



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



YAPAY SİNİR AĞLARININ EĞİTİMİ İÇİN
SALP SÜRÜ OPTİMİZASYONU
ALGORİTMASININ İYİLEŞTİRİLMESİ

Durmuş Furkan ATLI

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ağustos-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Durmuş Furkan ATLI tarafından hazırlanan “Yapay Sinir Ağlarının Eğitimi için Salp Sürü Optimizasyonu Algoritmasının İyileştirilmesi” adlı tez çalışması 03/08/2022 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğr. Üyesi Semih YUMUŞAK

.....

Danışman

Dr. Öğr. Üyesi Şaban GÜLCÜ

.....

Üye

Doç. Dr. Mehmet HACİBEYOĞLU

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun/.../20.. gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. İbrahim KALAYCI
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Durmuş Furkan ATLI

Tarih: 03.08.2022

ÖZET

YÜKSEK LİSANS TEZİ

YAPAY SİNİR AĞLARININ EĞİTİMİ İÇİN SALP SÜRÜ OPTİMİZASYONU ALGORİTMASININ İYİLEŞTİRİLMESİ

Durmuş Furkan ATLI

**Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Dr. Öğr. Üyesi Şaban GÜLCÜ

2022, 45 Sayfa

Jüri

**Dr. Öğr. Üyesi Şaban GÜLCÜ
Doç. Dr. Mehmet HACİBEYOĞLU
Dr. Öğr. Üyesi Semih YUMUŞAK**

Son yıllarda popülerlik kazanan meta sezgisel algoritmalarından birisi de salp sürü optimizasyon algoritmasıdır (SSA). Derin okyanuslarda yaşayan bu canlılar besin bulmak ve yer değiştirmek için gruplar halinde dolaşırlar. Salplar dairesel bir zincir şeklinde ilerleyerek daha hızlı besin arama ve koordineli yer değiştirme yeteneği ile SSA algoritmasına ilham olmuşturlar. Bu tez çalışmasında çok katmanlı algılayıcıları (ÇKA) eğitmek için iyileştirilmiş salp sürü algoritması (İSSA) önerilmiştir. İSSA çalışmasında arama uzayının keşfi ve global en iyi değere en hızlı şekilde ulaşmak için trigonometrik fonksiyonlardan ve rassallıktan yararlanılmıştır. Aynı zamanda metasezgisel algoritmaların genel problemlerinden biri olan yerel optimuma takılma sorunundan kaçınmak için kaos haritalarından yararlanılmıştır. Önerilen İSSA algoritması yapay sinir ağları (YSA) eğitimlerinde ağırlık ve bias değerlerinin optimum noktaya getirilmesi için kullanılmıştır. ÇKA'ları eğitmek için kullanılan İSSA bu tez çalışmasında İSSA-ÇKA olarak adlandırılmıştır. İSSA-ÇKA algoritmasının başarısını ölçmek için literatürde en çok kullanılan beş farklı veri seti ele alınmış farklı ÇKA yapıları kullanılmış ve iki farklı optimizasyon algoritması ile kıyaslama yapılmıştır. Kullanılan veri setleri xor, balon, iris, meme kanseri ve kalp veri setidir. SSA tabanlı İSSA-ÇKA algoritması, SSA-ÇKA ve SMS algoritması tabanlı SMS-ÇKA ile kıyaslanmaktadır. Kıyaslama sonucunda ortalama doğruluk oranlarında İSSA-ÇKA algoritmasının SMS-ÇKA algoritmasını üstün bir başarı ile geçtiği, SSA-ÇKA algoritmasını ise beş farklı veri setinden xor, balon, iris ve kalp veri setlerinde geçtiğini, meme kanseri veri setinde ise SSA-ÇKA algoritmasının İSSA-ÇKA algoritmasını geçtiği görülmüştür. İSSA-ÇKA algoritmasının arama uzayını daha iyi keşfederek global en iyi noktaya ulaşmayı hedeflemiştir. İSSA-ÇKA algoritmasının global en iyi noktasına diğer algoritmalarla göre düşük iterasyonlarda daha hızlı yakınsadığı yakınsama grafikleri ile ispatlanmıştır.

Anahtar Kelimeler: Çok katmanlı algılayıcılar, Kaos, Salp Sürü optimizasyon algoritması, Meta sezgisel, Yapay sinir ağları, Yapay sinir ağları eğitimi.

ABSTRACT

MS THESIS

IMPROVEMENT OF SALP SWARM OPTIMIZATION ALGORITHM FOR TRAINING OF ARTIFICIAL NEURAL NETWORKS

Durmuş Furkan ATLI

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Asst. Prof. Dr. Şaban GÜLCÜ

2022, 45 Pages

Jury

**Asst. Prof. Dr. Şaban GÜLCÜ
Assoc. Prof. Dr. Mehmet HACIBEYOĞLU
Asst. Prof. Dr. Semih YUMUŞAK**

One of the meta-heuristic algorithms that have gained popularity in recent years is the salp swarm optimization algorithm (SSA). Living in deep oceans, these creatures roam in groups to find food and move places. Salps were inspired by the SSA algorithm with their ability to move in a circular chain, faster foraging, and coordinated displacement. In this thesis, an improved salp swarm algorithm (ISSA) is proposed to train multilayer perceptrons (MLP). In the ISSA study, trigonometric functions and randomness were used to maintain the balance of exploration and exploitation of the search space. At the same time, chaos maps are used to avoid the problem of getting stuck in the local optimum, which is one of the general problems of metaheuristic algorithms. The proposed ISSA algorithm is used to optimize the weight and bias values in artificial neural networks (ANN) training. ISSA used to train MLPs is named ISSA-MLP in this thesis. To measure the success of the ISSA-MLP algorithm, five different datasets most commonly used in the literature were handled, different MLP structures were used, and a comparison was made with two different optimization algorithms. The datasets used are xor, balloon, iris, breast cancer, and heart dataset. The SSA-based SSA-MLP algorithm is compared with the SSA-MLP and the SMS algorithm-based SMS-MLP. As a result of the comparison, in average accuracy rates, the ISSA-MLP algorithm passed the SMS-MLP algorithm with superior success, the SSA-MLP algorithm passed the xor, balloon, iris, and heart datasets from five different datasets, and the SSA-MLP algorithm in the breast cancer dataset was found to be ISSA-MLP algorithm. It was discussed that he passed the MLP algorithm. It aimed to reach the global best point by better exploring the search space of the ISSA-MLP algorithm. It is shown in the line graphs that the ISSA-MLP algorithm converges to the global best point faster with a slightly less number of iterations compared to other algorithms.

Keywords: Artificial neural networks, Salp swarm optimization algorithm, Chaos, Metaheuristic, Multilayer perceptron, Training artificial neural networks.

ÖNSÖZ

Yüksek lisans hayatım boyunca gerek maddi gerek manevi olarak yardımcı olan, bilgi, deneyim ve tecrübesi ile bana yol gösteren danışmanım Sayın Dr. Öğr. Üyesi Şaban GÜLCÜ'ye, tez çalışmasının son halini alması aşamasında kıymetli fikirleri ile katkıda bulunan jüri üyeleri Sayın Doç. Dr. Mehmet HACIBEYOĞLU'na ve Sayın Dr. Öğr. Üyesi Semih YUMUŞAK'a, hayatımın belli bir ders döneminde beni aydınlatan ve farklı bakış açısı ile beni bilgilendiren hocalarım Prof. Dr. Sabri KOÇER, Dr. Öğr. Üyesi Ayşe Merve ACILAR, Doç. Dr. Hüseyin HAKLI ve Dr. Öğr. Üyesi Murat KARAKOYUN'a teşekkürü borç bilirim.

Tez çalışmam boyunca bana maddi ve manevi destekte bulun aileme ve dayıma teşekkürlerimi sunarım.

Durmuş Furkan ATLI
KONYA-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	ix
ŞEKİLLER LİSTESİ	x
ÇİZELGELER LİSTESİ	xi
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Önemi	2
1.2. Tezin Organizasyonu	3
2. KAYNAK ARAŞTIRMASI	5
3. MATERYAL VE YÖNTEM.....	9
3.1. Yapay Sinir Ağı (YSA).....	9
3.1.1. Çok katmanlı algılayıcı (ÇKA).....	11
3.2. Salp Sürü Algoritması (SSA).....	17
3.2.1. Salp sürüsünün hareketi	18
3.2.2. Lider salp için konum güncelleme denklemi	19
3.2.3. SSA'da $c1$ parametresi	19
3.2.4. Takipçi salpların konumunu güncelleme denklemi	19
3.2.5. SSA sözde kodu ve akış diyagramı.....	19
3.3. Sınıflandırma Veri Setleri.....	20
3.4. İstatistiksel Başarım Metrikleri.....	21
3.4.1. Ortalama karesel hata (MSE).....	21
3.4.2. Friedman testi	22
3.4.3. Duyarlılık (Sensitivity)	23
3.4.4. Özgüllük (Specificity).....	23
3.4.5. Kesinlik (Precision)	23
3.4.6. F1-Score.....	24
4. ÖNERİLEN YÖNTEM	25
4.1. İyileştirilmiş Salp Sürü Algoritması (İSSA).....	25
4.1.1. Kaotik haritalar	26
4.1.2. Lider salp ve trigonometrik salp konum güncelleme.....	28
4.1.3. Takipçi salp konum güncelleme	28
4.1.4. Rassal pozisyon güncelleme	29

4.1.5. İSSA sözde kodu ve akış diyagramı	29
4.2. YSA'nın İyileştirilmiş Salp Sürü Algoritması İle Eğitimi (İSSA-ÇKA)	31
5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	34
6. SONUÇLAR VE ÖNERİLER.....	42
6.1. Sonuçlar	42
6.2. Öneriler	42
7. KAYNAKLAR	44
ÖZGEÇMİŞ	47



SİMGELER VE KISALTMALAR

AB	:	Ateş böceği algoritması
AKO	:	Aslan karınca optimizasyonu
ÇBPSO	:	Çok boyutlu parçacık sürü optimizasyonu
ÇKA	:	Çok katmanlı algılayıcılar
ÇOA	:	Çekirge optimizasyon algoritması
GA	:	Genetik algoritma
GAA	:	Güve alevi algoritması
GKO	:	Gri kurt optimizasyonu
GSA	:	Yerçekimi arama algoritması
HGA	:	Hayvan göçü optimizasyon algoritması
HSKA	:	Hibrit sinüs kosinüs optimizasyon algoritması
İHGA	:	İyileştirilmiş hayvan göçü optimizasyon algoritması
KAA	:	Karga arama algoritması
KKO	:	Karınca kolonisi optimizasyonu
MSE	:	Ortalama kare hatası
PSO	:	Parçacık sürü optimizasyonu
SKA	:	Sinüs kosinüs algoritması
SMS	:	Maddenin halleri arama algoritması
YA	:	Yusufçuk algoritması
YSA	:	Yapay sinir ağları
YSH	:	Yapay sinir hücreleri

ŞEKİLLER LİSTESİ

Şekil 3.1. İnsan sinir ağı (1, 3) ve yapay sinir ağı (2, 4) (Matarollo ve ark., 2013).	9
Şekil 3.2. Biyolojik sinir ağının modellenmesi (Alpay).....	10
Şekil 3.3. Üç katmanlı ve dört katmanlı YSA modeli (Kızrak ve Bolat, 2018).	11
Şekil 3.4. İleri beslemeli sinir ağının modellenmesi (Öztemel, 2006)	12
Şekil 3.5. Çıkış katmanından giriş katmanına doğru geri beslemeli bir yapay sinir ağı örneği (Kabalıcı, 2014)	13
Şekil 3.6. Çok katmanlı algılayıcı yapısı (Kabalıcı, 2014).....	14
Şekil 3.7. Basamak (Step) fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018)....	14
Şekil 3.8. Doğrusal (Linear) fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018) .	15
Şekil 3.9. Sigmoid fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018).....	16
Şekil 3.10 Hiperbolik tanjant fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018)	17
Şekil 3.11 (a) Bireysel Salp, (b) Salp Sürüsü (Salp Zinciri) (Mirjalili ve ark., 2017)....	18
Şekil 3.12. Salp sürü algoritmasının sözde kodu.....	20
Şekil 3.13. Salp sürü algoritmasının akış diyagramı	20
Şekil 4.1. İyileştirilmiş salp sürü algoritmasının akış diyagramı	30
Şekil 4.2. İyileştirilmiş salp sürü algoritmasının akış diyagramı	31
Şekil 4.3. İSSA-ÇKA çalışma prensibi	32
Şekil 5.1. Algoritmaların yakınsama grafikleri	38
Şekil 5.2. Algoritmaların kutu grafikleri	39

ÇİZELGELER LİSTESİ

Çizelge 3.1. Sınıflandırma veri kümelerinin özellikleri	21
Çizelge 5.1. Sınıflandırma veri kümelerinin özellikleri	35
Çizelge 5.2. Eğitim verilerine ilişkin MSE sonuçlarının ortalama ve standart sapması	35
Çizelge 5.3. Test verilerinde ortalama sınıflandırma doğruluğu	36
Çizelge 5.4. Algoritmaların ortalama çalışma süreleri (saniye)	37
Çizelge 5.5. En iyi sınıflandırma doğruluk oranı (%)	37
Çizelge 5.6. Algoritmaların sınıflandırma oranına göre ortalama sıralaması (Friedman testi)	39
Çizelge 5.7. Xor veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları	40
Çizelge 5.8. Balon veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları ...	40
Çizelge 5.9. İris veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları	40
Çizelge 5.10. Meme Kanseri veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları.....	41
Çizelge 5.11. Kalp veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları...	41

1. GİRİŞ

Yapay Sinir Ağları (YSA), insan beyninin çalışma yapısı temel alınarak oluşturulmuş, makinelere karar/tahmin yeteneği kazandıran bir bilgi işlem teknolojisidir. Bu teknoloji ile makineler veri sınıflandırma, tahmin etme, algılama ve optimizasyon gibi alanlarda kullanılmaktadır. YSA insan beynindeki sinir ağlarının yapısı gibi, yapay bir sinir ağı oluşturarak günlük yaşamdaki birçok soruna en uygun çözümü bulmaktadır (Şen, 2004). Optimizasyon alanında da kullanılan YSA verimli ve başarılı sonuçlar vermektedir.

21. yüzyılın ilk çeyreği biterken teknolojinin de üstün bir seviyeye geldiğini görmekteyiz. Bu yükseliş ile beraber günlük yaşam sorun ve problemlerini çözmek için teknolojiden yararlanmak kaçınılmaz olmaktadır. Ancak klasik yöntemler yetersiz kaldığı için makinelere insanlar gibi düşünme özelliği getirmek amaçlanmıştır. "Makineler düşünebilir mi?" sorusundan yola çıkarak 1943 yılında Alan Mathison Turing 'Bilgisayar Kavramları' ve 'Yapay Zekâ' terimlerinin öncüsü olmuştur (Pirim, 2006). Bu yıllarda yapay zekâ çalışmaları daha az önem taşıyordu çünkü klasik yöntemler insan problemlerini çözmekte yeterliydi. Zaman ilerledikçe günlük yaşam problemleri de daha zorlu ve karmaşık hale gelmişti. Artık insan düşünce sistemi buna yetmiyordu. Bu sorunları çözmek için makinelerden yardım alınması gerekiyordu. Çünkü makineler insanlardan çok daha hızlı çalışıyor, daha hızlı sonuç verebiliyor ve bu işleri daha düşük maliyet ile gerçekleştiriyordu. İlerleyen zaman içerisinde yapay zekâ birçok alt dallara ayrılmaktadır: Doğal dil işleme, yapay sinir ağları, görüntü işleme, uzman sistemler (günümüzde borsa ve hava durumu gibi veriye dayalı tahmin işlemlerinde sıkça kullanıldığı gözlemlenmektedir.), genetik algoritmalar ve bulanık mantık.

Makinelere insanlar gibi düşünme özelliğini kazandıran yapay sinir ağı insanların beyin sinir sistemlerinin çalışmasından esinlenerek oluşturulmuştur. İnsan beyin sinir sistemindeki nöronlar modellenerek yapay sinir hücrelerini ve bu nöronların arasındaki sinaptik bağların modellenmesi ile yapay sinir ağları oluşur. Yapay sinir ağları sınıflandırma, sistem modelleme (Erdal Tümer ve ark., 2020), yüz tanıma (Sarkar ve KB, 2020), konuşma tanıma (Patel ve ark., 2021) ve optimizasyon (Madenci ve Gülcü, 2020) gibi alanlarda kendisini ispatlamıştır.

