



GİRESUN
ÜNİVERSİTESİ



FEN BİLİMLERİ ENSTİTÜSÜ

**Pİ-SİGMA YAPAY SİNİR AĞLARININ
DİFERANSİYEL GELİŞİM ALGORİTMASI İLE
EĞİTİMİ**

**MATEMATİK
ANA BİLİM DALI
Yüksek Lisans Tezi
Oğuzhan YILMAZ
20152110028
Haziran 2019**

GİRESUN

**T.C.
GİRESUN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**Pİ-SİGMA YAPAY SİNİR AĞLARININ
DİFERANSİYEL GELİŞİM ALGORİTMASI İLE
EĞİTİMİ**

YÜKSEK LİSANS TEZİ

Oğuzhan YILMAZ

Enstitü Ana Bilim Dalı : Matematik

Tez Danışmanı : Doç. Dr. Eren BAŞ

Haziran 2019

T.C.
GİRESUN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**Pİ-SİGMA YAPAY SİNİR AĞLARININ
DİFERANSİYEL GELİŞİM ALGORİTMASI İLE
EĞİTİMİ**

YÜKSEK LİSANS TEZİ

Oğuzhan YILMAZ

Enstitü Anabilim Dalı : Matematik

Bu tez 27/06/2019 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.



**Prof. Dr.
Erol EĞRİOĞLU**

Jüri Başkanı



**Doç. Dr.
Cem KOÇAK**

Üye



**Doç. Dr.
Eren BAŞ**

Üye

Doç. Dr. Bahadır KOZ

Enstitü Müdürü

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Oğuzhan YILMAZ

.../.../2019



TEŞEKKÜR

Tez çalışmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam sayın Doç. Dr. Eren BAŞ' a sonsuz teşekkür ve saygılarımı sunarım.

Lisans ve yüksek lisans eğitimim boyunca yardım, bilgi ve tecrübeleri ile bana sürekli destek olan başta Prof. Dr. Erol EĞRİOĞLU olmak üzere İstatistik bölümündeki tüm hocalarıma teşekkür ederim.

Çalışmalarım boyunca yardımını hiç esirgemeyen değerli arkadaşım Ali KARADAĞOĞLU' na da teşekkürlerimi bir borç bilirim.

Çalışmalarım boyunca maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme ve başta ağabeyim Tufan YILMAZ' a teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	I
İÇİNDEKİLER	II
SİMGELER VE KISALTMALAR LİSTESİ	IV
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ.....	VI
ÖZET.....	VII
SUMMARY	VIII
BÖLÜM 1. GİRİŞ.....	1
1.1. YAPAY SİNİR AĞI	1
1.1.1. Yapay Sinir Ağları	1
1.1.2. Yapay Sinir Ağı Nedir?.....	1
1.1.3. Yapay Sinir Ağlarının Avantajları ve Özellikleri	2
1.1.4. Yapay Sinir Ağlarının Dezavantajları	3
1.1.5. Yapay Sinir Ağı'nın Kullanım Alanları	3
1.1.6. Yapay Sinir Ağı'nın Yapısı	4
1.1.7. Yapay Sinir Hücresi Temel Yapısı	4
1.1.8. Yapay Sinir Ağı'nda Öğrenme	7
1.1.9. Yapay Sinir Ağlarının Sınıflandırılması	7
1.1.9.1. Öğrenme Algoritmalarına Göre Sınıflandırma	8
1.1.9.1.1. Eğitici Öğrenme	8
1.1.9.1.2. Eğitici Öğrenme	9
1.1.9.1.3. Destekleyici Öğrenme	9
1.1.9.2. Yapılarına Göre Sınıflandırma	10
1.1.9.2.1. İleri Beslemeli Ağlar	10
1.1.9.2.2. Geri Beslemeli Ağlar	11
1.2. DİFERANSİYEL GELİŞİM ALGORİTMASI	12
1.3. PI-SIGMA YAPAY SİNİR AĞLARI.....	14

BÖLÜM 2. KAYNAK ARAŞTIRMASI.....	16
BÖLÜM 3. MATERYAL VE YÖNTEM.....	20
3.1. ÖNERİLEN YÖNTEM.....	20
BÖLÜM 4. ARAŞTIRMA BULGULARI.....	24
4.1. UYGULAMALAR	24
4.1.1. Avustralya Bira Tüketimi Zaman Serisi Analizi.....	24
4.1.2. Türkiye Elektrik Tüketimi Zaman Serisi Analizi.....	27
BÖLÜM 5. TARTIŞMA VE SONUÇ	29
KAYNAKLAR	30
EKLER.....	34
ÖZGEÇMİŞ	38

SİMGELER VE KISALTMALAR LİSTESİ

YSA	: Yapay Sinir Ağı
DGA	: Diferansiyel Gelişim Algoritması
HKOK	: Hata Kareler Ortalamasının Karekökü
OMYH	: Ortalama Mutlak Yüzde Hata kriteri
PSO	: Parçacık Sürü Optimizasyonu
GA	: Genetik Algoritması
ÇKA	: Çok Katmanlı Algılayıcı
PS-YSA	: Pi-Sigma Yapay Sinir Ağı
ÇNM-YSA	: Çarpımsal Nöron Modeli Yapay Sinir Ağı
RP-YSA	: Ridge Polinomal Yapay Sinir Ağı
YYA	: Yapay Yarasa Algoritması

ŞEKİLLER LİSTESİ

Şekil 1. Biyolojik Sinir Hücre Yapısı	4
Şekil 2. Yapay Sinir Hücre Yapısı	6
Şekil 3. Eğitici Öğrenme.....	8
Şekil 4. Eğitici Öğrenme.....	9
Şekil 5. Destekleyici Öğrenme.....	10
Şekil 6. İleri Beslemeli Ağ Modeli.....	10
Şekil 7. Geri Beslemeli Ağ Modeli	11
Şekil 8. PS-YSA'nın mimarisi	14
Şekil 9. Bir kromozomun yapısı	21
Şekil 10. AUST Zaman serisi grafiği	24
Şekil 11. AUST test kümesi ile önerilen yöntemden elde edilen öngörülerin birlikte grafiği	26
Şekil 12. TEC Zaman serisi grafiği	27
Şekil 13. TEC test kümesi ile önerilen yöntemden elde edilen öngörülerin birlikte grafiği	28

TABLÖLAR LİSTESİ

Tablo 1. AUST test verisi için tüm yöntemlerden elde edilen HKOK ve OMYH değerleri.....	26
Tablo 2. TEC test verisi için tüm yöntemlerden elde edilen HKOK ve OMYH değerleri.....	28



PI-SİGMA YAPAY SINİR AĞLARININ DİFERANSİYEL GELİŞİM ALGORİTMASI İLE EĞİTİMİ

ÖZET

Literatürde öngörü problemi üzerine çalışmalar temelde klasik öngörü modelleri ile başlamış olup son yıllarda yapay sinir ağları ile başarılı bir şekilde devam etmektedir. Literatürde çok katmanlı ileri beslemeli yapay sinir ağları, çarpımsal nöron modeli yapay sinir ağları, Pi-Sigma yapay sinir ağları gibi birçok yapay sinir ağının öngörü problemi için kullanımında oldukça başarılı sonuçlar verdiği bilinmektedir.

Bu yapay sinir ağlarından Pi-Sigma yapay sinir ağları hem toplamsal hem de çarpımsal birleştirme fonksiyonuna sahip olması ve yüksek dereceli bir yapay sinir ağı türü olması nedeni ile diğer sinir ağlarından farklılık göstermektedir.

Literatürde, öngörü amacı ile kullanılan Pi-Sigma yapay sinir ağlarının eğitimi için genetik algoritma, parçacık sürü optimizasyonu gibi sezgisel optimizasyon algoritmaları kullanılmasına rağmen henüz diferansiyel gelişim algoritması kullanılmamıştır.

Bu tez kapsamında Pi-Sigma yapay sinir ağlarının eğitimi ilk kez diferansiyel gelişim algoritması ile gerçekleştirilmektedir.

Önerilen yöntemin performansının değerlendirilmesi için literatürde sıklıkla kullanılan iki farklı gerçek hayat zaman serisi çözümlenmiş ve önerilen yöntemin performansı diğer bazı yapay sinir ağı türleriyle karşılaştırılmıştır.

Anahtar kelimeler: Pi-Sigma Yapay Sinir Ağları, Diferansiyel Gelişim Algoritması, Öngörü

THE TRAINING OF PI-SIGMA ARTIFICIAL NEURAL NETWORKS WITH DIFFERENTIAL EVOLUTION ALGORITHM

SUMMARY

In the literature, studies on forecasting problem have started with classical forecasting models and have been successfully continued with artificial neural networks in recent years. In the literature, it is known that many artificial neural networks such as multilayer feed-forward neural networks, multiplicative neuron model neural networks, Pi-Sigma artificial neural networks are very successful in the use of forecasting problem.

Pi-Sigma artificial neural networks from these neural networks differ from other neural networks due to the fact that they have both additive and multiplicative functions and are a high order artificial neural network type.

In the literature, although the use of heuristic optimization algorithms such as genetic algorithm, particle swarm optimization for the training of Pi-Sigma artificial neural networks used for forecasting purpose, the differential evolution algorithm has not been used yet.

Within the scope of this thesis, the training of Pi-Sigma artificial neural networks is carried out for the first time with differential evolution algorithm.

In order to evaluate the performance of the proposed method, two different real life time series, which are frequently used in the literature, were analyzed and the performance of the proposed method was compared with some other types of artificial neural networks.

Keywords: Pi-Sigma Artificial Neural Networks, Differential Evolution Algorithm, Forecasting

BÖLÜM 1. GİRİŞ

1.1. YAPAY SİNİR AĞI

1.1.1. Yapay Sinir Ağları

YSA insan beyninin öğrenme yöntemi kullanarak yeni bilgilerin üretilmesi için geliştirilmiş bilgisayar sistemleridir. Nörolog olan Donald Hebb (1949) beynin öğrenme yapısı ile ilgili araştırmalar yaparak şu anda kullanmış olduğumuz sinir ağlarının temel modelini oluşturmuştur. İlk sinir ağı modelini oluşturan Hebb, beynin temel birimi olan sinir hücrelerini inceleyerek iki sinir hücresinin birbirleri ile nasıl bir ilişki içerisinde olduğunu gözlemlemiştir. Genel olarak YSA'lar insan beynindeki sinir hücrelerinin birbirleri ile etkileşim halinde olan karmaşık sistemler bütünüdür. Bu sistemler, çözülmesi zor ve karmaşık problemlerin sonuca ulaştırılmasında kullanılan bir bilim dalı olarak geliştirilmiştir.

1.1.2. Yapay Sinir Ağı Nedir?

Nörolog Donald Hebb tarafından ilk kez ortaya atılan YSA'lar, paralel olarak dağılmış zeka gerektiren bir bilgi işleme merkezine sahiptir. Bu merkez tek yönlü olarak hareket eder ve diğer sinir hücreleri ile etkileşim halinde olur. İnsan beyni ile benzerlik gösterdiği için biyolojik olarak karmaşık sistemlerdir. Bu yüzden sistemdeki girdi değerleri ile çıktı değerleri o anki duruma göre farklılıklar gösterebilir.

