



**HOSOYA YAPAY SİNİR AĞLARI İLE KİSMİ
DİFERANSİYEL DENKLEMLERİN ÇÖZÜMÜ**

Merve Zeynep KAYA

Danışman: Doç. Dr. Mesut KARABACAK

Doktora Tezi

Matematik Ana Bilim Dalı

2024

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANA BİLİM DALI

HOSOYA YAPAY SİNİR AĞLARI İLE KISMİ DİFERANSİYEL DENKLEMLERİN ÇÖZÜMÜ

(Solutions of Partial Differential Equations with Hosoya Neural Network)

DOKTORA TEZİ

Merve Zeynep KAYA

Danışman: Doç. Dr. Mesut KARABACAK
İkinci Tez Danışmanı: Prof. Dr. Ercan ÇELİK

Erzurum
Kasım, 2024

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü

TEZ KABUL VE ONAY TUTANAĞI

HOSOYA YAPAY SİNİR AĞLARI İLE KISMİ DİFERANSİYEL DENKLEMLERİN
ÇÖZÜMÜ

Doç. Dr. Mesut KARABACAK danışmanlığında, Merve Zeynep KAYA tarafından hazırlanan bu çalışma, 08/11/2024 tarihinde aşağıdaki jüri tarafından Matematik Anabilim Dalı Uygulamalı Matematik Bilim Dalı'nda doktora tezi olarak **oybirliği / oy çokluğu (.../...)** ile kabul edilmiştir.

Jüri Başkanı:	Prof. Dr. Hasan BULUT <i>Fırat Üniversitesi</i>	Aslı Islak İmzalıdır
Danışman:	Doç. Dr. Mesut KARABACAK <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Prof. Dr. Alaattin ESEN <i>İnönü Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Doç. Dr. Sıdika Şule ŞENER KILIÇ <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Doç. Dr. Bilal NİŞANCI <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Dr. Öğr. Üyesi Arzu AYKUT <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
İkinci Tez Danışmanı:	Prof. Dr. Ercan ÇELİK <i>Kırgızistan-Türkiye Manas Üniversitesi</i>	Aslı Islak İmzalıdır

Enstitü Yönetim
Kurulunun .../.../.... tarih
ve sayılı kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Aslı Islak İmzalıdır

Prof.Dr. Saltuk Buğrahan CEYHUN

Enstitü Müdürü

Bu çalışma BAP projeleri kapsamında desteklenmiştir.

Proje No: FDK-2022-11492

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Doktora Tezi olarak *Doç. Dr. Mesut KARABACAK* danışmanlığında sunulan “Hosoya Yapay Sinir Ağları ile Kısmi Diferansiyel Denklemlerin Çözümü” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	8	30
Kuramsal Temeller	11	30
Materyal ve Yöntem	1	35
Bulgular	15	20
Tartışma	9	20
Tezin Geneli	11	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Merve Zeynep KAYA	Doç. Dr. Mesut KARABACAK
8.11.2024	8.11.2024
İmza: Aslı Islak İmzalıdır	İmza: Aslı Islak İmzalıdır

* Tez ile ilgili YÖKTEZ’de yayımlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Doktora tezi olarak sunduğum bu çalışma sırasında danışmanlığımı üstlenip değerli bilgi ve katkılarını esirgemeyen Atatürk Üniversitesi Fen Fakültesi Matematik Bölümü öğretim üyesi doktora danışmanım Sayın Doç. Dr. Mesut KARABACAK ve Kırgızistan-Türkiye Manas Üniversitesi öğretim üyesi doktora eş danışmanım ve aynı zamanda yüksek lisans danışmanım Sayın Prof. Dr. Ercan ÇELİK hocalarıma en içten teşekkürlerimi sunmayı bir borç bilirim. Çalışmamda desteğini ve bana olan güvenini benden esirgemeyen sevgili eşime ve beni bu günlere getiren ve benden hiçbir zaman desteğini esirgemeyen canım aileme sonsuz teşekkürler.

Merve Zeynep KAYA

ÖZET

DOKTORA TEZİ

HOSOYA YAPAY SİNİR AĞLARI İLE KİSMİ DİFERANSİYEL DENKLEMLERİN ÇÖZÜMÜ

Merve Zeynep KAYA

Danışman: Doç. Dr. Mesut KARABACAK
İkinci Tez Danışmanı: Prof. Dr. Ercan ÇELİK

Amaç: Bu tezde kısmi diferansiyel denklemlerin bir sınıfı olan adveksiyon-reaksiyon difüzyon denklemlerini çözmek için bir metot sunulmuştur.

Yöntem: İlk olarak Hosoya yapay sinir ağı mimarisi oluşturularak mimariye uygun şekilde bir algoritma elde edilmiştir. Belirli bir algoritma dahilinde oluşturulan kodlar, adveksiyon-reaksiyon difüzyon denklem örneklerine uygulanmıştır.

Bulgular: Üç farklı test problemi ele alınmıştır ve analitik çözümlerle yaklaşık çözümlerin uyumlu olduğu görülmüştür.

Sonuç: Hosoya yapay sinir ağlarının adveksiyon-reaksiyon difüzyon denklemlerinin çözümü için etkili ve pratik bir metot olduğu görülmüştür.

Anahtar Kelimeler: Hosoya Polinomu, Adveksiyon-Reaksiyon Difüzyon Denklemi, Yapay Sinir Ağları

Kasım 2024, 62 sayfa

ABSTRACT

PhD THESIS

SOLUTIONS OF PARTIAL DIFFERENTIAL EQUATIONS WITH HOSOYA NEURAL NETWORK

Merve Zeynep KAYA

Supervisor: Assoc. Prof. Dr. Mesut KARABACAK

Co-supervisor: Prof. Dr. Ercan ÇELİK

Purpose: In this thesis, a method is described for solving of advection-reaction diffusion equation is class of partial differential equations.

Method: First, the Hosoya neural network architecture was created and an algorithm was obtained in accordance with the architecture. The codes generated within a certain algorithm were applied to the advection-reaction diffusion equation examples.

Findings: Three different test problems were considered and the approximate solutions were found to be in good agreement with the actual solutions.

Results: Hosoya artificial neural networks are found to be an effective and practical method for solving advection-reaction diffusion equations.

Keywords: Hosoya Polynomial, Advection-Reaction Diffusion Equation, Artificial Neural Networks

November 2024, 62 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ	ix
GİRİŞ.....	1
KURAMSAL TEMELLER.....	5
Yapay Sinir Ağı	5
Yapay Sinir Ağı Mimarisi.....	6
Yapay Sinir Ağı'nın Eğitilmesi	9
Denetimli öğrenme.....	9
Denetimsiz veya uyarlanabilir öğrenme.....	10
Pekiştirmeli öğrenme	10
Aktivasyon Fonksiyonu	11
Lineer aktivasyon fonksiyonu	11
Sigmoid aktivasyon fonksiyonu.....	12
Uni-polar sigmoid aktivasyon fonksiyonu	12
Bi-polar sigmoid aktivasyon fonksiyonu	12
Hiperbolik tanjant aktivasyon fonksiyonu	13
Gauss aktivasyon fonksiyonu.....	13
Gauss simetrik aktivasyon fonksiyonu	14
Radyal tabanlı aktivasyon fonksiyonu	14
Eliot aktivasyon fonksiyonu.....	14
Konik kesit aktivasyon fonksiyonu.....	15
Yapay Sinir Ağlarının Tasarlanması.....	15
Nöron sayısının seçilmesi	15
Başlangıç ağırlıklarının seçilmesi	16
Sinir Ağı Öğrenimi.....	17
Hebbian öğrenme	17
Delta öğrenme kuralı.....	17

Perceptron öğrenme kuralı	18
Geri yayımlı öğrenme	19
Sinir Ağı Türleri.....	20
İleri beslemeli sinir ağı.....	20
Radyal tabanlı sinir ağı	21
Çok katmanlı sinir ağı	21
Modüler sinir ağı.....	22
Tekrarlayan sinir ağı	22
Yapay Sinir Ağı Yüksek Dereceli Öğrenme Algoritmaları	23
Hosoya Polinomu.....	24
Kesirli Adveksiyon-Reaksiyon Difüzyon Denklemi	25
Gama Fonksiyonu	27
Beta Fonksiyonu	28
Mittag-Leffler Fonksiyonu.....	28
Kesirli Hesaplamanın Temel Denklem Tipleri	28
Riemann-liouville tipi kesirli diferansiyel denklem.....	29
Caputo tipi kesirli diferansiyel denklem	29
MATERYAL VE METOT.....	31
Hosoya Yapay Sinir Ağı Metodu.....	32
Marquardt Algoritması Adımları	34
ARAŞTIRMA BULGULARI	36
Test Problemi 1	36
Test Problemi 2	38
Test Problemi 3	41
TARTIŞMA VE SONUÇ.....	45
KAYNAKLAR.....	47
ÖZGEÇMİŞ.....	51

TABLÖLAR DİZİNİ

Tablo 1. Test Problemi 1 Maksimum Mutlak Hata Tablosu	38
Tablo 2. Test Problemi 2 Maksimum Mutlak Hata Tablosu	41
Tablo 3. Test Problemi 3 Maksimum Mutlak Hata Tablosu	43



ŞEKİLLER DİZİNİ

Şekil 1. İnsan Vücudundaki Sinir Ağı.....	3
Şekil 2. Biyolojik Nöron	5
Şekil 3. Yapay Nöronun Çalışması	6
Şekil 4. Yapay Nöronun Ağırlık Güncellemesi (Bias).....	7
Şekil 5. Çok Katmanlı Algılayıcı (Sinir Ağı).....	8
Şekil 6. Tek Katmanlı Algılayıcı (Sinir Ağı)	8
Şekil 7. Uni-Polar Sigmoid Aktivasyon Fonksiyonu Grafiği	12
Şekil 8. Bi-Polar Sigmoid Aktivasyon Fonksiyonu Grafiği.....	13
Şekil 9. Hiperbolik Tanjant Aktivasyon Fonksiyonu Grafiği	13
Şekil 10. Konik Kesit Aktivasyon Fonksiyonu Parabol Grafiği	15
Şekil 11. Ağırlık Başlatma Tekniklerinin Sınıflandırılması.....	16
Şekil 12. Perceptron sinir ağı	18
Şekil 13. Tek katmanlı ileri beslemeli sinir ağı.....	20
Şekil 14. Radyal tabanlı sinir ağı	21
Şekil 15. Çok katmanlı sinir ağı.....	22
Şekil 16. Modüler sinir ağı.....	22
Şekil 17. Tekrarlayan sinir ağı	23
Şekil 18. Hosoya yapay sinir ağı mimarisi.....	31
Şekil 19. Test Problemi 1 Analitik Çözüm Grafiği.....	37
Şekil 20. Test Problemi 1 Yaklaşık Çözüm Grafiği.....	37
Şekil 21. Test Problemi 1 Hata Grafiği	38
Şekil 22. Test Problemi 2 Analitik Çözüm Grafiği	39
Şekil 23. Test Problemi 2 Yaklaşık Çözüm Grafiği.....	40
Şekil 24. Test Problemi 2 Hata Grafiği	40
Şekil 25. Test Problemi 3 Analitik Çözüm Grafiği	42
Şekil 26. Test Problemi 3 Yaklaşık Çözüm Grafiği.....	42
Şekil 27. Test Problemi 3 Hata Grafiği	43

GİRİŞ

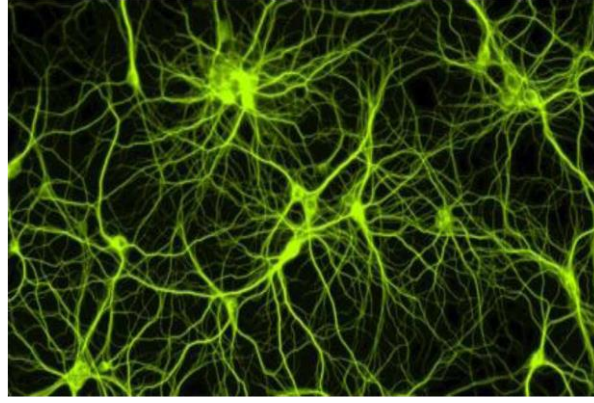
İnsan beyni birbirine bağı 10 milyardan fazla nörondan oluşan bir sistemdir. Her nöron biyokimyasal reaksiyonları almak, işlemek ve iletmek için bilgiyi kullanan bir hücredir. Beyin oldukça yüksek paralel bilgi işleme yapısına sahiptir. İnsan beynindeki bu bilgi işleme yeteneğı, yirminci yüzyılın büyük bölümünde bilgisayarların beyni modelleyebilecek şekilde programlama yapabilmesi üzerine yapılan çalışmalara konu olmuştur. Yapay sinir ağı (YSA), biyolojik sinir sistemlerinin matematiksel modellerinin genelleştirilmesi olarak geliştirilmiştir (Abraham 2005). 1943'te nörofizyolog Warren McCulloch ve matematikçi Walter Pitts tarafından yapay sinir ağlarının temelleri atılmıştır. McCulloch ve Pitts beynin temel işlem elemanlarının nasıl çalıştığını açıklamak için basit elektrik devrelerinin çeşitli mantıksal fonksiyonları nasıl hesaplayabileceğini gösterdiler. McCulloch ve Pitts, uyarıcı/engelleyici sinapsların etkisi hesaba katılarak gelen sinyal modellerinin her bir nöron üzerindeki net etkisinin matematiksel olarak hesaplanmasının mümkün olduğunu belirlemişlerdir ve yapay sinir ağlarının tasarımına dayanan yeni bir hesaplama biçiminin kapısını açmışlardır (Bishop 2015).

1958'de Rosenblatt algılayıcı(perceptron) adını icat etti ve daha sonra büyük ilgi gören bir mimari tasarladı (Rosenblatt 1958). 1960 yılında Widrow ve Hoff, delta kuralı olarak bilinen algılayıcı eğitim algoritmasının önemli bir genellemesini sundu (Widrow and Hoff 1960). Daha sonra Minsky ve Papert, birçok özelliğini kanıtladıkları ve ilgili birkaç modelin sınırlamalarına işaret ettikleri algılayıcının titiz bir analizini ortaya koydu (Minsky and Papert 1969). 1982 yılında Hopfield optimizasyon problemlerini çözmek için belirli bir doğrusal olmayan dinamik yapı uygulamıştır (Hopfield 1982). 1950 ve 1960 yıllarındaki bu çalışmalar YSA'larının teorik yönü üzerine öncü çalışmalar olmuştur. Rumelhart, Hinton ve Williams 1986 yılında Parallel Distributed Processing (PDP) araştırma grubunun monografisinde algılayıcı tipi YSA'larındaki yapay nöronların ağırlıklarının güncellenmesi için bir gradyan iniş öğrenme kuralı olan delta kuralını çok katmanlı ileri beslemeli ağlar için uygulayarak geri yayılım eğitim algoritmasını (BP) çalışmışlardır (Rumelhart et al. 1986). Bu çalışma, BP algoritmasının tanınmasını sağlamış ve geliştirmiş, beynin mekanizmaları ve yapısının incelenmesine güçlü bir ivme kazandırmış ve YSA'larının sonraki araştırma ve uygulamalarının çoğu için emsal niteliğinde olmuştur (Huang 2009).

Hesaplama tekniklerinin bir birleşimi olan esnek hesaplama bulanık mantık, makine öğrenimi, yapay sinir ağları (YSA), veri analizi ve genetik algoritmaları içerir. Esnek hesaplama analitik hesaplama göre gerçek zamanlı verilerin daha hızlı işlenmesi ve analiz edilmesi, değişen verilere hızla uyum sağlaması, öğrenilmemiş veri gibi belirsizlikleri daha etkili bir şekilde çözmesi gibi yönleri ile karmaşık problemlerin çözümünde daha etkilidir. Önemli bir esnek hesaplama teknolojisi olarak yapay sinir ağları son otuz yıl boyunca kapsamlı bir şekilde çalışılmış ve uygulanmıştır. Çok katmanlı algılayıcı ağlar için geri yayılım eğitim algoritmaları üzerine yapılan araştırmalar, radyal tabanlı fonksiyon, tekrarlayan ağ, geri besleme ağ ve denetimsiz Kohonen kendi kendini organize eden ağlar gibi diğer sinir ağlarının geliştirilmesine temel olmuştur. Geri yayılım algoritmasına sahip olan çok katmanlı algılayıcı ağı birçok bilimsel araştırma ve mühendislik alanında çalışılmıştır (Huang 2009).

Yapay Sinir Ağları (YSA), son zamanlarda ortaya çıkan ve karmaşık gerçek dünya problemlerini modellemek için birçok disiplinde geniş kabul gören hesaplama modelleme araçlarıdır. Yapay sinir ağları birbirine ağırlıklar ile bağlı olan yapay nöronlardan meydana gelen, büyük veri setlerinde aynı anda birçok verinin işlendiği bir yapı olarak tanımlanabilir (Schalkoff 1997).

Yapay Nöron, temelde biyolojik nöronun mühendislik yaklaşımıdır. Birçok girdisi ve bir çıktısı olan bir sisteme sahiptir. YSA, birbiriyle ağırlıklarla bağlantılı ve aynı zamanda çok katmanlı bir yapıda olan çok sayıda basit işlem elemanından oluşur. Nöron, bilginin elektriksel ve kimyasal sinyallerle iletildiği sinir sisteminin temel yapısı olan bir hücredir. Bir hücre gövdesi akson ve dentrit denilen ağaç gibi uzanan iki tip daldan oluşur. Hücrenin genetik özelliklerinin bulunduğu hücre gövdesi, nöronlardan gelen sinyallerin toplandığı nöronun iletim fonksiyonlarının yönetildiği kısımdır. Tüm sinyal alma ve gönderme süreci, bir nöronun dendritler aracılığıyla diğer nöronlardan sinyal alması gibi belirli bir şekildedir. Sinir Ağı, sayıları milyonlarca olan birbirine bağlı nöronlardan oluşan bir ağıdır. Bu birbirine bağlı nöronların yardımıyla insan vücudunda tüm paralel işlemler yapılır ve insan vücudu paralel işleminin en iyi örneğidir (Mishra and Srivastava 2014).



Şekil 1. İnsan Vücudundaki Sinir Ağı

Yapay sinir ağlarının amacı biyolojik sistemlerin işleyişini taklit etmek değil, karmaşık problemleri çözmek için biyolojik sinir ağlarının işlevselliği hakkında bilinenlerden faydalanmaktır. Yapay sinir ağları bilinen verilerden öğrenme, paralel hesaplama, esneklik ve adaptasyon, genelleme ve gürültüye dayanıklılık gibi biyolojik sinir ağlarından esinlenen avantajlara sahiptir. Bu özelliklerin tercih edilmesinin sebebi doğrusal olmayan veriye daha iyi uyum sağlamaları, yüksek paralellik özelliği ile hızlı veri işlemesi, genellenmiş modelin öğrenilmemiş verilere uygulanması gibi yeteneklerinden kaynaklanmaktadır. YSA tabanlı hesaplamanın temel amacı, insan beynindeki bilgi işleme ve bilgi edinme işlevi ilham alınarak YSA'larının öğrenmesini sağlayacak matematiksel algoritma modelleri geliştirmektir (Basheer and Hajmeer 2000).

