T.C

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN GELİŞTİRİLMESİ VE SİSTEM MODELLEMEDEKİ BAŞARIMLARININ ANALİZİ

DOKTORA TEZİ

GİZEM ATAÇ KALE

TEZ DANIŞMANI

PROF. DR. CİHAN KARAKUZU

BİLECİK, 2022

10510344

T.C

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN GELİŞTİRİLMESİ VE SİSTEM MODELLEMEDEKİ BAŞARIMLARININ ANALİZİ

DOKTORA TEZİ

GİZEM ATAÇ KALE

TEZ DANIŞMANI

PROF. DR. CİHAN KARAKUZU

BİLECİK, 2022

10510344

BEYAN

"Çok Katmanlı Aşırı Öğrenme Makinelerinin Geliştirilmesi ve Sistem Modellemedeki Başarımlarının Analizi" adlı Doktora tezi hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

<table>
<tr><td colspan="4">Bu çalışmanın,
Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.</td></tr>
<tr><td>DESTEK ALINMIŞTIR</td><td></td><td>DESTEK ALINMAMIŞTIR</td><td>X</td></tr>
<tr><td colspan="4">Destek alındı ise;</td></tr>
<tr><td colspan="4">Destekleyen kurum;</td></tr>
<tr><td colspan="2">Desteğin Türü</td><td colspan="2">Proje Numarası</td></tr>
<tr><td colspan="2">1- BAP (Bilimsel Araştırma Projesi)</td><td colspan="2" rowspan="3"></td></tr>
<tr><td colspan="2">2- TÜBİTAK</td></tr>
<tr><td colspan="2">Diğer;……</td></tr>
<tr><td colspan="4">ETİK KURUL onayı var ise;</td></tr>
<tr><td colspan="2">ETİK KURUL karar tarih/sayı:</td><td colspan="2">……/……</td></tr>
</table>

Gizem ATAÇ KALE

Tarih

…….....

İmza

…….....

ÖN SÖZ

Tez çalışmamda benden bilgisini, tecrübesini ve desteğini esirgemeyen sayın hocam Prof. Dr. Cihan KARAKUZU'ya ve tezin ilerlemesinde emekleri olan tez izleme jüri üyelerim; sayın Prof. Dr. Alpaslan DUYSAK ve Prof. Dr. Uğur YÜZGEÇ'e teşekkürlerimi ve saygılarımı sunarım.

Ayrıca, maddi ve manevi desteklerini esirgemeyen aileme, özellikle de eşim Mehmet Cemil KALE ve kızım Gönül Serpil KALE'ye tüm kalbimle teşekkür ederim.

Gizem ATAÇ KALE

2022

ÖZET

ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNELERİNİN GELİŞTİRİLMESİ VE SİSTEM MODELLEMEDEKİ BAŞARIMLARININ ANALİZİ

Aşırı Öğrenme Makinesi (AÖM), sınıflandırma ve regresyon uygulamalarında yüksek verimlilik ve kolay uygulanabilirdiğinden dolayı son on yılda önemli araştırma konusu olmuştur. Çalışmamız kapsamında regresyon ve sınıflandırma problemleri için geliştirilen klasik AÖM, Radyal Tabanlı Fonksiyon Kullanan Aşırı Öğrenme Makinesi (RTF-AÖM), Meta Aşırı Öğrenme Makinesi (Meta-AÖM) ve Çok Katmanlı Aşırı Öğrenme Makinesi (ÇK-AÖM) gibi mevcut AÖM yapıları incelenmiştir. Bu çalışmada, anılan yapıların zayıf ve üstün yönleri incelenerek yapıların iyileştirilmesi ve yeni hibrit yapılar önerilmiştir. Çalışmamızdaki motivasyon unsuru, AÖM yöntemlerinin yaygın olarak sınıflandırma problemlerinde kullanılmış olması ve sistem modelleme üzerinde çalışılmamış olmasıdır. Bu bağlamda zamana dinamik bir şekilde bağlı sistemlerin yanısıra engebeli ve/veya keskin değişimli yüzeylere denektaşı sistemlerin modellenmesi üzerinde çalışılarak iyileştirme ve yeni karma ağ yapısının geliştirilmesi hedeflenmiş olup çalışmamızda istenen hedefe ulaşılmıştır. ÇK-AÖM yapısının sistem modelleme başarımını arttırmayı ve hesaplama zamanını düşürmeyi amaçlayan iki İyileştirilmiş Çok Katmanlı Aşırı Öğrenme Makinesi (İÇK-AÖM) adını verdiğimiz AÖM ağ yapıları geliştirilmiş olup deney sonuçları ile başarım üstünlünlükleri denektaşı sistemler üzerinde gösterilmiştir. İÇK-AÖM yapıları ile yedi farklı dinamik sistem modelleme uygulaması üzerinde başarımı incelenmiştir. Ayrıca çok katmanlı aşırı öğrenme makinesi ile radyal tabanlı fonksiyon ağları birleştirerek HybRBF-ML-ELM adını verdiğimiz karma bir ağ yapısı daha geliştirilmiştir. Engebeli ve/veya keskin değişimli yüzey problemlerinde başarımı deney sonuçları ile kıyaslamalı olarak gösterilmiştir.

Dinamik sistem modelleme deneylerinde geliştirilen İÇK-AÖM'nin bazı durumlarda ÇK-AÖM'ye kıyasla hem eğitim hem de test veri setleri için %70lere varan daha iyi modelleme başarım sergilediği gözlenmiştir. Örneğin, Denektaşı Dinamik Sistem (DDS) 7 için 100 düğüm kullanıldığında ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 yapıları 0.627977, 0.104272 ve 0.092683 ortalama test Karesel Ortalama Hatanın Karakökü (RMSE) sonuçları verilmiştir. ÇK-AÖM ile karşılaştırıldığında İÇK-AÖM1 yapısı 83% daha iyi ortalama test RMSE sonuçları sağlarken İÇK-AÖM2 yapısı 85% daha iyi ortalama test RMSE sonuçları sağlamıştır. Yüzey modelleme deneylerinde geliştirilen HybRBF-ML-ELM yapısının ÇK-AÖM yapısına göre her problem için daha başarılı olduğu gözlemlenmiştir. Örneğin, *Griewank, Dropwave* ve *Schaffer2*

fonksiyonları için hem eğitim hem de test RMSE sonuçlarına bakıldığında ÇK-AÖM yapısına göre yaklaşık olarak %99 daha iyi başarım elde edildiği görülmüştür. *Peaks* fonksiyonu için de eğitim verisinde ortalama RMSE değerine bakıldığında ÇK-AÖM yapısına göre %76,87 daha iyi modelleme başarım sergilediği gözlenmiştir.

Anahtar Kelimeler: Sistem modelleme, dinamik sistem, keskin değişimli yüzey, aşırı öğrenme makinesi, çok katmanlı aşırı öğrenme makinesi.

ABSTRACT

IMPROVEMENT OF MULTILAYER EXTREME LEARNING MACHINES AND THEIR ANALYSIS OF PERFORMANCE IN SYSTEM MODELING

Extreme Learning Machine (ELM) has been the subject of significant research in the last decade due to its high efficiency and easy implementation in classification and regression applications. Within the scope of our study, existing ELM structures such as classical ELM, Radial Basic Function Extreme Learning Machine (RBF-ELM), Meta Extreme Learning Machine (Meta-ELM) and Multilayer Extreme Learning Machine (ML-ELM), which were developed for regression and classification problems, were examined. In this study, the weak and superior aspects of the aforementioned structures were examined, and the improvement of the structures and new hybrid structures were proposed. The motivation factor in our study is that ELM methods are widely used in classification problems and system modeling has not been studied. In this context, it was aimed to improve and develop a new mixed network structure by working on the modeling of touchstone systems on uneven and/or sharply changing surfaces as well as systems that are dynamically dependent on time, and the desired goal was achieved in our study. Two ELM network structures, which we call Improved Multilayer Extreme Learning Machine (IML-ELM), aiming to increase the system modeling performance of the ML-ELM structure and reduce the computation time, have been developed and the test results and performance advantages are shown on benchmark systems. Their performances on seven different dynamic system modeling applications compared to the ML-ELM structure performance have been examined. In addition, a hybrid network structure, which we call HybRBF-ML-ELM, has been developed by combining multi-layer extreme learning machine and radial basis function networks. Its performance on uneven and/or sharply changing surface problems has been shown in comparison with the test results.

It has been observed that IML-ELM developed in dynamic system modeling experiments exhibits up to 70% better modeling performance for both training and test datasets compared to ML-ELM in some cases. For example, when 100 nodes are used for Benchmark Dynamic System (BDS) 7, the average test Root Mean Square Errors (RMSE) results of 0.627977, 0.104272 and 0.092683 are achieved for the ML-ELM, IML-ELM1 and IML-ELM2 structures. Compared to the ML-ELM, the IML-ELM1 structure provided 83% better mean test RMSE results, while the IML-ELM2 structure provided 85% better mean test RMSE results. It has

been observed that the HybRBF-ML-ELM structure developed in surface modeling experiments is more successful for each problem than the ML-ELM structureFor example, when both the training and test RMSE results of HybRBF-ML-ELM for the *Griewank*, *Dropwave* and *Schaffer2* functions are examined, it is seen that approximately 99% better performance is obtained compared to the ML-ELM structure. Considering the average RMSE value in the training data for the *peaks* function, it was observed that the modeling performance was 76.87% better than the ML-ELM structure.

Keywords: System modeling, dynamic system, sharply altered surface, extreme learning machine, multilayer extreme learning machine.

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

KISALTMALAR VE SİMGELER LİSTESİ

b_i : Eşik, i=1 … L

β_i : Çıkış Ağırlıkları i=1 … L

fi : Radyal Tabanlı Fonksiyon

g : Aktivasyon Fonksiyonu

H : Katmanın Çıktı Matrisi

T : Eğitim verisi hedef matrisi

λ : Düzenleme faktörü

L : Düğüm yada hücre sayısı

M : Katman Sayısı

μ : Çekirdeğin Merkezi

N : Harici Giriş Sayısı

$\tilde{N}$: Çekirdek Sayısı

O : Çıkış Matrisi

o_j : O Çıkış Matrisinin Elamanları j=1 … P

P : Örnek Sayısı

S : Aşırı Öğrenme Makine Sayısı

σ : Standart Sapma

T : Eğitim Verisinin Hedef Matrisi

u : Harici Kontrol Sinyali

y : Sistem Çıkışı

X : Giriş Matrisi

x_j : Giriş, j=1 … P

w_i : Giriş Ağırlıkları, i=1 … L

q : Çıkış Sayısı

AE : Otomatik Kodlayıcı (Auto Encoding)

AÖM : Aşırı Öğrenme Makinesi

ÇK-AÖM : Çok Katmanlı Aşırı Öğrenme Makinesi

İÇK-AÖM : İyileştirilmiş Çok Katmanlı Aşırı Öğrenme Makinesi

RMSE : Karesel Ortalama Hatanın Karakökü (Root Mean Square Errors)

DDS : Denektaşı Dinamik Sistemler (Benchmark Dynamic Systems)

META-AÖM : META Aşırı Öğrenme Makinesi

RTF : Radyal Tabanlı Fonksyon

RTFA : Radyal Tabanlı Fonksiyon Ağı

RTF-AÖM : Radyal Tabanlı Fonksiyon Kullanan Aşırı Öğrenme Makinesi

SLFN : Tek Gizli Katmanlı İleri Beslemeli Sinir Ağı

YSA : Yapay Sinir Ağları

İÇİNDEKİLER

Sayfa

1. GİRİŞ

Yapay öğrenme günümüzde bir çok problemin çözümünde yaygın olarak kullanılan kavramdır. Yapay sinir ağları (YSA) bu kavram ile ortaya çıkmıştır. YSA, birçok mühendislik ve bilimsel problemin çözümü için yaygın olarak kullanılmaktadır ve popüler genel modelleme ve hesaplama yöntemleridir (Haykin S, 1999). Bu yapılar genellikle yinemeleri algoritmalar tarafından eğitilir. Alışıla gelmiş YSA eğitiminde kullanılan türeve dayalı öğrenme yöntemleri iyi başarım sonucu vermesine rağmen öğrenme süresi oldukça zaman almaktadır. Bu yüzden 2004 yılında Huang ve arkadaşları tarafından geliştirilen aşırı öğrenme makine (AÖM) yöntemi, analitik olarak hesaplama yaptığı için çok kısa sürede çok hızlı eğitim yapabilmektedir. Oldukça hızlı çözüme ulaştığı için bu alanda bir çok araştırmacı bir çok problem için AÖM kullanmıştır. AÖM'nin belirgin zayıf yönlerini iyileştirmek için bir çok yeni AÖM yöntemleri geliştirilmiştir. Literatür taraması bölümünde bu çalışmalardan ve geliştirilen AÖM yöntemlerinden kronolojik olarak kısaca bahsedilmiştir.

1.1. Literatür Taraması

*Aşırı öğrenme makin**esi*** 2004 yılında Huang ve arkadaşları tarafından tek gizli katmanlı ileri beslemeli sinir ağ yapısında kullanılan yeni analitik öğrenme algoritmasıdır (G. bin Huang vd., 2004). Sinir ağlarında yaygın kullanılan türev tabanlı öğrenme yöntemlerinde öğrenme fazla sayıda eğitim/test verisi ve tur sayısı fazla olduğu için geliştirilen AÖM yöntemine göre oldukça yavaştır. AÖM'de giriş katmanındaki ağırlık parametrelerini rastgele atar ve çıkış katmanındaki ağırlık parametreleri analitik olarak belirlenir. Bu yüzden AÖM ile öğrenme hızlı ve başarımı yüksektir.

2004 yılında Huang ve arkadaşları tarafından *'Extreme Learning Machine: RBF Network Case'* başlıklı çalışma ile AÖM yapısı radyal tabanlı fonksiyon (RTF) ağ durumuna göre genişletilmiştir. RTF'deki merkez ve varyans rastgele atanır ve çıkış ağırlık parametreleri iteratif olarak ayarlanmak yerine analitik olarak hesaplanır (G. bin Huang ve Slew, 2004) . Makaledeki deney sonuçları ile RTF ağları için AÖM algoritmasının öğrenmeyi son derece hızlı bir şekilde tamamlayabildiğini ve birçok yapay ve gerçek kıyaslama fonksiyonu yaklaşımı ve sınıflandırma probleminde destek vektör makinelerininkine (SVM) çok yakın bir genelleme başarımı ürettiği gösterilmiştir.

2005 yılında Huang ve arkadaşları tarafından yazılan *'On-Line Sequential Extereme Learning Machine'* adlı makalede, Çevrimiçi Sıralı Aşırı Öğrenme Makinesi (OS-ELM) olarak adlandırılan özyinelemeli en küçük kareler (RLS) algoritmasına dayanan sıralı modifikasyonu

sunulmuştur. OS-ELM'e dayanarak, Sıfır dereceli TSK modelini ve birinci dereceden TSK modelini uygulamak için Çevrimiçi Sıralı Bulanık Ekstrem Öğrenme Makinesi (Fuzzy-ELM) de tanıtıldı (G.-B. Huang vd.,2005) OS-ELM ve Fuzzy-ELM, diğer popüler sıralı öğrenme algoritmaları ile kıyaslanmıştır ve deney sonuçlarına göre bazı gerçek regresyon benchmark problemlerinde daha hızlı öğrenme ve iyi genelleme başarımı vermiştir.

AÖM'de giriş katmanındaki ağırlık parametreleri ve eşik parametreleri rastgele olarak belirleniyordu fakat 2005 yılında Huang ve arkadaşları tarafından yayınlanan *'Evolutionary extreme learning machine'* başıklı makalede karma öğrenme algoritması geliştirilmiştir. Bu yöntem ile rastgele atanan bu parametreler seçmek için farksal gelişim algoritmasını ve çıktı ağırlıklarını analitik olarak belirlemek için Moore Penrose (MP) yöntemini kullanan bir karma öğrenme algoritması önerilmiştir. Bu çalışmadaki deney sonuçlarına bakarak, bu yaklaşımın çok daha kompakt ağlarla iyi genelleme başarımı sağlayabildiği bildirilmiştir (Zhu vd., 2005).

2009 yılında Rong ve arkadaşları tarafından yayınlanan, *'Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems' (OS-Fuzzy-ELM)* adlı çalışmada, fonksiyon yaklaşımı ve sınıflandırma problemleri için çevrimiçi sıralı bulanık aşırı öğrenme makinesi (OS-Fuzzy-ELM) geliştirilmiştir (Rong vd., 2009).

Lan ve arkadaşları 2010 yılında *'Constructive hidden nodes selection of ELM'* başlıklı çalışmada AÖM algoritmasına dayalı yapıcı bir yöntem uygulayarak AÖM regresörünün mimari tasarımını ele almışlardır (Lan vd., 2010a). Önerilen CS-ELM'in diğer AÖM ağları (AÖM, EM-ELM ve I-ELM) ile karşılaştırılması, bazı gerçek kıyaslamalı regresyon uygulamalarında yapılmıştır.

'Two-stage extreme learning machine for regression' çalışması 2010 yılında Lan ve arkadaşları tarafından yayınlamıştır. Bu çalışmada AÖM'in eksik kalan ön ağ yapısının belirlenmesi üzerinde durmuşlardır. Bu çalışma kapsamında geliştirilen TS-ELM yöntemi iki aşamadan oluşmaktadır. Birinci aşamada, ileriye dönük öz yinelemeli algoritma her adımda rastgele üretilen bireylerden gizli düğümü seçmek için uygulanır ve düğümler durdurma kriteri minimum değere gelene kadar ağa eklenir. İkinci aşamada, her gizli düğümün önemi gözden geçirilir ve önemsiz olanlardan ağdan çıkarılır, ki bu da ağ karmaşıklığını önemli oranda azaltır. TS-ELM, denektaşı (benchmark) regresyon problemleri için AÖM ve EM-ELM ile karşılaştırılmıştır. AÖM ile kıyaslandığında daha iyi sonuç vermektedir, diğer yandan EM-ELM ile kıyaslandığında da benzer ya da daha iyi sonuç verdiği görülmektedir (Lan vd., 2010b).

2013 yılında Zhang ve Ji tarafından yayınlanan *'Fuzzy Extreme Learning Machine for Classification'* adlı çalışmada FELM yapısı önerilmiştir. Önerilen bu yapı, geleneksel AÖM yapısındaki oransız problemler ve ağırlıklı sınıflandırma problemleri gibi farklı giriş noktalarında sınıfların birini tam olarak atamayabilme eksikliğini giderebilmektedir. Geleneksel AÖM yöntemine bulanık mantık kavramının dahil edildiği bir yapı olarak görülebilir. Ağırlıklı sınıflandırma problemleri için FELM, AÖM'den daha mantıklı sonuçlar verildiği deneysel sonuçlar ile gösterilmiştir (W. B. Zhang ve Ji, 2013).

R. Zhang ve arkdaşları tarafından 2013 yılında yayınlanan *'Dynamic Extreme Learning Machine and Its Approximation Capability'* başlıklı çalışmada dinamik aşırı öğrenme makinesi yöntemi geliştirilmiştir. (R. Zhang vd., 2013).

2013 Han ve arkadaşları, *'Hierarchical extreme learning machine for feedforward neural network'* isimli çalışmasında bir hiyerarşik ileri besleme sinir ağı (HFNN) ağırlıklarının eğitimi için bir yapı önerilmiştir. Geleneksel tek gizli katman ileri besleme ağlarından (SLFN'ler) farklı olarak, bu hiyerarşik AÖM (HELM) çevrimiçi olarak sıralı bilgileri hiyerarşik öğrenebilen hiyerarşik bir yapıya dayanır (Han vd., 2014).

2013 yılında Liao ve arkadaşları tarafından *'Meta-ELM: ELM with ELM hidden nodes'* isimli çalışmada Meta-ELM adında AÖM'nin meta öğrenme modeli önerilmiştir. Meta-ELM'in yapısı bir kaç temel AÖM'den ve bir üst AÖM'den oluşur. Bu nedenle, Meta-AÖM öğrenmesi iki aşamada ilerler. İlk olarak, her bir temel AÖM, eğitim verilerinin bir alt kümesi üzerinde eğitilmiştir. Daha sonra, üst AÖM eğitilen temel AÖM'leri sanki gizli düğümlermiş gibi alarak tüm very ile eğitilir (Liao ve Feng, 2014).

Yong arkadaşları ile 2014 yılında gerçekleştirdiği *'Meta-cognitive Fuzzy Extreme Learning Machine'* adlı çalışmasında meta-bilişsel bulanık aşırı öğrenme makinesi (McFELM) olarak adlandırılan sinirsel bulanık çıkarım sistemi (NFIS) için hızlı bir öğrenme metodolojisi önermiştir. Öğrenmeyi daha etkili kılmak için insan meta-biliş ilkelerini içeren orijinal OS-Fuzzy-ELM algoritması kullanılmıştır (Yong vd., 2014).

Mevcut AÖM yapılarının, AÖM'nin başarımı için çok önemli olan çekirdek seçimini optimize etmeye çok az dikkat edildiği gerekçesiyle 2015 yılında Liu ve arkadaşları *'Multiple Kernel Extreme Learning Machine'* isimli çalışmada bu bahsedilen eksikliği gidermek için çoklu çekirdek aşırı öğrenme makineleri (MK-ELM) olarak adlandırılan genel bir öğrenme yöntemi önermişlerdir. Önerilen MK-ELM'de, optimum çekirdek kombinasyon ağırlıkları ve AÖM'in yapısal parametreleri birlikte optimize edilmiştir (Liu vd., 2015).

Jagtap ve arkadaşları tarafından 2015 yılında yayınlanan *'Extreme-ANFIS: A Novel Learning Approach for Inverse Model Control of Nonlinear Dynamical Systems'* isimli çalışmada Takagi-Sugeno bulanık çıkarım sisteminin (TS FIS) öncül ve sonuç parametrelerini ayarlamak için "Extreme ANFIS" adlı yeni, basit ve daha hızlı bir öğrenme yaklaşımı önerilmiştir (Jagtap vd., 2015).

AÖM'nin bir çevrim içi sürümü daha önceden ileri sürülmüştür fakat eksik yönlerinden dolayı Scardapen ve arkadaşları tarafından 2015 yılında *'Online Sequential Extreme Learning Machine with Kernels'* isimli çalışması ile çevrimiçi çekirdek tabanlı AÖM (KOS-ELM) ağı önerilmiştir (Scardapane vd., 2015).

2015 yılında B.Y. Qu ve arkadaşaları *'Two Hidden Layer Extreme Learning Machine for Regression and Classification'* isimli çalışmada özel bir yapı olarak 2 gizli katmanlı AÖM ağı tanımlamışlardır. Giriş katmanı ile birinci gizli katman arasındaki ağırlık ve eşik parametreleri klasik AÖM'deki gibi rastgele atanmaktadır. Birinci gizli katman ile ikinci gizli katman arasındaki ağırlık ve eşik parametreleri için ise yeni bir yöntem geliştirilmiştir, böylece iki gizli katmanlı ileri beslemeli ağ yapısı için yeni bir AÖM yöntemi tanımlanmıştır (Qu vd., 2016).

2016 yılında Tang ve arkadaşları AÖM'nin sığ mimarisi nedeniyle, AÖM kullanarak özellik öğrenme, çok sayıda gizli düğümde bile doğal sinyaller (örneğin görüntüler / videolar) için etkili olmayabilir tespitini yapmışlardır. Bu konuyu ele almak için, *'Extereme Learning Machine for Multilayer Perceptron '* adlı çalışmada, çok katmanlı algılayıcı için yeni bir AÖM tabanlı hiyerarşik öğrenme çerçevesi önerilmiştir. Önerilen mimari iki ana bileşene ayrılmıştır: 1) kendi kendine öğretilen özellik çıkarımı ve ardından denetlenen özellik sınıflandırması ve 2) rastgele başlatılan gizli ağırlıklar ile köprüleme. Yaygın olarak kullanılan çeşitli sınıflandırma veri setleri üzerinde yapılan kapsamlı deneyler, önerilen algoritmanın mevcut son teknoloji hiyerarşik öğrenme yöntemlerinden daha iyi ve daha hızlı yakınsamaya ulaştığını göstermektedir. Ayrıca, bilgisayar ortamında yapılan uygulamalar, önerilen öğrenme planının genelliğini ve kabiliyetini daha da doğrulamaktadır (Tang vd., 2016).

Büyük veri kümerlerinde öğrenme için giriş örnekleri düzgün bir şekilde alınamadığı için AÖM'nin sınıflandırma başarımı düşük olabilir. Bu sorun Phurattanaprapin ve Horata tarafından 2016 yılında *'Extended Hierarcihical Extreme Learning Machine with Multilayer Perceptron'* isimli çalışmada ele alınmıştır. Çok katmanlı algılayıcının hiyerarşik öğrenme mimarisine dayanarak hiyerarşik aşırı öğrenme makinesi (H-ELM) çerçevesi önerildi. H-ELM iki bölümden oluşur: İlki denetlenmemiş çok katmanlı kodlama kısmı ve ikinci kısmı

denetlenen özellik sınıflandırma kısmıdır. H-ELM, geleneksel AÖM'den daha yüksek doğruluk oranı verebilir, ancak, sınıflandırma başarımının da arttırması gerekmektedir. Bu nedenle, bu makale genişleyen hiyerarşik aşırı öğrenme makinesi (EH-ELM) gibi yeni bir yöntem önermektedir. EH-ELM'nin genişletilmiş denetleyici kısmı için, iki katmanlı aşırı öğrenme makinesinden yararlanılmaktadır. Üç farklı görüntü veri seti ile (Semeion, MNIST ve NORB) EH-ELM başarımı inceleme çalışmalarının deneysel sonuçları, EH-ELM'nin H-ELM'den ve diğer çok katmanlı çerçeveden daha iyi başarım elde ettiğini göstermektedir (Phurattanaprapin ve Horata, 2016).

