

T.C.
AKDENİZ ÜNİVERSİTESİ



**YAPAY SİNİR AĞI KULLANARAK YAPISAL OPTİMİZASYON
PROBLEMLERİNDEKİ ÇÖZÜM SÜREÇLERİNİN KISALTILMASI**

Tevfik Oğuz ÖRMECİOĞLU
FEN BİLİMLERİ ENSTİTÜSÜ
İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI
DOKTORA TEZİ

HAZİRAN 2024

ANTALYA

**T.C.
AKDENİZ ÜNİVERSİTESİ**



**YAPAY SİNİR AĞI KULLANARAK YAPISAL OPTİMİZASYON
PROBLEMLERİNDEKİ ÇÖZÜM SÜREÇLERİNİN KISALTILMASI**

Tevfik Oğuz ÖRMECİOĞLU

FEN BİLİMLERİ ENSTİTÜSÜ

İNŞAAT MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

HAZİRAN 2024

ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞI KULLANARAK YAPISAL OPTİMİZASYON
PROBLEMLERİNDEKİ ÇÖZÜM SÜREÇLERİNİN KISALTILMASI

Tevfik Oğuz ÖRMECİOĞLU

İNŞAAT MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

HAZİRAN 2024

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY SİNİR AĞI KULLANARAK YAPISAL OPTİMİZASYON
PROBLEMLERİNDEKİ ÇÖZÜM SÜREÇLERİNİN KISALTILMASI**

Tevfik Oğuz ÖRMECİOĞLU
İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI
DOKTORA TEZİ

Bu tez 04/06/2024 tarihinde jüri tarafından Oybirliği ile kabul edilmiştir.

Doç.Dr. İbrahim AYDOĞDU (Danışman)
Prof.Dr. Ömer CİVALEK
Prof.Dr. Erkan DOĞAN
Doç.Dr. Bekir AKGÖZ
Dr.Ögr.Üyesi Serdar ARSLAN

ÖZET

YAPAY SİNİR AĞI KULLANARAK YAPISAL OPTİMİZASYON PROBLEMLERİNDEKİ ÇÖZÜM SÜREÇLERİNİN KISALTILMASI

Tevfik Oğuz ÖRMECİOĞLU

Doktora Tezi, İnşaat Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. İbrahim AYDOĞDU

Eş Danışman: Prof. Dr. Niyazi Uğur KOÇKAL

Haziran 2024; 76 sayfa

Son yirmi yılda bilim dünyamıza ve günlük hayatımıza yoğun bir biçimde girmiş olan yapay zekâ çalışmaları aslında 1950’li yıllara dayanmaktadır. Keşfedildiği dönemdeki donanımsal imkansızlıkların günümüzde hızla giderilmesi, birçok alanda yapay zekâ çözümlerinin geliştirilmesini ve yaygınlaşmasını sağlamıştır. Bu ilerlemeler, matematiksel yoğunlukta ve çok tekrarlı olan algoritmaların daha hızlı bir biçimde çözülebilmesine olanak tanımıştır. Özellikle yapı optimizasyonu alanında geleneksel yöntemlerin hesaplama yoğunluğu, karmaşık yapıların analizini ve optimizasyonunu zorlaştırmaktadır. Yapı optimizasyonu, yapısal sistemlerin tasarım değişkenlerini belirleyerek belirli amaç fonksiyonlarını en üst düzeye çıkarmak veya belirli kısıtları karşılamak için kullanılır. Bu, yapıların mekanik performansını, enerji verimliliğini ve maliyet etkinliğini artırmada önemli bir rol oynar. Gelişen optimizasyon algoritmaları, daha hafif ve güçlü yapılar tasarlama imkânı sunarak daha karmaşık problemlere uygulanabilirlik sağlar.

Meta-sezgisel yöntemler, yapı optimizasyonu için yaygın bir şekilde kullanılır ve yapıların performansını ve verimliliğini iyileştirmede etkilidir. Ancak, sonlu elemanlar yöntemi (FEM) gibi yapı analizi yöntemleriyle birlikte kullanıldığında çözüm süresi artabilir. Bu durum, FEM’in yapıların davranışını doğru bir şekilde modellemek için yoğun hesaplama gerektirmesinden kaynaklanır. Bu tez çalışmasında yapısal optimizasyon algoritmalarının işlem sürelerinin kısaltılması amaçlanmıştır. Bunun için optimizasyon algoritmalarının ortak ve en çok süre alan adımı olan “Değerlendirme Aşaması”nın süre açısından iyileştirilmesi için bir yapay sinir ağı (YSA) geliştirilmiştir. Yapı analizi işleminin ana adımı olan deplasman hesapları esnasında oluşan sürelerin azaltılmasına odaklanmıştır. Geleneksel yöntemlerle yapılan deplasman hesaplamaları zaman alıcı ve karmaşık olabilirken, YSA ve optimizasyon algoritmalarının birleşimi sayesinde bu süreler önemli ölçüde azaltılabilmektedir.

Çalışmada geliştirilen YSA tabanlı yaklaşım, geleneksel yöntemlerin sınırlamalarını aşarak daha hızlı ve verimli çözümler sunmayı hedeflemiştir. YSA-FEM çözümünün yapı problem tipinden bağımsız olması, farklı yapı türleri için de uygulanabilir bir çözüm sunmaktadır. Geliştirilen YSA tabanlı matrissiz yöntem, FEM analizi içerisinde en çok zaman ve bellek tüketen Gauss eliminasyon işlemlerini atlayarak deplasman vektörünü tahmin etmek üzerine kurulu bir mekanizmaya sahiptir. YSA’nın tahminlerini optimizasyon sürecine entegre etmek için üç yeni hibrit Biyocoğrafya Tabanlı Optimizasyon (BTO) arama yöntemi (YT-1, YT-2 ve YT-K) geliştirilmiştir. Bu yöntemler, YSA’nın tahminlerini stratejik noktalarda kullanarak optimizasyon sürecini

yönlendirir ve daha hızlı yakınsama sağlamayı hedeflemesi için geliştirilmişlerdir. Çalışma kapsamında gerçekleştirilen testler, farklı yapı tipleri üzerinde YSA tabanlı yöntemin FEM'e kıyasla hesaplama süresinde önemli bir azalma sağladığını göstermiştir. YSA'nın doğruluk seviyesi FEM'e ulaşmasa da sunduğu hız avantajı, özellikle gerçek zamanlı uygulamalar veya hızlı tasarım yinelemeleri gerektiren senaryolarda önemli bir faktör haline gelmektedir. Bu bulgular, YSA tabanlı yöntemlerin yapısal optimizasyon için uygulanabilir ve verimli bir alternatif sunduğunu ve daha ileri araştırmalar için umut vadeden bir yol açtığını göstermektedir.

ANAHTAR KELİMELELER: Optimizasyon, Sonlu Elemanlar Metodu, Yapay Sinir Ağı, Yapısal Analiz.

JURİ: Doç. Dr. İbrahim AYDOĞDU (Danışman)

Prof. Dr. Ömer CİVALEK

Prof. Dr. Erkan DOĞAN

Doç. Dr. Bekir AKGÖZ

Dr. Öğr. Üyesi Serdar ARSLAN

ABSTRACT

REDUCING THE SOLUTION PROCESSES IN STRUCTURAL OPTIMIZATION PROBLEMS USING NEURAL NETWORKS

Tevfik Oğuz ÖRMECİOĞLU

PhD Thesis in Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. İbrahim AYDOĞDU

Co-Supervisor: Prof. Dr. Niyazi Uğur KOÇKAL

June 2024; 76 pages

In the past twenty years, artificial intelligence (AI) research has significantly entered our scientific and daily lives, although its origins date back to the 1950s. The rapid resolution of hardware limitations from its discovery period has enabled the development and widespread use of AI solutions in many fields. These advancements have allowed algorithms, which are mathematically intensive and repetitive, to be solved more quickly. In particular, the computational intensity of traditional methods in structural optimization complicates the analysis and optimization of complex structures. Structural optimization involves determining the design variables of structural systems to maximize certain objective functions or meet specific constraints. This plays a crucial role in enhancing the mechanical performance, energy efficiency, and cost-effectiveness of structures. Advancing optimization algorithms provide the capability to design lighter and stronger structures, making them applicable to more complex problems.

Metaheuristic methods are widely used for structural optimization and are effective in improving the performance and efficiency of structures. However, when used in conjunction with structural analysis methods such as the finite element method (FEM), the solution time can increase. This is due to the intensive computation required by FEM to accurately model the behavior of structures. This thesis aims to reduce the processing times of structural optimization algorithms. To achieve this, an artificial neural network (ANN) has been developed to improve the "Evaluation Phase," which is the common and most time-consuming step of optimization algorithms. The focus is on reducing the time generated during the displacement calculations, which is the main step of the structural analysis process. While displacement calculations performed by traditional methods can be time-consuming and complex, the combination of ANN and optimization algorithms significantly reduces these times.

The ANN-based approach developed in this study aims to offer faster and more efficient solutions by overcoming the limitations of traditional methods. The ANN-FEM solution is independent of the type of structural problem, providing an applicable solution for different types of structures. The ANN-based matrix-free method developed focuses on predicting the displacement vector by bypassing the Gauss elimination processes, which consume the most time and memory in FEM analysis. To integrate ANN predictions into the optimization process, three new hybrid Biogeography-Based Optimization (BBO) search methods (YT-1, YT-2, and YT-K) have been developed. These methods strategically use ANN predictions to guide the optimization process and

aim for faster convergence. Tests conducted within the scope of the study showed that the ANN-based method significantly reduced computation time compared to FEM for different types of structures. Although the accuracy level of ANN does not reach FEM, its speed advantage becomes a significant factor, especially in scenarios requiring real-time applications or rapid design iterations. These findings demonstrate that ANN-based methods provide a viable and efficient alternative for structural optimization and open a promising path for further research.

KEYWORDS: Artificial Neural Network, Finite Element Method, Optimization, Structural Analysis.

COMMITTEE: Assoc. Prof. Dr. İbrahim AYDOĞDU (Advisor)

Prof. Dr. Ömer CİVALEK

Prof. Dr. Erkan DOĞAN

Assoc. Prof. Dr. Bekir AKGÖZ

Assist. Prof. Dr. Serdar ARSLAN

ÖNSÖZ

Bu tez çalışmasının hazırlanmasında emeği geçen herkese teşekkürü bir borç bilirim. Yapay zekâ ve yapısal optimizasyon alanlarında yapılan bu çalışma, günümüz teknolojik gelişmelerinin yapı mühendisliği üzerindeki etkilerini araştırmayı amaçlamaktadır. Bu süreçte, karşılaştığım zorlukları aşmamda büyük rol oynayan, başta danışman hocam Doç. Dr. İbrahim Aydoğdu ve yardımcı danışman hocam Prof. Dr. Niyazi Uğur Koçkal ve tez izleme komitesi üyeliği yapan hocalarım olmak üzere, çalışmam boyunca yanımda olan aileme ve arkadaşlarıma teşekkür ederim.



İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	iii
SİMGELER VE KISALTMALAR.....	ix
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. KAYNAK TARAMASI	5
2.1. YSA Kullanarak Yapısal Optimizasyon ile İlgili Yapılan Çalışmalar.....	5
2.2. Yapısal Optimizasyonda BTO (Biyocoğrafya Tabanlı Optimizasyon) Kullanımı ile İlgili Çalışmalar.....	6
2.3. Yapısal Optimizasyonunda Yapı Davranışının Temsili Model Kullanımı ile İlgili Çalışmalar.....	7
3. MATERYAL VE METOT	9
3.1. Çerçeve ve Kafes Sistemlerinin Analizi (FEM Metodu)	9
3.2. Yapay Sinir Ağları.....	12
3.3. Biyolojik Sinir Ağlarının Yapısı	14
3.4. Yapay Sinir Ağlarının Temel Yapısı.....	15
3.4.1. Ağ katmanları	15
3.4.2. Aktivasyon fonksiyonları.....	20
3.4.3. Yapay sinir ağı eğitimi.....	23
3.5. Eğitim Setinin Oluşturulması	26
3.5.1. Yapı örneklerinin eğitim setine dönüştürülmesi.....	29
3.6. Geliştirilen Yapay Sinir Ağı Modelleri	33
3.6.1. Yapay sinir ağı modeli 1	34
3.6.2. Yapay sinir ağı modeli 2.....	36
3.6.3. Yapay sinir ağı modeli 3.....	39
3.6.4. Yapay sinir ağı modeli 4.....	41
3.6.5. Yapay sinir ağı modeli 5.....	43
3.7. Modellerin Karşılaştırılması.....	46
3.8. Yapay Sinir Ağlarının İteratif Çözüm Teknikleri ile Birleştirilmesi	47
3.8.1. Biyocoğrafya tabanlı optimizasyon (BTO) algoritması	48

3.8.2. Geliştirilen BTO optimizasyon algoritma sonuçlarının karşılaştırılması	50
3.8.3. Geliştirilen BTO arama yöntemleri	52
4. BULGULAR VE TARTIŞMA	54
4.1. Kullanılan Örnekler	54
4.2. BTO Arama Yöntemlerinin Yakınsama Açısından Karşılaştırılması	55
4.2.1. BTO arama yöntemlerinin istatistiksel performans açısından karşılaştırılması.....	60
4.2.2. BTO arama yöntemlerinin çözüm hızı açısından karşılaştırılması.....	67
5. SONUÇLAR	72
6. KAYNAKLAR	74
ÖZGEÇMİŞ	

AKADEMİK BEYAN

Doktora Tezi olarak sunduğum “Yapay Sinir Ağı Kullanarak Yapısal Optimizasyon Problemlerindeki Çözüm Süreçlerinin Kısaltılması” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

13/06/2024

Y. Müh. Tefik Oğuz ÖRMECİOĞLU



SİMGELER VE KISALTMALAR

Kısaltmalar

BCE	: İkili Çapraz Entropi (Binary Cross-Entropy)
BTO	: Biyocoğrafya Tabanlı Optimizasyon (Bio-Geography Optimization)
CCE	: Kategorik Çapraz Entropi (Categorical Cross-Entropy)
CNN	: Evrişimli Sinir Ağı (Convolutional Neural Networks)
CPN	: Karşı Yayılım Ağı (Counter Propagation Network)
CUDA	: Birleşik Cihaz Mimarisi ile Hesaplama (Compute Unified Device Architecture)
FEA	: Sonlu Elemanlar Analizi (Finite Element Analysis)
FEM	: Sonlu Elemanlar Metodu (Finite Element Method)
HAD	: Hesaplamalı Akışkanlar Dinamiği
HUI	: Habitat Uygunluk İndeksi
MAE	: Ortalama Mutlak Hata (Mean Absolute Error)
MAPE	: Ortalama Mutlak Yüzdesel Hata (Mean Absolute Percentage Error)
MCS	: Monte Carlo simülasyonu
MSE	: Ortalama Karesel Hata (Mean Squared Error)
R2	: Belirleme Katsayısı (Coefficient of Determination)
RMSE	: Ortalama Mutlak Karesel Hata (Root Mean Square Error)
SPAI	: Ters Seyrek Yaklaşık Çözüm (Sparse Approximate Inverse)
STD	: Standart Sapma (Standard Deviation)
YSA	: Yapay Sinir Ağları (Artificial Neural Network)
YSD	: Yapı Serbestlik Derecesi

ŞEKİLLER DİZİNİ

Şekil 3.1 Bir çubuk üzerindeki düğüm uçlarında oluşan kuvvetler, momentler ve deplasmanlar (Aydogdu 2017).....	9
Şekil 3.2 Rijitlik matrisi.....	10
Şekil 3.3 Çerçeve elemanının her iki ucunu da tüm serbestlik derecelerinde temsil eden koordinat transformasyon matrisi	10
Şekil 3.4 Lokal eksen ile global eksen arasındaki koordinat transformasyonu (Aydogdu 2017)	11
Şekil 3.5 Renkli bir resmin üç kanallı bir tensör olarak imgenenmiş hali (Anonymous 1 2024)	12
Şekil 3.6 Renkli bir resmin üç kanallı bir tensör olarak imgenenmiş hali (Anonymous 2 2024)	13
Şekil 3.7 Yapay nöron ve biyolojik nöron karşılaştırması (Anonymous 3 2024)	14
Şekil 3.8 Conv2D Katmanı (Anonim 4 2024)	16
Şekil 3.9 “Max Pooling” ve “Average Pooling” işlem örneği (Islam 2020)	17
Şekil 3.10 Bir dropout katmanı uygulama şeması (Anonymous 5 2024).....	18
Şekil 3.11 Flatten Katmanı	19
Şekil 3.12 Dense Katmanı (Izyumov vd. 2021).....	20
Şekil 3.13 Aktivasyon fonksiyonları ve türevleri (a-sigmoid fonksiyonu; b-tanh fonksiyonu; c-ReLu fonksiyonu; d-leaky ReLu fonksiyonu; e-swish fonksiyonu)	21
Şekil 3.14 Bünyesinde çerçeve yapı tipi kullanılan Örnekler a) Dar kule; b) Geniş kule	27
Şekil 3.15 Veri Seti Oluşturulması İşlem Hattı	29
Şekil 3.16 Girdinin Sıkıştırılması İşlemi	30
Şekil 3.17 Sıkıştırıldıktan Sonra Büyütülme İşlemine Tabi Tutulmuş Rijitlik Matrisi Örneği.....	31
Şekil 3.18 YSA Çıktısının Genişletilmesi İşlemi	32
Şekil 3.19 Model 1 YSA şematik gösterimi	35
Şekil 3.20 YSA Model 1 en iyi altı hiper parametre sonuçları.....	35
Şekil 3.21 YSA Model 1’e ait R2 ve Doğrulama R2 sonuçları	36
Şekil 3.22 YSA Model 1’e ait MAE ve Doğrulama MAE sonuçları.....	36
Şekil 3.23 Model 2 YSA şematik gösterimi	37
Şekil 3.24 YSA Model 2 en iyi beş hiper parametre sonuçları.....	38
Şekil 3.25 YSA Model 2’e ait R2 ve Doğrulama R2 sonuçları	38
Şekil 3.26 YSA Model 2’e ait MAE ve Doğrulama MAE sonuçları.....	39
Şekil 3.27 Model 3 YSA şematik gösterimi	40
Şekil 3.28 YSA Model 3 en iyi beş hiper parametre sonuçları.....	40
Şekil 3.29 YSA Model 3’e ait R2 ve Doğrulama R2 sonuçları	41
Şekil 3.30 YSA Model 3’e ait MAE ve Doğrulama MAE sonuçları.....	41
Şekil 3.31 Model 4 YSA şematik gösterimi	42
Şekil 3.32 YSA Model 4 en iyi beş hiper parametre sonuçları.....	42
Şekil 3.33 YSA Model 4’e ait R2 ve Doğrulama R2 sonuçları	43
Şekil 3.34 YSA Model 4’e ait MAE ve Doğrulama MAE sonuçları.....	43
Şekil 3.35 YSA Model 5 Grafikselsel Gösterimi	44
Şekil 3.36 YSA Model 5 en iyi beş hiper parametre sonuçları.....	45
Şekil 3.37 YSA Model 5’e ait R2 ve Doğrulama R2 sonuçları	45
Şekil 3.38 YSA Model 5’e ait MAE ve Doğrulama MAE sonuçları.....	46

Şekil 3.39 Tüm YSA modellerinin kendileri arasında karşılaştırılmaları.....	46
Şekil 3.40 Standart BTO optimizasyon algoritması akış diyagramı.....	49
Şekil 3.41 18 çubuklu konsol kafes örneği BTO arama geçmişi.....	51
Şekil 3.42 18 çubuklu düzlem kafes yapı örneği	51
Şekil 3.43 BTO-Yöntem-v1 Arama Yöntemi akış diyagramı	52
Şekil 3.44 BTO-Yöntem-v2 Arama Yöntemi akış diyagramı	53
Şekil 3.45 BTO-Yöntem-Karma Arama Yöntemi akış diyagramı	53
Şekil 4.1 AnkastreKiriş-1 örneği için arama geçmişi	55
Şekil 4.2 AnkastreKiriş-2 örneği için arama geçmişi	56
Şekil 4.3 Dar kule örneği için arama geçmişi	56
Şekil 4.4 Geniş kule örneği için arama geçmişi.....	57
Şekil 4.5 KlasikKöprü-1 örneği için arama geçmişi.....	58
Şekil 4.6 KlasikKöprü-2 örneği için arama geçmişi.....	58
Şekil 4.7 Piramit örneği için arama geçmişi	59
Şekil 4.8 AnkastreKiriş-1 örneği için arama geçmişi	60
Şekil 4.9 AnkastreKiriş-2 örneği için arama geçmişi	61
Şekil 4.10 Dar kule örneği için arama geçmişi	62
Şekil 4.11 Geniş kule örneği için arama geçmişi.....	64
Şekil 4.12 KlasikKöprü-1 örneği için arama geçmişi.....	65
Şekil 4.13 KlasikKöprü-2 örneği için arama geçmişi.....	66
Şekil 4.14 YT-1, YT-2, YT-K ve FEM arama yöntemlerinin her bir yapı tipi için maksimum, minimum süre tüketim değerleri ve standart sapmaları	68
Şekil 4.15 YT-1, YT-2 ve YT-K arama yöntemlerinin her bir yapı tipi için standart sapma sonuçları	69
Şekil 4.16 YT-1, YT-2 ve YT-K arama yöntemlerinin her bir yapı tipi içerisindeki FEM/YSA kullanım süre oranları	70

ÇİZELGELER DİZİNİ

Çizelge 3.1 Eğitim Seti Yapı Tipleri ve Temel Özellikleri	26
Çizelge 3.2 “Klasik Köprü” örneğini oluşturmak için gerekli değişkenlerin limitleri ...	28
Çizelge 3.3 Ana Yapı Tipleri ve Farklılık İhtimalleri.....	28
Çizelge 3.4 YSA Modellerine ait parametre değerleri.....	34
Çizelge 3.5 YSA Model 5 en iyi beş hiper parametre çizelgesi	47
Çizelge 3.6 18 çubuklu konsol kafesin optimum tasarım karşılaştırma sonuçları.....	51
Çizelge 4.1 BTO yöntemleri ile kullanılan örneklerin yapısal özellikleri	54
Çizelge 4.2 AnkastreKiriş-1 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)...	60
Çizelge 4.3 AnkastreKiriş-2 örneği ile ilgili arama yöntemleri sonuçları (Birim:kg)....	61
Çizelge 4.4 Dar Kule örneği ile ilgili arama yöntemleri sonuçları (Birim: kg).....	62
Çizelge 4.5 Geniş Kule örneği ile ilgili arama yöntemleri sonuçları (Birim: kg).....	63
Çizelge 4.6 KlasikKöprü-1 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg).....	65
Çizelge 4.7 KlasikKöprü-2 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg).....	66
Çizelge 4.8 Deney bilgisayarı donanım özellikleri.....	67

1. GİRİŞ

Yapı optimizasyonu, yapısal sistemlerin tasarım değişkenlerini belirleyerek belirli amaç fonksiyonlarını en üst düzeye çıkarmak veya belirli kısıtları karşılamak için kullanılan bir süreçtir. Bu süreç, yapıların mekanik performansını, enerji verimliliğini ve maliyet etkinliğini artırmada önemli bir rol oynar. Gelişen optimizasyon algoritmaları, daha hafif ve güçlü yapılar tasarlama imkânı sunarak daha karmaşık problemlere uygulanabilirlik sağlar. Meta-sezgisel yöntemler, yapı optimizasyonunda geniş bir uygulama alanı bulur ve yapıların performansını ve verimliliğini iyileştirmede etkilidir. Sezgisel algoritmaların sezgiye dayalı kurallara dayanmasının aksine, meta-sezgisel algoritmalar, üst düzey mantık kullanarak sezgisel algoritmaların performansını artırır. Her iki yöntem de karmaşık problemleri çözmek için sezgisel yaklaşımlar kullanır. Ancak, meta-sezgisel algoritmalar, yerel optimumlara takılma, uyum sağlama ve çeşitlilik gibi sorunları aşmak için daha gelişmiş stratejiler uygular.

Ancak, sonlu elemanlar yöntemi (FEM) gibi yapı analizi yöntemleriyle birlikte kullanıldığında, meta-sezgisel yöntemlerin çözüm süresi artabilir. Bu durum, FEM'in yapıların davranışını doğru bir şekilde modellemek için yoğun hesaplama gerektirmesinden kaynaklanır. Algoritma parametrelerini ayarlamak performansı artırabilir, ancak yapısal optimizasyon problemlerinde süre açısından iyileştirme sınırlı kalabilir.

Doğadan ilham alan teknikler, parçacık sürü optimizasyonu ve biyocoğrafya tabanlı optimizasyon (BTO) gibi yaygın meta-sezgisel teknikler, büyük arama uzaylarını keşfederken iteratif çözüm kullanır. Bu, özellikle karmaşık problemlerde uzun çözüm sürelerine yol açar. Binlerce iterasyon gerektirebilen bu yöntemler, gerçek zamanlı uygulamalar veya hızlı tasarım optimizasyonu gerektiren senaryolarda önemli bir dezavantajdır. Bu süre dezavantajını tersine çevirebilmek için yapılan iyileştirmeler genellikle üç ana başlık altında toplanır: Donanımsal İyileştirmeler, Optimizasyon Teknikleri ile Alakalı İyileştirmeler ve FEM ile Alakalı İyileştirmeler.

Donanımsal iyileştirmeler, optimizasyon problemlerinin çözümünü hızlandırmak için etkili bir yaklaşımdır. Ancak, bu yöntemler donanım sayısının artırılmasına bağlı oldukları için genellikle maliyetli çözümlerdir. Birden çok bilgisayarı bir ağ üzerinden birleştirerek iş paylaşımı yapmak, hesaplama gücünü artırabilir. Ancak, bu yaklaşım donanım maliyetlerini artırır ve veri iletimi darboğazlarına yol açabilir. Bilgisayar sayısı arttıkça, iletişim yükü de artar ve bu da işlem süresinde kayıplara neden olur. (Örmecioğlu, 2020) GPU işlemcileri, paralel hesaplama yetenekleriyle optimizasyon problemlerini hızlandırmada etkilidir. Ön koşullandırılmış iteratif metotlar GPU üzerinde kullanılan etkin sonuç veren yöntemlerdendir. Ancak, bu çözümler büyük ve karmaşık yapılar için avantajlı olup, CUDA destekli özel GPU'lara ihtiyaç duyar. Özel GPU'lara ihtiyaç duyan bu çözümler ek maliyetler oluşturmaktadır. (Örmecioğlu, 2020)

Optimizasyon alanında, çeşitli teknikler ve metodolojiler kullanılarak algoritmalar üzerinde değişiklikler yapılmış ve bu sayede çeşitli iyileştirmeler elde edilmiştir. Bu tekniklerden biri olan Levy uçuşu, rastgele arama algoritmasının bir parçası olarak kullanılır ve genellikle global optimizasyon problemlerinde etkilidir (Li vd. 2022). Açıgözlü seçim ise, yerel en iyi çözümü hızlı bir şekilde bulmayı amaçlar, ancak genellikle yerel optimuma takılır (Cormen vd. 2001). Kaos teorisi, deterministik ama

öngörülemeyen davranışları modellemek için kullanılır ve optimizasyon algoritmalarının performansını artırmak için kullanılabilir, çünkü kaotik sistemler genellikle geniş bir parametre aralığını keşfeder (Aslimani ve Ellaia 2018). Hibrid teknikler, birden fazla optimizasyon tekniğini birleştirerek her birinin avantajlarından yararlanır ve dezavantajlarını hafifletir. Çok kriterli karar verme teknikleri, birden fazla hedefi dikkate alarak optimizasyon problemlerini çözer (Taherdoost ve Madanchian 2023). Bu, genellikle bir dizi kabul edilebilir çözüm olan Pareto-optimal setini bulmayı içerir (Das ve Dennis 1998). Bu tekniklerin her biri, belirli bir optimizasyon problemine bağlı olarak, farklı durumlar ve uygulamalar için daha uygun olabilir. Bu nedenle, en iyi sonuçları elde etmek için genellikle bir dizi teknik kullanılır. Bu tekniklerin birleşimi, karmaşık optimizasyon problemlerinin çözümünde büyük bir etkiye sahip olabilir ve bu da daha etkili ve verimli sistemlerin geliştirilmesine yardımcı olabilir.

Optimizasyon süreçlerinde verimliliği artırmanın bir diğer yolu ise tasarım odaklı optimizasyon teknikleridir. Bu yaklaşım, bir meta-sezgisel arama algoritmasına verimli ve akıllı arama stratejileri entegre ederek optimum çözümü bulmak için gereken yapısal analiz sayısını minimize etmeye odaklanır. Başka bir deyişle, daha az sayıda analizle kabul edilebilir bir tasarım elde etmek hedeflenir. Bu bağlamda, önceki tasarımların analizlerinden elde edilen bilgiler, arama sürecini tasarım alanının daha umut verici bölgelerine yönlendirmek için kullanılır. Böylece, daha az analizle daha iyi çözümler elde edilir. Tam Gerilmeli Tasarım (FSD), bu stratejinin yapısal optimizasyon problemlerine uygulandığı öncü yöntemlerden biridir (Razani 1965, Gallagher ve Zienkiewicz 1973). Problem kısıtlamaları elemanlar için gerilme sınırlamalarından oluştuğunda etkili bir şekilde kullanılır. FSD, elemanların kısıtlama oranlarına göre bir arama yönü belirler. Kısıtlama oranı düşük olan elemanların boyutları, malzeme kullanımını optimize etmek için küçültülürken, yüksek kısıtlama oranına sahip elemanlar kısıtlama ihlalinin önlemek için büyütülür. Böylece, tüm elemanlar için yük-kapasite oranı (DCR) birliğe yaklaştırılır. FSD'nin etkinliğine rağmen, yalnızca gerilme veya mukavemet kısıtlamaları olan problemlerle sınırlıdır ve deplasman kısıtlamalarını dikkate alamaz. Bu sınırlamayı aşmak için Tam Kullanılmış Tasarım (FUD) (Saka vd. 2016) ve değiştirilmiş versiyonu olan MFUD (Patnaik vd. 1998) gibi yöntemler geliştirilmiştir. MFUD, ihlal edilen bir deplasman kısıtlamasının önemini belirlemek için entegre kuvvet yöntemi (IFM) kullanır (Patnaik vd. 1998).

Her ne kadar bu optimizasyon teknikleri birçok avantaj sunsa da her birinin kendine özgü zorlukları ve potansiyel dezavantajları da vardır. Örneğin, Levy uçuşu global optimizasyonda etkili olsa da yerel optimumları bulmakta zorlanabilir ve parametre ayarlaması karmaşık olabilir. Açgözlü seçim ise hızıyla öne çıksa da yerel optimuma takılma eğilimi gösterir ve global çözümlere ulaşmada yetersiz kalabilir. Kaos teorisi geniş bir parametre aralığını keşfetme potansiyeli sunsa da kaotik sistemlerin kontrolü ve yönetimi zorluk yaratabilir ve optimizasyon sürecini karmaşıkleştirebilir. Hibrid teknikler birden fazla tekniğin gücünü birleştirmeyi amaçlasa da parametre ayarlama ve uyumluluk sorunları beraberinde getirebilir ve hesaplama maliyetlerini artırabilir. Tasarım odaklı optimizasyon belirli hedeflere odaklansa da bu hedeflerin belirsizliği ve değişkenliği, uygulanabilirliği sınırlayabilir ve kullanım alanını belirli disiplinlerle kısıtlayabilir. Çok kriterli karar verme teknikleri birden fazla hedefi ele almada etkili olsa da Pareto-optimal setinin belirlenmesi ve yorumlanması zor olabilir ve uzlaşma çözümleri gerektirebilir. Bu nedenle, belirli bir optimizasyon problemini ele alırken, her tekniğin avantajlarını ve

dezavantajlarını dikkatlice değerlendirmek ve en uygun yaklaşımı seçmek, optimal sonuçlar elde etmek için kritik öneme sahiptir.

Bu alandaki bir diğer önemli yaklaşım, meta modellerin kullanımudur. Meta modeller, genellikle "yedek" veya "meta" modeller olarak da adlandırılır, temelde bir ana modelin daha basit bir temsildir ve bu modeller, ana modelin çıktılarını tahmin etmek için kullanılır. Meta modeller, genellikle ana modelin çıktılarına karşı girdileri eşleştiren bir dizi matematiksel denklem kullanır. Bu, ana modelin karmaşıklığını ve hesaplama yükünü önemli ölçüde azaltır.

Sonlu elemanlar metodu (FEM) tam ve zaman alıcı uygulamalarının yerine, yapısal davranışın tahmin edilmesi ve yanıt yüzeyi yaklaştırma yöntemleri aracılığıyla ara model kullanımı, mühendislikte yaygın bir uygulamadır. Bu yaklaşım, karmaşık mühendislik problemlerinin çözümünde önemli bir zaman ve kaynak tasarrufu sağlar. Yanıt yüzeyi yaklaştırma yöntemleri, genellikle bir dizi deneyden elde edilen verilere dayalı olarak bir sistemin çıktısını tahmin etmek için kullanılır. Bu yöntemler, genellikle polinomlar veya diğer basit fonksiyonlar kullanılarak bir yanıt yüzeyi oluşturur. Bu yüzey, daha sonra sistemin çıktısını tahmin etmek için kullanılır. Yapısal davranışın tahmin edilmesi için ara modellerin ve yanıt yüzeyi yaklaştırma yöntemlerinin kullanılması, genellikle daha hızlı ve daha az kaynak içeren bir alternatif sunar. Bu yaklaşım, bir yapının belirli yükler altında nasıl tepki vereceğini hızlı ve doğru bir şekilde tahmin etmek için kullanılır. Ara modellerin ve yanıt yüzeyi yaklaştırma yöntemlerinin kullanılması, mühendislikte karmaşık problemlerin çözümünde önemli bir araçtır. Bu yöntemler, daha hızlı ve daha az kaynak yoğun çözümler sunarak, mühendislerin zaman ve kaynaklarını daha verimli bir şekilde kullanmalarını sağlar. Bu, mühendislik tasarım ve analiz süreçlerini hızlandırır ve mühendislik projelerinin genel verimliliğini ve etkinliğini artırır. Bu nedenle, bu yöntemlerin anlaşılması ve uygulanması, modern mühendislik pratiği için hayati öneme sahiptir.

FEM çözüm süreçlerini kolaylaştırmak için çeşitli teknikler geliştirilmiştir. Geliştirilen bu tekniklerden olan, iteratif teknikler, büyük ve seyrek matrislerin çözümünde etkili bir yaklaşımdır. Kaczmarz yöntemi, eşlenik gradyan yöntemi ve ters seyrek yaklaşık çözümü (SPAI) gibi yöntemler, bu tür problemler için çözüm süresini önemli ölçüde azaltabilir. Ancak, bu tekniklerin etkinliği, problem tipine ve matris özelliklerine bağlıdır.

FEM çözümleri için kullanılabilecek bir başka yöntem tipi ise Matrisiz yöntemlerdir. Bu yöntemde, sistem matrisini açıkça oluşturmada veya depolamada lineer sistemleri çözmek için tasarlanmış bir algoritma sınıfı kullanılır. Bu yaklaşım, bellek kısıtlamalarının veya matris işlemleriyle ilişkili hesaplama maliyetlerinin engelleyici olduğu büyük ölçekli problemlerle uğraşırken çok önemli hale gelir.

Geliştirilen başka bir yöntem olan matrisiz zaman adımlama yöntemi, Jacobi matrisini açıkça oluşturmak ve depolamak yerine, Jacobi-vektör çarpımlarını yaklaşık olarak hesaplar. Bu, genellikle sonlu farklar yaklaşımı kullanılarak yapılır. Matrisiz zaman adımlama yöntemleri, daha düşük bellek kullanımı, modülerlik ve ölçeklenebilirlik bakımından genellikle büyük HAD simülasyonlarının hesaplanması için kullanılır (Sarshar vd. 2017).

Bu metotların dezavantajı olarak, özellikle blok dengeleme veya tensör dengeleme gibi varyantlar için, bazı teknik zorluklar içerebilmesi gösterilebilir. Metotlar, özellikle dağınık bellek sistemlerinde, iletişim maliyetlerinden etkilenebilir. Bu durum, Jacobi-vektör çarpımlarının hesaplanması sırasında farklı işlemciler arasında veri alışverişi gerektiğinde ortaya çıkar.

Matrissiz yöntemlere başka bir örnek de bu yazıda kullanılan YSA-FEM hibrit çözümleridir. YSA'lar, karmaşık doğrusal olmayan ilişkileri öğrenme ve tahmin etme yetenekleriyle öne çıkan öğrenme tekniklerindendir.

Bu çalışmada geliştirilen YSA tabanlı yaklaşım, geleneksel yöntemlerin sınırlamalarını aşarak daha hızlı ve verimli çözümler sunmayı hedeflemiştir. YSA-FEM çözümünün yapı problem tipinden bağımsız olması, farklı yapı türleri için de uygulanabilir bir çözüm sunar. Geliştirilen YSA tabanlı matrissiz yöntem, FEM analizi içerisinde en çok zaman ve bellek tüketen Gauss eliminasyon işlemlerini atlayarak deplasman vektörünü tahmin etmek üzerine kurulu bir mekanizmaya sahiptir. Bu, özellikle büyük yapılarda hesaplama süresini önemli ölçüde azaltır. YSA'nın tahminlerini optimizasyon sürecine entegre etmek için, üç yeni hibrit Biyocoğrafya Tabanlı Optimizasyon (BTO) arama yöntemi (YT-1, YT-2 ve YT-K) geliştirilmiştir. Bu yöntemler, YSA'nın tahminlerini stratejik noktalarda kullanarak optimizasyon sürecini yönlendirir ve daha hızlı yakınsama sağlamayı hedeflemesi için geliştirilmişlerdir.

Çalışma kapsamında gerçekleştirilen testler, farklı yapı tipleri üzerinde YSA tabanlı yöntemin FEM'e kıyasla hesaplama süresinde önemli bir azalma sağladığını göstermiştir. YSA'nın doğruluk seviyesi FEM'e ulaşmasa da, sunduğu hız avantajı, özellikle gerçek zamanlı uygulamalar veya hızlı tasarım yinelemeleri gerektiren senaryolarda önemli bir faktör haline gelir. Bu bulgular, YSA tabanlı yöntemlerin yapısal optimizasyon için uygulanabilir ve verimli bir alternatif sunduğunu ve daha ileri araştırmalar için umut vadeden bir yol açtığını göstermektedir.

2. KAYNAK TARAMASI

Yapısal optimizasyon, mühendislik tasarımının temel bir yönüdür ve performansı iyileştirilmiş, kaynak kullanımını en aza indirilmiş ve genel verimliliği artırılmış yapılar oluşturmayı amaçlar. Son yıllarda, yapay sinir ağları (YSA) ve optimizasyon algoritmalarının hızlı gelişimi, yapısal optimizasyona yeni yaklaşımlar ve teknikler getirmiştir. Bu çalışma, yapısal optimizasyon alanında yapılan son araştırmaları kapsamlı bir şekilde inceleyerek, geleneksel yöntemlerden YSA ve optimizasyon algoritmalarının entegrasyonuna kadar çeşitli yönleri ele almaktadır.

Bu kaynak taraması, üç ana alana odaklanmaktadır: Yapay Sinir Ağları (YSA) kullanılarak yapısal optimizasyon, Biyocoğrafya Tabanlı Optimizasyon (BTO) ve yapısal davranışın temsili model kullanımı. Bu alanlar, hesaplama verimliliğini artırma, optimum tasarım çözümlerini bulma ve belirsizlikleri etkili bir şekilde ele alma potansiyeline sahip oldukları için seçilmiştir. Her bölüm, bu tekniklerin yapısal optimizasyon problemlerine nasıl uygulandığını gösteren ilgili çalışmaların derinlemesine bir analizini sunmaktadır. Ayrıca, bu alanlardaki güncel eğilimler, zorluklar ve gelecekteki araştırma yönleri de tartışılacaktır.

2.1. YSA Kullanarak Yapısal Optimizasyon ile İlgili Yapılan Çalışmalar

Son yıllarda, yapısal optimizasyonda YSA'ların uygulanmasını araştıran çok sayıda çalışma yapılmıştır. Bu çalışmalar, YSA'ların yapısal analizleri simüle etme, optimum tasarımlar tahmin etme, belirsizlikleri ele alma ve güvenilirlik tabanlı optimizasyon problemlerini çözme gibi çeşitli görevlerde etkili olduğunu göstermektedir. YSA'lar, geleneksel yöntemlerle karşılaştırıldığında daha hızlı ve daha verimli çözümler sunarak karmaşık problemlerin üstesinden gelmede büyük potansiyel sergilemektedir.

