



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YARASA ALGORİTMASI KULLANARAK YAPAY SİNİR AĞLARININ EĞİTİLMESİ

Lubna Luay Kamal KAMAL

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Mayıs-2018
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

Yarasa Algoritması Kullanarak Yapay Sinir Ağlarının Eğitilmesi

Lubna Luay Kamal KAMAL

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Halife KODAZ

2018, 62 Sayfa

Jüri
Doç.Dr. Halife KODAZ
Dr.Öğr.Üyesi. Ömer Kaan BAYKAN
Dr.Öğr.Üyesi. Şaban GÜLCÜ

Son zamanlarda araştırmacılar tarafından doğadan ilham alınan birçok meta-sezgisel algoritmalar geliştirilmektedir. Bu meta-sezgisel algoritmalardan birisi sürü zekasına dayalı ve popülasyon tabanlı olan Yarasa algoritmasıdır. Yarasa algoritması birçok alanda kullanılmaktadır. Bu tez çalışmasında Yarasa algoritması yapay sinir ağlarının eğitimi için kullanılmıştır. Yapay sinir ağlarının sınıflandırma başarısını etkileyen birden fazla faktör bulunmaktadır. Bunlardan en önemlisi yapay sinir ağlarının ağırlıklarının belirlenmesidir. Bu tez çalışmasında yarasa algoritması yapay sinir ağlarının ağırlıklarının güncellenmesinde kullanılmıştır. Önerilen yöntemin performansını değerlendirmek için UCI veri ambarından alınan 10 adet veri kümesi üzerinde testler yapılmıştır. Aynı zamanda yarasa algoritmasının parametrelerinden ses şiddeti, sinyal yayma oranı ve yarasa sayısı parametrelerinin analizi yapılmıştır. En iyi sınıflandırma doğruluğunu veren parametre değerleri tespit edilmiştir. Elde edilen sonuçlar literatürde yapılan çalışmalarla karşılaştırılmıştır. Deneysel sonuçlara göre önerilen sınıflandırma yöntemi 6 adet veri kümesinde en iyi sınıflandırma doğruluğuna ulaşmıştır, 4 adet veri kümesinde ise makul sınıflandırma doğruluğuna ulaşmıştır.

Anahtar Kelimeler: Yarasa Algoritması, Meta-Sezgisel Algoritmalar, Optimizasyon, Yapay Sinir Ağları, Sınıflandırma

ABSTRACT

MS THESIS

Training Artificial Neural Networks Using Bat Optimization Algorithm

Lubna Luay Kamal KAMAL

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING

Advisor: Assoc.Prof.Dr. Halife KODAZ

2018, 62 Pages

Jury
Assoc.Prof.Dr. Halife KODAZ
Asst.Prof.Dr. Ömer Kaan BAYKAN
Asst.Prof.Dr. Şaban GÜLCÜ

Recently, many meta-heuristic algorithms that are inspired by the nature have been developed by researchers. One of these meta-heuristic algorithms is the Bat algorithm, which is based on swarm intelligence and population. The bat algorithm has been used in many areas. In this thesis study, Bat algorithm is used for the training of artificial neural networks. There are many factors that affect the classification success of artificial neural networks. The most important of them is the determination of the weights of artificial neural networks. In this thesis study, the bat algorithm was used to update the weights of artificial neural networks. In order to evaluate the performance of the proposed method, tests were performed on 10 data sets taken from the UCI data warehouse. At the same time, the parameters of the bat algorithm were analyzed for loudness, signal emission rate and number of bats parameters. Parameter values giving the best classification accuracy have been determined. The results obtained are compared with the literature studies. According to the experimental results, the proposed classification method has reached the best classification accuracy in 6 data sets and the reasonable classification accuracy in 4 data sets.

Keywords: Bat Algorithm, Meta-Heuristic Algorithms, Optimization, Artificial Neural Network, Classification

ÖNSÖZ

Bu tez çalışmamın gerçekleştirilmesinde beni yönlendiren, çalışmanın hazırlama sürecinin her adımında, tecrübelerini, desteklerini ve değerli zamanını esirgemeyen değerli danışman hocam Sayın Doç.Dr. Halife KODAZ'a teşekkürü borç bilirim.

Tüm eğitim hayatım boyunca desteklerini esirgemeyen ve bugünlere gelmemi sağlayan, haklarını hiçbir zaman ödeyemeyeceğim Babam Luay KAMAL, Annem Gulsan FOUAD ve kardeşim Luma LUAY' a sonsuz teşekkür ederim.

Üzerimde çok fazla emeği olan ve beni her zaman destekleyen dayım Nevzat ESEN'e sonsuz teşekkür ederim

Çalışmam süresince manevi desteğini benden esirgemeyen Ümit SAİT'e teşekkür ederim

Çalışmam boyunca benden yardımlarını esirgemeyen Muhammed İBRAHİM'e teşekkür ederim

Lubna Luay Kamal KAMAL
KONYA-2018

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

KISALTMALAR ..... **viii**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **4**

3. MATERYAL VE YÖNTEM **10**

3.1. Yapay Sinir Ağları 10
3.1.1. Biyolojik sinir ağları 10
3.1.2. YSA özellikleri 11
3.1.3. YSA'nın avantajları ve dezavantajları 12
3.1.4. YSA'nın kullanım alanları 13
3.1.5 Yapay sinir hücresi 14
3.1.6. YSA sınıflandırılması 16
3.1.7. Temel öğrenme kuralları 27
3.2. Optimizasyon 28
3.3. Yarasa Algoritması (YA) 32
3.3.1. Yarasaların ekolokasyon kabiliyeti 32
3.3.2. YA yapısı 33
3.3.3. Yarasaların hareketi 34
3.3.4. Ses şiddeti ve sinyal yayma oranı 35
3.4. Performans Değerlendirme Ölçütleri 35
3.5. Önerilen Yöntem 38

4. DENEYSEL SONUÇLAR **41**

4.1. Tezde Kullanılan Veri Kümeleri 41
4.2. Parametre Analizi 44
4.3. Karşılaştırma Sonuçları 51

5. SONUÇ **57**

KAYNAKLAR **58**

ÖZGEÇMİŞ **62**

KISALTMALAR

AB	: Ateş Böceği
BTO	: Biyocoğrafya Tabanlı Optimizasyon
CEP	: Sınıflandırma Hata Yüzdesi
ÇKA	: Çok Katmanlı Algılayıcı
ÇKİBYSA	: Çok Katmanlı İleri Beslemeli YSA
DDE	: Değiştirilmiş Diferansiyel Evrim
DE	: Diferansiyel Evrim
DN	: Doğru Negatif
DP	: Doğru Pozitif
GA	: Genetik Algoritması
GKA	: Guguk Kuşu Arama
GY	: Geri Yayılım algoritması
GYA	: Geliştirilmiş Yarasa Algoritması
HA	: Harmoni Arama
HF	: Havai Fişek Algoritması
İBYSA	: İleri Beslemeli YSA
KÇO	: Kuş Çiftleşme Optimizasyonu
KHA	: Krill Herd Algoritması
KKO	: Karınca Koloni Algoritması
LM	: Levenberg-Marquardt algoritması
MSE	: Ortalama Karesel Hata
PSO	: Parçacık Sürü Optimizasyonu
RBF	: Radyal Tabanlı Fonksiyon
SD	: Sınıflandırma Doğruluğu
SÖO	: Sosyal Örümcek Optimizasyonu
SS	: Standart Sapma
SSE	: Toplam Karesel Hata
YA	: Yarasa Algoritması
YAKA	: Yapay Arı Koloni Algoritması
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
YSA	: Yapay Sinir Ağları

1. GİRİŞ

Yapay Sinir Ağı (YSA), insan beyninin öğrenme prosedürünü simüle etmeye çalışan yapay bir sistemdir. YSA, olayların örneklerine bakmakta, bu örneklerden yararlanarak olay hakkında genelleme yapmakta, bilgiler toplamakta ve daha sonra hiç görmediği örneklerle karşılaştığında öğrendiği bilgileri kullanarak karar verebilmektedir (Öztemel, 2003). YSA öğrenme yoluyla yeni bilgiler üretebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir (Yurdakul, 2014). Bir sinir ağı, birbirine bağlı bir yapay nöron grubundan oluşmaktadır. YSA temel olarak biri eğitim, diğeri test olmak üzere iki aşamada değerlendirilir. İlk olarak, eğitim verisi YSA'yı eğitmek için kullanılır ve eğitilen YSA'nın sınıflandırma performansını değerlendirmek için test verileri kullanılır (Günler, 2012).

Günümüzde YSA örneklerden öğrenebilme yeteneği, etraftan gelen olaylara karşı tepkiler üretebilme kabiliyeti, esnek olması ve güçlü yapıları nedeniyle (Ozturk, 2011) sınıflandırma, sistem modelleme, genelleme, desen tanıma, tahmin, kontrol ve optimizasyon gibi alanlarda başarılı bir şekilde kullanılmaktadır. Bu nedenle, değişik YSA mimarileri geliştirilmiş ve bu mimariler çok farklı alanlarda uygulanmıştır.

YSA öğrenme yeteneği ile karmaşık problemlerin modellenmesinde oldukça önemli bir araç olmaktadır (Özdemir, 2010). YSA'nın en önemli özelliği, kendisine tanıtılan bilgileri öğrenerek edindiği deneyimlerle farklı durumlar için çıkarsama yapabilme yeteneğidir (Yurdakul, 2014). Oldukça zor bir optimizasyon problemi olan YSA'nın öğrenmesi süreci ağ mimarisindeki ağırlıklara uygun değerlerin bulunması işlemidir. YSA'nın mükemmel bir şekilde öğrenmesini sağlayacak bağlantı ağırlıklarının belirlenmesi yani YSA'nın eğitilmesi bilim adamlarının son zamanlarda çalıştığı optimizasyon problemlerinden bir tanesidir (Whitley ve ark., 1990; Verma ve Ghosh, 2002). YSA'nın iki türü vardır ve bunlar ileri beslemeli ve geri beslemeli YSA'dır. En çok kullanılan YSA mimarisi ileri beslemeli Çok Katmanlı Algılayıcı (ÇKA) YSA modelleridir. Birçok öğrenme algoritması YSA'da ağırlıkların belirlenmesinde kullanılmaktadır (Özdemir, 2010). En çok kullanılan öğrenme algoritmalarından birisi Geri Yayılım (GY) algoritmasıdır. GY algoritması eldeki veri ile ağın çıktısı arasındaki farka dayalı olarak ağırlıkların güncellenmesini gerçekleştirir

(Özdemir, 2010). Sınıflandırma amaçlı kullanılan YSA'nın en çok tercih edilenleri ileri beslemeli YSA, vektör kuantalamalı öğrenme ağları ve radyal tabanlı fonksiyon ağlarıdır (Ozturk, 2011).

YSA veri madenciliğin en önemli sınıflandırma algoritmalarından biridir. YSA'da sınıflandırma başarısını etkileyen birden fazla faktör vardır. Bunların birisi ağırlıkların güncellenmesidir. Bu tezde YSA'nın sınıflandırma başarısını artırmak için ağırlıkların güncellenmesi üzerine çalışılmıştır.

Optimizasyon, herhangi problem için, verilen şartlar altında tüm çözümler içinden en iyi çözümü elde etme işlemidir (Alataş, 2007). Genel olarak günlük hayatın her alanında karşılaşabileceğimiz bir kavram olduğu için optimizasyonun ne amaçladığını iyi bilmemiz gerekir (Canayaz, 2015). Örneğin bir kurumda kârın maksimum düzeyde olması ve aynı zamanda giderlerin minimuma indirilmesi veya çalışan bir sistemin optimum kararlılıkta çalışması herkes tarafından istenen bir durumdur (Canayaz, 2015; Canayaz ve Karcı, 2015). Meta-sezgisel algoritmalar en iyi sonuç aramada son zamanlarda popüler hale gelmiştir, ve birçok optimizasyon problemlerinde kullanılmaktadır.

Doğadaki canlılar bilime ilham vermiştir ve çoğu icadın ortaya çıkmasında büyük rolleri vardır. Bilim insanları bu canlıların davranışlarının taklidi yoluyla birçok algoritma geliştirmiş ve bu algoritmaları problemlerin çözümünde kullanmış ve kullanmaya da devam etmektedir. Sürü zekâsı algoritmaları, doğada sürü halinde yaşayan bazı canlıların besin arama davranışları incelenerek geliştirilen meta sezgisel algoritmalardır. Sürü kelimesi birlikte ve işbirliği halinde yaşayan canlı gruplarını ifade etmektedir (Özkış, 2013). Genetik Algoritma (GA), Karınca Koloni Algoritması (KKO), Parçacık Sürü Optimizasyonu (PSO), Diferansiyel Evrim Algoritması (DE), Yapay Arı Koloni Algoritması (YAKA) başta olmak üzere literatürde çok sayıda doğa esinli meta-sezgisel optimizasyon algoritması mevcuttur. Bu algoritmalardan birisi de Yarasa Algoritmasıdır (YA).

YA, yarasaların ekolokasyon adı verilen sesin yankılanmasından faydalanılarak bir cismin bulunduğu yön ve uzaklık saptama davranışından esinlenen bir popülasyon tabanlı optimizasyon algoritmasıdır (Yılmaz, 2014). YA, 2010 yılında Yang (Yang, 2010b) tarafından sürü zekasına dayanan ve meta-sezgisel bir optimizasyon algoritması olarak önerilmiştir.

Bu tez çalışmasında YSA'nın sınıflandırma başarısını artırmak için ağırlıkların güncellenmesi işleminde YA kullanılmıştır. Tez beş ana bölümden oluşmaktadır.

Birinci bölümde teze giriş yapıldıktan sonra ikinci bölümde tezle ilgili kaynak araştırması verilmiştir. Üçüncü bölümde tezde kullanılan materyal ve yöntem hakkında bilgi verilmiştir. Bu bölümde YSA, optimizasyon, YA, performans değerlendirme ölçütleri ve tezde önerilen yöntem olan YA tabanlı YSA hakkında detaylı bilgi verilmiştir. Dördüncü bölümde kullanılan veri kümeleri, önerilen yöntemin parametre analizleri ve önerilen yöntemin sınıflandırma başarısının literatürle karşılaştırıldığı deneysel sonuçlar verilmiştir. Son olarak beşinci bölümde tezin sonuçları verilmiştir.

2. KAYNAK ARAŞTIRMASI

Bu bölümde tezle ilgili literatürdeki çalışmaların kaynak araştırması verilmiştir. Öncelikle YSA ile ilgili çalışmalardan bahsedildikten sonra YA ile ilgili çalışmalar hakkında bilgi verilmiştir.

Güler ve Übeyli (2006) dört değişik öğrenme algoritması ile eğitilen ÇKA sinir ağları diyabet teşhisi için kullanmış ve en başarılı öğrenme algoritması tespit edilmiştir. Bu hastalığın tahmini için en başarılı algoritma hızlı yayılım algoritması olmuştur.

Zhang ve ark. (2007) PSO algoritmasını GY algoritması ile birleştiren hibrid bir algoritma olarak önermişlerdir ve PSO-GY algoritması olarak ifade etmişlerdir. Önerilen algoritma İleri Beslemeli Sinir Ağının (İBYSA) ağırlıklarını eğitmek için kullanılmıştır. Bu hibrid algoritma, PSO'nun global arama becerisini ve GY algoritmasının güçlü lokal yeteneğini birleştirmiştir. Deneysel sonuçlara göre aynı hedef için PSO-GY algoritmasının PSO algoritması ve GY algoritmasından daha az zaman harcadığını ve önerilen hibrid PSO-GY algoritmasının Adaptif Parçacık Sürü Optimizasyon Algoritması ve GY algoritmasından daha kararlı ve daha iyi olduğunu göstermiştir.

Carvalho ve Ludermir (2007) İBYSA mimarisi ve ağırlıklarının optimizasyonu için PSO algoritmasının kullanımını analiz etmiştir. Önerilen algoritmayı değerlendirmek için üç veri kümesi (kanser, diyabet, kalp) kullanılmıştır. Sonuçlar, PSO'ya dayalı bu yaklaşımın İBYSA'nın ağırlıklarını ve mimarilerini optimize etmek için uygun bir alternatif olduğunu göstermiştir.

Delice (2008) yapmış olduğu tez çalışmasında eğitilmiş YSA'dan sınıflandırma kurallarını çıkarmak için ikili bir PSO algoritması geliştirmiştir. Önerilen algoritmanın performansını değerlendirmek için UCI Makine Öğrenmesi Veri Ambarından alınan 11 veri kümesi kullanılmıştır. Test veri kümeleri üzerinde yapılan analizlerle, önerilen algoritmanın doğru ve etkili sınıflandırma kuralları üretebileceği gösterilmiştir.

Mirjalili ve ark. (2012) PSO ve Yerçekimi Arama algoritmasının hibrid bir şekilde kullanmıştır. Hibrid yöntem İBYSA için yeni eğitim yöntemi olarak kullanılmıştır. Sonuçlar literatürdeki çalışmalarla karşılaştırılmıştır. Deneysel sonuçlara göre önerdikleri yöntem, yakınsama oranı ve yerel minimumdan kaçınma açısından İBYSA'ları eğitmek için iyi performans vermiştir.

Günler (2012) tez çalışmasında YSA'ya dayalı yüz ifadelerini tanımlamak için üç farklı yaklaşım önermiştir. Başlangıç olarak ağaç temelli sinir ağları yapısı önermiş sonra duygu sınıflandırılması için gizli katman çıktıları kullanmış ve en son olarak yüz niteliklerine dayalı bir sistem tasarlamıştır. Önerilen yöntemleri Matlab kullanılarak uygulamış ve her biri gülen, kızgın ve sinirli yüz ifadeleri başarılı bir şekilde tanınabilmiştir.

Yaghini ve ark. (2013) YSA'yı eğitmek için yeni bir yöntem sunmuştur. PSO ve GY algoritmasını birlikte kullanarak Meta-sezgisel ve gradyan tabanlı algoritmaların yeteneğini birleştirmiştir. Önerilen yöntemin etkinliği, UCI Makine Öğrenmesi Veri Ambarından alınan kanser, kart, diyabet, gen, kalp, at, iris ve tiroit veri kümeleri üzerinde birkaç YSA eğitim algoritması ile karşılaştırmıştır. Önerilen yöntem ile diğer yöntemler arasındaki test sınıflandırma doğruluk oranı ve tüm veri kümeleri üzerinde tahmini için bazı veri madenciliği teknikleri açısından bir karşılaştırma sunmuştur. Sonuçların değerlendirilmesi, önerilen algoritmanın üstünlüğünü göstermiştir.

Askarzadeh ve Rezazadeh (2013) çalışmalarında İBYSA'yı eğitmek için yakın zamanda keşfedilen meta-sezgisel optimizasyon algoritması olan Kuş Çiftleşme Optimizasyonunu (KÇO) kullanmıştır. Önerilen algoritmanın Iris, Kanser ve Diyabet veri kümesi olmak üzere üç gerçek dünya sınıflandırma probleminin çözümünde YSA'nın ağırlık eğitimine uygulanmıştır. KÇO'nun performansı diğer sınıflandırıcılarla karşılaştırılmıştır. Deneysel sonuçlara göre diğer sınıflandırıcılarla karşılaştırıldığında umut verici sonuçlar KÇO'nun potansiyelini doğrulamıştır.

Ahadzadeh ve Menhaj (2014) İBYSA için yeni bir eğitim algoritması olarak Değiştirilmiş Diferansiyel Evrim (DDE) algoritması kullanmıştır. Bu yeni öğrenme algoritmasının etkinliğini ve verimliliğini değerlendirmek için, UCI Makine Öğrenmesi Veri Ambarından alınan sekiz veri kümesi kullanılmıştır. DDE'den elde edilen sonuçlar da GY algoritması, Global Harmoni Arama, Adaptif Harmoni Arama, Genetik Algoritma-Geri Yayılım ve Yapay Arı Koloni Algoritması-Geri Yayılım algoritmalarıyla karşılaştırmıştır. Deneysel sonuçlara göre DDE algoritmasının çeşitli sınıflandırma problemlerini çözmek için YSA'yı başarılı bir şekilde eğitebildiğini göstermiştir.

Uzlu ve ark. (2014) yıllık hidrolik enerji üretimini tahmin etmek için YAKA ile YSA modelini uygulamışlardır. YAKA-YSA modeli performansı, GY algoritması ile eğitilmiş klasik YSA modellerinin sonuçları ile karşılaştırmıştır. Türkiye için gelecekteki hidroelektrik üretim değerlerini tahmin etmek için önerilen üç senaryo

altında çeşitli göstergeleri temel alınarak modellemiştir. Sonuçlar, YAKA-YSA yönteminin hidroelektrik üretimi için GY algoritması ile eğitilen klasik YSA'dan daha iyi tahmin ettiğini göstermiştir.

Sulaiman ve ark. (2015) Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı (ÇKİBYSA) ile Ateş Böceği (AB) algoritmasının hibrid olarak kullanıldığı bir yöntem önermiştir. Yöntem güç modellenmesi için kullanılmıştır. AB, gizli katmandaki optimum nöron sayısını, öğrenme hızını ve momentum oranını belirlemede başarılı bir şekilde kullanılmış. Böylece modelin Ortalama Karesel Hatası (MSE) en aza indirilmiştir. Daha sonra ÇKİBYSA'nın performansı güç modellemesinde Klasik Evrimsel Programlama ÇKİBYSA'nın performansıyla karşılaştırmıştır. Sonuçlara göre ÇKİBYSA hem eğitim hem de test sırasında düşük MSE üretmiştir.

Chen ve ark. (2015) hem PSO hem de Guguk Kuşu Arama (GKA) algoritmalarının avantajlarından faydalanmak için PSO ve GKA dayalı hibrid bir algoritma önermiştir. Önerilen hibrid algoritma, İBYSA için yeni bir eğitim yöntemi olarak kullanılmaktadır. Önerilen algoritmanın performansını araştırmak için, iki problem kullanılmış ve sonuçlar orijinal PSO ve GKA algoritmaları ile eğitilmiş İBYSA'dan elde edilen sonuçlarla karşılaştırmıştır. Deneysel sonuçlara göre, önerilen hibrid algoritma PSO-İBYSA ve GKA-İBYSA'dan daha iyi performans göstermiştir.

Mirjalili ve ark. (2015) son zamanlarda önerilen Sosyal Örümcek Optimizasyonu algoritması olarak bilinen sezgisel algoritmayı kullanan yeni bir eğitim algoritması önermişlerdir. Önerilen yöntem tarafından eğitilmiş İBYSA, standart sınıflandırma veri kümesi (XOR, balon, iris, göğüs kanseri ve kalp) üzerinde kıyaslamıştır. Önerilen yöntemin sonuçları literatürde bulunan beş algoritma ile karşılaştırmıştır. Deneylerde önerilen yöntemin diğer algoritmalara kıyasla çok umut verici sonuçlar verebildiği gösterilmiştir.

