

**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**



**COOT OPTİMİZASYON ALGORİTMASI TEMELLİ
YAPAY SİNİR AĞI MODELİ**

Yüksek Lisans Tezi

Ayşenur ÖZDEN

Danışman

Dr. Öğr. Üyesi İsmail İŞERİ

SAMSUN
2023

BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Yüksek Lisans tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet ☐

Hayır ☒

10 / 10 / 2023
Ayşenur ÖZDEN

TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

Tez Başlığı : COOT OPTİMİZASYON ALGORİTMASI TEMELLİ YAPAY SİNİR AĞI MODELİ

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 11 / 10 / 2023 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 19

Tek kaynak oranı : % 3 çıkmıştır.

11 / 10 / 2023
Dr. Öğr. Üyesi İsmail İŞERİ

ÖZET

COOT OPTİMİZASYON ALGORİTMASI TEMELLİ

YAPAY SİNİR AĞI MODELİ

Ayşenur ÖZDEN

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Ana Bilim Dalı

Yüksek Lisans, Ekim/2023

Danışman: Dr. Öğr. Üyesi İsmail İŞERİ

Yapay Sinir Ağları (YSA) biyolojik beyinde bulunan sinir hücrelerinin bilgiyi işleme ve iletme sistemlerini modelleyerek geliştirilmiş yapay zeka tekniğidir. Son yıllarda yapay sinir ağı modellerinde önemli ilerlemeler kaydedilmiş ve bu modeller örüntü tanıma, görüntü işleme ve sınıflandırma gibi çeşitli gerçek dünya problemlerine başarıyla uygulanmıştır. Bununla birlikte, yapay sinir ağlarının bir sınırlaması, eğitim aşamasında yerel minimumlara takılabilmesidir. Bu durum, gradyan inişine dayalı tekniklerin kullanılmasının bir sonucudur ve ağın genelleme performansını olumsuz bir şekilde etkilemektedir. Ağın performansının iyileştirilmesi ve oldukça zor bir problem olan yapay sinir ağlarının parametre optimizasyonu için literatürde çeşitli metasezgisel algoritmalar kullanılmıştır. Metasezgisel algoritmalar doğadan ilham alan ve birçok karmaşık optimizasyon probleminde başarıyla kullanılmakta olan algoritmalar.

Bu tez çalışmasında, yapay sinir ağı parametre optimizasyonunun sağlanması amacıyla COOT-ANN adı verilen yeni bir hibrit yapay sinir ağı modeli önerilmektedir. COOT-ANN modeli, yapay sinir ağı parametrelerini optimize etmek için yeni meta-sezgisel bir yaklaşım olan COOT optimizasyon algoritmasını kullanmaktadır. COOT optimizasyon algoritması coot kuşlarının yiyecek arayışında su üzerindeki farklı hareketleri modellenerek geliştirilmiş sürü zekası temelli bir metasezgisel algoritmadır. Metasezgisel tabanlı COOT optimizasyon algoritması kullanımı sayesinde COOT-ANN modeli, eğitim aşamasında yerel minimumlara takılmamaktadır. Çalışmada şarap, göğüs kanseri, iris cam veri setleri ve video sınıflandırma için spor veri seti üzerinde sınıflandırma işlemi yapılmış ve önerilen model ile gradyan inişi yöntemlerinin performansı karşılaştırılmıştır. Çalışmanın sonuçları, önerilen yöntemin gradyan inişi, ölçeklenmiş eşlenik gradyan ve Levenberg-Marquardt optimizasyon teknikleriyle karşılaştırıldığında test veri setleri üzerinde elde edilen doğruluk, çapraz entropi, F1 puanı ve Cohen's Kappa metrikleri açısından oldukça başarılı olduğunu göstermektedir. Tez çalışması, sınıflandırma problemleri için yapay sinir ağı eğitiminde COOT optimizasyon algoritmasının kullanımının performansı artırdığını ortaya koymaktadır.

Anahtar Sözcükler: Metasezgisel, Yapay sinir ağları, COOT optimizasyon algoritması, Sınıflandırma

ABSTRACT

ARTIFICIAL NEURAL NETWORK MODEL BASED ON COOT OPTIMIZATION ALGORITHM

Ayşenur ÖZDEN

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computer Engineering

Master, October/2023

Supervisor: Assist. Prof. Dr. İsmail İŞERİ

Artificial Neural Networks (ANNs) are an artificial intelligence technique that models the information processing and transmission systems of biological neurons in the brain. In recent years, significant progress has been made in artificial neural network models, and these models have been successfully applied to various real-world problems such as pattern recognition, image processing, and classification. However, a limitation of artificial neural networks is that they can get stuck in local minima during the training phase. This is a consequence of using gradient descent-based techniques and negatively affects the generalization performance of the network. Various metaheuristic algorithms have been used in the literature to improve the performance of the network and address the challenging problem of parameter optimization in artificial neural networks. Metaheuristic algorithms are inspired by nature and have been successfully used in many complex optimization problems.

In this thesis, a new hybrid artificial neural network model called COOT-ANN is proposed to achieve parameter optimization of artificial neural networks. The COOT-ANN model utilizes a novel metaheuristic approach called the COOT optimization algorithm to optimize the parameters of the artificial neural network. The COOT optimization algorithm is a swarm intelligence-based metaheuristic algorithm developed by modeling the different movements of coots, waterbirds, in their food foraging on the water. By employing the metaheuristic-based COOT optimization algorithm, the COOT-ANN model avoids getting stuck in local minima during the training phase. In the study, classification processes were performed on wine, breast cancer, iris flower data sets, and sports data set for video classification, and the performance of the proposed model was compared with gradient descent methods. The results of the study demonstrate that the proposed approach achieves high accuracy, cross-entropy, F1 score, and Cohen's Kappa metrics on test datasets compared to Gradient Descent, Scaled Conjugate Gradient, and Levenberg-Marquardt optimization techniques. The thesis work highlights the improved performance of using the COOT optimization algorithm in ANN training for classification problems.

Keywords: Metaheuristic, Neural network, COOT optimization algorithm, Classification

ÖN SÖZ VE TEŞEKKÜR

Yüksek lisans eğitimim süresince bilgilerini ve desteklerini esirgemeyen, çalışmanın hazırlanması aşamasında değerli zamanını ayırarak bana yol gösteren değerli danışmanım Dr. Öğr. Üyesi İsmail İŞERİ'ye, fikirleri ve yardımlarıyla çalışmama katkı sağlayan değerli hocam Dr. Öğr. Üyesi Gökhan KAYHAN'A teşekkürü borç bilirim.

Tez sürecim ve eğitim hayatım boyunca maddi ve manevi desteklerini esirgemeyen başta annem Sevim ÖZDEN olmak üzere sevgili aileme ve değerli arkadaşlarıma sonsuz teşekkür ederim. Son olarak bu tezi bugünlerimi görmesini çok istediğim fakat uzun zaman önce kaybetmiş olduğum babam Necmettin ÖZDEN'e atfediyorum.

Ayşenur ÖZDEN

İÇİNDEKİLER

TEZ KABUL VE ONAYI	i
BİLİMSEL ETİĞE UYGUNLUK BEYANI	ii
TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI	ii
ÖZET	iii
ABSTRACT	iv
ÖNSÖZ VE TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
SİMGELER VE KISALTMALAR	viii
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	xii
1. GİRİŞ.....	13
1.1. Literatür Özeti	15
2. MATERYAL VE YÖNTEM.....	22
2.1. Yapay Sinir Ağları.....	22
2.1.1. Biyolojik Sinir Ağları	23
2.1.2. Yapay Sinir Hücresi.....	24
2.1.3. YSA'nın Yapısı	27
2.1.4. YSA'nın Sınıflandırılması.....	28
2.1.5. Derin Sinir Ağları	33
2.2. YSA Eğitiminde Kullanılan Optimizasyon Algoritmaları	45
2.2.1. Gradyan İniş (Gradient Descent) Öğrenme Algoritması.....	45
2.2.2. Scaled Conjugate Gradient (SCG) Algoritması.....	47
2.2.3. Levenberg–Marquardt (LM) Algoritması	49
2.3. Metasezgisel Optimizasyon Algoritmaları.....	52
2.3.1. Evrim Temelli Algoritmalar	54
2.3.2. Sürü Zekası Temelli Algoritmalar	54
2.3.3. Fizik Temelli Algoritmalar.....	55
2.3.4. İnsan Davranışlarıyla İlgili Algoritmalar	55
2.4. COOT Optimizasyon Algoritması.....	56
2.4.1. Matematiksel Model ve Algoritma	56
2.4.2. Algoritma Akış Diyagramı ve Sözde Kodu.....	59
2.5. YSA Eğitiminde COOT Optimizasyon Algoritmasının Kullanımı	60
2.5.1. Önerilen Model (COOT-ANN)	61
2.6. Video Sınıflandırma	62
2.6.1. COOT-ANN Modelinin Video Sınıflandırmada Kullanımı	63
2.7. Performans Değerlendirme Ölçütleri.....	64
2.7.1. Doğruluk Oranı.....	65
2.7.2. Çapraz Entropi.....	65
2.7.3. F1 Skoru	65
2.7.4. ROC-AUC Eğrisi.....	66
2.7.5. Cohen's Kappa Metriği	67
3. UYGULAMA VE BULGULAR.....	69
3.1. Veri Kümeleri.....	69
3.1.1. Şarap Veri Kümesi.....	69
3.1.2. Göğüs Kanseri Veri Kümesi.....	70
3.1.3. İris Veri Kümesi	71
3.1.4. Cam Veri Kümesi	71
3.1.5. Video Sınıflandırma İçin Spor Veri Kümesi	72
3.2. Çalışma Ortamının Hazırlanması	72
3.3. Deneysel Sonuçlar	73
3.3.1. Şarap, Kanser, İris ve Cam Veri Setleri ile Elde Edilen Sonuçlar	73
3.3.2. Video Sınıflandırma Veri Seti ile Elde Edilen Sonuçlar	83
4. SONUÇLAR VE ÖNERİLER.....	87

4.1. Sonular	87
4.2. neriler	87



SİMGELER VE KISALTMALAR

AAA	: Yapay Alg Algoritması
ABC	: Yapay Arı Kolonisi Algoritması
ACO	: Karınca Kolonisi Optimizasyonu
ALO	: Karınca Aslanı Optimizasyon
ART	: Adaptif Rezonans Teori
AUC	: Eğrinin Altındaki Alan
BBO	: Biyocoğrafya Tabanlı Optimizasyon
BMO	: Kuş Çiftleşme Optimizasyonu
BOA	: Kelebek Optimizasyon Algoritması
BP	: Geri Yayılım
CE	: Çapraz Entropi
CK	: Cohen's Kappa
CNN	: Evrimsel Sinir Ağları
CSA	: Guguk Kuşu Arama Algoritması
DA	: Yusufçuk Algoritması
DNN	: Derin Sinir Ağları
ES	: Evrimsel Stratejiler
FOA	: Ateşböceği Optimizasyon Algoritması
GA	: Genetik Algoritma
GD	: Gradyan İniş
GOA	: Çekirge Optimizasyon Algoritması
GSA	: Yerçekimi Arama Algoritması
GWO	: Gri Kurt Optimizasyonu
LM	: Levenberg-Marquardt
LSTM	: Uzun Kısa Süreli Bellek Ağları
LVQ	: Öğrenen Vektör Kuantizasyonu
MLP	: Çok Katmanlı Algılayıcı
MOA	: Manyetik Optimizasyon Algoritması
MSE	: Ortalama Karekök Hatası
NFL	: Ücretsiz Öğle Yemeği Yok Teoremi
PBIL	: Olasılık Temelli Artımlı Öğrenme
PCA	: Temel Bileşenler Analizi
PSO	: Parçacık Sürü Optimizasyonu

ReLU	: Doğrultulmuş Doğrusal Birim
RNN	: Yinelemeli Sinir Ağları
ROC	: Alıcı Çalışma Karakteristikleri
SCG	: Ölçekli Eşlenik Gradyan
SMS	: Maddenin Durumu Araması
SSA	: Salp Sürü Algoritması
WOA	: Balina Optimizasyon Algoritması
YSA	: Yapay Sinir Ağı



ŞEKİLLER DİZİNİ

Şekil 2.1. Üç katmanlı YSA modeli.....	23
Şekil 2.2. Biyolojik sinir sistemi diyagramı	23
Şekil 2.3. Sinir hücresi yapısı (Oztürk, 2011).....	24
Şekil 2.4. Yapay sinir hücresi yapısı	25
Şekil 2.5. Aktivasyon fonksiyonları (Yalçın, 2012).....	27
Şekil 2.6. İleri beslemeli YSA modeli.....	29
Şekil 2.7. Geri beslemeli YSA modeli (Gallo, 2015).....	30
Şekil 2.8. Tek katmanlı YSA modeli	31
Şekil 2.9. Çok katmanlı YSA mimarisi örneği	32
Şekil 2.10. RNN döngü şeması.....	34
Şekil 2.11. Tek katmanlı standart RNN modeli	35
Şekil 2.12. Dört etkileşim katmanlı LSTM modeli.....	35
Şekil 2.13. Unutma kapısı katmanı	36
Şekil 2.14. Giriş kapısı ve yeni aday değer vektörü.....	36
Şekil 2.15. Yeni hücre durumu	37
Şekil 2.16. Çıkış kapısı ve yeni bilgi.....	37
Şekil 2.17. CNN katmanlarının blok diyagramı	38
Şekil 2.18. Evrişim işleminden özellik haritası çıkarma örneği.....	39
Şekil 2.19. Maksimum havuzlama örneği	40
Şekil 2.20. Örnek bir 2D evrişimli sinir ağı mimarisi	41
Şekil 2.21. Artık öğrenmenin yapı taşı (He vd., 2016).....	41
Şekil 2.22. ResNet50 mimarisi	43
Şekil 2.23. VGG-16 mimarisi	44
Şekil 2.24. YSA eğitimi temel akış diyagramı	45
Şekil 2.25. GD akış diyagramı.....	46
Şekil 2.26. GD algoritması sözde kodu	47
Şekil 2.27. SCG algoritması akış diyagramı.....	48

Şekil 2.28. SCG algoritması sözde kodu	49
Şekil 2.29. LM algoritması akış diyagramı	50
Şekil 2.30. LM algoritması sözde kodu	52
Şekil 2.31. Metasezgisel algoritmaların kategorik bazda sınıflandırılması	54
Şekil 2.32. COOT optimizasyon algoritması sözde kodu	60
Şekil 2.33. COOT optimizasyon algoritması akış diyagramı	60
Şekil 2.34. Önerilen modelin sözde kodu	61
Şekil 2.35. Önerilen modelin akış diyagramı	62
Şekil 2.36. COOT-ANN modeli ile video sınıflandırma işlemi	64
Şekil 2.37. ROC eğrisi ve AUC	67
Şekil 3.1. Şarap veri kümesinin test verisi için model bazlı karışıklık matrisi	76
Şekil 3.2. Şarap veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri	77
Şekil 3.3. Göğüs kanseri veri kümesinin test verisi için model bazlı karışıklık matrisi	78
Şekil 3.4. Göğüs kanseri veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri	79
Şekil 3.5. Cam veri kümesinin test verisi için model bazlı karışıklık matrisi	80
Şekil 3.6. Cam veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri	81
Şekil 3.7. İris veri kümesinin test verisi için model bazlı karışıklık matrisi	82
Şekil 3.8. İris veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri	83
Şekil 3.9. Spor veri kümesinin test verisi için model bazlı karmaşıklık matrisi	85
Şekil 3.10. Spor veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri	86

TABLÖLAR DİZİNİ

Tablo 2.1. Biyolojik sinir sistemi ve yapay sinir sistemi elemanlarının karşılaştırılması	25
Tablo 2.2. İki sınıflı karışıklık matrisi.....	64
Tablo 2.3. İki puanlayıcının k-kategori ölçeğindeki değerlendirmelerinin ortak olasılıkları .	68
Tablo 3.1. Sınıflandırma için kullanılan veri kümelerinin özellikleri	69
Tablo 3.2. Şarap veri kümesinin nitelikleri ve aralık değerleri.....	69
Tablo 3.3. Göğüs kanseri veri kümesinin nitelikleri ve aralık değerleri	70
Tablo 3.4. İris veri kümesinin nitelikleri ve aralık değerleri	71
Tablo 3.5. Cam veri kümesinin nitelikleri ve aralık değerleri	71
Tablo 3.6. Veri kümelerinin ortalama performansları (5 kat).....	74
Tablo 3.7. Veri kümelerinin örnek kat performansı.....	75
Tablo 3.8. Test verileri üzerinde elde edilen model performansları	84

1. GİRİŞ

Yapay zekanın bir alt dalı olan makine öğrenimi mevcut veriler ile öğrenme sonucunda yeni veri üzerinde çıkarımlar yapılmasını sağlayan bilgisayar bilimidir (Abdolrasol vd., 2021). Günlük hayatta sosyal medya uygulamaları, online alışveriş, sağlık, eğitim ve finans sektörü gibi alanlarda makine öğrenimi uygulamalarına sıklıkla rastlanmaktadır. Makine öğreniminde en fazla çalışma yapılan problemler sınıflandırma, regresyon ve kümeleme problemleridir. Problemlerde farklı makine öğrenme yöntemleri kullanılmakta olup, yapay sinir ağları hem sınıflandırma, hem regresyon hem de kümeleme problemlerine kolay şekilde uygulanabilirliği açısından önemli yöntemlerden birisidir.

Yapay Sinir Ağları (YSA) ilk kez 1943'te McCulloch ve Pitts (McCulloch ve Pitts, 1943) tarafından önerilmiş ve hala üzerine çalışmalar yapılan önemli bir makine öğrenme alanıdır. YSA'lar beyindeki nöronlara benzeyen ve bilgi işleme için kullanılan yapay nöron adı verilen küçük birimlerle çalışır. Bu birimlere düğüm de denir. YSA'da düğümler ağırlık değerleri ile birbirine bağlıdır ve her katman arasındaki düğümler beyindeki sinapsları taklit ederek bilgi aktarımını sağlar. Çıktı, YSA'nın girdi ve gizli katmanları arasındaki özel matematiksel hesaplamalarla üretilir. Sinir ağı öğrenme algoritması, YSA tarafından üretilen çıktı ile hedef çıktı arasındaki hataya bağlı bir maliyet fonksiyonunu en aza indirerek düğüm ağırlıklarını ve bias değerlerini ayarlar veya değiştirir. YSA'nın daha doğru sonuçlar üretebilmesi ve başarılı sınıflandırmalar yapabilmesi için bias değeri ve ağırlık değerlerinin maliyet fonksiyonu minimizasyonuna göre güncellenmesi gerekmektedir.

Öğrenme algoritmaları temel olarak matematiksel/geleneksel ve stokastik olarak ikiye ayrılır: Geri yayılım ve Levenberg-Marquardt (LM) gibi klasik algoritmalar matematiksel/geleneksel algoritmalar olarak adlandırılırken, sürü zekası algoritmaları ve genetik algoritmalar (GA) stokastik öğrenme algoritmaları olarak adlandırılır (Si vd., 2022). YSA'lar genellikle GD tekniklerine dayalı geriye yayılım algoritmaları ile eğitilirler. Bu yöntemler büyük ölçüde başlangıç değerlerine bağlıdır, ancak yerel optimumda kolayca takılıp kalırlar. Bu nedenle, YSA'ların eğitiminde geleneksel yöntemlerin yerine metasezgisel yöntemlerin kullanımı önemli bir araştırma konusu olmuştur.

Metasezgisel algoritmalar, doğal optimizasyon algoritmaları olarak da bilinir

ve karmaşık optimizasyon problemlerini çözmek için doğadan ilham alırlar. Genel amaçlı metasezgisel yöntemler sekiz farklı kategoriye göre değerlendirilmektedir. Son yıllarda çeşitli mühendislik optimizasyon problemlerinde metasezgisel algoritmalar başarılı bir şekilde kullanılmış ve karmaşık gerçek dünya optimizasyon problemlerinde geleneksel yöntemlere göre daha başarılı çözümler sağlamıştır. Metasezgisel yöntemler, problemin doğasından bağımsızdır, yani stokastik bir yaklaşım kullandıkları için problemin türev bilgisine ihtiyaç duymazlar (Faramarzi vd., 2020). Hayvanların hareketlerini ya da avlanma sırasındaki davranışlarından ilham alınarak geliştirilmiş olan sürü temelli algoritmalar metasezgisel algoritmaların bir kategorisidir (Abualigah vd., 2021). Sürü temelli metasezgisel algoritmalar Gri Kurt Optimizasyonu (Gray Wolf Optimization) (Mirjalili vd., 2014), Ateşböceği Optimizasyon Algoritması (Firefly Optimization Algorithm) (Yang, 2010), Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm) (Karaboğa ve Akay, 2009), Yarasa Algoritması (Bat Algorithm) (Yang ve Hossein Gandomi, 2012) ve Yusufçuk Algoritması (Dragonfly Algorithm) (Mirjalili, 2016) gibi temelinde sürü davranışını modelleyen algoritmalarlardır.

Metasezgisel algoritmalarda keşif ve sömürü olarak adlandırılan iki özellik vardır. Keşif mevcut alanı küresel olarak aramadır ve yerel optimumdan kaçmayı sağlar. Sömürü ise kalitelerini artırmayı hedefleyen çözümler etrafında yerel olarak arama yapmaktadır. Bu iki özellik arasında denge sağlanması algoritmanın iyi performans göstermesini sağlar (Faramarzi vd., 2020). Literatür çalışmaları incelendiğinde YSA eğitiminde kullanılmış olan optimizasyon algoritmalarının yerel minimumlara takılma sorununun kesin çözüme ulaştırılamadığı ve farklı gerçek dünya optimizasyon problemleri için farklı algoritmalar kullanılması gerektiği belirtilmiştir.

Bu tez çalışmasında YSA eğitimi için ilk defa yeni bir metasezgisel yöntem olan COOT optimizasyon algoritması kullanılmıştır. COOT algoritması (Naruei ve Keynia, 2021), su yüzeyindeki bir saka kuşu sürüsünün davranışından esinlenmiştir. Cootların su yüzeyindeki davranışları 3 hareketten oluşur. İlk aşamada cootlar düzensiz hareket eder. İkinci aşamada düzenli hareket ederler. Sürü, bir besin kaynağına ulaşmak için lidere doğru hareket eder ve bu hareket bir zincir şeklindedir (Hussien vd., 2022; Memarzadeh ve Keynia, 2021). Çalışma, yapay sinir ağı eğitiminde gradyan iniş algoritması yerine COOT algoritmasını kullanarak ağırlık ve

bias değerlerini optimize etmeyi amaçlamaktadır (Özden ve İşeri, 2023). Deneylerde YSA modelinin sınıflandırma problemlerindeki performansı beş farklı veri seti üzerinde test edilmiştir. Tez çalışmasının başlıca katkıları aşağıdaki gibidir:

- Metasezgisel COOT optimizasyon algoritmasına dayalı yeni bir hibrit YSA modeli (COOT-ANN) önerilmiştir.
- Önerilen COOT-ANN modeli, yerel minimumlara takılıp kalmamak, verimli ve sağlam çözümler üretmek açısından geleneksel YSA modelinden üstündür.
- YSA modelinin optimum parametrelerinin belirlenmesi açısından COOT metasezgisel algoritmanın yenilikçi bir kullanımıdır.
- Sınıflandırma problemleri için yüksek performanslı bir model önerilmiştir.

Birinci bölümde teze giriş yapıldıktan sonra aynı bölümün devamında tezle ilgili literatür araştırması verilmiştir. İkinci bölümde tezde kullanılan materyal ve yöntem hakkında bilgi verilmiştir. Bölümde YSA, YSA eğitiminde kullanılan algoritmalar, metasezgisel algoritmalar, COOT optimizasyon algoritması, tezde önerilen yöntem olan COOT algoritması ile YSA eğitimi, video sınıflandırma ve performans değerlendirme ölçütleri hakkında detaylı bilgi verilmiştir. Üçüncü bölümde kullanılan veri kümeleri, çalışma ortamının hazırlanması, önerilen yöntemin parametre analizleri ve sınıflandırma başarısının farklı yöntemler ile karşılaştırıldığı deneysel sonuçlar verilmiştir. Son olarak dördüncü bölümde tezin sonuçları ve öneriler verilmiştir.

1.1. Literatür Özeti

Literatürde YSA eğitimi için kullanılan çeşitli metasezgisel algoritmalar vardır. Yapılan çalışmalardan bazıları bu bölümde incelenmektedir.

Blum ve Socha (2005), örüntü sınıflandırma için ACO algoritmasının bir varyantı ile YSA ağırlıklarını optimize etmişlerdir. Kanseri, diyabet ve kalp veri setlerinin kullanıldığı deneysel sonuçlarda BP, LM ve GA algoritmaları ile önerilen model karşılaştırılmış ve önerilen modelin daha iyi performans gösterdiği belirlenmiştir.

Meissner vd. (2006), küçük organik moleküllerin kan-beyin bariyeri

geçirgenlik katsayılarının tahmini için optimize edilmiş parçacık sürü optimizasyonu (OPSO) algoritmasını önermiş ve önerilen metot YSA eğitiminde kullanılarak beş yapay uygunluk fonksiyonu üzerinde sonuçlar değerlendirilmiştir. OPSO algoritmasının eğitim süresinin diğer PSO yöntemlerine oranla azaldığı belirlenmiştir.

Carvalho ve Ludermir (2007), çalışmalarında ileri beslemeli YSA mimarisi ve ağırlık optimizasyonu için PSO algoritmasını kullanmışlardır. Geliştirdikleri modeli kanser, diyabet ve kalp veri kümelerinde test etmişler ve deneysel sonuçlar ile PSO algoritmasının ileri beslemeli YSA mimari ve ağırlık parametre değerlerinin optimizasyonunda üstün başarı gösterdiğini ispatlamışlardır.

Karaboğa vd. (2007), yapay sinir ağı parametrelerinin optimizasyonunda ABC algoritmasını kullanarak farklı veri setleri üzerinde yaptıkları örüntü sınıflandırma çalışmasında elde ettikleri sonuçları genetik algoritma ve geri yayılım yöntemi ile karşılaştırmış ve önerilen yöntemin üstünlüğünü ispatlamışlardır. Ayrıca Ghaleini vd. (2019), istinat duvarlarının güvenlik faktörünün belirlenebilmesi ve Shah vd. (2011), deprem zaman serisi veri tahmini için ABC algoritmasını YSA parametrelerinin optimizasyonunda kullanarak önerdikleri yöntemin başarısını kanıtlamışlardır.

Zamani ve Sadeghian (2010), çalışmalarında YSA eğitimi için PSO algoritmasını kullanmış ve geliştirdikleri modeli farklı sınıflandırma problemleri için farklı veri setleri üzerinde test etmişlerdir. Modelin performans değerlendirmesinde PSO algoritmasının YSA parametre optimizasyonunda sınıflandırma problemleri için başarılı sonuçlar ürettiği gözlemlenmiştir.

Kulluk vd. (2012), sınıflandırma görevleri için yaygın olarak kullanılan ileri beslemeli Yapay Sinir Ağlarının denetimli eğitimi için Harmony arama algoritmasını uygulamışlardır. Uyum arama algoritmalarının beş farklı varyasyonu, kendine uyarlamalı küresel en uygun arama algoritmasına özel olarak odaklanılarak değerlendirilmiştir. Önerilen yöntemin etkinliği, altı sınıflı bir sınıflandırma problemi ve gerçek dünya problemi üzerindeki ampirik testlerle belirlenmiştir.

Askarzadeh ve Rezazadeh (2013), BMO algoritmasını ileri beslemeli yapay sinir ağı optimizasyonunda kullanmış ve çalışma sonucunda Iris, Wisconsin breast cancer ve Pima Indian diabetes veri setleri üzerinde yapılan sınıflandırma çalışmalarında en iyi sınıflandırıcı olmasa da önerilen yöntemin karşılaştırılan diğer

sınıflandırıcılar ile rekabet edebilir nitelikte olduğunu belirlemişlerdir.

Ahadzadeh ve Menhaj (2014), ileri beslemeli YSA parametre optimizasyonunda Değiştirilmiş Diferansiyel Evrim (DDE) algoritmasını kullanmış ve önerdikleri modelin sınıflandırma problemlerindeki performansını incelemişlerdir. Çalışmada UCI veri ambarından aldıkları sekiz veri kümesi üzerinde yapılan deneylerde DDE ile geliştirilen modelin Geri Yayılım algoritması, Global Harmoni Arama, Adaptif Harmoni Arama, Genetik Algoritma-Geri Yayılım ve Yapay Arı Koloni Algoritması-Geri Yayılım modellerine göre yüksek performans gösterdiği belirtilmiştir.

Mirjalili vd. (2014), MLP eğitiminde öğrenme algoritması olarak BBO kullanmış ve (balon, iris, kalp vb) veri setleri üzerinde yapılan çalışmalarda BBO'nun PSO, GA, ACO, ES ve PBIL'e göre yerel minimumlara daha az takıldığını ispatlamışlardır.

Pereira vd. (2014), sosyal örümcek algoritmasını YSA eğitiminde kullanmışlar ve geliştirmiş oldukları modeli sağlık verilerinin sınıflandırılması problemine uygulamışlardır. Önerilen model yüksek performans gösterememiş olsa da çoğu bilinen algoritmalar ile rekabetçi sonuçlar göstermiştir.

Mandal vd. (2015), geri yayımlı sinir ağı eğitiminde hızlı yakınsama ve eğitim hatasını en aza indirmek için Ateş Böceği algoritmasını kullanmış, yüksek eğitim verisi sayısı, popülasyon ve yinleme ile yüksek doğruluk değerleri elde etmişlerdir.

Mirjalili (2015), MLP eğitiminde ilk kez GWO algoritmasını kullanmış, beş sınıflandırma ve üç fonksiyon yaklaşım veri seti ile yaptığı deneylerde PSO, GA, ACO, ES ve PBIL algoritmalarına göre yüksek düzeyde doğruluk değeri elde etmiştir.