Son zamanlarda, çok katmanlı aşırı öğrenme makinesi (ÇK-AÖM) temsili öğrenme için yığılmış otomatik kodlayıcıya (SAE) uygulandı. Geleneksel SAE'nin aksine, ÇK-AÖM'in eğitim süresi yüksek doğrulukla saatlerden saniyelere önemli ölçüde azaldı. Ancak ÇK-AÖM'in eksikleri olduğu için Chi Man Wong ve arkadaşları tarafından 2018 yılında, '*Kernel-Based Multilayer Extreme Learning Machines for Representation Learning'* isimli çalışmada ML-KELM adında bir öğrenme yöntemi ileri sürülmüştür. ÇK-AÖM birçok sakıncadan muzdariptir: 1) her katmandaki gizli düğümlerin sayısı üzerinde elle ayarlama, eğitim süresi ve genelleme için belirsiz bir faktördür; 2) ÇK-AÖM'nin her katmanındaki eşik ve giriş ağırlıklarının rastgele üretilmesi, yetersiz model genellemesine yol açar; 3) her katmandaki çıkış ağırlıkları için sözde ters çözümü nispeten büyük yeniden yapılanma hatası meydana getirir; ve 4) temsil öğreniminde dönüşüm matrislerinin depolanma ve yürütme süresi, gizli katmanların sayısıyla orantılıdır. Çekirdek öğrenmesinden esinlenerek, ÇK-AÖM'nin bir çekirdek sürümü , çok katmanlı çekirdek AÖM (ML-KELM) geliştirilmiştir: Bu yapının katkıları: 1) her katmanda gizli düğümlerin sayısının elle ayarlanmasının kaldırılması; 2) optimal model genellemesi elde etmek için rastgele üretim mekanizması yoktur; 3) çıktı ağırlıkları için tam ters çözüm, daha küçük yeniden yapılandırma hatasıyla sonuçlanan, ters çevrilebilir çekirdek matrisi altında garanti edilir; ve 4) tüm dönüşüm matrisleri sadece iki matriste birleştirilir, böylece depolama azaltılabilir ve model uygulama süresi kısaltabilir. Deneysel sonuçlar önerilen ML-KELM'in katkılarını doğrulamıştır. Benchmark veri setlerine göre doğruluktaki gelişme %7'ye kadar artmıştır (Wong vd., 2018a).

Dalgacıkların zaman-frekans düzlem özelliklerini ve sinir ağının (NN) öğrenme yeteneklerini, bulanık çıkarım sisteminin yaklaşık akıl yürütme özelliklerini ve AÖM'in son derece hızlı öğrenme hızında tek geçişli öğrenme ve iyi bir genelleme başarımına dahil ederek bir çok uygulamada etkili bir çözüm sergilenmektedir. Bu yüzden 2017 yılında Golestaneh ve arkadaşları tarafından yapılan *'Fuzzy Wavelet Extreme Learning Machine'* adlı yayında yeni

bulanık dalgacık aşırı öğrenme makinesi (FW-ELM) ileri sürülmüştür. FW-ELM'in temel amaçları, doğrusal öğrenme parametrelerinin sayısını azaltarak ağ karmaşıklığını önemli ölçüde azaltmak ve kabul edilebilir doğruluk ve genelleme başarımları korunurken rastgele başlatma ile oluşan duyarlılığı azaltmaktır. Önerilen yapıda, her bir bulanık kural bir alt dalgacık sinir ağına karşılık gelir ve farklı genişleme ve dönüşümlü dalgacıklardan oluşur. Bu modelde, ağ karmaşıklığı ve başarım doğruluğu arasındaki dengeyi sağlamak için her bir bulanık kuralın THEN kısmında, her iki giriş için bir katsayı göz önünde bulundurulur. Bu çalışmada, önce bir FW modelinin ve bir SLFN'nin denkliği kanıtlanmış ve daha sonra AÖM doğrudan modele uygulanabilir kılınmıştır. Üyelik fonksiyonun ve dalgacık katsayıların tüm serbest parametreleri rastgele üretilir ve sadece çıkış ağırlıkları analitik olarak belirlenir. FW-ELM'yi değerlendirmek için, çeşitli denektaşı verilerinde OS-Fuzzy-ELM, Simpl_eTS, ANFIS gibi popüler bulanık modelle ve AÖM, BP ve SVR gibi diğer ilgili algoritmalarla karşılaştırıldı. Deney sonuçları, önerilen yaklaşımın verimli sonuç verdiğini göstermektedir. FW-ELM'in başarım doğruluğunun OS-Fuzzy-ELM ile karşılaştırılabilir olduğu ve bilinen yöntemlerin geri kalanından daha iyi olduğu gösterilmiştir (Golestanesh vd., 2018).

AÖM'nin son zamanların popüler derin öğrenme ağlarının eğitimini hızlandırmak amaçlı çalışmalar da literatürde göze çarpmaktadır. Bu çalışmalardan bir kesit şu şekilde sunulabilir: Tissera ve McDonnell, sınıflandırma için denetimli otomatik kodlama mimarisi olarak bir derin uç öğrenme makinesi tasarladı (Tissera ve McDonnell, 2016). Kim ve arkadaşları AÖM kullanarak Evrişimli Sinir Ağları (CNN) için hızlı bir öğrenme yöntemi geliştirdi (Kim vd., 2017). Yousefi-Azar ve McDonnell, yarı denetimli bir evrişimli aşırı öğrenme makinesi tasarladı (Yousefi-Azar ve McDonnell, 2017). Vong ve arkadaşları ampirik çekirdek haritası tabanlı ML-ELM kullanarak temsili öğrenme üzerine başka bir araştırma yaptı. Altan ve Kutlu, bir Hessenberg ELM otomatik kodlayıcı geliştirdikleri derin öğrenme alanında çalıştı (Altan, 2018). Chen ve arkadaşları ML-ELM (Chen vd., 2019) kullanarak akıllı sağlık için bir insan etkinliği tanıma şeması geliştirdi. Zhang ve diğ. ML-ELM (J. Zhang vd., 2020) kullanarak yinelemeli olmayan ve hızlı bir derin öğrenme şeması geliştirdi. Ezzati Khatab ve diğ. bir otomatik kodlayıcı tabanlı yarı denetimli derin aşırı öğrenme makinesi geliştirdi (Ezzati Khatab vd., 2021)

1.2. Hedefler ve Katkılar

Çalışmamızda modelleme ve sınıflandırma problemleri için geliştirilen klasik AÖM, RTF-AÖM, Meta-AÖM ve ÇK-AÖM gibi mevcut AÖM yapıları incelenmiştir. Bu yapıların

artı ve eksi yönleri incelenerek yapıların iyileştirilmesi ve yeni hibrit yapı öne sürülmesi hedeflenmiştir.

Çalışmamızdaki motivasyon unsuru, AÖM yöntemleri yaygın olarak sınıflandırma problemlerinde kullanılmış olup sistem modelleme üzerinde çalışılmamıştır. Bu yüzden dinamik sistem ve engebeli, keskin değişimlere sahip sistemlerin modellenmesi üzerinde çalışılarak iyileştirme ve yeni karma ağ yapısının geliştirilmesi hedeflenmiş olup çalışmamızda istenen hedefe ulaşılmıştır.

Bu çalışmanın özgün katkıları aşağıda listelenmiştir.

1. AÖM ve geliştirilmiş olan diğer bir çok AÖM yöntemleri yaygın olarak sınıflandırma, görüntü sınıflandırma, regresyon ve tahmin problemlerinde kullanılmaktadır. Dinamik sistem modelleme üzerinde az çalışmıştır. Bu yüzden bu doktora tez çalışmasında AÖM yöntemlerinin dinamik sistem ve engebeli, keskin değişimli yüzeylere sahip modelleme problemlerine değilinilmiştir.
2. Dinamik sistem modelleme problemi üzerinde ÇK-AÖM yapısında iyileştirilme yapılmıştır. İyileştirilmiş iki farklı çok katmanlı AÖM yapısı geliştirilerek tanımlanmıştır. Geliştriren ağ yapılarının başarım üstünlüğü 5 farklı dinamik sistem üzerinde gösterilmiştir.
3. Engebeli, keskin değişimli yüzeylere sahip sistemlerin modellenmesi için de yeni bir karma ağ yapısı geliştirilmiştir. Bu ağın kıyaslamalı başarımı literatürden seçilen denektaşı işlevlerin modellenmesi çerçevesinde sunulmuştur.

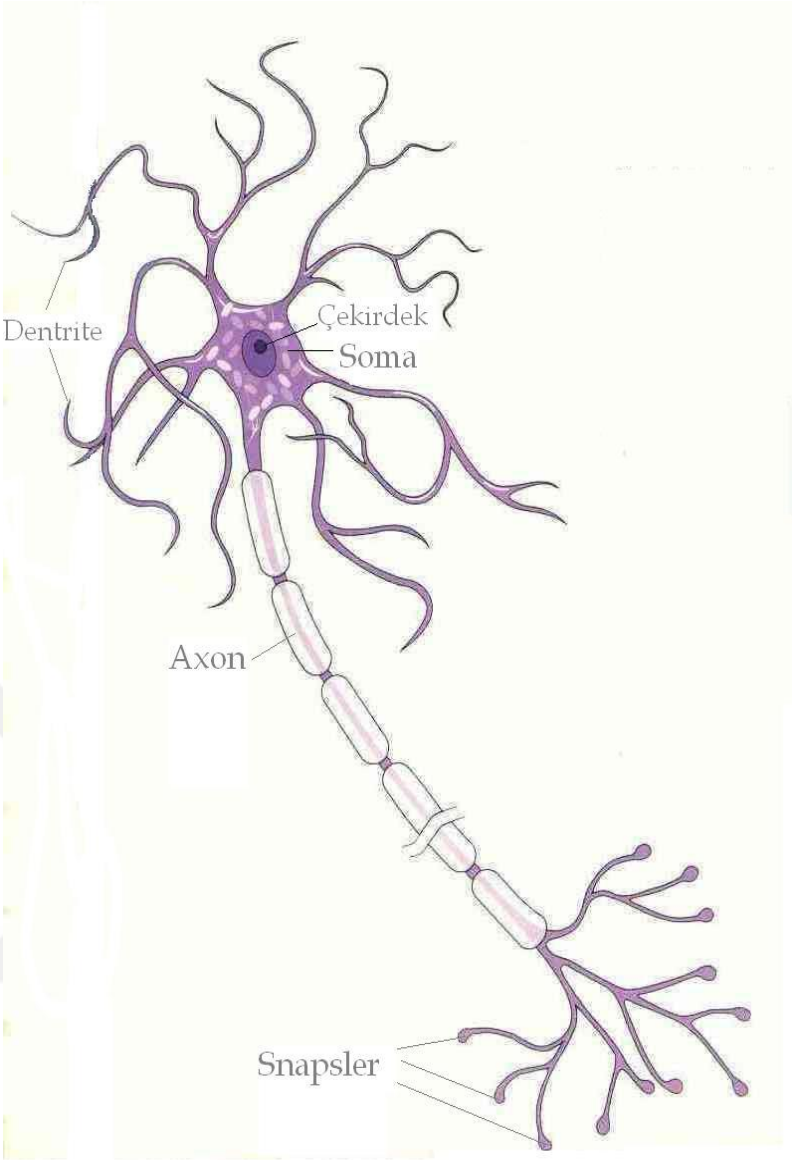
1.3. Tezin Yapısı

Tezin yapısı aşağıdaki gibidir. Bölüm 2, AÖM yönteminin temelini oluşturulan yapay öğrenme ve yapay sinir ağlarının tanımını kapsamaktadır. Doktora çalışmasında regresyon ve sınıflandırma problemleri için geliştirilen klasik AÖM, RTF-AÖM, Meta-AÖM ve ÇK-AÖM gibi incelenen mevcut AÖM yapıları Bölüm 3'te tanıtılmıştır. Bölüm 4'te dinamik sistem modelleme problemleri için iyileştirme yapılan ÇK-AÖM yapıları anlatılmaktadır. Engebeli ve keskin değişimli yüzeylere sahip sistem modelleme problem için geliştirilen karma ağ yapısı ve kullanılan yüzey modelleri Bölüm 5'te tanıtılmıştır. Bölüm 4 ve 5'te geliştirelen yapıların deney sonuçları Bölüm 6'da verilmiştir. Son olarak, Bölüm 7, çalışmanın sonuç ve önerlerini içermektedir.

2. YAPAY ÖĞRENME VE YAPAY SİNİR AĞLARI

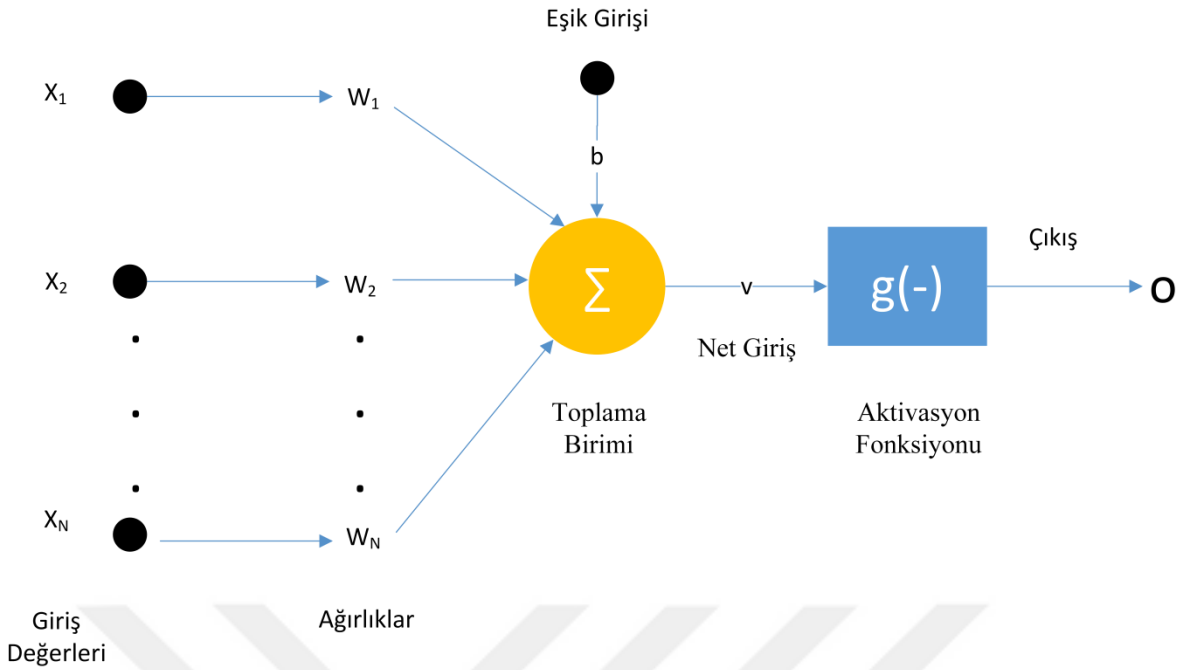
Yapay öğrenme günümüzde bir çok problemin çözümünde yaygın olarak kullanılan kavramdır. Yapay sinir ağları (YSA) bu kavram ile ortaya çıkmıştır. YSA, ilk olarak McCulloch ve Pitts (Mcculloch ve Pitts, 1943) tarafından önerilen sinirsel hesaplama sistemleridir. YSA'lar, kendi kendini organize etme özelliklerine ve paralel bilgi sistemlerine dayalı hesaplama tekniklerindeki önemli gelişmeler sayesinde 1980'lerden itibaren yaygın olarak kullanıldı. Rumelhart ve arkadaşları (Rumelhart vd., 1986), nöron benzeri birimlerin ağları için yeni bir öğrenme prosedürü, geri yayılımı (BP) önerdi. BP, ağın gerçek çıktı vektörü ile istenen çıktı vektörü arasındaki farkına dayalı olarak tanımlanan bir ölçütü en aza indirmek için ağdaki bağlantıların ağırlıklarını tekrar tekrar ayarlar. Bu çalışma, paralel dağıtılmış bilgi işleme çerçevelerinde BP kuralının geliştirilmesine bağlı olarak çeşitli araştırma alanlarında YSA'ların yaygın olarak kullanılmasına katkıda bulunmuştur. YSA'lar ayrıca 1980'lerin sonlarında makine zekasındaki karmaşık ve doğrusal olmayan fenomenleri yorumlamak için yaygın olarak kullanıldı.

YSA'ların gelişimi, insan beyninin karakteristik işleyişinden esinlenerek olmuştur, ancak bunlar biyolojik eşdeğerleri ile yalnızca uzaktan ilişkilidir. YSA'lar beynin karmaşıklığına yaklaşmazlar, ancak biyolojik sinir ağları ile YSA'lar arasında iki temel benzerlik vardır. İlk olarak, her iki ağın yapı taşları, yüksek oranda birbirine bağlı basit hesaplama cihazlarıdır. İkincisi, nöronlar arasındaki bağlantılar ağın işlevini belirler. YSA'lar paralel dağıtılmış bilgi işlem ağları olarak işlev görür ve bazı temel özelliklerde biyolojik sinir sistemlerine benzerdir. Şekil 2.1 ve 2.2'de sırasıyla, biyolojik sinir hücresinin ve YSAnın hücre yapısı verilmiştir. Sinir sistemi ile yapay sinir sisteminin karşılık gelen elemanları da Tablo 2.1'de verilmiştir.



Şekil 2.1. Biyolojik sinir hücre yapısı

Kaynak: (Tübitak Bilim ve Teknik Dergisi, 2000)



Şekil 2.2. YSA hücre yapısı

Tablo 2.1. Sinir sistemi ile yapay sinir sisteminin karşılık gelen elemanlar

Sinir Sistemi	Yapay Sinir Sistemi
Akson	Çıktı
Hücre Gövdesi	Toplama Birimi
Çekirdek	Aktivasyon Fonksiyonu
Sinaps	Ağırlıklar
Dentrit	Giriş

2.1. Yapay Sinir Ağlarının Özellikleri

YSA yapısı gereği öğrenebilme ve genelleme yeteneğine sahiptir. Genelleme yeteneği ile eğitim sürecinde karşılaşmadığı giriş değerleri için de uygun sonuç değerleri üretmektedir. Bu yeteneği ile karmaşık problemleri de çözebilme yeteneğine sahiptir. YSA, öğrenme, ilişkilendirme, sınıflandırma, genelleme, tahmin, özellik belirleme ve optimizasyon işlemlerini gerçekleştirebilir. Bu işlemlerde YSAnın temel katkısı dışarıdan müdahale gerekmezsizin

problem için verilmiş olan bilgiler doğrultusunda sonuç üretebilmesidir. YSA, sinir sisteminden esinlenerek geliştirildiği için bir çok üstünlüklere de sahiptir. Aşağıda bir kaç özelliklerinden kısaca bahsedilmiştir (Haykin S, 1999).

Öğrenebilirlik: Çoğu klasik algoritma, verilen formüller hesaplanırken aynı girdiler için her zaman aynı çıktıyı üretir. Bu doğrusal algoritmaların aksine YSA sayesinde programlar öğrenme yeteneğine sahip olmuşlardır. Çözümü yolu tanımlanamayan problemlerin çözümü hakkında YSA bilgi vermeksizin öğrenebilirlik yeteneği sayesinde problemi çözebilmektedir. Problemin çözümünde YSA'ya örnek girdi ve çıktı değerlerinin verilmesi gerekmektedir.

Genelleme: YSA üzerinde çalışılan problem için eğitildikten sonra eğitim esnasında karşılaşmadığı durum için de sonuç üretmektedir. Örnek: bir sandelye görüntüsü tanıtılıp gürültülü sandelye görüntüsü verildiğinde sandelye olarak tanıyacaktır. Bu da YSAnın genelleme yeteğine sahip olduğunu ifade etmektedir.

Uygulanabilirlik: YSA, çalıştığı probleme göre ağırlıklarını kendisi düzenleyerek belirler. Eğitilmiş YSA, her hangi bir problemde kullanılabilir. Sadece, yeni problemin giriş ve çıkış değerleri ağa verilerek ağ tekrar eğitilmelidir.

Tasarım Kolaylığı: YSAnın hücre yapısı ve modeli, tüm YSA modellerinde yaklaşık olarak aynıdır. YSA yapısı kolaylıkla oluşturulup çeşitli problemlerin çözümünde kullanılabilir.

Doğrusal Olmama: YSAnın temel elamanı olan hücre doğrusal sonuç vermemektedir. Bu özelliği tüm ağa yansımıştır. Bu yüzden doğrusal olmayan sistemlerde sonuç almak için yaygın olarak YSA kullanılmaktadır.

Paralellik: Klasik problem çözme algoritmalarının aksine YSA, parallel çalışmaya uygun yapıya sahip olduğundan sorunların çözümü için çok daha hızlı sonuç vermektedir.

Hata Toleransı: YSA parallelik özelliği ile bağımsız çalışabildiği için yapıdaki bir eleman arızalandığında öğrenebilme ve genelleme yeteneği sayesinde yapı sorunsuz çalışmaya devam edecektir.

YSAnın en temel eksiği kara kutu gibi çalışıyor olmasıdır. Bir sorunu çözmek için eğitilen ağın öğrenme ve genelleme yetisiyle çözülmesi imkansız sorunların üstesinden geldiği yadsınamaz bir gerçektir. Fakat bunu nasıl yaptığına dair bir bilinmezlik vardır. Bir de, YSAnın

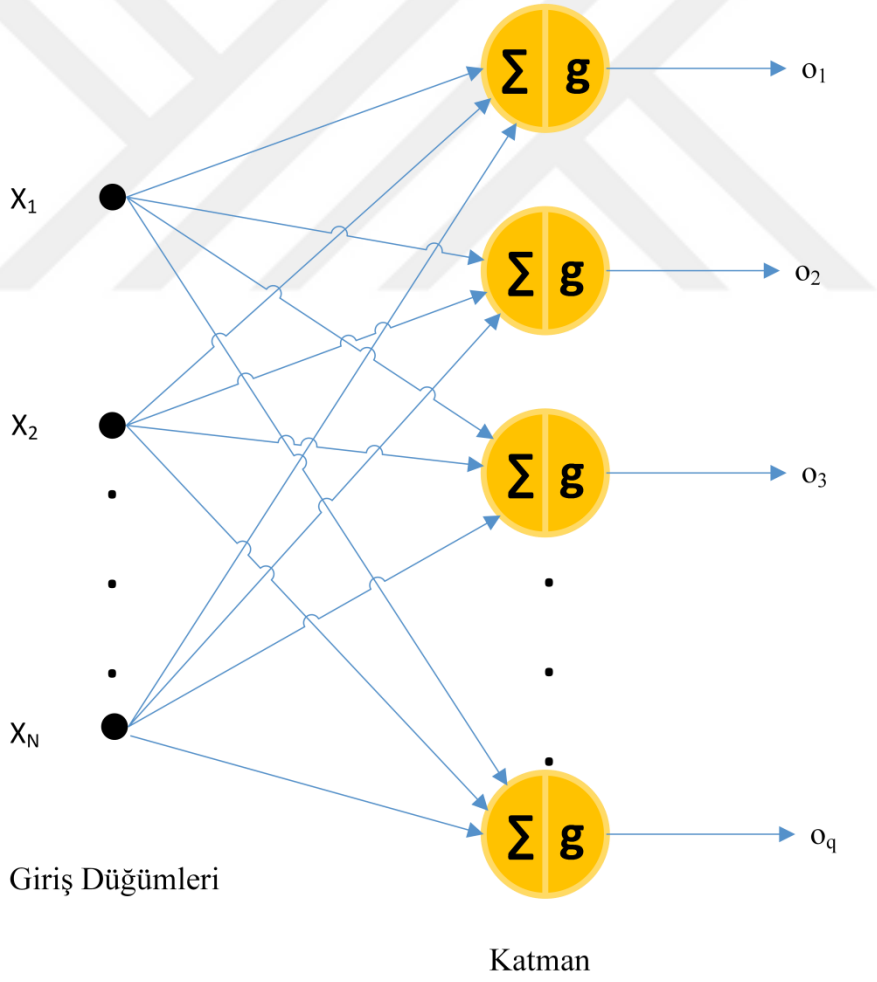
eğitimi esnasında problemin büyüklüğüne ve ağ yapısına bağlı olarak oldukça çok zaman harcanabilmektedir.

2.2. Yapay Sinir Ağ Modelleri

Yapay sinir ağ modelleri, tek katmanlı ve çok katmanlı algılayıcılar, ileri ve geri beslemeli yapay sinir ağları olmak üzere dört grupta incelenebilir.

2.2.1. Tek katmanlı yapay sinir ağları

Giriş ve çıkış düğümleri arasında tek katmanda sıralanmış nörünlardan oluşan bir yapıdır. Katmandaki hücrelere tüm giriş değerleri bağlanmaktadır ve bunların bağlantı ağırlıkları vardır. Eşik girişinin 1 olması çıkış değerinin sıfır olmasını engellemektedir. Şekil 2.3'te tek katmanlı ağ yapısı gösterilmiştir.