YSA'ların yapısal optimizasyon süreçlerine entegrasyonu, hesaplama verimliliğinde önemli gelişmelere yol açmıştır. Optimizasyon sırasında YSA kullanmak, zaman alıcı sonlu eleman analizlerinin (FEA) sayısını azaltmanın etkili bir yoludur. Yapısal tepkinin bir modelini oluşturmak üzere eğitilmiş YSA, optimizasyon sürecinde pahalı hesaplama analizlerinin yerini alarak hesaplama süresini önemli ölçüde azaltabilir. Berke ve Hajela (1992), bu yaklaşımı beş ve on çubuklu kafeslerle yapılan çalışmalarıyla göstermişlerdir. Eğitilen YSA'lar, yapısal analizleri etkili bir şekilde simüle ederek kabul edilebilir doğrulukta optimum tasarımlara ulaşılmasını sağlamıştır.

YSA'lar ayrıca, deneyimli tasarımcıların bilgilerini yakalayıp doğrudan optimum tasarım tahminleri yapmak için kullanılabilir. Bu yaklaşımda, YSA belirli bir yapısal kavram yelpazesi içinde optimum tasarımlar oluşturmak için gerekli uzmanlığı edinmek üzere eğitilir. Eğitim sonrasında YSA, geleneksel optimizasyon yinelemelerine gerek kalmadan optimum tasarımları tahmin ederek tasarım sürecini hızlandırabilir. Berke ve Hajela (1992) tarafından yapılan çalışmalar, bu yaklaşımın kafes tasarımlarında başarılı bir şekilde uygulanabileceğini göstermiştir. Çalışmada geliştirilen YSA'lar, verilen tasarım kısıtlamaları için optimum kafes tasarımlarını tahmin etmede uzman bir tasarımcının uzmanlığını yakından taklit edebilmiştir.

Güvenilirlik tabanlı yapısal optimizasyon, yapısal sistemin belirsizlik altında başarısızlık olasılığını dikkate alan kritik bir alandır. YSA'lar, bu alanda da etkili çözümler

sunmaktadır. (Papadrakakis ve Lagaros 2002), güvenilirlik tabanlı yapısal optimizasyonda YSA'ların kullanımını araştıran kapsamlı bir çalışma gerçekleştirmiştir. Çalışmaları, YSA'ların hem deterministik hem de olasılıksal kısıtlama kontrollerini gerçekleştirmek veya Monte Carlo simülasyonunda (MCS) yer alan yapısal analizleri tahmin etmek üzere eğitilebileceğini göstermiştir. YSA tabanlı yaklaşım, geleneksel yöntemlere kıyasla hesaplama süresini önemli ölçüde azaltarak daha büyük ve daha karmaşık yapıların optimizasyonunu mümkün kılmıştır.

YSA'ların yapısal optimizasyondaki gücü, belirsiz yük konumları ve malzeme özellikleri altında topoloji optimizasyonu için kullanılan yeni bir yöntemde açıkça görülmektedir (Freitag vd. 2022). Bu çalışmada, hesaplama maliyetini azaltmak ve tasarım keşfini hızlandırmak için sonlu eleman simülasyonları yerine YSA'lar kullanılmıştır. Yapısal stres ve yer değiştirmeleri tahmin etmek üzere çeşitli YSA'lar eğitilirken, ek bir YSA, belirli bir hacim oranı için optimum malzeme yoğunluğu dağılımını tahmin etmek üzere eğitilmiştir. Bu yaklaşım, gerçek dünya uygulamaları için kritik bir faktör olan belirsizliklere karşı dayanıklı topolojiler üretmiştir.

(Iranmanesh ve Kaveh 1999) , yapısal optimizasyon için gradyan tabanlı sinir ağları kullanarak yenilikçi bir strateji sunmuştur. Bu stratejide, kısıtlamalar ve gradyanlar, Kohonen kendi kendine organize haritası ve Grossberg yıldızları algoritmalarını birleştiren geliştirilmiş bir karşı yayılım ağı (CPN) kullanılarak tahmin edilir. CPN, hızlı ve kolay bir öğrenme süreci sağlar ve doğruluğu artırmak için bir interpolasyon modu ile geliştirilmiştir. Sinir ağları ve optimizasyon algoritmalarının birleştirilmesiyle oluşan "nöro-optimize edici" strateji hem düzlemsel hem de uzaysal kafes yapıların optimizasyonunda yüksek doğrulukta sonuçlar üreterek hesaplama süresini önemli ölçüde azaltmaktadır.

(Papadrakakis vd. 1998) çalışmaları, yapısal optimizasyon problemlerinde evrim stratejileri (ES) ve YSA'ların birlikte kullanılmasının etkinliğini vurgulamaktadır. Araştırmacılar, ES optimizasyon sürecinde tekrarlanan yapısal analizlerin zaman alıcı etkisini azaltmak için YSA'lar kullanmışlardır. Seçilen analizlerden elde edilen veriler kullanılarak bir geri yayılım algoritması ile YSA eğitilmiştir. Eğitilen YSA, amaç ve kısıt fonksiyonlarının değerlerini kabul edilebilir bir doğrulukla tahmin etmek için kullanılmış, böylece optimizasyon süreci önemli ölçüde hızlandırılmıştır. Bu yaklaşım, ES gibi olasılıksal arama algoritmalarının uygulanabilirliğini artırarak onları verimli genel amaçlı optimizasyon araçlarına dönüştürmektedir.

Sunulan çalışmalar, yapay sinir ağlarının (YSA) yapısal optimizasyona yeni olanaklar sunduklarını, yapısal tasarım ve optimizasyon gibi kompleks problemler için oldukça uygun özelliklere sahip olduklarını göstermektedir.

2.2. Yapısal Optimizasyonda BTO (Biyocoğrafya Tabanlı Optimizasyon) Kullanımı ile İlgili Çalışmalar

Doğanın optimizasyon prensiplerinden esinlenen Biyocoğrafya Tabanlı Optimizasyon (BTO), son yıllarda popülerlik kazanan nispeten yeni bir meta-sezgisel optimizasyon algoritmasıdır. Algoritma, hayvan türlerinin göç ve yerleşim modellerini taklit ederek optimum çözümler bulmayı amaçlar. BTO, karmaşık, doğrusal olmayan ve çok modlu optimizasyon problemlerinde geleneksel yöntemlere göre avantajlar sunar.

BTO algoritması, yapısal optimizasyon alanında çeşitli problemleri ele almak için başarıyla kullanılmıştır. (Aydogdu ve Akin 2015), betonarme istinat duvarı tasarımlarının maliyetini ve CO₂ emisyonunu en aza indirmek için BTO algoritmasının kullanılmasını araştırmışlardır. Çalışmaları, BTO'nun betonarme istinat duvarlarının maliyet optimizasyonu için optimum çözümü bulmada etkili ve sağlam olduğunu göstermiştir. Çalışma ayrıca, hem CO₂ emisyonunu en aza indirme hem de maliyet en aza indirme hedeflerini göz önünde bulundurarak istinat duvarı tasarımlarının çevresel etkisini vurgulamıştır. Sonuçlar, BTO algoritmasının, istinat duvarı tasarımlarının çevresel etkisini azaltmada umut vadeden bir araç olduğunu göstermiştir.

(Aydogdu 2017), BTO algoritmasını Lévy uçuş dağılımıyla geliştirerek (LFBBO) betonarme konsol istinat duvarlarının sismik yük altında optimum tasarımını araştırmıştır. Lévy uçuşları, optimizasyon probleminin arama alanını daha etkili bir şekilde keşfetmesini sağlayarak algoritmanın performansını artırır. Çalışma, LFBBO algoritmasının geleneksel BTO algoritmasından ve literatürdeki diğer optimizasyon yöntemlerinden daha düşük maliyetli tasarımlar ürettiğini göstermiştir. Bu, LFBBO algoritmasının karmaşık yapısal optimizasyon problemlerine yönelik etkinliğini ve sağlamlığını vurgulamaktadır. Çalışma ayrıca, yüksek sismik aktiviteye sahip bölgelerde istinat duvarlarının tasarımında sismik tehlike analizi yapmanın önemli maliyet avantajları sağlayabileceğini göstermiştir.

Sunulan bu çalışmalar doğrultusunda BTO'nun etkinliğini ve sağlamlığını kanıtlanmaktadır. Bu nedenlerle, BTO algoritması tez çalışması için uygun bir meta-sezgisel yöntem olarak seçilmiştir.

2.3. Yapısal Optimizasyonunda Yapı Davranışının Temsili Model Kullanımı ile İlgili Çalışmalar

Yapısal optimizasyon problemleri genellikle tekrar tekrar yapısal analizler yapılmasını gerektirir, bu da hesaplama açısından yoğun olabilir ve optimizasyon sürecini yavaşlatabilir. Bu zorluğun üstesinden gelmek için yapı davranışını temsil eden yaklaşık modeller kullanılabilir. Bu modeller, optimizasyon sürecinde karmaşık ve zaman alıcı yapısal analizlerin yerini alarak hesaplama verimliliğini önemli ölçüde artırabilir.

YSA, yapı davranışını temsil etmek için kullanılabilen güçlü bir modelleme aracıdır. Karmaşık ilişkileri öğrenme ve genelleme yetenekleri sayesinde, yapısal tepkiyi yüksek doğrulukla simüle etmek için kullanılabilirler. (Salajegheh ve Gholizadeh 2005), yapısal analizi yaklaşık olarak hesaplamak için YSA kullanan iyileştirilmiş bir genetik algoritma (GA) ile optimum yapı tasarımı üzerine bir çalışma gerçekleştirmiştir. YSA'nın kullanımı, standart GA'ya kıyasla hesaplama yükünü önemli ölçüde azaltmış ve büyük ölçekli yapıların optimizasyonunu daha uygulanabilir hale getirmiştir.

(Papadrakakis vd. 1998), yapısal optimizasyonda evrim stratejilerini ve YSA'ları araştırmışlardır. Çalışmalarında, tekrarlanan yapısal analizlerin zaman alıcı etkisini azaltmak için evrim stratejilerinin optimizasyon süreci içinde YSA kullanmışlardır. Seçilen analizlerden elde edilen veriler kullanılarak bir geri yayılım algoritması ile YSA eğitilmiştir. Eğitilen YSA, amaç ve kısıt fonksiyonlarının değerlerini kabul edilebilir bir doğrulukla tahmin etmek için kullanılmış, böylece optimizasyon süreci hızlandırılmıştır.

Bu yaklaşım, evrim stratejileri gibi olasılıksal arama algoritmalarının uygulanabilirliğini artırarak onları verimli genel amaçlı optimizasyon araçlarına dönüştürmektedir.

Sonuç olarak, YSA'lar ve diğer temsili modeller, yapısal optimizasyonda önemli bir rol oynamaktadır. Tasarım sürecini hızlandırabilir, belirsizlikleri ele alabilir ve daha verimli ve sağlam çözümler üretebilirler. Bu alandaki devam eden araştırmalar, daha da gelişmiş yöntemlere ve uygulamalara yol açarak yapısal optimizasyonun geleceğini şekillendirecektir.

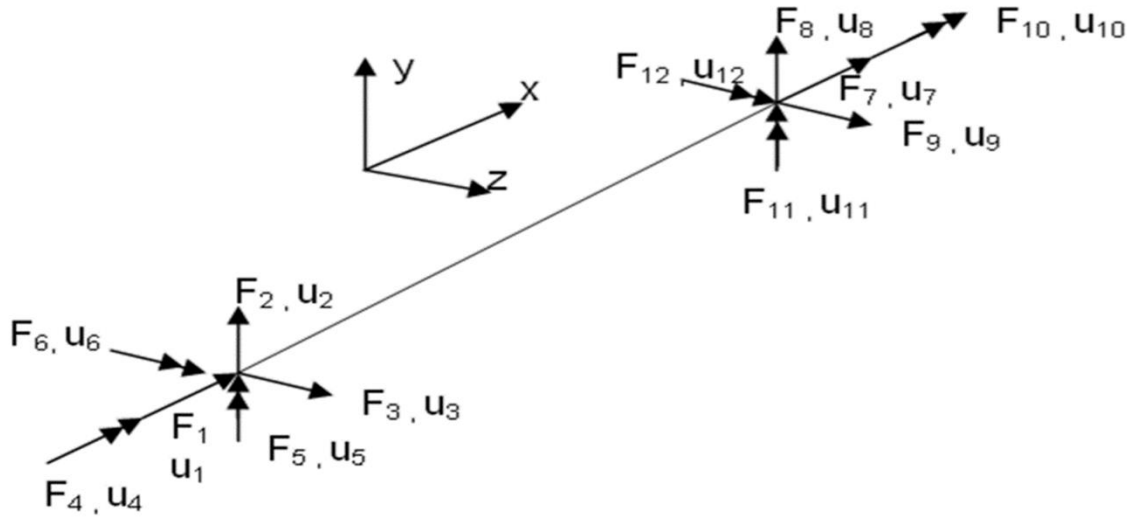


3. MATERYAL VE METOT

3.1. Çerçeve ve Kafes Sistemlerinin Analizi (FEM Metodu)

Çerçeve sistemler kolon, kiriş gibi tek boyutlu yapı elemanlarından oluşan inşaat mühendisliğinde kullanılan taşıyıcı sistem türlerinden biridir. Çerçeve sistemlerin birden çok kiriş ve kolondan oluşması, bu taşıyıcı sisteminin analizi esnasında dış etki ve deplasman arasındaki ilişkinin FEM ile doğrusal denklem sistemi şeklinde tanımlanmasına imkân vermiştir. Bu tip çözüm metodlarının genel adı Sonlu Elemanlar Metodu, İngilizce terim adı ise Finite Element Method (FEM) ‘dur. FEM analizlerinde birçok doğrusal denklemin çözümü gerekmektedir. Bu denklemler kendi içerisinde adımlardan oluşmakta ve bu adımlar da birçok aritmetik işlemten oluşmaktadır. Bir problemi sonlu sayıdaki birçok küçük parçaya bölüp çözme metodolojisi sadece yapı mühendisliğinde değil, birçok farklı bilim dalında da kullanılmaktadır.

Birimden bütüne ulaşarak sonuca varmayı amaçlayan bu yaklaşımda temel düşünce, çerçeveyi oluşturan her bir elemana etki eden yükleri ve bu yükler doğrultusunda oluşan deplasmanları belirlemektir. Çerçeve sistemler üzerinde bu yaklaşım incelendiğinde, sistemin tek bir elemanı üzerinde oluşan yükler ve uzamaların 3 boyutlu (3B) ekseninde Şekil 3.1’deki gibi oluştuğu görülür.



Şekil 3.1 Bir çubuk üzerindeki düğüm uçlarında oluşan kuvvetler, momentler ve deplasmanlar (Aydogdu 2017).

Burada $\{F_k\} = \{F_1, F_2, F_3, F_4, F_5, F_6\}$ çubuğun ilk düğüm noktasında oluşan yüklerin vektörü, $\{u_k\} = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ ise bu düğüm noktasında oluşan deplasmanların vektörüdür. F_1, F_2 ve F_3 kuvvetleri temsil ederken, F_4, F_5 ve F_6 ise momentleri temsil etmektedir. Sistemdeki her bir elemanın uç noktalarındaki yükler ile yine uç noktalarındaki deplasmanlar arasındaki bağlantı aşağıdaki gibidir.

$$\{f_i\} = [k_i]\{u_i\} \quad (3.1)$$

Burada $[k_i]$ i numaralı çerçeve elemanının lokal eksenindeki rijitlik matrisini ifade eder. Rijitlik matrisi 3 Boyutlu (3B) çerçeve elemanları için 12 kolon ve 12 satırdan oluşmaktadır (Şekil 3.2). 3B çerçeve matrisinin elde edilmesi ile ilgili ayrıntılar referans çalışmasında verilmiştir (Aydogdu 2017).

$$[k] = \begin{bmatrix} \frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{\ell^3} & 0 & 0 & 0 & \frac{6EI_z}{\ell^2} & 0 & -\frac{12EI_z}{\ell^3} & 0 & 0 & 0 & \frac{6EI_z}{\ell^2} \\ 0 & 0 & \frac{12EI_y}{\ell^3} & 0 & -\frac{6EI_y}{\ell^2} & 0 & 0 & 0 & -\frac{12EI_y}{\ell^3} & 0 & -\frac{6EI_y}{\ell^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{\ell} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{\ell} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{\ell^2} & 0 & \frac{4EI_y}{\ell} & 0 & 0 & 0 & \frac{6EI_y}{\ell^2} & 0 & \frac{2EI_y}{\ell} & 0 \\ 0 & \frac{6EI_z}{\ell^2} & 0 & 0 & 0 & \frac{4EI_z}{\ell} & 0 & -\frac{6EI_z}{\ell^2} & 0 & 0 & 0 & \frac{2EI_z}{\ell} \\ -\frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{\ell^3} & 0 & 0 & 0 & -\frac{6EI_z}{\ell^2} & 0 & \frac{12EI_z}{\ell^3} & 0 & 0 & 0 & -\frac{6EI_z}{\ell^2} \\ 0 & 0 & -\frac{12EI_y}{\ell^3} & 0 & \frac{6EI_y}{\ell^2} & 0 & 0 & 0 & \frac{12EI_y}{\ell^3} & 0 & \frac{6EI_y}{\ell^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{\ell} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{\ell} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{\ell^2} & 0 & \frac{2EI_y}{\ell} & 0 & 0 & 0 & \frac{6EI_y}{\ell^2} & 0 & \frac{4EI_y}{\ell} & 0 \\ 0 & \frac{6EI_z}{\ell^2} & 0 & 0 & 0 & \frac{2EI_z}{\ell} & 0 & -\frac{6EI_z}{\ell^2} & 0 & 0 & 0 & \frac{4EI_z}{\ell} \end{bmatrix}$$

Şekil 3.2 Rijitlik matrisi

Çerçeve elemanların lokal rijitlik matrislerinin belirlenmesinden sonra da eleman rijitlik matrisleri denklem (3.2) yardımıyla global eksene göre dönüştürülür.

$$[K_i] = [B_i]^T [k_i] [B_i] \quad (3.2)$$

Burada $[K_i]$ i numaralı çerçeve elemanının global eksenindeki rijitlik matrisini, $[B_i]$ dönüşüm matrisini temsil eder. $[B_i]$ 'nin genel görünümü Şekil 3.3'te verilmiştir.

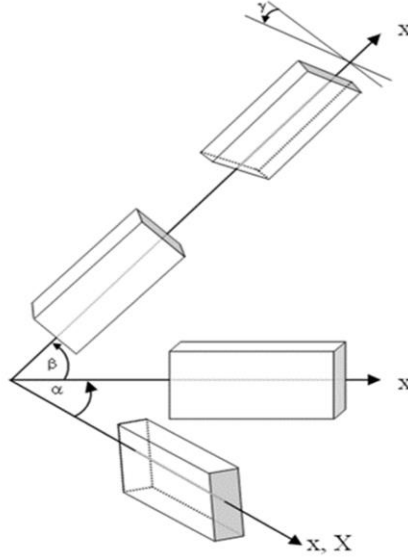
$$[B] = \begin{bmatrix} b_{11} & b_{12} & b_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{21} & b_{22} & b_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{31} & b_{32} & b_{33} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{11} & b_{12} & b_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{21} & b_{22} & b_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{31} & b_{32} & b_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{11} & b_{12} & b_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{21} & b_{22} & b_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{31} & b_{32} & b_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{11} & b_{12} & b_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{21} & b_{22} & b_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Şekil 3.3 Çerçeve elemanının her iki ucunu da tüm serbestlik derecelerinde temsil eden koordinat transformasyon matrisi

$b_{i,j}$, $i = 1,2,3$ ve $j = 1,2,3$ ifadelerinin açılımı 3×3 matris $([b])$ şeklinde denklem (3.3)'te verilmiştir.

$$[b] = \begin{bmatrix} \cos \beta \cdot \cos \alpha & \sin \beta & -\cos \beta \cdot \sin \alpha \\ \sin \alpha \cdot \sin \gamma - \cos \alpha \cdot \sin \beta \cdot \cos \gamma & \cos \beta \cdot \cos \gamma & \cos \alpha \cdot \sin \gamma + \sin \alpha \cdot \sin \beta \cdot \cos \gamma \\ \sin \alpha \cdot \cos \gamma + \cos \alpha \cdot \sin \beta \cdot \sin \gamma & -\cos \beta \cdot \sin \gamma & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma \end{bmatrix} \quad (3.3)$$

Burada α , β ve γ sırasıyla lokal ve global y Y, z Y ve x Y eksenleri arasındaki açıları temsil etmektedir. α , β ve γ açılarının görsel temsili içerisinde gerçekleştirilmiştir.



Şekil 3.4 Lokal eksen ile global eksen arasındaki koordinat transformasyonu (Aydogdu 2017)

Bu aşamadan sonra eleman rijitlik matrisleri serbestlik derecelerine göre toplanarak sistem rijitlik matrisi (K_{sys}) aşağıdaki şekilde elde edilir.

$$[K_{sys}] = \sum_{i=1}^m ([B_i]^T [k_i] [B_i]) \quad (3.4)$$

Burada n, çerçeve sistemindeki toplam taşıyıcı eleman sayısını temsil etmektedir. Bu aşamadan sonra sistem yük vektörü (P_{sys}) elde edilir. P_{sys} yapı sistemine etki eden dış yüklerinin yapının serbestlik derecelerine karşılığıdır. Yük doğrudan yapının düğüm noktasına etki ederse dış yük P_{sys} vektörüne eklenir. Eğer doğrudan düğüm noktasına etki etmeyen yük var ise bu yükler eleman uç noktalarına bir başka değişle düğüm noktalarına aktarılır. Bu işlem aşağıdaki ifade ile gösterilebilir.

$$\begin{aligned} \text{Eğer } i \in \text{Düğüm ise } \{P_{sys}\} &= \{P_{sys}\} + \{P_i\} & i = 1, 2, \dots, m \\ \text{Eğer } i \notin \text{Düğüm ise } \{P_i\} &\rightarrow \{YK_i\}; \{P_{sys}\} &= \{P_{sys}\} + \{YK_i\} \end{aligned} \quad (3.5)$$

Bu aşamadan sonra rijitlik ve yük ifadeleri sistem uzayına aktarılmıştır. Denklem (3.1)'deki kuvvet deplasman arasındaki bağıntı lokal eksen, global eksen veya sistem

düzeyinden bağımsızdır. Bu nedenle denklemi sistem düzeyine uyarlayarak aşağıdaki denklem elde edilir.

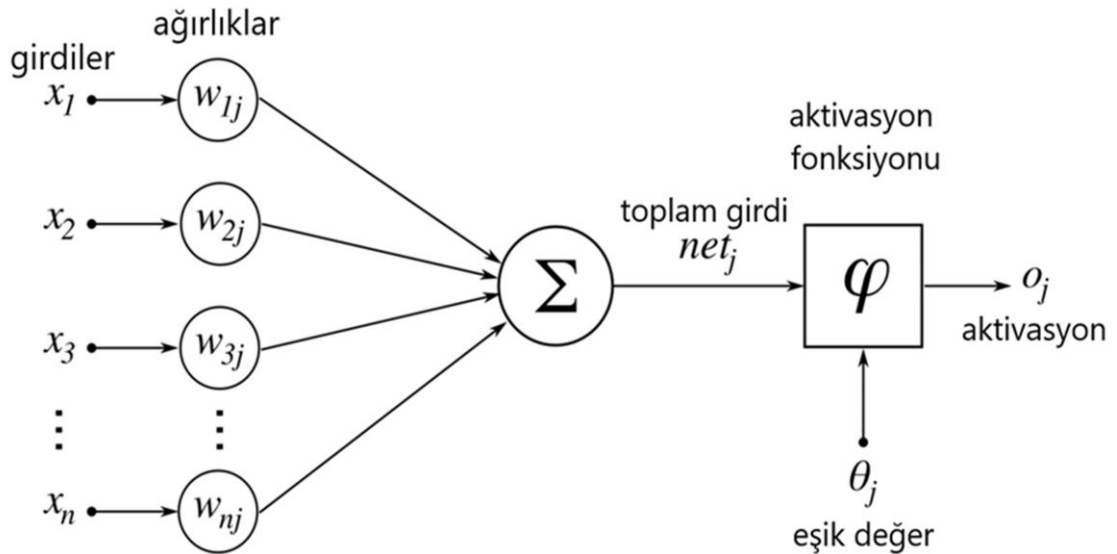
$$\{P_{sys}\} = [K_{sys}]\{d_{sys}\} \quad (3.6)$$

Yukarıdaki denklemde $\{d_{sys}\}$ yapı sisteminde her bir serbestlik derecesine gelen deplasman ve dönme değerlerini veren sistem deplasman vektörüdür. Denklem (3.6) ifadesi doğrusal denklem sistemini temsil etmektedir. Bu nedenle Gauss Eliminasyon gibi sayısal analiz teknikleri ile denklem (3.6) çözülerek $\{d_{sys}\}$ vektörü elde edilir.

Sistem deplasmanları bulunduktan sonra $\{d_{sys}\}$ vektörü her bir çerçeve elemanına aktarılır ve her bir elemanın global koordinatlardaki deplasman vektörü elde edilir ($\{U_i\}$). Aktarma işlemi yine serbestlik derecesine yapılır ancak işlem yönü sistemden global eksene doğrudur (Örmecioğlu 2019).

3.2. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), biyolojik sinir sisteminden esinlenerek geliştirilen, makine öğrenmesinin önemli bir dalıdır (Haykin 1998). Bu algoritma türü, birçok veri setlerinden anlamlı bir bilgi çıkarmak ve nesnelerin ve olayların içindeki gizli örüntüleri bulabilmek için kullanılır. Temel olarak, YSA, matematiksel işlemleri gerçekleştiren yapay nöronlardan oluşan bir ağ yapısıdır (Murphy 2012).

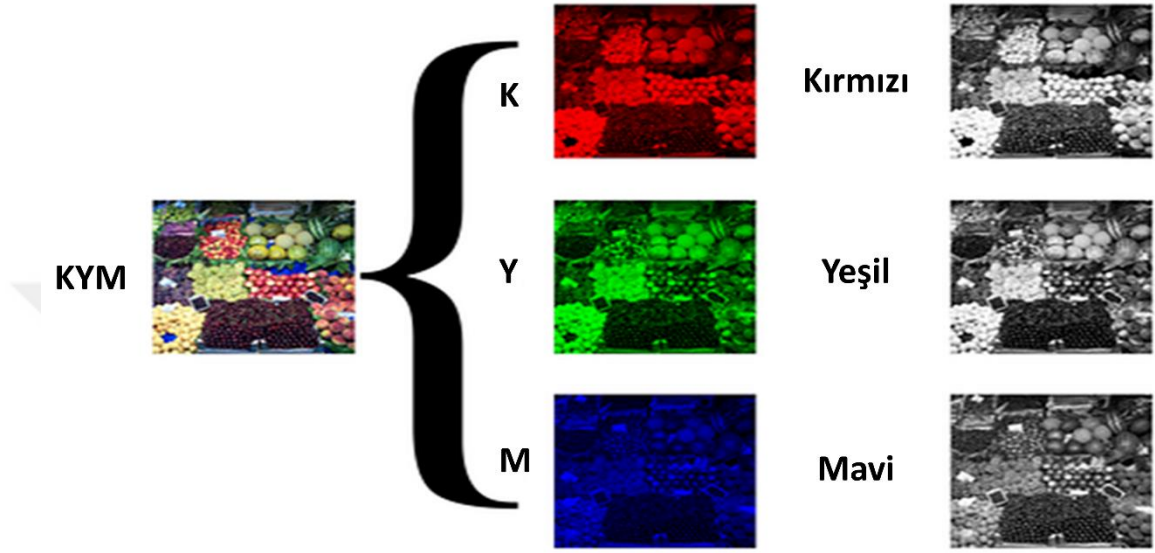


Şekil 3.5 Renkli bir resimin üç kanallı bir tensör olarak imgenilmiş hali (Anonymous 1 2024)

Yapay sinir ağları, sinir hücrelerinin, yani nöronların biyolojik sinir sisteminden ilham alınarak geliştirilmiştir (Şekil 3.5). Her bir yapay nöron, girdi verilerini alır, bu verileri ağırlıklarla çarpar, bir aktivasyon fonksiyonunu uygular ve sonuç olarak bir çıkış üretir. Ağ üzerindeki nöronlar, farklı katmanlarda düzenlenir ve her katmanındaki nöronlar, alttaki katmandan gelen çıktıları alır ve üstteki katmana aktarır. Girdi katmanından çıktı katmanına doğru bu bilgi akışı gerçekleşirken, ara katmanlarda verilerin işlenmesi ve

öğrenme süreci gerçekleştirilir. Aslında YSA'ların eğitimi, bu nöronlardaki ağırlıkların belirlenmesi sürecinden ibarettir.

Tensörler, YSA'da verilerin temel yapı taşlarıdır. Tensörler, vektörlerin ve matrislerin genelleştirilmiş hali olarak düşünülebilir ve verileri temsil etmek için kullanılırlar. Örneğin, gri tonlamalı bir resim bir matris olarak ifade edilebilir ve renkli bir resim, üç kanallı bir tensör olarak düşünülebilir.



Şekil 3.6 Renkli bir resmin üç kanallı bir tensör olarak imgenenmiş hali (Anonymous 2024)

Yapay sinir ağlarının başarısının arkasındaki temel unsur, derinlik yani çok katmanlı yapılarıdır. "Derin Öğrenme" terimi de buradan gelmektedir. Yapay sinir ağları, çok sayıda ara katman veya gizli katman içerebilir ve bu katmanlar, verilerin karmaşık örüntülerini öğrenme yeteneğine sahiptir (Mcculloch ve Pitts 1943).

Kayıp fonksiyonları, yapay sinir ağlarının öğrenme sürecinin en önemli bileşenlerinden biridir. Doğru kayıp fonksiyonunun seçilmesi, ağın performansını önemli ölçüde etkileyebilir (Bishop 1995). Yapay sinir ağları içerisinde, belirli problemleri çözmek için tasarlanmış kayıp fonksiyonları kullanılır. Kayıp fonksiyonları, ağın tahminlerini gerçek etiketlerle karşılaştırarak ağın performansını ölçer. Ölçülen bu performans değerlerine göre ağın hatalarını minimize etmek ve daha iyi sonuçlar elde etmek için eğitim sürecinde bu kayıp fonksiyonları sürekli kullanılırlar.

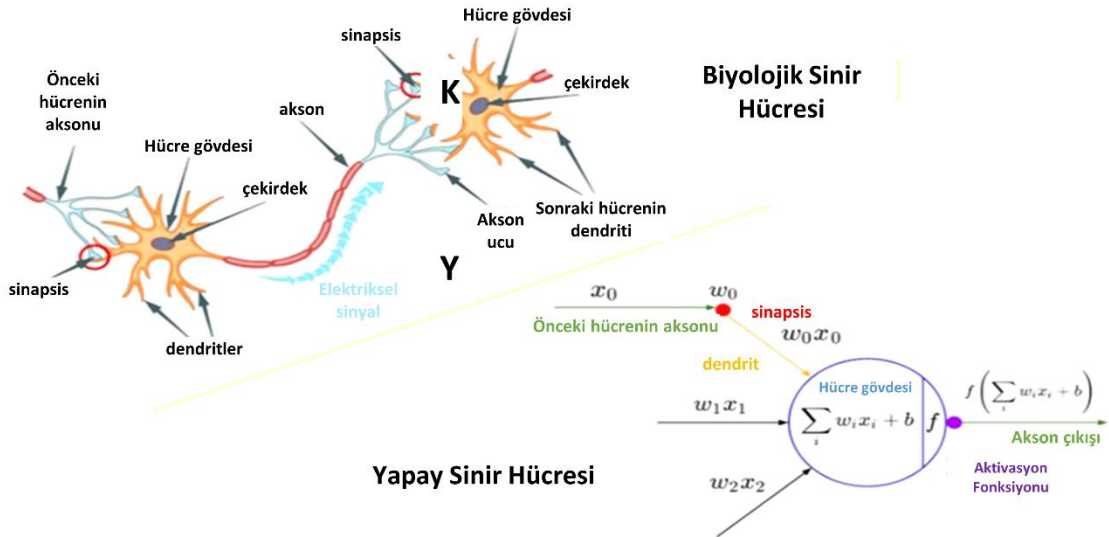
Öğrenme süreci (fitting), YSA'nın eğitildiği aşamadır. Öğrenme süreci, YSA'nın çıktılarını, beklenen çıktılarla karşılaştırarak ve hataları azaltmak için ağırlıkları ayarlayarak gerçekleştirilir." (Bishop 1995). Ağ, eğitim verileriyle beslenir ve hataların azaltılması için ağırlıklar düzenlenir. Bu süreç, Geriye doğru yayılım (backpropagation) adı verilen algoritma sayesinde gerçekleştirilir. Geriye doğru yayılım algoritması, kayıp fonksiyonunu minimize etmek için ağırlıkları ayarlayarak ağın öğrenme sürecini gerçekleştirir.

Sonuç olarak, yapay sinir ağları, yüksek miktarda veri işleme kapasitesine sahip, öğrenme ve genelleştirme yeteneği olan güçlü algoritmalar olarak bilinirler. Bu nedenle, birçok farklı alanda, görüntü işlemeden doğal dil işlemeye, çeşitli strateji oyunları oynamaktan otomasyona kadar birçok karmaşık problemi çözmek için başarıyla kullanılmaktadır.

3.3. Biyolojik Sinir Ağlarının Yapısı

Yapay Sinir Ağları, biyolojik sinir ağlarının temel işleyiş prensiplerini taklit ederek bilgi işleme yeteneklerini simüle eden önemli bir hesaplama modelini temsil eder. Bu ağların temel yapı taşları, biyolojik nöronların işlevselliğini yansıtan yapay nöronlardır. Biyolojik örneklerine dayanarak, yapay nöronlar da karmaşık bir morfoloji sergiler ve bilgi işleme işlevini gerçekleştirirler. Camillo Golgi ve Santiago Ramón y Cajal 1906 yılında yaptıkları ayrıntılı mikroskopik incelemeler sonucunda nöronların dendritler, hücre gövdesi ve akson gibi temel bileşenlere sahip olduğu anlaşılmıştır. Dendritler, çevresel kaynaklardan gelen veriyi toplar ve bu verileri hücre gövdesine ileterek bütünleştirme işlevini yerine getirir. Aksonlar ise uzun ve özellikle iletim odaklı uzantılardır; diğer nöronlara veya hücrelere bilgi iletimini kolaylaştırırlar (Shepherd vd. 2004).

Yapay sinir ağlarındaki yapay nöronlar da benzer bir düzeni takip eder. Girdi sinyalleri, yapay nöronun dendritleri tarafından alınır, hücre gövdesinde işlenir ve sonuç olarak elde edilen sinyal, akson boyunca iletilir. Yapay nöronlar arasındaki bağlantılar, sinaps adı verilen ağırlıklar aracılığıyla temsil edilir ve bu ağırlıkların değerleri, iletişim gücünü ve etkisini yansıtır. Bu biyolojik ilhamlı yapı, yapay sinir ağlarının öğrenme yeteneklerini gerçekleştirmesine olanak tanır. Öğrenme, sinaps ağırlıklarının uygun şekillerde ayarlanması yoluyla yapay nöronların veriye uyum sağlamasını ve yeni bilgiye cevap verme yeteneklerini kazanmasını içerir.



Şekil 3.7 Yapay nöron ve biyolojik nöron karşılaştırması (Anonymous 3 2024)

3.4. Yapay Sinir Ağlarının Temel Yapısı

3.4.1. Ağ katmanları

Çalışma sırasında, tasarımı denenen ilk yapay sinir ağı modelleri sınıflandırma ve girdilerin özelliklerini tanımlama kabiliyetlerinden ötürü CNN tabanlı modellerden seçilmiştir. Seçilen modeller içerisinde Conv2D, MaxPooling, Dropout, LeakyReLU, Flatten ve Dense katmanlarının kombinasyonları kullanılarak inşa edilmiştir. Bu katmanların çalışma mantıklarına aşağıda kısaca değinilmiştir.

Tensör Kavramı:

Tensör, çok boyutlu veri yapılarını temsil eden ve matematiksel işlemlere tabi tutulabilen bir kavramdır (Tucker 1966). Tensörler, matrislerin genelleştirilmiş bir versiyonu olarak düşünülebilir. Matematiksel olarak, bir tensör, sıralı bir şekilde düzenlenmiş elemanların bir koleksiyonunu temsil eden çok boyutlu bir dizi veya matristir.

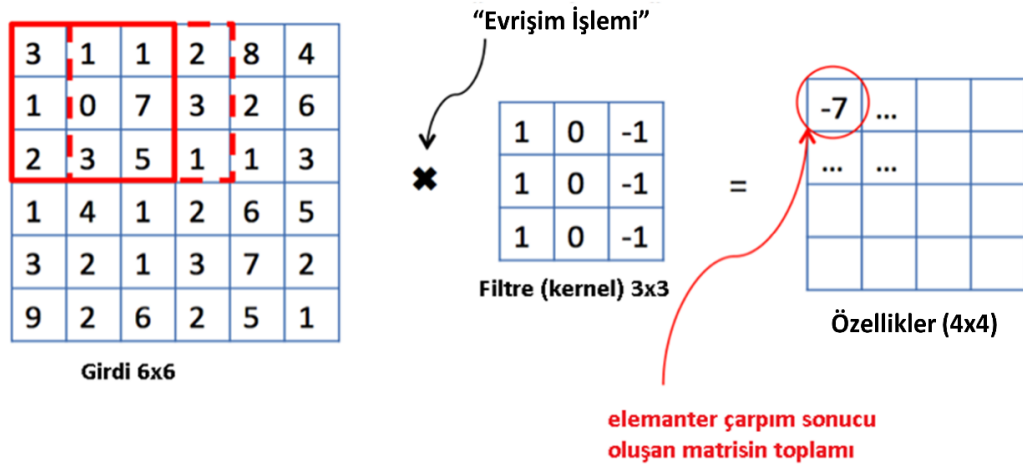
Tensörler, içerisinde veri barındıran birçok alanda veri tanımlaması ve işlemesi için yaygın olarak kullanılır. Özellikle, derin öğrenme ve yapay zekâ alanında, tensörlerin kullanımı önemlidir. Çünkü görüntüler, metinler, ses verileri gibi karmaşık ve çok boyutlu verileri temsil etmek için tensörler kullanılır (Goodfellow vd. 2016). Tensörlerin matematiksel işlemleri, lineer matematikte kullanılan işlemlere benzer. Bu işlemler arasında tensör toplama, tensör çarpma, tensör transpozisyonu ve tensörleme bulunur. Bu işlemler, tensörlerin boyutlarına ve yapılarına göre tanımlanır ve tensörler arasında bilgi alışverişini mümkün kılar.

Örneğin, bir 3D tensörü bir görüntünün piksellerini temsil edildiği varsayalım. Her piksel, üç renk kanalının (kırmızı, yeşil, mavi) değerlerinden oluşur. Bu 3D tensör üzerinde gerçekleştirilebilecek işlemler arasında görüntüleri birleştirme, kesme, yeniden boyutlandırma gibi işlemler bulunur. Tensörler, matematiksel olarak ifade edilirken, genellikle indeksleme ve tensör cebri kullanılır. İndeksleme, tensörün elemanlarına erişmek için kullanılan bir yöntemdir. Örneğin, $T[i, j, k]$ şeklinde bir tensörün i, j, k indeksli elemanına erişilebilir. Tensör cebri, tensörler arasında işlemleri gerçekleştirmek için kullanılan matematiksel kurallar ve formüllerden oluşur.

Tensörlerin kullanımı, büyük veri setlerinde ve karmaşık yapıları olan verilerde etkili bir şekilde bilgi temsilini ve işlemlerini sağlar. Derin öğrenme ve yapay zekâ gibi alanlarda tensörlerin kullanımı, model eğitimi ve tahmin yapma süreçlerinde önemli bir rol oynar.

