

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**ISI YALITIM LEVHASI ÜRETİMİ ŞİŞİRME
PROSESİNDE BONCUK YOĞUNLUĞUNUN YAPAY
SİNİR AĞLARI İLE TAHMİNİ**

**Hazırlayan
Burçak BULDANLI**

**Danışman
Dr. Öğr. Üyesi Feyza GÜRBÜZ**

Yüksek Lisans Tezi

**Temmuz 2021
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**ISI YALITIM LEVHASI ÜRETİMİ ŞİŞİRME
PROSESİNDE BONCUK YOĞUNLUĞUNUN YAPAY
SİNİR AĞLARI İLE TAHMİNİ**

(Yüksek Lisans Tezi)

**Hazırlayan
Burçak BULDANLI**

**Danışman
Dr. Öğr. Üyesi Feyza GÜRBÜZ**

**Temmuz 2021
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Burçak BULDANLI

“Isı Yalıtım Levhası Üretimi Şişirme Prosesinde Boncuk Yoğunluğunun Yapay Sinir Ağları ile Tahmini” adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ne uygun olarak hazırlanmıştır.

Hazırlayan

Danışman

Burçak BULDANLI

Dr. Öğr. Üyesi Feyza GÜRBÜZ

Endüstri Mühendisliği ABD Başkanı

Prof. Dr. Lale ÖZBAKIR

TEŞEKKÜR

Tez çalışmamın hazırlanmasında yoğun çalışmalarına rağmen desteğini esirgemeyen, farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan sayın hocam Dr. Öğr. Üyesi Feyza GÜRBÜZ'e çok teşekkür ederim.

Çalışmalarım süresince maddi, manevi pek çok fedakârlık gösteren, destekleri ile her zaman yanımda olan, dualarını eksik etmeyen annem Nurten BULDANLI'ya ve babam İsmail BULDANLI'ya, manevi destekleri ile her zaman yanımda olan ablam Burçin BULDANLI KARA'ya ve eniştem Kadir KARA'ya, varlıklarıyla güç kazandığım yeğenlerime minnettarlığımı belirterek en içten teşekkürlerimi sunarım.

Burçak BULDANLI

Temmuz 2021, KAYSERİ

ISI YALITIM LEVHASI ÜRETİMİ ŞİŞİRME PROSESİNDE BONCUK YOĞUNLUĞUNUN YAPAY SİNİR AĞLARI İLE TAHMINİ

Burçak BULDANLI

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Temmuz 2021

Danışman: Dr. Öğr. Üyesi Feyza GÜRBÜZ

ÖZET

Yapay zeka uygulamalarının kullanımı her geçen gün artmaktadır. Bu uygulamalardan biri, insan beyni dikkate alınarak geliştirilen ve öğrenme yoluyla birçok probleme çözüm sunan yapay sinir ağlarıdır. Bu çalışmada, Türkiye’de kimya sektöründeki bir firmadan ön şişirme prosesine ait veri seti temin edilmiştir. Eğitim ve test olmak üzere iki gruptan oluşan bu veri seti üzerinde farklı metodlar uygulanarak ağırlık, ölçülen yoğunluk, çevrim süresi, buhar süresi ve buhar sıcaklığı gibi parametreler ile bloklarda istenilen yoğunluğu elde etmek için bloğu oluşturan boncukların hangi yoğunlukta şişirilmesi gerektiğinin tahmin edilmesi amaçlanmıştır. Bu amaç doğrultusunda, MATLAB R2015 programı yardımıyla yapay sinir ağı teknikleri, PolyAnalyst programı yardımıyla veri madenciliği kural bulma algoritması ve SPSS programı yardımıyla regresyon analizi ile veriler analiz edilmiş ve sonuçlar yorumlanmıştır. Elde edilen sonuçlara bakıldığında zaman en yüksek R^2 değeriyle en iyi sonucu yapay sinir ağlarının verdiği görülmüştür. Gizli katmanlarında sırasıyla tansig, logsig; çıkış katmanında purelin aktivasyon fonksiyonuna sahip, birinci gizli katmanda 10 ve ikinci gizli katmanda 7 nöron bulunduran ileri beslemeli ağ yapısı en iyi sonucu veren ağ olarak seçilmiştir. Ardından bu ağ yapısı dikkate alınarak ağ test edilmiştir ve elde edilen çıktı ile gerçek çıktı karşılaştırılmıştır. Sonuçlar, ağın hata oranının düşük, performansının yüksek olduğunu ve tahminleme için yapay sinir ağlarının olumlu etkisini göstermektedir.

Anahtar Kelimeler: Yapay Sinir Ağları, Makine Öğrenmesi, Veri Madenciliği, Üretim, Yoğunluk Tahmini

ESTIMATING BEAD DENSITY IN THE BLOWING PROCESS OF THERMAL INSULATION BOARD PRODUCTION WITH ARTIFICIAL NEURAL NETWORKS

Burçak BULDANLI

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, July 2021

Supervisor: Asst. Prof. Dr. Feyza GÜRBÜZ

ABSTRACT

The use of artificial intelligence applications is increasing day by day. One of these applications is artificial neural networks, which are developed considering the human brain and offer solutions to many problems through learning. In this study, a data set for the pre-expansion process was obtained from a company in the chemical industry in Turkey. By applying different methods on this data set, which consists of two groups, training and testing, it is aimed to predict at what density the beads forming the block should be inflated in order to obtain the desired density in the blocks with parameters such as weight, measured density, cycle time, steam time and steam temperature. For this purpose, data were analyzed with artificial neural network techniques with the help of MATLAB R2015 program, data mining rule finding algorithm with the help of PolyAnalyst program and regression analysis with the help of SPSS program and the results were interpreted. When the results obtained are examined, it is seen that artificial neural networks give the best results with the highest R^2 value. The feedforward network structure, which has tansig, logsig activation functions in the hidden layers, and purelin activation functions in the output layer, has 10 neurons in the first hidden layer and 7 neurons in the second hidden layer, has been chosen as the network that gives the best results. Then, considering this network structure, the network was tested and the obtained output was compared with the actual output. The results show that the error rate of the network is low, the performance is high, and the positive effect of neural networks for prediction.

Keywords: Artificial Neural Network, Machine Learning, Data Mining, Production, Density Estimation

İÇİNDEKİLER

ISI YALITIM LEVHASI ÜRETİMİ ŞİŞİRME PROSESİNDE BONCUK YOĞUNLUĞUNUN YAPAY SİNİR AĞLARI İLE TAHMİNİ

BİLİMSEL ETİĞE UYGUNLUK.....	ii
YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY	iv
TEŞEKKÜR.....	v
ÖZET	vi
ABSTRACT.....	vii
İÇİNDEKİLER	viii
KISALTMALAR.....	xi
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ	xiii
GİRİŞ	1

1. BÖLÜM

LİTERATÜR ÇALIŞMASI

1.1 Literatür Araştırması	3
---------------------------------	---

2. BÖLÜM

YÖNTEM

2.1 Yapay Sinir Ağları	9
2.1.1 Genel Bakış.....	9
2.2 Veri Madenciliği.....	12
2.3 Regresyon Analizi	13
2.4 Yapay Sinir Ağlarının Yapısı ve Temel Elemanları	13

2.4.1 Yapay Sinir Ağlarının Yapısı.....	13
2.4.2 Yapay Sinir Ağlarının Temel Elemanları.....	14
2.4.2.1 Biyolojik Sinir Hücreleri	14
2.4.2.2 Yapay Sinir Hücreleri.....	16
2.5 Yapay Sinir Ağı Modelleri.....	17
2.5.1 İleri Beslemeli Yapay Sinir Ağları	18
2.5.2 Geri Beslemeli Yapay Sinir Ağları	18
2.6 Çok Katmanlı Algılayıcılar	19
2.6.1 Çok Katmanlı Algılayıcıların Mimarisi	19
2.6.2 Geri Yayılım Algoritması	20
2.6.3 Eğitim Algoritmaları	21
2.6.4 Çok Katmanlı Algılayıcıların Çalışma Prosedürü.....	22
2.7 Yapay Sinir Ağının Eğitimi ve Öğrenmesi	23
2.7.1 Yapay Sinir Ağlarında Öğrenme Yöntemleri	24
2.7.1.1 Denetimli Öğrenme	24
2.7.1.2 Denetimsiz Öğrenme	25
2.7.1.3 Takviyeli Öğrenme.....	25
2.8 Yapay Sinir Ağlarında Öğrenme Kuralları.....	26
2.8.1 Hebb Kuralı.....	26
2.8.2 Hopfield Kuralı	27
2.8.3 Kohonen Kuralı.....	27
2.8.4 Delta Kuralı	28
2.9 Yapay Sinir Ağlarının Tasarımı	28
2.9.1 Yapay sinir ağı ağ yapısının seçimi	28
2.9.2 Öğrenme Algoritmasının Seçimi	29
2.9.3 Gizli Katman Sayısının Belirlenmesi	29
2.9.4 Nöron Sayısının Belirlenmesi.....	29

2.9.5 Normalizasyon.....	30
2.9.5.1 İstatistiksel veya Z-score Normalizasyonu	30
2.9.5.2 Min-Max Normalizasyonu.....	31
2.9.5.3 Medyan Normalizasyonu.....	31
2.9.5.4 Sigmoid Normalizasyonu.....	31
2.9.5.5 D-Min-Max Normalizasyonu	32
2.9.6 Performans Fonksiyonu Seçimi.....	32

3. BÖLÜM

UYGULAMA VE BULGULAR

3.1 Uygulama Yapılan Firma Hakkında Bilgi.....	34
3.2 Veri Madenciliği PolyAnalyst Uygulaması	35
3.3 Regresyon Analizi SPSS Uygulaması	35
3.4 Yapay Sinir Ağı MATLAB Uygulaması	36

4. BÖLÜM

SONUÇ VE DEĞERLENDİRME

4.1 Sonuç ve Değerlendirme	52
KAYNAKÇA	55
EKLER.....	61
EK-1 (Verilerin bir kısmı)	61
EK-2 (Çalışmada kullanılan MATLAB kodu)	62
ÖZGEÇMİŞ.....	64

KISALTMALAR

YSA: Yapay Sinir Ağları

LM: Levenberg Marquardt

BR: Bayesian Regularization

SCG: Scaled Conjugate Gradient

MLP: Çok Katmanlı Algılayıcı

RBFNN: Radial Basis Functional Neural Network

ART: Adaptif Rezonans Teori

LVQ: Doğrusal Vektör Parçalama

MAD: Ortalama Mutlak Sapma

SSE: Toplam Karesel Hata

MSE: Ortalama Kare Hatası

RMSE: Karesel Ortalama Hata Karekök

MAPE: Ortalama Mutlak Yüzde Hatası

MATLAB: Matrix Laboratory

TABLOLAR LİSTESİ

Tablo 2.1 Biyolojik sinir sistemi elemanları ve yapay sinir ağındaki karşılıkları.....	16
Tablo 2.2 Öğrenme algoritmaları ve kullanım alanları.....	26
Tablo 2.3 Ağ yapıları ve kullanıldıkları alanlar	29
Tablo 3.1 Regresyon analizi sonuçları.....	35
Tablo 3.2 Programa tanıtılan örnek veriler	37
Tablo 3.3 Normalize edilmiş örnek veriler	38
Tablo 3.4 Gizli nöron sayısının tespiti için yapılan denemeler.....	39
Tablo 3.5 Aktivasyon fonksiyonunun ve eğitim fonksiyonunun tespiti için yapılan denemeler	40
Tablo 3.6 Gerçek değerler ile minimum hata değerine sahip ağ yapısının test edilmesi sonucu elde edilen tahmin değerlerinin karşılaştırılması	44
Tablo 3.7 İkinci gizli katmandaki nöron sayısının ve aktivasyon fonksiyonunun tespiti için yapılan denemeler	45
Tablo 3.8 Gerçek değerler ile minimum hata değerine sahip ağ yapısının test edilmesi sonucu elde edilen tahmin değerlerinin karşılaştırılması	50
Tablo 3.9 Kabul edilen ağ yapısına ait bilgiler	50
Tablo 4.1 Elde edilen R^2 değerlerinin karşılaştırılması.....	53

ŞEKİLLER LİSTESİ

Şekil 2.1. Basit bir yapay sinir ağı yapısı.....	14
Şekil 2.2. Basit bir biyolojik sinir hücresi yapısı	15
Şekil 2.3. İki nöron arasındaki bilgi akış mekanizması	15
Şekil 2.4. Yapay sinir ağı mimarisi.....	16
Şekil 2.5 İleri beslemeli yapay sinir ağları.....	18
Şekil 2.6 Geri beslemeli yapay sinir ağları	19
Şekil 2.7 Çok katmanlı yapay sinir ağ modeli	20
Şekil 2.8 Geri yayılım algoritmasının akış şeması.....	20
Şekil 2.9 Denetimli öğrenme yöntemi	24
Şekil 2.10 Denetimsiz öğrenme yöntemi	25
Şekil 2.11 Takviyeli öğrenme yöntemi	25
Şekil 2.12 Hopfield ağ mimarisi	27
Şekil 3.1 Tasarlanan ileri beslemeli çok katmanlı ağ yapısı	36
Şekil 3.2 Matlab'ta YSA eğitim sonuç ekranı	40
Şekil 3.3 Eğitim, geçerleme ve test verilerine ait hata performansları	41
Şekil 3.4 Matlab regresyon sonuçları.....	42
Şekil 3.5 Matlab'ta YSA eğitim sonuç ekranı	47
Şekil 3.6 Eğitim, geçerleme ve test verilerine ait hata performansları	48
Şekil 3.7 Matlab regresyon sonuçları.....	49

GİRİŞ

Bilgisayarların insan beyni gibi öğrenme, yorumlama, genelleme yapma vb. kabiliyetlere sahip olması ile yapay zeka uygulamalarına olan ilgi artmıştır. Bu uygulamalar sayesinde çözümü zor veya imkansız olan problemler kolaylıkla sonuçlanmıştır. Yapay zeka uygulamalarından biride yapay sinir ağıdır. Bu yöntem karmaşık yapıların tahminleme, sınıflandırma gibi problemlerine çözüm bulmak için kullanılmaktadır. Yapay sinir ağı insan beyni dikkate alınarak oluşturulmuş bir sistem olduğu için öğrenme kabiliyetine sahiptir.

Bu çalışmada ısı yalıtım levhası üretiminin hammaddesi olan boncukların hangi yoğunlukta şişirilmesi gerektiğinin tahmin edilmesi hedeflenmiştir. Tahmin için çoklu doğrusal regresyon yöntemi, veri madenciliği kural bulma algoritması ve yapay sinir ağı teknikleri kullanılmıştır. Boncukların ilk prosesi olan ön şişirme aşamasında ürün ve ürün kalitesine etki eden faktörler ele alınmıştır. Bu aşamadan temin edilen veri setinin %70'i ağı eğitimi, %30'u ağı test edilmesi için kullanılmıştır. Girdi parametreleri olarak ağırlık, ölçülen yoğunluk, çevrim süresi, buhar süresi ve buhar sıcaklığı, çıktı parametresi olarak istenilen yoğunluk seçilmiştir. Yapay sinir ağlarında tahmin problemlerinde yaygın olarak kullanılan model olduğu için ileri beslemeli çok katmanlı ağ yapısı oluşturulmuştur. Bu veriler ile oluşturulan farklı ağ yapıları MATLAB R2015 programı kullanılarak eğitilmiştir. Çalışmada performans kriteri olarak MSE ve MAPE kullanılmıştır. Eğitim tamamlandıktan sonra, en düşük hata değerine sahip ağ yapısı dikkate alınarak sisteme daha önce tanıtılmamış veriler ile ağ test edilmiştir. Veri madenciliği yazılımlarından biri olan PolyAnalyst aracı kullanılarak veriler analiz edilmiştir. Programdaki algoritmalarından, bağımlılıkları bulma (Find Dependencies) algoritması ve kümeleme analizi (Cluster Analysis) ön işleme aracı olarak kullanılmıştır. Sınıflandırma algoritması olarak kural bulma (Find Laws) algoritması kullanılmıştır. Eğitim verileri, bağımlılıkları bulma analizi sonucu elde edilen veriler ve kümeleme analizi sonucu elde edilen veriler olmak üzere üç farklı veri setine kural

bulma algoritması ile ilgili analizler yapılmıştır. Bir diğer yöntem olan çoklu doğrusal regresyon yönteminde istenilen yoğunluğu en iyi tahmin eden regresyon modelini bulmak için veriler analiz edilmiştir.

Çalışmanın sonraki bölümlerine bakıldığı zaman, birinci bölümde yapay sinir ağları ile tahmin konusunda literatürde yapılmış çalışmalara yer verilmiştir. İkinci bölümde yapay sinir ağları, veri madenciliği ve regresyon analizi hakkında genel bilgiler verilmiştir. Üçüncü bölümde çalışma yapılan firma hakkında bilgi verilmiştir ve ileri beslemeli geri yayılım algoritması kullanılarak en iyi ağ yapısına ulaşılmıştır. Dördüncü bölümde ise sonuç ve değerlendirmelerden bahsedilmiştir.



1. BÖLÜM

LİTERATÜR ÇALIŞMASI

1.1 Literatür Araştırması

Keskinkılıç, F., (2010), döküm parametrelerinin önemli bir kalite değişkeni olan sertlik değeri üzerindeki etkilerini yapay sinir ağları kullanarak tahmin edilmesini amaçlamıştır. Döküm kalitesinin iyileştirilmesi ve maliyetinin azaltılması için girdi olarak belirlenen döküm parametrelerine karşılık gelen sertlik değerleri yapay sinir ağına öğretilmiş ve birbirleri ile arasındaki ilişki net olarak ortaya konulamayan parametrelerin optimum seviyeleri belirlenmiştir. Tedarikçi kısıtından dolayı parametrelerden bazıları kontrol edilmezken istenen sertlik değeri için kontrol edilebilen diğer parametrelerin hangi seviyede olması gerektiği oluşturulan ağ sayesinde belirlenebilmektedir. Sonuç olarak en etkili parametrelerin bulunması için kullanılan yöntem ve algoritmalar arasında en iyi sonucu genetik algoritmanın verdiği görülmüştür [1].

Mashhadban vd. (2016), elyafların kendini sıkıştıran betonun performansı üzerindeki etkilerini araştırmışlardır. İki tür elyaf içeren dokuz beton karışımı ve güçlendirilmemiş numuneler test edilerek karşılaştırılmıştır. Bu veriler ileri beslemeli yapay sinir ağları ile eğitilmiştir. Çelik liflerin, mekanik özelliklerle ilişkili olarak, polifenilen sülfür liflerine kıyasla daha iyi performans gösterdiği gözlemlenmiştir [2].

Akkaya, G., Gökçen, T. (2006), atölye çizelgeleme tasarımında benzetim ve yapay sinir ağı yöntemlerini kullanarak, tezgah seçimi, malzeme taşıma sistemi seçimi ve sistemde kullanılacak öncelik kuralı seçimi için bir sistem geliştirmişlerdir. Yapay sinir ağı modeli olarak geri yayılım algoritmasını kullanmışlardır. Çalışmada, ilk olarak gerçeğe

uygun sonuçlar üretebilen, eğitilmiş yapay sinir ağları elde edilmiştir. Yapay sinir ağını eğitmek ve gerekli olan örnekleri elde etmek için bilgisayar ortamında benzetim yönteminden faydalanmışlardır [3].

Satyanarayana vd. (2017), bu çalışmada takviye ve deformasyonun karıştırma ile sentezlenen kırmızı çamur nano partikül takviyeli alüminyum matriks kompozitlerin hacimsel aşınması üzerindeki etkisinin tahmin edilmesinde yapay sinir ağı yaklaşımı kullanılmıştır. Kompozitin aşınma oranını tahmin etmek için regresyon modeli ve yapay sinir ağı modeli geliştirilmiştir ve yapay sinir ağı tahminlerinin diğer modellere göre ölçülen değerlerle mükemmel bir uyum içinde olduğu gözlenmiştir [4].

Wilson, R., Sharda, R. (1994), diskriminant analizi ve yapay sinir ağları yöntemlerini karşılaştırmışlardır. Çok hisseli ya da halka açık şirketlerin finansal geleceklerini tahmin etmek için diskriminant analizi ve lojistik regresyon metotlarını veri madenciliğinin yapay sinir ağları metoduyla karşılaştırıldığında en iyi sonucu yapay sinir ağlarının verdiği görülmüştür [5].