Liao ve ark. (2015) YSA parametrelerinin eğitimi için Nelder ve Mead'in (NM) karmaşık yöntemini ve PSO algoritması, NM-PSO olarak kısaltılan yeni ve basitleştirilmiş bir hibrid algoritması önermiştir. YSA'nın eğitiminde önerilen yöntemin performanslarını PSO ve benzeri diğer hibrid PSO yöntemleri ile karşılaştırmışlardır. Diğer benzer hibrid PSO yöntemleri ile NM-PSO algoritması dört veri kümesi ve beş gerçek dünya veri kümesi ile test edilmiş ve performansları karşılaştırılmıştır. Deneysel sonuçlara göre önerilen NM-PSO yönteminin elde edilen minimum ortalama karesel hata (MSE)'de genellikle diğer benzer yöntemlerden daha iyi performans sergilediği ve diğer algoritmalara göre daha hızlı yakınsadığı görülmüştür.

Kowalski ve Łukasik (2016) YSA'yı eğitmek için kullanılan Kril Herd Algoritmasını (KHA), GY, GA ve Harmoni Arama (HA) algoritmaları ile karşılaştırmıştır. Önerilen YSA eğitim yöntemi, sınıflandırma görevi için doğrulanmıştır. Bu amaçla, önerilen yöntem UCI Makine Öğrenmesi Veri Ambarından alınan Iris, Ionosphere, Cam ve Tiroit veri kümelerini değerlendirmede kullanmıştır. KHA'nın uygulanmasının YSA'nın eğitimi için umut verici bir performans sergilediği sonucuna varılmıştır.

Bolaji ve ark. (2016) Havai Fişek (HF) algoritmasını YSA'ların parametrelerinin eğitimi için kullanmıştır. Önerilen yöntemin performansını değerlendirmek için, UCI Makine Öğrenmesi Veri Ambarından alınan yedi veri kümesi üzerinde deneyler yapılmıştır. Önerilen yöntemle KHA, HA algoritması ve GA algoritması karşılaştırılmıştır. Değerlendirmenin sonuçları hem karesel hata toplamı (SSE) hem de Sınıflandırma Doğruluğu (SD) için önerilen algoritmanın üstünlüğünü göstermiştir. HF algoritmasının YSA'ların eğitimi için yeni algoritmalarından biri olarak kabul edilebileceği sonucuna varmıştır.

Faris ve ark. (2016) ÇKA sinir ağının eğitiminde Multi-Verse Optimizer adı verilen, son zamanlarda önerilen doğaya özgü algoritma kullanmıştır. Farklı özelliklere sahip dokuz problemi kullanılarak karşılaştırma yapılmıştır. Sonuçlar, GA, PSO, DE, AB, GKA, GY ve Levenberg-Marquardt (LM) algoritmaları olmak üzere güncel evrimsel meta-sezgisel algoritmalar ile karşılaştırılmıştır. Önerilen yaklaşımın ÇKA'larını eğitmek için veri kümelerinin çoğunda diğer eğitim algoritmalarından daha iyi performans sergilediği gösterilmiştir.

Kankal ve Uzlu (2017) Türkiye'de elektrik enerjisi talebini modellemek için öğretme-öğrenme tabanlı optimizasyon ile YSA performansını incelemiştir. Önerilen yöntem geri yayılımlı YSA (YSA-GY) ve YAKA modelleri ile karşılaştırılmıştır. Sonuçlar, önerilen yöntemin, elektrik enerjisi talebi tahmini için YSA-GY ve YAKA modellerinden daha iyi performans gösterdiğini ortaya koymuştur.

Yang (2010a) Yarasaların ekolokasyon davranışına dayalı yeni bir meta-sezgisel yöntem olan YA önermiştir. Önerilen algoritma PSO ve GA algoritması ile birkaç test fonksiyonu (Michalewicz, Rosenbrock, De Jong, Schwefel, Ackley, Rastrigin, Easom, Griewangk ve Shubert) üzerinde ve farklı problem boyutlarında karşılaştırılmıştır. Sonuçlara göre önerilen algoritmanın diğer algoritmalara göre daha üstün olduğu gösterilmiştir.

Nakamura ve ark. (2012) yarasaların davranışına dayanan, doğadan ilham alan yeni bir özellik seçme tekniği önermişlerdir. Bu yaklaşımda, özellik kümesini bulmak için yarasaların keşfedilme gücü Optimum Orman Yol sınıflandırıcısının hızı ile birleştirilmiştir. Beş veri kümesi üzerinde yapılan deneyler sonucunda önerilen tekniğin, çeşitli meta-sezgisel algoritmalara göre (PSO, Yerçekimsel Arama Algoritması ve AB algoritması) daha iyi performans sergilediği gösterilmiştir.

Fister Jr ve ark. (2013) YA'yı DE algoritması ile hibrid kullanmıştır. Sürekli problemlerin çözümü için önermişlerdir. Önerilen yöntemin performansını test etmek için 5 adet fonksiyon (Griewangk, Rosenbrock, Ackley, Rastrigin ve Sphere) kullanmışlardır. Önerilen yöntemden elde edilen sonuçlar standart YA'dan elde edilen sonuçlar ile karşılaştırılmıştır. Deneysel sonuçlara göre önerilen yöntem standart YA'dan daha iyi bir performans sergilemiştir.

Gandomi ve ark. (2013) YA'yı kısıtlı optimizasyon problemlerini çözmek için kullanmıştır. YA 13 klasik problem kullanılarak test edilmiştir. Ayrıca YA, literatürde bulunan üç kısıtlı mühendislik (Basınçlı kanal tasarımı, Kaynaklı kiriş tasarımı, Gerilme-sıkıştırma yayı tasarımı) problemine uygulanmıştır. YA'nın performansı literatürdeki algoritmalar ile kıyaslamıştır. YA tarafından elde edilen çözümlerin, mevcut yöntemlerle sağlanan çözümlerden daha iyi olduğu tespit edilmiştir.

Sakthivel ve ark. (2013) YA ekonomik dağıtım problemine uygulamışlardır. Ekonomik dağıtım problemi optimizasyon problemlerinin NP zor türündendir. Bu problemin amacı toplam yakıt maliyetini en aza indirmektir. Elde edilen sonuçlar literatürde bulunan diğer algoritmalar ile karşılaştırmıştır. Deneysel sonuçlar önerilen algoritmanın yakın zamanda bildirilen diğer algoritmalardan daha iyi sonuçlar verdiğini açıkça göstermiştir.

Wang ve Guo (2013) sayısal optimizasyon problemini çözmek için YA ile HA algoritmasını hibrid olarak kullanmıştır. Hibrid yönteme HA-YA adını vermişlerdir. Önerilen yöntemin performansını test etmek için on dört standart test fonksiyonu kullanılmıştır. Çoğu durumda, HA-YA standart YA'ya göre daha başarılı sonuçlar ürettiği veya en azından oldukça rekabetçi olduğu gözlemlenmiştir. Ayrıca önerilen yöntemin başarısı diğer popülasyon tabanlı optimizasyon yöntemleri olan Karınca Kolonisi Optimizasyonu (KKO), YA, Biyocoğrafya Tabanlı Optimizasyon (BTO), DE Algoritması, Evrim Stratejisi Algoritması (ES), GA, HA Algoritması ve PSO ile karşılaştırılmıştır. Sonuçlara göre HA-YA diğer optimizasyon yöntemlerine göre daha başarılı olmuştur.

Yılmaz (2014) tez çalışmasında YA'nın yerel ve global arama özelliklerini üç farklı yöntemle geliştirmiştir. Geliştirilmiş Yarasa Algoritmasının (GYA) performansını doğrulamak için sırasıyla 22 adet standart ve 50 adet karmaşık test fonksiyonları ile üç kısıtlı gerçek dünya problemleri (kaynaklı kiriş tasarımı, gerilim-sıkıştırma yay tasarımı ve basınçlı kap tasarımı) üzerinde test yapmıştır. Sonuçlara göre GYA, YA'ya göre daha başarılı olduğu kanıtlanmıştır. Üstelik bu yöntem, gerçek dünya problemleri ile ilgili literatürdeki çalışmalarla karşılaştırılmış ve sonuç olarak bu yöntemin literatürdeki çalışmalardan daha etkili olduğu gösterilmiştir.

3. MATERYAL VE YÖNTEM

Bu bölümde tez çalışmasında kullanılan YSA, optimizasyon, Yarasa Algoritması (YA), performans değerlendirme ölçütleri ve önerilen yarasa algoritması tabanlı YSA modeli hakkında ayrıntılı bilgi verilmiştir.

3.1. Yapay Sinir Ağları

Bu bölümde biyolojik sinir ağları ve özellikleri hakkında bilgi verildikten sonra YSA hakkında detaylı bilgi verilmektedir. YSA'nın avantajlarından ve dezavantajlarından, kullanım alanlarından ve sınıflandırılmasından bu bölümde bahsedilmektedir.

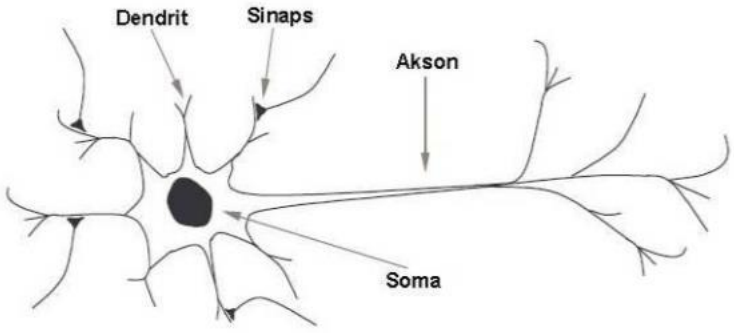
3.1.1. Biyolojik sinir ağları

İnsanın sinir sistemi yapay zekâ çalışmalarına ilham veren sistemlerden birisidir. İnsan beyninin fonksiyonlarının en önemlisi öğrenmedir (Yurdakul, 2014). YSA insanın sinir sistemini temel alarak öğrenme yoluyla yeni bilgiler öğrenebilme, yeni bilgiler üretebilme, yeni bilgiler keşfedilme gibi işlemleri gerçekleştiren sistemdir (Yurdakul, 2014).

YSA, biyolojik sinir ağlarının işlevlerinden esinlenilerek meydana gelmiştir (Ozturk, 2011) ve verimli bir bilgi işleme sistemidir (Gallo, 2015). Değişik bilim dallarından birçok bilim adamı YSA çalışmalarına katkıda bulunmuştur. Mühendisler, biyologlar, psikologlar, fizikçiler ve matematikçiler bazen kendilerine göre bazen de grup olarak işbirliğine açık olarak yapay zekâ ve YSA'nın günümüzdeki başarıya gelmelerine büyük katkıda bulunmuşlardır (Yurdakul, 2014).

Bir sinir ağı, basit işlem elemanlarının, birimlerin veya düğümlerin birbirine bağlanmasından meydana gelmektedir (Gurney, 1997; Haykin, 1999). Biyolojik sinir ağları milyarlarca sinir hücresinin (nöronların) toplanması neticesiyle meydana gelmektedir (Öztemel, 2003). İnsan beyninde tahminen 10 milyar nöron vardır ve bunların 6 trilyonu geçen bir bağlantısının veya sinapslarının bulunduğunu tahmin edilmektedir (Haykin, 2009). Şekil 3.1'de görüldüğü gibi tipik bir biyolojik nöron dört kısımdan oluşmaktadır ve bunlar dendrit, akson, soma ve sinapslardır. Dendritler bağlandıkları diğer nöronlardan bilgiyi (sinyali) almak için sorumlu olan ağaç benzeri dallardır. Soma, nöronun hücre gövdesidir ve dendrit tarafından aktarılan bilgileri

toplanmasından sorumludur ve sonra bunları aksona iletmektedir. Akson soma tarafından iletilen bilgiyi işleyerek diğer nöronlara iletmekle görevlidir. Sinaps ise akson ve diğer nöron dendritleri arasındaki bağlantıdır.



Şekil 3.1. Biyolojik Sinir Hücresi (Ozturk, 2011)

3.1.2. YSA özellikleri

YSA'nın özellikleri aşağıdaki gibi sıralanabilir;

- **Kümeleme:** Veriler arasındaki benzerliği araştırmaktadır ve benzer verileri bir kümede toplamaktadır. Bu özellik veri sıkıştırma ve veri madenciliğinde sıklıkla kullanılmaktadır.
- **Sınıflandırma / Desen tanıma:** Giriş verisinin özellikleri kullanılarak ağ yapısı ve ağırlıkları sınıflandırma yapılacak şekilde düzenlenmektedir (Öğücü, 2006b). Elde edilen yapıya test verileri giriş olarak sunularak sınıflandırma performansı analiz edilmektedir.
- **Veri ilişkilendirme:** YSA daha önce öğrendiği bilgi ile eksik bilgiyi tamamlamaktadır.
- **Tahmin:** Ağa sunulan girdilere karşılık bir çıktı değerini tahmin etmektedir. Birçok mühendislik bilim dalı fonksiyon tahminine ihtiyaç duymaktadır. Sistem dinamik olabilir ve aynı giriş tabanlı sistem durumu için farklı sonuçlar üretebilir. Günlük iş kararlarında (örneğin, satışlar, ürünler arasındaki finansal dağılım) ekonomik ve para politikasında, finans ve borsada yoğun olarak gereklidir. Tahmin farklı tıp alanlarında kullanılmaktadır örneğin hastanın bekleme süresi tahminleri; tedavi maliyeti tahmini vb.
- **Genelleme**: Başlangıç girdilerini ve ilişkilerini öğrendikten sonra, daha önce karşılaşılmamış veriler üzerinde görünmeyen ilişkileri de çıkarabilir, böylece modeli görünmeyen veriler üzerinde genelleştirip öngörülebilir hale getirebilir.

- **Bilinmeyen bilgileri işleyebilme:** YSA'nın yeteneklerinden biri de tam olmayan bilgileri işleyebilmesidir. Olayları öğrendikten sonra belirsizlikler altında ağlar öğrendikleri olaylar ile ilgili ilişkileri kurarak kararlar verebilirler (Öztemel, 2003).
- **Kendi kendine organize olma:** Bir YSA, öğrenme süresince aldığı bilginin kendi organizasyonunu veya sunumunu yaratabilir.
- **Sayısal girdiler ile çalışabilmek:** YSA sadece sayısal girdiler ile çalışmaktadır.
- **Eksik bilgi ile çalışabilmek:** YSA eksik bilgi durumunda geleneksel sistemlerin aksine eğitildikten sonra çalışabilirler ve YSA'ya gelen yeni örneklerde bilgiler eksik olduğu halde sonuç üretebilir.
- **Hataya toleranslı olma**: YSA'nın eksik bilgiler ile çalışabilme özelliği, hatalara karşı toleranslı olmasını temin etmektedir. Ağın herhangi bir hücresi bozulsa bile ağ kesintisiz çalışmaya devam etmektedir.

3.1.3. YSA'nın avantajları ve dezavantajları

YSA'nın en önemli avantajı, öğrenme yeteneğine sahip olmasıdır. YSA'nın bazı problemler ve durumlar için onları en uygun hale getiren bazı önemli avantajları vardır. YSA'nın avantajları aşağıdaki gibidir (Öztemel, 2003):

- Eksik bilgi durumunda iyi çalışabilirler.
- Hiç görmediği örnekler hakkında çözüm üretebilirler.
- YSA doğrusal olmayan ve karmaşık ilişkileri öğrenme ve modelleme yeteneğine sahiptir; bu da gerçekten önemlidir, çünkü gerçek hayatta, girişler ve çıktılar arasındaki ilişkilerin çoğu doğrusal değil karmaşıktır.
- Belirsiz, yetersiz bilgileri işleyebilirler.
- Sınıflandırma ve tahmin yapabilirler.
- Öğrenebilme kabiliyetleri vardır.
- Sorunu çözen algoritma bilgisine ihtiyaç duymazlar (otomatik öğrenir)
- Bir sinir ağı doğrusal bir programın yapamayacağı görevleri yerine getirebilir.
- Herhangi bir uygulamada uygulanabilir.
- Sorunsuz uygulanabilir.
- YSA hesaplamaları paralel yürütülebilir ve bu özelliğinden yararlanan özel donanım cihazları tasarlanır ve üretilir.

YSA'nın dezavantajları ise (Öztemel, 2003; Kulluk, 2009):

- Yalnızca nümerik bilgilerle çalışmaktadır.
- Sinir ağının çalışması için eğitim gerekmektedir.
- Büyük sinir ağları için yüksek işlem süresi gerekmektedir.
- Sonuçların nasıl anlamlı bir şekilde hesaplandığını açıklamak mümkün değildir. Bu da kara kutu gibidir.
- YSA donanım bağımlı çalışmaktadır.
- Ağ yapısını belirlemek için genellikle deneme yanılma yöntemi kullanılmaktadır.

3.1.4. YSA'nın kullanım alanları

YSA farklı alanlarda kullanılır: örnek olarak bu alanlardan bazıları aşağıda verilmiştir (Öztemel, 2003):

- Veri madenciliği
- Tahmin
- Örüntü tanıma
- Sınıflandırma
- Optimizasyon problemleri
- Desen tanıma
- Fiziksel Sistem Modellemesi
- İletişimde konuşanların tanınması
- Hepatit teşhisi
- Hatalı yazılımlardan telekomünikasyonun kurtarılması
- Denizaltı maden tespiti
- Doku analizi
- Optimizasyon Problemleri
- İflas Tahmini
- Üç boyutlu nesne tanıma
- Ekonomik Gösterge Tahminleri
- Dolandırıcılık Algılama
- Koku Analizi ve Tanımı
- Elektrik Yükü Tahmini

- Gaz veya Kömür Endeks Fiyatlarının Öngörülmesi

3.1.5 Yapay sinir hücresi

Yapay nöron beş temel elemandan oluşmaktadır. Bunlar girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılardır. Biyolojik nörona göre daha basit bir yapıya sahiptir. Şekil 3.2'de yapay sinir hücresi görülmektedir.

Girdiler harici ortamdan veya diğer işlem elemanlarından gelen verilerdir ve bunlar, ağın öğrenmesi istenen örnekler yoluyla belirlenir (Öztemel, 2003).

Yapay nöronlara gelen veriler ağırlıklar yardımı ile nörona bağlanırlar ve böylece girdilerin üretilecek çıktılar üzerindeki etkisi ayarlanabilir. Ağırlık değerleri pozitif, negatif veya sıfır olabilir (Arsan ve Çimenli, 2016)

Toplama (birleştirme) fonksiyonu ise hücreye gelen net girdiyi hesaplamaktadır. Genellikle net girdiyi hesaplamak için çeşitli toplama fonksiyonları bulunur ancak en çok kullanılan ve yaygın olanı ise ağırlıklı toplamadır (Arsan ve Çimenli, 2016) ve bu işlem Denklem 3.1'de ifade edilmektedir. Denklem 3.1'de G girdileri, A ağırlıkları, k ise hücreye gelen toplam girdi sayısını ifade etmektedir (Öztemel, 2003). Her hangi bir probleme doğru birleştirme fonksiyonu seçmek için bir formül bulunmamaktadır ancak genel olarak deneme yanılma yöntemi ile toplama fonksiyonu belirlenmektedir.

$$NET = \sum_{i}^{k} G_i A_i \tag{3.1}$$

Transfer fonksiyonu veya öğrenme eğrisi olarak da adlandırılan aktivasyon fonksiyonları (Kulluk, 2009) toplama fonksiyonundan gelen net girdiyi bazı işlemlerden geçirerek hücrenin bu girdiye karşılık gelen çıktısını tanımlamaktadır ve genellikle doğrusal olmayan bir fonksiyondur (Günler, 2012). Diğer bir ifade ile YSA'da hücrenin çıkış genliğinin istenilen değer arasında olmasını sağlamaktadır ve bu değerler genellikle [-1,1] veya [0,1] arasında değerler ile sınırlıdır (Öztemel, 2003; Yalçın, 2012). YSA'da bir aktivasyon fonksiyonu belirlerken en önemli nokta ise bu fonksiyonun türevi alınabilir olmasıdır (Kulluk, 2009). Doğrusal veya lineer (doğrusal) olmayan aktivasyon fonksiyonlarının kullanılması YSA'nın karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır (Kulluk, 2009). Genel olarak günümüzde en çok kullanılan aktivasyon fonksiyonu ise "Sigmoid fonksiyonu" dur (Arsan ve Çimenli, 2015).