Conv2D Katmanı:

Conv2D, evrişimli (convolution) sinir ağlarında yaygın olarak kullanılan bir katmandır ve görüntü işleme, nesne tanıma, doğal dil işleme gibi birçok görevde etkilidir. Bu katman, verilen bir girdi görüntüsünü veya tensörünü alır ve onu daha düşük boyutlu bir özellik haritasına dönüştürür. Conv2D'nin çalışma prensibi, evrişim operasyonuna dayanır. Evrişim, bir filtre matrisinin (ağırlık matrisi) girdi üzerinde kaydırılması ve nokta çarpımı işlemlerinin uygulanmasıyla gerçekleştirilir. Bu işlem, her bir filtre matrisinin girdi üzerinde tarama yapması ve her konumda nokta çarpımı işlemini gerçekleştirmesiyle tekrarlanır. Nokta çarpımı sonucunda elde edilen değerler, çıktı özellik haritasının piksellerini oluşturur (Lecun vd. 1989).



Şekil 3.8 Conv2D Katmanı (Anonim 4 2024)

Conv2D katmanı, filtre matrislerinin öğrenilebilir ağırlıklarını kullanarak girdi üzerinde farklı özelliklerin algılanmasını sağlar. Her filtre, belirli bir özelliği yakalamak için optimize edilir. Bu şekilde, birden fazla filtre kullanılarak girdi üzerinde farklı özelliklerin çıkarılması ve temsil edilmesi sağlanır.

Matematiksel olarak, Conv2D'nin çıktı değeri, girdi tensörü ile filtre matrisinin evrişimi ve bir bias terimi eklenmesiyle hesaplanır (Goodfellow vd. 2016). Hesaplamanın mevcut aşamasında, evrişim işlemi, giriş tensörlerinin her bir bileşeni üzerinde bir nokta çarpımı gerçekleştirme eylemi yoluyla yürütülür. Bu şekilde elde edilen sonuç, çıktı özellik haritasının piksellerini oluşturur. Conv2D katmanı, aynı zamanda parametrelerin (ağırlıkların) otomatik olarak alınmasını kolaylaştırırken, evrişim işlemini yinelemeli olarak gerçekleştirerek özellik haritaları oluşturur. Bu edinme işlemi, gerçek veri örneklerinde kayıp fonksiyonunun en aza indirilmesi ile gerçekleştirilir. Sonuç olarak, Conv2D katmanı, verilerdeki özellikleri tespit etmeyi ve temsil etmeyi amaçlayan kişiselleştirilmiş filtrelerin otomatik olarak alınmasını sağlar.

Pooling Katmanı:

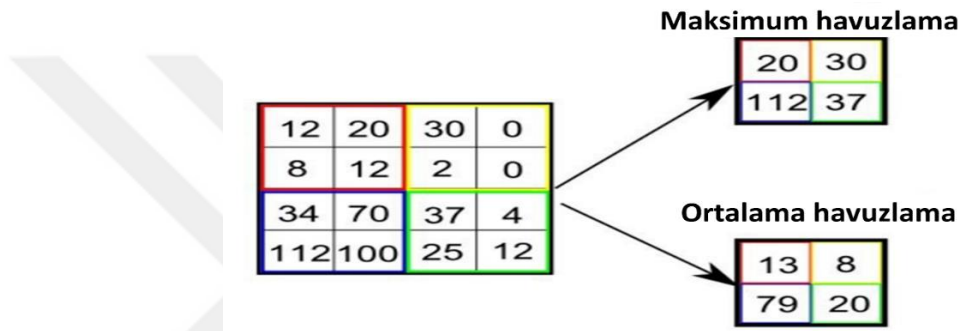
Pooling (Havuzlama), derin öğrenme modellerinde sıkça kullanılan bir işlem olan örnekleme (subsampling) tekniğidir. Örnekleme, girdi verisinin boyutunu küçültmek ve özellik girdi matrisindeki önemli özellikleri vurgulamak amacıyla kullanılır. Genellikle CNN sinir ağlarında kullanılan pooling, ağız öğrenme kapasitesini artırmak ve hesaplama yükünü azaltmak için etkili bir şekilde uygulanır. Ayrıca Pooling, CNN'lerin karar verme sürecini stabilize etmeye yardımcı olur, çünkü girdi verisinin küçük değişikliklerine karşı daha az hassastır (Simonyan ve Zisserman 2014).

Pooling işlemi, girdi verisini küçük bölgelere böler ve her bölgedeki özelliklerin bir özetini alır. Bu özetleme işlemi, belirli bir bölgeye ait en büyük değeri (max pooling) veya ortalama değeri (average pooling) kullanarak gerçekleştirilebilir. Max pooling, özelliklerin en belirgin olanını vurgularken, average pooling ise özelliklerin ortalamasını alarak daha genel bir özet sunar. Formel olarak, max pooling işlemi bir girdi özellik haritasını $N \times N$ boyutundaki bir pencere ile tarar ve her pencere için maksimum değeri alır. Bu işlem, aşağıdaki formülle gösterilebilir:

$$y[i,j] = \max (x[(i-1)N:N_i, (j-1)N:N_j]) \quad (3.7)$$

Burada, x girdi özellik haritasını, y ise çıktı özellik haritasını temsil eder. N pencere boyutunu belirtir.

Pooling işlemi, ağı boyutunu azaltırken özelliklerin kaybını minimize etmek için “stride” ve “padding” parametreleri ile kontrol edilebilir. Stride, pencerenin her adımda kaç birim hareket edeceğini belirlerken, padding ise kenar bölgelerdeki özelliklerin korunmasını sağlar. Pooling, ağı genelleştirme yeteneğini artırırken aynı zamanda hesaplama maliyetini azaltır. Büyük boyutlu özellik haritalarında detayları korurken boyutu azaltarak özelliklerin önemini özetler. Bu nedenle, pooling işlemi, özellik çıkarma (feature extraction) sürecinde önemli bir adımdır ve CNN modellerinde yaygın olarak kullanılır.



Şekil 3.9 “Max Pooling” ve “Average Pooling” işlem örneği (Islam 2020)

Dropout Katmanı:

Dropout katmanı Türkçede “Atış Katmanı” ya da “Bırakma Katmanı” olarak adlandırılır. Dropout, derin öğrenme modellerinde yaygın olarak kullanılan bir düzenleme (regularization) tekniğidir. Dropout, derin öğrenme modellerinde aşırı uyumu önlemek için kullanılan en popüler yöntemlerden biridir (Gal ve Ghahramani 2015). Dropout, ağı öğrenme sürecinde rastgele seçilen nöronların çıkışlarını geçici olarak kapatır, böylece her bir eğitim örneği için farklı alt ağlar oluşturulur (Srivastava vd. 2014).

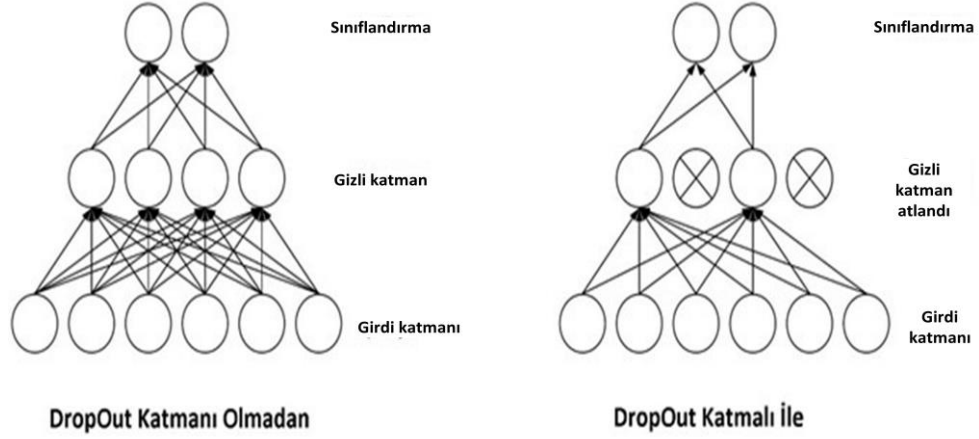
Dropout katmanı, girdi verisindeki her bir öğeyi belirli bir olasılıkla sıfıra eşitler. Bu işlem, nöronların bağlantılarını geçici olarak kırarak ağı genel karmaşıklığını azaltır. Böylece, ağı farklı özellik kombinasyonlarını öğrenmesi teşvik edilir ve aşırı uyum riski azaltılır.

Dropout işlemi matematiksel olarak aşağıdaki gibi ifade edilebilir:

$$y = x \times \text{mask} \quad (3.8)$$

Burada, x girdi verisini, y çıktı verisini ve mask ise girdi verisine karşılık gelen dropout maskelemesini temsil eder. Maskedeki 1 değerleri, bağlantıların korunduğunu ve 0 değerleri, bağlantıların geçici olarak kapatıldığını gösterir.

Dropout, ağın aşırı uyumu azaltırken aynı zamanda genelleştirme yeteneğini artırır. Her bir eğitim örneği için farklı alt ağlar kullanarak ağın farklı özellikleri öğrenmesini sağlar. Bu sayede, ağ daha sağlam ve daha iyi genelleme yapabilen bir yapıya sahip olur.



Şekil 3.10 Bir dropout katmanı uygulama şeması (Anonymous 5 2024)

Flatten Katmanı:

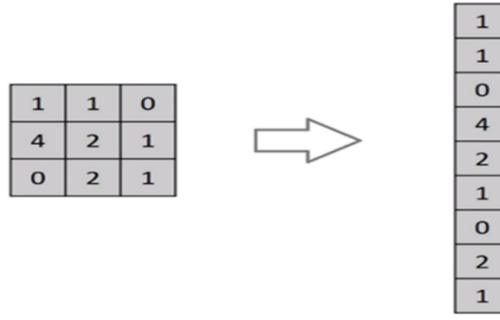
Flatten (Düzleştirme) katmanı, derin öğrenme modellerinde sıklıkla kullanılan bir katmandır. Bu katman, veri yapısını düzleştirerek, çok boyutlu verileri tek boyutlu bir vektör haline getirir. Bu sayede, girdi verilerinin boyutunu düşürerek ardışık bir yapı elde edilmesini sağlar (Jin vd. 2014).

Flatten katmanı, özellikle CNN gibi katmanlardan çıkan 3 boyutlu tensörleri veya havuzlama (pooling) katmanlarından çıkan 2 boyutlu tensörleri düzleştirmek için kullanılır. Bu şekilde, girdi verilerinin düz bir liste haline getirilmesi sağlanır ve bu liste, ardışık olarak bağlı (fully connected) katmanlara verilebilir.

Matematiksel olarak, flatten katmanı girdi tensörünü $(n_1, n_2, \dots, n_m)$ boyutunda bir tensör olarak kabul eder ve çıktı tensörünü $(n_1 * n_2 * \dots * n_m)$ boyutunda bir tensör olarak üretir. Bu işlem, girdi tensöründeki tüm elemanları sıralayarak tek bir vektör haline getirme işlemidir. Flatten katmanının temel amacı, veri yapısını düzleştirerek ardışık bir yapı elde etmektir. Bu yapı, ardışık olarak bağlı katmanlarda daha kolay işlenebilir. Özellikle sınıflandırma veya regresyon gibi görevlerde, flatten katmanı modelin girdi verilerini anlamasını kolaylaştırır ve daha iyi sonuçlar elde etmeyi sağlar.

Flatten katmanının tipleri veya varyasyonları pek bulunmamaktadır. Genellikle tek bir standart yapıya sahip olan bu katman, veri yapısını düzleştirmek için kullanılan basit bir ara katmandır. Farklı derin öğrenme çerçevelerinde farklı isimlere sahip olabilir, ancak temel işlevi aynıdır.

Flatten katmanı, derin öğrenme modellerinde veri yapısını düzleştirerek girdi verilerinin daha etkili bir şekilde işlenmesini sağlayan önemli bir katmandır (Bakınız: Şekil 3.11). Bu katman, veri boyutlarının uygun hale getirilmesi ve ardışık olarak bağlı katmanlara verilerin iletilmesi konusunda önemli bir rol oynar.



Şekil 3.11 Flatten Katmanı

Dense Katmanı:

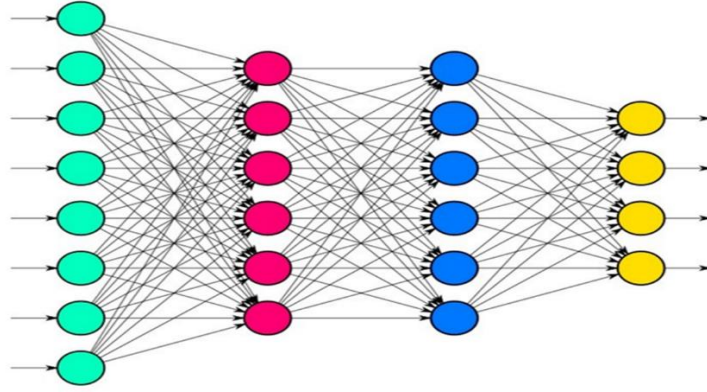
Değınilecek son katman ise dense katmandır. Dense katmanı (Tam Bağlantılı Katman), derin öğrenme modellerinde yaygın olarak kullanılan bir katman türüdür. Bu katman, her bir girdi birimini her bir çıktı birimiyle tamamen bağlar ve her bağlantıya bir ağırlık atanır. Dense katmanı, veri özelliklerini çıkararak karmaşık ilişkileri öğrenmek için kullanılır (Cybenko 1989).

Matematiksel olarak, dense katmanı, girdi birimlerini (n) ve çıktı birimlerini (m) içeren bir ağırlık matrisi (W) ve bir bias vektörü (b) kullanarak girdi ve çıktı arasındaki ilişkiyi ifade eder. Girdi birimleri ile çıktı birimleri arasındaki bağlantıyı ifade etmek için bir aktivasyon fonksiyonu da kullanılır. Bu katmanda her bir çıktı birimi, girdi birimlerinin ağırlıklı toplamına bir aktivasyon fonksiyonu uygular.

Dense katmanının temel işlevi, veriye ait girdi özelliklerini öğrenmek ve bu özelliklerin çıktı üzerindeki etkisini hesaplamaktır. Bu katman, her bir girdi biriminin her bir çıktı birimiyle bağlantılı olduğu için verinin daha yüksek seviyede temsil edilmesini sağlar. Bu sayede, derin öğrenme modeli, karmaşık veri yapılarını ve ilişkilerini öğrenebilir.

Dense katmanı, farklı varyasyonlara sahip olabilir. Örneğin, aktivasyon fonksiyonu olarak sigmoid, ReLU veya tanh gibi farklı fonksiyonlar kullanılabilir. Ayrıca, dropout veya Batch Normalization gibi düzenleme teknikleri de dense katmanına uygulanabilir. Bu farklı varyasyonlar, modelin performansını artırmak veya overfitting'i önlemek için kullanılabilir.

Dense katmanı, derin öğrenme modellerinde sınıflandırma, regresyon, dil işleme ve görüntü işleme gibi birçok görevde kullanılır. Bu katman, girdi verilerini yüksek seviyeli özelliklerle temsil ederek modelin öğrenme yeteneğini artırır. Ayrıca, modelin karmaşık veri yapılarını ve ilişkilerini öğrenmesine olanak tanır.



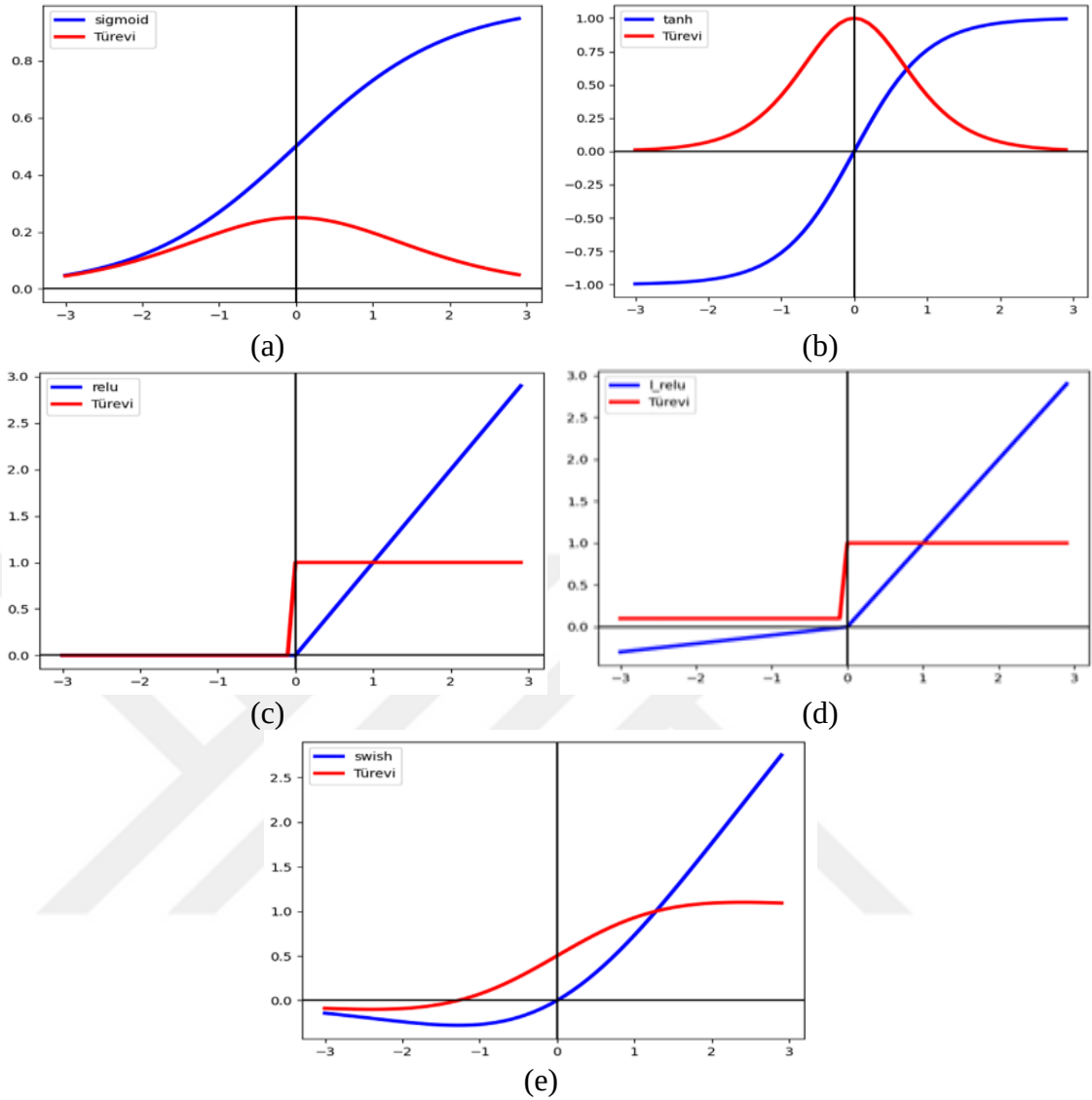
Şekil 3.12 Dense Katmanı (Izyumov vd. 2021)

3.4.2. Aktivasyon fonksiyonları

Aktivasyon fonksiyonları, yapay sinir ağlarında her bir nöronun çıktısını belirlemek için kullanılan matematiksel fonksiyonlardır. Bu fonksiyonlar, sinir ağı modelinin temel bileşenlerinden biridir ve sinir ağının öğrenme sürecinde büyük öneme sahiptir. Aktivasyon fonksiyonları, nöronun girdi toplamını alarak bir çıktı değeri üretir ve bu çıktıyı bir sonraki katmana veya sinir ağındaki diğer nöronlara iletebilir (Rumelhart vd. 1986).

Yapay sinir ağlarında kullanılan aktivasyon çeşitli fonksiyonlarının mevcuttur. Bunlar arasında sigmoid fonksiyonu, ReLU (Rectified Linear Unit), Leaky ReLU, tanh (hiperbolik tanjant) ve softmax sıklıkla kullanılanlarından bazılarıdır. Her bir aktivasyon fonksiyonunun farklı özellikleri ve kullanım alanları vardır. Aktivasyon fonksiyonları, sinir ağının öğrenme yeteneğini artırmak, girdi değerlerini sınırlamak, doğrusal olmayan özellik eklemek ve çıktıları olasılık dağılımına dönüştürmek gibi çeşitli amaçlar için kullanılırlar. Ayrıca, Aktivasyon fonksiyonları, sinir ağlarının girdi değerlerini doğrusal olmayan forma dönüştürerek daha karmaşık örüntüleri ve ilişkileri yakalayabilmesini sağlarlar.

Derin öğrenme modellerinde yaygın kullanılan geri yayılım algoritması, ağı eğitirken aktivasyon fonksiyonlarının türevlerinden yararlanır. Ağı eğitirken ağırlıkların güncellenmesi ve hataların düzeltilmesi için türev hesaplamalarına ihtiyaç duyulur. Aktivasyon fonksiyonlarının türevleri, bu hesaplamaların gerçekleştirilmesine yardımcı olur.



Şekil 3.13 Aktivasyon fonksiyonları ve türevleri (a-sigmoid fonksiyonu; b-tanh fonksiyonu; c-ReLU fonksiyonu; d-leaky ReLU fonksiyonu; e-swish fonksiyonu)

Sigmoid fonksiyonu:

Sigmoid fonksiyonları, özellikle sınıflandırma problemlerinde kullanılır. Sınırları 0 ile 1 arasında olan ve S şeklinde bir eğriye sahiptir. Bu fonksiyon, girdi değerini olasılık değerine dönüştürerek sınıflandırma kararlarını vermek için kullanılır. Sigmoid fonksiyonunun matematiksel modeli aşağıdaki gibidir:

$$f(x) = 1/(1 + \exp(-x)) \quad (3.9)$$

ReLU fonksiyonu:

ReLU fonksiyonları, sıfırdan küçük girdi değerlerinde sıfır çıktı üretirken, pozitif girdi değerlerini doğrudan geçişi sağlayan lineer bir ilişkiyle aktarır. Bu fonksiyon, hesaplaması hızlı olduğu için tercih edilir ve sinir ağlarının daha hızlı öğrenmesini sağlar. Fonksiyonu matematiksel olarak, maksimum() fonksiyonunu kullanarak tanımlayabiliriz. ReLU fonksiyonunun matematiksel modeli aşağıdaki gibidir:

$$f(x) = \max(0, x) \quad (3.10)$$

Leaky ReLU fonksiyonu.

Leaky ReLU fonksiyonları, ReLU fonksiyonunun sıfırdan küçük girdi değerlerinde küçük bir sızıntı sağlayan bir varyasyonudur. Bu, gradiyentin sıfıra yakın olmasını önleyerek daha iyi bir öğrenme yeteneği sağlar. Bu fonksiyon, girdi x için aşağıdaki matematiksel modeli kullanır:

$$f(x) = \max(ax, x) \quad (3.11)$$

Burada, a sızıntı katsayısıdır ve tipik olarak küçük pozitif bir değerdir.

Tanh fonksiyonu:

Tanh fonksiyonu, sınırları -1 ile 1 arasında olan ve S şeklinde simetrik bir eğriye sahip olan bir aktivasyon fonksiyonudur. Girdi değerlerini -1 ile 1 arasında sıkıştırarak sonuca doğrusal olmayan özellik ekler ve simetrik bir çıktı üretir. Tanh fonksiyonunun matematiksel modeli aşağıdaki gibidir:

$$f(x) = \frac{(\exp(x) - \exp(-x))}{(\exp(x) + \exp(-x))} \quad (3.12)$$

Burada x , girdi değerini temsil eder.

Softmax fonksiyonu:

Softmax fonksiyonu, genellikle çok sınıflı sınıflandırma problemlerinde kullanılır. Bu fonksiyon, çıktıları birbirine orantılı hale getirerek olasılık dağılımı elde eder. Bu sayede sinir ağı, girdilere dayalı en olası sınıflandırma kararını verebilir.

$$f(x_i) = \frac{\exp(x_i)}{(\sum \exp(x_j))} \rightarrow j : 1, n \quad (3.13)$$

Burada, x_i , i . nöronun girdi değerini temsil eder ve n , toplam nöron sayısını temsil eder.

Swish fonksiyonu:

Swish fonksiyonu, aktivasyon fonksiyonları arasında son zamanlarda popülerlik kazanan bir türdür. Swish fonksiyonu, girdi değerlerine dayalı olarak çıkış değerlerini hesaplamak için kullanılır. Bu fonksiyon, sigmoid fonksiyonunun bir türevidir ve daha yumuşak bir gradyan geçişine sahiptir. Matematiksel olarak Swish fonksiyonu, aşağıdaki formülle ifade edilir:

$$f(x) = x \times \text{sigmoid}(\beta \times x) \quad (3.14)$$

Burada x , girdi değerini temsil eder ve β , bir parametredir. Sigmoid fonksiyonu, girdi değerini 0 ile 1 arasında sınırlar, ardından bu değeri girdiyle çarparak çıkış değerini hesaplar. β parametresi, fonksiyonun eğimini kontrol eder ve Swish fonksiyonunun genel davranışını etkiler.

Swish fonksiyonunun bazı avantajları vardır. Özellikle derin sinir ağlarında kullanıldığında, daha iyi bir performans sergileyebilir. Bu fonksiyon, ReLU aktivasyon

fonksiyonuna benzer bir şekilde, gradyanın daha verimli bir şekilde akmasını sağlar ve ağıın öğrenme sürecinin hızlanmasına yardımcı olur. Swish fonksiyonunun grafiksel olarak sigmoid fonksiyonuna benzediği görülür, ancak Swish fonksiyonunun daha hızlı bir artış oranı ve daha geniş bir doğrusal bölgesi vardır. Bu, Swish fonksiyonunun, derin öğrenme modellerinde kullanılan diğer aktivasyon fonksiyonlarına göre daha iyi performans göstermesine yardımcı olabilir.

3.4.3. Yapay sinir ağı eğitimi

Yapay Sinir Ağı (YSA) eğitimi, bir YSA'nın veri setlerine dayalı olarak öğrenme yeteneğini kazanması sürecidir. Bu süreçte, ağına veri seti sunulur ve ağı, veriler arasındaki örüntüleri tanımak ve çıkarımlar yapmak için ağırlıklarını ve parametrelerini günceller. YSA eğitiminde kullanılan yöntemler arasında çeşitli yaklaşımlar bulunmaktadır. Bunlar;

- İteratif Yöntemler: Bu yöntemlerde, ağıın hedef çıktıya ulaşması için ağırlıklar ve parametreler iteratif olarak güncellenir. Geri yayılım (backpropagation) yöntemi bu kategoriye dahildir. Bu yöntemde, ağıın çıktıları ile gerçek çıktılar arasındaki hata hesaplanır ve bu hata kullanılarak ağırlıklar geriye doğru güncellenir. Bu süreç, bir hata fonksiyonunun türevlerini hesaplayarak ağırlık güncellemelerini gerçekleştirir.
- Optimizasyon Yöntemleri: Bu yöntemler, bir hedef fonksiyonunun minimize edilmesi veya maksimize edilmesiyle ilgilidir. YSA eğitiminde, genellikle hedef fonksiyon olarak kayıp fonksiyonu kullanılır. Gradyan inişi (gradient descent) ve stokastik gradyan inişi (stochastic gradient descent) gibi optimizasyon algoritmaları, ağırlık güncellemelerini yapmak için hedef fonksiyonunun gradyanını kullanır. Bu yöntemler, iteratif bir şekilde ağırlıkları ayarlayarak hedef fonksiyonunu eniyilemeye çalışır.
- Evrimsel Algoritmalar: Bu yöntemler, doğal seleksiyon ve genetik algoritma prensiplerine dayanır. YSA eğitiminde, ağıın genleri olarak düşünülen ağırlıklar ve parametreler, bir nesil içinde popülasyon üzerinde değiştirilir ve en iyi performans gösteren bireyler seçilerek bir sonraki nesille aktarılır. Bu süreç, popülasyonun zamanla daha iyi performans gösterecek şekilde evrilmesini sağlar.

YSA eğitimi için kullanılan yöntemler, veri setinin doğası, problemin karmaşıklığı ve hedeflenen performans kriterlerine bağlı olarak seçilir. Bu yöntemlerin amacı, ağıın veri setindeki örüntüleri doğru bir şekilde tanımasını ve genelleme yapabilmesini sağlamaktır. Eğitim sürecinde, ağıın ağırlıkları ve parametreleri iteratif olarak ayarlanırken, hedef fonksiyonun daha iyi olması ve performansın artırılması amaçlanır.

3.4.3.1. Kayıp fonksiyonları ve optimizatörler

Kayıp fonksiyonları (loss functions), makine öğrenmesi ve derin öğrenme modellerinin eğitiminde kullanılan önemli bir bileşendir. Bu fonksiyonlar, modelin gerçek değerlerden ne kadar uzak olduğunu ölçerek eğitim sürecinde modelin optimize edilmesini sağlar. Kayıp fonksiyonları, modelin tahminlerinin gerçek değerlerle ne kadar uyumlu olduğunu hesaplamak için kullanılır ve bu uyumun derecesi, kayıp fonksiyonunun değeriyle temsil edilir. Genel olarak, amaç en düşük kaybı elde etmek ve

modeli gerçek değerlere mümkün olan en iyi şekilde uydurmak için kayıp fonksiyonunu minimize etmektir.

Kayıp fonksiyonları, farklı problemler için farklı şekillerde tanımlanabilir. En yaygın problem tipleri Regresyon ve Sınıflandırma problemleridir. Yapay sinir ağlarının eğitiminde sıklıkla kullanılan kayıp fonksiyonları arasında ortalama karesel hata (MSE), ortalama mutlak hata (MAE), ortalama mutlak yüzdesel hata (MAPE), kök ortalama karesel hata (RMSE), LogCosh, Huber ve kategorik çapraz entropi (CCE) 'den bahsedebiliriz. MSE, MAE, MAPE, RMSE, Log-Cosh regresyon problemler için, CCE sınıflandırma problemleri için uygun sonuçlar üretirler.

Ortalama Karesel Hata (MSE): Tahmin edilen ve gerçek değerler arasındaki farkların karelerinin ortalamasıdır. Bu fonksiyon, regresyon problemlerinde yaygın olarak kullanılır ve genellikle sürekli değerlerin tahmin edildiği durumlarda tercih edilir. MSE, aşağıdaki formülle gösterilir:

$$MSE = \left(\frac{1}{n}\right) \times \sum (y - y_t)^2 \quad (3.15)$$

Burada, y gerçek değerleri, y_t tahmin edilen değerleri ve n veri noktalarının sayısını temsil eder.

Ortalama Mutlak Hata (MAE): Tahmin edilen ve gerçek değerler arasındaki mutlak farkların ortalamasıdır. Bu fonksiyon da regresyon problemlerinde kullanılır ve özellikle aykırı değerlere karşı daha dayanıklıdır. MAE, aşağıdaki formülle gösterilir:

$$MAE = \left(\frac{1}{n}\right) \times \sum (y - y_t) \quad (3.16)$$

Burada, y gerçek değerleri, y_t tahmin edilen değerleri ve n veri noktalarının sayısını temsil eder.

Ortalama Mutlak Yüzde Hata (MAPE): MAPE, özellikle tahminlerin yüzdesel hatalarının önemli olduğu durumlarda kullanılan bir kayıp fonksiyonudur. Genellikle regresyon problemlerinde kullanılır ve tahmin hatalarını yüzdesel olarak ifade eder. MAPE, aşağıdaki formülle gösterilir:

$$MAPE = \left(\frac{1}{n}\right) \times \sum (|(y - y_t)/y|) \times 100 \quad (3.17)$$

Burada, y gerçek etiketleri, y_t tahmin edilen olasılıkları temsil eder.

Kök Ortalama Kare Hata (RMSE): RMSE, tahminlerin gerçek değerlerle ne kadar uyumlu olduğunu ölçen bir kayıp fonksiyonudur. MSE'ye benzer şekilde regresyon problemlerinde kullanılır. RMSE, tahmin hatalarının karelerinin ortalamasının karekökü olarak hesaplanır. RMSE, aşağıdaki formülle gösterilir:

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \times \sum (y - y_t)^2} \quad (3.18)$$

Burada, y gerçek etiketleri, y_t tahmin edilen olasılıkları temsil eder.

Logaritmik Hiperbolik Kosinüs Hata (LogCosh): LogCosh kayıp fonksiyonu, MAE ve MSE hataları arasında bir tür birleştirme olarak düşünülebilir. MAE, aykırı değerlere karşı daha dayanıklıdır, ancak eğitim sürecinde gradyanların hesaplanması sırasında bazı sorunlara neden olabilir. MSE ise aykırı değerlere daha hassas bir şekilde tepki verir, ancak gradyan patlaması veya silinmesi sorunlarına neden olabilir. LogCosh, bu iki yaklaşımı dengelemek için tasarlanmıştır. LogCosh'un formülasyonu aşağıdaki gibidir.

$$\text{LogCosh} = \log \cosh(y - y_t) \quad (3.19)$$

Burada, y gerçek etiketleri, y_t tahmin edilen olasılıkları temsil eder.

Huber kayıp Hata (Huber): Huber kaybı, LogCosh fonksiyonun da olduğu gibi MAE ve MSE nin arasında bir denge sağlamak amacıyla tasarlanmış bir kayıp fonksiyonudur. Huber kaybı bu iki yaklaşımı birleştirir ve hem düşük hem de yüksek hata değerlerine karşı istikrarlı bir şekilde davranır.

$$\text{Huber}(y - y_t) = \begin{cases} \frac{1}{2}(y - y_t)^2, & (y - y_t) \leq \Delta \\ \Delta \left| (y - y_t) - \frac{1}{2}\Delta \right|, & (y - y_t) > \Delta \end{cases} \quad (3.20)$$

Burada, y gerçek etiketleri, y_t tahmin edilen olasılıkları ve Δ (delta) bir eşik değeri temsil eder. Eşik değeri Δ , bir tür hata toleransını ifade eder. Eğer tahmin ile gerçek değer arasındaki mutlak fark Δ değerinden küçükse, kayıp, hata karesinin yarısı olarak hesaplanır (MSE'ye benzer). Eğer bu mutlak fark Δ değerinden büyükse, kayıp, hata miktarının doğrusal bir şekilde arttığı bir terim içerir (MAE'ye benzer).

3.4.3.2. Eğitim metrikleri

Yapay sinir ağının kayıp değerlerinin hesaplanmasında kullanılan birçok teknik aynı zamanda eğitim durumunu değerlendirmek için kullanılan metrikler ile aynı isim ve fonksiyonlara sahiplerdir. Bu metriklerden bazıları kayıp fonksiyonlarında detaylıca anlatılan MSE, RMSE, MAE ve MAPE'dir. Bu listeye eklenebilecek önemli bir diğer metrik ise R2 (Belirleme Katsayısı - Coefficient of Determination) metriğidir.

Belirleme Katsayısı (R2): R2, genel adıyla R2 skoru, bir regresyon modelinin gözlemlenen veriye ne kadar iyi uyduğunu ölçen istatistiksel bir ölçüdür. Değer aralığı 0 ile 1 arasında değişebilmekte ve 1'e ne kadar yakınsa modelin veriyi o kadar iyi açıkladığını göstermektedir. R2 skor değeri, aşağıdaki formülle hesaplanır:

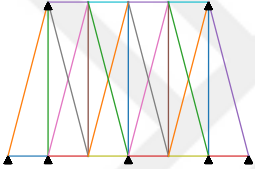
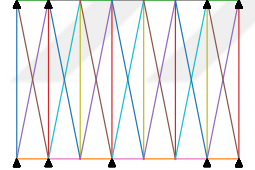
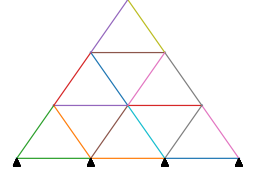
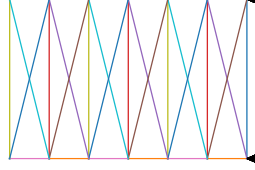
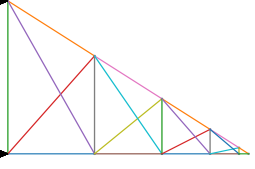
$$R2 = 1 - \left(\sum (y - y_t)^2 / \sum (y - y_{mean})^2 \right) \quad (3.21)$$

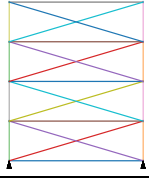
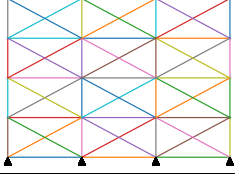
Burada, y gerçek değerleri, y_t tahmin edilen değerleri ve y_{mean} gerçek değerlerin ortalamasını temsil eder.

3.5. Eğitim Setinin Oluşturulması

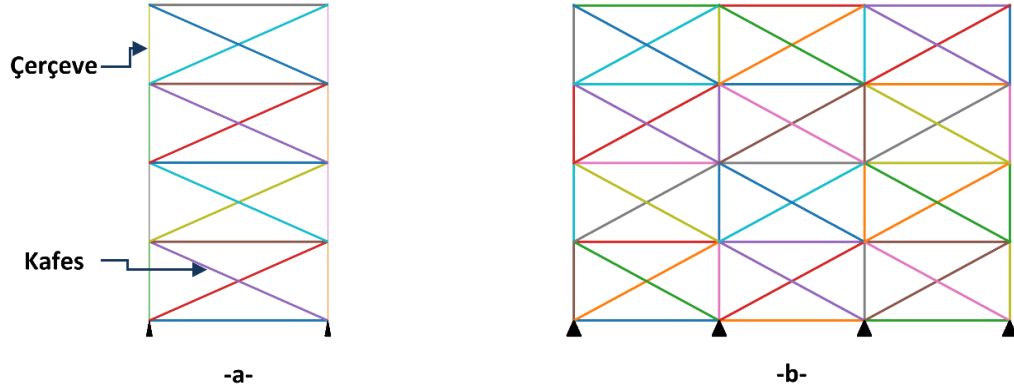
Geliştirilen yapay sinir ağı modellerinin doğru bir biçimde eğitimleri için birbirinde farklı yapısal özelliklere sahip yapı örnekleri kullanılması planlanmıştır. Bu örnekleri oluştururken yapay sinir ağı modellerinin daha kolay bir biçimde yapıları öğrenilmesi için yapılar iki boyutlu ve yapı tipleri ise sınırlı sayıda tutulmuştur. Eğitim setinin oluşturulması için yedi farklı yapı örneği hazırlanmış ve bu örneklerin ortak ve değişken özellikleri tespit edilmiştir. Daha sonra bu özellikler belirli limitler içerisine sokularak her bir örnek tipi için değişkenler oluşturulmuştur. Çizelge 3.1’de bu örnekler, örneklere ait belirleyici özellikler ve değişken aralıkları görülmektedir.

Çizelge 3.1 Eğitim Seti Yapı Tipleri ve Temel Özellikleri

Görsel	Tip Adı	Eleman Sayısı Aralığı (adet)	Düğüm Sayısı Aralığı (adet)	Yapı Elemanı Kesit Çapı Aralığı (mm)
	Klasik Köprü	20-105	5-44	1-300
	Klasik Köprü 2	21-61	10-26	1-300
	Piramit	9-570	6-210	1-300
	Ankastre Kiriş	16-56	8-24	1-300
	Ankastre Kiriş 2	8-56	5-21	1-300

	Dar Kule	18-120	8-42	1-300
	Geniş Kule	29-263	12-84	1-300

Örnekler, tüm yapıların aynı kesit tipinde çelik profillerden oluştuğu varsayılarak kafes yapı tipi kullanılarak oluşturulmuştur. Sadece “Dar kule” ve “Geniş kule” örnekleri içerisinde çerçeve yapı tipi, kafes yapı tipine ek olarak kullanılmıştır. (Şekil 3.14)



Şekil 3.14 Bünyesinde çerçeve yapı tipi kullanılan Örnekler a) Dar kule; b) Geniş kule

Başarılı bir eğitim setini oluşturmak için içerisinde oluşabilecek toplam farklı yapı tiplerinin bilinmesi gerekmektedir. Çizelge 3.1’de seçilen yapı örnek tipleri ile elde edilebilecek farklı yapıların ihtimal sayısını hesaplayabilmek için yapılar ile ilgili bu değişkenler, üretilmek istenen yapı türevi sayısı ile çarpılarak toplam ihtimal sayısına ulaşılır. Değişkenler ve istenen örnek türevi sayısı ile her bir örneğe ait farklı tipte yapı üretebilme ihtimali hesaplamaları aşağıdaki şekilde gerçekleştirilebilmektedir;

$$S = \left(\prod_{k=1}^n V_k \right) \times t \quad (3.22)$$

S : Toplam farklı yapı ihtimali

V_k : Her yapı tipi için gerekli değişkenlerin limitleri içerisindeki ihtimalleri

t : Türetilcek yapı miktarı

Yapılan işlemlere örnek olarak “Klasik Köprü” örneği için denklem (3.22) değerlendirildiğinde, örneğin oluşturulması için gereken değişken tipleri sırasıyla ‘Yatay Açıklık 1’, ‘Yatay Açıklık 2’, ‘Yapı Yüksekliği’, ‘Yatay Açıklık Sayısı’ ve ‘Çapraz

Durumu’ dan oluşmaktadır. Türev sayısı bu örnek için 10 olarak alınmıştır. Diğer değişkenler tam sayısı değişkenleri olup, değişken limitleri aşağıdaki gibidir.