Nöronlar birbirleri arasında bağ kurarak öğrenirler, hafızaya alırlar ve/veya gelen veriler arasında çıkarımda bulunurlar. Sonuç çıkarımlarına göre kendilerini

tekrardan eğitir, günceller ve düzenler. Tam bu noktada nasıl bir insan kendini geliştirmek, öğrenmek ve eğitmek için görsel, işitsel, dokunsal yollar ile bilgi alıyorsa, yapay sinir ağları da mühendisler tarafından geliştirilen öğrenme yollarını kullanmaktadır. Literatürde bu öğrenme yollarından genel olarak en çok, çok katmanlı algılayıcılar (ÇKA) kullanılmaktadır. Çok katmanlı algılayıcıların yapısı 3 ana bölümden oluşur: Giriş, gizli (ara) ve çıkış katmanlarıdır. Giriş ve çıkış katmanları tek katmandan oluşurken. Gizli katman ise bir veya birden çok katmandan oluşabilmektedir (Irmak ve Gülcü, 2021). Literatüre bakıldığında ÇKA eğitiminde kullanılabilen birçok meta sezgisel algoritma önerildiği gözlemlenmektedir. Önerilen algoritmalar yapay sinir ağının eğitiminin doğruluğunu ve hızını geliştirmek için yapılmıştır. Yapılan çalışmalar benchmark testleri, günlük yaşam problemleri vb. testlerden geçirilir ve sonuçlarına göre değerlendirilirler.

1.1. Tezin Amacı ve Önemi

Makine öğreniminin bir alt dalı olan YSA; kendisine verilen verileri kullanarak tahmin etmek ve sınıflandırılma yapmak için literatürde sıklıkla kullanıldığını gördüğümüz bir yöntemdir. Tahmin ve sınıflandırma yaparken en çok kullanılan metotlardan biri çok katmanlı algılayıcılarıdır.

Problem çözebilme yeteneği kazandırdığımız YSA'nın sonuçlarının optimum olması ve sadece soyut problemlere değil, gerçek dünya problemlerine de yüksek başarı gösterilmesi amaçlanmıştır. Alışılmış teknikler ile eğitilen YSA; yüksek hesaplama süresi, yüksek hata payı ve yüksek miktarda hafızaya ihtiyaç duymaktadır. Bu zorlukların üstesinden gelmek için meta sezgisel algoritmalar YSA eğitiminde kullanılmaya başlanmıştır (Jaddi ve Abdullah, 2018). Sürü zekâsına dayalı bu meta sezgisel yöntemler bias ve ağırlık değerlerini alışlagelmiş tekniklere göre daha iyi optimize etmektedir.

Meta sezgisel yöntemlerin ilk ortaya çıkışı ise teknolojik gelişmelerin ve yapay zekânın da ışığında 1990'lara dayanmaktadır (Albayrak ve Özdemir, 2016). Canlıların biyolojik veya yaratılışlarından sahip oldukları özelliklerin; hareket etmek, yemek yemek, barınmak, üremek, sürü halinde dolaşmak, haberleşmek gibi vb. özelliklerinden esinlenilerek oluşturulmuştur. Canlılardan ilham alınarak oluşturulan bu yöntemler gerçek dünya problemlerine çözüm getirmek için kullanılmaya başlanmıştır.

Bu tezde YSA'yı eğitmek için kullanılacak sürü zekâsı tabanlı olan salp sürü algoritması (SSA) Seydedali Mirjalili ve arkadaşları tarafından 2017 yılında doğadan

ilham alınarak geliştirilmiştir (Mirjalili ve ark., 2017). SSA, salpların zincir haline gelerek koordineli bir şekilde hareket etmeleri ve daha hızlı besin arayışlarından esinlenilmiştir.

Bu tezde, YSA'nın eğitilmesi, bias ve ağırlık değerlerinin optimum değerlere güncellenmesi için SSA algoritmasının iyileştirilmesi amaçlanmıştır.

YSA'yı eğitmek için literatürde birçok algoritma vardır. Bu algoritmaların amacı ağırlık ve bias değerlerini optimum hale getirmektir. Araştırmalara göre bias ve ağırlık değerlerini optimuma getirme konusunda en başarılı algoritmaların meta sezgisel algoritmalar olduğu görülmektedir (Gülcü, 2021; 2022). Ancak bazı meta sezgisel algoritmalar yerel optimuma takılma, erken yakınsama, arama uzayını tam keşfedememe gibi sorunlardan dolayı istenilen başarıyı elde edememiştir. Bu yüzden literatürdeki meta sezgisel algoritmalar ele alınmış, incelenmiş ve yerel optimumdan kaçma, sömürü ile keşif arasındaki dengeyi sağlama konusunda başarısını ispat etmiş olan SSA algoritması ele alınmıştır. Tez çalışmasında, elde edilen sonuçlara göre farklı arama stratejileri denenerek SSA'yı iyileştirme çalışmaları veya denklem güncelleme ve kaotik haritalar yardımı ile YSA'nın eğitimindeki başarının artırılması amaçlanmaktadır.

YSA'nın performans başarısında optimum ağırlık ve bias değerlerini bulmak önemli rol oynamaktadır. Optimum değerlerle eğitilen YSA'nın sınıflandırma değerlerini bulmada daha başarılı sonuçlara ulaştığı gözlemlenmiştir.

Yukarıda bahsedilen YSA'nın hangi alanlarda kullanıldığına baktığımızda oldukça geniş bir kullanım alanı olduğunu görmekteyiz. Bu nedenle YSA'nın performansını arttırmak oldukça önem arz etmektedir. Bu tez çalışmasında geliştirilmiş olan iyileştirilmiş salp sürü algoritması optimizasyonu (İSSA) ile YSA'nın performansı arttırılmaktadır. Böylece YSA'dan daha doğru geri dönütler alma hususunda başarılı sonuçlar elde edilmiştir. Makine öğrenmesinde de kullanılan YSA tez çalışmasında geliştirilmiş olan İSSA yöntemi sayesinde makine öğrenmesi alanına da önemli katkılar sağlamaktadır.

1.2. Tezin Organizasyonu

Bu tez çalışması 6 ana bölümden oluşmaktadır ve bu bölümler aşağıdaki şekilde organize edilmiştir.

Birinci bölümde, tez çalışmasının konusu hakkında giriş kısmı bulunmaktadır. Tez çalışmasının önemi ve bu tez çalışmasının literatüre katkılarına değinilmektedir.

İkinci bölümde, tez konusuyla ilgili literatürdeki mevcut farklı çalışmalar incelenmektedir.

Üçüncü bölümde, tez çalışmasında kullanılan materyal ve metotlar anlatılmaktadır. Öncelikle yapay sinir ağının tanımı, nereden esinlendiği, kullanım alanlarıyla ilgili bilgi verilerek çok katmanlı algılayıcılar ve parametrikleri anlatılmaktadır. Ardından önerilen İSSA'nın yapı taşı olan SSA algoritması detaylı bir biçimde anlatılmaktadır. Sonrasında İSSA'nın deneysel çalışmalarında kullanılan sınıflandırma veri setleri anlatılmaktadır. En son, tez çalışmasında kullanılan istatistiksel başarımlar metrikleri hakkında detaylı bilgi verilmiştir.

Dördüncü bölümde, YSA'yı eğitmek için geliştirilmiş olan İSSA algoritması ve YSA'nın İSSA algoritması ile eğitimi (İSSA-ÇKA) detaylı bir biçimde anlatılmıştır.

Beşinci bölümde, geliştirilen algoritmaların deneysel sonuçları bulunmaktadır. Literatürde sıklıkla kullanılan beş veri seti üzerinde ve iki farklı algoritma ile kıyaslanmaktadır. İstatistiksel başarımlar metrikleri ile test edilen İSSA-ÇKA algoritmasının başarılı sonuçlar verdiğini görülmektedir.

Altıncı bölümde, tez çalışması olan İSSA-ÇKA algoritmasının sonuçları özetlenmiş olarak sunulmuştur. Ayrıca, tez konusuyla alakalı ileride yapılabilecek çalışmalar hakkında öneriler verilmektedir.

2. KAYNAK ARAŞTIRMASI

Sürü zekâsına dayalı çalışan meta sezgisel algoritmalar genellikle canlıların yaşamlarını sürdürebilmesi için yaptıkları eylemlerden yararlanmaktadır. Bu algoritmalarda canlıların besin arama, konum değiştirme, haberleşme, saldırı/savunma düzeni, üreme vb. gibi eylemlerini arama uzayını keşfetme, en iyi konumdaki bireyi takip etme veya küçük gruplar halinde arama uzayını sömürmek gibi modellere simule etmektedir (Çelik ve ark., 2019). Her problemin farklı çözümleri olduğu için, bir algoritmanın tüm problemleri çok iyi bir şekilde çözebileceği bir durum yoktur. Bu yüzden literatürde birden çok algoritma bulunmaktadır. Problemin türüne ve amacına göre farklı algoritmalar kullanılmaktadır. Giriş bölümünde de değinildiği gibi YSA'nın eğitiminde meta sezgisel algoritmalar kullanılmaktadır ve son yıllarda bu konuda yapılan çalışmalar aşağıda sunulmaktadır.

YSA eğitmek için yapılan çalışmalardan biri gri kurt optimizasyon algoritmasıdır. Meta-sezgisel yöntemlerle sıkça çalışan Seyedali Mirjalili ve arkadaşları tarafından (2015) oluşturulan bu çalışma literatürde oldukça popüler bir çalışmadır. Önerilen algoritmanın test edilmesi için beş sınıflandırma ve sekiz standart veri kümesi kullanılmıştır. Popüler olarak görülen algoritmalara göre gayet güzel sonuçlar vermiştir. GKO-ÇKA'nın sonuçları diğer algoritmalara göre ya daha iyi ya da rekabetçi sonuçlar vermiştir (Mirjalili, 2015).

Literatürde YSA eğitimi için kullanılan bir başka çalışmada güve alevi algoritmasıdır. MFO, Waleed Yamany ve arkadaşları tarafından (2015) oluşturulan bu çalışma beş farklı popüler veri seti ile beş farklı optimizasyon algoritması ile test edilmiştir. Algoritmalarından dördünü gözle görülecek düzeyde geçmektedir. Genetik algoritma (GA) ile rekabetçi bir durumda olduğu gözlemlenmiştir (Yamany ve ark., 2015a).

Son yıllarda popülerliğini arttıran bir çalışmada Alboaneen ve arkadaşları tarafından sunulan (2017) Glowworm optimizasyon algoritmasıdır. YSA'yı eğiten bu algoritma beş farklı veri seti ile test edilmiştir. Sonuçları dört farklı optimizasyon algoritması ile karşılaştırılmış. Algoritma yüksek başarı gösteremese bile geçerli denilebilecek düzeydedir (Alboaneen ve ark., 2017). Yapılan çalışmanın geliştirilebilecek bir yapıya sahip olması ileriye dönük avantajdır.

YSA eğitiminde sıklıkla ilgilenen Heidari ve arkadaşları (2015) aslan karınca optimizasyon algoritmasını önerdiler. Yaptıkları çalışma on altı veri seti üzerinde test

edilmiş, dört optimizasyon algoritması ile kıyaslanmıştır. Kıyaslama sonucunda algoritmanın iyi bir başarı elde ettiği görülmektedir. On altı veri setinin sekiz tanesinde rakiplerini geride bırakan AKO-ÇKA algoritması, geriye kalan sekiz test sonucunda ise rekabetçi bir performans göstermiştir (Heidari ve ark., 2020).

Son beş yıl içerisinde yapılan çalışmalardan birisi de hibrit sinüs kosinüs optimizasyon algoritmasıdır. Gupta ve Deep (2020) küresel optimizasyon için HSCA ile ÇKA'yı eğitmişlerdir. Bu çalışma erken yakınsamayı önlemek için ve sömürü ile keşif arasındaki dengeyi kurmak için sıcaklık parametresi eklenmiştir. Eklenen parametre ile HSCA, orijinal SCA ya göre arama uzayını daha efektif kullanmaktadır. Böylece erken yakınsamayı önlemektedir. HSCA, beş sınıflandırma veri seti ile test edilip orijinal SCA algoritması ve üç farklı algoritma ile kıyaslanmıştır. Kıyaslama sonucunda HSCA'nın üstün başarısı görülmektedir. HSCA'nın gerçek dünya problemlerine ait çözüm sonuçları ele alındığında da yine yüksek bir performans verdiği gözlemlenmektedir (Gupta ve Deep, 2020).

Doğadan ilham alınarak geliştirilen bir başka çalışma çekirge optimizasyon algoritmasıdır. Çekirgelerin yiyecek kaynağı arama davranışlarından esinlenilerek oluşturulmuştur. Heidari ve arkadaşları tarafından (2018) geliştirilen ÇOA-ÇKA, YSA'nın ağırlık ve bias değerlerini ÇOA ile optimize ederek oluşturulmuştur. Bu çalışma beş sınıflandırma veri seti üzerinde, sekiz optimizasyon algoritması ile karşılaştırılmış ve algoritmanın rekabetçi bir yapıda olduğu gözlemlenmiştir (Heidari ve ark., 2019).

Literatürde YSA eğitmek için yapılan hibrit çalışmalar da vardır. Bu çalışmalardan birisi Kiranyaz ve arkadaşlarının oluşturduğu (2009) çok boyutlu parçacık sürü optimizasyon algoritması (ÇBPSO) ile YSA'nın eğitilmesidir. Çalışmada orijinal PSO'nun sömürü etkisini azaltılıp keşfe yöneltilmiştir. PSO'ya özel boyut özelliği kazandırılması ile oluşan ÇBPSO, arama uzayını konumsal ve boyutsal olmak üzere iki yönlü keşfetmektedir. Önerilen teknik zorlu testlerden başarı ile geçmiştir. Diğer gelişmiş algoritmalarla kıyaslanmasında başarılı sonuçlar vermektedir (Kiranyaz ve ark., 2009).

Mirjalili ve arkadaşlarının 2012 de yaptıkları çalışmada YSA'nın eğitilmesi PSO ile GSA'nın hibrit hali olan PSOGSA ile sunulmuştur. Bu çalışmada Mirjalili ve arkadaşları tarafından geliştirilen bu algoritma, arama uzayındaki çözüme yakın olan ajanlar, diğer ajanları kendilerine çekerek sömürüyü arttırmaktadır. Bu sebepten dolayı erken yakınsama oluşacağından Mirjalili ve arkadaşları bu durumun önüne geçmek ve

keşiften düşmemek için çözüm değeri iyi olan ajanları keşfe yönlendirirken diğer ajanları çok yavaş hareketlerle iyi çözüm bulan ajanlara doğru yaklaştırmaktadır. Bu algoritma içerisinde önemli iki parametre bulunmaktadır. Bu parametre değerlerine göre algoritmanın keşfe ya da sömürüye yönelmesi düzenlenebilir. Yapılan algoritma 3 farklı algoritma ile kıyaslanmıştır. Kıyaslamalar doğrultusunda PSOGSA'nın üstün başarısı görülmektedir (Mirjalili ve ark., 2012).

Literatürde meta sezgisel algoritmalarından biri olan karınca kolonisi algoritması (KKO) 2015 yılında Yamany ve arkadaşları tarafından geliştirilerek KKO-ÇKA algoritması sunulmuştur. Bu çalışma dört farklı veri seti ile test edilip üç algoritma ile kıyaslanmıştır. Kıyaslama sonucunda KKO'nun diğer algoritmalara göre üstün başarı elde ettiği görülmektedir (Yamany ve ark., 2015b).

Yapay sinir ağı eğitimi için yapılan çalışmalardan biri de 2020 yılında Şaban Gülcü tarafından oluşturulmuştur. Bu çalışmada yapay sinir ağı maddenin halleri arama algoritması ile eğitilmiştir. Deneysel, geliştirilen algoritma UCI platformundan alınan farklı veri setleri üzerinde ve altı farklı optimizasyon algoritması ile kıyaslanmıştır. Kıyaslama sonucunda çoğu optimizasyon algoritmasını üstün bir başarı ile geçmiştir. Diğerleri ile ise rekabetçi bir yaklaşım gösterse de başarılı sonuçlar vermiştir (Gulcu, 2020).