Yamany vd. (2015), MLP eğitiminde ALO algoritmasını kullanmış ve xor, iris, kalp ve göğüs kanseri veri setleri üzerinde deneysel çalışmalar yapmıştır. Önerilen modelin performansı PSO, ACO ve GA yöntemleri ile karşılaştırılmış ve modelin yüksek doğruluk oranı ile üstünlüğü ispatlanmıştır.

Dutta vd. (2016), 5 tıbbi veri seti üzerinde yapılan sınıflandırma işlemi için MLP eğitiminde Fireworks Algoritmasını uygulamışlardır. LM yöntemi ve PSO ile karşılaştırma yaptıkları çalışmada önerilen yöntemin sınıflandırmada daha yüksek

oranla performans gösterdiği belirlenmiştir.

Faris vd. (2016), MLP eğitiminde Multi-Verse optimizasyon algoritmasını kullanarak dokuz farklı veri setinde modelin performansını değerlendirmişler ve GA, PSO, DE, BP, LM, FOA ve CSA algoritmaları ile yapılan karşılaştırmalarda önerdikleri modelin başarısını ispatlamışlardır.

Alwaisi ve Baykan (2017), sınıflandırma için YSA eğitiminde Shuffled Frog Leaping Algorithm (Karışık Kurbağa Atlama Algoritması) kullanmış ve kategorik, sayısal ve karma özelliklere sahip on veri seti ile yaptıkları deneylerde önerilen yöntemin sınıflandırma doğruluğunun geri yayımlı sınıflandırma algoritmasından daha yüksek olduğunu belirlemişlerdir.

Kankal ve Uzlu (2017), öğretme-öğrenme tabanlı optimizasyon algoritmasını YSA parametrelerinin optimizasyonunda kullanarak Türkiye’de elektrik enerjisi talebini modellemeyi amaçlamışlardır. Çalışmalarında önermiş oldukları modeli geri yayılım ve ABC algoritması ile karşılaştırarak elektrik enerjisi talebi tahmininde önerdikleri modelin üstünlüğünü ispatlamışlardır.

Zhang vd. (2017), çalışmalarında YSA eğitimi için toplu karar optimizasyon algoritması (CDOA) kullanarak model geliştirmişlerdir. İnsanın karar verme mekanizması temel alınarak geliştirilmiş olan metasezgisel yöntem YSA eğitiminde diğer modellere göre üstün başarı göstermiştir.

Abusnaina vd. (2018), SSA’yı ilk kez YSA eğitiminde kullanmış ve örüntü sınıflandırma için Haberman, Wisconsin Breast Cancer, Diabetes, Thyroid and Liver veri setleri üzerinde yapılan deneysel çalışmalarda önerilen modelin diğer optimizasyon algoritmaları ile rekabet edebilecek seviyede olduğunu belirtmişlerdir. Ayrıca SSA'nın performansını iyileştirmek için Bairathi ve Gopalani (2021), daha fazla keşif gücüne sahip çoklu lider salp sürü algoritmasını (MLSSA) geliştirerek YSA ağırlık ve bias optimizasyonunda kullanmış, on üç farklı veri seti ile yapılan deneylerde önerilen modelin diğer modellerden daha yüksek doğruluk değerine sahip olduğu belirlenmiştir.

Aljarah vd. (2018) WOA’yı YSA ağırlık ve bias değerlerini güncellemek için kullanmışlardır. Farklı zorluk seviyelerinden 20 veri seti üzerinde gerçekleştirilen deneysel çalışmalarda önerilen modelin doğruluk ve yakınsama açısından etkinliği geriye yayılım algoritması ve altı evrimsel teknikle karşılaştırılarak ispatlanmıştır.

Yasen vd. (2018), hastalık tahmin modeli için DA'yı YSA parametre optimizasyonunda kullanmış, beş farklı gerçek tıbbi veri seti üzerinde yapılan deneylerde önerilen model ile dört veri setinde en yüksek F ölçütü ve AUC değeri, üç veri setinde en iyi kesinlik ve geri çağırma değeri elde etmişlerdir. Xu ve Yan (2019), DA algoritmasının arama performansının iyileştirilmesi için geleneksel DA algoritması ve geliştirilmiş Nelder-Mead algoritmasını kullanarak hibrit bir optimizasyon algoritması (INMDA) önermiş, üç farklı sınıflandırma problemi üzerinde yapılan deneylerde önerilen model ile en yüksek doğruluk değeri elde edilmiştir.

Heidari vd. (2019), MLP eğitiminde GOA kullanmış ve önerilen yöntem GOAMLP olarak adlandırılmıştır. Meme kanseri, parkinson, diyabet, koroner kalp hastalığı ve ortopedik hastalar veri setleri ile yapılan deneylerde GOAMLP ile sekiz algoritma karşılaştırılmış, önerilen modelin yüksek doğruluk ile karmaşık tıbbi sınıflandırma problemlerinde başarısı ispatlanmıştır.

Tran-Ngoc vd. (2019), yapılarda hasar tespiti için CSA kullanılarak YSA eğitimi gerçekleştirmiş ve deneysel sonuçlarda önerilen modelin PSO ve GA algoritmalarından üstünlüğünü ispatlamışlardır. Valian vd. (2011) sınıflandırma problemi için YSA eğitiminde parametrelerin sabit tutulduğu standart CSA algoritması yerine ayarlanabilen parametreye sahip Geliştirilmiş Cuckoo Search (ICS) algoritmasını kullanmış, iris ve göğüs kanseri veri setleri üzerinde yapılan deneylerde ICS'nin YSA optimizasyonunda CSA'ya göre üstünlüğünü belirtmişlerdir.

Gülcü (2020), MLP eğitiminde SMS algoritmasını kullanarak 5 farklı veri seti (xor, balon, iris, göğüs kanseri, kalp) üzerinde deneysel çalışmalar yapmıştır. SMS-MLP ile GWO-MLP, ACO-MLP, GA-MLP, PBIL-MLP, PSO-MLP ve ES-MLP algoritmalarını karşılaştırarak önerilen yöntemin YSA optimizasyon problemi için başarılı bir şekilde kullanılabileceğini ortaya koymuştur.

Heidari vd. (2020), ALO algoritmasını MLP eğitiminde kullanarak hibrit ALO-MLP modelini geliştirmişler ve geliştirdikleri modelin üstünlüğünü GA, PBIL, PSO ve DE algoritmalarıyla karşılaştırarak ispatlamışlardır.

Kumar ve Kumar (2020), çalışmalarında ağırlık ve bias değerlerinin optimizasyonunda yarasa optimizasyon algoritmasını kullanmış ve önerilen metot

tıbbi görüntü sınıflandırma probleminde test edilmiştir. Deneysel sonuçlarda doğruluk değerlerine bakıldığında Bat algoritmasının BP algoritmasından daha üstün olduğu ispatlanmıştır.

Türkoğlu ve Kaya (2020), optimum ağ parametrelerinin bulunmasına yönelik yaptıkları çalışmada YSA eğitimi için AAA ile model önermişlerdir. On sınıflandırma veri seti üzerinde yapılan deneysel çalışmalarda önerilen modelin performansı altı sürü tabanlı optimizasyon algoritması ve geri yayılım algoritması ile karşılaştırılmış ve AAA'nın YSA optimizasyonunda daha başarılı bir yöntem olduğu belirlenmiştir.

Irmak ve Gülcü (2021), kelebeklerin davranışları modellenerek tasarlanan BOA ilk kez çok katmanlı algılayıcıyı eğitmek için kullanmışlardır. Geliştirilen algoritma BOA-MLP olarak adlandırılmış ve burada BOA algoritması ile MLP'deki düğüm ağırlıkları ve bias değeri optimize edilmiştir. Xor ve balon veri setlerinde %100, iris veri setinde %97,18, göğüs kanseri veri setinde %99,37 ve kalp veri setinde %74,46 başarı elde etmiştir. Duyarlılık, özgüllük, kesinlik ve F1-skoru sonuçlarına göre BOA-MLP algoritmasının performansı tüm veri setleri için oldukça iyidir.

Si vd. (2022), YSA eğitiminde 11 farklı öğrenme algoritması denemiş ve 15 tıbbi veri seti üzerinde çalışmalar yapmışlardır. Sınıflandırma sonuçları, Çok Kriterli Karar Verme (MCDM) yöntemi kullanılarak analiz edilmiş ve Denge Optimize Edicinin (EO) ele alınan diğer tüm algoritmalarından daha iyi performans gösterdiğini belirlemişlerdir.

Literatürde hibrit algoritmalar kullanılarak YSA eğitimi ve parametre optimizasyonunun yapıldığı çalışmalar da mevcuttur. Bu çalışmalardan bazıları aşağıda verilmiştir.

Zanchettin vd. (2011), çalışmalarında tabu arama algoritması, tavlama benzetimi ve GA algoritmalarının hibridizasyonu ile geliştirmiş oldukları algoritmayı YSA eğitiminde kullanmışlardır. Deneysel sonuçlarda algoritmaların YSA eğitiminde ayrı olarak kullanıldıkları modeller ile hibrit algoritma ile geliştirilen modelin başarımı karşılaştırıldığında hibrit modelin başarımının yüksek olduğu belirlenmiştir.

Mirjalili vd. (2012), GSA ve PSO algoritmaları kullanılarak geliştirilmiş olan

hibrit algoritmayı YSA eğitiminde kullanarak parametre optimizasyonunu sağlamaya çalışmışlardır. Literatürdeki çalışmalarla karşılaştırılan sonuçlarda geliştirilmiş hibrit modelin yakınsama oranı ve yerel minimumlardan kaçınmada yüksek başarı gösterdiği görülmüştür.

Yaghini vd. (2013), PSO ve BP algoritmalarını kullanarak hibrit bir algoritma geliştirmiş ve geliştirdikleri algoritmayı YSA eğitiminde kullanmışlardır. Metasezgisel ve gradyan tabanlı algoritma yeteneklerini birleştirdikleri modelin etkinliğini ispatlamak için kanser, kart, diyabet, en, kalp, iris, at ve tiroit veri setleri üzerinde deneyler yapılmış ve modelin yüksek performans değeri ile diğer modellerden üstünlüğü ispatlanmıştır.

Chen vd. (2015), yaptıkları çalışmada PSO ve CSA algoritmalarını birleştirerek YSA eğitiminde kullanmışlardır. Geliştirmiş oldukları modeli fonksiyon tahmini ve sınıflandırma problemlerine uygulamışlar ve deneysel sonuçlarda model performansını CSA ve PSO algoritmalarının YSA eğitiminde tek kullanıldıkları modeller ile karşılaştırmışlardır. Hibrit algoritma ile geliştirilen model diğer iki modelden yüksek performans göstererek başarısı ispatlanmıştır.

Al Nuaimi ve Abdullah (2017), YSA eğitimi için PSO ve ABC algoritmalarını birleştirerek hibrit bir algoritma geliştirmişler ve deneysel sonuçlarda geliştirdikleri modelin diğer iki algoritma ile geliştirilen modellerden daha yüksek performans gösterdiği belirtilmiştir.

Ansari vd. (2020), iflas tahmini için iki metasezgisel algoritma, MOA ve PSO algoritmalarını yeni bir yöntemi önermek için hibridizasyon yoluyla geliştirilmişlerdir. MOA-PSO, YSA'nın iflas tahminindeki performansını iyileştirmek için YSA eğitiminde kullanılmış ve hibrit MOA-PSO'nun performansı, mevcut dört algoritmanın performansı ile karşılaştırılmıştır. Önerilen hibrit MOA-PSO algoritması, %99,7 doğrulukla daha hızlı ve daha doğru bir tahminle umut verici sonuçlar sergilemiştir.

Ghanem vd. (2020), ABC algoritmasını DA ile birleştiren yeni bir hibrit metasezgisel kullanarak MLP eğitime dayanan HADMLP-IDS modeli adı verilen bir saldırı tespit sistemi için yeni bir model tanıtmışlardır. Çalışmada, KDD Cup 99, NSL-KDD, UNSW-NB15 ve ISCX2012 veri setleriyle eğitilen HADMLP-IDS modeli yüksek başarı göstermiştir.

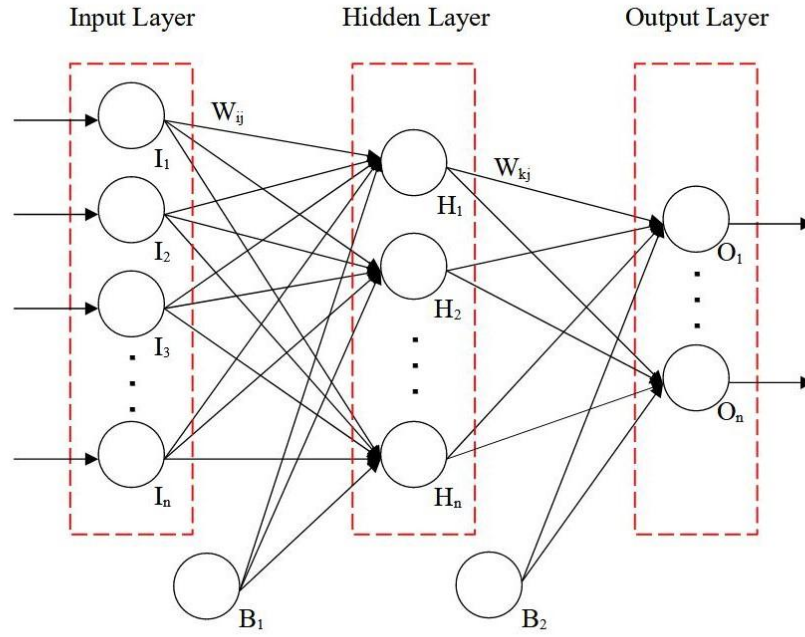
2. MATERYAL VE YÖNTEM

Bu bölümde tez çalışmasında kullanılan YSA (YSA'nın yapısı ve sınıflandırılması), YSA eğitiminde kullanılan optimizasyon algoritmaları, metasezgisel optimizasyon algoritmaları, COOT optimizasyon algoritması, YSA eğitiminde COOT algoritmasının kullanımı, video sınıflandırma ve performans değerlendirme ölçütleri hakkında ayrıntılı bilgi verilmiştir.

2.1. Yapay Sinir Ağları

Yapay sinir ağları, beyin hücrelerinin bilgi iletim sistemini modelleyerek ve biyolojik sistemlerde gerçekleşen öğrenme süreçlerinden ilham alarak geliştirilen en popüler makine öğrenme algoritmalarından biridir (Adamowski and Chan, 2011). YSA'lar, karmaşık, doğrusal olmayan ilişkileri modelleme yetenekleri aracılığıyla doğrusal olmayan optimizasyon problemlerinin çözümünü yaklaşık olarak hesaplamak için kullanılır (Pineda ve Serpa, 2021) ve sınıflandırma, desen tanıma, kümeleme gibi çeşitli makine öğrenme görevlerinde yaygın olarak kullanılır. Ayrıca, YSA'nın performansını iyileştirmek için hibrit sinir ağı modellerinin geliştirilmesi, bilim insanları için önemli bir araştırma konusudur (Kayhan ve İşeri, 2023).

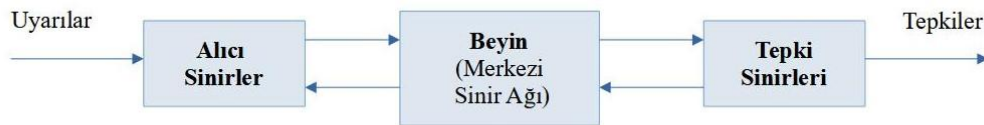
YSA'lar, nöronlar olarak adlandırılan birden çok işlem biriminden oluşur. Net çıktı elde edildiğinde, giriş verileri bağlantı ağırlıkları ile çarpılır ve ardından birleştirme işlevini kullanarak birleştirilir. Bu işlemle elde edilen çıktı değeri bir aktivasyon işlevi tarafından işlenir (Irmak ve Gülcü, 2021). En yaygın olarak kullanılan aktivasyon işlevleri hiperbolik tanjant, sigmoid, softmax ve ReLU şeklindedir. YSA'lar, giriş, gizli ve çıkış olmak üzere üç tür katmandan oluşur. Gizli katman bir veya daha fazla katmandan oluşabilir. İstenilen çıkış değeri, giriş verileri nöronların aktivasyon işleviyle işlenerek oluşturulur. Nöronlar arasındaki bağlantı, eğitim sonucunda ayarlanan uygun ağırlıklarla belirlenir (Sharma vd., 2022). Bilgi giriş katmanında işlenmez; alınan bilgi gizli katmana iletilir. Çıkış katmanında, gizli katmandan gelen bilgi işlenir ve sonuç verisi üretilir. Üç katmanlı bir yapay sinir ağının yapısı Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Üç katmanlı YSA modeli

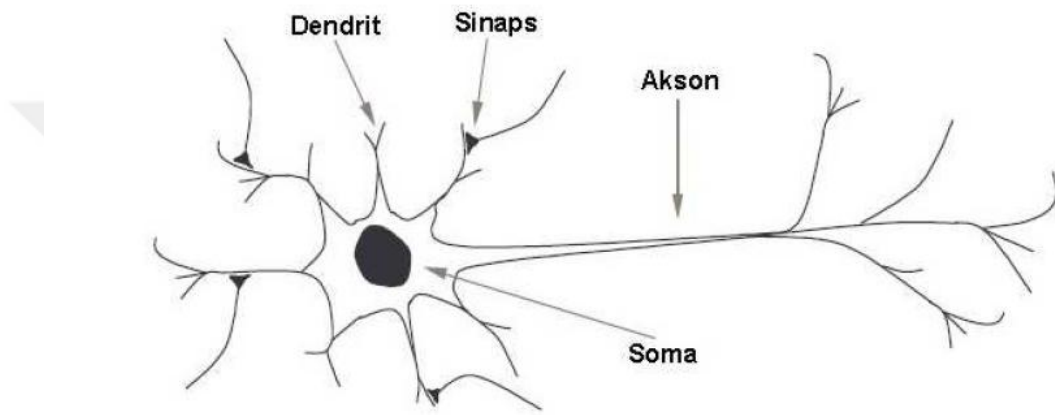
2.1.1. Biyolojik Sinir Ağları

YSA'lar biyolojik sinir ağlarından esinlenerek geliştirilmiş olan matematiksel modellerdir. Biyolojik sinir ağları insan beyninin çalışma mekanizması için en önemli yapıdır. Biyolojik sinir ağları, sinir hücrelerinden meydana gelir. İnsan beyninde yaklaşık olarak on milyar sinir hücresi olduğu ve bu hücrelerin arasında altı trilyonu geçen bağlantı olduğu düşünülmektedir (Haykin, 2009). Biyolojik sinir ağları duyu organlarından gelen sinyali (bilgiyi) beyne taşır. Aynı şekilde beynin bu bilgiler sonucunda ürettiği tepkiyi ya da kararı yine biyolojik sinir sistemi aracılığıyla organlara eylem olarak gönderir. Biyolojik sinir sisteminin diyagramı Şekil 2.2'de verilmiştir.



Şekil 2.2. Biyolojik sinir sistemi diyagramı

Biyolojik sinir hücreleri dendrit, akson, hücre gövdesi (soma) ve sinapslar olmak üzere dört elemandan oluşmaktadır. Dendritler diğer sinir hücrelerinden bilgi toplamak ile görevli olan ağaç dalları benzerindeki yapıdır. Dendritler sayesinde toplanılan bilgiler hücre gövdesine iletilerek değerlendirilir. Değerlendirilen bilgiler hücrelere aktararak bilgi alışverişini sağlamak için aksona iletilir. Akson hücre gövdesinden gönderilen bilgiyi işleyerek nöronlara iletir. Sinaps ise akson ve diğer nöron dendritleri arasındaki bağlantıyı sağlar. Biyolojik sinir hücresinin yapısı Şekil 2.3'te verilmiştir.

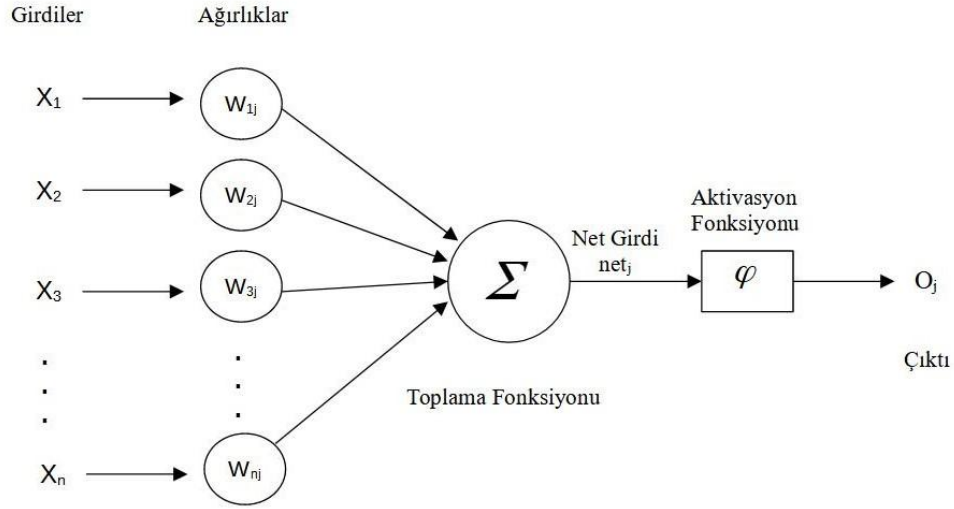


Şekil 2.3. Sinir hücresi yapısı (Oztürk, 2011)

2.1.2. Yapay Sinir Hücresi

Biyolojik sinir ağlarında yer alan sinir hücreleri gibi YSA'larda da yapay sinir hücreleri vardır ve bu hücrelere nöron denilmektedir. Biyolojik sinir hücresinin matematiksel gösterimi ilk olarak McCulloch ve Pitts (McCulloch ve Pitts, 1943) tarafından oluşturulmuştur. Bir yapay sinir ağı birbiriyle bağlantılı yapay sinir hücrelerinden oluşmaktadır.

Yapay sinir hücresi temel olarak giriş, ağırlık, ön işlem fonksiyonu, aktivasyon fonksiyonu ve çıkış olarak adlandırılan beş elemandan oluşmaktadır. Yapısı biyolojik nöronlara göre daha basittir. Şekil 2.4'te yapay sinir hücresinin yapısı verilmiştir.



Şekil 2.4. Yapay sinir hücresi yapısı

Yapay sinir hücresine dışarıdan ya da farklı bir nörondan gönderilen bilgiler girdi verileri olarak adlandırılmaktadır. Girdiler ağırlık öğrenmesi istenen örnekler ile belirlenmektedir (Öztemel, 2003). Biyolojik sinir sistemi elemanları ve yapay sinir sistemi elemanlarının karşılaştırılmış gösterimi Tablo 2.1’ de verilmiştir.

Tablo 2.1. Biyolojik sinir sistemi ve yapay sinir sistemi elemanlarının karşılaştırılması

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	Yapay sinir hücresi
Dentrit	Toplama fonksiyonu
Hücre gövdesi	Aktivasyon fonksiyonu
Akson	Çıktı
Sinaps	Bağlantı ağırlıkları

YSA’ların eğitilmesi, modele verilen girdi verilerine göre beklenen çıktı değerler üretmesini sağlayacak ağırlık değerlerinin bulunması işlemidir. Ağırlıklar yardımı ile gelen girdi verilerinin nöronlara bağlantısı sağlanır ve girdi verilerinin üretilen çıktılar üzerindeki etkileri ayarlanır. Ağırlıklar pozitif, negatif ya da sıfır değerler alabilmektedir. Yapay sinir hücresinde ön işlem fonksiyonu olarak toplama fonksiyonu yaygın olarak kullanılmaktadır. Toplama fonksiyonu hücreye gelen net girdiyi hesaplamaktadır. Bu işlem genellikle girdi verilerinin kendi ağırlık değerleriyle çarpımlarının toplanmasıyla elde edilir.

Aktivasyon fonksiyonları literatürde transfer fonksiyonu ya da öğrenme eğrisi olarak da kullanılmaktadır ve toplama fonksiyonundan gelen net girdiyi işleminden geçirerek hücrenin çıktısını üretir. Diğer bir ifade ile hücrenin çıkış değerlerinin istenilen değerler arasında olmasını sağlar. Yapay sinir hücrelerinde genellikle doğrusal olmayan bir aktivasyon fonksiyonu kullanılmaktadır (Günler, 2012). Kullanılan hücre modeline göre aktivasyon fonksiyonları da farklılık gösterebilir. En yaygın kullanılan aktivasyon fonksiyonları basamak (step), sigmoid, doğrusal (linear) ve hiperbolik tanjant fonksiyonlarıdır.

Doğrusal (linear) fonksiyon, doğrusal problemlerin çözümünde doğrusal fonksiyon tercih edilir. Sürekli ve türevlenebilir fonksiyon olan doğrusal aktivasyon fonksiyonunda girdi verileri herhangi bir değişiklik olmadan çıktıya iletilir. Doğrusal fonksiyonun formülü Denklem (2.1)'de verilmiştir.

$$f(x) = x \quad (2.1)$$

Basamak (Step) fonksiyonu ile net girdinin belirli bir eşik değerinin altında ya da üstünde olması durumuna göre bir çıktı değeri üretilir. Çıktı değeri 1 ve 0 olmak üzere iki değer alır. Basamak fonksiyonunun formülü Denklem (2.2)'de verilmiştir.

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad (2.2)$$

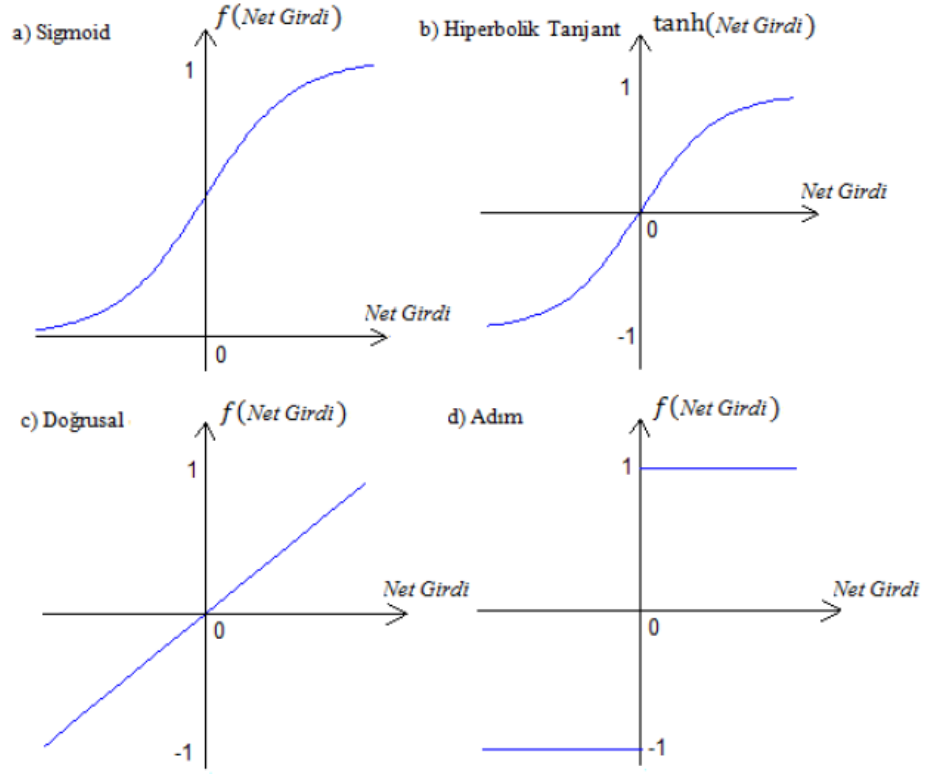
Doğrusal olmayan ve türevlenebilir bir fonksiyon olan sigmoid fonksiyonu yapay sinir ağlarında sıklıkla kullanılmakta olan aktivasyon fonksiyonlarından biridir. Her girdi değeri için sigmoid fonksiyonu 0 ile 1 arasında çıktı değeri üretmektedir. Sigmoid fonksiyonunun formülü Denklem (2.3)'te verilmiştir.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Sürekli, türevlenebilir ve doğrusal olmayan bir fonksiyon olan hiperbolik tanjant fonksiyonu sigmoid fonksiyonuna benzemektedir. Sigmoid fonksiyonu 0 ve 1 arası çıktı değerleri üretirken hiperbolik tanjant fonksiyonu -1 ve 1 arasında değerler üretmektedir. Sigmoid fonksiyonuna göre avantajı türevinin daha dik olmasından dolayı hızlı öğrenme sağlamasıdır. Hiperbolik tanjant fonksiyonunun formülü Denklem (2.4)'te verilmiştir.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.4)$$

Aktivasyon fonksiyonlarının grafikleri sigmoid fonksiyonu için Şekil 2.5.(a)'da, hiperbolik tanjant fonksiyonu için Şekil 2.5.(b)'de, doğrusal (linear) fonksiyon için Şekil 2.5.(c)'de ve basamak (step) fonksiyonu için Şekil 2.5.(d)'de verilmiştir.



Şekil 2.5. Aktivasyon fonksiyonları (Yalçın, 2012)

Son olarak çıkış elemanında yapay sinir hücresinin üretmiş olduğu çıktı değeri sonuç olarak verilir. Sonuç değeri farklı bir hücreye giriş verisi olarak gönderilebileceği gibi kendi kendine de bu veriyi giriş olarak gönderebilmektedir.

2.1.3. YSA'nın Yapısı

Yapay sinir hücreleri bir topluluk biçimine işlem gördüklerinde bu topluluğa ağ denilmektedir. Bir yapay sinir ağında çok sayıda yapay sinir hücresi bulunmaktadır. Bu hücreler aynı doğrultu ve farklı doğrultu üzerinde konumlanabilir. Aynı doğrultu üzerinde bulunan yapay sinir hücrelerinin oluşturmuş olduğu yapıya katman adı verilmektedir. YSA'da üç katman türü bulunmaktadır. Bu katmanlara girdi katmanı, gizli katman ve çıktı katmanı denilmektedir.