Şekil 2.3 Tek katmanlı YSA ağ yapısı

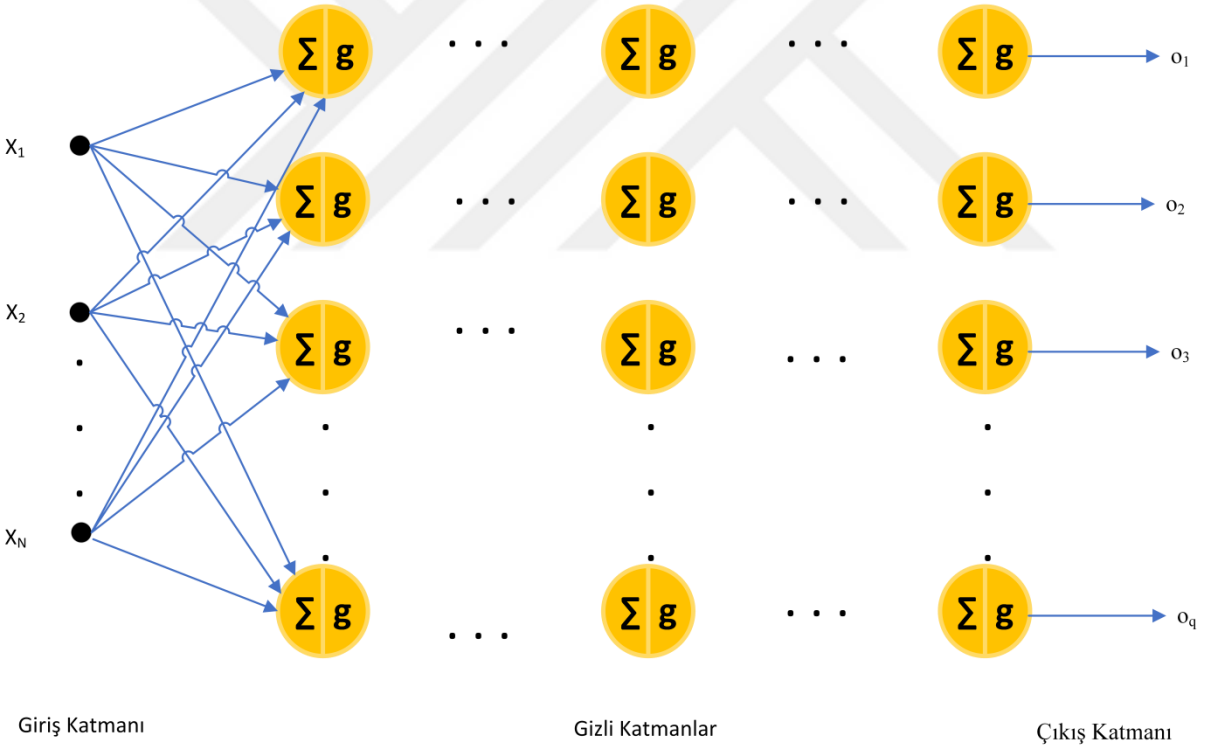
2.2.2. Çok katmanlı yapay sinir ağları

Çok katmanlı YSA giriş, gizli ve çıkış katmanından oluşmaktadır. Karmaşık ve güç doğrusal olmayan problemlerin çözümünde kullanılmaktadır. Giriş katmanındaki giriş birden fazla, gizli katmanda da birden fazla gizli katman ve hücreye sahip olabilmektedir. Gizli katman sayısı ve hücre sayısı problem göre belirlenmektedir. Şekil 2.4'de çok katmanlı bir YSAnın ağ yapısı verilmiştir.

Giriş Katmanı: Giriş bilgilerin alınıp gizli katmana aktarıldığı katmandır.

Gizli Katman: Giriş katmanındaki her bir giriş verisini gizli katmandaki her bir hücreye bağlar ve bilgiler işlenir.

Çıkış Katmanı: Gizli katmanın iletmiş olduğu veriyi işleyerek sonuç elde edilmektedir.



Şekil 2.4. Çok katmanlı YSAnın ağ yapısı

2.3. Kullanım Alanları

YSA yaygın olarak, sınıfladırma, tahmin ve modelleme gibi bir çok alanda kullanılmaktadır. Ayrıca YSA birçok problemle uğraşırken, bilgi kaynakları ne eksiksiz ne de yanıltıcıdır. Karar kuralları bazen çelişkili bazen de yoktur. Bu, geleneksel bilgi işleme

yöntemleri için büyük zorluklar yaratır, ancak sinir ağı bu problemlerle çok iyi başa çıkabilir ve makul bir tanımlama ve yargılar verebilir (Wu ve Feng, 2018). Aşağıda YSAnın kullanım alanlar kısaca verilmeye çalışmıştır.

Bilgi İşleme: Modern bilgi işleme ile çözülmesi gereken sorunlar çok karmaşıktır. Yapay sinir ağları, insanların düşüncelerini taklit etme veya değiştirme işlevine sahiptir ve otomatik teşhis, problem çözme ve geleneksel yöntemlerle çözülemeyen problemlerin çözülmesini gerçekleştirebilir. Mevcut akıllı bilgi sistemleri arasında akıllı enstrümanlar, otomatik izleme ve izleme enstrümantasyon sistemleri, otomatik kontrol yönlendirme sistemleri, otomatik arıza teşhis ve alarm sistemleri yer almaktadır.

Örüntü Tanıma: Örüntü tanıma, nesneleri veya fenomenleri karakterize eden çeşitli bilgi biçimlerini işleyerek ve analiz ederek şeyleri veya fenomenleri tanımlama, sınıflandırma ve yorumlama sürecidir. Artık iki temel örüntü tanıma yöntemi vardır, istatistiksel örüntü tanıma ve yapısal örüntü tanıma. Yapay sinir ağları, örüntü tanımada yaygın olarak kullanılan bir yöntemdir. Son yıllarda, geleneksel örüntü tanıma yönteminin yerini yavaş yavaş yapay sinir ağı örüntü tanıma yöntemi almıştır. Yıllarca süren araştırma ve geliştirmeden sonra, örüntü tanıma günümüzün daha ileri teknolojisi haline geldi ve karakter tanıma, konuşma tanıma, parmak izi tanıma, uzaktan algılama görüntü tanıma, yüz tanıma, el yazısı karakterleri tanıma, endüstriyel arıza algılama, biyolojik sinyallerin tespiri ve otomatik analizi, tıbbi uzman sistem gibi bir çok amaç için yaygın olarak uygulandı.

Biyolojik Sinyal Tespiti ve Analizi: Çoğu tıbbi test ekipmanı, verileri teşhisin temeli olan sürekli bir dalga biçiminde çıkış verir. Yapay sinir ağı, çok sayıda basit işlem birimiyle birbirine bağlanan bir tür uyarlanabilir dinamik sistemdir. Geleneksel yasa ile çözülemeyen biyomedikal sinyal analizi ve işleme problemlerini çözmek için kullanılabilecek büyük miktarda paralellik, dağıtılmış depolama ve kendi kendine uyarlamalı öğrenme işlevlerine sahiptir. Biyomedikal sinyal algılama ve işlemede sinir ağının uygulanması temel olarak EEG sinyalinin analizine, işitsel uyarılmış potansiyel sinyalin çıkarılmasına, EMG ve gastrointestinal sinyallerin tanımlanmasına, EKG sinyallerinin sıkıştırılmasına, tıbbi görüntülerin tanınmasına vb. alanlardır.

Tıbbı uzman sistemi: Geleneksel uzman sistem, uzmanların deneyim ve bilgilerini kurallar şeklinde bilgisayarda depolamak, bilgi tabanı oluşturmak ve tıbbi teşhis için mantıksal akıl yürütme yolunu kullanmaktır. Ancak pratik uygulamalarda veri tabanının boyutu arttıkça bilginin “patlamasına” ve bilgiye erişimde “darboğaz”a yol açarak verim düşüklüğüne neden

olacaktır. Doğrusal olmayan paralel işlemeye dayalı sinir ağı, uzman sistem araştırması için yeni bir gelişme yönüne işaret eder, yukarıdaki uzman sistem problemlerini çözer ve bilgi çıkarımını, kendi kendine organizasyonu ve kendi kendine öğrenme yeteneğini geliştirir, böylece sinir ağı yaygın olarak kullanılır.

Piyasa fiyatı tahmini: Emtia fiyatlarındaki değişikliklerin analizi, piyasadaki arz talep ilişkisini etkileyen birçok faktörün kapsamlı bir analizine atfedilebilir. Doğal sınırlamaları nedeniyle, geleneksel istatistiksel iktisat yönteminin fiyat değişikliklerini bilimsel olarak tahmin etmesi zordur. Bununla birlikte, yapay sinir ağının eksik, bulanık belirsiz veya düzenli verilerle başa çıkması kolaydır. Bu nedenle yapay sinir ağı fiyat tahmini, geleneksel yöntemlerle karşılaştırılamayacak kadar iyidir. Piyasa fiyatının belirlenmesi mekanizmasından yola çıkılarak emtia fiyatlarından etkilenen hane sayısı, kişi başına harcanabilir gelir, kredi faiz oranı ve şehirleşme düzeyi gibi karmaşık ve sürekli değişen faktörler baz alınarak daha doğru ve güvenilir bir model oluşturulmaktadır. Model, emtia fiyatlarının değişen trendini tahmin edebilir ve doğru ve objektif değerlendirme sonuçları elde edebilir.

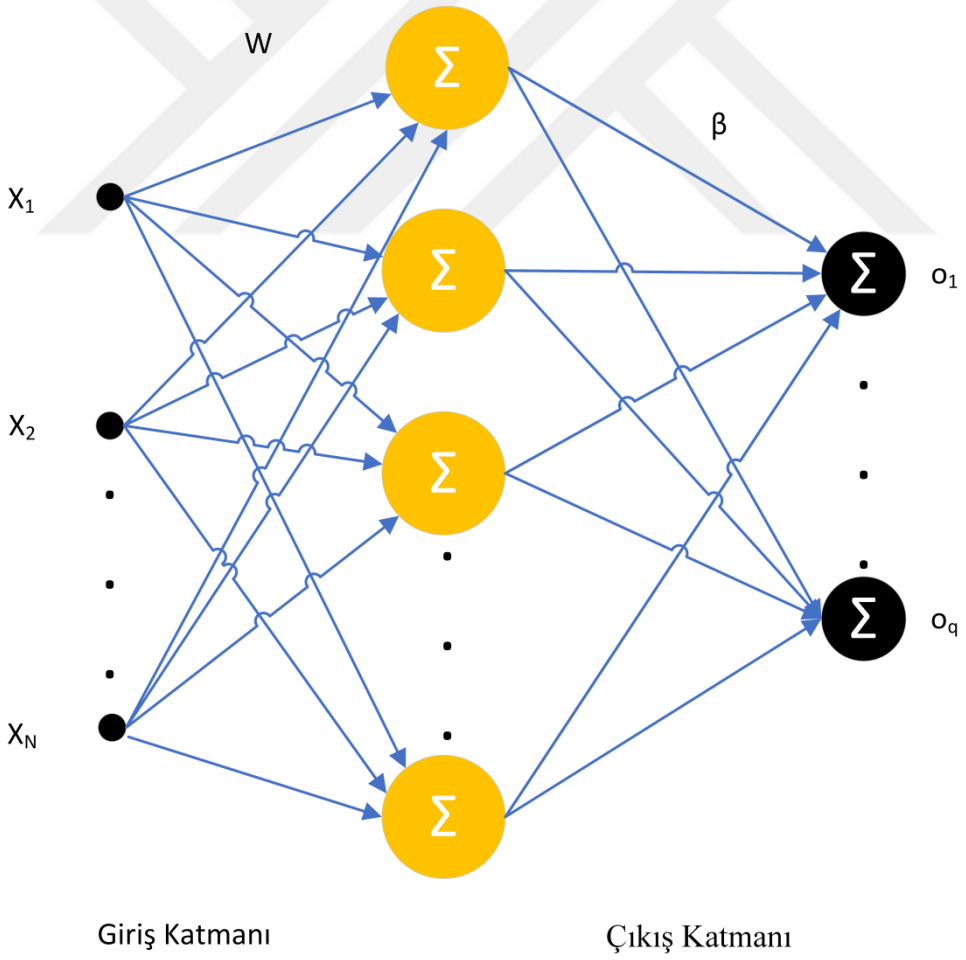
Risk değerlendirmesi: Risk, belirli bir faaliyette bulunma sürecindeki belirsizliklerin neden olduğu doğal veya finansal zarar veya hasar olasılığını ifade eder. Bir riski önlemenin en iyi yolu, riski önceden bilimsel bir tahmin ve değerlendirme yapmaktır. Yapay sinir ağı uygulamasının öngörüsü, gerçek risk kaynağına göre fiili duruma uygun kredi riski modelinin yapısını ve algoritmasını oluşturmak, risk değerlendirme katsayısını almak ve ardından asıl problemin çözümünü belirlemektir. Bu modeli kullanan ampirik analiz, öznel değerlendirme eksikliğini giderebilir, tatmin edici sonuçlar elde edebilir.

3. AŞIRI HIZLI ÖĞRENME MAKİNASI

Aşırı hızlı öğrenme makinası (AÖM) tek gizli katmanlı yapay sinir ağlarını eğitmek için yeni analitik öğrenme algoritması olarak önerilmiştir (G. bin Huang vd., 2004; G. bin Huang vd., 2006). Sinir ağlarında yaygın olarak kullanılan türev tabanlı öğrenme yöntemlerinde yüksek öğrenme başarımı elde etmek için çok sayıda eğitim/test verisi ve tur sayısı kullanıldığından AÖM yöntemine göre eğitim seyri çok yavaştır. AÖM'de giriş katmanındaki ağırlık parametreleri rastgele atanır ve çıkış katmanındaki ağırlık parametreleri analitik olarak belirlenir. Bu yüzden AÖM ile öğrenme hızlı ve başarımı yüksektir.

3.1. Aşırı Hızlı Öğrenme Makinası

Aşırı öğrenme makinesi (AÖM), tek gizli katmanlı yapay sinir ağları (SLFN)'i baz alan bir modeldir (G. bin Huang vd., 2004; G. bin Huang vd., 2006). Şekil 3.1'de AÖM'nin ağ yapısı verilmiştir.



Şekil 3.1. AÖM'nin ağ yapısı

Şekil 3.1'de görüldüğü üzere AÖM'nin yapısı giriş, gizli ve çıkış katmanından oluşmaktadır. AÖM'nin gizli katman parametreleri (w), rastgele olarak atanır ve ileri eğitim aşamalarında güncellenmez. Öte yandan, çıkışla gizli katman arasındaki ağırlık parametreleri (β) bir doğrusal model ile analitik ve hızlı bir şekilde belirlenir. $\mathbf{x}_j = [x_{j1}, ..., x_{jN}]^T$ ve $\mathbf{o}_j = [o_{j1}, ..., o_{jq}]^T$ olarak P ayrı örnek ($\mathbf{x}_j$, $\mathbf{o}_j$) için ELM'in matematiksel yapısı Eşitlik 3.1'de gösterilmektedir.

$$\sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \mathbf{x}_j + b_i) = o_j, j = 1, \dots, P \tag{3.1}$$

Bu modelde i = 1, ..., L ve j = 1, ..., P için β_i çıkış ağırlıkları, $g(.)$ nöron aktivasyon fonksiyonu, $\mathbf{w}_i$ giriş ağırlıkları ve $\mathbf{x}_j$ giriş değerleridir. Burada L düğüm veya nöron sayısıdır. Geleneksel SLFN'de β_i ağırlıkları bir döngü içinde güncellenerek tespit edilir. Ancak, Huang ve çalışma arkadaşları bu ağırlık parametrelerini bir kerede hesaplayan ELM yapısını keşfetmiştir (G. bin Huang, Zhu ve Siew, 2004c).

$L \geq P$ durumu için, P adet örnek sıfır hata ile tahmin edebilir, başka bir deyişle o_j eğitim veri hedefi t_j'e değerine yakınsatılmıştır. Bu durumda Eşitlik 3.2, β_i, $\mathbf{w}_i$, $\mathbf{x}_j$ ve b_i için ispatlanmıştır.

$$\sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \mathbf{x}_j + b_i) = \boldsymbol{t_j}, j = 1, \dots, P \tag{3.2}$$

Eşitlik 3.2'de $\mathbf{t}_j$ eğitim veri hedef matrisi **T**'nin elemanlarıdır. Eşitlik 3.2'de tanıtılan P adet denklem Eşitlik 3.3'teki gibi özetlenebilir.

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{3.3}$$

Eşitlik 3.3'te

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \mathbf{x}_P + b_1) & \cdots & g(\mathbf{w}_L \mathbf{x}_P + b_L) \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_P \end{bmatrix} \tag{3.4}$$

Eşitlik 3.4 ele alındığında **H**, gizli katmanın çıkış matrisidir. **H** matrisinin i numaralı sütunu da i numaralı gizli düğümün çıktısını temsil eder. Vektör **t** ise, eğitim verisi hedef vektörüdür.

Daha önce gösterildiği üzere, $L > P$ için sıfır hata sunan birçok çözüm vardır. Ancak $L < P$ için aşağıdaki ölçüt fonksiyonunu minimize edecek bir çözüm bulunmalıdır.

$$C = \sum_{j=1}^{P} \left(\sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \mathbf{x}_j + b_i) - t_j \right)^2 \tag{3.5}$$

Ölçüt fonksiyonu minimize edildikten sonra, **H** matrisinin Moore-Penrose tersi $\mathbf{H}^{\dagger}$ tespit edildir ve AÖM, $\widehat{\boldsymbol{\beta}}$'yı Eşitlik 3.6'da gösterildiği gibi öğrenir.

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{3.6}$$

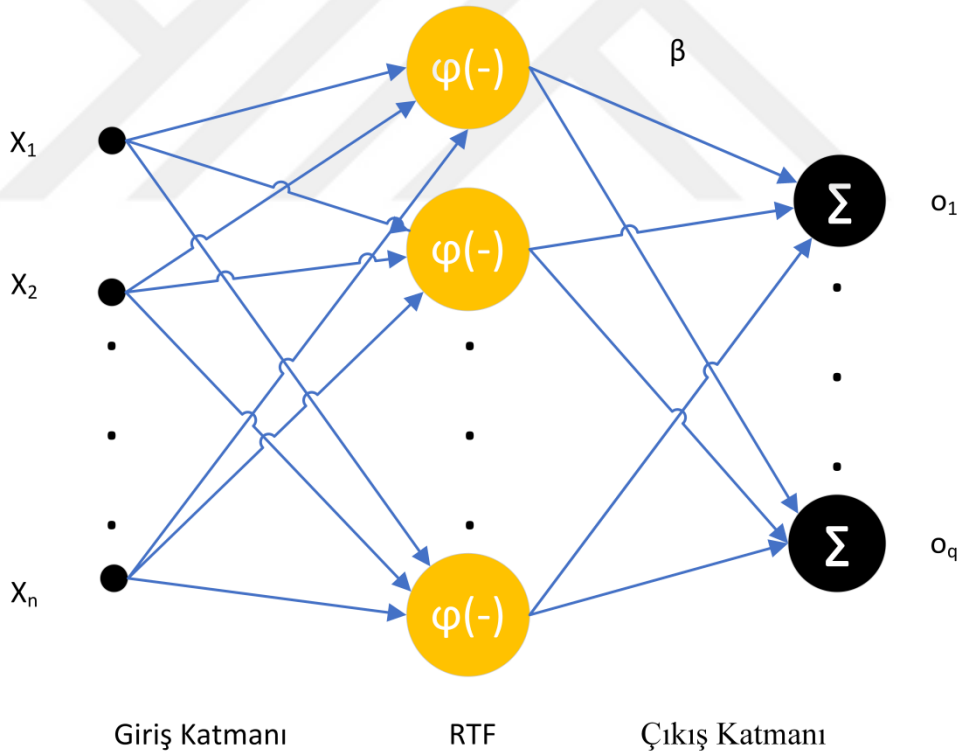
Çıkış matrisi **O** Eşitlik 3.7'de gösterildiği gibi hesaplanır.

$$\mathbf{O} = \mathbf{H}\boldsymbol{\beta} \tag{3.7}$$

3.2. Radyal Tabanlı Fonksiyon Kullanan Aşırı Öğrenme Makinesi

Huang ve arkadaşları tarafından 2004 yılında yapılan çalışmada, AÖM yapısını radyal tabanlı fonksiyon (RTF) ağ yapısına göre uyarlanmıştır. RTF'deki merkez ve varyans değerleri rastgele atanır ardından çıkış ağırlık parametreleri iteratif hesaplamak yerine AÖM yapısındaki anatilik hesaplama yöntemi ile hesaplanmaktadır (G. bin Huang ve Slew, 2004).

Şekil 3.2'de RTF-AÖM'nin ağ yapısı verilmiştir. Ağ yapısının matematiksel ifadeleri aşağıda verilmiştir ve Algoritma 1'de de algoritması yer almaktadır.



Şekil 3.2. RTF-AÖM'nin ağ yapısı

$\widetilde{N}$ çekirdeğe sahip bir RTF ağının $\mathbf{x} \in \mathbf{R}^n$ girdi vektörüne çıktısı Eşitlik 3.8'de gösterildiği gibidir.AÖM$_1$

$$f_{\tilde{N}}(\mathbf{x}) = \sum_{i=1}^{\tilde{N}} (\beta_i \varphi_i(\mathbf{x})) \tag{3.8}$$

Bu denklemde $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$, i çekirdeğini çıkış nöronuna bağlayan ağırlık vektörüdür. $\varphi_i(\mathbf{x})$ ise i çekirdeğinin çıkışıdır ve genellikle bir Gauss dağılımıdır.

$$\varphi_i(\mathbf{x}) = \varphi(\mu_i, \sigma_i, \mathbf{x}) = \exp\left(\frac{\|\mathbf{x}-\mu_i\|^2}{\sigma_i}\right) \tag{3.9}$$

Burada, $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$ i çekirdeğinin merkezi ve σ_i ise etki genişliğidir.

$\mathbf{x_i} = [x_{i1}, x_{i2,\dots,} x_{in}]^T \in \mathbf{R}^n$ ve $\mathbf{t}_i = [t_{i1}, t_{i2,\dots,} t_{im}]^T \in \mathbf{R}^m$ iken, P keyfi farklı örnek $(\mathbf{x_i}, \mathbf{t}_i)$ için $\tilde{N}$ çekirdeğe sahip RTF matematiksel olarak

$$\sum_{i=1}^{\tilde{N}} \beta_i \varphi_i(\mathbf{x}_j) = \mathbf{o}_j, j = 1, \dots, P \tag{3.10}$$

şeklinde tanımlanır. $\tilde{N}$ çekirdeğe sahip standart RTF, bu P örneği sıfır hata ortalaması ile yakınsatır ($\sum_{i=1}^{\tilde{N}} \|\mathbf{o}_j - \mathbf{t}_j\| = 0$). Kısaca varolan β_i, μ_i ve σ_i için

$$\sum_{i=1}^{\tilde{N}} \beta_i \exp\left(\frac{\|\mathbf{x}_j-\mu_i\|^2}{\sigma_i}\right) = \mathbf{t}_j, j = 1, \dots, P \tag{3.11}$$

Yukarıdaki denklemler matrisler ile aşağıdaki gibi yazılabilir.

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{3.12}$$

Burada

$$\mathbf{H}(\mu_1, \dots, \mu_{\tilde{N}}, \sigma_1, \dots, \sigma_{\tilde{N}}, \mathbf{x}_1, \dots, \mathbf{x}_n) = \begin{bmatrix} \varphi(\mu_1, \sigma_1, \mathbf{x}_1) & \cdots & \varphi(\mu_{\tilde{N}}, \sigma_{\tilde{N}}, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi(\mu_1, \sigma_1, \mathbf{x}_n) & \cdots & \varphi(\mu_{\tilde{N}}, \sigma_{\tilde{N}}, \mathbf{x}_n) \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_P^T \end{bmatrix} \tag{3.13}$$

H gizli katman çıkış matrisidir. En küçük kareler çözümü olan $\hat{\beta}$ ise aşağıdaki gibi hesaplanır.

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{3.14}$$

Algoritma 1: RTF - AÖM Algoritması

Başlangıç:

Bir eğitim kümesi Ꭻ = {($\mathbf{x}_i$,t_i) | $\mathbf{x}_i$ Є R^n , $\mathbf{t}_i$ Є R^m , i=1,...,N} ve çekirdek sayısı $\tilde{\mathbf{N}}$ için:

Adım 1:

Keyfi merkezler $\boldsymbol{\mu}_i$ ve etki genişlikleri $\boldsymbol{\sigma}_i$ ataması yap. (i = 1, ... , $\tilde{\mathbf{N}}$)

Adım 2:

Gizli (çekirdek) katmanı çıkış matrisi **H**'yi hesapla.