Doğrusal transfer fonksiyonunun grafiği Şekil 3.3'te verilmektedir. Şekil 3.3'te görüldüğü gibi hücrenin girişi, herhangi bir değişiklik yapılmaksızın çıktıya verilmektedir (Sağirlibaş, 2010). Bu fonksiyonun formülü Denklem 3.2'de verilmektedir.

$$f(NET) = NET \tag{3.2}$$

Step (Adım) fonksiyonu perseptron tarafından kullanılan bir fonksiyondur. Hücrenin giriş değeri belirli bir eşik değerinin üstünde ise hücrenin çıkışı 1 değerini almaktadır ama eğer giriş değeri eşik değerinin altındaysa çıkış 0 değerini almaktadır. Bu fonksiyon tek veya çift kutuplu olabilir (Kulluk, 2009). Step fonksiyonun grafiği Şekil 3.3'te verilmektedir formülü ise Denklem 3.3'te verilmektedir.

$$F(NET) = \begin{cases} 1 & eğer\ NET > eşik - değer \\ 0 & aksi \qquad takdirde \end{cases} \tag{3.3}$$

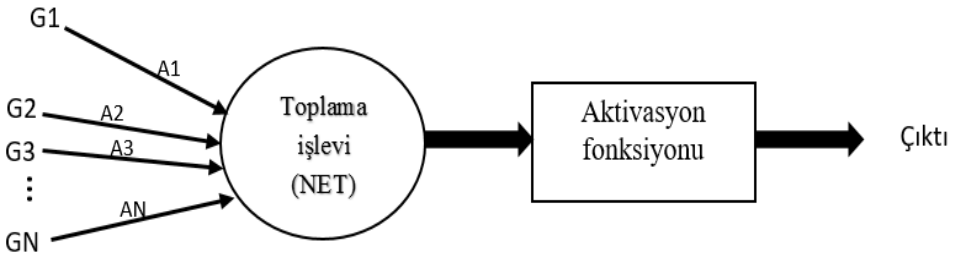
Hiperbolik tanjant fonksiyonu sigmoid fonksiyonuna benzer, bu fonksiyonun çıktıları [-1,1] değer aralığındadır (Küçüksayacıgil, 2014). Bu fonksiyonun grafiği Şekil 3.3'te formülü ise Denklem 3.4'te verilmektedir.

$$F(NET) = {(e^{NET} + e^{-NET})}\Big/{(e^{NET} - e^{-NET})} \tag{3.4}$$

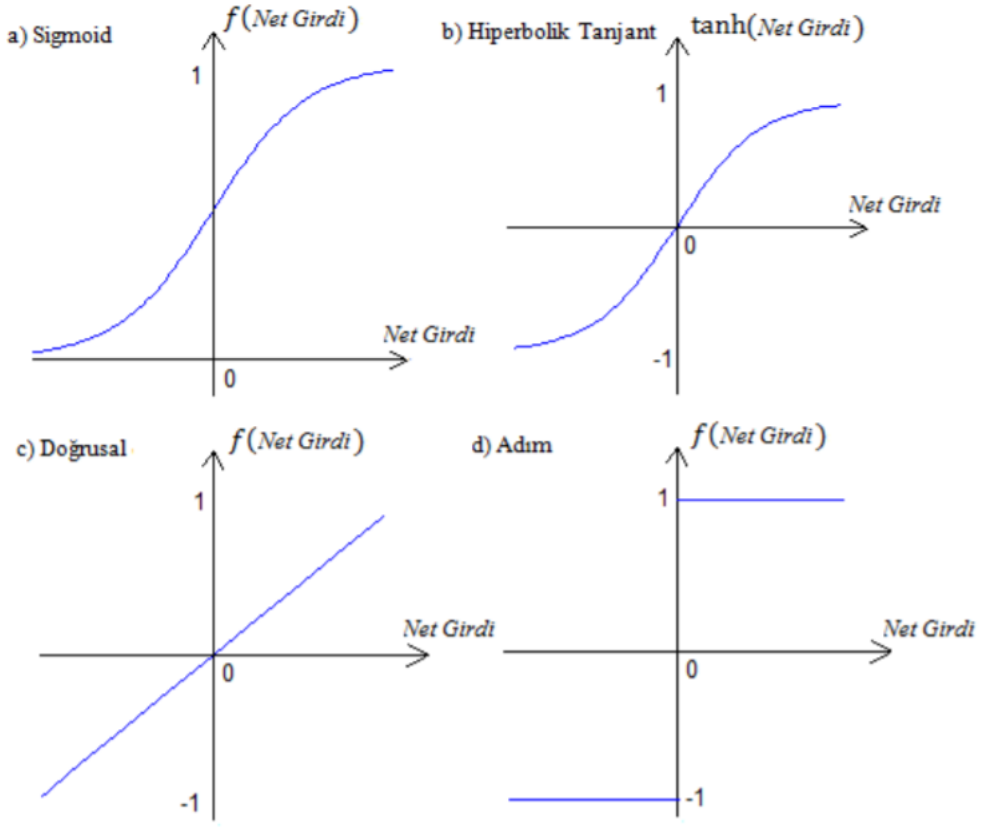
Sigmoid fonksiyonu uygulamalarda en çok kullanılan aktivasyon fonksiyonlarından biri olmasının nedeni sürekli, türevi alınabilir ve doğrusal olmayan bir fonksiyon olmasıdır. Sigmoid fonksiyonu, her girdi değeri için [0,1] aralığında bir değer üretmektedir. Bu fonksiyonun grafiği Şekil 3.3'te formülü ise Denklem 3.5'te verilmektedir.

$$F(x) = sigm(x) = \frac{1}{1 + \exp(-x)} \tag{3.5}$$

Aktivasyon fonksiyonlarının değeri hücre çıkış değeridir (Arsan ve Çimenli, 2015). Her hücrenin birden fazla girişi olabilir ancak yalnızca bir çıkışa sahiptir.



Şekil 3.2. Yapay Sinir Hücresi (Öztemel, 2003)



Şekil 3.3. Aktivasyon Fonksiyonları (Yalçın, 2012)

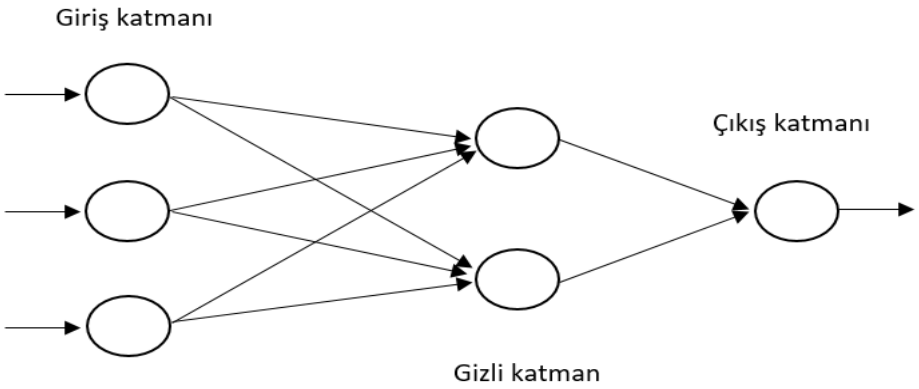
3.1.6. YSA sınıflandırılması

Literatürde YSA'nın sınıflandırılması araştırmacılar tarafından YSA'nın modeli, yapısı ve öğrenme algoritmalarına göre sınıflandırılmaktadır (Öğücü, 2006a).

3.1.6.1. YSA'nın modellerine göre sınıflandırılması

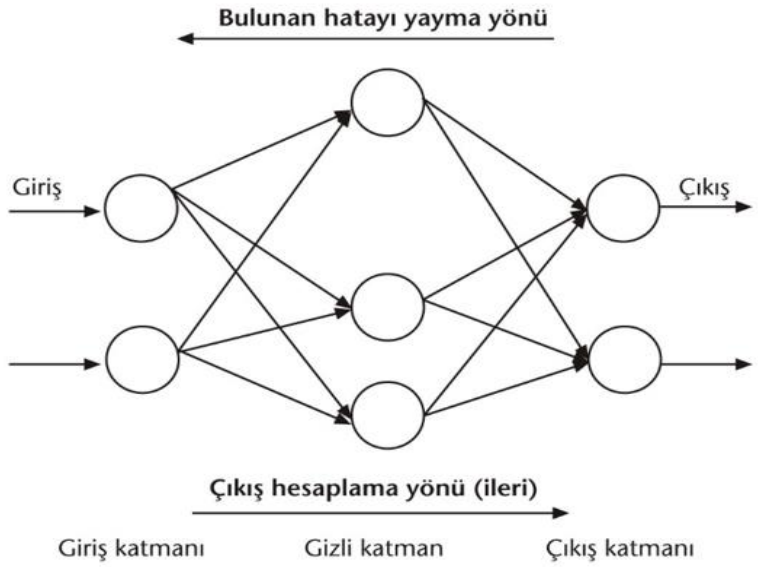
Ağ topolojisine göre, YSA modelleri iki gruba ayrılmaktadır. Bunlar ileri beslemeli ve geri beslemeli ağlar olarak sınıflandırılmaktadır.

İleri beslemeli YSA'lar birimler arasındaki bağlantıların bir döngü oluşturmadığı YSA'lardır. İleri beslemeli YSA, YSA'nın ilk türüdür ve geri beslemeli YSA'dan daha basittir. İleri beslemeli YSA olarak adlandırılır çünkü bir katmandaki her bir nöron yalnızca bir sonraki katmanın (çıktı katmanına doğru) nöronlarına yönlendirilmiş bağlantılara sahiptir (Gurney, 1997) ve bilgiler sadece ağda (döngü yok) ilerlemektedir, yani bilgi akışı giriş katmanından çıkış katmanına doğru yalnızca tek yönde ilerlemektedir. Şekil 3.4'te örnek bir ileri beslemeli YSA verilmiştir. Bu tür ağlara Perseptron ve Adaline örnek olarak verilebilir.



Şekil 3.4. İleri Beslemeli YSA (Sağirlibaş, 2010)

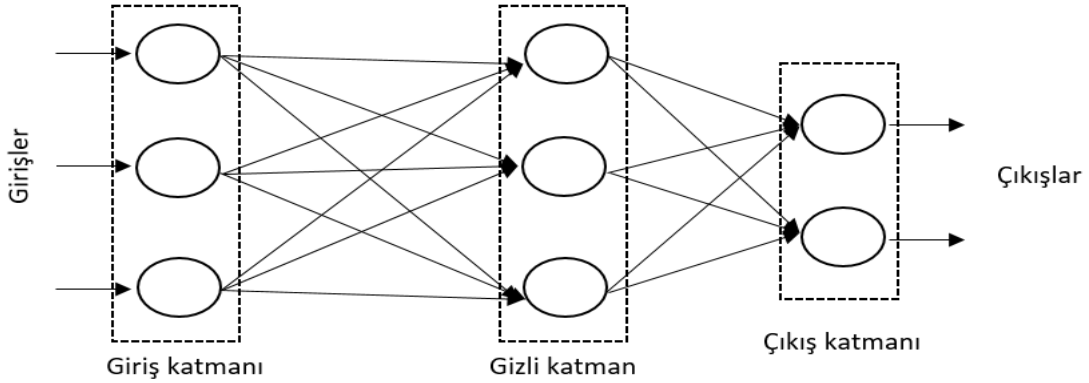
Geri beslemeli YSA'da bir nöronun çıktısı ya doğrudan ya da dolaylı olarak başka bağlı nöronlar vasıtasıyla ya kendisine veya başka nöronlara giriş olarak geri beslenmektedir. Yani nöronların çıktıları diğer nöronlar için geri besleme girdileri olarak kullanılmaktadır. Geri besleme bir katmandaki nöronlar arasında olduğu gibi, katmanlar arasındaki nöronlar arasında da olabilir (Kulluk, 2009). Geribildirimin özelliği, ağları dinamik bilgi işleme için nitelendirmesidir; yani zaman serisi tahmini, sistem tanımlama, optimizasyon ve benzeri gibi zaman değişkenli sistemlerde kullanılabilir. Şekil 3.5'te örnek bir geri beslemeli YSA verilmiştir.



Şekil 3.5. Geri Beslemeli YSA (Gallo, 2015)

2.1.6.2. YSA'nın yapısına (mimarisine) göre sınıflandırılması

YSA yapay sinir hücrelerinin toplanması sonucu oluşmaktadır (Ozturk, 2011). Genel olarak YSA giriş ve çıkış katmanlarından oluşmaktadır ve bu tür YSA'ya tek katmanlı YSA denmektedir. Fakat bu tek katmanlı YSA karmaşık sınıflandırma problemlerinin çözümünde yetersiz kalmaktadır dolayısıyla giriş ve çıkış katmanlarının arasında gizli katman(lar)a ihtiyaç duyulmaktadır ve bu tür YSA'ya da çok katmanlı YSA ismi verilmektedir. Günümüzdeki birçok problemlerde çok katmanlı YSA kullanılmaktadır. Giriş katmanları dış ortamdan bilgi (veri), sinyaller, özellikler veya ölçümlerin alınmasından sorumludur. Sinyalleri (bilgi) alır ve bu sinyalleri herhangi bir işleme tabi tutmadan gizli katmanlara iletir. Gizli katman bu girdiyi alır ve üzerinde işlem gerçekleştirir ve sonra bunu çıkış katmanına iletir. Çıkış katmanı gizli katmandan gelen girdileri işleyip ve girdi katmanından iletilen girdi seti için üretilmesi beklenen çıktıyı üretmeye çalışırlar (Ozturk, 2011). Gizli katmanların sayısı tasarımcıya bağlıdır ve doğru sayıdaki düğümlerin ve katmanların seçilmesi, daha sonra belirli bir problemi yerine getirmek için sinir ağı optimize edilirken önemlidir. Şekil 3.6 (Özdemir, 2010) de YSA'nın üç katmanını göstermektedir.

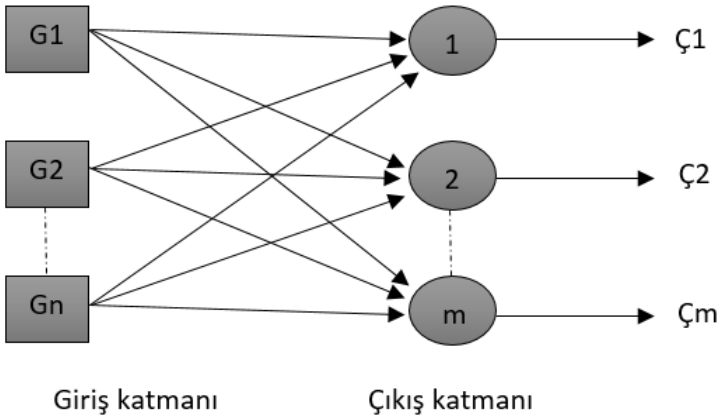


Şekil 3.6. YSA katmanları (Özdemir, 2010)

YSA'nın ana mimarileri, nöron yapıları ve bunların birbirine nasıl bağlı oldukları ve katmanların nasıl yapıldıkları göz önünde bulundurularak, tek katmanlı ileri YSA ve çok katmanlı ileri YSA olarak sınıflandırılabilir.

- **Tek katmanlı ileri YSA**

Bu tür bir ağda sadece bir giriş katmanı ve çıktı katmanı söz konusudur. Şekil 3.7'de n adet giriş ve m adet çıktıdan oluşan basit bir tek katmanlı ileri ağ (feed forward network) gösterilmiştir (da Silva ve ark., 2017). Bilgiler her zaman girdi katmanından çıktı katmanına tek bir yönde (dolayısıyla tek yönlü) aktarılmaktadır. Tek katmanlı mimariye ait ana ağ türleri arasında öğrenme algoritmaları eğitim süreçlerinde kullanılan Perseptron ve Adaline örnek verilebilir.



Şekil 3.7. Tek katmanlı YSA (da Silva ve ark., 2017)

Perceptron 1950'lerde Warren McCulloch ve Walter Pitts'ın eserlerinden esinlenerek bilim adamı Frank Rosenblatt tarafından geliştirilmiştir. YSA'nın en basit türüdür ve doğrusal olarak ayrılabilir desenleri sınıflandırmak için kullanılan YSA'nın temel yapısıdır. Bir algılayıcı birden fazla girdi alır ve bir çıktı üretir ilkesine dayanmaktadır. Basit algılayıcı ağının çıktısı 1 veya 0 cevap veren danışmanlı öğrenme algoritmasını kullanmaktadır. Çıktı değerinin hesaplamasında eşik değer fonksiyonu kullanılır (Öztemel, 2003). Perseptron algoritmasında temel öğrenme, hedef çıktı ile gerçek çıktı arasındaki farkı en aza indirgeyecek ağırlık değerlerinin ayarlanması ile gerçekleşmektedir (Günler, 2012). Perseptrona gelen net girdi $NET = \sum AG$ hesaplanmaktadır. Girdinin eşik değerden büyük veya küçük olmasına göre çıktı ünitesinin değeri 1 veya 0 olur (Öğücü, 2006a) ve Denklem 3.6'daki gibi ifade edilmektedir.

$$y = f(NET) = \begin{cases} 1 & eğer \quad NET > eşik\ değeri \\ 0 & aksi \qquad\quad takdirde \end{cases} \tag{3.6}$$

Eğer gerçekleşen çıktı ile hedef çıktı aynı ise ağırlıklarda hiçbir değişim olmaz. Ancak eğer ağ farklı bir çıktı üretmiş ise o zaman iki durum söz konusu olabilir ve bunlar:

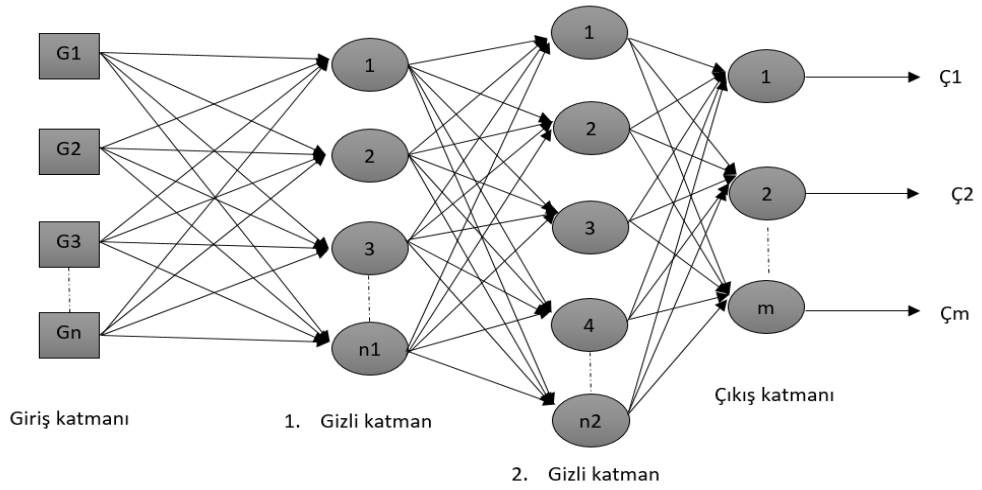
- Hedef çıktı 0 ve gerçekleşen çıktı 1 ise $\rightarrow w_i(k+1) = w_i(k) - \eta . x_i$
- Hedef çıktı 1 ve gerçekleşen çıktı 0 ise $\rightarrow w_i(k+1) = w_i(k) + \eta . x_i$

Burada η öğrenme katsayısıdır ve [0-1] arasında bir değerdir (6.ST).

Adaline yöntemi 1959 yılında Widrow ve Hoff tarafından geliştirilmiştir ve adaptif doğrusal eleman (adaptive linear element) ağının kısaltılmasıdır. Bu algoritma tek yönlü (bir sinir hücresinden oluşan) bir algoritmadır ve danışmanlı öğrenme algoritması kullanmaktadır. Adaline yapısı tüm sinir ağlarının en basitidir, en küçük kareler ortalamaların karesi yöntemine dayanmaktadır (Öztemel, 2003) ve Perseptron algoritmasına çok benzeyen bir algoritmadır. Bu algoritma, eğitim seti üzerindeki doğrusal hataların karelerinin toplamını en aza indirir ve lineer transfer fonksiyonu kullanırlar. Öğrenme kuralı, ağın çıktısının beklenen çıktı değerine bağlı olarak hatasını minimum edecek şekilde ağın ağırlıklarının değiştirilmesi prensibine dayanır. Birçok uygulama için oldukça iyi çalışmasına rağmen doğrusal problem uzayıyla sınırlıdır.

- **Çok Katmanlı İleri YSA (ÇKİYSA)**

Tek katmanlı ileri beslemeli ağlardan farklı olarak, çok katmanlı ileri beslemeli ağlar bir veya daha fazla gizli nöron katmandan oluşmaktadır. Sınıflandırma, tanıma, optimizasyon, robotik vb. gibi doğrusal olmayan çeşitli problemlerin çözümünde kullanılmaktadır. Şekil 3.8'de n adet bir girdi katmanı, sırasıyla n1 ve n2 adet nöronlardan oluşan iki gizli katmanı ve analiz edilen sorunun ilgili çıktı değerlerini temsil eden m adet bir çıktı katmanı gösterilmektedir (da Silva ve ark., 2017). Çok katmanlı ileri beslemeli ağları kullanan ana ağlar arasında, eğitim süreçlerinde kullanılan öğrenme algoritmalarının genelleştirilmiş delta kuralı ve delta kuralı esas alınarak hazırlanan ÇKA ve Radyal Tabanlı Fonksiyon (RBF) bulunmaktadır. Gizli katmanların sayısı ve nöronların miktarı, ağ tarafından eşleştirilen sorunun doğasına ve karmaşıklığına ve aynı zamanda soruna ilişkin mevcut verilerin niceliğine ve kalitesine bağlıdır.



Şekil 3.8. Çok katmanlı YSA (da Silva ve ark., 2017)

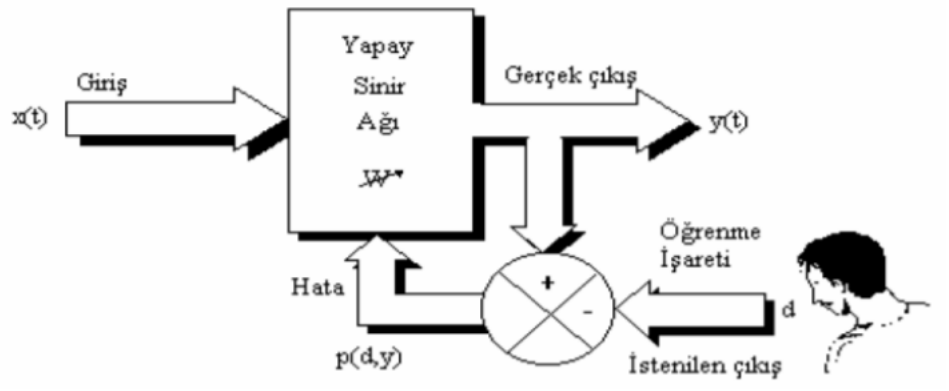
3.1.6.3. YSA'nın öğrenme algoritmalarına göre sınıflandırılması

Araştırmacılar tarafından literatürlerde bu güne kadar pek çok öğrenme algoritmaları geliştirilmiştir. Her öğrenme algoritmasının kendi özelliklerine göre avantaj ve dezavantajları bulunmaktadır ve bu öğrenme algoritmaları YSA'ların sınıflandırma başarısını doğrudan etkilemektedir, bu nedenle YSA için iyi bir öğrenme algoritması seçilmelidir. Genelde gradyan tabanlı olan Geri Yayılım (GY) ve LM öğrenme algoritmaları kullanılmaktadır. YSA'da öğrenme, belirli bir ağın nöronları arasındaki bağlantılarının ağırlıklarını değiştirme metodudur (Gallo, 2015) YSA'da

öğrenme, danışmanlı öğrenme, danışmansız öğrenme ve takviyeli öğrenme olmak üzere üç kategoriye ayrılmaktadır.

3.1.6.3.1. Danışmanlı öğrenme

Bu tür öğrenme bir danışmanın gözetiminde gerçekleşmektedir (Gallo, 2015). Girdi değişkenleri (x) ve çıktı değişkeni (Y) verilmektedir. Hedef yeni veri girildiği zaman (x), o veri için çıktı değişkeni (Y) tahmin edebilmektir. Başlangıçta sistem rastgele ağırlıklarla başlamaktadır, çıktı hakkında ilk tahmin yapılmaktadır, ağın ürettiği çıktılar ile hedef çıktılar arasındaki fark hata olarak ele alınır ve bu hata en aza indirgemeye çalışılır (Kulluk, 2009; Yalçın, 2012). Hata oranını en aza indirgemek için ağırlıklar değiştirilmektedir. Bu tekrarlayan tahmin süreci, veriler çıktı katmanına ulaşana kadar devam etmektedir (Günler, 2012). Öğrenme, algoritma kabul edilebilir düzeyde bir performans elde ettiğinde sonlanmaktadır. Donald Hebb, 1949 yılında ilk danışmanlı öğrenme stratejisini önermiştir. Çok katmanlı algılayıcı ağı danışmanlı öğrenme algoritmalarına örnek olarak verilebilir (Kulluk, 2009). Şekil 3.9'da danışmalı öğrenmenin çalışma mantığı gösterilmektedir.