Çok katmanlı algılayıcıların eğitimlerinde kullanılan ve yeni yapılan meta sezgisel çalışmalardan birisi de karga arama algoritmasıdır. Erdoğan ve Gülcü'nün yaptığı bu çalışma kargaların besin depolaması ve depodan besinlerin geri alınmasından esinlenilmiştir. KAA-ÇKA eğitiminin deneysel testlerinde beş farklı veri seti kullanılmıştır. KAA-ÇKA algoritması ortalama hata karesi, sınıflandırma oranı, istatistiksel metrikler (duyarlılık, özgülük, kesinlik, fl-skor) ve yakınsama grafiği açısından MHA-ÇKA ile karşılaştırıldı. Ayrıca önerilen KAA-ÇKA algoritması en iyi sınıflandırma doğruluğu açısından literatürdeki yedi farklı algoritma ile de karşılaştırılmıştır. Deneysel sonuçlar, karga arama algoritmasının çok katmanlı algılayıcı eğitiminde kaliteli bir yaklaşım olduğunu göstermektedir (Erdogan ve Gulcu, 2021).

Literatürdeki YSA eğitimlerinde gradyan tabanlı yaklaşımların yerel optimuma takıma ve yetersiz keşif yeteneği gibi dezavantajlarından dolayı YSA eğitiminde meta sezgisel bir yöntem olan hayvan göçü optimizasyon algoritmasıdır (HGA). Şaban Gülcü bu algoritmayı levy uçuşu ile hibritleyerek ÇKA eğitimlerinde kullanmıştır. IHGA-ÇKA algoritması beş veri seti üzerinde test edilmiştir. Sonuçlar on farklı algoritma ve

bir gerek dnya problemi zerinde kıyaslanmıřtır. nerilen IHGA-KA algoritması bařarısını ispatlamıřtır (Glc, 2021).

řaban Glc'nn KA eęitimini gerekleřtirmek iin sunduęu bir bařka yntem meta sezgisel olan yusufuk algoritmasıdır. Yaptıęı alıřmada YA-KA'nın aęırlık ve bias deęerlerini bařarılı bir řekilde optimize ettięi grlmřtr. YA-KA'nın test ve kıyaslamasında sekiz farklı veri seti ile test edilmiř ve drt algoritmanın sonuları ile kıyaslanmıřtır. Kıyaslama sonucunda YA-KA verimli sonular elde etmiřtir (Glc, 2022).



3. MATERYAL VE YÖNTEM

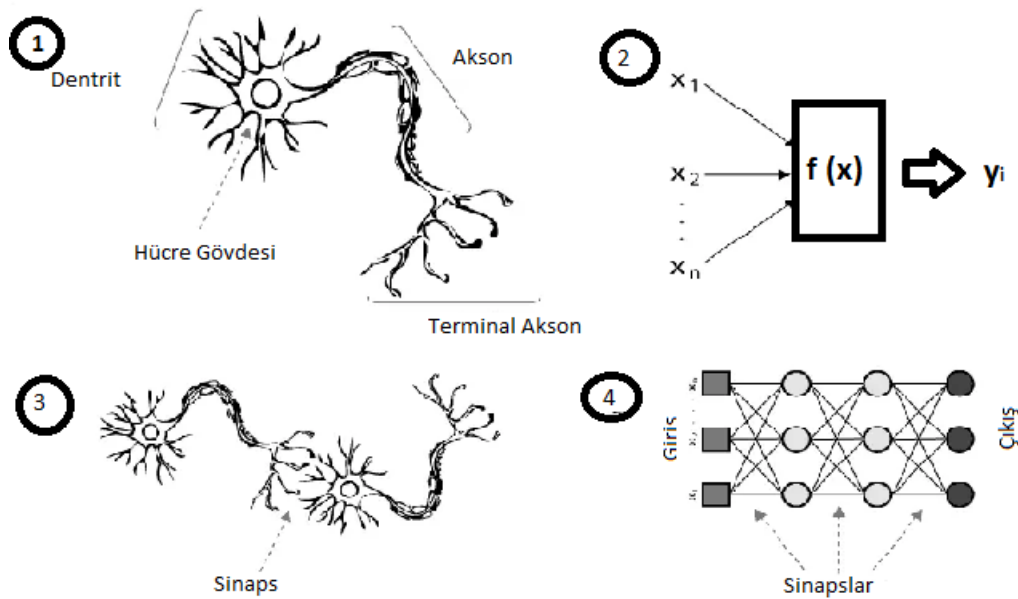
Bu bölümde yapay sinir ağları, çok katmanlı algılayıcılar, çok katmanlı algılayıcılarda kullanılan fonksiyonlar, salp sürü optimizasyon algoritması ve iyileştirilmiş salp sürü algoritması detaylı olarak anlatılmıştır.

3.1. Yapay Sinir Ağı (YSA)

Yapay sinir ağları (YSA), insan beynindeki nöronların çalışma prensibi ele alınarak oluşturulan, mekanik sinir ağıdır. Nöronların birbiri ile gerçekleştirdikleri iletişim ağının bilişim dünyasına uyarlanmış şeklidir (Öztürk ve Şahin, 2018).

Sağlıklı bir insan beyninde milyarlarca nöron bulunmaktadır. Bu nöronlar birbirleri ile sadece tek yönlü değil, her yöne doğru iletim yapmaktadır. Bir nöron; çekirdek, hücre gövdesi, dentrit ve aksondan oluşmaktadır. Dentritler hücre gövdesi etrafında binlerce dallardan oluşmaktadır. Dentritler bir önceki nörondan gelen veriyi girdi bilgisi olarak almaktadır.

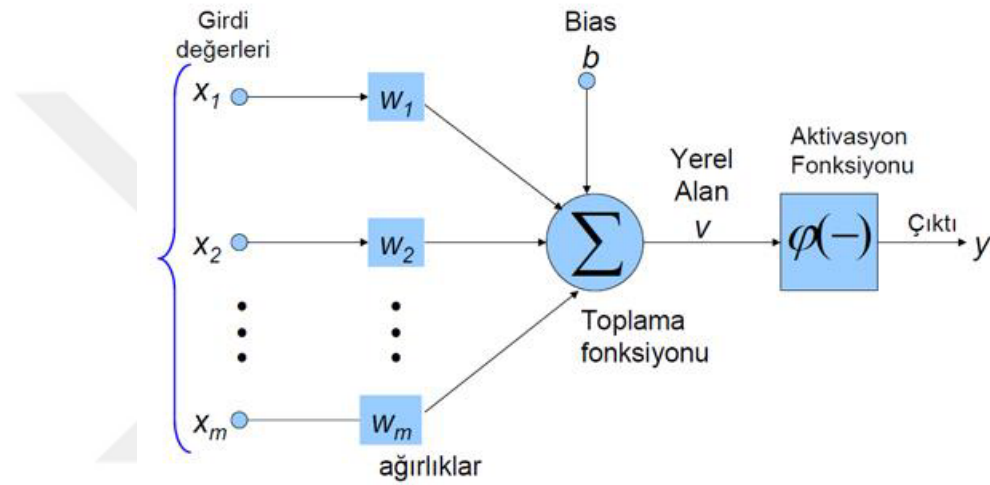
Aksonlar, çekirdekten çıkan verilerin bir sonraki nöron hüresine çıktı verisinin iletilmesini sağlamaktadır. Aksonlar ve dentritlerin birleşip veri aktardığı noktaya ise sinaps denilmektedir. Şekil 3.1’de görüldüğü gibi nöronlar işlem elemanı, dentritler toplama fonksiyonu, hücre gövdesi aktivasyon fonksiyonu, aksonlar eleman çıkışları ve sinapslar ağırlıkları temsil etmektedir (Zhang, 2018).



Şekil 3.1. İnsan sinir ağı (1, 3) ve yapay sinir ağı (2, 4) (Maltarollo ve ark., 2013).

İnsan beynindeki nöronların birbiri ile etkileşiminden esinlenilerek yapay sinir ağı oluşturulmuştur. Nöronun girdi bilgisi ile aksonlar üzerindeki ağırlık değerleri çarpılır daha sonra birleşme fonksiyonu ile veriler birleştirilir. Çıkan veri aktivasyon fonksiyonundan geçerek yeni çıktı elde edilir. Son nörona gelene kadar bu işlemler tekrardan gerçekleşir. Son nörondan çıkan sonuç net çıktıdır.

İnsan beynindeki nöronlardan esinlenerek oluşturulan yapay sinir ağının içerisindeki yapay sinir hücresinin görsel şeması Şekil 3.2’de gösterilmiştir. Bu yapay sinir hücreleri (YSH) bir araya gelerek yapay sinir ağını oluşturmaktadır.

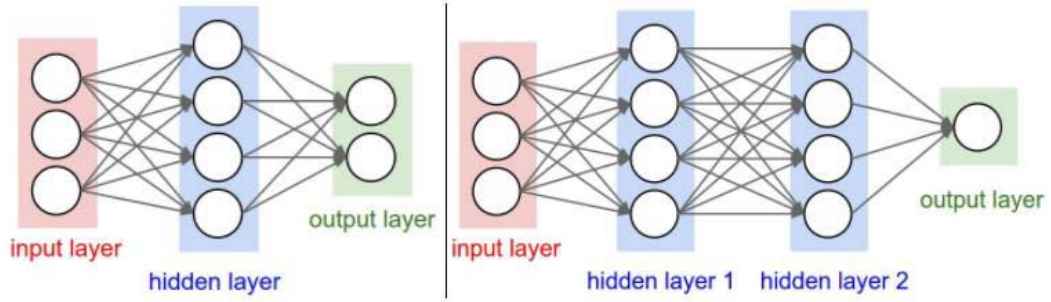


Şekil 3.2. Biyolojik sinir ağının modellenmesi (Alpay).

Yapay sinir hücreleri gruplanarak katmanlar oluştururlar. Bu katmanlar; girdi katmanı, gizli katman (ara katman) ve çıktı katmanı olmak üzere 3 gruba ayrılır. Şekil 3.3’de YSA’nın genel bir yapısı gösterilmiştir. YSA’da giriş katmanı ve çıkış katman sayısı her zaman 1’dir. Gizli katman ise birden fazla olabilir. YSA’nın katman özellikleri şu şekildedir:

- Giriş katmanı: Veri girişinin olduğu katmandır. Bu katmanda matematiksel veya herhangi bir işlem yapılmaz, tüm hücrelerin bir adet girdisi ve bir adet çıktısı vardır.
- Gizli katman/katmanlar: Giriş katmanından veya kendinden önceki gizli katmandan gelen verileri işleyerek, işlenmiş veriyi kendinden bir sonraki gizli katmana veya çıkış katmanına aktarırlar. Giriş katmanı ile çıkış katmanı arasındaki iletişimi sağlar. Yukarıda da belirttiğimiz gibi YSA içerisinde bir veya birden fazla gizli katmana sahip olabilir.

- Çıkış katmanı: Gizli katmandan gelen verileri alır ve YSA dışına çıkararak katmandır.



Şekil 3.3. Üç katmanlı ve dört katmanlı YSA modeli (Kızrak ve Bolat, 2018).

YSA'lar genellikle tek katmanlı algılayıcılar ve çok katmanlı algılayıcılar olmak üzere iki grupta incelenir.

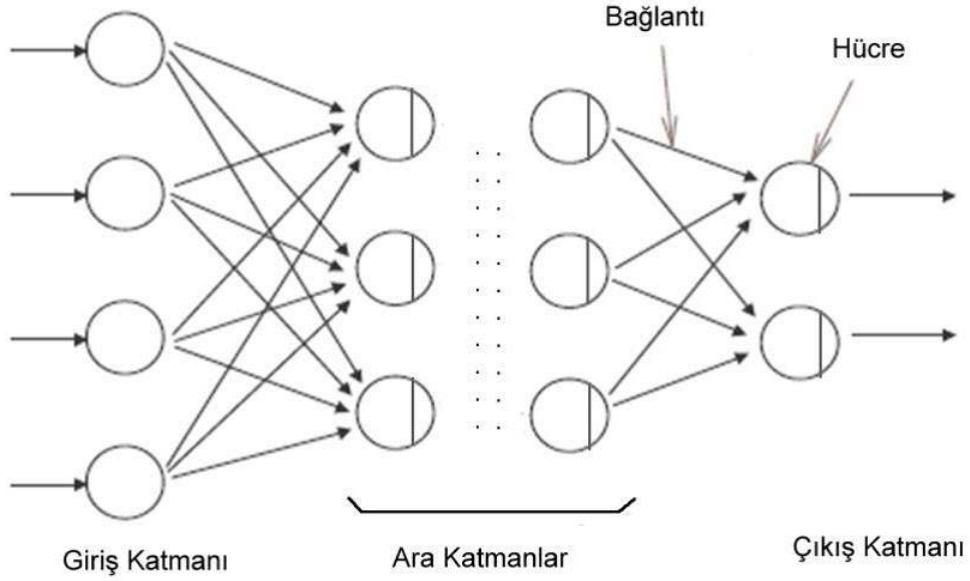
Tek katmanlı algılayıcılar, giriş ve çıkış katmanlarından oluşur. Bu ağ yapısında çıktının sıfır olmasını önleyen bir eşik değeri bulunmakta. Bu ağ yapısında girdi değerleri ile eşik değeri toplanır, ardında elde edilen veri aktivasyon fonksiyonuna girer ve ağın çıktısı hesaplanır. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusaldır. Çıkış değeri olarak 1 veya -1 değerini alır.

Çok katmanlı algılayıcılar (ÇKA) ise giriş, çıkış ve gizli katmandan oluşur. Yukarıda da belirttiğimiz gibi giriş ve çıkış katmanları bir katmandan oluşur. Gizli katman ise bir veya birden fazla katmana sahip olabilirler. Tek katmanlı algılayıcıların aksine aktivasyon fonksiyonları doğrusal değildir.

3.1.1. Çok katmanlı algılayıcı (ÇKA)

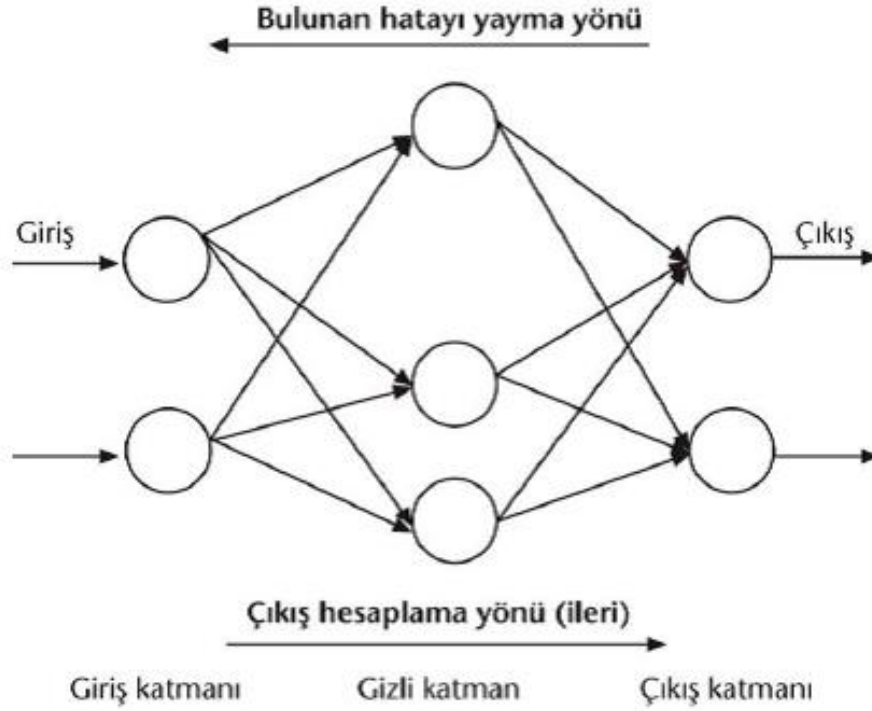
Çok katmanlı algılayıcılar, giriş katmanı ile çıkış katmanı arasında bir veya birden fazla ara katmandan oluşmaktadır. Literatürde ÇKA'ler genel olarak iki gruba ayrılırlar. Bunlar ileri beslemeli ve geri beslemeli YSA'lardır.

