



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**KARMA YAPILI YENİ BİR BULANIK-SİNİRSEL AĞ
YAPISININ GELİŞTİRİLMESİ VE BAŞARIMININ
İNCELENMESİ**

Gizem ATAÇ KALE

Yüksek Lisans

Tez Danışmanı

Doç.Dr. Cihan Karakuzu

BİLECİK, 2017

Ref.No: 10158264



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**KARMA YAPILI YENİ BİR BULANIK-SİNİRSEL AĞ
YAPISININ GELİŞTİRİLMESİ VE BAŞARIMININ
İNCELENMESİ**

Gizem ATAÇ KALE

Yüksek Lisans

Tez Danışmanı

Doç.Dr. Cihan Karakuzu

BİLECİK, 2017



BİLECİK ŞEYH EDEBALI UNIVERSITY

Graduate School of Sciences

Department of Computer Engineering

**DEVELOPMENT AND PERFORMANCE ASSESSMENT
OF A NOVEL HYBRID FUZZY-NEURAL NETWORK
STRUCTURE**

Gizem ATAÇ KALE

Master's Thesis

Thesis Advisor

Assoc. Prof. Dr. Cihan Karakuzu

BİLECİK, 2017



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS

JÜRİ ONAY FORMU

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 05/07/2017 tarih ve 34 sayılı kararıyla oluşturulan jüri tarafından 21/07/2017 tarihinde tez savunma sınavı yapılan Gizem ATAÇ KALE' nin "KARMA YAPILI YENİ BİR BULANIK-SİNİRSEL AĞ YAPISININ GELİŞTİRİLMESİ VE BAŞARIMININ İNCELENMESİ" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği / oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI) : Doç. Dr. Cihan KARAKUZU

ÜYE: Doç. Dr. Uğur YÜZGEÇ

ÜYE: Yrd. Doç. Dr. Suhap ŞAHİN

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun
.../ .../ tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

TEŐEKKÖR

Tez alıőmamda benden bilgisini, tecrűbesini ve desteęini esirgemeyen sayın hocam Do. Dr. Cihan KARAKUZU' ya teőekkűrlerimi sunarım.

Ayrıca, maddi ve manevi desteklerini esirgemeyen aileme teőekkűrlerimi sunarım.

Gizem ATA KALE

Temmuz 2017

ÖZET

Bu çalışmada, yeni bir ağ yapısı olarak karma radyal tabanlı fonksiyon sinir ağı (HybRbfNN) geliştirilmiş ve başarımı incelenmiştir. Ağ, 3 gizli, 1 çıkış katmanı olmak üzere toplam 4 katmanlı bir ağ yapısındadır. Ağ özellikle engebeli yüzeye sahip sistemlerin modellenmesi amacıyla geliştirilmiş olup, bu alanda en anlamlı rakipleri olan uyarlamalı ağ tabanlı bulanık sonuç çıkarma sistemi (ANFIS) ağı ve konik kesit fonksiyonlu sinir ağı (CSFNN) ile modelleme başarımı kıyaslanmıştır. Kıyaslama engebeli yüzeye sahip denektaşı bir sistemin modellenmesi problemi üzerinde yapılmıştır. Ağların parametreleri, geliştirilmiş parçacık sürü optimizasyonu (iPSO) kullanılarak belirlenmiştir. HybRbfNN, ANFIS ve CSFNN ağlarının modelleme etkinlikleri, eğitim seyri grafikleri, modelledikleri yüzey ve hata yüzeyleri kullanılarak incelenmiştir. Elde edilen sonuçlar, özellikle engebeli yüzeyler ve keskin geçişlerde bu çalışmada geliştirilen HybRbfNN ağının , ANFIS ve CSFNN ağlarından daha iyi bir öğrenme ve sonucunda da daha üstün bir modelleme başarımı gösterdiğini göstermiştir.

Anahtar Kelimeler

Yeni bir ağ; Karma yapıllı ağ; Sezgisel öğrenme; iPSO

ABSTRACT

In this study, as a novel network structure, Hybrid Radial Based Function Neural Network (HybRbfNN) is developed and its performance is investigated. The network has a total of 4 layers of network structure, with 3 hidden layers and 1 output layer. The network has been developed specifically for modelling systems with uneven surfaces and its modelling performance has been compared with the most significant competitors, adaptive-network based fuzzy inference systems (ANFIS) and conic section function neural network (CSFNN). The comparison is made on modelling of a benchmark system having an uneven surface. The parameters of the networks are trained using improved particle swarm optimization (iPSO). The modeling efficiencies of HybRbfNN, ANFIS and CSFNN are investigated using their training measurement graphs, surfaces they modelled, and error surfaces. The obtained results show that the HybRbfNN network developed in this study, especially on uneven surfaces and sharp transitions, shows a better learning outcome than ANFIS and CSFNN networks and a superior modelling performance over the result.

KeyWords

A novel network; Hybrid structured network; System modeling; Heuristic Learning; iPSO

İÇİNDEKİLER

Sayfa No

JÜRİ ONAY FORMU	
TEŞEKKÜR	
ÖZET.....	I
ABSTRACT	II
İÇİNDEKİLER	III
ŞEKİLLER DİZİNİ	IV
ÇİZELGELER DİZİNİ	V
SİMGELER VE KISALTMALAR DİZİNİ	VI
1. GİRİŞ	1
2. KULLANILAN YÖNTEM VE ALGORİTMALAR	3
2.1. Yapay Sinir Ağları (YSA).....	3
2.1.1. Çok katmanlı algılayıcı yapay sinir ağı (ÇKA).....	5
2.1.2. Radyal tabanlı fonksiyon(RTF)	6
2.2. Parçacık Sürü Optimizasyonu (PSO)	7
3. KULLANILAN MODELLEME YÖNTEMLERİ	10
3.1. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi (ANFIS).....	10
3.1.1. ANFIS mimarisi (Elmas, 2003).....	11
3.2. Konik Kesit Fonksiyonlu Ağlar (CSFNN).....	14
3.3. Karma Radyal Tabanlı Fonksiyon Sinir Ağı (HybRbfNN)	16
3.3.1. HybRbfNN'nin yapısı	16
3.3.2. HybRbfNN'de sezgisel öğrenme	18
4. HybRbfNN AĞININ KARŞILAŞTIRMALI BAŞARIMI	20
4.1. Peaks Fonksiyonu.....	20
4.2. HybRbfNN, ANFIS ve CSFNN ile Karşılaştırılmalı Deney.....	21
5. SONUÇLAR	31
KAYNAKLAR	32
EK-1: Ana Programın Kodları	34
ÖZGEÇMİŞ.....	38

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1. Biyolojik sinir hücre yapısı (Sağiroğlu, vd., 2003).	3
Şekil 2.2. YSA nöron yapısı (Karakuzu,2015).	4
Şekil 2.3. ÇKA' nın ağ yapısı (Karakuzu,2015).	5
Şekil 2.4. RTF ağının genel yapısı (Şenol ve Yıldırım,2004).	7
Şekil 2.5. Parçacık sürü optimizasyonu akış şeması.....	9
Şekil 3.1. Sugeno bulanık modeli (Jang,1993).	11
Şekil 3.2. ANFIS mimarisi (Jang,1993).	12
Şekil 3.3. İki girişli bir çıkışlı dört kurallı ANFIS yapısı.	13
Şekil 3.4. Kesme düzleminin kesme açısına göre oluşturduğu konik kesit düzlemleri (Erkmen, 2007).	14
Şekil 3.5. Düzlemin konileri tepe noktasından kestiği durumlar (Erkmen,2007).	15
Şekil 3.6. N-boyut için konik kesit fonksiyon ağının yapısı (Erkmen, 2007).....	15
Şekil 3.7. HybRbfNN ağ yapısı.	17
Şekil 4.1. Peaks fonksiyonun yüzeyi.	20
Şekil 4.2. iPSO ile ANFIS' in eğitim seyri (Deney No 6), x: iterasyon, y: ortalama karesel hata.....	22
Şekil 4.3. 6 nolu deney ile eğitilen ANFIS'in gerçeklediği giriş-çıkış yüzeyi.....	23
Şekil 4.4. 6 nolu deney ile eğitilen ANFIS için hata yüzeyi.....	23
Şekil 4.5. iPSO ile CSFNN' nin eğitim seyri (Deney No 9), x: iterasyon, y: ortalama karesel hata.....	25
Şekil 4.6. 9 nolu deneyde eğitilen CSFNN'nin gerçeklediği giriş-çıkış yüzeyi.....	26
Şekil 4.7. 9 nolu deneyle eğitilen CSFNN'nin hata yüzeyi.....	27
Şekil 4.8. iPSO ile HybRbfNN'nin eğitim seyri (Deney No 9), x: iterasyon, y: ortalama karesel hata.....	28
Şekil 4.9. 9 nolu deney ile eğitilen HybRbfNN'nin gerçeklediği giriş-çıkış yüzeyi.....	29
Şekil 4.10. 9 nolu deney ile eğitilen HybRbfNN'nin hata yüzeyi.....	29

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 4.1. ANFIS ağının deneysel sonuçları.	22
Çizelge 4.2. CSFNN ağının deneysel sonuçları.	24
Çizelge 4.3. HybRbfNN ağının deneysel sonuçları.	27

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

$\alpha_1, \alpha_2, \alpha_3$: Öğrenme faktörleri
A_1, A_2	: Bulanık küme
a	: Parçacık indisi
B_1, B_2	: Bulanık küme
b	: İterasyon sırası
β_1, β_2	: Koninin eksenini tanımlayan birim vektör
c	: Merkez noktası
d	: Odak ile doğrultman mesafesi
e	: Konik kesitlerde düzlemin eğimi
E	: Eğim parametresi
gb	: Sürünün en iyi konumu
m	: Parçacığın boyutu
n	: Sürünün boyutu
N	: Boyut
o	: Katman çıkışları
p	: Sugeno tip bulanık çıkarımın parametresi
pb	: Parçacığın yerel en iyi konumu
q	: Sugeno tip bulanık çıkarımın parametresi
r	: Yarı çap
rand1,rand2	: Normal dağılımlı rastgele sayı

s	: Sugeno tip bulanık çıkarımın parametresi
sb	: Sürü boyutu
t	: Koninin tepe noktası
U	: Ağ çıkış matrisinin uzunluğu
v	: Parçacığın hızı
W_0	: Bias
W_{ij}	: Birinci katmanın ağırlık parametresi
W_{jk}	: İkinci katmanın ağırlık parametresi
x_1, x_2	: Giriş
y	: Çıkış
ρ	: Parçacık konumu
σ	: Standart sapma
ϕ	: Aktivasyon fonksiyonu
θ	: Kutupsal koordinat sisteminde açı
ω	: Açılma açısı
γ	: HybRbfNN' nin toplam katmanı ağırlık parametresi
γ'	: HybRbfNN' nin NN katmanı ağırlık parametresi
γ''	: HybRbfNN' nin çıkış katmanı ağırlık parametresi
ξ	: Yapı faktörü
λ	: Normal dağıtılmış rastgele sayı vektörü

Kısaltmalar

ANFIS	: Uyarlamalı ađ tabanlı bulanık ıkarım sistemi (Adaptive Neuro Fuzzy Inference System)
BP	: Geriye yayılım algoritması(Back Propagation algorithm)
CSFNN	: Konik kesit fonksiyonlu ađları (Conic Section Funtion Neural Network)
KA	: ok Katmanlı Algılayıcı (Multi LayerPerceptron)
HYBRBFNN	: Hybrid Radial Basis Function Neural Network
IPSO	: Geliştirilmiş paracık sürü optimizasyonu (Improved Particle Swarm Optimization)
NN	: Sinir ađı (Neural Network)
PSO	: Paracık Sürü Optimizasyonu (Particle Swarm Optimization)
RTF	: Radyal Tabanlı Fonksiyon (Radial Basic Function)
YSA	: Yapay Sinir Ađları

1. GİRİŞ

Matematiksel modellere dayanan doğrusal olmayan sistem modelleme, özellikle belirsiz ve tanımlanmamış sistemlerle baş etmek için çok karmaşıktır. Bu sebepten dolayı, bulanık if-then, yapay sinir ağları ve söz konusu iki popüler yöntemi eğitmek/öğretmek için çalışan evrimsel veya sezgisel algoritmaları kullanan bulanık çıkarım sistemi gibi akıllı modelleme yöntemleri birleştirilerek modelleme başarımı arttırılmaya çalışılmıştır.

Bulanık modelleme, birçok mühendislik problemlerinde çok sayıda pratik uygulama bulan Takagi ve Sugeno tarafından ilk kez sistematik olarak tanımlanmıştır (Takagi ve Sugeno, 1985). Bu yaklaşıma Jang'ın uyarlamalı ağ tabanlı bulanık çıkarım sistemi (ANFIS) (Jang, 1993) olarak adlandırılan yeni bulanık modelleme ağ mimarisini önermesi ile çok bilinen hale gelmiştir. ANFIS bulanık modelleme yöntemlerinin en yaygın kullanılanlarından biridir. Beş katmanlı bir ağ yapısını kullanarak Sugeno bulanık çıkarım modeli olarak kullanılır. ANFIS sistematik Takagi-Sugeno bulanık çıkarım sayesinde fonksiyon modelleme yeteneğine sahiptir.

Yapay Sinir Ağları (YSA), nörobiyolojist olan Warren McCulloch ve iktisatçı olan Walter Pitts tarafından tanımlanmıştır (McCulloch ve Pitts, 1943). Çok yaygın olarak bilinen YSA türleri çok katmanlı algılayıcı (ÇKA) ağlar ve radyal tabanlı fonksiyon (RTF) ağlarıdır. ANFIS, ÇKA ve RTF mimarilerinin popüler olması geriye yayılım (Backpropagation - BP) algoritması ile yakından ilişkilidir (Bryson, vd., 1963). Paul Werbos hatanın geriye yayılımı vasıtasıyla, yapay sinir ağlarının eğitim sürecini açıklamış ve bu yöntemi sinir ağlarında ilk defa uygulamıştır (Werbos, 1975). BP söz konusu zeki modelleme yaklaşımlarının eğitilmesi/öğretilmesi için yaygın bir yöntemdir ve danışmanlı öğrenme yöntemi olarak kullanılmaktadır.