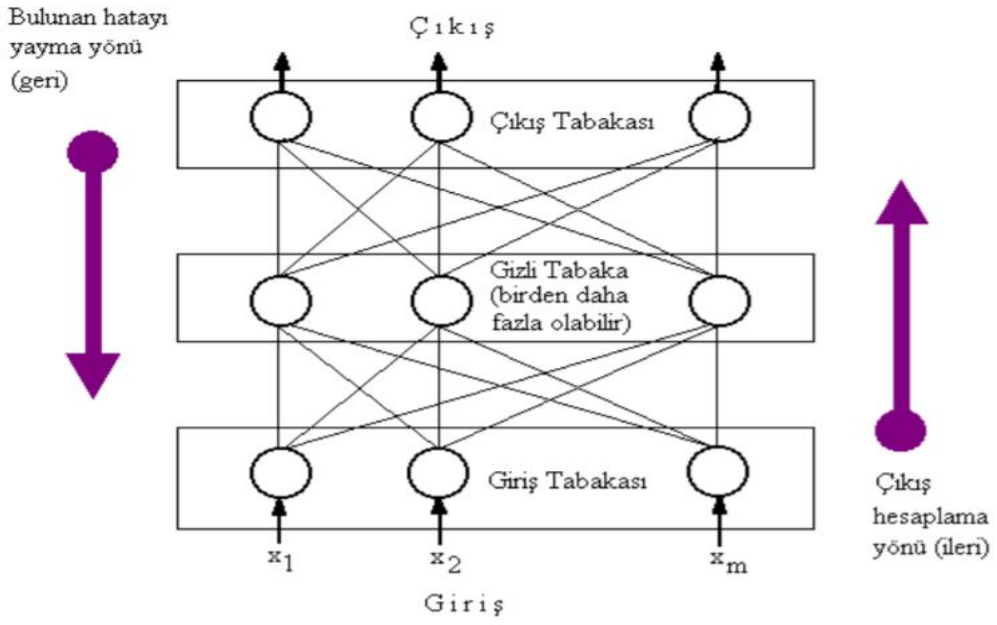


Şekil 3.9. Danışmalı öğrenme çalışma mantığı (Sağıroğlu ve ark., 2003)

- **GY algoritması**

Geri yayılım (GY) algoritması gradyan tabanlı bir öğrenme algoritmasıdır. Beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak için YSA'nın elemanlarının bağlantılarına yönelik ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. ÇKA modeli XOR problemini çözmek amacı ile yapılan araştırmalar neticesinde ortaya çıkmıştır (Kulluk, 2009). Rumelhart ve ark. tarafından önerilen geri yayılım modelinin temel amacı YSA'nın gerçek çıktısı ile hedef çıktı arasındaki hatayı en aza indirgemektir. ÇKA sayesinde bu güne kadar mühendislik

problemlerin çoğuna çözümler bulunmuştur. Geri yayılım modeli giriş katmanı, en az bir gizli katman ve çıkış katmanı içeren çok katmanlı bir sinir ağıdır. Bir katmandaki bütün elemanlar bir önceki elemanlarla bağlıdır. ÇKA ağlarda bilgi akışı ileri doğrudur. Şekil 3.10'da çok katmanlı algılayıcı yapısı gösterilmiştir. Birçok öğrenme algoritmasının çok katmanlı algılayıcı ağı eğitmede kullanılması, bu modelin yaygın kullanılmasının sebebidir ve GY algoritması bu öğrenme algoritmalardan biridir. GY algoritması delta kuralına dayanan danışmanlı bir öğrenme algoritmasıdır (Günler, 2012). Ana ilkesi, girdi verisine karşılık gelen özel fonksiyon karakteristikleri ile elde edilen çıktıları sağlamak için ağda bulunan tüm ağırlıkları güncellemektir (Günler, 2012). Bu algoritmada elde edilen hata çıkıştan geriye doğru yönlendirilmektedir. Adından da anlaşılacağı üzere, bu ağda geriye yayılma gerçekleşmektedir (Gallo, 2015). Ağın ağırlıkları ayarlanarak hata azaltılmaya çalışılmaktadır. Hedef çıktı ile gerçek çıktı arasındaki hata, giriş katmanına doğru geri yayılmaktadır (Gallo, 2015). Bu işlem minimum hataya ulaşılıncaya kadar devam etmektedir.



Şekil 3.10. Çok katmanlı algılayıcı yapısı (Sağıroğlu ve ark., 2003)

Girdi katmana verilen bilgiler herhangi bir değişiklik yapılmadan gizli (ara) katmana iletilmektedir ve giriş katmanın çıkışı Denklem 3.7 gibi hesaplanmaktadır.

$$Ç = G \tag{3.7}$$

Ara katmandaki tüm birimler girdi katmandaki tüm birimlerden gönderilen bilgileri bağlantı ağırlıklarını da kullanarak almaktadır (Öztemel, 2003). Ara katmanın net girdisi Denklem 3.8 gibi hesaplanmaktadır ve ara katmanın çıktısı ise net girdinin

genellikle sigmoid aktivasyon fonksiyonundan geçirilerek hesaplanmaktadır ve ara katmanın çıktısı Denklem 3.9 ile hesaplamaktadır.

$$net_j = \sum_i x_i w_{ij} \tag{3.8}$$

$$x_j = \frac{1}{1 + e^{net_j}} \tag{3.9}$$

Burada w_{ij} birim i'den j'ye bir ağırlık değeri olarak tanımlanmaktadır. Ağın çıkışı ile hedef çıkış arasındaki farkın karesinin toplamı alınarak hata hesaplanmaktadır. Ayrıca negatif ve pozitif hataların birbirini etkilememesi için bu farklar toplanmadan önce kareleri alınmaktadır. Burada amaç hatanın en aza indirilmesidir. Bu işlem Denklem 3.10 ve Denklem 3.11 gibi hesaplanmaktadır.

$$E_k = (d_k - o_k) \tag{3.10}$$

$$E_{(t)} = \frac{1}{2}\sum_{k=1}^{k} E_k^2 \tag{3.11}$$

Burada d_k beklenen çıktı ve o_k k. elemanın hedef çıktısını ifade etmektedir. Üretilen hatayı en aza indirgemek için ağırlıklar geriye doğru hesaplanarak değiştirilmektedir. Hatalar birimin sorumlu olduğu bu hatanın miktarına göre her birime dağıtarak geriye doğru yayılmaktadır. Ağın ağırlıklarını değiştirmek için iki durum söz konusudur. Bunlar gizli katman ile çıkış katmanı arasındaki ağırlıkların değiştirilmesi ve giriş katmanı ile gizli katman arasındaki ağırlıkların değiştirilmesidir. λ terimi öğrenme katsayısını ifade etmektedir (öğrenme katsayısı ağırlıkların değişim miktarıdır) ve α momentum katsayısını ifade etmektedir. Gizli katmandaki j birimi çıktı katmandaki i birimine bağlayan ağırlıkların değişimi Denklem (3.12) ile hesaplanmaktadır. Ağırlıkların t iterasyonundaki yeni değerleri Denklem (3.13) ile hesaplanmaktadır. Aynı şekilde eşik değer birimin de ağırlıklarının değiştirilmesi gerekmektedir ve bu değişiklik Denklem (3.14) ile hesaplanmaktadır. Ağırlıkların eşik değer birimindeki değişimi Denklem (3.15) ile hesaplanmaktadır. Benzer şekilde gizli katmanlar veya gizli-giriş katmanları arasındaki ağırlıkların değişimi ve eşik değer birimin ağırlıklarının değişimi Denklem (3.16), (3.17), (3.18), (3.19) ile hesaplanmaktadır.

$$\Delta D_{jk}^{\alpha}(t) = \lambda \delta_k X_j^{\alpha} + \alpha \Delta D_{jk}^{\alpha}(t-1) \tag{3.12}$$

$$D_{jk}^{\alpha}(t) = D_{jk}^{\alpha}(t-1) + \Delta D_{jk}^{\alpha}(t) \quad (3.13)$$

$$\Delta\beta_{k}^{\varsigma}(t) = \lambda\delta_{k} + \alpha\Delta\beta_{k}^{\varsigma}(t-1) \quad (3.14)$$

$$\beta_{k}^{\varsigma}(t) = \beta_{k}^{\varsigma}(t-1) + \Delta\beta_{k}^{\varsigma} \quad (3.15)$$

$$\Delta D_{lj}^{i}(t) = \lambda\delta_{j}^{\alpha}X_{l}^{i} + \alpha\Delta D_{yj}^{i}(t-1) \quad (3.16)$$

$$D_{lj}^{i}(t) = D_{lj}^{i}(t-1) + \Delta D_{lj}^{i}(t) \quad (3.17)$$

$$\Delta\beta_{j}^{\alpha}(t) = \lambda\delta_{j}^{\alpha} + \alpha\Delta\beta_{j}^{\alpha}(t-1) \quad (3.18)$$

$$\beta_{j}^{\alpha}(t) = \beta_{j}^{\alpha}(t-1) + \Delta\beta_{j}^{\alpha}(t) \quad (3.19)$$

- **LM algoritması**

LM algoritması, temel olarak maksimum komşuluk fikri üzerine kurulmuş bir algoritmadır, doğrusal olmayan fonksiyonların karelerinin toplamı olarak ifade edilen bir fonksiyonun yerel bir minimumunu bulan tekrar eden bir tekniktir. Doğrusal olmayan en küçük kareler problemleri için standart bir teknik haline gelmiştir ve en dik inişin ve Gauss-Newton yönteminin bir birleşimi olarak düşünülebilir. Geçerli çözüm doğru olanın aksine, algoritma en dik bir iniş metodu gibi davranmaktadır. Yavaş ancak yakınsama garantisi söz konusudur. Geçerli çözüm doğru çözüme yakın olduğunda, bir Gauss-Newton yöntemi haline gelmektedir (Lourakis, 2005) İleri beslemeli ağlarda en hızlı öğrenme metodudur ve hızlı bir şekilde çözümlere ulaşmasına karşın çok fazla bellek gerektirmektedir. Doğrusal olmayan fonksiyonların karelerinin bir toplamı olan F(x) fonksiyonunun minimumunu bulma Denklem 3.20'de ve Ağırlıklar Denklem 3.21'deki gibi hesaplanmaktadır.

$$F(x) = E(x) = \frac{1}{2}\sum_{i=1}^{m}[f_i(x)]^2 \quad (3.20)$$

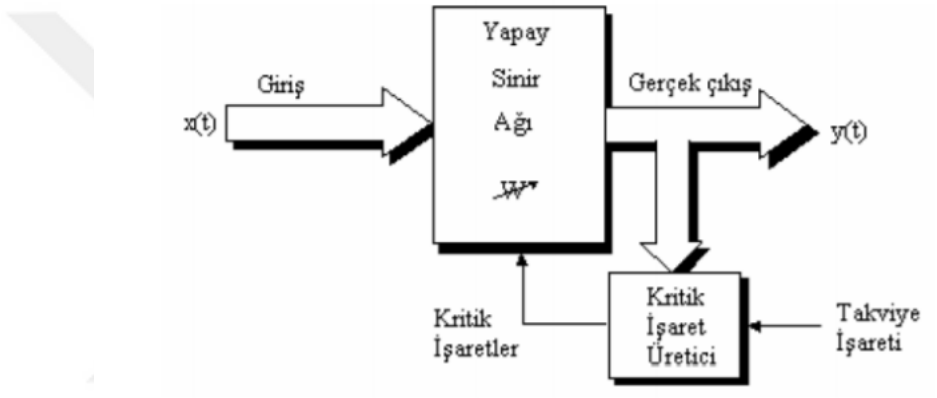
$$w(t+1) = w(t) - [J^T(t)J(t) + \mu I]^{-1}J^T(t)E(t) \quad (3.21)$$

Ağırlıkların hesabında kullanılan μ Marquardt parametresi, I birim matrisi, J Jakobian Matrisi ve E uygunluk değerini göstermektedir (Kılıç, 2015). μ ilk olarak küçük bir değer olarak seçilmektedir ve uygun değer bulunana kadar yeni ağırlık değeri eskisinden küçük ise 10 kat arttırılır büyük ise 10 kat azaltılır (Kılıç, 2015). Eğer bu

parametre çok büyükse yöntem dik iniş metodu gibi davranır ancak çok küçükse Newton metodu gibi davranmaktadır (Lourakis, 2005). Levenberg algoritması GY algoritmasından daha hızlıdır (Kılıç, 2015).

3.1.6.3.2 Takviyeli öğrenme

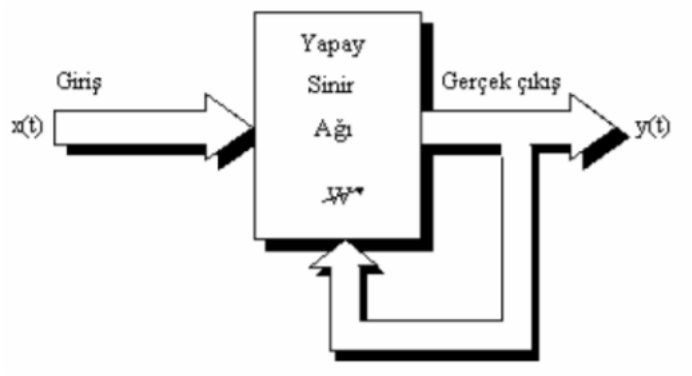
Bu öğrenmede öğrenen sisteme bir danışman yardımcı olur, her girdi seti için istenen çıktı setini sisteme vermek yerine, sistemin kendisine gösterilen girdilere karşılık kendi çıktısını üretmesini bekler ve elde ettiği çıktının iyi veya kötü olup olmadığına dair bir bilgi verir (Yalçın, 2012). LVQ ağı takviyeli öğrenme algoritmalarına örnek olarak verilebilir (Öztemel, 2003). Şekil 3.11'de takviyeli öğrenmenin çalışma mantığı gösterilmiştir.



Şekil 3.11. Takviyeli Öğrenme (Kulluk, 2009)

3.1.6.3.3. Danışmansız öğrenme

Bu tür öğrenme herhangi bir danışman gözetimi söz konusu değildir. Sisteme sadece girdi değerleri verilir ancak hedef çıktıları verilmemektedir (Kriesel, 2007). Ağa sunulan örnekten çıkış bilgisini ağ kendi başına geliştirmeye çalışmaktadır (Kulluk, 2009). Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Bu tip öğrenme işlemine danışmansız öğrenme adı verilmektedir çünkü burada çıkış değerleri kullanılmamaktadır. Şekil 3.12'de danışmansız öğrenmenin çalışma mantığı gösterilmiştir. ART ağları örnek olarak verilebilir (Öztemel, 2003).



Şekil 3.12. Danışmansız Öğrenme (Sağıroğlu ve ark., 2003)

3.1.7. Temel öğrenme kuralları

Literatürde kullanılan çok fazla sayıda öğrenme kuralı bulunmaktadır. YSA'da öğrenme kurallarının çoğu Hebb öğrenme kuralına dayanmaktadır (Öztemel, 2003). Temel olarak dört farklı öğrenme kuralı kullanılmaktadır bunlar (Hebb, Hopfield, Delta ve Kohonen) öğrenme kurallarıdır.

3.1.7.1. Hebb öğrenme kuralı

1949'da Donald Hebb tarafından önerilen bir öğrenme kuralıdır. En eski ve basit öğrenme kurallarından biridir ve diğer öğrenme kurallarının temelini oluşturmaktadır. Büyük ölçüde biyolojik sistemlerin dinamiklerini temel almaktadır. Bu öğrenme kuralına göre YSA'da bir nöron eğer başka bir nörondan sinyal (bilgi) alıyorsa ve her iki nöronda aktif ise her iki nöron arasındaki bağlantı artırılmaktadır. Aslında bu öğrenme kuralı şu şekilde özetlenebilir eğer bir nöron kendisi aktif ise bağlı olduğu diğer nöronu aktif yapmaya, pasif ise pasif yapmaya çalışmaktadır (Öztemel, 2003). Çoğu öğrenme kuralları bu felsefeyi temel alarak geliştirilmiştir (Öztemel, 2003).

3.1.7.2. Hopfield öğrenme kuralı

Hopfield öğrenme kuralı John Hopfield tarafından 1982'de icat edilen bir öğrenme kuralıdır. Hopfield kuralı Hebb kuralına benzeyen bir algoritmadır. Bu kuralda YSA nöronlarının bağlantılarının ne kadar artırılması (kuvvetlendirilmesi) veya zayıflatılması gerektiği belirlenmektedir. Eğer beklenen çıktı ve girişlerin her ikisi de aktif veya her ikisi de pasif ise öğrenme katsayısı kadar ağırlık değerleri artırılmaktadır veya zayıflatılmaktadır (Öztemel, 2003). Ağırlıkların artırılması veya zayıflatılması

öğrenme katsayısı ile gerçekleştirilmektedir. Genellikte öğrenme katsayısı 0-1 arasında sabit veya pozitif bir değerdir.

3.1.7.3. Delta öğrenme kuralı

Widrow ve Hoff tarafından geliştirilen ve en az ortalama kare yöntemi ve geri yayılım olarak da adlandırılan delta kuralı, en çok kullanılan öğrenme kurallarından biridir. Verilen bir girdi vektörü için, çıktı vektörü doğru çıkış ile karşılaştırılır. Beklenen çıktı ile gerçekleşen çıktı arasındaki farkın en aza indirgemek için YSA'daki nöronların bağlantılarının ağırlık değerlerinin devamlı olarak değiştirilmesi prensibine dayanarak bu kural geliştirilmiştir (Öztemel, 2003). Ağın gerçek çıktısı ile hedef çıktısı arasındaki hatanın karelerinin ortalamasını en aza indirgemek amaçlanmaktadır.

3.1.7.4. Kohonen öğrenme kuralı

Biyolojik sistemlerdeki öğrenmeye dayalı olarak (Kohonen, 1990) tarafından geliştirilmiştir. Bu kuralda hücreler yarış içerisindedir ve yarışı kazanan hücrenin ağırlıkları güncellenmektedir. En büyük çıkışa sahip olan nöron kazanmaktadır. Bunun anlamı kazanan nöronun yakın çevresinde olan nöronlara karşı daha güçlü olduğudur. Kohonen kuralı danışmansız öğrenme kuralıdır çünkü hedef çıkışa ihtiyaç duyulmamaktadır.

3.2. Optimizasyon

Optimizasyon, yapılacak olan işlerde belirli şartları gözeterek en iyinin aranması işlemleri olarak tanımlanmaktadır. Başka bir ifadeyle optimizasyon, hedeflenen amaca ulaştıracak paydaşların aranması olarak da tanımlanabilir.

Optimizasyon, bir fonksiyonun maksimum veya minimumunu veren koşulların bulunması işlemidir. Verilen koşullar altında en iyi sonucu elde etme eylemidir (Ozturk, 2011). Başka bir deyişle, optimizasyon belirli bir alan içindeki bir fonksiyon için en uygun değeri bulmaktır (Yurdakul, 2014). Optimizasyon, alternatifler arasında en iyi çözümü bulmak için kullanılan bir süreçtir. Farklı türde problemler için genellikle farklı türde optimizasyon algoritmaları kullanılmaktadır. Bir insan yeni bir görüş ortaya koyar, optimizasyon bu görüşün geliştirilmesine yardımcı olur.

Optimizasyon probleminin temel öğelerinden olan $f(x)$ maksimize veya minimize edilmek istenen amaç fonksiyonu, maliyet fonksiyonu veya uygunluk fonksiyonu olarak tanımlanabilir. Amaç fonksiyonunun değişkenlerine belirli

aralıklarda olmak şartıyla karar değişkeni denmektedir. $f(x) = 0$ ve $g_k(x) \leq 0$, $(k = 1,2,..,K)$ fonksiyonun sağlanmasında gereken sınırlamaları sağlayan fonksiyonlara da sınırlamalı fonksiyon adı verilir. Buradaki amaç en basit anlamda bir minimum veya maksimizasyon problemi olarak düşünülebilir ve verilen soruna göre amaç fonksiyonun minimum veya maksimum yapılmasıdır (Rao, 2009; Canayaz, 2015).

Çoğu optimizasyon yöntemleri, kökü veya sıfırı bulma işlemini kullanmaktadır. Matematiksel yaklaşımda, kök bulmak için fonksiyonun sıfır olduğu yerler araştırılırken, optimizasyon da türevin sıfır olduğu yerleri araştırmak gerekmektedir.

Optimizasyonda diğer bir sorun; bulunan bir sonucun, global veya lokal bir çözüm olup olmadığının tespit edilmesidir. Örneğin; elde edilen bir kökün optimal bir çözüm olduğunu anlamak zordur. Çünkü tüm kökler, fonksiyonu sıfır yapmaktadır.

Sürekli ve ayrık parametreli optimizasyonda, sürekli parametreler sonsuz değer alırken, ayrık parametreler sınırlı değerler almaktadır. Ayrık parametreli optimizasyon kombinasyonel olarak da adlandırılabilir. Bir $f(x)$ fonksiyonunun minimum değerini bulmaya çalışmak, sürekli parametreli optimizasyon olarak ifade edilmektedir.

Optimizasyon algoritmaları deterministik veya stokastik algoritmalar olarak sınıflandırılabilir. Deterministik algoritmalarda parametreler sabit değerler alır ve belirli bir girdi seti probleme her sunulduğunda, algoritma aynı hesaplamaları yapar ve girdi setinin sunulduğu diğer zamanlarda aynı sonuçları verir. Böyle bir algoritma için, aynı başlangıç noktasına başlarsak sistem aynı son çözüme ulaşır. Tepe Tırmanma algoritması deterministik algoritmalarının bir örneğidir. Öte yandan, parametreler rastgele değerler aldığında ve algoritmada bazı rasgelelik varsa, algoritma aynı başlangıç noktası için kullanılsa bile, algoritma çalıştırıldığında her zaman farklı bir noktaya yani her zaman farklı sonuca ulaşacaktır. GA, YA ve PSO, stokastik algoritmalarının birer örneğidir.

Deterministik yöntemler iki gruba ayrılabilir: gradyan tabanlı ve gradyan tabanlı olmayan algoritmalar. Örneğin, Newton-Raphson yöntemi, optimum noktayı bulmak için fonksiyon değerlerini ve türevlerini kullanan bir gradyan tabanlı algoritmadır. Gradyan tabanlı olmayan yöntemler, türev bilgileri kullanmaz, yalnızca fonksiyon değerlerini kullanır ve fonksiyonda süreksizlik olduğunda tercih edilmektedir. Nelder-Mead downhill, gradyan tabanlı olmayan algoritmalara örnektir.