Çizelge 3.2 “Klasik Köprü” örneğini oluşturmak için gerekli değişkenlerin limitleri

Değişken Adı	Limit	İhtimal Sayı
Yatay Açıklık 1	[2, 10]	8
Yatay Açıklık 2	[2, 10]	8
_Yapı Yüksekliği	[1, 20]	19
Yatay Açıklık Sayısı	[1, 10]	9
Çapraz Durumu	[0, 2]	3

Çizelge 3.2’den alınacak bu değerler denklem (3.22)’ye göre değerlendirildiğinde “Klasik Köprü” örneği için oluşturulabilecek toplam farklı yapı ihtimali sayısı $S = [(8 \times 8 \times 19 \times 9 \times 3) \times 10] = 328.320$ şeklinde hesaplanır. Denklem (3.22)’den yararlanılarak her bir yapı örnek tipi için elde edilebilecek farklı yapı tipi ihtimalleri Çizelge 3.3’de verilmiştir.

Çizelge 3.3 Ana Yapı Tipleri ve Farklılık İhtimalleri

Tip Adı	Yapı Değişken Sayısı	Değişken İhtimalleri	1. Türev Sayısı	1.Toplam İhtimal (adet)	2. Türev Sayısı	2.Toplam İhtimal (adet)
Klasik Köprü	5	8x8x19x9x3	10	328.320	3000	984.960.000
Klasik Köprü 2	4	18x18x9x9	10	262.440	3000	787.320.000
Piramit	3	18x18x196	10	635.040	3000	1.905.120.000
Ankastre Kiriş	5	18x18x9x9x3	10	787.320	3000	2.361.960.000
Ankastre Kiriş 2	3	18x9x3	10	4.860	3000	14.580.000
Dar Kule	3	18x196x196	10	6.914.880	3000	20.744.640.000
Geniş Kule	4	18x196x196x4	10	27.659.520	3000	82.978.560.000
Tüm Örnekler Genelinde Olası İhtimaller Toplamı:				36.592.380		109.777.140.000

Yukarıda bahsedilen yönergeler doğrultusunda yedi farklı yapı modelinden binlerce örnek türetilerek eğitim setinin oluşturulması için gerekli birincil ön hazırlık aşaması tamamlanmıştır.

Ön hazırlığın ikinci aşamasında, ilk aşamada üretilen her bir yapı türevine ait yapı elemanlarının kesit kalınlığı (1,0 – 300,0) limit aralığında doğrusal bir rakam ile rastgele belirlenmektedir. Bu işlem ikinci türev sayısı kadar daha tekrar edilerek aynı yapıdan yeni türevler üretilmektedir. Her bir yapı için seçilen ikinci türev sayısı 3000 olarak sınırlandırılmıştır. Böylece eğitim seti içerisinde bulunan toplam örnek sayısı aşağıdaki şekilde hesaplanabilmektedir.

$$\text{Eğitim seti örnek sayısı} = P1 \times t\ddot{u}v_1 \times t\ddot{u}v_2 \quad (3.23)$$

$P1$: Örnek tipi sayısı

$t\ddot{u}v_1$: 1. Türev sayısı

$t\ddot{u}v_2$: 2. Türev sayısı

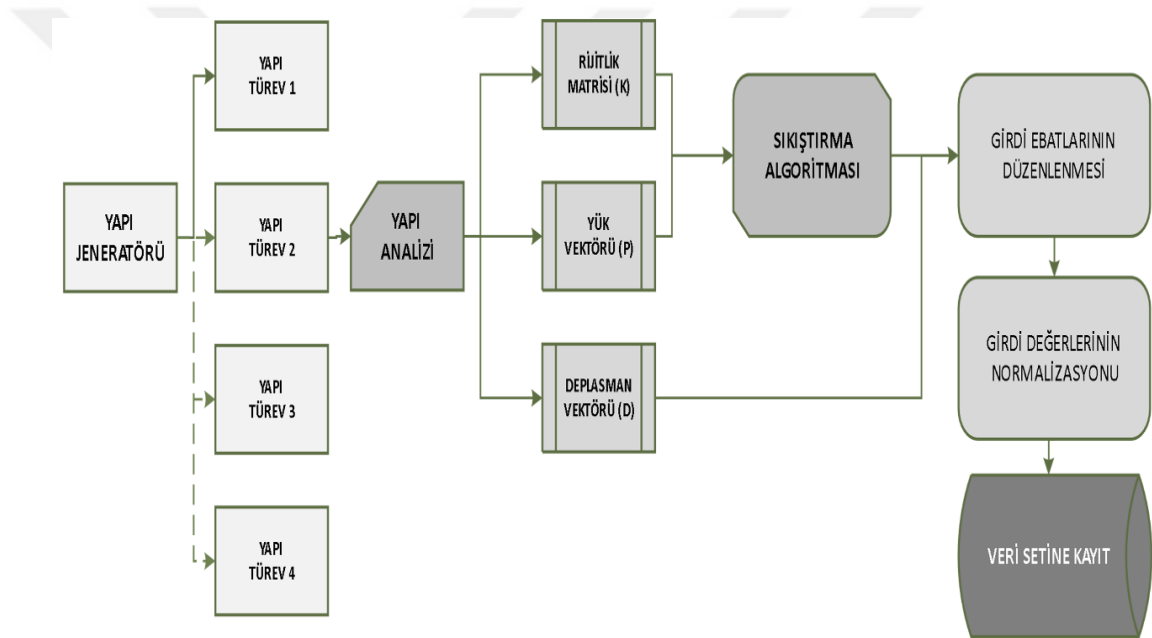
İşlemler formül üzerinde gerçekleştirildiğinde;

$$\text{Eğitim seti örnek sayısı} = 7 \times 10 \times 3000 = 210.000 \quad (3.24)$$

Şeklinde hesaplanabilecektir.

3.5.1. Yapı örneklerinin eğitim setine dönüştürülmesi

Eğitim setini hazırlama işlemi üretilen yapı örneklerinin bir dizi işlemten geçirilerek eğitim seti veri tabanına kaydedilmesiyle son bulmaktadır. Bu işlem hattının grafiksel tanımı Şekil 3.15'te verilmiştir. Bu hattı oluşturan işlemler aşağıda açıklanmıştır.



Şekil 3.15 Veri Seti Oluşturulması İşlem Hattı

ALGORİTMA 1: VERİ SETİ OLUŞTURULMASI İŞLEMİ SÖZDE KODU

Girdi: $Yapi_d \rightarrow$ Yapı Dosyası Adı, $YapiCl \rightarrow$ Yapı Özellikleri Detayları,

$YTS \rightarrow$ Yapı Türevi Sayısı

Çıktı: $KM \rightarrow$ Rijitlik Matrisi, $Dv \rightarrow$ Deplasman Vektörü, $Coeff \rightarrow$ Genişletme

Katsayı Matrisi, $scale \rightarrow$ Genişletme katsayısı, $KN \rightarrow$ Sıkıştırılmış Rijitlik Matrisi

1 $data=[]$

2 **Döngü** ($i \leftarrow YTS$) //Yapı türev sayısı kadar yeni yapı türet

3 $dosyaOku() \rightarrow YapiCl$ //Yapı tip dosyasını oku

4 $YapiCl.Sections \rightarrow Rand(YapiCl.NumberOfGroups)$ //Yapı kesiti rastgele seç

5 $YapiCl.Analiz()$ //Yapıyı analiz et

6 $sYapi \leftarrow Sıkıştırıcı(YapiCl)$ //Yapıyı sıkıştır

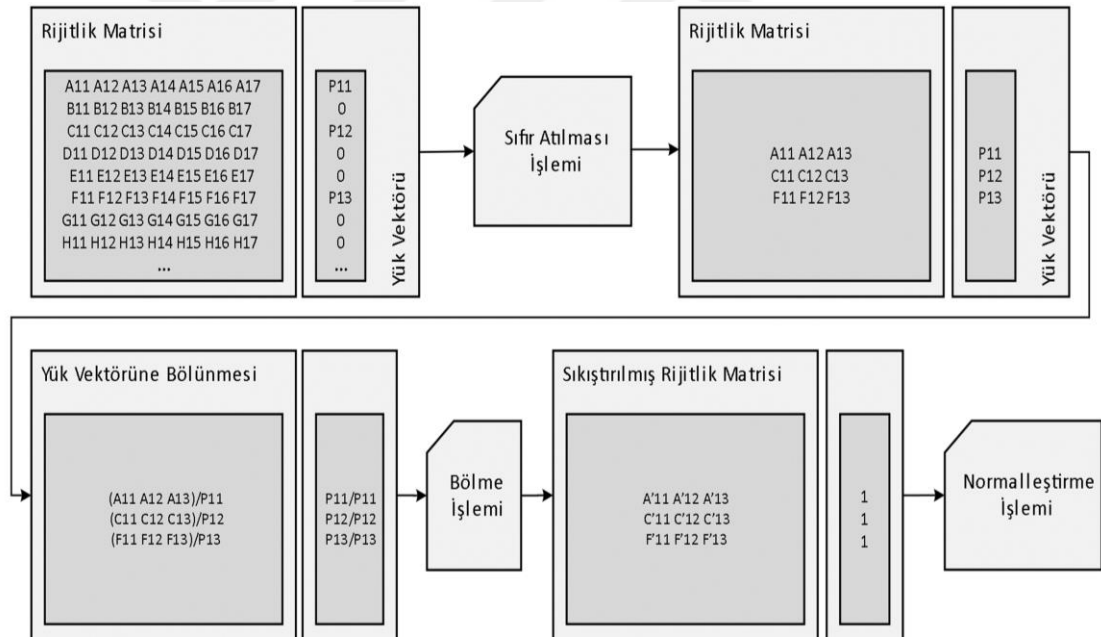
```

7      data[i]=sYapı
8  son
9  data verisini JSON biçimine dönüştür
10 dosyaKayıt()←data //Dataset dosyasını diske kaydet

```

3.5.1.1. Girdilerin küçültülmesi ve normalizasyonu

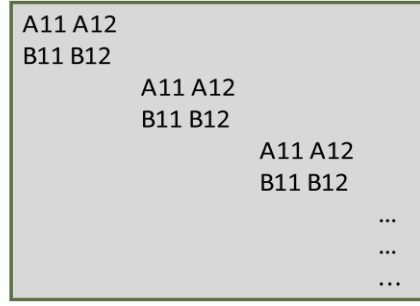
Tasarlanan yapay sinir ağlarının veri girişleri sabit bir ebatta olması planlanmıştır. 80x80'lik bir matris formunda olan bu girişler, yük vektörü ve rijitlik matrisinin sıkıştırılmış halidir. Sıkıştırma işlemi çalışmada kullanılan yapay sinir ağları için özel olarak geliştirilmiş bir dizi algoritmadan oluşmaktadır. Şekil 3.16'da görüldüğü üzere sıkıştırma işlemi içerisinde öncelikle yük vektörünün sıfır olduğu satırlar ve bunlara karşı gelen rijitlik matrisi üzerindeki satırlar atılır. Satır silinme işlemi sonrasında rijitlik matrisi ve yük vektöründeki tüm veriler yük vektörünün o satırdaki değerine bölünür. Bölme işlemi sonrasında yük vektörünün tamamı 1 olacağı için yük vektörü ayrı bir girdi olarak yapay sinir ağına gönderilmesine gerek kalmaz. Bölme işlemi sonrasında rijitlik matrisindeki tüm veriler rijitlik matrisindeki en büyük değerine bölünür ve en büyük değer genişletme algoritmasında kullanılmak üzere kaydedilir. Böylece matris normalizasyonu gerçekleştirilmiş olur. Matris normalizasyonu işlemi sonunda matristeki tüm değerler -1 ile +1 aralığına indirgenir.



Şekil 3.16 Girdinin Sıkıştırılması İşlemi

Matris normalizasyonu işlemi sonrasında iki farklı matris boyutundan biri elde edilir. Bunlardan birincisi üretilen matrisin yapay sinir ağı modeli için gerekli olan, ikincisi ise yeni oluşan matrisin, 80 x 80 boyutuyla aynı ebatta ya da büyük olmasıdır.

Matris normalizasyonu işlemi sonrasında mevcut matrisin 80 x 80 boyutundan küçük bir ebatta olması durumunda, matrisin 80 x 80 boyutlarına büyütülmesi için 80 x 80 matris ebatları içerisine sığacak kadar kopyasının Şekil 3.17'de görüldüğü şekilde diyagonal bir biçimde uç uca eklenmesi ile 80 x 80 boyutunda yeni bir matris oluşturulur.



Şekil 3.17 Sıkıştırıldıktan Sonra Büyütülme İşlemine Tabi Tutulmuş Rijitlik Matrisi Örneği

Matris normalizasyonu işlemi sonrasında mevcut matrisin 80 x 80 olması durumunda, hiçbir işlem yapılmadan yapay sinir ağına eğitime gönderilir.

Matris normalizasyonu işlemi sonrasında mevcut matrisin 80 x 80 boyutundan büyük bir ebatta olması durumunda, resim küçültme algoritmaları yardımıyla istenen 80 x 80 boyutlarına küçültülür. Bu küçültme işlemi Python (Python.Org 2024) dilinde çalışan, resimler üzerine çalışmak için kullanılan “scipy.ndimage (Scipy.Org 2024)” kütüphanesi içerisindeki “interpolation.zoom” metodunun kullanılması sonucunda gerçekleştirilmiştir.

Son aşamada, yukarıda bahsi geçen ihtimaller sonrası oluşan 80 x 80 ‘lık matris, 80 x 1 ‘lik ve her bir elemanı “1” olan temsili bir yük vektörü ile çarpılır. Böylece üretilen sonuç normalizasyon yapılarak yapay sinir ağına eğitimi için gerekli 80 x 1 ‘lik deplasman vektörü de elde edilmiş olur.

Tüm bu işlemlerin sırasıyla işlem hattı içerisinde gerçekleştirilmesi ile yapay sinir ağı için gerekli eğitim setinin oluşturulması aşaması tamamlanmış olmaktadır. Sıkıştırma işlemine ait algoritma aşağıda gösterilmiştir.

ALGORİTMA 2: GİRDİNİN SIKIŞTIRILMASI İŞLEMİ SÖZDE KODU

Girdi: $KM \rightarrow$ Rijitlik Matrisi, $PV \rightarrow$ Yük Vektörü
Çıktı: $KP \rightarrow$ YSA Girdi Matrisi, **Coeff** $\rightarrow$ Genişletme Katsayı Matrisi, **scale** $\rightarrow$ Genişletme katsayısı

- 1 $KN \leftarrow KM$ ile PV birleştir
- 2 $filter_1 \leftarrow PV$ içerisindeki sıfırların indeksleri
- 3 **döngü** ($i \rightarrow uzunluk(filter_1)$) // $filter_1$ eleman sayısı kadar tekrar et
 - 4 $zindex \leftarrow filter_1[i]$
 - 5 **döngü** ($j \rightarrow uzunluk(KP[0])$) // KP ’nin sütun sayısı kadar tekrar et
 - 6 $KP[i,j] \leftarrow KP[i,j] - KP[i,zindex] * (KP[zindex,j] / KP[zindex,zindex])$
 - 7 **son**
 - 8 KP içerisinde $filter_1$ indeks numaralı satır ve sütunları sil
- 9 **son**
- 10 KP matrisine normalizasyon işlemi gerçekleştir
- 11 $KN \leftarrow KP$ matrisini PV vektörüne böl ve ayır

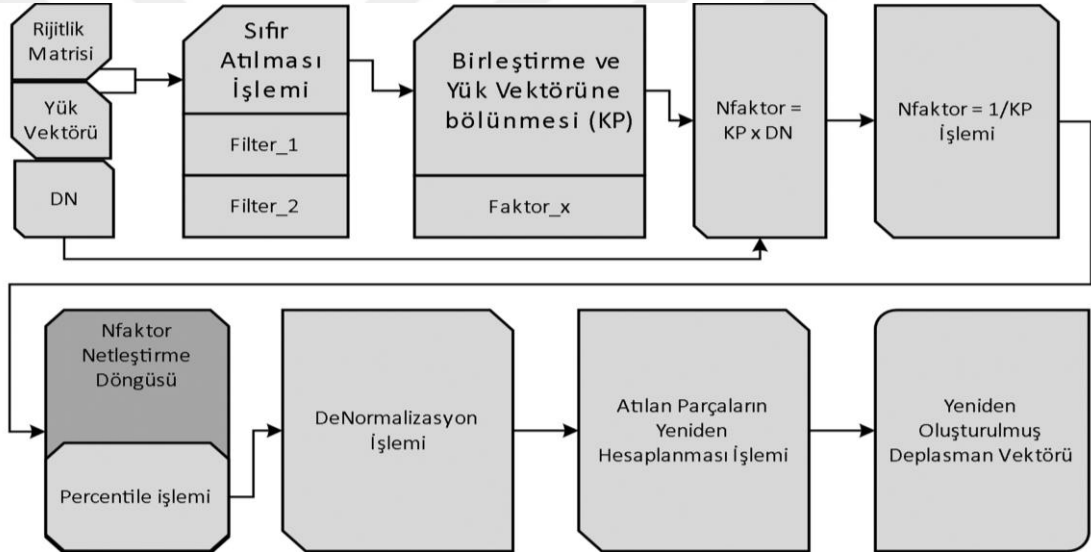
```

12 Eğer (boyut(KN) < 80x80) // eğer KN boyutları 80 birimden küçük ise(Şekil
    3.17)
13     döngü (i→uzunluk(KN[0])/80) // 80'in katı kadar tekrar et
14         | KN←KN matrisini sona ekle
15     son
16 son
17 Eğer (boyut(KN) > 80x80)
18     | Libscipy.ndimage kullnarak KN matrisini küçült
19 son

```

3.5.1.2. Genişletme algoritması

Genişletme işlem hattı için girdi olarak sadece yapay sinir ağından elde edilen 80x1'lik vektör yeterli olmamaktadır. Çıktı vektörünün yanı sıra rijitlik matrisi ve yük vektörü de gerekmektedir. Genişletme işlem hattına ait grafiksel tanımı Şekil 3.18'de gösterilmektedir.



Şekil 3.18 YSA Çıktısının Genişletilmesi İşlemi

Aynı sıkıştırma işleminde olduğu gibi rijitlik matrisinden yük vektörü üzerindeki sıfırlara karşılık gelen satırlar kaldırılır. Kaldırılan satırların konumları “Filtre_1”, kaldırılmayan satırların konumları “Filtre_2” değişkenlerine kaydedilir. Daha sonra sıkıştırma işleminde olduğu gibi küçültülen rijitlik matrisinin her elemanı yük vektörünün ilgili satırındaki eleman ile bölünür. Her bir satır için ortaya çıkan vektörün en büyük elemanı “faktör_x” isimli bir değişkene aktarılır. Sıkıştırılan rijitlik matrisi faktör_x vektörünün en büyük elemanı ile genişletilme işlemine tabi tutulur. Oluşan bu yeni matris işlemin geri kanalında, KP olarak adlandırılacaktır.

Bir sonraki aşamada KP matrisinin her elemanı 1 ‘e bölünerek “Nfaktör” vektörü elde edilir. Daha sonra bu vektör median_filter(vektör içerisindeki her bir değer, diğer değerlere göre) + Percentile işlemlerine sokularak içerisindeki üst ve alt bant dışında kalan aykırı değerlerin elenmesi işlemi gerçekleştirilir. İşlemin bu aşamasında amaç 80x1

'lık çıktı vektörünün DeNormalizasyonun ilk adımı gerçekleştirilerek rakamsal boyut özelliklerine geri getirilmeye çalışılmıştır.

Bir sonraki aşamada elde edilen vektörün adetsel olarak gerçek vektöre genişletilmesi işlemi yapılacaktır. Bu işlem için yük vektörü içerisinde sıfır değerlerine karşılık gelen ve kaldırılan elemanların sıra numaralarının kaydının tutulduğu Filtre_1 değişkeninden yararlanılarak KP matrisinin ilgili satırlarındaki elemanlar ile işleme sokulup genişletme işlemi tamamlanır.

ALGORİTMA 3: SIKIŞTIRILMIŞ VERİNİN GENİŞLETİLMESİ SÖZDE KODU

Girdi: YapiCl → Yapı Özellikleri Detayları, DN → YSA Ağ Matrisi

Çıktı: DR → Yapılandırılmış deplasman vektörü

- 1 *En ← 80*
 - 2 *K_ntwrk ← K_ntwrk reshape (En, En)*
 - 3 *KN ← K_ntwrk / max(K_ntwrk) // K_ntwrk'yi normalize et*
 - 4 *nfactor_ ← Nfonk(KN.dot(YapiCl.DV)) // Genişletme faktör sayısını hesapla*
 - 5 *D2N ← D_ntwrk * nfactor // D2N matrisini hesapla*
 - 6 *D2N ← D2N / YapiCl.scale_nt // Ölçek faktörünü uygula*
 - 7 *DR ← genişletme(YapiCl, D2N)*
 - 8 *son*
-

3.6. Geliştirilen Yapay Sinir Ağı Modelleri

Bu çalışmada, yapısal optimizasyon problemlerindeki çözüm süreçlerini kısaltmak amacıyla Keras kütüphanesi (Keras 2024) kullanılarak beş farklı yapay sinir ağı (YSA) modeli geliştirilmiştir. Keras, Python tabanlı bir derin öğrenme kütüphanesidir ve kullanıcı dostu arayüzü, modüler yapısı ve genişletilebilirliği sayesinde YSA modelleri oluşturmayı ve eğitmeyi kolaylaştırır. Keras, TensorFlow, Theano veya CNTK gibi popüler derin öğrenme altyapıları üzerinde çalışabilir ve bu da farklı donanım ve yazılım ortamlarında esneklik sağlar.

Geliştirilen YSA modelleri, yapısal analiz işleminin ana adımı olan deplasman hesaplamaları esnasında oluşan sürelerin azaltılmasına odaklanmıştır. Geleneksel FEM analizi içerisinde en çok zaman ve bellek tüketen Gauss eliminasyon işlemlerini atlayarak deplasman vektörünü tahmin etmek üzerine kurulu bir mekanizmaya sahiptirler. Bu modeller Conv2D, MaxPooling, Dropout, LeakyReLU, Flatten ve Dense gibi çeşitli katman türlerini içermektedir. Modellerin tamamında giriş katmanı ortak olarak kullanılmıştır. Modellerin diğer kısımları çok katmanlı geri yayımlı ağ (multilayer perceptron) olarak tasarlanmışlardır.

Modelin girdi katmanı, 80x80 piksel boyutunda bir tensördür. Bu katman yapının geometrik ve malzeme özelliklerini temsil eden rijitlik matrisinin sıkıştırma algoritmasından sonra kalan boyutlardaki halini içermektedir. Modelleri oluşturan diğer katmanlar modelden modele değişiklik göstermektedir. Keras, bu katmanların kolayca tanımlanmasını ve birbirine bağlanmasını sağlayarak model oluşturma sürecini basitleştirir. Ayrıca, Keras, optimizasyon algoritmaları (örneğin, Adam, SGD), kayıp fonksiyonları (örneğin, MAE, MSE, Huber) ve metrikler (örneğin, R2, accuracy) gibi derin öğrenme modellerinin eğitimi ve değerlendirilmesi için gerekli araçları sağlar.

Her YSA modelinde, Keras'ın sunduğu ModelCheckpoint ve EarlyStopping gibi geri çağırımlar (callbacks) kullanılarak model eğitimi sırasında en iyi ağırlıkların kaydedilmesi ve aşırı öğrenmenin önlenmesi sağlanmıştır. Keras'ın esnek yapısı, farklı hiperparametre kombinasyonlarını (örneğin, katman sayısı, filtre boyutu, drop oranı, öğrenme oranı) denemeyi ve model performansını optimize etmeyi kolaylaştırır.

Geliştirilen her bir YSA modelinin mimarisi, hiper parametreleri, eğitim sonuçları ve performans değerlendirmeleri ayrıntılı bir şekilde sunulmuştur. Ayrıca, modellerin veriye uyumu ve genelleme yeteneği, eğitim ve doğrulama verileri üzerinde elde edilen R2 skoru ve MAE gibi metrikler kullanılarak değerlendirilmiştir. (Bakınız Çizelge 3.4)

Çizelge 3.4 YSA Modellerine ait parametre değerleri

Model	Katman Sayısı	Batch Sayısı	Drop Oranı	Öğrenme Faktörü	Kayıp Fonk.
Model 1	3	32	0,1	0,001	MAE
Model 2	4	64	0,2	0,0001	Huber
Model 3	3	32	0,2	0,001	LogCosh
Model 4	6	32	0,5	0,0001	RMSE
Model 5	3-3	20	0,25	0,0005	MAE

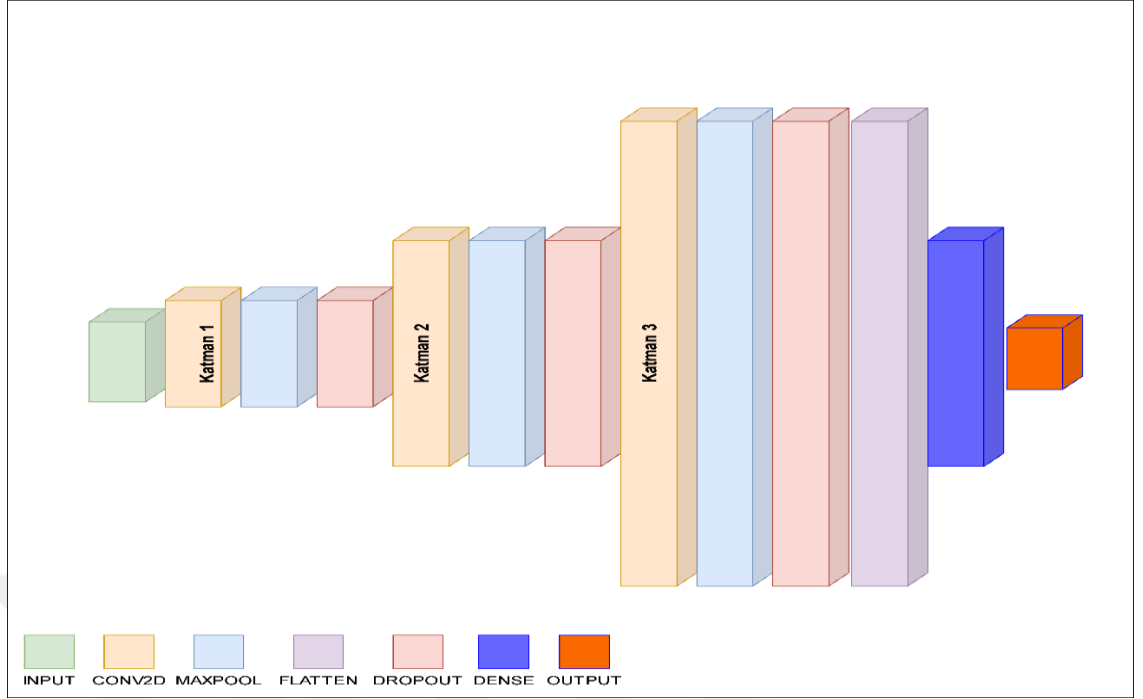
3.6.1. Yapay sinir ağı modeli 1

Bu Yapay Sinir Ağı (YSA) modeli, toplamda 13 katmandan oluşmaktadır, bunların 3'ü evrişimli katmanlardan oluşurken, diğerleri yoğun katmanlar ve pooling katmanlarıdır. Modelin katman dizilimi Şekil 3.19'da görsel olarak sunulmuştur. Bu katmanların temel özelliklerinden biri, benzer yapıda olmalarıdır, kernel genişlikleri 2x2 olarak belirlenmiş olsa da filtre sayıları farklı bırakılmıştır.

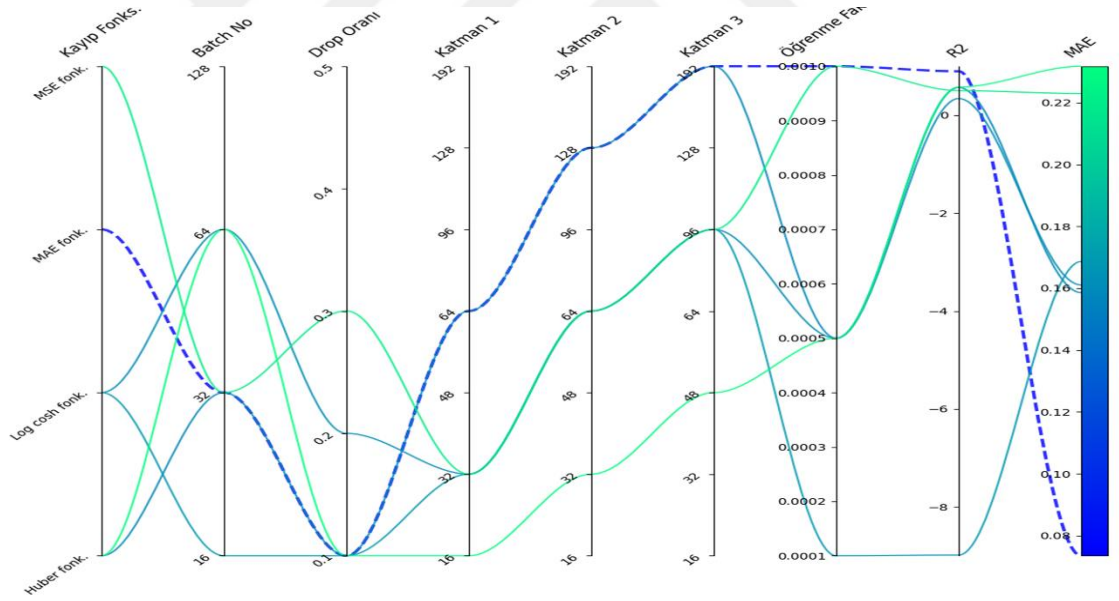
Model, hiperparametre araştırmaları sonucunda optimize edilip, sonuçlar Şekil 3.20'de sunulmuştur. Bu araştırmalar neticesinde, en düşük MAE kaybını sağlayan hiperparametre kombinasyonu, kayıp fonksiyonu olarak MAE kayıp fonksiyonu, 32 batch sayısı, 0,1 drop oranı ve 0,0001 öğrenme faktörünü içermektedir. Ayrıca, evrişimli katmanlara ait filtre sayıları sırasıyla 64, 128 ve 192 olarak belirlenmiştir. Bu hiper parametreler, ile model 0,986976 R2 skoru ve 0,005152845 MAE hata oranına ulaşmıştır.

Modelin sonundaki tek yoğun katmanın genişliği, istenen çıkış vektör büyüklüğü kadar yani 80 birim olarak belirlenmiştir. Ayrıca, modeldeki evrişimli katmanların küçükten büyüğe doğru sıralanması, özellik çıkarma sürecinin daha etkili ve verimli olmasının sağlaması amacıyla tasarlanmıştır. Bu tasarım kararı, genellikle daha düşük seviyeli özelliklerin daha önce öğrenilip, daha yüksek seviyeli özelliklere doğru gidildikçe geniş bağlam anlama yeteneğini artırmaya yönelik olarak tasarlanmıştır.

Pooling katmanlarının Conv2D katmanları arasına eklenmesi ile modelin hesaplama yükünü ve aşırı öğrenme riskini azaltılması hedeflenmiştir. Ayrıca, özellik haritalarının boyutunu azaltarak modelin genelleme yeteneğini artırması amaçlanmıştır. Tüm bu tasarım özellikleri ile modelin etkili bir şekilde çalışması ve istenilen sonuçları üretmesi hedeflenmiştir.

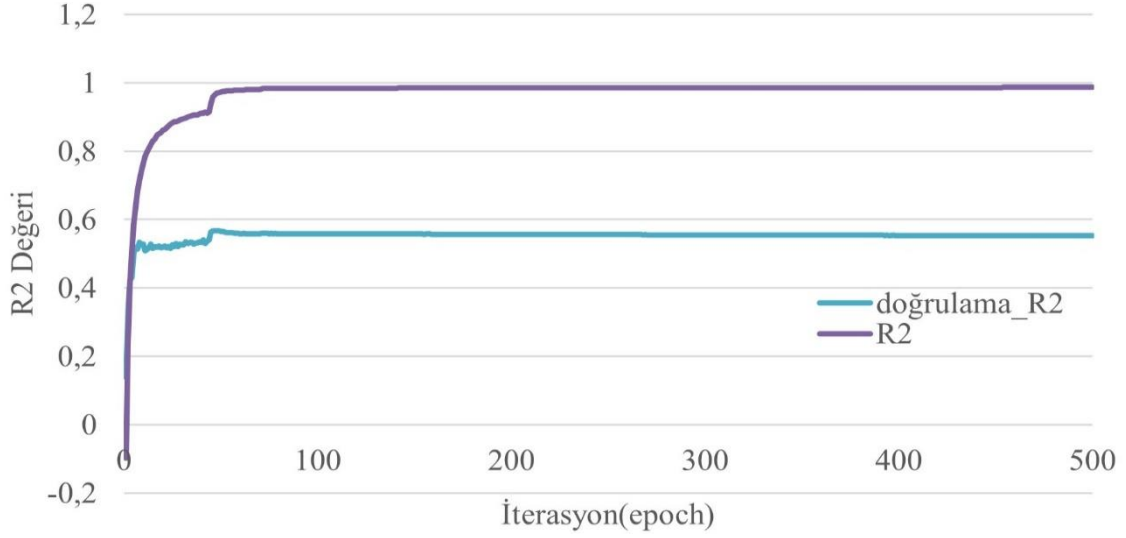


Şekil 3.19 Model 1 YSA şematik gösterimi



Şekil 3.20 YSA Model 1 en iyi altı hiper parametre sonuçları

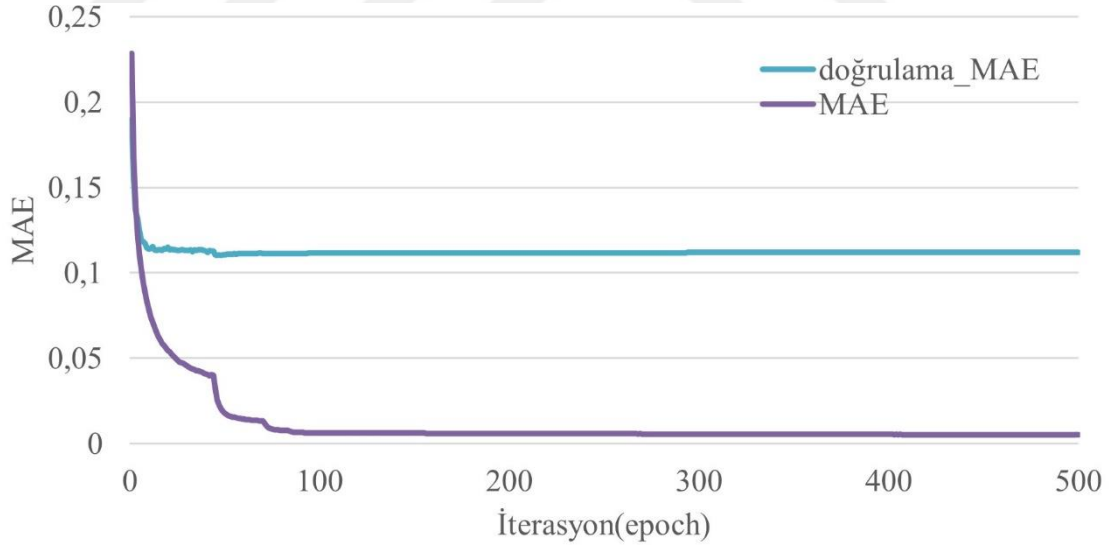
Şekil 3.21’de YSA model 1’ in en başarılı hiperparametre kombinasyonu altındaki R2 skoru ve Doğrulama R2 Skoru (validation_R2) değerleri görülmektedir. Şekil üzerindeki bu iki doğru arasındaki fark ileri iterasyonlarda paralelleşmeye başladığı görülmektedir. R2 ve Doğrulama R2 değerleri arasındaki bu farklılık YSA model 1’in girdiye uyumluluğunun zayıfladığını göstermekte ve 80 ‘inci iterasyondan sonra bu farkın sabitlendiği görülmektedir.



Şekil 3.21 YSA Model 1'e ait R2 ve Doğrulama R2 sonuçları

YSA model 1'e ait Şekil 3.22 MAE ve Doğrulama MAE (validation_MAE) değerleri incelendiğinde Şekil 3.21'de olduğu gibi doğrulama verilerinde iyileşmenin belirli bir iterasyondan sonra durduğu ve daha çok ilerlemediği gözlemlenmektedir.

Değerlerdeki bu hareket YSA model 1'in girdi verisine aşırı uyum sağlayarak eğitim seti dışında gelen verilere güvenilir tepki veremez hale geldiğini göstermektedir.



Şekil 3.22 YSA Model 1'e ait MAE ve Doğrulama MAE sonuçları

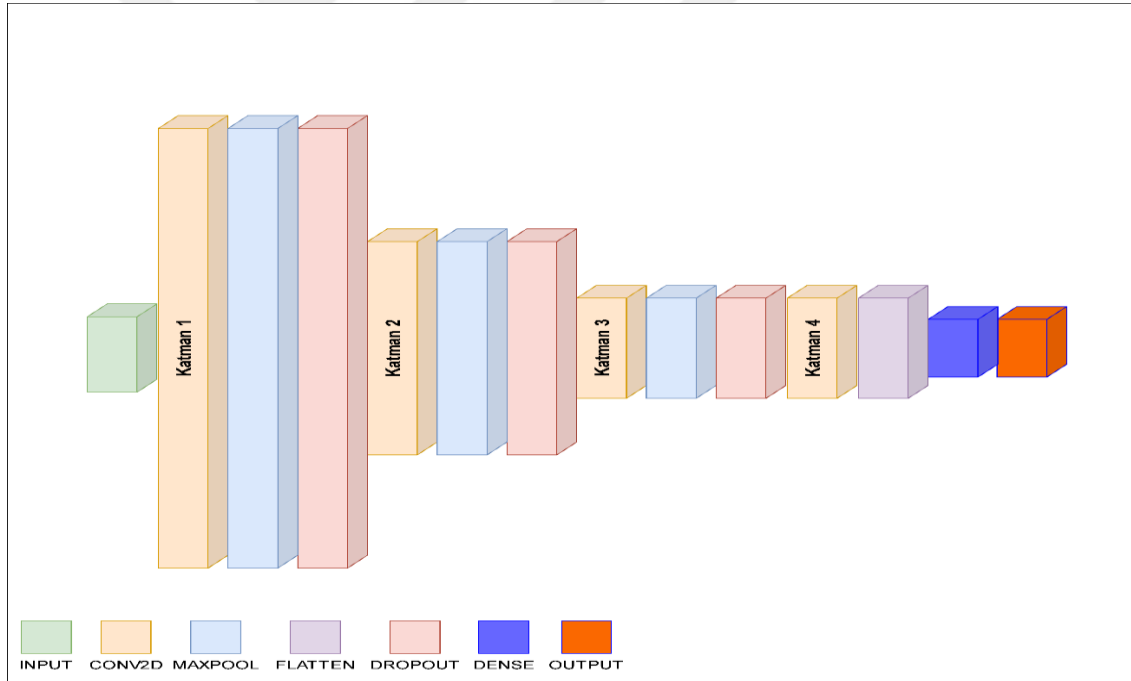
3.6.2. Yapay sinir ağı modeli 2

Bu Yapay Sinir Ağı (YSA) modeli, toplamda 14 katmandan oluşmaktadır, bunların 4'ü evrişimli katmandır ve modelin yapısal düzenlemesi Şekil 3.23'te görsel olarak sunulmuştur. Modeldeki evrişimli katmanlar benzerdir, ancak kernel genişlikleri sabit olarak 2x2 belirlenmiş ve farklı filtre sayıları kullanılmıştır. Ayrıca, Dropout katmanı ile, aşırı öğrenmenin önlenmesi hedeflenmiştir.