Kant, G., Sangwan, K. (2015), bu çalışmada ilk olarak, kesme karakteristiğinin performans özelliği olarak deneysel verileri, orta karbonlu çeliklerin öğütülmesi sırasında yapay sinir ağları tekniği kullanılarak tahmini modeli geliştirmek için kullanılmıştır. Ardından YSA sonuçları bağıl hata analizi kullanılarak deneysel verilerle karşılaştırılır. Daha sonra, geliştirilen model bazı temsili hipotez testleri kullanılarak doğrulanmıştır. Son olarak, işleme parametrelerinin kesim enerjisine etkisi 3D grafikler kullanılarak incelenmiştir. Tahmin edilen sonuçların deneysel verilere çok yakın olduğu, hipotez testi sonuçları, bir tahmin modeli olarak YSA'nın modelleme bakış açısından istatistiksel olarak tatmin edici bir uyum iyiliğine sahip olduğu ve 3B grafikler, işlem süresinin, yüksek enerji artışlarında bile toplam enerji tüketiminin azalmasından dolayı enerji tüketimine hâkim olduğunu göstermektedir sonuçlarına varılmıştır [6].

Sudha vd. (2015), yem üretiminde bazı sorunların üstesinden gelinmesine ve daha iyi performans elde edilmesine yardımcı olacağından proses parametrelerinin optimizasyonu amaçlamışlardır. Çalışma aynı zamanda çok dönemli planlamanın artılarını ve eksilerini ve bunun maliyetler üzerindeki etkisini de analiz etmektedir. Bu

nedenle, envanter yönetimi, yem tedarik zincirinde yeterince umut vaat eden önemli yönetim sorunlarından biridir [7].

Vedaramana vd. (2017), işlem verilerinin zaman geçmişini temel alan bir işlemin doğrusal olmayan giriş-çıkış dinamiklerini simüle etmek için yapay sinir ağı kullanmışlardır. CCD ve YSA yöntemleri ile tanımlanan optimum ekstraksiyon koşullarının bir karşılaştırması yapılmıştır. Sonuç olarak maksimum ekstraksiyon yüzdesi için optimize edilmiş bir işlem koşulları listesi elde etmek için bir CCD tasarımı başarıyla kullanılmıştır [8].

Altinkaya vd. (2014), yapay sinir ağı tekniklerini kullanarak Türkiye'nin tek demiryolu üretim tesisi olan Kardemir Ray ve Profil Haddehanesinde ray üretim sürecini modelleyip yeni bir ray tipi üretmek için gerekli olan parametre değerlerinin elde edilmesini amaçlamışlardır. Sonuç olarak, raylı haddeleme gibi son derece karmaşık ve pahalı bir sistemde YSA kullanarak zaman ve para kaybına neden olmadan yeni bir ürün simülasyonu yapmak mümkündür ve YSA algoritmasının, sürecin modellenmesinde ve simülasyonunda alternatif bir yöntem olarak kullanılabileceği gösterilmiştir [9].

Hatipoğlu vd. (2013), çinko maliyetinin hesaplanması için gerekli olan kaplama kalınlığının tahmin edilmesini amaçlamışlardır. Maliyet fonksiyonlarının tahmin edilmesinde çoklu regresyon ve yapay sinir ağı yöntemi kullanılmıştır. Kaplama kalınlığı için seçilen model yapısı; 9 girdi değişkenli, tek ara katmanlı, tek çıktılı ileri beslemeli geri yayılım ağıdır. Çalışmanın sonucunda çoklu regresyon yöntemiyle elde edilen sonuçların düşük performansla çalıştığı, yapay sinir ağı modelinde ise gerçek değerler ile tahmini değerler arasındaki fark hipotez testiyle karşılaştırılmıştır ve %95 güven aralığında değerler arasında anlamlı bir farklılık olmadığı görülmüştür [10].

Cai vd. (2009), Çin'in Guangzhou kentindeki bir arterin yakınlarındaki saatlik hava kirletici konsantrasyonlarının tahmin edilmesini amaçlamışlardır. Bu hava kirletici konsantrasyonları etkileyen dört faktör vardır. Bunlar trafikle ilgili, arka plan konsantrasyonu, meteorolojik ve coğrafidir. Konsantrasyonun tahmin edilmesinde yapay sinir ağı yöntemi ile çoklu doğrusal regresyon kullanılmıştır. Elde edilen veriler geri yayılım algoritmasına dayalı yapay sinir ağı ile eğitilmiş ve ardından test edilmiştir.

İki yöntemden elde edilen sonuçlar karşılaştırıldığında yapay sinir ağlarının daha iyi sonuç verdiği görülmüştür [11].

Mia vd. (2016), bu çalışmada kesme hızı, ilerleme hızı, malzeme sertliği ve kesme koşulları olmak üzere dört girdisi olan ağ levenberg-marquardt (LM), bayesian regularization (BR) ve scaled conjugate gradient (SCG) algoritmaları ile eğitilmiş ve karar vermek için en düşük ortalama kare hatası (RMSE) dikkate alınmıştır. Bayesian regularization (BR) ile eğitilen ağın en düşük RMSE değerini verdiği görülmüştür [12].

Li vd. (2015), menopoz semptomlarının şiddetinin (KMI puanı) yapay sinir ağı yöntemlerini kullanarak tahmin edilmesini amaçlamışlardır. Girdiler olarak yaş, eğitim durumu, aylık gelir, iş durumu, vücut kitle indeksi, menarş yaşı, eşitlik, doğum kontrolü ve kronik rahatsızlığı olmak üzere dokuz faktör kabul edilmiştir. Çıktı olarak ise KMI puanı dikkate alınmıştır. En iyi sonucu bulmak için iki farklı eğitim algoritması kullanılarak oluşturulan ağların sonuçları karşılaştırılmıştır. Ardından KMI puanı üzerinde önemli etkisi olan faktörleri belirlemek için duyarlılık analizi yapılmıştır. Bunun sonucunda yaş, aylık gelir, eğitim durumu, iş durumu ve kronik rahatsızlığı olmak üzere beş faktörün semptomlar üzerinde önemli bir etkiye sahip olduğu görülmüştür [13].

Şimşek, E., Alkay, T., (2019), bu çalışmada yapay sinir ağları tekniği kullanılarak Kocaeli ilinde bulunan atıksu arıtma tesisinde çıkış suyunun parametrelerinin tahmini için model geliştirilmiştir. Seçilen ağın yapısı; 5 girdi parametrelili, tek gizli katmanlı ve 3 çıktı parametrelidir. En uygun algoritmayı bulmak için çeşitli denemeler yapılmıştır. Tansig ve purelin aktivasyon fonksiyonlarının 0.94, 0.96 ve 0.95 regresyon değerleri ile en iyi sonucu verdiği görülmüştür [14].

Dombaycı, A., (2010), yapay sinir ağı tekniklerini kullanarak Denizli’de özel olarak tasarlanmış bir evin bir saatte ısınma enerjisi tüketiminin tahmin edilmesi için model geliştirmişlerdir. Oluşturulan ağ 3 katmanlı ileri beslemeli bir ağ yapısıdır. Öğrenme fonksiyonu olarak ‘trainlm’ seçilmiştir ve gizli katmanda tansig, çıkış katmanında purelin aktivasyon fonksiyonu kullanılmıştır. 2004-2007 yıllarına ait veriler ile ağ eğitilmiş ve 2008 yılına ait veriler ile de ağ test edilmiştir. En iyi sonucu RMSE (1,2575), R^2 (0,9907) ve MAPE (0,2081) değerleri ile gizli katmanında 29 nörona sahip ağ yapısı vermiştir [15].

Zain vd. (2010), frezelemede yüzey pürüzlülüğü performansının tahmin edilmesini amaçlamışlardır. Üç giriş parametresi ve bir çıkış parametresi mevcuttur. Eğitim fonksiyonu olarak 'traingdx', öğrenme fonksiyonu olarak 'learnngdx' ve aktivasyon fonksiyonu olarak 'logsig' seçilmiştir. Tek gizli katman ve iki gizli katman dikkate alınarak farklı denemeler yapılmıştır. Yapılan denemeler sonucunda 3-1-1 ağ yapısının en iyi sonucu verdiği görülmüştür [16].

Jorjani vd. (2008), yapay sinir ağları tekniklerini kullanarak kömür flotasyon konsantresinin yanıcı değerinin ve geri kazanımının tahmin edilmesini amaçlamışlardır. Yanıcı değerin tahmini için seçilen ağın yapısı; 5 girdi parametrelili, iki gizli katmanlı ve 1 çıktı parametrelidir (5-30-2-1). R^2 (eğitim) = 1, R^2 (test) = 0,91 yeterli görülmüştür. Geri kazanımının tahmini için seçilen ağın yapısı; 7 girdi parametrelili, iki gizli katmanlı ve 1 çıktı parametrelidir (7-10-3-1). R^2 (eğitim) = 1, R^2 (test) = 0,95 yeterli görülmüştür [17].

Tiryaki vd. (2014), ısıtma işlemi görmüş ağaçların gücünün tahmin edilmesini amaçlamışlardır. En iyi sonucu veren ağ yapısını bulmak için MATLAB programı kullanılarak çeşitli denemeler yapılmıştır. Ardından yapay sinir ağı sonuçları ile çoklu doğrusal regresyon sonuçları karşılaştırılmıştır. Yapay sinir ağlarının daha iyi sonuçlar verdiği görülmüştür [18].

Nabipour, M., (2018), ikili soğutucu karışımlarının yüzey geriliminin tahmin edilmesini amaçlamışlardır. Girdi parametresi olarak kritik sıcaklık, kritik basınç, kritik hacim, karışımların sıcaklığı ve asentrik olmak üzere 5 faktör kabul edilmiştir. Oluşturulan ağ 3 katmanlı ileri beslemeli bir ağ yapısıdır. Öğrenme fonksiyonu olarak 'trainlm' seçilmiştir ve gizli katmanda tansig, çıkış katmanında purelin aktivasyon fonksiyonu kullanılmıştır. En iyi sonucu 0,9997 korelasyon katsayısı ile gizli katmanında 19 nörona sahip ağ yapısı vermiştir [19].

Ekonomou, L., (2010), bu çalışmada yapay sinir ağı teknikleri kullanılarak Yunanistan enerji tüketiminin tahmin edilmesi amaçlanmıştır. Çok katmanlı ağ yapısı oluşturularak farklı ağ yapıları denenmiştir. Geliştirilen ağ yapısında girdi parametresi olarak 2005-2008 arasındaki veriler kullanılmıştır ve bu veriler ile ağ eğitilmiştir. 2010, 2012 ve 2015 yıllarında Yunanistan enerji tüketiminin tahmin edilmesi istenmiştir. Ardından yapay sinir ağı sonuçları ile doğrusal regresyon ve destek vektör makinesi yöntemleri

ile elde edilen sonuçlar karşılaştırılmıştır. Yapay sinir ağlarının daha iyi sonuçlar verdiği görülmüştür [20].



2. BÖLÜM

YÖNTEM

2.1 Yapay Sinir Ağları

2.1.1 Genel Bakış

Yapay sinir ağları, insan beyninin bilgiyi işleme özelliğinden yola çıkarak, öğrenmeye dayalı yeni bilgiler üretebilme ve keşfetme gibi kabiliyetleri taklit etmek amacıyla tasarlanmış bilgisayar programlarıdır. Bu özelliklerin gerçekleştirilmesi geleneksel programlama yöntemleriyle oldukça zordur [21].

Literatürde yapay sinir ağları ile ilgili tanımlamalardan bazıları aşağıdaki gibidir:

Yapay sinir ağları, insan beyni dikkate alınarak geliştirilmiş, ağırlıklı bağlantılar sayesinde birbirlerine ilişkilendirilen, paralel çalışan ve aralarında bilgi akışı olan bir bilgi işleme sistemidir. Bu nedenle yapay sinir ağları için biyolojik sinir hücrelerinden oluşan sinir ağlarının bilgisayar ile taklit edilmesidir denilebilir [22,23].

Yapay sinir ağları, insan beyninin bilgiyi işleme şekline benzer olarak öğrenme, sınıflandırma, ilişkilendirme gibi konularda oldukça başarılı sonuçlar verdiği görülmektedir. Ayrıca öğrenme yolu ile yeni bilgilerin keşfi gibi yetenekleri hiçbir destek almadan otomatik olarak gerçekleştiren bilgisayar programlarıdır [24].

Günümüzde yapay sinir ağları uygulamaları, insan ihtiyaçlarının çeşitli alanlarında kullanılmasıyla popüler bir hal almıştır. Kuruluşlar tarafında çeşitli alanlardaki sorunları çözmek için sinir ağlarına yatırımlar yapılmaktadır. Yapay sinir ağları, tarım, bilim, eğitim, finans, güvenlik, mühendislik, ticari mal ve sanat konularında ortaya çıkan problemleri çözme yeteneğine sahiptir [25].

Yapay sinir ağıları, insan beyni dikkate alınarak öğrenme sürecinin taklit edilmesiyle elde edilen verilerden yeni veri üretebilme gibi işlevlerin matematiksel olarak modellenmesi ile gerçekleştirilen veri işleme sistemidir [26]. Yapay sinir ağlarının en önemli özelliği karmaşık yapıdaki ilişkilere sahip sistemlerin geçmiş bilgileri dikkate alınarak öğrenme yolu ile tahminleme, sınıflandırma, kümeleme, modelleme, optimizasyon gibi birçok probleme çözüm bulmasıdır [27].

Yapay sinir ağları kullanıcının yeteneklerine gerek duymadan, öğrenmeyi kendi kendine gerçekleştiren sistemlerdir. Yapay sinir ağlarının birçok çeşidi bulunmaktadır fakat bunlardan bazıları diğerlerine göre daha yaygın olarak kullanılmaktadır. İnsan beyninin fonksiyonel özelliklerine benzer şekilde; öğrenme, sınıflandırma, modelleme, tahminleme ve optimizasyon gibi konularda uygulanmaktadır. Yapay sinir ağının en önemli görevi, ağa gönderilen bir giriş veri setine karşılık gelecek bir çıkış veri seti belirlemektir. Bunu gerçekleştirebilmek için ağ, ilgili durumun örnekleri ile eğitilerek benzer durumlar için genelleme yapabilme kabiliyetini elde eder ve böylelikle benzer durumlara karşılık gelen çıkış veri setleri belirlenir [24].

Yapay sinir ağı kullanmanın bazı avantajları aşağıdaki gibidir [28]:

- ❖ Geleneksel yöntemlerde bilgiler veri tabanında saklanırken yapay sinir ağlarından ağın tamamında depolanabilir.
- ❖ Yapay sinir ağları eksik bilgiler olsa bile çıktı üretebilir. Ancak performansı eksik bilgilerin ne derecede önemli olduğuna bağlıdır.
- ❖ Yapay sinir ağlarında hücre bozulması çıktı üretilmesini engellemez. Sinir ağlarının bu özelliği ağı hatalara karşı dayanıklı hale getirir.
- ❖ Yapay sinir ağları eğitildikten sonra genelleme yaparak ağa daha önce tanıtılmamış verileri tahmin edebilir.

Bu tarz yöntemlerin dezavantajlarından bazıları aşağıdaki gibidir [22]:

- ❖ Uygun sinir ağını belirlemek için belirli bir kural yoktur genel olarak deneme yanılma yöntemi ile elde edilir. Eğer uygun ağ belirlenemezse düşük performanslı çözümler elde edilir. Sinir ağları kabul edilebilir çözümler üretirler yani optimum sonuca ulaşamayabilir.
- ❖ Ağ eğitiminin biteceği zamanı belirlemek için herhangi bir yöntem yoktur. Hatanın belli bir değerin altına düşmesi eğitimin tamamlanmasını sağlayabilir ancak bu en iyi öğrenmenin olduğu anlamına gelmez.

- ❖ Sinir ağının davranışlarının açıklanamaması, üretilen çözümün neden ve nasıl üretildiğine yönelik herhangi bir bilgi bulunamaması ağın sonucuna olan güveni azaltır.

2.1.2 Yapay Sinir Ağlarının Genel Özellikleri

Yapay sinir ağlarının özellikleri kullanılan ağın modeline ve algoritmalarına göre değişiklik gösterebilir ancak genel olarak özellikleri aşağıdaki gibidir [21,29]:

- ❖ Yapay sinir ağlarının temel görevi bilgisayarın öğrenmesini sağlamaktır. Öğrenilen olaylara benzer bir olay ile karşılaşıldığında benzer davranışlar sergilerler.
- ❖ Yapay sinir ağları ağa verilen örnekler aracılığıyla öğrenirler.
- ❖ Yapay sinir ağlarının en iyi şekilde çalışabilmesi için ağın ilk önce eğitilmesi ve performanslarının test edilmesi gerekmektedir.
- ❖ İstenilen sonuçlar elde edilmediği takdirde ağ tekrar eğitilebilir.
- ❖ Ağa tanıtılmamış örnekler hakkında bilgi üretebilirler. Ağ, daha önce kendisine verilen örneklerden genelleme yaparak tanımadığı örnekler hakkında bilgi üretebilir.
- ❖ Matematiksel modele ihtiyaç duymazlar.
- ❖ Eksik bilgi ile çalışabilirler. Ağ eğitildikten sonra, ağa verilen yeni örneklerde eksik bilgi olsa bile çıktı verebilirler.
- ❖ Sadece nümerik bilgiler ile çalışırlar; sembolik ifadelerin sayısal değerlere dönüştürülmesi gerekmektedir.
- ❖ Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.

2.1.3 Yapay Sinir Ağlarının Tarihçesi

1970 yılı bilim için önemli bir dönüm noktası olduğu için yapay sinir ağları ile ilgili günümüze kadar yapılmış olan çalışmaları 1970 öncesi ve 1970 sonrası diye ikiye ayırmak gerekmektedir. Yapılan bu çalışmalar kronolojik sırasına göre şu şekildedir. 1840 yılında uzun yıllar araştırılan insan beyninin işlevleri hakkında ilk yayın yayınlanmıştır. 1943 yılında hekim Worsen McCulloch ve matematikçi Walter Pitts tarafından insan beyninin bilgiyi işleme şeklini taklit eden yapay sinir ağlarının temeli atılmıştır. 1949 yılında, Donald Hebb tarafından 'Hebbian Öğrenme Kuralı' geliştirilmiştir. 1953 yılında, Frank Rosentblatt tarafından tek katmanlı ağ yapısı

(perceptron) geliştirilmiştir. Bu model yapay sinir ağları için önemlidir. Çünkü geliştirilerek daha önemli bir yapının yani çok katmanlı ağ yapılarının oluşmasına katkı sağlamıştır. 1959 yılında, Bernard Widrow ve Marcian Hoff tarafından ADALINE ve MADALINE yapay sinir ağları modelleri geliştirilmiştir. 1974 yılında, geri yayılım algoritması geliştirilmiştir. 1982 yılında, çok katmanlı ağ yapısı geliştirilmiştir. 1987 yılında, 200 kişinin katılımıyla yapay sinir ağları hakkında ilk konferans düzenlenmiştir. 1990'lı yıllardan günümüze kadar birçok çalışma yapılmıştır. Ancak yapay sinir ağlarının daha önce uygulandığı alanların aksine günlük hayatta pratik olarak kullanılmasıyla insanlara faydalı olmaya başlamıştır [22].

2.2 Veri Madenciliği

Günlük hayatımızda sürekli veri üretilmektedir. Bu veriler çok çeşitli yöntemlerle oluşturularak çeşitli veritabanlarında kayıt altına alınmaktadır. Ancak ham veri işlenmediği sürece faydalı olmayacaktır. Elde edilen büyük miktarda veri içinden anlamlı bilgiler çıkarılması veri madenciliğinin alanıdır [30]. Veri madenciliği, istatistiksel ve matematiksel teknikler kullanılarak çeşitli veritabanlarında kayıt altına alınmış verilerin işlenmesi ile anlamlı bağlantı ve kuralların keşfedilmesidir [31]. Günümüzde veri madenciliğinin; bankacılık, pazarlama, sigortacılık, telekomünikasyon, borsa, sağlık, endüstri, bilim ve mühendislik gibi birçok dalda uygulama alanı bulunmaktadır. Veri madenciliği, istatistik, matematik ve bilgisayar bilimlerinin bir kesişimidir. Bundan dolayı 'disiplinlerarası' bir disiplin olarak nitelendirilmektedir. Veri madenciliği için kullanılacak olan verilerin sürece adım adım hazırlanması gerekmektedir. Bu süreç KDD (Knowledge Discovery in Database) yani Veri Tabanı Bilgi Keşfi olarak adlandırılır [32].

Veri madenciliği sürecinde izlenmesi gereken adımlar şu şekildedir.

1. Problemin Tanımlanması
2. Verilerin Hazırlanması
3. Modelin Kurulması ve Değerlendirilmesi
4. Modelin Kullanılması
5. Modelin İzlenmesi

Veri madenciliğinde kullanılan modeller, tahmin edici ve tanımlayıcı olmak üzere iki ana başlık altında toplanabilir [33].