Girdi katmanında dış dünyadan alınan veriler işlem elemanları aracılığıyla gizli katmana taşınır. Bu katmanda herhangi bir bilgi işleme olması gerekmez. Gizli katmanda girdi katmanından gelen veriler işlem elemanları aracılığıyla ileterek çıktı katmanına gönderilmektedirler. Yapay sinir ağında gizli katman tek bir katmandan oluşabildiği gibi birden fazla katman ile de YSA modellerinde kullanılabilir. Çıktı katmanında ise gizli katmandan gelen bilgileri işleyerek girdi katmanında sunulan girdi kümesi için üretmesi gereken çıktıyı üretir ve ürettiği çıktı verisini dış dünyaya gönderir. Çıktı katmanında bir yapay sinir hücresi olabildiği gibi birden fazla hücre de ağ yapısında kullanılabilir. Bu sayı problemin türüne göre değişiklik göstermektedir.

2.1.4. YSA'nın Sınıflandırılması

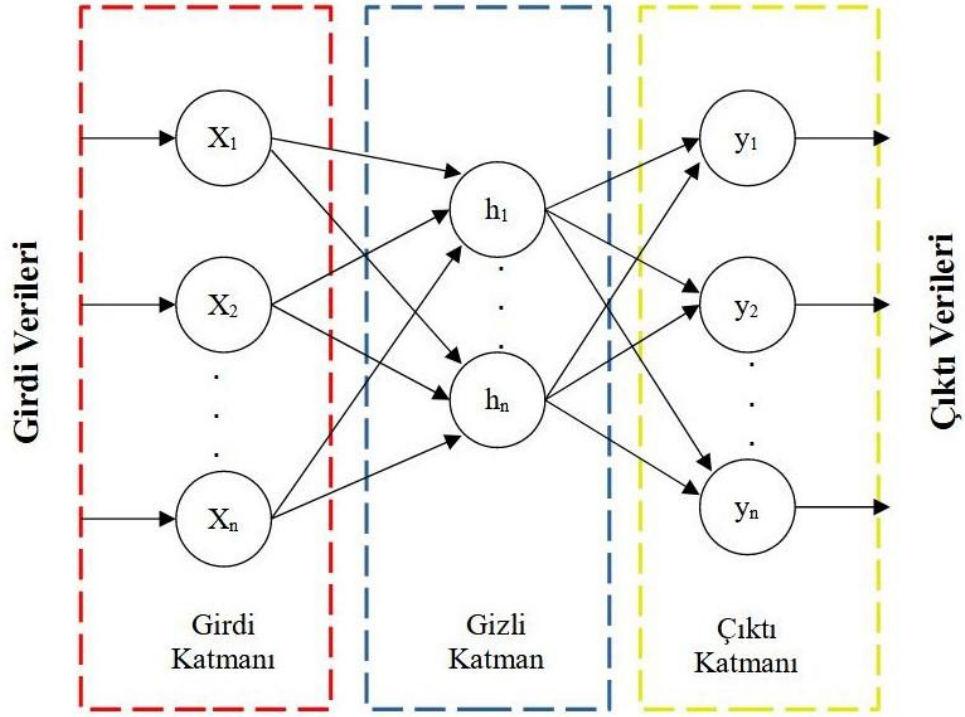
YSA'lar farklı şekillerde sınıflandırılabilir. YSA'ları yapılarına, katman sayılarına ve öğrenme stratejilerine göre farklı şekillerde inceleyebiliriz.

2.1.4.1. YSA'nın Yapılarına Göre Sınıflandırılması

YSA'lar oluşturuldukları hücrelerin birbirlerine bağlantı şekillerine göre ileri beslemeli sinir ağları, geri beslemeli sinir ağları olmak üzere iki farklı sınıfa ayrılırlar.

İleri Beslemeli Sinir Ağları

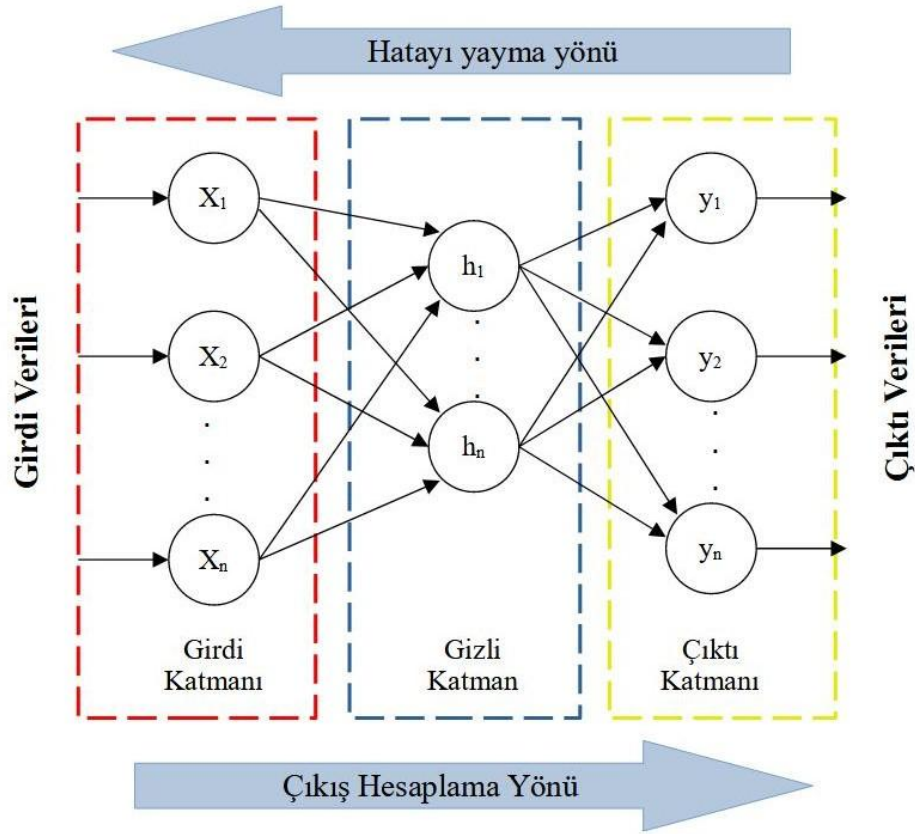
YSA'nın en basit modeli olan ileri beslemeli yapay sinir ağlarında bilgi akışı girdi katmanından çıktı katmanına doğru gerçekleşmektedir. Geri yönlü bir bilgi akışı söz konusu değildir. Katmanlar arasındaki bağlantılarda döngü oluşmaz. Çıkış değerleri sadece giriş verilerinin ara katmanlarda işlenmesi ile elde edilir ve daha önceki giriş verilerine bağlı değildir (Karakaş, 2020). YSA modellerinde en geniş kullanıma sahip sinir ağlarıdır. İleri beslemeli yapay sinir ağlarına örnek olarak Çok Katmanlı Algılayıcılar (Multi Layer Perceptron - MLP) ve Öğrenen Vektör Kuantizasyonu (Learning Vector Quantization - LVQ) verilebilir. İleri beslemeli yapay sinir ağı modelinin yapısı Şekil 2.6'da verilmiştir.



Şekil 2.6. İleri beslemeli YSA modeli

Geri Beslemeli Sinir Ağları

Geri beslemeli sinir ağlarında sadece ileri yönlü bilgi akışı yerine geri yönlü bilgi akışı da gerçekleşmektedir. Bir yapay sinir hücresinin çıktı verisi diğer hücrelere ya da kendisine farklı bağlı yapay sinir hücreleri aracılığıyla giriş verisi olarak geri beslenmektedir. Geri besleme aynı katmandaki hücreler arasında olabildiği gibi farklı katmanlarda bulunan yapay sinir hücreleri arasında da gerçekleşebilmektedir (Kulluk, 2009). Bu sayede geri beslemeli ağ modeli doğrusal olmayan dinamik bir yapı oluşturur. Geri beslemeli yapay sinir ağı modeli Şekil 2.7’de verilmiştir.



Şekil 2.7. Geri beslemeli YSA modeli (Gallo, 2015)

Geri beslemeli ağlar tahminleme uygulamalarında başarılı bir şekilde kullanılmaktadır. Bu ağ modellerinden Hopfield, Jordan ve kendi kendine organize olan haritalar (Self Organizing Map - SOM) ağları bilinen geri beslemeli ağ modellerine örnek olarak verilebilir (Ataseven, 2013).

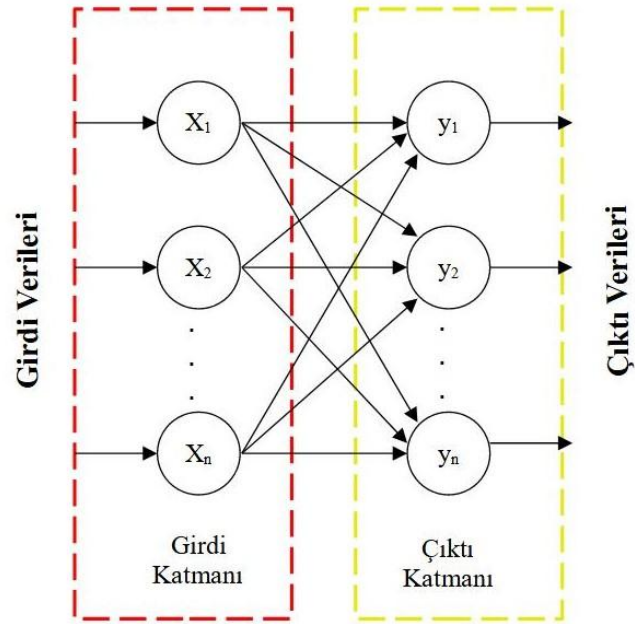
2.1.4.2. YSA'nın Katman Sayılarına (Mimarilerine) Göre Sınıflandırılması

Yapay sinir ağları girdi katmanı ve çıktı katmanından oluşur. Bazı modellerde bu iki katmana ek olarak gizli katman adı verilen ara katman da bulunabilir. Gizli katman tek bir katman ya da birden çok sayıda katman içerebilmektedir.

Tek Katmanlı YSA

Tek katmanlı YSA'lar sadece girdi ve çıktı katmanından oluşan YSA modelleridir. Bu ağlarda gizli katman bulunmaz. F. Rosenblatt tarafından geliştirilmiş olan ilk YSA modellerinden biridir (Rosenblatt, 1958) ve geliştirilen YSA modelleri için de tek katmanlı YSA'lar temel model olmuşlardır (Karakaş, 2020). Doğrusal problemlerin çözümünde kullanılırlar. Doğrusal olmayan problemlerin çözümü için ise çok katmanlı YSA'lar geliştirilmiştir. Tek katmanlı

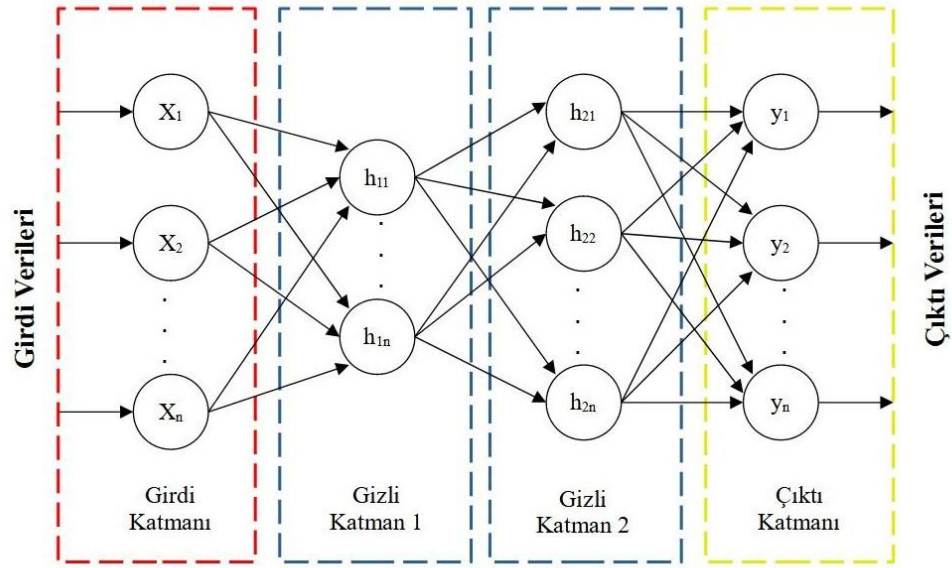
YSA modeli Şekil 2.8’de verilmiştir.



Şekil 2.8. Tek katmanlı YSA modeli

Çok Katmanlı YSA

Çok katmanlı YSA’lar doğrusal olmayan problemlerin çözümünde kullanılmak için geliştirilmiş modellerdir. Yapılarında girdi ve çıktı katmanına ek olarak bir veya birden fazla ara katman bulundurur. Bu ara katmanlara gizli katman adı da verilmektedir. Günümüzde birçok mühendislik probleminde ve sınıflandırma problemlerinde yüksek performanslı modeller olarak kullanılmaktadır (Yılmaz, 2014). Örnek bir çok katmanlı YSA mimarisi Şekil 2.9’da verilmiştir.



Şekil 2.9. Çok katmanlı YSA mimarisi örneği

2.1.4.3. YSA'nın Öğrenme Stratejilerine Göre Sınıflandırılması

YSA'nın girdi olarak gönderilen veriler ile sonuç olarak istenen çıktıyı üretebilmesi ağız öğrenmesi sayesinde gerçekleşir. Öğrenme işlemi yapay sinir hücreleri arasında bulunan ağırlık ve bias değerlerinin değiştirilmesi ile yapılır. Araştırmacılar YSA eğitimi için çok fazla çalışma yapmış ve birçok öğrenme algoritmaları geliştirmiştir. Geliştirilen öğrenme algoritmalarının YSA model performansında performansını artıran ya da azaltan özellikleri mevcuttur. Bu nedenle YSA için seçilecek olan öğrenme algoritmasının iyi bir algoritma olması önemlidir. Yapay sinir ağızları öğrenme stratejilerine göre üç farklı grupta incelenmektedir. Bunlar danışmanlı öğrenme, danışmansız öğrenme ve takviyeli öğrenme stratejileridir.

Danışmanlı Öğrenme

Danışmanlı öğrenme stratejisi, ilk kez 1949 yılında Donald Hebb tarafından önerilmiştir. Bu stratejide, öğrenme danışman gözetiminde gerçekleştirilir. YSA modeli, kullanılmadan önce eğitilir. Eğitim sürecinde, hem girdi değerleri hem de bu girdi değerleri için istenen çıktı değerleri sisteme sunulur. Ağ, sunulan örneklerden genellemeler yaparak bir çözüm uzayı oluşturur. Daha sonra bu çözüm uzayı, benzer girdiler için ağızdan sonuç üretmesini sağlar. Eğitim aşaması genellikle zaman alıcıdır (Yalçın, 2012). Model başlangıçta rastgele ağırlık değerleriyle başlatılır ve çıktı

değeri için bir tahmin yapar. Modelin ürettiği çıktı ile hedef çıktı arasındaki fark hata olarak kabul edilir ve bu hata değeri minimize edilmeye çalışılır (Kulluk, 2019). Öğrenme işlemi, belirlenen hata oranına yaklaşıldığında veya maksimum iterasyon sayısına ulaşıldığında sonlandırılır. Sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılmaktadır (Liu ve Wu, 2012). MLP'ler, danışmanlı öğrenme stratejisini kullanan YSA'ların bir örneğidir.

Danışmansız Öğrenme

Danışmansız öğrenme stratejisinde, modelin eğitiminde girdi verileriyle birlikte çıktı verileri sağlanmaz. Sadece girdi verileri ağa iletilir. Bu stratejide, modelin öğrenmesine yardımcı bir danışman bulunmaz. Girdi katmanına verilen girdi verileriyle, yapay sinir ağı modeli, parametreler arasındaki ilişkileri öğrenerek sınıflandırma kuralları geliştirir. Danışmanlı öğrenmeye göre daha hızlı çalışabilir. Özellikle kümeleme problemlerinde ve boyut indirgemedede danışmansız öğrenme stratejisi yaygın olarak kullanılır (Francis, 2014; Wang, 2001). ART (Adaptive Resonance Theory) ağları, danışmansız öğrenme stratejisini kullanan modellere bir örnektir.

Takviyeli Öğrenme

Takviyeli öğrenme stratejisinde modelin öğrenmesine yardımcı bir danışman bulunur. Modele verilen girdi verileri için üretmesi istenen çıktı verilere sisteme verilmez bunun yerine modele gönderilen girdi verilerine karşılık bir çıktı üretmesi beklenir. Üretilen çıktı değerinin doğru ya da yanlış olduğu danışman tarafından belirtilir. Bu geri bildirimler sayesinde ağ deneme yanılma yöntemi ile öğrenmesini devam ettirir (Öztemel, 2003; Türkoğlu, 2019). Takviyeli öğrenme stratejisi robotik çalışmalarında sıklıkla kullanılmaktadır (Sutton ve Barto, 1998). LVQ ağları, takviyeli öğrenme stratejisini kullanan YSA modelidir.

2.1.5. Derin Sinir Ağları

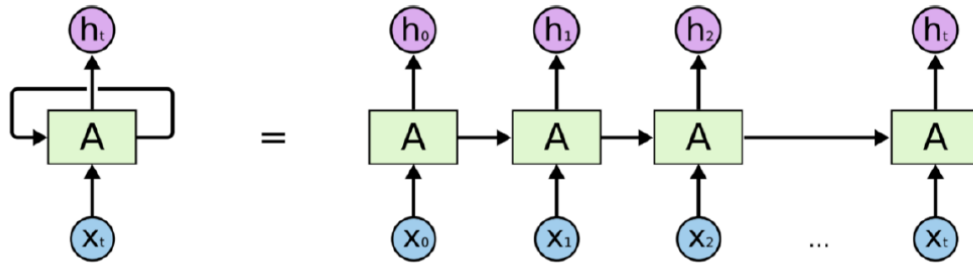
Derin öğrenmenin alt dalı olan derin sinir ağları giriş ve çıkış katmanı arasında birden çok gizli katmandan oluşan yapay sinir ağı türüdür (Akpak, 2017). Gizli katmanlarda girdi verilerinin farklı özellikleri öğrenilir ve öğrenilen özellikler birbirleriyle etkileşimli çalışarak çıktı verisi üretilir (Bengio, 2009).

Derin sinir ağları (DNN) yüksek miktarda veri ile eğitilerek karmaşık problem çözümünde kullanılabilir. DNN'ler her bir katmanda bulunan yapay sinir

hücreleri arasındaki bağlantıları ifade eden ağırlık değeri kullanır. Eğitim boyunca hata değeri minimize edilerek ağırlık değerleri ağı verilen verilere göre güncellenir. Bu şekilde ağı doğru sonuçlar üretme yeteneği geliştirilir. Günümüzde DNN'ler özellikle ses tanıma, görüntü işleme ve doğal dil işleme alanlarında başarılı bir şekilde kullanılmaktadır. Yinelemeli sinir ağları (Recurrent Neural Networks), uzun kısa süreli bellek ağları (Long Short-Term Memory) ve evrişimsel sinir ağları (Convolutional Neural Networks) popüler derin sinir ağı modelleridir.

2.1.5.1. Yinelemeli Sinir Ağları (RNN)

RNN'ler yapılarında gizli katmanlar arasında en az bir döngüsel bağlantıya yani geri besleme döngüsüne sahip ağlardır. Geri besleme döngüleri RNN modellerinin hafızayı kodlamasını sağlar (Talathi ve Vartak, 2015). RNN modellerinde girdi verileri ağı verildiğinde ürettiği çıktı bilgisini tekrar giriş olarak kullanır. Bu sayede hafızanın korunmasını sağlayan bir döngü oluşturulmuş olur (Salouhou, 2019). Şekil 2.10'da RNN döngü şeması verilmiştir.



Şekil 2.10. RNN döngü şeması

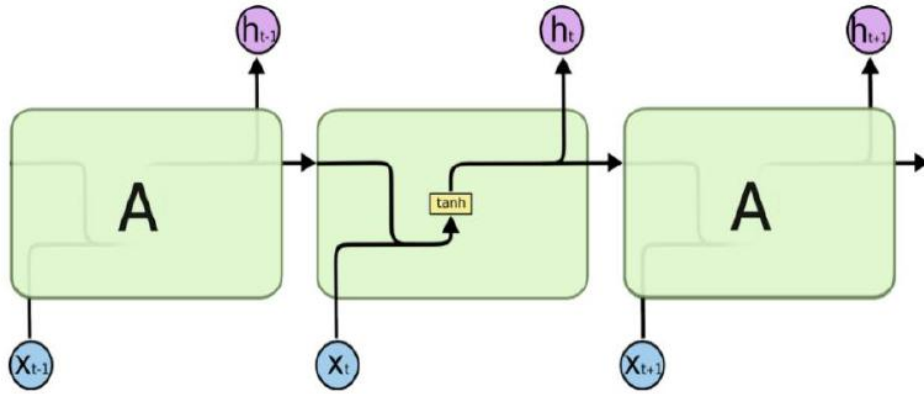
RNN'ler, bir dizi hücre (cell) veya gizli durum (hidden state) kullanır. Her bir hücre, mevcut girişi işlerken bir önceki gizli durumu yani geçmiş bilgiyi içerir. Bu sayede RNN, gelen verinin yanı sıra geçmiş durum bilgisini de hesaba katarak çıktı üretir.

RNN'ler ardışık veri dizilerini işlemek ve zaman bağımlılığına sahip verileri modellemek için tasarlanmıştır. Zaman serileri, doğal dil işleme, konuşma tanıma gibi zaman bağımlı verilerin işlenmesi için yaygın olarak kullanılmaktadır. RNN'ler uzun vadeli bağımlılıkları etkili bir biçimde işleyemedikleri için LSTM (Long Short-Term Memory) gibi daha gelişmiş RNN varyantları geliştirilmiştir.

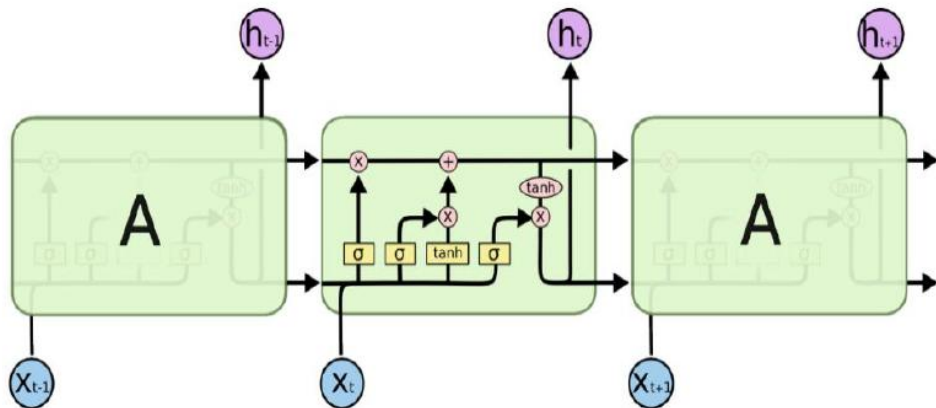
2.1.5.2. Uzun Kısa Süreli Bellek Ağları (LSTM)

LSTM ağları temelinde RNN'e dayanan bir derin sinir ağıdır. RNN'ler önceki girişi hatırlar ve tekrarlayan yapıları kullandığından önceki çıktıları depolamak için harici kayıt ve bellek gerektirmez. LSTM ağında bir önceki duruma bağlı olarak mevcut çıktı belirlenir. Bu algoritma uzun vadeli bağımlılık problemini çözmekte ve gradyan iniş gibi yöntemler kullanarak RNN modelinin dezavantajlarını aşmaktadır (Bengio vd., 1994).

LSTM, bir hafıza hücresi (memory cell) ve çeşitli kapılar (gates) kullanarak bilgi akışını kontrol eder. LSTM'lerde tekrarlayan hücreler 4 katmanlı yapıdadır, bu nedenle RNN'e göre karmaşık bir yapıya sahiptir. Şekil 2.11'de standart tek katmanlı RNN mimarisi ve Şekil 2.12'de ise dört etkileşim katmanlı LSTM'nin mimari yapıları verilmiştir (Salouhou, 2019).



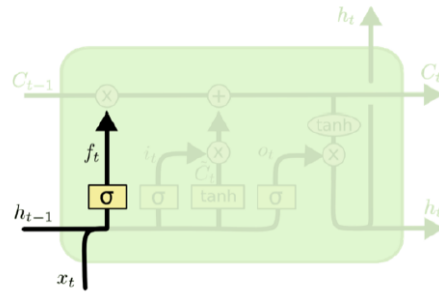
Şekil 2.11. Tek katmanlı standart RNN modeli



Şekil 2.12. Dört etkileşim katmanlı LSTM modeli

LSTM işlemi aşağıdaki belirtilen dört adımda gerçekleşmektedir (Salouhou, 2019);

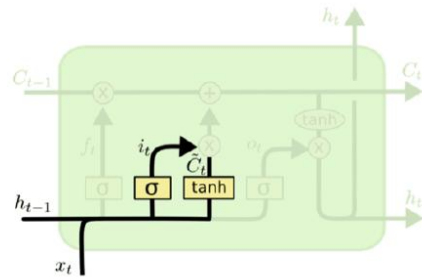
1. Giriş vektörü LSTM'ye zaman adımı boyunca verilir. Giriş vektörü mevcut zamandaki veriyi temsil etmektedir. Bu adımda ağ unutmaya kapısı katmanı (forget gate layer) olarak bilinen sigmoid fonksiyon katmanında hangi bilgilerin hafızadan silineceğine karar vermektedir. Şekil 2.13'te unutmaya geçidi katmanının nasıl kullanıldığı verilmiştir.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Şekil 2.13. Unutmaya kapısı katmanı

2. İkinci adımda ağ hücre durumuna depolanacak olan bilgiye karar verir. Giriş kapısı katmanı adı verilen sigmoid fonksiyon katmanı güncellenecek olan giriş verisine karar verir. Yeni aday değerlerin vektörünü oluşturan tanh katmanı Şekil 2.14'te gösterildiği şekilde kullanılmaktadır.

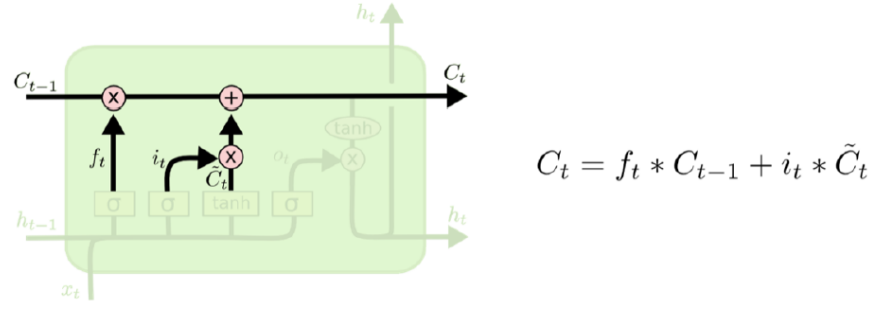


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

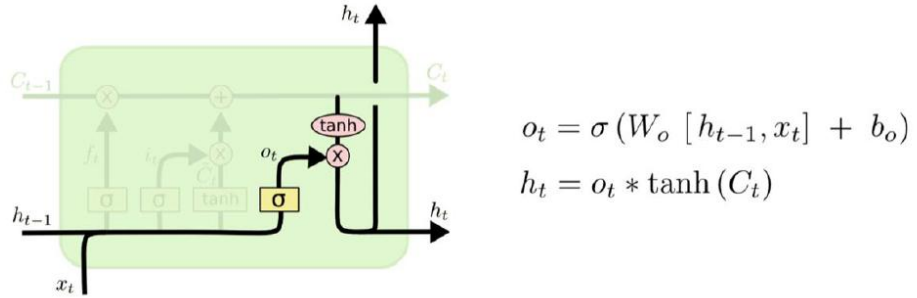
Şekil 2.14. Giriş kapısı ve yeni aday değer vektörü

3. Bu adımda eski hücre durumu yeni hücre durumuna güncellenmektedir. Eski bilgiler unutulur ve yeni bilgiler hafıza durumuna eklenir. Ölçeklenen yeni aday oluşumu Şekil 2.15'te verilmiştir.



Şekil 2.15. Yeni hücre durumu

4. Son adımda güncellenen hafıza durumu ve çıkış kapısı kullanılarak çıktı değeri hesaplanır. Hesaplanan çıktı değeri bir sonraki zaman adımında kullanılmak üzere kayıt edilir ya da dışarı verilir. Şekil 2.16'da çıktı değerinin oluşumu verilmiştir.



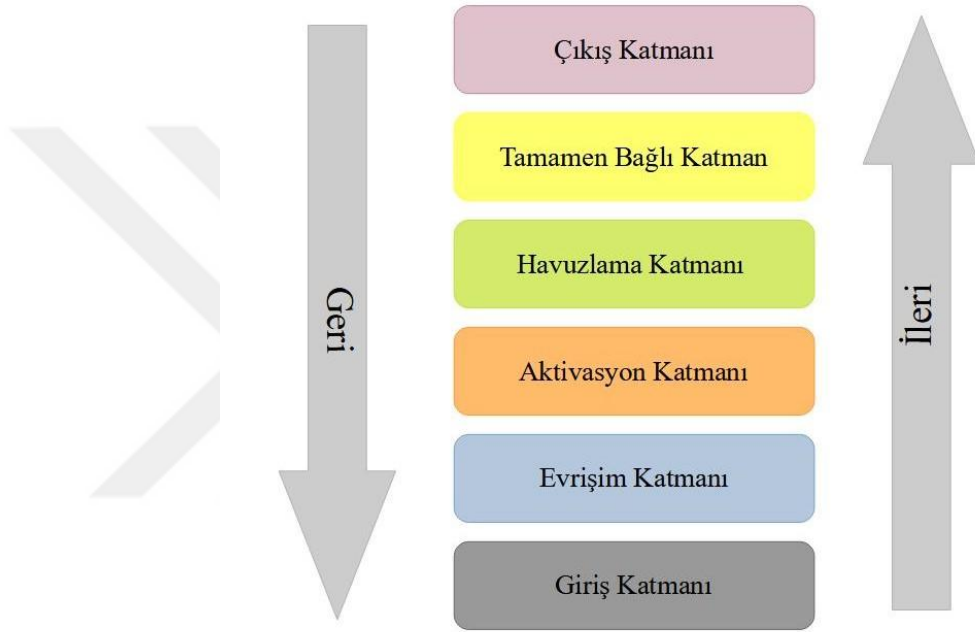
Şekil 2.16. Çıkış kapısı ve yeni bilgi

2.1.5.3. Evrişimsel Sinir Ağları (CNN)

Evrişimsel sinir ağı (Convolutional Neural Network) modelleri, derin öğrenme yöntemlerinin yaygın olarak kullanılan özel bir mimarisidir. Özellikle son yıllarda yüz algılama, konuşma tanıma, el yazısı tanıma, davranış tanıma ve görüntü sınıflandırma alanında başarılı bir yöntem olarak kullanılmaktadır. Evrişimsel sinir ağları görüntü ve video gibi iki boyutlu verilerden anlamlı bilgiler çıkarabildiği gibi tek boyutlu ve üç boyutlu veriler için de kullanılabilirler (Toprak, 2022).