Genelde, danışmanlı öğrenme, bazı mühendislik problemlerinde yapay sinir ağlarında türevlenebilir aktivasyon fonksiyonu gerekmesi, türev operasyonlarının karmaşıklığı gibi kısıtlamalardan dolayı uygulanmamaktadır. Bu kısıtlama kendi kendine uyum sağlayan sistemler için önemlidir. Bu sebeple, evrimsel ve sezgisel algoritmalar son zamanlarda önem kazanmıştır. Bu algoritmalar daha hızlı problem

çözmek için klasik yöntemlerin çok yavaş ya da yetersiz olduğu durumlar için tasarlanmıştır.

Sistem modelleme açısından, ÇKA ilgili uzaydaki farklı düzlem sınıflarına sahip örnekleri birbirinden ayırır. Diğer yandan, RTF aynı görev için daire veya elips kullanır. ANFIS ise, sadece beş katmanlı ağ yapısı kullanarak Sugeno bulanık çıkarım modeli kullanır. ÇKA, RTF ve ANFIS tipik özelliklerinden dolayı çoklu minimum/maksimum ve keskin geçişli giriş-çıkış harita yüzeylerini modelleme sisteminde başarısız olabilir.

Bu çalışmada, yukarıda açıklanan yetersizliği gidermek amacıyla yeni bir karma yapay sinir ağı tanımlanmıştır. Bu ağ modeli, dört katmanlı yapısında ÇKA ve RTF'nin karma bir şekilde kullanımını içerir. Böylece ÇKA ve RTF ağlarının üstünlükleri ve bulanık mantık uygun bir yapıda toplanmıştır. Karma radyal tabanlı fonksiyon sinir ağı (Hybrid radial based function neural network, HybRbfNN) olarak adlandırdığımız bu ağın en yakın rakipleri olarak ANFIS ve CSFNN ağları belirlenmiştir. HybRbfNN'in rakipleri ile başarımların kıyaslaması denektaşı bir sistemin modellenmesi sorunu üzerinde yapılmıştır. Bu kıyaslamada bahsi geçen üç ağın parametrelerinin belirlenmesi işlemi (eğitimi) parçacık sürü optimizasyonunu (iPSO) kullanılarak yapılmıştır. iPSO ağların eğitimi için onar kez koşturulmuş, elde edilen sonuçlar ile ağların başarımlarını incelenmiştir.

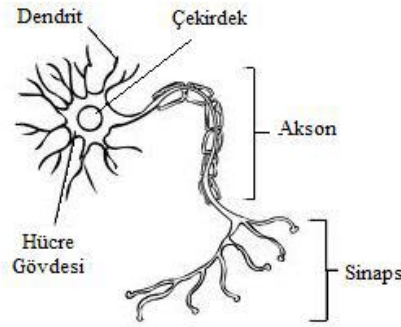
Bir sonraki bölümde, HybRbfNN, ANFIS ve CSFNN ağlarının parametrelerinin eğitilmesi için kullanılan sezgisel öğrenme algoritması ve geliştirilen ağ yapısının temel yapı taşları olan yöntemler anlatılmaktadır. Bölüm 3'te ise geliştirilen ağ (HybRbfNN), geliştirilen ağın rakibi olan ANFIS ve CSFNN ağlarının yapısından bahsedilmiştir. Bölüm 4'de geliştirilen HybRbfNN ağı ile rakipleri ANFIS ve CSFNN ağlarının başarımlarını engelleyen yüzeye sahip olan denektaşı işlevin modellenmesi problemi üzerinde karşılaştırılmıştır. Son olarak, Bölüm 5'de başarımların karşılaştırmaları değerlendirilmiş ve elde edilen sonuçlar vurgulanmıştır.

2. KULLANILAN YÖNTEM VE ALGORİTMALAR

Bu bölümde, geliştirilen HybRbfNN ağını oluşturan temel yapı taşları olan ÇKA ve RTF ağ yapıları tanımlanmış olup bu çalışmada ele alınan ağların eğitiminde kullanılan iPSO algoritmasının temeli olan PSO algoritması anlatılmaktadır.

2.1. Yapay Sinir Ağları (YSA)

YSA insan beyninin sinir hücresinden esinlenerek modellenmiştir ve yapısı insan beyninin sinir hücresine benzetilmiştir (Haykin,1998). Şekil 2.1'de görüldüğü gibi insan beyni sinir hücresi sinapslar, çekirdek, dendrit ve aksondan oluşmaktadır.



Şekil 2.1. Biyolojik sinir hücre yapısı (Sağiroğlu, vd., 2003).

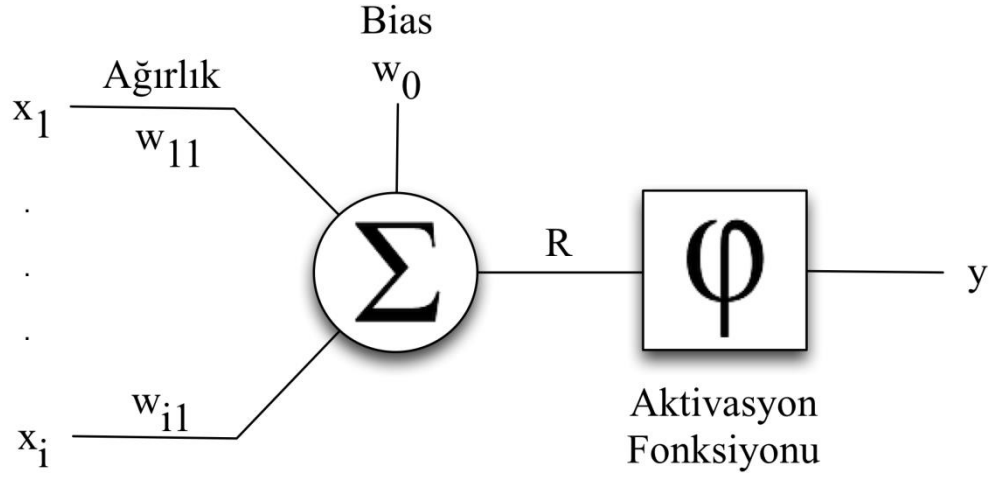
Sinaps: Diğer hücreler ile bağlantıları elektriksel olarak sağlayan bir çeşit boşluktur.

Çekirdek: Diğer hücrelerden gelen sinyalleri işleme tabi tutan ve ait olduğu hücre için bir elektrik sinyali oluşturan birimdir.

Dendrit: Somada üretilen elektrik sinyalleri sinapslar üzerinden diğer hücrelere ileten birimdir.

Akson: Sinapslarla gelen sinyalleri somaya ulaştıran hücreler arası kordondur.

YSA ile insan beyninin sinir hücre benzerliği sinaps, çekirdek, dendrit ve akson sırasıyla ağırlıklar, aktivasyon fonksiyonu, toplama işleci ve çıktıya karşılık gelmektedir. Şekil 2.2'de basit bir YSA nöron modeli verilmiştir.



Şekil 2.2. YSA nöron yapısı (Karakuzu,2015).

Şekil2.2'ye göre giriş ile giriş düğümünden bir sonraki katmandaki hücreye bağlantı ağırlığı ile bağlanır. Bu bağlantı giriş ile bağlantı ağırlık parametresinin çarpımı ile sağlanır. Girişlerin ilgili bağlantı ağırlıkları ile çarpımları ile eşik (bias) değeri toplanıp aktivasyon fonksiyonundan geçirilerek çıkış elde edilir. Eşitlik (2.1) ve (2.2)'de bu işlemlerin formüle edilmiş hali verilmektedir. Eşitlik (2.1)'e göre x_i : giriş verileri, w_{i1} : ağırlık parametreleri, w_0 :bias değeri ve Eşitlik (2.2)'e göre ϕ : aktivasyon fonksiyonunu ifade etmektedir.

$$R = \sum x_i w_i + w_0 \quad (2.1)$$

$$y = \phi(R) \quad (2.2)$$

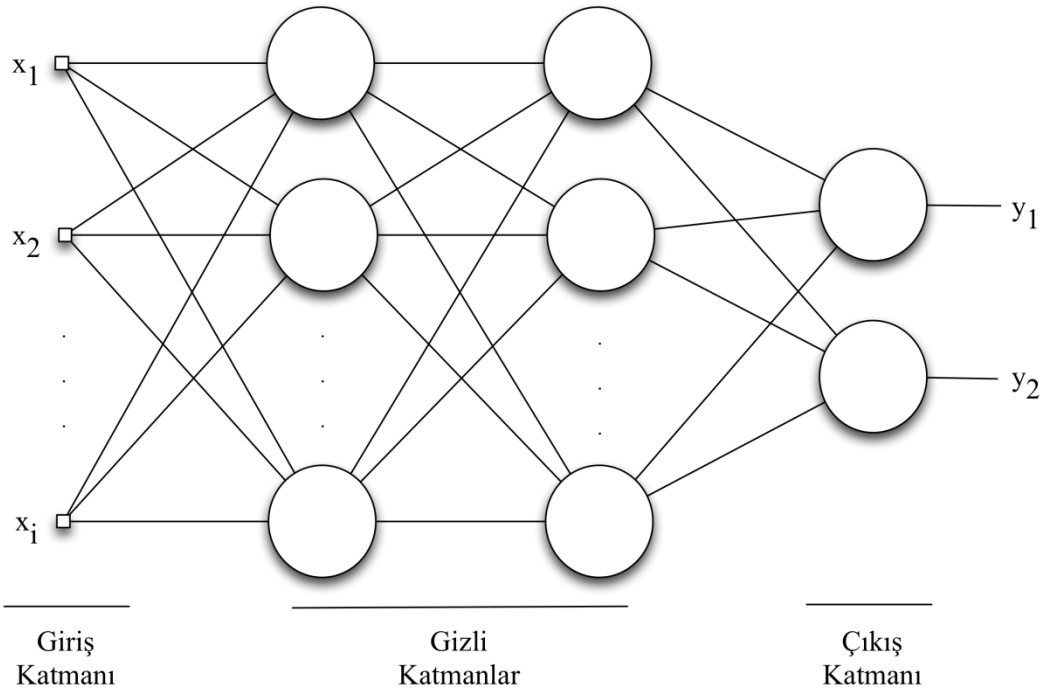
YSA' da genellikle kullanılan aktivasyon fonksiyonları; hiperbolik tanjant sigmoidal transfer fonksiyonu (tansig), doğrusal transfer fonksiyonu (purelin), logaritmik sigmoidal transfer fonksiyonu (logsig), sert-limitli transfer fonksiyonu (hardlim) ve doyunlaşan doğrusal transfer fonksiyonu (satlin)'dur (Karakuzu,2015).

YSA, yapılarına göre ileri beslemeli YSA ve geri beslemeli YSA olarak ikiye ayrılır. İleri beslemeli YSA, girişten çıkışa doğru sıralı katmanlar şeklindedir yani bir nöronun çıkışı kendisinden sonraki katmandaki nöronlara giriş olur ve bir nöron sadece kendisinden önceki nöronun çıkışını giriş olarak alır. Örnek olarak tek katmanlı YSA ve çok katmanlı YSA verilebilir. Geri beslemeli YSA, ileri besleme YSA'dan farklı

olarak bir nöronun çıkışı sadece kendinden sonraki katmanın nöronlarına giriş olmaz, kendi katmanındaki nöronlara veya kendisinden önceki katmanların nöronlarına giriş olur. Örnek olarak Elman ağı ve Hopfield ağı verilebilir (Yalçın,2012).

2.1.1. Çok katmanlı algılayıcı yapay sinir ağı (ÇKA)

YSA'nın ileri beslemeli olan önemli bir sınıfıdır. Tipik olarak çok katmanlı algılayıcı ağ yapısı giriş katmanı, bir veya birden fazla gizli katman ve çıkış katmanından oluşmaktadır (Haykin,1998). Gizli katmanlar ve çıkış katmanında ileri beslemeli YSA'da işlem yaparken aktivasyon fonksiyonu kullanılır. Şekil 2.3'de ÇKA'nın ağ yapısı verilmiştir.



Şekil 2.3. ÇKA'nın ağ yapısı (Karakuzu,2015).

Giriş Katmanı: YSA'ya dışarıdan verilecek olan veriler (girdiler) bu katmana gelir. Burada giriş verisinin sayısı kadar nöron bulunmaktadır ve giriş verisine herhangi bir işlem yapılmadan gizli katmana iletilmektedir. Ayrıca bu nöronlar kendinden sonraki katmandaki bütün nöronlara bağlıdır (Yalçın,2012).

Gizli Katmanlar: Giriş katmanından aldığı verileri işler ve kendinden bir sonraki katmana işlenen verinin sonucunu iletir. Gizli katmanların sayısı ve bu katmanlardaki

nöron sayıları probleme göre değişmektedir ve istenilen sayıda gizli katman ve nöron sayısı olabilir. Fakat fazla gizli katman ve bu katmanlardaki fazla nöron sayısı, hesap karmaşıklığı ve süresinin artışı demektir, öte yandan da karmaşık problemlerin çözümü anlamına da gelebilir (Yalçın,2012).

Çıkış Katmanı: Gizli katmandan gelen işlenen veriyi çıktı olarak dış dünyaya iletir. Çıkış katmanındaki nöron sayısı probleme göre bir ve birden fazla olabilir. Her bir nöron bir önceki katmandaki bütün nöronlara bağlıdır ve çıkış katmanındaki her bir nöronun sadece bir adet çıkış değeri vardır (Yalçın,2012).

2.1.2. Radyal tabanlı fonksiyon(RTF)

RTF ağı çok boyutlu uzaydaki veri kümesinin noktalarını tam ara değerlendirme yapma kökenine sahiptir. Bu RTF ağı evrensel bir yaklaşımdır ve ÇKA'ya karşı popüler alternatif bir yöntemdir. Çünkü RTF daha basit yapıda ve eğitim süreci çok daha hızlıdır. Her iki model de yaygın olarak sınıflama ve fonksiyon modellemede kullanılır. Aşağıda kısaca (Şenol ve Yıldırım, 2004)' den derlenen bilgiler verilmiştir.