Yapay sinir ağları çevreden gelen olaylara karşı nasıl tepki verdiğini belirleyen karakteristik özellikleri mevcuttur. Bunlar;

- Genelleme
- Sınıflandırma
- Optimizasyon
- Özellik belirleme
- Öğrenme

Günümüzde yapay sinir ağları birçok probleme çözüm üretme yeteneği ile ön plana çıkmaktadır. Değişik şekillerde tanımlanıp modellenenmektedir. Yapılan tanımların ortak noktası bulunmaktadır. Bu ortak noktalar yapay sinir ağlarının birbirlerine hiyerarşik olarak bağlı ve paralel olarak çalışabilen yapay hücrelerinden oluşması gerekmektedir.

1.1.3. Yapay Sinir Ağlarının Avantajları ve Özellikleri

Standart olarak yapay sinir ağlarının özellikleri, karşılaşılan problemde uygulanan ağ modeline göre değişiklik göstermektedir. Aşağıda bütün modeller için geçerli olan genel karakteristik özellikleri ve avantajları verilmiştir.

- YSA' lar örneklerden faydalananarak öğrenirler.
- YSA'nın kaliteli çalışabilmesi için ilk olarak eğitilmeleri ve performanslarının test edilmesi gereklidir.
- Algılamaya yönelik olaylarda kullanılabilirler.
- Şekil ve sınıflandırma yapabilirler.
- Eksik bilgi ile çalışabilir ve hata toleransına sahiptirler.
- Tam olamayan belirsiz verileri işleyebilirler.
- Makine öğrenmesi ile çalışıp, sadece numerik öğrenme gerçekleştirebilirler.
- Kendi kendini organize etme ve öğrenebilme yeteneğine sahiptir.

1.1.4. Yapay Sinir Ağlarının Dezavantajları

Yapay sinir ağlarında probleme uygun model belirleyebilmek için genellikle deneme yanılma yolu kullanılır. Buda problem için oluşturulan modellemede bazı aksaklıklara yol açabilmektedir.

- Problem için uygun bir ağ modeli oluşturulamaz ise çözümü olan problemin çözülemez ya da düşük çözümlerin elde edilmesi söz konusu olabilir.
- Ağın öğreneceği problemin ağa gösterimi problem oluşturabilir.
- Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belirli bir yöntem yoktur.
- Uygun ağ yapısının belirlenmesinde belli bir kural yoktur.
- YSA'nın yapısı gereği paralel işlem gücüne sahip işlemcilere ihtiyaç duymaktadır. Bu nedenle gerçekleştirilmesi donanıma bağımlıdır.

1.1.5. Yapay Sinir Ağının Kullanım Alanları

YSA'nın kullanım alanları yapılan projelere ve karşılaşılan problemlere göre günümüzde birçok alanda farklılık gösterebilir. Bunlar aşağıdaki gibidir.

- Optik tanıma sistemleri.
- Kredi kartı güvenlik uygulamaları.
- Radar ve sonar sinyalleri sınıflandırma.
- Kalite kontrol ve ürünün pazardaki payını tahmin etme.
- Robotlar ve otonom çalışan araçlar için optimum durumu belirleme.
- Kanseri teşhisinde ve kalp krizinin tedavisinde.
- Üretim bandı çizelgesi ve planlamasında.
- Veri madenciliği alanında.
- Öngörü

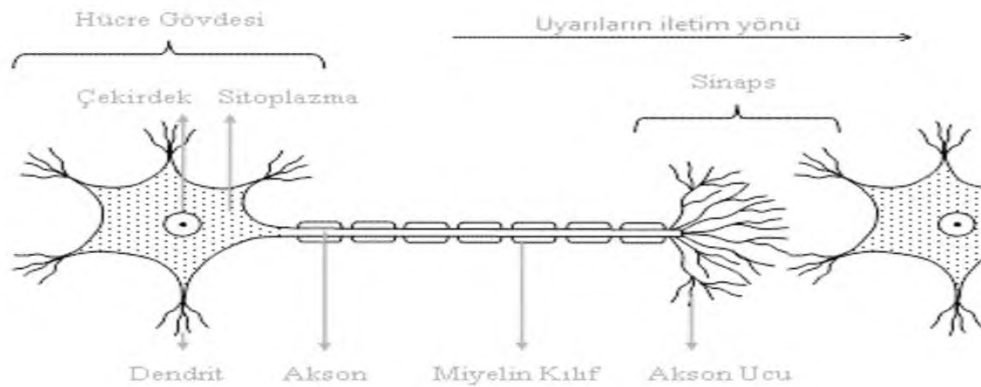
Yukarıdaki kullanım alanlarını geliştirmek mümkündür. Bu alanlar ile kısıtlı tutulamaz ve her alanda kullanıma açıktır.

1.1.6. Yapay Sinir Ağının Yapısı

YSA' lar biyolojik sinir hücrelerinden esinlenerek geliştirilmiştir. Biyolojik sinir ağları beynimizde bulunan birçok sinir hücresinden meydana gelmektedir. Sinir hücreleri birbirleri ile bağlanarak fonksiyonları yerine getirirler. İnsan beyni çok hızlı çalışabilen bir bilgisayara benzetilebilir. İnsan beyninin yapmış olduğu veri işleme bütünlüğünü sıradan bilgisayarlar ile yapmak çok uzun zaman alabilir. Günümüzde insan beynin kapasitesinin belirli bir oranı bilgisayar sistemlerine aktarılmış olsa, çok hızlı olağanüstü bilgi işleme ve kontrol etme merkezleri ile mükemmel sonuçlar elde edilebilir. Biyolojik sinir ağlarının performans hızları karmaşık olayları işleyebilecek yeteneğe sahiptir. Bu yeteneği yapay sinir ağları kullanılarak bilgisayarlara kazandırılması amaçlanmaktadır.

1.1.7. Yapay Sinir Hücresi Temel Yapısı

Yapay sinir ağları, temel olarak basit yapıda ve yönlü bir düğüm biçiminde tasarlanmıştır. Her bir düğüm bir devreyi ifade etmektedir. Düğümler işlem elemanı olarak tanımlanır ve düğümler arasında ilişki söz konusudur. Her bir ilişki tek yönlü iletim yolu ile görev yapmaktadır. Her işlem elemanı istenildiği sayıda giriş bağlantısı ve tek bir çıkış bağlantısı alabilir. Yani tek bir çıkış birçok hücreyi besleyebilir. Aşağıdaki Şekil 1 'de basit bir sinir hücre yapısı verilmiştir.



Şekil 1. Biyolojik Sinir Hücre Yapısı

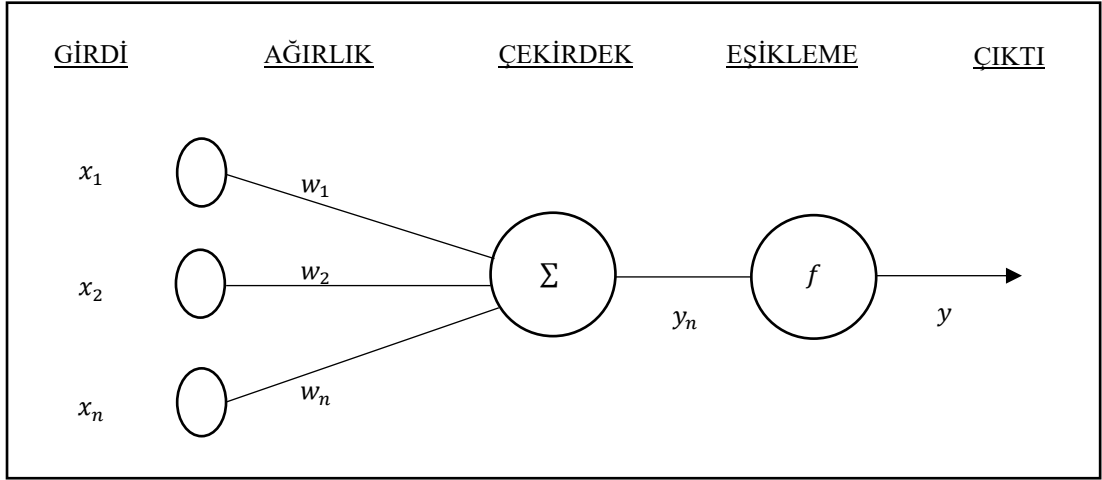
Dendrit, diğ er sinir h ucreslerinden gelen sinyalleri alarak sinir h ucresinin  ekirdeğine iletmektedir. Yani bulunduğ u sinir h ucresinin iletiřim kanalıdır. H ucresinin  ekirdeğ i ile her bir dendrit arasında farklı bir iletiřim s oz konusudur. Bu durumdan sebeple bazı dendritlerin bulundukları etkileřimde dominant pay sahibi, bazılarının ise pasif olduğ u g ozlenmiřtir. Yani dıřarıdan alınan sinyallerin se ilerek sinir h ucresi tarafından ger ekleřtirildiğ i anlamını tařımaktadır.

Soma, dendritler yolu ile gelen b ut un sinyalleri toplayan merkezdir. H ucre  ekirdeğ i olarak da bilinmektedir.  ekirdek, gelen b ut un sinyalleri bir arada tutarak diğ er sinir h ucresine iletilmesi i in bilgiyi aksona iletir.

Akson, somadan ( ekirdek) gelen bilgileri bir sonraki sinir h ucresine dağ ıtmak ile g orevlidir. Bu iřlemi yaparken gelen bilgileri bir  on s uzge ten ge irerek diğ er h ucreye iletir. Bunu da sinapsis denilen biyolojik yapı ile ger ekleřtirir.

Sinapsis, aksondan gelen toplam bilgiyi  on s uzge ten ge irdikten sonra bir sonraki sinir h ucresinin dendritine iletmek ile g orevlidir. Sinapsisin yapmıř olduğ u bu iřlem  ok  onem tařımaktadır. Yani gelen b ut un sinyal belirli bir eřik değ erine g ore değ erlendirilir. B oyece toplam sinyal olduğ u gibi değ il belirli bir aralığ a indirgenerek iletilmiř olur. Gelen sinyal ile dendrite iletilen sinayal arasında bir iliřki kurulmuř olur. Kurulan iliřki ile  ğrenme iřlemi sinapsiste sinapsiste ger ekleřir. Bu durumdan yola  ıkararak yapay sinir ağlarının  ğrenme teorisi ortaya atılmıřtır.

Burada yapay sinir h ucresinin biyolojik olarak tanımını ve  alıřma prensibinden bahsedilmiřtir.



Şekil 2. Yapay Sinir Hücre Yapısı

Şekil 2’ de yapay sinir hücresinin dendritleri x_n ve her dendritin ağırlık katsayısı (önem derecesi) w_n olarak belirtilmiştir. Buradaki x_n girdi değerlerini, w_n ise sinyallerin ağırlık katsayılarını ifade etmektedir. Gelen bütün sinyaller y_n ile belirtilmiş ve sinapsise yani eşikleme fonksiyonuna girdi olarak yönlendirilmiştir. Sinapsisten çıkan y değeri de bir sonraki hücreyi beslemek için yönlendirilmiştir. Bu durum diğer bütün sinir hücreleri için aynıdır.