➤ Tahmin Edici Modeller

- Sınıflandırma
 - ✓ Karar Ağaçları
 - ✓ Yapay Sinir Ağları
 - ✓ Destek Vektör Makinesi
 - ✓ Genetik Algoritma

- Regresyon

➤ Tanımlayıcı Modeller

- Kümeleme
- Birliktelik Kuralları
- Ardışık Zaman Örüntüleri

2.3 Regresyon Analizi

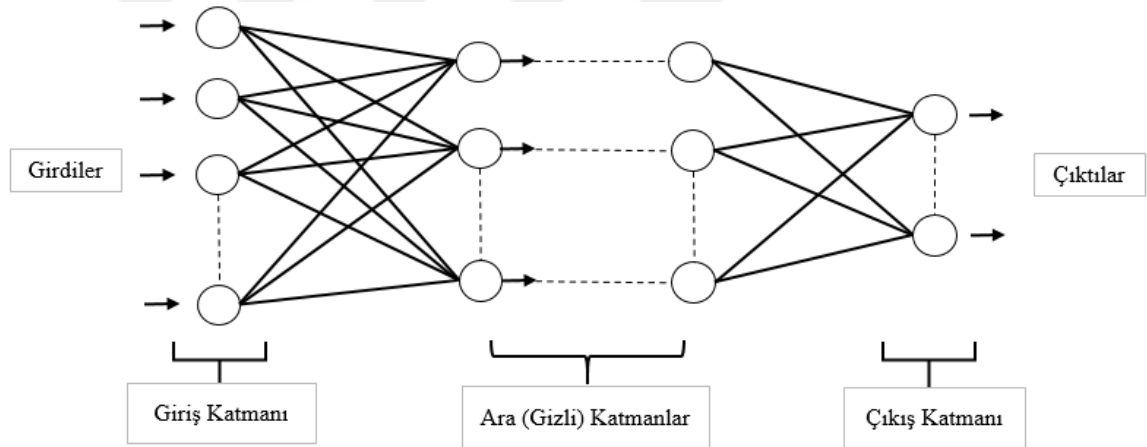
Regresyon analizi tüm alanlar için son derece önemli bir yöntemdir. Yöntemin anlaşılabilir olması ve paket programlar sayesinde sonuca hızlı ulaşılması, yapılan çalışmalarda regresyon analizine olan ilginin artmasına neden olmaktadır. Regresyon yönteminin amaçlarından biri bağımlı değişken ile bağımsız değişken(ler) arasındaki ilişkinin belirlenmesidir. Regresyon analizi basit regresyon modeli ve çoklu regresyon modeli olmak üzere olmak üzere ikiye ayrılır. Basit regresyon modeli, bir bağımlı değişken ve bir bağımsız değişkenden oluşan modeldir. Çoklu regresyon modeli ise bir bağımlı değişken ve çok sayıda bağımsız değişkenden oluşan modeldir. Bu yöntemdeki amaç bağımsız değişken(ler) dikkate alınarak bağımlı değişken için tahmin modeli oluşturmaktır [34].

2.4 Yapay Sinir Ağlarının Yapısı ve Temel Elemanları

2.4.1 Yapay Sinir Ağlarının Yapısı

Yapay sinir ağları bir dizi düğüm ve bağlantılardan oluşmaktadır. Düğümler işlemlerin gerçekleştirildiği bağlantılar ise düğümler arasında bilgi akışının gerçekleştirildiği noktalardır. Bağlantılarda bilgi akışı tek yönlü veya çift yönlü olarak gerçekleşebilir. Bu ağlarda, birbirleriyle bağlantılı nöronların bulunduğu giriş katmanı, ara (gizli)

katman(lar) ve çıkış katmanı olmak üzere genellikle üç katman bulunmaktadır. İlk katman olan giriş katmanında girdi verileri üzerinde herhangi bir işlem uygulanmamaktadır. Giriş katmanının görevi dışarıdan gelen verilerin yapay sinir ağına giriş yapmasını sağlamaktır. Bu veriler istatistikte bağımsız değişkenlere karşılık gelen girdi değişkenlerinden oluşur. Bu girdi verileri, giriş katmanı ve ara katman arasında bulunan nöronlara ait ağırlık değerleri ile çarpılır ve her biri ara katmana iletilir. Son katman olan çıkış katmanının görevi ise bilgilerin dış ortama aktarılmasını sağlamaktır. Ağ yapısındaki diğer katmanlar ise giriş katmanı ile çıkış katmanı arasında yer alan ara (gizli) katmanlardır. Bu katmanın görevi yalnızca giriş katmanından gelen sinyalleri çıkış katmanına iletmektir. Ara katmanda bulunan nöronların dış ortamla herhangi bir bağlantıları bulunmamaktadır [35,36].



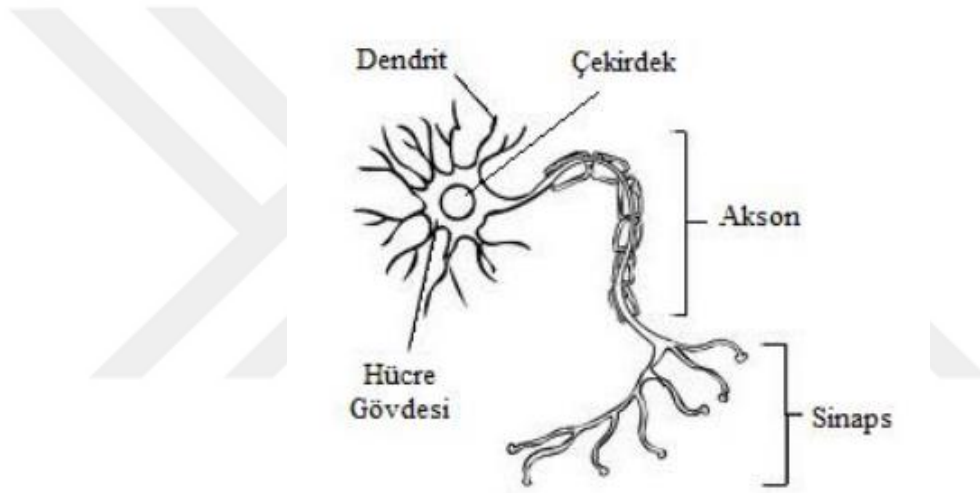
Şekil 2.1. Basit bir yapay sinir ağı yapısı

2.4.2 Yapay Sinir Ağlarının Temel Elemanları

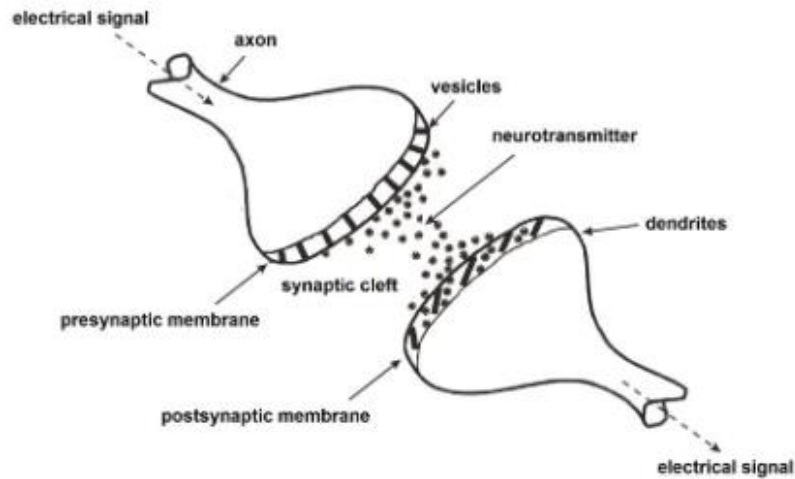
2.4.2.1 Biyolojik Sinir Hücreleri

Yapay sinir ağları, insan beyninin bilgiyi işleme şekli dikkate alınarak geliştirilmiştir. Programlama yerine insan beyni gibi uygun örnekler ile deneyimleyerek öğrenirler bu yüzden öncelikle biyolojik sinir sisteminin yapısına bir göz atmak gerekmektedir. İnsan beyni, diğer nöronlarla bağlantılı yaklaşık 100 milyar nöron içermektedir [37]. Hücre gövdesi, dendrit, akson ve sinaps olmak üzere dört bölümden oluşan nöronlar Şekil 2.2’de gösterilmiştir. Dendritler nöronların uç kısmında yer almaktadır ve görevi komşu

nöronlardan gelen bilgileri hücre gövdesine iletmektedir. Hücre gövdesine gelen bilgiler aksona iletilmektedir. Bilgiler sinaps yardımıyla komşu nöronların dentritlerine iletilmektedir. Sinaps yardımıyla komşu nöronlar arasındaki bilgi akışı Şekil 2.3'te gösterilmiştir. Elektrik sinyali şeklindeki bilgi, hücre gövdesi içinde yer alan zara iletilir. Ardından sinaps boşluğunda kimyasallar salgılanır. Burada komşu nörona gelen bilgiler belirli bir eşik değerine göre yeniden oluşturulabilir. Komşu nörona iletilen bilgi miktarı, sinaptik güç ve eşik değerine bağlıdır. Bir nörona birden çok bilgi iletebilir çünkü bir nöronun yapısında çok sayıda dentrit bulunmaktadır. Bu bilgi akışı yapay sinir ağlarının gelişmesine katkı sağlamıştır [38].



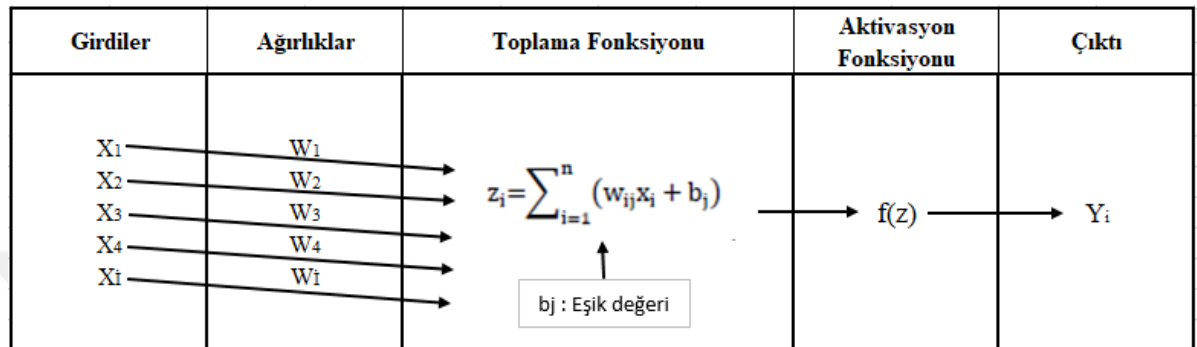
Şekil 2.2. Basit bir biyolojik sinir hücresi yapısı



Şekil 2.3. İki nöron arasındaki bilgi akış mekanizması [38]

2.4.2.2 Yapay Sinir Hücreleri

Yapay sinir ağı, biyolojik sinir hücrelerinin (nöronların) matematiksel olarak modellenmesine dayanmaktadır. Yapay sinir hücreleri girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere beş bileşenden oluşmaktadır.



Şekil 2.4. Yapay sinir ağı mimarisi [21]

Şekil 2.4'te görüldüğü gibi ağı i tane veri girişi yapılmaktadır ve X_i girdi verilerini temsil etmektedir. Yapay sinir hücresinde, girdilerin ($X_1, X_2, \dots, X_i$) her biri ağırlıklar (W_i) ile çarpılıp eşik değeri (b_j) ile toplanır ve ardından aktivasyon fonksiyonu ($f(z)$) yardımıyla çıktı (Y_i) üretilir [39].

Tablo 2.1 Biyolojik sinir sistemi elemanları ve yapay sinir ağındaki karşılıkları [27]

Biyolojik Sinir Sistemi	Yapay Sinir Ağı
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

Girdiler: Dış ortamdan yapay sinir hücresine gelen bilgilerdir. Hücreye bilgi dış ortamdan gelebileceği gibi başkısı hücrelerden veya kendisinden de gelebilir.

Ağırlıklar: Ağırlıklar yapay sinir hücresine gelen bilginin önemini ve hücreye nasıl bir etkisi olduğunu belirler. Şekil 2.4'te görüldüğü gibi ağırlık W_i , girdi X_i 'nin sinir hücresi üzerindeki etkisini göstermektedir. Ağırlıklar artı, eksi ve sıfır değerlerini alabilirler bu onların önemli veya önemsiz olduğu anlamına gelmez. Aldığı değerin artı olması hücreye etkisinin pozitif olduğu, eksi olması hücreye etkisinin negatif olduğu ve değerin

sıfır olması ise hücreye hiçbir etkisinin olmadığı anlamına gelmektedir.

Toplama Fonksiyonu: Hücreye gelen girdileri ($X_1, X_2, X_3, \dots, X_i$) ve ağırlıkları ($W_1, W_2, \dots, W_i$) kullanarak net girdiyi hesaplar. Ağırlık yapısına göre değişen çeşitli toplama fonksiyonları mevcuttur. En çok tercih edilen ağırlıklı toplama fonksiyondur. Toplama fonksiyonuna aşağıdaki fonksiyonlar örnek olarak verilebilir:

- *Toplam:* $NET = \sum (X_i * W_i) \quad (i=1,2,\dots,n)$ (2.1)

- *Çarpım:* $NET = \prod (X_i * W_i)$ (2.2)

- *Maksimum:* $NET = \text{Max} (X_i * W_i)$ (2.3)

- *Minimum:* $NET = \text{Min} (X_i * W_i)$ (2.4)

- *Çoğunluk:* $NET = \sum \text{sgn} (X_i * W_i)$ (2.5)

Aktivasyon Fonksiyonu: Genellikle doğrusal olmayan bir fonksiyon olan bu fonksiyonun amacı hücreye gelen net girdiyi işleyerek üretilecek çıktıyı belirlemektir. Hücrenin çıktı değerleri genellikle $[-1, 1]$ veya $[0, 1]$ aralığında yer alır. Aktivasyon fonksiyonu içinde birçok formül mevcuttur. Bu fonksiyonlardan bazıları aşağıdaki tabloda verilmiştir. Aktivasyon fonksiyonuna aşağıdaki fonksiyonlar örnek olarak verilebilir:

- ✓ *Step Fonksiyonu:*

$$F(NET) = (1; NET > \text{eşik değeri}, 0; NET \leq \text{eşik değeri}) \quad (2.6)$$

- ✓ *Lineer Fonksiyon:*

$$F(NET) = NET \quad (2.7)$$

- ✓ *Sinüs Fonksiyonu:*

$$F(NET) = \text{Sin} (NET) \quad (2.8)$$

- ✓ *Hiperbolik Tanjant Fonksiyonu*

$$F(NET) = (e^{NET} + e^{-NET}) / (e^{NET} + e^{-NET}) \quad (2.9)$$

Çıktılar: Aktivasyon fonksiyonu tarafından üretilen çıktı değeridir. Üretilen çıktı dış ortama, başka bir hücreye veya kendisine girdi olarak iletilebilir [22].

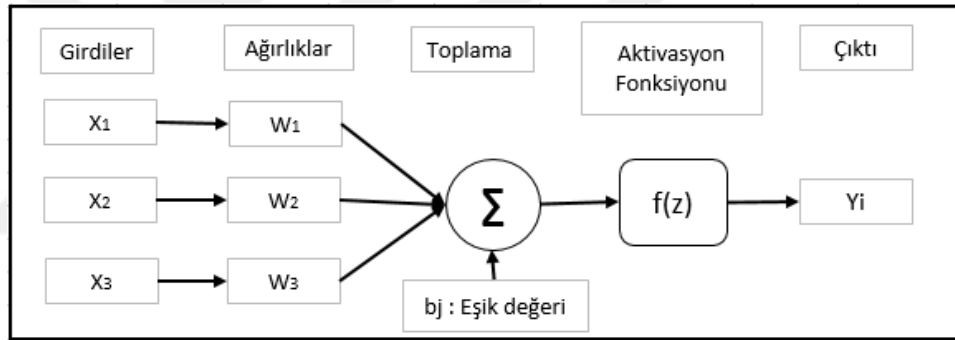
2.5 Yapay Sinir Ağı Modelleri

Yapay sinir ağları katmanlar arası bağlantı yapısına göre ileri beslemeli ve geri beslemeli, katman sayısına göre ise tek katmanlı ve çok katmanlı olmak üzere ikiye ayrılmaktadır.

2.5.1 İleri Beslemeli Yapay Sinir Ağları

İleri beslemeli ağlarda bilgi akışı her bir katmanın çıktısı bir sonraki katmanın girdisi olacak şekilde sırasıyla girdilerin ağırlıklı olarak giriş katmanından, varsa ara (gizli) katmanlar ve son olarak çıktılarının alındığı çıkış katmanına doğru tek yönlü bağlantılarla gerçekleşmektedir. İleri beslemeli ağlarda katmanlar arasındaki bağlantılar tek yönlü olduğu için bir döngü oluşturmazlar.

Şekil 2.5'te görüldüğü gibi ara (gizli) katman W_i ağırlığına bağlı olarak girdi verisini (X_i) alır ve aktivasyon fonksiyonu kullanılarak bir çıktı üretilir. Eğer ağ çok katmanlı ise bu süreç tüm gizli katmanlar için tekrarlanır. Üretilen çıktı değerleri, ağırlıklı olarak çıktı değeri ile karşılaştırılarak bir hata değeri elde edilir [40,41]. Şekil 2.5'te ileri beslemeli bir ağın genel yapısı gösterilmektedir.



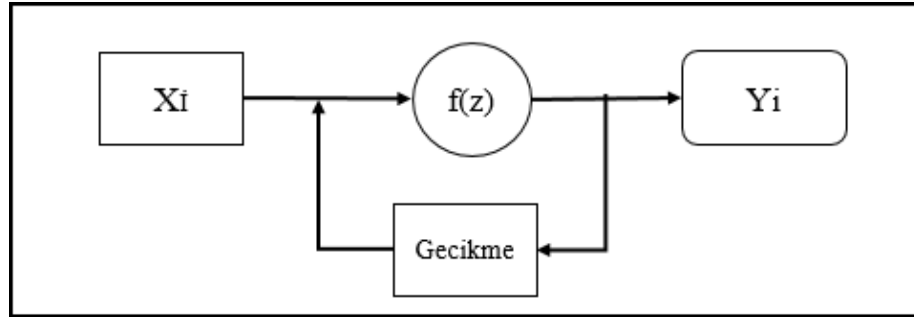
Şekil 2.5 İleri beslemeli yapay sinir ağları [21]

Tek katmanlı ağlar ve çok katmanlı ağlar olmak üzere iki çeşit ileri beslemeli ağ vardır. Tek katmanlı ileri beslemeli ağ yapısında bir giriş katmanı ve bir çıkış katmanı mevcuttur. Çok katmanlı ileri beslemeli ağ yapısında ise birinci gizli katmana iletilen giriş verileri mevcuttur. Ağ yapısında bulunan gizli katmanların sayısının artırılması, özellikle çok sayıda ağ girişi olduğunda, ağın kabiliyetini arttırdığı yönünde etki ettiği gözlemlenmiştir [42]. İleri beslemeli ağlara multilayer perceptron (MLP), radial basis functional neural network (RBFNN) örnek olarak söylenebilir.

2.5.2 Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli ağların ileri beslemeli ağlardan farkı bir hücrenin çıktısının yalnızca bir sonraki katmana girdi olarak verilmemesidir. Bu tür ağlarda hücre çıktısı kendisine, kendinden önceki bir katmana veya kendi katmanında yer alan bir hücreye girdi olarak verilir. Bu özelliğinden dolayı geri beslemeli ağları, doğrusal olmayan

dinamik bir davranış sergilemektedir. Bu ağlarda geri besleme genellikle, gecikme olarak adlandırılan işlemci üzerinden yapılmaktadır.



Şekil 2.6 Geri beslemeli yapay sinir ağları

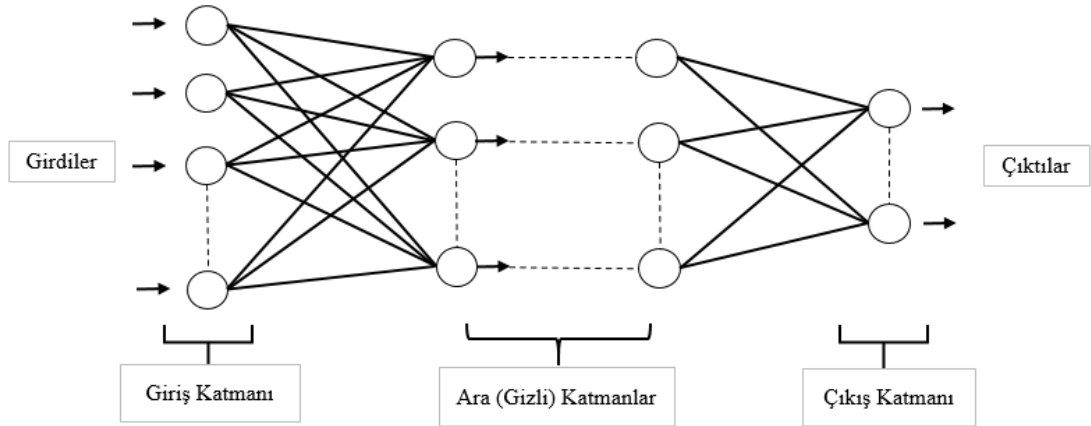
Geri beslemeli ağlarda bağlantılar bir döngü içinde olduğu için çıktının oluşumu yavaştır. Bundan dolayı ağın eğitimi daha uzun sürmektedir [27,41]. Geri beslemeli ağlara Hopfield, Elman ve Jordan ağları örnek olarak söylenebilir.

