur. Yapay sinir ağlarının düğümleri ve bağlantıları çok değişik biçimlerde bir araya getirilebilir. Ağlar bu düğüm ve bağlantı mimarilerine göre değişik isimler alırlar. Yapay sinir ağ mimarileri, sinirler arasındaki bağlantıların yönlerine göre veya ağ içindeki işaretlerin akış yönlerine göre birbirlerinden ayrılmaktadır. Buna göre yapay sinir ağları için, ileri beslemeli

(feed forward) ve geri beslemeli (feedback) olmak üzere iki temel ağ mimarisi vardır [Fine, 1999].

5.2.1. İleri beslemeli ağlar

İleri beslemeli bir ağ yapısında işlemci elemanlar genellikle katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir [Patterson, 1996]. İşlemci elemanlarla bir katmandan diğerine bağlantı kurulurken, bu elemanla aynı katman içinde bağlantı kurulamaz. Şekil 5.11'de ileri beslemeli ağ yapısı için blok diyagram gösterilmiştir. İleri beslemeli ağlara örnek olarak çok katmanlı algılayıcı (Multi Layer Perceptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir.



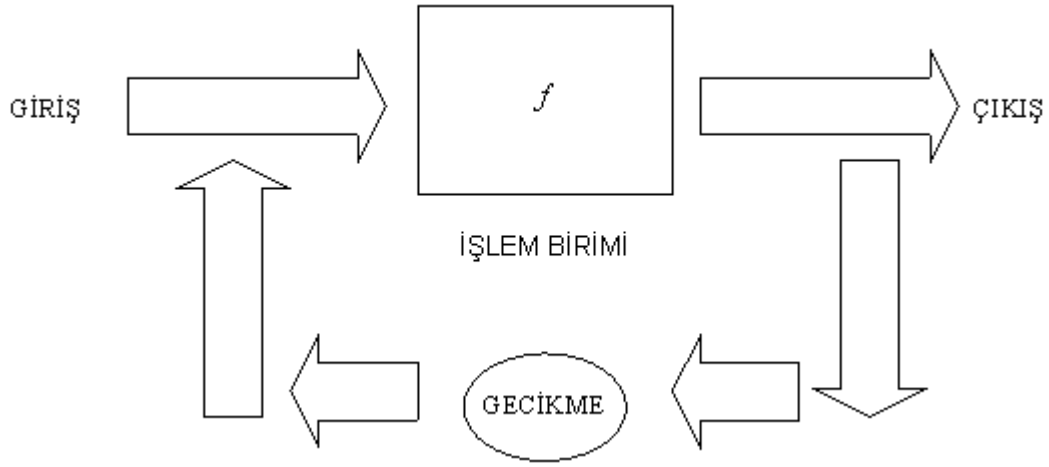
Şekil 5.11. İleri beslemeli ağ yapısı için blok diyagramı

İleri beslemeli ağ yapısı, en genel anlamda giriş uzayıyla çıkış uzayı arasında statik haritalama yapar. Bir andaki çıkış, sadece o andaki girişin bir fonksiyonudur.

5.2.2. Geri beslemeli ağlar

Bir geri beslemeli sinir ağı yapısı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir yapıdır. Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Şekil 5.12'de bir geri beslemeli ağ yapısı görülmektedir. Bu çeşit sinir ağlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, bu ağ yapısı özellikle önceden tahmin uygulamaları için uygundur [Şen, 2004a]. Bu ağlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak

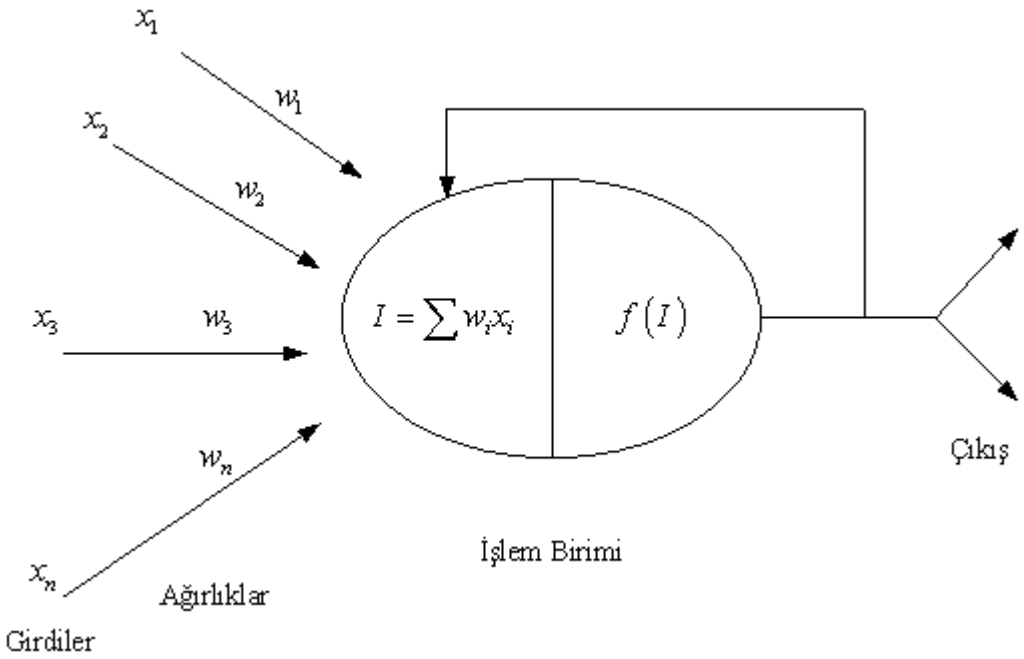
Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir [Sağiroğlu ve ark., 2003].



Şekil 5.12. Geri beslemeli ağ yapısı için blok diyagramı.

5.2.3. ADALINE

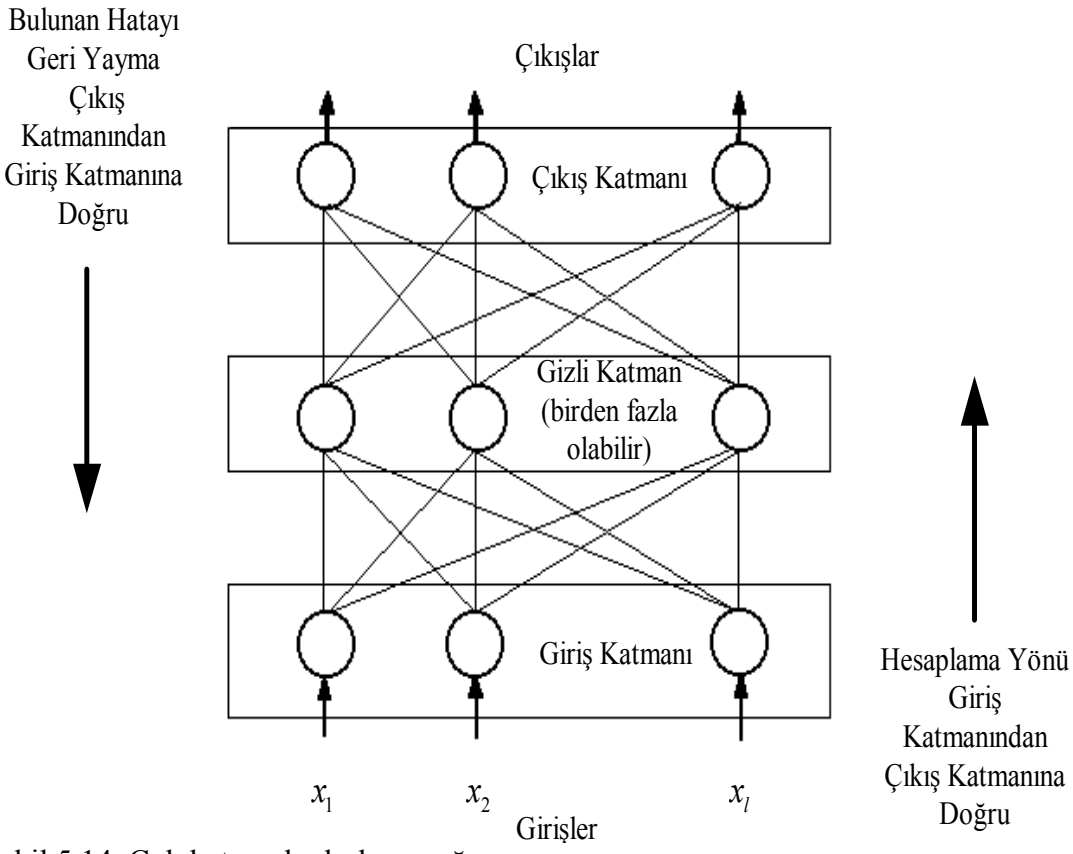
Widrow tarafından önerilen uyarlanabilir doğrusal eleman (ADALINE) her bir iterasyonda ortalama hata kareyi azaltarak ağırlıkları ayarlayan ve sınıflandırmayı sağlayan basit bir algılayıcıdır [Bishop, 1995]. ADALINE ağ yapısı Şekil 5.13'de gösterilmektedir. ADALINE ağ yapısı tüm sinir ağlarının en basitidir. ADALINE yapısı içindeki işlemci, giriş verilerini iki sınıfa ayırma yeteneği olan basit bir elemandır. Bu işlemci, öğrenme için danışmanlı öğrenmeyi kullanır. ADALINE birçok uygulama için oldukça iyi çalışmasına rağmen doğrusal problem uzayıyla sınırlıdır [Patterson, 1996] ve doğrusal aktivasyon fonksiyonunu kullanır. ADALINE eğitim kümesindeki veriler, doğrusal ayrıştırılabilir olmalıdır. Aksi takdirde ağın hatası en küçük değerine ulaştığında, ADALINE eğitim verilerini asla doğru olarak sınıflandıramayacaktır. Veriler doğrusal olarak uygun ise, giriş ve çıkış verilerinin tekrar tekrar ağa uygulanması ile eğitim gerçekleştirilir. Eğitim verilerinin doğru sınıflara ayrılmasıyla, öğrenme gerçekleştirilmiş olur. Eğitimden sonra ADALINE yeni gelen verileri kazandığı deneyime göre sınıflandırabilir.



Şekil 5.13. ADALINE ağ yapısı.

5.2.4. Çok katmanlı ağlar

Çok katmanlı sinir ağı yapısı, Şekil 5.14'de gösterilmiştir. Bu modelin yaygın bir şekilde kullanılmasının sebebi, birçok öğrenme algoritmasının bu ağı eğitmede kullanılabilir olmasıdır. Çok katmanlı bir sinir ağı, bir giriş, bir veya daha fazla gizli ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli bir sinir ağı yapısına uymaktadır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemdeki giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı ise, deneme-yanılma yolu ile veya tecrübeye dayalı olarak bulunur [Elmas, 2003]. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayalı olarak belirlenir.



Şekil 5.14. Çok katmanlı algılayıcı ağ yapısı

Çok katmanlı algılayıcı ağ yapısında, ağa bir örnek gösterilir ve örnek neticesinde nasıl bir sonuç üreteceği de bildirilir (danışmanlı öğrenme). Örnekler giriş katmanına uygulanır, gizli katmanlarda işlenir ve çıkış katmanından çıkışlar elde edilir. Kullanılan eğitme algoritmasına göre, ağın çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak hata minimuma düşünceye kadar ağın ağırlıkları değiştirilir.

5.3. Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağlarında işlem elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesine "ağın eğitilmesi" denir. Başlangıçta bu ağırlık değerleri rasgele olarak atanır. Yapay sinir ağları kendilerine örnekler gösterildikçe bu ağırlık değerlerini değiştirirler. Amaç ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Örnekler ağa defalarca (iterasyonlarda) gösterilerek en doğru ağırlık değerleri bulunmaya çalışılır. Ağın doğru ağırlık değerlerine ulaşması

örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine kavuşması işlemine "ağın öğrenmesi" denir. Ağırlık değerlerinin değişmesi belirli kurallara göre yürütülmektedir. Bu kurallara öğrenme kuralları denir [Sağıroğlu ve ark., 2003].

Yapay sinir ağlarında öğrenme olayının iki aşaması vardır. Birinci aşamada ağa gösterilen örnek için ağın üreteceği çıktı belirlenir. Bu çıktı değerinin doğruluk derecesine göre ikinci aşamada ağın bağlantılarının sahip olduğu ağırlıklar değiştirilir. Ağın çıktısının belirlenmesi ve ağırlıklarının değiştirilmesi öğrenme kuralına bağlı olarak farklı şekillerde olmaktadır. Ağın eğitimi tamamlandıktan sonra öğrenip öğrenemediğini (performansını) ölçmek için yeni verilerle denenmesine ağın test edilmesi denmektedir. Test etmek için ağın öğrenme sırasında görmediği örnekler kullanılır. Test etme sırasında ağın ağırlık değerleri değiştirilmez. Test örnekleri ağa gösterilir. Ağ, eğitim sırasında belirlenen bağlantı ağırlıklarını kullanarak görmediği bu örnekler için çıktılar üretir. Elde edilen çıktıların doğruluk değerleri ağın öğrenmesi hakkında bilgiler verir. Sonuçlar ne kadar iyi olursa eğitimin performansı o kadar iyi demektir. Eğitimde kullanılan örnek setine "eğitim seti", test için kullanılan sete ise "test seti" adı verilmektedir [Elmas, 2003].

5.3.1. Temel öğrenme kuralları

Literatürde kullanılan birçok öğrenme algoritması mevcuttur. Bu öğrenme algoritmalarının çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesinde kullanılır. Öğrenme işlemi çok parametreli, karmaşık ve matematiksel olarak ifade edilmesi zor bir işlemdir. Bugün kullanılan öğrenme kuralları bu işlemin basitleştirilmiş veya matematiksel olarak ifade edilmiş biçimleridir. Mevcut olan öğrenme algoritmalarının birçoğu Hebb, Delta, Kohonen ve Hopfield olmak üzere dört farklı öğrenme kuralından esinlenerek geliştirilmiştir. Bu kurallar aşağıda açıklanmıştır [Sağıroğlu ve ark., 2003].

Hebb kuralı

Bu kuralın temelinde, "bir nöron diğer bir nörondan giriş alıyorsa ve her iki nöronda aktif ise (matematiksel olarak aynı işarete sahip ise), nöronlar arasındaki ağırlık kuvvetlendirilir" düşüncesi yatmaktadır.

Hopfield kuralı

Bu kural zayıflatma veya kuvvetlendirme büyüklüğü dışında Hebb kuralına benzerdir. Eğer istenilen çıkış ve girişin her ikisi aktif veya her ikisi de aktif değilse, öğrenme oranı tarafından bağlantı ağırlığı artırılır, diğer durumlarda azaltılır.

Delta kuralı

Delta kuralı Hebb kuralının değişik bir formudur ve en çok kullanılan öğrenme algoritmalarından biridir. Bu kural, nöronun gerçek çıkışı ile istenilen çıkış değerleri arasındaki farkı azaltan bir yapıya sahiptir. Kural; ortalama hata karenin, bağlantı ağırlık değerlerinin değiştirilmesiyle (azaltıp veya artırma), küçültme prensibine dayanır. Hata, aynı anda bir katmandan önceki katmanlara geri yayılarak azaltılır. Ağın hatalarının düşürülmesi işlemi, çıkış katmanından giriş katmanına ulaşıncaya kadar devam eder. Bu kural, geri yayılım veya Widrow-Hoff öğrenme kuralı olarak da bilinir.

Delta-Bar-Delta kuralı

Delta-Bar-Delta (DBD) çok katmanlı algılayıcılarda bağlantı ağırlıklarının yakınsama hızını arttırmak için kullanılan bir sezgisel yaklaşımdır. DBD, Robert Jacobs tarafından, standart ileri beslemeli geri yayılım ağlarının öğrenme oranını iyileştirmek amacıyla geliştirilmiştir [Bishop, 1995].

DBD, her bir değerin kendine has yöntemlerle kendini uyarlayabilen katsayıya sahip olduğu bir öğrenme metodu kullanmaktadır. Ayrıca, geri yayılım mimarisindeki momentum katsayısını kullanmaz. Bu yöntemde eski hata değerleri hesaplanacak

hata değerlerini tahmin etmek için kullanılabilmektedir. Muhtemel hataları bilmek, sistemin ağırlık değerlerini ayarlamasına yardım edecektir. Ancak, her bir ağırlık değeri hatanın tamamı üzerinde çok farklı etkilere sahip olmaktadır. Jacobs bundan dolayı, geri yayılım öğrenme kurallarının hatanın tamamı üzerindeki etkilerin bu çeşitliliğini göz önüne alması gerektiğini belirten ortak duyu görüşünü ortaya atmıştır. Diğer bir deyişle, bir ağın bağlantı değerinin kendi öğrenme oranı olmalıdır. Dahası, bu öğrenme oranlarının zaman içinde değişmesini sağlamalıdır.

Bu algoritmada her bağlantı değerinin kendi öğrenme oranı vardır ve bu öğrenme oranları standart geri yayılım ile birlikte bulunan mevcut hataya bağlı olarak değişir. Bağlantı değeri değiştiğinde, eğer bölgesel hata çeşitli iterasyonlar için aynı değerlere sahipse, o bağlantının öğrenme oranı doğrusal olarak artırılır. Doğrusal olarak artırma, öğrenme oranlarının çok büyük ve çok hızlı hale gelmesini önler. Bölgesel hata değerleri sık sık değiştirdiğinde, öğrenme oranı geometrik olarak azaltılır. Geometrik olarak azaltma, bağlantı öğrenme oranlarının her zaman pozitif olmasını sağlar. Dahası, bu oranlar, hata değerindeki değişimlerin büyük olduğu bölgelerde daha hızlı bir şekilde azaltılabilir.

Kohonen kuralı

Bu kural, biyolojik sistemlerdeki öğrenmeden esinlenen Kohonen tarafından geliştirilmiştir. Bu kuralda, nöronlar öğrenmek için yarışır ve kazanan nöronun ağırlıkları güncellenir. En büyük çıkışa sahip işlemci nöron kazanır, bu nöron komşularını uyarma ve yasaklama kapasitesine sahiptir.

5.3.2. Sinir ağlarının öğrenme algoritmalarına göre sınıflandırılması

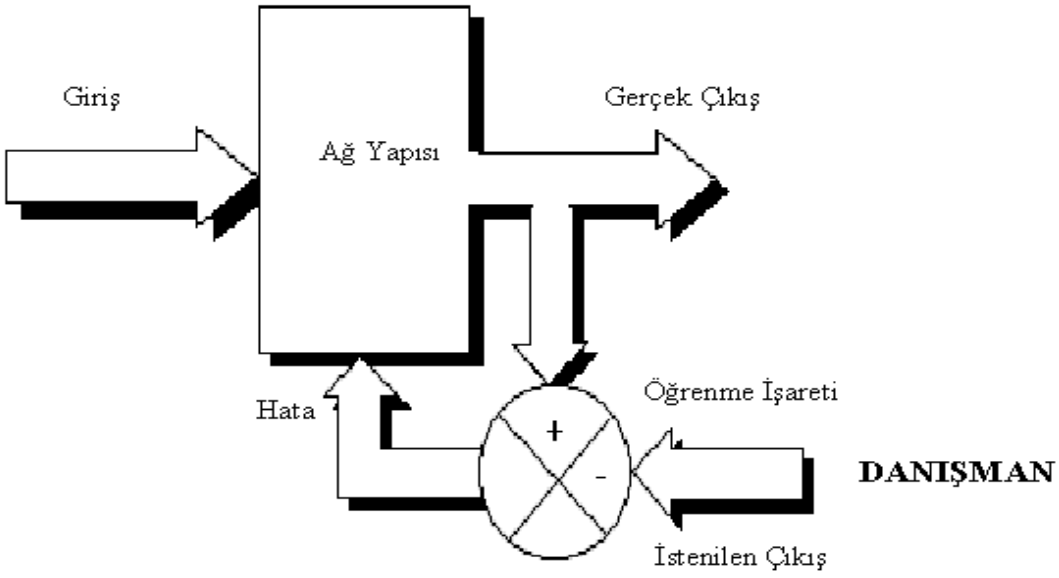
Yapay sinir ağlarında istenen sonucun elde edilmesi için ağın uyarlanabilir olması gerekir. Bunu sağlamak için uygun değerli ağırlıklar ve doğru bağlantılar seçilmelidir. Ağ bu şartları karşılayabilmek için sistemin davranışlarını öğrenmeli yada kendi kendini örgütlemelidir. Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. O halde,

birtakım yöntem ve kurallar ile gözlem ve eğitime göre ağdaki ağırlıkların değiştirilmesi sağlanmalıdır.

Ağların eğitimi için kullanılan öğrenme kuralları genellikle danışmanlı öğrenme (supervised learning), danışmansız öğrenme (unsupervised learning) ve takviyeli öğrenme (reinforcement learning) olmak üzere üç öğrenme yöntemi başlığı altında toplanabilir [Elmas, 2003].

Danışmanlı öğrenme

Danışmanlı öğrenme kuralları, arzu edilen ağ çıkışının elde edilebilmesi için, çıkış hatasının düşürülmesinde ağırlıkların uyarlanabilir hale getirilmesini gerektirir. Danışmanlı öğrenmede her giriş değeri için istenen çıkış sisteme tanıtılır ve sinir ağı giriş-çıkış ilişkisini gerçekleştirene kadar ağırlıkları aşama aşama ayarlar. Bu sebeple danışmanlı öğrenme algoritmasının bir "öğretmene" veya "danışmana" ihtiyacı vardır. Şekil 5.15'de danışmanlı öğrenme yapısı gösterilmiştir.

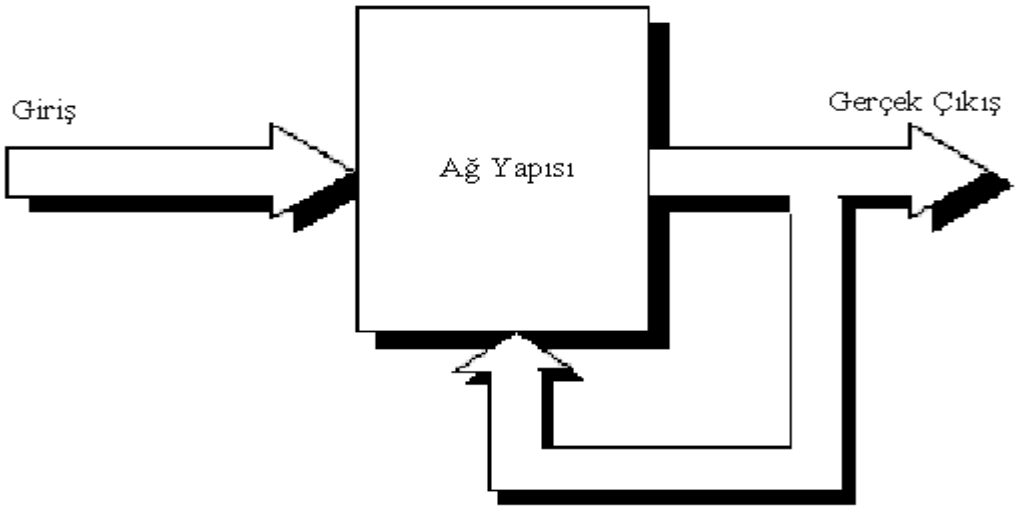


Şekil 5.15. Danışmanlı öğrenme yapısı

Widrow-Hoff tarafından geliştirilen Delta kuralı ve Rumelhart ve McClelland [Patterson, 1996] tarafından geliştirilen genelleştirilmiş Delta kuralı veya geri yayılım (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.

Danışmansız öğrenme

Danışmansız öğrenmede bir danışman veya öğretmen, sinir ağına girişin hangi veri parçası sınıfına ait olduğunu veya ağın nerede iyi sonuç vereceğini söylemez. Ağ, veriyi üyeleri birbirinin benzeri olan öbeklere (kümelere) yol göstermeksizin ayırır. Girişe verilen örnekten elde edilen çıkış bilgisine göre ağ sınıflandırma kurallarını kendi kendine geliştirmektedir.

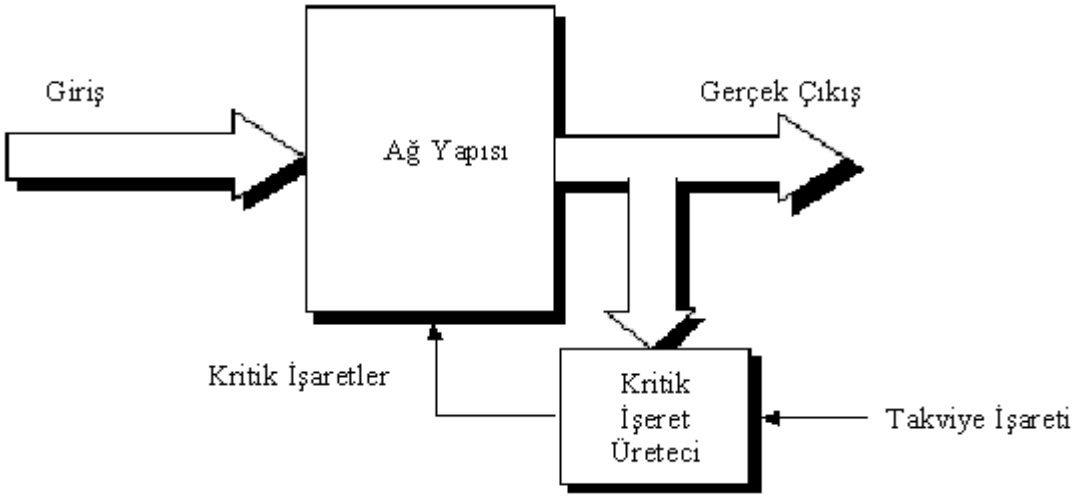


Şekil 5.16. Danışmansız öğrenme yapısı

Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri verilir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar. Şekil 5.16'da danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir [Sağıroğlu ve ark., 2003].