Burada yapay sinir hücresinin yapması gereken x_n girdi değerine karşılık gelecek bir y çıktı sinyali oluşturmaktır. Oluşturulan bu sinyali de bir sonraki sinir hücresine iletmektir. x_n ile y arasındaki ilişkiyi temsil eden w_n ağırlık değeri yeni girdi ve çıktı sinyallerine göre ayarlanır. Bu ayarlanma süreci sinir hücresinin öğrenmesi olarak adlandırılır. Bu işlem hücre durağanlaşınca kadar devam etmektedir. Yukarıda yapay sinir hücresinin temel yapı bileşenlerinin tanımı yapılmıştır. Sıradaki işlem öğrenme işlemidir.

1.1.8. Yapay Sinir Ağında Öğrenme

YSA öğrenme işlemi ağı eğitilmesi ile başlamaktadır. Bir sinir ağının eğitilmesi ağırlık değerlerinin belirlenmesi ile devam etmektedir. Başlangıçta bu ağırlık değerleri rastgele olarak belirlenir. Optimal ağırlık değerleri elde edilinceye kadar ağı örnekler gönderilir ve ağırlık değerleri güncellenir. Ağı doğru ağırlık değerlerini seçmesi, temsil ettiği olay hakkında yorum yapabilme yeteneğine ulaşması anlamına gelir. Doğru yorum yapabilme özelliğine sahip olduğunda sinir hücresi öğrenme işlemini tamamlamış olur. Dışarıdan alınan girdi değerleri ile her defasında ayarlanan ağırlıklar öğrenmenin gerçekleştiğinin temsilcisidir. Sayısal olarak ağırlık değerleri girilen bütün girdi değerlerini en iyi temsil etmeye çalışan ve bu girdi değerlerinin uzaklıklar toplamının minimum olduğu regresyon eğrisinin temsil edildiği geometrik şeklin en belirleyici noktalarını oluşturmaktadır. Böylelikle girilen değere en doğru cevap verilmiş olur.

1.1.9. Yapay Sinir Ağlarının Sınıflandırılması

Genel olarak YSA birbirleri ile iletişimde olan sinir hücrelerinden işlemci birimlerinden oluşmaktadır. Her bir sinir hücresi arasındaki ilişkilerin yapısı ağı yapısını belirler. Belirlenen amaca ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan öğrenme yöntemine göre hatayı minimize edecek şekilde ağda kullanılan ağırlık değerleri değiştirilir. Yapay sinir ağları ile birçok öğrenme metotları kullanılmaktadır. Öğrenmeyi gerçekleştirecek olan sistem bu metotlara bağlı olarak değişmektedir. Bu öğrenme metotları birkaç ana başlık altında toplanmıştır.

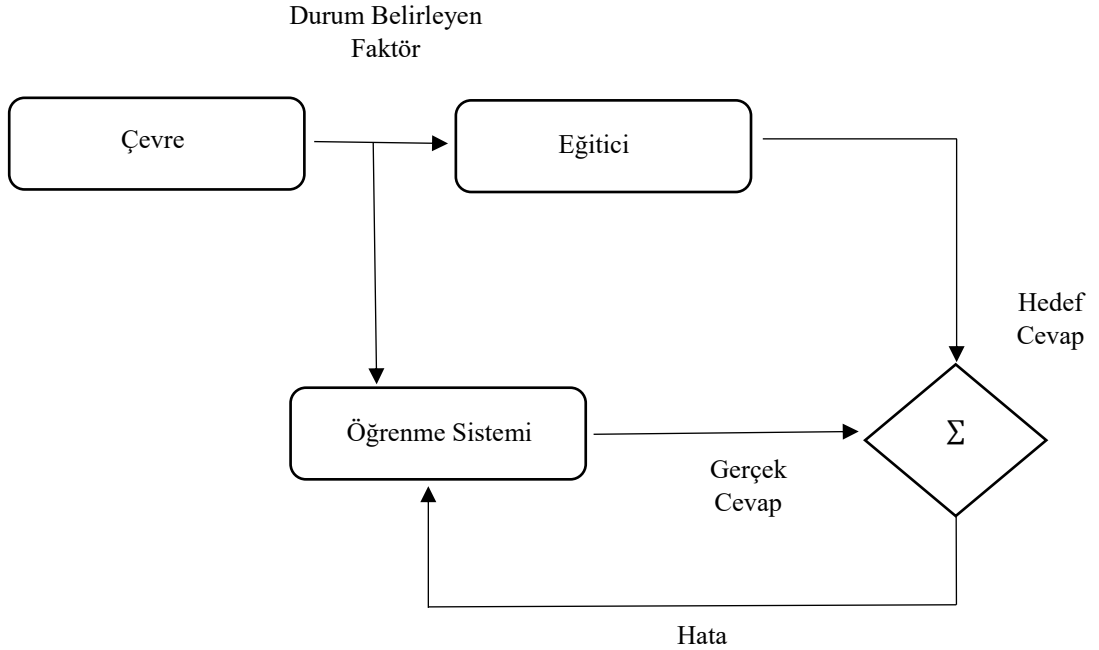
1.1.9.1. Öğrenme Algoritmalarına Göre Sınıflandırma

Yapay sinir ağları, öğrenme algoritmalarına göre üç başlık altında toplanmaktadır. Bunlar;

- Eđiticiili Öğrenme
- Eđiticisiz Öğrenme
- Destekleyici Öğrenme

1.1.9.1.1. Eđiticiili Öğrenme

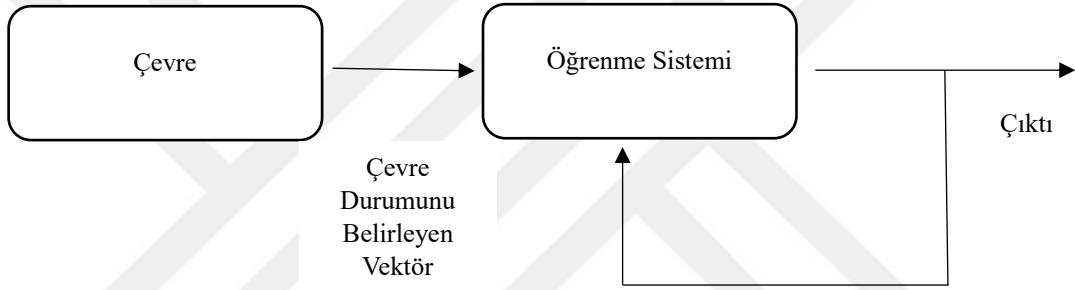
Bu öğrenme metodunda sistemin öğrenebilmesi için bir eđitici yardımcı olmaktadır. Eđitici, sisteme öğrenilmesi istenen olay ile ilgili örnekleri girdi ve çıktı birimi olarak öğretir. Tam olarak örnek için hem girdiler hem de girdilere karşılık gelen çıktı değeri sistemi gösterilir. Hata oranı azaltılır ve öğrenme işlemi gerçekleşmiş olur. Eđiticiili öğrenme yöntemine örnek olarak çok katmanlı algılayıcılar verilebilir. Eđiticiili ağ modeli aşağıdaki Şekil 3' te verilmiştir.



Şekil 3. Eđiticiili Öğrenme

1.1.9.1.2. Eđiticisiz Öğrenme

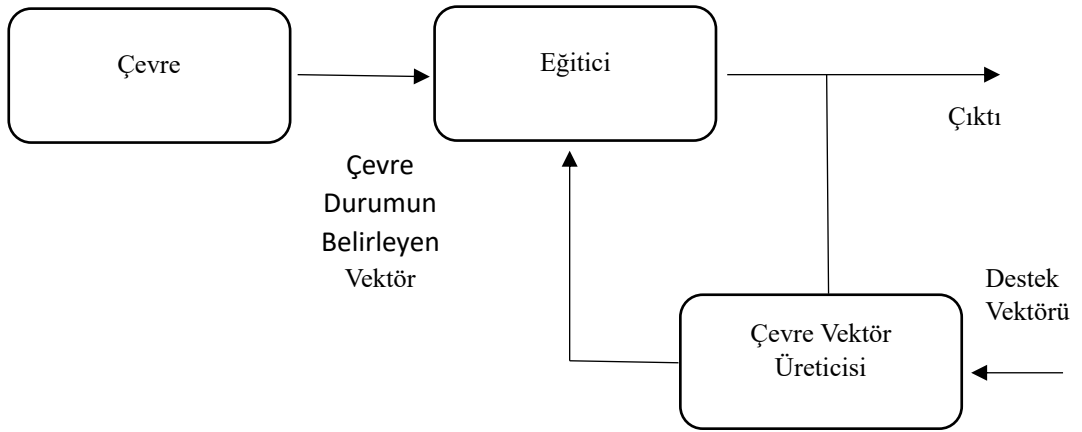
Eđiticisiz öğrenme sisteminde, eđiticili öğrenme sistemindeki gibi bir ađı önceden eđitici yoktur. Yani girdi deđerleri sisteme tek seferlik gösterilir. Modeldeki deđerler arasındaki ilişkiyi sistem kendi kendine öğrenir. Bu öğrenme metodu daha çok sınıflandırma problemlerinde kullanılan bir yöntemdir. Bu öğrenme yöntemini eđiticili sistemden ayıran durum, model oluşturulup öğrenme gerçekleştirildikten sonra çıktı deđerlerinin ne anlama geldiđinin kullanıcı tarafından belirtilmesidir. Eđiticisiz öğrenme modeli Şekil 4'te verilmiştir.



Şekil 4. Eđiticisiz Öğrenme

1.1.9.1.3. Destekleyici Öğrenme

Destekleyici öğrenme yönteminde sisteme eđitici yardımcı olmaktadır. Yalnız eđitici girilen bütün deđerlere karşılık gelen çıktı setini sisteme önceden öğretmek yerine, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler. Üretilen çıktıların dođru ya da yanlış olduđunu gösteren sinyaller üretir. Bu sinyaller vasıtası ile sistem öğrenme sürecini devam ettirir. Destekleyici öğrenme ađ modeli Şekil 5'te verilmiştir.



Şekil 5. Destekleyici Öğrenme

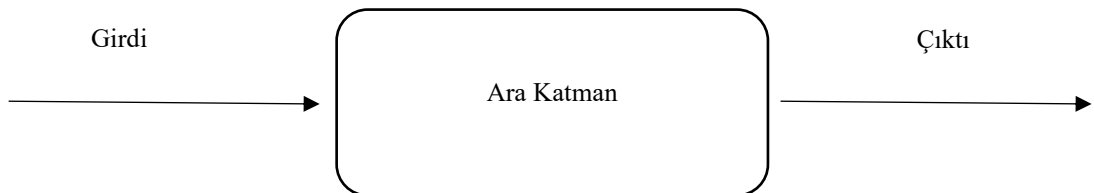
1.1.9.2. Yapılarına Göre Sınıflandırma

Yapay sinir ağları yapılarına göre iki başlık altında sınıflandırılmıştır. Bunlar;

- İleri Beslemeli Ağlar
- Geri Beslemeli Ağlar

1.1.9.2.1. İleri Beslemeli Ağlar

Bu ağ modelinde veri işlemesi katmanlar halinde oluşmaktadır. Nöronlar girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katman ile iletişim bulunmaktadır. Yani tek yönlü bir etkileşim söz konusudur. Gelen bilgiler bir katmandan diğer bir katman ile iletişim kurarken, öncelikle giriş katmanına daha sonra ara katmana ve son olarak çıkış katmanından geçerek çıktı olarak sunulmaktadır. İleri beslemeli ağlara örnek olarak Multi Layer Perceptron-MLP ve Learning Vector Quantization –LVQ verilebilir. İleri beslemeli ağ modeli aşağıdaki Şekil 6’ da gösterilmiştir..