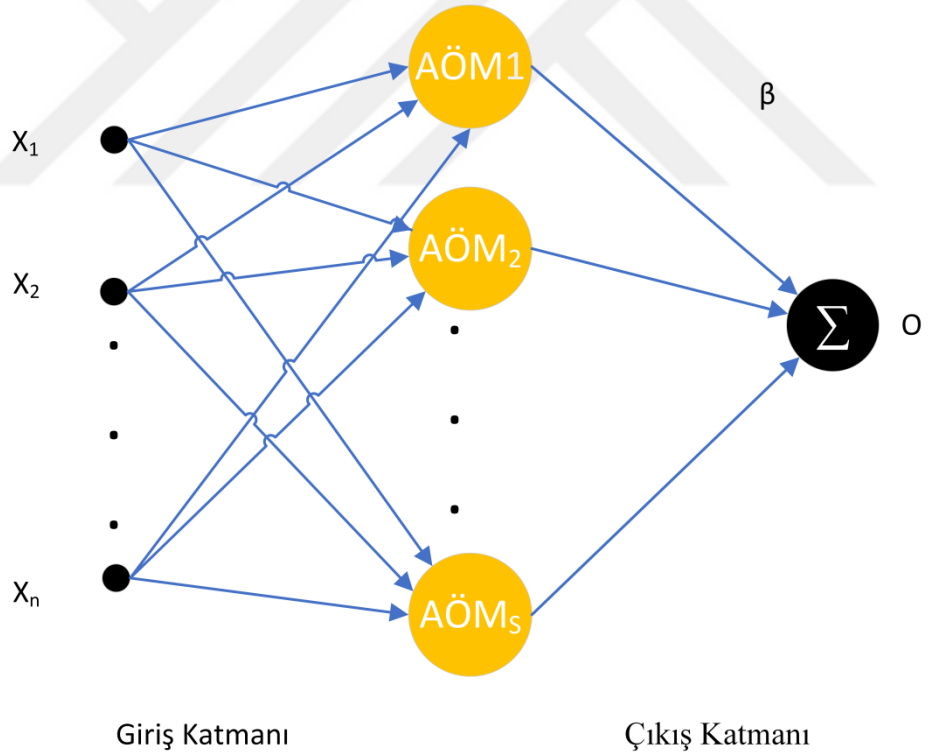
Adım 3:

Çıkış ağırlığı **β**'yı $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$ formülünü kullanarak hesapla.

3.3. Meta Aşırı Öğrenme Makinesi (Meta-AÖM)

Meta Aşırı Öğrenme Makinesi (Meta-AÖM) basit olarak öğrencilerden öğrenen bir AÖM modelidir. Şekil 3.3'de görüldüğü gibi birkaç taban AÖM ve bu taban AÖM'lerden öğrenen bir meta öğreniciden oluşur (Liao ve Feng, 2014). Her taban AÖM bir taban tahmin edici üretir ve meta öğrenici de meta tahmin edici üretir.

Şekil 3.3'de Meta-AÖM'nin basit bir ifade ile ağ yapısı verilmiştir.



Şekil 3.3. Meta-AÖM'nin ağ yapısı

TKYSA açısından bakıldığında, Meta-AÖM AÖM ağını Şekil 3.3'de gösterilen şekilde AÖM gizli katmanlarıyla eğitir. Meta-AÖM, veri kümesinin ayrık alt kümelerinde taban AÖM' leri ve hiyerarşik mimarinin oluştuğu tüm veri kümesinde bir "üst" AÖM' yi eğitir.

Meta-AÖM hiyerarşik bir öğrenme modelidir. Eğitim veri kümesini rastgele yeniden karıştırdıktan sonra, Meta-AÖM tüm veri kümesini ayrık alt kümelere böler, alt kümeler üzerinde tahminciler oluşturur ve AÖM' nin yaptığı gibi tahmin edicilerin ağırlıklarını analitik olarak belirler.

Bu model, aşağıdaki ölçüt fonksiyonunu minimize etmelidir:

$$C = \sum_{i=1}^{P}[\sum_{s=1}^{S} \beta_s(\mathbf{x}_i) A\ddot{O}M_s(\mathbf{x}_i) - t_i]^2 \quad (3.15)$$

Burada, *P* örnek sayısı, *S* ise taban AÖM sayısına karşılık gelir. $A\ddot{O}M_s(\mathbf{x}_i)$, *s.* sayılı taban AÖM'nin $\mathbf{x}_i$ girdisi için çıkışıdır. $\beta_s(\mathbf{x}_i)$ ise, $\mathbf{x}_i$ girdisi için *s.* sayılı taban AÖM için ağırlık parametresidir.

Bu yöntem için gizli katman matrisi **H** aşağıda gösterilmektedir.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_P) \end{bmatrix} = \begin{bmatrix} A\ddot{O}M_1(\mathbf{x}_1) & \cdots & A\ddot{O}M_S(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ A\ddot{O}M_1(\mathbf{x}_P) & \cdots & A\ddot{O}M_S(\mathbf{x}_P) \end{bmatrix} \quad (3.16)$$

Bu modelde de ağırlık parametreleri aşağıdaki denklem yardımıyla bulunur.

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \quad (3.17)$$

3.4. Çok Katmanlı Aşırı Öğrenme Makinesi

Bu bölümde ÇK-AÖM anlatılmaktadır. ÇK-AÖM yapısında tek katmanlı AÖM yerine çok katmanlı AÖM katmanları vardır. Bu mimaride, son katman haricinde tüm katmanlar otomatik kodlayıcılardır (AE). $\mathbf{HH}^{\mathrm{T}}$ tekil olmadığında **H** matrisinin Moore-Penrose tersi $\mathbf{H}^{\mathrm{T}}(\mathbf{HH}^{\mathrm{T}})^{-1}$ olarak da gösterilebilir. Sırt regresyonu teorisi kullanıldığında, $\mathbf{HH}^{\mathrm{T}}$ matrisinin diyagonal elemanlarına (1/λ) değerleri eklenir. Sonuç olarak, bir katman için, ÇK-AÖM yapısının öğrenme denklemi Eşitlik 3.18'deki gibi olur.

$$\boldsymbol{\beta} = \mathbf{H}^{T}\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^{T}\right)^{-1}\mathbf{T} \quad (3.18)$$

ÇK-AÖM yapısında her gizli katman için temsili girişler bulunmaktadır. Katman *k* için temsili giriş *N* adet harici giriş kullanıldığında $\mathbf{X}^{(k)} = \left[\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_N^{(k)}\right]$ olarak gösterilebilir. Katman *k* için $\boldsymbol{\beta}^{(k)} = \left[\boldsymbol{\beta}_1^{(k)}, \ldots, \boldsymbol{\beta}_N^{(k)}\right]$ transformasyon matrisidir. Bu şekilde temsili giriş hesaplaması Eşitlik 3.19'da gösterilmiştir.

$$\mathbf{X}^{(k)} = \mathbf{H}^{(k)}\boldsymbol{\beta}^{(k)} \quad (3.19)$$

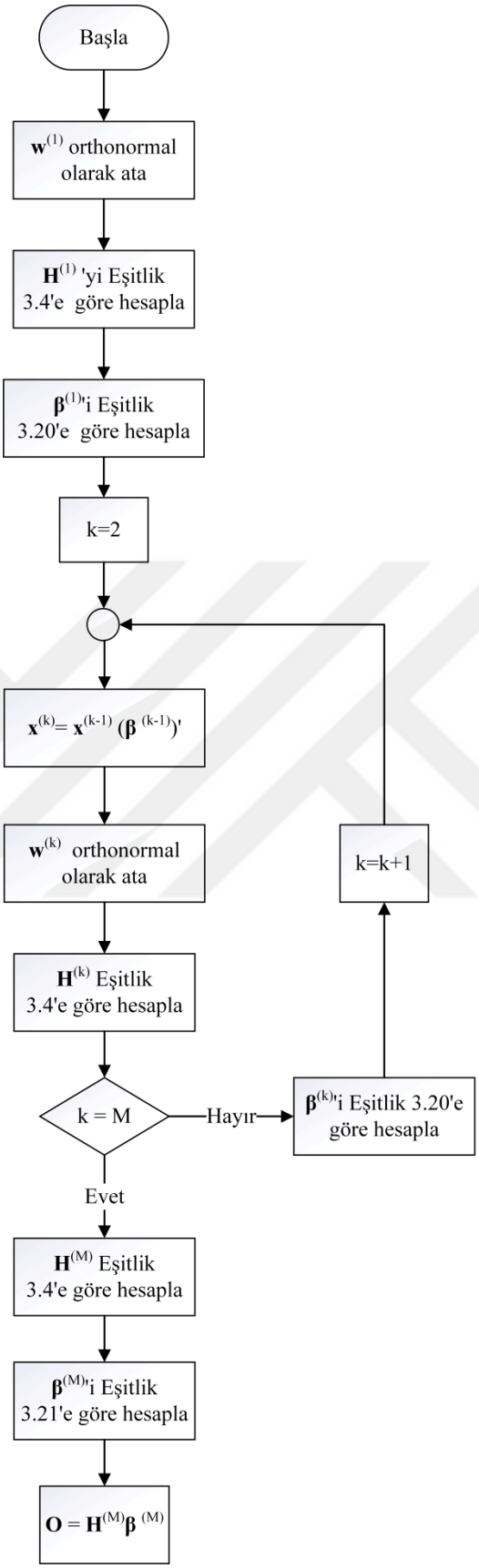
Eşitlik 3.19 kullanılarak $\boldsymbol{\beta}^{(k)}$ eşitlik 3.20'deki gibi öğrenilir.

$$\boldsymbol{\beta}^{(k)} = \left(\mathbf{H}^{(k)}\right)^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^{(k)}\left(\mathbf{H}^{(k)}\right)^T\right)^{-1} \mathbf{X}^{(k)} \quad (3.20)$$

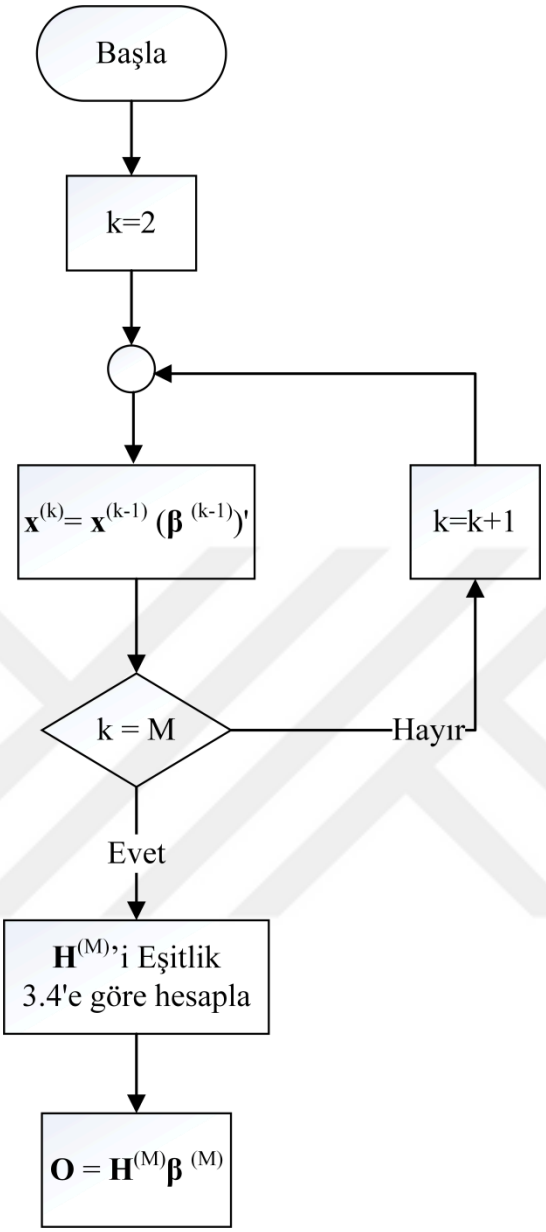
M katman için son katmanın çıkış matrisi $\mathbf{H}^{(M)}$ olarak adlandırılır. Bu çıkış matrisini kullanarak son katmanın çıkış ağırlık matrisi $\boldsymbol{\beta}^{(M)}$ Eşitlik 3.21'de gösterilen şekilde öğrenilir.

$$\boldsymbol{\beta}^{(M)} = \left(\mathbf{H}^{(M)}\right)^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^{(M)}\left(\mathbf{H}^{(M)}\right)^T\right)^{-1} \mathbf{T} \quad (3.21)$$

ÇK-AÖM'nin eğitim ve test olmak üzere iki aşaması vardır. ÇK-AÖM eğitim ve test işlemlerinin akış şeması Şekil 3.4-3.5'de gösterilmektedir. Eğitim ve test yapılarının farklı olduğuna dikkat edin. ÇK-AÖM yapısında eğitim aşamasında öğrenilen β değerlerinin test aşamasında kullanılabilmesi için saklanması gerekir. Son olarak, test aşamasında eğitim aşamasının son katmanının ağırlık parametreleri kullanılır.



Şekil 3.4. M gizli katman için ÇK-AÖM eğitiminin akış şeması



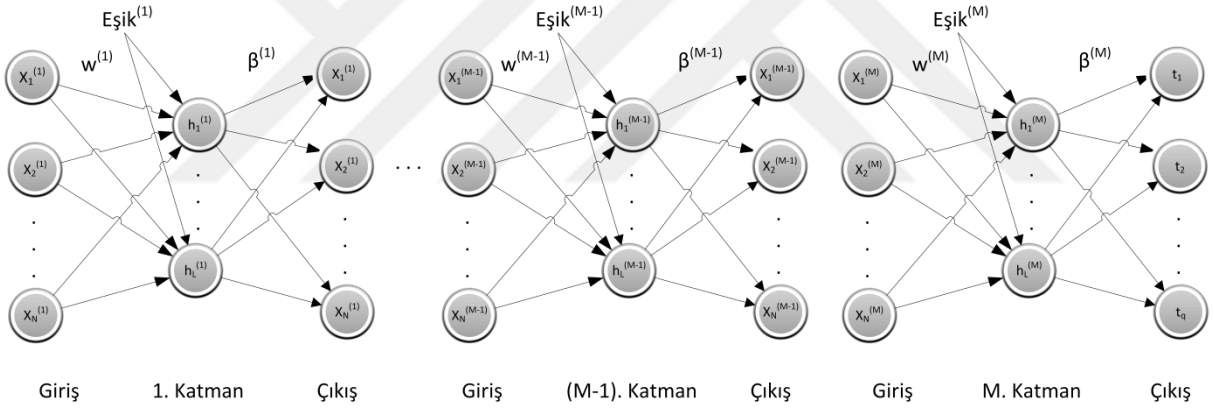
Şekil 3.5. M gizli katman için ÇK-AÖM testinin akış şeması

4. ÇOK KATMANLI AŞIRI ÖĞRENME MAKİNESİ YAPILARI İLE DİNAMİK SİSTEM MODELLEME

Bu bölümde ÇK-AÖM yapısının sistem modelleme başarımını arttırmayı ve hesaplama zamanını düşürmeyi amaçlayan iki İÇK-AÖM yapısı tanıtılacaktır. Ayrıca bu yapıları geliştirirken bir diğer motivasyonumuz da eğitim ve test aşamaları için aynı yapıyı kullanmaktır. İlk olarak geliştirilmiş birinci yapı ardından da ikinci yapı şekiller ve algoritma ile anlatılmıştır.

4.1 İyileştirilmiş Çok Katmanlı Aşırı Öğrenme Makinesi Yapıları

Geliştirilmiş olan ilk İÇK-AÖM (İÇK-AÖM1) yapısında bir katmanın temsili girişi bir önceki katmanın çıkış matrisi (**H**)'nin yine bir önceki katmanın çıkış ağırlık parametrelerinin çarpımı ile bulunmaktadır. ÇK-AÖM eğitim ve test aşamaları için iki ayrı yapı kullanırken İÇK-AÖM1 aynı yapıyı eğitim ve test için kullanmaktadır. İÇK-AÖM1'in eğitim ve test aşamaları için kullandığı yapı Şekil 4.1'de gösterilmektedir.



Şekil 4.1. M adet gizli katman için İÇK-AÖM1 yapısı

İÇK-AÖM1'in ÇK-AÖM'e göre katkısı, İÇK-AÖM1'in eğitim ve test için Şekil 4.1'de görülen aynı yapıyı kullanmasıdır. İÇK-AÖM1'de her katman için ağırlık parametreleri (**w**) ve eşikler rastgele atanmaktadır. Bu durum da sistemdeki rastgeleliği arttırmaktadır. Sistemin öğrenebilirliği arttırmak için ikinci bir İÇK-AÖM (İÇK-AÖM2) yapısı tasarlanmıştır.

Her katmanın temsili giriş hesaplama yolu ve aynı yapının eğitim ve test aşamaları için kullanması nedeniyle İÇK-AÖM1 hesaplama zamanını ÇK-AÖM'e göre düşürmektedir ve yapı daha basitleşmiştir. İÇK-AÖM1'in hesaplama işlemleri Algoritma 2'de gösterilmektedir.

Algoritma 2: İÇK-AÖM1 Yapısının Algoritması

Başlangıç;

İlk katman için, ağırlık parametreleri $\mathbf{w}^{(1)}$ ve $Eşik^{(1)}$ rastgele atanır

$\mathbf{H}^{(1)}$ Eşitlik 3.4'e göre hesaplanır.

Çıkış ağırlık parametreleri $\boldsymbol{\beta}^{(1)}$ Eşitlik 3.20'a göre öğrenilir.

for k=2:M **do**

Mevcut katmanın temsili girişi bir önceki katmanın çıkış matrisi $\mathbf{H}^{(k-1)}$ ile bir önceki katmanın çıkış ağırlık parametrelerinin $\boldsymbol{\beta}^{(k-1)}$ çarpımı ile bulunur.

Mevcut katman için $\mathbf{w}^{(k)}$ ve $Eşik^{(k)}$ rastgele atanır.

$\mathbf{H}^{(k)}$ Eşitlik 3.4'e göre hesaplanır.

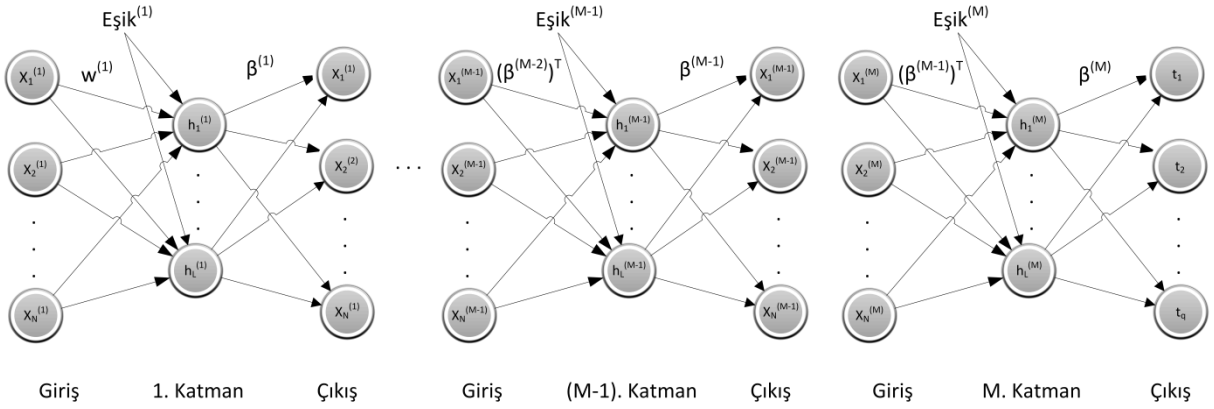
Çıkış ağırlık parametreleri $\boldsymbol{\beta}^{(k)}$ Eşitlik 3.20'e (k=M ise Eşitlik 3.21) göre öğrenilir.

End

Test aşaması için eğitim aşamasında kullanılan yapı kullanılmaktadır ve eğitim aşamasında kullanılan/öğrenilen giriş ağırlık parametreleri **w**, eşikler ve çıkış ağırlık parametreleri **β** kullanılmaktadır.

Kısaca, öğrenme için ÇK-AÖM'e benzer bir yapı kullanılırken, test için ÇK-AÖM'den farklı bir yapı kullanılmaktadır. İÇK-AÖM1 öğrenmeden sonra da hücresel düğümler kullanmaktadır. ÇK-AÖM ise öğrenme aşamasında oluşturulmuş bir dönüşüm matrisi kullanmaktadır.

İÇK-AÖM2 yapısı ise, temel olarak İÇK-AÖM1'i kullanmaktadır. İÇK-AÖM2, İÇK-AÖM1 gibi öğrenme ve öğrenmeden sonra kullanım için aynı yapıyı kullanmaktadır. İÇK-AÖM2 yapısında ilk katmanda giriş bağlantı ağırlıkları birbirlerine ortonormal olma koşuluyla rastgele atanmaktadır. Eşik parametreleri de (-1,1) aralığında orthonormal olarak rastgele atanmaktadır. Sonraki katmanların ağırlık parametre matrisi bir önceki katmanın çıkış ağırlık parametre matrisinin transpozesidir. Eşik parametreleri de ilk katmandaki şekilde atanmaktadır. Bu şekilde hesaplama zamanı çok daha fazla azaltılmıştır. Ağın yapısı Şekil 4.2'de gösterilmektedir.



Şekil 4.2. M adet gizli katman için İÇK-AÖM2 yapısı

Daha önce bahsettiğimiz gibi İÇK-AÖM2 için yaptığımız değişikliler yapımızı çok daha sadeleştirmiştir. Artık tekrarlanan rastgele atamalar yoktur. Rastgele atama sadece ilk katmanda yapılmaktadır. Son katmanın çıkış ağırlık parametrelerini hariç tutarsak, her çıkış ağırlık parametresi 2 defa kullanılmaktadır (her katmanın çıkış ağırlık parametresi olarak ve bir sonraki katmanın giriş ağırlık parametresi olarak). İÇK-AÖM2 yapısının algoritması Algoritma 3'de gösterilmektedir.

Algoritma 3: İÇK-AÖM2 Yapısının Algoritması

Başlangıç;

İlk katman için, ağırlık parametreleri $\mathbf{w}^{(1)}$ ve $Eşik^{(1)}$ rastgele atanır.

$\mathbf{H}^{(1)}$ Eşitlik 3.4'e göre hesaplanır.

Çıkış ağırlık parametreleri $\boldsymbol{\beta}^{(1)}$ Eşitlik 3.20'a göre öğrenilir.

for k=2:M **do**

Mevcut katmanın temsili girişi bir önceki katmanın çıkış matrisi $\mathbf{H}^{(k-1)}$ ile bir önceki katmanın çıkış ağırlık parametrelerinin $\boldsymbol{\beta}^{(k-1)}$ çarpımı ile bulunur.

Mevcut katman için giriş ağırlık parametreleri $\mathbf{w}^{(k)}$ bir önceki çıkış ağırlık parametrelerinin transpozesi $(\boldsymbol{\beta}^{(k-1)})^{T}$ olarak alınmıştır ve $Eşik^{(k)}$ ise rastgele atanır.

$\mathbf{H}^{(k)}$ Eşitlik 3.4'e göre hesaplanır.

Çıkış ağırlık parametreleri $\boldsymbol{\beta}^{(k)}$ Eşitlik 3.20'e (k=M ise Eşitlik 3.21) göre öğrenilir.

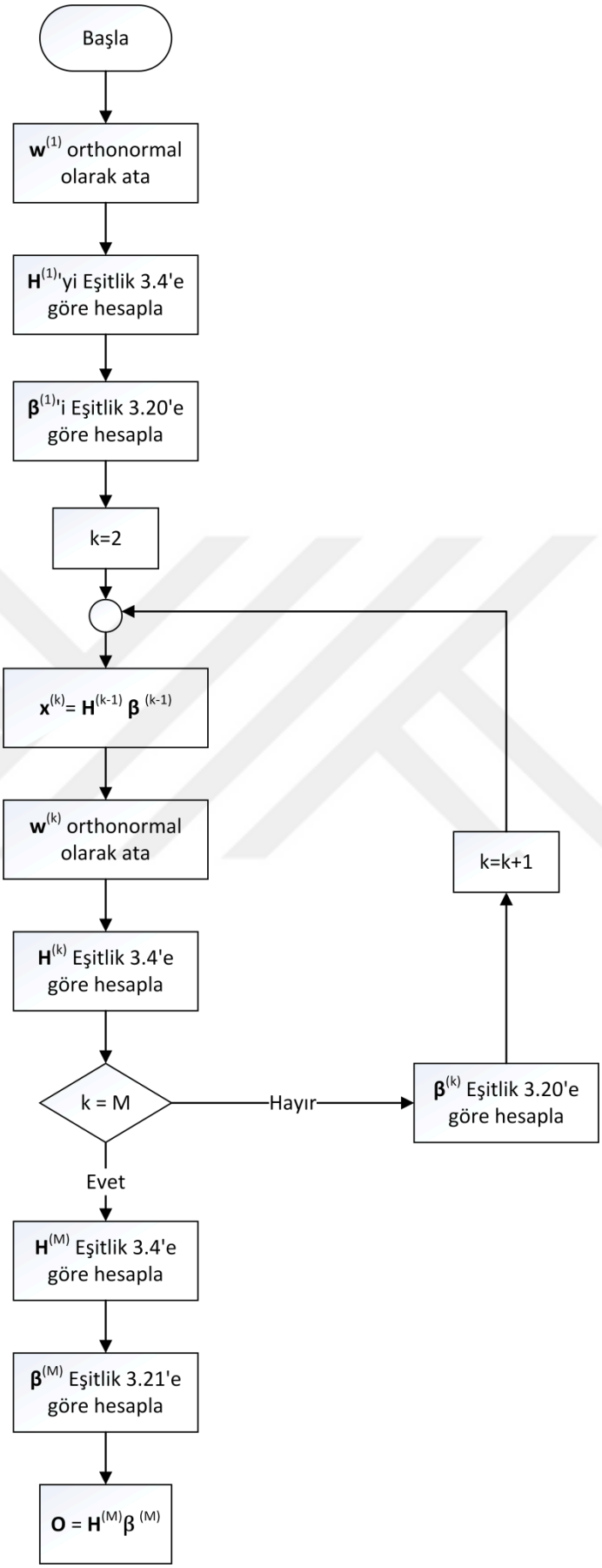
End

Daha önce de bahsettiğimiz gibi İÇK-AÖM2 yapısında eğitim ve test aşamaları aynı yapıyı kullanmaktadır. İÇK-AÖM2'de, eğitim aşamasında kullanılan giriş ağırlık parametreleri $\mathbf{w}$ ve çıkış ağırlık parametreleri $\boldsymbol{\beta}$ test aşamasında da kullanılmaktadır.