Takviyeli öğrenme

Bu öğrenme kuralı danışmanlı öğrenmeye yakın bir yöntemdir. Takviyeli öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Hedef çıktıyı vermek için bir "öğretmen" yerine, burada YSA'ya bir çıkış verilmemekte fakat elde edilen çıkışın verilen girişe karşılık iyiliğini değerlendiren bir kriter kullanılmaktadır. Şekil 5.17'de takviyeli öğrenme yapısı gösterilmiştir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı veya genetik algoritma takviyeli öğrenmeye örnek olarak verilebilir [Öztemel, 2003].



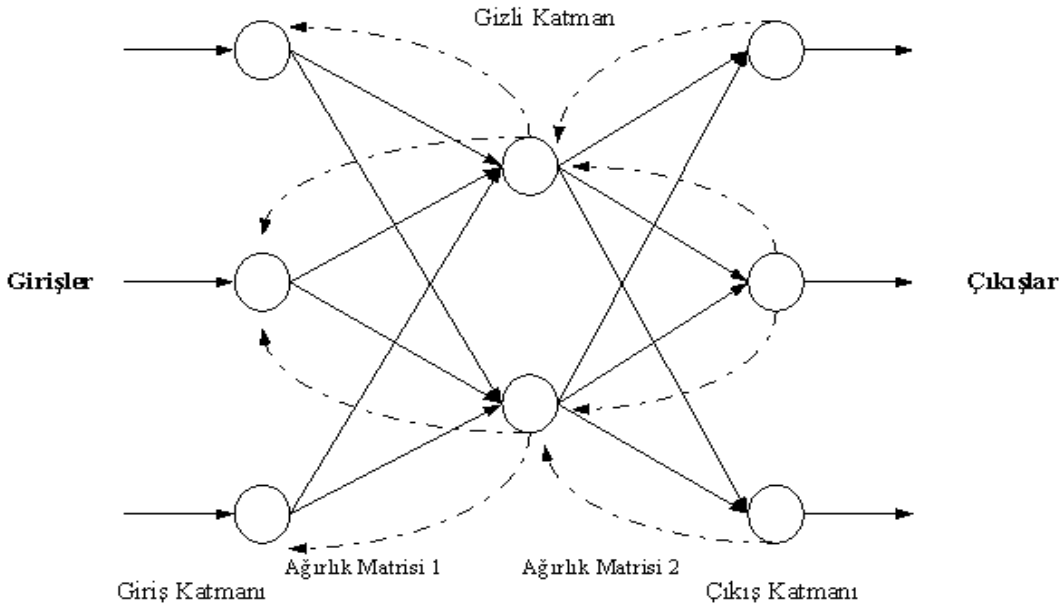
Şekil 5.17. Takviyeli öğrenme yapısı

5.3.3. Geri yayılım algoritması ve çok katmanlı ağın eğitiminde kullanılması

Birçok uygulamalarda kullanılmış en yaygın öğretme algoritmasıdır. Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğretme algoritmasıdır. Bu algoritma, hataları geriye doğru, çıkıştan girişe azaltmaya çalışması nedeniyle geri yayılım ismini almıştır.

Geri yayılım ağı, Geofrey Hinton ve James McClelland tarafından geliştirilmiştir [Bishop, 1995]. Geri yayılımlı öğrenen ağlar hiyerarşik yapıdadır. Giriş, çıkış ve en az bir gizli katman olmak üzere üç katmandan oluşurlar. Gizli katman ve gizli katmandaki düğüm sayısı değiştirilebilir. Düğüm sayısının artması ağın hatırlama

yeteneğini artırmakla birlikte öğrenme işleminin süresini uzatmaktadır. Düğüm sayısının azaltılması ise eğitim süresini kısaltmakta birlikte hatırlama yeteneğini azaltmaktadır. Giriş katmanındaki her bir düğüm gizli katmandaki her düğüme, gizli katman birden fazla ise bu katmandaki her bir düğüm kendisinden sonra gelen katmandaki her düğüme ve gizli katman çıkışındaki her düğüm çıkış katmanındaki her düğüme bağlıdır. Her katmanın çıkış değerleri bir sonraki katmanın giriş değeridir. Bu şekilde giriş değerlerinin ağın girişinden çıkışına doğru ilerlemesine ileri besleme denilir. Şekil 5.18'de bir geri yayılım ağı örneği görülmektedir.



Şekil 5.18. Geri yayılım ağ yapısı

Geri yayılım çok katmanlı ağlarda kullanılan Delta kuralı için genelleştirilmiş bir algoritmadır.

Geri yayılım ağında hatalar, ileri besleme aktivasyon işlevinin türevi tarafından, ileri besleme mekanizması içinde kullanılan aynı bağlantılar aracılığıyla, geriye doğru yayılmaktadır. Öğrenme işlemi, bu ağ yapısında basit çift yönlü hafıza birleştirmeye dayanmaktadır [Elmas, 2003].

Geri yayılım öğrenme yöntemi, türevi alınabilir aktivasyon fonksiyonlarını çok katmanlı herhangi bir ağa uygulayabilir. Delta kuralı gibi, geri yayılım algoritması da

sistem hatasını veya maliyet işlevini azaltma esasına dayanan bir en iyileme işlemidir. Bu yöntemde ağırlık ayarlamaları çıkıştan girişe doğru yapıldığı için "geri yayılım" ismi kullanılmıştır. Öğrenme esnasında, giriş örnekleri ağa belli bir sırada sunulur. Her bir veri çıktı hesaplanana kadar katman katman ileri yayılır. Hesaplanan çıktı daha sonra olması beklenenle karşılaştırılıp aradaki fark "hata" olarak bulunur.

Geri yayılım öğrenme algoritması kullanıldığında, sonraki katmanın hataları kullanılarak gizli katmanın ağırlıkları ayarlanır. Böylece çıkış katmanında hesaplanan hatalar son gizli katman ile çıkış katmanı arasındaki ağırlıklar ayarlanır. Aynı biçimde, bu işlemler ilk gizli katmana kadar tekrarlanır. Bu yolla hatalar katman katman ilgili katmanın ağırlık düzeltmeleri yapılarak geriye doğru yayılır. Çalışma süresi içinde "toplam hata" en aza indirilinceye kadar bu işlemler tekrarlanır.

Bir geri yayılımlı öğrenmede ağ değişkenleri için aşağıdaki ifadeler kullanılacaktır.

v_{ji}; i giriş katman siniri ile j gizli katman siniri arasındaki ağırlık bağlantısı ($i = 1, \ldots, n$, $j = 1, \ldots, h$).

w_{kj}; j gizli katman siniri ile k çıkış katmanı siniri arasındaki ağırlık bağlantısı ($k = 1, \ldots, m$).

x^p; p. birimin girdisi ($p = 1, \ldots, P$).

y_j^p; x^p girdisi için j gizli katman sinirinin çıktısı ($j = 1, \ldots, h$).

z_k^p; x^p girdisi için k çıktı katman sinirinin çıktısı ($k = 1, \ldots, m$).

t_k^p; k çıkış katmanında gerçekleşmesi beklenen/istenen çıktı ($k = 1, \ldots, m$).

$H_j = \sum_i v_{ij} x_i$; j gizli katmanının net girdisi ($j = 1, \ldots, h$).

$I_k = \sum_j w_{kj} y_j$; k çıktı katmanının net girdisi ($k = 1, \ldots, m$).

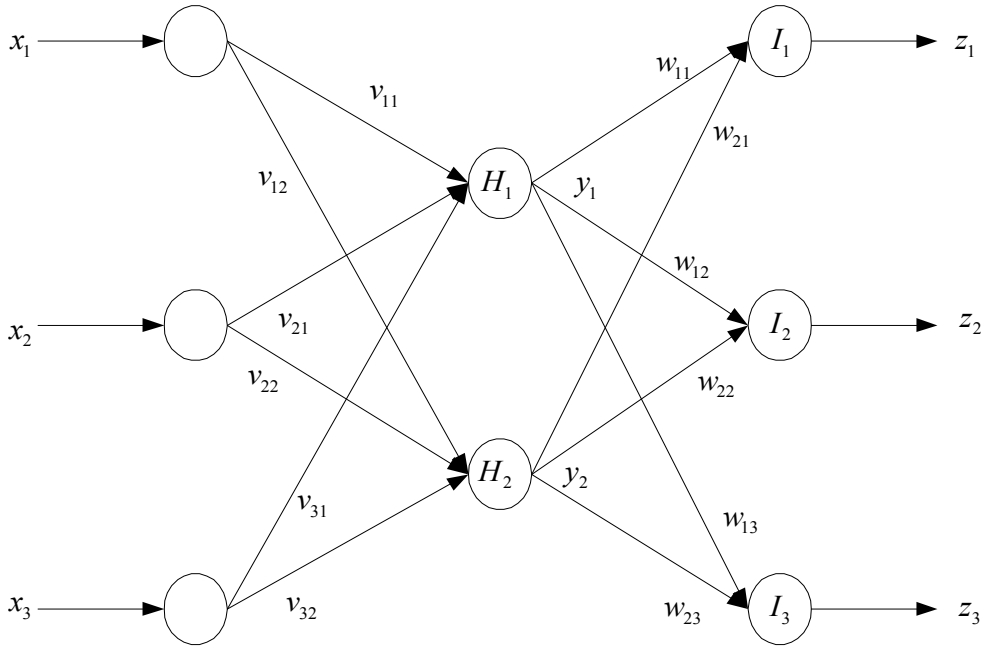
$y_j = f\left(H_j\right)$; j gizli katman sinirinde üretilen çıktı.

$z_k = f\left(I_k\right)$; k çıktı katmanında üretilen çıktı.

Burada, f keyfi, sınırlı ve türevlenebilir bir fonksiyondur. f fonksiyonu aracılığı ile, k çıktı birimi için x girdisine karşılık

$$z_k = f\left(I_k\right) = f\left(\sum_j w_{kj} y_j\right) = f\left(\sum_j w_{kj} f\left(H_j\right)\right) = f\left(\sum_j w_{kj} f\left(\sum_i v_{ji} x_i\right)\right) \tag{5.6}$$

sonuç elde edilir. Çıktıların elde edilmesi Şekil 5.19'da gösterilmektedir. Doğrusal olmayan f fonksiyonundan dolayı v_i, x girdi modelinin doğrusal olmayan bir fonksiyonudur. Burada sistem değişkenleri $W = (v, w)$'dir bir başka deyişle, $z = g(f, x, W)$'dir ve g; m boyutlu bir vektör işlevidir.



Şekil 5.19. Çok katmanlı ileri beslemeli bir ağın bağlantısı ve değişkenleri

Bu ağ için bir öğrenme algoritması gerçekleştirmek amacıyla, tüm modeller için hatayı azaltan bir yöntem gerekmektedir. Bu amaçla ortalama çıktı hatası tanımlanmaktadır. E_{ort}, ortalama ağ hatası, E^p, p modelinin çalışmasında karşılaşılan hata olmak üzere,

$$E_{ort} = \frac{E^1 + E^2 + \ldots + E^P}{P} = \frac{\sum_{p=1}^{P} E^p}{P} \tag{5.7}$$

olarak hesaplanır. Burada p sonlu veya sonsuz olabilir. Ancak gerçek hayatta örnek küme sonlu sayıdadır. Eğer herhangi bir p büyüklüğündeki çalışmadaki E^p hatası küçültülebilirse, sistemin hatası da azalacaktır. Böylece, Delta kuralında olduğu gibi, ağırlıkları hatadaki azalmaya göre ayarlayan bir ağırlık düzeltme yöntemi geliştirilebilir. Sonraki katmanda ağırlıklar öyle bir şekilde değiştirilecektir ki, katman hataları bir önceki değerinden daha düşük çıkacaktır. Bu durum, hata eğiminin negatif yönünde ağırlık belirleme işlemi ile başarılabilir. Böylece, $(s+1)$. adımdaki ağırlık ayarlaması E^p'nin s. adımdaki türevi ile orantılı olmaktadır. Bu durum

$$\Delta w(s+1) = -\eta \frac{\partial E^p}{\partial w(s)} \tag{5.8}$$

şeklinde ifade edilebilir. Burada, η öğrenme katsayısı yada öğrenme oranı olarak isimlendirilir. Buradan,

$$\frac{\partial E^p}{\partial w} = \left[\frac{\partial E^p}{\partial v_{11}}, \frac{\partial E^p}{\partial v_{12}}, \ldots, \frac{\partial E^p}{\partial w_{11}}, \ldots, \frac{\partial E^p}{\partial w_{hm}} \right] \tag{5.9}$$

elde edilir. Toplam sistem hatasının eğimi

$$\frac{\partial E_{ort}}{\partial w} = \frac{\sum_{p=1}^{P} \frac{\partial E^p}{\partial w}}{P} \tag{5.10}$$

olarak bulunur. E^p, ortalama hata kare, mutlak hata gibi birçok yol ile ifade edilebilir. p girdi modeli için ortalama hata kare

$$E^p = \frac{1}{2}\sum_{k=1}^{m}\left(t_k^p - z_k^p\right)^2 \tag{5.11}$$

olarak ifade edilmektedir. Burada $\frac{1}{2}$ faktörü, matematiksel uygunluk için konulmuştur [Elmas, 2003]. Eşitlik 5.8'deki ağırlık ayarlamasını ifade etmek için E^p'nin v_{ji} ve w_{kj} ağırlıklarına göre kısmi türevleri alınacaktır. Ayarlamalar sistem değişkenlerine (girdiler, hesaplanan çıktılar, ağırlıklar) göre yapılacağı için, zincir kuralının uygulanması gerekmektedir. Ağırlık ayarlamaları Eş. 5.11'e göre yapılmaktadır.

$$w_{kj}(s+1) = w_{kj}(s) + \Delta w_{kj} = -\eta \frac{\partial E^p}{\partial w_{kj}(s)} \tag{5.12}$$

Bu bağıntıda, $E^p = \frac{1}{2}\sum_{k=1}^{m}\left(t_k^p - z_k^p\right)^2$, $I_k = \sum_j w_{kj} y_j$ ve $z_k = f(I_k)$ ilişkileri kullanılmaktadır. İlk ağırlık güncelleştirmesine göre güncelleştirme kuralını bulabilmek için gerçek hatalar kullanılmaktadır.

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial I_k}\frac{\partial I_k}{\partial w_{kj}} = \frac{\partial E}{\partial I_k}\left(\frac{\partial}{w_{kj}}\sum_j y_j w_{kj}\right) \tag{5.13}$$

Eş. 5.13'deki parantez içindeki terim doğrudan hesaplanabilir. Böylece $\frac{\partial}{w_{kj}}\sum_j y_j w_{kj} = y_k$ olur. Zincir kuralı ile,

$$\frac{\partial E}{\partial I_k} = \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial I_k} = -\left(t_k - z_k\right) f^{'}\left(I_k\right) \tag{5.14}$$

elde edilir. $\delta_k = \left(t_k - z_k\right) f^{'}\left(I_k\right)$ bağıntısı yardımı ile güncelleme kuralı yeniden yazılırsa,

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \delta_k y_j \tag{5.15}$$

bulunur. Tek bir model için elde edilen kural tüm gizli ve çıkış sinir ağırlıklarına uygulanır.

Gizli katmanlarda, hedeflenen bir çıkış değeri yoktur ve çıkış sinirlerinin hataları gizli katman ağırlıklarının ayarlanmasında kullanılmaktadır. Zincir kuralı uygulanırsa,

$$\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}} = -\eta \frac{\partial E}{\partial H_j}\frac{\partial H_j}{\partial v_{ji}} \tag{5.16}$$

elde edilir. $H_j = \sum_i v_{ij} x_i$ kullanılarak,

$$\frac{\partial H_j}{\partial v_{ji}} = \sum_i \frac{\partial}{\partial v_{ji}}\left(v_{ji} x_i\right) = x_i \tag{5.17}$$

elde edilir. Eş. 5.16'daki ilk kısmi türevi elde etmek için zincir kuralı uygulanırsa,

$$\frac{\partial E}{\partial H_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial H_j} = \frac{\partial E}{\partial y_j} f^{'}\left(H_j\right) \tag{5.18}$$

elde edilir. $\partial E / \partial y_j$ türevi

$$\frac{\partial E}{\partial y_j} = \frac{1}{2}\sum_k \frac{\partial\left(t_k - f\left(\sum_j w_{kj} y_j\right)\right)^2}{\partial y_j}$$

$$= -\sum_k \left(t_k - z_k\right) f'\left(I_k\right) w_{kj} \tag{5.19}$$

şeklinde verildiği gibidir. Yapılan işlemler birleştirildiğinde gizli katman için güncelleştirme kuralı yazılabilir.

$$\Delta v_{ji} = \eta \delta_j x_i = \eta x_i f'\left(H_j\right) \sum_k \delta_k w_{kj} \tag{5.20}$$

Burada, $\delta_j = f'\left(H_j\right)\sum_k \delta_k w_{kj}$ ve $\delta_k = \left(t_k - z_k\right) f'\left(I_k\right)$'dır.

Özet olarak, iki katman için ağırlık güncelleştirme kuralları aşağıdaki şekildedir.

Çıkış birimleri için,

$\delta_k = \left(t_k - z_k\right) f'\left(I_k\right)$ olmak üzere,

$$\Delta w_{kj} = \eta \delta_k y_j = \eta \left(t_k - z_k\right) f'\left(I_k\right) y_j . \tag{5.21}$$

Gizli birimler için,

$\delta_j = f'\left(H_j\right)\sum_k \delta_k w_{kj}$ olmak üzere,

$$\Delta v_{ji} = \eta \delta_j x_i = \eta x_i f'\left(H_j\right) \sum_k \delta_k w_{kj} . \tag{5.22}$$

Geri yayılım öğrenme işleminin uygulanmasından önce tüm katmanlardaki bağlantı ağırlıklarına, küçük rasgele sayılar atanarak ağırlık matrisleri oluşturulur. Daha sonra x^p örnek vektörleri sisteme sunulur ve karşılık gelen z^p çıktıları hesaplanır. z^p hesaplanan çıktısı t^p arzu edilen çıktısı ile karşılaştırılır ve $\left(t_k^p - z_k^p\right)$, $k = 1, \ldots, m$ model hataları bulunur. Bu hatalar algoritmaya uygun şekilde geriye doğru yayılır.

Geri yayılım eşitlikleri daha yakından incelendiğinde, ileri doğru geçişlerde gizli katman sinirlerinden çıkan y_j sinyalleri karşı gelen k çıkış katmanındaki sinirlere bağlayan w_{kj} ağırlıkları ile çarpılır. k siniri bu değerleri toplar ve z_k çıkış etkinlik değerini hesaplar. Daha sonra $\left(t_k^p - z_k^p\right)$ hata değeri hesaplanır ve bu hata değerleri k sinirine bağlı w_{kj} ağırlıklarının her birini ayarlamak için "geri yayılım" işleminde kullanılır. Bu işlem, her bir $k = 1, \ldots, m$ birimi için çıkış katmanında

$$w_{kj}^{yeni} = w_{kj}^{eski} + \Delta w_{kj} = w_{kj}^{eski} + \eta y_j \left(t_k - z_k\right) f'\left(I_k\right) \tag{5.23}$$

eşitliğine uygun olarak gerçekleştirilir.

Gizli katmanların girdi ağırlıkları (v_{ji}) için hataların doğrudan hesaplanmasında hedef değerler yoktur. Bunun yerine, bu çıktı hataları kullanılarak girdi düğümlerini gizli katman düğümlerine bağlayan ağırlıklar ayarlanır ve bu hatalar uygun bir biçimde dağıtılır. Bunun için, δ_j değeri her bir j gizli birimi için hata olarak kullanılır.

$$v_{ji}^{yeni} = v_{ji}^{eski} + \Delta v_{ji} = v_{ji}^{eski} + \eta x_i f'\left(H_j\right) \sum_k \delta_k w_{kj} \tag{5.24}$$

Başlangıç değerleri kadar öğrenme ve momentum katsayılarının belirlenmesi ağın öğrenme performansı ile yakından ilgilidir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağın dolaşması söz konusu olabilir. Küçük değerler seçilmiş ise öğrenme zamanı artmaktadır. Genellikle 0,2–0,4 arasında bir öğrenme katsayısı kullanılmakla beraber, öğrenme katsayısının seçilmesi ilgilenilen probleme göre değişebilir. Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler. Momentum katsayısı bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesi olarak görülmektedir. Bu özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacı ile önerilmiştir. Bu değerin küçük olması yerel çözümlerden kurtulmayı zorlaştırılabilir. Çok büyük

değerler seçildiğinde ise tek bir çözüme ulaşmada sorunlar çıkabilir. Genellikle 0,6–0,8 arasında bir momentum katsayısı kullanılmakla beraber, momentum katsayısının seçilmesi de ilgilenilen probleme göre değişebilir

5.3.4. Levenberg-Marguardt yöntemi ile çok katmanlı ağın eğitilmesi

Levenberg-Marguardt algoritması kare toplamları olarak ifade edilen doğrusal olmayan fonksiyonların minimumun bulunmasında kullanılan iteratif bir tekniktir. Levenberg-Marguardt algoritması doğrusal olmayan en küçük kareler problemleri için bir standart tekniktir ve gradyent (eğim) azalması ve Newton yöntemlerinin bir kombinasyonu olarak düşünülmektedir [Hagan ve Menjah, 1994].

Newton öğrenme algoritmaları ya da yöntemleri Hessian matrisi yardımı ile çalışmaktadırlar. Hessian matrisi ağın performansı olarak adlandırılan ortalama hata kare fonksiyonunun ağırlık değişkenlerine göre ikinci türevlerinden oluşan bir matristir. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyent (eğim) değişimini göstermektedir. Performans fonksiyonu olarak, t_k^p; p. birimin x^p girdisine ilişkin k. çıkış katmanında gerçekleşmesi beklenen/istenen çıktı, z_k^p; p. birimin x^p girdisi için k. çıktı katman sinirinin çıktısı ($k = 1, \ldots, m$) olmak üzere $E^p = \frac{1}{2}\sum_{k=1}^{m}\left(t_k^p - z_k^p\right)^2$ hata kare fonksiyonu kullanıldığında, ortalama hata kare fonksiyonu

$$E_{ort} = \frac{E^1 + E^2 + \ldots + E^n}{n} = \frac{\sum_{p=1}^{n} E^p}{n}$$

olmaktadır ve Hessian matrisi

$$H = \frac{\partial^2 E}{\partial w_i \partial w_j}$$

olarak elde edilmektedir. Burada w daha önce de değinildiği gibi, genel olarak düğümler arasındaki bağlantıları simgelemektedir. $\nabla(f)$, fonksiyonun gradyenti olmak üzere, Hessian matrisinin tersi kullanılarak ağırlıklar değiştirilebilir.

$$w^{yeni} = w^{eski} - H^{-1}\left(x^{eski}\right)\nabla f\left(x^{eski}\right) \tag{5.25}$$

Ancak değişken boyutu büyüdükçe ve fonksiyon türevleri karmaşıklaştıkça Hessian matrisi ile işlem yapmak hayli güç olacaktır. Newton yöntemlerinin içinde, ikinci dereceden türevleri hesaplanmadan işlem yapılan bir sınıf vardır. Bu sınıftaki yöntemler quasi-Newton yöntemleri olarak adlandırılırlar ve bu yöntemler algoritmanın her iterasyonunda Hessian matrisinin yaklaşık bir şeklini kullanırlar.

Levenberg-Marguardt algoritması da quasi-Newton yöntemleri gibi, Hessian matrisinin yaklaşık değerlerini kullanır. Levenberg-Marguardt algoritması için Hessian matrisinin yaklaşık değeri, μ Marguardt parametresi ve I birim matris olmak üzere $H = J^t J + \mu I$ ile hesaplanabilir. J vektörü ise ağırlıklara göre birinci türevlerden oluşmaktadır, $J = \dfrac{\partial E}{\partial w_i}$. Ağın gradyenti $g = J^t E$ olarak hesaplanır ve ağırlıklar

$$w^{yeni} = w^{eski} - H^{-1} g \tag{5.26}$$

yardımı ile değiştirilir.

Marguardt parametresi, μ, skaler bir sayıdır. Eğer μ sıfırsa, bu yöntem yaklaşık Hessian matrisini kullanan Newton algoritması, eğer μ büyük bir sayı ise küçük adımlı eğim azalması haline gelir [Hagan ve Menjah, 1994].

6. GENETİK ALGORİTMALARIN ÇOK KATMANLI AĞIN EĞİTİMİNDE KULLANILMASI

Çok katmanlı sinir ağlarının eğitilmesinde hata fonksiyonunun gradyentine dayanan geri yayılım algoritması en çok kullanılan tekniktir. Ağın gerçekleşen çıktısı ve arzu edilen çıktısı arasındaki farka bağlı olarak tanımlanan hata fonksiyonu genellikle karmaşık yapıda ve birçok yerel minimuma sahip bir fonksiyondur. Gradyente bağlı yöntemlerin yerel bir çözüme takılma olasılıkları çok yüksek olduğundan hata fonksiyonu için en iyi çözümden uzakta bir çözüm elde edilebilir. Bu durum çok katmanlı ağ ile elde edilen sınıflandırma başarısını da olumsuz yönde etkileyecektir. Bu bölümde hata fonksiyonunun minimumunun araştırılmasında gradyent yöntemlerden çok daha etkin olan genetik algoritma yöntemi ele alınmakta ve genetik algoritmaların çok katmanlı ağların eğitilmesine adapte edilmesi incelenmektedir.