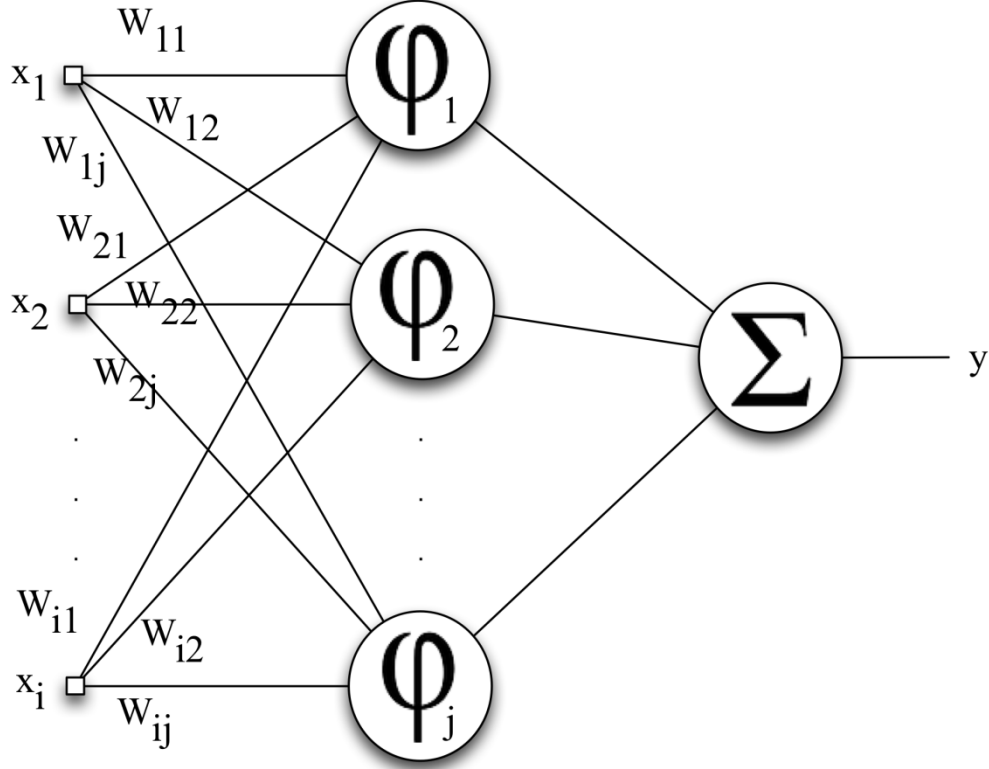
Radyal tabanlı fonksiyon, ileri beslemeli yapay sinir ağının özel halidir. Şekil2.4'de görüldüğü gibi giriş, gizli ve çıkış katmanından oluşmaktadır. Gizli katmandaki her bir düğümde doğrusal olmayan aktivasyon fonksiyonu olarak $\varphi(r)$ RTF kullanılır. Gizli katmandan çıkış katmanına ise doğrusal bir dönüşüm gerçekleşir. RTF ağında eğitilebilecek serbest parametreler; merkez vektörleri, çıkış katmanı ağırlıkları ve radyal fonksiyon genişliğidir. RTF'nin matematiksel ifadesi Eşitlik (2.3)'de verilmiştir.

$$G(X) = \sum_{i=1}^N W_{ij} \varphi_i(x) \quad (2.3)$$

Burada W_{ij} gizli katmandaki i . nöronun çıkış katmanındaki j .nöron arasındaki ağırlık parametresi, $\varphi_i(x)$ aktivasyon fonksiyonunu temsil etmektedir. RTF ağında gizli katman aktivasyon fonksiyonu olarak genellikle Gauss fonksiyonu kullanılır ve Eşitlik (2.4)'de verilmiştir.

$$\varphi(r) = e^{\left(-\frac{\|x-c_i\|^2}{2\sigma_i^2}\right)} \quad (2.4)$$

Eşitlik (2.4)'e göre x giriş vektörünü, c_i merkezi, $\|x - c_i\|$ standart öklid uzaklığını, σ_i de genişliği yani standart sapmayı ifade etmektedir.



Şekil 2.4. RTF ağının genel yapısı (Şenol ve Yıldırım,2004).

2.2. Parçacık Sürü Optimizasyonu (PSO)

Sezgisel optimizasyon algoritmaları, herhangi bir amacı gerçekleştirmek ya da istenilen hedefe varmak için doğal biyolojik davranışlarından esinlenen algoritmalarıdır. Bu algoritmalar yakınsama özelliğine sahip olup kesin çözüm yerine kesin çözüm için en iyi sonucu garanti etmektedir (Karaboğa,2011).

Sezgisel optimizasyon yöntemlerinden biri olan PSO tekniği, 1995'de James Kennedy ve Russel Eberhart tarafından kuş ve balık sürülerinin sosyal davranışlarından ilham alarak doğrusal olmayan problemlerin optimal çözümünü bulmak için ortaya atılmıştır (Kennedy ve Eberhart, 1995).

Doğrusal olmayan probleme çözüm arama, kuş sürülerinin yerini bilmedikleri yiyecekleri aramasına benzemektedir. Hangi kuş aranan yiyeceğe en yakın ise diğer kuşlar da onu takip etmektedir. Arama uzayındaki bir kuş, her tekil çözüm için adlandırılan parçacıktır. Her bir parçacık hareket ettiğinde uygunluk değeri hesaplanan fonksiyona gönderilir ve hareket ettiği konum daha iyi ise parçacığın yeni konumu güncellenmektedir. PSO'nun temelinde bir parçacığın hareket edeceği konum, ne kadar hızla ilerleyeceği ile belirlenmektedir.

Parçacık sürüsü başlangıçta rastgele çözüm ile başlatılır ve daha sonra güncellemelerle en iyi çözüm yolunu bulmaya çalışmaktadır. Parçacığın konumu her iterasyonda, bir önceki paragrafta bahsedildiği gibi güncellenen konum ile bir önceki konum karşılaştırılıp en iyi hangisi ise yeni konumu olmaktadır. En iyi çözümü bulmak için "pb" olarak adlandırılan matris her parçacığın en iyi konumunu ifade etmektedir. Diğer yandan "gb" olarak adlandırılan vektör parçacık sürüsünün içinden en iyi konuma sahip olanı belirtmektedir. Son olarak "v" hız matrisi de her parçacığın konum değişim miktarını göstermektedir. Örneğin n adet parçacıktan oluşan m elemanlı (boyutlu) parçacık olduğunu varsayalım. Bu durumda parçacık matrisi Eşitlik (2.5)'deki gibidir. Ayrıca, pb ve v matrisleri de nxm boyutundadır.

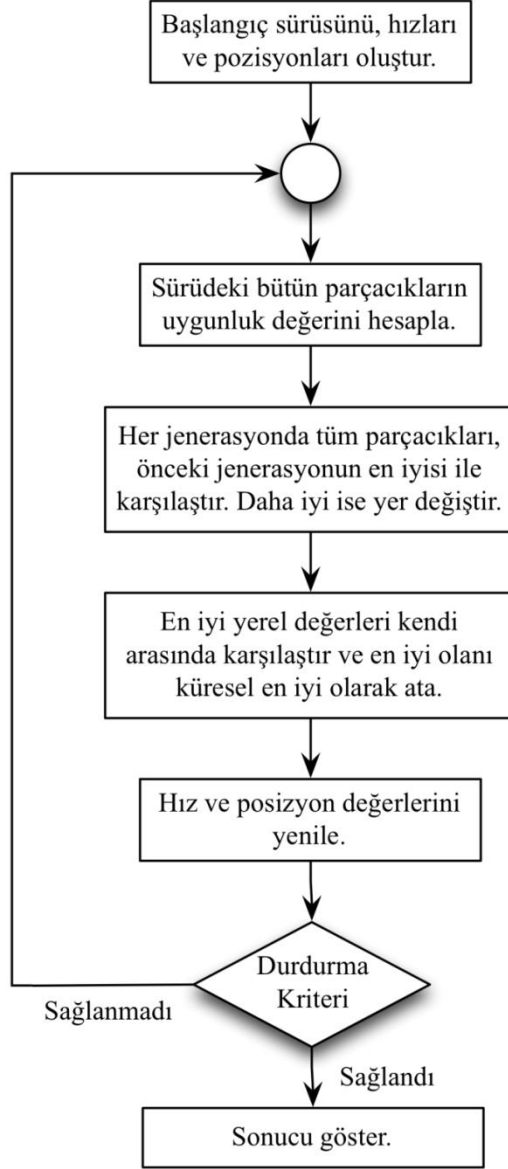
$$\rho = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1m} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nm} \end{bmatrix}_{n \times m} \quad (2.5)$$

Yukarıdaki matrise göre a. parçacık ve pb sırasıyla, $\rho_a = [\rho_{a1}, \rho_{a2}, \dots, \rho_{am}]$, $pb_a = [pb_{a1}, pb_{a2}, \dots, pb_{am}]$ olarak ifade edilmektedir. gb parçacık sürüsü için m boyutlu bir vektördür ve $gb = [gb_1, gb_2, \dots, gb_m]$ şeklinde belirtilmektedir. Son olarak da a. parçacığın hızı $v_a = [v_{a1}, v_{a2}, \dots, v_{am}]$ olarak gösterilmektedir. Parçacığın hızları ve konumu aşağıda verilen Eşitlik (2.6) ve (2.7)'e göre güncellenmektedir (Kennedy ve Eberhart, 1995).

$$v_a^{b+1} = v_a^b + \alpha_1 rand_1^b (pb_a^b - \rho_a^b) + \alpha_2 rand_2^b (gb^b - \rho_a^b) \quad (2.6)$$

$$\rho_a^{b+1} = \rho_a^b + v_a^b \quad (2.7)$$

PSO algoritmasının akış şeması Şekil 2.5'de verilmiştir. Şekilden de görüleceği üzere algoritma durma kriteri olarak belirlenen iterasyon/nesil sayısına ulaştığında ya da en iyi parçacığın uygunluk değeri istenilen sonuca ulaştığında durdurulabilir.



Şekil 2.5. Parçacık sürü optimizasyonu akış şeması.

Bu çalışmada, PSO'nun gelişmiş bir versiyonu olan geliştirilmiş parçacık sürü optimizasyonu (iPSO) kullanılmıştır (Bkz. Bölüm 3.3.2).

3. KULLANILAN MODELLEME YÖNTEMLERİ

Bu bölümde, bu çalışmada geliştirilen HybRbfNN ağ yapısının yanı sıra ona en yakın rakip olarak belirlenen ANFIS ve CSFNN ağ yapılarının yapıları açıklanmaktadır. HybRbfNN ağının modelleme başarımını rakipleri ile kıyaslarken ağların eğitimi/öğrenmesi aşamasında kullanılan sezgisel öğrenme algoritması olan iPSO ise bölümün en sonunda tanımlanmaktadır.

3.1. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi (ANFIS)

İngilizce açılımı 'Adaptive Network Based Fuzzy Inference Systems' olan ağ yapısının kısaltılmış adına 'ANFIS' denilmektedir. Doğrusal olmayan fonksiyonların modellenmesinde ve kaotik zaman serilerinin tahmininde kullanılan bu yöntem, Jang tarafından 1990'lar da geliştirilmiştir (Wikipedia,17.03.2017).

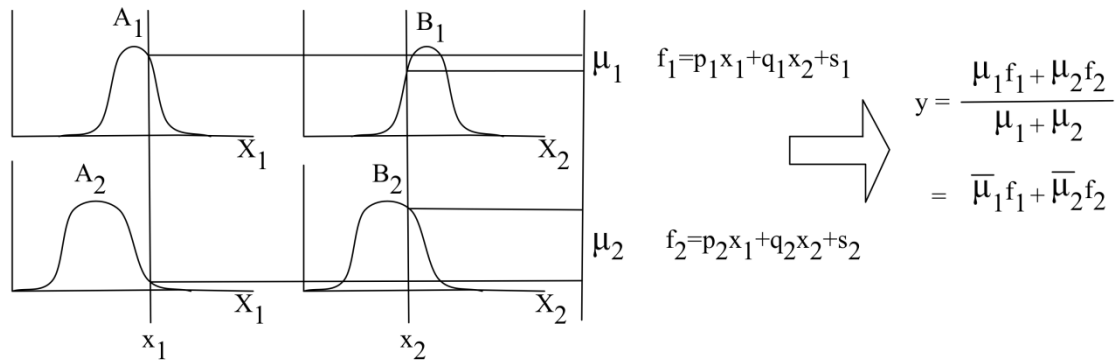
ANFIS, Takagi-Sugeno bulanık çıkarım sistemi ve yapay sinir ağlarına dayalı karma bir yöntemdir. Bulanık çıkarım sisteminin modelleme yeteneği ile sinir ağlarının öğrenme yeteneği birleştirilmiştir. Böylece her iki yöntemin de olumsuz yönleri ortadan kaldırılarak birbirlerini tamamlanmışlardır. Bu yapı ile bulanık sistemin YSA'daki öğrenme yöntemleri ile üyelik fonksiyonu ile kural parametrelerinin en uygun değerinin bulunmasını sağlamıştır. Özetle öğrenme yeteneği olmayan bulanık sisteme öğrenme yeteneği de katılarak modelleme kapasitesi daha da arttırılmıştır. Öğrenme yöntemi olarak genellikle geriye yayılım ya da karma (geriye yayılım+ en küçük kareler) yöntem kullanılmaktadır (Jang, 1991; Jang, 1993; Abraham, 2005). Son yıllarda sezgisel algoritmalar da bu yapının öğrenmesinde kullanıla gelmiştir. Bu çalışmada, öğrenme yöntemi olarak sezgisel öğrenme yöntemlerinden olan geliştirilmiş parçacık sürü optimizasyonu (iPSO) (Çavuşlu, vd., 2012) kullanılmıştır. Bu algoritma bölüm 3.3.2'de anlatılmıştır.