Graf teorisinde yer alan birçok polinom çeşidinden biri olan Hosoya polinomu, 1988'de Haruo Hosoya tarafından geliştirilen bir sayım polinomudur (Farahani 2013). Sayım polinomları 1936'da Polya tarafından geliştirilmiştir. Hosoya polinomu mesafe dağıtımli genel bir fonksiyondur (Hosoya 1988). Hosoya polinomu yardımıyla üretilen indislerden biri olan Wiener indisi, alkanların kaynama noktası ile bağlantılı olarak kullanılmış olup Harry Wiener tarafından 1947'de bulunmuştur (Ar and Farman 2018). Organik moleküllerin kimyasal ve fiziksel özelliklerini tanımlamak için kullanılan mesafe tabanlı grafların tümü Hosoya polinomuyla çalışılmıştır. (Bonchev 1983; Diudea 1996).

Adveksiyon-reaksiyon difüzyon denklemi bimoleküler kimyasal reaksiyonlar, Brusselator sistemi, tüberküloz iletim modellemesi, Covid-19 matematiksel modellemesi ve daha birçok fiziksel konuyu modellemek için kullanılan önemli bir kısmi diferansiyel denklem sınıfıdır. Bu denklem fraksiyonel türevi içerecek şekilde genişletilerek fraksiyonel adveksiyon-reaksiyon difüzyon denklemi olarak adlandırılmıştır. Araştırmacılar, kesirli türev modeli ile dünya yüzey bilimlerinde, taşınımındaki parçacıkların kolektif davranışını, ısı ve kütle transferini içeren taşınım problemleri gibi birçok konuyu tanımlamışlardır. Bu araştırmalar da en çok kullanılan kesirli türev ise Caputo kesirli türevidir (Nasrudin et al. 2023). Kesirli diferansiyel

denklemin çözümünün bulunabilmesi için kesirli integral ve türev değerlerinin başlangıç zaman noktasında belirtilmesi gerekir. Riemann-Liouville kesirli türevinin klasik tanımı, tamsayı dereceli diferansiyel denklemlerin ihtiyaç duyduğu klasik başlangıç koşullarını kullanmak için Caputo tarafından yeniden düzenlenmiştir. M. Caputo'nun 1967 yılında elde ettiği sonuçlar, kesirli türev hesabında en önemli çalışmalar arasında yerini almıştır (Lazarević et al. 2014).

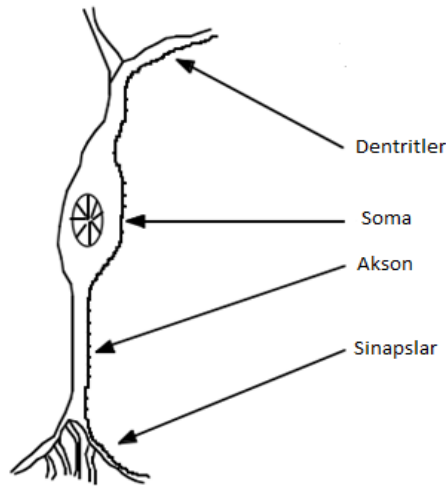
Bu tez çalışmasında fraksiyonel adveksiyon-reaksiyon difüzyon denklemi, Hosoya yapay sinir ağı ile belirli bir algoritma kullanılarak Python programlama dili ile çözülmüş olup test problemlerine uygulanmıştır. Elde edilen grafikler ve hata sonuçları tablolastırılmıştır. Bu çalışma ile geleneksel yöntemlere göre daha pratik ve hızlı çözümlerin elde edilmesi amaçlanmıştır.



KURAMSAL TEMELLER

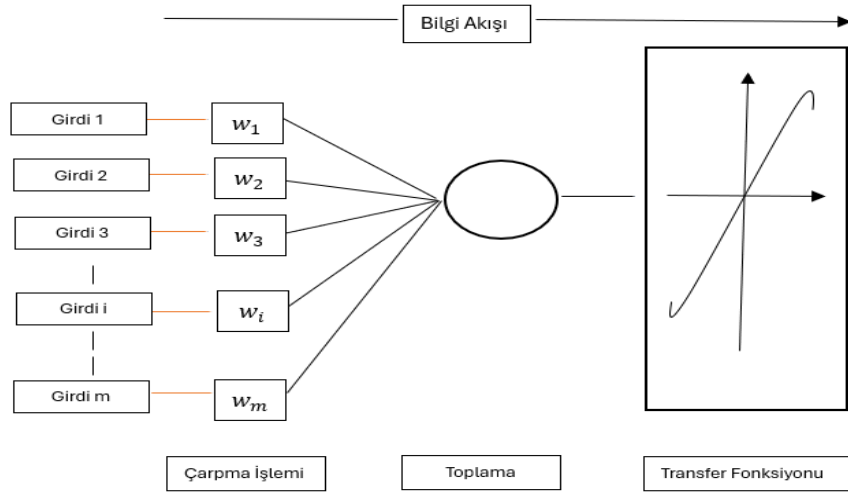
Yapay Sinir Ağı

Yapay sinir ağı beyinde bulunan sinir ağlarının yapısından ve işlevinden esinlenilerek oluşturulmuş matematiksel bir modeldir. Yapay sinir ağı ağırlıklar aracılığı ile birbirine bağlanan doğrusal olmayan yapay nöronlardan oluşur. Her yapay sinir ağının temel yapısı matematiksel bir fonksiyon olarak tanımlanan nörondur (Suzuki 2011). Temel olarak bir biyolojik nöron diğer kaynaklardan girdiler alarak bunları birleştirir ve sonuç üzerinde doğrusal olmayan bir işlem yaparak bir çıkış sonucu elde eder. Biyolojik nöronlar dentrit, soma, akson ve sinapslar olmak üzere dört temel bileşene sahiptir. Girdiler nöronların sinapsları aracılığı ile alınır. Soma, gelen sinyalleri işler ve sinapslar aracılığı ile diğer nöronlara göndererek bir çıktıya dönüştürür (Anderson and McNeil 1992).



Şekil 2. Biyolojik Nöron

Yapay sinir ağı modeli çarpma, toplama ve aktivasyon olmak üzere üç temel işlemten meydana gelir. Yapay nöron biyolojik nöronun işlevselliği taklit edilerek oluşturulmuştur. Buna göre yapay nöronun giriş kısmında her girdi değeri ayrı bir ağırlıkla çarpılarak girdiler ağırlıklandırılır. Yapay nöronun sonraki bölümünde tüm ağırlıklı girdi ve aktivasyon eşikini(bias) toplayan toplam fonksiyonu bulunur ve yapay nöronun çıkışında ağırlıklı girdiler ve bias, transfer fonksiyonundan geçerek çıkış fonksiyonu elde edilir (Krenker et al. 2011).



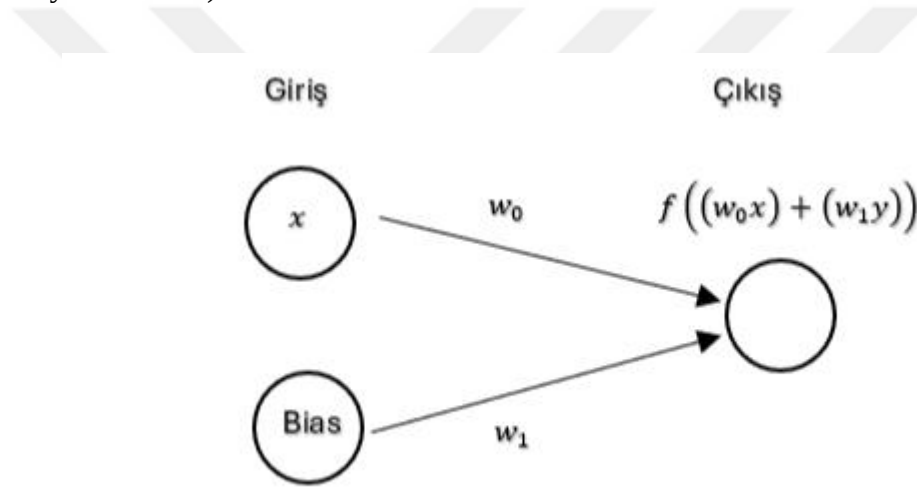
Şekil 3. Yapay Nöronun Çalışması

Bir yapay sinir ağı girdiler arasındaki bağlantı düzeni, bağlantılar üzerindeki ağırlıkları hesaplama tekniği ve aktivasyon fonksiyonu olmak üzere üç ana başlıkta tanımlanır (Thakur and Konde 2021). Sinir ağları ilişkilendirme, sınıflandırma, tahmin ve kümeleme gibi çeşitli işlemleri gerçekleştirmek için kullanılır ve uyarlanabilir öğrenme, hata toleransı, gerçek zamanlı müdahale, kendi kendini organize etme özellikleri ile nörobilim, tıp, mühendislik, kimya, fizik gibi alanlarda çalışılmakta olan makine öğrenimi temel araçlarıdır (Yang and Wang 2020).

Yapay Sinir Ağı Mimarisi

Yapay sinir ağı mimarisi üç katmandan meydana gelir. Bu katmanlar girdi katmanı, gizli katman ve çıktı katmanıdır. Gizli katmandaki nöronlar girdi katmanına bağlanırken, çıktı katmanındaki nöronlar gizli katmandaki nöronlara bağlanır (Abraham 2005). İnsan beynindeki sinapslar gibi her bir bağlantı sinyal iletime işlevine sahiptir. Girdi katmanı gelen verileri alarak her bir veriyi işlenmesi için gizli katmana gönderir. Gizli katmandaki nöronlar kendisine gelen girdileri ağırlıklar ile çarparak aktivasyon fonksiyonu ile işlenecek bir toplam verisi oluşturur. Nöronların çıktı değerleri girdi değerlerinin toplamalarının bir fonksiyonudur. Nöronların ağırlığı öğrenme boyunca sürekli optimize edilir. Ağırlıklar nöronlar arasındaki bağlantının sinyal gücünün ayarlanmasını sağlar. Nöronun topladığı sinyaller eşik değerini aştığında nöron depolarize olarak aksiyon potansiyeli oluşturup sinyal gönderir. Ağırlık ve aktivasyon fonksiyonları katmanlar arasındaki iletimin bir parçasıdır. Sinir ağının öğrenme yeteneği nöronlar arasındaki ağırlığa göre belirlenir. Ağırlıklar başlangıçta rastgele atanır. Nöronun çıktısını normalleştirmek için "aktivasyon fonksiyonu" kullanılır (Thakur and Konde 2021). Aktivasyon fonksiyonu, yapay sinir ağında verilerin ortalama ve standart sapma değerlerinin

belirlenerek sinir ağı çıktı denklemini belirleyen matematiksel fonksiyondur. Sinir ağındaki her bir nöronun aktivasyon (transfer) fonksiyonu vardır. Aktivasyon fonksiyonu, ağırlıklı toplamı hesaplayarak bir nöronun etkinleştirilip etkinleştirilmeyeceğine karar verir. Aktivasyon fonksiyonunun amacı, bir nöronun çıkışına doğrusal olmayan bir özellik katmaktır. Aktivasyon fonksiyonu, girdinin doğrusal olmayan dönüşümünü yaparak daha karmaşık görevleri öğrenmesini ve gerçekleştirmesini sağlar (Anonymous 2023). Aktivasyon fonksiyonu bir ağırlık ile çarpılmış bir girdi alır. Bias değeri, girdiye bir sabit ekleyerek aktivasyon fonksiyonunu güncellemeye olanak tanır. Sinir ağında, yapay bir nörona bazı girdiler sağlanır ve her girdiyle bir ağırlık ilişkilendirilir. Ağırlık, aktivasyon fonksiyonunun eğimini artırır. Bu, ağırlığın aktivasyon fonksiyonunun ne kadar hızlı tetikleneceğine karar verdiği anlamına gelirken, bias aktivasyon fonksiyonunun tetiklenmesini geciktirmek için kullanılır (Anonymous 2018).



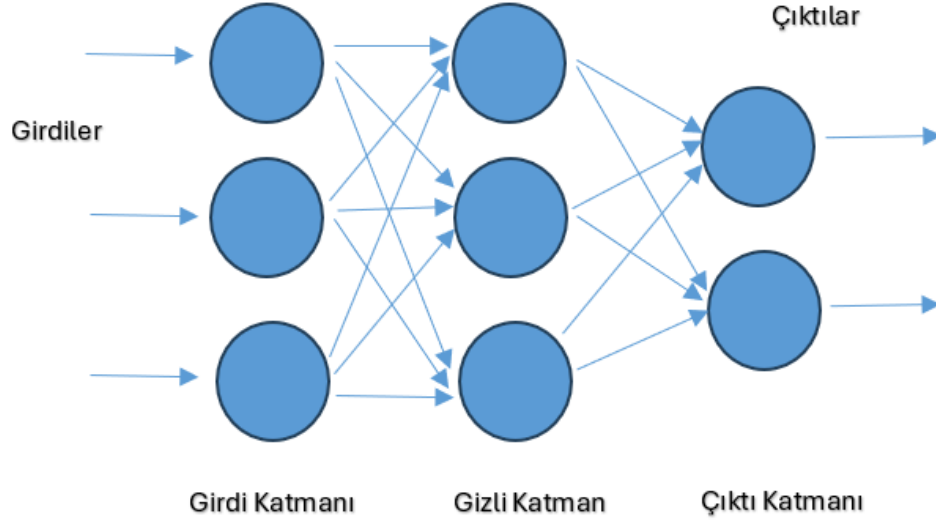
Şekil 4. Yapay Nöronun Ağırlık Güncellemesi (Bias)

Sapmanın olduğu ağırlık güncellemesi işleminde, aktivasyon fonksiyonuna giriş x çarpı bağlantı ağırlığı w_0 artı sapma çarpı sapma için bağlantı ağırlığı w_1 dir. Bu, aktivasyon fonksiyonunu sabit bir miktarda (bw_1) kaydırma etkisine sahiptir (Anonymous 2024).

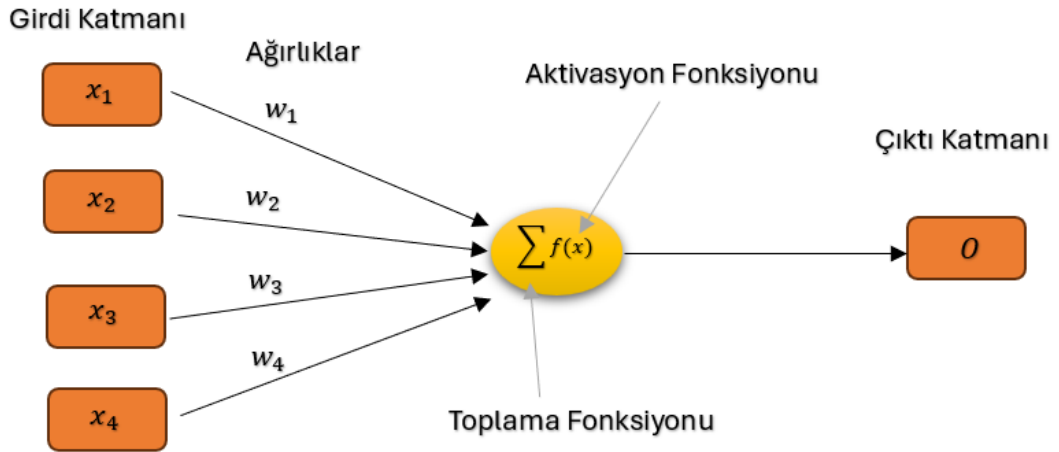
Tek katmanlı ileri beslemeli sinir ağı yapay sinir ağlarının başlangıç aşamasıdır. Algılayıcı (perceptron), yapay sinir ağlarının ilk modellerinden biridir. Bir çıkış düğümü katmanına tamamen bağlı olan tek bir giriş düğümü katmanından oluşan en basit ileri beslemeli sinir ağı türüdür. Basit bir model olmasına rağmen, algılayıcının belirli kategorizasyon sorunlarını çözmede başarılı olduğu kanıtlanmıştır (Rosenblatt 1961; Anonymous 2023).

Farklı sinir ağı türleri için tek katmanlı algılayıcı ve çok katmanlı algılayıcı olmak üzere iki farklı tipte algılayıcı bulunmaktadır. Tek katmanlı algılayıcı verilerin düz bir çizgi aracılığıyla farklı kategorilere ayrılabilirdiği görevler için etkilidir. Çok katmanlı algılayıcılar,

iki veya daha fazla katmandan oluştukları için gelişmiş işleme yeteneklerine sahiptir ve verilerdeki daha karmaşık ilişkileri ele almada kullanılırlar. Bir sinir ağının temel birimi olan algılayıcı, bilgi işlemede işbirliği yapan temel bileşenlerden oluşur. Bu bileşenler önceden bahsedilen ve sinir ağını meydana getiren girdiler, ağırlıklar, bias, aktivasyon fonksiyonu ve çıktı fonksiyonlarının tamamıdır.



Şekil 5. Çok Katmanlı Algılayıcı (Sinir Ağı)



Şekil 6. Tek Katmanlı Algılayıcı (Sinir Ağı)

Bu tez çalışmasında da kullanılan çok katmanlı sinir ağının modellenmesi Şekil 5’de gösterilmiştir. Buna göre girişlerden gelen sinyal akışının tek yönlü olarak aktarıldığı ve çıkış sinyalinin elde edildiği görülmektedir (Abraham 2005). Buna göre nöron çıkış sinyali (O),

$$O = f(\sum_{i=1}^n w_i x_i) \quad (1)$$

şeklindedir. (1) denkleminde w_i ağırlık vektörü ve f aktivasyon fonksiyonudur. Buna göre,

$$f = w^T x = w_1 x_1 + w_2 x_2 + \dots + w_n x_n \quad (2)$$

şeklindedir. Burada T bir matrisin transpozesidir ve en basit haliyle, O çıkış değeri şu şekilde hesaplanır,

$$O = f = \begin{cases} 1, & w^T x \geq \phi \\ 0, & \text{diğer} \end{cases} \quad (3)$$

burada ϕ eşik seviyesidir.

Yapay Sinir Ağının Eğitilmesi

Yapay sinir ağlarında öğrenme, belirli bir ağın nöronları arasındaki bağlantıların ağırlıklarının değiştirilmesidir. YSA' da öğrenme, denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere üç ana başlık da sınıflandırılabilir.

Denetimli öğrenme

Denetimli öğrenme, bir yapay sinir ağının parametrelerini eğitim verilerine göre ayarlayan bir makine öğrenme tekniğidir. Eğitilen yapay sinir ağının görevi, çıktı değerini gördükten sonra herhangi bir geçerli girdi değeri için parametrelerinin değerini ayarlamaktır. Eğitim verileri, veri vektörlerinde temsil edilen girdi ve istenen çıktı değerlerinin çiftlerinden oluşur (Suzuki 2011).

Denetimli eğitimde veri ağı girdileri işler ve elde ettiği çıktıları istenen çıktılarla karşılaştırır. Gerçek çıktı ile istenen çıktı vektörü arasında bir fark varsa bir hata sinyali üretilir. Bu hata sinyaline dayanarak, gerçek çıktı istenen çıktı ile eşleşene kadar ağırlıklar ayarlanır (Anonymous 2022). Bu süreç, ağırlıklar sürekli olarak ayarlandıkça tekrarlanır. Bir ağın eğitimi sırasında, bağlantı ağırlıkları sürekli olarak iyileştirilirken aynı veri kümesi birçok kez işlenir (Anderson and McNeill 1992).

Denetimli öğrenme, bir havuzdaki nesneleri verilen özellikler veya ek açıklamalar olasılığı ile sınıflandırma yeteneğine sahiptir. Denetimli öğrenmenin tipik bir örneği, öğrencilerin bir sınav yazması ve sınavın öğretmen tarafından işaretlenmesi ve öğrencilerin hangi soruları yanlış cevapladıklarının gösterilmesidir. Doğru cevaplar gösterildikten sonra, öğrencilerin bu soruları nasıl doğru cevaplayacaklarını öğrenmeleri ve anlamaları beklenir (Abiodun 2018).