Genetik algoritmalar (GA) hakkındaki çalışmalar ilk olarak 1975 yılında, Michigan Üniversitesinde psikolog ve aynı zamanda bilgisayar uzmanı olan Holland (1975) tarafından yapılmıştır. Genetik algoritmalar uyarlanabilir sezgisel arama algoritmalarındandır ve doğal seçim mekaniğine ve doğal genetiğe bağlı arama yaparlar. Arama işlemi, canlıların biyolojik gelişimine benzemektedir. Canlılar, doğarlar ve üreyerek yeni nesillerini yetiştirirler. Doğal seçim mekaniğine bağlı olarak, canlı grubu açısından yapısı bakımından en uygun olan birey hayatta kalmaktadır. Genetik algoritmalar, genlerden oluşmuş canlılar gibi, 0 ve 1 bilgilerinden oluşmuş ikili düzendeki dizileri kullanmaktadırlar. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinden hayatta kalanların birleştirilmesi ile elde edilmektedir [Gen ve Cheng, 1996].

Bu yöntemde bireylerin popülasyonları ile işe başlanır, ikili dizinin temsil ettiği her birey parametre uzayı R^p'de bir noktayı temsil eder. Her nesilde her bir birey (çözüm) için amaç fonksiyonunun değeri onun uygunluğu olarak değerlendirilir ve daha uygun olan bireyler seçilerek yeni bir popülasyon elde edilir. Böylece bireylerin uygunluğuna dayalı yeni çözümler oluşturulur, yüksek uygunluk değerine sahip

bireyler daha sıklıkla seçildiği için daha uygun olan bireylerin popülasyona katılması yönünde baskı vardır. Birkaç nesil sonra en iyi bireyin optimal çözümü temsil etmesi umulur [Goldberg, 1989].

Genetik algoritmaları diğer yöntemlerden ayıran bazı temel özellikler vardır. Bunlar aşağıdaki şekilde özetlenebilir.

- Genetik algoritmalar çözüm değerlerini kodlayarak çalışır.
- Genetik algoritmalar tek bir nokta ile değil noktalar kümesi ile çalışır.
- Genetik algoritmalar amaç fonksiyonu bilgisini kullanır. Türev ya da başlangıç noktası gibi diğer yardımcı bilgileri kullanmaz.
- Genetik algoritmalar deterministik değil olasılıklıdır. Çünkü yüksek uygunluk değerine sahip bireylerin seçilme prensibine dayanmaktadır.

Genetik algoritmaların en yaygın olarak kullanıldığı problemler, geleneksel yöntemler ile çözümü mümkün olamayan ya da çözüm süresi problemin büyüklüğüne göre üstel şekilde artanlardır.

6.1. İkili Kodlamalı Genetik Algoritmalar

6.1.1. İkili kodlamalı genetik algoritmaların yapısı

Genetik algoritmaların çalışma prensibi aşağıdaki adımlarda özetlenmektedir.

Adım 1. Olası çözümlerin kodlandığı bir çözüm kümesi oluşturulur. Bu çözüm kümesi popülasyon olarak, çözümlerin kodları da kromozom olarak adlandırılır. İkili kodlama başta olmak üzere problemin türüne göre değişik kodlama şekilleri de mevcuttur.

Adım 2. Popülasyondaki her bir kromozomun uygunluk fonksiyonuna göre ne kadar "iyi" olduğu bulunur. Uygunluk fonksiyonuna göre iyi sonuçları veren kromozomlar,

yeni popülasyona alınmak üzere seçilirler. Problemin türüne göre geliştirilmiş birçok seçim mekanizması mevcuttur.

Adım 3. Seçilen kromozomlar eşlenerek, gen takası ve mutasyon operatörleri uygulanır. Bu sayede yeni bir popülasyon oluşturulur.

Adım 4. Tüm kromozomların uygunluk değerleri yeniden hesaplanır.

Adım 5. İstenilen nesil sayısına ulaşılmışsa ya da popülasyonda bir durağanlık sağlandıysa durulur ve o ana kadarki bulunmuş en iyi kromozom optimum sonuç olarak alınır. Aksi halde Adım 2'den itibaren adımlar tekrarlanır.

6.1.2. Kodlama

Genetik algoritmanın en büyük özelliği parametre değerlerinin kodlanmasıdır. İkili kodlama başta olmak üzere problemin türüne göre permütasyon kodlaması, değer kodlaması gibi değişik kodlama şekilleri de mevcuttur [Akan, 1997]. İkili kodlama en yaygın olarak kullanılan kodlama türüdür. Bu kodlama türünde kromozomlar yani bir popülasyondaki her bir birey genellikle sabit uzunluktaki bir ikili dizi (0 ve 1 şeklinde) olarak kodlanır. Dizinini uzunluğu, alan parametrelerine ve istenen kesinliğe bağlıdır. Fonksiyon optimizasyonunda değişkenlerin ikili kodlaması sıklıkla kullanılmaktadır. İkili düzende kodlanmış iki kromozom örneği Çizelge 6.1'de gösterilmektedir.

Çizelge 6.1. İkili düzende kodlanmış iki kromozom

Kromozom A	1000101010001011
Kromozom B	0001100101110010

Kromozom uzunluğunun 16 bit olduğunu ve x karar değişkeninin $[-10,10]$ aralığında değer aldığını varsayalım. Kromozomdaki değerlerin b_i olduğunu

varsayalım. Öncelikle, ikili dizi 2 tabanından 10 tabanına $x' = \sum_{i=0}^{15} b_i 2^i$ yardımı ile dönüştürülür. Örneğin ikili kodda verilen kromozom A'nın ondalık sistemdeki karşılığı $x' = (1)(2^0) + (1)(2^1) + (0)(2^2) + \ldots + (1)(2^{15}) = 35467$ olarak hesaplanır. x'e karşılık gelen gerçek değer ise $x = -10 + (x') \times \left[\frac{(10 - (-10))}{2^{16}} \right] = 0,8236$ olarak hesaplanmaktadır. Karar değişkeninin boyutu 1'den büyükse her bir parametre tek bir dizi halinde birleştirilebilir.

6.1.3. Popülasyon büyüklüğü

Popülasyon büyüklüğü için genel bir sayı olmamakla beraber, problemin karmaşıklığına ve aramanın derinliğine göre 200–300 aralığında alınması önerilmektedir [Akan, 1997].

6.1.4. Uygunluk fonksiyonu

Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir. Uygunluk fonksiyonu tüm amaçları içinde barındıran tek bir fonksiyon yapısındadır. Popülâsyonda bulunan kodlanmış kromozomlar gerçek değerlerine dönüştürülerek bu fonksiyonda yerine yazılır ve böylece her bir kromozomun uygunluğu hesaplanır. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunmasına değerlendirme denilmektedir. Bu fonksiyon genetik algoritmaların beynini oluşturmaktadır. Uygunluk fonksiyonu kromozomların şifresini çözer ve sonra hesaplama yaparak bu kromozomların uygunluğunu bulur [Holland, 1975].

6.1.5. Üreme ve seçim mekanizmaları

Üreme işlemi, daha geniş uygunluk değerlerine sahip dizilerin daha yüksek olasılık ile yeni nesilde geniş sayıda kopyalarını üretebilen işlemdir. Uygunluk değeri, ortalama değer ile normalize edilir. Üreme için diziler normalize edilmiş uygunluk

değerlerine göre seçilir. Böylece ortalama uygunluk değerinin üzerindeki diziler, ortalama uygunluk değerinin altındaki dizilerden daha fazla ürüne sahip olur. Uygunluk değerlerine göre dizileri kopyalamak, bir sonraki nesilde daha fazla ürünün oluşma olasılığının yüksek olması demektir. Doğal seçim, oluşturulan diziler arasından en uygun olanın kalması olarak tanımlandığında bu işlem, doğal seçimin yapay sürümü olmaktadır.

Kromozom seçilmesi yani üreme işlemi için geliştirilmiş değişik yöntemler vardır. Rulet tekerleği yöntemi en çok kullanılan seçim yöntemidir [Gen ve Cheng, 1996]. Bu teknikte, popülasyondaki her dizi için uygunluk değerleri ile orantılı olarak tekerlek üzerinde hisse verilir. Uygunluk değerine bağlı olarak her dizi tekerlek üzerinde belli bir yüzdeye sahip olur. Dizilerin yeterli sayıda üreme işlemleri için rulet tekerliği çevrilir. Diziler, uygunluk değerlerine göre hesaplanan olasılıklarla kopyalanırlar. Kopyası üretilen diziler eşleştirme havuzunda toplanarak diğer işlemlerin uygulanması için hazırlanırlar. Bir A_i dizisinin seçilme olasılığı $pseçim_i = \frac{f_i}{\sum f}$ olup, her A_i dizisi için beklenen dizi sayısı, $e_i = (n)(pseçim_i)$ olarak hesaplanmaktadır. e_i değerinin tamsayı kısmına göre dizilerin kopyaları oluşturulur. Beklenen dizi sayısının ondalıklı kısmı, rulet tekerliği seçim metodunda olasılık değeri olarak kullanılır. Böylece dizilerin gerekli sayıda kopyaları elde edilmiş olur [Akan, 1997].

6.1.6. Çaprazlama (Gen takası)

Çaprazlama işlemi, bilgilerin iki dizi arasındaki değişimi ile ilgilidir. Çaprazlama işleminde, üretilmiş iki yeni dizi eşleştirme havuzundan seçilerek rasgele seçilmiş çaprazlama pozisyonuna göre ikili dizilerdeki bilgi değişimi gerçekleştirilmektedir. Bu işlem, tercih edilmiş iyi dizilerin arasındaki daha iyi özellikleri birleştirir.

Çaprazlama işlemini gerçekleştirmek için ilk olarak, üreme işlemi ile oluşturulmuş eşleştirme havuzundaki yeni kopyalanmış dizinin elemanları rasgele eşlenir. İkinci

olarak, seçilen dizilerin bitleri, rasgele seçilmiş çaprazlama noktasından itibaren karşılıklı olarak değiştirilirler. Çaprazlama işleminin uygulanacağı dizinin l uzunluğunda olduğu kabul edilirse, çaprazlama pozisyonu, dizi boyunca 1 ile $(l-1)$ arasında rasgele seçilir ve seçilen bu pozisyon k ile ifade edilirse $(k+1)$ ve l pozisyonları arasındaki bütün bitlerin karşılıklı değişmesi ile oluşturulmaktadır. Tek nokta çaprazlama, iki nokta çaprazlama, tekdüze çaprazlama ve sıralı çaprazlama gibi çaprazlama yöntemleri mevcuttur.

6.1.7. Mutasyon

Mutasyon, üreme ve çaprazlama işlemlerinin tamamlayıcı işlemidir. Mutasyon işlemi, basitçe bit durumlarını tersine çevirmektir. Rasgele seçilen bit pozisyonundaki değerin ara sıra değişikliği sayesinde optimum noktaya ulaşma kuvvetlendirilmiş olur. Mutasyon işlemi, p_m olasılığı ile tek bir pozisyonun rasgele değişimi olup bu işlem oluşturulmuş nesilin elverişli durumunu birden bozabileceği için önemlidir. Bu yüzden, p_m olasılığı küçük tanımlanır [Akan, 1997].

6.1.8. Şema teoremi

Popülasyon içindeki dizilerin genel karakteristiği hakkında bilgi veren diziler şema olarak tanımlanmaktadır. Popülâsyon içindeki diziler, şemalar ile benzerlikleri açısından sınıflandırılmış olmaktadırlar. Şemalar, popülâsyon içinde genetik operatörlerin (üreme, çaprazlama, mutasyon) etkisini analiz etmek için kullanılırlar. Genel olarak H ile ifade edilen şema $\{0,1,\#\}$ elemanlarından oluşmaktadır. $\#$ sembolü, 0 veya 1 değerini alabilen önemsiz bit anlamına gelmektedir. l uzunluklu bir dizi 2^l tane şemayı temsil etmektedir. Buna göre n boyutlu bir dizi popülâsyonunda $(n)(2^l)$ adet şema bulunmaktadır.

Şemalar, şema derecesi ve şema uzunluğu özellikleri ile tanımlanmaktadır. H şemasının derecesi, şemadaki kesin bit pozisyon sayısı (0 ve 1 bilgisinin sayısı) olup

$O(H)$ ile ifade edilmektedir. H şemasının uzunluğu, ilk ve son kesin bit pozisyonları arasındaki uzaklık olup $\delta(H)$ ile gösterilmektedir. Örnek olarak, $A = 0111000$ dizisi için H şeması $H = \#1\#\#0\#\#$ olarak tanımlanmaktadır. H şeması için, son kesin bit pozisyonu 5, ilk kesin bit pozisyonu 2 olup, aralarındaki uzaklık şema uzunluğudur. Buna göre H şemasının uzunluğu $\delta(H) = 3$ olarak bulunur. H şemasının derecesi, 0 ve 1'lerin sayısı olup $O(H) = 2$ 'dir.

Diziler popülasyonunda, diziler üzerindeki genetik operatörlerin etkisi şema teoremi ile açıklanmaktadır. Şema teoremine göre üreme, çaprazlama ve mutasyon işlemlerinin etkisi açıklanmaktadır.

Popülasyondaki beklenen şema sayısı üzerinde üreme etkisini belirlemek kolaydır. t. nesilde $A(t)$ popülasyonu, A_j $(j = 1, 2, \ldots, n)$ dizilerinden oluşmuştur. t nesli için $A(t)$ popülasyonu tarafından içerilen H dizisinin m tane örneği vardır. Bu $m = m(H, t)$ olarak ifade edilebilmektedir. Farklı t nesilleri için farklı H dizileri farklı miktarlarda vardır. Üreme süresince bir dizi kendi uygunluk değerine göre kopyalanır veya popülasyon içerisindeki bir A_i dizisi

$$p_i = f_i / \sum f_i \tag{6.1}$$

olasılığı ile seçilir. f_i, A_i dizisinin uygunluk değerini, $\sum f_i$, $A(t)$ popülâsyonundaki dizilerin toplam uygunluk değerini ifade etmektedir. n boyutlu $A(t)$ popülasyonunun $(t+1)$ nesli için içerdiği H şemasının karakteristiğinin $m = m(H, t+1)$ olması beklenir. Bu eşitlik

$$m(H, t+1) = m(H, t) \times n \times f(H) / \sum f_j \tag{6.2}$$

şeklinde yazılır. $f(H)$, t nesline ait H şeması ile gösterilen dizinin ortalama uygunluk değeridir. Popülasyonun ortalama uygunluk değeri $\overline{f} = \sum f_j / n$ şeklinde yazıldığında kopya ile çoğalan şemanın büyüme eşitliği

$$m(H, t+1) = m(H, t) \times f(H) / \overline{f} \tag{6.3}$$

olarak yazılmaktadır. Eş. 6.3'den görüldüğü gibi şemanın ortalama uygunluk değerinin popülasyonun ortalama uygunluk değerine oranı kadar şema büyür. Popülasyon ortalamasının altındaki uygunluk değeri ile şema azalan sayıda örnek alırken, popülasyon ortalamasının üstündeki uygunluk değeri ile şema sonraki nesilde artan sayıda örnekler almaktadır. Yalnız üreme işlemi altında dizi oranlarına göre bütün şemalar büyür veya küçülürler.

Çaprazlama işlemi sonucunda şemanın hayatta kalma olasılığı $p_s = 1 - \delta(H)/(l-1)$'dir. Çaprazlama işleminden sonra şemanın hayatta kalması demek çaprazlama pozisyonunun şema uzunluğunun dışında seçilmesi anlamına gelir. Bir pozisyonun belirlenmesinde kullanılan çaprazlama olasılığı p_c olarak tanımlanırsa, çaprazlama işlemi sonucu hayatta kalma olasılığı $p_s \geq 1 - p_c \delta(H)/(l-1)$ eşitsizliği ile ifade edilir. Genellikle çaprazlama olasılığı p_c, %50'den daha büyük seçilerek diziler arasında bit değişimi yeterince sağlanmaktadır.

Çaprazlama işleminin etkisini incelemek amacıyla uzunluğu $l = 7$ olan A dizisi ve bu diziyi temsil eden iki şema aşağıda tanımlanmaktadır.

$A = 0111000$

$H_1 = \#1\#\#\#\#0$

$H_2 = \#\#\#10\#\#$

Çaprazlama pozisyonunun 3 olduğu kabul edildiğinde H_1 ve H_2 şemaları için çaprazlama işlemi dördüncü bitlerden itibaren gerçekleşecektir. Buna göre çaprazlama işlemi sonucunda yeni şemalar aşağıdaki şekilde elde edilir.

$H_1 = \#1\#10\#\#$

$H_2 = \#\#\#\#\#\#0$

Burada görüldüğü gibi H_1 şemasının çaprazlama sonucunda bozulmadan hayatta kalma şansı H_2 şemasından daha azdır. H_1 şeması $p_d = \delta(H_1)/(l-1) = 5/6$ olasılığı ile bozulmaktadır. Bu durumda $p_s = 1 - p_d = 1/6$ olasılığı ile hayatta kalacaktır. H_2 şeması $p_d = \delta(H_2)/(l-1) = 1/6$ olasılığı ile bozulmaktadır. Bu durumda, $p_s = 1 - p_d = 5/6$ olasılığı ile hayatta kalacaktır.

Çaprazlama işlemi ile $(t+1)$ nesil sonraki beklenen şema sayısı

$$m(H, t+1) \geq m(H, t) \times f(H) \Big/ \overline{f} \left[1 - p_c \frac{\delta(H)}{(l-1)} \right] \tag{6.4}$$

şeklinde ifade edilmektedir.

Üreme ve çaprazlama işlemlerinin birleştirilmiş etkisi, dizinin popülasyon ortalamasının üstünde olup olmadığı ve dizinin boyutu dizilerin çeşitlendirilmesinde önem taşımaktadır.

Mutasyon işleminde bir H şemasının hayatta kalması, kesin pozisyondaki bitlerin hayatta kalması anlamına gelmektedir. Bunun için kesin pozisyondaki bir bitin $(1 - p_m)$ olasılıkla hayatta kalması gerekmektedir. Buradan hareketle, şemanın hayatta kalma olasılığı $(1 - p_m)^{o(H)}$ olarak ifade edilir.

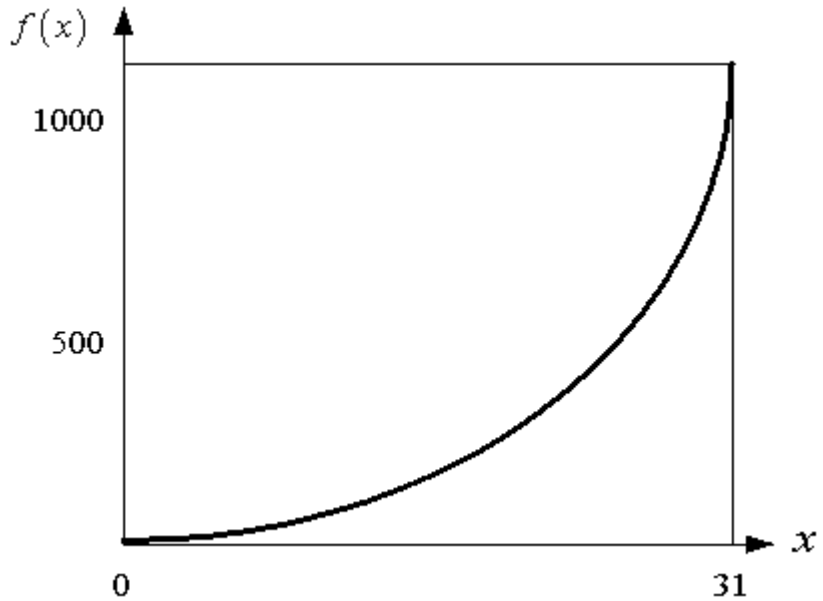
Sonuç olarak, H şemasının üreme, çaprazlama ve mutasyon işlemlerinden sonra $(t+1)$ nesil sonra beklenen sayısı

$$m(H,t+1) \geq m(H,t) \times f(H) \Big/ \overline{f} \left[1 - p_c \frac{\delta(H)}{(l-1)} - (1-p_m)^{o(H)} \right] \tag{6.5}$$

ile ifade edilmektedir. Uygunluğu yüksek olan kromozomların bir sonraki nesilde var olma ve çoğalma olasılığının, uygunluğu düşük olan kromozomlara göre daha fazla olduğu söylenilebilir [Holland, 1975].

6.1.9. İkili kodlamalı genetik algoritmalar kullanılarak bir optimizasyon probleminin eniyilenmesi örneği

Goldberg (1989) tarafından da örnek olarak verilen, $[0,31]$ tamsayı aralığında x^2 fonksiyonunun maksimumunu genetik algoritma yöntemiyle araştıralım. $[0,31]$ aralığında $f(x) = x^2$ fonksiyonunun grafiği Şekil 6.1'de yer almaktadır.



Şekil 6.1. $[0,31]$ tamsayı aralığında x^2 fonksiyonunun grafiği

Genetik algoritma ile optimizasyon işleminin ilk adımı, x parametresini sonlu uzunluklu dizi olarak kodlamaktır. Bu problem için x uzunluğu 5 olan ikili tamsayı olarak basitçe kodlanabilir.

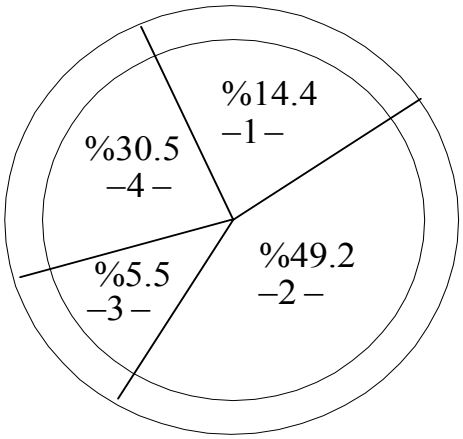
Başlangıç olarak, ilk popülasyon rasgele seçilir. Madeni paranın 20 kere atılması ile rasgele 4 boyutlu bir popülasyon seçilir (tura 0, yazı 1). Başlangıç popülasyonu Çizelge 6.2'de görülmektedir.

Genetik algoritma üreme işlemi ile işe başlar. Üreme işlemi için rulet tekerleği yönteminin kullanıldığını varsayalım. Bir numaralı dizinin uygunluk değeri, 169 olup toplam uygunluk değerinin %14,4'üdür. Birinci dizi rulet tekerleğinin %14,4'ü olarak verilmiş olup her devirde 0,144 olasılıkla çevrilir. Uygunluk değerine göre üreme işlemi için kullanılacak rulet tekerleği Şekil 6.2'de gösterilmektedir.

Çizelge 6.2. Örnek için başlangıç popülasyonu

No	*Diziler*	*x*	*Uygunluk Değerleri*	*%*
1	01101	13	169	14,4
2	11000	24	576	49,2
3	01000	8	64	5,5
4	10011	19	361	30,9
Toplam			1170	100,0

Sonraki nesilin eşleştirme havuzu Şekil 6.2.'deki rulet tekerleğinin dört defa çevrilmesi ile oluşturulur. Dizilerin seçilme olasılıkları $pseçim_i = \frac{f_i}{\sum f}$ ile hesaplanır. Çizelge 6.3'de ($pseçim_i$) sütununda her dizinin seçilme olasılık değerleri gösterilmiştir. Rulet tekerleğinin dört kere çevrilmesi sonucunda her dizi için beklenen kopya sayısı $e_i = (n)(pseçim_i)$ olup bu işlem sonucunda birinci ve dördüncü dizilerin birer kopyaları, ikinci dizinin iki kopyası ve üçüncü dizinin ise sıfır kopyası oluşmuştur.



Şekil 6.2. Üreme işleminde kullanılan rulet tekerleği

Çizelge 6.3. Örnek için üreme işlemi

No	*Rasgele Üretilmiş İlk Popülasyon*	*x*	$f(x) = x^2$	$pseçim_i$	*Beklenen Sayı*	*Kopya Sayısı*
1	01101	13	169	0,14	0,58	1
2	11000	24	576	0,49	1,97	2
3	01000	8	64	0,06	0,22	0
4	10011	19	361	0,31	1,23	1
Toplam			1170	1,00	4,0	4.0
Ortalama			293	0,25	1,00	1.0
Maksimum			**576**	0,49	1,97	2.0

Üreme işleminden sonra çaprazlama işlemi iki adımda devam eder.

1. Eşleştirme havuzundaki yeni kopyalanmış diziler rasgele eşlenirler.

2. Dizi çiftlerinin karşılıklı bit değişimi için rasgele çaprazlama pozisyonu seçilir.

Üreme işlemi sonucunda, kopyası oluşan diziler eşleştirme havuzunda çaprazlama işlemine tabi tutulmak için beklerler. Hangi dizilerin eşleneceği rasgele seçilir. Çizelge 6.4'te ikinci sütunda belirtildiğine göre birinci dizi ikinci dizi ile üçüncü dizi dördüncü dizi ile eşlenecektir.

Çizelge 6.4. Örnek için çaprazlama işlemi

Üreme Sonucu Eşleştirme Havuzu	*Rasgele Seçilmiş Eşleştirme*	*Rasgele Seçilmiş Çaprazlama Pozisyonu*	*Yeni Popülasyon*	x	$f(x) = x^2$
0110↕1	2	4	01100	12	144
1100↕0	1	4	11001	25	625
11↕000	4	2	11011	27	729
10↕011	3	2	10000	16	256
Toplam					1754
Ortalama					439
Maksimum					**729**

Çaprazlama işlemi için diziler bu şekilde eşlendikten sonra çaprazlama pozisyonlarının rasgele belirlenmesi gerekmektedir. Çizelge 6.4'te üçüncü sütunda çaprazlama pozisyonları verilmiştir. Bu değerlere göre birinci ve ikinci diziler için çaprazlama pozisyonu 4, üçüncü ve dördüncü diziler için çaprazlama pozisyonu 2 olarak seçilmiştir. Bu değerlere göre birinci ve ikinci diziler için çaprazlama işlemi aşağıdaki gibi gerçekleşir.