2.6 Çok Katmanlı Algılayıcılar

2.6.1 Çok Katmanlı Algılayıcıların Mimarisi

Yapay sinir ağlarının tarihine bakıldığında zaman ilk geliştirilen modellerin sadece doğrusal problemleri çözebildiği görülmektedir. Bu probleme çözüm olarak çok katmanlı algılayıcı sinir ağları geliştirilmiştir. Özellikle mühendislik alanında çok kullanılan bu modelin geliştirilmesiyle birlikte yapay sinir ağlarına olan ilgi artmıştır. Çok katmanlı sinir ağı modeli Şekil 2.7’de verilmiştir. Bu modeller, bir giriş, gizli katman(lar) ve çıkış katmanından oluşmaktadır. Bu modelde, bir katmandaki nöronlar kendinden sonra gelen katmandaki nöronlarla ilişkilidir. Aynı zamanda ileri beslemeli ağ olarak ifade edilen bu modelde ağdaki bilgi akışı giriş katmanından çıkış katmanına doğrudur ve geri besleme yoktur. Giriş katmanında ağa tanıtılan veriler üzerine herhangi bir işlem uygulanmaz. Ağ yapısı oluşturulurken gizli katman sayısı hakkında giriş ve çıkış katmanında olduğu gibi herhangi bir bilgi yoktur. Gizli katman sayısı ve bu katmanlardaki nöron sayısı deneme yanılma yöntemi ile tespit edilir. Birkaç gizli nörona sahip ağlar sadece doğrusal tahmin yapabildikleri için karmaşık ilişkileri ayırt edemezler. Gizli katman sayısı ve gizli nöron sayısı ağın performansı için büyük önem taşımaktadır. Gizli nöron sayısının az veya çok fazla olması ağın genelleştirme yeteneğini zayıflatmaktadır. Aynı zamanda gizli katman sayısının az veya çok fazla

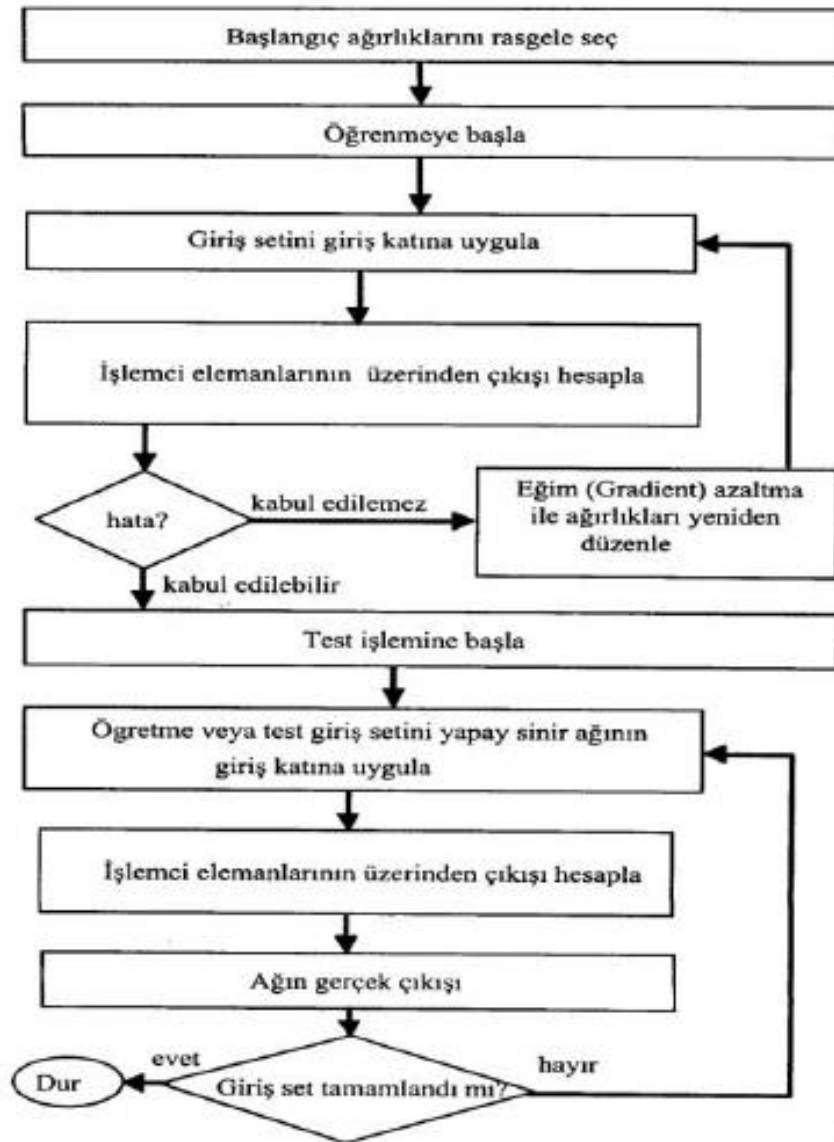
olması ağın performansını olumsuz yönde etkiler. Gizli katman sayısının artırılması ile ağın eğitim süresi artmaktadır. Ayrıca çok katmanlı ağ yapıları denetimli öğrenmeye örnektir. Çünkü bu modelde ağa verilen girdi parametreleri ve çıktı parametreleri gizli katman(lar)da işlendikten sonra çıkış katmanından çıktılar elde edilir [22,43].



Şekil 2.7 Çok katmanlı yapay sinir ağ modeli

2.6.2 Geri Yayılım Algoritması

Çok katmanlı algılayıcıların öğrenme kuralı *Delta Kuralının* genelleştirilmiş halidir. Buradaki amaç gerçek çıktı ile tahmin edilen çıktı arasındaki farkı yani hatayı minimize etmektir. Rumelthart ve arkadaşları tarafından geliştirilen modele geriye yayılım modelide denilmektedir [22]. Geri yayılım algoritması iki aşamada gerçekleşir. Birincisi ileri yayılım aşamasıdır. Bu aşamada, sistemde sinir ağına tanıtılan girdi verileri ve her bir girdiye karşılık gelen çıktı verileri bulunmaktadır. İkincisi ise geri yayılım aşamasıdır. Bu aşamada ağın tahmin ettiği çıktılarda oluşan hata değerleri dikkate alınarak yeni ağırlık değerlerinin hesaplanması işlemi yapılmaktadır. Yapay sinir ağları ile tahmin modellemesinde kullanılan geri yayılım algoritmasına ait akış şeması Şekil 2.8’de verilmiştir [44].



Şekil 2.8 Geri yayılım algoritmasının akış şeması[44]

2.6.3 Eğitim Algoritmaları

Matlab kütüphanesine bakıldığında çok sayıda eğitim algoritmasının olduğu görülmektedir. Bu bölümde çalışmada kullanılan trainlm (Levenberg-Marquardt algoritması), trainrp (The Resilient Backpropagation algoritması), trainbfg (BFGS Quasi-Newton algoritması) ve trainbr (Bayesian Regularization algoritması) algoritmaları hakkında bilgi verilecektir.

Öğrenme algoritmaları ile daha önce geliştirilen algoritmaların iyileştirilmesi hedeflenmiştir. Newton ve gradyen azalması algoritmalarının iyileştirilmiş hali olan

Levenberg-Marquardt algoritması (trainlm), ileri beslemeli geri yayılım algoritmalarından biridir [45]. Bu algoritmanın en bilinen özelliği algoritmalar arasında en hızlı yakınsamaya sahip olmasıdır. Ancak ağdaki ağırlık sayısının artması algoritmayı olumsuz yönde etkiler. Daha önce bu algoritmanın kullanıldığı çalışmalara bakıldığı zaman diğer algoritmalarla göre daha düşük hata değerine sahip olduğu görülmektedir. Diğer algoritmalarla göre daha fazla depolama alanına gereksinim duymaktadır.

The Resilient Backpropagation algoritması (trainrp), fonksiyon yaklaşım problemlerinde iyi sonuç vermemesine rağmen örüntü tanıma problemlerinde en hızlı algoritmadır. Diğer algoritmalarla göre daha küçük depolama alanına gereksinim duymaktadır. Ancak hedeflenen hata değeri azaldığı zaman algoritmanın performansıda doğru orantılı olarak azalır.

BFGS Quasi-Newton algoritması (trainbfg), Levenberg-Marquardt (trainlm) algoritmasına benzer performans göstermektedir. Ancak onun kadar depolama alanına ihtiyaç duymaz. Bu algortmada hesaplamalar ağı boyutuya göre geometrik olarak artar.

Bayesian Regularization algoritması (trainbr), ağırlık değerlerini Levenberg-Marquardt algoritmasına göre günceller. Bu algoritma, hataları minimize eder ve en iyi sonucu veren ağı tespit etmek için kombinasyonlar oluşturur [46].

2.6.4 Çok Katmanlı Algılayıcıların Çalışma Prosedürü

Çok katmanlı ağların çalışması aşağıdaki adımlardan oluşmaktadır:

Örneklerin toplanması: Problemin çözümü için eğitim verisi ve test verisi olmak üzere iki veri seti halinde gruplandırılan verilerin toplanması gerekmektedir.

Ağ mimarisinin belirlenmesi: Problem için kullanılacak veriler ağ yapısının belirlenmesinde önemlidir. Bu aşamada giriş, varsa gizli ve çıkış katmanlarındaki nöron sayısı ile gizli katman sayısı belirlenmektedir.

Öğrenme parametrelerinin belirlenmesi: Ağı en iyi çıktıyı üretebilmesi için önemli olan öğrenme katsayısı, momentum katsayısı, aktivasyon fonksiyonu gibi parametreler belirlenmektedir.

Ağırlıkların başlangıç değerlerinin atanması: Nöronlar arasındaki ilişkiyi oluşturan ağırlık değerleri başlangıçta rastgele atanır ancak ağ eğitilirken değerler ağa uygun olacak şekilde güncellenir.

Eğitim veri setinin ağa tanıtılması: Ağı eğitimi için kullanılacak girdi ve çıktı verileri ağa tanıtılır.

Öğrenme sırasında ileri hesaplama yapılması: Ağ tanıtılan girdi verilerinden çıktı verilerinin üretilmesidir..

Gerçek çıktı ile üretilen çıktının karşılaştırılması: Ağı doğru bir şekilde eğitilip eğitilmediğini anlamak için üretilen çıktı ile gerçek çıktı arasındaki fark yani hata değeri hesaplanır.

Ağırlıkların değiştirilmesi: Hata değerini minimize etmek için ağırlık değerlerinin güncellenmesi gerekmektedir [22].

2.7 Yapay Sinir Ağı Eğitimi ve Öğrenmesi

Yapay sinir ağlarında öğrenme Pavlov'un köpekler üzerinde yaptığı deney ile en iyi şekilde açıklanabilir. Öğrenme, köpeklerin yiyecekleri gördükleri anda salya akıtmaları ile başlamaktadır. İkinci aşamaya geçildiği zaman köpeklerin bulunduğu ortama zil yerleştirilmiştir. Ancak yeterli öğrenme gerçekleşmediği için köpekler tepki göstermemiştir. Öğrenme tamamen gerçekleştikten sonra ortamda yiyecek bulunmasa bile zil çaldığında köpekler salya akıtmaya başlamıştır. Buradan da anlaşılacağı gibi salya ile zilin herhangi bir bağlantısı olmamasına rağmen öğrenme gerçekleştikten sonra köpeklerin davranışları değişmektedir. Yapay sinir ağlarında eğitimin amacı, ağa tanıtılan veriler için en iyi çıktıyı üretecek ağırlıkların belirlenmesidir. Ağı bu ağırlıkları belirlemesi yani olaylar hakkında genelleme yapabilecek seviyeye gelmesi öğrenmenin gerçekleşmiş olduğu anlamına gelmektedir [47].

Yapay sinir ağı, ağırlıkların tekrarlanarak ayarlanmasıyla gerçek çıktı ile tahmin edilen çıktı arasındaki farkı en aza indirmek için eğitilir. Ağ bu farkı kabul edilebilir düzeye indirebilmek için ağırlık değerlerini değiştirerek bu işlemi tekrarlar. Öğrenme, ağırlıkların tekrarlanmasıyla istenilen sonuca ulaşılan kadar geçen süreçtir. Tahmin edilen çıktı gerçek çıktıya yaklaşıyorsa sinir ağı öğrenmiş demektir. Sinir ağının bir kez işlenmesine iterasyon (epoch) denir. İterasyon sayısının az veya çok olmasının

öğrenmede önemli bir etkisi vardır. İterasyon sayısının çok olması ağıın performansını düşürmektedir. Eğer ağ çok eğitilirse genelleme yeteneğini kaybedecek ve ezberleyecektir [37,41].

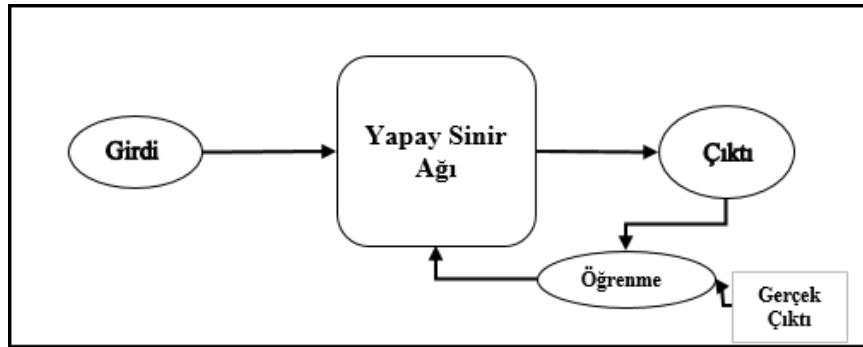
2.7.1 Yapay Sinir Ağlarında Öğrenme Yöntemleri

Yapay sinir ağlarında ağa girdi olarak tanıtılan verilerden çıktının elde edilmesi ağırlıkların güncellenerek ağıın öğrenmesiyle gerçekleşir. Bu öğrenme çeşitli yollarla gerçekleşir. Yapay sinir ağları öğrenme yöntemlerine göre 3'e ayrılmaktadır:

- Denetimli Öğrenme
- Denetimsiz Öğrenme
- Takviyeli Öğrenme

2.7.1.1 Denetimli Öğrenme

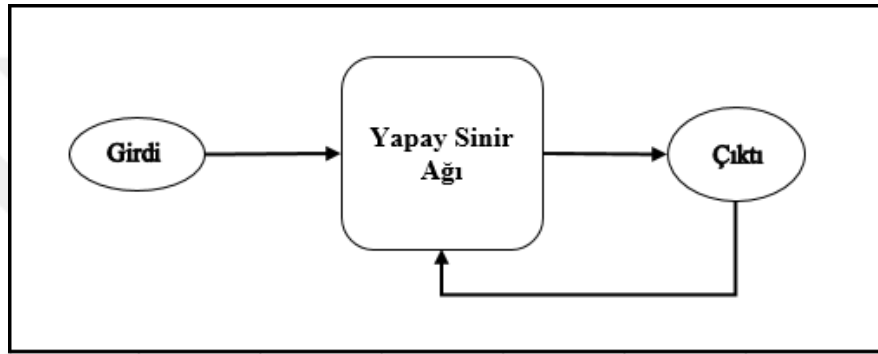
Bu öğrenme türünde, öğrenme danışman yardımıyla gerçekleşmektedir. Sistemde hem girdi parametreleri hem de bunlara karşılık gelen çıktı parametreleri yani veri çiftleri bulunmaktadır. Bu yüzden danışmanın, girdi ve çıktı olarak ifade edilen çevre hakkında bilgisi vardır diyebiliriz. Ancak başlangıçta ağıın bu çevre hakkında bilgisi yoktur. Danışmanın kullanacağı bilgi eğitim yoluyla ağa verilir ve bu bilgi sinaptik ağırlıklar şeklinde depolanır. En çok tercih edilen öğrenme algoritması geri yayılım algoritmasıdır. Tahmin edilen çıktı parametreleri ile ağa tanıtılan çıktı parametreleri arasındaki fark hata olarak tanımlanmaktadır. Bu algorithmada tahmin sonucunda elde edilen hata değeri ağırlıkları güncellemek ve hatayı minimize etmek için ağda geriye doğru iletilir [37,48]. Şekil 2.9'da denetimli öğrenme yöntemi gösterilmiştir.



Şekil 2.9 Denetimli öğrenme yöntemi [48]

2.7.1.2 Denetimsiz Öğrenme

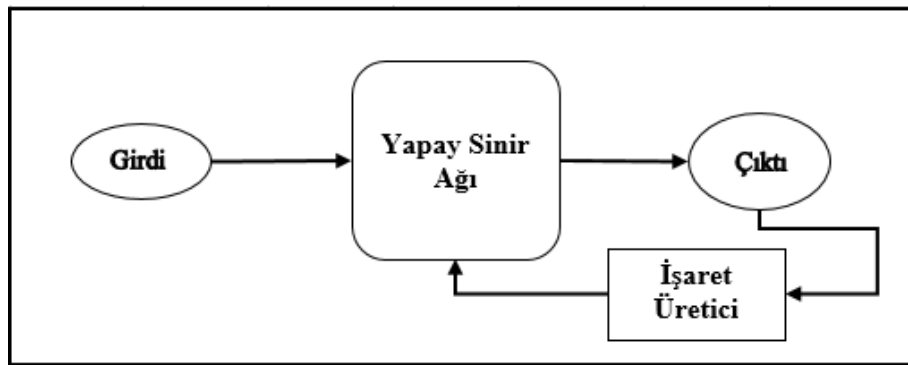
Bu öğrenme türünde, öğrenmeye yardımcı danışman bulunmamaktadır. Ağa sadece girdi parametreleri tanıtılır. Ağa tanıtılan parametreler arasındaki ilişkiyi ağın kendi kendine öğrenmesi amaçlanmıştır. Öğrenme tamamlandıktan sonra elde edilen çıktıların ne ifade ettiği kullanıcı tarafından belirlenmektedir. Öğrenmeden önce ağa tanıtılan çıktı parametreleri olmadığı için hata değeri hesaplanamaz. Bu öğrenme türü yapay sinir ağlarında çok nadir tercih edilmektedir [47]. Şekil 2.10'da denetimsiz öğrenme yöntemi gösterilmiştir.



Şekil 2.10 Denetimsiz öğrenme yöntemi [49]

2.7.1.3 Takviyeli Öğrenme

Denetimli öğrenmede olduğu gibi öğrenme danışman yardımıyla gerçekleşmektedir. Ağın eğitimi sonucunda elde edilen çıktıların doğru veya yanlış olduğu bilgisi (takviye işareti) işaret üretici tarafından oluşturulur. Her iterasyon sonunda ağa gelen bu bilgi ile öğrenme devam eder [47]. Şekil 2.11'de takviyeli öğrenme yöntemi gösterilmiştir.



Şekil 2.11 Takviyeli öğrenme yöntemi [49]

Tablo 2.2 Öğrenme algoritmaları ve kullanım alanları [49]

Öğrenme Yöntemi	Öğrenme Kuralı	Ağ Yapısı	Öğrenme Algoritması	Kullanıldığı Alan
Denetimli	Hata Düzeltme	Bir veya daha fazla katman	Geriye Yayımlı Algılayıcılar	Şekil Tanıma Tahmin Kontrol
	Boltzman	Geri Dönüşümlü	Boltzman Öğrenmesi	Şekil Sınıflandırma
	Hebbian	Çok Katmanlı	Doğrusal Diskriminant	Veri Analizi Şekil Sınıflandırma
	Rekabetli	Rekabetli	Vektör Kuantalama Öğrenmesi	Veri Sıkıştırma
		ART Ağı	Art Haritalaması	Sınıflandırma
Denetimsiz	Hata Düzeltme	Çok Katmanlı	Projeksiyon	Veri Analizi
	Hebbian	İleri Beslemeli	Temel Analiz	Veri Analizi Veri sıkıştırma
		Hopfield Ağı	İlişkili Öğrenme	İlişki
	Rekabetli	Rekabetli	Vektör Kuantalama Öğrenmesi	Veri Sıkıştırma Sınıflandırma
		Kohonen	Kohonen	Sınıflandırma Veri Analizi
		ART	ART 1, ART 2	Sınıflandırma

2.8 Yapay Sinir Ağlarında Öğrenme Kuralları

Yapay sinir ağlarında birden çok öğrenme kuralı mevcuttur. Literatürde en çok tercih edilen öğrenme kuralları aşağıda verilmiştir.