Denetimsiz veya uyarlanabilir öğrenme

Denetimsiz öğrenme, verilen verilere ve minimize edilecek bir hata fonksiyonuna dayalı olarak yapay sinir ağının parametrelerini belirleyen bir makine öğrenme tekniğidir (Suzuki 2011). Denetimsiz eğitim, ağın dışarıdan yardım almadan girdileri anlamlandırması gereken eğitimidir. Denetimsiz eğitim, girdiler üzerinde bazı başlangıç karakterizasyonları gerçekleştirmek için kullanılır (Anderson and McNeill 1992).

Denetimsiz öğrenmede YSA'nın eğitimi sırasında, benzer türdeki giriş vektörleri kümeler oluşturmak için birleştirilir. Yeni bir girdi deseni uygulandığında, sinir ağı girdi deseninin ait olduğu sınıfı gösteren bir çıktı yanıtı verir. Çevreden istenen çıktının ne olması gerektiğine dair bir geri bildirim yoktur. Dolayısıyla, bu öğrenme türünde, ağın kendisi girdi verilerinden özellikleri ve girdi verilerinin çıktı ile ilişkisini keşfetmelidir (Anonymous 2022).

Denetimsiz öğrenmeye kendi başına araba kullanmayı öğrenen bir adam örnek olarak verilebilir. Bu örnekte ilk adımda adam, arabaya girip kontak anahtarıyla motoru çalıştıracak, ardından debriyayı bırakacak ve arabayı birinci vitese takarak arabanın ilerlemesi için gazı basacak ve ardından direksiyonu kontrol etmeyi başaracaktır. Sürüş adımlarını sürekli olarak geniş bir açık alanda uygulayacak ve yavaş yavaş sürüş tekniğinde ustalaşarak zamanla ustaca sürmek için yola çıkmaya başlayacaktır (Oludare 2018).

Pekiştirmeli öğrenme

Pekiştirmeli öğrenme, verilerin genellikle verilmediği, ancak çevre ile etkileşimlerle oluşturulduğu bir yapay sinir ağının parametrelerini belirleyen bir makine öğrenme tekniğidir. Maksimize edilmesi gereken getiri fonksiyonu tanımlandıktan sonra, pekiştirmeli öğrenme maksimum getiriyi üreten kuralı bulmak için çeşitli algoritmalar kullanır. İlk adımdaki kapsamlı deneme yanılma algoritması, her olası kural için getiri fonksiyonunu hesaplar ve en büyük getiriye sahip yolu seçer (Suzuki 2011). Pekiştirmeli öğrenmede, bir sistemin etkileşim yoluyla ve bu etkileşimlerin sonuçlarını gözlemleyerek çevrede edindiği önceki deneyimlerden öğrenmesini sağlar (Oludare 2018).

Standart pekiştirmeli öğrenme modelinde bir aracı çevresiyle etkileşime girer. Bu etkileşim, aracının çevreyi algılaması ve bu duyuşal girdiye dayanarak çevrede gerçekleştirilecek bir eylem seçmesi şeklinde alır. Eylem çevreyi bir şekilde değiştirir ve bu değişiklik bir skaler takviye sinyali aracılığıyla aracıya iletilir. Pekiştirmeli öğrenme, güçlü makine öğrenme sistemleri elde etmek için dinamik programlama ve denetimli öğrenme alanlarını birleştirir. Dinamik Programlama, geleneksel olarak optimizasyon ve kontrol problemlerini çözmek için kullanılan bir matematik alanıdır. Ancak geleneksel dinamik

programlama, ele alabileceği problemlerin boyutu ve karmaşıklığı açısından sınırlıdır. Pekiştirmeli öğrenme, her iki disiplinin de ayrı ayrı ele alamayacağı sorunları başarıyla çözmek için iki disiplini birleştiren bir makine zekası yaklaşımıdır (Harmon 1996). Bir pekiştirmeli öğrenme probleminin bir diğer önemli özelliği, herhangi bir durumda etkenin çevre hakkındaki mevcut bilgisini kullanmak veya o durumda daha önce hiç denenmemiş eylemleri keşfetmek arasında seçim yapmak zorunda olmasıdır (Naeem et al. 2020). Bu öğrenme yöntemi robot kontrolü, telekomünikasyon ve satranç gibi oyunlar ve diğer sıralı karar verme görevleri dahil olmak üzere çeşitli problemlere başarıyla uygulanmıştır (Suzuki 2011).

Aktivasyon Fonksiyonu

Sinir ağının gizli birimleri, ağlara doğrusal olmayan özellikler katmak için aktivasyon fonksiyonlarına ihtiyaç duyar. Aktivasyon fonksiyonları, çıkış birimleri için hedef değerlerin dağılımına uygun olacak şekilde seçilmelidir (Widrow and Lehr 1990). Sinir ağı yapısında ağ girdilerinden birimin aktivasyonu adı verilen bir çıktı değerinin elde edilmesini sağlayan en önemli birimlerden biri aktivasyon (eşik/transfer) fonksiyonudur. Nöronun çıkış genliği aktivasyon fonksiyonu tarafından sınırlanır. Aktivasyon fonksiyonu çıkış sinyalinin izin verilen genlik aralığını bazı sonlu değerlere sıkıştırır. Sınırlı bir işlem aralığındaki aktive etme işlemi ezme (squash) olarak adlandırılır. Aktivasyon fonksiyonu doğrusal ve doğrusal olmayan problemleri çözmek için yaygın olarak kullanılır. Uni-polar sigmoid, Bi-polar sigmoid, Tanh, Konik Kesit ve Radyal Taban Fonksiyonu (RBF) doğrusal olmayan problemleri çözmek için en yaygın olarak kullanılan aktivasyon fonksiyonlarıdır. Doğrusal problemleri çözmek için en yaygın kullanılan aktivasyon fonksiyonları ise özdeşlik(identity) fonksiyonu, adım (step) fonksiyonu ve ramp fonksiyonudur (Karlık ve Olgac 2011).

Lineer aktivasyon fonksiyonu

Doğrusal aktivasyon fonksiyonu, tüm gerçek sayı aralığı boyunca yalnızca pozitif sayılar üretir. Aralık değeri

$$-\infty < y < \infty, y = xt \quad (4)$$

şeklindedir. Burada x aktivasyon fonksiyonu girdi değeri, y çıktı değeri ve t ise diklik değeridir (Sibi et al. 2013).

Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu, 0 ve 1 arasında yalnızca pozitif sayılar üretir. Bu fonksiyon 0 ve 1 arasındaki eğitim verileri için kullanışlıdır ve en çok kullanılan aktivasyon fonksiyonlarından biridir. Aralık değeri

$$0 < y < 1, y = \frac{1}{1+e^{-2sx}}, d = 2sy(1 - y) \quad (5)$$

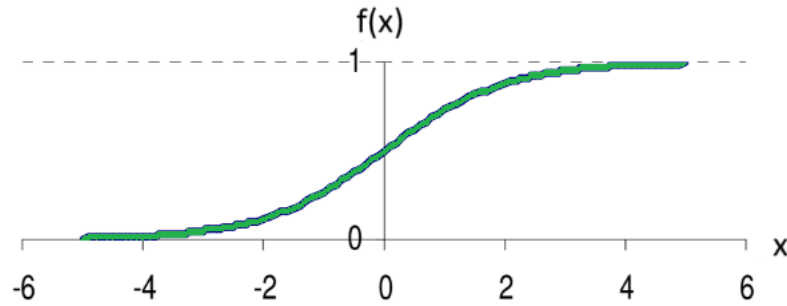
şeklindedir. Burada x aktivasyon fonksiyonu girdi değeri, y çıktı değeri, d türev değeri ve s diklik değeridir (Sibi et al. 2013).

Uni-polar sigmoid aktivasyon fonksiyonu

Bu fonksiyon eğitme için hesaplama kapasitesini en aza indirdiği için geri yayılım algoritmalarında en çok kullanılan aktivasyon fonksiyonudur. Uni-polar sigmoid aktivasyon fonksiyonu,

$$f(x) = \frac{1}{1+e^{-x}} \quad (6)$$

şeklinde tanımlanır. Uni-polar sigmoid aktivasyon fonksiyonu grafiği Şekil 7’de verilmiştir (Karlık ve Olgac 2011).



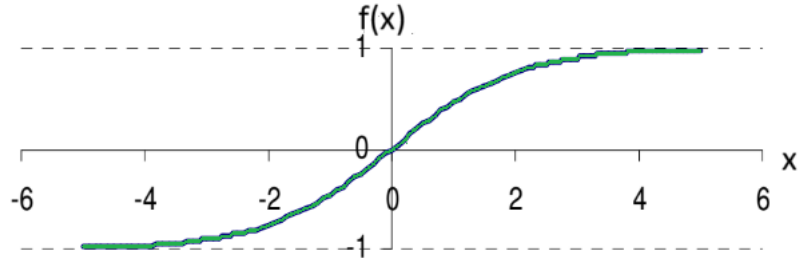
Şekil 7. Uni-Polar Sigmoid Aktivasyon Fonksiyonu Grafiği

Bi-polar sigmoid aktivasyon fonksiyonu

Bi-polar sigmoid aktivasyon fonksiyonu,

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad (7)$$

şeklinde tanımlanır. Bu fonksiyonun grafiği Şekil 8’de gösterildiği gibidir (Karlık ve Olgac 2011).



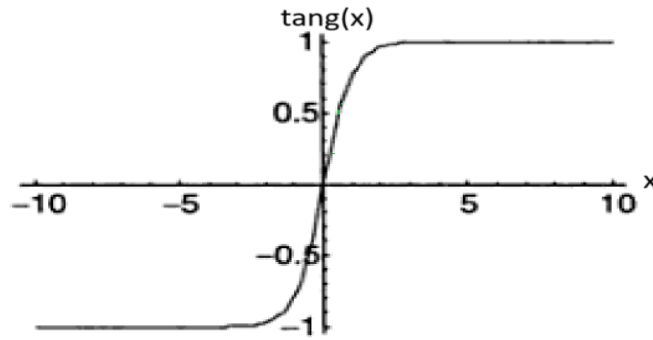
Şekil 8. Bi-Polar Sigmoid Aktivasyon Fonksiyonu Grafiği

Hiperbolik tanjant aktivasyon fonksiyonu

Bu fonksiyon hiperbolik sinüs ve kosinüs fonksiyonları arasındaki oran olarak tanımlanır. Buna göre hiperbolik tanjant aktivasyon fonksiyonu,

$$\text{tang}(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

şeklinde tanımlanır. Bu fonksiyonun grafiği Şekil 9’da gösterildiği gibidir (Karlık ve Olgac 2011).



Şekil 9. Hiperbolik Tanjant Aktivasyon Fonksiyonu Grafiği

Gauss aktivasyon fonksiyonu

Gauss aktivasyon fonksiyonu hassas aralıklar için kullanılan aktivasyon fonksiyonudur. Çıktı aralığı tanımlanan Gauss aktivasyon fonksiyonu,

$$0 < y < 1, y = e^{-x^2 t^2} = -xydt^2 \quad (9)$$

şeklinde tanımlanır. Burada x aktivasyon fonksiyonu girdi değeri, y çıktı değeri, d türev değeri ve t ise diklik değeridir (Sibi et al. 2013).

Gauss simetrik aktivasyon fonksiyonu

Gauss aktivasyon fonksiyonu ile benzer olarak aktivasyon fonksiyonu üzerinde daha hassas bir hesaplama gerektiğinde Gauss simetrik aktivasyon fonksiyonu kullanılır. Çıktı aralığı tanımlanan Gauss simetrik aktivasyon fonksiyonu,

$$-1 < y < 1, y = 2e^{-x^2t^2} - 1, d = -2xt^2(y + 1) \quad (10)$$

şeklinde tanımlanır. Burada x aktivasyon fonksiyonu girdi değeri, y çıktı değeri, d türev değeri ve t ise diklik değeridir (Sibi et al. 2013).

Radyal tabanlı aktivasyon fonksiyonu

Radyal tabanlı aktivasyon fonksiyonunun temeli Gauss eğrisine dayanır. Bu aktivasyon fonksiyonu değeri orijinden uzaklığa bağlı olan bir fonksiyon olduğundan radyal tabanlı fonksiyonların toplamı verilen fonksiyon değerine yaklaşmak için kullanılır. Radyal tabanlı fonksiyon,

$$f(x) = f(\|x\|) \quad (11)$$

şeklinde tanımlanır ya da

$$f(x) = f(\|x - c\|) \quad (12)$$

c merkez noktası uzaklığına bağlı olarak tanımlanır. Radyal tabanlı aktivasyon fonksiyonu

$$y(x) = \sum_{k=1}^M w_k f(\|x - c_k\|) \quad (13)$$

şeklindeki her biri farklı bir c_k merkeziyle ilişkilendirilmiş ve uygun bir w_k katsayısıyla ağırlıklandırılmış M radyal temel fonksiyonun toplamı olarak temsil edildiği formdaki fonksiyon yaklaşımlarını oluşturmak için kullanılır (Karlık ve Olgac 2011).

Eliot aktivasyon fonksiyonu

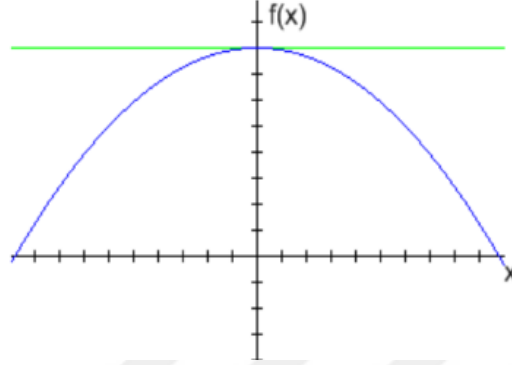
Bu aktivasyon fonksiyonu hiperbolik tanjant aktivasyon fonksiyonun daha yüksek hızlı yaklaşımıdır. Elliot aktivasyon fonksiyonu,

$$0 < y < 1, y = \frac{\frac{xt}{2}}{1+|xt|} + 0,5, d = \frac{t}{2(1+|xt|)^2} \quad (14)$$

şeklinde tanımlanır. Burada x aktivasyon fonksiyonu girdi değeri, y çıktı değeri, d türev değeri ve t ise diklik değeridir (Sibi et al. 2013).

Konik kesit aktivasyon fonksiyonu

Konik kesit aktivasyon fonksiyonu bir koninin kesit değerine göre hesaplanan fonksiyondur. Bu fonksiyon, Şekil 10'da görüldüğü gibi fonksiyonun aç değeri belirleyen bir parametre alır.



Şekil 10. Konik Kesit Aktivasyon Fonksiyonu Parabol Grafiği

Konik kesit fonksiyonu,

$$f(x) = \sum_{j=1}^{M+1} (b_i - a_i)w_i - \cos w_i(\|b - a_i\|) \quad (15)$$

şeklinde tanımlanır. Burada b_i girdi katsayısı, a_i merkez, w_i çok katmanlı algılayıcı ağırlık değeridir (Karlık ve Olgac 2011).

Yapay Sinir Ağlarının Tasarlanması

Bir sinir ağının en önemli özelliklerinden biri genelleme yapabilmesi yani eğitim setinde yer almayan problemleri de tahmin edebilmesidir. Yapay sinir ağı modellemesi çok karmaşık olduğunda sinir ağı verilerdeki rastgele dalgalanmaları yakalamaya başlayarak daha önce karşılaşılmayan verilere genelleme yaparken kötü performans gösterebilir. Bu sebeple yapay sinir ağının tasarlanması için modellemeyi oluşturan adımlar büyük önem taşımaktadır (Dongare et al. 2012).

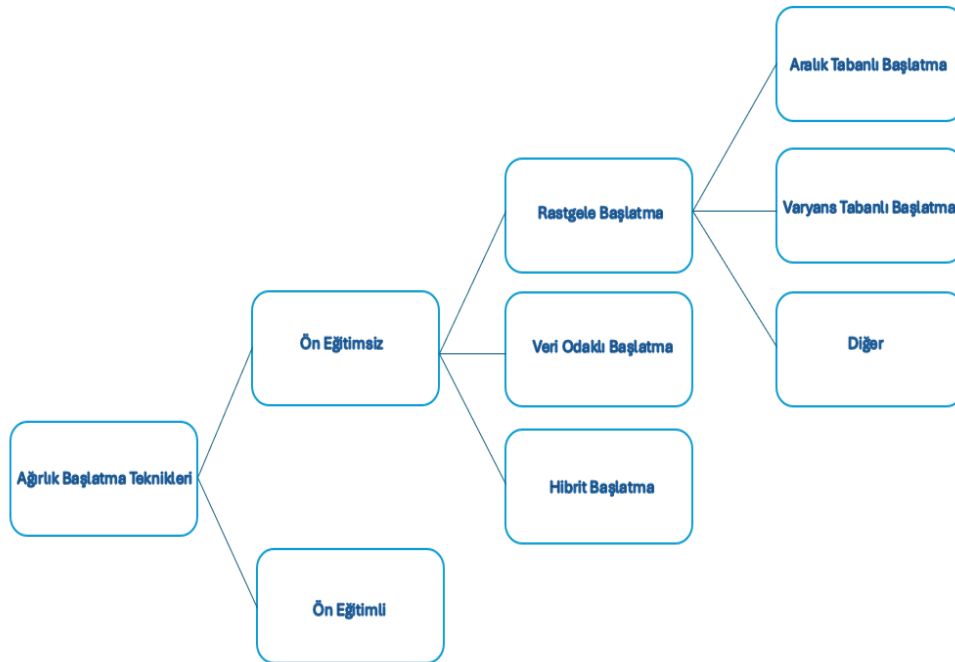
Nöron sayısının seçilmesi

Sinir ağındaki nöron sayısı, ağın verileri ne kadar iyi ayırabildiğini etkiler (Abraham 2005). Gizli nöron veri çıkışlarının bağlı olduğu hatayı etkileyebilir. Sinir ağının kararlılığı hata ile tahmin edilir. Minimum hata en iyi kararlılık anlamına gelirken, daha yüksek hata en kötü kararlılık anlamına gelir (Sheela and Deepa 2013). Gizli nöronların sayısının fazla seçilmesi sinir ağında aşırı uyuma neden olabilir, bu da hedef problemin karmaşıklığının artmasına sebep olabilir (Jinchuan and Liu 2008). Karmaşıklığın artması ise sinir ağının genelleme kabiliyetini

azaltarak tahminde önemli sapmalara yol açabilir. Bu nedenle uygun sayıda nöron sayısının seçilmesi sinir ağı modellemesinde önemli bir yere sahiptir.

Başlangıç ağırlıklarının seçilmesi

Ağırlık başlatma, herhangi bir sinir ağını eğitmeden önce kullanılan önemli bir adımdır. Bir ağı ağırlıkları başlatılır ve ardından ağ eğitilirken tekrar tekrar ayarlanır. Bu, hata minimum değere yakınsayana ve ideal bir ağırlık matrisi elde edilene kadar yapılır. Bu nedenle, ağırlık başlatma doğrudan bir ağı yakınsamasını yönlendirir ve uygun bir ağırlık başlatma şemasının seçilmesi gerekli hale gelir. Uygun bir teknik, ağı eğitimi hızlandıracak ve performansı artıracak şekilde ağırlıkları başlatır. Sinir ağlarının eğitim hızı başlangıç aralıklarına, öğrenme oranına ve ağı derinliğine bağlıdır. Başlangıç ağırlıklarının seçimi için araştırmacıların kullandığı çeşitli ağırlık başlatma teknikleri bulunmaktadır. Ağırlık başlatma teknikleri yinelemeli ve yinelemesiz eğitim mekanizmaları için iki farklı sınıfa ayrılmıştır. Yinelemeli eğitim mekanizması da kendi için de ön eğitilmiş ve ön eğitimsiz başlatma teknikleri olarak iki alt sınıfa ayrılmıştır.