Temel olarak evrişimsel sinir ağı modeli girdi katmanı, gizli katman ve çıktı katmanından oluşmaktadır. Gizli katman ise konvolüsyonel katman (evrişim katmanı), havuzlama katmanı ve tam bağlantılı katman olarak adlandırılan üç katmandan oluşur. Evrişim katmanında giriş verisi filtrelenir ve özellik haritaları elde edilir. Giriş verisi görsel ya da ses gibi veriler olabilmektedir. Havuzlama katmanında ise evrişim katmanında elde edilen özellik haritaları örneklenir ve

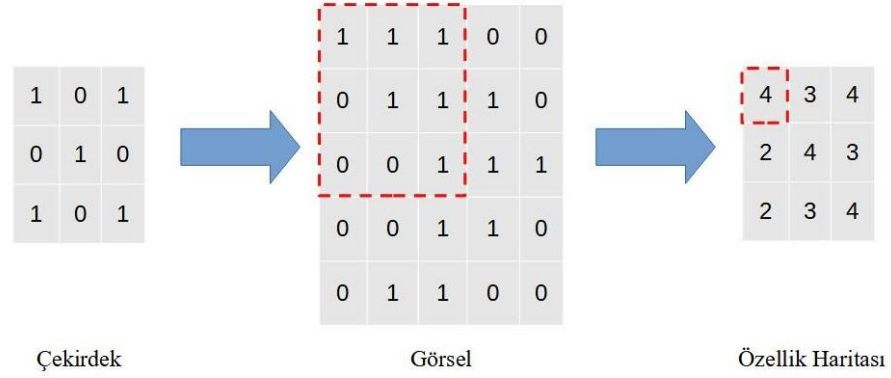
modelin kapsamlı ve hızlı öğrenmesi sağlanır. Tam bağlantılı katmandaki sinir hücreleri, önceki katmandan iletilen tüm giriş verilerine uygun olacak şekilde çıktı verileri üretir. Her katman, önceki katmanın çıktısına göre bir özellik oluşturur ve tüm katmanları bir araya getirerek eğitir, böylece özellik hiyerarşisi öğrenilir. Sonuç olarak, düşük seviye detaylardan yüksek seviye detaylara doğru etkili bir öğrenme hedeflenir (Acı ve Çırak, 2019). CNN ağı katmanlarının blok diyagramı Şekil 2.17’de verilmiştir.



Şekil 2.17. CNN katmanlarının blok diyagramı

Evrişim Katmanı

Evrişim katmanı evrişimsel sinir ağlarına adını veren katmandır. Evrişim katmanında amaç girdi verilerinin özelliklerini algılamaktır. Bu katmanda konvolüsyon yani evrişim adı verilen işlem gerçekleştirilir. Evrişim işlemi bir dizi ağırlık ile giriş verisinin çarpılması ile gerçekleştirilen doğrusal bir işlemdir (Toprak, 2022). Evrişim katmanı özellikleri filtre ve çekirdek yardımcılığıyla çıkarmaktadır. Evrişim katmanında içerdiği filtreler giriş verileri üzerinde yatay ve dikey şekilde gezecek biçimde konumlandırılır ve filtrelerin ağırlık değerleri ile giriş verisi filtreler kaydırılarak çarpılır. Şekil 2.18’de evrişim işlemi örneği verilmiştir.



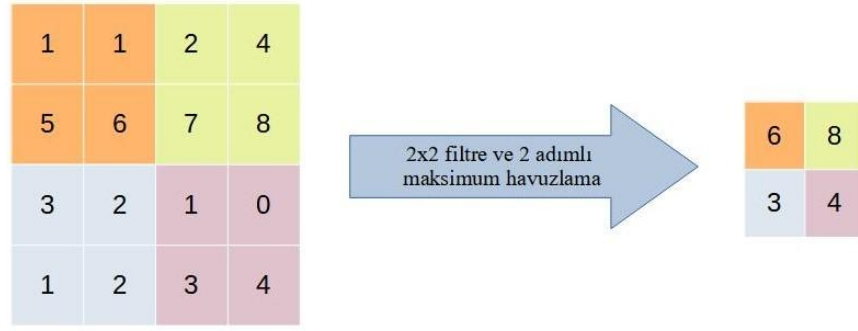
Şekil 2.18. Evrişim işleminden özellik haritası çıkarma örneği

CNN modelleri genellikle iki boyutlu verilerle kullanıldığından, evrişim işlemi girdi verileri dizisi ile iki boyutlu bir ağırlık dizisi olan çekirdek arasında gerçekleştirilir (Sun vd., 2020). Evrişim katmanının çıktısı evrişim işleminden sonra elde edilen özellik haritalarıdır. Derinlik (depth), adım (stride) ve sıfır dolgusu (zero-padding) adı verilen parametreler özellik haritalarının boyutunu belirlemektedir (Akpak, 2017).

Evrişim katmanında çıktı verisi olarak özellik haritaları elde edildikten sonra ağırlık karmaşık görevleri öğrenebilmesi için çıktı aktivasyon fonksiyonuna gönderilir. Aktivasyon fonksiyonu çıktı değerinin kullanılan fonksiyon türüne göre dönüşümünü yaparak bir sonraki nörona aktarır.

Havuzlama Katmanı

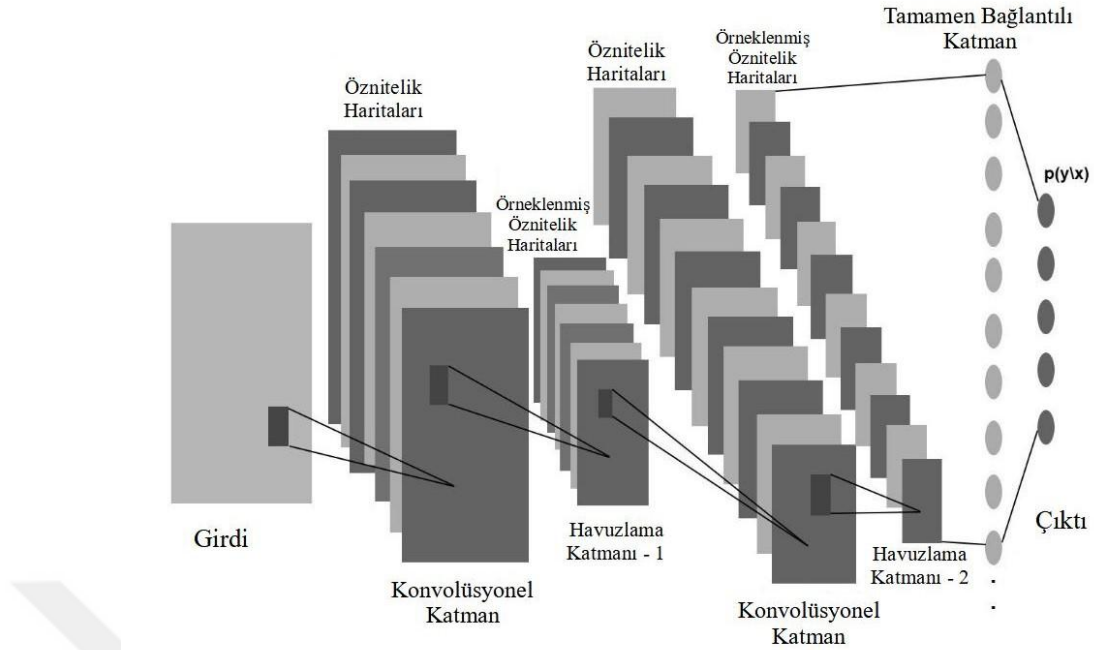
İki konvolüsyon katmanı arasında bulunan havuzlama katmanının amacı giriş boyutunu (genişlik ve yükseklik) düşürerek karmaşıklığı azaltmak (Wang vd., 2019) ve verinin önemli özelliklerini korumaktır. Havuzlama katmanı giriş verisinin her alt bölümüne uygulanır ve veri boyutunu küçülttüğü için hesaplama maliyetini azaltır (Toprak, 2022). En popüler havuzlama işlemi maksimum havuzlamadır. Maksimum havuzlamada veri alt bölmelere bölünerek her bölgedeki maksimum değer seçilir (Sun vd., 2017). Şekil 2.19’da maksimum havuzlama işlem örneği verilmiştir.



Şekil 2.19. Maksimum havuzlama örneği

Tam Bağlantılı Katman

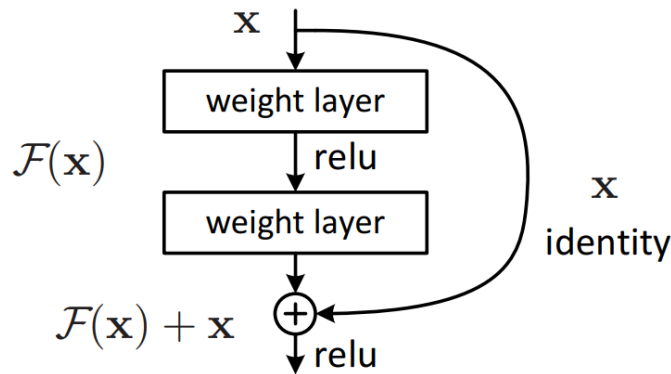
Tam bağlantılı katman CNN modellerinin sonuna yerleştirilmiş olan katmandır. Evrişim ve havuzlama katmanlarının uygulanması sonrası elde edilen özellik haritalarını yorumlar. Yapısı klasik sinir ağlarında olduğu gibi tam bağlantılı katmanda da tüm yapay sinir hücreleri önceki katmandaki hücrelerle bağlantılı olacak şekildedir (Cantemir, 2019). Tam bağlantılı katmanda özellikler birleştirilir ve düzleştirilerek vektör formuna dönüştürülür ve sonuç çıktı katmanına iletilir. Bu katmanlar ağı daha derin ve karmaşık ilişkileri öğrenmesini sağlar. Genellikle sınıflandırma problemlerinin çözümüne uyarlanır ve softmax fonksiyonu kullanılır. Örnek bir iki boyutlu girdi verisi içeren evrişimli sinir ağı mimarisi Şekil 2.20’de verilmiştir.



Şekil 2.20. Örnek bir 2D evrişimli sinir ağı mimarisi

ResNet50

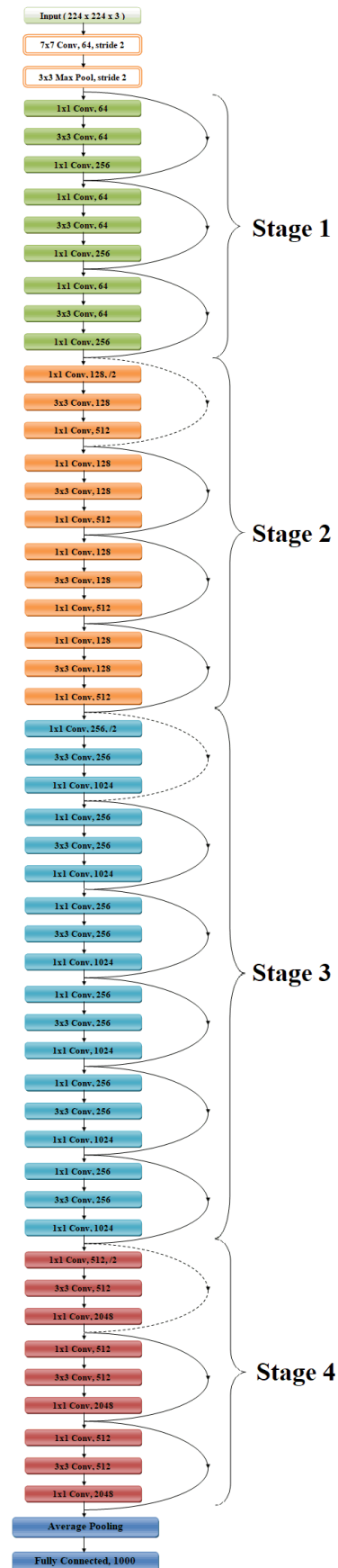
ResNet, Residual Network (artık ağı) isminin kısaltmasıdır. ResNet modelinin geliştirilmiş versiyonu olan ResNet50 modeli 2015 yılında artık öğrenmeyi kullanan ilk CNN mimarisidir. ResNet50 modeli 50 katmandan oluşmaktadır ve ImageNet veritabanındaki 1000 kategoriye ait milyonlarca görüntü üzerinde eğitilmiştir (Nedjai, 2020). Artık öğrenme, derin evrişimli sinir ağlarını eğitmek için girişi doğrudan diğer sonraki katmanlara (yalnızca bitişik katmana değil) bağlamak için kısayol bağlantıları kullanır (He vd., 2016) ve birçok görüntü işleme çalışmasında diğer mimariler için omurga görevi görmektedir.



Şekil 2.21. Artık öğrenmenin yapı taşı (He vd., 2016)

ResNet-50, bilgisayar görüşü alanında doğruluk oranını artırmada başarılı olmuş ve 2015 yılında düzenlenen ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ILSVRC, 2015) ve Microsoft Ortak Nesnelerin Bağlamı (MS COCO, 2015) yarışmalarında birincilik elde etmiştir (He vd., 2016). Geleneksel CNN modellerinden farklı olarak ek bir kimlik eşleme yeteneğine sahiptir. Bu bağlantılar derin katmanlarda bilgi kaybının önlenmesi amacıyla ağ mimarisine eklenmektedir. ResNet'te kullanılan kimlik eşleme, mevcut katman gereksiz ise CNN ağırlık katmanını atlamak için bir yol sağlar. Bu, eğitim kümesinde aşırı uyum sorununu önlemeye yardımcı olur (Theckedath ve Sedamkar, 2020).

ResNet50 mimarisinde 5 evrişim katmanı kullanılmış ve ardından bir maksimum havuzlama katmanı yerleştirilmiştir. Tam bağlantılı katmanda ortalama havuzlama kullanılır ve sınıflandırma işlemi için softmax fonksiyonu tercih edilir (Al-Haija ve Adebajo, 2020). ResNet50 mimarisi Şekil 2.22'de verilmiştir.

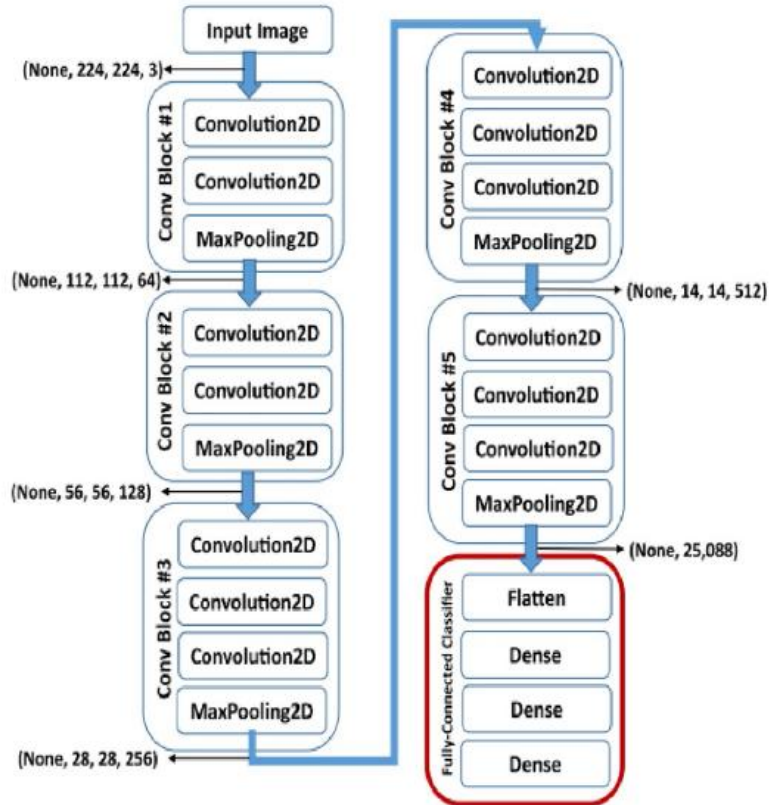


Şekil 2.22. ResNet50 mimarisi

VGG-16

VGG-16 ImageNet veritabanında eğitilmiş, yaygın olarak kullanılan ve en iyi bilgisayar görüşü modellerinden biri olarak bilinen CNN türüdür (Nezla vd., 2021). VGG kısaltması, Oxford Üniversitesi'ndeki “Visual Geometry Group” tarafından tasarlandığı anlamına gelir. VGG-16 ağı kapsamlı olarak eğitilmiş olması sayesinde küçük veri setleri üzerinde bile yüksek performans sağlamaktadır (Theckedath ve Sedamkar, 2020).

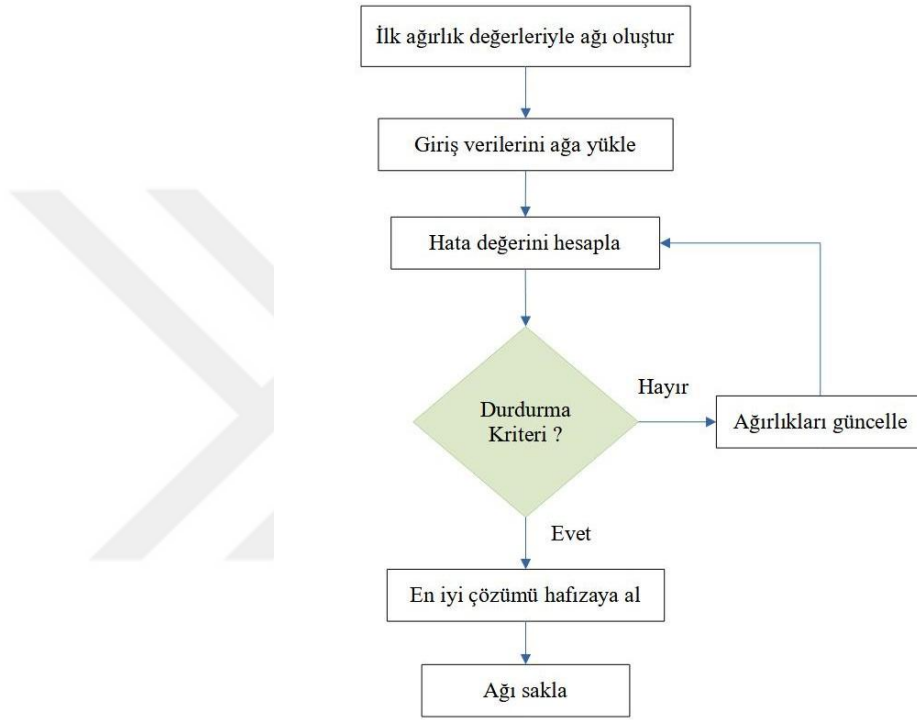
VGG-16 ağı aynı dolgu (padding) kullanarak 3x3 filtre boyutuna sahip 16 evrişim katmanından oluşmaktadır. Evrişim katmanının ardından 2x2 boyutlu maksimum havuzlama (max pooling) işlemi gerçekleştirilir ve ağ yapısında 5 adet havuzlama katmanı bulunmaktadır. Model üç adet tamamen bağlı katmandan oluşmakta ve son katmanda softmax sınıflandırıcısı kullanılmaktadır. VGG-16 mimarisinde yaklaşık olarak 138 milyon eğitilebilir parametre mevcuttur (Pravitasari vd., 2020). VGG-16 ağının mimarisi Şekil 2.23'te verilmiştir.



Şekil 2.23. VGG-16 mimarisi

2.2. YSA Eğitiminde Kullanılan Optimizasyon Algoritmaları

Yapay sinir ağlarının en önemli özelliği öğrenebilme yetenekleridir. Yapay sinir ağları istenilen çıktıyı elde edebilmemiz için ağırlıklarını sürekli olarak güncelleyerek öğrenir. Öğrenme işleminin gerçekleşmesi için öncelikle öğrenme algoritmaları kullanılarak yapay sinir ağının eğitilmesi gereklidir. YSA eğitiminin temel akış diyagramı Şekil 2.24’te verilmiştir (Oztürk, 2011).

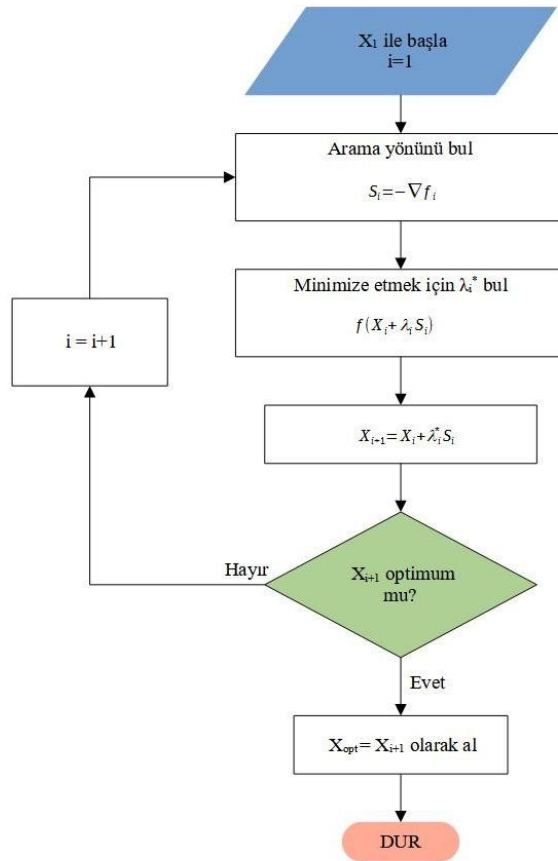


Şekil 2.24. YSA eğitimi temel akış diyagramı

Literatürde yapay sinir ağlarının eğitimi için birçok öğrenme algoritması bulunmaktadır. Öğrenme algoritmalarına örnek olarak Gradient Descent, Scaled Conjugate Gradient ve Levenberg-Marquardt algoritmaları verilebilir.

2.2.1. Gradyan İniş (Gradient Descent) Öğrenme Algoritması

Gradyan iniş algoritması makine öğrenim modellerini ve yapay sinir ağlarını eğitmek için kullanılan en popüler optimizasyon algoritmalarından biridir. Diğer algoritmalar ile birlikte kullanılabilir olması, anlaşılır ve kolay uygulanabilir olması gradyan iniş algoritmasının yaygın olarak kullanılmasında önemli etkenlerdir. GD algoritmasının akış diyagramı Şekil 2.25’te verilmiştir.



Şekil 2.25. GD akış diyagramı

Gradyan iniş algoritması kullanmanın temel amacı, yinelemeli olarak maliyet fonksiyonunu en aza indirmektir.

$$f(w^{(i)}) = f^i \text{ ve } \nabla f(w^{(i)}) = g^{(i)} \quad (2.5)$$

Metot $w^{(0)}$ noktasında başlar ve sonlandırma kriteri gerçekleşene kadar $w^{(i)}$ 'den $w^{(i+1)}$ 'e kadar $d^{(i)} = -g^{(i)}$ eğitim yönünde hareket eder. Gradyan iniş yöntemi Denklem 2.6'da gösterilen formül ile yinelenir.

$$w^{(i)} = w^{(i)} - g^{(i)} \eta^{(i)} \quad i=0,1,\dots \quad (2.6)$$

Burada η ile belirtilen ifade öğrenme oranıdır ve optimuma ulaşmak için atılan adımların boyutu olarak tanımlanır. Öğrenme oranı sabit bir değere ayarlanabilir ya da eğitim yönü boyunca her adımda tek boyutlu optimizasyon ile bulunabilir (Wilson ve Martinez, 2003). GD algoritmasının sözde kodu Şekil 2.26.'da verilmiştir.

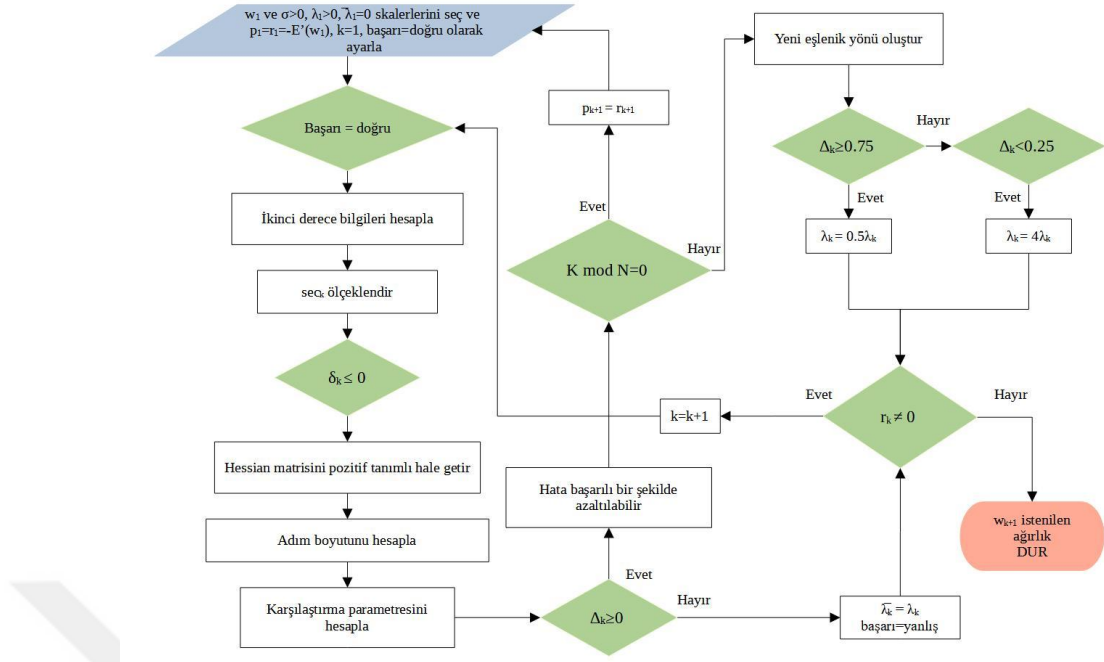
Algoritma 1 Gradyan İniş Algoritması

```
1: Başla  $x \in \mathbb{R}^n$ 
2: Başla  $\alpha \in \mathbb{R}$  { adım boyutu }
3: Başla  $\epsilon$ 
4: while iterations < maxIterations do
5:    $x = x - \alpha \nabla f(x)$ 
6:   if  $f(x) < \epsilon$  then
7:     return  $x$ 
8:   end if
9: end while
10: return  $x$ 
```

Şekil 2.26. GD algoritması sözde kodu

2.2.2. Scaled Conjugate Gradient (SCG) Algoritması

Ölçekli eşlenik gradyan (SCG) algoritması 1993'te Moller tarafından geliştirilmiştir (Møller, 1993). Ağırlıkları ve giriş değerlerinin kullanılabilir ve türevi alınabilir transfer fonksiyonları olan tüm yapay sinir ağlarının eğitiminde tercih edilen bir öğrenme algoritmasıdır. Bu algoritmanın temel fikri çok kullanılan başka bir öğrenme algoritması olan Levenberg-Marquardt (LM)'de de kullanılan model-güven alanı (model-trust region) yaklaşımını eşlenik gradyan yaklaşımı ile birlikte kullanmaktır. Bu sayede zaman alan satır aramasını ortadan kaldırır. Ağırlık ve bias değerleri Conjugate Gradient algoritmasına göre değiştirilmektedir. SCG algoritması, sürekli olarak hesaplama açısından maliyetli olan eğitim girişlerinde ağ tepkilerini ve arama için satır arama sürecini daha hızlı bir yöntemle değiştirir. SCG algoritmasının akış diyagramı Şekil 2.27'de verilmiştir.



Şekil 2.27. SCG algoritması akış diyagramı

Algoritma, Denklem 2.7 ile ikinci dereceden yaklaşımın bilgilerine dayanarak arama çizgisini ve adım boyutunu etkili bir şekilde seçer.

$$E(w + y) \approx E(w) + E'(w)^T + \frac{1}{2}y^T E''(w)y \quad (2.7)$$

SCG'de, her iterasyonda Denklem 2.8 ile optimum mesafeyi hesaplayarak mevcut arama yönünde hareket etmek için bir satır araması yapılır.

$$w_{k+1} = w_k + a_k \times p_k \quad (2.8)$$

Sonraki çağrı yönü, önceki çağrı çizgi yönlerine karşı konjugasyonu hesaplanır. Burada p_k , ikinci türevler matrisi ve Hessian matrisinin hata işlevine bağlı bir fonksiyondur. a_k , Hessian matrisinin belirsizliğini keskinleştirmek ve düzleştirmek için kullanılır (Othman ve Salih, 2021). SCG algoritmasının sözde kodu Şekil 2.28.'de verilmiştir.