Stokastik algoritmalar ’da sezgisel algoritmalar ve meta-sezgisel algoritmalar olmak üzere iki gruba ayrılmaktadır. Sezgisel algoritmalar, son zamanlarda birçok mühendislik problemlerinin çözümünde kullanılmaktadır. Bunun sebepleri arasında; hesaplama gücünün iyi olması, uygulamasının kolay olması ve dönüştürülebilir olması yani herhangi problem için yazılmış bir sezgisel programın basit bir şekilde başka problemlere uygulanabilmesidir (Alataş, 2007; Özkış, 2013). Sezgisel algoritmalar problem çözme tekniğidir, yakınsama özelliğine sahiptir ve en iyi çözümü bulacaklarını garanti etmezler, ancak bunun yerine en iyi çözüme yakın bir çözüm bulacaklarını garanti ederler. Genellikle en iyiye yakın olan çözüm yoluna hızlı ve kolay bir şekilde ulaşırlar. Genel olarak sezgisel yöntemler aşağıdaki gibi sınıflandırılabilir:

- Biyoloji tabanlı
- Fizik tabanlı
- Sürü tabanlı
- Sosyal tabanlı
- Müzik tabanlı
- Kimya tabanlı

Eğer Algoritma bir çözümden başlarsa ve bunu operatörlerle devam ettiriyorsa bunlara tek noktalı yöntemler denmektedir. Tabu arama gibi bütün yerel arama tabanlı algoritmalar tek noktalı yöntemlere örnek olarak verilebilir. Algoritma bir popülasyon üzerinden çözüme başlayıp bu değişik noktalarla optimizasyon yapılıyorsa bunlar çok noktalı ya da popülasyon tabanlı yöntemlerdir, PSO, KKO algoritması ve YA popülasyon tabanlı yöntemlere örnek olarak verilebilir (Alataş, 2007).

Meta-sezgisel algoritmalar ise optimal sonuca yakın çözümleri verimli bir şekilde bulabilme ve arama alanında araştırma ve keşfetme için farklı kavramları zeki bir şekilde birleştirerek, bir alt seviye sezgisellere rehberlik eden iteratif bulma süreci kullanmaktadır. Meta-sezgisel terimi Yunan kökenli iki kelimenin birleşiminden oluşmaktadır ve ilk önce Glover (Glover, 1986) tarafından kullanılmıştır (Özkış, 2013). Bu algoritmaların en önemli özellikleri yaptığı yerel aramalarda daha önce bulduğu değerleri hafızasında tutabilme kabiliyetine sahip olmalarıdır (Canayaz, 2015). Meta-sezgisel algoritmalar sezgisel algoritmalardan çoğu zaman daha iyi performans göstermektedir.

Meta-sezgisel algoritmaların genel özellikleri aşağıdaki gibidir (Blum ve Roli, 2003; Özkış, 2013):

- Arama sürecini "yönlendiren" stratejileri vardır.

- Hedef, en uygun çözümleri bulmak için arama alanını verimli bir şekilde araştırmaktır.
- Meta-sezgisel algoritmaları oluşturan teknikler basit yerel arama algoritmalarından karmaşık öğrenme süreçlerine kadar uzanan bir yelpazeyi içermektedir.
- Meta-sezgisel algoritmalar yaklaşıktır ve genellikle deterministik değildir.
- Arama alanının tuzaklardan kurtulmak için mekanizmalar içerebilirler.
- Meta-sezgisel probleme özgü değildir.
- Arama esnasında elde edilen bilgiyi (hafızayı) kullanırlar.
- Farklı metotlar kullanarak arama uzayını keşfeden yüksek seviyeli algoritmalardır.

Meta-sezgisel algoritmaların iki stratejileri vardır bunlar: yoğunlaştırma (intensification) ve çeşitlendirmedir (diversification). Çeşitlendirme, arama alanını global ölçekte keşfetmek için çeşitli çözümler üretmek anlamına gelmektedir (exploration); yoğunlaşma ise, mevcut çözümde arama tecrübelerinden faydalanılması (exploitation) anlamına gelmektedir (Blum ve Roli, 2003; Özkış, 2013). Algoritma yakınsama oranını artırmak için en iyi çözümleri seçerken, yoğunlaşma ve çeşitlendirme arasında iyi bir denge bulunmalıdır. En iyi seçim, çözümlerin optimuma yakınlaşmasını sağlarken, rasgele arama yoluyla çeşitlendirme, aramayı yerel optimalden ve aynı zamanda da çözüm çeşitliliğini arttırmaya olanak tanımaktadır. Bu iki ana bileşenin iyi bir kombinasyonu genellikle küresel optimalliğe ulaşılmasını sağlamaktadır.

Meta-sezgisel algoritmalar, yörünge tabanlı ve popülasyon tabanlı olarak ta sınıflandırılabilir. Bir yörünge tabanlı algoritma, genellikle, iterasyon devam ederken bir yolu izleyecek tek bir aracı veya bir seferde bir çözümü kullanmaktadır. Tepe Tırmanma algoritması, yörünge tabanlıdır ve başlangıç noktasını, son noktayla zikzak yolu aracılığıyla birleştirir. Bir diğer önemli örnek, yaygın olarak kullanılan bir meta-sezgisel algoritması olan Benzetimli Tavlamadır. Öte yandan, PSO gibi popülasyon tabanlı algoritmalar, çoklu yolları izleyecek birden fazla aracı kullanmaktadır dolayısıyla birden fazla çözüm kullanmaktadır.

Meta-sezgisel algoritmalar genellikle doğadan ilham alınan ve günümüzde optimizasyon için en yaygın kullanılan algoritmalardır. Doğa uzun zamandır karşılaştığı sorunlara üstün çözümler bulmayı başarmıştır, bundan dolayı problem çözme yeteneğini doğayı taklit ederek öğrenebilir ve doğa esinli sezgisel ya da meta-sezgisel algoritmalar

geliştirilebilir. Meta-sezgisel algoritmalar, GA, Benzetimli Tavlama, DE, KKO, YAKA, YA, PSO, HA ve AB algoritması dâhil olmak üzere çok çeşitlidir.

Öncelikle bilgisayar bilimleri olmak üzere, mekanik, inşaat, elektronik gibi ilgili alanlardaki optimizasyon problemlerinin çözümünde son zamanlarda bu algoritmaların kullanımını görmek mümkündür (Canayaz, 2015). Optimizasyon problemlerinin çözümünde seçilecek olan optimizasyon tekniği problemin türüne göre değişiklik göstermektedir (Canayaz, 2015). Yani doğrusal olmayan herhangi bir problem için global arama yapan bir meta-sezgisel algoritmanın tercih edilmesi gerekmektedir (Canayaz, 2015).

3.3. Yarasa Algoritması (YA)

3.3.1. Yarasaların ekolokasyon kabiliyeti

Meta-sezgisel algoritmalar, doğadan ilham alınarak veya doğayı taklit ederek önerilen ve birçok optimizasyon problemlerine esnek çözümler üretebilen algoritmalardır. YA, yarasaların ekolokasyon adı verilen sesin yankılanmasından faydalanılarak bir cismin bulunduğu yön ve uzaklık saptama davranışından esinlenen bir popülasyon tabanlı optimizasyon algoritmasıdır. YA 2010 yılında Yang (Yang, 2010a) tarafından sürü zekasına dayanan ve meta-sezgisel bir optimizasyon algoritması olarak önerilmiştir.

Yaklaşık 1000 farklı yarasa türü vardır bunlardan üçü vampirdir ve Latin Amerika'da yaşarlar, yeryüzünde 18 familyaya bağlı 986 tür yarasa vardır ve Türkiye'de 4 familyaya bağlı 30 tür bulunmaktadır, Türkiye'de yaşayan yarasa familyaları şunlardır (Uçan köpekler, Nal burunlu yarasalar, Düz burunlu yarasalar ve Kuyruklu yarasalar). Çoğu yarasa türü belirli bir dereceye kadar ekolokasyon kullanır; Tüm türler arasında mikro yarasalar ekolokasyonu yoğun bir şekilde kullanırlar (Yang, 2010b).

Çoğu yarasa türü, iletişim kurmak, farklı böcek türlerini tanımak, avlarının konumlarını belirlemek, avlarına olan mesafeyi hissetmek, karanlıkta bile gündüz gibi etraflarındaki her türlü cismi algılayıp herhangi bir engele çarpmadan hareket etmek ve bu karanlıkta hareket eden farklı türdeki böcekleri ayırt etmek için ekolokasyon adı verilen bir tür sonar kullanırlar (Fenton, 2004; Yılmaz, 2014).

Ekolokasyon kullanan yarasalar da dâhil olmak üzere tüm canlılar belli bir frekansta bazı sinyaller yayarlar (Yılmaz, 2014). Bu sinyaller, çoğunlukla insan kulağının algılayabileceği ses sınırlarının ötesindedir (Koç, 2016).

Yarasaların gözleri görür ama buna rağmen, ekolokasyon sistemi ile yönlerini tayin eder, avlanır ve hareket ederler. Yarasaların ekolokasyon sistemi hem yönünü bulması ve aynı zamanda avını tespit edebilmesi bakımından son derece önemlidir (Demir, 2015).

Yarasa yaydıkları sinyallerin etrafındaki nesnelere veya avlara çarpmasından sonra yankılar oluşturur. Yarasa yankıyı dinler ve daha sonra bu yankılardaki kodları analiz ederek etrafındaki nesneleri belirler, kategorize eder ve yerini tespit eder (Schnitzler ve Kalko, 2001; Yılmaz, 2014).

3.3.2. YA yapısı

YA, Xin-She Yang tarafından 2010 yılında, yarasa ekolokasyonunun bir önceki bölümde belirtilen özelliklerine dayanarak, aşağıdaki üç kuralla geliştirmiştir (Yang, 2010b):

1. Tüm yarasalar mesafeyi algılamak için, yiyecek / av ve engelleri ayırt etmek için ekolokasyon kullanır.
2. Yarasalar yiyeceklerini veya avlarını bulmak için v_i hızında, x_i konumunda, sabit frekansta f_{min}, dalga boyu λ ve ses şiddeti A_0 ile rasgele uçarlar. Yaydıkları sinyallerin frekansını otomatik olarak ayarlayabilir ve hedeflerinin yakınlığına bağlı olarak sinyal yayma oranını $r \in [0, 1]$ ayarlayabilirler.
3. Ses şiddeti değişiklik göstermektedir, ses şiddetinin büyük değer A_0'dan minimum değer A_{min} 'e değişir. Orijinal YA adımları Çizelge 3.1'de verilmiştir.

Çizelge 3.1. YA'nın Adımları

Her bir yarasaya ait başlangıç frekansı f_i *belirle*
Ses şiddeti A_i *ve sinyal yayılım oranı* r_i *değerlerini belirle*
While$(t <$ *maksimum iterasyon sayısı*$)$
    Frekansı ayarlayarak konumları, hızları güncelle
    ve yeni uygunluk değeri hesapla
  if $(rand\ (0,1) > r_i)$
    En iyi çözümler arasında bir çözüm seç
    Seçilen en iyi çözümü etrafında yerel bir çözüm üret
  end if
    Rasgele uçarak yeni bir çözüm üret
  if $\left(rand < A_i\ \&f(\mathrm{x_i}) < f(\mathrm{x_*})\right)$
    Yeni çözümleri kabul et
    r_i *arttır ve* A_i *azalt*
  end if
  Yarasaları sırala ve en iyi olanını bul
end while
En iyi olanın uygunluk değerini göster

3.3.3. Yarasaların hareketi

Her yarasa, arama uzayında belli bir frekans f_i ve popülasyondaki tüm yarasalar arasında mevcut en iyi bireyin sahip olduğu çözüm değerine x_* bağlı olarak üretilen hız değeri v_i^t ile bir sonraki adımda bulunacakları pozisyonlarını belirlemektedirler (Yılmaz, 2014). Bu nedenle, bir önceki bölümde belirtilen üç kurala uygun olarak konum x_i^t ve hız v_i^t için güncellenme işlemi Denklem 3.22, Denklem 3.23 ve Denklem 3.24 kullanılarak gerçekleştirilir (Yang ve He, 2013).

$$f_i = f_{min} + (f_{\max} - f_{\min})\beta \tag{3.22}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{3.23}$$

$$x_i^t = x_i^{t-1} + v_i^t, \tag{3.24}$$

Burada β ϵ [0, 1] aralığında rasgele bir değer, x_* tüm yarasalar arasında şu anki en iyi konumu göstermektedir, yani n adetten oluşan yarasalar için tüm yarasaların sonuçlarının en iyi sonucunu göstermektedir. Algoritmada başka en iyiler çıktıkça bu değer güncellenmektedir. f_i ise her problem için değişebilir çünkü f_i'nin değerini doğrudan etkileyen $[f_{min}, f_{max}]$ değerleri vardır, bu değerler ise problemdeki boyutların arama uzayını temsil etmektedir. Her yarasa popülasyon içerisinden

uygunluk değerinin kalitesine bağlı olarak bir başka çözüm seçmekte ve seçtiği çözümün etrafında yeni kaynağa yönelmektedir. Konum güncelleme için Denklem 3.25 kullanılmaktadır.

$$x_{new} = x_{old} + \epsilon A^t \tag{3.25}$$

Burada ε değeri [−1, 1] aralığında rastgele bir sayıdır, A^t ise, t anında tüm yarasaların ortalama ses şiddetini ifade etmektedir.

3.3.4. Ses şiddeti ve sinyal yayma oranı

İterasyon ilerledikçe ve istenilen hedefe yaklaşma oranında ses şiddeti A_i ve sinyal yayma oranı r_i güncellenmelidir. Ses şiddeti genellikle yarasa avını bulduğunda azalır, buna karşılık sinyal yayma oranı artar (Yang, 2010b). Ses şiddeti A_{min} ve A_{max} arasında herhangi bir değer olarak seçilebilir, $A_{min} = 0$ varsayılarak bir yarasa avını bulduğu ve geçici olarak herhangi bir sesi yaymayı bıraktığı anlamına gelmektedir. Bu varsayımlarla ses şiddeti güncelleme işlemi Denklem 3.26 ve Denklem 3.27'deki gibi gerçekleşmektedir (Yang, 2010b; 2011).

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \tag{3.26}$$

Burada α ve γ [0,1] aralığında bir değer. α parametresi 0 ile 1 aralığında olduğunda ve $\gamma > 0$ olduğunda Denklem 3.27'deki işlem yapılmaktadır.

$$A_i^t \to 0, \quad r_i^t \to r_i^0, \text{ gibi } t \to \infty. \tag{3.27}$$

3.4. Performans Değerlendirme Ölçütleri

Önerilen yöntemin sınıflandırma başarısı test veri kümesi üzerinde karışıklık matrisi (confusion matrix) oluşturularak yapılmaktadır. Bu nedenle veri kümesi eğitim ve test olmak üzere ikiye ayrılır. Eğitim veri kümesi üzerinde model oluşturulduktan sonra, bu oluşturulan model test veri kümesinde sınanır (Coşkun, 2015). Karışıklık matrisinde satırlar test kümesindeki örneklere ait gerçek sayıları ifade etmektedir, sütunlar ise modelin tahminini ifade etmektedir (Coşkun, 2015).

Makine öğrenmesinde kullanılan sınıflandırma modellerinin performansını tanımlamak için hedef niteliğe ait tahminlerin ve gerçek değerlerin karşılaştırıldığı karışıklık matrisi son zamanlarda sıklıkla kullanılan bir matristir. Sınıflandırma tahminleri dört adet değerlendirmeden oluşmaktadır. Bunlar Doğru Pozitif (DP), Yanlış Negatif (YN), Yanlış Pozitif (YP) ve Doğru Negatiftir (DN). Sınıflandırma işleminin

doğruluğunun belirlenebilmesinde duyarlılık (sensitivity) ve özgüllük (specificity) kavramlarından yararlanılır (Öğücü, 2006a). Bunlar Denklem 3.28 ve Denklem 3.29'da verilmiştir.

DP: Doğru olarak pozitif olarak sınıflandırılan veri sayısının toplamıdır.

YN: Yanlış olarak negatif olarak sınıflandırılan veri sayısının toplamıdır.

YP: Yanlış olarak pozitif olarak sınıflandırılan veri sayısının toplamıdır.

DN: Doğru olarak negatif olarak sınıflandırılan veri sayısının toplamıdır.

Duyarlılık doğru olarak sınıflandırılmış pozitif veri sayısının, toplam pozitif sınıflandırılmış veri sayısına oranıdır. Duyarlılık formülü Denklem 3.28'de verilmiştir.

$$Duyarlılık = \frac{DP}{DP + YN} \tag{3.28}$$

Özgüllük gerçek negatif oranı olarak da adlandırılır, doğru olarak sınıflandırılmış negatif veri sayısının, toplam negatif sınıflandırılmış veri sayısına oranıdır. Özgüllük formülü Denklem 3.29'da verilmiştir.

$$\text{Özgüllük} = \frac{DN}{DN + YP} \tag{3.29}$$

Kesinlik doğru olarak sınıflandırılmış pozitif veri sayısının, pozitif değere sınıflandırılan toplam veri sayısına oranıdır. Kesinlik formülü Denklem 3.30'da verilmiştir.

$$kesinlik = \frac{DP}{DP + YP} \tag{3.30}$$

F ölçütü kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. F ölçütünün formülü ise Denklem 3.31'te verilmiştir.

$$F \text{ ölçütü} = \frac{2 \times kesinlik \times duyarlılık}{kesinlik + duyarlılık} \tag{3.31}$$

Tezde önerilen yöntemin sınıflandırma performansını değerlendirmek için sınıflandırma doğruluğu (SD) hesaplanmaktadır. SD, sınıflandırıcının doğru sonuç üretme kabiliyetini ölçmektedir. SD için formül Denklem 3.32'de verilmiştir (Bolaji ve ark., 2016).

$$SD = \frac{nc}{N} \tag{3.32}$$

Burada SD sınıflandırma doğruluğunu ifade etmektedir, nc doğru sınıflandırılmış örneklerin sayısıdır, N toplam örnek sayısıdır. Standart Sapma (SS) ise Denklem 3.33 kullanılarak hesaplanmaktadır.

$$SS = \sqrt{\frac{\sum|x - \bar{x}|^2}{N - 1}} \tag{3.33}$$

Burada SS standart sapmayı ifade etmektedir, x veri kümesinde bir değeri, $\bar{x}$ veri kümesinin aritmetik ortalamasını ve N veri kümesinde toplam örnek sayısını ifade etmektedir.

Sınıflandırma hata yüzdesi (Classification Error Percentage, CEP) hatalı olarak sınıflandırılmış örneklerin yüzdesidir. Bu tüm test verileri için hesaplanmaktadır. Yanlış olarak sınıflandırılmış toplam örnek sayısını, toplam test veri kümesi sayısına bölerek CEP hesaplanmaktadır (Ozturk, 2011). CEP formülü Denklem 3.34'te verilmiştir.

$$CEP = \frac{hatalı\ sınıflandırılan\ test\ örnek\ sayısı}{toplam\ test\ örnek\ sayısı} \times 100 \tag{3.34}$$

Makine öğrenmesi çalışmalarında uygulanan yöntemin başarısını test etmek için, veri kümesi eğitim ve test kümeleri olarak ayırılmaktadır. Bu ayrılma işlemi çeşitli şekillerde yapılmaktadır ve bunlardan biriside k kat çapraz onaylama yöntemidir. k kat çapraz onaylama yöntemi sınıflandırıcı modellerin veri kümesi üzerinde yapılan sınıflandırma işleminin sonuçlarının tutarlı olması için kullanılmaktadır (Kaya, 2014). İlk önce k değeri seçilir ve bu k değeri veri kümesinin kaç parçaya bölüneceğini belirtmektedir. Veri kümesi k adet parçalara ayrıldıktan sonra k kat çapraz onaylama işlemi çalışmaya başlar. Her adımda bölünen parçalardan birisi test kümesi olarak kullanılırken geriye kalan k-1 adet parça eğitim işlemi için kullanılmaktadır. Tüm adımlar tamamlandıktan sonra elde edilen sınıflandırma sonuçlarının ortalaması alınarak genel sınıflandırma doğruluğu veya hatası elde edilmektedir. Bu tez çalışmasında literatürde ilgili veri kümelerinde kullanılan k değerleri ile çapraz onaylama yapılmıştır.

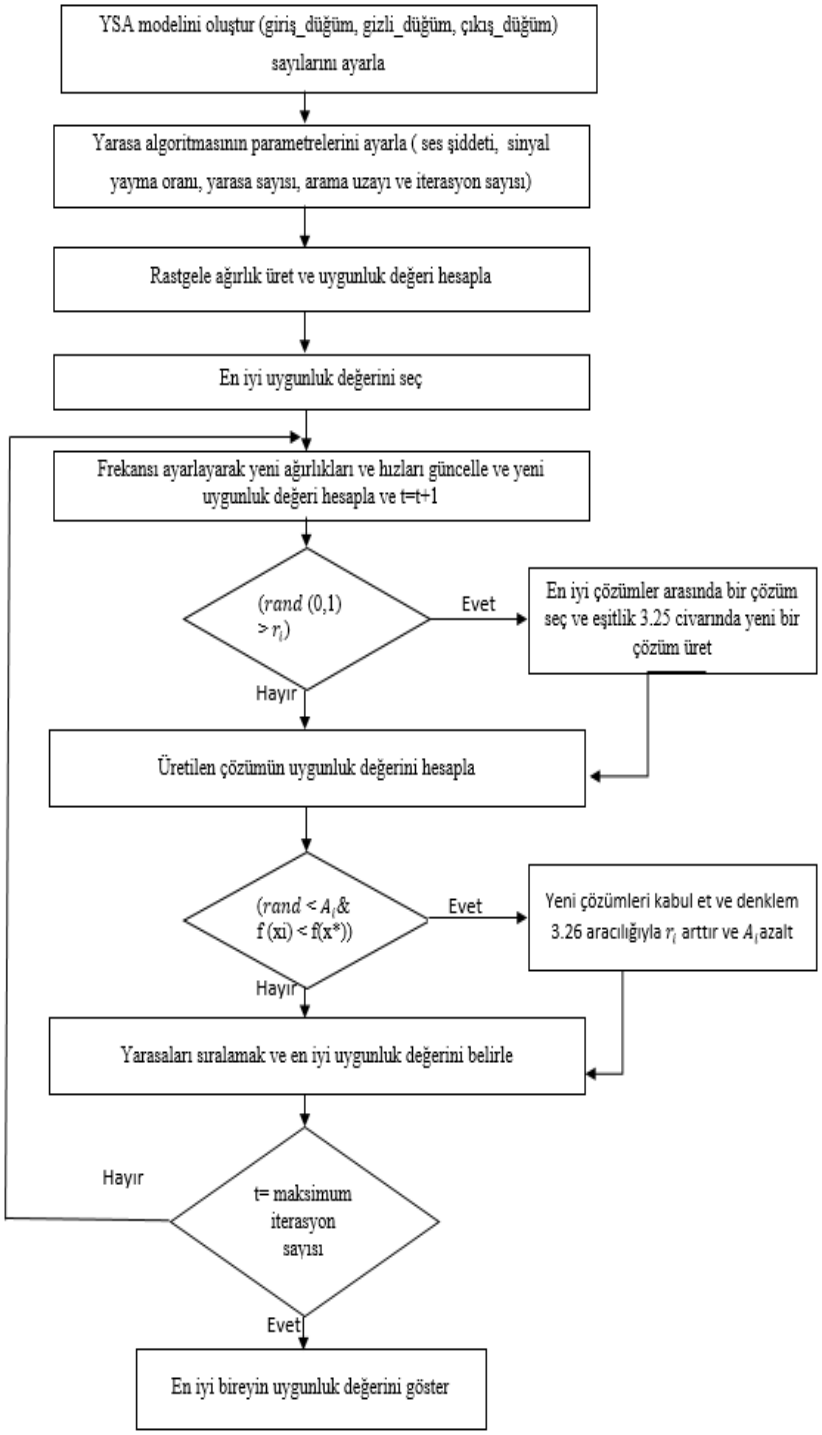
3.5. Önerilen Yöntem

Bu bölümde tez çalışmasında önerilen YA'nın çok katmanlı YSA modeline uyarlanması ile ilgili temel işlemler anlatılmıştır. Literatür incelendiğinde YA birçok yapay zekâ uygulamalarında ve mühendislik alanlarında kullanılmıştır. Bu tezde YA-YSA modelinde bulunan ağırlıkların güncellenmesinde kullanılmıştır. YA global bir arama yeteneğine sahip olduğundan dolayı, YSA modelinde en iyi sınıflandırma doğruluğunu veren ağırlıkların belirlenmesinde kullanılmıştır. Bu ağırlık değerleri YSA modelinin başarısını doğrudan etkilemektedir. YA, her iterasyonda daha iyi sonuca ulaşmak için bir uygunluk fonksiyonuna ihtiyaç duymaktadır. Bu tez çalışmasında elde edilen sonuçlar (Bolaji ve ark., 2016) tarafından elde edilen sonuçlar ile karşılaştırıldığında Denklem 3.35'de verilen SSE ve (Ozturk, 2011) tarafından elde edilen sonuçlarla karşılaştırıldığında Denklem 3.36'da verilen MSE uygunluk fonksiyonu olarak kullanılmıştır. Amaç verilen uygunluk fonksiyonu değerini en küçük olarak elde eden ağ ağırlıklarını belirlemektir. Dolayısıyla problem, minimizasyon problemi olarak ele alınmıştır.

$$SSE = \sum_{i=1}^{n} (t_i - y_i)^2 \tag{3.35}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (t_i - y_i)^2 \tag{3.36}$$

Burada, n veri kümesinde bulunan örnek sayısını, y önerilen YA-YSA yönteminin elde edilen çıkış değerini ve t ise istenen çıkış değerini ifade etmektedir. Önerilen YA tabanlı YSA yönteminin akış şeması Şekil 3.13'de verilmiştir.



