

T.C.
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KİSMİ TÜREVLİ DİFERANSİYEL DENKLEMLERİN DERİN
ÖĞRENME VE MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE
SAYISAL ÇÖZÜMLERİ

DOKTORA TEZİ
Burcu Ece ALP
1800005232

Anabilim Dalı : Matematik ve Bilgisayar Bilimleri
Programı : Matematik

Tez Danışmanı : Dr. Öğr. Üyesi Mehmet Fatih UÇAR

ŞUBAT 2025

T.C.
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KİSMİ TÜREVLİ DİFERANSİYEL DENKLEMLERİN DERİN
ÖĞRENME VE MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE
SAYISAL ÇÖZÜMLERİ

DOKTORA TEZİ
Burcu Ece ALP
1800005232

Anabilim Dalı: Matematik ve Bilgisayar Bilimleri
Programı: Matematik

Tez Danışmanı: Dr. Öğr. Üyesi Mehmet Fatih UÇAR
Diğer Jüri Üyeleri: Prof. Dr. Ozan KOCADAĞLI
Doç Dr. Alper ÜLKER
Dr. Öğr. Üyesi Canan AKKOYUNLU
Dr. Öğr. Üyesi Selmahan SELİM

ŞUBAT 2025

ÖNSÖZ

Bu çalışma boyunca bana yol gösteren ve destek olan danışman hocam sayın Dr. Öğr. Üyesi Mehmet Fatih UÇAR' a, doktora öğrenimim boyunca sabır ve özveriyle yanımda olan, bilgi birikimini benimle paylaşan tez izleme jüri komitesi üyeleri değerli hocalarıma teşekkürlerimi sunarım. Eğitim hayatım boyunca çok fazla fedakarlık yapıp her zaman yanımda olan ve beni hep destekleyen canım anneme de teşekkürü bir borç bilirim.

İçindekiler

ÖNSÖZ	i
İÇERİK	ii
KISALTMALAR	iv
ŞEKİL LİSTESİ	v
TABLO LİSTESİ	vii
TÜRKÇE ÖZET	viii
İNGİLİZCE ÖZET	ix
1 GİRİŞ	1
2 KAYNAK ÖZETİ	3
2.1 Kısmi Diferansiyel Denklemlerin Sayısal Çözümlerine Bakış ..	3
2.2 Makine Öğreniminin Kısmi Diferansiyel Denklemlerin Sayısal Çözümlerinde Etkisi ..	4
2.3 Yapay Sinir Ağlarının Kısmi Diferansiyel Denklemlerin Sayısal Çözümlerine Etkisi ..	5
2.3.1 Fizikle Öğrenen Sinir Ağları(PINN) ..	6
3 METODOLOJİ	7
3.1 Yapay Zeka ..	7
3.2 Makine Öğrenimi ..	8

3.2.1	Makine Öğrenimi Türleri	8
3.3	Derin Öğrenme	9
3.4	Yapay Sinir Ağları	10
3.4.1	Yapay Sinir Ağı Mimarisi	13
3.5	Fizikle Öğrenen Sinir Ağları (PINN)	16
3.5.1	Modelleme ve Hesaplama	18
4	KESİRLİ DİFÜZYON DENKLEMLERİN PINN İLE SA- YISAL ÇÖZÜMLERİ	22
5	DİFERANSİYEL DENKLEM SİSTEMLERİNİN PINN YÖNTEMİ İLE SAYISAL ÇÖZÜMLERİ	27
5.1	Adi Difereansiyel Denklem Sistemi	27
5.2	Benney-Roskes/Zakharov-Rubenchik Sistemi	29
6	SONUÇ	33
	KAYNAKÇA	43

KISALTMALAR

YZ	: Yapay Zeka
YSA	: Yapay Sinir Ağı
PINN	: Physics Informed Neural Network
KDD	: Kısmi Diferensiyel Denklem
PDE	: Partial Differential Equation
ADD	: Adi Diferansiyel Denklem
DÖ	: Derin Öğrenme
BR/ZR	: Benny-Roskes/ Zakharov-Rubenchik

Şekil Listesi

Şekil 3.1	Yapay zeka, makine öğrenimi ve derin öğrenme arasındaki ilişkinin grafiksel özeti	10
Şekil 3.2	Yapay sinir ağı yapısı [6]	11
Şekil 3.3	Yaygın olarak kullanılan aktivasyon fonksiyonlarının grafikleri .	13
Şekil 3.4	Üstte yer alan şematik gösterim, Dirichlet benzeri sınır koşullarına sahip iki boyutlu doğrudan bir problemle ilişkili kısmi diferansiyel denklemin çözümü için uygulanan Fizikle Öğrenen Sinir Ağının (PINN) yapısını göstermektedir. Girdi katmanı, iki uzay koordinat değişkeni x ve y 'yi temsil eden iki giriş değişkeni içerir. Ağ, her biri beş nörondan oluşan ve hem giriş hem de çıkış katmanlarına bağlanan üç gizli katmandan oluşmaktadır. İkinci katman, tahmin edilen çözümü $u_\theta(x, y)$ temsil eden tek bir değişkene (bir nöron) sahiptir. Otomatik Türevlendirme, kısmi diferansiyel denklemin kayıp fonksiyonu $L_{KDD}(\theta)$ 'yı oluşturmak için gerekli kısmi türevleri (yani $u_{x,\theta}, u_{xy\theta} \dots$) değerlendirmek için prosedürde kullanılır. Kayıp fonksiyonu $L_{veri}(\theta)$, $\partial\Omega$ sınır alanına uygulanan yumuşak kısıtlamalarla (yani eğitim veri seti aracılığıyla) elde edilir.	21
Şekil 4.1	$N = 2^7$ ve $\Delta t = 1/128$ için çözülen Örnek 1 in çözüm grafiği . .	24
Şekil 4.2	Örnek 1 için hatanın davranışı	24
Şekil 4.3	Örnek 2'nin çözüm grafikleri	25
Şekil 4.4	Örnek 2 için hata davranışı	26
Şekil 5.1	PINN ve analitik çözümün karşılaştırılması	28
Şekil 5.2	PINN modeli ile verilen mutlak hata yüzeyi	29
Şekil 5.3	3 boyutlu grafiği	31
Şekil 5.4	Hata grafiği	31

Şekil 5.5	Hata grafiği	32
-----------	------------------------	----



Tablo Listesi

Tablo 3.1	Aktivasyon fonksiyonların matematiksel gösterimi	13
Tablo 4.1	Maksimum mutlak hatalar ($\Delta t = -\frac{1}{128}$)	24
Tablo 4.2	Maksimum mutlak hatalar, $\Delta t = 0.01$	25
Tablo 5.1	Literatür çalışmaları ve farklı parametrelerdeki PINN sonuçları- nın karşılaştırılması	28
Tablo 5.2	Farklı parametrelerde PINN hata analizi	31

Enstitüsü : Lisansüstü Eğitim Enstitüsü

Anabilim Dalı : Matematik ve Bilgisayar Bilimleri

Programı : Matematik Doktora

Tez Danışmanı : Dr. Öğr. Üyesi Mehmet Fatih UÇAR

Tez Türü ve Tarihi : Doktora - Şubat 2025

ÖZET

KİSMİ TÜREVLİ DİFERANSİYEL DENKLEMLERİN DERİN ÖĞRENME VE MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE SAYISAL ÇÖZÜMLERİ

Burcu Ece ALP

Diferansiyel denklemler, hesaplamalı bilimlerde çeşitli doğa olayların modellenmesinde kullanılmaktadır. Sayısal yöntemler bu tür denklemleri çözmek için bir seçenek olmuştur. Yarı analitik ve yakınsak çözüm fonksiyonları da, sayısal çözümler gibi bir çok araştırmacı tarafından tercih edilmektedir. Makine öğrenmesi ve derin öğrenme yöntemlerinin alt başlığı olan yapay sinir ağları, çeşitli diferansiyel denklem türlerini çözerek sürekli çözüm fonksiyonları oluşturmak için giderek daha fazla kullanılmaktadır.

Bu tezde, kısmi türevli diferansiyel denklemleri derin öğrenme ve makine öğrenimi algoritması ile modellemek için fizikle öğrenen sinir ağları (PINN) metodu kullanılmaktadır. PINN rezidü, başlangıç ve sınır koşulu parçalarından oluşan fiziksel kayıp fonksiyonunu en az indirmek için eğitilerek kısmi diferansiyel denkleme yaklaşık çözümler üretmektedir.

Sonlu bir alan üzerinde değişken katsayılı kesirli difüzyon denklemlerini, diferansiyel denklem sistemlerinin sayısal çözümlerinde PINN metodunun uygulanabilirliği gösterilecektir. Oluşan hata analizine göre, mevcut PINN yönteminin dikkate alınan hesaplama uzay-zaman alanında doğru çözümler sağlayabildiği sunulacaktır.

Anahtar Kelimeler : kısmi türevli diferansiyel denklemler, PINN, kesirli kısmi diferansiyel denklemler, derin öğrenme, makine öğrenimi, diferansiyel denklem sistemleri



University : İstanbul Kültür University

Institute : Institute of Graduate Studies

Department : Mathematics and Computer Science

Programme : Mathematics

Supervisor : Dr. Mehmet Fatih UCAR

Degree Awarded and Date : PHD - February 2025

ABSTRACT

NUMERICAL SOLUTIONS OF PARTIAL DIFFERENTIAL EQUATIONS WITH DEEP LEARNING AND MACHINE LEARNING ALGORITHMS

Burcu Ece ALP

In the domain of computational sciences, differential equations play a pivotal role in the modeling of diverse natural phenomena. Numerical methods have emerged as a prevalent approach for addressing these equations. Researchers have demonstrated a strong predilection for semi-analytic and convergent solution functions, along with a preference for numerical solutions. A notable trend in recent times has been the increased utilization of artificial neural networks, a subfield of machine learning and deep learning methods, for generating continuous solution functions through the solution of various types of differential equations.

In this thesis, the physics learning neural network (PINN) method is employed to model partial differential equations with deep learning and machine learning algorithms. The PINN model is trained to minimize the physical loss function, which consists of residual, initial, and boundary condition components, thereby producing approximate solutions to the partial differential equation.

The efficiency of the PINN method in numerical solutions of systems of differential equations, including fractional diffusion equations with variable coefficients over a finite domain, will be demonstrated. A detailed error analysis will be conducted to demonstrate the efficacy of the existing PINN method in producing accurate solutions within the specified computational space-time domain.

Keywords : partial differential equation, physics informed neural networks, fractional partial differential equations, deep learning, machine learning, system of differential equations.



BÖLÜM 1

GİRİŞ

Birçok mühendislik problemleri ve fen bilimleri uygulamalarında karşılaştığımız kısmi türevli diferansiyel denklemlerin analitik çözümleri kimi zaman zor veya karmaşık hesaplamalar içermektedir. Çünkü bu problemlerinin çoğu analitik çözüm yöntemlerinin kullanımını engelleyen karmaşık geometriler, karışık sınır koşulları ve/veya doğrusal olmayan özellikler içermektedir [52]. Bununla birlikte, sayısal yöntemler ile çözümü olması halinde de bu çözümler her zaman avantajlı değildir; sayısal yöntemlerin hesaplamalı fizik problemlerinin çözümüne uygulanması için ilgili alanın discretize edilebilmesi, diferansiyel denklemin ve sınır, başlangıç koşullarının uzay ve zamanda ayrık noktalarda geçerli olan yaklaşık, cebirsel denklemlere dönüştürülebilmesi ve kısmi diferansiyel denklemlerin çözümlerinin hesaplanmasının maliyetli ve uzun zaman gerektirici gibi engelleri aşmasını gerektirmektedir. Bu denklemlerin istenilen çözümlere daha pratik olarak bizi ulaştırabilecek bazı yeni yöntemler geliştirilmiştir. Bu yöntemleri kullanarak denklemlerin sayısal çözümleri daha kısa zamanda istenilen şartlara uygun olarak elde edilebilmektedir.

Sinir ağlarının fonksiyon yaklaştırma yeteneklerinden yararlanan veriye dayalı modelleri, yukarıdaki gerekliliklere tabi olmadan ve eğitildikten sonra minimum hesaplama çabasıyla hızlı bir şekilde doğru sonuçlar verebildikleri için sayısal yöntemler uygulayan modellere iyi bir alternatiftir [65]. Büyük ölçüde paralel programlama genel amaçlı hesaplama yapabilen grafik işlem birimlerinin (GPU'lar) mevcudiyeti (örneğin, NVIDIA'nın CUDA uyumlu GPU'ları) ve güçlü, açık kaynaklı makine öğrenimi yazılımlarının erişilebilirliği (örneğin, Python' da TensorFlow kütüphaneleri) de sinir ağlarının hesaplamalı fizik problemlerine uygulanmasına olan ilginin artmasına katkı sağlamaktadır. Bununla birlikte, geleneksel sinir ağı çalışmaları, ilgilenilen alan üzerinde bir fonksiyona doğru bir şekilde yaklaşmak için tipik olarak büyük eğitim veri kümeleri gerektirir.