Şekil 11. Ağırlık Başlatma Tekniklerinin Sınıflandırılması

Ön eğitimsiz başlatma:

1. Rastgele başlatma: Bu kategori, ağırlıkları rastgele dağılımlardan seçen başlatma tekniklerini kapsar. Üç alt sınıfa ayrılır.
 - a. Aralık tabanlı başlatma: Bu kategori, optimum türetilmiş bir aralık içinde başlangıç ağırlıklarını seçen başlatma tekniklerini araştırır.

- b. Varyans tabanlı başlatma: Bu kategori, ağırlıkların girdilerin ve çıktı değerlerinin aktivasyonlarının varyansını koruyacak şekilde ölçeklendirildiği çeşitli başlatma şemalarını kapsar.
 - c. Diğer: Aralık tabanlı ve varyans ölçekleme tabanlı dışındaki rastgele başlatma teknikleri kapsar.
2. Veri Odaklı Başlatma: Bu kategori, eğitim verilerinden türetilen ağ ağırlıklarının başlatılmasıyla ilgilenir.
 3. Hibrit: Bu kategori hem rastgele hem de veri odaklı başlatma yaklaşımlarının kombinasyonu olan başlatma tekniklerini araştırır.

Ön eğitim ile başlatma:

Bu kategori, başlangıç noktasının denetimli bir eğitim algoritması için denetimsiz bir yaklaşımla tanımlandığı yöntemleri kapsar (Narkhede et al. 2022).

Sinir Ağı Öğrenimi

Hebbian öğrenme

Hebbian öğrenme, ağdaki bağlantı gücündeki değişiklikler ile sinir ağı modeline uygulanmaktadır. Sinir ağındaki girdi biriminin çıktı birimini etkinleştirmedeki yeterliliği bağlantı gücüne bağlıdır. Hebbian öğrenmede ağırlıklar birimlerin etkinlik seviyelerine göre değişir. Bu öğrenme yönteminde ağlar kendi kendini organize eder ve ağın farklı birimleri çevresel düzenlilikleri göstermeyi öğrenir. Temel Hebbian öğrenme kuralı,

$$\Delta w_{nm} = \varepsilon a_n a_m \quad (16)$$

şeklinde gösterilir. Burada Δw_{nm} ağırlık değişimini, $a_n a_m$ n ve m ağ birimlerinin aktivasyon seviyelerini ve ε öğrenme oranını göstermektedir. Herhangi bir t zamanındaki ağırlık değeri ise

$$\Delta w_{nm} = w_{nm}(t - 1) + \Delta w_{nm} \quad (17)$$

şeklindedir. Burada, herhangi bir t zamanında n biriminden m birimine giden ağırlık, bir önceki zaman aralığındaki ağırlık değerlerinden elde edilen ağırlık değişimidir. Hebbian öğrenme kuralları, hesaplama açısından faydalı bir öğrenme biçimi sağlamanın yanı sıra, birimlerin çevredeki korelasyonları temsil etmesini de sağlar (Munakata and Pfaffly 2004).

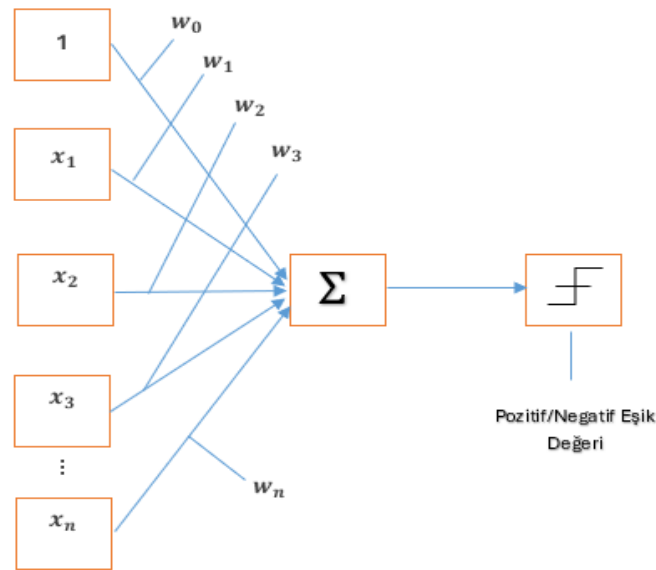
Delta öğrenme kuralı

Delta Kuralı, Hebbian Kuralının bir başka türüdür ve istenen çıktı değeri ile nöronun gerçek çıktısı arasındaki farkı azaltmak amacıyla girdi bağlantılarının güçlerini değiştirmek için

mevcut olan en yaygın yollardan biridir. Bu kural, bağlantı ağırlıklarını ağırlık ortalaması karesel hatasını en aza indirecek şekilde değiştirir. Hata, her seferinde bir katman olmak üzere önceki katmanlara geri yayılır. Ağ hatalarını geri yayma işlemi ilk katmana ulaşılan kadar devam eder. Bu kural Windrow-Hoff öğrenme kuralı ve en küçük ortalama kare öğrenme kuralı olarak da adlandırılır (Khalil Alsmadi et al. 2009).

Perceptron öğrenme kuralı

Perceptron öğrenme kuralı, doğrusal olarak ayrılabilen özel bir tür modelin sınıflandırılması için kullanılan bir sinir ağının en basit şeklidir. Ayarlanabilir sinaptik ağırlıklara ve eşik değeri olan tek bir McCulloch-Pitts nöronundan oluşur. Algılayıcıyı eğitmek için kullanılan örüntüler (vektörler) doğrusal olarak ayrılabilir sınıflardan alınırsa, algılayıcı algoritması yakınsar ve karar yüzeyini sınıflar arasında bir hiper düzlem şeklinde konumlandırır. Algoritmanın yakınsama kanıtı, perceptron yakınsama teoremi olarak bilinir (Khalil Alsmadi et al. 2009).



Şekil 12. Perceptron sinir ağı

$$g(x) = \sum_{j=0}^m w_j x_j = w^T x \quad (18)$$

$$y = \begin{cases} 1, & g \geq 0 \\ 0, & g < 0 \end{cases} \quad (19)$$

Eşitlik (18)'de gösterilen $g(x) = 0$ denklemi, algılayıcının tepki verdiği bölge ile sıfır çıktı verdiği bölge arasında bir sınır tanımlar. Bu sınır bir karar çizgisidir ve boş veya dolu örüntüleri doğrusal olarak ayrılabilirlerse birbirinden ayırt edebilmelidir (Khalil Alsmadi et al.

2009). Perceptron kuralının doğrusal olarak ayrılabilen herhangi bir eğitim örüntüsü kümesini ayırabildiği kanıtlanmıştır. Eğitim örüntüleri doğrusal olarak ayrılabilir değilse, Perceptron algoritması sonsuza kadar devam eder ve genellikle bir tane olsa bile düşük hatalı bir çözüm vermez (Widrow and Lehr 1990).

Geri yayımlı öğrenme

En yaygın kullanılan denetimli YSA modellerinden biri, geri yayılım öğrenme algoritmasını kullanan geri yayılım ağıdır. Geriye yayılım algoritması, sinir ağlarında iyi bilinen algoritmalarından biridir. 1970'lerde tek katmanlı algılayıcının basit bir XOR problemini çözmede başarısız olduğu en yakın komşu algoritmasından sonra geriye yayılım algoritması bu başarısızlığın üstesinden gelerek tanıtılmıştır. Geriye yayımlı sinir ağı esasen karmaşık bir çıktı üretmek için birlikte çalışan basit işlem elemanlarından oluşan bir ağıdır. Bu elemanlar veya düğümler farklı katmanlar halinde düzenlenmiştir. Bu katmanlar girdi katmanı, gizli katman ve çıktı katmanıdır. Geriye yayımlı bir sinir ağından elde edilen çıktı, ileri geçiş olarak bilinen bir prosedür kullanılarak hesaplanır. Girdi katmanı, belirli bir giriş vektörünün bileşenlerini orta katmandaki her bir düğüme yayar. Gizli katman düğümleri, çıktı değerlerini hesaplar ve bunlar çıktı katmanının düğümlerine girdi olur. Çıktı katmanı düğümleri, belirli bir girdi vektörü için ağ çıktısını hesaplar. İleri yönlü geçiş, ağ ağırlıklarının mevcut durumuna bağlı olarak belirli bir girdi vektörü için bir çıktı vektörü üretir. Ağ ağırlıkları rastgele değerlerle başlatıldığından, eğitimden önce uygun çıktıların elde edilmesi pek olası değildir. Ağırlıklar, çıktı hatasını ağ üzerinden geriye doğru yayarak hatayı azaltmak için ayarlanır (Shihab 2006).

Basit algılayıcı sadece doğrusal olarak ayrılabilir veya doğrusal olarak bağımsız problemleri ele alabilir. Ağın hatasının her bir ağırlığa göre kısmi türevini alarak, ağın hatasının hareket ettiği yön hakkında karara varılabilir. Aslında, bu türevin negatifi alınırsa (yani, ağırlığın değeri arttıkça hatanın oran değişimi) ve daha sonra ağırlığa eklemeye devam edilirse, hata yerel bir minimuma ulaşana kadar azalacaktır. Kısmi türevlerin alınması ve ardından ağırlıkların her birine uygulanması, çıkış katmanından başlayarak gizli katman ağırlıklarına, ardından gizli katmandan giriş katmanı ağırlıklarına doğru gerçekleştiği için bu algoritmaya geriye yayılma algoritması adı verilmiştir (Abraham 2005).

Geriye yayılım algoritma adımları şu şekildedir,

1. İlk olarak eğitim örneği girdileri girilir ve çıktılar hesaplanır.
2. Her bir p çıktı birimi için

$$\delta_p \leftarrow o_p(1 - o_p)(t_p - o_p) \quad (20)$$

hesaplanır.

3. Her bir h gizli birimi için

$$\delta_p \leftarrow o_h(1 - o_h) \sum w_{h,p} \delta_p \quad (21)$$

hesaplanır.

4. Her bir ağ ağırlığı için

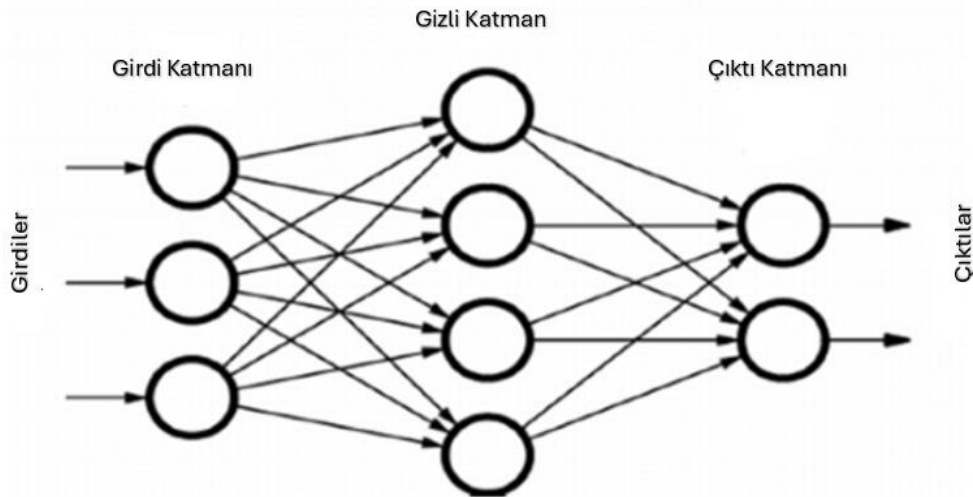
$$w_{n,m} \leftarrow w_{n,m} + \Delta w_{n,m} \quad (22)$$

ağırlık güncellemesi yapılır. Burada $\Delta w_{n,m} = \eta \delta_m x_{n,m}$ şeklindedir (Anonymous 2005).

Sinir Ağı Türleri

İleri beslemeli sinir ağı

İleri beslemeli bir sinir ağındaki veriler tek bir yönde giriş katmanından gizli katmana ve daha sonra ise çıkış katmanına doğru ilerler. Bu sinir ağında nöronlar kendilerine gelen sinyalleri toplayarak ağırlıklar ile çarpar ve veriler aktivasyon fonksiyonu ile işlenerek önden yayılan dalga şeklinde iletilir. İleri beslemeli sinir ağında veriler tek bir yönde ilerler ve geri yayılan dalga yoktur. İleri beslemeli sinir ağı birçok katmandan meydana gelebilir. Katman sayısı, nöron sayısı gibi parametreler sinir ağının performansını etkiler (Thakur and Konde 2021). Katman sayısı, her bir yapay nöronda kullanılan transfer fonksiyonu türü veya her bir yapay nöron arasındaki bağlantı sayısı konusunda herhangi bir sınırlama yoktur. En basit ileri beslemeli yapay sinir ağı, yalnızca doğrusal ayrılabilir problemleri öğrenebilen tek bir algılayıcıdır (Suzuki 2011).



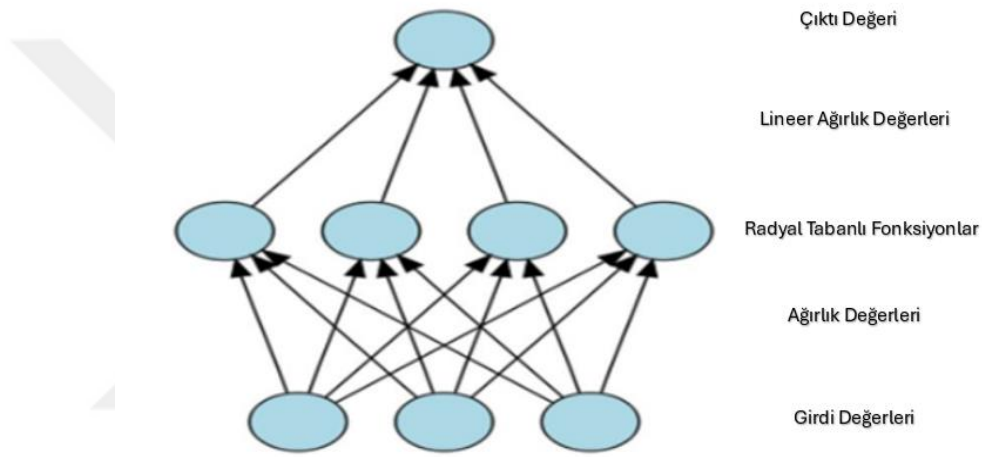
Şekil 13. Tek katmanlı ileri beslemeli sinir ağı

Makine öğrenmesinde yer alan sınıflandırma ve regresyon konularından görüntü tanıma ve bilgisayarla görme gibi birçok konuda ileri beslemeli sinir ağları kullanılmaktadır. Temel bir

ileri beslemeli sinir ağı sınıflandırmayı kolaylaştırarak veri setindeki hatalı veya bozuk verileri kolayca işleyip temizleyebilir (Thakur and Konde 2021).

Radyal tabanlı sinir ağı

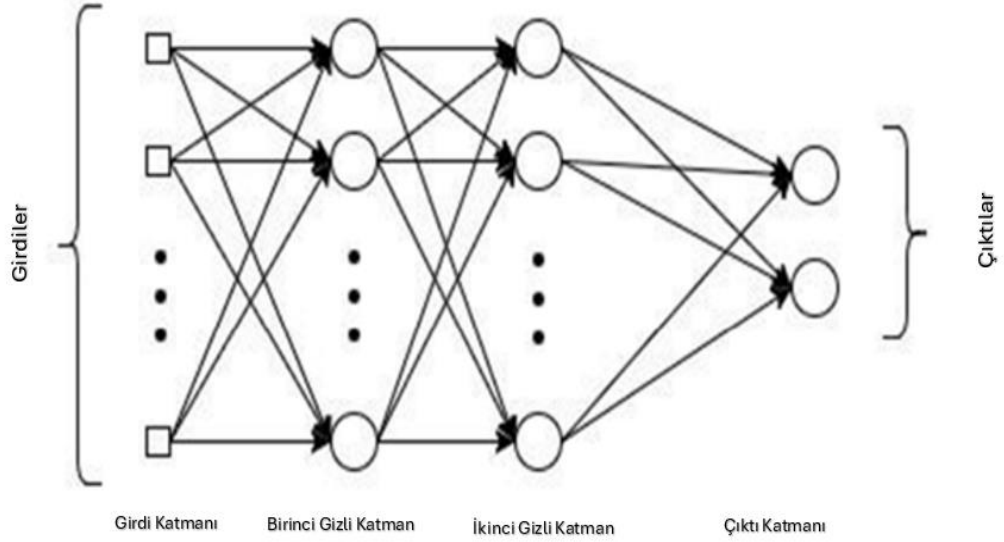
Radyal tabanlı sinir ağında nöronların merkeze olan uzaklığı hesaplanır. Bu sinir ağları işlevselliik açısından temel olarak iki katmandan oluşmaktadır. Sinir ağındaki parametreler gizli katmandaki radyal temel fonksiyonu ile birleştirilir. Bu parametrelerin çıktı değeri daha sonra bir sonraki zaman aralığında aynı sonucu hesaplamak için kullanılır. Radyal tabanlı sinir ağı, elektriğin en kısa sürede yüklenmesinin gerekli olduğu güç restorasyon sistemlerinde kullanılır (Thakur and Konde 2021).



Şekil 14. Radyal tabanlı sinir ağı

Çok katmanlı sinir ağı

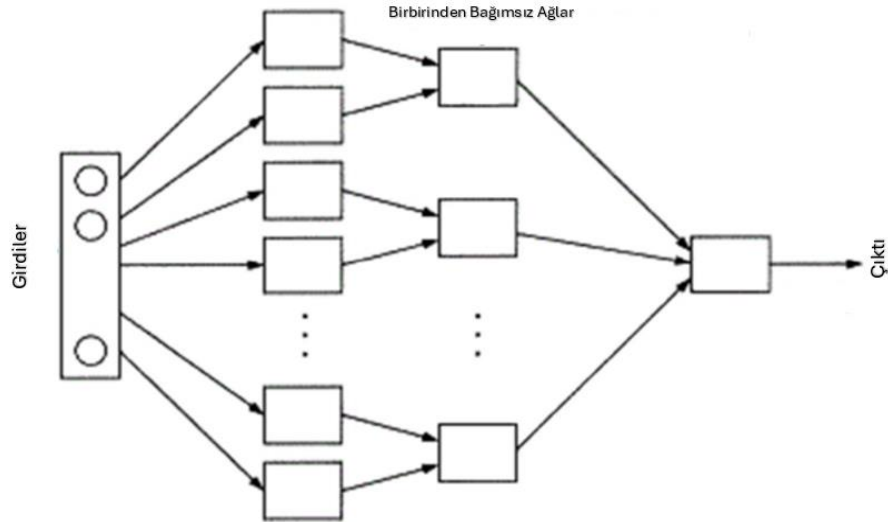
Çok katmanlı bir algılayıcı üç veya daha fazla katmandan oluşur. Bu sinir ağı görüntü verisi, ses verisi gibi yapısal verilerin sınıflandırılmasında kullanılır. Tam bağlantılı bir yapay sinir ağı biçimidir. Bunun nedeni, bir katmandaki her düğümün bir sonraki katmandaki her düğüme bağlı olmasıdır. Doğrusal olmayan bir aktivasyon fonksiyonu çok katmanlı bir algılayıcı tarafından kullanılır (çoğunlukla hiperbolik tanjant veya lojistik fonksiyon) (Thakur and Konde 2021).