- Hebb Kuralı
- Hopfield Kuralı
- Kohonen Kuralı
- Delta Kuralı

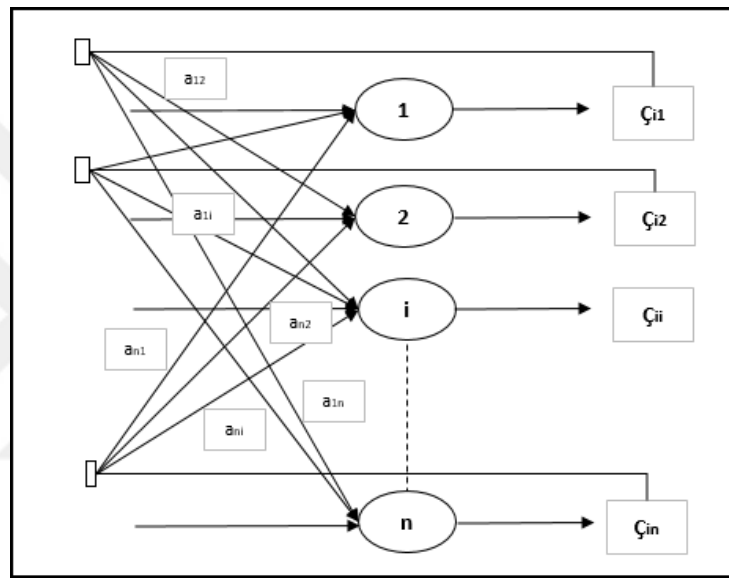
2.8.1 Hebb Kuralı

Hebbian öğrenme, en eski öğrenme kuralıdır. Bu öğrenme kuralının açıklanması biyolojik sistemlere dayanmaktadır. Hebb'in yazmış olduğu 'The Organization of Behaviour' adlı kitapta öğrenme kuralı şu şekilde açıklanmıştır: 'Sinapsın giriş ve çıkış

nöronları aynı anda aktif olursa iki nöron arasındaki bağlantı güçlenir. Eğer giriş ve çıkış nöronları farklı zamanlarda aktif olursa iki nöron arasındaki bağlantı zayıflar [48].’

2.8.2 Hopfield Kuralı

Hopfield ağ yapısı, John Hopfield tarafından geliştirilmiş denetimsiz öğrenme modelidir. Diğer sinir ağlarına göre oldukça basit bir yapısı vardır. Birbirine bağlı nöronlardan oluşan tek katmanlı ağ yapısına sahiptir. Her bir nöron hem giriş hem çıkış katmanıdır. Hopfield ağ mimarisi Şekil 2.12’de gösterilmiştir.



Şekil 2.12 Hopfield ağ mimarisi [51]

Şekilde görüldüğü gibi ağda yer alan i ve j hücreleri birbirine a_{ij} ağırlığı ile bağlıdır. a_{ij} i nöronundan çıkan bilginin j nöronuna olan etkisini ifade eder. Hopfield ağı, ağın eğitildiği depolama aşaması ve test edildiği geri çağırma aşaması olmak üzere iki aşamadan oluşur [51].

2.8.3 Kohonen Kuralı

Kohonen tarafından 1982 yılında geliştirilen bu ağ yapısında amaç girdi parametreleri arasındaki ilişkinin öğrenilmesidir. Sisteme çıktı parametreleri tanıtılmadığı için denetimsiz öğrenmeye örnek olarak verilebilir. Burada ‘Kazanan hepsini alır’ (winner takes all) politikası uygulanmaktadır. Yani girdi verilerine en yakın çıktıyı üreten nöron *kazanan nöron* olarak nitelendirilir ve bu nöron ağırlık değerlerini günceller. Kazanan nöron, komşu nöronlardan daha güçlü olur ve komşu nöronlarında ağırlık değerlerini

güncellemesine olanak sağlar. Nöronlar ağırlık değerlerini güncelledikleri zaman en iyi sonucu verme şansını yakalayabilir bu nedenden dolayı kazanan nörona komşu olan nöronlarında en iyi sonucu verme şansı artmış olur [41,52].

2.8.4 Delta Kuralı

Widrow ve Hoff tarafından geliştirilen delta kuralı, hebb kuralının geliştirilmiş halidir. Bu kural, sinir ağına tanıtılan çıktı verileri ile ağıın ürettiği çıktı verileri arasındaki farkı (delta) minimize etmek için ağıın nöronlara ait ağırlık değerlerini sürekli güncellemesi prensibine dayanmaktadır. Yani delta kuralı, gerçek çıktı ile ağıın ürettiği çıktı arasındaki hatanın kareleri ortalamasının minimize edilmesidir [22].

2.9 Yapay Sinir Ağlarının Tasarımı

Yapay sinir ağları kullanılarak yapılan çalışmalarda başarılı olmanın en önemli adımı doğru metodolojinin belirlenmesidir. Probleme uygun sinir ağı geliştirilirken karar verilmesi gereken bazı durumlar vardır:

- ✓ Ağ yapısına karar verilmesi (katman sayısı, katmanlardaki nöron sayısı vs.)
- ✓ Nöronlardaki fonksiyonların karakteristik özelliklerinin belirlenmesi
- ✓ Öğrenme algoritmasının seçimi
- ✓ Veri setinin eğitim ve test verisi olarak gruplandırılması

Sinir ağı yukarıda bahsedilen durumlar dikkate alınarak uygun parametrelerle oluşturulduğu zaman ağı tutarlı sonuçlar üretecektir.

2.9.1 Yapay sinir ağı ağı yapısının seçimi

Sinir ağı oluşturulurken probleme uygun ağı yapısının seçilmesi önemlidir. Probleme göre ağı yapıları Tablo 2.3'te verilmiştir. Uygun ağı yapısının seçiminde öğrenme algoritması da büyük önem taşır. Çünkü her öğrenme algoritmasının kullanılması gereken ağı yapısı vardır. Bundan dolayı öğrenme algoritması seçildiği zaman otomatik olarak ağı yapısı da seçilmiş olur. Örneğin, ağıda geri yayılım algoritması kullanılacaksa ağı yapısının ileri beslemeli ağı olması gerekir.

Tablo 2.3 Ağ yapıları ve kullanıldıkları alanlar [22]

Kullanım Amacı	Ağ Yapısı	Ağın Kullanımı
Tahmin	Çok Katmanlı Ağ	Ağın girdilerinden bir çıktı değerinin tahmin edilmesi
Sınıflandırma	LVQ	Girdilerin hangi sınıfa ait olduklarının belirlenmesi
	ART	
	Counterpropagation	
	Olasılıklı Sinir Ağları	
Veri İlişkilendirme	Hopfield	Girdilerdeki hatalı bilgilerin bulunması ve eksikliklerin tamamlanması
	Boltzman Machine	
	Bidirectional As.Memory	

2.9.2 Öğrenme Algoritmasının Seçimi

Sinir ağının yapısına karar verdikten sonra ağın başarısını etkileyen bir diğer etken öğrenme algoritmasıdır. Genellikle öğrenme algoritmasını seçerken ağ yapısı dikkate alınır. Literatüre bakıldığı zaman yapay sinir ağlarında kullanılacak çok fazla öğrenme algoritması olduğu görülmektedir [53].

2.9.3 Gizli Katman Sayısının Belirlenmesi

Gizli katman(lar), sinir ağlarının veriler arasındaki ilişkiyi öğrenerek genelleme yapma yeteneğine olanak sağlar. Ağ yapısında bir veya iki gizli katman en iyi sonuca ulaşmak için yeterli görülmektedir. Gizli katman sayısının çok olması ağın ezberleme ihtimalini artırır. Literatüre bakıldığı zaman tek gizli katmanlı ağ yapılarının daha çok tercih edildiği görülmektedir. Ancak bazı durumlarda tek gizli katman çok sayıda gizli nörona gerek duyabilir. Bundan dolayı ağın genelleme yeteneği zayıflar ve ezberlemeye başlar. Böyle durumlarda birden fazla gizli katman (çoğunlukla iki gizli katman) tercih edilmektedir. Yapılan bazı çalışmalarda ağın performansı için ikiden fazla gizli katmana ihtiyaç duyulmadığı görülmüştür [54].

2.9.4 Nöron Sayısının Belirlenmesi

Gizli katmandaki nöron sayısı ağın performansı için önemli bir etkidir. Gizli nöron sayısını belirlemek için en çok tercih edilen yöntem deneme yanılma yoludur. Genelleme yeteneğinin daha iyi olması nedeniyle genellikle nöron sayısının az olması tercih edilir. Ancak nöron sayısının çok az olması genelleme yeteneğini zayıflatarak

yanlış sonuca ulaştırabilir. Nöron sayısı çok fazla olduğu zamanda ağıın ezberlemiş olma ihtimali yüksektir. Nöron sayısının belirlenmesinde sonuçlarının kesinliği bilinmeyen çeşitli kurallar mevcuttur. Giriş katmanında ve çıkış katmanında sırasıyla n ve m adet nöron bulunduran tek gizli katmanlı ağ yapısında, gizli nöron sayısı $(n*m)^{1/2}$ olabilir. Baily ve Thompson, tek gizli katmanlı ağ yapısı için gizli nöron sayısını giriş katmanındaki nöron sayısının %75'i olarak hesaplamaktadır. Katz, gizli nöron sayısının girdi katmanındaki nöron sayısının bir buçuk ile üç katı arasında olması gerektiği belirtmektedir [55].

2.9.5 Normalizasyon

Yapay sinir ağlarında, ağıın eğitiminin daha verimli olmasını sağlamak için parametrelere normalizasyon adı verilen bir ön işlem uygulanır. Normalize edilmemiş ham veriler kullanılarak yapılan eğitim oldukça yavaş olabilir. Normalizasyon işlemi, farklı veri setlerindeki dengesizliğin ortadan kaldırılmasını veya aynı veri setindeki veriler arasında aşırı büyük ve aşırı küçük değerlerin olmasından kaynaklanacak olumsuzlukların engellenmesini sağlar. Bu nedenden dolayı, normalizasyon verilerin düzenli bir şekilde dağılmasını sağlar diyebiliriz. Literatürde normalizasyon işlemi için farklı yöntemler kullanılmıştır. Bu yöntemlerden bazıları aşağıda verilmiştir [21].

2.9.5.1 İstatistiksel veya Z-score Normalizasyonu

Aynı veri setindeki değerlerden bazılarının negatif bazılarının pozitif değere sahip olması sonucunda aralarında oluşan uzaklıklar, elde edilecek sonuçlar üzerinde önemli bir etkindir. Verilerin normalleştirilmesi ile parametrelerin her birinin ağıın performansına katkılarının aynı ölçüde olması sağlanır. Bu normalizasyon türü ile değerler arasındaki uzaklık azaltılmış olur [21].

$$X_i = (X_0 - \mu_i) / S_i \quad (2.10)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

μ_i = i. verinin bulunduğu sütunun ortalaması

S_i = i. verinin bulunduğu sütunun standart sapması

2.9.5.2 Min-Max Normalizasyonu

Bu normalizasyon türünde veriler minimum 0 değerini, maksimum 1 değerini alarak doğrusal olarak normalize edilir. Maksimum ve minimum değer alan veriler dışındakiler $[0,1]$ aralığında doğrusal olarak değerler alırlar [21].

$$X_i = (X_0 - X_{\min}) / (X_{\max} - X_{\min}) \quad (2.11)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

$X_{\min}$ = Sütundaki minimum değer

$X_{\max}$ = Sütundaki maksimum değer

2.9.5.3 Medyan Normalizasyonu

Veriler arasındaki aşırı sapmalardan etkilenmeyen bu normalizasyon türünde veriler bulundukları sütunun medyanı alınarak normalize edilir. Aşağıdaki denklemde yer alan Median (α_i) ifadesi iki farklı şekilde hesaplanabilir. Birincisi, verinin bulunduğu sütundaki girdi sayısı tek ise o sütundaki değerler küçükten büyüğe doğru sıralanır ve ortadaki değer medyanı verir. İkincisi ise verinin bulunduğu sütundaki girdi sayısı çift ise o sütunun orta noktasındaki iki değer ortalama medyanı verir [21].

$$X_i = X_0 / \text{Median}(\alpha_i) \quad (2.12)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

α_i = i. verinin bulunduğu sütunun medyanı

2.9.5.4 Sigmoid Normalizasyonu

Bu normalizasyon türünde veriler $[0,1]$ veya $[-1,1]$ arasında değerler alacak şekilde normalize edilir. Tanjant sigmoid fonksiyonu (tansig) ağın süreçlerinin daha hızlı gerçekleşmesini sağladığı için daha çok tercih edilebilir [21].

$$X_i = (e^{X_0} - e^{-X_0}) / (e^{X_0} + e^{-X_0}) \quad (2.13)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

e = Doğal logaritma

2.9.5.5 D-Min-Max Normalizasyonu

Bu normalizasyon türünde veriler [0.1,0.9] arasında değerler alacak şekilde normalize edilir [21].

$$X_i = 0.8 * ((X_0 - X_{\min}) / (X_{\max} - X_{\min})) + 0.1 \quad (2.14)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

$X_{\min}$ = Sütundaki minimum değer

$X_{\max}$ = Sütundaki maksimum değer

Verilerin hangi aralıkta normalleştirileceği, çıkış katmanındaki nöronların sahip oldukları aktivasyon fonksiyonuna göre değişmektedir. Çıkış katmanında sigmoid fonksiyonu kullanıldığı zaman veriler [0,1] arasında değerler alacak şekilde, hiperbolik tanjant fonksiyonu kullanıldığı zaman ise [-1,1] arasında değerler alacak şekilde normalize edilir [51].

2.9.6 Performans Fonksiyonu Seçimi

Performans fonksiyonunun seçimi ağıın öğrenmesini etkileyen önemli bir etkidir. Bir yapay sinir ağı performansının ölçüsü tahminin ne kadar doğru olduğudur. Bu ölçü, gerçek veriler ile tahmin verileri arasındaki farktır. Literatürde birçok performans fonksiyonu vardır. En çok kullanılan fonksiyonlar aşağıda verilmiştir [54]. (N = Çıktı Sayısı)

- **Mean Absolute Deviation (Ortalama Mutlak Sapma)**

$$MAD = \sum | \text{gerçek değer} - \text{tahmin değer} | / N$$

- **Sum of Squared Error (Toplam Karesel Hata)**

$$SSE = \sum | \text{gerçek değer} - \text{tahmin değer} |^2$$

- **Mean Squared Error (Ortalama Kare Hatası)**

$$MSE = \sum | \text{gerçek değer} - \text{tahmin değer} |^2 / N$$

- **Root Mean Squared Error (Karesel Ortalama Hata Karekökü)**

$$\text{RMSE} = (\text{MSE})^{1/2}$$

- **Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hatası)**

$$\text{MAPE} = \sum |(\text{gerçek değer} - \text{tahmin değer}) / \text{gerçek değer}| / N$$



3. BÖLÜM

UYGULAMA VE BULGULAR

3.1 Uygulama Yapılan Firma Hakkında Bilgi

Tez kapsamında çalışmalar yapılacak firma, kimya sektöründe ısı yalıtım grubu ürünleri üreten bir kuruluştur. Firmada farklı markalar adı altında ürünler üretilmektedir. Özellikle ısı yalıtım levhası ve kuru harç (sıva, yapıştırıcı, mineral kaplama) ürün gruplarında üretim kapasitesi olarak Avrupa'nın en büyük tesislerinden biridir. Üretim prosesi içinde dar boğaz olarak ön şişirme bölümü seçilmiş olup burada ürün ve ürün kalitesine etki eden faktörler ele alınmak istenmiştir.

Isı yalıtım levhası üretiminin hammadde olan ve içeriğinde belirli oranda pentan gazı bulunan boncukların ilk prosesi ön şişirme aşamasıdır. Bu aşamadaki amaç, boncukların blok haline gelmeden önce ısı yalıtım levhasında istenilen yoğunlukta şişirilmesidir. Örnek olarak 14.5 kg/m^3 yoğunlukta istenilen levhalar için genel kabul ile ön şişirmede istenilecek yoğunluk 13.5 kg/m^3 'dür. Fakat hammadde kalitesi, hava koşulları gibi etmenler bu yoğunluğun değişmesine neden olabilir. Buradaki yoğunluk bloktaki yoğunluğa direkt etki ettiği için ön şişirme yoğunluğu kritik derecede önemlidir.

Ön şişirme makinesi verilerinde tarih, kullanılan hammadde, çevrim sayısı, her çevrimdeki ağırlık, elde edilen yoğunluk ve kapasite açısından çevrim süresi önemlidir. Her döngüde ölçülen yoğunlukların ortalaması bloğu temsil edecektir. 3 tonluk silolarda stoklanan boncuklar en az 8 saat dinlendirildikten sonra blok yapma aşamasında kullanılır. Yoğunluğun yüksek olması blokların ağır olmasına dolayısı ile fazla hammadde kullanımına, düşük olması ise istenen kalitede blok üretilmemesine neden olmaktadır. Nihayetinde ön şişirmede çevrim ağırlıkları, elde edilen yoğunluklar ve kapasite için çevrim süreleri devamlı olarak kontrol altında tutulmalıdır.

3.2 Veri Madenciliği PolyAnalyst Uygulaması

Bu bölümde, veri madenciliği modellerinden tahmin edici modeller ile ilgili bir uygulama yapılmıştır. Elde edilen veri seti ürün ve ürün kalitesine etki eden 6 parametreden oluşmaktadır. Ağırlık, çevrim süresi, buhar süresi, buhar sıcaklığı ve ölçülen yoğunluk girdi parametreleri iken, istenilen yoğunluk çıktı parametresidir. Buradaki amaç, girdi parametrelerini kullanarak çıktı parametresinin tahminine yönelik bir formül elde etmektir. Verilerin analizi için PolyAnalyst programı kullanılmıştır. Programdaki algoritmalar, bağımlılıkları bulma (Find Dependencies) algoritması ve kümeleme analizi (Cluster Analysis) ön işleme aracı olarak kullanılmıştır. Sınıflandırma algoritması olarak kural bulma (Find Laws) algoritması kullanılmıştır. Veri madenciliği sürecine geçmeden önce yapılacak olan analizlerin daha etkili olması için parametreler ön işleme ile uygun hale getirilmiştir. Bu amaçla veri madenciliği yazılımı olan PolyAnalyst programına ait bağımlılıkları bulma algoritması ve kümeleme analizi ön işleme aracı olarak kullanılmıştır. Bağımlılıkları bulma analizi mevcut 6 parametre ile yapılmıştır. Hedef değişken istenilen yoğunluktur. Bağımlılıkları bulma analizi ile istenilen yoğunluğa etki eden parametreler ve bu parametrelerin etkili olup olmadığı tespit edilmiştir. Kümeleme analizi sonucunda ise üç küme elde edilmiştir. 27629 adet veri içeren küme-1 en uygun küme olduğu için analizler kural bulma algoritması ile bu veri kümesi üzerinden gerçekleştirilmiştir. Daha sonra eğitim verileri, bağımlılıkları bulma analizi sonucu elde edilen veriler ve kümeleme analizi sonucu elde edilen en uygun veriler üzerinde kural bulma algoritması ile analizler yapılmıştır. Bağımlılıkları bulma analizi verileri ile yapılan analiz sonucu en yüksek R^2 değerine (0,8116) sahip formül kullanılarak istenilen yoğunluğa ulaşılabilir. Daha ayrıntılı sonuçlar için Buldanlı ve Gürbüz'ün çalışmasına bakınız [56].

3.3 Regresyon Analizi SPSS Uygulaması

Bu bölümde, ağırlık, çevrim süresi, ölçülen yoğunluk, buhar sıcaklığı ve buhar süresi bağımsız değişken, istenilen yoğunluk bağımlı değişken olarak kabul edilerek SPSS programı yardımıyla çoklu regresyon analizi gerçekleştirilmiştir. Analiz sonucu elde edilen değerler aşağıdaki tabloda verilmiştir.