PINN gözlemlenen verilere uyum sağlama ve KDD' ler üzerindeki rezidüleri en aza indirme gibi ikili bir amacı olan yeni bir yapay zeka çerçevesidir. Rezidü, başlangıç ve sınır koşulu parçalarından oluşan fiziksel kayıp fonksiyonunu en az indirmek için eğitilerek kısmi diferansiyel denkleme yaklaşık çözümler üretebildiği anlaşılmaktadır. Bu tez çalışmasında, sonlu bir alan üzerinde değişken katsayılı kesirli difüzyon denklemlerinin, diferansiyel denklem sistemlerinin sayısal çözümlerinde PINN metodunun uygulanabilirliği gösterilecektir. Oluşan hata analizine göre, mevcut PINN yönteminin dikkate alınan hesaplama uzay-zaman alanında doğru çözümler sağlayabildiği sunulacaktır[24]. Tez çalışmasının bölümlerinde anlatılacak konular aşağıdaki şekilde yapılandırılmıştır.

- Bölüm 2'de bölümünde literatürdeki ilgili çalışmalar anlatılacaktır.
- Bölüm 3'de, yapay zeka, makine öğrenmesi ve derin öğrenme kavramsal tanımları ve yapay sinir ağı mimarisi tanıtılacaktır. Tezde problem çözümlerinde uygulanan PINN' in temel teorik yönleri sunulacaktır.
- Bölüm 4' te, kesirli difüzyon denklemi tanıtılarak ve PINN yöntemi uygulanarak elde edilen sayısal çözümleri, hata analizi grafikler ile birlikte anlatılacaktır.
- Bölüm 5' te, ele alınan diferansiyel denklem sistemleri tanıtılarak ve PINN yöntemi uygulanarak elde edilen sayısal çözümleri, hata analizi grafikler ile birlikte anlatılacaktır.
- Bölüm 6' da, bu çalışmanın katkıları ve sonucu özetlenecektir. Ayrıca gelecekteki araştırmalar için yön gösterilecek çalışma alanlarından bahsedilecektir.

BÖLÜM 2

KAYNAK ÖZETİ

2.1 Kısmi Diferansiyel Denklemlerin Sayısal Çözümlerine Bakış

Kısmi diferansiyel denklemlerin (KDD'ler) incelenmesi 18. yüzyılda Euler, D'Alembert, Lagrange ve Laplace'ın çalışmalarıyla, sürekli mekaniğin tanımlanmasında merkezi bir araç olarak ve daha genel olarak, fizik bilimindeki modellerin analitik çalışmalarında başlamıştır. KDD'ler 19. yüzyılın ortalarından itibaren, özellikle Riemann'ın çalışmalarıyla birlikte, matematiğin diğer dallarında da önemli bir araç haline gelmiştir [9].

Fiziksel KDD'ler sırasıyla dalga, ısı ve sıvı hareketi problemlerini modelleyen dalga denklemleri, difüzyon denklemleri ve konveksiyon denklemleri olarak kategorize edilebilir [18], [38]. Yaygın olarak bilinen bazı KDD'ler, kısmi diferansiyel denklemi, zamana bağlı difüzyon denklemi ve ısı denkleminin yanı sıra kütle korunumu yasasına dayanan Navier-Stokes denklemini içermektedir [48].

Kısmi diferansiyel denklemlerin (KDD'ler) çözümüne yönelik sayısal yöntemlerin uygulanması farklı yaklaşımlar altında gerçekleştiği bilinmektedir. Sonlu farklar [75], sonlu elemanlar yöntemi [43], sınır elemanlar yöntemi [78], spektral yöntemler olarak adlandırılan tüm bu sayısal teknikler arasındadır. Bu yöntemlerden bazılarıyla diferansiyel denklemlerin fark denklemleri veya cebirsel denklemler olarak ifade edilebilen klasik Taylor serisi açılımına dayanmaktadır. Burada belirtilen yöntemlerin her birinde, sürekli denklemler, yaklaşık çözümler elde etmek için ayrık forma dönüştürülmektedir. Bunların sayısal çözümünde geleneksel yaklaşımların bir takım sınırlamaları olduğu görülmektedir. Bu yöntemler zaman açısından hassastır ve genellikle ayrıklaştırma yöntemi tarafından düzenlenen doğruluk sırasına göre doğru ve güven-

nilir sonuçların elde edilmesi kolay değildir.

Herhangi bir sayısal yöntem için nihai amaç, problemin gerçek çözümüne yeterince yaklaşımdır. Genel olarak KDD'lerin doğal karmaşıklığı, mevcut birçok çözümü tekniğinin veya yazılımın en verimli veya doğru olmayabileceği büyük bir zorluk teşkil etmektedir.

Kısmi diferansiyel denklemleri (KDD'ler) çözmek için klasik sayısal yöntemlerin doğrudan uygulanmasından kaynaklanan zorlukları hafifletmek için alternatif bir yaklaşım, makine öğrenimi (MÖ) tekniklerinin uygulanmaya başlandığını görülmektedir. Çeşitli makine öğrenimi yöntemleri, kısmi diferansiyel denklemlere dayalı temel matematiksel modelden çok az müdahale ile veya hiç müdahale etmeden tamamen veriye dayalı bir felsefe kullanılmaktadır.

2.2 Makine Öğrenimin Kısmi Diferansiyel Denklemlerin Sayısal Çözümlerinde Etkisi

KDD'lerin zengin yapısı ve KDD'leri doğru bir şekilde çözmek için gereken çözüm uzayı hakkındaki kalite ve ön bilgi göz önüne alındığında, sinir ağı mimarisi, yapay zeka ve makine öğreniminden yararlanmak doğal görünmektedir [58].

Son zamanlarda, devam eden makine öğrenimi devrimi, hem simülasyonlardan hem de deneylerden üretilen yüksek kaliteli verilerin artan zenginliğine dayanarak KDD'leri çözmek için tamamen yeni bir yaklaşım sunmaktadır. Makine öğrenimi KDD'leri incelemek için yaklaşık otuz yıldır uygulanıyor olsa da, hesaplama yetenekleri ve algoritmalarındaki bazı önemli gelişmeler, son on yılda bu çabaları önemli ölçüde ortaya koymaktadır. Fiziki makine öğrenimine dahil etmenin önemi de son yıllarda giderek daha açık hale gelmiştir [7, 8, 10, 11, 21].

KDD'leri veriden öğrenmek, karmaşık bir sistemin davranışını "taklit etmek" için derin öğrenmeyi kullanan alternatif yaklaşımlara göre birçok avantaja sahiptir [44]. Birincisi, KDD'ler doğrudan geometriye, korunum yasalarına, simetrilere, kısıtlamalara vb. bağlanabilmeleri anlamında doğası gereği yorumlanabilir. İkincisi, KDD'ler sınır koşullarını ve parametreleri değiştirerek çok çeşitli fenomenler ve ayrıklaştırmalar ortaya çıkabileceği için oldukça genelleştirilebilir. Geleneksel sayısal yöntemlere benzer şekilde, makine öğrenimi diferansiyel denklemlerin çözümlerine yüksek doğrulukla

yaklaşabilmektedir. Bilinen örnekler arasında derin Ritz yöntemi [30], derin Galerkin yöntemi [71], nöral adi diferansiyel denklemler [15] yer almaktadır.

2.3 Yapay Sinir Ağlarının Kısmı Diferansiyel Denklemlerin Sayısal Çözümlerine Etkisi

Yapay sinir ağlarının (YSA) kullanımının, çok sayıda mühendislik problemi için oldukça etkili bir çözüm olduğu bilinmektedir. Örüntü tanıma, tahmin etme, resim sınıflandırma, finansal tahmin ve veri sıkıştırma gibi mühendislik problemlerinin yanı sıra matematik problemlerinin çözümünde de yapay sinir ağları kullanılmaktadır. Literatürde lineer adi diferansiyel denklemlerin çözümü için ileri beslemeli ağlar kullanılmıştır [54]. YSA'da diferansiyel denklemin çözümüne yaklaşmak için toplam karesel hata miktarını öznel hataya dayalı bir yaklaşım üretmiştir [32]. Riccati diferansiyel denkleminin yaklaşık çözümü için hibrit metot önerilmiştir [61]. Diferansiyel denklemlerin çözümünde çok katmanlı algılayıcı ve radyal temelli fonksiyon kullanıldığı çalışmaların tanıtıldığı geniş bir araştırma önermiştir [45]. Hornik [40] ve Cybenko [22], çok katmanlı ileri beslemeli sinir ağlarının evrensel fonksiyon yaklaşımçıları olduğunu kanıtlayarak, yeterli büyüklükte ve yeterli eğitime sahip, ısı transferi olaylarını yöneten diferansiyel denklemler de dahil olmak üzere diferansiyel denklemlerin çözümlerini doğru bir şekilde yakınsayabileceğini dolaylı olarak kanıtlandığı çalışmalarda görülmektedir.

Yapay sinir ağları, adi ve kısmi diferansiyel denklemlerin sayısal çözümünde de çok sayıda uygulama alanı bulmuştur [62, 29, 49]. YSA'lar, eldeki sistemin fiziksel özellikleri hakkında herhangi bir ön varsayım olmaksızın yüksek boyutlu bir uzayda çözümü öğrenebildikleri için karmaşık bağımlılıkları öğrenmek için özellikle uygundur [77]. Yapay sinir ağları, derin öğrenme teknikleri, özellikle geleneksel yöntemlerin zorlandığı durumlarda, KDD'leri daha verimli ve esnek bir şekilde çözmek için bir alternatif yöntem olmuştur.

2.3.1 Fizikle Öğrenen Sinir Ağları(PINN)

PINN metodunun tanımı, teknik bilgisi detaylı olacak şekilde bir sonraki bölümde verilecek olup, bu bölümde PINN' in tarihsel oluşumundan bahsedilecektir.

Raissi ve arkadaşları 2019 yılında kısmi diferansiyel denklemleri çözmeye yönelik bir sinir ağı yaklaşımı olarak tanıtmışlardır;[66]; ancak Psychogios ve arkadaşlarının [64] ve Lagaris ve arkadaşlarının [49] çalışmaları, PINN çerçevesinin geliştirilmesi için önemli adımlar atılmış olup ve teorik temelini çoğu bu çalışmalarda sağlamıştır. Raissi ve arkadaşlarından önceki çalışmaları özetlersek; Psychogios ve arkadaşları, bir sinir ağı ve ön bilgi içeren bir ilk prensipler fizik modelinden oluşan “hibrit” süreç modellerinin, yalnızca bir sinir ağından oluşan süreç modellerinin doğruluğunu önemli ölçüde artırabileceğini ve eğitim verisi gereksinimlerini azaltabileceğini göstermiştir. Birkaç yıl sonra, Lagaris ve arkadaşları [47], veriye dayalı eğitim ihtiyacını ortadan kaldıran diferansiyel denklemlerin çözümüne yaklaşmak, sinir ağlarını eğitmek için yeni bir yöntem önermişlerdir. Bu tekniğin, geleneksel şemalara kıyasla sayısal basitlik de dahil olmak üzere çeşitli avantajlar sunmaya başlamıştır. Performans açısından (doğruluk ve eğitim hesaplama süresi açısından) mükemmel olmamasına rağmen, PINN'ler kısmi diferansiyel denklemler (KDD'ler) gibi geleneksel yöntemler için zor olan zorlukları ele almak için bir alternatif sunmaktadır.

PINN tekniği, kollokasyon noktaları olarak bilinen önceden tanımlanmış bir dizi veri noktasında denklemin kalıntısını minimize etmesiyle ayırt edilir. Bu noktalarda, tahmin edilen çözümün diferansiyel denklemini sağlaması gerekmektedir. Bunu başarmak için, kalıntı ile ilişkili fizik tabanlı bir kayıp fonksiyonu formüle edilir ve kullanılmaktadır.

BÖLÜM 3

METODOLOJİ

Bu bölümde, fizikle öğrenen sinir ağları (PINN) metodolojisi tartışılacaktır. İlk olarak bu metodolojinin temelini oluşturan tanımlar verilecektir. Yapay zeka, yapay zeka alt metodlarından makine öğrenimi ve derin öğrenme konularından bahsedilerek yapay sinir ağlarına geçilecektir. Daha sonra, PINN metodu anlatılacaktır.

3.1 Yapay Zeka

Yapay zekanın en kapsamlı tanımı, insanlarda gözlemlenen bilişsel süreçleri taklit ederek veri kullanımı yoluyla yeteneklerini geliştirme kapasitesi sergileyen sistemleri veya makineleri kapsar. Diğer bir deyişle, yapay zeka, bilgisayarların şu anda insanlar tarafından gerçekleştirilen görevleri yerine getirmek üzere nasıl programlanabileceğinin incelenmesiyle ilgilenen bir alandır.

Yapay zeka (YZ) terimi ilk kez 1956 yılında John McCarthy tarafından kullanılmış ve McCarthy sinir ağları üzerine yaptığı çalışmalarla bu alanın yönelimini sağlamlaştırmıştır [27]. McCarthy'nin ortaya koyduğu gibi, akıllı makinelerin ve özellikle de akıllı bilgisayar programlarının geliştirilmesiyle ilgilenen mühendislik disiplini. Yapay zekanın geliştirilmesi, insan beyninin işleyişinin, problem çözmenin temelini oluşturan öğrenme ve karar verme süreçlerinin kapsamlı bir şekilde anlaşılmasına ve bu anlayışların akıllı yazılım ve sistemler oluşturmak için kullanılmasına dayanmaktadır [76].