Şekil 15. Çok katmanlı sinir ağı

Modüler sinir ağı

Modüler bir sinir ağı, birçok sinir ağının bir araya geldiği ve her bir sinir ağının farklı bir problemi çözmek üzere modellendiği bir sistemdir. Alt sınıflar hesaplama işlemi sırasında birbirleriyle iletişime girmeyerek ayrı bir şekilde çalışırlar. Sonuç olarak, büyük ve karmaşık bir hesaplama süreci ayrı bileşenlere bölünerek çok daha hızlı bir şekilde tamamlanabilir. Ağlar birbirleriyle etkileşim halinde olmadıkları ve hatta birbirlerine bağlı olmadıkları için hesaplama hızı artar (Thakur and Konde 2021).

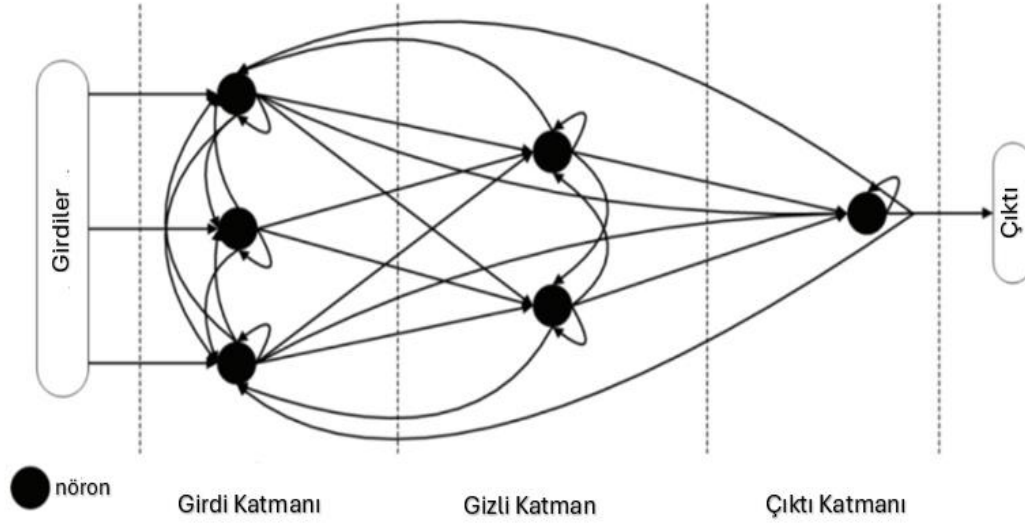


Şekil 16. Modüler sinir ağı

Tekrarlayan sinir ağı

Tekrarlayan Sinir Ağı, katmanların çıktı değerlerinin saklanarak giriş katmanına doğru geriye dalga gönderilerek beslendiği sinir ağıdır. Geriye besleme işleminde katmanın çıktı

değerinin tahmini amaçlanır. Giriş katmanı değerleri diğer sinir ağlarındaki gibi hesaplanır. Bu sinir ağında her bir nöron önceki adımlarla sonraki adımlar arasındaki çoğu veriyi belleğinde tutar. Yani veriler işlenirken nöronlar bellek hücreleri gibi davranırlar. Çıktı değeri doğru tahmin edilene kadar geriye yayılım yapılarak sinir ağı kendi kendine öğrenir (Thakur and Konde 2021).



Şekil 17. Tekrarlayan sinir ağı

Tekrarlayan sinir ağı geri döngülerle ilgili herhangi bir sınırlama olmaksızın ileri beslemeli sinir ağına benzer. Bu durumlarda bilgi artık sadece bir yönde iletilmez, aynı zamanda geriye doğru da iletilir. Bu, ağın dinamik zamansal davranış sergilemesini sağlayan bir iç durum yaratır. Tekrarlayan yapay sinir ağları, herhangi bir girdi dizisini işlemek için dahili belleklerini kullanabilir. Tekrarlayan yapay sinir ağının en temel topolojisi, her temel yapı taşının (yapay nöron) diğer tüm temel yapı taşlarına her yönde doğrudan bağlı olduğu tam tekrarlayan yapay ağıdır. Hopfield, Elman, Jordan, çift yönlü ve diğer ağlar gibi diğer tekrarlayan yapay sinir ağları, tekrarlayan yapay sinir ağlarının sadece özel durumlarıdır (Suzuki 2011).

Yapay Sinir Ağı Yüksek Dereceli Öğrenme Algoritmaları

Geriye yayılım algoritması, ağırlıkların rastgele başlatılması nedeniyle genellikle yerel bir minimumda takılır. Bazı başlangıç ağırlık ayarları için, geri yayılım algoritması ağırlık uzayının mutlak minimumuna ulaşamayabilirken, diğer başlangıçlar için aynı ağ optimum minimuma ulaşabilir. Hata yüzeyi analizinin ve eğitim algoritmalarının performansının uzun zamandır bilinen bir sıkıntısı, çoklu minimumlar da dahil olmak üzere çoklu durağan noktaların varlığıdır. Eğitim algoritmalarıyla ilgili ampirik deneyimler, farklı ağırlık başlangıçlarının

farklı sonuç ağları verdiğini göstermektedir. Bu nedenle, çoklu minimumlar sadece var olmakla kalmaz, aynı zamanda çok sayıda da olabilir (Abraham 2005).

Uygulamada, ağırlıkları optimize etmek için kullanılan dört tür optimizasyon algoritması vardır. İlk üç yöntem olan gradyan inişi, eşlenik gradyanlar ve quasi-Newton, ikinci dereceden bir hata fonksiyonunun minimizasyonu bağlamında anlaşılabilir genel optimizasyon yöntemleridir. Hata yüzeyi kesinlikle ikinci dereceden olmasa da türevlenebilir düğüm fonksiyonları için, yerel bir minimumun yeterince küçük bir komşuluğunda öyle olacaktır ve böyle bir analiz, eğitim algoritmasının birkaç iterasyon boyunca davranışı hakkında bilgi sağlar (Abraham 2005).

Levenberg ve Marquardt'ın dördüncü yöntemi, en küçük kareler probleminin çözümünde ortaya çıkan bir hata fonksiyonunun minimizasyonunun elde edildiği ikinci dereceden bir yöntemdir. Levenberg-Marquardt (LM) sinir ağı optimize edicili bir yapay sinir ağıdır. Bir optimize edici olarak Levenberg-Marquardt algoritması, hızlı yakınsaması ve küçük bellek kullanımı nedeniyle LM sinir ağlarında geniş mühendislik uygulamalarına sahiptir. Bu sinir ağı ile yapılan son çalışmaların çoğunda LM algoritması diğer algoritmalarla birleştirilerek, LM algoritmasının iyi yerel minimum değerlerini elde etmesi çalışılmıştır. Yapılan çalışmalar arasında, Levenberg-Marquardt algoritmasının Bayesian tahmini model değişkenlerini kalibre etmek için kullanılması, dalgacık yöntemi kullanılarak verileri analiz etmek için LM sinir ağları yönteminin kullanılması, genetik algoritma kullanılarak LM sinir ağının başlangıç ağırlıklarının optimize edilmesi gibi çeşitli bilimsel araştırmalar yer almaktadır (Yan 2021).

Hosoya Polinomu

Temel bir graf tanımı olan $G = (V, E)$ kenarlar olarak adlandırılan G' nin farklı köşe çiftlerinden oluşan bir $E(G)$ kümesi ve köşeler olarak adlandırılan nesnelerden oluşan bir $V(G)$ kümesinden meydana gelen bir küme setidir. E' deki her bir $x = (u, v)$ nokta çifti G' nin bir kenarıdır. Eğer u ve v noktaları bir kenar ile birleştirilirse, u ve v noktaları komşu noktalardır. $v_1, v_2, \dots, v_n$ 'e kadar olan noktalar G' nin köşeleri olmak üzere, $v_1, v_2, \dots, v_n$ 'e kadar olan n köşeli P_n grafi ve $i = 1, 2, \dots, n - 1$ olmak üzere v_i, v_{i+1} 'e komşudur. G grafının her nokta çifti bir yolla birleştirilirse buna bağlantılı graf denir. G grafindaki v_i ve v_j köşeleri arasındaki uzaklık onlarla birleşen en kısa yolun uzaklığına eşittir ve bu uzaklık $d(u_i, v_j)$ şeklinde gösterilir (Ramane et al. 2017). Kimyasal graflarda, grafın köşeleri molekülün atomlarına karşılık gelir ve kenarlar kimyasal bağları temsil eder (Sadeghieh et al. 2017). 1988 yılında H. Hosoya $H = (G, \phi)$ şeklinde gösterilen yeni bir graf polinomunu geliştirmiştir. G , n köşeli ve

m kenarlı bir graf olmak üzere bağlantılı graf G için $H(G, \phi)$ olarak gösterilen Hosoya polinomu,

$$H(G) \equiv H(G, \phi) = \sum_{l \geq 0} d(G, l) \phi^l \quad (23)$$

şeklinde tanımlanır (Geçmen et al. 2021). Burada $d(G, l)$, l uzaklığında olan köşe çiftleridir (Gutman et al. 2001). Bağlı bir G grafının Wiener indisi $W(G)$, G 'nin tüm sıralanmamış köşe çiftleri arasındaki uzaklıkların toplamı olarak tanımlanır. Bu indis 1947 yılında Harold Wiener tarafından alkanların kaynama noktalarının yaklaştırılması için ortaya atılmıştır (Ramane et al. 2017). Buna göre Hosoya polinomu ile Wiener polinomu arasındaki denklem bağlantısı,

$$W(G) = H'(G) \quad (24)$$

şeklindedir. P_n grafının Hosoya polinomu,

$$H(P_n, \phi) = n + (n-1)\phi + (n-2)\phi^2 + \dots + [n - (n-1)]\phi^{n-1} \quad (25)$$

şeklinde tanımlanır. $H_1, \dots, H_n$ 'e kadar olan Hosoya polinomu değerleri ise

$$H(P_1, \phi) = 1$$

$$H(P_2, \phi) = \phi + 2$$

$$H(P_3, \phi) = \phi^2 + 2\phi + 3$$

$$H(P_4, \phi) = \phi^3 + 2\phi^2 + 3\phi + 4$$

⋮

$$H(P_n, \phi) = n + (n-1)\phi + (n-2)\phi^2 + \dots + [n - (n-1)]\phi^{n-1}$$

şeklinde hesaplanır. Hosoya polinomu, kompozit graflar, benzenoid graflar, tori, zig-zag açık uçlu nanotüpler, Fibonacci ve Lucas küpleri gibi alanlarda çalışılmıştır (Ramane et al. 2017; Geçmen and Çelik 2021).

Kesirli Adveksiyon-Reaksiyon Difüzyon Denklemi

Adveksiyon-reaksiyon difüzyon denklemi ikinci dereceden homojen olmayan bir kısmi diferansiyel denklem sınıfıdır. Adveksiyon-reaksiyon difüzyon denklemi mühendislik ve çeşitli bilim alanlarındaki birçok fiziksel problemin çözümünde kullanılmaktadır. Bu denklem reaksiyonda üretilebildikleri veya tüketilebildikleri için korunmaları gerekmeyebilen

büyükliklerin (örneğin kütle ve enerji) taşınmasını ifade etmeye yarar. Mühendislik uygulamalarında, adveksiyon-reaksiyon difüzyon denklemi ısı transferini ve kütle ve kimyasalların gözenekli veya gözeneksiz bir ortama taşınmasını ifade eder (Gharib et al. 2017). Adveksiyon-reaksiyon difüzyon denklemi, mezoskopik problemleri simüle etmek için transport dissipatif parçacık dinamiği modeli, bimoleküler kimyasal reaksiyonlar, Brusselator sistemi, kızamıkçık salgını, tüberküloz iletim modellemesi, COVID-19 matematiksel modellemesi ve daha birçok fiziksel süreci modellemek için kullanılan önemli bir kısmi diferansiyel denklem sınıfıdır (Nasrudin et al. 2023).

Einstein'ın Brownian hareket teorisi, rastgele hareket eden bir parçacığın ortalama kare yer değiştirmesinin zamanla orantılı olduğunu ortaya koymaktadır. Ancak kesirli hesaplamaların ilerlemesinden sonra, kesirli zaman türevine sahip anormal bir difüzyon denklemi için ortalama kare yer değiştirmenin zamanla yavaş büyüdüğü görülmüştür. Basit kesirli mertebeden difüzyon denklemi

$$\frac{\partial^\beta u}{\partial t^\beta} = \frac{\partial^2 u}{\partial x^2} \quad (26)$$

için ortalama kare yer değiştirme

$$X^2(t) \approx t^\beta, 0 < \beta < 1 \quad (27)$$

şeklindedir. Burada $0 < \beta < 1$ anormal difüzyon üstelidir.

Adveksiyon, difüzyon ve reaksiyon denklemlerinin kombinasyonu, fiziksel bir sisteme aktarılan fiziksel olayları, maddeyi veya diğer fiziksel büyüklükleri tanımlar (El Arabi et al. 2022). Adveksiyon-reaksiyon difüzyon denkleminde fiziksel olarak iki eğilim vardır. Bunlar üstel eğilim ve yayılma eğilimidir. Üstel aralıkta, karakteristik polinomun kökleri gerçektir ve bu nedenle, çözümler üstel bir büyüme ile tanımlanır. Yayılma aralığında ise karakteristik polinomun kökleri karmaşık eşleniklerdir ve bu nedenle çözümler üstel bir fonksiyon tarafından değiştirilen sinüzoidal fonksiyonlardır (Hauke and García-Olivares 2001). Adveksiyon-reaksiyon difüzyon denkleminin diferansiyel formu,

$$\frac{\partial K}{\partial t} - V \frac{\partial K}{\partial y} + D \frac{\partial^2 K}{\partial y^2} - \lambda K = 0 \quad (28)$$

şeklindedir. Burada K değeri kütle transfer türlerinin konsantrasyonu, D kütle için difüzyon sabiti, V taşıyıcı akışkanın hız bileşeni ve λ ise ayrıştırma oranıdır. Genel olarak, bu denklemler çok basitleştirilmiş durumlar dışında analitik çözüm kabul etmez. Bu nedenle sayısal çözüm yöntemlerine başvurmak gerekir. Adveksiyon-reaksiyon difüzyon denkleminin sayısal çözümü için en sık kullanılan yöntemler sonlu farklar ve operatör bölme yöntemidir. Bu yöntemler başta

kimya, çevre bilimleri, hidroloji, jeoloji, petrol mühendisliği ve biyoloji olmak üzere çeşitli alanlarda çalışılmaktadır (El Arabi et al. 2022).

Kesirli diferansiyel denklemler son yirmi yılda sadece mühendislik ve bilim alanında değil ekonomi ve finans alanlarında da çalışılmıştır. Kesirli diferansiyel denklemler, geniş zaman veya uzunluk ölçeği ile karakterize edilen çok ölçekli problemlerin modellenmesinde de güçlü bir araçtır. Kesirli dereceden diferansiyel operatörün özelliği, gelecekteki durumun yalnızca mevcut duruma değil, aynı zamanda önceki durumlarının tüm geçmişine bağlı olduğu gerçeğini dikkate alan yerel olmayan özelliğidir. Günümüzde, kesirli mertebe sistemi, modelde daha fazla esnekliğe izin verdiği için dinamik sistemlerin incelenmesinde oldukça yaygınlık kazanmıştır (Das et al. 2018).

Gama Fonksiyonu

Gama fonksiyonu tanımı gereği kesirli kalkülüs ile doğrudan bağlantılıdır. Gama fonksiyonunun en temel tanımı, tüm reel sayılar için faktöriyelin genelleştirilmesidir. Gama fonksiyonunun tanımı,

$$\Gamma(y) = \int_0^{\infty} e^{-x} x^{y-1} dx, y \in \mathbb{R} \quad (29)$$

şeklindedir. Bu fonksiyon, (30)'da gösterildiği gibi herhangi bir büyüklüğün değerinin, integralin biçiminin bir sonucu olarak, o büyüklüğün $y - 1$ katının eksi bir gamasına eşit olması bakımından benzersizdir.

$$\Gamma(y + 1) = y\Gamma(y), y \in \mathbb{N}_+ \quad (30)$$

$$\Gamma(y) = (y - 1)!$$

Bu bağıntının y 'nin tam sayı değerleri için sonucu faktöriyel tanımıdır. Gama fonksiyonu kullanılarak kesirli integralin alternatif biçimlerini göstermek için yararlı olacak $\phi(t)$ fonksiyonu da tanımlanır. Bu fonksiyon,

$$\phi_{\beta}(t) := \frac{t^{\beta-1}}{\Gamma(\beta)} \quad (31)$$

eşitliği şeklinde tanımlanır (Loverro 2004). Gama fonksiyonu faktöriyelin $\mathbb{N}$ tamsayılar kümesinden $\mathbb{C}$ kompleks sayılar kümesine olan bir uzantısıdır. Leonard Euler (1707-1783) tarafından tanımlanmış ve çalışılmıştır. Olasılık teorisi, matematiksel fizik, sayı teorisi ve özel fonksiyonlar gibi birçok bilim alanı için büyük öneme sahiptir (Batir 2008).

Beta Fonksiyonu

Kesirli hesapta önemli bir yeri olan Beta fonksiyonu Euler integrali olarak da bilinir. Beta fonksiyonunun çözümü birden fazla Gama fonksiyonu kullanılarak tanımlanır. Bu fonksiyon aynı zamanda Mittag-Leffler fonksiyonu ve t^α formundaki fonksiyonların karakteristik çözümüne benzer bir formda çözüm sağlar. Beta fonksiyonun gama fonksiyonu cinsinden çözümü,

$$B(r, z) = \int_0^1 (1-v)^{r-1} v^{z-1} dv = \frac{\Gamma(r)\Gamma(z)}{\Gamma(r+z)} = B(z, r), r, z \in \mathbb{R}_+ \quad (32)$$

şeklindedir (Loverro 2004). Beta fonksiyonu olasılık ve istatistik alanındaki en önemli fonksiyonlardan biridir. Beta fonksiyonu sonlu aralıklara dağılmış çok çeşitli teorik ve ampirik yoğunluk fonksiyonlarına eğri uydurmanın yanı sıra Bayesian analizindeki uygulamalarda da çalışılmaktadır (Dutka 1981).

Mittag-Leffler Fonksiyonu

Bu fonksiyon Mittag-Leffler ve Wiman tarafından bazı ıraksak serilerin toplamını çözümlmek için geliştirilmiştir (Haubold et al. 2011). $\mathbb{C}$ kompleks sayılar olmak üzere Mittag Leffler fonksiyonu,

$$E_\alpha(y) = \sum_{m=0}^{\infty} \frac{1}{\Gamma(\alpha m + \beta)} y^m, (y, \alpha \in \mathbb{C}, Re(\alpha) > 0) \quad (33)$$

$$E_{\alpha, \beta}(y) = \sum_{m=0}^{\infty} \frac{1}{\Gamma(\alpha m + \beta)} y^m, (y, \alpha, \beta \in \mathbb{C}, Re(\alpha) > 0, Re(\beta) > 0) \quad (34)$$

şeklinde tanımlanır (Salim 2009). Mittag-Leffler fonksiyonu, kesirli mertebeden integral denklemlerin veya kesirli mertebeden diferansiyel denklemlerin çözümünde ve özellikle kinetik denklemin kesirli genellemesi, süper difüzyon taşıma ve karmaşık sistemlerin incelenmesinde çalışılmıştır (Haubold et al. 2011).