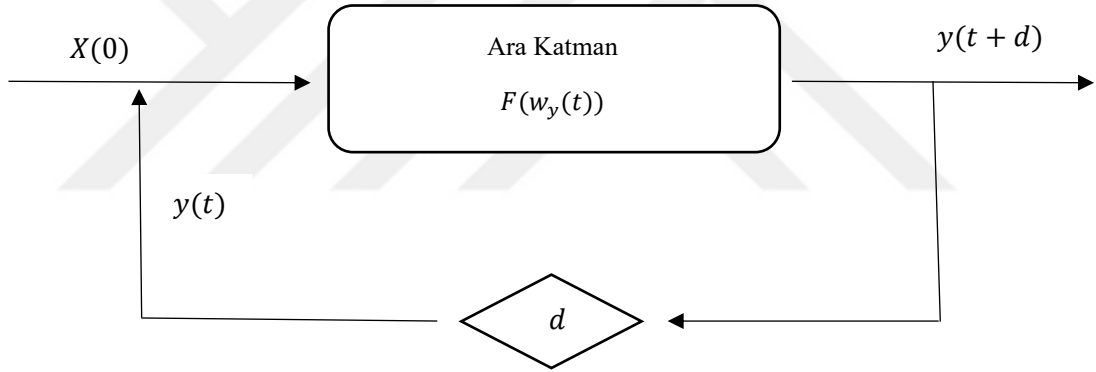


Şekil 6. İleri Beslemeli Ağ Modeli

1.1.9.2.2. Geri Beslemeli Ağlar

Geri beslemeli yapay sinir ağlarında, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir. Bu yapısı ile geri beslemeli yapay sinir ağları, doğrusal olmayan dinamik bir davranış gösterir.

Geri beslemeli yapay sinir ağı, çıkış ve ara katmanlardaki giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Bu hali ile hem ileri yönde hem geri yönde bir iletişim gerçekleşmiş olur. Dolayısıyla geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli yapay sinir ağı yapıları elde edilmiş olur. Geri beslemeli ağlara örnek olarak Hopfield, Elman, Jordan ve Self Organizing Map ağları verilebilir. Geri beslemeli ağ modeli Şekil 7’ de verilmiştir.



Şekil 7. Geri Beslemeli Ağ Modeli

1.2. DİFERANSİYEL GELİŞİM ALGORİTMASI

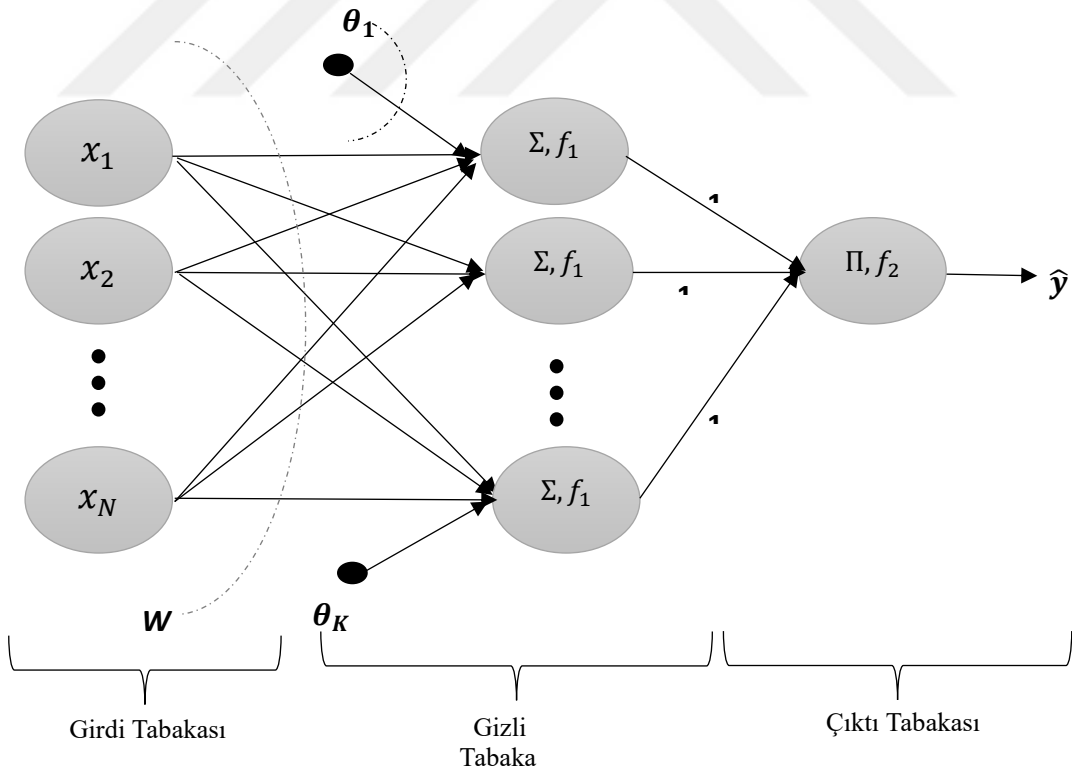
DGA, Price ve Storn (1995) tarafından önerilen popülasyon tabanlı sezgisel bir optimizasyon algoritmasıdır. DGA, popülasyon tabanlı bir diğer sezgisel optimizasyon algoritması olan GA ile benzer özelliklere ve operatörlere sahip olup benzer operatörlerin uygulanışı bakımından farklılıklara sahiptir. DGA’da belli sayıda genden oluşan her bir kromozom tıpkı GA’da olduğu gibi bir çözüme karşılık gelmektedir. Kromozomların bir araya gelmesiyle oluşan popülasyon ise yine GA’da olduğu gibi DGA’da da bir çözüm kümesine karşılık gelmektedir. GA’da popülasyonda çeşitlilik yaratmak için kullanılan çaprazlama ve mutasyon operatörleri DGA’da da kullanılmakta olup daha önce de bahsedildiği gibi bu operatörlerin DGA’da uygulanması GA’dan farklıdır. Bir DGA süreci öncelikle ilgili problemin yapısına göre gen sayısının ve yapısının belirlenmesi ile başlar. İlgili problem için belirlenen gen sayısına göre bir kromozom oluşturulur ve belli sayıda kromozomların bir araya getirilmesi ile de çözüm kümesi oluşturularak DGA operatörlerinin uygulanmasına geçilir. DGA sürecinde çeşitlilik yaratmak için tıpkı GA’da olduğu gibi mutasyon ve çaprazlama operatörleri kullanılır. Bu operatörlerden öncelikle mutasyon operatörü uygulanır. Mutasyon operatörü öncelikle popülasyonda bulunan ilk kromozoma daha sonra da sırası ile popülasyonda bulunan tüm kromozomlara uygulanır. Mutasyon operatörü uygulanacak kromozoma ilgili kromozom adı verilir. İlgili kromozoma mutasyon operatörü uygulanırken ilgili kromozom dışında popülasyonda bulunan kromozomlardan rasgele üç farklı kromozom seçilir. Bu nedenle popülasyonda bulunan kromozom sayısı en az dört olmalıdır. Rasgele seçilen bu üç farklı kromozomdan, öncelikle ilk ikisinin farkı alınır. Alınan fark sonucu elde edilen kromozoma fark vektörü adı verilir. Bu fark vektörü ölçekleme faktörü adı verilen bir değer ile çarpılır ve ağırlıklandırılmış fark vektörü adı verilen bir kromozom oluşturulur. Son olarak, ağırlıklandırılmış fark vektörü ile rasgele seçilen üçüncü kromozom toplanır ve toplam vektör adı verilen bir kromozom oluşturulur. Elde edilen bu toplam vektör çaprazlama operatöründe kullanılmak için hazırlanmış bir

kromozomdur. Çaprazlama operatörünün uygulanmasında ise mutasyon operatörü sonucunda elde edilen toplam vektör ile ilgili kromozom karşılıklı genleri karşılaştırılarak aday kromozom elde edilir. Çaprazlama operatörünün uygulanmasına geçmeden önce bir çaprazlama oranı (*cor*) belirlenir ($0 \leq cor \leq 1$). Daha sonra düzgün dağılım yardımıyla 0 ile 1 arasında gen sayısı kadar rastgele sayı üretilir. Her bir gen için ayrı ayrı değerlendirilen bu rasgele sayılara eğer üretilen bu çaprazlama oranından küçükse, gen toplam vektöründen; büyük ise, gen ilgili vektörden alınır ve aday kromozom oluşturulur ve aday kromozomun uygunluk değeri hesaplanır.

Aday kromozom ile ilgili kromozomun uygunluk değerleri amaç problem doğrultusuna karşılaştırılır. Amaç problemine uygun olarak, karşılaştırılan kromozomlardan uygunluğu daha iyi olan kromozom yeni jenerasyona aktarılır. Bu işlemler popülasyonda bulunan her bir kromozoma ayrı ayrı uygulanmalıdır.

1.3. PI-SIGMA YAPAY SİNİR AĞLARI

PS-YSA ilk olarak Shin and Ghosh (1991) tarafından önerilmiştir. PS-YSA' da girdilerin farklı doğrusal kombinasyonlarının çarpımı ağın çıktısını oluşturmaktadır. Lineer kombinasyonların sayısı da PS-YSA derecesidir. PS-YSA' nın derecesinin artırılması, girdi ile çıktı arasındaki ilişkiyi tanımlayan fonksiyonun daha fazla parametreye bağlı ve daha karmaşık bir yapıda olmasına neden olur. Bu durum daha iyi tahmin sonuçları elde edilmesine olanak sağlar ancak fazla parametre ile çalışılması overfitting problemine ve eğitim algoritması için daha fazla hesaplama zamanına ihtiyaç duyulmasına sebep olur. Bu nedenle PS-YSA için çapraz geçerlilik yöntemleri kullanılarak, ağın ezberlemeden öğrenmesine dikkat edilmelidir. K . dereceden N girdili bir PS-YSA' nın mimarisi Şekil 8' de verilmiştir.