Dizi uzunluğu $l = 5$, çaprazlama pozisyonu $k = 4$ olduğuna göre çaprazlama işlemi $(k+1) = 5$ ve $l = 5$ pozisyonlarındaki bitler arasında olacaktır. Bu durumda çaprazlama pozisyonunu ↕ sembolü ile gösterilirse, ↕'in sağ tarafında kalan koyu olarak belirtilmiş bitler karşılıklı olarak yer değiştirirler.

Birinci dizi: 0110↕**1**

İkinci dizi: 1100↕**0**

Çaprazlama işleminden sonra oluşan iki dizi aşağıda gösterilmiştir.

Birinci yeni dizi: 0110**0**

İkinci yeni dizi: 1100**1**

Aynı işlemler üçüncü ve dördüncü diziler için aşağıdaki şekilde tekrarlanır.

Dizi uzunluğu $l = 5$, çaprazlama pozisyonu $k = 2$ olduğuna göre çaprazlama işlemi $(k+1) = 3$ ve $l = 5$ pozisyonlarındaki bitler arasında olacaktır. Bu durumda çaprazlama pozisyonunu ↕ sembolü ile gösterilirse, ↕'in sağ tarafında kalan koyu olarak belirtilmiş bitler karşılıklı olarak yer değiştirirler.

Üçüncü dizi: 11↕**100**

Dördüncü dizi: 10↕**011**

Çaprazlama işleminden sonra oluşan iki dizi aşağıda gösterilmiştir.

Üçüncü yeni dizi: 11**011**
Dördüncü yeni dizi: 10**000**

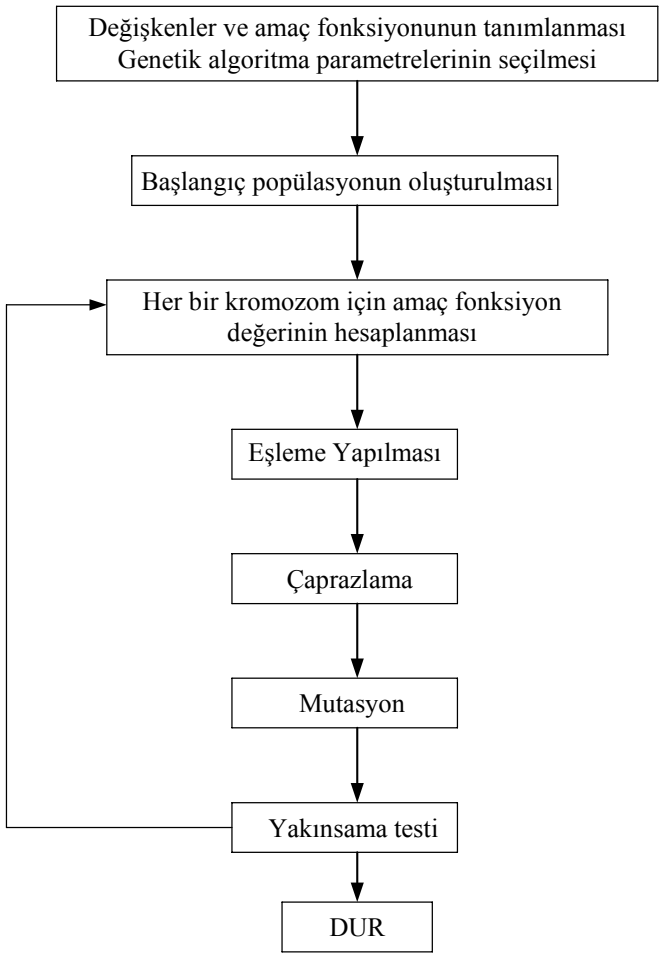
Son işlem, mutasyon işlemi olup, bit-bit işlem esasına dayanır. Örnek için değişim olasılığı örneğin p_m'in 0,001 alınırsa, her biri beş bitten oluşan dört dizi için $(5)(4)(0,001) = 0,02$ bitin değişime uğrayacağı beklenir. Bu olasılık değeri için bu adımda, karşılıklı bit değişimi haricinde, hiçbir bit 0'dan 1'e veya 1'den 0'a değişmemiştir.

Örnek için sonuçlar Çizelge 6.4'ün son sütunlarında gösterilmiştir. Maksimum ve ortalama değerler yeni popülasyonda artmıştır. Sadece bir nesilde popülasyon ortalama uygunluk değeri 293'ten 439'a artmıştır. Maksimum uygunluk değeri 576'dan 729'a çıkmıştır. İlk nesilin en iyi dizisinin (11000) iki kopyası vardır. Çünkü en yüksek ortalama uygunluk değerine sahiptir. Sonraki en iyi dizi (10011) ile rasgele birleştiğinde sonuç dizilerden biri (11011) gerçekten çok iyi seçim olduğunu kanıtlar.

6.2. Sürekli Parametreli Genetik Algoritmalar

Genetik algoritmanın ikili kodlamasında parametrelerin 0 ve 1 kullanarak ifade edilmesi, kromozomların boyutlarını oldukça artırmaktadır. Bu yüzden ikili kodlamalı genetik algoritmalar sınırlı hassasiyete sahiptir. Bunun yerine gerçek rakamlarla kodlama yapabilen sürekli parametreli genetik algoritmaları kullanmak avantajlıdır. Sürekli parametreli genetik algoritmalar hem daha hassas hem de bilgisayar belleğinde daha az yer kaplamaktadırlar [Haupt ve Haupt, 2004]. Sürekli parametreli genetik algoritmaların akış diyagramı Şekil 6.3'de gösterilmektedir.

Sürekli parametreli genetik algoritmalar gerçek değer kodlu genetik algoritmalar olarak da bilinmektedir [Şen, 2004b].



Şekil 6.3. Sürekli parametreli genetik algoritmaların akış diyagramı

6.2.1. Sürekli parametreli genetik algoritmaların bileşenleri

Sürekli parametreli genetik algoritmalar ikili kodlu genetik algoritmalara çok benzemektedirler. Aralarındaki en önemli farklılık parametreleri 0 ve 1'lerle temsil edilmemekte bunların yerine gerçek değerli sayılarla ifade edilmektedir.

Eğer $d_1, d_2, \ldots, d_{N_{değ}}$ olmak üzere $N_{değ}$ sayıda değişken varsa bunlar $1 \times N_{değ}$ boyutlu kromozom olarak,

$$K = \left[d_1,\ d_2,\ \ldots,\ d_{N_{değ}} \right] \tag{6.6}$$

şeklinde yazılır. Burada değişkenlerin her biri ondalıklı sayı olarak ifade edilir. Her bir kromozom için amaç fonksiyonunun değeri, H amaç fonksiyonu olmak üzere,

$$H = h(K) = h\left(d_1,\ d_2,\ \ldots,\ d_{N_{değ}} \right) \tag{6.7}$$

ile ifade edilir. Kromozomlar ile ifade edilen çözümlerin amaç fonksiyon değerleri hesaplanır.

6.2.2. Parametrelerin kodlanması, doğruluğu ve sınırları

İkili kodlu genetik algoritmalarda olduğu gibi sürekli parametreli genetik algoritmalarda da değişkenlerin temsili için kaç hane gerektiği düşünülmez. Değişkenler, değişim aralıkları sınırları arasında düşen sayılarla ondalık olarak temsil edilir. Her ne kadar ondalık sayı sistemine göre değişkenler herhangi bir sayıyı kabul ederlerse de bunların haneleri bilgisayar işlemcisinin haneleri ile temsil edilir. Burada bilgisayarın kendi işlemcisinde bulunan iki tabanlı sayılardan yararlanarak

ondalık değişkenleri en ince hassasiyete dönüştürürler. Artık değişkenlerin hassasiyeti bilgisayarların yuvarlama hatası kadardır. Tek hassasiyetli hesaplamalarda $2^4 = 16$ haneli ikili sayı sistemini, çift hassasiyetli hesaplamalarda $2^5 = 32$ haneli ikili sayı sistemini kullanırlar [Haupt ve Haupt, 2004].

Genetik algoritmalar arama teknikleri olduklarından, her değişkenin sınırlarının iyi belirlenmesi gerekir. Sınırların bilinmemesi durumunda değişken aralıkları mümkün olduğunca geniş tutularak çözümler araştırılır.

6.2.3. Başlangıç popülasyonu

Genetik algoritmalar ile arama işleminin başlatılabilmesi için içinde N_{bpop} kadar öğesi bulunan kromozomlar seçilmelidir. Bunun için her satırında $1 \times N_{değ}$ boyutunda bir kromozom olan N_{bpop} satırlı bir matris göz önüne alınır. Başlangıçta, N_{bpop} kadar popülasyon kromozomun verilmesi ile rasgele sayılar üretilerek $N_{bpop} \times N_{değ}$ boyutunda matris

$$BPOP = \left(d_{eb} - d_{ek}\right)\left\{rand\left[N_{bpop}, N_{değ}\right]\right\} + d_{ek} \tag{6.8}$$

kuralına göre kurulur. Eş. 6.8'de, $BPOP$ başlangıç popülasyonunu, d_{ek} ve d_{eb} sırası ile değişkenlerin en küçük ve en büyük değerlerini, $rand\left[N_{bpop}, N_{değ}\right]$ ise $N_{bpop} \times N_{değ}$ boyutunda değerleri 0 ve 1 arasında bulunan düzgün dağılımdan gelen rasgele değerleri gösterir. Başlangıç popülasyonundaki her bir satır amaç fonksiyonunu iyileştirmeye aday bir kromozomu gösterir. Başlangıç popülasyonunun sütunları ise değişkenlere karşılık gelmektedir. Kromozomlar yani aday çözümlerin hepsi eşit önemde değildirler. Kromozomların her birinin önemi amaç fonksiyon değeri miktarına göredir [Şen, 2004b].

6.2.4. Doğal seçim

Bu adımda başlangıç kromozomlarından hangisinin yeterli derecede kuvvetli olarak bir sonraki popülasyona geçeceğine karar verilir. Bunun için önce N_{bpop} tane kromozom amaç fonksiyon değerlerine göre en büyükten en küçüğe veya en küçükten en büyüğe doğru sıralanır (amaç fonksiyonunun maksimum veya minimum olmasına göre). N_{top} kromozom sayısı olmak üzere, bunlardan ilk N_{top} kadarı sonraki popülasyona yani nesile intikal ettirilir. Bunların dışındakilerin popülasyondan dışlanarak öldükleri varsayılır. Amaç fonksiyon değerini iyileştirecek kromozomların hayatta kalabilmesi yani uygun çözümlerin elde edilebilmesi için, bu işlem aramanın sonraki tüm iterasyonlarında (popülasyonlarında) uygulanmalıdır. Bundan sonraki tüm iterasyon geçişlerinde kromozom sayısı N_{top} olarak sabit alınır. Kalan N_{top} sayıdaki kromozomların yarısı bundan sonra yapılacak eşleme ve çaprazlama işlemlerine tabi tutulur. N_{top} sayıdaki kromozom kendi arasında N_{iyi} ve $N_{\text{kötü}}$ olarak büyükten küçüğe doğru sıralanır. N_{iyi} kadar kromozom eşleme ve çaprazlama işlemi için bırakılırken $N_{\text{kötü}}$ kadarı atılır.

6.2.5. Eşleme ve çaprazlama

Eşleme için ikili kodlu genetik algoritmalarda da kullanılan rulet tekerleği seçim yöntemi kullanılmaktadır. Çaprazlama için çeşitli yaklaşımlar geliştirilmiştir [Şen, 2004b]. Kromozomlar arasından N_{iyi} kadarı seçildikten sonra, bunların arasından yarısı anne yarısı da baba olarak itibar edilecek kromozom rasgele olarak seçilir ve çaprazlanır. Her çift kendilerinden kalıntıları olan iki yeni çifti meydana getirir. Anne ve baba kromozomları birbirlerine ne kadar benzerlerse bunlardan doğan kromozomlar da birbirine daha fazla benzer (çözümler tekdüze olmaya başlar). Kromozomları parçalayarak yeni çocuklar (yeni çözümler) meydana getirmek için değişik yöntemler vardır. Bu yöntemler ikili kodlu genetik algoritmalarda verilmiştir.

İkili kodlu genetik algoritmalarda olduğu gibi, sürekli parametreli genetik algoritmalarda da iki kromozomun bir araya gelmesi ile ortaya çıkan yeni iki kromozomda kısmen önceki kromozomların kalıntılarına sahip olacaktır [Şen, 2004b]. Sürekli parametreli genetik algoritma yönteminde çaprazlama noktalarının tespiti için değişik yaklaşımlar vardır. En basit yaklaşım, kromozomda bir veya daha fazla noktayı çaprazlama noktası olarak almak ve bu noktalar arasındaki parametreleri anne ve baba kromozomları arasında değiş tokuş ettirmektir. Örnek olarak birbirinden farklı kromozomlar

$$Eş_1 = \left[d_{a1},\ d_{a2},\ d_{a3},\ d_{a4},\ d_{a5},\ \ldots,\ d_{aNdeğ} \right]$$

ve

$$Eş_2 = \left[d_{b1},\ d_{b2},\ d_{b3},\ d_{b4},\ d_{b5},\ \ldots,\ d_{bNdeğ} \right]$$

olsun. a alt indisli eş anne kromozomu, b indisli eş ise baba kromozomu göstersin. Çaprazlama noktaları ise rasgele olarak ikinci ve dördüncü konumlar olsun. Çaprazlama yapıldıktan sonra çocuk kromozomlar,

$$Çocuk_1 = \left[d_{a1},\ d_{a2},\ \mathbf{d_{b3}},\ \mathbf{d_{b4}},\ d_{a5},\ \ldots,\ d_{aNdeğ} \right]$$

ve

$$Çocuk_2 = \left[d_{b1},\ d_{b2},\ \mathbf{d_{a3}},\ \mathbf{d_{a4}},\ d_{b5},\ \ldots,\ d_{bNdeğ} \right]$$

elde edilir. Bu uygulama düzgün çaprazlama olarak isimlendirilir. Burada görüleceği gibi parametrelerin değerleri değişmemektedir. Parametreler sadece gelecek nesil içerisinde farklı yerlerde yer almaktadır. Bu tür bir çaprazlama ikili kodlu genetik algoritmalar için iyi bir yöntem olarak kabul edilmesine rağmen sürekli parametreli genetik algoritmalarda iyi sonuç vermemektedir [Haupt ve Haupt, 2004].

Daha etkin çaprazlama yöntemleri geliştirilmiştir. Burada anne ve babada bulunan bilgilerin karıştırılması ile yeni çocuklar (çözümler) elde edilir. Tek çocuk değişken değeri, d_{yeni}, iki eşin kromozomlarının karışımından orantılı olarak

$$d_{yeni} = \alpha d_{an} + (1-\alpha) d_{bn} \tag{6.9}$$

şeklinde ifade edilir. Eş. 6.9'da, α sıfırla bir arasında değer alabilen ağırlık değişkeni d_{an} ve d_{bn} ise sırasıyla anne ve baba kromozomundaki *n.* değişkeni simgelemektedir. İkinci çocuğun aynı değişkeni doğrudan birincisinin tamamlayıcısıdır. Eğer $\alpha = 1$ ise, d_{an} kendi içinde çoğalır ama d_{bn} ölür. Bunun aksi olarak, eğer $\alpha = 0$ olursa, d_{bn} kendi içinde yayılır d_{an} ölür. $\alpha = 0,5$ olduğunda çocuk yani yeni çözüm anne ve baba kromozomların karışımından meydana gelir. İyi bir karışım için hangi değişken seçiminin uygun olduğunun belirlenmesi bir başka sorundur. Bazen bu doğrusal karışım tüm değişkenler için çaprazlama noktasının sağında veya solundan başlanarak bütün parametreler için bir doğrusal kombinasyon işlemi yapılır. Her değişken aynı α değeri için karıştırılabileceği gibi her biri için farklı α'lar da seçilebilir. Bütün bu karışımlar popülasyonda bulunan uç değerlerin ötesindeki değerlerin ortaya çıkmasına meydan vermezler. Bunun için bir ekstrapolasyon yöntemine gerek duyulur. Bunların en basiti ise doğrusal çaprazlama yöntemidir. Bu durumda anne ve baba kromozomlarından yani çözümlerden farklı üç çocuk yani çözüm oluşur. Bunlar yine ağırlıklarının toplamı 1 olan,

$$\begin{aligned} d_{yeni1} &= 0,5 d_{an} + 0,5 d_{bn} \\ d_{yeni2} &= 1,5 d_{an} - 0,5 d_{bn} \\ d_{yeni3} &= -0,5 d_{an} + 1,5 d_{bn} \end{aligned} \tag{6.10}$$

kromozomlardır. Değişken sınırlarının dışında olan herhangi bir değişken diğer ikisinin yanında dışlanır. Bu yaklaşımlarda $0,5$ ağırlığı kullanılabilecek yegâne faktör değildir [Şen, 2004b].

Pratik uygulamalarda çaprazlama yöntemi ile ekstrapolasyon yönteminin kombinasyonundan oluşan yöntem sıklıkla kullanılmaktadır. İlk ana kromozom çiftinden rasgele olarak bir değişkenin seçilmesi işe başlanarak

$$\alpha = \text{yukarıya yuvarla}\left[rand\ N_{değ} \right] \tag{6.11}$$

ile çaprazlama noktası seçilir. Buradan eşler

$$Eş_1 = \left[d_{a1},\ d_{a2},\ \ldots,\ d_{a\alpha},\ \ldots,\ d_{aNdeğ} \right]$$

ve

$$Eş_2 = \left[d_{b1},\ d_{b2},\ \ldots,\ d_{b\alpha},\ \ldots,\ d_{bNdeğ} \right]$$

alınırsa, seçilen değişkenlerin bir araya getirilmesi ile yeni çocuklar

$$d_{yeni1} = d_{a\alpha} - \alpha\left[d_{a\alpha} - d_{b\alpha} \right] \tag{6.12}$$

ve

$$d_{yeni2} = d_{b\alpha} + \alpha\left[d_{a\alpha} - d_{b\alpha} \right] \tag{6.13}$$

olur. Son aşama kromozomlarla çaprazlamalar şeklinde tanımlanmalıdır.

$$Çocuk_1 = \left[d_{a1},\ d_{a2},\ \ldots,\ d_{yeni1},\ \ldots,\ d_{bNdeğ} \right]$$

$$Çocuk_2 = \left[d_{b1},\ d_{b2},\ \ldots,\ d_{yeni2},\ \ldots,\ d_{aNdeğ} \right]$$

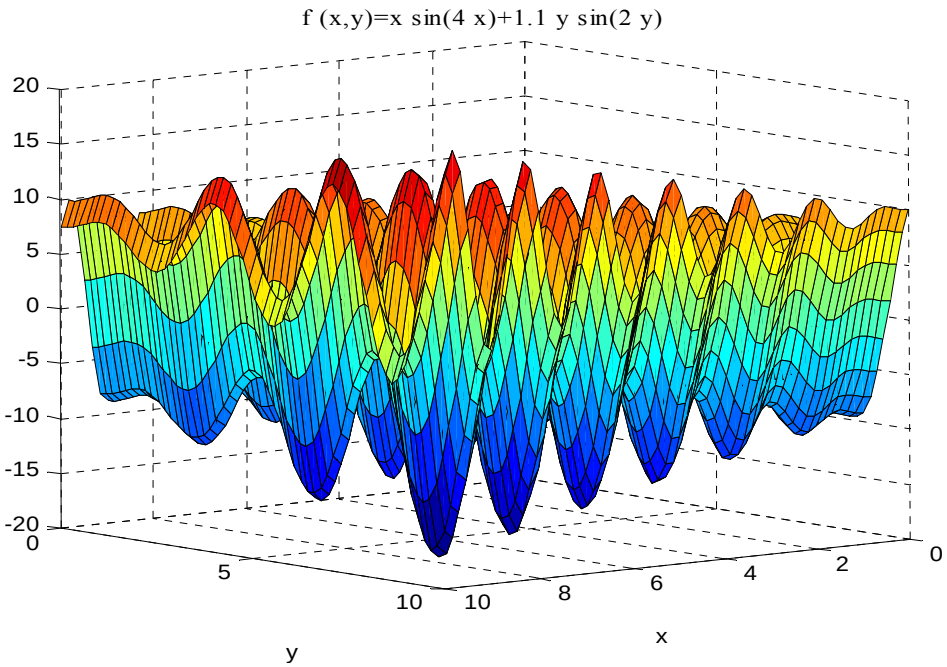
Kromozomların ilk değişkeni seçilirse, bu değişkenin sağındaki değişkenler değiş tokuş edilir. Eğer kromozomun son değişkeni seçilirse, bu değişkenin solundaki değişkenler değiş tokuş edilir. Bu yöntemle $\alpha > 1$ olmadıkça sınır değerleri aşan herhangi bir parametre üretilmez.

6.2.6. Mutasyon

Genetik algoritma ile kromozomlar amaç fonksiyonunun bir noktasına çok hızlı bir şekilde yakınsayabilirler. Bu nokta amaç fonksiyonun genel minimum noktası ise bir sorun yoktur. Çoğu uygulamalarda kullanılan amaç fonksiyonları ise çok fazla yerel minimum çözümlere sahiptir. Genetik algoritmaların bir noktaya çok hızlı bir şekilde yakınsamasına bir çare bulunmazsa yerel minimum çözümlerden genel çözüme ilerleme imkânı bulunamaz. Hızlı yakınsamadan kurtulmanın yolu çözüm uzayında mutasyon yani kromozomlardaki değişiklikler aracılığıyla yeni çözümler elde etmektir. Kromozomlardaki değişiklikler rasgele bir yöntemle yapılmalıdır. İkili kodlu genetik algoritmalarda bu işlem çaprazlama noktasındaki hanenin ya birden sıfıra veya sıfırdan bire değiştirilmesi olarak sağlanmaktadır [Gen ve Cheng, 1996]. Sürekli parametreli genetik algoritmalarda da ikili kodlu genetik algoritmalarda olduğu gibi küçük bir mutasyon oranı seçilir. Mutasyon oranı toplam değişken sayısı ile çarpıldığında mutasyona uğratılacak yani değişiklik yapılacak değişken sayısı bulunur. Bundan sonra, mutasyon yapılacak değişkenleri belirlemek için matris satır ve sütunu seçilir ve mutasyona uğratılacak değişken değişkenin değer aralığında kalmak üzere yeni bir rasgele değişken ile belirlenir.

6.2.7. Sürekli parametreli genetik algoritmalar kullanılarak bir optimizasyon probleminin eniyilenmesi örneği

$0 \le x \le 10$ ve $0 \le y \le 10$ olmak üzere $f(x,y) = x\sin(4x) + 1{,}1\, y \sin(2y)$ fonksiyonun minimumunu araştıralım [Haupt ve Haupt, 2004]. $f(x,y)$, x ve y'ye bağlı iki değişkenli bir fonksiyon olup Şekil 6.4'de grafiği gösterilmektedir. Fonksiyonun çok sayıda yerel minimuma sahip olduğu ve genel minimumun klasik yöntemlerle bulunmasının zor olduğu anlaşılmaktadır.



Şekil 6.4. $f(x,y)=x\sin(4x)+1{,}1\,y\sin(2y)$ fonksiyonunun grafiği

Kromozom $[x,y]$ olarak yazılabilir ve buradan $N_{değ}=2$ olduğu anlaşılır. Sınır değerleri, en küçük değer $d_{ek}=0$ ve en büyük değer ise $d_{eb}=10$'dur. $f(x,y)$ fonksiyonu karmaşık yapıda olduğundan başlangıç popülasyonun yüksek tutulması iyi sonuç vermektedir. Başlangıç popülasyonun sayısı $N_{bpop}=48$ seçilir ve bu durumda popülasyon matrisi 48×2 boyutunda olur. Başlangıç popülasyonun büyüklüğü çözüm uzayının geniş bir şekilde taranmasını sağlar. İlk 24 adet kromozomun dizilimi Çizelge 6.5'de verilmektedir.

Örnekte 48 adet kromozomun ortalama fonksiyon değeri $0{,}9039$ ve en iyi kromozomun fonksiyon değeri ise $-16{,}2555$ olarak bulunmuştur. Son 24 tanesi dikkate alınmazsa geri kalan yani ilk 24 kromozomun ortalama fonksiyon değeri $-4{,}27$ olmaktadır. Her iterasyonda $N_{top}=24$ kromozom kullanılacak ve bunun 12 tanesi N_{iyi}, 12 tanesi de $N_{\text{kötü}}$ olarak tanımlanacaktır.