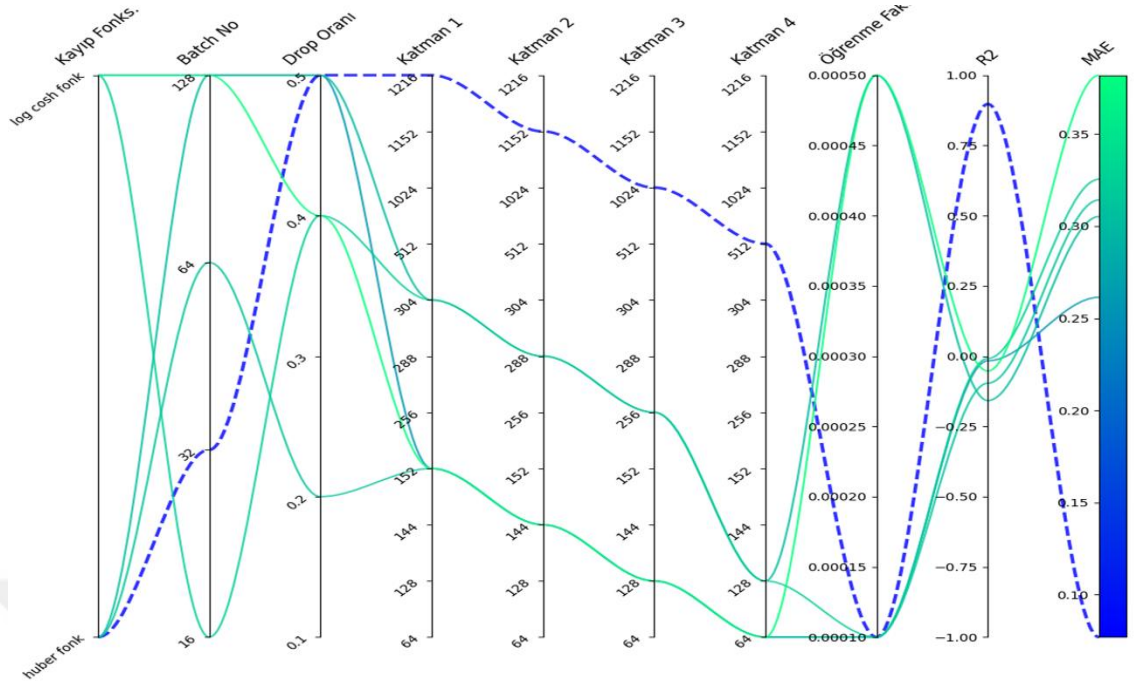
Modelle ilgili yapılan hiper parametre araştırmaları sonuçları Şekil 3.24'te sunulmuştur. En düşük MAE kaybını elde eden hiperparametre kombinasyonu içerisinde, kayıp fonksiyonu olarak Huber kayıp fonksiyonunu kullanırken, batch boyutunu 32 olarak, drop oranını 0,5 olarak ve öğrenme faktörünü 0,0001 olarak belirlenmiştir. Ayrıca, evrişimli katmanlara ait filtre sayıları da bu çalışma sonucunda sırasıyla 1216, 1152, 1024 ve 512 olarak belirlenmiştir. Bu hiperparametrelerle model, yüksek bir R2 skoru (0,99983) ve düşük bir MAE hata oranı ($9,005 \times 10^{-6}$) elde etmiştir.

Modelin son katmanı, istenen çıkış vektör büyüklüğü olan 80 birim olarak belirlenmiştir. Ayrıca, modeldeki evrişimli katmanları büyükten küçüğe sıralanmasıyla, daha yüksek seviyeli özellikleri öğrenmeye odaklanma ve global bağlamı daha etkili bir şekilde anlama amacı güdülmüştür. Böylece modelin daha iyi genelleme yeteneğine sahip olması amaçlanmıştır.

Son olarak, Pooling katmanlarının Conv2D katmanları arasına eklenmesiyle, hesaplama yükünün azaltılması ve aşırı öğrenmenin önüne geçilmesi amaçlanmıştır. Aynı zamanda özellik haritalarının boyutunu azaltarak modelin genelleme yeteneğinin artırılması hedeflenmiştir. Böylece, modelin daha iyi bir performans elde etmesine ve daha genel veriler üzerinde daha iyi sonuçlar vermesine yardımcı olması amaçlanmıştır.

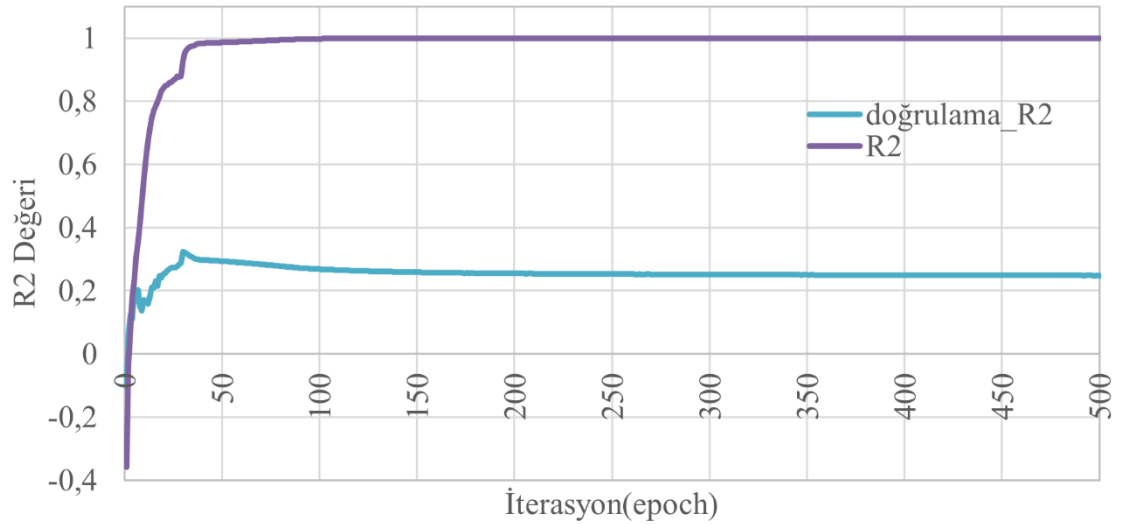


Şekil 3.23 Model 2 YSA şematik gösterimi

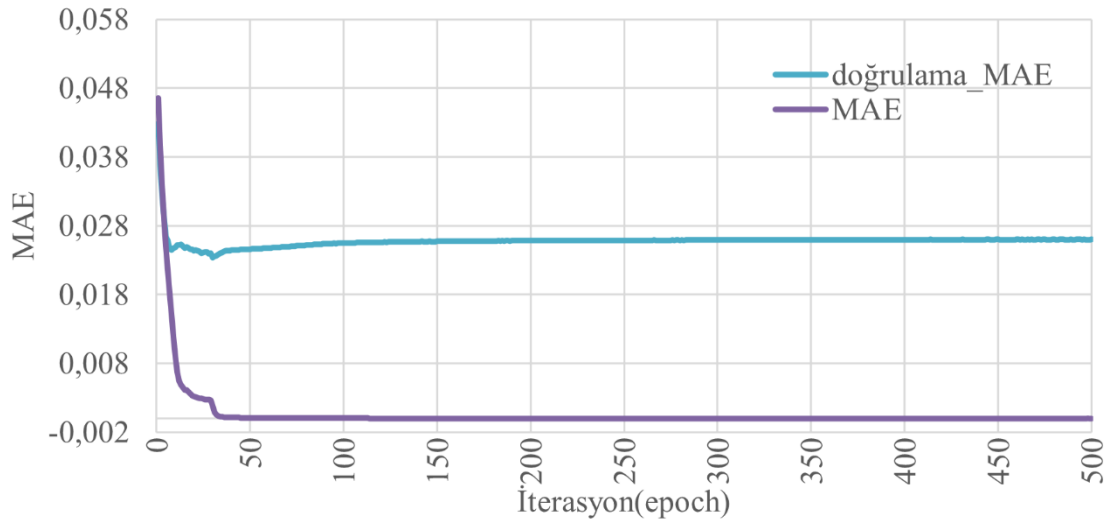


Şekil 3.24 YSA Model 2 en iyi beş hiper parametre sonuçları

YSA model 2'ye ait R2 skoru ve doğrulama R2 skoru (validation_R2) değerleri ile MAE ve doğrulama MAE değerleri Şekil 3.25 ve Şekil 3.26'da gösterilmektedir. Bu değerleri incelediğimizde açıkça görülmektedir ki YSA model 2 eğitimi esnasında belirli bir iterasyondan sonra girdi verilerine aşırı uyum sağlamış ve doğrulama için kullanılan verileri işlemede başarısız olmuştur. Ayrıca şekillerden de görülmektedir ki YSA model 2 veri uyum konusunda YSA model 1'den kısmen daha iyi sonuçlar vermiştir.



Şekil 3.25 YSA Model 2'e ait R2 ve Doğrulama R2 sonuçları



Şekil 3.26 YSA Model 2'e ait MAE ve Doğrulama MAE sonuçları

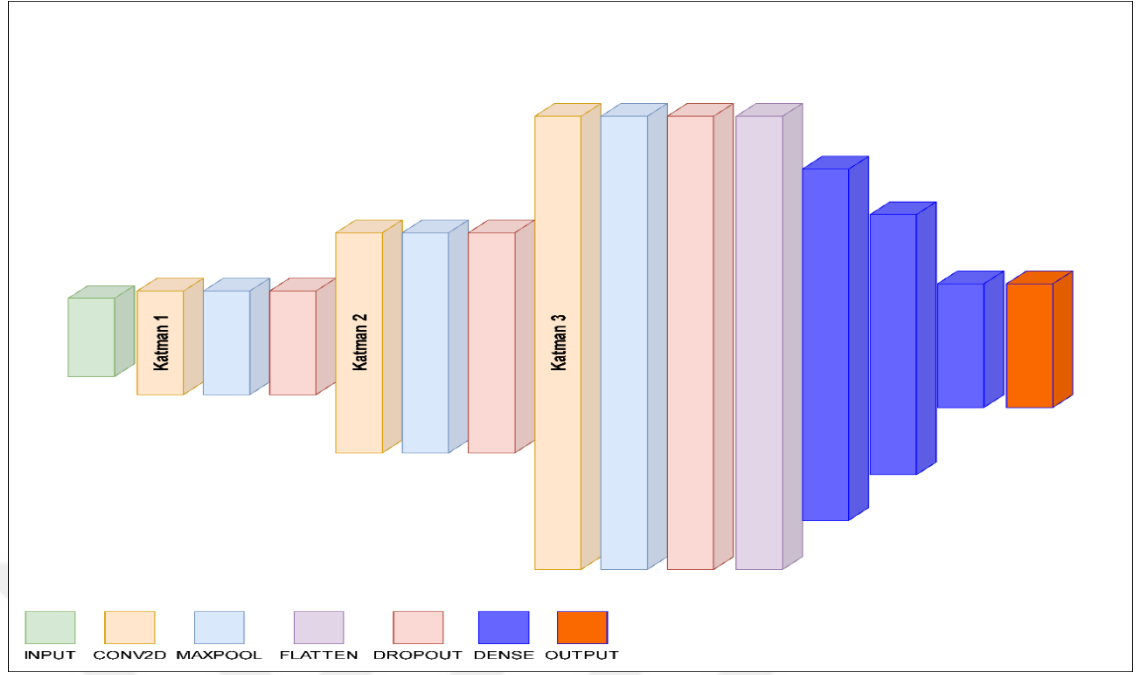
3.6.3. Yapay sinir ağı modeli 3

Elde ettiğimiz Yapay Sinir Ağı (YSA) modeli, toplamda 15 katmandan oluşmaktadır. Bu katmanlar içerisinde 3 evrişimli katman yer almaktadır ve modelin mimarisi Şekil 3.27'de görsel olarak sunulmuştur. Özellikle, bu modele özgü olan özelliklerden biri, tüm evrişimli katmanların benzer bir yapıda olmasıdır. Bu katmanlar için 2x2 kernel boyutları tercih edilmiş, ancak farklı filtre sayıları kullanılmıştır. Ayrıca, dropout katmanı, modelin her evrişimli katmanı sonrasında uygulanmıştır. Bu katmanın kullanılma nedeni, özellik haritalarındaki aşırı bağımlılığı azaltmak ve aşırı öğrenmeyi engellemektir.

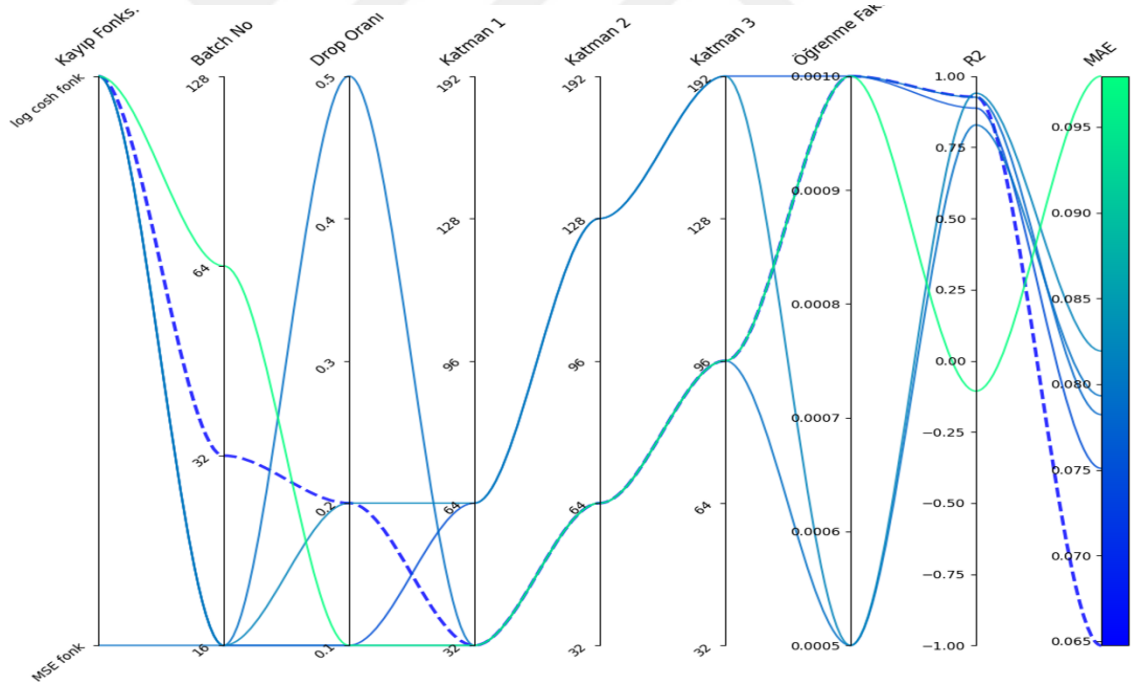
Hiperparametre araştırmaları sonuçları Şekil 3.28'de gösterilmiştir. En iyi performansı gösteren hiper parametre kombinasyonu, log_cosh kayıp fonksiyonu, 32 batch boyutu, 0,2 drop oranı ve 0,001 öğrenme faktörü ile belirlenmiştir. Ayrıca, evrişimli katmanlara ait filtre sayıları da 32, 64 ve 96 olarak sıralanmıştır. Bu hiper parametrelerin kullanılmasıyla model, 0,94869709 R2 skoru ve 0,000447747 MAE hata oranına ulaşmıştır. Bu sonuçlar, modelin veriye iyi uyarlandığını ve yüksek tahmin doğruluğu sağladığını göstermektedir.

Modelin son katmanı, istenen çıkış vektörünün boyutu olan 80 birim olarak belirlenmiştir. Ayrıca, modeldeki evrişimli katmanlar "Küçükten büyüğe" yaklaşımıyla sıralanmıştır. Bu yaklaşım, daha önceki katmanlarda yüksek düzeyde özellik çıkartımı yapılmasına olanak sağlayarak modelin daha iyi performans göstermesini destekler.

Son olarak, bu modelin evrişimli katmanlarının çıkışı, AutoEncoder yapısından esinlenerek oluşturulmuştur. AutoEncoder ağı, veriyi sıkıştırma ve ardından yeniden oluşturma prensibine dayalıdır. Modelimizde evrişimli katmanlarla elde edilen yüksek özellikli veriler, 3 farklı Yoğun katmana iletilir ve bu aşamada sıkıştırılarak 80 birim boyutunda çıktı elde edilir. Bu tasarım, veri temsiliyi iyileştirirken gereksiz bilgi kaybını önler.

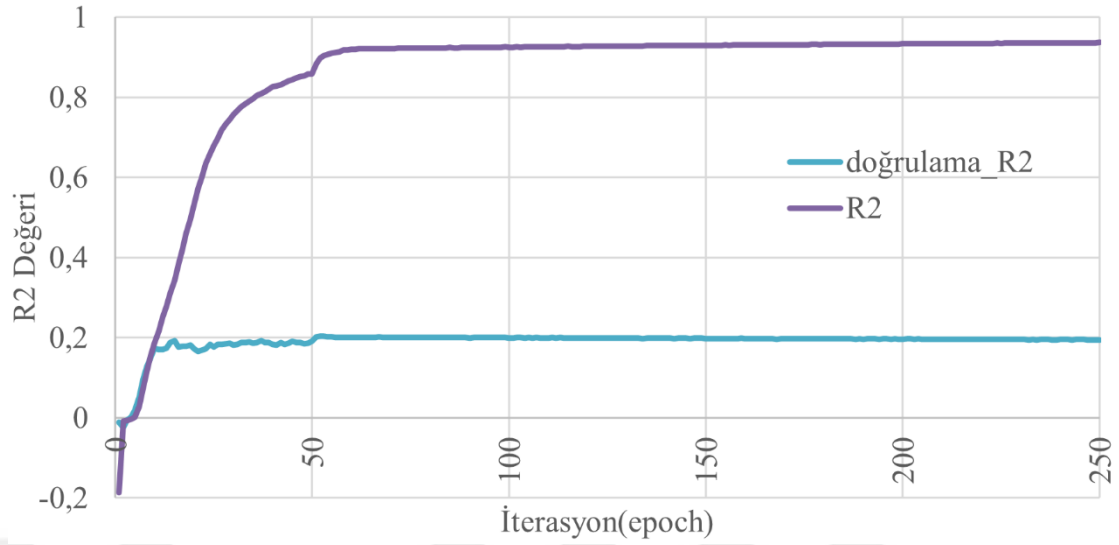


Şekil 3.27 Model 3 YSA şematik gösterimi

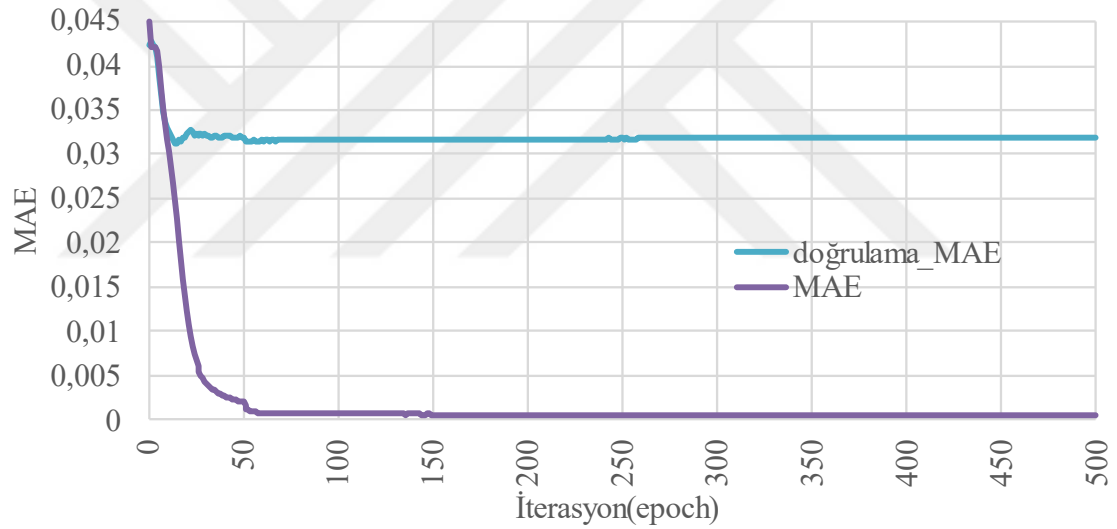


Şekil 3.28 YSA Model 3 en iyi beş hiper parametre sonuçları

YSA model 3 için oluşturulan MAE/doğrulama MAE ve R2 skoru/doğrulama R2 skoru şekillerinde açıkça görülmektedir ki modelin eğitim ve doğrulama verileri ile yapılan denemelerdeki veriye uyumsuzluk yüksek gerçekleşmiştir. Bu da modelin eğitim verilerine aşırı uyum gerçekleştirdiği, bu sebepten dolayı doğrulama verileri ile çalışmalarda başarılı sonuçlar verememesine sebep olmuştur.



Şekil 3.29 YSA Model 3'e ait R2 ve Doğrulama R2 sonuçları

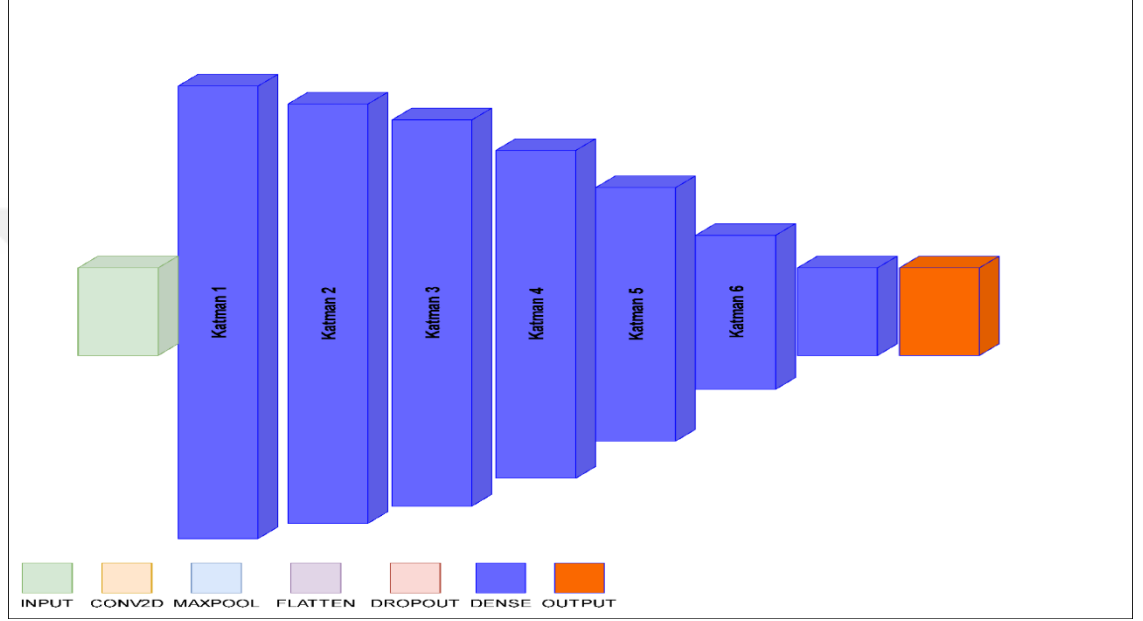


Şekil 3.30 YSA Model 3'e ait MAE ve Doğrulama MAE sonuçları

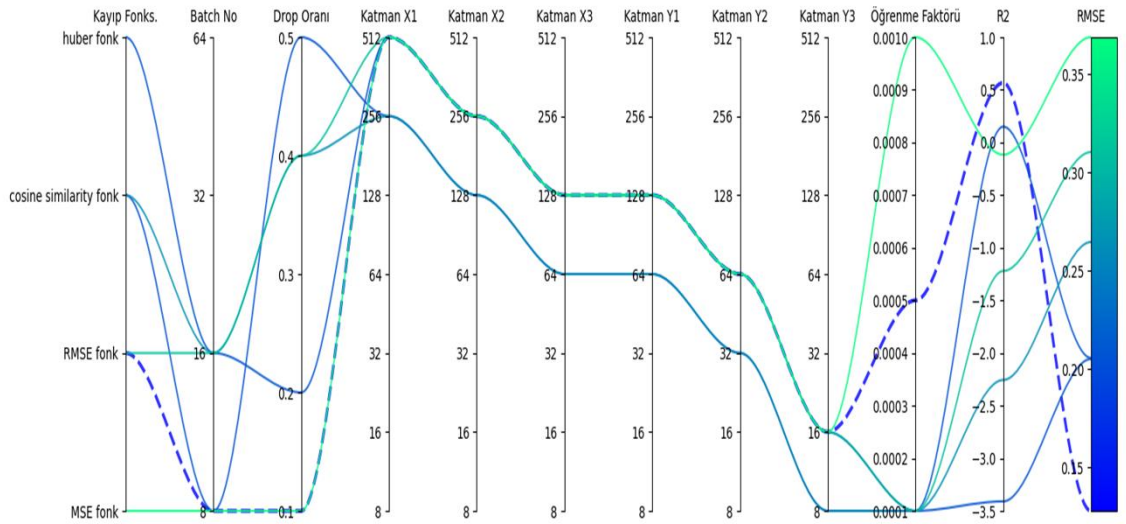
3.6.4. Yapay sinir ağı modeli 4

Bu Yapay Sinir Ağı (YSA) modeli, toplamda dokuz katmandan oluşmakta olup, altı yoğun katman içermektedir. Modelin yapısal düzeni, Şekil 3.31'de görsel olarak sunulmuştur. Hiper parametrelerin araştırılmasıyla elde edilen sonuçlar, Şekil 3.32'de sunulmuştur. Bu sonuçlara göre, en düşük RMSE kayıp miktarını sağlayan hiperparametre kombinasyonu şu şekildedir: Kayıp Fonksiyonu olarak RMSE fonksiyonu, Batch sayısı 32, Drop oranı 0,5 ve Öğrenme faktörü 0,0001 olarak belirlenmiştir. Ayrıca, Yoğun Katmanlar 1, 2, 3, 4, 5 ve 6 için belirlenen filtre sayıları sırasıyla 10000, 5000, 2500, 1200, 600 ve 300'dür. Bu hiper parametrelerle model, 0,747211099 R2 skoru ve 0,0316738 MAE hata oranı elde etmiştir. Bu sonuçlar, modelin yüksek performansını göstermektedir.

Modelin sonundaki tek yoğun katman, istenen çıkış vektör büyüklüğü olan 80 birim olarak belirlenmiştir. Model, AutoEncoder denilen YSA modelinden esinlenerek oluşturulmuştur. AutoEncoder ağırları, girdinin sıkıştırılması ve ardından yeniden oluşturulması işlemini gerçekleştiren iki temel parçadan oluşur: Kodlayıcı (Encoder) ve Çözücü (Decoder). Geliştirilen bu modelde, girdi katmanından gelen yüksek özellikli veriler, Kodlayıcı biçiminde düzenlenmiş altı ayrı yoğun katmana iletilir, her katman bir diğerinden daha küçük olduğu için katmanlar arasında veri hareket ederken giderek küçülmek zorunda kalmaktadır. Böylece girdi verisini en etkili bir şekilde temsil eden son katman büyüklüğünde yeni bir temsili veri elde edilmiş olur.



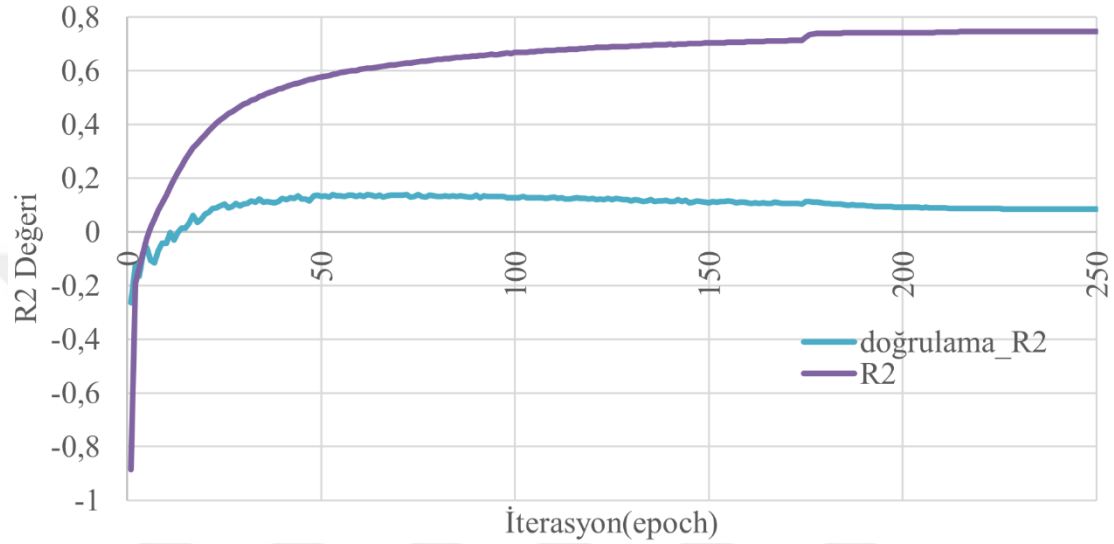
Şekil 3.31 Model 4 YSA şematik gösterimi



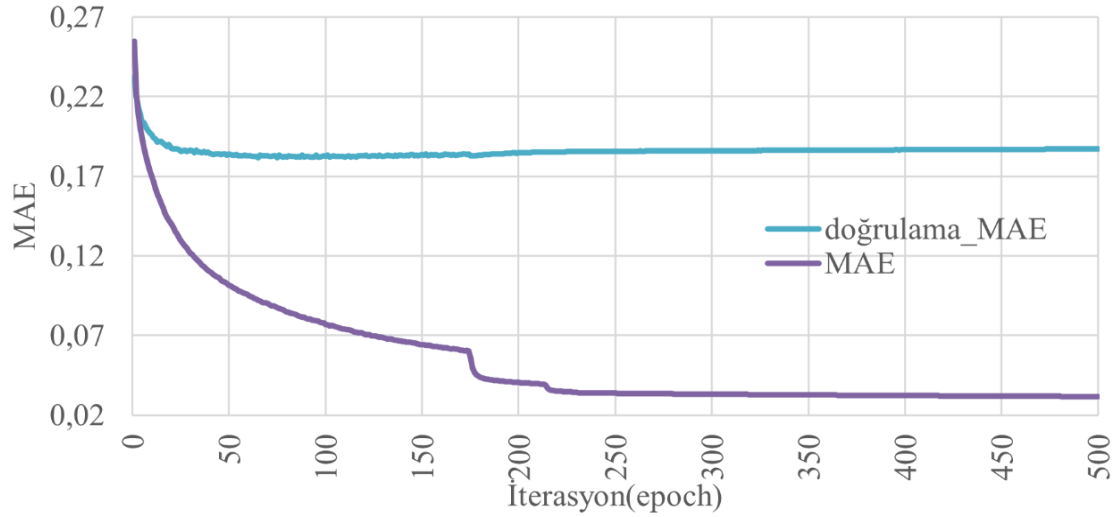
Şekil 3.32 YSA Model 4 en iyi beş hiper parametre sonuçları

YSA model 4'e ait Şekil 3.33'te ve Şekil 3.34'deki R2 ve MAE grafiklerinde bu değerlerin doğrulama karşılıkları ile aralarındaki farklılıkların iterasyon ilerledikçe giderek açıldığı daha sonra da sabit bir fark değerinde kaldığı açıkça görülmektedir.

Değerlerdeki bu hareket YSA model 4'ün eğitim veri seti ile aşırı uyum sağlayıp, doğrulama verilerini tanımlayamadığını açık bir şekilde göstermektedir. Grafikler incelendiğinde, gerçek ve değerlendirme verileri arasındaki bu farklılıkların diğer YSA modellerine göre daha fazla olduğu görülmektedir.



Şekil 3.33 YSA Model 4'e ait R2 ve Doğrulama R2 sonuçları



Şekil 3.34 YSA Model 4'e ait MAE ve Doğrulama MAE sonuçları

3.6.5. Yapay sinir ağı modeli 5

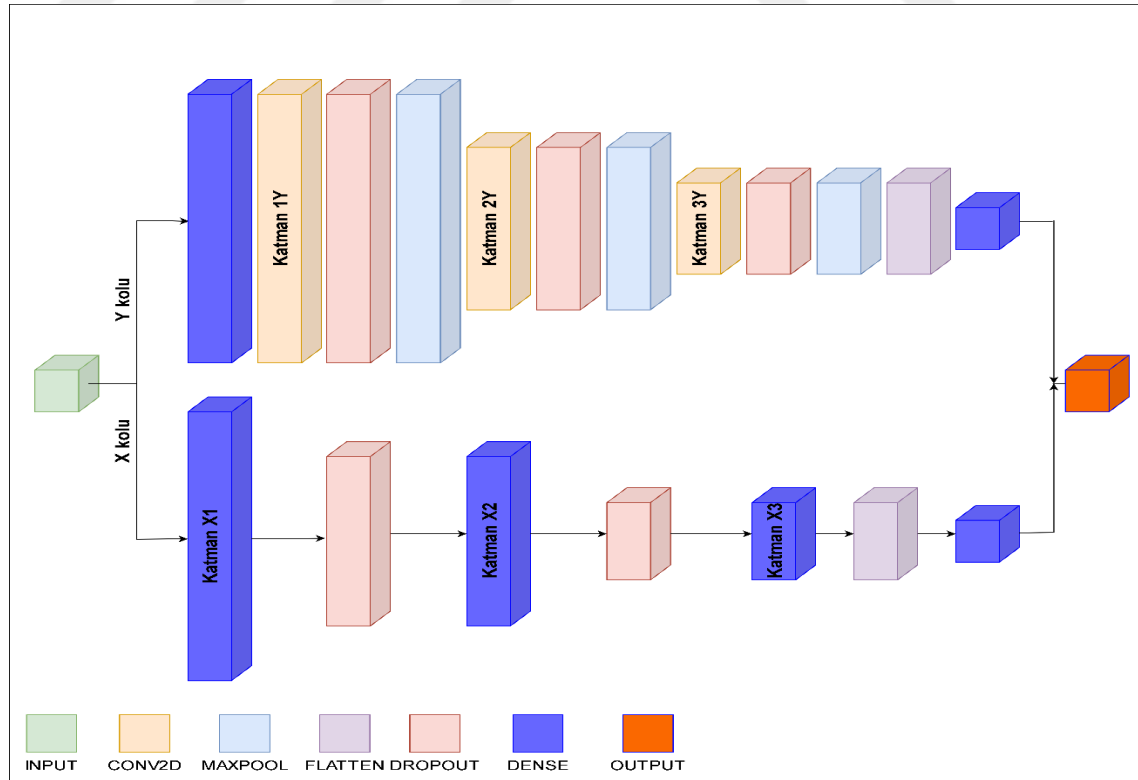
Bu Yapay Sinir Ağı (YSA) modeli, X ve Y isimleri verilmiş iki ayrı hat üzerinden oluşmaktadır. Şekil 3.35'de gösterilen model dizilimine göre, X hattı yoğun katmanlardan oluşan bir encoder ağını içermekte, Y hattı ise Büyükten Küçüğe (Deep to Shallow)

Yaklaşım kullanarak sıralanmış evrişimli katmanlardan oluşmaktadır. Bu yapı, modelin iç yapısını iki farklı yaklaşımı birleştirerek tanımlamaktadır. Bir dizi temel öğrenme algoritmasını bir araya getirerek kurulmuş bu tip modellere Ensemble modeller de denilmektedir (Ganaie vd. 2022).

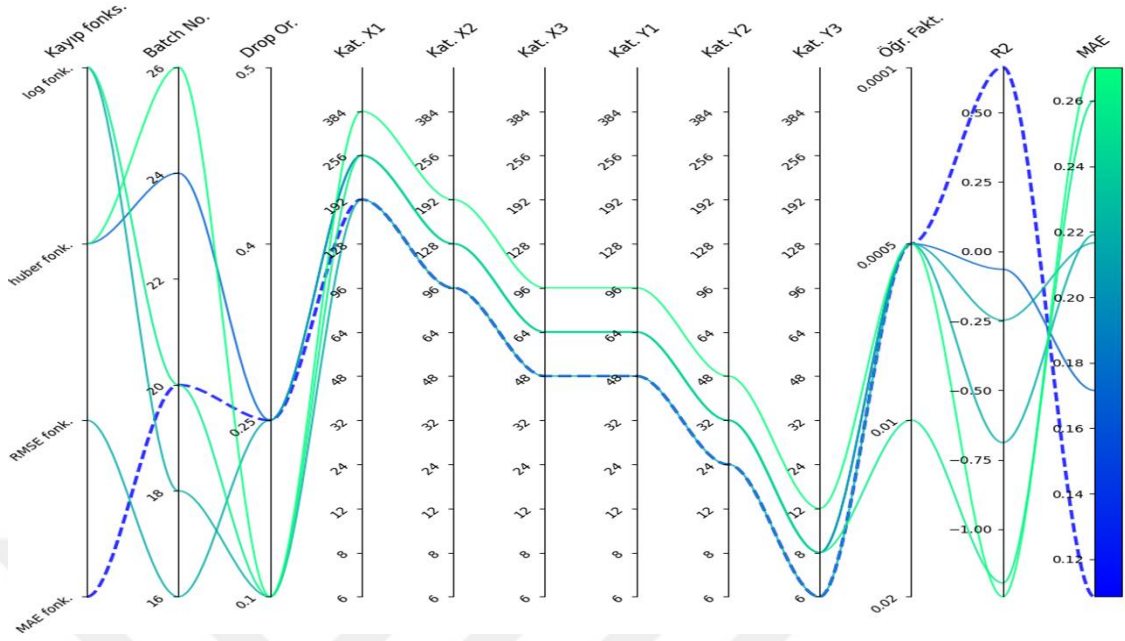
X hattı, AutoEncoder modellerinin kodlayıcı kısmından esinlenerek oluşturulmuştur. AutoEncoder ağları, girdinin eleman sayısını azaltan bir kodlayıcı (encoder) kısmı ve eleman sayısını artıran bir çözücü (decoder) kısmı içerir. Geliştirilen modelde, girdi katmanından gelen veriler 6 ayrı yoğun katmana yönlendirilerek 80 birim boyutunda sıkıştırılmış bir çıktı elde edilir. Bu, verinin yüksek özellikli temsilini yakalamak için kullanılan bir yöntemdir.

Y hattı üzerindeki evrişimli katmanlar birbirine benzer olarak boyutlandırılmış, ancak farklı filtre sayılarına sahiptir. Bu katmanlar, Büyükten Küçüğe Yaklaşımı kullanarak sıralanmıştır. Bu yaklaşım, modelin daha derin katmanlardan başlayarak özelliklerini çıkarmasına ve daha sonra daha geniş katmanlarda bu özellikleri işlemesine olanak tanımasını ve böylece performans artışı hedeflenmiştir.

Hiper parametrelerin araştırılmasıyla elde edilen sonuçlar, Şekil 3.36'da sunulmuştur. Modelin performansı ile ilgili olarak, en düşük MAE kaybını veren hiperparametre kombinasyonu belirlenmiştir. Bu kombinasyonda MAE kaybı kullanılmış, batch sayısı 8, drop oranı 0,1, ve öğrenme faktörü 0,0005 olarak ayarlanmıştır. X hattı için yoğun katmanların genişlik sayıları sırasıyla 512, 256 ve 128 olarak belirlenmiştir. Y hattı için evrişimli katmanların filtre sayıları sırasıyla 128, 64 ve 16 olarak belirlenmiştir. Bu hiper parametrelerle model, 0,619779944 R2 skoru ve 0,102063134 MAE hata oranına ulaşmıştır.