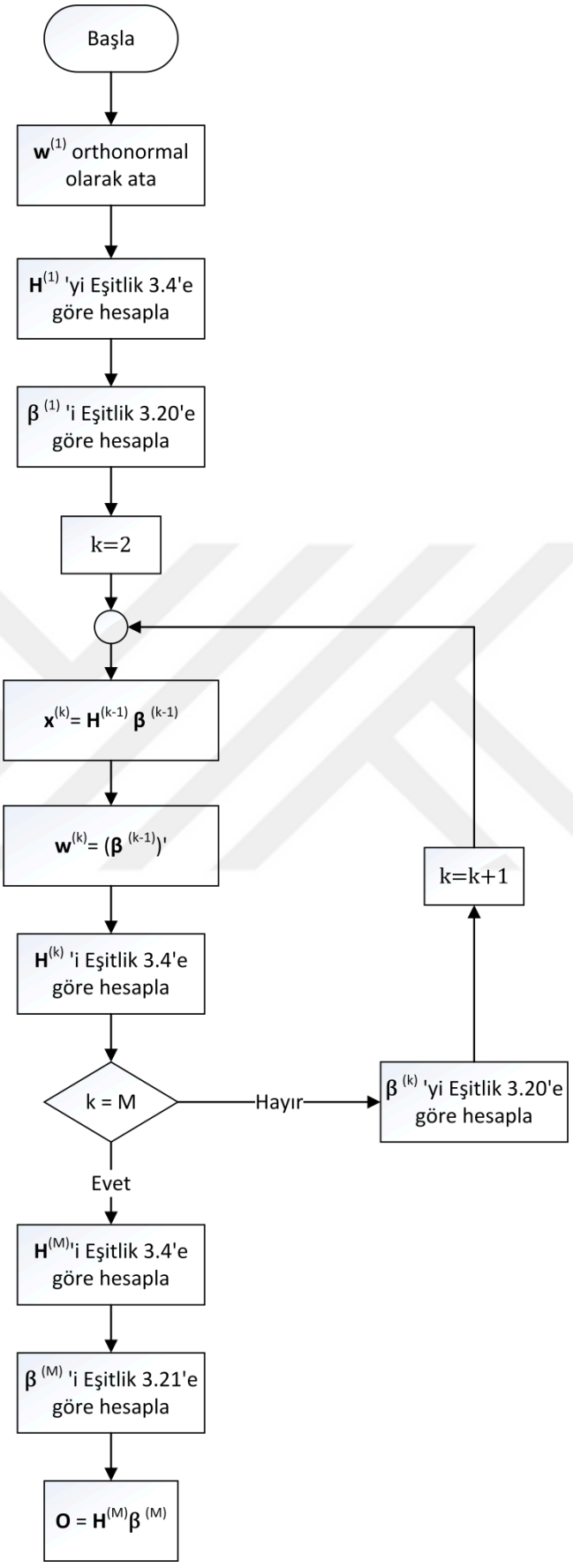
Yeniden belirtmek gerekirse, İÇK-AÖM2, İÇK-AÖM1 gibi eğitimde ve eğitimden sonra hesaplama yapmak için nöronlar kullanmaktadır. Bu özellik sayesinde geliştirdiğimiz 2 yapı, temsili matris kullanmamaları ve adaptif bir hesaplama sunmaları ile ÇK-AÖM'den ayrılmaktadır.

İÇK-AÖM yapılarının kısıtları: Her katmanın nöron sayıları birbirine eşittir. Ayrıca kullanıcı nöron sayısını ve aktivasyon fonksiyonun kendisi seçmektedir.

Algoritmaların daha iyi bir tasviri Şekil 4.3 ve 4.4'de gösterilen akış şemalarında gösterilmektedir. Şemalardaki atama ve öğrenme işlem öbekleri eğitim aşamasında kullanılır. Eğitilen ağ işletilirken aynı işlemleri atama ve öğrenme öbeklerinde eğitim sırasında atanan ya da öğrenilen vektör/matrisi kullanılır. Sadece bu küçük fark ile ağ hem eğitim hem de eğitim sonrası işletilme aşaması için aynı işlem yapısına sahiptir.



Şekil 4.3. M gizli katmanlar için İÇK-AÖM1 ağ yapısının hesaplama akış şeması



Şekil 4.4. M gizli katmanlar için İÇK-AÖM2 ağ yapısının hesaplama akış şeması

Sonuç olarak İÇK-AÖM1 yapıları ÇK-AÖM'den daha hızlıdır (İÇK-AÖM2 çok daha hızlıdır) ve deneylerimizde gösterdiğimiz sonuçlara bakarak belirtebiliriz ki İÇK-AÖM yapıları dinamik sistem modelleme uygulamalarında ÇK-AÖM'e göre çok daha iyi bir başarım elde etmektedir. Bu arada bahsetmeliyiz ki, daha iyi sistem modelleme başarımları elde edebilmek için İÇK-AÖM yapıları otokodlayıcılar yerine aktivasyon fonksiyonları kullanmaktadır.

Geliştirmiş olduğumuz yapı Elsevier Applied Soft Computing dergisinde "Multilayer extreme learning machines and their modeling performance on dynamical systems" başlıklı ile yayınlanmıştır (Kale ve Karakuzu, 2022). Yayınlanmış olduğumuz makaledeki kodların Code Ocean tarafından yeniden üretilebilir olduğu onaylanmıştır. Tekrarlanabilir Kapsül için bağlantı linki: https://doi.org/10.24433/CO.0688672.v1

4.2 Dinamik Sistem Modelleme Problemleri

Sistem modelleme, dinamik bir sistemin modelinin gerçek sistemden alınan giriş-çıkış ölçümleriyle bulunması anlamına gelir. Sistem modellemedeki amaç, belirli bir sisteme veri giriş-çıkışı ilişkisine dayanarak sistem üzerinde tekrar yapılacak çalışmalarda kullanılabilecek bir model kurmaktır.

Sistem modelleme deneyleri için kullanılan denektaşı dinamik sistemleri (DDS) Tablo 4.1'de listelenmiştir. Tablo 4.2'de de DDSler için giriş bileşenleri verilmiştir. Bu çalışmamızda ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 yapılarının modelleme başarımları 7 değişik DDS üzerinde test edilmiştir. ÇK-AÖM ve İÇK-AÖM yapılarında aktivasyon fonksiyonu olarak logaritmik sigmoid transfer fonksiyonu (logsig) kullanılmıştır.

Tablo 4.1. Sistem tanımlama için deneylerde kullanılan denektaşı dinamik sistemleri

Sayı	Denektaşı Dinamik Sistemi
1	$y(k) = \frac{y(k-1)y(k-2)(y(k-1)+2.5)}{1+y^2(k-1)+y^2(k-2)} + u(k)$, (Narendra ve Parthasarathy, 1990) $u(k)_{\text{eğitim}} = \cos\left(\frac{2\pi k}{100}\right), u(k)_{\text{test}} = \sin\left(\frac{2\pi k}{25}\right)$
2	$y(k+1) = \frac{y(k)}{y^2(k)+1} + 1 + u^3(k)$, (Narendra ve Parthasarathy, 1990) $u(k)_{\text{eğitim}} = \cos\left(\frac{2\pi k}{100}\right), u(k)_{\text{test}} = \sin\left(\frac{2\pi k}{25}\right)$
3	$y(k+1) = y(k) + u(k)e^{-3\lvert y(k)\rvert}$, (Babuska, 2001) $u(k)_{\text{eğitim}} = [-1,1]$ aralığında rastgele mutlak değer $u(k)_{\text{test}} = [-1,1]$ aralığında rastgele mutlak değer
4	$y(k+1) = \frac{24+y(k)}{30} y(k) + 0.8\frac{u^2(k)}{1+u^2(k)} y(k-1) + 0.5u(k)$, (Oussar vd., 1998)

	$u(k)_{eğitim} = [-5,5]$ aralığında rastgele mutlak değer $u(k)_{test} = [-5,5]$ aralığında rastgele mutlak değer
5	$y(k+1) = 0.5\left(\frac{y(k)}{1+y^2(k)} + \left(1+u(k)\right)u(k)(1-u(k))\right)$, (Sastry vd., 1994) $u(k)_{eğitim} = [-2,2]$ aralığında rastgele mutlak değer $u(k)_{test} = [-2,2]$ aralığında rastgele mutlak değer
6	$y(k+1) = \frac{y(k)y(k+1)y(k-2)u(k-1)(y(k-2)-1)+u(k)}{1+y^2(k-1)+y^2(k-2)}$, (Juang, 2002) $u(k)_{eğitim} = \begin{cases} \sin\left(\frac{\pi k}{25}\right), & k < 250 \\ 1, & 250 \le k < 500 \\ -1, & 500 \le k < 750 \\ 0.3\sin\left(\frac{\pi k}{25}\right) + 0.1\sin\left(\frac{\pi k}{32}\right) + 0.6\sin\left(\frac{\pi k}{10}\right), & k \ge 750 \end{cases}$ $u(k)_{test} = \begin{cases} -0.15\cos\left(\frac{\pi k}{50}\right) + 0.5\cos\left(\frac{\pi k}{16}\right) - 0.3\cos\left(\frac{\pi k}{20}\right), & k < 250 \\ 1, & 250 \le k < 500 \\ -1, & 500 \le k < 750 \\ \cos\left(\frac{\pi k}{50}\right), & k \ge 750 \end{cases}$
7	$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(k)$, (Jang, 1993) $f(k) = \sin \pi u(k) + 0.3\sin 3\pi u(k) + 0.1\sin 5\pi u(k)$ $u(k)_{eğitim} = \begin{cases} \sin\left(\frac{2\pi k}{250}\right), & k < 500 \\ 0.5\sin\left(\frac{2\pi k}{250}\right) + 0.5\sin\left(\frac{2\pi k}{25}\right), & k \ge 500 \end{cases}$ $u(k)_{test} = \begin{cases} -0.15\cos\left(\frac{\pi k}{50}\right) + 0.5\sin\left(\frac{\pi k}{16}\right) - 0.3\cos\left(\frac{\pi k}{20}\right), & k < 234 \\ \cos\left(\frac{2\pi k}{125}\right), & 234 \le k < 467 \\ 0.3\sin\left(\frac{2\pi k}{250}\right) + 0.1\cos\left(\frac{\pi k}{64}\right) - 0.6\sin\left(\frac{\pi k}{20}\right), & k \ge 467 \end{cases}$

Bu denektaşı dinamik sistemlerinde *u(k)* harici kontrol işaretidir. Her DDS için *u(k)* eğitim ve test veri kümelerini hazırlamak için kullanılır. DDS 1, DDS 2, ..., DDS 5 üzerinde yapılan deneylerde eğitim için 100 örnek, test için 100 örnek kullanılmıştır. DDS 6 üzerinde yapılan deneyde eğitim için 1000 örnek, test için 1000 örnek kullanılırken DDS 7 üzerinde yapılan deneylerde her iki aşama için 700 örnek kullanılmıştır. Modelleme için kullanılan giriş yapılandırması Tablo 4.2'de veridiği gibidir.

Tablo 4.2. Sistem tanımlaması için giriş bileşimleri

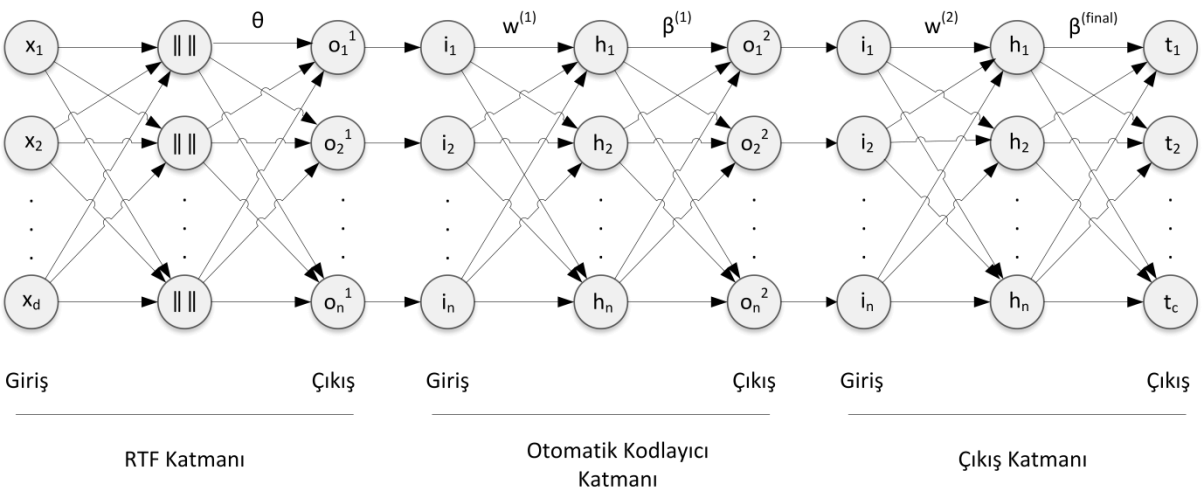
DDS#	Girişler
DDS 1	$y(k-1), y(k-2), u(k)$
DDS 2	$y(k), u(k)$
DDS 3	$y(k), u(k)$
DDS 4	$y(k), y(k-1), u(k)$
DDS 5	$y(k), u(k)$
DDS 6	$y(k), y(k-1), y(k-2), u(k), u(k-1)$
DDS 7	$y(k), y(k-1), u(k)$

5. ENGEBELİ VE KESKİN DEĞİŞİMLİ YÜZEYLERE SAHİP SİSTEM MODELLEME

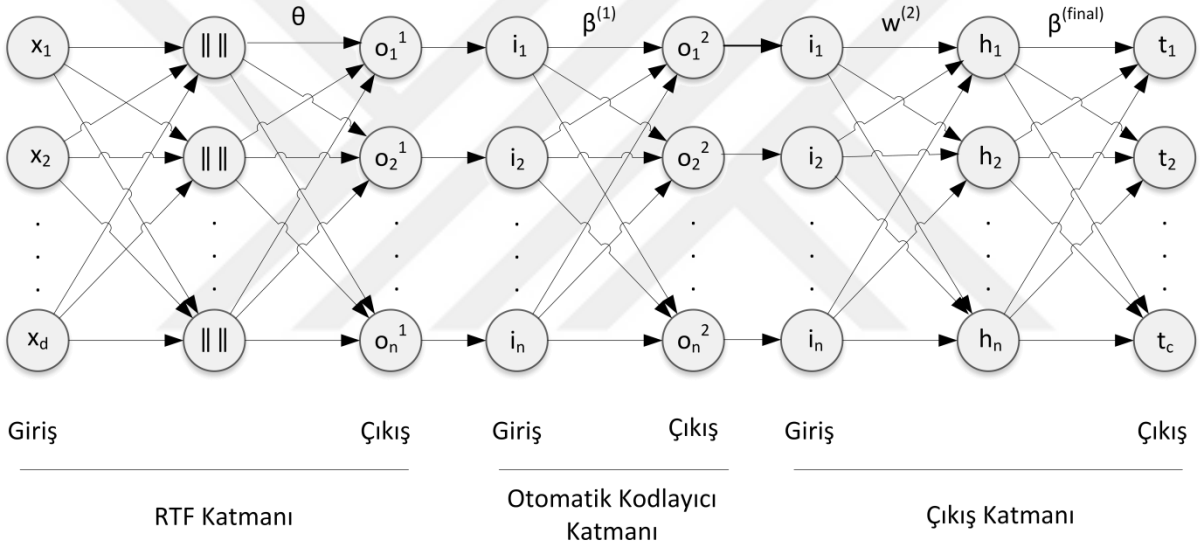
Bu çalışmada çok katmanlı aşırı öğrenme makinesi ile radyal tabanlı fonksiyonlu ağları birleştirerek HybRBF-ML-ELM adını verdiğimizağ yapısını tasarladık. Çok katmanlı aşırı öğrenme makinelerinde yüzey modelleme lokal optima noktalarını belirlemeden yapılmaktadır. Bu dezavantaj nedeniyle ÇK-AÖM yapısında lokal optimaları belirleyebilmek ve bu şekilde ÇK-AÖM yapısının yüzey modelleme başarımını arttırabilmek için yapıya bir Radyal Tabanlı Fonksiyon Ağı (RTFA) katmanı eklenmiştir. RTFA ÇK-AÖM yapısının ilk katmanına entegre edilmiştir. Deneysel sonuçlar, HybRBF-ML-ELM yapısının ÇK-AÖM yapısına göre çok daha iyi bir yüzey modelleme başarımı sağladığını açıkça göstermektedir.

5.1. Yeni Hibrit Radyal Temelli Fonksiyonlu Çok Katmanlı AÖM Yapısı

Geliştirilmiş olan yapının ağ yapısı toplamda 3 katmandan oluşmaktadır. Bu katmanlar sırasıyla RTF, AE ve Çıkış katmanlarıdır. İlk katmanda, Gauss fonksiyonu çekirdek olarak kullanılmaktadır. AE ve Çıkış katmanlarında kullanılan aktivasyon fonksiyonu hiperbolik tanjant (*tansig*)'dir. Şekil 5.1'de HybRF-ML-ELM'in yapısını eğitim ve test aşamaları için gösterilmektedir. RTF katmanında, Gauss aktivasyon fonksiyonunun merkez parametreleri homojen olarak örnek aralığına dağıtılmıştır. Standart sapma değerleri [0.5-1] aralığında rastgele atanmıştır. RTF katmanının ardından mevcut katmanın rastgele atanmış ağırlık parametreleri bir önceki katmanın çıkışı ile çarpılır ve AE katmanının temsili girişi hesaplanmış olur.



(a)



(b)

Şekil 5.1. HybRBF-ML-ELM yapısı: eğitim aşaması (a), test aşaması (b)

RTF katmanı bir giriş verisini belirli bir kümeyle ilişkilendirir. Eşitlik 5.1'de gösterilen Gauss fonksiyonu çekirdek olarak kullanılır. RTF'nin avantajı, keskin geçişler için esnek modelleme yeteneğinde ve girdi-çıktı haritalama yüzeyinde çoklu minimum/maksimumda yatmaktadır. Bu katmanın bağlantıları birimdir ve girdileri bu katmandaki nöronlara iletirler.

$$u(x) = \sum_{i=1}^{N} e^{-\frac{\|x-c_i\|^2}{\sigma_i^2}} \quad (5.1)$$

Gauss fonksiyon parametrelerinde x girişi, c_i merkez vektörlerini ve σ_i standart sapmaları temsil eder. AE ve Çıkış katmanları için, **H** matrisi *tansig* kullanılarak hesaplanır. Tasarlanan yapının eğitim ve test aşamaları için işlemler sırasıyla Algoritma 4 ve Algoritma 5'de verilmektedir.

Algoritma 4: HybRBF-ML-ELM yapısının eğitim aşaması algoritması.

Başlangıç

RTF Katmanı için;

Merkez parametreleri örnek aralığında homojen olarak dağıtılır.
Standart sapma parametreleri rastgele olarak [0.5-1] aralığında atanır.
Eşitlik 5.1'e göre Gauss çekirdek çıkışı hesaplanır.
Bir sonraki katmanın temsili girişi, elde edilmiş sonucun rastgele orthonormal olarak atanan θ ağırlık parametreleriyle çarpılmasıyla hesaplanır.

AE Katman için;

Ağırlık parametreleri $\mathbf{w}^{(1)}$ rastgele orthonormal olarak atanır.
$\mathbf{H}^{(1)}$ Eşitlik 3.4 kullanılarak hesaplanır.
Çıkış ağırlık parametresi $\boldsymbol{\beta}^{(1)}$ Eşitlik 3.20 kullanılarak öğrenilir.

Çıkış Katmanı için;

Mevcut katmanın giriş matrisi, bir önceki katmanın giriş matrisi ile $\boldsymbol{\beta}^{(1)}$ parametresinin çarpılmasıyla elde edilir.
Ağırlık parametreleri $\mathbf{w}^{(2)}$ rastgele orthonormal olarak atanır.
Eşitlik 3.4 kullanılarak H^{final} hesaplanır.
Çıkış ağırlık parametresi $\boldsymbol{\beta}^{(final)}$ Eşitlik 3.21 kullanılarak öğrenilir.

Algoritma 5: HybRBF-ML-ELM yapısının test aşaması algoritması.

Başlangıç

RTF Katmanı için;

Eşitlik 5.1'e göre eğitim aşamasında atanmış merkez ve standart sapma değerleriyle Gauss çekirdek çıkışı hesaplanır.
Bir sonraki katmanın temsili girişi, Gauss aktivasyon fonksiyonunun çıkışıyla eğitim aşamasında atanan θ ağırlık parametrelerinin çarpılmasıyla hesaplanır.

AE Katman için;

Temsili giriş $\boldsymbol{\beta}^{(1)}$ ile çarpılır ve bir sonraki katmanın temsili girişi bulunur.

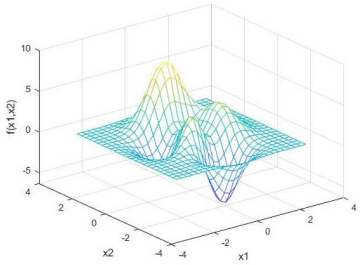
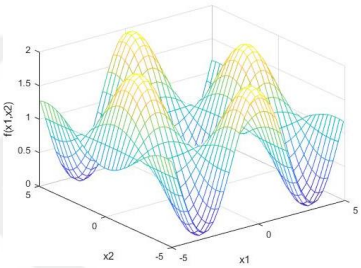
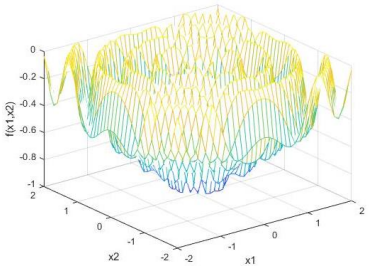
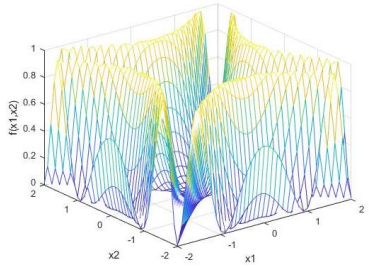
Çıkış Katmanı için;

Eğitim aşamasında atanmış ağırlık parametreleriyle ($\mathbf{w}^{(2)}$) Eşitlik 3.4 kullanılarak $\mathbf{H}^{final}$ hesaplanır.
$\mathbf{O} = \mathbf{H}^{final}\,\boldsymbol{\beta}^{(final)}$ ile çıkış hesaplanır.

5.2. Yüzey Modellemede Kullanılan Kıyaslama Fonksiyonları

Bu çalışmada yüzey modelleme problemleri için Peaks, Griewank, Dropwave ve Schaffer 2 olmak üzere 4 adet kalite testi fonksiyonu kullanılmıştır. Bu fonksiyonların matematiksel tanımları, örnek uzayları ve 3 boyutlu yüzey betimlemeleri Tablo 5.1'de gösterilmiştir.

Tablo 5.1. Denektaşı test fonksiyonları

FONKSİYON ADI	Matematiksel Tanımı		3 Boyutlu Yüzeyi
Peaks	$f(x) = 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2-x_2^2} - \frac{1}{3} e^{-(x_1+1)^2-y^2}$	$x_i \in [-3, 3]$, i=1, 2.	
Griewank	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-5, 5]$, i=1, …, d.	
Dropwave	$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	$x_i \in [-2, 2]$, i=1, 2.	
Schaffer2	$f(x) = 0.5 + \frac{sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	$x_i \in [-2, 2]$, i=1, 2.	

6. DENEY SONUÇLARI

Dinamik sistem modelleme ve engebeli, keskin değişimki yüzeylere sahip sistem modelleme problemleri üzerinde bu çalışma kapsamında geliştirilen üç özgün AÖM yapısının deney sonuçları bu bölümde yer almaktadır.

6.1. Çok Katmanlı AÖM Yapıları ile Dinamik Sistem Modelleme Uygulaması

Deneyler Intel (R) Core (TM) i7-4510U CPU 2 GHz, 8.00 GB RAM, 64 bit Windows 8.1 işletim sistemli dizüstü bilgisayarda gerçekleştirilmiştir. Her DDS için ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 yapıları 100 defa koşulmuştur. Deneyler değişik düğüm sayıları ile yapılmıştır.

Her DDS için başarım analizleri 10, 15, 20, 25, 30, 40, 50, 100 düğüm kullanarak yapılmıştır. Katman sayısı 3 olarak alınmış ve her katmandaki düğüm sayısı aynıdır.DDS 1'den DDS 7'ya kadar olan analiz sonuçları Tablo 6.1'de listelenmektedir. Bu tabloda verilen metrikler, yukarıda anılan her bir düğüm sayısı için ayrı ayrı 100er kez yapılmış deneyler sonucunda elde edilmiş metriklerin ortalamasıdır. Bu tabloda 3 kategoride başarım metrikleri verilmiştir. Eğitim için geçen süre saniye, kare kök ortalama karesel hata (RMSE) cinsinden eğitim veri seti için başarım, RMSE cinsinden test veri seti için başarım. Bunun yanı sıra İÇK-AÖM yapılarının ÇK-AÖM yapısına oranla başarım farkı yüzde cinsinden eklenmiştir.