Takagi, Sugeno ve Kang tarafından ileri sürülmüş Takagi-Sugeno'ya da yaygın olarak kullanıldığı şekliyle Sugeno bulanık modeli, bilinen giriş ve çıkış örneklerinden bulanık kuralları elde etmek için sistematik bir yaklaşım ortaya çıkarmıştır. x_1 ve x_2 olmak üzere iki giriş ve y çıkış olarak alınırsa, Sugeno bulanık modeline ait iki kural Eşitlik (3.1) ve (3.2)'de verilmiştir (Karakuzu, 2006; Elmas, 2003).

$$\text{Kural 1: Eğer } x_1 = A_1 \text{ ve } x_2 = B_1 \text{ ise o halde } y_1 = p_1 x_1 + q_1 x_2 + s_1 \quad (3.1)$$

$$\text{Kural 2: Eğer } x_1 = A_2 \text{ ve } x_2 = B_2 \text{ ise o halde } y_2 = p_2 x_1 + q_2 x_2 + s_2 \quad (3.2)$$

Yukarıda verilen kurallarda x_1 ve x_2 giriş değişkenleri, $y_{1,2}$ ilgili kuralın belirlediği çıkış değerleri, $A_{1,2}$ ve $B_{1,2}$ sırasıyla x_1 ve x_2 giriş değişkenlerine ait bulanık kümeleri, $p_{1,2}$, $q_{1,2}$, $s_{1,2} \in \mathbb{R}$ kural/sonuç parametreleridir. İki girişli ve iki kurallı Sugeno tip bulanık çıkarım yöntemi Şekil3.1'de gösterilmiştir (Elmas,2003).



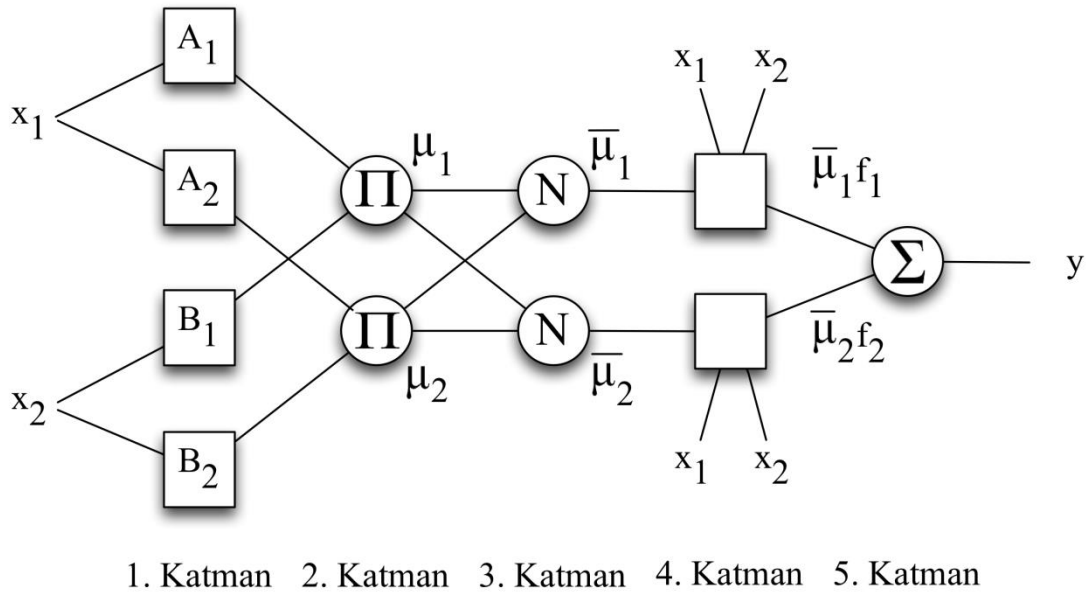
Şekil 3.1.Sugeno bulanık modeli (Jang,1993).

3.1.1. ANFIS mimarisi (Elmas, 2003)

Şekil 3.1'de verilen tipik Sugeno bulanık modelin ANFIS mimarisi Şekil 3.2'de verilmiştir. Ayrıca her katmana ait düğüm ve katmanların işlevlerinden sırasıyla bahsedilmiştir.

Giriş Katmanı: Giriş verilerinin bir sonraki katmana hiç bir işlem yapılmadan aktarıldığı katmandır.

1. Katman: Bulanıklaştırma katmanı olarak adlandırılır. Giriş değerlerini bulanık kümelere ayırmada, Jang'ın ANFIS modelinde, üyelik fonksiyonu olarak Gauss aktivasyon fonksiyonunu kullanılmaktadır. Burada, her bir düğümün çıkışı, giriş değerlerine ve kullanılan üyelik fonksiyonuna bağlı olan üyelik derecelerinden oluşmaktadır. Bu katmandaki her düğüm, birer kare şeklinde gösterilmiştir. Düğümün çıkışı, Eşitlik (3.3) ile belirlenir.



Şekil 3.2. ANFIS mimarisi (Jang,1993).

$$o_h^1 = \mu_{A_h}(x_1)$$

$$o_{h+2}^1 = \mu_{B_h}(x_2) \quad (3.3)$$

$$h = 1,2$$

Eşitlik (3.4)'de c_h ve σ_h çan eğrisi şekilli üyelik işlevinin sırasıyla orta(merkez) noktasını ve standart sapmasını gösterir. Bu parametreler üyelik fonksiyonu veya öncül/şart parametreleri olarak bilinirler ve ağ eğitilirken ayarlanırlar.

$$\mu_{A_h}(x_1) = \frac{1}{1 + \left| \frac{x_1 - c_h}{\sigma_h} \right|^2} \quad (3.4)$$

$$\mu_{B_h}(x_2) = \frac{1}{1 + \left| \frac{x_2 - c_h}{\sigma_h} \right|^2}$$

2. Katman: Her düğüm Π ile ifade edilmektedir ve düğüme giren tüm verilerin çarpımını verir. Bu katmanın çıkışı Eşitlik (3.5)'de verilmiştir.

$$o_h^2 = \mu_h = \mu_{A_h}(x_1) \cdot \mu_{B_h}(x_2) \quad h = 1,2 \quad (3.5)$$

3. *Katman*: Normalleştirme katmanıdır ve her düğüm N ile ifade edilmektedir. Eşitlik (3.6)'da görüldüğü gibi, h. düğüm için h. kuralın normalleştirilmesi, h. kuralının aktifliğinin bütün kuralların aktiflik değerlerinin toplamına bölümü ile bulunur.

$$o_h^3 = \bar{\mu}_h = \frac{\mu_h}{\mu_1 + \mu_2}, \quad h = 1, 2 \quad (3.6)$$

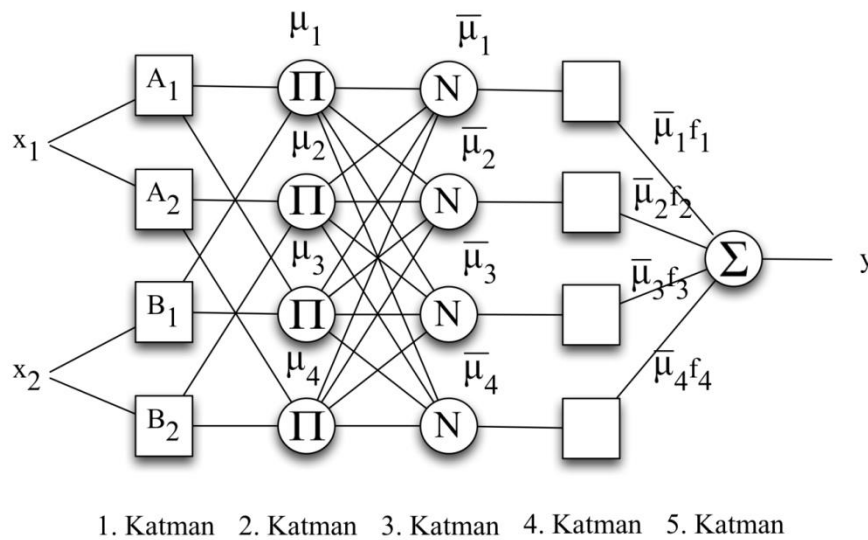
4. *Katman*: Durulama katmanıdır. Her bir h düğümünün normalize aktifliği o kurala ait sonuç polinomu değeri ile Eşitlik (3.7) verildiği gibi çarpılır.

$$o_h^4 = \bar{\mu}_h f_h = \bar{\mu}_h (p_h x_1 + q_h x_2 + s_h) \quad (3.7)$$

5. *Katman*: Toplam katmanıdır, bir düğüm vardır ve Σ ile temsil edilir. Bu katmanda 4. katmanın çıkışından elde edilen değerler toplanır ve elde edilen sonuç (y) sistemin çıkışıdır. Eşitlik (3.8)'de verilmiştir.

$$o_h^5 = y = \sum_h \bar{\mu}_h f_h = \frac{\sum_h [\mu_h f_h]}{\sum_h \mu_h} \quad (3.8)$$

Bu çalışmada 2 girişli, 1 çıkışlı ve 5 katmanlı ANFIS yapısı, 4 kurallı Sugeno bulanık modeli ile kullanılmıştır. Üyelik fonksiyonu olarak da Gauss aktivasyon fonksiyonu kullanılmıştır. Gauss aktivasyon fonksiyonunun merkez ve standart sapma parametrelerine ek olarak Sugeno bulanık modelin kural/sonuç parametreleri, iPSO ile eğitilmiştir. Kullanılan ANFIS'in yapısı Şekil 3.3'de verilmiştir.

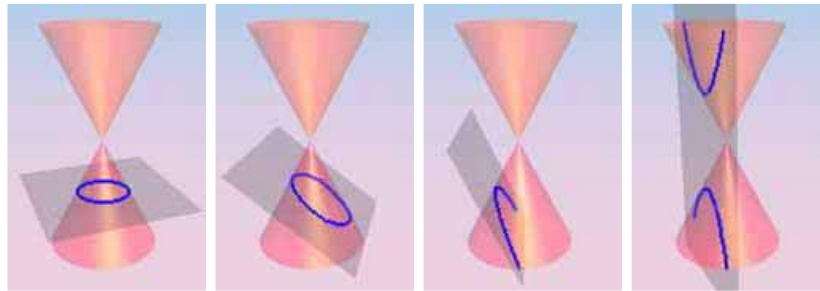


Şekil 3.3. İki girişli bir çıkışlı dört kurallı ANFIS yapısı.

3.2. Konik Kesit Fonksiyonlu Ağlar (CSFNN)

İlk olarak 1994 yılında Dorffner tarafından geliştirilmiştir (Dorffner,1994). ÇKA ve RTF ağının birleşimi yeni bir modeldir. Sınıflama yaparken, ÇKA'nın karar sınırı düz çizgi ve RTF ağının karar sınırı dairedir. Bu iki karar sınırlarının arasında hiperbol, parabol ve elips karar sınırları da vardır. Karar sınırlarının bazıları sınırsız ve sonsuzdur (hiperbol, parabol gibi) fakat bazıları ise sınırlı ve yereldir (daire, elips gibi). Aşağıda CSFNN hakkında (Erkmen, 2007)'den derlenen genel bilgiler verilmiştir.

Bu yapıda temel kuram, tepe noktaları birbirlerine bitişik iki koniden oluşmaktadır. Bu 3 boyutlu koniler ile 2 boyutlu bir düzlemin, belli bir açı ile kesişmesinden, konik kesit düzlemleri oluşur. Bu düzlemler, 2 boyutlu düzlemin kesme açısına göre, Şekil 3.4'te görüldüğü üzere, daire, elips, parabol veya hiperbol konik kesit düzlemleri olabilir.



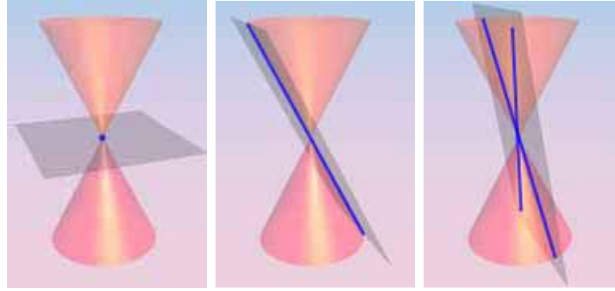
Şekil 3.4. Kesme düzleminin kesme açısına göre oluşturduğu konik kesit düzlemleri (Erkmen, 2007).

Konik kesitler polar koordinat sisteminde Eşitlik (3.9) ile açıklanır.

$$r = \frac{e \cdot d}{1 + e \cdot \cos \theta} \quad (3.9)$$

Odak ile doğrultman arasındaki mesafe d ile ifade edilir. Bu eşitlik $e \geq 0$ için geçerlidir. $0 < e < 1$ için eşitlik bir elipstir, $e = 1$ için eşitlik parabol iken, $e > 1$ için eşitlik bir hiperboldur. $e = 0$ ' da ise eşitlik dairedir.

Şekil 3.4'de gösterilen 4 durum dışında başka özel durumlar da vardır. Örneğin Şekil 3.5'de görüldüğü gibi düzlem konileri sadece tepe noktasından kestiği zaman nokta oluşur. Eğer düzlem tepe noktasıyla beraber konileri teğet olarak keserse bir doğru oluşur. Düzlemin konilerin tepe noktasını kestiği öteki durumlarda ise paralel olmayan iki doğru oluşur.



Şekil 3.5. Düzlemin konileri tepe noktasından kestiği durumlar (Erkmen,2007).

Koninin, konik kesit düzlemlerinde açıklaması Eşitlik (3.10)'da gösterilmektedir.