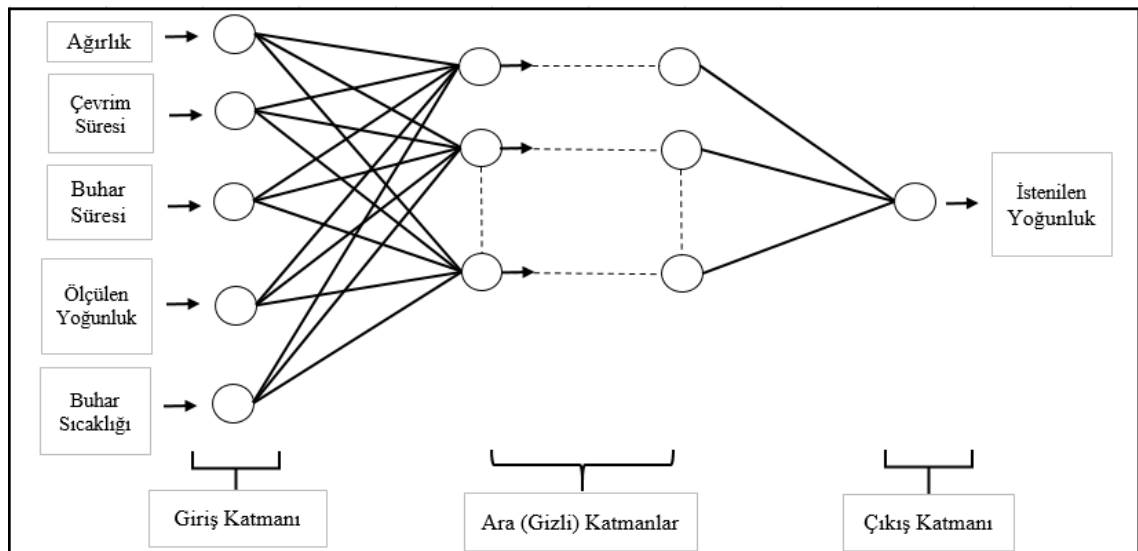
Tablo 3.1 Regresyon analizi sonuçları

Model	R	R ²	Düzeltilmiş R ²	Standart Hata
1	0,75	0,563	0,561	0,09859

Tablo 3.1'e bakıldığı zaman bağımsız değişkenler ile bağımlı değişken arasındaki korelasyonun 0,75; determinasyon katsayısının ise $R^2 = 0,563$ olduğu görülmektedir. Korelasyon katsayısı ($R=0,75$) bağımsız değişkenler ile bağımlı değişken arasında yüksek düzeyde pozitif yönde bir ilişki olduğunu göstermektedir.

3.4 Yapay Sinir Ağı MATLAB Uygulaması

Bu çalışmada ileri beslemeli çok katmanlı ağ yapısı kullanılmıştır. Çünkü yapay sinir ağlarında tahmin amacıyla kullanılan en yaygın ağ modeli çok katmanlı (MLP) algılayıcılardır. Oluşturulan ağ yapısında ağırlık, ölçülen yoğunluk, çevrim süresi, buhar süresi ve buhar sıcaklığı girdi verisi olarak kabul edilmiştir ve bu verilerden istenilen yoğunluk tahmin edilmiştir. Bu çalışmada 3000 adet verinin %70'i ağı eğitimi, %30'u ağı test edilmesi için kullanılmıştır. Oluşturulan farklı sinir ağı yapıları MATLAB R2015 programı kullanılarak analiz edilmiştir. Çalışmada kullanılan MATLAB kodları ekte verilmiştir.



Şekil 3.1 Tasarlanan ileri beslemeli çok katmanlı ağ yapısı

Bu bölümde, yapay sinir ağının nasıl oluşturulduğu ve nasıl eğitildiği kodlar verilerek anlatılmıştır.

Öncelikle ‘*veri*’ isimli excel dosyasındaki veriler aşağıdaki komut ile MATLAB’a tanıtılmıştır.

```
>>veri = xlsread('veri.xlsx');
```

Tablo 3.2 Programa tanıtılan örnek veriler

Malzeme	Ağırlık	Ölçülen Yoğunluk	Çevrim Süresi	Buhar Süresi	Buhar Sıcaklığı	İstenilen Yoğunluk
N5300-15-R	70,37	15,1	133	90	102,6	14,5
N5300-15-R	69,92	15	130	88	103,4	14,5
N5300-15-R	69,92	14,8	132	89	101,3	14,5
N5300-15-R	69,85	14,56	128	86	102,5	14,5
N5300-15-R	69,66	14,7	128	87	101,8	14,5
N5300-15-R	69,59	14,66	128	86	98,5	14,5
N5300-15-R	69,4	14,7	132	90	102,4	14,5
N5300-15-R	69,27	14,5	127	84	101,7	14,5
N5300-15-R	69,59	14,5	132	89	102,5	14,5
N5300-15-R	69,59	14,43	129	87	101,9	14,5
N5300-15-R	69,66	14,4	131	88	101,4	14,5
N5300-15-R	69,85	14,4	127	85	103,2	14,5

>>**girdi = veri(:,1:5);** Kodu ‘*veri*’ dosyası içindeki tüm satırlar dikkate alınarak 1. ve 5. sütunlar arasındaki verilerin tamamının girdi verisi olarak alındığını göstermektedir.

>>**çikti = veri(:,end);** Kodu ‘*veri*’ dosyası içindeki tüm satırlar dikkate alınarak en son sütundaki verilerin tamamının çıktı verisi olarak alındığını göstermektedir.

Yapay sinir ağı kullanılarak en iyi sonucu veren ağı bulunması amaçlanmıştır. Çalışmamızda uç değerlerden kurtulmak ve verilerin daha iyi modellenenebilmesi için veriler üzerinde [0,1] aralığında olacak şekilde normalleştirme işlemi yapılmıştır. Kullanılan normalizasyon formülü 3.1’de verilmiştir.

$$X_i = (X_0 - X_{\min}) / (X_{\max} - X_{\min}) \quad (3.1)$$

X_i = Normalize edilmiş i. veri

X_0 = Gerçek veri

$X_{\min}$ = Sütundaki minimum değer

$X_{\max}$ = Sütundaki maksimum değer

Ağa tanıtılan veriler [0,1] aralığında normalize edildiği için ağı tahmin ettiği çıktıda [0,1] aralığında yer alacaktır.

Tahmin edilen bu değerlerin gerçek değerlerini tespit etmek için gerekli kod programa yazılmıştır.

Tablo 3.3 Normalize edilmiş örnek veriler

Ağırlık	Ölçülen Yoğunluk	Çevrim Süresi	Buhar Süresi	Buhar Sıcaklığı	İstenilen Yoğunluk
0,50	0,17	0,26	0,27	0,78	0,67
0,50	0,16	0,25	0,25	0,80	0,67
0,50	0,15	0,26	0,26	0,76	0,67
0,49	0,15	0,23	0,24	0,78	0,67
0,49	0,15	0,23	0,25	0,77	0,67
0,49	0,15	0,23	0,24	0,70	0,67
0,49	0,15	0,26	0,27	0,78	0,67
0,48	0,14	0,23	0,23	0,76	0,67
0,49	0,14	0,26	0,26	0,78	0,67
0,49	0,14	0,24	0,25	0,77	0,67
0,49	0,14	0,25	0,25	0,76	0,67
0,49	0,14	0,23	0,24	0,79	0,67

En iyi sonucu veren ağ yapısını bulmak için ileri beslemeli (feed forward) tek gizli katmanlı ve iki gizli katmanlı ağ yapıları oluşturulmuştur. İlk etapta tek gizli katmanlı ağ yapısı oluşturulmuştur. Gizli katmandaki nöron sayısı 2-10 ile 15 olarak belirlenmiştir. Eğitim fonksiyonu olarak ‘trainlm’ seçilmiştir ve gizli nöron sayısının tespiti için çeşitli denemeler yapılmıştır.

Tablo 3.4 Gizli nöron sayısının tespiti için yapılan denemeler

No	Network	Layer-1	Layer-2	Learning Func.	MSE
1	5,2,1	logsig	tansig	Trainlm	0,0034019
2	5,3,1	logsig	tansig	Trainlm	0,0044175
3	5,4,1	logsig	tansig	Trainlm	0,0024979
4	5,5,1	logsig	tansig	Trainlm	0,0032850
5	5,6,1	logsig	tansig	Trainlm	0,0025433
6	5,7,1	logsig	tansig	Trainlm	0,0022985
7	5,8,1	logsig	tansig	Trainlm	0,0023056
8	5,9,1	logsig	tansig	Trainlm	0,0036857
9	5,10,1	logsig	tansig	Trainlm	0,0016883
10	5,15,1	logsig	tansig	Trainlm	0,0018601

Tablo 3.4'te görüldüğü gibi gizli katmanda 'logsig', çıkış katmanında 'tansig' aktivasyon fonksiyonu ve 'trainlm' eğitim fonksiyonu dikkate alınarak oluşturulan yapay sinir ağı için gerekli kodlar aşağıda verilmiştir.

```
>>net=newff(normegitimgirdi,normegitimcikti,[n1],{t1,t2},'trainlm','learnqdm','mse');
>>view(net);
>>net.trainParam.epochs=3000;
>>net.trainParam.show=NaN;
>>net.trainParam.max_fail=3000;
>>net.trainParam.goal=1e-10000000000000000000;
>>net.trainParam.min_grad=1e-10000000000000000000;
```

Burada 'n1' birinci gizli katmandaki nöron sayısını, '{t1,t2}' ise sırasıyla birinci gizli katmandaki ve çıkış katmanındaki aktivasyon fonksiyonlarını ifade etmektedir. Öncelikle 't1=logsig', 't2=tansig' olarak kabul edilmiştir. Ardından ağı eğitimi için 3000 iterasyon yeterli görülmüştür. Hedeflenen hata değeri olarak $10^{-10000000000000000000}$ gibi çok küçük bir değer seçilmiştir. Sistem bu hata değerine ulaştığı zaman durdurulacaktır. Eğitim gerçekleşirken değerlerin belirli aralıklarla gösterilmesi yerine eğitim bittiği zaman gösterilmesi için 'show' parametresi 'NaN' olarak verilmiştir.

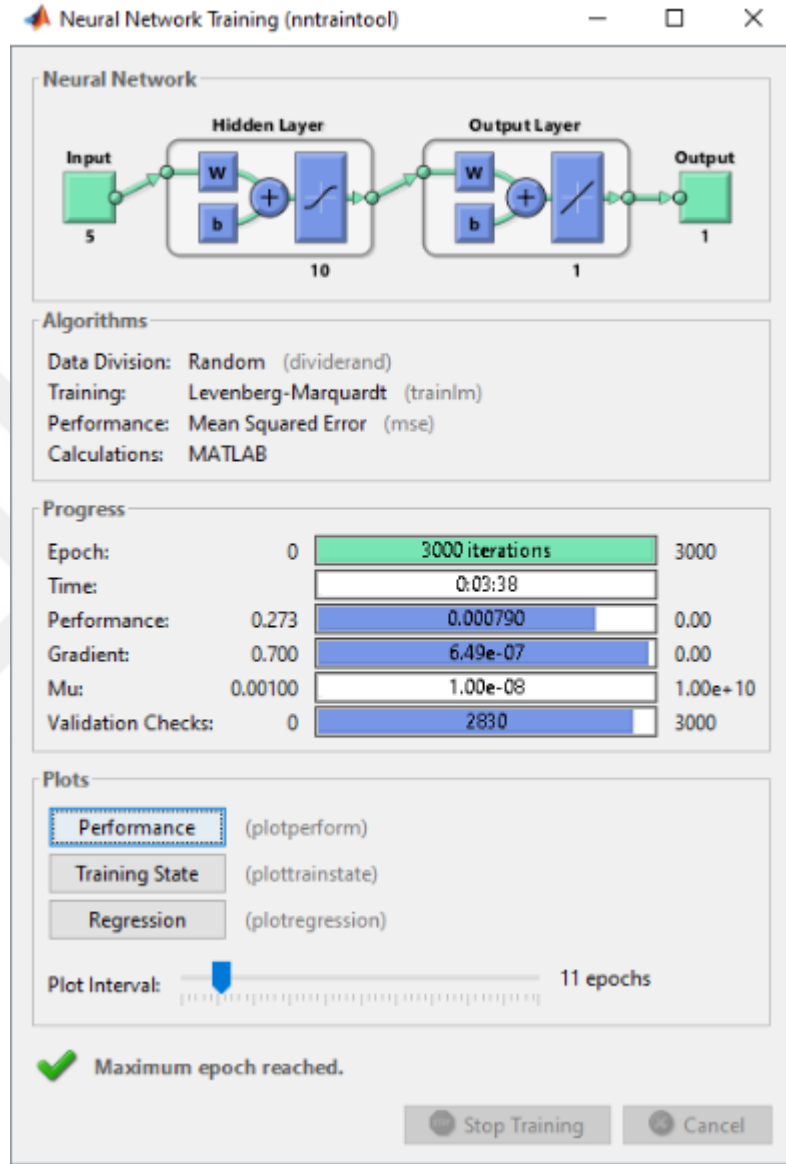
Tablo 3.4'te oluşturulan ağ yapılarına ait MSE değerleri verilmiştir. Bu denemeler sonucunda minimum MSE değerine sahip olan 10 gizli nöronlu ağ yapısı seçilmiştir.

Giriş katmanında 5 nöron, gizli katmanında 10 nöron ve çıkış katmanında 1 nöron bulunan ağ yapısı dikkate alınarak aktivasyon fonksiyonunun ve eğitim fonksiyonunu tespiti için çeşitli denemeler yapılmıştır. Aktivasyon fonksiyonu olarak logsig, tansig ve purelin, eğitim fonksiyonu olarak trainlm, trainbr, trainbfg ve trainrp kullanılarak farklı kombinasyonlar denenmiştir. Sonuçlar tablo 3.5'te verilmiştir.

Tablo 3.5 Aktivasyon fonksiyonunun ve eğitim fonksiyonunun tespiti için yapılan denemeler

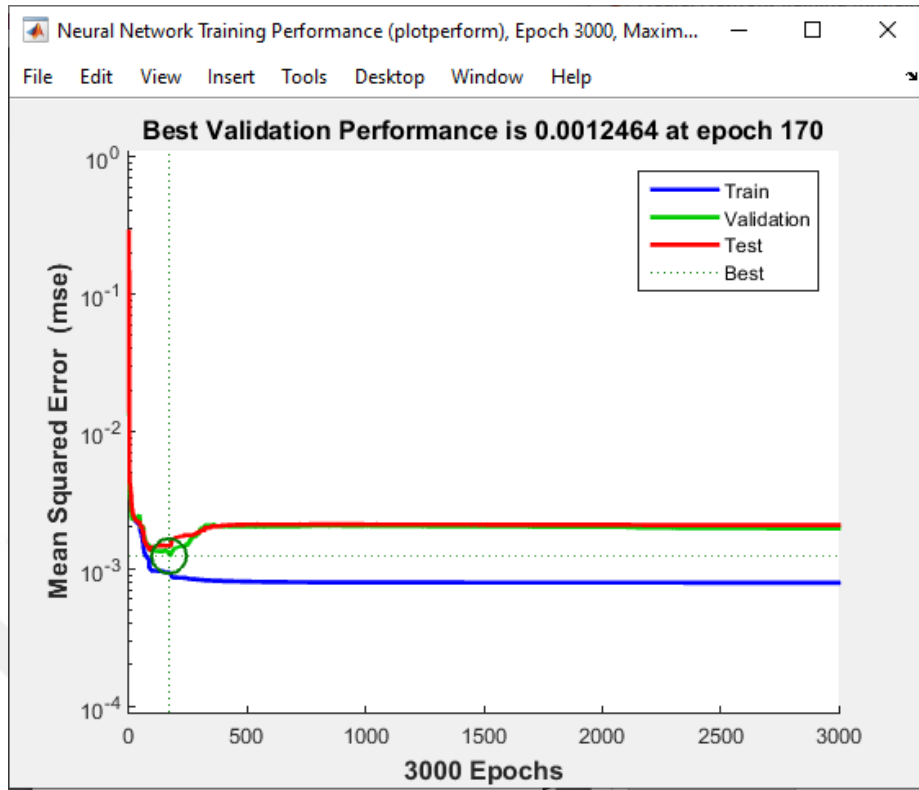
No	Network	Layer-1	Layer-2	Learning Func.	MSE
1	5,10,1	logsig	logsig	trainlm	0,0256280
2	5,10,1	logsig	purelin	trainlm	0,0019550
3	5,10,1	logsig	tansig	trainlm	0,0021874
4	5,10,1	tansig	logsig	trainlm	0,0275290
5	5,10,1	tansig	purelin	trainlm	0,0012464
6	5,10,1	tansig	tansig	trainlm	0,0012968
7	5,10,1	logsig	logsig	trainrp	0,0300250
8	5,10,1	logsig	purelin	trainrp	0,0034661
9	5,10,1	logsig	tansig	trainrp	0,0023444
10	5,10,1	tansig	logsig	trainrp	0,0293460
11	5,10,1	tansig	purelin	trainrp	0,0018011
12	5,10,1	tansig	tansig	trainrp	0,0027819
13	5,10,1	logsig	logsig	trainbfg	0,0295670
14	5,10,1	logsig	purelin	trainbfg	0,0026482
15	5,10,1	logsig	tansig	trainbfg	0,0016474
16	5,10,1	tansig	logsig	trainbfg	0,0256110
17	5,10,1	tansig	purelin	trainbfg	0,0032308
18	5,10,1	tansig	tansig	trainbfg	0,0018138
19	5,10,1	logsig	logsig	trainbr	0,0275170
20	5,10,1	logsig	purelin	trainbr	0,0027489
21	5,10,1	logsig	tansig	trainbr	0,0024242
22	5,10,1	tansig	logsig	trainbr	0,0269600
23	5,10,1	tansig	purelin	trainbr	0,0032044
24	5,10,1	tansig	tansig	trainbr	0,0020840

Tablo 3.5 incelendiğinde minimum MSE değerini verdiği için gizli katmanda ‘tansig’, çıkış katmanında ‘purelin’ aktivasyon fonksiyonu ve ‘trainlm’ eğitim fonksiyonu seçilmiştir.



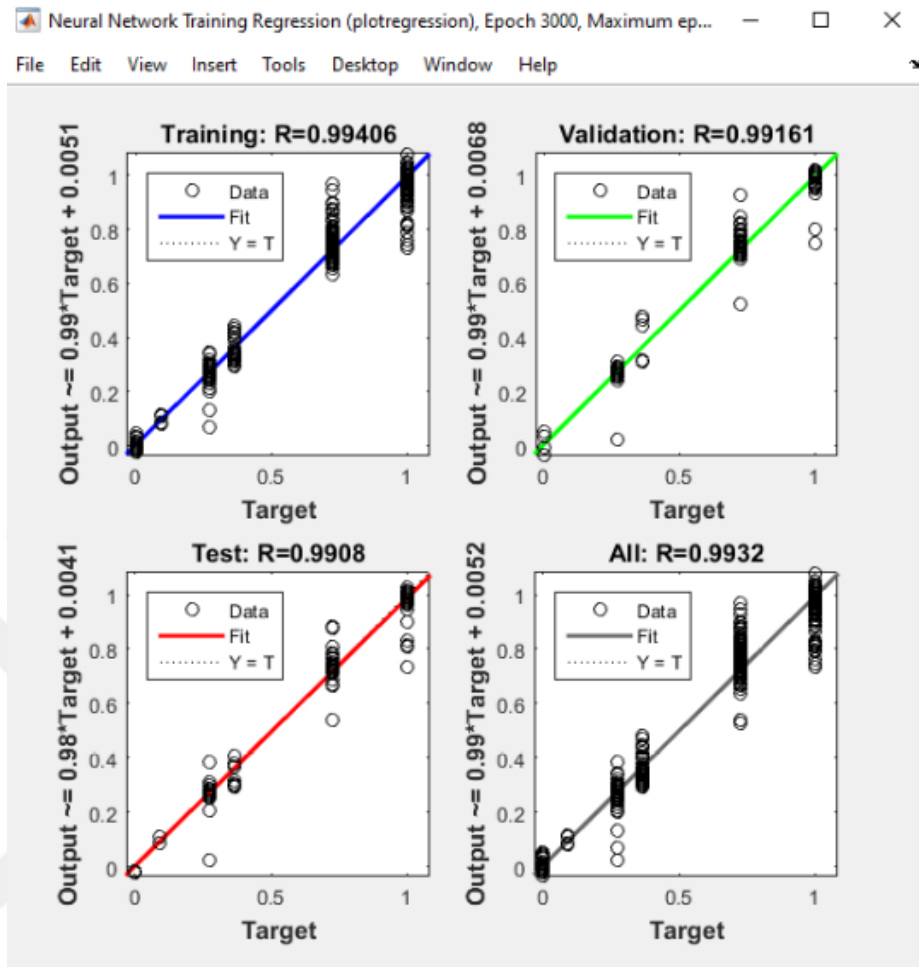
Şekil 3.2 Matlab'ta YSA eğitim sonuç ekranı

Şekil 3.2'de görüldüğü gibi ağı eğitimi 3000 iterasyonda en iyi sonuca ulaşmıştır. Eğitim sonucu oluşan hata değerinin iterasyona göre değişimi Şekil 3.3'te gösterilmiştir.



Şekil 3.3 Eğitim, geçerleme ve test verilerine ait hata performansları

Şekil 3.3'ten anlaşılacağı gibi 170 iterasyon sonunda ağın MSE değerine (0,0012464) ulaşılmıştır. Tasarlanan ağa ait regresyon değerleri şekil 3.4'te verilmiştir. Regresyon değerleri (0,1) arasında değer alır. Bu değer 1'e yaklaştığı zaman ağın tahmininin iyi olduğunu söyleyebiliriz.