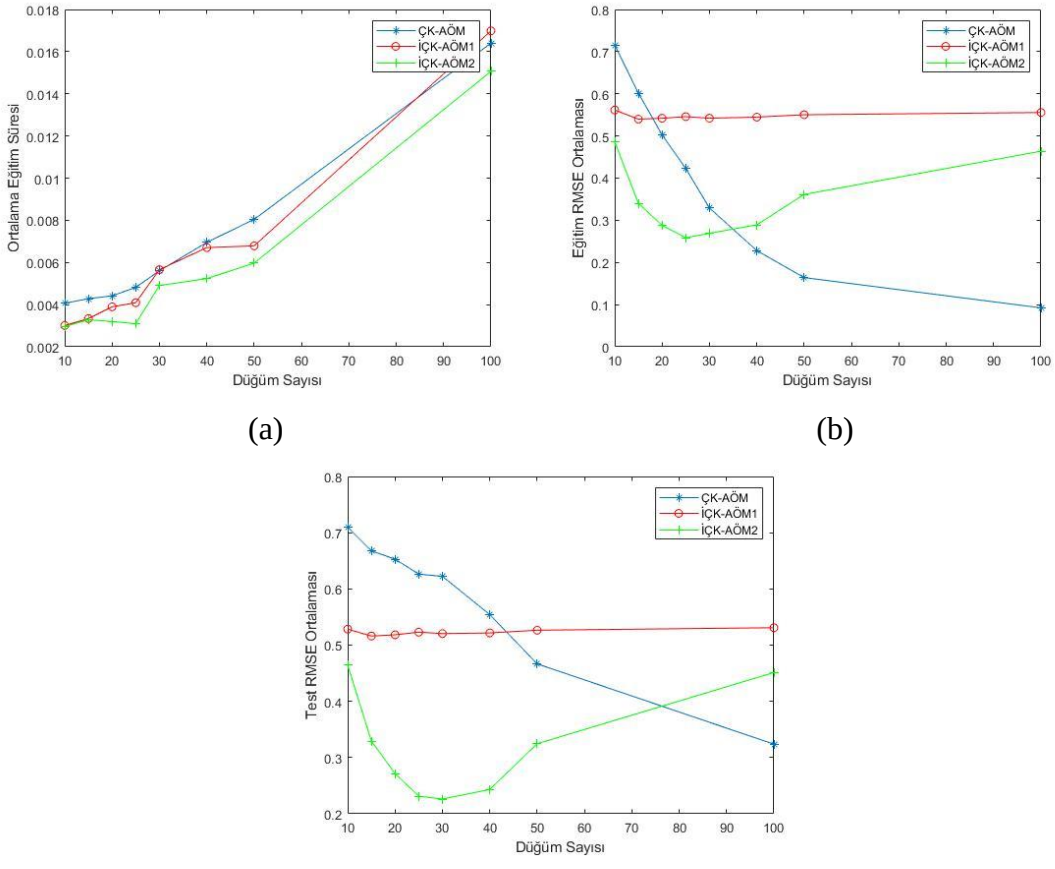
Tablo 6.1. Tüm DDS'ler için başarım analiz metrikleri.

DDS	Yapı	Eğitim Zamanı				Eğitim RMSE				Test RMSE			
		Ort	En İyi	En Kötü	Std	Ort	En İyi	En Kötü	Std	Ort	En İyi	En Kötü	**Std**
1	ÇK-AÖM	0.007754	0.005295	0.031980	0.003166	0.691660	0.374657	0.897934	0.107618	0.939327	0.513410	1.656150	0.198839
	İÇK-AÖM1	0.006968 (-10%)	0.005487 (4%)	0.062573 (96%)	0.005900 (86%)	0.158001 (-77%)	0.137317 (-63%)	0.182700 (-80%)	0.008671 (-92%)	0.254354 (-73%)	0.217501 (-58%)	0.304714 (-82%)	0.016661 (-92%)
	İÇK-AÖM2	0.005465 (-30%)	0.003312 (-37%)	0.040225 (26%)	0.005721 (81%)	0.195109 (-72%)	0.092293 (-75%)	0.333822 (-63%)	0.050543 (-53%)	0.313705 (-67%)	0.159599 (-69%)	0.576434 (-65%)	0.081994 (-59%)
2	ÇK-AÖM	0.008257	0.005756	0.030778	0.002972	0.111888	0.043956	0.200160	0.033147	0.255077	0.118665	0.502613	0.077562
	İÇK-AÖM1	0.008302 (1%)	0.005494 (-5%)	0.136396 (343%)	0.013360 (350%)	0.113521 (1%)	0.109538 (149%)	0.119640 (-40%)	0.001948 (-94%)	0.213350 (-16%)	0.205675 (73%)	0.222668 (-56%)	0.003305 (-96%)
	İÇK-AÖM2	0.004934 (-40%)	0.003307 (-43%)	0.027449 (-11%)	0.003094 (4%)	0.098776 (-12%)	0.084419 (92%)	0.114241 (-43%)	0.005869 (-82%)	0.162801 (-36%)	0.131467 (11%)	0.205828 (-59%)	0.013406 (-83%)
3	ÇK-AÖM	0.006757	0.004935	0.022812	0.002049	0.043788	0.016991	0.086269	0.013856	0.116940	0.040256	0.261525	0.042354
	İÇK-AÖM1	0.006503 (-4%)	0.005479 (11%)	0.0029863 (31%)	0.002820 (38%)	0.096702 (121%)	0.090577 (433%)	0.103280 (20%)	0.002374 (-83%)	0.137063 (17%)	0.111311 (181%)	0.169053 (-35%)	0.011942 (-72%)
	İÇK-AÖM2	0.005088 (-25%)	0.003296 (-33%)	0.035670 (56%)	0.004138 (102%)	0.084991 (94%)	0.073025 (330%)	0.097113 (13%)	0.004944 (-64%)	0.138689 (19%)	0.102517 (155%)	0.204633 (-22%)	0.020060 (-53%)
4	ÇK-AÖM	0.008639	0.006764	0.032403	0.003101	0.457440	0.368015	0.559396	0.040371	0.880225	0.740992	1.028787	0.054026
	İÇK-AÖM1	0.006466 (-25%)	0.005499 (-19%)	0.027912 (-14%)	0.002500 (-19%)	0.197119 (-57%)	0.162733 (-56%)	0.230555 (-59%)	0.013488 (-67%)	0.445663 (-49%)	0.389518 (-47%)	0.506575 (-51%)	0.024916 (-54%)
	İÇK-AÖM2	0.004304 (-50%)	0.003295 (-51%)	0.013222 (-59%)	0.001491 (-52%)	0.205383 (-55%)	0.128052 (-65%)	0.289463 (-48%)	0.033586 (-17%)	0.553100 (-37%)	0.419298 (-43%)	0.692630 (-33%)	0.053157 (-2%)
5	ÇK-AÖM	0.006823	0.005046	0.025599	0.002349	0.382042	0.241470	0.573292	0.066938	0.577983	0.324601	0.909941	0.116529
	İÇK-AÖM1	0.006308 (-8%)	0.005480 (9%)	0.018392 (-28%)	0.001679 (-29%)	0.547743 (43%)	0.493925 (105%)	0.581726 (1%)	0.017752 (-73%)	0.523209 (-9%)	0.473988 (46%)	0.550691 (-39%)	0.015721 (-87%)
	İÇK-AÖM2	0.005463 (-20%)	0.003380 (-33%)	0.022324 (13%)	0.002654 (13%)	0.344570 (-10%)	0.241380 (0%)	0.472900 (-18%)	0.048500 (-28%)	0.317759 (-45%)	0.208029 (-36%)	0.469107 (-48%)	0.052907 (-55%)
6	ÇK-AÖM	0.499755	0.458654	0.805358	0.047345	0.093329	0.046668	0.149796	0.021389	0.130810	0.078266	0.190190	0.025004
	İÇK-AÖM1	0.379242 (-24%)	0.320060 (-30%)	0.517136 (-36%)	0.035066 (-26%)	0.034227 (-63%)	0.029672 (-36%)	0.040873 (-73%)	0.002423 (-89%)	0.029657 (-77%)	0.023857 (-70%)	0.039111 (-79%)	0.003263 (-87%)
	İÇK-AÖM2	0.206773 (-59%)	0.175776 (-62%)	0.310811 (-61%)	0.031368 (-34%)	0.032556 (-65%)	0.024729 (-47%)	0.044372 (-70%)	0.003631 (-83%)	0.030308 (-77%)	0.020966 (-73%)	0.049632 (-74%)	0.005343 (-79%)
7	ÇK-AÖM	0.201866	0.177324	0.288117	0.017631	1.129291	0.769299	1.301125	0.110983	1.399192	0.867911	2.120719	0.211331
	İÇK-AÖM1	0.154516 (-23%)	0.128529 (-28%)	0.313913 (9%)	0.029553 (68%)	0.109072 (-90%)	0.095379 (-88%)	0.133171 (-90%)	0.006761 (-94%)	0.121563 (-91%)	0.109850 (-87%)	0.143644 (-93%)	0.006111 (-97%)
	İÇK-AÖM2	0.074915 (-63%)	0.063109 (-64%)	0.130063 (-55%)	0.012709 (-28%)	0.367961 (-67%)	0.165305 (-79%)	0.672807 (-48%)	0.112064 (1%)	0.413887 (-70%)	0.178144 (-79%)	0.802708 (-62%)	0.132409 (-37%)

Tüm deney sonuçlarının ortalamaları Tablo 6.1'da verilmektedir. İÇK-AÖM sonuçlarının altında yazan yüzdelik rakamlar İÇK-AÖM yapılarının ÇK-AÖM yapısının başarımına göre değişimini göstermektedir. Örneğin, eğer yüzdelik rakam negatifse, bu İÇK-AÖM yapısının daha başarılı olduğunu belirtmektedir. Bu veriler ışığında Tablo 6.1 göstermektedir ki DDS 3 haricinde İÇK-AÖM2 yapısı ÇK-AÖM yapısından daha iyi sistem modelleme sonuçları vermektedir. Öte yandan İÇK-AÖM1 yapısı DDS 2, 3, ve 5 için daha iyi eğitim RMSE'si sunarken DDS 2 ve 5 için daha iyi test RMSE'si sunmaktadır.

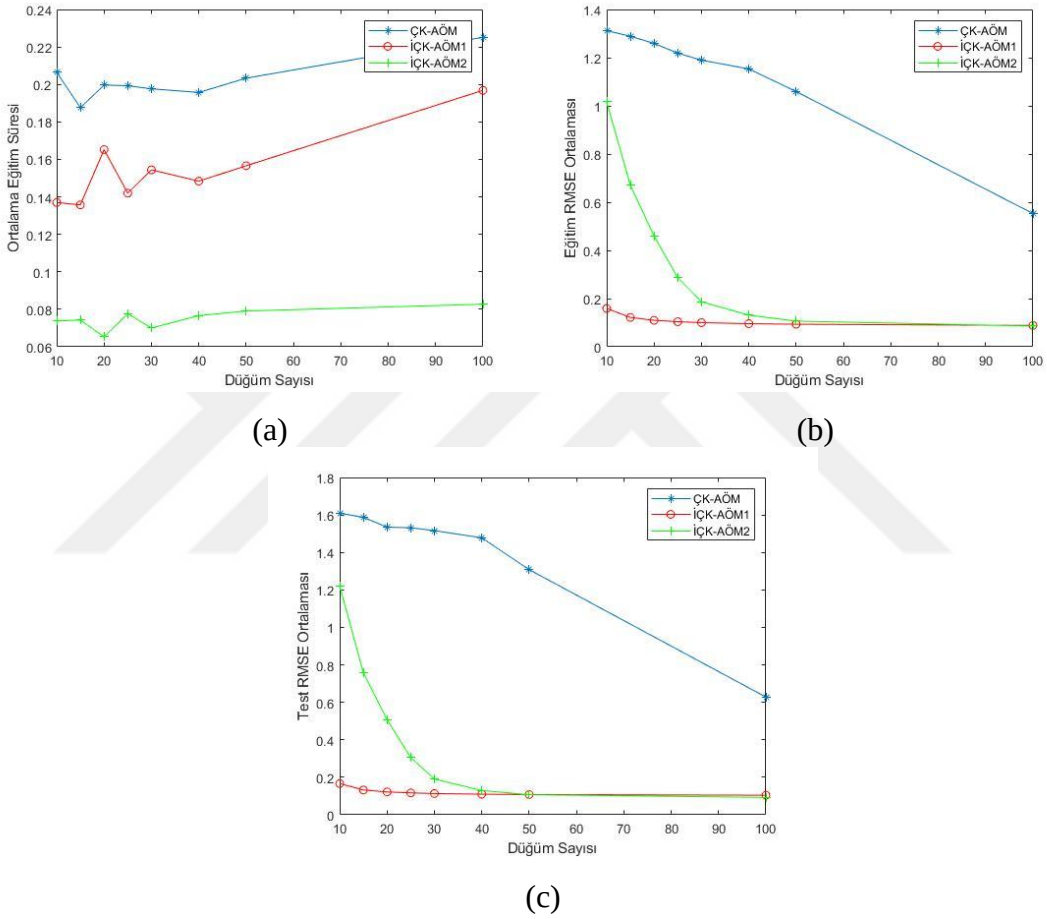
DDS 3 detayları ve keskin geçişleri olmayan bir problemdir ve bu da ÇK-AÖM'in İÇK-AÖM yapılarına göre ortalama eğitim RMSE ve ortalama test RMSE açılarından daha iyi sonuç verme nedenidir. Ayrıca tüm DDSler için İÇK-AÖM yapıları çok daha hızlı hesaplama yapmaktadır.

Değişik düğüm sayılarına göre DDS 5 ve DDS 7 için ortalama eğitim zamanı, ortalama eğitim RMSE ve ortalama test RMSE değişimleri Şekil 6.1 ve 6.2'de verilmektedir.



Şekil 6.1. Düğüm sayısına göre DDS 5 için eğitim zamanı (a), eğitim veri kümesi için RMSE (b), test veri kümesi için RMSE (c) cinsinden başarım sonuçları.

DDS 5 için deney sonuçlarına bakıldığında, ÇK-AÖM en iyi başarımını 100 düğüm kullanarak elde ederken İÇK-AÖM1 en iyi başarımını 15 düğüm kullanarak elde etmiştir. Öte yandan İÇK-AÖM2 ise en iyi başarımını 30 düğüm kullanarak elde etmiştir. 100 düğüm kullanarak ÇK-AÖM, 0.016405 eğitim zamanında 0.323780 ortalama test RMSE elde etmektedir. Buna karşılık olarak İÇK-AÖM2, 0.004907 eğitim zamanında 0.226226 ortalama test RMSE elde etmektedir. DDS 5 için İÇK-AÖM2 sadece daha başarılı değil aynı zamanda %70 daha hızlı çalışmaktadır.

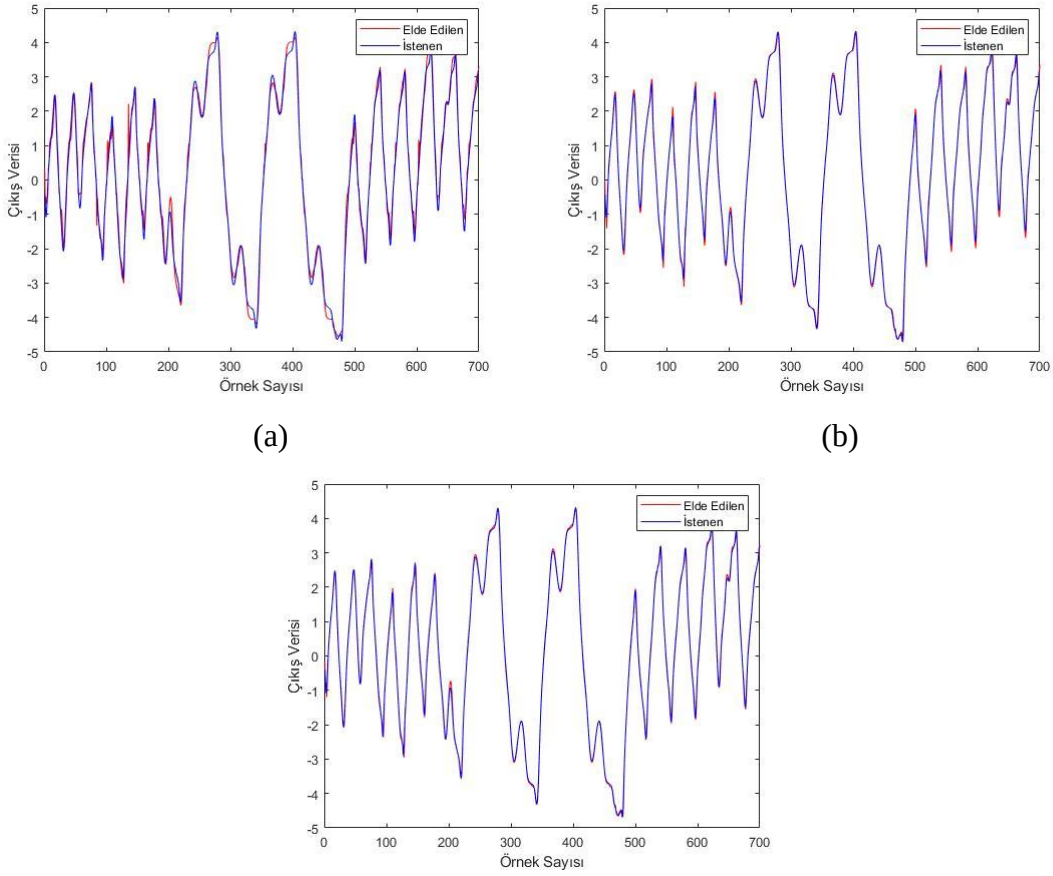


Şekil 6.2. Düğüm sayısına göre DDS 7 için eğitim zamanı (a), eğitim veri kümesi için RMSE (b), test veri kümesi için RMSE (c) cinsinden başarım sonuçları.

Şekil 6.2'deki hesaplama zamanlarına bakıldığında, DDS 7 modellleme başarımları DDS 5 modelleme başarımlarına benzemekte ve 3 model için farklılıklar görülmektedir. ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 için ortalama eğitim zamanları 0.201866, 0.154516, ve 0.074915'dir. Açıkça İÇK-AÖM yapılarının ÇK-AÖM yapısından daha hızlı oldukları görülmektedir. İÇK-AÖM1, ÇK-AÖM yapısından 23% hızlıyken, İÇK-AÖM2 yapısı ÇK-AÖM yapısından 63% hızlıdır. DDS 7 için her düğüm sayısında İÇK-AÖM yapıları ÇK-AÖM

yapısından ortalama test RMSE cinsinden daha iyi başarım sunmaktadır. ÇK-AÖM yapısının en iyi başarımı 100 düğüm kullanarak elde edilmiştir. 100 düğüm kullanıldığında ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 yapıları 0.627977, 0.104272 ve 0.092683 ortalama test RMSE sonuçları vermektedir. ÇK-AÖM ile karşılaştırıldığında İÇK-AÖM1 yapısı 83% daha iyi ortalama test RMSE sonuçları sağlarken İÇK-AÖM2 yapısı 85% daha iyi ortalama test RMSE sonuçları sağlamaktadır. 100 düğüm için hesaplama hızları karşılaştırıldığında İÇK-AÖM1 ÇK-AÖM yapısından 13% hızlıyken, İÇK-AÖM2 ÇK-AÖM yapısından 63% hızlıdır. DDS 7 için açıkça İÇK-AÖM yapıları ÇK-AÖM yapısına göre çok daha iyi sonuçlar vermiştir.

Şekil 6.3 ve 6.4'de DDS 7 'nin 100 düğüm için ÇK-AÖM, İÇK-AÖM1 ve İÇK-AÖM2 yapılarının en iyi ve en kötü modelleme başarımları sergilenmektedir. En iyi RMSE değerleri sırasıyla; 0.231169, 0.102676 ve 0.080124'dir. En kötü RMSE değerleri ise sırasıyla; 1.328345, 0.106210 ve 0.108676'dır. RMSE değerlerine ve bu şekillere bakılarak İÇK-AÖM2'nin İÇK-AÖM1'e göre daha başarılı bir sistem modelleme yapısı olduğu söylenebilirken, İÇK-AÖM1 yapısı da ÇK-AÖM yapısına göre daha iyi bir sistem modelleme yapısıdır.

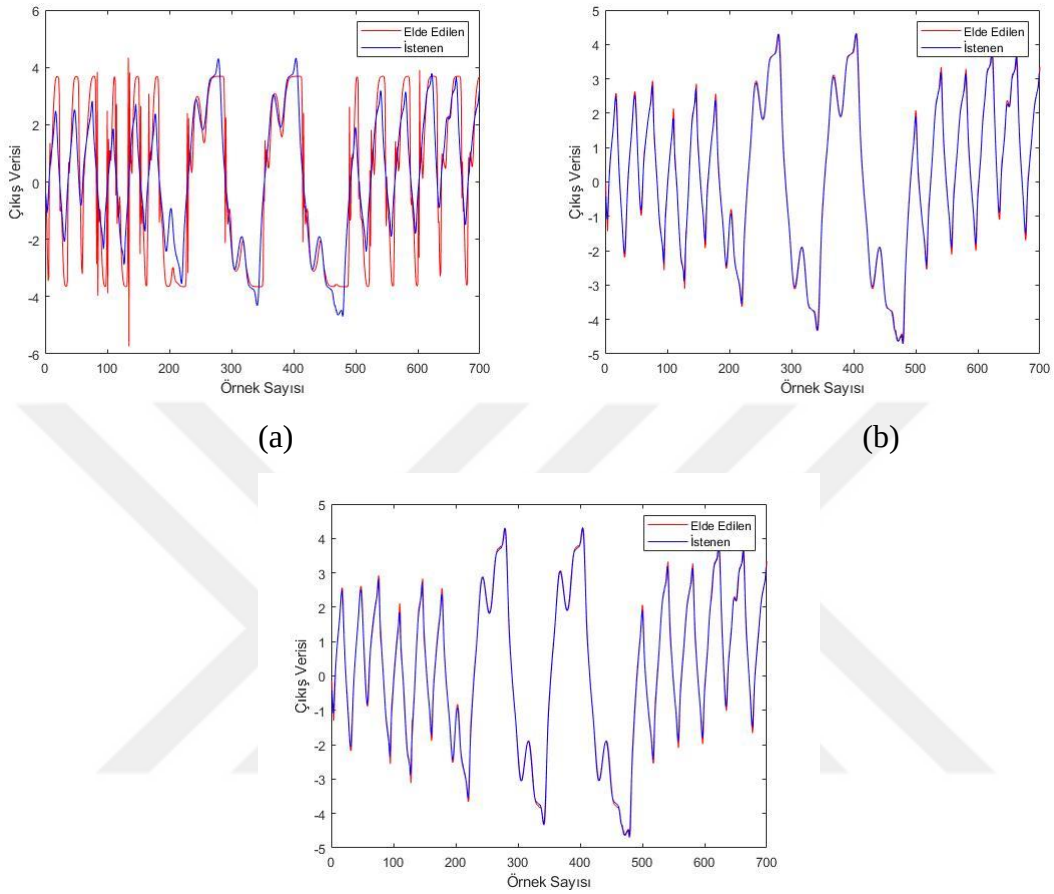


(a) (b)

(c)

Şekil 6.3. DDS 7'nin 100 düğüm için ÇK-AÖM (a), İÇK-AÖM1 (b), İÇK-AÖM2 (c) yapılarının en iyi sistem modelleme başarımları.

Şekil 6.3'e bakıldığında, detaylı ve keskin geçişlerde, İÇK-AÖM yapıları ÇK-AÖM yapısına göre daha iyi başarımlar sunmaktadır. Kendi aralarında kıyaslandıklarında, İÇK-AÖM2 yapısı İÇK-AÖM1 yapısına göre daha başarılıdır.



(a) (b)

(c)

Şekil 6.4. DDS 7'nin 100 düğüm için ÇK-AÖM (a), İÇK-AÖM1 (b), İÇK-AÖM2 (c) yapılarının en kötü sistem modelleme başarımları.

Tüm yapılar için Şekil 6.4'de 100 kez koşma sınucunda elde edilen en kötü modelleme başarımlarını gösterilmektedir. ÇK-AÖM yapısının en kötü modelleme defoları İÇK-AÖM yapılarına göre çok daha belirgindir. En kötü durumlarında bile İÇK-AÖM yapıları ÇK-AÖM yapısına göre daha iyi modelleme sunmaktadır.

6.2. Engebeli ve Keskin Değişimli Yüzeylere Sahip Sistem Modelleme Uygulaması

Bu çalışmada deneyler Intel (R) Core (TM) i7-8565U CPU @ 1.80 GHz, 32.00 GB RAM, 64 bit Windows 10 Enterprise işletim sistemli masaüstü bilgisayar üzerinde yapılmıştır.

ÇK-AÖM ve HybRBF-ML-ELM yapılarında, yüzey modelleme deneylerinde eğitim için *Peaks*, *Griewank*, *Dropwave* ve *Schaffer2* fonksiyonları sırasıyla 256, 1156, 1681 ve 1681 örnek kullanır. Test aşaması için ise *Peaks*, *Griewank*, *Dropwave* ve *Schaffer2* fonksiyonları sırasıyla 169, 676, 441 ve 441 örnek kullanır. *Peaks*, *Griewank*, *Dropwave* ve *Schaffer2* fonksiyonlarının HybRBF-ML-ELM ağı ile modellenmesinde sırasıyla 225, 841, 729 ve 729 düğüm kullanılmaktadır. HybRBF-ML-ELM ağının her katmanında aynı sayıda düğüm kullanılmıştır. Her fonksiyon için deneyler 30 defa yapılmıştır. Bu çalışmadaki deneyler için ÇK-AÖM ve HybRBF-ML-ELM yapılarında aktivasyon fonksiyonu olarak *tansig* kullanılmıştır.