Matematik, YZ'de her zaman çok önemli ve vazgeçilmez bir rol oynamaktadır. On yıllar önce, K-en yakın komşu [20], destek vektör makinesi [19] ve AdaBoost [34] gibi YZ ile ilgili pek çok klasik yaklaşım, matematiksel formülasyonları oluşturulduktan sonra önerilmiş ve geliştirilmiştir. Son yıllarda, derin öğrenmenin hızla gelişmesiyle

birlikte YZ, matematik camiasında giderek daha fazla ilgi görmeye başlamıştır [53]. Markov süreci, minimax optimizasyonu, Bayesian istatistikleri ve Bayesian öğrenmesi ile donatılmış birçok YZ uygulamasında en uygun araçlar haline geldi [33]. Sonraki bölümde yapay zekanın birer alt alanı olan makine öğrenimi ve derin öğrenme yöntemlerinden bahsedilecektir.

3.2 Makine Öğrenimi

Makine öğrenimi, bir veri setine dayalı olarak akıllı tahminler gerçekleştiren geniş bir algoritma yelpazesini ifade eden bir terimdir [50]. Yapay zekanın bir alt dalı olan makine öğreniminin amacı, insanların öğrenme kapasitesini taklit etmektir. Bu, bir modelin öngörülemeyen ve belirtilmemiş koşullara uyum sağlamasını gerektirir. Bu amaca ulaşmak için makine öğreniminde, büyük veri setlerini modelleyecek algoritmalar kullanılmaktadır.

Makine öğrenimi, ham verilerden bilgi elde etmek ve bunları yapılandırılmış bir biçimde temsil etmek için algoritmalar kullanılmaktadır. Model daha sonra henüz modellemeye tabi tutulmamış veriler hakkında bilgi çıkarmak için uygulanmaktadır.

3.2.1 Makine Öğrenimi Türleri

- **Denetimli Makine Öğrenimi:** Denetimli makine öğrenimi, sinir ağının etiketli veri kümelerine dayalı olarak tahminler yapmayı veya verileri sınıflandırmayı öğrendiği makine öğrenimi tekniğidir. Burada hedef değişkenlerle birlikte her iki giriş özelliğini de kullanır. Sinir ağı, tahmin edilen ve gerçek hedef arasındaki farktan kaynaklanan hataya dayalı tahminler yapmayı öğrenir, bu süreç geriye yayılma olarak bilinir. Derin öğrenme algoritmaları, görüntü sınıflandırmaları ve tanıma, duygu analizi, dil çevirileri vb. gibi birçok denetimli görev için kullanılmaktadır.
- **Denetimsiz Makine Öğrenimi:** Denetimsiz makine öğrenimi, sinir ağının etiketsiz veri kümelerine dayalı olarak örüntüleri keşfetmeyi veya veri kümesini kümelemeyi öğrendiği makine öğrenimi tekniğidir. Burada hedef değişkenler yoktur. Makine, veri kümeleri içindeki gizli kalıpları veya ilişkileri kendi kendine

belirlemek zorundadır. Oto kodlayıcılar ve üretken modeller gibi derin öğrenme algoritmaları kümeleme, boyut azaltma ve anomali tespiti gibi denetimsiz görevler için kullanılmaktadır.

- **Pekiştirmeli Makine Öğrenimi:** Kendi kendini denetleyen öğrenmenin bir uzantısı da pekiştirmeli öğrenmedir. Her iki algoritma aynı yaklaşımı kullanarak tanımlanmamış etiketsiz verilerle öğrenme işlemini gerçekleştirir. Ancak pekiştirmeli öğrenme modele bir geri bildirim döngüsü, bir nevi hafıza görevi gören anahtar döngü ekler [59]. Görevini doğru bir şekilde yerine getiren pekiştirmeli öğrenme, girdi ve çıktı değerlerini ilişkilendirirken modeli güçlendiren olumlu bir geri bildirim olarak modeli ödüllendirir. Aynı şekilde yanlış çözümler için olumsuz geri bildirim olarak modeli cezalandırır. Bu açılardan sistem bir nevi hafıza gibi davranan, ödül –ceza sistemine dayalı bir şekilde çalışmaktadır [57].

Matematiksel bakış açısıyla, makine öğrenimi genellikle belirli fonksiyonlara yaklaşmak için sinir ağları ile doğrusal olmayan modeller oluşturmaktadır. Ayrıca bir sonraki bölümde anlatılacak olan derin öğrenme geleneksel makine öğrenimi algoritmalarına göre en önemli avantajlarından biri, otomatik özellikler çıkarma yeteneğine sahiptir [1].

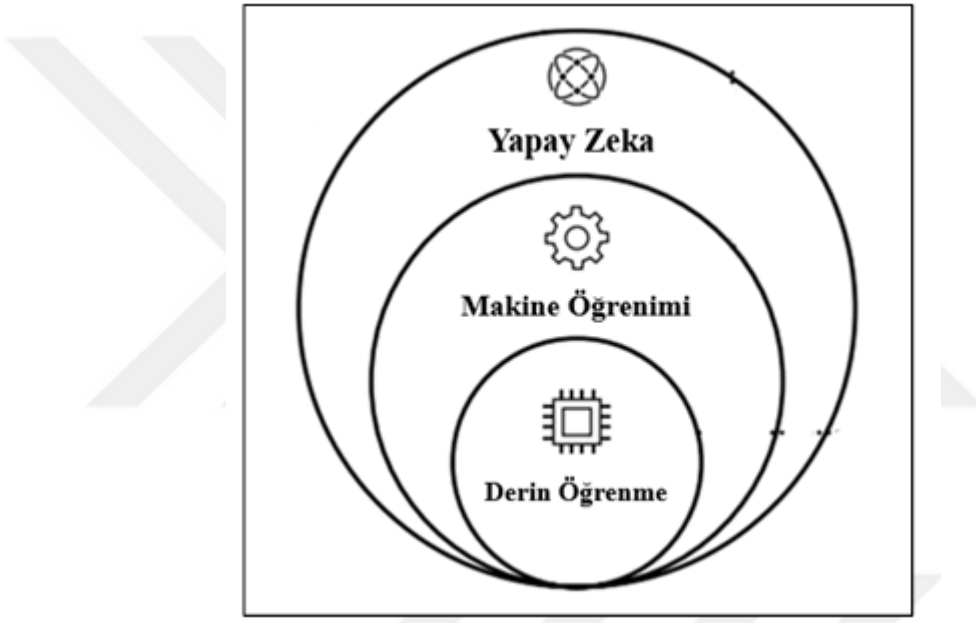
3.3 Derin Öğrenme

Derin öğrenme, yapay sinir ağı mimarisine dayanan bir makine öğrenimi dalıdır. Bir yapay sinir ağı, girdi verilerini işlemek ve bunlardan bir şeyler öğrenmek için birlikte çalışan nöron adı verilen birbirine bağlı düğüm katmanlarını kullanmaktadır. Tam bağlantılı derin sinir ağı diye ifade edilen sinir ağları ise, bir giriş katmanı ve birbiri ardına bağlanmış bir veya daha fazla gizli katmanlara sahiptir. Her nöron bir önceki katman nöronlarından veya giriş katmanından girdi almaktadır. Bir nöronun çıktısı, ağın bir sonraki katmanındaki diğer nöronlara girdi olur ve bu işlem son katman ağın çıktısını üretene kadar devam eder. Sinir ağının katmanları, girdi verilerini bir dizi doğrusal olmayan dönüşüm yoluyla dönüştürerek ağın girdi verilerinin karmaşık temsillerini öğrenmesini sağlamaktadır. Günümüzde derin öğrenme yapay zekası, bilgisayarla görme, doğal dil işleme ve takviyeli öğrenme gibi çeşitli uygulamalardaki başarısı nedeniyle makine öğreniminin en popüler ve görünür alanlarından biri haline

gelmektedir.

Özetleyecek olursak; YZ, “insan” zekasını taklit etmeyi amaçlayan bilgisayar sistemlerinin teorisi ve geliştirilmesi olarak tanımlanabilmektedir. Makine öğrenimi, bilgisayar sistemlerine “kendi kendine öğrenme” yeteneği kazandırmak için istatistiksel teknikler kullanan bir YZ alt disiplini; yani makine öğrenimi, algoritmaların tahminlerini iyileştirmek için modellerini uyarladıkları yerdir, böylece belirli bir görev için performansta aşamalı bir iyileşme olmaktadır.

Derin yapılandırılmış öğrenme veya hiyerarşik öğrenme olarak da bilinen derin öğrenme, YSA’dan önemli ölçüde yararlanan bir makine öğrenimi yöntemleri ailesidir: “derin”, girdi ve çıktı arasındaki katman veya yineleme sayısını ifade eder. Şekil 3.1 de yapay zeka, makine öğrenimi ve derin öğrenme arasındaki ilişkinin görseli yer almaktadır.

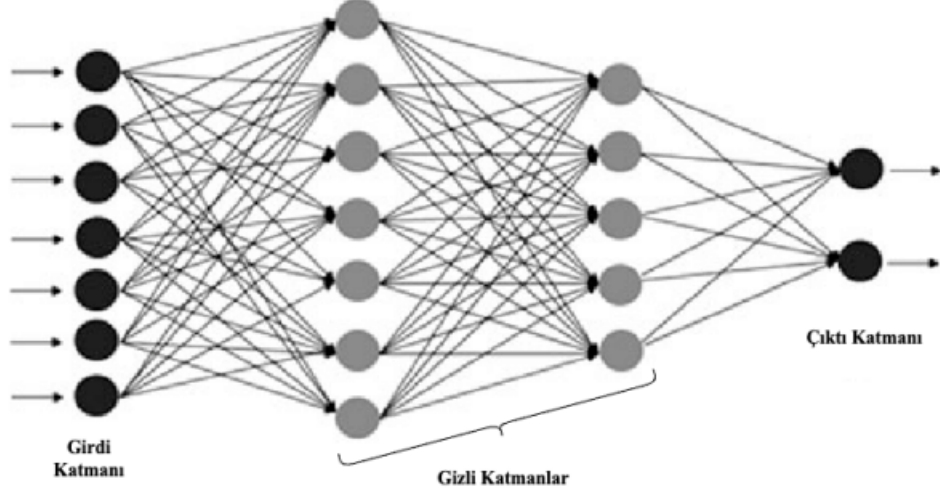


Şekil 3.1: Yapay zeka, makine öğrenimi ve derin öğrenme arasındaki ilişkinin grafiksel özeti

3.4 Yapay Sinir Ağları

Yapay sinir ağları, derin öğrenme alanında kullanılan bir model türüdür. Yapay sinir ağları, birbirine bağlı yapay nöronlardan oluşan bir koleksiyondur. Bu nöronlar diğer

nöronlardan bilgi alır ve aktarır. Diğer bir anlatımla ham verilerden çıkarılan bilgileri birleştirmek için model görevi gören yapısal tanımlamalardır. Bu yapılar veya modeller, bilinmeyen verilerin tahmin edilmesi amacıyla kullanılabilir. Yapay sinir ağı şeması Şekil 3.2’de gösterildiği gibidir. Bir yapay sinir ağı yapısı aşağıdaki gibidir:



Şekil 3.2: Yapay sinir ağı yapısı [6]

1. Girdi/Giriş Katmanı:

- Sistem ham verileri girdi olarak alır.
- Bu katmandaki nöronlar giriş verilerinin tek bir özelliği ile ilişkilidir.

2. Gizli Katmanı:

- Nöronlar girdi verilerinin işlenmesinden sorumludur.
- Modelin birbirini izleyen her katmanı veriler üzerinde bir dönüşüm gerçekleştirir ve böylece değişkenler arasında var olan karmaşık ilişkileri yakalamaktadır.
- Katman ve nöron sayısı, çözülmekte olan problemin karmaşıklığına bağlı olarak değişebilmektedir.

3. Çıktı/Çıkış Katmanı:

- Nihai sonuç üretilir ve bu sonuç bir sınıflandırma etiketi, regresyon değeri veya karar şeklinde olabilir.

4. Bağlantılar:

- Nöronlar arasında iletilen sinyallerin gücü, aralarındaki bağlantılara atanan ağırlıklara bağlıdır.
- Her nöron ayrıca çıkışı ayarlamak için bir sapma (bias) değerine sahiptir.