Kesirli Hesaplamanın Temel Denklem Tipleri

Kesirli hesaplamanın en temel ve önemli türleri olan Riemann-Liouville ve Caputo'nun tanımları şu şekildedir (Fernandez and Baleanu 2021).

Tanım 1: Bir $g \in L^1[c, d]$ fonksiyonunun Riemann-Liouville kesirli integrali, integrasyon sabiti c olmak üzere, $x \in [c, d]$ ve integrasyon mertebesi $\beta \in \mathbb{C}$ ve $Re(\beta) > 0$ için,

$${}^{RL}_c I^\beta g(x) = \frac{1}{\Gamma(\beta)} \int_c^x (x-y)^{\beta-1} g(y) dy \quad (35)$$

şeklinde tanımlanır.

Tanım 2: İntegrasyon sabiti c olan bir $g \in C^n[c, d]$ fonksiyonunun Riemann-Liouville kesirli türevi $x \in [c, d]$ ve türev mertebesi $\beta \in \mathbb{C}$ ve $Re(\beta) \geq 0$ ve $n - 1 \leq Re(\beta) < n$ için,

$${}^{RL}_c D^\beta g(x) = \frac{d^n}{dx^n} ({}^{RL}_c I^{n-\beta} g(x)) \quad (36)$$

şeklinde tanımlanır.

Tanım 3: Bir $g \in C^n[c, d]$ fonksiyonunun Caputo kesirli türevi, integrasyon sabiti c olmak üzere, $x \in [c, d]$ ve türev mertebesi $\beta \in \mathbb{C}$ ve $Re(\beta) \geq 0$ ve $n - 1 \leq Re(\beta) < n$ için,

$${}_c D^\beta g(x) = {}^{RL}_c I^{n-\beta} \left(\frac{d^n}{dx^n} g(x) \right) \quad (37)$$

şeklinde tanımlanır.

Riemann-liouville tipi kesirli diferansiyel denklem

Kesirli kalkülüsün modern formülasyonu Riemann tarafından Liouville'in yaklaşımı temel alınarak oluşturulmuştur. Riemann-Liouville yaklaşımı son zamanlarda kesirli mekanik ve kesirli kuantum mekaniği gibi alanlarda çalışılmıştır. Kesirli Friedmann denklemleri, fraktal diferansiyel geometri ve kesirli varyasyonel prensiplere yönelik bazı yaklaşımlar da yakın zamanda geliştirilmiştir. Riemann-Liouville tipi kesirli diferansiyel denklem $r > 0$ için,

$$D_{x_0}^r g(x) := \frac{1}{\Gamma(n-r)} \frac{d^n}{dx^n} \int_{x_0}^x \frac{g(v)}{(x-v)^{r-n+1}} dv \quad (38)$$

şeklinde tanımlanır. (38) denkleminde n sayısı $n - 1 < r \leq n$ aralık değerlerinde r tanımlanır. D^r 'nin n boyutlu bir çekirdeğe sahip olduğu iyi bilinmektedir ve bu nedenle kesirli diferansiyel denklemin basit formunun benzersiz bir çözümünü elde etmek için kesinlikle n başlangıç koşulunun belirtilmesi gerekmektedir (Diethelm and Ford 2002).

Caputo tipi kesirli diferansiyel denklem

Kesirli mertebeden türevlerin birbirinden farklı ve birbiriyle görünürde uyuşmayan birçok tanımı literatürde mevcuttur. Bunlar Riemann-Liouville, Grünwald-Letnikov, Caputo, Hadamard, Riezs, Weyl, Abel tanımlarıdır. Fakat literatür incelediğinde, bu tanımların aslında Riemann-Liouville türev tanımının genelleştirilmiş şekli, varyantları ya da belirli şartlar altında Riemann-Liouville türev tanımı ile bağlantılı olduğu görülür. Bu tanımlar arasındaki temel fark ele alınan fonksiyonların tanım kümesi ve seçilen yardımcı parametrelerdir (Li 2003).

Kesirli hesap çalışmalarına uygulanabilen matematiksel teoremin çoğu 20. yüzyılın başlarından önce geliştirilmiştir. Bununla beraber mühendislik ve bilimsel uygulamadaki en

ilgi çekici çalışmalar son 100 yılda gerçekleşmiştir. Matematik bazı durumlarda fiziksel gerçekliğin gerekliliklerini karşılamak için değişmek zorunda kalmıştır. Caputo, Riemann-Liouville kesirli türevinin daha 'klasik' tanımını, kesirli mertebeden diferansiyel denklemlerini çözmek için tamsayı mertebeden başlangıç koşullarını kullanmak amacıyla yeniden formüle etmiştir (Loverro 2004). Çalışmamızda başlangıç koşullarını içeren kesirli diferansiyel denklemlerin nümerik çözümü esas alındığından, bu tip problemler için literatürde daha fazla tercih edilen Caputo türev tanımları aşağıda verilmiştir.

Caputo anlamında kesirli diferansiyel operatör

$$(D^\omega f)(m) = \frac{1}{\Gamma(m-\omega)} \int_0^m (m-\zeta)^{(m-\omega-1)} f^{(m)}(\zeta) d\zeta, \omega > 0, \zeta > 0 \quad (39)$$

şeklinde ifade edilir. (39) denkleminde $m-1 < \omega < m, m \in N$ şeklindedir. Bu denklemde $m-1 < \omega < m$ için D^ω ,

$$D^\omega m^l = \begin{cases} 0, l \in 0, 1, 2, \dots, [\omega] \\ \frac{\Gamma(l+1)}{\Gamma(l+1-\omega)}, l \in N, l \geq [\omega] \end{cases} \quad (40)$$

şeklinde tanımlanır. (40) denkleminde $[\omega]$ gösterimi, $[\omega] = \min\{m \in Z | m \geq \omega\}$ denklem kümesindeki tepe fonksiyonunu ifade eder (Dwivedi and Rajeev 2021).

Ayrıca $-1 < \alpha \leq m, m \in Z^+, x > 0, f \in C_a^m, \alpha \geq -1$ olmak üzere, Caputo türev tanımına ait,

$$i) (D_a^\alpha J_a^\alpha f)(x) = f(x) \quad (41)$$

$$ii) (J_a^\alpha D_a^\alpha f)(x) = f(x) - \sum_{k=0}^{m-1} f^{(k)}(a) \frac{(x-a)^k}{k!}, a \geq 0 \quad (42)$$

özellikleri mevcuttur. $m-1 < \alpha \leq m$ olmak üzere, α . mertebeden ($\alpha > 0$) Caputo zaman-kesirli türev operatörü,

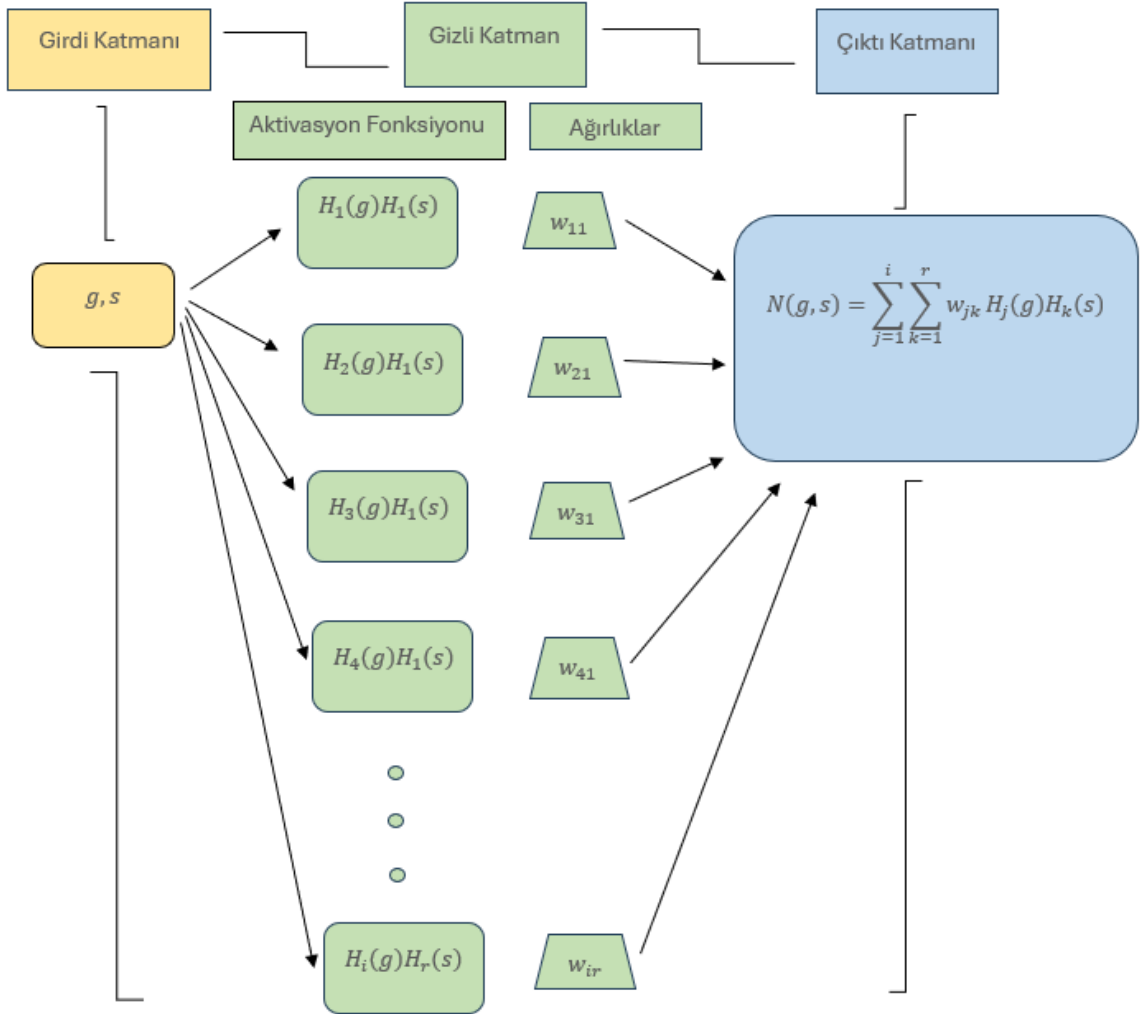
$$D_t^\alpha u(x, t) = \frac{\partial^\alpha u(x, t)}{\partial t^\alpha} = \begin{cases} \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-\tau)^{m-\alpha-1} \frac{\partial^m u(x, \tau)}{\partial \tau^m} d\tau & ; \quad m-1 < \alpha < m \\ \frac{\partial^m u(x, t)}{\partial t^m} & ; \quad \alpha = m \in N \end{cases} \quad (43)$$

biçiminde tanımlanır (Podlubny 1999).

MATERYAL VE METOT

Hosoya Yapay Sinir Ağı Mimarisi

Bu tez çalışmasında kullanılan Hosoya yapay sinir ağı mimarisinin genel modeli,



Şekil 18. Hosoya yapay sinir ağı mimarisi

şeklinde. Bu sinir ağı mimarisinde girdi katmanında yer alan g, s giriş sinyalleri, gizli katmandaki $H_i(g)H_r(s)$ Hosoya polinomunun çeşitli derecelerdeki aktivasyon fonksiyonları, w_{ir} sinir ağının ağırlık değerleri ve $N(g, s)$ ise çıktı katmanındaki aktivasyon fonksiyonu ve ağırlık değerlerinin çarpımından meydana gelen çıktı fonksiyonudur.

Hosoya Yapay Sinir Ağı Metodu

Hosoya yapay sinir ağı metodunun amacı Hosoya yapay sinir ağı mimarisi ile kısmi diferansiyel denklemlerin bir sınıfı olan kesirli dereceli adveksiyon-reaksiyon difüzyon denklemlerinin sayısal çözümünü bulmaktır. Bu çalışmada kullanılan kesirli mertebeden reaksiyon-difüzyon denklemi başlangıç ve sınır şartları ile,

$$\frac{\partial^\alpha l(u,v)}{\partial t^\alpha} = g(l,u,v) \frac{\partial^\beta l(u,v)}{\partial u^\beta} + h(l,u,v) \quad (44)$$

$$l(u, 0) = f_0(u)$$

$$l(0, v) = f_1(v) \quad (45)$$

$$l(1, v) = f_2(v)$$

şeklindedir. Burada $\frac{\partial^\alpha l(u,v)}{\partial v^\alpha}$ ve $\frac{\partial^\beta l(u,v)}{\partial u^\beta}$ denklemleri $l(u,v)$ denkleminin kesirli dereceden türevleridir. Şekil 18’de gösterilen çıktı katmanı denklemi,

$$N(g, s) = \sum_{j=1}^i \sum_{k=1}^l w_{jk} H_j(g) H_k(s) \quad (46)$$

şeklinde gösterilmişti. Daha sonra denklem (46), denklem (44)’de başlangıç ve sınır şartları ile beraber yerine yazılırsa,

$$\frac{\partial^\alpha N(g,s)}{\partial s^\alpha} = h(N, g, s) \frac{\partial^\beta N(g,s)}{\partial g^\beta} + v(N, g, s) \quad (47)$$

$$N(g, 0) = f_0(g)$$

$$N(0, s) = f_1(s) \quad (48)$$

$$N(1, s) = f_2(s).$$

elde edilir.

Caputo tipi kesirli türev denklemi olan denklem (39)’da, Hosoya polinomu tanımı olan denklem (25) yerine konulursa,

$$D^\alpha H_r(x) = \begin{cases} 0, & r \in 0, 1, 2, \dots, [\alpha] \\ \sum_{j=[\alpha]}^r \frac{(n-r)}{(r-\alpha)!} x^{(r-\alpha)}, & r \in N, r \geq [\alpha] \end{cases} \quad (49)$$

Hosoya polinomunun Caputo tipi kesirli türevi elde edilir. Eşitlik (49), eşitlik (47)' de yerine yazılırsa,

$$\frac{\partial^\alpha N(g,s)}{\partial s^\alpha} = \sum_{j=1}^i \sum_{k=1}^l w_{jk} H_j(g) \frac{\partial^\alpha H_k(s)}{\partial s^\alpha}, \quad (50)$$

$$\frac{\partial^\beta N(g,s)}{\partial s^\beta} = \sum_{j=1}^i \sum_{k=1}^l w_{jk} \frac{\partial^\beta H_j(g)}{\partial s^\beta} H_k(s)$$

elde edilir. Sonrasında başlangıç ve sınır şartları ile alınan eşitlik (47) bir minimizasyon problemine dönüştürülür. Elde edilen minimizasyon denklemi,

$$\min_{\omega} \left\{ \sum_{g_r, s_z} \left(\left(\frac{\partial^\alpha N(g_r, s_z)}{\partial s^\alpha} - h(N, g_r, s_z) \frac{\partial^\beta N(g_r, s_z)}{\partial g^\beta} - v(N, g_r, s_z) \right)^2 + \right. \right. \\ \left. \left. (N(g_r, 0) - f_0(g_r))^2 + (N(0, s_z) - f_1(s_z))^2 + (N(1, s_z) - f_2(s_z))^2 \right) \right\} \quad (51)$$

şeklindedir. (51) denkleminde g_r, s_z , yapay sinir ağının eğitme noktalarıdır. Sinir ağı eğitme işlemi tamamladıktan sonra ω ağırlık noktaları gradyan tabanlı Marquardt algoritması ile güncellenir. Marquardt algoritması hata fonksiyonunun minimizasyonunun elde edildiği ikinci dereceden bir yöntemdir. Yani bu algoritmanın uygulanmasındaki amaç yerel minimum değerlerinin elde edilmesidir.

Marquardt Algoritması Adımları

Adım 1. ω^0 ağırlık değerini tanımla (ilk tahmin değeri)

M_s =maksimum iterasyon sayısı

ε = yakınsama kriteri

Adım 2. $p = 0, \lambda^0 = 10^4$ ayarla

Adım 3. $\nabla A(\omega^p)$ değerini hesapla.

Adım 4. If $A(\omega^p) < \varepsilon$

Adım 11'e git

Else: Devam et

Adım 5. If $p \geq M_s$

Adım 11'e git

Else: Devam et

Adım 6. $z(\omega^{(p)}) = -(B^{(p)} + \lambda^{(p)}I)^{-1}\nabla A(\omega^{(p)})$ hesapla.

Adım 7. $\omega^{(p+1)} = \omega^{(p)} - z(\omega^{(p)})$

Adım 8. If $A(\omega^{(p+1)}) < A(\omega^{(p)})$

Adım 9'a git

Else: **Adım 10'a git**

Adım 9. $\lambda^{(p+1)} = \lambda^{(p)}/4$ olarak ayarla ve $p = p + 1$. Adım 3'e git.

Adım 10. $\lambda^{(p)} = 2\lambda^{(p)}$. **Adım 6'ya git**

Adım 11. Sonuçları ve grafikleri yazdır.

Marquardt algoritmasında, gradyanın bileşenleri genellikle tasarım değişkenlerinin doğrusal olmayan fonksiyonları olduğundan, gradyan tabanlı yöntemler yinelemeli bir süreç gerektirir. Bu yöntemde baştan sona kadar $A(\omega^{(p)})$, $\nabla A(\omega^{(p)})$ değerlerinin var olduğu ve sürekliliği kabul edilir. Gradyanın bileşenlerinin kapalı formda tanımlandığı ya da sayısal olarak güvenilir bir şekilde tahmin edilebildiği kabul edilir. Bu algoritmada $\omega^{(p)}$, $A(\omega^{(p)})$, $\nabla A(\omega^{(p)})$ ve $B^{(p)}$ değerleri p^{th} değerinin iterasyon sayısına göre hesaplanır. Her yinelemede $\omega^{(p)}$ ve $s(\omega^{(p)})$ nin belirlenme şekli belirli bir yöntemi tanımlar. p^{th} değeri $A(\omega^{(p)})$ 'yi $s(\omega^{(p)})$ yönünde an aza indirecek şekilde seçilir. Bu nedenle bu yöntemi uygulayabilmek için tek değişkenli minimizasyon algoritmasına ihtiyaç duyulur (Ravindran et al. 2006). Marquardt algoritma adımlarında yer alan denklemler ise

$$A(\omega^{(p)}) = \sum_{g_r s_z} \left(\left(\frac{\partial^\alpha N(g_r, s_z)}{\partial s^\alpha} - g(N, g_r, s_z) \frac{\partial^\beta N(g_r, s_z)}{\partial u^\beta} - h(N, g_r, s_z) \right)^2 + \right. \\ \left. (N(g_r, 0) - f_0(g_r))^2 + (N(0, s_z) - f_1(s_z))^2 + (N(1, s_z) - f_2(s_z))^2 \right) \quad (52)$$

$$\omega^{(p)} = \begin{bmatrix} \omega_{11} \\ \omega_{21} \\ \vdots \\ \omega_{df} \end{bmatrix}_{d \times f}, \quad \nabla A(\omega^{(p)}) = \begin{bmatrix} \frac{\partial A(\omega^{(p)})}{\partial \omega_{11}} \\ \frac{\partial A(\omega^{(p)})}{\partial \omega_{21}} \\ \vdots \\ \frac{\partial A(\omega^{(p)})}{\partial \omega_{df}} \end{bmatrix}_{d \times f} \quad (53)$$

$$B^{(p)} = \begin{bmatrix} \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{11}^2} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{11} \partial \omega_{21}} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{11} \partial \omega_{df}} \\ \vdots & & \\ \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{d1} \partial \omega_{11}} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{d1} \partial \omega_{21}} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{d1} \partial \omega_{df}} \\ \vdots & & \\ \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{df} \partial \omega_{11}} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{df} \partial \omega_{21}} & \frac{\partial^2 A(\omega^{(p)})}{\partial \omega_{df}^2} \end{bmatrix}_{df \times df} \quad (54)$$

şeklinde hesaplanır.