ÇK-AÖM yapısı 3 katman ile kullanılmıştır. Her katman için kullanılan λ parametreleri sırasıyla 1000, 10^7 ve 10^9 olarak alınmıştır. Bu rakamlar (Wong, Vong, Wong ve Cao, 2018b)'dan esinlenerek ve çeşitli denemeler sonucunda uygun olduğu görüldüğünden tercih edilmiştir.

HybRBF-ML-ELM yapısının RTF katmanında Gauss aktivasyon fonksiyonu kullanılmıştır. Bu fonksiyonun merkez değerleri denektaşı fonksiyonunun örnek uzayında homojen dağılımlı olarak atanmıştır. Standart sapma değeri de ayrıca 0.5 ve 1 değerleri arasında rastgele atanmıştır. HybRBF-ML-ELM yapısının öteki katmanlarında (*AE* ve *Çıkış* katmanları) λ parametreleri sırasıyla 1000 ve 10^9 olarak alınmıştır.

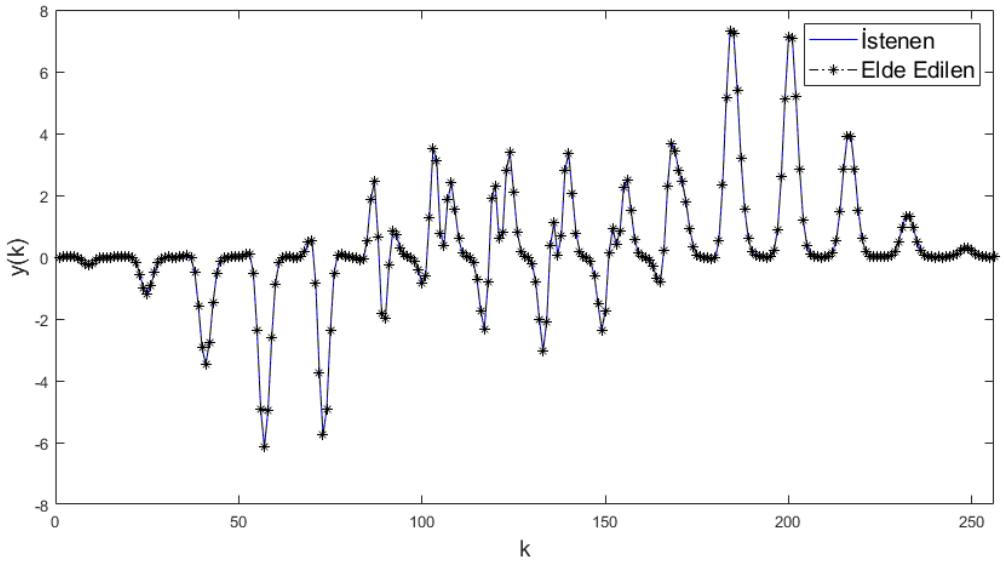
Tablo 6.2'de yüzey modelleme başarımları her denektaşı test fonksiyonu için ortalama en iyi RMSE, en kötü RMSE ve standart sapma cinslerinden verilmektedir. Her bir yapı için ilgili algoritmalar 30 kez koşturulup ortalama değerleri verilmiştir. Bunun yanı sıra HybRBF-ML-ELM yapısının ÇK-AÖM yapısına oranla başarım farkı yüzde cinsinden eklenmiştir. Bu yüzdelerin verildiği satırlarda siyah nokta işareti (•) bu çalışmada geliştirilen HybRBF-ML-ELM yapısının rakibi ÇK-AÖM yapısından daha başarılı olduğunu göstermektedir.

Tablo 6.2. HybRBF-ML-ELM ile ÇK-AÖM yapılarının yüzey modelleme başarımlarının karşılaştırması

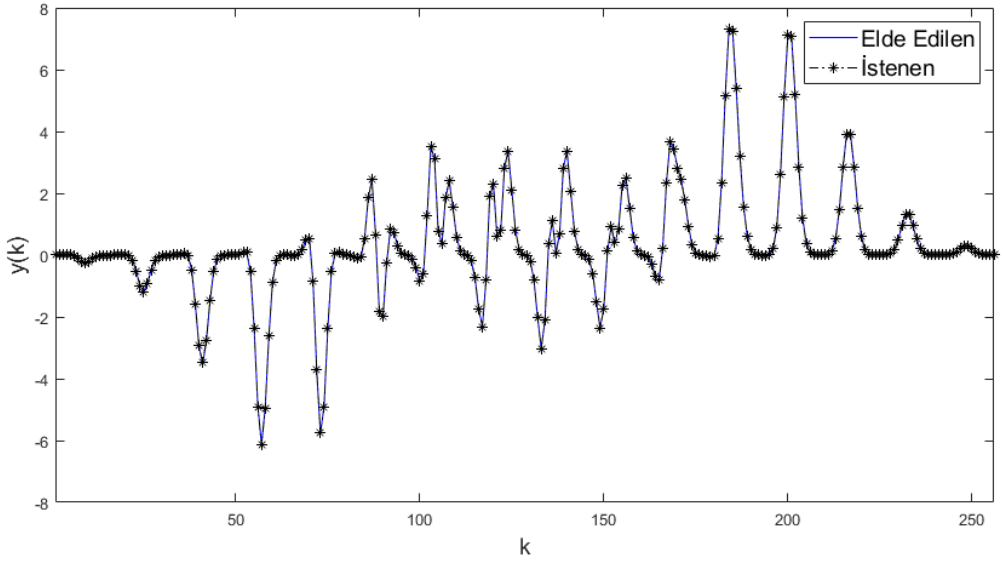
		Eğitim RMSE				**Test RMSE**			
		Ort	En İyi	En Kötü	Std	Ort	En İyi	En Kötü	Std
Peaks Fonksiyonu	ÇK-AÖM	1,84E-02	1,38E-02	2,52E-02	2,85E-03	4,78E-02	2,38E-02	1,33E-01	2,84E-02
	HybRBF-ML-ELM	4,26E-03 (●76,87%)	2,50E-03 (●81,83%)	6,66E-03 (●73,59%)	1,08E-03 (●62,00%)	1,24E-02 (●74,10%)	8,32E-03 (●64,99%)	1,87E-02 (●85,88%)	2,43E-03 (●91,45%)
Griewank Fonksiyonu	ÇK-AÖM	1,83E+00	4,35E-01	6,01E+00	1,19E+00	2,00E+00	7,98E-01	6,09E+00	1,14E+00
	HybRBF-ML-ELM	2,62E-03 **(●99,86%)**	1,35E-03 **(●99,69%)**	4,60E-03 **(●99,92%)**	8,28E-04 **(●99,93%)**	7,66E-03 **(●99,62%)**	5,98E-03 **(●99,25%)**	1,00E-02 **(●99,84%)**	1,18E-03 **(●99,90%)**
Dropwave Fonksiyonu	ÇK-AÖM	1,77E+00	2,83E-01	4,16E+00	8,92E-01	1,80E+00	2,83E-01	4,24E+00	9,11E-01
	HybRBF-ML-ELM	2,88E-02 (●98,37%)	1,34E-02 (●95,24%)	5,83E-02 (●98,60%)	1,13E-02 (●98,73%)	2,86E-02 (●98,41%)	1,33E-02 (●95,31%)	5,79E-02 (●98,64%)	1,12E-02 (●98,77%)
Schaffer2 Fonksiyonu	ÇK-AÖM	3,00E+00	5,26E-01	9,83E+00	2,00E+00	3,06E+00	5,42E-01	1,00E+01	2,04E+00
	HybRBF-ML-ELM	2,40E-02 (●99,20%)	1,48E-02 (●97,18%)	3,32E-02 (●99,66%)	1,30E-02 (●99,35%)	2,42E-02 (●99,21%)	1,52E-02 (●97,21%)	3,32E-02 (●99,67%)	1,27E-02 (●99,38%)

Tablo 6.2 incelendiğinde geliştirilen HybRBF-ML-ELM yapısının ÇK-AÖM yapısına göre her problem için yüzey modellemede daha başarılı olduğu gözlemlenmektedir.

Ortalama RMSE değerleri ele alındığında, ÇK-AÖM yapısı HybRBF-ML-ELM yapısına göre sadece *Peaks* fonksiyonu için yakın metrik değerler vermektedir. Öteki fonksiyonlar için ÇK-AÖM yapısı, HybRBF-ML-ELM yapısına yakın metrik sonuçlar verememektedir. Her kıyaslamalı test fonksiyonu için HybRBF-ML-ELM ve ÇK-AÖM yapılarının eğitim ve test kümeleri için en iyi modelleme başarım grafikleri Şekil 6.5-6.12'de verilmektedir.

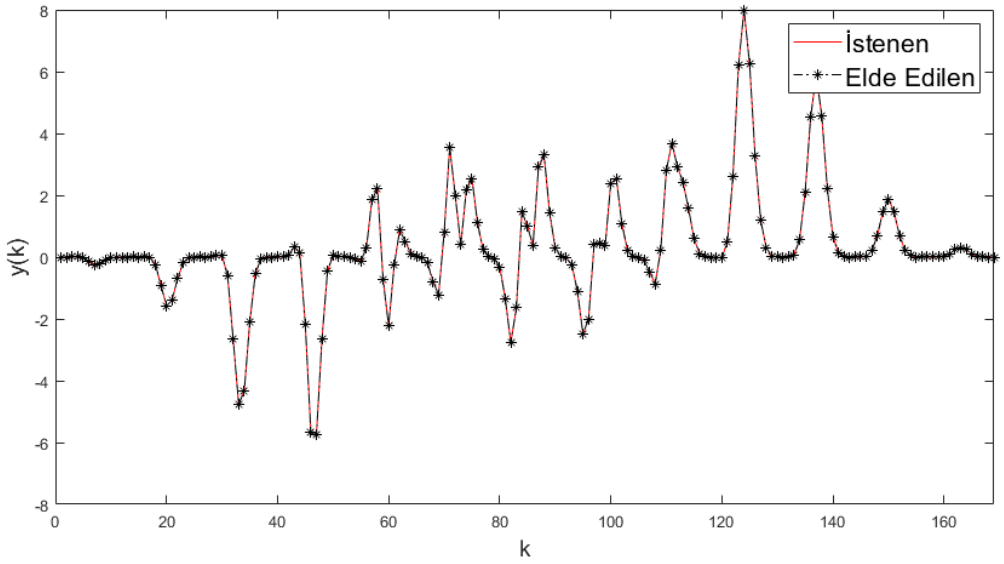


(a)

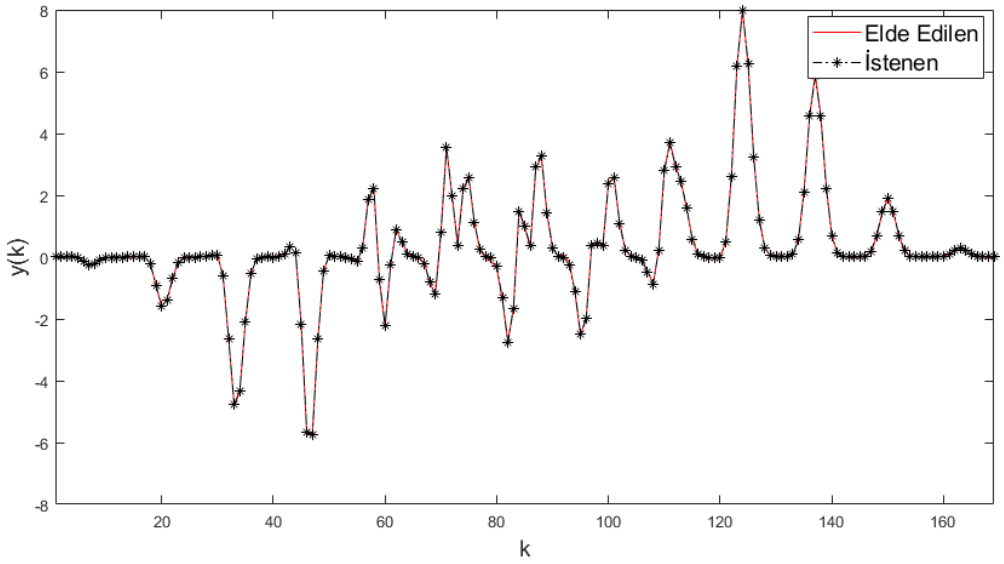


(b)

Şekil 6.5. *Peaks* fonksiyonu için eğitim başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

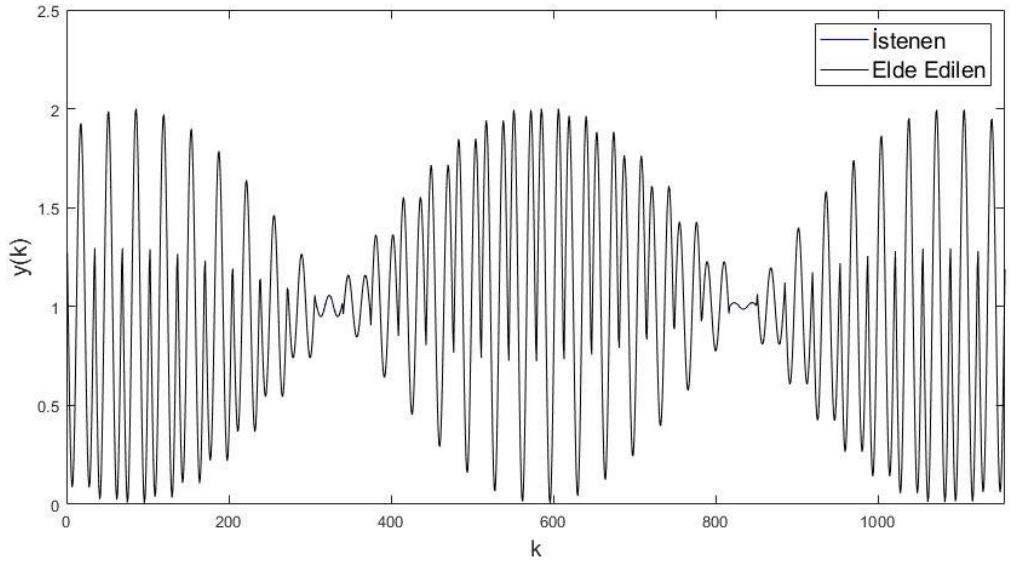


(a)

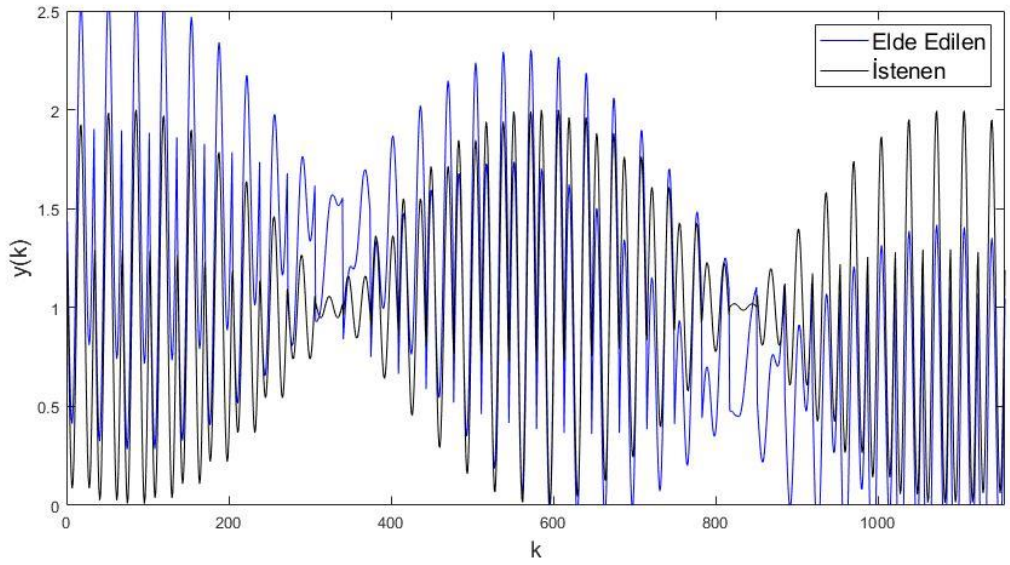


(b)

Şekil 6.6. *Peaks* fonksiyonu için test başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

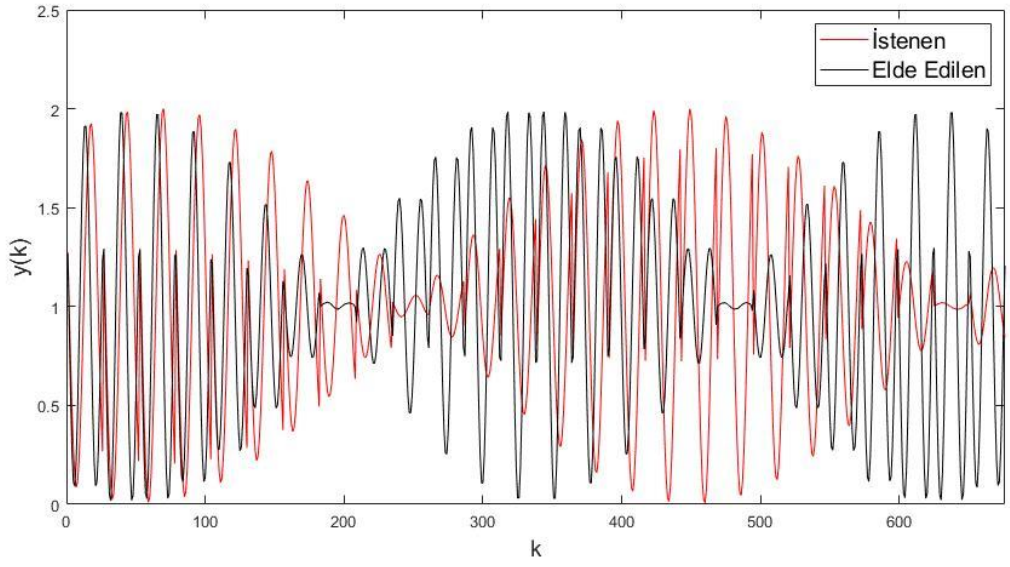


(a)

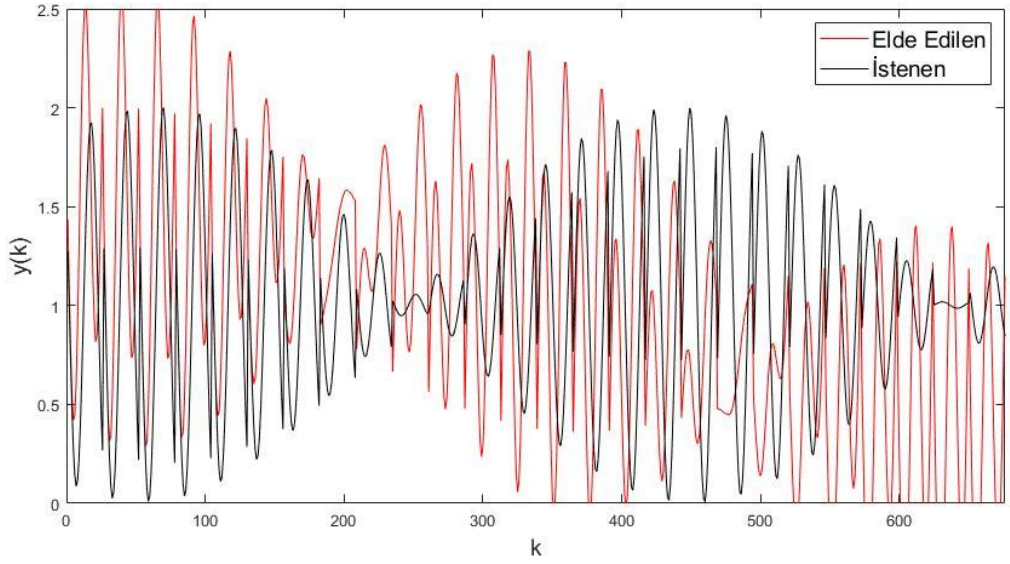


(b)

Şekil 6.7. *Griewank* fonksiyonu için eğitim başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

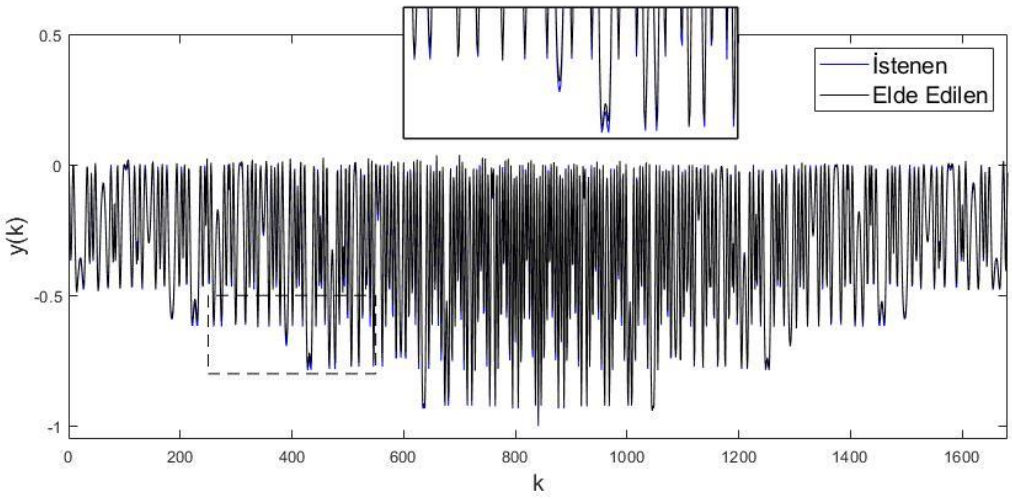


(a)

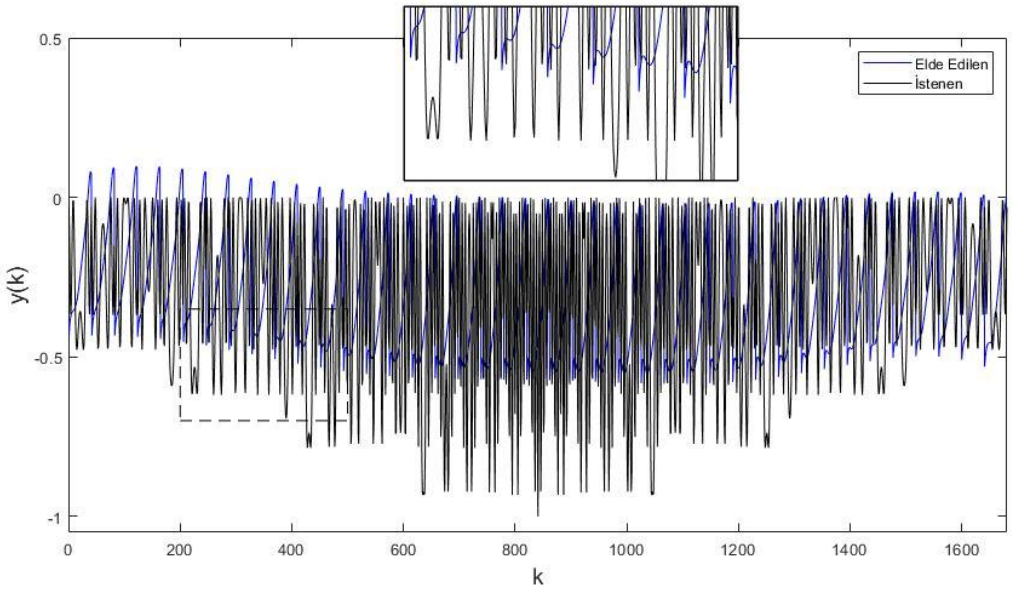


(b)

Şekil 6.8. *Griewank* fonksiyonu için test başarımı; HybRBF-ML-ELM(a) ve ÇK-AÖM (b)

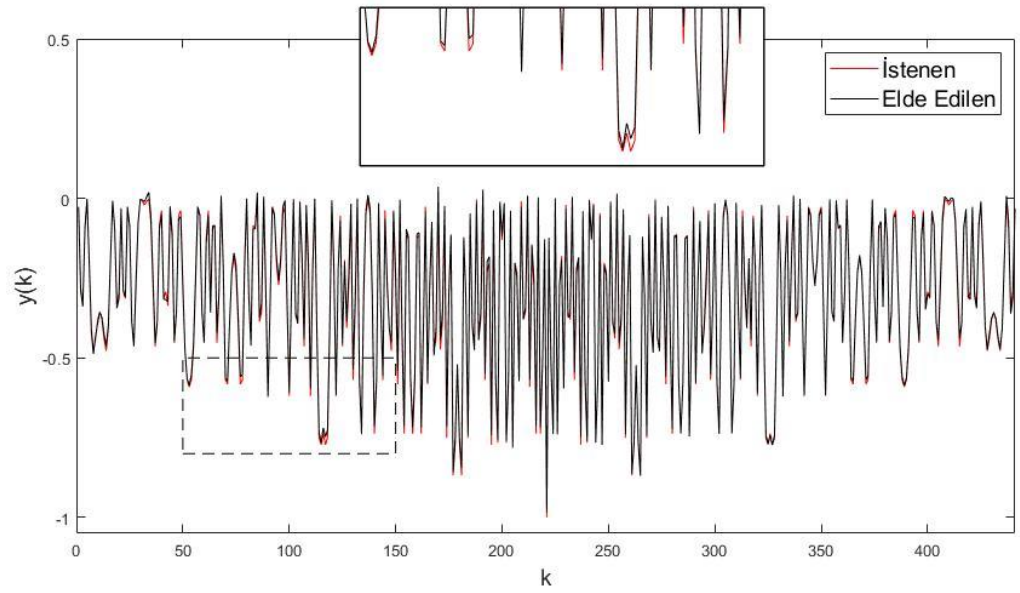


(a)