Şekil 3.13. Önerilen YA-YSA yönteminin akış şeması.

Önerilen YA tabanlı YSA modelinin akış şemasına baktığımız zaman önce YSA modeli oluşturulur, YSA modelinin giriş-gizli-çıkış katmanları ayarlanır, Giriş katmanı nöron sayısı veri kümesinin giriş sayısına eşittir. Gizli katmandaki nöron sayısı ise (Ozturk, 2011) ve (Bolaji ve ark., 2016) çalışmalardan alınmıştır. Çıkış katmanı nöron sayısı ise veri kümesinin çıkış sayısına eşittir. YSA modeli oluştururken modelde

bulunan ağırlıkların sayısı Denklem 3.37 ile hesaplanmaktadır. Daha sonra YA'ya ait olan parametreler (ses şiddeti, sinyal yayma oranı, yarasa sayısı, arama uzayı ve iterasyon sayısı) kullanıcı tarafından girilmektedir. Bu parametreler girildikten sonra [-10, 10] arasında Denklem 3.37 ile hesaplanan değer kadar rasgele ağırlıklar belirlenerek yarasalar (çözüm vektörleri) elde edilir. Daha sonra bu çözüm vektörlerinin uygunluk değerleri hesaplanır. Uygunluk değerleri hesaplandıktan sonra en iyi uygunluk değeri seçilir ve belirlenen iterasyon sayısı kadar rastgele verilen ağırlıklar YA ile güncellenir. Maksimum iterasyon sayısına ulaştıktan sonra en iyi yarasa önerilen yöntemin sonucu olarak çıktı verilir.

$$\text{çözüm vektörü uzunluğu} = \text{giriş sayısı} * \text{gizli katmandaki nöron sayısı} * \text{çıkış sayısı} \quad (3.37)$$

4. DENEYSEL SONUÇLAR

4.1. Tezde Kullanılan Veri Kümeleri

Bu tez çalışmasında kullanılan veri kümeleri UCI makine öğrenmesi veri ambarından alınmıştır (Frank ve Asuncion, 2010). Genel olarak, bu veri kümeleri literatürde sınıflandırma problemleri için kullanılmaktadır. YA ile ağırlıkları güncellenen YSA modelinin farklı veri kümeleri arasında nasıl etkili bir şekilde çalışabileceğini incelemek için farklı özelliklere sahip olan veri kümeleri seçilmiştir. UCI makine öğrenmesi veri ambarından alınan 10 adet veri kümesi önerilen YA-YSA yönteminin sınıflandırma performansını test etmek için kullanılmıştır. Aşağıda bu veri kümeleri hakkınca bilgi verilmiştir:

İris verisi, sınıflandırma literatüründe bulunan en iyi bilinen veri kümesidir. Amaç, İris çiçeğini üç tür arasında sınıflandırmaktır (iris setosa, iris versicolor ve iris virginica). İris veri kümesi, 150 adet veri içermektedir. Bu veri kümesi 4 giriş ve 3 çıkıştan (sınıftan) oluşmaktadır. Her sınıf bir çeşit iris bitkisini ifade etmektedir.

Ionosphere veri kümesi toplam 351 adet veriden oluşmaktadır. Bu veri kümesi 33 giriş ve 2 çıkıştan oluşmaktadır.

Cam veri kümesi ele alınan tüm problem içinde ikinci en büyük sınıfa sahip bir veri kümesidir. Cam türlerini sınıflandırmak için kullanılmaktadır. Düz işlenmiş veya işlenmemiş bina pencereleri, araç pencereleri, yiyecek kapları, sofra takımları veya baş lambaları olarak sınıflandırmak için kullanılmaktadır. Cam veri kümesi 9 giriş ve 6 çıkıştan (sınıftan), toplam 214 adet veriden oluşmaktadır.

Tiroit hastalığı veri kümesi toplam 7200 adet veri ile en çok örnek içeren veri kümelerinden biridir. Bu veri kümesi hastanın tiroit fonksiyonun aşırı, normal veya anormal oluşuna göre 21 giriş ve 3 çıkış şeklinde sınıflandırmada kullanılmaktadır.

Kanser veri kümesi göğüs kanseri teşhis (iyi huylu veya kötü huylu tümörü sınıflandırır) verisidir. Göğüs kanseri veriş kümesi 9 adet giriş, 2 adet çıkış (sınıf) bilgisi içeren 699 veriden oluşan bir veri kümesidir.

Şeker hastalığı teşhis verisi olan diyabet veri kümesi, bir Pima Hintli bireyin diyabet teşhisinin pozitif olup olmadığına karar vermeye çalışmaktadır. Diyabet veri kümesi 8 giriş ve 2 çıkış toplam 768 veriden oluşmaktadır.

Magic veri kümesi, görüntüleme tekniğini kullanarak zemin tabanlı bir atmosferik Cherenkov gama teleskopunda yüksek enerjili gama parçacıklarının kaydını taklit etmek için üretilen verileri içermektedir. Bu veri kümesi 19020 örnek içerir her biri 10 giriş verisi ve 2 çıkışa (sınıfa) sahip bir veri kümesidir.

Kalp veri kümesi, kalp durumunun teşhisi amacıyla kullanılmaktadır. Kalp veri kümesi 270 adet veriden oluşmaktadır, her biri 13 adet giriş ve 2 çıkışa sahip bir veri kümesidir.

Australian kredi kartı veri kümesi toplam 690 adet veri içermektedir, veri kümesi 14 giriş ve 2 çıkışa (sınıfa) sahip bir veri kümesidir.

At veri kümesi, mide rahatsızlığı olan bir atın kaderini önceden tahmin edilmesinde kullanılmaktadır. At hayatta kalacak mı, ölecek mi yoksa ata ötenazi yapılmalı mı durumlarına göre üç tür sınıflandırma yapılmaktadır. At veri kümesi toplam 368 adet veriden oluşmaktadır. Bu veri kümesi 27 giriş ve 2 çıkışa sahip bir veri kümesidir.

Tez çalışmasında kullanılan bu 10 adet veri kümesinin özellikleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1. Tezde kullanılan veri kümelerin özellikleri

Veri Kümeleri	Giriş Sayısı	Çıkış Sayısı	Veri Sayısı
İris	4	3	150
Ionosphere	33	2	351
Cam	9	6	214
Tiroit	21	3	7200
Kanser	9	2	699
Diyabet	8	2	768
Magic	10	2	19020
Kalp	13	2	270
Kart	14	2	690
At	27	2	368

YA-YSA modeli Çizelge 4.1'de verilen her bir veri kümesinin özelliklerine göre yapılmaktadır. Giriş katmanı nöron sayısı veri kümesinin giriş sayısına eşittir. Gizli katmandaki düğüm sayısı ise (Ozturk, 2011) ve (Bolaji ve ark., 2016) çalışmalardan alınmıştır. Çıkış katmanı nöron sayısı ise veri kümesinin çıkış sayısına eşittir. YA-YSA modelinde gizli katmandaki çıkışlar ve çıkış katmandaki çıkışlar için sigmoid uygunluk fonksiyonu kullanılmıştır. Veri kümeleri YA-YSA modeline verilmeden önce her bir giriş 0 ile 1 aralığında Min-Max normalizasyon yöntemi ile normalize edilmiştir. Tez çalışmasında kullanılan her bir veri kümesi için YSA yapısı Çizelge 4.2'de verilmiştir.

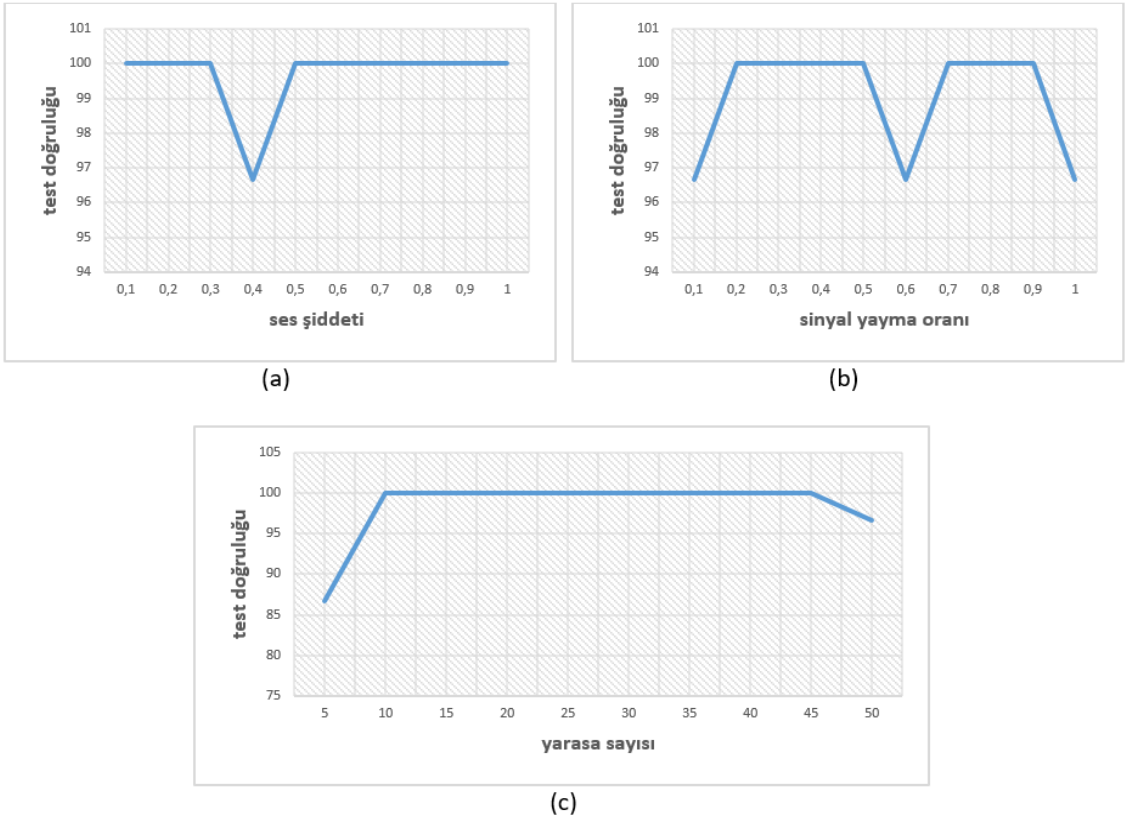
Çizelge 4.2. Tez çalışmasında kullanılan her bir veri kümesi için YSA yapısı

Veri Kümesi	YSA yapısı (Giriş-Gizli-Çıkış)
İris	4-5-3
Ionosphere	33-4-2
Cam	9-12-6
Tiroit	21-12-3
Kanser	9-8-2
Diyabet	8-6-2
Magic	10-4-2
Kalp	13-6-2
Kart	14-6-2
At	27-6-2

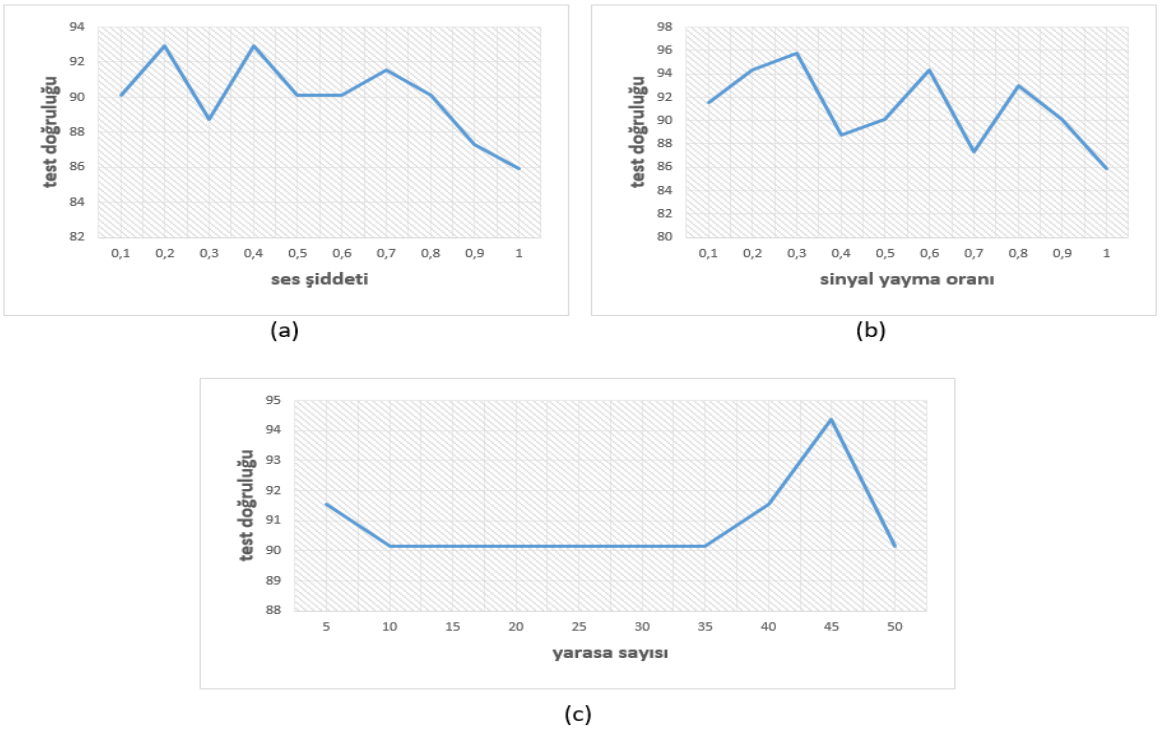
4.2. Parametre Analizi

Bu bölümde YA ile eğitilen YSA modelinden elde edilen sonuçlar literatürdeki çalışmaların sonuçları ile karşılaştırılmıştır. Önerilen YSA modeli visual studio 2015 programlama dilinde, Windows 10, işlemci Intel Core i5-2.40GHz ve Ram 6 GB özelliklerine sahip olan bilgisayarda kodlanmıştır. YA kullanılan ses şiddeti, sinyal yayma oranı ve yarasa sayısı parametreleri analiz edilmiştir. Ses şiddeti ve sinyal yayma oranı 0,1 ile 1,0 arasında 0,1 arttırılarak analiz edilmiştir. Frekans değeri minimum 0 maksimum 2 olarak alınmıştır. Yarasa sayısı parametresi ise 5 ile 50 arasında 5'er 5'er arttırılarak analiz edilmiştir. Ses şiddeti parametresi analiz edilirken sinyal yayma oranı parametresi değeri 0,5 ve yarasa sayısı parametresi değeri 10 alınmıştır. Sinyal yayma oranı parametresi analiz edilirken ses şiddeti parametresi değeri 0,5 ve yarasa sayısı parametresi değeri 10 alınmıştır. Yarasa sayısı parametresi analiz edilirken, ses şiddeti parametresi değeri 0,5 ve sinyal yayma oranı parametresi değeri 0,5 alınmıştır. Her bir veri kümesi için ses şiddeti, sinyal yayma oranı ve yarasa sayısı parametrelerinin sınıflandırma doğruluğuna olan etkisi Şekil 4.1 ile Şekil 4.10 arasında gösterilmiştir. Sınıflandırma işlemi yapılırken literatürdeki çalışmalarda kullanılan çapraz onaylama kullanılarak eğitim ve test verileri belirlenmiştir ve sonuçlar 1000 iterasyon sonucunda elde edilmiştir. Literatürde bulunan (Ozturk, 2011) çalışmasındaki verilerde 4 kat çapraz onaylama kullanılmıştır ve 30 kez çalıştırılmıştır. (Bolaji ve ark., 2016)

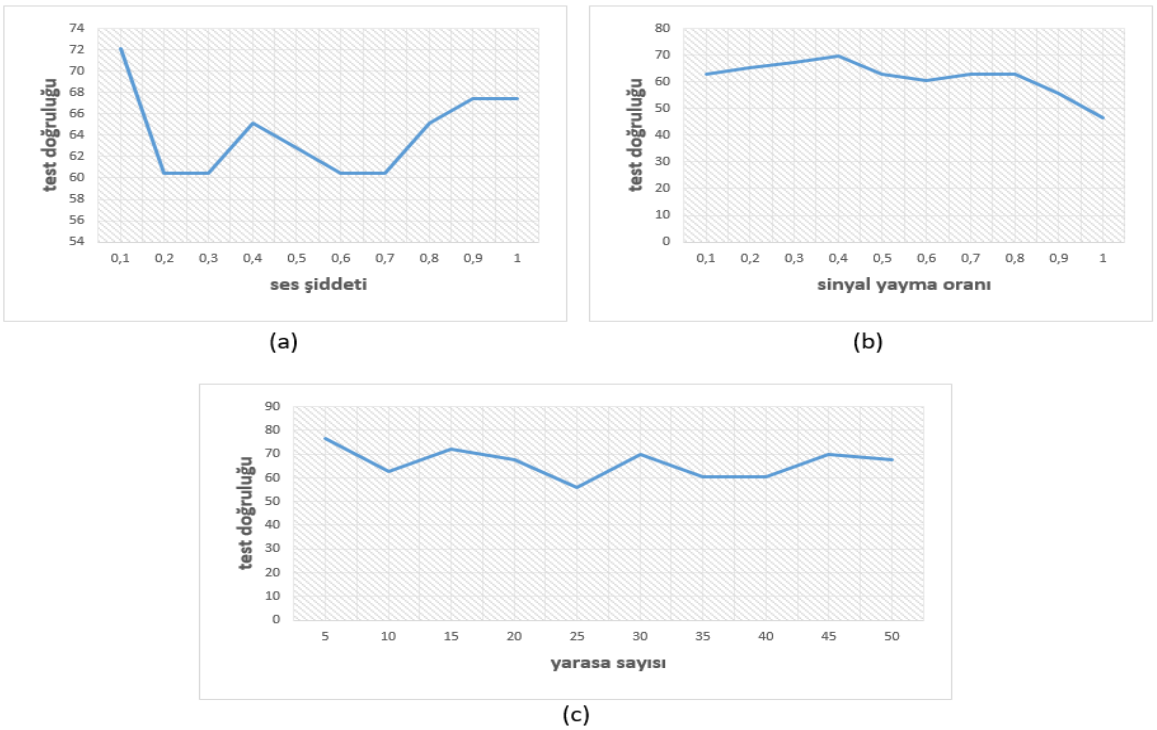
çalışmasındaki verilerde ise 5 kat çapraz onaylama kullanılmıştır ve 10 kez çalıştırılmıştır.