Bir YSA, tipik olarak nöronlar veya düğümler olarak adlandırılan ve katmanlar halinde organize edilen birbirine bağlı birimlerden oluşmaktadır. İki den fazla katmana, çok sayıda parametre ve katmanlar arası bağlantılara ağırlık değeri denmektedir. DÖ sürecinde, YSA'ya uygun veri işlemeyi sağlaması için ağırlık ve sapma (bias) değeri yerleştirilmektedir. Her birim, önceki birimlerden karşılık gelen bir w_i ağırlığı ile bilgi almaktadır. Bu birimlerden gelen tüm bilgiler toplanır. Daha sonra bu toplama bir sapma değeri eklemektedir. Öğrenme, genelde büyük verinin incelenmesi sonucu gerçekleşmektedir. Süreçte temel algoritmalar değişmez, ancak kodun belirli bir cevabı seçmek için kullanılan ağırlık ve sapma değeri değişir. Son olarak, bir fonksiyon bu toplamı alır ve bir y çıktısı sağlamaktadır:

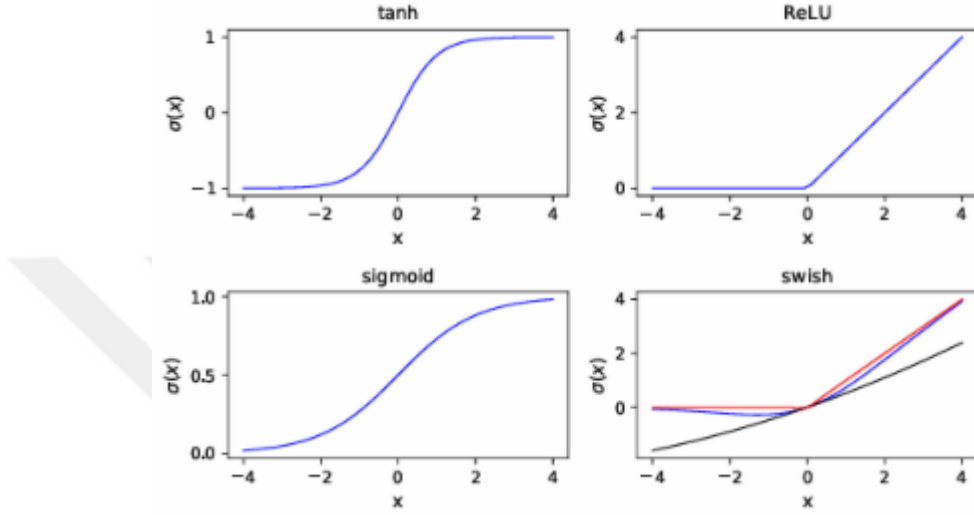
$$y = \sigma\left(\sum_{i=1}^N w_i x_i + b\right) \quad (3.1)$$

Burada σ aktivasyon fonksiyonu ve N önceki bilgi birimlerinin sayısıdır.

Aktivasyon Fonksiyonu: Bir YSA'da aktivasyon fonksiyonu biyolojik sinir hücre-sini taklit eden anahtar yapılarıdır. Bir katmanın nöronlarının (düğümlerinin) çıktı değerlerini bir sonraki katmana yaymak için aktivasyon fonksiyonları kullanılmaktadır. Aktivasyon fonksiyonları, nöronun aktivasyonunu sağlayan skallerden skalere bir fonksiyondur. YSA'nın modelleme yeteneklerine eğrisellik katmak için de aktivasyon fonksiyonları kullanılmaktadır [74]. Doğrusal olmayan bir aktivasyon fonksiyonu, sinir ağının karmaşıklığını artırmanın anahtarıdır. Bazı yaygın aktivasyon fonksiyonları arasında düzeltilmiş doğrusal birim (ReLU) [35], hiperbolik tanjant, sigmoid [23] ve swish [69] bulunmaktadır. Bu fonksiyonların matematiksel ifadeleri ve grafikleri aşağıda verilmektedir.

	Fonksiyon	Türevi
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
ReLU	$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
Swish	$swish_\beta(x) = swish_1(x) = \frac{x}{2} * \tanh(\frac{x}{2}) + \frac{x}{2}$	$swish'_1(x) = \frac{x + \sinh x}{4 \cosh^2 \frac{x}{2}} + \frac{1}{2}$

Tablo 3.1: Aktivasyon fonksiyonların matematiksel gösterimi



Şekil 3.3: Yaygın olarak kullanılan aktivasyon fonksiyonlarının grafikleri

Kayıp Fonksiyonları: Amaç, belirli bir yapay sinir ağının (YSA) başlangıçta programlandığı eğitim verilerine yakınlığını tespit etmektir. YSA'nın tahminlerinde gözlemlenen hatadan bir metrik türetilir. Bu hataların daha sonra tüm veri kümesi üzerinden ortalaması alınır ve sonuçta YSA'nın ideale yakınlığını temsil eden tek bir sayı elde edilir. Kayıp fonksiyonları yardımıyla eğitilen YSA her adımda optimize edilir.

3.4.1 Yapay Sinir Ağı Mimarisi

YSA mimarisini $A.x = b$ modeli bağlamında incelersek, A matrisi girdi verileri ve b sütun vektörü A matrisindeki her satırın çıktı değerleridir. YSA'daki ağırlık değerleri, x (parametre vektörü) değerleridir. YSA mimarisinde nöron sayısı, katman sayısı, katmanlar arasındaki bağlantı türleri bilgilerine göre tanımlanır [26]. Bir yapay sinir

ağının gelişimi sürecindeki yapılar şunlardır:

Algılayıcı (Perseptron): Aktivasyon fonksiyonu olarak adım fonksiyonunu kullanan yapay bir nörona yani düğüme “algılayıcı (perseptron)” denir. Girdi değerleriyle çarpılan ve daha sonra düğümden toplanan ağırlık değerleri reel sayılardır.

Genellikle tam bağlı ağ veya çok katmanlı perceptron olarak adlandırılan en basit sinir ağı, birbirine bağlı birden fazla yapay birimden oluşur. Bu tür bir ağ, bir giriş ve bir çıkış katmanından oluşur ve aralarında, belirli derinlik ve genişliğe ve doğrusal olmayan bir aktivasyon fonksiyonuna sahip gizli katmanlar vardır.

Çok katmanlı perceptrondaki gizli katmanlar bir önceki katmandan gelen girdileri alır. Her girdi belirli bir ağırlıkla çarpılır. Gizli katmanlardaki düğümler ve çıktı katmanı da bir sapma değerine sahiptir. Daha önce de belirtildiği gibi, gizli katmanlar doğrusal olmayan bir aktivasyon fonksiyonuna sahiptir. Genel olarak, L katmanlı bir ağ şu şekilde yazılabilir:

$$\mathcal{N}^0(x) = x \quad (3.2)$$

$$\mathcal{N}^i(x) = \sigma(W^i \mathcal{N}^{i-1}(x) + b^i) \quad (3.3)$$

$$\mathcal{N}^L(x) = \sigma(W^L \mathcal{N}^{L-1}(x) + b^L) \quad (3.4)$$

Burada W ve b sırasıyla ağırlık matrisi ve bias vektörüdür. $(m \times n)$ matrisidir; burada n giriş vektörünün, m ise çıkış vektörünün uzunluğudur. b bir $(m \times 1)$ vektördür. Bu çarpım $(m \times 1)$ boyutunda bir y çıktı vektörü üretir. Matris formunda, katmanlar arasındaki hesaplama şu şekilde yazılabilir:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \sigma \left(\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \right) \quad (3.5)$$

Ağırlıklar ve sapma değerleri, θ ile etiketlenen hiper parametreler olarak adlandırılan eğitilebilir parametrelerdir. Sinir ağlarının öğrenme prosedürü, belirli bir kayıp fonksiyonunu en aza indiren optimize edilmiş bir hiper parametre seti bulmaktır. En yaygın

kayıp fonksiyonlarından biri, tahmin edilen ve gerçek değer arasındaki farkın karesinin ortalamasını ölçen ortalama karesel hatadır (OHK). Şu şekilde ifade edilebilir:

$$\mathcal{L}_{OHK} = \frac{1}{N_d} \sum_{i=1}^{N_d} |\hat{y}_i - y_i|^2 \quad (3.6)$$

Burada $\hat{y}_i$ sinir ağının çıktısı ve y_i hedef değerdir. Herhangi bir öğrenme prosedürü başlamadan önce hiper parametrelerin başlatılması gerekir. Rastgele başlatma ile öğrenme prosedürü derin sinir ağlarında kötü performans göstermektedir [51]. Buradaki temel amaç kesin değeri tahmin etmek olduğundan, kaybın en aza indirilmesi gerekir. Bu optimizasyon problemi şu şekilde gösterilebilir:

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{J}(\theta; x), \quad (3.7)$$

Burada $\mathcal{J}$, Denklem 3.7 'de amaç fonksiyonudur. (θ) öğrenilmekte olan model parametreleridir. Hiperparametreler, bu minimizasyon probleminin çözümlerine yaklaştırılarak ardışık iterasyonlarda güncellenebilir. Gradyan inişi güncelleme kuralı şu şekildedir:

$$\theta = \theta - \alpha \nabla \mathcal{J}(\theta), \quad (3.8)$$

Burada α öğrenme oranıdır ve yerel bir minimuma ulaşmak için negatif gradyan yönünde atılacak adımın boyutunu belirleyen bir parametredir [52]. Öğrenme oranının seçimi, gradyan inişinin davranışını ve yolunu etkiler [54]. Öğrenme oranı uygulaması için yaygın olarak kullanılan teknik öğrenme oranı azalmasıdır. Bu yaklaşım, ilk iterasyonlarda (epoklarda) büyük bir öğrenme oranıyla başlar, ardından öğrenme oranını formla kademeli olarak azaltır. $\nabla \mathcal{J}(\theta)$ ise kayıp fonksiyonunun gradyanıdır. Burada güncelleme stokastik gradyan inişi algoritması kullanılarak gerçekleştirilebilir [52].

Gradyan inişi algoritması, çok boyutlu bir uzayda, bir fonksiyonun eğimi (gradyanı) boyunca hareket ederek minimum noktayı arar. Bu süreçte algoritma, fonksiyonun eğimine göre adım adım ilerler. $\nabla_{\theta} \mathcal{J}(\theta)$ kayıp fonksiyonunun gradyanının tersi yönünde yeni θ parametreleri ekleyerek $\mathcal{J}(\theta)$ amaç fonksiyonunu en aza indirir. Bu kayıp fonksiyonunun hiperparametrelere göre gradyanı geriye yayılım ile hesaplanır [53]. İlk

olarak mevcut ağırlıkları ve sapma değerlerini kullanarak kaybı hesaplar (ileri geçiş), daha sonra ağda geriye doğru hareket ederken zincir kuralını kullanarak kaybın gradyanını bulunmaktadır.

Bu çalışmada, uyarlanabilir moment tahmini olarak adlandırılan en yaygın optimize edicilerden biri olan Adam kullanılmaktadır. Stokastik kayıp fonksiyonlarının birinci dereceden gradyan tabanlı optimizasyonu için bir algoritmadır. TensorFlow ve PyTorch gibi en popüler makine öğrenimi çerçeveleri bu yöntemi içerir.

3.5 Fizikle Öğrenen Sinir Ağları (PINN)

Fizikle öğrenen sinir ağları (PINN) belirli bir veri kümesini yöneten herhangi bir fiziksel yasanın bilgisini öğrenme sürecine dahil edebilen ve kısmi diferansiyel denklemlerle (KDD'ler) tanımlanabilen bir tür evrensel fonksiyon yaklaşımıdır [67]. Bazı biyolojik ve mühendislik problemleri için düşük veri kullanılabilirliği, bu uygulamalar için kullanılan geleneksel makine öğrenimi modellerinin sağlamlığını sınırlamaktadır. Genel fiziksel yasaların ön bilgisi, sinir ağlarının eğitiminde, kabul edilebilir çözümlerin uzayını sınırlayan ve fonksiyon yaklaşımının genelleştirilebilirliğini artıran bir düzenleme göreviyle hareket etmektedir. Bu şekilde, bu ön bilginin bir sinir ağına gömülmesi, mevcut verilerin bilgi içeriğini artırarak öğrenme algoritmasının doğru çözümü yakalamasını ve düşük miktarda eğitim örneğiyle bile iyi genelleme yapmasını kolaylaştırmaktadır.

Bir sistemin dinamiğini yöneten fiziksel yasaların çoğu kısmi diferansiyel denklemlerle tanımlanabilmektedir. Örneğin, Navier-Stokes denklemleri akışkanlar mekaniğini yöneten korunum yasalarından (yani kütle, momentum ve enerjinin korunumu) türetilen bir dizi kısmi diferansiyel denklemdir [4]. Navier-Stokes denklemlerinin uygun başlangıç ve sınır koşulları ile çözümü, kesin olarak tanımlanmış bir geometride akış dinamiklerinin ölçülmesini sağlamaktadır. Bununla birlikte, bu denklemler tam olarak çözülemez ve bu nedenle sayısal yöntemler kullanılmaktadır (sonlu farklar, sonlu elemanlar ve sonlu hacimler gibi). Bu ortamda, bu yöneten denklemler, önceki varsayımlar, doğrusallaştırma ve yeterli zaman ve uzay ayrıklaştırması hesaba katılarak çözülmektedir.