- İleri beslemeli sinir ağları: giriş katmanından çıkış katmanına doğru yalnızca tek yönlü olarak ilerlemektedir. Şekil 3.4'de görüldüğü üzere bir giriş ve bir çıkış katmanı vardır. Gizli katman sayısı ise bir veya birden fazladır (Wilamowski ve ark., 2003).



Şekil 3.4. İleri beslemeli sinir ağının modellenmesi (Öztemel, 2006)

- Geri beslemeli yapay sinir ağı: YSA'da ez az bir YSH çıkışı bir başka YSH'nin girişi veya aynı YSH'nin girişi olarak kullanılır. Geri besleme katmanlar arasındaki YSH arasında olduğu gibi aynı katman içerisindeki YSH arasında da olabilir. Bu özelliğinden dolayı doğrusal olmayan yapıya sahiptir. Şekil 3.5'de üç katmanlı, çıkış katmanından giriş katmanına doğru geri beslemeli bir yapay sinir ağı örneği gösterilmektedir.



Şekil 3.5. Çıkış katmanından giriş katmanına doğru geri beslemeli bir yapay sinir ağı örneği (Kabalcı, 2014)

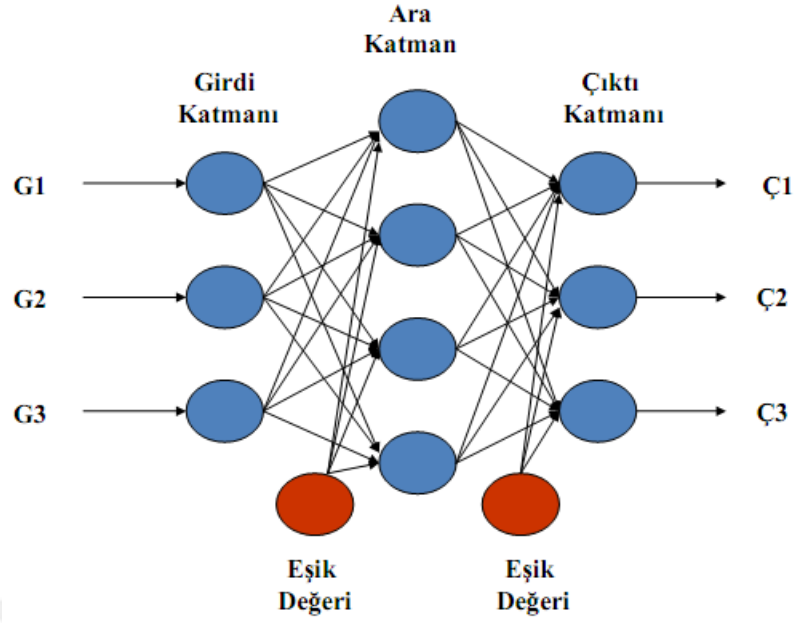
Geri yayılım algoritması yukarıda da belirttiğimiz gibi girişlerle çıkışlar arasında hata değeri bularak ağırlıkları bu hata değerlerine göre güncellemektedir. Hata değeri şu şekilde gösterilir. Denklem (3.1)'de hata değerinin bulunması için kullanılan matematiksel formül gösterilmiştir.

$$\text{Hata Değeri} = e$$

$$\text{İstenilen çıkış değeri} = z \quad \Rightarrow \quad e_t = z_t - y_t; t = 1, \dots, n \quad (3.1)$$

$$\text{YSA'nın çıkış değeri} = y$$

Şekil 3.6'da girdi verilerinin ağırlıklarla çarpıldıktan sonra çıkan değerlerin toplandığı fonksiyona toplama (birleşme) fonksiyonu denir. Toplama fonksiyonundan çıkan veriler aktivasyon fonksiyonuna gönderilerek yeniden işlenir. Yukarıda da belirttiğimiz gibi literatürde aktivasyon fonksiyonuna aynı zamanda eşik değeri fonksiyonu da denilmektedir. Eşik fonksiyonu denilmesinin sebebi çıktı verilerinin genellikle $[-1,1]$ veya $[0,1]$ aralığında bulunmasındandır. Literatürde birçok aktivasyon fonksiyonu bulunmaktadır. Bunlardan en çok kullanılanları ise; basamak (step) fonksiyonu, doğrusal (linear) fonksiyon, sigmoid fonksiyonu, hiperbolik tanjant fonksiyon.

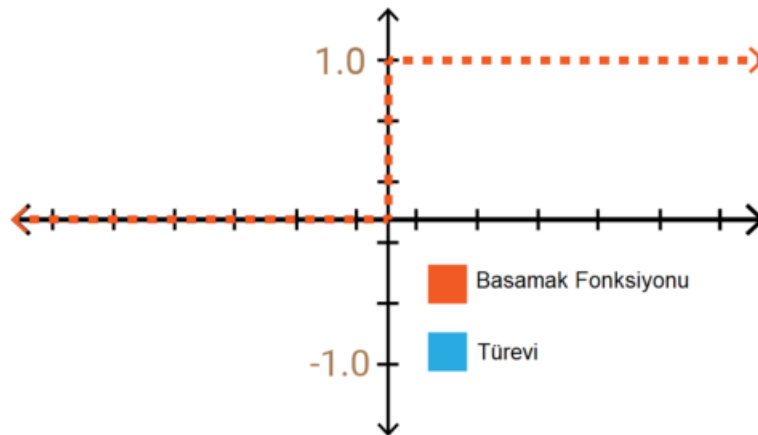


Şekil 3.6. Çok katmanlı algılayıcı yapısı (Kabalcı, 2014).

Basamak (Step) Fonksiyonu:

Adından da anlaşılacağı üzere koordinat sistemi üzerinde basamak şeklinde gözükten bu fonksiyon, ikili değer almaktadır. Yani diğer bir ifadeyle, fonksiyon çıkışı 0 veya 1 olur. Bu durumdan dolayı ikili sınıflandırıcı olarak kullanılmaktadır. Denklem (3.2)'de basamak fonksiyonunun eşitliği gösterilmiştir. Şekil 3.7'de basamak fonksiyonunun koordinat düzleminde örnekleme yapılmıştır.

$$f(x) = \begin{cases} 0 & \text{için } x < 0 \\ 1 & \text{için } x \geq 0 \end{cases} \quad (3.2)$$



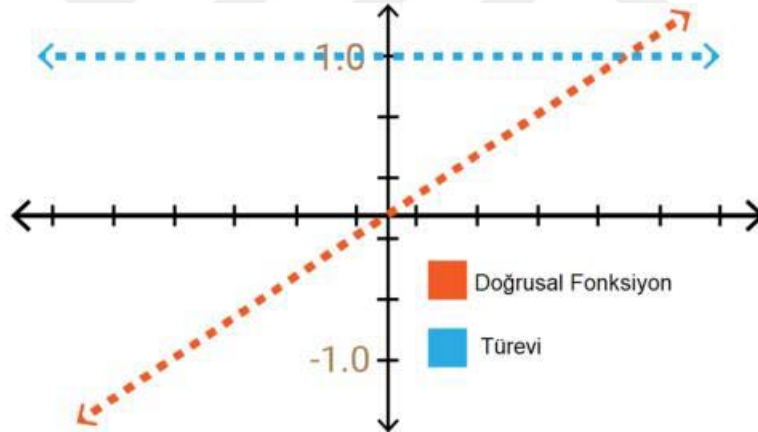
Şekil 3.7. Basamak (Step) fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018)

Doğrusal (Linear) Fonksiyonu:

Step fonksiyonunun aksine yalnızca 0 ve 1 değerlerini üretmez. Birden çok aktivasyon çıkışı verebilir. Fakat bu fonksiyonun bir sorunu var. Doğrusal fonksiyon türevi sabittir. Geri yayılım aşaması türev alan sistemlerden oluşmaktadır. Bu durumda geri yayılımda türevi alınan veri hep sabit değer verecektir. Bu sebepten dolayı da öğrenme gerçekleşmeyecektir. Aynı zamanda başka bir sorun daha var: Yapay sinir ağında tüm katmanlarda doğrusal fonksiyon kullanılacak olursa giriş katmanından gelen veriler doğrusal olarak çıkış katmanına geçecektir. Bu durumda da ÇKA sanki bir tek katmanlı algılayıcı gibi çalışacaktır. Denklem (3.3)'de doğrusal fonksiyonun denklemi ve türevinin denklemi gösterilmiştir. Şekil 3.8'de doğrusal fonksiyonunun koordinat düzleminde örnekleme yapılmıştır.

$$f(x) = x \quad (3.3)$$

$$f'(x) = 1$$



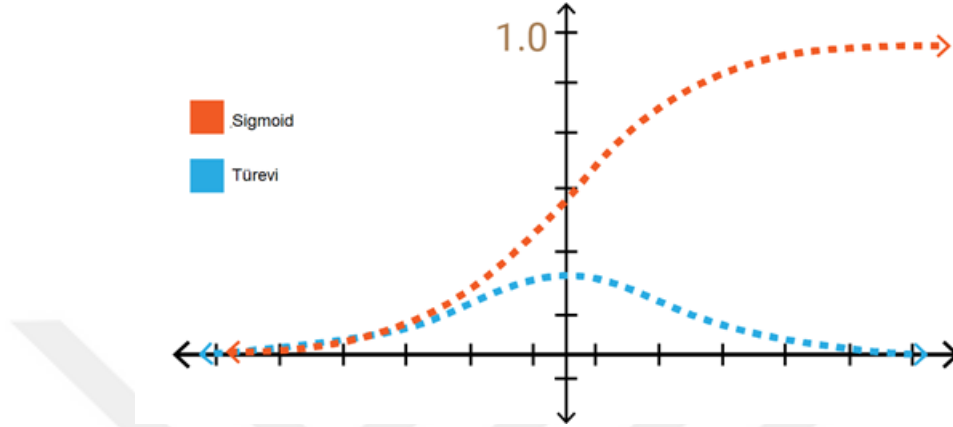
Şekil 3.8. Doğrusal (Linear) fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018)

Sigmoid Fonksiyonu:

Klasik algoritmaların veya programların çözemediği, kalitesiz çözümler verdiği, yüksek bellek ve zaman ihtiyacı duymasının sebebi, problemlerin doğrusal olmamasındandır. Sigmoid fonksiyonu ne basamak fonksiyonu gibi sabit çıktılar verir ne de doğrusal fonksiyon gibi türevi alındığında sabit değer verir. Sigmoid fonksiyonunda da bir sorun var. Bu sorun giriş değeri çok büyük ya da sıfır olduğu zaman, fonksiyonunun gradyanın sıfır olmasıdır. Böylece geriye doğru gradyan ilerlemez. Literatürde bu duruma genellikle “doymuş gradyan problemi” denilmektedir. Denklem (3.4)'de sigmoid fonksiyonunun eşitliği gösterilmiştir. Şekil 3.9'da sigmoid

fonksiyonunun koordinat düzleminde (0, 1) aralığında sınırlandırılmış grafiği verilmiştir.

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \quad (3.4)$$

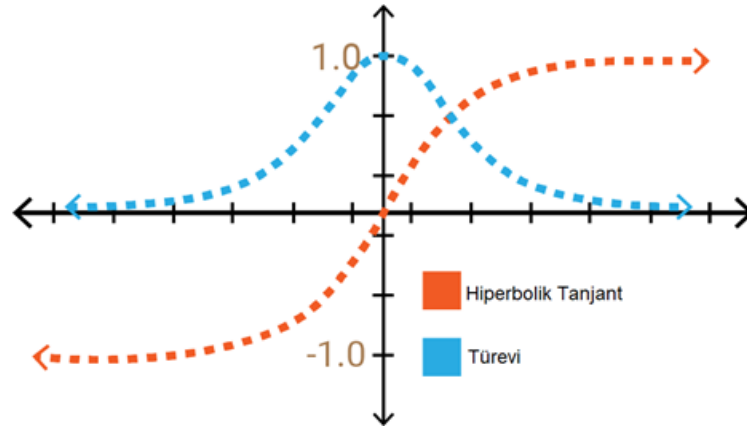


Şekil 3.9. Sigmoid fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018).

Hiperbolik Tanjant Fonksiyonu:

Bu aktivasyon fonksiyonu sigmoid fonksiyonuna yakın bir yapıya sahiptir. Lakin fonksiyonun aralığı (-1, 1) aralığında tanımlıdır. Hiperbolik tanjant fonksiyonunun türevi daha dik olduğu için daha fazla değer alabilmektedir. Bu durum ise sigmoid fonksiyonuna göre daha avantajlıdır. Sigmoid fonksiyonundaki doymuş gradyan problemi, fonksiyonun uçlarında görülmektedir. Denklem (3.5)'de hiperbolik tanjant aktivasyon fonksiyonunun eşitliği verilmiştir. Şekil 3.10'da hiperbolik tanjant aktivasyon fonksiyonunun (-1, 1) değer aralığında koordinat düzleminde grafiği verilmiştir.

$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (3.5)$$



Şekil 3.10 Hiperbolik tanjant fonksiyonu ve türevinin grafiği (Kızrak ve Bolat, 2018)

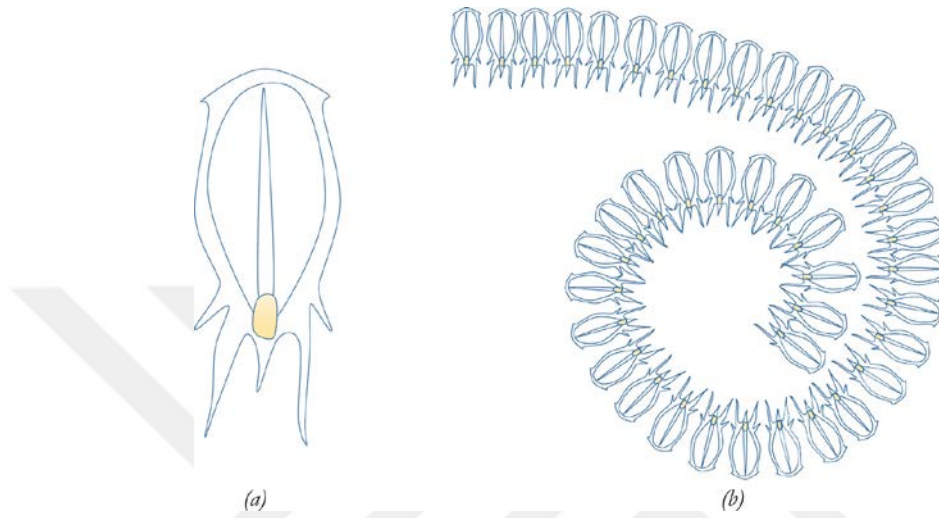
ÇKA, girdi değerleri ile ağırlıkları çarpılır ve ardından birleştirme fonksiyonuna gönderilir. Buradan gelen değer aktivasyon fonksiyonunu gönderilir ve aktivasyon fonksiyonundan sonuç değeri çıkar. Çıkan sonuç YSH'nın net çıktısını oluşturur. Yukarıda da bahsettiğimiz gibi aktivasyon fonksiyonu $[0,1]$ veya $[-1,1]$ aralıklarında değer çıkırları verdikleri için literatürde sıkıştırma veya eşit değeri de denilmektedir. Yukarıda literatürde sıklıkla kullanılan dört farklı aktivasyon fonksiyonunun açıklaması mevcuttur. Bu çalışmada eşik fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Sigmoid fonksiyonunun seçilmesinin sebebi literatürde ÇKA eğitiminde sıklıkla kullanılıyor olması ve türevinin alınabilir olmasıdır.

Hata fonksiyonu ÇKA'dan çıkan sonuç ile çıkması gereken sonucu alır. Fonksiyona gönderilen bu değerler, işlendikten sonra bir çıktı verir. Çıktı değerine göre geri yayılım yapılırken ÇKA güncellenir. Bu güncelleme işlemi ÇKA eğitiminin en önemli parçalarından biri olan geri yayılım algoritması ile öğrenirler. ÇKA'nın geri yayılım algoritması ile eğitilmesi üç aşamada gerçekleşir. Girdi verilerinin giriş katmanından çıkış katmanına doğru ilerlemesi, çıktı birimlerinde hata değerinin hesaplanması, geriye doğru yayılım yapılırken hata değerine göre ağırlıkların güncellenmesi. ÇKA'nın eğitimi tamamlandıktan sonra ÇKA doğrusal bir şekilde ileri yönlü çalışacaktır.