ARAŞTIRMA BULGULARI

Bu bölümde metot üç problem üzerinde test edilmiştir. Bu problemlerde metot kesirli dereceli adveksiyon-reaksiyon difüzyon denklemlerine uygulanmıştır. Metot için özel bir algoritma oluşturulmuştur. Bu algoritmanın Python program kodları yazılarak metodun doğruluğu analitik çözüm ve yaklaşık çözüm değerleriyle karşılaştırılarak hata sonuçları ve grafikler elde edilerek test edilmiştir. Problemlerde kullanılan hata fonksiyonu,

$$L_{\infty} = ||x_{analitik} - x_{yaklaşık}||_{\infty} = ||x_{analitik}(u_y, v_z) - x_{yaklaşık}(u_y, v_z)||_{\infty}$$

şeklindedir.

Test Problemi 1

Aşağıda tam çözümü

$$u(x, t) = x^2 + t^2$$

olarak verilen kesirli dereceli reaksiyon-difüzyon denklemi başlangıç ve sınır şartları ile beraber

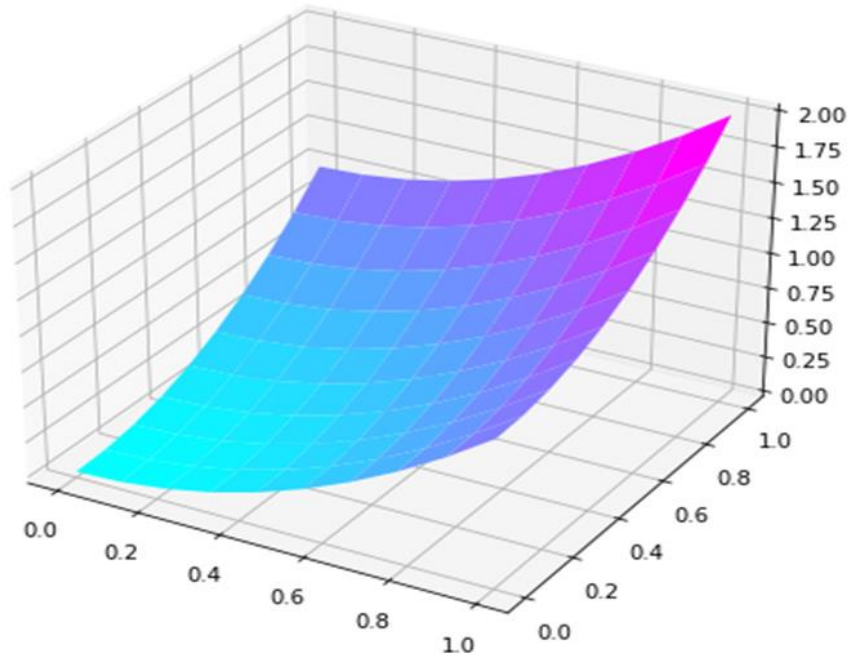
$$\frac{\partial^{\alpha} u(x, t)}{\partial t^{\alpha}} = \frac{\partial^2 u(x, t)}{\partial x^2} - \frac{\partial u(x, t)}{\partial x} + \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} + 2x - 2$$

$$u(x, 0) = x^2$$

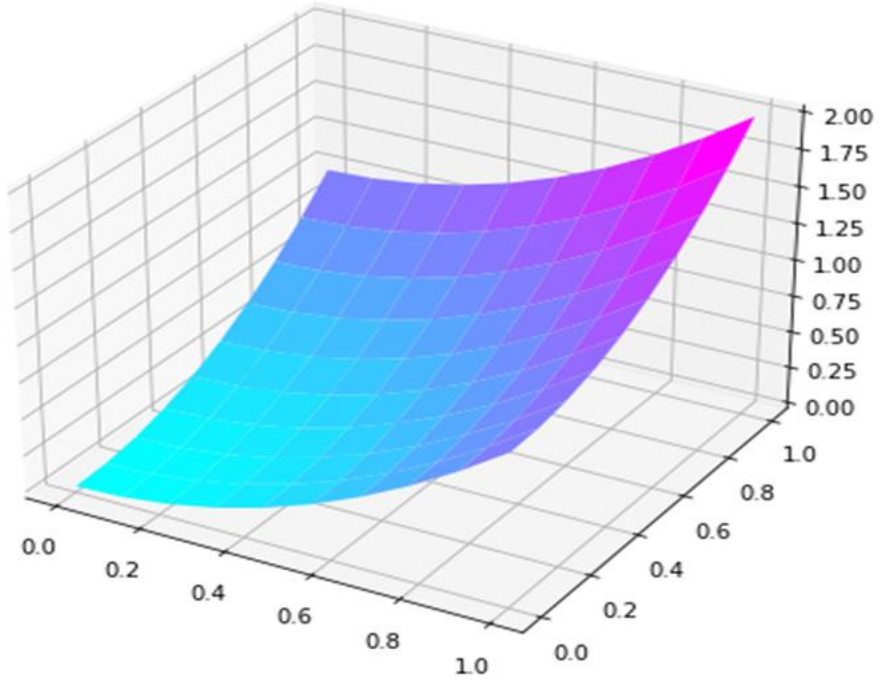
$$u(0, t) = t^2$$

$$u(1, t) = 1 + t^2$$

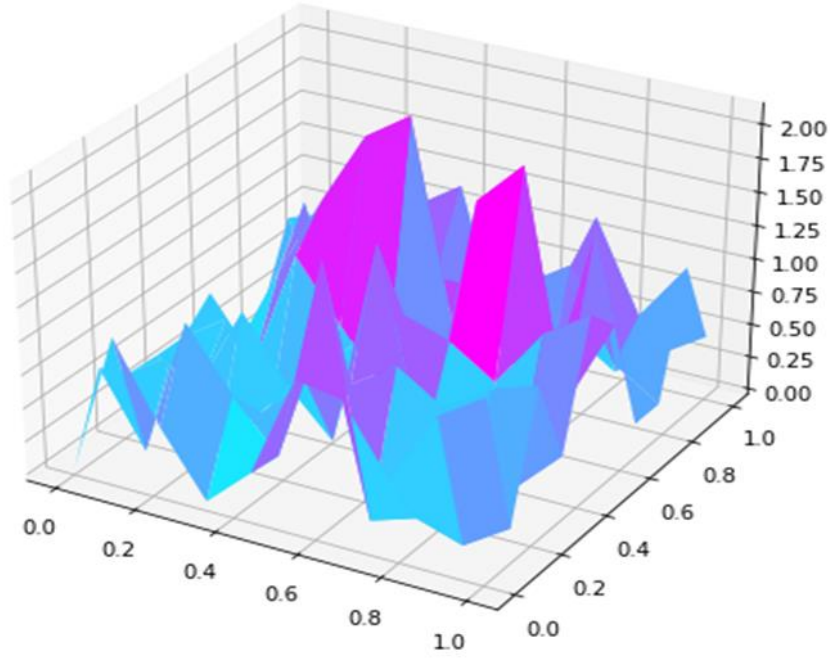
şeklindedir. Bu probleme Hosoya yapay sinir ağı metodu uygulanmış olup problem 22 iterasyonda çözülmüştür. Hosoya yapay sinir ağı metodu ile elde edilen sonuçlar, Fibonacci yapay sinir ağı metodu sonuçları ile karşılaştırılmıştır. Elde edilen yaklaşık çözüm ve analitik çözüm değerleri aşağıdaki şekil ve tablolarda gösterilmiştir.



Şekil 19. Test Problemi 1 Analitik Çözüm Grafiği



Şekil 20. Test Problemi 1 Yaklaşık Çözüm Grafiği



Şekil 21. Test Problemi 1 Hata Grafiği

Tablo 1. Test Problemi 1 Maksimum Mutlak Hata Tablosu

α	Hosoya İterasyon Sayısı	HSYA Metodu L_{∞}	Fibonacci İterasyon Sayısı	FYSA Metodu L_{∞}
0,1	22	$6,22645447e - 29$	26	$6,6613e - 16$
0,3	22	$1,08043129e - 28$	25	$1,1102e - 15$
1	22	$9,46817976e - 29$	26	$7,2164e - 16$

Tablo 1 değerleri incelendiğinde Hosoya yapay sinir ağı metodu ile çözülen problemde elde edilen sonuçların Fibonacci yapay sinir ağı metodu ile çözülen problemde elde edilen sonuçlara dayanarak daha iyi olduğu görülmüştür.

Test Problemi 2

Aşağıda tam çözümü

$$u(x, t) = x^2(1 - x)e^{-t}$$

olarak verilen kesirli dereceli reaksiyon-difüzyon denklemi başlangıç ve sınır şartları ile beraber

$$\frac{\partial u(x, t)}{\partial t} = p(x) \frac{\partial^{1.8} u(x, t)}{\partial x^{1.8}} + q(x, t)$$

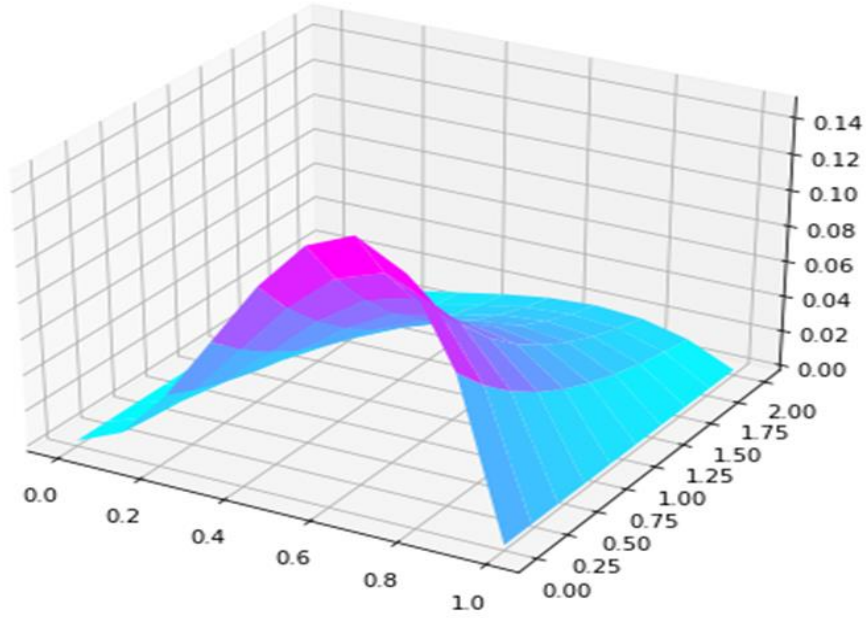
$$p(x, t) = \Gamma(1.2)x^{1.8}, \quad q(x, t) = (2x - 1)3x^2e^{-t}$$

$$u(x, 0) = x^2(1 - x), \quad 0 \leq x \leq 1,$$

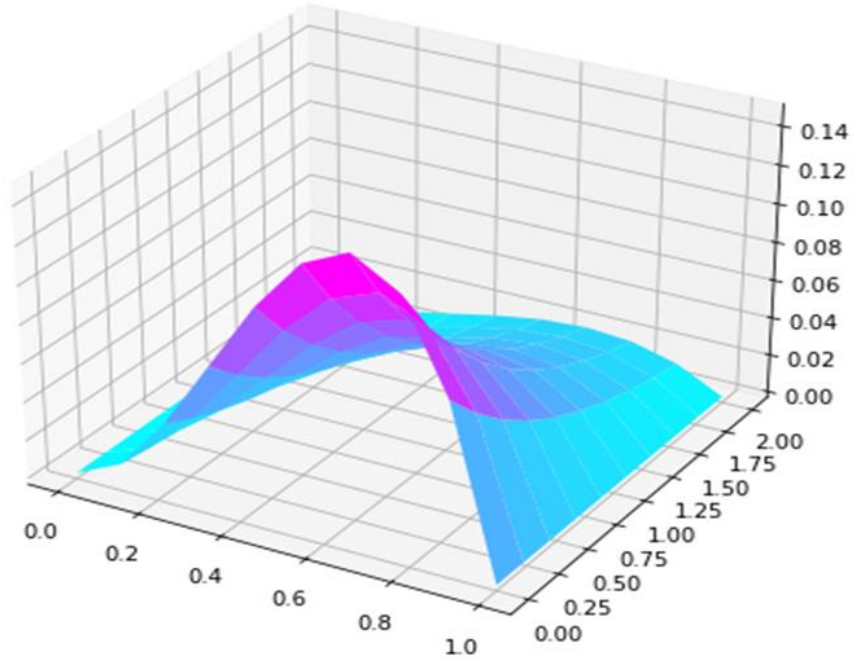
$$u(0, t) = 0, \quad t > 0,$$

$$u(1, t) = 0, \quad t > 0,$$

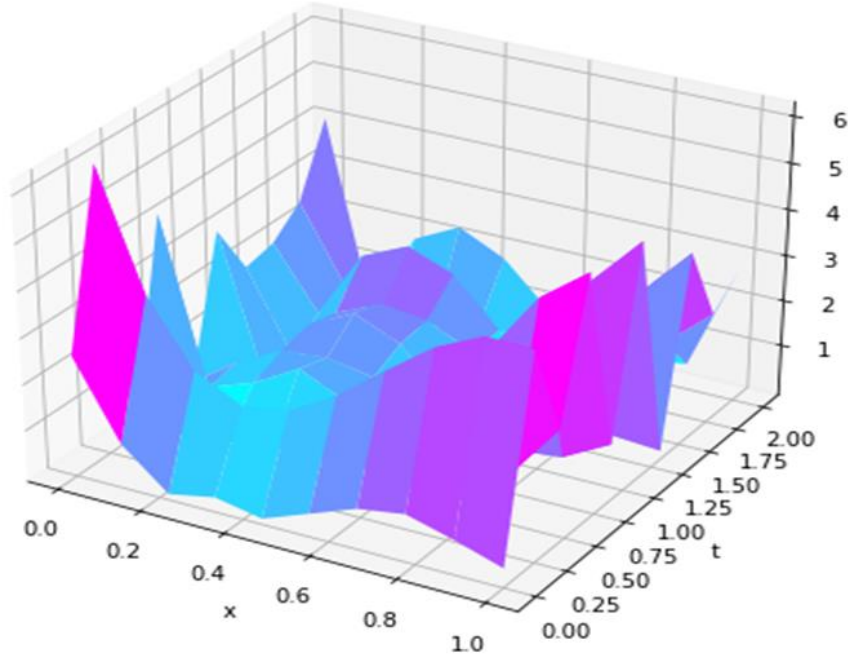
şeklindedir. Bu probleme Hosoya yapay sinir ağı metodu uygulanmış olup problem 29 iterasyonda çözülmüştür. Test problemi 2 için 29 iterasyonda elde edilen hata değeri $L_\infty = 3.273092e - 12$ şeklindedir. Hosoya yapay sinir ağı metodu ile elde edilen sonuçlar, Tau yaklaşımı ve ikinci tür kaydırılmış Chebyshev polinomları metotları ile çözülen problem sonuçları ile karşılaştırılmıştır (Saadatmandi and Dehghan 2011; Sweilam and Nagy 2015). Elde edilen yaklaşık çözüm ve analitik çözüm değerleri aşağıdaki şekil ve tablolarda gösterilmiştir.



Şekil 22. Test Problemi 2 Analitik Çözüm Grafiği



Şekil 23. Test Problemi 2 Yaklaşık Çözüm Grafiği



Şekil 24. Test Problemi 2 Hata Grafiği

Tablo 2. Test Problemi 2 Maksimum Mutlak Hata Tablosu

x	Tau Metodu	Chebyshev Metodu	Hosoya Metodu
0.0	0.00	0.00	$2.03155e - 08$
0.1	$4.66e - 5$	$5.46e - 6$	$8.36882e - 09$
0.2	$7.74e - 4$	$8.51e - 6$	$1.14053e - 09$
0.3	$5.00e - 5$	$9.60e - 6$	$1.96352e - 09$
0.4	$2.30e - 5$	$9.18e - 6$	$1.53748e - 09$
0.5	$2.74e - 5$	$7.69e - 5$	$1.82445e - 09$
0.6	$4.38e - 5$	$5.60e - 6$	$7.5281e - 09$
0.7	$3.87e - 5$	$3.33e - 6$	$1.49793e - 08$
0.8	$1.01e - 5$	$1.34e - 6$	$2.35839e - 08$
0.9	$3.35e - 5$	$8.39e - 8$	$3.27476e - 08$
1.0	0.00	0.00	$4.18764e - 08$

Tablo 2 değerleri incelendiğinde Hosoya yapay sınır ağı metodu ile çözülen problemten Tau ve Chebyshev metotları ile çözülen probleme göre elde edilen sonuçların daha iyi olduğu görülmüştür.