Son zamanlarda, derin öğrenme kullanarak fiziksel olayların kısmi diferansiyel denk-

lemelerini çözmek, evrensel yaklaşım teoreminden¹ ve sinir ağlarının yüksek ifade gücünden yararlanarak yeni bir bilimsel makine öğrenimi (SciML)² alanı olarak ortaya çıkmıştır. Genel olarak, derin sinir ağları yeterli eğitim verisi sağlandığında herhangi bir yüksek boyutlu fonksiyona yaklaşabilmektedir [2]. Bununla birlikte, bu tür ağlar problemin altında yatan fiziksel özellikleri dikkate almaz ve sağladıkları yaklaşım doğruluğu seviyesi hala büyük ölçüde problem geometrisinin yanı sıra başlangıç ve sınır koşullarının dikkatli özelliklerine bağlıdır. Bu ön bilgi olmadan, çözüm benzersiz değildir ve fiziksel doğruluğunu kaybedebilir. Öte yandan, PINN'ler sinir ağı eğitiminde geçerli fiziksel denklemlerden yararlanmaktadır. Yani, PINN'ler, verilen eğitim verilerinin yanı sıra dayatılan yönetim denklemlerini de karşılayacak şekilde eğitilmek üzere tasarlanmıştır. Bu şekilde, bir sinir ağı, büyük ve eksiksiz olması gerekmeyen eğitim verileriyle yönlendirilebilir [2]. Potansiyel olarak, kısmi diferansiyel denklemlerin doğru bir çözümü, sınır koşulları bilinmeden bulunabilmektedir [3]. Bu nedenle, problemin fiziksel özellikleri hakkında bazı bilgiler ve bir tür eğitim verisi (seyrek ve eksik olsa bile) ile PINN, yüksek doğrulukla optimum bir çözüm bulmak için kullanılabilir.

PINN'ler hesaplamalı bilimlerde çok çeşitli problemlerin ele alınmasına olanak tanımaktadır ve KDD'ler için yeni sayısal çözücü sınıflarının geliştirilmesine öncülük eden bir teknolojiyi temsil etmektedir [66]. PINN'ler geleneksel yaklaşımlara (örn, özellikle, eğitilmiş PINN ağı, yeniden eğitime gerek kalmadan farklı çözünürlüklerdeki simülasyon ızgaralarındaki değerleri tahmin etmek için kullanılabilir) [51]. Ayrıca, kısmi diferansiyel denklemlerde gerekli türevleri hesaplamak için otomatik türevden yararlanmaya izin verirler; bu, sayısal veya sembolik türevden daha üstün olduğu değerlendirilen sinir ağlarını türetmek için yaygın olarak kullanılan yeni bir türev teknikleri sınıfıdır [5].

¹Evrensel Yaklaşım Teoremi, sinir ağları ve derin öğrenme alanında, özellikle yapay sinir ağlarının incelenmesi ve uygulanmasıyla ilgili temel bir sonuçtur. Bu teorem esasen, sonlu sayıda nöron içeren tek bir gizli katmana sahip bir ileri beslemeli sinir ağının, uygun aktivasyon fonksiyonları verildiğinde kompakt alt kümeleri üzerindeki herhangi bir sürekli fonksiyonun yaklaşıklığını hesaplayabildiğini belirtir. Bu sonucun sinir ağlarının tasarımı, yetenekleri ve anlaşılması üzerinde derin etkileri vardır.

²SciML, bilimsel makine öğrenimi paketlerini birleştirmek için oluşturulmuş açık kaynaklı bir yazılım organizasyonudur. Bu, ters problemler ve otomatik model keşfi için metodolojilerin yanı sıra diferansiyel denklem çözücüler gibi modüler bilimsel simülasyon destek yazılımlarının geliştirilmesini içerir.

3.5.1 Modelleme ve Hesaplama

Genel bir doğrusal olmayan kısmi diferansiyel denklemi ele alalım:

$$u_t + \mathcal{N}[u; \lambda] = 0, \mathbf{x} \in \Omega, t \in [0, T] \quad (3.9)$$

Burada $u(t, x)$ çözümü, u_t u' nun t' ye göre türevini, $\mathcal{N}$ λ ile parametrelendirilmiş doğrusal olmayan bir operatörü ve Ω ise R^D 'nin bir alt kümesi olarak $\Omega = [x_{min}, x_{max}] \times [t_{min}, t_{max}]$ göstermektedir.

Yukarıdaki denklemle tanımlanan genel bir dinamik sistemin gürültülü ölçümleri göz önüne alındığında, PINN'ler iki sınıf problemi çözmek için tasarlanabilir:

- veri odaklı çözüm
- kısmi diferansiyel denklemlerin veri odaklı keşfi

Kısmi diferansiyel denklemlerin veriye dayalı çözümü

KDD' in veri odaklı çözümü, sınır verileri ve/veya ölçümler sabit model parametreleri λ verildiğinde sistemin gizli katmanı $u(t, x)$ i hesaplar. Bunun çözümünde:

$$u_t + \mathcal{N}[u] = 0, \mathbf{x} \in \Omega, t \in [0, T] \quad (3.10)$$

$f(t, x)$ rezidüyü aşağıdaki şekilde tanımlayarak,

$$f := u_t + \mathcal{N}[u; \lambda] = 0 \quad (3.11)$$

$u(t, x)$ derin bir sinir ağı tarafından yaklaştırılmaktadır. Bu ağ, otomatik türevlendirme kullanılarak türevleri otomatik alınmaktadır. $u(t, x)$ ve $f(t, x)$ parametreleri daha sonra aşağıdaki kayıp fonksiyonunun minimize edilmesiyle öğrenme süreci başlamaktadır.

PINN'lerdeki kayıp fonksiyonu üç ana bileşenden oluşur:

- **Rezidü Kaybı:** $\mathcal{L}_f$ rezidü kaybı, $\mathcal{N}(x_i, t_i \in \Omega)$ kollokasyon noktası üzerinden

ortalama karesel rezidüler olarak hesaplanmaktadır:

$$\mathcal{L}_f = \frac{1}{N} \sum_{i=1}^N [f(u_i, x_i, t_i)]^2 \quad (3.12)$$

burada $x_{min} < x_i < x_{max}$ ve $t_{min} < t_i < t_{max}$

- **Başlangıç Koşulu Kaybı:** $\Gamma_b = \{(x, t) \in \Omega : x = x_{min} \vee x = x_{max}\}$ başlangıç koşulunu gösterebilir. Başlangıç koşulu kaybı $\mathcal{L}_b$, $\mathcal{N}_b$ başlangıç noktaları $(x_j, t_{min}) \in \Gamma_b$ üzerinden sınır ağ çözümü ile başlangıç koşulu u_b arasındaki ortalama karesel fark olarak hesaplanır:

$$\mathcal{L}_b = \frac{1}{\mathcal{N}_b} \sum_{j=1}^{\mathcal{N}_b} [u_b(x_j) - (u, x_j, t_{min})]^2 \quad (3.13)$$

- **Sınır Koşulu Kaybı:** $\Gamma_s = \{(x, t) \in \Omega : x = x_{min} \vee x = x_{max}\}$ sınır koşulunu gösterebilir. Sınır koşulu kaybı $\mathcal{L}_s$, $\mathcal{N}_s$ başlangıç noktaları $(x_k, t_k) \in \Gamma_s$ üzerinden sınır ağ çözümü ile başlangıç koşulu u_s arasındaki ortalama karesel fark olarak hesaplanır:

$$\mathcal{L}_s = \frac{1}{\mathcal{N}_s} \sum_{K=1}^{\mathcal{N}_s} [u_s(x_k, t_k) - (u, x_k, t_k)]^2 \quad (3.14)$$

Burada x_{min} ve x_{max} sırasıyla x eksenindeki minimum ve maksimum değerlerini temsil etmektedir.

Toplam kayıp fonksiyonu $\mathcal{L}_{tot}$, her bir kayıp fonksiyonu bileşenlerinin toplanmasıyla elde edilmektedir

$\mathcal{L}_{tot}$:

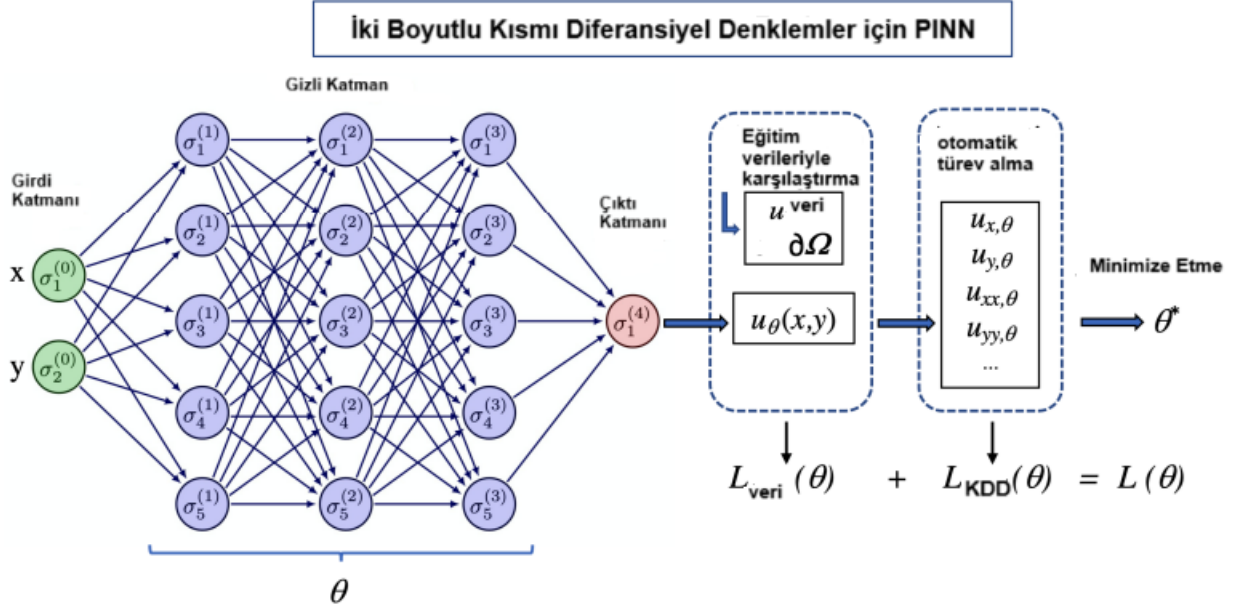
$$\mathcal{L}_{tot} = \mathcal{L}_u + \mathcal{L}_f \quad (3.15)$$

Burada $\mathcal{L}_u = \|u - z\|_{\Gamma}$, PINN $u(t, x)$ başlangıç ile sınır koşulları kümesi verilerinin tanımlandığı Γ noktaları kümesinde ölçülen veriler arasındaki hatadır ve $\mathcal{L}_f = \|f\|_{\Gamma}$ rezidü fonksiyonun ortalama karesel hatasıdır. Bu ikinci terim, PINN' i eğitim süreci sırasında kısmi diferansiyel denklem tarafından ifade edilen yapısal bilgiyi öğrenmeye yönlendirmektedir. Bu yaklaşım, fiziksel süreçlerin tahmini, çoklu fizik ve çok ölçekli modelleme ile simülasyon uygulamalarının hesaplamalı olarak fizikle öğrenen modeller elde etmek için kullanılmaktadır [68]. KDD'nin çözümüne

yakınsadığı belirtilmektedir. [28].

PINN, iterasyonlar boyunca toplam kaybın büyüklüğünü azaltarak kendini optimize eder. Ayrıca, λ öğrenilebilir bir değişken olduğundan, şu ana kadar ele alınan süreç KDD'nin veri odaklı keşfi olarak adlandırılır. Buna karşılık, KDD'nin veri odaklı çözümünde λ sabittir ve eğitilebilir bir bileşen olarak kabul edilmez, bu nedenle Denklem 3.11'de ihmal edebiliriz [67]. Çözüme sinir ağı ile yaklaşımı bir görsel ile göstermek gerekirse;





Şekil 3.4: Üstte yer alan şematik gösterim, Dirichlet benzeri sınır koşullarına sahip iki boyutlu doğrudan bir problemle ilişkili kısmi diferansiyel denklemin çözümü için uygulanan Fizikle Öğrenen Sinir Ağının (PINN) yapısını göstermektedir. Girdi katmanı, iki uzay koordinat değişkeni x ve y 'yi temsil eden iki giriş değişkeni içerir. Ağ, her biri beş nörondan oluşan ve hem giriş hem de çıkış katmanlarına bağlanan üç gizli katmandan oluşmaktadır. İkinci katman, tahmin edilen çözümü $u_\theta(x, y)$ temsil eden tek bir değişkene (bir nöron) sahiptir. Otomatik Türevlendirme, kısmi diferansiyel denklemin kayıp fonksiyonu $L_{KDD}(\theta)$ 'yı oluşturmak için gerekli kısmi türevleri (yani $u_{x,\theta}$, $u_{xy,\theta}$...) değerlendirmek için prosedürde kullanılır. Kayıp fonksiyonu $L_{\text{veri}}(\theta)$, $\partial\Omega$ sınır alanına uygulanan yumuşak kısıtlamalarla (yani eğitim veri seti aracılığıyla) elde edilir.