3.2. Salp Sürü Algoritması (SSA)

Seyedali Mirjalili ve arkadaşları tarafından 2017 yılında doğadan ilham alınarak oluşturulmuştur (Mirjalili ve ark., 2017). Sürü zekâsına dayalı olan SSA'nın oluşum amacı arama alanını verimli kullanarak yerel optimuma takılma ihtimalini azaltmaktır. Salp canlıları fiçî şeklinde saydam bir yapıya sahiptirler. Vücutları ile su pompalayarak

hareket ederler. Okyanuslarda yaşayan bu canlılar genellikle planktonlarla beslenirler. Derin okyanuslarda salplar genellikle salp zincirleri olarak adlandırılan bir sürü oluştururlar. Salp zincirleri oluşturmalarının amacı hızlı, koordineli yer değiştirme ve yem arama kolaylığıdır. Şekil 3.11’de salp zincirinin görünümüne bir örnek gösterilmiştir.



Şekil 3.11 (a) Bireysel Salp, (b) Salp Sürüsü (Salp Zinciri) (Mirjalili ve ark., 2017).

Salplar okyanuslarda gruplar halinde ilerler ve avlanırlar. Daha hızlı ilerlemek ve daha rahat yiyecek bulmak için okyanuslarda zincir şeklinde dolanırlar. Rastgele konumlarla başlayan salplar içerisinde, matematiksel modele göre sürü lideri seçilir. Geriye kalan salplar takipçi salp olarak adlandırılır. Takipçi salplar kendinden bir önceki salpları kademeli olarak takip ederek erken yakınsamayı engeller.

3.2.1. Salp sürüsünün hareketi

Salp zincirleri 2 gruba ayrılmaktadır;

- Lider salp, tüm salpların en önünde arama uzayını keşfeder.
- Takipçi salplar, kendisinden bir önceki salpı kademeli olarak takip eder

Salp zincirlerini matematiksel olarak modellemek için, popülasyon önce iki gruba ayrılır; **lider** ve **takipçiler**. Lider, zincirin önündeki salptır. Diğer salplar ise takipçiler olarak kabul edilir. Bu salpların adından da anlaşılacağı gibi, liderin önderliğinde ilerleyerek takipçiler birbirini kademeli olarak takip eder.

3.2.2. Lider salp için konum güncelleme denklemi

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j) * c_2 + lb_j c_3 < 0.5 \\ F_j - c_1((ub_j - lb_j) * c_2 + lb_j c_3 \geq 0.5 \end{cases} \quad (3.5)$$

Burada; x_j^1 j -inci boyuttaki lider salpın konumunu temsil eder. F_j j -inci boyuttaki hedef besin kaynağını, C_1 toplam adım sayısı ile o anki adım sayısının kademeli ilerleyişi, C_2 ve C_3 değerleri $[0, 1]$ aralığında üretilen rastgele sayıyı temsil etmektedir. ub_j ve lb_j sırasıyla j -inci boyuttaki üst ve alt sınırları temsil etmektedir.

3.2.3. SSA'da c_1 parametresi

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \quad (3.6)$$

Burada: l şuan ki adımı, L ise toplam adımı temsil eder.

C_1 parametresinin önemli olmasının nedeni toplam yapılan adım sayısı ile o anki adım sayısının orantısına göre arama uzayının global arama ve yerel arama bölgelerini dengeler.

3.2.4. Takipçi salpların konumunu güncelleme denklemi

$$x_j^i = \frac{1}{2}at^2 + v_0t \quad (3.7)$$

Burada: X_j^i j -boyuttaki i .salpın konumunu, t zamanı, v_0 başlangıç hızı temsil etmektedir.

α ve v ise $v = \frac{x-x_0}{t}$, $\alpha = \frac{v_{final}}{v_0}$ denkleminde elde edilen değerdir.

Takipçi salpların konumlarını güncelerken kullandığımız bu denklemde $v=0$ ve $i \geq 2$ olma durumunda Denklem (3.7) hata verecektir. Bu özel durum için farklı bir denklem kullanılmaktadır. Kullanılan denklem, Denklem (3.8)'de gösterilmiştir.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (3.8)$$

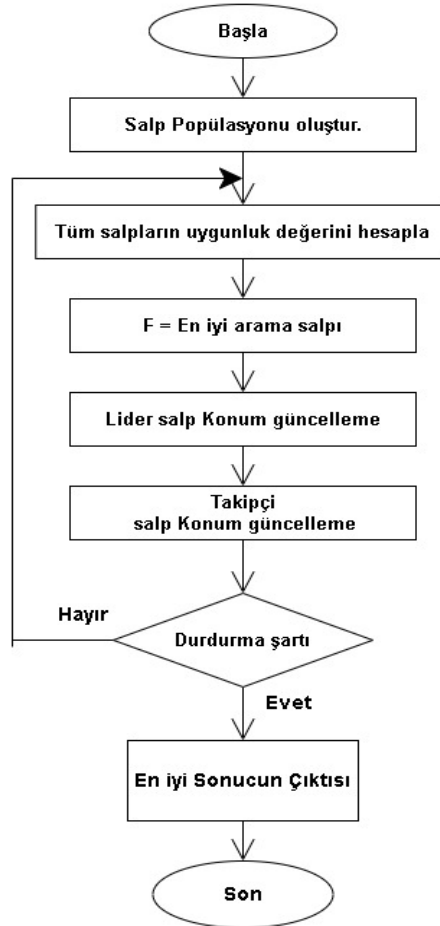
Burada: X_j^i j -inci boyuttaki i -inci takipçi salpın yerini temsil eder.

3.2.5. SSA sözde kodu ve akış diyagramı

SSA'nın sözde kodu Şekil 3.12'de ve akış diyagramı Şekil 3.13'de gösterilmektedir.

ub ve lb'yi dikkate alarak salp popülasyonunu $x_i (i = 1, 2, \dots, n)$ oluşturulur.
while (bitiş koşulu sağlanmadığı sürece devam edilir)
 Tüm arama ajanlarının (salp) uygunluğu hesaplanır.
 F=en iyi arama ajanı
 C_1 parametresi Denklem (3,6)'ya göre güncellenir.
For each salp (xi)
 if (i==1)
 Lider salpın konumu Denklem (3,5)'e göre güncellenir.
 else
 Takipçi salpların konumu Denklem (3,8)'e göre düzenlenir.
End
End
 Arama uzayından çıkan salplar, alt ve üst sınıra göre tekrar arama uzayına katılır.
End
 return F

Şekil 3.12. Salp sürü algoritmasının sözde kodu.



Şekil 3.13. Salp sürü algoritmasının akış diyagramı

3.3. Sınıflandırma Veri Setleri

İSSA-ÇKA optimizasyon algoritmasının başarı düzeyini test etmek ve kıyaslamak için literatürde sıklıkla kullanılan, farklı zorluk seviyelerine sahip beş veri

seti ile eğitildikten sonra test edilmiştir. Bu farklı zorluk seviyelerine sahip veri setleri şu şekildedir; xor, balon, iris, meme kanseri ve kalp. Bu veri setleri UCI Makine Öğrenim Deposu'ndan alınmıştır. Veri setlerinin özellikleri Çizelge 3.1'de gösterilmiştir. Çizelgeye bakıldığı zaman; xor veri seti üç nicelik iki sınıflandırma ve sekiz eğitim sekiz test sayısı ile en kolay sınıflandırma problem olduğu dikkat çekmektedir. Balon veri seti dört nitelik iki sınıflandırma sayısı ve yirmi eğitim, yirmi test örneği sayısı ile orta kolay bir veri seti olduğunu söyleyebiliriz. Iris veri seti dört nitelik üç sınıflandırma sayısı ve yüz elli eğitim, yüz elli test örneği sayısı ile orta seviyede bir sınıflandırma problemi olduğunu görmekteyiz. Meme kanseri veri seti dokuz nitelik sayısı, iki sınıflandırma sayısı ve yüz test, beş yüz doksan dokuz eğitim sayısı ile orta zor derecede sınıflandırma problemi olduğu dikkat çekmektedir. Kalp veri seti ise yirmi iki nitelik sayısı iki sınıf sayısı ve yüz seksen yedi test, seksen eğitim sayısı ile zor seviye bir sınıflandırma problemi olduğu görülmektedir. Bu çalışmadaki eğitim ve test veri setleri Seyedali Mirjalili'nin çalışmalarından alınmıştır.

Çizelge 3.1. Sınıflandırma veri kümelerinin özellikleri

Veri seti	Nitelik sayısı	Eğitim örneği sayısı	Test örneği sayısı	Sınıf sayısı
Xor	3	8	8	2
Balon	4	20	20	2
Iris	4	150	150	3
Meme kanseri	9	599	100	2
Kalp	22	80	187	2

3.4. İstatistiksel Başarım Metrikleri

3.4.1. Ortalama karesel hata (MSE)

Ortalama kare hatası veya ortalama sapması literatürde genellikle makine öğrenmesi modelinin başarımın performansını ölçer. Hata değerlerinin büyüklükleri birbirine benzer ise MSE kullanılır. Fakat bu değerlerden bir veya birden fazla değer ortalamasının üzerindeyse MSE kullanmak pek uygun bir yöntem olmaz. MSE'nin formülü Denklem (3.9)'da gösterilmektedir. Ayrıca MSE hataların karelerini aldığı için her zaman pozitifdir ve değeri sıfıra yakın olması performansının yüksek olduğunu belirtmektedir.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.9)$$

Burada N toplam eleman sayısını ifade etmektedir. i . eleman için y_i gerçek değeri, aynı şekilde i . eleman için \hat{y}_i tahmin değerini temsil etmektedir.

ÇKA çıktı katmanında birden fazla YSH olabileceği için İSSA-ÇKA eğitiminde Denklem (3.10)'da ki MSE formülü kullanılmıştır.

$$MSE = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^K (y_j^i - \hat{y}_j^i)^2 \quad (3.10)$$

Burada N Denklem (3.10)'daki gibi toplam eleman sayısını, K ise ÇKA'nın çıktı katmanındaki YSH'ları temsil etmektedir. i . eleman ve j . YSH için y_j^i gerçek değeri, i . eleman ve j . YSH için \hat{y}_j^i tahmin değerini temsil etmektedir.

3.4.2. Friedman testi

Friedman testi, bağımlı gruplar arasında normallik varsayımı sağlanmadığı durumlarda ortalama karşılaştırması yapabilmek amacı ile kullanılan bir istatistiksel analiz tekniğidir. Friedman testini kullanabilmek için bir adet grup (ya da blok) değişkeni; bir adet de ortalama karşılaştırması yapılabilecek ölçüm değişkeni bulunmalıdır. Ölçüm yapılacak olan değişken, sürekli bir veri ya da sıralayıcı ölçeğe sahip bir veri olmalıdır.

Bu test normal dağılıma uygun olmayan ölçümler için, istatistik analizi uygulamalarında k -adet ($k > 2$) eşleştirilmiş veya bağımlı grup arasındaki ortalamaları karşılaştırmak için kullanılmaktadır. Testin genel yapısına bakıldığında, Wilcoxon eşleştirilmiş işaretli sıra testinin genelleştirilmiş hali olduğu söylenebilir.

Wilcoxon testinde bağımlı veya eşleştirilmiş birimler arasında yalnızca iki grup arasında ortalama karşılaştırması yapılırken, bu test kapsamında en az üç grup arasında ortalama karşılaştırması yapılmaktadır (istmer).

Friedman testinin uygulanabilmesi için aşağıdaki temel varsayımların sağlanmış olması gereklidir:

- Grup içerisinde gözlemlerin bağımsız olması
- Bloklar arası gözlem gruplarının eşleştirilmiş ya da bağımlı bir yapıda olması
- Ortalama karşılaştırması yapılacak olan ölçüm değişkeninin sürekli veya sıralayıcı ölçekli bir yapıda olması
- Ortalama karşılaştırması yapılacak verilerin normal dağılıma uygun olmaması

- Bağımlı veya eşleştirilmiş verileri içeren blok sayısının ikiden fazla ($k > 2$) olması

3.4.3. Duyarlılık (Sensitivity)

Duyarlılık testi, diğer adıyla gerçek pozitif oranı olarak ta isimlendirilir. Bir koşulun olma durumunu bildiren bir testin doğruluğunu matematiksel olarak tanımlar. Bu testin duyarlılığının 100% olması, testin pozitif tahminlerinin başarılı olduğunu göstermektedir. Duyarlılık testinin formülü Denklem (3.11)'de gösterilmektedir.

$$\text{Duyarlılık} = \frac{DP}{DP+YN} \quad (3.11)$$

Burada DP, doğru pozitif ve YN, yanlış negatif değerini temsil etmektedir.

3.4.4. Özgüllük (Specificity)

Özgüllük testi, diğer adı ile gerçek negatif oranı olarak ta isimlendirilir. Bir koşulun olmama durumunu bildiren bir testin doğruluğunu matematiksel olarak tanımlamaktadır. Bu testin duyarlılığının 100% olması, testin Negatif tahminlerinin başarılı olduğunu göstermektedir. Duyarlılık testinin formülü Denklem (3.12)'de gösterilmektedir.

$$\text{Özgüllük} = \frac{DN}{DN+YP} \quad (3.12)$$

Burada DN, doğru negatifi ve YP, yanlış pozitif değerini temsil etmektedir.

3.4.5. Kesinlik (Precision)

Kesinlik, pozitif olarak tahmin ettiğimiz değerlerin, gerçekte kaç tanesinin pozitif olduğunu göstermektedir. Eğer kesinlik düşük ise çok sayıda hatalı pozitif olduğunu ifade eder. Kesinlik testinin formülü Denklem (3.13)'de gösterilmiştir.

$$\text{Kesinlik} = \frac{DP}{DP+YP} \quad (3.13)$$

Burada DP, doğru pozitif ve YP, yanlış pozitif değerini temsil eder.

3.4.6. F1-Score

F1-score, yukarıda bahsettiğimiz duyarlılık ve kesinlik değerlerinin harmonik (değerlerin terslerinin aritmetik ortalaması) ortalamasıdır. Sınıflandırıcının ne kadar iyi performans gösterdiğinin bir ölçüsüdür ve sınıflandırıcıları karşılaştırmakta literatürde sıklıkla kullanılmaktadır. Denklem (3.14)'de f1-score formülü gösterilmiştir.

$$f1 - score = 2 * \frac{kesinlik * duyarlılık}{kesinlik + duyarlılık} \quad (3.14)$$



4. ÖNERİLEN YÖNTEM

4.1. İyileştirilmiş Salp Sürü Algoritması (İSSA)

Bölüm 3’de anlatılan SSA algoritmasının başarılı bir çalışma olduğu ve geliştirilmeye açık olduğu gözlemlenmektedir. SSA algoritmasının lokale takılma ve iterasyonlar boyu arama uzayının keşfinden düşmesi gibi dezavantajlarının olduğu tespit edilmiştir. Bu sorunların üstesinden gelmek için literatürdeki çalışmalar incelendi ve literatürde bu sorunlara karşı hibrit modeller, parametre analizleri vb. yöntemler kullanıldığı görülmüştür. Bu tez çalışmasında bu sorunları çözmek için salpları Johannes Kepler’ın "*Geometrinin iki büyük hazinesi vardır. Bunlardan biri Pythagaros kuramı, öteki de bir çizginin eşit ve ortalama orana bölünmesidir*" (Tekkanat, 2006) diyerek belirttiği altın orana göre bölerek büyük parçayı literatürde en çok kullanılan on kaotik haritalar yardımı ve trigonometrik denklemler ile birlikte lider salp konum güncelleme denklemine yönlendirilmiştir. Arama uzayına daha fazla hâkim olabilmek için ise takipçi salpların bir kısmı rassal olarak güncellenmiştir.