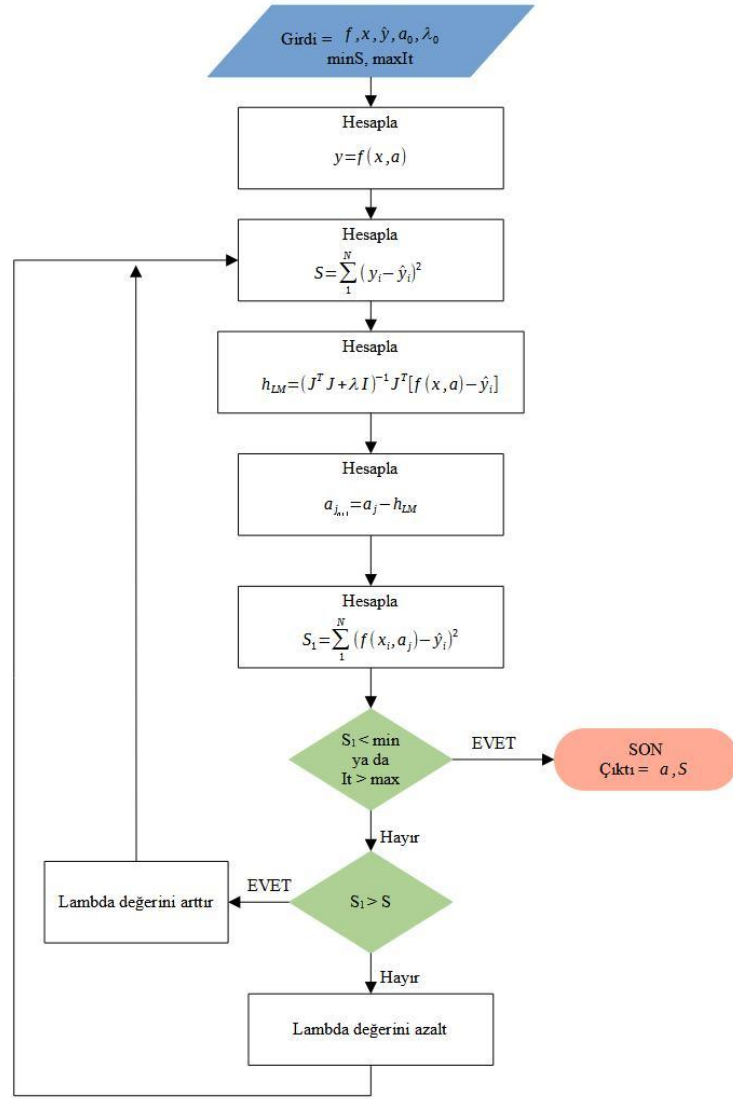
Algoritma 2 Ölçekli Eşlenik Gradyan Algoritması

```
1: Başlangıç ağırlık vektörünü  $w_1$  seçin ve  $d_1 = -g_1$  olsun
2: Hesapla  $\alpha_j, \lambda$  ;
3: Hesapla  $\delta = d_j^T H d_j + \lambda \|d_j\|^2$ 
4: if  $\delta < 0$  then
5:    $\lambda = 2(\lambda - \delta / \|d_j\|^2)$ 
6: end if
7: Hesapla  $\alpha_j = -d_j^T g_j / (d_j^T H d_j + \lambda \|d_j\|^2)$ 
8: Hesapla  $\Delta = \frac{E(w_j) - E(w_j + \alpha_j d_j)}{E(w_j) - E_Q(w_j + \alpha_j d_j)}$ 
9: if  $\Delta > 0.75$  then
10:    $\lambda = \lambda / 2$ 
11: else if  $\Delta < 0.25$  then
12:    $\lambda = 4 \lambda$ 
13: end if
14:  $w_{j+1} = w_j + \alpha_j d_j$ 
15: Değerlendir  $g_{j+1}$ 
16:  $d_{j+1} = -g_{j+1} + \beta_j d_j$ 
17:  $\beta_j = g_{j+1}^T (g_{j+1} - g_j) / g_j^T g_j$ 
18:  $j = j + 1$  ve 2.adıma git
```

Şekil 2.28. SCG algoritması sözde kodu

2.2.3. Levenberg–Marquardt (LM) Algoritması

Levenberg–Marquardt (LM) algoritması doğrusal olmayan optimizasyon problemlerini çözmek için geliştirilmiş yinelemeli bir öğrenme algoritmasıdır. Gradyan inişi ve Newton-Gauss'un bir kombinasyonudur. Özellikle karesi alınmış hataların toplamını alan kayıp fonksiyonlarıyla çalışmak üzere geliştirilmiştir. LM algoritmasının akış diyagramı Şekil 2.29'da verilmiştir.



Şekil 2.29. LM algoritması akış diyagramı

Levenberg-Marquardt yöntemi, parametreler optimal değerlerinden uzak olduğunda daha çok gradyan iniş yöntemi gibi davranır ve parametreler optimal değerlerine yakın olduğunda daha çok Gauss-Newton yöntemi gibi davranır (Shakibjoo vd., 2022).

$$f = \sum_{i=1}^m e_i^2 \quad (2.9)$$

Karesi alınmış hataların toplamı şeklini alan bir kayıp fonksiyonu düşünelim. Burada m eğitim örneklerinin sayısıdır. Kayıp fonksiyonunun Jacobian matrisini parametrelerle ilgili hataların türevlerini içeren matris olarak tanımlayabiliriz,

$$J_{i,j} = \frac{\partial e_i}{\partial w_j} \quad (2.10)$$

$i = 1, \dots, m$ ve $j = 1, \dots, n$. M veri setindeki örnek sayısıdır ve n sinir ağındaki parametre sayısıdır. Jacobian matrisinin boyutu ise $m.n$ 'dir. Kayıp fonksiyonunun gradyan vektörünü şu şekilde hesaplayabiliriz:

$$\nabla f = 2J^T \cdot e \quad (2.11)$$

E değeri tüm hata terimlerinin vektörüdür. Son olarak, Hessian matrisini aşağıdaki ifadeyle yaklaşık olarak hesaplayabiliriz.

$$Hf \approx 2J^T \cdot J + \lambda I \quad (2.12)$$

Burada λ , Hessian'ın pozitifliğini sağlayan bir sönümleme faktörüdür. I ise birim matristir. Sonraki ifade, Levenberg-Marquardt algoritması ile parametre iyileştirme sürecini tanımlar.

$$w^{(i+1)} = w^{(i)} \quad (2.13)$$

$$-(J^{(i)T} \cdot J^{(i)} + \lambda^{(i)} I)^{-1} \cdot (2J^{(i)T} \cdot e^{(i)}), \quad i = 0, 1, \dots \quad (2.14)$$

Sönüm parametresi λ sıfır olduğunda, formül Newton'un yaklaşık Hessian matrisini kullanan yöntemine dönüşmüş olur. λ büyük olduğunda ise bu küçük bir eğitim oranıyla gradyan inişi olur (Wilamowski ve Irwin, 2011). λ parametresi büyük olacak şekilde başlatılır, böylece ilk güncellemeler gradyan iniş yönünde küçük adımlar olur. Herhangi bir iterasyon başarısızlıkla sonuçlanırsa, λ bir faktör tarafından artırılır. Aksi halde kayıp azaldıkça λ azalır, dolayısıyla Levenberg-Marquardt algoritması Newton yöntemine yaklaşır. Bu süreç tipik olarak yakınsamayı minimuma kadar hızlandırır (Çavuşlu vd., 2012). LM algoritmasının sözde kodu Şekil 2.30'da verilmiştir.

Algoritma 3 Levenberg-Marquardt Algoritması

```
1: Başla  $x \in \mathbb{R}^n$ 
2: Başla  $x \in \mathbb{R}\{ \text{sönümleme parametresi} \}$ 
3: while iterasyon < max Iterasyon do
4:    $(J^T J + \lambda I_n)d = -J^T F'$  de  $d$ 'yi çöz
5:   if  $f(x + d) < f(x)$  then
6:      $x = x + d$ 
7:     azalt  $\lambda$ 
8:     iterasyon++
9:   else
10:    arttır  $\lambda$ 
11:   end if
12: end while
13: return  $x$ 
```

Şekil 2.30. LM algoritması sözde kodu

LM algoritmasının bazı dezavantajları da vardır. Bunlardan ilki, çapraz entropi veya MSE gibi hataları en aza indirememesidir. Ayrıca Jacobian matrisi, büyük veri kümeleri ve sinir ağları için yüksek boyutlara ulaşır ve çok fazla bellek gerektirir. Bu nedenle, büyük veri kümeleri veya sinir ağları söz konusu olduğunda LM algoritması önerilmez.

2.3. Metasezgisel Optimizasyon Algoritmaları

Optimizasyon bir problem için bulunabilecek en iyi sonuç olarak tanımlanmaktadır. Kelime anlamı olarak optimizasyon en iyilemedir. Belirli bir probleme ait sınırlar içerisinde istenilen parametrelerin değerlerinin bulunmasına ise optimizasyon problemi denilmektedir. Metasezgisel algoritmalar optimizasyon problemlerinin optimal ya da yaklaşık optimal çözümünü elde etmek için geliştirilmiş olan optimizasyon yöntemleri olarak tanımlanır. Bu algoritmaların optimizasyon problemlerinde sıklıkla tercih edilmesinin nedeni türev hesabı gerektirmemeleri, basit ve esnek yapıda olmaları, ve yerel optimumlardan kaçınma yetenekleridir (Mirjalili vd., 2014).

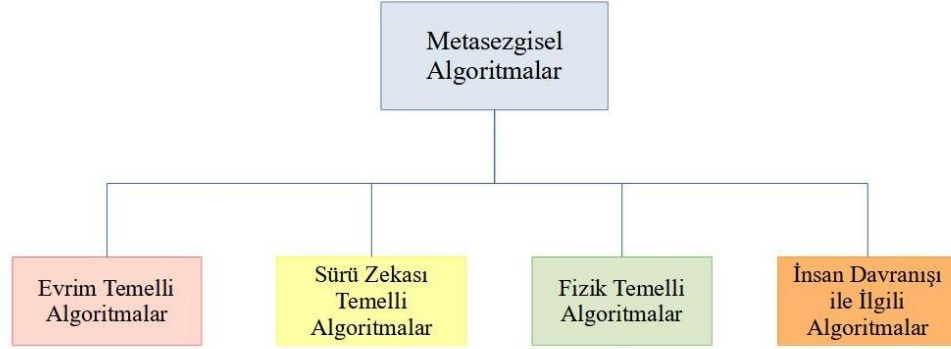
Metasezgisel algoritmaların keşif ve sömürü olmak üzere iki temel bileşeni mevcuttur ve iki bileşen arasında bir denge sağlar (Olorunda ve Engelbrecht, 2008). Bu sayede en iyi veya en iyiye yakın çözüm arayışında arama uzayında etkili bir şekilde arama yapar. Keşif ile arama uzayının keşfinde çeşitli çözümler üretirken sömürü ile mevcut en iyi çözüm ile yerel bölgede aramaya odaklanır. Bu iki bileşenin kombinasyonu problemin en iyi çözümünü bulmada önemlidir. Elektrik

mühendisliği, endüstriyel alanlar, inşaat mühendisliği, iletişim, veri madenciliği vb. gibi çeşitli mühendislik ve bilim problemlerinde metasezgisel algoritmalar başarıyla uygulanmaktadır (Agrawal vd., 2021).

Metasezgisel algoritmalar, literatürde yer alan farklı algoritmaların performansını geliştirmek veya farklı problemlere uygun hale getirmek amacıyla tasarlanır. Metasezgisel algoritmaların etkinliği, No Free Lunch (NFL) teoremi tarafından vurgulanan önemli bir kavramı göz önüne sermektedir (Wolpert ve Macready, 1997). No Free Lunch teoremi, herhangi bir optimizasyon algoritmasının tüm problemler için en iyi performansı sağlayamayacağını, yani evrensel olarak en iyi algoritmanın mevcut olmadığını belirtir (Korkmaz, 2019). Bu nedenle, bir optimizasyon problemi için ideal bir algoritma tasarlarlarken veya mevcut bir algoritmayı iyileştirirken, problemi ve problem özelliklerini dikkate almak çok önemlidir. Metasezgisel algoritmalar, bu bağlamda önemli bir rol oynar, çünkü farklı problemlere uygun olarak uyarlanabilirler ve bazı problemlerde daha etkili olabilirler.

Metasezgisel algoritmalar tek çözüme dayalı algoritmalar ve çoklu çözüme dayalı (popülasyon tabanlı) algoritmalar olmak üzere iki ana kategoriye ayrılır. Tek çözüme dayalı algoritmalar optimizasyon sürecine bir çözümle başlar ve iterasyonlar süresince sürekli olarak bu çözümleri günceller. Bu nedenle arama alanını tam olarak keşfedemeyebilir ve yerel optimuma hapsolabilir. Çoklu çözüme dayalı algoritmalar ise başlangıçta bir çözüm popülasyonu oluşturarak optimizasyon sürecine başlar ve nesil/iterasyon sayısı ile birlikte çözüm popülasyonu güncellenir. Birden fazla çözümün birbirine yardımcı olması ve bu sayede arama uzayının büyük bir bölümünün keşfedilmesi yerel optimumlardan kaçınmak için faydalıdır. Ayrıca optimum arayışında arama uzayının umut vaat eden bölümüne atlama özelliklerinin olması popülasyon tabanlı algoritmaların karmaşık gerçek dünya problemlerinde tercih edilmesinin ana sebeplerindendir.

Metasezgisel algoritmalar literatürde farklı çeşitli sınıflandırılmaktadır. Bir metasezgisel algoritmanın birden fazla sınıfa dâhil olması mümkündür. Davranışlarına göre metasezgisel algoritmalar evrim temelli, sürü zekâsı temelli, fizik temelli ve insan ilişkili olmak üzere dört ana kategoriye ayrılabilir.



Şekil 2.31. Metasezgisel algoritmaların kategorik bazda sınıflandırılması

2.3.1. Evrim Temelli Algoritmalar

Evrin temelli algoritmalar biyolojiden esinlenen ve doğal evrimden ilham alınarak geliştirilen algoritmalarlardır. Darwin'in doğal seleksiyon teorisine göre yani ortama en uygun olan bireyin hayatta kalmasına dayanır (Holland, 1992). Süreç rastgele oluşturulan çözüm popülasyonu ile başlatılır ve en iyi çözümler bir araya getirilerek yeni bireyler oluşturulur. Ardışık nesiller boyunca arama süreci devam eder ve optimum çözüm bulunur. Genetik algoritma (Holland, 1992), tabu arama (Glover, 1986), diferansiyel evrim (Storn ve Price, 1997) ve biyocoğrafya tabanlı optimizier (Ma vd., 2017) popüler evrimsel temelli algoritmalarından bazılarına örnek olarak verilebilir.

2.3.2. Sürü Zekası Temelli Algoritmalar

Sürü zekâsı terimi ilk kez Beni ve Wang tarafından hücrel robotik sistem sınıfını tanımlamak amacıyla tanıtılmıştır (Beni ve Wang, 1989). Sürü zekâsı temelli algoritmalar sosyal hayvanların/böceklerin kolektif davranışlarından ilham alınarak geliştirilmiş metasezgisel algoritmalar ve sosyal yaşamlı varlıkların sürüler şeklinde karmaşık görevleri çözme yeteneklerine dayanır. Karmaşık görevlerde aynı görevin bireysel düzeyde çözülmesi imkânsız bir durumdur. Tek bir karınca, arı veya kuş ve balıkların bireysel olarak çok sınırlı bir zekaya sahip olduğu görülür, ancak birbirleriyle ve çevreleriyle sosyal olarak etkileşime geçtiklerinde, yiyecek kaynağına en kısa yolun bulunması, yuvalarını organize etme, hareketlerini senkronize etme ve

yüksek hızda tek bir uyumlu varlık olarak seyahat etme gibi zorlu görevleri başarabildikleri görülmektedir (Kolias vd., 2011). Bu nedenle sürü zekâsı temelli algoritmalar pek çok karmaşık gerçek yaşam problemlerinin optimizasyonunda kullanılmaktadır. Bu algoritmalar arasında karınca kolonisi optimizasyonu (Dorigo ve Blum, 2005), parçacık sürü optimizasyonu (Hu vd., 2004), yapay arı kolonisi (Karaboğa ve Akay, 2009), balık sürü algoritması (Li, 2002), aslan algoritması (Rajakumar, 2012), balina optimizasyon algoritması (Tripathi vd., 2021), yusufoçuk optimizasyonu (Bhardwaj ve Kim, 2021), arı sürü optimizasyonu (Bitam vd., 2018), koronavirüs optimizasyonu (Martínez-Álvarez vd., 2020), at sürü optimizasyon algoritması (MiarNaeimi vd., 2021) ve yapay sinek kuşu algoritması (Zhao vd., 2022) popüler sürü zekâsı temelli metasezgisel algoritmalarla örnek olarak verilebilir.

2.3.3. Fizik Temelli Algoritmalar

Fizik temelli yöntemler, evrende bulunan fizik yasalarından örneğin maddelerin soğuması ya da ısınması, Newton'un yer çekimi yasası gibi kurallardan ilham alınarak geliştirilmiş algoritmalar (Kirkpatrick vd., 1983). Bu algoritmalar, fiziksel sistemlerin davranışlarını matematiksel modellerle taklit ederek çözüm alanında arama yaparlar ve bir başlangıç noktası stratejileri yoktur. Kara delik algoritması (Hatamlou, 2013), yerçekimi tabanlı yerel arama algoritması (Rashedi vd., 2009), elektron arama algoritması (Tabari ve Ahmad, 2017), uyum arama (Geem vd., 2001), karıştırılmış kurbağa sıçraması algoritması (Eusuff vd., 2006) ve simüle edilmiş tavlama algoritması (Kirkpatrick vd., 1983) fiziksel temelli metasezgisel algoritmalarla bazılarıdır.

2.3.4. İnsan Davranışlarıyla İlgili Algoritmalar

Bu algoritmalar insan davranışı ve düşünme süreçlerinden ilham alınarak geliştirilmiş olan algoritmalar (Mohamed vd., 2020). İnsanların problem çözme yeteneklerini taklit eder ve deneyimlerden faydalanan yaklaşımları kullanır. İnsanların problemi anlama, bilgiyi işleme, deneme ve yanılma yapma, öğrenme ve optimize etme gibi aşamalarını matematiksel olarak modellemeye çalışır. İnsan davranışlarıyla ilgili metasezgisel algoritmalar iteratif süreç, bilgi paylaşımı ve deneyim kullanımı özelliklerine sahiptir. Bu algoritmalar insana benzer çözümler ürettiği için gerçek dünya problemlerinde ve karmaşık karar verme süreçlerinde kullanılmak üzere tasarlanmıştır. Öğretme-Öğrenme Temelli Optimizasyon (TLBO) (Rao vd., 2012), Lig Şampiyonası algoritması (Kashan, 2009) vb. algoritmalar insan

davranışlarıyla ilgili algoritmalarından bazılarıdır.

2.4. COOT Optimizasyon Algoritması

Naruei ve Keynia tarafından önerilmiş yeni bir metasezgisel yöntem olan COOT optimizasyon algoritması adını coot isimli bir kuş sürüsünden almaktadır ve algoritma kuşların su üzerinde yapmış oldukları hareketlerden ilham alınarak geliştirilmiştir (Naruei ve Keynia, 2021).

Coot'lar, Rallidae adlı su tavuğu ailesinin üyeleri olan küçük su kuşlarıdır. Adı Latince “coot” olan Fulica cinsini oluştururlar (Paillisson ve Marion, 2001). Türkçe’de bu kuşlar “sakarmek” olarak bilinmektedir. Cootların alnında belirgin ön kalkanları ya da farklı süslemeleri, kırmızıdan koyu kırmızıya gözleri ve renkli gagaları mevcuttur. Hepsi olmasa da birçoğunun kuyruğunun altında beyaz vardır.

COOT optimizasyon algoritması cootların yiyecek arayışında su üzerinde yaptıkları hareketleri baz alarak optimal alana ulaşmalarını sağlayan sistemi modellemektedir. Algoritmada su yüzeyindeki dört farklı coot hareketi;

1. Rastgele hareket
2. Zincir hareketi
3. Grup liderlerine göre pozisyonun ayarlanması
4. Grubun liderler tarafından optimal alana doğru yönlendirilmesi (lider hareketi) olarak ele alınmaktadır.

2.4.1. Matematiksel Model ve Algoritma

Algoritma bir başlangıç rastgele popülasyonu $(\vec{x}) = \{ \vec{x}_1, \vec{x}_2, \dots, \vec{x}_n \}$ ile başlar. Bu rastgele popülasyon, hedef fonksiyon tarafından sürekli olarak değerlendirilir ve $(\vec{O}) = \{ \vec{O}_1, \vec{O}_2, \dots, \vec{O}_n \}$ olacak şekilde hedef değer belirlenir.

Popülasyon tabanlı optimizasyon teknikleri optimal optimizasyon problemlerini aradığından, tek seferde bir çözüm bulma garantisi yoktur. Ancak yeterli sayıda rastgele çözüm ve optimizasyon adımı (yineleme) ile global optimali bulma olasılığı artar.

Popülasyon, aşağıda belirtilen Denklem 2.15 kullanılarak görüş alanında rastgele oluşturulur.

$$\text{CootPos}(i) = \text{rand}(1, d) * (ub - lb) + lb \quad (2.15)$$

$\text{CootPos}(i) \rightarrow$ coot konumu, $d \rightarrow$ değişkenlerin veya problem boyutlarının sayısı, Denklem 2.16'da tanımlanmış değerler olan $lb \rightarrow$ arama uzayının alt sınırı ve $ub \rightarrow$ arama uzayının üst sınırıdır. Her değişkenin farklı bir alt sınır ve üst sınır sorunu olabilir.

$$lb = [lb_1, lb_2, \dots, lb_d], ub = [ub_1, ub_2, \dots, ub_d] \quad (2.16)$$

Başlangıç popülasyonu oluşturulduktan ve her cootun konumu belirlendikten sonra, amaç fonksiyonu $O_i = f(\vec{x})$ kullanılarak her çözümün uygunluğu hesaplanmalıdır. NL (lider sayısı) değeri cootların grup lider sayısı olarak belirlenir ve lider seçimi rastgele olacak şekilde yapılır.

2.4.1.1. Rastgele Hareket

Bu hareketi uygulamak için, arama uzayında Denklem 2.17'ye göre rastgele bir konum düşünüyoruz ve cootu bu rastgele konuma doğru hareket ettiriyoruz.

$$Q = \text{rand}(1, d) * (ub - lb) + lb \quad (2.17)$$

Bu coot hareketi, arama uzayının farklı bölümlerini araştırır. Algoritma yerel optimalde takılıp kalırsa, bu hareket algoritmanın yerel optimalden kaçmasına neden olacaktır. Coot'un yeni konumu Denklem 2.18'e göre hesaplanır.

$$\text{CootPos}(i) = \text{CootPos}(i) + A \times R2 \times (Q - \text{CootPos}(i)) \quad (2.18)$$

$R2 \rightarrow [0, 1]$ aralığında rastgele bir sayıdır ve A değeri Denklem 2.19'a göre hesaplanır.

$$A = 1 - L \times \left(\frac{1}{\text{Iter}}\right) \quad (2.19)$$

$L \rightarrow$ geçerli iterasyon ve $\text{Iter} \rightarrow$ maksimum iterasyon

2.4.1.2. Zincir Hareketi

Bir zincir hareketi, önce iki kuyruk arasındaki mesafe vektörünü hesaplamamız ve ardından kuyruğu mesafe vektörünün yaklaşık yarısı kadar diğer kümeye doğru hareket ettirmemiz ile gerçekleştirilebilir. Algoritmada zincir hareketi iki cootun ortalama konumu bulunarak uygulanmaktadır. Cootun yeni konumu Denklem 2.20'ye göre hesaplanır.

$$\text{CootPos}(i) = 0.5 \times (\text{CootPos}(i - 1) + \text{CootPos}(i)) \quad (2.20)$$

CootPos(i-1) değeri ikinci cootu ifade eder.

2.4.1.3. Grup Liderlerine Göre Pozisyon Ayarlanması

Genellikle grup, grubun önündeki birkaç kişi tarafından yönetilir ve geri kalanı grup liderlerine göre konumlarını ayarlamak ve onlara doğru hareket etmek zorundadır. Burada merak edilen her bir cootun konumunu hangi lidere göre ayarlayacağıdır. Liderlerin ortalama konumu dikkate alınabilir ve cootlar bu ortalama konuma göre konumlarını güncelleyebilir. Fakat ortalama konumu dikkate almak erken yakınsamaya neden olur. Lider seçmek için kullanmış olacağımız formül Denklem 2.21’de belirtilmiştir.

$$K = 1 + (i \text{ MOD } NL) \quad (2.21)$$

$i \rightarrow$ Mevcut coot'un indeks numarası, $NL \rightarrow$ lider sayısı ve $K \rightarrow$ liderin indeks numarasıdır. Coot(i), liderin k'sine göre pozisyonunu güncellemelidir. Denklem 2.22, seçilen lidere dayalı olarak cootun bir sonraki konumunu hesaplar.

$$\begin{aligned} \text{CootPos}(i) = & \text{LeaderPos}(k) + 2 \times R1 \\ & \times \cos(2R\pi) \times (\text{LeaderPos}(k) - \text{CootPos}(i)) \end{aligned} \quad (2.22)$$

$\text{CootPos}(i) \rightarrow$ cootun mevcut konumu,

$\text{LeaderPos}(k) \rightarrow$ lider konumu,

$R1 \rightarrow [0,1]$ aralığında rastgele bir sayı,

$\pi \rightarrow 3.14$,

$R \rightarrow [-1,1]$ aralığında rastgele bir sayıdır.

2.4.1.4. Lider Hareketi

Grup bir hedefe (optimal alan) yönelik olmalıdır, bu nedenle liderlerin hedefe yönelik konumlarını güncellemeleri gerekir. Liderlerin konumunu güncellemek için Denklem 2.23 önerilir. Bu denklem, mevcut optimal nokta etrafında daha iyi konumlar arar. Bazen liderler daha iyi pozisyonlar bulmak için mevcut optimal pozisyonundan uzaklaşmak zorunda kalırlar. Denklem, optimum konuma yaklaşmak ve ondan uzaklaşmak için iyi bir yol sağlar.

$$\begin{aligned} & \text{LeaderPos}(i) \\ = & \begin{cases} B \times R3 \times \cos(2R\pi) \times (gBest - \text{LeaderPos}(i)) + gBest, & R4 < 0.5 \\ B \times R3 \times \cos(2R\pi) \times (gBest - \text{LeaderPos}(i)) - gBest, & R4 \geq 0.5 \end{cases} \end{aligned} \quad (2.23)$$

$gBest \rightarrow$ bulunmuş en iyi konum, $R3$ ve $R4 \rightarrow [0,1]$ aralığında rastgele sayılar, $\pi \rightarrow 3.14$, $R \rightarrow [-1,1]$ değer aralığında rastgele sayı ve B değeri aşağıda verilen Denklem 2.24 ile hesaplanmaktadır.

$$B = 2 - L \times \left(\frac{1}{Iter}\right) \quad (2.24)$$

$L \rightarrow$ geçerli iterasyon

$Iter \rightarrow$ maksimum iterasyon

$2 \times R3 \rightarrow$ algoritmanın yerel optimumda sıkışmaması için daha büyük rastgele hareketler yapar. Bu, sömürü aşamasında da keşif yapabildiğimizi gösterir. $\cos(2R\pi) \rightarrow$ Arama ajanı çevresinde daha iyi bir konum bulmak için farklı yarıçapa sahip en iyi arama ajanının etrafında arama yapar.

Burada ortaya çıkan soru, bu farklı hareketlerin nasıl ve ne zaman gerçekleştirildiğidir. Optimizasyon algoritmalarının rastgele doğasını korumak için tüm bu hareketler rastgele olarak ele alınmaktadır. Bu, algoritmanın yürütülmesi sırasında, coot'un rastgele, zincir halinde veya grup liderlerine doğru hareket edebileceği anlamına gelir.

2.4.2. Algoritma Akış Diyagramı ve Sözde Kodu

COOT algoritması sözde kodu Şekil 2.32.'de ve akış diyagramı Şekil 2.33.'te verilmiştir.

Algoritma 4 COOT Optimizasyon Algoritması

- 1: İlk coot popülasyonunu Denklem 2.15 ve Denklem 2.16 ile rastgele başlat
- 2: $P=0.5$, NL (lider sayısı), N_{coot} (coot sayısı), $N_{coot}=N_{pop}-NL$ parametrelerini başlat
- 3: Cootlardan rastgele lider seçimi
- 4: Cootların ve liderlerin uygunluğunu hesapla
- 5: Global optimum ($gBest$) olarak en iyi cootu veya lideri bulun
- 6: **while** sonlandırma kriteri karşılanmıyorsa **do**
- 7: A, B parametrelerini Denklem 2.19 ve Denklem 2.24 ile hesapla
- 8: **if** $rand < P$ **then**
- 9: $R, R1$ ve $R3$ problemin boyutları boyunca rastgele vektörlerdir
- 10: **else**
- 11: $R, R1$ ve $R3$ rastgele sayılardır
- 12: **end if**
- 13: **for** $i=1$, cootların sayısına kadar **do**
- 14: K parametresini Denklem 2.21 ile hesapla
- 15: **if** $rand > 0.5$ **then**
- 16: Coot'un konumunu Denklem 2.22 ile güncelle
- 17: **else if** $rand < 0.5$ **1 then**
- 18: Coot'un konumunu Denklem 2.20 ile güncelle

sınıflandırma, optimizasyon, fonksiyon yaklaşımları gibi farklı problemlere başarıyla uygulanan yöntemdir. Mimari tasarımı, eğitim algoritması ve eğitim parametreleri YSA'ların başarısını büyük ölçüde etkilemektedir. Bu nedenle YSA tasarımı zor bir optimizasyon problemi haline gelmektedir. Araştırmacılar literatürde YSA eğitiminde metasezgisel algoritmaları YSA'nın ağırlık ve bias değerlerinin optimize edilmesinde, YSA için uygun bir mimari tasarımının yapılmasında ve öğrenme hızı, momentum gibi parametre ayarlamada kullanmaktadırlar (Türkoğlu, 2019). Bu tez çalışmasında COOT optimizasyon algoritması YSA'nın ağırlık ve bias değerlerinin optimizasyon problemi için kullanılmıştır.

2.5.1. Önerilen Model (COOT-ANN)

YSA'ların doğru sonuçlar üretmesi ve sınıflandırmaları başarılı yapabilmesi için ağırlık ve bias değerlerinin en uygun şekilde güncellenmesi gereklidir. Bu güncelleme işlemi için araştırmacılar tarafından literatürde birçok algoritma önerilmiştir. Önerilen algoritmalar bir YSA'nın eğitimi için gerekli parametreleri optimize etmekte yetersiz kalmakta ve problemle karşılaşmaktadır. Klasik teknikler yüksek miktarda bilgi işlem süresine ve belleğe ihtiyaç duyarlar. Bununla birlikte süreç içerisinde yerel optimumlara takılarak kalitesiz çözümler üretirler. Tüm bu sorunları ortadan kaldırmak ve eğitimde gerekli optimal parametreleri bulabilmek için metasezgisel algoritmalar kullanılarak YSA'lar eğitilmiştir. Bu tez çalışmasında ağırlık ve bias değerlerinin optimizasyonu için hibrit COOT-ANN algoritması önerilmiştir. Önerilen modelin sözde kodu Şekil 2.34'de verilmiştir.