BÖLÜM 4

KESİRLİ DİFÜZYON DENKLEMLERİN PINN İLE SAYISAL ÇÖZÜMLERİ

Kesirli türev, türev mertebesinin bir kesir olduğu tamsayı mertebeli türevin genelleştirilmesidir. Gerçek dünya problemlerinin çözümündeki geniş uygulamaları nedeniyle, kesirli kısmi diferansiyel denklemler alanında çalışan araştırmacılar için bir ilgi odağı haline gelmiştir. Kesirli difüzyon denklemleri ile ilgilenen konu, tamsayı zaman ve/veya uzay türevlerinin kesirli türevlerle değiştirilmesiyle verilir. Kesirli difüzyon denklemleri uygulamaları arasında parçacık takibi [56], iyon kanalı geçişi [39], popülasyon dinamikleri [6], piyasalardaki opsiyon fiyatları [79] ve elektromanyetik [73] gibi diğer bilim ve mühendislik problemleri yer almaktadır. Literatürde, kesirli türevlerin ortaya çıkışından bu yana kesirli difüzyon denklemlerin ve diğer kesirli modellerin çözümü için farklı yöntemler çalışılmıştır [31, 46, 63]. Sonlu farklar yöntemi, sayısal çözümü daha iyi bir doğrulukla sağlamak için popüler yöntemlerden biridir [36]. Günümüzde, kesirli difüzyon denklemleri için başlangıç ve sınır değer problemleri çeşitli difüzyon sistemlerinde ayrıntılı olarak tartışılmıştır [80, 55, 70, 41].

$1 < \alpha < 2$ mertebeden kesirli difüzyon denkleminin sınır değer problemini ele alalım:

$$\frac{\partial u(x, t)}{\partial t} - d_+(x) \frac{\partial^\alpha u(x, t)}{\partial_- x^\alpha} - d_-(x) \frac{\partial^\alpha u(x, t)}{\partial_+ x^\alpha} = f(x, t), \quad (4.1)$$

$$x_L < x < x_R, \quad 0 < t \leq T,$$

$$u(x_L, t) = 0, \quad u(x_R, t) = 0, \quad 0 \leq t \leq T, \quad (4.2)$$

$$u(x, 0) = u_0(x), \quad x_L \leq x \leq x_R. \quad (4.3)$$

Önceki çalışmalara bakıldığında, fiziksel olarak en anlamlı ve en çok kullanılan kesirli KDD' lerde mertebe $1 < \alpha < 2$ olarak alınmıştır. Biz de problemlerimizde bu şekilde ele aldık. Burada $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ ve $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ türev terimleri PINN metodu kullanımına uygun hale getirebilmek için Grünwald-Letkinov dönüşümleri uygulanır:

$$\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} = \lim_{h \rightarrow 0^+} \frac{1}{h^\alpha} \sum_{k=0}^{\lfloor \frac{(x-x_L)}{h} \rfloor} = g_k^{(\alpha)} u(x - kh, t), \quad (4.4)$$

$$\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} = \lim_{h \rightarrow 0^+} \frac{1}{h^\alpha} \sum_{k=0}^{\lfloor \frac{(x_R-x)}{h} \rfloor} = g_k^{(\alpha)} u(x + kh, t), \quad (4.5)$$

Burada, $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k} \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}$ ile $\binom{\alpha}{k}$ kesirli binom katsayılarını ifade eder. $g_k^{(\alpha)}$ katsayıları aşağıdaki şekilde hesaplanır:

$$g_0^{(\alpha)} = 1, \quad g_k^{(\alpha)} = \left(1 - \frac{\alpha+1}{k}\right) g_{k-1}^{(\alpha)} \quad k \geq 1. \quad (4.6)$$

Örnek 1: Kesirli difüzyon denkleminin başlangıç-sınır değer problemi

$$d_+(x) = \Gamma(1.2)x^{1.8} \quad \text{ve} \quad d_-(x) = \Gamma(1.2)(2-x)^{1.8}.$$

x ekseninde $[x_R, x_L] = [0, 2]$ ve t zaman aralığında $[0, T] = [0, 1]$ şeklindedir. Kaynak fonksiyon ve başlangıç koşulu sırasıyla şu şekilde verilir:

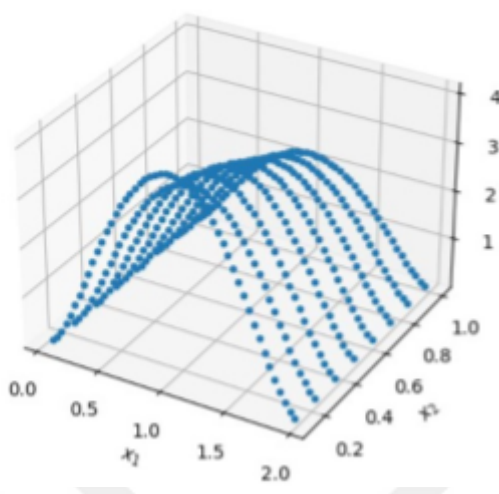
$$f(x, t) = -32e^{-t}[x^2 + (2-x)^2 + 0.125x^2(2-x)^2 - 2.5(x^3 + (2-x)^3) + \frac{25}{22}(x^4 + (2-x)^4)] \quad (4.7)$$

$$u_0(x) = 4x^2(2-x)^2 \quad (4.8)$$

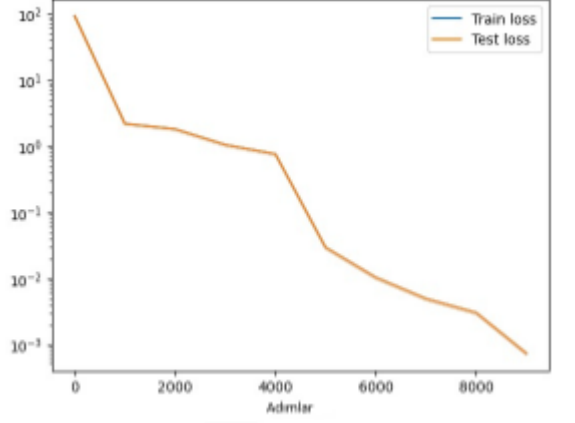
Söz konusu denklemin analitik çözümü aşağıdadır:

$$u(x, t) = 4e^{-t}x^2(2-x)^2 \quad (4.9)$$

Farklı Δx deęerleri ve $\Delta t = \frac{1}{128}$ için hesaplanan mutlak hatalar Tablo ??' de verilmiştir. Ayrıca elde edilen sonuçlar ileri fark ve hızlı ileri fark metodlarından bulunan sonuçlar ile karşılaştırılmıştır ve Tablo 4.1' de gösterilmiştir.



Şekil 4.1: $N = 2^7$ ve $\Delta t = 1/128$ için çözülen Örnek 1 in çözüm grafięi



Şekil 4.2: Örnek 1 için hatanın davranışı

n	PINN	FD[18]	FFD[18]
2^6	8.59×10^{-4}	1.74×10^{-2}	1.55×10^{-2}
2^7	2.98×10^{-4}	8.35×10^{-3}	7.02×10^{-3}
2^8	7.25×10^{-5}	4.08×10^{-3}	3.18×10^{-3}

Tablo 4.1: Maksimum mutlak hatalar ($\Delta t = -\frac{1}{128}$)

Örnek 2: Şimdi ise;

$$d_+(x) = d_-(x) = \frac{\Gamma(2.2)x^{2.8}}{6}. \quad (4.10)$$

olmak üzere başlangıç-sınır koşullu kesirli difüzyon denklemini ele alalım: x uzay ekseninde $[x_R, x_L] = [0, 1]$, $t > 0$ olduğu zaman ekseninde verilen denklemin kaynak fonksiyonu şu şekildedir:

$$f(x, t) = (1 + x)e^{-t}x^3, \quad (4.11)$$

Başlangıç koşulu ise şu şekilde verilir:

$$u(x, 0) = x^3, 0 < x < 1.$$

Dirichlet sınır koşulları şunlardır:

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

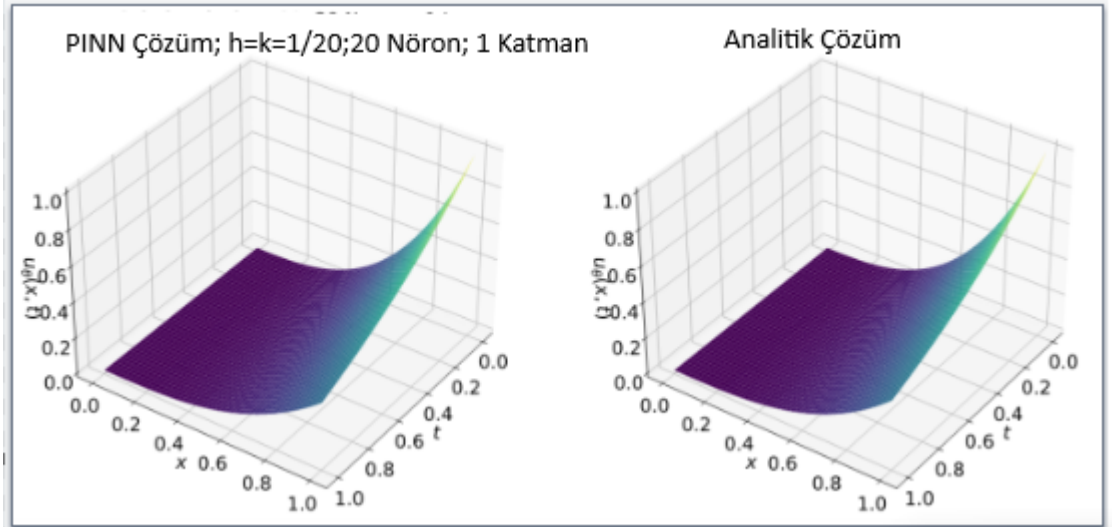
İlgili kesirli difüzyon denkleminin analitik çözümü ise

$$u(x, t) = e^{-t}x^3 \quad (4.12)$$

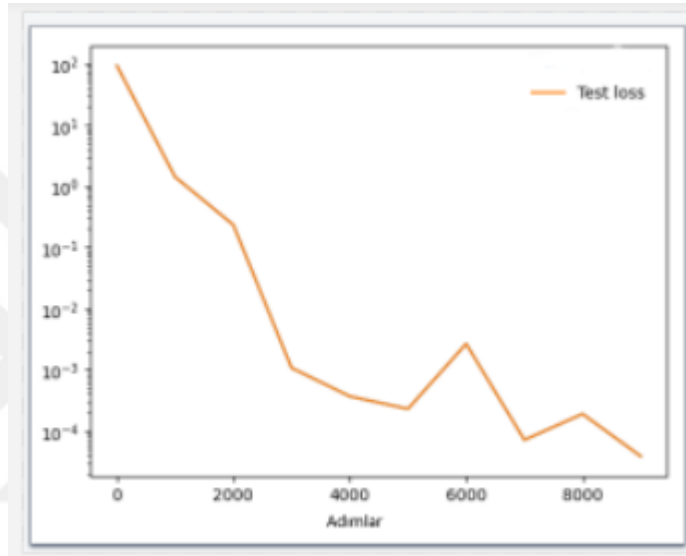
Farklı Δx değerleri ve $\Delta t = \frac{1}{128}$ için hesaplanan mutlak hatalar Tablo 4.2' de verilmiştir. Bu problemin diğer sayısal yöntemlerden Spline [13] ve yerel polinom regresyon yöntemi [72] çalışmadaki çözümlerinden elde edilen sonuçlarla karşılaştırılarak ve bulunan değerler Tablo 4.2' de gösterilmiştir.

n	PINN (N=5, L=1)	PINN (N=20, L=1)	Spline M.[19]	LPR M.[20]
11	0.03687	0.01125	0.1762	0.10446
21	0.01482	0.009588	0.1663	0.10518
61	0.008754	0.005224	0.1307	
121	0.001547	0.0009874	0.0859	

Tablo 4.2: Maksimum mutlak hatalar, $\Delta t = 0.01$



Şekil 4.3: Örnek 2'nin çözüm grafikleri



Şekil 4.4: Örnek 2 için hata davranışı

BÖLÜM 5

DİFERANSİYEL DENKLEM SİSTEMLERİNİN PINN YÖNTEMİ İLE SAYISAL ÇÖZÜMLERİ

Bu bölümde, diferansiyel denklem sistemlerinin sayısal çözümlerinin PINN metodu uygulanarak yapılabildiği gösterilecektir. Uygulanan diferansiyel denklem sistemleri tanıtılacak ve ele alınan problemlerin çözümleri verilecektir.

5.1 Adi Diferansiyel Denklem Sistemi

Fizik, mühendislik, biyoloji ve benzeri alanlardaki en önemli matematiksel model diferansiyel denklemlerdir. Nesnelerin hareketi, sıvı ve ısı akışı, kimyasal reaksiyonlar diferansiyel denklem sistemleri ile modellenir. Ne yazık ki analitik araçlar bu tür sistemlerin çözümü için çoğu zaman yetersiz kalmaktadır. Analitik yollarla analitik çözüm bulunabilen durumların sayısı çok sınırlıdır, analitik çözümlerin her zaman bulunabildiği tek genel sistem sınıfı sabit katsayılı doğrusal sistemlerden oluşmaktadır. Bununla birlikte, uygulamalarda çözümü sonlu sayıda ondalık basamaktan daha fazlasına kadar bulmak gerekli olmamaktadır. Bu nedenle Newton, Taylor ve Euler zamanından beri adi diferansiyel denklemlerin çözümü için sayısal yöntemler geliştirilmiştir. Analitik çözümleri bilinen belirli doğrusal değişken katsayılı veya doğrusal olmayan sistemlerin birçok örneği de vardır, ancak genel olarak bu tür sistemler için ya yaklaşık ya da sayısal bir yönteme başvurmak gerekmektedir [37, 42, 12, 16, 17]. Aşağıdaki doğrusal ikinci mertebeden sınır değer problemleri sistemini ele alıyoruz

[14]:

$$\begin{aligned} u''(x) + xu(x) + xv(x) &= f_1(x) \\ v''(x) + 2xv(x) + 2xu(x) &= f_2(x) \end{aligned} \quad (5.1)$$

Sınır koşulları: $u(0) = u(1) = 0, v(0) = v(1) = 0$.