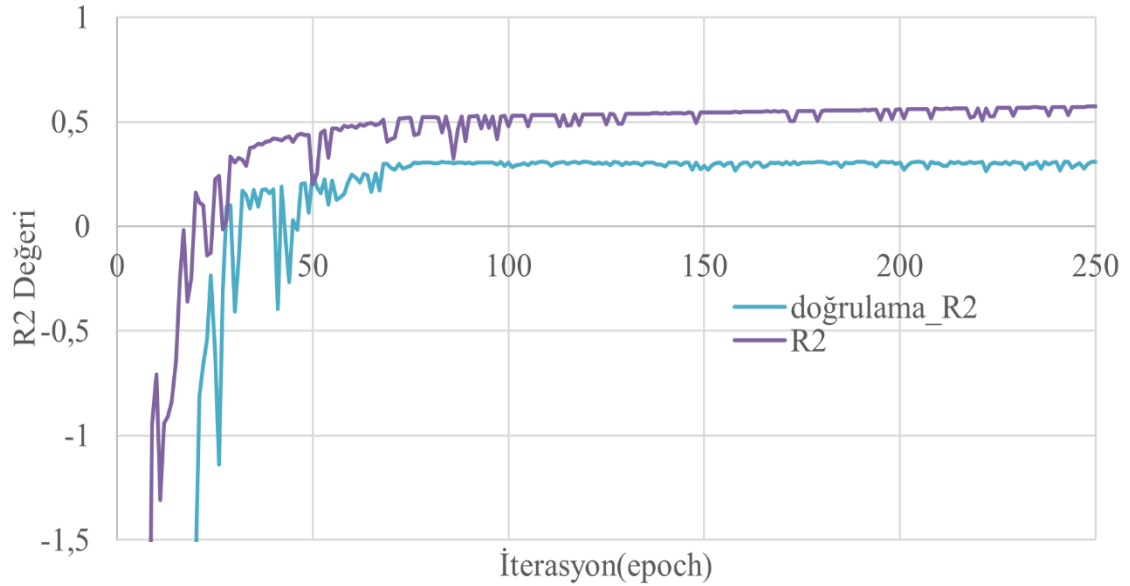


Şekil 3.35 YSA Model 5 Grafikselsel Gösterimi

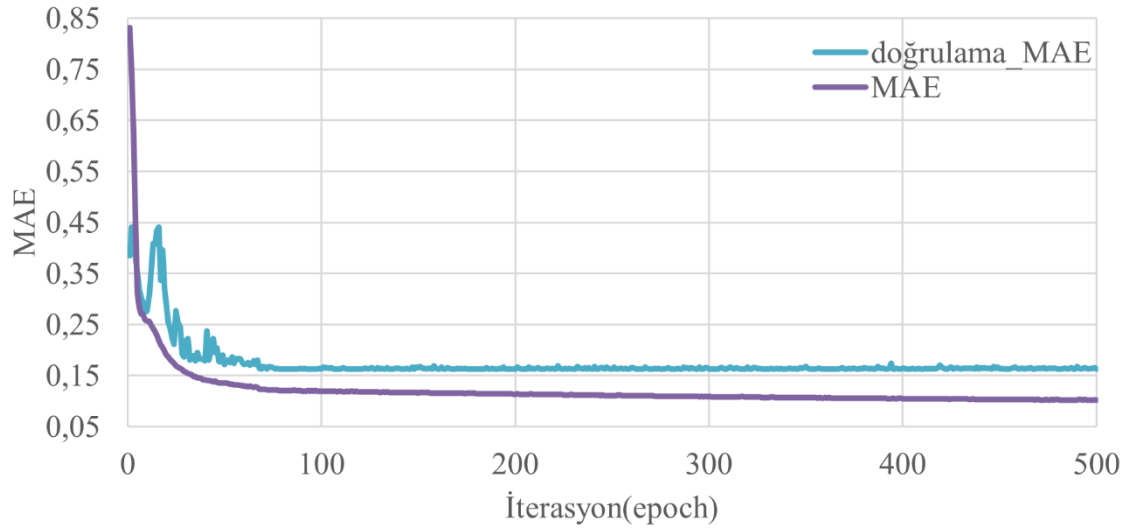


Şekil 3.36 YSA Model 5 en iyi beş hiper parametre sonuçları

YSA model 5'e ait R2 skor ve MAE hata oranlarının doğrulama değerlerine göre karşılaştırmalarını içeren Şekil 3.37 ve Şekil 3.38 grafiklerini incelediğimizde bu modelin eğitim veri seti ile doğrulama veri setine uyumunun diğer YSA modellerine göre çok daha dengeli gerçekleştiğini ayrıca R2 skoru değerine bakarak eğitim ve doğrulama verilerine daha başarılı bir şekilde tepki verdiği görülebilmektedir. Ayrıca, doğrulama değerlerinin metrikler ile yakın paralel bir biçimde seyretmesi YSA modelinin diğer modellere göre aşırı uyum gerçekleşmeden eğitim işlemini tamamladığı söylenebilir.



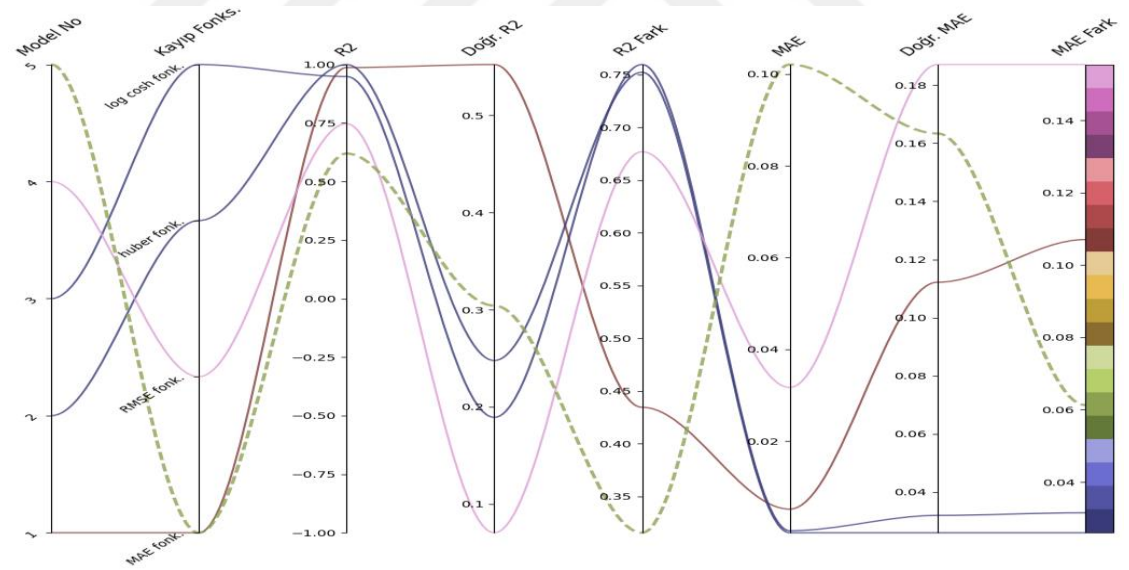
Şekil 3.37 YSA Model 5'e ait R2 ve Doğrulama R2 sonuçları



Şekil 3.38 YSA Model 5'e ait MAE ve Doğrulama MAE sonuçları

3.7. Modellerin Karşılaştırılması

Geliştirilen tüm modellerin belirlenmiş metrikler altındaki sonuçları Şekil 3.39'da verilmiştir. Modellerin analizi için kullanılan sekiz farklı kayıp fonksiyonu içerisinde sadece üçü; huber, logcosh ve MAE fonksiyonları en iyi sonuçlar arasına girebilmişlerdir.



Şekil 3.39 Tüm YSA modellerinin kendileri arasında karşılaştırmaları

Model 1, derin evrişimli ağ kullanımıyla kompleks özellikleri çıkartabilen bir yapıya sahiptir. Ancak, yüksek eğitim R2 değeri ve düşük doğrulama R2 değeri, modelin aşırı uyuma eğilimli olduğunu göstermektedir. Ayrıca, modelin karmaşıklığı eğitim süresini uzatabilir.

Çizelge 3.5 YSA Model 5 en iyi beş hiper parametre çizelgesi

Model 5 – Hiper parametre çizelgesi			
Kayıp Fonk.	MAE	Katman Sayısı (Y2)	24
Batch No.	20	Katman Sayısı (Y3)	6
Katman Sayısı (X1)	192	Öğrenme Faktörü	0,0005
Katman Sayısı (X2)	96	R2	0,662920177
Katman Sayısı (X3)	48	MAE	0,108563229
Katman Sayısı (Y1)	48		

Model 2, Huber kaybı kullanarak aykırı değerlere karşı dirençli olabilir. Yüksek eğitim R2 değeri ve düşük doğrulama R2 değeri, aşırı uyuma işaret ediyor. Ayrıca, yüksek drop oranı genel performansı etkileyebilir.

Model 3'ün küçük filtre boyutları, lokal özellikleri çıkartabilme avantajına sahiptir. Ancak, yüksek eğitim R2 değeri ve düşük doğrulama R2 değeri, aşırı uyuma işaret edebilir. Ayrıca, avg pooling önemli özellikleri kaybetme riskini taşır.

Basit bir yapıya sahip olan Model 4, eğitim süresini diğer modellere göre ciddi oranda kısaltmıştır. Ancak, düşük R2 değeri modelin veriyi iyi açıklayamadığını göstermekte ve basit yapısı karmaşık özellikleri çıkartmakta zorlandığı anlaşılmaktadır.

Model 5, hem tam bağlı hem de evrişimli katmanları birleştirerek çok yönlü özelliklerin çıkartılması üzerine odaklanılmış bir modeldir, ancak model karmaşıklığı diğer modellere göre kısmen eğitimin süresinin uzamasına ve R2 ve MAE metriklerindeki doğrulama değerleri ile aralarının açılmasına sebep olmuştur.

Doğrulama R2/R2 farkının mutlak değeri incelendiğinde, en büyük farka sahip model Model 3'tür (0,76). Model 2'nin 0,7518 ve Model 4'ün 0,677 mutlak fark değerleri ile geldiği görülmektedir. Model 1'in 0,435 ve Model 5'in 0,316 mutlak fark değerleri ile daha düşük farklara sahip olduğu gözlemlenmektedir.

Detaylı analiz, Model 2'nin düşük MAE değeri ve yüksek R2 değeri ile iyi performans sergilemesine rağmen, aşırı uyuma eğilimli olduğunu ortaya koymaktadır. Model 3 ve Model 1, benzer şekilde aşırı uyuma eğilim gösterirken, Model 4 basit yapısıyla karmaşık verileri iyi açıklayamadığı metriklerden anlaşılmaktadır. Model 5 ise çoklu özellik çıkartma avantajına rağmen, karmaşıklığı nedeniyle aşırı uyum riski taşımaktadır. İncelenen beş model arasında verilere uyum ve en yüksek tahmin doğruluğu bakımından sıralandığında seçilebilecek en iyi modelin model 5 olduğu anlaşılmaktadır.

3.8. Yapay Sinir Ağlarının İteratif Çözüm Teknikleri ile Birleştirilmesi

Belirli bir Alan ile ilgili spesifik denklem çözümleri geliştirmenin temel amacı, denklem çözme sürecini problemin doğrudan çözümlere entegre edilmiş bilgi ile hızlandırmaktır. Bu nedenle, özel amaçlı denklem çözümleri, belirli yapı tipleri için optimize edilmiş bir şekilde tasarlanabilir (örneğin, kafes yapılar). Fakat bu çözümler farklı yapı tipleri ile karşılaştıklarında başarı oranları ciddi miktarda düşebilmektedir.

Bu tez ile açıklanan teknikteki temel fikir, sinir ağlarının, görece az hesaplama gereksinimi ile hızlı bir şekilde karma yapıların yaklaşık deplasmanlarını tahmin etme yeteneğini kullanmaktır. Ancak, bu sinir ağları net sonuçlar üretmediği için, "kesin" deplasmanlara nihai olarak ulaşılabilir geniş kapsamlı bir algoritmik çerçeveye entegre edilmeleri gerekmektedir. Bu amaçla soruna yaklaşırsak, iteratif denklem çözme yöntemleri, böyle bir çerçevenin oluşturulması için doğal bir seçenek olarak ortaya çıkmaktadır. Bu çalışmada, daha önce bahsi geçen 5 farklı YSA içerisinde seçilen model 5, önceden koşullandırılmış iteratif algoritma ile birleştirilerek (BTO), bütünlük bir YSA-BTO algoritması oluşturulmuştur. Bu bütünlük yaklaşımında, sinir ağları her iterasyonda yaklaşık deplasmanları hesaplarken, bu sırada genel iteratif BTO süreci, kesin çözüme doğru yaklaşmayı kolaylaştırır.

3.8.1. Biyocoğrafya tabanlı optimizasyon (BTO) algoritması

Biyocoğrafya Tabanlı Optimizasyon (BTO), doğadan ilham alınmış bir optimizasyon algoritmasıdır, biyolojik organizmaların coğrafi dağılımıyla ilgili ilkeleri temel alır (Simon 2008). Bu yöntem, bir optimizasyon problemine farklı habitatları temsil eden potansiyel çözüm seçenekleri olarak bakar. Algoritma, bu habitatlar arasındaki uygunluk değerlerine dayalı olarak türlerin (çözüm bileşenleri) göçünü simüle eder. Amacı, gerçek ekosistemlerdeki gibi, bu çözümlerin uygunluklarını zaman içinde uyum sağlamalarını ve evrimleşmelerini teşvik ederek artırmaktır.

Bu yöntemde, bireyler, Habitat Uygunluk İndeksi (HUI) tarafından belirlenen şekilde ya göç eder ya da mevcut bölgelerinde kalma eğilimindedirler. Aynı zamanda Duyarlılık İndeksi Değişkeni (SIV) tarafından yönlendirilen dış dağılım etkileri sayesinde diğer bölgelere göç etme eğilimi de gösterirler. Bu hareketlerle, yalnızca tek bir çözüm noktasına odaklanmak yerine, BTO yaklaşımı belirli dış etkiler tarafından yönlendirilen farklı çözüm bölgelerine odaklanmaya yönelir.

BTO'da bu dış etkiler, farklı çözüm bölgelerinin uygunluğunu etkileyen çeşitli çevresel faktörleri temsil eder. Bir noktaya odaklanmak yerine bireyler (çözümler) çok sayıda bölgeyi keşfeder, türlerin çeşitli habitatları keşfetmesine benzer bir şekilde. Bazı bölgeler optimizasyon için daha elverişli koşullar sunabilirken, diğerleri zorluklar sunabilir. Bu geniş keşif, optimizasyon arayışını çeşitlendirir ve daha optimal bir çözüm bulma olasılığını artırır. Dış ipuçlarına yanıt vererek bireyler, çevresel değişikliklere yanıt olarak yeni bölgeleri keşfetmeye teşvik edilir, bu da yerel optimumlarda sıkışmayı önler.

Özetle, BTO'nun özü, dış etkiler tarafından yönlendirilen farklı çözüm bölgelerini adaptif bir şekilde keşfetme yeteneğindedir. Bu yaklaşım, tek bir noktaya takılı kalmadan ve optimizasyon sürecinin çeşitli çözüm alanlarını dolaşmasına izin vererek, daha optimal çözümlerin keşfedilmesini teşvik eder.

Standart BTO sürecinin matematiksel modellemesi, göç ve mutasyon olmak üzere iki temel aşamayı içerir. Göç aşaması sırasında, bir habitat içindeki bireyler belirli bir nüfus içinde başka bir habitata taşınır. Göçleri sırasında nüfus içindeki her birey, dış etkilerin bir sonucu olarak değişir (tasarım değişkeni). Bu sürecin bir sonucu olarak göç eden nüfus, değişen çözüm olarak adlandırılır ve göçü alan habitat referans çözüm olarak adlandırılır. Referans çözümü, rulet yöntemi kullanılarak nüfus içindeki çözümler arasından seçilir. Her habitatın (çözümün) referans çözüm olma olasılığı, göçü alan

habitatın göçü alması olasılığı oranına bağlıdır ve aşağıdaki formülle hesaplanır:

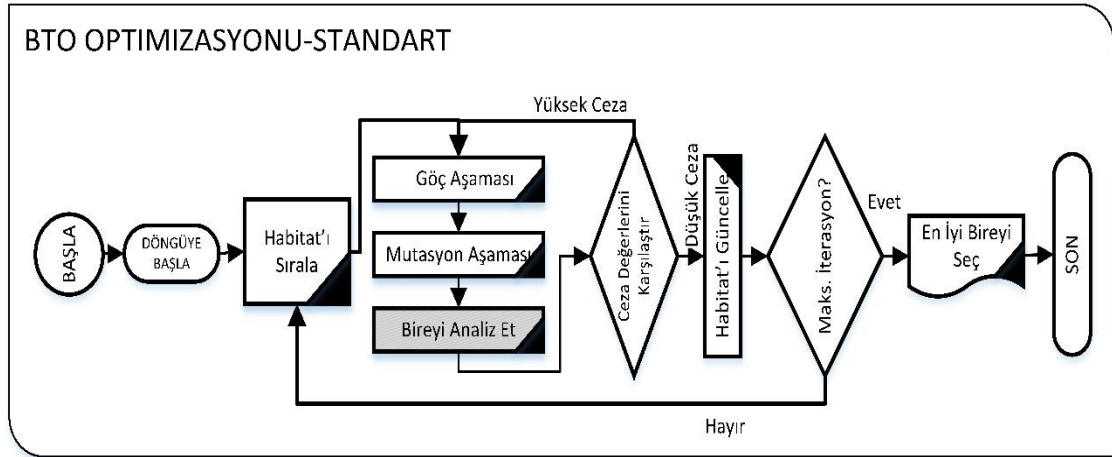
$$P(x_j) = \frac{\mu_j}{\sum_{i=1}^N \mu_i}; j = 1, \dots, PS \quad (3.25)$$

Burada, N habitat içinde bulunan bireylerin sayısını, " μ_j " habitat için göç katsayısını ve PS nüfus büyüklüğünü ifade etmektedir.

Mutasyon aşamasında, göç eden nüfus içindeki bireyler dış etkiler nedeniyle mutasyona uğrar ve ardından diğer habitatlara eklenirler. BTO algoritmasında, mutasyon süreci, üretilen rasgele sayının daha önce tanımlanmış bir mutasyon oranının altına düşmesi durumunda meydana gelir. Mutasyon gerçekleşirse, göç aşamasında mutasyona uğrayan tasarım değişkeni rastgele belirlenir.

$$x_i = x_{li} + rand(0,1)(x_{ui} - x_{li}); i = 1, \dots, PS \quad (3.26)$$

Burada, " x_i " mutasyona uğrayan çözümün tasarım değişkenini, " x_{ui} " ve " x_{li} " sırasıyla değişkenin üst ve alt tasarım sınırlarını ve PS nüfus büyüklüğünü ifade etmektedir. "rand(0,1)" işlevi, 0 ile 1 arasında rasgele bir değer atayan bir işlevdir.



Şekil 3.40 Standart BTO optimizasyon algoritması akış diyagramı

Çalışmanın bağlamında çelik çerçeve ve kafes sistemlerin FEM analizi, Şekil 3.40'da algoritmik akış adımlarının görselleştirildiği şekilde BTO optimizasyonu ile entegre edilmiştir. Ardından, işlem sürecinin döngüsel doğası, Algoritma 1 kullanılarak ardışık aşamalar dizisine dönüştürülmüştür. Bu aşamalar, Göç, Mutasyon gibi önemli BTO prosedürlerini, aynı zamanda Yapısal Analiz ve Değerlendirme aşamalarını içerir.

Yapı Analiz modülü YSA ve klasik çözüm işlemlerinin karşılaştırılabilmesi için aynı yapı üzerine her iki tip de çözümün gerçekleştirildiği bir algoritma geliştirilmiştir. Modül içerisinde Önceki çalışmalarda eğitimi tamamlanan en iyi YSA modeli olan YSA model 5'in ağ yapısını kullanarak deplasman hesaplayan bir fonksiyonun bulunmasının yanı sıra, klasik yöntem ile FEM çözümü yaparak yapının deplasmanına ulaşan bir fonksiyon daha bulunmaktadır. BTO döngüsü esnasında belirlenen her bir habitat küme elemanının

çözümleri bu modül içerisinde gerçekleştikten sonra optimizasyon işlemi sonunda analizleri yapılmak üzere çözüm süreleri ve diğer detaylar ile birlikte kaydedilmektedir.

ALGORİTMA 4: GİRDİNİN SIKIŞTIRILMASI İŞLEMİ SÖZDE KODU

Girdi: KM→*Rijitlik Matrisi, PV*→*Yük Vektörü*
Çıktı: KP→*YSA Girdi Matrisi, Coeff*→*Genişletme Katsayı Matrisi,*
scale→*Genişletme katsayısı*

```

1  KN←KM ile PV birleştir
2  filter_1←PV içerisindeki sıfırların indeksleri
3  döngü (i→uzunluk(filter_1)) //filter_1 eleman sayısı kadar tekrar et
4      zindex←filter_1[i]
5      döngü (j→uzunluk(KP[0])) //KP'nin sütun sayısı kadar tekrar et
6          | KP[i,j]←KP[i,j]-KP[i,zindex]*(KP[zindex,j]/KP[zindex,zindex])
7          son
8      KP içerisinde filter_1 indeks numaralı satır ve sütunları sil
9  son
10 KP matrisine normalizasyon işlemi gerçekleştir
11 KN←KP matrisini PV vektörüne böl ve ayır
12 Eğer (boyut(KN) < 80x80) // eğer KN boyutları 80 birimden küçük ise(Şekil 3.17)
13     | döngü (i→uzunluk(KN[0])/80) // 80'in katı kadar tekrar et
14         | KN←KN matrisini sona ekle
15     son
16 son
17 Eğer (boyut(KN) > 80x80)
18     | Libscipy.ndimage kullanarak KN matrisini küçült
19 son
```

3.8.2. Geliştirilen BTO optimizasyon algoritma sonuçlarının karşılaştırılması

Bu bölümde Geliştirilen BTO algoritmasıyla sadece FEM yöntemi kullanarak elde edilen sonuçlarının optimalliğinin karşılaştırılabilmesi için Şekil 3.42'te gösterilen 18 çubuklu konsol kafes örneği kullanılmıştır.

18 çubuklu konsol kafes problemi, daha önce (Imai ve Schmit Jr 1981) tarafından ve (Lee ve Geem 2004) tarafından analiz edilen bir boyutlandırma optimizasyon problemidir. Malzeme yoğunluğu 0.1 lb/in^3 'tür. Çubuklardaki gerilimi 20000psi'yi geçmemelidir (germe ve basınçta aynı limit). Euler burkulma dayanımına ilişkin kısıtlamalar da göz önünde bulundurulmuştur. Her bir yapı elemanı için izin verilen limit şu şekilde hesaplanır:

$$\sigma_i^b = -\frac{KEA_i}{l_i^2}; i = 1, \dots, ES \quad (3.27)$$

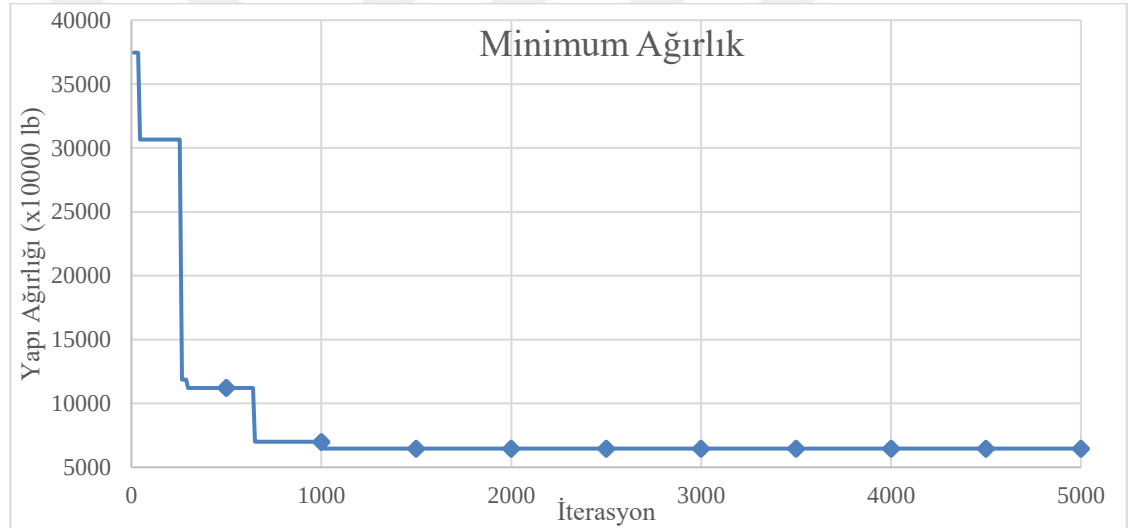
Burada K burkulma sabitidir ($K = 4$). E elastisite modülüdür ($E = 10000\text{ksi}$) ve l_i^2 elemanın uzunluğudur. $F = 20\text{kips}$ aşağı yönlü yük (1), (2), (4), (6) ve (8) düğümlerindedir. Eleman kesit grupları aşağıdaki gibidir:

Birinci Grup; “9,12,4,16” , ikinci Grup; “6,10,14,18”, üçüncü grup; “7,11,15”, dördüncü grup; “9,13,17”

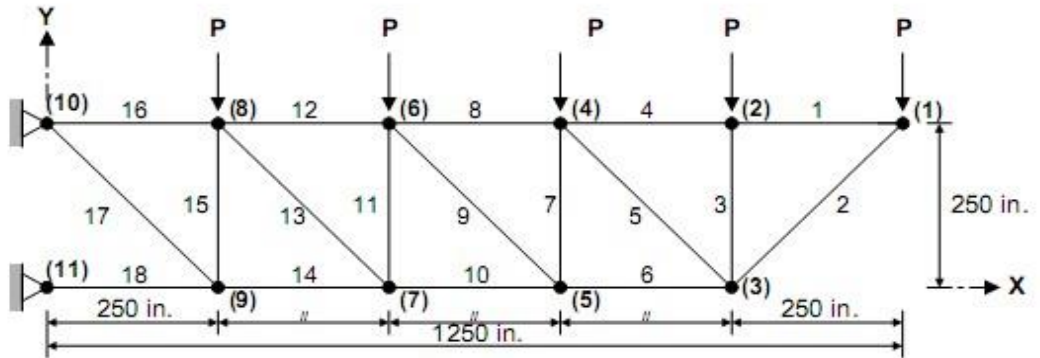
Çizelge 3.6 18 çubuklu konsol kafesin optimum tasarım karşılaştırma sonuçları

Grup No	Imai and Schmit Çalışma Sonuçları	Lee and Geem Çalışma Sonuçları	Geliştirilen BTO Algoritması
1	9,998	9,98	319,97
2	21,65	21,63	190,38
3	12,5	12,49	18,89
4	7,072	7,057	20,71
Ağırlık (lb)	6430	6421,88	6464,89

Minimum kesit alanı $0,1 \text{ in}^2$ 'dir. Bu çalışma ile geliştirilen BTO algoritması sonuçları ile akademik çalışmalarla elde edilen sonuçların karşılaştırılması Şekil 3.41'de sunulmuştur.



Şekil 3.41 18 çubuklu konsol kafes örneği BTO arama geçmişi



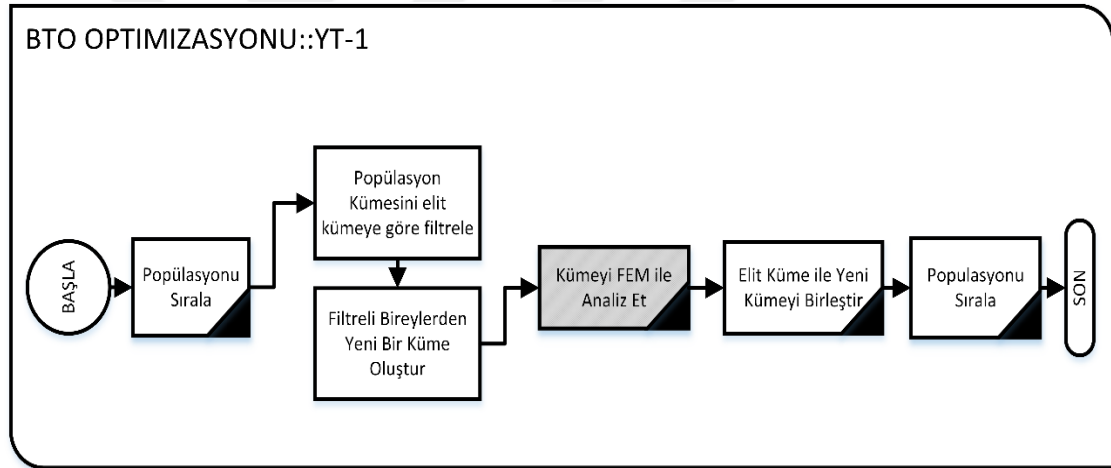
Şekil 3.42 18 çubuklu düzlem kafes yapı örneği

3.8.3. Geliştirilen BTO arama yöntemleri

Çalışma ile geliştirilen YSA analiz metodu tarafından bulunan analiz sonuçlarının döngü içerisinde doğruluk oranlarının tespit edilmesi ve optimizasyon işlemi sonucu bulunan yapıların doğrulanması amacıyla BTO içerisine entegre edilebilen üç farklı arama yöntemi geliştirilmiştir. Bu yöntemler, Şekil 3.40’da tarif edildiği üzere her iterasyonun bitiminde Birey Analizi aşaması yerine döngüye dahil olacak şekilde konumlandırılmışlardır.

3.8.3.1. BTO-Yöntem.V1 arama yöntemi

BTO-Yöntem-1 (YT-1), Şekil 3.43’de görüldüğü gibi bir dizi sıralı aşamadan oluşan bir optimizasyon arama yöntemidir. İlk adımda, her iterasyonda BTO tarafından oluşturulan ve YSA ile yapısal analizleri yapılan popülasyondaki bireyler, elit kümenin en düşük fitness değeriyle karşılaştırılır ve daha düşük değere sahip olanlar başka bir kümeye taşınır. Ardından, bu yeni kümedeki bireyler fitness değerlerine göre sıralanır ve elit kümeye girebilecek fitness değerine sahip olanlar, FEM kullanılarak yeniden yapısal analize tabi tutulur ve elit kümeye eklenir. Her BTO döngüsünün sonunda bu süreç tekrarlanarak optimizasyon süreci tamamlanır.



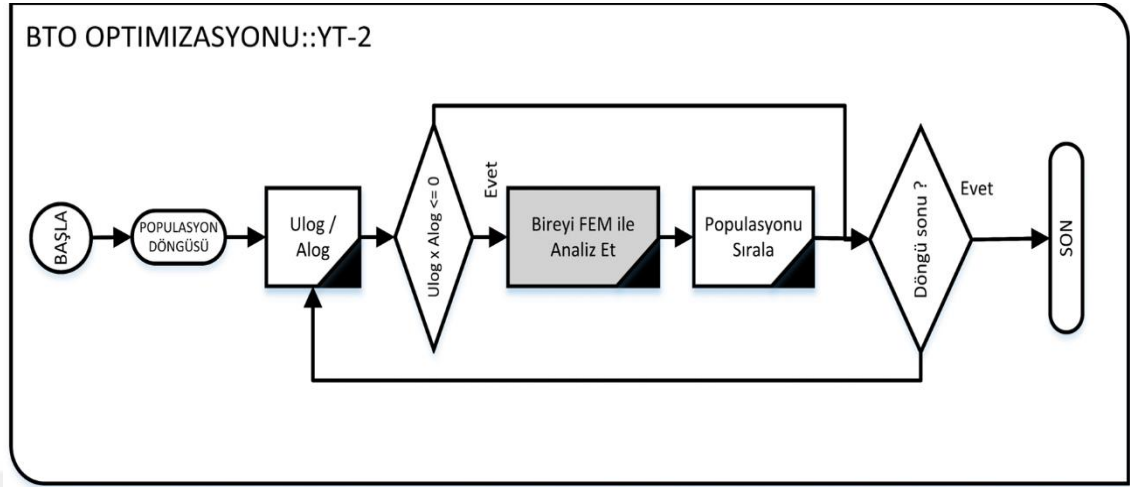
Şekil 3.43 BTO-Yöntem-v1 Arama Yöntemi akış diyagramı

3.8.3.2. BTO-Yöntem.v2 arama yöntemi

BTO-Yöntem-2 (YT-2) ‘ye ait işleyiş şeması Şekil 3.44’de gösterilmektedir. YT-2 bir önceki arama yöntemi olan YT-1 ‘de olduğu gibi YSA yardımıyla analizleri gerçekleştirilmiş popülasyon kümesinin sıralanması tamamlandıktan sonra devreye girecek şekilde konumlandırılmıştır. Temelde YT-2 arama algoritmasıyla amaçlanan, popülasyon içerisindeki bireylerin YSA çözümlerine ait doğruluk oranlarını tespit edip, bu oranları kullanarak popülasyon içerisindeki her bir bireye ait Üst ve Alt doğruluk limitlerini belirleyerek bu değer dışında kalan bireyleri tespit etmektir.

Sonraki aşamada, YSA yapı çözümlerinin doğrulukları limitler dışında kalan bu bireyler FEM ile tekrar çözüp popülasyon içerisine eklenmektedir. Böylece YSA ‘ya ait yanlış tahminlerinin popülasyon kümesini aşırı kirleterek popülasyonun optimal minimum

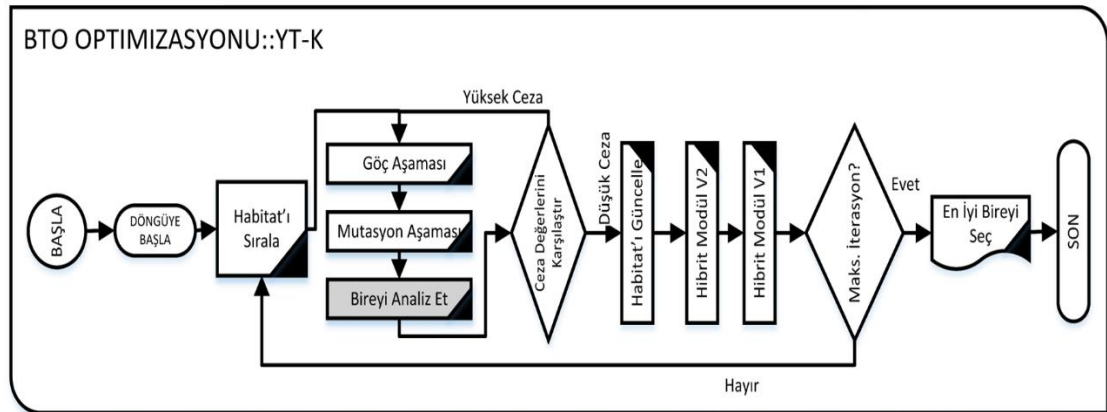
noktayı kaçırmamasına engel olunması amaçlanmıştır. Her BTO döngüsünün sonunda bu süreç tekrarlanarak optimizasyon süreci tamamlanır.



Şekil 3.44 BTO-Yöntem-v2 Arama Yöntemi akış diyagramı

3.8.3.3. BTO-Yöntem-Kombinasyon arama yöntemi

YT-K arama algoritması YT-1 ve YT-2 ‘de olduğu gibi YSA yardımıyla analizleri gerçekleştirilmiş popülasyon kümesinin sıralanması tamamlandıktan sonra devreye girecek şekilde konumlandırılmıştır. YT-K arama algoritması temelde YT-2 algoritmasından hemen sonra YT-1 algoritması sokularak BTO içerisinde bir iterasyonun tamamlanmasından oluşmaktadır. Bu sıralamadaki amaç, YT-2 arama yönteminin popülasyondaki sonuçlarının doğruluk oranlarını artırıcı etkisini önden kullanıp, daha sonra YT-1 arama yönteminin daha belirgin bir popülasyon içerisinde çalışması amaçlanmıştır. YT-K Arama yönteminin BTO algoritması içerisinde konumlandırılmaları Şekil 3.45’de gösterilmektedir.



Şekil 3.45 BTO-Yöntem-Karma Arama Yöntemi akış diyagramı

4. BULGULAR VE TARTIŞMA

4.1. Kullanılan Örnekler

Çalışma kapsamında geliştirilen Yapay Sinir Ağları (YSA) modelleri, yedi farklı örnek tipine ait binlerce yapı kullanılarak eğitilmiştir. Bu yedi örnek tipinden her birine ait birer örnek seçilerek, BTO kapsamında geliştirilen ve Bölüm 3.8.2 ‘de ayrıntıları sunulan arama yöntemlerinin performansı değerlendirilmiştir. Kullanılan bu örnekler ile ilgili bilgiler Çizelge 4.1 Tablosunda detaylı olarak verilmiştir.

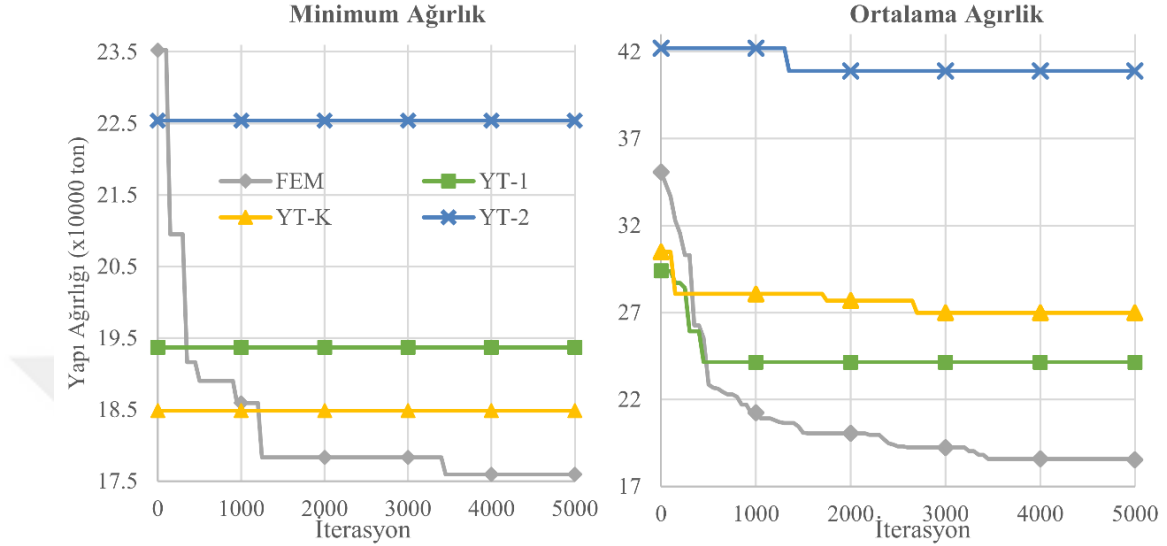
Çizelge 4.1 BTO yöntemleri ile kullanılan örneklerin yapısal özellikleri

Tip Adı/Kısaltması	Yapıdaki Düğüm Sayısı	Yükseklik/Uzunluk	Mesnet Sayısı	Yapı Serbestlik Derecesi
Ankastre Kiriş (AK)	162	55/4374	2	320
Ankastre Kiriş 2 (AK2)	161	320/6400	2	318
Dar Kule (DK)	82	1760/357	2	406
Geniş Kule (GK)	85	2048/1588	5	415
Klasik Köprü (KK)	168	856/9890	7	322
Klasik Köprü 2 (KK2)	172	26/5307	9	326
Piramit (P)	105	754/78	14	182

Çizelge 4.1’de sunulan yapı özelliklerini içeren örnekler üzerinde, BTO’nun dört farklı hibrit versiyonu kullanılarak yapıların optimum tasarımı incelenmiştir. Elde edilen sonuçlar, hibrit arama yöntemlerinin yakınsama performansı, optimum sonuçlara erişim yeteneği, istatistiksel güvenilirlik ve optimum çözüme ulaşma hızı açısından aşağıdaki alt başlıklar altında karşılaştırılmıştır.