Test Problemi 3

Aşağıda tam çözümü

$$u(x, t) = x^3 e^{-t}$$

olarak verilen kesirli dereceli reaksiyon-difüzyon denklemi başlangıç ve sınır şartları ile beraber

$$\frac{\partial u(x, t)}{\partial t} = p(x, t) \frac{\partial^{1.8} u(x, t)}{\partial x^{1.8}} + q(x, t)$$

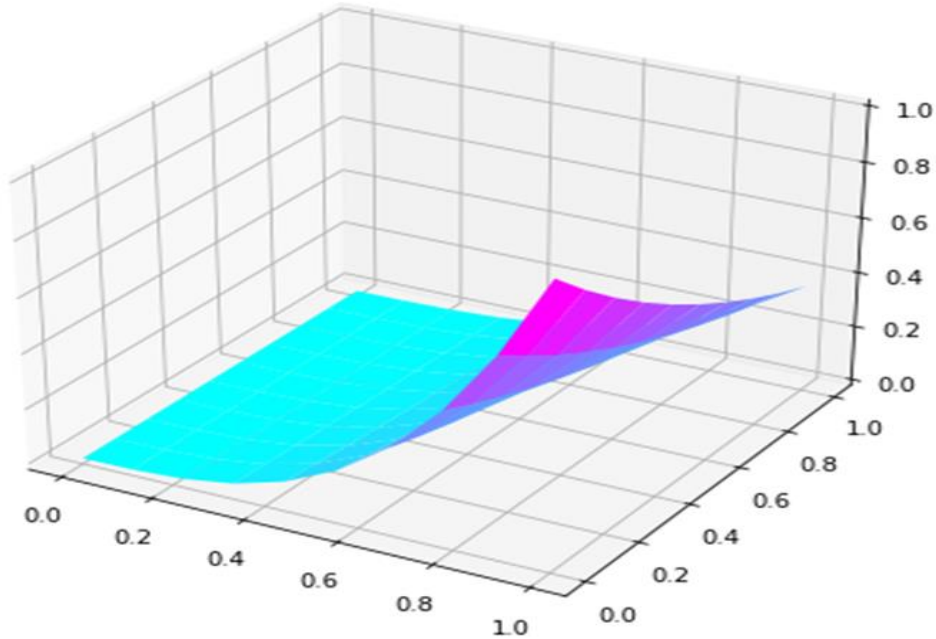
$$p(x, t) = \frac{\Gamma(2.2)}{6} x^{2.8}, \quad q(x, t) = -(1 - x)x^3 e^{-t}$$

$$u(x, 0) = x^3, \quad 0 \leq x \leq 1,$$

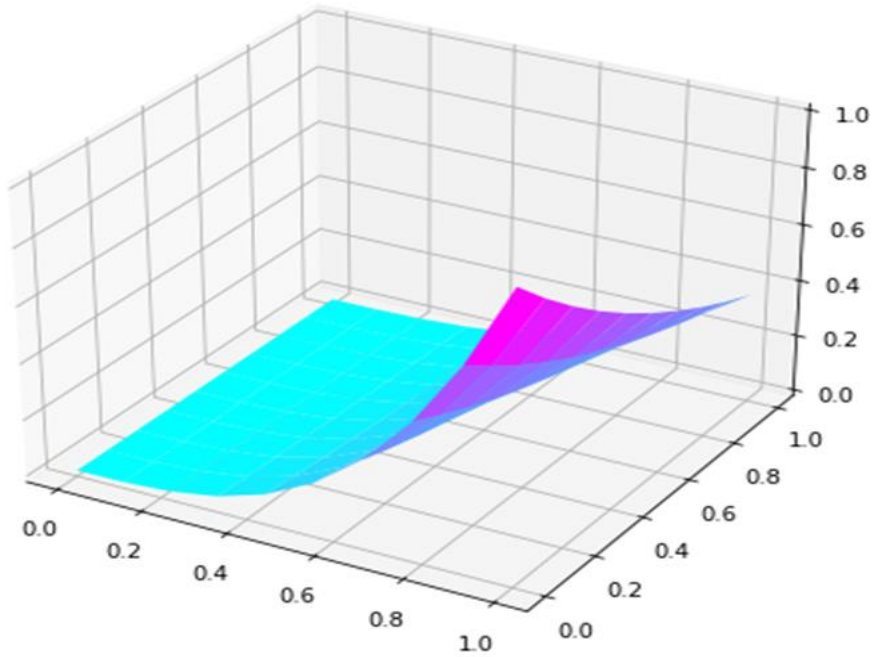
$$u(0, t) = t, \quad t > 0,$$

$$u(1, t) = e^{-t}, \quad t > 0,$$

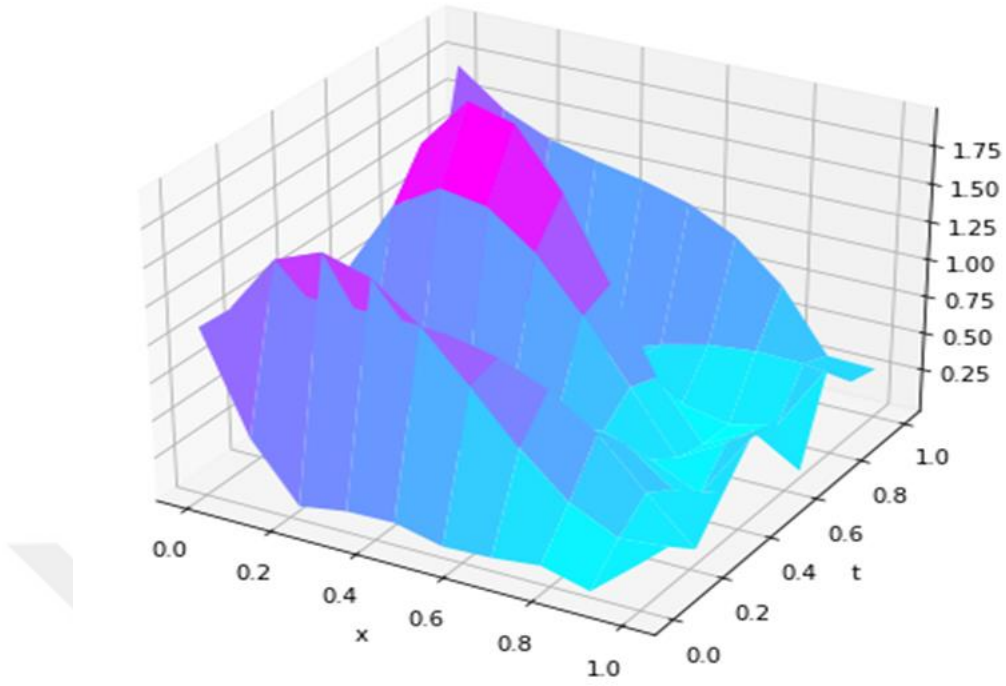
şeklindedir. Bu probleme Hosoya yapay sinir ağı metodu uygulanmış olup problem 29 iterasyonda çözülmüştür. Hosoya yapay sinir ağı metodu ile elde edilen sonuçlar, Fibonacci yapay sinir ağı metodu sonuçları ile karşılaştırılmıştır. Elde edilen yaklaşık çözüm ve analitik çözüm değerleri aşağıdaki şekil ve tablolarda gösterilmiştir.



Şekil 25. Test Problemi 3 Analitik Çözüm Grafiği



Şekil 26. Test Problemi 3 Yaklaşık Çözüm Grafiği



Şekil 27. Test Problemi 3 Hata Grafiği

Tablo 3. Test Problemi 3 Maksimum Mutlak Hata Tablosu

x	İterasyon Sayısı	HSYA Metodu L_{∞}	FSYA Metodu L_{∞}
0.0	29	$1.27883e - 07$	$1.61148e - 07$
0.1	29	$1.40269e - 07$	$1.67595e - 07$
0.2	29	$1.43928e - 07$	$1.67475e - 07$
0.3	29	$1.39490e - 07$	$1.60802e - 07$
0.4	29	$1.27588e - 07$	$1.47590e - 07$
0.5	29	$1.08854e - 07$	$1.27852e - 07$
0.6	29	$0.83920e - 07$	$1.01602e - 07$
0.7	29	$5.34177e - 08$	$6.88543e - 08$
0.8	29	$1.79788e - 08$	$2.96218e - 08$
0.9	29	$2.17646e - 08$	$1.60814e - 08$
1.0	29	$6.51807e - 08$	$6.82415e - 08$

Tablo 3 deęerleri problemdeki x deęerlerinin eřitli deęerleri iin maksimum hata L_{∞} hesaplanarak elde edilmiřtir. Tablo 3 deęerleri incelendięinde Hosoya yapay sinir aęı metodu ile özölen problemde elde edilen sonuların Fibonacci yapay sinir aęı metodu ile özölen problemde elde edilen sonulara göre daha iyi olduęu görölmüřtür.



TARTIŞMA VE SONUÇ

Bu tez çalışmasında temel olarak kısmi diferansiyel denklemlerin çözümü için yapay sinir ağı mimarisinin uygulanması araştırılmıştır. Tek gizli katmanlı bir yapay sinir ağı ile geliştirilen Hosoya yapay sinir ağı metodu ile kesirli dereceli adveksiyon-reaksiyon difüzyon denklemlerinin çözümleri incelemiştir. İlk olarak çalışmada kullanılacak olan kısmi diferansiyel denklem tipi belirlenerek Hosoya yapay sinir ağı mimarisi oluşturulmuştur. Marquardt algoritması ile oluşturulan belirli bir algoritmayla Python kodları yazılarak Hosoya yapay sinir ağı metodu, 3 farklı kesirli dereceli adveksiyon-reaksiyon difüzyon denklemine uygulanmıştır.

Test problemi 1’de adveksiyon-reaksiyon difüzyon denklemi için $\alpha = 0,1$, $\alpha = 0,3$ ve $\alpha = 1$ kesirli dereceleri seçilerek denklemin mutlak hata sonuçları Hosoya yapay sinir ağı metodu için 22 iterasyonda hesaplanmış olup Fibonacci yapay sinir ağı metodu sonuçları ile karşılaştırılmıştır. Elde edilen yaklaşık çözüm, analitik çözüm ve hata grafikleri Şekil 19, Şekil 20 ve Şekil 21’de gösterilmiştir. Test problemi 2’de adveksiyon-reaksiyon difüzyon denkleminin kesirli derecesi $\alpha = 1,8$ seçilmiş olup x ’in farklı değerleri için 29 iterasyonda hesaplama yapılmıştır. Bu test probleminde Tau metodu ve Chebyshev metodu ile elde edilen sonuçlar Hosoya metodu ile elde edilen sonuçlar ile karşılaştırılmıştır. Elde edilen yaklaşık çözüm, analitik çözüm ve hata grafikleri Şekil 22, Şekil 23 ve Şekil 24’te gösterilmiştir. Test problemi 3’te ise kesirli derecesi test problemi 2’de olduğu gibi $\alpha = 1,8$ olan difüzyon denklemi x ’in farklı değerleri için 29 iterasyonda hesaplanarak Hosoya yapay sinir ağı ile elde edilen hata sonuçları Fibonacci yapay sinir ağı ile elde edilen hata sonuçları ile karşılaştırılmıştır. Elde edilen yaklaşık çözüm, analitik çözüm ve hata grafikleri Şekil 25, Şekil 26 ve Şekil 27’de gösterilmiştir.

Her bir test probleminin farklı parametrelerindeki değişikliklere bağlı olarak problemlerin çözüm sonuçlarından elde edilen yaklaşık çözüm, analitik çözüm, maksimum hata oranı ve grafikler, tablo ve şekillerle gösterilmiştir. Hosoya yapay sinir ağı metodu ile farklı iterasyon sayıları ve difüzyon denkleminin çeşitli kesirli dereceleri için hesaplanan sonuçların Chebyshev, Tau ve Fibonacci yapay sinir ağı metodlarına kıyasla daha verimli sonuçların elde edildiği görülmüştür. Bu tez çalışmasında çözülen 3 farklı problem ile Hosoya yapay sinir ağı yönteminin daha önce çözülmüş mevcut yöntemlerden daha iyi performans gösterdiği görülmüş

olup yapay sinir ağı metodunun diferansiyel denklemlerin çözümünü daha doğru ve kolay şekilde bulmaya yardımcı olabileceği kanısına varılmıştır.



KAYNAKLAR

- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H., 2018. State-of-the-art in artificial neural network applications. A survey, Heliyon, 4(11).
- Abraham, A., 2005. Artificial neural networks. Handbook of measuring system design.
- Anderson, D., and McNeill, G., 1992. Artificial neural networks technology. Kaman Sciences Corporation, 258(6), 1-83.
- Anonymous, <https://www.geeksforgeeks.org/activation-functions-neural-networks> (2023)
- Anonymous, <https://www.geeksforgeeks.org/effect-of-bias-in-neural-network> (2018)
- Anonymous, <https://www.pico.net/kb/the-role-of-bias-in-neural-networks> (2024)
- Anonymous, <https://www.geeksforgeeks.org/what-is-perceptron-the-simplest-artificial-neural-network> (2023)
- Anonymous, https://mrcet.com/downloads/digital_notes/CSE/III%20Year/AIML/Neural-Networks.pdf (2022)
- Anonymous, https://www.d.umn.edu/~rmaclin/cs8751/fall2005/Notes/L04_Neural_Networks.pdf (2005)
- Ar, N. and Farman, T., 2018. Hosoya Polynomial and Topological Indices of the Jahangir Graph $J7,m$, Journal of Applied & Computational Mathematics, 7, 1-5.
- Basheer, I. A., and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), 3-31.
- Bishop, M. J., 2015. History and philosophy of neural networks.
- Bonchev, D., 1983. Information theoretic indices for characterization of chemical structures, Research Studies Press, No. 5.
- Das, S., Singh, A., and Ong, S. H., 2018. Numerical solution of fractional order advection-reaction-diffusion equation.
- Diethelm, K., and Ford, N. J., 2002. Analysis of fractional differential equations. Journal of Mathematical Analysis and Applications, 265(2), 229-248.
- Diudea, M., 1996. Wiener and hyper-Wiener numbers in a single matrix. Journal of chemical information and computer sciences, 36, 833-836.
- Dongare, A. D., Kharde, R. R., and Kachare, A. D., 2012. Introduction to artificial neural network. International Journal of Engineering and Innovative Technology (IJEIT), 2(1), 189-194.
- Dutka, J., 1981. The incomplete beta function—a historical profile. Archive for history of exact sciences, 11-29.
- Dwivedi, K. D., and Rajeev., 2021. Numerical solution of fractional order advection reaction diffusion equation with fibonacci neural network. Neural Processing Letters, 53(4), 687-2699.
- El Arabi, I., Chafi, A., and Alami, S. K., 2022. Numerical simulation of the advection-diffusion-reaction equation using finite difference and operator splitting methods: Application on the 1D transport problem of contaminant in saturated porous media. In E3S Web of Conferences, 351, p. 01003.

- Farahani, M. R., 2013. On the Schultz polynomial and Hosoya polynomial of circumcoronene series of benzenoid. *Journal of applied mathematics & informatics*, 31(5-6), 595-608.
- Fernandez, A., and Baleanu, D., 2021. Classes of operators in fractional calculus: a case study. *Mathematical Methods in the Applied Sciences*, 44(11), 9143-9162.
- Geçmen, M. Z., Çelik, E., and Dokuyucu, M. A., 2021. Numerical solution of Volterra integral equations using Hosoya polynomial. *Turkish Journal of Science*, 6(3), 110-117.
- Geçmen, M. Z., and Çelik, E., 2021. Numerical solution of Volterra–Fredholm integral equations with Hosoya polynomials. *Mathematical Methods in the Applied Sciences*, 44(14), 11166-11173.
- Gharib, M., Khezri, M., and Foster, S. J., 2017. Meshless and analytical solutions to the time-dependent advection-diffusion-reaction equation with variable coefficients and boundary conditions. *Applied Mathematical Modelling*, 49, 220-242.
- Gutman, I., Klavzar, S., Petkovšek, M., and Zigert, P., 2001. On Hosoya polynomials of benzenoid graphs. *MATCH Commun. Math. Comput. Chem*, 43, 49-66.
- Harmon, M. E., and Harmon, S. S., 1996. Reinforcement learning: A tutorial. WL/AAFC, WPAFB Ohio, 45433, 237-285.
- Haubold, H. J., Mathai, A. M., and Saxena, R. K., 2011. Mittag-Leffler functions and their applications. *Journal of applied mathematics*, 2011(1), 298628.
- Hauke, G., and García-Olivares, A., 2001. Variational subgrid scale formulations for the advection–diffusion-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 190(51-52), 6847-6865.
- Hopfield, J. J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
- Hosoya, H., 1988. On some counting polynomials in chemistry, *Discrete Appl. Math.*, 19, 239–257.
- Huang, Y., 2009. Advances in artificial neural networks–methodological development and application. *Algorithms*, 2(3), 973-1007.
- Karlik, B., and Olgac, A. V., 2011. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111-122.
- Ke, J., and Liu, X., 2008. Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Vol. 2, pp. 828-832.
- Khalil Alsmadi, M., Omar, K. B., Noah, S. A., and Almarashdah, I., 2009. Performance comparison of multi-layer perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in neural networks. In *2009 IEEE International Advance Computing Conference*, pp. 296-299.
- Krenker, A., Bešter, J., and Kos, A., 2011. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications*. InTech, 1-18.
- Lazarević, M. P., Rapaić, M. R., Šekara, T. B., Mladenov, V., and Mastorakis, N., 2014. Introduction to fractional calculus with brief historical background. *Advanced topics on applications of fractional calculus on control problems, system stability and modeling*, -16.

- Loverro, A., 2004. Fractional calculus: history, definitions and applications for the engineer. Rapport technique, Univeristy of Notre Dame: Department of Aerospace and Mechanical Engineering, 1-28.
- Li, X., 2003. Fractional Calculus, Fractal Geometry and Stochastic Processes, PhD Thesis, The University of Western Ontario, Canada.
- Md Nasrudin, F. S., Phang, C., and Kanwal, A., 2023. Fractal-fractional advection–diffusion–reaction equations by Ritz approximation approach. *Open Physics*, 21(1).
- Minsky, M., and Papert, S., 1969. An introduction to computational geometry. Cambridge tiass., HIT, 479(480), 104.
- Mishra, M., and Srivastava, M., 2014. A view of artificial neural network. In 2014 international conference on advances in engineering & technology research (ICAETR-2014), pp. 1-3.
- Munakata, Y., and Pfaffly, J., 2004. Hebbian learning and development. *Developmental science*, 7(2), 141-148.
- Naeem, M., Rizvi, S. T. H., and Coronato, A., 2020. A gentle introduction to reinforcement learning and its application in different fields. *IEEE access*, 8.
- Narkhede, M. V., Bartakke, P. P., and Sutaone, M. S., 2022. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1), 291-322.
- Podlubny, I., 1999. *Fractional Differential Equations*, Academic Press, San Diego, California.
- Ramane, H. S., Shiralashetti, S. C., Mundewadi, R. A., and Jummannaver, R. B., 2017. Numerical solution of Fredholm integral equations using Hosoya polynomial of path
- Ravindran, A., Ragsdell, K. M., and Reklaitis, G. V., 2006. *Methods and Applications*. 681, New Jersey, Canada graphs. *American Journal of Numerical Analysis*, 5(1), 11- 15.
- Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Rosenblatt, F., 1961. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Cornell Aeronautical Lab Inc Buffalo NY.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 6, 71(599-607).
- Saadatmandi, A., and Dehghan, M., 2011. A tau approach for solution of the space fractional diffusion equation. *Computers & Mathematics with Applications*, 62(3), 1135-1142.
- Sadeghieh, A., Alikhani, S., Ghanbari, N., and Khalaf, A. J. M., 2017. Hosoya polynomial of some cactus chains. *Cogent Mathematics*, 4(1), 1305638.
- Salim, T. O., 2009. Some properties relating to the generalized Mittag-Leffler function. *Adv. Appl. Math. Anal*, 4(1), 21-30.
- Schalkoff, R. J., 1997. *Artificial neural networks*. McGraw-Hill Higher Education.
- Sheela, K. G., and Deepa, S. N., 2013. Review on methods to fix number of hidden neurons in neural networks. *Mathematical problems in engineering*, 2013(1), 425740.
- Shihab, K., 2006. A backpropagation neural network for computer network security. *Journal of Computer Science*, 2(9), 710-715.

- Sibi, P., Jones, S. A., and Siddarth, P., 2013. Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, 47(3), 1264-1268.
- Suzuki, K. (Ed.), 2011. *Artificial neural networks: methodological advances and biomedical applications*. BoD–Books on Demand, Rijeka-Croatia, 362.
- Sweilam, N. H., Nagy, A. M., and El-Sayed, A. A., 2015. Second kind shifted Chebyshev polynomials for solving space fractional order diffusion equation. *Chaos, Solitons & Fractals*, 73, 141-147.
- Thakur, A., and Konde, A., 2021. Fundamentals of neural networks. *International Journal for Research in Applied Science and Engineering Technology*, 9(VIII), 407-426.
- Widrow, B., and Lehr, M. A., 1990. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9), 1415-1442.
- Yan, Z., Zhong, S., Lin, L., and Cui, Z., 2021. Adaptive Levenberg–Marquardt algorithm: A new optimization strategy for Levenberg–Marquardt neural networks. *Mathematics*, 9(17), 2176.
- Yang, G. R., and Wang, X. J., 2020. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6), 1048-1070.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Merve Zeynep KAYA
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
Tel:	
E-mail:	
Eğitim	
Lise:	
Lisans:	
Yüksek lisans:	
Doktora:	
Yabancı Dil Bilgisi	
Tezden Üretilmiş Yayınlar	