Çizelge 6.5. 24 adet kromozomun sırayla dizilimi

No	x	y	$f(x,y)$
1	9,0465	8,3097	-16,2555
2	9,1382	5,2693	-13,5290
3	7,6151	9,1032	-12,2231
4	2,7708	8,4617	-11,4863
5	8,9766	9,3469	-10,3505
6	5,9111	6,3163	-5,4305
7	4,1208	2,7271	-5,0958
8	2,7491	2,1896	-5,0251
9	3,1903	5,2970	-4,7452
10	9,0921	3,8350	-4,6841
11	0,6056	5,1942	-4,2932
12	4,1539	4,7773	-3,9545
13	8,4598	8,8471	-3,3370
14	7,2541	3,6534	-1,4709
15	3,8414	9,3044	-1,1517
16	8,6825	6,3264	-0,8886
17	1,2537	0, 4746	-0,7724
18	7,7020	7,6220	-0,6458
19	5,3730	2,3777	-0,0419
20	5,0071	5,8898	0,0394
21	0,9073	0,6684	0,2900
22	8,8857	9,8255	0,3581
23	2,6932	7,6649	0,4857
24	2,6614	3,8342	1,6448

Eşleme için rulet tekerleği seçim yöntemi kullanılmıştır ve 12 kromozomun seçilme olasılıkları Çizelge 6.6'da verilmektedir. Bir rasgele sayı üreteci ile 6 rasgele sayı çiftinin $(0.4679, 0.5344)$, $(0.2872, 0.4985)$, $(0.1783, 0.9554)$, $(0.1537, 0.7483)$, $(0.5717, 0.5546)$ ve $(0.8024, 0.8907)$ olarak üretildiği düşünülsün (12 tane iyi kromozomun 6'sı anne 6'sı baba kromozom için ayrılmaktadır) . Üretilen bu sayılar kullanılarak çaprazlanmak üzere seçilen anne ve baba kromozomlar $Anne = [3,2,1,1,4,5]$ ve $Baba = [3,3,10,5,3,7]$ olarak gerçekleşir. Örneğin $(0.4679, 0.5344)$ rasgele sayısı ile birikimli olasılıklardan annenin de babanın da 3. kromozomları alınmaktadır. Buradan, 3. kromozomun 3. kromozomla, 2.

kromozomun 3. kromozomla, . . . , 5. kromozomun 7. kromozomla eşleşeceği anlaşılır.

Çizelge 6.6. Kromozomların eşleme havuzunda fonksiyon değerlerine göre dizilimi

N	$f(x,y)$	P_n	$\sum_{i=1}^{n} P_i$
1	-16,2555	0,2265	0,2265
2	-13,5290	0,1787	0,4052
3	-12,2231	0,1558	0,5611
4	-11,4863	0,1429	0,7040
5	-10,3505	0,1230	0,8269
6	-5,4305	0,0367	0,8637
7	-5,0958	0,0308	0,8945
8	-5,0251	0,0296	0,9241
9	-4,7452	0,0247	0,9488
10	-4,6841	0,0236	0,9724
11	-4,2932	0,0168	0,9892
12	-3,9545	0,0108	1,0000

İlk eş kümesi birbirinin aynıdır (3.kromozomlar) ve kendi parçalarını üretirler. İkinci çift kromozomlar (2. ve 3. kromozomlar) $[9.1382, 5.2693]$ ve $[7.6151, 9.1032]$ olarak elde edilir. Rasgele olarak d_1 çaprazlama noktası ve $\alpha = 0.7147$ olarak alınırsa, bu durumda yeni çocuklar Eş. 6.12 ve Eş. 6.13 kullanılarak

$$Çocuk_1 = \left[9.1032 + 0.7147\left(5.2693\right) - 0.7147\left(9.1032\right), 9.1382\right] = \left[6.3631, 9.1382\right]$$

$$Çocuk_2 = \left[5.2693 - 0.7147\left(5.2693\right) + 0.7147\left(9.1032\right), 7.6151\right] = \left[8.0094, 7.6151\right]$$

olarak bulunur. Diğer dört tane eş (1 ve 10, 1 ve 5, 4 ve 3, 5 ve 7) ile bu işlemler yapılırsa sekiz tane daha çocuk yani yeni çözüm elde edilir.

Çizelge 6.7. İkinci nesil kromozomlarının dizilimi

No	x	y	$f(x,y)$
1	9,0465	8,3128	-16,2929
2	9,0465	8,3097	-16,2555
3	9,1382	5,2693	-13,5290
4	7,6151	9,1032	-12,2231
5	7,6151	9,1032	-12,2231
6	7,6151	9,1032	-12,2231
7	2,7708	8,4789	-11,6107
8	2,7708	8,4617	-11,4863
9	9,1382	8,0094	-11,0227
10	8,9766	9,3438	-10,4131
11	8,9766	9,3469	-10,3505
12	9,0465	7,93469	-9,8737
13	1,5034	9,0860	-6,6667
14	4,4224	9,3469	-5,6490
15	5,9111	6,3163	-5,4305
16	7,6151	6,3631	-5,1044
17	9,0921	4,2422	-5,0619
18	2,7491	2,1896	-5,0251
19	3,1903	5,2970	-4,7452
20	9,0921	3,8350	-4,6841
21	0,6956	5,1942	-4,2932
22	4,1539	4,7773	-3,9545
23	8,6750	2,7271	-3,4437
24	4,1208	3,1754	-2,6482

Mutasyon işlemi ise mutasyon oranı $0,04$ olmak üzere $(0,04)(24)(2) \cong 2$ kromozomda gerçekleştirilecektir. 7. kromozomun ikinci bileşeninde değişiklik yapılacağını varsayalım. Bu durumda 7. kromozomun ikinci bileşeni silinerek 0 ile 10 sınır değerleri arasında yeni bir rasgele değerle değiştirilir. 7. kromozom; $[4.1208, 2.7271]$ iken $[4.1208, 3.1754]$ durumuna gelmiştir. Yeni çocuklardan sonra mutasyon işlemlerinin tamamlanmasıyla ikinci nesil kromozomlarına yani aday çözümlerine ulaşılır. İkinci nesil çözümleri Çizelge 6.7'de verilmektedir.

Yedi iterasyon yani yedi popülasyon sonra $x = 9,0215$, $y = 8,6806$ ve $f = -18,53$ olarak elde edilmektedir.

6.3. Yeni Önerilen Çok Katmanlı Sinir Ağlarının Genetik Algoritmalar ile Eğitilmesi

Geri yayılım algoritması, yapay sinir ağlarının eğitilmesinde en çok kullanılan algoritmadır [Hornik ve ark., 1989]. Geri yayılım algoritması öğrenme sırasında, her girdi verisini, çıkış nöronlarında sonuç üretmek üzere gizli katmanlardaki nöronlardan geçirir. Daha sonra çıkış katmanındaki hataları bulabilmek için, beklenen çıktı ile elde edilen çıktı karşılaştırılır. Bundan sonra, çıkış hatalarının türevi çıkış katmanından geriye doğru gizli katmanlara iletilir. Hata değerleri bulunduktan sonra, nöronlar kendi hatalarını azaltmak için ağırlıklarını ayarlar. Ağırlık değiştirme denklemleri, ağın performans fonksiyonunu (ortalama hata karesi) en küçük yapacak şekilde düzenlenir.

İleri beslemeli ağlarda kullanılan öğrenme algoritmaları, performans fonksiyonunu en küçük yapacak ağırlıkları ayarlayabilmek için, performans fonksiyonunun gradyentini kullanırlar. Geri yayılım algoritması da ağ boyunca gradyent hesaplamalarını geriye doğru yapar. En basit geri yayılım öğrenme algoritması eğim azalması (delta kuralı) algoritmasıdır. Bu algoritmada ağırlıklar, performans fonksiyonunun azalması yönünde ayarlanır. Fakat bu yöntem pek çok problem için çok yavaş kalmakta ve genel çözümlere ulaşılamamaktadır [Brent, 1991].

Eğim azalması yöntemine alternatif çok sayıda yöntem geliştirilmiştir. Bu yöntemler arasında momentum terimli geriye yayılım algoritması, öğrenme hızı değişen geriye yayılım, esnek geriye yayılım, eşlenik gradyent öğrenme algoritması, Newton öğrenme algoritmaları ve Levenberg-Marguardt öğrenme algoritması örnek olarak verilebilir. Bu yöntemler bir yerel çözüme takılmayı önlemek ve daha hızlı çözüm vermek için geliştirilmişlerdir. Levenberg-Marguardt algoritması Bölüm 5.3.4'de genel hatları ile verilmiştir.

Bu çalışmada çok katmanlı ağın ikili kodlu ve sürekli parametreli genetik algoritmalar ile eğitilmesi önerilmiştir. Genetik algoritmalar çok değişkenli fonksiyonların optimizasyonunda kullanılan sezgisel bir yöntemdir. Sadece amaç

fonksiyonuna ihtiyaç duyan bir optimizasyon tekniğidir. Genetik algoritmalar kör bir arama motoruna benzetebilir [Akan, 1997]. Genetik algoritmalar problemin yapısına bakmaksızın çok karmaşık optimizasyon problemleri için bile çözüm bulabilirler. Problemin karmaşıklığı genetik algoritmalar için hiç önemli değildir. Genetik algoritmaların ihtiyaç duyduğu şey problemin karar değişkenlerinin uygun bir yöntemle kodlanması ve neyin iyi olduğunu Genetik algoritmaya belirtmek üzere tasarlanan bir uygunluk (amaç) fonksiyonudur. Genetik algoritmalar çözüm uzayını taramaya bir topluluk ile başladıkları için genel optimum çözüme yaklaşmak diğer yöntemlere göre daha kolay olmaktadır. Genel olarak global optimum çözümü bulmayı garanti etmezlerse de buna yakın bir sonucu bulduğu bir çok araştırmayla ispatlanmıştır. Genetik algoritmalar bir topluluk (başlangıçta bu topluluk genelde rasgele oluşturulur) ile başlar ve bu topluluk üzerinde çaprazlama, seçme ve mutasyon gibi yöntemlerin uygulanmasıyla problemin her aşamasında en iyiye doğru bir gidiş sağlanır.

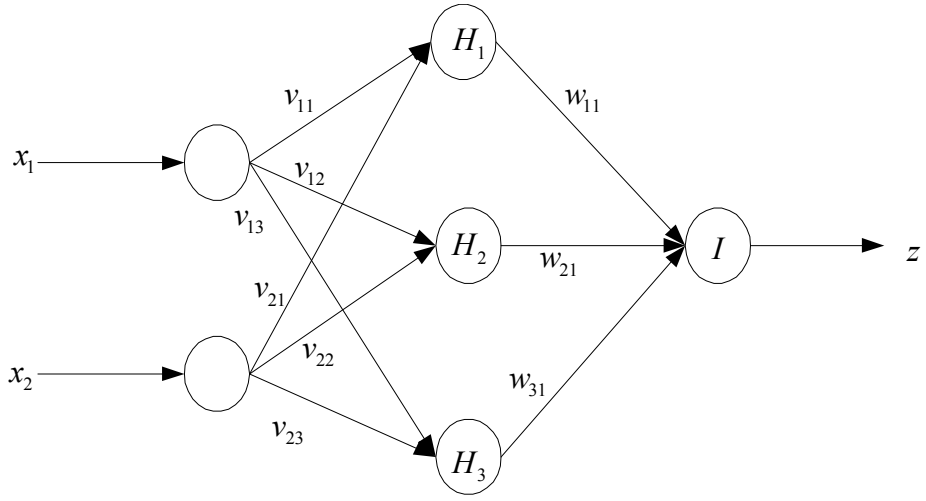
Genetik algoritmalar sezgisel bir yöntem olmaları nedeniyle verilen bir problem için her zaman optimum sonucu bulamayabilir, fakat bilinen yöntemlerle çözülemeyen ya da çözüm zamanı problemin büyüklüğü ile üstel olarak artan problemlerde optimuma yakın çözümler vermektedir. Çoğu optimizasyon probleminde karma değişkenler (sürekli ve kesikli) söz konusudur. Eğer bu durumlarda standart doğrusal olmayan programlama teknikleri kullanılırsa hesaplamalar açısından çok pahalı ve etkin olmayan durumlarla karşılaşılabilir. Genetik algoritmalar bu durumlar için iyi çözümler üretmektedirler.

Genetik algoritma, tek nokta üzerinde değil noktalar popülasyonu (aday çözümler kümesi) ile araştırma yapmaktadır. Geleneksel yöntemler ise genellikle tek noktadan çözüme başlar. Bu yöntem ile çok karmaşık olan ve bir çok minimum veya maksimum noktası bulunan problemler arasında optimum veya optimuma yakın bir çözüme ulaşılabilir. Bu şekilde yerel optimum tuzağına düşme olasılığı daha zayıftır. Genetik algoritmalar, rasgele arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan bir arama tekniğidir [Gen ve Cheng, 1996]. Genetik algoritmalar mümkün olan çözümlerin bir popülasyonu üzerinde işlem

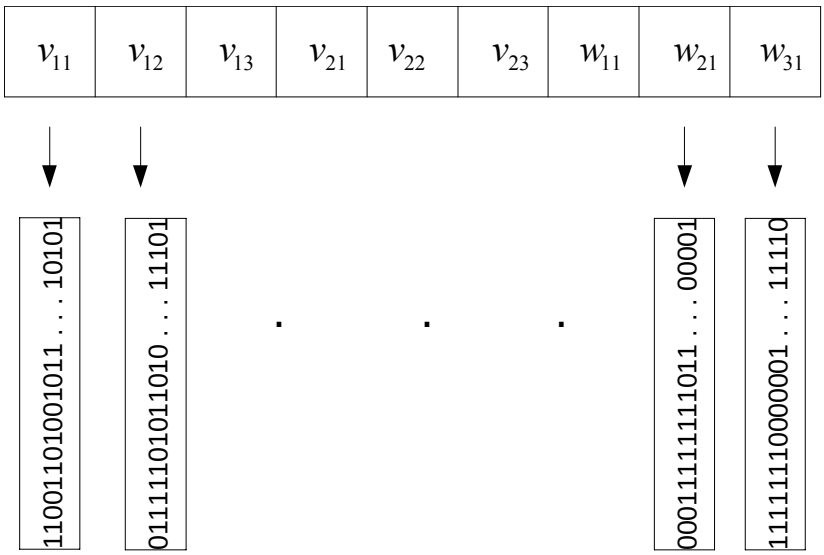
yapan olasılıklı arama algoritmalarıdır [Akan, 1997]. Geleneksel programlama teknikleriyle çözülmesi güç olan çok boyutlu optimizasyon problemleri, genetik algoritmaların yardımıyla daha kolay ve hızlı olarak çözüme ulaştırılmaktadır. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için "iyi"nin ne olduğunu belirleyen bir *uygunluk* fonksiyonu ve yeni çözümler üretmek için *yeniden kopyalama*, *mutasyon* gibi operatörleri kullanır.

Basit bir ağ yapısı üzerinde, sınıflandırma hatasının azaltılmasında kullanılan ağ ağırlık değişkenlerinin genetik algoritma ile işlem yapılmaya nasıl hazırlanacağı aşağıda incelenmektedir. Ağ yapısı ileri beslemeli bir ağdır ve bir girdi, bir gizli ve bir çıktı katmanı vardır. Girdi katmanında iki girdi nöronu, gizli katmanında üç gizli nöronu ve çıktı katmanında ise tek çıktı nöronu vardır. Ağ yapısı Şekil 6.5'de gösterilmektedir.

Ağ yapısının ağırlık değişkenleri genetik algoritma ile işlem yapılmaya hazır hale getirilmelidir yani değişkenler kromozomlara kodlanmalıdır. Şekil 6.6'da ağırlık değişkenlerinin kromozoma yerleştirilmesi ve değişkenlerin ikili kodda gösterimi yer almaktadır.



Şekil 6.5. Tek girdi, tek gizli ve tek çıktılı bir çok katmanlı ileri beslemeli ağ



Şekil 6.6. Ağ yapısı ağırlık değişkenlerinin kromozoma kodlanması

Sürekli parametreli genetik algoritmaların kullanımında da ağırlık değişkenleri için uygun aralıklar belirlenerek kodlama yapılır.

Ağırlık değişkenlerinin kodlanmasından sonra, genetik algoritmanın ihtiyaç duyacağı bilgi, uygunluk fonksiyonunun ne olduğudur.

t_k^p; p. birimin x^p girdisine ilişkin k. çıkış katmanında gerçekleşmesi beklenen/istenen çıktı, z_k^p; p. birimin x^p girdisi için k. çıktı katman sinirinin çıktısı ($k = 1, \ldots, m$) olmak üzere $E^p = \frac{1}{2}\sum_{k=1}^{m}\left(t_k^p - z_k^p\right)^2$ p. birime ilişkin hata fonksiyonudur. Burada $z_k = f\left(\sum_j w_{jk} f\left(\sum_i v_{ij} x_i\right)\right)$'dır. Aktivasyon fonksiyonu olarak $f(x) = \frac{1}{1+e^{-x}}$ lojistik fonksiyonu kullanıldığında, çıktı değeri

$$z_1 = f\left(w_{11}\frac{1}{1+e^{-(v_{11}x_1+v_{21}x_2)}} + w_{21}\frac{1}{1+e^{-(v_{12}x_1+v_{22}x_2)}} + w_{31}\frac{1}{1+e^{-(v_{13}x_1+v_{23}x_2)}}\right)$$

olarak yazılır. İlk birimin beklenen çıktısı t_1 ve gerçekleşen çıktısı da z_1 olmak üzere ortaya çıkan sınıflandırma hatası $E^1 = (t_1 - z_1)^2$ ile ifade edilir (burada çıktı katmanındaki nöron sayısı $m = 1$'dir). Tüm birimler için hatalar hesaplandığında ortalama hata kare fonksiyonu yani toplam sınıflandırma hatası $E_{ort} = \frac{\sum_{p=1}^{n} E^p}{n}$ uygunluk fonksiyonu olacaktır.

Ağırlık değişkenleri için başlangıçta rasgele çözüm popülasyonları alınır. Genetik operatörler yardımı ile uygunluk fonksiyonu olarak seçilen ve toplam sınıflandırma hatası yerine geçen ortalama hata kare fonksiyonu en küçüklenmeye çalışılır. Ortalama hata kare fonksiyonunun ulaşabileceği en küçük değer ağın sınıflandırma performansını belirleyecektir. Genetik algoritmalar üzerine yapılan çalışmalar çok karmaşık fonksiyonlarda bile minimum değerlerinin bulunmasında oldukça başarılı olduklarını göstermektedir. Toplam sınıflandırma hata fonksiyonu özellikle de sürekli parametreli genetik algoritma ile diğer ağ eğitme yöntemlerine göre çok küçük değerlere ulaşabilmekte ve sürekli parametreli genetik algoritma ile eğitilen çok katmanlı sinir ağı çok yüksek sınıflandırma başarılarına sahip olmaktadır.

7. SINIFLANDIRMA MODELLERİNİN KARŞILAŞTIRILMASI

Bu bölümde, iki gruplu ve çok gruplu durumlar için Fisher'in doğrusal ayırma fonksiyonu, yeni önerilen çok gruplu matematiksel programlama yaklaşımı ve literatürde önerilen matematiksel programlama yaklaşımları, geri yayılım algoritması, Levenberg-Marguardt, ikili kodlamalı ve yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilmiş çok katmanlı ağ yapısı yaklaşımlarının karşılaştırılması literatürden alınan gerçek sınıflandırma problemi verileri ve simülasyon aracılığı ile yapılmaktadır.

Sınıflandırma yöntemlerinin performanslarını karşılaştırmak için çapraz geçerlilik (cross validation) teknikleri kullanılmaktadır. Literatürde en çok kullanılan çapraz geçerlilik teknikleri; doğrulama örneği çapraz geçerlilik (test-holdout sample cross validation), bir birimi dışarıda tutma çapraz geçerlilik (leave-one-out cross validation), bazı birimleri dışarıda tutma çapraz geçerlilik (leave-some-out cross validation) ve k – kat çapraz geçerlilik (k – fold cross validation) yaklaşımlarıdır.

Doğrulama örneği çapraz geçerlilik yaklaşımında veri seti rasgele olarak iki sete ayrılır. Setlerden biri ayırma fonksiyonunun bulunuşunda kullanılır ve bu set eğitim veya geliştirme örneği (training sample, development sample) olarak isimlendirilmektedir. Diğer set ise doğrulama örneği (test-holdout sample) olarak isimlendirilir ve bu örnekler ayırma fonksiyonunun bulunuşunda kullanılmaz. Eğitim örneğinde ayırma fonksiyonu elde edildikten sonra yöntemin doğru sınıflandırma performansı doğrulama örneği ile sınanır.

Bir birimi dışarıda tutma çapraz geçerlilik yaklaşımında bir birim veri setinden çıkartılır ve geriye kalan $n-1$ birim ile ayırma fonksiyonu elde edilir. Başlangıçta veri setinden çıkartılan birim $n-1$ birimden elde edilen ayırma fonksiyonu yardımıyla gruplardan birine sınıflandırılır. Bu işlem her bir birim için yani n defa tekrarlanır. Her birim için elde edilen doğru sınıflandırma sayılarının ortalaması ilgili

yöntem için bir birimi dışarıda tutma çapraz geçerlilik doğru sınıflandırma performansı olarak alınır.

Bazı birimleri dışarıda tutma çapraz geçerlilik yaklaşımında veri setinden rasgele seçilen birkaç tane birim veri setinden çıkartılır ve geriye kalan birimlerle ayırma fonksiyonu elde edilir. Başlangıçta veri setinden çıkartılan birimler geriye kalan birimlerden elde edilen ayırma fonksiyonu yardımıyla gruplardan birine sınıflandırılır. Bu işlemler başlangıçta verilen tekrar sayısı kadar tekrarlanır. Her birim için elde edilen doğru sınıflandırma sayılarının ortalaması ilgili yöntem için bazı birimleri dışarıda tutma çapraz geçerlilik doğru sınıflandırma performansı olarak alınır.

$k-$kat çapraz geçerlilik yaklaşımında ise veri seti rasgele olarak k alt kümeye ayrılır. Bir birimi dışarıda bırakma yaklaşımına benzer olarak k alt kümeden bir tanesi veri setinden çıkartılır ve geriye kalan $k-1$ kümedeki birimlerle ayırma fonksiyonu elde edilir. Başlangıçta veri setinden çıkartılan küme içindeki birimler $k-1$ kümedeki birimlerden elde edilen ayırma fonksiyonu yardımıyla gruplara sınıflandırılırlar. Bu işlem her bir küme için yani k defa tekrarlanır. Her kümeden elde edilen doğru sınıflandırma sayılarının ortalaması ilgili yöntem için $k-$kat çapraz geçerlilik doğru sınıflandırma performansı olarak alınır. Literatürdeki çalışmalarda k sayısı genellikle 10 alınmakta yani veri seti 10 alt kümeye ayrılmaktadır. Bu çalışmada da veri seti rasgele olarak 10 alt kümeye ayrılarak $10-$kat çapraz geçerlilik doğru sınıflandırma performansları incelenmiştir.

Bu çalışmada, iki gruplu durum ve çok gruplu durum için incelenen gerçek sınıflandırma problemlerinde yöntemlerin sınıflandırma performansları eğitim örneği, doğrulama örneği ve $10-$kat çapraz geçerlilik yaklaşımları ile karşılaştırılmış, iki gruplu durumda yapılan simülasyon çalışması için ise doğrulama örneği çapraz geçerlilik yaklaşımı kullanılmıştır.

7.1. İki Gruplu Durum

7.1.1. Literatür örnekleri

Fisher'in doğrusal ayırma fonksiyonu (FLDF), Fred ve Glover'in MSD modeli, Ragsdale ve Stam'ın iki aşamalı MSD modeli (R&S), Lam, Choo ve Moy'un modeli (LCM), Bal ve ark. tarafından önerilen GPMED modeli, Sueyoshi'nin genişletilmiş iki aşamalı modeli (DEA-DA), geri yayılım algoritması, Levenberg-Marguardt, ikili kodlamalı ve yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilmiş çok katmanlı ağ yapısı yaklaşımlarının performansını değerlendirmek için 7 farklı veri seti dikkate alınmıştır. Veri setleri University of California-Irvine internet veri tabanından alınmıştır ve kısaca izleyen şekilde açıklanan iyonosfer, göğüs kanseri tanısı (WBCD), göğüs kanseri tedavi süreci (WBCP), kalp hastalığı, BUPA karaciğer bozuklukları, şeker hastalığı ve kredi talebi gibi bazı gerçek hayat veri setlerini içermektedir.

1.İyonosfer Veri Seti: Veriler Labrador yarımadasında bulunan bir sistem tarafından toplanmıştır. İyonosferdeki serbest elektronlar bu çalışmanın hedefidir. Radarın iyi olarak verdiği sonuçlar iyonosferde oluşan bazı yapıların göstergesidir. Kötü olarak değerlendirilen sonuçlar ise sinyalleri iyonosferi geçip gittiği için herhangi bir yapı oluşumuna işaret etmemektedirler. Elektronlar için 17 farklı titreşim ölçülmektedir. Veri tabanındaki her bir titreşim için karmaşık elektromanyetik sinyallerden oluşan fonksiyonlar ile elde edilen karmaşık değere karşı gelen 2 nitelik tanımlanmıştır. Böylelikle toplam 34 farklı ölçüm yapılmıştır. Veri iyi olarak değerlendirilen 223 yapı ve kötü olarak değerlendirilen 127 yapı olmak üzere toplam 350 yapıdan oluşmaktadır.

İyi olarak değerlendirilen yapıların 100 tanesi, kötü olarak değerlendirilen yapılardan da yine 100 tanesi eğitim setine ayrılmıştır. İyi olarak değerlendirilen yapılardan kalan 123 tanesi ve kötü olarak değerlendirilen yapılardan kalan 27 tanesi de doğrulama seti olarak kullanılmıştır.

2. Wisconsin Göğüs Kanseri Tanısı Veri Seti (WBCD): Göğüs kanseri veri tabanında, Wisconsin Üniversitesi Hastanesi'nden elde edilen veriler yer almaktadır. Veri kümesinde iyi huylu ve kötü huylu olarak adlandırılan iki grup bulunmaktadır. Veri, göğüs kanseri tipi iyi huylu olarak tanımlanan 357 hasta ve kötü huylu olarak tanımlanan 212 hasta olmak üzere toplam 569 hastadan oluşmaktadır. Hastalardan çoğunluğu kanserli olduğu düşünülen hücreler üzerinde yapılan çeşitli ölçümler olmak üzere 30 farklı ölçüm yapılmıştır.