Şekil 3.4 Matlab regresyon sonuçları

Eğitilen ağ yapısı ((5,10,1),{'tansig','purelin'},'trainlm','learngdm','mse') test edilmiştir. Sinir ağına eğitilirken tanıtılmamış 900 adet test verisi girilmiştir ve ağın bu verilere ait çıktıyı tahmin etmesi istenmiştir. Ağı test etmek için gerekli kodlar aşağıda verilmiştir.

```
>>net=train(net,normegitimgirdi,normegitimcikti);
>>normtahmin=sim(net,normtestgirdi);
>>normtahmin=normtahmin';
>>tahmin=(normtahmin*(maxtestcikti-mintestcikti)+ (mintestcikti);
```

Tablo 3.6'da gerçek çıktı ile ağın tahmin ettiği çıktıların bir kısmı gösterilmiştir. Hata değeri aşağıdaki gibi formülize edilmektedir.

$$\text{Hata} = | \text{Gerçek değer} - \text{Tahmin değer} | \quad (3.2)$$

Tablo 3.6 Gerçek değerler ile minimum hata değerine sahip ağ yapısının test edilmesi sonucu elde edilen tahmin değerlerinin karşılaştırılması

Gerçek	Tahmin	Hata	MAPE
14,2	14,38	0,18	1,2676
14,2	14,37	0,17	1,1972
14,2	14,37	0,17	1,1972
14,2	14,41	0,21	1,4789
14,2	14,38	0,18	1,2676
14,2	14,39	0,19	1,3380
14,2	14,36	0,16	1,1268
14,2	14,38	0,18	1,2676
14,2	14,36	0,16	1,1268
14,2	14,37	0,17	1,1972
14,2	14,38	0,18	1,2676
14,2	14,38	0,18	1,2676
14,2	14,35	0,15	1,0563
14,2	14,37	0,17	1,1972
14,2	14,35	0,15	1,0563
14,2	14,37	0,17	1,1972
14,2	14,35	0,15	1,0563
14,2	14,37	0,17	1,1972
14,2	14,38	0,18	1,2676

Çalışmada ortalama 1,1005’lik bir oranla MAPE değeri elde edilmiştir. Gerçek çıktı değerlerinin ortalaması 14,27 iken ağı tahmin ettiği çıktı değerlerinin ortalaması 14,36’dır. Tek gizli katman dikkate alınarak yapılan denemeler sonucunda 0,0012464 MSE değerini veren ((5,10,1),{‘tansig’,‘purelin’},’trainlm’,’learnngdm’,’mse’) ağ yapısı seçilmiştir. ‘Daha düşük MSE değeri elde edebilir miyiz?’ diye iki gizli katmanlı ağ yapısı oluşturulmuş ve çeşitli denemeler yapılmıştır. En iyi sonucu veren ağ yapısını bulmak için ileri beslemeli (feed forward) iki gizli katmanlı bir ağ yapısı oluşturulmuştur. Eğitim fonksiyonu olarak ‘trainlm’, birinci gizli katmadaki nöron sayısı ise 10 olarak seçilmiştir ve ikinci gizli katmandaki nöron sayısının ve aktivasyon fonksiyonunun tespiti için çeşitli denemeler yapılmıştır.

Tablo 3.7 İkinci gizli katmandaki nöron sayısının ve aktivasyon fonksiyonunun tespiti için yapılan denemeler

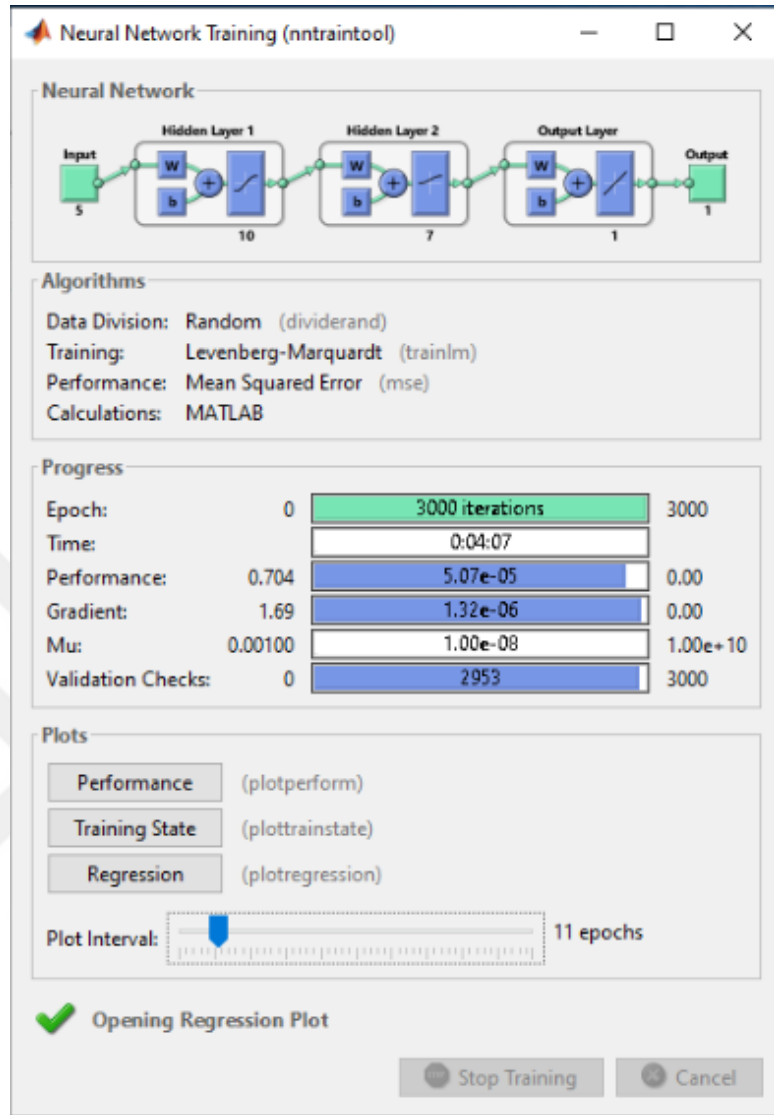
No	Network	Layer-1	Layer-2	Layer-3	Learning Func.	MSE
1	5,10,2,1	tansig	logsig	purelin	trainlm	0,00140140
2	5,10,2,1	tansig	tansig	purelin	trainlm	0,00139810
3	5,10,3,1	tansig	logsig	purelin	trainlm	0,00167580
4	5,10,3,1	tansig	tansig	purelin	trainlm	0,00209060
5	5,10,4,1	tansig	logsig	purelin	trainlm	0,00115300
6	5,10,4,1	tansig	tansig	purelin	trainlm	0,00152260
7	5,10,5,1	tansig	logsig	purelin	trainlm	0,00183980
8	5,10,5,1	tansig	tansig	purelin	trainlm	0,00122000
9	5,10,6,1	tansig	logsig	purelin	trainlm	0,00164820
10	5,10,6,1	tansig	tansig	purelin	trainlm	0,00097130
11	5,10,7,1	tansig	logsig	purelin	trainlm	0,00072320
12	5,10,7,1	tansig	tansig	purelin	trainlm	0,00145170
13	5,10,8,1	tansig	logsig	purelin	trainlm	0,00130470
14	5,10,8,1	tansig	tansig	purelin	trainlm	0,00101200
15	5,10,9,1	tansig	logsig	purelin	trainlm	0,00165180
16	5,10,9,1	tansig	tansig	purelin	trainlm	0,00166300
17	5,10,10,1	tansig	logsig	purelin	trainlm	0,00147190
18	5,10,10,1	tansig	tansig	purelin	trainlm	0,00170300
19	5,10,15,1	tansig	logsig	purelin	trainlm	0,00273190
20	5,10,15,1	tansig	tansig	purelin	trainlm	0,00064498

Tablo 3.7’de görüldüğü gibi birinci gizli katmanda ‘tansig’, çıktı katmanında ‘purelin’ ve ikinci gizli katmanda logsig, tansig aktivasyon fonksiyonları dikkate alınarak oluşturulan yapay sinir ağı için gerekli kodlar aşağıda verilmiştir.

```
>>net=newff(normegitimgirdi,normegitimcikti,[n1,n2],[t1,t2,t3],’trainlm’,’learnngd
m’,’mse’);
>>view(net);
>>net.trainParam.epochs=3000;
>>net.trainParam.show=NaN;
>>net.trainParam.max_fail=3000;
>>net.trainParam.goal=1e-1000000000000000000;
>>net.trainParam.min_grad=1e-1000000000000000000;
```

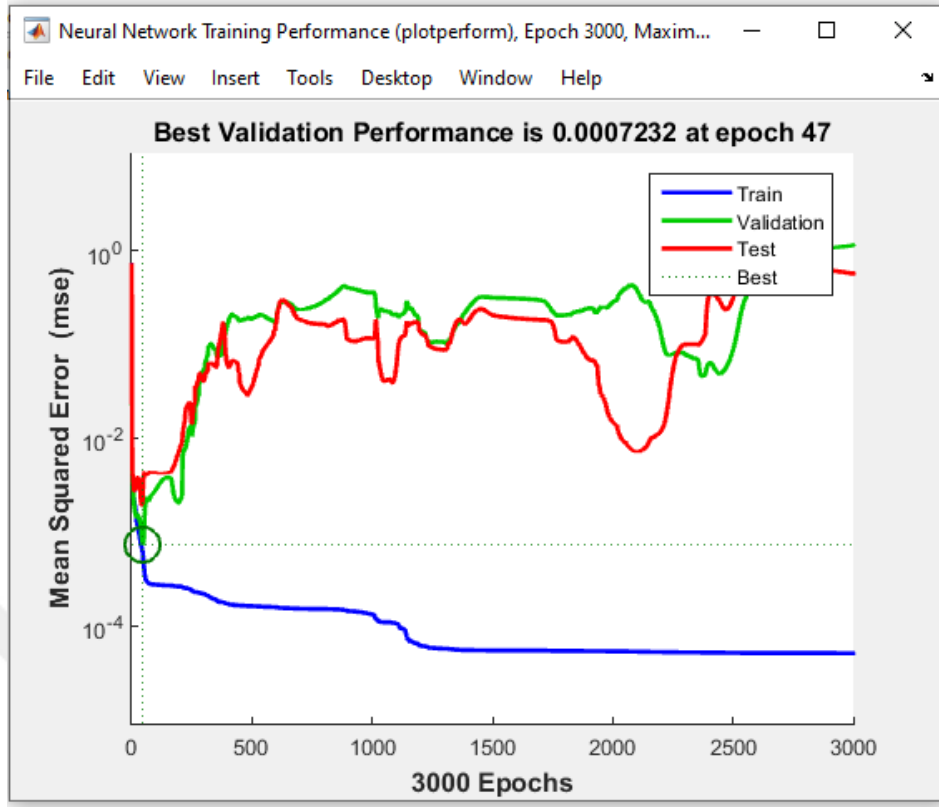
Burada ‘n1’ birinci gizli katmandaki nöron sayısını, ‘n2’ ikinci gizli katmandaki nöron sayısını ‘{t1,t2,t3}’ ise sırasıyla birinci gizli katmandaki, ikinci gizli katmandaki ve çıkış katmanındaki aktivasyon fonksiyonlarını ifade etmektedir. ‘t1=tansig’, ‘t3=purelin’ olarak kabul edilmiştir. ‘t2’ ise tansig ve logsig kullanılarak farklı kombinasyonlar oluşturulmuştur. Ardından ağı eğitimi için 3000 iterasyon yeterli görülmüştür. Hedeflenen hata değeri olarak $10^{-1000000000000000000}$ gibi çok küçük bir değer seçilmiştir. Sistem bu hata değerine ulaştığı zaman durdurulacaktır. Eğitim gerçekleşirken değerlerin belirli aralıklarla gösterilmesi yerine eğitim bittiği zaman gösterilmesi için ‘show’ parametresi ‘NaN’ olarak verilmiştir.

Tablo 3.7’de oluşturulan ağ yapılarına ait MSE değerleri verilmiştir. İkinci gizli katmanında 15 nörona sahip ağ minimum MSE değerini vermesine rağmen seçilmemiştir. Çünkü nöron sayısı fazla olduğunda ağı ezberlemiş olma ihtimali bulunmaktadır. Bundan dolayı minimum MSE değerine sahip olmamasına rağmen ikinci gizli katmanında 7 nörona sahip ağ yapısı seçilmiştir.



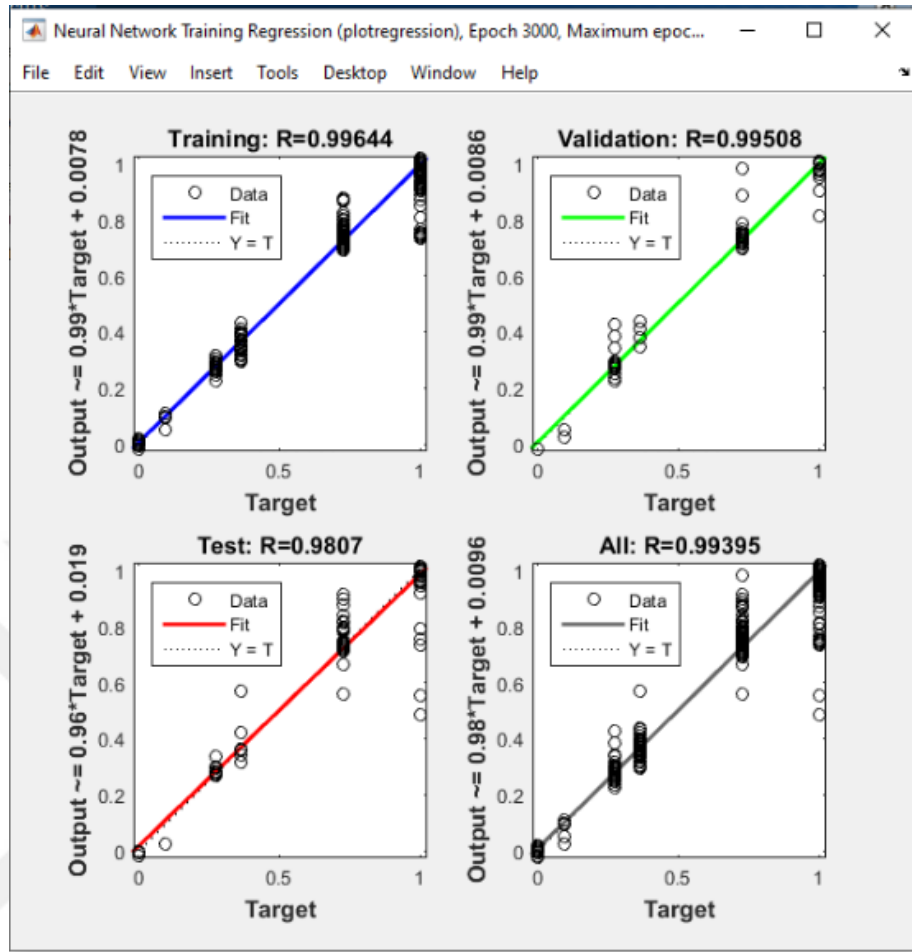
Şekil 3.5 Matlab'ta YSA eğitim sonuç ekranı

Şekil 3.5'te görüldüğü gibi ağın eğitimi 3000 iterasyonda en iyi sonuca ulaşmıştır. Eğitim sonucu oluşan hata değerinin iterasyona göre değişimi şekil 3.6'da gösterilmiştir.



Şekil 3.6 Eğitim, geçerleme ve test verilerine ait hata performansları

Şekil 3.6'dan anlaşılabacağı gibi 47 iterasyon sonunda ağın MSE değerine (0,0007232) ulaşılmıştır. Tasarlanan ağa ait regresyon değerleri şekil 3.7'de verilmiştir. Regresyon değerleri (0,1) arasında değer alır. Bu değer 1'e yaklaştığı zaman ağın tahmininin iyi olduğunu söyleyebiliriz.



Şekil 3.7 Matlab regresyon sonuçları

Eğitilen ağ yapısı ((5,10,7,1),{'tansig','logsig','purelin'},'trainlm','leangdm','mse') test edilmiştir. Sinir ağına eğitilirken tanıtılmamış 900 adet test verisi girilmiştir ve ağın bu verilere ait çıktıyı tahmin etmesi istenmiştir. Ağı test etmek için gerekli kodlar aşağıda verilmiştir.

```
>>net=train(net,normegitimgirdi,normegitimcikti);
>>normtahmin=sim(net,normtestgirdi);
>>normtahmin=normtahmin';
>>tahmin=(normtahmin*(maxtestcikti-mintestcikti)+ (mintestcikti);
```

Tablo 3.8’de gerçek çıktı ile ağı tahmin ettiği çıktıların bir kısmı gösterilmiştir. Hata değeri aşağıdaki gibi formülize edilmektedir.

$$\text{Hata} = | \text{Gerçek değer} - \text{Tahmin değeri} | \quad (3.3)$$

Tablo 3.8 Gerçek değerler ile minimum hata değerine sahip ağ yapısının test edilmesi sonucu elde edilen tahmin değerlerinin karşılaştırılması

Gerçek	Tahmin	Hata	MAPE
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155
14,2	14,16	0,04	0,2817
14,2	14,11	0,09	0,6338
14,2	14,33	0,13	0,9155
14,2	14,35	0,15	1,0563
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155
14,2	14,08	0,12	0,8451
14,2	14,24	0,04	0,2817
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155
14,2	14,34	0,14	0,9859
14,2	14,33	0,13	0,9155
14,2	14,28	0,08	0,5634
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155
14,2	14,33	0,13	0,9155

Çalışmada ortalama 1,0345’lik bir oranla MAPE değeri elde edilmiştir. Gerçek çıktı değerlerinin ortalaması 14,27 iken ağın tahmin ettiği çıktı değerlerinin ortalaması 14,32’dir.

Kabul edilen ağ yapısına ait bilgiler aşağıda verilmiştir:

Tablo 3.9 Kabul edilen ağ yapısına ait bilgiler

Ağ tipi	İleri Beslemeli
Eğitim Fonksiyonu	Trainlm
Öğrenme Fonksiyonu	Learngdm
Katman Sayısı	4
Gizli Katman Sayısı	2
1.gizli katmandaki nöron sayısı	10
2.gizli katmandaki nöron sayısı	7
1.gizli katmandaki aktivasyon fonksiyonu	tansig
2.gizli katmandaki aktivasyon fonksiyonu	logsig
Çıkış katmanındaki aktivasyon fonksiyonu	purelin
MSE	0,0007232
R ² (Train)	0,99644
R ² (Test)	0,98070
R ² (All)	0,99395

4. BÖLÜM

SONUÇ VE DEĞERLENDİRME

4.1 Sonuç ve Değerlendirme

Tez kapsamında yapılan bu çalışmada, yapay sinir ağı teknikleri kullanılarak ısı yalıtım levhasının hammaddesi olan boncukların üretiminde istenilen yoğunluğun tahmini için model geliştirilmiştir. En iyi sonucu veren ağ yapısını tespit etmek için farklı gizli nöron sayısı, aktivasyon fonksiyonu ve eğitim fonksiyonu kullanılarak çeşitli denemeler yapılmıştır.

Çalışmada tasarlanan ağ yapısının ağırlık, ölçülen yoğunluk, çevrim süresi, buhar süresi, buhar sıcaklığı olmak üzere 5 adet girdi parametresi ve istenilen yoğunluk olarak 1 adet çıktı parametresi vardır. Giriş katmanında girdi parametrelerine ait beş nöron, çıkış katmanında ise çıktı parametresine ait bir nöron bulunmaktadır.

Yukarıda bahsedilen parametrelere ait 3000 adet veriden 2100 adet veri girdi, 900 adet veri ise çıktı olarak ağın eğitilmesi ve test edilmesi için kullanılmıştır. Veriler kullanılmadan önce ağ tarafından $[0,1]$ aralığında normalize edilmiştir. En iyi sonucu veren ağ yapısını tespiti için tek gizli katmanlı ve iki gizli katmanlı ileri beslemeli geri yayılım ağ yapıları oluşturulmuştur. Çünkü yapılan literatür çalışmasında birçok problem için bir veya iki gizli katmanlı ağ yapılarının yeterli olduğu görülmüştür.

Öncelikle tek gizli katmanlı ağ yapısı oluşturularak gizli nöron sayısı, aktivasyon fonksiyonu ve eğitim fonksiyonu için denemeler yapılmıştır. Oluşturulan farklı sinir ağı yapıları MATLAB R2015 programı kullanılarak analiz edilmiştir. Gizli katmandaki nöron sayısı 2-10 ve 15 olarak belirlenmiştir. Çünkü gizli katmandaki nöron sayısı ağın

performansı için önemli bir etkidir. Nöron sayısı çok fazla olduğu zaman ağın ezberlemiş olma ihtimali yüksektir. Bu nedenden dolayı en iyi sonuca ulaşmak için nöron sayısına dikkat edilmelidir.