Algoritma 5 Önerilen COOT-ANN Modeli

```

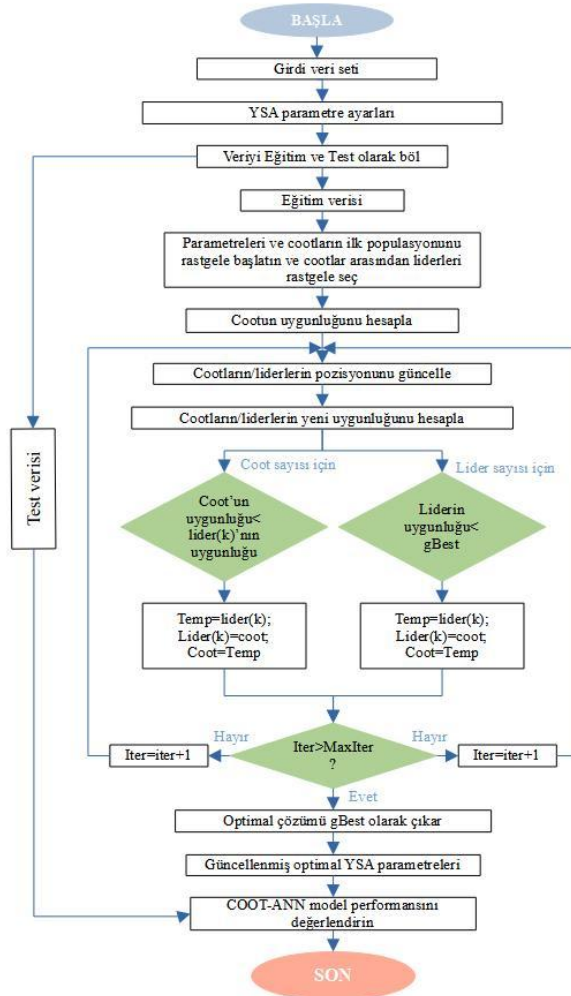
1: Girdi: birey sayısı (N) ve toplam iterasyon sayısı ( $t_{max}$ ).
2: Çok katmanlı sinir ağını ve N birey X kümesini oluştur
3: Verileri eğitim ve doğrulama bölümlerine ayır
4:  $t \leftarrow 1$ 
5: while  $t \leq t_{max}$  do
6:   Her bir bireyin  $X_i$  kalitesini hesapla
7:   En iyi çözümü bul,  $X_{best}$ 
8:   COOT operatörlerini kullanan diğer bireyleri güncelle
9:    $t \leftarrow t+1$ 
10: end while
11: return  $X_{best}$ 
12: COOT-ANN modeli performansını değerlendir

```

Şekil 2.34. Önerilen modelin sözde kodu

Tez çalışmasında öncelikle geliştirilen hibrit model için bir YSA

oluşturulmuştur. Şarap, göğüs kanseri, iris ve cam veri setleri ile k-katlamalı doğrulama yöntemi kullanılmıştır. Sonraki adımda, yapılandırmaların kalitesi, CE kullanılarak eğitim ve test verileri üzerinde bir uyum fonksiyonu olarak hesaplanmıştır. Sürecin devamında, COOT algoritmasının adımları uygulanmış ve parametreler güncellenmiştir. Öğrenme süreci sırasında elde edilen en iyi yapılandırmayla model performansı değerlendirilmiştir. Önerilen COOT-ANN modelinin akış diyagramı Şekil 2.35'te verilmiştir.



Şekil 2.35. Önerilen modelin akış diyagramı

2.6. Video Sınıflandırma

Video sınıflandırma, büyük miktarda video içeriğini özel kategorilere, sınıflara veya etiketlere atama sürecini ifade eder. Bu süreç, videoların içeriğini anlama ve benzer türdeki videoları gruplama amacı taşır. Genellikle görsel, metin veya ses

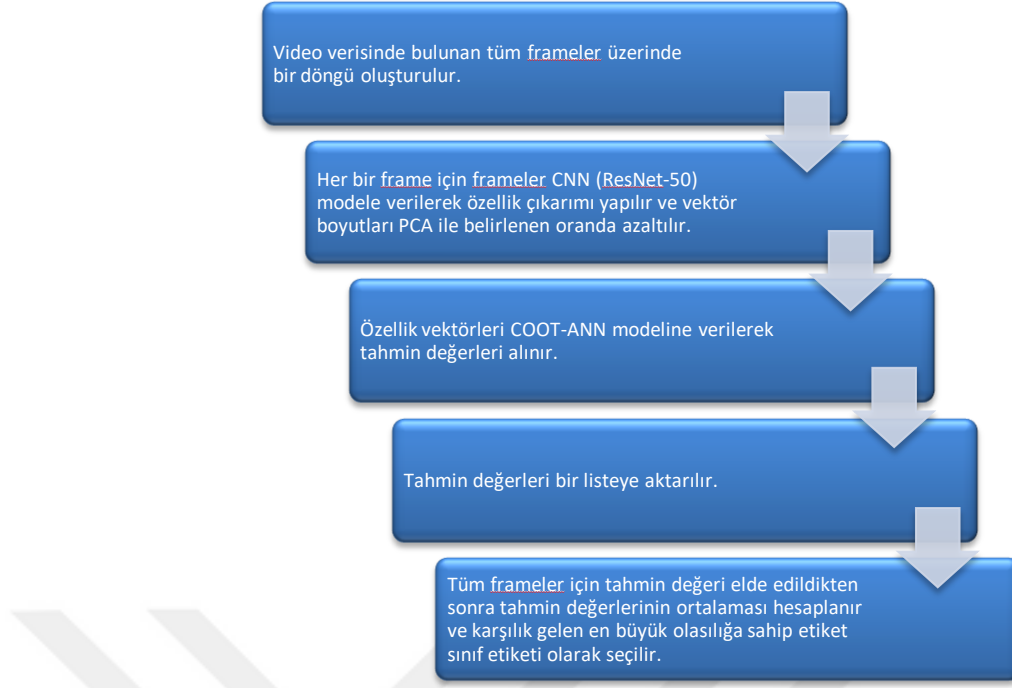
özelliklerini kullanarak bir videoyu tanımlama ve sınıflandırma işlemini içerir. Video sınıflandırma, videoların etiketlenmesi, arama motorlarına indekslenmesi, içerik önerilerinin sunulması ve güvenlik uygulamalarında anormalliklerin tespit edilmesi gibi birçok farklı uygulama alanında kullanılır.

Video sınıflandırma için çeşitli yaklaşımlar bulunmaktadır. Metin tabanlı yöntemler, ses tabanlı yöntemler ve video tabanlı yöntemler gibi. Günümüzde ise ses ve görüntü özelliklerinin birleştirildiği kombinasyonel yaklaşımlar da kullanılmaktadır. Metin tabanlı yöntemler, giriş video dosyasının içeriğini metin bilgisi kullanarak sınıflandırır. Ses tabanlı yaklaşım, giriş video dosyasının ses kısmını kullanarak sınıflandırma yapar ve sistemi ses bilgilerine dayandırır. Video tabanlı yöntem ise giriş video dosyasının görsel özelliklerini çıkararak hangi sınıfa ait olduğunu belirler (Rani vd., 2020).

2.6.1. COOT-ANN Modelinin Video Sınıflandırmada Kullanımı

Birçok video sınıflandırma modeli, veri setlerinde bulunan görsel özellikleri kullanarak eğitilir. Özellik çıkarımı aşamasında, derin öğrenme mimarileri gibi güçlü araçlar, video framelerini temsil eden yüksek boyutlu vektörler oluşturur. Bu bölümde, tez çalışmasında sınıflandırma işlemi için geliştirilen COOT-ANN modelinin, video sınıflandırma problemi üzerindeki başarımı incelenmiştir..

Çalışma kapsamında ResNet50 tabanlı bir özellik çıkarım ağı kullanılarak PCA ile boyut azaltma işlemi uygulanmış, elde edilen özellik vektörleri COOT-ANN modeline verilerek model eğitimi ve devamında test işlemleri gerçekleştirilmiştir. Şekil 2.26'da önerilen video sınıflandırma yöntemi akış diyagramı görülmektedir.



Şekil 2.36. COOT-ANN modeli ile video sınıflandırma işlemi

2.7. Performans Değerlendirme Ölçütleri

Bir sınıflandırma veya regresyon problemi için geliştirilmiş olan modellerden hangisinin daha doğru sonuçlar ürettiğini anlayabilmemiz için bazı performans değerlendirme metrikleri kullanılmaktadır. Makine öğrenimi ve istatistiksel sınıflandırmada karmaşıklık matrisi adı verilen tablo kullanılarak performans değerlendirme metrikleri hesaplanmaktadır. Matriste satırlar gerçek değerleri ifade ederken, sütunlar tahmin edilen değerleri ifade etmektedir (Alan, 2020). İki sınıflı bir karmaşıklık matrisi ve alacağı değerler Tablo 2.2’de verilmiştir.

Tablo 2.2. İki sınıflı karmaşıklık matrisi

		Tahmini Sınıf		
		Sınıf=Evet	Sınıf=Hayır	Toplam
Gerçek Sınıf	Sınıf=Evet	TP	FN	P
	Sınıf=Hayır	FP	TN	N
	Toplam	P_1	N_1	P+N

Birçok değerlendirme ölçütünde kullanılan bazı terimler aşağıda belirtilmiştir.

- Doğru pozitif (TP): Sınıflandırma modelinin doğru şekilde etiketlendirdiği pozitif veri sayısını ifade eder.
- Doğru Negatif (TN): Sınıflandırma modelinin doğru şekilde

etiketlendirdiği negatif veri sayısını ifade eder.

- Yanlış Pozitif (FP): Gerçekte negatif veri olarak etiketlenen fakat sınıflandırma modelinin pozitif olarak tahmin ettiği veriyi ifade eder.
- Yanlış Negatif (FN): Gerçekte pozitif veri olarak etiketlenen fakat sınıflandırma modelinin negatif olarak tahmin ettiği veriyi ifade eder.

Karmaşıklık matrisi kullanılarak sınıflandırma modelinin performansını ölçmek için kullanılan bazı ölçütler; doğruluk, duyarlılık, anma, kesinlik, özgüllük, F ölçütü, AUC ve ROC eğrisi metrikleridir.

2.7.1. Doğruluk Oranı

Doğruluk, eğitim verileri ile oluşturulmuş olan bir modelin test verilerini doğru sınıflandırma oranıdır. Doğru tahmin edilerek sınıflandırılan veri sayısının toplam veri sayısına bölünmesi ile hesaplanır. Doğruluk değerinin formülü Denklem 2.25'te belirtilen denklem ile hesaplanmaktadır.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.25)$$

2.7.2. Çapraz Entropi

Entropi bir veri kaynağındaki belirsizlik miktarını ifade eden bit sayısıdır. Entropi H sembolüyle gösterilmektedir ve bir bilginin bir ortamdaki düzensizliğini belirlemek için kullanılır. Çapraz entropi ise iki durum arasındaki olayın düzensizlik temelinde değerlendirilmesidir (Japkowicz ve Shah, 2011). Çapraz Entropi Denklem 2.26 ile gösterilmektedir.

$$H(p, q) = - \sum_i p(x_i) \log(q(x_i)) \quad (2.26)$$

Formülde belirtilen p değeri hedef dağılımı, r değeri ise tahmini dağılımı ifade etmektedir. Çapraz entropi formülü ile bu iki dağılımdaki her olayın olasılık değerlerinin birbirine yakınlık derecesi belirlenmiş olur. Çapraz entropi simetrik bir metrik olmadığı için $H(p, q)$ ve $H(q, p)$ değerleri aynı olmayabilir.

2.7.3. F1 Skoru

F1 puanı kesinlik ve anma değerlerinin harmonik ortalaması alınarak elde edilir. Kesinlik değeri sınıflandırmada pozitif tahminlerin ne oranda doğru tahmin

edildiğidir ve Denklem 2.27 ile hesaplanmaktadır.

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (2.27)$$

Anma değeri ise sınıflandırma modelinin pozitif sınıfa ait olan verileri doğru tahmin etme oranıdır. Aşağıda verilen Denklem 2.28 ile hesaplanır.

$$\text{Anma} = \frac{TP}{TP+FN} \quad (2.28)$$

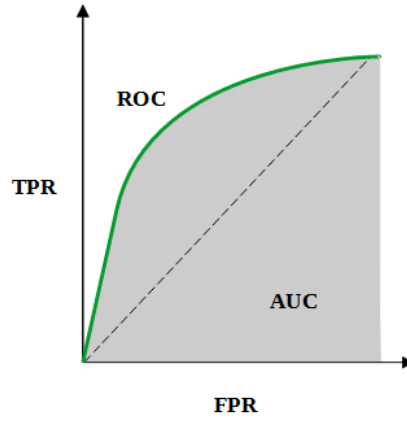
Kesinlik ve anma ölçütleri model performans değerlendirmesi sırasında anlamlı bir karşılaştırma yapabilmemiz için yeterli değildir. Bu nedenle iki ölçüt kullanılarak elde edilen F1 ölçütü daha anlamlı değerlendirme yapabilmemizi sağlar. F1 ölçütü formülü Denklem 2.29’da verilmiştir.

$$\text{F1skoru} = 2 \frac{\text{Kesinlik} * \text{Anma}}{\text{Kesinlik} + \text{Anma}} \quad (2.29)$$

2.7.4. ROC-AUC Eğrisi

AUC (Eğrinin Altındaki Alan) ROC (Alıcı Çalışma Karakteristikleri) eğrisi bir sınıflandırma probleminin performansı kontrol edilmek istendiğinde veya görselleştirmek gerektiğinde kullanılmaktadır. ROC-AUC herhangi bir sınıflandırma modelinin performansını kontrol etmede kullanılan en önemli değerlendirme metriklerinden biridir. AUROC (Alıcı Çalışma Karakteristiklerinin Altındaki Alan) olarak da yazılır.

ROC bir olasılık eğrisidir ve AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil eder. Modelin sınıfları ne kadar ayırt edebildiğini söyler. AUC ne kadar yüksekse, model sınıfları doğru bir şekilde tahmin etmede o kadar iyidir. AUC değeri düştükçe modelin doğru çalışmadığı tespit edilmiş olur.



Şekil 2.37. ROC eğrisi ve AUC

Şekilde belirtilen True Positive Rate (TPR) değeri Denklem 2.28’de açıklanan anma değeridir.

False Positive Rate (FPR) ile belirtilen değer sınıflandırma sonucunda modelin pozitif sınıfta etiketlediği fakat gerçekte negatif sınıfa ait FP adet verinin tüm negatif sınıfa bölümüdür.

$$FPR = \frac{FP}{TN+FP} \quad (2.30)$$

Ayrıca FPR değeri özgüllük değerinin 1’e tümleyenidir. Özgüllük değeri ise sınıflandırma modelinin negatif sınıf etiketlerini tahmin etmedeki etkililiğidir.

$$FPR = 1 - \text{Özgüllük} \quad (2.31)$$

$$\text{Özgüllük} = \frac{TN}{TN+FP} \quad (2.32)$$

2.7.5. Cohen’s Kappa Metriği

Cohen’s Kappa istatistiği Amerikalı psikolog ve istatistikçi Jacob Cohen tarafından 1960 yılında psikoloji alanında iki puanlayıcı arasındaki güvenilirliğin ölçüsü olarak geliştirilmiş (Cohen, 1960), daha sonra ise literatürde sınıflandırma problemlerinde performans ölçütü olarak kullanılmaya başlanmıştır. Cohen’s Kappa metriği sınıflandırma çalışmalarında test veri setindeki durumlar için gözlemlenen ve tahmin edilen sınıflar arasındaki uyuşmanın bir ölçüsü olarak tanımlanmaktadır

(Delgado ve Tibau, 2019).

Tablo 2.3. İki puanlayıcının k-kategori ölçeğindeki değerlendirmelerinin ortak olasılıkları

Tahminci A	Tahminci B				Toplam
	1	2	...	k	
1	p_{11}	p_{12}	...	p_{1k}	$p_{1.}$
2	p_{21}	p_{22}	...	p_{2k}	$p_{2.}$
.	.	.	...	.	.
.	.	.	...	.	.
k	p_{k1}	p_{k2}	...	p_{kk}	$p_{k.}$
Toplam	$p_{.1}$	$p_{.2}$	...	$p_{.k}$	1

Doğruluk ve CE değerleri sınıflandırıcı performansının değerlendirilmesinde yeterli olmayabilir. Bu nedenle Cohen's Kappa ölçütü performans karşılaştırmasında daha doğru veriler elde etmemize yardımcı olur. Cohen's Kappa denklemi Denklem 2.33'te belirtilmiştir.

$$K = \frac{P_o - P_e}{1 - P_e} \quad (2.33)$$

$$M = TP + FN + FP + TN \quad (2.34)$$

$$P_o = \frac{TP + TN}{M} \quad (2.35)$$

$$d = \frac{(TN + FP)}{M} \times \frac{(TN + FN)}{M} \quad (2.36)$$

$$y = \frac{(FN + TP)}{M} \times \frac{(FP + TP)}{M} \quad (2.37)$$

$$p_e = d + y \quad (2.38)$$

Denklemden P_o değeri gözlenen uyumluluk oranı ve P_e rastgele beklenen uyumluluk oranıdır.

3. UYGULAMA VE BULGULAR

Bu bölümde çalışmada kullanılan veri setleri ve çalışma ortamı hakkında bilgiler, önerilen model olan COOT-ANN modelinin veri setleri üzerinde yapılan çalışmalar sonucunda performans değerleri ve diğer algoritmalar ile elde edilen sonuçların karşılaştırmalı olarak açıklaması verilmiştir.

3.1. Veri Kümeleri

Önerilen modelin performans değerlendirmesi için 4 farklı sınıflandırma veri kümesi kullanılmıştır. Deneyler için kullanılan şarap, göğüs kanseri, cam ve iris veri setleri UCI makine öğrenimi deposundan (Kelly vd., yıl bilinmiyor) alınmıştır. Veri kümelerinin özellikleri Tablo 3.1’de verilmiştir.

Tablo 3.1. Sınıflandırma için kullanılan veri kümelerinin özellikleri

Veri Seti	Örnek Sayısı	Öznitelik	Sınıf
Şarap	178	13	3
Göğüs Kanserisi	699	9	2
İris	150	4	3
Cam	214	9	2

3.1.1. Şarap Veri Kümesi

Şarap veri seti Stefan Aeberhard, M. Forina ve arkadaşları tarafından oluşturulmuştur. Veri seti içerisindeki veriler İtalya’da aynı bölgede yetişmiş üzümlerden yapılmış olan şarapların kimyasal analiz sonuçlarıdır ve 3 farklı yetiştiriciden elde edilen verilerden oluşmaktadır. Kimyasal analiz sonuçları bir şarabın kökenini bulmamızı sağlar. Elde edilen analizlerde üç şarap türünün her birinde mevcut olan 13 bileşenin miktarı belirtilmiştir. Veri setinde 178 adet örnek mevcuttur. Şarap veri kümesinin bileşen ve aralık değerleri Tablo 3.2’de verilmiştir.

Tablo 3.2. Şarap veri kümesinin nitelikleri ve aralık değerleri

Nitelik	Aralık Değerleri
Alkol - (ALC)	11,03 – 14,83
Malik Asit - (MAC)	0,74 – 5,80
Ash - (ASH)	1,36 – 3,23

Bazik Ash - (ALA)	10,60 – 30,00
Magnezyum - (MAG)	70,00 – 162,00
Toplam Phenols – (TPH)	0,98 – 3,88
Flavanoids - (FLA)	0,34 – 5,08
Nonflavanoid phenols - (NPH)	0,13 – 0,66
Proanthocyanins - (PRT)	0,41 – 3,58
Renk Yoğunluğu – (CIN)	1,28 – 13,00
Renk - (HUE)	0,48 – 1,71
OD280/OD315 ile Sulandırılmış - (ODW)	1,27 – 4,00
Proline - (PRL)	278,00 – 1680,00

3.1.2. Göğüs Kanseri Veri Kümesi

Göğüs kanseri veri seti ABD'deki Wisconsin Üniversitesi Hastanesinde doktor olan Dr. William H. Wolberg tarafından oluşturulmuştur. Veri setinde 9 farklı öznelik ve 699 örnek mevcuttur. 2 farklı sınıf için verilerin 458'i (%65.5) kötü huylu ve 241'i (%34.5) iyi huylu örneklerle sahiptir. Veri kümesinin nitelikleri ve aralık değerleri Tablo 3.3'te verilmiştir.

Tablo 3.3. Göğüs kanseri veri kümesinin nitelikleri ve aralık değerleri

Nitelik	Aralık Değerleri
Küme kalınlığı	1-10
Hücre boyutunun homojenliği	1-10
Hücre şeklinin homojenliği	1-10
Marjinal adezyon	1-10
Tek epitel hücre boyutu	1-10
Çıplak çekirdekler	1-10
Yumuşak kromatin	1-10
Normal nükleoller	1-10
Mitozlar	1-10

3.1.3. İris Veri Kümesi

İris veri kümesi R. A. Fisher'in oluşturduğu özellikle örüntü tanımda en iyi bilinen veri tabanıdır. Veri seti her biri bir tür iris bitkisi olan türlere ait 4 farklı nitelikten oluşmaktadır. Veri setinde 150 adet veri mevcuttur ve her bir tür için 50 örnekten oluşan 3 farklı sınıf içerir. Sınıf 1 (IrisSetosa), Sınıf 2 (IrisVersicolor) ve Sınıf 3 (IrisVirginica) olarak sınıflandırılmıştır. Veri kümesinin nitelikleri ve aralık değerleri Tablo 3.4'te verilmiştir.

Tablo 3.4. İris veri kümesinin nitelikleri ve aralık değerleri

Nitelik	Aralık Değerleri
Çanak YaprığınınUzunluğu - (SL)	4,3 – 7,9
Çanak YaprığınınGenişliği - (SW)	2 – 4,4
Taç Yaprığının Uzunluğu - (PL)	1 – 6,9
Taç Yaprığının Genişliği - (PW)	0,1 – 2,5

3.1.4. Cam Veri Kümesi

Diagnostic Products Corporation'dan Vina Spiehler tarafından oluşturulan cam kimliklendirme veri seti kriminolojik inceleme amacıyla kullanılmıştır. 9 niteliğe sahip veri kümesi 2 sınıf ve 214 örnekten oluşmaktadır. Veri kümesinin nitelikleri ve aralık değerleri Tablo 3.5'te verilmiştir.

Tablo 3.5. Cam veri kümesinin nitelikleri ve aralık değerleri

Nitelik	Aralık Değerleri
Kırılma Katsayısı - (RI)	1,5112 – 1,5339
Sodyum - (NA)	10,73 – 17,38
Magnezyum - (MG)	0 – 4,49
Aliminyum - (AL)	0,29 – 3,5
Silikon - (SI)	69,81 – 75,41
Potasyum - (K)	0 – 6,21
Kalsiyum - (CA)	5,43 – 16,19
Baryum- (BA)	0 – 3,15
Demir - (FE)	0 – 0,51

3.1.5. Video Sınıflandırma İçin Spor Veri Kümesi

Video sınıflandırma için kullanılan 3 sınıflı spor veri seti <https://github.com/jurjsorinliviu/Sports-Type-Classifier/tree/master/data> linki üzerinden GitHub'dan alınmış olup futbol, tenis ve ağırlık kaldırma spor dallarına ait görsellerden oluşmaktadır. Veri setinde futbol etiketli 799, tenis etiketli 718 ve ağırlık kaldırma etiketli 577 adet görsel mevcuttur. Görseller '.jpg' ve '.png' uzantılıdır.

3.2. Çalışma Ortamının Hazırlanması

Bu tez çalışmasında metasezgisel bir optimizasyon yöntemi olan COOT optimizasyon algoritması tabanlı yapay sinir ağı modeli (COOT-ANN) önerilmiş olup modelin performansı, geleneksel yapay sinir ağı modelleriyle karşılaştırılarak değerlendirilmiştir. İyi bilinen optimizasyon algoritmalarını kullanan Levenberg-Marquardt-YSA (LM-ANN), ölçeklenmiş eşlenik gradyan-YSA (SCG-ANN) ve gradyan inişi-YSA (GD-ANN) algoritmalarıyla önerilen modelin performansı dört farklı veri seti ve benchmarklar kullanılarak karşılaştırmalı olarak test edilmiştir.

Çalışmada şarap, göğüs kanseri, iris ve cam veri kümeleri için deneyler MATLAB kullanılarak gerçekleştirilmiştir. Deneylerde 10 nöronlu tek gizli katman yapısı kullanılmış ve giriş ve çıkış katmanları kullanılan veri setine göre uyarlanmıştır. Çapraz entropi (CE) fonksiyonu, sınıflandırma problemlerinde YSA model parametrelerinin belirlenmesinde sıklıkla tercih edilen bir maliyet fonksiyonudur çünkü konveks olması ve kolay optimizasyonu bilindir. Ayrıca, CE, test sürecinde modelin test performansını ölçmek için bir karşılaştırma metriği olarak da kullanılır. CE değeri 0 ile 1 aralığındadır ve sıfıra yakın değerler iyi sınıflandırma performansını gösterir. CE ve doğruluk metrikleri, modellerin eğitim sırasındaki performansını karşılaştırmak için kullanılmıştır. Modellerin test performansını karşılaştırmak için bu metriklere F1-skor ve Cohen's Kappa metrikleri de eklenmiştir. Ayrıca, modellerin performansını görselleştirmek için karmaşıklık matrisleri ve ROC-AUC eğrileri sunulmuştur. Eğitim ve test sürecinde, k-katlı doğrulama yöntemi kullanılmıştır. Her kat için elde edilen doğruluk ve CE değerlerinin aritmetik ortalaması, kat sayısının beş olarak seçilmesiyle hesaplanmıştır.

Video sınıflandırma çalışması için spor veri kümesi üzerinde deneyler Anaconda Navigator platformunda Spyder tümleşik geliştirme ortamı (IDE) ile Python 3.8.13 kullanılarak geliştirilmiştir. Yazılımda scikit-learn, TensorFlow, Keras ve OpenCV gibi önemli kütüphaneler kullanılmıştır. İlk adım olarak, veri kümesindeki spor etkinliklerini içeren görseller üzerinde işlem yapmak için bir ResNet-50 CNN modeli kullanılarak özellik çıkarımı yapılmıştır. Daha sonra, bu özellik vektörlerinin boyutunu azaltmak ve daha işlenebilir hale getirmek amacıyla Principal Component Analysis (PCA) yöntemi uygulanmıştır. PCA, çok boyutlu verilerin içerdikleri bilgiyi koruyarak daha az boyuta indirgeyen istatistiksel bir tekniktir. Veri seti, özellik vektörlerini belirli bir oranda indirgeyerek %75 eğitim ve %25 test verisi olarak ayrılmıştır. Eğitim verileri kullanılarak bir COOT-ANN modeli geliştirilmiş ve bu model, spor etkinliklerini sınıflandırmak için eğitilmiştir. Test sürecinde, modelin performansını daha ayrıntılı bir şekilde değerlendirmek amacıyla doğruluk oranının yanı sıra F1-skor ve Cohen's Kappa gibi metrikler de kullanılmıştır. Ayrıca, modelin performansını görselleştirmek ve karşılaştırmak için karmaşıklık matrisleri ve ROC-AUC eğrileri gibi grafiksel araçlar da kullanılmıştır. Model denemeleri sırasında farklı vektör boyutları, popülasyon sayıları, alt sınır ve üst sınır değerleri deneylerle test edilmiş ve en yüksek doğruluk değerine ulaşan ağırlık ve bias değerleri kaydedilmiştir. Bu şekilde, çalışmanın başarıyla tamamlanması için farklı parametrelerle yapılan denemeler ve değerlendirmeler gerçekleştirilmiştir.

3.3. Deneysel Sonuçlar

Bu bölümde yapılan deneysel çalışmalarda şarap, kanser, cam, iris ve spor veri setleri ile elde edilen performans değerleri karşılaştırmalı olarak verilmiştir.

3.3.1. Şarap, Kanser, İris ve Cam Veri Setleri ile Elde Edilen Sonuçlar

Şarap, kanser, iris ve cam veri setleri üzerinde 5 katlı çapraz doğrulama ile elde edilen ortalama değerler Tablo 3.6'da verilmiştir. Çalışmada kullanılan metriklerden biri doğruluktur, bu metrik karışıklık matrisindeki gerçek pozitif ve gerçek negatif sayısının genel popülasyona oranını ifade eder. Değerleri 0 ile 1 arasında alır ve 1'e yakın değerler yüksek bir sınıflandırma performansını gösterir. Tablo 3.6'da veri setleri için elde edilen test doğruluk değerleri incelendiğinde, COOT-ANN modelinin şarap veri seti için %97.21, kanser veri seti için %95.28, iris veri seti için %96.67 ve cam veri seti için %93.91 gibi en yüksek doğruluk değerlerine ulaştığı

görülebılır. Hem eğitim hem de test verileri için düşük CE metrik değeri, modelin başarılı ve doğru bir şekilde oluşturulduğunu göstermektedir.