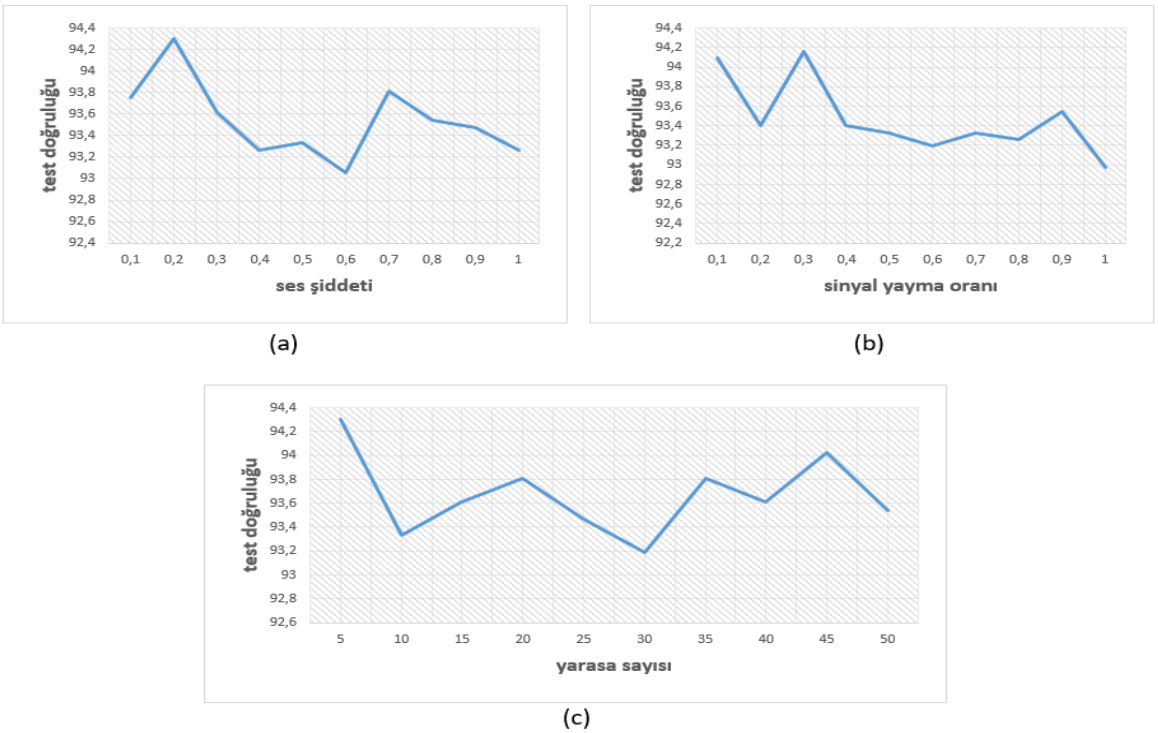
Şekil 4.1. İris veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



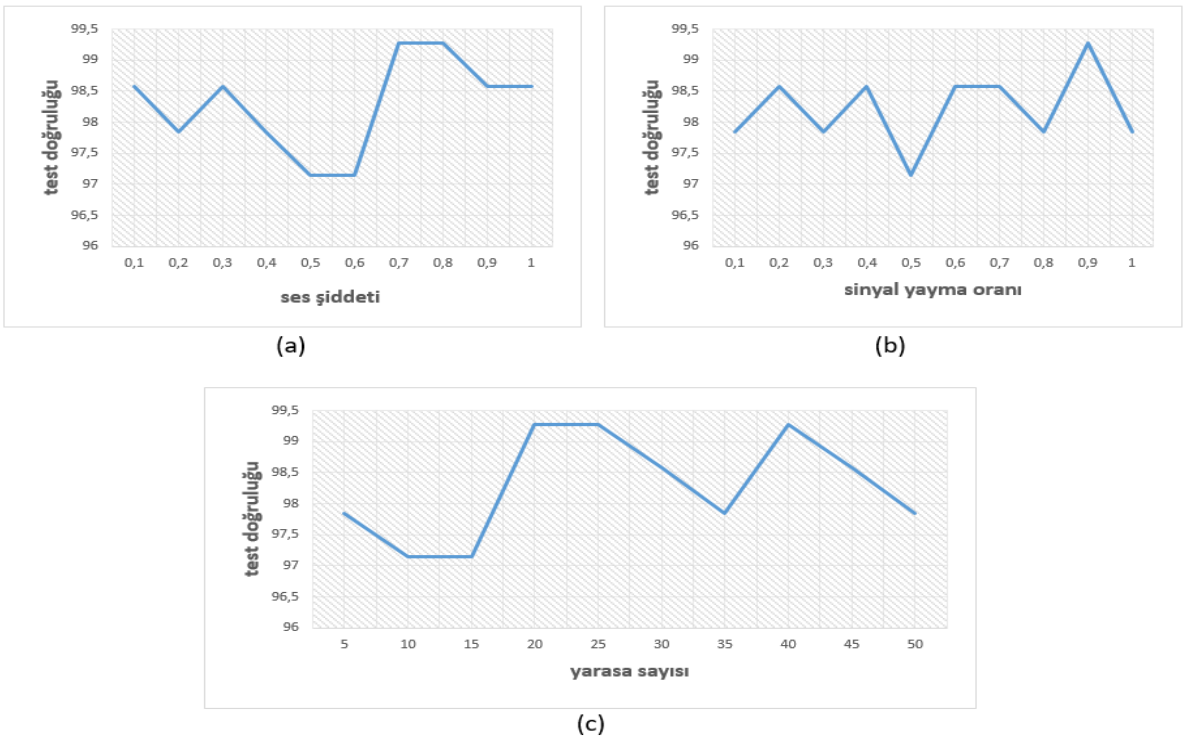
Şekil 4.2. Ionosphere veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



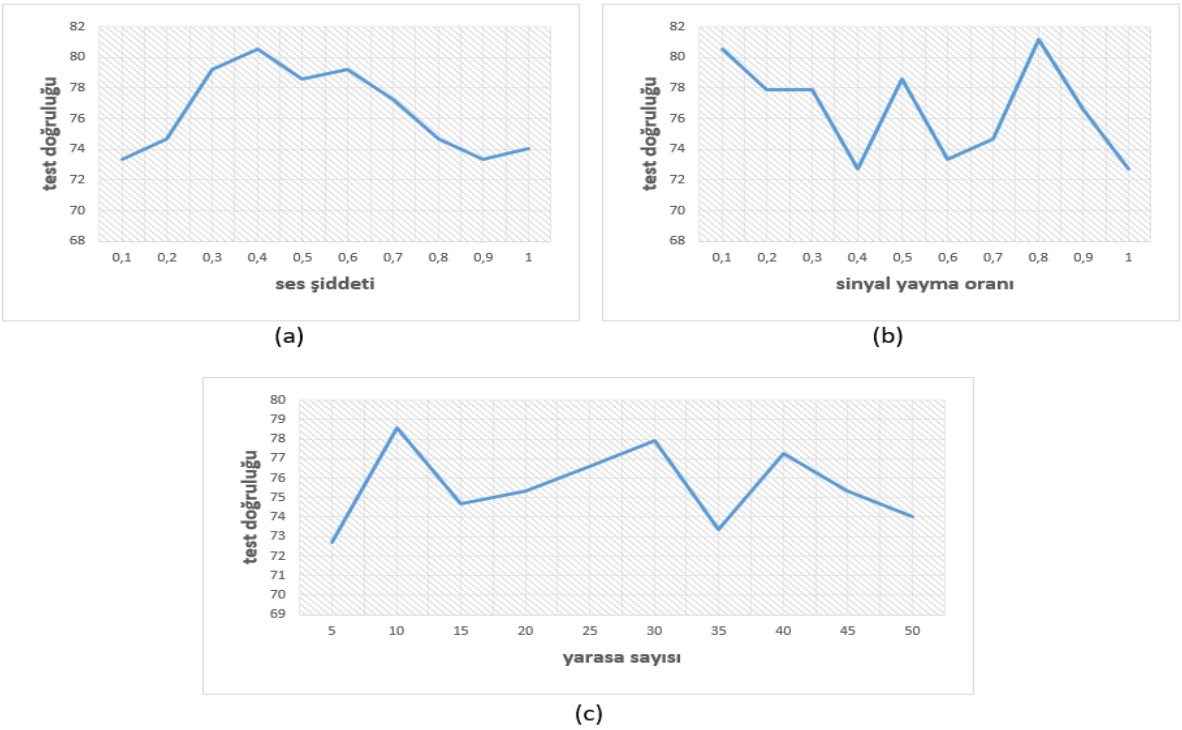
Şekil 4.3. Cam veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



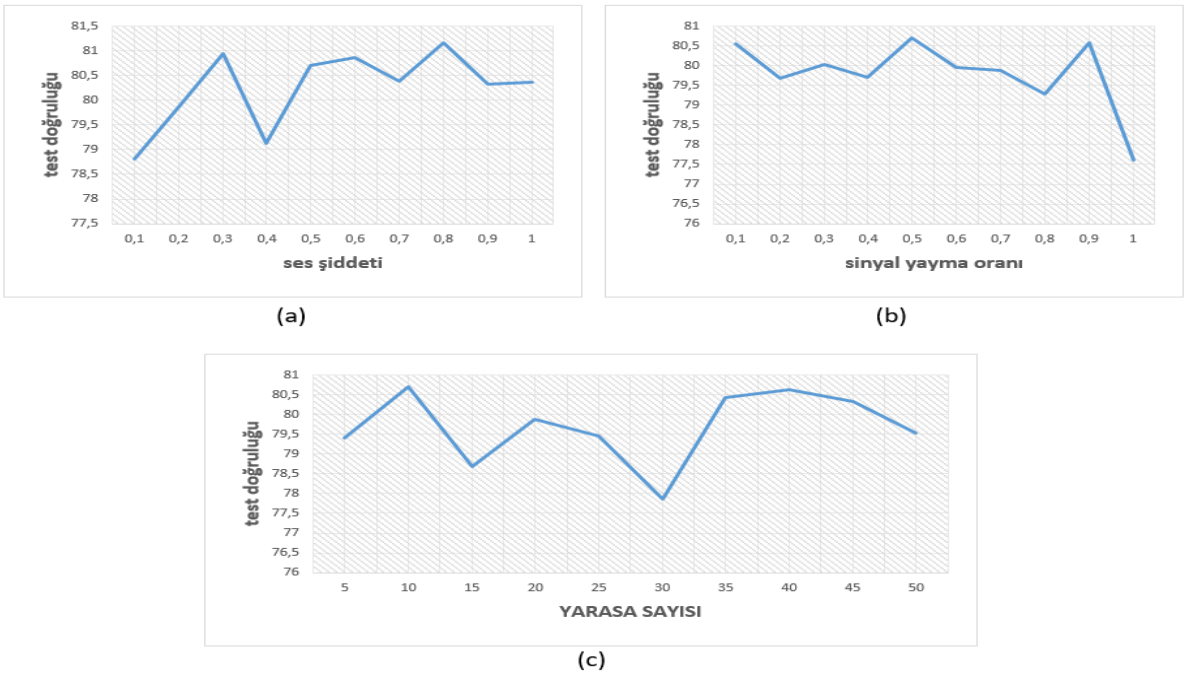
Şekil 4.4. Tiroit veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



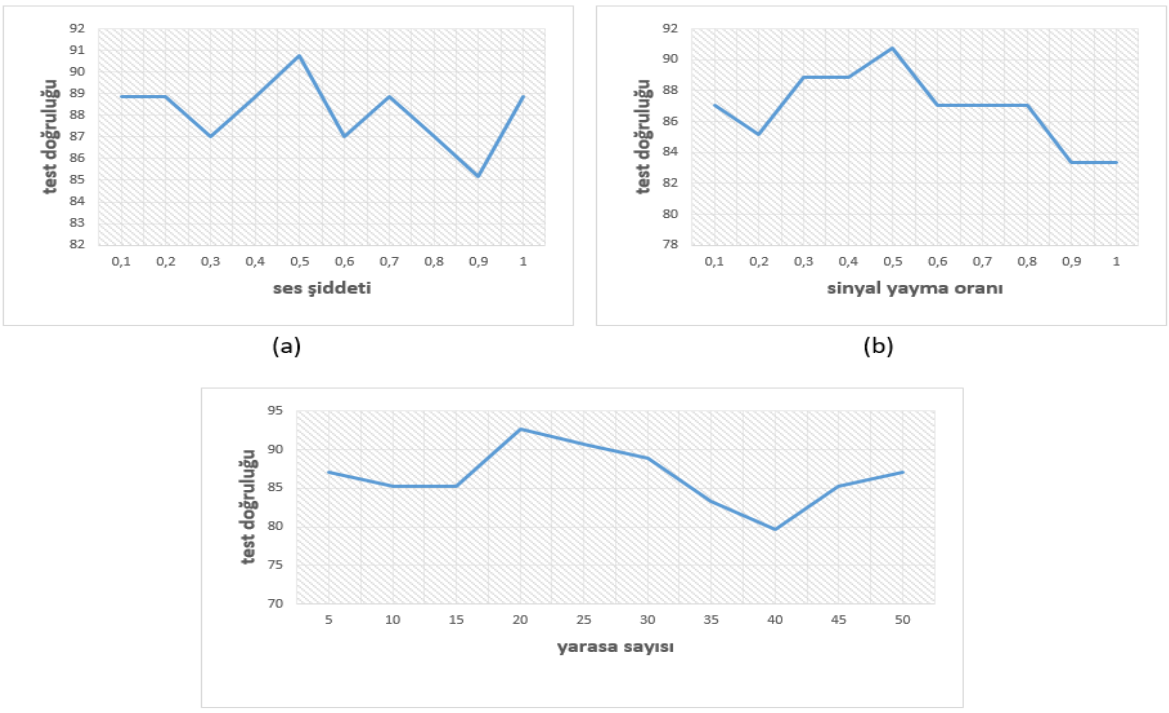
Şekil 4.5. Kanser veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



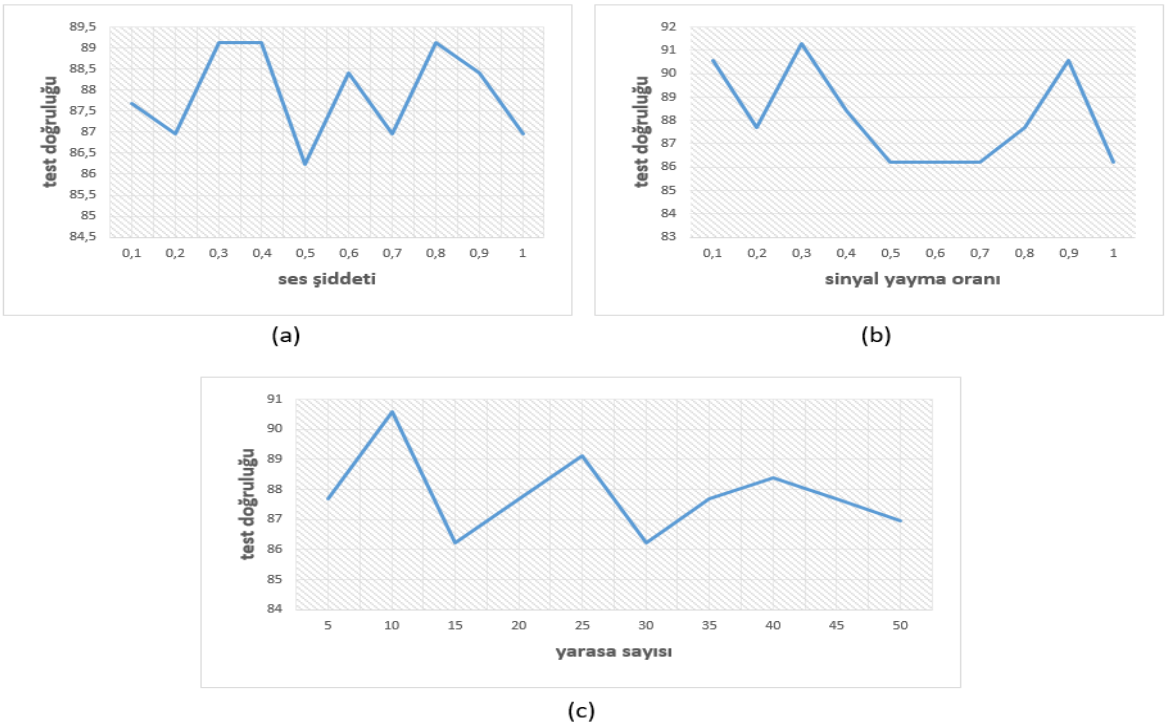
Şekil 4.6. Diyabet veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



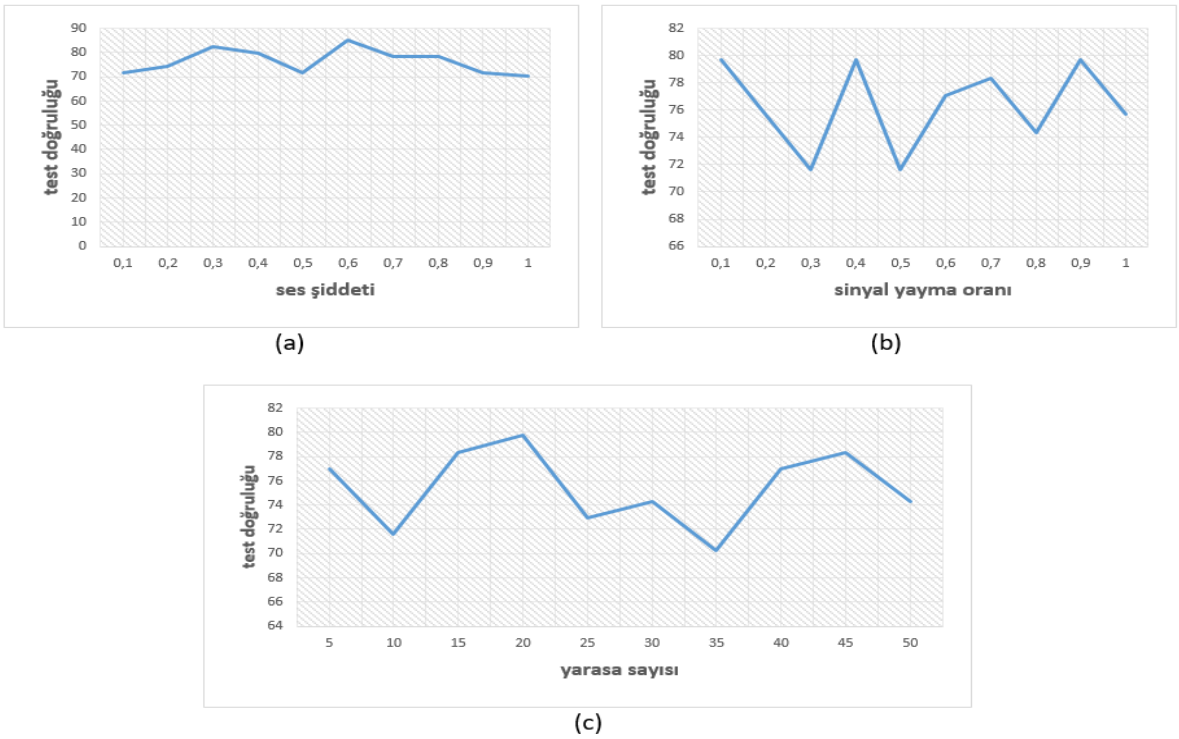
Şekil 4.7. Magic veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



Şekil 4.8. Kalp veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



Şekil 4.9. Kart veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi



Şekil 4.10. At veri kümesi için (a) ses şiddeti, (b) sinyal yayma oranı, (c) yarasa sayısı parametresinin sınıflandırma doğruluğuna etkisi

Her bir veri kümesi için en iyi sınıflandırma doğruluğunu veren parametre değerleri Çizelge 4.3'de verilmiştir.

Çizelge 4.3. Veri kümeleri için en iyi sınıflandırma doğruluğu veren parametre değerleri

Veri setleri	Ses şiddeti	Sinyal yayama oranı	Yarasa sayısı
İris	0,5	0,4	10
Ionosphere	0,2	0,3	45
Cam	0,1	0,4	5
Tiroit	0,2	0,3	5
Kanser	0,7	0,9	20
Diyabet	0,4	0,8	10
Magic	0,8	0,5	10
Kalp	0,5	0,5	20
Kart	0,4	0,3	10
At	0,6	0,1	20

Önerilen YA-YSA modeli için Çizelge 4.3'deki parametre değerleri kullanılarak çalıştırılmış ve literatürdeki çalışmaların sonuçları ile karşılaştırılmıştır. Test veri kümeleri için elde edilen en iyi sınıflandırma doğruluğu, en kötü sınıflandırma

doğruluğu, ortalama sınıflandırma doğruluğu, SS ve çalışma süreleri Çizelge 4.4'te verilmiştir. Aynı zamanda her veri kümesi için performans değerlendirme ölçütlerinden elde edilen Duyarlılık, Özgüllük, Kesinlik ve F ölçütü değerleri Çizelge 4.5'te verilmiştir.

Çizelge 4.4. Tüm veri kümeleri için test en iyi, en kötü, ortalama, standart sapma ve zaman

Veri Kümesi	En iyi	En kötü	Ortalama	Standart sapma	Zaman
İris	100,00	53,33	83,33	20,06	00:00:13
Ionosphere	94,37	80,28	84,92	3,93	00:00:58
Cam	74,41	41,86	56,51	10,04	00:00:18
Tiroit	94,30	91,45	92,76	0,76	00:12:04
Kanser	99,28	93,57	96,57	1,57	00:01:21
Diyabet	76,62	65,58	69,22	0,03	00:00:58
Magic	80,88	78,20	79,63	0,82	00:22:07
Kalp	87,03	70,37	80,37	4,79	00:00:28
Kart	89,13	78,26	82,60	3,90	00:01:01
At	74,32	67,56	71,75	2,48	00:01:46

Çizelge 4.5. Tüm veri kümeleri için performans değerlendirme ölçütleri sonuçları

Veri Kümesi	Duyarlılık	Özgüllük	Kesinlik	F ölçütü
İris	0,833	0,916	0,833	0,833
Ionosphere	0,849	0,849	0,849	0,849
Cam	0,565	0,913	0,565	0,565
Tiroit	0,927	0,963	0,927	0,927
Kanser	0,965	0,965	0,965	0,965
Diyabet	0,692	0,692	0,692	0,692
Magic	0,796	0,796	0,796	0,796
Kalp	0,803	0,803	0,803	0,803
Kart	0,826	0,826	0,826	0,826
At	0,717	0,717	0,717	0,717

4.3. Karşılaştırma Sonuçları

Bu bölümde kanser, diyabet, kalp, kart, cam, at ve tiroit olmak üzere yedi adet veri kümeleri için elde edilen sonuçların (Ozturk, 2011) çalışmaların sonuçları ile karşılaştırılması verilmiştir. Çizelge 4.6'da önerilen yöntemin performansını değerlendirmek için literatürde bulunan popülasyon tabanlı (GA, PSO, YAKA, DE algoritması) ve gradyan tabanlı (LM, GY) optimizasyon algoritmaları ile

karşılaştırılmıştır. Bu bölümde (Ozturk, 2011) çalışmadaki gibi 4 kat çapraz onaylama kullanılmıştır ve deneyler her bir problem için 30 kez çalıştırılmıştır ve 1000 iterasyon için uygulanmıştır. Problemde veri kümesi iki parça halinde uygulanır, bunlar eğitim seti ve test seti. Sinir ağı eğitim verileriyle eğitildikten sonra YSA'nın hiç görmediği test verileri ağa sunulur ve ağın bu verilere karşı vereceği tepki hesaplanır. Önerilen YA-YSA modeli için kullanılan parametre değerleri Çizelge 4.3'deki gibi alınmıştır ve ağ mimarisi Çizelge 4.2'deki YSA yapısı (giriş-gizli-çıkış) ve Çizelge 4.1'deki gibi örnek sayısı verilmiştir.

Algoritmanın sınıflandırma başarılarının analizi için CEP ve SS hesaplanmıştır. CEP test verileri için hesaplanmakta ve yanlış olarak sınıflandırılmış toplam örnek sayısını test veri kümesine bölünerek hesaplanmaktadır. Ortalama CEP ve SS değerleri Çizelge 4.6'da verilmiştir.

Çizelge 4.6. Test ortalama CEP hata ve standart sapma değerleri

Veri Kümesi		YA (Önerilen yöntem)	YAKA (Ozturk, 2011)	PSO (Ozturk, 2011)	DE (Ozturk, 2011)	GA (Ozturk, 2011)	GY (Ozturk, 2011)	LM (Ozturk, 2011)
Kanser	CEP	3,42	1,14	2,01	1,19	1,35	1,89	8,18
	SS	1,07	0,00	0,86	0,28	0,28	0,43	10,29
Diyabet	CEP	24,68	25,22	27,50	24,84	26,51	28,27	29,80
	SS	4,51	0,97	2,05	1,32	1,21	6,30	3,77
Kalp	CEP	18,43	19,48	21,87	20,33	20,60	21,44	24,96
	SS	3,85	1,41	1,45	1,29	0,68	0,55	7,40
Kart	CEP	14,73	13,53	15,58	13,90	13,56	13,86	21,59
	SS	3,04	1,17	1,59	1,26	1,21	0,47	7,98
Cam	CEP	47,28	45,62	52,49	44,90	50,18	59,09	42,99
	SS	7,38	3,11	7,14	2,84	3,15	9,52	11,55
At	CEP	28,84	28,63	31,14	29,29	29,12	27,84	33,79
	SS	3,49	2,61	3,94	2,72	2,84	2,12	4,50
Tiroit	CEP	7,15	6,95	7,27	7,08	7,09	7,26	7,22
	SS	0.56	0,01	0,02	0,95	0,03	0,00	0,07

Veri kümelerine baktığımız zaman, kanser veri kümesinde önerilen yöntem %3,42 oranında ortalama CEP değeri ve 1,07 oranında SS değeri elde etmiştir, bu veri kümesi YAKA %1,14, PSO %2,01, DE %1,19, GA %1,35, GY %1,89 ve LM %8,18 oranında ortalama CEP değerini üretmişler.