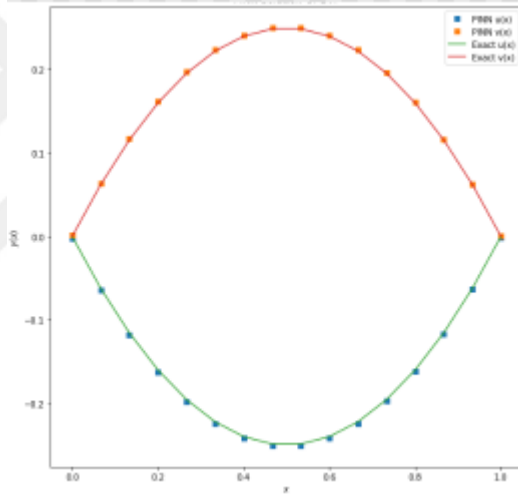
Burada $0 < x < 1, f_1(x) = 2$ ve $f_2(x) = -2$.

Bu sistemin analitik çözümü; $u(x) = x^2 - x$ ve $v(x) = x - x^2$ dir.

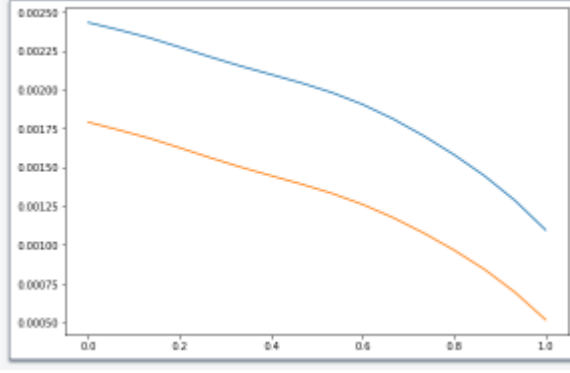
Sayısal sonuçlar Tablo 5.1'de ortaya konmuştur.

Adım Sayısı	$L = 1; N = 15$
5000	$6.034541e^{-05}$
10000	$2.86675e^{-07}$
20000	$8.25724e^{-07}$

Tablo 5.1: Literatür çalışmaları ve farklı parametrelerdeki PINN sonuçlarının karşılaştırılması



Şekil 5.1: PINN ve analitik çözümün karşılaştırılması



Şekil 5.2: PINN modeli ile verilen mutlak hata yüzeyi

5.2 Benney-Roskes/Zakharov-Rubenchik Sistemi

Bu bölümde, öncelikle Benney-Roskes ve Zakharov-Rubenchik denklemlerinin matematiksel ve fiziksel özelliklerini inceleyerek bu denklemlerin tarihsel arka planı ve kullanım alanlarına yönelik genel bir bakış sunulacaktır. Daha sonra, Benney-Roskes/Zakharov-Rubenchik (BR/ZR) sisteminin belirli koşullar altında PINN yöntemi ile sayısal çözümü incelenecektir. Bu çalışmanın önemi, bu sistemin sabit ve değişken katsayılı özellikleriyle birlikte, PINN yöntemi uygulanabilir olmasına yönelik bir perspektif sunarak anlaşılmasını sağlamaktır.

Benney-Roskes denklemi, dalga yayılımı ve suyun yüzeyindeki dalgaların modellenmesi için kullanılan bir parabolik denklem türüdür. Bu denklem, su dalgalarının modellenmesinde oldukça etkilidir ve dalga davranışlarının matematiksel ve fiziksel özelliklerini incelemek için kullanılmaktadır. Lineer olmayan dalga teorisi alanında önemli bir rol oynamaktadır. Benney-Roskes denklemi, su dalgalarını, dalga kırılmasını ve diğer dalga fenomenlerini anlamak için temel bir modelleme aracıdır. Bu denklemin özellikleri ve genel çözümleri, suyun yüzeyindeki dalgaların karmaşık davranışlarının incelenmesinde kullanışlı bir matematiksel araç sunmaktadır.

Zakharov-Rubenchik denklemi, lineer olmayan dalga kırılma modellerinde kullanılan bir matematiksel denklem olarak karşımıza çıkmaktadır. Bu denklem, optik sistemlerin davranışını analiz etmek ve tahmin etmek için kullanılmaktadır. Aynı zamanda bu denklem, plazma fiziği, hidrodinamik ve akustik sistemler gibi farklı fiziksel sistemlerde de uygulama bulmaktadır. Zakharov-Rubenchik denkleminin matematiksel ve fiziksel özellikleri, dalga hareketinin karmaşıklığını ve lineer olmayan etkileri anlamak

için oldukça önemlidir [60].

Benney-Roskes/Zakharov-Rubenchik denklem sistemi ise, kısmi diferansiyel denklemlerle doğrudan ilişkilidir çünkü genellikle uzun dalga bozulmalarını modellemektedir. Bu tür sistemler genellikle denklemlerin çözümlerini analiz etmek için kısmi diferansiyel denklemlerinin kullanılmasını gerektirmektedir. Bu ilişki, sistemin davranışını analiz etmek ve gelecekteki durumlarını öngörmek için kritik bir rol oynar. Kısmi diferansiyel denklemleri kullanarak, Benney-Roskes/Zakharov-Rubenchik sisteminin dalgaların ilerlemesi ve etkileşimi üzerine daha derin bir anlayış sağladığı görülmektedir. Bu ilişki, sistemin matematiksel ve fiziksel özelliklerini incelemek için de mümkün kılmaktadır [25].

$$i\psi_t + \alpha\psi_{xx} + \epsilon_0\alpha\psi_{yy} = (b - \omega m)|\psi|^2\psi - \omega\psi\left\{m\frac{\partial}{\partial t} + (mc - d)\frac{\partial}{\partial y}\right\}\varphi,$$

$$\varphi_{xx} + (1 - mc^2)\varphi_{yy} - m\varphi_{tt} - 2mc\varphi_{yt} = \left\{m\frac{\partial}{\partial t} + (mc - d)\frac{\partial}{\partial y}\right\}|\psi|^2. \quad (5.2)$$

Aşağıdaki dönüşümler uygulandığında,

$$\Psi = e^{i\theta}\psi(v), \varphi = \phi(v), v = v_1x + v_2y + v_0t, \theta = \theta_1x + \theta_2y + \theta_0t$$

burada reel sabit, (5.2) denklemi BR/ZR sistemini doğrusal olmayan adi diferansiyel denklemler sistemine indirgenir.

$$(av_1^2 + \epsilon_0v_2^2)\psi'' + i(2av_1\theta_1 + 2\epsilon_0v_2\theta_2 + v_0)\psi'(a\theta_1^2 + \epsilon_0\theta_2^2 + \theta_0)\psi + (mw - b)\psi^3 + w(mv_0 + mcv_2 - dv_2)\psi\phi' = 0, \quad (5.3)$$

$$2(dv_2 - mv_0 - mcv_2)\psi\psi' + (v_1^2 + v_2^2 - m(v_0 + cv_2)^2)\phi'' = 0 \quad (5.4)$$

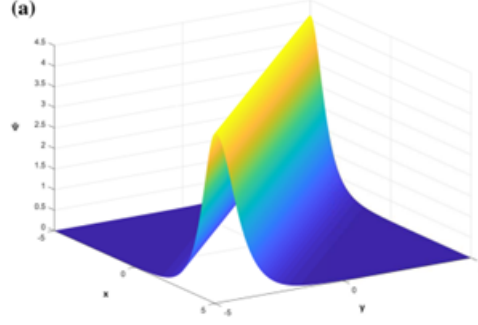
ki bu da,

$$2av_1\theta_1 + 2\epsilon_0v_1\theta_1 = 0. \quad (5.5)$$

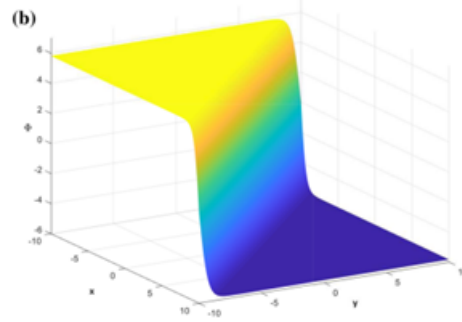
$w = 3, \theta = -3/2$ ve diğer parametrelerin değerleri 1'e eşit olacak şekilde alındığında, (5.3) deki BR/ZR sisteminin tam çözümü aşağıdaki gibi elde edilebilir.

$$\psi(v) = \sqrt{17}\operatorname{sech}\left(\sqrt{\frac{17}{8}}(x + y)\right), \phi(v) = -\sqrt{34}\tanh\left(\sqrt{\frac{17}{8}}(x + y)\right). \quad (5.6)$$

Literatürde (2+1) boyutlu BR/ZR sisteminin kesin çözümü bazı değerler alınarak bulunmuş olup [12] ve (5.3) de yer alan BR/ZR sistemini $w = 3$, $\theta = -3/2$ ve diğer parametreleri 1'e eşit olacak şekilde ele alınmıştır. Bunun sonucunda bu bölümde incelemek istediğimiz (5.7) denklemi ortaya çıkmıştır. PINN yöntemi yardımıyla bu denklem sisteminin sayısal çözümü elde edilmektedir. Önerilen PINN yöntemini çeşitli katmanlar ve iterasyonlar için uygulanmaktadır ve Tablo 5.2'de listelenen hatalar bulunmaktadır. Şekil 5.3 ve Şekil 5.4' de çözümün optimizasyonunu göstermek için hataların grafikleri aşağıda yer almaktadır.



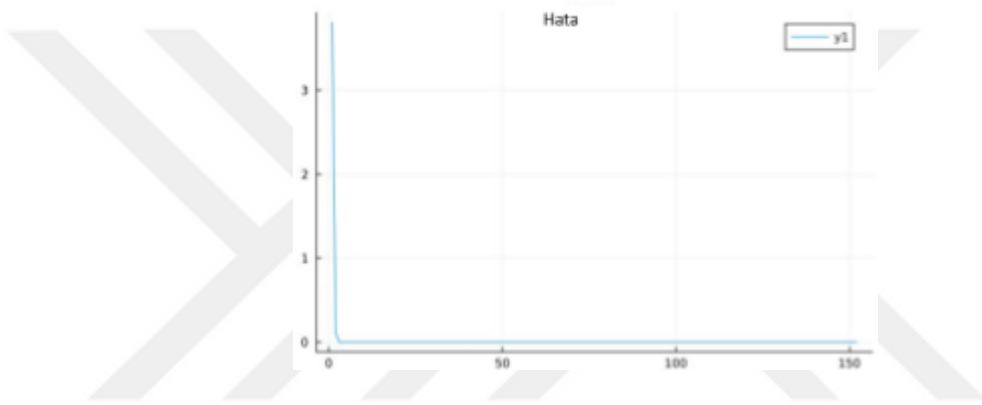
Şekil 5.3: 3 boyutlu grafiği



Şekil 5.4: Hata grafiği

N	Adım Sayısı	PINN
15	1000	$3.86 \times e^{-09}$
15	2000	$5.76 \times e^{-04}$
50	1000	$1.55 \times e^{-10}$
50	2000	$2.81 \times e^{-10}$

Tablo 5.2: Farklı parametrelerde PINN hata analizi



Şekil 5.5: Hata grafiği

BÖLÜM 6

SONUÇ

Bu tezde kısmi diferansiyel denklemlerin sayısal çözümlerini elde etmek için makine öğrenmesi ve derin öğrenme metodu olan PINN kullanılmaktadır. Bu ağlar, ele alınan kısmi diferansiyel denklemin doğasından öğrenmektedir. PINN, çeşitli kısmi diferansiyel denklemleri (KDD'ler) ele almada basit uygulama ve çok yönlülük avantajları sunmaktadır. Bununla birlikte, sonlu farklar yöntemi ve sonlu elemanlar yöntemi gibi diğer sayısal tekniklerle karşılaştırıldığında, PINN'in verimliliği otomatik türev alımından dolayı başarılı olduğu anlaşılmaktadır. Bu nedenle, PINN'nin KDD'lerin çözümü için kullanılan sayısal tekniklere değerli bir katkı olarak görülmesi gerektiği açıktır.

Kısmi diferansiyel denklemlerin arasında yer alan kesirli difüzyon denklemi, diferansiyel denklem sistemlerinden ise adi diferansiyel denklem sistemi, Benney-Roskes/akharov-Rubenchik sistemlerin sayısal çözümünde, sınır ağı tabanlı bir model oluşturmada ortaya çıkan yavaş eğitim ve yakınsama hızı gibi performansı etkileyen sorunları azaltmamızı sağlayan PINN yöntemi başarılı bir çalışma olmuştur. PINN rezidü, sınır koşulu ve başlangıç koşulu kısımlarından oluşan fiziksel kayıp fonksiyonunu minimize etmek için eğitim yaparak ilgili denklemlere yaklaşık çözümler üretebilmektedir.