İyi huylu olarak değerlendirilen hasta grubunun 150 tanesi, kötü huylu olarak değerlendirilen hasta grubundan da 100 tanesi eğitim setine ayrılmıştır. İyi huylu olarak değerlendirilen hastalardan kalan 207 tanesi ve kötü olarak değerlendirilen hastalardan kalan 112 tanesi de doğrulama seti olarak kullanılmıştır.

3. Wisconsin Göğüs Kanseri Tedavi Süreci Veri Seti (WBCP): Wisconsin Üniversitesi Hastanesi'nden elde edilen göğüs kanseri hastalarını iki yıl boyunca tedavi süreçlerinin izlenmesi ile ilgili veriler yer almaktadır. Veri kümesinde tedaviye olumlu yanıt verenler ve olumlu yanıt vermeyenler olarak adlandırılan iki grup bulunmaktadır. Veri, göğüs kanseri tedavi sürecinde tedaviye olumlu yanıt veren 47 hasta ve tedaviye olumlu yanıt vermeyen 151 hasta olmak üzere toplam 198 hastadan oluşmaktadır. Hastalardan çoğunluğu kanserli olduğu düşünülen hücreler üzerinde yapılan çeşitli ölçümler olmak üzere 32 farklı ölçüm yapılmıştır.

Tedaviye olumlu yanıt veren hasta grubunun 35 tanesi, tedaviye olumlu yanıt vermeyen hasta grubundan da 100 tanesi eğitim setine ayrılmıştır. Tedaviye olumlu yanıt veren hastalardan kalan 12 tanesi ve tedaviye olumlu yanıt vermeyen hastalardan kalan 51 tanesi de doğrulama seti olarak kullanılmıştır.

4. Kalp Hastalığı Veri Seti: Veri bir hastaneye kalp rahatsızlığı şikâyetiyle başvuran hastalardan oluşmaktadır. Her hasta için 13 özellik (ölçüm) olmak üzere 297 hastadan veri toplanmıştır. Hastalara ait özellikler yaş, cinsiyet, göğüs ağrısı tipi, tansiyon gibi bilgilerin yanı sıra kan ve efor testlerinden elde edilmiştir. Veri

kümesinde hasta ve sağlıklı olarak adlandırılan iki grup bulunmaktadır. Veri, kalp rahatsızlığı bulunan 150 hasta ve kalp rahatsızlığı bulunmayan 147 hasta olmak üzere toplam 297 hastadan oluşmaktadır.

Kalp rahatsızlığı olan hasta grubunun 100 tanesi, kalp rahatsızlığı olmayan hasta grubundan da yine 100 tanesi eğitim setine ayrılmıştır. Kalp rahatsızlığı olan hastalardan kalan 50 tanesi ve kalp rahatsızlığı olmayan hastalardan kalan 47 tanesi de doğrulama seti olarak kullanılmıştır.

5. BUPA Karaciğer Bozuklukları Veri Seti: BUPA veri seti 6 sayısal ölçümü olan 345 bekâr erkeğe ilişkin verileri içermektedir. Ölçümlerin beşi karaciğer bozukluğu ile ilgili olduğu düşünülen kan testleri diğeri de her gün içilen alkollü içecek sayısıdır. Veri kümesinde hasta ve sağlıklı olarak adlandırılan iki grup bulunmaktadır. Veri, karaciğer bozukluğu saptanan 125 hasta ve karaciğer bozukluğu saptanmayan 220 hasta olmak üzere toplam 345 hastadan oluşmaktadır.

Karaciğer bozukluğu saptanan hasta grubunun 50 tanesi, karaciğer bozukluğu saptanmayan hasta grubundan da yine 120 tanesi eğitim setine ayrılmıştır. Karaciğer bozukluğu saptanan hastalardan kalan 75 tanesi ve karaciğer bozukluğu saptanmayan hastalardan kalan 100 tanesi de doğrulama seti olarak kullanılmıştır.

6. Pima Hintli Şeker Hastalıkları Veri Seti: Pima şeker hastalığı veri seti, en az 21 yaşında olan ve Pima Hindistan kalıtımına sahip olan 768 bayan hastanın verisini içermektedir. Belirlenen 8 özellik her bir hastanın fiziksel özelliklerini belirtmektedir. Veri kümesinde şeker hastası olan ve şeker hastası olmayan olmak üzere iki grup bulunmaktadır. Veri, şeker hastası 268 hasta ve şeker hastası olmayan 500 hasta olmak üzere toplam 768 hastadan oluşmaktadır.

Şeker hastası olanlardan 150 tanesi, şeker hastası olmayanlardan da 250 tanesi eğitim setine ayrılmıştır. Şeker hastası olanlardan kalan 118 tanesi ve şeker hastası olmayanlardan kalan 250 tanesi de doğrulama seti olarak kullanılmıştır.

7. Kredi Talebi Değerlendirme Veri Seti: Veri, bir bankaya kredi kartı başvurusu yapan 400 kişiyi kapsamaktadır ve kredi kartı değerlendirmesi bilgilerini içermektedir. 400 kredi kartı başvurusundan 150 tanesi kabul edilmiş, 250'si ise geri çevrilmiştir. Müşterilerden derlenen değişkenler, yaş, ev sahibi olup olmaması, yıllık gelir, işindeki çalışma yılı, medeni hali, eşinin yıllık geliri, bankada hesabı olup olmaması, başvuruda referansının olup olmaması değişkenleridir.

Kredi kartı başvurusu kabul edilen müşterilerden 80 tanesi, başvurusu reddedilen müşterilerden de 150 tanesi eğitim setine ayrılmıştır. Kredi kartı başvurusu kabul edilen müşterilerden kalan 70 tanesi ve başvurusu reddedilen müşterilerden kalan 100 tanesi de doğrulama seti olarak kullanılmıştır.

Veri setlerine ilişkin, gruplardaki birim sayıları (n_{G1}, n_{G2}), eğitim setindeki birim sayıları (n_{EG1}, n_{EG2}), doğrulama setindeki birim sayıları (n_{TG1}, n_{TG2}) ve değişken (p; özellik, ölçüm) sayıları gibi bilgiler Çizelge 7.1'de özetlenmektedir.

Çizelge 7.1. Literatür veri setleri için özetleyici bilgiler

Veri Seti	n_{G1}	n_{G2}	n	n_{EG1}	n_{EG2}	n_{TG1}	n_{TG2}	p
İyonosfer	223	127	350	100	100	123	27	34
WBCD	357	212	569	150	100	207	112	30
WBCP	47	151	198	35	100	12	51	32
Kalp	150	147	297	100	100	50	47	13
BUPA	125	220	345	50	120	75	100	6
Şeker Hastalığı	268	500	768	150	250	118	250	8
Kredi Talebi	150	250	400	80	150	70	100	8

Çok katmanlı ağ geri yayılım algoritması ile eğitilirken, beş gizli katmanı olan ağ yapısı kullanılmıştır. Maksimum iterasyon sayısı 10000 olarak öğrenme oranı ise 0,1 olarak seçilmiştir. Levenberg-Marguardt algoritmasında μ parametresi 0,9 olarak seçilmiştir. Çok katmanlı ağ için parametrelerin seçiminde literatür çalışmalarından faydalanılmıştır. Çok katmanlı ağ genetik algoritmalar (ikili kodlamalı ve sürekli parametreli) ile eğitilirken, yine beş gizli katmanı olan ağ yapısı kullanılmıştır. Ağ değişkenleri için popülasyon büyüklüğü 200, bit uzunluğu 24, mutasyon olasılığı 0,01 ve gen takası olasılığı olarak ise 0,8 olarak seçilmiştir.

Yöntemlerin eğitim ve doğrulama setindeki doğru sınıflandırma performansları Çizelge 7.2'de, bir birimi dışarıda tutma ve $10-$kat çapraz geçerlilik doğru sınıflandırma oranları ise Çizelge 7.3'de verilmektedir. Çizelge 7.2 ve Çizelge 7.3 incelendiğinde genel olarak yapay sinir ağları yaklaşımlarının (geri yayılım, Levenberg-Marguardt, ikili kodlamalı ve sürekli parametreli genetik algoritmalar ile eğitilen) yedi veri setinde de yüksek oranlarda doğru sınıflandırma başarılarına sahip oldukları gözlenmektedir.

Yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının *iyonosfer* verisi için eğitim setinde %97, doğrulama setinde %94, $10-$kat çapraz geçerlilik yönteminde %90,5; *göğüs kanseri tanısı* verisi için eğitim setinde %96, doğrulama setinde %92.5, $10-$kat çapraz geçerlilik yönteminde %90,5; *göğüs kanseri tedavi* verisi için eğitim setinde %97, doğrulama setinde %95,2, $10-$kat çapraz geçerlilik yönteminde %90; *kalp hastalığı* verisi için eğitim setinde %99,5, doğrulama setinde %95,9, $10-$kat çapraz geçerlilik yönteminde %94; *BUPA karaciğer bozuklukları* verisi için eğitim setinde %97,6, doğrulama setinde %96, $10-$kat çapraz geçerlilik yönteminde %92,5; *şeker hastalığı* verisi için eğitim setinde %99,3, doğrulama setinde %96,7, $10-$kat çapraz geçerlilik yönteminde %93,5 ve *kredi talebi* verisi için eğitim setinde %97,4, doğrulama setinde %95,3, $10-$kat çapraz geçerlilik yönteminde %91,5 olmak üzere en yüksek doğru sınıflandırma oranlarına sahip olduğu gözlenmektedir.

Çizelge 7.2. İki gruplu literatür veri setleri için yöntemlerin *eğitim* ve *doğrulama* örneklerindeki doğru sınıflandırma oranları

YÖNTEM	Eğitim/Doğrulama	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
FLDF	Eğitim	0,835	0,852	0,837	0,835	0,812	0,830	0,817
	Doğrulama	0,827	0,821	0,762	0,814	0,794	0,804	0,800
MSD	Eğitim	0,845	0,828	0,830	0,865	0,818	0,848	0,830
	Doğrulama	0,813	0,824	0,794	0,794	0,800	0,810	0,812
R&S	Eğitim	0,865	0,848	0,867	0,875	0,853	0,873	0,883
	Doğrulama	0,847	0,831	0,810	0,794	0,823	0,837	0,847
LCM	Eğitim	0,850	0,832	0,852	0,855	0,824	0,875	0,874
	Doğrulama	0,827	0,818	0,762	0,773	0,817	0,837	0,829
GPMED	Eğitim	0,900	0,900	0,885	0,865	0,850	0,900	0,905
	Doğrulama	0,840	0,835	0,810	0,820	0,835	0,850	0,875
DEA-DA	Eğitim	0,915	0,920	0,926	0,910	0,929	0,925	0,939
	Doğrulama	0,867	0,875	0,889	0,887	0,909	0,897	0,906
YSA-Geri yayılım	Eğitim	0,925	0,920	0,933	0,920	0,941	0,930	0,948
	Doğrulama	0,887	0,875	0,905	0,907	0,909	0,905	0,918
YSA-Levenberg-Marguardt	Eğitim	0,920	0,920	0,926	0,915	0,929	0,925	0,943
	Doğrulama	0,873	0,875	0,905	0,897	0,903	0,902	0,912
YSA-İkili kodlu GA	Eğitim	0,965	0,960	0,963	0,965	0,965	0,990	0,970
	Doğrulama	0,933	0,925	0,937	0,948	0,954	0,959	0,947
Önerilen YSA-Sürekli parametreli GA	Eğitim	0,970	0,960	0,970	0,995	0,976	0,993	0,974
	Doğrulama	0,940	0,925	0,952	0,959	0,960	0,967	0,953

Çizelge 7.3. İki gruplu literatür veri setleri için yöntemlerin *10-kat çapraz geçerlilik* doğru sınıflandırma oranları

YÖNTEM	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
FLDF	0,790	0,805	0,740	0,785	0,780	0,790	0,770
MSD	0,800	0,795	0,760	0,800	0,770	0,810	0,805
R&S	0,835	0,810	0,800	0,815	0,805	0,800	0,830
LCM	0,805	0,820	0,790	0,795	0,790	0,825	0,810
GPMED	0,830	0,825	0,800	0,800	0,785	0,825	0,835
DEA-DA	0,835	0,820	0,805	0,805	0,790	0,830	0,825
YSA-Geri yayılım	0,855	0,835	0,815	0,825	0,805	0,835	0,835
YSA-Levenberg-Marguardt	0,855	0,840	0,835	0,830	0,815	0,840	0,830
YSA-İkili kodlu GA	0,885	0,900	0,910	0,895	0,905	0,905	0,910
Önerilen YSA-Sürekli parametreli GA	0,905	0,905	0,900	0,940	0,925	0,935	0,915

Sueyoshi (2006) tarafından önerilen DEA-DA ve Bal ve ark. (2006) tarafından önerilen GPMED yaklaşımları ise FLDF, MSD, R&S ve LCM yaklaşımlarından daha üstün doğru sınıflandırma başarısına sahiptir. Ayrıca DEA-DA yaklaşımı geri yayılım ve Levenberg-Marguardt ile eğitilen çok katmanlı ağ yapısına yakın doğru sınıflandırma başarısına sahiptir.

İkili kodlamalı genetik algoritma ile eğitilen çok katmanlı ağ yapısı yüksek oranlarda sınıflandırma performansına sahip olmasına rağmen, diğer yöntemlerden yaklaşık olarak üç kat daha fazla sürede çözüm yapabilmektedir.

7.1.2. Simülasyon çalışması

İki gruplu sınıflandırma probleminde Fisher'in doğrusal ayırma fonksiyonu (FLDF), Fred ve Glover'in MSD modeli, Ragsdale ve Stam'ın iki aşamalı modeli (R&S), Lam, Choo ve Moy'un modeli (LCM), Bal ve ark. tarafından önerilen GPMED modeli, Sueyoshi'nin iki aşamalı modeli (DEA-DA), geri yayılım algoritması, Levenberg-Marguardt ve yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilmiş çok katmanlı ağ yapısı yaklaşımlarının performansını incelemek için, bir *1000 tekrarlı* Monte Carlo simülasyon çalışması tasarlanmıştır. İkili kodlamalı genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının diğer yöntemlere göre çok uzun sürede işlem yapabilmesi nedeniyle simülasyon çalışmasında bu yönteme yer verilmemiştir.

Simülasyon çalışmasında, iki grup için altı farklı veri tipinde üç bağımsız değişkenli Düzgün (Uniform) dağılımdan veri üretilmiştir. Bu veri tiplerinden üç tanesi (B, D ve F veri tipleri) aykırı değerler tarafından bozulmuş veri tipleridir. Simülasyon çalışması aşamasında, veri tipi ve parametrelerin seçiminde Sueyoshi (2006)'dan yararlanılmıştır. Çalışmada kullanılan veri tipleri Çizelge 7.4'de özetlenmiştir.

Çizelge 7.4. Simülasyon çalışmasında kullanılan veri tipleri

Veri Tipi	Grup 1	Grup 2
A	$\{U[5,9],U[5,9],U[5,9]\}$	$\{U[3,7],U[3,7],U[3,7]\}$
B	%90 $\{U[5,9],U[5,9],U[5,9]\}$ %10 $\{U[3,5],U[3,5],U[3,5]\}$	%90 $\{U[3,7],U[3,7],U[3,7]\}$ %10 $\{U[7,9],U[7,9],U[7,9]\}$
C	$\{U[5,9],U[5,9],U[5,9]\}$	$\{U[4,8],U[4,8],U[4,8]\}$
D	%90 $\{U[5,9],U[5,9],U[5,9]\}$ %10 $\{U[3,5],U[3,5],U[3,5]\}$	%90 $\{U[4,8],U[4,8],U[4,8]\}$ %10 $\{U[8,10],U[8,10],U[8,10]\}$
E	$\{U[3,11],U[3,11],U[3,11]\}$	$\{U[4,8],U[4,8],U[4,8]\}$
F	%90 $\{U[3,11],U[3,11],U[3,11]\}$ %10 $\{U[1,3],U[1,3],U[1,3]\}$	%90 $\{U[4,8],U[4,8],U[4,8]\}$ %10 $\{U[8,10],U[8,10],U[8,10]\}$

Çok katmanlı ağ geri yayılım algoritması ile eğitilirken, gerçek sınıflandırma problemlerinde olduğu gibi, beş gizli katmanı olan ağ yapısı kullanılmıştır. Maksimum iterasyon sayısı 10000 olarak öğrenme oranı ise 0,1 olarak seçilmiştir. Çok katmanlı ağ genetik algoritmalar (ikili kodlamalı ve sürekli parametreli) ile eğitilirken, yine beş gizli katmanı olan ağ yapısı kullanılmıştır. Ağ değişkenleri için popülasyon büyüklüğü 200, bit uzunluğu 24, mutasyon olasılığı 0,01 ve gen takası olasılığı olarak ise 0,8 olarak seçilmiştir.

Simülasyon çalışması MATLAB 7.0 programı kullanılarak hazırlanmıştır. MATLAB programında doğrusal programlama problemlerini çözmek için, yapay sinir ağ yapılarını oluşturmak için ve genetik algoritma ile bir fonksiyonun minimumunu bulmak için alt modüller mevcuttur. Matematiksel programlama modellerinin çözümleri her biri için ayrı ayrı, model yapılarına uygun şekilde amaç fonksiyonu ve kısıt fonksiyonları tanımlanarak doğrusal programlama çözümleyicisi yardımıyla elde edilmiştir. MATLAB programında yapay sinir ağ yapıları da mevcut

olduğundan klasik geri yayılım algoritması ve Levenberg-Marguardt ile eğitilen ağ yapısının sınıflandırma performansı programda mevcut olan alt modüllerle, genetik algoritmalar ile eğitilen ağ yapısının sınıflandırma performansı ise sinir ağı modülünün içinde minimizasyon amaçlı olarak genetik algoritma modülünün kullanılmasıyla değerlendirilmiştir.

Toplam olarak $n = 200$ birim ve gruplarda da 100'er birim olduğu varsayılmıştır. İki grup için 100'lük birimlere ilişkin özellikler Çizelge 7.4'deki veri tiplerinden rasgele olarak üretilmiştir. Toplam 200 birimin 100 tanesi eğitim örneği (training sample), geriye kalan 100 tanesi de doğrulama örneği (holdout sample) olarak kullanılmıştır. Çizelge 7.5'de $n_1 = n_2 = 50$ durumunda, ilgili veri tipinden birinci gruba 100 ve ikinci gruba 100 olmak üzere 200 birim alınmış, birinci ve ikinci gruptaki 100'er birimin 50 tanesi eğitim örneği, 50 tanesi de tahmin örneği olarak kullanılmıştır. Çizelge 7.6'da $n_1 = 30, n_2 = 70$ durumunda, ilgili veri tipinden birinci gruba 100 ve ikinci gruba 100 olmak üzere yine 200 birim alınmış, fakat birinci gruptaki 100 birimden 70 tanesi eğitim örneği geriye kalan 30 tanesi de tahmin örneği olarak kullanılmıştır. Aynı şekilde ikinci gruptaki 100 birimden 30 tanesi eğitim örneği, geriye kalan 70 tanesi de tahmin örneği olarak kullanılmıştır. Çizelge 7.7'deki $n_1 = 70, n_2 = 30$ durumu ise Çizelge 7.6'daki durumun tersidir. n_1 ve n_2 için üç farklı durumun incelenmesi yeterli görülmüştür.

Çizelge 7.5, 7.6 ve 7.7 incelendiğinde yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının, bütün veri tiplerinde çok büyük çoğunlukla diğer 8 yaklaşımdan daha büyük doğru sınıflandırma oranına sahip olduğu görülmektedir. Ayrıca literatür verilerinde olduğu gibi genel olarak Fisher'in istatistiksel ayırma fonksiyonu ve matematiksel programlama yaklaşımlarına göre tüm durumlarda yapay sinir ağı yaklaşımlarının yüksek oranlarda doğru sınıflandırma başarılarına sahip oldukları gözlenmektedir.

Çizelge 7.5. Doğrulama örneklerindeki 9 yaklaşıma ilişkin doğru sınıflandırma oranları ($n_1=n_2=50$)

	Veri Tipi					
YÖNTEM	**A**	**B**	**C**	**D**	**E**	**F**
FLDF	83,98 (4,60)*	81,03 (4,73)	83,39 (4,56)	81,70 (5,49)	82,28 (4,99)	80,88 (5,17)
MSD	82,88 (4,76)	80,93 (5,28)	80,05 (4,97)	81,32 (4,95)	82,89 (5,32)	80,56 (4,89)
R&S	84,60 (5,03)	82,60 (5,18)	84,35 (5,37)	82,16 (4,81)	84,90 (4,86)	82,34 (5,17)
LCM	85,32 (5,27)	82,83 (5,30)	84,85 (4,70)	82,86 (5,16)	86,58 (5,44)	82,65 (5,23)
GPMED	86,12 (5,02)	84,11 (4,59)	85,30 (5,41)	83,01 (5,27)	87,30 (4,57)	84,10 (4,32)
DEA-DA	86,30 (4,97)	84,29 (5,42)	85,33 (5,15)	84,90 (4,60)	87,28 (5,04)	84,06 (4,95)
YSA-Geri yayılım	87,12 (4,58)	85,15 (3,97)	86,48 (4,26)	85,27 (5,01)	88,68 (3,48)	85,21 (3,76)
YSA-Levenberg-Marguardt	87,83 (4,52)	85,05 (4,24)	86,62 (5,04)	85,17 (4,25)	89,26 (5,05)	84,87 (3,86)
Önerilen YSA-Sürekli parametreli GA	88,05 (4,73)	85,85 (4,35)	86,96 (4,63)	86,45 (4,28)	89,45 (4,05)	85,54 (3,62)

* 1000 tekrardaki doğru sınıflandırma sayılarına ilişkin standart sapma

Çizelge 7.6. Doğrulama örneklerindeki 9 yaklaşıma ilişkin doğru sınıflandırma oranları (n_1=30, n_2=70)

	Veri Tipi					
YÖNTEM	**A**	**B**	**C**	**D**	**E**	**F**
FLDF	83,04 (5,17)*	80,59 (4,84)	82,65 (4,61)	81,16 (4,77)	81,70 (4,98)	79,59 (4,73)
MSD	83,16 (5,26)	80,71 (5,31)	82,67 (4,69)	81,90 (5,42)	82,42 (5,09)	79,01 (4,50)
R&S	83,44 (5,35)	82,19 (4,79)	83,43 (4,89)	81,85 (5,44)	84,18 (4,75)	81,44 (4,50)
LCM	84,74 (5,44)	82,36 (5,43)	84,13 (4,76)	81,90 (4,66)	85,17 (4,71)	82,21 (4,55)
GPMED	84,95 (5,01)	83,06 (4,88)	84,25 (5,08)	83,45 (4,21)	85,91 (5,16)	83,10 (4,91)
DEA-DA	85,60 (4,56)	83,31 (4,66)	84,30 (4,63)	84,22 (4,76)	86,95 (5,21)	83,13 (5,44)
YSA-Geri yayılım	85,87 (3,97)	84,75 (4,45)	86,01 (4,06)	84,47 (4,81)	86,85 (4,09)	84,59 (4,25)
YSA-Levenberg-Marguardt	85,27 (4,25)	84,95 (4,49)	86,24 (4,21)	84,12 (4,71)	86,92 (4,87)	84,55 (5,09)
Önerilen YSA-Sürekli parametreli GA	86,45 (4,47)	84,98 (5,08)	86,78 (4,69)	85,59 (4,24)	87,29 (4,19)	84,98 (5,16)

* 1000 tekrardaki doğru sınıflandırma sayılarına ilişkin standart sapma

Çizelge 7.7. Doğrulama örneklerindeki 9 yaklaşıma ilişkin doğru sınıflandırma oranları (n_1=70, n_2=30)

	Veri Tipi					
YÖNTEM	**A**	**B**	**C**	**D**	**E**	**F**
FLDF	82,70 (5,22)*	80,47 (5,45)	81,87 (5,27)	80,76 (4,66)	81,12 (4,78)	78,56 (5,45)
MSD	82,91 (5,12)	80,15 (4,50)	80,13 (4,51)	80,49 (4,85)	80,78 (5,46)	77,82 (4,66)
R&S	82,02 (5,12)	81,39 (4,81)	82,03 (4,72)	81,49 (4,63)	83,99 (5,42)	80,41 (5,01)
LCM	83,72 (5,21)	81,62 (4,97)	83,84 (4,78)	81,70 (4,76)	84,18 (5,41)	81,99 (4,51)
GPMED	84,42 (4,91)	82,78 (5,12)	84,11 (4,70)	82,05 (5,25)	84,45 (5,01)	82,32 (4,99)
DEA-DA	85,32 (4,50)	83,21 (5,14)	83,59 (5,26)	83,15 (5,21)	85,47 (4,73)	82,23 (4,76)
YSA-Geri yayılım	85,06 (4,97)	84,15 (5,15)	85,48 (4,76)	83,71 (4,11)	86,05 (4,75)	86,89 (4,41)
YSA-Levenberg-Marguardt	85,41 (4,15)	84,04 (4,47)	85,95 (5,18)	83,21 (4,98)	86,48 (5,16)	87,08 (4,64)
Önerilen YSA-Sürekli parametreli GA	86,54 (5,04)	85,08 (4,67)	86,63 (4,51)	84,68 (5,07)	87,25 (4,36)	87,82 (4,39)

* 1000 tekrardaki doğru sınıflandırma sayılarına ilişkin standart sapma

Bu kesimde Çizelge 7.5, 7.6 ve 7.7'den elde edilen sonuçlar kullanılarak geri yayılım algoritması ve sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının FLDF ve diğer matematiksel programlama yaklaşımları ile doğru sınıflandırma performansları eşleştirme t testi yardımıyla karşılaştırılmaktadır.