4.2. BTO Arama Yöntemlerinin Yakınsama Açısından Karşılaştırılması

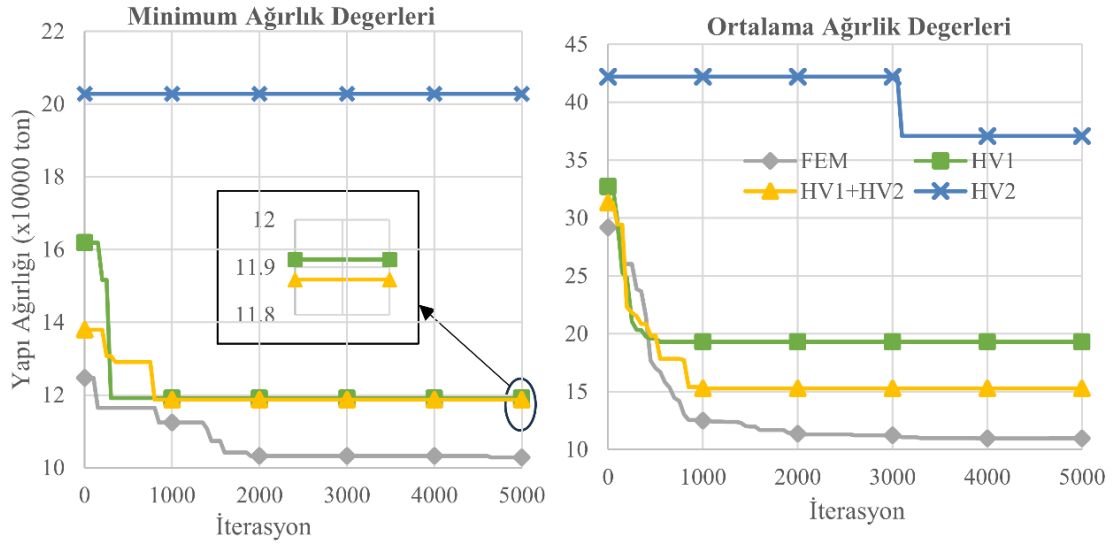
Aşağıdaki yedi farklı örneğe ait grafikler BTO optimizasyon yöntemi içerisinde FEM, YT-1, YT-2 ve YT-K yöntemleri kullanılarak gerçekleştirilen arama geçmişlerine ait sonuçları gösterilmektedir.



Şekil 4.1 AnkastrKiriş-1 örneği için arama geçmişi

Şekil 4.1’de AnkastrKiriş-1 yapısına ilişkin olarak FEM ve geliştirilen Heuristic arama yöntemlerin kullanıldığı araştırmada minimum ve ortalama ağırlık değerlerinin arama geçmişi incelenmektedir. Analiz, FEM yönteminin iterasyon sayısı arttıkça yakınsamanın yavaşladığını; fakat 3500’üncü iterasyondan sonra sabit kalmakta olduğunu göstermektedir. Bu durum, FEM’in diğer yöntemlere nazaran daha hızlı yakınsama sağladığını ve daha çabuk sonuçlar elde edildiğini ortaya koymaktadır. Diğer yandan, YT-1 yönteminin minimum değerler açısından zaman içinde önemli bir değişiklik göstermemekte olmasına karşın, diğer yöntemlere göre daha etkili bir yakınsama sergilediği belirlenmektedir.

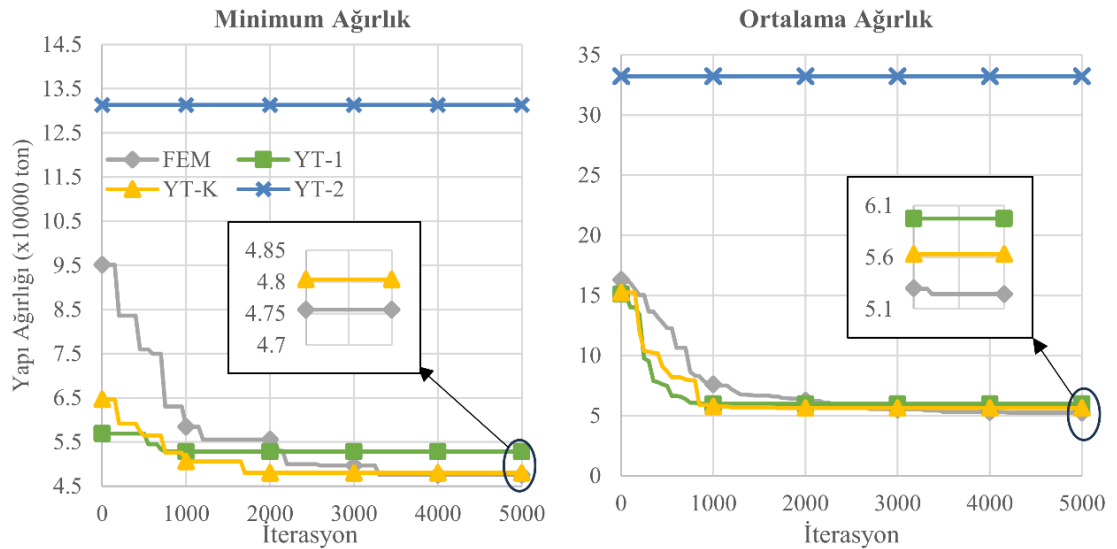
Araştırma bulguları, YT-K yönteminin arama performansının, diğer geliştirilen yöntemlere kıyasla ideal bir yakınsama grafiği sergilemekte olduğunu; ancak en iyi çözümler açısından yeni bir gelişme sağlamadığını, başlangıç çözümleriyle optimizasyon döngüsünü tamamladığını göstermektedir. Minimum ağırlık değerleri üzerinden yapılan değerlendirme, YT-1 ve YT-K yöntemlerinin FEM çözümlerine yakınsamakta olduğunu; ancak AnkastrKiriş-1 yapısında bu yöntemlerin FEM ile kıyaslandığında belirgin bir başarı sergilemediklerini ortaya koymaktadır. Ayrıca, ortalama değerler açısından FEM’in iyi bir yakınsama gelişimi sergilediği; diğer yöntemlerin ise çözüm geliştirme kapasitelerinin sınırlı kaldığı anlaşılmaktadır. Bu tespitler, incelenen yöntemlerin performansının kapsamlı bir şekilde değerlendirilmesine imkân tanımakta ve AnkastrKiriş-1 yapısının optimizasyonunda FEM yönteminin etkinliğini vurgulamaktadır.



Şekil 4.2 AnkastreKiriş-2 örneği için arama geçmişi

AnkastreKiriş-2 yapısının optimizasyonu üzerine yapılan analizde, minimum ağırlık değerleri baz alındığında, YT-1 ve YT-K yöntemlerinin FEM ile elde edilen çözümlere yakınsamakta olduğu görülmektedir. Özellikle minimum değerlerin grafik analizlerinde, YT-K ve YT-1 yöntemlerinin yakınsama performansı açısından benzer sonuçlar verdiği; ancak YT-K'nin YT-1'e göre kısmen daha iyi performans sergilediği belirlenmektedir. FEM yönteminin başlangıçtaki çözümünün iyi olmasına bağlı olarak, iterasyonların sonunda daha avantajlı bir değere ulaştığı gözlemlenmektedir.

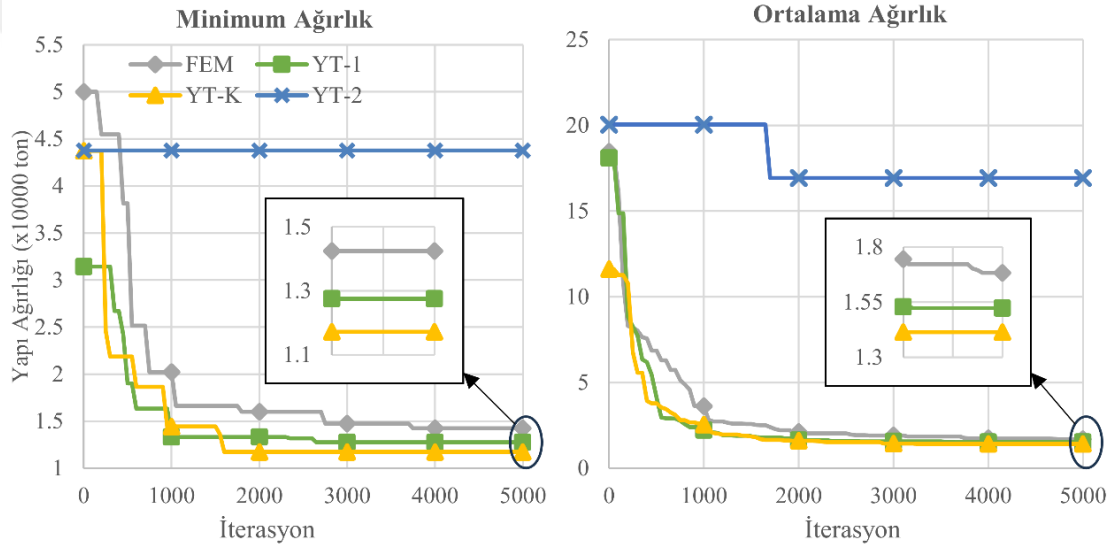
Ortalama değerlerin incelenmesiyle, FEM'in diğer yöntemlere kıyasla daha üstün bir yakınsama grafiği sergilediği anlaşılmaktadır. YT-K ve YT-1 yöntemleri, yeterli bir yakınsama performansı göstermekle birlikte, YT-2 yönteminin yakınsama performansında bazı problemler yaşandığı tespit edilmektedir.



Şekil 4.3 Dar kule örneği için arama geçmişi

Şekil 4.3'te YT-K yönteminin diğer yaklaşımlara kıyasla daha üstün sonuçlar sergilediği belirlenmektedir. HV1'in başlangıç çözümünün YT-K ve YT-2'ye göre daha iyi olmasına rağmen, optimizasyon sürecinde beklenen performans gelişimini sağlayamadığı ve yakınsama performansı açısından tatmin edici olmadığı gözlemlenmektedir. FEM'in ise, başlangıç çözümü olarak YT-K ve YT-1'den daha yüksek bir noktadan başlamasına rağmen, en iyi çözümünü sürekli olarak geliştirdiği ve bu performansını özellikle 3000'li iterasyonlardan sonra büyük ölçüde tamamladığı anlaşılmaktadır.

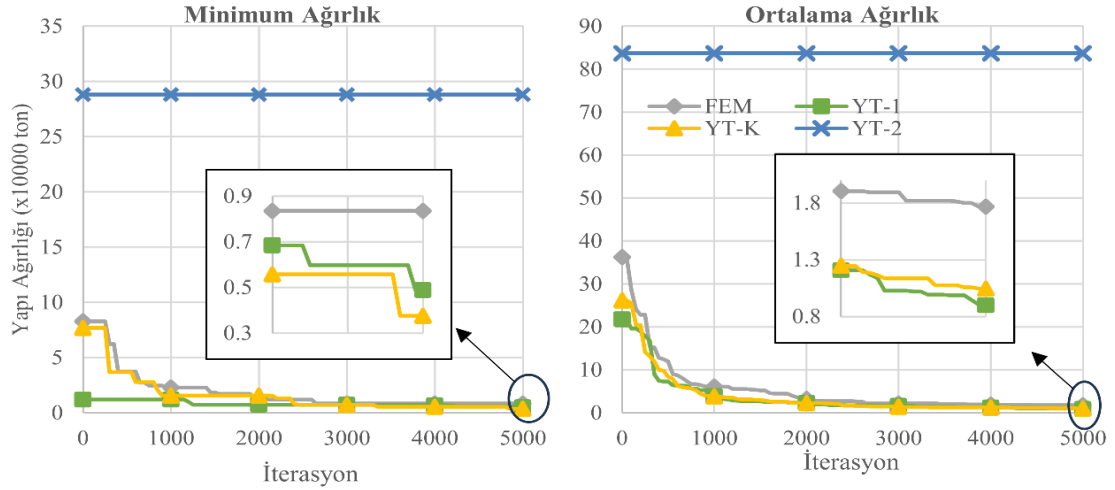
Ortalama değerlerin karşılaştırılmasında, YT-1, YT-2 ve YT-K'nin yakınsama performanslarının birbirine oldukça yakın olduğu tespit edilmektedir. Ancak, YT-2 yönteminin optimizasyon süreci boyunca çözümünde herhangi bir ilerleme kaydetmediği belirlenmektedir. Bu durum, kullanılan yöntemlerin performansının detaylı bir şekilde analiz edilmesi ve özellikle YT-2'nin optimizasyon kapasitesinin iyileştirilmesi gerektiğini göstermektedir. Bu tür analizler, optimizasyon stratejilerinin etkinliğini artırmak ve daha verimli çözümler üretmek için önemli bir katkı sağlamaktadır.



Şekil 4.4 Geniş kule örneği için arama geçmişi

Geniş kule örneğinin minimum değerler grafiğine yönelik analiz, YT-K yönteminin yakınsama performansı açısından en iyi değerleri sunduğunu göstermektedir. YT-1 ve FEM tarafından elde edilen sonuçlar da bu performansa oldukça yakın bir düzeyde gerçekleşmektedir. Bu durum, YT-K'nin bu özel örnekte etkili bir optimizasyon stratejisi sunduğunu ve YT-1 ile FEM'in de rekabetçi sonuçlar elde ettiğini göstermektedir.

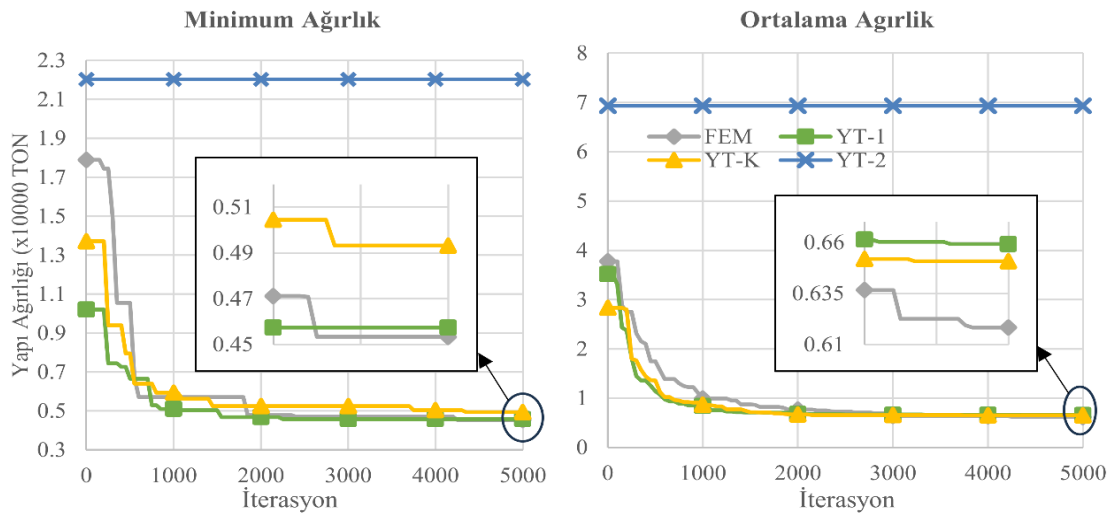
Ortalama değerlerin karşılaştırılması, elde edilen sonuçların genel olarak birbirine yakın olduğunu ortaya koymaktadır. Ancak, YT-2 yöntemi, bu örnekte de yakınsama performansı bakımından diğer yöntemlere kıyasla olumsuz bir şekilde ayrılmaktadır. Bu, YT-2'nin bu özel durumda diğer yöntemlerle karşılaştırıldığında daha az etkili olduğunu ve optimizasyon stratejilerinde iyileştirmelere ihtiyaç duyulduğunu göstermektedir. Bu tespitler, farklı optimizasyon yöntemlerinin performansını anlamak ve potansiyel iyileştirme alanlarını belirlemek açısından önem taşımaktadır.



Şekil 4.5 Klasik Köprü-1 örneği için arama geçmişi

Klasik Köprü-1 örneğinin minimum değerler grafiği incelendiğinde, YT-1 yönteminin ikinci en iyi çözümü elde etmiş olmasına rağmen, daha düşük bir başlangıç noktasından yola çıkması nedeniyle YT-K ve FEM'e kıyasla belirgin bir performans üstünlüğü sergileyemediği belirlenmektedir. Bu durum, optimizasyon sürecinde başlangıç değerlerinin sonuçlar üzerindeki etkisinin önemini vurgulamaktadır.

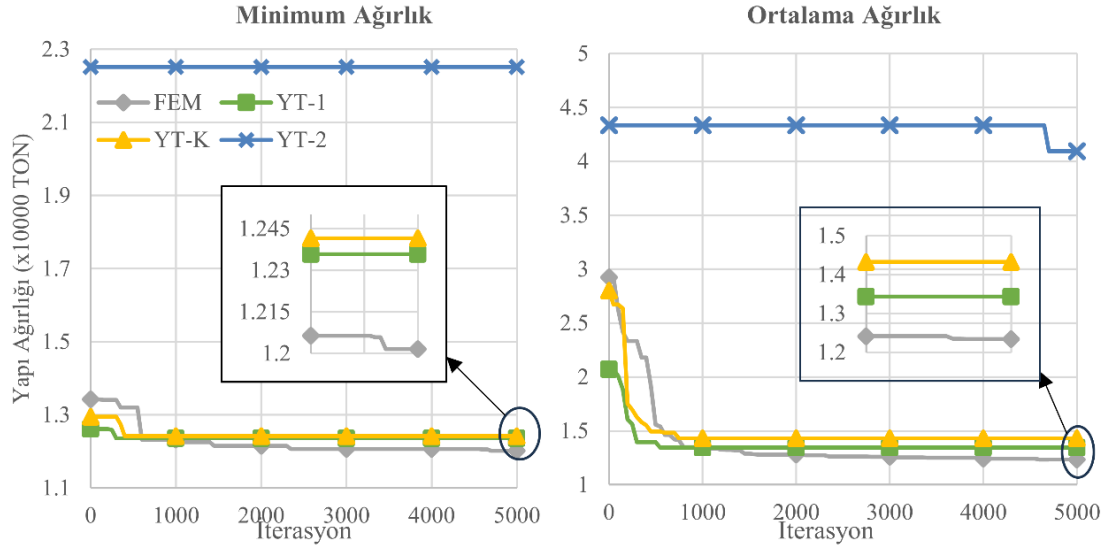
Ortalama değerler grafiğine bakıldığında, YT-K ve YT-1 yöntemleri arasında belirgin bir üstünlük olmadığı, ancak çok küçük bir farkla YT-1'in daha iyi bir performans sergilediği gözlemlenmektedir. Buna karşın, FEM yöntemi de bu değerlere oldukça yakın bir performans göstermektedir, bu da farklı yöntemlerin ortalama değerler açısından birbirlerine yakın sonuçlar verebildiğini göstermektedir. YT-2 yönteminin hem minimum hem de ortalama değerler grafiğinde benzer bir eğilim sergilemesine rağmen, optimizasyon sürecinde sonuçları anlamlı bir şekilde geliştiremediği anlaşılmaktadır. Bu, YT-2 yönteminin bu özel durumda diğer yöntemlerle kıyaslandığında daha az etkili olduğunu ve potansiyel iyileştirme alanlarına ihtiyaç duyulduğunu göstermektedir. Bu analizler, yöntemlerin performansını derinlemesine anlamak ve optimizasyon stratejilerini daha da geliştirmek için değerli bilgiler sunmaktadır.



Şekil 4.6 Klasik Köprü-2 örneği için arama geçmişi

KlasikKöprü-2 örneğinde yapılan incelemede, minimum değerler üzerinden FEM yönteminin, daha yüksek bir başlangıç noktasından başlamasına rağmen, en başarılı sonucu elde ettiği belirlenmektedir. FEM ve YT-1 arasında belirgin bir üstünlük olmamakla birlikte, FEM'in çok küçük bir farkla daha iyi bir performans sergilediği görülmektedir. YT-K yöntemi de bu iki yöntemin elde ettiği değerlere oldukça yakın bir performans göstermektedir. Bu durum, özellikle FEM ve YT-1 yöntemlerinin KlasikKöprü-2 örneğinde benzer düzeyde etkili olduğunu göstermektedir.

Ortalama değerler açısından yapılan analizde, FEM yönteminin yakınsama performansının en iyi olduğu tespit edilmiştir. Ancak, YT-K ve YT-1 yöntemleri, FEM'e çok az bir farkla geride kalmakta ve benzer bir başarı sergilemektedirler. Diğer yandan, YT-2 yöntemi hem minimum hem de ortalama değerler grafiğinde benzer bir eğilim sergilese de optimizasyon sürecinde sonuçları anlamlı bir şekilde geliştiremediği anlaşılmaktadır. Bu bulgular, KlasikKöprü-2 örneğinde kullanılan optimizasyon yöntemlerinin performanslarını değerlendirme ve iyileştirme potansiyellerini ortaya koymakta, özellikle YT-2 yönteminin geliştirilmesine yönelik ihtiyaçları vurgulamaktadır.



Şekil 4.7 Piramit örneği için arama geçmişi

Minimum değerlerin incelenmesi, YT-K, YT-1 ve FEM yöntemlerinin ideal bir yakınsama performansı sergileyemediklerini göstermektedir. Başlangıç çözümleri ile en iyi çözümler arasında belirgin bir fark oluşmadığı belirlenmektedir. Buna rağmen, FEM yöntemi diğerlerine kıyasla en iyi değere ulaşmıştır. Bu, FEM yönteminin, optimizasyon sürecinde diğer yöntemlere göre daha etkili olduğunu göstermektedir, ancak genel olarak yöntemler arasındaki farkların küçük olduğu anlaşılmaktadır.

Ortalama değerlerin analizi, YT-K, YT-1 ve FEM'in ideal bir performans grafiği çizdiklerini ortaya koymaktadır. En iyi sonuçların birbirlerine oldukça yakın olduğu görülse de FEM yöntemi az bir farkla en iyi ortalama değeri bulmuştur. Bu, FEM yönteminin ortalama değerler açısından da diğer yöntemlere göre hafif bir üstünlüğe sahip olduğunu göstermektedir. Bu bulgular, söz konusu yöntemlerin performanslarının

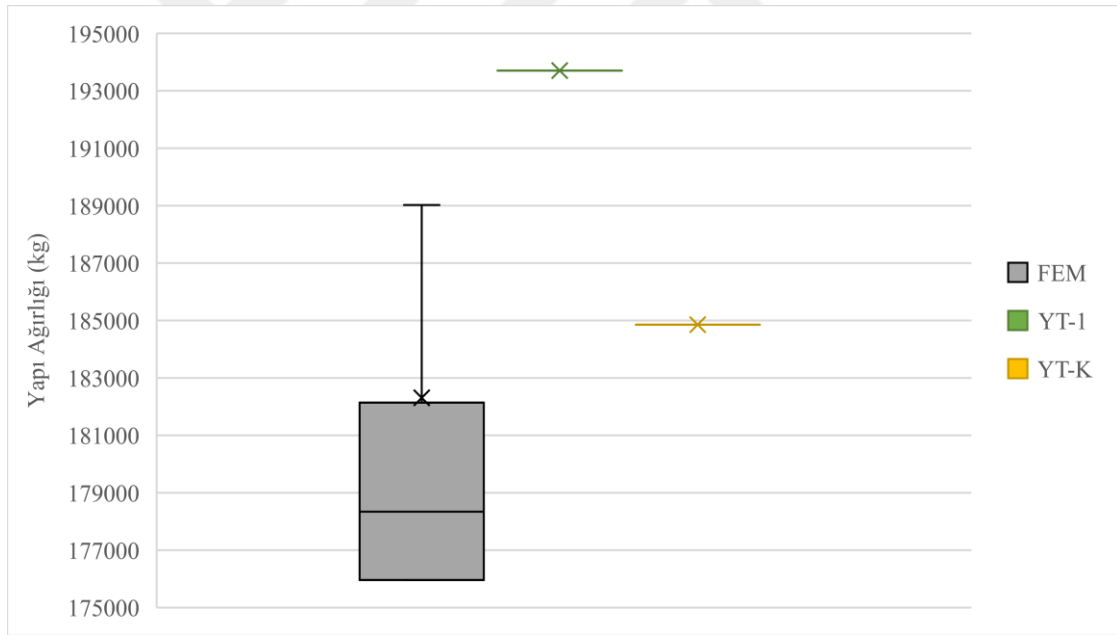
yakın olduğunu ve optimizasyon stratejilerinin daha da geliştirilmesi için potansiyel alanlar sunabileceğini göstermektedir.

4.2.1. BTO arama yöntemlerinin istatistiksel performans açısından karşılaştırılması

Bu bölümde her bir arama yöntemi için teknik veriler ve Kutu Çizgi (Box Plot) grafikleri çıkartılmış ve değerlendirilmeleri yapılmıştır.

Çizelge 4.2 AnkastreKiriş-1 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)

	FEM	YT-1	YT-K	YT-2
MIN	175952,24	193708,19	184852,64	225393,16
Q1	175952,24	193708,19	184852,64	225393,16
MEDYAN	178339,23	193708,19	184852,64	225393,16
ORTALAMA	182307,14	193708,19	184852,64	225393,16
Q3	178339,23	193708,19	184852,64	225393,16
STD	11875,95	0,00	0,00	0,00
MAKS	235245,64	193708,19	184852,64	225393,16



Şekil 4.8 AnkastreKiriş-1 örneği için arama geçmiş

FEM arama yöntemi, AK-1 yapı problemi bağlamında diğer yöntemlere kıyasla daha geniş bir değer aralığı sergilemektedir. Özellikle minimum ve maksimum değerler arasındaki fark, FEM arama yönteminin problem uzayında daha geniş bir araştırma yapabildiğini göstermektedir. Bununla birlikte, ortalama değerinin diğer yöntemlerin minimum değerlerine yakın olması, bu yöntemin uygun koşullar altında etkili çözümler üretebileceğini belirtmektedir.

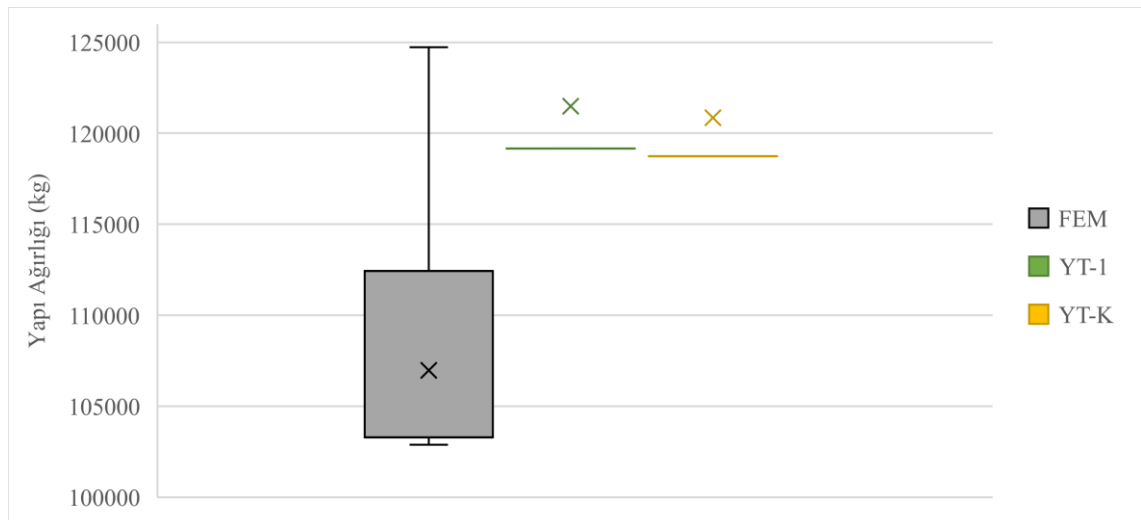
YT-1 arama yönteminin sonuçları, standart sapmanın sıfır olmasıyla dikkat çekicidir. Bu durum, YT-1 arama yönteminin belirli bir optimizasyon problemi için tutarlı sonuçlar ürettiğini göstermektedir. Ancak, elde edilen değerlerin diğer yöntemlerin ürettiği değerlere göre daha yüksek olduğu gözlemlenmiştir, bu da YT-1'in bu problem seti için optimal olmayan çözümlere yönelebileceğini göstermektedir.

YT-K arama yönteminin sonuçları da benzer şekilde, standart sapmanın sıfır olmasıyla karakterize edilmektedir. Bu durum, YT-K arama yönteminin de belirli koşullar altında tutarlı sonuçlar üretebileceğini belirtmektedir. Ancak, YT-K'nin minimum, medyan ve ortalama değerleri, YT-1'e göre daha düşük olmakla birlikte, hala FEM arama yönteminin ortalama değerinden yüksek gerçekleşmiştir.

YT-2 arama yöntemi, incelenen tüm arama yöntemleri içinde en yüksek minimum, medyan ve ortalama değerlere sahiptir. Bu durum, YT-2 arama yönteminin, diğer yöntemlere kıyasla bu özel optimizasyon problemi için daha az etkili olduğunu göstermektedir. Standart sapmanın sıfır olması, YT-2'nin tutarlı sonuçlar ürettiğini; ancak bu sonuçların optimizasyon problemi bağlamında ideal olmadığını belirtmektedir.

Çizelge 4.3 AnkastreKiriş-2 örneği ile ilgili arama yöntemleri sonuçları (Birim:kg)

	FEM	YT-1	YT-K	YT-2
MIN	102879,33	119163,06	118744,64	202837,02
Q1	103289,31	119163,06	118744,64	202837,02
MEDYAN	103289,31	119163,06	118744,64	202837,02
ORTALAMA	106963,54	121496,30	120859,17	202837,02
Q3	112438,02	119163,06	118744,64	202837,02
STD	5893,37	9404,78	5151,06	0,00
MAKS	124726,80	161844,09	137953,66	202837,02



Şekil 4.9 AnkastreKiriş-2 örneği için arama geçmişi

AK-2 için sunulan tabloda belirtilen dört farklı arama yöntemi üzerinden bir değerlendirme yapıldığında, FEM arama yönteminin minimum, birinci çeyrek (Q1),

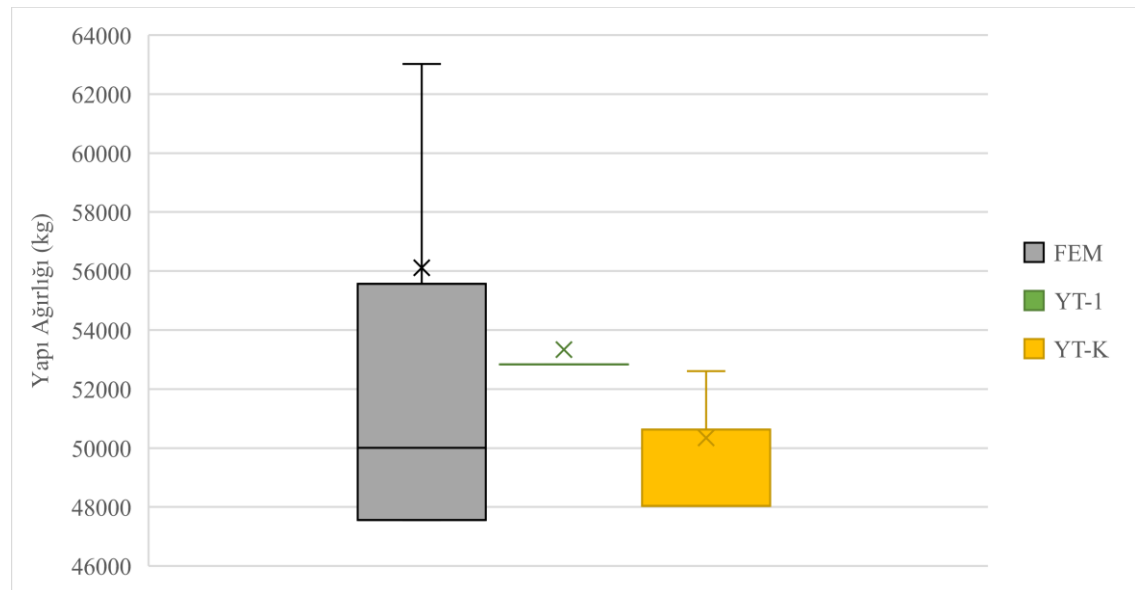
medyan, ortalama ve üçüncü çeyrek (Q3) değerlerinin diğer yöntemlere kıyasla daha düşük olduğu gözlemlenmiştir. Bu durum, FEM arama yönteminin belirli bir optimizasyon problemi için daha iyi bir başlangıç noktası sunabileceğini ve potansiyel olarak daha az iterasyon ile optimum çözüme ulaşabileceğini belirtmektedir.

YT-1 ve YT-K arama yöntemlerine bakıldığında, her iki yöntemin de benzer minimum, Q1, medyan ve ortalama değerlere sahip olduğu görülmektedir. Bu benzerlik, her iki yöntemin de söz konusu optimizasyon probleminde benzer bir performans gösterdiğini ve aralarında önemli bir fark olmadığını göstermektedir. Ancak YT-K arama yönteminin maksimum değeri, YT-1'e göre daha yüksek olmuştur, bu da YT-K'nin bazı durumlarda daha yüksek sapmalara maruz kalabileceğini göstermektedir.

Diğer taraftan, YT-2 arama yöntemi, minimum, Q1, medyan, ortalama ve Q3 değerleri bakımından diğer yöntemlerden önemli ölçüde farklılık göstermektedir. YT-2'nin bu değerlerinin diğerlerine kıyasla önemli ölçüde daha yüksek olması, bu yöntemin belirli koşullar altında diğer yöntemlere göre daha az etkili olabileceğini göstermektedir. Özellikle, standart sapma değerinin sıfır olması, YT-2 arama yönteminin sonuçlarının tutarlı olduğunu ancak optimizasyon probleminin minimizasyonu açısından ideal olmayan bir çözüm sunduğunu belirtmektedir.

Çizelge 4.4 Dar Kule örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)

	FEM	YT-1	YT-K	YT-2
MIN	47560,89	52833,94	48037,57	131334,66
Q1	47560,89	52833,94	48037,57	131334,66
MEDYAN	50008,78	52833,94	48037,57	131334,66
ORTALAMA	56106,88	53332,11	50342,11	131334,66
Q3	55567,65	52833,94	50623,18	131334,66
STD	12593,94	1287,73	4281,25	0,00
MAKS	95167,07	56904,54	64707,54	131334,66



Şekil 4.10 Dar kule örneği için arama geçmişi

FEM optimizasyon arama yönteminin sonuçları incelendiğinde, minimum ve maksimum değerler arasındaki geniş aralık dikkat çekmektedir. Bu geniş aralık, FEM yönteminin problem uzayında geniş bir keşif gerçekleştirdiğini göstermektedir. Ayrıca, ortalama değer diğer yöntemlere kıyasla daha düşük olup, bu da FEM yönteminin genel olarak daha iyi çözümler üretebildiğini belirtmektedir. Ancak, standart sapmanın yüksek olması, elde edilen sonuçların önemli bir değişkenlik gösterdiğini göstermektedir. Bu da FEM yönteminin problem uzayını kapsamlı bir şekilde araştırdığını, fakat bu süreçte tutarlılık açısından bazı sınırlılıklarla karşılaştığını belirtmektedir.

YT-1 optimizasyon arama yöntemiyle elde edilen sonuçlar ise farklı bir profile sahiptir. Bu yöntemin minimum ve maksimum değerler açısından daha dar bir aralıkta sonuçlar ürettiği görülmektedir. Ortalama değerin, FEM yöntemine kıyasla biraz daha yüksek olması, YT-1 yönteminin bu problem bağlamında biraz daha yüksek amaç fonksiyonu değerleri ürettiğini göstermektedir. Ayrıca, standart sapmanın oldukça düşük olması, YT-1 yönteminin sonuçlarının oldukça tutarlı olduğunu göstermektedir. Bu da optimizasyon problemi için güvenilir çözümler sunabileceğini işaret etmektedir.

Öte yandan, YT-K optimizasyon arama yönteminin sonuçları incelendiğinde, bu yöntemin de FEM ve YT-1'e benzer bir değer aralığında sonuçlar ürettiği görülmektedir. Ortalama değer, FEM ve YT-1 yöntemlerinin ortalamalarına yakın olup, YT-K yönteminin ortalama bir performans sergilediğini belirtmektedir. Ancak, standart sapmanın YT-1'den daha yüksek olması, YT-K yönteminin sonuçlarının daha fazla değişkenlik gösterdiğini göstermektedir. Bu da YT-K yönteminin problem uzayını geniş bir yelpazede keşfettiğini ancak sonuçların tutarlılığı açısından bazı sınırlılıklarla karşılaştığını belirtmektedir.

Son olarak, YT-2 optimizasyon arama yönteminin sonuçları, diğer üç yönteme kıyasla önemli ölçüde farklılık göstermektedir. YT-2 yöntemi, tüm istatistiksel ölçütlerde sabit bir değer sergilemekte olup, bu da YT-2 yönteminin bu optimizasyon problemi için tek bir çözüm ürettiğini göstermektedir. Bu sabit değer, diğer yöntemlerin minimum ve maksimum değerlerine kıyasla önemli ölçüde daha yüksek olup, YT-2 yönteminin bu problemi minimize etme konusunda diğer yöntemlere göre daha az etkili olduğunu belirtmektedir. Bu sebepten dolayı YT-2 yöntemine ait gösterim Şekil 4.11 içerisinde gösterilememiştir. Standart sapmanın sıfır olması, YT-2 yönteminin son derece tutarlı ancak optimizasyon açısından alt optimal sonuçlar ürettiğini göstermektedir.

Çizelge 4.5 Geniş Kule örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)

	FEM	YT-1	YT-K	YT-2
MIN	14244,58	12755,68	11713,63	43749,00
Q1	14244,58	12755,68	11713,63	43749,00
MEDYAN	15993,50	13161,79	11713,63	43749,00
ORTALAMA	19248,23	15017,44	14849,76	43749,00
Q3	16628,46	13299,43	14421,84	43749,00
STD	9717,58	5134,72	7306,78	0,00
MAKS	49993,12	31430,63	43710,19	43749,00



Şekil 4.11 Geniş kule örneği için arama geçmişi

FEM optimizasyon arama yönteminin analizine odaklandığımızda, bu yöntemin elde ettiği minimum ve maksimum değerlerin geniş bir aralıkta olduğu gözlemlenmiştir. Ortalama değer diğer yöntemlere göre daha yüksek olması, FEM yönteminin bu problem için elde ettiği çözümlerin genel olarak daha yüksek amaç fonksiyonu değerlerine sahip olduğunu göstermektedir. Standart sapmanın yüksek olması ise FEM yönteminin sonuçlarının önemli ölçüde değişkenlik gösterdiğini belirtmektedir. Bu durum, yöntemin problem uzayını geniş bir yelpazede araştırdığını, ancak tutarlılık açısından sınırlılıklar yaşadığını göstermektedir.

Diğer yandan, YT-1 optimizasyon arama yönteminin değerlendirilmesi, bu yöntemin daha dar bir değer aralığında sonuçlar ürettiğini ortaya koymaktadır. Özellikle, YT-1 yönteminin ortalama değeri FEM'e kıyasla daha düşük olup, bu da yöntemin daha etkili çözümler bulma eğiliminde olduğunu göstermektedir. Standart sapmanın daha düşük olması, YT-1 yönteminin sonuçlarının daha tutarlı olduğunu ve optimizasyon problemi için daha güvenilir çözümler sunduğunu işaret etmektedir.

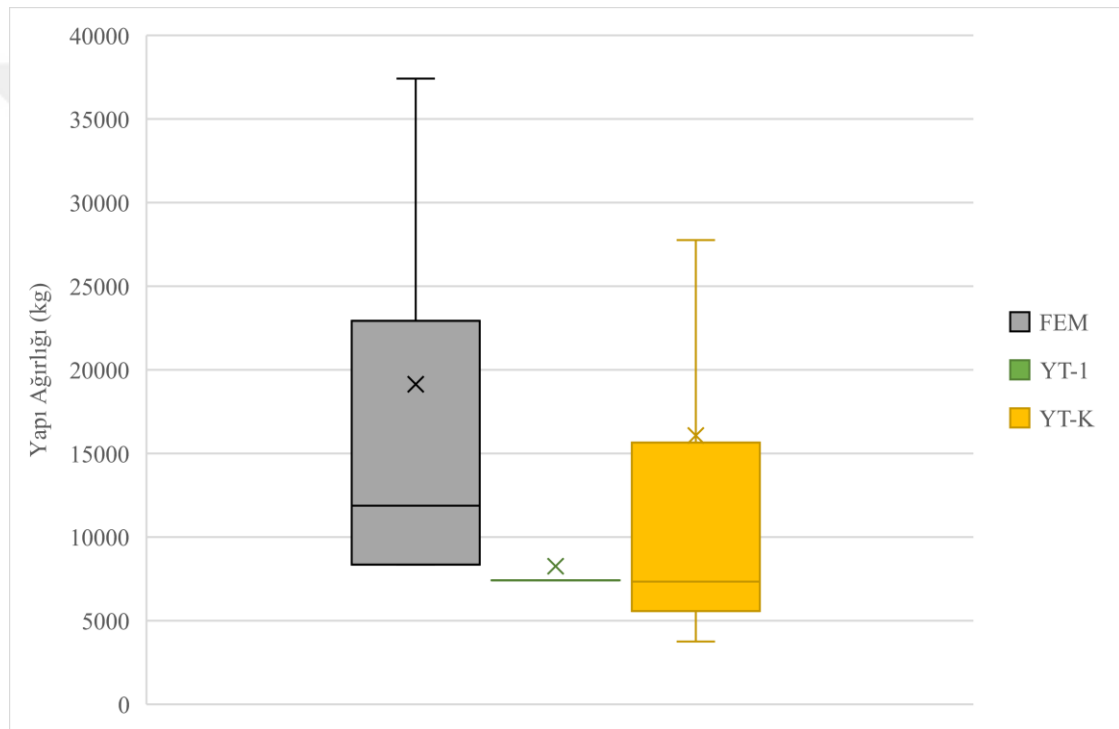
YT-K optimizasyon arama yöntemi incelendiğinde, minimum ve maksimum değerler arasındaki geniş aralık, yöntemin problem uzayını geniş bir yelpazede keşfettiğini göstermektedir. Ortalama değer, FEM yöntemine kıyasla daha düşük olmakla birlikte, YT-1 yöntemine göre daha yüksek bulunmuştur. Bu, YT-K yönteminin incelenen problem bağlamında ortalama bir performans sergilediğini göstermektedir. Standart sapmanın YT-1'e kıyasla daha yüksek olması, YT-K yönteminin sonuçlarının daha fazla değişkenlik gösterdiğini belirtmektedir.