SSA algoritmasında salplar iki grup halinde (lider ve takipçi salplar) ayrılarak ilerlemesi ile birlikte lider salplar Denklem (3.5) ile en iyi çözüm değerine ulaşmaya çalışmaktadır. Denklem (3.6)’da c_1 parametresi ile başlangıçta yüksek adımlarla ilerleyen lider salplar ilerleyen iterasyonlarda arama uzayı keşfinden düşerek sömürüye yönelmektedir. Arama uzayının keşfi ve sömürü yeteneği arasındaki dengeyi korumak için lider salp güncelleme denklemi ile birlikte trigonometrik lider salp konum güncelleme denklemi kullanılmaktadır. Lider salpların arama uzayında hareket yönlerini belirlemek için kaotik haritalardan yararlanıldı. Takipçi salpların kendinden bir önceki salplara doğrusal (Linear) yaklaşımı başlangıçta kademeli ilerleme için faydalı olarak görülse de ilerleyen iterasyonlarda sömürüye yönelen lider salplardan dolayı takipçi salplar birbirlerini daha ufak adımlarla takip edecektir. Bu sorunu çözmek için zincir şeklindeki salpların son sıralarda ki salplarına rassallık katmak için, Hüseyin Haklı ve arkadaşları tarafından (2015) oluşturulan ABCVSS (Kiran ve ark., 2015) çalışmasından ilham alınarak oluşturulan denklem ile güncelleştirilerek hem arama uzayında daha geniş alanlara yönelmesini hem de yerel optimumda sıkışma ihtimalini azaltmıştır.

Seydali Mirjalili’nin yapmış olduğu SSA çalışmasında salplar iki eşit gruba ayrılmaktadır. Bunların yarısı lider konumunu güncelleme denklemine diğer yarısında takipçi salp konum güncelleme denklemine gitmektedir. Bu tez çalışmasında ise salplar, beş eşit gruba ayrılmaktadır. Birinci ve ikinci gruptaki salplar kendi içerisinde altın

orana göre ayrılarak büyük olan parça lider salp sürü güncelleme denklemine, küçük olan parça ise takipçi salp konum güncelleme denklemine gitmektedir. Üçüncü ve dördüncü gruplar ise aynı şekilde kendi içerisinde altına orana bölünür ve büyük parça trigonometrik lider salp konum güncelleme denklemine, küçük parça ise takipçi salp konum güncelleme denklemine gitmektedir. Geriye kalan beşinci grup son sıralardaki salpların ilerleyen iterasyonlarda keşiften düşmemesi ve rassallık sağlamak için lider salpların konumları ve trigonometrik lider salp konumları ile rastgele bir biçimde konumlandırılmaktadır. Denklem (4.1)'de altın oranın formülü verilmiştir.

$$\text{Altın oran} = \frac{1+\sqrt{5}}{2} \quad (4.1)$$

4.1.1. Kaotik haritalar

Kaotik haritalar, karmaşık problemleri kurgulama, sayısal analiz yöntemlerinde, kriptolojide, meta sezgisel optimizasyon algoritmalarında vb. yerlerde sıklıkla kullanılmaktadır. Meta sezgisel algoritmalarda rassal sayı dizelerinin aynı olmaması, farklı değerler alması ve fazla zamana ihtiyaç duymaması algoritmanın verimliliği için önemli bir durumdur. Meta sezgisel algoritmaların yerel optimuma sıkışmasını önlemek için literatürde sıklıkla kullanılan yöntemlerden biri de kaotik haritalardır (Balakrishnan ve ark., 2019). Bu tez çalışmasında literatürde sıkça kullanılan on kaotik harita seçildi ve kullanıldı. Seçilen kaotik haritalar şunlardır:

Chebyshev haritası: Denklem (4.2)'de chebyshev haritasının formülü gösterilmektedir.

$$x_{k+1} = \cos(k \cos^{-1}(x_k)) \quad (4.2)$$

Circle haritası: Denklem (4.3)'de circle haritasının formülü gösterilmektedir.

$$x_{k+1} = x_k + 0.2 - (0.5 - 2\pi x_k) \text{mod}(1) \quad (4.3)$$

Gauss haritası (fare haritası): Denklem (4.4)'de gauss haritasının formülü gösterilmektedir.

$$x_{k+1} = \begin{cases} 0 & x_k=0 \\ \frac{1}{x_k \text{mod}(1)} & \text{değilse} \end{cases} \quad (4.4)$$

İterative haritası: Denklem (4.5)'de iterative haritasının formülü gösterilmektedir.

$$x_{k+1} = \sin \frac{0.7\pi}{x_k} \quad (4.5)$$

Logistic haritası: Denklem (4.6)'da logistic haritasının formülü gösterilmektedir.

$$x_{k+1} = 4x_k(1 - x_k) \quad (4.6)$$

Burada x_k sıfır ile bir arasında bir sayı olduğu için popülasyonun mevcut olan maksimum popülasyona oranını temsil etmektedir.

Piecewise haritası: Denklem (4.7)'de piecewise haritasının formülü gösterilmektedir.

$$x_{k+1} = \begin{cases} \frac{x_k}{\pi}, & 0 \leq x_k \leq \pi \\ \frac{x_k - \pi}{0.5 - \pi}, & \pi \leq x_k \leq 0.5 \\ \frac{1 - \pi - x_k}{0.5 - \pi}, & 0.5 \leq x_k \leq 1 - \pi \\ \frac{1 - x_k}{\pi}, & 1 - \pi \leq x_k \leq 1 \end{cases} \quad (4.7)$$

Bu denklemde π 'nin değeri $(0, 0.5]$ aralığındadır.

Sine haritası: Denklem (4.8)'de sine haritasının formülü gösterilmektedir.

$$x_{k+1} = \frac{\alpha}{4} \sin(\pi x_k) \quad (4.8)$$

Singer haritası: Denklem (4.9)'da singer haritasının formülü gösterilmektedir.

$$x_{k+1} = 1.07(7.86x_k - 23.31x_k^2 + 28.75x_k^3 + 13.302875x_k^4) \quad (4.9)$$

Sinusoidal haritası: Denklem (4.10)'da sinusoidal haritasının formülü gösterilmektedir.

$$x_{k+1} = \sin(\pi x_k) \quad (4.10)$$

Tent haritası: Denklem (4.11)'de tent haritasının formülü gösterilmektedir.

$$x_{k+1} = \begin{cases} \frac{x_k}{0.7}, & x_k < 0.7 \\ \frac{10}{3}(1 - x_k), & x_k \geq 0.7 \end{cases} \quad (4.11)$$

Burada x_k değerine bağlı olarak tent haritası kaotik olana kadar bir dizi dinamik davranış gösterir.

4.1.2. Lider salp ve trigonometrik salp konum güncelleme

Lider salp konum güncelleme denklemi SSA çalışmasında Denklem (3.5)'de olduğu gibidir. Lakin Denklem (3.5)'de c_3 parametresi (0,1) aralığında random bir sayı değeri almaktadır. İSSA tez çalışmasında c_3 parametresini her iterasyonda yukarıda belirttiğimiz kaotik haritalardan birini rastgele seçerek, lider salpların konumları güncellenmektedir. Bu yöntem ile yerel en iyi noktaya takılma ihtimalini azaltılmaktadır.

Trigonometrik lider salp konum güncelleme denklemi ise salpların arama uzayı içerisinde sömürü değerini arttırmak için Denklem (3.5)'de kullanılan c_2 parametresini güncelleyerek sinüs ve kosinüs algoritması kullanılmıştır. Denklem (4.12)'de trigonometrik lider salp konum güncelleme denklemini gösterilmiştir.

$$x_j^i = \begin{cases} F_j + \sin(c_2) * |(c_1 * F_j) - x_j^i| & c_3 < 0.5 \\ F_j + \cos(c_2) * |(c_1 * F_j) - x_j^i| & c_3 \geq 0.5 \end{cases} \quad (4.12)$$

Burada; x_j^i j-inci boyuttaki i-ninci salpın konumunu temsil eder. F_j j-inci boyuttaki hedef besin kaynağını, C_1 toplam adım sayısı ile o anki adım sayısının kademeli ilerleyişimi, $C_2 = 2\pi * rand$ ve C_3 değerleri yukarıda anlatılan rastgele seçilen kaotik haritalardan dönen değeri temsil etmektedir.

4.1.3. Takipçi salp konum güncelleme

Denklem (3.8)'de orijinal SSA çalışmasındaki takipçi salp konum güncelleme denklemi kullanılmıştır. SSA çalışmasında salplar eşit bir şekilde bölünerek lider ve takipçi salp konum güncelleme denklemlerine gitmektedir. İSSA tez çalışmasında beşe bölünen salplar 3 grup halinde ilerlemektedir. Birinci ve ikinci parçalar birinci grup, üçüncü ve dördüncü parçalar ikinci grubu oluşturmaktadır. Birinci ve ikinci gruptaki

salplar kendi içerisinde altın orana göre ayrılarak dağılmaktadır. Büyük salp grubu lider, küçük salp grupları ise takipçi salp konum güncelleme denkleminde gösterilmektedir.

4.1.4. Rassal pozisyon güncelleme

SSA algoritmasında arama uzayında rastgele bir şekilde başlatılan salplar, ilk iterasyonlarda arama uzayının keşfine yönelmektedir. İlerleyen iterasyonlarda keşiften hızla düşmektedir. Bu durumun önüne geçebilmek için literatürdeki çalışmalardan ilham alınarak yeni denklem oluşturulmuştur. Bu denkleme salp zincirinin son halkalarında ki salplar kullanılmıştır. Denklem (4.13)'de rassal pozisyon güncelleme denklemi gösterilmektedir.

$$x_j^i = \frac{1}{2} (x_j^{Lider\ salp(r)} + c_2 * (x_j^{trigonometrik\ salp(r)} - x_j^i)) \quad (4.13)$$

Burada x_j^i j -inci boyuttaki i -inci takipçi salpın yerini temsil etmektedir. $x_j^{Lider\ salp(r)}$ j -inci boyuttaki rastgele bir lider salpı, $x_j^{trigonometrik\ salp(r)}$ ise j -inci boyuttaki rastgele bir trigonometrik lider salpın konumunu temsil etmektedir.

4.1.5. İSSA sözde kodu ve akış diyagramı

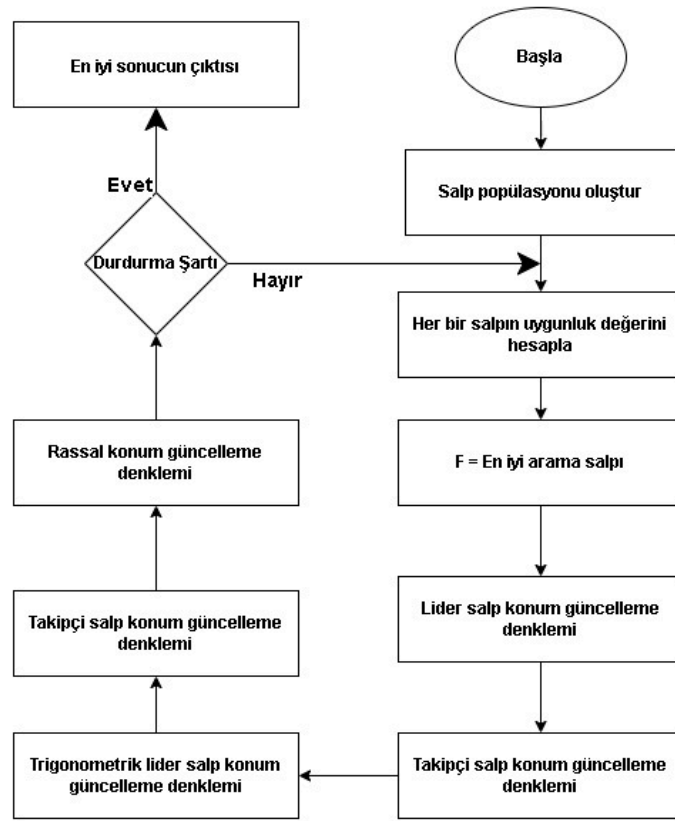
İSSA'nın sözde kodu Şekil 4.1'de ve akış diyagramı Şekil 4.2'de gösterilmektedir.

```

ub ve lb'yi dikkate alarak salp popülasyonunu  $x_i (i = 1, 2, \dots, n)$  oluşturulur.
while (bitiş koşulu sağlanmadığı sürece devam edilir)
Tüm arama ajanlarının (salp) uygunluğu hesaplanır.
F=en iyi arama ajanı
 $C_1$  parametresi Denklem (3,6)'ya göre güncellenir.
N=popülasyon boyutu
 $NN = ((2 * N / 5) / ((1 + \sqrt{5}) / 2))$ 
For each salp (xi)
  if ( $i \leq NN$ )
    Lider salpın konumu Denklem (3,5)'e göre güncellenir.
  elseif ( $i > NN \ \&\& \ i \leq (2 * N) / 5$ )
    Takipçi salpların konumu Denklem (3,8)'e göre düzenlenir.
  elseif ( $i > (2 * N) / 5 \ \&\& \ i \leq ((2 * N) / 5 + NN$ )
    Trigonometrik lider salpların konumu Denklem (4,12)'e göre düzenlenir
  elseif ( $i > ((2 * N) / 5 + NN) \ \&\& \ i \leq (4 * N) / 5$ )
    Takipçi salpların konumu Denklem (3,8)'e göre düzenlenir.
  else
    Salpların konumu Denklem (4,13)'e göre düzenlenir.
  End
End
Arama uzayından çıkan salplar, alt ve üst sınıra göre tekrar arama uzayına katılır.
End
return F

```

Şekil 4.1. İyileştirilmiş salp sürü algoritmasının akış diyagramı



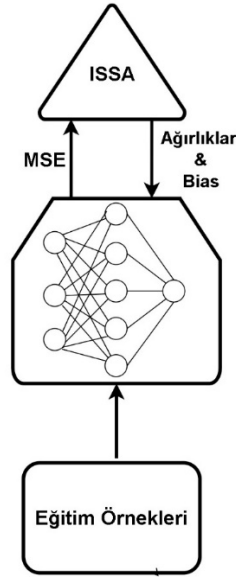
Şekil 4.2. İyileştirilmiş salp sürü algoritmasının akış diyagramı

4.2. YSA'nın İyileştirilmiş Salp Sürü Algoritması İle Eğitimi (İSSA-ÇKA)

YSA'lar eğitim süreçleri boyunca doğrusal olmayan yapıda ilerlemektedir. Eğitim tamamlandığında doğrusal bir yapıya geçer ve verilen testleri aynı şekilde sonuçlandırır. Test aşamasına geçmeden önce YSA'lar doğru bir şekilde eğitilmelidir ki başarılı sonuçlar verebilsin. Eğitim aşamasında YSA'lar ağırlık ve bias değerlerini günceller. Başlangıçta rastgele atanan bu değerler eğitim sonunda test edilmektedir. Literatürde ÇKA eğitimini gerçekleştirmek için birçok algoritma önerilmiştir. Önerilen klasik algoritmaların kalitesiz çözüm ve yüksek maliyet gerektirdiği için YSA eğitimlerinde meta sezgisel algoritmalar kullanılmaya başlanmıştır. Bu tez çalışmasında ÇKA eğitimlerinde, ağırlık ve bias değerlerini optimum noktaya taşımak için İSSA algoritması geliştirilmiştir. İSSA-ÇKA algoritmasının çalışma prensibi Şekil 4.3'de gösterilmiştir.

ÇKA'da YSH'lar arasındaki ağırlıklar ve katmanlarda kullanılan bias değerini optimum noktaya getirmek için öncelikle İSSA-ÇKA yapısında bias ve ağırlıkları

matematiksel olarak tanıtmamız gerekir. Bias ve ağırlıkları temsil etmek için vektör denklemini kullanılmaktadır ve Denklem (4.14)'de gösterilmiştir.



Şekil 4.3. İSSA-ÇKA çalışma prensibi

$$\text{Vektör} = (w_{g,a} \cup w_{a,\zeta} \cup v_{0,a} \cup v_{0,\zeta}) \quad (4.14)$$

Burada $w_{g,a}$ giriş katmanı ile ara katman arasındaki ağırlık değerlerini temsil etmektedir. $w_{a,\zeta}$ ara katman ile çıktı katmanı arasındaki ağırlık değerlerini temsil etmektedir. $v_{0,a}$ ara katmanda bulunan bias değerini, $v_{0,\zeta}$ ise çıkış katmanındaki bias değerini temsil etmektedir.

İSSA-ÇKA algoritması çalıştırıldığında salplar arama uzayı içerisinde rastgele oluşturulurlar. Her bir salp farklı ÇKA'yı temsil eder. Aday çözümü oluşturan her salp, tasarlanan ÇKA modelinde toplam bağlantı ağırlığı ve bias değeri boyutunun d boyutlu bir dizisi olarak temsil edilir ve Denklem (4.14)'e olduğu gibi çözüm vektörü oluşturulur.