Hipotez testleri 6 veri tipi durumu (A, B, C, D, E ve F), geliştirme örneği ve tahmin örneği olarak seçilen 3 farklı durum ($n_1 = n_2 = 50, n_1 = 30, n_2 = 70$ ve $n_1 = 70, n_2 = 30$) için tekrarlanmıştır. Hipotezler şu şekilde oluşturulmuştur:

H_0: Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalaması ile *i.* yöntemin doğru sınıflandırma ortalaması arasında fark yoktur.

H_1: Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalaması *i.* yöntemin doğru sınıflandırma ortalamasından büyüktür (i = FLDF, MSD, R&S, LCM, DEA-DA)

Örneğin, Çizelge 7.5'de, A veri tipi (grup 1; $\{U[5,9], U[5,9], U[5,9]\}$, grup 2; $\{U[3,7], U[3,7], U[3,7]\}$) durumundaki FLDF modeline ait olan 83,98 ve 4,60 değerleri, sırasıyla 1000 tekrardaki (örnekteki) FLDF modelinin doğru sınıflandırma ortalaması ve standart sapmasıdır. Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısına ait 87,12 ve 4,58 değerleri de sırası ile 1000 örnekteki çok katmanlı ağ yapısının doğru sınıflandırma ortalaması ve standart sapmasıdır. Bu bilgiler kullanılarak aşağıdaki hipotez testi yapılmaktadır.

H_0: Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalaması ile FLDF'nin doğru sınıflandırma ortalaması arasında fark yoktur.

H_1: Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalaması FLDF'nin doğru sınıflandırma ortalamasından büyüktür.

Çizelge 7.8'de geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalamasının, FLDF, MSD, R&S, LCM ve DEA-DA modellerinin doğru sınıflandırma ortalamasından büyük olduğunu iddia eden hipotez testinin sonuçları yer almaktadır.

Çizelge 7.8'deki $n_1 = 50, n_2 = 50$ durumundaki 15,23[a] değeri, A veri tipi durumunda, geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalamasının FLDF yönteminin doğru sınıflandırma ortalamasından büyük olduğunun testindeki, *t* test istatistiğinin aldığı değeri gösterir. [a] ise bu hesaplanan 15,23'ün *p* değeridir ve *p* değeri de 0,01' den küçük bir değerdir. Test istatistiğinin aldığı 15,23 değerine göre veya p değerine göre hipotezin sonucuna karar verilebilir. *p* değerinin 0,01' den küçük olması 0,01 anlamlılık düzeyinde H_0'ın reddedildiği sonucunu çıkarır. Sonuçta, istatistiksel olarak geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının doğru sınıflandırma ortalaması FLDF'nin doğru sınıflandırma ortalamasından büyük olduğu hipotezi kabul edilmiştir.

Çizelge 7.8'de elde edilen sonuçlar incelendiğinde geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının FLDF, MSD, R&S ve LCM yaklaşımlarından her durumda üstün olduğu gözlenmektedir. Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısı DEA-DA yaklaşımı ile karşılaştırıldığında ise $n_1 = 50, n_2 = 50$ durumu için A, B, C, E ve F veri setlerinde istatistiksel olarak üstün, D veri setinde ortalama olarak büyük olmasına rağmen istatistiksel olarak aynı, $n_1 = 30, n_2 = 70$ durumu için B, C ve F veri setlerinde istatistiksel olarak üstün, A ve D veri setlerinde ortalama olarak büyük E veri setinde ortalama olarak küçük olmasına rağmen istatistiksel olarak aynı, $n_1 = 70, n_2 = 30$ durumu için B, C, D, E ve F veri setlerinde istatistiksel olarak üstün, A veri setinde ortalama olarak küçük olmasına rağmen istatistiksel olarak aynı olduğu gözlenmektedir.

Çizelge 7.8. Doğrulama örneklerindeki *geri yayılım* ağı ile eğitilmiş çok katmanlı ağ yapısının diğer (FLDF, MSD, R&S, LCM ve DEA-DA) yöntemlere karşı hipotez testi sonuçları (eşleştirme t değerleri)

	$n_1 = 50, n_2 = 50$	$n_1 = 30, n_2 = 70$	$n_1 = 70, n_2 = 30$
Test 1 (FLDF)			
A	15,23[a]	13,72[a]	10,32[a]
B	21,07[a]	19,99[a]	15,49[a]
C	15,60[a]	17,22[a]	16,05[a]
D	15,13[a]	15,43[a]	14,98[a]
E	33,16[a]	25,24[a]	23,11[a]
F	21,39[a]	24,82[a]	23,98[a]
Test 2 (MSD)			
A	20,27[a]	12,96[a]	9,91[a]
B	20,19[a]	18,41[a]	18,46[a]
C	31,03[a]	16,97[a]	25,76[a]
D	17,70[a]	11,16[a]	15,97[a]
E	28,71[a]	21,43[a]	22,96[a]
F	23,81[a]	28,48[a]	29,87[a]
Test 3 (R&S)			
A	11,69[a]	11,46[a]	13,41[a]
B	12,32[a]	12,35[a]	12,36[a]
C	9,77[a]	12,77[a]	16,25[a]
D	14,14[a]	11,40[a]	11,31[a]
E	20,11[a]	13,40[a]	8,98[a]
F	14,16[a]	16,03[a]	16,46[a]
Test 4 (LCM)			
A	8,14[a]	5,28[a]	5,88[a]
B	11,07[a]	10,73[a]	11,13[a]
C	8,12[a]	9,45[a]	7,65[a]
D	10,59[a]	12,11[a]	10,07[a]
E	10,28[a]	8,49[a]	8,17[a]
F	12,56[a]	12,04[a]	9,51[a]
Test 5 (DEA-DA)			
A	3,79[a]	1,40	–1,26
B	4,01[a]	7,06[a]	4,07[a]
C	5,41[a]	8,74[a]	8,40[a]
D	1,69	1,16	2,63[a]
E	7,20[a]	–0,49	2,70[a]
F	5,81[a]	6,67[a]	8,07[a]

[a] H_0 Red (p değ. $< 0,01$)

Çizelge 7.9. Doğrulama örneklerindeki *sürekli parametreli genetik algoritmalar* ile eğitilmiş çok katmanlı ağ yapısının diğer (FLDF, MSD, R&S, LCM ve DEA-DA) yöntemlere karşı hipotez testi sonuçları (eşleştirme t değerleri)

	$n_1 = 50, n_2 = 50$	$n_1 = 30, n_2 = 70$	$n_1 = 70, n_2 = 30$
Test 1 (FLDF)			
A	19,49[a]	15,76[a]	16,72[a]
B	23,70[a]	19,79[a]	20,33[a]
C	17,35[a]	19,85[a]	21,69[a]
D	21,58[a]	21,96[a]	17,99[a]
E	35,26[a]	27,18[a]	29,98[a]
F	23,33[a]	24,35[a]	41,86[a]
Test 2 (MSD)			
A	24,33[a]	15,07[a]	15,96[a]
B	22,75[a]	18,38[a]	24,01[a]
C	32,16[a]	19,60[a]	32,25[a]
D	24,81[a]	16,96[a]	18,87[a]
E	31,02[a]	23,37[a]	29,26[a]
F	25,87[a]	27,58[a]	49,41[a]
Test 3 (R&S)			
A	15,81[a]	13,64[a]	19,88[a]
B	15,19[a]	12,63[a]	17,40[a]
C	11,63[a]	15,62[a]	22,25[a]
D	21,07[a]	17,14[a]	14,70[a]
E	22,76[a]	15,53[a]	14,82[a]
F	16,01[a]	16,34[a]	35,17[a]
Test 4 (LCM)			
A	12,19[a]	7,68[a]	12,30[a]
B	13,92[a]	11,14[a]	16,03[a]
C	10,12[a]	12,53[a]	13,42[a]
D	16,93[a]	18,53[a]	13,54[a]
E	13,38[a]	10,63[a]	13,97[a]
F	14,36[a]	12,73[a]	29,31[a]
Test 5 (DEA-DA)			
A	8,06[a]	4,21[a]	5,70[a]
B	7,09[a]	7,96[a]	8,51[a]
C	7,43[a]	11,90[a]	13,88[a]
D	7,80[a]	6,80[a]	6,64[a]
E	10,60[a]	1,60	8,75[a]
F	7,63[a]	7,80[a]	27,27[a]

[a] H_0 Red (p değ. $< 0,01$)

Geri yayılım algoritması ile eğitilen çok katmanlı ağ yapısının FLDF ve matematiksel programlama yaklaşımları ile karşılaştırmasına benzer olarak, Çizelge 7.9'da sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının ilgili yaklaşımlarla istatistiksel olarak karşılaştırılması yer almaktadır.

Sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının DEA-DA yaklaşımı $n_1 = 30, n_2 = 70$ durumu haricinde her durumda FLDF ve matematiksel programlama yaklaşımlarından istatistiksel olarak üstün olduğu gözlenmektedir.

İki gruplu durum için, literatür verileri ve simülasyon çalışması sonuçlarına göre çok katmanlı sinir ağı yaklaşımlarının özellikle de sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının sınıflandırma problemlerinde başarı ile kullanılabileceği sonucuna varılabilir.

7.2. Çok Gruplu Durum

Fisher'in doğrusal ayırma fonksiyonu (FLDF), Gehrlein'in çok fonksiyonlu sınıflandırma modeli (GMFC), Lam ve Moy'un çok gruplu sınıflandırma modeli (MLM), Sueyoshi'nin çok gruplu sınıflandırma modeli (DEA-DA), Gochet ve ark. tarafından önerilen GCH modeli, önerilen çok gruplu sınıflandırma modeli, geri yayılım algoritması, Levenberg-Marguardt, ikili kodlamalı ve yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilmiş çok katmanlı ağ yapısı yaklaşımlarının performansını değerlendirmek için 5 farklı veri seti dikkate alınmıştır. Veri setleri University of California-Irvine internet veri tabanından alınmıştır ve kısaca izleyen şekilde açıklanan *IRIS*, *üzüm tanıma*, *cam tanımlama*, *ecoli* ve *yeast protein tanımlama* gibi bazı gerçek hayat veri setlerini içermektedir. Matematiksel programlama modellerinin çözümü WINQSB programı yardımıyla, çok katmanlı ağ yapısı ile yapılan işlemler ise MATLAB programı yardımıyla elde edilmiştir.

1. FISHER'in IRIS Veri Seti: Bu veri setinde. 3 farklı süs çiçeği türü yani 3 farklı grup vardır; *setosa*, *versicolor*, *virginica*. Veri seti üç grubun her birinde 50 çiçek

olmak üzere toplam 150 çiçekten oluşmaktadır. Her bir çiçekten 4 farklı özellik gözlenmiştir; *sepal* uzunluğu, *sepal* genişliği, *petal* uzunluğu ve *petal* genişliği.

2. Üzüm Tanıma Veri Seti: Bu veri tabanında bulunan veriler, İtalya'nın aynı bölgesinde yetişen ancak üç farklı biçimde ekilen üzümlerin kimyasal bir analizinin sonuçlarıdır. Gruplarda yer alan üzümlerin sayısı sırasıyla 59, 71 ve 48'dir ve üzümler için 13 farklı kimyasal ölçüm yapılmıştır.

3. Cam Tanımlama Veri Seti: Cam veri kümesi, bir cam örnekleminin öz yapısını belirlemek için kullanılmaktadır. Cam türlerinin sınıflandırıldığı bu çalışma kriminolojik araştırma ile desteklenmiştir. Toplam 214 tane cam olmak üzere 6 farklı cam türü yani 6 farklı grup vardır. Gruplardaki cam sayıları sırasıyla 70, 17, 76, 13, 9 ve 29'dur ve camlardan 10 farklı kimyasal ölçüm yapılmıştır.

4. Ecoli-Protein Tanımlama Veri Seti: Ecoli veri kümesi, ecoli proteinlerinin hücresel konumlarının tahmin edilmesinde kullanılmaktadır. Toplam 327 tane ecoli proteini olmak üzere 5 farklı ecoli protein türü yani 5 farklı grup vardır (veri tabanında toplam 8 farklı grup olmasına rağmen 3 gruptaki birim sayısı çok az olduğundan 5 grup dikkate alınmıştır). Gruplardaki protein sayıları sırasıyla 143, 77, 52, 35 ve 20'dir ve proteinlerle ilgili 7 farklı kimyasal ölçüm yapılmıştır.

5. Yeast-Protein Tanımlama Veri Seti: Yeast veri kümesi de ecoli veri setine benzer olarak, yeast proteinlerinin hücresel konumlarının tahmin edilmesinde kullanılmaktadır. Toplam 1481 tane ecoli proteini olmak üzere 9 farklı ecoli protein türü yani 9 farklı grup vardır (veri tabanında toplam 10 farklı grup olmasına rağmen 1 gruptaki birim sayısı çok az olduğundan 9 grup dikkate alınmıştır). Gruplardaki protein sayıları sırasıyla 463, 429, 244, 163, 51, 44, 37, 30 ve 20'dir ve proteinlerle ilgili 8 farklı kimyasal ölçüm yapılmıştır.

5 farklı veri seti için de gruplardaki birim sayıları yaklaşık olarak eşit olacak şekilde eğitim ve doğrulama seti olarak ayrılmışlardır.

Çok katmanlı ağ geri yayılım algoritması ile eğitilirken, yedi gizli katmanı olan ağ yapısı kullanılmıştır. Maksimum iterasyon sayısı 10000 olarak öğrenme oranı ise 0,1 olarak seçilmiştir. Levenberg-Marguardt algoritmasında μ parametresi 0,9 olarak seçilmiştir. Çok katmanlı ağ genetik algoritmalar (ikili kodlamalı ve sürekli parametreli) ile eğitilirken, yine yedi gizli katmanı olan ağ yapısı kullanılmıştır. Ağ değişkenleri için popülasyon büyüklüğü 200, bit uzunluğu 24, mutasyon olasılığı 0,01 ve gen takası olasılığı olarak ise 0,8 olarak seçilmiştir.

Yöntemlerin eğitim ve doğrulama setindeki doğru sınıflandırma performansları Çizelge 7.10'da, 10 – kat çapraz geçerlilik performansları ise Çizelge 7.11'de verilmektedir. Çizelge 7.10 ve Çizelge 7.11 incelendiğinde genel olarak yapay sinir ağları yaklaşımlarının (geri yayılım, Levenberg-Marguardt, ikili kodlamalı ve sürekli parametreli genetik algoritmalar ile eğitilen) beş veri setinde de yüksek oranlarda doğru sınıflandırma başarılarına sahip oldukları gözlenmektedir.

Yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının *IRIS* verisi için eğitim setinde %100, doğrulama setinde %96,7, 10 – kat çapraz geçerlilik yönteminde %94; *üzüm tanıma* verisi için eğitim setinde %100, doğrulama setinde %98,2, 10 – kat çapraz geçerlilik yönteminde %93,5; *cam tanımlama* verisi için eğitim setinde %93,8, doğrulama setinde %82,1, 10 – kat çapraz geçerlilik yönteminde %85; *ecoli proteini tanımlama* verisi için eğitim setinde %90,5, doğrulama setinde %83,6, 10 – kat çapraz geçerlilik yönteminde %86 ve *yeast proteini tanımlama* verisi için eğitim setinde %95,4, doğrulama setinde %87,6, 10 – kat çapraz geçerlilik yönteminde %93 olmak üzere en yüksek doğru sınıflandırma oranlarına sahip olduğu gözlenmektedir.

Çizelge 7.10. Çok gruplu literatür veri setleri için yöntemlerin *eğitim* ve *doğrulama* örneklerindeki doğru sınıflandırma oranları

YÖNTEM	Eğitim/Doğrulama	Set 1	Set 2	Set 3	Set 4	Set 5
FLDF	Eğitim	0,940	0,950	0,885	0,876	0,900
	Doğrulama	0,857	0,847	0,721	0,776	0,785
GMFC	Eğitim	0,950	0,950	0,875	0,885	0,900
	Doğrulama	0,840	0,853	0,715	0,783	0,785
MLM	Eğitim	0,955	0,935	0,885	0,860	0,908
	Doğrulama	0,862	0,824	0,730	0,751	0,792
DEA-DA	Eğitim	0,970	0,935	0,885	0,860	0,908
	Doğrulama	0,885	0,824	0,730	0,751	0,792
GCH	Eğitim	0,990	0,970	0,890	0,855	0,890
	Doğrulama	0,912	0,900	0,786	0,785	0,805
Önerilen Matematiksel Programlama Modeli	Eğitim	1,000	0,990	0,890	0,865	0,910
	Doğrulama	0,926	0,932	0,795	0,800	0,818
YSA-Geri yayılım	Eğitim	1,000	0,995	0,890	0,865	0,905
	Doğrulama	0,926	0,932	0,795	0,800	0,810
YSA-Levenberg-Marguardt	Eğitim	1,000	0,995	0,900	0,855	0,900
	Doğrulama	0,926	0,940	0,800	0,793	0,810
YSA-İkili kodlu GA	Eğitim	1,000	1,000	0,910	0,870	0,925
	Doğrulama	0,942	0,955	0,800	0,808	0,834
Önerilen YSA-Sürekli parametreli GA	Eğitim	1,000	1,000	0,938	0,905	0,954
	Doğrulama	0,967	0,982	0,821	0,836	0,876

Çizelge 7.11. Çok gruplu literatür veri setleri için yöntemlerin *10-kat çapraz geçerlilik* doğru sınıflandırma oranları

YÖNTEM	Set 1	Set 2	Set 3	Set 4	Set 5
FLDF	0,830	0,825	0,740	0,775	0,745
GMFC	0,830	0,830	0,745	0,755	0,780
MLM	0,845	0,830	0,760	0,790	0,795
DEA-DA	0,860	0,850	0,795	0,800	0,795
GCH	0,865	0,860	0,815	0,810	0,800
Önerilen Matematiksel Programlama Modeli	0,890	0,905	0,835	0,830	0,840
YSA-Geri yayılım	0,900	0,910	0,830	0,825	0,835
YSA-Levenberg-Marguardt	0,900	0,910	0,835	0,830	0,830
YSA-İkili kodlu GA	0,910	0,915	0,840	0,845	0,845
Önerilen YSA-Sürekli parametreli GA	0,940	0,935	0,850	0,860	0,900

İkili kodlamalı genetik algoritma ile eğitilen çok katmanlı ağ yapısı yüksek oranlarda sınıflandırma performansına sahip olmasına rağmen, iki gruplu durumda olduğu gibi diğer yöntemlerden çok daha fazla sürede çözüm yapabilmektedir.

Yeni önerilen çok gruplu matematiksel programlama yaklaşımının *IRIS* verisi için eğitim setinde %100, doğrulama setinde %92,6, 10 – kat çapraz geçerlilik yönteminde %89; *üzüm tanıma* verisi için eğitim setinde %99, doğrulama setinde %93,2, 10 – kat çapraz geçerlilik yönteminde %90,5; *cam tanımlama* verisi için eğitim setinde %89, doğrulama setinde %79,5, 10 – kat çapraz geçerlilik yönteminde %83,5; *ecoli proteini tanımlama* verisi için eğitim setinde %86,5, doğrulama setinde %80, 10 – kat çapraz geçerlilik yönteminde %83, *yeast proteini tanımlama* verisi için eğitim setinde %91, doğrulama setinde %81,8, 10 – kat çapraz geçerlilik yönteminde %84 olmak üzere Fisher'in istatistiksel sınıflandırma fonksiyonu ve diğer çok gruplu matematiksel programlama yöntemlerine göre daha başarılı sınıflandırma performansına sahip olduğu gözlenmektedir.

8. SONUÇLAR

Ayırma Analizi; tıp ve psikoloji bilim dallarında hastaların sınıflandırılması, biyoloji bilim dalında canlıların sınıflandırılması, bitki türlerinin belirlenmesi, kimya bilim dalında maddelerin ayırımı ve antropoloji bilim dalında ırkların belirlenmesi gibi çok çeşitli alanlarda başarı ile kullanılan bir tekniktir.

1930'larda Fisher'in istatistiksel ayırma analizi ile başlayan araştırmalar sürekli artarak günümüze kadar gelmiştir. En yaygın kullanılanları doğrusal ve karesel istatistiksel ayırma fonksiyonları, matematiksel programlama teknikleri ve yapay sinir ağları yöntemleridir. İstatistiksel yöntemler verinin alındığı dağılımla ilgili normallik ve gruplar için varyans-kovaryans matrislerinin eşitliği varsayımlarının sağlanmadığı durumlarda düşük doğru sınıflandırma performanslarına sahiptirler. Matematiksel programlama yöntemleri verinin alındığı dağılımla ilgili herhangi bir varsayıma ihtiyaç duymamalarına rağmen, bazı modellerin tamsayılı oluşu ve çoğunun iki gruplu sınıflandırma problemlerinde kullanılıyor olması matematiksel programlama tekniklerinin kullanışlılığını kısıtlamaktadır. Yapay sinir ağları son yıllarda istatistiksel ve matematiksel programlama yöntemleri ile kıyaslandığında birçok sınıflandırma probleminde başarı ile kullanılmaktadır. Yapay sinir ağlarında ağın eğitilmesi en önemli problem olarak gözükmektedir. Ağ eğitilmesinde en sık kullanılan klasik geri yayılım algoritması gradyente bağlı bir yaklaşım olduğundan yerel çözümlere takılabilmesi ve bu nedenle düşük sınıflandırma performansı vermesi ve ayrıca genel çözüme ulaşmak için işlemlerinin çok uzun zaman alması gibi dezavantajlara sahiptir.

Bu çalışmada sınıflandırma problemlerinin çözümünde kullanılmak üzere bir tanesi matematiksel programlama açısından diğeri de yapay sinir ağları açısından olmak üzere iki farklı yaklaşım sunulmaktadır.

Özellikle çok gruplu sınıflandırma problemlerinin çözümünde kullanılabilecek yeni matematiksel programlama yaklaşımı, Gochet ve ark. (1997) tarafından önerilen çok gruplu LP^q matematiksel programlama yaklaşımı ve Sueyoshi (2006) tarafından

önerilen DEA-DA yaklaşımlarının güçlü özelliklerinin bir birleşimidir. Gochet ve ark.(1997) tarafından önerilen çok gruplu sınıflandırma modeli verinin özel bir sıra yapısında olmasına ihtiyaç duymadığı gibi tamsayılı bir model yapısında da değildir. Sueyoshi (2006) tarafından önerilen DEA-DA sınıflandırma modeli verinin özel bir yapıda olması ve tamsayılı bir model yapısında olmasına rağmen, birimleri konveks bir küme içerisine alarak sınıflandırması ve veride negatif değerlere izin vermesi özelliklerinden dolayı metodolojik açıdan diğer yöntemlerden daha güçlüdür. Bu iki modelin bir birleşimi olan yeni çok gruplu matematiksel programlama modelinde veride özel bir sıralamaya ihtiyaç duyulmamakta, birimler konveks bir küme ile sınıflandırılmakta ve modelde tamsayılı bir değişken kullanılmamaktadır.

Yapay sinir ağları da sınıflandırma amaçlı olarak çok geniş bir kullanıma sahiptir. Yapay sinir ağlarında ağın eğitilmesi genellikle gradyente bağlı bir yaklaşım olan geri yayılım algoritması ile yapılmaktadır. Geri yayılım algoritması ile eğitim, ağ tarafından oluşturulan çıktı ile olması beklenen çıktı arasındaki farkın karesine dayanan hata fonksiyonunun minimizasyonu ile yapılmaktadır. Geri yayılım algoritması gradyente bağlı bir yaklaşım olduğundan yerel çözümlere takılabilmekte, genel çözüm elde edebilmek için çok fazla uzun sürede işlem yapması gerekebilmekte ve yerel bir çözüme takıldığında da düşük sınıflandırma performansı verebilmektedir.

Genetik algoritmalar karmaşık yapıdaki fonksiyonların minimumlarının bulunmasında kullanılan modern sezgisel yöntemlerden birisidir ve türev bilgisine ihtiyaç duymazlar [Goldberg, 1989]. Bu çalışmada, klasik geri yayılım algoritmasına alternatif olarak, çok katmanlı ağ yapısının eğitilmesinde kullanılan hata fonksiyonunun minimizasyonu ikili kodlu ve sürekli parametreli genetik algoritmalar ile ele alınmıştır. İkili kodlu genetik algoritma ile eğitilen çok katmanlı ağ yapısının geri yayılım algoritmasından da çok fazla uzun sürede işlem yaptığı gözlenmiştir. Bu yüzden tez çalışmasında eğitme işlemini geri yayılım algoritmasına göre daha kısa sürede yapabilen ve hata fonksiyonunun minimumu için genel çözüme daha yakın çözümler verebilen sürekli parametreli genetik algoritma yaklaşımı kullanılmıştır. Sınıflandırma problemlerinde kullanılmak üzere; yeni bir, çok gruplu matematiksel

programlama yaklaşımının yanı sıra, sürekli parametreli genetik algoritma ile eğitilen çok katmanlı ağ yapısı sınıflandırma problemlerinin çözümü için önerilmiştir.