Son olarak, YT-2 optimizasyon arama yönteminin sonuçları, tüm istatistiksel ölçütlerde sabit değerler sergilemektedir. Bu durum, YT-2 yönteminin bu optimizasyon problemi için tek bir çözüm ürettiğini ve bu çözümün diğer yöntemlere kıyasla daha yüksek bir amaç fonksiyonu değerine sahip olduğunu göstermektedir. Standart sapmanın sıfır

olması, YT-2 yönteminin son derece tutarlı ancak optimizasyon açısından alt optimal sonuçlar ürettiğini belirtmektedir.

Çizelge 4.6 Klasik Köprü-1 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)

	FEM	YT-1	YT-K	YT-2
MIN	8352,87	4888,33	3757,45	288118,48
Q1	8352,87	7417,36	5577,17	288118,48
MEDYAN	11873,45	7417,36	7329,91	288118,48
ORTALAMA	19147,43	8262,82	16074,60	288118,48
Q3	22924,65	7417,36	15645,88	288118,48
STD	19095,37	2231,08	17554,38	0,00
MAKS	82825,22	12071,78	76834,16	288118,48



Şekil 4.12 Klasik Köprü-1 örneği için arama geçmişi

FEM yöntemi, geniş bir değer aralığı ve yüksek standart sapma ile karakterize edilmiştir, bu da önemli ölçüde heterojen sonuçların elde edildiğine işaret etmektedir. Bu heterojenlik, FEM yönteminin bu optimizasyon problemi bağlamında çok çeşitli çözümler üretebilme kapasitesine işaret eder. Ancak, bu aynı zamanda sonuçların tutarlılığının ve öngörülebilirliğinin sınırlı olduğunu da gösterir.

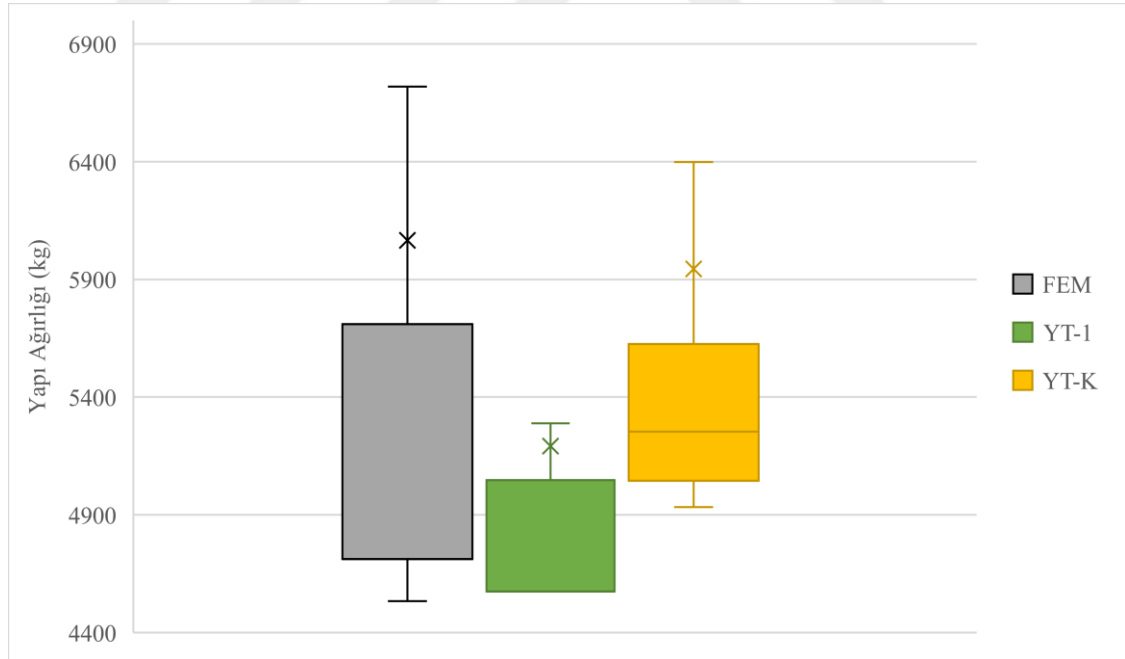
YT-1 arama yöntemi, özellikle minimum ve maksimum değerler bakımından daha dar bir aralıkta sonuçlar üretmiştir. Bu yöntemin ortalama değeri, FEM arama yöntemine kıyasla önemli ölçüde daha düşüktür; bu da YT-1'in genel olarak daha etkili çözümler bulduğunu gösterir. Standart sapmanın nispeten düşük olması, YT-1 arama yönteminin bu optimizasyon problemi için tutarlı sonuçlar ürettiğini göstermektedir.

YT-K arama yöntemi, minimum ve maksimum değerler açısından büyük bir aralık göstermektedir, bu da bu yöntemin problem uzayında geniş bir keşif gerçekleştirdiğini ancak bazı durumlarda yüksek değerli çözümlere de yol açabileceğini gösterir. Ortalama değeri, YT-1 arama yönteminin ortalamasından daha yüksek olmasına rağmen, FEM arama yönteminin ortalamasından daha düşüktür. Bu, YT-K arama yönteminin, incelenen problem için orta derecede başarımlı olduğunu gösterir.

YT-2 arama yöntemi, tüm istatistiksel ölçütlerde sabit bir değer sergilemektedir. Bu durum, YT-2 arama yönteminin optimizasyon sürecinde tek bir çözüme ulaştığını ve bu çözümün problemi minimize etme konusunda diğer yöntemlere kıyasla daha az etkili olduğunu gösterir. Standart sapmanın sıfır olması, YT-2'nin son derece tutarlı ancak bu bağlamda optimal altı sonuçlar ürettiğini göstermektedir.

Çizelge 4.7 Klasik Köprü-2 örneği ile ilgili arama yöntemleri sonuçları (Birim: kg)

	FEM	YT-1	YT-K	YT-2
MIN	4533,56	4574,20	4932,48	22024,68
Q1	4711,82	4574,20	5045,06	22024,68
MEDYAN	4711,82	4574,20	5253,71	22024,68
ORTALAMA	6065,83	5192,46	5944,39	22024,68
Q3	5710,02	5048,03	5625,02	22024,68
STD	3324,43	1360,71	2014,04	0,00
MAKS	17892,55	1020,29	13718,16	22024,68



Şekil 4.13 Klasik Köprü-2 örneği için arama geçmişi

FEM optimizasyon arama yönteminin sonuçları, minimum ve maksimum değerler arasında geniş bir aralık göstermektedir. Bu durum, FEM yönteminin problem uzayında geniş bir keşif gerçekleştirdiğini ancak bu sürecin bazı durumlarda yüksek değerlerle sonuçlandığını göstermektedir. Ortalama değer diğer yöntemlere kıyasla daha yüksek

olması, FEM yönteminin genel başarımının bu problem bağlamında optimal olmadığını işaret etmektedir. Standart sapmanın diğer yöntemlere göre daha yüksek olması ise sonuçların daha fazla varyans gösterdiğini ve dolayısıyla tutarlılığın daha düşük olduğunu göstermektedir.

YT-1 optimizasyon arama yönteminin analizi, bu yöntemin minimum ve maksimum değerler açısından daha dar bir aralıkta sonuçlar ürettiğini göstermektedir. Ortalama değer, FEM yöntemine kıyasla daha düşük olup, YT-1 yönteminin genel olarak daha etkili çözümler bulduğunu göstermektedir. Standart sapmanın nispeten düşük olması, YT-1 yönteminin bu optimizasyon problemi için tutarlı sonuçlar ürettiğine işaret etmektedir.

YT-K optimizasyon arama yöntemi ile ilgili olarak, minimum ve maksimum değerler arasındaki aralığın, FEM ve YT-1 yöntemleriyle kıyaslandığında geniş olduğu görülmektedir. Bu, YT-K yönteminin problem uzayında geniş bir aralıkta arama yaptığını ancak bu sürecin bazı durumlarda yüksek değerlerle sonuçlanabileceğini göstermektedir. Ortalama değer FEM'e yakın fakat YT-1'den yüksek olması, YT-K yönteminin bu problem için orta derecede başarımlı olduğunu göstermektedir.

YT-2 optimizasyon arama yönteminin sonuçları, tüm istatistiksel ölçütlerde sabit bir değer sergilemektedir. Bu durum, YT-2 yönteminin optimizasyon sürecinde tek bir çözüme ulaştığını ve bu çözümün diğer yöntemlerle kıyaslandığında problemi minimize etme konusunda etkisiz olduğunu göstermektedir. Standart sapmanın sıfır olması, YT-2'nin son derece tutarlı olduğunu ancak optimum çözüme erişim yeteneği açısından diğer yöntemlerden geride kaldığını göstermektedir.

4.2.2. BTO arama yöntemlerinin çözüm hızı açısından karşılaştırılması

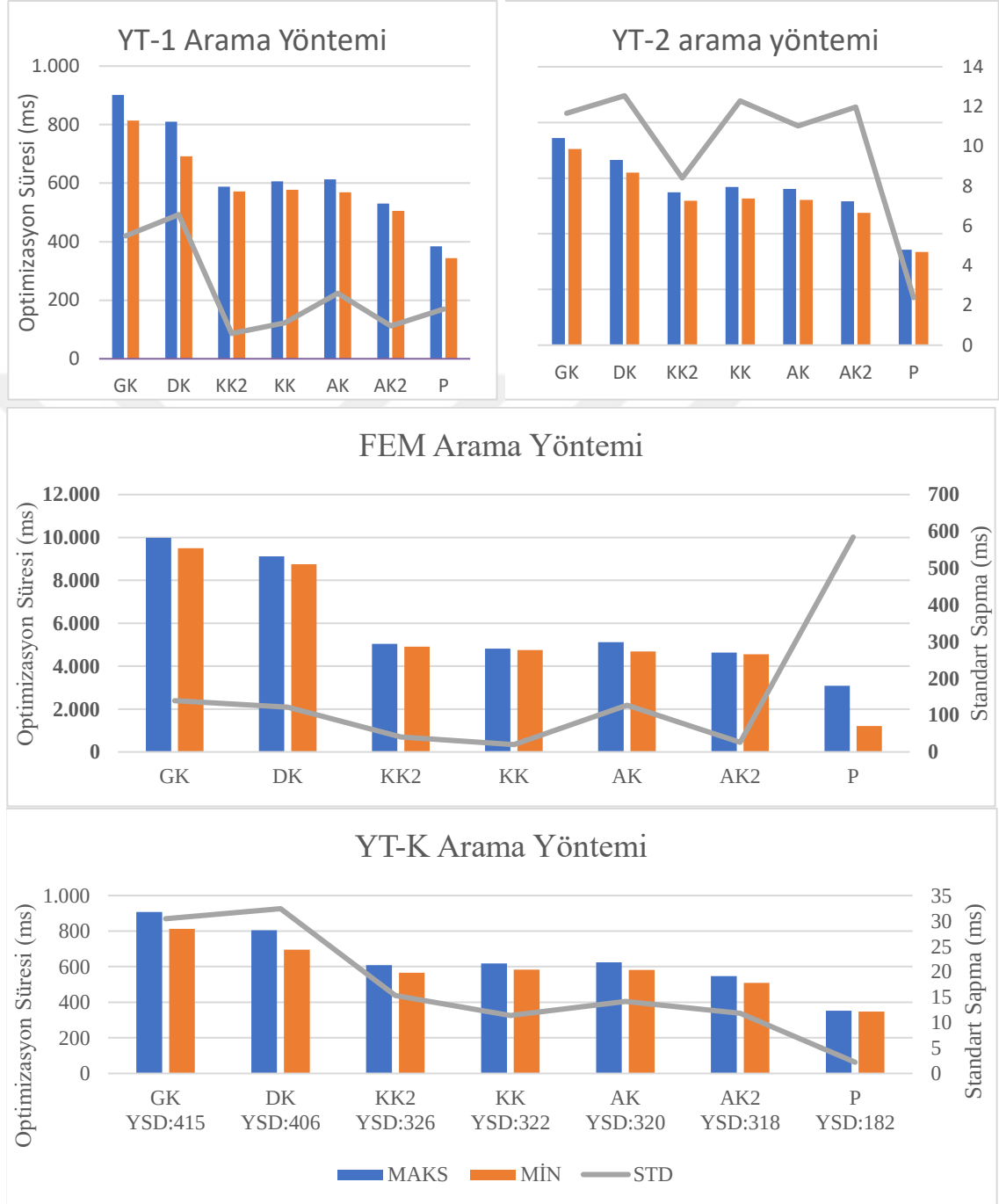
Önceki bölümlerde, yedi farklı yapı tipi üzerinde test edilen gelişmiş Yapay Sinir Ağları arama yöntemlerinin yapısal performans sonuçları incelenmiştir. Bu bölümde ise yöntemlere ait süresel karşılaştırmalar yapılarak sonuçlar değerlendirilmiştir.

Bu deneylerde kullanılan deney bilgisayarına ait özellikler Çizelge 4.8'de sunulmaktadır.

Çizelge 4.8 Deney bilgisayar donanım özellikleri

Özellik		Değer
CPU	İsim :	Intel Core i7-12700H
	Çekirdek sayısı :	14
	Threads sayısı :	20
	İşlem hızı :	4,7 GHz
	Ön hafıza(cache) miktarı :	L1:80KB/L2:1,25MB/L3:24MB
RAM	İsim :	Crucial C16FR
	Miktar :	32 GB
	DRAM hızı :	1600MHz
GPU	İsim :	NVIDIA GeForce RTX 3060L
	İşlem Hızı :	1,4 GHz
	RAM Miktarı :	6 GB
	RAM Hızı :	1,75 GHz

Deneylerde kullanılan yazılım ortamı olarak Python 3.10.0(Python.Org 2024) ve YSA algoritmalarını eğitmek ve çalıştırmak için Tensorflow 2.10.0(Tensorflow.Org 2024) YSA kütüphanesi kullanılmıştır.

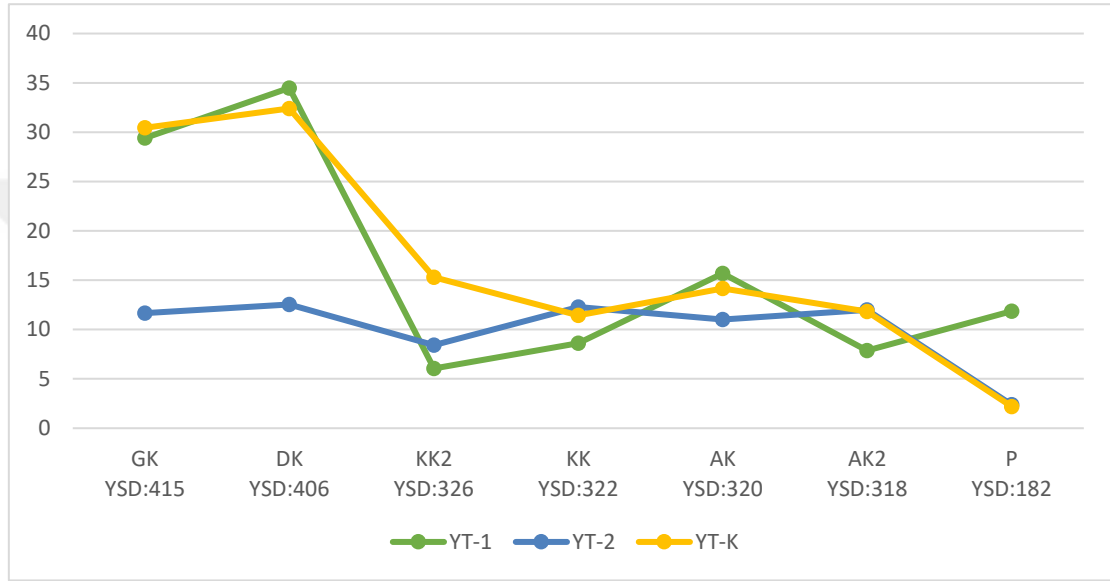


Şekil 4.14 YT-1,YT-2, YT-K ve FEM arama yöntemlerinin her bir yapı tipi için maksimum, minimum süre tüketim değerleri ve standart sapmaları

Şekil 4.14, her bir arama yöntemi ile 10 kere tekrarlanmış optimizasyon işlem süre sonuçlarını göstermektedir. Şekildeki optimizasyonlar, süresel farklılıkların daha iyi anlaşılabilmesi için YSD (Yanıt Süresi Dağılımı) değerlerine göre sıralanmıştır. Her bir

yapı tipi üzerindeki maksimum, minimum değerler ve sağ eksenden de STD (Standart Sapma) değerleri okunabilmektedir.

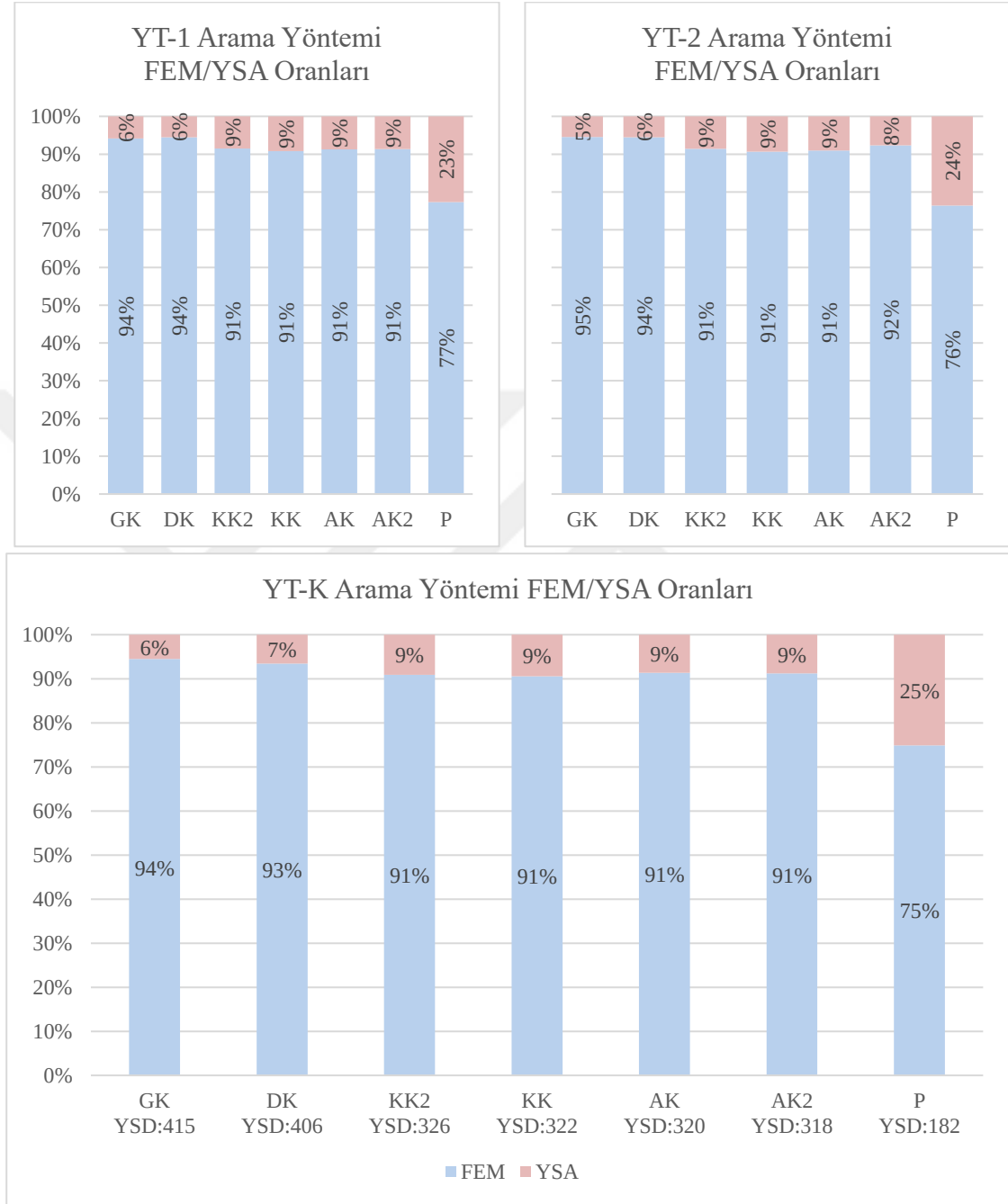
FEM arama yöntemi grafiği incelendiğinde, tüm yapılarda YSD değerine göre ters orantılı olarak süresel sonuçların gerçekleştiği görülmektedir. Tüm yapılar için STD değerleri eşit düzeylerde gerçekleşmiş gibi görünse de P yapı tipinde STD değeri ciddi farklılık göstermiştir. Şekil 4.14 incelendiğinde, FEM arama yönteminin süresel farklılıklarının, YSA ile geliştirilen diğer arama yöntemlerine göre çok yüksek olduğu görülmektedir. Bu nedenle değerlendirme yapılmadan önce, YSA arama yöntemlerine ait standart sapma grafikleri Şekil 4.15 içerisinde ayrıştırılarak gösterilmiştir.



Şekil 4.15 YT-1, YT-2 ve YT-K arama yöntemlerinin her bir yapı tipi için standart sapma sonuçları

Şekil 4.15 incelendiğinde YT-K ve YT-1 arama yöntemlerinin YT-2 'ye göre GK ve DK yapı tiplerinde farklılık gösterdiği belirgin bir biçimde görülebilmektedir. AK2 yapısı içinde bu durum tam tersi yönde gerçekleşmiş ve YT-2 arama yöntemi ile diğer iki arama yöntemine ait STD değerleri arasında ciddi bir biçimde YT-2 aleyhine açılma gerçekleştiği görülmektedir.

Arama yöntemlerinin süresel karşılaştırmaları birbirleri arasında olduğu gibi aynı zamanda her bir yöntemin kendi içerisindeki süre tüketimleri açısından ayrı ayrı değerlendirilmiştir.



Şekil 4.16 YT-1, YT-2 ve YT-K arama yöntemlerinin her bir yapı tipi içerisindeki FEM/YSA kullanım süre oranları

Geliştirilen YT-1, YT-2 ve YT-K arama yöntemlerine ait içsel işleyişler, bölüm 3.8.2’de detaylı olarak anlatılmaktadır. Bu içsel işleyişler esnasında, arama yöntemi kullanılan optimizasyon döngülerinin bazı evrelerinde klasik FEM yapı analizi kullanırken, bazı evrelerinde de YSA yardımıyla yapı analizleri gerçekleştirilmektedir. Bu sebeple, arama

yöntemlerinin tükettikleri sürelerin bir kısmı FEM işlemlerine, bir kısmı ise YSA işlemlerine harcanmaktadır. Şekil 3.1’de bu arama yöntemlerine ait FEM/YSA süre kullanım oranları verilmiştir.

Şekil 4.16 incelendiğinde, çözülen yapıya ait YSD oranının yüksek ya da düşük olmasına bağlı olarak, her bir yöntemin FEM/YSA oranlarında farklılaşmalar olduğu görülmektedir. Şekilden anlaşılabacağı üzere, bu farklılaşmanın en yüksek oranda gerçekleştiği arama yöntemi YT-K arama yöntemidir. Yöntemde özellikle P yapısı içerisinde, diğer yöntemlere oranla daha çok YSA kullanımı görülmekle beraber, AK2 ve DK yapılarında da benzer bir farklılık göze çarpmaktadır. Şekil 4.16 düşük YSD sayısına sahip yapıların, arama süreleri içerisinde YSA paylarının YSD sayısı ile ters orantılı bir biçimde arttığını göstermektedir. Bu durum, belirgin bir biçimde P yapısında görülmektedir.



5. SONUÇLAR

Bu çalışma, Sonlu Elemanlar Metoduna alternatif olarak Yapay Sinir Ağlarının kullanımının yapısal optimizasyon problemleri üzerindeki potansiyelini incelemektedir. Yapılan analizler, YSA tabanlı yöntemlerin doğruluğunun FEM seviyesine ulaşamasa da hesaplama süresinde sağladığı önemli azalmanın bu alanda heyecan verici fırsatlar sunduğunu göstermektedir.

Çalışma, yedi farklı yapı örneği üzerinde üç hibrit BTO optimizasyon yönteminin (YT-1, YT-2 ve YT-K) standart FEM yöntemiyle karşılaştırılmasını içermektedir. Her bir yöntem, yakınsama açısından farklı özellikler sergilemiştir. YT-1, minimum ağırlık değerlerinde zaman içinde önemli bir değişiklik göstermese de diğer sezgisel yöntemlere kıyasla daha üstün bir yakınsama sergilemiştir. Bu da tutarlı ve güvenilir optimizasyon potansiyelini ortaya koymaktadır. YT-K, YT-1 ve YT-2'nin güçlü yönlerini birleştirerek neredeyse ideal bir yakınsama grafiği sunmuştur. Ancak, birkaç örnekte yakınsamada zorluk yaşamıştır, bu da optimizasyon yeteneklerinde potansiyel sınırlamalar olduğunu göstermektedir. YT-2 yöntemi ise başlangıç çözümlerine büyük ölçüde bağımlı kalmış ve optimizasyon süreci boyunca önemli bir gelişme göstermemiştir.

Farklı yapılar üzerindeki yöntemlerin analizi ilginç sonuçlar ortaya koymuştur. Her iki konsol giriş örneğinde de, YT-1 ve YT-K'nın FEM çözümlerine doğru yakınsadığı ve YT-K'nın YT-1'e göre biraz daha iyi performans sergilediği gözlemlenmiştir. Dar kule ve geniş kule örneklerinde, YT-K'nın karmaşık yapılar için optimum sonuçlar elde etme yeteneği üstün bir yöntem olduğunu ortaya koymuştur. Klasik köprü örneklerinde ise, FEM ve YT-1 benzer performanslar sergileyerek köprü benzeri yapılar için etkinliğini vurgulamıştır. Piramit yapısında ise FEM de dahil olmak üzere tüm yöntemlerin ideal bir yakınsama göstermekte zorlandığı gözlemlenmiştir. Bu durum, benzersiz yapı formlarını optimize etmek için daha fazla araştırma ve geliştirmenin gerekliliğini vurgulamaktadır.

Yöntemlerin istatistiksel performansının analizi, tutarlılıkları ve değişkenlikleri hakkında önemli bilgiler sağlamıştır. FEM yöntemi, geniş bir değer aralığını kapsayarak problem uzayında kapsamlı bir araştırma yapmıştır. Ancak, yüksek standart sapma sonuçlarda bazı tutarsızlıklar olduğunu göstermektedir. YT-1, birçok durumda dikkate değer bir tutarlılık sergilemiştir. Ancak, çözümleri genellikle diğer yöntemlere kıyasla daha yüksek ağırlık değerleriyle sonuçlanmıştır. YT-K da YT-1'e benzer şekilde tutarlı sonuçlara yönelmiş, ancak FEM'e kıyasla biraz daha yüksek ağırlık değerleri elde etmiştir. YT-2 ise her örnek için sürekli olarak aynı çözümü üretmiş ve deterministik doğasını vurgulamıştır. Ancak, çözümleri genellikle optimumdan uzaktır, bu da iyileştirme gerekliliğini vurgulamaktadır. Bu bulgular, uygun bir BTO yöntemi seçerken ortalama performansı ve sonuçların değişkenliğini dikkate almanın önemini vurgulamaktadır.

Farklı yöntemler arasında hesaplama süresinin karşılaştırılması, YSA tabanlı yaklaşımların avantajını daha da güçlendirmiştir. FEM, hesaplama süresi ile yapının serbestlik dereceleri (YSD) arasında doğrudan bir korelasyon göstermiştir; daha yüksek YSD, daha uzun optimizasyon sürelerine yol açmıştır. YT-1, YT-2 ve YT-K yöntemleri ise hesaplama hızı açısından sürekli olarak FEM'den daha iyi performans göstermiştir, bu da YSA ile geliştirilmiş arama yöntemlerinin önemli zaman tasarrufu potansiyelini göstermektedir. Analiz sonuçları, zaman verimliliğinin önemli bir faktör olduğu

durumlarda YSA tabanlı arama yöntemlerinin yapısal optimizasyon için FEM'e uygulanabilir bir alternatif sunduğunu net bir şekilde ortaya koymaktadır.

Bu çalışma, yapay zekâ destekli yapısal optimizasyon alanında daha fazla keşif ve gelişme için kapılar açmaktadır. Transfer öğrenmesinin mevcut YSA modelleriyle entegrasyonunu araştırmak veya yeni birleşik modeller geliştirmek, optimizasyon sürecinin doğruluğunu ve verimliliğini artırabilir. Araştırmanın karmaşık ve 3 boyutlu geometrilere veya malzeme özelliklerine sahip yapılar gibi daha geniş bir yapı yelpazesini kapsaması, YSA tabanlı yöntemlerin yetenekleri ve sınırlamaları hakkında daha kapsamlı bir anlayış sağlayabilir.

Geliştirilen BTO arama yöntemleri sayesinde, gerçek zamanlı BTO süreçleri tasarım aşamasına entegre edilebilir. Bu sayede, estetik ve fonksiyonel gereksinimler ile verimli malzeme kullanımı arasındaki denge gözetilerek, daha bütüncül bir yapısal tasarım yaklaşımına ulaşılması da mümkün olabilecektir. Böylece, tasarım sürecinde keşfedilen sinerji hem işlevsel hem de estetik açıdan başarılı yapıların ortaya çıkmasını kolaylaştıracaktır.

Bu araştırma, yapısal optimizasyonda gelecekteki ilerlemeler için bir adım taşıyarak, yapısal tasarım ve analizine daha verimli ve sürdürülebilir bir yaklaşımın yolunu açmaktadır. Gelecekteki çalışmalar, daha karmaşık yapısal problemler için YSA modellerinin geliştirilmesini ve optimizasyon algoritmalarıyla entegrasyonunu araştırabilir. Ayrıca, transfer öğrenmesi ve yeni birleşik modeller gibi yaklaşımlar, YSA tabanlı optimizasyonun doğruluğunu ve verimliliğini daha da artırabilir.

6. KAYNAKLAR

- McCulloch ve Pitts (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5: 115-133.
- Razani (1965). Behavior of fully stressed design of structures and its relationship to minimum-weight design. AIAA journal 3(12): 2262-2268.
- Tucker (1966). Some mathematical notes on three-mode factor analysis. Psychometrika 31(3): 279-311.
- Gallagher ve Zienkiewicz (1973). Optimum structural design: Theory and applications. CiNii.
- Imai ve Schmit Jr (1981). Configuration optimization of trusses. Journal of the Structural Division 107(5): 745-756.
- Rumelhart, Hinton ve Williams (1986). Learning representations by back-propagating errors. nature 323(6088): 533-536.
- Cybenko (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2(4): 303-314.
- LeCun, Boser, Denker, Henderson, Howard, Hubbard ve Jackel (1989). Backpropagation applied to handwritten zip code recognition. Neural computation 1(4): 541-551.
- Bishop (1995). Neural networks for pattern recognition. Oxford university press:
- Das ve Dennis (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM journal on optimization 8(3): 631-657.
- Haykin (1998). Neural networks: a comprehensive foundation. Prentice Hall PTR:
- Papadrakakis, Lagaros ve Tsompanakis (1998). Structural optimization using evolution strategies and neural networks. Computer methods in applied mechanics and engineering 156(1-4): 309-333.
- Patnaik, Gendy, Berke ve Hopkins (1998). Modified fully utilized design (MFUD) method for stress and displacement constraints. International journal for numerical methods in engineering 41(7): 1171-1194.
- Iranmanesh ve Kaveh (1999). Structural optimization by gradient-based neural networks. International journal for numerical methods in engineering 46(2): 297-311.
- Cormen, Leiserson, Rivest ve Stein (2001). 15.2: Matrix-chain multiplication. Introduction to Algorithms. Second Edition. MIT Press and McGraw-Hill. pp. 331-338. ISBN 978-0-262-03293 3.
- Papadrakakis ve Lagaros (2002). Reliability-based structural optimization using neural networks and Monte Carlo simulation. Computer methods in applied mechanics and engineering 191(32): 3491-3507.
- Lee ve Geem (2004). A new structural optimization method based on the harmony search algorithm. Computers & structures 82(9-10): 781-798.
- Shepherd, Chen ve Greer (2004). Olfactory bulb. Sensory Systems II: 66.

- Salajegheh ve Gholizadeh (2005). Optimum design of structures by an improved genetic algorithm using neural networks. *Advances in Engineering Software* 36(11-12): 757-767.
- Simon (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation* 12(6): 702-713.
- Murphy (2012). *Machine learning: a probabilistic perspective*. MIT press:92-103
- Jin, Dondar ve Culurciello (2014). Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*.
- Simonyan ve Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Srivastava, Hinton, Krizhevsky, Sutskever ve Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1): 1929-1958.
- Aydogdu ve Akin (2015). Biogeography based co2 and cost optimization of RC cantilever retaining walls. *17th international conference on structural engineering*, 1480-1485.
- Gal ve Ghahramani (2015). Dropout as a Bayesian approximation: Insights and applications. *deep learning workshop, ICML*, 2.
- Goodfellow, Bengio ve Courville (2016). *Deep learning*. MIT press:
- Saka, Hasançebi ve Geem (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation* 28: 88-97.
- Aydogdu (2017). Cost optimization of reinforced concrete cantilever retaining walls under seismic loading using a biogeography-based optimization algorithm with Levy flights. *Engineering Optimization* 49(3): 381-400.
- Sarshar, Tranquilli, Pickering, McCall, Roy ve Sandu (2017). A numerical investigation of matrix-free implicit time-stepping methods for large CFD simulations. *Computers & Fluids* 159: 53-63.
- Aslimani ve Ellaia (2018). A new chaos optimization algorithm based on symmetrization and levelling approaches for global optimization. *Numerical Algorithms* 79(4): 1021-1047.
- Örmecioğlu (2019). Çelik çerçeve sistemlerin GPU tabanlı optimizasyon. *İnşaat Mühendisliği Akdeniz Üniversitesi*.
- Islam (2020). Reduced dataset neural network model for manuscript character recognition.
- Izyumov, Vasilyeva, Alentsov ve Ostapovich (2021). Self-optimization of mobile robot actions using voice commands. *IOP Conference Series: Materials Science and Engineering*, 012128, IOP Publishing.
- Freitag, Peters, Edler ve Meschke (2022). Reliability-based optimization of structural topologies using artificial neural networks. *Probabilistic Engineering Mechanics* 70: 103356.

- Ganaie, Hu, Malik, Tanveer ve Suganthan (2022). Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence 115: 105151.
- Li, An, Lei, Deng ve Wang (2022). Survey of lévy flight-based metaheuristics for optimization. Mathematics 10(15): 2785.
- Taherdoost ve Madanchian (2023). Multi-criteria decision making (MCDM) methods and concepts. Encyclopedia 3(1): 77-87.
- Anonim 4 Evriřimli Sinir Ağlarının Görselleřtirilmesi. <https://dltr.asmaamir.com/3-cnn-konseptleri/3-gorsellestirme> [Son eriřim tarihi: 18.05.2024].
- Anonymous 1 Backpropagation. <https://en.wikipedia.org/wiki/Backpropagation> [Son eriřim tarihi: 18.05.2024].
- Anonymous 2 Building a Road Sign Classifier. <https://towardsdatascience.com/road-sign-classification-learning-to-build-a-cnn-7771373179d3> [Son eriřim tarihi: 18.05.2024].
- Anonymous 3 Deep Learning. https://bigdata.gazi.edu.tr/akcayol/files/DL_L1_Introduction.pdf [Son eriřim tarihi: 18.05.2024].
- Anonymous 5 HCIA-AI-Course. <https://github.com/gabboraron/HCIA-AI-Course/blob/main/README.md> [Son eriřim tarihi: 18.05.2024].
- KERAS KERAS 3 API documentation. <https://keras.io/api/> [Son eriřim tarihi: 18.05.2024].
- python.org Python's documentation. <https://www.python.org/doc/> [Son eriřim tarihi: 18.05.2024].
- SciPy.org Scipy.ndimage — SciPy v1.7.1 Manual. <https://docs.scipy.org/doc/scipy/reference/ndimage.html> [Son eriřim tarihi: 18.05.2024].
- tensorflow.org TensorFlow basics. <https://www.tensorflow.org/guide/basics> [Son eriřim tarihi: 18.05.2024].

ÖZGEÇMİŞ

Tevfik Oğuz ÖRMECİOĞLU

ÖĞRENİM BİLGİLERİ

Doktora 2019-2024	Akdeniz Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Ana Bilim Dalı, Antalya
Yüksek Lisans 2018-2019	Akdeniz Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Ana Bilim Dalı, Antalya
Lisans 2006	Akdeniz Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Ana Bilim Dalı, Antalya

MESLEKİ VE İDARİ GÖREVLER

Ortak & Müdür 2022-...	SOAD Yazılım İnşaat Şti.
Ortak & Müdür 2016-...	Yanartaş İnşaat Şti.
Ortak & Müdür 2007-2016	Örmecioğlu İnşaat Şti.

ESERLER

Uluslararası hakemli dergilerde yayımlanan makaleler

- Taymus, R. B., Aydogdu, I., Carbas, S., & Örmecioğlu, T. O. (2024). Seismic design optimization of space steel frame buildings equipped with triple friction pendulum base isolators. Journal of Building Engineering, 109748. DOI: 10.1016/j.job.2024.109748 (SCIE-Q1)
- Örmecioğlu, T. O., Aydoğdu, İ., & Örmecioğlu, H. T. (2024). GPU-based parallel programming for FEM analysis in the optimization of steel frames. Journal of Asian Architecture and Building Engineering, 1-22. DOI: 10.1080/13467581.2024.2345310 (SCIE-Q4)
- Örmecioğlu, Er Akan, A., Bingol, K., Örmecioğlu, H. T., Er, A., & Örmecioğlu, T. O. (2024). Towards an earthquake-resistant architectural design with the image

classification method. Journal of Asian Architecture and Building Engineering, 23(1), 157-170.

DOI: 10.1080/13467581.2023.2213299 (SCIE-Q4)

4. Aydogdu İ, Örmecioglu T.O., Tunca O, Carbas.S, 2022, “Design of large-scale real-size steel structures using various modified grasshopper optimization algorithms”, Neural Computing and Applications, (34), pp.13825–13848.
DOI: 10.1007/s00521-022-07196-3 (SCIE-Q2)
5. Aydoğdu İ., Örmecioglu, T.O., Carbas, S. 2021, “Electrostatic discharge algorithm for optimum design of real size truss structures” (Eds.Carbas, S., Toktas, A., & Ustun, D.). Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications. Springer Tracts in Nature-Inspired Computing, Springer Publishing.
DOI: 10.1007/978-981-33-6773-9_5 (Sense B-eBook)
6. Örmecioglu H.T., Örmecioglu T.O., 2021, “Tarihi Betonarme Yapılarda Bozulmalar ve Tahribatsız Hasar Tespit Yöntemleri” 2nd International Congress on Engineering Sciences and Multidisciplinary Approaches, 1(1), 276-282.
7. Tunca, O., Aydoğdu İ., Örmecioglu, T.O., Carbas, S. 2018, “Grasshopper-Genetic Optimization Algorithm Based Design of Structures” International Conference on Applied Mathematics, Istanbul.
8. Aydoğdu İ., Örmecioglu, T.O., 2018, “Optimization of High-Rise Buildings having Concrete Encased Steel Column using Tree Seed Algorithm” 4th International Conference on Mechanics of Composites, Madrid.