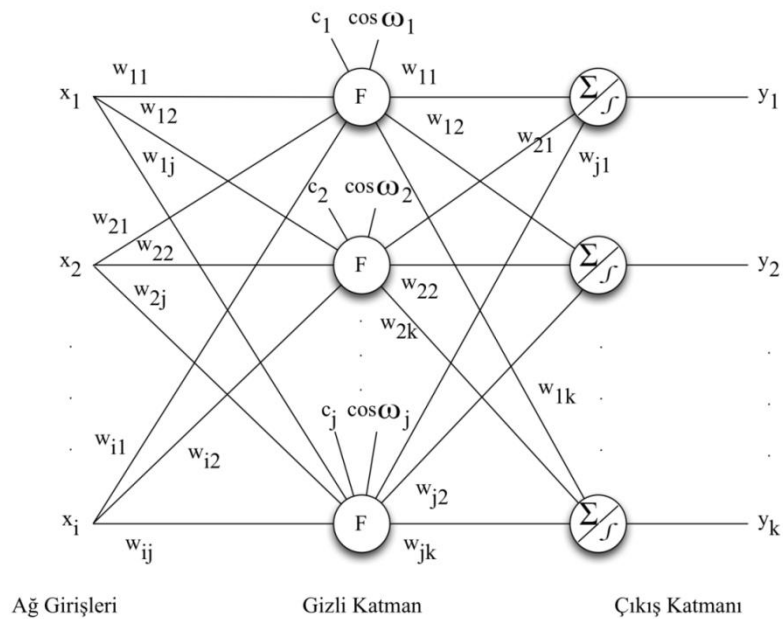
$$(\vec{x} - \vec{t})\vec{\beta} = \cos \omega \|\vec{x} - \vec{t}\| \quad (3.10)$$

$\vec{x} = (x_1, x_2)$, $\vec{t} = (t_1, t_2)$ ve $\vec{\beta} = (\beta_1, \beta_2)$ için, Eşitlik (3.10), Eşitlik (3.11)'deki şekilde yazılabilir ve bu eşitlik konik kesit fonksiyonlu sinir ağının yayılımını tanımlar.

$$(x_1 - t_1)\beta_1 + (x_2 - t_2)\beta_2 = \cos \omega \sqrt{(x_1 - t_1)^2 + (x_2 - t_2)^2} \quad (3.11)$$

N -boyut için Eşitlik (3.11), Eşitlik (3.12)'deki şekilde geliştirilir. Konik kesit fonksiyon ağının yapısı Şekil 3.6'da N -boyut için verilmiştir.

$$\sum_{i=1}^{n+1} (x_i - t_i)\beta_i = \cos \omega \sqrt{\sum_{i=1}^{n+1} (x_i - t_i)^2} \quad (3.12)$$



Şekil 3.6. N -boyut için konik kesit fonksiyon ağının yapısı (Erkmen, 2007).

x noktaları ve t arasındaki uzaklık, açılma açısı $2\omega = 90^\circ$ ise dairenin yarıçapına eşit olduğu için, c merkez koordinat t'nin yerine kullanılabilir. Eşitlik (3.13), (3.14) ve (3.15)'de konik kesit fonksiyonlu sinir ağının ileri yönlü hesaplama formülleri verilmiştir.

$$u_j = \sum_{i=1}^n (x_i - c_{ij})w_{ij} - \cos \omega_j \sqrt{\sum_{i=1}^n (x_i - c_{ij})^2} \quad (3.13)$$

$$o_j = \varphi(u_j) \quad (3.14)$$

$$o_k = u_k = \sum_{j=1}^m o_j w_{jk} \quad (3.15)$$

Eşitlik (3.13) ve (3.15)'den, n ve m sırasıyla giriş, gizli ve çıkış katmanın nöron sayısını ifade etmektedir. c_{ij} merkez koordinatları, w_{ij} giriş ile gizli katman arasındaki ve w_{jk} gizli ile çıkış katman arasındaki ağırlık parametrelerini göstermektedir. o_j ve o_k sırasıyla, gizli ve çıkış katmanın çıkışıdır ve $\varphi(\cdot)$ aktivasyon fonksiyonunu ifade etmektedir.

Bu çalışmada, CSFNN'nin ağ yapısı $\{2,10,1\}$ şeklindedir. Aktivasyon fonksiyonu olarak hiperbolik tanjant sigmoid transfer fonksiyonu (tansig) kullanılmıştır. Ağın parametrelerinin eğitimi sezgisel optimizasyon tekniklerinden iPSO ile yapılmıştır.

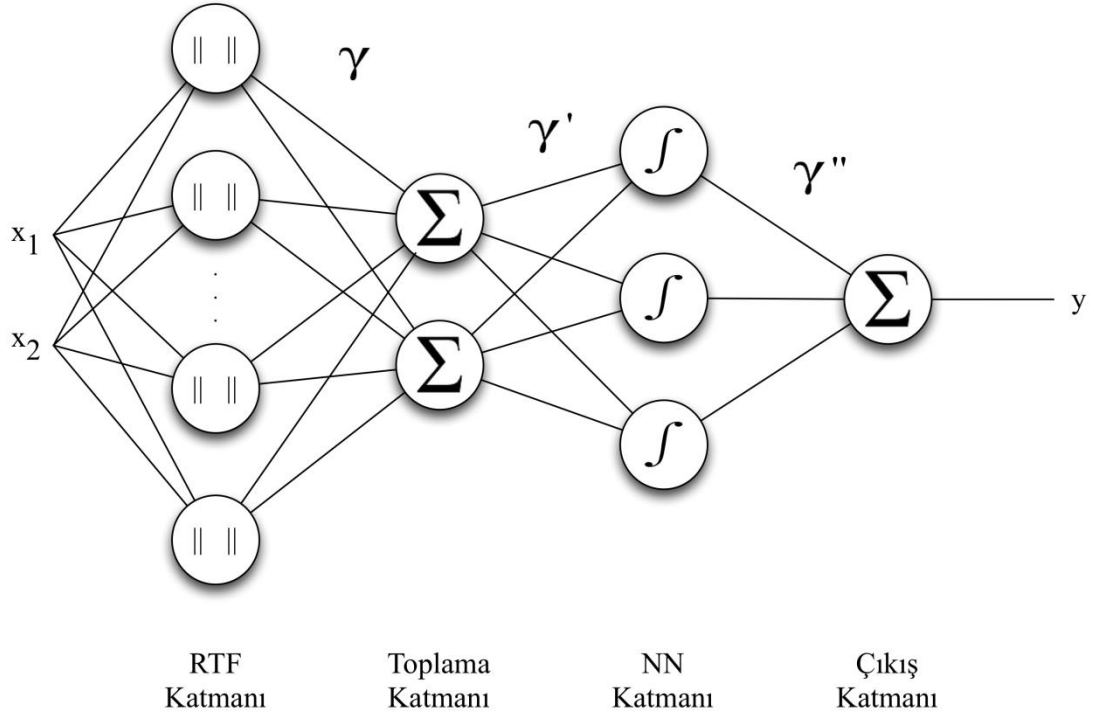
3.3. Karma Radyal Tabanlı Fonksiyon Sinir Ağı (HybRbfNN)

Bu çalışmanın temel amacı, doğrusal olmayan engibeli yüzeye sahip problemlerin çözümü için yeni bir ağ tanımlayıp, bu ağın sistem modellemede gösterdiği başarımının incelenmesidir. Tanımlanan ağ yapısında RTF ile ÇKA ağ yapıları ve bulanık mantık birleştirilmiştir. Geliştirilen yapının adı HybRbfNN'dir (Ataç Kale ve Karakuzu, 2016).

3.3.1. HybRbfNN'nin yapısı

Tasarlanan HybRbfNN, 2 giriş için $\{10,2,3,1\}$ birimli 4 katmandan oluşan yapısı Şekil 3.7'de verilmiştir. x_1 ve x_2 giriş düğümleridir. RTF katmanında, Gauss aktivasyon fonksiyonu kullanılmıştır. RTF katmanının ardından toplam katmanında, toplama işlemi yapılmıştır, γ bu katmanın ağırlık parametreleri matrisidir. NN katmanında, γ'

bu katmanın ağırlık parametreleri matrisidir ve aktivasyon fonksiyonu olarak *tansig* kullanılmıştır. Son olarak, çıkış katmanında, γ'' bu katmanın ağırlık parametrelerini içeren bir vektördür.



Şekil 3.7. HybRbfNN ağ yapısı.

RTF katmanı olan birinci gizli katmanda, her bir nöronun fonksiyonu Eşitlik (3.16)'da tanımlanmıştır. Bu katmanda kullanılan Gauss aktivasyon fonksiyonunun avantajı, keskin değişim ve giriş-çıkış harita yüzeyindeki çoklu minimum/maksimum noktaları için sahip olduğu esnek modelleme yeteneğidir.

$$o_{1i} = e^{-\frac{\|x-c_i\|^2}{\sigma_i^2}} \quad (3.16)$$

Burada, o_{1i} çıkışı ifade etmektedir. RTF katmanının parametreleri her bir nöron için $c_i = [c_{i,1} \ c_{i,2}]$ merkez vektörü ve σ_i standart sapmadır.

Toplam katmanı olarak tanımlanan ikinci gizli katmanda, RTF katmanının çıkışı ile bu katmanın ağırlık parametresi olan γ nın işlem formülü Eşitlik (3.17)'de verilmiştir. o_{2i} bu katmanın çıkışıdır.

$$o_{2_i} = \sum_j o_i \gamma_{ij} \quad (3.17)$$

NN katmanı üçüncü ve son gizli katmandır. Her bir nöronun aktivasyon fonksiyonu Eşitlik (3.18)'de gösterilmiştir. Burada ek olarak, aktivasyon fonksiyonu olan tansig' in eğim parametresi olan E bulunmaktadır. o_3 çıkışı ifade etmektedir.

$$o_3 = \varphi(o_2 \times \gamma') = \frac{2}{(1 + e^{-2E \times (o_2 \times \gamma')})} - 1 \quad (3.18)$$

Son katmanda, Eşitlik (3.19)'da verildiği gibi, γ'' ağırlık parametresi ile bir önceki katmanın çıkışının çarpımıyla oluşan çıkış bulunmaktadır. o_4 çıkışı ifade etmektedir.

$$o_4 = o_3 \times \gamma'' \quad (3.19)$$

3.3.2. HybRbfNN'de sezgisel öğrenme

Sezgisel öğrenme tekniklerinden biri olan iPSO, HybRbfNN'nin ağ yapısındaki γ , γ' , γ'' ağırlık parametreleri ile c merkez vektör, σ standart sapma ve E eğim parametrelerinin eğitiminde kullanılmıştır.

Eşitlik (3.20)'de iPSO kullanılarak yukarıda bahsedilen altı parametrenin en uygun değerinin bulunması için oluşturulan parçacık yapısı verilmiştir. Burada parçacık sürü matrisinin $sb.$ satırı tanımlanmıştır. sb, ρ adındaki sürünün herhangi bir parçacık indisini ifade etmektedir.

$$\rho_{sb} = [c_{1,1} c_{1,2} \dots c_{M,1} c_{M,2} \sigma_1 \dots \sigma_M \gamma_1 \dots \gamma_J \gamma'_1 \dots \gamma'_K E_1 \dots E_L \gamma''_1 \dots \gamma''_L] \quad (3.20)$$

Eşitlik (3.20)'deki M , RTF katmanındaki nöron sayısını temsil eder. J ise, RTF katmanındaki nöron sayısı ile toplam katmanındaki nöron sayısının çarpımını ifade eder. K toplam katmanındaki nöron sayısı ile NN katmanındaki nöron sayısının çarpımıdır. Son olarak L , NN katmanındaki nöron sayısına karşılık gelir.

Eşitlik (3.21) ve (3.22)'de iPSO'nun, parçacığın hızını güncelleme formülü ile konumunu güncelleme formülü verilmiştir (Çavuşlu, vd., 2012). iPSO standart PSO'dan farklı olarak daha detaylı arama yapmakta olup yerel çukurlara takılmayı engellemektedir.

$$v_{sb}(b+1) = \xi[v_{sb}(b) + \alpha_1 \text{rand}_1(pb_{sb}(b) - \rho_{sb}(b)) + \alpha_2 \text{rand}_2(gb_{sb} - \rho_{sb}(b))] + [\alpha_3 \lambda(b)] \quad (3.21)$$

$$\rho_{sb}(b+1) = \rho_{sb}(b) + v_{sb}(b+1) \quad (3.22)$$

Burada, v_{sb} parçacığın hızıdır, yani arama uzayında parçacığın pozisyonunda yapılacak güncelleme değerlerini barındıran vektördür. ξ yapı faktörü ve çoğunlukla 0.76 olarak seçilir. α_1 , α_2 ve α_3 öğrenme oranı sabitleridir. α_3 , 3. bir ilave öğrenme sabiti ve λ ise normal olarak dağıtılmış rastgele sayı vektörüdür. Bu iki parametre, PSO algoritmasını iPSO algoritmasına geliştirerek daha iyi sonuç elde edilmesini sağlar. α_1 ve α_2 2.1 olarak seçilir ve α_3 Eşitlik (3.23)'e göre ayarlanır ve X giriş örneklerinin toplandığı matrisi temsil eder (Çavuşlu, vd., 2012). rand_1 ve rand_2 [0,1) aralığında rastgele dağıtılmış değerlerdir. ρ_{sb} bir önceki paragrafta bahsedilen parçanın yapısıdır. pb her bir parçacığın yerel en iyi konum değeridir ve gb elde edilen yerel en iyi değerlerin içerisinde en iyi konum değerini ifade etmektedir.

$$\alpha_3 \ll \frac{1}{\max\{eig(XX^T)\}} \quad (3.23)$$

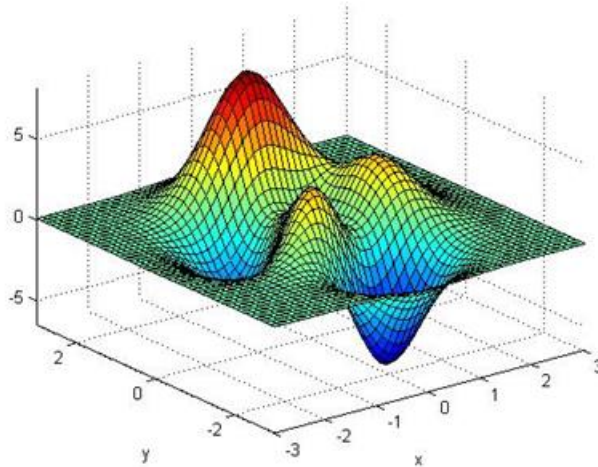
4. HybRbfNN AĞININ KARŞILAŞTIRMALI BAŞARIMI

HybRbfNN ağ yapısı ile rakipleri ANFIS ve CSFNN ağ yapılarının, *peaks* fonksiyonu modellemesinde karşılaştırmalı başarımı bu bölümde incelenmektedir. Bu karşılaştırmalarda, her bir ağ iPSO ile *peaks* fonksiyonunu modellemek üzere eğitilmişlerdir. Bu eğitim her bir ağ için onar kez tekrar edilerek elde edilen veriler kaydedilmiştir. Eğitimler sırasında gözlenen başarımlar ölçütleri baz alınarak karşılaştırma tabloları hazırlanmıştır. Başarımlar değerlendirilmesinde, eğitim ölçüt değerleri göz önüne alınarak bu üç ağ yapısının eğitim seyri, ağın eğitim sonunda bulunan parametrelerle gerçeklediği giriş-çıkış yüzeyi ve hata yüzeyi grafikleri ele alınmıştır.