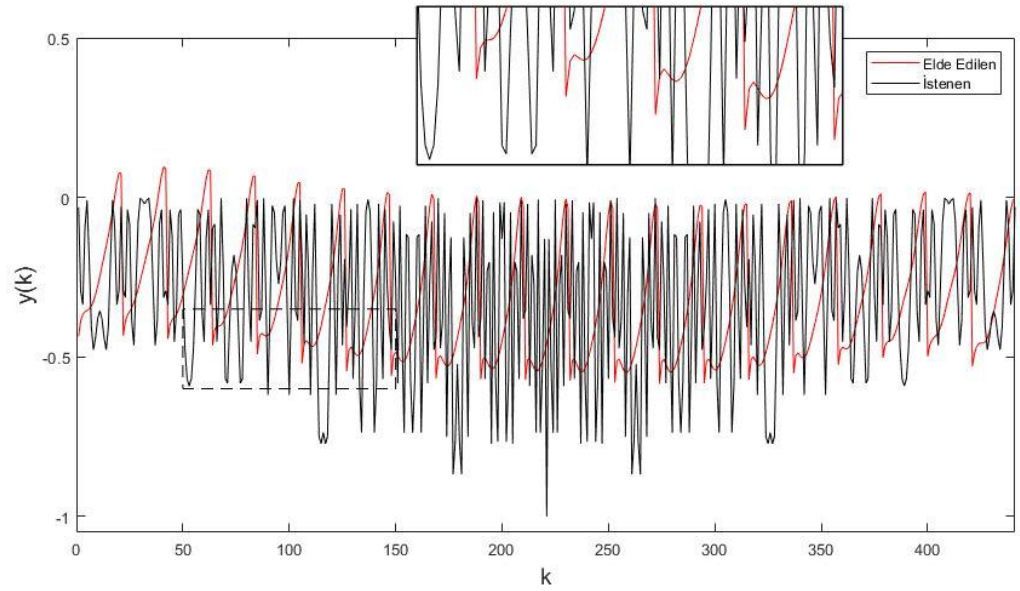


(b)

Şekil 6.9. *Dropwave* fonksiyonu için eğitim başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

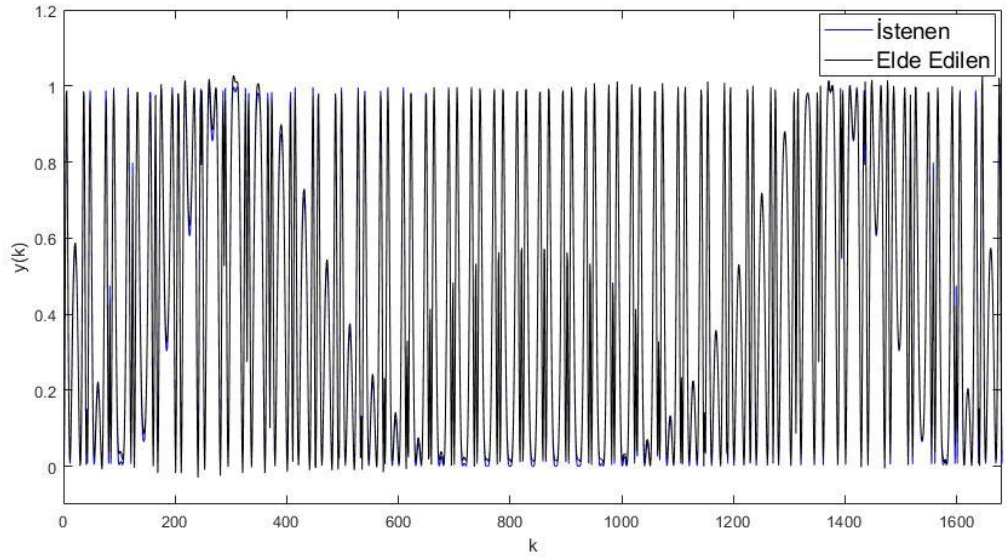


(a)

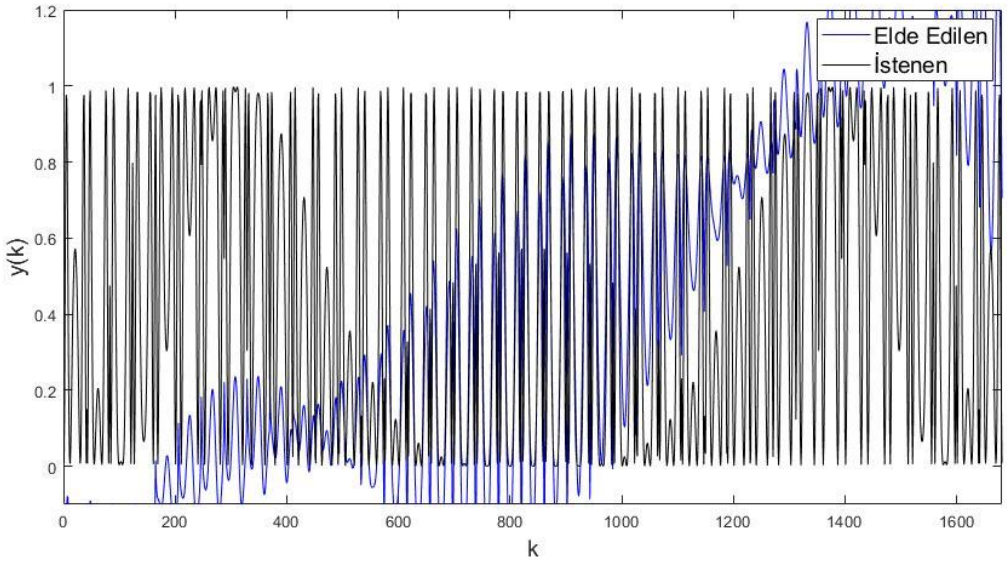


(b)

Şekil 6.10. *Dropwave* fonksiyonu için test başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

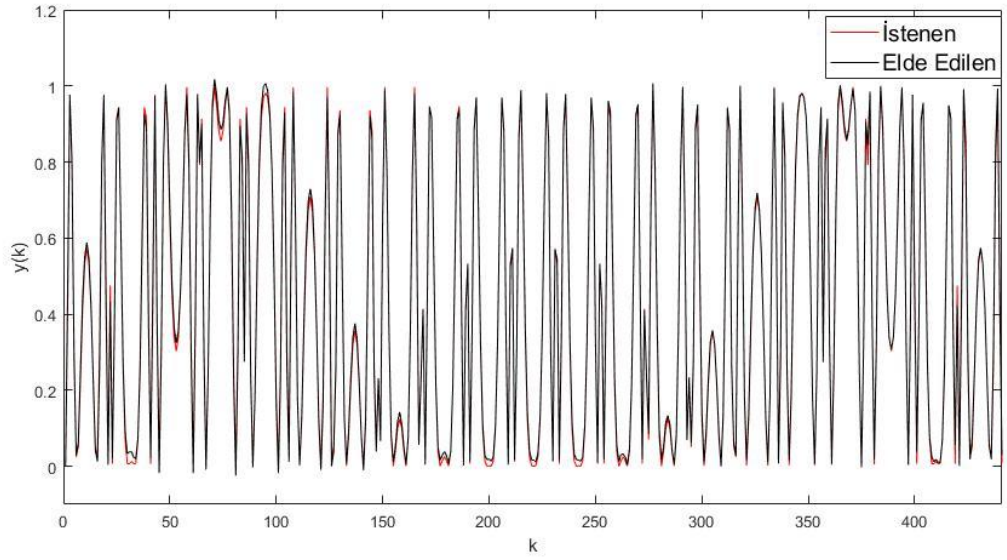


(a)

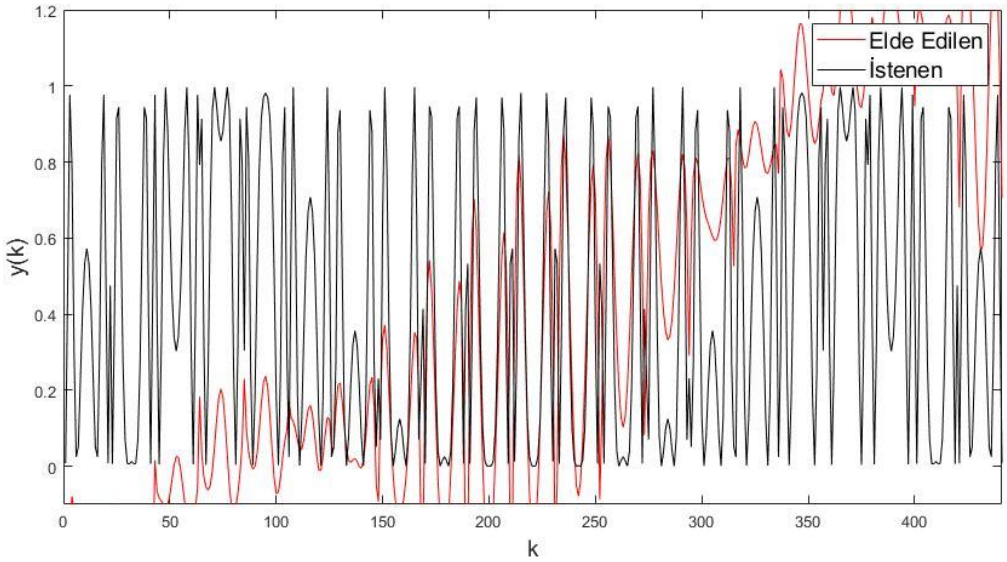


(b)

Şekil 6.11. *Schaffer2* fonksiyonu için eğitim başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)



(a)



(b)

Şekil 6.12. *Schaffer2* fonksiyonu için test başarımı; HybRBF-ML-ELM (a) ve ÇK-AÖM (b)

Tablo 6.2 incelendiğinde, HybRBF-ML-ELM yapısı en iyi modelleme başarımını Griewank fonksiyonu üzerinde elde etmiştir. Yukarıdaki şekillerde görüldüğü üzere, HybRBF-ML-ELM için yüzey modelleme başarımları eğitim ve test aşamaları için oldukça iyidir.

7. SONUÇLAR VE ÖNERİLER

Çalışmamızda çok hızlı öğrenme özelliği ile öne çıkan AÖM'nin klasik AÖM, RTF-AÖM, Meta-AÖM ve ÇK-AÖM gibi mevcut ağ yapıları ve öğrenme algoritmaları incelenmiştir. Anılan bu yapılar Bölüm 3'de anlatılmıştır. İncelenmiş olan yapıların üstün ve sakıncalı yönleri belirlenerek edinilen deneyim çerçevesinde AÖM yapılarında iyileştirme ve yeni karma yapı öne sürülürek toplam 3 ağ yapısı geliştirilmiştir.

Çalışmamız kapsamında ÇK-AÖM mimarisinde iyileştirme yapılmıştır. Bölüm 4'de geliştirilmiş olan yapıdan ayrıntılı olarak bahsedilmiştir. Bölüm 6.1 de de elde edilen deney sonuçlarına yer verilmiştir. Deney sonuçları incelendiğinde; Tablo 6.1 ile ilgili olarak, tasarlanan İÇK-AÖM mimarileri, keskin geçişlere sahip dinamik sistemlerin (DDS 1, DDS 2, DDS 4-7) modellenmesinde ÇK-AÖM'den daha iyi sonuçlar vermektedir. Ancak, İÇK-AÖM mimarileri, yumuşak geçişlere (DDS 3) sahip dinamik sistemlerde ÇK-AÖM'den daha iyi başarımı göstermez. Bu araştırmada tanıtılan yeni girdi temsili hesaplamaları nedeniyle, Tablo 6.1'de verilen sonuçlara göre İÇK-AÖM1, ÇK-AÖM'den daha hızlıdır. Yalnızca DDS 2'de, ÇK-AÖM, İÇK-AÖM1'den daha hızlı bir eğitim süresine sahiptir. Öte yandan, İÇK-AÖM2, tüm durumlarda hem ÇK-AÖM hem de İÇK-AÖM1'den çok daha hızlıdır. Yumuşak geçişlere sahip dinamik sistemlerin modellenmesinde İÇK-AÖM'lerin ÇK-AÖM'den daha başarılı olmayabileceği özetlenebilir; bununla birlikte, İÇK-AÖM'ler keskin geçişlere sahip dinamik sistemleri modellemede daha iyi başarım gösterir. Son olarak, İÇK-AÖM1'in ÇK-AÖM'den daha hızlı olduğunun ve İÇK-AÖM2'nin geleneksel ÇK-AÖM'den çok daha hızlı olduğunun altı çizilebilir.

Sonuç olarak; eğitim süresi bakımından İÇK-AÖM1 yapısı, ÇK-AÖM yapısına göre biraz daha hızlı iken İÇK-AÖM2 yapısı her iki yapıdan da çok daha hızlıdır. Çünkü İÇK-AÖM2 rastgele giriş ağırlık parametreleri atamaları sadece ilk katmanda yapmaktadır. Keskin olmayan geçişlere sahip dinamik sistemlerin modellemesinde, İÇK-AÖM yapıları ÇK-AÖM yapısı kadar başarılı değildir. Öte yandan keskin geçişlere sahip dinamik sistemlerin modellemesinde İÇK-AÖM yapıları ÇK-AÖM'e göre çok daha başarılıdır. Tüm DDSler için ÇK-AÖM en iyi başarımını 100 düğüm kullanıldığında elde etmiştir ve bu hesaplama zamanını dramatik olarak etkilemekte ve yavaşlatmaktadır. DDS 3 haricinde, İÇK-AÖM yapıları tüm DDS modellemelerinde ÇK-AÖM'e göre daha başarılıdır. Özetle, İÇK-AÖM yapıları ÇK-AÖM yapısına göre sadece daha iyi değil aynı zamanda hızlı yapılardır. Bu gerçeklere dayanarak İÇK-AÖM yapıları verimli ve uygulanabilir sistem modelleme mimarisidir denilebilir.

Bu tez kapsamında diğer bir özgün çalışmamız yeni bir karma AÖM yapısının önerilmesidir. Bölüm 5'te de anlatılmış olan HybRBFNN-ML-ELM yapısı engebeli ve keskin değişimlere sahip yüzey modelleme problemlerinde ÇK-AÖM yapısının eksikliklerini gidermiştir. Elde edilen deney sonuçları incelendiğinde; Tablo 6.2 ve Şekiller 6.5-6.12'de HybRBF-ML-ELM yapısının ÇK-AÖM yapısına göre yüzey modelleme alanında daha üstün olduğu görülmektedir. Yüzey modelleme alanında ÇK-AÖM yapısı yaygın olarak kullanılmamaktadır. Ancak, Tablo 6.2 ele alındığında ÇK-AÖM yapısının yüzey modelleme başarımının iyi olmadığı gürülmüştür.

Sonuç olarak, ÇK-AÖM yapısı yüzey modelleme için bu çalışmada geliştirilen HybRBF-ML-ELM yapısından daha kötüdür. ÇK-AÖM yapısı, yüzey modelleme konusunda sadece *Peaks* fonksiyonu için HybRBF-ML-ELM'in metrik değerlerine daha yakın değer vermektedir. Öte yandan, geliştirilen HybRBF-ML-ELM yapısının dört farklı kıyaslama test fonksiyonunda modelleme başarımı iyidir. Açıkça görüldüğü gibi, geliştirilen HybRBF-ML-ELM yapısı, ÇK-AÖM'den daha iyi yüzey modelleme başarımına sahiptir.

Çalışmamız kapsamında önermiş olduğumuz yeni karma yapının iyileştirilmesi mümkündür. HybRBFNN-ML-ELM yapısında otomatik kodlayıcı ve çıkış katmanında ÇK-AÖM yapısı kullanılmıştır. Bu katmanlarda İÇK-AÖM1 ve İÇK-AÖM2 yapıları kullanılarak yeni bir karma ağ yapısı geliştirilebilir. Diğer bir öneri olarak, bu çalışmada geliştirilen AÖM yapılarının sınıflama problemlerinde başarımı ayrıca incelenebilir. Bu yapıda kullanılan düğüm sayıları bu çalışma çerçevesinde eğitim ve test örnek sayılarının arasında olacak şekilde alınmıştır. Düğüm sayıları ile ilgili daha detaylı bir analiz yapılarak kullanılacak düğüm sayısı için bir yaklaşım geliştirilebilir.

KAYNAKÇA

Altan, G. & Kutlu, Y. (2018). Hessenberg Elm Autoencoder Kernel For Deep Learnnng. *Journal of Enggneerrng Technology and Applled Sccences*, 3(2), 141-151

Babuska, R. (2001). Fuzzy Systems, Modeling and Identification. *Electrical Engineering*.

Chen, M., Li, Y., Luo, X., Wang, W., Wang, L. & Zhao, W. (2019). A novel human activity recognition scheme for smart health using multilayer extreme learning machine. *IEEE Internet of Things Journal*, *6*(2), 1410–1418. doi:10.1109/JIOT.2018.2856241

Ezzati Khatab, Z., Hajihoseini Gazestani, A., Ghorashi, S. A. & Ghavami, M. (2021). A fingerprint technique for indoor localization using autoencoder based semi-supervised deep extreme learning machine. *Signal Processing*, *181*. doi:10.1016/j.sigpro.2020.107915

Golestaneh, P., Zekri, M. & Sheikholeslam, F. (2018). Fuzzy wavelet extreme learning machine. *Fuzzy Sets and Systems*, *342*, 90–108. doi:10.1016/j.fss.2017.12.006

Han, H. G., Wang, L. D. & Qiao, J. F. (2014). Hierarchical extreme learning machine for feedforward neural network. *Neurocomputing*, *128*, 128–135. doi:10.1016/j.neucom.2013.01.057

Haykin S. (1999). *Neural Networks : A Comprehensive Foundation* , Prentice Hall.

Huang, G. bin & Slew, C. K. (2004). Extreme learning machine: RBF network case. *ICARCV 2004 8th International Conference on Control, Automation, Robotics and Vision Conference*, C. 2, ss. 1029–1036. doi:10.1109/icarcv.2004.1468985

Huang, G. bin, Zhu, Q. Y. & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat.No.04CH37541)*, C. 2, ss. 985–990. doi:10.1109/IJCNN.2004.1380068

Huang, G. bin, Zhu, Q. Y. & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1–3), 489–501. doi:10.1016/j.neucom.2005.12.126

Huang, G.-B., Liang, N.-Y., Rong, H.-J., Saratchandran, P. & Sundararajan, N. (2005). On-Line Sequential Extreme Learning Machine. *Proceedings of the IASTED International Conference on Computational Intelligence*. s. 232-237.

Jagtap, P., Raut, P., Pillai, G. N., Kazi, F. & Singh, N. M. (2015). Extreme-ANFIS: A novel learning approach for inverse model control of Nonlinear Dynamical Systems. *ICIC 2015 International Conference on Industrial Instrumentation and Control*, s. 718–723. doi:10.1109/IIC.2015.7150836

Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, *23*(3), 665–685. doi:10.1109/21.256541

Juang, C. F. (2002). A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. *IEEE Transactions on Fuzzy Systems*, *10*(2), 155–170. doi:10.1109/91.995118

Kale, G. A. & Karakuzu, C. (2022). Multilayer extreme learning machines and their modeling performance on dynamical systems. *Applied Soft Computing*, *122*. doi:10.1016/j.asoc.2022.108861

Kim, J., Kim, J., Jang, G. J. & Lee, M. (2017). Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. *Neural Networks*, *87*, 109–121. doi:10.1016/j.neunet.2016.12.002

Lan, Y., Soh, Y. C. & Huang, G. bin. (2010a). Constructive hidden nodes selection of extreme learning machine for regression. *Neurocomputing*, *73*(16–18), 3191–3199. doi:10.1016/j.neucom.2010.05.022

Lan, Y., Soh, Y. C. & Huang, G. bin. (2010b). Two-stage extreme learning machine for regression. *Neurocomputing*, *73*(16–18), 3028–3038. doi:10.1016/j.neucom.2010.07.012

Liao, S. & Feng, C. (2014). Meta-ELM: ELM with ELM hidden nodes. *Neurocomputing*, *128*, 81–87. doi:10.1016/j.neucom.2013.01.060

Liu, X., Wang, L., Huang, G. bin, Zhang, J. & Yin, J. (2015). Multiple kernel extreme learning machine. *Neurocomputing*, *149*(Part A), 253–264. doi:10.1016/j.neucom.2013.09.072

Mcculloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, 1522-9602, *The Bulletin Of Mathematical Biophysics*, 5, 115-133. doi:10.1007/BF02478259

Narendra, K. S. & Parthasarathy, K. (1990). Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, *1*(1), 4–27. doi:10.1109/72.80202

Oussar, Y., Rivals, I., Personnaz, L. & Dreyfus, G. (1998). Training wavelet networks for nonlinear dynamic input-output modeling, 0925-2312. *Neurocomputing* 20, 1473-1888. doi:10.1016/S0925-2312(98)00010-1

Phurattanaprapin, K. & Horata, P. (2016). Extended hierarchical extreme learning machine with multilayer perceptron. *JCSSE 2016 13th International Joint Conference on Computer Science and Software Engineering,* 1-5. doi:10.1109/JCSSE.2016.7748874

Qu, B. Y., Lang, B. F., Liang, J. J., Qin, A. K. & Crisalle, O. D. (2016). Two-hidden-layer extreme learning machine for regression and classification. *Neurocomputing*, *175*(PartA), 826–834. doi:10.1016/j.neucom.2015.11.009

Rong, H. J., Huang, G. bin, Sundararajan, N., Saratchandran, P. & Rong, H. J. (2009). Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *39*(4), 1067–1072. doi:10.1109/TSMCB.2008.2010506

Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning representations by back-propagating errors. *Nature 323,* 533–536. doi:https://doi.org/10.1038/323533a0

Sağıroğlu, Ş., Beşdok, E. & Erler, M. (2003). *Mühendislikte Yapay Zeka Uygulamaları 1 / Yapay sinir Ağları*. Kayseri: Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd.Şti.

Sastry, P. S., Santharam, G. & Unnikrishnan, K. P. (1994). Memory Neuron Networks for Identification and Control of Dynamical Systems. *IEEE Transactions on Neural Networks*, *5*(2), 306–319. doi:10.1109/72.279193

Scardapane, S., Comminiello, D., Scarpiniti, M. & Uncini, A. (2015). Online Sequential Extreme Learning Machine with Kernels. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(9), 2214–2220. doi:10.1109/TNNLS.2014.2382094

Tang, J., Deng, C. & Huang, G. bin. (2016). Extreme Learning Machine for Multilayer Perceptron. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(4), 809–821. doi:10.1109/TNNLS.2015.2424995

Tissera, M. D. & McDonnell, M. D. (2016). Deep extreme learning machines: Supervised autoencoding architecture for classification. *Neurocomputing*, *174*, 42–49. doi:10.1016/j.neucom.2015.03.110

Tübitak Bilim ve Teknik Dergisi (2000). Bilim ve Teknik Dergisi 389. Sayısının Eki

Wong, C. M., Vong, C. M., Wong, P. K. & Cao, J. (2018a). Kernel-Based Multilayer Extreme Learning Machines for Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(3), 757–762. doi:10.1109/TNNLS.2016.2636834

Wong, C. M., Vong, C. M., Wong, P. K. & Cao, J. (2018b). Kernel-Based Multilayer Extreme Learning Machines for Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(3), 757–762. doi:10.1109/TNNLS.2016.2636834

Wu, Y. chen & Feng, J. wen. (2018). Development and Application of Artificial Neural Network. *Wireless Personal Communications*, *102*(2), 1645–1656. doi:10.1007/s11277-017-5224-x

Yong, Z., Joo, E. M. & Sundaram, S. (2014). Meta-cognitive fuzzy extreme learning machine. *ICARCV 2014 13th International Conference on Control Automation Robotics and Vision*, 613–618. doi:10.1109/ICARCV.2014.7064374

Yousefi-Azar, M. & McDonnell, M. D. (2017). Semi-supervised convolutional extreme learning machine. *IJCNN 2017 International Joint Conference on Neural Networks*, 1968-1974. doi:10.1109/IJCNN.2017.7966092

Zhang, J., Li, Y., Xiao, W. & Zhang, Z. (2020). Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. *Journal of the Franklin Institute*, *357*(13), 8925–8955. doi:10.1016/j.jfranklin.2020.04.033

Zhang, R., Lan, Y., Huang, G. bin, Xu, Z. ben & Soh, Y. C. (2013). Dynamic extreme learning machine and its approximation capability. *IEEE Transactions on Cybernetics*, *43*(6), 2054–2065. doi:10.1109/TCYB.2013.2239987

Zhang, W. B. ve Ji, H. B. (2013). Fuzzy extreme learning machine for classification. *Electronics Letters*, *49*(7), 447–448. doi:10.1049/el.2012.3642

Zhu, Q. Y., Qin, A. K., Suganthan, P. N. ve Huang, G. bin. (2005). Evolutionary extreme learning machine. *Pattern Recognition*, *38*(10), 1759–1763. doi:10.1016/j.patcog.2005.03.028