Aktivasyon fonksiyonu olarak logsig, tansig ve purelin, eğitim fonksiyonu olarak trainlm, trainbr, trainbfg ve trainrp kullanılarak farklı kombinasyonlar denenmiştir. Tasarlanan ağ yapısında aktivasyon fonksiyonu olarak gizli katmanda ‘tansig’, çıkış katmanında ‘purelin’ kabul edilmiştir ve eğitim fonksiyonu olarak ‘trainlm’ seçilmiştir. Eğitim sonucunda elde edilen sonuçların değerlendirilmesi için performans fonksiyonu olarak MSE (Mean Square Error) seçilmiştir. Elde edilen sonuçlar tablo ve grafik şeklinde gösterilmiştir. Denemeler sonucunda minimum MSE değerine (0,0012464) sahip olan 10 gizli nöronlu ağ yapısı seçilmiştir. Ağın test edilmesi sonucu elde edilen çıktı ile gerçek çıktı arasındaki fark yani hata değerleri dikkate alınarak 0,16 ortalama hata değerine ulaşılmıştır.

Ardından giriş katmanında 5 nöron, birinci gizli katmanında 10 nöron ve çıkış katmanında 1 nöron bulunduran, eğitim fonksiyonu ‘trainlm’ olarak seçilmiş iki gizli katmanlı ağ yapısı oluşturularak ikinci gizli katmandaki nöron sayısının ve aktivasyon fonksiyonunun tespiti için çeşitli denemeler yapılmıştır. Denemeler sonucunda 0,0007232 MSE değerine sahip ikinci gizli katmanında 7 nöron bulunduran ağ yapısı seçilmiştir. Tasarlanan ağ yapısında aktivasyon fonksiyonu olarak birinci gizli katmanda ‘tansig’, ikinci gizli katmanda ‘logsig’ ve çıkış katmanında ‘purelin’ kabul edilmiştir. Ağın test edilmesi sonucu elde edilen tahmin değerleri ile gerçek değerler karşılaştırılmıştır. Aralarındaki fark yani hata değerleri dikkate alınarak 0,13 ortalama hata değerine ulaşılmıştır. Değerler arasındaki fark tasarlanan ağın tutarlı olduğunu göstermektedir.

Uygulamada kullanılan diğer iki yöntem veri madenciliği ve çoklu regresyon analizidir. Veri madenciliği modellerinden tahmin edici modeller kullanılarak PolyAnalyst programı yardımıyla veriler analiz edilmiştir ve en yüksek R^2 değerine (0,8116) sahip bağımlılıkları bulma analizi verileri ile elde edilen formül kullanılarak istenilen yoğunluğa ulaşılabilir.

Çoklu regresyon analizinde ise bağımsız değişkenler dikkate alınarak bağımlı değişken için tahmin modelinin belirlenmesi amaçlanmıştır. SPSS programı yardımıyla regresyon

analizi gerçekleştirilmiştir ve korelasyonun 0,75; determinasyon katsayısının ise $R^2 = 0,563$ olduğu sonucuna ulaşılmıştır.

Tablo 4.1 Elde edilen R^2 değerlerinin karşılaştırılması

Veri Madenciliği	Regresyon Analizi	Yapay Sinir Ağları
0,8116	0,5630	0,9964

Tablo 4.1 incelendiğinde en yüksek R^2 değeriyle en iyi sonucu yapay sinir ağlarının verdiği görülmüştür.



KAYNAKÇA

1. Keskinçilic, F., 2010. Yapay Sinir Ağları İle Döküm Parçaların Sertliğine Etki Eden Parametrelerinin Analizi Ve Bir Uygulama. Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kırıkkale, 68 s.
2. Mashhadban, H., Kutanaei, S. S. ve Sayarinejad, M., A. 2016. Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network, **Construction and Building Materials**, **119**: 277-287.
3. Akkaya, G., Gökçen, T., 2006. Job-shop scheduling design with artificial neural networks. Journal of Engineering and Natural Sciences. Yıldız Teknik Üniversitesi, Makine Fakültesi, Endüstri Mühendisliği Bölümü, İstanbul.
4. Satyanarayana, G., Naidu, G., Babu. N., 2017. Artificial neural network and regression modelling to study the effect of reinforcement and deformation on volumetric wear of red mud nano particle reinforced aluminium matrix composites synthesized by stir casting. Boletín de la Sociedad Española de Cerámica y Vidrio. 57: 91-100.
5. Wilson, R. L., & Sharda, R. 1994. Bankruptcy prediction using neural networks. **Decision support systems**, **11**(5), 545-557.
6. Kant, G., Sangwan, K., 2015. Predictive modelling for energy consumption in machining using artificial neural network. Procedia CIRP, 37: 205 – 210.
7. Sudha, L., Sudha, R., Dillibabu, Srivatsa Srinivas, S., Annamalai A., 2015. Optimization of process parameters in feed manufacturing using artificial neural network. **Computers and Electronics in Agriculture**, **120**: 1-6

8. Vedaramana, N., Sandhyaa, K.V., Charukeshb, N.R.B., Venkatakrishnana, B., Haribabub, K., Sridharana, M.R., Nagarajanc, R., 2017. Ultrasonic extraction of natural dye from *Rubia Cordifolia*, optimisation using response surface methodology (RSM) & comparison with artificial neural network (ANN) model and its dyeing properties on different substrates. **Chemical Engineering and Processing**, **114**:46–54.
9. Altınkaya, H., Orak, İ., Esen, İ., 2014. Artificial neural network application for modeling the rail rolling process. **Expert Systems with Applications**, **41**: 7135–7146.
10. Hatipoğlu, T., Boran, S., Özcan, B., Fığlalı, A., 2013. Yapay sinir ağı yaklaşımıyla çinko kalınlığının tahminlenmesi. **SAÜ Fen Bilimleri Dergisi**, **17** (1):61-69.
11. Cai, M., Yin, Y., Xie, M., 2009. Prediction of hourly air pollutant concentrations near urban arterials using artificial neural network approach. *Transportation Research Part D: Transport and Environment*, **14** (1):32-41.
12. Mia, M., Dhar, N., 2016. Prediction of surface roughness in hard turning under high pressure coolant using artificial neural network. *Measurement*, **92**:464-474.
13. Li, X., Feng, C., Sun, D., Tao, M., 2015. Predicting menopausal symptoms with artificial neural network. **Expert Systems With Applications**, **42**: 8698-8706.
14. Şimşek, E., Alkay, T., 2019. Artificial neural network approach for the prediction of effluents streams from a wastewater treatment plant: a case study in Kocaeli (Turkey). **Mugla Journal of Science and Technology**, **164**-171.
15. Dombaycı, A., 2010. The prediction of heating energy consumption in a model house by using artificial neural networks in Denizli–Turkey. **Advances in Engineering Software**, **41** (2): 141-147.
16. Zain, A., Haron, H., Sharif, S., 2010. Prediction of surface roughness in the end milling machining using Artificial Neural Network. **Expert Systems with Applications**, **37** (2): 1755-1768.

17. Jorjani, E., Mesroghli, S., Chelgani, S., 2008. Prediction of operational parameters effect on coal flotation using artificial neural network. **Journal of University of Science and Technology**, **15** (5): 528-5
18. Tiryaki, S., Özşahin, Ş., Yıldırım, İ., 2014. Comparison of artificial neural network and multiple linear regression models to predict optimum bonding strength of heat treated woods. **International Journal of Adhesion and Adhesives**, **55**: 29-36.
19. Nabipour, M., 2018. Prediction of surface tension of binary refrigerant mixtures using artificial neural network. **Fluid Phase Equilibria**, **456**: 151-160.
20. Ekonomou, L., 2010. Greek long term energy consumption prediction using artificial neural networks. **Energy**, **35** (2): 512-517.
21. Yavuz, S., Deveci, M., 2012. İstatiksel normalizasyon tekniklerinin yapay sinir ağı performansına etkisi. **Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, **40**: 167-187.
22. Öztemel, E., 2006. Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul, 232s.
23. Öztürk, K., Şahin, M., 2018. Yapay sinir ağları ve yapay zeka'ya genel bir bakış. **Takvim-i Vekayi**, **6** (2): 25-36.
24. Yürük, M., Ekşi, İ., 2019. Yapay zeka yöntemleri ile işletmelerin finansal başarısızlığının tahmin edilmesi: Bist imalat sektörü. **Mukaddime**, **10** (1): 393-422.
25. Karaatlı, M., Helvacıoğlu, C., Ömürbek, N., Tokgöz, G., 2012. Yapay sinir ağları yöntemi ile otomobil satış tahmini. **Uluslararası Yönetim İktisat ve İşletme Dergisi**, **8** (17): 1-14.
26. Abiodun, I., Jantan, A., Omolara, E., Dada, V., Mohamed, A., Arshad, H., 2018. State of the art in artificial neural network applications: **A survey**. **Heliyon**, **4** (11): 1-4.
27. Öztürk, K., Şahin, M., 2018. Yapay sinir ağları ve yapay zeka'ya genel bir bakış. **Takvim-i Vekayi**, **6** (2): 25-36.

28. Mijwel, M. M., 2018. Artificial Neural Networks Advantages and Disadvantages.
29. Ballı, M., 2014. Yapay Sinir Ağları ile Talep Tahmini ve Gıda Sektöründe Uygulanması. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 180 s.
30. Aydın Ö., Elektronik Harp İle Toplanan Verilerin Veri Madenciliği Yöntemleri İle Analiz Edilmesi. Bahçeşehir Üniversitesi, Yüksek Lisans Tezi, İstanbul, 94 s, 2017.
31. Akpınar, H., Veri tabanlarında bilgi keşfi ve veri madenciliği. **İstanbul Üniversitesi İşletme Fakültesi Dergisi**, **29**: 1-22, 2000.
32. Fayyad U., Piatetsky-Shapiro G., Smyth P., From data mining to knowledge discovery in databases. **AI Magazine**, **17** (3): 37-54, 1996.
33. Asilkan Ö., Veri Madenciliği Kullanılarak İkinci El Otomobil Pazarında Fiyat Tahmini. Akdeniz Üniversitesi, Doktora Tezi, Antalya, 170 s., 2008.
34. Hoş, S., 2020. Çoklu doğrusal regresyon analizi güvenilirliğinin Jackknife tekniği ile sınanmasına yönelik bir araştırma. **Yönetim ve Ekonomi araştırmaları Dergisi**, **18** (4): 304-316.
35. Toğa, G., 2012. Koroner Arter Hastalarında Stent İmplantasyonu Sonrası Ölümünün Yapay Sinir Ağları İle Modellenmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri, 112 s.
36. Budak, H., Erpolat, S., 2012. Kredi riski tahmininde yapay sinir ağları ve lojistik regresyon analizi karşılaştırılması. **Online Academic Journal of Information Technology**, **3** (9): 24-30.
37. Agatonovic-Kustrin, S., Beresford, R., 2000. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. **Journal of Pharmaceutical and Biomedical Analysis**, **22**: 717-727.
38. Cain, G., 2016. Artificial Neural Network- New Research. Nova Science Publishers, New York, 100 s.

39. Koç, M., Balas, C., Arslan, A., 2004. Taş dolgu dalgakıranların yapay sinir ağları ile ön tasarımı. **İMO Teknik Dergi**, **225**: 3351-3375.
40. Yeşilkanat, C., Kobyay, Y., Taşkın, H., Çevik, U., 2017. Spatial interpolation and radiological mapping of ambient gamma dose rate by using artificial neural networks and fuzzy logic methods. *Journal of Environmental Radioactivity*, **175-176**: 78-93.
41. Sevinçtekin, E., 2014. İmalat Sektöründe Yapay Sinir Ağları Uygulaması. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 84 s.
42. Turkson, R., Yan, F., Ali, M., Hu, J., 2016. Artificial neural network applications in the calibration of spark-ignition engines: an overview. **Engineering Science and Technology, an International Journal**, **19** (3): 1346-1359.
43. Güler, İ., Übeyli, D., 2006. Çok katmanlı perseptron sinir ağları ile diyabet hastalığının teşhisi. **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, **21** (2): 319-326.
44. Ataseven, B. 2013. Yapay sinir ağları ile öngörü modellemesi. **Öneri Dergisi**, **10** (39): 101-115.
45. Bolat, S., Kalenderli, Ö., 2003. Levenberg-Marquardt algoritması kullanılan yapay sinir ağları ile elektrot biçim optimizasyonu, XII. Turkish Symposium on Artificial Intelligence and Neural Networks, 02-04 Temmuz, 2003, Çanakkale.
46. Beale, M., Hagan, M., Demuth, H., 2017. Neural Network Toolbox 7 User's Guide. Natick, MA, 931 s.
47. Yılmaz, B., 2019. Maliyet fonksiyonun belirlenmesinde yapay sinir ağı modellerinin kullanımı, s. 329-344. **V. Uluslararası Muhasebe ve Finans Sempozyumu**, **01-04** Mayıs, 2019, Bursa.
48. Haykin, S., 2008. Neural Network and Learning Machines, Third Edition. Pearson Education Ltd., New Jersey, 938 s.

49. Ballı, M., 2014. Yapay sinir ağıları ile talep tahmini ve gıda sektöründe uygulanması. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 180 s.
50. Yazıcı, H., 2010. Yapay Sinir Ağları İle Banka Müşterisi Bekleme Süresi Tahmini. Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Erzurum, 76 s.
51. Sezgin, N., Tekin, R., Çalışkan, A., 2012. Örüntü tanımda hopfield ağıının kullanılması. **Batman Üniversitesi Yaşam Bilimleri Dersigi**, **1** (2): 257-266.
52. Oğuzlar, A., 2005. Kümeleme analizinde yeni bir yaklaşım: kendini düzenleyen haritalar (Kohonen Ağları). **İktisadi ve İdari Bilimler Dergisi**, **19** (2): 94-107.
53. Keçe, A., 2006. Yapay Sinir Ağları ile Plastik Enjeksiyon Süreci Başlangıç Parametrelerinin Belirlenmesi. Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Bursa, 88 s.
54. Zhang, G., Patuwo, B. Hu, M., 1998. Forecasting with artificial neural networks: The state of the art. **International journal of forecasting**, **14** (1): 35-62.
55. Kaastra, I., Boyd, M., 1996. Designing a neural network for forecasting financial and economic time series. **Neurocomputing**, **10** (3): 215-236.
56. Gürbüz, F., Buldanlı, B., 2019. Isı yalıtım levhası üretimi şişirme prosesinde boncuk yoğunluğunun tahmini. III. International scientific and Vocational studies Congress– Engineering (BILMES EN 2019), 27-30 Haziran, 2019, Nevşehir.

EKLER

EK-1 (Verilerin bir kısmı)

Malzeme	Ağırlık	Ölçülen Yoğunluk	Çevrim Süresi	Buhar Süresi	Buhar Sıcaklığı	İstenilen Yoğunluk
N5300-15-R	70,37	15,1	133	90	102,6	14,5
N5300-15-R	69,92	15	130	88	103,4	14,5
N5300-15-R	69,92	14,8	132	89	101,3	14,5
N5300-15-R	69,85	14,56	128	86	102,5	14,5
N5300-15-R	69,66	14,7	128	87	101,8	14,5
N5300-15-R	69,59	14,66	128	86	98,5	14,5
N5300-15-R	69,4	14,7	132	90	102,4	14,5
N5300-15-R	69,27	14,5	127	84	101,7	14,5
N5300-15-R	69,59	14,5	132	89	102,5	14,5
N5300-15-R	69,59	14,43	129	87	101,9	14,5
N5300-15-R	69,66	14,4	131	88	101,4	14,5
N5300-15-R	69,85	14,4	127	85	103,2	14,5
N5300-15-R	69,92	14,43	124	81	102	14,5
N5300-15-R	69,79	14,46	129	88	103,8	14,5
N5300-15-R	70,05	14,43	126	84	102,6	14,5
N5300-15-R	69,92	14,4	129	86	101,6	14,5
N5300-15-R	69,85	14,56	133	90	103,3	14,5
N5300-15-R	69,79	14,56	132	88	102,7	14,5
N5300-15-R	69,98	14,5	134	93	103,6	14,5
N5300-15-R	70,05	14,5	133	91	101,6	14,5
N5300-15-R	69,98	14,46	133	90	103,5	14,5
N5300-15-R	70,05	14,46	134	91	104,1	14,5
N5300-15-R	69,85	14,46	130	87	102,9	14,5
N5300-15-R	69,85	14,56	127	85	102,6	14,5
N5300-15-R	69,92	14,56	130	87	103,4	14,5
N5300-15-R	69,92	14,53	131	89	103,1	14,5
N5300-15-R	70,11	14,53	125	83	102,1	14,5
N5300-15-R	69,92	14,6	126	83	102,5	14,5
N5300-15-R	69,92	14,56	127	85	103,1	14,5
N5300-15-R	70,05	14,6	134	92	102,8	14,5
N5300-15-R	69,98	14,4	129	86	103,2	14,5
N5300-15-R	70,11	14,46	125	82	101,1	14,5
N5300-15-R	69,92	14,5	129	87	103,4	14,5
N5300-15-R	70,11	14,43	126	83	102,1	14,5

EK-2 (Çalışmada kullanılan MATLAB kodu)

function [tahmin normtahmin hata] = neuralnetwork(girdi,cikti,oran,n1,t1,t2)

% Detailed explanation goes here

% girdi: Gerçekleşen verilerin girdileri

% çıktı: Gerçekleşen verilerin çıktıları

% oran: Verilerin yüzde kaçının eğitim için kullanılacağını belirtir

% n1: Birinci gizli katmandaki nöron sayısı

% n2: İkinci gizli katmandaki nöron sayısı

% t1: Birinci gizli katmandaki aktivasyon fonksiyonu

% t2: İkinci gizli katmandaki aktivasyon fonksiyonu / Çıkış katmanındaki aktivasyon fonksiyonu

% t3: Çıkış katmanındaki aktivasyon fonksiyonu

% Verilerin programa tanıtılması

verisayisi=size(girdi,1);

oran=0.7;

vs=round(verisayisi*oran); *% Eğitim için kullanılacak veriler elde edilir*

egitimgirdi=girdi(1:vs,:);

testgirdi=girdi(vs+1:end,:);

egitimcikti=cikti(1:vs);

testcikti=cikti(vs+1:end);

minegitimgirdi=min(egitimgirdi);

minegitimcikti=min(egitimcikti);

mintestgirdi=min(testgirdi);

mintestcikti=min(testcikti);

maxegitimgirdi=max(egitimgirdi);

maxegitimcikti=max(egitimcikti);

maxtestgirdi=max(testgirdi);

maxtestcikti=max(testcikti);

% Verilerin normalize edilmesi (0,1)

```
normegitimgirdi=bsxfun(@rdivide,bsxfun(@minus,egitimgirdi,minegitimgirdi),maxegitimgirdi-minegitimgirdi);      % Eğitim için normalize girdi verisi
normegitimcikti=bsxfun(@rdivide,bsxfun(@minus,egitimcikti,minegitimcikti),maxegitimcikti-minegitimcikti);      % Eğitim için normalize çıktı verisi
normtestgirdi=bsxfun(@rdivide,bsxfun(@minus,testgirdi,mintestgirdi),maxtestgirdi-mintestgirdi);      % Test için normalize girdi verisi
normtestcikti=bsxfun(@rdivide,bsxfun(@minus,testcikti,mintestcikti),maxtestcikti-mintestcikti);      % Test için normalize çıktı verisi
```

% Verilerin sütunlarda olması için transpozunun alınması

```
normegitimgirdi=normegitimgirdi';
normtestgirdi=normtestgirdi';
normegitimcikti=normegitimcikti';
```

% Ağın oluşturulması

```
net=newff(normegitimgirdi,normegitimcikti,[n1],{t1,t2,t3},'trainlm','learngdm','mse');
view(net);
net.trainParam.epochs=3000;
net.trainParam.show=NaN;
net.trainParam.max_fail=3000;
net.trainParam.goal=1e-10000000000000000000;
net.trainParam.min_grad=1e-10000000000000000000;
```

% Ağın eğitilmesi

```
net=train(net,normegitimgirdi,normegitimcikti);
normtahmin=sim(net,normtestgirdi);
normtahmin=normtahmin';
tahmin=(normtahmin*(maxtestcikti-mintestcikti)) + (mintestcikti);      % normalize
olmayan tahmin verisi
hata=abs(normtestcikti-normtahmin);
end
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Burçak BULDANLI
Uyruğu: Türkiye (T.C.)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Endüstri Mühendisliği	2021
Lisans	Erciyes Üniversitesi, Endüstri Mühendisliği	2017
Lise	Kalaba Lisesi, ANKARA	2010

KATILDIĞI KONGRE/SEMPOZYUM

Gürbüz, F., Buldanlı, B., 2019. Isı yalıtım levhası üretimi şişirme prosesinde boncuk yoğunluğunun tahmini . III. International scientific and Vocational studies Congress – Engineering (BILMES EN 2019), 27-30 Haziran, 2019, Nevşehir.

YABANCI DİL

İngilizce