4.1. Peaks Fonksiyonu

Bu çalışmada modelleme örneği olarak *peaks* fonksiyonu kullanılmıştır. Şekil 4.1'de görüldüğü gibi bu fonksiyon engebeli ve çok tepeli/çukurlu yüzeye sahip olması sebebi ile tercih edilmiştir. Bu fonksiyonun matematiksel denklemi Eşitlik (4.1)'de verilmiştir.

$$y = peaks(x_1, x_2) = 3(1 - x_1)^2 e^{-(x_1^2 - (x_2 + 1)^2)} - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) e^{-(x_1^2 - x_2^2)} - \frac{1}{3} e^{-(x_1 + 1)^2 - x_2^2} \quad (4.1)$$



Şekil 4.1. Peaks fonksiyonun yüzeyi.

4.2. HybRbfNN, ANFIS ve CSFNN ile Karşılaştırılmalı Deney

Bu çalışmada, HybRbfNN, ANFIS ve CSFNN ile ele alınan *peaks* işlevinin modellenmesi amacıyla bu üç ağın iPSO ile eğitimini gerçekleyen bir Matlab programı yazılmıştır. Programda, HybRbfNN, ANFIS ve CSFNN eğitimleri sırayla yapılmaktadır. Eğitim, ağ parametrelerinin rastgele atanan başlangıç değerleri ile başlatılır. Bu üç yapı için, eğitim onar kez 300 iterasyon olarak çalıştırılmıştır. Deneyler Intel(R) Core(TM) i7-4510U CPU 2.00 Ghz 2.60 Ghz işlemcili, 8.00 GB RAM, 64 bit işletim sistemli dizüstü bilgisayarda gerçekleştirilmiştir.

Eşitlik (4.2)'de HybRbfNN, ANFIS ve CSFNN ağ yapılarının eğitim ölçüt değerlerinin hesaplanmasında kullanılan ortalama karesel hata yöntemi gösterilmektedir. Bu eşitliğe göre Z_d istenilen(hedef) çıkış ve Z_a ağın çıkışını ifade etmektedir. Çizelge 4.1-4.3'deki başlangıç, son ve ortalama eğitim ölçüt değerleri bu eşitliğe dayanarak hesaplanmaktadır. Ayrıca, ortalama eğitim ölçüt değeri ise, ilk ve son iterasyon arasındaki tüm iterasyonlardaki en iyi parçacıkların eğitim ölçüt (uygunluk) değerinin toplamalarının, çalıştırılan iterasyon sayısına (300) bölünmesi ile hesaplanmaktadır.

$$mse = \frac{1}{U} \sum_{i=1}^U (Z_d(i) - Z_a(i))^2 \quad (4.2)$$

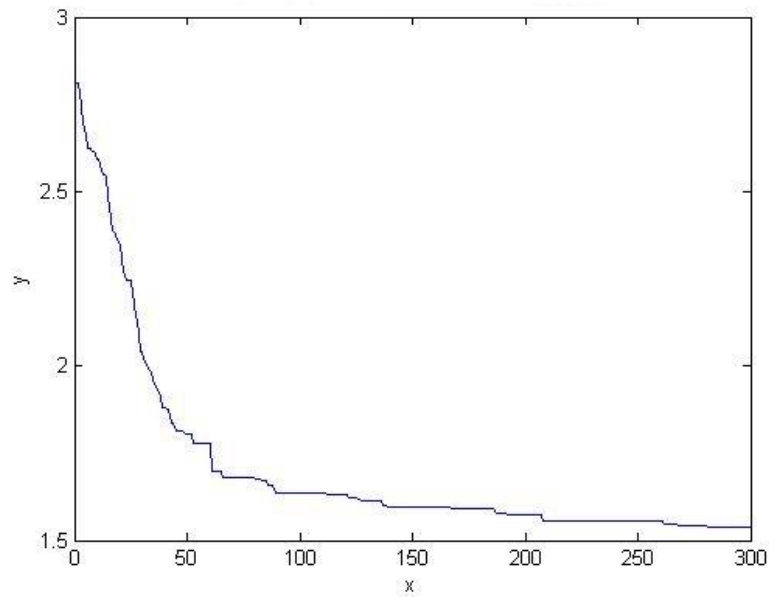
2 girişli, 1 çıkışlı ve toplam 5 katmandan oluşan ANFIS yapısında 4 kurallı Sugeno tipi bulanık çıkarım yöntemi kullanılmıştır. Bu yapı her biri iki nöronlu olmak üzere iki bulanık kümeye sahiptir ve üyelik fonksiyonu olarak Gauss aktivasyon fonksiyonu kullanılmıştır. Merkez noktaları, standart sapmaları ve kural parametreleri iPSO ile ayarlanmıştır. iPSO' nun parçacık boyutu 20, sürü boyutu 60'dır.

10 farklı deneyin başlangıç, son ve ortalama eğitim ölçüt değeri ile eğitimin iterasyon başına süresi saniye cinsinden Çizelge 4.1'de verilmiştir.

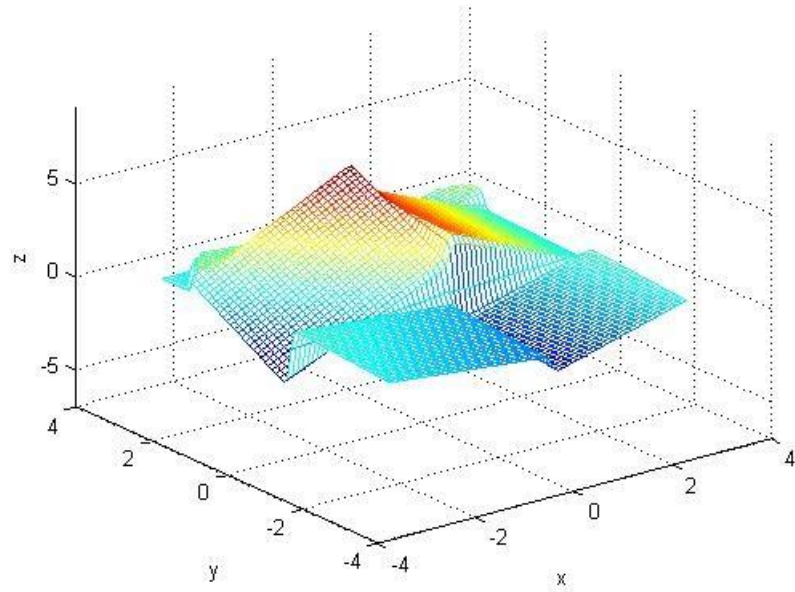
Çizelge 4.1. ANFIS ağının deneysel sonuçları.

Deney	Başlangıç	Son	Ortalama	Süre
1	2.827129	1.930882	1.979433	0.336327
2	2.825390	1.976672	2.031466	0.338134
3	2.811543	1.952845	2.007490	0.360630
4	2.808167	1.971097	2.037852	0.378794
5	2.816552	1.713699	1.871947	0.392096
6	2.811543	1.537585	1.703031	0.346804
7	2.793741	1.977281	2.103124	0.327647
8	2.810334	2.313250	2.348258	0.324962
9	2.808167	1.765834	1.940983	0.364344
10	2.793741	2.360576	2.407553	0.446337

Çizelge 4.1'e göre ANFIS'in *peaks* fonksiyonunun en başarılı modellenmesi, sıfıra en yakın son ölçüt değerini veren 1.537585 ile deney 6'dır. Ayrıca ANFIS yapısının sezgisel öğrenme ile eğitilecek parametre sayısı az olduğu için, iterasyon başına süresi diğer iki yapıdan da kısa sürmektedir. Şekil 4.2- 4.4'de 6. deneye göre ANFIS'in eğitim başarımı verilmiştir.

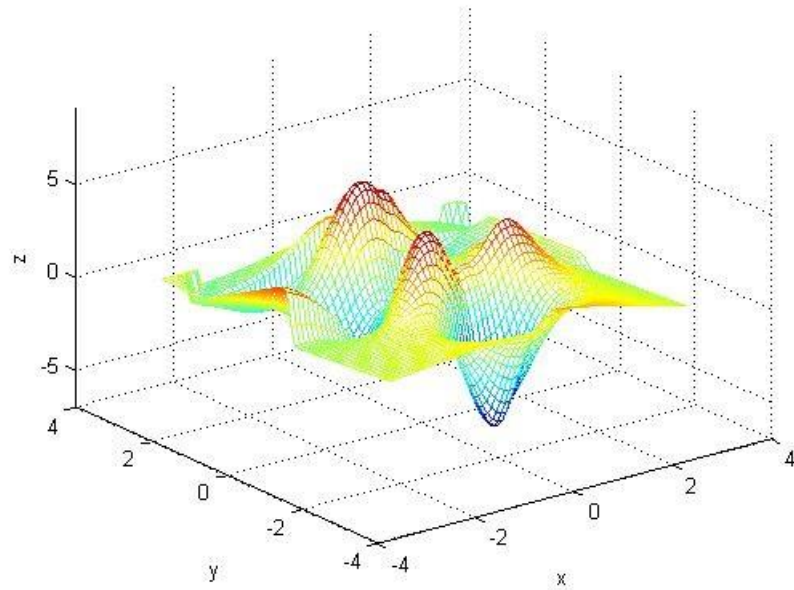


Şekil 4.2. iPSO ile ANFIS' in eğitim seyri (Deney No 6), x: iterasyon, y: ortalama karesel hata.



Şekil 4.3. 6 nolu deney ile eğitilen ANFIS'in gerçeklediği giriş-çıkış yüzeyi.

Şekil 4.3'den görüldüğü gibi, maalesef, çıkış yüzeyinin Şekil 4.1'deki *peaks* fonksiyonun yüzeyi ile hiç bir benzerliği yoktur. Çizelge 4.1'den de görüldüğü gibi, ANFIS'in iterasyon başına süresi diğer iki yapıdan kısa sürmektedir, ancak engebeli yüzeyleri modelleme yeteneği Şekil 4.3 ve Şekil 4.4'ten görüldüğü üzere kısıtlıdır.



Şekil 4.4. 6 nolu deney ile eğitilen ANFIS için hata yüzeyi.

Şekil 4.4'de görüldüğü üzere, hata yüzeyi sıfır olması gerekirken neredeyse *peaks* fonksiyonunun yüzeyine benzemektedir. Şekil 4.2, 4.3 ve 4.4'de görüldüğü gibi çıkış yüzeyi istenilen *peaks* fonksiyonun yüzeyini modelleyememiştir. Zaten eğitim seyrinden de anlaşıldığı gibi sıfıra yakınsaması gerekirken en iyi değer olarak 1.537585 değerine yakınsayabilmiştir. Sonuç olarak ANFIS *peaks* fonksiyonunu modellemede başarısız olmuştur.

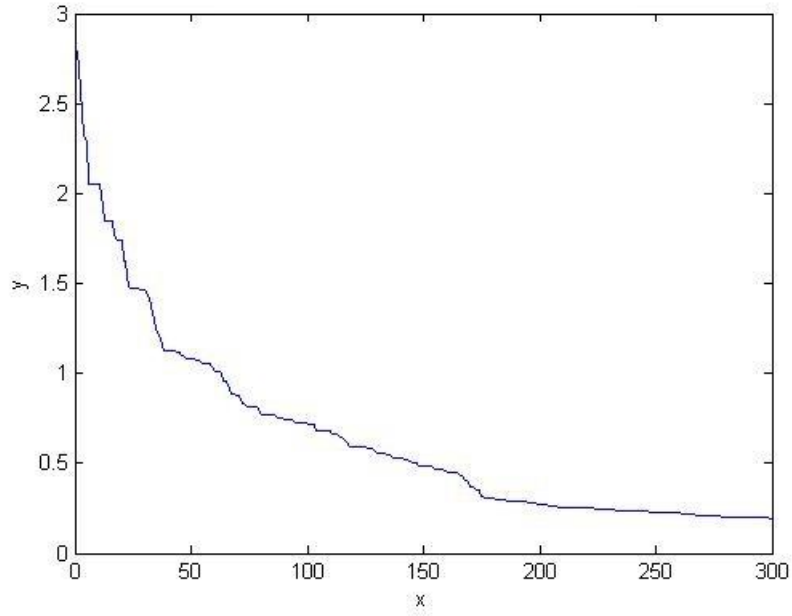
CSFNN {2,10,1} şeklinde 2 giriş nöronuna sahip giriş katmanı, 10 adet nörona sahip bir gizli katman ve bir nörona sahip çıkış katmanı olmak üzere toplam üç katmandan oluşmaktadır. CSFNN'deki açı, merkezler, ağırlık parametreleri iPSO ile eğitilmiştir. Parçacık boyutu 60, sürü boyutu 180'dir ve aktivasyon fonksiyonu olarak da hiperbolik tanjant sigmoid transfer fonksiyonu (tansig) kullanılmıştır. Çizelge 4.2'de CSFNN'nin 10 farklı deney sonuçlarının ilk, son ve ortalama eğitim ölçüt değerleri ve eğitimin iterasyon başına süresi saniye cinsinden verilmiştir.

Çizelge 4.2. CSFNN ağının deneysel sonuçları.