PINN metodu uygulanan diferansiyel denklemlerin sayısal çözümlerinde elde edilen hata analizleri de bulunmaktadır. PINN metodunun uygulanması kısmi diferansiyel denklemler için tahmin yapabildiğini ortaya konmaktadır.

Uygulanan yöntemin sonuçlarına dair ilgili grafiklerde çalışmada yer almaktadır. Birbirinden farklı adım sayıları veparametrelerden elde edilen sonuçlar karşılaştırılmaktadır.

Gelecek çalışmalarda bu problemlerin PINN in diğer versiyonlarında çrneğin vanilla PINN, vs-PINN, A-PINN gibi uygulanarak sonuçlar karşılaştırılabilir. Yüksek mertebeden diferansiyel denklemlerin sayısal çözümlerinin de PINN yöntemi uygulanabilir

olup olmadığı araştırma konuları arasında yer alabilir.



Kaynakça

- [1] Chris Albon. *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*. O'Reilly Media, Inc., 1st edition, 2018.
- [2] Amirhossein Arzani and Scott T. M. Dawson. Data-driven cardiovascular flow modelling: examples and opportunities. *Journal of The Royal Society Interface*, 18(175), February 2021.
- [3] Amirhossein Arzani, Jian-Xun Wang, and Roshan M. D'Souza. Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Physics of Fluids*, 33(7), July 2021.
- [4] G. K. Batchelor. *An introduction to fluid dynamics* /. Cambridge mathematical library. Cambridge University Press,, Cambridge, U.K. ;, 2nd pbk. edition. edition, 1999. "First published 1967 ... first paperback edition 1973"—T.p. verso.
- [5] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey, 2018.
- [6] Henri Berestycki, Jean-Michel Roquejoffre, Luca Rossi, et al. The periodic patch model for population dynamics with fractional diffusion. *Discrete and Continuous Dynamical Systems. Series S*, 4:1–13, 2011.
- [7] Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K. Gupta. Clifford neural layers for pde modeling, 2023.
- [8] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers, 2023.

- [9] Haiim Brezis and Felix Browder. Partial differential equations in the 20th century. *Advances in Mathematics*, 135(1):76–144, 1998.
- [10] Steven Brunton and J. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. 01 2019.
- [11] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, January 2020.
- [12] Tesfaye Aga Bullo, Gemechis File Duressa, and Guy Degla. Accelerated fitted operator finite difference method for singularly perturbed parabolic reaction-diffusion problems. *Computational Methods for Differential Equations*, 9(3):886–898, 2021.
- [13] Hikmet Caglar, Nazan Caglar, Mehmet Fatih Uçar, and Canan Akkoyunlu. Non-polynomial spline method for fractional diffusion equation. *Journal of Computational Analysis & Applications*, 14(1), 2012.
- [14] Nazan Caglar and Hikmet Caglar. B-spline method for solving linear system of second-order boundary value problems. *Computers Mathematics with Applications*, 57(5):757–762, 2009.
- [15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [16] Shihua Chen, Jia Hu, Li Chen, and Changping Wang. Existence results for n-point boundary value problem of second order ordinary differential equations. *J. Comput. Appl. Math.*, 180(2):425–432, August 2005.
- [17] Xiyou Cheng and Chengkui Zhong. Existence of positive solutions for a second-order ordinary differential system. *Journal of Mathematical Analysis and Applications*, 312(1):14–23, 2005.
- [18] Ivan Christov. *Wave Solutions*, pages 6495–6506. 08 2012.
- [19] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

- [20] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [21] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks, 2020.
- [22] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989.
- [23] George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.
- [24] Tim De Ryck and Siddhartha Mishra. Numerical analysis of physics-informed neural networks and related models in physics-informed machine learning. *arXiv preprint arXiv:2402.10926*, 2024.
- [25] Mehdi Dehghan, Baharak Hooshyarfarzin, and Mostafa Abbaszadeh. Radial basis function partition of unity procedure combined with the reduced-order method for solving zakharov–rubenchik equations. *Engineering Analysis with Boundary Elements*, 145:93–116, 2022.
- [26] Anand Deshpande and Manish Kumar. *Artificial Intelligence for Big Data: Complete guide to automating Big Data solutions using Artificial Intelligence techniques*. Packt Publishing, 2018.
- [27] Pavitra Dhamija and Surajit Bag. Role of artificial intelligence in operations environment: a review and bibliometric analysis. *The TQM Journal*, ahead-of-print, 03 2020.
- [28] Nathan Doumèche, Gérard Biau, and Claire Boyer. Convergence and error analysis of pinns, 2023.
- [29] Dinh Dũng and Van Kien Nguyen. Deep relu neural networks in high-dimensional approximation. *Neural Networks*, 142:619–635, 2021.
- [30] Weinan Ee and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems, 09 2017.

- [31] A. Elsaid. The variational iteration method for solving riesz fractional partial differential equations. *Computers Mathematics with Applications*, 60(7):1940–1947, 2010.
- [32] Josef Fojdl and Rüdiger W. Brause. The performance of approximating ordinary differential equations by neural nets. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 457–464, 2008.
- [33] Vincent Fortuin. Priors in bayesian deep learning: A review, 2022.
- [34] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
- [35] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biol. Cybern.*, 20(3–4):121–136, September 1975.
- [36] Guang-hua Gao, Zhi-zhong Sun, and Ya-nan Zhang. A finite difference scheme for fractional sub-diffusion equations on an unbounded domain using artificial boundary conditions. *Journal of Computational Physics*, 231:2865–2879, 04 2012.
- [37] Han Gao, Luning Sun, and Jian-Xun Wang. Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. *Journal of Computational Physics*, 428:110079, 2021.
- [38] Arief Goeritno, Muhammad Nanang Prayudyanto, Puspa Eosina, Tika Siregar, and Roy Waluyo. Pde-based mathematical models to diagnose the temperature changes phenomena on the single rectangular plate-fin. *Mathematical Modelling of Engineering Problems*, 10:266–275, 02 2023.
- [39] Igor Goychuk and Peter Hänggi. Fractional diffusion modeling of ion channel gating. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70:051915, 12 2004.
- [40] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

- [41] Guang hua Gao, Zhi zhong Sun, and Hong wei Zhang. A new fractional numerical differentiation formula to approximate the caputo fractional derivative and its applications. *Journal of Computational Physics*, 259:33–50, 2014.
- [42] Syed M Hussain and Wasim Jamshed. A comparative entropy based analysis of tangent hyperbolic hybrid nanofluid flow: Implementing finite difference method. *International Communications in Heat and Mass Transfer*, 129:105671, 2021.
- [43] August Johansson, Benjamin Kehlet, Mats G. Larson, and Anders Logg. Multimesh finite element methods: Solving pdes on multiple intersecting meshes. *Computer Methods in Applied Mechanics and Engineering*, 343:672–689, 2019.
- [44] Anastasis Kratsios and Léonie Papon. Universal approximation theorems for differentiable geometric deep learning. *J. Mach. Learn. Res.*, 23(1), January 2022.
- [45] Manoj Kumar and Neha Yadav. Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey. *Computers Mathematics with Applications*, 62(10):3796–3811, 2011.
- [46] S. Kumar, A. Yildirim, Yasir Khan, and L. Wei. A fractional model of the diffusion equation and its analytical solution using laplace transform. *Scientia Iranica*, 19(4):1117–1123, 2012.
- [47] Isaac Lagaris, Aristidis Likas, and Dimitrios Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9:987–1000, 09 1998.
- [48] Alejandro C. Limache, P.J. Sánchez, Lisandro D. Dalcín, and Sergio R. Idelsohn. Objectivity tests for navier–stokes simulations: The revealing of non-physical solutions produced by laplace formulations. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4180–4192, 2008.
- [49] Qidong Lin, Liang Zhang, Zhang Yaolong, and Bin Jiang. Searching configurations in uncertainty space: Active learning of high-dimensional neural network reactive potentials. *Journal of chemical theory and computation*, 17, 04 2021.
- [50] S.D. Lowe. *Deep Learning for Dummies, Hpe Special Edition (Custom)*. John Wiley & Sons, Incorporated, 2018.

- [51] Stefano Markidis. The old and the new: Can physics-informed deep-learning replace traditional linear solvers?, 2021.
- [52] Claudio Mattiussi. The finite volume, finite difference, and finite elements methods as numerical methods for physical field problems. 05 2001.
- [53] Warren Mcculloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.
- [54] A.J. Meade and A.A. Fernandez. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, 19(12):1–25, 1994.
- [55] Mark M. Meerschaert and Charles Tadjeran. Finite difference approximations for fractional advection–dispersion flow equations. *Journal of Computational and Applied Mathematics*, 172(1):65–77, 2004.
- [56] Mark M. Meerschaert, Yong Zhang, and Boris Baeumer. Particle tracking for fractional diffusion with two time scales. *Computers Mathematics with Applications*, 59(3):1078–1086, 2010. Advance in Fractional Differential Equations.
- [57] A.C. Müller and S. Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media, 2016.
- [58] Xin Ning, Weijuan Tian, Zaiyang Yu, Weijun Li, Xiao Bai, and Yuebao Wang. Hcfn: High-order coverage function neural network for image classification. *Pattern Recognition*, 131:108873, 2022.
- [59] I. Omelianenko. *Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms*. Packt Publishing, 2019.
- [60] Alfred R. Osborne. Chapter 2 - nonlinear water wave equations. In Alfred R. Osborne, editor, *Nonlinear Ocean Waves and the Inverse Scattering Transform*, volume 97 of *International Geophysics*, pages 33–47. Academic Press, 2010.
- [61] M. Otadi and M. Mosleh. Numerical solution of quadratic riccati differential equation by neural network. *Math. Sci.*, 5:249–257, 01 2011.

- [62] Thomas O’Leary-Roseberry, Umberto Villa, Peng Chen, and Omar Ghattas. Derivative-informed projected neural networks for high-dimensional parametric maps governed by pdes. *Computer Methods in Applied Mechanics and Engineering*, 388:114199, 2022.
- [63] Hong-Kui Pang and Hai-Wei Sun. Multigrid method for fractional diffusion equations. *Journal of Computational Physics*, 231(2):693–703, 2012.
- [64] Dimitris C Psychogios and Lyle H Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.
- [65] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 06 2019.
- [66] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [67] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.
- [68] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data, 2018.
- [69] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.
- [70] John Paul Roop. Numerical approximation of a one-dimensional space fractional advection–dispersion equation with boundary layer. *Computers Mathematics with Applications*, 56(7):1808–1819, 2008.
- [71] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

- [72] Charles Tadjeran and Mark M. Meerschaert. A second-order accurate numerical method for the two-dimensional fractional diffusion equation. *Journal of Computational Physics*, 220(2):813–823, 2007.
- [73] V. E. Tarasov. Fractional integro-differential equations for electromagnetic waves in dielectric media. *Theoretical and Mathematical Physics*, 158(3):355–359, March 2009.
- [74] T. Teofili. *Deep Learning for Search*. Manning, 2019.
- [75] F. Ureña, L. Gavete, A. García, J.J. Benito, and A.M. Vargas. Solving second order non-linear hyperbolic pdes using generalized finite difference method (gfdm). *Journal of Computational and Applied Mathematics*, 363:1–21, 2020.
- [76] Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, Junjun Qiu, Keqin Hua, Wentao Su, Jian Wu, Huiyu Xu, Yong Han, Chenguang Fu, Zhigang Yin, Miao Liu, Ronald Roepman, Sabine Dietmann, Marko Virta, Fredrick Kengara, Ze Zhang, Lifu Zhang, Taolan Zhao, Ji Dai, Jialiang Yang, Liang Lan, Ming Luo, Zhaofeng Liu, Tao An, Bin Zhang, Xiao He, Shan Cong, Xiaohong Liu, Wei Zhang, James P. Lewis, James M. Tiedje, Qi Wang, Zhulin An, Fei Wang, Libo Zhang, Tao Huang, Chuan Lu, Zhipeng Cai, Fang Wang, and Jiabao Zhang. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4):100179, 2021.
- [77] Shaojie Zeng, Zong Zhang, and Qingsong Zou. Adaptive deep neural networks methods for high-dimensional partial differential equations. *Journal of Computational Physics*, 463:111232, 2022.
- [78] Zi-Ke Zhang, Chuang Liu, Xiu-Xiu Zhan, Xin Lu, Chu-Xu Zhang, and Yi-Cheng Zhang. Dynamics of information diffusion and its applications on complex networks. *Physics Reports*, 651:1–34, 2016. Dynamics of information diffusion and its applications on complex networks.
- [79] Álvaro Cartea and Diego del Castillo-Negrete. Fractional diffusion models of option prices in markets with jumps. *Physica A: Statistical Mechanics and its Applications*, 374(2):749–763, 2007.

- [80] Cem Çelik and Melda Duman. Crank–nicolson method for the fractional diffusion equation with the riesz fractional derivative. *Journal of Computational Physics*, 231(4):1743–1750, 2012.