Sonra eğitim veri seti kullanılarak salp zinciri içerisinde besin kaynağına en yakın olan salp seçilir. Ardından salplar gruplara ayrılarak konumlarını (ağırlık ve bias değerleri) Denklem (3.5), Denklem (3.8), Denklem (4.12) ve Denklem (4.13)'e göre günceller. Kaotik haritalar yardımı ile Denklem (3.5) ve Denklem (4.12)'de salpların tekrar eden değerlerin önüne geçer. Sonrasında takipçi salplar, kendilerinden bir önceki salpı takip ederler. Lokale takılma ve arama uzayı içerisindeki daha iyi çözümlere ulaşabilmek için

Denklem (4.13) kullanılarak arama uzayı içerisinde global en iyi deęerler hedeflenmektedir. Bu süreç sonlandırma şartı gerekleşene kadar devam eder.



5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde tez çalışmasında geliştirilmiş olan İSSA ve İSSA-ÇKA algoritmalarının deneysel sonuçları sunulmaktadır. Elde edilen sonuçlar yorumlanarak sunulmuştur. Deneysel çalışmada kullanılan donanım ve yazılımların özellikleri şu şekildedir: Microsoft Windows 10, Intel i7-10750h 2.6 -5.0 GHz, 16 GB bellek. Algoritmalar MATLAB R2016a'da kodlandı ve çalıştırıldı. Bu tez çalışmasındaki tüm istatistiksel analizler Microsoft Excel 2016 yazılımı ile yapılmıştır.

İSSA-ÇKA algoritmasının test etmek ve performansını ölçmek için UCI Makine Öğrenimi Deposu'ndan alınan beş farklı veri seti kullanılmıştır. Kullanılan veri setleri: Xor, balon, iris, meme kanseri ve kalp veri setleridir. Bölüm 3'de sınıflandırma veri setleri başlığı altında bu veri setlerinin özellikleri verilmiştir. İSSA-ÇKA algoritmasının başarısını görebilmek için literatürdeki başarılı farklı optimizasyon algoritmaları seçilmiştir ve ile kıyaslama yapılmıştır. Bu optimizasyon algoritmaları salp sürü optimizasyon algoritması (Mirjalili ve ark., 2017) tabanlı SSA-ÇKA ve maddenin halleri optimizasyonu algoritması (Cuevas ve ark., 2014) tabanlı SMS-ÇKA algoritmalarıdır. Veri setlerinin bias ve ağırlıkların başlangıç değeri literatürdeki çalışmalarla aynı olması için [-10,10] aralığında rastgele atanmıştır. Kıyaslama için kullanılan veri setlerinin parametreleri: nitelik sayısı, sınıf sayısı, ÇKA yapısı Vektör boyutu, eğitim ve test sayıları Çizelge 5.1'de verilmiştir.

Bu tez çalışmasında gizli katman sayısı literatürde en çok kullanılan $(2 \times n) + 1$ formülü ile belirlenmiştir. Formülde n ÇKA'nın girdi katmanındaki nöron sayısıdır. Veri setlerine göre ÇKA yapısına baktığımızda, xor veri seti üç girdi ile yedi gizli katmana sahiptir. Balon veri seti dört girdi ile dokuz, iris veri seti dört girdi ile dokuz, meme kanseri veri seti dokuz girdisi ile on dokuz, kalp veri seti ise yirmi iki girdi sayısı ile kırk beş ara katmana sahiptir. Çizelge 5.1'de veri setlerinin nitelikleri verilmiştir.

İSSA-ÇKA algoritmasının MSE sonuçları literatürdeki SSA-ÇKA ve SMS-ÇKA algoritmalarının MSE sonuçları ile kıyaslanmıştır. Çizelge 5.2'de algoritmaların MSE sonuçlarının ortalama ve standart sapması değerleri gösterilmiştir. Çizelge 5.2'de SMS-ÇKA algoritmasının sonuçları (Gulcu, 2020) yayınından alınmıştır.

Çizelge 5.2 incelendiğinde İSSA-ÇKA algoritmasının xor, iris ve meme kanseri veri setlerinde SSA-ÇKA ve SMS-ÇKA algoritmalarını geride bıraktığını görmekteyiz. Balon veri setinde İSSA-ÇKA ile SSA-ÇKA'nın rekabetçi bir yaklaşımda olduğunu

söylemek mümkündür. Kalp veri setinde ise SMS-ÇKA'nın İSSA-ÇKA'yı 0,38 ortalama MSE fark değeri ile geçtiğini görebiliriz.

Çizelge 5.1. Sınıflandırma veri kümelerinin özellikleri

Veri Setleri	Özellik Sayısı	Eğitim	Test	Sınıf Sayısı	ÇKA Yapısı	Vektör boyutu
Xor	3	8	8	2	3-7-1	36
Balon	4	20	20	2	4-9-1	55
Iris	4	150	150	3	4-9-3	75
Meme Kanseri	9	599	100	2	9-19-1	210
Kalp	22	80	187	2	22-45-1	1081

Çizelge 5.2. Eğitim verilerine ilişkin MSE sonuçlarının ortalama ve standart sapması

Veri Setleri		İSSA-ÇKA	SSA-ÇKA	SMS-ÇKA
Xor	Ortalama	1,82E-04	9,43E-03	1,33E-01
	Standart sapma	3,37E-04	2,98E-02	3,39E-02
Balon	Ortalama	7,41E-09	1,15E-10	1,11E-02
	Standart sapma	2,47E-08	3,51E-10	1,31E-02
Iris	Ortalama	2,28E-02	3,28E-02	2,58E-01
	Standart sapma	2,90E-03	5,90E-02	5,12E-02
Meme kanseri	Ortalama	1,47E-03	1,53E-03	2,27E-02
	Standart sapma	1,07E-04	2,05E-04	4,33E-03
Kalp	Ortalama	8,65E-02	8,27E-02	1,18E-01
	Standart sapma	1,91E-02	2,14E-02	3,13E-02

İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının test veri setlerinde ortalama sınıflandırma doğruluğu Çizelge 5.3'de sunulmuştur. Çizelge 5.3 incelendiğinde, xor veri setinde İSSA-ÇKA tam doğruluk ile başarısını göstermektedir. Balon veri setinde İSSA-ÇKA ve SSA-ÇKA algoritmaları tam doğruluk ile başarısını göstermektedir. Iris veri setinde en yüksek başarı oranına İSSA-ÇKA algoritması ulaşmıştır. Meme kanseri veri setinde İSSA-ÇKA algoritması SSA-ÇKA algoritması ile yüksek rekabet içerisindedir. Kalp veri setinde ise İSSA-ÇKA diğer algoritmalara kıyasla başarılı sonuç vermektedir. İSSA-ÇKA algoritma beş veri seti içerisinde xor, balon ve iris veri setlerinde yüksek başarı elde ettiği, meme kanseri veri setinde 0.56% fark ile SSA-ÇKA algoritması ile rekabetçi sonuçlar sergilediği gözlemlenmiştir. Kalp

veri setinde 72.58% sınıflandırma doğruluk ortalaması ile diğer algoritmaların önüne geçmektedir.

Çizelge 5.3. Test verilerinde ortalama sınıflandırma doğruluğu

Veri Setleri		İSSA-ÇKA	SSA-ÇKA	SMS-ÇKA
Xor	Minimum	100.00	75.00	75.00
	Maksimum	100.00	100.00	100.00
	Ortalama	100,00%	98,33%	83,75%
	Standart sapma	0.00	5.43	6.69
Balon	Minimum	100.00	100.00	85.00
	Maksimum	100.00	100.00	100.00
	Ortalama	100,00%	100,00%	99,17%
	Standart sapma	0.00	0.00	2.96
İris	Minimum	98.67	66.67	76.00
	Maksimum	99.33	99.33	93.33
	Ortalama	98,82%	97,78%	86,78%
	Standart sapma	0.29	5.88	4.90
Meme Kanseri	Minimum	97.00	98.00	61.00
	Maksimum	100.00	100.00	93.00
	Ortalama	98,67%	99,23%	85,67%
	Standart sapma	0.66	0.68	7.37
Kalp	Minimum	65.24	62.57	50.27
	Maksimum	81.82	81.28	77.54
	Ortalama	72,58%	72,10%	68,57%
	Standart sapma	4.49	5.02	6.25

Çizelge 5.4’de İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının veri setleri üzerinde ortalama çalışma süreleri verilmiştir. SSA-ÇKA algoritması en hızlı algoritma olduğu gösterilmiştir. SSA tabanlı İSSA’nın kaotik haritalar ve arama denklemleri eklenmesi ile ÇKA eğitimlerinde çalışma hızı düşmüştür. Lakin SMS-ÇKA algoritmasına kıyasla İSSA-ÇKA algoritmasının daha hızlı çalıştığını söylenebilir.

Önerilen İSSA-ÇKA ve SSA-ÇKA algoritmaları literatürdeki yedi algoritma ile karşılaştırılmıştır. SMS-ÇKA algoritmasının sonuçları (Gulcu, 2020) yayınından alınmıştır. GWO-ÇKA, ACO-ÇKA, GA-ÇKA, PBIL-ÇKA, PSO-ÇKA ve ES-ÇKA sonuçları (Mirjalili, 2015) yayınından alınmıştır. Çizelge 5.5’de algoritmaların en iyi sınıflandırma doğruluk oranları verilmiştir. Beş veri setinin beşinde İSSA-ÇKA algoritmasının en yüksek sonuç verdiğini rahatlıkla söyleyebiliriz. SSA-ÇKA

algoritması beş veri setinden dördünde başarılı sonuç vermiştir. Kalp veri setinde SSA tabanlı İSSA algoritmasının ÇKA eğitimi sonucunda İSSA-ÇKA'nın SSA-ÇKA'yı geçtiği görülmektedir. Çizelge 5.5'de dikkat çeken bir nokta ise balon veri setinde tüm algoritmaların başarılı sonuç vermesidir.

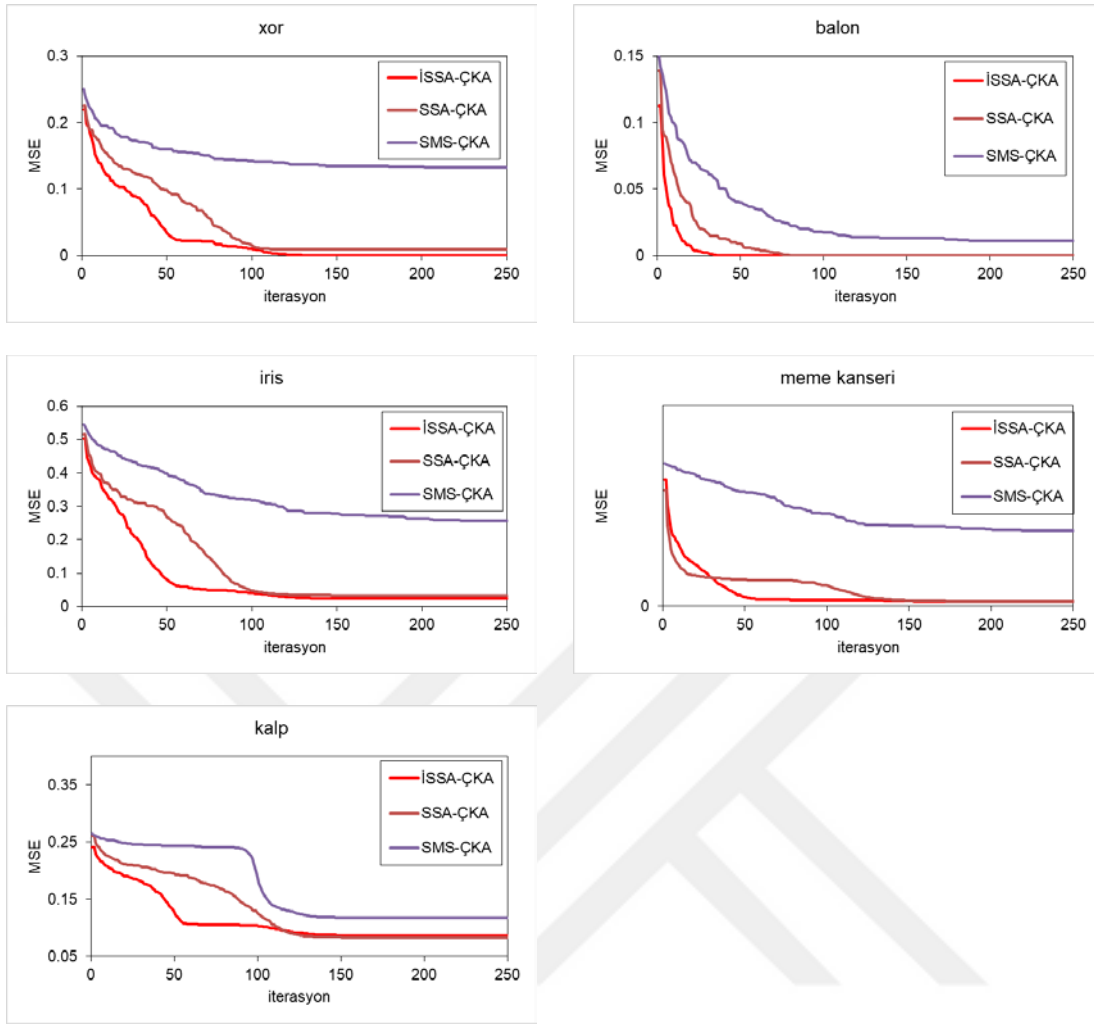
Çizelge 5.4. Algoritmaların ortalama çalışma süreleri (saniye)

Veri seri	İSSA-ÇKA	SSA-ÇKA	SMS-ÇKA
Xor	0,9	0,6	5,5
Balon	10,5	9,0	20,2
Iris	245,4	234,2	881,6
Meme Kanseri	485,1	478,6	2830,3
Kalp	424,3	400,4	892,4

Çizelge 5.5. En iyi sınıflandırma doğruluk oranı (%)

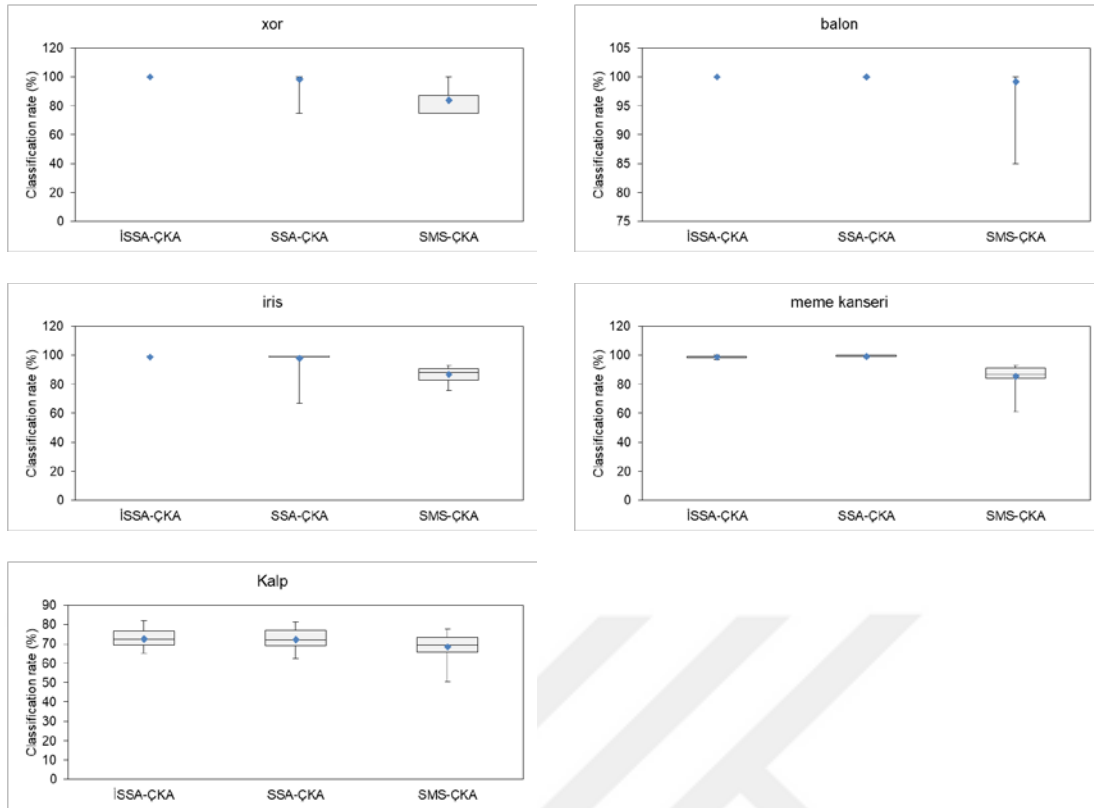
Veri Setleri	İSSA-ÇKA	SSA-ÇKA	SMS-ÇKA	GWO-ÇKA	PSO-ÇKA	GA-ÇKA	ACO-ÇKA	ES-ÇKA	PBIL-ÇKA
Xor	100,00	100,00	100,00	100,00	37,50	100,00	62,50	62,50	62,50
Balon	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
Iris	99,33	99,33	93,33	91,33	37,33	89,33	32,66	46,66	86,66
Meme Kanseri	100,00	100,00	93,00	99,00	11,00	98,00	40,00	06,00	07,00
Kalp	81,82	81,28	77,54	75,00	68,75	58,75	00,00	71,25	45,00

Şekil 5.1'de algoritmaların iterasyonlar boyunca yakınsama grafikleri verilmiştir. Grafiklerde İSSA-ÇKA algoritmasının yerel minimumlardan SSA-ÇKA ve SMS-ÇKA algoritmalarına göre daha iyi kaçındığını görebiliriz. Aynı zamanda daha az iterasyon ile global optimum noktasına diğer algoritmalara göre daha hızlı yakınsadığı görülmektedir.