Şekil 8. PS-YSA'nın mimarisi

Girdi birimlerinin doğrusal kombinasyonları w_{ij} ($i = 1, 2, \dots, N$, $j = 1, 2, \dots, K$) ağırlıkları ve θ_j ($j = 1, 2, \dots, K$) yanları ile elde edilmektedir. w_{ij} i . girdiden j . gizli tabaka birimine giden ağırlığı, θ_j ise j . gizli tabaka birimi için yan değerini göstermektedir. Gizli tabaka birim sayısı kadar doğrusal kombinasyon, doğrusal aktivasyon fonksiyonundan geçerek gizli tabakaların çıktıları oluşturulmaktadır. h_j j . gizli tabaka biriminin çıktısını gösterir ve aşağıdaki formül ile hesaplanır.

$$h_j = f_1\left(\sum_{i=1}^N w_{ij}x_i + \theta_j\right), \quad j = 1, 2, \dots, K \quad (1.1)$$

Burada $f_1(x) = x$ şeklinde doğrusal aktivasyon fonksiyonunu göstermektedir. Ağırlık çıktısı ise $f_2(x) = \frac{1}{1+\exp(-x)}$ lojistik aktivasyon fonksiyonu olmak üzere aşağıdaki formül ile hesaplanır.

$$\hat{y} = f_2\left(\sum_{j=1}^K h_j\right) = \frac{1}{1+\exp(-\sum_{j=1}^K h_j)} \quad (1.2)$$

BÖLÜM 2. KAYNAK ARAŞTIRMASI

Yapay sinir ağları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar programlama sistemleridir. Bu amaç ile YSA üzerindeki çalışmalar, ilk olarak insan beynini oluşturan biyolojik veri üniteleri ve sinir sistemimizin en temel ögesi olan nöronların modellenmesi ve bu modellerin bilgisayar sistemlerinde uygulanması ile başlamıştır.

YSA, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgi türetebilme, yeni bilgi oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan doğrudan gerçekleştirmek amacı ile geliştirilen algoritmalar olarak tanımlanabilir (Öztemel, 2003).

YSA'nın en önemli özelliğinin veriden öğrenebilme kabiliyeti olduğu söylenebilir. Öğrenme işlemi, genellikle nöronları birbirine bağlayan ağırlıkların ve yan değerlerinin de optimize edilmesi sürecidir. Bu süreç, YSA'nın öğrenme süreci olarak adlandırılmaktadır. YSA'nın öğrenme sürecinde; öncelikle ilgili girdilere karşılık gelen çıktılar hesaplanır. Hesaplanan bu çıktılar ile bu çıktılara karşılık gelen hedef değerler karşılaştırılarak bir hata değeri hesaplanır. Bununla birlikte ağırlık ve yan değerleri güncellenerek süreç ilgili durdurma koşulları da göz önünde bulundurulur ve ağırlık öğrenme süreci tamamlanır. YSA'nın optimal ağırlık ve yan değerlerinin elde edilmesi bir optimizasyon problemidir. Literatürde, YSA'nın öğrenme sürecinde en sık kullanılan algoritmalar geri yayılım öğrenme algoritması ve Levenberg-Marquardt öğrenme algoritmaları olmasına rağmen, bu algoritmaların türev bilgisine ihtiyaç duyması ve karmaşık işlemlere dayanması nedeni ile son yıllarda bu algoritmaların yerine türev bilgisine ihtiyaç duymayan ve kullanımı geri yayılım öğrenme algoritması ve Levenberg-Marquardt öğrenme algoritmasına göre daha basit olan genetik algoritma (GA), parçacık sürü optimizasyonu (PSO), armoni

arama algoritması, yapay arı kolonisi, yapay yarası algoritması (YYA) ve diferansiyel gelişim algoritması (DGA) gibi birçok sezgisel optimizasyon algoritması yapay sinir ağlarının eğitiminde sıklıkla kullanılmaktadır.

Bir biyolojik nöronun yapay matematiksel modeli ilk kez McCulloch ve Pitts (1943) tarafından oluşturulmuştur. Bu nöron modelden hareketle, farklı özelliklere, yapılar ve kullanımlara sahip birçok yapay sinir ağı geliştirilmiştir. YSA ile ilgili ilk çalışmalar ise tek tabakalı yapay sinir ağları ile başlamıştır. Tek tabakalı yapay sinir ağları sadece girdi ve çıktı katmanlarından oluşurlar. Her ağ, bir veya birden çok girdiye ve bir çıktıya sahiptir ve ağ içerisindeki her bağlantının bir ağırlığı vardır. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusaldır ve genellikle sınıflama problemi için kullanılmıştır. Bununla birlikte girdi tabakası ile çıktı tabakası arasındaki ilişkinin doğrusal olmadığı bir problemi, tek katmanlı algılayıcılarla çözmek mümkün olmamıştır. (Bas, 2014). Bu sorundan hareketle doğrusal olmayan bir problemi çözmek için literatürde sıklıkla kullanılan ve temelini Rosenblatt (1958) tarafından önerilen basit algılayıcının oluşturduğu çok katmanlı algılayıcı modeli Rumelhart ve ark. (1986) tarafından önerilmiştir. Önerilen bu modele, geriye yayılım modeli de denilmektedir. İleriki yıllarda, Basu ve Ho (1999), Labib (1999), Plate (2000), Zhang ve ark. (2000) tarafından birçok YSA modeli önerilmiştir. Bu YSA türlerinin yanında Yadav ve ark. (2006) tarafından önerilen genelleştirilmiş ortalama nöron modele, Yadav ve ark. (2007) tarafından önerilen çarpımsal nöron modele, Shiblee ve ark. (2010) tarafından önerilen geometrik ortalama nöron modele, Yolcu ve ark. (2015) tarafından önerilen budanmış ortalama nöron modele dayalı ve Eğrioğlu ve ark. (2019) tarafından önerilen medyan nöron modele dayalı YSA modelleri gibi modeller de mevcuttur.

Zaman serisi öngörü problemi için, toplamsal birleştirme fonksiyonuna sahip çok katmanlı algılayıcı yapay sinir ağları (ÇKA-YSA) literatürde oldukça sık kullanılmaktadır. Bir diğer sık kullanılan YSA türü de Yadav ve ark. (2007) tarafından önerilen ve çarpımsal birleştirme fonksiyonuna sahip çarpımsal nöron modelli YSA'dır (ÇNM-YSA). ÇNM-YSA, ÇKA-YSA ile karşılaştırıldığında daha az parametreye ve daha az hesaplama zamanı sahip olduğu bilinmektedir. Ayrıca

literatürdeki birçok çalışmada ÇNM-YSA'nın ÇKA-YSA'dan daha iyi sonuçlar ürettiği bilinmektedir.

Literatürde sıklıkla kullanılan bir diğer YSA türü de toplamsal ve çarpımsal birleştirme fonksiyonlarına dayanan yüksek dereceli sinir ağlarıdır. Literatürde, Giles ve Maxwell (1978) tarafından önerilen unit neural network, Durbin ve Rumelhart (1989) tarafından önerilen product unit neural network, Shin ve Gosh (1995) tarafından önerilen Ridge Polinomial (RP) YSA ve Shin ve Gosh (1991) tarafından önerilen Pi-Sigma yapay sinir ağları (PS-YSA) sık kullanılan yüksek dereceli sinir ağlarından biridir. Bu yüksek dereceli sinir ağlarından biri olan PS-YSA, bir girdi tabakasından, birimlerin lineer toplamaları ile hesaplanan bir gizli tabakadan ve birimlerin çarpımları ile hesaplanan bir çıktı tabakasından oluşur. Pi-Sigma terimi, bu ağların girdi bileşenlerinin toplamının çarpımının kullanılmasından kaynaklanmaktadır. PS-YSA tıpkı ÇKA-YSA ve ÇNM-YSA gibi zaman serisi öngörü problemi için sıklıkla kullanılmaktadır. Husaini ve ark. (2014) PS-YSA'yı, Batu Pahat, Malezya için 2005-2009 döneminin günlük sıcaklık tahmini için kullanmışlardır. Ghazali ve Al-Jumeily (2009) finansal zaman serisi öngörüsü için PS-YSA ve RP YSA'yı (RP-YSA) kullanmışlardır. Ghazali ve ark. (2006) ve Ghazali ve ark. (2008) finansal zaman serisi problemi için RP-YSA'yı kullanmışlardır. Akdeniz ve ark. (2018) lineer olmayan zaman serisi öngörüsü için geri beslemeli bir PS-YSA önermişlerdir.

PS-YSA'nın eğitimi için ise, Yong ve Wei (2008) melez bir genetik algoritma, Nayak ve ark. (2014b) PSO ve GA'nın birlikte kullanıldığı yeni bir melez algoritma, Nayak ve ark. (2015), PSO ve gradyent descent'e dayalı melez bir öğrenme algoritması ve Nayak ve ark. (2015) ise GA'yı kullanmışlardır.

Literatürde PS-YSA için GA, PSO ve bazı melez öğrenme algoritmaları kullanılmasına rağmen henüz PS-YSA'nın eğitimi için DGA kullanılmamıştır. Bu tez kapsamında zaman serisi öngörü problemi için kullanılacak olan PS-YSA'nın eğitimi ilk kez DGA ile gerçekleştirilecektir.

Tezin ikinci bölümünde YSA ile ilgili temel kavramlara, üçüncü bölümünde ise DGA hakkında bazı bilgilere değinilmiştir. Dördüncü bölümde, önerilen yöntemin algoritması adım adım verilmiştir. Beşinci bölümde uygulama ana başlığı altında, önerilen yöntemin farklı gerçek hayat zaman serilerine uygulanmasından elde edilen sonuçlar literatürdeki bazı yöntemler ile karşılaştırılarak önerilen yöntemin performansı değerlendirilmiştir. Son olarak altıncı bölümde ise elde edilen sonuçlar sonuç ve tartışma ana başlığı altında değerlendirilmiştir.



BÖLÜM 3. MATERYAL VE YÖNTEM

3.1. ÖNERİLEN YÖNTEM

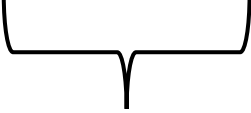
Literatürde, farklı türde YSA; sınıflama, modelleme ve öngörü gibi birçok farklı uygulama alanlarında kullanılmaktadır. Özellikle zaman serisi öngörü probleminde ÇKA-YSA, ÇNM-YSA ve PS-YSA gibi birçok YSA türü oldukça yaygın ve başarılı bir şekilde kullanılmaktadır.

Bu tez kapsamında zaman serisi öngörü problemi için kullanılan PS-YSA'nın eğitimi için ilk defa DGA kullanılmıştır. Önerilen yöntemin algoritması adımlar halinde aşağıdaki gibi verilebilir.

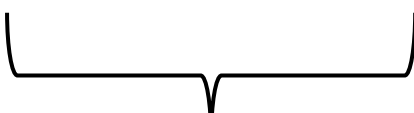
Adım 1 PS-YSA'nın öğrenme sürecinde kullanılan parametrelerinin belirlenmesi sağlanır. Bu parametreler; girdi sayısı (m), PS-YSA'nın derecesi (drc), kromozom sayısı (ks), bir kromozomda bulunan gen sayısı (gs) ve çaprazlama oranı (cor)'dır.

Adım 2 başlangıç popülasyonu oluşturulmadan önce, DGA'nda her bir çözüme karşılık gelecek olan kromozomlardan herhangi birini tanımlayalım. PS-YSA'nın öğrenme sürecinde, DGA'nın bir kromozomun genleri PS-YSA'nın ağırlık ve yan değerlerinden oluşmaktadır. PS-YSA'nın ağırlık ve yan değerlerinden oluşan bir kromozomun Şekil 9'da verilmiştir.