Tablo 3.6. Veri kümelerinin ortalama performansları (5 kat)

Veri Seti	Metot	Eğitim verisi		Test verisi	
		Doğruluk	CE	Doğruluk	CE
Cam	COOT-ANN	0.9826	0.0011	0.9391	0.1887
	SCG- ANN	1.0000	0.0000	0.9318	0.2661
	LM- ANN	1.0000	0.0000	0.9343	0.0663
	GD- ANN	0.8891	0.1221	0.8919	0.1211
İris	COOT- ANN	1.0000	0.0000	0.9667	0.4887
	SCG- ANN	1.0000	0.0000	0.9467	0.2929
	LM- ANN	1.0000	0.0000	0.9400	0.0399
	GD- ANN	0.8567	0.1528	0.8333	0.1580
Kanser	COOT- ANN	0.9843	0.0001	0.9528	0.2030
	SCG- ANN	1.0000	0.0000	0.9528	0.3179
	LM- ANN	1.0000	0.0000	0.9428	0.0537
	GD- ANN	0.9535	0.0700	0.9456	0.0756
Şarap	COOT- ANN	1.0000	0.0000	0.9721	0.0403
	SCG- ANN	1.0000	0.0000	0.9662	0.0486
	LM- ANN	1.0000	0.0000	0.9662	0.0233
	GD- ANN	0.9087	0.1449	0.8932	0.1483

Tablo 3.7, sınıflandırma doğruluğunu, F1-skorunu, karışıklık matrisini ve ROC-AUC metriklerini karşılaştırmak için rastgele seçilmiş bir kat için performans değerlerini sunmaktadır. ROC eğrisi altında kalan alanı ifade eden AUC indeksi, sınıflandırıcının performansını değerlendirmek için önemli bir metriktir. 1.0 AUC değeri, sınıflandırıcının en yüksek performansa sahip olduğunu gösterir. Ayrıca, F1-skoru ve Cohen's Kappa metrikleri, doğruluk ve CE metriklerinden daha bilgilendirici ölçütlerdir. Bu metrikler, çok sınıflı ve dengesiz dağıtılan veri setlerinde sınıflandırma performansını değerlendirmek için yaygın olarak kullanılır.

Tablo 3.7. Veri kümelerinin örnek kat performansı

Veri seti	Metot	Eğitim verisi		Test verisi		F1 Skoru	Cohen's Kappa
		Doğruluk	CE	Doğruluk	CE		
Cam	COOT-ANN	1.0000	0.0015	0.9545	0.0972	0.8999	0.8709
	SCG- ANN	1.0000	0.0000	0.9318	0.2661	0.8570	0.8125
	LM- ANN	1.0000	0.0000	0.9545	0.0456	0.8999	0.8709
	GD- ANN	0.8824	0.1147	0.9318	0.0830	0.8420	0.8000
İris	COOT-ANN	1.0000	0.0000	1.0000	0.0000	1.0000	1.0000
	SCG- ANN	1.0000	0.0000	1.0000	0.0000	1.0000	1.0000
	LM- ANN	1.0000	0.0000	1.0000	0.0000	1.0000	1.0000
	GD- ANN	0.9417	0.1389	0.9000	0.1410	0.8695	0.8500
Kanser	COOT-ANN	1.0000	0.0001	0.9714	0.1634	0.9777	0.9377
	SCG- ANN	1.0000	0.0000	0.9571	0.4042	0.9670	0.9058
	LM- ANN	1.0000	0.0000	0.9643	0.0295	0.9723	0.9218
	GD- ANN	0.9589	0.0626	0.9643	0.0694	0.9723	0.9218
Şarap	COOT-ANN	1.0000	0.0000	1.0000	0.0052	1.0000	1.0000
	SCG- ANN	1.0000	0.0000	0.9429	0.1034	0.9504	0.9140
	LM- ANN	1.0000	0.0000	1.0000	0.0045	1.0000	1.0000
	GD- ANN	0.9091	0.1503	0.8571	0.1653	0.8668	0.7817

Tablo 3.7'deki test verilerine yönelik sonuçlar incelendiğinde, aşağıdaki sonuçlar elde edilmiştir:

- Cam veri seti için, COOT-ANN ve LM-ANN modelleri test doğruluğu, test F1-skoru ve Cohen's Kappa metrikleri açısından en yüksek performansa sahiptir.
- Iris veri seti için, COOT-ANN, SCG-ANN ve LM-ANN modelleri tüm metrikler açısından en yüksek performansa sahiptir.
- Kanser veri seti için, COOT-ANN modeli test doğruluğu, test F1-skoru ve Cohen's Kappa metrikleri açısından en yüksek performansa sahiptir.
- Şarap veri seti için, COOT-ANN ve LM-ANN modelleri test doğruluğu, test CE, test F1-skoru ve Cohen's Kappa metrikleri açısından en yüksek performansa sahiptir.

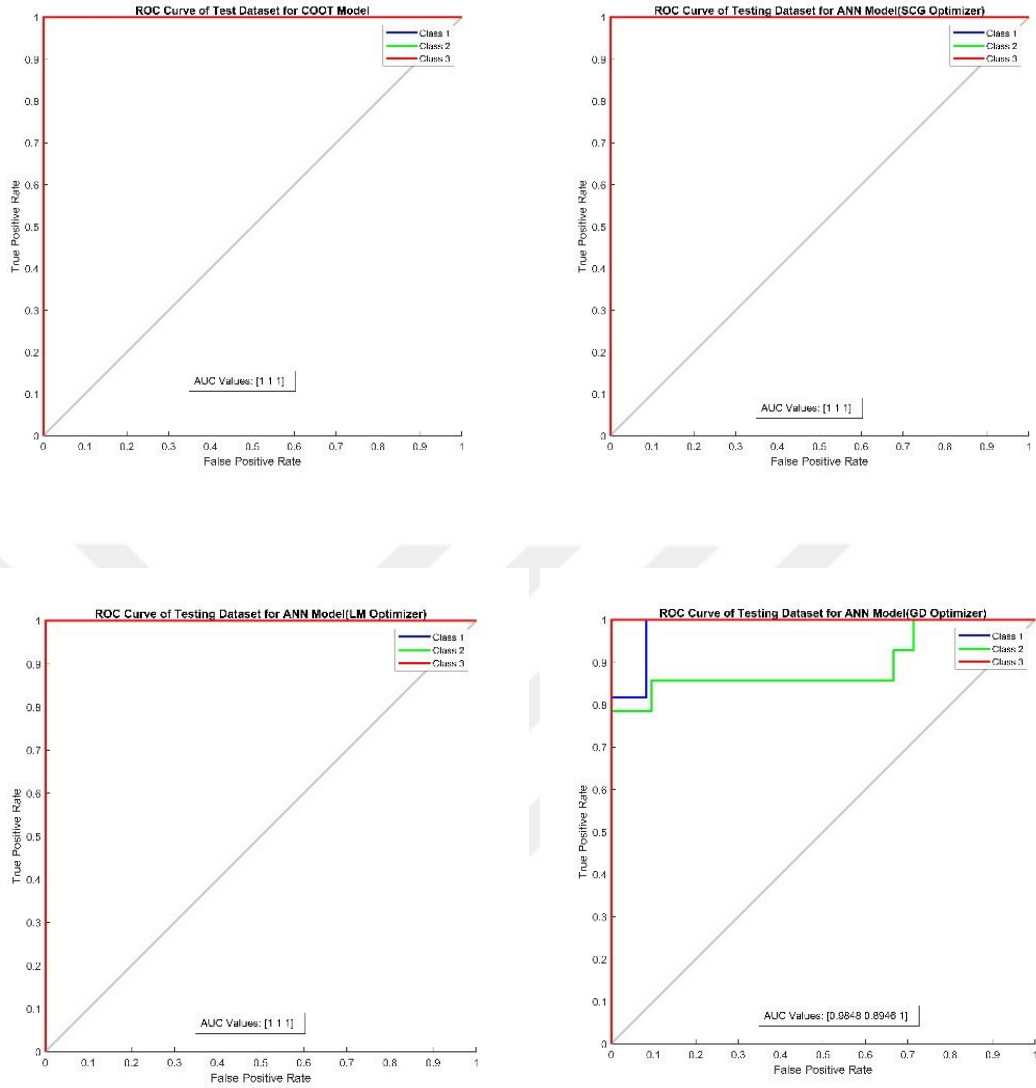
Önerilen COOT-ANN modelinin ve diğer modellerin şarap veri setindeki karışıklık matrisleri Şekil 3.1'de sunulmaktadır. 3 sınıflı bir veri seti olan şarap veri seti üzerinde COOT-ANN modeli için elde edilen doğruluk değeri karışıklık matrisi

ile doğrulanmaktadır. Ayrıca, karışıklık matrisleri diğer modellerin sınıflandırma hatalarını da ortaya koymaktadır. SCG-ANN ve GD-ANN ikinci sınıfa ait iki örneği birinci sınıf olarak, GD-ANN ise üçüncü sınıfa ait üç örneği ikinci sınıfa ait olarak yanlış sınıflandırmıştır.



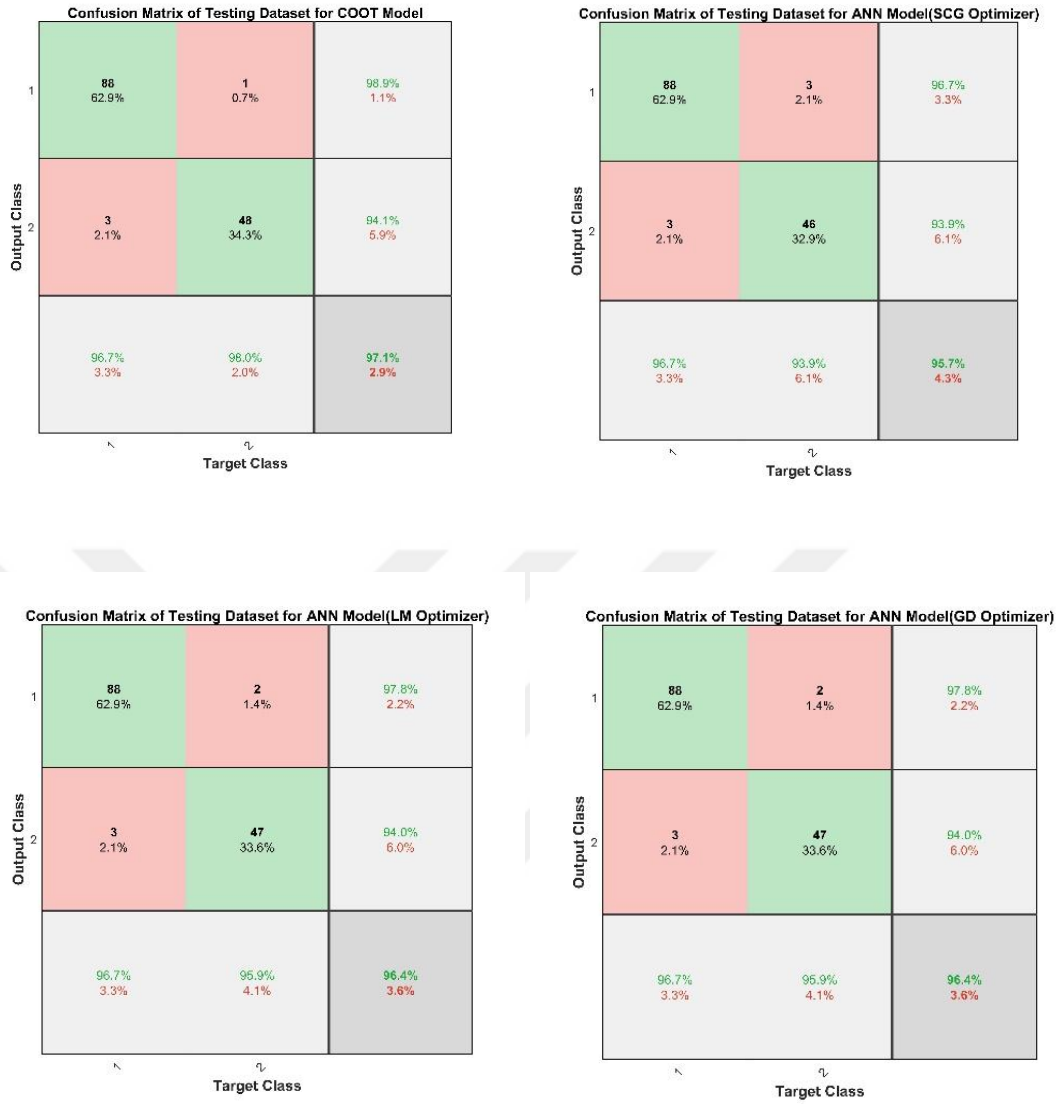
Şekil 3.1. Şarap veri kümesinin test verisi için model bazlı karışıklık matrisi

Şekil 3.2’de, şarap veri setine uygulanan yöntemlerin ROC eğrileri ve AUC değerleri sunulmaktadır. Bu eğriler, en yüksek performansın SCG-ANN, LM-ANN ve önerilen COOT-ANN yöntemi tarafından elde edildiğini, en düşük performansın ise GD-ANN modeli tarafından sergilendiğini göstermektedir. Şarap veri seti dengeli bir veri seti olduğundan, F1-skoru metriği ile uyumlu bir davranış göstermektedir.



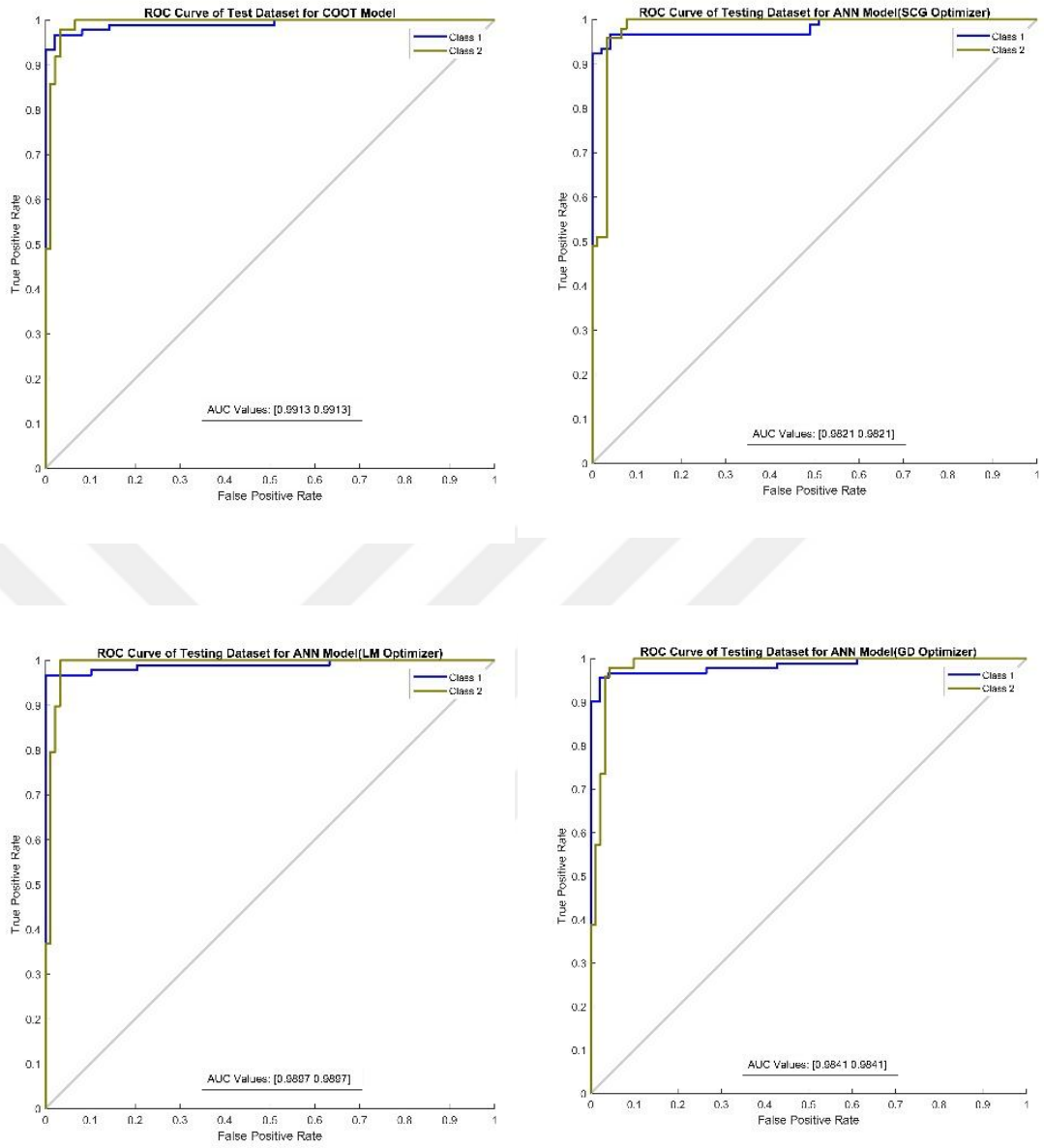
Şekil 3.2. Şarap veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri

Göğüs kanseri veri setindeki önerilen COOT-ANN modelinin ve diğer modellerin karışıklık matrisleri Şekil 3.3'te sunulmaktadır. İki sınıflı bir veri seti olan kanser veri seti üzerinde COOT-ANN modeli için elde edilen doğruluk değeri karışıklık matrisi ile doğrulanmaktadır. Ayrıca, karışıklık matrisleri diğer modellerin sınıflandırma hatalarını da ortaya koymaktadır. COOT-ANN modeli dört örneği yanlış sınıflandırırken, LM-ANN ve GD-ANN modelleri beş örneği yanlış sınıflandırmış ve SCG-ANN modeli altı örneği yanlış sınıflandırmıştır.



Şekil 3.3. Göğüs kanseri veri kümesinin test verisi için model bazlı karışıklık matrisi

Göğüs kanseri veri setine uygulanan yöntemlerin ROC eğrileri ve AUC değerleri Şekil 3.4'te sunulmaktadır. Bu eğriler, COOT-ANN yönteminin diğer yöntemlere kıyasla daha yüksek bir AUC değerine sahip olduğunu göstermektedir. Kanser veri seti dengesiz bir veri dağılımına sahip olduğundan, F1-skorunun diğer modellere kıyasla COOT-ANN modeli için daha yüksek olması modelin performansını desteklemektedir.



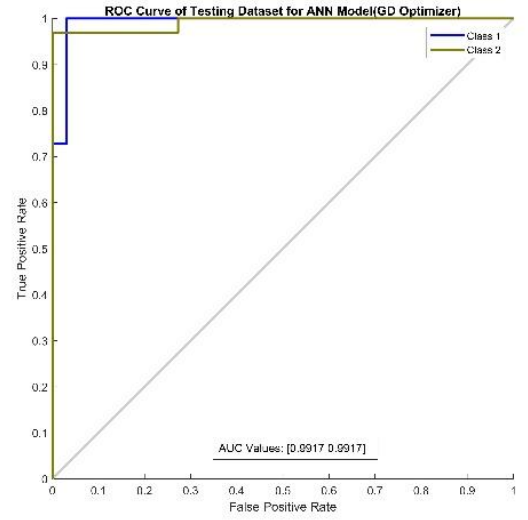
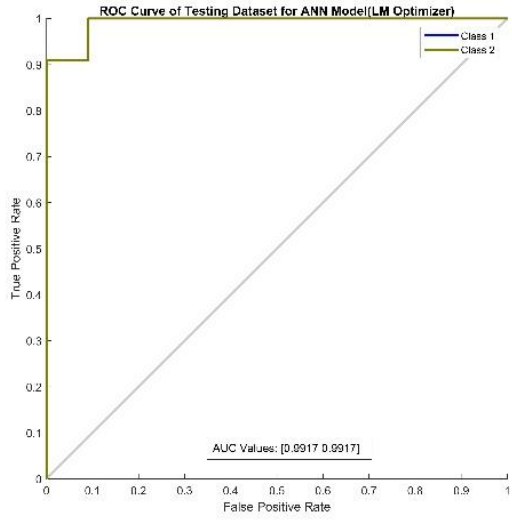
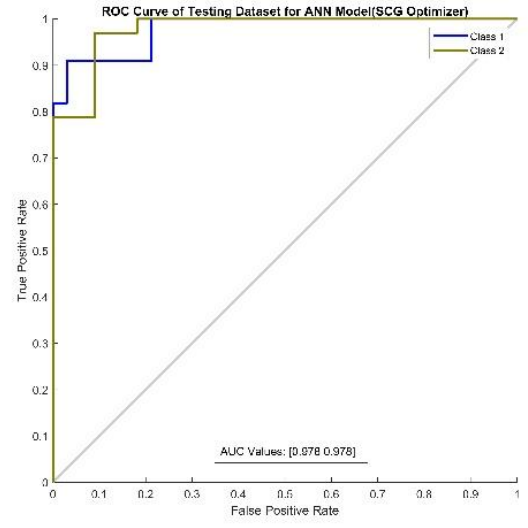
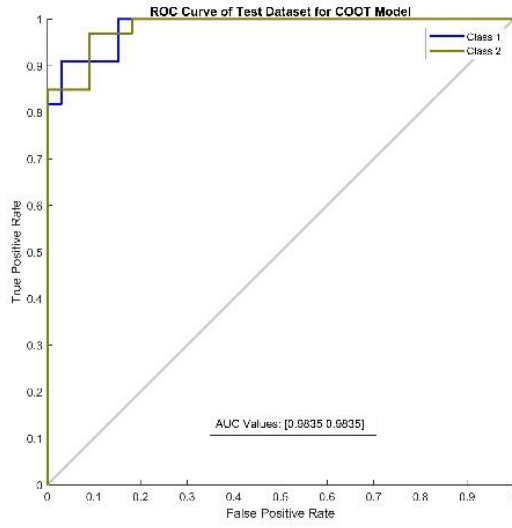
Şekil 3.4. Göğüs kanseri veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri

Cam veri setindeki önerilen COOT-ANN modelinin ve diğer modellerin karışıklık matrisleri Şekil 3.5’te sunulmaktadır. İki sınıflı bir veri seti olan cam veri seti üzerinde COOT-ANN modeli için elde edilen doğruluk değeri karışıklık matrisi ile doğrulanmaktadır. Ayrıca, karışıklık matrisleri diğer modellerin sınıflandırma hatalarını da ortaya koymaktadır. COOT-ANN ve LM-ANN birinci sınıfa ait 2 örneği ikinci sınıf olarak etiketlerken, SCG-ANN ve GD-ANN ise 3 örneği yanlış sınıflandırmıştır.



Şekil 3.5. Cam veri kümesinin test verisi için model bazlı karışıklık matrisi

Şekil 3.6, cam veri setine uygulanan yöntemlerin ROC eğrilerini ve AUC değerlerini sunmaktadır. Eğriler, en iyi performansın LM-ANN ve GD-ANN modelleri tarafından elde edildiğini, en düşük performansın ise SCG-ANN modeli tarafından sergilendiğini göstermektedir. Bununla birlikte, bu sadece önerilen modelin başarısız olduğu anlamına gelmez. Cam veri seti gibi sınıf dengesizliği gösteren veri setlerinde F1-skoru daha bilgilendirici bir değerlendirme ölçütüdür. Bu metrik kullanılarak modeller değerlendirildiğinde, COOT-ANN modelinin, LM-ANN modeliyle birlikte en yüksek F1-skoruna sahip olduğu görülmektedir.



Şekil 3.6. Cam veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri

Önerilen COOT-ANN modelinin ve diğer modellerin iris veri kümesi için karışıklık matrisleri Şekil 3.7’de sunulmaktadır. Karışıklık matrisi, iris veri setinde COOT-ANN modelinin doğruluğunu doğrulamaktadır, bu birçok sınıflı sınıflandırma görevidir. COOT-ANN, SCG-ANN ve LM-ANN modelleri hiçbir hatalı sınıflandırma yapmazken, GD-ANN modeli ikinci sınıfa ait iki veriyi birinci sınıfa ve bir veriyi üçüncü sınıfa ait olarak yanlış sınıflandırmıştır.

Confusion Matrix of Testing Dataset for COOT Model

Output Class	1	2	3	
1	10 33.3%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 33.3%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 33.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	Target Class

Confusion Matrix of Testing Dataset for ANN Model(SCG Optimizer)

Output Class	1	2	3	
1	10 33.3%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 33.3%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 33.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	Target Class

Confusion Matrix of Testing Dataset for ANN Model(LM Optimizer)

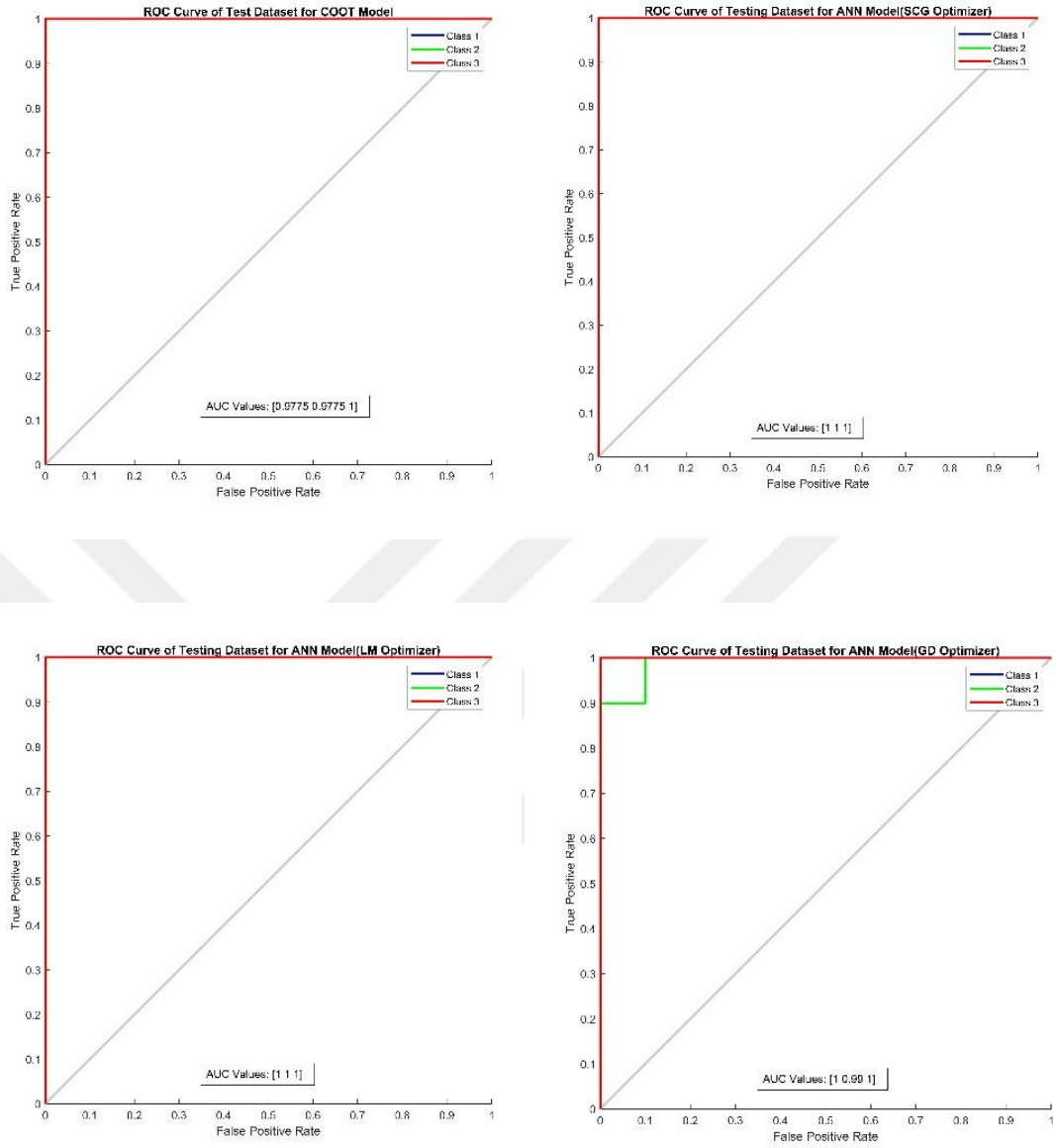
Output Class	1	2	3	
1	10 33.3%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 33.3%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 33.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	Target Class

Confusion Matrix of Testing Dataset for ANN Model(GD Optimizer)

Output Class	1	2	3	
1	10 33.3%	2 6.7%	0 0.0%	83.3% 16.7%
2	0 0.0%	7 23.3%	0 0.0%	100% 0.0%
3	0 0.0%	1 3.3%	10 33.3%	90.9% 9.1%
	100% 0.0%	70.0% 30.0%	100% 0.0%	90.0% 10.0%
	1	2	3	Target Class

Şekil 3.7. İris veri kümesinin test verisi için model bazlı karışıklık matrisi

Şekil 3.8, iris veri kümesine uygulanan yöntemlerin ROC eğrilerini ve AUC değerlerini sunmaktadır. Eğriler, önerilen COOT-ANN yönteminin SCG-ANN ve LM-ANN yöntemleriyle aynı sınıflandırma performansına sahip olduğunu, GD-ANN modelinin ise en düşük AUC değerine sahip olduğunu göstermektedir.