Tez çalışmasında yeni önerilen çok gruplu matematiksel programlama modeli ve yeni önerilen sürekli parametreli genetik algoritma ile eğitilmiş çok katmanlı ağ yapısının sınıflandırma performanslarını diğer istatistiksel, matematiksel programlama ve yapay sinir ağları yaklaşımları ile karşılaştırmak için iki gruplu durumda gerçek sınıflandırma problemi verileri ile simülasyon verileri, çok gruplu durumda ise gerçek sınıflandırma problemi verileri kullanılmıştır. İki gruplu durum ve çok gruplu durum için incelenen gerçek sınıflandırma problemlerinde yöntemlerin sınıflandırma performansları eğitim örneği, doğrulama örneği ve 10 – kat çapraz geçerlilik yaklaşımları ile karşılaştırılmış, iki gruplu durumda yapılan simülasyon çalışması için ise doğrulama örneği çapraz geçerlilik yaklaşımı kullanılmıştır.

İki gruplu durum için kullanılan gerçek sınıflandırma problemleri *iyonosfer*, *göğüs kanseri tanısı (WBCD)*, *göğüs kanseri tedavi süreci (WBCP)*, *kalp hastalığı*, *BUPA karaciğer bozuklukları*, *şeker hastalığı* ve *kredi talebi* gibi bazı gerçek hayat veri setlerini içermektedir. İki gruplu durumda gerçek sınıflandırma problemlerinden elde edilen sonuçlar incelendiğinde genel olarak yapay sinir ağları yaklaşımlarının (geri yayılım, Levenberg-Marguardt, ikili kodlamalı ve sürekli parametreli genetik algoritmalar ile eğitilen) yedi veri setinde de yüksek oranlarda doğru sınıflandırma başarılarına sahip oldukları gözlenmektedir.

Yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının *iyonosfer* verisi için eğitim setinde %97, doğrulama setinde %94, 10 – kat çapraz geçerlilik yönteminde %90,5; *göğüs kanseri tanısı* verisi için eğitim setinde %96, doğrulama setinde %92,5, 10 – kat çapraz geçerlilik yönteminde %90,5; *göğüs kanseri tedavi* verisi için eğitim setinde %97, doğrulama setinde %95,2, 10 – kat çapraz geçerlilik yönteminde %90; *kalp hastalığı* verisi için eğitim setinde %99,5, doğrulama setinde %95,9, 10 – kat çapraz geçerlilik yönteminde %94; *BUPA karaciğer bozuklukları* verisi için eğitim setinde %97,6, doğrulama setinde %96,

10 – kat çapraz geçerlilik yönteminde %92,5; *şeker hastalığı* verisi için eğitim setinde %99,3, doğrulama setinde %96,7, 10 – kat çapraz geçerlilik yönteminde %93,5 ve *kredi talebi* verisi için eğitim setinde %97,4, doğrulama setinde %95,3, 10 – kat çapraz geçerlilik yönteminde %91,5 olmak üzere en yüksek doğru sınıflandırma oranlarına sahip olduğu gözlenmektedir.

Yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının doğru sınıflandırma performansı diğer yöntemlerle karşılaştırıldığında oldukça yüksektir.

DEA-DA ve GPMED modellerinin ise FLDF, MSD, R&S ve LCM yaklaşımlarından daha üstün doğru sınıflandırma performansına sahip olduğu gözlenebilir.

Simülasyon çalışmasında, iki grup için altı farklı veri tipinde üç bağımsız değişkenli Düzgün dağılımdan veri üretilmiştir. Bu veri tiplerinden üç tanesi (B, D ve F veri tipleri) aykırı değerler tarafından bozulmuş veri tipleridir. Simülasyon sonuçları incelendiğinde sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının, bütün veri tiplerinde çok büyük çoğunlukla diğer 8 yaklaşımdan (FLDF, MSD, R&S, LCM, GPMED, DEA-DA, YSA-geri yayılım, YSA-Levenberg-Marguardt) daha büyük doğru sınıflandırma oranına sahip olduğu görülmektedir.

İki gruplu durum için, gerçek sınıflandırma problemi verileri ve simülasyon çalışması sonuçlarına göre çok katmanlı sinir ağı yaklaşımlarının özellikle de sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı ağ yapısının sınıflandırma problemlerinde başarı ile kullanılabileceği sonucuna varılabilir.

Çok gruplu durum için kullanılan gerçek sınıflandırma problemleri *IRIS*, *üzüm tanıma*, *cam tanımlama*, *ecoli* ve *yeast protein tanımlama* gibi bazı gerçek hayat veri setlerini içermektedir. Çok gruplu durumda gerçek sınıflandırma problemlerinden elde edilen sonuçlar incelendiğinde genel olarak yapay sinir ağları yaklaşımlarının

(geri yayılım, Levenberg-Marguardt, ikili kodlamalı ve sürekli parametreli genetik algoritmalar ile eğitilen) beş veri setinde de yüksek oranlarda doğru sınıflandırma başarılarına sahip oldukları gözlenmektedir.

Yeni önerilen sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının *IRIS* verisi için eğitim setinde %100, doğrulama setinde %96,7, 10 – kat çapraz geçerlilik yönteminde %94; *üzüm tanıma* verisi için eğitim setinde %100, doğrulama setinde %98,2, 10 – kat çapraz geçerlilik yönteminde %93,5; *cam tanımlama* verisi için eğitim setinde %93,8, doğrulama setinde %82,1, 10 – kat çapraz geçerlilik yönteminde %85; *ecoli proteini tanımlama* verisi için eğitim setinde %90,5, doğrulama setinde %83,6, 10 – kat çapraz geçerlilik yönteminde %86 ve *yeast proteini tanımlama* verisi için eğitim setinde %95,4, doğrulama setinde %87,6, 10 – kat çapraz geçerlilik yönteminde %93 olmak üzere en yüksek doğru sınıflandırma oranlarına sahip olduğu gözlenmektedir.

İki gruplu durumda olduğu gibi çok gruplu durumda da, sürekli parametreli genetik algoritmalar ile eğitilen çok katmanlı yapay sinir ağı yapısının doğru sınıflandırma performansı diğer yöntemlerle karşılaştırıldığında oldukça yüksektir.

Yeni önerilen çok gruplu matematiksel programlama yaklaşımının literatürde önerilen matematiksel programlama yaklaşımlarına göre daha kolay çözüm yapabildiği ve sınıflandırma performansının bu yöntemlere göre daha üstün bir durumda olduğu gözlenmektedir.

Yeni önerilen çok gruplu matematiksel programlama yaklaşımının *IRIS* verisi için eğitim setinde %100, doğrulama setinde %92,6, 10 – kat çapraz geçerlilik yönteminde %89; *üzüm tanıma* verisi için eğitim setinde %99, doğrulama setinde %93,2, 10 – kat çapraz geçerlilik yönteminde %90,5; *cam tanımlama* verisi için eğitim setinde %89, doğrulama setinde %79,5, 10 – kat çapraz geçerlilik yönteminde %83,5; *ecoli proteini tanımlama* verisi için eğitim setinde %86,5, doğrulama setinde

%80, 10 – kat çapraz geçerlilik yönteminde %83, *yeast proteini tanımlama* verisi için eğitim setinde %91, doğrulama setinde %81,8, 10 – kat çapraz geçerlilik yönteminde %84 olmak üzere Fisher'in istatistiksel sınıflandırma fonksiyonu ve diğer çok gruplu matematiksel programlama yöntemlerine göre daha başarılı sınıflandırma performansına sahip olduğu gözlenmektedir.

Çok gruplu durum için, gerçek sınıflandırma problemi verileri sonuçlarına göre yeni önerilen sürekli parametreli genetik algoritma ile eğitilmiş çok katmanlı ağ yapısının ve Gochet ve ark. (1997) tarafından önerilen çok gruplu LP^q matematiksel programlama yaklaşımı ve Sueyoshi (2006) tarafından önerilen DEA-DA yaklaşımlarının güçlü özelliklerinin bir birleşimi olan yeni çok gruplu matematiksel programlama yaklaşımının sınıflandırma problemlerinde başarı ile kullanılabileceği sonucuna varılabilir.

KAYNAKLAR

Akan, F.C., "Genetik algoritmalar yardımıyla optimizasyon", Yüksek Lisans Tezi, ***Uludağ Üniversitesi, Fen Bilimleri Enstitüsü***, Bursa, 6-19 (1997).

Anderson, T.W., "An introduction to multivariate analysis", ***Wiley***, New York, USA, 10-25 (1984).

Bajgier, S.M., Hill, A.V., "An experimental comparison of statistical and linear programming approaches to the discriminant problem", ***Decision Sciences***, 13: 604-618 (1982).

Bal, H., "Çok gruplu ayırma analizi problemine doğrusal ve tamsayılı programlama yaklaşımı ve yeni bir model", ***Gazi Üniversitesi Fen Bilimleri Dergisi***, 12 (4): 957-962 (1999).

Bal, H., Örkcü, H.H., Çelebioğlu S., "An alternative model to Fisher and linear programming approaches in two-group classification problem: minimizing deviations from the group median", ***G.U. Journal of Science***, 19 (1): 49-55 (2006a).

Bal, H., Örkcü, H.H., Çelebioğlu S., "An experimental comparison of the new goal programming and linear programming approaches in the two-group discriminant problems", ***Computers&Industrial Engineering***, 50 (3): 296-311 (2006b).

Bal, H., Örkcü, H.H., "Data envelopment analysis approach to two-group classification problems and an experimantal comparison with some classification models", ***Hacettepe Journal of Mathematics and Statistics***, 36 (2): 169-180 (2007).

Bishop, C., "Neural Networks for Pattern Recognotion", ***Clarendon Press***, Oxford, 25-32 (1995).

Brent, R., "Fast training algorithms for multilayer neural nets", ***IEEE Trans. on Neural Networks***, 2: 346–354 (1991).

Charnes, A., Cooper, W.W., Rhodes, E., "Measuring the efficiency of decision making units", ***European Journal of Operational Research***, 2: 429-444 (1978).

Choo, E.U., Wedley, W.C., "Optimal criterion weights in repetitive multicriteria decision making", ***Journal of Operational Research Society***, 36: 983-992 (1985).

Cooper, W.W., Seiford, L.M., Tone, K., "Data envelopment analysis", ***Kluwer Academic Publishers***, Boston USA., 25-60 (2000).

Curram, C., Mingers, J., "Neural networks, decision tree induction and discriminant Analysis", ***Journal of Operations Research Society***, 45 (4): 440-450 (1994).

Elmas, Ç., "Yapay sinir ağları", ***Seçkin Yayıncılık***, Ankara, 25-40 (2003).

Fine, T.L., "Feedforward neural network methodology", ***Springer***, New York, 5-30 (1999).

Fisher, R., "The use of multiple measurements in taxonomic problems", ***Annals of Eugenics***, 7 (2): 179-188 (1936).

Fred, N., Glover, F., "A Linear programming approach to the discriminant problem", ***Decision Sciences***, 12: 68-74 (1981a).

Fred, N., Glover, F., "Simple but powerful goal programming formulations for the statistical discriminant problem", ***European Journal of Operational Research***, 7: 44-60 (1981b).

Fred, N., Glover, F., "Evaluating alternative linear programming models to solve the two-group discriminant problem", ***Decision Sciences***, 17: 151-162 (1986a).

Fred, N., Glover, F., "Resolving certain difficulties and improving the classification power of LP discriminant analysis formulations", ***Decision Sciences***, 17: 589-595 (1986b).

Gehrlein, W.W., "General mathematical programming formulations for the statistical classification problem", ***Operations Research Letters***, 5: 299-304 (1986).

Gen, M., Cheng, R., "Genetic algorithms and engineering design", ***Wiley***, New York, 90-159 (1996).

Goldberg, D.E., "Genetic algorithms in search, optimization and machine learning", ***Addison Wesley***, Readin, MA, 55-110 (1989).

Glover, F., "Improving linear programming models for the discriminant problem", ***Decision Sciences***, 21: 771-785 (1990).

Gochet, W., Stam, A., Srinivisan, V., Chen, Shaoxiang, C., "Multigroup discriminant analysis using linear programming", ***Operations Research***, 45 (2): 213-225 (1997).

Hagan, M., Menjah, M., "Training feedforward networks with the Marguardt algorithm", ***IEEE Trans. on Neural Networks***, 5: 989–993 (1994).

Haupt, R.L., Haupt, S.E., "Practical genetic algorithms", ***Wiley***, USA, 72-100 (2004).

Holland, J., "Adaptation in natural and artificial systems", ***University of Michigan Press***, Ann Arbor, 80-100 (1975).

Holmstrom, L., Koistinen, P., Laaksonen, J., Oja, E., "Neural and statistical classifiers taxonomy and two case studies", ***IEEE Trans. Neural Networks***, 8: 5–17 (1997).

Hornik, K., Stinchcombe, H., White, H., "Multilayer feed-forward networks are universal approximators", ***Neural Networks***, 2 (5): 359-366 (1989).

Hosseini, J.H., Armacost, R.L., "Two-group discriminant problem with equal group mean vectors: An experimental evaluation of six linear/nonlinear programming formulations", ***European Journal of Operational Research***, 77: 241-252 (1994).

Joachimsthaler, E.A., Stam, A., "Four approaches to the classification problem in discriminant analysis: An experimental study", ***Decision Sciences***, 19: 322-333 (1988).

Koehler, G.J., "Characterization of unacceptable solutions in LP discriminant analysis", ***Decision Sciences***, 21: 239-257 (1989).

Koehler, G.J., Erenguc, S.S, "Minimizing misclassifications in linear discriminant problem", ***Decision Sciences***, 21: 63-85 (1990).

Lachenburch P.A., "Discriminant analysis", ***Hafner Press***, New York, USA, 40-90 (1975).

Lam, K.F., Choo, E.U., Moy, J.W., "Minimizing deviations from the group mean: A new linear programming approach for the two-group classification problem", ***European Journal of Operational Research***, 88: 358-367 (1996).

Lam, K.F., Moy, J.W., "Improved linear programming formulations for the multi-group discriminant problem", ***Journal of Operational Research Society***, 47: 1526-1529 (1996).

Lam, K.F., Moy, J.W., "An experimental comparison of some linear programming approaches to the discriminant problem", ***Computers and Operations Research***, 24 (7), 593-599 (1997).

Lam, K.F., Moy, J.W., "Combining discriminant method in solving classification problems in two-group discriminant analysis", ***European Journal of Operational Research***, 138: 294-301 (2002).

Lee, C.K., Ord, J.K., "Disciminant analysis using least absolute deviations", ***Decision Sciences***, 21: 86-96 (1990).

Leshno, M., Spector, Y., "Neural network prediction analysis: The bankrupty case", ***Neurocomputing***, 10: 125-147 (1996).

Loucopoulos, C., Pavur, R., "Computational characteristics of a new mathematical programming model for the three-group discriminant problem", ***Computers and Operations Research***, 2: 179-191 (1997).

Markowski, E.P., Markowski, C.A., "Some difficulties and improvements in applying linear programming formulations to the discriminant problem", ***Decision Sciences***, 16: 237-247 (1985).

Marks, S., Dunn, O.J., "Discriminant functions when covariance matrices unequal", ***Journal of American Statistical Association***, 69 (3): 555-559, (1974) .

Mengiameli, P., West, D., "An improved neural classification network for the two group problem", ***Computers and Operations Research***, 26: 943-960 (1999).

Östermark, R., Höglund, R., "Adressing the multigroup discriminant problem using multivariate statistics and mathematical programming", ***European Journal of Operational Research***, 108: 224-237 (1998).

Öztemel, E., "Yapay sinir ağları", ***Papatya Yayıncılık***, İstanbul, 40-100 (2003).

Patterson, D.W., "Artificial neural networks", ***Prentice Hall***, Singapore, 10-50 (1996).

Pendharkar, P.C., "A threshold varying artificial neural network approach for classification and its application to bankruptcy prediction problem", ***Computers and Operations Research***, 32: 2561-2582 (2005).

Pavur, R., Loucopoulos, C., "Examining optimal criterion weights in mixed integer programming approaches to the multi group classification problem", ***Journal of Operational Researh Society***, 46: 626-640 (1995).

Patwo, E., Hu, M.Y., Hung, M.S., "Two-group classification using neural networks", ***Decision Sciences***, 24 (4): 825-845 (1993).

Rubin, A., "A comparison of linear programming and parametric approaches to the two-group discriminant problem", ***Decision Sciences***, 21: 373-386 (1990).

Sağıroğlu, Ş., Beşdok, E., Erler, M., "Mühendislikte yapay zeka uygulamaları-I: yapay sinir ağları", ***Ufuk Yayıncılık***, Kayseri, 35-70 (2003).

Stam, A., Jones, D.G., "Classification performance of mathematical programming techniques in discriminant analysis: Results for small and medium sample sizes", ***Managerial and Decision Economics***, 11: 243-253 (1990).

Stam, A., Ragsdale, C.T., "On the classification gap in mathematical programming based approaches to the discriminant problem", ***Naval Research Logistic***, 39: 545-559 (1992).

Sueyoshi, T., "DEA-Discriminant analysis in the view of goal programming", ***European Journal of Operational Research***, 115: 564-582 (1999).

Sueyoshi, T., "Extended DEA-Discriminant analysis", ***European Journal of Operational Research***, 131: 324-351 (2001).

Sueyoshi, T., "Mixed integer programming approach of extended DEA-Discriminant analysis", ***European Journal of Operational Research***, 152: 45-55 (2004).

Sueyoshi, T., "DEA-Discriminant analysis: Methodological comparison among eight discriminant analysis approaches", ***European Journal of Operational Research***, 169: 247-272 (2006).

Şen, Z., "Yapay sinir ağı ilkeleri", ***Su Vakfı Yayınları***, İstanbul, 20-75 (2004a).

Şen, Z., "Genetik algoritmalar ve en iyileme yöntemleri", ***Su Vakfı Yayınları***, İstanbul, 10-80 (2004b).

Tam, K.Y., Kiang, M.Y., "Managerial application of neural networks: The case of bank failure predictions" ***Management Sciences***, 38 (7): 926-947 (1992).

İnternet: University of California-Irvine "İki ve Çok Gruplu Gerçek Sınıflandırma Problemleri" www.ics.uci.edu/mlearn/~MLRepository.html (2009).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖRKCÜ, H.Hasan
Uyruğu : T.C.
Doğum tarihi ve yeri : 17.09.1980 / Yeşilova-AKSARAY
Medeni hali : Evli
Telefon : 0 (312) 202 14 86
e-mail : hhorkcu@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi /İstatistik Bölümü	2004
Lisans	Gazi Üniversitesi/ İstatistik Bölümü	2001
Lise	Yeşilova Lisesi/AKSARAY	1997

İş Deneyimi

Yıl	Yer	Görev
2002–2009	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar:

Uluslararası Hakemli Dergilerde Yayınlanan Makaleler (SCI, SCIE)

1. Bal, H., Örkcü, H.H., Çelebioğlu S, "Improving the Discrimination Power and Weights in the Data Envelopment Analysis", ***Computers&Operations Research***, April, 2009 (in press).

2. Bal, H., Örkcü, H.H., Çelebioğlu S, "A New Method Based on the Dispersion of Weights in Data Envelopment Analysis", ***Computers&Industrial Engineering***, 54(3), 502-512, 2008.

3. Bal, H., Örkcü, H.H., "Data Envelopment Analysis Approach to Two-Group Classification Problems and an Experimantal Comparison with Some Classification Models", ***Hacettepe Journal of Mathematics and Statistics***, 36(2), 169-180, 2007.

4. Bal, H., Örkcü, H.H., Çelebioğlu S, "An Experimental Comparison of the New Goal Programming and Linear Programming Approaches in the Two-Group Discriminant Problems", ***Computers&Industrial Engineering***, 50 (3), 296-311, 2006**.**

Uluslararası Diğer Hakemli Dergilerde Yayınlanan Makaleler (Inspec, Scopus)

5. Örkcü, H.H., Bal, H. "A Combining Mathematical Programming Method for Multi-Group Data Classification", ***G.U. Journal of Science***, düzeltme incelemede, 2009.

6. Bal, H., Örkcü, H.H. "A Goal Programming Approach to Weight Dispersion Problem in Data Envelopment Analysis and Simulation Study", ***G.U. Journal of Science***, 20 (4), 117-125, 2007.

7. Bal, H., Örkcü, H.H., Çelebioğlu S, "An Alternative Model to Fisher and Linear Programming Approaches in Two-Group Classification Problem: Minimizing Deviations from the Group Median", ***G.U. Journal of Science***, 19 (1), 49-55, 2006.

8. Kardiyen, F., Örkcü, H.H., "The Comparison of Principal Component Analysis and Data Envelopment Analysis in Ranking of Decision Making Units", ***G.U. Journal of Science***, 19 (2), 127-133, 2006.

9. Bal, H., Örkcü, H.H., "Combining the Discriminant Analysis and Data Envelopment Analysis in view of Multiple Criteria Decision Making: A New Model", ***G.U. Journal of Science*****,** 18 (3), 355-364, 2005.

Ulusal Hakemli Dergilerde Yayınlanan Makaleler

10. Örkcü, H.H., Alp, İ., "Düzenli Hiyerarşik Karar Verme Birimlerinin Etkinlik Değerlendirmesi Üzerine Bir Monte Carlo Çalışması", ***Milli Prodüktivite Merkezi, Verimlilik Dergisi***, 2007/3, 41-56, 2007.

11. Örkcü, H.H., Kardiyen, F. "İllerin Gelişmişlik Düzeylerini Sıralama ve Sınıflama Bakımından Veri Zarflama Analizi ve Çok Değişkenli İstatistiksel Yöntemlerin Karşılaştırılması Üzerine Bir Çalışma", ***Hacettepe Üniversitesi İ.İ.B.F. Dergisi***, 24(2), 127-152, 2006.

12. Bal, H., Örkcü, H.H., "Veri Zarflama Analizi'nde Karar Verme Birimlerinin Sıralanması İçin Sınıflandırma Kriteri Tabanlı Yeni Bir Model", ***TUİK İstatistik Araştırma Dergisi***, 4(2), 15-25, 2005.

Ulusal Bilimsel Toplantılarda Sunulan Bildiriler

13. Bal, H., Örkcü, H.H., "Ülkelerin Sosyo-Ekonomik Etkinliklerinin Farklı Veri Zarflama Analizi Sıralama Yöntemleri İle İncelenmesi", ***18. İstatistik Araştırma Sempozyumu***, Türkiye İstatistik Kurumu, Ankara, 2009.

14. Örkcü, H.H., Bal, H., "İki Gruplu Ayırma Analizine Matematiksel Programlama ve Yapay Sinir Ağları Yaklaşımları", ***VI. İstatistik Günleri Sempozyumu***, Ondokuzmayıs Üniversitesi, Samsun, 28-30 Ağustos 2008.

15. Bal, H., Örkcü, H.H.**,** "Veri Zarflama Analizinde Ağırlık Dağılımının İyileştirilmesi", ***VI. İstatistik Günleri Sempozyumu***, Ondokuzmayıs Üniversitesi, Samsun, 28-30 Ağustos 2008.

16. Bal, H., Örkcü, H.H., "Çok kriterli Veri Zarflama Analizi modeli ile Türkiye ve Avrupa Birliği Ülkelerinin Ekonomik Etkinliklerinin İncelenmesi", ***16. İstatistik Araştırma Sempozyumu***, Türkiye İstatistik Kurumu, Ankara, 2007.

17. Bal, H., Örkcü, H.H., "Temel Bileşenler ile Veri Zarflama Analizinin Karar Verme Birimlerinin Sıralanmasında Kullanılması", ***15. İstatistik Araştırma Sempozyumu***, Türkiye İstatistik Kurumu, Ankara, 2006.

18. Çelebioğlu, S., Olmuş, H., Örkcü, H.H., Gökpınar, F., "Ankara'nın Başkent Oluşunun 80. yılında Nüfus İstatistikleri Üzerine Bir Çalışma", ***Cumhuriyetin 80. yılı Sempozyumu***, Ankara, 2003.

Eserlerine Yapılan Atıflar

- **Atıf Yapılan Makale:**

Bal, H., Örkcü, H.H., Çelebioğlu S, "An Experimental Comparison of the New Goal Programming and Linear Programming Approaches in the Two-Group

Discriminant Problems", ***Computers and Industrial Engineering***, 50(3), 296-311, 2006.

Atıf Yapan:

Xu, G., Papageorgiou, L.G., "A Mixed Integer Optimization Model for Data Classification", ***Computers and Industrial Engineering***, article in press, 2009.

Ekinci, Y., "Demand Assignment: A DEA and Goal Programming Approach", ***12th WSEAS International Conference on Applied Mathematics,*** Cairo, Egypt, December, 29-31, page: 394-397, 2007.

- **Atıf Yapılan Makale:**

Bal, H., Örkcü, H.H., Çelebioğlu S, "A New Method Based on the Dispersion of Weights in Data Envelopment Analysis", ***Computers and Industrial Engineering***, 54(3), 502-512, 2008.

Atıf Yapan:

Wang, Y.M., Luo, Y., "A note on a new method based on the dispersion of weights in data envelopment analysis", ***Computers and Industrial Engineering***, article in press, 2009.

- **Atıf Yapılan Makale:**

Bal, H., Örkcü, H.H. "A Goal Programming Approach to Weight Dispersion Problem in Data Envelopment Analysis and Simulation Study", ***G.U. Journal of Science***, 20 (4), 117-125, 2007**.**

Atıf Yapan:

Segota, A, "Evaluating shops efficiency using data envelopment analysis: Categorical approach", ***Zbornik Radova Ekonomskog Fakulteta U Rijeci-Proceedings of Rijeka Faculty of Economics***, 26 (2), 325–343, 2008.

Hobiler

Türk sineması, Türk halk müziği, Bilgisayar teknolojileri, Futbol.