Genler	1	...	$(m \times drc)$	$(m \times drc) + 1$	...	$(m \times drc) + drc$
Ağırlık ve Yanlar	w_1	...	$w_{m \times drc}$	θ_1	...	θ_{drc}



Ağırlık
Değerleri



Yan
Değerleri

Şekil 9. Bir kromozomun yapısı

Şekil 9’ da, 1’ den $(m \times drc)$ ’ye kadar olan genler bir kromozomda bulunan ağırlık değerlerini ve $(m \times drc)+1$ ’ den $(m \times drc)+drc$ ’ ye kadar olan genler ise bir kromozomda bulunan yan değerlerini göstermektedir. Toplamda bir kromozomda, $(m \times drc)+drc$ adet gen bulunmaktadır.

O halde başlangıç popülasyonunun oluşturulmasında, $U(0,1)$ aralığında sürekli düzgün dağılım yardımı ile üretilen $(m \times drc)+drc$ adet genden oluşan ks adet kromozom rasgele üretilir.

Adım 3 her bir kromozom için uygunluk fonksiyonu değerinin hesaplanması sağlanır. Öncelikle her bir gizli tabaka için (h_j) j . gizli tabaka biriminin çıktısını nöronun net sinyalleri Eşitlik (3.1) ile hesaplanır.

$$h_j = f_1\left(\sum_{i=1}^N w_{ij}x_{t-i} + \theta_j\right), \quad j = 1, 2, \dots, K \quad (3.1)$$

Burada, w_{ij} i . Kromozomun j . Gen değerine karşılık gelen ağırlık değerini, x_{t-i} i . girdiye karşılık gelen gecikmeyi ve θ_j , i . Kromozoma karşılık gelen yan değerini göstermektedir. $f_1(x) = x$ ise doğrusal aktivasyon fonksiyonunu göstermektedir.

Son olarak, lojistik aktivasyon fonksiyonu (f_2) olmak üzere, $f_2(x) = \frac{1}{1+\exp(-x)}$, ağın çıktısı ise Eşitlik (3.2) ile hesaplanır.

$$\hat{x} = f_2(\prod_{j=1}^K h_j) = \frac{1}{1+\exp(-\prod_{j=1}^K h_j)} \quad (3.2)$$

Son olarak her bir kromozom için, uygunluk fonksiyonu olarak belirlenen ve Eşitlik (3.3)' de verilen Hata Kareler Ortalaması Karekök (*HKOK*) değeri hesaplanır.

$$HKOK = \sqrt{\frac{\sum_{t=1}^n (x_t - \hat{x}_t)^2}{n}} \quad (3.3)$$

Burada n, x_t ve $\hat{x}_t$ sırasıyla; öğrenme örneği sayısı, gözlenen değer ve öngörü değerini göstermektedir.

Adım 4 çaprazlama ve mutasyon operatörleri başlangıç popülasyonunda bulunan her bir kromozoma sıra ile uygulanır.

Mutasyon operatörünün uygulanması için öncelikle bir ilgili kromozom ve ilgili kromozomun dışında rasgele üç kromozom olmak üzere toplamda dört kromozom belirlenir. İlgili kromozom dışında seçilen bu rasgele kromozomun ilk ikisi birbirinden çıkarılır ve çıkarma işlemi sonucu elde edilen yemi kromozoma fark vektörü adı verilir. Daha sonra elde edilen bu fark vektörü F ile çarpılır ve ağırlıklandırılmış fark vektörü adı verilen vektör elde edilir. Elde edilen ağırlıklandırılmış fark vektörü rasgele olarak seçilmiş üçüncü kromozom ile toplanır ve toplam vektör adı verilen kromozom elde edilir ve mutasyon işlemi tamamlanır.

Toplam vektör ve ilgili kromozomun her bir geni karşılıklı ilgili genleri göz önünde bulundurularak karşılaştırılarak aday kromozom elde edilir. Öncelikle bir çaprazlama oranı belirlenir ve daha sonra düzgün dağılım yardımı ile 0 ile 1 arasında gen sayısı kadar rasgele sayı üretilir. Eğer üretilen rasgele sayı, cor ' dan küçük ise gen toplam vektörden değilse ilgili kromozomdan alınır ve aday kromozom elde edilir. Aday kromozomun uygunluk değeri hesaplanır.

Adım 5 aday kromozomun uygunluk değeri ile ilgili kromozomun uygunluk değeri karşılaştırılır. En düşük HKOK değerine sahip olan kromozom yeni jenerasyona aktarılır. Çaprazlama ve mutasyon operatörleri popülasyonda bulunan tüm kromozomlara uygulanır.

Adım 6 eğer maksimum iterasyon sayısına ulaşıldıysa veya en iyi uygunluk değerine sahip kromozomun uygunluk değeri önceden belirlenen bir hata değerinden (ε) küçük ise süreç akış durdurulur. Aksi halde ise “Adım 3” gidilir.

Algoritma durduğunda, optimal ağırlık ve yan değerleri popülasyonda bulunan en düşük HKOK değerine sahip olan kromozomdan elde edilir.

BÖLÜM 4. ARAŞTIRMA BULGULARI

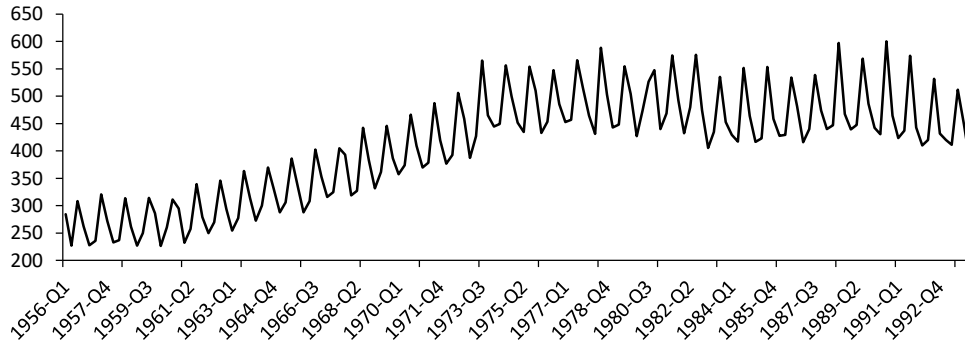
4.1. UYGULAMALAR

Bu tez çalışmasında önerilen yöntemin, performansının değerlendirilmesi için iki gerçek hayat zaman serisi çözümlemesi gerçekleştirilmiştir. Bu zaman serilerinden ilki 1957 yılının ikinci çeyreği ile 1994 yılının ilk çeyreği arasındaki gözlem değerlerine sahip Avustralya bira tüketimi (AUST) zaman serisi ve ikincisi ise 2002 yılının Ocak ayı ile 2013 yılının Aralık ayları arasında aylık olarak gözlemlenen Türkiye elektrik tüketimi (TEC) zaman serileridir. Çözümlenen bu iki zaman serisi önerilen yöntemin dışında literatürde mevcut olan birçok yöntem ile karşılaştırılmıştır. Bu amaçla, HKOK ve Eşitlik (4.1)' de verilen Ortalama Mutlak Yüzde Hata kriteri yöntemlerin karşılaştırılmasında kullanılmıştır.

$$OMYH = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (4.1)$$

4.1.1. Avustralya Bira Tüketimi Zaman Serisi Analizi

Önerilen yöntemin performansının değerlendirilmesinde, Şekil 10' da verilen AUST zaman serisi analiz edilmiştir.



Şekil 10. AUST Zaman serisi grafiği

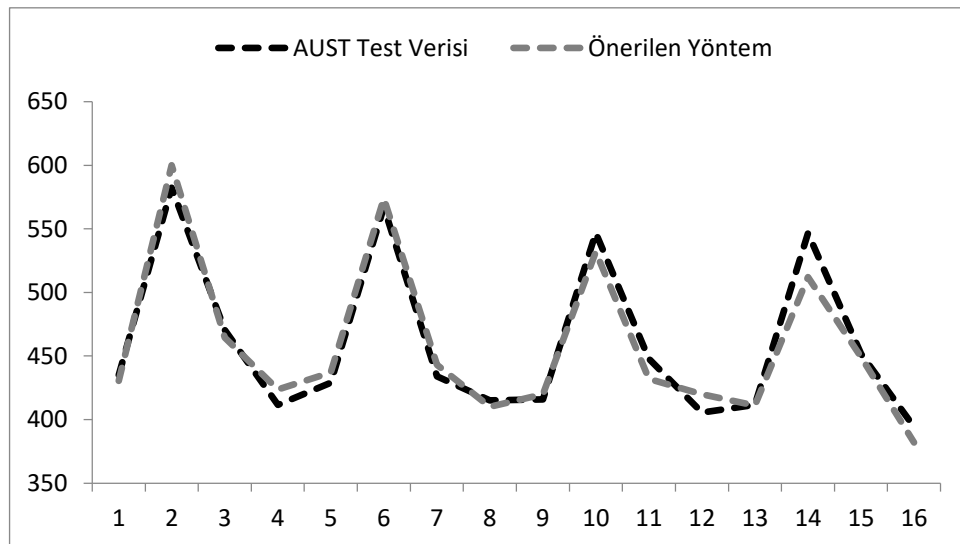
AUST zaman serisinin son 16 gözlemi test kümesi olarak alınmıştır. AUST zaman serisinin çözümlenmesinde modelin girdi sayısı 4 ile 12 arasında dört arttırılarak, modelin derecesi 2 ile 5 arasında birer arttırılarak, çaprazlama oranı 0.1 ile 0.5 arasında 0.1 arttırılarak ve kromozom sayısı da 30 ile 100 arasında on arttırılarak denenmiştir. Ayrıca, iterasyon sayısı 200 olarak belirlenmiştir. Yapılan denemeler sonucunda en iyi çözüm, girdi sayısının 12, model derecesinin 4 ve kromozom sayısının 40 ve çaprazlama oranının 0.2 olduğu durumda elde edilmiştir.

AUST zaman serisi önerilen yöntemin dışında Geri yayılım öğrenme algoritmasına dayalı ÇNM-YSA (GY-ÇNM-YSA), PSO'ya dayalı çok katmanlı YSA (PSO-ÇK-YSA), Radyal tabanlı YSA (RT-YSA), PSO'ya dayalı ÇNM-YSA (PSO-ÇNM-YSA), Çok katmanlı ileri beslemeli YSA (ÇK-İB-YSA), Elman (1990) tarafından önerilen Elman Tip geri beslemeli YSA (E-YSA), Aladağ ve ark. (2013) tarafından önerilen PSO'ya dayalı çarpımsal mevsimsel YSA (ÇAR-S-YSA), Budanmış ortalama nöron modeline dayanan çok katmanlı ileri besleme YSA (BO-ÇK-İB-YSA), Shin ve Gosh (1991) tarafından önerilen PSO'ya dayalı PS-YSA (PSO-PS-YSA), Bas (2016) tarafından önerilen DGA'ya dayalı ÇNM-YSA (DGA-ÇNM-YSA), Yolcu ve ark. (2013) tarafından önerilen doğrusal ve doğrusal olmayan YSA (D&DO-YSA), Eğrioğlu ve ark. (2015) tarafından önerilen geri beslemeli ÇNM-YSA (GÇNM-YSA), Bas ve ark. (2016) tarafından önerilen YYA'na dayalı ÇNM-YSA (YYA-ÇNM-YSA), Cagcag Yolcu ve ark. (2017) tarafından önerilen otoregresif katsayılı ÇNM-YSA (OK-ÇNM-YSA), Gündoğdu ve ark. (2015) tarafından önerilen Gauss aktivasyon fonksiyonuna dayalı ÇNM-YSA (Gauss-ÇNM-YSA), Akdeniz ve ark. (2018) tarafından önerilen geri beslemeli PS-YSA (GB-PS-YSA) aşağıda verilen yöntemlerle analiz edilmiştir.