Şekil 3.8. İris veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri

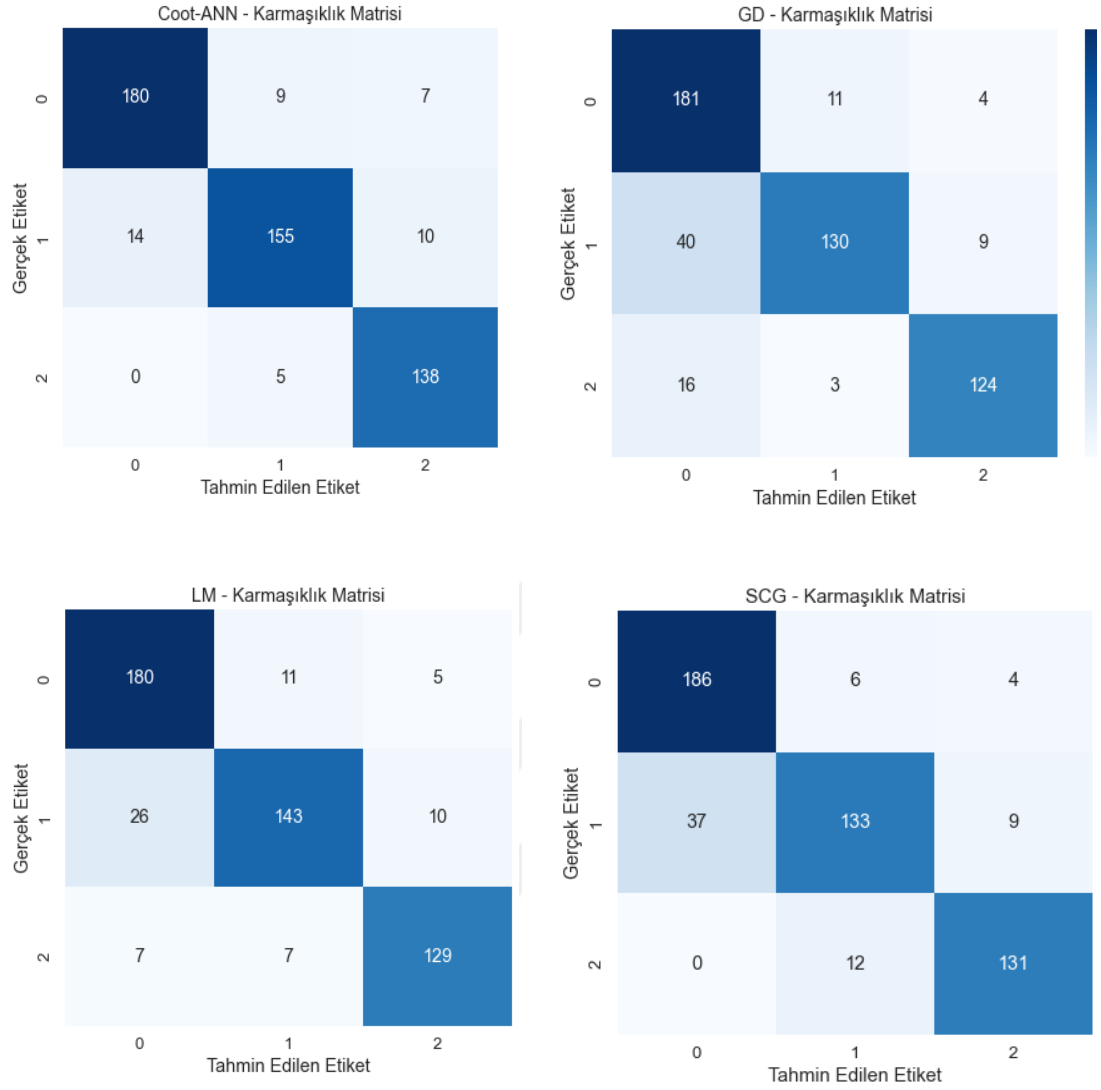
3.3.2. Video Sınıflandırma Veri Seti ile Elde Edilen Sonuçlar

Tablo 3.8’de COOT-ANN, SCG-ANN, GD-ANN ve LM-ANN modellerinin 3 sınıflı spor video sınıflandırmada test verileri üzerindeki başarımlar değerlendirilmiştir.

Tablo 3.8. Test verileri üzerinde elde edilen model performansları

Model	Doğruluk	CK	F1
COOT-ANN	0.9247	0.8863	0.9207
SCG-ANN	0.8687	0.8009	0.8664
LM-ANN	0.8725	0.8070	0.8718
GD-ANN	0.8397	0.7564	0.8390

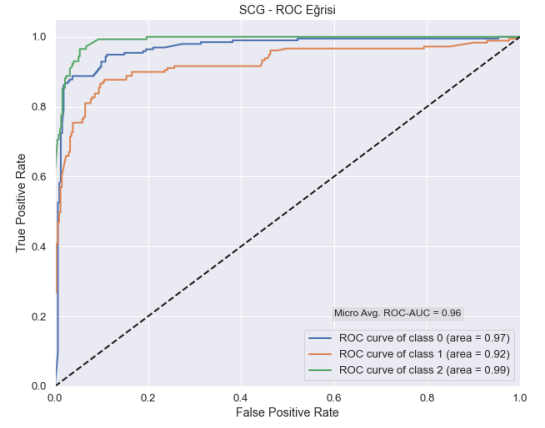
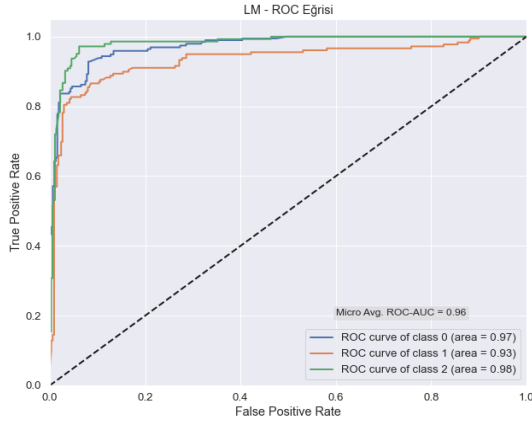
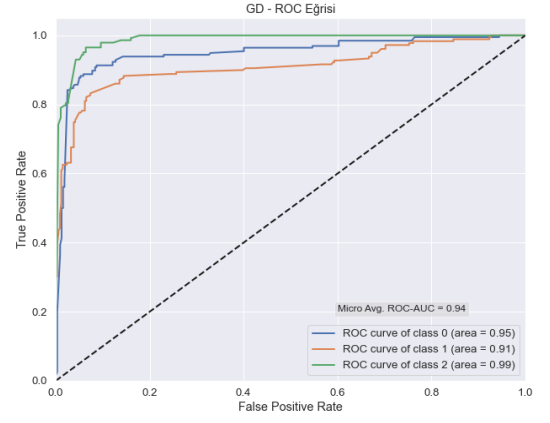
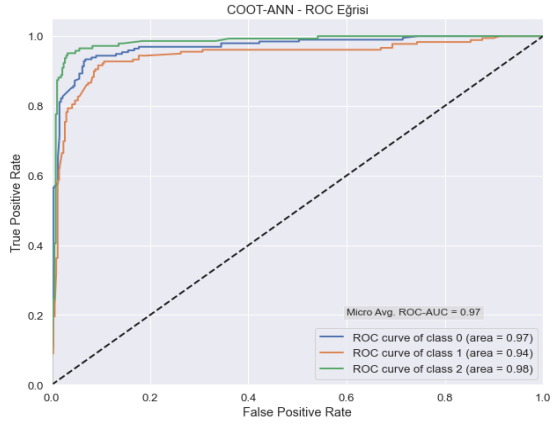
Tablo 3.8’de elde edilen performans değerlendirme metrikleri incelendiğinde en yüksek doğruluk, CK ve F1 skorunun COOT-ANN modelinde elde edildiği görülmektedir. Deneysel çalışmalar sonucunda en düşük doğruluk CK ve F1 değerinin ise GD-ANN modeli ile elde edildiği görülmektedir. Performans metriklerinin başarımlarını doğrulayabilmek amacıyla Şekil 3.9’da modellere ait karmaşıklık matrisleri verilmiştir.



Şekil 3.9. Spor veri kümesinin test verisi için model bazlı karmaşıklık matrisi

Üç sınıflı bir veri seti olan spor veri seti üzerinde COOT-ANN modeli için elde edilen doğruluk değeri karmaşıklık matrisi ile doğrulanmaktadır. Ayrıca, karmaşıklık matrisleri diğer modellerin sınıflandırma hatalarını da ortaya koymaktadır.

Şekil 3.10’da, spor veri setine uygulanan yöntemlerin ROC eğrileri ve AUC değerleri sunulmaktadır. Bu eğriler, en yüksek performansın COOT-ANN yöntemi tarafından elde edildiğini, en düşük performansın ise GD-ANN modeli tarafından sergilendiğini göstermektedir.



Şekil 3.10. Spor veri kümesinin test verisi için model bazlı ROC Eğrisi ve AUC değeri

4. SONUÇLAR VE ÖNERİLER

4.1. Sonuçlar

Literatürde, yerel minimumlara takılma sorunu nedeniyle gradyan iniş algoritmalarının yerine metasezgisel algoritmaların kullanıldığı ve yüksek performans gösterdiği gözlemlenmektedir. Bu çalışmada, en yeni metasezgisel algoritmalarından biri olan COOT algoritması, yapay sinir ağlarının eğitiminde ilk kez kullanılmış ve etkili bir hibrit metasezgisel optimizasyon tabanlı COOT-ANN modeli önerilmiştir. Önerilen model, UCI Deposu'ndan iris, kanser, cam, şarap ve video sınıflandırma çalışması için spor türleri örnek veri setlerindeki sınıflandırma problemlerine uygulanmıştır. COOT optimizasyon algoritmasına ait parametreler ve YSA parametreleri değişiminde veri setlerinde oluşturduğu etkiler deneysel çalışmalarda gözlemlenmiş ve en iyi parametre düzeyleri belirlenerek veri kümelerindeki sınıflandırma başarıları elde edilmiştir.

Önerilen model, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere 3 katmandan oluşmakta ve ikili veya çoklu sınıflandırma veri setlerine adapte olabilme yeteneği göstermektedir. Modelin performansı CE, doğruluk, F1-skoru ve Cohen's kappa (CK) metrikleri açısından değerlendirildiğinde, COOT-ANN modelinin performansının LM-ANN, SCG-ANN ve GD-ANN modellerinden daha yüksek olduğu görülmektedir. COOT-ANN modeli ile şarap ve iris test veri setlerinde %100 doğruluk, cam test veri setinde %95,45 doğruluk, meme kanseri test veri setinde %97,14 doğruluk ve spor veri setinde %92,47 doğruluk elde edilmiştir. F1-skoru ve CK değerlerine göre COOT-ANN modelinin performansı diğer test veri setlerinde daha yüksek çıkmıştır. COOT-ANN modeli, şarap ve iris test veri setlerinde F1-skoru = 1 ve CK = 1, cam veri setinde F1-skoru = 0,8999 ve CK = 0,8709, meme kanseri veri setinde F1-skoru = 0,9777 ve CK = 0,9377, spor veri setinde F1-skoru = 0,9207 ve CK = 0,8863 elde etmiştir. Karşılaştırmalı deney sonuçları incelendiğinde, önerilen yöntemin veri setleri üzerinde gerçekleştirilen sınıflandırma çalışmalarında diğer gradyan iniş dayalı algoritmalarından daha iyi sonuçlar verdiği gözlemlenmiştir. Sinir ağı parametrelerinin optimizasyonunda COOT algoritmasının kullanılmasının sınıflandırma sürecinin genel doğruluğunu artırabileceği görülmüştür.

4.2. Öneriler

Bu tez çalışmasında COOT algoritması ile yapay sinir ağı eğitimi yapılarak

ağırlık ve bias parametrelerinin optimizasyonu sağlanmaya çalışılmıştır. Çalışmaların devamında COOT algoritmasında geliştirmeler yapılarak YSA eğitimindeki başarısı artırılabilir ve algoritma evrimsel sinir ağlarının mimari yapı tasarımında ve ağ parametrelerinin optimizasyonunda kullanılmak üzere uyarlanabilir.

Çalışmada COOT-ANN modeli sadece sınıflandırma problemlerinde test edilmiştir. Bu nedenle, önerilen yöntem bu çalışma kapsamında sınıflandırma problemlerinin kullanımında sınırlıdır. Gelecekteki çalışmalarda, uygulanacak problemin yapısında düzenlemeler yapılarak modelin performansındaki değişiklikler gözlemlenebilir. Yapılacak olan düzenlemeler ve geliştirmeler yapay sinir ağlarının eğitiminde model performansını olumlu yönde etkileyebilir. Geliştirilen COOT-ANN yöntemi regresyon ve kümeleme problemleri gibi farklı problemlerde kullanılabilir.

KAYNAKLAR

- Abdolrasol, M. G., Hussain, S. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., ... and Milad, A. (2021). Artificial neural networks based optimization techniques: A review. *Electronics*, 10(21), 2689.
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., and Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, 113609.
- Abusnaina, A. A., Ahmad, S., Jarrar, R., and Mafarja, M. (2018). Training neural networks using salp swarm algorithm for pattern classification. In *Proceedings of the 2nd international conference on future networks and distributed systems* (pp. 1-6).
- Acı, Ç. and Çırak, A. (2019). Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması. *Bilişim Teknolojileri Dergisi*, 12(3), 219-228. DOI: 10.17671/gazibtd.457917
- Adamowski, J., and Chan, H. F. (2011). A wavelet neural network conjunction model for groundwater level forecasting. *Journal of Hydrology*, 407(1-4), 28-40.
- Agrawal, P., Abutarboush, H. F., Ganesh, T., and Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee Access*, 9, 26766-26791.
- Ahadzadeh, B., and Menhaj, M. B. (2014). Training neural networks using modified differential evolution algorithm for classification problems. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)* (pp. 598-603). IEEE.
- Akpak, Ç. U. (2017). *Social media image classification using deep convolutional neural networks* (Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi).
- Alan, A. (2020). *Makine öğrenmesi sınıflandırma yöntemlerinde performans metrikleri ile test tekniklerinin farklı veri setleri üzerinde değerlendirilmesi* (Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü).
- Al-Haija, Q. A., and Adebajo, A. (2020, September). Breast cancer diagnosis in histopathological images using ResNet-50 convolutional neural network. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)* (pp. 1-7). IEEE.
- Aljarah, I., Faris, H., and Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22, 1-15.
- Alwaisi, S., and Baykan, O. K. (2017). Training of artificial neural network using Metaheuristic algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, 5(Special Issue), 12-16.
- Ansari, A., Ahmad, I. S., Bakar, A. A., and Yaakub, M. R. (2020). A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction. *IEEE access*, 8, 176640-176650.
- Askarzadeh, A., and Rezazadeh, A. (2013). Artificial neural network training using a new efficient optimization algorithm. *Applied Soft Computing*, 13(2), 1206-1213.
- Ataseven, B. (2013). Yapay sinir ağları ile öngörü modellemesi. *Öneri Dergisi*, 10(39), 101-115.
- Bairathi, D., and Gopalani, D. (2021). Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: multiple leader salp swarm algorithm. *Evolutionary Intelligence*, 14(3), 1233-1249.

- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1), 1-127.
- Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.
- Beni, G., and Wang, U. (1989). Swarm intelligence in cellular robotic systems. In *NATO Advanced Workshop on Robots and Biological Systems*. Il Ciocco, Tuscany, Italy.
- Bhardwaj, S., and Kim, D. S. (2021). Dragonfly-based swarm system model for node identification in ultra-reliable low-latency communication. *Neural Computing and Applications*, 33, 1837-1880.
- Bitam, S., Zeadally, S., and Mellouk, A. (2018). Fog computing job scheduling optimization based on bees swarm. *Enterprise Information Systems*, 12(4), 373-397.
- Blum, C., and Socha, K. (2005, November). Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)* (pp. 6-pp). IEEE.
- Cantemir, Z. (2019). *Derin Öğrenme Yöntemleriyle Medikal Görüntü Sınıflandırma*. (Yüksek lisans tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı.)
- Carvalho, M., and Luderger, T. B. (2007). Particle swarm optimization of neural network architectures and weights. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)* (pp. 336-339). IEEE.
- Chen, J. F., Do, Q. H., and Hsieh, H. N. (2015). Training artificial neural networks by a hybrid PSO-CS algorithm. *Algorithms*, 8(2), 292-308.
- Cohen J. (1960). A coefficient of agreement for nominal scales. *Educ Psychol Measur* 20(1):37-46. <https://doi.org/10.1177/001316446002000104>
- Çavuşlu, M. A., Becerikli, Y., and Karakuzu, C. (2012). Levenberg-Marquardt algoritması ile YSA eğitiminin donanımsal gerçekleştirilmesi. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5(1).
- Delgado, R., and Tibau, X. A. (2019). Why Cohen's Kappa should be avoided as performance measure in classification. *PloS one*, 14(9), e0222916.
- Dorigo, M., and Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2-3), 243-278.
- Dutta, R. K., Karmakar, N. K., and Si, T. (2016). Artificial neural network training using fireworks algorithm in medical data mining. *Int. J. Comput. Appl*, 137(1), 1-5.
- Eusuff, M., Lansey, K., and Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization*, 38(2), 129-154.
- Faramarzi, A., Heidarinejad, M., Stephens, B., and Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, 105190.
- Faris, H., Aljarah, I., and Mirjalili, S. (2016). Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence*, 45, 322-332.
- Francis, L. (2014). Unsupervised learning. In E. W. Frees, R. A. Derrig, and G. Meyers (Eds.), *Predictive Modeling Applications in Actuarial Science* (pp. 280-312). Cambridge University Press.
- Gallo, C. (2015). Artificial neural networks tutorial. In *Encyclopedia of Information Science and Technology, Third Edition* (pp. 6369-6378). IGI Global.

- Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.
- Ghaleini, E. N., Koopialipoor, M., Momenzadeh, M., Sarafraz, M. E., Mohamad, E. T., and Gordan, B. (2019). A combination of artificial bee colony and neural network for approximating the safety factor of retaining walls. *Engineering with Computers*, 35, 647-658.
- Ghanem, W. A. H., Jantan, A., Ghaleb, S. A. A., and Nasser, A. B. (2020). An efficient intrusion detection model based on hybridization of artificial bee colony and dragonfly algorithms for training multilayer perceptrons. *IEEE Access*, 8, 130452-130475.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5), 533-549.
- Gülcü, Ş. (2020). Training of the artificial neural networks using states of matter search algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, 8(3), 131-136.
- Günler, M. A. (2012). *Identification of facial expressions using artificial neural networks (ANN)* (Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü).
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175-184.
- Haykin, S. (2009). *Neural networks and learning machines*. [sl] pearson upper saddle river. NJ, USA, 3.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Heidari, A. A., Faris, H., Aljarah, I., and Mirjalili, S. (2019). An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Computing*, 23, 7941-7958.
- Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., and Mafarja, M. (2020). Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*, 23-46.
- Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1), 66-73.
- Hu, X., Shi, Y., and Eberhart, R. (2004). Recent advances in particle swarm. In *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)* (Vol. 1, pp. 90-97). IEEE.
- Hussien, A. M., Turkey, R. A., Alkuhayli, A., Hasanien, H. M., Tostado-Véliz, M., Jurado, F., and Bansal, R. C. (2022). Coot bird algorithms-based tuning PI controller for optimal microgrid autonomous operation. *IEEE Access*, 10, 6442-6458.
- Irmak, B., and Gülcü, Ş. (2021). Training of the feed-forward artificial neural networks using butterfly optimization algorithm. *MANAS Journal of Engineering*, 9(2), 160-168.
- Japkowicz, N., and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.
- Kankal, M., and 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, 737-747.
- Karaboğa, D., Akay, B., and Oztürk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In *Modeling Decisions for*

- Artificial Intelligence: 4th International Conference*, MDAI 2007, Kitakyushu, Japan, August 16-18, 2007. Proceedings 4 (pp. 318-329). Springer Berlin Heidelberg.
- Karaboğa, D., and Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132.
- Karakaş, B. (2020). *Sıncap arama algoritması ile yapay sinir ağlarının eğitimi*. (Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Kayseri, Türkiye.)
- Kashan, A. H. (2009, December). League championship algorithm: a new algorithm for numerical function optimization. In *2009 international conference of soft computing and pattern recognition* (pp. 43-48). IEEE.
- Kayhan, G., and İşeri, İ. (2023). Counter propagation network based extreme learning machine. *Neural Processing Letters*, 55(1), 857-872.
- Kelly, M., Longjohn, R., & Nottingham, K. (Yıl bilinmiyor). The UCI Machine Learning Repository. <https://archive.ics.uci.edu>
- Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Kolias, C., Kambourakis, G., and Maragoudakis, M. (2011). Swarm intelligence in intrusion detection: A survey. *computers & security*, 30(8), 625-642.
- Korkmaz, S. (2019). *İkili optimizasyon problemlerinin çözümü için yapay alg algoritması tabanlı yeni yaklaşımlar*. (Doktora Tezi, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Konya, Türkiye.)
- Kulluk, S. (2009). *Karınca koloni optimizasyonu ile yapay sinir ağlarından kural çıkarımı*. (Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Makine Mühendisliği Anabilim Dalı, Kayseri.)
- Kulluk, S., Ozbakir, L., and Baykasoglu, A. (2012). Training neural networks with harmony search algorithms for classification problems. *Engineering Applications of Artificial Intelligence*, 25(1), 11-19.
- Kumar, N. and Kumar D. (2020). Classification using artificial neural network optimized with bat algorithm.
- Li, X. L. (2002). An optimizing method based on autonomous animats: fish-swarm algorithm. *Systems Engineering-Theory & Practice*, 22(11), 32-38.
- Liu, Q., and Wu, Y. (2012). Supervised learning. In N. M. Seel (Ed.), *Encyclopedia of 250 the sciences of learning* (pp. 3243-3245).
- Ma, H., Simon, D., Siarry, P., Yang, Z., and Fei, M. (2017). Biogeography-based optimization: a 10-year review. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(5), 391-407.
- Mandal, S., Saha, G., and Pal, R. K. (2015). Neural network training using firefly algorithm. *Global Journal on Advancement in Engineering and Science (GJAES)*, 1(1), 7-11.
- Martínez-Álvarez, F., Asencio-Cortés, G., Torres, J. F., Gutiérrez-Avilés, D., Melgar-García, L., Pérez-Chacón, R., ... and Troncoso, A. (2020). Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model. *Big data*, 8(4), 308-322.
- McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.

- Meissner, M., Schmuker, M., and Schneider, G. (2006). Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC bioinformatics*, 7(1), 1-11.
- Memarzadeh, G., and Keynia, F. (2021). A new optimal energy storage system model for wind power producers based on long short term memory and Coot Bird Search Algorithm. *Journal of Energy Storage*, 44, 103401.
- MiarNaeimi, F., Azizyan, G., and Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, 106711.
- Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43, 150-161.
- Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, 27, 1053-1073.
- Mirjalili, S. M. S. M., Mirjalili, S. M., and Lewis, A. (2014). Grey Wolf Optimizer Adv Eng Softw 69: 46–61.
- Mirjalili, S., Hashim, S. Z. M., and 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., Mirjalili, S. M., and Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. *Information sciences*, 269, 188-209.
- Mohamed, A. W., Hadi, A. A., and Mohamed, A. K. (2020). Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. *International Journal of Machine Learning and Cybernetics*, 11(7), 1501-1529.
- Møller M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *neural networks*. *Neural Netw* 6:525–533. [https://doi.org/10.1016/S0893-6080\(05\)80056-5](https://doi.org/10.1016/S0893-6080(05)80056-5)
- Naruei, I., and Keynia, F. (2021). A new optimization method based on COOT bird natural life model. *Expert Systems with Applications*, 183, 115352.
- Nedjai, A. (2020). *Training and optimizing deep artificial neural networks using dragonfly algorithm for medical prediction*. (Yüksek lisans tezi)
- Nezla, N. A., Haridas, T. M., and Supriya, M. H. (2021, March). Semantic segmentation of underwater images using unet architecture based deep convolutional encoder decoder model. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 28-33). IEEE.
- Nuaimi, Z. N. A. M. A., and Abdullah, R. (2017). Neural network training using hybrid particle-move artificial bee colony algorithm for pattern classification. *Journal of Information and Communication Technology (JICT)*, 16(2), 314-334.
- Olorunda, O., and Engelbrecht, A. P. (2008). Measuring exploration/exploitation in particle swarms using swarm diversity. In *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)* (pp. 1128-1134). IEEE.
- Othman, K. M. Z., and Salih, A. M. (2021). Scaled conjugate gradient ANN for industrial sensors calibration. *Bulletin of Electrical Engineering and Informatics*, 10(2), 680-688.
- Oztürk, C. (2011). *Yapay sinir ağlarının yapay arı kolonisi algoritması ile eğitilmesi*. (Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Kayseri, Türkiye.)

- Özden, A., and İşeri, İ. (2023). COOT optimization algorithm on training artificial neural networks. *Knowledge and Information Systems*, 1-31.
- Öztemel, E. (2003). Yapay sinir ağları. *PapatyaYayincılık, Istanbul*.
- Paillisson, J. M., and Marion, L. (2001). Interaction between coot (*Fulica atra*) and waterlily (*Nymphaea alba*) in a lake: the indirect impact of foraging. *Aquatic Botany*, 71(3), 209-216.
- Pereira, L. A., Rodrigues, D., Ribeiro, P. B., Papa, J. P., and Weber, S. A. (2014). Social-spider optimization-based artificial neural networks training and its applications for Parkinson's disease identification. In *2014 IEEE 27th international symposium on computer-based medical systems* (pp. 14-17). Ieee.
- Pineda, L. R., and Serpa, A. L. (2021). Determination of confidence bounds and artificial neural networks in non-linear optimization problems. *Neurocomputing*, 463, 495-504.
- Pravitasari, A. A., Iriawan, N., Almuhayar, M., Azmi, T., Irhamah, I., Fithriasari, K., ... and Ferriastuti, W. (2020). UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(3), 1310-1318.
- Rajakumar, B. R. (2012). The Lion's Algorithm: a new nature-inspired search algorithm. *Procedia Technology*, 6, 126-135.
- Rani, P., Kaur, J., & Kaswan, S. (2020). Automatic video classification: a review. *EAI Endorsed Transactions on Creative Technologies*, 7(24).
- Rao, R. V., Savsani, V. J., and Vakharia, D. P. (2012). Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences*, 183(1), 1-15.
- Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychoanalytic Review*, 65, 386-408.
- Salouhou, A. (2019). *El yazısı karakter tanıma ve resim sınıflandırmada derin öğrenme yaklaşımları* (Yüksek Lisans Tezi, Fatih Sultan Mehmet Vakıf Üniversitesi, Lisansüstü Eğitim Enstitüsü).
- Shah, H., Ghazali, R., and Nawı, N. M. (2011). Using artificial bee colony algorithm for MLP training on earthquake time series data prediction. *arXiv preprint arXiv:1112.4628*.
- Shakibjoo, A. D., Moradzadeh, M., Moussavi, S. Z., Mohammadzadeh, A., and Vandevelde, L. (2022). Load frequency control for multi-area power systems: A new type-2 fuzzy approach based on Levenberg-Marquardt algorithm. *ISA transactions*, 121, 40-52.
- Sharma, V., Chouhan, A. P. S., and Bisen, D. (2022). Prediction of activation energy of biomass wastes by using multilayer perceptron neural network with Weka. *Materials Today: Proceedings*, 57, 1944-1949.
- Si, T., Bagchi, J., and Miranda, P. B. (2022). Artificial neural network training using metaheuristics for medical data classification: an experimental study. *Expert Systems with Applications*, 193, 116423.
- Storn, R., and Price, K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341.
- Sun, M., Song, Z., Jiang, X., Pan, J., and Pang, Y. (2017). Learning pooling for convolutional neural network. *Neurocomputing*, 224, 96-104.

- Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J. (2020). Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE transactions on cybernetics*, 50(9), 3840-3854.
- Sutton, R. S., and Barto, A. G. (1998). Reinforcement learning: An introduction (1st ed.). *MIT Press*.
- Tabari, A., and Ahmad, A. (2017). A new optimization method: Electro-Search algorithm. *Computers & Chemical Engineering*, 103, 1-11.
- Talathi, S. S., and Vartak, A. (2015). Improving performance of recurrent neural network with relu nonlinearity. *arXiv preprint arXiv:1511.03771*.
- Theckedath, D., and Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, 1, 1-7.
- Toprak, A. G. (2022). *Early Detection Of Breast Cancer by Using Image Processing and Deep Learning*. (Yüksek Lisans Tezi, İzmir Demokrasi Üniversitesi.)
- Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., and Wahab, M. A. (2019). An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm. *Engineering Structures*, 199, 109637.
- Tripathi, A. K., Mittal, H., Saxena, P., and Gupta, S. (2021). A new recommendation system using map-reduce-based tournament empowered Whale optimization algorithm. *Complex & Intelligent Systems*, 7, 297-309.
- Türkoğlu, B. (2019). Yapay sinir ağlarının yapay alg algoritması ile eğitimi. *Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Türkoğlu, B., and Kaya, E. (2020). Training multi-layer perceptron with artificial algae algorithm. *Engineering Science and Technology, an International Journal*, 23(6), 1342-1350.
- Valian, E., Mohanna, S., and Tavakoli, S. (2011). Improved cuckoo search algorithm for feedforward neural network training. *International Journal of Artificial Intelligence & Applications*, 2(3), 36-43.
- Wang, D. (2001). Unsupervised learning: foundations of neural computation. *Ai Magazine*, 22(2), 101-101.
- Wang, H. X., Zhou, J. Q., GU, C. H., and Lin, H. (2019). Design of activation function in CNN for image classification. *Journal of Zhejiang University (Engineering Science)*, 53(7), 1363-1373.
- Wilamowski BM and Irwin JD (2011). The industrial electronics handbook. In: Bogdan MW, Irwin JD (eds) Control and mechatronics. CRC Press, Boca Rton
- Wilson, D. R., and Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks*, 16(10), 1429-1451.
- Wolpert, D. H., and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67-82.
- Xu, J., and Yan, F. (2019). Hybrid Nelder–Mead algorithm and dragonfly algorithm for function optimization and the training of a multilayer perceptron. *Arabian Journal for Science and Engineering*, 44, 3473-3487.
- Yaghini, M., Khoshraftar, M. M., and 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*. (Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.)
- Yamany, W., Tharwat, A., Hassanin, M. F., Gaber, T., Hassanien, A. E., and Kim, T. H. (2015). A new multi-layer perceptrons trainer based on ant lion optimization algorithm. In *2015 Fourth international conference on information science and industrial applications (ISI)* (pp. 40-45). IEEE.
- Yang, X. S. (2010). Nature-inspired metaheuristic algorithms. Luniver press.
- Yang, X. S., and Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering computations*, 29(5), 464-483.
- Yasen, M., Al-Madi, N., and Obeid, N. (2018, July). Optimizing neural networks using dragonfly algorithm for medical prediction. In *2018 8th international conference on computer science and information technology (CSIT)* (pp. 71-76). IEEE.
- Yılmaz, A. R. (2014). *FPGA üzerinde diferansiyel gelişim algoritması ile yapay sinir ağı eğitimi*. (Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, İstanbul.)
- Zamani, M., and Sadeghian, A. (2010). A variation of particle swarm optimization for training of artificial neural networks. In *Computational Intelligence and Modern Heuristics*. IntechOpen.
- Zanchettin, C., Ludermir, T. B., and Almeida, L. M. (2011). Hybrid training method for MLP: optimization of architecture and training. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(4), 1097-1109.
- Zhang, Q., Wang, R., Yang, J., Ding, K., Li, Y., and Hu, J. (2017). Collective decision optimization algorithm: A new heuristic optimization method. *Neurocomputing*, 221, 123-137.
- Zhao, W., Wang, L., and Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388, 114194.