Diyabet veri kümesinde önerilen yöntem %24,68 ortalama CEP değeri ve 4,51 SS değeri ile en iyi sonucu gösteren algoritmadır. ABC %25,22, PSO %27,50, DE

%24,84, GA %26,51, GY %28,27 ve LM %29,80 oranında ortalama CEP değerini elde etmişler.

Kalp veri kümesinde önerilen yöntem %18,43 ortalama CEP değeri ile birinci en iyi sonucu elde etmiştir ve tüm algoritmaları geride bırakmıştır, YAKA algoritması %19,48 ortalama CEP değeriyle ikinci en iyi sonucu elde etmiş algoritmadır. PSO %21,87, DE %20,33, GA %20,60, GY %21,44 ve LM %24,96 oranında CEP değerlerini elde etmişlerdir.

Kart veri kümesinde önerilen yöntem %14,73 oranında ortalama CEP değeri elde etmiştir. YAKA 13,53, PSO %15,58, DE %13,90, GA %13,56, GY %13,86 ve LM %21,59 oranında ortalama CEP değerlerini üretmişlerdir. Kart veri kümesine bakıldığı zaman LM hariç tüm algoritmaların ortalama CEP değerleri %13-15 arasında değerlerdir.

Cam veri kümesi en büyük sınıfa (6 sınıflı) sahiptir. Bu veri kümesinde önerilen yöntem % 47,28 ortalama CEP değerini üretmiştir. YAKA %45,62, PSO %52,49, DE %44,90, GA %50,18, GY %59,09 ve LM %42,99 ortalama CEP değerleri elde etmişlerdir.

At veri kümesinde CEP değerlerine bakıldığı zaman %27-34 arasında farklılık göstermektedir. At veri kümesi için önerilen yöntem %28,84 CEP değerini elde etmiştir, YAKA %28,63, PSO %31,14, DE %29,29, GA %29,12, GY %27,84ve LM %33,79 CEP değerleri elde etmişlerdir

Tiroit veri kümesi en fazla örnek içeren sınıflandırma problemlerinden birisidir. CEP değeri tüm algoritmalar için sırasıyla YAKA %6,95, DE %7,08, GA %7,09, önerilen yöntem %7,15, LM %7,22, GY %7,26 Ve PSO %7,27

Önerilen yöntem, YAKA, PSO, DE, GA, GY VE LM optimizasyon algoritmaların her bir veri kümesi için başarı sıraları Çizelge 4.7’da verilmiştir, her bir veri kümesi için en iyi performans üretmiş olandan, en düşük performans gösterene göre algoritmalar sıralandıklarında YAKA, DE, YA-YSA (önerilen yöntem), GA, GY, PSO ve LM şeklinde sırlanmaktadırlar.

Çizelge 4.7. Sınıflandırma başarılarına (CEP) göre tüm algoritmalarının problemler için sıralaması

Veri Kümesi	YA-YSA (Önerilen Yöntem)	YAKA (Ozturk, 2011)	PSO (Ozturk, 2011)	DE (Ozturk, 2011)	GA (Ozturk, 2011)	GY (Ozturk, 2011)	LM (Ozturk, 2011)
Kanser	6	1	5	2	3	4	7
Diyabet	1	3	5	2	4	6	7
Kalp	1	2	6	3	4	5	7
Kart	5	1	6	4	2	3	7
Cam	4	3	6	2	5	7	1
At	3	2	6	5	4	1	7
Tiroit	4	1	7	2	3	6	5
Toplam	3.4	1.8	5.8	2.8	3.6	4.5	5.8
	(3)	(1)	(6)	(2)	(4)	(5)	(7)

Çizelge 4.8'de İris, Ionosphere, cam, tiroit, kanser, diyabet ve magic veri kümeleri için elde edilen sonuçların literatürdeki (Bolaji ve ark., 2016) çalışmaların sonuçları ile karşılaştırılması verilmiştir. Çizelge 4.8'de karşılaştırılan algoritmalar HF algoritması, KHA, GA ve HA algoritmasıdır. Çizelge 4.8'de verilen sonuçlar test veri kümeleri için en iyi sınıflandırma doğruluklarıdır. Burada (Bolaji ve ark., 2016) literatürdeki çalışmadaki gibi 5 kat çapraz onaylama kullanılmıştır, 10 kez çalıştırılmıştır ve 1000 iterasyon ile uygulanmıştır. Önerilen yöntem için kullanılan parametre değerleri Çizelge 4.3'de verilmiştir.

Çizelge 4.8. Tüm veri kümeleri için en iyi sınıflandırma doğruluğu (SD)

Veri Kümesi	YA (Önerilen Yöntem)	HF (Bolaji ve ark., 2016)	KHA (Bolaji ve ark., 2016)	HA (Bolaji ve ark., 2016)	GA (Bolaji ve ark., 2016)
İris	100,00	100,00	100,00	96,67	90,00
Ionosphere	94,37	90,14	91,43	94,37	94,37
Cam	74,41	60,47	58,14	72,09	67,44
Tiroit	94,30	93,82	92,90	92,78	92,57
Kanser	99,28	96,43	-----	100,00	98,57
Diyabet	76,62	66,88	-----	77,27	79,87
Magic	80,88	72,41	-----	81,18	77,87

Çizelge 4.8'den de görüldüğü gibi önerilen YA-YSA modeli 7 veri kümesinden 4 veri kümesinde diğer algoritmalardan daha iyi sonuç vermiştir. Veri kümelerine baktığımız zaman, İris veri kümesinde önerilen yöntem %100 sınıflandırma doğruluğu

elde etmiştir. İris veri kümesi için HA (%96,67) ve GA (%90,00)'dan daha iyi sonuçlar elde edilmiştir. Yedi adet veri kümesi arasında en büyük giriş sayısına (33) sahip olan Ionosphere veri kümesinde %94,37 SD elde edilmiştir. Önerilen yöntem FWA (%90,14) ve KHA (%91,43) algoritmalardan daha iyi bir sonuca ulaşmıştır. En fazla sınıfa (6 sınıflı) sahip ve 214 örnek içeren Cam veri kümesinde ise %74,41 değerinde SD elde edilmiştir. Bu veri kümesi için önerilen yöntem diğer yöntemlerden (HF, KHA, HA, GA) daha yüksek sınıflandırma doğruluğuna ulaşmıştır. Tiroit veri kümesi için % 94,30 değerinde en iyi SD elde edilmiştir. Bu veri kümesi için HF (%93,82), KHA (%92,90), HA(%92,78) ve GA (%92,57) algoritmalarından daha iyi SD değerine ulaşılmıştır. Kanser veri kümesinde önerilen yöntem %99,28 değeriyle ikinci en iyi SD değerine ulaşmıştır. Bu veri kümesi için HF (%96,43) ve GA'dan (%98,57) daha iyi sonuçlar elde edilmiştir. Diyabet veri kümesinde önerilen yöntem %76,62 değeriyle üçüncü en iyi SD değerine ulaşmıştır. Magic veri kümesinde ise önerilen yöntem % 80,88 değeri ile ikinci en iyi SD değerine ulaşmıştır. Bu veri kümesi için HF (%96,43) ve GA'dan (%98,57) daha iyi bir SD elde edilmiştir. Kanser, Diyabet ve Magic veri kümelerinin sonuçlarına göre önerilen YA-YSA modeli en iyi sınıflandırma doğruluğuna sahip olan algoritmaların sınıflandırma doğruluğuna çok yakın bir sınıflandırma doğruluğuna ulaşmıştır. İris, Ionosphere, cam ve tiroit veri kümeleri için önerilen yöntem en iyi SD sonucuna ulaşmıştır. Çizelge 4.8'de sınıflandırma doğruluklarına baktığımız zaman önerilen YA'nın YSA'nın eğitilmesinde üstün bir performansa sahip olduğunu göstermektedir. Çizelge 4.9'de tüm veri kümeleri için algoritmaların başarı sırası verilmiştir. Çizelge 4.9'de her bir veri kümesi için en iyi performans üretmiş olandan, en düşük performans gösterene göre algoritmalar sıralandıklarında; YA (Önerilen Yöntem), HA, GA, KHA ve HF şeklinde sırlanmaktadırlar.

Çizelge 4.9. sınıflandırma başarılarına (CA) göre tüm algoritmalarının veri kümeleri için sıralaması

Veri Kümesi	YA (Önerilen Yöntem)	HF (Bolaji ve ark., 2016)	KHA (Bolaji ve ark., 2016)	HA (Bolaji ve ark., 2016)	GA (Bolaji ve ark., 2016)
İris	1	1	1	2	3
Ionosphere	1	3	2	1	1
Cam	1	4	5	2	3
Tiroit	1	2	3	4	5
Kanser	2	4	-------	1	3
Diyabet	3	4	-------	2	1
Magic	2	4	-------	1	3
Toplam	1,5	3,1	2,8	1,8	2,7
	(1)	(5)	(4)	(2)	(3)

5. SONUÇ

YSA'da ağırlıkların güncellenmesi doğrudan sınıflandırma başarısını etkilemektedir. Dolaysıyla bu ağırlıkların iyi bir öğrenme algoritması ile güncellenmesi gerekmektedir. Bu tez çalışmasında yarasa optimizasyon algoritması kullanarak bu ağırlıkların güncellenmesi gerçekleştirilmiştir ve önerilen yönteme YA-YSA adı verilmiştir. Yöntemin global arama özelliği ile belirlenen ağırlıkların değeri arasında en iyi veya optimum ağırlık değerlerinin bulunması sayesinde YSA'nın sınıflandırma başarısı artmaktadır. Bu nedenle, önerilen yöntemin YSA sınıflandırma algoritması için umut verici bir araç olarak kullanılabilir. Önerilen yöntem 10 farklı veri kümesi üzerinde test edilmiştir. YA parametrelerinden ses şiddeti, sinyal yayılma oranı ve yarasa sayısı parametrelerinin her bir veri kümesi için ayrıntılı analizi yapılmıştır. Test sonuçları literatürdeki sınıflandırıcıların sonuçları ile karşılaştırılmıştır. Çoğu durumda önerilen yöntem literatürdeki sınıflandırıcılardan daha başarılı sonuçlara ulaşmıştır. Ayrıca, bu yöntem birçok tipte sinir ağları ile çalışmaktadır, özellikle de gradyan öğrenme yönteminin uygulanamadığı veya kullanımının eğitim prosedürü yakınsaması açısından kötü sonuçlara yol açtığı durumlarda kullanılabilir. Özellikle YA, benzersiz transfer (aktivasyon) fonksiyonuna sahip olan sinir ağları veya nöron ağın katmanlarında kullanılan çeşitli fonksiyon setlerine sahip olan YSA'lar için çok yararlı olabilir. Bu durumda önerilen yöntem çok başarılı bir şekilde yüz tanıma, hastalık teşhisi, örüntü tanıma gibi birçok sınıflandırma problemlerinde kullanılabilir.

KAYNAKLAR

Ahadzadeh, B. ve Menhaj, M. B., 2014, Training neural networks using modified differential evolution algorithm for classification problems, *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*, 598-603.

Alataş, B., 2007, Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme, Doktora Tezi, Fırat Üniversitesi, *Fen Bilimleri Enstitüsü*.

Arsan, T. ve Çimenli, S., 2015, Churn Analysis And Prediction With Decision Tree And

Artificial Neural Network, *Kadir Has University* 48.

Arsan, T. ve Çimenli, S., 2016, Churn Analysis and Prediction, *International Journal of Computer Science and Information Security*, 14 (8), 550.

Askarzadeh, A. ve Rezazadeh, A., 2013, Artificial neural network training using a new efficient optimization algorithm, *Applied Soft Computing*, 13 (2), 1206-1213.

Blum, C. ve Roli, A., 2003, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM computing surveys (CSUR)*, 35 (3), 268-308.

Bolaji, A. L. a., Ahmad, A. A. ve Shola, P. B., 2016, Training of neural network for pattern classification using fireworks algorithm, *International Journal of System Assurance Engineering and Management*, 1-8.

Canayaz, M., 2015, Cırcır böceği algoritması: Yeni bir meta-sezgisel yaklaşım ve uygulamaları, *İnönü Üniversitesi*, 77.

Canayaz, M. ve Karcı, A., 2015, Investigation of cricket behaviours as evolutionary computation for system design optimization problems, *Measurement*, 68, 225-235.

Carvalho, M. ve Ludermir, T. B., 2007, Particle swarm optimization of neural network architectures andweights, *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, 336-339.

Chen, J.-F., Do, Q. H. ve Hsieh, H.-N., 2015, Training artificial neural networks by a hybrid PSO-CS algorithm, *Algorithms*, 8 (2), 292-308.

Coşkun, C., 2015, Veri madenciliği algoritmaları karşılaştırılması, Master, *Dicle Üniversitesi*.

da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B. ve dos Reis Alves, S. F., 2017, Artificial Neural Network Architectures and Training Processes, In: Artificial Neural Networks, Eds: Springer, p. 21-28.

Delice, Y., 2008, Parçacık Sürü Optimizasyonu ile Yapay Sinir Ağlarından Sınıflandırma Kuralı Çıkarımı, *Yüksek Lisans Tezi, Erciyes Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı, Kayseri*.

Demir, M., 2015, Sezgisel yöntemlerde altın oran, *İnönü Üniversitesi*, 129.

Faris, H., Aljarah, I. ve Mirjalili, S., 2016, Training feedforward neural networks using multi-verse optimizer for binary classification problems, *Applied Intelligence*, 45 (2), 322-332.

Fenton, M. B., 2004, Bat natural history and echolocation, *Bat Echolocation Research: tools, techniques and analysis*, 2.

Fister Jr, I., Fister, D. ve Yang, X.-S., 2013, A hybrid bat algorithm, *arXiv preprint arXiv:1303.6310*.

Frank, A. ve Asuncion, A., 2010, UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California, *School of information and computer science*, 213.

Gallo, C., 2015, Artificial Neural Networks Tutorial, In: Encyclopedia of Information Science and Technology, Third Edition, Eds: IGI Global, p. 6369-6378.

Gandomi, A. H., Yang, X.-S., Alavi, A. H. ve Talatahari, S., 2013, Bat algorithm for constrained optimization tasks, *Neural Computing and Applications*, 22 (6), 1239-1255.

Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, 13 (5), 533-549.

Gurney, K., 1997, An introduction to neural networks, CRC press, p.

Güler, İ. ve Übeyli, E. D., 2006, Çok Katmanli Perseptron Sinir Ağlari İle Diyabet Hastaliğinin Teşhisi, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 21 (2).

Günler, M. A., 2012, Identification of Facial Expressions Using Artificial Neural Networks (Ann), *a Master's Thesis, Electrical & Electronics Engineering*.

Haykin, S., 1999, Neural Networks A comprehensive Foundation: Pretice Hall International, *Inc., Englewood Cliffs*.

Haykin, S. S., 2009, Neural networks and learning machines, Pearson Upper Saddle River, NJ, USA:, p.

Kankal, M. ve Uzlu, E., 2017, Neural Network Approach With Teaching–Learning-Based Optimization For Modeling And Forecasting Long-Term Electric Energy Demand In Turkey, *Neural Computing and Applications*, 28 (1), 737-747.

Kaya, E., 2014, Genetik algoritma ile bulanık kural kümesinin otomatik olarak oluşturulmasında yeni bir yaklaşım, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Kılıç, G., 2015, Yapay Sinir Ağlari İle Yemekhane Günlük Talep Tahmini, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*.

Koç, İ., 2016, Sınıflandırma problemlerinde meta-sezgisel optimizasyon yöntemlerinin özellik seçimi ve ayrıklaştırma amacıyla kullanımı, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Kohonen, T., 1990, The self-organizing map, *Proceedings of the IEEE*, 78 (9), 1464-1480.

Kowalski, P. A. ve Łukasik, S., 2016, Training neural networks with krill herd algorithm, *Neural Processing Letters*, 44 (1), 5-17.

Kriesel, D., 2007, A brief introduction on neural networks, p.

Kulluk, S., 2009, Karınca koloni optimizasyonu ile yapay sinir ağlarından kural çıkarımı, *Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı, Kayseri*.

Küçüksayacıgil, F., 2014, Use of genetic algorithms in multi-objective multi-project resource constrained project scheduling.

Liao, S.-H., Hsieh, J.-G., Chang, J.-Y. ve Lin, C.-T., 2015, Training neural networks via simplified hybrid algorithm mixing Nelder–Mead and particle swarm optimization methods, *Soft Computing*, 19 (3), 679-689.

Lourakis, M. I., 2005, A brief description of the Levenberg-Marquardt algorithm implemented by levmar, *Foundation of Research and Technology*, 4 (1), 1-6.

Mirjalili, S., Hashim, S. Z. M. ve Sardroudi, H. M., 2012, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, *Applied Mathematics and Computation*, 218 (22), 11125-11137.

Mirjalili, S. Z., Saremi, S. ve Mirjalili, S. M., 2015, Designing evolutionary feedforward neural networks using social spider optimization algorithm, *Neural Computing and Applications*, 26 (8), 1919-1928.

Nakamura, R. Y., Pereira, L. A., Costa, K., Rodrigues, D., Papa, J. P. ve Yang, X.-S., 2012, BBA: a binary bat algorithm for feature selection, *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*, 291-297.

Ozturk, C., 2011, Yapay Sinir Ağlarının Yapay Arı Koloni Algoritması İle Eğitilmesi, Doktora Tezi., Erciyes Universitesi, *Fen Bilimleri Enstitüsü*.

Öğücü, M. O., 2006a, Yapay Sinir Ağları İle Sistem Tanıma, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, 85.

Öğücü, M. O., 2006b, Yapay Sinir Ağları İle Sistem Tanıma, *Fen Bilimleri Enstitüsü*.

Özdemir, A., 2010, Genetik AlgorİtmaİleYapay Sinir Ağlarında Yapi Ve Parametre Optimizasyonu, *FIRAT ÜNİVERSİTESİ*.

Özkış, A., 2013, Sayısal optimizasyon problemlerinin çözümü için yapay arı kolonisi algoritmasının iyileştirilmesi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Öztemel, E., 2003, Yapay Sinir Ağlari, PapatyaYayincilik, Istanbul, p.

Rao, S. S., 2009, Engineering optimization: theory and practice, John Wiley & Sons, p.

Sağirlibaş, D., 2010, Image registration using artificial neural networks, *DEÜ Fen Bilimleri Enstitüsü*.

Sağıroğlu, Ş., Beşdok, E. ve Erler, M., 2003, Mühendislikte yapay zeka uygulamaları-1: Yapay sinir ağları, Ufuk Kitap Kırtasiye-Yayıncılık, p.

Sakthivel, S., Natarajan, R. ve Gurusamy, P., 2013, Application of bat optimization algorithm for economic load dispatch considering valve point effects, *International Journal of Computer Applications*, 67 (11).

Schnitzler, H.-U. ve Kalko, E. K., 2001, Echolocation by Insect-Eating Bats: We define four distinct functional groups of bats and find differences in signal structure that correlate with the typical echolocation tasks faced by each group, *Bioscience*, 51 (7), 557-569.

Sulaiman, S. I., Othman, Z., Musirin, I. ve Abidin, N. S. M. Z., 2015, Optimization of an Artificial Neural Network using Firefly Algorithm for modeling AC power from a photovoltaic system, *SAI Intelligent Systems Conference (IntelliSys), 2015*, 591-594.

Uzlu, E., Akpınar, A., Özturk, H. T., Nacar, S. ve Kankal, M., 2014, Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for Turkey, *Energy*, 69, 638-647.

Verma, B. ve Ghosh, R., 2002, A novel evolutionary neural learning algorithm, *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, 1884-1889.

Wang, G. ve Guo, L., 2013, A novel hybrid bat algorithm with harmony search for global numerical optimization, *Journal of Applied Mathematics*, 2013.

Whitley, D., Starkweather, T. ve Bogart, C., 1990, Genetic algorithms and neural networks: Optimizing connections and connectivity, *Parallel computing*, 14 (3), 347-361.

Yaghini, M., Khoshraftar, M. M. ve Fallahi, M., 2013, A hybrid algorithm for artificial neural network training, *Engineering Applications of Artificial Intelligence*, 26 (1), 293-301.

Yalçın, N., 2012, Sezgisel algoritma öğrenmeli yapay sinir ağları ile epilepsi hastalığının teşhisi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Yang, X.-S., 2010a, A new metaheuristic bat-inspired algorithm, In: Nature inspired cooperative strategies for optimization (NICSO 2010), Eds: Springer, p. 65-74.

Yang, X. S., 2010b, A New Metaheuristic Bat-Inspired Algorithm, *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization*, 284, 65-74.

Yang, X. S., 2011, Bat algorithm for multi-objective optimisation, *International Journal of Bio-Inspired Computation*, 3 (5), 267-274.

Yang, X. S. ve He, X. S., 2013, Bat algorithm: literature review and applications, *International Journal of Bio-Inspired Computation*, 5 (3), 141-149.

Yılmaz, S., 2014, Yarasa algoritmasının unimodal, multimodal ve kaydırılmış sayısal optimizasyon problemleri (cec05) üzerinde geliştirilmesi, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü*.

Yurdakul, E. M., 2014, Türkiye'de ithalatın gelişimi ve ithalatın yapay sinir ağları yöntemi ile tahmin edilebilirliğine yönelik bir analiz, *Adnan Menderes Üniversitesi Sosyal Biimler Enstitüsü*, 243.

Zhang, J.-R., Zhang, J., Lok, T.-M. ve Lyu, M. R., 2007, A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training, *Applied Mathematics and Computation*, 185 (2), 1026-1037.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	Lubna Luay Kamal KAMAL
Uyruğu	Irak
Doğum Yeri ve	Kerkük
Tarihi	27-04-1993
Telefon	05395001848
e-mail	lubnaluay9393@hotmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	**Kerkük Kız lisesi, Kerkük**	**2011**
Üniversite	:	**AL-Qalam Üniversitesi, Kerkük**	**2015**
Yüksek Lisans	:	**Selçuk Üniversitesi, Konya**	**---**
Doktora	:		

UZMANLIK ALANI:

YABANCI DİLLER: İngilizce, Türkçe, Arapça ve Fransızca

YAYINLAR

1- Lubna Luay Kamal, Halıfe Kodaz, "Training Artificial Neural Network By Bat Optimization Algorithms", International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-5, Issue-8, Aug.-2017.