Tablo 1. AUST test verisi için tüm yöntemlerden elde edilen HKOK ve OMYH değerleri

Yöntemler	HKOK	OMYH
GY-ÇNM-YSA	74,2551	0,0983
PSO-ÇK-YSA	44,7780	0,0856
RT-YSA	41,7000	0,0686
PSO-ÇNM-YSA	26,7831	0,0489
ÇK-İB-YSA	24,1052	0,0476
E-YSA	22,6581	0,0436
ÇAR-S-YSA	22,1700	0,0394
BO-ÇK-İB-YSA	21,0623	0,0399
PSO-PS-YSA	20,0886	0,0352
DGA-ÇNM-YSA	19,7819	0,0372
D&DO-YSA	18,7888	0,0357
GÇNM-YSA	17,8573	0,0329
ABA-MNM-ANN	17,7054	0,0323
AC-S-MNM	17,2930	0,0309
Gauss-ÇNM-YSA	15,837	0,0300
GB-PS-YSA	15,7100	0,0282
Önerilen Yöntem	13,1928	0,0226

Tablo 1 incelendiğinde, önerilen yöntem hem HKOK hem de OMYH kriterleri bakımından tüm yöntemler arasında en iyi performansa sahiptir. Ayrıca, AUST test kümesine ait gerçek gözlemlerin ve AUST test verileri için önerilen yöntemden elde edilen tahminlerin grafiği Şekil 11' de verilmiştir.

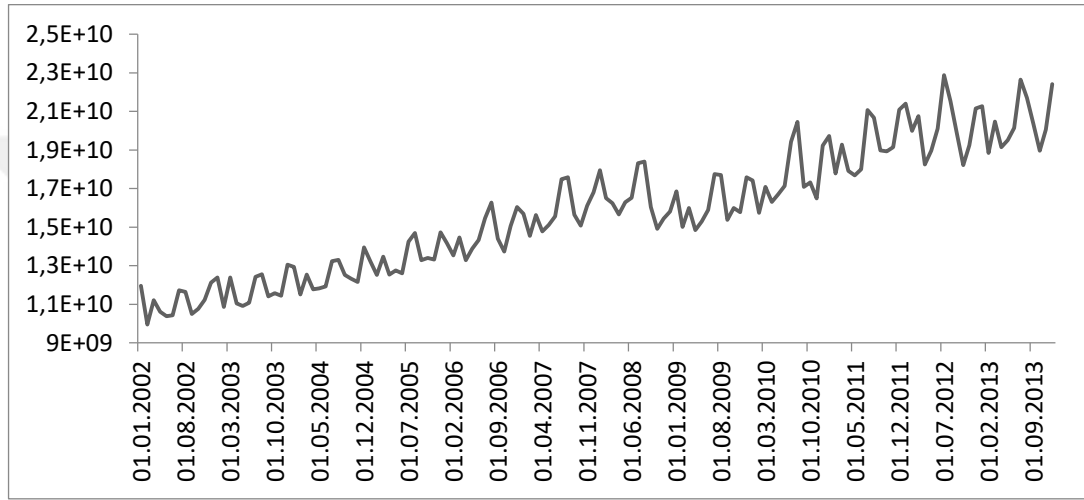


Şekil 11. AUST test kümesi ile önerilen yöntemden elde edilen öngörülerin birlikte grafiği

Şekil 11’ de, AUST test kümesine ait gerçek gözlemler ile AUST test verileri için önerilen yöntemden elde edilen tahminlerin oldukça uyumlu olduğu görülmektedir.

4.1.2. Türkiye Elektrik Tüketimi Zaman Serisi Analizi

Önerilen yöntemin performansının değerlendirilmesinde son olarak, Şekil 12’ de verilen TEC zaman serisi analiz edilmiştir.



Şekil 12. TEC Zaman serisi grafiği

TEC zaman serisinin son 12 gözlemi test kümesi olarak alınmıştır. TEC zaman serisinin çözülmesinde modelin girdi sayısı 5 ile 15 arasında birer arttırılarak, modelin derecesi 2 ile 5 arasında birer arttırılarak, çaprazlama oranı 0.1 ile 0.5 arasında 0.1 arttırılarak ve kromozom sayısı da 30 ile 100 arasında on arttırılarak denenmiştir. Ayrıca, iterasyon sayısı 200 olarak belirlenmiştir. Yapılan denemeler sonucunda en iyi çözüm, girdi sayısının 13, model derecesinin 2 ve kromozom sayısının 40 ve çaprazlama oranının 0.2 olduğu durumda elde edilmiştir.

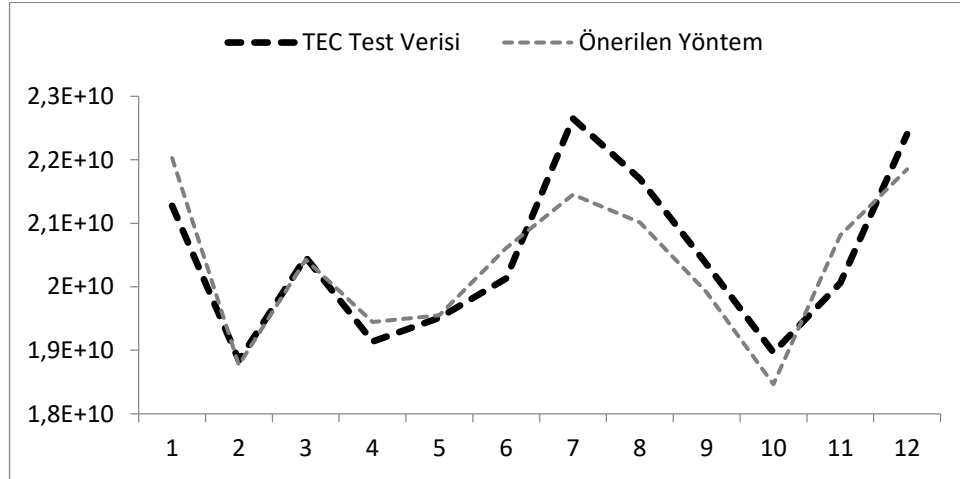
TEC zaman serisi önerilen yöntemin dışında; GY-ÇNM-YSA, ÇK-İB-YSA, YYA-ÇNM-YSA, PSO-ÇK-YSA, D&DO-YSA, ÇNM-YSA, Eğrioğlu ve Dalar (2017) tarafından önerilen deterministik eğilim ve mevsimsellik bileşenlerine sahip YSA (DT&M-YSA), DT&M-YSA, PSO-PS-YSA, PSO-ÇNM-YSA, GB-PS-YSA yöntemleri ile analiz edilmiştir. TEC zaman serisinin analizi sonucu, önerilen yöntem

ile tüm bu yöntemlerden elde edilen HKOK ve OMYH değerleri Tablo 2' de verilmiştir.

Tablo 2. TEC test verisi için tüm yöntemlerden elde edilen HKOK ve OMYH değerleri

Yöntemler	HKOK	OMYH
GY-ÇNM-YSA	4243944603	0,1954
ÇK-İB-YSA	1065900000	0,398
YYA-ÇNM-YSA	924149635	0,373
PSO-ÇK-YSA	915190000	0,339
D&DO-YSA	820980000	0,254
ÇNM-YSA	813260000	0,301
DT&M-YSA	810300852	0,247
PSO-PS-YSA	697763268	0,281
PSO-ÇNM-YSA	672790000	0,289
GB-PS-YSA	651383248	0,222
Önerilen Yöntem	585195649	0,23

Tablo 2 incelendiğinde, tıpkı TEC zaman serisinde olduğu gibi önerilen yöntem hem HKOK hem de OMYH kriterleri bakımından tüm yöntemler arasında en iyi performansa sahiptir. Ayrıca, TEC test kümesine ait gerçek gözlemlerin ve TEC test kümesi verileri için önerilen yöntemden elde edilen tahminlerin grafiği Şekil 13' te verilmiştir.



Şekil 13. TEC test kümesi ile önerilen yöntemden elde edilen öngörülerin birlikte grafiği

Şekil 13' te TEC test kümesine ait gerçek gözlemler ile TEC test verileri için önerilen yöntemden elde edilen tahminlerin uyumlu olduğu söylenebilir.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Bu tez kapsamında öngörü problemi için kullanılan PS-YSA'nın eğitimi literatürde kullanılan birçok sezgisel optimizasyon algoritmadan farklı olarak ilk kez DGA ile gerçekleştirilmiş ve literatürdeki birçok yapay sinir ağı yönteminden daha iyi öngörü sonuçlarına ulaşılmıştır. Önerilen yöntemin öngörü performansını test etmek için ise gerçek hayat zaman serilerinin analizi için iki farklı veri seti kullanılarak çözümlerler yapılmıştır. Yapılan çözümlerler sonucunda, önerilen yöntemin performansının literatürdeki mevcut birçok yapay sinir ağı modelinden daha iyi olduğu sonucuna varılmıştır.

İleriki çalışmalarda, PS-YSA'nın diferansiyel gelişim algoritması ile eğitimi sürecinde farklı mutasyon operatörleri kullanılabilir ya da diferansiyel gelişim algoritması farklı yapay sinir ağlarının eğitiminde kullanılabilir.

KAYNAKLAR

- Aladag, C.H., Yolcu, U., and Egrioglu, E. 2013. A new multiplicative seasonal neural network model based on particle swarm optimization, *Neural Processing Letters* 37(3), 251-262.
- Akdeniz, E., Egrioglu, E., Bas, E., Yolcu, U. 2018. An ARMA type Pi-Sigma artificial neural network for nonlinear time series forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, 8(2) 121-132.
- Anderson, J. A. 1972. A simple neural network generating on interactive memory. *Mathematical Biosciences*, 14, 197-220.
- Bas, E. 2016. The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting, *Journal of Artificial Intelligence and Soft Computing Research*, 6(1), 5-11.
- Broomhead, D. S., & Lowe, D. 1988. Radial basis-function, multi-variable functional interpolation and adaptive networks. *Royal signals and radar establishment memorandum*, 41-48.
- Cagcag, Yolcu, O., Bas, E., Egrioglu, E., Yolcu, U. 2017. Single multiplicative neuron model artificial neural network with autoregressive coefficient for time series modelling. *Neural Processing Letters*, 1-15.
- Egrioglu, E., Aladag, C.H., Yolcu, U and Bas, E. 2015. Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting, *Neural Processing Letters*, 41(2), 249-258.
- Efe, & Kaynak, O. 1999. A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems. *Mechatronics*, 9 (3), 287-300.
- Efe, & Kaynak, O. 2000. *Yapay Sinir Ağları ve Uygulamaları: Boğaziçi Üniversitesi*.
- Efe, Abaoglu, E., & Kaynak, O. 1999. Analysis and Design of a Neural Network Assisted Nonlinear Controller for a Bioreactor. *International Journal of Robust and Nonlinear Control*, 9(11), 799-815.
- Egrioglu, E., Aladag, CH., Yolcu, U., Bas, E., Dalar, A.Z. 2017. A new neural network model with deterministic trend and seasonality components for time series forecasting. *Advances in Time Series Forecasting*, 2, 76-92.
- Elmas, P. D. Ç. 2007. *Yapay Zeka Uygulamaları*. Ankara: Seçkin.