Deney	Başlangıç	Son	Ortalama	Süre
1	2.833204	0.267446	0.679962	4.142215
2	2.801306	0.373145	0.788782	4.171395
3	2.742250	0.418212	0.814468	4.671874
4	2.787611	0.293919	0.733657	4.406483
5	2.775565	0.216566	0.632128	4.084161
6	2.742250	0.228029	0.784263	4.256378
7	2.744816	0.266347	0.706071	4.298749
8	2.773193	0.336636	0.771277	3.962561
9	2.787611	0.194299	0.64332	4.216016
10	2.744816	0.228649	0.625448	4.043661

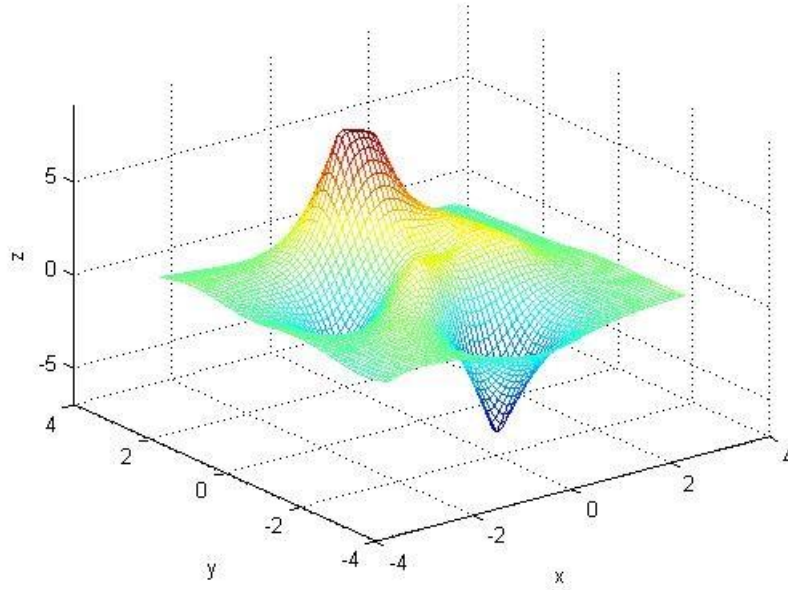
Çizelge 4.2'e göre en iyi eğitim 0.194299 son ölçüt değeri ile 9. deneyde elde edilmiştir. İstenilen, eğitim ölçüt değerinin sıfıra yakınsamasıdır ve CSFNN sıfıra çok yakınsanmasa bile ANFIS ile karşılaştırıldığında daha iyi sonuç vermiştir. Fakat CSFNN, ÇKA ve RTF yayılım kurallarını içerip, bir koninin analitik denklemini de kullanmasına rağmen istenilen sonuca ulaşamamıştır. Bir diğer sakınca ise işlem

zamanıdır çünkü iterasyon başına süresi en az 3.962561 sn'dir. Şekil 4.5-4.7'de CSFNN yapısının iPSO ile eğitim başarımı verilmiştir. Şekil 4.5'de eğitim ölçüt değerinin 300 iterasyon boyunca değişimi görülmektedir.



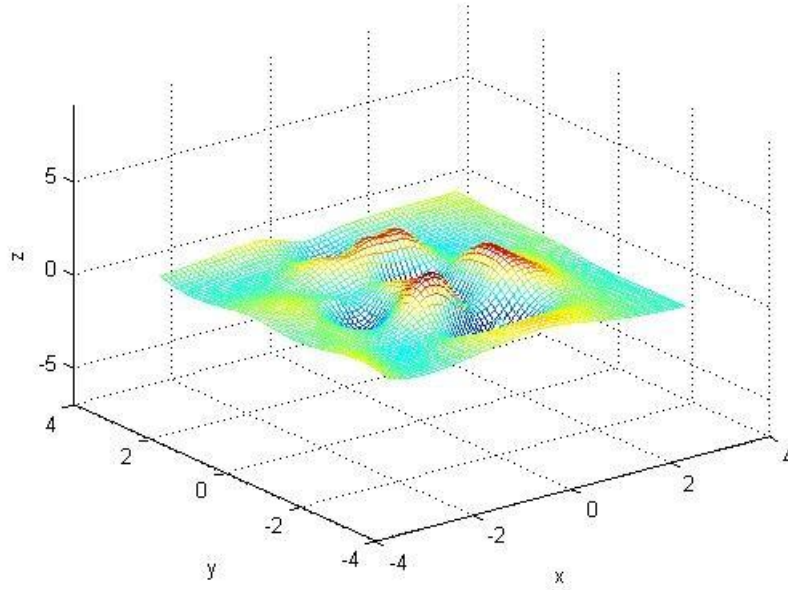
Şekil 4.5. iPSO ile CSFNN' nin eğitim seyri (Deney No 9), x: iterasyon, y: ortalama karesel hata.

Şekil 4.6'da 9 nolu deneyde eğitilen CSFNN'nin gerçeklediği giriş-çıkış yüzeyi verilmiştir. Çıkış yüzeyinin ilk bakışta Şekil 4.1'deki *peaks* fonksiyonun yüzeyine benzediği görülmektedir. Fakat CSFNN giriş-çıkış yüzeyi tepe ve çukur noktalarını kısmen bulmuş olsa bile tepe ve çukur noktaları *peaks* fonksiyonun yüzeyine göre daha keskindir.



Şekil 4.6. 9 nolu deneyde eğitilen CSFNN'nin gerçeklediği giriş-çıkış yüzeyi.

Şekil 4.7'de 9 nolu deney ile eğitilen CSFNN ağ yapısının hata yüzeyi verilmiştir. İstenilen, hata yüzeyini sıfır değerinde düz bir düzlem olmasıdır, fakat görüldüğü gibi hata yüzeyi engebelidir, her kartezyen giriş noktasında sıfır değerinde değildir. Sonuç olarak Şekil 4.5-4.7'de görüldüğü gibi, CSFNN yapısı *peaks* fonksiyonunun yüzeyini kısmen modellemiştir, fakat özellikle tepe ve çukur noktalarında başarımı iyi değildir.



Şekil 4.7. 9 nolu deneyle eğitilen CSFNN'nin hata yüzeyi.

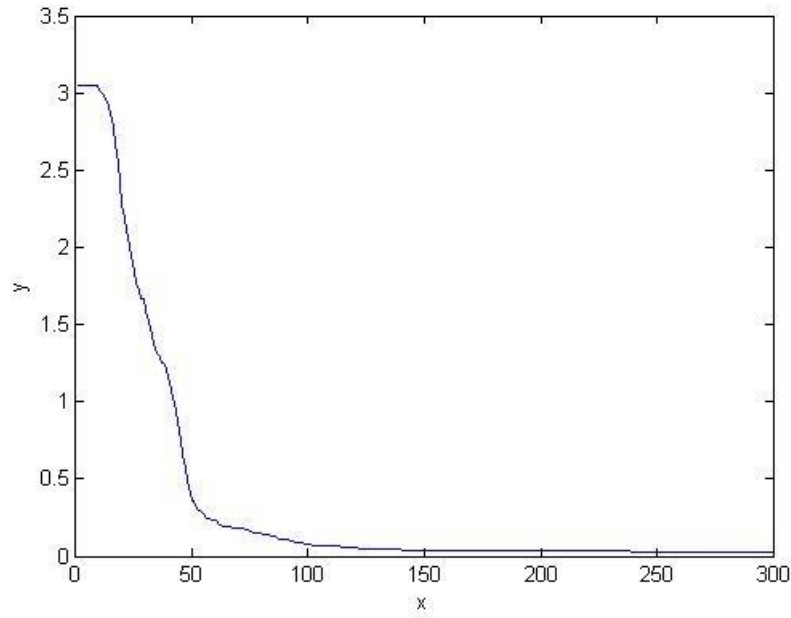
HybRbfNN, bölüm 3.3.1'de anlatıldığı gibi, $\{10,2,3,1\}$ şeklinde ağ yapısına sahiptir ve bahsedildiği gibi iPSO ile ağ parametreleri eğitilmiştir. Burada parçacık boyutu 62 ve sürü boyutu 186'dır. Çizelge 4.3'de HybRbfNN ağ yapısının 10 farklı deney sonucunun eğitim ölçüt değerinin ilk, son ve ortalama değeri ve iterasyon başına süresi saniye cinsinden verilmiştir.

Çizelge 4.3. HybRbfNN ağının deneysel sonuçları.

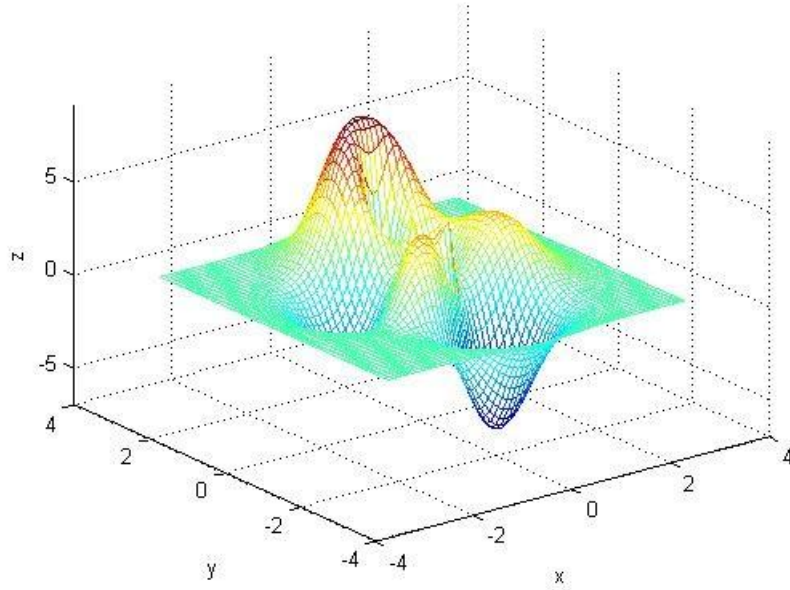
Deney	Başlangıç	Son	Ortalama	Süre
1	3.052473	0.192890	0.634419	1.553724
2	3.052473	0.049883	0.397866	1.530815
3	3.052473	0.048263	0.348132	1.670408
4	3.052473	0.040459	0.641288	1.990532
5	3.052473	0.089191	0.426686	1.682610
6	3.052473	0.205026	0.649690	1.624484
7	3.052473	0.193490	0.482344	1.526975
8	3.052473	0.134852	0.611580	1.541288
9	3.052473	0.028745	0.375113	1.794896
10	3.052473	0.124610	0.465402	1.891209

Çizelge 4.3'de listelendiği gibi en iyi eğitim son ölçüt değeri 0.028745 ile 9 nolu deneyle elde edilmiştir. İstenilen, ölçüt değerin sıfıra yakınsamasıdır ve bu üç ağ modeli arasında en iyi sonucu veren ağ bu araştırmada geliştirilen HybRbfNN ağ yapısıdır. 9. deneyin HybRbfNN ağ yapısının iPSO ile eğitim başarımı Şekil 4.8-4.10'da verilmiştir. Şekil 4.8'de HybRbfNN ağ yapısının eğitim seyri verilmiştir. Şekilde görüldüğü üzere eğitim ölçüt değeri sıfıra yakınsamıştır ve Çizelge 4.3'e göre 300. iterasyondaki eğitim ölçüt değeri 0.028745'dir.

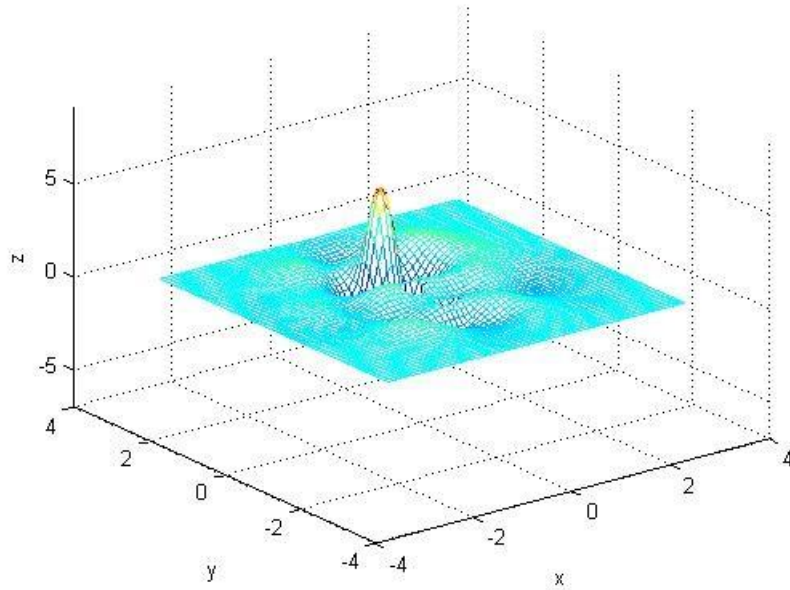
Şekil 4.9'a göre HybRbfNN'nin çıkış yüzeyi hemen hemen şekil 4.1'deki *peaks* fonksiyonun yüzeyinin aynısıdır. İstenilen modeli en iyi modelleyen ağ yapısı, bu araştırmada tasarlanan HybRbfNN ağ yapısıdır.



Şekil 4.8. iPSO ile HybRbfNN'nin eğitim seyri (Deney No 9), x: iterasyon, y: ortalama karesel hata.



Şekil 4.9. 9 nolu deney ile eğitilen HybRbfNN'nin gerçeklediği giriş-çıkış yüzeyi.



Şekil 4.10. 9 nolu deney ile eğitilen HybRbfNN'nin hata yüzeyi.

Şekil 4.10'da HybRbfNN ağı yapısının eğitim sonucunda bulunan parametreleri ile gerçeklediği hata yüzeyi verilmiştir. Görüldüğü gibi hata yüzeyi iyi bir modelleme yönteminden beklenildiği üzere sifra yakındır. Görülen küçük sapma okuyucuyu

yanılmamalıdır, zira ağırlıklı renk mavi olduğundan dolayı, değerler rakip ağ yapılarının değerlerinden çok daha fazla sıfıra yakındır.

Şekil 4.8-4.10'da görüldüğü üzere, bu çalışmada tasarlanan HybRbfNN ağ yapısı diğer rakip iki ağ modeline göre daha iyi sonuçlar vermiştir ve bu sonuçlarda HybRbfNN'nin modelleme yeteneğinin rakip yöntemlerden daha üstün olduğu görülmüştür.

Çizelge 4.1-4.3'de görüldüğü gibi 10 farklı deneyin sonuçlarına dayanarak belirtilebilir ki, HybRbfNN ağ yapısı, rakip ANFIS ve CSFNN ağ yapılarına kıyasla her deneyde en iyi eğitim ölçüt değerini sağlamıştır. Ayrıca, iterasyon süresi açısından bakılırsa, HybRbfNN ağ yapısı CSFNN ağ yapısından daha iyi sonuçlar verirken, daha hızlıdır. Öte yandan, HybRbfNN, ANFIS ağ yapısı kadar hızlı değildir, fakat sunduğu modelleme sonuçları, engebeli *peaks* fonksiyonu modellenmesinde başarısız olan ANFIS'ten çok daha iyidir.