Şekil 5.1. Algoritmaların yakınsama grafikleri

Şekil 5.2’de İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının, sınıflandırma doğruluğu sonuçlarının kutu grafikleri görülmektedir. Burada sıçrama oranına göre İSSA-ÇKA’nın diğer algoritmalarından daha başarılı olduğu görülmektedir.



Şekil 5.2. Algoritmaların kutu grafikleri

Çizelge 5.6'da İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının sınıflandırma oranlarına göre sıralanması (Friedman testi) gösterilmiştir. Test sonucuna göre İSSA-ÇKA algoritması 2,7 puan ile birinci sıradadır.

Çizelge 5.6. Algoritmaların sınıflandırma oranına göre ortalama sıralaması (Friedman testi)

Algoritma	Sıralama
İSSA-ÇKA	2,7
SSA-ÇKA	2,3
SMS-ÇKA	1

Xor veri seti için İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının duyarlılık, özgüllük, kesinlik ve F1-score testlerinde verdikleri sonuçlar Çizelge 5.7'de gösterilmiştir. 3-7-1 ÇKA yapısına sahip, sekiz eğitim sekiz test örneği ile kolay seviye olan xor veri setinde İSSA-ÇKA algoritması tam başarı elde etmektedir.

Çizelge 5.7. Xor veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları

Algoritma	Duyarlılık	Özgüllük	Kesinlik	F ₁ -Score
İSSA-ÇKA	100,00%	100,00%	100,00%	1,0000
SSA-ÇKA	97,50%	99,17%	99,17%	0,9821
SMS-ÇKA	81,70%	86,70%	88,90%	0,8349

Çizelge 5.8’de dört özellik sayısı, iki sınıf sayısı ve yirmi eğitim, yirmi test sayısına sahip balon veri setine karşı İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının duyarlılık, özgüllük, kesinlik ve f1-score testlerindeki sonuçları verilmiştir. İSSA-ÇKA ve SSA-ÇKA tam başarı elde etmektedir. SMS-ÇKA algoritması ise tam başarıya yaklaşmıştır.

Çizelge 5.8. Balon veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları

Algoritma	Duyarlılık	Özgüllük	Kesinlik	F ₁ -Score
İSSA-ÇKA	100,00%	100,00%	100,00%	1,0000
SSA-ÇKA	100,00%	100,00%	100,00%	1,0000
SMS-ÇKA	99,20%	99,20%	98,80%	0,9894

Iris veri setinin üç sınıf sayısı olması bu veri setini diğer veri setlerinden ayıran en büyük özellik denilebilir. Yüz elli eğitim ve yüz elli test sayısına sahip olması iris veri setinin zorluk seviyesinin orta derecede olduğunu gösterir. Çizelge 5.9’da bakıldığında İSSA-ÇKA algoritmasının diğer algoritmalara karşı iris veri setinde duyarlılık, özgüllük, kesinlik ve F1-score testlerinde başarılı sonuç verdiğini görebilmekteyiz.

Çizelge 5.9. Iris veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları

Algoritma	Duyarlılık	Özgüllük	Kesinlik	F ₁ -Score
İSSA-ÇKA	98,82%	99,41%	98,83%	0,9882
SSA-ÇKA	97,78%	98,89%	97,05%	0,9734
SMS-ÇKA	82,10%	91,10%	81,00%	0,8002

Eğitim örneği sayısı ile diğer veri setlerinden farklı olan meme kanseri veri seri; beş yüz doksan dokuz eğitim, yüz test örneği, iki yüz on vektör sayısı ve dokuz girdi, on dokuz ara, bir çıktı ÇKA yapısına sahiptir. Duyarlılık, özgüllük, kesinlik, F1-score testlerinden geçen algoritmaların sonuçları Çizelge 5.10’da görülmektedir. Çizelge 5.10

incelendiğinde İSSA-ÇKA algoritması ile SSA-ÇKA algoritmasının rekabetçi bir yapıda olduğu söylenmektedir. SMS-ÇKA algoritması ise duyarlılık ve F1-score testlerinden verdiği sonuçta yeterince başarılı olmadığı gözlemlenmektedir.

Çizelge 5.10. Meme Kanseri veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları

Algoritma	Duyarlılık	Özgüllük	Kesinlik	F ₁ -Score
İSSA-ÇKA	99,37%	98,48%	94,62%	0,9691
SSA-ÇKA	99,52%	99,16%	96,96%	0,9821
SMS-ÇKA	50,20%	95,10%	83,70%	0,5828

Son olarak İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmalarının kalp veri setinde duyarlılık, özgüllük, kesinlik ve F1-score sonuçları Çizelge 5.11’de gösterilmiştir. Çizelge 5.11 incelendiğinde İSSA-ÇKA algoritması tüm testlerde diğer algoritmalara kıyasla daha iyi sonuç verdiği görülmektedir.

Çizelge 5.11. Kalp veri seti için duyarlılık, özgüllük, kesinlik ve F1-score sonuçları

Algoritma	Duyarlılık	Özgüllük	Kesinlik	F ₁ -Score
İSSA-ÇKA	72,42%	74,44%	97,02%	0,8284
SSA-ÇKA	71,98%	73,56%	96,92%	0,8247
SMS-ÇKA	68,39%	70,67%	96,41%	0,7980

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

Yapay sinir ağlarının eğitimini arttırmak için literatürde sıklıkla ağırlık ve bias değerleri güncellenmesi için meta sezgisel algoritmalar kullanılmaktadır. Bu tez çalışmasında ilk defa önerilen ve kullanılan İSSA algoritması ÇKA'ların ağırlık ve bias değeri üzerindeki optimizasyon başarısını gösterilmektedir. Yeni İSSA algoritmasının trigonometrik fonksiyonlar, kaotik haritalar ve salplar arasında kattığı rassallık denklemleri sayesinde arama uzayının keşfi ile sömürü arasındaki denge, yerel optimum noktalarından kaçınma ve global en iyi noktaya daha hızlı yakınsaması sağlanmıştır. İSSA-ÇKA algoritmasının başarısını gözlemlemek için beş farklı veri seti kullanılmıştır. Literatürdeki meta sezgisel algoritmalarla eğitilen ÇKA algoritmaları ile kıyaslanmıştır. Kıyaslama İSSA-ÇKA, SSA-ÇKA ve SMS-ÇKA algoritmaları ile yapılmıştır. Aynı zamanda İSSA-ÇKA algoritmasının başarısını kanıtlamak için istatistiksel başarımler metriklerinden Friedman testi, duyarlılık, özgülük, kesinlik ve F1-score testleri ile test edilmiştir. Test sonucunda İSSA-ÇKA algoritması SSA-ÇKA algoritması ile rekabetçi yaklaşımlarda bulunmaktadır. Hatta sonuçlar detaylı bir şekilde ele alındığında global en iyiye hızlı yakınsama noktasında İSSA-ÇKA algoritmasının SSA-ÇKA algoritmasını geride bıraktığı görülmüştür. Sınıflandırma doğruluk oranlarına baktığımızda ise veri setlerinin beş tanesinin dördünde İSSA-ÇKA algoritmasının diğer algoritmalarla göre daha iyi sonuç verdiğini tespit edilmiştir. Son olarak İSSA algoritması ÇKA'larda bias ve ağırlık değerlerini güncelleme konusunda başarılı olmuştur. Bunun en büyük sebebi İSSA algoritmasının keşif yeteneği sayesinde global en iyi noktaya daha hızlı yakınsaması ve yerel optimumdan kaçma yeteneğidir.

6.2. Öneriler

YSA eğitimleri için literatürde birçok algoritma bulunmaktadır. Klasik algoritmaların aksine meta sezgisel algoritmalar son yıllarda daha popüler olmuştur. Meta sezgisel algoritmalar olan SSA algoritması Arama uzayının keşfi ile global en iyi değere hızlı yakınsama ve yerel optimum değerlerinden kaçma özellikleri sayesinde başarılı bir algoritmadır. SSA algoritması ayarlanabilir parametre ve kolay anlaşılabilirliği ile gelişmeye ve geliştirmeye uygun bir algoritmadır. Literatürde geliştirilmiş SSA algoritmaları bulunmaktadır ve hala gelişmeye, geliştirilmeye açık durumdadır. İlerleyen çalışmalarda başlangıç popülasyon değer güncellemeleri, parametre analizleri

ve literatürdeki algoritmalar ile hibritlenerek ÇKA ağırlık ve bias güncellemelerini optimum değere getirmeye yönelik çalışmalar yapılması önerilmektedir.



7. KAYNAKLAR

- Albayrak, G. ve Özdemir, İ., 2016, Yapı projelerinin süre-maliyet optimizasyonunda metasezgisel algoritma kullanımı, *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*3(5), 39-49.
- Alboaneen, D. A., Tianfield, H. ve Zhang, Y., 2017, Glowworm swarm optimisation for training multi-layer perceptrons, *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, 131-138.
- Alpay, U., Yapay Sinir Ağları, <https://www.psikolojik.gen.tr/yapay-sinir-aglari.html>: [17.02.2021].
- Balakrishnan, H. N., Kathpalia, A., Saha, S. ve Nagaraj, N., 2019, ChaosNet: A chaos based artificial neural network architecture for classification, *Chaos: An Interdisciplinary Journal of Nonlinear Science*29 (11), 113125.
- Cuevas, E., Echavarría, A. ve Ramírez-Ortegón, M. A., 2014, An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation, *Applied intelligence*, 40 (2), 256-272.
- Çelik, Y., YILDIZ, İ. ve Karadeniz, A. T., 2019, Son Üç Yılda Geliştirilen Metasezgisel Algoritmalar Hakkında Kısa Bir İnceleme, *Avrupa Bilim ve Teknoloji Dergisi*, 463-477.
- Erdal Tümer, A., Edebalı, S., Gülcü, Ş. J. I. J. o. C. ve Engineering, C., 2020, Modeling of removal of chromium (VI) from aqueous solutions using artificial neural network, 39 (1), 163-175.
- Erdogan, F. ve Gulcu, S., 2021, Training of the Artificial Neural Networks using Crow Search Algorithm, *International Journal of Intelligent Systems Applications in Engineering*, 9 (3), 101-108.
- Gulcu, Ş., 2020, Training of the artificial neural networks using states of matter search algorithm, *International Journal of Intelligent Systems Applications in Engineering* 8(3), 131-136.
- Gupta, S. ve Deep, K., 2020, A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons, *Applied intelligence*, 50 (4), 993-1026.
- Gülcü, Ş., 2021, An improved animal migration optimization algorithm to train the feed-forward artificial neural networks, *Arabian Journal for Science Engineering*, 1-25.
- Gülcü, Ş., 2022, Training of the feed forward artificial neural networks using dragonfly algorithm, *Applied Soft Computing*, 109023.

- Heidari, A. A., Faris, H., Aljarah, I. ve Mirjalili, S., 2019, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, *Soft Computing*, 23 (17), 7941-7958.
- Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I. ve Mafarja, M., 2020, Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks, *Nature-Inspired Optimizers*, 23-46.
- Irmak, B. ve Gülcü, Ş., 2021, Training of the feed-forward artificial neural networks using butterfly optimization algorithm, *MANAS Journal of Engineering*, 9 (2), 160-168.
- istmer, Friedman Testi ve Temel Amacı, <https://www.istmer.com/friedman-testi-nedir/>: [18.02.2022].
- Jaddi, N. S. ve Abdullah, S., 2018, Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting, *Engineering Applications of Artificial Intelligence*, 67, 246-259.
- Kabalıcı, E., 2014, Yapay Sinir Ağları, <https://ekblc.files.wordpress.com/2013/09/ysa.pdf>:
- Kiran, M. S., Hakli, H., Gunduz, M. ve Uguz, H., 2015, Artificial bee colony algorithm with variable search strategy for continuous optimization, *Information Sciences* 300, 140-157.
- Kiranyaz, S., Ince, T., Yildirim, A. ve Gabbouj, M. J. N. n., 2009, Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, 22 (10), 1448-1462.
- Kızrak, M. A. ve Bolat, B., 2018, Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma, *Bilişim Teknolojileri Dergisi*, 11 (3), 263-286.
- Madenci, E. ve Gülcü, Ş., 2020, Optimization of flexure stiffness of FGM beams via artificial neural networks by mixed FEM, *Structural Engineering Mechanics, An Int'l Journal*, 75 (5), 633-642.
- Maltarollo, V. G., Honório, K. M. ve da Silva, A. B. F., 2013, Applications of artificial neural networks in chemical problems, *Artificial neural networks-architectures applications*, 203-223.
- Mirjalili, S., Hashim, S. Z. M. ve Sardroudi, H. M., 2012, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, *Applied Mathematics Computation*, 218 (22), 11125-11137.
- Mirjalili, S., 2015, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, *Applied intelligence*, 43 (1), 150-161.

- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. ve Mirjalili, S. M., 2017, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, 114, 163-191.
- Öztemel, E., 2006, Yapay Sinir Ağları, 2, *Baskı, İstanbul, Papatya Yayıncılık*
- Öztürk, K. ve Şahin, M. E., 2018, Yapay sinir ağları ve yapay zekâ'ya genel bir bakış, *Takvim-i Vekayi*, 6 (2), 25-36.
- Patel, P., Arockia Doss, A. S., PavanKalyan, L. ve Tarwadi, P. J., 2021, Speech recognition using neural network for mobile robot navigation, *Trends in Mechanical Biomedical Design*, 665-676.
- Pirim, A. G. H., 2006, Yapay zeka, *Yaşar Üniversitesi E-Dergisi*, 1 (1), 81-93.
- Sarkar, S. D. ve KB, A. S., 2020, Face recognition using artificial neural network and feature extraction, *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, 417-422.
- Şen, Z., 2004, Yapay sinir ağları, Su Vakfı, p.
- Tekkanat, N., 2006, Altın oran'ın kaynakları ve sanat'a yansımaları, *Akdeniz Üniversitesi*.
- Wilamowski, B. M., Hunter, D. ve Malinowski, A., 2003, Solving parity-N problems with feedforward neural networks, *Proceedings of the International Joint Conference on Neural Networks, 2003.*, 2546-2551.
- Yamany, W., Fawzy, M., Tharwat, A. ve Hassanien, A. E., 2015a, Moth-flame optimization for training multi-layer perceptrons, *2015 11th International computer engineering Conference (ICENCO)*, 267-272.
- Yamany, W., Tharwat, A., Hassanin, M. F., Gaber, T., Hassanien, A. E. ve Kim, T.-H., 2015b, A new multi-layer perceptrons trainer based on ant lion optimization algorithm, *2015 Fourth international conference on information science and industrial applications (ISI)*, 40-45.
- Zhang, Z., 2018, Artificial neural network, In: *Multivariate time series analysis in climate and environmental research*, Eds: Springer, p. 1-35.

ÖZGEÇMİŞ