- Farely, B. G., & Clark, W. A. 1954. Simulation of self-organizing systems by digital computers. *IEEE Transactions of Professional Group of Information Theory*, PGIT, 4, 76-84.
- Fukushima, K. 1986. A neural network model for selective attention in visual pattern recognition. *Biological Cybernetics*, 55(1), 5-15.
- Fukushima, K., Miyake, S., & Ito, T. 1983. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, SMC-13.
- Ghazali, R., & al-Jumeily, D. 2009. Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction. in M. Zhang (Ed.). *Artificial Higher Order Neural Networks for Economics and Business*. Hershey, New York: IGI Global, 271-293.
- Ghazali, R., Hussain, A. J., Liatsis, P. and Tawfik, H. 2008 . “The application of ridge polynomial neural network to multi-step ahead financial time series prediction,” *Neural Computing and Applications*, vol. 17, no. 3, 311–323.
- Gundogdu, O., Egrioglu, E., Aladag, C.H. and Yolcu, U. 2015. Multiplicative neuron model artificial neural network based on gauss activation function, *Neural Computing and Applications*, 27(4), 927935
- Hebb, D. O. 1949. The organization of behaviour. The first stage of perception: growth of the assembly, 4, 60-78.
- Hopfield, J. J. 1982 a. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National, Academy of Sciences*, 79, 2554-2558.
- Hopfield, J. J. 1982 b. Neurons with graded response have collective computational properties like those of two state neurons. *Proceedings of the National, Academy of Sciences*, 81, 3088-3092.
- Husaini, N. A., Ghazali, R., Nawi, N. M., Ismail, L. H., Deris, M. M., and Herawan, T. 2014. “Pi-Sigma neural network for a one-step-ahead temperature forecasting,” *International Journal of Computational Intelligence and Applications*, vol. 13, no. 4, 1450023-1-1450023– 16
- James, W. 1890. *Psychology (Briefer Course)*. Association, 16, 253- 279.
- Kohonen, T. 1972. Correlation matrix memories. *IEEE Transactions on Computers*, 21(4), 353-359.
- Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.

- McClelland, J. L., & Rumelhart, D. E. 1986. Parallel distributed processing, explorations in the microstructure of cognition. Psychological and biological models, MIT Press, Cambridge, MA, 2.
- McCulloch, W. S., & Pitts, W. A. 1943. A logical calculus of the ideas immanent in nervous activity. *Buttetin of Mathematics and Biophysics*, 5, 115-133.
- Nayak, J., Naik, B., and Behera, H.S., and Abraham, A. 2015. Particle swarm optimization based higher order neural network for classification, *Smart Innovation, Systems and Technologies*, 31, 401414.
- Nayak, J., Naik, B., and Behera, H.S., A. 2014b. Hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification. *International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, art. no. 6993082, 878885.
- Nayak, S. C., Misra, B. B., and Behera, H. S. 2015 . “A pi-sigma higher order neural network for stock index forecasting,” *Computational Intelligence in Data Mining*, vol. 2, 311–319.
- Storn, R., & Price, K. 1995. DE-a simple and efficient adaptive scheme for global optimization over continuous space. *Technical report*, 25(6), 95-102.
- Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychoanalytic Review*, 65, 386-408.
- Rumelhart, D. E., & McClelland, J. L. 1986. Parallel distributed processing, explorations in the microstructure of cognition. *Foundations*, MIT Press Cambridge, MA, 1.
- Rumelhart, D. E., & McClelland, J. L. 1988. Parallel distributed processing, explorations in the microstructure of cognition, *A handbook of models, programs and exercies*. MIT Press, Cambridge, MA.
- Rumelhart, D. E., Hinton, D. E., & Williams, R. J. 1986. Learning representation by backpropagating errors. *Nature*, 323(9), 533-536.
- Shin Y, Ghosh, J., 1995. Ridge polynomial networks. *IEEE TransNeural Netwo*, 6(3), 610–622
- Specht, D. F. 1988. Probabilistic neural networks for classification, mapping or associative memory. *IEEE Conference on Neural Networks*, 1, 525-532.
- Specht, D. F. 1991. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6), 568-576.

- Yolcu, U., Egrioglu, E., Aladag, C.H. 2013. A new linear & nonlinear artificial neural network model for time series forecasting, *Decision Support Systems*, 1340–134.
- Yong, N. and Wei, D. 2008. A hybrid genetic learning algorithm for Pi– sigma neural network and the analysis of its convergence, In: *IEEE fourth international conference on natural computation*, 19–23.
- Widrow, B., & Hoff, M. E. 1960. Adaptive switching circuits. *WESTCON Convention, Record Part IV*, 96-104.



EKLER

Ek- Avustralya Bira Tüketimi ile Türkiye Elektrik Tüketimi Zaman Serisi Analizinin MATLAB programı akış diyagramı aşağıdaki gibi tanımlanmıştır.

```
function
[rmsetest,mapetest,msebest,pgbest,rmse,yhatg,yhattestg,rn,mape,itn]=pisgmpsotestsi
mpr(xt,m,drc,ps,vmaps,w1,w2,c1i,c2i,c1f,c2f,itr,ntest)
% m girdi sayısı
msebestEski=10^6;kk=0;
x0=xt;
n0=length(xt);
xtest=x0((n0-ntest+1):n0);
xt=x0(1:(n0-ntest));
x=(xt-min(xt))/(max(xt)-min(xt));
saat=clock;
rand('seed',saat(6)*10000000);
%rand('seed',254740000);
rn=rand('seed');
p=m*drc+drc;
%p parametre sayısı
n=length(x);
A=unifrnd(0,1,ps,p);
V=unifrnd(-vmaps,vmaps,ps,p);
%Pso paramterleri program hızlı çalışması için başta oluşturuluyor;
w=zeros(1,itr);
c1=zeros(1,itr);
c2=zeros(1,itr);
for k=1:ps
    W(1,:)=A(k,1:drc);
    for j=2:m
```

```

        W(j,:)=A(k,drc*(j-1)+1:drc*j);
    end
    b=A(k,drc*m+1:drc*m+drc);
    yhat=multifpigm(x,W,b,m,drc);
    nh=length(yhat);
    for i=1:nh
        hata(i)=(yhat(i)-x(n-nh+i))^2;
    end
    mse(k)=mean(hata);
end
%en iyi parçacığı saklıyor.
MSEegt=min(mse);
for i=1:ps
    if MSEegt==mse(i)
        dd=i;
        break
    end
end
pgbest=A(dd,:);
msebest=mse(dd);
pid=A;
msepid=mse;
i22=0;
for i1=1:itr
    i1
    i22=i22+1;
    % Güncelleme
    w(i1)=(w1-w2)*((itr-i1)/itr)+w2;
    c1(i1)=(c1f-c1i)*(i1/itr)+c1i;
    c2(i1)=(c2f-c2i)*(i1/itr)+c2i;
    for i2=1:ps
        for i3=1:p

```

```

        V(i2,i3)=V(i2,i3)*w(i1)+c1(i1)*unifrnd(0,1)*(pid(i2,i3)-
A(i2,i3))+c2(i1)*unifrnd(0,1)*(pgbest(i3)-A(i2,i3));
        V(i2,i3)=min(vmaps,max(-vmaps,V(i2,i3)));
        A(i2,i3)=A(i2,i3)+V(i2,i3);
    end
end
if i22>=100
    A=unifrnd(0,1,ps,p);
    V=unifrnd(-vmaps,vmaps,ps,p);
    i22=0;
end
for k=1:ps
    W(1,:)=A(k,1:drc);
    for j=2:m
        W(j,:)=A(k,drc*(j-1)+1:drc*j);
    end
    b=A(k,drc*m+1:drc*m+drc);
    yhat=multifpigm(x,W,b,m,drc);
    nh=length(yhat);
    for i=1:nh
        hata(i)=(yhat(i)-x(n-nh+i))^2;
    end
    mse(k)=mean(hata);
end
%en iyi parçacığı saklıyor.
MSEegt=min(mse);
for i=1:ps
    if MSEegt==mse(i)
        dd=i;
        break
    end
end
if MSEegt<msebest

```

```

        pgbest=A(dd,:);
        msebest=mse(dd);
        [i1,msebest]
    end
    for j1=1:ps
        if mse(j1)<=msepid(j1)
            pid(j1,:)=A(j1,:);
            msepid(j1)=mse(j1);
        end
    end
    if abs((msebestEski-msebest)/msebest)<10^-3
        kk=kk+1;
    end
    if kk>50
        break
    end
    msebestEski=msebest;
end
itn=i1;
W(1,:)=pgbest(1:drc);
for j=2:m
    W(j,:)=pgbest(drc*(j-1)+1:drc*j);
end
b=pgbest(drc*m+1:drc*m+drc);
yhat=multifpigm(x,W,b,m,drc);
nh=length(yhat);
yhatg=(yhat)*(max(xt)-min(xt))+min(xt);
for i=1:nh
    hata(i)=(yhatg(i)-xt(n-nh+i))^2;
    hata2(i)=abs((yhatg(i)-xt(n-nh+i))/xt(n-nh+i));
end
rmse=mean(hata)^0.5;
mape=mean(hata2);
x=(x0-min(x0))/(max(x0)-min(x0));
yhattum=multifpigm(x,W,b,m,drc);
nt=length(yhattum);
yhattest=yhattum((nt-ntest+1):nt);
yhattestg=(yhattest)*(max(xt)-min(xt))+min(xt);
for i=1:ntest
    hata3(i)=(xtest(i)-yhattestg(i))^2;
    hata4(i)=abs((xtest(i)-yhattestg(i))/xtest(i));
end
rmsetest=(mean(hata3))^0.5;
mapetest=mean(hata4);
end

```

ÖZGEÇMİŞ

1993'te Zonguldak'ta doğdu. İlk ve Ortaöğretimini Zonguldak Gazi İlköğretim okulunda tamamladı. Lise öğrenimini Zonguldak Anadolu Teknik ve Endüstri Meslek lisesinde tamamladı. 2011'de Karadeniz Teknik Üniversitesi Trabzon MYO Bilişim Teknolojileri bölümünü kazandı. 2013'te Karadeniz Teknik Üniversitesinden mezun oldu. 2013 Kasım ayında Giresun Üniversitesi Fen-Edebiyat Fakültesi İstatistik bölümünü kazandı. 2015'te Giresun Üniversitesinden mezun oldu. 2016 Ocak ayında Giresun Üniversitesi Fen Bilimleri Enstitüsü Matematik ABD Yüksek Lisans programına başladı. Aktif olarak Emniyet Genel Müdürlüğü Kaçakçılık ve Organize Suçlarla Mücadele Daire Başkanlığı Analiz Şube Müdürlüğünde Polis Memuru olarak görev yapmakta.