5. SONUÇLAR

HybRbfNN, ÇKA ile RTF'nin birleştirilmesi ile elde edilen yeni bir ağ yapısıdır. Bu ağ yapısının ağırlık parametreleri iPSO ile eğitilmiştir. HybRbfNN'nin öğrenme başarımı, rakip olarak belirlenen ANFIS ve CSFNN ağları ile karşılaştırılmıştır.

Bir önceki bölümde çizelgeler 4.1-4.3 ve şekiller 4.2-4.10'da, ANFIS, CSFNN ve HybRbfNN ağ yapılarının Şekil 4.1'deki *peaks* fonksiyonunu modelleme başarımı verilmiştir. Şekil 4.4, Şekil 4.7 ve Şekil 4.10'da resmedilen hata yüzeyinin iyi bir modelleme için sıfıra yaklaşması beklenir. Şekil 4.10'da görüldüğü gibi HybRbfNN ağ yapısı sıfıra en yakın hata yüzeyine sahiptir.

Çizelge 4.1-4.3'de ise modelleme yöntemlerinin eğitim ölçüt değeri ile iterasyon başına eğitim süreleri listelenmiştir. Yine beklenen, en iyi modelleme yönteminin en düşük eğitim ölçüt değerini vermesidir, yani sıfıra yakınsamasıdır. Çizelge 4.3'de listelenen HybRbfNN ağ yapısı eğitim ölçüt değeri sonuçları, sıfıra en yakın sonuçlardır. Çizelgeler 4.1-4.3'de görüldüğü üzere eğitim her deneyde/tekrarda HybRbfNN diğer rakip ANFIS ve CSFNN ağ yapılarından daha iyi sonuçlar vermektedir.

ANFIS, HybRbfNN'e kıyasla daha az parametreye sahip ve daha az sürede çalışmış olmasına rağmen engebeli yüzeylerde (*peaks* fonksiyonuna dayanarak) modelleme yeteneği bakımından başarısızdır.

CSFNN, HybRbfNN ile aşağı yukarı aynı sayıda parametreye sahipken, CSFNN'nin eğitim süresi, HybRbfNN'nin eğitim süresinden daha uzundur. Ayrıca CSFNN'nin modelleme yeteneği HybRbfNN'nin modelleme yeteneğine göre daha kötüdür.

Sonuç olarak, *peaks* fonksiyonun yapısını göz önüne aldığımızda iddia edilebilir ki, HybRbfNN ağ yapısı, engebeli (inişli/çıkışlı) yüzey problemlerin çözümünde ANFIS ve CSFNN ağ yapılarına tercih edilebilir.

KAYNAKLAR

- Abraham, A. “Adaptation of Fuzzy Inference System Using Neural Learning”, *Fuzzy Systems Engineering, Springer Berlin Heidelberg* (2005).
- Ataç Kale, G. ve Karakuzu, C., “A Novel Modeling Network Structure and Its Heuristic Learning Performance”, *4th International Symposium on Innovative Technologies in Engineering and Science*, Antalya (2016).
- Bryson, A.E., Denham, W.F. ve Dreyfus, S.E., “Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for External Solutions”, *AIAA*, 2544-2550 (1963).
- Çavuşlu, M.A., Karakuzu, C. ve Karakaya, F., “Neural identification of dynamic systems on FPGA with improved PSO learning”, *Applied Soft Computing Elsevier*, 12: 2707-2718 (2012).
- Dorffner, G., “A Unified Framework for MLPs and RBFs: Introducing Conic Section Function Networks”, *Cybernetics and Systems* (1994).
- Du, K-L. ve Swamy, M.N.S., “Radial Basis Function Networks”, *Neural Networks and Statistical Learning, Springer* (2013).
- Elmas, Ç., “Bulanık Mantık Denetleyicileri”, *Seçkin Yayıncılık* (2003).
- Erkmen, B., “Genel Amaçlı Bir Yapay Sinir Ağının Karma Bir Donanımla Gerçeklenmesi”, Doktora Tezi, *Yıldız Teknik Fen Bilimleri Enstitüsü*, İstanbul (2007).
- Haykin, S., “Neural Networks : A comprehensive Foundation”, *Prentice Hall 2th ed.* , New Jersey (1998)
- Jang, J-S.R., “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, *IEEE Transactions on Systems, Man and Cybernetics*, 23(3): 665-685 (1993) .
- Jang, J-S.R., “Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm”, *Conference Proceedings of the 9th National*, California, 762-767 (1991).
- Karaboğa, D., “Yapay Zeka Optimizasyon Algoritmaları”, *Nobel Yayın dağıtım* (2011)
- Karakuzu, C., Bulanık Modelleme ve Akıllı Denetim Ders Notları, *Kocaeli Üniversitesi* (2006)
- Karakuzu, C., Yapay Sinir Ağları Ders Notları, *Bilecik Şeyh Edebali Üniversitesi* (2015)
- Kennedy, J. ve Eberhart, R.C., “Particle Swarm Optimization”, *IEEE International Conference on Neural Networks*, Piscataway, 1942-1948 (1995).
- McCulloch, W. ve Pitts, W. “A Logical Calculus of Ideas Immanent in Nervous Activity”, *Bulletin of Mathematical Biophysics*, 115-133 (1943).

KAYNAKLAR (DEVAM EDİYOR)

- Özsağlam, M.Y. ve Çunkaş, M. “Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması”, *Politeknik Dergisi*, 11: 299-305 (2008).
- Sağiroğlu, Ş., Beşdok, E. ve Erler, M., “Mühendislikte Yapay Zeka Uygulamaları I Yapay Sinir Ağları”, *Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd.Şti.*, Kayseri (2003).
- Şenol, C. ve Yıldırım, T., “Standart ve Hibrit Yapılar Kullanılarak Yapay Sinir Ağları ile İmza Tanıma”, *Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, Bursa (2004).
- Takagi, T. ve Sugeno, M. “Fuzzy Identification of Systems and Its Applications to Modeling and Control”, *IEEE Trans. on Systems, Man and Cybernetics*, 15: 116-132 (1985).
- Werbos, P. “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences”, Phdthesis, *Harvard University* (1974).
- Wikipedia*,
 “https://tr.wikipedia.org/wiki/Uyarlamalı_ağ_tabanlı_bulanık_çıkarm_sistemi”,
 (17.03.2017)
- Yalçın, N. “Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları ile Epilepsi Hastalığının Teşhisi”, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya (2012)

EK-1: Ana Programın Kodları

```

%%

clear all
close all
clc
format long

Gexternal=300;
%%
%Anfis için başlangıç tanımlama
D=20; %8UF:için(2x2+2x2) ; 12: 4kural için
%başlangıç tanımlamaları
sb_A=3*D;
%%
%HybRbfNN için başlangıç tanımlama
ornek=10;
RBFL=10;
HDL=2;%2;
SNL=3;%3;
D_H=2*RBFL+RBFL+RBFL*HDL+HDL*SNL+SNL*1+SNL*1;
sb_H=D_H*3;
%%
%CSFNN için başlangıç tanımlama
n=2; %Giriş sayısı
m=10; %Gizli katmandaki nöron sayısı
l=1; %Çıkış katmandaki nöron sayısı
D_C=n*m+m*1+m*2+m;
sb_C=D_C*3;

%% Eğitim örnekleri alınıyor...
xrange= linspace(-3, 3, ornek);
yrange=linspace(-3, 3, ornek);
x1=[];
x2=[];
zd=[];
%normal istenen çıkış
for i=1:ornek
for j=1:ornek
x=xrange(i); x1=[x1; x];
y=yrange(j); x2=[x2; y];
z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
- 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
- 1/3*exp(-(x+1).^2 - y.^2);

zd=[zd; z];
end% j

```

```

end%i

x=-3:0.1:3;
[X,Y]=meshgrid(x,x);
[satir,sutun]=size(X);

sure_H=0;
sure_A=0;
sure_C=0;
%%
%HybRbfNN' nin eđitimi
fprintf('-----ipso algoritm for HYRBFNN-----');
saat_H =tic;
[teta,omega,gama,A,ci,ss,C_H]=PSOAlgorithm(x1,x2,zd,sb_H,Gexternal,RBFL,HDL,
SNL,D_H);

for j=1:satir
for k=1:sutun

Zhyrbfnn(j,k)=HYRBFNN(X(j,k),Y(j,k),teta,omega,gama',RBFL,HDL,SNL,A,ci,ss);
end
end
sonsaat_H=toc(saat_H);
sure_H=(sure_H+sonsaat_H)/Gexternal;

figure(6)
%hata yüzeyi
for j=1:satir
for k=1:sutun
Zd(j,k)=peaks(X(j,k),Y(j,k));
end
end
mesh(X,Y,Zd-Zhyrbfnn)
title('Error Surface of Trained HYRBFNN');
xlabel('x');
ylabel('y');
zlabel('z');
axis([-4 4 -4 4 -7 9]);
saveas(gcf,'deneme\hyrbfnn_error_surface.fig')
%%
%ANFIS' in eđitimi
fprintf('-----ipso algoritm for Anfis-----');

saat_A =tic;
[cA,cB,ssA,ssB,p,q,r,C_A]=PSOAlgorithmAnfis(x1,x2,zd,sb_A,Gexternal,D);
sonsaat_A=toc(saat_A);
sure_A=(sure_A+sonsaat_A)/Gexternal;

```

```

for j=1:satir
for k=1:sutun
Zanfis(j,k)=anfis2(X(j,k),Y(j,k),cA,cB,ssA,ssB,p,q,r);

end
end

figure(11)
mesh(X,Y,Zd-Zanfis)
title('Error Surface of Trained ANFIS');
xlabel('x');
ylabel('y');
zlabel('z');
axis([-4 4 -4 4 -7 9]);
saveas(gcf,'deneme\anfis_error_surface.fig')

%%
% CSFNN' nin eđitimi
fprintf('-----ipso algoritm for CSFNN-----');

saat_C =tic;
[k_omega,k_teta,k_c,k_aci,C_C]=PSOAlgorithmKonik(x1,x2,zd,D_C,sb_C,Gexternal,
n,m,l);
sonsaat_C=toc(saat_C);
sure_C=(sure_C+sonsaat_C)/Gexternal;

for j=1:satir
for k=1:sutun
Zcsfnn(j,k)=CSFNN(X(j,k),Y(j,k),k_omega,k_teta,k_c,k_aci,m);
end
end

figure(16)
%hata yüzeyi

mesh(X,Y,Zd-Zcsfnn)
title('Error Surface of Trained CSFNN');
xlabel('x');
ylabel('y');
zlabel('z');
axis([-4 4 -4 4 -7 9]);
saveas(gcf,'deneme\csfnn_error_surface.fig')

%%

```

%Eğitim ölçüm değerleri

```
fprintf('\n---300. iterasyonun uygunluk değeri---\n');  
fprintf('HybRbfNNCost = %f\n',C_H(Gexternal));  
fprintf('ANFIS Cost = %f\n',C_A(Gexternal));  
fprintf('CSFNN Cost = %f\n',C_C(Gexternal));  
ort_C_H=sum(C_H)/Gexternal;  
ort_C_A=sum(C_A)/Gexternal;  
ort_C_C=sum(C_C)/Gexternal;  
fprintf('\n---Ortalama uygunluk değeri---\n');  
fprintf('HybRbfNNCost Ortalama= %f\n',ort_C_H);  
fprintf('ANFIS Cost Ortalama= %f\n',ort_C_A);  
fprintf('CSFNN Cost Ortalama= %f\n',ort_C_C);
```

%iterasyon başına geçen süre

```
fprintf('\n---iterasyon başına düşen süre---\n');  
fprintf('HybRbfNN Süre= %fsn\n',sure_H);  
fprintf('ANFIS Süre= %fsn\n',sure_A);  
fprintf('CSFNN Süre= %fsn\n',sure_C);
```

```
figure(12)  
peaks(50);  
saveas(gcf,'deneme\peaks.fig')  
savedeneme\matlab.mat
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Gizem Ataç Kale
Doğum Yeri ve Tarihi : Yenişehir / 01.01.1994



Eğitim Durumu

Lisans Öğrenimi : Bilecik Şeyh Edebali Üniversitesi, Müh. Fakültesi,
Bilgisayar Mühendisliği
Bildiği Yabancı Diller : İngilizce
Bilimsel Faaliyetleri :

1. Ataç Kale, G. ve Karakuzu, C., “A Novel Modeling Network Structure and Its Heuristic Learning Performance”, International Symposium on Innovative Technologies in Engineering and Science, 2016.
2. Kale Mehmet C., Ataç G. ve Gerek Ö. N., “Biorthogonal Wavelet Vanishing Moment Increment Technique”. SIU 2015 IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 2015.

İş Deneyimi

Stajlar :1- Anadolu Üniversitesi, Müh. Fak. Elektrik Elektronik Müh. Eskişehir
2- İncir ar-ge San. ve Tic. Ltd. Şti. Eskişehir
Projeler :Erasmus Otomasyon Sistemi, Akıllı Sınıf Bilgilendirme Sistemi
Çalıştığı Kurumlar :

İletişim

Adres : 75 Yıl. Mah. Hayri Başar Sok. No: 10
Odupazarı/Eskişehir
Tel : 0542 294 71 17
E-Posta Adresi : gzm.atac@gmail.com

Akademik Çalışmaları

1. Ataç Kale, G. ve Karakuzu, C., “A Novel Modeling Network Structure and Its Heuristic Learning Performance”, International Symposium on Innovative Technologies in Engineering and Science, 2016.
- 2.Kale Mehmet,C, Ataç, G., Gerek Ö. N. “A BiorthogonalWavelet Design Technique Using Karhunen–LoèveTransformApproximation”. DigitalSignal Processing, 51, 202-222.,Doi: 10.1016/j.dsp.2015.06.002, 2016.
- 3.Kale Mehmet C., Ataç G., Gerek Ö. N.,“BiorthogonalWaveletVanishing Moment IncrementTechnique”. SIU 2015 IEEE 23. Sinyal Isleme ve Iletisim Uygulamaları Kurultayı,2015.

Yabancı Dil Bilgisi

İngilizce **Okuma:** İyi **Yazma:** İyi **Konuşma:** Orta **Mesleki:** İyi

Tarih: 21/07/2017